

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Komoditas hortikultura merupakan sektor penting dalam perekonomian masyarakat Indonesia. Salah satu komoditas hortikultura yang banyak dibudidayakan adalah salak. Salak termasuk tanaman yang memiliki nilai ekonomis cukup tinggi. Salak merupakan tumbuhan palem yang tergolong ke dalam marga *Salacca*, anak suku *Calamoideae*, suku *Arecaceae*. Tahun 1791 *Gaertner* mendeskripsikan *Calamus zalacca* berdasarkan organ buah dari koleksi *Thunberg* yang ternyata adalah *S. edulis Reinw*. Nama *Salacca Reinw* telah diakui sebagai publikasi yang sah untuk marga salak. Namun, dalam beberapa literatur lain, penulisan nama marga tersebut masih sering ditemukan sebagai *Zalacca*. Selanjutnya, pada tahun 1825, marga ini dideskripsikan oleh Reinwardt dengan mengacu pada spesies *Salacca edulis* [1].

Salak merupakan buah asli Indonesia yang mampu tumbuh dengan baik di dataran rendah dan memiliki potensi besar dalam memenuhi kebutuhan pasar, baik di dalam negeri maupun luar negeri. Buah tropis ini dikenal memiliki rasa manis dan tekstur renyah, serta dapat dikonsumsi secara langsung maupun diolah menjadi berbagai produk lainnya. Seiring dengan meningkatnya permintaan pasar terhadap buah salak, kebutuhan akan bibit unggul juga semakin tinggi [1].

Namun, dalam praktiknya, masyarakat khususnya petani dan calon pembeli bibit salak masih menghadapi kesulitan dalam membedakan jenis bibit salak pada

tahap awal pertumbuhan. Hal ini disebabkan karena bibit dari berbagai jenis salak memiliki kemiripan secara visual, baik dari segi bentuk, warna, maupun struktur, sehingga sulit dikenali terutama oleh orang yang belum berpengalaman. Proses pemilihan bibit salak menjadi kurang akurat karena tidak cukup pengetahuan dan pengalaman untuk mengetahui ciri-ciri masing-masing jenis bibit salak. Ini sering menimbulkan kebingungan, dan memungkinkan terjadinya tindak kecurangan dalam penjualan bibit oleh oknum tertentu.

Oleh karena itu, untuk mengatasi kendala tersebut, penelitian ini mengusulkan pengembangan sebuah sistem untuk penentuan jenis bibit salak berbasis kecerdasan buatan dengan menerapkan metode *Convolutional Neural Network* (CNN). Berdasarkan penelitian R. Soekarta *Convolutional Neural Network* (CNN) memiliki keunggulan dalam pengolahan dan analisis data visual gambar. Dengan fitur yang bisa mengekstraksi secara otomatis dapat mengidentifikasi fitur-fitur penting dari data visual tanpa memerlukan fitur manual yang dirancang manusia [2].

Sedangkan menurut S. A. Adzkie and T. Arifin *Convolutional Neural Network* (CNN) merupakan bidang khusus dalam *deep learning* yang secara dominan diterapkan dalam analisis data visual. CNN terdiri dari sejumlah lapisan terlatih, termasuk lapisan *input*, *convolution*, *pooling*, dan *fully connected* [3]. Arsitektur CNN yang dikombinasikan dengan *transfer learning* telah menunjukkan efektivitas yang lebih tinggi dalam klasifikasi citra, melampaui model *machine learning* konvensional CNN merupakan salah satu cara yang paling efisien untuk klasifikasi dalam *computer vision*. metode berbasis CNN

mencapai keberhasilan yang signifikan dalam pengenalan citra. CNN mampu mengekstrak fitur-fitur komponen serta memperoleh informasi yang lebih kaya dari citra [4].

Berdasarkan uraian pada paragraf sebelumnya, maka penelitian ini dirumuskan dalam bentuk tugas akhir dengan judul “Rancang Bangun Aplikasi Deteksi Bibit Tanaman Salak Menggunakan Metode *Convolutional Neural Network* (CNN) Berbasis Web” yang bertujuan untuk mengembangkan suatu sistem berbasis web yang mampu mengidentifikasi jenis bibit tanaman salak secara otomatis guna membantu masyarakat dalam membedakan varietas bibit secara lebih akurat dan efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, maka rumusan masalah dalam penelitian ini adalah bagaimana menerapkan metode *Convolutional Neural Network* (CNN) dalam aplikasi deteksi jenis bibit tanaman salak berbasis Web.

1.2 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah menerapkan metode *Convolutional Neural Network* (CNN) dalam aplikasi deteksi jenis bibit tanaman salak berbasis Web.

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini, yaitu :

1. Metode yang digunakan untuk deteksi bibit tanaman salak adalah *Convolutional Neural Network*

2. *Dataset* yang digunakan dalam penelitian ini yaitu salak pondoh, salak pusaka, salak lokal, salak gading, dan kelas negatif Bukan_Salak, dengan total keseluruhan data sebanyak 5.000 citra *Dataset*.
3. Backbone yang digunakan dalam model *Convolutional Neural Network* (CNN) pada penelitian ini adalah ResNet (Residual *Network*).

1.3 Manfaat Penelitian

1. Bagi peneliti : Penelitian ini memberikan manfaat dalam menambah wawasan dan pemahaman peneliti terkait penerapan metode *Convolutional Neural Network (CNN)* dalam bidang pengolahan citra digital, khususnya untuk deteksi jenis bibit tanaman salak.
2. Bagi pengguna : Penelitian ini diharapkan dapat memberikan kemudahan bagi masyarakat, khususnya petani atau pembudidaya salak, dalam mengidentifikasi jenis bibit tanaman salak secara lebih cepat dan akurat.

1.4 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri dari pokok-pokok permasalahan yang dibahas dan diuraikan menjadi beberapa bagian :

BAB 1 PENDAHULUAN

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

BAB 2 LANDASAN TEORI

Bab ini berisi teori-teori yang digunakan pada penelitian ini. Teori-teori yang berhubungan dengan pengembangan sistem yang akan dibangun.

BAB 3 METODOLOGI PENELITIAN

Bab ini berisi tahapan-tahapan dalam pengumpulan data, perancangan sistem, perumusan masalah dan analisa.

BAB 4 ANALISA DAN PERANCANGAN

Bab ini berisi analisa dan pengembangan sistem yang akan dibangun atau dikembangkan.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari analisa dan perancangan dan pengujian pada sistem yang berhasil dibangun.

