

LAPORAN FINAL UJIAN AKHIR SEMESTER
MATA KULIAH PENGOLAHAN CITRA DIGITAL



Judul

**IMPLEMENTASI ALGORITMA YOLO DAN OPTICAL
CHARACTER RECOGNITION (OCR) PADA SISTEM DETEKSI
PLAT NOMOR KENDARAAN**

TIM:

- | | |
|-----------------------------------|-------------|
| 1. Divangga Revansa Arya Pradhana | (200103215) |
| 2. Raditya Koesyan Dipo Pramukti | (200103183) |
| 3. Bagus Muttaqi | (200103212) |

FAKULTAS ILMU KOMPUTER
UNIVERSITAS DUTA BANGSA
SURAKARTA

TAHUN 2024

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini, perkembangan teknologi telah berkembang secara cepat dan telah diimplementasikan dalam beberapa bidang. Salah satunya adalah dalam bidang keamanan dan ketertiban. Pada bidang tersebut kepolisian republik indonesia tengah menerapkan tilang elektronik yang dimana dapat diimplementasikan sebuah sistem deteksi plat nomor. Tilang elektronik memiliki kelebihan dalam kecepatan, kepraktisan, dan ketepatan jika dibandingkan dengan metode konvensional[1].

Dengan membangun sistem pendeteksi karakter plat nomor menggunakan algoritma YOLO v8 dan metode Optical Character Recognition (OCR) maka proses untuk mengenali identitas pemilik kendaraan apabila terjadi pelanggaran atau kecelakaan dapat dilakukan dengan tepat dan cepat. Proses identifikasi dimulai dengan pengambilan gambar atau citra melalui kamera webcam atau cctv yang selanjutnya dilakukan proses segmentasi dan konversi citra seperti konversi greyscale, bilateral filter, dan canny edge sehingga karakter dan numerik pada plat nomor dapat dikenali[2]. Dimana hal-hal tersebut merupakan bagian dari pengolahan citra digital.

Pemilihan topik Image Character Recognition mengenai plat nomor bukan lain ialah karena seringnya ditemui tilang elektronik yang menggunakan plat nomor sebagai objek utama untuk mengenali identitas pemilik untuk dilakukan penilangan, selain itu juga dikarenakan topik tersebut menerapkan beberapa proses pengolahan citra digital seperti segmentasi, konvolusi dan konversi citra untuk mendapatkan hasil gambar yang bagus dan jelas.

Project yang dikembangkan yaitu menggunakan dua metode, yaitu OCR dan YOLO. Metode OCR dilakukan dengan data lokal sehingga tidak diperlukan train data, sedangkan untuk metode YOLO diperlukan beberapa proses train data dalam beberapa epochs. Metode OCR dipilih karena akurasi yang tinggi, open source, dapat dikolaborasikan dengan library lain, dan kemampuan pre-processing seperti binarisasi dan masking/deteksi tepi yang dapat membantu meningkatkan akurasi pengenalan teks dan peningkatan kualitas gambar. Sedangkan algoritma YOLO v8

dpilih karena kecepatan, kemudahan penggunaan, dan terdapat banyak tutorial dan dokumentasi program di platform github.

1.2. Batasan Pengembangan Sistem

1. Pada metode OCR tidak dilakukan train data dan hanya menggunakan data lokal
2. Pada YOLO v8 - CNN tidak dilakukan train data dengan epoch yang banyak, sehingga dimungkinkan hasil yang kurang maksimal
3. Karakter yang dideteksi adalah alphabet dan numerik standar internasional (abc dan 123).

