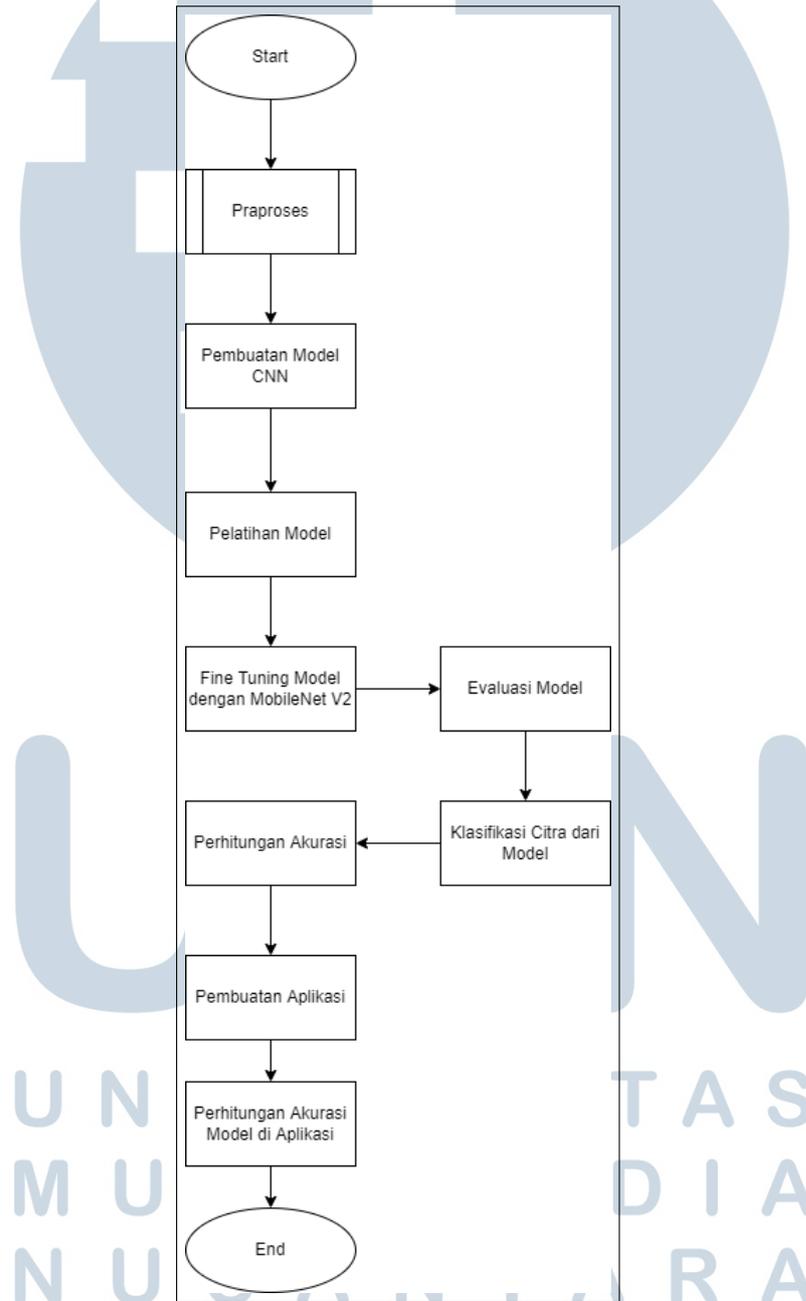


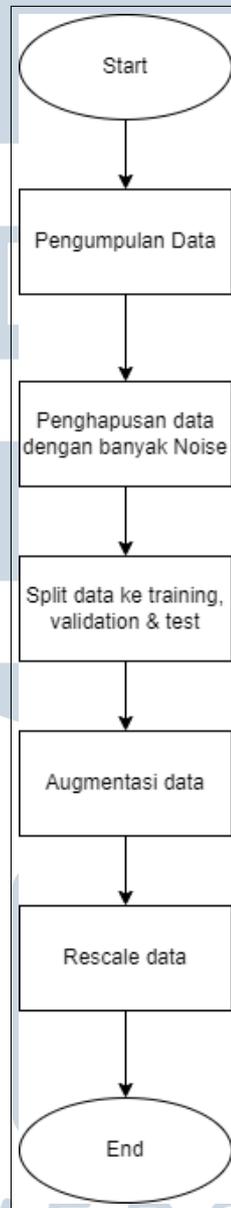
BAB 3 METODOLOGI PENELITIAN

Alur Penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1. Alur Penelitian

3.1 Praproses



Gambar 3.2. Alur Praproses

Tujuan dari praproses adalah untuk membuat data yang digunakan menjadi lebih rapi sehingga pada pembuatan model nanti, hasilnya menjadi lebih valid dan efektif. Gambar 3.2 merupakan alur yang terjadi selama praproses. Praproses dimulai dengan pengumpulan data yang dilakukan dengan mengambil citra-citra jamur yang ada pada situs *Kaggle*. Terdapat 4 buah *folder* pada data aslinya, yaitu *edible mushroom sporocarp*, *edible sporocarp*, *poisonous mushroom sporocarp*,

dan *poisonous sporocarp*. Jumlah data yang terdapat pada *folder* tersebut adalah 1181 untuk *edible* dan 2220 untuk *poisonous*. Setelah dilakukan penghapusan manual untuk data atau citra yang mempunyai terlalu banyak *noise* dan yang kurang relevan dengan penelitian. Setelah itu jumlah sampel data diseragamkan untuk label *edible* dan *poisonous*, maka didapatkan jumlah sampel yang digunakan sebanyak 903 citra jamur *edible* dan 903 citra jamur *poisonous*. Mengeliminasi citra yang kurang jelas membantu proses pembuatan model sehingga dapat mempercepat proses *training* nantinya. Setelah itu, data dibagi menjadi tiga, yaitu *training* atau data latih, *validation* atau data validasi, dan *test* atau data tes. Data dibagi dengan skala 8:2, atau 80% untuk data *training*, dan 20% untuk data *validation*, setelah itu sebanyak 20% dari data validasi diambil sebagai data *test*. Setelah dilakukan pembagian data, maka dilakukan augmentasi untuk data-data tersebut, tujuan dari augmentasi data adalah untuk menambah jumlah dataset agar dapat meningkatkan akurasi yang didapatkan. Setelah itu, dilakukan *rescale* untuk setiap data, karena *pre-trained* model MobileNet V2 membutuhkan data dengan bentuk piksel tertentu untuk dapat dilatih.

