Klasifikasi Gejala Defisiensi Nutrisi Pada Tanaman Padi Menggunakan CNN Dengan Arsitektur Googlenet

1st Indah Mutiah Utami. MZ
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
indahmutiahmz@student.telkomuniversity.ac.id

2nd Syamsul Rizal
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
syamsul@telkomuniversity.co.id

3rd Nor Kumalasari Caecar Pratiwi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
caecarnkcp@telkomuniversity.ac.id

Abstrak—Tanaman padi merupakan sumber kehidupan bagi manusia, tidak hanya sebagai kebutuhan pangan, padi juga menjadi objek mata pencaharian bagi petani. Defisiensi nutrisi atau kekurangan nutrisi juga sering terjadi pada tanaman padi, sehingga mempengaruhi tingkat kualitas produksi. Defisiensi nutrisi pada umumnya dapat dilihat dari warna dan bentuk daun yang tidak sehat, oleh karena itu dapat dideteksi sejak dini untuk mengurangi gejala kekurangan nutrisi pada tanaman padi. Penelitian ini, mengklasifikasikan gejala defisiensi nutrisi pada tanaman padi dengan menggunakan *Convolutional Neural Network (CNN)* arsitektur GoogleNet berbasis pengolahan citra. Terdapat 1156 citra dengan dataset bersumber dari Kaggle yang terbagi menjadi tiga kelas yaitu defisiensi Nitrogen(N), Fosfor(P), dan Kalium (K). Parameter yang dianalisis pada penelitian ini adalah akurasi, loss, presisi, recall, dan *F1-Score*. Tiap pengujian, dilakukan lima skenario pengujian terhadap *hyperparameter* berupa *optimizer*, *learning rate*, *batch size*, *input size*, serta pengujian terhadap citra asli dan CLAHE. Dari pengujian yang telah dilakukan didapatkan hasil terbaik dengan optimizer yang digunakan Adam, learning rate bernilai 0.001, *batch size* 64,dan *input size* 512 × 512. Dari konfigurasi tersebut didapatkan *test accuracy* sebesar 93.94% dengan testing loss yaitu 0.2370.

Kata Kunci—defisiensi nutrisi, padi, CNN ,googlenet, klasifikasi.

I. PENDAHULUAN

Padi merupakan salah satu tanaman yang paling banyak dikonsumsi oleh masyarakat Indonesia. Tidak hanya sebagai kebutuhan pangan, padi juga menjadi salah satu objek mata pencaharian bagi para petani. Produksi padi sangat berperan dalam perekonomian Indonesia yang dikenal sebagai negara agraris. Berdasarkan Badan Pusat Statistik menyatakan bahwa luas panen padi pada tahun 2020 mengalami penurunan dibandingkan pada tahun 2019 sebesar 0,19% [1]. Salah satu faktor yang mempengaruhi kualitas produksi tanaman padi adalah kualitas unsur hara yang terdapat pada pupuk seperti Nitrogen (N), Fosfor (P), dan Kalium (K) yang paling banyak dibutuhkan dalam jumlah besar [2]. Defisiensi nutrisi atau kekurangan nutrisi sering terjadi pada tanaman padi yang mengakibatkan kualitas produksi pada tanaman padi menurun. Hal itu terjadi dikarenakan praktik pemupukan yang tidak ilmiah sering gagal untuk memenuhi kebutuhan nutrisi tanaman yang sedang tumbuh [3] dan prediksi defisiensi nutrisi yang dilakukan masih manual sehingga memberikan hasil yang tidak akurat [4]. Selain itu, mendiagnosis defisiensi nutrisi membutuhkan teknik dengan biaya tenaga kerja yang besar dan terbatasnya laboratorium [3]. Maka dari itu, memperhatikan nutrisi pada tanaman padi merupakan hal yang sangat penting untuk masa depan pertanian.

