

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Riset yang dilaksanakan oleh Didit Iswantoro beserta Dewi Handayani UN di tahun 2022 berjudul [6] “Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode *Convolutional Neural Network* (CNN)” memanfaatkan algoritma CNN dalam pengembangan, penelitian tersebut menghasilkan akurasi sebesar 97,5%, dataset yang digunakan sebesar 2000 data yang terbagi kedalam 2 penyakit yaitu hawar daun dan karat daun. Metode yang digunakan sudah cukup baik dalam melakukan klasifikasi penyakit, namun demikian perlu adanya penambahan beberapa dataset penyakit tanaman jagung.

Penelitian yang merujuk pada Muhammad Imron Rosadi dan Moch. Lutfi yang diterbitkan pada tahun 2021 dengan judul [7] “Identifikasi Jenis Penyakit Daun Jagung Menggunakan Deep Learning Pre-Trained Model”. Pada penelitian tersebut memanfaatkan metode CNN (*Convolutional Neural Network*) dalam pengembangannya dengan menggunakan *dataset* yang terdiri dari 1923 gambar, hasil pelatihan data pada lima kategori daun jagung, yang terdiri dari satu kategori daun sehat dan empat kategori penyakit daun, yaitu karat daun, hawar daun, serta bulai daun menghasilkan akurasi sebesar 88%. Namun demikian, akurasi tersebut dapat ditingkatkan untuk menghasilkan analisis data yang jauh lebih baik.

Penelitian yang dilakukan oleh Ary Yoggyanto, Arfan Maulana, dan Diky Tri Cahyo A pada tahun 2024 dengan judul [8] “Penerapan Metode Convolutional Neural Network (CNN) Dalam Klasifikasi Penyakit Tanaman Jagung” Penelitian ini memanfaatkan *Convolutional Neural Network* sebagai metodenya dalam menjalankan sesi pelatihan *dataset*, hasil riset menyatakan bahwa model yang dikembangkan mampu memperoleh akurasi yang didapatkan yaitu 91,5%, *dataset* yang digunakan sebesar 4188 data dengan proses training mencapai 10 *epoch*.

Namun demikian jumlah training dapat dilakukan lebih dari 10 epoch agar menghasilkan akurasi dan validasi gambar lebih baik.

Penelitian dengan judul [9] “Implementasi Metode CNN Pada Klasifikasi Penyakit Jagung” oleh Reza Mawarni, Resty Wulaningrum, dan Risa Helilintar pada tahun 2023 tersebut menggunakan metode CNN dalam proses pengembangan model, akurasi yang dihasilkan yaitu sebesar 97% dengan menggunakan epoch 40. Data gambar yang digunakan sebesar 592 gambar yang terdiri atas 480 training dan 112 testing. Dari hasil akhir yaitu dengan menggunakan metode CNN terbukti cukup efisien dalam mengenali dan mengklasifikasikan penyakit pada tanaman jagung.

Penelitian yang merujuk pada Ulfa Khaira, Indra Weni, dan Weni Wilia pada tahun 2024 dengan judul [10] “Rancang Bangun Aplikasi Deteksi Penyakit Tanaman Jagung Melalui Citra Daun Berbasis Android Menggunakan Algoritma *Convolutional Neural Network*”, dataset yang digunakan yaitu 2600 gambar yang terdiri *Bacterial Leaf Blight*, *Common Rust*, dan *Healthy*, aplikasi yang dikembangkan sudah cukup baik dalam melakukan deteksi di semua versi Android.

Penelitian dengan judul [11] “Pemanfaatan Algoritma *Convolutional Neural Network* Dengan Untuk Mendeteksi penyakit Pada Tumbuhan Jagung” oleh Alfin Kabir Rifai, Muhammad Rafi Muttaqin, dan Dede Irmayanti yang diterbitkan pada tahun 2024, jurnal ini menggunakan *Convolutional Neural Network* (CNN) sebagai algoritmanya dengan *dataset* sebanyak 4188 gambar yang terdiri atas 4 kategori yaitu *Blight* dengan 1146 gambar, *Common Rust* dengan 1306 gambar, *Gray Leaf Spot* dengan 574 gambar, dan *Healthy* dengan 1162 gambar. Pada penelitian ini menunjukkan kinerja yang baik dengan nilai akurasi mencapai 94% dan *loss* 24%, model yang dibangun juga diimplementasikan kedalam aplikasi berbasis *Android*.

