

Implementasi *Deep Convolutional Generative Adversarial Network* untuk Pewarnaan Citra *Grayscale*

<http://dx.doi.org/10.28932/jutisi.v8i3.5218>

Riwayat Artikel

Received: 03 Agustus 2022 | Final Revision: 15 November 2022 | Accepted: 15 November 2022

Creative Commons License 4.0 (CC BY – NC)



Muhammad Ricky^{✉#1}, Muhammad Ezar Al Rivan^{#2}

[#] Program Studi Informatika, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang
Jl. Rajawali No. 14, Palembang, 30113, Indonesia

¹muhammadricky974@gmail.com

²meedzhar@mdp.ac.id

[✉]Corresponding author: muhammadricky974@gmail.com

Abstrak — Proses menambahkan warna pada citra *grayscale* diperlukan agar perbaikan pada citra dapat dilakukan secara cepat dan tanpa pengetahuan khusus. Pewarnaan citra menggunakan metode *Deep Convolutional Generative Adversarial Network* (DCGAN) dan metode *Generative Adversarial Network* (GAN). Pelatihan model menggunakan dataset Places365, yang berisikan 98.721 data pelatihan dan 6.600 data pengujian. Citra dikonversi ke dalam ruang warna CIELAB, dengan memanfaatkan *channel L* sebagai input *grayscale* dan *channel AB* sebagai input lainnya. Pengujian dilakukan dengan membandingkan nilai akurasi menggunakan metode *Mean Absolute Error* (MAE) dan *Structural Similarity Index Matrix* (SSIM). Hasil perhitungan metode MAE menunjukkan bahwa rata-rata nilai MAE metode DCGAN lebih kecil dibandingkan metode GAN, dengan skor 10,18 dan 10,81. Hasil perhitungan metode SSIM menunjukkan bahwa metode DCGAN memiliki rata - rata yang lebih tinggi dengan skor 91,54% dan 68,32% untuk metode GAN. Hasil kuesioner yang dilakukan terhadap 30 responden menunjukkan bahwa metode DCGAN dipilih oleh lebih banyak responden dibandingkan metode GAN, masing-masing sebesar 88,40% dan 11,60%.

Kata kunci— DCGAN; GAN; Pewarnaan Citra; Places365 Dataset; SSIM.

Implementation of *Deep Convolutional Generative Adversarial Network* for *Grayscale Image Colorization*

Abstract — The process of adding color to a grayscale image is needed so that improvements to the image can be done quickly and without special knowledge. Image coloring using *Deep Convolutional Generative Adversarial Network* (DCGAN) and *Generative Adversarial Network* (GAN) methods. The model training uses the Places365 dataset, which contains 98,721 training data and 6,600 test data. The image is converted into the CIELAB color space, using the L channel as grayscale input and the AB channel as the other input. The test is done by comparing the accuracy values using the *Mean Absolute Error* (MAE) and *Structural Similarity Index Matrix* (SSIM) methods. The calculation results of the MAE method show that the average MAE value of the DCGAN method is smaller than the GAN method, with a score of 10.18 and 10.81. The results of the calculation of the SSIM method show that the DCGAN method has a higher average with a score of 91.54% and 68.32% for the GAN method. The results of the questionnaire conducted on 30 respondents showed that the DCGAN method was chosen by more respondents than the GAN method, respectively 88.40% and 11.60%.

Keywords—DCGAN; GAN; Image Colorization; Places365 Dataset; SSIM.

I. PENDAHULUAN

Pewarnaan citra adalah proses menambahkan warna pada citra *grayscale*. Secara konvensional, proses pewarnaan citra sangat membosankan, menyita waktu, serta membutuhkan keahlian dan pengetahuan di bidang artistik untuk memberikan warna yang tepat pada sebuah citra *grayscale* [1]. Pada mata manusia umumnya citra *grayscale* akan tampak sedikit aneh, karena terdapat beberapa informasi yang hilang pada citra tersebut [2]. Pewarnaan citra merupakan proses yang sederhana untuk imajinasi manusia, kita hanya perlu mengingat bahwa langit berwarna biru dan rumput berwarna hijau, sedangkan untuk objek lainnya pikiran bebas berimajinasi dengan beberapa warna yang masuk akal.

Beberapa penelitian yang mengangkat topik tentang pewarnaan citra, diantaranya adalah penelitian yang menggunakan metode sederhana namun efektif yang menggabungkan petunjuk pewarnaan dari pengguna dalam *quadratic cost function*, sehingga akan memaksa *neighboring pixels* dalam *space-time* dengan intensitas yang sama harus memiliki warna yang sama dengan petunjuk yang diberikan dalam bentuk “coretan (*scribbles*)” pada input citra *grayscale* dan tanpa informasi tambahan tentang citra yang diinputkan, metode ini mampu menghasilkan pewarnaan berkualitas tinggi secara efisien [3], perluasan dari metode ini semakin meningkatkan kinerjanya dalam melakukan pewarnaan citra [4].

Penelitian lainnya dilakukan dengan memanfaatkan metode *Convolutional Neural Network* (CNN) yang telah dilatih sebelumnya untuk klasifikasi gambar sebagai ekstraksi fitur dalam arsitektur *Novel Residual-Style* yang secara langsung mengeluarkan *full color channel* untuk input gambar. Dilatih menggunakan dataset ImageNet dengan fungsi kerugian *Euclidean* pada nilai *chrominance*, pendekatan ini mencapai hasil yang beragam, yaitu warna yang diprediksi hampir selalu masuk akal, tetapi secara umum cenderung ke arah desaturasi dan bahkan warna kecoklatan (*sephia*) [5]. Penelitian serupa juga dilakukan oleh Zhang et al., pada 2016 menggunakan metode CNN dan distribusi probabilitas dari tiap *pixel*, penelitian ini berhasil mendapat nilai akurasi 17,2% pada SUN dataset [6].

