

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jamur merupakan jenis tumbuhan yang mudah tumbuh dan berkembang di berbagai ekosistem, terdapat ribuan jenis jamur yang dapat dimanfaatkan untuk membantu kehidupan manusia maupun untuk dikonsumsi, namun beberapa jenis jamur juga mempunyai zat yang berbahaya sehingga dapat mengakibatkan keracunan apabila dikonsumsi makhluk hidup, selama kurun waktu 10 tahun terakhir (2010-2020) tercatat telah terjadi sebanyak 76 kasus keracunan berbagai jamur liar di Indonesia, dengan 550 korban dan 9 di antaranya meninggal dunia [1]. Fenomena ini menunjukkan bahwa pengetahuan masyarakat mengenai perbedaan antara jamur yang layak konsumsi dan jamur beracun masih sangat terbatas. Oleh karena itu, upaya untuk mengidentifikasi dan mengklasifikasikan jenis jamur secara akurat menjadi sangat penting guna meminimalisir risiko keracunan dan meningkatkan keamanan pangan.

Beberapa jenis jamur tidak beracun dan memiliki nilai gizi tinggi, seperti jamur kancing, enoki, kuping, *maitake*, *shitake*, grigit, merang, dan jamur tiram. Namun, terdapat pula jenis jamur yang bersifat toksik dan berbahaya bagi kesehatan manusia, seperti jamur dari *Autumn Skullcap*, *Death Cap*, *Destroying Angels*, *False Morel*, *Poison Fire Coral*, *Chlorophyllum*, *omphalotus* dan *paxillus involutus*. Dalam mengenali jamur yang tidak beracun dan yang tidak bisa dikenali dengan memperhatikan ciri-ciri morfologi jamur seperti bentuk payung, warna, tekstur payung, dan ciri lain yang dapat dilihat. Timbul permasalahan,

beberapa jamur beracun memiliki ciri morfologi yang sangat mirip dengan jamur yang dapat dikonsumsi, akan cukup sulit jika hanya dilihat menggunakan mata telanjang untuk mengidentifikasi jamur tersebut termasuk beracun atau dapat dikonsumsi. Mengenali jamur yang bisa dikonsumsi sangat diperlukan, karena jika jamur yang dikonsumsi merupakan jamur beracun, maka akan mengakibatkan keracunan yang dapat mengganggu fungsi organ pencernaan [2].

Seiring dengan perkembangan teknologi informasi dan komputer, berbagai metode telah dikembangkan untuk membantu proses identifikasi dan klasifikasi jamur, salah satunya adalah dengan memanfaatkan teknologi pengolahan citra digital. Pengolahan citra digital memungkinkan ekstraksi fitur-fitur visual dari gambar jamur, yang kemudian dapat digunakan sebagai dasar untuk membedakan antara jamur beracun dan tidak beracun. Salah satu metode yang banyak digunakan dalam pengolahan citra untuk klasifikasi adalah *Support Vector Machine (SVM)*.

Support Vector Machine (SVM) adalah algoritma dalam *machine learning* yang digunakan untuk menyelesaikan masalah klasifikasi dan *regresi*. *SVM* merupakan metode klasifikasi yang *semi-eager learner*, *Support Vector Machine* tidak hanya membutuhkan proses pelatihan, tetapi juga menyimpan beberapa data latih untuk digunakan kembali selama proses prediksi [3].

Dalam penelitian yang dilakukan oleh Nuke L. Chusna, dkk dijelaskan bahwa algoritma *Multiclass Support Vector Machine* dapat digunakan untuk melakukan klasifikasi citra jenis tanaman jamur layak konsumsi. Metode yang

dikembangkan diimplementasikan pada *Matlab*, agar menghasilkan sistem dalam bentuk *GUI* sehingga dapat digunakan oleh pengguna umum dengan mudah. Berdasarkan hasil pengujian akurasi rata-rata akurasi sebesar 83% [4].

Penelitian lainnya Klasifikasi Penyakit Tanaman Bawang Merah Menggunakan Metode *SVM* dan *CNN* Pada penelitian ini, klasifikasi menggunakan *CNN* dan juga *SVM* dengan kernel *RBF*, *linear*, dan *polynomial* memberikan nilai *accuracy*, *precision*, *recall*, dan *F1 score* masing-masing 100%. *SVM* didahului dengan proses ekstraksi fitur tekstur menggunakan *GLCM*. Sedangkan, apabila *SVM* menggunakan kernel *sigmoid*, performanya jauh lebih buruk. Pada penelitian selanjutnya disarankan menggunakan beberapa penyakit bawang merah dengan menggunakan ekstraksi [5].

Permasalahan utama dalam penelitian ini adalah bagaimana mengimplementasikan algoritma *SVM* untuk mengklasifikasikan citra jamur beracun dan tidak beracun secara otomatis dengan akurasi tinggi. Meskipun beberapa penelitian telah mengangkat penggunaan *SVM* dalam klasifikasi citra, penerapannya secara spesifik untuk mengklasifikasikan jamur beracun dan tidak beracun di Indonesia masih terbatas. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan algoritma *Support Vector Machine (SVM)* dalam klasifikasi citra jamur guna membedakan jamur beracun dan tidak beracun secara otomatis berdasarkan fitur visual dari gambar digital. Penelitian ini diharapkan dapat menjadi kontribusi dalam meningkatkan keamanan konsumsi jamur dan meminimalisir risiko keracunan yang diakibatkan oleh kesalahan identifikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, rumusan masalah yang dapat diambil adalah sebagai berikut:

1. Bagaimana merancang, membangun dan mengimplementasikan algoritma *Support Vector Machine* untuk klasifikasi jamur beracun dan tidak beracun?
2. Bagaimana tingkat akurasi dari hasil klasifikasi jamur menggunakan algoritma *SVM*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Mengimplementasikan algoritma *Support Vector Machine* dalam proses klasifikasi jamur beracun dan tidak beracun.
2. Mengukur tingkat akurasi dari hasil klasifikasi algoritma *SVM* dalam melakukan klasifikasi jamur berdasarkan dataset yang digunakan.

1.4 Batasan Masalah

Batasan masalah ini dibatasi pada:

1. Dataset yang digunakan dalam penelitian ini diperoleh dari situs Kaggle dan roboflow. Dataset tersebut berisi atribut-atribut yang merepresentasikan karakteristik fisik dari berbagai jenis jamur.
2. Kategori jamur beracun yang digunakan terdiri dari 8 jenis, yaitu Autumn Skullcap, Death Cap, Destroying Angels, False Morel, poison fire coral, paxillus involutus, Omphalotus dan Chlorophyllum. Total data jamur

beracun yang digunakan adalah 4.000 gambar, dengan masing-masing jenis berjumlah 500 gambar.

