

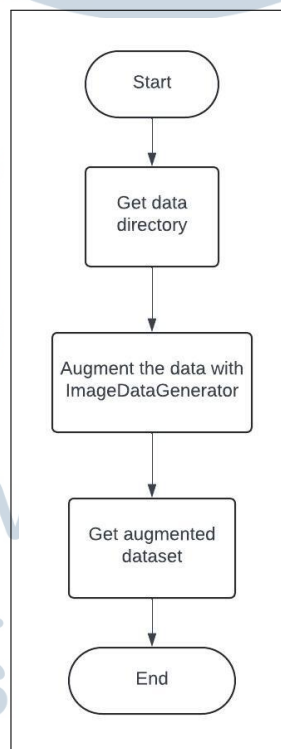
## BAB 3 METODE PENELITIAN

### 3.0.1 Pengumpulan Data

Pada penelitian ini, *dataset* yang digunakan adalah gambar-gambar batik Cirebon yang terdiri dari tiga motif batik. *Dataset* ini didapatkan dari berbagai sumber yang ada di internet terutama pada *website* Roboflow. Tiga motif batik tersebut adalah motif Mega Mendung, motif Singa Barong, motif Keratonan.

### 3.0.2 Pre-processing

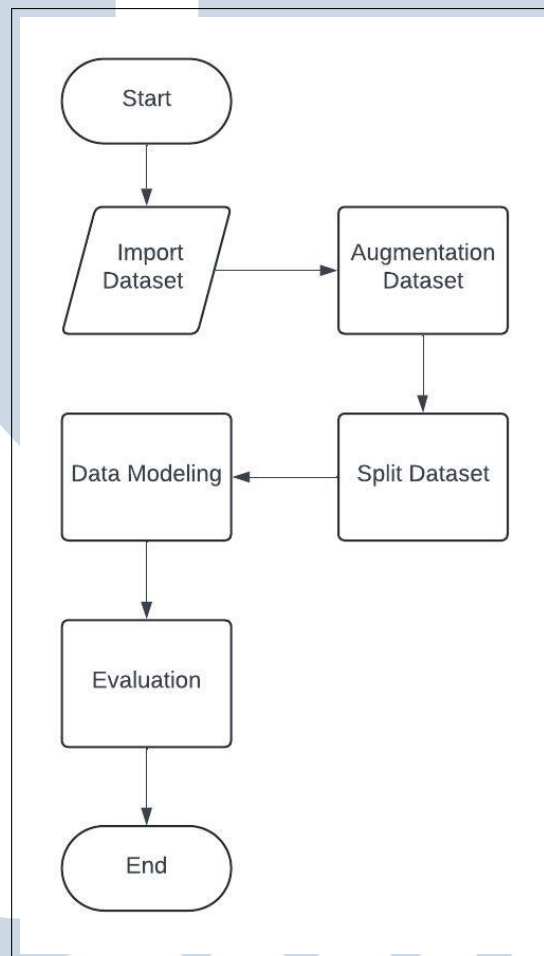
Pada tahapan ini, data akan diunggah ke sistem yang dikelompokkan berdasarkan motif batik. Data ini akan ditambahkan berdasarkan hasil augmentasi menggunakan "imageDataGenerator". Dengan "imageDataGenerator", data yang sudah diunggah akan dilakukan *rescalling* untuk mengubah nilai piksel dari rentang 0-255 menjadi rentang 0-1 [34]. Gambar 3.1 menunjukkan alur dari *pre-processing*.



Gambar 3.1. Flowchart pre-processing

### 3.0.3 Split Dataset

Pada tahap ini, dilakukan pembagian *dataset* dengan perbandingan 75:25, dimana sebanyak 75% data akan digunakan untuk melatih model dengan menggunakan data *training*, dan 25% sisanya akan digunakan untuk menguji model dengan data *validation*. Gambar 3.2 menunjukkan alur dari *split dataset*.

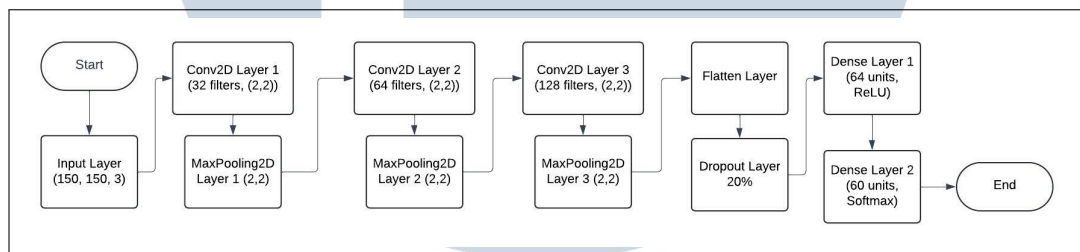


Gambar 3.2. Flowchart split dataset

### 3.0.4 Pembangunan Model

Pada tahapan ini, akan dilakukan pembangunan model secara bertahap dari *layer* ke *layer*. Dibangun 3 *layer* menggunakan aktivasi ReLu dengan masing-masing filter sebanyak 32, 64, dan 128 dengan ukuran *kernel* 3x3. Dari setiap *layer* konvolusi, akan digunakan fitur *max pooling* untuk mengurangi jumlah parameter yang dihasilkan. Setelah 3 *layer* dijalankan, dilakukan perubahan matrik 2

