

BAB II

LANDASAN TEORI

1.1 Konsep Dasar Pendeteksi Kerusakan

Pendeteksian kerusakan adalah proses identifikasi dan analisis kondisi suatu objek untuk menentukan apakah terjadi degradasi atau cacat yang dapat mempengaruhi fungsi dan kualitasnya. Dalam industri manufaktur dan logistik, deteksi kerusakan sangat penting untuk memastikan kualitas produk tetap terjaga sebelum dikirim ke konsumen[1].

Metode pendeteksian kerusakan berkembang seiring dengan kemajuan teknologi, dari metode manual hingga pemanfaatan teknologi berbasis kecerdasan buatan (Artificial Intelligence/AI). Dengan AI, sistem dapat secara otomatis mendeteksi kerusakan, Meminimalkan kemungkinan terjadinya kesalahan oleh manusia, sekaligus mempercepat proses inspeksi dan meningkatkan efisiensinya. AI mampu mengklasifikasikan dan menganalisis pola kerusakan dengan cepat menggunakan data citra yang dikumpulkan melalui sensor atau kamera.

Selain itu, dalam industri pengemasan, penggunaan AI untuk mendeteksi kerusakan kardus menjadi solusi inovatif untuk meningkatkan kualitas produk sebelum didistribusikan. Dengan metode seperti computer vision dan deep learning, sistem dapat mengenali kerusakan sekecil apapun dengan tingkat akurasi tinggi. Hal ini tidak hanya meningkatkan efektivitas produksi, tetapi juga mengurangi risiko pengiriman barang dalam kondisi rusak kepada pelanggan.

Pada praktiknya, pendeteksian kerusakan memiliki beberapa pendekatan yang digunakan, antara lain:

1. Pendeteksian visual manual: Dilakukan oleh tenaga kerja manusia dengan mengevaluasi kondisi kardus secara langsung. Metode ini memiliki keterbatasan karena tergantung pada pengalaman dan ketelitian pengamat.
2. Pendeteksian berbasis sensor: Menggunakan sensor optik atau termal untuk mendeteksi perubahan warna atau deformasi yang menandakan adanya kerusakan.
3. Pendeteksian berbasis AI: Memanfaatkan teknologi pembelajaran mesin untuk menganalisis dan mengklasifikasikan citra secara otomatis.

1.1.1 Jenis-jenis Kerusakan Kardus

Kardus merupakan material utama dalam pengemasan yang sering mengalami berbagai jenis kerusakan selama proses distribusi[2]. Berikut adalah beberapa jenis kerusakan kardus yang umum terjadi:

1. Sobekan: Biasanya terjadi akibat gesekan dengan benda lain atau akibat perlakuan yang tidak hati-hati selama transportasi. Sobekan dapat mengurangi daya tahan kardus dan meningkatkan risiko kerusakan pada barang di dalamnya.
2. Penyok: Disebabkan oleh tekanan dari tumpukan barang yang terlalu berat atau benturan selama pengiriman. Kardus yang penyok dapat mengakibatkan barang di dalamnya mengalami kerusakan struktural.

3. Lubang: Dapat terjadi akibat benda tajam yang menembus permukaan kardus. Lubang pada kardus berpotensi menyebabkan kebocoran isi dan menurunkan keamanan produk.
4. Kerusakan akibat kelembaban: Kardus yang terkena air atau kelembaban tinggi akan kehilangan kekuatannya dan menjadi rapuh. Hal ini sering terjadi pada proses pengiriman di lingkungan dengan tingkat kelembaban tinggi atau akibat penyimpanan yang tidak memadai.
5. Kerusakan struktural: Kardus yang mengalami deformasi akibat tekanan berlebih atau proses penyimpanan yang tidak tepat. Struktur kardus yang melemah dapat menyebabkan keruntuhan saat disusun dalam tumpukan tinggi.
6. Kerusakan akibat infestasi serangga atau hama: Kardus yang disimpan dalam waktu lama dapat menjadi sarang serangga atau hama, yang dapat merusak material kemasan.

Untuk mengatasi permasalahan ini, teknologi AI dapat diterapkan dalam sistem deteksi kerusakan kardus dengan menggunakan model klasifikasi berbasis gambar yang memungkinkan pengenalan otomatis terhadap berbagai jenis kerusakan tersebut.

1.2 Artificial Intelligence

1.2.1 Pengertian Artificial Intelligence

Artificial Intelligence (AI) atau Kecerdasan Buatan adalah cabang dari ilmu komputer yang bertujuan untuk menciptakan sistem atau mesin yang memiliki kemampuan untuk melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Tugas-tugas tersebut meliputi pemecahan masalah, pengambilan keputusan, pengenalan pola, pemrosesan bahasa alami, serta kemampuan belajar dan beradaptasi dengan pengalaman yang diperoleh. Dengan kata lain, AI berusaha menciptakan mesin yang dapat meniru atau bahkan melebihi kemampuan kognitif manusia dalam sejumlah bidang tertentu[3].

AI dapat dibagi menjadi dua kategori utama:

1. **AI Sempit (Narrow AI):** Merupakan bentuk AI yang dirancang untuk menyelesaikan satu tugas spesifik, seperti sistem rekomendasi di platform e-commerce, pengenalan suara pada asisten digital, atau sistem deteksi wajah. AI sempit ini lebih banyak diterapkan dalam kehidupan sehari-hari dan terbatas pada konteks tertentu.
2. **AI Umum (General AI):** Jenis AI yang lebih canggih dan dapat mengerjakan berbagai tugas seperti manusia, dengan kemampuan untuk berpikir dan menyelesaikan masalah dalam berbagai situasi yang berbeda. Namun, hingga kini, AI umum masih menjadi konsep teori dan belum sepenuhnya terwujud dalam praktik.

