

BAB 2

TINJAUAN PUSTAKA

2.1 Kucing

Kucing (*Felis catus*) adalah hewan berbulu yang sering dipelihara sebagai hewan peliharaan. Seperti hewan peliharaan lainnya, kucing merupakan hasil domestikasi dari miacis, nenek moyang anjing dan beruang. Miacis berevolusi menjadi kucing besar seperti singa dan harimau, dan kemudian menjadi nenek moyang kucing domestik. Fosil mumi kucing yang ditemukan di Mesir memberikan bukti pertama tentang nenek moyang kucing domestik ini [7].

Penelitian oleh Claudio Ottoni dari University of Rome menunjukkan bahwa domestikasi kucing terjadi dalam dua jenis. Ia menyatakan bahwa semua kucing domestik berasal dari nenek moyang yang sama, yaitu kucing liar Afrika Utara atau Asia Barat. Melalui analisis DNA kucing purba dari berbagai belahan dunia, peneliti menemukan bahwa domestikasi kucing dimulai pada periode Neolitik dan berlanjut hingga Mesir kuno. Penelitian Ottoni juga menjelaskan penyebaran kucing ke seluruh dunia. Dengan menganalisis DNA purba dari sisa-sisa kucing yang ditemukan di kota pelabuhan, ilmuwan menyimpulkan bahwa kucing dibawa dalam kapal untuk melindungi persediaan makanan dari hewan pengerat, sehingga mereka menyebar ke berbagai belahan dunia selama pelayaran [7].



Gambar 2.1 Domestic Cat

Sumber: <https://www.nationalgeographic.com/animals/mammals/facts/domestic-cat>

Kucing adalah salah satu hewan yang sering dipelihara oleh manusia. Sebagai hewan peliharaan yang populer, kucing memiliki daya tarik tersendiri berkat bentuk tubuh, mata, hidung, dan variasi warna bulunya yang beragam [8]. Ciri-ciri kucing

meliputi panjang tubuh sekitar 76 cm, tinggi tubuh 25-28 cm, berat tubuh jantan 3-4 kg dan betina 2-3 kg, serta masa hidup yang berkisar antara 13 hingga 17 tahun. Kucing yang telah didomestikasi dikenal dengan nama ilmiah *Felis catus* atau *Felis domesticus*. Mereka berkomunikasi melalui berbagai vokalisasi dan bahasa tubuh, termasuk meowing, purring, hissing, growling, squeaking, chirping, clicking, dan grunting [9].

2.2 Feline Calicivirus

Feline Calicivirus (FCV) adalah virus RNA untai tunggal kecil dari keluarga Caliciviridae yang menyebabkan infeksi saluran pernapasan atas pada kucing melalui mulut dan hidung di seluruh dunia. Virus ini sangat lazim dan menyebabkan penyakit Upper Respiratory Tract Disease (URTD) dan Virulent Systemic Disease (FCV-VSD), yang dapat mengakibatkan kematian pada kucing terutama yang berusia di bawah 1 tahun [10]. Kucing yang terinfeksi biasanya menunjukkan gejala klinis secara visual yang dapat dilihat di Tabel 2.1.

Tabel 2.1 Gejala Klinis Feline Calicivirus

No	Gejala	Deskripsi	Contoh Gambar
1	Luka di mulut	Luka atau ulserasi pada lidah, gusi, atau langit-langit mulut kucing.	
2	Produksi air liur berlebih	Kucing mengalami peningkatan produksi air liur yang berlebihan, sering disertai dengan drooling.	
3	Radang gusi	Gusi kucing merah dan bengkak, sering disertai dengan bau mulut.	

4	Peradangan mata	Mata kucing terlihat merah, bengkak, atau berair.	
5	Bersin dan Batuk	Kucing mengalami keluarnya cairan dari hidung dan mata, sering bersin, dan batuk.	
6	Hidung dan mata berair	Kucing mengalami keluarnya cairan dari hidung dan mata, sering disertai bersin.	
7	Demam	Kucing mengalami peningkatan suhu tubuh, terlihat lesu, kurang nafsu makan, dan mungkin mencari tempat yang lebih sejuk.	
8	Lemas dan Kehilangan Nafsu Makan	Kucing tampak lesu dan tidak aktif, dengan penurunan energi yang signifikan.	

Adapun parameter suara yang dapat diamati pada kucing yang terinfeksi FCV, yaitu:

1. Rate pernapasan: FCV sering menyebabkan infeksi saluran pernapasan atas, yang dapat menyebabkan peningkatan frekuensi pernapasan.

2. Suara napas: Pada kucing yang terinfeksi FCV, mungkin terdengar suara mengi atau napas yang berat akibat peradangan dan sekresi di saluran pernapasan.
3. Kelelahan: Kucing dengan FCV mungkin menunjukkan tanda-tanda kelelahan dan kesulitan bernapas, yang bisa menyebabkan mereka sering berbaring atau menghindari aktivitas fisik.
4. Suara batuk: Infeksi ini dapat menyebabkan batuk yang dapat terdengar seperti suara menggonggong atau batuk yang kering.

Berdasarkan penelitian, berikut adalah rentang frekuensi pernafasan kucing yang sehat dan beberapa indikator klinis yang relevan, dapat dilihat pada Tabel 2.2.

Tabel 2.2 Frekuensi Pernafasan Kucing

Kondisi Kucing	Frekuensi Pernafasan (Napas per Menit)	Keterangan
Kucing Sehat	32 - 135	Median: 64 napas per menit saat di ruang konsultasi [11].
Kucing dengan Masalah Pernafasan	> 135	Frekuensi yang lebih tinggi dapat mengindikasikan adanya masalah pernafasan atau infeksi.

Penjelasan ini didasarkan pada data observasional dan studi klinis yang telah dilakukan untuk mengidentifikasi karakteristik utama dari suara-suara ini dalam konteks infeksi FCV [11].

Infeksi FCV juga dapat menyebabkan pneumonia berat dan kematian mendadak pada beberapa kasus. Virus ini sangat menular dan biasanya ditularkan melalui rute oronasal dan kontak dengan kotoran dari kucing yang terinfeksi [12].



