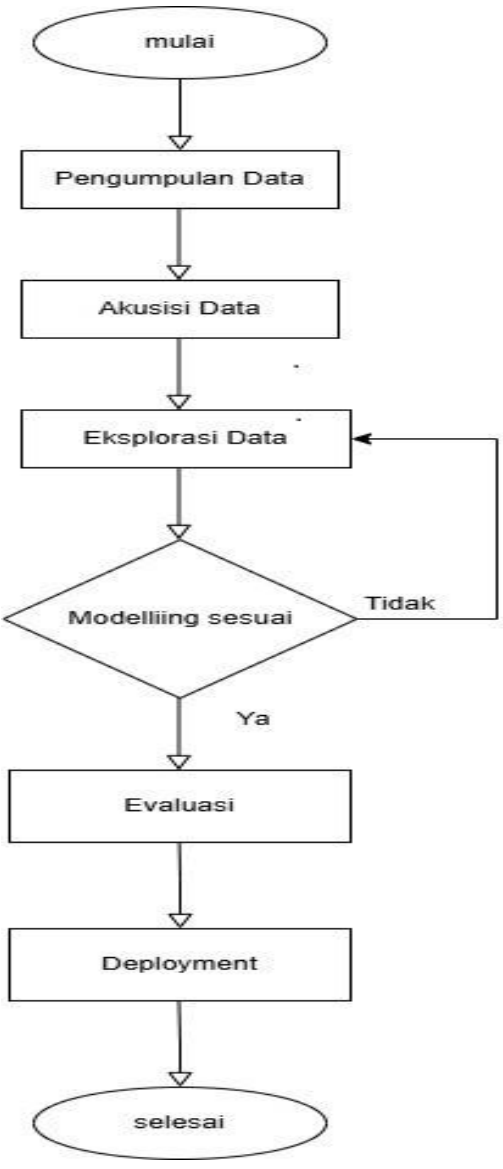
BAB III
METODOLOGI PENELITIAN

Penelitian ini mengaplikasikan metode pengembangan Deep Learning dengan menggunakan CNN dan arsitektur MobileNetV2. Terdapat beberapa langkah yang harus dilakukan untuk mencapai hasil yang diinginkan, di antaranya sebagai berikut. Studi literatur dilakukan bertujuan untuk mencari informasi mengenai *Deep learning* dengan metode CNN dengan menggunakan algoritma *MobileNetV2.* Pada tahapan ini dimulai dari pengenalan dan analisis permasalahan yang terjadi pada kehidupan sehari-hari atau biasa disebut dengan *problem scoping.* Setelah itu tahap selanjutnya yaitu pengambilan *Dataset* atau *data acquisition*. Jika *Dataset*nya sudah terkumpul, maka langkah selanjutnya yaitu tahap *pre-processing* data untuk pengolahan data yang akan digunakan.

Data processing adalah operasi pada data untuk mengelompokkan, mengambil, dan mengubahnya, sedangkan data *exploration* merupakan proses investigasi untuk menentukan pola atau tren, dan mengidentifikasi hubungan yang mungkin ada diantara variabel. Pada proses ini, data akan diolah secara keseluruhan dengan berbagai macam model AI. Hasil dari proses ini merupakan informasi-informasi yang digunakan dalam merepresentasikan hubungan antar data. Tahap selanjutnya adalah data training dan data testing, skala yang digunakan adalah 80:20 dimana 80% untuk *data train* dan 20% untuk data *test. Training* data adalah kumpulan data awal yang digunakan untuk melatih suatu model dalam mengimplementasikan teknologi dan menghasilkan keluaran yang baik. Setelah melalui tahapan training dan testing, model akan dievaluasi dengan berbagai metrik kinerja, seperti akurasi, presisi, recall, dan F1-score menggunakan *Confussion matrix. Confussion matrix* memberikan gambaran yang lebih jelas tentang bagaimana model bekerja, terutama ketika ada ketidakseimbangan kelas dalam *Dataset*. Hal ini bertujuan untuk mencari model mana yang memberikan hasil evaluasi terbaik.

