

# Pewarnaan Otomatis Sketsa Gambar Menggunakan Metode Conditional GAN Untuk Mempercepat Proses Pewarnaan

Regan Reinaldo Kalendesang, Liliana, Djoni Haryadi Setiabudi  
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra  
Jl. Siwalankerto 121 – 131 Surabaya 60236  
Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: reganreinal@gmail.com , lilian@petra.ac.id, djonihs@peter.petra.ac.id

## ABSTRAK

Anime adalah animasi dari Jepang yang dibuat dari kumpulan gambar. Gambar yang digunakan untuk membuat anime dapat dibuat menggunakan tangan atau menggunakan teknologi komputer. Untuk membuat anime diperlukan waktu yang banyak untuk membuat gambarnya. Dalam pembuatan anime 1 detik biasanya memerlukan sekitar 24 frames/gambar yang digunakan, ini yang membuat pembuatan anime sangat panjang dan juga mahal. Selain membuat gambar satu per satu, setiap gambar juga harus diberi warna, dan ini dapat memakan waktu yang sangat lama untuk memproduksi 1 episode sebuah anime.

Metode yang diajukan dalam penelitian ini adalah GAN (Generative Adversarial Network) atau lebih tepatnya C-GAN (Conditional Generative Adversarial Network) yang digunakan untuk memudahkan pengerjaan mewarnai pada sketsa anime. Dataset yang digunakan adalah pasangan sketsa dan sketsa yang sudah diwarnai.

**Kata Kunci:** GAN, Generative Adversarial Network, Auto coloring, Conditional-GAN

## ABSTRACT

*Anime is a Japanese animation that consists of many frames of images. Images that used to make an anime can be made using hand-drawn or using digital-drawn. It takes a lot of time to make an anime. In making anime for 1 second, it needs a total of 24 frames, this is why it takes a lot of time to make anime and also takes a lot of money. Each image also needs to be colored, this is also why making anime takes so much time.*

*The method used in this research is GAN (Generative Adversarial Network) or should we call C-GAN (Conditional Generative Adversarial Network) to make coloring anime sketches easier. Dataset that is used in this research is a pair of sketch images and sketch images that have already been colored.*

**Keywords:** GAN, Generative Adversarial Network, Auto coloring, Conditional-GAN

## 1. PENDAHULUAN

Anime adalah animasi dari Jepang yang dibuat dari kumpulan gambar [1]. Gambar yang digunakan untuk membuat anime dapat dibuat menggunakan tangan atau menggunakan teknologi komputer. Anime disini merujuk kepada semua animasi yang dibuat oleh animasi Jepang tidak ada batasan dalam style pembuatannya. Untuk membuat anime diperlukan waktu yang banyak untuk membuat gambarnya. Dalam pembuatan anime 1

