

**Jurnal Politeknik Caltex Riau**Terbit Online pada laman <https://jurnal.pcr.ac.id/index.php/jkt/>

| e- ISSN : 2460-5255 (Online) | p- ISSN : 2443-4159 (Print) |

## Implementasi OCR dengan Metode Autoencoder untuk verifikasi data KTP

**Muhamad Aldi Rizaldi<sup>1</sup>, Emil R. Kaburuan<sup>2</sup>**

<sup>1,2</sup> Universitas Mercu Buana Jakarta, Teknik Informatika, Fakultas Ilmu Komputer  
email: <sup>1</sup>41518110189@student.mercubuana.ac.id, <sup>2</sup>emil.kaburuan@mercubuana.ac.id

### [1] Abstrak

*KTP adalah kartu identitas resmi yang diakui negara Indonesia sebagai bukti kependudukan. kartu tersebut dapat digunakan untuk berbagai hal seperti membuat kartu rekening, membuat pasport, membayar pajak dll. Diera teknologi informasi yang berkembang saat ini, ktp juga banyak digunakan untuk keperluan mendaftar akun website atau aplikasi yang akan digunakan dalam melakukan transaksi keuangan atau hanya sekedar memverifikasi identitas asli pengguna tersebut dengan cara mengupload kartu KTP lalu dilakukan verifikasi secara manual oleh petugas. sebagian website sudah menerapkan sistem verifikasi otomatis dengan menerapkan OCR (Optical Character recognition) pada gambar kartu KTP. Untuk mendapatkan hasil OCR yang baik diperlukan gambar dengan tulisan yang jelas, posisinya yang tidak miring dan ukurannya yang tidak terlalu kecil. Oleh karena itu diperlukan berbagai tahap preprocessing sebelum melakukan proses OCR, salah satunya dengan menerapkan denoise menggunakan metode Autoencoder agar gambar menjadi lebih bersih dan hasil OCR menjadi lebih akurat.*

**Kata kunci:** *KTP, Autoencoder, optical character recognition*

### [2] Abstract

*KTP is an official identity card recognized by the Indonesian state as proof of residence. the card can be used for various things such as making bank account, making passports, paying taxes etc. In the current era of developing information technology, KTP is also widely used for the purposes of registering website or application accounts that will be used in conducting financial transactions or just verifying the original identity of the user by uploading an ID card and then verifying it manually by the officer. some websites have implemented an automatic verification system by applying OCR (Optical Character recognition) on the image of the ID card. To get a good OCR result, an image with clear words is needed, its position is not tilted and its size is not too small. Therefore, various preprocessing steps are needed before performing the OCR process, one of which is by applying denoise using the Autoencoder method so that the image becomes cleaner and the OCR results become more accurate.*

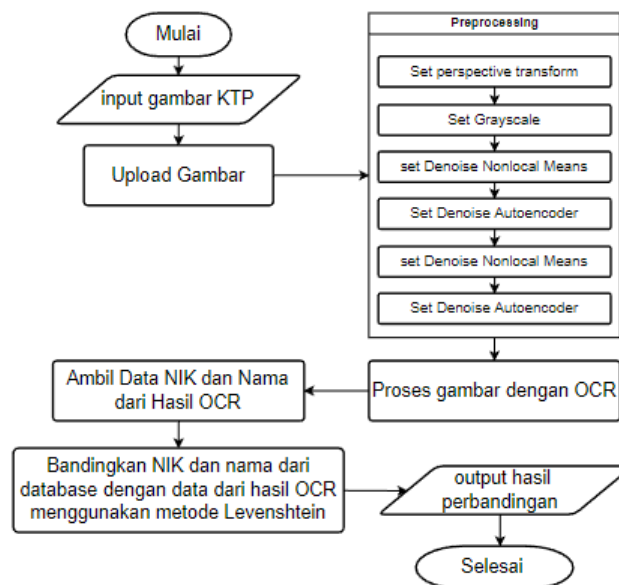
**Keywords:** *ID Card, Autoencoder, optical character recognition*

