

# IMPLEMENTASI DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS (DCGAN) UNTUK AUGMENTASI DATA DALAM DETEKSI PENYAKIT DAUN PADI

Winda Asmarawati<sup>1</sup>, Kania Evita Dewi<sup>2</sup>

<sup>1,2</sup> Program Studi Teknik Informatika, Universitas Komputer Indonesia  
Jl. Dipati Ukur No.112-116, Lebakgede, Kecamatan Coblong, Kota Bandung, Jawa Barat 40132  
E-mail : kania.evita.dewi@email.unikom.ac.id<sup>2</sup>

## Abstrak

Penelitian ini mengevaluasi pengaruh augmentasi data menggunakan *Deep Convolutional Generative Adversarial Networks* (DCGAN) dalam mendeteksi penyakit daun pada padi (*Oryza sativa L.*), seperti *bacterial leaf blight*, *brown spot*, dan *leaf smut*, dengan dataset terbatas yang terdiri dari 40 gambar per penyakit. Pra-pemrosesan dilakukan dengan mengubah ukuran gambar menjadi 128x128 piksel menggunakan interpolasi bilinear, serta menormalisasi piksel ke rentang -1 hingga 1, sesuai kebutuhan jaringan DCGAN yang menggunakan aktivasi tanh. Augmentasi data dilakukan dengan menambah jumlah dataset 1 hingga 4 kali lipat dari dataset asli, dan hasilnya dievaluasi menggunakan *Fréchet Inception Distance* (FID). Hasil terbaik diperoleh ketika augmentasi dilakukan 3 kali lipat, menghasilkan 360 gambar baru dengan skor FID sekitar 300, menunjukkan bahwa gambar sintesis masih kurang mirip dengan dataset asli. Model klasifikasi GoogLeNet berbasis *transfer learning* yang dilatih dengan dataset hasil augmentasi ini mencapai akurasi 93% dalam mendeteksi penyakit daun padi. Penelitian ini membuktikan bahwa augmentasi data menggunakan DCGAN dapat meningkatkan akurasi deteksi, meskipun kualitas gambar sintesis masih perlu diperbaiki.

**Kata kunci :** Penyakit daun padi, Augmentasi data, DCGAN, *Transfer learning*, *Fréchet Inception Distance*

## Abstract

*This study evaluates the effect of data augmentation using Deep Convolutional Generative Adversarial Networks (DCGAN) in detecting leaf diseases in rice (Oryza sativa L.), such as bacterial leaf blight, brown spot, and leaf smut, with a limited dataset consisting of 40 images per disease. Pre-processing was done by resizing the images to 128x128 pixels using bilinear interpolation, as well as normalizing the pixels to a range of -1 to 1, as required by the DCGAN network that uses tanh activation. Data augmentation was performed by increasing the number of datasets 1 to 4 times the original dataset, and the results were evaluated using Fréchet Inception Distance (FID). The best results were obtained when augmentation was performed 3 times, resulting in 360 new images with an FID score of around 300, indicating that the synthesized images were still less similar to the original dataset. The transfer learning-based GoogLeNet classification model trained with this augmented dataset achieved 93% accuracy in detecting rice leaf diseases. This research proves that data augmentation using DCGAN can improve detection accuracy, although the quality of the synthesized images still needs to be improved.*

**Keywords :** Rice leaf disease, Data augmentation, DCGAN, Transfer learning, Fréchet Inception Distance

## 1. PENDAHULUAN

Padi (*Oryza sativa L.*) merupakan salah satu komoditas pangan utama di Indonesia dan menjadi sumber karbohidrat esensial bagi jutaan penduduk di Asia. Selain sebagai makanan pokok, padi juga memiliki nilai ekonomi yang tinggi dan mempengaruhi stabilitas pangan di berbagai negara berkembang. Namun, produksi padi seringkali menghadapi tantangan serius akibat serangan berbagai penyakit, terutama pada bagian daun, seperti *Bacterial leaf blight*, *Brown spot*, dan *Leaf smut*. Penyakit-penyakit ini tidak hanya menyebabkan penurunan kualitas, tetapi juga kuantitas hasil panen secara signifikan jika tidak segera dideteksi dan diatasi dengan metode yang tepat [1].

Dalam beberapa tahun terakhir, kemajuan teknologi di bidang *Computer Vision*, terutama yang berbasis *Convolutional Neural Networks* (CNN), telah membuka peluang baru dalam mendeteksi penyakit daun padi secara otomatis. Teknologi ini memungkinkan deteksi dini yang lebih cepat dan akurat, sehingga petani dapat mengambil tindakan preventif sebelum penyakit menyebar luas [2], [3]. Meski demikian, keberhasilan implementasi CNN sangat bergantung pada ketersediaan dataset yang cukup seimbang dan beragam. Dataset yang terbatas dan tidak representatif sering kali menjadi hambatan, mengakibatkan model CNN sulit untuk menggeneralisasi saat diterapkan pada data baru atau kondisi lapangan yang berbeda [4], [5].