3. Kategori jamur yang tidak beracun terdiri dari 8 jenis, yaitu Jamur kancing, Jamur Enoki, Jamur Kuping, maitake, Jamur shitake, grigit, merang dan Jamur tiram. Total data jamur tidak beracun yang digunakan adalah 4.000 gambar, dengan masing-masing jenis berjumlah 500 gambar.
4. Proses klasifikasi dalam penelitian ini menggunakan algoritma Support Vector Machine (SVM) dengan kernel Radial Basis Function (RBF).
5. Evaluasi performa model terbatas pada akurasi, *precision*, *recall*, dan *F1-score* sebagai matrik evaluasi utama.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan wawasan bagi mahasiswa dalam menerapkan algoritma *Support Vector Machine (SVM)* pada kasus nyata, serta meningkatkan pemahaman mereka tentang pemanfaatan teknologi kecerdasan buatan.
2. Memberikan kontribusi dalam pengembangan aplikasi berbasis *machine learning* untuk identifikasi jenis jamur.
3. Membantu masyarakat dalam membedakan jamur beracun dan tidak beracun secara cepat dan akurat.
4. Menjadi referensi bagi peneliti lain dalam mengembangkan metode klasifikasi pada data biologis.

1.6 Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari lima bagian utama sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan pada penelitian ini, teori-teori yang berhubungan dengan kecerdasan buatan, *Support Vector Machine (SVM)*, Klasifikasi.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tahapan-tahapan dalam pengumpulan data, perancangan sistem perumusan masalah dan analisa, serta alat dan teknik yang digunakan untuk implementasi dan pengujian.

BAB 4 ANALISA DAN PERANCANGAN

bab ini berisi analisa dan perancangan implementasi. Dalam bab ini akan dijelaskan proses analisa kebutuhan, perancangan sistem, serta desain antar muka.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari hasil analisa dan perancangan yang telah dilakukan. Selain itu, bab ini juga akan menjelaskan proses pengujian aplikasi, termasuk mode pengujian, hasil pengujian, dan evaluasi terhadap aplikasi yang telah dikembangkan.

BAB 6 PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran-saran untuk pengembangan aplikasi atau penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Jamur

Jamur (fungi) merupakan organisme yang memiliki inti, spora, hifa atau miselium. Umumnya jamur tumbuh secara liar di alam dan jenisnya ada yang beracun dan tidak beracun. Jenis jamur yang tidak beracun adalah jamur konsumsi (*edible mushroom*) [6].

Beberapa spesies jamur diketahui dapat dimakan bahkan mengandung zat yang bisa digunakan sebagai obat-obatan. Namun banyak juga jamur yang tidak dapat dimakan karena mengandung zat beracun [7]. Dalam mengenali jamur yang bisa dikonsumsi dan yang tidak, bisa dikenali dengan memperhatikan ciri-ciri morfologi jamur seperti bentuk payung, warna, tekstur payung, dan ciri lain yang dapat dilihat [2].

Jamur merupakan salah satu keanekaragaman hayati yang banyak tumbuh di Indonesia. Jamur adalah jenis tumbuhan yang tidak memiliki akar dan daun sejati serta tidak bisa melakukan fotosintesis. Keunikan jamur adalah memiliki berbagai macam ukuran, bentuk dan warna. Terdapat 1.500.000 jenis jamur di dunia, telah teridentifikasi 74.000 jenis jamur. Diantara jamur-jamur yang telah teridentifikasi tersebut beberapa jamur memiliki banyak manfaat seperti dapat meningkatkan perekonomian, meningkatkan agrikultur dan berperan di bidang kesehatan [8].

2.1.1 Jamur Beracun

jenis jamur yang beracun, bahkan dapat membahayakan nyawa. Dimana 138.000 species lain masih belum bisa ditentukan dapat dimakan atau tidak dan bahkan mengandung racun yang membahayakan tubuh. Dilaporkan menurut *Center for Disease Control and Prevention (CDC)* lebih dari 41.000 orang meninggal pada tahun 2008 dikarenakan secara tidak sengaja keracunan, sedangkan *World Health Organization (WHO)* melaporkan sekitar 0,346 juta kematian sejak tahun 2004 [9], berdasarkan jurnal yang ditulis oleh Herchenröder et al. (2020) berjudul *Mushroom Poisoning*, dijelaskan bahwa beberapa jenis jamur beracun seperti *Death Cap*, *Destroying Angel*, *Autumn Skullcap*, *False Morel*, *Paxillus Involutus*, *Omphalotus* mengandung senyawa amatoksin yang sangat mematikan, dengan mekanisme racun yang serupa dan berpotensi menyebabkan kerusakan hati akut serta kematian pada manusia [10].

Beberapa Jamur Liar yang Berpotensi Menyebabkan Keracunan di Indonesia", disebutkan bahwa jamur *Poison Fire Coral* mengandung senyawa *trichothecene* yang sangat toksik dan dapat menyebabkan kerusakan organ hingga kematian, sedangkan *Chlorophyllum* dikenal sebagai salah satu penyebab umum keracunan *gastrointestinal* akut yang ditandai dengan mual, muntah, dan diare hebat [11].

Adapun jenis jamur beracun antara lain:



a. Autumn Skullcap



b. Death Cap



c. Destroying Angels



d. False Morel



e. Poison Fire Coral



f. Paxillus Involutus



g. Chlorophyllum



h. Omphalotus

Gambar 2. 1 Jamur Beracun (Sumber: Kaggle <https://www.kaggle.com>)

2.1.2 Jamur Tidak Beracun

Jamur tidak beracun adalah jamur yang aman dikonsumsi oleh manusia, baik secara mentah (dalam beberapa kasus) maupun setelah dimasak. Jamur-macam ini tidak mengandung toksin atau racun yang membahayakan tubuh, dan bahkan banyak di antaranya memiliki nilai gizi tinggi dan digunakan sebagai bahan makanan atau obat tradisional [12]. *Scrutinizing the Nutritional Aspects of Asian Mushrooms, Its Commercialization and Scope for Value-Added Products*” yang secara eksplisit menyatakan bahwa berbagai jenis jamur, seperti jamur kancing, enoki, kuping, tiram, *shiitake*, *maitake*, dan *truffle*, masuk dalam kategori *edible* (aman dan tidak beracun) [13]. *volvariella volvacea* atau jamur merang adalah jenis jamur *edible* yang biasa dikonsumsi secara luas, serta memiliki banyak senyawa *bioaktif* (*antioxidant*, *antimikroba*, anti penuaan, dan nutrisi lengkap). Hal ini menegaskan bahwa jamur ini tidak beracun dan layak dikonsumsi dalam diet manusia [14]. Berikut beberapa contoh jamur tidak beracun yang umum dikenal dan dikonsumsi:



Gambar 2. 2 Jamur Tidak Beracun (Sumber: Kaggle <https://www.kaggle.com>)

2.2 Dataset

Dataset adalah sebuah kumpulan data yang berasal dari informasi-informasi pada masa lalu dan siap untuk dikelola menjadi sebuah informasi baru. Dataset memiliki semua karakteristik, fitur dan fungsi dari database biasa [15]. Dataset adalah suatu *database* di dalam memori (*in-memory*). Dataset memiliki semua karakteristik, fitur dan fungsi dari basis data biasa. Dataset dapat memiliki banyak tabel, dan tabel tabel dapat memiliki hubungan (*relationship*). Tabel-tabel pada suatu dataset dapat memiliki *foreign key* dan integritas referensial [16].