Untuk mengatasi prediksi warna yang secara umum cenderung ke arah desaturasi dan bahkan kecoklatan maka penelitian ini diusulkan menggunakan metode *Generative Adversarial Network*, seperti yang pertama kali diperkenalkan oleh Goodfellow [7], dalam penelitian ini *generator network* akan mengambil citra *grayscale* sebagai input tambahan untuk beberapa *random noise* dan menghasilkan prediksi warna pada citra. Sementara itu, *discriminator network* akan secara acak diberikan citra hasil pewarnaan atau citra asli, *discriminator network* kemudian akan memprediksi apakah inputan yang diberikan merupakan citra asli atau bukan.

Discriminator network akan terus mencoba memaksimalkan akurasi prediksinya, sedangkan *generator network* akan melakukan hal sebaliknya, yaitu meminimalkan akurasi *discriminator*, yang mengarah pada fungsi kerugian alami untuk *backpropagation* yang tidak bergantung pada ukuran jarak *Euclidean* saat bekerja untuk mencocokkan distribusi yang dihasilkan dari pewarnaan ke distribusi sebenarnya yang ada pada dataset.

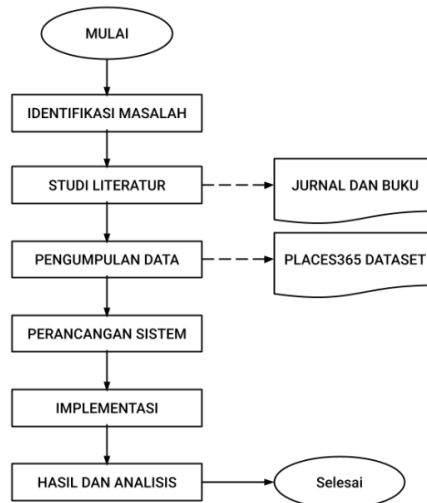
Generative Adversarial Network atau GAN adalah sebuah arsitektur jaringan saraf tiruan yang bertujuan untuk membentuk atau membangkitkan suatu data yang benar-benar baru, dari tidak ada menjadi ada, dikembangkan dan diperkenalkan oleh Ian J. Goodfellow pada 2014, umumnya target utama dari GAN adalah data citra. Secara singkat, jaringan GAN dilatih untuk mampu membangkitkan suatu gambar baru berdasarkan kumpulan gambar yang telah dilihat sebelumnya selama proses pelatihan. Penerapan metode GAN untuk melakukan pewarnaan citra mendapat hasil akurasi 65,5% dengan menggunakan dataset CIFAR-10 [8].

Deep Convolutional Generative Adversarial Network atau DCGAN dianggap sebagai modifikasi dari GAN dengan beberapa perubahan model yaitu penggunaan *convolutional layers* dan *convolutional-transpose layers* pada masing-masing jaringan *discriminator* dan *generator*. Penerapan metode DCGAN untuk melakukan pewarnaan citra, berhasil memperoleh akurasi 82,8% pada dataset LSUN *Bedroom* [9]. Pada 2017 Cao et al., menggunakan metode *Wasserstein Generative Adversarial Network* (WGAN) yang merupakan salah satu variasi dari GAN, untuk melakukan pewarnaan citra dan menguji modelnya menggunakan dataset LSUN *Bedroom*, berhasil memperoleh akurasi 62,6% [10].

Metode DCGAN yang merupakan salah satu variasi dari GAN memiliki akurasi yang lebih baik jika dibandingkan dengan metode yang lain, sehingga pada penelitian ini, dipilih metode DCGAN karena akurasinya yang lebih baik, model pada penelitian ini diuji menggunakan dataset Places365. Penelitian ini dilakukan bertujuan untuk mengetahui implementasi metode DCGAN dan metode GAN pada pewarnaan citra, serta mengetahui tingkat akurasi dari kedua metode tersebut.

II. METODE PENELITIAN

Berikut adalah beberapa tahapan yang dilakukan pada proses pewarnaan citra *grayscale* dengan menggunakan metode DCGAN dan metode GAN sebagai pembandingan. Tahapan pewarnaan citra *grayscale* dapat dilihat pada Gambar 1.



Gambar 1. Kerangka kerja penelitian.

A. Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah penelitian mengenai bagaimana melakukan pewarnaan citra *grayscale* dengan menggunakan metode DCGAN dan metode GAN.

B. Studi Literatur

Pada tahapan ini dimulai dengan pencarian dan pembelajaran literatur berupa jurnal dan buku yang berkaitan dengan topik penelitian, yaitu pewarnaan citra *grayscale* khususnya yang menggunakan metode DCGAN dan GAN.

C. Pengumpulan Data

Pada tahap ini pengumpulan dataset berasal dari Places365 dataset yang terdapat pada situs *website* resmi Places365 [11]. Dataset berjumlah sebanyak 98.721 gambar untuk data *training* dan 6.600 gambar untuk data *testing* dengan ukuran masing – masing 256 x 256 *pixel*. Beberapa gambar yang diambil dari dataset dapat dilihat pada Gambar 2.