Gambar 2.2 Ilustrasi Kucing Terkena Penyakit Feline Calicivirus
 Sumber: <https://www.wikihow.com/Diagnose-and-Treat-Feline-Calicivirus>

Dalam studi kasus di Klinik K dan P Surabaya, seekor kucing jantan bernama Sapi berusia 9 bulan, dengan berat 1,94 kg, menunjukkan gejala lemas, sempoyongan, mata belekan, dan belum defekasi. Pemeriksaan fisik menunjukkan suhu tubuh 37,2°C, ulser korneal pada mata, gingivitis, dan ulser di lidah. Pemeriksaan penunjang menggunakan Rapid Test Kit menunjukkan hasil positif FCV. Pemeriksaan hematologi mengindikasikan adanya leukositosis, granulocytosis, dan lymphocytosis [12].

Penanganan kucing yang terinfeksi FCV memerlukan pendekatan komprehensif dan intensif, seperti yang diuraikan dalam studi kasus di Klinik K dan P Surabaya [12]. Langkah-langkah penanganan yang diterapkan meliputi:

1. Terapi Cairan dan Elektrolit: Pemberian saline normal dan Neurobion secara intravena untuk menjaga keseimbangan cairan dan elektrolit kucing.
2. Antibiotik: Injeksi Cefotaxime 0,2 ml setiap 12 jam untuk mengatasi infeksi bakteri sekunder yang sering menyertai infeksi virus.
3. Anti-inflamasi dan Kortikosteroid: Injeksi Glucortin 0,25 ml setiap 24 jam untuk mengurangi peradangan dan reaksi imun yang berlebihan.
4. Vitamin dan Suplemen:
 - Injeksi Vitolin 1 ml setiap 24 jam untuk mendukung sistem kekebalan tubuh.
 - Injeksi Calcidex 0,2 ml setiap 12 jam sebagai suplemen kalsium.
 - Injeksi Biodin 0,2 ml sebagai suplemen tambahan.
 - Terapi oral Fibumin 1 kapsul setiap 12 jam.
 - Terapi oral Transfer Factor 1 kapsul setiap 12 jam untuk mendukung respon imun.
5. Perawatan Simptomatik:
 - Oxyfresh water additive 0,5 ml setiap 12 jam untuk menjaga hidrasi.
 - Tetes mata serum setiap 12 jam untuk mengatasi konjungtivitis.
 - Cendo Hyalub setiap 12 jam untuk menjaga kelembaban mata.
 - Penyemprotan Bio Ion pada area scabies di telinga untuk mengatasi infeksi sekunder dan menjaga kebersihan

Untuk mengidentifikasi cepat FCV dalam sampel klinis, metode seperti Reverse Transcriptase Polymerase Chain Reaction (RT-PCR), isolasi virus, dan imunofluoresensi dapat digunakan. Metode ini penting untuk mencegah penyebaran virus dan menghindari kerugian finansial [10].

Penanganan ini memerlukan pengawasan ketat dari dokter hewan dan harus disesuaikan dengan kondisi klinis kucing yang bersangkutan. Pemantauan rutin terhadap respons pengobatan dan kondisi kesehatan umum kucing sangat penting untuk memastikan pemulihan yang optimal dan mencegah komplikasi yang lebih serius.

2.3 Smartphone

Smartphone adalah telepon seluler dengan komputer terintegrasi dan fitur lainnya yang awalnya tidak terkait dengan telepon, seperti sistem operasi (OS), penjelajahan web, dan kemampuan untuk menjalankan aplikasi perangkat lunak. Smartphone beroperasi pada sistem operasi (OS) yang mengelola seluruh operasi dan prosedur telepon. Peran sistem operasi adalah sebagai antarmuka dalam menggunakan perangkat keras dan membantu pengguna menggunakan telepon serta aplikasi-aplikasi di dalamnya [13]. David Wood, Wakil Presiden Eksekutif PT.Symbian OS, menyatakan bahwa sebuah ponsel dapat disebut sebagai smartphone jika ponsel tersebut memiliki sistem operasi tertentu yang mendukung pengelolaan berbagai fitur yang tersedia. Smartphone sepenuhnya mendukung fasilitas email, keyboard QWERTY, layar sentuh, kamera, manajemen kontak, penghitung kecepatan, navigasi perangkat lunak dan perangkat keras, kemampuan membaca dokumen bisnis, pemutar musik, penjelajah foto, menonton klip video, penjelajahan internet, dan lain-lain [14].



Gambar 2.3 Smartphone

Sumber: <https://www.pexels.com/search/mobile%20phone/>

Sejarah smartphone bermula dari era konsep dan eksperimen, ketika teknologi telepon seluler mulai berevolusi menjadi lebih dari sekadar alat komunikasi. Generasi pertama smartphone dinamakan Simon, dirancang oleh IBM pada tahun 1992 dan dipamerkan sebagai konsep produk di COMDEX, sebuah pameran komputer di Las Vegas, Nevada. Simon mulai dipasarkan kepada publik pada tahun 1993 oleh BellSouth. Fitur-fitur yang dimiliki Simon antara lain kalender, buku telepon, jam dunia, pencatat, surel, kemampuan mengirim dan menerima faks, serta permainan. Simon tidak memiliki tombol-tombol fisik, melainkan layar sentuh untuk memilih nomor telepon dengan jari atau membuat faksimile dan memo dengan tongkat stylus. Teks dimasukkan melalui papan ketik "prediksi" yang unik di layar. Meskipun bagi standar masa kini Simon dianggap produk tingkat rendah, fitur-fiturnya pada saat itu sangatlah canggih.

2.4 Android

Android adalah sistem operasi untuk perangkat seluler yang dikembangkan oleh Google, didasarkan pada kernel Linux yang dimodifikasi dan perangkat lunak sumber terbuka lainnya. Sistem ini terutama dirancang untuk perangkat dengan layar sentuh seperti smartphone dan tablet. Android pertama kali dikembangkan oleh Android Inc., perusahaan yang didirikan pada tahun 2003 oleh Andy Rubin dan lainnya. Pada tahun 2005, Google mengakuisisi Android Inc., dan Android

resmi diperkenalkan pada tahun 2007. Perangkat Android komersial pertama, HTC Dream (atau T-Mobile G1), diluncurkan pada September 2008 [15].