Dataset adalah pengelompokan data dari informasi sebelumnya yang siap dikelola menjadi informasi baru [17].

2.3 *Machine Learning*

Pembelajaran Mesin adalah cabang dari ilmu komputer yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dari data dan membuat prediksi atau keputusan berdasarkan data tersebut. Konsep dasar pembelajaran mesin mencakup pelatihan model menggunakan data historis untuk menemukan pola dan melakukan prediksi terhadap data baru. Algoritma pembelajaran mesin dapat dikategorikan menjadi beberapa jenis, seperti *supervised learning*, *unsupervised learning*, dan *reinforcement learning* [18].

Machine Learning (ML) adalah cabang dari kecerdasan buatan (*artificial intelligence*) yang berfokus pada pengembangan algoritma dan model komputer yang dapat belajar dari data serta melakukan tugas-tugas tertentu tanpa perlu secara eksplisit diprogram [19].

Machine Learning (ML) atau Mesin Pembelajaran adalah cabang dari *AI* yang fokus belajar dari data (*learn from data*), yaitu fokus pada pengembangan sistem yang mampu belajar secara “mandiri” tanpa harus berulang kali diprogram manusia. Jika *Machine Learning* diibaratkan kendaraan bermotor maka data adalah bahan bakar utama dari *machine learning*, hal ini dikarenakan *Machine Learning* membutuhkan data untuk dapat membuat sebuah metode penyelesaian masalah [20].

2.4 *Support Vector Machine (SVM)*

Metode *Support Vector Machine (SVM)* merupakan suatu metode klasifikasi dalam pemecahan masalah untuk dua kelas. *SVM* biasanya digunakan

dalam mengklasifikasikan data yang bisa dipisah secara *linier*, untuk data yang tidak dapat dilakukan pemisahan secara linier maka digunakan fungsi kernel yang tujuannya adalah untuk memetakan data input ke data fitur [21].

SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)*. *SRM* sendiri bertujuan untuk menjamin batas atas dari generalisasi pada data pengujian dengan cara mengontrol "kapasitas" (*fleksibilitas*) dari hipotesis hasil pembelajaran. Ada tiga hal yang harus diperhatikan dalam menggunakan algoritma *SVM* yaitu [22] :

- a. *Support Vectors*, yaitu kelas kelas yang ada pada saat melakukan klasifikasi.
- b. *Hyperplane*, adalah sebuah fungsi untuk memisahkan kelas dengan kelas lainnya.
- c. *Margin*, yaitu jarak pada kelas pertama terhadap kelas kedua

Pada *SVM* masalah pengklasifikasian dilakukan dengan mencari pemisah *hyperplane* yang optimal antar kelas. *Hyperplane* bergantung dari kasus pelatihan yang meletakkannya pada tepi dari deskriptor kelas yang disebut *support vector*. Prinsip dasar *SVM* adalah *linear classifier*, dan selanjutnya dikembangkan agar dapat bekerja pada masalah *non-linear*, dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi [23].

Rumus *SVM* yang sering dipakai adalah sebagai berikut.[24] :

1. *Polynomial Kernel*

Polynomial Kernel merupakan proses yang harus dilakukan untuk mengklasifikasikan *dataset training* yang sudah normal. Rumus ini dapat diimplementasikan menggunakan rumus persamaan 1 sebagai berikut :

$$K(X_i, X_j = X_i, X_j + 1)^d \dots \dots \dots (1)$$

2. *Radial Bias Function (RBF)*

RBF merupakan salah satu rumus fungsi dalam *SVM* yang digunakan untuk klasifikasi dataset yang tidak terpisah secara linier. Pada proses ini *RBF* mempunyai kelebihan tingkat akurasi training dan prediction yang sangat baik. Rumus ini dapat diimplementasikan menggunakan berikut :

$$K(X_i, X_j = \exp(-||X_i, X_j||)y \dots \dots \dots (2)$$

3. *Sigmoid Kernel*

Sigmoid merupakan proses pengembangan dari jaringan saraf tiruan. Rumus ini dapat diimplementasikan menggunakan persamaan 3 berikut :

$$K(X_i, X_j = \tanh(\alpha x_1 + X_j + \beta) \dots \dots \dots (3)$$

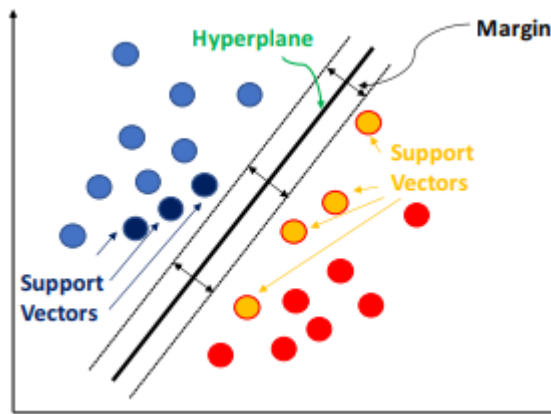
Keterangan :

$K = \text{Kernel}$

$X \ \& \ Y = \text{Vector input space}$

$d = \text{quadratic}$

$\alpha = \text{scalar parameter}$



Gambar 2. 3 *support vector machine (SVM)* (Sumber : Manjrekar and Dudukovic 2019)

2.5 Kecerdasan Buatan (*Artificial Intelligence/AI*)

Kecerdasan buatan (*Artificial Intelligence/AI*) mengacu pada kemampuan mesin untuk meniru kecerdasan manusia dalam mempelajari, berpikir, dan mengambil keputusan, ini melibatkan penggunaan algoritma dan teknik komputasi yang kompleks untuk mengolah data, mengenal pola, dan membuat prediksi atau tindakan yang cerdas, contoh penggunaannya adalah *machine learning*, *neural networks*, *natural language processing*, *computer vision*, dan robotika, kecerdasan buatan ini bertujuan untuk meningkatkan efisiensi dan kualitas hidup melalui teknologi cerdas. [26].

