

## **BAB II**

### **KAJIAN PUSTAKA**

#### **A. Kajian Teoritis**

##### **1. Identifikasi**

Dalam Kamus Besar Bahasa Indonesia, identifikasi mempunyai arti menentukan atau menetapkan identitas seseorang, benda, dan sebagainya (Ariesta, 2021). Identifikasi merupakan proses atau kegiatan untuk mengenali, menetapkan, atau membedakan sesuatu berdasarkan karakteristik atau ciri-ciri khas yang dimilikinya. Proses identifikasi dapat dilakukan dalam berbagai konteks dan bidang, termasuk ilmu pengetahuan, teknologi, keamanan, dan administrasi.

##### **2. Jamur**

Jamur adalah tubuh buah (*Basidiomycota*) yang muncul di permukaan seperti tempat lembab dan berkembang dari kelompok jamur (*Basidiomycota*) berbentuk seperti payung dengan batang dan kepala bulat. Kata jamur berasal dari bahasa latin jamur. Jamur (jamur) berkembang biak secara aseksual maupun nonseksual, meliputi pembentukan spora, tunas, dan fragmentasi (Sarimole & Diadi, 2022).

Jamur mempunyai keistimewaan seperti kaya nutrisi, kaya serat, mengandung senyawa-senyawa yang berkhasiat untuk mencegah penyakit serta harga yang cukup murah. Masyarakat umumnya mengenal jamur sebagai bahan makanan yang memiliki rasa yang enak, bertekstur kenyal

seperti daging tetapi banyak masyarakat yang belum mengetahui manfaat jamur bagi kesehatan (Daryanti *et al.*, 2022).

### 3. Pengolahan Citra Digital

Menurut Ratna (2020) Pengolahan citra digital, juga dikenal sebagai pengolahan citra digital, adalah bidang ilmu yang mempelajari metode untuk mengolah gambar, gambar yang dimaksud dapat berupa gambar yang tidak bergerak (seperti foto) atau gambar yang diam (seperti video yang direkam). Di sisi lain, "digital" mengacu pada proses pengolahan gambar atau foto digital yang dilakukan melalui penggunaan komputer. RGB adalah singkatan dari *Red*, *Green*, dan *Blue*, tiga warna utama yang biasanya digunakan sebagai referensi untuk warna lainnya. Kami dapat mengubah warna dari basis RGB menjadi kode angka yang dapat digunakan secara universal. Pengolahan warna RGB menjadi mudah karena model warna yang sama disimpan di komputer.

Citra digital adalah citra dua dimensi yang dibuat dari citra analog dua dimensi yang berkesinambungan melalui proses pengambilan sampel. Gambar serupa dibagi menjadi  $N$  baris dan  $M$  kolom menjadi gambar diskrit. Gambar digital adalah gambar yang dapat diolah oleh komputer. Yang tersimpan di komputer hanyalah angka-angka yang mewakili intensitas disetiap piksel. Karena berbentuk data digital, maka gambar digital dapat diolah oleh komputer (Munantri *et al.*, 2020).

#### 4. *Pre-Processing*

Data *preprocessing* merupakan tahap melakukan proses pengolahan data awal. Pada tahap ini data diolah dengan tujuan untuk menghindari interupsi (*noise*) atau inkonsistensi data. Dalam penelitian ini, *preprocessing* yang dilakukan pada citra diantaranya:

- a. *Resizing, resizing* pada citra dilakukan dengan mengubah ukuran menjadi 224x223 *pixel*.
- b. Normalisasi, normalisasi pada citra dilakukan ke dalam rentang 0 hingga 1 membagi nilai *pixel* dengan 225.
- c. *Augmentasi* data, augmentasi dilakukan dengan menambah variasi pada data dengan membuat memodifikasi rotasi dan pemotongan.

#### 5. **Transfer Learning**

*Transfer learning* adalah metode atau teknik yang mengambil jaringan yang telah dilatih sebelumnya dan menggunakannya sebagai titik awal untuk mempelajari tugas atau masalah baru. Dengan menggunakan *transfer learning*, fitur yang dipelajari dapat ditransfer ke tugas baru menggunakan kumpulan data yang lebih kecil selama pelatihan, sehingga mempercepat prosesnya. Karena jaringan telah menjalani pelatihan sebelumnya dan belajar mengekstrak fitur yang berbeda, jaringan baru akan menjalankan tugasnya dengan lebih akurat (Faturrahman *et al.*, 2023).

