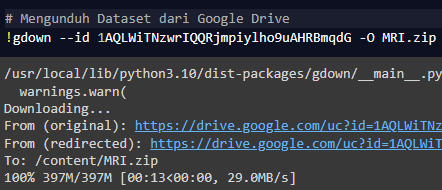
**Hasil Penelitian dan Pembahasan**

**5.1. Persiapan Dataset**

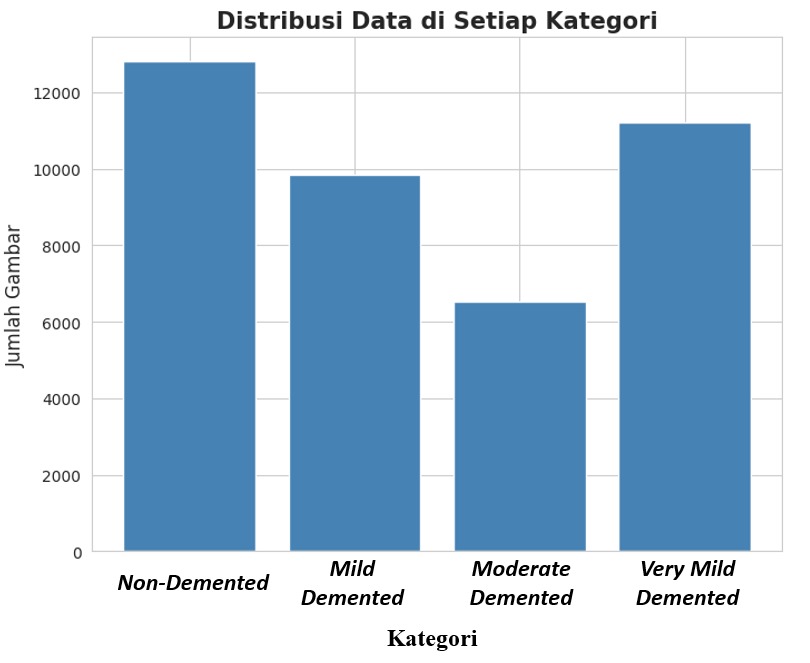
Dalam penelitian ini, digunakan dataset berupa data sekunder yang diambil dari sumber *open source Kaggle.com*. Dataset ini dikumpulkan, dilakukan augmentasi, dan dipublikasikan oleh pengguna bernama Uraninjo pada tahun (2022). Terdapat total 40.384 gambar MRI otak pasien, yang terbagi ke dalam empat kategori atau tahapan Alzheimer: *Non-Demented*, *Very Mild Demented*, *Mild Demented*, dan *Moderate Demented*. Dataset tersebut diunduh ke penyimpanan lokal, kemudian diunggah ke Google Drive sebagai basis data. Setelah itu, dataset dimuat ke dalam Google Colab dengan membuat folder khusus. Proses ini dapat dilihat pada gambar 5.1.



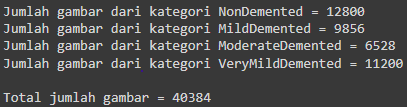
Gambar 5.1 Pengunduhan dataset

**5.2. *Exploratory Data Analysis* (EDA)**

Dataset yang telah dimuat ke dalam Google Colab kemudian dieksplorasi untuk memperoleh informasi penting terkait isi dataset. Proses eksplorasi ini meliputi penayangan gambar dari setiap kategori, penghitungan jumlah data per kategori, visualisasi distribusi data, serta identifikasi berbagai ekstensi *file* gambar yang tersedia. Langkah eksplorasi ini penting untuk merencanakan tahap-tahap selanjutnya agar data siap digunakan dalam proses pelatihan.

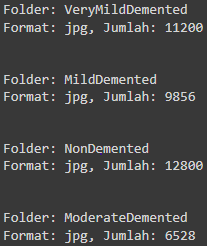


Gambar 5.2 Grafik distribusi dataset



Gambar 5.3 Jumlah gambar pada setiap kategori dan totalnya

Berdasarkan gambar 5.2 dan 5.3 terlihat bahwa jumlah data pada setiap kategori tidak sama. Meskipun distribusi dataset yang digunakan tidak seimbang, dengan jumlah data yang bervariasi di setiap kategori, penyamarataan distribusi data tidak direncanakan. Keputusan ini didasarkan pada pertimbangan bahwa model yang dikembangkan diharapkan mampu menangani ketidakseimbangan data secara efektif. Ketidakseimbangan ini juga dapat memberikan tantangan tambahan bagi model selama proses pembelajaran, yang berpotensi meningkatkan kemampuan generalisasi tanpa perlu melakukan penyamarataan distribusi data.