3.2 Pembuatan Model

Dibangun sebuah model yang mempunyai fungsi untuk melakukan *feature extraction* terlebih dahulu yang nantinya akan dipakai lagi untuk pemanfaatan *pre-trained* model. Model dibuat menggunakan metode *Convolutional Neural Network* dengan memanfaatkan *library* Keras.

3.3 Pelatihan Model

Pelatihan model dilakukan dengan menggunakan data validasi yang sudah dibagi pada praproses. Validasi citra dilakukan dengan cara melihat apakah citra yang dideteksi dari model yang dibuat sudah sesuai atau belum dengan cara melakukan *testing* untuk citra-citra yang belum pernah digunakan pada saat pelatihan data.

3.4 Fine Tuning Model dengan MobileNet V2

Model yang sudah dilatih untuk melakukan *feature extraction* pada data dilatih lagi dengan melakukan *transfer learning* dan *fine tuning* dengan menggunakan *pre-trained* model MobileNet V2. Tujuan dari dilakukan *fine tuning*

ini adalah untuk meningkatkan lagi akurasi yang didapatkan untuk klasifikasi citra, karena dengan *transfer learning*, model yang dipakai sudah dipakai untuk melatih sangat banyak data citra sehingga model tersebut sudah mempunyai akurasi yang tinggi.

3.5 Evaluasi Model

Model yang sudah dilatih dengan menggunakan *pre-trained* model dievaluasi dengan menampilkan dan melihat akurasi saat pelatihan, akurasi validasi, *loss* saat proses pelatihan, dan *loss* untuk validasi. Tujuan dari evaluasi model ini adalah untuk melihat apakah model yang sudah dilatih tersebut bermasalah atau tidak seperti *underfit* atau *overfit*.

3.6 Klasifikasi Citra dari Model

Setelah proses pelatihan model selesai dilakukan, maka model digunakan untuk melakukan klasifikasi citra jamur pada data *test* untuk melihat prediksi yang dihasilkan, apakah citra yang dideteksi *edible* atau *poisonous*.

