

ARTICLE

Klasifikasi Citra Sintetis Hasil Model Difusi Menggunakan Gray Level Co-Occurrence Matrix (GlcM) Dan Convolutional Neural Network (CNN)

Classification of Synthetic Images Generated by Diffusion Models Using Gray Level Co-Occurrence Matrix (GlcM) and Convolutional Neural Network (CNN)

Andri Hardiyanto* dan Mutaqin Akbar

Informatika, Fakultas Teknologi Informasi, Universitas Mercu Buana Yogyakarta, Yogyakarta, Indonesia

*Penulis Korespondensi: 211110006@student.mercubuana-yogya.ac.id

(Disubmit 03-07-25; Diterima 14-07-25; Dipublikasikan online pada 20-10-25)

Abstrak

Kemajuan teknologi kecerdasan buatan khususnya dalam bidang pengolahan citra telah memungkinkan penciptaan gambar buatan yang sangat menyerupai gambar nyata, sehingga menimbulkan tantangan dalam verifikasi keaslian citra digital. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi citra untuk membedakan antara citra asli dan citra hasil kecerdasan buatan dengan pendekatan hybrid Gray Level Co-occurrence Matrix (GLCM) untuk mengekstraksi enam fitur tekstur dari citra grayscale dengan fitur spasial dari Convolutional Neural Network (CNN) yang kemudian digabungkan untuk membentuk vektor fitur gabungan. Dataset yang digunakan terdiri dari 3.410 citra berwarna yang terbagi secara seimbang ke dalam dua kelas real dan fake. Hasil pengujian CNN murni mencapai akurasi 97%, dengan presisi dan recall antara 0.95-0.99, serta f1-score 0.97. Sementara itu, pada model GLCM-CNN akurasinya mencapai 98% dengan nilai presisi dan recall 0.96-1.00 serta f1-score 0.98. Integrasi fitur tekstur dari GLCM terbukti mampu meningkatkan sensitivitas model terhadap pola mikro pada citra buatan yang tidak dapat ditangkap oleh CNN. Penelitian ini menunjukkan potensi pendekatan hybrid sebagai dasar pengembangan sistem pendeteksi citra sintetis yang adaptif dan akurat di masa mendatang.

Kata kunci: Klasifikasi citra; GLCM; CNN; Citra sintetis; Kecerdasan buatan

Abstract

Advances in artificial intelligence technology, particularly in image processing, have enabled the creation of artificial images that closely resemble real images, thus creating challenges in verifying the authenticity of digital images. This study aims to develop an image classification system to distinguish between original and artificially generated images using a hybrid Gray Level Co-occurrence Matrix (GLCM) approach to extract six texture features from grayscale images with spatial features from a Convolutional Neural Network (CNN) which are then combined to form a combined feature vector. The dataset used consists of 3,410 color images divided equally into two classes, real and fake. The results of the pure CNN test achieved an accuracy of 97%, with precision and recall between 0.95–0.99, and an F1-score of 0.97. Meanwhile, the GLCM-CNN model achieved an accuracy of 98% with precision and recall values of 0.96–1.00 and an F1-score of 0.98. The integration of texture features from GLCM has been shown to increase the model's sensitivity to micro-patterns in artificial images that cannot be captured by CNN. This research demonstrates the potential of a hybrid approach as a basis for developing an adaptive and accurate synthetic image detection system in the future.

KeyWords: Image classification; GLCM; CNN; Synthetic image; Artificial Intelligence

This is an Open Access article - copyright on authors, distributed under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY SA) (<http://creativecommons.org/licenses/by-sa/4.0/>)

How to Cite: A. Hardiyanto *et al.*, "Klasifikasi Citra Sintetis Hasil Model Difusi Menggunakan Gray Level Co-Occurrence Matrix (GlcM) Dan Convolutional Neural Network (CNN)", *JIKO (JURNAL INFORMATIKA DAN KOMPUTER)*, Volume: 9, No.3, Pages 572–582, Juni 2025, doi: 10.26798/jiko.v9i3.2033.

1. Pendahuluan

Citra hasil kecerdasan buatan adalah gambar atau visual yang dihasilkan oleh sistem *Artificial Intelligence* (AI) melalui proses komputasi, baik itu dengan menciptakan gambar baru yang belum pernah ada sebelumnya, memanipulasi gambar asli, maupun melakukan analisis dan pengenalan pola pada citra digital [1]. Kemajuan teknologi generatif seperti *Generative Adversarial Networks* (GAN), *Variational Autoencoders* (VAE), dan model difusi telah memungkinkan penciptaan citra sintetis yang sangat mirip dengan citra asli [2, 3]. Teknologi ini banyak digunakan dalam berbagai aplikasi seperti seni digital, pelatihan data, *Augmented Reality* (AR), hingga konten media sosial [4]. Namun, kemiripan visual yang tinggi antara citra asli dan citra buatan menimbulkan tantangan serius dalam proses verifikasi keaslian, terutama dalam konteks penyebaran informasi palsu dan manipulasi visual.