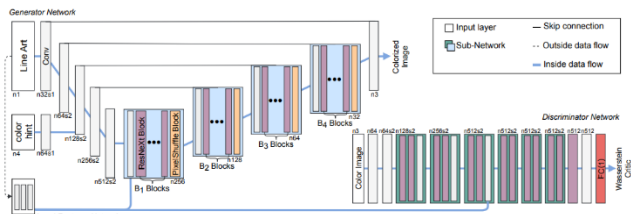
detik biasanya memerlukan sekitar 24 frames/gambar yang digunakan, ini yang membuat pembuatan anime sangat panjang dan juga mahal. Selain membuat gambar satu per satu, setiap gambar juga harus diberi warna, dan ini dapat memakan waktu yang sangat lama untuk memproduksi 1 episode sebuah anime. Untuk coloring di dalam anime juga ada beberapa step yang diperlukan, ada pembuatan warna dasar, shading dan highlight. Warna dasar adalah tahap yang paling penting, karena warna dasar adalah tahap bagi seorang artist untuk membuat karakteristik dari character yang dibuatnya. Tahap ini juga dapat memakan waktu yang sangat panjang, karena perlu adanya percobaan-percobaan yang dilakukan untuk mencocokkan warna. Untuk tahap shading atau pemberian bayangan di dalam gambar juga membutuhkan waktu yang lama, karena shading memerlukan ketelitian dan ketajaman seorang artis dalam memberikan efek bayangan ke dalam gambarnya agar terlihat lebih bagus, rapi dan lebih realistis. Semua tiga langkah dalam tahap coloring ini sangatlah penting, tetapi memakan waktu yang sangat panjang. Maka dari itu, muncul ide untuk memanfaatkan teknologi otomasi dari sebuah metode Deep Learning yaitu GAN (Generative Adversarial Network) dimana mengambil gambar art line sebagai input dan menghasilkan output gambar art line yang sudah memiliki warna dasar dan shading. Dengan memanfaatkan teknologi seperti ini melakukan coloring untuk sebuah sketsa akan menjadi lebih cepat dan mudah. GAN akan cocok untuk digunakan untuk men generate gambar 2D maupun 3D [4], dan berharap jika berhasil, ini akan dapat menguntungkan pembuat anime dalam hal waktu dan dapat mengurangi pengeluaran dalam melakukan coloring. Ini dikarenakan dengan menggunakan metode deep learning seperti GAN, seseorang hanya perlu memberikan instruksi warna apa saja yang nanti akan digunakan dan nanti hanya perlu menunggu hasilnya lalu hasilnya dapat langsung dilengkapi untuk highlight dan shading yang mungkin kurang. GAN yang digunakan adalah cGAN atau Conditional Generative Adversarial Network yang memiliki input layer tambahan yang dapat diatur agar dapat mengatur batasan output yang dikeluarkan. Dengan fitur cGAN ini dapat mempercepat outputnya karena sudah ada batasan-batasan yang sudah ditentukan seperti warnanya.

## 2. LANDASAN TEORI

### 2.1 Conditional-Generative Adversarial Network

Generative Adversarial Network (GAN) adalah sebuah metode yang dikemukakan oleh Goodfellow pertama kali pada tahun 2014 untuk menghasilkan sebuah gambar [7], [9]. Metode GAN sendiri akan menggunakan dua network untuk dapat mengenerate sebuah gambar. Network pertama disebut *generator*, *network* ini bertugas untuk menghasilkan sebuah gambar yang realistis dari input vektor

yang dimasukkan dari training data. *Generator* akan di training terlebih dahulu agar dapat menghasilkan gambar yang realistis. Setelah *generator* dapat menghasilkan sebuah gambar, gambar tersebut akan diteruskan ke *network* ke dua yaitu *discriminator*. *Discriminator* di *training* untuk membedakan apakah gambar yang diberi oleh *generator* adalah gambar asli atau palsu. Jadi, *generator* harus menghasilkan gambar yang dapat membuat *discriminator* meyakini bahwa gambar yang diberikan adalah gambar asli. GAN memiliki perluasan/kepanjangan yaitu cGAN (Conditional Generative Adversarial Network) [6], [7], [8], [13]. Perbedaan yang dimiliki oleh cGAN adalah ia memiliki sebuah tambahan *input layer*, dimana *input layer* ini bertujuan untuk dapat mengatur *output* yang nanti akan dikeluarkan. *Input* berupa nilai vektor yang nanti akan dimasukkan kedalam generator atau discriminator. Input akan direpresentasikan sebagai Y seperti di dalam Gambar 1 dibawah ini.

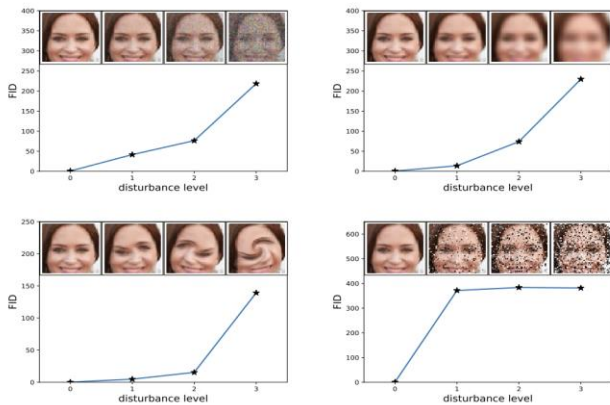


Gambar 1. Struktur Conditional-Gan

Untuk membuat mengenerate gambar berwarna dengan input hanya outline sangatlah sulit, karena informasi yang diterima juga sangat sedikit, berbeda dengan input yang menggunakan input berupa gambar grayscale. Maka dari itu diperlukan constraint yang lebih banyak untuk mendapatkan hasil yang lebih baik.

