

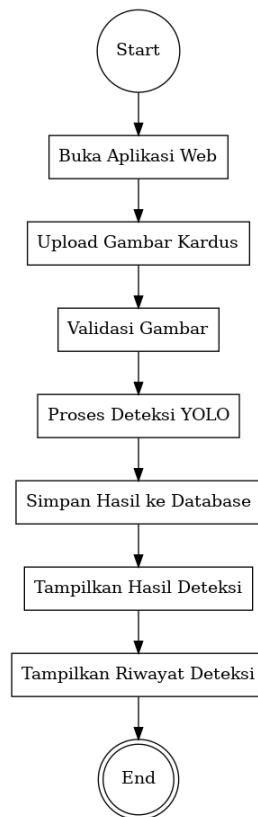
BAB III

ANALISIS DAN PERANCANGAN

Penelitian ini termasuk dalam kategori **penelitian rekayasa (*engineering research*)**, yaitu jenis penelitian yang berfokus pada perancangan, pembangunan, dan pengujian suatu sistem sebagai solusi terhadap permasalahan tertentu. Dalam konteks ini, penelitian tidak dilakukan secara langsung ke lapangan untuk mengumpulkan data primer, tetapi lebih menekankan pada proses pengembangan perangkat lunak secara sistematis berdasarkan kebutuhan fungsional yang telah dianalisis. Sistem yang dibangun menggunakan pendekatan simulasi, di mana data pelatihan model diperoleh dari dataset publik melalui platform Roboflow, serta pengujian dilakukan pada lingkungan lokal secara modular. Dengan demikian, penelitian ini menghasilkan sebuah prototipe aplikasi berbasis web yang dapat dijadikan dasar untuk implementasi lebih lanjut pada situasi nyata.

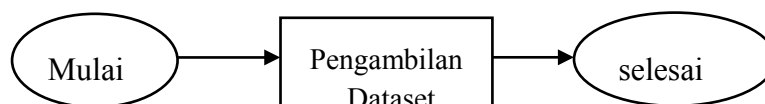
1.1 Analisa Dan Perancangan

Flowchart merupakan gambaran proses berjalannya alur dari sistem pendeteksian kerusakan kardus menggunakan algoritma YOLOv8. Berikut adalah format flowchart utama sistem yang dikembangkan:



Gambar 3.1 Perancangan Sistem

1.1.1 Pengambilan Dataset

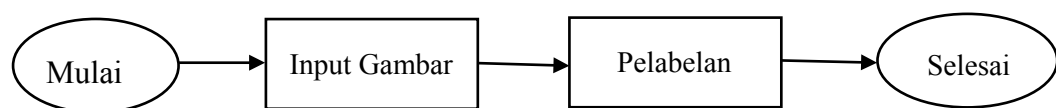


Gambar 3.2 Pengambilan Dataset

Proses pengambilan dataset dilakukan dalam bentuk foto menggunakan kamera smartphone atau laptop. Dataset ini harus mencakup berbagai kondisi

pencahayaannya, sudut pengambilan gambar, dan jenis kardus untuk memastikan model yang dilatih mampu mendeteksi berbagai macam kerusakan. Pengambilan gambar dilakukan secara sistematis dengan mengkategorikan tingkat kerusakan, seperti robek, penyok, atau berlubang. Selain itu, jumlah dataset yang diambil harus mencukupi agar model dapat dilatih dengan variasi data yang luas, sehingga meningkatkan akurasi dalam mendeteksi kerusakan pada kardus yang berbeda-beda.

