



## Deteksi Penyakit Daun Teh Menggunakan Metode *Convolutional Neural Network* (CNN)

Kiki Diah Ayu Puspita<sup>1</sup>, Agung Nilogiri<sup>2</sup>, Hardian Oktavianto<sup>3</sup>

Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember<sup>1,2,3</sup>

Email: kiki.diah52@gmail.com<sup>1\*</sup>, agungnilogiri@unmuhjember.ac.id<sup>2</sup>, hardian@unmuhjember.ac.id<sup>3</sup>

### ABSTRAK

Teh menjadi salah satu minuman paling populer di dunia dan banyak diminati. Indonesia merupakan negara eksportir teh terbesar keenam dunia. Berbagai macam usaha pengolahan dan implementasi secara berkala menemui keterbatasan sehingga tidak dapat memenuhi kebutuhan produksi yang diprediksikan. Salah satu penyebab turunnya produksi teh di Indonesia adalah penyakit pada daun teh, penyakit yang sering ditemui pada daun teh yaitu *Brown Blight* dan penyakit *Alga Leaf Spot*. Untuk itu dibutuhkan sebuah metode yang mampu mendeteksi penyakit pada daun teh salah satunya *Convolutional Neural Network* menggunakan arsitektur VGG-16. Data yang digunakan berjumlah 1500 data terdiri dari 3 kelas yaitu *healthy*, *brown blight*, *alga spot*. Data dibagi menjadi 2 bagian, 1350 data yang digunakan sebagai validasi silang dan 150 data untuk uji akhir. Pengujian akhir menggunakan 150 citra didapatkan akurasi sebesar 97,7%. Kesimpulan dari penelitian ini menunjukkan bahwa algoritma *Convolutional Neural Network* (CNN) yang dibangun relatif mampu mendeteksi jenis penyakit daun teh dengan baik. Nilai sensitivitas lebih kecil dibanding dengan spesifisitas artinya tingkat pendeteksi model penyakit lebih rendah dari tingkat pendeteksi daun sehat untuk mendeteksi citra penyakit daun teh.

**Kata Kunci:** *Convolutional Neural Network*, VGG-16, Penyakit Daun, Akurasi, Sensitivitas, Spesifisitas.

### ABSTRACT

*Tea is one of the most popular drinks in the world and is in great demand. Indonesia is the world's sixth largest exporter. Various processing and implementation efforts periodically encounter limitations so that they cannot meet the predicted production needs. One of the causes of the decline in tea production in Indonesia is tea leaf disease, a disease that is often found in tea leaves, namely Brown Blight and Algae Leaf Spot disease. For this reason, a method is needed that can detect diseases in tea leaves, namely the Convolutional Neural Network using the VGG-16 architecture. The data used amounted to 1500 data consisting of 3 classes, namely healthy, brown blight, spot algae. The data is divided into 2 parts, 1350 data used as cross validation and 150 data for the final test. The final test using 150 images obtained an accuracy of 97.7%. The conclusion of this study shows that the developed Convolutional Neural Network (CNN) algorithm is relatively good at detecting leaf disease. The sensitivity value is smaller than the specificity, meaning that the level of detection of disease models is lower than the level of detection of healthy leaves to detect image of tea leaf disease.*

**Keywords:** *Convolutional Neural Network*, VGG-16, leaf disease, accuracy, sensitivity, specificity.

## 1. PENDAHULUAN

Teh menjadi salah satu minuman paling populer di dunia dan banyak diminati. Selain itu pengetahuan tentang manfaat minum teh bagi kesehatan menjadikan teh sebagai ekspor penting bagi Indonesia. Indonesia sendiri merupakan negara eksportir teh terbesar keenam dunia. Produksi dan pengembangan komoditas teh yang diunggulkan salah satu *core business* perkebunan dengan upaya pemerintah untuk terus mendorong produktivitas perekonomian nasional (Zakariya dkk., 2012). Pengembangan usaha komoditas teh merupakan salah satu kunci keberhasilan dalam meningkatkan daya saing dan keunggulan. Indonesia menghadapi tantangan untuk meningkatkan kualitas tehnya karena permintaan dan persaingan di pasar internasional, salah satu cara untuk melakukannya adalah dengan memilah atau memilih daun teh terbaik.