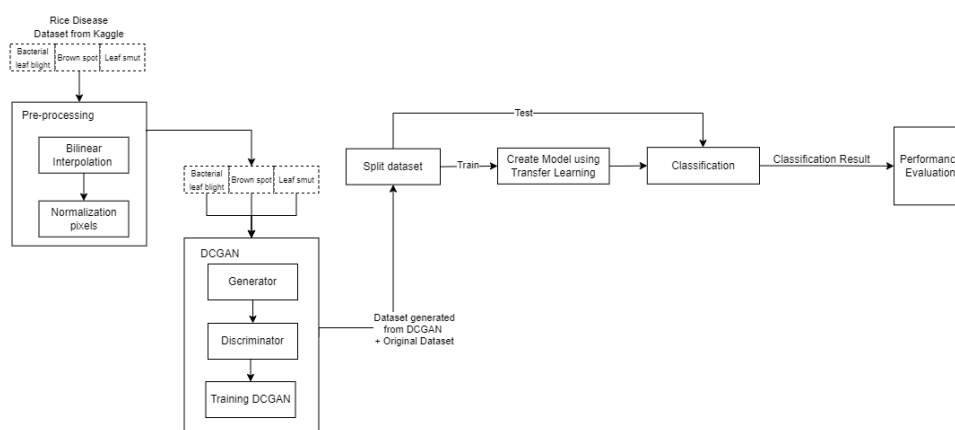
Metode augmentasi data tradisional seperti rotasi, pemotongan, dan perubahan kecerahan gambar sering digunakan untuk meningkatkan jumlah dan keragaman dataset. Augmentasi data juga telah terbukti efektif dalam meningkatkan kinerja CNN pada pengenalan tulisan tangan aksara Sunda [6] dan citra notasi musik [7]. Namun, metode-metode ini memiliki keterbatasan dalam menambahkan variasi yang cukup realistis dan sering kali gagal menangkap seluruh spektrum gejala penyakit daun padi yang mungkin terjadi di lapangan. Hal ini menyebabkan model yang dilatih dengan dataset yang diaugmentasi secara tradisional masih kurang optimal dalam mengenali pola penyakit yang lebih kompleks [8], [9].

Untuk mengatasi keterbatasan augmentasi data tradisional, metode *Deep Convolutional Generative Adversarial Networks* (DCGAN) telah diperkenalkan sebagai solusi yang efektif. DCGAN mampu menghasilkan gambar sintetis yang realistis dan bervariasi, meningkatkan keragaman dataset secara signifikan, terutama pada dataset yang kecil atau terbatas [10], [11]. Wu et al. [3] menunjukkan bahwa DCGAN dapat meningkatkan akurasi model deteksi penyakit daun tomat, sementara Talukdar [5] menemukan bahwa DCGAN efektif dalam menangani ketidakseimbangan kelas dalam klasifikasi penyakit tanaman. Selain itu, Zhu et al. [13] mengungkapkan bahwa DCGAN yang ditingkatkan dapat mengurangi *overfitting*, dan Patmawati et al. [14] menunjukkan manfaat DCGAN dalam augmentasi data gambar tanah. Maka dari itu, aplikasi DCGAN pada deteksi penyakit daun padi bisa dieksplorasi lebih lanjut.

Penelitian ini bertujuan untuk mengevaluasi penggunaan DCGAN dalam augmentasi data guna meningkatkan akurasi klasifikasi penyakit daun padi, khususnya pada dataset yang terbatas. Pengujian dilakukan dengan mengukur kualitas gambar sintetis menggunakan *Frechet Inception Distance* (FID) [12] dan menerapkan *transfer learning* untuk klasifikasi, sejalan dengan temuan Wu et al. [3] yang menunjukkan peningkatan akurasi deteksi kesehatan melalui kombinasi DCGAN dan *transfer learning*. Hasil penelitian ini diharapkan dapat memperkuat efektivitas DCGAN sebagai alat augmentasi data, meningkatkan akurasi dan generalisasi model, serta memberikan kontribusi positif dalam pengelolaan penyakit tanaman padi di sektor pertanian.

## 2. METODOLOGI

Berikut adalah penjelasan mengenai metodologi penelitian yang akan diterapkan dalam penelitian ini yang ditunjukkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

### 2.1 Dataset

Dataset yang digunakan mengambil dari sumber Kaggle vbookshelf berisikan 120 citra yang dikelompokkan menjadi 3 jenis penyakit pada daun padi, yaitu: *Bacterial Leaf Blight*, *Brown Spot*, dan *Leaf Smut* [15]. Setiap jenis penyakit memiliki jumlah gambar yang sama, yaitu 40 gambar.

## 2.2 Pre-processing

*Pre-processing* adalah tahap awal dalam pemrosesan data yang bertujuan untuk mempersiapkan dataset sebelum digunakan dalam pelatihan model. Dalam penelitian ini, terdapat dua tahapan *pre-processing*:

### 2.2.1 Bilinear Interpolation

*Bilinear interpolation* atau biasa disebut interpolasi bilinear adalah metode penskalaan gambar yang digunakan untuk memastikan gambar memiliki resolusi konsisten, dalam hal ini adalah 128x128 piksel, sebelum dimasukkan ke dalam model DCGAN. Teknik ini bekerja dengan menggunakan empat piksel tetangga terdekat untuk memperkirakan nilai piksel baru dalam gambar yang diubah ukurannya. Prosesnya melibatkan penskalaan dalam dua arah, horizontal dan vertikal, yang dijelaskan oleh persamaan 1.

$$I' = S(I, r_x, r_y) = S_y(S_x(I, r_x), r_y) = S_x(S_y(I, r_y), r_x) \quad (1)$$

Dengan  $S$  sebagai fungsi penskalaan, dan  $r_x$  serta  $r_y$  sebagai rasio penskalaan horizontal dan vertikal [16]. Nilai piksel di posisi  $(Y', X')$  dihitung dengan memadukan nilai-nilai piksel dari empat tetangga terdekat, berdasarkan jarak mereka dari titik yang diinginkan, memberikan hasil yang lebih halus. Pembulatan ke bawah sering digunakan untuk memastikan koordinat piksel tetangga tetap berada di dalam batas gambar asli [17].