## 2.2 Frechet Inception Distance

FID atau kepanjangannya Frechet Inception Distance adalah sebuah metric yang biasa digunakan untuk mengevaluasi kualitas dari gambar yang di output [2], [3], [5], [9], [12]. FID dapat menghitung jarak perbedaan dari feature vectors yang didapat dari dari perhitungan gambar asli dan gambar yang telah dihasilkan. Untuk hasil penilaian FID dapat dilihat pada Gambar 2.



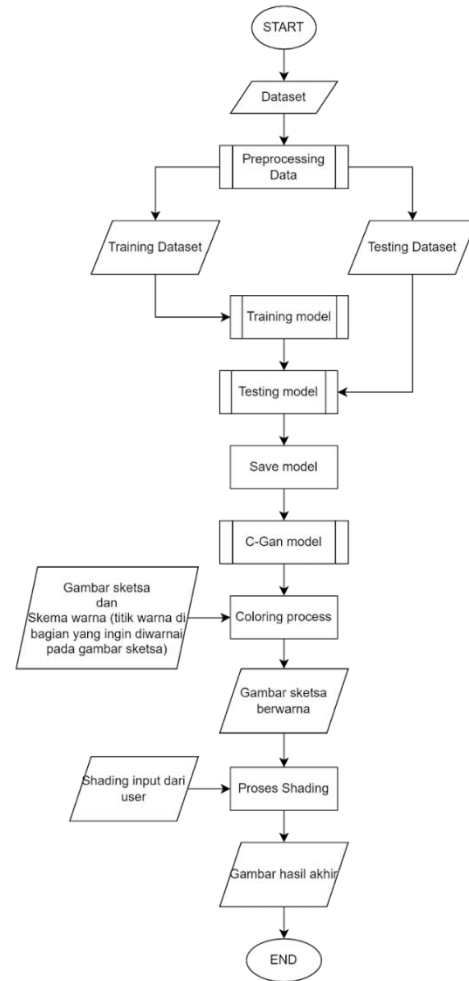
Gambar 2. Pengukuran FID

## 2.3 Mean Opinion Score

MOS atau kepanjangannya Mean Opinion Score adalah salah satu cara untuk mengukur sesuatu berdasarkan nilai yang diberikan dari orang lain melewati survey [3], [11]. Nilai yang diberikan oleh pengambil survey berupa angka dari 1-5 sesuai dengan kriteria yang ingin diberikan.

## 3. DESAIN SISTEM

Sistem yang digunakan dalam skripsi ini memiliki fungsi untuk melakukan pewarnaan otomatis pada gambar sketsa sesuai dengan *input* yang diberikan oleh *user*. Pada bagian ini akan dijelaskan bagaimana proses di dalam sistem tersebut seperti Gambar 3 dibawah ini.



Gambar 3. Main Flowchart

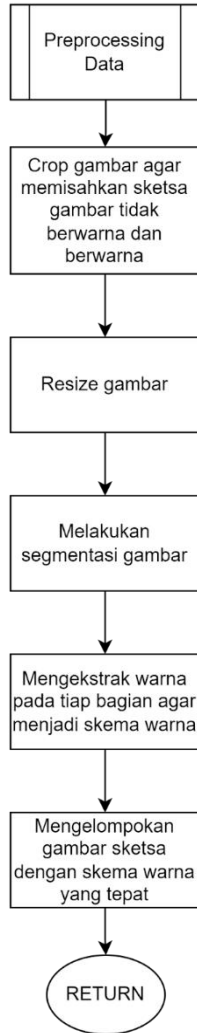
### 3.1 Dataset

Dataset yang digunakan adalah dataset “Anime Sketch Colorization Pair” dan diambil dari sebuah website dataset [10], yaitu Kaggle. Dataset ini berisi beberapa data, yaitu gambar sketsa tidak berwarna, gambar sketsa berwarna, dan warna yang digunakan dalam masing-masing gambar berupa angka RGB (Red Green Blue). Total dataset yang akan digunakan ada sekitar 17.000 gambar sketsa yang tidak berwarna dan berwarna. Dataset akan dibagi menjadi data training sekitar 14,000 data dan data testing sekitar 3,000 data.

