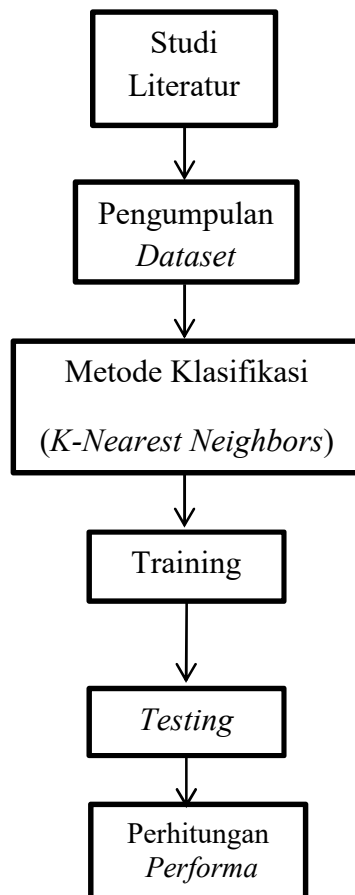


BAB III

METODOLOGI PENELITIAN

Pada tahap penelitian ini yang dilakukan terdiri dari studi literatur, pengumpulan *dataset*, pembangunan model klasifikasi, *training* model klasifikasi, *testing*, dan perhitungan performa.



Gambar 3.1 Alur Penelitian

3.1 Studi Literatur

Langkah pertama yang dilakukan adalah studi literatur yaitu mencari referensi dan pengetahuan dari berbagai sumber seperti buku, jurnal dan artikel yang berhubungan dengan penelitian yang akan dilakukan. Tujuan utama studi literatur dalam penelitian ini adalah untuk mempelajari sumber-sumber terdahulu sebagai bahan panduan untuk mendukung penelitian ini. Studi literatur dimulai dengan mempelajari penelitian sebelumnya yang terkait tentang data *mining*, pengklasifikasi, Algoritma KNN, kemudian mempelajari cara evaluasi data menggunakan *k-fold cross Validation dan confusion matrix*.

3.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data primer yang diambil dengan menggunakan kamera DSLR dan menggunakan *background* berwarna putih, dari pengambilan data ini mendapatkan citra 80 beringin putih, 80 beringin elegan, 80 beringin dolar dan 80 beringin kimeng.

3.3 Pemrosesan Awal

Setelah melakukan proses pengumpulan data selanjutnya melakukan proses seperti membuang *background* atau latar belakang citra yang tidak diperlukan. Berbeda dengan data sekunder yang diambil dari internet karena biasanya data sekunder digunakan tidak perlu membuang *background* atau latar belakang citra yang diambil.

3.4 Klasifikasi Daun Beringin

Pada tahap ini setelah mendapatkan *dataset* gambar dengan jumlah 320 gambar daun beringin, dari data yang sudah didapatkan kemudian mengelompokkan *dataset* daun beringin dibagi menjadi 4 kelas dengan membuat 4 folder yaitu beringin putih, beringin elegan, beringin dolar, dan beringin kimeng. Setelah membuat 4 folder seterusnya mengelompokkan dengan memindahkan gambar daun beringin putih ke dalam folder beringin putih, beringin elegan kedalam folder beringin elegan, beringin dolar kedalam folder beringin dolar dan beringin kimeng kedalam folder beringin kimeng.

3.5 *Image embedding*

Tahap pemrosesan data menggunakan *Widget Image Embedding* dimana fungsi *image embedding* ini akan membaca *dataset* citra daun beringin agar dapat dibaca oleh sistem atau komputer. Melakukan *image embedding* melalui *deep neural network* dan nantinya akan menerjemahkan citra yang sudah diubah menjadi angka, untuk *embedernya* digunakan *SqueezeNet* sebagai *embedder*.

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 mengidentifikasi daun beringin berbasis pengklasifikasi KNN.

3.6 Test and Score

Pada tahap *Widget test and score* akan menguji model klasifikasi KNN yang terpilih, pengujian *widget test* dan *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 akurasi kinerja klasifikasi yang berbeda, seperti hasil perhitungan *accuracy*, *precision*, *recall*, *f1-score*, dan *AUC*. Hal yang kedua, yaitu menampilkan hasil *output* yang digunakan oleh *widget* lain untuk menganalisis kinerja *confusion matrix*.

3.7 Pembentukan Model Klasifikasi

Pembentukan model klasifikasi algoritma dibangun menggunakan beberapa parameter agar dapat mengetahui pengaruh parameter terhadap *performa* yang dihasilkan. Pada algoritma *K-Nearest Neighbors* (KNN) parameter yang digunakan adalah tipe *distance* dan jumlah *neighbors*. Untuk parameter *K-Nearest Neighbors* (KNN) dapat di lihat pada tabel 3.1 berikut ini :

Tabel 3.1 Parameter

Parameter	Deskripsi
Jumlah <i>Neighbor</i>	2
<i>Type Distance</i>	<i>Euclidean</i>

3.8 Perhitungan Performa

Pada tahap ini penelitian menghitung *performa* dari Algoritma *K-Nearest Neighbors* (KNN) menggunakan *Widget Test and Score* dan hasil evaluasi menggunakan *widget confusion matrix* untuk menampilkan tiap kelas. *Performa* yang dihitung adalah *accurasi*, *precision*, *recall* dan *F1 Score*. Hasil presentasi dari tiap-ll dan *F1 Score*, Persamaan menunjukkan rumus perhitungan *accurasi*, *precision*, *recall* dan *F1 Score*. *TP* adalah *True Positive*, *TN* adalah *True Negative*, *FP* adalah *False Positive* dan *FN* adalah *False Negative*.

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 berikut ini :

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN} \dots\dots\dots 1.5$$

Precision dapat digambarkan sebagai tingkat akurasi data yang diminta dengan hasil predikksi yang diberikan oleh model. Maka dapat diartikan itu merupakan perbandingan dari prediksi benar positif dibagi dengan keseluruhan hasil yang diprediksi 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 *precision* dapat di lihat pada persamaan berikut ini :

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots 1.6$$

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 berikut ini :

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots 1.7$$

Score F1 dapat dikatakan sebagai perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Score F1* dapat dikatakan terbaik jika ada semacam keseimbangan antara *precision* dan *recall* dalam sistem. Nilai dari *Score F1* dapat dilihat pada persamaan berikut ini :

$$F1 - Score = \frac{2 \cdot precision \cdot recall}{precision + recall} \dots\dots\dots 1.8$$

3.9 Analisis Perangkat Keras dan Perangkat Lunak Yang Dibutuhkan Dalam Pembuatan Sistem Identifikasi

1. Perangkat Keras

Perangkat keras yang digunakan pada sistem ini adalah sebagai berikut :

- a. PC / Laptop
- b. *Prosesor* : Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz (4CPUs), ~2.3GHz
- c. RAM : 4.00 GB
- d. Kamera *smarTPhone* Oppo A12 .
- e. Mistar Mika
- f. *Tripot* Hp

g. Kertas HVS Putih

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam membangun sistem ini adalah sebagai berikut :

a. Sistem Operasi *Windows 10*

b. *Orange*

