

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Dalam pola makan sebagian besar penduduk Indonesia, daging memainkan peran penting sebagai sumber gizi yang memberikan kenikmatan rasa dan juga kandungan vitamin serta mineral yang esensial bagi kesehatan tubuh manusia. Meskipun demikian, penting untuk diingat bahwa tidak semua jenis daging cocok untuk dikonsumsi oleh masyarakat Indonesia, terutama bagi mereka yang mengikuti kepercayaan atau keyakinan agama tertentu yang melarang konsumsi daging babi.

Dalam upaya untuk mengidentifikasi dan membedakan jenis daging sapi dengan daging babi dengan lebih tepat, para peneliti telah mengadopsi berbagai teknik klasifikasi, termasuk Convolutional Neural Network (CNN), sebagai salah satu pendekatan dalam Deep Learning. CNN telah terbukti memberikan hasil yang lebih baik dalam mengklasifikasikan objek, termasuk membedakan antara daging sapi dan daging babi. Melalui metode klasifikasi ini, data tentang daging dapat dianalisis dengan lebih akurat, memungkinkan pengenalan pola dan karakteristik khusus dari setiap jenis daging.

Klasifikasi sendiri merupakan salah satu teknik yang sangat signifikan dalam data mining. Teknik ini memiliki kemampuan untuk melakukan prediksi terhadap label kelas suatu data. Dengan menggunakan algoritma klasifikasi, data dapat dikelompokkan atau dikategorikan ke dalam kelas-kelas yang telah ditentukan sebelumnya berdasarkan pada fitur-fitur yang ada dalam data tersebut[9].

Dalam konteks Convolutional Neural Network (CNN), ResNet50 adalah salah satu arsitektur jaringan yang sangat relevan. ResNet50 merupakan salah satu varian dari ResNet (Residual Network) yang memiliki 50 lapisan. Arsitektur ini dikenal karena kemampuannya dalam mengatasi masalah vanishing gradient yang sering terjadi pada jaringan saraf dalam deep learning. ResNet50 telah berhasil digunakan dalam berbagai tugas visi komputer, termasuk dalam klasifikasi gambar.

Penelitian pertama, tentang penerapan *Convolutional Neural Network* untuk mengklasifikasikan Sidik Jari menggunakan *ResNet-50* oleh [8] pada tahun 2020. Penelitian tersebut memiliki tujuan mengklasifikasikan sidik jari melalui citra digital. Sistem ini melibatkan tahap *pra*-pemrosesan data untuk mempersiapkan citra sidik jari sebelum dilakukan klasifikasi. Metode yang digunakan untuk klasifikasi adalah *Convolutional Neural Network* (CNN) dengan menggunakan arsitektur *ResNet-50*. Data yang digunakan pada penelitian tersebut didapatkan pada situs *website National Institute of Standards and Technology (NIST)* berupa citra sidik jari *grayscale* 8-bit berukuran 512×512 piksel. Hasil dari penelitian tersebut mendapatkan akurasi data pelatihan 99,52% yang dihasilkan dari *preprocessing CLAHE*, *learning rate* 0,01, dan optimasi SGD.

Penelitian kedua, tentang mengidentifikasi jenis daging menggunakan algoritma *convolutional neural network* oleh [5] pada tahun 2021. Penelitian ini bertujuan untuk membedakan berbagai jenis daging menggunakan teknologi *computer vision*. Dalam studi ini, identifikasi jenis daging dilakukan melalui ekstraksi warna dan pendeteksian tepi dengan CNN, menghasilkan identitas berupa teks sesuai dengan jenis daging yang terdapat pada citra. Untuk keperluan ini, digunakan dataset yang terdiri dari 2,250 gambar untuk setiap jenis daging, menjadikan total keseluruhan dataset sebanyak 11,250 gambar.. Hasil penelitian ini menghasilkan 82,82% *Training Accuracy* dengan ukuran citra 50x50 piksel dan 85,42% *Training Accuracy* dengan ukuran citra 100x100 piksel yang dihasilkan dari konfigurasi algoritma *Convolutional Neural Network*.