### 3.2 Flowchart Preprocessing

Pada bagian ini akan membahas tentang alur dan proses yang akan dilalui saat melakukan preprocessing data. Preprocess diawali dengan melakukan crop pada tiap gambar agar gambar sketsa tidak berwarna dan berwarna dipisahkan. Kemudian gambar-gambar tersebut akan diubah ukurannya menjadi 512 \* 512.

Setelah itu gambar berwarna akan memasuki tahap segmentasi, agar mendapatkan skema warna yang nanti akan digunakan untuk training model. Setelah berhasil mendapatkan skema warna, skema warna akan dipasangkan dengan gambar sketsa tidak berwarna ke dalam satu tag. Skema warna akan digunakan sebagai input yang dapat mempengaruhi hasil output dari model yang akan dibuat. Untuk alur preprocessing dapat dilihat pada Gambar 4.



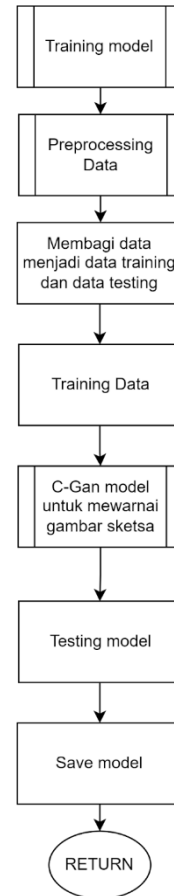
Gambar 4. Flowchart Preprocessing

### 3.3 Flowchart Training

Pada bagian ini akan membahas tentang alur yang dilalui untuk melakukan training model pada sistem. Training menggunakan dataset dari “Anime Sketch Colorization Pair” berjumlah 17,000 data.

Alur flowchart model training diawali dengan melakukan preprocessing data agar data menjadi rapi dan sesuai dengan kebutuhan. Setelah memproses data, data akan dibagi menjadi data untuk training dan data untuk testing. Sesuai dengan yang sudah dijelaskan sebelumnya, jumlah data yang akan digunakan untuk training adalah 14,000 data dan yang digunakan untuk testing adalah 3,000 data.

Training data dilakukan dengan memasukan data training ke dalam model C-Gan. Setelah melakukan training, model nanti akan disimpan agar model dapat dipakai kembali di dalam aplikasi. Untuk alur preprocessing dapat dilihat pada Gambar 5.



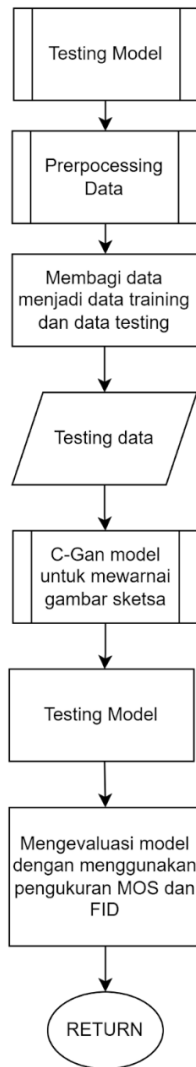
Gambar 5. Flowchart Training

### 3.4 Flowchart Testing

Pada bagian ini akan membahas tentang alur yang dilalui untuk melakukan testing model pada sistem. Pada Gambar 6 menunjukkan alur yang akan dilalui untuk testing model.

Alur flowchart model training diawali dengan melakukan preprocessing data agar data menjadi rapi dan sesuai dengan kebutuhan. Setelah memproses data, data akan dibagi menjadi data untuk training dan data untuk testing. Sesuai dengan yang sudah dijelaskan sebelumnya, jumlah data yang akan digunakan untuk training adalah 14,000 data dan yang digunakan untuk testing adalah 3,000 data.