## 1. Pendahuluan

KTP adalah kartu identitas resmi yang diakui negara Indonesia sebagai bukti kependudukan, pada kartu ini tercantum data identitas seperti nama, tanggal lahir, tempat tinggal dll[1]. Kartu ini sering digunakan untuk verifikasi data pengguna saat mendaftar akun website yang didalamnya terdapat transaksi keuangan atau hanya sekedar memverifikasi identitas pengguna diwebsite media sosial. sebagian website masih melakukan verifikasi data secara manual sehingga untuk mendapatkan hasilnya harus menunggu giliran tergantung jumlah permintaan verifikasi. Maka dari itu diperlukan cara untuk memanfaatkan kartu KTP agar dapat dilakukan verifikasi data secara otomatis yaitu dengan mengambil gambar KTP tersebut dan dilakukan proses Optical Character Recognition (OCR) untuk mendapatkan teks yang tercantum di kartu KTP. OCR sendiri merupakan proses mengubah gambar menjadi karakter ASCII yang dikenali oleh komputer[2]. Untuk mendapat hasil yang bagus, diperlukan berbagai tahap preprocessing seperti grayscale, perspective transform dan denoise. Pada penelitian ini menggunakan 2 jenis proses denoise yang berbeda yaitu Non-local Means dan Autoencoder.

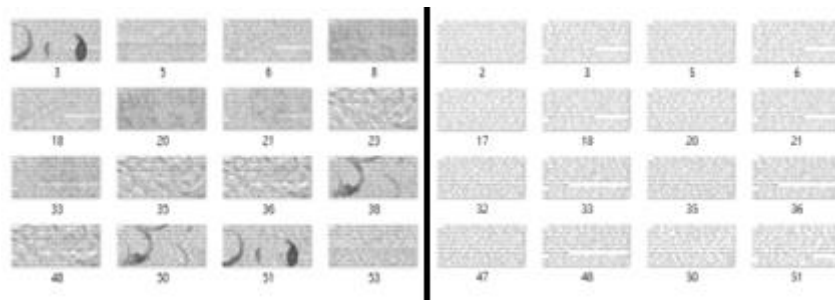
## 2. Metode Penelitian

Teknik pengumpulan data yang digunakan untuk penelitian ini adalah Studi dokumentasi yang dimana Menurut Sugiyono pengertian Studi dokumentasi merupakan suatu teknik pengumpulan data dengan cara mempelajari dokumen untuk mendapatkan data atau informasi yang berhubungan dengan masalah yang diteliti[3]. Pada penelitian ini dibutuhkan sejumlah dokumen berupa kartu E-KTP sebagai bahan pengujian.



Gambar 1. Alur Penelitian

Sebelum memulai proses OCR KTP perlu ada training untuk model Autoencoder agar dapat melakukan denoise gambar dengan baik. Autoencoder merupakan salah satu Artificial Neural Network (ANN) yang berguna untuk melakukan reconstruct pada input gambar seperti mengurangi ukuran gambar, menghilangkan noise[4] dan mendeteksi outline yang berguna untuk kebutuhan proses Natural Language Processing (NLP). Autoencoder merupakan metode yang ditraining secara unsupervised[5] walaupun secara teknis pada penelitian ini ditraining secara supervised. Proses training model Autoencoder menggunakan dataset yang bisa didapatkan dari UCI Machine Learning Repository[6]. Dataset tersebut berisi 144 gambar training dan 144 gambar testing. Semua gambar training dan testing memiliki ukuran panjang 540 pixel dengan lebar bervariasi tetapi tidak lebih dari 420 pixel.



Gambar 2. Dataset training

.Untuk konfigurasi layer model dari Autoencoder yang digunakan pada penelitian ini terdiri dari proses encoding yaitu layer konvolusi 2 dimensi sebanyak 2x, batch normalization, max pooling layer dan dropout lalu dilanjut proses decoding yaitu konvolusi 2 dimensi sebanyak 2x, batch normalization dan up sampling 2d. Model ini menggunakan adam optimizer dan loss mean squared error. Dibawah ini merupakan ringkasan model Autoencode yang dibuat.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 420, 540, 1)]	0
conv2d (Conv2D)	(None, 420, 540, 64)	640
conv2d_1 (Conv2D)	(None, 420, 540, 128)	73856
batch_normalization (Batch Normalization)	(None, 420, 540, 128)	512
max_pooling2d (MaxPooling2D)	(None, 210, 270, 128)	0
dropout (Dropout)	(None, 210, 270, 128)	0
conv2d_2 (Conv2D)	(None, 210, 270, 128)	147584
conv2d_3 (Conv2D)	(None, 210, 270, 64)	73792
batch_normalization_1 (Batch Normalization)	(None, 210, 270, 64)	256
up_sampling2d (UpSampling2D)	(None, 420, 540, 64)	0
conv2d_4 (Conv2D)	(None, 420, 540, 1)	577
Total params: 297,217		
Trainable params: 296,833		
Non-trainable params: 384		