Tabel 2.1 Penelitian Terkait

No	Judul Penelitian	Penulis	Metode	Akurasi	Jumlah Dataset	Aplikasi Android
1	Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode <i>Convolutional Neural Network</i> (CNN) (2022)	Didit Iswanto dan Dewi Handayani UN	<i>Convolutional Neural Network</i> (CNN)	97,5%	2000 gambar	Tidak
2	Identifikasi Jenis Penyakit Daun Jagung Menggunakan Deep Learning Pre-Trained Model (2021)	Muhammad Imron Rosadi dan Moch. Lutfi	<i>Convolutional Neural Network</i> (CNN)	88%	1923 gambar	Tidak
3	Penerapan Metode <i>Convolutional Neural Network</i> (CNN) Dalam Klasifikasi Penyakit Tanaman Jagung (2024)	Ary Yoggyanto, Arfan Maulana, dan Diky Tri Cahyo A	<i>Convolutional Neural Network</i> (CNN)	91.10%	4188 gambar	Tidak
4	Implementasi Metode CNN Pada Klasifikasi Penyakit Jagung (2023)	Reza Mawarni, Resty Wulaningrum, dan Risa Helilintar	<i>Convolutional Neural Network</i> (CNN)	97%	592 gambar	Tidak
5	Rancang Bangun Aplikasi Deteksi Penyakit Tanaman Jagung Melalui Citra Daun Berbasis Android Menggunakan Algoritma <i>Convolutional Neural Network</i> (2024)	Ulfa Khaira, Indra Weni, dan Weni Wilia	<i>Convolutional Neural Network</i> (CNN)	99%	2600 gambar	Iya
6	Pemanfaatan Algoritma <i>Convolutional Neural Network</i> Dengan Untuk Mendeteksi penyakit Pada Tumbuhan Jagung (2024)	Alfin Kabir Rifai, Muhammad Rafi Muttaqin, dan Dede Irmayanti	<i>Convolutional Neural Network</i> (CNN)	94%	4188 gambar	Iya

Berdasarkan penelitian terkait, hanya sedikit penelitian yang menerapkan hasil training data yang sudah dibuat hingga menjadi aplikasi *Mobile*. Dengan demikian, penelitian ini akan menerapkan training data penyakit jagung kedalam aplikasi Android sehingga aplikasi tersebut dapat melakukan deteksi pada penyakit tanaman jagung.

2.2 Landasan Teori

a. Tanaman Jagung

Jagung merupakan produk dari pertanian setelah padi yang akan diperkirakan akan menggantikan padi sebagai makanan pokok di Indonesia. Jagung merupakan jenis biji-bijian dengan nilai strategis dan ekonomi yang tinggi, serta memiliki potensi besar untuk dikembangkan sebagai sumber utama karbohidrat dan protein [1].

1. Penyakit Tanaman Jagung

Gangguan kesehatan tanaman jagung umumnya mampu diamati melalui daunnya. Sejumlah penyakit yang kerap menyerang tanaman jagung diantaranya hawar daun (*blight*) yang ditimbulkan oleh *Helminthosporium turcicum*, bercak daun (*gray leaf spot*) akibat infeksi jamur *Cercospora zea-maydis*, serta karat daun (*common rust*) yang diakibatkan oleh jamur *Puccinia sorghi* Schw dan *Puccinia polypore* Underw [4].

Infeksi hawar (*blight*) tanaman jagung merupakan penyakit yang banyak menyebar di Amerika, Asia, Afrika, dan juga Eropa, kerugian apabila tanaman jagung terkena penyakit hawar dapat mencapai 50%. Penyakit hawar jagung terindikasi berbentuk bercak kecil yang berbentuk oval, yang kemudian bercak tersebut menjadi memanjang seperti ellips serta tumbuh menjadi nekrotik [6].

Penyakit bercak daun (*gray leaf spot*) merupakan penyakit yang ditandai dengan daun abu-abu yang tampak seperti lesi bulat kecil dengan memiliki zona kuning melingkar yang berada disekitar daun. Lesi tersebut dapat bertransformasi kecoklatan sebelum sporulasi jamur berlangsung. Pada fase permulaan, penyakit

tersebut cukup kompleks dikenali karena bercaknya menyerupai gejala umum karat daun [12].

