BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi dalam bidang kecerdasan buatan (*Artificial Intelligence*) dan visi komputer (*Computer Vision*) telah memberikan dampak signifikan terhadap berbagai aspek kehidupan, termasuk keamanan rumah tangga. Salah satu teknologi yang menjadi perhatian utama adalah sistem pendeteksian objek berbasis citra digital, yang mampu meningkatkan kenyamanan dan keamanan, terutama dalam pengelolaan akses kendaraan ke dalam area privat. Sistem ini tidak hanya mempermudah dalam akses otomatis, tetapi juga berfungsi sebagai lapisan perlindungan tambahan untuk mencegah akses oleh pihak yang tidak diizinkan (Putro, H. P., 2023).

Teknologi pendeteksian objek seperti *YOLO* (*You Only Look Once*) dikenal dengan kemampuannya untuk mendeteksi objek secara real-time dengan akurasi yang tinggi dan waktu komputasi yang efisien. *YOLO* menggunakan pendekatan jaringan saraf convolutional (*Convolutional Neural Network*) yang dirancang untuk mendeteksi banyak objek dalam sebuah gambar dengan kecepatan tinggi, menjadikannya sangat cocok untuk aplikasi berbasis perangkat keras berdaya rendah seperti *Raspberry Pi 4*. Dalam konteks ini, *YOLO* digunakan untuk mendeteksi area nomor plat kendaraan sebagai langkah awal dalam proses pengenalan (Rahman, R. A. H., 2024).

Selain pendeteksian objek, teknologi *OCR* (*Optical Character Recognition*) seperti *Tesseract* memungkinkan pembacaan teks pada gambar secara otomatis. *Tesseract* dikenal sebagai perangkat lunak *OCR open-source* yang handal dalam pengenalan teks dari berbagai bahasa, meskipun memerlukan kualitas input yang memadai agar dapat bekerja dengan akurasi optimal. Dengan integrasi antara *YOLO* dan *Tesseract OCR*, sebuah sistem dapat dikembangkan untuk mendeteksi dan membaca nomor plat kendaraan dengan tingkat akurasi yang tinggi (Suhartono, S., 2024).

Pemilihan *Raspberry Pi 4* sebagai perangkat keras utama dalam penelitian ini juga didasarkan pada kemampuannya untuk menjalankan algoritma berbasis kecerdasan buatan dalam lingkungan terbatas. Dengan spesifikasi yang mencakup prosesor yang lebih cepat dan dukungan untuk kamera berkualitas tinggi, *Raspberry Pi 4* menawarkan platform ideal untuk eksperimen dalam pengembangan sistem otomatis yang hemat daya dan ekonomis (Nugraha, F. A., 2020).

Meningkatnya kebutuhan masyarakat akan solusi keamanan yang terjangkau dan efisien mendorong pentingnya pengembangan sistem otomatisasi berbasis teknologi seperti ini. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem pendeteksi nomor plat kendaraan yang menggabungkan keunggulan *YOLO* dan *Tesseract OCR*, serta menguji implementasinya pada perangkat *Raspberry Pi 4*. Sistem ini diharapkan dapat menjadi solusi inovatif untuk mendukung keamanan di era digital saat ini (Rosidin, R., 2024).

B. Rumusan Masalah

1. Bagaimana cara mengintegrasikan algoritma *YOLO* untuk mendeteksi nomor plat kendaraan?
2. Bagaimana akurasi *Tesseract OCR* dalam membaca teks pada plat kendaraan?
3. Bagaimana performa pembacaan pada kondisi siang dan malam dengan jarak dan sudut kemiringan antara kamera dengan plat?

C. Tujuan Penelitian

1. Mengembangkan sistem pendeteksi nomor plat kendaraan menggunakan *YOLO* dan *Tesseract OCR*.
2. Mengukur tingkat akurasi dan performa sistem dalam mendeteksi dan membaca nomor plat kendaraan.
3. Menguji implementasi pada kondisi siang dan malam dengan menentukan jarak dan sudut kemiringan terhadap plat nomor dengan kamera.

D. Batasan Masalah

1. Algoritma deteksi objek menggunakan *YOLO* hanya berfokus pada identifikasi area plat nomor kendaraan.
2. Pengenalan teks pada plat nomor menggunakan Tesseract *OCR* terbatas pada teks alfanumerik dan tidak mendukung simbol atau karakter spesial lainnya.
3. faktor jarak dan sudut kemiringan plat kendaraan dengan kamera, serta pengaruh kondisi pencahayaan di siang dan malam .

E. Manfaat Penelitian

Memberikan kontribusi pada pengembangan teknologi pengenalan citra digital menggunakan perangkat keras sederhana. Agar dapat diketahui seberapa persen akurasi dalam mendeteksi dan membaca nomor plat kendaraan, sehingga mempermudah dalam pengumpulan database nomor plat kendaraan yang telah dideteksi.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Pendeteksi Plat Nomor

Deteksi plat nomor adalah sistem yang memanfaatkan video atau gambar untuk mengidentifikasi dan membaca plat nomor kendaraan. Sistem ini menggunakan pengenalan pola, pemprosesan gambar, dan *OCR* (*Otical Character Recognition*) untuk mengonversi gambar plat nomor menjadi teks yang dapat dibaca. Teknologi ini mendukung keamanan, manajemen lalu lintas, dan pengambilan keputusan berbasis data (Felisa, J., 2022).

Awalnya dikembangkan pada tahun 1980-an untuk mendukung penegakan hukum, deteksi plat nomor kini memiliki banyak aplikasi, seperti pembayaran tol, pengelolaan parkir, dan pelacakan pengiriman. Sistem ini bekerja dengan algoritma canggih untuk mendeteksi plat nomor dari gambar atau video dan menggunakan *OCR* untuk membaca teks pada plat (Puteri, N. R., 2024).

Kemajuan teknologi, seperti kamera beresolusi tinggi, komputer cepat, pembelajaran mesin, dan kecerdasan buatan, telah meningkatkan keakurantan dan kemampuan sistem deteksi plat nomor untuk menghadapi berbagai kondisi dan situasi (Yuadi, I., Sos, S., 2023).

Adapun fitur utama dari teknologi pendeteksi plat nomor :

a. Akurasi

   Sistem modern memiliki akurasi yang mengesankan, bahkan dalam kondisi yang menantang.

b. Pemrosesan Waktu Nyata

   Banyak sistem menawarkan pemrosesan real-time untuk mengidentifikasi secara cepat.

c. Integrasi

   Sistem ini terintegrasi dengan baik dengan sistem keamanan atau manajemen lainnya.

d. Kustomisasi

   Pengguna dapat menyesuaikan parameter agar sesuai dengan kebutuhan pemilik.

2. Algoritma YOLO (You Only Look Once)

Algoritma YOLO (You Only Look Once) merupakan sebuah algoritma pendeteksian objek yang pertama kali diperkenalkan yakni di tahun 2015 dalam jurnal You Only Look Once: Unified, Real Time Object Deteksion. Algoritma YOLO mengambil pendekatan yang berbeda dengan metode pendeteksian objek lainnya karena mendeteksi dan mengklasifikasikan objek secara bersamaan dalam satu langkah (Alfarizi, D. N., 2023).

Algoritma YOLO juga merupakan asitektur dari deep learning dan merupakan bagian dari machine learning yang terkenal dalam pendeteksian suatu objek atau wajah. YOLO membagi gambar menjadi beberapa segmen n x n menjadi

suatu grid yang dirancang untuk secara utuh melatih gambar dan mengoptimasi pendeteksian objek. YOLO mengoptimasi proses deteksi dengan mengerjakan setiap prosesnya dalam satu neural network. Hal tersebut dicapai dengan cara mengubah pendekatan dalam deteksi objek menjadi satu masalah regresi sehingga langsung dari setiap pixel pada gambar untuk menentukan koordinat bounding box dan pengklasifikasian pada objek. Apabila di suatu grid terdeteksi objek, maka akan diprediksi bounding box yang mengelilingi objek tersebut.

1. Arsitektur dan Cara Kerja Algoritma YOLO (You Only Look Once)

YOLO menggunakan 24 convolutional layers dan 2 fully connected layers. YOLO mendeteksi objek dengan membuat SxS grid cells pada frame atau gambar input dimana setiap grid cell memiliki satu objek saja dan sejumlah bounding box yang berjumlah tetap. Setiap bounding box memiliki satu confidence score dari objek yang terdeteksi di dalam bounding box. Dengan metode pendeteksian ini, setiap bounding box akan memiliki nilai x, y, w dan h yang memprediksikan koordinat dan ukuran bounding box pada objek yang dideteksi.

Setiap grid cell akan terhubung kepada urutan kelas objek secara berurutan untuk memprediksi klasifikasi objek berdasarkan C daftar kelas objek yang harus diklasifikasikan. Rumus matematis pada YOLO untuk membangun convolutional neural network dan memprediksi dimensi tensor yang dimana rumus tersebut :

$$S \times S \times (B * 5 + C)$$

Rumus tersebut memiliki nilai ukuran grid cells pada gambar yang diwakili oleh SxS. B digunakan untuk mewakili jumlah bounding box yang akan dibuat untuk setiap grid cells. Kemudian C adalah jumlah kelas objek yang harus dideteksi

dan diprediksi sesuai klasifikasi objek yang terdeteksi. Untuk memilih bounding box yang paling tepat adalah dengan menggunakan IoU (Intersection over Union) yang bernilai paling tinggi dan dibandingkan dengan ground truth ketika melakukan proses training network di setiap iterasi.

YOLO menggunakan Loss Function yang terdiri dari tiga loss function dimana masing-masing digunakan untuk menghitung confidence loss, localization loss, dan classification loss yang menggunakan squared-error untuk memprediksi kemungkinan kelas pada objek yang terdeteksi. YOLO juga menggunakan non-maximal suppression (NMS) untuk memfilter dan membatasi duplikasi deteksi pada objek.

a. Struktur Masukan (Input)

   Algoritma YOLO mengambil masukan (input) gambar dengan ukuran tetap dan membaginya menjadi kotak sel. Setiap sel bertanggung jawab untuk memprediksi kotak pembatas objek dan kelas objek yang ada di dalamnya (Putro, E. C., 2020).

b. Struktur Konvolusi

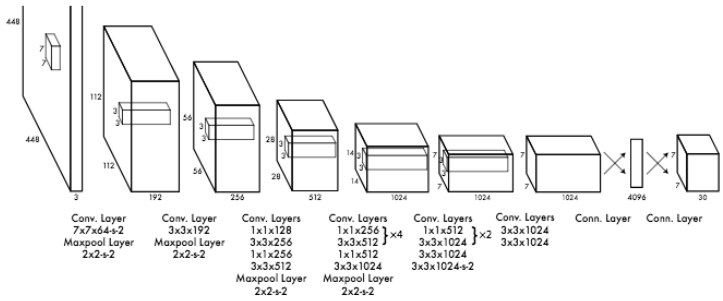
   Untuk mengekstrak fitur dari masukan (input) gambar, algoritma YOLO menggunakan lapisan konvolusi. Lapisan konvolusi ini membantu algoritma dalam memahami konteks dan membedakan objek berdasarkan fitur visual yang ada pada gambar (Khairunisa, N., 2024).

c. Struktur Deteksi

Memprediksi kotak pembatas objek dan kelas objek setelah proses konvolusi dan menghasilkan beberapa kotak pembatas dengan tingkat kepercayaan yang sesuai untuk setiap sel dalam grid (Hutauruk, J. S. W., 2020).

d. Struktur Keluaran (Output)

Daftar kotak pembatas objek dengan label kelas dan tingkat kepercayaan adalah hasil akhir dari algoritma YOLO. Untuk mendapatkan deteksi objek yang tepat, teknik non-maximum suppression digunakan untuk menghilangkan kotak-kotak yang tumpang tindih (Anjani, N., 2024).