Berbagai macam usaha pengolahan dan implementasi secara berkala menemui keterbatasan sehingga tidak dapat memenuhi kebutuhan produksi yang diprediksikan. Salah satu penyebab turunnya produksi teh di Indonesia adalah penyakit pada daun teh, penyakit yang sering ditemui pada daun teh yaitu *Brown Blight* disebabkan oleh *Collectrichum Camelliae*, dan penyakit *Alga Leaf Spot* disebabkan oleh *Alga Cephaleuros Virescens*. Untuk memenuhi produksi daun teh diprediksikan

dibutuhkan sistem deteksi penyakit daun teh menggunakan informasi dan memanfaatkan kemajuan teknologi informasi, untuk identifikasi penyakit pada daun teh sehingga diharapkan dapat digunakan oleh petani daun teh agar budidaya daun teh terjaga kualitasnya. Menggunakan *Convolutional Neural Network* adalah salah satu pendekatan yang dapat digunakan untuk deteksi penyakit daun teh. *Convolutional Neural Network* (CNN) adalah jenis jaringan saraf tertentu yang biasanya diterapkan pada data gambar. CNN yang dapat digunakan untuk mendeteksi dan mengidentifikasi objek dalam gambar (Ilahiyah & Nilogiri, 2018). Metode *Convolutional Neural Network* cocok untuk mengenali jenis penyakit pada daun teh sehingga pada penelitian ini model CNN akan digunakan sebagai pendeteksian jenis penyakit pada daun teh.

## 2. KAJIAN PUSTAKA

### A. Citra Digital

Citra digital adalah gambar  $f(x,y)$  dengan koordinat/sampel spasial yang didiskritisasi dan tingkat kuantisasi yang didiskritisasi (pencahayaannya/skala abu-abu). Intensitas cahaya  $f(x,y)$ , di mana  $x$  dan  $y$  adalah koordinat spasial merupakan definisi citra digital. Tingkat kecerahan citra pada setiap titik  $(x,y)$  direpresentasikan oleh nilai fungsi pada titik tersebut. Citra digital adalah matriks, di mana indeks baris dan kolom sesuai dengan titik citra dan elemen matriks, juga dikenal sebagai elemen gambar atau piksel, piksel yang mewakili tingkat keabuan dari titik tersebut (Ilahiyah & Nilogiri, 2018).

### B. *Convolutional Neural Network*

*Convolutional Neural Network* (CNN/ConvNet) adalah algoritma dari *deep learning* yang dirancang untuk memproses data dua dimensi seperti gambar dan audio yang merupakan pengembangan dari *Multilayer Perceptron* (MLP). Menggunakan teknik *supervised learning* CNN digunakan untuk mengklasifikasi data berlabel (Ilahiyah & Nilogiri, 2018). CNN adalah lapisan dengan susunan neuron tiga dimensi (lebar, tinggi, kedalaman). Lebar dan tinggi adalah ukuran lapisan, dan kedalaman adalah jumlah lapisan. CNN adalah konstruksi matematika yang biasanya terdiri dari tiga jenis lapisan (atau blok penyusun) yaitu *convolution*, *pooling*, dan *fully connected layers*. Dua yang pertama, *convolution and pooling layers*, perform *feature extraction*, sedangkan yang ketiga, *fully connected layer*, memetakan *extracted features* menjadi keluaran akhir, seperti klasifikasi (Ilahiyah & Nilogiri, 2018). Arsitektur dari CNN dibagi menjadi 2 bagian besar, *feature learning* dan *classification* (MLP). Adapun tahap dari CNN adalah *convolution* (konvolusi), ReLu (fungsi aktivasi layer) dan *pooling layer* merupakan komponen dari *feature learning*. Namun terkadang ada beberapa studi atau makalah yang tidak menggunakan *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke tahap *classification* (klasifikasi).

*Convolution layer* adalah *filter* yang dikenal sebagai kernel digunakan oleh CNN (*Convolutional Neural Networks*) untuk mengidentifikasi fitur seperti sudut yang ada di seluruh gambar. *Filter* adalah matriks nilai sederhana yang disebut bobot yang telah dilatih untuk mengenali karakteristik tertentu. *Filter* bergerak melalui setiap bagian gambar untuk melihat apakah fitur yang diklaimnya dapat dideteksi tersedia. *Filter* melakukan operasi konvolusi untuk memberikan nilai yang menunjukkan keakuratan fitur yang tersedia (Felix dkk., 2019).