Landasan Teori

a. Pengolahan Citra digital

Citra digital adalah sebuah dokumentasi visual atau gambar yang ditangkap dengan alat atau properti digital seperti kamera, satelit, smartphone, scanner, dan sebagainya. Sedangkan pengolahan citra digital merupakan bidang ilmu yang mempelajari tentang bagaimana citra dibentuk, diolah, dan dianalisis sehingga dapat dikenali dan dipahami oleh manusia[3].

b. YOLO v8

Algoritma YOLO atau You Look Only Once adalah sebuah algoritma yang dikembangkan untuk proses deteksi objek secara real-time dengan menggunakan sistem deteksi repurpose classifier atau localizer. Yolo v8 adalah versi terbaru dari YOLO (You Only Look Once) yang dikembangkan oleh *ultralytics*[4].

c. Optical Character Recognition

OCR adalah teknologi yang digunakan untuk mendeteksi teks atau karakter yang terdapat di dalam gambar, foto atau citra dengan hasil deteksi berupa karakter dan numerik. Saat ini telah banyak penelitian yang menerapkan metode Optical Character Recognition, karena metode tersebut dianggap dapat dijadikan sebagai solusi untuk pengenalan plat kendaraan bermotor[5].

d. Tilang Elektronik

E-tilang atau tilang elektronik merupakan implementasi tilang berbasis teknologi yang diluncurkan tahun 2016 oleh Kepolisian Negara Republik Indonesia(POLRI) guna mengurangi tindakan pungli atau pungutan liar oleh oknum-oknum yang tidak bertanggung jawab[6]. Tilang elektronik memanfaatkan fungsi cctv yang dipasang di tiang lampu lalu lintas untuk mengetahui apabila ada pengendara yang tidak taat aturan berlalu lintas dengan mengenali pengendara atau pemilik kendaraan tersebut dengan

mendeteksi plat nomor kendaraan tersebut.

e. Konvolusi Citra

Konvolusi citra adalah metode untuk memperhalus atau memperjelas suatu gambar dengan menggantikan nilai-nilai pixel dengan sejumlah nilai pixel yang sesuai atau berdekatan dengan pixel aslinya dengan tetap mempertahankan ukuran citra yang asli[7].

f. Ultralytics

Ultralytics adalah pustaka sumber terbuka yang menyediakan implementasi YOLOv8, sebuah arsitektur implementasi objek yang terkenal dalam bidang penglihatan komputer[8].

g. Matplot Library

Matplotlib merupakan pustaka open source dari python yang dapat digunakan sebagai visualisasi data. Dapat menampilkan sebuah grafik, plot, dan gambar.

h. Pandas

Pandas adalah sebuah paket pustaka Python open source yang digunakan untuk mengolah dan menganalisis data terstruktur. Pandas singkatan dari "Panel Data" atau "Python Data Analysis".

i. Imutils

Imutils adalah library dari python yang berfungsi untuk pengambilan nilai kontur dari sebuah objek referensi[9]. Imutils merupakan salah satu library python yang digunakan untuk melakukan image processing seperti translasi, rotasi, perubahan ukuran, menampilkan kerangka, menampilkan gambar dari matplotlib, klasifikasi kontur, dan deteksi tepi[10].

j. Numpy

Numpy adalah kependekan dari Numerical Python, berfungsi untuk menjalankan operasi vektor matriks dengan mengolah array[10]. Numpy biasa digunakan untuk kebutuhan analisis data.

Alat dan Bahan

No.	Nama	Jenis	Fungsi
1.	Google Colab	Online IDE (software)	Memrogram dan menjalankan program deteksi plat nomor

			berbasis YOLO v8
2.	Visual Studio Code	Offline IDE (Software)	Memrogram dan menjalankan program deteksi plat nomor berbasis tesseract-OCR
3.	Data gambar	Citra	Bahan Uji / Percobaan
4.	Laptop	Hardware	Alat utama untuk menjalankan semua software
5.	CV2	Library Python	Mengolah citra objek secara umum, baik manusia atau objek kotak.
6.	PIL	Library citra python	Memproses citra atau gambar.