Berbagai pendekatan telah dikembangkan untuk klasifikasi citra, salah satunya *Gray Level Co-occurrence Matrix* (GLCM) yang mengeksplorasi informasi tekstur mikro dalam citra [5] serta *Convolutional Neural Network* (CNN) yang unggul dalam mengenali pola spasial kompleks [6, 7]. Penelitian sebelumnya seperti dalam [8] dan [9] telah menerapkan kombinasi GLCM dan CNN dalam klasifikasi citra medis dan motif batik menunjukkan akurasi yang tinggi. Sementara itu, penelitian dalam [10] dan [11] menerapkan pendekatan serupa pada citra daun untuk klasifikasi jenis tanaman herbal, serta pada citra telur untuk mendeteksi tingkat keretakan, hasil pengujian menunjukkan peningkatan akurasi klasifikasi. Namun, penelitian yang secara khusus mengkaji efektivitas pendekatan hybrid dalam membedakan antara citra sintetis dan citra asli khususnya dari model difusi masih sangat terbatas. Oleh karena itu, penelitian ini bertujuan untuk menguji kontribusi fitur tekstur GLCM dan fitur spasial dari CNN dalam konteks deteksi citra hasil model difusi, serta mengukur kinerjanya terhadap baseline CNN murni. Penelitian ini diharapkan dapat berkontribusi dalam pengembangan sistem deteksi konten digital berbasis AI (*Artificial Intelligence*) yang lebih akurat dan adaptif terhadap perkembangan teknologi generatif.

2. Metode Penelitian

2.1 Diagram Alur Penelitian

model klasifikasi citra hasil kecerdasan buatan menggunakan GLCM dan CNN. Penelitian ini terdiri dari beberapa tahap utama yang saling terkait dan berurutan sebagai berikut dapat dilihat pada Gambar 1.

2.2 Pengumpulan Data

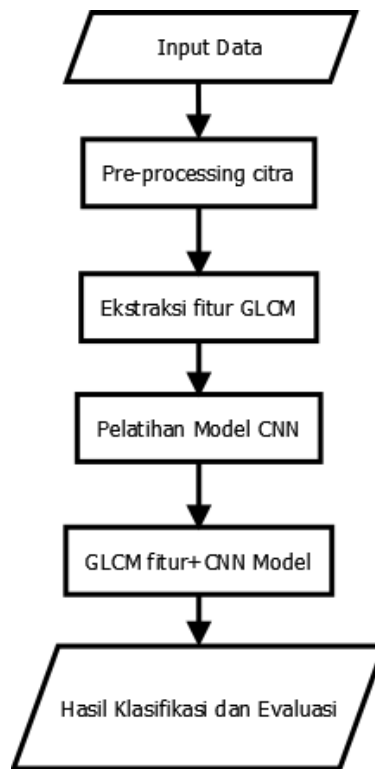
Dataset yang digunakan dalam penelitian ini berasal dari platform *Kaggle* yang terindeks Springer [12], sehingga memiliki kualitas yang dapat dipertanggungjawabkan secara ilmiah. Generator yang dipakai menggunakan model berbasis *latent diffusion*, yaitu pendekatan yang menggunakan VAE dan U-Net. Dataset ini terdiri dari 3.410 citra dengan ekstensi .jpg yang terbagi ke dalam dua kelas, yaitu *real* (merupakan gambar yang diambil dari dunia nyata) dan *fake* (gambar yang dihasilkan oleh model *latent diffusion*). Adapun sampel citra dataset dari kedua label, disajikan pada Gambar ???. Dataset dibagi menjadi tiga subset dengan rasio 80:10:10. 80 sebagai data pelatihan, 10 untuk data validasi, dan 10 untuk data uji. Sehingga terdapat 2.761 data pelatihan, 307 data validasi dan 342 data pengujian yang terbagi ke dalam dua kelas.

2.3 Pre-processing Data

Sebelum citra digunakan dalam pelatihan model, tahapan pertama yang dilakukan adalah *resize* citra, yaitu proses mengubah ukuran menjadi selaras [13]. Semua gambar yang awalnya memiliki dimensi yang beragam diubah ukurannya menjadi 128x128 piksel agar dimensi *input* konstan dan kompatibel dengan model yang digunakan. Langkah selanjutnya yaitu menormalisasikan nilai piksel. Setiap gambar memiliki tiga kanal warna (RGB). Nilai *piksel* pada masing-masing kanal (merah, hijau, biru) dinormalisasikan ke rentang [0, 1] menggunakan rumus (1):

$$I_{norm} = \frac{I}{255} \quad (1)$$

di mana I adalah nilai piksel dalam rentang 0-255 dan adalah nilai piksel yang telah dinormalisasi. Tujuannya adalah untuk mempercepat konvergensi saat proses pelatihan dan menghindari perbedaan skala antar fitur [14]. Setelah normalisasi, citra RGB dikonversi menjadi citra *grayscale*, adalah citra yang hanya



Gambar 1. Tahapan penelitian



Gambar 2. Contoh perbandingan citra asli (a) dan citra sintetis (b)

memiliki satu kanal warna yang merepresentasikan intensitas cahaya (keabuan) dengan rentang nilai 0-255 [13]. Konversi ini penting karena metode GLCM hanya dapat bekerja pada gambar dua dimensi yang menyimpan informasi intensitas piksel. Konversi dilakukan menggunakan rumus (2):

$$I_{gray} = 0,2989R + 0,587G + 0,114B \quad (2)$$

Rumus tersebut merupakan standar konversi *perceptual* yang mempertimbangkan sensitivitas mata manusia terhadap masing-masing kanal warna [10, 14]. *Preprocessing* yang terakhir adalah label *encoding*, di mana label real dan *fake* dikonversi menjadi bentuk numerik integer sehingga format label akhir menjadi 1 untuk real dan 0 untuk *fake*.

