

BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan dibahas mengenai metodologi penelitian yang dilakukan dalam Tugas Akhir ini.

3.1 Subyek dan Obyek Penelitian

Berdasarkan latar belakang yang telah dijelaskan pada bab 1 diketahui bahwa subjek yang digunakan dalam penelitian ini adalah *dataset* penyakit daun kentang yang diperoleh melalui website kaggle. Sedangkan objek pada penelitian ini adalah klasifikasi penyakit daun kentang.

3.2 Alat dan Bahan Penelitian

Terdapat 3 jenis alat dan bahan yang digunakan dalam penelitian ini yakni perangkat keras, perangkat lunak dan *dataset*.

3.2.1 Spesifikasi Kebutuhan Perangkat Keras

Penelitian ini menggunakan spesifikasi komputer yang bisa dilihat pada tabel 3.1.

Tabel 3.1 Spesifikasi Perangkat Keras

| Komponen | Spesifikasi |
|----------|------------------|
| Prosesor | Intel i5 Gen 10 |
| RAM | 4 GB |
| SSD | 128 Nvme M2 |
| HDD | 1 TB |
| GPU | Nvidia 250MX 2GB |

3.2.2 Spesifikasi Kebutuhan Perangkat Lunak

Penelitian ini menggunakan beberapa perangkat lunak yang dapat dilihat pada tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Lunak

| Nama | Versi | Kegunaan |
|---------------------|--------|-------------|
| <i>Google Colab</i> | - | Kode Editor |
| <i>VS Code</i> | 1.67.2 | Kode Editor |

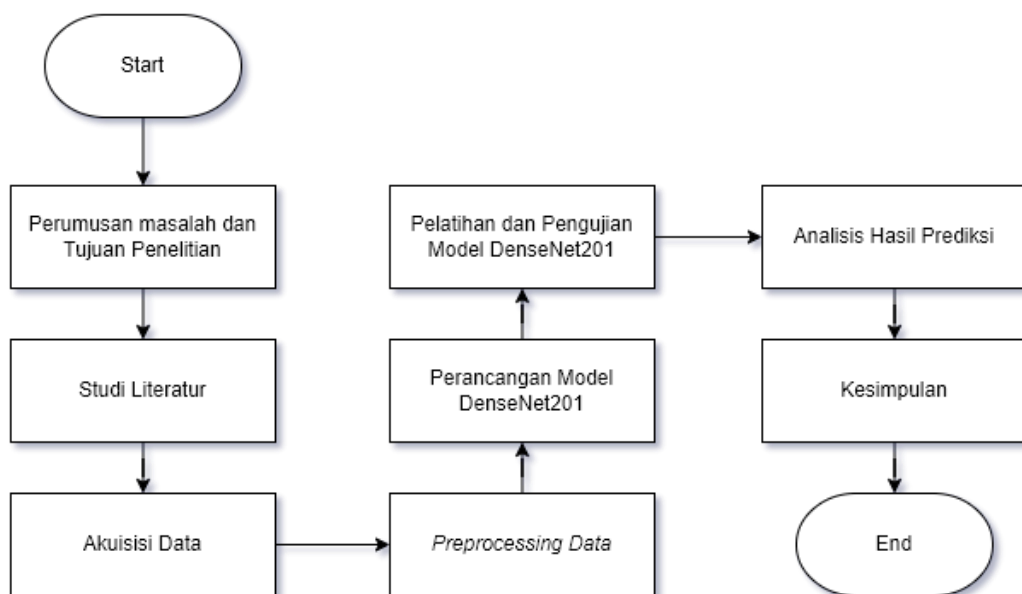
| Nama | Versi | Kegunaan |
|--------------------------------|---------|--------------------------|
| <i>Github</i> | - | <i>Version Control</i> |
| <i>Anaconda</i> | 2020.11 | <i>Library</i> |
| <i>Python</i> | 3.8 | Bahasa Pemrograman |
| <i>FastStone Photo Resizer</i> | - | <i>Image Enhancement</i> |

3.2.3 Dataset

Penelitian ini menggunakan *dataset* penyakit daun kentang yang diunduh dari *website Kaggle* <https://www.kaggle.com/datasets/rizwan123456789/potato-disease-leaf-datasetpld>, dengan jumlah citra sebanyak 3900 citra dengan perincian 1300 citra *potato early blight*, 1300 citra *potato healthy* dan 1300 citra *potato late blight* [38].

3.3 Diagram Alir Penelitian/Proses Penelitian

Gambar 3.1 adalah diagram alir penelitian yang berisikan tahapan-tahapan selama penelitian berlangsung.



Gambar 3.1 Tahapan-tahapan Penelitian

3.3.1 Perumusan Masalah dan Tujuan Penelitian

Untuk mengetahui permasalahan yang ada maka dilakukanlah perumusan masalah dan untuk mengetahui tujuan dari penelitian ini maka dilakukanlah penyusunan tujuan penelitian.

