

Klasifikasi Citra X-Ray Tuberkulosis Menggunakan *Convolutional Neural Networks*

Haykal Alya Mubarak*, Rice Novita

Fakultas Sains dan Teknologi, Prodi Sistem Informasi, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia

Email: ^{1,*}12150313837@students.uin-suska.ac.id, ²rice.novita@uin-suska.ac.id,

Email Penulis Korespondensi: 12150313837@students.uin-suska.ac.id

Submitted: 25/12/2024; Accepted: 26/02/2025; Published: 01/03/2025

Abstrak—Tuberkulosis (TBC) merupakan penyakit menular serius yang masih menjadi salah satu penyebab utama kematian di dunia, terutama di negara berkembang. Analisis citra X-ray merupakan langkah penting dalam pengendalian penyakit ini. Penelitian ini bertujuan untuk mengklasifikasikan citra X-ray penyakit tuberkulosis menggunakan pendekatan deep learning dengan tiga arsitektur *Convolutional Neural Networks* (CNN): DenseNet201, Xception, dan MobileNetV2. Dataset yang digunakan terdiri dari 3.000 gambar X-ray, terbagi dalam dua kategori: normal dan TBC, yang diperoleh dari Kaggle, yang kemudian diproses melalui normalisasi, augmentasi, dan pembagian data menggunakan metode *hold-out* dengan rasio 70:30, 80:20, dan 90:10. Hasil penelitian menunjukkan bahwa DenseNet201 dengan optimizer Nadam pada pembagian data 90:10 menghasilkan akurasi tertinggi sebesar 100%, menjadikannya kombinasi terbaik untuk klasifikasi citra X-ray TBC. Xception mencapai akurasi terbaik sebesar 96,66% dengan optimizer Nadam pada pembagian data 80:20. MobileNetV2 menunjukkan akurasi optimal sebesar 98,69% menggunakan optimizer Adam pada pembagian data 90:10. Penelitian ini membuktikan bahwa DenseNet201 dengan optimizer Nadam sangat efektif dalam menangani klasifikasi citra medis, khususnya untuk tuberkulosis. Hasil ini memberikan kontribusi penting dalam pengembangan teknologi berbasis deep learning untuk meningkatkan akurasi diagnosis penyakit tuberkulosis.

Kata Kunci: Tuberkulosis; Klasifikasi; Convolutional Neural Network; Optimisasi Adam; Optimisasi Nadam

Abstract—Tuberculosis (TB) is a serious infectious disease that is still one of the main causes of death in the world, especially in developing countries. X-ray image analysis is an important step in controlling this disease. This research aims to classify X-ray images of tuberculosis using a deep learning approach with three Convolutional Neural Networks (CNN) architectures: DenseNet201, Xception, and MobileNetV2. The dataset used consists of 3,000 X-ray images, divided into two categories: normal and TBC, obtained from Kaggle, which are then processed through normalization, augmentation, and data division using the hold-out method with a ratio of 70:30, 80:20, and 90:10. The research results show that DenseNet201 with the Nadam optimizer at 90:10 data division produces the highest accuracy of 100%, making it the best combination for TBC X-ray image classification. Xception achieved the best accuracy of 96.66% with the Nadam optimizer at a data split of 80:20. MobileNetV2 shows an optimal accuracy of 98.69% using the Adam optimizer at a 90:10 data split. This research proves that DenseNet201 with the Nadam optimizer is very effective in handling medical image classification, especially for tuberculosis. These results provide an important contribution to the development of deep learning-based technology to improve the accuracy of tuberculosis diagnosis.

Keywords: Tuberculosis; Classification; Convolutional Neural Network; Adam Optimizer; Nadam Optimizer

1. PENDAHULUAN

Infeksi yang disebabkan oleh bakteri *Mycobacterium tuberculosis*, tuberkulosis (TBC) merupakan salah satu penyakit tertua yang diketahui menyerang manusia dan merupakan penyebab utama kematian di seluruh dunia. Di seluruh dunia, terutama di negara berkembang, penyakit ini menjadi salah satu penyebab kematian paling umum. Laporan global tentang tuberkulosis pada tahun 2023 menunjukkan bahwa tuberkulosis adalah penyakit menular terbesar kedua di dunia setelah penyakit coronavirus 2019 (COVID-19) dan penyebab kematian ketiga belas di seluruh dunia [1]. Menurut data Organisasi Kesehatan Dunia (WHO), sekitar 750.000 kasus global setiap tahun, tuberkulosis yang resistan terhadap obat memiliki dampak yang menghancurkan kesehatan Masyarakat [2]. Sangat penting untuk mendapatkan diagnosis dan pengobatan yang tepat untuk mengurangi penyebaran dan dampak fatal dari penyakit ini. Radiografi X-ray adalah teknik sederhana dan umum untuk mendeteksi berbagai penyakit. Dengan itu, beberapa rumah sakit di seluruh dunia menyimpan sejumlah gambar dan laporan radiografi untuk arsip setiap harinya yang dijadikan sebagai sumber informasi berharga dalam memproses analisis gambar [3].

Dalam beberapa tahun terakhir, perkembangan teknologi kecerdasan buatan telah membuka peluang baru dalam dunia medis, khususnya dalam pengolahan dan analisis citra medis [4]. Diantaranya deep learning, ditunjukkan sebagai cara potensial untuk mengklasifikasikan tuberkulosis secara otomatis dengan menggunakan gambar x-ray [5]. Akan tetapi, masih jarang yang menerapkan pendekatan deep learning sebagai cara menganalisis tuberkulosis [6]. Teknik ini memproses data seperti neuron di otak manusia untuk mengklasifikasikan gambar, mendeteksi objek, menghasilkan gambar, dan menerjemahkan [7]. Didalam *deep learning* terdapat algoritma yang menawarkan solusi ampuh, yaitu *Convolutional Neural Networks* (CNN). dianggap memiliki kemampuan luar biasa dalam memproses data gambar, berkat kemampuannya untuk mengekstraksi fitur abstrak dari kumpulan data yang kompleks. Hal ini membuat CNN sangat cocok untuk pelatihan pada kumpulan data gambar berskala besar [8]. Algoritma ini terdiri dari beberapa lapisan konvolusi dan penyatuan yang saling berhubungan, yang memungkinkan menangkap fitur terperinci dalam gambar, dari tepi sederhana hingga bentuk rumit, sehingga sangat berharga dalam bidang seperti analisis gambar medis, di mana ekstraksi fitur mendalam sangat penting [9]. Oleh karena itu cnn dapat digunakan untuk menganalisis citra x-ray dada penyakit tuberkulosis dengan akurasi tinggi.

Dalam menerapkan model *deep learning*, untuk meningkatkan kinerja model tersebut dapat menggunakan arsitektur jaringan syaraf tiruan diantaranya DenseNet201, Xception dan MobileNetV2 yang telah banyak digunakan serta menghasilkan yang sangat baik dalam kasus klasifikasi data citra. DenseNet201 menggunakan konektivitas padat, memungkinkan setiap lapisan menerima input dari lapisan sebelumnya, sehingga meningkatkan efektivitas pelatihan dan mengatasi masalah *gradient vanishing*. Fitur yang dipelajari dapat digunakan kembali, mengurangi redundansi dan jumlah parameter, serta mengurangi risiko *overfitting*, membuatnya ideal untuk klasifikasi citra [10]. Xception memiliki *convolution layer* yang menggabungkan *pointwise convolution* yang kemudian diikuti oleh *depthwise convolution*, Kombinasi tersebut kemudian terbukti memberikan akurasi yang lebih baik dalam melakukan klasifikasi citra suatu [11]. MobileNetV2 menawarkan efisiensi tinggi untuk perangkat dengan daya dan memori terbatas dengan blok inversi residu dan convolution yang terpisah secara *depthwise*[12].