Konfigurasi Awal

- Instal ultralytics di notebook google collab
- Koneksi drive google
- Impor modul atau library yang diperlukan
- Unduh dan link dataset melalui kode
- Mengatur direktori target
- Menentukan direktori path dari data uji
- Membuat kode dan run di setiap potongan kodenya.

PENGEMBANGAN

Deskripsi Sistem

a) Metode OCR

Program dibangun dengan beberapa library seperti CV2, imutils, numpy, pandas, dan yang paling utama ialah tesseract-OCR. Sistem yang berjalan pada metode OCR yaitu bertahap dimulai dari:

- 1) Pembacaan dan mengatur ukuran citra,
- 2) Lalu citra tersebut dikonversi ke dalam bentuk grayscale,
- 3) Dilakukan blurring menggunakan metode bilateral filtering,
- 4) Deteksi tepi dan pencarian kontur pada tepi gambar,
- 5) Masking bagian yang bukan termasuk bagian dari plat (misal gambar yang dipilih adalah gambar full terlihat kendaraannya,
- 6) Proses deteksi karakter menggunakan tesseract-OCR,
- 7) Karakter terdeteksi dan ditampilkan
- 8) Simpan data hasil deteksi ke dokumen excel atau .xls.

Program deteksi karakter pada plat nomor ini dapat dijalankan melalui VS Code yang berupa aplikasi windows sederhana. Di dalamnya user dapat menginputkan atau memilih gambar/citra dari sebuah plat nomor yang hendak di deteksi karakternya. Hasil dari deteksi akan dimunculkan pada terminal IDE dalam bentuk alphabet dan numerik, serta akan dimasukkan ke dalam data rekap dalam bentuk excel.

b) Metode Algoritma YOLO v8-CNN

Pada YOLO v8 library yang digunakan adalah ultralytics, google.colab.drive, PIL.Image dan matplotlib.pyplot. Tahapan pada sistem yang menggunakan metode YOLO v8 sebagai berikut :

- 1) Instalasi dan Setup Lingkungan.
- 2) Akses Data dari Google Drive.
- 3) Persiapan Data.
- 4) Pelatihan Model.
- 5) Prediksi Menggunakan Model Terlatih.
- 6) Visualisasi Hasil.

Kelebihan dan Kekurangan

a. Metode OCR

- Kelebihan
 - 1) Karakter alphabet dan numerik yang dihasilkan akurat
 - 2) Tidak diperlukan train data, sehingga dapat langsung memilih gambar mana yang hendak dideteksi
 - 3) Dapat dijalankan secara offline
 - 4) Terdapat UI dan UX yang dapat dijalankan melalui windows.
- Kekurangan
 - 1) Terdapat error jika klik deteksi lebih dari satu kali dengan tanpa memilih gambar baru
 - 2) Hanya dapat mendeteksi plat nomor berlatarbelakang putih dan fill karakter berwarna hitam
 - 3) Sensitif terhadap gambar yang memiliki banyak objek di dalamnya dan gambar yang diambil dengan posisi tidak tegak lurus, sehingga karakter sulit dideteksi.

b. Metode YOLO v8

- Kelebihan
 - 1) YOLOv8 dikenal dengan kemampuannya dalam melakukan deteksi objek secara real-time dengan kecepatan tinggi.
 - 2) Akurasi yang tinggi dalam mendeteksi dan mengenali plat kendaraan.
 - 3) Dengan pustaka ultralytics, implementasi YOLOv8 menjadi sangat mudah dan user-friendly.
 - 4) Dengan visualisasi hasil deteksi evaluasi kinerja model dan pemahaman tentang bagaimana model bekerja lebih mudah.
- Kekurangan
 - 1) Untuk mencapai akurasi tinggi, YOLOv8 memerlukan dataset yang besar dan berkualitas tinggi.
 - 2) Kinerja model dapat menurun jika data pelatihan tidak mencakup variasi yang cukup seperti kondisi pencahayaan yang berbeda, sudut pengambilan gambar, dan variasi plat kendaraan.
 - 3) Pelatihan model YOLOv8 memerlukan waktu yang cukup lama, terutama jika dataset yang digunakan sangat besar.