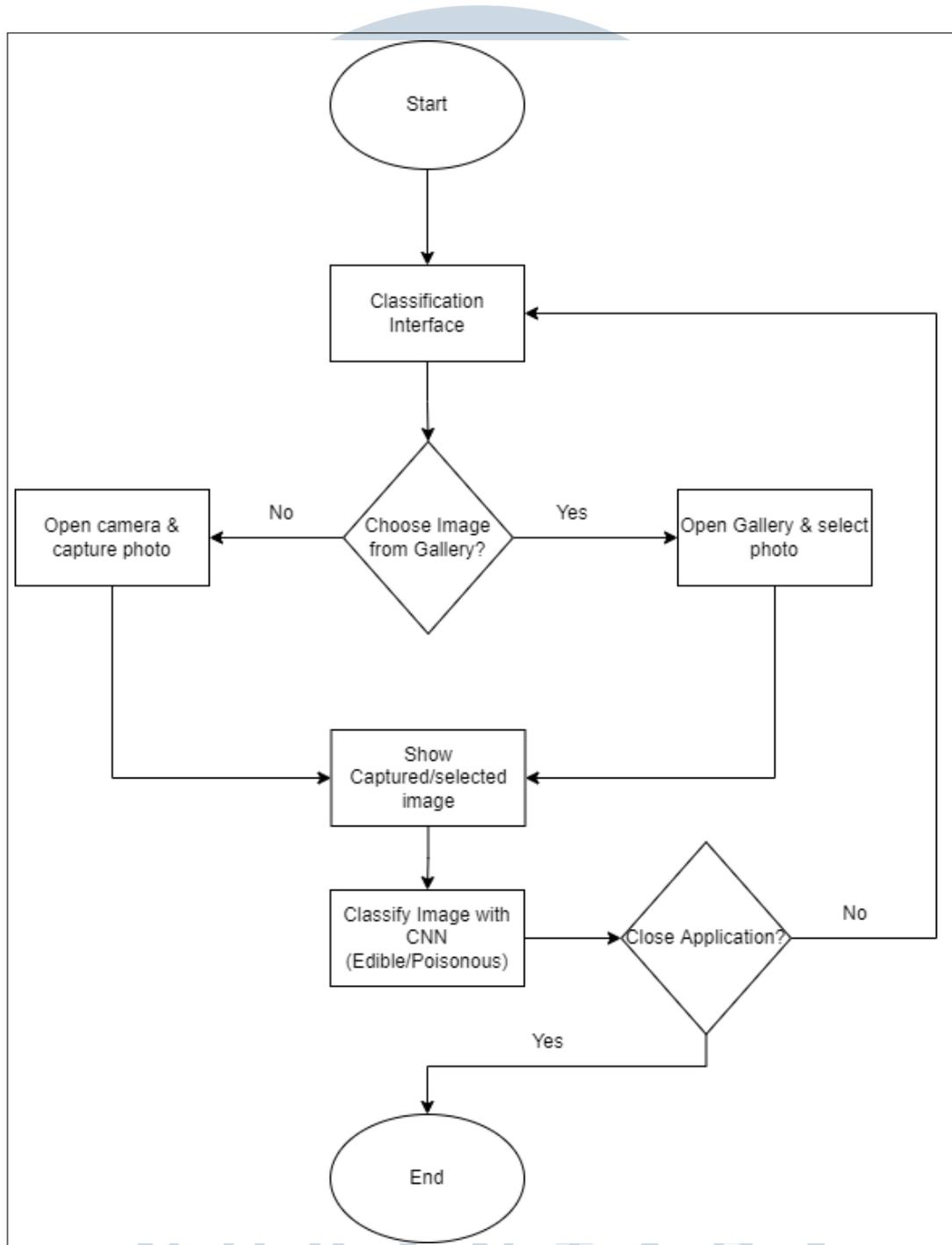
3.7 Perhitungan Akurasi

Perhitungan akurasi, presisi, *recall*, dan F-1 *score* dari model yang sudah dibuat tadi dengan menggunakan *confusion matrix* yang dibuat dari hasil klasifikasi citra.

3.8 Perancangan dan Pembuatan Aplikasi

Perancangan dan pembuatan aplikasi yang dengan menggunakan model yang sudah dibuat dan diekspor untuk digunakan di aplikasi *mobile*. Aplikasi dibuat dengan menggunakan Android Studio dan dapat digunakan secara *mobile*.

