BAB 2 TINJAUAN PUSTAKA

Pada bab ini kami menjabarkan dasar teori dari penelitian yang akan dilakukan. Pembahasan dimulai dengan konsep dasar yang mendukung penelitian, termasuk definisi serta prinsip yang relevan. Selanjutnya, dijelaskan teori-teori yang berkaitan dengan permasalahan yang dikasih, serta metode yang digunakan dalam penelitian ini.

2.1. *Kebun Raya Balikpapan (KRB)*

Kebun Raya atau *Botanical Garden* merupakan kawasan dimana berbagai jenis tumbuhan dikumpulkan, ditanam, dan dipamerkan dengan tujuan utama untuk mendukung aktivitas ilmiah dan pendidikan. Kebun Raya umumnya memiliki koleksi tumbuhan yang dikelola di tempat terbuka maupun dalam rumah kaca. Selain itu, kawasan ini juga dilengkapi dengan berbagai fasilitas seperti ruang belajar, laboratorium, perpustakaan, museum, dan area penelitian tanaman (Ensiklopedia Encarta) (Hidayat., 2013).

Kebun Raya Balikpapan merupakan hasil kolaborasi antara Pemerintah Kota Balikpapan, Pusat Konservasi Tumbuhan Kebun Raya Bogor, dan Balai Besar Penelitian Dipterocarpaceae di bawah naungan Kementerian Kehutanan. Kawasan Kebun Raya Balikpapan secara resmi ditetapkan melalui Keputusan Menteri Kehutanan RI Nomor 68/Menhut-II/2009 pada tanggal 26 Februari 2009. Keputusan ini mengesahkan area hutan seluas 309,22 hektar yang berada di dalam wilayah Hutan Lindung Sungai Wain (HLSW) sebagai kawasan dengan fungsi khusus yang difokuskan pada kegiatan penelitian, pengembangan, pendidikan, dan pelatihan dalam bentuk kebun raya (Usmadi et al., 2015).

Kebun Raya Balikpapan tidak hanya berfungsi sebagai tempat untuk melestarikan tanaman, tetapi juga sebagai pusat pembelajaran dan penelitian yang membantu mengembangkan ilmu pengetahuan dalam bidang botani. Di sini, berbagai jenis tanaman langka dan endemik, termasuk yang dilindungi, dirawat untuk menjaga keragaman hayati. Kebun Raya Balikpapan juga menawarkan berbagai program edukasi dan pelatihan yang bisa diikuti oleh masyarakat, siswa, dan peneliti untuk meningkatkan pemahaman tentang pentingnya menjaga alam dan tanaman. Dengan berbagai fasilitas seperti ruang belajar dan laboratorium, kebun ini menjadi tempat yang sangat cocok untuk riset ilmiah dan edukasi kepada masyarakat mengenai pentingnya konservasi lingkungan.

Gambar 2.1 Kebun Raya Balikpapan

2.1.1. *Pengenalan Klasifikasi Otomatis Pada Tumbuhan*

Klasifikasi merupakan metode analisis data yang bertujuan untuk mengidentifikasi pola serta membentuk model yang mampu membedakan berbagai kategori data yang dianggap penting. Model yang dihasilkan, dikenal sebagai classifier, digunakan untuk memprediksi label kelas dalam bentuk nilai kategoris yang bersifat diskrit dan tidak memiliki urutan (Ha dkk., 2011). Pendekatan klasifikasi tradisional telah digunakan sejak lama dalam tugas pengelompokan data. Seiring dengan perkembangan teknologi, pendekatan berbasis Deep Learning (DL) semakin dikenal luas dan banyak digunakan dalam berbagai bidang.

Pada penelitian yang dilakukan oleh Dhani dkk. (2024) mereka membandingkan performa model pembelajaran transfer ResNet dalam klasifikasi tumbuhan. Dalam penelitian ini, mereka menggunakan dataset tumbuhan dari Kebun Raya Balikpapan yang terdiri dari 52 kelas tumbuhan dengan 40 citra per kelas. Model yang diuji mencakup beberapa varian ResNet, yaitu ResNet-34, ResNet-50, dan ResNet-101. Hasil penelitian menunjukkan bahwa model ResNet mampu memberikan performa yang baik dalam klasifikasi tumbuhan, dengan akurasi yang bervariasi tergantung pada arsitektur yang digunakan. ResNet-34 mencapai akurasi sebesar 93.32%, ResNet-50 memperoleh akurasi

tertinggi sebesar 96.06%, sedangkan ResNet-101 mencatat akurasi sebesar 95.38%. Dengan demikian, ResNet-50 menunjukkan performa terbaik dalam klasifikasi tumbuhan, yang mengindikasikan bahwa arsitektur ini memiliki keseimbangan optimal antara kedalaman jaringan dan efisiensi dalam ekstraksi fitur.