Testing data dilakukan dengan memasukan data testing ke dalam model C-Gan. Setelah melakukan testing, model akan dievaluasi hasilnya. Evaluasi hasil dari model C-Gan akan menggunakan pengukuran FID dan MOS. FID (Frechet Inception Distance) akan memberikan penilaian terhadap kualitas gambar yang dihasilkan oleh model C-Gan tersebut dengan menggunakan formula perhitungan. MOS (Mean Opinion Score) menggunakan hasil nilai yang diberikan dari survey yang dibagikan kepada orang-orang yang biasa menggambar dan yang melihat anime. Jika hasil evaluasi masih kurang, maka perlu adanya penyesuaian hyper parameter.



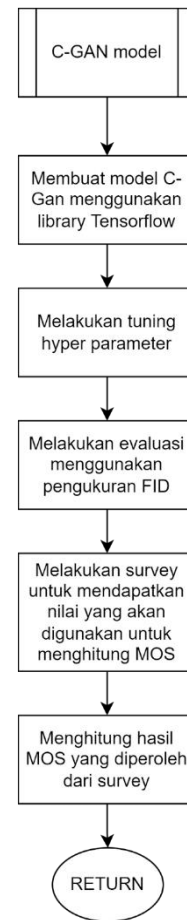
Gambar 6. Flowchart Testing

### 3.5 Pembuatan C-GAN

Bagian ini akan menggambarkan dan menjelaskan alur pembuatan model C-Gan untuk pewarnaan otomatis. Alur pengerjaan akan digambarkan dengan menggunakan flowchart pada Gambar 7.

Proses pembuatan diawali dengan preprocessing data. File dataset yang digunakan berisi sepasang gambar yang sama, kiri adalah gambar sketsa tidak berwarna dan kanan adalah gambar sketsa berwarna. Sepasang gambar ini harus dipisah terlebih dahulu agar menjadi lebih gampang untuk melakukan training dan testing model.

Pembuatan layer model C-Gan menggunakan library Tensorflow. Pembuatan model menggunakan bahasa Python. Setelah membuat layer model maka akan dilakukan tuning hyper parameter agar model sesuai dengan yang diinginkan. Setelah itu dilakukan evaluasi model dengan melakukan pengukuran menggunakan FID dan MOS.



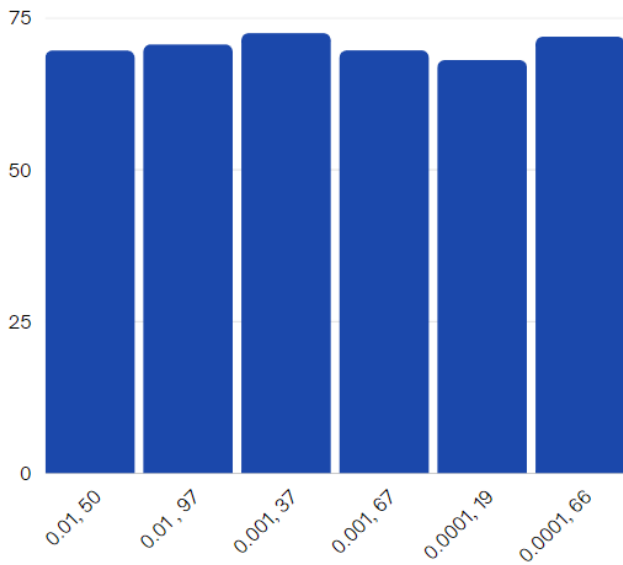
Gambar 7. Flowchart Pembuatan C-GAN

## 4. PENGUJIAN

Pengujian ini akan dilakukan dengan menggunakan parameter yang sudah ditetapkan, yaitu learning rate 0.1, 0.01, 0.001 dan 0.0001. Pengujian ini bertujuan untuk mendapatkan parameter yang dapat menghasilkan hasil pewarnaan yang terbaik.