*Transfer Learning* mengambil fitur yang dihasilkan dan menerapkannya pada tugas baru menggunakan beberapa kumpulan data selama pelatihan sehingga prosesnya selesai lebih cepat. Keuntungan

menggunakan *transfer learning* adalah pertama, hasil yang diperoleh dengan menggunakan *transfer learning* akan lebih baik dibandingkan tanpa menggunakan *transfer learning* (Wiratama *et al.*, 2023).

## 6. Metode Citra Digital

### a. *Convolutional Neural Network*

#### 1) *Pengertian Convolutional Neural Network*

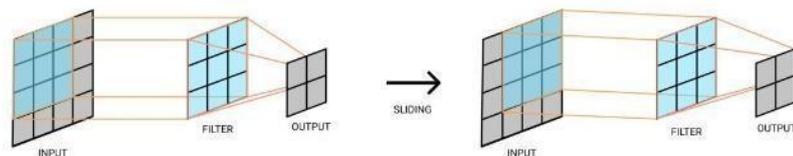
*Convolutional Neural Network* (CNN) adalah cabang pembelajaran mesin yang mengajarkan komputer untuk melakukan hal-hal mirip manusia. Dibandingkan *multi-layer perceptron* (MLP), CNN dinilai lebih baik karena memiliki kemampuan menyimpan informasi spasial dari data gambar. Pada saat klasifikasi data di CNN digunakan fungsi aktivasi (*softmax*) dan koneksi penuh yang keluarannya berupa klasifikasi. Struktur CNN meliputi masukan, proses ekstraksi fitur, proses klasifikasi, dan keluaran. CNN beroperasi secara hierarki, sehingga keluaran konvolusi pertama digunakan untuk konvolusi berikutnya (Syahputra & Wibowo, 2020).

Dengan menggabungkan proses ekstraksi ciri dan klasifikasi, CNN adalah salah satu model pembelajaran *deep learning* yang mampu melatih sistem dengan banyak data. *AlexNet* adalah salah satu model CNN. CNN dapat memproses gambar dua dimensi dan data video dalam struktur grid (Azizah, 2023). Menurut Peryanto *et al.*, (2020) *Convolutional Neural Network*

terdiri dari 4 jenis komponen *diantaranya convolutional layer, pooling layer, fully connected layer* dan *drop out*.

## 2) *Convolution Layer*

*Convolution Layer* adalah hasil dari proses *convolution* yang mana proses tersebut melakukan pengenalan pola, pola yang dimaksud diantaranya ialah *edges, shapes, colors, surfaces* pada suatu gambar. Pola yang dihasilkan bergantung pada channel, channel tersebut berupa matriks dengan nilai awal didalamnya berupa nilai arbitrary, lalu ukuran channel definisikan nilai baris dan kolomnya.



Gambar 2. 1 *Convolutional Layer*  
Sumber: Ersyad *et al.*, (2020)

Seperti yang terlihat pada 2.1 sebuah channel akan melakukan proses *speck item* pada *neuron* yang berada didalam lingkup channel, lalu melakukan proses *sliding* untuk menghitung nilai selanjutnya, hasil dari proses tersebut akan menjadi *output* pada proses *convolution*, proses ini akan terus dilakukan selama ada *input* yang diterima (Ersyad *et al.*, 2020).

### 3) *Pooling Layer*

Menurut Husna *et al.*, (2022) *Pooling layer* adalah lapisan di mana pengurangan ukuran gambar terjadi setelah lapisan terkontaminasi. Ada dua jenis *pooling layer*, yaitu *layer pooling* maksimum dan *layer pooling avarage*. Dalam *layer max pooling*, *pooling* membagi keluaran jaringan konvolusional menjadi banyak grid kecil, dimana nilai maksimum dari setiap grid akan dimasukkan ke dalam matriks gambar yang telah dikurangi oleh *pooling layer*. Berikut adalah contoh aksi *max pooling layer* yang ditunjukkan pada gambar berikut:

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Gambar 2. 2 Operasi *max layer*  
Sumber: Husna *et al.*, (2022)

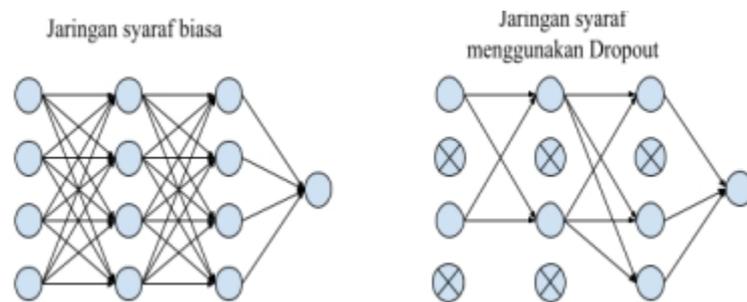
### 4) *Fully Connected Layer*

*fully connected layer* merupakan lapisan yang biasa digunakan dalam aplikasi *MLP* dan dimaksudkan untuk melakukan transformasi dimensi data untuk memungkinkan klasifikasi linier. Setiap *neuron* di lapisan konvolusional harus terlebih dahulu diubah menjadi data satu dimensi sebelum dapat dimasukkan ke dalam *fully connected layer*. Hal ini mengakibatkan data informasi spasial hilang dan tidak dapat dipulihkan, *fully connected layer* hanya dapat diterapkan di ujung