Gambar 5.4 Ekstensi file dan jumlah datanya

Hasil eksplorasi yang ditampilkan pada gambar 5.4 menunjukkan bahwa semua file gambar memiliki ekstensi yang sama, yaitu *.jpg* atau *.jpeg* (Joint Photographic Experts Group). Format ini merupakan format gambar umum yang sering digunakan untuk menyimpan foto dengan kualitas baik dan ukuran file yang lebih kecil. Karena semua ekstensi file sudah seragam, proses penyeragaman ekstensi tidak perlu dilakukan.

**5.3. Implementasi Pra-pemrosesan Data**

Berdasarkan informasi yang diperoleh dari tahap *Exploratory Data Analysis* (EDA), langkah berikutnya adalah melakukan penyesuaian dataset untuk memastikan dataset siap digunakan dalam proses pelatihan dan pengujian. Penyesuaian ini mencakup pengubahan ukuran gambar, penentuan *batch size*, serta pengacakan dataset.

**5.3.1. Implementasi Pemrosesan Data**

Pemrosesan data dilakukan bertujuan untuk menyesuaikan dataset dengan kebutuhan model dalam proses pelatihan dan pengujian. Penyesuaian ini meliputi pengubahan ukuran gambar, penetapan *batch size*, serta pengacakan dataset (*shuffle*). Pengubahan ukuran gambar (*image resizing*) dilakukan untuk menyamakan dimensi gambar agar sesuai dengan *input* model. Penetapan *batch size* berfungsi menentukan jumlah gambar yang akan diproses dalam setiap iterasi, sementara pengacakan (*shuffle*) bertujuan untuk mengacak urutan gambar dalam dataset sebelum model dilatih.

Pengubahan ukuran gambar mengikuti ukuran *input* gambar pada model yang digunakan. Pada penelitian ini digunakan arsitektur model ResNet152V2 yang memiliki ukuran *input* gambar sebesar 224×224 piksel. Penerapan *image resizing* akan mengubah ukuran seluruh gambar pada dataset menjadi ukuran yang dibutuhkan, dilakukan dengan parameter *target\_size*.

Penentuan *batch size* dilakukan dengan parameter *batch\_size* yang diatur menggunakan nilai 64. Nilai ini berarti model akan mengambil 64 gambar dalam satu iterasi. Nilai *batch size* akan memengaruhi durasi pelatihan dan kemampuan model dalam melakukan tugasnya.

Pengacakan gambar (*shuffle*) dilakukan dengan mengatur parameter *shuffle* sebagai ‘*True*’ untuk melakukan pengacakan dan ‘*False*’ jika tanpa pengacakan. Pengacakan adalah proses mengacak urutan data sebelum pelatihan model, dengan tujuan agar model tidak menghafal urutan data dan mampu belajar secara lebih umum untuk menghasilkan prediksi yang lebih akurat.

**5.4. Implementasi Pembagian Data**

Pembagian dataset dilakukan dengan membagi dataset menjadi tiga bagian, yaitu data pelatihan, data validasi, dan data uji. Data pelatihan dan data validasi akan digunakan dalam proses pelatihan model, sedangkan data uji digunakan untuk mengevaluasi kinerja model. Pembagian dataset dilakukan dengan acak dengan rasio tertentu antara data pelatihan, validasi, dan pengujian. Dataset dibagi menjadi data pelatihan dan data uji dengan rasio 5:1, kemudian data pelatihan dibagi lagi menjadi dua bagian yaitu data pelatihan dan data validasi dengan rasio 3:1. Rasio pembagian ini memastikan model mendapatkan cukup data untuk melakukan pelatihan dan evaluasi (validasi), serta memiliki data terpisah yang belum dilihat sebelumnya untuk tahap pengujian. Hasil pembagian data dapat dilihat pada tabel 5.1 berikut