3.3.2 Studi Literatur

Tahap studi literatur merupakan tahap untuk mencari pemahaman konsep dan pemahaman masalah *deep learning* melalui jurnal, buku ilmiah maupun penelitian sebelumnya yang selanjutnya digunakan untuk acuan penulisan dan penelitian yang akan dilakukan.

3.3.3 Akuisisi Data

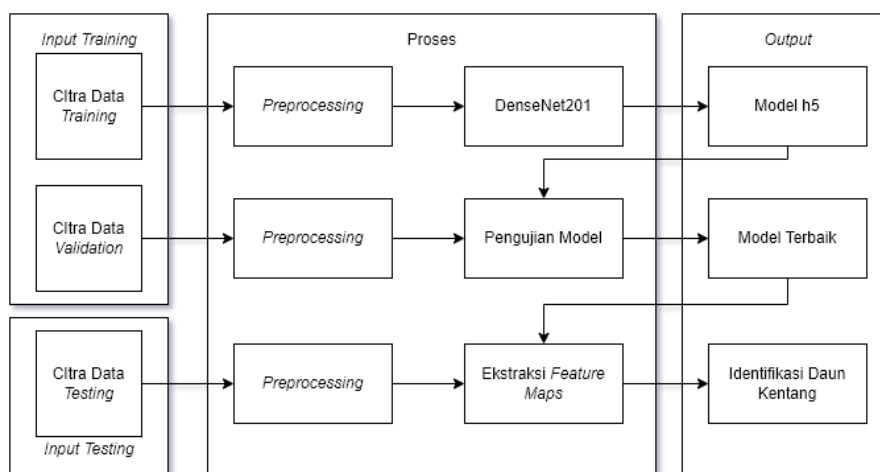
Pengumpulan data pada penelitian ini dilakukan secara kuantitatif. Data citra daun kentang diperoleh melalui *website Kaggle* [38] dengan total citra sebanyak 3900 citra daun kentang.

3.3.4 Preprocessing Data

Tujuan dari tahap *preprocessing* data adalah agar memudahkan model mendapatkan ekstraksi ciri. Metode *preprocessing* data yang digunakan dalam penelitian ini ada *resize* citra. *Resize* merupakan metode untuk mengubah ukuran piksel dari citra pada arah *horizontal* maupun *vertical*. Penelitian ini semua citra akan diubah ukurannya menjadi 224x224 piksel.

3.3.5 Perancangan Model *DenseNet201*

Pada penelitian ini terdapat beberapa tahapan untuk mendeteksi penyakit daun kentang yang didasarkan pada beberapa penelitian terdahulu. Tahapan secara umum untuk mengidentifikasi penyakit daun kentang ada pada gambar 3.2.



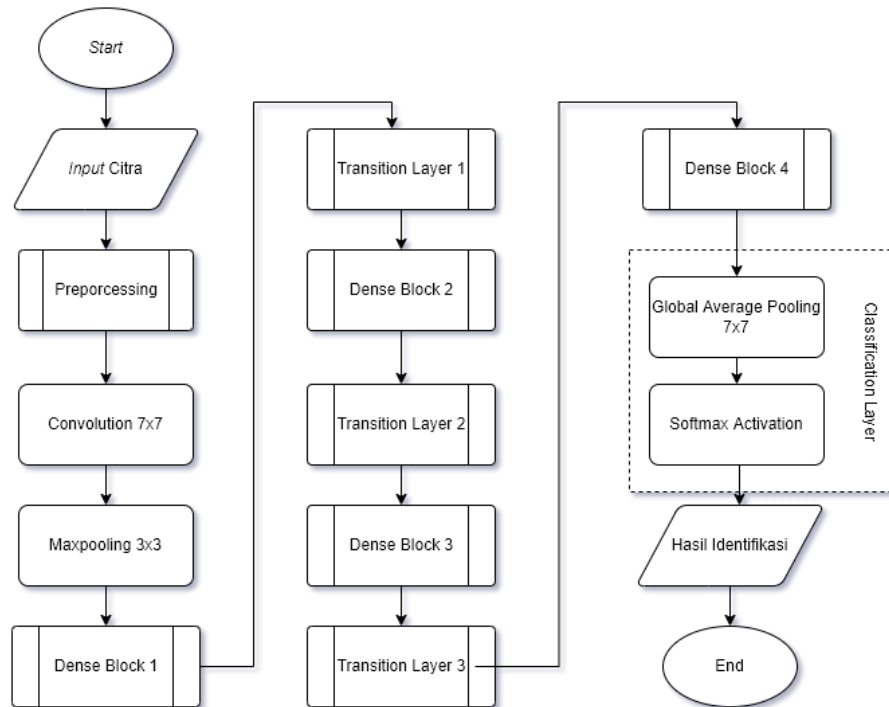
Gambar 3.2 Tahapan Umum Proses Klasifikasi

Input training merupakan bagian pada tahap *training* data dengan memasukkan citra data *training* dan *validation* ke dalam *preprocessing* data.

