

PERBANDINGAN ARSITEKTUR MOBILENET DAN NASNETMOBILE UNTUK KLASIFIKASI PENYAKIT PADA CITRA DAUN KENTANG

Anwar Fuadi¹⁾, Aries Suharso²⁾

^{1, 2)} Fakultas Ilmu Komputer – Universitas Singaperbangsa Karawang
Jl. HS.Ronggo Waluyo, Telukjambe Timur, Kabupaten Karawang,
e-mail: anwar.fuadi18185@student.unsika.ac.id¹⁾, aries.suharso@unsika.ac.id²⁾

ABSTRAK

Tanaman kentang adalah tanaman yang dapat berkembang biak secara vegetatif melalui umbi. Kentang mempunyai peranan cukup penting untuk dimanfaatkan sebagai usaha. Penurunan produksi kentang 80% diakibatkan oleh penyakit yang dapat mengakibatkan penurunan produksi secara maksimal karena terjadi penghambatan pertumbuhan. Early blight dan late blight merupakan penyakit yang umum menyerang tanaman kentang. Penyakit tersebut dapat diidentifikasi melalui bercak dan warna pada daun tanaman kentang. Dengan majunya proses perkembangan teknologi komputasi maka proses identifikasi tersebut dapat dilakukan dengan menggunakan deep learning. Penelitian ini bertujuan untuk membandingkan arsitektur MobileNet dan NASNetMobile dalam melakukan deteksi penyakit pada daun tanaman kentang. Kedua arsitektur tersebut digunakan untuk menyelesaikan masalah klasifikasi gambar pada perangkat mobile. Penelitian ini menggunakan beberapa skema pemisahan data latih dan data uji seperti 90:10, 80:20, 70:30, 60:40, dan 50:50. Data yang digunakan pada penelitian ini terbagi menjadi gambar daun kentang sehat, gambar daun kentang terinfeksi Early Blight, dan gambar daun kentang terinfeksi Late Blight. Di akhir penelitian ditemukan bahwa pada skema pengujian menggunakan arsitektur NASNetMobile dengan perbandingan data latih dan data uji sebesar 90:10 menghasilkan accuracy sebesar 90.96%, precision sebesar 90.86%, recall sebesar 91.11%, dan f1 score sebesar 92.93%.

Kata Kunci: Deep Learning, Klasifikasi Gambar, MobileNet, NASNetMobile

ABSTRACT

Potato plants are plants that can reproduce vegetatively through tubers. Potatoes have an important role to be used as a business. The decline in potato production by 80% was caused by diseases that could result in a maximum decrease in production due to growth inhibition. Early blight and late blight are common diseases that attack potato plants. The disease can be identified by spotting and discoloration on the leaves of potato plants. With the advancement of the information technology development process, the identification process can be carried out using deep learning. This study aims to compare the architecture of MobileNet and NASNetMobile in detecting disease in potato leaves. Both architectures are used to solve image classification problems on mobile devices. This study uses several schemes for separating training data and test data such as 90:10, 80:20, 70:30, 60:40, and 50:50 to produce a good evaluation model. The data used in this study were divided into pictures of healthy potato leaves, pictures of potato leaves infected with early blight, and pictures of potato leaves infected with late blight. At the end of the study, it was found that the testing scheme using the NASNetMobile architecture with a comparison of training data and test data of 90:10 resulted accuracy of 90.96%, a precision of 90.86%, a recall of 91.11%, and an f1 score of 92.93%.

Keywords: Deep Learning, Image Classification, MobileNet, NASNetMobile

I. PENDAHULUAN

KENTANG merupakan salah satu tanaman dikotil dari keluarga *Solanaceae*. Tanaman kentang adalah tanaman yang dapat berkembang biak secara vegetatif melalui umbi. Tanaman kentang akan tumbuh subur pada daerah yang beriklim dingin [1]. Kentang mempunyai merupakan salah satu sumber karbohidrat untuk kebutuhan manusia [2]. Pada industri pangan, kentang merupakan salah satu bahan penting sehingga pada pertanian tanaman kentang perlu melakukan pengembangan, tetapi pada pengembangan pertanian tanaman kentang banyak memiliki permasalahan, salah satu permasalahan yang sering terjadi ialah adanya penyakit pada daun tanaman kentang yang dapat memengaruhi hasil produksi.

Penurunan produksi kentang 80% diakibatkan oleh penyakit yang dapat mengakibatkan penurunan produksi secara maksimal karena terjadi penghambatan pertumbuhan [1]. Diduga jamur penyebab penyakit ini berasal dari umbi-umbi bibit. Tumbuhnya jamur pada umbi kentang dipengaruhi oleh suhu. Jamur tersebut dapat menghasilkan senyawa bersifat racun yaitu solanin. Secara umum ada dua penyakit akibat jamur yang menyerang tanaman kentang yaitu *early bright* dan *late blight*.

Early blight disebabkan oleh dua jamur yang berkerabat dekat yaitu *Alternaria tomatophila* dan *Alternaria solani* yang hidup di tanah dan sisa-sisa tanaman [3]. Gejala awal penyakit pertama kali muncul di pangkal tanaman yang terserang, dengan bintik-bintik coklat bulat kasar pada daun dan batang lalu menghasilkan lesi berwarna gelap yang dikenali oleh cincin konsentrisnya yang khas dengan diameter sekitar 3 hingga 12 mm [4]. Sedangkan *late blight* disebabkan oleh *Phytophthora infestans* yaitu suatu mikroorganisme yang menyukai lingkungan lembab dan sejuk [5]. Tanda awal penyakit busuk daun yang paling jelas adalah bercak coklat atau bercak penyakit pada batang tanaman.