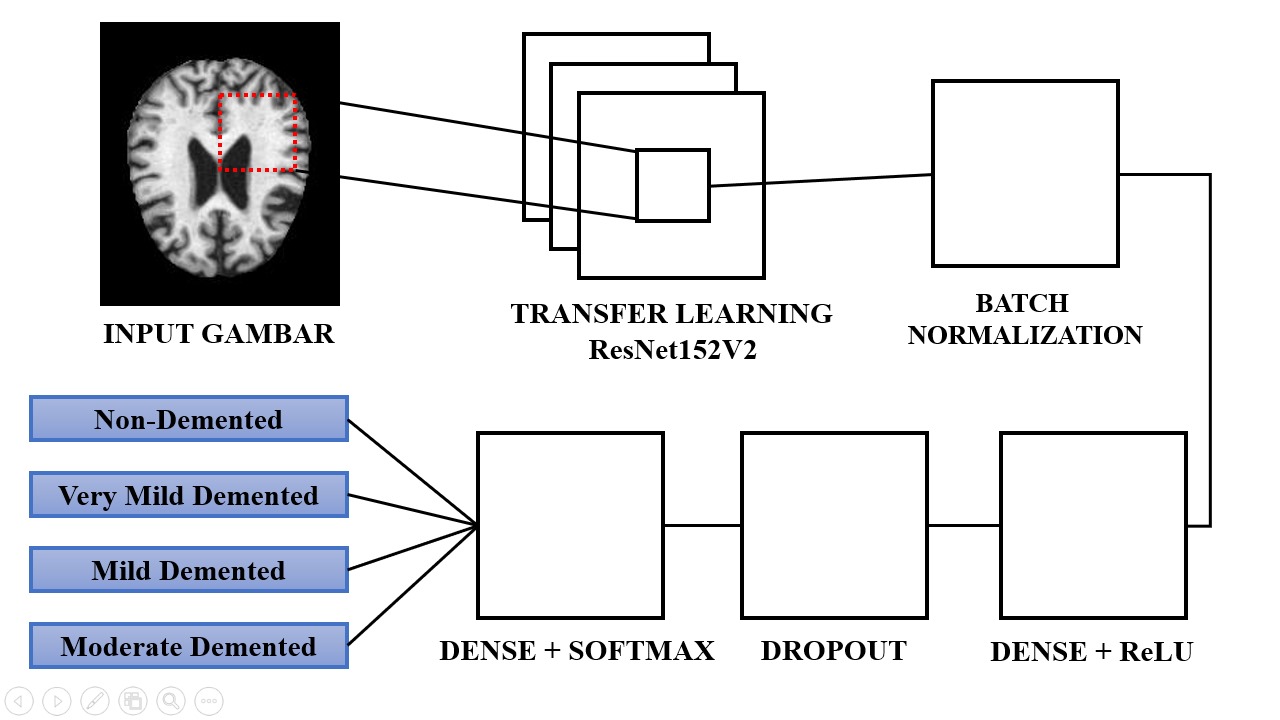
Tabel 5.1 Pembagian dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Kategori | Data Pelatihan | Data Validasi | Data Uji | Jumlah |
|  |
| 1 | Non-Demented | 7259 | 2341 | 3200 | 12800 |  |
| 2 | Very Mild Demented | 6695 | 2265 | 2240 | 11200 |  |
| 3 | Mild Demented | 6669 | 2291 | 896 | 9856 |  |
| 4 | Moderate Demented | 4865 | 1599 | 64 | 6528 |  |
| Total | | 25488 | 8496 | 6400 | 40384 |  |

**5.5. Hasil Pemodelan**

**5.5.1. Pengembangan Model**

Model dilatih dengan menggunakan metode *transfer learning* ResNet152V2 dengan tambahan lapisan *batch normalization*, *dense* dengan fungsi aktivasi ReLU, *dropout* dengan nilai sebesar 0,3, dan *dense* dengan fungsi aktivasi *softmax* pada lapisan akhir. Sebagai arsitektur dasar model, ResNet152V2 dikonfigurasi dengan beberapa parameter, yaitu include\_top = ‘False’; weights = ‘imagenet’; input\_shape = ‘IMG\_SHAPE’; dan pooling = ‘max’. Parameter *include\_top* yang dikonfigurasi *False* ini berarti lapisan *fully-connected* (*dense*) terakhir pada arsitektur model tidak disertakan dan hanya mneggunakan dasar *convolutional* modelnya saja. Parameter *weights* yang dikonfigurasi imagenet berarti bobot yang digunakan pada model adalah bobot hasil pelatihan dari dataset ImageNet. Parameter *input\_shape* dikonfigurasi berdasarkan dimensi tinggi, lebar, dan jumlah kanal dari *input* gambar model yaitu (224, 224, 3). Sedangkan parameter *pooling* yang dikonfigurasi *max* berarti model akan menggunakan *max pooling* setelah lapisan konvolusi dimana hanya nilai maksimal dari setiap *feature map* yang akan dipertahankan.



Gambar 5.5 Arsitektur model

Penggunaan lapisan *batch normalization* berperan penting dalam menstabilkan serta mempercepat proses pelatihan model dengan menormalkan data dari setiap *batch* menggunakan perhitungan rata-rata dan variansi. Proses ini membantu menyelaraskan data di setiap lapisan, mengurangi risiko masalah seperti perubahan nilai yang terlalu besar atau kecil, sehingga model dapat belajar lebih cepat dan efisien. Lapisan berikutnya adalah lapisan *dense* dengan fungsi aktivasi ReLU, yang berfungsi untuk mengubah nilai input menjadi positif. Aktivasi ini membantu model dalam mempelajari hubungan kompleks pada data, serta meningkatkan performa dan kecepatan pelatihan. Selain itu, lapisan ini juga menggunakan regularisasi untuk mencegah terjadinya *overfitting*. Selanjutnya, digunakan lapisan *dropout* yang berfungsi untuk secara acak menonaktifkan sejumlah neuron pada setiap iterasi pelatihan. Nilai 0,3 yang ditetapkan pada lapisan ini menunjukkan bahwa 30% unit pada lapisan tersebut akan diabaikan, sedangkan sisanya digunakan selama pelatihan. Pada lapisan terakhir, digunakan lapisan *dense* dengan aktivasi *softmax*, yang bertugas mengonversi nilai probabilitas setiap kelas ke dalam rentang 0 − 1. Probabilitas tertinggi di antara kelas-kelas tersebut dianggap sebagai hasil klasifikasi.

