

## ORNAMENTAL PLANT IDENTIFICATION SYSTEM USING TRANSFER LEARNING ON CONVOLUTIONAL NEURAL NETWORK

Kestrlia Rega Prilianti<sup>\*1</sup>, Vidian Vito Oktariyanto<sup>2</sup>, Hendry Setiawan<sup>3</sup>

<sup>1,2,3</sup>Department of Informatics Engineering, Universitas Ma Chung, Indonesia

Email: <sup>1</sup>[kestrlia.rega@machung.ac.id](mailto:kestrlia.rega@machung.ac.id), <sup>2</sup>[vidianvito@gmail.com](mailto:vidianvito@gmail.com), <sup>3</sup>[hendry.setiawan@machung.ac.id](mailto:hendry.setiawan@machung.ac.id)

(Article received: March 31, 2024; Revision: May 17, 2024; published: July 29, 2024)

### Abstract

There was a spike in the ornamental plants as a hobby while spending time at home during the COVID pandemic when people were restricted to activities outside the house. Unfortunately, along with this trend also came the serious issues associated with fake reports claiming that some ornamental plants were harmful to people's health. The public is more worried and perplexed by this situation, which also erodes their confidence in ornamental plants. This research aims to develop a real-time ornamental plant identification system as an educational medium for the public. To increase the system's accuracy, the transfer learning method is applied to the modified MobileNet CNN model. There are 9 species of popular ornamental plants in this identification system. From the experiments, it is known that the best accuracy has been achieved using the Adagrad optimization method (96% for training and 88% for testing data). The CNN model is then embedded in PLANTIS, an Android-based application prototype for ease of use purpose.

**Keywords:** CNN, mobileNet, ornamental plant, transfer learning.

## SISTEM IDENTIFIKASI TANAMAN HIAS DENGAN IMPLEMENTASI TRANSFER LEARNING PADA CONVOLUTIONAL NEURAL NETWORK

### Abstrak

Sejak pandemi COVID, tren hobi mengkoleksi tanaman hias meningkat di kalangan masyarakat. Seiring dengan maraknya pasar tanaman hias, muncul juga permasalahan terkait informasi yang kurang tepat terkait dampak tanaman hias tertentu terhadap kesehatan. Bahkan juga muncul beberapa hoax yang tidak dapat dipertanggungjawabkan serta menambahkan kecemasan bagi masyarakat. Tujuan dari penelitian ini adalah membuat sistem identifikasi tanaman hias secara *real-time* sebagai media edukasi bagi masyarakat. Dalam upaya meningkatkan akurasi sistem, diterapkan metode *transfer learning* pada model CNN MobileNetV2 yang dimodifikasi. Terdapat 9 spesies tanaman hias populer yang digemari masyarakat di dalam sistem identifikasi ini. Dari eksperimen yang dilakukan, diketahui metode optimasi Adagrad menghasilkan akurasi terbaik yaitu sebesar 96% untuk data *training* dan 88% untuk data *testing*. Model tersebut kemudian ditanamkan pada prototipe aplikasi berbasis Android PLANTIS agar mudah digunakan oleh masyarakat.

**Kata kunci:** CNN, mobileNet, tanaman hias, transfer learning.

### 1. PENDAHULUAN

Selama pandemi Covid-19, kegiatan menanam tanaman hias sangat digemari masyarakat. Tujuannya pun berbeda-beda, salah satunya dijadikan sebagai hiasan di rumah. Masyarakat perkotaan memanfaatkan tanaman hias sebagai penralisir polusi udara. Selain untuk kepentingan individu, tanaman hias ini juga berperan di dalam pembangunan sektor agrowisata di Indonesia. Sektor agrowisata memanfaatkan usaha pertanian sebagai objek wisata, oleh karena itu bisnis tanaman hias mempunyai potensi yang baik untuk dikembangkan secara profesional [1].

Dalam membeli tanaman hias, konsumen pada umumnya akan fokus pada visualnya. Mereka akan memilih yang bagus dilihat dan murah, tanpa berusaha mengetahui apa manfaat dan mungkin juga potensi bahayanya. Hal ini kurang sejalan dengan fakta pesatnya perkembangan teknologi yang memungkinkan informasi sangat mudah didapatkan lewat internet. Namun, terkait dengan pesatnya sebaran informasi, masyarakat juga harus waspada. Tidak sedikit berita hoax atau informasi palsu beredar di media *online*. Hoax merupakan sebuah kebohongan yang dikarang sedemikian rupa oleh seseorang untuk menutupi atau mengalihkan

perhatian dari kebenaran dengan tujuan untuk kepentingan pribadi baik secara intrinsik maupun ekstrinsik [2]. Salah satu berita hoax mengenai tanaman hias, terjadi pada tahun 2019. Tanaman Sri Rejeki diberitakan mengandung getah racun yang bisa mematikan anak-anak kurang dari 1 menit. Setelah dikonfirmasi, salah satu pakar dari Universitas Indonesia menyatakan bahwa racun dari tanaman tersebut tergolong ringan karena hanya menyebabkan gatal-gatal ringan [3]. Dari satu berita hoax yang tersebar di internet dapat berdampak buruk bagi penjual tanaman Sri Rejeki. Begitu juga bagi pembeli atau pemelihara Sri Rejeki, yang dapat langsung membuang tanaman tersebut. Oleh karena itu, diperlukan media edukasi yang dapat secara mudah dan murah dimanfaatkan oleh masyarakat untuk mendapatkan informasi yang terjamin kebenarannya.

Dengan pesatnya pemanfaatan teknologi berbasis kecerdasan buatan, media edukasi yang mudah dan murah tersebut dapat dikembangkan dengan cepat. Pengenalan jenis tanaman dapat dilakukan melalui citra tanaman secara *real-time* dengan teknologi *deep learning* [4]-[7]. Hasil pengenalan kemudian dapat dihubungkan dengan berbagai informasi terkait baik yang tersimpan di dalam database internal maupun sumber lain yang ada di internet. Salah satu teknologi dalam *deep learning* yang banyak digunakan untuk pengenalan citra digital adalah *Convolutional Neural Network* (CNN).

