**Metode Penelitian**

**4.1. Alat dan Bahan Penelitian**

**4.1.1. Alat**

Pada penelitian ini digunakan peralatan sebagai berikut:

1. Perangkat Komputer

Model : Lenovo IdeaPad Flex 5 14IIL05

CPU : Intel® Core™ i3-1005G1

GPU : Intel UHD Graphics

RAM : 8 GB

Penyimpanan : 512 GB SSD

Sistem Operasi : Microsoft Windows 11

1. *Platform* dan Perangkat Lunak

Bahasa Pemrograman : Python 3.10.12

IDE : Google Colab

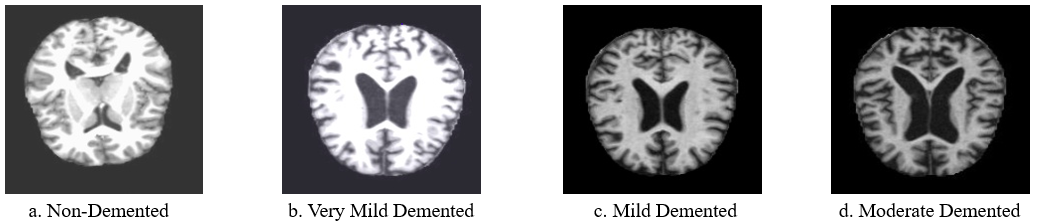
*Library Machine Learning* : Tensorflow dan Keras

*Library* Augmentasi Data : ImageDataGenerator

*Library* Evaluasi Model : Matplotlib

**4.1.2. Bahan**

Penelitian ini menggunakan dataset yang berisi citra MRI otak yang dikelompokkan ke dalam 4 kategori kondisi pasien, yaitu *non demented*, *very mild demented*, *mild demented*, dan *moderate demented*. Dataset yang digunakan dalam penelitian ini adalah dataset open source yang dipublikasikan oleh Uraninjo pada tahun 2022 di situs Kaggle.com, yang merupakan hasil augmentasi dari data yang diunggah oleh Dubey pada tahun 2019. Data tersebut berasal dari publikasi oleh Open Access Series of Imaging Studies (OASIS). Contoh data gambar MRI pada masing-masing kategori dapat dilihat pada gambar 4.1.



Gambar 4.1 Dataset MRI otak

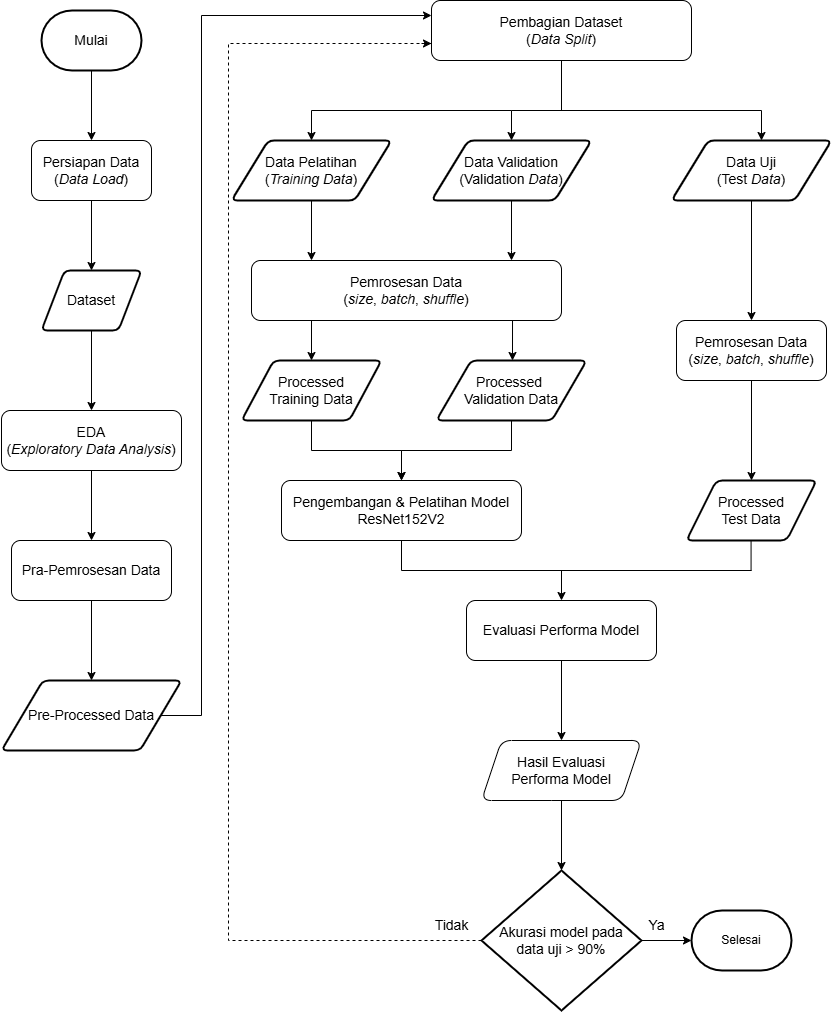
Dataset berisi total 40.384 data gambar. Distribusi data dari tiap kategori dapat dilihat pada tabel 4.1.