Setelah model selesai dibangun, langkah berikutnya adalah mendefinisikan optimizer, mengompilasi model, dan mendefinisikan *callbacks*. Optimizer yang digunakan adalah Adamax, salah satu varian dari Adam yang lebih stabil dan menunjukkan performa baik, dengan *learning rate* sebesar 0,001. Selanjutnya, dilakukan kompilasi model, yang merupakan tahap penting sebelum pelatihan dimulai. Parameter yang digunakan dalam kompilasi model meliputi *optimizer*, *loss*, dan *metrics*. Parameter *loss* berfungsi untuk mengukur seberapa baik kinerja model selama pelatihan dengan menghitung kerugian menggunakan fungsi *categorical crossentropy*. Sementara itu, *metrics* digunakan untuk mengevaluasi performa model selama pelatihan dan pengujian dengan menghitung jumlah prediksi yang benar dari total prediksi yang dihasilkan oleh model. Langkah terakhir sebelum pelatihan adalah mendefinisikan *callbacks*, yang akan menghentikan pelatihan secara otomatis ketika kondisi tertentu tercapai. Dalam hal ini, digunakan *callbacks* untuk menghentikan pelatihan jika tidak ada peningkatan akurasi pada dataset validasi selama 5 *epoch* atau iterasi berturut-turut.

**5.5.2. Pelatihan Model**

Pada tahap pelatihan model, digunakna beberapa parameter seperti dataset pelatihan, *epochs*, dataset validasi, *batch\_size*, dan *callbacks*. Dataset pelatihan merupakan dataset yang digunakan model untuk proses pelatihan. Parameter *epochs* adalah berapa kali seluruh dataset dilalui selama pelatihan, dalam pelatihan ini ditetapkan maksimal sebanyak 50 *epochs*. Dataset validasi merupakan dataset hasil pemrosesan sebelumnya yang akan digunakan untuk validasi model selama pelatihan. Parameter *batch\_size* yang digunakan sebesar 48. Sedangkan *callbacks* yang digunakan sesuai dengan yang telah didefinisikan pada tahap pengembangan model.