Penelitian ketiga, tentang klasifikasi retak ban kendaraan menggunakan arsitektur *ResNet50* oleh [10] pada tahun 2023. Pada penelitian tersebut digunakan metode *Deep Learning* seperti *Resnet-50*. Data yang digunakan pada penelitian tersebut didapatkan melalui *website Kaggle* yang bersifat *open source*. Dengan 6 skenario parameter *tunning* yang dibuat, enghasilkan bahwa model mendapatkan nilai yang baik dengan tingkat akurasi sebesar 94%, presisi 94%, dan *recall* 94%.

Penelitian keempat, tentang klasifikasi citra daging sapi dan daging babi menggunakan ekstraksi ciri dan *convolutional neural network* oleh [11] pada

tahun 2022. Tujuan dari penelitian ini adalah untuk mencari tingkat akurasi yang paling tinggi melalui serangkaian eksperimen dan untuk mengetahui apakah penerapan CLAHE pada data citra dapat meningkatkan akurasi tersebut. Dalam penelitian ini, metode *Deep Learning* dengan arsitektur *Convolutional Neural Network* EfficientNet-B0 digunakan untuk klasifikasi citra. Dataset terdiri dari 3.000 citra yang dibagi ke dalam tiga kelas: daging sapi, daging babi, dan daging campuran. Penelitian ini membandingkan penggunaan data citra asli dengan data yang telah diolah menggunakan *Contrast Limited Adaptive Histogram Equalization*. Hasil eksperimen menunjukkan bahwa performa klasifikasi terbaik dicapai dengan akurasi 95.17%, presisi 92.72%, *recall* 95.5%, dan skor f1 94.09% pada data citra asli. Hal ini dicapai dengan pengaturan neuron sebanyak 256 pada lapisan dense pertama, ukuran batch 32, laju pembelajaran 0.01, dan penggunaan Adam Optimizer.

Penelitian kelima, tentang perbandingan arsitektur *ResNet50* dan *ResNet101* dalam Klasifikasi Kanker Serviks pada Citra *Pap Smear* oleh [12] pada tahun 2021. Tujuan utama dari studi ini adalah untuk melakukan perbandingan terhadap efektivitas penggunaan dua arsitektur jaringan saraf tiruan yang berbeda, yaitu *ResNet50* dan *ResNet101*, dalam pengolahan citra *Pap smear*. Fokus utama adalah untuk mengidentifikasi kanker serviks dan menilai kinerja kedua arsitektur tersebut dalam mengklasifikasikan dan mendeteksi indikator penyakit. Untuk keperluan penelitian, dataset yang digunakan berasal dari SIPaKMeD, yang mencakup lima kategori berbeda dengan total 4049 sampel data. Penelitian ini memberikan hasil bahwa model arsitektur *ResNet50* mendapatkan hasil akurasi sebesar 91% lebih besar dibanding dengan arsitektur *ResNet101* dengan hasil akurasi sebesar 89%.

Penelitian keenam, mengenai Pengembangan sistem pendeteksi *Indonesia Sign Language* berbasis *web* [13] pada tahun 2021. Penelitian ini memiliki tujuan untuk menjadikan sebuah sistem sebagai jembatan antara masyarakat non tuna rungu dengan masyarakat tuna rungu. Data pada penelitian ini menggunakan dataset bernama ISLBISINDO1 yang berisi 2080 citra dari 26 kelas. Penelitian ini

menghasilkan sistem pendeteksi Bahasa isyarat yang mampu mendeteksi berbagai macam bentuk dengan jumlah 26 kelas bahasa isyarat dari huruf alfabet.