Salah satu komponen utama AI adalah penggunaan algoritma pembelajaran mesin (machine learning) yang memungkinkan sistem untuk belajar dan mengidentifikasi pola dari data yang besar. Proses ini dapat dilengkapi dengan teknologi deep learning, yang menggunakan jaringan saraf tiruan untuk menyelesaikan tugas yang lebih kompleks. Misalnya, dalam pengolahan gambar, deep learning digunakan untuk mengidentifikasi objek dalam gambar dengan akurasi tinggi.

Dalam konteks pendeteksian kerusakan kardus, penerapan AI sangat relevan dan bermanfaat. AI dapat digunakan untuk menganalisis citra atau gambar kardus yang diambil menggunakan kamera atau sensor lainnya. Dengan algoritma pemrosesan citra, AI dapat mengenali berbagai jenis kerusakan, seperti sobekan, penyok, atau lubang pada kardus. Pendeteksian otomatis ini tidak hanya meningkatkan efisiensi dalam inspeksi produk, tetapi juga memastikan tingkat akurasi yang tinggi dalam mendeteksi kerusakan yang mungkin terlewatkan oleh manusia.

Berikut adalah beberapa contoh penerapan AI dalam pendeteksian kerusakan kardus:

1. **Pengenalan Pola Kerusakan:** AI dapat dilatih untuk mengenali pola atau karakteristik tertentu pada permukaan kardus yang menunjukkan adanya kerusakan. Dengan mengumpulkan data citra kerusakan yang berbeda, model AI dapat membedakan antara kardus yang masih dalam kondisi baik dan yang sudah rusak.

2. **Pemrosesan Gambar dan Analisis Citra:** Menggunakan teknik seperti *image segmentation* dan *edge detection*, AI dapat mengidentifikasi dengan tepat bagian mana dari kardus yang mengalami kerusakan. Proses ini melibatkan analisis intensif terhadap piksel-piksel gambar untuk mendeteksi anomali.
3. **Klasifikasi Kerusakan:** Setelah mendeteksi kerusakan, AI dapat mengklasifikasikan jenis kerusakan yang terjadi, seperti sobekan kecil, penyok, atau lubang besar. Setiap jenis kerusakan ini mungkin memerlukan penanganan yang berbeda, sehingga pengklasifikasian yang tepat sangat penting.
4. **Automatisasi Laporan:** Setelah proses analisis selesai, AI dapat secara otomatis menghasilkan laporan mengenai status kondisi kardus, jenis kerusakan yang ditemukan, dan rekomendasi tindak lanjut. Laporan ini dapat dikirim melalui platform berbasis web, memungkinkan pemantauan real-time dari berbagai lokasi.

Keuntungan utama penerapan AI dalam pendeteksian kerusakan kardus adalah kemampuannya untuk melakukan pemeriksaan secara lebih cepat dan dengan akurasi yang lebih tinggi dibandingkan dengan pemeriksaan manual. Selain itu, AI juga dapat beradaptasi dan terus belajar dari data baru, meningkatkan kemampuannya seiring berjalannya waktu. Dalam industri pengemasan dan logistik, ini dapat mengurangi waktu inspeksi, meningkatkan

kualitas kontrol, serta mengurangi biaya operasional yang terkait dengan kerusakan produk.

Dalam konteks pendeteksian kerusakan kardus, AI dapat digunakan untuk menganalisis gambar kardus, mengenali pola kerusakan seperti sobekan, penyok, atau lubang, serta memberikan laporan secara otomatis melalui platform berbasis web.

1.3 Machine Learning dalam Deteksi Kerusakan

Machine Learning (ML) adalah sub-bidang AI yang memungkinkan komputer belajar dari data tanpa pemrograman eksplisit. Dalam pendeteksian kerusakan kardus, ML digunakan untuk menganalisis gambar dan mengenali pola kerusakan dengan tingkat akurasi tinggi[4].

Proses kerja ML dalam deteksi kerusakan mencakup beberapa tahapan penting:

1. Pengumpulan dan Preprocessing Data: Mengumpulkan gambar kardus dalam berbagai kondisi, baik yang masih baik maupun yang rusak. Data ini kemudian diproses dengan teknik seperti resizing, normalisasi warna, augmentasi data, serta penghapusan noise untuk meningkatkan kualitas dataset.
2. Ekstraksi Fitur: Model ML menggunakan teknik pengolahan citra untuk mengekstrak fitur utama dari gambar, seperti tekstur, pola tepi, bentuk, dan distribusi warna yang dapat menunjukkan adanya kerusakan.

3. **Pelatihan Model:** Algoritma seperti Convolutional Neural Network (CNN) dilatih dengan dataset yang telah diproses untuk mengenali pola kerusakan dengan akurasi tinggi. Proses pelatihan melibatkan backpropagation, optimasi bobot jaringan, dan evaluasi kinerja model.
4. **Validasi dan Pengujian:** Model yang telah dilatih diuji menggunakan dataset terpisah untuk mengukur akurasi, recall, precision, dan F1-score dalam mendeteksi kerusakan kardus
5. **Implementasi dan Penggunaan:** Model yang telah diuji dan dioptimalkan kemudian diintegrasikan ke dalam aplikasi berbasis website. Sistem ini memungkinkan pengguna untuk mengunggah gambar kardus dan mendapatkan analisis otomatis mengenai jenis dan tingkat kerusakan yang terdeteksi.
6. **Peningkatan Berkelanjutan:** Model dapat terus diperbarui dengan dataset baru untuk meningkatkan performa dan kemampuannya dalam mengenali pola kerusakan yang lebih kompleks. Teknik seperti transfer learning dan continual learning juga dapat digunakan untuk mempertahankan akurasi model dalam jangka panjang.