Mengklasifikasi penyakit yang menyerang daun tanaman kentang dapat dilakukan dengan menggunakan pendekatan *deep learning* yaitu metode *transfer learning*. Metode ini lebih efektif, efisien dan obyektif serta aman dimana semua prosesnya dilakukan oleh sistem [6]. Deep Learning merupakan sebuah teknik baru dalam mempelajari representasi data. *Deep learning* mengacu pada teknik *machine learning* pada jaringan syaraf tiruan dengan 9 *hidden layers* untuk meningkatkan performa [7]. Pendekatan *deep learning* sering diimplementasikan dalam berbagai aplikasi kehidupan nyata dan dapat dilakukan secara otomatis oleh komputer seperti dalam pemrosesan gambar digital, biometrik, dan sebagainya.

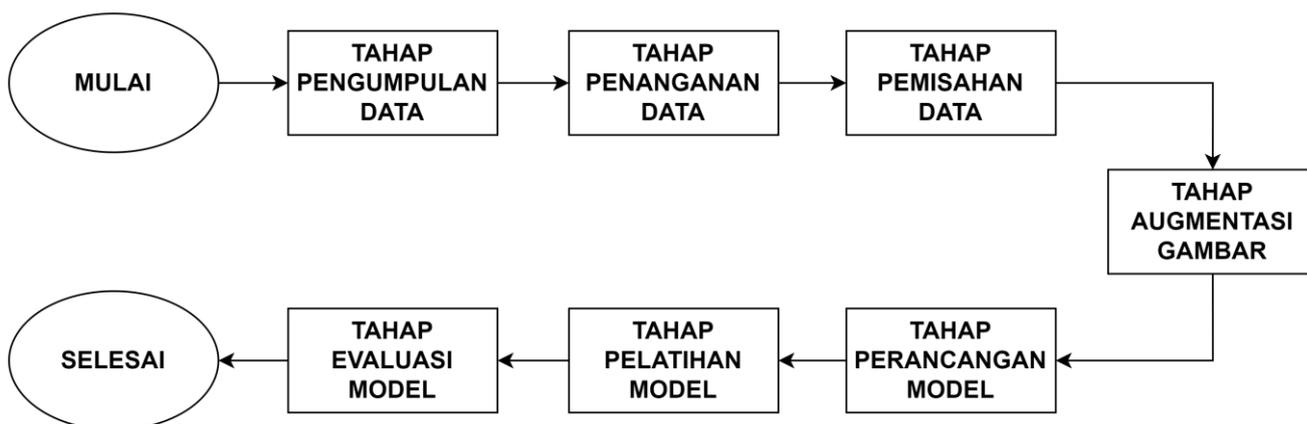
Transfer learning merupakan teknik yang menggunakan model yang terlatih yang digunakan untuk mengklasifikasikan kumpulan dataset baru sehingga tidak perlu dilakukan pelatihan data terlebih dahulu [8]. Tujuan dari *Transfer learning* adalah untuk meningkatkan pembelajaran pada tugas yang baru ditargetkan memanfaatkan pengetahuan dari tugas yang berasal dari sumber sebelumnya [9].

Tujuan dari penelitian ini yaitu untuk membandingkan kinerja arsitektur NASNetMobile dan MobileNet dalam melakukan deteksi penyakit pada daun kentang melalui gambar (*image classification*) berdasarkan hasil evaluasi model. NASNetMobile dan MobileNet merupakan arsitektur yang ringan dan memiliki kinerja yang cepat yang dapat diimplementasikan pada perangkat *mobile* [10]. NASNetMobile adalah arsitektur NASNet dengan parameter lebih sedikit daripada arsitektur MobileNet dan dikembangkan oleh Google Inc, NASNetMobile adalah salah satu varian arsitektur NASNet, pengembangan arsitektur NASNetMobile difokuskan pada *embedded system* dan perangkat seluler [11]. Sedangkan MobileNet adalah model arsitektur CNN yang telah dilatih sebelumnya menggunakan konvolusi yang dapat dipisahkan secara mendalam untuk membangun bobot yang ringan pada jaringan saraf [12].

Penelitian sebelumnya tentang klasifikasi objek menggunakan metode *transfer learning* sudah cukup banyak, salah satunya ialah pada penelitian yang berjudul *Face Attribute Detection with MobileNetV2 and NasNet-Mobile*. Pada penelitian tersebut melakukan percobaan deteksi atribut wajah menggunakan metode *transfer learning* dengan menggunakan pendekatan *deep learning* yaitu metode *transfer learning* arsitektur CNN. Pada penelitian ini melakukan percobaan dengan model MobileNet dan NASNetMobile menyimpulkan dari hasil percobaan bahwa NASNetMobile mendapatkan hasil yang terbaik karena memiliki akurasi sebesar 91.6% dan MobileNet mendapatkan hasil akurasi sebesar 91.5% [13].