Tabel 2.1 Penelitian Terdahulu

No	Peneliti, Tahun	Judul	Metode	Hasil
1.	Novelita Dwi Miranda, Ledy Novamizanti, Syamsul Rizal. Tahun 2020	<i>Convolutional Neural Network</i> Pada Klasifikasi Sidik Jari Menggunakan <i>Resnet-50</i>	<i>Convolutional Neural Network</i> dengan arsitektur <i>ResNet-50</i>	Sistem dapat mengidentifikasi 5 pola sidik jari dengan akurasi pelatihan 99,52%, akurasi validasi 95,05%, loss pelatihan 0,016, dan loss validasi 0,229
2.	Peter Winardia , Endang Setyati. 2021	Identifikasi Jenis Daging Menggunakan Algoritma <i>Convolution Neural Network</i>	<i>Convolutional Neural Network</i>	<p>a. Ukuran Citra 50 x 50 piksel dengan 150 epoch menghasilkan Training Accuracy 82.82% dan Validation Accuration 72.27%</p> <p>b. Ukuran citra 100 x 100 piksel dengan 150 epoch menghasilkan Training Accuracy 85.42% dan Validation Accuracy 72.69%</p>
3.	Iwansyah Edo Hendrawan, Riza Ibnu Adam, Chaerur Rozikin	Klasifikasi Retak Ban Kendaraan Menggunakan Arsitektur ResNet 50	ResNet 50	<ul style="list-style-type: none"> - Nilai Akurasi 94% - Presisi 94% - Recall 94%

No	Peneliti, Tahun	Judul	Metode	Hasil
4.	Gusrifaris Yuda Alhafis, Jasril, Suwanto Sanjaya, Fadhilah Syafria, Elvia Budianita	Klasifikasi Citra Daging Sapi dan Daging Babi Menggunakan Ekstraksi Ciri dan <i>Convolutional Neural Network</i>	Convolutional Neural Network arsitektur EfficientNet-B0	<ul style="list-style-type: none"> - Adam Optimizer menghasilkan 95.17% accuracy, 92.72% precision, 95.5% recall, dan 94.09% f1 score. - Penggunaan CLAHE pada penelitian ini tidak dapat meningkatkan akurasi.
5.	Za'imatun Niswati, Rahayuning Hardatin, Meia Noer Muslimah, Siti Nur Hasanah	Perbandingan Arsitektur ResNet50 dan ResNet101 dalam Klasifikasi Kanker Serviks pada Citra <i>Pap Smear</i>	<ul style="list-style-type: none"> - ResNet101 - ResNet 50 	<ul style="list-style-type: none"> - ResNet50 memberikan hasil akurasi 91% - ResNet101 memberikan hasil akurasi sebesar 89%
6.	Mohammad Idham Fachrurrozi, Yisti Vita Via, Wahyu S.J Saputra	Implementasi Sistem Pendeteksi <i>Indonesia Sign Language</i> Bisindo Berbasis <i>Web Flask</i>	<i>Framework Flask</i>	Aplikasi yang berbasis pada framework Flask ini dirancang untuk mendeteksi bahasa isyarat, mampu mengenali berbagai bentuk dari 26 kelas isyarat yang mewakili huruf alfabet dari A sampai Z.

2.2 Landasan Teori

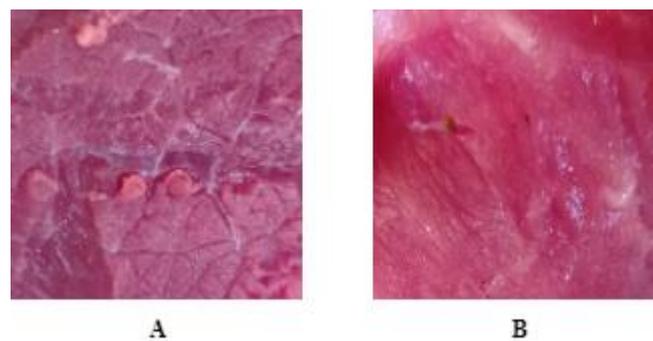
Berikut beberapa landasan teori yang mengkaji tentang beberapa pembahasan yang akan digunakan dalam penelitian kali ini.