jaringan. Lapisan konvolusional dengan ukuran kernel 1 x 1 menjalankan fungsi yang sama seperti *fully connected layer* tetapi tetap mempertahankan karakteristik spasial datanya (Cahya *et al.*, 2021).

### 5) *Dropout*

*Dropout* adalah teknik regularisasi jaringan saraf di mana sejumlah *neuron* dipilih secara acak dan tidak digunakan selama pelatihan. Kita dapat mengatakan bahwa *neuron-neuron* ini dibuang secara acak. Artinya, kontribusi *neuron* yang dihapus akan dihentikan selama jaringan dan bobot baru tidak akan diterapkan ke neuron selama propagasi mundur (Peryanto *et al.*, 2020). Penggunaan *dropout* dapat dilihat pada gambar 2.4 sebagai berikut:



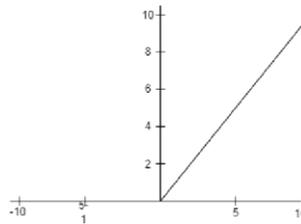
Gambar 2. 3 Contoh penggunaan *dropout*  
Sumber: Kholik, (2021)

### 6) *Activation Function*

*Activation function* dihitung setelah operasi konvolusi selesai. Pada *convolution neural network* (CNN), fungsi aktivasi yang sering digunakan adalah *rectified linear units* (ReLU) dan *softmax*.

a) ***Rectified Linear Units (ReLU)***

*ReLU* merupakan fungsi aktivasi yang membatasi *output* yang dihasilkan oleh *pooling layer*, yang dapat mempercepat training dengan mengalokasikan nilai negatif ke nol dan menyimpan nilai positif di *ReLU*. Berikut grafik aktivasi *ReLU* yang ditunjukkan pada gambar di bawah ini (Husna *et al.*, (2022)).



Gambar 2. 4 Aktivasi *ReLU*  
Sumber: Husna *et al.*, (2022)

b) ***Softmax***

aktivasi *softmax* merupakan aktivasi yang memetakan nilai vektor berjumlah satu. *Softmax* merupakan bagian dari tahap klasifikasi dan fungsi aktivasi ini digunakan untuk model multiklasifikasi dengan lebih dari dua kelas (Santosa *et al.*, (2023)).

*Softmax* adalah fungsi yang mengambil *input* vektor bilangan real  $K$  dan menormalkannya menjadi distribusi probabilitas yang terdiri dari  $K$  probabilitas. Sebelum menerapkan *softmax*, beberapa komponen vektor mungkin negatif atau lebih besar dari satu; dan mungkin tidak berjumlah 1, namun setelah diterapkan *softmax*, setiap

komponen akan berada pada rentang (0 - 1) dan komponen tersebut berjumlah 1, sehingga dapat diartikan sebagai probabilitas. Selain itu, komponen masukan yang lebih besar akan menghasilkan probabilitas yang lebih besar. *Softmax* umumnya digunakan dalam jaringan saraf. *Softmax* digunakan untuk menentukan output yang sesuai. Berikut rumus yang digunakan dalam aktivasi (Pangestu *et al.*, 2020).

$$p(x) = \frac{e^x}{\sum_{k=1}^k e^x}$$

#### 7) *Epoch*

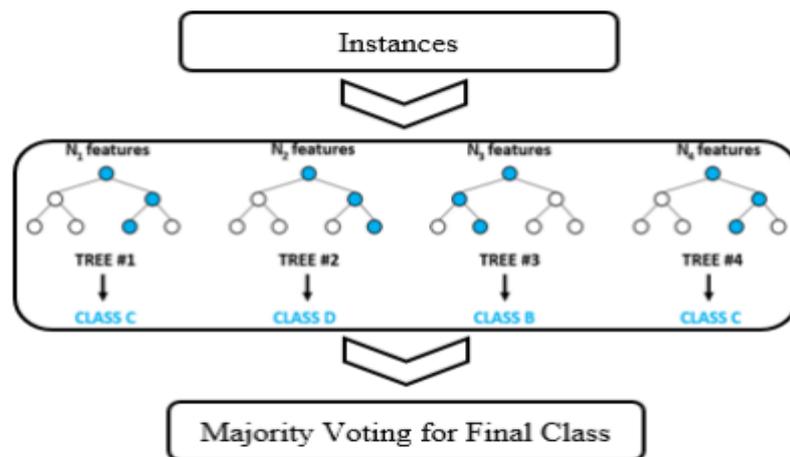
*Epoch* adalah representasi numerik dari siklus lengkap selama pelatihan pada kumpulan data. Semakin tinggi nilai epoch maka semakin tinggi skor akurasi dan semakin rendah nilai loss. Proses ini berhenti jika nilai kerugian dan nilai akurasi tidak berubah secara signifikan (Kurnia & Wibowo, 2021).