2.4 Gray Level Co-occurrence Matrix (GLCM)

Parameter GLCM yang digunakan dalam penelitian ini yaitu arah (Angle) 0° dengan jarak antar piksel (*Distance*) satu piksel. Konfigurasi ini dipilih berdasarkan studi sebelumnya dan eksperimen awal yang menunjukkan bahwa arah horizontal 0° paling konsisten dalam menangkap pola tekstur pada dataset yang digunakan serta menjaga efisiensi komputasi. Citra dinormalisasikan dengan 8-bit *grayscale* (0-255) terlebih dahulu. Fitur tekstur yang diekstraksi dari setiap citra ada enam, yakni *contrast*, *correlation*, *homogeneity*, *ASM*, *dissimilarity*, dan *energy* [15]. Matriks GLCM dihitung menggunakan fungsi *greycomatrix* dari pustaka *skimage.feature*. Enam fitur tersebut dihitung dari matriks menggunakan *greycoprops* yang disimpan dalam bentuk *array* dan digunakan sebagai *input* model CNN. *Contrast* berfungsi untuk mengukur perbedaan intensitas piksel dengan tetangganya pada seluruh citra, seperti rumus (3):

$$Contrast = \sum_{i,j} (i,j)^2 P(i,j) \quad (3)$$

Kemudian *correlation* berfungsi untuk mengukur hubungan linear antara piksel dan tetangganya dengan rumus (4):

$$Correlation = \sum_{i,j} \frac{(i - \mu_i)^2 (j - \mu_j) P(i,j)}{\sigma_i \sigma_j} \quad (4)$$

Sedangkan *homogeneity* berfungsi untuk mengukur kedekatan distribusi elemen dalam GLCM terhadap diagonal GLCM dengan rumus (5):

$$Homogeneity = \sum_{i,j} \frac{P(i,j)}{1 + |i - j|} \quad (5)$$

Kemudian *Angular Second Moment* (ASM) berfungsi untuk mengukur keseragaman atau homogenitas tekstur seperti pada rumus (6):

$$ASM = \sum_{i,j} P(i,j)^2 \quad (6)$$

Dissimilarity berfungsi untuk mengukur perbedaan absolut rata-rata antara pasangan nilai piksel, rumus (7):

$$Dissimilarity = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |i - j| \cdot P(i,j) \quad (7)$$

Terakhir *energy* merupakan akar kuadrat dari ASM seperti terlihat pada rumus (8):

$$Energy = \sqrt{ASM} \quad (8)$$

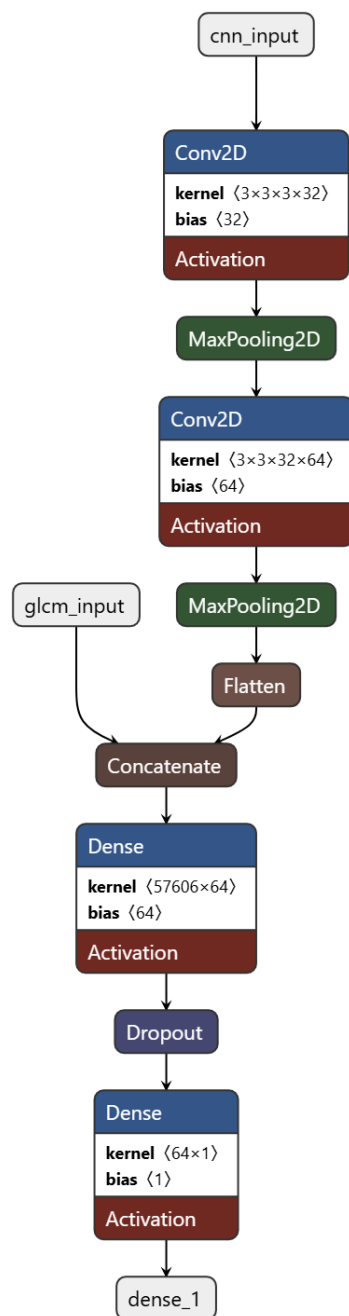
Keterangan :

- i, j : nilai intensitas piksel.
- $P(i, j)$: nilai probabilitas atau frekuensi kemunculan pasangan intensitas.
- N : jumlah level keabuan, misalnya 8, 16, atau 256 tergantung reduksi level gambar.
- μ_i : rata-rata nilai intensitas piksel baris (i).
- μ_j : rata-rata nilai intensitas piksel kolom (j).
- $\sigma_i \sigma_j$: simpangan baku dari masing-masing.