2.2.1 Daging Sapi dan Daging Babi

Daging sapi tidak hanya menjadi makanan penting bagi manusia, tetapi juga sangat bernilai karena kandungan gizinya, khususnya protein, yang sangat berguna untuk pertumbuhan. Oleh karena harga daging babi yang relatif lebih rendah dibandingkan daging sapi, hal ini sering kali memicu praktik

mencampuradukkan daging sapi dengan daging babi, sebagai strategi untuk mendapatkan keuntungan lebih besar.

Bagi kebanyakan orang, membedakan antara daging sapi dan daging babi bisa menjadi tugas yang sulit karena perbedaannya tidak selalu jelas. Daging babi umumnya memiliki warna yang lebih terang daripada daging sapi. Dari segi tekstur, daging sapi biasanya lebih keras dan kurang lunak dibandingkan daging babi. Perbedaan pada tekstur dan warna pada kedua daging ini direpresentasikan pada gambar 2.1. Selain itu, serat yang ada pada daging sapi tampak lebih jelas dan terdefinisi dibandingkan dengan serat pada daging babi. Namun, perbedaan-perbedaan ini mungkin tidak diketahui oleh konsumen jika mereka tidak memiliki pengetahuan dasar mengenai perbedaan antara kedua jenis daging tersebut [14].



Gambar 2.1 Daging Sapi (A) Daging Babi (B)

2.2.2 *Machine Learning*

Pembelajaran mesin atau *Machine Learning* adalah bagian dari kecerdasan buatan yang merupakan disiplin ilmu yang melibatkan perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku berdasarkan pada data empiris[15]. *Machine Learning* memiliki dua konsep pembelajaran utama.

Pertama adalah *Supervised Learning*, yang merupakan teknik *Machine Learning* yang membangun suatu fungsi berdasarkan data latihan yang sudah diketahui secara detail dan terklasifikasi dengan baik. Dalam teknik ini, terdapat data latihan yang telah diberi label dengan baik yang akan digunakan untuk membentuk model data. Setelah itu, model tersebut akan diuji dengan

menggunakan data tes yang baru dan menghasilkan keluaran yang diharapkan berdasarkan informasi yang telah dipelajari dari data Latihan[12].

Teknik kedua adalah *Unsupervised Learning*, yang merupakan teknik *Machine Learning* yang berfokus pada representasi pola dari *input* data latihan tanpa adanya pengklasifikasian yang telah ditentukan sebelumnya. Dalam teknik *Unsupervised Learning*, tidak ada *role* model atau informasi yang tersedia sebagai acuan dalam proses pembelajaran dan pengujian. Hal ini berbeda dengan *Supervised Learning*. Teknik *Unsupervised Learning* menjadi penting dalam *Machine Learning* karena mencerminkan cara kerja otak manusia, di mana pembelajaran dilakukan tanpa memiliki model atau contoh yang telah diberikan sebelumnya untuk menyelesaikan masalah dengan data yang baru[16].

2.2.3 Deep Learning

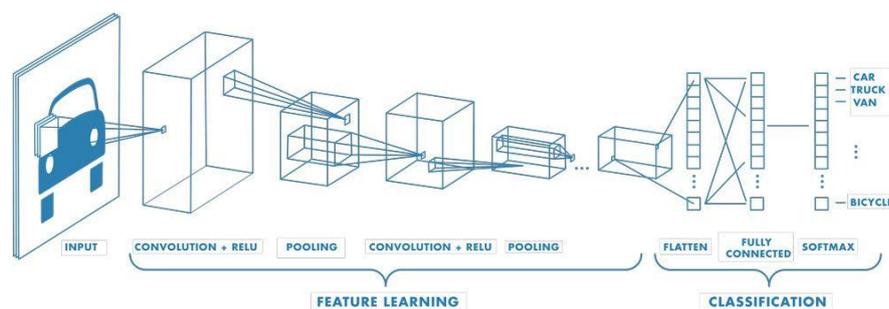
Deep Learning, yang merupakan subbidang dari kecerdasan buatan dan pembelajaran mesin (*machine learning*), berkembang dari konsep jaringan saraf tiruan dengan berlapis-lapis (*neural network multiple layer*). Tujuannya adalah untuk mencapai tingkat akurasi yang lebih tinggi dalam berbagai tugas, termasuk deteksi objek, pengenalan suara, terjemahan bahasa, dan sebagainya. Salah satu pembeda utama *Deep Learning* dari teknik pembelajaran mesin tradisional adalah kemampuannya untuk secara otomatis belajar representasi data, seperti gambar, video, atau teks, tanpa perlu aturan kode atau pengetahuan spesifik manusia tentang domain tersebut. [11].