BAB 6 PENUTUP

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

BAB 2

LANDASAN TEORI

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan Buatan (AI) adalah merupakan kemampuan mesin atau sistem komputer untuk meniru atau melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia, seperti penalaran logis, pembelajaran, dan pemecahan masalah [5]. Kecerdasan buatan mengacu pada pengembangan dan implementasi sistem komputer yang memiliki kemampuan untuk melakukan tugas-tugas yang biasanya membutuhkan kecerdasan manusia. Tugas-tugas tersebut termasuk belajar dari pengalaman, pemikiran dari permasalahan yang kompleks, memahami bahasa alami, dan beradaptasi dengan situasi baru [6].

Salah satu arah perkembangan AI di masa kini adalah kemampuan untuk menjadi lebih pintar dan memahami konteks. AI saat ini sudah mampu melakukan tugas-tugas seperti pengenalan suara dan gambar dimasa kini AI diharapkan mampu meningkatkan kemampuan untuk memahami dan membuat keputusan berdasarkan situasi dan konteks [7] . Salah satu teknik yang banyak digunakan dalam pengembangan kecerdasan buatan saat ini adalah *machine learning* dan *deep learning*. Pada penelitian ini, kecerdasan buatan digunakan untuk membantu proses mengidentifikasi jenis bibit buah salak berdasarkan citra gambar dengan memanfaatkan metode *Convolutional Neural Network* (CNN).

2.2 *Machine Learning*

Machine learning sendiri merupakan bagian dari AI yang berisi kumpulan algoritma yang membuat mesin(sistem) memiliki pengetahuan dari data train yang diberikan. Pengetahuan terhadap permasalahan yang ada tidak serta merta didapat tanpa adanya proses belajar. Mesin terlebih dahulu di latih(train) menggunakan data yang sebelumnya telah melalui serangkaian proses hingga layak menjadi data train [8]. Selain itu *Machine learning* atau Pembelajaran mesin, cabang dari AI-kecerdasan buatan, adalah disiplin ilmu yang mencakup pe-rancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengem-bangkan perilaku yang didasarkan pada data empiris, seperti dari sensor data basis data [9].

Machine learning banyak digunakan dalam berbagai aplikasi seperti sistem deteksi *spam*, pengenalan suara, sistem rekomendasi, serta pengolahan citra digital. Dalam pengolahan citra, *machine learning* dapat digunakan untuk mengenali objek yang terdapat dalam gambar. Salah satu perkembangan dari *machine learning* adalah *deep learning*. *Deep learning* adalah aspek jaringan saraf tiruan yang bertujuan untuk meniru teknik *machine learning* yang digunakan manusia untuk memperoleh jenis pengetahuan tertentu. *Deep learning* dapat didefinisikan sebagai mempelajari berbagai tingkat representasi dan abstraksi yang membantu kita memahami data seperti gambar, suara, dan teks [10].

2.3 *Deep Learning*

Deep learning adalah area pembelajaran mesin yang menggunakan jaringan saraf tiruan untuk menyelesaikan masalah dengan kumpulan data besar. Metode *deep learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*.

Dengan menambahkan lapisan, memungkinkan model pembelajaran untuk merepresentasikan gambar berlabel dengan lebih baik [2].

Deep learning digunakan untuk mempelajari pola-pola kompleks dalam data seperti gambar, suara, teks dan lain-lain dengan memanfaatkan komputasi yang sangat paralel dan berdaya tinggi [11]. *deep learning* juga diintegrasikan dalam berbagai produk berteknologi canggih seperti mobil otonom, serta hadir dalam produk dan layanan yang digunakan sehari-hari seperti asisten digital [12].

2.4 Citra Digital (*Digital Image*)

Citra digital merujuk pada proses manipulasi gambar dua dimensi dengan menggunakan komputer, Citra digital harus diwakili dalam bentuk angka numerik dengan nilai intensitas cahaya yang diskrit, sehingga komputer dapat memprosesnya dengan baik. Citra Digital bertujuan memberikan pemahaman dan penguasaan pengetahuan serta keterampilan tentang konsep dasar tipografi, citra *bitmap*, dan citra *vector* [13].

Pengolahan citra digital mengacu pada suatu pengolahan citra menggunakan komputer digital dimana citra sebagai masukan dan keluarannya yang prosesnya meliputi ekstraksi atribut dari citra hingga melakukan rekognisi objek secara individual [14]. Citra digital memberikan keuntungan dibandingkan dengan tipe informasi yang disajikan dalam bentuk visual. Format digital memberikan kemudahan untuk transfer, penyimpanan dan pe-manggilan kembali data citra [15].

2.5 *Convolutional Neural Network (CNN)*

Convolutional Network atau yang dikenal dengan *convolutional neural Network (CNN)* adalah tipe khusus dari *neural Network* untuk memproses data yang mempunyai topologi jala atau *grid-like topology*. Pemberian nama *convolutional neural Network* mengindikasikan bahwa jaringan tersebut menggunakan operasi matematika yang disebut *konvolusi*. *Konvolusi* sendiri adalah sebuah operasi linear. Jadi *Convolutional Network* adalah *neural Network* yang menggunakan *konvolusi* minimal pada salah satu lapisannya [16].

Convolutional Neural Network (CNN) merupakan salah satu bagian dari *Deep Neural Network* yang diimplementasikan untuk image recognition. Metode ini merupakan pengembangan dari *Multilayer Perception (MLP)*. Namun *MLP* menerima input data satu dimensi, sementara *CNN* data dua dimensi dan salah satunya adalah citra [17].

Metode *CNN* terdiri dari beberapa lapisan utama:

2.5.1 *Convolutional layer*

Convolutional layer merupakan lapisan inti dari arsitektur *CNN* yang melakukan proses komputasi dengan menggeser matriks filter (*kernel*) ke seluruh permukaan citra untuk menghasilkan *feature map* yang berisi informasi penting dari gambar tersebut. *Convolution Layer* bisa manipulasi gambar yang dilakukan menggunakan maks eksternal atau subwindow, dengan menggunakan operasi *konvolusi* untuk mengubah ukuran gambar dan membuat fungsi ekstraksi penting dari gambar, seperti: Buat gambar baru lainnya dengan proses coding [18].

2.5.2 *Pooling Layer*

Pooling layer berfungsi untuk mereduksi dimensi data citra agar proses komputasi menjadi lebih cepat, sekaligus membantu menjaga fitur-fitur yang paling signifikan agar tetap terdeteksi oleh model [19].

2.5.3 *Fully Connected Layer*

Fully connected layer bertujuan untuk memproses data dari lapisan-lapisan sebelumnya sehingga dapat diklasifikasikan ke dalam kategori yang sesuai melalui perhitungan bobot pada setiap neuronnya.