3.1 Alur Penelitian

Sebagai bagian penting dari penelitian ini, penulis telah merancang metodologi yang sistematis untuk memastikan bahwa setiap tahap dari proses penelitian dapat dilaksanakan dengan baik. Diagram alir berikut ini akan memberikan gambaran yang jelas tentang langkah-langkah yang akan diambil dalam penelitian ini. Dengan visualisasi ini, penulis berharap dapat memperjelas alur kerja dan memudahkan pemahaman mengenai bagaimana penulis akan mencapai tujuan penelitian yang telah ditetapkan. berikut diagram alir pada Gambar 3.1 dari sistematika projek ini:



Gambar 3.1 Diagram Alir Penelitian

Gambar 3.1 merupakan proses pembuatan model *Deep Learning* untuk *Citrus Disease Detector* secara umum. Implementasi pemrograman pada *Deep Learning* yang menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur Mo,bileNetV2 dapat menjadi jauh lebih kompleks dan bervariasi tergantung pada kebutuhan dan tujuan spesifik model yang akan dibangun. Langkah-langkah yang diuraikan sebelumnya mencerminkan tahapan umum dalam pengembangan model *Deep Learning* berbasis CNN, meskipun penyesuaian tertentu sering kali diperlukan berdasarkan tugas yang dihadapi.

Metode CNN pada *Deep Learning* dirancang untuk menangkap pola lokal dan ketergantungan spasial dalam data visual. Hal ini membuat CNN sangat efektif dalam analisis citra, terutama untuk klasifikasi dan deteksi objek. CNN bekerja dengan memanfaatkan lapisan konvolusi untuk mengekstraksi fitur dari data input, yang memungkinkan jaringan mengenali pola-pola penting dalam citra seperti tepi, tekstur, dan bentuk. MobileNetV2, sebagai salah satu arsitektur CNN, dirancang untuk meningkatkan efisiensi komputasi tanpa mengorbankan akurasi, menjadikannya pilihan populer untuk aplikasi pada perangkat dengan sumber daya terbatas.

CNN terinspirasi oleh cara kerja korteks visual di otak manusia, yang bertugas memproses informasi visual. Korteks visual mampu mengenali pola visual melalui hierarki yang dimulai dari pola sederhana seperti garis hingga pola kompleks seperti objek. Pendekatan ini diadaptasi dalam CNN melalui susunan lapisan konvolusi, *pooling*, dan lapisan *fully connected*, yang semuanya berkontribusi pada kemampuan model untuk memahami dan mengklasifikasikan data visual. Dalam konteks *Citrus Disease Detector*, CNN bertujuan untuk mengenali pola visual spesifik yang menunjukkan keberadaan penyakit pada tanaman jeruk. MobileNetV2 digunakan karena arsitekturnya yang ringan dan efisien, memungkinkan implementasi yang optimal meskipun pada perangkat dengan keterbatasan daya komputasi.

3.2 Komponen Penelitian

Penelitian deteksi penyakit pada buah jeruk menggunakan metode algoritma *Convolutional Neural Network* (CNN) dengan arsitektur MobileNetV2

memerlukan sejumlah komponen yang mendukung kelancaran dan efektivitas pelaksanaannya. Pemilihan komponen yang tepat menjadi sangat penting untuk memastikan hasil yang optimal. Dalam penelitian ini, komponen utama yang digunakan terdiri dari perangkat keras dan perangkat lunak. Pada aspek perangkat keras, penelitian ini menggunakan komputer atau server dengan spesifikasi tinggi yang dilengkapi dengan unit pemrosesan grafis (*GPU*). Komponen ini dipilih karena kemampuan GPU dalam menangani komputasi intensif yang dibutuhkan dalam pelatihan model CNN, terutama dengan arsitektur seperti MobileNetV2 yang dirancang untuk efisiensi tinggi. Selain itu, perangkat keras seperti kamera atau alat pengambilan gambar berkualitas tinggi juga digunakan untuk mendapatkan citra buah jeruk yang akurat. Setiap komponen dipilih berdasarkan kriteria spesifik, seperti kompatibilitas, kinerja, dan kemampuan mendukung kebutuhan penelitian. Kombinasi perangkat keras dan perangkat lunak yang sesuai memastikan penelitian dapat berjalan dengan optimal, menghasilkan model yang akurat dan efisien untuk mendeteksi penyakit pada buah jeruk. Tabel 3.1 merupakan Perangkat keras yang digunakan saat penelitian berlangsung.