Tujuan *pooling* adalah untuk mengubah fitur gabungan menjadi sesuatu yang lebih berguna dan penting sambil membuang detail yang tidak relevan. Penggunaan *pooling layer* pada CNN bertujuan untuk mencapai berbagai perubahan terhadap perubahan posisi atau kondisi pencahayaan, *lightfastness* dan kohesi representasi (Putri, 2020). Secara umum, *pooling layer* mengenkapsulasi *output* dari kelompok *neuron* yang berdekatan ke dalam peta kernel yang sama. CNN memiliki dua metode *pooling* lainnya, termasuk *max pooling* dan *average pooling*.

*Rectified Linear Unit (ReLU)* adalah fungsi aktivasi komputasi sederhana. Aktivasi *Rectified Linear Unit (ReLU)* adalah lapisan aktivasi pada model CNN yang melakukan *thresholding* pada nilai piksel citra masukan dengan nilai nol menggunakan fungsi  $f(x) = \max(0, x)$ . Saat aktivasi ini terjadi, semua nilai piksel dalam citra yang lebih rendah dari akan dijadikan 0. kondisi if hanya digunakan pada proses *forward* dan *backward* melalui ReLU Jika suatu elemen negatif, nilainya diatur ke 0 dan tidak ada operasi eksponensial, perkalian atau pembagian yang dilakukan. Dengan karakteristik seperti itu menjadikan keunggulan ReLU untuk jaringan dengan banyak neuron, secara signifikan mengurangi waktu pelatihan dan pengujian (Ilahiyah & Nilogiri, 2018).

Bentuk lain dari algoritma regresi logistik yang memungkinkan klasifikasi lebih dari dua kelas adalah *softmax activation*, juga dikenal sebagai *softmax classifier* (Nugroho dkk., 2020). Klasifikasi kelas biner adalah tugas klasifikasi standar yang dilakukan oleh algoritma regresi logistik. Persamaan untuk *softmax* ditunjukkan pada rumus 1.

$$f_i(Z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (1)$$

Keterangan:

$z$  = Persamaan linear fungsi preaktivitas

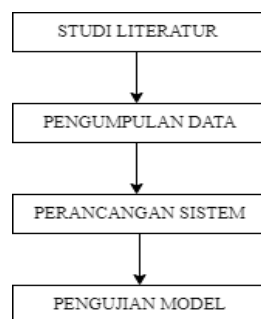
$j$  = Salah satu class dari  $K$  class yang ada

### C. K-Fold Cross Validation

*K-fold Cross Validation* digunakan untuk menghitung kesalahan prediksi dalam mengevaluasi kinerja model. Data dibagi menjadi himpunan bagian  $k$  yang ukurannya hampir sama. Model dalam klasifikasi dilatih dan diuji sebanyak  $k$ . Di setiap pengulangan, salah satu kelompok bagian akan digunakan sebagai data *training* dan data *testing* (Mardiana dkk., 2022).

## 3. METODE PENELITIAN

Pada penelitian ini membahas tentang implemantasi metode-metode CNN dengan arsitektur VGG-16 yang diolah menggunakan bahasa pemrograman *python* dan menggunakan *platform Google Colab* untuk mendeteksi penyakit pada daun teh menggunakan metode *convolutional neural network*. Dalam penyusunan penelitian ini dibutuhkan kerangka kerja yang bertujuan agar penelitian menjadi lebih teratur. Kerangka kerja yang digunakan pada Gambar 1.



Gambar 1. Metode Penelitian

### A. Studi Literatur

Terkait hal meneliti dan memahami materi yang berkaitan dengan deep learning khususnya metode *Convolutional Neural Network (CNN/Covnet)*. Studi literatur dilakukan dengan membaca jurnal penelitian yang berkaitan dengan metode *Convolutional Neural Network*.