2.5 Convolutional Neural Network (CNN)

Dalam penelitian ini digunakan arsitektur sederhana dengan dua blok konvolusi untuk mengekstraksi fitur spasial dari citra RGB. Setiap blok terdiri dari lapisan Conv2D dengan fungsi aktivasi ReLU, diikuti oleh MaxPooling untuk mereduksi dimensi spasial dan mengurangi risiko *overfitting*. Setelah proses konvolusi dan *pooling*, *output* dilanjutkan ke *Flatten layer* dan *Fully Connected Layer (Dense)* dengan aktivasi *Sigmoid* sebagai *output biner*. Pemilihan arsitektur ini mempertimbangkan efisiensi komputasi, mengingat ukuran dataset yang tergolong menengah, serta kebutuhan generalisasi pada citra sintesis yang memiliki variasi halus. Struktur dua blok konvolusi dianggap cukup untuk menangkap pola spasial dominan seperti kontur dan tekstur global tanpa menyebabkan *overfitting*. Selain itu, aktivasi *Sigmoid* digunakan karena klasifikasi bersifat biner, dengan interpretasi probabilitas pada *output* akhir [10, 16, 17].

2.6 Desain Arsitektur GLCM-CNN



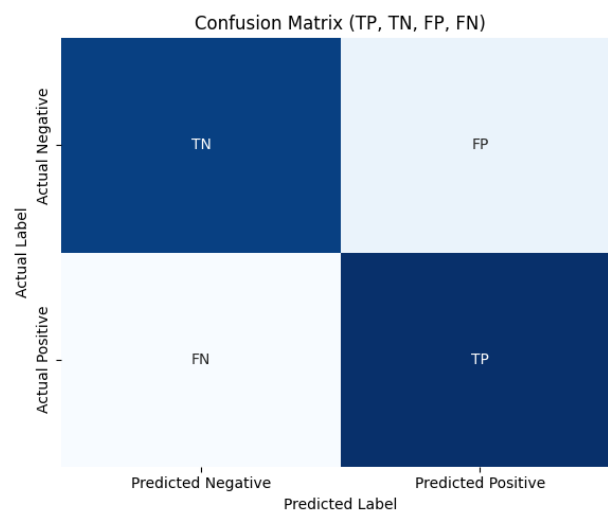
Gambar 3. Desain arsitektur model hybrid

Pada bagian awal, model menerima *input* berupa citra berwarna dengan ukuran 128×128 piksel. Citra ini di proses melalui dua lapisan *Convolutional Layer* (Conv2D) berturut-turut. Lapisan konvolusi pertama menggunakan 32 filter berukuran 3×3 dengan *stride default*, yang berarti filter digeser satu piksel per langkah secara horizontal maupun vertikal. Proses ini menghasilkan 32 buah peta fitur yang menangkap pola lokal dari citra. Setelah konvolusi, hasilnya diteruskan ke fungsi aktivasi ReLU untuk menambahkan *non-linearitas*, sehingga model mampu mempelajari hubungan yang lebih kompleks. Selanjutnya, digunakan *MaxPooling2D* sebagai metode *subsampling* atau *downsampling*. *Pooling* ini memiliki ukuran *kernel* 2×2 dengan *stride* 2, yang artinya *output* dari peta fitur diperkecil setengahnya secara spasial sambil mempertahankan fitur penting. Proses ini juga membantu mengurangi jumlah parameter dan mengontrol *overfitting*. Lapisan konvolusi kedua kembali menggunakan filter berukuran 3×3 , namun jumlah filternya ditingkatkan menjadi 64. Proses konvolusi ini diikuti oleh aktivasi ReLU dan pooling yang sama seperti sebelumnya, yaitu *MaxPooling2D* dengan ukuran 2×2 dan *stride* 2 [18]. Hasil dari proses ini kemudian

diratakan menggunakan lapisan *Flatten* untuk mengubah bentuk data multidimensi menjadi vektor satu dimensi agar dapat diproses lebih lanjut oleh lapisan *dense*. Model juga menerima input dari fitur GLCM yang terdiri dari enam nilai fitur tekstur kemudian digabung menggunakan *concatenate* dengan *output* dari *flatten* CNN untuk membentuk representasi fitur gabungan [10]. Gabungan data ini kemudian diteruskan ke lapisan *Fully Connected Layer (Dense)* dengan jumlah neuron sebanyak 64 unit, diikuti oleh fungsi aktivasi ReLU. Setelah itu, digunakan *Dropout Layer* dengan nilai *dropout rate* 0,5 untuk mengurangi *overfitting* dengan menonaktifkan secara acak 50% neuron selama proses pelatihan. Terakhir, model memiliki lapisan *output dense* dengan satu neuron dan fungsi aktivasi *Sigmoid* yang menghasilkan *output* berupa probabilitas antara 0 hingga 1. Nilai ini menunjukkan prediksi apakah citra yang diberikan tergolong sebagai citra asli atau hasil kecerdasan buatan. Arsitektur lengkap ditunjukkan pada Gambar 3.

