

IMPLEMENTASI ARSITEKTUR *ALEXNET* DAN *RESNET34* PADA KLASIFIKASI CITRA PENYAKIT DAUN KENTANG MENGGUNAKAN *TRANSFER LEARNING*

Mochammad Kevin Santosa, Made Hanindia Prami Swari, Anderas Nugroho Sihananto

Informatika, Universitas Pembangunan Nasional Veteran Jawa Timur

Jl Raya Rungkut Madya Gunung Anyar Surabaya, Indonesia

santosakevin0@gmail.com

ABSTRAK

Penelitian ini mengimplementasikan arsitektur *Alexnet* dan *Resnet34* dengan teknik *transfer learning* pada klasifikasi citra penyakit daun kentang. Kentang adalah komoditas penting dalam pertanian global, tetapi sering terkena penyakit seperti *Late blight* dan *Early blight*, yang mengancam hasil panen. Penelitian bertujuan untuk mengevaluasi akurasi *transfer learning* menggunakan *Alexnet* dan *Resnet34* dalam mengklasifikasikan penyakit daun kentang. Kami juga membandingkan waktu pelatihan kedua model. Tujuan penelitian adalah mencapai tingkat akurasi yang tinggi dalam identifikasi penyakit daun kentang untuk membantu petani mengambil tindakan pencegahan yang cepat dan efektif. Kami melatih kedua model selama 16 *Epoch* dengan batch size 14 dan parameter yang telah ditentukan. *Learning rate* dan *weight decay* diatur pada nilai tertentu. Model *Alexnet* mencapai akurasi 98% dengan waktu pelatihan 82 menit, sedangkan model *Resnet34* mencapai akurasi 99% dalam waktu yang sama. Kedua model mampu memberikan hasil identifikasi yang sangat baik pada citra penyakit daun kentang. Penelitian ini penting untuk meningkatkan produktivitas pertanian dan memberikan solusi efisien dalam mengatasi masalah penyakit pada tanaman kentang.

Kata kunci : *Alexnet*, *Resnet34*, *Transfer Learning*, Daun Tanaman Kentang

1. PENDAHULUAN

Kentang (*Solanum tuberosum*) merupakan komoditas yang telah ditanam di sekitar 19 juta hektar lahan pertanian di seluruh dunia dan hasil produksinya telah mencapai 378 ton. Di Indonesia, total area panen kentang 10 tahun terakhir berfluktuasi antara 66.00 hingga 76.000 hektar [1]. Berdasarkan data dari Badan Pusat Statistik (BPS), produksi kentang di Indonesia pada tahun 2021 mencapai 1,36 juta ton, mengalami kenaikan sebesar 6,1% dari tahun sebelumnya yang sebesar 1,28 juta ton. Dalam mengelola pertanian kentang tentu saja akan menemui beberapa kendala. Penyakit busuk daun (*Late blight*) dan penyakit busuk daun awal (*Early blight*) merupakan penyakit daun yang paling merusak tanaman kentang dan menyebabkan penurunan hasil panen yang signifikan di sebagian besar wilayah pertanian kentang di seluruh dunia.

Banyak upaya dilakukan untuk mengatasi masalah penyakit pada daun kentang, tidak hanya di bidang pertanian tetapi juga melalui pemanfaatan teknologi. Salah satu contohnya adalah penggunaan *image processing* atau pengolahan citra digital untuk mengidentifikasi penyakit pada tanaman kentang. Dengan menggunakan teknologi ini, pengelola pertanian dapat memberikan penanganan yang efektif dan efisien pada tanaman yang tidak sehat [2].

Transfer learning merupakan sebuah teknik *machine learning* yang dapat meningkatkan kinerja pada tugas tertentu dengan memanfaatkan pengetahuan yang telah diperoleh oleh model yang telah dilatih sebelumnya pada tugas atau domain yang berbeda. Dalam implementasi *transfer learning* pada *Convolutional Neural Network* (CNN), lapisan fitur yang telah dipelajari dari satu CNN dapat digunakan

kembali pada CNN lainnya untuk menginisialisasi jaringan tersebut [3].

CNN mempunyai beragam model arsitektur seperti LeNet, *Alexnet*, *Resnet*, GoogleNet, SqueezeNet, dan lain-lain yang terus berkembang seiring kemajuan teknologi. Setiap model arsitektur memiliki spesifikasi yang berbeda, termasuk proses komputasi dan hasil akurasi yang berbeda. Misalnya, sebuah penelitian yang dilakukan oleh Setiawan (2020) menggunakan CNN untuk mengklasifikasikan Fundus dengan menggunakan beberapa parameter model arsitektur seperti *Alexnet*, VGG16, VGG19, *Resnet50*, *Resnet101*, GoogleNet, Inception-V3, Inception-ResnetV2, dan SqueezeNet. Penelitian ini menggunakan dataset dengan 2 kategori, yaitu normal dan neovaskularisasi. Hasil dari penelitian tersebut menunjukkan bahwa setiap model arsitektur yang digunakan memiliki nilai akurasi yang berbeda, yang dipengaruhi oleh berbagai faktor komponen seperti dataset, proses komputasi arsitektur, dan Optimizer yang digunakan [4].

Penelitian ini bertujuan untuk menghasilkan sebuah sistem yang dapat membantu para petani pengelola pertanian dalam mengidentifikasi penyakit pada daun kentang dengan menggunakan data berupa gambar daun kentang. Identifikasi daun pada tanaman kentang dibagi menjadi tiga kategori, yaitu daun sehat atau normal, *Late blight*, dan *Early blight*. Oleh karena itu, pada penelitian ini akan menggunakan *transfer learning* dengan arsitektur *Alexnet* dan *Resnet34* untuk melakukan identifikasi tersebut.