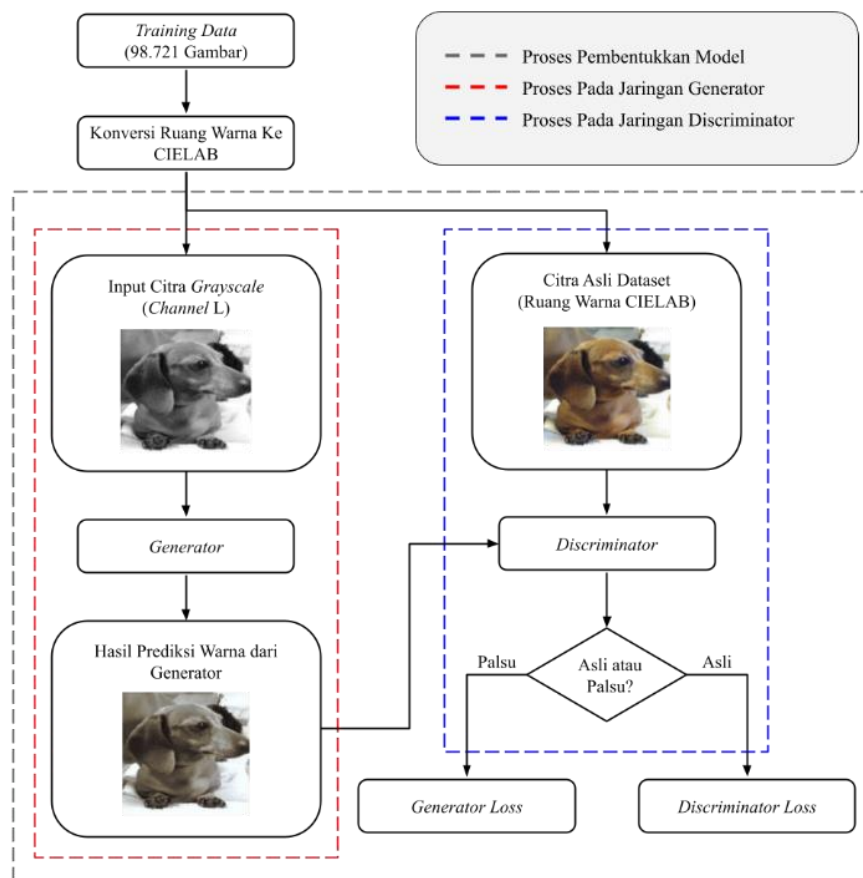


Gambar 2. Contoh dataset Places365.

D. Perancangan Sistem

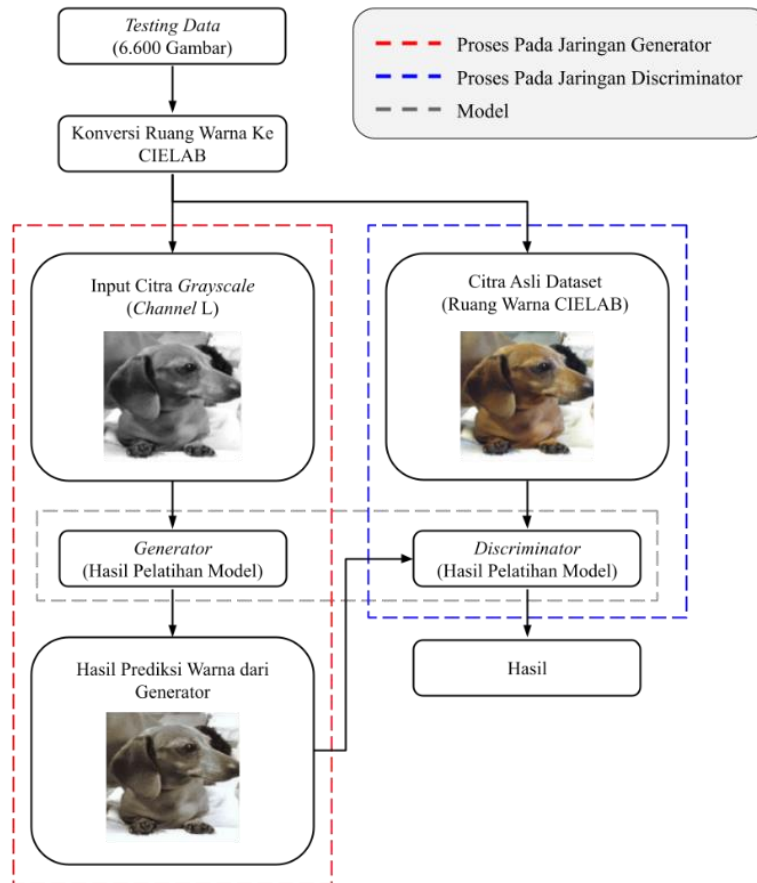
Mula – mula melakukan tahapan *pra-processing*, dimana ruang warna citra yang ada pada dataset dikonversi ke dalam bentuk CIELAB atau LAB, ini dikarenakan ruang warna LAB memiliki *channel* khusus untuk menggambarkan tingkat kecerahan atau skala abu-abu, yaitu *channel L*, sedangkan untuk informasi warna lainnya dikodekan dalam dua *channel* tersisa, *channel A* dan B. Setelah itu akan dilanjutkan proses menambahkan warna pada citra dengan menggunakan metode DCGAN dan GAN. Ada dua tahapan yang akan dilakukan pada proses ini, yaitu tahap pelatihan model dan tahap pengujian.