Tabel 3.1 Spesifikasi Laptop Yang Digunakan

Spesifikasi Laptop	*Thosiba sattelite C840*
Processor	*Intel*(R) *Core*(TM) i3-238M CPU @ 2.20GHz
RAM	2.00 GB
ROM	HDD 1 TB
System	64-*Bit Operating System*

Tabel 3.1 spesifikasi laptop yang digunakan sebagai perangkat utama dalam melakukan penelitian ini, dimana perangkat laptop yang digunakan yaitu laptop *Thosiba sattelite C840* yang bertujun untuk menjalankan program yang akan dirancang menggunakan *google colab.* Tabel 3.2 merupakan perangkat lunak yang digunakan pada saat penelitian berlangsung.

Tabel 3.2 Perangkat Lunak

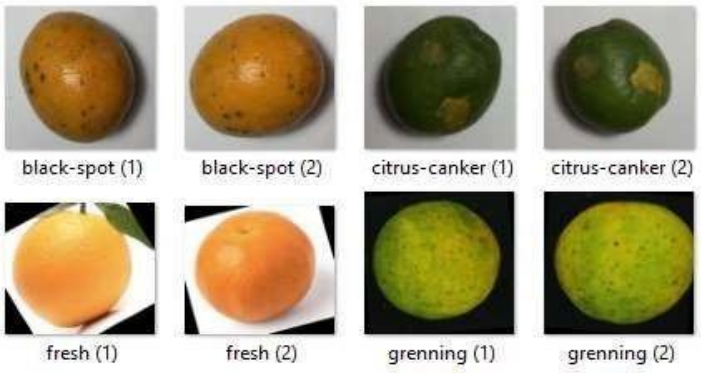
Google Colab	layanan berbasis *cloud* yang disediakan oleh Google untuk menjalankan kode *Python*. *Colab* mendukung berbagai pustaka *Deep learning* seperti *TensorFlow, Keras, dan PyTorch*, sehingga dapat dengan mudah membangun dan melatih model CNN. Dengan integrasi *Google Drive, google colab* digunakan untuk menyimpan *Dataset* dan model yang telah dilatih.
Command Prompt	*Command Prompt* digunakan untuk menjalankan berbagai perintah atau *input*. Sebagian besar perintah ini dirancang untuk mengotomatisasi tugas melalui skrip dan *file batch*, memungkinkan fungsi administratif seperti konfigurasi sistem, serta membantu dalam memecahkan masalah dan menyelesaikan beberapa jenis *error* yang umum terjadi pada sistem operasi *Windows.*
Pemrograman *phyton*	Bahasa Pemrograman Menjadi Jembatan Untuk Pengaplikasian *Deep Learning. Phyton* menjadi bahasa pemrograman yang memiliki Pendekatan yang fokus pada kesederhanaan dan kejelasan sintaksis Pada Bidang Kecerdasan Buatan. *phyton* sangat popular dilengkapi dengan pustaka-pustaka yang disediakan seperti *Numpy, Pandas, TensorFlow* dll. *TensorFlow* sering digunakan karena pustaka yang paling mudah dikembangkan dan memiliki komputasi tingkat tinggi. Selain itu, *TensorFlow* memberikan dukungan untuk berbagai jenis model pembelajaran termasuk *Convolutional Neural Network*

Tabel 3.2 di atas merupakan perangkat lunak yang digunakan dalam penelitian ini. Setiap perangkat lunak dipilih berdasarkan kemampuannya untuk mendukung proses pengembangan dan analisis data yang efisien. Penggunaan perangkat lunak ini secara sinergis mendukung kelancaran dan keberhasilan penelitian, memungkinkan penulis untuk mencapai hasil yang optimal dalam pengembangan model AI.

3.3 Pengumpulan Data

Proses pengumpulan data merupakan tahap penting dalam penelitian ini untuk memastikan keandalan dan validitas hasil. Data yang digunakan terdiri dari 2.400 sampel citra yang dibagi ke dalam dua kategori utama, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*), dengan proporsi pembagian 80% untuk pelatihan dan 20% untuk pengujian. Proporsi ini dipilih untuk memberikan keseimbangan yang optimal antara pelatihan model dan evaluasi akurasi prediksi.