### 4.1 Hasil Training C-GAN

Hasil training yang telah dilakukan dan ditunjukkan pada grafik loss dibawah untuk tiap parameter menunjukan kalau hasil loss yang terbaik diperoleh oleh parameter learning rate 0.0001 dengan loss terendah 68.020. Kemudian selanjutnya dengan parameter learning rate 0.01 dengan hasil loss terendah 69.598 dan parameter learning rate 0.001 dengan hasil loss terendah 69.620. Hasil loss dari masing-masing model dapat dilihat pada Gambar 8.



Gambar 8. Grafik Loss Generator

## 4.2 Hasil Testing C-GAN

Ada beberapa hasil Analisa dari Gambar 9. yang dapat diambil berdasarkan hasil testing yang dihasilkan dari model dari beberapa parameter yang sudah ditentukan. Untuk parameter *learning rate* 0.01, epoch dengan hasil terbaik yang dapat menghasilkan gambar dengan pewarnaan yang mendekati warna gambar asli adalah epoch 50 dan 97. Lalu dari epoch 50 dan 97, epoch dengan hasil terbaik adalah epoch 97, karena *learning rate* 0.01 epoch 97 dapat menghasilkan gambar dengan pewarnaan yang lebih tebal dan juga tepat daripada *learning rate* 0.01 epoch 50. Untuk beberapa sampel gambar di training juga memperlihatkan *learning rate* 0.01 epoch 97 memiliki shading yang lebih bagus karena lebih jelas daripada *learning rate* 0.01 epoch 50. Jadi dari hasil analisis, dapat disimpulkan bahwa, *learning rate* 0.01 memiliki hasil terbaik pada epoch 97.

Untuk parameter *learning rate* 0.001, epoch dengan hasil terbaik yang dapat menghasilkan gambar dengan pewarnaan yang mendekati warna gambar asli adalah epoch 37 dan 67. Diantara *learning rate* 0.001 epoch 37 dan 67, epoch yang terbaik adalah epoch 67, karena *learning rate* 0.001 epoch 67 dapat menghasilkan warna yang lebih tepat sesuai dengan target, dan shading yang dihasilkan lebih terlihat daripada *learning rate* 0.001 epoch 37. Jadi dari hasil analisis, dapat disimpulkan bahwa *learning rate* 0.001 memiliki hasil terbaik pada epoch 67.

Untuk parameter *learning rate* 0.0001, epoch dengan hasil terbaik yang dapat menghasilkan gambar dengan pewarnaan yang mendekati warna gambar asli adalah epoch 19 dan 66. Diantara *learning rate* 0.0001 epoch 19 dan 66, epoch yang terbaik adalah epoch 66, karena *learning rate* 0.0001 epoch 66 dapat menghasilkan warna yang lebih padat dan tebal dan hampir sesuai dengan target warna yang diinginkan. Jadi dari hasil analisis, dapat disimpulkan bahwa *learning rate* 0.0001 memiliki hasil terbaik pada epoch 66. Pewarnaan yang dihasilkan dari masing-masing model dapat dilihat pada Gambar 9.



Gambar 8. (a) *learning rate* 0.01 epoch 50 (b) *learning rate* 0.01 epoch 97 (c) *learning rate* 0.001 epoch 37 (d) *learning rate* 0.001 epoch 67 (e) *learning rate* 0.0001 epoch 19 (f) *learning rate* 0.0001 epoch 66

## 4.3 Pengujian Kuantitatif

Pada tahap ini, semua model yang sudah terpilih dan dapat menghasilkan hasil pewarnaan yang baik akan melalui pengujian secara kuantitatif agar dapat diukur pencapaiannya dalam melakukan pewarnaan otomatis pada sebuah gambar sketsa. Pengujian menggunakan pengukuran dengan metode FID (Fréchet Inception Distance) yang sudah dijelaskan sebelumnya. Untuk mendapatkan nilai FID dari tiap model yang akan melalui tahap pengujian, pertama hasil gambar yang sudah melalui proses pewarnaan akan dikumpulkan. Hasil gambar dari tiap model kemudian dikumpulkan dan dibandingkan dengan gambar berwarna yang asli (ground truth) menggunakan rumus yang sudah dibahas. Hasil pengujian FID pada model *learning rate* 0.01, 0.001 dan 0.0001 dapat dilihat pada Tabel 1.