Dengan perkembangan AI yang semakin pesat, aplikasi pendeteksian kerusakan berbasis AI dapat menjadi standar baru dalam industri logistik dan manufaktur, membantu meningkatkan efisiensi operasional dan memastikan kualitas produk tetap terjaga.

1.4 Deep Learning dalam Pendeteksian Kerusakan

Deep Learning adalah bagian dari Machine Learning yang menggunakan jaringan saraf tiruan (Artificial Neural Networks) dengan banyak lapisan (deep networks) untuk menganalisis dan mengenali pola data secara otomatis. Dalam konteks pendeteksian kerusakan kardus, deep learning memungkinkan sistem untuk mempelajari fitur kompleks dari gambar dan meningkatkan akurasi deteksi dibandingkan metode tradisional[5].

1.5 Convolutional Neural Networks (CNN)

CNN adalah jenis deep learning yang paling umum digunakan dalam analisis citra. CNN terdiri dari beberapa lapisan utama[6].

1. Convolutional Layer – Mengekstraksi fitur penting dari gambar dengan menerapkan filter pada piksel.
2. Pooling Layer – Mengurangi ukuran data untuk mempercepat proses komputasi dan menghindari overfitting.
3. Fully Connected Layer – Menghubungkan fitur yang telah diekstraksi ke dalam jaringan saraf untuk klasifikasi.
4. Output Layer – Menentukan hasil akhir apakah gambar menunjukkan kardus yang rusak atau tidak.

1.6 Arsitektur YOLO untuk Deteksi Objek

You Only Look Once (YOLO) adalah salah satu pendekatan untuk melakukan pendeteksian objek *secara real-time* berbasis *Convolutional Neural*

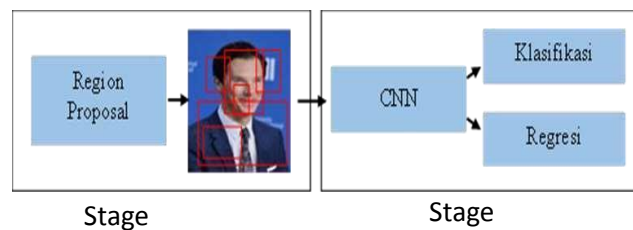
Network. Yolo menggunakan pendekatan jaringan syaraf tunggal (*Single neural network*) untuk melakukan pendeteksian objek pada sebuah frame. Jaringan ini menggunakan fitur dari semua gambar untuk memprediksi setiap bounding box yang dapat melakukan prediksi pada kotak-kotak pembatas dan probabilitas secara langsung dalam satu evaluasi[7].

(convolutional layer)

Gambar 2.1 Arsitektur *Yolo*

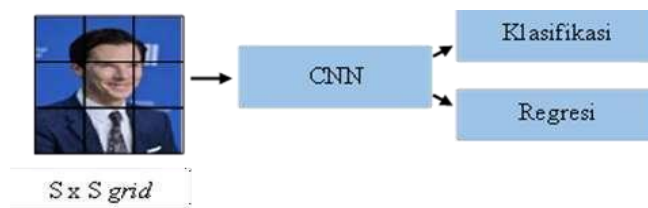
Dalam deteksi wajah, dikenal dua pendekatan utama, yaitu *multi-stage detector* dan *one-stage detector*. Pendekatan *multi-stage* melibatkan dua langkah, yakni tahap pertama untuk menghasilkan area

kandidat (*region proposal*), dan tahap kedua untuk melakukan klasifikasi pada area tersebut. Sementara itu, pendekatan *one-stage* menggabungkan kedua proses tersebut dalam satu langkah terpadu. One stage detector menggabungkan proses region proposal dan proses klasifikasi dalam satu network. Salah satu metode yang menerapkan konsep *one stage detector* adalah *You Only Look Once* (YOLO)



Gambar 2.2 Multi Stage Detector

Sumber: (Yusqi Alfian Thoriq., 2023).

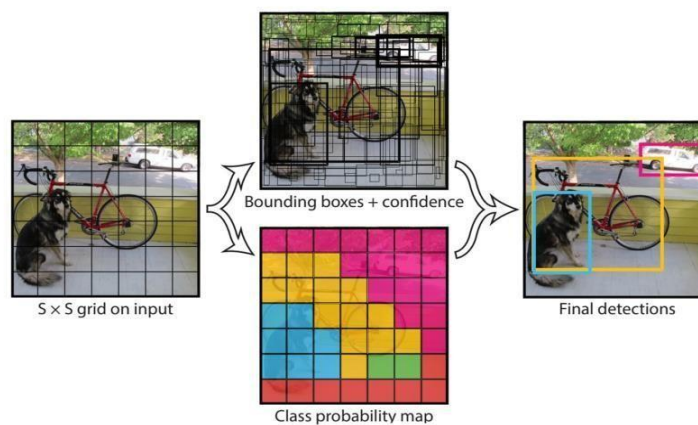


Gambar 2.3 One Stage Detector

Sumber: (Yusqi Alfian Thoriq., 2023)

Dalam *YOLO*, gambar dibagi menjadi grid berukuran $S \times S$. Setiap grid bertugas memprediksi sejumlah *bounding box* sebanyak B serta nilai probabilitas

untuk masing-masing kelas sebanyak C . Setiap *bounding box* menghasilkan lima jenis prediksi, yaitu: *confidence score* (p), koordinat x dan y sebagai pusat objek, serta w dan h yang menggambarkan lebar dan tinggi dari *bounding box*. *Confidence score* menunjukkan kemungkinan keberadaan objek dalam grid tersebut, sementara x dan y menunjukkan posisi pusat objek, dan w serta h merepresentasikan ukuran *bounding box* objek tersebut. Saat dilakukan dengan pengujian data uji, *Yolo* mengaplikasikan arsitektur yang hampir sama dengan *Convolutional neural networks*.