Gambar 2.4 Logo Android

Sumber: https://id.wikipedia.org/wiki/Berkas:Android_logo_2019.png

Seiring waktu, banyak pembaruan telah diberikan pada Android, dengan masing-masing versi membawa fitur dan perbaikan baru. Beberapa versi penting termasuk:

1. Android 1.0, versi pertama, menawarkan fitur dasar seperti browser web, kamera, Google Maps, dan sinkronisasi dengan layanan Google.
2. Android 2.0 (Eclair): Diluncurkan pada 2009, mendukung HTML5, wallpaper hidup, dan sinkronisasi akun yang lebih baik.
3. Android 3.0 (Honeycomb): Dirilis pada 2011, dioptimalkan untuk tablet dengan antarmuka pengguna baru dan peningkatan layar besar.
4. Android 4.0 (Ice Cream Sandwich): Diluncurkan pada 2011, menggabungkan elemen dari tablet dan smartphone dengan antarmuka terpadu dan fitur pengenalan wajah untuk membuka kunci.
5. Android 5.0 (Lollipop): Diluncurkan pada 2014, memperkenalkan Material Design yang memberikan antarmuka yang lebih konsisten dan menarik secara visual.
6. Android 6.0 (Marshmallow): Diluncurkan pada 2015, memperkenalkan izin aplikasi baru, sistem manajemen daya, dan dukungan pengenalan sidik jari.
7. Android 7.0 (Nougat): Diluncurkan pada 2016, menghadirkan dukungan multi-jendela, balasan langsung untuk notifikasi, dan mode Doze yang lebih baik untuk penghematan baterai.

8. Android 8.0 (Oreo): Diluncurkan pada 2017, menambahkan fitur seperti mode picture-in-picture, titik notifikasi, dan pengisian otomatis kata sandi.

Google terus mengembangkan dan menyempurnakan Android, menjadikannya sistem operasi seluler paling populer di dunia. Platform ini mendukung berbagai perangkat dan aplikasi, yang membantu dalam adopsi dan fleksibilitasnya yang luas [15].

2.5 Android Studio

Android Studio adalah sebuah Integrated Development Environment (IDE) yang dirancang khusus untuk mengembangkan aplikasi yang berjalan pada platform Android. IDE ini berbasis pada IntelliJ IDEA, yang merupakan sebuah IDE untuk bahasa pemrograman Java.



Gambar 2.5 Software Android Studio

Sumber: https://en.wikipedia.org/wiki/Android_Studio

Bahasa pemrograman utama yang digunakan dalam Android Studio adalah Java, sementara untuk membuat tampilan atau layout, digunakan bahasa XML. Android Studio juga terintegrasi dengan Android Software Development Kit (SDK) yang memungkinkan aplikasi untuk di-deploy ke perangkat Android.

Android Studio adalah evolusi dari Eclipse yang telah dikembangkan menjadi lebih kompleks dan profesional, menyatukan berbagai alat pengembangan dalam satu IDE. Di dalam Android Studio, sudah terdapat Android SDK tools yang memudahkan pengembangan aplikasi [16]. Setiap proyek di Android Studio terdiri dari satu atau beberapa modul yang mencakup file kode sumber dan file sumber daya. Jenis-jenis modul tersebut antara lain:

1. Modul Aplikasi Android: Modul ini berisi kode sumber dan file sumber daya yang dibutuhkan untuk membangun aplikasi Android.
2. Modul Pustaka: Modul ini digunakan untuk membuat pustaka yang dapat dibagikan dan digunakan kembali oleh proyek aplikasi lain.
3. Modul Google App Engine: Modul ini memungkinkan integrasi dengan Google App Engine untuk mengembangkan aplikasi berbasis cloud.

Tambahan fitur yang ditawarkan oleh Android Studio mencakup editor kode yang cerdas dengan dukungan lengkap untuk fitur refactoring, penganalisis kode, serta alat debugging dan pengujian yang kuat. Dengan kemampuan untuk menjalankan emulator Android yang cepat dan integrasi dengan alat pengembangan Google lainnya, Android Studio menjadi pilihan utama bagi para pengembang aplikasi Android [17].

2.6 Firebase

Firebase adalah platform pengembangan aplikasi yang dikembangkan oleh Google untuk membantu para pengembang dalam membangun, meningkatkan, dan mengelola aplikasi. Firebase menyediakan berbagai layanan yang sangat berguna dalam pengembangan aplikasi modern, baik untuk mobile maupun web [18].



Gambar 2.6 Logo Firebase
Sumber: <https://firebase.google.com/>

1. Authentication

Firebase Authentication menyediakan layanan autentikasi yang mudah digunakan, memungkinkan pengembang untuk mengautentikasi pengguna menggunakan email dan kata sandi, nomor telepon, serta melalui penyedia identitas federasi seperti Google, Facebook, dan Twitter. Fitur ini mengurangi waktu dan

usaha yang diperlukan untuk membangun sistem autentikasi yang aman dan dapat diandalkan.

2. Realtime Database

Firebase Realtime Database adalah basis data NoSQL yang memungkinkan data disimpan dan disinkronkan antar pengguna secara real-time. Perubahan data dapat langsung dilihat oleh semua klien yang terhubung, yang sangat berguna untuk aplikasi yang memerlukan update data secara instan, seperti aplikasi chat atau kolaborasi.

3. Cloud Firestore

Firestore adalah database NoSQL yang skalabel dan fleksibel untuk pengembangan mobile, web, dan server. Berbeda dengan Realtime Database, Firestore menawarkan kueri yang lebih kompleks, struktur data yang lebih fleksibel, dan dukungan offline yang lebih baik.

4. Cloud Storage

Firebase Cloud Storage memungkinkan penyimpanan dan pengunduhan file, seperti gambar, video, dan dokumen, dengan cara yang mudah dan aman. Fitur ini sangat penting untuk aplikasi yang memerlukan penyimpanan media dalam jumlah besar.

5. Cloud Functions

Cloud Functions for Firebase adalah solusi serverless yang memungkinkan pengembang untuk menjalankan kode backend sebagai respons terhadap event yang dipicu oleh Firebase dan layanan Google Cloud lainnya. Ini memudahkan dalam menambahkan logika backend tanpa perlu mengelola server secara manual.