Deep learning juga dapat diartikan sebagai pemanfaatan jaringan syaraf tiruan yang memiliki struktur yang kompleks, dengan beberapa lapisan yang saling terhubung secara mendalam, untuk menghasilkan *output* yang diinginkan.

2.2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan perkembangan dari *Multilayer Perceptron* (MLP) yang termasuk dalam jenis *neural network feed forward* (*non-recurrent*). CNN dirancang khusus untuk mengolah data dua dimensi, terutama dalam konteks analisis gambar. CNN termasuk dalam kategori *Deep Neural Network* karena memiliki struktur jaringan yang dalam dan sering digunakan dalam pemrosesan data citra. CNN digunakan untuk analisis visual

gambar, deteksi objek, dan pengenalan objek pada gambar, yang melibatkan representasi vektor berdimensi tinggi dan parameter yang banyak untuk menggambarkan jaringan tersebut. Secara keseluruhan, CNN tidak jauh berbeda dengan neural *network* konvensional lainnya. CNN terdiri dari neuron dengan bobot (*weight*), bias, dan fungsi aktivasi[17]. Gambar berikut merupakan ilustrasi dari model arsitektur CNN.



Gambar 2.2 Arsitektur CNN

2.2.5 *Transfer Learning*

Transfer Learning merupakan metode yang digunakan untuk mempercepat dan meningkatkan proses pembelajaran dengan mengambil pengetahuan yang telah diperoleh dari satu domain dan menerapkannya ke domain yang berkaitan. Proses ini terbagi menjadi dua tahapan utama: tahap *pre-training* dan tahap *finetuning*. Selama tahap *pre-training*, model dasar digunakan dan bobotnya dipertahankan tanpa perubahan atau optimasi, hal ini bertujuan untuk mempercepat pelatihan. Sementara itu, tahap *finetuning* melibatkan pelatihan ulang dari model yang sudah dikembangkan pada tahap *pre-training*, dengan menyesuaikan sebagian lapisan dari model dasar untuk meningkatkan akurasi secara menyeluruh[18].

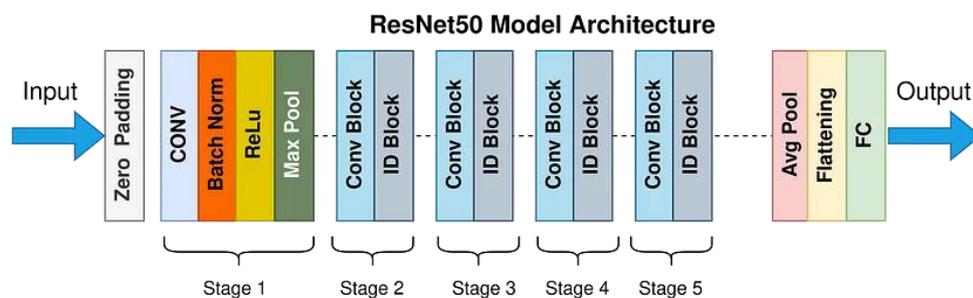
2.2.6 *Arsitektur ResNet-50*

Proses sampling dilaksanakan secara langsung oleh lapisan konvolusi, yang melibatkan proses konversi setiap byteplot ke dalam format gambar RGB, serta penerapan normalisasi batch yang terjadi langsung setelah setiap konvolusi

dan sebelum aktivasi ReLU. Transformasi byteplot menjadi gambar RGB ini mengubah ukuran gambar menjadi dimensi 224x224, serta mengurangi nilai rata-rata RGB dari data dataset pada tiap piksel[18].