Dataset yang digunakan dalam penelitian ini diperoleh dari Kaggle, sebuah platform yang terkenal sebagai komunitas global bagi data scientist dan peneliti. Kaggle menyediakan berbagai dataset berkualitas tinggi untuk mendukung analisis data dan pengembangan model berbasis pembelajaran mesin. Dalam konteks penelitian ini, dataset dari Kaggle mengandung informasi yang relevan untuk klasifikasi penyakit pada buah jeruk. Dataset tersebut mencakup beragam citra yang merepresentasikan kondisi buah jeruk, termasuk kategori normal dan yang terkena penyakit. Data ini diolah untuk mempersiapkan model yang mampu mendeteksi dan mengklasifikasi penyakit dengan efisien. Gambar 3.2 menunjukkan dataset yang digunakan dalam penelitian ini, memberikan visualisasi yang memperlihatkan keragaman citra yang diolah dalam proses pelatihan dan pengujian model.



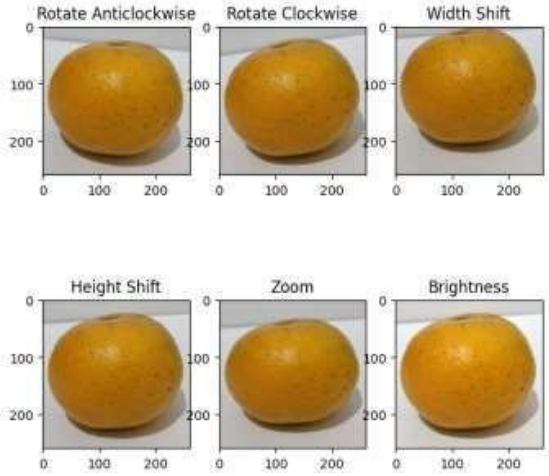
Gambar 3.2 Contoh *Dataset* Citrus

Gambar 3.2 menunjukkan contoh dataset yang digunakan dalam penelitian deteksi penyakit pada buah jeruk. Dataset ini terdiri dari citra yang dikelompokkan ke dalam empat kelas, yaitu *black spot*, *canker*, *greening*, dan *fresh*. Setiap kelas merepresentasikan kondisi spesifik buah jeruk, baik yang sehat (*fresh*) maupun yang terinfeksi penyakit tertentu. Dataset ini disiapkan untuk melatih model dalam

mengenali dan mengklasifikasi jenis penyakit dengan tingkat akurasi yang tinggi. Citra-citra ini menjadi dasar dalam proses pelatihan dan pengujian model berbasis algoritma *Convolutional Neural Networ*k (CNN).

3.4 Pra-pemrosesan data

Tahap pertama dalam pemrosesan data adalah proses augmentasi data. Selain itu, langkah-langkah pra-pemrosesan data juga dilakukan untuk membersihkan dan mempersiapkan data sebelum digunakan dalam analisis atau pemodelan. Gambar 3.3 merupakan contoh augmentasi data.



Gambar 3.3 Augmentasi Data

Gambar 3.3 merupakan proses augmentasi data pada penelitian ini yang bertujuan untuk meningkatkan variasi data, mengurangi overfitting, meningkatkan robustitas model, mengatasi data tidak seimbang, dan mempercepat proses pelatihan. Augmentasi membantu menyediakan lebih banyak contoh untuk pelatihan dan membuat model lebih tahan terhadap gangguan.

Setelah tahap augmentasi data selesai dilakukan, tahap selanjutnya adalah proses load *Dataset*. Pada tahap ini dilakukan proses input *Dataset* serta memisahkan file-filenya, setelah itu dilakukan proses penentuan direktori pelatihan dan pengujian pada dokumen *Dataset* dengan cara labelling dan splitting *Dataset*. Splitting *Dataset* dilakukan dengan proporsi 80:20 dimana 80% untuk data latih dan 20% untuk data uji.. Ukuran dimensi data yang digunakan adalah 224 x 224 dan *batch size* yang digunakan adalah 16.

3.5 *Modeling*

Modelling Citrus yang menggunakan model *MobilenetV2.* Penggunaan model ini memiliki akurasi terbaik dan cocok untuk digunakan dalam *image classification citru*s. *MobileNetV2* adalah model arsitektur CNN yang dikembangkan oleh google. *MobileNetV2* memprioritaskan ukuran dan kebutuhan komputasi yang kecil namun tetap mempertahankan nilai akurasi. *MobileNetV2* sangat tepat digunakan pada perangkat yang memiliki sumber daya yang terbatas.