**Tabel 1. Pengujian Kuantitatif**

<i>Learning rate</i>	<b>Epoch</b>	<b>FID</b>
0.01	50	79.341
	97	80.183
0.001	37	80.880
	67	80.359
0.0001	19	81.554
	66	58.389

Setelah melalui tahap pengujian kuantitatif menggunakan metode FID, terlihat bahwa parameter learning rate 0.0001 epoch 66 memiliki FID terendah yaitu 58.389 yang menjadikannya FID terkecil dari pada parameter yang lainnya. Meskipun dalam analisis secara visual di tahap pengujian hasil testing learning rate 0.0001 memiliki hasil terburuk, tetapi saat dihitung FID parameter ini menjadi yang paling bagus dari antara semua parameter. Untuk parameter learning rate 0.01 dan 0.001 memiliki perbedaan FID yang sangat tipis yaitu 0.176 dengan learning rate 0.01 FID nya bernilai 80.183 sedangkan learning rate 0.001 FID nya 80.359.

#### 4.4 Pengujian Kualitatif

Pada tahap ini, semua model yang sudah terpilih dan dapat menghasilkan hasil pewarnaan yang baik akan melalui pengujian secara kualitatif agar dapat diukur pencapaiannya dalam melakukan pewarnaan otomatis pada sebuah gambar sketsa oleh orang yang ahli atau yang biasa melihat anime dan menggambar anime. Pengujian menggunakan pengukuran dengan metode MOS (Mean Opinion Score) yang sudah dijelaskan sebelumnya. Metode ini menggunakan kuesioner untuk mendapatkan penilaian dari hasil tiap model yang akan melalui pengujian. Di dalam kuesioner akan diberikan gambar yang dihasilkan oleh tiap model dan nantinya responden akan memberikan nilai kepada tiap gambar yang diberikan. Nilai nanti akan dikumpulkan dan diambil rata-rata agar mendapatkan hasil akhir dari nilai keseluruhan. Batas nilai yang dapat diberi untuk setiap gambar yang akan dinilai adalah 1-5, 1 untuk sangat tidak suka dan 5 untuk sangat suka.

**Tabel 2. Pengujian Kualitatif**

<i>Learning rate</i>	<b>Epoch</b>	<b>MOS</b>
0.01	50	4
	97	3.850
0.001	37	3.715
	67	3.797
0.0001	19	3.457
	66	3.675

Hasil pengujian Tabel 2 didapat dari 107 responden yang memiliki kriteria sering menggambar anime, penggemar anime, belajar art. Ada beberapa analisis yang dapat diambil dari hasil pengujian yang sudah dilakukan. Hasil MOS terbaik yang didapat adalah model learning rate 0.01 epoch 50 dengan MOS bernilai 4. Kemudian berikutnya disusul dengan model learning rate 0.01 epoch 97 dengan nilai MOS 3.850 dan learning rate 0.001 epoch 67 dengan nilai MOS 3.797. Meskipun di penelitian kuantitatif menggunakan penilaian FID learning rate 0.0001 mendapat nilai FID yang lebih baik dari pada semua model, tetapi dari penilaian responden, learning rate 0.01 lebih disukai oleh responden.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