2.2. *Model* RestNet-50

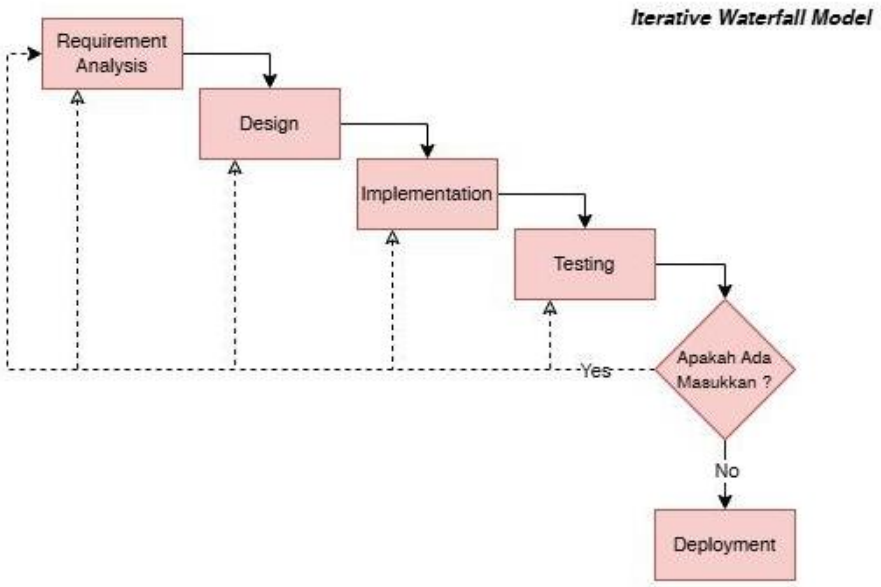
RestNet50 adalah model jaringan saraf yang memungkinkan pembuatan jaringan yang sangat dalam, bahkan lebih dari 150 lapisan. Model ini memiliki 50 lapisan yang telah dilatih menggunakan dataset ImageNet dan memanfaatkan arsitektur jaringan yang sangat mendalam. Keunggulan dari RestNet-50 terletak pada penggunaan *blok bottleneck* yang meningkatkan efisiensi model. Model ini terdiri dari lima blok jaringan saraf konvolusional yang saling terhubung melalui jalur pintas atau *shortcut*. *Deep Residual Features* (DRF) diambil dari lapisan konvolusi terakhir untuk menghasilkan performa yang optimal (Shukla, & Tiwari,. 2023).

ResNet-50 banyak digunakan dalam berbagai aplikasi berbasis *computer vision*, termasuk deteksi objek, klasifikasi gambar, dan segmentasi citra. Model ini mampu mengatasi permasalahan degradasi akurasi yang sering terjadi pada jaringan saraf dalam dengan memanfaatkan residual learning. Selain itu Dhani dkk (2024) juga menggunakan ResNet-50 dalam penelitian mereka untuk meningkatkan akurasi klasifikasi gambar pada dataset tumbuhan dengan jumlah kelas sebanyak 52. Mereka menunjukkan bahwa penggunaan teknik fine-tuning pada model ini dapat mengoptimalkan kinerja dalam mendeteksi objek dengan latar belakang kompleks. Dengan kemampuannya dalam mengekstraksi fitur tingkat tinggi, ResNet-50 menjadi pilihan utama dikarenakan mendapatkan hasil yang bagus dan akurat dalam pengenalan spesies tumbuhan secara otomatis.

2.3. Iterative Waterfall

Iterative Waterfall merupakan pengembangan dari metode *Waterfall* (Reyza Yana Putra dkk., 2018), di mana setiap tahap harus diselesaikan secara berurutan sebelum melanjutkan ke tahap berikutnya. Berbeda dengan model *Waterfall* klasik yang berjalan lurus, *Iterative Waterfall* memungkinkan perulangan pada tahap tertentu jika masih ada kesalahan atau kekurangan yang perlu diperbaiki. Proses ini terus diulang hingga setiap

tahap benar-benar selesai (Stinjak & Masya, 2021). Metodologi ini dimulai dengan membuat model dasar aplikasi berdasarkan spesifikasi awal proyek. Setelah model diuji dan mendapat masukan, dilakukan perbaikan serta penyesuaian untuk menyempurnakannya. Proses ini terus berulang hingga aplikasi benar-benar siap digunakan dan memenuhi semua kebutuhan pengguna.



Gambar 2.2 *Iterative Waterfall*

Model ini diterapkan dengan cara perulangan, dimana proyek pengembangan perangkat lunak menjadi bagian kecil yang disebut feedback. Dengan pendekatan ini, tim pengembang bisa membuat bagian-bagian perangkat lunak yang bisa diuji dan mendapatkan masukan dari pengguna secara berulang. Setiap feedback memiliki proses yang mirip dengan *mini-waterfall*, dimana setiap tahap memberikan umpan balik yang penting untuk tahap berikutnya. Jadi, proses *design*, *implementation*, *testing*, dan *deployment* akan dilakukan berulang kali untuk setiap iterasi sampai seluruh proyek selesai dengan baik tanpa harus ada kesalahan (Nur Adiya dkk., 2024).