II. METODE

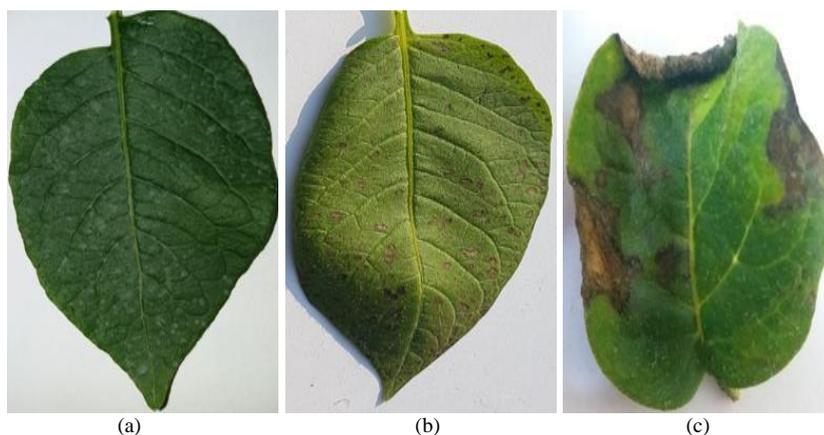
Tahapan awal pada penelitian ini dimulai dengan melakukan pengumpulan data yang akan digunakan untuk membuat model *deep learning*. Kemudian data tersebut melalui proses penanganan data tidak seimbang (*imbalanced data*) dan pemisahan menjadi data latih dan data uji. Selanjutnya data melalui proses augmentasi gambar guna memperkaya data. Setelah itu, penelitian ini akan masuk ke tahap perancangan dan pelatihan model. Diakhir penelitian dilakukan evaluasi model yang sudah dilatih sebelumnya. Tahapan penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Tahapan Metodologi Penelitian

A. Tahap Pengumpulan Data

Penelitian ini menggunakan data yang diambil dari situs Kaggle dan bersifat publik. Data tersebut berisi 4072 gambar yang dikumpulkan dari wilayah Punjab Tengah di Pakistan dan terbagi menjadi 3 kategori yaitu daun kentang sehat (*healthy*), daun kentang yang terinfeksi virus *Early Blight*, dan daun kentang yang terinfeksi virus *Late Blight* [14]. Gambar daun kentang yang sehat, terinfeksi virus *Early Blight*, dan d terinfeksi virus *Late Blight* dapat dilihat pada gambar 2.

Gambar 2. Gambar Daun Kentang a) Sehat, b) Terinfeksi Virus *Early Blight*, c) Terinfeksi Virus *Late Blight*

Selanjutnya karena data tersebut terbagi menjadi 3 bagian yaitu *training*, *validation*, dan *testing* maka data akan digabungkan berdasarkan kategori penyakitnya untuk mempermudah pada tahap selanjutnya. Oleh karena itu maka persebaran data pada masing – masing kategori menjadi 1020 gambar daun kentang sehat (*healthy*), 1628 gambar daun kentang yang terinfeksi virus *Early Blight*, dan 1424 gambar daun kentang yang terinfeksi virus *Late Blight*.

B. Tahap Penanganan Data

Klasifikasi data tidak seimbang adalah masalah di mana ukuran kelas proporsional dari kumpulan data berbeda relatif dengan margin yang cukup besar. Dalam hal ini, satu kelas setidaknya digambarkan hanya dengan beberapa nomor sampel (disebut kelas minoritas) dan sisanya termasuk ke dalam kelas yang lain (disebut kelas mayoritas) [15]. Untuk mengatasi hal tersebut maka diperlukan penanganan data terhadap data yang tidak seimbang dan salah satunya adalah *oversampling*. *Oversampling* adalah teknik untuk mengubah kelas data yang tidak sama untuk membuat set data yang seimbang. Teknik ini mencoba untuk menambah ukuran sampel langka untuk menciptakan keseimbangan ketika data tidak mencukupi. Proses *oversampling* pada penelitian ini dapat dilihat pada Tabel I.

TABEL I
PROSES OVER SAMPLING

Kategori	Jumlah Data Sebelum Proses Over Sampling	Jumlah Data Sesudah Proses Over Sampling
Sehat (<i>healthy</i>)	1020	1628
<i>Early Blight</i>	1628	1628
<i>Late Blight</i>	1424	1628

C. Tahap Pemisahan Data

Pada tahap ini data akan dipisah menjadi data latih dan data uji. Data latih berfungsi untuk melatih model *machine learning* berdasarkan metode yang telah ditentukan sebelumnya. Lalu data uji berfungsi untuk memvalidasi kinerja model *machine learning* yang dihasilkan. Proses pemisahan data terbagi menjadi beberapa skema seperti 90:10, 80:20, 70:30, 60:40, dan 50:50. Hal tersebut dilakukan untuk membuktikan hubungan antara pemisahan data terhadap kinerja model *machine learning* yang dihasilkan. Proses pemisahan data menjadi data latih dan data uji dapat dilihat pada Tabel II.