Gambar 2.1 Asitektur YOLO

2. Cara Kerja Algoritma YOLO (You Only Look Once)

a. Preprocessing

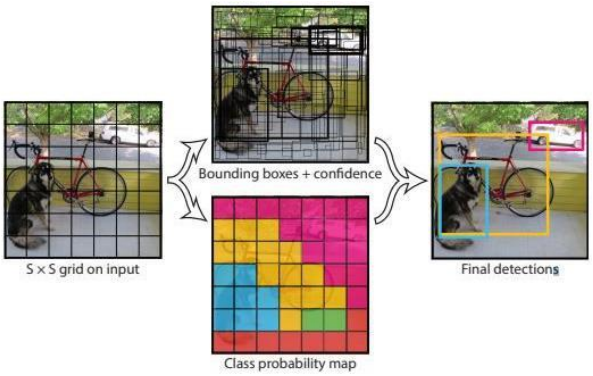
Sebelum memasukkan gambar ke dalam algoritma YOLO, prores preprocessing dilakukan pada tahap ini, ukuran gambar diubah untuk menyesuiakan dengan masukan (input) yang diterima oleh algoritma, dan gambar juga dinormalisasikan untuk memastikan nilai pixel tetap dalam rentang yang diharapkan (Muhamad Zulkarnaen, F, 2024).

b. Pembagian Gambar Menjadi Grid

Setelah tahap preprocessing selesai, gambar akan dibagi menjadi grid yang berbeda dalam ukurannya. Setiap grid akan mendeteksi objek dan menghasilkan beberapa bounding box untuk memperbaiki posisi dan ukuran objek (Saputra, R. D., 2023).

c. Prediksi dan NMS

Pada saat ini, algoritma YOLO akan memprediksi kemungkinan adanya objek dalam setiap kotak batas yang dibuat oleh grid. Untuk mengurangi adanya kotak batas yang tumpang tindih dan mendapatkan hasil deteksi yang lebih akurat, Non-Maximum Suppression (NMS) menghilangkan kotak batas yang memiliki overlap yang tinggi (Virgiawan, I., 2024).



Gambar 2.2 Ilustrasi proses pengolahan data

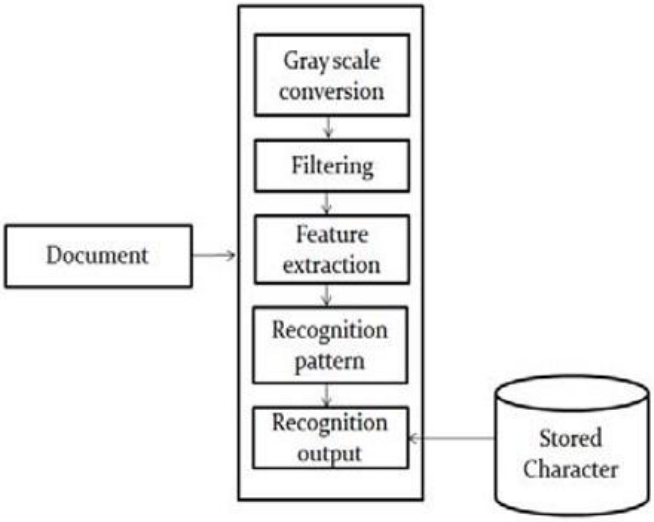
Pada algoritma YOLO, image data pada tahap awal akan dibuat dan di bagi ke dalam SxS grid cells yang digunakan untuk memprediksi objek yang berada di dalam masing-masing grid cells. Setiap grid cells memprediksi confidence score dan class probabilites dengan membuat anchor box pada setiap grid cells hingga objek dapat diketahui lokasi dan kelas objek pada gambar.

3. OCR (Optical Character Recognition)

OCR (Optical Character Recognition) adalah sebuah sistem komputer yang dapat membaca huruf yang berasal dari printer atau mesin ketik maupun tulisan tangan. Aplikasi yang dikenal sebagai OCR yang mengubah gambar karakter, atau gambar karakter menjadi teks dengan menyusaikan pola karakter perbaris dengan pola yang tersimpan dalam database aplikasi. Hasil proses OCR adalah teks yang mirip dengan gambar yang dihasilkan scanner, dan tingkat keakuratan penerjemahan karakter bergantung pada metode yang digunakan dan tingkat kejelasan gambar (Ibrahim, R. M., 2023).

OCR memiliki dua metode; yang pertama adalah Template Marching. Pada dasarnya, Template Marching adalah langkah sederhana. Gambar masukan (input) dengan template tertentu dibandingkan dengan gambar pada basis data Template ditempatkan di pusat bagian gambar yang akan dibandingkan, dan jumlah titik yang paling sesuia dengan template dihitung. Langkah ini diulangi pada gambar masukan (input) keseluruhan yang akan dibandingkan. Nilai kesesuian titik tertinggi antara gambar masukan dan gambar template menunjukkan bahwa gambar template paling sesuia dengan gambar masukan (input) (Pranadipa, R., 2012).

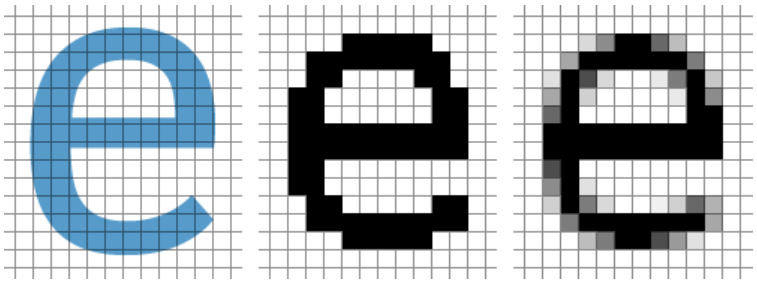
Salah satu cara untuk mengenali suatu objek dengan melihat ciri-ciri khususnya adalah Feature Extraction. Tujuan dari metode ini adalah untuk melakukan perhitungan dan perbandingan yang dapat digunakan untuk mengklasifikasikan ciri-ciri yang dimiliki oleh suatu gambar (Bahri, R. S., 2012).



Gambar 2.3 System Blog Diagram OCR

Grayscale pada gambar bermaksud untuk memastikan intensitas setiap pixel, untuk meningkatkan akurasi data yang diinput. Dalam prosesnya grayscale merubah warna dasar yang sebelumnya merupakan RGB model, menjadi model grayscale menggunakan fungsi berikut :

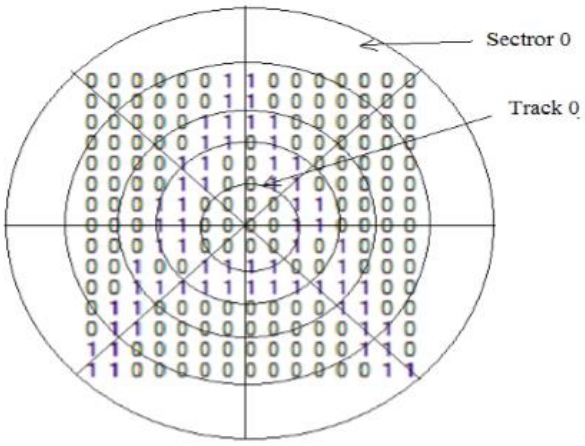
$$Y = 0.2126R + 0.7152G + 0.0722B$$



Gambar 2.4 Hasil Operasi Black-White dan Grayscale

Feature Extraction adalah proses untuk mendapatkan informasi terhadap object ataupun kelompok object untuk memfasilitasi proses klasifikasi. Diantaranya yang dilakukan adalah memotong data gambar per-karakter yang ditentukan, menormalisasi ukuran pixel, menerjemahkan data pixel kedalam binary.

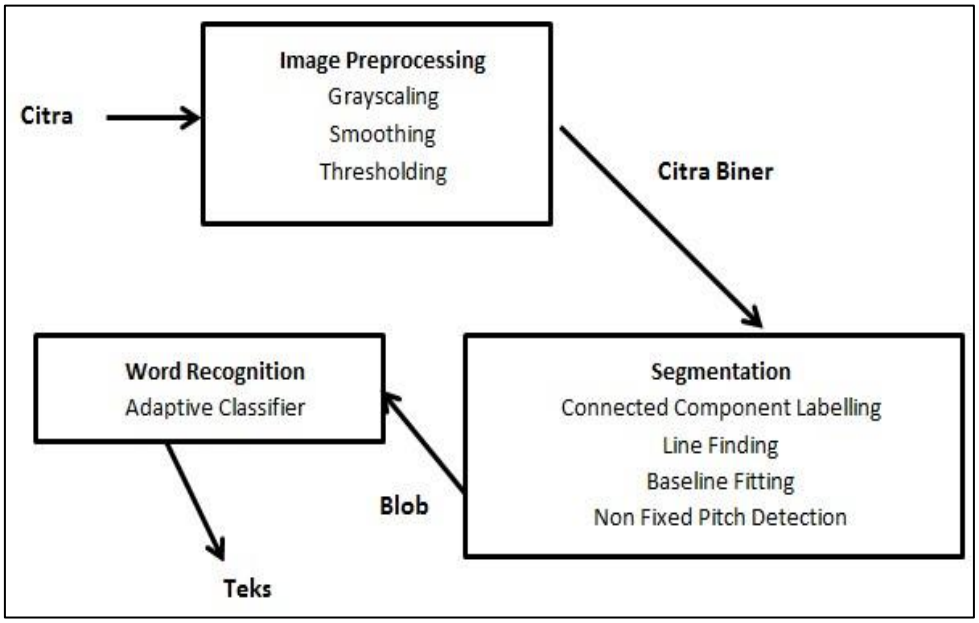
Pattern Recognation dilakukan berdasarkan format data binary yang ada, kemudian membagi binary kedalam 5 track, dimana setiap track dibagi lagi menjadi 8 sector. Keterkaitan matrix track dan sector berguna untuk mendeteksi sekelompok pixel di setiap bagian. Pada masing-masing bagian, jumlah angka 1 dihitung kemudian output recognation dapat dilakukan dengan menggunakan metode klasifikasi tertentu, misalnya ANN atau SVM.



Gambar 2.5 Pembagian Kedalam Track dan Sector

4. *Tesseract* OCR

Tesseract merupakan library Optical Character Recognition (OCR) open source untuk berbagai sistem operasi yang awalnya dikembangkan oleh Hewlet Packard (HP) pada tahun 1985 hingga 1995. Setelah tahun 2006, *Tesseract* dilepas oleh HP untuk digunakan secara bebas. Sejak saat itu, *Tesseract* dikembang secara luas oleh Google dan dirilis di bawah lisensi Apache 2.0 (Ramadhani, A., 2020).



Gambar 2.6 Asitektur *Tesseract OCR Engine*

Berikut adalah penjelasan dari gambar arsitektur diatas :

1. Image Preprocessing

Proses *image preprocessing* terdiri dari *grayscaling*, *smoothing*, dan *thresholding*. Bagian ini menghasilkan citra biner yang menjadi masukan pada tahap pengenalan karakter (Muhtadii, M., 2016).

a. Grayscaling

Grayscaling merupakan tahapan awal dalam image preprocessing yang merupakan proses pengolahan citra dengan mengubah nilai pixel awal citra menjadi sebuah citra keabuan (Nabuasa, Y. N., 2019).

b. Smoothing

Smoothing yaitu proses pemulusan gambar untuk mengurangi noise dari citra (Maximillian, L., 2023).

c. Thresholding

Thresholding yaitu proses untuk memisahkan antara objek dengan background sehingga menghasilkan citra biner dengan mengubah gambar berwarna maupun grayscale menjadi binary image dengan mengubah masing masing pixel dalam kisaran tertentu (Sulistiyo, A., 2019).

2. Segmentation

Ada beberapa tahap dalam segmentation agar karakter pada citra berhasil dikenali, antara lain connected component lebelling, line finding, baseline fiting, dan non fixed pitch detection (Apriyanti, K., 2016).

a. Connected Component Labelling

Connected component labelling adalah proses untuk mendeteksi komponen-komponen karakter yang saling terhubung, melakukan pencarian sepanjang citra kemudian mengidentifikasi pixel latar depan dan menandainya sebagai Outline. Outline dari setiap karakter akan dikumpulkan menjadi blob (Prasetyo, D. (2018).

b. Line finding

Line finding adalah proses untuk mencari baris teks yang dirancang supaya halaman miring dapat dikenali tanpa harus melakukan deskew (proses untuk mengubah halaman yang miring menjadi tegak lurus) sehingga tidak menurunkan kualitas gambar (Abdurrahman, M. W., 2012).

c. Baseline Fitting

Proses menentukan baris teks dan garis pangkal (baseline) dicocokan secara lebih tepat menggunakan quadratic spline. Quadratic spline merupakan metode untuk menghasilkan titik pada sebuah rentang data yang telah diketahui sebelumnya (Fajar, M., 2015).

d. Non fixed pitch detection

Bila teks yang digunakan tidak memiliki lebar garis tepi yang tetap non fixed pitch detection akan melakukan dengan cara mengukur batasan kesenjangan antara base line dan mean line. Baris teks dipisah ke dalam kata-kata dan dibedakan menurut jenis spasi karakter. *Tesseract* menguji garis teks (teks line) untuk menentukan apakah teks merupakan *fixed pitch*. Bila ditemukan *fixed pitch text* maka *Tesseract* memotong kata-kata menjadi karakter-karakter (Widiani, P. E., 2014).

