

Perbaikan dan Peningkatan Kualitas Citra Menggunakan CNN

M Syahrul Surya Putra / 13520161

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13520161@std.stei.itb.ac.id

Abstract—CNN merupakan alat yang sangat mahir dalam proses pengolahan citra, dimana *use-case* yang paling sering digunakan adalah pengkategorian citra berdasarkan tipe dari citra tersebut. Dalam makalah ini akan dicoba untuk menggunakan CNN dalam perbaikan dan peningkatan kualitas citra. Fokus utama terletak pada penjelasan implementasi CNN dan hasil eksperimen menggunakan dataset relevan, memberikan pemahaman lebih mendalam mengenai efektivitas CNN dalam mengatasi pemrosesan citra

Kata-kunci; CNN, citra, noise

I. PENDAHULUAN

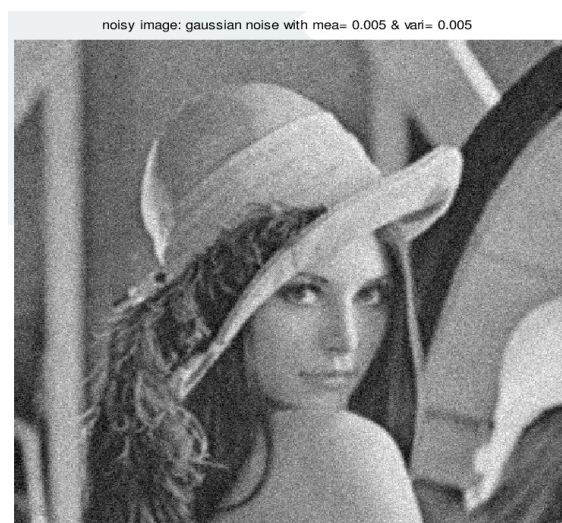
Dalam era digitalisasi yang berkembang pesat, pengolahan citra menjadi salah satu aspek utama yang mendapat perhatian besar dalam berbagai bidang, seperti pengenalan pola, visi komputer, dan pemrosesan gambar medis. Peningkatan kualitas citra menjadi hal krusial dalam mendukung berbagai aplikasi tersebut. Kualitas citra yang baik tidak hanya diperlukan untuk mendukung penelitian ilmiah, tetapi juga penting dalam penerapan teknologi di berbagai sektor industri.

Pentingnya kualitas citra melibatkan sejumlah tantangan, seperti noise, deformasi, dan ketidakjelasan. Solusi untuk mengatasi tantangan tersebut semakin berkembang seiring dengan kemajuan teknologi, dan salah satu metode yang paling menonjol adalah Convolutional Neural Network (CNN). CNN merupakan bentuk khusus dari jaringan saraf dalam mata yang sangat efektif dalam menangani masalah pengolahan citra, terutama karena kemampuannya dalam mengekstrak fitur-fitur yang kompleks.

Dalam konteks ini, makalah ini akan membahas secara mendalam tentang perbaikan dan peningkatan kualitas citra menggunakan CNN. Akan dieksplorasi konsep dasar CNN, memahami cara kerjanya dalam konteks pengolahan citra, dan meninjau berbagai teknik yang telah dikembangkan untuk peningkatan kualitas citra. Selain itu, akan dipertimbangkan juga aspek-aspek seperti efisiensi komputasional, kecepatan konvergensi, dan interpretabilitas hasil.

Kualitas citra yang rendah seringkali menjadi tantangan serius dalam berbagai konteks pengolahan citra. Beberapa faktor yang dapat menyebabkan penurunan kualitas citra melibatkan berbagai aspek teknis dan lingkungan. Salah satu faktor utama yang dapat menyebabkan citra memiliki kualitas

yang rendah adalah keberadaan noise, yang dapat berasal dari sumber-sumber seperti gangguan sensor, transmisi sinyal yang tidak stabil atau kondisi lingkungan yang tidak terkendali. Berikut adalah contoh dari citra yang mengandung noise.