2.7 Pengujian dan Evaluasi

Pengujian dilakukan dengan parameter ukuran citra 128x128 piksel, jumlah *epoch* 20, serta *batch* 32, dan *random state* 42 dengan *dropout* 0,5. Kedua model diuji dengan parameter yang sama agar hasilnya dapat dibandingkan secara objektif. Evaluasi model dilakukan menggunakan *confusion matrix* dan juga sejumlah metrik untuk mengukur kualitas klasifikasi model terhadap data uji. *Confusion matrix* adalah alat evaluasi yang digunakan untuk mengukur kinerja suatu model klasifikasi dengan memberikan gambaran mengenai jumlah prediksi yang benar dan salah. *Confusion matrix* terdiri dari empat komponen utama, yaitu TP (*True Positive*) mengacu pada kondisi di mana model dengan benar memprediksi kelas positif. TN (*True Negative*) adalah kondisi di mana model dengan benar memprediksi kelas negatif. FP (*False Positive*) terjadi ketika model salah memprediksi kelas positif, padahal sebenarnya negatif [19, 20]. Ini juga dikenal sebagai “*Type I Error*”, sedangkan FN (*False Negative*) adalah kondisi di mana model salah memprediksi kelas negatif, padahal sebenarnya positif. Ini dikenal sebagai “*Type II Error*”. Visualisasi dari *confusion matrix* dapat dilihat pada Gambar 4.



Gambar 4. Heatmap confusion matrix

Metrik pertama yang digunakan untuk mengevaluasi performa model adalah akurasi (*accuracy*), yang menunjukkan rasio prediksi yang benar terhadap total prediksi yang dilakukan oleh model. Akurasi memberikan gambaran seberapa sering model membuat keputusan yang tepat, baik dalam mengenali kelas positif maupun negatif. Metrik ini dapat dihitung menggunakan rumus (9):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

Selain akurasi, metrik presisi (*precision*) juga digunakan untuk menilai performa model, khususnya dalam konteks klasifikasi kelas positif. Presisi mengukur seberapa tepat model dalam memberikan label positif, atau dengan kata lain dari seluruh prediksi positif yang dibuat oleh model, berapa banyak yang benar-benar positif. Nilai presisi yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan *false positive*

yang rendah. Presisi dihitung dengan rumus (10):

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

Kemudian, metrik *recall* berfokus pada kemampuan model dalam menangkap seluruh data yang benar-benar positif. Metrik ini bertujuan untuk memastikan bahwa semua kasus positif berhasil terdeteksi oleh model, meskipun dengan risiko terdapat beberapa *false positive*. *Recall* dihitung dengan rumus (11):

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

Metrik yang terakhir yaitu F1-score, adalah rata-rata harmonis dari presisi dan recall. Metrik ini memberikan keseimbangan antara keduanya, sehingga sangat berguna terutama pada situasi di mana distribusi kelas tidak seimbang. Rumus F1-score adalah rumus (??)berikut:

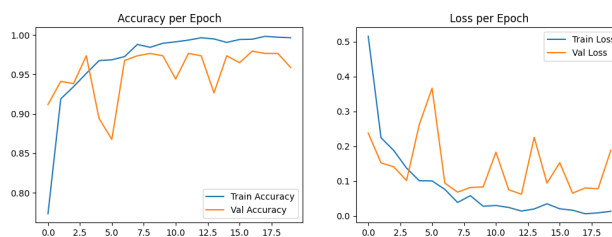
$$F1 - score = 2 \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (12)$$

Untuk memberikan gambaran performa model yang lebih lengkap, digunakan pula *classification report*. Laporan ini memuat nilai presisi, *recall*, *f1-score*, dan *support* dalam satu tabel. *Classification report* sangat membantu dalam mengevaluasi pada dataset yang tidak seimbang, karena memungkinkan untuk melihat apakah model cenderung bias terhadap salah satu kelas tertentu atau tidak. Dengan laporan ini, analisis performa model menjadi lebih transparan dan menyeluruh.

3. Hasil dan Pembahasan

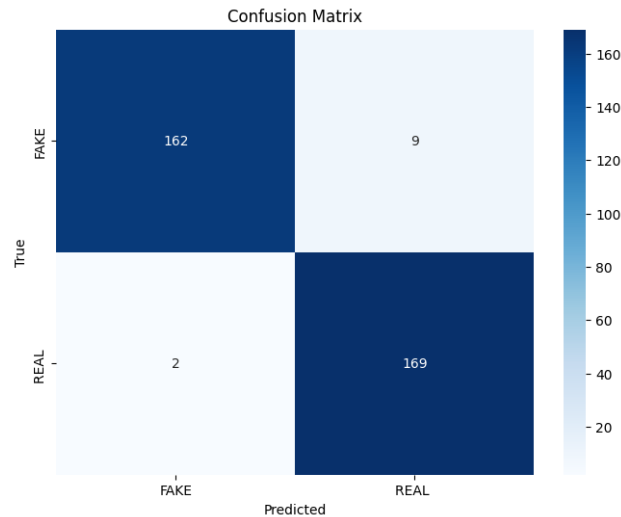
3.1 Hasil Model CNN

Berdasarkan hasil pengujian model CNN murni jika dilihat dari grafik akurasi dan loss per epoch pada Gambar 5, grafik menunjukkan bahwa nilai *training loss* secara konsisten menurun hingga mendekati nol, sedangkan *validation loss* juga mengalami penurunan meskipun terdapat fluktuasi kecil di beberapa *epoch* terakhir. *Training accuracy* meningkat secara progresif dan hampir mencapai 100%, sementara *validation accuracy* stabil di angka sekitar 97%. Hal ini menunjukkan bahwa model CNN tidak mengalami *overfitting* yang berarti dan mengindikasikan bahwa model mampu generalisasi dengan baik terhadap data baru.