1.1.2 Pelabelan Dataset



Gambar 3.3 Pelabelan Dataset

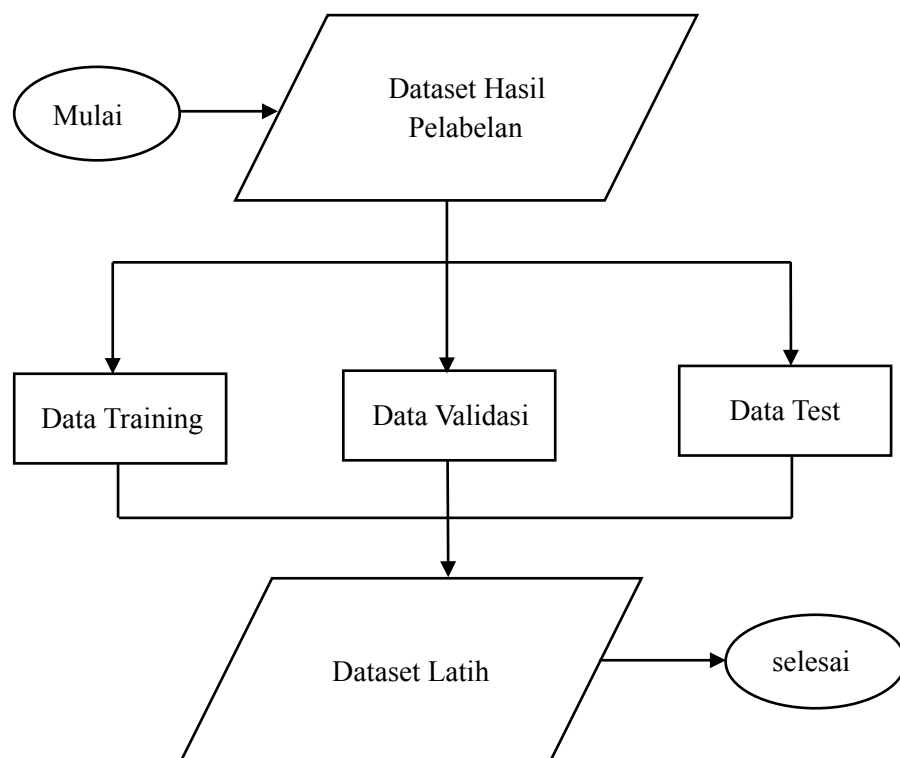
Dataset yang telah dikumpulkan akan melalui proses pelabelan menggunakan Roboflow. Pelabelan bertujuan untuk memberikan bounding box pada objek yang terdapat dalam gambar, sehingga dapat digunakan untuk melatih model deteksi.

1.1.3 Pembagian Dataset

Dataset yang telah dilabeli akan dibagi menjadi tiga bagian:

1. **Data Training:** Digunakan untuk melatih model dalam mendeteksi kerusakan.

2. **Data Validasi:** Digunakan untuk mengevaluasi model selama proses pelatihan.
3. **Data Testing:** Digunakan untuk menguji performa akhir model dan mendapatkan akurasi deteksi.



Gambar 3.4 Pembagian Dataset

1.1.4 Training Dataset

Pada tahap ini, proses pelatihan model deteksi kerusakan kardus dilakukan menggunakan dataset yang telah tersedia dan dipublikasikan melalui platform **Roboflow**. Dataset tersebut sudah dilengkapi dengan anotasi (*bounding box* dan label) yang menandai berbagai jenis kerusakan pada kardus seperti penyok, sobekan, maupun lipatan

Dataset diambil langsung dari proyek Roboflow yang telah disediakan sebelumnya oleh pihak ketiga. Penulis tidak melakukan proses anotasi secara manual, melainkan memanfaatkan dataset yang sudah teranotasi dengan baik dan siap digunakan untuk keperluan pelatihan model. Roboflow menyediakan fitur **Export Dataset**, di mana dataset dapat diakses langsung melalui **link API**. Format yang digunakan disesuaikan dengan kebutuhan model YOLOv8.

Dataset tersebut diakses melalui **API Roboflow** menggunakan skrip Python yang dijalankan di **Visual Studio Code**. Proses ini memudahkan integrasi antara data dan model tanpa perlu mengunduh file secara manual. Dataset sudah dibagi secara otomatis oleh Roboflow ke dalam:

1. **Training Set (data latih)**
2. **Validation Set (data validasi)**
3. **Test Set (data uji)**

Pembagian ini sangat penting untuk menjaga kualitas dan evaluasi model secara objektif.