### b. *Random Forest*

#### 1) *Pengertian Random Forest*

*Random forest* adalah metode pembelajaran mesin yang menggunakan konsep terbimbing dalam membuat kelas klasifikasi. Algoritma ini menggabungkan prediksi berdasarkan beberapa pohon keputusan. Dengan menggunakan metode ini, Anda bisa mendapatkan model pembelajaran dari pola atau tren pada dataset. Informasi ini kemudian akan digunakan sebagai masukan untuk tahap pengujian untuk mengukur kinerja

peramalan sistem. *random forest* adalah algoritma pembelajaran mesin terawasi yang secara iteratif mengimplementasikan konsep pohon keputusan untuk membentuk hutan. Di bawah ini berikut ilustrasi *random forest* dapat dilihat pada gambar 2.5 berikut (Saadah & Salsabila, 2021):



Gambar 2. 5 Ilustrasi *Random Forest*  
Sumber: Widjiyati, (2021)

## 2) *Bagging*

*Bagging* adalah metode agregasi yang populer, sedangkan *Random Forest* (RF) adalah metode agregasi yang relatif baru dalam pohon keputusan, yang merupakan evolusi dari metode *Bagging*. Berbeda dengan *Bagging*, *Random Forest* dikembangkan dengan ide untuk menambahkan lapisan pada proses pengambilan sampel acak saat mengantongi. Jadi, tidak hanya sampel data yang diambil secara acak untuk membentuk pohon klasifikasi, tetapi variabel independen juga diambil sebagian secara acak dan kemudian dipilih sebagai pengklasifikasi

terbaik saat menentukan pohon klasifikasi, sehingga diharapkan dapat memberikan prediksi yang lebih akurat.

### 3) *Regresi Random Forest*

*Regresi Random Forest* (RF) adalah metode prediksi *maintenance* yang populer dalam machine learning. Namun diperlukan analisis mendalam untuk memahami kelebihan, keterbatasan, dan kinerja RF dalam memprediksi waktu perawatan optimal berdasarkan berbagai faktor yang mempengaruhi. Dalam studi ini, kami melakukan tinjauan literatur sistematis untuk mengidentifikasi tren terkini seputar penggunaan RF dalam konteks ini. Tujuan utama kami adalah untuk membangun basis pengetahuan yang kuat dan memberikan rekomendasi yang tepat untuk penelitian masa depan di bidang pemeliharaan prediktif (Yasa et al., 2024).

### 4) *Ensemble Learning*

*Ensemble learning* adalah teknik klasifikasi yang menggunakan beberapa pengklasifikasi. Dalam klasifikasi konvensional, hanya satu pengklasifikasi yang digunakan. Pengklasifikasi tujuan umum ini digunakan untuk memprediksi kelas objek yang kelasnya tidak diketahui. Proses prediksi dilakukan dengan menggunakan mekanisme voting, dimana objek akan diberi label kelas dengan suara terbanyak (Hidayat, 2021)

*Ensemble Learning* adalah model *machine learning* di mana beberapa model sering disebut ("*weak learners*") dilatih untuk memecahkan masalah yang sama dan digabungkan untuk mencapai hasil yang lebih baik. Hipotesis utamanya adalah dengan menggabungkan model yang lemah, kita dapat memperoleh model yang lebih akurat (Cendani & Wibowo, 2022).

#### 5) *Variable Importance*

*Variabel importance* adalah konsep dalam analisis statistik dan pembelajaran mesin yang digunakan untuk menentukan seberapa besar pengaruh atau kontribusi masing-masing variabel prediktor terhadap model prediktif. Tujuannya adalah untuk mengidentifikasi variabel mana yang paling signifikan dalam mempengaruhi hasil atau prediksi model. Pemahaman tentang pentingnya variabel membantu dalam interpretasi model, meningkatkan performa model, dan dalam pengambilan keputusan berbasis data.