Metode GAN dan DCGAN memiliki *subnetwork* yang sama yaitu jaringan *generator* dan *discriminator*. Dalam tahap pelatihan model, jaringan *generator* akan diberikan input berupa gambar skala abu-abu (gambar satu *channel*, yaitu *channel L*) dan menghasilkan gambar dua *channel*, yaitu *channel A* dan B dari hasil prediksi warna yang dilakukan. Sedangkan jaringan *discriminator* akan mengambil dua *channel* yang dihasilkan oleh jaringan *generator* dan menggabungkannya dengan citra *grayscale* (gambar satu *channel*) dan memutuskan apakah gambar tiga *channel* yang dihasilkan merupakan citra asli atau bukan. Proses ini akan terus berulang, dimana jaringan *generator* akan terus belajar untuk membentuk citra yang menyerupai citra asli, sedangkan jaringan *discriminator* akan belajar untuk menentukan apakah citra yang dihasilkan oleh jaringan *generator* merupakan citra asli atau bukan, setelah proses tersebut selesai maka ruang warna pada citra akan kembali dikonversi ke dalam bentuk RGB. Proses pelatihan dapat dilihat pada Gambar 3.



Gambar 3. Skema proses pelatihan model.

Pada tahap pengujian, citra yang digunakan berupa citra *grayscale* (*channel L*), lalu dilanjutkan pada tahap memberikan warna pada dua *channel* lainnya (*channel A* dan B). Setelah proses pemberian warna selesai maka, akan didapatkan data yang mendekati atau sama dengan model yang telah dilatih, dapat dilihat pada Gambar 4. Setelah tahap pelatihan dan pengujian yang dilakukan pada masing - masing metode selesai, maka akan dilakukan perbandingan tingkat akurasi.



Gambar 4. Skema proses pengujian model.

E. Implementasi

Setelah melakukan tahap perancangan sistem, maka dilakukan tahapan implementasi ke dalam bentuk program menggunakan Bahasa Pemrograman Python dan Google Colab Pro sebagai *software* implementasi dan uji.

F. Pengujian

Setelah melakukan tahap implementasi, maka dilakukan proses pengujian model pada masing – masing metode dengan menggunakan data *testing*. Untuk mengukur kinerja dari hasil pengujian digunakan metode *Mean Absolute Error* (MAE) dan *Structural Similarity Index Metrix* (SSIM). MAE dihitung dengan mengambil nilai rata – rata kesalahan *absolute* dari gambar yang dihasilkan dan tingkat *pixel* untuk setiap *channel* warna [12], yang secara matematis dapat dilihat pada persamaan berikut.

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \quad (1)$$

Dimana f_i merupakan nilai prediksi, y_i merupakan nilai sebenarnya, dan n adalah jumlah data. Sedangkan SSIM dihitung dengan mengambil nilai *luminance*, *contrast*, dan *structure* dari masing – masing gambar (gambar asli dan gambar hasil prediksi) untuk dibandingkan [13], secara matematis dapat dilihat pada persamaan berikut.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2)$$

Dimana, dengan μ_x merupakan rata – rata x , μ_y merupakan rata – rata y , σ_x merupakan varian dari x , σ_y merupakan varian dari y , dan C_1, C_2 merupakan variabel untuk menstabilkan pembagian, semakin tinggi nilai SSIM berarti semakin baik kualitas citra yang dihasilkan.

Pengujian juga dilakukan dengan melibatkan responden untuk mengetahui perbandingan hasil pewarnaan citra pada masing – masing metode, responden diberikan perbandingan gambar hasil pewarnaan dari kedua metode, untuk kemudian memilih manakah hasil pewarnaan yang lebih antara dua metode yang diuji.

G. Hasil dan Analisis

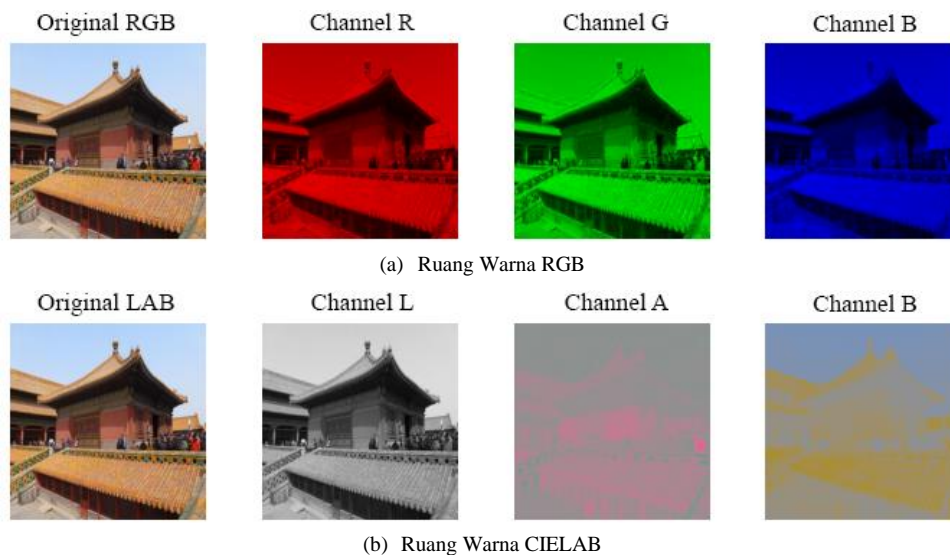
Tahap ini menjabarkan hasil pengujian metode DCGAN dan GAN pada proses pewarnaan citra, untuk berikutnya dilakukan analisis terkait hasil pengujian pada masing - masing metode untuk pengembangan selanjutnya agar bisa dilakukan oleh peneliti berikutnya ataupun sebagai perbandingan untuk penelitian berikutnya.

III. HASIL DAN PEMBAHASAN

Tahapan ini merupakan hasil dari tingkat akurasi yang diperoleh dalam proses pengujian pada metode DCGAN dan GAN. Hasil pengujian dilakukan dengan menggunakan metode perhitungan *Mean Absolute Error* (MAE) dan *Structural Similarity Index Metrix* (SSIM).