Gambar 3. Ringkasan model Autoencoder

Proses training model dilakukan sebanyak 600x untuk mendapatkan hasil yang baik, dan batch size = 17 tergantung RAM yang akan digunakan. Proses training model Autoencoder dipenelitian ini dilakukan menggunakan Google Colaboratory selama 1 jam dan 24 menit.

```
# melakukan proses training sebanyak 600 kali dengan ukuran batch sebanyak 17 sesuai kapasitas RAM yang digunakan untuk proses training
callback = EarlyStopping(monitor='loss', patience=30)
history = model.fit(X_train, Y_train, validation_data = (X_val, Y_val), epochs=600, batch_size=17, verbose=0, callbacks=[TqdmCallback(ve
```

100% 600/600 [1:24:25<00:00, 8.44s/epoch, loss=0.000736, mae=0.0126, val\_loss=0.000383, val\_mae=0.0095]

100% 8.00/8.00 [00:37<00:00, 2.12s/batch, loss=0.141, mae=0.275]

WARNING:tensorflow:Callback method 'on\_train\_batch\_end' is slow compared to the batch time (batch time: 0.3562s vs 'on\_train\_batch\_end'

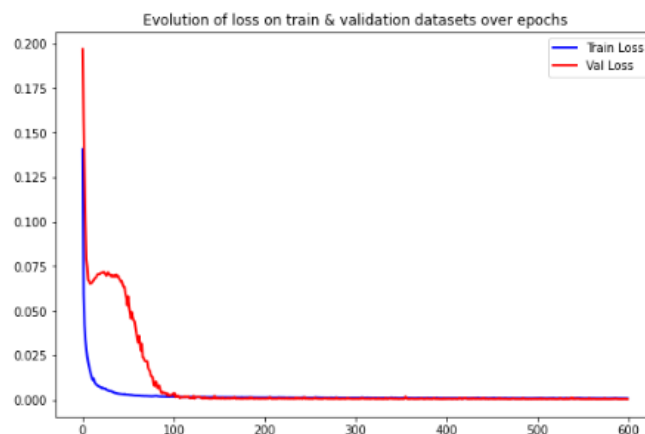
100% 8.00/8.00 [00:07<00:00, 1.36batch/s, loss=0.0592, mae=0.17]

100% 8.00/8.00 [00:07<00:00, 1.36batch/s, loss=0.0422, mae=0.144]

100% 8.00/8.00 [00:07<00:00, 1.34batch/s, loss=0.0339, mae=0.126]

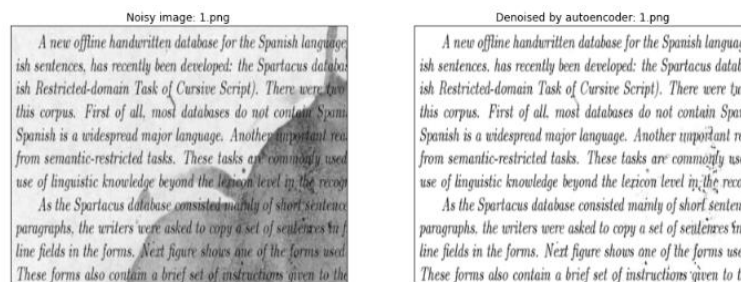
**Gambar 4. Proses training model Autoencoder**

Dibawah ini merupakan statistik loss selama training berjalan, dapat dilihat sempat ada kenaikan error disekitar epochs ke 10, tapi setelah itu terus berkurang mendekati angka 0. Proses training berakhir dengan nilai loss = 0.000736.