Sebuah teknik pembelajaran mesin*(machine learning)* diciptakan untuk membantu manusia dalam pengolahan data. Teknik ini terus berkembang hingga pada tahun 2016, telah muncul suatu teknik pembelajaran baru yang disebut dengan *deep learning*, dengan kemampuan *feature engineering* yang dapat merekayasa fitur secara otomatis dan mampu memberikan peningkatan akurasi yang sebanding dengan penambahan jumlah data [5]. Pada penelitian yang dilakukan oleh Zhe Xu , Xi Guo, Anfan Zhu, Xiaolin He, Xiaomin Zhao , Yi Han, and Roshan Subedi yang menggunakan *DCNN(Deep Convolutional Neural Network)* dengan arsitektur Inception-v3, ResNet50 layer, NasNet-Large, DenseNet121 layer untuk mendiagnosis defisiensi nutrisi pada tanaman padi mendapatkan rata-rata hasil akurasi lebih dari 90% dengan hasil tertinggi yaitu DenseNet121 sekitar 97% [3]. Namun, pada penelitian tersebut menggunakan DenseNet121 memiliki layer yang banyak sehingga dibutuhkan komputasi yang kompleks. Pada penelitian [6] khusus membahas defisiensi nitrogen(N) pada tanaman padi, menggunakan CNN dengan arsitektur yang digunakan adalah ResNet18, ResNet50, GoogleNet, VGG-16, dan VGG-19 serta penambahan *classifier* seperti *Support Vector Machine (SVM).* Pada penelitian tersebut hasil terbaik didapatkan dari arsitektur ResNet50+SVM dengan akurasi 99.84%.

Berdasarkan penelitian sebelumnya, klasifikasi defisiensi nutrisi pada tanaman padi masih terbatas, dimana pada penelitian [3] memiliki layer yang sangat banyak untuk mendapatkan akurasi yang baik sehingga dibutuhkan komputasi yang kompleks, sedangkan pada penelitian [6] khusus membahas defisiensi nitrogen (N) saja. Maka dari itu, pada penelitian ini mengusulkan klasifikasi defisiensi nutrisi pada tanaman padi menggunakan *Convolutional Neural Network (CNN)* arsitektur GoogleNet dengan parameter yang dianalisis yaitu akurasi, *loss*, presisi, *recall*, dan *F1-Score*. Penulis berharap penelitian ini dapat berguna untuk

membantu para petani dan ahli pertanian dalam mengklasifikasi defisiensi nutrisi pada tanaman padi.

II. KAJIAN TEORI

A. Defisiensi Nutrisi

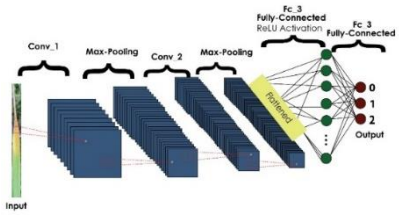
Defisiensi nutrisi pada tanaman padi dapat terjadi ketika tanaman gagal tumbuh meskipun persiapan tanah, penyiraman dan pemberian mulsa tercukupi. Tanaman padi yang mengalami defisiensi nutrisi kondisi pertumbuhannya buruk karena tidak dapat mengambil nutrisi yang terdapat dalam pupuk. Defisiensi nutrisi menyebabkan beberapa gejala pada daun seperti daun menguning atau kecoklatan, pertumbuhan kerdil dan terhambat, pembungaan dan pembuahan buruk [7]. Adapun unsur hara yang sangat berpengaruh dalam pertumbuhan tanaman padi adalah Nitrogen (N), Fosfor(P), Kalium(K).

B. Pengolahan Citra

Citra merupakan gambar dua dimensi yang dapat dilihat secara visual oleh manusia [10]. Pada penelitian ini *dataset* yang digunakan merupakan citra digital dengan resolusi 300 *dpi(dots per inch)*. Pengolahan citra digital merupakan pemrosesan citra digital dengan menggunakan operasi-operasi pemrosesan sinyal, seperti perbaikan atau modifikasi citra. Tujuan dari pengolahan citra adalah untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin [10].