2. TINJAUAN PUSTAKA

Teori dasar yang akan dipakai untuk penelitian diperlukan untuk mendukung penelitian ini. Landasan

teori yang akan digunakan adalah Kecerdasan Buatan, Citra Digital, Machine Learning, Deep Learning, Alexnet, Resnet34, Adam Optimizer, Transfer Learning.

2.1. Kecerdasan Buatan

Pada tahun 1955, John McCarthy, pelopor kecerdasan buatan, pertama kali mendefinisikan istilah tersebut sebagai tujuan mengembangkan mesin yang berperilaku cerdas. Definisi Encyclopedia Britannica menyebutkan kecerdasan buatan sebagai kemampuan komputer atau robot yang dikendalikan komputer untuk pemecahan masalah dengan kemampuan pemrosesan intelektual melebihi manusia. Namun, definisi ini rentan pada kelemahan karena bisa menganggap komputer dengan kapasitas memori besar dan akses teks panjang sebagai cerdas. Elaine Rich mengusulkan definisi yang lebih tepat, menyatakan bahwa kecerdasan buatan adalah studi membuat komputer melakukan tugas yang lebih baik dikerjakan oleh manusia. [5].

2.2. Citra Digital

Citra digital adalah representasi dua dimensi dari gambar analog melalui proses sampling, yang menghasilkan gambar diskrit dengan N baris dan M kolom. Pengolahan citra digital dilakukan menggunakan komputer dengan nilai intensitas pada setiap piksel sebagai datanya. Ini melibatkan perbaikan kualitas gambar, transformasi seperti translasi dan rotasi, seleksi citra ciri, penyimpanan data tereduksi, kompresi, transmisi data, dan proses waktu. Secara esensial, pengolahan citra digital adalah manipulasi gambar dua dimensi dengan representasi array nilai real atau kompleks, terdiri dari piksel-piksel dengan koordinat dan intensitas/warna yang direpresentasikan sebagai $f(x,y)$ pada titik (x,y) dalam matriks $M \times N$. [6]

2.3. Machine Learning

Machine learning merupakan suatu teknik yang memungkinkan mesin untuk mampu melakukan tugas-tugas yang kompleks, salah satunya adalah kemampuan untuk mengenali objek dalam gambar. Cara pembelajaran mesin dilakukan melalui proses trial and error, di mana algoritme diberikan contoh-contoh dan kemudian secara otomatis mengoreksi kesalahan-kesalahan yang terjadi, bukan melalui cara-cara eksplisit seperti memprogramkan serangkaian instruksi secara manual [7].

2.4. Deep Learning

Deep learning adalah sub-bidang kecerdasan buatan yang fokus pada pengembangan model jaringan saraf besar untuk pengambilan keputusan berbasis data yang akurat. Metode ini cocok untuk situasi dengan data kompleks dan besar. Dalam deep learning, model jaringan saraf mewakili fungsi yang diekstraksi dari data selama pelatihan, dan sebaliknya, model ini juga mewakili fungsi sebagai program komputer. Proses

pelatihan dimulai dengan inisialisasi parameter jaringan secara acak, dan model jaringan saraf cenderung overfitting, bukan underfitting, terhadap data karena bias induktifnya yang lemah [8].

2.5. Alexnet

Salah satu struktur Convolutional Neural Network (CNN) yang paling terkenal adalah Alexnet, yang dikembangkan oleh Krizhevsky dan Sutskever. Alexnet mampu mencapai tingkat akurasi klasifikasi yang sangat tinggi dalam dataset ImageNet, sehingga dianggap sebagai suatu terobosan signifikan dalam bidang pembelajaran mesin. Sejak pengembangan Alexnet, peneliti mulai memusatkan perhatian lebih dalam dan menginvestasikan lebih banyak waktu serta upaya dalam mengembangkan model-model pembelajaran mendalam [9].

2.6. Resnet34

Jaringan syaraf terinspirasi oleh cara berpikir manusia mampu menangani masalah kompleks dengan solusi mendalam. Tantangan utama jaringan syaraf adalah masalah gradien yang menghilang. Resnet, jenis jaringan khusus, mengatasi ini dengan koneksi lompatan, memungkinkan "loncatan" lapisan konvolusi dalam setiap blok dasar. Ini mempercepat dan mempermudah pelatihan jaringan. Persamaan $y = F(x) + x$ menggambarkan koneksi lompatan yang menambahkan output blok sebelumnya ke blok berikutnya. Resnet menggunakan fungsi residual $f(x)$ dengan lapisan konvolusi ganda, normalisasi batch, dan ReLU. Resnet telah menghasilkan kinerja lebih baik dalam jaringan syaraf, dengan kemampuan pembelajaran yang lebih dalam. Resnet34 dengan 34 lapisan termasuk konvolusi, max-pooling, dan average pooling, serta lapisan fully connected, menjadi contoh arsitektur yang sukses dalam penggunaan tersebut. [10].

2.7. Adam Optimizer

Adam Optimizer merupakan pengembangan dari metode Stochastic Gradient Descent (SGD) yang lebih populer digunakan dalam deep learning untuk mengklasifikasikan gambar. Adam Optimizer efisien dalam komputasi, memerlukan sedikit memori, dan mudah diimplementasikan. Dalam penelitian ini, Adam Optimizer digunakan untuk memodifikasi atribut neural network agar dapat meminimalkan loss [11].