6. Firebase Analytics

Firebase Analytics adalah layanan analitik gratis yang menyediakan wawasan tentang penggunaan aplikasi dan interaksi pengguna. Dengan Firebase Analytics, pengembang dapat melacak berbagai metrik dan membuat laporan kustom untuk memahami bagaimana pengguna berinteraksi dengan aplikasi.

7. Firebase Crashlytics

Firebase Crashlytics adalah layanan pelaporan kerusakan ringan yang membantu pengembang untuk melacak, memprioritaskan, dan memperbaiki

masalah stabilitas yang mengurangi kualitas aplikasi. Ini memberikan laporan kerusakan yang terperinci yang memudahkan pengembang untuk menemukan dan memperbaiki bug.

8. Firebase Performance Monitoring

Firebase Performance Monitoring membantu pengembang untuk mendapatkan wawasan tentang kinerja aplikasi di berbagai perangkat dan kondisi jaringan. Dengan informasi ini, pengembang dapat mengoptimalkan kinerja aplikasi untuk memberikan pengalaman pengguna yang lebih baik.

2.7 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer, termasuk ponsel. Bahasa ini diciptakan oleh James Gosling saat masih bekerja di Sun Microsystems, yang kini menjadi bagian dari Oracle, dan dirilis pada tahun 1995. Java mengadopsi banyak sintaksis dari C dan C++, namun dengan model objek yang lebih sederhana dan dukungan untuk rutinitas dari atas ke bawah yang minimal.



Gambar 2.7 Java Programming Language

Sumber: https://en.wikipedia.org/wiki/Java_%28programming_language%29

Aplikasi berbasis Java umumnya dikompilasi menjadi p-code (bytecode) yang dapat dijalankan pada berbagai mesin Virtual Java (JVM). Java adalah bahasa pemrograman yang bersifat umum (general-purpose) dan dirancang khusus untuk meminimalkan ketergantungan pada implementasi tertentu. Karena kemampuannya untuk membuat aplikasi yang dapat berjalan di berbagai platform

sistem operasi, Java dikenal dengan slogannya, "Tulis sekali, jalankan di mana saja." Saat ini, Java adalah salah satu bahasa pemrograman yang paling populer dan digunakan secara luas dalam pengembangan berbagai jenis perangkat lunak aplikasi maupun aplikasi berbasis web.

Java adalah bahasa pemrograman yang multiplatform dan multidevice. Setelah menulis program dengan Java, Anda dapat menjalankannya di hampir semua komputer dan perangkat lain yang mendukung Java, dengan sedikit atau tanpa perubahan pada kodenya. Aplikasi berbasis Java dikompilasi ke dalam p-code dan dapat dijalankan menggunakan mesin virtual Java. Fungsi Java ini dapat berjalan di berbagai platform sistem operasi karena sifatnya yang umum dan tidak spesifik [19].

2.8 Microphone

Mikrofon adalah perangkat yang berguna untuk memperkuat suara berintensitas rendah atau untuk menangkap suara agar intensitasnya dapat diubah sesuai kebutuhan. Mikrofon merupakan sejenis transduser yang mengubah energi akustik atau gelombang suara menjadi sinyal listrik. Mikrofon digunakan dalam berbagai perangkat seperti telepon, alat perekam, dan penguat suara. Berdasarkan komponen, fungsi, dan bentuknya, terdapat empat jenis mikrofon, yaitu:

1. Mikrofon dinamik
2. Mikrofon kondensor
3. Mikrofon pita (ribbon)
4. Mikrofon piezoelektrik.

Mikrofon dinamik, yang sering digunakan, memiliki membran tipis untuk menerima tekanan suara, dan cara kerjanya mirip dengan speaker. Mikrofon kondensor bekerja berdasarkan prinsip elektrostatik dan memiliki akurasi sangat tinggi. Mikrofon pita menggunakan pita konduktif listrik yang bergetar dalam medan magnet, mirip dengan mikrofon dinamik. Mikrofon piezoelektrik, sering disamakan dengan mikrofon kristal atau keramik, sangat tahan banting dan kuat, tetapi memiliki kualitas suara yang kurang baik, biasanya digunakan dalam telepon.

Boomer atau gun mic adalah jenis mikrofon kondensor yang dapat menangkap suara dari jarak jauh, biasanya digunakan dalam industri perfilman untuk merekam suara dalam pengambilan adegan. Tim operator boomer biasanya terdiri dari tiga hingga lima orang, satu orang bertugas untuk mixing suara dan lainnya bergantian memegang boomer. Tantangan yang dihadapi oleh operator boomer adalah akurasi tinggi dari mikrofon ini, sehingga sering kali menangkap suara yang tidak diinginkan seperti angin atau suara objek di sekitarnya [20].

2.9 Kamera

Kamera adalah perangkat yang digunakan untuk menangkap gambar, baik diam maupun bergerak, yang merupakan hasil proyeksi melalui sistem lensa. Seperti manusia yang memiliki sejarah dan perjalanan sendiri, kamera juga memiliki evolusi sejarahnya.



Gambar 2.8 Kamera Smartphone

Sumber: <https://www.idntimes.com/tech/gadget/kena-zakiah-1/aplikasi-setting-kamera-hp-manual-terbaik>

Awalnya, kamera dikenal sebagai camera obscura, yang berarti ruang gelap dalam bahasa Latin. Camera obscura adalah alat yang terdiri dari ruang gelap atau kotak yang memantulkan cahaya melalui dua lensa konveks, menghasilkan gambar objek eksternal pada kertas atau film yang ditempatkan di pusat fokus lensa tersebut.

Di masa lalu, kamera memiliki bentuk yang besar dan kaku. Namun, kamera modern telah berkembang menjadi lebih canggih, sederhana, portabel, praktis, dan bergaya [21].

2.10 Library

Library adalah lapisan tempat berbagai fitur tambahan Android berada. Umumnya, library digunakan untuk mempermudah dan mempercepat pembuatan aplikasi. Berikut adalah jenis library dibagian Interface.