TABEL II
DISTRIBUSI DATA LATIH DAN DATA UJI

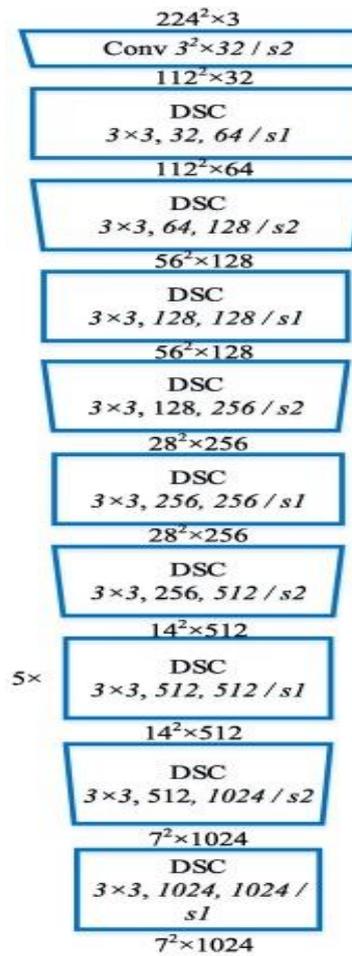
Skema	Arsitektur	Perbandingan Data Latih dan Data Uji	Jumlah Data Latih	Jumlah Data Uji
1	MobileNet	90:10	4395	489
2	MobileNet	80:20	3906	978
3	MobileNet	70:30	3420	1464
4	MobileNet	60:40	2931	1953
5	MobileNet	50:50	2442	2442
6	NASNetMobile	90:10	4395	489
7	NASNetMobile	80:20	3906	978
8	NASNetMobile	70:30	3420	1464
9	NASNetMobile	60:40	2931	1953
10	NASNetMobile	50:50	2442	2442

D. Tahap Augmentasi Gambar

Augmentasi gambar adalah teknik yang sering digunakan dalam visi komputer dan telah melihat peningkatan minat sejak popularitas pembelajaran mendalam. Kegunaannya menjadi semakin dikenal karena jaringan saraf membutuhkan jumlah data yang lebih besar untuk dilatih, dan karena dalam bidang tertentu, seperti pencitraan biomedis sejumlah besar data berlabel sulit didapat atau mahal untuk diproduksi [16]. Parameter augmentasi gambar yang digunakan pada penelitian ini antara lain *rescale*, *rotation range*, *vertical flip*, *shear range*, *zoom range*, dan *fill mode* untuk data latih. Sedangkan untuk data uji hanya menggunakan parameter *rescale*.

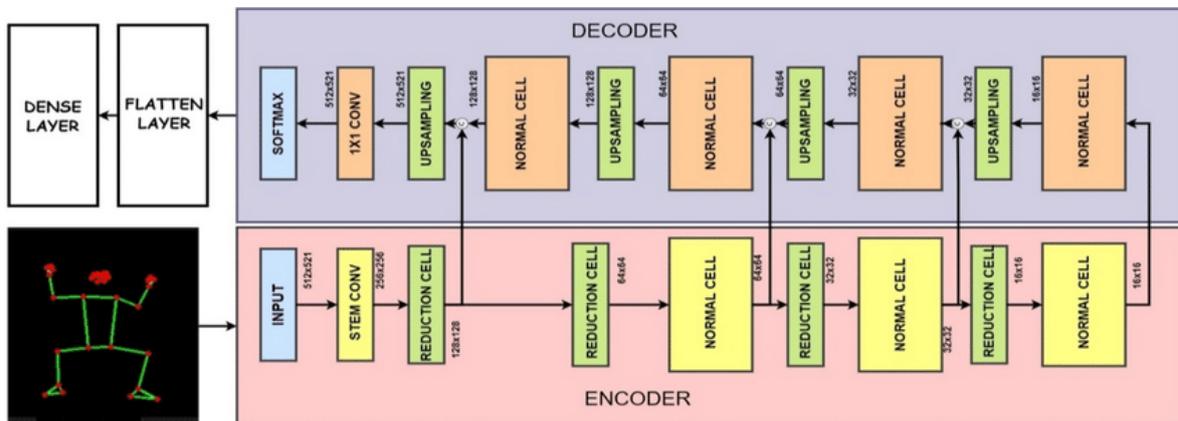
E. Tahap Perancangan Model

MobileNet merupakan salah satu arsitektur yang dapat digunakan untuk mengatasi kebutuhan akan *computing resource* yang berlebih. MobileNet memiliki daya komputasi yang sangat sedikit untuk menjalankan atau menerapkan *transfer learning*. Hal tersebut sangat cocok untuk perangkat seluler, *embedded system*, dan komputer tanpa GPU atau efisiensi komputasi yang rendah dengan mengorbankan akurasi hasil secara signifikan. Lapisan konvolusi MobileNet diklasifikasikan menjadi sepuluh blok, yang pertama menggunakan konvolusi standar yang menghasilkan 32 fitur, sedangkan blok berikutnya menggunakan DSC dan *down-sampling* dengan *max-pooling*. *Feature map* meningkat dengan perkalian biner hingga 1024 fitur di blok terakhir [17].



Gambar 3. Rincian Arsitektur MobileNet

NASNetMobile adalah jaringan saraf konvolusional yang dilatih pada lebih dari satu juta gambar dari database ImageNet. Jaringan dapat mengklasifikasikan gambar ke dalam 1000 kategori objek, seperti keyboard, mouse, pensil, dan banyak binatang. Akibatnya, jaringan telah mempelajari representasi fitur yang kaya untuk berbagai gambar. NASNetMobile memiliki parameter lebih sedikit daripada arsitektur MobileNet. Arsitektur ini memiliki dua blok utama, sering disebut sel reguler dan sel tereduksi. Susunan *layer* yang benar dari kedua sel ini ditentukan menggunakan *recurrent neural network*.