#### 7. *Confusion Matrix*

*Confusion Matrix* merupakan matriks yang memuat informasi terkait perbandingan hasil yang diberikan sistem pada keputusan klasifikasi akhir. Pada matriks konfusi terdapat 4 suku dari proses pengambilan keputusan untuk hasil akhir matriks konfusi, yaitu positif benar, negatif benar, positif palsu, negatif palsu (Wiratama, *et al.*, 2023).

*Confusion Matrix* adalah sebuah metode perhitungan performansi untuk permasalahan klasifikasi, metode ini sangatlah berguna untuk mengukur nilai *Review*, *Exactness*, *Exactness*, *F-Measure* dan yang terpenting *Confusion Matrix* dapat memberikan informasi perihal jumlah *course* prediksi yang diberikan terlepas dari benar atau salah prediksi tersebut (Ersyad *et al.*, 2020).

## 8. *Python*

*Python* adalah bahasa pemrograman berorientasi objek dinamis yang dapat digunakan untuk berbagai jenis pengembangan perangkat lunak. *Python* memberikan dukungan kuat untuk integrasi dengan bahasa dan alat pemrograman lain. *Python* hadir dengan perpustakaan standar yang dapat diperluas dan dapat dipelajari hanya dalam beberapa hari. Bahasa pemrograman yang ditafsirkan untuk tujuan umum dengan filosofi desain yang berfokus pada keterbacaan kode. *Python* dianggap sebagai bahasa yang menggabungkan kemampuan, kemungkinan, dengan sintaks kode yang sangat jelas dan dilengkapi dengan perpustakaan standar yang luas dan lengkap (Nugroho *et al.*, 2020).

## 9. *Keras*

*Keras* adalah perpustakaan yang bekerja dengan blok penyusun jaringan saraf seperti lapisan, target, fungsi aktivasi, dan pengoptimal. *Keras* juga memiliki banyak fitur untuk bekerja dengan gambar dan gambar teks yang sangat berguna saat mengkode jaringan saraf dalam. Selain

jaringan saraf standar, *Keras* juga mendukung jaringan saraf konvolusional dan berulang (Alfarizi *et al.*, 2023).

#### 10. *TensorFlow*

*Tensorflow* adalah perpustakaan populer untuk membuat model pembelajaran mesin. *TensorFlow* mendukung banyak toolkit berbeda untuk membuat model pada berbagai tingkat abstraksi (Alfarizi *et al.*, 2023).

*TensorFlow* adalah pustaka perangkat lunak pembelajaran mesin sumber terbuka dan gratis. *TensorFlow* didedikasikan untuk pelatihan dan inferensi jaringan neural dalam, tetapi dapat digunakan untuk berbagai tugas. *Tensorflow* adalah pustaka matematika simbolis yang menggunakan dataflow dan pemrograman (Rahma *et al.*, 2021).

#### 11. *Flask*

*Flask* adalah kerangka web yang ditulis dengan *Python* dan diklasifikasikan sebagai jenis kerangka mikro. *Flask* bertindak sebagai kerangka aplikasi dan antarmuka situs *web*. Dengan menggunakan *Flask* dan bahasa *Python*, pengembang dapat membuat situs *web* terstruktur dan mengelola pengoperasian situs *web* dengan lebih mudah (Nugroho *et al.*, 2020).

*Flask* adalah *microframework* yang diluncurkan oleh Armin Ronacher. *Flask* jauh lebih ringan dan cepat karena *Flask* dibuat dengan ide untuk menyederhanakan kerangka inti seminimal mungkin. Dengan slogan “*web development, one drop at a time*”, *Flask* dapat membantu kita

membuat *website* dengan sangat cepat, bahkan dengan perpustakaan sederhana sekalipun (M. G. L. Putra & Putera, 2019).

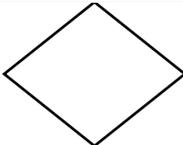
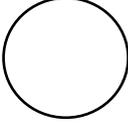
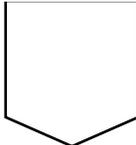
## **12. *Extreme Programming***

*Extreme Programming* (XP) adalah cara yang ringan, efisien, berisiko rendah, fleksibel, dapat diprediksi, ilmiah, dan menyenangkan untuk mengembangkan perangkat lunak. *Extreme programming* (XP) disusun dan dikembangkan untuk memenuhi kebutuhan spesifik pengembangan perangkat lunak oleh tim kecil dalam menghadapi persyaratan yang tidak jelas dan berubah. *Extreme Programming* adalah salah satu metodologi pengembangan perangkat lunak *Agile*. XP menyediakan nilai-nilai dan prinsip-prinsip untuk memandu perilaku tim. *Extreme Programming* menyediakan praktik inti yang spesifik. (Tutorials Point, 2015)

## **13. *Flowchart***

Menurut Rosaly & Prasetyo (2020) *Flowchart* atau sering disebut dengan diagram alir merupakan suatu bagan yang merepresentasikan algoritma atau langkah-langkah instruksi yang berurutan dalam sistem. Menurut Septima *et al.*, (2024) Selain itu flowchart juga dapat digunakan untuk mempresentasikan ataupun merancang program.