Penyakit karat daun (*common rust*) yang diakibatkan oleh *Puccinia sorghi* ditandai oleh munculnya bintik-bintik merah yang mengeluarkan butiran halus menyerupai tepung berwarna coklat kekuningan. Kehadiran penyakit ini menghambat proses *fotosintesis* pada anaman jagung, sehingga pertumbuhannya melambat dan dapat menyebabkan kematian pada tumbuhan [6].

b. *Artificial Intelligence* (Kecerdasan Buatan)

Artificial Intelligence (AI) merupakan teknologi yang banyak diadopsi pada era industri 4.0. AI mampu menghubungkan setiap perangkat, sehingga seseorang dapat mengotomatisasi semua perangkat yang berada didalam satu lokasi yang sama. Selain itu, saat ini sudah banyak mesin yang mampu menafsirkan suatu kondisi atau peristiwa tertentu dengan bantuan *Artificial Intelligence* (AI) [13].

Dalam penjelasan lain, kecerdasan buatan (AI) merupakan bidang multidisiplin yang bertujuan mengotomatisasikan aktivitas yang saat ini memerlukan kecerdasan manusia. *Artificial Intelligence* (AI) dan manusia dapat berkolaborasi dalam pengambilan keputusan agar lebih objektif dan tidak terlalu dipengaruhi oleh nilai-nilai pribadi [14].

c. *Machine Learning*

Machine Learning (ML) dikategorikan sebagai sebuah pembelajaran mesin yang menjadi cabang dari AI untuk terfokus pada mempelajari data atau (*learn from data*), fokus daripada *Machine Learning* yaitu pembangunan sistem yang mampu belajar secara mandiri tanpa membutuhkan pemrograman terus menerus oleh manusia. Apabila diibaratkan *Machine Learning* sebagai kendaraan maka data merupakan sebuah bahan bakar dari kendaraan tersebut, karena *Machine Learning* sangat amat membutuhkan data sebagai metode dalam menyelesaikan permasalahan [15].

Dalam penjelasan lain, *Machine Learning* dapat dikategorikan sebagai kecerdasan buatan dimana sistem elektronik dapat mengasah kemampuan berdasarkan kumpulan data dan proses belajar, sehingga dapat mengoptimalkan potensinya dalam menyelesaikan operasi tertentu tanpa memerlukan pengkodean secara langsung [16].

d. *Deep Learning*

Deep learning merupakan bagian dari *machine learning* yang algoritmanya terinspirasi dari bentuk dan cara kerja otak manusia. Struktur ini dikenal sebagai Jaringan Saraf Tiruan (JST) atau *Artificial Neural Network* (ANN). *Deep learning* mencakup algoritma, seperti *Convolutional Neural Network* (CNN), *Long Short Term Memory Network* (LSTM), *Recurrent Neural Network* (RNN), dan *Self Organizing Maps* (SOM). Salah satu algoritma populer, CNN dirancang untuk mengolah dalam format dua dimensi (2D). CNN kerap dimanfaatkan dalam pengelompokan data berlabel melalui metode *supervised learning* [15].

e. *Artificial Neural Network* (ANN)