C. Convolutional Neural Network(CNN)

Algoritma *Convolutional Neural Network (CNN)* merupakan algoritma *Deep Learning* yang pertama kali ditemukan pada tahun 1990-an oleh Yann LeCun dengan *paper* berjudul *"Gradient-Based Learning Applied to Document Recognition"*[11]. Secara umum CNN (lihat Gambar 1) memiliki tiga jenis layer utama yaitu *Convolutional Layer, Pooling Layer, dan Fully Connected Layer*. *Convolutional layer* adalah blok bangunan inti dari CNN yang bertujuan untuk mempelajari representasi fitur dari input dimana pada layer ini terdiri dari beberapa kernel atau filter yang digunakan untuk menghitung *feature maps*. *Feature map* baru dihasilkan dengan menggabungkan input dengan kernel dan menerapkan fungsi aktivasi seperti Sigmoid, ReLu, dll pada hasil konvolusi. Pooling layer mengambil wilayah kecil dari output konvolusi sebagai input dan melanjutkannya untuk menghasilkan satu output. Ada berbagai teknik pooling layer seperti max-pooling, min-pooling, average pooling, dll. Pooling layer mengurangi jumlah parameter yang akan dihitung serta membuat terjemahan jaringan menjadi invarian. Bagian terakhir dari CNN pada dasarnya terdiri dari satu atau lebih *Fully Connected Layer*. *Fully Connected Layer* mengambil input dari pooling layer dan menghasilkan output [12].



GAMBAR 1 Arsitektur CNN

D. GoogleNet

GoogLeNet merupakan realisasi dari arsitektur *inception* yang digunakan pada kompetisi *ImageNet Scale Visual Recognition Challenge(ILSVRC) 2014.* Jaringan pada *GoogleNet* dilatih dengan metode pengambilan sampel gambar yang berbeda dengan menggunakan 6 model. Semua proses konvolusi, termasuk yang ada dalam *inception*, menggunakan aktivasi linier. Arsitektur GoogleNet terdiri dari 22 layer (27 layer termasuk *pooling layer)* dengan total sembilan modul *inception* [15].

III. METODE

A. Dataset

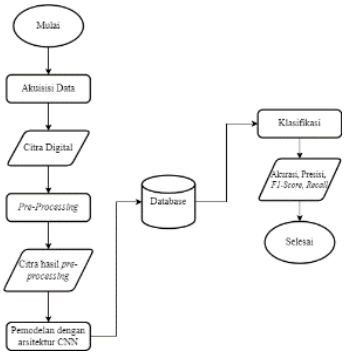
(a) (b) (c)

GAMBAR 2 (a) Dataset N, (b) Dataset P, (c) Dataset K

Pada penelitian ini menggunakan *dataset* dari *Kaggle* yang terdiri dari 1156 citra dengan resolusi 300 dpi, berformat .jpg, dan RGB. Dataset ini terbagi 3 kelas citra yaitu defisiensi Nitrogen (N) yang berjumlah 440 citra, Fosfor(P) 333 citra, dan Kalium (K) 383 citra. Citra tersebut diambil menggunakan kamera iPhone 11. Data latih merupakan sekumpulan data citra yang digunakan untuk melatih pola citra agar mampu dipelajari oleh sistem. Data uji merupakan data citra yang digunakan untuk menguji performansi dari model yang digunakan. Sedangkan data validasi digunakan untuk proses validasi model dalam meningkatkan performansi dan mencegah terjadinya *overfitting.*

B. Desain Sistem

Penelitian ini dirancang sistem klasifikasi defisiensi nutrisi yang terdiri dari tiga kelas yaitu defisiensi N, P, K menggunakan CNN dengan arsitektur GoogleNet. Alur kerja pada sistem ini dapat dilihat dari diagram alir sebagai berikut.



GAMBAR 3 *Flowchart* proses *Training dan Testing*

Pada gambar 3 merupakan gambaran umum dari proses klasifikasi, dimana terdapat tiga tahapan utama sebelum lanjut ke proses *training data* dan *testing data* yaitu tahap akuisisi data, *pre-processing*, dan pemodelan dengan menggunakan arsitektur CNN yaitu GoogleNet. Akuisisi data merupakan proses pengumpulan dataset yang bersumber dari Kaggle yang terdiri dari tiga kelas yaitu defisiensi N, P, dan K dengan total 1156 citra RGB. Pada penelitian ini, dilakukan skenario pengujian dengan data *balance* dan *unbalance*. Diketahui bahwa jumlah data unbalance dengan kelas defisiensi Nitrogen (N) berjumlah 440 citra, Fosfor(P) 333 citra, dan Kalium (K) 383 citra.