2.5.4 *Softmax*

Softmax adalah fungsi aktivasi yang mengubah vektor *output* dari *fully connected layer* menjadi nilai probabilitas dalam rentang 0 hingga 1, di mana total seluruh nilai kelas berjumlah 1 guna menentukan hasil klasifikasi akhir [20].

Selain itu metode *Convolutional Neural Network* (CNN) lebih baik daripada metode *Multi Layer Perceptron* (MLP) dikarenakan metode *Convolutional Neural Network* (CNN) memiliki kedalaman jaringan yang tinggi dan sering diaplikasikan pada data pengenalan citra sehingga mampu menghasilkan tingkat akurasi tinggi dan hasil yang baik [21].

2.6 *Flask Framework*

Flask adalah *microframework* yang jauh lebih ringan dan cepat dibandingkan dengan *framework* lain karena *Flask* dibuat dengan menyederhanakan inti *framework*-nya seminimal mungkin. Dengan tagline “*web development, one drop at a time*”. *Flask* yang pertama kali dikembangkan oleh

Armin Ronacher dapat membantu kita dalam membuat sebuah situs *website* dengan sangat cepat meskipun dengan *library* yang sederhana [22].

Flask menyediakan *library* dan kode yang bisa digunakan untuk membangun *web* tanpa perlu melakukan semuanya dari nol. Menggunakan *Flask* membuat *web* lebih terstruktur [23]. *Flask* menyusun kategorinya menjadi 2 bagian yaitu *Static File* yang berisi semua kode yang dibutuhkan untuk *website* seperti *CSS*, *JavaScript* dan *File* gambar, dan *Template File* yang berisikan semua halaman *HTML* [24].

2.7 *ResNet*

ResNet merupakan salah satu arsitektur dari CNN yang dirancang untuk dapat mengaktifkan ratusan atau bahkan ribuan lapisan *konvolusi*. Terdapat dua tipe ResNet yang digunakan sebagai backbone dalam algoritma *Convolutional Neural Network*), yaitu ResNet-50 dan ResNet-101. Perbedaan mendasar dari ResNet-50 dan ResNet-101 adalah pada jumlah lapisan serta parameternya. Pada ResNet-50 memiliki 50 lapisan dengan 23,52 juta parameter, sedangkan ResNet-101 memiliki 101 lapisan dengan 42,51 juta parameter [25].

Kelebihan model ResNet dibandingkan dengan model arsitektur CNN yang lain adalah kinerja dari model ini tidak menurun walaupun arsitektur nya semakin dalam. Selain itu, perhitungan komputasi yang dilakukan lebih ringan dan kemampuan untuk melatih jaringan yang lebih baik [26].

2.8 *Dataset*

Dataset adalah kumpulan data yang terorganisir dalam format yang terstruktur. *Dataset* merupakan sekumpulan data yang dikumpulkan dan disusun

secara sistematis untuk digunakan dalam proses analisis maupun pelatihan model dalam bidang *Machine learning* dan *Deep Learning*. Data yang terdapat dalam sebuah *Dataset* dapat berupa teks, angka, audio, maupun citra yang digunakan sebagai sumber pembelajaran bagi sistem komputer [27].

Dataset juga merupakan kumpulan data, yang mana biasanya satu *Dataset* merepresentasikan satu tabel *database*, atau bisa juga suatu matriks data yang mana tiap kolom mewakili variabel tertentu, tiap baris merepresentasikan banyaknya data [28]. Sama halnya yang dikatakan J. Nasional and S. Informasi bahwa *Data set* adalah kumpulan dari objek dan sifat atau karakteristik dari suatu objek itu sendiri (atribut). *Dataset* terdiri dari sekumpulan *File* atau dokumen. Pada statistik, *Dataset* biasanya berasal dari hasil pengamatan aktual yang diperoleh dari dengan mengambil sampel populasi. *Dataset* dapat menghasilkan algoritma yang bisa digunakan untuk pengujian pada perangkat lunak [29].

2.9 Bibit Salak

Bibit salak (*Salacca zalacca*) dapat diperoleh melalui dua metode perbanyakan utama, yaitu perbanyakan secara generatif dan vegetatif. Bibit generatif merupakan bibit yang berasal dari perkecambahan biji yang telah disemai, sedangkan bibit vegetatif diperoleh tanpa melalui biji, melainkan melalui pembiakan aseksual seperti teknik pencangkakan dari tunas anakan tanaman induk. Asal-usul bibit ini menentukan kerapatan dan teknik penanamannya di lapangan [30]. Berdasarkan perjanjian kinerja direktorat jenderal hortikultura. Tanaman salak (*Salacca zalacca*) merupakan tanaman tropis yang dapat ditemukan hampir di seluruh wilayah Indonesia [31].



Gambar 2. 1 Salak Pondoh

Salak Pusaka merupakan varietas buah salak unggulan yang berasal dari daerah Rokan Hulu. Buah ini dikenal sebagai salah satu komoditas khas daerah yang memiliki nilai ekonomi sekaligus nilai budaya karena telah dibudidayakan secara turun-temurun oleh masyarakat setempat [32].



Gambar 2. 2 Salak Pusaka

Salak lokal juga dikenal sebagai salak lokal adalah varietas buah salak tradisional yang tumbuh secara alami di berbagai daerah di Indonesia, sering kali dibedakan dari varietas budidaya komersial.



Gambar 2. 3 Salak lokal

Salak gading merupakan salah satu varietas salak lokal yang umumnya ditemukan di wilayah Riau, termasuk di daerah pedesaan seperti Rokan Hulu. Varietas ini dikenal sebagai salak yang tumbuh secara tradisional dan belum banyak mengalami pengembangan intensif seperti varietas unggulan lainnya [33].



Gambar 2. 4 Salak Gading

2.10 kelas negatif Bukan_Salak

Dalam pengembangan model *Convolutional Neural Network* (CNN), penyertaan kelas negatif atau data non-target (seperti latar belakang, tanah, atau objek asing di luar kategori utama) memegang peranan krusial untuk meningkatkan ketahanan (*robustness*) dan kemampuan generalisasi sistem pada lingkungan nyata. Pengelolaan variasi data latih yang mencakup objek di luar target utama sangat diperlukan untuk mengoptimalkan batas keputusan (*decision boundary*) model serta menekan tingkat kesalahan klasifikasi atau *false positive* [34]. Dengan mendefinisikan kelas negatif seperti *Bukan_Salak*, sistem dilatih tidak hanya untuk mengenali ciri spesifik objek target, tetapi juga dibekali kemampuan diskriminatif yang akurat guna mengabaikan elemen pengganggu di sekitar area pengamatan. Kelas negatif adalah kelas milik tulisan berwarna hitam atau yang dianggap benar. Agar lebih jelas, TP adalah tulisan yang dimerahkan dan ternyata valid [35].