### 2.2.2 Normalization Pixels

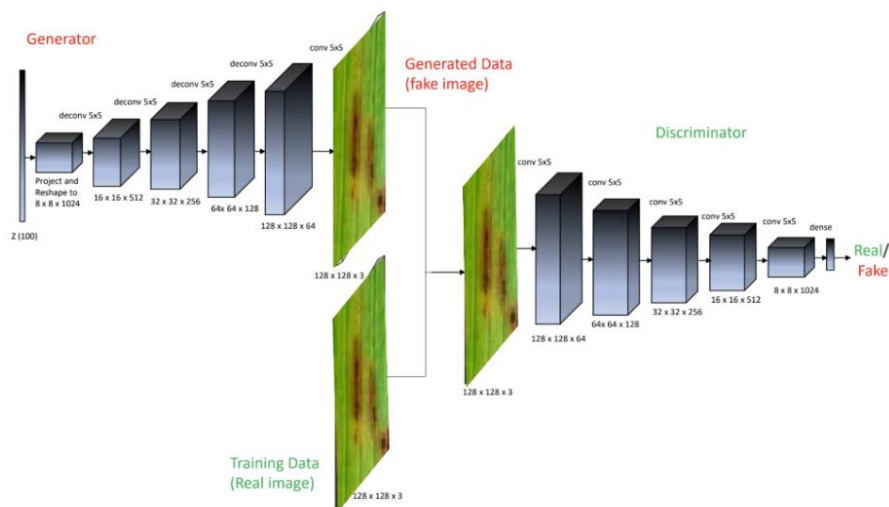
*Normalization Pixels* atau biasa disebut normalisasi piksel adalah teknik yang digunakan untuk mengubah nilai piksel gambar sehingga berada dalam rentang tertentu, dalam hal ini dari [-1, 1]. Teknik ini penting untuk meningkatkan stabilitas dan kecepatan konvergensi model DCGAN, yang menggunakan fungsi aktivasi seperti tanh pada lapisan *output* generator. Berikut akan dijelaskan pada persamaan 2 [18].

$$X'(i, j) = \frac{X(i, j)}{127.5} - 1.0 \quad (2)$$

Di mana  $X'(i, j)$  adalah nilai piksel lapisan fitur yang telah dinormalisasi, dan  $X(i, j)$  adalah nilai piksel asli yang berada dalam rentang [0, 255] [28]. Dalam proses ini, nilai piksel asli pertama-tama dibagi dengan 127.5 untuk mengubah rentangnya dari [0, 255] menjadi [0, 2]. Selanjutnya, nilai yang diperoleh dikurangi dengan 1.0, yang menggeser rentang tersebut menjadi [-1, 1].

## 2.3 Deep Convolutional Generative Adversarial Networks (DCGAN)

*Deep Convolutional Generative Adversarial Networks* (DCGAN) bertujuan mengatasi masalah stabilitas pelatihan GAN dan meningkatkan kualitas representasi fitur dari data gambar tanpa label. Berikut Gambar 2 adalah arsitektur yang digunakan dalam proses implementasi DCGAN dalam augmentasi data yang diadaptasi dari jurnal *Chest X-ray* [19].



**Gambar 2.** Arsitektur *Deep Convolutional Generative Adversarial Networks* (DCGAN) [17]

### 2.3.1 Generator

Generator pada DCGAN dimulai dengan vektor laten berdimensi rendah yang berisi nilai acak dari distribusi normal (Gaussian) dengan mean 0 dan standar deviasi 1 [10]. Vektor ini kemudian diproyeksikan menjadi bentuk 3D dan diubah ukurannya melalui serangkaian lapisan konvolusi *transpose* yang diikuti oleh fungsi aktivasi ReLU atau LeakyReLU untuk menambahkan non-linearitas, dengan lapisan terakhir menggunakan fungsi aktivasi Tanh untuk menghasilkan nilai piksel dalam rentang  $[-1, 1]$ . Struktur ini memungkinkan generator untuk mengubah vektor laten menjadi gambar realistis yang mirip dengan data asli. Detail lengkap mengenai arsitektur generator ini dapat dilihat pada Tabel 1.

**Tabel 1.** Arsitektur Generator

Layer Type	Output Shape	Filters	Kernel Size	Stride	Padding	Activation
Input	(100, 1) noise vector	-	-	-	-	-
Dense	8x8x1024	-	-	-	-	ReLU
Reshape	(8, 8, 1024)	-	-	-	-	-
Batch Normalization (BN)						
Conv2DTranspose + BN	(16, 16, 512)	512	5x5	2	same	ReLU
Conv2DTranspose + BN	(32, 32, 256)	256	5x5	2	same	ReLU
Conv2DTranspose + BN	(64, 64, 128)	128	5x5	2	same	ReLU
Conv2DTranspose + BN	(128, 128, 64)	64	5x5	2	same	ReLU
Output	(128, 128, 3)	3	5x5	1	same	Tanh

### 2.3.2 Diskriminator

Diskriminator pada DCGAN menggunakan arsitektur CNN standar dengan beberapa lapisan konvolusi yang bertugas mengklasifikasikan gambar sebagai asli atau palsu. Proses dimulai dengan gambar berukuran 128x128 piksel yang telah dinormalisasi ke rentang  $[-1, 1]$ . Diskriminator secara berturut-turut mengurangi resolusi spasial data melalui lima lapisan konvolusi, masing-masing diikuti oleh fungsi aktivasi LeakyReLU untuk menangani nilai negatif dan mempercepat konvergensi selama pelatihan. Serta *batch normalization* diterapkan di setiap lapisan untuk menstabilkan dan mempercepat proses pelatihan. Lapisan terakhir menggunakan fungsi aktivasi sigmoid untuk menghasilkan probabilitas bahwa gambar input adalah asli (1) atau palsu (0) [10]. Detail lengkap mengenai arsitektur generator ini dapat dilihat pada Tabel 2.