A. Implementasi Ruang Warna CIELAB

Pada tahap implementasi konversi ruang warna RGB ke dalam bentuk ruang warna CIELAB, sebagai tahapan *pre-processing*. Ruang warna CIELAB atau LAB digunakan karena memiliki *channel* khusus untuk menggambarkan tingkat kecerahan atau skala abu-abu, yaitu *channel* L, sedangkan untuk informasi warna lainnya dikodekan dalam dua *channel* tersisa, *channel* A dan B, dapat dilihat pada Gambar 5.



Gambar 5. Channel pada ruang warna RGB dan CIELAB.

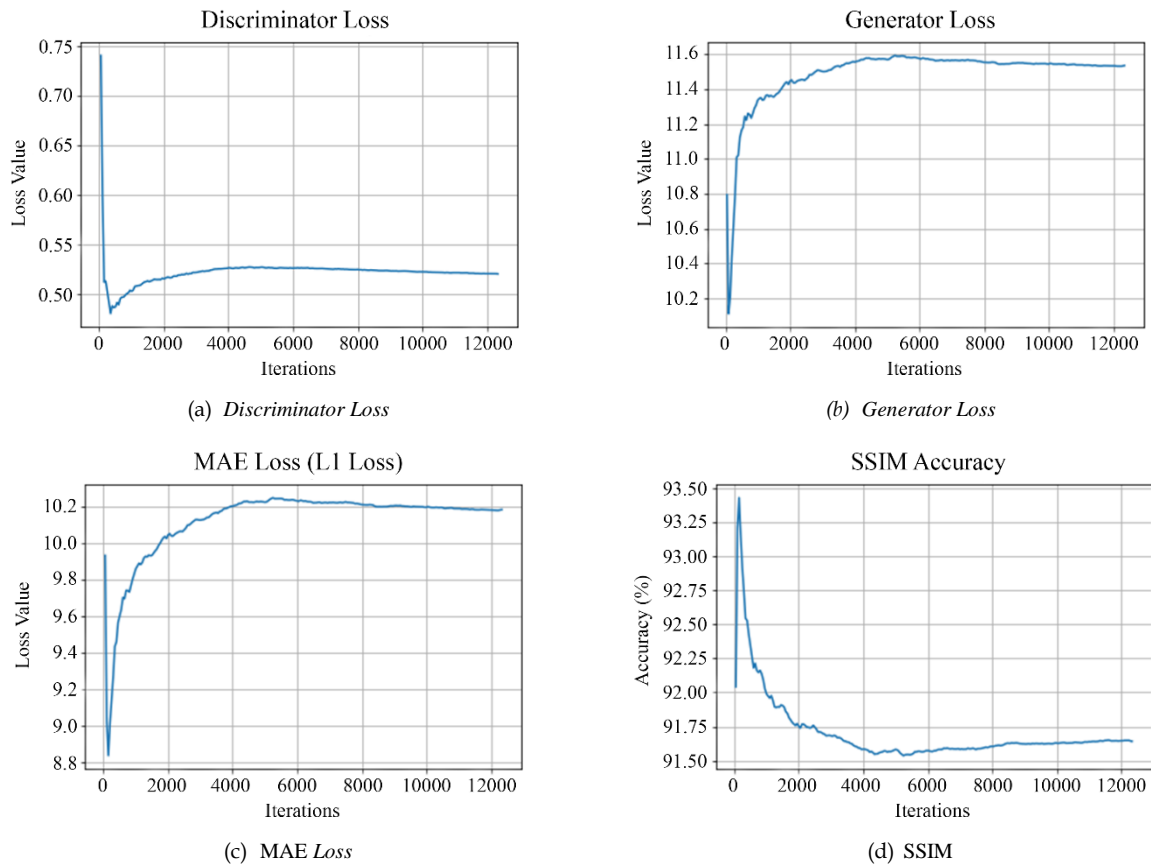
B. Hasil Pengujian Metode DCGAN

Pada tahap ini menjelaskan hasil pewarnaan citra *grayscale* menggunakan metode DCGAN dengan Places365 sebagai dataset. Hasil pengujian yang didapatkan berupa rata - rata nilai *Discriminator Loss*, *Generator Loss*, MAE, dan SSIM yang dapat dilihat pada Tabel 1.

TABEL 1.
HASIL RATA – RATA *DISCRIMINATOR LOSS*, *GENERATOR LOSS*, MAE, DAN SSIM PADA METODE DCGAN

<i>Discriminator Loss</i>	<i>Generator Loss</i>	MAE	SSIM (%)
0.52	11.52	10.18	91.64