Gambar 3. Arsitektur NASNetMobile

F. Tahap Pelatihan Model

Pada tahap ini model akan masuk ke tahap *fine tuning*, model yang sudah dibuat akan dibekukan atau *freeze* supaya nilai weights tidak berubah dan model tidak akan dilatih ulang. Setelah itu model akan melalui proses

compiling dengan optimizer Adam dan nilai *learning rate* yaitu 0.0001 supaya tingkat ketelitian model semakin tinggi. Selain itu metrik pengukuran model *machine learning* menggunakan loss yang bernilai *categorical_crossentropy* karena pada penelitian ini menggunakan 3 kategori atau label pada data. Proses selanjutnya yaitu melatih model *machine learning*. Parameter yang digunakan untuk melatih model antara *epochs* atau banyak iterasi dalam melatih model sebanyak 50 supaya model mencapai akurasi dan loss secara maksimum. Selain itu *verbose* yang digunakan bernilai 1 supaya proses pelatihan model dapat terlihat.

G. Tahap Evaluasi Model

Evaluasi model merupakan bagian dari proses pengembangan model. Ini membantu untuk menemukan model terbaik yang mewakili data yang digunakan dan seberapa baik model yang dipilih akan bekerja dengan baik di masa depan. Mengevaluasi kinerja model dengan data yang digunakan untuk pelatihan tidak dapat diterima dalam ilmu data karena dapat dengan mudah menghasilkan model yang terlalu optimis dan terlalu cocok sehingga evaluasi digunakan untuk data uji. Pada penelitian ini digunakan empat parameter untuk mengevaluasi model antara lain *accuracy*, *precision*, *recall*, dan *f1 score*. Parameter tersebut dapat dirumuskan sebagai berikut:

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{True Positive} + \text{True Negative}} \quad (1)$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

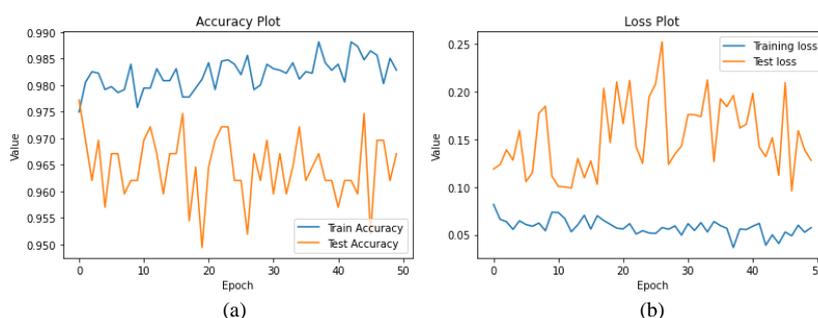
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (3)$$

$$\text{F1 score} = \frac{2 \times \text{Recall} \times \text{Presisi}}{\text{Recall} + \text{Presisi}} \quad (4)$$

III. HASIL DAN PEMBAHASAN

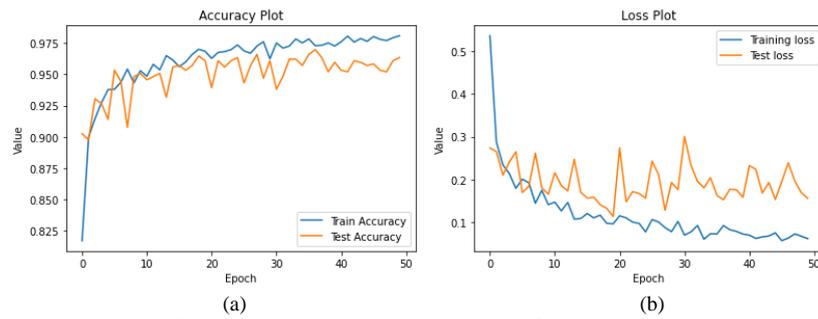
A. Hasil Pelatihan Model

Pada skema 1 perbandingan data latih dan data uji sebesar 90:10 dan menggunakan arsitektur MobileNet menghasilkan model yang *underfitting*. Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang memiliki perbedaan nilai yang cukup jauh. Sedangkan pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.