**Tabel 2.** Arsitektur Diskriminator

Layer Type	Output Shape	Filters	Kernel Size	Stride	Padding	Activation
Input	128x128x3	-	-	-	-	-
Conv2D	(128, 128, 64)	-	5x5	1	same	LeakyReLU
Conv2D + Batch Normalization	(64, 64, 128)	-	5x5	2	same	LeakyReLU
Conv2D + Batch Normalization	(32, 32, 356)	512	5x5	2	same	LeakyReLU
Conv2D + Batch Normalization	(16, 16, 512)	256	5x5	2	same	LeakyReLU
Conv2D + Batch Normalization	(8, 8, 1024)	128	5x5	2	same	LeakyReLU
Output	(0, 1)	64	-	-	-	Sigmoid

### 2.3.2 Training DCGAN

Pelatihan DCGAN melibatkan Generator dan Discriminator yang dilatih secara bergantian dalam kerangka permainan *minimax*. Discriminator dilatih untuk membedakan antara gambar asli dan palsu, dengan fungsi kerugian  $L_D$  yang memaksimalkan kemampuan Discriminator untuk mengenali gambar asli dan menolak gambar palsu. Sebaliknya, Generator dilatih untuk menghasilkan gambar palsu yang dianggap asli oleh Discriminator, dengan meminimalkan fungsi *loss* atau kerugian  $L_G$ . Berikut ini dapat dilihat pada persamaan dibawah [20].

$$L_D = -(\mathbb{E}_{x \sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z)))]]) \quad (3)$$

Dimana :

$\mathbb{E}_{x \sim P_{data}(x)}[\log D(x)]$  : ekspektasi dari log probabilitas bahwa sampel nyata x dianggap nyata oleh discriminator D

$\mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z)))]$  : ekspektasi dari log probabilitas bahwa sampel palsu G(z) dianggap palsu oleh discriminator D

$$L_G = -\mathbb{E}_{z \sim P_Z(z)} [\log D(G(z))] \quad (4)$$

Dimana :

$\mathbb{E}_{z \sim P_Z(z)} [\log D(G(z))]$  : ekspektasi dari log probabilitas bahwa sampel palsu  $G(z)$  dianggap nyata oleh discriminator  $D$

Proses ini dilakukan secara iteratif, dengan tujuan akhir mencapai keseimbangan *Nash* di mana Generator menghasilkan gambar yang sangat mirip dengan gambar asli, dan Discriminator tidak bisa lagi secara konsisten membedakan antara gambar asli dan palsu. Selama pelatihan, evaluasi kinerja dilakukan untuk memastikan bahwa keseimbangan antara Generator dan Discriminator tercapai, menghindari masalah seperti mode *collapse* [20].

## 2.4 Splitting Data

Setelah data tambahan dari DCGAN dihasilkan, data asli dibagi secara manual menjadi dua bagian, dengan sebagian besar dimasukkan ke dalam dataset pelatihan dan sebagian lainnya digunakan untuk dataset pengujian.

### a. Dataset pelatihan (*train*)

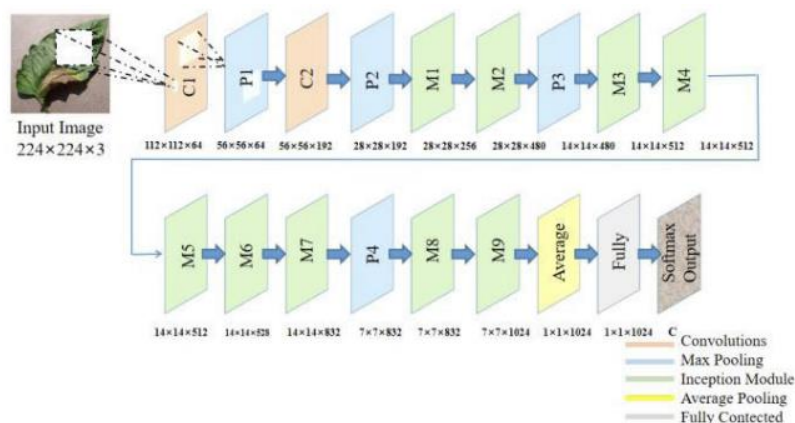
Dataset ini terdiri dari data asli dan data yang dihasilkan oleh DCGAN. Data hasil DCGAN langsung dimasukkan ke dalam dataset pelatihan tanpa dibagi lebih lanjut. Proporsi gambar yang digunakan dalam dataset pelatihan berkisar antara 75% hingga 88.89% dari total gambar yang tersedia. Semakin banyak gambar yang dihasilkan dari proses augmentasi, semakin besar proporsi yang dialokasikan untuk pelatihan.

### b. Dataset pengujian (*test*)

Dataset ini hanya menggunakan data asli. Terdiri dari 60 gambar dari data asli dalam setiap pengujian. Jumlah ini tetap konstan di setiap pengujian, yang secara keseluruhan mewakili 11.11% hingga 25% dari total gambar.

## 2.5 Transfer Learning

Setelah dataset sintesis penyakit daun padi diperoleh, langkah selanjutnya adalah melatih model CNN menggunakan *transfer learning* dengan arsitektur GoogleNet (Inception V1). Arsitektur ini memperkenalkan modul Inception, yang menggabungkan berbagai ukuran kernel konvolusi (1x1, 3x3, 5x5) dan lapisan max-pooling untuk menangkap fitur pada berbagai skala, meningkatkan efisiensi dan akurasi. GoogleNet menggunakan lebih sedikit parameter dibandingkan AlexNet dan mengandalkan average pooling dan dropout untuk mencegah *overfitting*. Dengan 22 lapisan dan 9 modul Inception, GoogleNet menjadi pilihan efektif untuk klasifikasi gambar [21]. Berikut Gambar 3 adalah arsitektur yang menggambarkan proses klasifikasi menggunakan arsitektur GoogleNet yang diadaptasi dari jurnal [3].



Gambar 3. Arsitektur GoogleNet [3]

## 2.6 Performance Evaluation

Setelah pelatihan, kinerja model dievaluasi menggunakan dataset pengujian. Evaluasi dilakukan dengan menghitung akurasi sebagai metrik utama, yang memberikan gambaran tentang seberapa efektif model dalam mengklasifikasikan gambar penyakit daun padi secara tepat. Akurasi ini mencerminkan persentase prediksi yang tepat dibandingkan dengan total jumlah prediksi yang dilakukan oleh model.