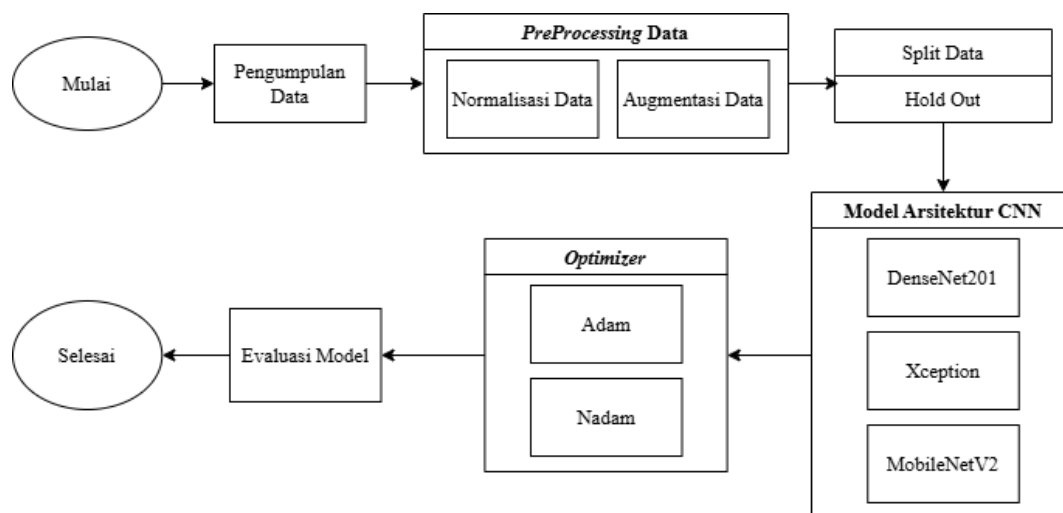
Adapun penelitian terdahulu Penelitian Rahman dkk menggunakan DenseNet201 untuk segmentasi citra X-ray, mencapai akurasi 99,8% pada dataset kecil [13]. Sharma dkk menguji model Xception pada 2000 citra X-ray tuberculosis, menghasilkan akurasi, presisi, recall, dan F1-score sebesar 99,29% [14]. Kawsher Mahbub dkk menggunakan MobileNetV2 untuk klasifikasi kelainan paru akibat Covid-19, Pneumonia, dan Tuberkulosis, dengan akurasi masing-masing 99,87%, 99,55%, dan 99,76% [15]. Penelitian Eram Mahamud dkk bertujuan meningkatkan interpretabilitas model deep learning untuk mendiagnosis COVID-19, pneumonia, dan tuberkulosis dari citra X-ray dengan menggunakan model DenseNet201 dalam kerangka transfer learning, dengan hasil model dengan augmentasi data mencapai akurasi 99,20%, precision, dan recall 99% [16]. Jiayi Lin dkk melakukan penelitian mengembangkan model few-shot learning (FSL) berbasis Xception untuk membedakan tuberkulosis usus dan penyakit Crohn menggunakan citra endoskopi dengan sampel terbatas. Model ini, dengan dual transfer learning, mencapai performa tinggi (AUC 0,81) dan melampaui kemampuan endoskopis [17]. Rarasmaya Indraswari dkk menggunakan MobileNetV2 untuk klasifikasi melanoma. Model ringan ini menunjukkan akurasi hingga 85% dan cocok untuk perangkat mobile berkat efisiensi jaringan dan arsitektur global average pooling dengan dua lapisan fully-connected [18].

Dari penelitian-penelitian tersebut menunjukkan bahwa arsitektur deep learning seperti DenseNet201, Xception, dan MobileNetV2 mampu mengklasifikasikan data gambar dengan tepat. Setiap arsitektur memiliki keunggulan khusus dalam menangani pengklasifikasian data gambar. Pada penelitian kali ini menggunakan dataset yang bersumber dari website Kaggle yang merupakan platform data public. Tujuan dari penelitian ini ialah untuk membandingkan beberapa arsitektur CNN yang dapat menghasilkan akurasi terbaik dalam melakukan klasifikasi citra terkait penyakit tuberculosis.

2. METODOLOGI PENELITIAN

2.1 Metodologi Penelitian

Gambar 1 menunjukkan proses penelitian yang terdiri dari pengumpulan data, preprocessing data, pembagian data, dan pemodelan deep learning. Selain itu, ada juga optimizer dan evaluasi model.



Gambar 1. Alur Metodologi Penelitian

Gambar 1 menjelaskan secara rinci semua tahapan penelitian, mulai dari pengumpulan data, *pre-processing* untuk memperbaiki format data, pembagian data menggunakan teknik validasi *hold-out*, pelatihan menggunakan model *deep learning*, evaluasi model menggunakan confusion matrix untuk mengevaluasi seberapa baik model yang dilatih dalam mengklasifikasikan data, dan analisis perbandingan untuk mendapatkan tingkat akurasi terbaik dari model pada proses klasifikasi data citra x-ray tuberculosis.

2.2 Pengumpulan Data

Data gambar X-ray tuberkulosis yang akan digunakan terdiri dari 3000 gambar X-ray yang dikumpulkan dari website data sumber Kaggle, dan diberi label Tuberculosis dan Normal, seperti yang ditunjukkan pada Gambar 2.



Normal

Tuberculosis

Gambar 2. Visualisasi Dataset Citra Tuberkulosis

2.3 Pre-Processing Data

Proses selanjutnya adalah preprocessing data citra X-ray dengan melakukan normalisasi gambar dan augmentasi. Normalisasi bertujuan untuk menyamakan skala gambar, sehingga data gambar menjadi seragam. Hal ini memungkinkan model untuk memproses data dengan lebih stabil dan mempercepat pelatihan model. Data mentah awalnya memiliki ukuran dan bentuk yang bervariasi, oleh karena itu normalisasi sangat penting untuk memastikan konsistensi data.

Sementara itu, augmentasi data bertujuan untuk meningkatkan keberagaman sumber data, yang pada gilirannya dapat meningkatkan akurasi proses diagnosis. Pada data pelatihan, augmentasi dilakukan dengan berbagai teknik, seperti rotasi gambar acak yang memutar gambar dalam berbagai sudut untuk memperkenalkan variasi orientasi dan membantu model mengenali pola meskipun citra terputar. Skala ulang nilai piksel dilakukan dengan membagi setiap nilai piksel dengan angka tertentu, yang membuat rentang nilai piksel seragam dan memudahkan proses pembelajaran model. Flipping gambar dilakukan secara vertikal dan horizontal untuk menciptakan variasi citra dari berbagai sudut pandang, memastikan model dapat mengenali objek dari berbagai orientasi. Selain itu, teknik pengisian piksel kosong diterapkan, dengan menggantikan piksel kosong akibat pergeseran atau pemotongan gambar menggunakan nilai piksel terdekat, menjaga agar citra tetap utuh dan informatif.

Pada data pengujian, tidak dilakukan augmentasi, hanya dilakukan skala ulang nilai piksel dan pembagian data untuk validasi. Proses berikutnya dalam preprocessing adalah menentukan ukuran batch, yang mengatur jumlah sampel gambar yang akan dihasilkan oleh generator setiap kali dijalankan, untuk memastikan efisiensi pelatihan model. Dengan menerapkan teknik augmentasi ini, model akan lebih mampu mengenali berbagai variasi gambar dan meningkatkan akurasi diagnosis pada data X-ray.

2.4 Pembagian Data

Untuk tahapan selanjutnya dengan melakukan pembagian dataset yaitu data latih, data validasi, serta data pengujian. Proses ini merupakan langkah penting dalam mempersiapkan dataset sebelum melakukan pengujian model. Pembagian data latih dan validasi dan pengujian menggunakan teknik *Hold-Out* dengan rasio 70:30, 80:20, dan 90:10. Data yang telah dibagi tersebut dipisahkan pada path yang berbeda untuk memudahkan proses. Tujuan dari pembagian data berikut agar dapat memastikan bahwa model yang dibangun dapat dievaluasi dengan baik dan memiliki kemampuan generalisasi yang baik terhadap data baru.