2.3.1. *Requirement Analysis*

Requirement Analysis atau analisis kebutuhan adalah tahap dimana peneliti melakukan observasi yang bertujuan untuk menentukan suatu permasalahan yang ada dalam sistem. Tujuannya adalah untuk memahami apa yang dibutuhkan dan menemukan solusi yang tepat agar sistem dapat bekerja lebih efektif. Pada tahap ini, analisis dibagi menjadi dua bagian yaitu analisis kebutuhan dan analisis spesifikasi. Analisis kebutuhan

dilakukan dengan mengamati, mewawancarai pihak terkait, serta mempelajari referensi yang relevan. Hasil dari analisis ini digunakan untuk menentukan fitur-fitur yang akan dikembangkan dalam penelitian ini. Untuk analisis spesifikasi berfokus pada menentukan perangkat lunak dan perangkat keras yang dibutuhkan untuk membangun sistem yang akan dikembangkan (Listiyan & Subhiyakto., 2021).

2.3.2. *Design*

Design adalah proses perancangan sistem dan perangkat lunak agar sesuai dengan kebutuhan yang telah ditentukan dalam sistem. Pada tahap ini, sistem dirancang secara detail untuk memastikan fungsionalitasnya sesuai dengan spesifikasi yang dibutuhkan. Perancangan ini mencakup aspek teknis, struktur data, serta arsitektur sistem yang akan digunakan.

2.3.3. *Implementation*

Implementation merupakan tahap dimana akan dilakukan pengkodean dari desain sistem yang telah dirancang. Pada tahap ini, seluruh rancangan yang telah dibuat sebelumnya mulai direalisasikan dalam bentuk kode. Dalam penelitian ini, pengembangan perangkat lunak akan dilakukan dengan membuat kode sumber yang terbagi ke dalam beberapa bagian atau subprogram, seperti fitur *create*, *read*, *update*, dan *delete* data. Setiap *subprogram* akan dikembangkan secara bertahap dan diuji untuk memastikan bahwa sistem berfungsi dengan baik sesuai dengan kebutuhan yang telah ditentukan.

2.3.4. *Testing*

Testing adalah tahap dimana ketika bagian bagian kecil dari program yang telah dibuat akan digabungkan menjadi satu kesatuan sistem. Setelah proses penggabungan selesai, sistem akan diuji untuk memastikan bahwa semuanya berfungsi dengan kebutuhan. Jika ditemukan kesalahan atau error maka pada tahap ini akan dilakukan perbaikan agar sistem dapat bekerja dengan baik dan siap digunakan.

2.3.5. *Feedback*

Feedback pada iterative waterfall ini adalah tahap di mana perubahan pada aplikasi dapat dilakukan dengan kembali ke tahap sebelumnya, seperti requirement analysis, design, implementation, atau testing, tanpa harus mengulang seluruh proses dari awal. Hal ini memungkinkan perbaikan dan penyempurnaan berdasarkan hasil evaluasi sebelum melanjutkan ke

tahap berikutnya.

2.3.6. *Deployment*

Deployment adalah tahapan dimana sistem yang sudah diuji dan dikembangkan siap digunakan oleh pengguna. Pada tahap ini, sistem akan diunggah ke *server* agar dapat diakses melalui web. Proses ini meliputi pengaturan server, pengelolaan *database*, dan memastikan semua fitur berjalan dengan baik dalam kondisi sebenarnya. Setelah sistem aktif, pengguna dapat langsung menggunakannya tanpa perlu pemeliharaan lebih lanjut.

2.4. *REST API*

REST API (Representational State Transfer) merupakan sebuah pendekatan arsitektur yang digunakan dalam pengembangan aplikasi berbasis jaringan dan sering diterapkan dalam pembuatan layanan web (Kim et al., 2022). Pendekatan ini menyajikan sekumpulan sumber daya yang dapat diakses melalui jaringan, beserta operasi-operasi yang dapat dijalankan terhadap sumber daya tersebut. Operasi tersebut dapat dipanggil oleh berbagai klien HTTP, termasuk kode JavaScript yang dijalankan di peramban. Setiap REST API memiliki jalur utama atau base path yang menjadi titik awal pengaksesan seluruh sumber daya. Penggunaan jalur utama ini memungkinkan pemisahan antar layanan REST API yang berbeda maupun antar versi dari layanan yang sama. Sebagai contoh, sebuah REST API yang dirancang untuk mengelola data mahasiswa dapat menggunakan jalur seperti /studentdb/v1 untuk versi pertama dan /studentdb/v2 untuk versi berikutnya (Surwase, 2016).