Gambar 2.4 Proses Deteksi Yolo

Sumber: (Salamah., 2022)

1.7 Website sebagai Platform Teknologi

1.7.1 Pengertian Website

Web, atau lebih dikenal dengan World Wide Web (WWW), adalah sistem informasi yang memungkinkan pengguna untuk mengakses dan berbagi data melalui internet. Web terdiri dari berbagai situs dan aplikasi yang dapat diakses menggunakan browser, yang memungkinkan pengguna untuk melihat informasi dalam bentuk teks, gambar, video, dan multimedia lainnya. Web menghubungkan berbagai jenis sumber daya digital di seluruh dunia, yang dapat diakses secara real-time. Teknologi web telah berkembang pesat, tidak hanya untuk tujuan informasi, tetapi juga untuk bisnis, pendidikan, komunikasi, dan berbagai sektor lainnya, memungkinkan interaksi antara pengguna dan perangkat lunak yang dapat diakses secara online dari mana saja[8].

Pengembangan aplikasi berbasis web telah menjadi pilihan utama dalam banyak industri karena kemudahan aksesnya yang dapat dilakukan melalui berbagai perangkat, seperti komputer, tablet, atau ponsel pintar. Aplikasi berbasis web memungkinkan pengelolaan data secara terpusat, di mana data dapat disimpan di server dan diakses kapan saja tanpa perlu instalasi perangkat lunak khusus pada setiap perangkat pengguna. Dengan fitur seperti responsif, integrasi dengan berbagai platform lain, serta pembaruan sistem yang dapat dilakukan secara langsung, aplikasi berbasis web memberikan fleksibilitas dan efisiensi yang lebih besar.

1.7.2 Keunggulan Aplikasi Berbasis Web

Penggunaan aplikasi berbasis web untuk pendeteksian kerusakan kardus menawarkan beberapa keunggulan, antara lain:

1. **Aksesibilitas:** Dapat diakses dari berbagai perangkat kapan saja dan di mana saja.
2. **Pembaruan Real-Time:** Data dan hasil analisis dapat diperbarui secara langsung.
3. **Integrasi Data:** Memungkinkan integrasi dengan sistem lain, seperti sistem manajemen inventaris.
4. **Efisiensi Biaya:** Tidak memerlukan instalasi perangkat lunak tambahan.

1.8 Arsitektur Client-Server

Aplikasi berbasis web biasanya menggunakan arsitektur client-server, yang terdiri dari dua komponen utama: klien dan server. Klien adalah perangkat pengguna yang bertugas mengirimkan permintaan, seperti mengunggah gambar kardus yang akan dianalisis. Permintaan ini dikirim melalui protokol HTTP atau HTTPS ke server[9].

Server bertugas memproses permintaan dengan menjalankan algoritma yang relevan, dalam hal ini model kecerdasan buatan (AI) yang dirancang untuk menganalisis gambar kardus. Server mengelola proses analisis dengan menggunakan teknologi pengolahan citra dan machine learning. Setelah selesai, server mengirimkan hasil analisis kembali ke klien. Hasil tersebut dapat berupa

informasi tentang jenis kerusakan, tingkat kerusakan, atau rekomendasi tindakan lebih lanjut.

Keunggulan dari arsitektur ini adalah fleksibilitasnya, yang memungkinkan pemrosesan dilakukan di server sehingga klien hanya memerlukan perangkat dengan kemampuan dasar untuk mengakses aplikasi melalui peramban web. Selain itu, pemisahan ini memungkinkan pengelolaan data secara terpusat dan pembaruan sistem yang lebih mudah dilakukan pada server tanpa memengaruhi perangkat klien.

1.9 Alat Bantu Pemrograman

1.9.1 Python

Python adalah bahasa pemrograman yang banyak digunakan dalam pengembangan aplikasi AI karena dukungan pustaka yang luas seperti TensorFlow, Keras, dan OpenCV. Selain itu, Python memiliki sintaks yang sederhana dan mudah dipahami, sehingga mempercepat proses pengembangan model AI. Python juga mendukung berbagai pustaka lain seperti Scikit-Learn untuk Pembelajaran mesin (*machine learning*) serta penggunaan Matplotlib dan Seaborn digunakan sebagai alat bantu untuk menampilkan visualisasi data., serta Pemanfaatan Pandas dan NumPy sebagai alat bantu dalam pengolahan serta analisis data. Dalam konteks pendeteksian kerusakan, Python digunakan untuk membangun model pembelajaran mesin yang mampu menganalisis dan mengenali pola dalam gambar dengan tingkat akurasi tinggi[10].

1.9.2 Flask

Flask adalah framework berbasis Python yang digunakan untuk membangun aplikasi web dengan cepat dan ringan. Flask mendukung integrasi dengan model machine learning dan AI, memungkinkan pembuatan API yang dapat menghubungkan backend AI dengan frontend aplikasi. Flask menyediakan berbagai fitur, seperti routing, template rendering, dan middleware, yang mempermudah pengelolaan dan pengembangan aplikasi berbasis web[11].