Langkah-langkah yang dilakukan oleh lapisan konvolusi dalam proses pengambilan sampel meliputi konversi setiap *byteplot* menjadi gambar RGB, aplikasi normalisasi *batch* setelah setiap konvolusi, dan sebelum aktivasi ReLU. Proses konversi *byteplot* ke gambar RGB ini menghasilkan gambar dengan dimensi 224x224 piksel dan melibatkan penyesuaian nilai rata-rata RGB di setiap piksel berdasarkan data dataset.[18].

Jika ukuran dimensi *input* dan *output* pada lapisan konvolusi sama, maka pintasan identitas diaplikasikan. Namun, ketika terjadi peningkatan dimensi, pintasan proyeksi digunakan, yang mengubah dimensi tersebut melalui operasi *Pointwise Convolution*. Dalam kedua skenario ini, pintasan-pintasan ini bertugas mengonversi setiap *byteplot* menjadi gambar RGB ketika melalui dua ukuran peta fitur yang berbeda. Jaringan ini juga termasuk 1000 lapisan yang terhubung sepenuhnya (*fully connected*) dengan fungsi aktivasi *softmax*. Total jumlah lapisan dengan bobot adalah 50, dengan total 25.534.592 parameter yang dapat dilatih[18]. Berikut adalah contoh gambaran dari arsitektur ResNet50.



Gambar 2.3 Arsitektur ResNet50

2.2.7 Confusion Matrix

Confusion matrix merupakan sebuah tabel yang berfungsi untuk menilai kinerja model klasifikasi. Tabel ini dibagi menjadi empat bagian, dengan setiap bagian mewakili tipe hasil klasifikasi yang berbeda. Ini memungkinkan perbandingan antara nilai-nilai yang diprediksi oleh model dan nilai-nilai aktual,

sehingga memberikan pandangan mengenai efektivitas model klasifikasi tersebut. Beberapa nilai penting yang termasuk dalam *confusion matrix* digunakan untuk evaluasi model.

a. Accuracy

Accuracy adalah probabilitas bahwa suatu kelas dalam klasifikasi akan terklasifikasi secara akurat, dengan mempertimbangkan semua nilai yang ada dalam kelas tersebut.

$$\mathbf{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

b. Precision

Precision merupakan rasio antara jumlah sampel yang dengan tepat terklasifikasi sebagai positif dan jumlah total sampel yang diklasifikasikan sebagai positif.

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

c. Recall

Recall mengindikasikan jumlah sampel yang diklasifikasikan dengan benar sebagai positif. Hal ini digunakan untuk mengevaluasi seberapa baik model dalam mengidentifikasi jumlah data yang diklasifikasikan dengan benar.

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

d. F1 Score

F1 score merupakan nilai rata-rata yang membandingkan *precision* dan *recall* secara bersamaan dengan memberikan bobot pada keduanya.

$$\mathbf{F1\ Score} = 2 \times \frac{\mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}}$$

2.2.8 Flask

Flask adalah sebuah *framework web* minimalis yang dibangun dengan menggunakan bahasa pemrograman Python. Ia dirancang untuk memudahkan pengembangan aplikasi *web* dengan pendekatan yang sederhana dan fleksibel. *Flask* sangat cocok digunakan untuk proyek-proyek kecil hingga menengah yang membutuhkan kecepatan pengembangan yang tinggi. *Flask* merupakan *framework web* yang minimalis dan tergolong sebagai "*micro framework*". Ini berarti bahwa *Flask* menyediakan inti yang sederhana dan minimal, tetapi dengan kemampuan untuk diperluas sesuai kebutuhan. *Flask* telah digunakan secara luas oleh pengembang aplikasi kecerdasan buatan untuk membangun berbagai jenis aplikasi *web*, mulai dari proyek kecil hingga skala besar. Kekuatan dan kelebihan *Flask* terletak pada kesederhanaannya yang memungkinkan pengembang untuk memiliki kontrol penuh atas aplikasi *web* mereka, serta fleksibilitasnya yang memungkinkan pengembang untuk memperluas dan menyesuaikan *framework* sesuai kebutuhan proyek.