Gambar 1. Contoh Citra yang Mengandung Noise

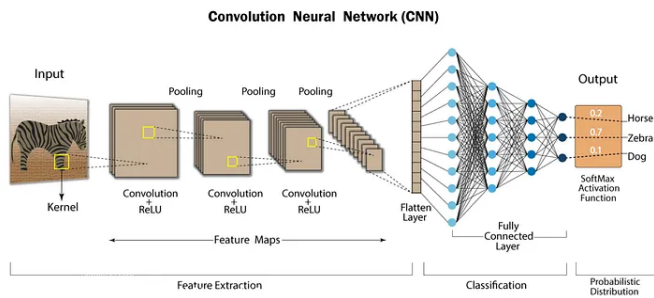
Selain itu, citra dapat mengalami degradasi akibat kompresi data yang tinggi, yang seringkali memang diperlukan dalam penyimpanan ataupun transmisi citra digital. Proses kompresi ini dapat mengakibatkan hilangnya informasi penting, sehingga menurunkan kualitas citra secara signifikan. Selain faktor teknis, aspek lingkungan seperti pencahayaan yang buruk atau ketidakstabilan objek yang difoto juga dapat berkontribusi pada kualitas citra yang rendah.

Dengan menyadari kompleksitas faktor-faktor yang dapat mempengaruhi kualitas citra, implementasi solusi yang tepat menjadi krusial dalam upaya perbaikan dan peningkatan kualitas citra. Sebagai tanggapan terhadap tantangan ini, dalam makalah ini akan dicoba mengusulkan penerapan CNN sebagai pendekatan utama. CNN telah terbukti sangat efektif dalam menangani masalah pengolahan citra. Dengan mengadopsi CNN, diharapkan dapat ditemukan solusi inovatif yang tidak hanya merespons permasalahan kualitas citra yang ada, tetapi juga memberikan kontribusi nyata terhadap kemajuan dalam bidang pengolahan citra secara menyeluruh

II. LANDASAN TEORI

A. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk memproses dan menganalisis data spasial, seperti citra. CNN terinspirasi oleh cara manusia menginterpretasikan informasi visual dan telah membuktikan keunggulannya dalam berbagai tugas pengolahan citra. Ada beberapa komponen kunci dalam struktur CNN yang memungkinkannya untuk secara efektif mengekstrak fitur dari data citra, yaitu Convolutional Layer, Rectified Linear Unit (ReLU), Pooling Layer, dan Fully-Connected Layer.

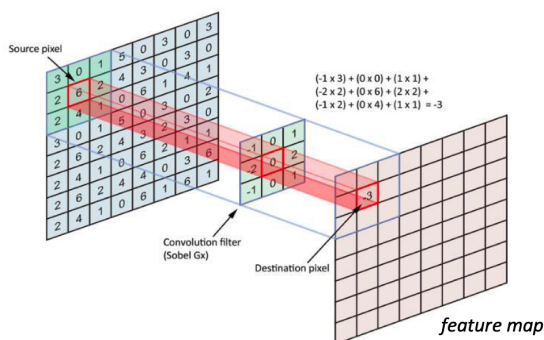


Gambar 2. Arsitektur CNN

Dengan gabungan komponen-komponen tersebut, CNN dapat menggambarkan hierarki kompleks fitur-fitur pada citra, memungkinkan aplikasi yang luas dalam pengenalan objek, analisis citra medis, dan berbagai tugas pengolahan citra. Keberhasilan CNN dalam memahami pola spasial dan hierarki fitur membuatnya menjadi alat yang efektif dalam mewujudkan potensi aplikatif pada domain pengolahan citra modern.

1) Convolutional Layer

Convolutional Layer adalah komponen kunci dalam arsitektur CNN yang berfungsi untuk mengekstrak fitur-fitur lokal dari citra input. Layer ini menerapkan operasi konvolusi pada citra menggunakan filter atau kernel untuk mendeteksi pola visual dan atribut penting. Proses konvolusi ini membantu CNN mengidentifikasi fitur-fitur lokal seperti tepi, sudut, atau tekstur dalam citra, sehingga menciptakan representasi yang lebih abstrak dan hierarkis dari data visual.