Tabel 4.1 Distribusi data

|  |  |  |
| --- | --- | --- |
| No | Kategori | Jumlah Data |
| 1 | Non-Demented | 12.800 |
| 2 | Very Mild Demented | 11.200 |
| 3 | Mild Demented | 9.856 |
| 4 | Moderate Demented | 6.528 |
| Total | | 40.384 |

**4.2. Prosedur Penelitian**

Penelitian ini secara garis besar dibagi dalam enam tahapan. Tahap pertama melibatkan persiapan data, yang dilakukan dengan mencari dan mengumpulkan data sekunder terlebih dahulu, kemudian memuatnya ke dalam proyek (*data load*) untuk proses lebih lanjut. Tahap kedua adalah melakukan Exploratory Data Analysis (EDA) untuk menganalisis dataset dengan menampilkan distribusi data pada setiap kelas serta mengidentifikasi ekstensi file gambar yang terdapat dalam dataset. Tahap ketiga melibatkan pra-pemrosesan data dengan mengatur ukuran gambar, *batch*, melakukan pengacakan (*shuffle*), dan memastikan keseragaman ekstensi *file* agar gambar siap digunakan dalam model. Tahap keempat adalah membagi data (*data split*) yang sudah diproses menjadi data pelatihan, data validasi, dan data uji dengan rasio tertentu. Tahap kelima adalah pengembangan dan pelatihan model yang dilakukan dengan mengembangkan model dengan metode *transfer learning* menggunakan arsitektur ResNet152V2 dan melatihnya dengan data pelatihan. Tahap terakhir adalah mengevaluasi model dengan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Melalui metrik evaluasi ini dapat ditentukan kemampuan model dalam mengklasifikasi gambar 4 tahapan penyakit Alzheimer. Diagam alir tahapan penelitian dapat dilihat pada gambar 4.2.



Gambar 4.2 Diagram alir penelitian

**4.2.1. Persiapan Data**

Dataset yang digunakan adalah dataset sekunder dari Kaggle.com yang dipublikasikan oleh pengguna dengan nama Uraninjo (2022). Dataset memiliki total 40.384 data gambar yang terdiri atas 4 kategori/tahapan penyakit Alzheimer yaitu *Non-Demented*, *Very Mild Demented*, *Mild Demented*, dan *Moderate Demented*. Pada tahap ini, dataset diunduh dan disimpan di Google Drive sebagai basis data untuk pembuatan model. Untuk menggunakannya, dataset dimuat ke dalam lingkungan proyek di Google Colab.

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

Data yang telah dimuat ke dalam proyek lalu dianalisis untuk mengetahui struktur dan polanya. Pada dataset ini dilakukan analisis untuk distribusi data dan keragaman ekstensi file. Distribusi data dapat dilihat dengan melakukan visualisasi terhadap jumlah data pada masing-masing kategori. Selain itu, dianalisa juga keragaman ekstensi file pada dataset dengan menampilkan seluruh ekstensi file gambar.

**4.2.3. Pra-pemrosesan Data**

Pada tahap ini, ketidaksesuaian dalam dataset akan diproses. Beberapa langkah yang akan dilakukan meliputi penyeragaman ekstensi *file* dan pemrosesan data, seperti pengubahan ukuran gambar, *batch*, dan pengacakan (shuffle). Penyeragaman ekstensi file dilakukan dengan menghapus gambar yang memiliki ekstensi selain jpg, jpeg, dan png. Pemrosesan data, termasuk pengubahan ukuran gambar, *batch*, dan pengacakan, akan dilakukan menggunakan ImageDataGenerator. Pengubahan ukuran gambar (*image resizing*) bertujuan untuk menyamakan ukuran gambar sesuai *input* model; *batch* adalah jumlah sampel yang diproses pada setiap iterasi; dan pengacakan (*shuffle*) menentukan apakah data akan diacak di setiap iterasi atau tidak. Meskipun pemrosesan ini dilakukan setelah tahap pembagian data, teknik ini tetap dianggap sebagai bagian dari pra-pemrosesan data.