Prerprocessing data yang dilakukan antara lain mengubah ukuran citra menjadi 224x224 piksel dan menormalisasi data yang sudah *dirsize*. Pada bagian citra data *training* terdapat proses *classification* dan *feature extraction* yang dilakukan menggunakan model *DenseNet201*. Operasi umum yang terdapat pada model *DenseNet201* yakni *batch normalization*, *ReLU Activation*, dan proses *convolution*. *Layer* yang terdapat pada model *DenseNet201* adalah *dense block 1*, *transition layer 1*, *dense block 2*, *transition layer 2*, *dense block 3*, *transition layer 3*, *dense block 4* dan *classification layer* yang nantinya menghasilkan model yang sudah di *training* dengan format h5. Pengujian model dilakukan pada data *validation* proses. *Output* yang dihasilkan dari proses tersebut adalah model terbaik dalam sisi bobot.

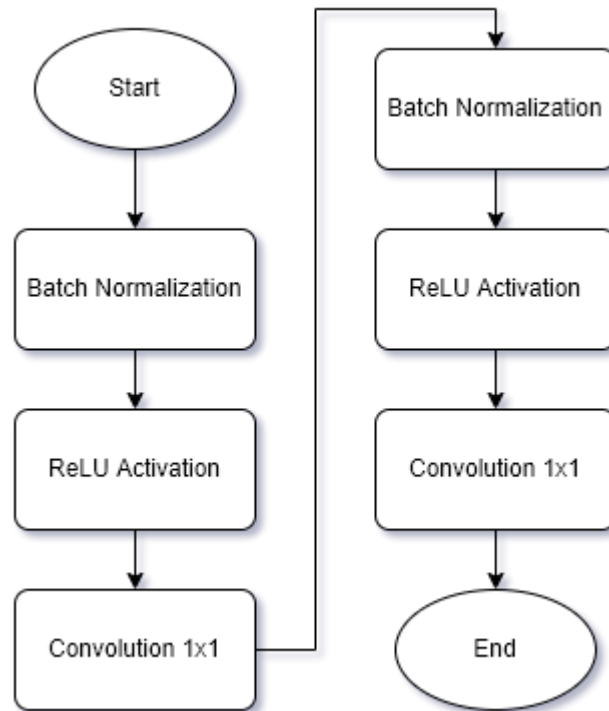
Citra data *testing* akan diubah ukurannya menjadi 224x224 piksel dan dinormalisasikan dengan memasukkannya pada tahap *preprocessing*. Kemudian akan dilakukan ekstraksi *feature maps* dengan cara mengambil nilai bobot terbaik dari data *training* yang nantinya akan menghasilkan output klasifikasi penyakit daun kentang.

Pada gambar 3.3 diilustrasikan *flowchart* algoritma *DenseNet201* dalam menentukan identifikasi penyakit daun kentang. Tahapan dari model *DenseNet201* dimulai dengan *input* citra dan *preprocessing*, lalu dilakukannya operasi *convolution 7x7* dengan *strides 2*. Selanjutnya akan dilakukan proses *max pooling 3x3* dengan *stides 2* yang nantinya akan mendapatkan nilai matriks, nilai matriks ini akan diproses ke dalam *layer dense block 1*, *transition layer 1*, *dense block 2*, *transition layer 2*, *dense block 3*, *transition layer 3*, *dense block 4*. Nilai matriks selanjutnya akan diproses ke dalam *classification layer* dengan operasi *global average pooling 7x7* dan kemudian masuk ke dalam *softmax activation*. *Softmax activation* merupakan *layer* terakhir pada model *DenseNet201* yang digunakan untuk menentukan kelas klasifikasi.