*Convolutional Neural Network* adalah salah satu jenis arsitektur jaringan syaraf tiruan yang populer digunakan untuk menangani *input* berupa citra [8]. Model CNN termasuk dalam keluarga *Deep Neural Network* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) [9]. Model CNN dapat digunakan untuk mengolah data *input* dengan banyak dimensi tidak seperti model jaringan syaraf tiruan yang umumnya mensyaratkan matriks satu dimensi sebagai *input*-nya. Beberapa penelitian yang menggunakan metode CNN untuk deteksi objek melaporkan akurasi yang sangat baik dengan waktu pelatihan yang relatif cepat [10]-[12]. Namun, isu terkait jumlah citra pada dataset menjadi kendala pada beberapa kasus yang menggunakan metode ini.

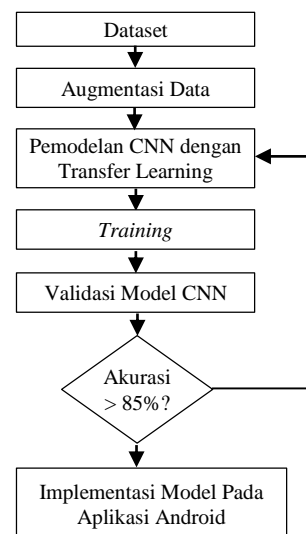
Pada tahap pengembangan model CNN, implementasi *transfer learning* diketahui dapat mempercepat proses pembelajaran sekaligus mengatasi masalah keterbatasan dataset. *Transfer learning* merupakan teknik yang memanfaatkan model yang sudah dilatih (*pre-trained model*) untuk meningkatkan kinerja dari model baru yang sedang dilatih [13]. Teknik ini telah digunakan di beberapa penelitian seperti klasifikasi citra makanan dan non makanan dengan model *Residual Network* [14] dan penelitian tentang identifikasi penyakit *leaf mold* daun tomat menggunakan model DenseNet21 [15]. Hasil dari penelitian tersebut menunjukkan bahwa implementasi *transfer learning* sangat membantu meningkatkan akurasi sistem klasifikasi.

Penelitian ini bertujuan untuk membantu edukasi kepada masyarakat terkait tanaman hias secara efektif dan efisien dengan memanfaatkan kecerdasan buatan dan aplikasi perangkat bergerak. Pengguna cukup mengambil foto dari tanaman yang hendak dicari informasinya, untuk selanjutnya model CNN yang ditanamkan pada aplikasi melakukan identifikasi jenis dari tanaman hias tersebut. Setelah jenis tanaman ditentukan, selanjutnya aplikasi akan menyajikan beberapa informasi penting dan juga tautan ke sumber berita yang terpercaya.

## 2. METODE PENELITIAN

### 2.1. Tahapan Pengembangan Sistem

Tahapan pengembangan sistem identifikasi tanaman hias pada penelitian ini dapat dilihat pada gambar 1. Dari gambar tampak bahwa sebelum aplikasi Android dibuat, model CNN yang digunakan untuk identifikasi dikembangkan terlebih dahulu dan diuji kinerjanya melalui proses validasi model. Jika model telah memenuhi syarat, yaitu akurasi identifikasi lebih dari 85%, maka baru dapat ditanamkan pada aplikasi Android.



Gambar 1. Tahapan Pengembangan Sistem Identifikasi Tanaman Hias

### 2.2. Dataset

Dataset berupa citra daun yang digunakan dalam penelitian ini diambil menggunakan kamera *smartphone* Infinix Smart 3 Plus, Model X627 dengan kamera belakang 13 MegaPixels. Pengambilan citra difokuskan ke objek daun di lingkungan dengan intensitas cahaya sedang (sekitar 500-700 lux), *background* bebas, jarak dari kamera kurang lebih 30 - 50 cm seperti contoh yang tampak pada gambar 2.

Terdapat 9 spesies tanaman yang digunakan untuk membangun prototipe aplikasi sistem identifikasi tanaman hias. Untuk masing-masing spesies disediakan 30 citra. Berikut adalah 9 spesies yang digunakan dalam penelitian ini:



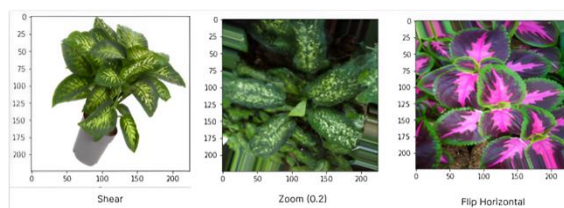
Gambar 2. Salah Satu Contoh Gambar pada Dataset

1. Andong (*Cordyline fruticosa*)
2. Kornus (*Cornus alba*)
3. Krokot Epah Daun Merah (*Alternanthera brasiliana*)
4. Lidah Buaya (*Aloe vera*)
5. Lidah Mertua (*Sansevieria trifasciata*)
6. Miana (*Coleus scutellarioides*)
7. Pucuk Merah (*Syzygium oleana*)
8. Puring (*Codiaeum variegatum*)
9. Sri Rejeki (*Agloenema commutatum*)

Dataset pada penelitian ini dibagi dan diletakkan kedalam dua *folder* yaitu *folder train* dan *folder test*. Selain 9 kategori spesies tanaman yang dapat diidentifikasi, ditambahkan satu lagi kategori untuk menangani citra selain kesembilan spesies yang telah ditentukan tersebut. Sehingga terdapat 10 kategori pada dataset dengan total 300 citra. Pembagian untuk data *training* dan *testing* (validasi) adalah 60%:40%.

### 2.3. Augmentasi Data

Pada tahap ini, dilakukan penambahan variasi terhadap citra yang telah ada pada dataset. Tujuannya adalah untuk menambah jumlah data agar proses *training* pada CNN dapat berjalan dengan lebih baik. Variasi terhadap citra dilakukan dengan menerapkan operasi *shear*, *zoom*, *flip*, dan *shift*. Seluruh citra pada dataset telah disesuaikan ukurannya menjadi 224x224. Contoh hasil augmentasi dapat dilihat pada gambar 3.