2.8. Transfer Learning

Transfer learning adalah teknik dalam machine learning yang memperbaiki kinerja pada tugas tertentu dengan memanfaatkan pengetahuan dari model yang telah dilatih di domain yang berbeda sebelumnya. Pada Convolutional Neural Network (CNN), transfer learning dilakukan dengan memanfaatkan kembali lapisan fitur dari satu CNN untuk menginisialisasi jaringan lainnya. Ini berdampak positif pada peningkatan kinerja tugas target dan telah menjadi

penting dalam berbagai aplikasi. Transfer *learning* antar CNN dikenal sebagai fine-tuning, di mana lapisan fitur dari CNN sumber disesuaikan dengan data target untuk menyempurnakan adaptasinya. Fine-tuning dapat melibatkan lapisan fitur secara penuh atau sebagian, disesuaikan dengan tugas target. Lalu, lapisan tersebut diperbarui dengan dataset baru pada tugas klasifikasi yang berbeda. Transfer *learning* memanfaatkan pengetahuan sebelumnya untuk meningkatkan performa pada tugas target pada CNN yang baru. [3].

3. METODE PENELITIAN

Pada bagian ini memuat metode yang digunakan pada pembuatan skripsi.

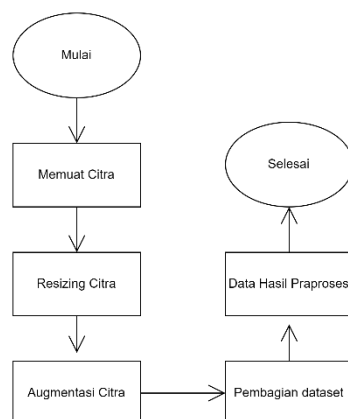
3.1. Pengumpulan Data

Tahap ini melibatkan berbagai metode pengumpulan data yang beragam, termasuk data sekunder yang diperoleh tidak langsung dari sumber asli melalui jejak pendapat narasumber, wawancara individu/kelompok, dan observasi hasil pengujian. Penelitian ini menggunakan sumber data dari website Kaggle. Data tersebut berfokus pada berbagai penyakit daun kentang, terdiri dari 1.000 sampel daun sehat, 1.000 sampel daun terkena *Early blight*, dan 1.000 sampel daun terkena *Late blight*, dengan total 3.000 sampel. Sampel data ini digunakan dalam penelitian sebagai contoh:



Gambar 1. Gambar sampel dataset daun kentang

3.2. Praproses Data



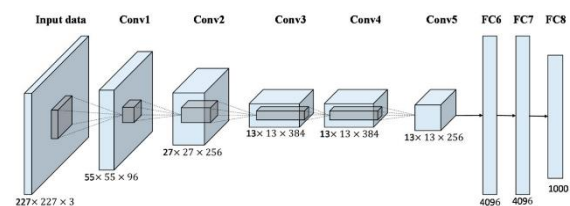
Gambar 2. Gambar diagram praproses data

Praproses data citra penyakit daun kentang pada Dataset terdiri dari tiga langkah, seperti dijelaskan pada Gambar 3.3. Pertama, ukuran citra diubah menjadi 128x128 piksel. Selanjutnya, dilakukan

transformasi citra untuk menyamakan tipe warna RGB. Setelah itu, dataset dibagi menjadi tiga bagian: data latih (2.167 citra), data validasi (383 citra), dan data uji (450 citra). Data uji digunakan untuk menguji performa model CNN yang telah dilatih.

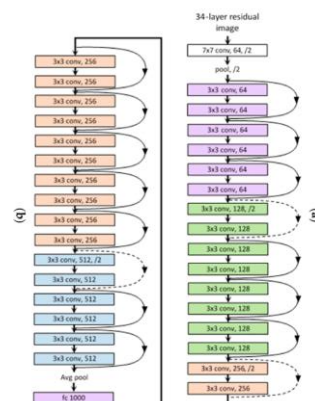
3.3. Perancangan Arsitektur

Pembangunan arsitektur *Convolutional Neural Network* (CNN) dimulai dengan pemilihan arsitektur dasar yang sesuai dengan karakteristik dataset dan masalah yang ingin diselesaikan. Selanjutnya, dilakukan tuning hyperparameter seperti *learning rate* dan momentum untuk meningkatkan performa model.



Gambar 3. Gambar arsitektur Alexnet

Alexnet, dikembangkan oleh Alex Krizhevsky, Ilya Sutskever, dan Geoffrey Hinton pada 2012, adalah model *Convolutional Neural Network* (CNN) yang meraih kemenangan dalam ImageNet Large Scale Visual Recognition Challenge. Dengan 8 lapisan, termasuk 5 lapisan konvolusi dan 3 lapisan fully connected, arsitektur ini memanfaatkan 96 filter 11x11 pada lapisan pertama, diikuti oleh filter 256, 384, dan 384 pada lapisan berikutnya, serta 256 filter 3x3 pada lapisan terakhir. ReLU dan pooling maksimum diterapkan setelah setiap lapisan konvolusi. Dropout digunakan pada lapisan fully connected untuk mencegah overfitting, dan optimasi hyperparameter dilakukan untuk meningkatkan kinerja model. *Alexnet* telah menginspirasi model CNN lebih canggih di masa depan.