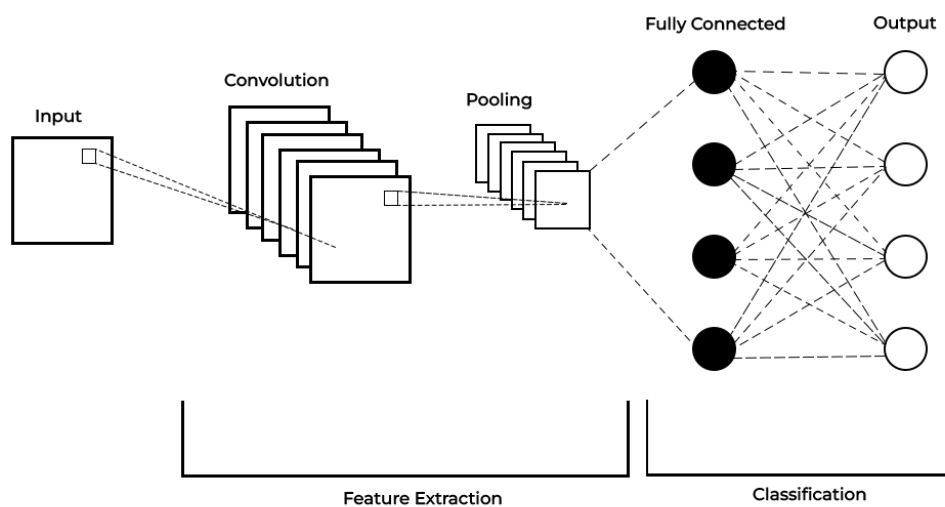
Artificial Neural Network (ANN) merupakan representasi buatan dari otak manusia yang berusaha meniru proses pembelajaran pada otak manusia. Struktur dasar dari ANN terdiri dari *neuron*, yang merupakan hasil perkalian antara nilai *input* dan bobot. Arsitektur jaringan syaraf tiruan terbagi menjadi tiga jenis, yaitu jaringan satu lapisan (*single layer network*), jaringan *multi* lapisan (*single layer network*), jaringan *multi* lapisan (*multi layer network*), dan jaringan dengan lapisan kompetitif. Beberapa faktor yang digunakan untuk mengukur akurasi kinerja ANN meliputi *learning rate*, jumlah iterasi, dan toleransi *error* [17].

Jaringan syaraf tiruan atau *Artificial Neural Network* (ANN) adalah algoritma kecerdasan buatan yang mampu belajar dari data dan dapat membangun model dalam waktu yang singkat. Namun, ANN memiliki kelemahan karena membutuhkan proses segmentasi dan ekstraksi citra agar dapat bekerja secara optimal. Salah satu pengembangan ANN yang dirancang untuk mengatasi kendala ini adalah *Convolutional Neural Network* (CNN) [10].

f. *Convolutional Neural Network (CNN)*

Convolutional Neural Network (CNN) merupakan tipe arsitektur *deep learning* yang mampu mengembangkan sistem menggunakan kumpulan informasi dalam jumlah besar, dengan menggabungkan proses ekstraksi fitur dan klasifikasi. CNN dirancang untuk mengolah data dengan arsitektur grid, seperti gambar 2D, serta sanggup mengolah parameter besar seperti video [18].

Dalam penjelasan lainnya CNN ialah salah satu model jaringan saraf tiruan yang mempunyai banyak lapisan didalamnya yang salah satunya yaitu lapisan konvolusi, lapisan penyatuan, non-linearitas, dan lapisan terhubung penuh. CNN memiliki performa yang baik dalam memproses data gambar [19].



Gambar 2.1 Arsitektur CNN

1. *Convolutional Layer*

Dalam arsitektur CNN, *Convolutional Layer* bertanggung jawab untuk menerapkan operasi konvolusi pada *output* dari layer *input*. Langkah konvolusi menjadi komponen utama dalam perancangan arsitektur CNN [6].

2. *Pooling Layer*

Pooling merupakan tahapan pada CNN yang berfungsi untuk mengurangi ukuran matriks. *Pooling layer* biasanya muncul setelah tahapan *convolutional layer*. Bentuk umum yang digunakan pada *pooling layer* yaitu dengan menggunakan filter 2 x 2, yang dilakukan dalam dua tahap dan bekerja pada potongan-potongan input [6].

3. *Fully-Connected Layer*

Lapisan *Fully-connected*, yang juga dikenal sebagai *dense layer*, adalah lapisan penghubung antara setiap *neuron* pada lapisan sebelumnya ke setiap *neuron* di lapisan berikutnya, menyerupai struktur neural network (jaringan saraf tiruan) [20].

g. *Matrix Evaluasi*

Pada tahap ini, performa model akan dievaluasi untuk menilai kinerjanya. Penilaian dilakukan dengan menggunakan *matrix accuracy*, *precision*, *f1-score*, dan *recall*. Performa model klasifikasi dinilai menggunakan pendekatan *confusion matrix*, yang terdiri dari beberapa istilah utama. Istilah tersebut yaitu : *True Positive* (TP) yaitu data yang berhasil diprediksi dengan benar oleh model, *True Negative* (TN) yaitu data negatif yang diprediksi dengan benar oleh model, *False Positive* (FP) yaitu data negatif yang keliru diprediksi sebagai data positif oleh model, dan *False Negative* (FN) yaitu data positif yang salah diprediksi sebagai data negatif oleh model. Rumus untuk menghitung *accuracy*, *precision*, *recall*, dan *f1-score* dapat ditentukan menggunakan persamaan 1 hingga persamaan 4 [21].