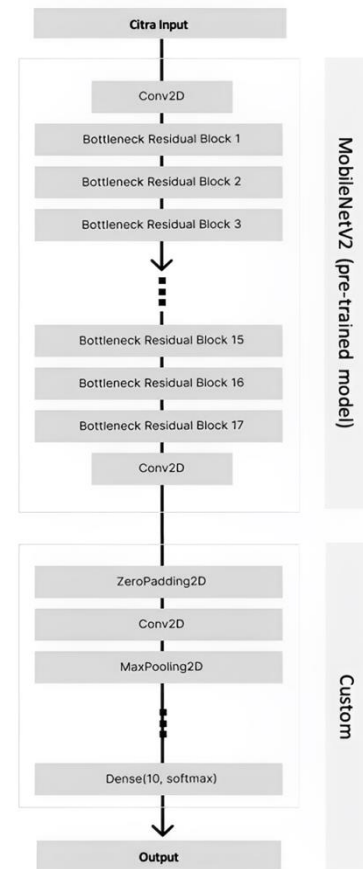


Gambar 3. Contoh Hasil Augmentasi Citra pada Dataset

### 2.4. Pemodelan CNN dengan *Transfer Learning*

Arsitektur CNN yang digunakan pada sistem dapat dilihat pada gambar 4. Tampak bahwa arsitektur CNN tersusun atas dua bagian utama. Bagian pertama adalah arsitektur MobileNetV2 [16] dan bagian kedua adalah arsitektur *custom* yang didesain sendiri oleh tim peneliti untuk kemudian ditambahkan setelah *layer* konvolusi terakhir milik MobileNetV2. MobileNetV2 dipilih karena dapat bekerja dengan cepat pada perangkat seluler. Jumlah parameter pada model yang dihasilkan oleh MobileNetV2 lebih sedikit dibandingkan arsitektur

CNN lainnya. Meskipun demikian, akurasi klasifikasi yang dihasilkan oleh model MobileNetV2 telah dibuktikan sangat baik [17].



Gambar 4. Arsitektur CNN Pada Sistem Identifikasi Tanaman Hias

Arsitektur MobileNetV2 yang digunakan dalam penelitian ini telah dilengkapi dengan bobot yang diperoleh melalui *training* dengan menggunakan dataset ImageNet. ImageNet terdiri dari 1.000 kategori gambar dan masing-masing kategori terdiri dari 1.200 contoh. Dengan demikian MobileNetV2 pada penelitian ini telah menjalani *training* menggunakan 1.200.000 citra. Pola yang telah dikenali melalui *training* tersebut kemudian ditransferkan kepada proses *training* berikutnya dengan dataset baru yang berisi citra 9 spesies tanaman hias sebagaimana yang telah dijelaskan pada sub bab 2.2. Dengan konsep *transfer learning* ini, proses *training* menjadi lebih cepat konvergen meskipun dengan jumlah dataset yang terbatas [18]. Dengan demikian waktu pengumpulan data dan proses *training* dapat dihemat sekaligus.

Untuk mendapatkan model terbaik, dilakukan eksperimen *fine-tuning* pada MobileNetV2 dengan dataset baru. Digunakan 6 metode optimasi dalam eksperimen pencarian model terbaik dengan *epoch* sejumlah 100. *Loss function* yang digunakan adalah *Categorical Cross Entropy* karena kasus identifikasi tanaman hias ini merupakan kasus klasifikasi multi kelas.

## 2.5. Training

Metode optimasi yang digunakan dalam penelitian ini adalah SGD, Adadelta, RMSProp, Adam, Adamax, dan Nadam. Berikut adalah formula yang digunakan pada masing-masing metode optimasi:

### 1. Stochastic Gradient Descent (SGD)

Update parameter  $\theta$  pada waktu ke  $t+1$  dihitung dengan menggunakan persamaan 1,  $\eta$  adalah *learning rate* dan  $J$  adalah fungsi error [19].

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \quad (1)$$

### 2. Adaptive Delta (Adadelta)

Konsep dasar dari Adadelta adalah mengakumulasikan beberapa gradien berpangkat  $w$  sebagaimana persamaan 2 dengan  $\beta$  adalah konstanta *decay* [20]. Update parameter pada waktu ke  $t+1$  dihitung dengan persamaan 3 dan 4.

$$RMS[g]_t = E[g^2]_t = \beta \cdot E[g^2]_{t-1} + (1 - \beta) \cdot g_t^2 \quad (2)$$

$$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t \quad (3)$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \quad (4)$$

### 3. Root Mean Square Propagation (RMSProp)

Metode ini membagi *learning rate* dengan *running average* dari kuadrat gradien ( $E[g^2]_t$ ) yang mengalami *decay* secara eksponensial [21]. Update parameter pada  $t+1$  dihitung menggunakan persamaan 5 dan persamaan 6.

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2 \quad (5)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_{t+\epsilon}}} \cdot g_t \quad (6)$$

### 4. Adaptive Momentum (Adam)

Adam menghitung *learning rate* secara adaptif untuk setiap parameter. Adam menyimpan kuadrat gradien dengan koreksi bias dari epoch sebelumnya ( $\hat{v}_t$ ). Adam juga menyimpan gradien lama yang sudah terkoreksi biasanya ( $\hat{m}_t$ ). Kedua nilai ( $v_t$  and  $m_t$ ) digunakan untuk mengestimasi *decay* dari moment pertama dan kedua dari gradien-gradien yang dihitung menggunakan persamaan 7 sampai dengan persamaan 10,  $\beta$  adalah konstanta *decay* [22]. Update parameter pada  $t+1$  dilakukan dengan menggunakan persamaan 11.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (7)$$

$$\hat{m}_t = m_t / (1 - \beta_1^t) \quad (8)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t \quad (9)$$

$$\hat{v}_t = v_t / (1 - \beta_2^t) \quad (10)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (11)$$