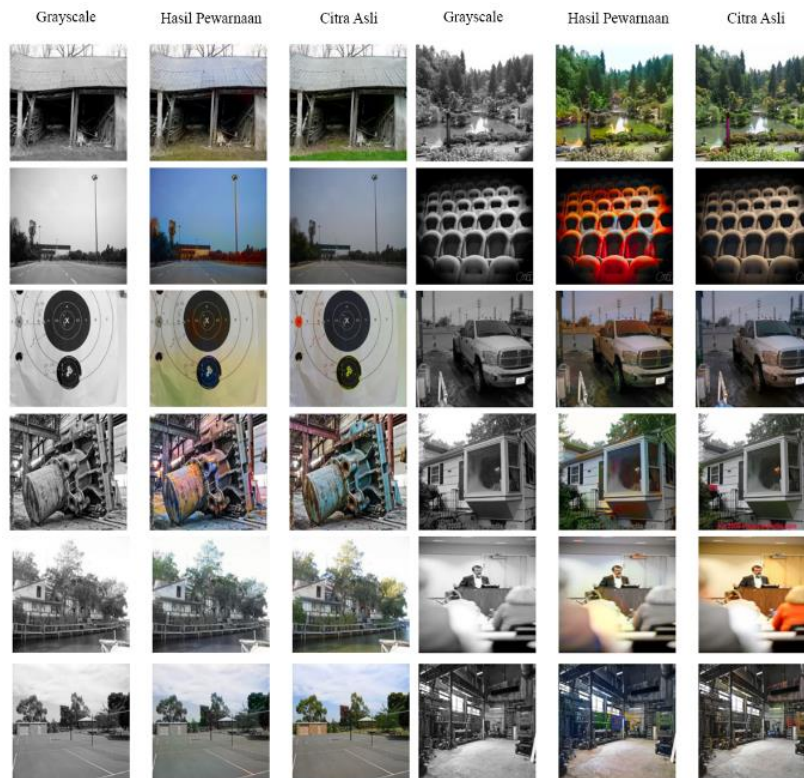
Evaluasi nilai *discriminator loss*, *generator loss*, dan MAE dilakukan dengan cara mencari nilai terkecil, semakin kecil nilai yang didapatkan maka hasil pewarnaan akan semakin baik, sedangkan untuk nilai SSIM dilakukan dengan cara mencari nilai tertinggi, semakin tinggi nilai SSIM maka tingkat kemiripan citra akan semakin baik.



Gambar 6. Perkembangan pelatihan model metode DCGAN.

Seperti yang terlihat pada Gambar 6, dimana pada awal proses pelatihan model nilai *discriminator loss* (a), *generator loss* (b), MAE (c), dan SSIM (d) berada pada posisi yang cukup baik, tetapi seiring dengan bertambahnya iterasi nilai dari *discriminator loss* (a), *generator loss* (b), dan MAE (c) semakin naik sedangkan untuk nilai dari SSIM (d) semakin menurun ini disebabkan karena kecenderungan gambar pada *data training* memerlukan prediksi warna yang cukup spesifik, seperti gambar langit cerah yang membutuhkan gabungan antara warna biru dan kuning, hal ini mengakibatkan kualitas gambar yang dihasilkan oleh generator mengalami sedikit penurunan. Pada awal iterasi performa dari *Discriminator Loss*, *Generator Loss*, MAE dan SSIM mengalami lonjakan. Lonjakan ini disebabkan oleh pada saat awal iterasi algoritma masih mencari parameter yang optimal dan data yang dilatih masih sedikit. Parameter ini terlihat sudah memberikan hasil yang terbaik namun kondisi ini hanya sesaat. Seiring dengan meningkatnya jumlah iterasi dan jumlah data latih parameter menjadi semakin optimal dan grafik menunjukkan kecenderungan lebih stabil.

Hasil pewarnaan citra menggunakan metode DCGAN dapat dilihat pada Gambar 7, dimana selama proses pengujian terdapat beberapa kasus dimana citra tidak sepenuhnya diwarnai dan beberapa citra yang berwarna sedikit kekuningan atau kemerahan, terdapat pula beberapa keluaran citra yang masih cenderung monokrom.



Gambar 7. Hasil pewarnaan metode DCGAN.

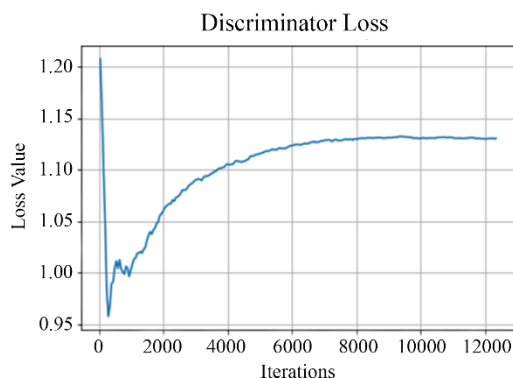
C. Hasil Pengujian Metode GAN

Pada tahap ini menjelaskan hasil pewarnaan citra *grayscale* menggunakan metode GAN dengan Places365 sebagai dataset. Hasil pengujian yang didapatkan berupa rata - rata nilai *Discriminator Loss*, *Generator Loss*, MAE, dan SSIM yang dapat dilihat pada Tabel 2.

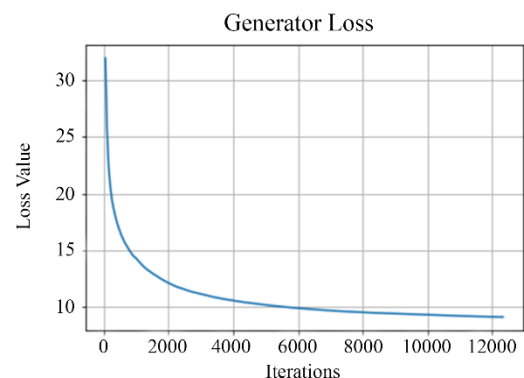
TABEL 2.
HASIL RATA – RATA *DISCRIMINATOR LOSS*, *GENERATOR LOSS*, MAE, DAN SSIM PADA METODE GAN

<i>Discriminator Loss</i>	<i>Generator Loss</i>	MAE	SSIM (%)
1.10	9.04	10.81	68.32