Deskripsi Dataset

Projek ini dikembangkan menggunakan dua metode, yaitu OCR dan YOLO. Metode

OCR dilakukan dengan data lokal jadi tidak diperlukan train data, hanya diperlukan data uji/data test. Untuk metode YOLO data yang digunakan sebagai masukan awal dari sistem pengenalan plat nomor kendaraan dibagi menjadi tiga, yaitu data train, data validasi dan data test.

Data untuk penelitian ini dikumpulkan melalui dataset dari Universe Roboflow yang diunggah pada tahun 2021. Dataset ini berisi total 1948 gambar yang menjadi dasar untuk melatih dan menguji sistem deteksi plat kendaraan. Berikut contoh datasetnya :

No	Dataset	Keterangan
1.		AA03BOJ
2.		AA14JAQ
3.		AA73AKA
4.		AA78XAY
5.		AB01UKH

6.		AB01EGM
7.		AB38YRQ
8.		AB58LSB
9.		B6349TAP
10		G1234AD

Split Dataset

Pada YOLO v8 split dataset yang dilakukan yaitu : Data Train sebanyak 87% (1704 Gambar), Data Validasi sebanyak 9% (166 Gambar) dan Data Test 4% (78 Gambar).

Epoch

Pada sistem pendeteksi plat kendaraan yang menggunakan metode YOLOv8, Telah dilakukan pelatihan selama 5 epoch. Dalam durasi **Epoch 1:** 11:42 menit, **Epoch 2:** 11:38 menit, **Epoch 3:** 11:36 menit, **Epoch 4:** 11:42 menit dan **Epoch 5:** 11:34 menit. Total durasi pelatihan selama 5 epoch adalah sekitar 1 jam 1 menit. Dengan penurunan yang konsisten dalam loss menunjukkan bahwa model belajar dan memperbaiki diri selama proses pelatihan. Ada peningkatan yang jelas dalam precision, namun recall mengalami fluktuasi. Ini menunjukkan bahwa model semakin baik dalam mengidentifikasi objek yang benar, tetapi mungkin masih kehilangan beberapa objek yang sebenarnya ada (false negatives).

Implementasi dan Source Code

- Implementasi Program pada YOLO v8

Langkah 1: Menginstal pustaka “ultralytics”.

```
%pip install ultralytics
```

Langkah 2: Mengimpor pustaka dan mengecek apakah semua persyaratan sistem terpenuhi.

```
import ultralytics
ultralytics.checks()
```

Langkah 3: Mount Google Drive ke Google Colab untuk mengakses file data.

```
from google.colab import drive
drive.mount('/content/drive')
```

Langkah 4: Mengekstrak file ZIP yang berisi dataset.

```
!unzip /content/drive/MyDrive/PROGRAMS/carPlate_CHAR.v1-2021-12-11-char.yolov8.zip
```

Langkah 5: Melatih model YOLOv8 menggunakan dataset yang telah diekstrak, dengan konfigurasi yang ditentukan dalam file “data.yaml”.

```
# train YOLOv8n on COCO8 for epochs
!yolo train model=yolov8n.pt data=/content/data.yaml epochs=5 imgsz=416
```

Langkah 6: Menggunakan model yang telah dilatih untuk membuat prediksi pada gambar baru.

```
!yolo predict model=/content/runs/detect/train3/weights/best.pt source="/content/test/images/carPlates_00195.jpg.rf.d6730747a4f35138f0ac41873448f2ba.
```