3. Word Recognition

Pengenalan selanjutnya dilakukan melalui dua proses yang disebut passtwo. Pass pertama dilakukan untuk mengenali setiap kata secara bergilir. Keberhasilan pass pertama dipengaruhi oleh keberadaan kata dalam kamus dan tidak mempunyai makna ganda (ambigu), sehingga kata tersebut diteruskan ke

Adaptive classifier sebagai data pelatihan. Adaptive classifier kemudian dapat mengenali teks lebih akurat pada halaman selanjutnya. Pass kedua berfungsi melakukan pengenalan kembali kata-kata yang sukar dikenali atau terlewatkan pada pass pertama (Suarti, I., 2024).

a. Pemisahan Karakter Terhubung (Chopping Joined Characters)

   Apabila hasil dari pengenalan kata tidak memuaskan, library *Tesseract* berusaha untuk memperbaiki hasil dengan memisahkan blob dengan keyakinan terburuk dari pengklasifikasian karakter. Kandidat untuk titik-titik pemisahan ditemukan dari simpul cekung dari pendekatan poligonal outline dan mungkin saja terdapat titik cekung berlawanan lainnya atau segmen garis. Ini akan menghabiskan sampai 3 pasang titik pemotong untuk memisahkan karakter yang terhubung dari set ASCII (Dermawan, M. S., 2019).

b. Asosiasi Karakter Patah

   Ketika potongan yang berpotensi tidak lagi ada, ketika kata tersebut masih belum cukup baik, hal ini diberikan kepada assosiator. Associator membuat perencanaan A* (Best First Search) dari segmentasi grafik yang mungkin kombinasi dari blob yang dipotong secara maksimal ke dalam kandidat karakter. Ketika A* segmentation digunakan untuk diimplementasikan pertama kali pada tahun 1989, akurasi *Tesseract* terhadap karakter yang rusak cukup baik yang menjadikan *Tesseract* mesin komersial pada saat itu (Fajar, M., 2015).

5. Raspberry Pi

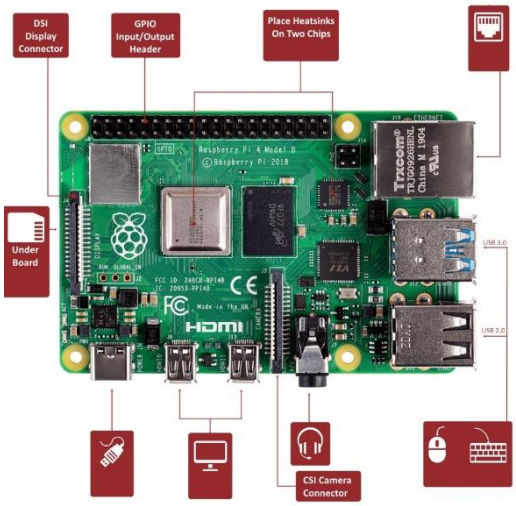
Raspberry Pi adalah suatu jenis komputer mini yang terdiri dari sebuah papan sirkuit yang tidak lebih besar dari kartu kredit. Dirancang untuk menjadi perangkat yang terjangkau dan dapat diakses oleh banyak kalangan, Raspberry Pi memiliki kemampuan komputasi yang memadai untuk menjalankan sistem operasi seperti Linux dan menjalankan berbagai jenis perangkat lunak. Meskipun ukurannya yang kecil, Raspberry Pi dilengkapi dengan berbagai port input/output, termasuk USB, HDMI, dan GPIO (*General Purpose Input/Output*), yang memungkinkan pengguna untuk terlibat dalam proyek-proyek kreatif dan eksperimen elektronik. Raspberry Pi tidak hanya sekadar perangkat keras, tetapi juga mencakup komunitas pengguna yang besar yang berbagi pengetahuan, proyek-proyek, dan dukungan. Ini membuat Raspberry Pi lebih dari sekadar komputer mini, tetapi juga menjadi alat pembelajaran yang ideal bagi mereka yang tertarik memahami dunia pemrograman dan teknologi (Hidayatullah, M. A., 2019).

Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris. Ide dibalik Raspberry Pi diawali dari keinginan untuk mencetak pemrogram generasi baru. Seperti disebutkan dalam situs resmi Raspberry Pi Foundation, waktu itu Eben Upton, Rob Mullins, Jack Lang, dan Alam Mycroft, dari Laboratorium Komputer Universitas Cambridge memiliki kekhwatiran melihat kian turunnya keahlian dan jumlah siswa yang hendak belajar ilmu komputer. Meraka lantas mendirikan yayasan Raspberry Pi bersama dengan Pete Lomas dan David Braben pada tahun 2009. Tiga tahun kemudian, Raspberry

Pi Model B memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit. Pada bulan Februari 2016, Raspberry Pi Foundation mengumumkan bahwa mereka telah menjual 8 juta perangkat Raspi, sehingga menjadikannya sebagai perangkat paling laris di Inggris pada waktu itu (Solekhan, S., 2020).

6. Raspberry pi 4

Raspberry Pi 4 adalah generasi ke-4 dari seri utama komputer papan tunggal Raspberry Pi. Dikembangkan oleh Raspberry Pi Trading dan rilis pada 24 Juni 2019, Pi 4 hadir dengan banyak peningkatan dari pendahulunya; Soc ditingkatkan ke Broadcom BCM2711, dua dari empat port USB Raspberry Pi ditingkatkan ke USB 3.0, dan opsi ditambahkan untuk kapasitas RAM yang lebih besar dari standar 1 GB untuk seri Raspberry Pi 3 sebelumnya. Pi 4 juga mengakhiri tren MSRP maksimum $35 yang dianut Raspberry Pi sebelumnya, karena kapasitas RAM yang lebih besar menambah biaya ekstra untuk papan; namun, model dasar 1 GB masil di jual seharga $35. Pada 28 September 2023, Raspberry Pi 5 diumumkan sebagai penerus Raspberry Pi 4 (Anggraini, R., 2019).



Gambar 2.7 Informasi Raspberry Pi 4

Spesifikasi fitur pada raspberry pi 4 :

1. **Prosesor**

   Raspberry Pi 4 menggunakan prosesor Quad-core ARM Cortex-A72 yang memiliki clock 1,5 GHz atau 1,8 GHz tergantung pada revisi SoC BCM2711 di papan tersebut. Stepping B0 yang lebih lama awalnya digunakan untuk Pi 4, tetapi ini digantikan dengan chip stepped C0 yang lebih baru mulai pertengahan tahun 2021. Laju clock yang lebih tinggi dari stepping C0 disebabkan oleh termal yang sedikit lebih baik. Prosesor ini juga berisi cache data L1 sebesar 32 KB dan cache intruksi L1 sebesar 48 KB, bersama dengan cache L2 bersama sebesar 1 MB (Sandy, A. K., 2021).

2. **GPU**

   Raspberry Pi 4 menggunakan GPU Broadcom BCM2711 SoC memiliki GPU yang ditingkatkan dibanding dengan iterasi Raspberry Pi sebelumnya, mulai dari VideoCore IV yang memiliki clock 400 MHz hingga VideoCore VI yang memiliki clock 500 MHz. Selain clock rate yang lebih cepat, VideoCore VI

juga memiliki pengelola memori sendiri, yang memungkinkannya mengakses lebih banyak memori dari pada pendahulunya. *VideoCore* VI sesuia dengan *Open GL ES* 3.1 dan *Vulkan* 1.2 (Yuliani, F., 2022).

3. RAM

Raspberry Pi 4 menggantikan 1 GB *LPDDR2* dengan pilihan 1 GB, 2 GB, 4GB, atau 8 GB *LPDDR4* 3200 MHz. Pi 4 RAM 8 GB dirilis setahun setelah model lainnya (Muthoá, M. F., 2017).

4. Input/Output dan Konektivitas

Banyak perbaikan diperlukan pada fungsionalitas *I/O* dengan *Raspberry Pi 4*. Total *bandwith* USB yang ditingkatkan, dari empat port USB 2.0 menjadi dua port USB 3.0 dan dua port USB 2.0. Kontroler Ethernet khusus pada BCM2711 Soc memungkinkan penambahan Gigabit Ethernet. Port HDMI berukuran penuh tunggal pada raspberry Pi 3 juga telah diganti dengan konektor mikro-HDMI ganda. Bluetooth ditingkatkan dari 4.2 pada raspberry Pi 3, menjadi 5.0 pada Raspberry Pi 4. Konektor catu daya juga berubah dari *Micro-USB* ke *USB-C*. Port *Ethernet* menampilkan *Power over Ethernet IEEE 802.3af* (802.3at Tipe 1) seperti *Raspberry Pi 3 Model B+* (Hangkawidjaja, I. A. D., 2021).

7. Kamera Webcam

Kamera Webcam sebagai piranti pengambilan citra secara real time. Kamera ini menggunakan koneksi USB sebagai komunikasi data citra yang ditangkap melalui optik dan ditransfer kedalam piranti Raspberry Pi 4 menggunakan koneksi USB. Adapun resolusi yang digunakan ialah 1080 Pixel

sehingga citra yang dihasilkan lebih maksimal dan deteksi citra pada plat nomor depat terdeteksi dikarenakan memiliki resolusi yang agak tinggi (Alandani, R., 2017).

8. Motor servo

Motor servo adalah sebuah perangkat sebagai aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. Potensiometer untuk memaksimalkan putaran pada motor servo, untuk arah putaran dan sudut dari sumbu motor servo dapat diatur menggunakan sinyal PWM (Pulse Width Modulation) (Amin, M., & Novelan, 2020).

Terdapat dua tipe motor servo, yaitu motor servo AC dan DC. Jika dibedakan berdasarkan rotasinya, terdapat dua jenis motor servo, yaitu 180º dan 360º. Dalam penelitian ini, motor servo yang digunakan adalah motor servo standar (servo 180º) (Fonna, Husaini, & Indrawati, (2020).

Motor servo standar (servo 180º) merupakan jenis yang paling lazim digunakan, dengan poros output yang hanya dapat berputar 90º ke kanan dan 90º ke kiri. Artinya, total putarannya cuma 180º. Karena motor ini hanya bergerak ke kanan, kembali ke tengah, dan ke kiri saja, tidak dapat mencapai satu putaran penuh (Deddy Pradana, 2021).



Gambar 2.8 Motor Servo

Tabel 2.1 Spesifikasi Motor Servo

Spesifikasi	Keterangan
Berat	9g
Rentang rotasi	180°
Tegangan oprasi	4.8V-6V
Suhu oprasi	-30°C to + 60°C
Kecepatan	0.1 sec/60° (4.8v) 0.08sec/60° (6v)
Rotasi maksimum	1.6 kg/cm(4.8v),2.2 kg/cm(6v)
Sistem kontrol	Pwm (pulse Width Modulation)

B. Kajian Hasil Penelitian Terdahulu

Adapun penelitian terdahulu yang menjadi salah satu acuan penulis dalam melakukan penelitian ini sehingga penulis dapat memperbanyak teori yang digunakan dalam mengkaji penelitian yang dilakukan. Penulis sengaja mengangkat beberapa penelitian sebelumnya untuk dijadikan sebagai referensi dalam memperkaya bahan kajian pada penelitian ini. Berikut ini yang merupakan penelitian terdahulu berupa jurnal yang terkait dengan penelitian yang dilakukan oleh penulis dalam penelitian pertama yang dilakukan oleh (B. Patel, H. Bhavsar, 2021) dengan judul "Deteksi dan Pengenalan Plat Nomor Kendaraan Menggunakan Metode *YOLO-V3* dan *OCR*" penelitian ini menggunakan model *YOLO-V3* untuk

mendeteksi plat nomor kendaraan, diikuti dengan pengenalan teks menggunakan OCR untuk membaca karakter pada plat. Studi ini menunjukkan hasil yang efektif untuk pengenalan plat dalam kondisi gambar yang bervariasi.

Penelitian kedua yang dilakukan oleh (S. Kumar, A. Singh, 2023) dengan judul "Analisis Perbandingan *EasyOCR* dan *Tesseract OCR* untuk Pengenalan Plat Nomor Otomatis Menggunakan Algoritma *Deep Learning*" penelitian ini membandingkan kinerja *Tesseract OCR* dengan *EasyOCR* pada sistem pengenalan plat nomor yang menggunakan *YOLOv5* untuk deteksi plat. Hasil menunjukkan keunggulan *EasyOCR* dari sisi kecepatan, namun *Tesseract* tetap relevan untuk tugas pengenalan teks yang kompleks.