1. Duo Navigation Drawer

Library ini menyediakan cara mudah untuk membuat navigasi alternatif di Android. Perpindahan antar aktivitas menjadi lebih menarik karena aktivitas utama akan digantikan oleh aktivitas atau menu lain. Secara default, ini menciptakan efek perputaran pada konten dan menu. Library ini mendukung minimal SDK versi 15 (Android 4.0.3–4.0.4 Ice Cream Sandwich) dan dikembangkan oleh PSD-Company.

2. MP Android Chart

MP Android Chart adalah library yang menyediakan berbagai jenis grafik, sangat berguna jika aplikasi yang dikembangkan berkaitan dengan data yang perlu divisualisasikan dalam bentuk grafik. Menyajikan data dalam bentuk grafik tentu membuatnya lebih menarik. Beberapa jenis grafik yang tersedia antara lain:

- Simple Bar Chart
- Grouped Bar Chart
- Horizontal Bar Chart
- Simple Line Chart
- Line Chart with Cubic Lines
- Grouped Line Chart
- Combined Line and Bar Chart
- Pie Chart
- Scatter Chart
- Candlestick Chart
- Radar Chart

3. Navigation Bar Icons Morphing

Library ini menyediakan navigasi yang menarik dengan ikon yang berpindah dan menandakan menu aktif saat perpindahan menu. Library ini dikembangkan oleh

Ahmed Tarek (tarek360) pada tanggal 17 Agustus 2017 dan tergolong baru. Navigation Bar Icons Morphing mendukung minimal SDK versi 14 (Android 4.0–4.0.2 Ice Cream Sandwich).

Dan masih banyak library lainnya yang dapat digunakan sesuai kebutuhan aplikasi [22].

2.11 Dataset

Dataset adalah sebuah basis data yang disimpan dalam memori (in-memory). Dataset memiliki semua karakteristik, fitur, dan fungsi yang dimiliki oleh basis data konvensional. Dataset dapat berisi banyak tabel yang bisa saling berhubungan satu sama lain. Tabel-tabel dalam sebuah dataset dapat memiliki kunci asing (foreign key) dan menjaga integritas referensial. Dataset adalah objek yang merepresentasikan data dan relasinya di dalam memori, dengan struktur yang serupa dengan yang ada di basis data [23]. Dataset berisi kumpulan tabel data dan datanya. Terdapat dua jenis dataset:

1. Private Dataset

Dataset jenis ini diperoleh dari organisasi yang dijadikan objek penelitian, seperti instansi, rumah sakit, pabrik, perusahaan jasa, dan sebagainya.

2. Public Dataset

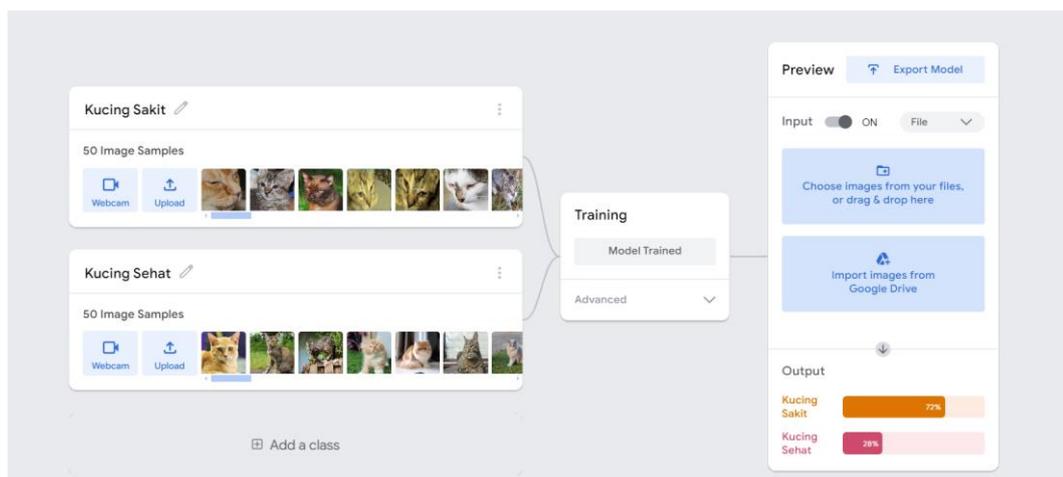
Dataset jenis ini dapat diambil dari repository publik yang telah disepakati oleh para peneliti, seperti UCI, Kaggle, dan RoboFlow.

2.12 Teachable Machine

Teachable Machine adalah alat yang dapat digunakan untuk membuat model klasifikasi yang mudah digunakan dalam pengembangan aplikasi machine learning. Teachable Machine adalah aplikasi berbasis web yang mampu membantu membuat model machine learning dengan cepat, mudah, dan dapat diakses oleh semua orang [24]. Alat ini disediakan oleh Google dan menggunakan sistem pembelajaran untuk menganalisis data tanpa harus diprogram secara eksplisit.

Teachable Machine memiliki fitur yang mampu memproses gambar, suara, dan bahkan gerakan. Tujuannya adalah untuk memudahkan pelajar, guru, desainer, dan profesional di bidang lainnya dalam memahami kecerdasan buatan dengan membuat model klasifikasi mereka sendiri.

Proses pelatihan data adalah proses yang memerlukan banyak sumber daya dalam bentuk kumpulan data objek bibit tanaman. Setiap objek bibit tanaman harus memiliki berbagai bentuk citra yang beragam untuk meningkatkan akurasi hasil klasifikasi. Semakin banyak data objek yang tersedia, semakin lama waktu yang diperlukan untuk melakukan proses pelatihan, sehingga dapat menghasilkan model klasifikasi yang akurat [25]. Pada Gambar 2.9, berikut ini adalah ilustrasi dari proses pelatihan data menggunakan Teachable Machine.



Gambar 2.9 Proses Training dengan Teachable Machine

2.13 TensorFlow Lite

TensorFlow Lite adalah library machine learning lintas platform yang dioptimalkan untuk menjalankan model machine learning di perangkat edge, termasuk perangkat seluler Android dan iOS. TensorFlow Lite sebenarnya adalah mesin inti yang digunakan di dalam ML Kit untuk menjalankan model machine learning. Ada dua komponen dalam ekosistem TensorFlow Lite yang memudahkan pelatihan dan deployment model machine learning pada perangkat seluler:

1. Sebuah library Python yang memudahkan pelatihan model TensorFlow Lite menggunakan data sendiri hanya dengan beberapa baris kode, tanpa memerlukan keahlian mendalam dalam machine learning.
2. Sebuah library lintas platform yang memudahkan deployment model TensorFlow Lite hanya dengan beberapa baris kode di aplikasi.