Berikut adalah beberapa elemen utama dalam Convolutional Layer:

a) Filter/Kernel:

Filter atau kernel adalah matrik bobot yang digunakan dalam operasi konvolusi. Setiap filter berfungsi sebagai pemetaan kecil yang bergerak melintasi citra. Setiap nilai dalam filter menunjukkan bobot yang diberikan pada piksel terkait dalam citra. Filter ini akan mengekstrak fitur-fitur tertentu dari citra, seperti edge, corner, atau tekstur, tergantung pada bobot yang ditentukan selama proses pelatihan.

b) Stride:

Stride adalah langkah atau jarak yang diterapkan filter ketika bergerak melintasi citra. Stride menentukan seberapa banyak filter akan bergeser setiap kali operasi konvolusi diaplikasikan. Stride yang lebih besar menghasilkan peta fitur yang lebih kecil, sementara stride yang lebih kecil dapat menghasilkan peta fitur yang lebih besar dengan lebih banyak informasi spasial.

c) Padding:

Padding adalah penambahan nilai nol di sekitar citra sebelum proses konvolusi. Ini dilakukan untuk mempertahankan informasi di tepi citra dan mengurangi efek perubahan ukuran selama beberapa lapisan konvolusi. Padding membantu mencegah pengurangan dimensi citra yang terlalu cepat.

d) Feature Map:

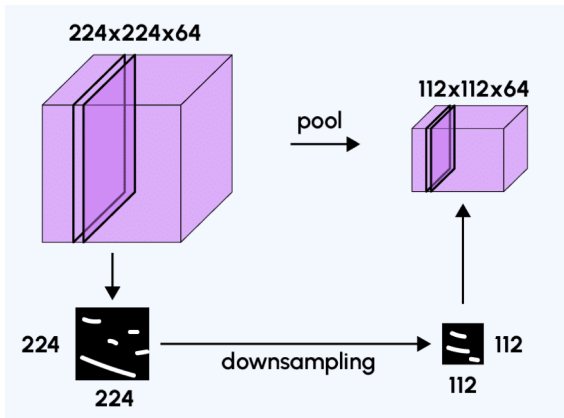
Hasil dari operasi konvolusi disebut peta fitur atau feature map. Peta fitur adalah representasi hasil konvolusi dari citra input menggunakan filter. Setiap nilai dalam peta fitur menunjukkan tingkat kehadiran fitur tertentu dalam citra.

e) ReLu:

Setelah operasi konvolusi, fungsi aktivasi ReLu seringkali diterapkan. ReLu (Rectified Linear Unit) mengubah setiap nilai negatif dalam peta fitur menjadi nol, mengenalkan unsur non-linearitas ke dalam CNN. Hal ini memungkinkan jaringan untuk mempelajari representasi yang lebih kompleks dan abstrak dari data.

2) Pooling Layer

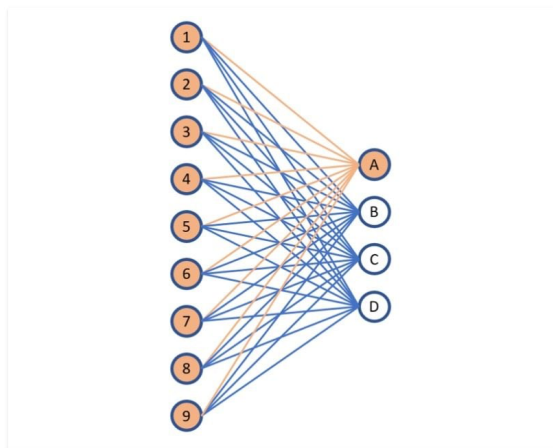
Pooling Layer adalah komponen kritis dalam CNN yang berfungsi untuk mengurangi dimensi spasial dari peta fitur yang dihasilkan oleh Convolutional Layer. Layer ini bertujuan untuk mempertahankan informasi esensial dari fitur-fitur yang telah diekstrak oleh Convolutional Layer, sambil mengurangi jumlah parameter dan beban komputasi. Jenis pooling yang paling umum digunakan adalah Max Pooling.



Gambar 4. Pooling Layer

3) Fully-Connected Layer

Fully-Connected Layer adalah salah satu komponen penting dalam arsitektur CNN, yang biasanya terletak di ujung jaringan setelah melalui serangkaian lapisan konvolusi dan pooling. Lapisan ini bertanggung jawab untuk menghubungkan setiap neuron atau unit aktivasi dalam lapisan sebelumnya dengan setiap neuron di lapisan fully-connected, menciptakan representasi global dari fitur-fitur lokal yang telah diekstrak.



Gambar 5. Fully-Connected Layer

Pentingnya Fully-Connected Layer ini terletak pada kemampuannya untuk menyusun dan mengintegrasikan informasi lokal menjadi representasi yang lebih abstrak, yang dapat diinterpretasikan untuk pengambilan keputusan akhir. Meskipun Fully-Connected Layer sering digunakan dalam tugas klasifikasi, arsitektur CNN juga dapat disusun tanpa Fully-Connected Layer tergantung pada tugas dan kebutuhan spesifik aplikasi.