Tabel 2.1 Contoh REST API (Surwase, 2016)

Resource	*Description*
/students	*All of the students in the database*
/students/12345	*Student #12345*
/students/12345/orders	*All orders for student #12345*
/students/12345/orders/67890	*Order #67890 for student #12345*

REST API menggambarkan kumpulan sumber daya yang dapat diakses oleh klien HTTP melalui jalur yang ditentukan secara relatif terhadap jalur utama. Jalur dapat disusun secara hierarkis, dan struktur yang dirancang dengan tepat akan membantu pengguna dalam memahami berbagai jenis sumber daya yang tersedia. Dalam contoh basis data

siswa, jalur-jalur mengacu pada berbagai entitas yang bisa diakses. Setiap sumber daya memiliki sejumlah operasi yang dapat dijalankan oleh klien, dan setiap operasi diidentifikasi melalui kombinasi antara nama dan metode HTTP seperti GET, POST, atau DELETE. Nama untuk setiap operasi perlu unik di seluruh sumber daya yang tersedia, dan satu sumber daya hanya diperbolehkan memiliki satu operasi yang menggunakan metode HTTP tertentu. Kombinasi antara jalur dan metode HTTP yang dikirimkan oleh klien digunakan untuk mengenali sumber daya serta operasi yang ingin dijalankan (Surwase, 2016).

Tabel 2.2 Contoh HTTP REST API (Surwase, 2016)

HTTP Method	*Operation Name*	*Description*
GET	*getStudent*	*Retrieve the student details from the database*
UPDATE	*updateStudent*	*Update the student details in the database.*
DELETE	*deleteStuden*	*Delete the student from the database.*

Untuk menjalankan operasi updateStudent, klien HTTP perlu mengirimkan permintaan menggunakan metode HTTP PUT ke jalur /studentdb/v1/students/12345. Setiap operasi pada REST API dapat disertai dengan sejumlah parameter yang digunakan oleh klien untuk menyampaikan argumen yang diperlukan dalam proses eksekusi. Seluruh parameter harus dijelaskan secara eksplisit dalam definisi REST API, dan masing-masing memiliki nama serta tipe data yang unik agar tidak menimbulkan ambiguitas dalam pemrosesan permintaan (Surwase, 2016)..

2.5. *Flask*

Flask merupakan kerangka kerja web yang dibangun dengan bahasa pemrograman *Python* yang ringan dan fleksibel. *Flask* biasa disebut sebagai *microframework* karena ukurannya yang kecil, tetapi tetap bisa diperluas dengan sesuai kebutuhan. *Flask* tidak memiliki fitur kompleks bawaan seperti *Object-Relational Manager* (ORM), tetapi mendukungnya melalui ekstensi tambahan (Jonsson, dkk., 2022). Dalam menangani banyak hal *Flask* secara default menggunakan *multithreading*, yang berarti dapat menangani beberapa permintaan secara bersamaan. Dengan pendekatan ini, Flask mampu membagi tugas berat ke beberapa thread untuk meningkatkan efisiensi dan performa aplikasi.

Python, seperti *JavaScript*, merupakan bahasa pemrograman yang diinterpretasikan. Namun, *Python* memiliki proses tambahan dalam mengeksekusi kodenya. Sebelum dijalankan, kode *Python* terlebih dahulu dikompilasi menjadi *bytecode*, yang kemudian dijalankan oleh *virtual machine Python*. Jika dibandingkan dengan *JavaScript modern*, cara kerja keduanya kurang lebih sama. Perbedaannya, *Node.js* menggunakan teknik *Just-In-Time* (JIT) *compilation* untuk mengoptimalkan *bytecode* saat dijalankan, sedangkan *Python* langsung menginterpretasikan *bytecode* melalui mesin virtualnya.

2.6. *ReactJS*

ReactJS adalah pustaka JavaScript yang digunakan untuk membangun antarmuka pengguna, khususnya dalam pengembangan aplikasi web berskala besar. ReactJS memungkinkan elemen-elemen dalam aplikasi untuk diperbarui secara dinamis tanpa harus memuat ulang seluruh halaman. Dalam arsitektur *Model View Controller* (MVC), ReactJS berfungsi sebagai bagian view yang menangani tampilan. Dengan memanfaatkan pendekatan Virtual DOM, ReactJS menyederhanakan manipulasi elemen-elemen pada halaman, sehingga pengembang dapat bekerja lebih efisien. Proses rendering dapat dilakukan di sisi server menggunakan Node.js, sementara untuk aplikasi mobile native, pengembang dapat menggunakan React Native. Alur data satu arah yang diterapkan membantu menjaga struktur kode tetap sederhana dan mudah dikendalikan dibandingkan dengan sistem pengikatan data dua arah (Dinku, 2022).

Konsep utama yang mendasari React.js adalah penggunaan Virtual DOM. React.js memanfaatkan pendekatan Virtual DOM secara efisien dalam pengelolaan antarmuka. Virtual DOM dapat di render baik di sisi klien maupun di sisi server, dan mampu saling berkomunikasi. Dengan memanfaatkan perubahan pada state, Virtual DOM membentuk subtree dari node yang relevan, lalu menerapkan manipulasi DOM seminimal mungkin agar komponen tetap diperbarui secara optimal tanpa perlu pembaruan secara eksplisit (Dinku, 2022).