Tabel 2. 1 Flowchart

NO	Gambar	Simbol	Keterangan
1		<i>Terminator</i>	Simbol yang digunakan untuk menunjukkan titik awal atau akhir dari sebuah program
2		<i>Garis Alir</i>	Simbol yang digunakan dalam menggambarkan alur atau aliran program
3		<i>Proses</i>	Simbol yang digunakan untuk memproses perintah pengolahan data
4		<i>Input Output Data</i>	Simbol yang digunakan untuk menggambarkan proses memasukan dan mengeluarkan data
5		<i>Decision</i>	Simbol yang digunakan untuk memberikan pilihan antara ya atau tidak
6		<i>On Page Connector</i>	Simbol yang menghubungkan bagian <i>Flowchart</i> dalam halaman sama yang berbeda-beda
7		<i>Off Page Connector</i>	Simbol yang menghubungkan bagian <i>flowchart</i> dalam halaman yang berbeda

Sumber: Zalukhu *et al.*, (2023)

Setelah memahami definis di atas dapat disimpulkan bahwa *flowchart* digunakan sebagai bukti dokumenter untuk menjelaskan kepada pengembang gambaran logis dari sistem yang sedang dibangun.

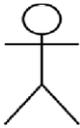
## 14. UML

*Unified Modeling Language* (UML) adalah bahasa pemodelan perangkat lunak standar yang mendukung penulisan rencana perangkat lunak (*Pressman*). UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan bagian-bagian tertentu dari suatu sistem dalam perangkat lunak (*Sumiati et al., (2021)*).

### a. Use Case

Use case adalah dialog antara aktor dan sistem yang menggambarkan fungsionalitas yang disediakan oleh sistem. Use case adalah pola atau bentuk perilaku yang mewakili suatu sistem. Setiap use case merupakan serangkaian transaksi terkait dari aktor dan sistem dalam sebuah dialog (*Risti, 2023*). Simbol-simbol *use case* diagram dapat dilihat sebagai berikut:

Tabel 2. 2 *Use case*

No	Gambar	Simbol	Keterangan
1	 Actor	<i>Actor</i>	Hubungan dimana objek anak ( <i>descendent</i> ) mewarisi struktur data dari objek induk ( <i>ancestor</i> ) yang berda di atasnya.

2		<i>Dependency</i>	Ketentuan yang secara eksplisit menunjukkan bahwa suatu <i>use case</i> menggabungkan <i>use case</i> sumber ke dalamnya
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) mewarisi struktur data dari objek induk ( <i>ancestor</i> ) yang berda diatasnya.
4		<i>Include</i>	Ketentuan yang menyatakan bahwa <i>use case</i> memperluas perilaku <i>use case</i> sumber pada titik tertentu.
5		<i>Extands</i>	Ketentuan yang menyatakan bahwa <i>use case</i> memperluas perilaku <i>use case</i> sumber pada titik tertentu.
6		<i>Association</i>	Koneksi penghubung antara satu objek dengan objek lainnya.

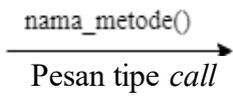
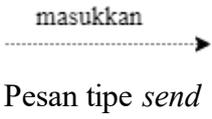
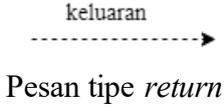
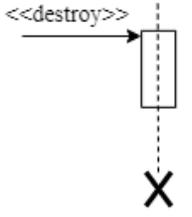
Sumber: Hutabri & Putri (2019)

### b. *Sequence Diagram*

Menurut Musthofa & Adiguna (2022) *Diagram sequence* adalah diagram yang menunjukkan kolaborasi dinamis antara beberapa objek. Kegunaannya adalah untuk menampilkan urutan pesan yang dikirim antar objek serta interaksi antar objek.