2.11 Data Augmentation

Data Augmentation memperkaya data pelatihan untuk menghindari munculnya *overfitting*. Proses *data augmentation* terdiri dari *horizontal flip*, *shear range*, dan *zoom range*. *Shear range* dan *zoom range* memiliki nilai 0.2. *Horizontal flip* bekerja untuk memperbanyak data pelatihan dengan cara memutar gambar secara horizontal 90 derajat. *Shear range* menerapkan metode *shear transformation* [34].

Selain itu Augmentasi data merupakan serangkaian strategi untuk memperbesar dan menyempurnakan dan bentuk suatu gambar dengan tetap

mempertahankan labelnya. Augmentasi data digunakan untuk meningkatkan kemampuan pembelajaran model dan menggeneralisasi kinerjanya. Saat ini augmentasi data adalah cara paling efektif untuk mengatasi permasalahan ketersediaan data berkualitas dalam jumlah besar [36].

2.12 *Anotasi*

Anotasi atau pelabelan data merupakan tahapan krusial dalam persiapan *Dataset* untuk algoritma *supervised learning* seperti CNN. Menurut N. Pramesti, R. Rianto Anotasi berfungsi untuk memberikan nilai target atau identitas pada setiap citra dalam *Dataset*, sehingga model dapat mengenali karakteristik spesifik dari masing-masing kelas objek yang dilatih [37]. Proses *annotation* dilakukan untuk memberikan panduan bagi mesin agar sistem dapat melatih dirinya sendiri dalam mendeteksi posisi atau jenis objek tertentu di dalam citra digital [38]. Selain itu Pelabelan merupakan tahap penambahan keterangan atau variabel penentu pada objek agar dapat diidentifikasi oleh sistem dengan menyesuaikan inputan kelas yang telah ditetapkan dalam basis data [39].

2.13 *Aplikasi*

Aplikasi diambil dari bahasa Inggris “*application*” yang dapat diartikan sebagai penerapan atau penggunaan. Secara harfiah merupakan suatu subkelas aplikasi personal komputer yg memanfaatkan kemampuan personal komputer pribadi untuk melakukan suatu tugas sesuai keinginan pengguna. Pada kemajuan teknologi, aplikasi sangat mendukung aktivitas manusia karena dapat diunduh dengan mudah pada gawai [40].

Selain itu aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Aplikasi juga merupakan suatu komponen yang bermanfaat sebagai media yang digunakan untuk melakukan pengelolaan data-data maupun kegiatan lainnya [41].

2.14 *Google Collab*

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

Selain itu *Colab* memungkinkan siapa saja menulis dan mengeksekusi kode *python* arbitrer melalui *browser*, dan sangat cocok untuk *machine learning*, analisis data, serta pendidikan. Secara lebih teknis, *Colab* merupakan layanan *notebook Jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi termasuk *GPU*. *Resource Colab* tidak dijamin dan sifatnya terbatas, serta batas penggunaannya terkadang berfluktuasi yang dikatakan [43].

2.15 *Google Drive*

Google Drive adalah *storage* berbasis *cloud* yang dikembangkan Google untuk tempat *storage* yang dapat diintegrasikan ke layanan milik Google yang lain. 24 April 2012 merupakan hari peluncuran *Google Drive* ini dan pada saat tahun 2018 *Google Drive* menyentuh 1 miliar pengguna [2]. Ukuran ruang *storage* yang didapatkan di akun *Google Drive* yaitu sebesar 15GB tetapi sudah termasuk *Google Mail*, *Google Slides*, *Google Docs*, *Google Photos*, serta *Google Sheets* [1]. Layanan *Google Drive* dapat diakses dan digunakan hanya dengan satu akun Google yang terdaftar, baik melalui platform *web*, *Android* maupun *iOS* [44].

Manfaat penggunaan *Google Drive* antara lain, dapat mengirim jenis *File*; dapat tersinkronisasi pada *folder* komputer/laptop serta *smartphone*; dapat melakukan perubahan menggunakan Google aplikasi seperti pengolah kata, presentasi dan *form*; dapat memberikan hak akses; dan menyediakan ruang penyimpanan yang besar. Setelah melakukan pembuatan *email* di Google selesai, maka akan pindah ke tab baru pada halaman *Google Drive*. *Google Drive* juga difasilitasi untuk pemilihan tampilan *File* / berkas dengan ikon besar (*grid*) atau sebagai gambar (*list*). Modus *List* akan memperlihatkan sekilas siapa pemilik berkas dan kapan berkas tersebut terakhir diubah [45].

2.16 *Visual Studio Code*

Visual Studio Code adalah *source code editor* yang dikembangkan oleh Microsoft yang dapat digunakan pada berbagai platform (*Linux*, *Mac OS*, dan *Microsoft Windows*). *Visual Studio Code* merupakan *software* yang sangat ringan

namun memiliki *source code editor* yang kuat dan berjalan pada *desktop*. Aplikasi ini dilengkapi dengan dukungan bawaan (*built-in support*) untuk *JavaScript*, *TypeScript*, dan *Node.js*, serta memiliki beragam *extension* yang tersedia untuk bahasa pemrograman lain, termasuk *C++*, *C#*, *Python*, dan *PHP* [46].

Visual Studio Code ditujukan terutama bagi pengembang *JavaScript* yang menginginkan alat pengembangan yang lengkap untuk *server-side scripting* mereka, serta bagi mereka yang ingin mengembangkan aplikasi menggunakan kerangka kerja berbasis *.NET*. *Visual Studio Code* merupakan *code editor* yang ringan, stabil, dan *cross-platform*, sehingga dapat digunakan oleh siapa saja untuk membangun aplikasi berbasis *web* [46].

2.17 UML (Unified Modeling Language)

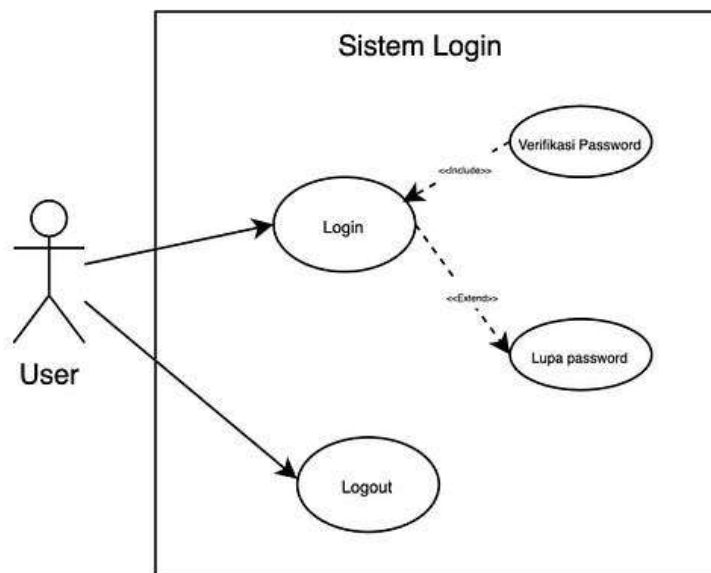
UML (Unified Modelling Language) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan merek dengan yang lain [47].