Tabel 5.2 Data akurasi dan loss pada tiap epoch pelatihan

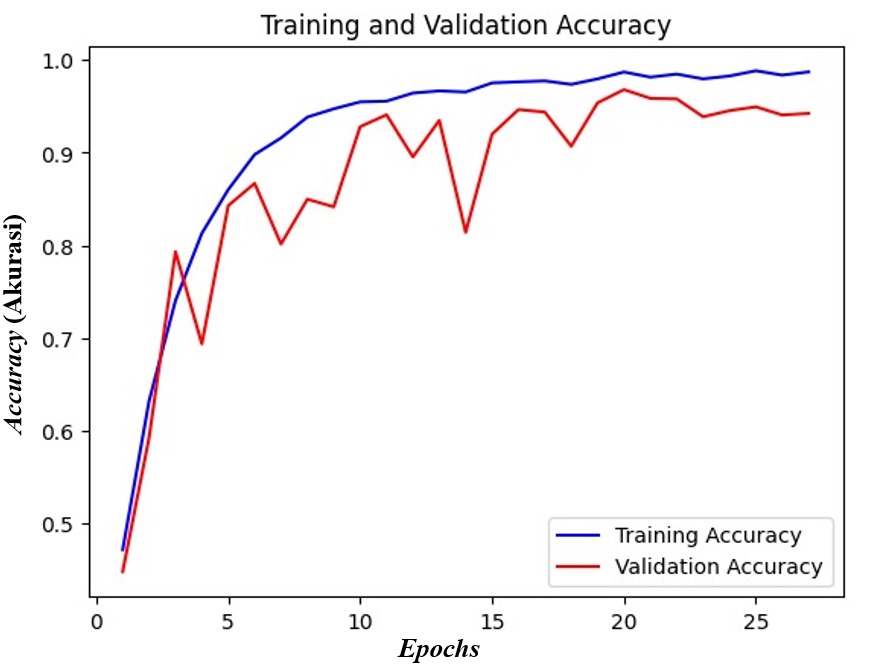
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Epoch* | *Accuracy* | *Validation Accuracy* | *Loss* | *Validation Loss* |
| 1 | 0,4743 | 0,4512 | 2,4074 | 2,8739 |
| 2 | 0,6537 | 0,6534 | 1,9573 | 2,1827 |
| 3 | 0,7619 | 0,7931 | 1,6825 | 1,6212 |
| 4 | 0,8232 | 0,7027 | 1,4553 | 2,1931 |
| 5 | 0,8625 | 0,8416 | 1,2743 | 1,2879 |
| 6 | 0,9038 | 0,8641 | 1,1324 | 1,1983 |
| 7 | 0,9217 | 0,8021 | 0,9945 | 1,3345 |
| 8 | 0,9342 | 0,8485 | 0,8531 | 1,1594 |
| 9 | 0,9448 | 0,8356 | 0,7549 | 1,2235 |
| 10 | 0,9523 | 0,9017 | 0,6571 | 0,7342 |
| 11 | 0,9497 | 0,9258 | 0,621 | 0,6521 |
| 12 | 0,9615 | 0,8918 | 0,5432 | 0,8438 |
| 13 | 0,9592 | 0,9234 | 0,5012 | 0,5892 |
| 14 | 0,9703 | 0,8053 | 0,4589 | 1,029 |
| 15 | 0,9728 | 0,9037 | 0,4012 | 0,6243 |
| 16 | 0,9691 | 0,9238 | 0,3369 | 0,3987 |
| 17 | 0,9773 | 0,9199 | 0,3041 | 0,3847 |
| 18 | 0,9786 | 0,9001 | 0,2573 | 0,5021 |
| 19 | 0,9754 | 0,934 | 0,221 | 0,3572 |
| 20 | 0,9878 | 0,9621 | 0,2154 | 0,2674 |
| 21 | 0,9805 | 0,9582 | 0,2221 | 0,2689 |
| 22 | 0,9836 | 0,9581 | 0,1926 | 0,2461 |
| 23 | 0,9774 | 0,9388 | 0,1874 | 0,3237 |
| 24 | 0,9808 | 0,9454 | 0,1743 | 0,2649 |
| 25 | 0,9879 | 0,9495 | 0,1314 | 0,2798 |
| 26 | 0,9795 | 0,9407 | 0,1584 | 0,2696 |
| 27 | 0,986 | 0,9424 | 0,1179 | 0,2898 |

Berdasarkan tabel 5.4, dapat diamati bahwa model mencapai akurasi terbaik pelatihan pada *epoch* ke-25 dengan nilai 0,9879 atau 98,8%. Pada *epoch* ke-27 fungsi *callbacks* aktif dikarenakan tidak terjadi penurunan nilai kerugian (*loss*) pada dataset validasi selama 5 *epoch*. Nilai kerugian (*loss*) pelatihan yang diperoleh sudah cukup kecil yaitu 0,1314. Sedangkan pada *epoch* yang sama, diperoleh akurasi validasi sebesar 0,9495 atau 95%, dengan nilai *loss* validasi sebesar 0,2798. Nilai akurasi antara dataset pelatihan dan validasi yang berada di bawah 5% menunjukkan bahwa model tidak mengalami *overfitting*.

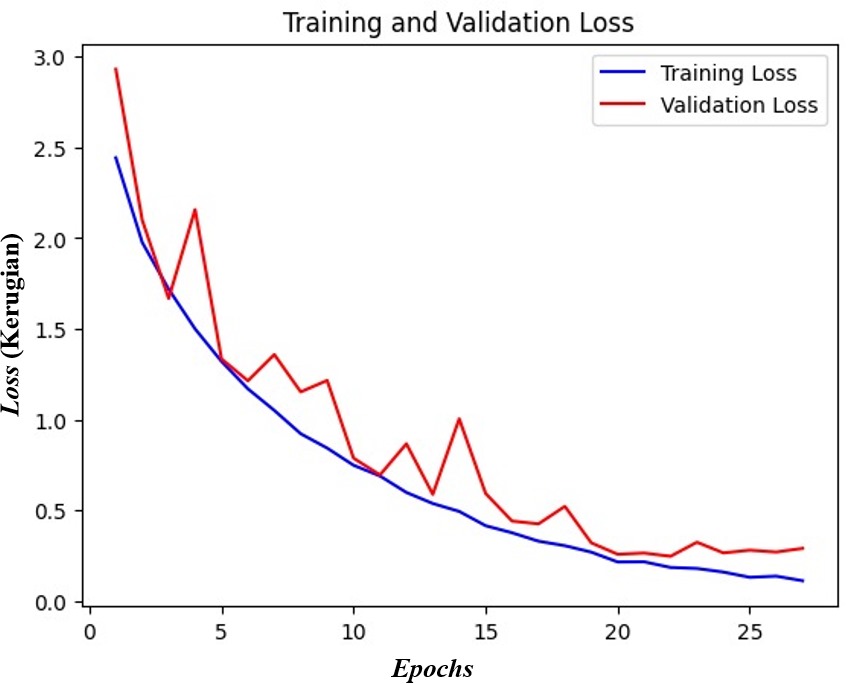
Pada umumnya, ketika nilai akurasi meningkat, *loss* cenderung menurun. Namun, keduanya tidak selalu bergerak seiringan secara sempurna seperti yang terlihat pada beberapa *epoch* dalam tabel di atas. Hal ini bisa terjadi ketika model mulai mendekati titik konvergensi. Selain itu, metrik yang digunakan dalam mengukur akurasi dan *loss* juga dapat menyebabkan perilaku ini. Akurasi hanya melihat prediksi benar atau salah, sedangkan *loss* menghitung seberapa jauh prediksi dari nilai sebenarnya, sehingga nilai *loss* bisa berfluktuasi lebih dibandingkan akurasi.