### 3. HASIL DAN PEMBAHASAN

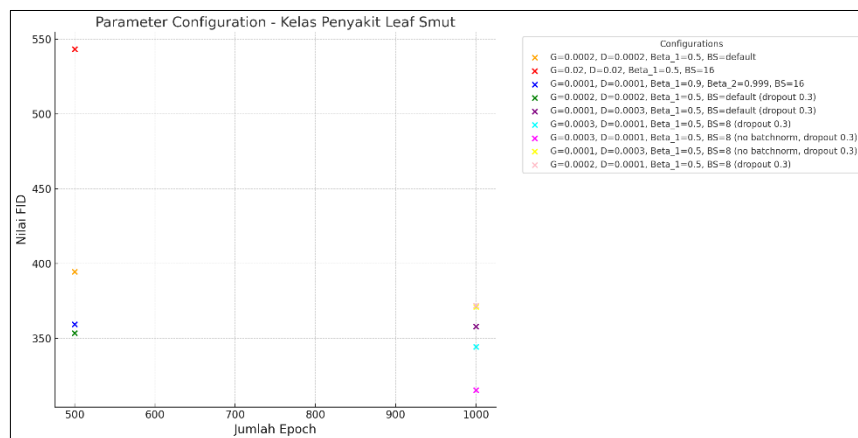
#### 3.1 Konfigurasi Pelatihan

Pelatihan model DCGAN untuk augmentasi data dan klasifikasi menggunakan *transfer learning* dengan arsitektur GoogleNet dilakukan di *platform* Google Colab. Untuk mempercepat proses pelatihan, digunakan TPU (*Tensor Processing Unit*) yang disediakan oleh Colab. Beberapa *library* dan *framework* yang digunakan termasuk TensorFlow untuk implementasi DCGAN dan PyTorch untuk pelatihan GoogleNet.

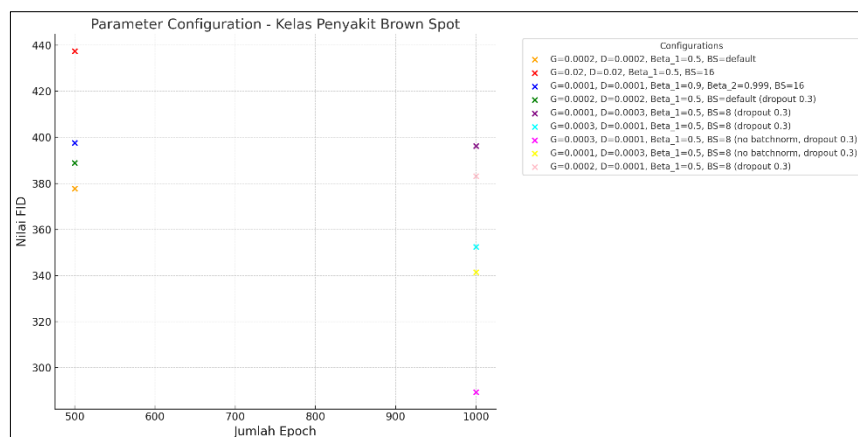
#### 3.2 Pengujian FID

Pengujian ini dilakukan menggunakan metode *Fréchet Inception Distance* (FID) yang memanfaatkan model *pre-trained* InceptionV3 sebagai *feature extractor* di Google Colab. Metode FID ini digunakan untuk mengukur seberapa mirip distribusi gambar yang dihasilkan dengan distribusi gambar asli, yang secara langsung menggambarkan kualitas sintesis gambar. Pengujian FID ini akan disajikan dalam Gambar 4 hingga Gambar 6, yang menampilkan nilai FID untuk berbagai konfigurasi dan skenario pengujian yang telah diterapkan.

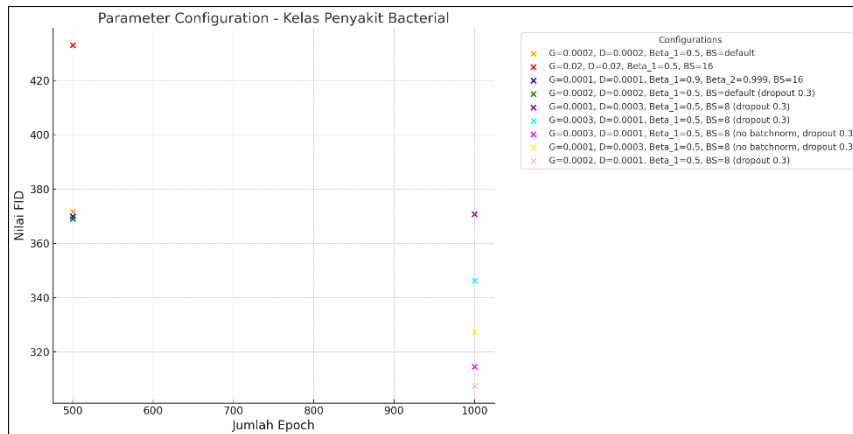
Kemudian keterangan untuk parameter yang digunakan pada gambar adalah sebagai berikut: G adalah *learning rate* untuk generator, berkisar antara 0.0001 hingga 0.0003, mengatur seberapa besar langkah pembelajaran generator. D adalah *learning rate* untuk discriminator dengan rentang nilai yang sama, namun sering kali sedikit lebih tinggi atau sama untuk menjaga keseimbangan. Beta 1 mengacu pada momentum dalam *optimizer* Adam, biasanya diatur antara 0.5 hingga 0.9 untuk mengurangi fluktuasi dan menjaga stabilitas GAN. BS menunjukkan *batch size*, dengan rentang 8 hingga 32, disesuaikan dengan kapasitas memori dan jumlah data.