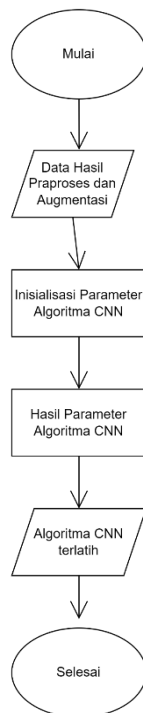


Gambar 4. Gambar arsitektur Resnet34

Resnet-34, dikembangkan oleh Kaiming He, Xiangyu Zhang, Shaoqing Ren, dan Jian Sun pada 2015, adalah model *Convolutional Neural Network*

(CNN) yang memenangkan kompetisi ImageNet Large Scale Visual Recognition Challenge. Dengan 34 lapisan neural, termasuk 32 lapisan konvolusi dan 2 lapisan fully connected, *Resnet-34* memiliki lapisan konvolusi pertama dengan 64 filter 7x7 dan stride 2, diikuti oleh blok residual. Setiap blok residual berisi lapisan konvolusi dengan ReLU dan dua konvolusi 3x3 dengan skip connection untuk mengurangi vanishing gradient. Regularisasi dropout digunakan pada lapisan fully connected untuk menghindari overfitting, dan optimasi hyperparameter dilakukan. *Resnet-34* adalah model CNN sukses yang menjadi dasar untuk pengembangan model CNN lebih kompleks dan akurat.

3.4. Pelatihan Model

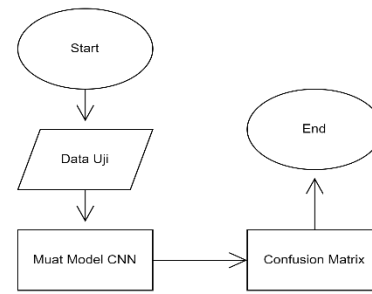


Gambar 5. Gambar diagram pelatihan model

Setelah merancang arsitektur model dan mengkonfigurasi hyperparameter pada model *Alexnet* dan *Resnet34*, langkah berikutnya adalah melakukan pelatihan pada dataset. Pelatihan dilakukan menggunakan Google Colaboratory dengan memanfaatkan akselerator GPU yang disediakan oleh Google Colaboratory untuk mempercepat waktu proses pelatihan dataset.

3.5. Pengujian Model

Model diuji dengan menggunakan 450 citra dari data uji yang telah melalui proses prapemrosesan. Setelah proses pengujian, nilai prediksi akan dibandingkan dengan nilai label asli pada data uji untuk menghitung jumlah prediksi yang benar. Selanjutnya, Confusion matrix dapat digunakan untuk mengevaluasi kinerja model. Berikut ini adalah diagram blok yang menunjukkan fungsi Adam.



Gambar 6. Gambar Pengujian Model

3.6. Evaluasi Model

Setelah selesai melakukan pelatihan model, langkah berikutnya adalah mengevaluasi performansi dari model tersebut. Evaluasi ini bertujuan untuk mengetahui sejauh mana model yang telah dibuat dapat berfungsi dengan baik. Dalam penelitian ini, evaluasi dilakukan dengan menggunakan model yang dievaluasi pada data uji dengan menggunakan metode confusion matrix untuk mendapatkan nilai akurasi, presisi, recall, dan f1 score. Hasil dari confusion matrix akan menjadi dasar untuk melakukan pelatihan model kembali jika hasil evaluasi tidak memuaskan. Berikut rumus precision, recall dan f1- score:

$$\text{Precision} = \frac{T_p}{T_p + F_p} \quad (1)$$

$$\text{Recall} = \frac{T_p}{T_p + T_n} \quad (2)$$

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

4. HASIL DAN PEMBAHASAN

Adapun hasil dan pembahasan yang akan dijelaskan menjadi beberapa sub bab.

4.1. Penyiapan Dataset

Data yang akan digunakan dalam penelitian ini berasal dari Kaggle. Data tersebut merupakan data publik yang terdiri dari tiga jenis, yaitu daun yang sehat, *Early blight*, dan *Late blight*. Setiap jenis data terdiri dari 1000 gambar, sehingga totalnya terdapat 3000 gambar yang akan digunakan dalam penelitian ini. Data tersebut akan digunakan untuk pengujian dalam penelitian ini.

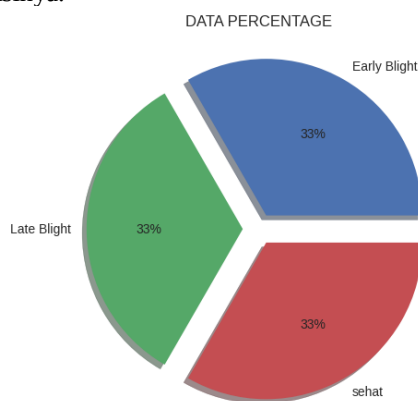
4.2. Menyiapkan Data Dan Label

Untuk melakukan proses klasifikasi dalam penelitian ini, diperlukan data berupa citra digital yang menampilkan daun kentang sebagai objeknya. Objek penelitian dibatasi menjadi tiga kelas, yaitu daun yang sehat, *Early blight*, dan *Late blight*. Jumlah data yang terkumpul sebanyak 3000 data, seperti yang ditampilkan pada Gambar 7.

	image	label
0	Healthy_83 (3).jpg	sehat
1	Healthy_81 (3).jpg	sehat
2	Healthy_75 (3).jpg	sehat
3	Healthy_85 (3).jpg	sehat
4	Healthy_89 (3).jpg	sehat
...	...	...
2995	Early_Blight_121.jpg	Early Blight
2996	Early_Blight_39.jpg	Early Blight
2997	Early_Blight_53.jpg	Early Blight
2998	Early_Blight_8.jpg	Early Blight
2999	Early_Blight_31.jpg	Early Blight
3000 rows x 2 columns		