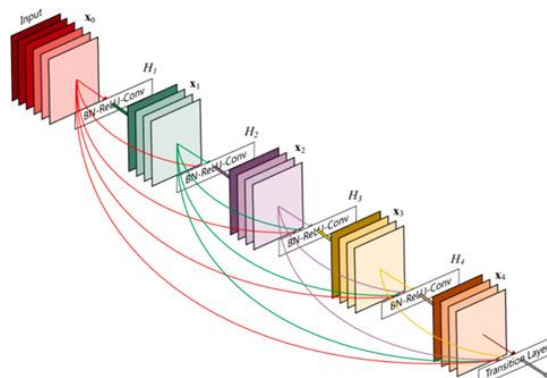
2.5 Pemodelan Arsitektur CNN

Pada tahapan ini dilakukan proses pengujian model dengan implementasi arsitektur CNN dalam klasifikasi citra x-ray tuberkulosis. Model ini terdiri dari beberapa layer utama yang bekerja secara bertahap untuk mengekstraksi dan mengabstraksi fitur-fitur penting dari citra. Beberapa arsitektur yang akan diterapkan yaitu DenseNet201, Xception, dan MobileNetV2 selanjutnya akan dapat membandingkan kinerja performa dari setiap model arsitektur yang digunakan. Secara keseluruhan, arsitektur CNN ini dirancang untuk secara efisien dan efektif mengekstraksi fitur dari citra dan melakukan klasifikasi dengan akurasi tinggi, menjadikannya salah satu pendekatan yang paling berhasil dalam penelitian klasifikasi citra.

a. DenseNet201

DenseNet201 adalah jaringan saraf konvolusional (CNN) pembelajaran mendalam yang terdiri dari 201 lapisan. Jaringan ini menonjol karena menggunakan koneksi padat, yang membentuk tautan langsung antara jalur maju dan mundur dalam jaringan saraf. Pilihan arsitektur yang cerdas ini memfasilitasi pembagian dan transmisi fitur yang efisien di berbagai lapisan. Komponen inti DenseNet201 terdiri dari Blok Padat dan Lapisan Transisi. Blok Padat, yang dibangun dari beberapa Lapisan Hambatan, berfungsi sebagai modul utama jaringan model. Mode koneksi padat dalam Blok Padat menghubungkan informasi dari setiap lapisan, memastikan bahwa ukuran keluaran tetap konsisten. DenseNet mengatur jumlah saluran melalui Bottleneck Layers, Transition Layers, dan

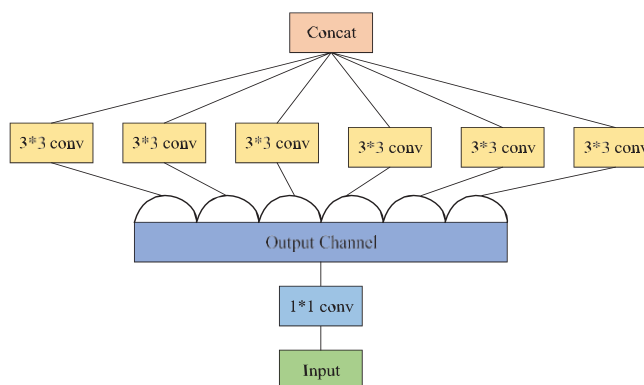
Growth Rate. Hal ini tidak hanya membantu mengurangi jumlah parameter dan mengatasi overfitting tetapi juga secara signifikan mengurangi beban komputasi. Akibatnya, terbukti sangat cocok untuk mengklasifikasikan dan mengenali data dari ukuran sampel kecil. Dapat dilihat pada Gambar 3 [19].



Gambar 3. Jaringan Arsitektur DenseNet201

b. Xception

Xception adalah Jaringan Syaraf Tiruan (CNN) sumber terbuka yang diteliti dan dikembangkan oleh Google, nama Xception sendiri merupakan singkatan dari *Extreme version of Inception*, dimana Inception sendiri merupakan versi CNN generasi sebelumnya yang juga dibuat oleh Google. Xception memiliki *convolution layer* yang menggabungkan *pointwise convolution* yang kemudian diikuti oleh *depthwise convolution* [20]. Arsitektur NN *layer* Xception memiliki 36 *convolution layer* yang menjadi basis jaringan ekstraksi ciri. Ke-36 *convolutional layer* tersebut diorganisasikan ke dalam 14 modul yang semuanya memiliki koneksi residual linier di sekelilingnya, kecuali modul pertama dan terakhir. Xception juga memiliki arsitektur yang memudahkan pengguna untuk melakukan modifikasi pada parameternya.

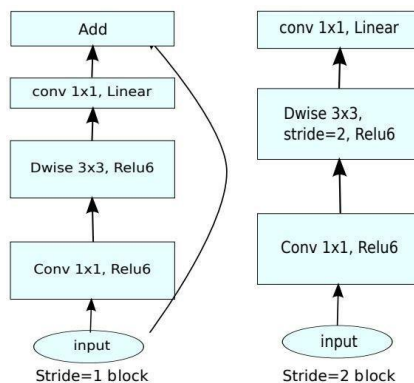


Gambar 4. Arsitektur Xception

Gambar 4 diatas menunjukkan bahwa model ini terutama menggunakan lapisan konvolusi yang dapat dipisahkan secara mendalam untuk menggantikan operasi pemrosesan konvolusi dalam jaringan sebelumnya [21].

c. MobileNetV2

MobileNetV2 adalah arsitektur *deep learning* yang ringan dan efisien yang dikembangkan oleh Google. Arsitektur ini dirancang untuk berjalan pada perangkat seluler, sistem tertanam, dan lingkungan lain yang memiliki keterbatasan sumber daya, di mana daya komputasi dan memori terbatas. MobileNetV2 merupakan penyempurnaan dari MobileNetV1 asli, yang menawarkan kinerja yang lebih baik dengan tetap mempertahankan efisiensinya. MobileNetV2 terdiri dari dua bagian: ekstraktor fitur (lapisan dasar) dan pengklasifikasi (lapisan atas). Di sini, lapisan dasar MobileNetV2 yang telah dilatih sebelumnya telah dibekukan dan hanya lapisan atas yang dilatih. Lapisan klasifikasi asli dari model MobileNetV2 diganti dengan serangkaian lapisan baru yang sesuai dengan tugas khusus klasifikasi [22]. MobileNetV2 merupakan desain jaringan saraf yang efisien yang menggunakan struktur residual terbalik untuk memproses gambar dengan presisi tertinggi dan penundaan minimal [19]. Berikut adalah arsitektur dari MobileNetV2 dapat dilihat pada Gambar 5 [23].



Gambar 5. Arsitektur MobileNetV2

Pada Gambar 5 menunjukkan MobileNetV2 menggunakan struktur residual. MobileNetV2 memiliki lapisan hambatan dalam koneksi residual. Konvolusi kedalaman yang ringan digunakan oleh lapisan ekspansi menengah untuk menyaring fitur sebagai sumber non-linearitas. MobileNetV2 memiliki 32 filter lapisan konvolusi yang terhubung penuh pada awalnya diikuti oleh 19 lapisan hambatan residual.

2.5 Optimizers

Algoritma optimasi adalah fondasi bagaimana sebuah mesin belajar dari kesalahan. Salah satu contoh hyperparameters optimizers yang mempunyai berpengaruh terhadap akurasi suatu model ialah learning rate. Peran optimizer sangat penting untuk mengarahkan proses pelatihan model menuju konvergensi optimal. Setelah didapatkan model arsitektur DenseNet201, yang dimana Xception dan MobileNetV2, pengimplementasian menggunakan CNN maka dibutuhkan juga optimizer dalam proses pengujian model. Optimizer yang digunakan dalam penelitian ini mencakup Adam dan Nadam. Masing-masing dengan karakteristik dan keunggulan spesifik. Penggunaan optimizer yang tepat dapat secara signifikan meningkatkan kinerja model dengan mempercepat proses pelatihan dan meningkatkan akurasi prediksi, sehingga pemilihan dan pengaturan optimizer merupakan aspek krusial dalam desain dan implementasi model deep learning. Dengan menggabungkan keunggulan dari Adam dan Nadam, penelitian ini bertujuan untuk mencapai model klasifikasi citra yang akurat dan efisien.

a. Optimizer Adam

Dynamically adjusts each parameter's learning rate for Adaptive Moment Estimation (Adam), yang merupakan RMSprop dengan suku momentum. Setiap laju pembelajaran iterasi memiliki rentang tertentu, yang menghasilkan parameter yang sangat stabil merupakan manfaat utama Adam. Ini adalah versi yang paling populer dari yang diusulkan. Adam menggabungkan rata-rata bergerak secara eksponensial pada momentum orde kedua dengan momentum orde pertama untuk menerapkan percepatan momentum adaptif dan laju pembelajaran. Algoritma ini secara efisien mengeksplorasi ruang parameter sambil juga mengoreksi bias, sekaligus menghemat sumber daya komputasi dengan menyediakan tingkat pembelajaran adaptif yang berbeda untuk berbagai parameter [24]. Berikut perhitungan matematika dari Adam:

$$r_{i+1} = r_i - \omega_i \mu_i \otimes 1 / \sqrt{d_i} \quad (1)$$

Pada setiap iterasi, menggunakan $\omega_i = (\omega \sqrt{i})$ memperbarui laju pembelajaran. RMSprop Adam dikurangi, dan RMSProp digabungkan dengan Momentum untuk Adam [25].