B. Citra

Citra adalah representasi visual dua dimensi dari suatu objek atau scene yang direkam melalui media seperti kamera atau alat pencitraan lainnya. Citra dapat merepresentasikan berbagai jenis informasi visual, termasuk gambar, grafik, atau

peta. Dalam konteks umum, istilah “citra” seringkali mengacu pada gambar yang dihasilkan oleh perangkat pengambil gambar atau alat pencitraan digital seperti kamera.

Suatu citra terdiri dari gabungan pixel-pixel. Pixel merupakan unit dasar dalam representasi citra digital. Pixel ini mewakili satu titik kecil dalam citra dan memiliki nilai intensitas atau warna tertentu. Kombinasi pixel ini membentuk gambar secara keseluruhan.



Gambar 6. Contoh Citra Hasil Ilustrasi

C. Noise

Dalam konteks pengolahan citra, noise atau kebisingan adalah gangguan acak yang muncul dalam citra dan dapat mengakibatkan perubahan tidak diinginkan pada informasi visual. Kebisingan dapat berasal dari berbagai sumber dan dampaknya dapat meliputi penurunan kualitas citra, pengurangan ketajaman, serta kesulitan dalam identifikasi atau analisis objek. Berikut adalah beberapa bentuk umum dari noise dalam citra:

- 1) **Gaussian Noise**
Gaussian Noise merupakan jenis noise acak yang mengikuti distribusi Gaussian atau distribusi normal. Noise ini seringkali terjadi sebagai variasi acak intensitas piksel dalam citra, menciptakan titik-titik yang tampak acak
- 2) **Salt-and-Pepper Noise**
Salt-and-Pepper noise terjadi ketika beberapa piksel dalam citra mengalami perubahan yang mendadak menjadi nilai intensitas maksimum (salt) atau nilai intensitas minimum (pepper). Noise ini dapat membuat titik-titik hitam dan putih yang tersebar acak
- 3) **Temporal Noise**
Temporal noise terjadi dalam serangkaian gambar atau dalam video. Variasi antar frame dapat menciptakan fluktuasi yang tidak diinginkan dan dapat merusak konsistensi visual antar frame
- 4) **Speckle Noise**

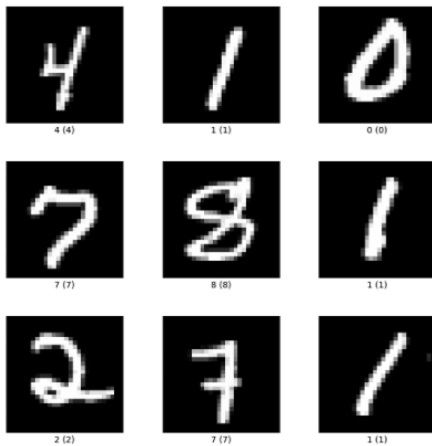
Speckle noise adalah noise yang muncul dalam bentuk butiran-butiran kecil dengan intensitas yang bervariasi. Noise ini seringkali terlihat seperti titik-titik kecil yang tersebar di seluruh citra dan dapat mengaburkan detail

Pengurangan noise menjadi langkah penting dalam pengolahan citra untuk mempertahankan atau meningkatkan kualitas citra. Teknik-teknik pengurangan noise melibatkan penggunaan filter khusus, seperti filter median atau filter Gaussian, yang dapat meredam atau menghilangkan efek noise tanpa mengorbankan detail yang signifikan. Pengelolaan noise sangat penting terutama dalam aplikasi yang memerlukan akurasi tinggi.

III. IMPLEMENTASI

A. Preprocess Dataset

Sebelum dilakukannya *training* dengan model CNN, pertama-tama kita perlu meload dataset yang kita perlukan terlebih dahulu. Dataset yang akan digunakan dalam proses pengujian disini adalah dataset MNIST, dimana MNIST merupakan kumpulan tulisan tangan angka.