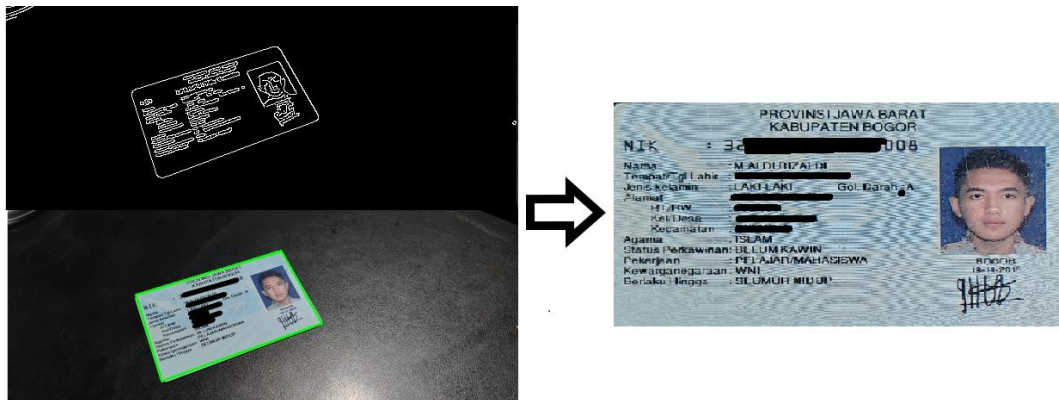
**Gambar 5. Statistik loss selama proses training**

Setelah dilakukan training, kita bisa langsung mencoba melakukan proses denoise, gambar dibawah ini adalah contoh sebelum dan sesudah gambar dilakukan proses denoise. Hasilnya model Autoencoder dapat menghilangkan noise background dengan baik dan dapat diimplementasikan untuk preprocessing OCR menggantikan proses Binarization.



**Gambar 6. Sebelum dan sesudah proses denoise Autoencoder**

Setelah mendapatkan model Autoencoder yang sudah training, kita bisa melanjutkan penelitian untuk memproses OCR KTP. Proses dimulai dari mengupload gambar KTP ke server website untuk diolah, setelah itu gambar KTP akan dilakukan proses preprocessing menggunakan API python. preprocessing yang pertama dilakukan adalah dengan menerapkan perspective transform pada gambar. Dalam prosesnya dilakukan grayscaling, blur dan edge detection untuk menemukan 4 garis border KTP untuk diterapkan sebagai parameter fungsi four point perspective.



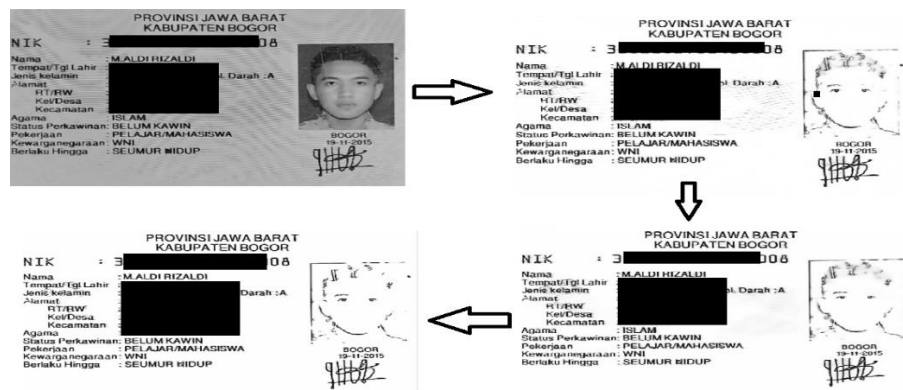
Gambar 7. Proses menerapkan perspective transform pada gambar

Proses perspective transform membantu merubah gambar menjadi tampilan bird view[7]. proses ini sangat penting agar mendapatkan hasil OCR yang baik, Langkah selanjutnya adalah merubah gambar tersebut menjadi hitam putih atau grayscaling..



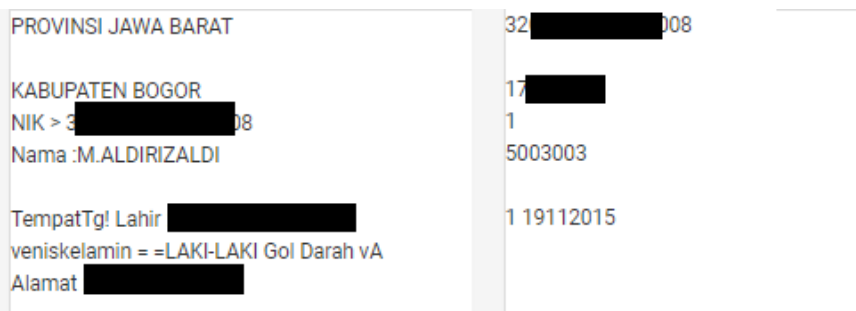
Gambar 8. Sebelum dan sesudah proses grayscaling gambar