Gambar 5. Grafik akurasi dan loss per epoch model CNN

Selain itu hasil evaluasi menggunakan *confusion matrix*, dari 171 citra *fake* terdapat 9 citra yang salah diklasifikasikan sebagai *real* dan 162 citra lainnya berhasil dikenali sebagai *fake*. Sementara pada kelas *real*, dari 171 citra *real* berhasil mengklasifikasikan 169 citra dan hanya 2 citra *real* yang diklasifikasikan sebagai *fake*, hasil tersebut dapat dilihat pada Gambar 6. Lebih lanjut evaluasi model berdasarkan *classification report* yang disajikan pada Tabel 1 menunjukkan nilai akurasi keseluruhan model adalah 0.97 dengan nilai *macro average* dan *weighted average* sebesar 0.97 pada semua metrik evaluasi. Nilai presisi yang dihasilkan pada kelas *fake* sebesar 0.99 yang artinya seluruh prediksi yang diklasifikasikan sebagai *fake* hampir benar semua. Tetapi nilai *recall*-nya 0.95 sehingga dari seluruh data *fake* yang sebenarnya ada, hanya 95% yang berhasil dikenali oleh model. Nilai *F1-score* sebesar 0.97 menunjukkan keseimbangan yang sangat baik antara presisi dan *recall* untuk kelas *fake*. Sebaliknya, pada kelas *real* nilai presisinya 0.95 sedikit lebih rendah namun memiliki nilai *recall* 0.99 yang berarti model hampir tidak melewatkan data *real* yang sebenarnya. *Support* untuk masing-masing kelas adalah 171 yang menandakan jumlah sampel data uji untuk masing-masing kelas seimbang. Berdasarkan klasifikasi menggunakan model CNN memperoleh akurasi sebesar 97%.



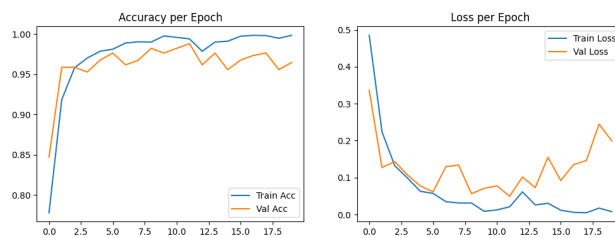
Gambar 6. Confusion matrix model CNN

Tabel 1. Classification report CNN

	Precision	Recall	F1-score	Support
FAKE	0.99	0.95	0.97	171
REAL	0.95	0.99	0.97	171
Accuracy			0.97	342
Macro avg	0.97	0.97	0.97	342
Weighted avg	0.97	0.97	0.97	342

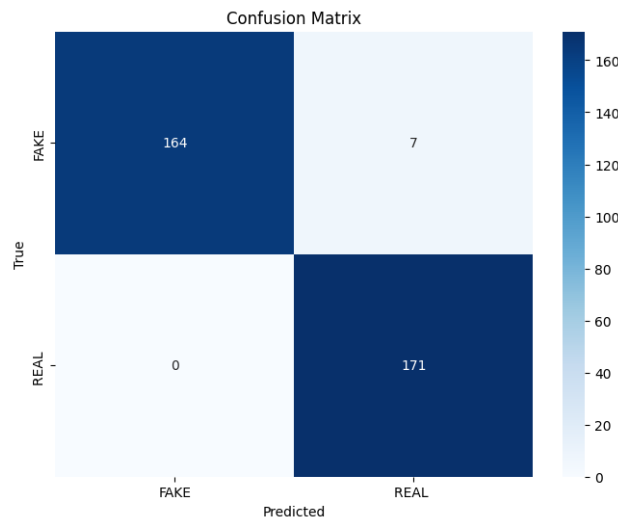
3.2 Hasil Model GLCM-CNN

Hasil pengujian model GLCM-CNN juga menunjukkan performa yang sangat baik, dilihat dari grafik akurasi dan *loss per epoch* pada Gambar 7, grafik *training loss* menurun tajam dan mendekati nol, ini menunjukkan bahwa model mempelajari data latih dengan efektif. *Validation loss* juga menunjukkan tren menurun, namun fluktuasi yang lebih besar dibanding model CNN terlihat di beberapa bagian akhir pelatihan. Hal serupa juga terjadi pada grafik akurasi, di mana *training accuracy* mencapai lebih dari 99%, sedangkan *validation accuracy* bertahan stabil di angka 98%. Fluktuasi ringan pada *validation loss* mengindikasikan bahwa model memiliki tingkat sensitivitas yang lebih tinggi terhadap data validasi, kemungkinan akibat peningkatan kompleksitas akibat penambahan fitur tekstur dari GLCM. Berdasarkan *confusion matrix*



Gambar 7. Grafik akurasi dan loss per epoch model GLCM-CNN

yang disajikan pada Gambar 8, pada kelas *fake* hanya terdapat 7 citra yang salah diklasifikasikan sebagai *real* dari 171 citra dan 164 citra lainnya berhasil diklasifikasikan sebagai *fake*, sedangkan pada kelas *real* seluruh citra berhasil diklasifikasikan dengan benar. Hal ini menunjukkan bahwa model lebih sensitif terhadap kelas *real* tanpa menghasilkan kesalahan klasifikasi pada kelas tersebut. Selain itu, hasil evaluasi menggunakan *classification report* yang disajikan pada Tabel 2, secara keseluruhan, model memperoleh nilai akurasi sebesar 0.98 dengan nilai *macro average* dan *weighted average* masing-masing sebesar 0.98 pada seluruh metrik evaluasi. Pada kelas *fake*, model berhasil mencapai nilai presisi 1.00 yang artinya