Kemudian dilakukan *pre-processing* untuk mengubah citra dari dataset menjadi citra yang siap diolah. Di tahap ini, untuk mendapatkan dataset yang *balance* dilakukan *pre-processing* dengan augmentasi data. Selanjutnya, data masuk ke proses pemodelan dengan arsitektur GoogleNet yang memiliki 22 layer (27 layer termasuk *pooling layer)* dengan total sembilan modul *inception*. Setelah itu, proses selanjutnya yaitu klasifikasi yang akan menghasilkan parameter performansi seperti akurasi, *loss*, presisi, *recall*, dan *F1-Score.*

C. Pre-Processing

Pre-Processing merupakan proses awal yang dilakukan dengan mengubah citra dari dataset menjadi citra yang siap diolah. Tujuan dari proses ini adalah agar mendapatkan data citra yang seragam baik dari kualitas maupun ukuran. Adapun beberapa proses *pre-processing* yang dilakukan adalah *resize*, normalisasi, *data augmentation*, dan CLAHE. *Contrast Limited Adaptive Histogram Equalization* (CLAHE) merupakan teknik yang digunakan untuk meningkatkan tingkat visibilitas citra berupa kontras [17]. *Data augmentation* adalah teknik yang digunakan untuk memanipulasi data citra dengan melakukan penambahan atau pengurangan jika diketahui bahwa data yang dimiliki *unbalance*. *Data augmentation* dapat dilakukan dengan *crop, flip, rotate*, dll. Namun pada penelitian ini, penulis hanya menggunakan *horizontal flip.*

Pada *pre-processing* pembagian *dataset* dilakukan antara *dataset unbalance* dan *balance.* Pada dataset *unbalance*, dapat dilihat pada tabel 1. Sedangkan Pembagian data *balance* bisa dilihat pada tabel 2. Pada data *balance* dilakukan *oversampling dataset* ke jumlah data paling banyak yaitu 440.

TABEL 1 Pembagian Dataset *Unbalance*

Data Latih 80%	Data Uji 10%	Data Validasi 10%
924	117	115

TABEL 2 Pembagian Dataset *Balance*

Data Latih 70%	Data Uji 20%	Data Validasi 10%
924	264	132

D. Parameter Performansi

Parameter performansi merupakan tolak ukur dalam menentukan seberapa bagus kualitas dari sistem yang telah dirancang. Adapun parameter yang diukur dalam penelitian ini adalah *loss*, akurasi (lihat persamaan 3.1), presisi(lihat persamaan 3.2), *recall*(lihat persamaan 3.3), dan *F1-Score* (lihat persamaan 3.4) [18].

$$Akurasi = \frac{(TP + TN)}{TP + FP + FN + TN} \times 100 \quad (3.1)$$

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3.3)$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100\% \quad (3.4)$$

IV. HASIL DAN PEMBAHASAN

A. Skenario I: *Optimizer*

Pengujian skenario pertama merupakan pengujian dengan membandingkan pengaruh dari empat *optimizer* yaitu Adam, Adamax, Nadam, dan SGD pada citra asli. Nilai *hyperparameter* lain seperti *learning rate*, *batch size*, dan *input size* bernilai sama di masing-masing pengujian *optimizer*. *Learning rate* bernilai 0.001, *batch size* yang dipakai yaitu 32, dan *input size* berukuran 224 × 224.

TABEL 3 **.** Hasil Pengujian *Optimizer* Data *Unbalance*

Optimizer	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
Adam	90.58%	0.2397	91.30%	0.2916	94.02%	0.2322
Adamax	91.56%	0.2479	89.57%	0.3179	94.02%	0.2461
Nadam	92.75%	0.2059	88.70%	0.2885	91.45%	0.2447
SGD	82.90%	0.5407	81.74%	0.5959	84.62%	0.5874

TABEL 4**.** Hasil Pengujian *Optimizer* Data *Balance*

Optimizer	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
Adam	92.55%	0.2209	92.42%	0.2133	88.26%	0.3073
AdaMax	89.42%	0.3193	85.61%	0.3632	82.20%	0.4135
Nadam	90.82%	0.246	92.42%	0.2409	87.50%	0.3272
SGD	84.13%	0.5144	84.85%	0.5485	79.92%	0.5907

Berdasarkan hasil yang didapatkan pada tabel 3 dan 4 , dapat dilihat bahwa didapatkan hasil *optimizer* terbaik yaitu Adam pada data *unbalance* yang ditunjukkan dengan hasil *test accuracy* mencapai 94.02% dengan *loss* 0.2322. Dengan hasil tersebut, *optimizer* Adam merupakan *optimizer* yang cocok digunakan untuk skenario pengujian selanjutnya.