2.7. *Tailwind CSS*

Tailwind CSS merupakan salah satu *framework* pengembangan web modern yang mengutamakan pendekatan berbasis utilitas (Tidu., 2023). *framework* ini memungkinkan pengembang untuk melakukan *customize* secara menyeluruh tanpa bergantung pada gaya

atau komponen bawaan yang sudah ditentukan. Setiap elemen desain dibangun melalui kombinasi kelas-kelas utilitas sederhana yang langsung diterapkan pada markup HTML. Berbeda dengan framework lain yang menyediakan komponen siap pakai, Tailwind memberikan fleksibilitas lebih dengan menyediakan alat minimalis berbasis CSS. Pengembang dapat mengontrol tampilan secara detail tanpa harus keluar dari struktur HTML. Dalam beberapa tahun terakhir, popularitas Tailwind CSS meningkat pesat dan banyak menarik perhatian dari kalangan desainer serta developer web. Gaya visual ditentukan melalui penggunaan kelas-kelas yang dirancang untuk mewakili properti CSS secara langsung (Al Salmi, 2023).

Tailwind CSS dirancang agar ramah bagi pengembang pemula yang ingin mempelajari dan menguasai cara penulisan gaya secara efisien. Berbeda dengan framework lain yang sering mengharuskan penggunaan kelas semantik yang kompleks, Tailwind menawarkan pendekatan melalui konvensi khusus yang mempermudah kolaborasi tim selama proses pengembangan. Kelas-kelas dalam Tailwind dituliskan langsung pada markup HTML, sehingga memisahkan antara kelas utilitas dan kelas semantik yang telah ada. Pendekatan ini mendorong konsistensi dalam desain dan meminimalkan konflik gaya. Efek gaya hanya diterapkan langsung pada elemen HTML yang bersangkutan, bukan melalui file CSS eksternal, sehingga perubahan tidak memengaruhi berkas lainnya. Proses verifikasi pun cukup dilakukan pada file HTML yang sedang dikerjakan, tanpa perlu menelusuri seluruh berkas CSS. Tailwind menawarkan pendekatan tingkat rendah dalam pengembangan gaya antarmuka, memungkinkan pengembang membangun tampilan secara fleksibel dan cepat (Al Salmi, 2023).

2.8. *Redux*

Redux merupakan *library* yang digunakan mengelola state dalam pengembangan aplikasi JavaScript agar perilakunya tetap konsisten. Library ini dapat diintegrasikan tidak hanya dengan JavaScript murni, tetapi juga dengan berbagai framework atau library antarmuka pengguna seperti React, Angular, maupun Vue.

Seiring bertambahnya fitur dalam sebuah aplikasi, jumlah data atau *state* yang dikelola pun ikut meningkat. Ketika data saling terhubung antar bagian aplikasi, pengelolaannya menjadi semakin kompleks. Dalam konteks penggunaan Tailwind CSS, pendekatan yang diterapkan memisahkan kelas utilitas dari kelas semantik karena ditulis langsung di dalam HTML. Cara ini membantu menjaga konsistensi tampilan desain.

Metode tersebut dinilai efektif karena efek styling hanya diterapkan langsung di markup HTML, bukan melalui file CSS yang bisa mempengaruhi komponen lain. Hal ini mempermudah pengembang dalam melakukan pengecekan, karena cukup memeriksa file HTML tanpa harus menelusuri seluruh file CSS. Tailwind sendiri merupakan kerangka kerja CSS tingkat rendah yang dirancang untuk mempercepat pembuatan antarmuka pengguna secara kustom (Pandey, 2019).

Redux berfungsi sebagai solusi untuk mengelola state dalam aplikasi. Berdasarkan dokumentasi resminya, Redux merupakan state container yang dapat diprediksi untuk aplikasi JavaScript. *Redux* juga menyimpan seluruh data aplikasi dalam satu tempat terpusat, yang terpisah dari antarmuka pengguna (UI). Untuk memastikan konsistensi dan kemudahan dalam pengelolaan data, Redux menerapkan aturan khusus yang harus diikuti ketika melakukan perubahan state. Pendekatan ini memungkinkan pengembang untuk lebih mudah melacak dan menangani permasalahan yang muncul akibat perubahan *state* dalam aplikasi (Pandey, 2019).

2.9. Axios

Axios merupakan pustaka (library) JavaScript yang berfungsi untuk melakukan permintaan HTTP dari sisi klien dalam pengembangan aplikasi web (Tidu., 2023). Penggunaan Axios memungkinkan aplikasi berkomunikasi secara efektif dengan server, baik untuk mengambil maupun mengirim data dalam berbagai format. Library ini banyak digunakan karena kemampuannya dalam menyederhanakan proses komunikasi data antara frontend dan backend. Library ini sering digunakan untuk berinteraksi dengan API (Application Programming Interface), termasuk dalam proses pengambilan (fetching) data dari server ataupun pengiriman data dari klien. Axios menawarkan antarmuka yang intuitif dan mudah digunakan untuk mengatur permintaan HTTP serta menangani respons dengan efisien (Rahmadhani dkk., 2024). Fungsionalitas ini sangat membantu developer dalam membangun aplikasi yang responsif dan terhubung secara dinamis dengan berbagai layanan eksternal.