Gambar 8. Confusion matrix model GLCM-CNN

seluruh prediksi terhadap kelas *fake* dilakukan dengan benar tanpa adanya *false positive*. Sementara itu, nilai *recall* sebesar 0.96 menunjukkan dari seluruh citra *fake* yang sebenarnya ada berhasil dikenali hingga 96% oleh model. Pada kelas *real* nilai presisi sebesar 0.96, artinya sebagian kecil dari prediksi *real* ternyata salah (*false positive*), namun *recall* mencapai 1.00 sehingga seluruh citra *real* yang sebenarnya ada dapat dikenali dengan benar oleh model tanpa ada yang terlewat. Jumlah data uji masing-masing kelas tetap seimbang, yaitu 171 dari total *support* 342. Akurasi yang didapat dari model ini adalah 98%. Model CNN

Tabel 2. Classification report GLCM-CNN

	Precision	Recall	F1-score	Support
FAKE	1	0.96	0.98	171
REAL	0.96	1	0.98	171
Accuracy			0.98	342
Macro avg	0.98	0.98	0.98	342
Weighted avg	0.98	0.98	0.98	342

murni menunjukkan performa klasifikasi yang cukup baik terhadap citra hasil kecerdasan buatan dengan akurasi sebesar 97%. Berdasarkan hasil evaluasi, model ini mampu mengenali sebagian besar citra dengan benar meskipun masih terdapat beberapa kesalahan klasifikasi, yakni 9 kesalahan pada kelas fake dan 2 kesalahan pada kelas real. Hal ini menunjukkan bahwa CNN mampu menangkap pola spasial dominan yang ada dalam citra, namun belum sepenuhnya efektif dalam membedakan pola mikro atau tekstur halus yang sering kali menjadi ciri khas dari citra buatan.

Sementara itu, model hybrid GLCM-CNN mencatatkan akurasi sebesar 98% dengan hanya 7 kesalahan klasifikasi pada kelas fake dan seluruh citra kelas real berhasil diklasifikasikan dengan benar yang mengindikasikan sensitivitas tinggi terhadap citra asli. Integrasi fitur tekstur dari GLCM dinilai membantu model dalam mengenali pola lokal yang tidak sepenuhnya tertangkap oleh model CNN murni.

Meskipun terdapat perbedaan akurasi sebesar satu persen antara kedua model, penelitian ini tidak melakukan uji signifikansi statistik. Oleh karena itu, perbedaan performa tersebut tidak dapat dianggap signifikan secara statistik, dan tidak cukup menyimpulkan bahwa salah satu model secara mutlak lebih unggul dari yang lain. Selain itu, proses pelatihan model bersifat stokastik sehingga meskipun konfigurasi parameter sama tetapi hasilnya dapat bervariasi karena faktor inisiasi bobot atau urutan data pada saat pelatihan. Dengan mempertimbangkan hal tersebut, perbedaan performa yang diperoleh dalam penelitian ini lebih tepat dianggap sebagai indikasi awal atau tren bukan sebagai bukti konklusif.

Visualisasi performa model dalam bentuk grafik akurasi dan loss selama pelatihan telah disajikan (Gambar 5 dan Gambar 7) untuk kedua model. Grafik tersebut menunjukkan bahwa model mengalami konver-

gensi yang stabil tanpa indikasi *overfitting* yang signifikan. Selain itu, analisis kesalahan telah disajikan dalam bentuk *classification report* (Tabel 1 dan Tabel 2) dan *confusion matrix* (Gambar 6 dan Gambar 8), yang menjelaskan distribusi klasifikasi yang benar dan salah untuk masing-masing kelas.

Jika dibandingkan dengan pendekatan dari penelitian sebelumnya dalam [8, 9, 10], dan [11] hasil penelitian ini menunjukkan performa yang kompetitif meskipun menggunakan arsitektur yang relatif sederhana. Meskipun penelitian terdahulu tidak secara langsung membahas klasifikasi citra sintetis, kesamaan pendekatan metodologis menunjukkan bahwa kombinasi fitur tekstur dan CNN memiliki potensi yang luas dalam berbagai domain pengolahan citra.

4. Simpulan

Penelitian ini bertujuan untuk mengevaluasi efektivitas pendekatan hybrid GLCM-CNN dalam klasifikasi citra sintetis. Hasil pengujian menunjukkan bahwa model GLCM-CNN memberikan performa yang lebih baik dibandingkan CNN murni, dengan akurasi 98% dan peningkatan pada nilai *precision*, *recall*, dan *f1-score*. Integrasi fitur tekstur dari GLCM terbukti meningkatkan sensitivitas model terhadap pola mikro dalam citra sintetis.