Selain itu, Flask mendukung ekstensi seperti Flask-RESTful untuk pembuatan API yang lebih terstruktur, Flask-SQLAlchemy untuk pengelolaan basis data, serta Flask-SocketIO untuk komunikasi real-time antara server dan klien. Dengan Flask, data gambar dapat diunggah ke server, diproses oleh model AI menggunakan pustaka seperti TensorFlow atau OpenCV, dan hasil analisis dapat ditampilkan kepada pengguna secara real-time dalam bentuk visualisasi atau laporan berbasis web.

Flask juga kompatibel dengan berbagai teknologi frontend seperti React, Vue.js, dan Angular, memungkinkan pengembangan aplikasi yang lebih interaktif dan responsif. Kombinasi antara Flask dan teknologi machine learning memungkinkan penerapan solusi AI dalam skala yang lebih luas, termasuk pendeteksian kerusakan kardus, klasifikasi objek, serta prediksi berbasis data visual.

1.9.3 Google Colaboratory

Google Colab, singkatan dari Google Colaboratory, adalah platform cloud yang disediakan oleh Google untuk menjalankan dan menulis kode Python melalui browser tanpa memerlukan konfigurasi tambahan. Ini memanfaatkan infrastruktur cloud Google dan memberikan lingkungan pengembangan yang kuat dengan akses ke GPU dan TPU (unit pemrosesan tensor) secara gratis. Dengan fitur kolaborasi yang memungkinkan pengguna untuk berbagi notebook dan bekerja secara bersama-sama secara real-time, Google Colab menjadi pilihan populer bagi para pengembang, peneliti, dan pelajar untuk menjalankan proyek-proyek Python mereka. Google Colab memanfaatkan infrastruktur cloud Google dan dapat diakses melalui peramban web, tidak memerlukan instalasi perangkat lunak tambahan, dan menyediakan kemudahan kolaborasi dengan memungkinkan pengguna untuk berbagi notebook secara langsung dengan orang lain. Platform ini sangat berguna bagi para pengembang, peneliti, dan pemula dalam bidang pemrograman untuk menjalankan proyek-proyek Python tanpa harus khawatir tentang konfigurasi perangkat keras[12].

1.9.4 OpenCV

OpenCV adalah pustaka open-source yang digunakan dalam pengolahan citra dan video untuk berbagai aplikasi, termasuk pendeteksian objek, analisis pola, dan segmentasi gambar. Dalam konteks pendeteksian kerusakan kardus, OpenCV digunakan untuk memproses gambar dan mengekstraksi fitur visual

seperti bentuk, warna, tekstur, serta mendeteksi tepi dan area kerusakan. Teknologi ini memungkinkan sistem untuk meningkatkan akurasi deteksi dengan menerapkan algoritma seperti Canny Edge Detection, Thresholding, dan Contour Detection. Selain itu, OpenCV dapat diintegrasikan dengan pustaka lain seperti TensorFlow dan Keras untuk membangun model deep learning yang lebih canggih[13].

OpenCV juga menawarkan berbagai metode pengolahan citra seperti transformasi morfologi, filtering, dan analisis histogram yang dapat membantu dalam meningkatkan visibilitas fitur kerusakan pada kardus. Teknik seperti adaptive thresholding dapat digunakan untuk memisahkan area kerusakan dengan lebih jelas, sementara metode morphological operations seperti erosion dan dilation membantu dalam memperhalus batas-batas objek yang terdeteksi. Selain itu, OpenCV memiliki fitur optical flow yang dapat digunakan untuk mendeteksi perubahan bentuk dari serangkaian gambar, yang berguna untuk mendeteksi deformasi kardus selama proses distribusi dan pengiriman. Dengan kombinasi fitur-fitur ini, sistem pendeteksian kerusakan kardus berbasis OpenCV dapat menghasilkan analisis yang lebih akurat dan andal.

1.9.5 Roboflow

Roboflow adalah framework pengembang computer vision untuk pengumpulan data yang lebih baik ke prapemrosesan, dan teknik pelatihan model. Roboflow memiliki kumpulan data publik yang tersedia bagi pengguna dan juga memiliki akses bagi pengguna untuk mengunggah data khusus mereka sendiri.

Cara kerja roboflow yaitu mengubah dataset yang telah dilabeli ke dalam format yang sesuai dengan YOLOv8 [14].

1.9.6 Aplikasi Visual Code Studio

Visual Studio Code (VS Code) adalah editor kode sumber yang ringan dan kuat yang dikembangkan oleh Microsoft, yang mendukung berbagai bahasa pemrograman seperti JavaScript, Python, C++, dan banyak lagi. Salah satu keunggulan utama dari VS Code adalah kemampuannya untuk disesuaikan melalui berbagai ekstensi yang memungkinkan penggunanya menambah fitur sesuai kebutuhan, seperti dukungan untuk bahasa pemrograman tambahan, integrasi sistem kontrol versi, hingga tools debugging. Dengan antarmuka yang sederhana dan intuitif, VS Code memungkinkan pengembang untuk menulis dan mengelola kode dengan lebih efisien. Fitur seperti auto-completion, syntax highlighting, dan linting sangat membantu dalam meningkatkan produktivitas pengembang.