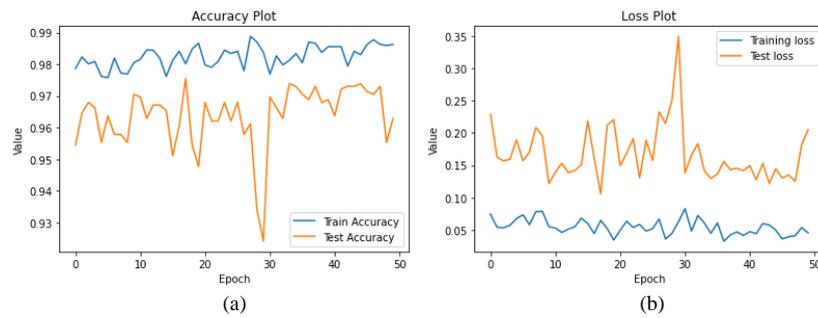


Gambar 3. Visualisasi a) *Accuracy* dan b) *Loss* pada skema 1

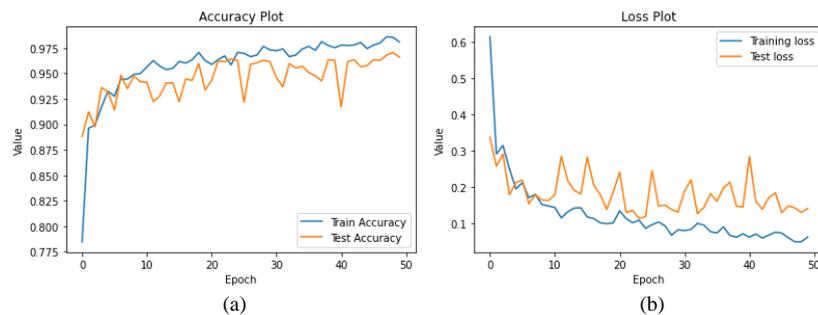
Pada skema 2 perbandingan data latih dan data uji sebesar 80:20 dan menggunakan arsitektur MobileNet menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Kemudian pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.

Gambar 4. Visualisasi a) *Accuracy* dan b) *Loss* pada skema 2

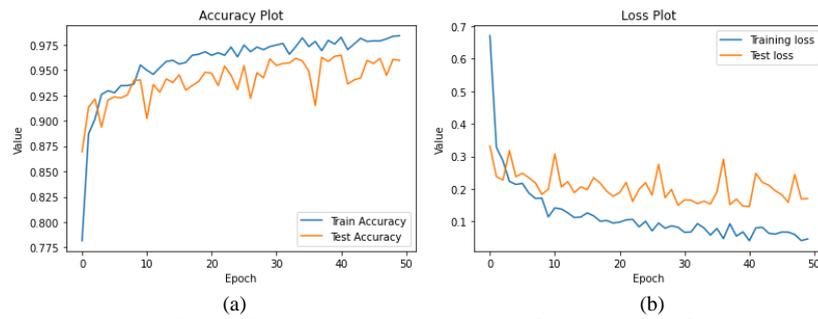
Pada skema 3 perbandingan data latih dan data uji sebesar 70:30 dan menggunakan arsitektur MobileNet menghasilkan model yang *underfitting*. Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang memiliki perbedaan nilai yang cukup jauh. Selanjutnya pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.

Gambar 5. Visualisasi a) *Accuracy* dan b) *Loss* pada skema 3

Pada skema 4 perbandingan data latih dan data uji sebesar 60:40 dan menggunakan arsitektur MobileNet menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Selanjutnya pada parameter *train accuracy* dan *test accuracy* juga tidak memiliki perbedaan nilai yang jauh.

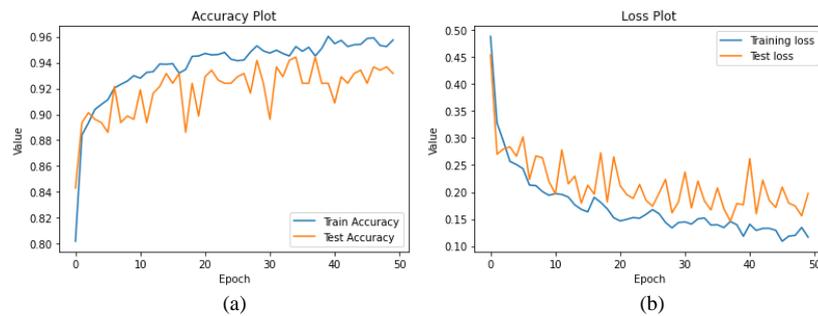
Gambar 6. Visualisasi a) *Accuracy* dan b) *Loss* pada skema 4

Pada skema 5 perbandingan data latih dan data uji sebesar 50:50 dan menggunakan arsitektur MobileNet menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Kemudian pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.



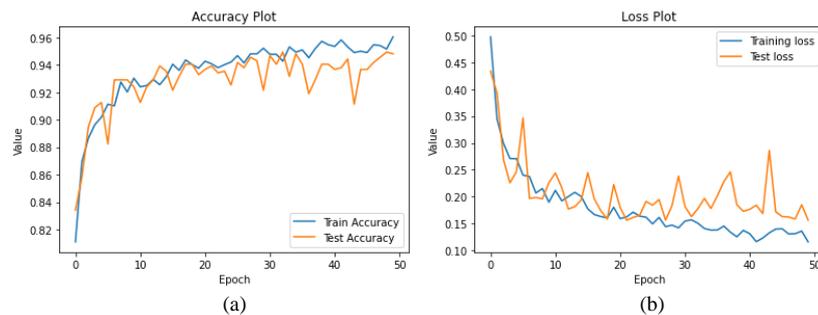
Gambar 7. Visualisasi a) Accuracy dan b) Loss pada skema 5

Pada skema 6 perbandingan data latih dan data uji sebesar 90:10 dan menggunakan arsitektur NASNetMobile menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Kemudian pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.