B. Skenario II: *Learning Rate*

Setelah mendapatkan hasil terbaik dari *optimizer* di pengujian sebelumnya yaitu Adam, maka selanjutnya adalah menguji pengaruh *learning rate* terhadap parameter performansi dengan menggunakan citra asli. Pada pengujian ini *learning*

rate yang digunakan dimulai dari nilai terbesar yaitu 0.1 sampai nilai terkecil yaitu 0.0001. Nilai *hyperparameter* lain seperti *batch size* digunakan ukuran 32, dan *input size* berukuran 224 × 224.

TABEL 5. Hasil Pengujian *Learning Rate* Data *Unbalance*

Learning Rate	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
0.1	87.88%	1.0731	86.96%	1.5797	92.31%	0.5474
0.01	90.69%	0.2447	86.09%	0.3263	88.89%	0.2754
0.001	90.58%	0.2397	91.30%	0.2916	94.02%	0.2322
0.0001	88.53%	0.39	86.09%	0.4543	85.47%	0.41

TABEL 6. Hasil Pengujian *Learning Rate* Data *Balance*

Learning Rate	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
0.1	91.58%	0.4094	89.39%	0.5708	83.71%	1.0372
0.01	91.36%	0.2122	93.94%	0.1900	86.36%	0.3571
0.001	92.55%	0.2209	92.42%	0.2133	88.26%	0.3073
0.0001	88.23%	0.3753	84.85%	0.4072	81.44%	0.4557

Berdasarkan hasil yang didapatkan pada tabel 5 dan 6, bahwa hasil pengujian dengan *optimizer* Adam, mendapatkan *learning rate* terbaik yaitu 0.001 baik data *unbalance*, maupun *balance*. Namun pengujian terbaik didapatkan dari data *unbalance* yang ditunjukkan dengan tingkat *test accuracy* mencapai 94.02% dengan *loss* 0.2322. Disini terbukti bahwa *learning rate* yang terlalu tinggi akan menyebabkan *loss* yang besar. Sedangkan, *learning rate* yang terlalu rendah akan membuat pembaruan bobot pada jaringan yang kecil dan memperlambat proses *training*.

C. Skenario III: *Batch Size*

Diketahui hasil terbaik yang didapatkan pada pengujian sebelumnya yaitu *optimizer* Adam dengan *learning rate* 0.001. Selanjutnya pada pengujian ini, sistem diuji dengan *batch size* 16, 32, dan 64 menggunakan citra asli. Sedangkan *hyperparameter* lain yang ditetapkan selain *optimizer* dan *learning rate*, ada juga *input size* dengan ukuran 224 × 224.

TABEL 7. Hasil Pengujian *Batch Size* Data *Unbalance*

Batch Size	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
16	92.42%	0.2135	89.57%	0.298	92.31%	0.2316
32	90.58%	0.2397	91.30%	0.2916	94.02%	0.2322
64	91.67	0.237	86.96%	0.3233	92.31%	0.2633

TABEL 8. Hasil Pengujian *Batch Size* Data *Balance*

Batch Size	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
16	92.22%	0.2076	93.18%	0.222	85.61%	0.3256
32	92.55%	0.2209	92.42%	0.2133	88.26%	0.3073
64	93.63%	0.1784	94.70%	0.1948	89.39%	0.2902

Berdasarkan hasil yang didapatkan pada tabel 7 dan 8, bahwa hasil pengujian yang didapatkan sama seperti pengujian sebelumnya dengan *optimizer* Adam, *learning rate* bernilai 0.001 dan *batch size* 32 merupakan hasil terbaik baik dari data *unbalance* dengan *test accuracy* 94.02% dengan *test loss* 0.2322. *Batch size* 64 dari data *balance* merupakan hasil terbaik dengan *test accuracy* 94.02% dengan *test loss* 0.2322. Dari table tersebut terlihat bahwa data *unbalance* mempunyai hasil yang lebih tinggi dibanding data *balance.*