Namun, peningkatan performa tersebut belum dapat dipastikan signifikan secara statistik karena belum dilakukan uji formal seperti McNemar atau t-test. Selain itu, dataset yang digunakan masih terbatas dalam jenis citra dan sumber generator AI, sehingga validitas eksternal model belum sepenuhnya teruji. Potensi *overfitting* juga dapat terjadi mengingat dimensi fitur bertambah akibat penggabungan dua pendekatan. Kontribusi utama penelitian ini adalah menunjukkan potensi integrasi fitur tekstur dan spasial dalam klasifikasi citra kompleks, khususnya yang dihasilkan oleh AI.

Sebagai saran untuk penelitian selanjutnya dapat melakukan pengujian terhadap dataset yang lebih luas dan beragam serta model generatif terbaru seperti DALL-E 3 atau Sora. Eksplorasi model CNN yang lebih efisien seperti *MobileNet* atau *EfficientNet* juga dapat menjadi arah pengembangan lanjutan. Selain itu, pengembangan sistem klasifikasi berbasis aplikasi juga dapat membuka peluang penerapan di bidang forensik digital atau validasi konten visual, meskipun efektivitasnya masih perlu dibuktikan melalui studi lanjutan.

Pustaka

- [1] S. A. Hakim *et al.*, "Klasifikasi citra generasi artificial intelligence menggunakan metode fine tuning pada residual network," *JTIK*, vol. 11, no. 3, pp. 655–666, Jul. 2024.
- [2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *arXiv*, 2020.
- [3] I. Goodfellow *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.
- [4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016.
- [5] M. M. Folla and S. J. Bulan, "Penerapan metode gray level co-occurrence matrix dalam mengklasifikasi tingkat kematangan buah naga berbasis citra," *JICON*, vol. 11, no. 2, pp. 210–216, Oct. 2023.
- [6] B. Kusuma, T. I. Hermanto, and C. D. Lestari, "Klasifikasi jenis penyakit pada tanaman padi menggunakan algoritma convolutional neural network," *Jurnal Informatika dan Komputer*, vol. 9, no. 1, p. 40, Feb. 2025.
- [7] Y. Li, "Research and application of deep learning in image recognition," in *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*. Shenyang, China: IEEE, Jan. 2022, pp. 994–999.
- [8] A. Alsalihi, H. K. Aljobouri, and E. A. K. AlTameemi, "Glcmm and cnn deep learning model for improved mri breast tumors detection," *International Journal of Online Engineering*, vol. 18, no. 12, pp. 123–137, Sep. 2022.

- [9] A. R. Dani and I. Handayani, "Klasifikasi motif batik yogyakarta menggunakan metode glcm dan cnn," *Jurnal Teknologi Terpadu*, vol. 10, no. 2, pp. 142–156, Dec. 2024.
- [10] P. Purnawansyah *et al.*, "Comparative study of herbal leaves classification using hybrid of glcm-svm and glcm-cnn," *Ilkom: Jurnal Ilmiah*, vol. 15, no. 2, pp. 382–389, Aug. 2023.
- [11] P. Setiaji, K. Adi, and B. Surarso, "Development of classification method for determining chicken egg quality using glcm-cnn method," *Ingénierie des Systèmes d'Information*, vol. 29, no. 2, pp. 397–407, Apr. 2024.
- [12] J. J. Bird, C. M. Barnes, and A. Lotfi, "Ai generated art: Latent diffusion-based style and detection," in *Advances in Computational Intelligence Systems*, ser. Advances in Intelligent Systems and Computing, N. Naik, P. Jenkins, P. Grace, L. Yang, and S. Prajapat, Eds. Cham: Springer Nature Switzerland, 2024, vol. 1453, pp. 157–169.
- [13] J. Arifin, "Klasifikasi citra tekstur kayu menggunakan gray level co-occurrence matrix dan local binary pattern," *JIKO*, vol. 6, no. 1, p. 34, Feb. 2022.
- [14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, fourth, global edition ed. New York, New York: Pearson Education, 2018.
- [15] A. S. Aziz and F. A. I. A. Putra, "Effect of features and angle on gray level co-occurrence matrix feature extraction on accuracy for object classification," *SMARTICS*, vol. 8, no. 2, pp. 66–72, Oct. 2022.
- [16] M. Akbar, "Traffic sign recognition using convolutional neural networks," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 2, pp. 120–125, Apr. 2021.
- [17] M. Akbar, A. S. Purnomo, and S. Supatman, "Multi-scale convolutional networks untuk pengenalan rambu lalu lintas di indonesia," *SISFOKOM*, vol. 11, no. 3, pp. 310–315, Dec. 2022.
- [18] M. Akbar, P. T. Prasetyaningrum, P. W. Setyaningsih, M. Ahsan, and A. E. Budianto, "Deteksi leukemia limfoblastik akut menggunakan convolutional neural network," *JTSIA*, vol. 7, no. 1, pp. 292–297, Jan. 2024.
- [19] W. A. Naseer, S. Sarwido, and B. B. Wahono, "Gradient boosting optimization with pruning technique for prediction of bmt al-hikmah permata customer data," *JINTEKS*, vol. 6, no. 3, pp. 719–727, Aug. 2024.
- [20] N. Iqbal, R. Mumtaz, U. Shafi, and S. M. H. Zaidi, "Gray level co-occurrence matrix (glcm) texture based crop classification using low altitude remote sensing platforms," *PeerJ Computer Science*, vol. 7, p. e536, May 2021.