- Target pada perumusan masalah ada yang tercapai dan ada yang tidak. Untuk pengujian FID, pada perumusan masalah memiliki target 25 tetapi, dalam pengujian FID yang paling kecil memiliki nilai 68.020. Ini disebabkan oleh hasil output yang memiliki resolusi rendah dan beberapa warna yang tidak terpenuhi atau tidak sama dengan warna asli. Untuk pengujian MOS, pada perumusan masalah memiliki target 3 lalu, dalam pengujian MOS yang paling tinggi mendapat nilai 4.
- Metode C-Gan dengan menggunakan U-Net dan VGG16 dapat membantu pengerjaan pewarnaan pada sketsa anime.
- Hasil gambar dari proses pewarnaan otomatis memberikan hasil yang bagus pada pewarnaannya, dan dapat memberikan efek shading pada beberapa sampel gambar. Tetapi memberikan output gambar dengan resolusi yang lebih kecil daripada gambar input.
- Metode C-Gan dengan menggunakan U-Net dan VGG16 dapat mengeluarkan hasil yang paling bagus dengan menggunakan parameter learning rate 0.01 dalam pengujian secara visual atau pengujian secara kualitatif, tetapi untuk pengujian secara kuantitatif learning rate 0.0001 memiliki keunggulan dari semuanya.
- Model dengan metode C-Gan dengan menggunakan U-Net dan VGG16 sebagian besar tidak dapat menghasilkan pewarnaan yang bagus dengan menggunakan dataset dari internet yang perbedaannya jauh dari training dan testing. Tapi beberapa kasus model dapat mewarnai dataset dari internet dengan baik

### 5.2 Saran

Untuk penelitian selanjutnya, bisa melakukan preprocessing yang lebih baik, terutama dalam hal memberikan titik-titik warna ke dalam gambar sketsa. Hal ini sangat penting karena pasti berpengaruh pada hasil training dan kesuksesan training juga. Harus berhati-hati dengan warna yang mendekati warna putih, seperti warna kulit. Bisa juga dengan merubah sistem yang dipakai seperti contoh merubah U-Net dengan Res-Net. Penambahan dataset juga diperlukan agar mendapatkan hasil yang lebih bagus lagi.

## 6. REFERENSI

- [1] Bond, J.-M. 2021, January 27. A brief history of anime genres, culture, and evolution. The Daily Dot. Retrieved August 16, 2021, from <https://www.dailydot.com/parsec/what-is-anime/>
- [2] Bynagari, N. B. 2019. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. Asian

Journal of Applied Science and Engineering.

[https://doi.org/ISSN 2305-915X\(p\); 2307-9584\(e\)](https://doi.org/ISSN 2305-915X(p); 2307-9584(e))

- [3] Ci, Y., Ma, X., Wang, Z., Li, H., & Luo, Z. 2018. User-guided deep anime line art colorization with conditional adversarial networks. Proceedings of the 26th ACM International Conference on Multimedia. <https://doi.org/10.1145/3240508.3240661>
- [4] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. 2018. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1), 53–65. <https://doi.org/10.1109/msp.2017.2765202>
- [5] Frans, K. 2017. Outline Colorization through Tandem Adversarial Networks. <https://doi.org/arXiv:1704.08834>
- [6] Goodfellow, I. J., Bengio, Y., Courville, A., Ozair, S., Warde-Farley, D., Xu, B., Mirza, M., & Pouget-Abadie, J. 2014. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*.
- [7] Hensman, P., & Aizawa, K. 2017. cGAN-Based manga Colorization using a single training image. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). <https://doi.org/10.1109/icdar.2017.295>
- [8] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. 2018. Advances in neural information processing systems. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.
- [9] Karras, T., Aila, T., Laine, S., & Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. <https://doi.org/arXiv:1710.10196>
- [10] Kim, T. 2018, December 14. Anime sketch colorization pair. Kaggle. Retrieved December 16, 2021, from <https://www.kaggle.com/ktaebum/anime-sketch-colorization-pair>
- [11] Liu, Y., Qin, Z., Wan, T., & Luo, Z. 2018. Auto-painter: Cartoon image generation from sketch by using conditional Wasserstein generative adversarial networks. *Neurocomputing*, 311, 78–87. <https://doi.org/10.1016/j.neucom.2018.05.045>
- [12] Nazeri, K., Ng, E., & Ebrahimi, M. 2018. Image colorization using generative adversarial networks. *Articulated Motion and Deformable Objects*, 85–94. [https://doi.org/10.1007/978-3-319-94544-6\\_9](https://doi.org/10.1007/978-3-319-94544-6_9)
- [13] Sun, Q., Chen, Y., Tao, W., Jiang, H., Zhang, M., Chen, K., & Erdt, M. 2021. A gan-based approach toward architectural line Drawing colorization Prototyping. *The Visual Computer*. <https://doi.org/10.1007/s00371-021-02219-x>