Tabel 2. 3 Squance Diagram

Simbol	Deskripsi
<p data-bbox="651 472 735 539">Aktor nama aktor Atau</p> <div data-bbox="616 539 778 618" style="border: 1px solid black; padding: 2px; text-align: center;"> <p data-bbox="651 566 743 589"><u>Nama aktor</u></p> </div> <p data-bbox="627 640 762 663">Tanpa waktu aktif</p> 	<p data-bbox="927 450 1362 595">Merupakan orang atau sistem atau proses di luar sistem yang dibuat, yang berhubungan dengan sistem yang dibuat.</p>
 <p data-bbox="552 994 831 1028"><i>Lifeline</i> (Garis hidup)</p>	<p data-bbox="927 786 1362 887">Merupakan garis hidup objek yang menerangkan kehidupan objek</p>
<div data-bbox="563 1144 842 1216" style="border: 1px solid black; padding: 2px; text-align: center;"> <p data-bbox="611 1167 794 1189"><u>Nama objek nama</u></p> </div> <p data-bbox="651 1238 735 1272">Objek</p>	<p data-bbox="927 1046 1362 1191">Merupakan objek yang melakukan representasi pada objek yang terlibat dalam interaksi.</p>
 <p data-bbox="616 1541 775 1574">Waktu Aktif</p>	<p data-bbox="927 1292 1362 1471">Merupakan suatu objek sedang dalam keadaan aktif atau berinteraksi, semua yang terlibat terhadap waktu aktif adalah dalam keadaan aktif.</p>
<p data-bbox="632 1597 751 1619">&lt;&lt;create&gt;&gt;</p>  <p data-bbox="584 1675 807 1709">Pesan tipe <i>create</i></p>	<p data-bbox="927 1592 1362 1738">Menggambarkan pesan saat objek yang baru dibuat dan dikaitkan dengan objek yang sesuai.</p>

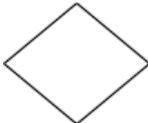
 <p>Pesan tipe <i>call</i></p>	<p>Merupakan gambaran yang menunjukkan pesan berupa pemanggilan operasi antara objek</p>
 <p>Pesan tipe <i>send</i></p>	<p>Merupakan gambaran pengiriman pesan antar objek yang menunjukkan komunikasi.</p>
 <p>Pesan tipe <i>return</i></p>	<p>Merupakan objek dapat melakukan atau menjalankan metode yang pada operasi memberikan hasil pengembalian atau keluaran ke objek tertentu.</p>
 <p>Pesan tipe <i>destroy</i></p>	<p>Merupakan pernyataan bahwa satu objek mengakhiri hidup dari objek lain, jika ada <i>create</i> lebih baik ada <i>destroy</i>.</p>

Sumber: Hutabri & Putri (2019)

### c. Activity Diagram

Menurut Musthofa & Adiguna (2022) *Activity diagram* menggambarkan alur kerja atau aktivitas sistem, proses bisnis atau menu-menu pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan actor.

Tabel 2. 4 Activity diagram

Simbol	Deskripsi
 Staus awal/akhir	Merupakan status awal dan akhir yang menunjukkan tempat eksekusi akan dimulai.
 Aktivitas	Merupakan kegiatan yang dilakukan sistem atau merepresentasikan aktivitas yang terlibat dalam proses sistem.
 <i>Decision</i> (Percabangan)	Merupakan hubungan untuk keputusan aktivitas yang harus dibuat dalam alur eksekusi dengan kondisi yang diberikan.
 <i>Join</i> (Penggabungan)	Merupakan titik alur eksekusi terbagi bertemu kembali menjadi satu eksekusi tunggal

Sumber: Hutabri & Putri (2019)

## 15. Blackbox Testing

Pengujian *black box* adalah metode pengujian perangkat lunak yang menguji fungsionalitas suatu aplikasi dibandingkan dengan struktur atau cara kerja internalnya. Tidak diperlukan pengetahuan khusus tentang kode aplikasi/struktur internal dan pengetahuan pemrograman umum. Gunakan deskripsi perangkat lunak eksternal, termasuk spesifikasi, persyaratan, dan desain, untuk mendapatkan kasus pengujian. Tes-tes ini mungkin fungsional atau non-fungsional, meskipun seringkali bersifat fungsional.

Perancang tes memilih masukan yang valid dan tidak valid dan menentukan keluaran yang benar (A. P. Putra *et al.*, 2020) .

Menurut Ramdani *et al.*, (2023) *blackbox* testing memiliki kelebihan dan kelemahan dalam pengujian sistem. Berikut kelebihan dan kelemahan pengujian *blackbox*:

- a) Kelebihan Blackbox
  - 1) Tidak memerlukan pengetahuan tentang bahasa pemrograman tertentu.
  - 2) Pengujian dilakukan dari sudut pandang pengguna,
  - 3) Ketergantungan antara programmer dan tester.
- b) Kelemahan *Blackbox*
  - 1) Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer.
  - 2) Beberapa bagian *back end* tidak diuji sama sekali.