3.8.1 Flowchart



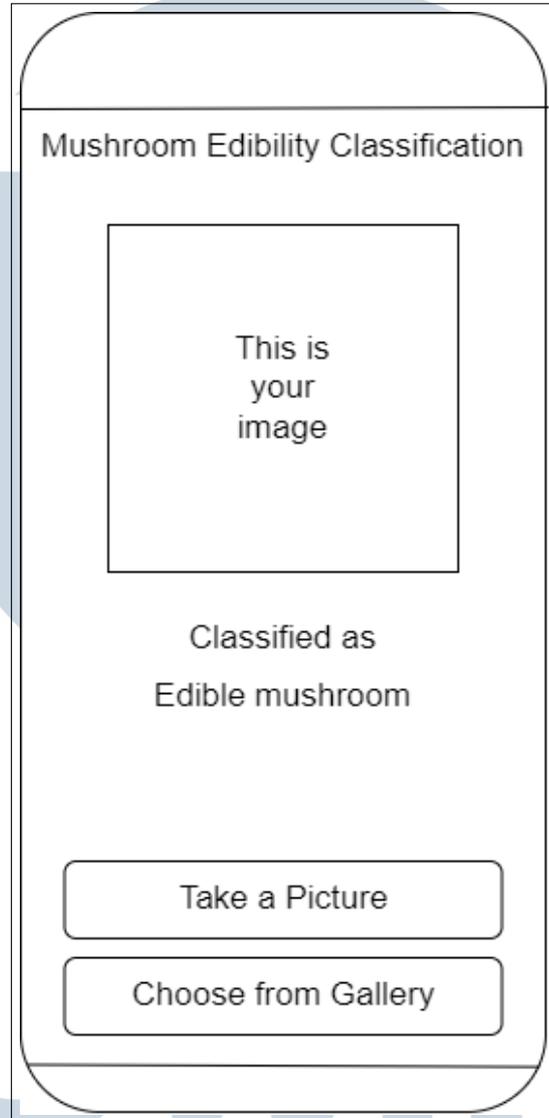
Gambar 3.3. Flowchart rancangan aplikasi

Flowchart untuk perancangan aplikasi dapat dilihat pada Gambar 3.3. Pada saat pengguna menjalankan aplikasi, sistem menampilkan *interface* untuk

melakukan klasifikasi. Pada halaman tersebut, terdapat dua *button* yang dapat dipilih oleh pengguna. *Button* pertama adalah untuk memilih foto yang terdapat pada *gallery device*, dan *button* kedua adalah menggunakan kamera *device* untuk mengambil foto sendiri. Jika pengguna menekan tombol *open camera*, maka sistem meminta *permission* terlebih dahulu kepada pengguna untuk mengaktifkan kamera, setelah itu pengguna baru dapat menangkap gambar yang dipakai untuk klasifikasi. Sedangkan jika pengguna menekan tombol *choose from gallery*, maka sistem membuka *gallery* untuk memilih foto yang digunakan untuk klasifikasi. Setelah citra berhasil difoto atau dipilih, maka sistem menampilkan citra yang sudah dipilih tersebut pada tampilan awal dan melakukan klasifikasi edibilitas untuk citra tersebut pada halaman yang sama. Setelah selesai melakukan klasifikasi, pengguna dapat langsung memilih ingin menggunakan kamera lagi atau memilih foto dari *gallery* pada halaman awal. Jika pengguna menekan tombol *back* pada *device*, maka aplikasi ditutup.



3.8.2 Mockup



Gambar 3.4. Rancangan halaman klasifikasi

Gambar 3.4 merupakan rancangan halaman klasifikasi untuk aplikasi yang nantinya dibuat. Halaman tersebut berisikan *image* yang sudah di-*upload* oleh pengguna, dua *button* yaitu *Take a Picture* dan *Choose from Gallery*, dan hasil klasifikasi berupa tulisan dibawah *Classified as*. Jika pengguna menekan tombol *Take a Picture*, maka kamera dibuka, sedangkan jika pengguna menekan tombol *Choose from Gallery*, maka *gallery* dibuka. *Field* yang berisikan tulisan *This is your image* hanya terisi jika pengguna sudah memilih *citra* yang digunakan untuk klasifikasi.

3.9 Perhitungan Akurasi Model di Aplikasi

Perlu dilakukan perhitungan akurasi kembali untuk model yang sudah diaplikasikan pada aplikasi yang sudah dibuat, yaitu dengan cara menggunakan citra-citra yang sudah diketahui label-nya dan dilihat pada aplikasi apakah hasil klasifikasi benar atau salah. Jumlah citra yang digunakan untuk evaluasi akurasi pada aplikasi adalah sebanyak 10 citra dengan menggunakan kamera dan 10 citra dengan menggunakan *gallery*.