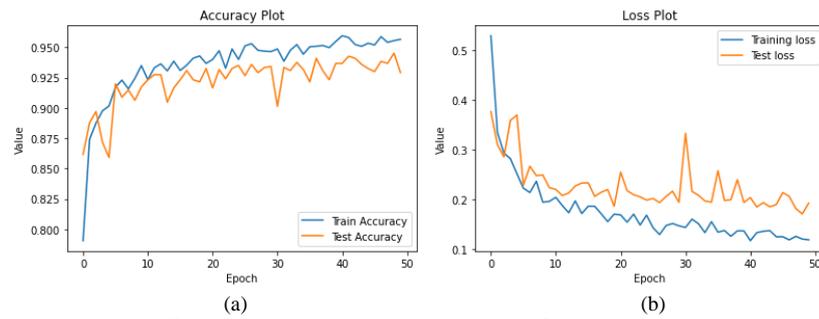
Gambar 8. Visualisasi a) Accuracy dan b) Loss pada skema 6

Pada skema 7 perbandingan data latih dan data uji sebesar 80:20 dan menggunakan arsitektur NASNetMobile menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Kemudian pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.



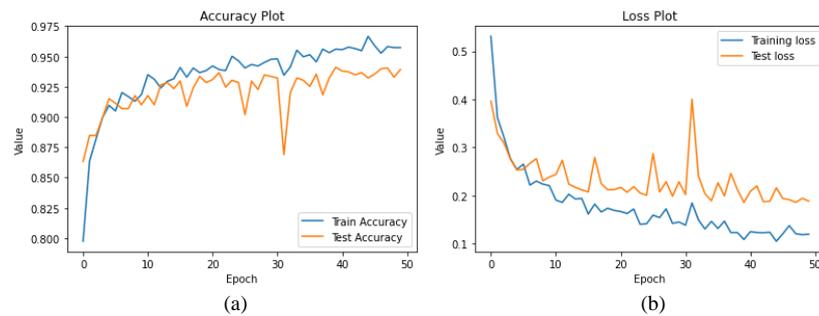
Gambar 9. Visualisasi a) Accuracy dan b) Loss pada skema 7

Pada skema 8 perbandingan data latih dan data uji sebesar 70:30 dan menggunakan arsitektur NASNetMobile menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Kemudian pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.



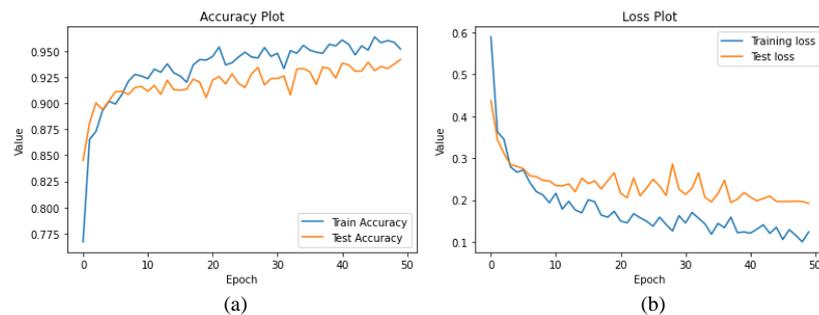
Gambar 10. Visualisasi a) Accuracy dan b) Loss pada skema 8

Pada skema 9 perbandingan data latih dan data uji sebesar 60:40 dan menggunakan arsitektur NASNetMobile menghasilkan model yang *underfitting* pada parameter *loss*. Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang memiliki perbedaan nilai yang jauh di akhir *epochs*. Namun pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.



Gambar 11. Visualisasi a) Accuracy dan b) Loss pada skema 9

Pada skema 10 perbandingan data latih dan data uji sebesar 50:50 dan menggunakan arsitektur NASNetMobile menghasilkan model yang baik (tidak mengalami *overfitting* ataupun *underfitting*). Hal tersebut ditandai dengan parameter *training loss* dan *test loss* yang tidak memiliki perbedaan nilai yang jauh. Kemudian pada parameter *train accuracy* dan *test accuracy* tidak memiliki perbedaan nilai yang jauh.



Gambar 12. Visualisasi a) Accuracy dan b) Loss pada skema 10

B. Hasil Evaluasi Model

Selanjutnya untuk membandingkan kinerja model *machine learning* yang menggunakan arsitektur MobileNet dan model yang menggunakan arsitektur NASNetMobile pada seluruh skema akan digunakan nilai rata - rata dari nilai parameter *akurasi*, *precision*, *recall*, dan *f1 score*. Untuk mempermudah perbandingan antar skema maka seluruh nilai yang dihasilkan ditampilkan dalam bentuk persentase. Hasil nilai rata - rata *confusion matrix* dengan arsitektur MobileNet dan arsitektur NASNetMobile seluruh skema dapat dilihat pada Tabel V.

TABEL V
HASIL *CONFUSION MATRIX* PADA SELURUH SKEMA PENGUJIAN

Skema	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>
1	81.60%	81.64%	81.64%	81.50%
2	79.14%	79.35%	77.13%	77.89%
3	77.46%	77.33%	77.47%	77.37%
4	75.88%	75.87%	76.07%	75.74%
5	74.82%	74.81%	74.99%	74.78%
6	90.96%	90.86%	91.11%	92.93%
7	88.82%	88.98%	88.82%	88.84%
8	86.48%	86.65%	86.47%	86.50%
9	84.49%	84.75%	84.65%	84.66%
10	81.82%	81.83%	81.87%	81.85%

IV. KESIMPULAN

Setelah melakukan penelitian ini maka dapat diambil kesimpulan yaitu model yang menggunakan arsitektur NASNetMobile pada skema 6 menghasilkan hasil evaluasi model yang lebih baik diantara arsitektur dan skema lainnya. Skema 6 menghasilkan *accuracy* sebesar 90.96%, *precision* sebesar 90.86%, *recall* sebesar 91.11%, dan *f1 score* sebesar 92.93%.