Langkah 7: Mengimpor pustaka untuk visualisasi dan membuka gambar asli serta gambar hasil prediksi.

```

from PIL import Image
import matplotlib.pyplot as plt

# Load the images
image1_path = '/content/test/images/carPlates_00195_jpg.rf.d6730747a4f35138f0ac41873448f2ba.jpg'
image2_path = '/content/runs/detect/predict/carPlates_00195_jpg.rf.d6730747a4f35138f0ac41873448f2ba.jpg'

# Open images using PIL
image1 = Image.open(image1_path)
image2 = Image.open(image2_path)

```

Langkah 8: Membuat visualisasi gambar asli dan gambar hasil prediksi dengan “matplotlib”.

```

# Create a figure with two subplots
fig, axes = plt.subplots(1, 2, figsize=(10, 5))

# Display the first image
axes[0].imshow(image1)
axes[0].axis('off') # Turn off axis labels
axes[0].set_title('Image Awal')

# Display the second image
axes[1].imshow(image2)
axes[1].axis('off') # Turn off axis labels
axes[1].set_title('Image Predict')

# Show the plot
plt.show()

```

- Implementasi program pada metode OCR

Langkah 1: Import library yang dibutuhkan

```

import time
import cv2
import imutils
import numpy as np
import pandas as pd
import pytesseract

pytesseract.pytesseract.tesseract_cmd = r"C:\Program
Files\Tesseract-OCR\tesseract.exe"

```

Langkah 2: Baca dan Atur ukuran citra

```

image = cv2.imread('Plat 1.jpg')
image = imutils.resize(image, width=500)

```

Langkah 3: Konversi citra ke mode grayscale

```

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Gambar Grayscale", gray)

```

Langkah 4: Blurring gambar untuk mereduksi noise

```

bilateral = cv2.bilateralFilter(gray, 11, 17, 17)

```

```

cv2.imshow("Bilateral Filter", bilateral)

kernel = np.ones((5,5),np.uint8)
opening = cv2.morphologyEx(bilateral, cv2.MORPH_OPEN, kernel)
cv2.imshow("Opening", opening)

gradient = cv2.morphologyEx(opening, cv2.MORPH_GRADIENT, kernel)
cv2.imshow("Gradient", gradient)

```

Langkah 5: Deteksi tepi dan cari kontur pada objek

```

# Deteksi tepi gambar
edged = cv2.Canny(gradient, 170, 200)
cv2.imshow("Canny Edges", edged)

# Cari kontur pada tepi gambar
(cnts, _) = cv2.findContours(edged.copy(), cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
cnts = sorted(cnts, key=cv2.contourArea, reverse=True)[:30]
NumberPlateCnt = None
#count = 0
# Loop kontur
for c in cnts:
    # Perkirakan konturnya
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, 0.02 * peri, True)
    # jika kontur yang diperkirakan memiliki empat titik, maka
    asumsikan bahwa layar ditemukan
    if len(approx) == 4:
        NumberPlateCnt = approx
        break

cv2.drawContours(image, [NumberPlateCnt], -1, (0,255,0), 3)

```

Langkah 6: Masking bagian yang selain plat nomor

```

mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [NumberPlateCnt], 0, 255, -1)
new_image = cv2.bitwise_and(image, image, mask=mask)
cv2.namedWindow("Gambar Final", cv2.WINDOW_NORMAL)
cv2.imshow("Gambar Final", new_image)

```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Langkah 7: Konfigurasi dan jalankan tesseract-OCR

```
config = ('-l eng --oem 1 --psm 3')

# jalankan tesseract OCR pada gambar
text = pytesseract.image_to_string(new_image, config=config)
print(" Plat Nomornya Adalah: ", text)
```