Fitur dan Kelebihan TensorFlow Lite:

TensorFlow Lite dirancang untuk menyediakan latensi rendah dan ukuran model yang kecil, memungkinkan aplikasi machine learning berjalan dengan efisien di perangkat dengan keterbatasan sumber daya. Beberapa fitur dan kelebihan utama TensorFlow Lite meliputi:

1. TensorFlow Lite mendukung berbagai teknik optimisasi seperti kuantisasi dan pruning yang mengurangi ukuran model dan meningkatkan kecepatan inferensi tanpa mengorbankan terlalu banyak akurasi.
2. TensorFlow Lite menggunakan interpreter yang ringan dan cepat untuk mengeksekusi model di perangkat. Interpreter ini dapat berjalan pada berbagai platform termasuk Android, iOS, dan perangkat IoT.
3. TensorFlow Lite menyediakan alat konversi yang mengubah model TensorFlow yang telah dilatih menjadi format TensorFlow Lite. Converter ini mendukung berbagai optimisasi untuk memperkecil ukuran model dan mempercepat inferensi.
4. TensorFlow Lite mendukung akselerasi perangkat keras seperti GPU dan TPU untuk meningkatkan performa model di perangkat seluler dan edge.

Penggunaan dalam Aplikasi Seluler:

TensorFlow Lite memudahkan integrasi model machine learning dalam aplikasi seluler dengan menyediakan berbagai API dan dokumentasi yang mendetail. Ini memungkinkan pengembang untuk menambahkan kemampuan machine learning ke dalam aplikasi mereka dengan mudah dan efisien [26].

2.14 Unified Modeling Language

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan dalam pengembangan perangkat lunak dan sistem. UML disebut

"unified" karena menggabungkan berbagai metode pemodelan yang sebelumnya terpisah menjadi satu standar yang komprehensif. UML dapat digunakan untuk memodelkan berbagai aspek sistem, termasuk struktur, perilaku, dan interaksi komponen perangkat lunak dan perangkat keras.

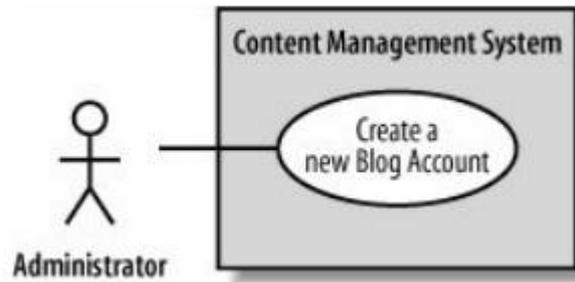
Pemodelan menggunakan UML membantu mengelola kompleksitas desain sistem. Sistem, mulai dari aplikasi desktop sederhana hingga sistem berskala besar multi-tier, terdiri dari ratusan hingga ribuan komponen. Pemodelan memungkinkan tim pengembang untuk melacak komponen yang dibutuhkan, fungsi masing-masing komponen, dan bagaimana komponen tersebut memenuhi kebutuhan pelanggan. Selain itu, UML memfasilitasi komunikasi antar anggota tim untuk memastikan bahwa semua bagian sistem bekerja bersama dengan baik.

Model adalah abstraksi dari sistem nyata, yang menyederhanakan kompleksitas dengan menghilangkan detail yang tidak relevan atau membingungkan. Model yang dibuat dengan UML memungkinkan desain dan kelayakan sistem untuk dipahami, dievaluasi, dan dikritik lebih cepat dibandingkan jika harus menganalisis sistem nyata secara langsung. Dengan bahasa pemodelan formal seperti UML, model yang dihasilkan bersifat abstrak namun tetap presisi seperti bahasa pemrograman, sehingga dapat dibaca oleh mesin, dieksekusi, dan diubah antar sistem.

UML menyediakan bahasa yang diperlukan untuk mendeskripsikan model sistem secara efektif, memungkinkan para pengembang untuk menangkap, mendokumentasikan, dan mengkomunikasikan aspek-aspek penting dari desain sistem mereka [27].

2.14.1 Use Case Diagram

Use Case Diagram adalah salah satu alat utama dalam Unified Modeling Language (UML) yang digunakan untuk menangkap interaksi antara aktor eksternal dan sistem yang sedang dikembangkan. Use Case Diagram membantu dalam mendokumentasikan fungsionalitas sistem dari perspektif pengguna dan mendefinisikan tujuan atau hasil yang diinginkan oleh pengguna eksternal.



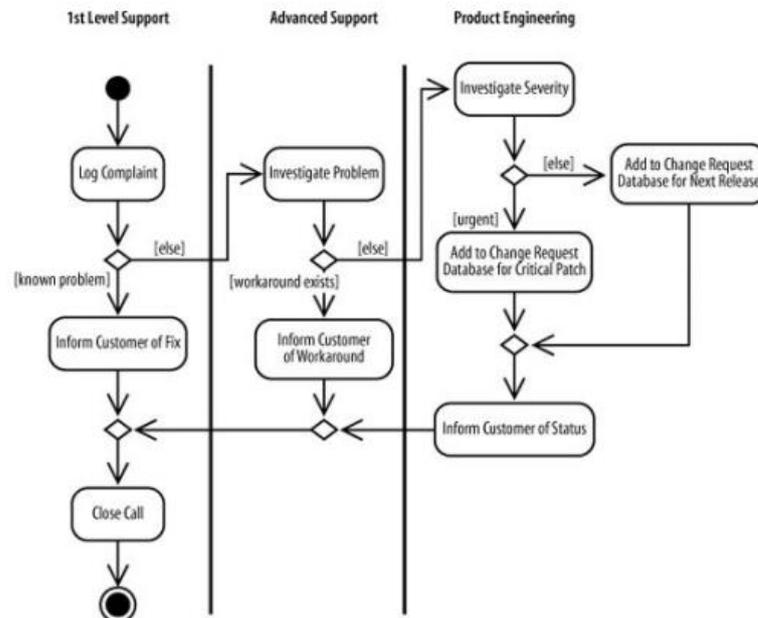
Gambar 2.10 Use Case Diagram Learning UML 2.0