## B. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah dataset yang didapatkan dari website Kaggle dengan nama “*Tea Leaf Disease*” yang diunggah oleh Shashwat Tiwari. Dataset ini terdapat 3 kelas, *Brown Blight*, *Alga spot*, *Healthy* (daun sehat). Total Data *Brown Blight* = 500 data, *Alga spot* = 500 data, *Healthy* (daun sehat) = 500 data.

## C. Perancangan Sistem

Pada tahap *preprocessing data* yang pertama ini adalah menyiapkan *dataset* citra penyakit daun teh. Data citra dari *dataset* penyakit daun teh dilakukan pengolahan hingga siap digunakan oleh arsitektur yang dibangun. Selanjutnya, citra ditransformasi dengan menggunakan *resize* piksel citra.

Ukuran resolusi *dataset* citra penyakit daun teh bervariasi. Arsitektur CNN dari penelitian ini hanya menerima input dengan resolusi ukuran yang sama. Mengurangi ukuran asli gambar ke ukuran yang ditetapkan adalah tujuan dari proses *resize* selama pelatihan. Dalam penelitian ini, ukuran resolusinya adalah  $(224 \times 224)$ .

Selanjutnya pada pembangunan model CNN, arsitektur ini membutuhkan citra input berukuran  $224 \times 224 \times 3$ . Maka dari itu, citra dengan berbagai ukuran akan diubah ukurannya terlebih dahulu. Kernel  $3 \times 3$ , *stride* 1, dan *zero padding* masing-masing digunakan dalam proses konvolusi pertama dan kedua, dengan ukuran  $224 \times 224 \times 3$  kurang lebih sebanyak 64 kali. Konvolusi ini menghasilkan hasil  $224 \times 224 \times 64$ , yang dapat dihitung menggunakan rumus 2.

$$N_{out} = \left\lfloor \frac{N_{in} + 2p - k}{s} \right\rfloor + 1 \quad (2)$$

Keterangan:

- $N_{out}$  = ukuran *feature map*
- $N_{in}$  = ukuran matriks masukan
- $k$  = ukuran matriks *filter*
- $p$  = ukuran *padding*
- $s$  = *stride*

Arsitektur VGG-16 Menggunakan fungsi aktivasi yaitu ReLu. Nilai piksel yang kurang dari nol akan majadi 0 saat fungsi aktivasi ReLu Menggunakan fungsi  $f = \max(0, x)$ . Kernel  $2 \times 2$  dan *stride* 2 memiliki ukuran yang sama dengan *pooling layer* yang digunakan pada arsitektur VGG-16. *Pooling layer* ini untuk mempercepat perhitungan dan meminimalkan ukurannya. Kernel  $3 \times 3$ , *stride* 1, *zero padding*, dengan ukuran input  $112 \times 112 \times 3$  dengan 128 filter, lapisan ketiga dan keempat dari proses konvolusi. Konvolusi ini menghasilkan keluaran  $112 \times 112 \times 128$ . Arsitektur VGG-16 Menggunakan fungsi aktivasi yaitu ReLu. Nilai piksel yang kurang dari nol akan majadi 0 saat fungsi aktivasi ReLu Menggunakan fungsi  $f = \max(0, x)$ . Dimensi akhir lapisan ini berukuran  $56 \times 56 \times 128$ .

*Convolutional layer* 5, 6, dan 7 menggunakan kernel  $3 \times 3$ , *stride* 1, *zero padding*, dan ukuran input  $56 \times 56 \times 3$  dengan 256 filter, lapisan ketiga dan keempat dari proses konvolusi. Keluaran dari konvolusi ini adalah  $56 \times 56 \times 256$ . Arsitektur VGG-16 menggunakan fungsi aktivasi yaitu ReLu. Nilai piksel yang kurang dari nol akan menjadi 0 saat fungsi aktivasi ReLu menggunakan fungsi  $f = \max(0, x)$ . Dimensi akhir lapisan ini berukuran  $28 \times 28 \times 256$ .