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Metode yang digunakan dalam penelitian ini adalah berupa penelitian eksperimental atau pengembangan sistem. Sehingga penelitian ini melibatkan pengujian model *YOLO* untuk mendeteksi objek (plat nomor) dan Tesseract *OCR* untuk pengenalan teks pada data yang telah disiapkan. Eksperimen dilakukan unutk mengukur efektivitas metode tersebut dalam berbagai kondisi pencahayaan, sudut pandang kamera, atau tipe plat nomor. Sedangkan pengembangan sistem menggabungkan pendekatan deep learning (*YOLO*) dengan teknologi *OCR* untuk membangun sistem terintegrasi yang mampu mendeteksi dan mengenali teks pada plat nomor kendaraan. Biasanya, penelitian ini mencakup implementasi, dan sistem evaluasi.

B. Lokasi dan Waktu

1. Lokasi

Penelitian ini dilakukan di laboratorium Teknik Elektro Universitas Muhammadiyah Parepare.

2. Waktu

Waktu penelitian dilaksanakan sejak dikeluarkannya izin penelitian dengan uraian yaitu studi pustaka, perancangan alat, pengadaan alat dan bahan, pemasangan dan instalasi alat, pengujian dan dilanjutkan dengan pembuatan laporan akhir yang dapat dilihat pada tabel dibawah ini :

Tabel 3.1 Jadwal pelaksanaan penelitian

No.	Uraian	Bulan					
		1	2	3	4	5	6
1	Studi Literatur	■					
2	Perancangan Perangkat Keras		■				
3	Perancangan Perangkat Lunak		■				
4	Realisasi Perangkat Keras dan Perangkat Lunak			■	■		
5	Pengujian Tahap Awal (Komponen)				■		
6	Pengujian Tahap Kedua (Blok Diagram)					■	
7	Pengujian Sistem Secara Keseluruhan					■	
8	Analisis dan Evaluasi					■	■
9	Penyusunan Laporan						■

C. Alat dan Bahan

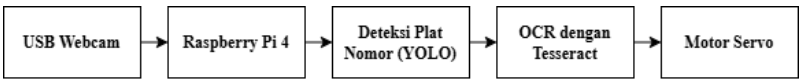
Adapun alat dan bahan yang digunakan dalam penelitian ini adalah :

Tabel 3.2 Alat dan Bahan

No.	Uraian	Jumlah
1.	Raspberry Pi 4	1
2.	Kamera Webcam	1
3.	Motor Servo	1

4. Rancangan Penelitian

Pada bagian ini akan di bahas suatu pernyataan rangkaian alat yang akan dirancang. Berikut blok diagram perancangan pada penelitian ini:



Gambar 3.1 Blok diagram rancangan penelitian

Pada gambar di atas merupakan blok diagram Sistem Pendeteksi Nomor Plat Kendaraan Menggunakan YOLO dan Tesseract OCR. Adapun komponen penyusun yang terdiri dari :

1. Input atau masukan yang berupa kamera webcam untuk mengambil gambar atau video secara realtime sehingga dapat mendeteksi nomor plat kendaraan
2. Preprocessing image atau praproses gambar yang dimana dapat melakukan transmisi awal pada gambar agar kualitas gambar lebih baik dan memudahkan deteksi nomor plat kendaraan. Adapun tahapan praproses yang pertama mengkonversikan gambar menjadi grayscale atau mengubah gambar berwarna menjadi hitam-putih untuk mengurangi kompleksitas.

Kedua menghilang kebisingin atau menghilangkan noise pada gambar agar lebih jernih. Ketiga resize image atau menyesuaikan ukuran gambar agar sesuai dengan input model deteksi. Keempat thresholding atau memperjelas kontras antara plat nomor dan latar belakang.

3. Deteksi plat nomor menggunakan algoritma YOLO (You Only Look Once) untuk mendeteksi lokasi plat nomor pada gambar. YOLO adalah model deteksi objek yang cepat dan akurat. Dalam proses ini YOLO akan memberikan pembatas kotak yang menandai lokasi area plat nomor. Yang menghasilkan koordinat lokasi plat nomor pada gambar.
4. Ekstrak gambar plat nomor yang dimana bertujuan untuk memotong area plat nomor dari gambar berdasarkan pembatas kotak yang diberikan oleh YOLO. Dan dari eksraksi ini berupa gambar plat nomor saja, yang betujuan untuk menyiapkan gambar plat nomor untuk tahap selanjutnya, yaitu OCR.
5. OCR dengan Tesseract yang berfungsi untuk menggunakan Tesseract OCR untuk mengenali karakter atau teks pada gambar plat nomor. OCR (Optical Character Recognition) adalah teknologi untuk mengubah gambar teks menjadi teks digital. Proses ini bertujuan untuk mengambil gambar plat nomor yang sudah dipotong diproses oleh Tesseract untuk membaca dan mengenali teks karakter. Yang dimana keluarannya berupa teks nomor plat kendaraan dalam format digital (misalnya, “B 1234 XYZ”).
6. Pasca-pemprosesan dan penyimpanan yang berfungsi untuk memproses hasil teks dari OCR untuk memastikan akurasi dan menyimpan hasil dalam sistem, adapun tahapan pertama melakukan pemeriksaan format atau

mengedit teks sesuai dengan format plat nomor yang diinginkan. Tahapan kedua error correction atau melakukan koreksi jika ada karekter yang tidak sesuai. Ketiga penyimpanan data atau menyimpan hasil teks ke dalam database atau sistem penyimpanan. Sehingga keluaran yang diperoleh berupa data nomor plat kendaraan yang valid.

7. Keluaran sistem yang berfungsi menampilkan hasil akhir berupa nomor plat kendaraan yang berhasil dikenal. Hasilnya dapat ditampilkan pada antarmuka pengguna (UI) atau disimpan untuk keperluan lain.

E. Teknik Pengumpulan Data

1. Pengujian Blok Diagram

Pengujian blok diagram bertujuan untuk dilakukan untuk memastikan setiap tahapan dalam sistem berjalan sesuai fungsinya. Berikut adalah penjelasan pengujian yang dilakukan berdasarkan blok diagram :

a. Uji kualitas input gambar dari berbagai sumber berupa kamera atau file dan variasi kondisi cahaya, sudut pandang, dan resolusi gambar.
b. Uji efektivitas preprocessing seperti grayscale, noise removal, dan resizing. Sehingga dapat dipastikan gambar siap untuk tahap deteksi plat nomor.
c. Evaluasi performa YOLO dalam mendeteksi plat nomor. Menguji akurasi, presisi, dan waktu eksekusi model YOLO.
d. Uji kemampuan sistem untuk memotong (crop) area plat nomor dengan akurat.

e. Evaluasi performa Tesseract OCR dalam mengenali teks pada gambar plat nomor. Sehingga dapat di uji keakurasian karakter yang dikenali.
f. Validasi format teks nomor plat. Sehingga dapat menguji penyimpanan hasil deteksi ke database.
g. Uji tampilan hasil pada sistem atau antarmuka pengguna. Mengevaluasi waktu respons sistem.

2. Pengujian Sistem Secara Keseluruhan

Tahapan ini dilakukan untuk mengukur kinerja dari seluruh tahapan proses dalam sistem secara integrasi. Memvalidasi bahwa sistem dapat menangani data dan alur kerja secara kompleks dengan benar, sehingga pengujian secara keseluruhan memastikan integrasi antara *YOLO* dan *Tesseract* berjalan dengan baik. Sistem diuji untuk kecepatan, akurasi, dan kestabilan dalam berbagai kondisi. Sehingga dapat menghasilkan pengujian akhir yang menunjukkan kinerja sistem memenuhi kriteria atau memerlukan optimasi lebih lanjut.

BAB IV

HASIL DAN PEMBAHASAN

1. Perancangan Perangkat Keras

Pada perancangan perangkat keras menggunakan *Raspberry Pi 4* sebagai komputer edge, sebuah webcam USB sebagai sensor visual, dan servo *MG996R* sebagai penggerak mekanis pintu portal adalah pilihan yang ekonomis dan praktis untuk prototipe sistem pendeteksi nomor plat kendaraan. Kombinasi ini memungkinkan pipelin: *capture frame* → deteksi *YOLO* → *crop* dan *preprocess* → *Tesseract OCR* → verifikasi → kendali servo. Namun setiap komponen punya keterbatasan yang mempengaruhi akurasi, latency, dan keandalan, sehingga masing-masing perangkat memiliki peran dan karakteristik yang mempengaruhi kinerja keseluruhan sistem.

1. Raspberry Pi 4

Raspberry Pi 4 digunakan sebagai pusat kendali utama dalam sistem, berfungsi untuk menjalankan model *YOLO* dan *Tesseract OCR* secara langsung. Dengan prosesor *quad-core Cortex-A72 1.5 GHz* dan RAM hingga 4 GB, perangkat ini mampu memproses data citra secara real-time tanpa harus terhubung ke komputer eksternal. Keunggulan *Raspberry Pi 4* adalah ukurannya yang rinkas dan konsumsi daya yang rendah, sehingga cocok ditempatkan di dekat pintu portal tanpa memerlukan ruang besar. Selain itu, *Raspberry Pi 4* dilengkapi dengan port USB 3.0 untuk koneksi kamera berkecepatan tinggi, serta konektivitas *LAN* dan *Wi-Fi* yang memudahkan pembaruan daftar nomor plat kendaraan yang terverifikasi.

Penggunaan Raspberry Pi 4 juga meminimalkan biaya karena tidak memerlukan komputer PC konvensional, namun tetap dapat menangani proses deteksi dan verifikasi secara simultan.

2. Kamera Webcam

Kamera Webcam berfungsi sebagai sensor utama untuk menangkap citra plat nomor kendaraan yang mendekati pintu portal. Kamera ini dipasang pada posisi strategis yang sejajar dengan tinggi rata-rata plat nomor kendaraan untuk meminimalkan distorsi. Kualitas kamera yang digunakan harus mendukung resolusi minimal 640x480 piksel pada 30 FPS untuk memastikan gambar yang diambil cukup jelas bagi YOLO dalam mendendeteksi objek plat nomor kendaraan. Dalam implementasi, kamera dihubungkan langsung ke Raspberry Pi 4 melalui port USB 3.0 untuk mengurangi latency pengiriman data. Hasil pengujian menunjukkan bahwa kamera dapat bekerja optimal pada pencahayaan siang hari.

3. Servo MG996R

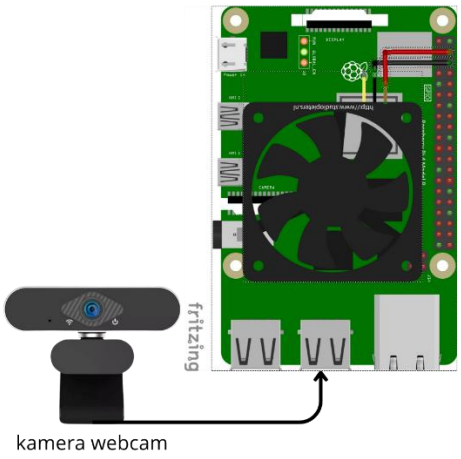
Servo MG996R digunakan sebagai aktuator yang menggerakkan mekanisme pintu portal setelah sistem mendeteksi dan memverifikasi nomor plat kendaraan. Servo ini dipilih karena memiliki torsi tinggi (hingga 11kg cm) sehingga mampu menggerakkan palang portal dengan beban moderat. Keunggulan *MG996R* adalah kemampuannya memberikan pergerakan presisi pada sudut tertentu, sehingga pintu portal dapat membuka dan menutup dengan halus. Servo dikendalikan langsung oleh *GPIO Raspberry Pi 4* menggunakan sinyal *PWM*, yang memungkinkan penyesuaian sudut secara fleksibel sesuai kebutuhan mekanisme portal. Pengujian

lapangan menunjukkan bahwa *MG996R* dapat bekerja stabil dalam siklus buka-tutup berulang.

Adapun dari hasil analisis integrasi antara perangkat keras:

a. Integrasi *Raspberry Pi 4* dan Kamera Webcam

Raspberry Pi 4 terhubung dengan kamera webcam melalui *port USB*, memanfaatkan antarmuka USB 3.0 untuk mempercepat transfer data video. Integrasi ini memungkinkan pengambilan gambar dengan *frame rate* yang stabil sehingga deteksi dapat dilakukan secara real-time. Penggunaan pustaka *driver Linux* pada *Raspberry Pi* memastikan kamera dapat langsung digunakan tanpa konfigurasi tambahan yang kompleks. Hasil uji menunjukkan latensi yang rendah, namun penggunaan resolusi tinggi (lebih dari 1280x720) dapat memperlambat pemprosesan *YOLO* jika tidak dioptimalkan.