### 5. Adaptive Max Pooling (Adamax)

Adamax adalah pengembangan dari Adam. Modifikasi dilakukan dengan menggunakan *infinity norm* ( $u_t$ ). Telah terbukti bahwa nilai  $v_t$  pada Adam dengan penambahan  $\ell_{\infty}$  akan konvergen pada nilai yang lebih stabil [22]. Update parameter pada  $t+1$  dihitung dengan persamaan 12 dan persamaan 13.

$$\theta_{t+1} = \theta_t - \frac{\eta}{u_t} \hat{m}_t \quad (12)$$

$$u_t = \beta_2^{\infty} v_{t-1} + (1 - \beta_2^{\infty}) |g_t|^{\infty} = \max(\beta_2 \cdot v_{t-1}, |g_t|) \quad (13)$$

### 6. Nesterov Adaptive Momentum (Nadam)

Nadam adalah kombinasi dari Adam and Nesterov Accelerated Gradient (NAG). Metode ini mengintegrasikan momentum Nesterov kedalam Adam [23]. Karena momentum Nesterov telah terbukti lebih baik daripada momentum klasik, dihipotesiskan bahwa Nadam akan lebih baik kinerjanya daripada Adam. Update parameter pada  $t+1$  dihitung dengan persamaan 14.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} (\beta_1 \hat{m}_t + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t}) \quad (14)$$

## 2.6. Validasi Model CNN

Untuk mendapatkan model terbaik, dilakukan sejumlah 3 eksperimen utama. Eksperimen pertama ditujukan untuk mengetahui efektifitas metode *transfer learning* dalam meningkatkan akurasi identifikasi. Eksperimen kedua ditujukan untuk menentukan jumlah *layer* pada arsitektur MobileNetV2 yang harus ikut dalam proses *training* lanjutan dengan dataset baru (*fine-tuning*). Eksperimen ketiga ditujukan untuk menentukan metode optimasi yang paling baik untuk diterapkan pada proses *training* sehingga menghasilkan model dengan akurasi terbaik. Keseluruhan eksperimen dilakukan dengan bantuan Google Colaboratory dan beberapa *library deep learning* untuk python. Untuk mendapatkan model terbaik, Indikator kinerja yang digunakan adalah akurasi sebagaimana persamaan 15.

$$Akurasi = \frac{Total\ Prediksi\ Benar}{Total\ Prediksi} \cdot 100\% \quad (15)$$

## 2.7. Implementasi pada Aplikasi Android

Model terbaik disimpan kedalam *file* berekstensi *.flite*. *File* ini kemudian diimpor ke dalam aplikasi android menggunakan Flutter. Pada pengembangan aplikasi android digunakan pula

format JSON untuk menyimpan nama-nama tanaman hias, manfaat, dan bahayanya. Aplikasi android menerima input berupa citra tanaman yang diambil oleh pengguna dengan menggunakan kamera *smartphone*. Model CNN kemudian melakukan klasifikasi dan hasil klasifikasi ditampilkan pada layar *smartphone*. Selain jenis dari tanaman hias, kepada pengguna juga akan disajikan beberapa keterangan terkait tanaman hias tersebut, seperti manfaat dan bahayanya.

### 3. HASIL DAN PEMBAHASAN

Berikut adalah pemaparan hasil-hasil yang telah diperoleh, utamanya pada tahap validasi model CNN untuk mendapatkan model terbaik dan juga aplikasi Android yang telah dibangun.

#### 3.1. Hasil Validasi Model CNN

##### 3.1.1. Efektifitas *Transfer Learning*

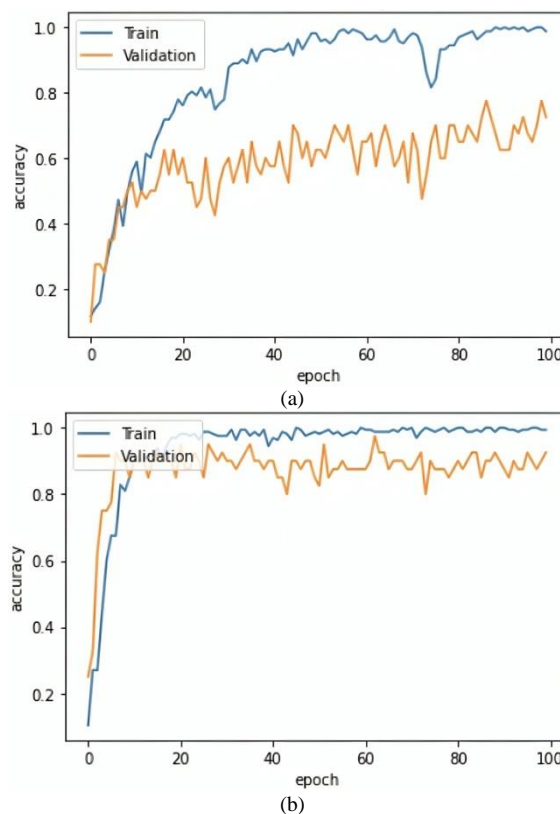
Pada eksperimen ini, sejumlah 3 model hasil *training* arsitektur *custom* dibandingkan akurasi dengan 3 model hasil *training* arsitektur gabungan *MobileNetV2* dan *custom*. Metode optimasi yang digunakan dalam eksperimen ini adalah Adam, RMSProp, dan Adagrad. Dari ringkasan hasil pada tabel 1, diketahui bahwa akurasi tertinggi yang dapat dicapai oleh model tanpa *transfer learning* adalah sebesar 70%. Nilai ini lebih kecil jika dibandingkan dengan akurasi tertinggi yang dapat dicapai oleh model dengan *transfer learning*, yaitu sebesar 88%.