Gambar 7. Figure Dataset MNIST

Setelah dataset terload, perlu dilakukan normalisasi terhadap data-data yang ada. Setelah dinormalisasi, kita tambahkan noise pada data sebelumnya. Karena data yang diberikan berkemungkinan tidak ternormalisasi, kita perlu melakukan normalisasi sekali lagi, sehingga value dari pixel yang ada hanya antara 0 dan 1. Dan terakhir, karena data perlu dimasukkan dalam model yang ukuran inputnya pre-determined, kita perlu juga mengubah ukuran dari image sesuai dengan ukuran model yang akan kita buat nanti

Berikut adalah kode implementasi dari penjelasan di atas:

```
# Load MNIST dataset
(x_train, _), (x_test, _) = mnist.load_data()

# Normalisasi
x_train = x_train.astype('float32') / 255.0
```

```
x_test = x_test.astype('float32') / 255.0

# Menambah noise secara manual
noise_factor = 0.5
x_train_noisy = x_train + noise_factor *
np.random.normal(loc=0.0, scale=1.0,
size=x_train.shape)
x_test_noisy = x_test + noise_factor *
np.random.normal(loc=0.0, scale=1.0,
size=x_test.shape)

# Clip pixel
x_train_noisy = np.clip(x_train_noisy, 0., 1.)
x_test_noisy = np.clip(x_test_noisy, 0., 1.)

# Reshape gambar
x_train = np.reshape(x_train, (len(x_train),
28, 28, 1))
x_test = np.reshape(x_test, (len(x_test), 28,
28, 1))
x_train_noisy = np.reshape(x_train_noisy,
(len(x_train_noisy), 28, 28, 1))
x_test_noisy = np.reshape(x_test_noisy,
(len(x_test_noisy), 28, 28, 1))
```

B. Tentukan Model

Setelah data di preprocess, langkah selanjutnya adalah mendefine model CNN itu sendiri. Model CNN ini terdiri dari 2 bagian, encoder dan decoder. Encoder berfungsi untuk mengkompresi citra input, dimana citra input ini sudah diberikan noise pada tahap preprocessing. Tahap kedua adalah decoding, dimana representasi dari input yang telah dikompres dibuat ulang sesuai dengan apa yang telah didapatkan pada tahap encoding.

Masing-masing weight yang terdapat dalam model ini akan disesuaikan dengan output yang akan ditentukan dalam fase training. Weight ini akan selalu terupdate sesuai dengan loss function yang juga akan didefinisikan.

Berikut adalah kode implementasi dari penjelasan di atas:

```
# Encoder
input_img = Input(shape=(28, 28, 1))
x = Conv2D(32, (3, 3), activation='relu',
padding='same')(input_img)
x = MaxPooling2D((2, 2), padding='same')(x)
x = Conv2D(32, (3, 3), activation='relu',
padding='same')(x)
```



```

encoded = MaxPooling2D((2, 2),
padding='same')(x)

# Decoder
x = Conv2D(32, (3, 3), activation='relu',
padding='same')(encoded)
x = UpSampling2D((2, 2))(x)
x = Conv2D(32, (3, 3), activation='relu',
padding='same')(x)
x = UpSampling2D((2, 2))(x)
decoded = Conv2D(1, (3, 3),
activation='sigmoid', padding='same')(x)

autoencoder = Model(input_img, decoded)
autoencoder.compile(optimizer='adam',
loss='binary_crossentropy')

```

C. Training

Dalam fase training ini, ditentukan data input yaitu citra latihan yang telah diberikan noise. Kemudian, output yang diharapkan adalah citra biasa yang belum diberikan noise. Banyak iterasi yang dilakukan (epoch) adalah 5 dengan ukuran batch sebanyak 128.

Berikut adalah kode implementasi dari kode di atas:

```




autoencoder.fit(x_train_noisy, x_train,
epochs=5, batch_size=128, shuffle=True,
validation_data=(x_test_noisy, x_test))
















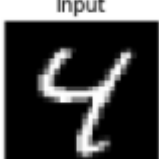
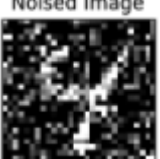
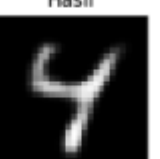
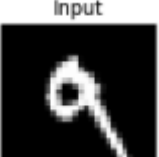
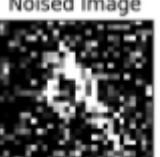
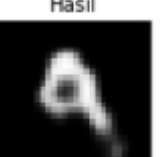
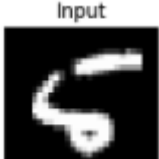
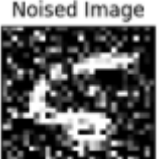
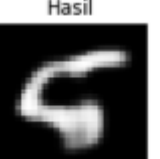
```