D. Skenario IV: *Input Size*

Pengujian *input size* merupakan pengujian *hyperparameter* terakhir sebelum dilakukan pengujian pengaruh data CLAHE terhadap citra asli. Pada pengujian ini *hyperparameter* yang ditetapkan yaitu *optimizer* Adam dengan *learning rate* 0.001 dan *batch size* yang digunakan yaitu 32. Pengujian ini dilakukan dengan menggunakan citra asli. Nilai *input size* yang akan dipakai pada pengujian ini adalah *input size* dengan nilai 224×224, 256×256, dan 512×512.

TABEL 9. Hasil Pengujian *Input Size* Data *Unbalance*

Input Size	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
224×224	93.63%	0.1784	94.70%	0.1948	89.39%	0.2902
256×256	94.60%	0.1828	94.70%	0.1806	89.39%	0.2594
512×512	94.71%	0.1960	96.97%	0.1896	93.94%	0.2370

TABEL 10. Hasil Pengujian *Input Size* Data *Balance*

Input Size	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
224×224	90.58%	0.2397	91.30%	0.2916	94.02%	0.2322
256×256	92.97%	0.2118	91.30%	0.3181	94.87%	0.2284
512×512	93.51%	0.2117	91.30%	0.2454	93.16%	0.2243

Berdasarkan hasil yang didapatkan pada tabel 9 dan 10, hasil pengujian dengan *input size* 256×256 merupakan hasil terbaik dari data *unbalance* yang ditunjukkan dengan tingkat *test accuracy* mencapai 94.87% dan *test loss* 0.2284. Sedangkan pada data *balance*, hasil pengujian dengan *input size* 512×512 merupakan hasil terbaik yang ditunjukkan dengan tingkat *test accuracy* mencapai 93.94% dan *test loss* 0.2370. Pada pengujian ini, *input size* memberikan dampak yang baik pada tingkat akurasi.

E. Skenario V : Perbandingan Citra Asli dan CLAHE

Setelah pengujian dari 4 *hyperparameter* telah selesai diuji, maka pada pengujian tahap ini membandingkan hasil citra asli dengan citra hasil CLAHE. Pengujian ini merupakan pengujian tahap akhir dimana hasil terbaik dari pengujian sebelumnya akan dibandingkan dengan citra yang menggunakan CLAHE, apakah terjadi perubahan yang signifikan atau justru terjadi penurunan akurasi. Berikut adalah hasil dari pengujian data *unbalance* dan *balance.*

TABEL 11. Hasil Pengujian Citra CLAHE Data *Unbalance*

Jenis Citra	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
Citra Asli	92.97%	0.2118	91.30%	0.3181	94.87%	0.2284
Citra CLAHE	91.02%	0.2606	88.70%	0.3329	92.31%	0.2518

TABEL 12. Hasil Pengujian Citra CLAHE Data *Balance*

Jenis Citra	Train Accuracy	Train Loss	Val Accuracy	Val Loss	Test Accuracy	Test Loss
Citra Asli	94.71%	0.1960	96.97%	0.1896	93.94%	0.2370
Citra CLAHE	92.55%	0.2632	88.64%	0.3157	87.50%	0.3373

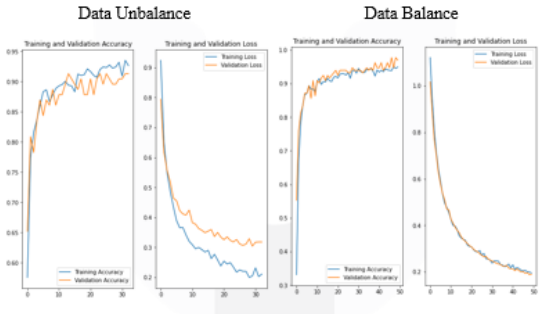
Pada hasil yang didapat pada tabel 11 dan 12, baik pengujian data *unbalance* maupun *balance* diketahui bahwa hasil citra asli masih lebih baik dari pada hasil citra CLAHE. Dimana hasil dengan akurasi yang tinggi didapatkan dari data *unbalance* dengan tingkat *test accuracy* mencapai 94.87% dan *test loss* 0.2284. Dengan hasil tersebut dapat disimpulkan bahwa citra CLAHE tidak memiliki pengaruh yang signifikan terhadap peningkatan akurasi, bahkan nilainya cenderung menurun.