Untuk penelitian selanjutnya disarankan menggunakan teknik penanganan data berbeda dari penelitian ini seperti *under sampling* ataupun tanpa penanganan data. Selain itu dapat dilakukan konfigurasi *fine tuning* dengan parameter yang berbeda sehingga dapat mengetahui pengaruhnya terhadap model yang dihasilkan.

DAFTAR PUSTAKA

- [1] Cucu Oktaviana and Dini Destiani Siti Fatimah, "Rancang Bangun Sistem Pakar Penanganan Penyakit dan Hama Tanaman Kentang", *Jurnal Algoritma*, vol. 14, no. 1, pp. 51 - 60, Aug. 2017.
- [2] D. Mulyono, M. J. A. Syah, A. L. Sayekti, and Y. Hilman, "Kelas Benih Kentang (*Solanum tuberosum* L.) Berdasarkan Pertumbuhan, Produksi, dan Mutu Produk," *Jurnal Hortikultura*, vol. 27, no. 2, p. 209, Feb. 2018, doi: 10.21082/jhort.v27n2.2017.p209-216.
- [3] L. Meno, O. Escuredo, M. S. Rodríguez-Flores, and M. C. Seijo, "Modification of the TOMCAST Model with Aerobiological Data for Management of Potato Early Blight," *Agronomy*, vol. 10, no. 12, p. 1872, Nov. 2020, doi: 10.3390/agronomy10121872.
- [4] J. E. Van der Waals, L. Korsten, and T. A. S. Aveling, "A review of early blight of potato," *African Plant Prot.*, vol. 7, no. 2, pp. 91–102, 2001, [Online]. Available: <http://content.ajarchive.org/cgi-bin/showfile.exe?CISOROOT=/10233121&CISOPTR=67%5Cnhttp://hdl.handle.net/10499/AJ4913>.
- [5] A. Majeed, Z. Muhammad, Z. Ullah, R. Ullah, and H. Ahmad, "Late Blight of Potato (*Phytophthora infestans*) I: Fungicides Application and Associated Challenges," *Turkish Journal of Agriculture - Food Science and Technology*, vol. 5, no. 3, p. 261, Mar. 2017, doi: 10.24925/turjaf.v5i3.261-266.1038.
- [6] C. Su and W. Wang, "Concrete Cracks Detection Using Convolutional NeuralNetwork Based on Transfer Learning," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, Oct. 2020, doi: 10.1155/2020/7240129.
- [7] B. Kim and S. Cho, "Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique," *Sensors*, vol. 18, no. 10, p. 3452, Oct. 2018, doi: 10.3390/s18103452.
- [8] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning," *Artificial Neural Networks and Machine Learning – ICANN 2018*, pp. 270–279, 2018, doi: 10.1007/978-3-030-01424-7_27.
- [9] E. S. Olivas, J. D. M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, and A. J. Serrano López, Eds., "Handbook of Research on Machine Learning Applications and Trends," 2010, doi: 10.4018/978-1-60566-766-9.
- [10] F. Saxen, P. Werner, S. Handrich, E. Othman, L. Dinges, and A. Al-Hamadi, "Face Attribute Detection with MobileNetV2 and NasNet-Mobile," 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Sep. 2019, doi: 10.1109/ispa.2019.8868585.
- [11] B. P. Amiruddin and R. E. Abdul Kadir, "CNN Architectures Performance Evaluation for Image Classification of Mosquito in Indonesia," *IEEE Xplore*, Jul. 01, 2020. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=9163732> (accessed Dec. 02, 2021).
- [12] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [13] F. Saxen, P. Werner, S. Handrich, E. Othman, L. Dinges, and A. Al-Hamadi, "Face Attribute Detection with MobileNetV2 and NasNet-Mobile," 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), Sep. 2019, doi: 10.1109/ispa.2019.8868585.
- [14] Saeed, R., 2022. Potato Disease Leaf Dataset (PLD). [online] www.kaggle.com. Tersedia: <https://www.kaggle.com/datasets/rizwan123456789/potato-disease-leaf-datasetpld> [Diakses pada 12 Juni 2022].
- [15] H. Kaur, H. S. Pannu, and A. K. Malhi, "A Systematic Review on Imbalanced Data Challenges in Machine Learning," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1–36, Sep. 2019, doi: 10.1145/3343440.
- [16] M. D. Bloice, P. M. Roth, and A. Holzinger, "Biomedical image augmentation using Augmentor," *Bioinformatics*, vol. 35, no. 21, pp. 4522–4524, Apr. 2019, doi: 10.1093/bioinformatics/btz259.
- [17] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, N. Suciati, and C. Fatchah, "Combining MobileNetV1 and Depthwise Separable convolution bottleneck with Expansion for classifying the freshness of fish eyes," *Information Processing in Agriculture*, Jan. 2022, doi: 10.1016/j.inpa.2022.01.002.