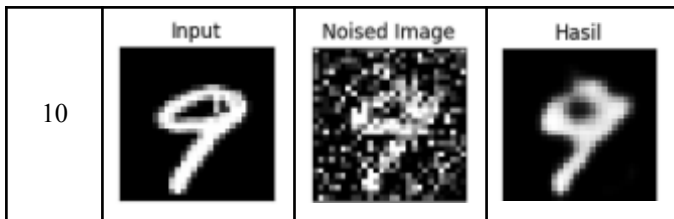
IV. HASIL

Dari hasil training, didapatkan bahwa loss dan validation loss secara berurutan sebesar 0.1019 dan 0.1003, dimana nilai tersebut bisa dibidang relatif kecil. Selain itu, berikut adalah hasil image test menggunakan 10 data dari citra test:

Tabel 1. Hasil Implementasi

No	Citra Input	Citra Noised	Citra Hasil
1			

2			
3			
4			
5			
6			
7			
8			
9			



Dari gambar-gambar di atas, dapat dilihat bahwa citra yang direkonstruksi terbilang tidak berbeda jauh dengan citra inputnya. Namun, setelah mengimplementasikan ini, terdapat batasan yang tidak bisa dihiraukan. Dimana, model ini sangat bergantung dengan dataset yang digunakan dalam tahap pelatihan. Jika kita gunakan, contohnya Gambar 6 yang diberikan noise, maka model ini tidak akan bisa merekonstruksi gambar tersebut menjadi gambar semula sebaik dengan gambar yang memang berasal dari dataset MNIST.

V. KESIMPULAN DAN SARAN

Implementasi perbaikan dan peningkatan kualitas citra menggunakan CNN terbukti bisa dilakukan. Dari hasil yang didapatkan, dimana image input memiliki noise yang relatif sangat besar, image yang dihasilkan terbilang dekat dengan input aslinya. Jadi, dapat disimpulkan bahwa dengan metode deep-learning, dalam kasus ini CNN, kita bisa meningkatkan kualitas ataupun memperbaiki image yang memiliki noise dalam gambarnya.

Adapun saran dari implementasi ini adalah, untuk menggunakan implementasi lain sebagai pembandingan yang konkrit. Selain itu, bisa juga diberikan metode lain selain pemberian noise pada citra agar bisa dicari batas dari penggunaan CNN dalam meningkatkan kualitas dari suatu citra. Selain itu, CNN memiliki keterbatasan, dimana model yang telah dibuat dalam makalah ini, hanya bisa meningkatkan kualitas citra dari dataset MNIST. Berbeda dengan penapis-penapis yang telah diajarkan dalam kelas, yang bisa menapis citra apapun dan langsung menghasilkan hasil yang diharapkan.

UCAPAN TERIMA KASIH

Penulis bersyukur kepada Allah SWT karena berkat-Nya, makalah dengan judul "Perbaikan dan Peningkatan Kualitas Citra Menggunakan CNN" berhasil diselesaikan dengan baik dan tepat dengan jadwal yang telah diberikan. Ucapan terima kasih pun ingin penulis berikan kepada dosen mata kuliah

IF4073 Interpretasi dan Pengolahan Citra, Bapak Dr. Ir. Rinaldi Munir, M.T. yang telah senantiasa memberikan materi dan juga pengetahuan mengenai pemrosesan citra sehingga penulis bisa menyelesaikan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada keluarga dan teman yang senantiasa memberikan dukungan kepada penulis

REFERENCES

- [1] R. Munir, "13 - Restorasi citra (Bagian 1)", Homepage Rinaldi Munir, 2023. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/13-Restorasi-citra-bagian1-2023.pdf>. Accessed: 19 Desember 2023
- [2] R. Munir, "21 - Convolutional Neural Network (CNN)", Homepage Rinaldi Munir, 2023. [Online]. Available: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2023-2024/21-CNN-2023.pdf>. Accessed: 19 Desember 2023
- [3] N. Shahriar, "What is Convolutional Neural Network - CNN (Deep Learning)". [Online]. Available: <https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5>. Accessed: 19 Desember 2023
- [4] "mnist", TensorFlow. [Online]. Available: <https://www.tensorflow.org/datasets/catalog/mnist>. Accessed: 19 Desember 2023

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2023



M Syahrul Surya Putra
13520161