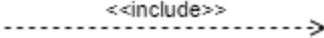
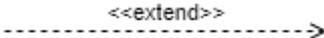
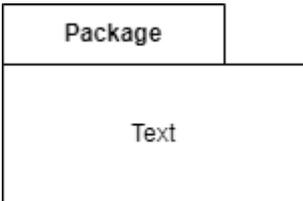
2.2.9 Unified Modelling Language (UML)

UML adalah suatu bentuk bahasa visual yang digunakan untuk melakukan pemodelan dan komunikasi tentang suatu sistem. Hal ini dilakukan melalui penggunaan diagram dan teks-teks pendukung yang membantu dalam menggambarkan aspek-aspek sistem secara jelas dan terstruktur[19]. Tujuan utama *UML* adalah menyediakan cara yang konsisten dan jelas untuk menggambarkan sistem secara visual, agar para pemangku kepentingan, seperti pengembang perangkat lunak, analis bisnis, arsitek sistem, dan pemilik produk, dapat memahami dan berkomunikasi mengenai sistem dengan lebih efektif. *UML* juga berfungsi sebagai alat untuk menganalisis, merancang, dan mengimplementasikan sistem dengan lebih baik. Pada penelitian ini, menggunakan diagram *UML* yang terdiri dari *Use Case Diagram* dan *Activity Diagram*.

A. Use Case Diagram

Use case atau diagram *use case* adalah bentuk pemodelan yang digunakan untuk menggambarkan perilaku atau kelakuan dari sistem informasi yang akan dikembangkan[19]. Berikut beberapa elemen pendukung pada *Use Case Diagram* yang direpresentasikan pada tabel 2.2

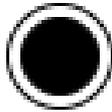
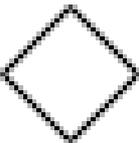
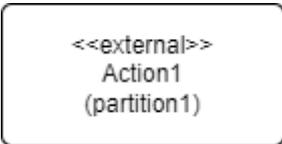
Tabel 2.2 Tabel Elemen *Use Case Diagram*

No.	Nama Elemen	Simbol Elemen
1.	Use Case	
2.	Actor	
3.	Association	
4.	Generalization	
5.	Inclusion	
6.	Extend	
7.	Package	

B. Activity Diagram

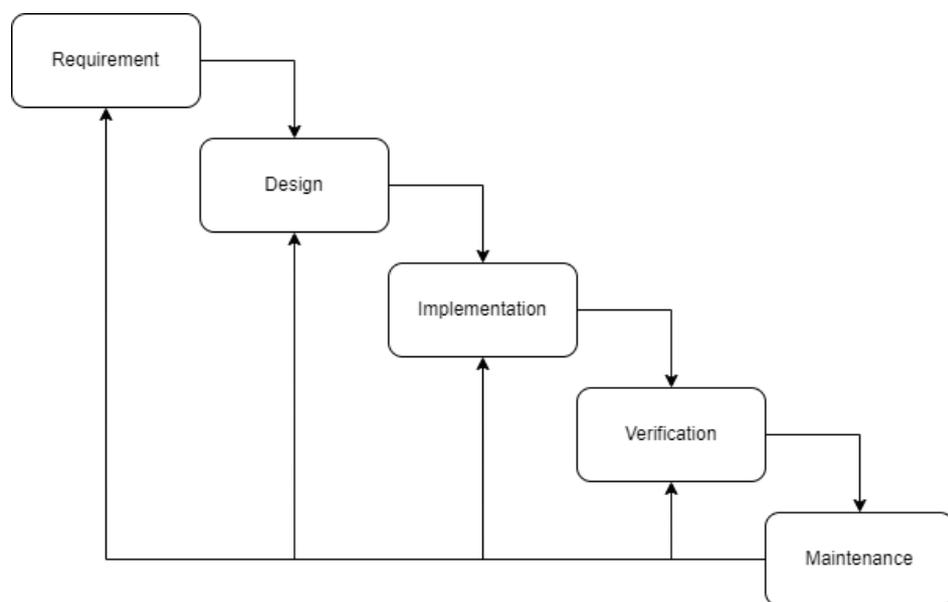
Merupakan salah satu jenis diagram yang digunakan dalam *Unified Modeling Language* (UML) untuk menggambarkan alur kerja atau urutan aktivitas dalam suatu proses atau sistem. Diagram ini membantu dalam memodelkan bagaimana aktivitas-aktivitas berhubungan satu sama lain, bagaimana alur kerja sistem berjalan, serta menunjukkan pemrosesan data atau objek yang terlibat dalam aktivitas tersebut. *Activity* diagram menggunakan notasi grafis untuk menggambarkan elemen-elemen yang terlibat dalam alur kerja, termasuk aktivitas, keputusan, penggabungan, titik awal, dan titik akhir. Berikut adalah beberapa elemen penting dalam *activity diagram* yang direpresentasikan pada Tabel 2.3