Model yang digunakan untuk pelatihan adalah **YOLOv8** (*You Only Look Once versi 8*), yaitu model deteksi objek berbasis deep learning yang cepat dan efisien. Pelatihan dilakukan dengan mengatur beberapa parameter penting, antara lain:

1. **Epoch**: jumlah iterasi pelatihan,
2. **Batch size**: jumlah data dalam setiap batch,
3. **Image size**: resolusi gambar yang diproses,

4. **Learning rate:** tingkat pembelajaran model.

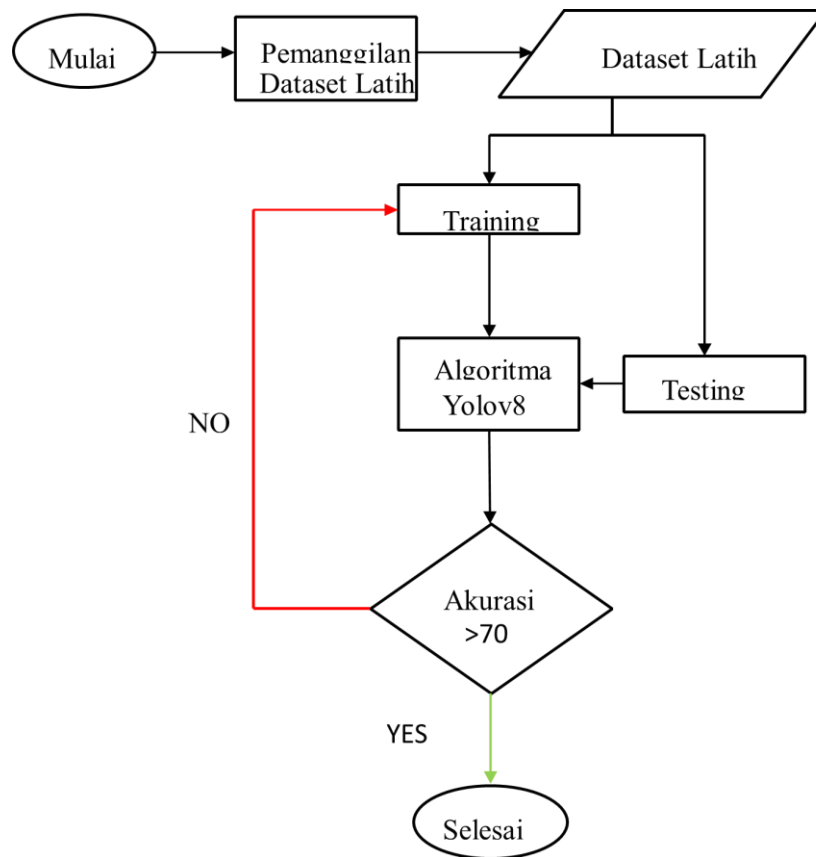
Model akan mempelajari pola dari objek-objek kerusakan kardus berdasarkan dataset yang tersedia.

Selama pelatihan berlangsung, dilakukan pemantauan terhadap metrik performa seperti:

1. **Training loss dan validation loss**
2. **Precision dan recall**
3. **mAP (mean Average Precision)**

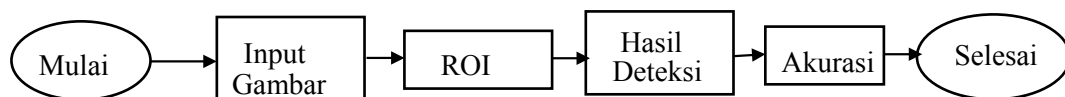
Jika model belum mencapai akurasi yang memuaskan, maka proses pelatihan dapat diulang dengan menyesuaikan parameter atau menambah variasi data (augmentasi). Proses pengulangan ini ditunjukkan oleh panah merah dalam diagram alur.

Setelah model mencapai akurasi yang diinginkan, proses pelatihan akan menghasilkan file output berupa **model terlatih (.pt)**, biasanya dengan nama `best.pt`. File ini kemudian digunakan dalam aplikasi pendeteksi kerusakan kardus berbasis web yang telah dikembangkan.