Tabel 1. Perbandingan Akurasi *Testing* Model Tanpa *Transfer Learning* dan Model Dengan *Transfer Learning*

Model	Model	Optimizer	Akurasi
1	Tanpa <i>Transfer Learning</i>	Adam	70%
2		RMSProp	66%
3		Adagrad	40%
4	Dengan <i>Transfer Learning</i>	Adam	84%
5		RMSProp	88%
6		Adagrad	88%

Fakta lain yang didapatkan adalah terdapat perbedaan metode optimasi terbaik yang dapat mengantarkan kedua model mencapai akurasi terbaiknya. Model tanpa *transfer learning* cocok dilatih dengan Adam sedangkan model dengan *transfer learning* lebih cocok dilatih dengan metode optimasi lainnya. Dengan demikian terbukti bahwa metode *transfer learning* dapat meningkatkan akurasi klasifikasi.

Selain perbedaan akurasi dan metode optimasi terbaik, dari percobaan diketahui bahwa tanpa metode *transfer learning* terjadi indikasi *overfitting* selama proses *training*. Gambar 5 menunjukkan contoh perbedaan kondisi pencapaian akurasi selama proses *training*. Pada contoh tersebut metode optimasi yang digunakan adalah Adam. Tampak terjadi *overfitting* pada proses *training* yang dijalankan tanpa menggunakan metode *transfer learning*. Kondisi yang sama juga terjadi saat model menjalani *training* dengan metode optimasi RMSProp dan Adagrad.



Gambar 5. Grafik Perbandingan Kondisi Proses *Training* dengan dan Tanpa *Transfer Learning*: a) Tanpa *Transfer Learning*, b) Dengan *Transfer Learning*

##### 3.1.2. *Fine-Tuning* pada *MobileNetV2*

Pada eksperimen ini dilakukan pemantauan pencapaian akurasi terhadap 3 pengaturan jumlah layer pada *MobileNetV2* yang ikut dilatih ulang (*fine-tuning*) bersama model *custom* dengan dataset baru. Pengaturan pertama adalah melatih ulang sejumlah 20% (31 layer) komponen *MobileNetV2*. Pengaturan kedua adalah melatih ulang sejumlah 10% (15 layer) dan pengaturan ketiga adalah tidak melatih ulang seluruh layer pada *MobileNetV2*. Perbandingan akurasi terkait perbedaan jumlah layer pada *MobileNetV2* yang dilatih ulang dapat dilihat pada tabel 2.

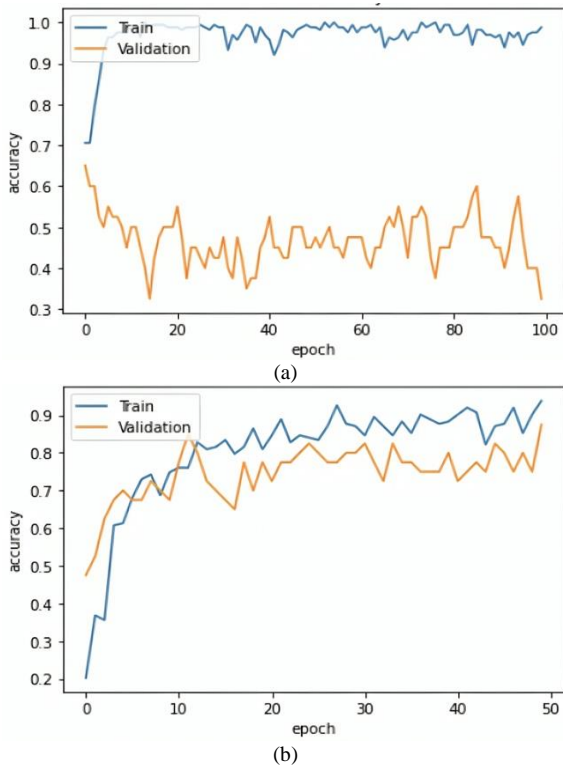
Tabel 2. Perbandingan Akurasi *Testing* Model Terhadap Pengaturan Jumlah Layer *MobileNetV2* yang Dilatih Ulang

No	Metode Optimasi	Jumlah Layer <i>MobileNetV2</i> yang Dilatih Ulang		
		31	15	0
1	Adam	68%	18%	80%
2	RMSprop	60%	10%	88%
3	Adagrad	82%	72%	88%

Dari tabel 2 diperoleh fakta bahwa seluruh layer pada *MobileNetV2* tidak perlu dilatih ulang. Tampak bahwa akurasi paling baik saat melakukan klasifikasi tanaman hias dicapai ketika seluruh layer pada model *MobileNetV2* tetap menggunakan bobot yang sudah diperoleh dari proses *training* sebelumnya dengan ImageNet. Sedangkan pada kondisi sebagian layer menjalani *training* ditemukan kondisi *overfitting* sebagaimana yang tampak pada contoh di gambar 6.



Pada contoh tersebut, model menjalani *training* dengan metode optimasi Adam. Kondisi yang sama juga ditemukan saat model menjalani *training* dengan metode optimasi RMSProp dan Adagrad.



Gambar 6. Grafik Perbandingan Akurasi *Training* pada Proses *Fine-Tuning* MobileNetV2; a) Sebagian *Layer* Menjalani *Training* Ulang, b) Tidak Ada *Layer* yang Menjalani *Training* Ulang

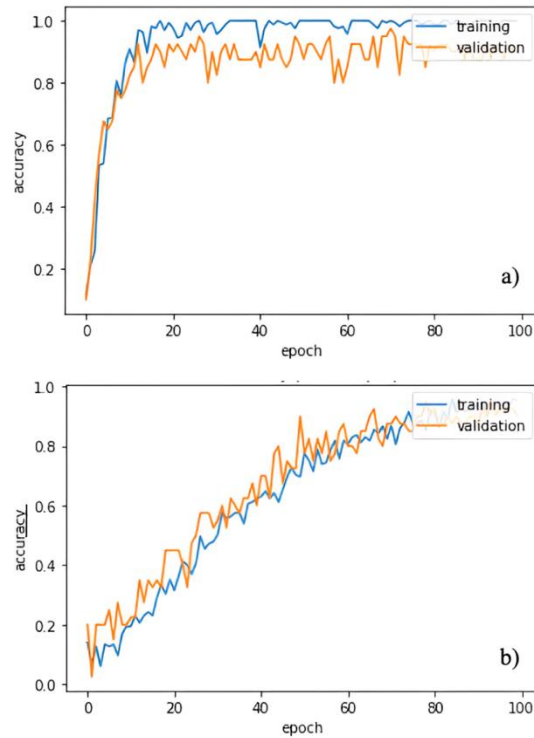
Hal ini sekaligus membuktikan bahwa kualitas dari *pre-trained model* sudah sangat baik sehingga tidak diperlukan pembobotan ulang. Mengacu pada hasil eksperimen kedua ini, hanya arsitektur *custom* saja yang akan menjalani *training* dengan dataset baru pada eksperimen ketiga.