*Convolutional layer* 8, 9, dan 10 menggunakan kernel  $3 \times 3$ , *stride* 1, *zero padding*, dan ukuran input  $28 \times 28 \times 3$  dengan 512 filter, lapisan ketiga dan keempat dari proses konvolusi. Keluaran dari konvolusi ini adalah  $28 \times 28 \times 512$ . Arsitektur VGG-16 menggunakan fungsi aktivasi yaitu ReLu. Nilai piksel yang kurang dari nol akan menjadi 0 saat fungsi aktivasi ReLu menggunakan fungsi  $f = \max(0, x)$ . Dimensi akhir lapisan ini berukuran  $14 \times 14 \times 512$ .

*Convolutional layer* 11,12, dan 13 menggunakan kernel  $3 \times 3$ , *stride* 1, *zero padding*, dan ukuran input  $14 \times 14 \times 3$  dengan 512 filter, lapisan ketiga dan keempat dari proses konvolusi. Keluaran dari konvolusi ini adalah  $14 \times 14 \times 512$ . Arsitektur VGG-16 menggunakan fungsi aktivasi

yaitu ReLu. Nilai piksel yang kurang dari nol akan majadi 0 saat fungsi aktivasi ReLu menggunakan fungsi  $f = \max(0, x)$ . Dimensi akhir lapisan ini berukuran  $7 \times 7 \times 512$ .

Agar vektor dapat digunakan sebagai inputan dari lapisan *fully connected layer* selama proses *fully connected layer*, *output* dari lapisan ekstrasi fitur harus dilakukan "*flatten*" juga dikenal sebagai *reshape feature map*. Pada *fully conncted Layer* pertama dan kedua hanya terdiri dari 4096 dan 1000 neuron dalam ketentuan arsitektur VGG-16. Dalam arsitektur VGG-16, fungsi aktivasi ReLU mengikuti setiap *fully-connected layer*. Ukuran yang diproses lapisan ini adalah  $25088 \times 1$ , yang merupakan hasil perkalian  $7 \times 7 \times 512$ . Oleh karena itu, *dropout* harus digunakan untuk membatasi ukuran input sehingga menjadi 4096 input. Karena lapisan keluaran biasanya berisi lebih dari satu neuron, aktivasi *Softmax*, juga dikenal sebagai pengklasifikasi *Softmax*, adalah aktivasi yang sering digunakan untuk model multi-klasifikasi.

#### D. Pengujian Model

Pada *Convotional Neural Network*, pengujian dilakukan untuk mengetahui seberapa baik kinerja arsitektur yang dipilih. Metode *K-fold CrossValidation* digunakan sebagai metode pengujian pada penelitian ini. Untuk menilai keefektifan model, *k-fold cross validation* digunakan untuk menghitung kesalahan prediksi. Data dibagi menjadi himpunan bagian k yang ukurannya sama. Model klasifikasi menjalani pengujian dan pelatihan sebanyak k. Salah satu himpunan bagian akan digunakan sebagai data training dan data testing disetiap pengulangan.

### 4. HASIL DAN PEMBAHASAN

#### A. Hasil Uji Coba K-Fold

Pada pengujian penelitian ini, data train terdapat 1500 data citra secara acak dibagi menjadi data *train* dan data *validation* disusun dalam 3 folder. Ada 1215 data citra untuk data *train* dan 135 citra untuk data *validation* disetiap folder. Hasil pengujian penelitian ini menggunakan *K-Fold Cross Validation* dimana jumlah iterasi atau  $K = 10$  dengan masing- masing  $K$  memiliki 1215 latih serta 135 data validasi yang dipilih secara acak. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Menggunakan *K-Fold Cross Validation*

<i>Fold</i>	Keseluruhan Akurasi
1	91,1%
2	60%
3	97%
4	45,9%
5	96,2%
6	98,5%
7	98%
8	47,4%
9	57,7%
10	51,8%

Hasil pengujian penelitian ini menggunakan *K-Fold Cross Validation* dimana jumlah iterasi atau  $K = 10$  dengan masing- masing  $K$  memiliki 1215 latih serta 135 data validasi yang dipilih secara acak. Dari pengujian 10 fold diatas didapati skenario 6 pada 10 *fold cross validation* memiliki akurasi tertinggi dengan nilai 98,5% dan skenario 4 pada 10 *fold cross validation* memiliki nilai akurasi terendah yaitu 45,9%.