Gambar 7. Tampilan keseluruhan data

Data penelitian ini terdiri dari 3000 data yang terbagi dalam tiga folder, yang masing-masing folder sesuai dengan objek daun kentang yang akan diteliti. Klasifikasi objek daun kentang ini mencakup tiga kondisi, yaitu daun yang sehat, daun yang terkena *Early blight*, dan daun yang terkena *Late blight*. Setiap folder berisi data yang representatif untuk masing-masing kondisi, sehingga memungkinkan analisis yang lebih rinci dan akurat dalam penelitian ini. Melalui penggunaan dataset yang kaya dan beragam ini, diharapkan penelitian dapat memberikan pemahaman yang lebih mendalam mengenai pengklasifikasian daun kentang berdasarkan kondisinya.



Gambar 8. Grafik presentase dataset

Dataset terdiri dari tiga kelas yang masing-masing memiliki pembagian data sebesar 33% dari total dataset. Pembagian tersebut mencakup 33% data yang merupakan daun kentang yang sehat, 33% data yang merupakan daun kentang yang terkena penyakit *Early blight*, 33% data yang merupakan daun kentang yang terkena penyakit *Late blight*.

4.3. Pembagian Data

Setelah mendapatkan informasi mengenai jumlah dataset dan kelas yang akan digunakan, langkah selanjutnya adalah membagi data menjadi tiga bagian, yaitu data train, data validasi train, dan data test.

Dalam penelitian ini terdapat 2.167 data latih. Selanjutnya, data validasi akan digunakan untuk melakukan validasi model guna menentukan bobot agar terhindar dari overfitting atau underfitting. Jumlah data validasi dalam penelitian ini adalah 383 data. Terakhir, data test akan digunakan untuk melakukan prediksi menggunakan model yang telah dilatih. Pada penelitian ini, terdapat 450 data test.

4.4. Praproses

Langkah ini dijalankan setelah dataset dibagi untuk digunakan. Tahapan ini hanya diterapkan pada sejumlah citra sebelum masuk ke dalam model yang akan digunakan. Citra-citra awal yang terlibat mencakup citra daun yang sehat, citra daun yang terkena *Early blight*, dan citra daun yang terkena *Late blight*.

Dua set data dibentuk untuk keperluan pelatihan dan validasi. Transformasi yang diterapkan pada kedua set ini berbeda. Data pelatihan dan validasi ini selanjutnya dimuat dalam DataLoader. Pada DataLoader, data pelatihan diacak dalam proses pengambilan batch untuk menjaga variasi, sementara data validasi tidak diacak. Kode ini memiliki tujuan untuk mengorganisir dan mempersiapkan data sesuai dengan standar yang diharapkan, memungkinkan penggunaan yang efisien dalam pelatihan dan evaluasi model.

4.5. Implementasi Alexnet

Pada riset ini, terdapat dua variasi arsitektur *Convolutional Neural Network* (CNN) yang akan dimanfaatkan. Arsitektur tersebut mencakup *Resnet34* dan *Alexnet*. Proses pembelajaran model dilakukan melalui tahapan pelatihan dan pengujian. Dalam konteks penggunaan data augmented sebanyak 3000, proporsi penggunaan data tersebut adalah 85% untuk tahap pelatihan dan 15% untuk tahap pengujian model. Setelah penerapan kedua model ini, dilakukan perbandingan terhadap hasil terbaik dari masing-masing model, untuk mengetahui arsitektur mana yang memiliki akurasi paling tinggi. Klasifikasi yang dihasilkan dari penelitian ini terdiri dari tiga kelas, yakni kondisi daun kentang yang sehat, daun kentang yang terkena *Early blight* dan daun kentang yang terkena *Late blight*.

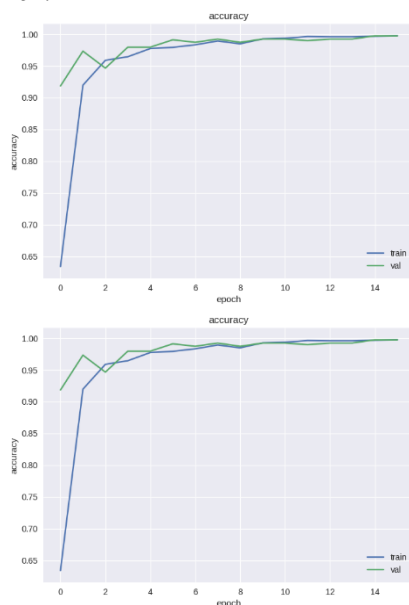
4.6. Implementasi Resnet34

Dijelaskan mengenai proses kerja dari arsitektur *Resnet34*. Pendekatan ini melibatkan pembangunan jaringan yang lebih mendalam daripada model konvensional lainnya, sambil secara simultan menangani masalah gradien yang menghilang. *Resnet34* terdiri dari total 34 lapisan dan menggunakan struktur Basic Block. Pada setiap tahap konvolusi,

setiap lapisan akan mempelajari filter untuk mengidentifikasi fitur-fitur yang relevan. Hasil dari proses pelatihan ini akan melewati tahap average pooling (dengan AdaptiveAvgPool2d) sebelum masuk ke jaringan fully connected. Pada akhirnya, keluaran dari jaringan akan melalui fungsi aktivasi softmax untuk menghasilkan hasil klasifikasi yang akhir.