Salah satu fitur unggulan dari Axios adalah dukungan terhadap konsep promise, yang mempermudah pengelolaan permintaan asynchronous tanpa perlu menggunakan callback secara berlebihan (Rahmadhani dkk., 2024). Dengan menggunakan promise, developer dapat menangani hasil request secara lebih terstruktur melalui pendekatan then() dan catch(). Selain itu, dibandingkan dengan metode native seperti fetch(), Axios

menawarkan keunggulan tambahan seperti konversi otomatis data respons ke dalam format JSON, kemampuan menetapkan batas waktu (timeout) untuk permintaan, serta fitur interceptor yang memungkinkan pengolahan request atau response sebelum dikirimkan atau diterima. Axios juga mendukung berbagai jenis metode HTTP seperti GET, POST, PUT, dan DELETE, serta mempermudah dalam pengelolaan header, termasuk dalam menyisipkan token autentikasi saat berkomunikasi dengan server secara aman dan efisien.

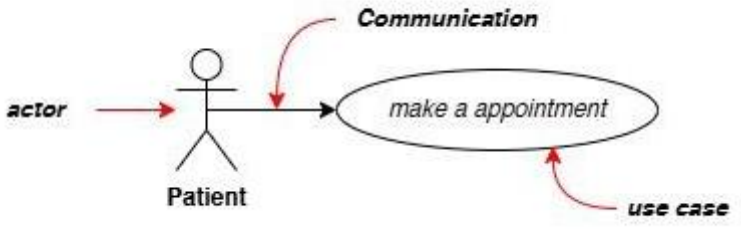
2.10. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang berfungsi untuk menggambarkan kebutuhan sistem, merancang desain, serta mendetailkan implementasi. Pengembangan UML dipelopori oleh Grady Booch, Jim Rumbaugh, dan Ivars Jacobson tiga tokoh utama di Rational Software Corp pada pertengahan tahun 1990-an. UML mengintegrasikan berbagai konsep dari metode yang berbeda dan secara khusus dirancang untuk pengembangan sistem berbasis objek (Siau & Cao, 2001).

UML berkembang pesat dan menjadi dominan dalam industri perangkat lunak dalam industri perangkat lunak. Bahasa pemodelan ini tidak hanya digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan komponen perangkat lunak, tetapi juga diakui oleh Objek Management Group (OMG) sebagai standar untuk analisis dan desain berbasis objek Selain itu, UML telah diusulkan sebagai standar internasional oleh International Standards Organization (ISO), dengan harapan dapat diterima secara luas dalam industri teknologi.

2.10.1. Use Case

Use case diagram adalah behaviour diagram yang termasuk dalam UML. menggambarkan kebutuhan fungsional dari perangkat lunak. Use Case diagram dapat digunakan untuk memahami bagaimana sistem seharusnya bekerja (Fauzan dkk., 2021). Untuk memahami bagaimana suatu sistem bekerja dan bagaimana pengguna berinteraksi dengan fitur-fitur yang tersedia, digunakan use case diagram. Diagram ini memberikan gambaran visual mengenai hubungan antara aktor dan fungsionalitas dalam sistem, sehingga mempermudah dalam menganalisis serta merancang kebutuhan perangkat lunak.

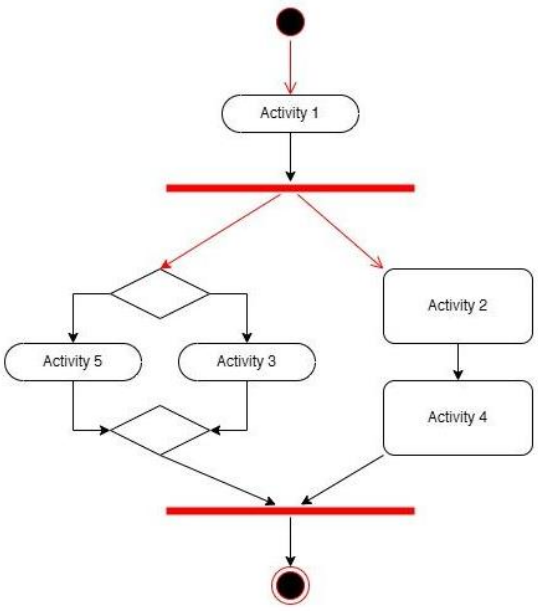


Gambar 2.3 Use Case (Hasanah, 2020).