#### B. Hasil Uji Coba Menggunakan *Unseen Data*

Pada percobaan kedua, citra baru yang tidak ada dalam sistem digunakan sebagai citra uji. Jumlah citra yang digunakan pada percobaan kedua ini sebanyak 150 citra yang mewakili 3 kelas citra. Nilai akurasi yang relatif tinggi diperoleh dari pengujian menggunakan 50 citra baru. Pada uji

cobanya mendapatkan nilai akurasi 92% sehingga diperoleh 46 citra prediksi benar dan 4 citra prediksi salah. Tabel 2 menampilkan hasil percobaan kedua ini.

Tabel 2. Hasil Uji Coba Menggunakan *Unseen Data*

Percobaan	Prediksi Benar	Prediksi Salah
0	52	2
1	45	1
2	48	2

Pengujian menggunakan 150 citra baru menghasilkan nilai akurasi yang relatif cukup tinggi. Pengujian yang dilakukan dengan 150 citra baru menghasilkan nilai akurasi yang relatif tinggi. Pada uji cobanya nilai akurasi sebesar mendapatkan prediksi benar sebanyak 145 citra dan prediksi salah sebanyak 5 citra sehingga mendapatkan nilai akurasi sebesar 97%.

## 5. KESIMPULAN

Akurasi yang diperoleh dari pengujian menggunakan *fold cross validation* dengan nilai  $K = 10$  mendapatkan nilai akurasi tertinggi 98,5% pada skenario 6 *fold cross validation*. Oleh karena itu, dapat diberi kesimpulan bahwa metode *Convolutional Neural Network* (CNN) mampu mengidentifikasi penyakit daun teh dengan baik. Berdasarkan uji coba model yang menggunakan *10-fold cross validation* dengan nilai  $K = 10$  dan menggunakan data *testing* atau *unseen data*. Skenario 6 pada 10 *fold cross validation* memiliki tingkat akurasi tertinggi sebesar 96,6%. Skenario 8 mengalami penurunan sebesar 1,9% pada pengujian data *testing*. Nilai spesifisitas keseluruhan adalah 98,33%, nilai spesifisitas terendah dan tertinggi dari masing-masing kelas adalah 97% dan 100%. Rata-rata per kelas 98,35%. Selanjutnya nilai sensitivitas keseluruhan adalah 96,6% dengan nilai tertinggi dan terendah dari masing-masing kelas adalah 97,82% dan 96%. Rata-rata per kelas adalah 96,70%. Sensitivitas lebih kecil dibanding dengan spesifisitas artinya tingkat pendeteksi model penyakit lebih rendah dari tingkat pendeteksi daun sehat untuk mendeteksi citra penyakit daun teh. Setelah dilakukan penelitian ini, disarankan untuk menambahkan lebih banyak data citra jenis penyakit daun teh untuk meningkatkan keragaman jenis penyakit pada daun teh dan penelitian ini dapat dikembangkan ke dalam sebuah aplikasi.

## 6. DAFTAR PUSTAKA

- Felix, F., Faisal, S., Butarbutar, T. F. M., & Sirait, P. (2019). Implementasi CNN dan SVM untuk Identifikasi Penyakit Tomat via Daun. *Jurnal SIFO Mikroskil*, 20(2), 117–134.
- Ilahiyah, S., & Nilogiri, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 3(2), 49–56.
- Mardiana, L., Kusnandar, D., & Satyahadewi, N. (2022). Analisis Diskriminan Dengan K Fold Cross Validation Untuk Klasifikasi Kualitas Air Di Kota Pontianak. *Bimaster: Buletin Ilmiah Matematika, Statistika Dan Terapannya*, 11(1), 97–102.
- Nugroho, P. A., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network ( CNN ) Pada Ekspresi Manusia. *ALGOR*, 2(1), 12–21.
- Putri, O. N. (2020). *Implementasi Metode CNN Dalam Klasifikasi Gambar Jamur Pada Analisis Image Processing (Studi Kasus: Gambar Jamur Dengan Genus Agaricus Dan Amanita)*. Yogyakarta: Universitas Islam Indonesia.
- Zakariya, M. Y., Anindita, R., & Baladina, N. (2012). Analisis Daya Saing Teh Indonesia di Pasar Internasional. *AGRIMETA: Jurnal Pertanian Berbasis Keseimbangan Ekosistem*, 29–37.