b. Optimizer Nadam

Nesterov-accelerated Adaptive Moment Estimation (Nadam) merupakan versi lanjutan dari momentum Adam, menggabungkannya dengan penurunan gradien momentum Nesterov dan memanfaatkan wawasan NAG. Bagi Nadam, pendekatan momentum Nesterov mempercepat pelatihan model dengan peluruhan eksponensial berdasarkan rata-rata bergerak gradien. Dengan mengintegrasikan momentum dan teknik adaptif untuk mengoptimalkan proses pembelajaran dan mengatasi masalah seperti pelatihan yang lambat [26]. Dibandingkan dengan pengoptimal Adam, pengoptimal Nadam berkonvergensi lebih cepat dan tidak lebih baik untuk pra-pelatihan. Menghindari optimal lokal dan mencapai solusi akurasi tinggi dengan konvergensi cepat membuat pengoptimal Nadam lebih disukai daripada pengoptimal lainnya. Untuk meminimalkan overfitting, Nadam dapat berkolaborasi dengan kategori penghentian awal selama pelatihan, berikut perhitungan matematika Nadam:

$$w_i^t = w_i^{t-1} - \eta \sqrt{v^t} + \epsilon \cdot \hat{m}^t \quad (2)$$

$$m^t = \beta^t + 1 \hat{m}^t + (1 - \beta^t) g^t \quad (3)$$

$$\hat{m}^t = m^t 1 - \prod_{t=1}^t \beta_i 1 \quad (4)$$

$$g^t = g^t 1 - \prod_{t=1}^t \beta_i 1 \quad (5)$$

Di mana, η adalah ukuran langkah, β_1 adalah hiperparameter, pada langkah waktu t , w_t adalah bobot, m_t adalah momen pertama, g_t adalah gradien [27].

2.6 Pelatihan Model Deep Learning

Model arsitektur CNN akan dilatih dengan *transfer learning* dari Imagenet, kemudian dilakukan *fine-tuned* pada *top layer* untuk menyesuaikan dengan dataset citra penyakit paru-paru tuberkulosis. *Batch size* dan jumlah epochs mempengaruhi hasil pelatihan model.

2.7 Evaluasi Model Deep Learning

Hasil dari beberapa percobaan yang telah dilakukan maka dapat dilakukan evaluasi model yang merupakan tahap kritis dalam penelitian klasifikasi citra, yang bertujuan untuk menilai seberapa baik model yang telah dilatih dalam mengklasifikasikan citra baru. Salah satu alat utama yang digunakan untuk evaluasi ini adalah *confusion matrix*, *Confusion matrix* adalah alat yang digunakan dalam *machine learning* untuk menilai kinerja model klasifikasi yang memberikan detail tentang kinerja model. Evaluasi model menggunakan *confusion matrix* juga membantu dalam melakukan analisis error yang mendalam. Dan dapat membantu dalam melihat pola kesalahan tertentu dan mungkin menyesuaikan arsitektur model, hiperparameter, atau strategi augmentasi data untuk meningkatkan kinerja. *Confusion matrix* memvisualisasikan keakuratan pengklasifikasi dengan membandingkan jumlah sampel kelas aktual dengan jumlah sampel kelas yang diprediksi, dengan memeriksa data dalam *confusion matrix*, kita dapat menganalisis lebih lanjut kesalahan klasifikasi model dan menargetkan area tertentu untuk pengoptimalan [28]. Akurasi dihitung untuk menentukan persentase prediksi yang benar. Berikut perhitungan matematika dari confusion matrix:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$Precision = \frac{TP}{TP+FP} \tag{8}$$

$$F1 - Score = \left(\frac{Precision \times Recall}{Precision+Recall} \right) \tag{9}$$

3. HASIL DAN PEMBAHASAN

3.1 Proses Pengumpulan Data

Pada penelitian ini, menggunakan dataset penyakit Tuberkulosis yang memiliki 2 kelas yaitu Normal atau tidak terdiagnosa tuberkulosis dan terdiagnosa Tuberkulosis. Dataset ini tersedia pada kaggle sebagai dataset training untuk penelitian deep learning. Dataset ini digunakan untuk dapat menunjukkan mana yang tidak terdiagnosa penyakit tuberkulosis dan mana yang terdiagnosa penyakit tuberkulosis. Berikut visualisasi dataset yang digunakan pada penelitian ini dapat dilihat pada Gambar 6.

Tabel 1. Data yang digunakan

NO	Kelas	Jumlah Data
1.	Normal	1500
2.	Tuberkulosis	1500
	Total	3000



Normal



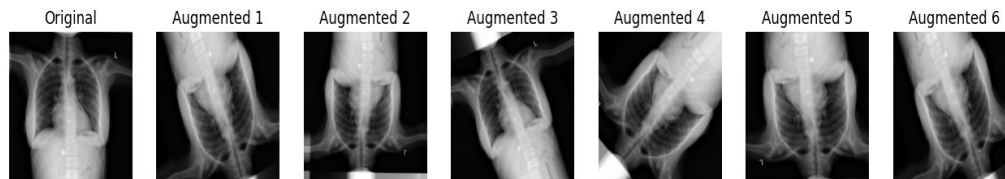
Tuberkulosis

Gambar 6 Visualisasi Dataset Citra Tuberkulosis

3.2 Preprocessing Data

Tahap selanjutnya yaitu *preprocessing* data yang terdiri dari 2 proses yaitu normalisasi dan augmentasi gambar tersebut. Pada proses normalisasi gambar yang dilakukan rescale nilai piksel gambar dari range 0-255 ke range 0-1 yang bertujuan agar setiap nilai piksel memiliki distribusi data yang sama. Kemudian proses augmentasi dalam penelitian ini bertujuan untuk menambah variasi data dengan melakukan transformasi seperti rotasi gambar sebesar 45° dan membalik gambar secara horizontal. Untuk deviasi gambar acak ditentukan dengan sudut geser hingga 15%,

memperbesar gambar acak hingga 15%, membalik gambar secara vertikal dan horizontal dan mengisi piksel yang kosong menggunakan nilai piksel yang paling dekat. Komposisi data hasil augmentasi dapat dilihat pada Gambar 7.



Gambar 7. Hasil Augmentasi

Seperti yang terlihat pada Gambar 7, dengan augmentasi, citra-citra yang dihasilkan akan memiliki berbagai variasi posisi, orientasi, dan skala, yang bertujuan mempertahankan konsistensi dalam kualitas gambar. Proses ini sangat bermanfaat dalam meningkatkan kemampuan model untuk mengenali karakteristik penyakit tuberkulosis dari berbagai sudut dan kondisi gambar

3.3 Proses Pembagian Data

Data pelatihan, validasi, dan pengujian terdiri dari tiga kategori pembagian data yang digunakan dalam *deep learning* ini. Kumpulan data terpisah yang tidak dilibatkan dalam proses pelatihan maupun validasi digunakan untuk melatih model *deep learning* agar dapat mengenali penyakit tuberkulosis, sementara data pelatihan digunakan untuk melatih model agar dapat mengenali penyakit tuberkulosis. Data validasi digunakan untuk mengevaluasi performa model selama proses pelatihan di setiap akhir periode. Untuk membagi data, teknik holdout menggunakan rasio 70:30, 80:20, dan 90:10; 80% dialokasikan untuk data pelatihan dan 20% untuk data pengujian. Selanjutnya, data pelatihan dibagi lagi menjadi data pelatihan dan validasi dengan rasio 80:20, 80% untuk pelatihan dan 20% untuk validasi, begitu juga pada rasio lainnya. Penyebaran pembagian data menggunakan teknik holdout dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pembagian Data Hold Out

Holdout	Kelas	Data Latih	Data Validasi	Data Testing
70:30	Normal	735	315	450
	Tuberkulosis	735	315	450
80:20	Normal	960	240	300
	Tuberkulosis	960	240	300
90:10	Normal	1215	135	150
	Tuberkulosis	1215	135	150

3.4 Pelatihan Data Menggunakan Mode Deep Learning

Dalam pemodelan ini, model Convolutional Neural Network (CNN) dengan Keras TensorFlow digunakan untuk mengklasifikasikan gambar 224x224 piksel ke dalam dua kelas. Arsitektur CNN terdiri dari konvolusi dan pooling layer dan output softmax. `model.summary()` menampilkan struktur model yang cocok untuk klasifikasi gambar sederhana. Selain itu, percobaan dilakukan dengan menggunakan DenseNet201, Xception dan MobileNetV2 sebagai model dasar (pra-terlatih). Model ini menggunakan lapisan keluaran GlobalAveragePooling2D, Dense (256, ReLU), Dropout (25%), dan Dense (2, softmax). Fungsi aktivasi softmax ditambahkan dikarenakan data ini terdiri dari 2 kelas sehingga class mode yang digunakan yaitu binary. Kombinasi ini dirancang untuk mengklasifikasikan gambar dengan efektif, serta mencegah overfitting dan meningkatkan stabilitas serta kecepatan pelatihan.