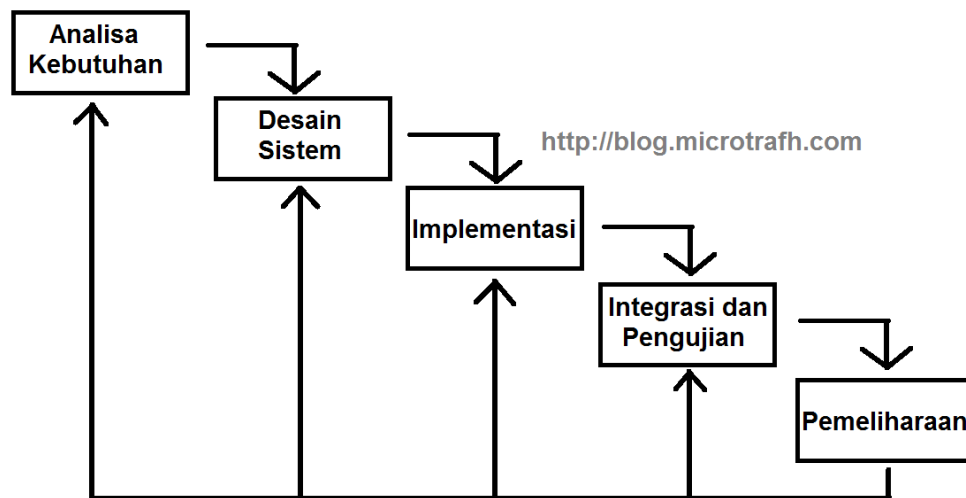
Selain itu, Visual Studio Code juga menawarkan berbagai fitur kolaborasi dan integrasi dengan platform pengembangan lain, seperti GitHub, yang memudahkan pengelolaan proyek bersama tim. Kemampuan debugging langsung dalam editor dan integrasi dengan terminal membuatnya menjadi pilihan populer bagi banyak pengembang perangkat lunak. Dukungan terhadap berbagai sistem operasi, seperti Windows, macOS, dan Linux, menjadikan VS Code sebagai alat yang sangat fleksibel untuk berbagai jenis proyek, dari pengembangan aplikasi web hingga perangkat lunak berbasis server. Karena sifatnya yang ringan, VS

Code sangat cocok untuk digunakan dalam proyek kecil maupun besar, memberikan keseimbangan antara kemudahan penggunaan dan kemampuan pengembangan tingkat lanjut[15].

1.10 Metode Pengembangan Sistem

1.10.1 Waterfall

Model pengembangan sistem yang diterapkan dalam penelitian ini adalah metode Waterfall, yang juga dikenal sebagai metode air terjun. Pendekatan ini digunakan untuk menggambarkan proses pengembangan perangkat lunak secara sistematis dan terstruktur, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya. Dengan pendekatan berurutan seperti ini, metode Waterfall memastikan bahwa semua kebutuhan sistem dapat terpenuhi dengan baik. Hasil akhirnya adalah sebuah sistem yang dirancang untuk berjalan secara optimal sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Metode ini sering dipilih karena kemampuannya untuk memberikan kerangka kerja yang jelas dan terorganisasi selama proses pengembangan perangkat lunak[16].



Gambar 2.5 Metode Waterfal

Sumber: <https://dwiwijaya97.blogspot.com/2017/02/metode-wterfall.html>

Tahap – tahap pengembangan waterfall model adalah :

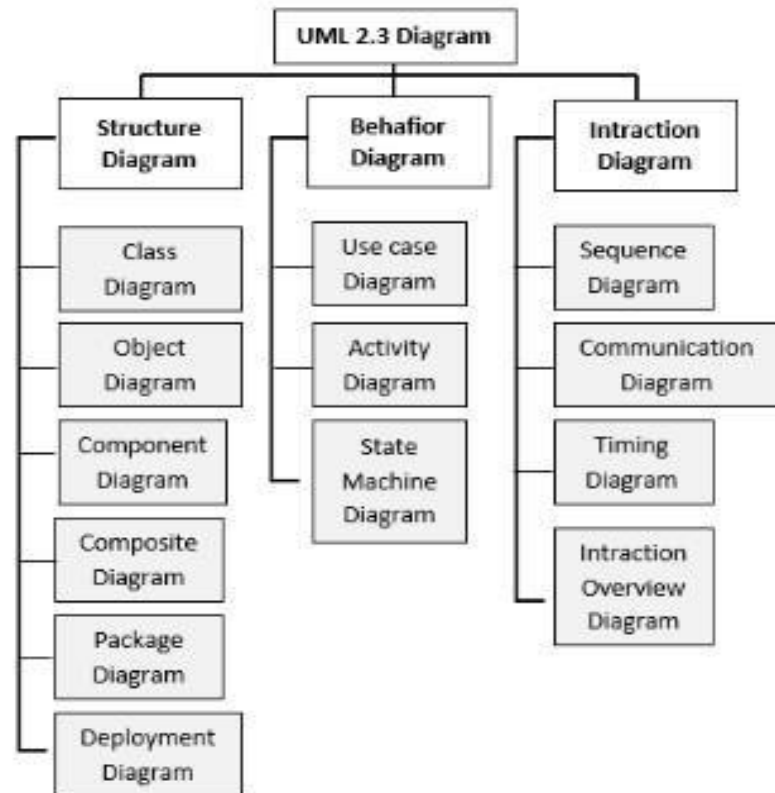
1. Analisis dan definisi persyaratan Pelayanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan user.
2. Perancangan sistem dan perangkat lunak Kegiatan ini menentukan arsitektur sistem secara keseluruhan.
3. Implementasi dan pengujian unit Perancangan perangkat lunak direalisasikan sebagai serangkaian program.
4. Integrasi dan pengujian sistem Unit program diintegrasikan atau diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sitem telah terpenuhi
5. Operasi dan pemeliharaan Merupakan fase siklus yang paling lama. Sistem diinstall dan dipakai. Perbaikan mencakup koreksi dari

berbagai error, perbaikan dan implementasi unit sistem dan pelayanan sistem.