Penggunaan model *MobilenetV2* Tahap persiapan *Mobilenet*V2 untuk transfer learning Menggunakan base model *Mobilenet*V2 dengan arsitektur sebagai berikut:

1. *Global Average Pooling* 2D untuk mengoperasikan *pooling*
2. *Layers Dense* dengan ukuran file 128 *layers* dan dengan aktivasi relu
3. untuk mencegah adanya overfitting maka ditambahkan Dropout 0.2
4. *Layer Dense* 1 dengan ukuran file 64 layers dan dilengkapi aktivasi relu
5. Kemudian dilakukan *compile* model dengan menggunakan *loss binary crossentropy*, *optimizer adam,* dan *metrics accuracy.*

3.6 Evaluasi

Evaluasi model ini sangat penting untuk dilakukan guna mengukur seberapa baik kinerja yang tercapai. model yang telah dirancang dalam menyelesaikan masalah tertentu. Dalam penelitian ini, evaluasi dilakukan dengan menggunakan beberapa metrics, seperti precision, recall, F1 score, dan accuracy, serta confusion matrix. Precision mengukur seberapa banyak prediksi positif yang benar dari semua prediksi positif yang dilakukan, sementara recall mengukur seberapa banyak prediksi positif yang benar dari semua data positif yang seharusnya terdeteksi. F1 score adalah rata-rata harmonik dari precision dan recall, yang menggambarkan keseimbangan antara keduanya. Accuracy mengukur seberapa banyak prediksi yang benar dibandingkan dengan total prediksi yang dibuat. Confusion matrix digunakan untuk memberikan gambaran lebih mendalam tentang kinerja model dengan menampilkan jumlah prediksi yang benar dan salah dalam kategori positif dan negatif.

Melalui evaluasi ini, kita dapat mendeteksi masalah seperti overfitting, di mana model terlalu baik pada data pelatihan namun gagal pada data baru, dan underfitting, di mana model gagal menangkap pola yang ada dalam data. Evaluasi juga memberikan umpan balik untuk perbaikan model lebih lanjut. Selain itu, hasil evaluasi membantu dalam pengambilan keputusan yang informatif mengenai model mana yang paling sesuai untuk aplikasi tertentu, sehingga dapat digunakan secara optimal dalam konteks dunia nyata.

3.7 *Deployment*

Deployment bertujuan untuk menghubungkan teknologi dengan kebutuhan pengguna dalam mengakses web deteksi penyakit buah jeruk secara efisien. Pada penelitian ini, deployment dilakukan di local host, yang berarti web dijalankan pada komputer lokal. Untuk mengaksesnya, pengguna perlu menggunakan command prompt untuk memulai server dan mengakses aplikasi melalui browser. Metode ini memungkinkan pengujian dan pengembangan web secara lebih mudah sebelum dipublikasikan secara lebih luas. Dengan deployment di local host, pengguna dapat menguji fungsionalitas dan performa web dalam lingkungan yang terkendali dan lebih terfokus pada perbaikan serta pengoptimalan.

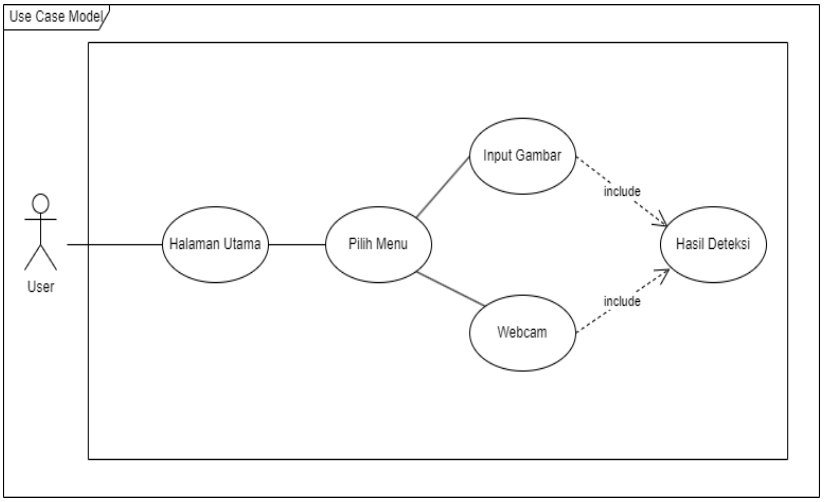
3.8 Perancangan Sistem

Perancangan sistem adalah proses merancang dan mengembangkan struktur, komponen, dan interaksi dalam sebuah sistem untuk memenuhi kebutuhan tertentu. Perancangan sistem yang baik membantu dalam memahami dan mendefinisikan kebutuhan pengguna serta tujuan dari model yang akan dibangun.