### 3.1.3. Metode Optimasi Terbaik

Pada eksperimen ini, arsitektur pada gambar 4 menjalani *training* dengan menggunakan 6 metode optimasi. Sesuai temuan pada eksperimen *fine-tuning*, proses *training* hanya akan dilakukan untuk menyesuaikan bobot pada arsitektur CNN bagian kedua (*custom*), sedangkan arsitektur CNN bagian pertama (MobileNetV2) tetap akan menggunakan bobot lamanya. Untuk setiap metode optimasi, dilakukan sejumlah 10 kali *training*.

Tabel 3. Perbandingan Rata-Rata Akurasi *Testing* dengan 6 Metode Optimasi

No	Metode Optimasi	Rata-Rata Akurasi
1	Adagrad	88%
2	RMSProp	88%
3	SGD	84%
4	Nadam	84%
5	Adam	80%
6	Adadelta	14%



Gambar 7. Grafik Perbandingan Akurasi *Training* dan Validasi; a) Menggunakan Metode Optimasi RMSProp, b) Menggunakan Metode Optimasi Adagrad

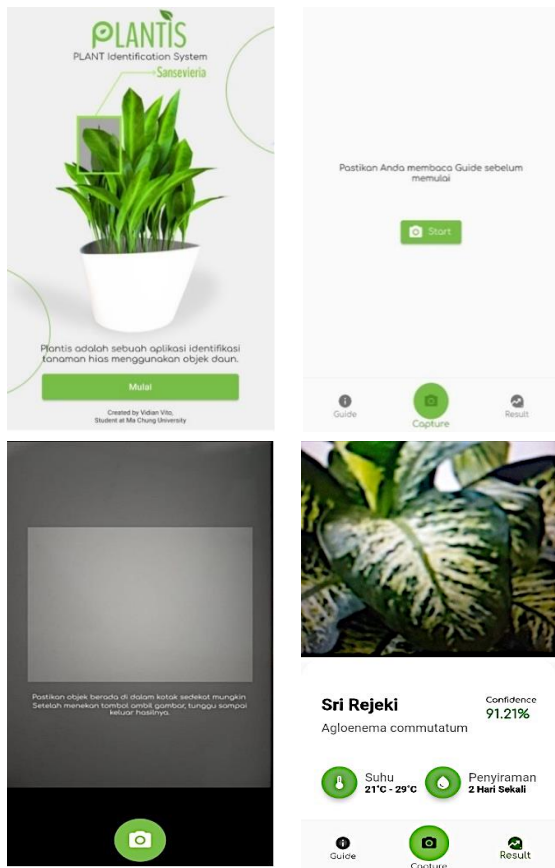
Dari tabel 3 dapat diketahui bahwa pada umumnya semua metode optimasi dapat menghasilkan akurasi yang baik. Namun hal ini tidak berlaku untuk Adadelta yang pencapaian akurasinya sangat rendah jika dibandingkan dengan pencapaian akurasi oleh metode optimasi lainnya. Akurasi tertinggi (88%) dicapai saat arsitektur CNN menjalani *training* dengan metode optimasi RMSProp dan Adagrad. Maka, perlu ditinjau aspek lain untuk menentukan metode optimasi mana yang akan digunakan untuk menghasilkan model yang akan ditanamkan pada aplikasi android. Gambar 7 menunjukkan grafik pencapaian akurasi *training* dan *testing* (validasi) di setiap *epoch* untuk metode optimasi RMSProp dan Adagrad.

Dari gambar 7 tampak bahwa pada grafik pencapaian akurasi dengan metode optimasi Adagrad nilai *training* dan validasi saling berdekatan. Pada pencapaian akurasi dengan metode optimasi RMSProp terdapat selisih yang cukup jelas terlihat diantara akurasi *training* dan validasi. Hal ini menunjukkan adanya potensi *overfitting* [24]. Oleh karena itu, model yang dihasilkan dari *training* dengan metode optimasi Adagrad yang akan diimplementasikan pada aplikasi Android.

### 3.2. Hasil Aplikasi Android

Aplikasi Android yang telah dibangun diberi judul PLANTIS yang merupakan singkatan dari *Plant Identification System* (lihat gambar 8). Terdapat 6 halaman utama ditambah 1 halaman untuk menampilkan *website* terkait. Berikut adalah daftar

halaman pada aplikasi PLANTIS: *Splash Screen, Introduction, Guide, Capture, Camera, Result, dan Web View.*



Gambar 8. Cuplikan Tampilan Aplikasi PLANTIS

Untuk menggunakan aplikasi ini, pengguna cukup membuka halaman *Capture* untuk kemudian terhubung dengan kamera *smartphone*. Pengguna harus mengarahkan jendela akuisisi citra ke arah tanaman sesuai petunjuk pada aplikasi. Setelah citra tanaman diperoleh maka secara otomatis halaman *Result* akan muncul. Pada halaman ini akan tampil hasil identifikasi tanaman hias berupa nama umum dari tanaman, yaitu nama yang lazim dikenal masyarakat. Selain itu juga akan disajikan nama latin dari tanaman tersebut. Terdapat pula informasi suhu ideal, durasi penyiraman, deskripsi, manfaat, bahaya, dan juga informasi lainnya. Ditampilkan pula nilai *confidence* yang menginformasikan tingkat keyakinan akan ketepatan hasil identifikasi oleh model CNN. Apabila citra tanaman yang difoto oleh pengguna mempunyai nilai *confidence* identifikasi dibawah 70%, akan muncul status "tidak diketahui". Nilai *confidence* pada rentang antara 70% - 89% akan menampilkan teks berwarna jingga. Nilai *confidence* diatas 89% akan menampilkan teks berwarna hijau, selain itu akan muncul teks berwarna merah.