F. Hasil Pengujian Terbaik

Jenis Data	Hyperparameter	Train Acc	Train Loss	Val Acc	Val Loss	Test Acc	Test Loss
Unbalance	Optimizer : Adam Learning rate : 0.001 Batch Size: 32 Input Size: 256 × 256 Citra Asli	92.97 %	0.2118	91.30 %	0.3181	94.87 %	0.2284
Balance	Optimizer : Adam Learning rate : 0.001 Batch Size: 64 Input Size: 512 × 512 Citra Asli	94.71 %	0.1960	96.97 %	0.1896	93.94 %	0.2370

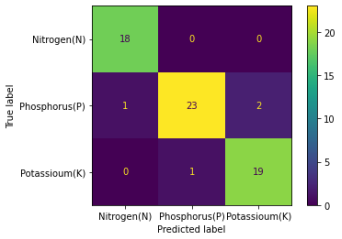
TABEL 13. Hasil Pengujian Terbaik

Hasil pengujian terbaik dapat ditinjau pada Tabel 13. Pada pengujian tersebut dapat dilihat bahwa tingkat akurasi dari data *unbalance* lebih tinggi dari pada data *balance*. Hasil tersebut ditunjukan dengan nilai *test accuracy* pada data *unbalance* mencapai 94.87% dengan *test loss* 0.2284, sedangkan pada data *balance* tingkat *test accuracy* mencapai 93.94% dan *test loss* 0.2370. Meskipun begitu, jika ditinjau dari grafik *training loss* dan *validation loss* pada Gambar 4, dapat dilihat bahwa data *balance* memiliki grafik yang *good fitting* dibanding data *unbalance* yang grafiknya cenderung *overfitting*. Maka dari itu, secara keseluruhan pengujian, hasil terbaik didapatkan dari data *balance*. Data *unbalance* cenderung memiliki grafik yang *overfitting* dari lima pengujian *hyperparameter*, yang mana kondisi *overfitting* dihindari, sebab jika diuji dengan data baru sistem tidak dapat memprediksi dengan baik.



GAMBAR 4. Grafik Perbandingan Data *Unbalance* dan Data *Balance*

Berdasarkan kesimpulan yang didapatkan pada tabel 3 bahwa data *balance* lebih tepat digunakan pada *dataset* yang dipakai di penelitian ini.



GAMBAR 5. *Confusion Matrix best model*

Selain itu, untuk meyakinkan apakah model sudah cukup baik dalam mengklasifikasi, maka pada Gambar 5 dipaparkan hasil dari *confusion matrix* yang menunjukkan bahwa model secara keseluruhan sudah baik dalam mengklasifikasi, meskipun masih terdapat kesalahan prediksi pada kelas fosfor dan kalium. Hal itu ditandai dengan tingkat warna yang semakin terang secara diagonal pada *confusion matrix*.

V. KESIMPULAN

Berdasarkan hasil pengujian dan simulasi yang telah dilakukan, dapat disimpulkan bahwa klasifikasi defisiensi nutrisi pada tanaman padi menggunakan CNN dengan arsitektur GoogleNet, dapat berjalan dengan optimal dalam mengklasifikasikan kelas defisiensi nutrsi seperti Nitrogen(N), Fosfor(P), dan Kalium(K). Hasil terebut berdasarkan pada lima skenario pengujian yang telah dilakukan dengan pembagian data unbalance dan data balance bahwa ditinjau dari nilai akurasi, data unbalance memiliki akurasi yang lebih tinggi dari data balance, yang mana nilai test accuracy pada data unbalance mencapai 94.87% dengan test loss 0.2284, sedangkan pada data balance tingkat test accuracy mencapai 93.94% dan test loss 0.2370. Meskipun begitu, jika ditinjau dari grafik pada Gambar 4 data *balance* memiliki grafik yang good fitting dibandingkan data *unbalance* yang grafiknya cenderung overfitting. Maka dari itu, data balance dengan menggunakan citra asli, optimizer Adam, learning rate 0.001, batch size 64, dan input size 512 × 512 merupakan hasil terbaik yang didapatkan pada penelitian ini.