Pada gambar 2.3 Use Case dijelaskan interaksi antara seorang aktor dan use case dalam sebuah sistem. Aktor, digambarkan sebagai figur stick figure, berinteraksi dengan use case yang berbentuk oval. Garis panah yang menghubungkan aktor dengan use case menunjukan bahwa aktor terlibat dalam proses yang dijelaskan oleh use case. Interaksi ini menggambarkan bagaimana aktor berkomunikasi dengan sistem untuk mencapai tujuan tertentu, di mana setiap use case mempresentasikan fitur atau layanan yang dapat digunakan oleh aktor. Dengan demikian, diagram ini menyoroti hubungan antara pengguna sistem dan berbagai fungsionalitas yang disediakan oleh sistem.

2.10.2. Activity Diagram

Activity Diagram adalah menggambarkan aliran kerja atau proses dalam sebuah sistem, bisnis, atau menu pada perangkat lunak. Activity Diagram menggambarkan aktivitas sistem bukan apa yang dilakukan oleh aktor (Musthofa & Adiguna, 2022).



Gambar 2.4 Activity Diagram (Hasanah, 2020).

Pada Gambar 2.4 Activity Diagram dijelaskan contoh alur proses dari activity diagram, yang menggambarkan urutan aktivitas dalam suatu sistem. Diagram ini menunjukan bagaimana aktivitas dimulai, bercabang dan berakhir berdasarkan keputusan yang dibuat dalam proses.

Alur dimulai dengan Activity 1 sebagai langkah awal, kemudian terbagi menjadi jalur utama. Jalur pertama mengarah ke Activity 2, sedangkan jalur kedua bercabang lagi menjadi Activity 3 dan Activity 5. Setelah kedua aktivitas selesai, terdapat sebuah keputusan yang menentukan arah proses selanjutnya sebelum menuju ke tahap akhir. Sementara itu, jalur dari Activity 2 dilanjutkan dengan Activity 4.

Keseluruhan proses akhirnya bertemu pada titik akhir, yang menandakan bahwa semua jalur telah dilalui dan dan seluruh aktivitas telah terselesaikan. Diagram ini memberikan visualisasi yang jelas tentang aliran proses dalam sistem serta pengambilan keputusan yang terjadi di dalamnya.

2.10.3. *Entity Relationship* Diagram

Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang paling umum digunakan dalam perancangan basis data. Entity Relationship Diagram digunakan untuk pemodelan basis data relational. Entity Relationship Diagram merupakan suatu diagram dalam bentuk gambar atau simbol yang mengidentifikasi tipe dari entitas di dalam suatu sistem yang diuraikan dalam daya dengan atributnya, dan menjelaskan hubungan atau relasi antar entitas tersebut (Hasanah, 2020).

2.11. *Referensi Terdahulu*

Beberapa hasil penelitian sebelumnya yang relevan dengan topik ini dirangkum untuk menjadi dasar dan bahan pertimbangan dalam pengembangan penelitian yang sedang dilakukan.

Tabel 2.3 Peneliti terdahulu

No	Nama dan Tahun Publikasi	Hasil
1	(ARIFIANTO, 2022)	**Judul :** IMPLEMENTASI SISTEM INFORMASI DETEKSI OTOMATIS SPESIES TUMBUHAN DI KEBUN RAYA BALIKPAPAN MENGGUNAKAN

No	Nama dan Tahun Publikasi	Hasil
		MODEL RESNET-50 DENGAN ITERATIVE WATERFALL **Metode** : Penelitian ini menggunakan pendekatan *deep learning* berbasis Convolutional Neural Network (CNN) untuk mendeteksi objek tanpa perlu melakukan ekstraksi ciri manual. Model yang digunakan dilatih dengan dataset gambar yang telah diberi intonasi berupa bounding box dan label objek. Model *deep learning* ini kemudian diintegrasikan ke dalam aplikasi web dengan menyediakan layanan web *API* menggunakan TensorFlow Serving, yang memungkinkan model dijalankan secara terpisah dengan efisiensi penggunaan memori dan waktu inferensi yang lebih optimal. **Hasil** : Hasil penelitian menunjukkan bahwa pendeteksian jerawat menggunakan *deep learning* memberikan performa yang baik dengan nilai *mean Average Precision* (mAP) sebesar 42,2% dan *Average Recall* (AR) sebesar 32%. Jika dibandingkan dengan anotasi manual oleh dokter kulit, model memiliki *recall* sebesar 77,2% dan *precision* 70,3%, yang menunjukkan tingkat akurasi yang cukup tinggi dalam mengidentifikasi jerawat. Aplikasi berbasis web yang dikembangkan memungkinkan pengguna mengunggah foto wajah, dan sistem akan secara otomatis mendeteksi serta menghitung jumlah jerawat yang terdeteksi dalam citra digital.
2	(Pratiwi & Nudin, 2023)	**Judul** : Pengembangan Sistem Deteksi Dini Kebakaran Menggunakan Pendekatan Transfer Learning dengan