#### 4. DISKUSI

*Transfer learning* telah menjadi pendekatan yang efektif dalam meningkatkan kinerja model CNN

dalam berbagai aplikasi, termasuk pada aplikasi identifikasi tanaman hias pada penelitian ini. *Transfer learning* memungkinkan model memanfaatkan pengetahuan yang telah dipelajari dari dataset yang lebih besar, seperti ImageNet[25], dan menerapkannya pada tugas identifikasi tanaman hias tanpa perlu melatihnya dari awal. Dengan memanfaatkan representasi fitur yang telah dipelajari dari ImageNet, model CNN dalam penelitian ini dapat dengan cepat mengidentifikasi pola visual yang kompleks dalam dataset citra tanaman hias yang terbatas jumlahnya.

Namun, meskipun *transfer learning* memberikan keunggulan awal dalam proses identifikasi tanaman hias, penyesuaian model dengan dataset tanaman hias yang spesifik tetap diperlukan [26]-[27]. Oleh karena itu pada penelitian ini disediakan dataset baru yang terdiri dari 9 spesies tanaman hias. Dari eksperimen *fine-tuning* diketahui bahwa *pre-trained model* tidak perlu dilatih ulang, oleh karena itu dilakukan penambahan beberapa *layer* konvolusi dibelakang *layer-layer* MobileNetV2 sehingga model dapat menyesuaikan diri dengan karakteristik unik dari dataset tanaman hias dan meningkatkan kinerjanya secara signifikan.

Dari eksperimen yang telah dilakukan, beberapa pertimbangan penting yang harus diperhatikan dalam pembentukan sistem identifikasi dengan implementasi *transfer learning* secara umum adalah sebagai berikut:

1. Pemilihan model yang sesuai. Memilih model yang sesuai untuk *transfer learning* adalah langkah penting. MobileNetV2 adalah salah satu pilihan yang populer untuk *transfer learning* dalam identifikasi objek visual karena kecepatan dan efisiensinya, tetapi ada pilihan lain yang juga layak dipertimbangkan tergantung pada kebutuhan spesifik aplikasi. Pada penelitian pendahuluan telah dibandingkan pula beberapa model *transfer learning* lain namun MobileNetV2 menunjukkan kinerja yang paling baik.
2. Penyesuaian pada tugas baru. Meskipun model telah dilatih sebelumnya dengan dataset yang berbeda, penyesuaian pada tugas identifikasi tanaman hias tetap diperlukan. Ini dapat dilakukan melalui teknik *fine-tuning* [28]-[29], beberapa lapisan akhir dari model disesuaikan dengan dilatih ulang menggunakan dataset tanaman hias yang spesifik. Namun eksperimen pada penelitian ini menunjukkan bahwa penambahan *layer* konvolusi lebih bermakna dibandingkan dengan *fine-tuning* pada layer MobileNetV2.
3. Evaluasi kinerja. Meskipun *transfer learning* telah dibuktikan dapat meningkatkan kinerja model, tetap perlu dilakukan evaluasi secara komprehensif. Tujuannya adalah untuk memahami sejauh mana peningkatan kinerja yang dicapai oleh *transfer learning* dan

bagaimana perbandingannya dengan model yang dilatih dari awal atau model lain yang mungkin cocok untuk digunakan pada kasus yang sedang ditangani.

## 5. KESIMPULAN

Berdasarkan hasil eksperimen untuk pemodelan dan juga pengembangan aplikasi, diketahui bahwa implementasi *transfer learning* terbukti dapat membantu pengembangan model untuk sistem identifikasi tanaman hias dengan efektif dan efisien. Hal ini tampak dari nilai akurasi model yang jauh lebih tinggi setelah menggunakan *pre-trained model* MobileNetV2. Selain itu, seluruh bobot dari *pre-trained model* tersebut dapat digunakan tanpa perlu dilatih ulang dengan dataset yang baru. Hanya model tambahan (*custom*) saja yang perlu disesuaikan bobotnya dengan dataset baru. Model terbaik diperoleh melalui proses *training* dengan metode optimasi Adagrad, rata-rata akurasi *training* mencapai 96% sedangkan rata-rata akurasi *testing* (validasi) mencapai 88%. Tidak ada indikasi *overfitting* selama proses *training* sehingga model dinyatakan baik kinerjanya. Model terbaik tersebut telah sukses ditanamkan kedalam prototipe aplikasi berbasis Android yang diberi nama PLANTIS. Aplikasi ini masih terus dikembangkan secara bertahap untuk menambahkan jenis tanaman hias yang dapat dikenali.