4.7. Pelatihan Model

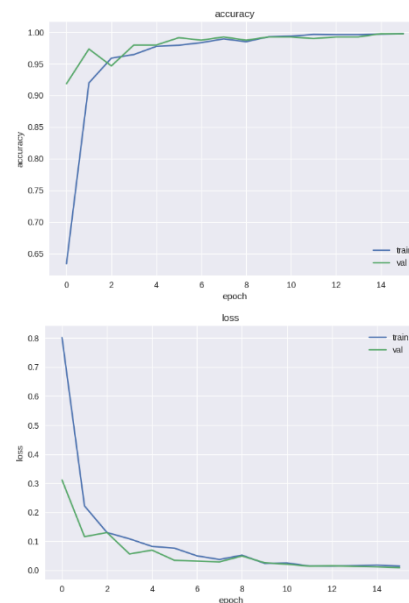
Pada tahap implementasi, akurasi dihitung dengan membandingkan prediksi yang benar dengan total sampel dalam batch menggunakan fungsi softmax untuk mengambil indeks kelas tertinggi. Hyperparameter seperti jumlah *Epoch* (16), laju pembelajaran (*lr*), dan kriteria loss (criterion) ditentukan dengan CrossEntropyLoss. Optimizer Adam dengan weight decay $1e-4$ digunakan. Scheduler OneCycleLR mengatur laju pembelajaran berdasarkan langkah per *Epoch*. Setiap *Epoch*, loss, akurasi pelatihan dan validasi, serta waktu eksekusi dicatat dan dicetak. Grafik akurasi dan loss direkam, menunjukkan variasi waktu eksekusi per *Epoch* untuk *Alexnet* (82 menit) dan *Resnet34* (199 menit) dalam 16 *Epoch* dan batch size 14. *Alexnet* membutuhkan waktu sedikit lebih lama dibanding *Resnet34* dalam pelatihan.



Gambar 9. Grafik accuracy dan loss pada model *Alexnet*

Pada Gambar 9, hasil dari skenario penggunaan model *Alexnet* dengan laju pembelajaran 0.0001 terlihat. Grafik pelatihan menunjukkan peningkatan drastis pada akurasi, dengan sedikit perbedaan antara akurasi pelatihan dan validasi. Hal ini menandakan model tidak mengalami overfitting.

Hasil akhir pelatihan menunjukkan akurasi pelatihan 0.988 dan validasi 0.980. Penggunaan laju pembelajaran 0.0001 pada *Alexnet* memberikan hasil yang baik, dengan kemampuan mengenali pola data pelatihan dan mampu generalisasi pada data baru.



Gambar 10. Grafik accuracy dan loss pada model *Resnet34*

Pada Gambar 4.4, hasil eksperimen menggunakan model *Resnet34* dengan *learning rate* 0.0001 ditampilkan. Grafik pelatihan menunjukkan peningkatan akurasi yang signifikan dan minimalnya perbedaan antara akurasi pelatihan dan validasi, menunjukkan penghindaran overfitting.

Hasil akhir pelatihan menunjukkan akurasi pelatihan 0.998 dan validasi 0.997. Penggunaan *learning rate* 0.0001 pada *Resnet34* memberikan hasil sangat memuaskan, dengan akurasi tinggi pada data pelatihan dan kemampuan yang kuat dalam menggeneralisasi pada data validasi.

4.8. Pengujian Model

Tahap pengujian dimulai dengan mempersiapkan dataset pengujian menggunakan objek 'test_set', yang mengambil citra dan label dari kumpulan data pengujian dengan menerapkan transformasi yang telah ditentukan sebelumnya.

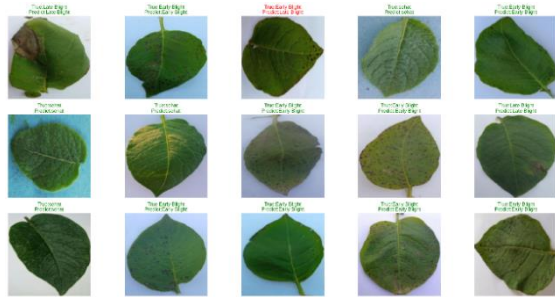
Fungsi predict kemudian digunakan untuk memprediksi label kelas dari citra yang diberikan. Prediksi akhir diperoleh dengan mengekstrak indeks dari kelas dengan probabilitas terbesar menggunakan 'torch.argmax' pada hasil fungsi softmax. Hasil prediksi ini kemudian dikembalikan dalam bentuk angka integer.

Pada tahap pengujian, iterasi dilakukan untuk setiap citra dalam dataset pengujian. Hasil prediksi dari setiap citra disimpan dalam 'test_predict', sementara label asli disimpan dalam 'true_label'. Dengan ini, kita dapat membandingkan prediksi model dengan label asli untuk mengukur akurasi dan melakukan analisis lebih lanjut terhadap performa model. Keseluruhan proses ini berguna untuk mengevaluasi sejauh mana model mampu melakukan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya, yakni data pengujian.

4.9. Analisa Hasil Pengujian

Analisis hasil pengujian meliputi tiga komponen utama: hasil klasifikasi, Confusion matrix, dan evaluasi metrik performa. Dalam penelitian ini, model arsitektur *Convolutional Neural Network* (CNN) berhasil mengklasifikasikan objek dengan akurat. Hasil klasifikasi menghasilkan citra 128 x 128 piksel yang telah diberi label, dengan hasil yang tepat ditandai oleh label berwarna hijau dan hasil yang salah diklasifikasikan ditandai oleh label merah.