**5.5.3. Evaluasi Model**

Berdasarkan hasil pelatihan yang ditunjukkan pada tabel 5.2 akan dilakukan visualisasi hasil akurasi dan kerugian (*loss*) pada proses pelatihan dan validasi dalam sebuah grafik. Visualiasi ini membantu dalam membandingkan hasil dari pelatihan dan validasi. Grafik visualisasi dapat dilihat pada gambar berikut.

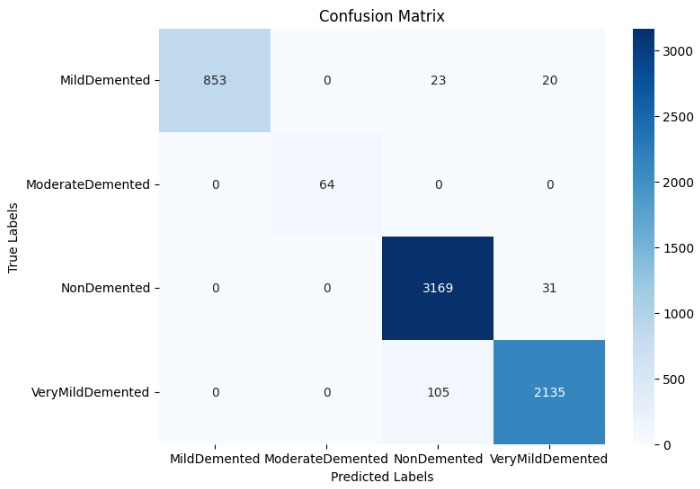


Gambar 5.6 Grafik akurasi pelatihan dan validasi



Gambar 5.7 Grafik kerugian (loss) pelatihan dan validasi

Berdasarkan grafik di atas, dapat dikatakan bahwa akurasi pelatihan mengalami peningkatan yang lebih stabil dibandingkan validasi yang cenderung fluktuatif. Grafik kerugian menunjukkan pola yang baik dengan nilai kerugian yang semakin kecil dengan dataset pelatihan memiliki pola yang lebih stabil. Perbedaan kestabilan antara proses pelatihan dan validasi ini wajar terjadi karena perbedaan distribusi data antara dataset pelatihan dan validasi. Selain itu, ukuran dataset pelatihan yang lebih besar memungkinkan model lebih mudah mempelajari pola di dalamnya, sementara pada data validasi, model mungkin menunjukkan hasil yang lebih fluktuatif karena variasi yang lebih tinggi atau kurangnya representasi pola yang serupa. Namun perbedaan tersebut dapat berisiko menyebabkan *overfitting*. Andrew Ng (2011) dalam kursus daring yang dipublikasikannya menjelaskan bahwa selisih 5 − 10% antara akurasi pelatihan dan validasi merupakan indikasi terjadinya *overfitting*. Perbedaan akurasi yang dicapai dalam penelitian ini masih di bawah 5%, sehingga belum menunjukkan tanda-tanda *overfitting*.



Gambar 5.8 Confusion matrix hasil pengujian

Berdasarkan *confusion matrix* di atas dapat dilihat bahwa model dapat menebak 6221 gambar dari total 6400 pada dataset pengujian yang belum pernah dilihat sebelumnya. Model berhasil menebak benar 3169 gambar kondisi Non-Demented, 2135 gambar kondisi Very Mild Demented, 853 gambar kondisi Mild Demented, dan 64 gambar kondisi Moderate Demented. *Confusion matrix* menunjukkan bahwa model mengalami sedikit kesulitan dalam membedakan sampel dari beberapa kelas yang memiliki karakteristik serupa, seperti *Non-Demented*, *Very Mild Demented*, dan *Mild Demented*. Secara umum model menunjukkan kinerja yang baik dengan kemampuan generalisasi yang cukup baik terhadap data baru dengan akurasi yang ditunjukkan pada tabel 5.3 berikut.