Accuracy merupakan presentase sampel yang berhasil diklasifikasikan dengan tepat ke dalam kategori yang sesuai. Hasil akurasi dihitung menggunakan rumus yang terdapat pada persamaan 1 [21].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision merupakan *matrix* yang menunjukkan tingkat ketepatan sebuah model dalam memprediksi kejadian positif dari keseluruhan hasil prediksi. Nilai *precision* dihitung menggunakan rumus pada persamaan 2 [21].

$$precision = \frac{TP}{TP + FP}$$

Recall adalah *matrix* yang mengukur tingkat keberhasilan model dalam mengidentifikasi tupel positif, dinyatakan dalam bentuk presentase tupel positif yang diklasifikasikan dengan benar sebagai kelas positif. Nilai *recall* dapat dihitung menggunakan rumus pada persamaan 3 [21].

$$recall = \frac{TP}{TP + FN}$$

F1-score adalah *matrix* evaluasi dalam pencarian informasi (retrieval) yang mempertimbangkan keseimbangan antara nilai *recall* dan *precision*. Nilai *F1-score* dapat dihitung menggunakan rumus pada persamaan 4 [21].

$$f1 - score = \frac{2 \times (Precision * Recall)}{(Precision + Recall)}$$

h. Google Colaboratory

Google Colaboratory (Colab) adalah sebuah layanan yang dikembangkan oleh Google Research untuk memberikan kesempatan kepada pengguna untuk menjalankan kode Python secara langsung dengan menggunakan *browser*, Colab cukup efektif digunakan untuk *machine learning*, dan analisis data. Dalam konteks lain, Colab berfungsi sebagai layanan notebook Jupyter yang tersedia secara daring tanpa memerlukan instalasi [22].

i. TensorFlow

TensorFlow adalah salah satu library pada *Machine Learning*, yang merupakan teknologi komputer yang berfungsi sebagai media melatih model *machine learning* dan *deep learning* [23], selain itu TensorFlow dapat digunakan sebagai pendeteksi berbagai macam objek yang sudah ditentukan dan dapat menggunakan alat bantu untuk mendeteksinya yaitu handphone [24].

1. TensorFlow Lite

TensorFlow Lite merupakan *library* yang dapat memudahkan para pengembang dalam mengembangkan atau membuat dan menjalankan model AI TensorFlow di perangkat *mobile*, *mikrokontroller*, dan perangkat IoT lainnya [25]. Dalam pengertian lainnya TensorFlow Lite merupakan solusi apabila ingin melakukan penanaman model ke perangkat *mobile phone*, pendekatan ini sangat membantu untuk mengurangi kebutuhan akses berulang ke server. TensorFlow kompatibel dengan perangkat Android dan iOS, dilengkapi dengan API C++ dilengkapi dengan Java Wrapper guna mendukung pengembangan aplikasi Android [26].

j. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah acuan yang umum diterapkan dalam ranah industri dalam mendesain sistem, visualisasi, dan mendokumentasikan *software*. UML merekomendasikan sebuah standar dalam mendesain model sebuah sistem [27].

1. *Use Case Diagram*

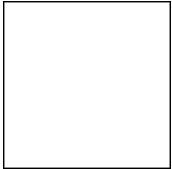
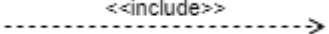
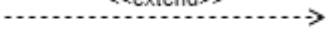
Use case diagram merupakan model yang bertujuan guna merepresentasikan pola interaksi dari sistem informasi yang dibangun atau dijalankan. *Diagram* ini bertujuan mengenali berbagai fitur di dalam aplikasi dan pihak-pihak dimana mereka dapat mengakses dan memanfaatkan fitur tersebut [28].