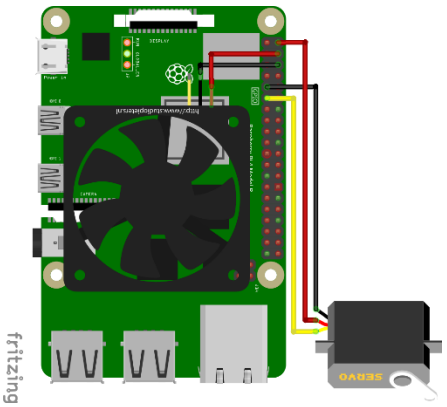


Gambar 4.1 Rangkaian USB Kamera Webcam di *Raspberry Pi 4*

b. Integrasi *Raspberry Pi 4* dan Servo *MG996R*

Servo *MG996R* dikendalikan menggunakan sinyal *PWM* dari *GPIO Rasberry Pi 4*. Agar sistem stabil, daya servo yang terhubung di pin 2 dengan sumber eksternal 5-6V dan ground servo yang terhubung di pin 9 untuk menghindari

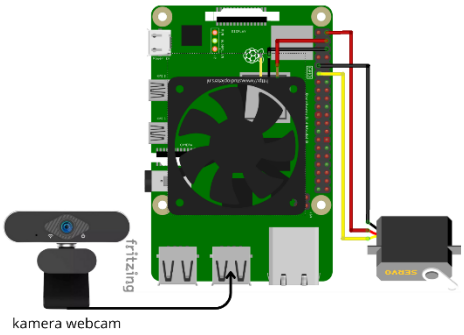
floating ground. Pemprograman kontrol servo menggunakan pustaka *RPi.GPIO* atau *pigpio* pada pin 11 memungkinkan pengaturan sudut secara presisi. Hasil pengujian menunjukkan servo dapat membuka dan menutup portal dengan waktu respons rata-rata dibawah 1 detik setelah verifikasi plat nomor.



Gambar 4.2 Rangkaian Servo *MG996R* di *Raspberry Pi 4*

c. Sinkronisasi Kamera, Pemrosesan, dan Servo

Sinkronisasi dilakukan melalui perangkat lunak di *Raspberry Pi 4*. Kamera menangkap gambar, *YOLO* memproses untuk mendeteksi plat, dan Tesseract *OCR* mengenali teks. Jika teks sesuai dengan daftar plat tervefirikasi, *Raspberry Pi* langsung mengirimkan sinyal *PWM* ke servo untuk membuka portal. Pengujian menunjukkan waktu total dari deteksi hingga portal terbuka berada pada kisaran 1,5-2 detik, tergantung kondisi pencahayaan dan kecepatan deteksi *YOLO*. Integrasi ini bekerja dengan baik selama *Raspberry Pi* tidak kelebihan beban pemrosesan, sehingga optimalisasi kode dan threading diperlukan untuk mempertahankan performa real-time.



Gambar 4.3 Rangkaian Keseluruhan



Gambar 4.4 Realisasi Rancangan Keseluruhan

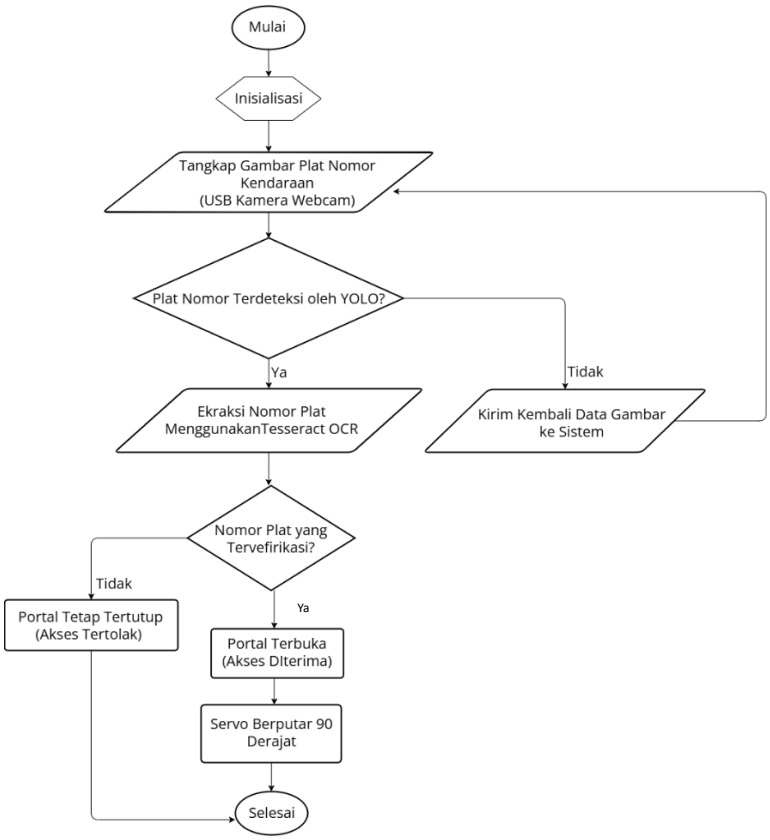
2. Perancangan Perangkat Lunak

Pada tahap perancangan perangkat lunak sistem pendeteksi plat nomor ini di kembangkan dan dijalankan langsung pada *Raspberry Pi* 4 dengan sistem operasi *Raspberry Pi OS*, menggunakan bahasa *Python* sebagai basis pemrograman. Model *YOLOv8* digunakan untuk proses deteksi objek plat nomor secara real-time melalui kamera USB/webcam yang dikonfigurasi pada resolusi 640x480 *piksel* dengan dengan frame rate 30 FPS untuk menjaga kestabilan kinerja *Raspberry Pi*. Setelah area plat nomor terdeteksi, citra tersebut diproses melalui tahap preprocessing seperti konversi *grayscale*, *adaptive thrsholding*, dan *morphological filtering* untuk mengoptimalkan pembacaan karakter oleh *Tesseract OCR*. Hasil pembacaan *OCR* kemudian diverifikasi dengan basis data yang telah diinput di *Raspberry Pi*, sehingga hanya nomor plat yang telah diverifikasi yang dapat memicu perintah

pembukaan pintu portal melalui servo MG996R. Pengujian menunjukkan *YOLOv8* pada *Raspberry Pi* 4 mampu mencapai akurasi *mAP50* di atas 95% dengan rata-rata waktu respon 1,5-2 detik dari proses deteksi hingga servo bergerak membuka portal. Pada bagian ini disajikan *flowchart* dan uraian program yang menunjukkan alur kerja sistem secara menyeluruh.

1. ***Flowchart***

Flowchart adalah grafik yang digunakan untuk menggambarkan alur kerja atau proses dalam bentuk simbol-simbol grafis untuk menunjukkan langkah-langkah aliran kontrol yang berperan dalam proses. *Flowchart* memudahkan penerimaan pengertian suatu sistem dalam cara yang lebih visual dan terstruktur. Diagram alir sistem kerja ditunjukkan pada gambar 4.4.



Gambar 4.5 *Flowchart*

2. Uraian Program

Adapun proses implementasi sistem pendeteksi nomor plat kendaraan berbasis *Raspberry Pi* yang terintegrasi dengan mekanisme pintul portal otomatis. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan bantuan pustaka pendukung seperti *OpenCV*, *YOLOv8*, dan *Tesseract OCR*. Sketch program yang digunakan untuk menjalankan fungsi dari sistem yang dibuat sesuia dengan *Flowchart*.

Program ini dirancang untuk melakukan akuisisi citra secara *real-time* dari kamera, mendeteksi lokasi plat nomor kendaraan menggunakan algoritma objek *detection YOLOv8*, membaca karakter plat menggunakan *Tesseract OCR*, dan mengontrol motor servo yang menggerakkan pintu portal apabila nomor plat sesuai dengan daftar terverifikasi.

a. Import Library

Kode awal program memuat pustaka yang diperlukan:

```
import cv2
import pytesseract
from ultralytics import YOLO
import RPi.GPIO as GPIO
from time import sleep
import re
import os
from datetime import datetime
import numpy as np
```

Pada bagian awal program, terdapat proses *import library* yang digunakan sebagai fondasi untuk menjalankan keseluruhan sistem diantaranya:

- *cv2* (*OpenCV*) digunakan untuk mengolah citra dari kamera secara real-time, seperti membaca frame, memotong (*crop*) gambar, mengubah format warna, hingga menampilkan hasil deteksi.

- *pytesseract* yang dimana sebagia Modul *Python* untuk mengakses *Tessract OCR Engine*, yang digunakan untuk mengenali teks pada citra, khususnya teks pada plat nomor kendaraan.
- *YOLO* (*You Only Look Once*) yang dimana sebagai *Library* dari *ultralytics* yang membuat algoritma objek detection modern dan efisien, digunakan di sini untuk mendeteksi area plat nomor kendaraan dalam frame video.
- *RPi. GPIO* yang dimana sebagai Modul untuk mengontrol *General Purpose Input/Output* pada *Raspberry Pi*, digunakan untuk mengendalikan servo yang membuka dan menutup pintu portal.
- *time.sleep* yang dimana memberikan jeda waktu tertentu di antara perintah, misalnya untuk memberi waktu servo bergerak atau pintu tetap terbuka sebelum menutup.
- *re* yang dimana berfungsi sebagai modul *regular expression* untuk memproses dan membersihkan hasil pembacaan teks *OCR* dari karakter yang tidak diinginkan.
- *os* yang dimana digunakan untuk membuat folder dan menyimpan file hasil deteksi.
- *datetime* yang dimana memberikan informasi tanggal dan waktu untuk menamai file hasil deteksi secara unik.
- *numpy* yang dimana digunakan untuk perhitungan numerik, termasuk pengaturan step gerakan servo agar bergerak secara halus.

b. Konfigurasi Tesseract OCR

```
pytesseract.pytesseract.tesseract_cmd = '/usr/bin/tesseract'
```

Pada bagian ini mengatur lokasi instalasi *Tesseract* di *Raspberry Pi*. Pada sistem operasi *Linux* (termasuk *Raspberry Pi OS*), *Tesseract* biasanya terletak di */usr/bin/tesseract*. Pengaturan ini penting agar *pytesseract* dapat memanggil *engine Tesseract* secara langsung untuk proses ekstraksi teks.

c. Konfigurasi Servo Motor

```
servo_pin = 11
GPIO.setmode(GPIO.BOARD)
GPIO.setup(servo_pin, GPIO.OUT)
pwm = GPIO.PWM(servo_pin, 50)
pwm.start(0)
CLOSED = 2.5
OPEN = 7.5
```

Pada bagian ini dilakukan inisialisasi motor servo yang mengontrol pintu portal. Pin fisik nomor 11 digunakan sebagai sinyal kontrol servo, dengan mode penomoran pin *GPIO.BOARD*. Servo dikendalikan menggunakan metode *PWM* (*Pulse Width Modulation*) pada *frekuenzi 50 Hz*, yang merupakan standar pengoperasian servo. Nilai CLOSED (2.5) dan OPEN (7.5) mempresentasikan *duty cycle* yang sesuai dengan sudut posisi tertutup dan terbuka dari pintu portal.

d. Pembuatan Folder Hasil Deteksi

```
save_folder = "deteksi_plat"
os.makedirs(save_folder, exist_ok=True)
```

Kode ini memastikan bahwa sistem memiliki direktori khusus untuk menyimpan hasil deteksi plat nomor. Jika folder sudah ada, parameter *exist_ok=True* mencegah program mengalami kesalahan.

e. Daftar Plat Nomor Terverifikasi

```
verified_plates = ["DP3772DS"]
```

Daftar ini berisi nomor plat yang memiliki izin untuk membuka pintu portal. Pengecekan dilakukan setelah proses *OCR* selesai, apabila hasil pembacaan sesui dengan daftar ini, maka pintu akan terbuka secara otomatis.

f. Membuat Model *YOLO*

```
model = YOLO("best.pt")
```

Bagian ini memanggil model *YOLOv8* yang telah dilatih (*custom trained*) menggunakan dataset plat nomor kendaraan. Model *best.pt* dipilih berdasarkan hasil evaluasi pelatihan dengan performa terbaik.