**4.2.4. Pembagian Data**

Dataset telah dibagi menjadi data pelatihan dan data uji dengan rasio 5:1, memastikan bahwa sebagian besar data digunakan untuk pelatihan model, sementara sisanya dialokasikan untuk pengujian. Selanjutnya, data pelatihan yang telah dipisahkan kemudian dibagi lagi menjadi dua bagian: data pelatihan dan data validasi, dengan rasio 3:1. Pembagian ini bertujuan untuk memberikan set data yang cukup untuk melatih model secara efektif, sekaligus memungkinkan evaluasi performa model selama proses pelatihan melalui data validasi. Data pelatihan dan data validasi digunakan pada tahap pelatihan model. Sedangkan data uji digunakan untuk mengevaluasi model seetelah pelatihan dengan melakukan pengujian terhadap data yang belum pernah dilihat sebelumnya.

**4.2.5. Pengembangan Model**

Pada penelitian ini, digunakan metode pembelajaran mendalam (deep learning) yang berbasis CNN. Metode yang diterapkan adalah transfer learning dengan arsitektur model ResNet152V2 yang sebagian besar berisikan lapisan konvolusi dan mengandalkan konsep *skip connection* dan pemetaan identitas. Model ini kemudian dilengkapi dengan lapisan tambahan seperti *batch normalization*, *dropout*, dan *fully-connected* untuk menyesuaikan dataset.

*Input* yang digunakan untuk model adalah gambar yang memiliki tiga dimensi yaitu tinggi, lebar, dan kedalaman. *Input* akan masuk ke dalam serangkaian lapisan konvolusi yang diikuti oleh *max pooling*, dengan setiap rangkaian memiliki parameter yang berbeda. Setiap gambar yang melewati lapisan konvolusi dan *max pooling* akan mengalami reduksi dimensi dan resolusi gambar (*down sampling*).

Setelah melalui *base model*, gambar akan melewati beberapa lapisan tambahan yaitu *batch normalization*, *dropout*, dan *fully-connected* atau lapisan *dense*. *Batch normalization* berfungsi untuk mempercepat proses konvergensi dan meningkatkan stabilitas selama pelatihan. Sementara itu, *dropout* digunakan untuk mengurangi overfitting dengan menghapus sebagian unit (neuron) secara acak dalam lapisan selama fase pelatihan.Sementara itu, *fully-connected layer* berfungsi untuk menghubungkan setiap neuron dari lapisan sebelumnya dengan setiap neuron di lapisan selanjutnya. Hal ini memungkinkan model untuk mempelajari representasi fitur yang kompleks dari data. Di lapisan ini juga terjadi proses klasifikasi gambar sesuai dengan kelas yang ada, berdasarkan ekstraksi fitur yang dilakukan sebelumnya.

Penelitian ini menggunakan sejumlah parameter penting yang dipilih secara khusus untuk memaksimalkan kinerja model. Parameter tersebut meliputi *optimizer* dan nilai *learning rate*. *Optimizer* yang digunakan adalah Adamax, yang merupakan versi terbaru dari Adam. Nilai learning rate yang diterapkan adalah 0,001. Penggunaan *learning rate* berpengaruh terhadap akurasi model dan kecepatan pelatihan. Secara umum, nilai *learning rate* berkisar antara 0,1 hingga 0,0001. Semakin kecil *learning rate*, semakin cepat proses pelatihan, namun berisiko model tidak mencapai *global optimum*, sehingga akurasinya tidak optimal. Sebaliknya, *learning rate* yang terlalu besar akan memperlambat pelatihan, tetapi memastikan model mencapai akurasi terbaik.

**4.2.6. Metrik Evaluasi**

Metrik evaluasi digunakan untuk menilai kinerja model yang telah dilatih sebelumnya. Proses evaluasi dilakukan dengan memberikan tugas klasifikasi pada data uji yang belum pernah dilihat oleh model. Metrik evaluasi yang digunakan meliputi akurasi (*accuracy*), *precision*, *recall*, dan *f1-score*, yang dihitung berdasarkan *confusion matrix* dari *library* Matplotlib. Semakin banyak data uji yang diprediksi dengan benar, semakin baik performa model tersebut.