Perancangan sistem dilakukan dengan untuk mengembangankan arsitektur yang efisien, memastikan bahwa model dapat diimplementasikan dengan optimal dalam infrastruktur yang tersedia.

3.9.1 *Use Case* Diagram

Use case diagram adalah representasi visual yang menggambarkan interaksi antara aktor atau pengguna dengan sistem yang sedang dikembangkan. Diagram ini merupakan bagian dari Unified Modeling Language (UML) dan digunakan untuk menggambarkan fungsionalitas sistem serta cara aktor berinteraksi dengan berbagai fitur dalam sistem aplikasi web. Gambar 3.4 Merupakan *use case* diagram pada penelitian ini.

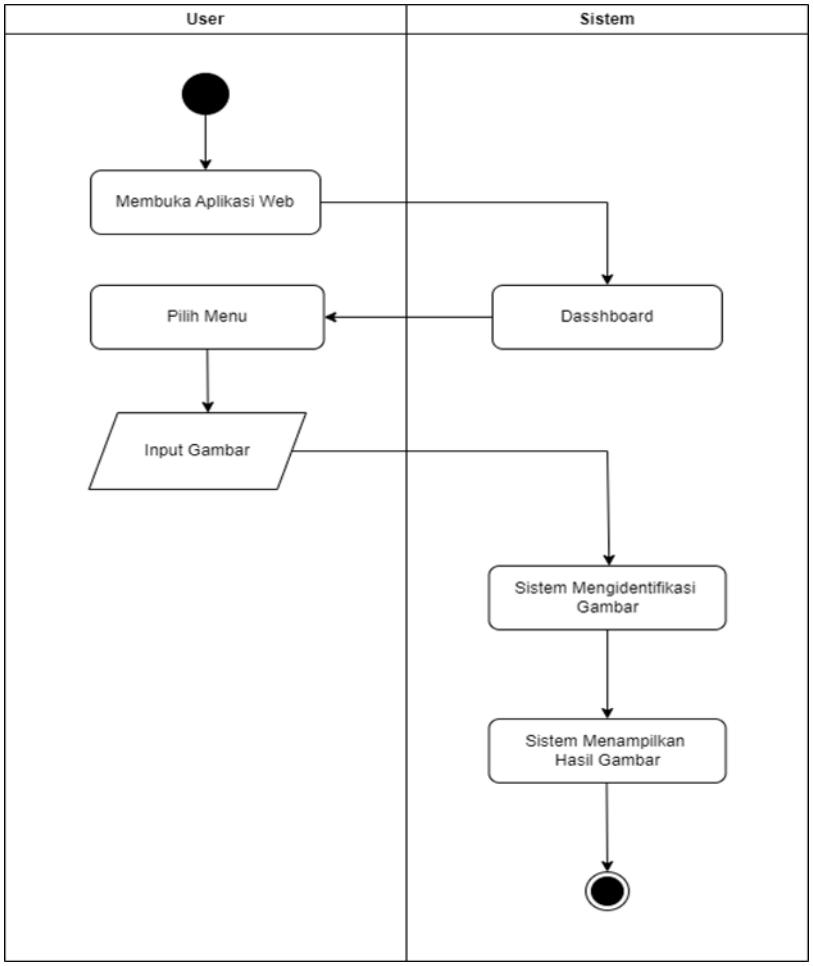


Gambar 3.4 *Use Case Diagram*

Gambar 3.4 diatas merupakan *use case diagram* deteksi penyakit buah jeruk. Diagram diatas menggambarkan interaksi antar user dan sistem aplikasi, dimana *user* dapat menggunakan fitur input gambar atau webcam lalu akan mendapatkan hasil deteksi penyakit pada buah jeruk.