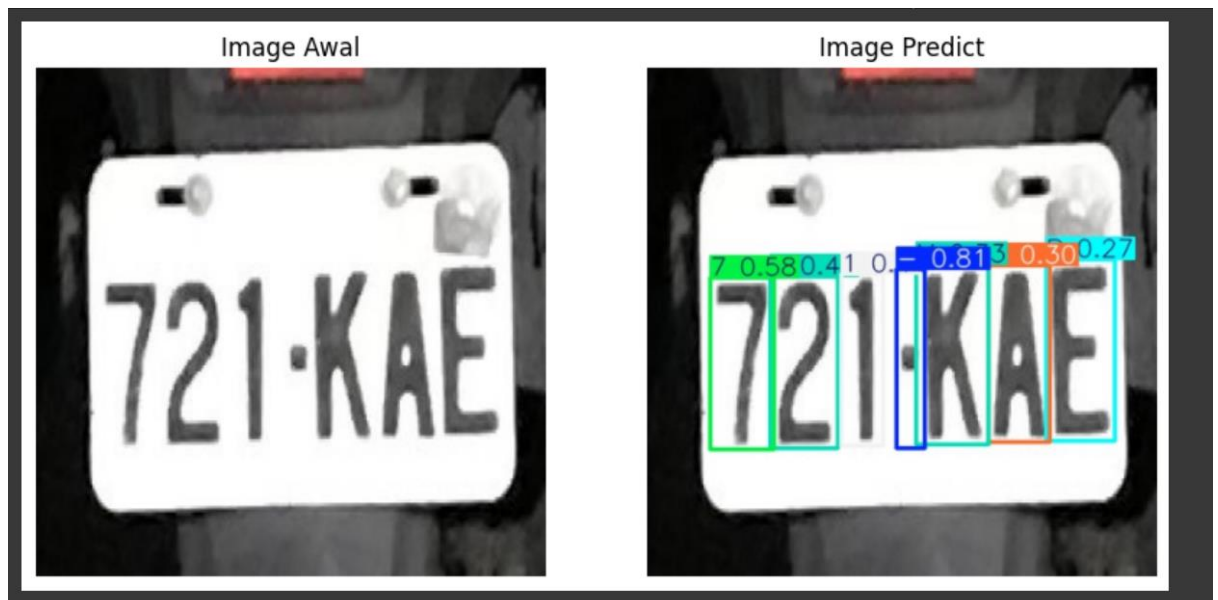
Langkah 8: Ekspor data deteksi ke file csv

```
raw_data = {'Tanggal Pendeteksian : ':
[time.asctime(time.localtime(time.time()))], '': [text]}
df = pd.DataFrame(raw_data)
df.to_csv('Data Plat Nomor Yang Terdeteksi.csv', mode='a')

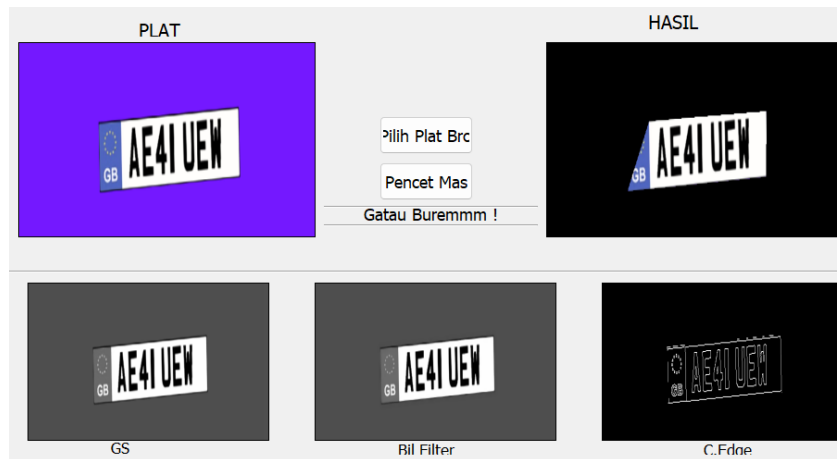
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Testing dan Hasil Akhir

- Metode YOLOv8-CNN



- Metode OCR
 - 1) Gagal



2) Berhasil



PENUTUP

Kesimpulan

Berdasarkan hasil uji dapat disimpulkan bahwa dengan menggunakan metode CNN-YOLO v8 plat nomor belum dapat diketahui karakternya dalam bentuk text, hal tersebut dapat terjadi karena hanya dilakukan training data dengan epochs kecil yaitu 5. Sedangkan dengan menggunakan metode OCR dengan tesseract-OCR, hasil dari deteksi plat nomor sudah dapat diketahui dan hasilnya dapat dikeluarkan dalam bentuk alphabet dan numerik. Namun dengan menggunakan metode OCR tidak dilakukan training data sehingga hanya plat nomor yang diambil citranya dengan jelas yang dapat dideteksi dengan baik, serta masih ditemukan beberapa galat pada UI nya.

DAFTAR PUSTAKA

- Alfan Qadar, M., Hidayatno, A., Alvin Adi S., Y., 2020. APLIKASI PENDETEKSI KOTAK UNTUK MENENTUKAN UKURAN TUBUH MENGGUNAKAN ALGORITMA HSV, TRANSIENT.
- Armala, Y., Yasir, M., 2022. IMPLEMENTASI ELECTRONIC TRAFFIC LAW ENFORCEMENT (ETLE) DI WILAYAH HUKUM KEPOLISIAN RESOR BOJONEGORO. JUSTITIABLE 5, 32–44.
- Gazali, W., Soeparno, H., Ohliati, ; Jenny, n.d. PENERAPAN METODE KONVOLUSI DALAM PENGOLAHAN CITRA DIGITAL.