1.11 Alat Bantu Perancangan

1.11.1 Pengertian UML (Unified Modelling Language)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk merancang dan mendokumentasikan sistem perangkat lunak. UML menyediakan cara standar untuk memvisualisasikan desain suatu sistem, termasuk interaksi antar komponen, alur kerja, serta struktur internalnya. Dikembangkan oleh Object Management Group (OMG), UML telah menjadi standar industri dalam pengembangan perangkat lunak berbasis objek. Dengan menggunakan UML, pengembang dapat memodelkan berbagai aspek sistem. Dengan penyusunan yang rapi dan konsisten, proses ini membantu memperjelas komunikasi serta mendukung kerja sama yang efektif di antara seluruh anggota tim proyek. UML (Unified Modeling Language) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (ObjectOriented)[17]. UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.



Gambar 2.6 Diagram UML

Sumber, (Voutama, A., & Novalia, E. (2022))

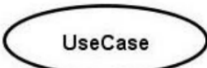
1.11.2 Use Case Diagram

Use Case Diagram secara umum merupakan representasi visual dari proses sistem secara menyeluruh yang melibatkan aktor sebagai pengguna. Use case berfungsi untuk menggambarkan bagaimana para pemangku kepentingan sistem berinteraksi dengan sistem tersebut. Pengembangan use case berguna untuk mendalami dan memahami kebutuhan sistem secara lebih rinci.

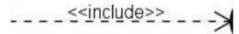
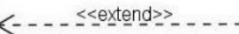
Berdasarkan kutipan tersebut, *Use Case Diagram* menekankan representasi fungsi atau layanan yang ditawarkan oleh sistem dari sudut pandang pengguna. Dalam diagram ini, aktor direpresentasikan sebagai entitas eksternal yang berhubungan dengan sistem untuk mencapai tujuan tertentu, sementara sistem digambarkan sebagai sebuah kotak yang mencakup berbagai fungsi yang disediakan.

Berikut adalah simbol-simbol yang ada pada *use case diagram* dapat dilihat pada tabel berikut ini :

Tabel 2.1 Use Case Diagram

Simbol	Nama	Deskripsi
	Use case	fungsi yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
	Aktor / actor	Abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan Use Case, tetapi tidak memiliki kontrol terhadap use case
	Asosiasi /	Asosiasi antara aktor dan use case,



	association	digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data
	Asosiasi / association	Asosiasi antara aktor dengan use case yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	Include	Include, merupakan di dalam use case lain (required) atau pemanggilan use case oleh use case contohnya adalah pemanggilan sebuah fungsi program
	Extend	Extend, merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi

Sumber: (http://repository.bsi.ac.id/repo/files/205310/download/File_9-Daftar-Symbol.pdf)

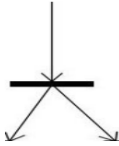
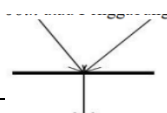
1.11.3 Activity Diagram

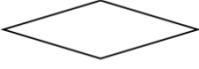
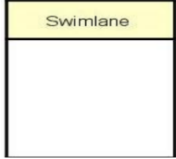
Activity Diagram adalah representasi alur aktivitas atau alur kerja dalam sebuah sistem yang akan dijalankan. Diagram ini memungkinkan perancang untuk menentukan bagaimana sistem aplikasi mencapai tujuannya serta menggambarkan urutan aktivitas dalam proses pengoperasian aplikasi.

Diagram ini menggambarkan cara suatu aktivitas atau proses berlangsung dalam sistem, menunjukkan urutan langkah-langkah yang harus dilakukan untuk mencapai tujuan tertentu. Activity Diagram sangat bermanfaat dalam memvisualisasikan alur kontrol dan pengambilan keputusan yang terjadi, serta menjelaskan proses atau alur kerja dari awal hingga akhir, termasuk aktivitas yang perlu dilakukan, keputusan yang harus diambil, dan kondisi yang harus dipenuhi.

Berikut adalah simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel berikut ini :

Tabel 2.2 Activity Diagram

Simbol	Nama	Deskripsi
	Start Point	Start Point, diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	End Point	End Point, akhir aktivitas
	Activities	Activities, menggambarkan suatu proses atau kegiatan bisnis
	Fork atau Percabangan	Fork atau percabangan, digunakan untuk menunjukan kegiatan yang dilakukan secara paralel untuk menggabungkan dua kegiatan paralel menjadi satu
	Join atau Penggabungan	Join (penggabungan) atau rake, digunakan untuk menunjukan adanya

		dekomposisi
	Decision Points	Decision points, menggambarkan pilihan untuk pengambilan keputusan, true atau false
	Swimlane	Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa

Sumber: (http://repository.bsi.ac.id/repo/files/205310/download/File_9-Daftar-Symbol.pdf)

1.11.4 Class Diagram

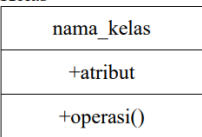
Class Diagram adalah representasi visual yang menunjukkan struktur sistem melalui definisi kelas-kelas yang akan digunakan dalam pengembangan sistem. Kelas (class) merupakan sekumpulan objek yang memiliki struktur, perilaku, hubungan, dan makna yang serupa. Penentuan atau identifikasi kelas dilakukan dengan menganalisis objek-objek yang terdapat dalam Sequence Diagram dan Collaboration Diagram. Sebuah kelas biasanya digambarkan dalam bentuk persegi panjang yang terbagi menjadi tiga bagian.