Tabel 5.3 Metrik evaluasi model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Precision* | *Recall* | *F1-score* | *Support* |
| *Non-Demented* | 0,96 | 0,99 | 0,98 | 3200 |
| *Very Mild Demented* | 0,98 | 0,95 | 0,96 | 2240 |
| *Mild Demented* | 1,00 | 0,95 | 0,98 | 896 |
| *Moderate Demented* | 1,00 | 1,00 | 1,00 | 64 |
|  |  |  |  |  |
| *Accuracy* | | | 0,97 | 6400 |
| *Macro Average* | 0,98 | 0,97 | 0,98 | 6400 |
| *Weighted Average* | 0,97 | 0,97 | 0,97 | 6400 |

Berdasarkan tabel 5.3, dapat dilihat bahwa model menghasilkan akurasi sebesar 97% pada dataset pengujian. Pada kategori *Non-Demented* mencapai nilai *precision* sebesar 96%, *recall* sebesar 99%, *f1-score* sebesar 98% dengan jumlah data sebanyak 3200 gambar. Pada kategori *Very Mild Demented* mencapai nilai *precision* sebesar 98%, *recall* sebesar 95%, *f1-score* sebesar 96% dengan jumlah data sebanyak 2240 gambar. Pada kategori *Mild Demented* mencapai nilai *precision* sebesar 100%, *recall* sebesar 95%, *f1-score* sebesar 98% dengan jumlah data sebanyak 896 gambar. Pada kategori *Moderate Demented* mencapai nilai *precision* sebesar 100%, *recall* sebesar 100%, *f1-score* sebesar 100% dengan jumlah data sebanyak 64 gambar.

Metrik *macro average* yang merupakan rata-rata metrik dari semua kategori menunjukkan nilai *precision* sebesar 98%, *recall* sebesar 97%, dan *f1-score* sebesar 98%. *Weighted average* menghitung rata-rata metrik dengan bobto yang sebanding dengan jumlah sampel pada tiap kategori menunjukkan nilai *precision* sebesar 97%, *recall* sebesar 97%, dan *f1-score* sebesar 97%. Géron (2022) dalam bukunya menyatakan suatu model *multi class classification* yang baik memiliki metrik evaluasi dengan nilai ambang batas sebesar 80% untuk *precision* dan *recall*, serta 60% untuk *f1-score*.

Secara keseluruhan, performa model dalam proses pelatihan, validasi, dan pengujian menunjukkan hasil yang sangat baik. Selisih antara akurasi pelatihan dan validasi dalam penelitian ini masih di bawah ambang batas yang mengindikasikan *overfitting*. Model juga dapat menggeneralisasi data baru dengan baik, ditunjukkan oleh metrik evaluasi pada dataset pengujian. Perbedaan antara akurasi pelatihan dan validasi dalam penelitian ini masih di bawah ambang batas yang mengindikasikan *overfitting*.

**5.6. Perbandingan Hasil Penelitian dengan Penelitian Terdahulu**

Pada penelitian yang dilakukan oleh Yildirim dan Cinar (2020), dilakukan pengembangan model deteksi dengan empat kategori yang sama dengan penelitian ini. Penelitian menggunakan model ResNet50 dalam dua versi yaitu model murni dan model dengan penambahan lapisan tambahan. Akurasi pengujian yang dicapai sebesar 70% untuk model ResNet50 murni dan 90% untuk model dengan lapisan tambahan.

Pada penelitian oleh Buvaneswari dan Gayathri (2021), dilakukan pengembangan model untuk mendeteksi tiga kondisi yaitu normal, kelainan kognitif ringan, dan *Alzheimer*. Dataset yang digunakan merupakan dataset kecil yang hanya berisi 240 gambar. Penelitian menggunakan model dengan dasar arsitektur ResNet101. Akurasi yang dicapai model adalah sebesar 96,3%.

Pada penelitian oleh Ullah dan Jamjoom (2023) menggunakan model yang dibangun sendiri untuk mendeteksi empat kondisi penyakit *Alzheimer* yang sama dengan penelitian ini. Dataset yang digunakan berisi total 6400 gambar. Model mencapai tingkat akurasi yang baik yaitu sebesar 99,38%.

Penelitian dengan objek berbeda dilakukan oleh Aref dan Kareem (2021) dengan model deteksi infeksi Covid-19 dan penelitian oleh Venkataramanan dkk. (2019) dengan model deteksi penyakit pada tanaman. Kedua penelitian ini mencapai akurasi yang baik menggunakan model dengan dasar arsitektur ResNet yang berbeda versi. Pada penelitian Aref dan Kareem (2021) diperoleh akurasi sebesar 96,1% dan 99,5% pada dua dataset berbeda dengan model ResNet50. Sedangkan pada penelitian oleh Venkataramanan dkk. (2019) model dengan dasar ResNet18 memperoleh akurasi sebesar 96%.