g. Fungsi Gerakan Servo Halus

```
def move_servo(target_duty):
 current_duty = CLOSED if target_duty == OPEN else OPEN
 step = 0.1 if target_duty > current_duty else -0.1
 for duty in np.arange(current_duty, target_duty, step):
 pwm.ChangeDutyCycle(duty)
 sleep(0.05)
```

Fungsi *move_servo* menggerakkan servo secara bertahap untuk menghasilkan pergerakan yang lebih halus (*smooth motion*). Alih-alih berpindah secara tiba-tiba dari posisi tertutup ke terbuka, servo bergerak sedikit demi sedikit dengan langkah sebesar 0.1 setiap 0.05 detik.

h. Fungsi Membuka dan Menutup Pintu

```
def open_gate():
 print("Pintu TERBUKA")
 move_servo(OPEN)
 sleep(5)
 print("Pintu MENUTUP")
 move_servo(CLOSED)
```

Fungsi ini memanggil move_servo untuk membuka pintu, menunggu selama 5 detik agar kendaraan dapat lewat, kemudian menutup kembali pintu.

i. Fungsi *Preprocessing* dan *OCR*

```
def preprocess_and_ocr(img):
 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
 gray = cv2.bilateralFilter(gray, 11, 17, 17)
 thresh = cv2.threshold(gray, 100, 255, cv2.THRESH_BINARY + 
cv2.THRESH_OTSU)[1]

 text = pytesseract.image_to_string(thresh, config='--psm 7')
 text = re.sub(r'\W+', '', text)
 return text, thresh
```

Fungsi ini memproses citra plat sebelum dilakukan *OCR*:

- *Grayscale* yang dimana Mengubah citra menjadi hitam putih untuk mengurangi kompleksitas.
- *Bilateral Filter* yang bisa engurangi noise sambil mempertahankan tepi karakter.
- *Thresholding Otsu* yang mengakibatkan gambar menjadi hitam-putih optimal.
- *OCR* yang dimana menggunakan *--psm* 7 yang mengasumsikan hanya ada satu baris teks pada citra.
- *Regular Expression* yang dapat menghapus semua karakter selain huruf dan angka.

j. Inisialisasi Kamera

```
cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
cap.set(cv2.CAP_PROP_FPS, 30)
```

Mengatur kamera untuk mengambil video dengan resolusi 640x480 piksel pada kecepatan 30 *FPS*, sehingga menghasilkan kinerja yang seimbang antara kualitas gambar dan kecepatan proses.

k. Proses Deteksi dan Pembacaan Plat

Bagian utama program berada pada *loop while True*, di mana frame dari kamera diproses:

```
results = model.predict(source=frame, conf=0.5,
verbose=False)
```

YOLO mendeteksi plat nomor dengan *confidence threshold* 0.5. Hasil deteksi berupa koordinat kotak pembatas yang digunakan untuk memotong bagian gambar plat nomor, kemudian dilakukan *OCR*.

Jika teks berhasil dibaca, sistem menyimpannya ke folder deteksi_plat dengan format nama [nomorplat]_[*timestamp*].jpg. Apabila nomor plat terverifikasi, pintu portal akan terbuka melalui pemanggilan fungsi open_gate().

l. Menampilkan Hasil pada Layar

```
cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
cv2.putText(frame, text, (x1, y1 - 10),
 cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
cv2.imshow("Deteksi Plat", frame)
```

Sistem memberikan umpan balik visual berupa kotak hijau di sekitar plat yang terdeteksi, serta menampilkan teks hasil pembacaan *OCR* di atas kotak tersebut.

m. Penanganan Penhentian Program

```
except KeyboardInterrupt:
 print("Program dihentikan.")
finally:
 pwm.stop()
 GPIO.cleanup()
 cap.release()
 cv2.destroyAllWindows()
```

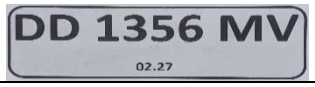
Bagian ini memastikan bahwa ketika program dihentikan dengan kombinasi tombol *Ctrl+C*, semua sumber daya dilepaskan dengan aman. *PWM* servo dihentikan, pin *GPIO* di-reset, kamera dilepaskan, dan semua jendela *OpenCV* ditutup.

C. **Pengujian**

Pada tahap ini dilakukan serangkaian pengujian terhadap sistem pendeteksi nomor plat kendaraan menggunakan *YOLO* sebagai pendeteksi objek dan *Tesseract OCR* sebagai pembaca teks. Pengujian dilakukan dengan variasi kondisi lingkungan (siang dan malam), variasi jarak (50 cm dan 80cm), serta variasi sudut pengambilan

gambar (15° dan 30°). Tujuan dari pengujian ini adalah untuk mengevaluasi tingkat keberhasilan deteksi plat, akurasi hasil pembacaan teks *OCR*, serta keterhubungan dengan status portal otomatis yang dikendalikan berdasarkan hasil dari verifikasi nomor plat ditunjukkan pada Tabel 4.1.

Tabel 4.1 Klasifikasi plat nomor yang digunakan

Jenis Plat Nomor Kendaraan	Status
	Terverifikasi
	Terverifikasi
	Terverifikasi
	Terverifikasi
	Terverifikasi

Tabel 4.2 Hasil pengujian plat putih DK 3153 XZ

No	Pengujian	Hasil Deteksi *YOLO*	Hasil *OCR* (*Tesseract*)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	pK3153XZ	Portal tidak terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	Pe3153XZ	Portal tidak terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	Pk3183i	Portal tidak terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	a3153XZ	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	K3153tz	Portal tidak terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	OK3135XZ	Portal tidak terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	BK3153	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	Ok3153	Portal tidak terbuka

Tabel 4.3 Hasil pengujian plat kuning DP 2724 AY

No	Pengujian	Hasil Deteksi *YOLO*	Hasil *OCR* (*Tesseract*)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	m214AY	Portal tidak terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	mp2124AY	Portal tidak terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	mp27gAY	Portal tidak terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	2104AY	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	mp2724AY	Portal tidak terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	w2124AY	Portal tidak terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	MP2724ay	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	m2124AY	Portal tidak terbuka

Tabel 4.4 Hasil pengujian plat hitam DP 2269 PJ

No	Pengujian	Hasil Deteksi *YOLO*	Hasil *OCR* (*Tesseract*)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	oF2269Ps	Portal tidak terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	pe1269Pd	Portal tidak terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	pP2269Ps	Portal tidak terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	pr7268	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	mF2265PJ	Portal tidak terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	DP2268Pd	Portal tidak terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	P2268PD	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	p2268PJ	Portal tidak terbuka

Tabel 4.5 Hasil pengujian plat merah DP 2724 AY

No	Pengujian	Hasil Deteksi *YOLO*	Hasil *OCR* (*Tesseract*)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	oP2124AY	Portal tidak terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	uP2704AY	Portal tidak terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	OP2724AY	Portal tidak terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	oP2724yy	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	oP2724AY	Portal tidak terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	ap2724AY	Portal tidak terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	oP2724AY	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	oP2724AY	Portal tidak terbuka

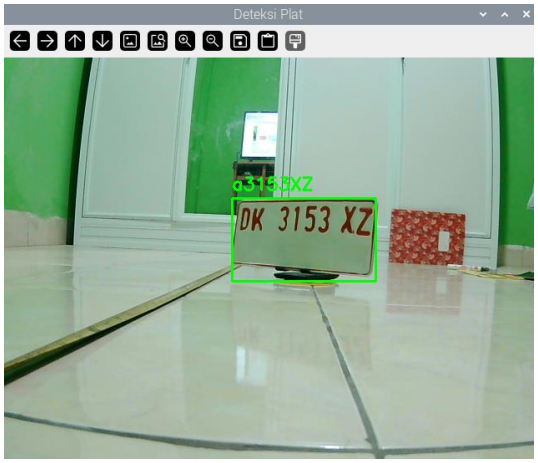
Tabel 4.6 Hasil pengujian plat cetak putih DD 1356 MV

No	Pengujian	Hasil Deteksi *YOLO*	Hasil OCR (*Tesseract*)	Status Portal
1	Siang, jarak 50 cm, sudut 15°	Plat terdeteksi	DD1356MV	Portal terbuka
2	Siang, jarak 50 cm, sudut 30°	Plat terdeteksi	DD1356MV	Portal terbuka
3	Siang, jarak 80 cm, sudut 15°	Plat terdeteksi	DD1356MV	Portal terbuka
4	Siang, jarak 80 cm, sudut 30°	Plat terdeteksi	OD1356MV	Portal tidak terbuka
5	Malam, jarak 50 cm, sudut 15°	Plat terdeteksi	DD1356MV	Portal terbuka
6	Malam, jarak 50 cm, sudut 30°	Plat terdeteksi	DD1356MV	Portal terbuka
7	Malam, jarak 80 cm, sudut 15°	Plat terdeteksi	OD1356MV	Portal tidak terbuka
8	Malam, jarak 80 cm, sudut 30°	Plat terdeteksi	iD1356MV	Portal tidak terbuka

Secara umum, *YOLO* mampu mendeteksi keberadaan plat dengan baik pada semua kondisi. Namun, tantangan terbesar muncul pada tahap *OCR* yang sering memberikan hasil pembacaan tidak sesuai dengan nomor plat sebenarnya. Akurasi *OCR* sangat dipengaruhi oleh jenis plat, kondisi pencahayaan, sudut pengambilan gambar, serta jarak kamera terhadap objek.

1. Analisis hasil pengujian Plat Putih DK 3153 XZ

Pengujian pada plat putih menunjukkan bahwa deteksi objek oleh *YOLO* berjalan dengan baik. Plat selalu terdeteksi, baik pada siang maupun malam hari, serta pada jarak dekat (50 cm) maupun lebih jauh (80 cm). Pada Gambar 4.6 plat nomor menunjukkan teks asli "*DK 3153 XZ*". Akan tetapi, hasil *OCR* malah membaca "*a3153XZ*". Perbedaan pembacaan ini tampak jelas bila dibandingkan dengan teks asli yang ditunjukkan pada Gambar 4.6 . Hal ini berdampak pada sistem validasi yang tidak pernah mengenali nomor plat sesuai dengan data yang terdaftar.



Gambar 4.6 Hasil pembacaan plat putih DK 3153 XZ

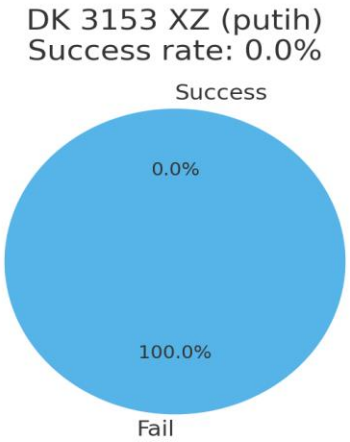
Dengan demikian, meskipun deteksi *YOLO* berhasil mendeteksi, akan tetapi *Tesseract OCR* gagal dalam melakukan pembacaan terhadap nomor plat yang tervefikasi sehingga portal tidak pernah terbuka yang ditunjukkan pada Gambar 4.7.



Gambar 4.7 Portal tidak terbuka

Sehingga presentase keberhasilan yang didapatkan pada pengujian plat nomor putih *DK 3153 XZ* yang ditunjukkan pada Tabel 4.2 yaitu:

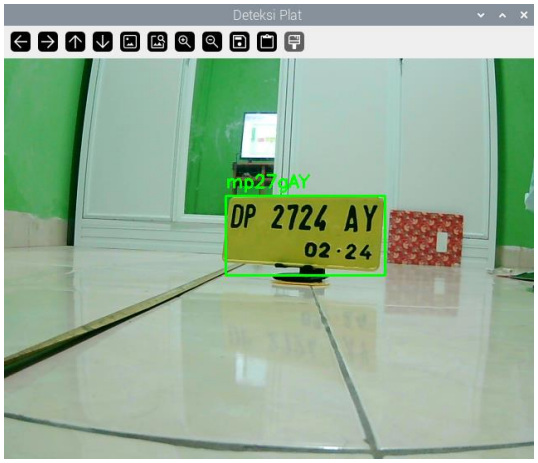
- Total pengujian: 8
- Berhasil (portal terbuka): 0
- Gagal (portal tidak terbuka): 8
- Tingkat keberhasilan: 0.0%



Gambar 4.8 Presentase keberhaasilan plat putih DK 3153 XZ

2. Analisis hasil pengujian Plat Kuning DP 2269 AY

Plat kuning juga berhasil terdeteksi dengan konsisten oleh *YOLO*, akan tetapi pada pembacaan *OCR* menunjukkan kesalahan serupa dalam melakukan pembacaan terhadap nomor plat kuning asli “DP 2269 AY” dengan hasil pembacaan nomor plat “mp27gAY” yang ditunjukkan pada Gambar 4.9.