## DAFTAR PUSTAKA

- [1] A. Nurfitriyani and S. Maulidah, "Analisis Potensi dan Prospek Industri Tanaman Hias di Indonesia," *J. AgriSosioekonomi*, vol.13, no.2, pp. 165–175, 2020, doi: 10.29244/jaes.13.2.165-175
- [2] G. Pennycook and D. G. Rand, "Fighting Misinformation on Social Media Using Crowdsourced Judgments of News Source Quality," in *Proceedings of the National Academy of Sciences*, vol.116, no.7, pp. 2521-2526 2019.
- [3] Kominfo, "Tanaman Hias Beracun Paling Mematikan," 2019. [https://www.kominfo.go.id/content/detail/17256/disinformasi-tanaman-hias-beracun-paling-mematikan/0/laporan\\_isu\\_hoaks](https://www.kominfo.go.id/content/detail/17256/disinformasi-tanaman-hias-beracun-paling-mematikan/0/laporan_isu_hoaks) (accessed Jan. 26, 2024).
- [4] N. Arifin, Maratutahirah, J. Rusman, and M.F. Rasyid, "Leaf Disease Detection in Tomato Plants using Xception Model in Convolutional Neural Network Method," *J. Tek. Inf.*, vol.5, no.2, pp. 571-577 2024, doi:10.52436/1.jutif.2024.5.2.1926
- [5] D.A. Wibowo, N. Suciati, and A. Yuniarti, "Hyperparameter Optimization of Convolutional Neural Network for Flower Image Classification using Grid Search Algorithm," *J. Tek. Inf.*, vol.5, no.1, pp. 313-320 2024, doi:10.52436/1.jutif.2024.5.1.1798
- [6] A. Pratiwi, A. Fauzi, "Implementation of Deep Learning on Flower Classification using CNN Method," *J. Tek. Inf.*, vol.5, no.2, pp. 487-495 2024, doi:10.52436/1.jutif.2024.5.2.1674
- [7] D. Zahirah, Purnawansyah, N. Kurniati, and H. Darwis, "Digital Image Classification of Herbal Leaves using KNN and CNN With GLCM Features," *Tek. Inf.*, vol.5, no.1, pp. 61-67 2024, doi: 10.52436/1.jutif.2024.5.1.1162
- [8] M. Arsal, B.A. Wardijono, and D. Anggraini, "Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning," *J. Nas. Tek. Sist. Inf.*, vol.6, no.1, pp. 55-63 2020, doi: 10.25077/TEKNOSI.v6i1.2020.55-63
- [9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791
- [10] P. Smith and C. Chen, "Transfer Learning with Deep CNNs for Gender Recognition and Age Estimation," in *IEEE International Conference on Big Data*, 2019, pp. 2564–2571.
- [11] T. Kaur and T.K. Gandhi, "Automated Brain Image Classification Based On VGG-16 and Transfer Learning," in *International Conference on Information Technology*, 2019, pp. 94–98.
- [12] X. Sun, G. Li, P. Qu, X. Xie, X. Pan, and W. Zhang, "Research on Plant Disease Identification Based on CNN," *Cognit. Robot*, vol.2, pp.155-163, 2022, doi: [10.1016/j.cogr.2022.07.001](https://doi.org/10.1016/j.cogr.2022.07.001)
- [13] C. Tan, F. Sun, and T. Kong, "A Survey on Deep Transfer Learning," in *International Conference on Artificial Neural Networks and Machine Learning*, 2018, pp. 270-279. doi: 10.1007/978-3-030-01424-7\_27
- [14] G. Thiodorus, A. Prasetya, L.A. Ardhani, and N. Yudistira, "Klasifikasi Citra Makanan/Nonmakanan Menggunakan Metode Transfer Learning dengan Model Residual Network," *J. Ilm. Sist. Inf.*, vol.11, no.2, pp.74-83, 2021, doi: 10.26594/teknologi.v11i2.2402
- [15] N. Awalia, and A. Primajaya, "Identifikasi Penyakit Leaf Molddaun Tomat Menggunakan Model Densenet121 Berbasis Transfer Learning," *J. Ilm. Ilmu Komput.*, vol.8, no.1, pp. 49-54, 2022, doi: 10.35329/jiik.v8i1.212
- [16] M. Sandler, A.G. Howard, M. Zhu, and A.



- Zhmoginov, L.C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474
- [17] J. Lu, X. Liu, X. Ma, J. Tong, and J. Peng, "Improved MobileNetV2 Crop Disease Identification Model for Intelligent Agriculture," *Peer J. Comput. Sci.*, vol. 9, no.e1595, 2023, doi: [10.7717/peerj-cs.1595](https://doi.org/10.7717/peerj-cs.1595)2023.
- [18] Y. Zhang and O. Yang, Q, *Medical Imaging and Augmented Reality*, Springer, Cham.
- [19] P. Toulis, T. Horel, and E.M. Airoidi "The Proximal Robbins-Monro Method," *J. R. Stat.*, vol.83, no.1, pp.188-2122, 2021, doi: 10.1111/rssb.12405.
- [20] M.D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," [arXiv preprint arXiv:1212.5701](https://arxiv.org/abs/1212.5701), 2012.
- [21] G. Hinton, N. Srivastava, and K. Swersky. Lecture 6a. Class Lecture, Topic : "Overview of Mini Batch Gradient Descent", Computer Science Department, University of Toronto, 2015.
- [22] D.P. Kingma and J.L. Ba, "Adam: A Method for Stochastic Optimization", in *International Conference on Learning Representations, 2015*.
- [23] T. Dozat, "Incorporating Nesterov Momentum into Adam," in *International Conference in Learning Representation Workshop*, Conference Proceeding, pp. 2013–2016, 2016.
- [24] S. Chaudhry, A. Faisal, and S. Khalil, "Identification of Overfitting in Deep Neural Networks using Sensitivity Analysis," in *IEEE International Conference on Emerging Trends in Communication, Control and Computing*, pp. 1-6, 2022.
- [25] Kaggle, "ImageNet Object Localization Challenge", 2020, <https://www.kaggle.com/c/imagenet-object-localization-challenge/data> (accessed Oct. 2, 2023).
- [26] H. Wang, Y. Guo, C. Wang, S. Liu, D. Zhang, and A.W.C. Liew, "Deep Transfer Learning-Based Identification of Tomato Plant Diseases," *Comput. Electron. Agric.*, vol.171, no.105310, 2020, doi:10.1016/j.compag.2020.105310
- [27] J. Chen, B. Li, X. Wang, and Y. Tian, "Transfer Learning for Plant Recognition Based on Neural Networks," in *5th International Conference on Computer and Communication Systems*, pp. 91-95, 2023.
- [28] L. Alzubaidi, J. Zhang, and A.J. Humaidi, "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *J. Big Data*, vol. 8, no. 53, 2021, doi:10.1186/s40537-021-00444-8
- [29] Y. Akhalifi and A. Subekti, "Bell Pepper Leaf Disease Classification Using Fine-Tuned Transfer Learning," *J. Elektron. Telekomun*, vol. 23, no.1, pp. 55-61, 2023, doi: [10.55981/jet.546](https://doi.org/10.55981/jet.546).