Klasifikasi benar terjadi saat prediksi hasil cocok dengan nilai sebenarnya pada citra pengujian (true). Sebaliknya, klasifikasi salah terjadi saat prediksi tidak sesuai dengan nilai sebenarnya. Label merah menunjukkan hasil klasifikasi yang salah, sedangkan label hijau menunjukkan hasil klasifikasi yang benar. Detil hasil klasifikasi dapat dilihat pada Gambar 11. dan 12.



Gambar 11. Gambar hasil klasifikasi pada *Alexnet*



Gambar 12. Gambar hasil klasifikasi pada *Resnet34*

Evaluasi performa klasifikasi dari setiap model arsitektur disajikan melalui Confusion matrix, yang terlihat pada Gambar 13 dan Gambar 14. Dalam rangka mengevaluasi hasil klasifikasi ini, digunakan empat metrik performa utama yaitu Akurasi (Accuracy), Presisi (Precision), Recall, dan Ukuran F (F-measure).

	Early Blight	Late Blight	sehat
Early Blight	145	2	3
Late Blight	1	148	1
sehat	2	1	147

Gambar 13. Gambar confusion matrix *Alexnet*

Tabel Confusion matrix yang terdapat pada model arsitektur *Alexnet*, sebagaimana terlihat pada Gambar 4.7, mampu berhasil mengklasifikasikan 440 gambar dari total 450 gambar yang diuji, sementara 10 gambar mengalami kegagalan dalam proses klasifikasinya. Proses ini memberikan gambaran visual tentang sejauh mana model mampu mengidentifikasi kelas-kelas objek dengan benar dan bagaimana distribusi hasil prediksinya terhadap data pengujian secara keseluruhan.

	Early Blight	Late Blight	sehat
Early Blight	149	0	1
Late Blight	1	148	1
sehat	0	0	150

Gambar 14. Gambar confusion matrix *Resnet34*

Tabel Confusion matrix yang terkait dengan model arsitektur *Resnet34*, sesuai dengan yang diilustrasikan pada Gambar 4.12, berhasil mengklasifikasikan 447 gambar dari total 450 gambar yang telah diujikan. Meskipun demikian, terdapat ketidakmampuan dalam mengklasifikasikan dengan benar pada 3 gambar yang termasuk dalam dataset pengujian. Hasil dari tabel Confusion matrix ini memberikan pandangan yang jelas tentang kemampuan model dalam mengidentifikasi kelas objek secara akurat, serta memberikan gambaran tentang sejauh mana distribusi hasil prediksi model pada seluruh data pengujian.

Tabel 1. Tabel hasil performa pelatihan

Parameter	Alexnet	Resnet34
Waktu/Iterasi	1.55 s	2.34 s
Train Accuracy	0.988	0.998
Train Loss	0.035	0.014
Validation Accuracy	0.980	0.997
Validation Loss	0.054	0.009

Dalam iterasi *Alexnet*, setiap iterasi memerlukan waktu sekitar 1.55 detik. Parameter-parameter tertentu menunjukkan bahwa *Resnet34* memiliki train loss terendah, 0.014 (1.4%), menunjukkan kerugian pelatihan yang rendah. Akurasi pelatihan tertinggi juga dimiliki oleh *Resnet34*, mencapai 0.998 (99.8%), menandakan kemampuan klasifikasi yang sangat akurat pada data pelatihan.

Evaluasi parameter validasi menunjukkan *Resnet34* memiliki performa yang baik. Validation loss terendah adalah 0.009 (0.09%), menunjukkan kerugian validasi minimal. Akurasi validasi tertinggi juga dimiliki oleh *Resnet34*, mencapai 0.997 (99.7%), menunjukkan kemampuan klasifikasi yang akurat pada data validasi. Hal ini menegaskan kinerja konsisten dan efektif *Resnet34* dalam pelatihan dan evaluasi pada data pelatihan dan validasi.

Tabel 2. Tabel hasil confusion matrix

Arsitektur	Akurasi (%)	Precision (%)	Recall (%)	Waktu
Alexnet	0.98	0.98	0.98	4932.276 s
Resnet34	0.99	0.99	0.99	11952.114 s

Dari Tabel 2, *Resnet34* mencapai akurasi tertinggi 99%, 1% lebih tinggi dari *Alexnet*. Ini menandakan *Resnet34* memiliki kemampuan mengklasifikasikan data dengan akurasi tinggi dan memahami pola-pola data kompleks.

Secara waktu, *Alexnet* menjadi yang tercepat dengan pelatihan 4932.276 detik. Ini menunjukkan meskipun efisien dalam waktu, *Alexnet* tetap memberikan akurasi tinggi pada pengujian. Perbandingan antara akurasi dan waktu ini mengilustrasikan trade-off antara prediksi berkualitas dan efisiensi waktu. Ini membantu memilih model yang sesuai dengan aplikasi dan sumber daya yang ada.