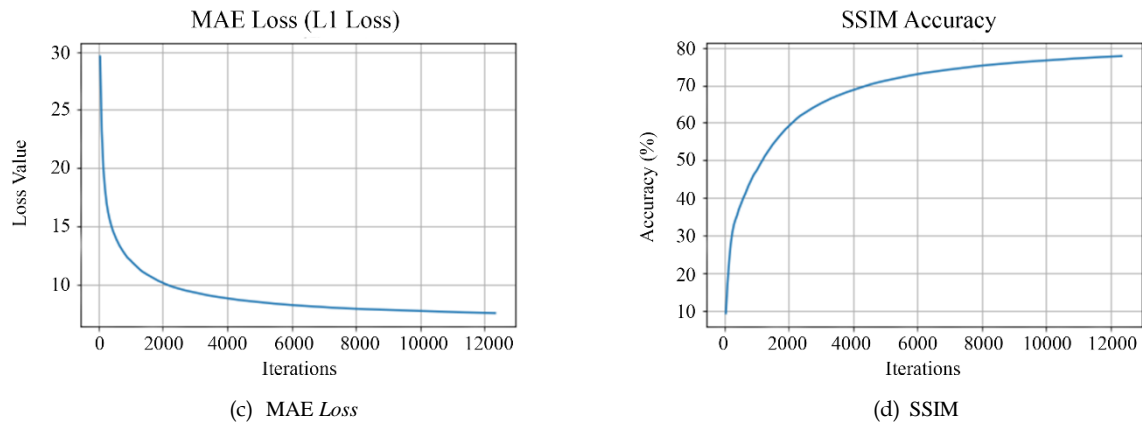
Evaluasi nilai *discriminator loss*, *generator loss*, dan MAE dilakukan dengan cara mencari nilai terkecil, semakin kecil nilai yang didapatkan maka hasil pewarnaan akan semakin baik, sedangkan untuk nilai SSIM dilakukan dengan cara mencari nilai tertinggi, semakin tinggi nilai SSIM maka tingkat kemiripan citra akan semakin baik.



(a) *Discriminator Loss*



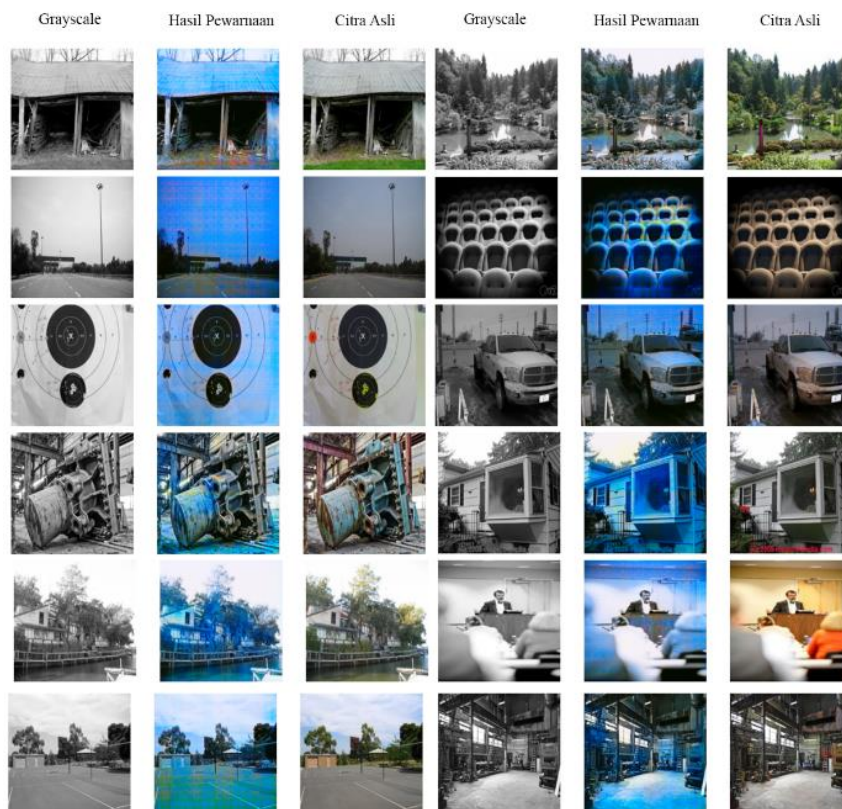
(b) *Generator Loss*



Gambar 8. Perkembangan pelatihan model metode GAN

Seperti yang terlihat pada Gambar 8, dimana seiring dengan bertambahnya iterasi nilai dari *discriminator loss* (a), *generator loss* (b), dan MAE (c) semakin menurun sedangkan untuk nilai dari SSIM (d) semakin naik, hal ini menunjukkan selama proses pelatihan model kualitas citra yang dihasilkan mengalami perkembangan, namun terjadi anomali untuk hasil *discriminator loss* pada Gambar 8(a) antara iterasi ke-300 sampai 900, hal ini dapat terjadi karena kebingungan *model discriminator* dalam memutuskan apakah gambar yang dihasilkan oleh *model generator* palsu atau asli dan ketidakmampuan *model generator* dalam menciptakan atau memprediksi warna pada citra monokrom juga menjadi salah satu faktor penyebab terjadinya anomali, selain itu kebanyakan gambar pada *data training* memerlukan prediksi warna yang cukup spesifik sehingga akan menyulitkan kerja *model generator*.

Hasil pewarnaan citra menggunakan metode GAN dapat dilihat pada Gambar 9, dimana selama proses pengujian terdapat beberapa kasus dimana rata-rata keluaran citra memiliki warna kebiruan, terdapat pula beberapa keluaran yang masih menjadi citra monokrom.



Gambar 9. Hasil pewarnaan metode GAN

D. Hasil Pengujian Keseluruhan

Pada tahap ini menjelaskan hasil rata - rata tingkat akurasi pewarnaan citra *grayscale* antara metode DCGAN dengan metode GAN.