No	Nama dan Tahun Publikasi	Hasil
		Model ResNet-50 Berbasis Website **Metode** : Penelitian ini menggunakan transfer learning dengan model ResNet-50V2 untuk mendeteksi kebakaran secara dini melalui sistem berbasis website. Pengembangan model dilakukan melalui data collection, data preparation, modeling, dan evaluasi. Model dioptimalkan menggunakan fully connected layer, dan sistem dikembangkan menggunakan framework Flask untuk mengintegrasikan model deep learning dengan antarmuka berbasis web. Pengujian dilakukan terhadap data training, validasi, dan testing untuk menilai performansi sistem. **Hasil** : Hasil penelitian menunjukan bahwa model ResNet-50V2 memiliki akurasi lebih tinggi dibandingkan ResNet-50, dengan akurasi training 98%, validasi 100%, dan testing 79%, serta nilai loss minimal. Model dapat mendeteksi kebakaran baik dari gambar yang diunggah maupun yang diambil langsung melalui kamera. Sistem berbasis Flask berhasil mengintegrasikan model deep learning dengan antarmuka pengguna yang interaktif, memproses data secara real-time, dan menampilkan hasil klasifikasi dengan akurat. Dengan demikian, sistem ini berpotensi diterapkan dalam deteksi kebakaran dini secara otomatis.
3	(MEKACAHYANI, 2024)	**Judul** : Klasifikasi Penyakit Kulit Deramatitis Atopik Dan Psoriasis menggunakana algoritma CNN dengan model arsitektur RestNet50

No	Nama dan Tahun Publikasi	Hasil
		Metode : Penelitian ini menggunakan algoritma Convolutional Neural Network (CNN) dengan arsitektur RestNet50 untuk mengklasifikasikan penyakit kulit dermatitis atopik dan psoriasis. Model diuji dengan tujuh skenario, termasuk pemotongan gambar manual, otomatis, dan perbandingan dengan MobileNetV2. Sistem dikembangkan dalam aplikasi web Streamline, memungkinkan pengguna mengunggah gambar untuk diprediksi secara otomatis. **Hasil** : Penelitian ini menunjukan bahwa ResNet-50 mencapai akurasi tinggi dalam mengklasifikasi dermatitis atopik dan psoriasis, dengan skenario terbaik menghasilkan akurasi pelatihan 92,75% dan pengujian 88%. Analisis confusion matrix menunjukkan akurasi validasi 88%, dengan precision, recall, dan f1-score untuk dermatitis atopik (95%, 80%, 87%) dan psoriasis (83%, 96%, 89%). Model ini terbukti efektif dalam identifikasi penyakit kulit dan telah diterapkan dalam aplikasi web Streamlit, yang memungkinkan pengguna mengunggah gambar dan memperoleh hasil prediksi serta informasi terkait penyakit.
4	(Purboningrum et al., 2024)	**Judul** : Sistem Deteksi Dini Autism Spectrum Disorder (ASD) Berbasis Face Recognition Menggunakan Metode Transfer Learning Resnet50 **Metode** : Penelitian ini menggunakan model ResNet-50 dengan optimasi epoch dan callbacks early stopping untuk meningkatkan akurasi model. Evaluasi kinerja dilakukan menggunakan confusion matrix, serta analisis kecepatan komputasi dalam mengklasifikasi

No	Nama dan Tahun Publikasi	Hasil
		citra wajah anak autistik dan non autistik. **Hasil** : Penelitian ini menunjukan bahwa epoch ke 51 adalah tidak optimal, dimana model mencapai akurasi 91%, precision 92%, recall 90%, dan f1-score 90%. Sistem mampu mengklasifikasikan citra dengan rata-rata akurasi 80%, dengan akurasi 100% pada kelas autistik dan 60% pada kelas non-autistik. Selain itu, waktu komputasi rata-rata 2,827 detik menunjukkan efisiensi model dalam proses inferensi. Penelitian ini membuktikan bahwa ResNet-50 dapat diterapkan secara efektif dalam sistem pendeteksian autisme berbasis citra wajah melalui aplikasi berbasis AI.
5	(Qisthiano, 2024)	**Judul :** HEWAN BERBASIS WEBSITE DENGAN PENDEKATAN MACHINE **Metode:** Penelitian ini mengembangkan teknologi AI untuk dengan pendekatan *machine learning* dengan metode *Extreme Programming* (XP) dalam sistem identifikasi hewan berbasis website. Sistem yang dikembangkan mampu menganalisis gambar dengan menggunakan algoritma *deep learning* dan mencocokannya dengan database spesies, menghasilkan identifikasi yang tepat dan relevan. **Hasil :** Hasil penelitian ini menunjukkan bahwa sistem yang dikembangkan berkontribusi secara signifikan dalam bidang identifikasi biologi. Dengan pembaruan data dan peningkatan model machine learning yang digunakan, sistem ini dapat terus berkembang untuk memberikan hasil identifikasi yang lebih akurat.