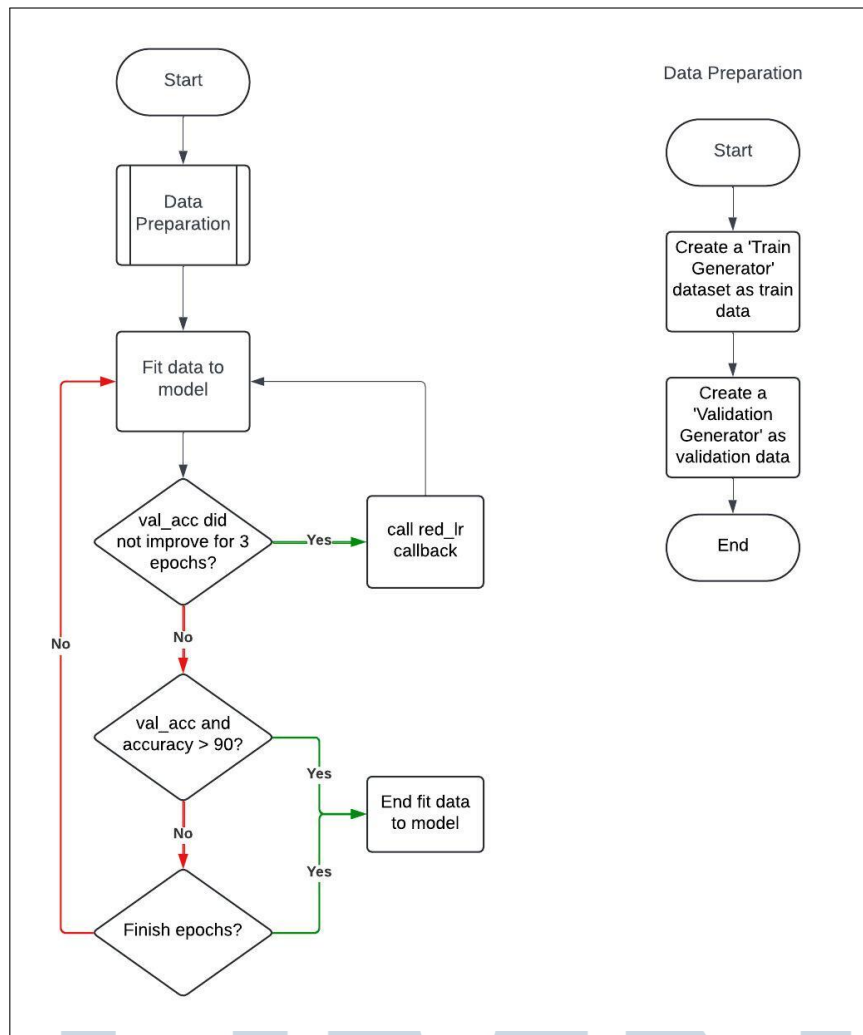
dimensi menjadi vektor 1 dimensi yang akan diinput ke dalam lapisan *dense*. Sebelum memasuki lapisan *dense*, dilakukan penambahan lapisan *dropout* dengan tingkat *dropout* sebesar 20% ( $rate = 0.2$ ). *Dropout* ini berfungsi sebagai salah satu metode regularisasi yang efektif untuk mencegah *overfitting*. Dalam proses *dropout*, secara acak sejumlah neuron akan dinonaktifkan selama proses pelatihan, sehingga model tidak terlalu bergantung pada neuron tertentu dan mampu menghasilkan model yang lebih general. Penerapan *dropout* ini dilakukan sebelum lapisan *dense* pertama, tepat setelah proses *flattening*, untuk memaksimalkan efek regularisasi pada lapisan-lapisan yang padat. Terdapat 2 lapisan *dense* yang digunakan, lapisan pertama menggunakan 64 neuron dengan aktivasi ReLu. Lapisan kedua menggunakan 3 neuron yang sesuai dengan jumlah kelas yang akan diklasifikasi dengan aktivasi *Softmax*. Gambar 3.3 menunjukkan alur dari pembangunan model.