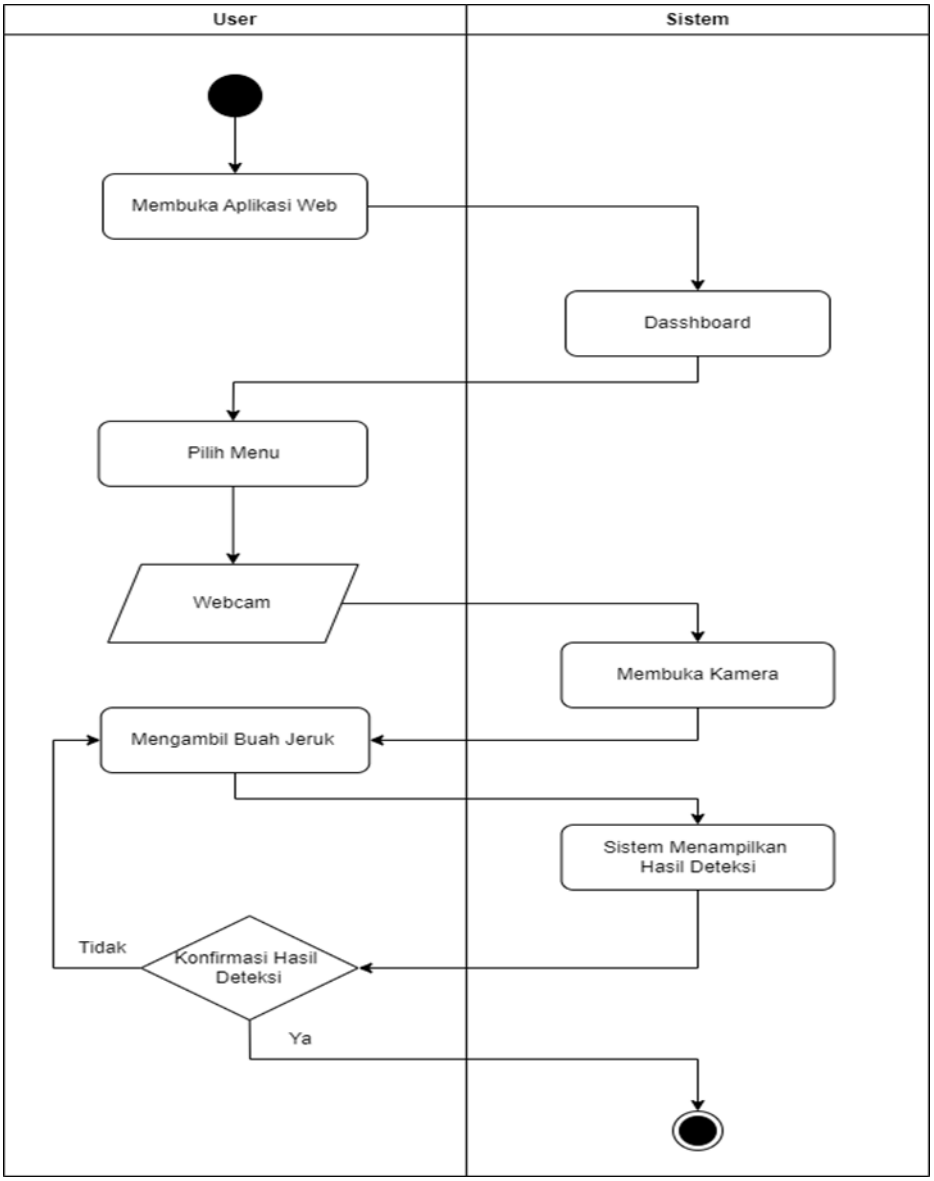
3.9.2 *Activity Diagram*

Activity diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja atau proses dalam suatu sistem. Diagram ini menggambarkan urutan langkah-langkah dalam sebuah aktivitas dan bagaimana aktivitas-aktivitas tersebut saling berinteraksi satu sama lain. Gambar 3.5 merupakan *activity diagram* input gambar pada penelitian ini.



Gambar 3.5 *Activity Diagram Input* Gambar

Gambar 3.5 merupakan *activity diagram* input gambar pada deteksi penyakit buah jeruk. Pada diagram tersebut menggambarkan proses *user* yaitu membuka aplikasi web, memilih fitur menu lalu input gambar yang ingin di deteksi. Sedangkan proses pada system yaitu ketika user menginputkan gambar maka sistem akan mengidentifikasi gambar tersebut lalu sistem akan menampilkan hasil deteksi gambar yang telah diinputkan oleh *user.* Pada Gambar 3.6 merupakan *activity diagram* pada fitur *webcam.*



Gambar 3.6 *Activity Diagram Webcam*

Gambar 3.6 diatas menggambarkan bagaimana alur kerja antara *user* dan sistem pada saat menggunakan fitur *webcam* untuk mendeteksi penyakit buah jeruk secara *real time.*