Tabel 2.3 Elemen *Activity Diagram*

No	Nama Elemen	Simbol Elemen
1.	Initial Node	
2.	Activity	
3.	Final Node	
4.	Decision Node	
5.	Partition	

2.2.10 Waterfall

Model Waterfall adalah salah satu metode *Software Development Life Cycle* (SDLC) yang umum digunakan dalam pengembangan sistem informasi atau perangkat lunak. Pendekatan ini mengikuti langkah-langkah yang sistematis dan berurutan. Proses dalam model ini dimulai dari tahap perencanaan dan berlanjut hingga tahap pengelolaan atau pemeliharaan, dilakukan secara bertahap[20].



Gambar 2.4 Metode *Waterfall*

a. *Requirement*

Dalam fase ini, interaksi dengan pengembang sistem menjadi krusial untuk memahami kebutuhan pengguna terhadap perangkat lunak dan batasan-batasan yang berhubungan dengan software tersebut. Informasi dan data yang relevan bisa didapatkan melalui berbagai cara, termasuk wawancara, diskusi, atau survei langsung. Data yang terkumpul lalu dianalisis untuk mengidentifikasi informasi yang dibutuhkan oleh pengguna.

b. Design

Dalam fase ini, pengembang bertanggung jawab untuk merancang desain sistem, yang penting untuk menetapkan kebutuhan perangkat keras dan sistem, serta berkontribusi dalam menentukan arsitektur keseluruhan dari sistem tersebut.

c. Implementation

Selama fase ini, sistem awalnya dibangun sebagai serangkaian program kecil atau unit, yang nantinya akan diintegrasikan dalam fase berikutnya. Setiap unit ini dikembangkan dan diuji secara terpisah untuk memverifikasi fungsionalitasnya, dalam proses yang dikenal sebagai pengujian unit.

d. Verification

Selama tahap ini, sistem menjalani verifikasi dan pengujian untuk memastikan bahwa ia memenuhi semua atau sebagian dari persyaratan yang ditetapkan. Pengujian ini terbagi ke dalam beberapa kategori, yang meliputi pengujian unit (yang dilakukan pada modul kode spesifik), pengujian sistem (untuk mengevaluasi reaksi sistem saat semua modulnya terintegrasi), dan pengujian penerimaan (yang dilakukan bersama dengan klien atau pengguna untuk menilai apakah sistem memenuhi kebutuhan mereka).

e. Maintenance

Ini merupakan fase akhir dalam proses pengembangan *waterfall*. Pada tahap ini, perangkat lunak yang sudah selesai dikembangkan akan diimplementasikan dan masuk ke fase pemeliharaan. Fase pemeliharaan ini melibatkan perbaikan dari kesalahan-kesalahan yang tidak teridentifikasi pada tahap-tahap sebelumnya.