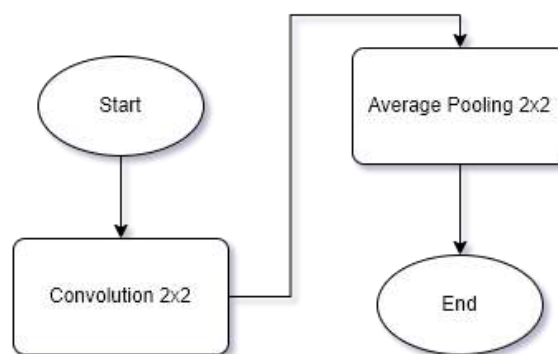
Gambar 3.3 Diagram Alir *DenseNet201*

Gambar 3.4 merupakan tahapan dari proses yang terdapat pada *layer denseblock*. Beberapa tahapan tersebut adalah *batch normalization*, *ReLU activation*, dan *convolution 1x1*, operasi tersebut disebut juga dengan *bottleneck*. Kemudian dilanjutkan dengan proses *batch normalization*, *ReLU activation*, *convolution 3x3* yang nantinya nilai dari setiap matriks akan digabungkan karena saling berhubungan satu sama lain. Di dalam *layer dense block* juga terdapat operasi yang dikalikan sehingga mencapai 201 *layer*. Pada *dense block 1* terdapat *layer convolution* yang dikalikan 6, *dense block 2* terdapat *layer convolution* yang dikalikan 12, *layer convolution* pada *dense block 3* akan dikalikan 48 dan *layer convolution* pada *dense block 4* akan dikalikan 32.



Gambar 3.4 *Dense Block*

Gambar 3.5 merupakan tahapan dari proses yang terdapat pada *transition layer*. Beberapa tahapan tersebut adalah operasi *convolution 2x2* dan *average pooling 1x1* dengan *strides 2*. *Transition layer* akan ditempatkan setelah proses tiap *dense block* sehingga jumlah *transition layer* menjadi 3.



Gambar 3.5 *Transition Layer*

3.3.6 Pelatihan dan Pengujian Model *DenseNet201*

Pengujian dan pelatihan model akan dilakukan ketika seluruh model sudah selesai dirancang. Pelatihan dan pengujian prediksi klasifikasi penyakit daun

kentang dilakukan menggunakan metode *DenseNet201* dimana terdapat beberapa percobaan:

- a) Jumlah *epoch* diatur menjadi 100. Jumlah data sebanyak 3900 citra dan ukuran citra sebesar 224 x 224 piksel merupakan ukuran data yang cukup besar dan membutuhkan waktu pelatihan yang lebih lama. Pemilihan jumlah epoch 100 didasarkan pada hasil penelitian sebelumnya [40] yang mengatakan bahwa jumlah epoch yang lebih rendah dapat mempercepat waktu pelatihan tanpa terlalu mempengaruhi performa [40].
- b) Jumlah *batch size* di *setting* menjadi 64 agar proses komputasi menjadi lebih cepat.
- c) Membandingkan *optimizer Adam, SGD, dan RMSprop*.
- d) Membandingkan *dropout* dengan nilai 0.1, 0.2, 0.3, 0.4, 0.5, 0.6. Nilai 0.1, 0.2, 0.3, 0.4, 0.5, dan 0.6 memberikan regulasi yang berbeda-beda yang mencakup tingkat regulasi rendah dan tinggi. Pemilihan nilai dropout mengacu pada hasil penelitian [41] yang mengatakan bahwa perbandingan nilai-nilai ini memberikan informasi yang berguna tentang bagaimana tingkat regulasi mempengaruhi performa model [41].
- e) Membekukan *bottom layer* pada arsitektur *DenseNet201* yang diharapkan dapat mencegah terjadinya *overfitting*.
- f) Proporsi pembagian data yang digunakan adalah 80% untuk data *training* dan 20% untuk data validasi.

3.3.7 Analisis Hasil Prediksi

Tahap analisis hasil prediksi adalah metode yang digunakan untuk menguji hasil prediksi dari model yang telah dibuat. Hasil dari eksperimen model tersebut kemudian akan dilakukan analisis sehingga dapat diambil sebuah kesimpulan. Tahapan analisis pada penelitian ini menggunakan *confusion matrix* untuk mengevaluasi kinerja model yang sudah dibuat. Tabel 3.3 merupakan komponen dari *confusion matrix*.

Tabel 3.3 *Confussion Matrix*

| <i>Confussion Matrix</i> | <i>Predicted Class</i> | |
|--------------------------|------------------------|----------------|
| | <i>Positif</i> | <i>Negatif</i> |
| | | |

| | | | |
|-------------------|----------------|----|----|
| <i>True Class</i> | <i>Positif</i> | TP | FP |
| | <i>Negatif</i> | FN | TN |

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (3.1)$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (3.2)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (3.3)$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \quad (3.4)$$

Keterangan:

TP (*True Positif*) = jumlah data positif yang diklasifikasi benar.

TN (*True Negatif*) = jumlah data negatif yang diklasifikasi benar.

FP (*False Positif*) = jumlah data positif yang diklasifikasi salah.

FN (*False Negatif*) = jumlah data negatif yang diklasifikasi salah.

3.3.8 Kesimpulan

Kesimpulan merupakan tahapan akhir dari penelitian ini yang berguna untuk mengambil inti dari penelitian ini sehingga dapat dijadikan pemahaman yang utuh dan komprehensif. Kesimpulan merupakan jawaban dari rumusan masalah yang ada, jika tujuan dari penelitian ini sudah dicapai maka penelitian ini dapat dijadikan referensi untuk penelitian selanjutnya.