Terdapat beberapa *DiagramUML* yang sering digunakan dalam pengembangan sebuah sistem, yaitu [48] :

2.17.1 Use Case

Use Case adalah sebuah gambaran dari hubungan antara aktor dan sistem yang saling berkaitan. Dapat diartikan juga dengan sebuah teknik untuk digunakan

dalam pengembangan perangkat lunak agar mengetahui kebutuhan fungsional dari sistem tersebut [49]. *Use Case* diagram juga merupakan pemodelan untuk kelakuan sistem yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu [50].

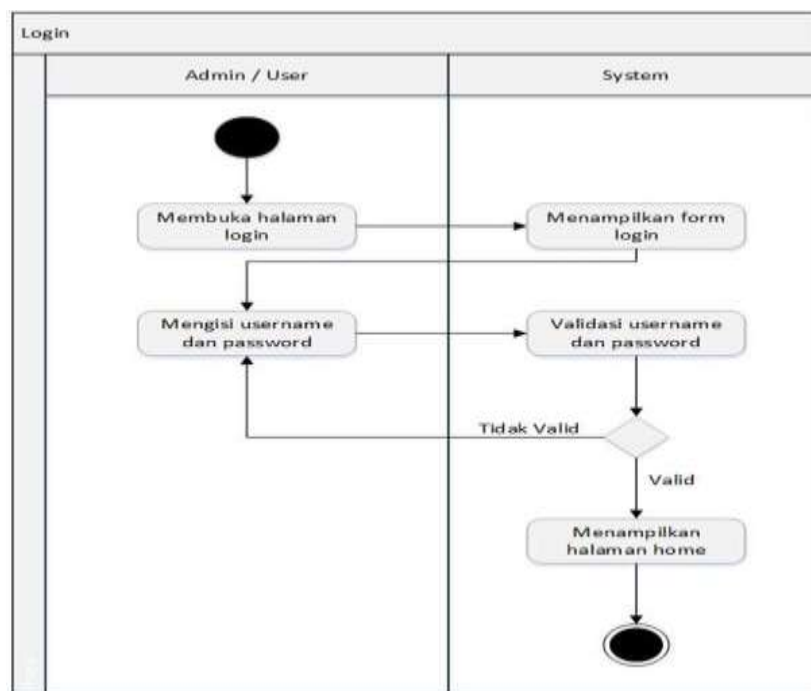


Gambar 2. 5 Use Case Diagram[51]

2.17.2 Activity Diagram

Diagram *Activity* menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity Diagram* juga dapat menggambarkan proses lebih dari satu aksi dalam waktu bersamaan [52]. *Activity*

diagram menggambarkan work flow atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem [50].

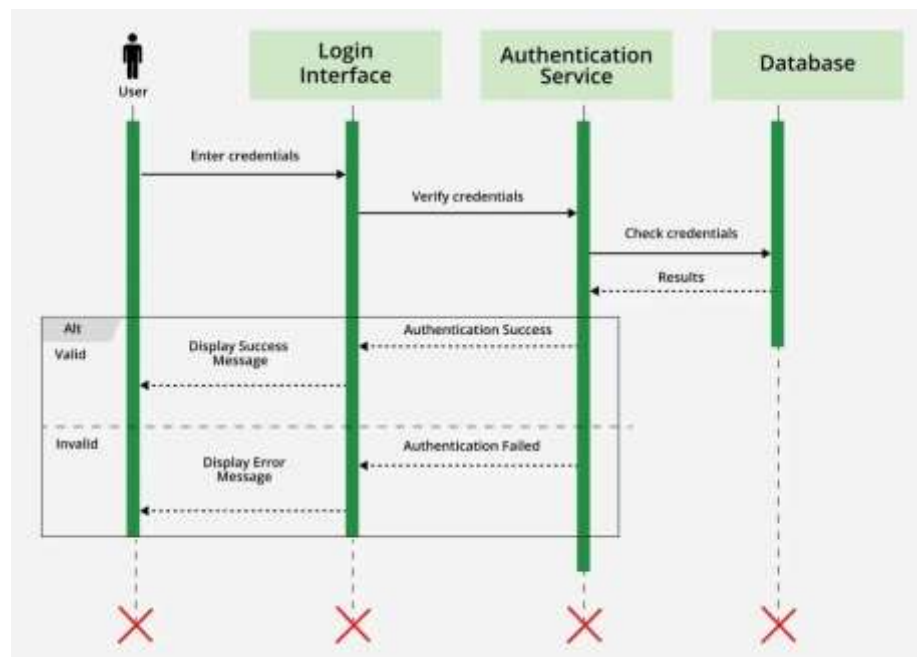


Gambar 2. 6 Activity Diagram[53]

2.17.3 Sequence Diagram

Sequence Diagram merupakan *Diagram* yang dipergunakan pada pemodelan sistem untuk memberikan gambaran interaksi antara objek dalam urutan waktu. *Diagram* ini memperlihatkan bagaimana objek saling berinteraksi dan mengirim pesan satu sama lain [54]. *Sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk

menggambarkan Sequence diagram maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat Sequence diagram juga dibutuhkan untuk melihat skenario yang ada pada use case [50].

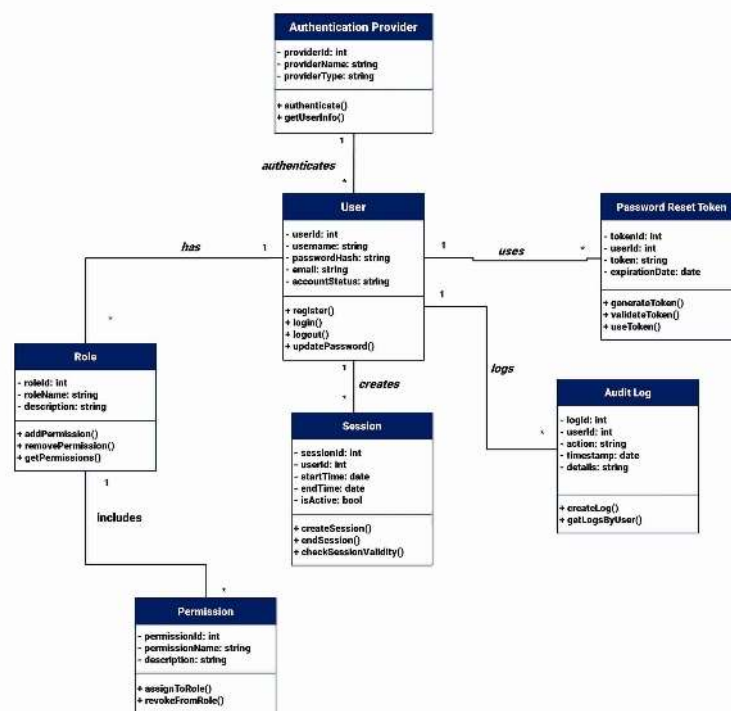


Gambar 2. 7 Sequence Diagram[55]