Tabel 3. Tabel hasil fi score

Kelas	Alexnet (%)	Resnet34 (%)
0 (Early blight)	0.97	0.99
1 (Late blight)	0.98	0.99
2 (Sehat)	0.98	0.99

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut : Hasil pengujian terhadap dataset yang telah melalui proses pelatihan dan evaluasi menggunakan dua model arsitektur berbeda menunjukkan pencapaian akurasi yang bagus. Pertama, model arsitektur *Alexnet* berhasil mencapai tingkat akurasi sebesar 98% setelah dilatih

selama 16 *Epoch* dengan batch size 14. Penentuan parameter seperti *learning rate* sebesar 0,0001 dan weight decay sebesar 0,0001 juga turut berkontribusi pada kesuksesan model ini dalam mencapai akurasi yang tinggi. Sementara itu, hasil pengujian pada model arsitektur *Resnet34* juga mengesankan dengan mencapai akurasi sebesar 99%. Model ini mengalami pelatihan selama 16 *Epoch* dengan menggunakan batch size 14, dan parameter *learning rate* serta weight decay yang sama juga diterapkan untuk mencapai tingkat akurasi yang signifikan. Kesimpulan dari kedua pengujian ini menegaskan bahwa kedua model arsitektur tersebut mampu memberikan performa yang sangat baik dalam mengklasifikasikan citra penyakit daun kentang menggunakan teknik transfer *learning*. Dari kedua model arsitektur dijelaskan bahwa akurasi tertinggi pada penelitian ini adalah model *Resnet34* dengan nilai akurasi 99%, akan tetapi waktu pelatihan model dengan waktu tercepat dimiliki *Alexnet* dengan total waktu 4932.276 s atau 82 menit.

Berdasarkan hasil implementasi arsitektur *Alexnet* dan *Resnet34* pada klasifikasi citra penyakit daun kentang, terdapat beberapa saran yang perlu diperhatikan untuk mengembangkan pada penelitian berikutnya, diantaranya : Disarankan untuk memastikan bahwa dataset yang digunakan memiliki jumlah yang cukup besar dan distribusi data yang seimbang di setiap kelas. Hal ini penting agar tidak terjadi ketidakseimbangan antar kelas selama proses pelatihan model, sehingga model dapat mengenali setiap kelas dengan baik. Selain itu, untuk mendapatkan dataset yang lebih representatif, disarankan untuk mengambil gambar daun kentang secara langsung, sehingga dataset yang digunakan lebih mewakili kondisi nyata dan variasi yang ada di lapangan. Perlu diingat bahwa pemilihan arsitektur model harus disesuaikan secara hati-hati, mengingat tidak semua model akan menghasilkan performa terbaik. Selain itu, disarankan untuk menjelajahi opsi lain dengan mempertimbangkan variasi arsitektur model seperti VggNet, LeNet, ZFNet, GoogLeNet, dan lainnya yang dapat memberikan beragam pilihan untuk meningkatkan performa model dan hasil secara keseluruhan. Dalam penelitian mendatang, disarankan untuk melakukan kustomisasi pada hyperparameter seperti *learning rate*, jumlah *Epoch*, dan batch size. Hal ini bertujuan untuk lebih lanjut meningkatkan kinerja dan performa tiap model yang dievaluasi.

DAFTAR PUSTAKA

- [1] Campos, H., & Ortiz, O. (2019). The potato crop: Its agricultural, nutritional and social contribution to humankind. In *The Potato Crop: Its Agricultural, Nutritional and Social Contribution to Humankind*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-28683-5>
- [2] Rozaqi, A. J., Sunyoto, A., & Arief, R. (2021). Deteksi Penyakit pada Daun Kentang Menggunakan Pengolahan Citra dengan Metode

- Convolutional Neural Network* Detection of Potato Leaves Disease Using Image Processing with *Convolutional Neural Network* Methods.
- [3] Afridi, M. J., Ross, A., & Shapiro, E. M. (2018). On automated source selection for transfer learning in *Convolutional Neural Networks*. *Pattern Recognition*, 73, 65–75. <https://doi.org/10.1016/j.patcog.2017.07.019>
- [4] Setiawan, W. (2019). PERBANDINGAN ARSITEKTUR CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI FUNDUS. 7(2).
- [5] Ertel, W. (n.d.-a). Undergraduate Topics in Computer Science Introduction to Artificial Intelligence. <http://www.springer.com/series/7592>
- [6] Zaid Munantri, N., Sofyan, H., & Yanu, M. (2019). APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK IDENTIFIKASI UMUR POHON. In *TELEMATIKA* (Vol. 16, Issue 2)
- [7] LeCun, Y. (2018). The Power and Limits of *Deep Learning*: In his IRI Medal address, Yann LeCun maps the development of machine learning techniques and suggests what the future may
- [12] 109/JCSSE.2019.8864165.
- hold. *Research Technology Management*, 61(6), 22–27. <https://doi.org/10.1080/08956308.2018.1516928>
- [8] Kelleher, J. D. (2019). *Deep learning*. MIT Press.
- [9] Lu, S., Wang, S. H., & Zhang, Y. D. (2021). Detection of abnormal brain in MRI via improved *Alexnet* and ELM optimized by chaotic bat algorithm. In *Neural Computing and Applications* (Vol. 33, Issue 17, pp. 10799–10811). Springer Science and Business Media Deutschland GmbH. <https://doi.org/10.1007/s00521-020-05082-4>
- [10] Al-Moosawi, N. M., & Khudeyer, R. S. (2021). *Resnet-34/DR: A Residual Convolutional Neural Network* for the Diagnosis of Diabetic Retinopathy. *Informatica (Slovenia)*, 45(7), 115–124. <https://doi.org/10.31449/inf.v45i7.3774>
- [11] M. F. Naufal & S. F. Kusuma. (2019) "Software defect detection based on selected complexity metrics using fuzzy association rule mining and defective module oversampling," 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), Chonburi, Thailand, pp. 330-335, doi: 10.1