Kecerdasan buatan (*AI*) adalah bidang ilmu komputer yang menggunakan simbol daripada angka untuk menyampaikan informasi dan menggunakan metode heuristik atau sejumlah aturan untuk memproses data [27]. Salah satu bidang sains dan teknik yang paling baru adalah kecerdasan buatan, atau *AI* penelitian tentang kecerdasan buatan dimulai secara resmi setelah Perang Dunia II, dan nama *AI* pertama kali digunakan pada tahun 1956 [28].

Semakin banyak pengembangan *hardware* dan *software* yang menggunakan teknik kecerdasan buatan menunjukkan betapa berkembangnya kecerdasan buatan saat ini di bidang ilmu komputer [29]. Kemampuan mesin untuk meniru kecerdasan manusia dalam hal berpikir, belajar, dan membuat keputusan disebut kecerdasan buatan (*AI*). Ini mencakup penggunaan teknik komputasi yang kompleks dan algoritma untuk mengolah data, mengenal pola, dan membuat prediksi atau tindakan cerdas [30].

2.6 Pengolahan Citra Digital

Pengolahan Citra digital merupakan suatu pemrosesan citra yang memiliki tujuan untuk mendapatkan kualitas citra yang lebih baik. Teknik yang dilakukan pada proses pengolahan citra adalah dengan mentransformasikan citra menjadi citra lain [21].

Pengolahan citra merupakan cabang ilmu dalam *Artificial Intelligence* yang menggunakan objek citra dalam bentuk digital untuk penyelesaian kasusnya. Metode dalam citra dapat digunakan baik perhitungan matematis pada objek secara piksel ataupun geometris. Masing-masing objek citra memiliki nilai perbedaan yang dapat diperhitungkan secara matematis, sehingga menunjukkan ciri yang berbeda antara objek satu dengan yang lain. Penciri dari perbedaan setiap objek dapat ditentukan dari warna, tekstur, ataupun bentuk [31].

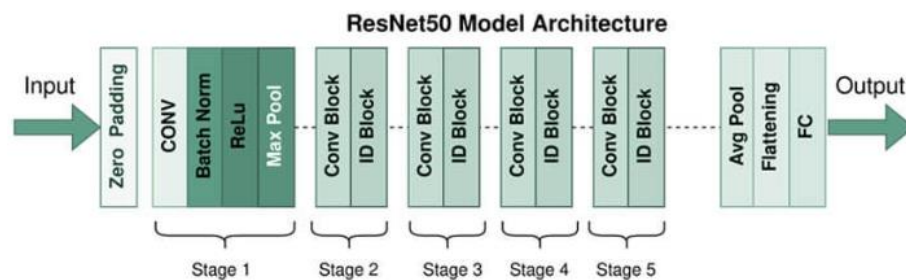
Citra merupakan suatu representasi (gambar), kemiripan, dan imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada

monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan [32].

2.7 Ekstraksi Fitur Gambar Menggunakan *Resnet 50*

Ekstraksi fitur merupakan proses penting dalam pengolahan citra digital yang bertujuan untuk menyaring informasi penting dari gambar dan mempresentasikannya dalam bentuk numerik. Salah satu metode modern yang banyak digunakan adalah dengan memanfaatkan arsitektur *Convolutional Neural Network (CNN)* seperti *ResNet50*, yang telah dilatih pada dataset besar seperti *ImageNet* [33].

ResNet-50 adalah model kecerdasan buatan canggih yang dirancang untuk mengklasifikasikan gambar. Model ini memiliki 50 lapis yang telah dilatih pada lebih dari 1 juta gambar dari database *ImageNet*, membuatnya sangat ahli dalam mengenali berbagai objek dan kategori [34].

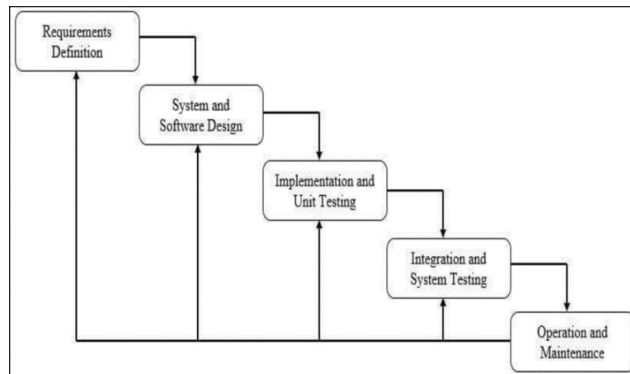


Gambar 2. 4 Arsitektur *ResNet-50* (Sumber : Hermanto et al., 2024)

2.8 Metode *Waterfall*

Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Dalam pengembangannya metode *waterfall* memiliki beberapa

tahapan yang runtut: requirement (analisis kebutuhan), design sistem (*system design*), Coding & Testing, Penerapan Program, pemeliharaan” [35].



Gambar 2. 5 Metode *Waterfall* (Sumber : Wahid, 2020)

2.9 Klasifikasi

Klasifikasi merupakan suatu teknik yang digunakan untuk mendapatkan model yang menjelaskan kelas data untuk melakukan prediksi kelas yang belum diketahui pada suatu objek yang diamati. Teknik ini melakukan manipulasi data dengan memberikan klasifikasi data baru untuk mendapatkan suatu aturan. Analisis Diskriminan dan Regresi Logistik merupakan metode klasifikasi yang biasanya digunakan dalam statistika, akan tetapi semakin berkembangnya teknologi maka dibutuhkan suatu metode analisis yang multifungsi untuk memberikan informasi dari suatu data yang sangat besar sehingga menjadi pengetahuan yang terorganisir. Pada machine learning metode klasifikasi yang sering digunakan diantaranya adalah *Classification and Regression Trees (CART)*, *Random Forest*, *Naïve Bayes*, *Support Vector Machines (SVM)*, dan sebagainya [21].

Klasifikasi merupakan suatu proses untuk menemukan fitur yang dapat membedakan atau mendeskripsikan kedalam kategori kelas data. Klasifikasi

bertujuan untuk memberikan perkiraan suatu objek yang belum diketahui labelnya kedalam suatu kelas [36].

Beberapa langkah yang digunakan dalam pengklasifikasian data diantaranya langkah training, dimana serangkaian data latih dilakukan analisis dengan tujuan untuk mendapatkan model klasifikasi, dan langkah klasifikasi, yaitu langkah yang digunakan untuk mengidentifikasi model klasifikasi, dimana model yang telah didapatkan digunakan untuk memprediksi kelas untuk data testing[37].

2.10 Website Kaggle.com

Kaggle.com adalah platform yang menyediakan ruang untuk berbagi ide, mendapatkan inspirasi, berkompetisi dengan ilmuwan data lainnya, mempelajari pengetahuan baru, serta mengeksplorasi berbagai aplikasi ilmu data dalam kehidupan nyata [38].

Website Kaggle merupakan *website* yang menyediakan kumpulan dataset dari berbagai bidang dengan tujuan untuk menunjang penelitian dari peneliti di berbagai belahan dunia [39].