Class Diagram digunakan untuk memvisualisasikan struktur statis suatu sistem, dengan menekankan pada kelas-kelas, atribut, metode (operasi), dan hubungan antar kelas dalam sistem tersebut. Class Diagram adalah alat yang sangat krusial dalam perancangan sistem berbasis objek karena memberikan

gambaran yang jelas tentang cara data dan fungsionalitas diorganisasikan dalam sistem[20].

Berikut adalah simbol-simbol yang ada pada class diagram dapat dilihat pada tabel berikut ini :

Tabel 2.3 Clas Diagram

Simbol	Nama	Deskripsi
	Kelas	Kelas pada struktur sistem
	Antar muka / interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
	Asosiasi / association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
	Asosiasi berarah / directed association	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
	Kebergantungan	Relasi antarkelas dengan makna

	/ dependency	kebergantungan antarkelas
	Agregasi / aggregation	Relasi antar kelas dengan makna semua – bagian (whole-part)

Sumber: (http://repository.bsi.ac.id/repo/files/205310/download/File_9-Daftar-Simbol.pdf)

1.11.5 Sequence Diagram

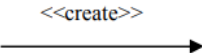
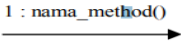
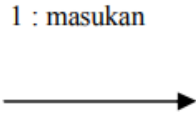
Sequence Diagram adalah diagram yang menunjukkan interaksi antar objek yang disusun berdasarkan urutan waktu. Diagram ini menggambarkan objek dan kelas yang terlibat dalam sebuah skenario, serta pesan-pesan yang dipertukarkan antara objek-objek tersebut untuk melaksanakan fungsi dalam skenario tersebut.

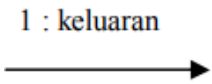
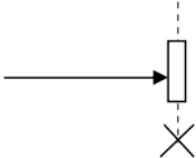
Sequence Diagram menggambarkan urutan interaksi antara objek-objek dalam sistem, dari awal hingga akhir suatu proses. Diagram ini memperlihatkan objek-objek yang terlibat serta urutan pesan yang dikirim antar objek untuk melaksanakan fungsi atau aktivitas tertentu. Setiap objek dalam diagram ini diwakili oleh sebuah lifeline yang digambarkan sebagai garis vertikal, sementara interaksi antar objek ditunjukkan dengan panah horizontal yang menggambarkan pesan yang dikirim.

Berikut adalah simbol-simbol yang ada pada Sequence diagram dapat dilihat pada tabel berikut ini :

Tabel 2.4 Sequence Diagram

Simbol	Nama	Deskripsi
--------	------	-----------

 nama aktor	Aktor	poses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
	Garis hidup / lifeline	Menyatakan kehidupan suatu objek
<u>nama objek :</u> <u>nama kelas</u>	Objek	Menyatakan objek yang berinteraksi pesan
	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
	Pesan tipe create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
	Pesan tipe call	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
	Pesan tipe send	Menyatakan bahwa satu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada

		objek yang dikirim
	Pesan tipe return	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
	Pesan tipe destroy	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

Sumber: (<https://elib.unikom.ac.id/download.php?id=118668>)

1.12 Kerangka Kerja Penelitian

Kerangka kerja penelitian yang akan dilakukan disajikan pada tabel dibawah ini yaitu sebagai berikut :

Tabel 2.5 Kerangka Kerja Penelitian

No	Kegiatan	Nov	Des	Jan	Feb	Mar
1	Penentuan Topik Penelitian	✓	✓			
2	Pengumpulan Data			✓		

3	Preprocessing dan Clening Data				✓	
4	Pelatihan Model dengan <i>Yolov8</i>				✓	✓
5	Pengujian Model					✓
6	Seminar Proposal					✓
7	Sidang Akhir					✓

Catatan :

- Durasi setiap kegiatan dapat disesuaikan dengan kebutuhan spesifikasi penelitian.
- Waktu pelaksanaan bergantung pada kalender akademik dan kesiapan

1.13 Penelitian Terdahulu

Tabel 2.6 Penelitian Terdahulu

No	Peneliti	Tahun	Judul Penelitian	Metode	Hasil
1	Putra Bagus Arya Adi A.	2020	Implementasi Metode Certainty	Certainty Factor	Sistem pakar dapat mendeteksi kerusakan dengan

			Factor Pada Sistem Pakar Pendeteksi Kerusakan Printer		akurasi tinggi
2	Hidayat Haryono, Rachmat Muhendra, Oki Wibowo Nugroho	2023	Analisis Paket Damage dan Perbaikan Kualitas Produk pada Jasa Logistik	PDCA dan FMEA	Identifikasi penyebab utama kerusakan paket dan strategi perbaikannya
3	Ahmad Kusuma Muhammad, Ikhsan Habriansyah, Putra Anggraini, Mutiarah Jannah	2024	Penerapan Machine Learning untuk Memantau Kerusakan Kontainer di Terminal Petikemas	Machine Learning	Pendeteksian otomatis meningkatkan efisiensi dan akurasi inspeksi

4	Satria Ferry Firmansyah	2023	Deep Learning untuk Pendeteksian Kemasan Biskuit Rusak	Deep Learning (CNN)	Model CNN berhasil mengklasifikasikan kemasan dengan akurasi tinggi
5	Dimas Naufal Alfarizi, Rizky Ardi Pangestu, Dwi Aditya, M. Aulia Setiawan, Putri Rosyani	2023	Penggunaan Metode YOLO pada Deteksi Objek	YOLO	Metode YOLO mampu mendeteksi objek dengan kecepatan tinggi dan akurat