Gambar 4. Pengujian Performansi *Fréchet Inception Distance* (FID) penyakit *Leaf smut*



Gambar 5. Pengujian Performansi *Fréchet Inception Distance* (FID) penyakit *Brown spot*



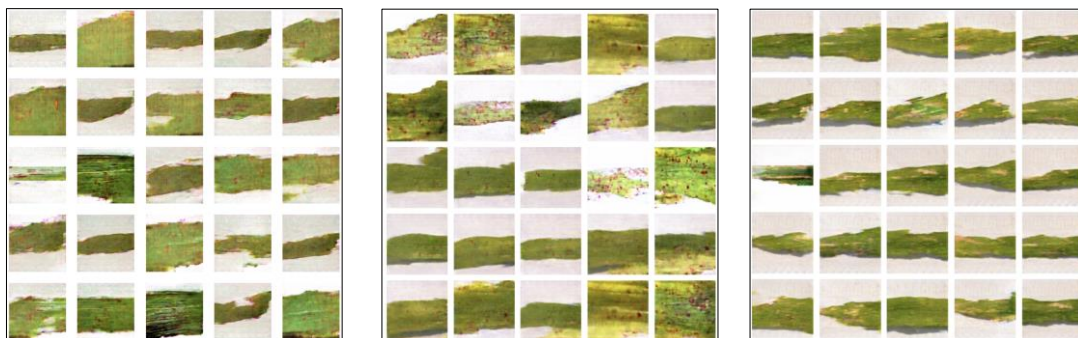
**Gambar 6.** Pengujian Performansi *Fréchet Inception Distance* (FID) penyakit *Bacterial leaf blight*

### 3.3 Pengujian Klasifikasi

Pengujian ini dilakukan dengan menggunakan arsitektur GoogleNet dan berfokus pada evaluasi akurasi model untuk mengidentifikasi penyakit daun pada padi. Model diuji dengan berbagai kombinasi *hyperparameter*, yaitu *learning rate* sebesar 0.001, *optimizer* SGD, *batch size* 32, dan epoch sebanyak 10, menggunakan data augmentasi yang dihasilkan melalui dua metode, yaitu *library* TensorFlow dan DCGAN. Skala augmentasi data yang digunakan bervariasi, mulai dari 1 kali lipat hingga 4 kali lipat. Skala 1 kali lipat (pengujian ke-1) memperbanyak jumlah gambar asli menjadi 120 gambar; skala 2 kali lipat (pengujian ke-2) memperbanyak gambar asli menjadi 240 gambar; skala 3 kali lipat (pengujian ke-3) memperbanyak jumlah gambar asli menjadi 360 gambar; dan skala 4 kali lipat (pengujian ke-4) memperbanyak jumlah gambar asli menjadi 480 gambar.

### 3.4 Hasil Pengujian FID

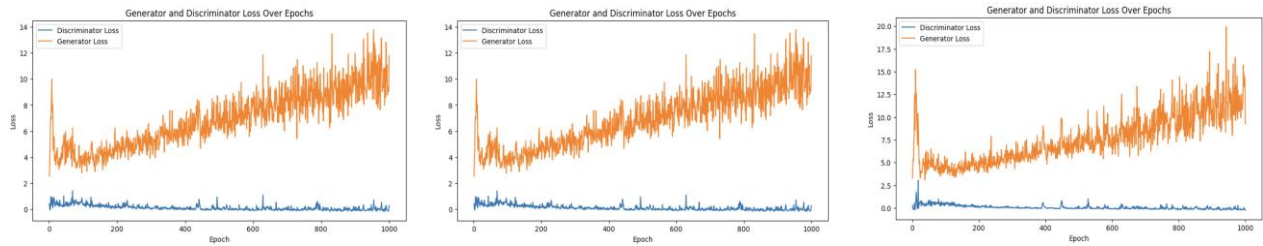
Pengujian FID (*Fréchet Inception Distance*) dilakukan untuk mengevaluasi kualitas gambar yang dihasilkan oleh model berdasarkan beberapa kombinasi parameter. Model dengan nilai FID terendah menghasilkan gambar yang paling mendekati dataset asli [17]. Hasil terbaik diperoleh dengan menggunakan arsitektur di pengujian ke-7 yaitu generator tanpa *batch normalization*, penerapan dropout pada discriminator, serta pengaturan *learning rate* yang berbeda antara generator dan discriminator [20]. Selain itu, penggunaan batch size kecil juga berkontribusi positif, terutama karena jumlah data yang relatif sedikit [21]. Pengujian ini dilakukan pada tiga kelas penyakit dalam dataset, yaitu *Leaf Smut*, *Brown Spot*, dan *Bacterial Leaf Blight*. Hasilnya menunjukkan bahwa konfigurasi ini memberikan kinerja terbaik dalam mereplikasi gambar yang serupa dengan data asli, seperti yang ditunjukkan pada Gambar 7.



**Gambar 7.** Citra Baru Penyakit *Leaf Smut*, *Brown Spot*, dan *Bacterial Leaf Blight* Pengujian ke-7

Grafik loss pada pengujian terbaik ini memperlihatkan bahwa *generator loss* meningkat secara bertahap selama proses pelatihan. Hal ini menunjukkan bahwa model generator semakin sulit menghasilkan citra yang realistis seiring dengan peningkatan kualitas discriminator. Sementara itu, *discriminator loss* tetap relatif stabil dan rendah, menandakan bahwa discriminator mampu dengan efektif membedakan antara citra asli dan hasil generator sepanjang pelatihan seperti yang ditunjukkan Gambar 8.