Gambar 4.9 Hasil pembacaan plat nomor kuning DP 2724 AY

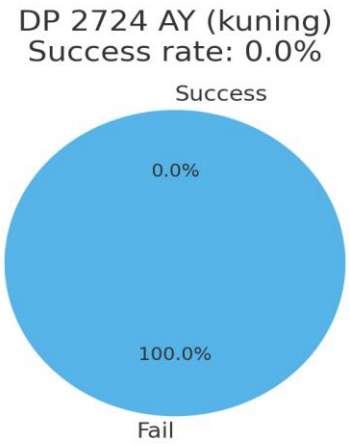
Kondisi ini menunjukkan bahwa *OCR* gagal dalam melakukan pembcaan nomor plat yang ada pada plat asli. Pada kondisi di malam hari, hasil pembacaan tidak lebih baik, menunjukkan bahwa minimnya sebuah pencahayaan tidak banyak membantu mengurangi *error*. Sama seperti pada plat putih, portal tidak pernah terbuka dikarenakan hasil pembacaan *OCR* tidak sesuai dengan data yang diverifikasi untuk membuka pintu portal yang ditunjukkan pada Gambar 4.10.



Gambar 4.10 Portal tidak terbuka

Sehingga presentase keberhasilan yang didapatkan pada pengujian plat nomor kuning DP 2724 AY yang ditunjukkan pada Tabel 4.3 yaitu:

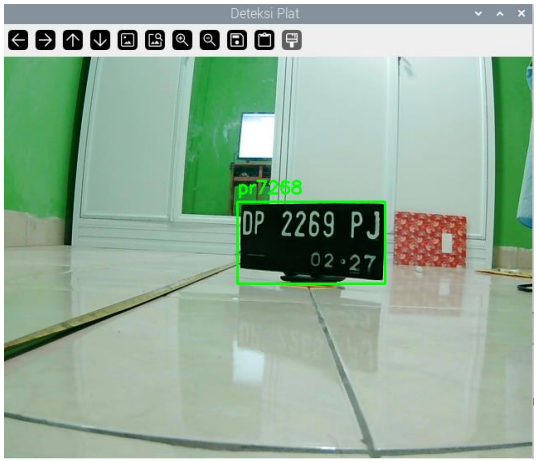
- Total pengujian: 8
- Berhasil (portal terbuka): 0
- Gagal (portal tidak terbuka): 8
- Tingkat keberhasilan: 0.0%



Gambar 4.11 Presentase keberhaasilan plat kuning DP 2724 AY

3. Analisis hasil Pengujian Plat Hitam DP 2269 PJ

Hasil pada plat hitam memperlihatkan tantangan yang lebih besar lagi. *YOLO* tetap mendeteksi plat dengan baik, akan tetapi *OCR* selalu gagal dalam membaca nomor plat dengan benar pada plat hitam "*DP 2269 PJ*". Sehingga hasil dari pembacaan *OCR* yang ditampilkan "*pr7268*" jauh dari nomor asli pada plat kendaraan yang ditunjukkan pada Gambar 4.12.



Gambar 4.12 Hasil pembacaan plat nomor hitam DP 2269PJ

Hasil ini menunjukkan bahwa sistem *OCR* gagal dalam melakukan pembacaan terhadap nomor plat, sehingga tidak ada satupun pengujian yang berhasil membuka portal dikarenakan data hasil pembacaan tidak sesuai dengan data verifikasi yang ditunjukkan pada Gambar 4.13.

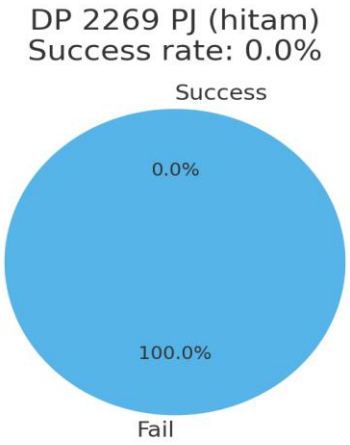


Gambar 4.13 Portal tidak terbuka

Sehingga presentase keberhasilan yang didapatkan pada pengujian plat nomor hitam DP 2269 PJ yang ditunjukkan pada Tabel 4.4 yaitu:

- Total pengujian: 8
- Berhasil (portal terbuka): 0

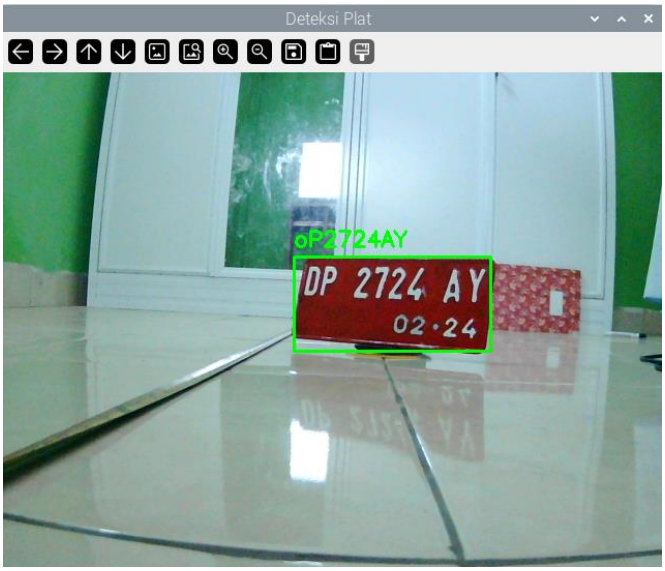
- Gagal (portal tidak terbuka): 8
- Tingkat keberhasilan: 0.0%



Gambar 4.14 Presentase keberhaasilan plat hitam DP 2269 PJ

4. Analisis Hasil Pengujian Plat Merah DP 2724 AY

Pengujian plat merah menunjukkan hasil yang sedikit lebih baik dari pada plat kuning dan hitam, tetapi tetap belum akurat. *YOLO* kembali berhasil mendeteksi plat, namun hasil *OCR* tetap gagal dalam pembacaan nomor plat yang ditunjukkan pada Gambar 4.15.



Gambar 4.15 Hasil pembacaan plat nomor merah DP 2724 AY

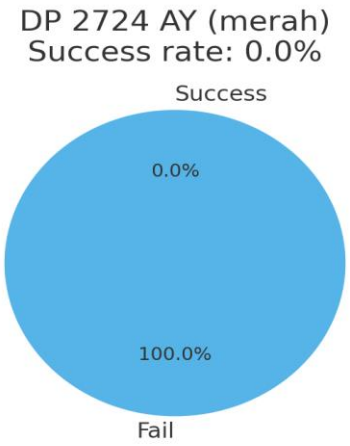
Kesalahan *OCR* yang ditunjukkan pada Gambar 4.15 dengan hasil pembacaan di atas seperti “oP2724AY” atau “uP2704AY”. Kondisi ini mengindikasikan bahwa belum cukup untuk menghasilkan pembacaan yang valid. Sehingga portal tetap tidak pernah terbuka karena sistem validasi membutuhkan kecocokan penuh antara hasil *OCR* dan data plat yang terdaftar yang ditunjukkan pada Gambar 4.16.



Gambar 4.16 Portal tidak terbuka

Sehingga presentase keberhasilan yang didapatkan pada pengujian plat nomor merah DP 2724 AY yang ditunjukkan pada Tabel 4.5 yaitu:

- Total pengujian: 8
- Berhasil (portal terbuka): 0
- Gagal (portal tidak terbuka): 8
- Tingkat keberhasilan: 0.0%



Gambar 4.17 Presentase keberhaasilan plat merah DP 2724 AY

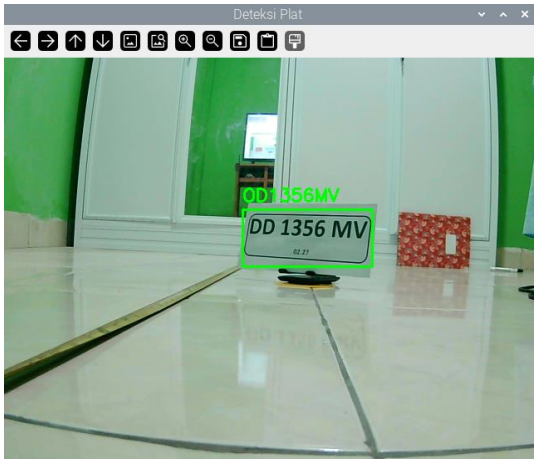
5. Analisis Hasil Pengujian Plat Cetak Putih DD 1356 MV

Plat cetak putih memberikan hasil yang berbeda dibandingkan keempat plat sebelumnya. Deteksi *YOLO* tetap konsisten, tetapi yang lebih penting adalah hasil *OCR* jauh lebih akurat. Sebagian besar hasil pembacaan sesuai dengan nomor asli "DD1356MV", baik pada kondisi siang maupun malam, serta pada jarak dekat maupun agak jauh yang ditunjukkan pada Gambar 4.18.

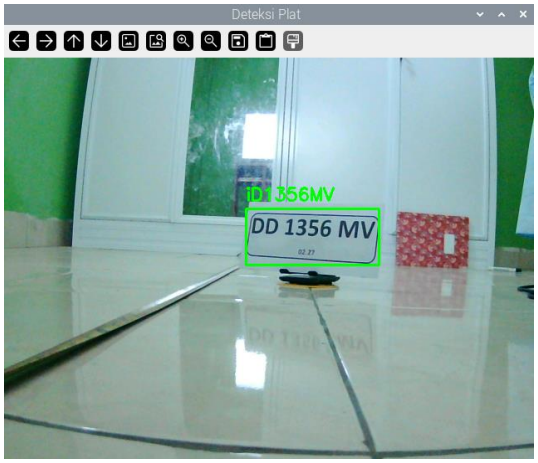


Gambar 4.18 Hasil pembacaan plat nomor cetak putih DD 1356 MV

Hanya pada kondisi tertentu, terutama sudut 30° atau pada pencahayaan malam, terjadi sedikit kesalahan seperti “OD1356MV” atau “iD1356MV” yang ditunjukkan pada Gambar 4.19 dan Gambar 4.20



Gambar 4.19 Hasil pembacaan plat nomor cetak putih DD 1356 MV



Gambar 4.20 Hasil pembacaan plat nomor cetak putih DD 1356 MV

Namun, kesalahan tersebut relatif minor dibandingkan *error* pada plat lain. Hal ini menyebabkan portal beberapa kali berhasil terbuka, menunjukkan bahwa sistem validasi dapat berjalan sesuai fungsinya apabila *OCR* membaca nomor plat dengan benar yang ditunjukkan pada Gambar 4.21.

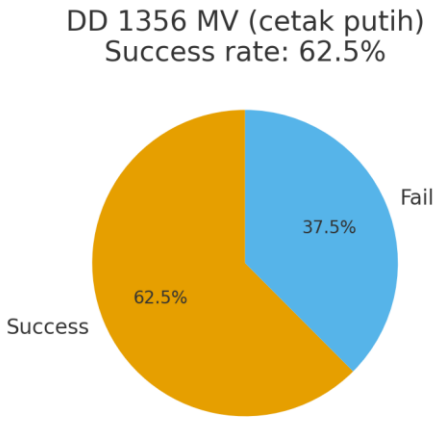


Gambar 4.21 Portal terbuka

Keberhasilan ini dapat disebabkan oleh kualitas cetakan plat yang lebih jelas, font yang tegas, serta kontras hitam-putih yang lebih ideal untuk dikenali oleh *Tesseract OCR*. Dengan demikian, plat cetak putih bisa dianggap sebagai kondisi terbaik dalam pengujian ini.