Gambar 3.5 Proses Training Dan Testing

1.1.5 Deteksi



Gambar 3.6 Proses Deteksi

Proses deteksi diawali dengan memasukkan gambar yang telah tersedia ke dalam sistem. Setelah itu, gambar tersebut dipanggil untuk dilakukan pengujian menggunakan pustaka (library) dari YOLOv8. Model kemudian diuji dengan memanfaatkan algoritma YOLOv8 terhadap gambar-gambar yang sebelumnya sudah dilabeli menggunakan bounding box. Hasil dari pengujian ini akan menampilkan nilai akurasi serta ROI (Region of Interest) yang menunjukkan area spesifik dari deteksi yang dilakukan.

1.2 Unified Modeling Language (UML)

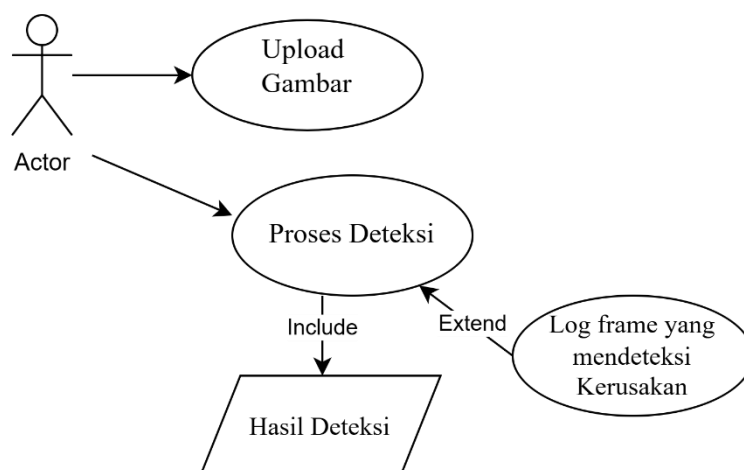
Perancangan sistem menggunakan beberapa diagram UML untuk memodelkan interaksi pengguna dan aliran kerja dalam aplikasi. Diagram UML membantu dalam mendokumentasikan proses bisnis, hubungan antar entitas, serta alur komunikasi dalam sistem. Beberapa diagram yang digunakan antara lain:

1. **Use Case Diagram:** Menggambarkan skenario interaksi antara pengguna dan sistem.
2. **Activity Diagram:** Menunjukkan alur kerja dari proses yang dilakukan dalam sistem.
3. **Sequence Diagram:** Memvisualisasikan urutan interaksi antar objek dalam sistem.
4. **Class Diagram:** Mengilustrasikan struktur data dan hubungan antar kelas dalam sistem.

Dengan menggunakan UML, proses perancangan dapat lebih terstruktur dan mudah dipahami oleh seluruh tim pengembang

1.2.1 Use Case Diagram

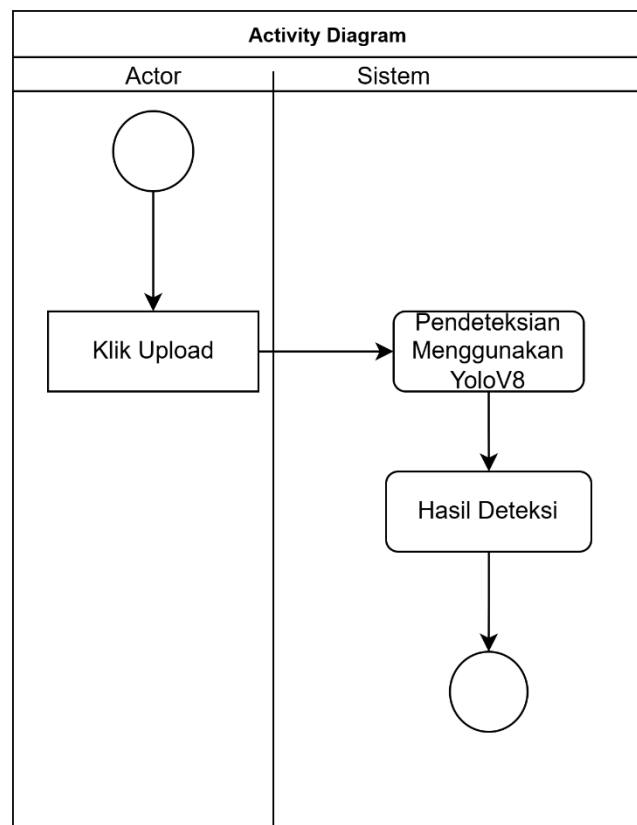
Use Case Diagram menggambarkan bagaimana pengguna berinteraksi dengan sistem dalam mendeteksi kerusakan kardus. Diagram ini menunjukkan berbagai aktor yang berperan, seperti pengguna dan sistem deteksi berbasis YOLOv8. Setiap interaksi menggambarkan fungsi utama, seperti unggah gambar, pemrosesan citra, analisis model, dan tampilan hasil. Dengan adanya Use Case Diagram, pengembang dapat memahami lebih jelas skenario penggunaan sistem dan memastikan bahwa setiap fungsi berjalan sesuai kebutuhan pengguna. Diagram ini juga membantu dalam proses pengembangan agar sesuai dengan kebutuhan industri logistik.