**Gambar 8.** Grafik Loss Penyakit *Leaf Smut*, *Brown Spot*, dan *Bacterial Leaf Blight* Pengujian ke-7

### 3.5 Hasil Pengujian Klasifikasi

Pengujian klasifikasi ini bertujuan untuk mengevaluasi performa model dalam mendeteksi penyakit daun padi dengan menggunakan kombinasi metode augmentasi data dan teknik klasifikasi. Beberapa pengujian dilakukan dengan variasi jumlah data yang dihasilkan oleh metode augmentasi dari TensorFlow dan DCGAN. Setiap pengujian melibatkan total data uji sebanyak 30 gambar, yang terdiri dari 10 gambar untuk masing-masing penyakit. Hasil terbaik ditunjukkan oleh *confusion matrix* dari model yang menggunakan augmentasi data DCGAN dengan menghasilkan sebanyak 360 gambar baru, ditunjukkan pada Tabel 3, di mana akurasi tertinggi sebesar 93% tercapai. Model ini berhasil mengklasifikasikan sebagian besar gambar dengan benar, yang menunjukkan bahwa penggunaan augmentasi data DCGAN secara signifikan meningkatkan performa klasifikasi.

**Tabel 3.** *Confussion Matrix* Pengujian ke-3 Klasifikasi dengan Augmentasi DCGAN

	Bacterial leaf blight	Brown spot	Leaf smut
Bacterial leaf blight	10	0	0
Brown spot	0	9	1
Leaf smut	1	0	9

### 3.6 Pembahasan

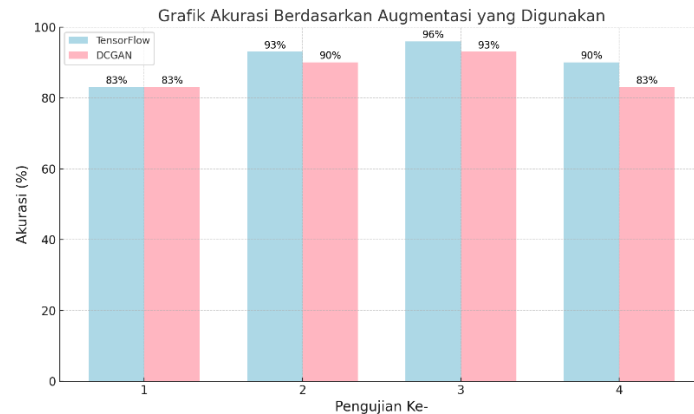
Hasil implementasi metode DCGAN pada dataset terbatas, yaitu 40 gambar per kelas penyakit, menunjukkan kemampuan menghasilkan gambar sintesis yang cukup baik secara visual yang ditunjukkan oleh Gambar 7, meskipun terdapat beberapa kendala. Selanjutnya, seperti yang ditunjukkan pada Gambar 4 hingga Gambar 6 diatas, pada pengujian ke-7 yang divisualisasikan dengan warna pink tua, pengaturan arsitektur generator tanpa *batch normalization*, penerapan dropout pada discriminator, serta pengaturan learning rate yang berbeda antara generator dan discriminator, berkontribusi terhadap hasil yang diperoleh [22]. Penggunaan *batch size* yang kecil juga memberikan kontribusi positif, terutama mengingat jumlah data yang terbatas [23].

Secara ideal, dalam pelatihan GAN, baik *generator loss* maupun *discriminator loss* seharusnya menurun dan mendekati satu sama lain, yang menandakan bahwa generator semakin mampu menghasilkan gambar yang realistis, sementara discriminator semakin sulit membedakan antara gambar asli dan sintesis [12], [20]. Namun, dalam pengujian ke-7 ini yang ditunjukkan oleh Gambar 8, *generator loss* mengalami peningkatan bertahap, sementara *discriminator loss* tetap stabil dan rendah, menunjukkan adanya ketidakseimbangan dalam proses pelatihan.

Meskipun gambar sintesis yang dihasilkan cukup baik secara visual, skor *Fréchet Inception Distance* (FID) yang diperoleh masih cukup tinggi, yaitu sekitar 300 untuk ketiga kelas penyakit, yang mengindikasikan bahwa gambar yang dihasilkan belum sepenuhnya menyerupai dataset aslinya [22]. Skor FID yang tinggi ini berdampak pada akurasi klasifikasi, sehingga performa klasifikasi dengan DCGAN lebih rendah dibandingkan dengan metode augmentasi data menggunakan *library* TensorFlow. Pada metode augmentasi menggunakan TensorFlow, diterapkan berbagai transformasi seperti transformasi skala, pembalikan horizontal dan vertikal, rotasi, pergeseran lebar dan tinggi, serta pembesaran gambar.

Hasil rekapitulasi akurasi untuk setiap pengujian disajikan dalam Gambar 9, di mana model menunjukkan performa terbaik ketika pengujian ke-3 klasifikasi yaitu jumlah gambar yang diaugmentasi mencapai 360 gambar, atau tiga kali lipat dari jumlah gambar aslinya.





**Gambar 9.** Rekapitulasi Akurasi Tiap Pengujian

Hasil pengujian klasifikasi menunjukkan bahwa model mencapai akurasi terbaik sebesar 96% menggunakan metode augmentasi TensorFlow, sementara metode DCGAN mencapai akurasi hingga 93%. Hasil ini mengindikasikan bahwa model bekerja dengan baik dalam kondisi augmentasi data, namun masih terdapat ruang untuk perbaikan, terutama dalam mengoptimalkan skor FID untuk meningkatkan performa keseluruhan model.

#### 4. PENUTUP

Penelitian ini berhasil menerapkan metode augmentasi data menggunakan *Deep Convolutional Generative Adversarial Networks* (DCGAN) untuk meningkatkan jumlah dataset penyakit daun padi, yang kemudian digunakan dalam kombinasi dengan *transfer learning* menggunakan GoogleNet. Meskipun gambar sintesis yang dihasilkan oleh DCGAN secara visual cukup baik, skor *Fréchet Inception Distance* (FID) yang tinggi mengindikasikan bahwa gambar tersebut masih kurang mirip dengan dataset asli, yang berpengaruh pada performa klasifikasi. Namun, kombinasi augmentasi DCGAN dan *transfer learning* terbukti mampu meningkatkan akurasi deteksi penyakit daun padi hingga 93%, menunjukkan potensi yang signifikan meskipun kualitas gambar sintesis belum optimal. Untuk penelitian selanjutnya, peningkatan kualitas gambar sintesis melalui penyesuaian arsitektur dan parameter DCGAN serta eksplorasi metode *Generative Adversarial Networks* (GAN) lainnya disarankan. Penggunaan dataset yang lebih besar dan beragam juga diharapkan dapat meningkatkan robusta model dan akurasi deteksi secara keseluruhan.