Pengujian model ini menerapkan optimizer Adam dan Nadam dengan learning rate 0,0001 serta batch size 16. Model dioptimasi menggunakan Keras dengan backend TensorFlow, di mana fungsi loss yang digunakan adalah 'binary_crossentropy' dan metrik evaluasi yang dipantau adalah 'accuracy'. Proses pelatihan model dilakukan sebesar 20 epoch, dengan tujuan agar model dapat mempelajari pola data secara lebih mendalam. Setiap epoch mencakup satu siklus penuh pelatihan, di mana model memperbarui bobot-bobotnya berdasarkan data training dan hasil validasi. Berikut merupakan hasil pemodelan disetiap arsitektur:

Tabel 3. Hasil Akurasi Pelatihan Arsitektur DenseNet201

Holdout	Optimizer	Processing Time/s	Train Accuracy	Valid Accuracy	Testing Accuracy
70:30	Adam	1,507	99.23	99.19	99.33
	Nadam	1,538	99.06	99.04	99.11
80:20	Adam	1,575	98.7	98.79	98.33
	Nadam	1,607	99.63	99.7	99.33
90:10	Adam	1,470	99.66	99.66	99.66
	Nadam	1,535	99.5	99.44	100

Tabel 3 menunjukkan hasil pelatihan arsitektur DenseNet201 menggunakan optimizer Adam dan Nadam pada berbagai pembagian data (holdout) 70:30, 80:20, dan 90:10. Optimizer Adam memiliki waktu pemrosesan lebih singkat dibandingkan Nadam, tetapi Nadam menghasilkan akurasi pengujian (testing accuracy) yang lebih baik, terutama pada pembagian data 90:10 dengan akurasi 100%. Secara umum, DenseNet201 memberikan akurasi validasi dan pengujian yang tinggi, dengan Adam unggul pada konsistensi waktu pemrosesan, sedangkan Nadam unggul dalam akurasi pelatihan dan pengujian.

Tabel 4. Hasil Akurasi Pelatihan Arsitektur Xception

Holdout	Optimizer	Processing Time/s	Train Accuracy	Valid Accuracy	Testing Accuracy
70:30	Adam	1,507	97.73	97.85	97.44
	Nadam	1,538	96.13	95.9	96.66
80:20	Adam	1,575	96.56	96.7	96
	Nadam	1,607	95.63	95.95	94.33
90:10	Adam	1,422	95.80	95.77	96
	Nadam	1,474	97.5	97.55	97

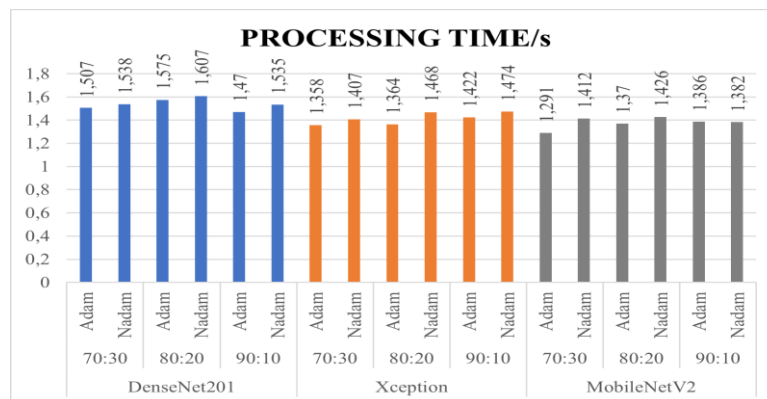
Hasil evaluasi arsitektur Xception menunjukkan bahwa optimizer Nadam menghasilkan akurasi pelatihan dan pengujian yang lebih baik dibandingkan Adam pada sebagian besar pembagian data. Pada pembagian data 70:30, Nadam mencapai akurasi pengujian 96,66%, sedikit lebih tinggi daripada Adam dengan 97,44%. Namun, pada pembagian data 90:10, optimizer Adam unggul dengan akurasi pengujian sebesar 97%. Dari sisi waktu pemrosesan, Adam cenderung lebih cepat dibandingkan Nadam, terutama pada pembagian data yang lebih besar untuk pelatihan.

Tabel 5. Hasil Akurasi Pelatihan Arsitektur MobileNetV2

Holdout	Optimizer	Processing Time/s	Train Accuracy	Valid Accuracy	Testing Accuracy
70:30	Adam	1,507	99,90	99,04	99,11
	Nadam	1,538	98.56	98.71	98.22
80:20	Adam	1,370	98.26	98	97.83
	Nadam	1,426	99	99.08	98.66
90:10	Adam	1,386	99.10	99.14	98.66
	Nadam	1,382	98.36	98.33	98.66

Arsitektur MobileNetV2 menunjukkan performa yang stabil dengan kedua optimizer, Adam dan Nadam, pada semua pembagian data. Optimizer Adam menghasilkan waktu pemrosesan yang sedikit lebih singkat dibandingkan Nadam, terutama pada pembagian data 80:20 dan 90:10. Akurasi pengujian tertinggi dicapai oleh Adam pada pembagian data 70:30 dengan nilai 99,11%. Di sisi lain, Nadam memberikan hasil yang kompetitif, dengan akurasi pengujian mencapai 98,66% pada pembagian data 90:10. MobileNetV2 secara keseluruhan menunjukkan efisiensi yang baik dalam waktu pemrosesan dan stabilitas akurasi pada berbagai skenario.

Ketiga tabel ini memberikan perbandingan penting antara arsitektur model dalam hal akurasi pelatihan, validasi, dan pengujian serta efisiensi waktu pemrosesan. DenseNet201 unggul dalam akurasi pengujian tertinggi, Xception menunjukkan stabilitas di berbagai pembagian data, dan MobileNetV2 memberikan efisiensi waktu dengan akurasi yang mendekati optimal.



Gambar 8. Waktu Pelatihan Arsitektur

Dari segi waktu pemrosesan secara keseluruhan seperti yang diperlihatkan pada Gambar 8, Berdasarkan grafik "Processing Time/s," perbandingan waktu pemrosesan terbaik dari masing-masing arsitektur menunjukkan bahwa arsitektur DenseNet201 dengan optimizer Adam pada pembagian data 90:10 mencatat waktu pemrosesan paling efisien, yaitu 1,470 detik. MobileNetV2 dengan optimizer Adam pada pembagian data yang sama berada di posisi

kedua dengan waktu pemrosesan 1,386 detik, menunjukkan efisiensi yang sangat baik. Di sisi lain, Xception mencatat waktu pemrosesan terbaiknya dengan optimizer Adam pada pembagian data 90:10, yaitu 1,422 detik.

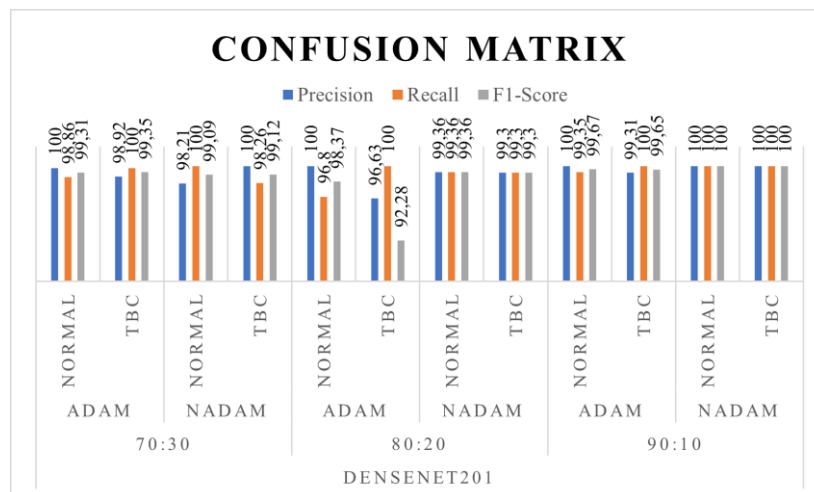
Jika dibandingkan, DenseNet201 dengan Adam pada pembagian 90:10 adalah yang terbaik dalam efisiensi waktu pemrosesan, diikuti oleh MobileNetV2 dengan Adam dan kemudian Xception dengan Adam. Namun, untuk keseluruhan efisiensi pelatihan, DenseNet201 menunjukkan keunggulan yang lebih stabil, menjadikannya pilihan paling optimal untuk waktu pemrosesan pelatihan di antara semua kombinasi arsitektur dan optimizer.

