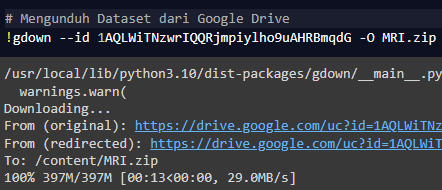
**Hasil Penelitian dan Pembahasan**

**5.1. Persiapan Dataset**

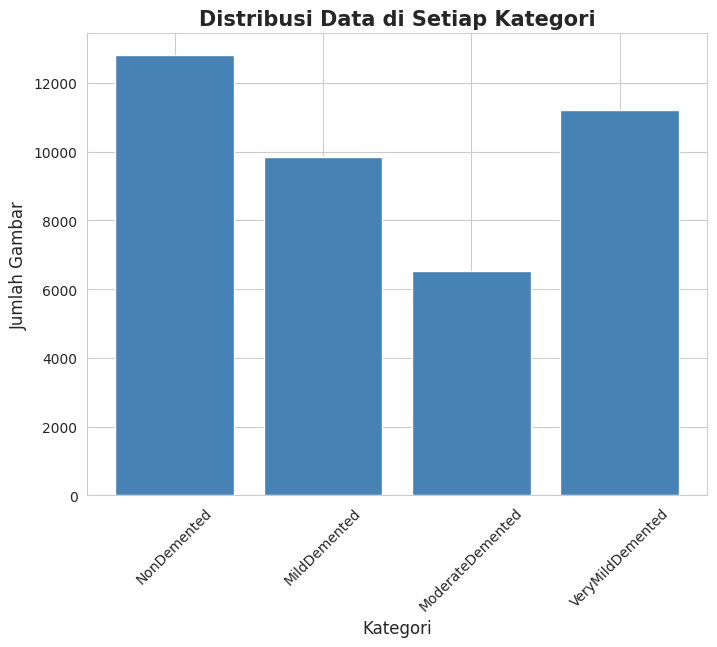
Dalam penelitian ini, digunakan dataset berupa data sekunder yang diambil dari sumber open source Kaggle.com. Dataset ini dikumpulkan, dilakukan augmentasi, dan dipublikasikan oleh pengguna bernama Uraninjo pada tahun 2022. Terdapat total 40.384 gambar MRI otak pasien, yang terbagi ke dalam empat kategori 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.



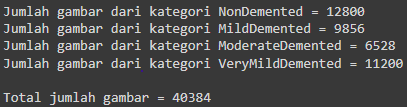
Gambar 5.1 Pengunduhan dataset

**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.

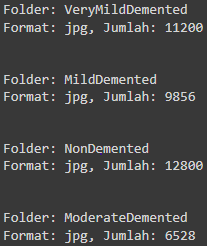


Gambar 5.2 Grafik distribusi dataset



Gambar 5.3 Jumlah gambar setiap kategori dan keseluruhan

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 prose pelatihan model, sedangkan data uji digunakan untuk mengevaluasi kinerja model. 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. Hasil pembagian data dapat dilihat pada tabel 5.1 berikut

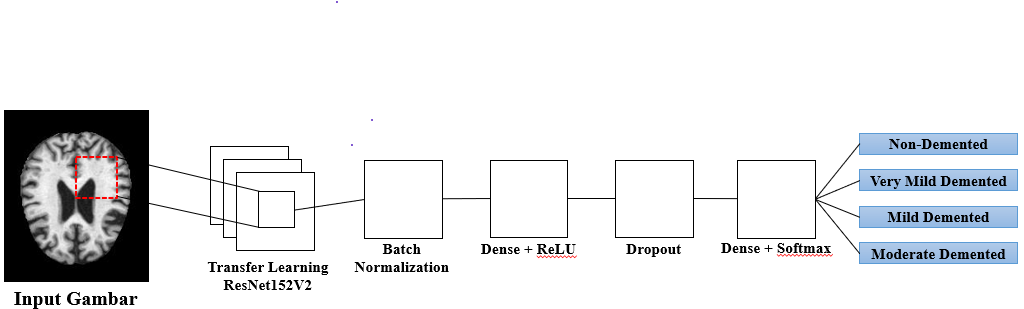
Tabel 5.3 Pembagian dataset

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 X, 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. 6 Arsitektur model

Penggunaan lapisan *batch normalization* berperan penting dalam menstabilkan serta mempercepat proses pelatihan model. 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 X yang ditetapkan pada lapisan ini menunjukkan bahwa X% 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 X *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.

[Gambar]

Berdasarkan gambar 5.7, dapat diamati bahwa model mencapai akurasi terbaik pelatihan pada *epoch* ke-X dengan nilai X atau X%. Pada *epoch* ke-X fungsi *callbacks* aktif, sehingga jika tidak terjadi peningkatan akurasi pada dataset validasi dalam X epoch berikutnya proses pelatihan akan berhenti. Nilai kerugian (*loss*) pelatihan yang diperoleh sudah cukup kecil yaitu X. Sedangkan pada *epoch* yang sama, diperoleh akurasi validasi sebesar X atau X%, dengan nilai *loss* validasi sebesar X. [……..].

**5.5.3. Evaluasi Model**

Berdasarkan hasil pelatihan yang ditunjukkan pada gambar 5.7 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 acc]

[Gambar loss]

Berdasarkan grafik di atas, dapat dikatakan bahwa pelaitihan terus mengalami peningkatan akurasi dan validasi ……… Grafik kerugian menunjukkan pola yang baik dengan nilai kerugian yang semakin kecil. ……………. Hasil yang berbeda antara pelatihan dan validasi ini wajar terjadi karena perbedaan distribusi data antara dataset pelatihan dan validasi, ukuran dataset pelatihan yang lebih besar sehingga model mempelajari pola di dalamnya. Namun perbedaan tersebut dapat mengarah pada *overfitting*. Andrew Ng (2011) dalam kursus daring yang dipublikasikannya menjelaskan bahwa selisih 5 − 10% antara akurasi pelatihan dan validasi mengindikasikan *overfitting*.

[Gambar conf]

Berdasarkan *confusion matrix* di atas dapat dilihat bahwa model dapat menebak X gambar dari total X gambar pada dataset engujian yang belum pernah dilihat sebelumnya. Model dapat menebak X gambar kondisi Non-Demented, X gambar kondisi Very Mild Demented, X gambar kondisi Mild Demented, dan X gambar kondisi Moderate Demented. Model menunjukkan kinerja yang baik dengan kemampuan generalisasi yang cuku baik terhadap data baru dengan akurasi yang ditunjukkan pada tabel 5.4 berikut.

[Tabel Report]

Berdasarkan tabel 5.4, dapat dilihat bahwa model menghasilkan akurasi sebesar X% pada dataset pengujian. Pada kategori Non-Demented mencapai nilai precision sebesar X%, recall sebesar X%, f1-score sebesar X% dengan jumlah data sebanyak 18 gambar. Pada kategori Very Mild Demented mencapai nilai precision sebesar X%, recall sebesar X%, f1-score sebesar X% dengan jumlah data sebanyak 18 gambar. Pada kategori Mild Demented mencapai nilai precision sebesar X%, recall sebesar X%, f1-score sebesar X% dengan jumlah data sebanyak 18 gambar. Pada kategori Moderate Demented mencapai nilai precision sebesar X%, recall sebesar X%, f1-score sebesar X% dengan jumlah data sebanyak 18 gambar.

[….]

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

Pada penelitian yang dilakukan oleh.