2.11 Roboflow

Roboflow merupakan platform *web* yang memiliki fungsi berhubungan dengan kumpulan dataset. *Roboflow* adalah kerangka kerja pengembang computer vision untuk pengumpulan data yang lebih baik hingga prapemrosesan, dan teknik pelatihan model [40].

bahan ukuran gambar, anotasi, dan pengelolaan dataset secara kolaboratif. Selain itu, *RoboFlow* juga mendukung pengembangan model dengan teknik augmentasi otomatis dan konversi dataset ke format yang kompatibel dengan

berbagai framework *Machine Learning*, *RoboFlow* menjadi kerangka kerja yang membantu para pengembang meningkatkan efisiensi, kualitas, dan akurasi model *computer vision* secara keseluruhan [41].

2.12 *Google Collab*

Google Collab adalah sebuah *IDE* untuk pemrograman *Python* dimana pemrosesan akan dilakukan oleh server *Google* yang memiliki perangkat keras dengan performa yang tinggi. Dari sisi perangkat lunak, *Google Collab* telah menyediakan hampir sebagian besar pustaka (*library*) yang dibutuhkan. Dalam penelitian ini pustaka yang dibutuhkan adalah *Keras*, *TensorFlow*, *NumPy*, *Pandas*, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat *Matplotlib*. Bahkan seluruh versi disediakan, misalnya *TensorFlow* versi 1.x maupun 2.x tersedia, begitu juga versi *Python* yang mulai dari versi 2.x hingga 3.x [42].

Google Collab adalah *platform* yang memungkinkan pengguna untuk membuat dan berbagi kode secara *online* dengan mudah dan gratis dengan menggunakan bahasa pemrograman *python* yang menyediakan fitur built in *interpreter* [43].

Google Colaboratory, juga dikenal sebagai *Colab*, adalah *platform* komputasi awan gratis yang menyediakan lingkungan *notebook* tanpa *server* interaktif untuk menulis, berbagi, dan mengeksekusi kode. *Colab* beroperasi pada *notebook Jupyter*, yang kompatibel dengan bahasa pemrograman populer seperti *Python*, *R*, dan *Julia*. *Collab* menawarkan sumber daya komputasi yang kuat, termasuk akses ke mesin virtual jarak jauh berbasis *Linux* dengan unit

pemrosesan grafis (*GPU*) yang canggih hingga 12 jam. Hal ini menjadikannya platform yang nyaman untuk berbagai aplikasi, termasuk tugas pembelajaran mendalam dan pembelajaran mesin [44].

2.13 *Python*

Python adalah bahasa pemrograman yang fleksibel dan sederhana yang didefinisikan dalam dokumen-dokumennya sebagai. *Python* adalah sebuah bahasa pemrograman dinamis yang sering digunakan dalam pengembangan aplikasi pada berbagai domain. Hal ini memungkinkan suatu program ditulis dalam beberapa pendekatan sekaligus. Misalnya, antar muka grafis dibuat dalam bentuk orientasi objek, sedangkan pemrosesan dalam bentuk fungsional atau prosedural. Bahasa pemrograman *Python* memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak [45].

Python adalah bahasa pemrograman yang populer dan mudah untuk dipelajari. *Python* banyak digunakan dalam pengembangan perangkat lunak, kecerdasan buatan, pengembangan *web*, *machine learning*, dan analisis data. *Python* menyiapkan berbagai *Library*, seperti *NumPy* untuk komputasi numerik dan *Pandas* untuk analisis data sehingga memudahkan seseorang untuk melakukan tugas tertentu dengan cepat dan efisien [46].

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* juga didukung oleh komunitas yang besar.

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional [47].

2.13.1 *Library Python*

a. *Numpy*

Numpy merupakan singkatan dari *Numerical Python*. *Numpy* adalah sebuah *library* yang biasa digunakan untuk mengolah *array multidimensi*. Itu berarti *library* ini dapat membantu menjalankan komputasi-komputasi ilmiah yang ada di dalam program. *NumPy* adalah pustaka fungsi yang memungkinkan Anda melakukan banyak tugas manipulasi data umum dengan *Python* [48].

b. *Scikit-Learn*

Library *Scikit-learn* atau *Sklearn* adalah salah satu *library* pada bahasa *Python* yang dibangun berdasarkan *NumPy*, *SciPy*, dan *Matplotlib*. Library *Scikit-learn* bekerja dengan melakukan ekstraksi menggunakan hasil analisis yang berasal dari multivariate data, lalu *Scikit-learn* akan membantu melakukan *processing* data ataupun melakukan *training data* untuk kebutuhan *machine learning*. Library *Scikit-Learn* dapat menganalisa berbagai fungsi atau model analisa dalam bahasa pemrograman *Python*, seperti model klasifikasi, *clustering*, regresi berbasis model *machine learning*, dan proses-proses yang dapat dimanfaatkan pada tahap *feature engineering* seperti reduksi menggunakan *PCA* [49].

c. *OpenCV*

(*Open Source Computer Vision Library*) adalah pustaka *open-source* yang digunakan untuk pemrosesan gambar dan video secara *real-time* serta membangun aplikasi berbasis *computer vision*. *OpenCV* banyak digunakan dalam bidang seperti deteksi wajah, deteksi objek, *tracking*, dan segmentasi gambar. Dalam implementasinya di *Python*, *OpenCV* dimanfaatkan untuk membaca citra, mengubah format warna (misalnya dari *RGB* ke *grayscale*), serta melakukan ekstraksi fitur sebelum digunakan dalam proses klasifikasi menggunakan algoritma seperti *Support Vector Machine (SVM)* atau metode *deep learning* lainnya [50].

2.14 Penelitian Terdahulu

Untuk penelitian terdahulu dapat dilihat pada tabel berikut:

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti (Tahun)	Judul Penelitian	Metode	Kesimpulan
1	Rahayu, D. (2021)	Klasifikasi Citra Daun Tanaman Menggunakan Ekstraksi Ciri dan <i>SVM</i>	Ekstraksi Ciri (<i>GLCM</i>), <i>SVM</i>	<i>SVM</i> mampu mengklasifikasikan jenis daun dengan akurasi sebesar 87% berdasarkan tekstur dan bentuk.