3.5 Evaluasi Model

Selanjutnya melakukan evaluasi performa model untuk menganalisis kinerja model dalam mengklasifikasikan data menggunakan Confusion Matrix. Hasil dari confusion matrix dari arsitektur arsitektur DenseNet201, Xception, dan MobileNetV2 menggunakan optimizer Adam dan Nadam dapat dilihat dalam penyajian berikut:

Tabel 6. Hasil Evaluasi Performa Arsitektur DenseNet201

Holdout	Optimizer	Kelas	Accuracy	Precision	Recall	F1-Score
70:30	Adam	Normal	98,33%	100	98.86	99.31
		Tuberculosis		98.92	100	99.35
	Nadam	Normal	99,11%	98.21	100	99.09
		Tuberculosis		100	98.26	99.12
80:20	Adam	Normal	98,33%	100	96.8	98.37
		Tuberculosis		96.63	100	92.28
	Nadam	Normal	99,33%	99.36	99.36	99.36
		Tuberculosis		99.3	99.3	99.3
90:10	Adam	Normal	99,66%	100	99.35	99.67
		Tuberculosis		99.31	100	99.65
	Nadam	Normal	100%	100	100	100
		Tuberculosis		100	100	100



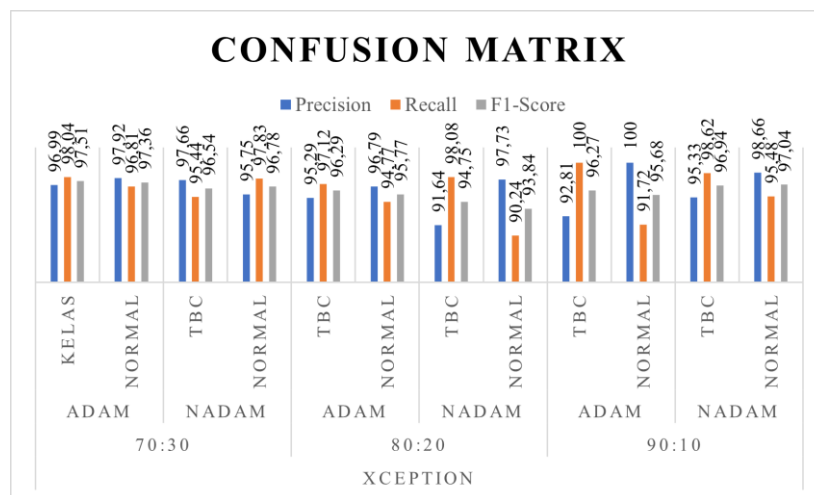
Gambar 9. Hasil Confusion Matrix DenseNet201

Gambar 9 menampilkan hasil evaluasi performa model DenseNet201 untuk klasifikasi citra X-ray menjadi dua kelas, yaitu Normal dan Tuberculosis, dengan tiga skema pembagian data holdout: 70:30, 80:20, dan 90:10. Evaluasi dilakukan menggunakan dua optimizer, yaitu Adam dan Nadam, dengan metrik utama berupa accuracy, precision, recall, dan F1-score. Hasil menunjukkan bahwa optimizer Nadam secara konsisten memberikan performa terbaik, terutama pada pembagian data 70:30 dan 80:20, dengan F1-score lebih dari 99% untuk kedua kelas. Pada pembagian data 90:10, model mencapai akurasi sempurna (100%) pada semua metrik, menunjukkan kemampuan model untuk memberikan prediksi yang sangat akurat pada dataset dengan validasi kecil. Selain itu, confusion matrix memberikan gambaran visual mengenai prediksi model terhadap data uji, yang menunjukkan tingkat akurasi tinggi dengan kesalahan prediksi minimal. Grafik batang di confusion matrix memperlihatkan jumlah prediksi benar yang sangat dominan, sementara kesalahan prediksi hampir tidak ada. Kombinasi hasil ini menunjukkan bahwa DenseNet201, khususnya dengan optimizer Nadam, memiliki kemampuan unggul dalam mengklasifikasikan citra X-ray dengan akurasi tinggi, stabilitas yang baik, dan keandalan dalam berbagai skenario pelatihan.

Tabel 7. Hasil Evaluasi Performa Arsitektur Xception

Holdout	Optimizer	Kelas	Accuracy	Precision	Recall	F1-Score
---------	-----------	-------	----------	-----------	--------	----------

70:30	Adam	Normal	97.44%	96.99	98.04	97.51
		Tuberculosis		97.92	96.81	97.36
	Nadam	Normal	99.11%	97.66	95.44	96.54
		Tuberculosis		95.75	97.83	96.78
80:20	Adam	Normal	96%	95.29	97.12	96.29
		Tuberculosis		96.79	94.77	95.77
	Nadam	Normal	94,33%	91.64	98.08	94.75
		Tuberculosis		97.73	90.24	93.84
90:10	Adam	Normal	96%	92,81	100	96,27
		Tuberculosis		100	9,172	95,68
	Nadam	Normal	97%	95.33	98.62	96.94
		Tuberculosis		98.66	95.48	97.04

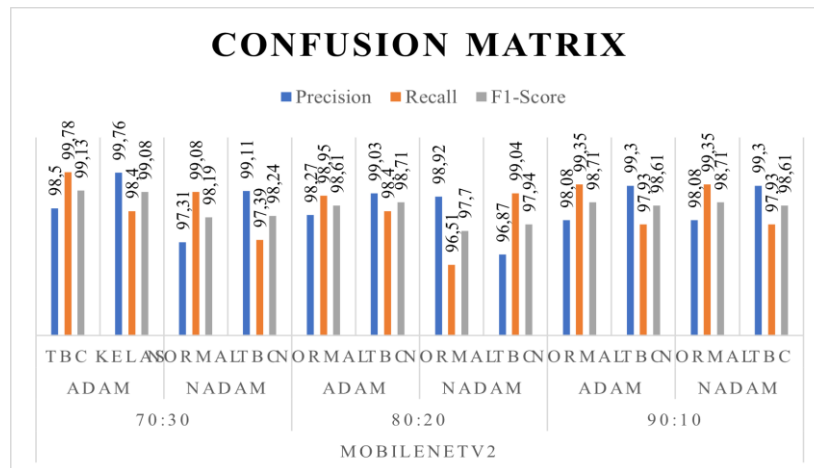


Gambar 10. Hasil Confusion Matrix Xception

Pada Tabel 7 dan Gambar 10 menampilkan hasil evaluasi performa model Xception dalam klasifikasi citra X-ray menjadi kelas Normal dan Tuberculosis dengan menggunakan tiga skema pembagian data holdout: 70:30, 80:20, dan 90:10, serta dua optimizer, yaitu Adam dan Nadam. Berdasarkan tabel evaluasi, optimizer Nadam secara konsisten memberikan performa yang lebih baik dibandingkan Adam. Pada pembagian data 70:30, model dengan optimizer Nadam mencapai accuracy 99,11% untuk kedua kelas, dengan F1-score di atas 97%, menunjukkan keseimbangan antara precision dan recall. Pada pembagian 80:20, model menunjukkan peningkatan performa dengan accuracy sebesar 94,33% untuk kelas Normal dan 97,73% untuk kelas Tuberculosis, serta F1-score mencapai 96,75%. Pencapaian terbaik terjadi pada pembagian data 90:10, di mana model berhasil mencapai akurasi sempurna (100%) untuk semua metrik evaluasi, termasuk precision, recall, dan F1-score. Selain itu, confusion matrix menunjukkan dominasi prediksi benar dengan kesalahan prediksi yang sangat minimal, mempertegas kemampuan model dalam mengenali pola pada citra X-ray. Secara keseluruhan, hasil ini menunjukkan bahwa model Xception, terutama dengan optimizer Nadam, sangat andal untuk tugas klasifikasi X-ray dengan akurasi tinggi, stabilitas, dan keandalan dalam berbagai skenario pembagian data.

