

## BAB 2

### DASAR TEORI

#### 2.1 KAJIAN PUSTAKA

*Artificial Intelligence* (AI) adalah studi dalam ilmu komputer yang dapat membuat suatu mesin komputer bisa melakukan tugas layaknya manusia[6]. Dalam AI ada 2 metode utama yang sering digunakan, yaitu ada ML dan DL. ML merupakan suatu metode dalam AI yang memanfaatkan teknik pembelajaran dari data untuk memberikan kemampuan pada sistem komputer agar dapat mengembangkan kemampuan belajar sendiri, berpikir sendiri, dan mengambil keputusan sendiri[7]. Sedangkan DL meniru proses cara berpikir manusia yaitu memungkinkan suatu sistem melakukan pembelajaran secara otomatis melalui data yang diberikan sehingga dapat mengidentifikasi secara kompleks. Untuk dapat mengklasifikasi suatu objek maka digunakan algoritma CNN yang sudah terbukti memiliki akurasi tinggi dalam proses klasifikasi objek. Dalam pendeteksian objek, arsitektur YOLO banyak digunakan. Ada beberapa versi dari YOLO salah satunya YOLOV5 yang akan digunakan karena memiliki akurasi pendeteksian objek yang mumpuni teruji dari beberapa penelitian yang sudah dilakukan dan mendapatkan akurasi yang terbaik[4].

Pada penelitian [1] yang membahas mengenai identifikasi citra kemasan berjenis kardus yang terdiri dari 2 kelas yaitu *defect* dan *no defect*. Tujuan dari penelitian ini yaitu untuk membedakan antara kemasan kardus *defect* dan *no defect* secara visual yaitu terdapat sobekan, lubang serta penyok pada kemasan kardus *defect* sehingga dapat mempermudah dalam melakukan *quality control* dalam melakukan distribusi barang. Model dibuat menggunakan metode CNN dengan *image processing*. Model dibuat dengan *Hyperparameter epoch* sebanyak 30, input size pada gambar sebesar 300x300 *pixel* dan *learning rate* sebesar 0,001. Model ini mendapatkan performa terbaik dengan nilai akurasi 95,77%, presisi 96%, *recall* 96%, *f1-score* 96% dan *loss* sebesar 0.1478 sehingga dapat dikatakan model yang telah dibuat sudah bagus dalam melakukan klasifikasinya. Namun, penelitian ini juga mengidentifikasi beberapa tantangan, seperti kesulitan dalam mendeteksi kerusakan yang lebih halus dan kebutuhan untuk dataset yang lebih besar untuk menguji generalisasi model[1].

Pada penelitian [8] membahas mengenai manajemen inventori yang dimanfaatkan untuk mengenali barang barang yang ada di Gudang secara otomatis menggunakan CNN. CNN digunakan untuk mengklasifikasi citra berupa gambar jenis jenis kemasan dan di implementasikan melalui *website*. Untuk arsitektur yang digunakan yaitu ResNet dan akurasi yang diperoleh sebesar 90%. Namun dalam skenario uji dengan dua atau lebih objek mengalami penurunan akurasi sebesar 56% hingga 54% Penelitian ini menyoroti pentingnya peningkatan algoritma untuk menangani pengenalan beberapa objek secara simultan. Salah satu solusi yang disarankan adalah dengan menggunakan metode deteksi objek yang lebih canggih seperti YOLO atau Faster R-CNN untuk meningkatkan akurasi dalam skenario multi-objek. Selain itu, integrasi teknologi sensor tambahan seperti RFID atau IoT dapat membantu meningkatkan keakuratan dan efisiensi sistem manajemen inventori otomatis.[8].

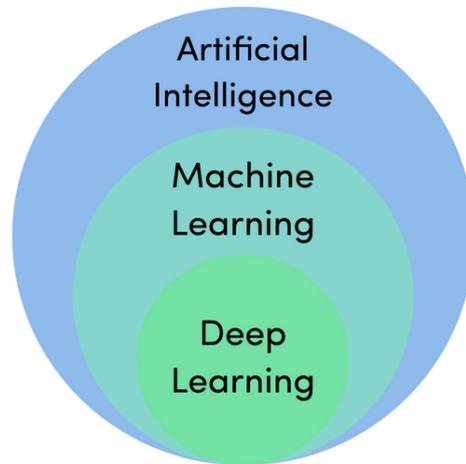
Penelitian selanjutnya [4] yang membahas mengenai perbandingan objek deteksi pada karung komoditi menggunakan dua metode yang berbeda yaout YOLOV4 dan *Mask R-CNN*. Model ini terdiri dari 2 jenis kelas yaitu karung dan lubang dengan masing masing terdapat 700 untuk data *training* dan 100 untuk data *testing*. Tujuan dari penelitian ini yaitu untuk mengidentifikasi keadaan karung komoditi pada suatu penyimpanan menggunakan sistem deteksi objek berupa YOLOV4 dan *Mask R-CNN*. Dalam melakukan pengujian, model yang sudah jadi digunakan untuk mendeteksi 20 data uji (karung komoditi) yang menghasilkan bahwa model dari YOLOV4 memberikan performa lebih baik dengan akurasi 96,8%, sedangkan model *Mask R-CNN* mendapatkan performa yang kurang dengan akurasi 65,78% pada data uji yang sama[4].

## **2.2 DASAR TEORI**

### **2.2.1 AI**

AI atau kecerdasan buatan adalah upaya untuk melakukan sistem otomatisasi dari tugas-tugas intelektual yang biasanya dilakukan oleh manusia, seperti pengenalan wajah, pengolahan bahasa, pengambilan keputusan, dan analisis

data. AI memanfaatkan berbagai teknik seperti ML, DL, dan jaringan saraf tiruan untuk meniru kemampuan kognitif manusia.