2	Firmansyah, A. (2022)	Pengenalan Karakter Tulisan Tangan Bahasa Indonesia Berbasis <i>SVM</i>	<i>PCA</i> + <i>SVM</i>	<i>SVM</i> menunjukkan performa akurasi yang tinggi dalam klasifikasi karakter tulisan tangan, mencapai 90%.
3	Lestari, S. (2022)	Deteksi Dini Penyakit Daun Padi Berdasarkan Citra Digital dengan <i>SVM</i>	Color <i>Histogram</i> + <i>SVM</i>	Sistem berhasil mengklasifikasikan penyakit daun padi dengan akurasi 88%.
4	Hidayat, R. (2023)	Klasifikasi Citra Batik Menggunakan <i>SVM</i> dan Ekstraksi Ciri <i>Histogram</i> Warna	<i>Histogram</i> Warna + <i>SVM</i>	Metode <i>SVM</i> efektif dalam membedakan motif batik, dengan akurasi mencapai 91%.
5	Nugraheni, M. (2024)	Klasifikasi Jenis Sampah Organik dan Anorganik Menggunakan Pengolahan Citra dan	Ekstraksi Warna + Bentuk + <i>SVM</i>	Sistem mampu membedakan jenis sampah dengan akurasi 86%, membantu sistem

		<i>SVM</i>		pendaurulangan otomatis.
6	Nabila Zatadinia, Henny Dwi Bhakti (2024)	Identifikasi Jamur Yang Dapat Dikonsumsi Dan Beracun Menggunakan Metode <i>Logistic Regression</i>	<i>metode Logistic Regression</i>	Logistic Regression terbukti efektif dan akurat untuk mengidentifikasi jenis jamur yang bisa dikonsumsi dan yang beracun dengan menggunakan data ciri-ciri morfologi
7	Yusuf Amrozi, Dian Yuliati, Agung Susilo, Nur Novianto, Rikza Ramadhan (2022)	Klasifikasi Jenis Buah Pisang Berdasarkan Citra Warna dengan Metode <i>SVM</i>	<i>Support Vector Machine (SVM)</i>	Kombinasi warna, fitur tekstur, dan pengukuran bentuk menggunakan <i>SVM</i> menghasilkan hasil yang akurat. Eksperimen yang ditunjukkan

				mencapai akurasi klasifikasi yang signifikan sebesar 89,86%. Semua fitur yang diekstraksi direduksi dan disimpan dalam vektor fitur <i>PCA</i> (<i>Principal Component Analysis</i>). Untuk penelitian selanjutnya disarankan untuk menggunakan metode dan model warna lain sebagai perbandingan hasil keakuratan sistem
8	Aji Prasetya Wibawa, Triyanna	Studi Perbandingan Klasifikasi Daun	Hibrida <i>GLCM-SVM</i> dan	<i>GLCM</i> dan <i>SVM</i> dengan kernel linear

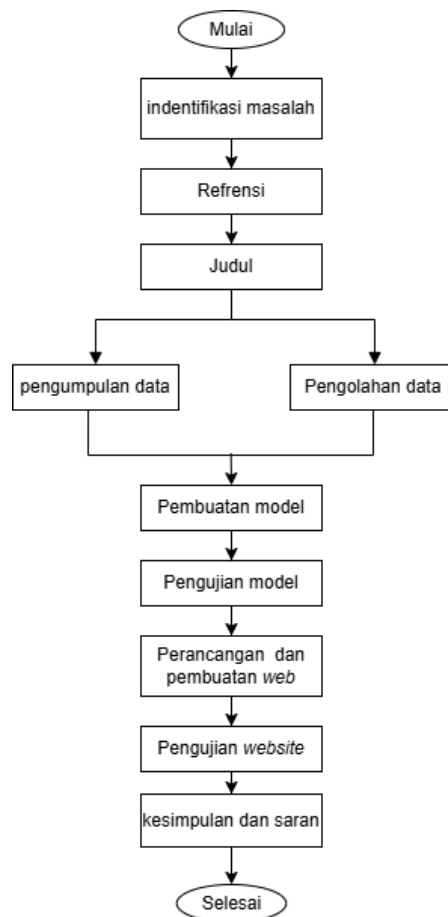
	Widyaningtyas, Haviluddin Cholisah Erman Hasihi, Ming Foey Teng, Herdianti Darwis	Herbal Menggunakan Hibrida <i>GLCM-SVM</i> dan <i>GLCM-CNN</i>	<i>GLCM-CNN</i>	menghasilkan akurasi tertinggi sebesar 96%, sedangkan kernel <i>sigmoid</i> hanya 35% dan kurang direkomendasikan. <i>GLCM-CNN</i> juga mencapai akurasi 96%, dengan hasil terbaik 98% pada kondisi terang. Kesimpulannya, kernel linear pada <i>SVM</i> paling optimal setelah ekstraksi fitur <i>GLCM</i> .
9	Suprihanto, Iwan Awaludin, Muhammad Fadhil, M.	Analisis Kinerja <i>ResNet-50</i> dalam Klasifikasi Penyakit pada Daun Kopi Robusta	embentukan model <i>ResNet-50</i>	Model <i>ResNet-50</i> berhasil mencapai akurasi 92,68% dan <i>F1-Score</i> 92,88% pada

	Andhika Zaini Zulfikor (2022)			klasifikasi penyakit daun kopi Robusta dalam kasus kelas <i>biner</i>, sementara pada <i>multiclass</i> akurasi mencapai 88,98% dan <i>F1-Score</i> 88,44%, menunjukkan bahwa kualitas dataset dan metode yang digunakan sangat mempengaruhi kinerja model.
--	----------------------------------	--	--	---

BAB 3

METODOLOGI PENELITIAN

Tahapan penelitian yang akan dilakukan dan penyelesaian masalah terhadap Implementasi algoritma *SVM (Support Vector Machine)* untuk klasifikasi jamur beracun dan tidak beracun. Adapun tahapan metodologi yang dilakukan selama penelitian dapat dilihat pada Gambar 3.1 yang mana merupakan proses yang dimulai dari studi literatur hingga diperoleh kesimpulan.



Gambar 3. 1 Tahapan Metodologi Penelitian

3.1 Identifikasi Masalah

Masalah utama yang diangkat dalam penelitian ini adalah kesulitan dalam membedakan jamur beracun dan tidak beracun secara visual oleh masyarakat awam. Hal ini menyebabkan tingginya risiko keracunan makanan akibat konsumsi jamur yang salah. Oleh karena itu, dibutuhkan suatu sistem klasifikasi berbasis komputer yang mampu memproses citra jamur dan menentukan tingkat bahayanya secara otomatis dan akurat.

3.2 Referensi

Penelitian ini didukung oleh studi literatur terhadap berbagai sumber seperti jurnal, artikel ilmiah, dan dokumentasi terkait klasifikasi jamur serta penerapan algoritma *machine learning*. Fokus utama adalah pemahaman tentang jenis-jenis jamur, fitur visual yang menjadi pembeda antara jamur beracun dan tidak beracun, serta tinjauan terhadap efektivitas algoritma *Support Vector Machine (SVM)* dalam klasifikasi berbasis citra. Referensi juga mencakup penelitian sebelumnya yang menggunakan pendekatan serupa untuk klasifikasi gambar objek biologis.

