

BAB III METODOLOGI PENELITIAN

Metodologi penelitian merupakan suatu proses perencanaan penelitian secara sistematis dan ilmiah. Penelitian ini akan menggunakan metode penelitian komparatif. Metode analisis yang digunakan adalah metode *data mining*, khususnya klasifikasi dengan menggunakan perangkat lunak *Orange*. Penelitian ini akan dilakukan melalui serangkaian tahapan yang dijelaskan secara ilustratif pada Gambar 3.1



Gambar 3. 1 Alur Metode Penelitian

3.1 Studi Literatur

Langkah awal yang dilakukan adalah melakukan studi literatur, yang melibatkan pencarian referensi dan pemahaman dari berbagai sumber seperti buku, jurnal, dan artikel yang relevan dengan penelitian ini. Tujuan utama dari studi literatur dalam penelitian ini adalah untuk memperoleh pemahaman yang komprehensif dari sumber-sumber terdahulu yang dapat membimbing penelitian ini. Studi literatur dimulai dengan menyelidiki penelitian terdahulu yang terkait dengan *data mining*, klasifikasi, Algoritma *K-Nearest Neighbors*, serta teknik evaluasi data seperti *k-fold Cross Validation* dan matriks kebingungan.

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data primer yang diambil dengan menggunakan kamera *android* dan memfoto daun terong sehat dan sakit dengan jumlah *data* gambar sebanyak 1000 gambar daun terong sehat dan sakit.

3.3 Pemrosesan Awal

Data yang digunakan dalam penelitian ini adalah *data primer*, dimana jika diperlukan seperti membuang *background* atau latar belakang citra yang tidak diperlukan

3.4 Klasifikasi Daun Terong Sehat dan Sakit

Pada tahap ini penulis sudah mendapatkan *dataset* gambar dengan jumlah 500 gambar daun terong sehat dan 500 gambar daun terong sakit, kemudian mengelompokan *dataset* daun terong menjadi 2 kelas dengan membuat dua folder yaitu daun terong sehat dan daun terong sakit setelah membuat 2 folder seterusnya mengelompokan dengan memindahkan gambar daun terong yang sehat ke dalam folder daun terong sehat dan daun terong yang sakit ke dalam folder daun terong yang sakit. Berikut tabel jumlah *dataset* dari setiap kelas. Setelah mengelompokan *dataset* gambar daun terong yang sehat dan sakit selanjutnya menggunakan bantuan *Orange* untuk melakukan proses klasifikasi. Setelah itu menggunakan *widget Import Images* untuk mengupload *dataset* yang sudah dikelompokan tadi kedalam aplikasi *Orange*.

3.5 Image Embedding

Widget Image Emmbedding melakukan *image embedding* melalui *deep neural network* dan nantinya akan menerjemahkan citra menjadi angka, untuk penyematnya digunakan *SqueezeNet* sebagai *emmbedder*. *SqueezeNet* adalah model mendalam untuk pengenalan gambar yang mencapai akurasi tingkat *Alexnet* di *ImageNet* dengan parameter 50x lebih sedikit. *SqueezeNet* merupakan salah *satu embedder* yang cepat dan tidak memerlukan koneksi internet. Proses ini menerjemahkan citra yang akan digunakan dalam mendeteksi penyakit yang ada pada daun terong berbasis pengklasifikasi *K-Nearest Neighbors*. *Image Embedding* nantinya akan merubah citra menjadi

angka, dimana dalam penelitian ini komputer berhasil *generate dataset* daun Terong yang sehat dan sakit sebanyak 1000 kolom dan 1000 baris.

3.6 Test and Score

Widget test and score akan menguji model klasifikasi pengklasifikasi *K-Nearest Neighbor* yang terpilih, pengujian *widget test and score* menggunakan *cross validation*. *Cross validation* berfungsi untuk membandingkan data secara acak sesuai dengan angka yang ada di dalam *number of fold* yakni 2, 3, 5, 10 dan 20. Ada dua hal yang dilakukan *widget test and score*. pertama akan menampilkan tabel dengan ukuran kinerja klasifikasi yang berbeda, seperti hasil perhitungan *accuracy*, *precision*, *recall*, *f1-score*, dan *AUC*. Dan kedua menampilkan hasil *output* yang digunakan oleh *widget* lain untuk menganalisis kinerja *confusion matrix*.

3.7 Perhitungan Performa

Pada tahap ini penulis menghitung performa dari algoritma *K-Nearest Neighbors* menggunakan *widget test and score* dan menggunakan *widget confusion matrix* untuk menampilkan hasil presentasi dari tiap-tiap kelas. Performa yang dihitung adalah *accuracy*, *precision*, *recall*, dan *f1-score*. Persamaan (3.1), (3.2), (3.3), (3.4) menunjukkan rumus perhitungan performa tersebut. *TP* adalah *True Positive*, *TN* adalah *True Negative*, *FP* adalah *False Positive*, dan *FN* adalah *False Negative*.

Tabel 3. 1 *Confusion Matrix*

<i>Class</i>	<i>Predictive Positive</i>	<i>Predictive Negatif</i>
<i>Actual Positive</i>	<i>TP</i>	<i>FN</i>
<i>Actual Negative</i>	<i>FP</i>	<i>TN</i>

1. Akurasi

Akurasi dapat diilustrasikan seberapa akurat model untuk mengklasifikasikan dengan benar. Dengan demikian, akurasi prediksi adalah perbandingan jumlah *data* positif benar dan negatif benar dengan data keseluruhan. Dengan kata lain, akurasi merupakan tingkat kedekatan nilai dari prediksi dengan nilai yang sebenarnya. Nilai dari akurasi dapat dilihat pada persamaan(3.1)

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

2. Presisi

Presisi dapat digambarkan sebagai tingkat akurasi *data* yang diminta dengan hasil prediksi yang diberikan oleh model. Maka dapat diartikan bahwa presisi itu merupakan perbandingan dari prediksi benar positif dibagi dengan keseluruhan hasil yang diprediksi positif. Dengan kata lain dari semua kelas positif yang telah diprediksi dengan benar, berapa banyak data yang benar-benar positif. Nilai dari presisi dapat dilihat pada persamaan(3.2)

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

3. Recall

Recall atau sensitifitas dapat digambarkan sebagai keberhasilan model dalam mendapatkan sebuah informasi. Maka dapat dikatakan bahwa *recall* itu adalah rasio prediksi benar positif dibagi dengan keseluruhan *data* yang benar positif. Nilai dari *recall* dapat dilihat pada persamaan(3.3)

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

4. Skor F1

Skore F1 dapat dikatakan sebagai perbandingan rata-rata presisi dan *recall* yang dibobotkan. Skor F1 dapat dikatakan terbaik jika ada ssemacam keseimbangan antara presisi dan *recall* dalam sistem. Nilai dari skor F1 dapat dilihat pada persamaan(3.4)

$$F1 - Score = \frac{precision \cdot recall}{precision + recall} \quad (3.4)$$

3.8 Analisis Perangkat Keras dan Perangkat lunak

Adapun perangkat keras dan perangkat lunak yang dibutuhkan dalam pembuatan sistem identifikasi.

A. Perangkat Keras

Perangkat keras yang digunakan pada sistem ini:

a. PC / Laptop

- b. *Procecor : Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz
(4CPUs), ~2.3GHz*
- c. *RAM : 4.00 GB*
- d. *Kamera smartphone 50 MP.*

B. Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun sistem ini:

- a. *Sistem Operasi Windows 10*
- b. *Orange Data Mining*