Gambar 3.7 Use Case Diagram

1.2.2 Activity Diagram

Activity diagram merupakan suatu proses yang menjelaskan alur kerja suatu sistem yang dibuat, bertujuan untuk menjelaskan lebih lanjut proses skenario use case.



Gambar 3.8 Activity Diagram

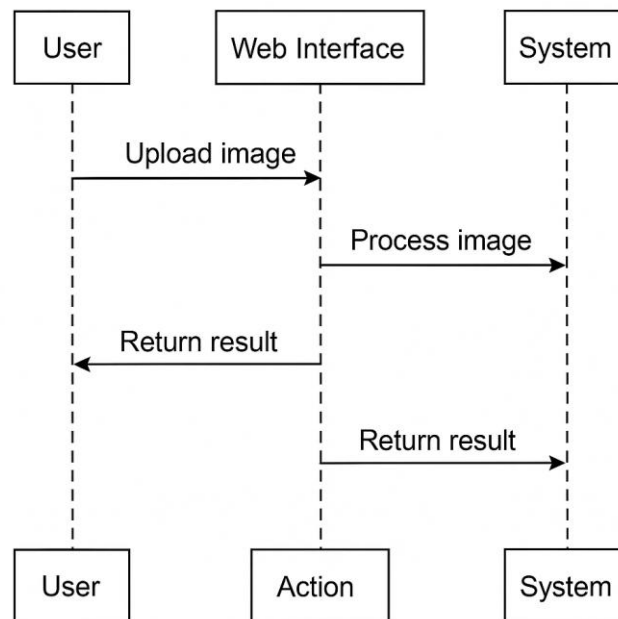
1.2.3 Sequence Diagram

Sequence Diagram menggambarkan interaksi antar objek dalam sistem berdasarkan urutan waktu. Dalam konteks aplikasi deteksi kerusakan kardus berbasis AI, Sequence Diagram menunjukkan bagaimana pengguna berinteraksi dengan sistem mulai dari mengunggah gambar hingga mendapatkan hasil analisis.

Urutan Proses dalam Sequence Diagram:

1. **Pengguna** mengunggah gambar kardus yang akan diperiksa.
2. **Sistem** menerima dan memproses gambar menggunakan model AI berbasis *YOLOv8*.
3. **Model AI** menganalisis gambar dan mengidentifikasi berbagai jenis kerusakan seperti sobekan, penyok, atau lubang.
4. **Sistem** menampilkan hasil analisis kepada pengguna, termasuk jenis dan tingkat kerusakan yang terdeteksi.

Sequence Diagram ini membantu dalam memahami bagaimana data mengalir dalam sistem dan bagaimana setiap komponen berinteraksi untuk menghasilkan output yang diinginkan.



Gambar 3.9 Sequence Diagram

1.2.4 Class Diagram

Class Diagram digunakan untuk memodelkan struktur sistem dengan mendefinisikan kelas-kelas utama serta hubungan di antaranya. Diagram ini memberikan gambaran tentang bagaimana data dan fungsionalitas diorganisir dalam sistem deteksi kerusakan kardus.