3.3 Judul

Berdasarkan permasalahan yang ditemukan dan hasil studi pustaka, maka ditetapkan judul penelitian yaitu “Implementasi Algoritma *Support Vector Machine (SVM)* untuk Klasifikasi Jamur Beracun dan Tidak Beracun”. Judul ini mencerminkan fokus penelitian pada penggunaan algoritma *SVM* untuk membangun model klasifikasi berbasis citra jamur guna membantu proses identifikasi yang lebih efisien dan akurat.

3.3.1 Pengumpulan Data

Sumber data diperoleh dari dataset publik yang tersedia secara daring melalui *platform* seperti *Kaggle* dan *Roboflow*. Data yang dikumpulkan terdiri atas gambar-gambar jamur yang telah dilabeli sesuai dengan jenis dan tingkat toksisitasnya. Pengumpulan data dilakukan dengan mempertimbangkan keragaman spesies, kualitas visual gambar, serta keseimbangan jumlah antara kedua kategori, agar model klasifikasi yang dibangun dapat berfungsi secara optimal. Secara rinci, data jamur yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Jamur Beracun: Terdiri dari 8 jenis jamur beracun, masing-masing berjumlah 500 gambar, sehingga total berjumlah 4.000 gambar. Adapun jenis jamur beracun yang digunakan antara lain: *Autumn Skullcap*, *Death Cap*, *Destroying Angels*, *False Morel*, *Poison Fire Coral*, *Paxillus Involutus*, *Omphalotus*, *Chlorophyllum*.
2. Jamur Tidak Beracun: Terdiri dari 8 jenis jamur tidak beracun, masing-masing juga berjumlah 500 gambar, dengan total 4.000 gambar. Jenis jamur tidak beracun yang digunakan yaitu: Jamur Kancing, Jamur Enoki, Jamur Kuping, *maitake*, Jamur shitake, grigit, Jamur Merang, Jamur Tiram

Total keseluruhan data yang digunakan adalah 8.000 gambar jamur. Data tersebut diunduh dalam format gambar (.jpg/.png) dan dilakukan seleksi awal

untuk memastikan resolusi gambar cukup baik, tidak buram, dan tidak mengandung noise berlebihan.

3.3.2 Pengolahan Data

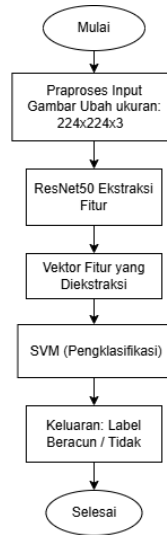
Pengolahan data merupakan tahap penting yang dilakukan sebelum data digunakan dalam proses pelatihan dan pengujian model klasifikasi. Langkah-langkah pengolahan data pada penelitian ini meliputi:

1. Seluruh gambar jamur yang telah dikumpulkan diubah ukurannya menjadi 224×224 piksel, setiap gambar diberi label berdasarkan kategorinya, yaitu beracun atau tidak beracun.
2. Dataset yang telah diproses kemudian dilakukan pembagian data kedalam *Set train* dan *validation* 80:20.
3. Vektor fitur hasil ekstraksi dari *ResNet 50* selanjutnya diklasifikasikan menggunakan algoritma *Support Vector Machine (SVM)* dengan kernel *Radial Basis Function (RBF)*. Model *SVM* dilatih menggunakan data latih dan kemudian dievaluasi menggunakan data validasi untuk mengukur performa model klasifikasi terhadap data yang belum dikenali sebelumnya

3.4 Pembuatan Model

Dataset yang digunakan dalam penelitian ini terdiri dari gambar jamur yang diklasifikasikan ke dalam dua kategori utama: jamur beracun dan jamur tidak beracun. Data diperoleh dari sumber terbuka seperti *Kaggle* dan *Roboflow*, dengan total 8.000 gambar, yaitu 4.000 gambar jamur beracun dan 4.000 gambar

jamur tidak beracun. Setiap kelas terdiri dari berbagai jenis jamur, yang masing-masing memiliki sekitar 500 gambar.



Gambar 3. 2 flowchart pembuatan model

3.4.1 Proses *Input* Gambar

Gambar jamur yang akan diklasifikasikan diproses terlebih dahulu. Proses preprocessing ini meliputi resize gambar ke ukuran 224x224 piksel dengan 3 channel warna (RGB). Ukuran ini disesuaikan agar kompatibel dengan input yang dibutuhkan oleh arsitektur *ResNet50*.

3.4.2 *Resnet50* Eksraksi Fitur

Gambar yang telah di-preprocessing kemudian dimasukkan ke dalam model *ResNet50* (tanpa lapisan klasifikasi akhir). Model ini digunakan sebagai *feature extractor* untuk menghasilkan representasi fitur dari gambar. Hasil ekstraksi fitur ini merupakan vektor numerik berdimensi tinggi yang merepresentasikan karakteristik penting dari gambar jamur.

3.4.3 Vektor Fitur Yang Diekstraksi

Fitur hasil ekstraksi dari *ResNet 50* akan dikumpulkan dalam bentuk vektor. Vektor inilah yang akan menjadi input bagi algoritma klasifikasi *SVM*.

3.4.4 SVM Klasifikasi

Algoritma Support Vector Machine (SVM) digunakan sebagai model klasifikasi. *SVM* akan mempelajari pola dari data fitur tersebut dan memisahkan antara dua kelas, yaitu jamur beracun dan tidak beracun, dengan menggunakan *hyperplane* terbaik. Dalam metode *SVM*, proses klasifikasi dilakukan menggunakan fungsi keputusan sebagai berikut:

$$f(x) = \sum_{i=1}^n \alpha_i \cdot y_i \cdot k(x_i, x) + b \dots \dots \dots (4)$$

- a. x_i : vektor fitur dari data latih ke- i
- b. x : vektor fitur dari data uji
- c. α_i : Lagrange multiplier dari hasil pelatihan
- d. y_i : label kelas dari data latih (misalnya: -1 untuk tidak beracun, 1 untuk beracun)
- e. $K(x_i, x)$: kernel function antara data latih dan data uji
- f. b : bias atau intersep dari model

3.4.5 Keluaran Label (Beracun/Tidak Beracun)

Setelah proses klasifikasi, sistem akan memberikan output berupa label:

Poisonous (beracun), *Non-Poisonous* (tidak beracun) Output ini ditampilkan kepada pengguna sebagai hasil dari proses klasifikasi.

3.5 Pengujian Model

Pada tahapan pengujian model, model yang telah dilatih sebelumnya diuji untuk mengevaluasi kinerjanya. Pengujian ini mencakup:

1. Evaluasi akurasi pada set data validasi.

2. Penggunaan metrik tambahan seperti *Accuracy*, *Precision*, *Recall* dan *F1-Score* bertujuan untuk memberikan gambaran yang lebih lengkap tentang performa model secara keseluruhan.
3. Analisis hasil pengujian untuk mengidentifikasi bagian yang perlu diperbaiki atau dioptimalkan.