Sehingga presentase keberhasilan yang didapatkan pada pengujian plat nomor cetak putih *DD 1356 MV* yang ditunjukkan pada Tabel 4.6 yaitu:

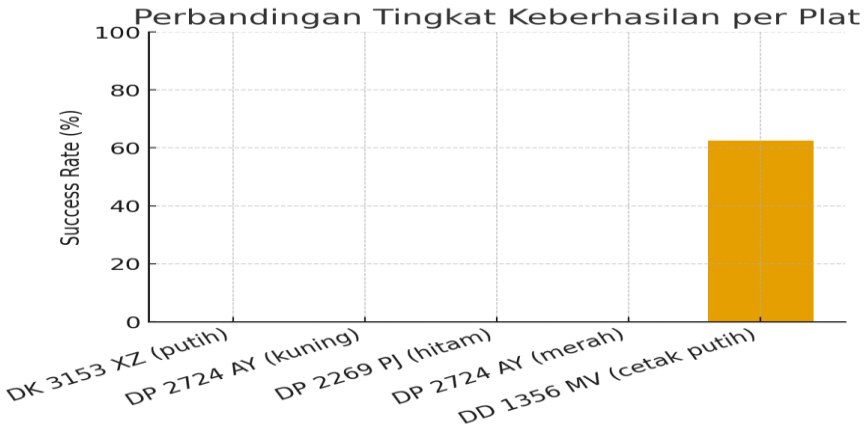
- Total pengujian: 8
- Berhasil (portal terbuka): 5
- Gagal (portal tidak terbuka): 3
- Tingkat keberhasilan: 62.5%



Gambar 4.22 Presentase keberhaasilan plat cetak putih DD 1356 MV

Hasil perbandingan keseluruhan dari presentase keberhasilan, dapat di lihat bahwa hanya plat nomor cetak putih (DD 1356 MV) yang terdeteksi benar mayoritas (5/8 = 62,5%). Sedangkan plat biasa (putih, kuning, hitam, merah) semua **0/8** (0%) berhasil. Data lengkap terdapat pada Tabel 4.2 sampai Tabel 4.6 pengujian yang menunjukkan tiap kasus uji dan hasil *OCR*-nya. Adapun rumus yang digunakan untuk mengetahui presentase tingkat keberhasilan yaitu:

$$\frac{jumlah\ berhasil}{Total\ Percobaan} \times 100\%$$



Gambar 4.23 Grafik perbandingan keberhasilan per plat

Sehingga dapat kita disimpulkan bahwa sistem *OCR* yang diuji mampu membuka portal pada ~62,5% kasus plat cetak putih “DD 1356 MV”, namun 0% pada plat biasa.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan dari hasil perancangan, implementasi dan pengujian sistem secara keseluruhan dapat disimpulkan dalam beberapa hal sebagai berikut:

1. Deteksi YOLO

YOLO terbukti konsisten mendeteksi keberadaan plat nomor pada semua kondisi, tanpa terpengaruh warna, jarak, atau sudut. Artinya, modul deteksi objek bekerja dengan baik dan tidak menjadi faktor penghambat utama.

2. Kinerja OCR

Tantangan terbesar ada pada Tesseract OCR yang kesulitan membaca karakter pada plat asli (putih, kuning, hitam, merah). Kesalahan meliputi substitusi karakter, hilangnya sebagian huruf/angka, hingga perubahan total. Plat kuning dan hitam memiliki tingkat kesalahan tertinggi, plat merah sedikit lebih baik, tetapi masih jauh dari akurat, dan plat cetak putih jauh lebih akurat dan terbaca dengan benar pada sebagian besar kondisi.

3. Pengaruh Pencahayaan dan Sudut

Pencahayaan malam menambah kesulitan OCR, terutama pada plat hitam. Sudut kemiringan 30° meningkatkan kesalahan pembacaan dibandingkan 15°, karena distorsi visual pada karakter, dan jarak 50 cm hingga 80 cm masih memungkinkan deteksi yang baik, tetapi semakin jauh, akurasi OCR menurun.

B. Saran

1. Sebelum OCR, lakukan penyesuaian citra seperti peningkatan kontras atau pengurangan noise. Misalnya: konversi ke grayscale, *contrast limited adaptive histogram equalization* (CLAHE), atau filter bilateral. Ini membantu mengatasi penerangan rendah dan sudut ekstrim.
2. Perluas dataset pelat dengan variasi warna latar dan font. Latih ulang model OCR agar sensitif pada kontras beragam (mis. teks putih di plat hitam dan sebaliknya).
3. Terapkan cek pola plat (misalnya awalan “DD” untuk plat khusus) untuk menolak hasil salah. Koreksi pasca-OCR seperti membedakan ‘O’ dan ‘0’ dapat menambah akurasi.

DAFTAR PUSTAKA

Abdurrahman, M. W. (2012). Pengembangan aplikasi translator Sunda-Indonesia-Inggris menggunakan capture camera pada smartphone Android.

Alandani, R. (2017). Pengembangan prototipe deteksi kebocoran pipa migas berbasis kamera long wave infrared pada aplikasi unmanned aerial vehicle development of pipe leak detection based on long wave infrared camera for unmanned aerial vehicle application.

Alfarizi, D. N., Pangestu, R. A., Aditya, D., Setiawan, M. A., & Rosyani, P. (2023). Penggunaan metode YOLO pada deteksi objek: Sebuah tinjauan literatur sistematis. *AI dan SPK: Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan, 1*(1), 54–63.

Anggraini, R. (2019). Rancang bangun prototipe stasiun klimatologi terintegrasi dengan Internet of Things (IoT).

Anjani, N. (2024). Aplikasi deteksi ekspresi pada wajah dengan mesin learning (Doctoral dissertation, Universitas Muhammadiyah Parepare).

Apriyanti, K., & Widodo, T. W. (2016). Implementasi optical character recognition berbasis backpropagation untuk text to speech perangkat Android. *IJEIS, 6*(1), 13–24.

Bahri, R. S. (2012). Perbandingan algoritma template matching dan feature extraction pada optical character recognition. *KOMPUTA: Jurnal Komputer dan Informatika, 1*(1).

Dermawan, M. S., Mulyawan, B., & Lauro, M. D. (2019). Perancangan aplikasi sistem manajemen dokumen dan pencarian teks dengan menggunakan optical character recognition (OCR). *Jurnal Ilmu Komputer dan Sistem Informasi, 7*(1), 81–86.

Fajar, M. (2015). Perancangan aplikasi penelurusan informasi persuratan berbasiskan optical character recognition (OCR) (Studi kasus Fakultas Sains dan Teknologi UIN Jakarta).

Felisa, J., Setiawan, D., & Khalisa, I. (2022). Perancangan perangkat lunak pengenalan karakter plat nomor kendaraan dengan metode convolutional neural network. *Media Informatika, 21*(3), 280–306.

Hangkawidjaja, I. A. D., MT, D. Y. A. P., & Jarden, J. J. (2021). Perancangan dan realisasi detektor retak permukaan dinding bangunan. Penerbit YLGI.

Hidayatullah, M. A. (2019). Monitoring area parkir jurusan teknik komputer menggunakan CCTV berbasis Raspberry Pi (Doctoral dissertation, Politeknik Negeri Sriwijaya).

Hutauruk, J. S. W., Matulatan, T., & Hayaty, N. (2020). Deteksi kendaraan secara real time menggunakan metode YOLO berbasis Android. *Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan, 9*(1), 8–14.

Ibrahim, R. M. (2023). Alat bantu baca tunanetra pengubah teks menjadi suara dengan Tesseract OCR dan Raspberry Pi (Doctoral dissertation, Universitas Komputer Indonesia).

Khairunisa, N., & Jamaludin, A. (2024). Analisis perbandingan algoritma CNN dan YOLO dalam mengidentifikasi kerusakan jalan. *Jurnal Informatika dan Teknik Elektro Terapan, 12*(3).

Maximillian, L., Riti, Y. F., Agung, M. A., & Palis, Y. J. (2023). Perbandingan algoritma Sobel dan Canny untuk deteksi tepi citra daun lidah buaya. *KOMPUTA: Jurnal Ilmiah Komputer dan Informatika, 12*(2), 69–79.

Muhtadii, M., & Tawakal, H. A. (2016). Pengembangan aplikasi Android untuk pengenalan citra nomor sertifikat halal MUI dengan library Tesseract optical character recognition (OCR). *Jurnal Informatika Terpadu, 2*(1).

Muhamad Zulkarnaen, F. (2024). Sistem deteksi objek manusia menggunakan algoritma YOLOv8 berbasis kamera depth sensor (Studi kasus: CV. Ateri Global Teknologi) (Doctoral dissertation, Universitas Sangga Buana YPKP).

Muthoá, M. F. (2017). Sistem notifikasi bel rumah pada ponsel menggunakan Raspberry Pi (Doctoral dissertation, Politeknik Negeri Sriwijaya).

Nabuasa, Y. N. (2019). Pengolahan citra digital perbandingan metode histogram equalization dan spesification pada citra abu-abu. *JI Komputer, 7*(1), 87–95.

Nugraha, F. A., Harani, N. H., & Habibi, R. (2020). Analisis sentimen terhadap pembatasan sosial menggunakan deep learning. Kreatif.

Pranadipa, R., Novitasari, V., Kurniawati, M., Puspitasari, N., & Bonita, Y. (2012). Pengenalan angka pada plat nomor dengan metode template matching. Universitas Brawijaya.

Prasetyo, D. (2018). Aplikasi pendeteksi jerawat di wajah dengan menggunakan teknik pengolahan citra pada foto.

Puteri, N. R., & Meirza, A. (2024). Implementasi metode YOLOv5 dan Tesseract OCR untuk deteksi plat nomor kendaraan. *Journal of Computer Science and Visual Communication Design, 9*(1), 424–435.

Putro, E. C., Awangga, R. M., & Andarsyah, R. (2020). Tutorial object detection people with faster region-based convolutional neural network (Faster R-CNN) (Vol. 1). Kreatif.

Putro, H. P., Widyaningsih, T. W., Englishtina, I., Nursanty, E., & Dema, H. (2023). *Development of artificial intelligence applications (Studi kasus & implementasi AI menggunakan berbagai bahasa pemrograman).* PT. Sonpedia Publishing Indonesia.

Ramadhani, A., & Shalludin, A. (2020). Rancang bangun aplikasi pengarsipan dokumen menggunakan optical character recognition pada Badan Kepegawaian dan Pengembangan Sumber Daya Manusia Kabupaten Kapuas berbasis visual. *Jurnal Sains Komputer dan Teknologi Informasi, 3*(1), 76–82.

Rahman, R. A. H., Sunarto, A. A., & Asriyanik, A. (2024). Penerapan you only look once (YOLO) v8 untuk deteksi tingkat kematangan buah manggis. *JATI (Jurnal Mahasiswa Teknik Informatika), 8*(5), 10566–10571.

Rosidin, R., Novianti, R., Ningsih, K. P., Haryadi, D., Chrisnawati, G., & Anripa, N. (2024). Peran kecerdasan buatan dalam pengembangan sistem otomatisasi proses bisnis. *Jurnal Review Pendidikan dan Pengajaran (JRPP), 7*(3), 9320–9329.

Sandy, A. K. (2021). Rancang bangun sistem monitoring rumah dengan Raspberry Pi melalui sensor PIR dan aplikasi WhatsApp (Doctoral dissertation, Universitas Komputer Indonesia).

Saputra, R. D. (2023). Pengembangan sistem deteksi objek pada produk retail dengan arsitektur YOLOv4-tiny (Doctoral dissertation, Universitas Islam Indonesia).

Solekhan, S., & Iqbal, M. (2020). Media pembelajaran pemancar wireless FM menggunakan Raspberry Pi. *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer, 11*(1), 257–262.

Suarti, I. (2024). Deteksi situs phishing berbasis neural network (Doctoral dissertation, Universitas Islam Negeri Maulana Malik Ibrahim).

Suhartono, S., Zain, S. G., & Ardilla, A. (2024). Detection of vehicle type and license plate with convolutional neural network model YOLOv7. *Jurnal Teknik Informatika (JUTIF), 5*(2), 621–636.

Sulistiyo, A. (2019). Perancangan sistem text extraction menggunakan library Tesseract OCR untuk mengambil nomor induk kependudukan pada foto kartu tanda penduduk: Studi kasus PT. Bank ABC, Tbk (Doctoral dissertation, Universitas Kristen Satya Wacana).

Virgiawan, I., Maulana, F., Putra, M. A., Kurnia, D. D., & Sinduningrum, E. (2024). Deteksi dan tracking objek secara real time berbasis computer vision menggunakan metode YOLO v3. *Humantech: Jurnal Ilmiah Multidisiplin Indonesia, 3*(3).

Widiani, P. E. (2014). Pengembangan aplikasi mobile penterjemah bahasa Korea-Indonesia dengan OCR dan Bing translate API (Doctoral dissertation, Universitas Atma Jaya Yogyakarta).

Yuadi, I., Sos, S., & MT, M. (2023). *Forensik digital dan analisis citra.* CV. AE Media Grafika.

Yuliani, F. (2022). Analisis dan implementasi object tracking pada kamera webcam dengan image processing menggunakan metode mean shift.