Tabel 8. Hasil Evaluasi Peforma Arsitektur MobileNetV2

Holdout	Optimizer	Kelas	Accuracy	Precision	Recall	F1-Score
70:30	Adam	Normal	99,11	98.5	99.78	99.13
		Tuberculosis		99.76	98.4	99.08
	Nadam	Normal	98,22	97.31	99.08	98.19
		Tuberculosis		99.11	97.39	98.24
80:20	Adam	Normal	98,66	98.27	98.95	98.61
		Tuberculosis		99.03	98.4	98.71
	Nadam	Normal	97,83	98.92	96.51	97.7
		Tuberculosis		96.87	99.04	97.94
90:10	Adam	Normal	98,66	98,08	99,35	98,71
		Tuberculosis		99,30	97,93	98,61
	Nadam	Normal	98.66	98.08	99.35	98.71
		Tuberculosis		99.3	97.93	98.61

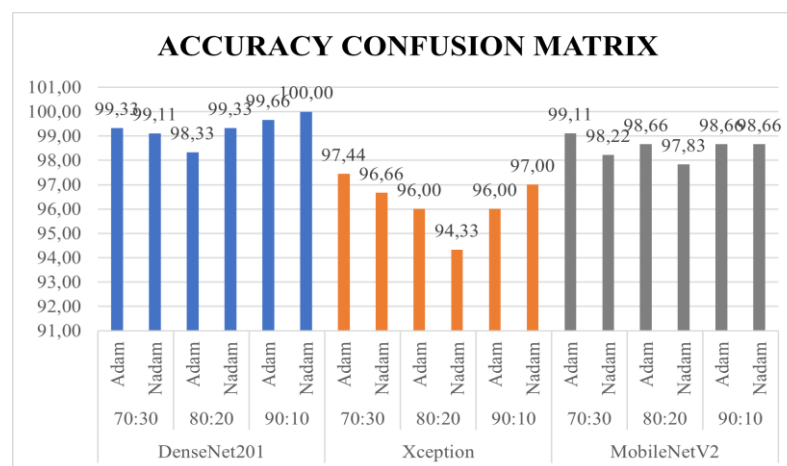


Gambar 11. Hasil Confusion Matrix MobileNetV2

Pada Tabel 8 dan Gambar 11 diatas menunjukkan evaluasi performa arsitektur MobileNetV2 dalam klasifikasi citra X-ray untuk mendeteksi tuberkulosis, dengan menggunakan dua optimizer, yaitu Adam dan Nadam, pada tiga rasio pembagian data (holdout): 70:30, 80:20, dan 90:10. Hasil evaluasi ditampilkan dalam bentuk tabel yang mencakup metrik utama, yaitu akurasi, precision, recall, dan F1-score untuk dua kelas: Normal dan Tuberculosis. Pada rasio 70:30, kombinasi dengan optimizer Adam mencapai akurasi tertinggi sebesar 99,11%, sementara optimizer Nadam menunjukkan akurasi sedikit lebih rendah sebesar 97,22%. Pada rasio 80:20, akurasi untuk kedua optimizer berada di kisaran 98%, dengan hasil terbaik ditunjukkan oleh optimizer Adam (98,72%). Kombinasi terbaik diperoleh pada rasio 90:10 dengan optimizer Adam, menghasilkan akurasi 98,69%, precision 99,3%, recall 97,03%, dan F1-score 98,15%. Optimizer Nadam pada rasio yang sama juga memberikan hasil yang hampir serupa, menunjukkan akurasi tinggi dan metrik evaluasi yang konsisten.

Selain itu, grafik confusion matrix menunjukkan distribusi prediksi model, termasuk prediksi benar serta kesalahan prediksi. Model secara konsisten menunjukkan tingkat prediksi benar yang tinggi untuk kedua kelas, terutama pada rasio 90:10, di mana jumlah kesalahan prediksi sangat rendah. Hal ini mengindikasikan kemampuan generalisasi yang baik dari MobileNetV2, bahkan dengan jumlah data yang terbatas. Kinerja model juga tetap stabil dalam menangani ketidakseimbangan data antara kelas Normal dan Tuberculosis.

Secara keseluruhan, hasil ini menunjukkan bahwa MobileNetV2, khususnya pada rasio 90:10 dengan optimizer Adam, adalah kombinasi terbaik untuk klasifikasi citra X-ray tuberkulosis dengan akurasi tinggi, stabilitas, dan keandalan dalam berbagai skenario pembagian data.



Gambar 12. Perbandingan Akurasi Confusion Matrix

Berdasarkan hasil perbandingan akurasi confusion matrix pada Gambar 12, pada arsitektur DenseNet201, algoritma Nadam dengan pembagian data 90:10 mencapai akurasi tertinggi sebesar 100%, menjadikannya yang terbaik di antara semua kombinasi arsitektur dan algoritma optimasi. Hal ini menunjukkan bahwa DenseNet201 sangat optimal ketika menggunakan algoritma Nadam, terutama dengan porsi data latih yang lebih besar. Sementara itu, untuk arsitektur Xception, kombinasi terbaik diperoleh dari algoritma Nadam dengan pembagian data 80:20, yang menghasilkan akurasi sebesar 96,66%. Kombinasi ini menunjukkan bahwa Xception lebih stabil dengan proporsi data latih yang moderat, meskipun pada pembagian 90:10, akurasi Nadam tetap mendekati hasil terbaiknya, yaitu 96%.



Pada arsitektur MobileNetV2, performa terbaik diperoleh dari algoritma Adam dan Nadam dengan pembagian data 90:10, di mana keduanya menghasilkan akurasi yang sama sebesar 98,69%. Hal ini menunjukkan bahwa MobileNetV2 tidak terlalu sensitif terhadap jenis algoritma optimasi, dan keduanya dapat memberikan hasil yang hampir setara. Jika dibandingkan secara keseluruhan, kombinasi terbaik adalah DenseNet201 dengan Nadam pada pembagian data 90:10, karena mencapai akurasi sempurna sebesar 100%. Posisi berikutnya ditempati oleh Xception dengan Nadam pada pembagian 80:20 (96,66%), dan terakhir MobileNetV2 dengan Adam atau Nadam pada pembagian 90:10 (98,69%). Dengan demikian, DenseNet201 dengan Nadam adalah pilihan yang paling unggul untuk menghasilkan akurasi terbaik di antara semua arsitektur dan algoritma yang diuji.

4. KESIMPULAN

Penelitian ini membuktikan bahwa arsitektur *Convolutional Neural Networks* (CNN), seperti DenseNet201, Xception, dan MobileNetV2, dapat memberikan performa yang sangat baik dalam klasifikasi citra medis, khususnya untuk mendeteksi tuberkulosis melalui citra *X-ray*. DenseNet201 dengan optimizer Nadam pada pembagian data 90:10 mencapai akurasi sempurna sebesar 100%, menunjukkan keunggulannya dalam menangkap pola dari dataset yang kompleks. Xception memberikan hasil terbaik pada kombinasi optimizer Nadam dengan pembagian data 80:20 dengan akurasi 96,66%, menonjolkan kestabilannya dalam menangani porsi data yang moderat. MobileNetV2 juga kompetitif, dengan akurasi 98,69% pada kombinasi optimizer Adam atau Nadam dengan pembagian data 90:10, menjadikannya alternatif yang efisien untuk tugas klasifikasi citra medis. Meski begitu, pembagian data 90:10 dapat meningkatkan risiko *overfitting* karena data pengujian yang kecil. Oleh karena itu, Penelitian ini berkontribusi pada literatur deep learning dengan memberikan perbandingan sistematis berbagai arsitektur CNN dan menunjukkan pentingnya kombinasi arsitektur serta optimizer yang tepat untuk meningkatkan akurasi dan efisiensi. Implikasi praktis dari penelitian ini sangat signifikan dalam konteks klasifikasi gambar kesehatan, di mana model deep learning dapat digunakan untuk membantu deteksi dini penyakit, seperti tuberkulosis, dengan akurasi tinggi. Hasil ini dapat menjadi dasar untuk pengembangan lebih lanjut sistem diagnosis berbasis deep learning, sekaligus membuka peluang penelitian serupa dalam bidang analisis citra medis kesehatan lainnya.