Berdasarkan hasil yang diperoleh oleh peneliti, penggunaan *transfer learning* dengan dasar model ResNet152V2 menghasilkan akurasi yang baik dalam mengklasifikasi gambar dalam jumlah yang besar sekalipun. Pengmbangan dalam penelitian dilakukan dengan menggunakan dataset empat tahapan penyakit *Alzheimer* yang lebih besar dan penggunaan model ResNet152V2. Empat kategori tahapan penyakit *Alzheimer* yang dimaksud terdiri dari *Non-Demented*, *Very Mild Demented*, *Mild Demented*, dan *Moderate Demented*. Dataset juga telah melalui pra-proses disesuaikan dengan kebutuhan model. Pengembangan lain adalah penggunaan model dasar ResNet152V2 dengan lapisan tambahan, pemilihan *optimizer* Adamax, pengaturan *learning rate* sebesar 0,001, dan beberapa parameter lain. Dengan langkah-langkah yang telah dilaksanakan, diperoleh model dengan selisih akurasi pelatihan dan validasi yang kecil, serta akurasi pengujian mencapai 97%.

Tabel 5.4 Perbandingan dengan Penelitian Terdahulu

|  |  |  |
| --- | --- | --- |
| **Deteksi** | **Peneliti** | **Hasil** |
| Deteksi empat tahapan awal penyakit *Alzheimer* yaitu *Non-Demented*, *Very Mild Demented*, *Mild Demented*, dan *Moderate Demented*. | Yildirim & Cinar. (2020) | * Model yang dibangun dengan arsitektur ResNet50 murni mendapatkan akurasi sebesar 78%. * Model ResNet50 dengan lapisan tambahan mendapat akurasi yang lebih baik, yaitu 90%. |
| Deteksi tiga kondisi pasien yaitu Normal, Kelainan kognitif ringan, dan *Alzheimer*. | Buvaneswari & Gayathri. (2021) | Model yang dikembangkan menunjukkan akurasi klasifikasi sebesar 96,3%. |
| Deteksi empat tahapan awal penyakit *Alzheimer* yaitu *Non Demented*, *Very Mild Demented*, *Mild Demented*, dan *Moderate Demented*. | Ullah & Jamjoom. (2023) | Model mencapai tingkat akurasi yang baik yaitu sebesar 99,38%. |
| Identifikasi penyakit pada tanaman dengan mengamati daunnya. | Venkataramanan dkk. (2019) | Model mencapai nilai akurasi sebesar 96%. |
| Membandingkan beberapa algoritma untuk deteksi infeksi Covid-19 dengan gambar X-Ray dada pasien | Aref & Kareem. (2021) | * Model yang mencapai performa klasifikasi tertinggi adalah ResNet50 dengan akurasi sebesar 96,1% pada dataset 1 dan 99,5% pada dataset 2. |
| Deteksi empat tahapan awal penyakit *Alzheimer* yaitu *Non-Demented*, *Very Mild Demented*, *Mild Demented*, dan *Moderate Demented*. | Rui Costa Raka Milanisti (Penelitian ini) | * Model dengan dasar ResNet152V2 mencapai akurasi 98,8% pada pelatihan dan 97% pada pengujian. |

Aref, N., & Kareem, H. (2021). Detection of Covid-19 Based on Chest Medical Imaging and Artificial Intelligent Techniques: A Review. *Iraqi Journal for Electrical and Electronic Engineering*, *17*(2), 176–182. https://doi.org/10.37917/ijeee.17.2.19

Buvaneswari, P. R., & Gayathri, R. (2021). Deep Learning-Based Segmentation in Classification of Alzheimer’s Disease. *Arabian Journal for Science and Engineering*, *46*(6), 5373–5383. https://doi.org/10.1007/s13369-020-05193-z

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (3 ed.). O’Reilly Media, Inc.

Ng, A. (2011). *Supervised Machine Learning: Regression and Classification*. Coursera. https://www.coursera.org/learn/machine-learning

Ullah, Z., & Jamjoom, M. (2023). A Deep Learning for Alzheimer’s Stages Detection Using Brain Images. *Computers, Materials and Continua*, *74*(1), 1457–1473. https://doi.org/10.32604/cmc.2023.032752

Uraninjo. (2022, September 21). *Uraninjo Dataset*. Kaggle.com. https://www.kaggle.com/datasets/uraninjo/augmented-alzheimer-mri-dataset-v2

Venkataramanan, A., Kumar, D., Honakeri, P., & Agarwal, P. (2019). *Plant Disease Detection and Classification Using Deep Neural Networks*.

Yildirim, M., & Cinar, A. (2020). Classification of Alzheimer’s disease MRI images with CNN based hybrid method. *Ingenierie des Systemes d’Information*, *25*(4), 413–418. https://doi.org/10.18280/isi.250402