Dalam penjelasan lain, *Use case diagram* adalah bagan yang memvisualisasikan interaksi antara satu atau lebih aktor dengan sistem yang akan dikembangkan. *Use case diagram* digunakan untuk menjelaskan elemen-elemen dalam sistem dan pihak-pihak yang diberi wewenang untuk mengakses fitur-fitur tertentu [29].

Tabel 2.2 Simbol *Use Case Diagram*

Simbol	Nama	Fungsi
	Aktor	Mewakili orang, sistem yang lain, atau digunakan dalam berinteraksi dengan <i>use case</i>

Tabel 2.2 Simbol *Use Case Diagram* (lanjutan)

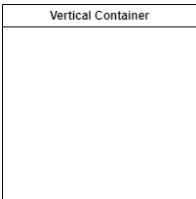
	<i>Use case</i>	Digunakan untuk berinteraksi antara sistem dan aktor
	<i>System</i>	Sistem yang sedang dikembangkan
	<i>Association</i>	Hubungan antara aktor dengan <i>use case</i>
	<i>Generalization</i>	Relasi dimana komponen turunan (descendent) mewarisi perilaku serta kerangka informasi dari objek <i>parent</i> (ancestor)
	<i>Include</i>	Sebuah <i>use case</i> seluruhnya mencakup atau merupakan bagian dari fungsionalitas yang dimiliki <i>use case</i> lainnya
	<i>Extend</i>	Jika suatu kondisi tertentu terpenuhi, sebuah <i>use case</i> dapat menjadi ekstensi yang menambahkan fungsionalitas pada <i>use case</i> lainnya

2. Activity Diagram

Activity Diagram ialah penggambaran rangkaian proses dari sebuah sistem atau aktivitas bisnis yang terdapat dalam *software*. Diagram aktivitas memberikan gambaran berupa proses yang dijalankan oleh sistem, bukan yang dijalankan oleh kantor, sehingga berfokus pada apa yang dijalankan oleh sistem tersebut [30].

Activity Diagram memiliki fungsi untuk menggambarkan seluruh kegiatan yang berlangsung dalam sistem, termasuk yang dijalankan oleh pengelola maupun pengguna [31].

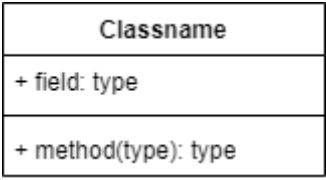
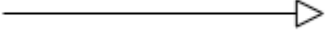
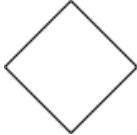
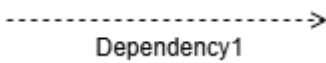
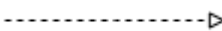
Tabel 2.3 Simbol *Activity Diagram*

Simbol	Nama	Fungsi
	<i>Initial node</i> / status awal	Digunakan untuk <i>start</i> ataupun status awal pada <i>activity diagram</i>
	<i>Activities</i> / Aktivitas	Digunakan untuk menuliskan aktivitas yang sedang berlangsung
	<i>Decision</i> / Percabang	Dimana aktivitas lebih dari satu
	<i>Join / Fork</i>	Digunakan untuk menggabungkan beberapa aktivitas menjadi satu kesatuan
	<i>Swimlane</i>	Digunakan untuk membedakan atau memisahkan organisasi yang bertanggung jawab atas aktivitas yang dilakukan
	<i>Final Node</i>	Digunakan untuk status akhir atau penutup

3. *Class Diagram*

Class diagram adalah representasi arsitektur perangkat lunak yang mendefinisikan kelas-kelas yang hendak dikembangkan dalam aplikasi. Setiap kelas mengandung properti serta fungsi atau aksi yang digunakan untuk menggambarkan karakteristik dan perilakunya [30].

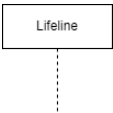
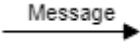
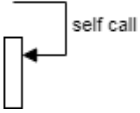
Tabel 2.4 Simbol *Class Diagram*