Komponen dalam Class Diagram:

Class Gambar – Merepresentasikan data gambar yang diunggah oleh pengguna.

Atribut:

1. id_gambar: Identitas unik setiap gambar.
2. nama_file: Nama file gambar.
3. path_file: Lokasi penyimpanan file gambar di server.
4. tanggal_upload: Tanggal gambar diunggah ke sistem.

Metode:

1. Upload(): Berfungsi untuk mengunggah gambar ke sistem.
2. Hapus(): Berfungsi untuk menghapus gambar dari sistem.

Class Detector – Merepresentasikan model AI yang digunakan untuk mendeteksi objek pada gambar.

Atribut:

1. id_detector: Identitas unik detektor.
2. nama_model: Nama model AI yang digunakan (misalnya YOLO).
3. versi: Versi model deteksi.

Metode:

prosesDeteksi(gambar): Berfungsi untuk memproses gambar dan menghasilkan hasil deteksi.

Class Hasil Deteksi – Menyimpan informasi hasil pendeteksian dari gambar yang diproses oleh model detektor.

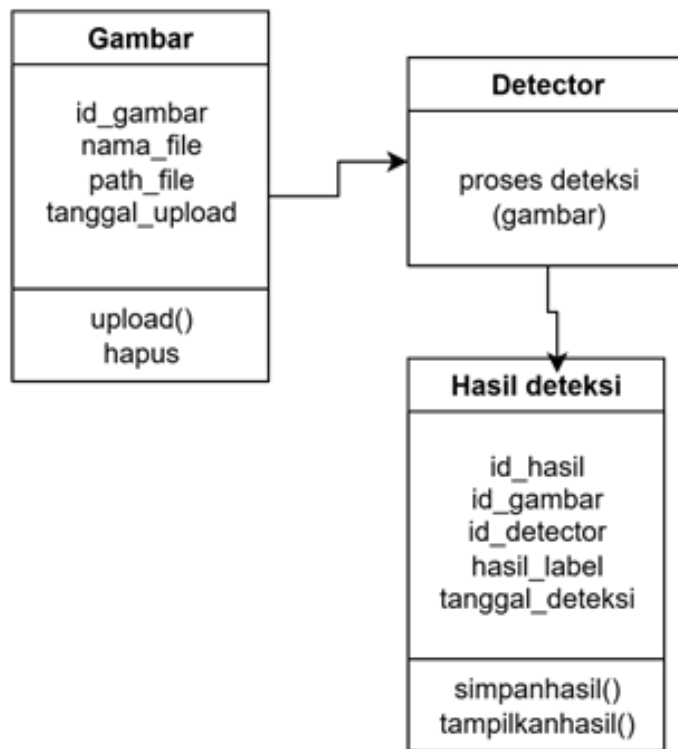
Atribut:

1. id_hasil: Identitas unik hasil deteksi.
2. id_gambar: ID gambar yang dianalisis.
3. id_detektor: ID detektor yang digunakan.
4. hasil_label: Label hasil deteksi (misalnya: “rusak”, “bagus”).
5. confidence_score: Tingkat kepercayaan model terhadap hasil deteksi.
6. tanggal_deteksi: Tanggal hasil deteksi dihasilkan.

Metode:

1. Upload(): Berfungsi untuk mengunggah gambar ke sistem.
2. Hapus(): Berfungsi untuk menghapus gambar dari sistem

Class Diagram ini membantu dalam memahami hubungan antara objek dalam sistem dan bagaimana data dipertukarkan di antara mereka.