#### DAFTAR PUSTAKA

- [1] S. A. Burhan, D. S. Minhas, D. A. Tariq, and M. Nabeel Hassan, "Comparative Study of Deep Learning Algorithms for Disease and Pest Detection in Rice Crops," in *Proceedings of the 12th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2020*, Institute of Electrical and Electronics Engineers Inc., Jun. 2020. doi: 10.1109/ECAI50035.2020.9223239.
- [2] A. Julianto and A. Sunyoto, "A performance evaluation of convolutional neural network architecture for classification of rice leaf disease," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 4, pp. 1069-1078, Dec. 2021, doi: 10.11591/ijai.v10.i4.pp1069-1078.
- [3] Q. Wu, Y. Chen, and J. Meng, "Dcgan-based data augmentation for tomato leaf disease identification," *IEEE Access*, vol. 8, pp. 98716-98728, 2020, doi: 10.1109/ACCESS.2020.2997001.
- [4] G. Yang, G. Chen, C. Li, J. Fu, Y. Guo, and H. Liang, "Convolutional Rebalancing Network for the Classification of Large Imbalanced Rice Pest and Disease Datasets in the Field," *Front Plant Sci*, vol. 12, Jul. 2021, doi: 10.3389/fpls.2021.671134.
- [5] B. Talukdar, "Handling of Class Imbalance for Plant Disease Classification with Variants of GANs," in *2020 IEEE 15th International Conference on Industrial and Information Systems, ICIIS 2020 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 466-471. doi: 10.1109/ICIIS51140.2020.9342728.

- [6] I. Maliki and A. S. Prayoga, "Implementation of Convolutional Neural Network for Sundanese Script Handwriting Recognition with Data Augmentation," *Journal of Engineering Science and Technology*, vol. 18, no. 2, pp. 1113-1123, Apr. 2023
- [7] D. M. Hakim and E. Rainarli, "Convolutional Neural Network untuk Pengenalan Citra Notasi Musik," *Techno.COM*, vol. 18, no. 3, pp. 214-226, Aug. 2019, doi: 10.33633/tc.v18i3.2387
- [8] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Comput Electron Agric*, vol. 153, pp. 46-53, Oct. 2018, doi: 10.1016/j.compag.2018.08.013.
- [9] Y. Lu, X. Tao, N. Zeng, J. Du, and R. Shang, "Enhanced CNN Classification Capability for Small Rice Disease Datasets Using Progressive WGAN-GP: Algorithms and Applications," *Remote Sens (Basel)*, vol. 15, no. 7, Apr. 2023, doi: 10.3390/rs15071789.
- [10] J. Wang and L. Perez, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *arXiv preprint arXiv:1712.04621*, Dec. 2017. Available: <https://arxiv.org/abs/1712.04621>.
- [11] L. Taylor and G. Nitschke, "Improving Deep Learning using Generic Data Augmentation," *arXiv preprint arXiv:1708.06020*, Aug. 2017. Available: <https://arxiv.org/abs/1708.06020>.
- [12] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," presented at the International Conference on Learning Representations (ICLR), 2016. [Online]. Available: [arXiv:1511.06434](https://arxiv.org/abs/1511.06434).
- [13] F. Zhu, M. He, and Z. Zheng, "Data augmentation using improved cDCGAN for plant vigor rating," *Computers and Electronics in Agriculture*, vol. 175, p. 105603, 2020. Available: <https://doi.org/10.1016/j.compag.2020.105603>
- [14] Patmawati, A. Sunyoto, and E. T. Luthfi, "Augmentasi Data Menggunakan DCGAN Pada Gambar Tanah," *TEKNIMEDIA*, vol. 4, no. 1, pp. 45-52, Jun. 2023.
- [15] Vbookshelf, "Rice Leaf Diseases," *Kaggle*, 2020. [Online]. Available: <https://www.kaggle.com/datasets/vbookshelf/rice-leaf-diseases/data>.
- [16] R.-G. Zhou and C. Wan, "Quantum Image Scaling Based on Bilinear Interpolation with Decimals Scaling Ratio," *International Journal of Theoretical Physics*, vol. 60, no. 7, pp. 2159-2174, Jul. 2021, doi: 10.1007/s10773-021-04829-6.
- [17] O. Rukundo, "Effects of Improved Floor Function on the Accuracy of Bilinear Interpolation Algorithm," *Computer and Information Science*, vol. 8, no. 4, pp. 1-11, 2015, doi: 10.5539/cis.v8n4p1.
- [18] X. Mou, T. Zhu, and X. Zhou, "Visible-Image-Assisted Nonuniformity Correction of Infrared Images Using the GAN with SEBlock," *Sensors*, vol. 23, no. 6, p. 3282, Mar. 2023, doi: 10.3390/s23063282.
- [19] S. Kora Venu and S. Ravula, "Evaluation of Deep Convolutional Generative Adversarial Networks for Data Augmentation of Chest X-ray Images," *Future Internet*, vol. 13, no. 1, p. 8, Jan. 2021, doi: 10.3390/fi13010008.
- [20] I. J. Goodfellow et al., "Generative Adversarial Nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, Montreal, QC, Canada, Dec. 2014, pp. 2672-2680.
- [21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1-9.
- [22] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S., "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *arXiv preprint arXiv:1706.08500*, Jan. 2018. [Online]. Available: <https://arxiv.org/abs/1706.08500>.
- [23] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *arXiv preprint arXiv:1812.04948v3*, Mar. 2019. [Online]. Available: <https://arxiv.org/abs/1812.04948v3>.