2.17.4 Class Diagram

Class Diagram merupakan *Diagram* yang digunakan untuk menampilkan struktur sistem berdasarkan pendefinisian kelas-kelas yang dipakai dalam pembangunan sistem. *Class Diagram* juga berfungsi untuk menggambarkan kondisi yang terdapat pada sistem serta layanan yang digunakan untuk

memanipulasi kondisi tersebut [56]. Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [50]



Gambar 2. 8 Class Diagram

2.18 Flowchart

Flowchart merupakan bagan (*chart*) yang mengarahkan alir (*flow*) didalam prosedur atau program sistem secara logika. *Flowchart* adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-

simbol tertentu yang mudah dipahami, mudah digunakan dan standar. Tujuan penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan simbol-simbol yang standar yang dapat dimengerti oleh *programmer* [57].

Selain itu *Flowchart* adalah *Diagram* yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program [58]. Selain itu *Flowchart* merupakan bentuk *Diagram* yang memperlihatkan arus kerja suatu sistem secara grafik dari langkah yang bertahap dalam gambaran simbol yang sudah ditetapkan. Apabila dibahasakan ke Bahasa Indonesia, yaitu menjadi *Diagramalir* yang di artikan mengarahkan atau mewakili sketsa penuntasan sebuah masalah [49]. Pada proses pembuatan *flowchart*, akan disimpan pada antrian dari setiap hasil *token* kemudian setelah semua selesai dalam pembuatannya barulah dilakukan pembuatan gambar semuanya. Sehingga gambar-gambar yang akan dibuat dilakukan penampungan terlebih dahulu

2.19 Draw.Io

Draw.io adalah sebuah aplikasi *open source* untuk membuat aplikasi yang berbasis diagram, dan aplikasi yang paling banyak dipakai *browser-based*, oleh yang mahir dalam aplikasi *Diagram*[59]. Aplikasi ini menyediakan beragam fitur yang mendukung pembuatan *Diagram* seperti *flowchart*, *entity relationship Diagram*(ERD), *Use Case diagram*, hingga *Network Diagram* secara mudah dan fleksibel. Selain itu, *Draw.io* memungkinkan pengguna untuk menyimpan hasil pekerjaan secara langsung ke berbagai platform penyimpanan seperti Google

Drive dan perangkat lokal, sehingga memudahkan dalam pengelolaan dan kolaborasi.

Draw io adalah *website* dan software yang digunakan untuk membuat flowchat, draw io berguna untuk merancang *Use Case Diagram* maupun *Activity Diagram*[60]. Selain itu, keunggulan utama Draw.io terletak pada kemudahan penggunaannya, antarmuka yang intuitif, serta kemampuannya untuk digunakan secara *online* maupun *offline*. Hal ini menjadikan Draw.io sebagai salah satu alat yang efektif dan efisien dalam mendukung proses analisis dan perancangan sistem, khususnya dalam menghasilkan dokumentasi visual yang mudah dipahami oleh pengembang maupun pemangku kepentingan.

2.20 Blackbox

Black Box Testing berfokus pada pengujian fungsional sistem, di mana sistem akan diuji secara fungsional sehingga dapat langsung menilai apakah sistem sudah memenuhi persyaratan fungsional untuk sebuah program. Pengujian ini dilakukan untuk menilai apakah semua fungsi dari sistem telah berjalan atau tidak sehingga memudahkan dalam pengujian sistem selanjutnya [61]. Uji coba *Blackbox* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

- 1) Fungsi-fungsi yang salah atau hilang.
- 2) Kesalahan *interface*.
- 3) Kesalahan dalam struktur data atau akses *database* eksternal.
- 4) Kesalahan performa.
- 5) Kesalahan inisialisasi dan terminasi.

2.21 Penelitian Terkait

Berikut beberapa penelitian terkait dengan aplikasi deteksi jenis bibit buah salak berbasis CNN (studi kasus desa rambah muda).

Tabel 2. 1: Penelitian Terkait

No	Nama	Tahun	Judul	Hasil
1	Muhammad Salman Ikrar Musyaffa, dkk.	(2023)	<i>IndoHerb: Indonesia Medicinal Plants Recognition using Transfer Learning and Deep Learning</i>	Penelitian ini mengembangkan <i>Dataset</i> luas tanaman herbal Indonesia dan melakukan klasifikasi menggunakan beberapa model <i>Deep Learning</i> . Hasil analisis menunjukkan bahwa arsitektur ResNet merupakan salah satu model yang memberikan performa akurasi tinggi dalam mengenali keragaman jenis tanaman berdasarkan karakteristik visualnya
2	Ardi Hidayat, dkk.	(2019)	<i>Detection of Disease on Corn Plants Using Convolutional Neural Network Methods</i>	Menggunakan metode CNN untuk mendiagnosa penyakit jagung di Indonesia dengan <i>Dataset</i> 3.854 gambar, menghasilkan akurasi sebesar 99%
3	Ego Oktafanda	(2022)	Klasifikasi Citra Kualitas Bibit dalam Meningkatkan Produksi Kelapa Sawit Menggunakan Metode CNN	Berhasil mengidentifikasi kualitas bibit kelapa sawit berdasarkan pola dan warna daun dengan tingkat akurasi pengenalan citra yang dikategorikan sangat

				baik
4	Made Doddy Adi Pranatha, dkk.	(2024)	<i>Implementasi Arsitektural Resnet-34 Dalam Klasifikasi Gambar Penyakit Pada Daun Kentang</i>	Menunjukkan bahwa arsitektur ResNet-34 mampu mengatasi masalah <i>vanishing gradient</i> dan mencapai akurasi sekitar 97% pada data uji
5	Samuel Samuel, dkk.	(2022)	Metode Deteksi Pokok Pohon Secara Otomatis pada Citra Perkebunan Sawit Menggunakan Model CNN	Menggunakan arsitektur ResNet-34 dan ResNet-50 untuk deteksi pohon sawit otomatis dengan akurasi pelatihan 84% dan akurasi pengujian rata-rata 71%
6	Sunu Jatmika & Danang Eka Saputra	(2022)	<i>Rice Plants Disease Identification Using Deep Learning with CNN Method</i>	Menyimpulkan bahwa sistem dengan pendekatan <i>transfer learning</i> menghasilkan akurasi 90% , jauh lebih tinggi dibanding pemodelan dari nol yang hanya mencapai 62%
	Arief Rais Bahtiar, dkk.	(2020)	<i>Deep Learning Detected Nutrient Deficiency in Chili Plant</i>	Menggunakan algoritma RCNN untuk mendeteksi kekurangan nutrisi pada tanaman cabai di Boyolali dengan akurasi pengujian sebesar 82,61%
	Jati Nugroho & Supatman	(2021)	Mendeteksi Salak BerLarva dan Tidak BerLarva Menggunakan	Mengembangkan sistem pendeteksi larva di dalam buah salak pondoh menggunakan