Gambar 3.10 Class Diagram

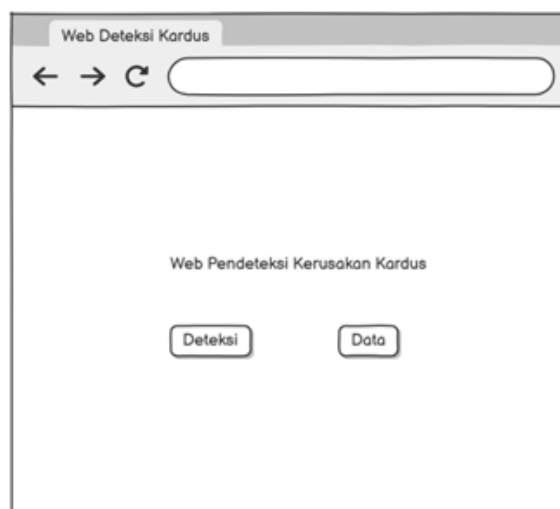
1.3 Desain Interface

Desain antarmuka sistem dibuat berbasis web dengan prinsip user-friendly untuk meningkatkan kemudahan akses bagi pengguna. Komponen utama yang disediakan:

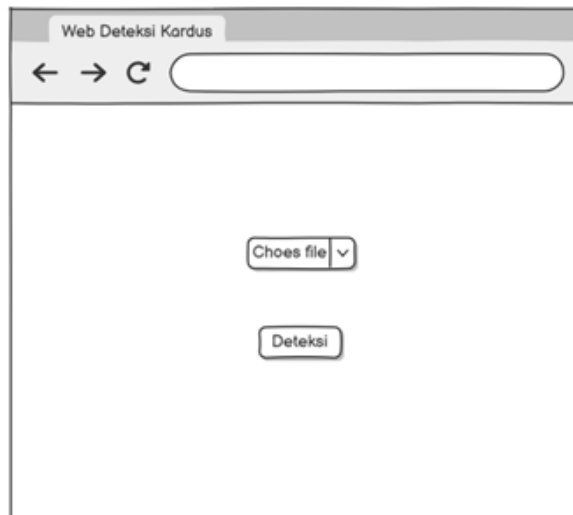
1. **Form unggah gambar** untuk mendeteksi kerusakan kardus. Pengguna dapat memilih file dari perangkat mereka dan mengunggahnya dengan mudah.

2. **Tombol Submit** untuk memulai proses deteksi. Setelah tombol ditekan, sistem akan mengirimkan gambar ke server untuk diproses oleh model deteksi.
3. **Area hasil deteksi** yang menampilkan gambar hasil analisis YOLOv8. Pada area ini, pengguna dapat melihat informasi seperti jenis kerusakan yang terdeteksi, persentase akurasi, dan visualisasi *bounding box* yang menunjukkan area kerusakan pada kardus.

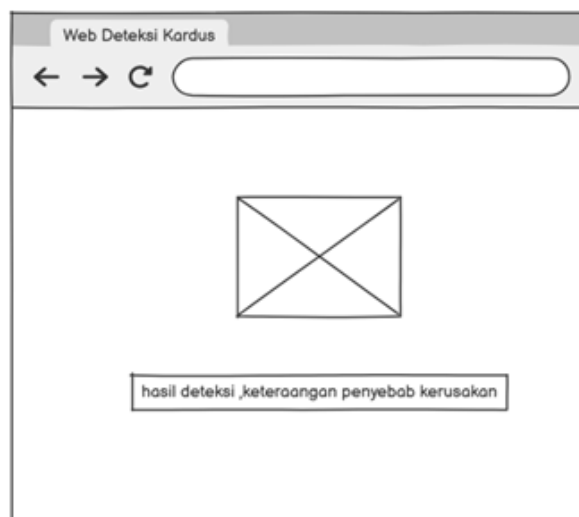
Dengan rancangan ini, sistem diharapkan mampu memberikan hasil deteksi kerusakan kardus secara akurat dan efisien. Selain itu, antarmuka yang responsif dan intuitif akan membantu pengguna dari berbagai latar belakang untuk memahami dan menggunakan sistem dengan mudah. Tampilan antarmuka yang bersih dan modern juga akan memberikan pengalaman yang lebih baik dalam penggunaan sistem deteksi kerusakan kardus berbasis AI ini.



Gambar 3.11 Dasbord



Gambar 3.12 Form Unggah Gambar



Gambar 3.13 Hasil Deteksi