Gambar 3.3. Flowchart model

### 3.0.5 Model Fit dan Callbacks

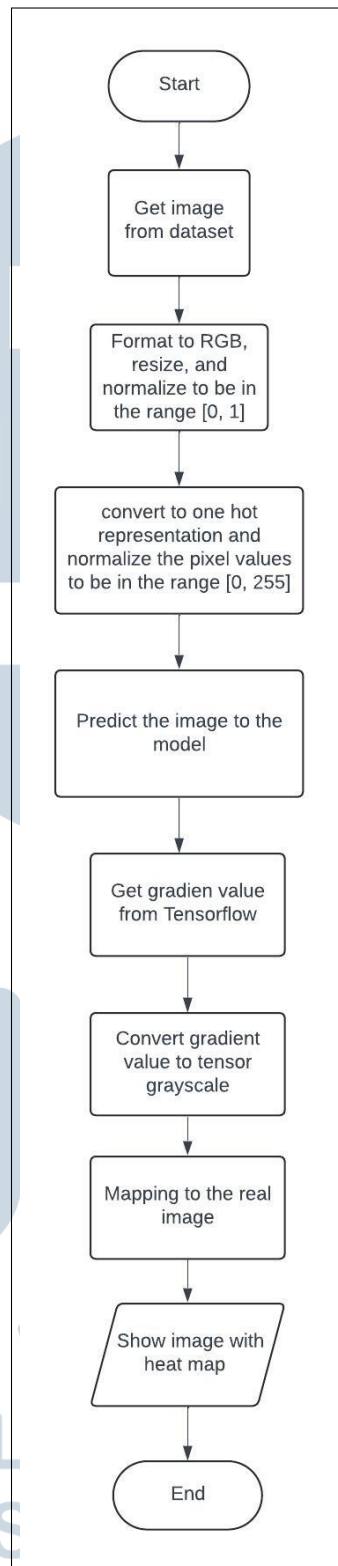
Pada tahapan ini, model yang sudah dibangun akan diuji berdasarkan data *validation* terhadap data *training*. Pengulangan awal ditentukan sebanyak 100 kali pengulangan dengan 2 *callback* yang disediakan. *Callback* 'red\_lr' digunakan untuk mengurangi *learning rate* dari model dengan *patience* sebanyak 3 kali. *Callback* 'my\_callback' digunakan untuk mencegah terjadinya *overfit* dengan cara menghentikan pengulangan apabila sudah mencapai nilai akurasi sebanyak 90%. Penghentian pengulangan dilakukan untuk mengurangi penggunaan *resource* serta waktu yang diperlukan dalam pelatihan model. Gambar 3.4 menunjukkan alur dari model *fit* dan *callbacks*.



Gambar 3.4. Flowchart model fit and callbacks

### 3.0.6 Saliency Map

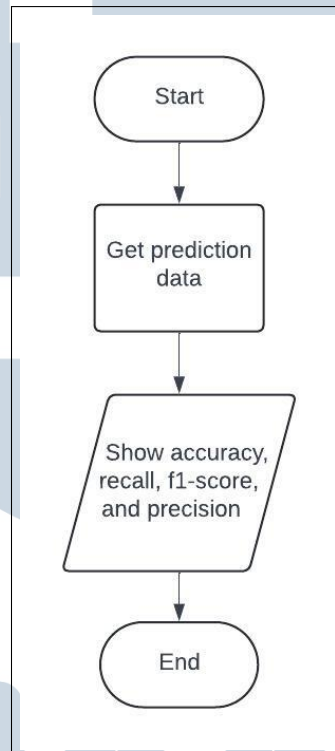
Pada tahapan ini, akan ditampilkan bagian dari batik yang paling mempengaruhi hasil dari prediksi model. Bagian tersebut akan ditampilkan dengan warna berbeda yang diletakan diatas dari gambar aslinya. Gambar 3.5 menunjukkan alur dari *saliency map*.



Gambar 3.5. Flowchart saliency map

### 3.0.7 Evaluasi

Pada tahapan ini, dilakukan evaluasi dengan menggunakan *confusion matrix* untuk mengetahui distribusi kelas dari hasil klasifikasi yang dihasilkan oleh model yang dipilih terhadap kelas sebenarnya. Selain itu, akan ditampilkan juga nilai akurasi, presisi, *recall*, dan *f1-score*. Gambar 3.6 menunjukkan alur dari evaluasi.



Gambar 3.6. Flowchart evaluasi

### 3.0.8 Komparasi dengan Arsitektur Lain

Pada tahapan ini, dilakukan komparasi nilai evaluasi antara arsitektur CNN yang dikembangkan dalam penelitian ini dengan beberapa arsitektur yang telah digunakan pada penelitian sebelumnya. Komparasi ini bertujuan untuk menilai keunggulan atau kelemahan model yang dibangun berdasarkan metrik evaluasi, seperti akurasi, presisi, *recall*, dan *F1-score*.