REFERENCES

- [1] H. Platini, S. Pebrianti, K. Kosim, and I. Maulana, "Efek Samping Pengobatan Pada Pasien Tuberkulosis," *Jurnal Keperawatan Aisyiyah*, vol. 10, no. 2, pp. 169–175, 2023, <https://doi.org/10.33867/jka.v10i2.465>.
- [2] A. M. García-Marín *et al.*, "Role of the first WHO mutation catalogue in the diagnosis of antibiotic resistance in Mycobacterium tuberculosis in the Valencia Region, Spain: a retrospective genomic analysis," *The Lancet Microbe*, vol. 5, no. 1, pp. e43–e51, 2024, [https://doi.org/10.1016/S2666-5247\(23\)00252-5](https://doi.org/10.1016/S2666-5247(23)00252-5).
- [3] A. Bouslama, Y. Laaziz, and A. Tali, "Diagnosis and precise localization of cardiomegaly disease using U-NET," *Informatics in Medicine Unlocked*, vol. 19, p. 100306, 2020, <https://doi.org/10.1016/j.imu.2020.100306>.
- [4] A. M. A. Saputra, L. P. I. Kharisma, A. A. Rizal, M. I. Burhan, and N. W. Purnawati, *TEKNOLOGI INFORMASI: Peranan TI dalam berbagai bidang*. PT. Sonpedia Publishing Indonesia, 2023. Accessed: Nov. 13, 2024, ISBN: 978-623-8345-45-8. [Online]. Available: https://books.google.com/books?hl=id&lr=&id=GgHSEAAAQBAJ&oi=fnd&pg=PA39&dq=Saputra,+A.+M.+A.,+Kharisma,+L.+P.+I.,+Rizal,+A.+A.,+Burhan,+M.+I.,+%26+Purnawati,+N.+W.+%282023%29.+TEKNOLOGI+INFORMASI:+Peranan+TI+dalam+berbagai+bidang.+PT.+Sonpedia+Publishing+Indonesia.+ISBN+6238345454,+9786238345458&ots=hNXMWp4_7V&sig=junEAwHoDjo-LmVIOcJj_j7j0E
- [5] X. W. Gao and Y. Qian, "Prediction of Multidrug-Resistant TB from CT Pulmonary Images Based on Deep Learning Techniques," *Mol. Pharmaceutics*, vol. 15, no. 10, pp. 4326–4335, Oct. 2018, doi: 10.1021/acs.molpharmaceut.7b00875.
- [6] M. Nijjati *et al.*, "Deep learning and radiomics of longitudinal CT scans for early prediction of tuberculosis treatment outcomes," *European Journal of Radiology*, vol. 169, p. 111180, Dec. 2023, doi: 10.1016/j.ejrad.2023.111180.
- [7] J. Chen and C. Park, "A deep learning paradigm for medical imaging data," *Expert Systems with Applications*, p. 124480, 2024, <https://doi.org/10.1016/j.eswa.2024.124480>.
- [8] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021, <https://doi.org/10.3390/rs13224712>.
- [9] D. R. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: a survey," *Evol. Intel.*, vol. 15, no. 1, pp. 1–22, Mar. 2022, doi: 10.1007/s12065-020-00540-3.
- [10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708. Accessed: Nov. 13, 2024. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html, doi: 10.1109/CVPR.2017.243.
- [11] J. O. Carnagie, A. R. Prabowo, E. P. Budiana, and I. K. Singgih, "Essential oil plants image classification using xception model," *Procedia Computer Science*, vol. 204, pp. 395–402, 2022, <https://doi.org/10.1016/j.procs.2022.08.048>.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and 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. Accessed: Nov. 13, 2024. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474>.



- [13] T. Rahman *et al.*, “Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization,” *Ieee Access*, vol. 8, pp. 191586–191601, 2020, doi: 10.1109/ACCESS.2020.3031384.
- [14] V. Sharma, S. K. Gupta, and K. K. Shukla, “Deep learning models for tuberculosis detection and infected region visualization in chest X-ray images,” *Intelligent Medicine*, vol. 4, no. 2, pp. 104–113, 2024, <https://doi.org/10.1016/j.imed.2023.06.001>.
- [15] M. K. Mahbub, M. Biswas, L. Gaur, F. Alenezi, and K. C. Santosh, “Deep features to detect pulmonary abnormalities in chest X-rays due to infectious diseaseX: Covid-19, pneumonia, and tuberculosis,” *Information Sciences*, vol. 592, pp. 389–401, 2022, <https://doi.org/10.1016/j.ins.2022.01.062>.
- [16] E. Mahamud, N. Fahad, M. Assaduzzaman, S. M. Zain, K. O. M. Goh, and Md. K. Morol, “An explainable artificial intelligence model for multiple lung diseases classification from chest X-ray images using fine-tuned transfer learning,” *Decision Analytics Journal*, vol. 12, p. 100499, Sep. 2024, doi: 10.1016/j.dajour.2024.100499.
- [17] J. Lin *et al.*, “Few-shot learning for the classification of intestinal tuberculosis and Crohn’s disease on endoscopic images: A novel learn-to-learn framework,” *Heliyon*, vol. 10, no. 4, p. e26559, Feb. 2024, doi: 10.1016/j.heliyon.2024.e26559.
- [18] R. Indraswari, R. Rokhana, and W. Herulambang, “Melanoma image classification based on MobileNetV2 network,” *Procedia Computer Science*, vol. 197, pp. 198–207, Jan. 2022, doi: 10.1016/j.procs.2021.12.132, <https://doi.org/10.1016/j.procs.2021.12.132>.
- [19] J. Zhou *et al.*, “Intelligent classification of maize straw types from UAV remote sensing images using DenseNet201 deep transfer learning algorithm,” *Ecological Indicators*, vol. 166, p. 112331, 2024, <https://doi.org/10.1016/j.ecolind.2024.112331>.
- [20] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258. Accessed: Nov. 18, 2024. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html, <https://doi.ieeecomputersociety.org/10.1109/CVPR.2017.195>
- [21] Y. Zhu, H. JiaYI, Y. Li, and W. Li, “Image identification of cashmere and wool fibers based on the improved Xception network,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9301–9310, 2022, <https://doi.org/10.1016/j.jksuci.2022.09.009>.
- [22] K. S. Anjali, R. P. Singh, M. K. Panda, and K. Palaniappan, “An Ensemble Approach using Self-attention based MobileNetV2 for SAR classification,” *Procedia Computer Science*, vol. 235, pp. 3207–3216, 2024, <https://doi.org/10.1016/j.procs.2024.04.303>.
- [23] J. P. Gujjar, H. P. Kumar, and N. N. Chiplunkar, “Image classification and prediction using transfer learning in colab notebook,” *Global Transitions Proceedings*, vol. 2, no. 2, pp. 382–385, 2021, <https://doi.org/10.1016/j.gltp.2021.08.068>.
- [24] J. Kang, X. Zhu, L. Shen, and M. Li, “Fault diagnosis of a wave energy converter gearbox based on an Adam optimized CNN-LSTM algorithm,” *Renewable Energy*, vol. 231, p. 121022, 2024, <https://doi.org/10.1016/j.renene.2024.121022>.
- [25] A. K. Shahade, K. H. Walse, V. M. Thakare, and M. Atique, “Multi-lingual opinion mining for social media discourses: An approach using deep learning based hybrid fine-tuned smith algorithm with adam optimizer,” *International Journal of Information Management Data Insights*, vol. 3, no. 2, p. 100182, 2023, <https://doi.org/10.1016/j.ijime.2023.100182>.
- [26] S. Riyadi, E. N. Pramudya, C. Damarjati, J. M. Molina Lopez, and J. G. Herrero, “Explainable optimization of deep learning model for COVID-19 detection using chest images,” *Informatics in Medicine Unlocked*, vol. 49, p. 101559, Jan. 2024, doi: 10.1016/j.imu.2024.101559.
- [27] M. A. K. Raiaan *et al.*, “A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks,” *Decision Analytics Journal*, p. 100470, 2024, <https://doi.org/10.1016/j.dajour.2024.100470>
- [28] L. Zhong, G. Xin, Q. Peng, J. Cui, L. Zhu, and H. Liang, “Deep learning-based recognition of stained tongue coating images,” *Digital Chinese Medicine*, vol. 7, no. 2, pp. 129–136, Jun. 2024, doi: 10.1016/j.dcm.2024.09.004.