REFERENSI

[1] Badan Pusat Statistik Kota Bandar Lampung, “Berita Resmi Statistik,” *Bps.Go.Id*, vol. 19, no. 27, pp. 1–16, 2021.

[2] D. Rina, “Manfaat Unsur N, P, dan K Bagi Tanaman,” *BPTP Kaltim*, no. 3, pp. 6–9, 2015.

[3] Z. Xu *et al.*, “Using deep convolutional neural networks for image-based diagnosis of nutrient deficiencies in rice,” *Comput. Intell. Neurosci.*, vol. 2020, 2020.

[4] T. Amirtha, T. Gokulalakshmi, P. Umamaheswari, and T. R. M. Tech, “Machine Learning Based Nutrient Deficiency Detection in Crops,” *Int. J.*

Recent Technol. Eng., vol. 8, no. 6, pp. 5330–5333, 2020.

[5] K. Dr. Suyanto S.T., M.Sc. and P. . Ramadhani, S.T., M.T. , Satria Mandala, *Deep Learning Modernisasi Machine Learning Untuk Big Data*. Bandung, Jawa Barat. Indonesia: Informatika Bandung.

[6] P. K. Sethy, N. K. Barpanda, A. K. Rath, and S. K. Behera, "Nitrogen Deficiency Prediction of Rice Crop Based on Convolutional Neural Network," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5703–5711, 2020.

[7] RHS, "Nutrient deficiencies," 2021. [Online]. Available: https://www.rhs.org.uk/prevention-protection/nutrient-deficiencies. [Accessed: 23-Nov-2021].

[8] H. Yamashita, R. Sonobe, Y. Hirono, A. Morita, and T. Ikka, "Dissection of hyperspectral reflectance to estimate nitrogen and chlorophyll contents in tea leaves based on machine learning algorithms," *Sci. Rep.*, pp. 1–11, 2020.

[9] M. Inti *et al.*, "Studi Pengaruh Pemberian Pupuk Kandang Dan Guano Fosfat Terhadap Serapan Kalium Tanaman Kacang Hijau (Vigna radiata L)," vol. 23, no. 1, pp. 44–52, 2021.

[10] R. Munir, "Pengantar Pengolahan Citra," *Pengolah. Citra Digit.*, no. Bagian 1, pp. 1–10, 2013.

[11] Y. D. Heryadi and T. D. Wahyono, *Dasar-Dasar Deep Learning Dan Impelementasinya*, Cetakan I. Yogyakarta: Penerbit Gava Media, 2021.

[12] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, *Fundamental concepts of convolutional neural network*, vol. 172, no. January. 2019.

[13] IBM Cloud Education, "What are Convolutional Neural Networks?," *Ibm.com*, 2021. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks. [Accessed: 23-Nov-2021].

[14] T. Wood, "Softmax Function Definition," *DeepAI*, 2019. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer.

[15] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, "Preparation of novel high copper ions removal membranes by embedding organosilane-functionalized multi-walled carbon nanotube," *J. Chem. Technol. Biotechnol.*, vol. 91, no. 8, pp. 2322–2330, 2016.

[16] H. A. Prihanditya and A. Alamsyah, "The Implementation of Z-Score Normalization and Boosting Techniques to Increase Accuracy of C4.5 Algorithm in Diagnosing Chronic Kidney Disease," *J. Soft Comput. Explor.*, vol. 1, no. 1, pp. 63–69, 2020.

[17] G. Yadav, Saurabh Maheswari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," *Contrast Ltd. Adapt. histogram Equal. based Enhanc. real time video Syst.*, pp. 2392–2397, 2014.

[18] F. Demir, "Deep autoencoder-based automated brain tumor detection from MRI data," *Artif. Intell. Brain-Computer Interface*, pp. 317–351, Jan. 2022.