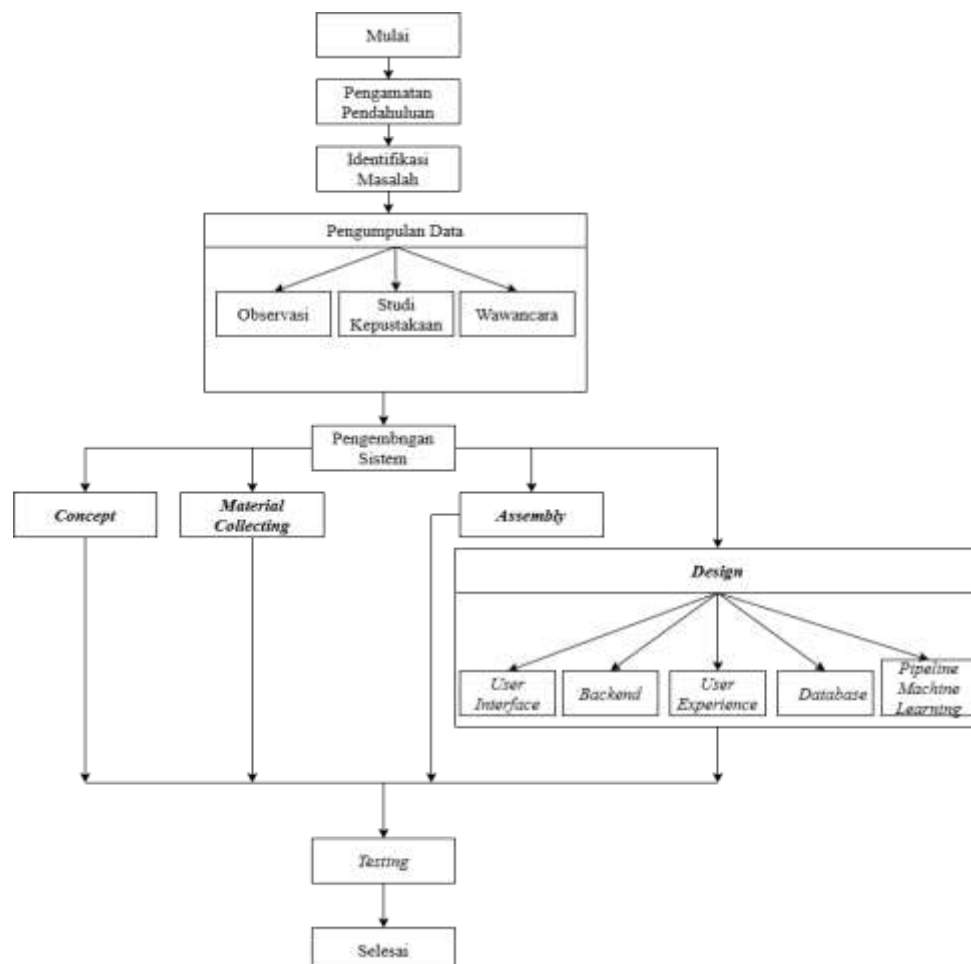
			Metode CNN	CNN untuk membantu identifikasi kualitas buah yang secara visual tampak baik
	Suprihanto, dkk.	(2022)	Analisis Kinerja ResNet-50 dalam Klasifikasi Penyakit pada Daun Kopi Robusta	Melakukan evaluasi kinerja arsitektur ResNet-50 untuk klasifikasi penyakit daun kopi baik dalam skenario kelas biner maupun <i>multiclass</i>
	Muhammad Faqih Dzulqarnain, dkk.	(2019)	<i>mprovement of CNN Accuracy on Salak Classification Based Quality on Digital Image</i>	Meningkatkan akurasi klasifikasi salak dengan cara optimasi <i>preprocessing</i> (noise removal) serta penambahan parameter <i>Stride</i> , <i>Zero Padding</i> , dan <i>Adam optimizer</i> pada lapisan <i>konvolusi</i> yang lebih dalam

BAB 3

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Metode penelitian merupakan tahapan atau langkah-langkah yang dilakukan dalam penelitian. Tujuan dari metodologi penelitian ini agar penelitian yang dibuat sesuai dengan tujuan dari penelitian itu sendiri. Pada penelitian Skripsi ini terdapat beberapa tahapan penelitian yang akan dilakukan seperti yang akan dilakukan seperti Gambar 3.1 Tahapan Penelitian.



Gambar 3. 1 Tahapan Penelitian

3.2 Pengamatan Pendahuluan

Pengamatan pendahuluan merupakan tahap awal dalam penelitian yang bertujuan untuk mengidentifikasi permasalahan, memahami kondisi lapangan, serta memperoleh gambaran umum terkait objek yang akan diteliti. Pada tahap ini, peneliti melakukan observasi secara langsung maupun tidak langsung untuk mengumpulkan informasi awal yang relevan sebagai dasar dalam perumusan masalah dan penentuan metode penelitian.

3.3 Identifikasi Masalah

Pada tahap awal penelitian, identifikasi masalah dilakukan dengan tujuan menemukan dan merumuskan masalah yang akan diteliti. Pada tahap ini, peneliti memeriksa situasi di lapangan untuk mengidentifikasi masalah yang terkait dengan topik penelitian.

3.4 Pengumpulan Data

Peneliti melakukan pengumpulan data pada penelitian ini melalui studi kepustakaan, observasi dan wawancara.

3.4.1 Studi Kepustakaan

Studi kepustakaan mencari teori dan penelitian terkait tentang kasus atau masalah pembuatan sistem ini. Referensi ini didapat dari berbagai sumber, seperti buku, jurnal, dan artikel.