TABEL 3.
HASIL RATA – RATA NILAI MAE DAN SSIM METODE DCGAN DAN GAN

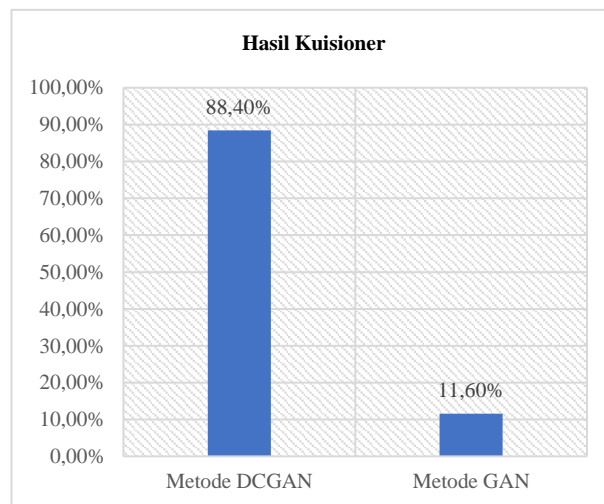
Dataset	Epochs	Batch	Metode	MAE	SSIM (%)
Places365 Dataset	1	8	DCGAN	10.18	91.64
			GAN	10.81	68.32

Berdasarkan Tabel 3 dibuktikan bahwa metode DCGAN lebih baik dibandingkan dengan metode GAN dalam proses pewarnaan citra, ini ditunjukkan dari hasil perhitungan MAE yang lebih rendah dengan selisih 0.63 dan hasil perhitungan SSIM yang lebih tinggi dengan selisih 23.32% jika dibandingkan dengan metode GAN, metode perhitungan SSIM digunakan sebagai matriks untuk mengukur kesamaan antara dua gambar yang diberikan yaitu gambar terdistorsi (citra hasil pewarnaan) dan gambar referensi (citra berwarna) menggunakan berbagai sifat yang diketahui dari sistem visual manusia (mata), semakin tinggi nilai SSIM maka semakin mendekati gambar aslinya (gambar referensi).

E. Hasil Kuesioner

Pada tahap ini menjelaskan hasil kuesioner yang dilakukan terhadap 30 responden dengan menggunakan 30 sampel data acak hasil pewarnaan dari masing – masing metode. Responden diberikan perbandingan gambar hasil pewarnaan dari kedua metode, untuk kemudian memilih manakah hasil pewarnaan yang lebih baik antara dua metode yang diuji.

Berdasarkan penilaian subjektif dari sudut pandang mata manusia metode DCGAN mendapatkan nilai sebesar 88.40% dari total 30 pertanyaan, dapat disimpulkan bahwa hasil pewarnaan metode DCGAN masih jauh lebih dibandingkan dengan metode GAN yang hanya mendapatkan nilai sebesar 11.60% dari aspek visual manusia, yang dapat dilihat pada Gambar 10.



Gambar 10. Grafik hasil kuisisioner.

IV. SIMPULAN

Berdasarkan penelitian yang telah dilaksanakan, dapat ditarik beberapa kesimpulan. Metode DCGAN dan GAN dapat diterapkan pada proses pewarnaan citra *grayscale* dengan nilai rata – rata MAE sebesar 10.18 dan 10.81, selanjutnya untuk nilai rata – rata SSIM sebesar 91.64% dan 68.32%. Hasil pewarnaan metode DCGAN masih lebih unggul, jika dibandingkan dengan metode GAN dari aspek visual manusia, dengan nilai masing – masing 88.40% dan 11.60% dari 30 responden. Untuk penelitian berikutnya disarankan menggunakan metode penelitian yang berbeda atau dapat menggunakan dataset yang berbeda dan dengan jumlah yang lebih banyak.

DAFTAR PUSTAKA

- [1] R. K. Gupta, A. Y.-S. Chia, D. Rajan and H. Zhiyong, "A learning-based approach for automatic image and video colorization," *Computer Graphics International*, 2012.
- [2] R. Zhang and J. Y. Zhu, "Real-time user-guided image colorization with learned deep priors," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.
- [3] A. Levin, D. Lischinski and Y. Weiss, "Colorization using optimization," *ACM Transactions on Graphics*, vol. 23, no. 3, p. 689–694, 2004.
- [4] Y. C. Huang, Y. S. Tung, J. C. Chen, S. W. Wang and J. L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proceedings of the 13th ACM International Conference on Multimedia*, 2005.
- [5] S. Guadarrama, R. Dahl, D. Bieber, M. Norouzi, J. Shlens and K. Murphy, "PixColor: Pixel Recursive Colorization," 2017. [Online]. Available: <https://arxiv.org/abs/1705.07208>.
- [6] R. Zhang, P. Isola and A. A. Efros, "Colorful image colorization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9907 LNCS, p. 649–666, 2016.
- [7] I. G. e. al, "Generative adversarial networks," *Commun ACM*, vol. 63, no. 11, p. 139–144, 2020.
- [8] K. Nazeri, E. Ng and M. Ebrahimi, "Image colorization using generative adversarial networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10945 LNCS, p. 85–94, 2018.
- [9] A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations*, 2016.
- [10] Y. Cao, Z. Zhou, W. Zhang and Y. Yu, "Unsupervised Diverse Colorization via Generative Adversarial Networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10534, p. 151–166, 2017.
- [11] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452-1464, 2018.
- [12] P. Subagyo, *Forecasting Konsep dan Aplikasi Edisi 2*, Yogyakarta: BPPE UGM, 1986.
- [13] P. Datta, "All about Structural Similarity Index (SSIM): Theory + Code in PyTorch," 2020. [Online]. Available: <https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e>. [Accessed 2 Jul 2022].