Simbol	Nama	Fungsi
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut dan operasi yang sama, <i>Class</i> terdiri atas atribut dan operasi/ <i>method</i>
	<i>Association</i>	Relasi antarkelas, menghubungkan objek yang satu dengan lainnya
	<i>Generalization</i>	Relasi antar kelas yang mengacu pada generalisasi dan spesialisasi (umum-ke-khusus), di mana objek turunan (<i>descendant</i>) mewarisi perilaku dan struktur data dari objek induk (<i>ancestor</i>)
	<i>Nary Association</i>	Digunakan untuk mencegah hubungan atau asosiasi yang melibatkan lebih dari dua objek
	<i>Collaboration</i>	Merupakan penjelasan mengenai rangkaian aksi-aksi yang dilakukan oleh sistem dan ditampilkan secara berurutan
	<i>Dependency</i>	Relasi antarkelas dengan kebergantungan antar kelas
	<i>Realization</i>	Aktivitas yang harus dilakukan oleh objek

4. *Sequence Diagram*

Sequence diagram merupakan representasi interaksi komponen pada *use case* yang menunjukkan alur hidup objek serta pesan-pesan yang dikirim dan diterima antar objek. Terkait hal ini, dalam pembuatan *sequence diagram*, penting untuk mengidentifikasi di tahap awal objek yang berinteraksi dalam *use case* tersebut serta fungsi yang terdapat pada setiap kelas [30].

Tabel 2.5 Simbol *Sequence Diagram*

Simbol	Nama	Fungsi
	<i>Actor</i>	Mengartikan seseorang atau perangkat dalam berinteraksi dengan sistem
	<i>Lifeline</i>	Objek entitas antarmuka yang saling berkomunikasi
	<i>Object Message</i>	Menunjukkan komunikasi atau hubungan antar objek yang menggambarkan urutan kejadian
	<i>Message to Self</i>	Menyatakan pesan atau hubungan antar objek yang menggambarkan urutan kejadian

k. Android

Android merupakan sebuah *operating system* (OS) *mobile* dengan basis *open source* yang sudah sangat besar untuk saat ini ini, Sistem Android dikembangkan dengan menggunakan *kernel linux*. Dengan basis *open source* Android memungkinkan untuk dimodifikasi secara bebas sehingga membuat para pengembang aplikasi menggunakan Android sebagai dasar pembuatan proyek aplikasi [32].

Android dalam pengertian lain, adalah OS (*operating system*) yang diperuntukan untuk *smartphone* dan *tablet*. *Platform* ini berperan selaku "penghubung" yang menjembatani pengguna dengan perangkat, memberikan kemampuan kepada pengguna untuk berinteraksi dengan perangkat melalui aplikasi yang dijalankan [33].

l. Android Studio

Android studio adalah sebuah *Intergrated Development Environment* (IDE) yang dikhususkan guna membangun aplikasi android. Basis yang digunakan oleh android studio adalah IntelliJ IDEA, android studio terintegrasi dengan SDK (Software

Development Kit) yang diperuntukan untuk memasang aplikasi ke perangkat Android. Android studio ialah *software* yang dikembangkan oleh eclipse, yang dalam pengembangannya diperuntukan untuk menjadi lebih kompleks dan professional [34].

Dalam penjelasan lain, Android Studio menyediakan berbagai alat yang ditujukan untuk pengujian, publikasi, serta mendukung seluruh proses pengembangan aplikasi untuk perangkat Android [35].

Android Studio didesain spesifik untuk pembangunan aplikasi berbasis Android, dan IntelliJ IDEA. Android Studio dilengkapi dengan beragam kemampuan pendukung yang dapat meningkatkan efisiensi dalam pembuatan aplikasi Android, seperti [36] :

1. Sistem build Gradle yang dapat disesuaikan dengan kebutuhan
2. Emulator yang responsif dan dilengkapi dengan berbagai fitur
3. *Platform* pengembangan yang komprehensif bagi seluruh perangkat berbasis Android
4. Fitur *Instant Run* yang memungkinkan penerapan penyesuaian pada aplikasi yang sedang berjalan tanpa perlu membangun APK baru
5. Dukungan untuk pengembangan dengan C++ dan NDK

m. Figma

Figma merupakan software dibidang desain dan sebagai alat prototyping. Aplikasi figma adalah aplikasi yang dapat berfungsi sebagai pembuat rancangan *interface pengguna* (UI) dan *user experience* (UX) yang dapat diterapkan dalam perancangan layout aplikasi, web, dan lain-lain. Figma sangat memungkinkan penggunaanya untuk bekerja dimana dengan menggunakan browser yang ada diperangkat sudah dapat menggunakan Figma [37].