3.4.2 Observasi (Pengamatan)

Pengumpulan data dengan menggunakan indera penglihatan atau melihat untuk menemukan masalah dikenal sebagai observasi atau pengamatan langsung.

Dalam temuan ini, peneliti melakukan peninjauan dan penelitian langsung di kebun salak di Desa Rambah Muda untuk mengidentifikasi jenis bibit salaka dan perbedaan jenis bibit.

3.4.3 Wawancara

Melakukan wawancara secara langsung dengan pemilik kebun salak untuk mengetahui jenis salak dan menemukan perbedaan pada setiap jenis bibit..

3.5 Pengembangan Sistem

Pengembangan sistem merupakan tahap yang dilakukan setelah proses pengumpulan data selesai, yang bertujuan untuk merancang dan membangun sistem sesuai dengan kebutuhan penelitian. Pada tahap ini, dilakukan perancangan hingga implementasi aplikasi yang mampu melakukan klasifikasi jenis bibit buah salak berbasis citra digital.

Dalam penelitian ini, pengembangan sistem dilakukan dengan mengintegrasikan metode *Convolutional Neural Network* (CNN) ke dalam aplikasi berbasis web menggunakan *Flask Framework*. Proses pengembangan sistem mencakup beberapa tahapan utama yaitu *concept*, *material collecting*, *design*, dan *assembly*.

3.5.1 Konsep (*Concept*)

Tahap *concept* merupakan tahap penentuan tujuan dan identifikasi pengguna (*user*). Adapun tujuan yang akan dicapai pada sistem ini adalah membangun suatu aplikasi deteksi jenis bibit buah salak berbasis *Convolutional Neural Network* (CNN) sebagai suatu sistem yang dapat membantu masyarakat dalam mengenali dan membedakan jenis bibit salak.

3.5.2 Perancangan (*Design*)

Tahap *design* merupakan tahap perancangan sistem yang dilakukan berdasarkan hasil analisis kebutuhan yang telah diperoleh sebelumnya. Pada tahap ini, sistem dirancang secara rinci agar dapat dipahami dengan baik serta menjadi acuan dalam proses pengembangan selanjutnya. Spesifikasi sistem dibuat secara jelas sehingga pada tahap berikutnya, yaitu *assembly*, tidak diperlukan pengambilan keputusan baru, melainkan hanya mengimplementasikan rancangan yang telah ditentukan.

Perancangan sistem pada penelitian ini meliputi beberapa bagian, yaitu:

1. Perancangan Antarmuka (*User interface*)

Perancangan antarmuka dilakukan untuk menentukan tampilan aplikasi yang akan digunakan oleh pengguna.

2. Perancangan Arsitektur Sistem (*Backend*)

Perancangan arsitektur sistem dilakukan untuk menentukan alur kerja sistem secara keseluruhan.

3. Perancangan *User Experience* (UX) / Desain Visual

Perancangan ini bertujuan untuk meningkatkan kenyamanan pengguna dalam menggunakan aplikasi.

4. Database

Perancangan basis data dilakukan untuk mengelola dan menyimpan seluruh data yang digunakan dalam sistem secara terstruktur dan terintegrasi.

5. Perancangan *Pipeline Machine Learning*

Pada tahap ini dirancang alur proses *Machine learning* yang digunakan dalam sistem, meliputi proses *preprocessing* data, pelatihan model CNN, evaluasi model, hingga proses prediksi terhadap data baru yang diinputkan oleh pengguna.

3.5.3 Pengumpulan Bahan (*Material Collecting*)

Tahap *material collecting* merupakan tahap pengumpulan bahan yang diperlukan dalam proses pengembangan sistem. Pada tahap ini dilakukan pengumpulan berbagai kebutuhan yang akan digunakan pada tahap selanjutnya, baik berupa data maupun perangkat pendukung sistem.

Bahan utama yang dikumpulkan dalam penelitian ini berupa data citra bibit buah salak dan tumbuhan lain yang diperoleh dari kebun salak di Desa Rambah Muda. Data citra tersebut digunakan sebagai *Dataset* dalam proses pelatihan dan pengujian model *Convolutional Neural Network* (CNN).

3.5.4 Implementasi Sistem (*Assembly*)

Tahap *assembly* merupakan tahap implementasi dimana seluruh komponen sistem yang telah dirancang pada tahap sebelumnya mulai dibangun dan diintegrasikan menjadi sebuah aplikasi yang utuh. Pada tahap ini, sistem dikembangkan sehingga dapat dijalankan dan diuji dalam kondisi sebenarnya untuk mengetahui apakah telah sesuai dengan tujuan yang diharapkan.

Pembuatan aplikasi pada penelitian ini dilakukan berdasarkan hasil perancangan pada tahap *design*, yang meliputi perancangan antarmuka (*user interface*), arsitektur sistem (*backend*), serta *pipeline machine learning*.

Beberapa komponen pendukung yang berperan penting dalam implementasi sistem antara lain perangkat keras (*hardware*) dan perangkat lunak (*software*). Adapun spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan adalah sebagai berikut :

1. Perangkat keras (*hardware*), antara lain :

Tabel 3 1 Tabel Perangkat Keras

<i>Processor</i>	:	Intel(R) Celeron(R) N5100 @ 1.10GHz 1.11 GHz
<i>Memory (RAM)</i>	:	8,00 GB (7,84 GB usable)
<i>System</i>	:	64-bit operating system, x64-based <i>processor</i>

2. Perangkat Lunak (*software*), antara lain :

Tabel 3 2 Tabel Perangkat Lunak

<i>Sistem Operasi</i>	:	<i>Windows 11 Home Single Language</i>
<i>Tool</i>	:	<i>Visual Studio Code Draw.io Google Colab</i>

3.5.5 Pengujian (*Testing*)

Pengujian yaitu uji coba yang dilakukan terhadap sistem yang dibangun apakah telah sesuai dengan yang diharapkan atau tidak. Pengujian yang dilakukan terdiri dari:

1. *Pengujian Blackbox*, digunakan untuk menguji tingkat kemampuan *User interface* terhadap sistem yang dibangun.

3.6 Kesimpulan dan Saran

Tahap penarikan kesimpulan merupakan tahap akhir dalam penelitian yang bertujuan untuk merangkum hasil yang diperoleh selama proses pengembangan sistem. Pada tahap ini dilakukan analisis terhadap hasil pengujian sistem serta performa model *Convolutional Neural Network* (CNN) dalam mengklasifikasikan jenis bibit buah salak.

Kesimpulan yang dihasilkan mencerminkan tingkat keberhasilan sistem dalam mencapai tujuan penelitian, yaitu membangun aplikasi deteksi jenis bibit buah salak berbasis CNN yang dapat membantu pengguna dalam mengenali jenis bibit secara otomatis.