Pada tahap ini biasanya dilakukan proses threshold pada gambar. Tetapi pada penelitian ini dilakukan proses denoise non local means dan denoise Autoencoder sebanyak 2x untuk mendapatkan hasil yang lebih maksimal. Denoise non local means merupakan fungsi library python untuk denoise gambar dengan cara mencari nilai rata rata dari warna pixel sejenis walaupun jaraknya berjauhan[8]. disekitarnya.yang siap pakai sedangkan denoise Autoencoder merupakan model Artificial Neural Network (ANN) yang sudah ditraining sebelumnya.



Gambar 9. Proses denoise gambar menggunakan non local means dan Autoencoder

Setelah selesai melakukan preprocessing gambar, sekarang saatnya melakukan proses OCR menggunakan tesseract sebanyak dua kali. Untuk OCR kedua menggunakan pengaturan khusus agar tesseract hanya membaca teks berupa angka untuk keperluan mencari data NIK. Dibawah ini adalah hasil OCR dari gambar KTP



Gambar 10. Hasil OCR dengan pengaturan biasa dan Digits only

Setelah mendapatkan hasil OCR, saatnya mencari data nama dan NIK untuk keperluan verifikasi. Pencarian data nama dilakukan dengan cara melakukan loop dari setiap baris yang ada, lalu dari setiap baris tersebut dicari kata yang mengandung kata "nama" dengan bantuan fungsi Levenshtein. Levenshtein sendiri merupakan algoritma yang sering digunakan dalam mendeteksi kemiripan antara dua string yang sering digunakan untuk mencari potensi tindakan plagiarisme[9]. Untuk mencari nomor NIK cukup dengan mencari baris yang mengandung angka terbanyak.



Gambar 11. Mendapatkan nama dan NIK dari hasil OCR

Pada tahap ini dilakukan proses membandingkan data user dengan teks hasil OCR. Dengan cara menggabungkan data nik dan nama dalam satu variable lalu perbandingan dilakukan dengan menggunakan algoritma levenshtein sama seperti saat mencari data nama diatas.

```
//gabungkan data nik dan nama
$nik_nama_db = $_POST['nik'].$_POST['nama'];
$nik_nama_ktp = $output['nik'].$output['nama'];
//hitung perbedaan dengan fungsi levenshtein
$levenshtein = levenshtein($nik_nama_db,$nik_nama_ktp);
//hitung presentase perbedaan antara dua string tersebut
$jml_txt = strlen($nik_nama_db);
if($levenshtein > 0){
    $output['score'] = (($jml_txt - $levenshtein)/$jml_txt)*100;
}else{
    $output['score'] = 100;
}
```

### 3. Hasil Penelitian dan Pembahasan

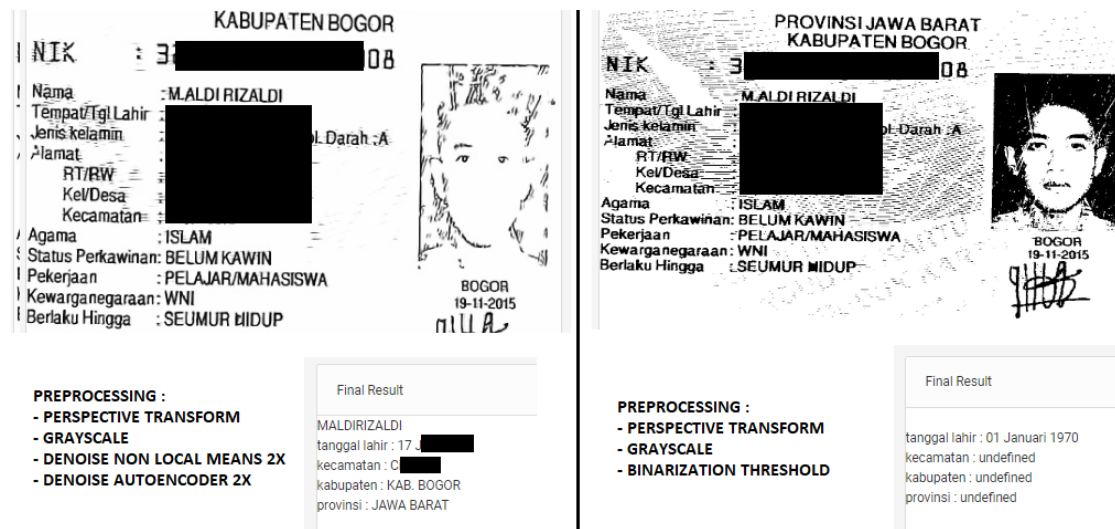
Gambar dibawah adalah hasil tingkat kemiripan antara data dari database dengan data dari hasil scan OCR kartu KTP. data tersebut menunjukkan bahwa hasil kemiripan tidak menunjukkan 100% karena ada beberapa error seperti nama depan dan belakang tidak ada pemisah spasi karena teks yang tercantum diKTP terlalu berdekatan. Adapula jika data didatabase dibandingkan dengan kartu ktp yang berbeda hasilnya tidak menunjukkan 0% karena ada sedikit kemiripan pada nama dan nomor NIK.