- Hanif, A.R., Nasrullah, E., Setyawan, F.X.A., 2023. DETEKSI KARAKTER PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN METODE OPTICAL CHARACTER RECOGNITION (OCR). Jurnal Informatika dan Teknik Elektro Terapan 11. <https://doi.org/10.23960/jitet.v11i1.2897>
- Hasil Penelitian dan Pembahasan, n.d.
- Pradeska Atmaja, Y., 2021. Sistem Penghitung Jumlah Pengunjung di Restoran Menggunakan Kamera Berbasis Single Shoot Detector (SSD).
- Putra Wardana, A., Nugraha, A., 2023. EFEKTIFITAS PENERAPAN PROGAM E-TILANG DALAM MENINGKATKAN KESADARAN HUKUM MASYARAKAT DALAM BERLALU LINTAS DI KOTA KLATEN. Jurnal Madani Hukum 1, 72–80.
- Ratna, S., 2020. PENGOLAHAN CITRA DIGITAL DAN HISTOGRAM DENGAN PHYTON DAN TEXT EDITOR PHYCHARM, Technologia.
- Satya, L., Septian, M.R.D., Sarjono, M.W., Cahyanti, M., Swedia, E.R., 2023. SISTEM PENDETEKSI PLAT NOMOR POLISI KENDARAAN DENGAN ARSITEKTUR YOLOV8. Sebatik 27, 753–761. <https://doi.org/10.46984/sebatik.v27i2.2374>
- Suhartono, Gunawan Zain, S., Sugiawan, 2022. Sistem Object Recognition Plat Nomor Kendaraan Untuk Sistem Parkir Bandara.

LINK G-DRIVE:

https://drive.google.com/drive/folders/1o1_leNdMPFUx2w8G2rtL0eqjDJ2WWTFg?usp=sharing

LINK GITHUB:

<https://github.com/misternobody24/deteksi--plat-nomor>

LAMPIRAN

Tabel Kontribusi Kelompok

No.	Nama Anggota	Presentase Kontribusi	Kontribusi
1.	Divangga Revansa Arya Pradhana	40%	Membuat program deteksi plat nomor berbasis Tesseract-OCR dengan data lokal, Menyusun PPT, dan Menyusun Laporan
2.	Raditya Koesyan Dipo Pramukti	30%	Membuat program deteksi plat nomor berbasis CNN-YOLOv8
3.	Bagas Muttaqi	30%	Mengumpulkan dataset dan menyusun laporan