Pengukuran kinerja menggunakan *Confusion Matrix* yaitu dengan menggunakan 4 perhitungan berikut ini:

1. *Accuracy* adalah rasio antara jumlah prediksi yang benar dengan total data yang diuji.

$$Accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negatif} \quad (5)$$

2. *Precision* adalah perbandingan antara jumlah prediksi positif yang benar dengan total prediksi positif yang dihasilkan.

$$Precision = \frac{true\ positive}{false\ positive + true\ positive} \dots \dots \dots (6)$$

3. *Recall* adalah perbandingan antara jumlah prediksi positif yang benar dengan total data positif yang sebenarnya.

$$Recall = \frac{true\ positive}{false\ negative + true\ positive} \dots \dots \dots (7)$$

4. Selanjutnya, nilai *F1-Score* dihitung sebagai rata-rata yang dibobotkan antara *Precision* dan *Recall*.

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \dots \dots \dots (8)$$

Hasil dari tahapan pengujian model ini berupa laporan performa model berdasarkan metrik evaluasi yang nantinya ketika performa model sudah mencapai akurasi tertinggi, maka model siap diproses lebih lanjut.

3.6 Perancangan dan Pembuatan *Website*

ada tahap ini, sistem klasifikasi citra jamur beracun dan tidak beracun mulai dirancang dan dikembangkan. Proses ini mencakup berbagai aspek perancangan, mulai dari pengumpulan dan preprocessing data hingga integrasi model klasifikasi menggunakan algoritma *Support Vector Machine (SVM)*.

Sistem ini diimplementasikan dalam bentuk *website* berbasis antarmuka grafis (*Graphical User Interface/GUI*), yang memungkinkan pengguna mengunggah gambar jamur dan memperoleh hasil klasifikasi secara otomatis. Desain antarmuka dibuat agar mudah digunakan dan informatif, dengan menampilkan hasil klasifikasi serta label (beracun/tidak beracun) dari gambar yang dimasukkan oleh pengguna.

3.6.1 Perancangan *website*

Perancangan *website* dilakukan dengan mempertimbangkan aspek fungsionalitas, pengalaman pengguna (*user experience*), serta efisiensi sistem dalam mendukung proses klasifikasi gambar jamur. *Website* ini bertujuan sebagai antarmuka pengguna yang dapat digunakan untuk mengunggah gambar jamur, melakukan klasifikasi, dan menampilkan hasil secara cepat dan informatif. Berikut adalah langkah-langkah utama dalam proses perancangan *website*:

1. Analisis Kebutuhan

- a) Menentukan fitur utama yang akan tersedia pada *website*, antara lain:

Fitur unggah gambar jamur, Tombol proses klasifikasi Hasil klasifikasi berupa label “beracun” atau “tidak beracun”.

- b) Memastikan *website* dapat memberikan pengalaman penggunaan yang dapat diakses dengan mudah oleh berbagai jenis perangkat, baik *desktop* maupun *mobile*

2. Perancangan *UI/UX*

- a) Mendesain tampilan antarmuka menggunakan *wireframe* dan *mockup* untuk memastikan navigasi yang mudah, tata letak yang rapi, dan tampilan hasil klasifikasi yang jelas.
- b) Memilih skema warna yang sesuai dan elemen desain yang sederhana agar pengguna merasa nyaman dalam mengoperasikan sistem. Desain dibuat dengan gaya minimalis namun informatif.

3. Perancangan Arsitektur Sistem

- a) *Frontend*: Antarmuka pengguna dibangun menggunakan Streamlit, yaitu *framework* berbasis *Python* yang memungkinkan pengembangan antarmuka *web* secara cepat dan interaktif. *Streamlit* dipilih karena mendukung pemrosesan gambar secara langsung dan memudahkan integrasi dengan model klasifikasi berbasis *Python*. Penggunaan *Streamlit* juga mempermudah visualisasi hasil klasifikasi dan interaksi pengguna melalui tombol unggah gambar, tampilan prediksi, serta antarmuka yang responsif dan minimalis.
- b) *Backend*: Menggunakan bahasa pemrograman *Python* dengan dukungan pustaka seperti *TensorFlow/Keras* dan *Scikit-learn*, serta *framework Streamlit* sebagai *backend* interaktif berbasis *web*. *Backend* juga mengelola alur integrasi model klasifikasi serta mengembalikan

hasil klasifikasi secara langsung ke antarmuka pengguna melalui *platform Streamlit*

3.6.2 Hasil Pembuatan Website

Setelah tahap perancangan dan pembuatan website selesai, dihasilkan sebuah sistem klasifikasi gambar jamur berbasis web dengan fitur utama sebagai berikut:

1. Antarmuka pengguna yang intuitif untuk mengunggah gambar jamur dan menerima hasil klasifikasi berupa label “beracun” atau “tidak beracun”.
2. *Backend* yang mampu memproses input gambar secara otomatis, menghasilkan klasifikasi gambar secara *real-time*.
3. *Website* dapat diakses melalui berbagai perangkat (*desktop* maupun *mobile*) setelah di-*deploy* ke server atau menggunakan *Streamlit Community Cloud*.

Dengan adanya *website* ini, pengguna dapat dengan mudah melakukan klasifikasi gambar jamur secara praktis, cepat, dan akurat tanpa memerlukan pemahaman teknis yang mendalam terhadap metode *machine learning* yang digunakan.

3.7 Pengujian Website

Pengujian sistem ini dilakukan dengan cara menggunakan *Black Box*. Dalam Pengujian *Black Box* ini berfokus pada perangkat lunak untuk mendapatkan hasil yang baik, apabila terjadi *error* atau tidak sesuai dengan tujuan yang diharapkan maka dilakukan penganalisa sistem kembali hingga tidak ditemukan *error*.

3.8 **Black Box Testing**

Black box testing merupakan pengujian kualitas perangkat lunak yang berfokus pada fungsionalitas perangkat lunak, *pengujian black box testing* bertujuan untuk menemukan fungsi yang tidak benar kesalahan antarmuka kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi, dalam pengujian *black box testing* digunakan alat untuk pengumpulan data yang disebut dengan *user acceptance test*, dokumen ini terdiri deskripsi indikator dari prosedur-prosedur pengujian fungsionalitas dari perangkat lunak.

3.9 **Kesimpulan dan Saran**

Tahapan terakhir adalah menarik kesimpulan dari hasil penelitian yang didapatkan dalam Implementasi algoritma *SVM (Support Vector Machine)* untuk klasifikasi jamur beracun dan tidak beracun, pada tahapan ini juga berisikan saran peneliti bagi pembaca untuk melakukan pengembangan terhadap penelitian ini kedepannya.