Gambar 12. Hasil akhir pembacaan data KTP



Sekarang kita coba bandingkan hasil OCR dengan proses preprocessing menggunakan denoise Autoencoder dibandingkan dengan binarization threshold.



Gambar 13. Perbandingan preprocessing Autoencoder dengan binarization threshold

#### 4. Kesimpulan

Dari hasil penelitian diatas dapat disimpulkan bahwa metode preprocessing yang digunakan menghasilkan gambar yang bersih dan memberikan hasil OCR yang cukup memuaskan, terutama jika dibandingkan dengan metode preprocessing biasa yang hanya mengandalkan binarization threshold karena masih terdapat banyak noise yang mengganggu proses OCR. Tetapi ada kemungkinan jika kedua metode tersebut dikombinasikan akan memberikan hasil yang lebih baik lagi. Hasil penelitian masih terdapat kekurangan dimana ada beberapa error seperti nama depan dan belakang tidak ada pemisah spasi karena teks yang tercantum digambar KTP terlalu berdekatan. algoritma levenshtein sangat membantu dalam mencari kata kunci dalam proses membersihkan teks OCR dan juga saat melakukan perbandingan antara data dari database dan hasil OCR kartu KTP.



**Daftar Pustaka**

- [1] F. M. Rusli, K. A. Adhiguna, and H. Irawan, "Indonesian ID Card Extractor Using Optical Character Recognition and Natural Language Post-Processing," 2021. doi: 10.1109/ICoICT52021.2021.9527510.
- [2] A. Patil, "Optical Character Recognition Implementation using Pattern Matching," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, no. 8, 2019, doi: 10.22214/ijraset.2019.8155.
- [3] Arikunto *et al.*, "Metode Penelitian Kuantitatif, Kualitatif, dan R&D," *J. Sci. Teacher Educ.*, vol. 3, no. 2, 2017.
- [4] M. Mehdizadeh, C. MacNish, D. Xiao, D. Alonso-Caneiro, J. Kugelman, and M. Bennamoun, "Deep feature loss to denoise OCT images using deep neural networks," *J. Biomed. Opt.*, vol. 26, no. 04, Apr. 2021, doi: 10.1117/1.jbo.26.4.046003.
- [5] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nat. Commun.*, vol. 10, no. 1, 2019, doi: 10.1038/s41467-018-07931-2.
- [6] F. Zamora-Martinez, S. España-Boquera, and M. J. Castro-Bleda, "Behaviour-based Clustering of Neural Networks applied to Document Enhancement, in: Computational and Ambient Intelligence, pages 144-151, Springer, 2007." <https://archive.ics.uci.edu/ml/datasets/NoisyOffice> (accessed Jul. 06, 2022).
- [7] R. C. Gerum *et al.*, "CameraTransform: A Python package for perspective corrections and image mapping," *SoftwareX*, vol. 10, 2019, doi: 10.1016/j.softx.2019.100333.
- [8] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005, vol. II. doi: 10.1109/CVPR.2005.38.
- [9] B. P. Pratama and S. A. Pamungkas, "Analisis Kinerja Algoritma Levenshtein Distance dalam Mendeteksi Kemiripan Dokumen Teks," *J. "LOG!K@,"* vol. 6, no. 2, 2016.