1. Use Case: Dilambangkan dengan oval yang mencakup nama yang menjelaskan interaksi atau fungsi yang diwakilinya. Meskipun sederhana, notasi ini sangat penting dalam menangkap kebutuhan sistem [27].
 2. Actor: Representasi dari pengguna atau sistem eksternal yang berinteraksi dengan sistem yang dimodelkan. Aktor biasanya digambarkan dengan stick figure di luar batas sistem.
 3. Communication Line: Garis yang menghubungkan aktor dan use case untuk menunjukkan partisipasi aktor dalam use case tersebut [27].
 4. System Boundary: Garis kotak yang mengelilingi semua use case untuk menunjukkan batas sistem, sedangkan aktor berada di luar batas ini [27].
1. Hubungan dalam Use Case Diagram
 - Include: Digunakan untuk menunjukkan bahwa sebuah use case mencakup fungsionalitas dari use case lain. Hal ini digunakan untuk menghindari pengulangan dan menunjukkan alur opsional yang penting dalam diagram use case [27].
 - Extend: Digunakan untuk menunjukkan alur opsional atau perilaku khusus yang ditambahkan ke use case utama.
 2. Pentingnya Use Case Diagram:
 - Diagram ini mempermudah komunikasi antara tim pengembang dan pemangku kepentingan dengan menyediakan gambaran yang jelas tentang bagaimana sistem seharusnya berperilaku.

- Membantu dalam mengelola kompleksitas sistem dengan memecah fungsionalitas ke dalam use case yang terdefinisi dengan baik.
- Membantu dalam memastikan bahwa semua kebutuhan sistem telah ditangkap dan dipahami dengan benar oleh semua pihak yang terlibat.

2.14.2 Activity Diagram

Activity Diagram adalah diagram UML yang digunakan untuk memodelkan proses atau aliran kerja dalam sistem. Diagram ini sangat berguna dalam menggambarkan perilaku dinamis dari sistem dengan fokus pada aliran aktivitas dan proses yang terjadi di dalamnya [27].



Gambar 2.11 Activity Diagram Learning UML 2.0

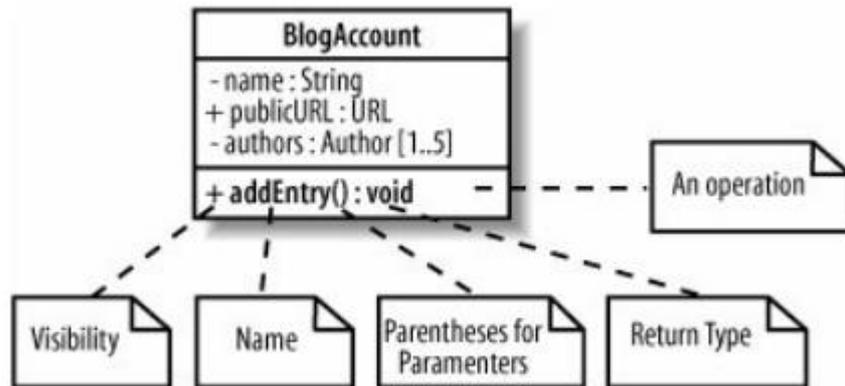
Berikut adalah beberapa elemen penting dan konsep dasar dari activity diagram yang diambil dari buku "Learning UML 2.0":

1. Initial Node, menandai titik awal dari aktivitas. Digambar sebagai lingkaran penuh.
2. Activity Final Node, menandai akhir dari aktivitas. Digambar sebagai dua lingkaran konsentris dengan lingkaran dalam terisi penuh.

3. Actions adalah langkah-langkah penting yang terjadi dalam keseluruhan aktivitas. Digambar sebagai persegi panjang dengan sudut membulat. Setiap action dapat berupa perilaku yang dilakukan, perhitungan, atau langkah kunci dalam proses.
4. Edges/Paths, Aliran aktivitas ditunjukkan dengan garis berpanah yang disebut edges atau paths. Panah menunjukkan arah aliran dari satu action ke action berikutnya.
5. Decision nodes adalah node berbentuk wajik yang digunakan untuk menentukan cabang aliran berdasarkan kondisi Boolean. Setiap cabang keluar dari decision node dilabeli dengan kondisi yang menentukan aliran mana yang akan diikuti.
6. Merge nodes menggabungkan aliran yang berasal dari decision node, menandai akhir dari perilaku kondisional.
7. Activity frame adalah persegi panjang dengan sudut membulat yang digunakan untuk membatasi sebuah aktivitas dan semua action yang terkait. Meskipun opsional, activity frame membantu menunjukkan lebih dari satu aktivitas dalam satu diagram.

2.14.3 Class Diagram

Class Diagram merupakan salah satu jenis diagram dalam UML yang digunakan untuk menggambarkan struktur statis dari suatu sistem. Diagram ini menunjukkan kelas-kelas dalam sistem dan hubungan di antara mereka.



Gambar 2.12 Class Diagram Learning UML 2.0

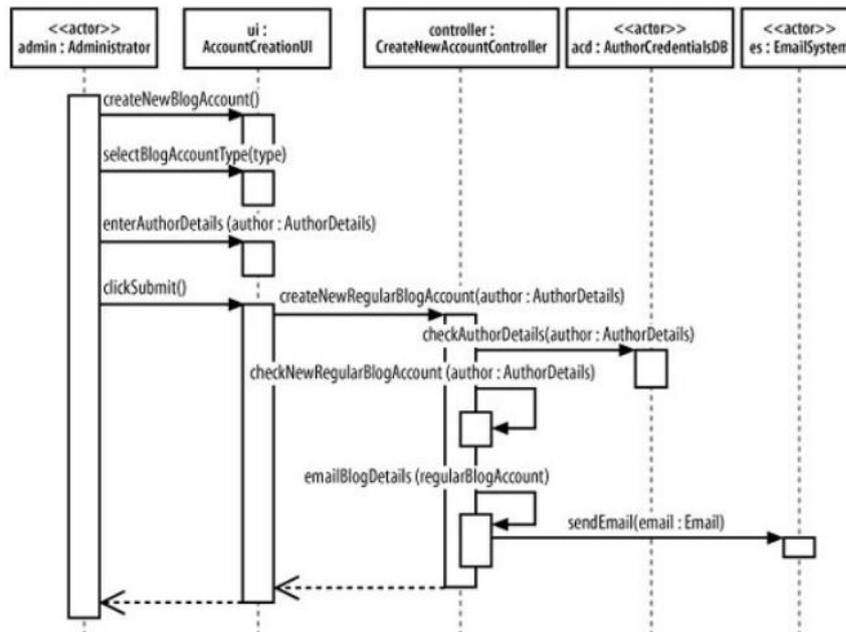
Berikut ini penjelasan mengenai elemen-elemen utama dari class diagram sesuai dengan buku "Learning UML 2.0":