Menurut Muhandi *et al.*, (2020) Pendekatan *blackbox* adalah metode pengujian yang melengkapi pengujian white-box dengan kemampuan untuk mengungkap lebih banyak jenis kesalahan. Pengujian *blackbox* fokus pada menemukan kesalahan dalam beberapa kategori berikut:

- a) Fungsi yang salah atau hilang
- b) Kesalahan antarmuka
- c) Kesalahan dalam struktur data atau akses basis data eksternal
- d) Kesalahan kinerja dalam sistem
- e) Kesalahan inisialisasi dan penghentian

## B. Kajian Empiris

Terdapat penelitian sebelumnya yang menggunakan metode *Convolutional Neural Network* dan *Random Forest* untuk sistem identifikasi tertentu. Penelitian yang dilakukan menggunakan berbagai jenis data dan model identifikasi yang sesuai dengan kebutuhan .

Penelitian serupa yang dilakukan Kurniadi dkk mengenai metode *Convolutional Neural Network* yang digunakan untuk mengklasifikasi topeng Cirebon. Penelitian dengan judul “Klasifikasi Topeng Cirebon Menggunakan Metode *Convolutional Neural Network*” ini mendapatkan hasil bahwa pengujian yang dilakukan menunjukkan bahwa metode CNN mampu mengklasifikasi topeng Cirebon dengan tingkat akurasi 99% (Kurniadi *et al.*, 2021).

Penelitian lain yang serupa dilakukan oleh Syahputra dan Wibowo pada tahun 2020 mengenai metode *Convolutional Neural Network* yang digunakan untuk mengklasifikasi Genus tanaman anggrek. Penelitian dengan judul “Klasifikasi Genus Tanaman Anggrek berdasarkan Citra Kuntum Bunga Menggunakan Metode *Convolutional Neural Network* (CNN)” ini mendapatkan hasil bahwa pengujian yang dilakukan mampu mengklasifikasi Genus tanaman anggrek dengan tingkat akurasi 97%. Pada penelitian ini menggunakan data 900 citra jenis Genus anggrek yang dikategorikan empat kelas yaitu *Cattleya*, *Dendrobium*, *Oncidium* dan *Phalaenopsis* (Syahputra & Wibowo, 2020).

Penelitian lain yang serupa dilakukan oleh Gusti Alfahmi Anwar dan Desti Rimirasih (2019) mengenai metode *Convolutional Neural Network* yang digunakan untuk citra genus panthera. Penelitian dengan judul “Klasifikasi Citra *Genus Panthera* menggunakan metode *Convolutional Neural Network* (CNN)” ini mendapatkan hasil bahwa pengujian yang dilakukan mampu mengklasifikasi *Genus Panthera* dengan tingkat akurasi 92,31%. Data training terdiri dari 3840 citra, data validasi sebanyak 960 citra (Anwar & Rimirasih, 2019).

Penelitian serupa mengenai metode *Random Forest* dapat digunakan untuk mengukur tingkat keparahan penyakit pada daun apel. Penelitian dengan judul “Penerapan *Random Forest* untuk mengukur tingkat keparahan penyakit pada daun apel”. Pada penggunaan dataset sebanyak 85-15 dengan 50 pohon keputusan. Pada tahap training menunjukkan hasil akurasi sebesar 99,9496% sedangkan pada tahap testing menunjukkan akurasi 66-76% (Ratnawati & Sulistyaningrum, 2020).

Penelitian lain yang serupa mengenai metode *Random Forest* dapat digunakan untuk klasifikasi awal resiko awal diabetes melitus. Penelitian ini dengan judul “Klasifikasi Menggunakan Metode *Support Vector Machine* dan *Random forest* Untuk Deteksi Awal Risiko Diabetes Melitus” menyatakan bahwa klasifikasi menggunakan metode *Random Forest* menghasilkan nilai akurasi 91%, *recall* sebesar 98%, *precision* sebesar 99% dan *F1\_Score* sebesar 98% (Junus *et al.*, 2023)

Penelitian lain yang serupa mengenai metode *Random forest* dapat digunakan untuk klasifikasi *Leaf Blast* Tumbuhan Padi. Penelitian ini dengan judul “Peningkatan *Random Forest* dengan menerapkan *GLCM* (*Gray Level Co-Occurrence Matrix*) pada Klasifikasi *Leaf Blast* Tumbuhan Padi”. Pada penggunaan 200 data uji (Miftahuddin *et al.*, 2022).

### C. Kerangka Berpikir