1. Kelas, merepresentasikan objek-objek yang ada dalam sistem. Setiap kelas digambarkan sebagai persegi panjang yang dibagi menjadi tiga bagian: nama kelas, atribut, dan operasi. Nama kelas ditempatkan di bagian atas, atribut di bagian tengah, dan operasi di bagian bawah.
2. Atribut adalah properti atau karakteristik dari kelas. Mereka menunjukkan data apa yang disimpan oleh objek dari kelas tersebut. Atribut ditulis dengan format visibility name: type, di mana visibility bisa berupa + (public), - (private), atau # (protected).
3. Operasi adalah fungsi atau metode yang dapat dilakukan oleh objek dari kelas tersebut. Operasi ditulis dengan format visibility name(parameters): returnType.
4. Hubungan (Relationships):
 - Asosiasi (Association), Menunjukkan hubungan antara dua kelas. Asosiasi dapat diberi label untuk menjelaskan peran dari masing-masing kelas dalam hubungan tersebut.
 - Agregasi (Aggregation), Jenis khusus dari asosiasi yang menunjukkan hubungan "whole-part". Agregasi digambarkan dengan garis dan ujung berlian kosong pada sisi kelas "whole".

- Komposisi (Composition), Mirip dengan agregasi, namun menunjukkan hubungan "whole-part" yang lebih kuat di mana bagian tidak dapat berdiri sendiri. Komposisi digambarkan dengan garis dan ujung berlian penuh pada sisi kelas "whole".
 - Generalization (Inheritance), Menunjukkan hubungan pewarisan antara kelas induk (superclass) dan kelas anak (subclass). Digambarkan dengan garis solid dengan panah terbuka mengarah ke superclass.
5. Kardinalitas (Multiplicity), Menunjukkan jumlah minimum dan maksimum objek yang dapat berpartisipasi dalam suatu hubungan. Kardinalitas ditulis di ujung garis asosiasi.
 6. Interface adalah kumpulan operasi yang dapat diimplementasikan oleh kelas. Interface digambarkan sebagai kelas dengan label <<interface>> atau sebagai lingkaran dengan nama interface.
 7. Template (Generic Class), Kelas atau metode yang dapat beroperasi pada tipe data yang spesifik. Template digambarkan dengan kurung siku di dalam nama kelas atau metode.
 8. Constraint, kendala atau batasan yang mengatur bagaimana kelas dan hubungan mereka berinteraksi. Kendala ditulis dalam kurung kurawal { }.

2.14.4 Sequence Diagram

Sequence diagram adalah jenis diagram dalam UML yang digunakan untuk menggambarkan urutan interaksi antar objek dalam sistem berdasarkan waktu. Interaksi ini terutama berbentuk pesan yang dikirim antar peserta (participants) di dalam sistem[27].



Gambar 2.13 Sequence Diagram Learning UML 2.0

Berikut adalah beberapa elemen penting dan konsep dasar dari sequence diagram yang diambil dari buku "Learning UML 2.0":

Berdasarkan buku "Learning UML 2.0", berikut adalah elemen-elemen utama yang digunakan dalam sequence diagram beserta penjelasannya:

1. Lifeline, mewakili keberadaan individu objek dalam sistem. Lifeline digambarkan sebagai garis vertikal yang dimulai dari peserta (participant) di bagian atas diagram. Contoh: Objek pengguna, sistem, atau komponen internal tertentu.
2. Participant adalah objek atau aktor yang terlibat dalam interaksi. Nama participant ditempatkan di kotak di bagian atas lifeline. Contoh: User, Admin, System.
3. Messages adalah komunikasi antara participant yang digambarkan dengan panah dari pengirim ke penerima. Tipe Pesan:
 - Synchronous Message: Digambarkan dengan panah solid dengan ujung panah penuh. Mengindikasikan bahwa pengirim menunggu respons sebelum melanjutkan.
 - Asynchronous Message: Digambarkan dengan panah solid dengan ujung panah terbuka. Pengirim tidak perlu menunggu respons untuk melanjutkan.

- Return Message: Digambarkan dengan panah putus-putus. Mengindikasikan pengembalian informasi atau respon dari penerima ke pengirim.
4. Activation bar menunjukkan periode waktu ketika participant melakukan tindakan atau berada dalam keadaan aktif. Digambarkan sebagai persegi panjang vertikal pada lifeline. Contoh: Saat sistem memproses permintaan dari pengguna.
 5. Fragments adalah blok yang mengelilingi bagian dari interaksi dalam sequence diagram, digunakan untuk menangani aliran kontrol kompleks.
 6. Combined Fragments, gabungan dari beberapa sequence fragments untuk menangani kondisi dan loop yang lebih kompleks. Contoh: Kondisi if-else yang memerlukan fragment 'alt' untuk menggambarkan jalur yang berbeda.
 7. Creation and Destruction Messages Menunjukkan penciptaan atau penghancuran objek dalam diagram. Tipe nya:
 - Create Message: Digambarkan dengan panah dari pengirim ke participant baru.
 - Destroy Message: Digambarkan dengan panah ke 'X' di lifeline penerima untuk menunjukkan penghancuran.
 8. Guard Conditions Kondisi penjaga yang mengontrol aliran pesan dalam sequence diagram. Digambarkan dalam kurung siku di sepanjang panah pesan. Contoh: [valid], [error], [success].