**Gambar 2.1 Korelasi antara AI, ML, dan DL[9]**

Pada Gambar 2.1 *artificial intelligence* mempunyai dua cabang utama yaitu ML dan DL. DL merupakan subbidang dari ML dimana DL menggunakan *deep neural network* dalam menganalisis sebuah data dan mempelajari pola sedangkan ML melatih komputer mempelajari dari sebuah data[9].

AI berfungsi sebagai jembatan atau “*bridge*” antara kecerdasan manusia dengan sistem komputer, dimana AI diciptakan atas kecerdasan manusia itu sendiri dan komputer berlatih berdasarkan *input* data yang besar untuk dapat mengeksekusi suatu tugasnya. Teknologi pada AI dapat mengambil suatu keputusan berdasarkan analisis dengan memanfaatkan data tersebut dalam jumlah yang sangat besar. Ada 3 tahapan utama dalam proses AI yaitu, pelatihan atau pembelajaran dari data yang akan digunakan, penalaran, dan kemampuan untuk mengoreksi diri atau “*self correction*” yang menyerupai proses pemikiran pada manusia sebelum melakukan pengambilan keputusan[10].

AI atau kecerdasan buatan ini sering disebut dengan berbagai istilah, seperti sistem cerdas buatan, kecerdasan artifisial, intelegensi artifisial, kecerdasan mesin, mesin cerdas dan sebagainya. Berikut merupakan definisi dari AI yaitu :

- 1) Kecerdasan buatan adalah bagian dari ilmu komputer yang terkait erat dengan pembelajaran kreasi dalam komputer yang menunjukkan beberapa bentuk sistem kecerdasan.
- 2) Suatu algoritma yang mempelajari konsep baru.

- 3) Sistem yang dapat berpikir, belajar dan dapat menyimpulkan dari hasil pembelajaran sehingga berguna bagi dunia sekitar
- 4) Sistem yang dapat mengerti suatu bahasa dan gambar.
- 5) Ilmu pengetahuan dalam dunia komputer yang ditujukan khusus untuk merancang sistem otomatisasi dalam kecerdasan komputer[11].

AI dalam penerapannya masih banyak kekurangan dan kelebihan, baik dari sisi teknis maupun penerapannya. Berikut merupakan kelebihan dan kekurangan dari *artificial intelligence* :

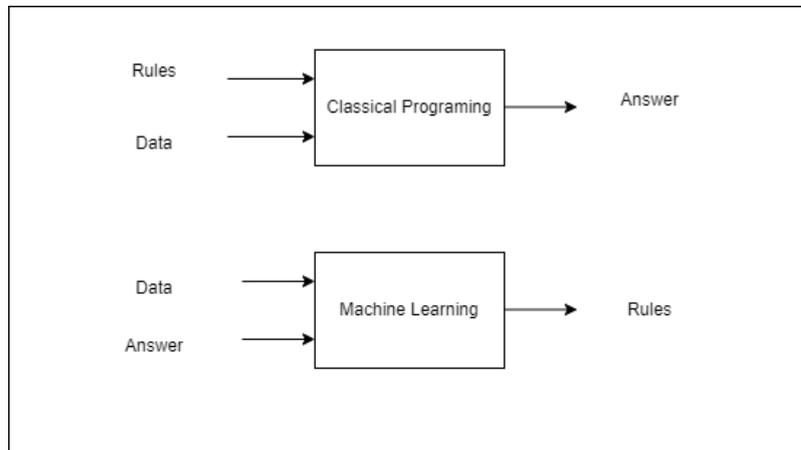
**Tabel 2.1 Kelebihan dan Kekurangan AI**

Kelebihan	Kekurangan
Efisiensi Produktifitas dalam suatu pekerjaan[2].	Minimnya Tenaga Ahli dalam bidang kecerdasan buatan [12].
Perkembangan Inovasi Teknologi <i>artificial</i> Intelegence yang pesat	Teknologi <i>artificial intelegence</i> yang canggih dapat menggantikan pekerjaan yang semula dilakukan oleh manusia dapat dilakukan oleh mesin[6].
Dapat diaplikasikan dalam berbagai ilmu pengetahuan [6].	Bergantung pada data yang digunakan untuk proses <i>training</i> , semakin banyak data maka akan menghasilkan data yang akurat dan valid
Dapat menganalisis sejumlah besar data dengan cepat[13].	Pemahaman yang terbatas, tidak mampu menyelesaikan masalah yang kompleks

### 2.2.2 ML

ML merupakan pembelajaran mengenai mengambil pengetahuan dari sebuah data. ML merupakan gabungan dari beberapa bidang pengetahuan yaitu statistika, kecerdasan buatan dan ilmu komputer[14]. Pendekatan di bidang AI yang biasanya digunakan untuk meniru dan mengadopsi tingkah laku manusia dalam melakukan penyelesaian masalah atau "*problem solving*". ML menerapkan proses otomatisasi dalam menyelesaikan masalah. ML berusaha untuk mengikuti bagaimana manusia tersebut dapat belajar dan memberikan keputusan atau dapat menyimpulkan lebih luas dari pola dan sifat umum dari data yang dipelajari. Ada dua penerapan yang digunakan pada metode ML, yaitu prediksi dan klasifikasi. Salah satu ciri khas pada ML adalah adanya proses pelatihan atau *training*. Proses pelatihan membutuhkan dataset yang besar sehingga nilai validasi atau *output* yang dihasilkan lebih akurat. Klasifikasi ialah metode yang digunakan pada ML yang memungkinkan mesin untuk dapat mengelompokkan objek atau kelas yang memiliki ciri khusus tertentu secara otomatis. Hal ini mirip seperti manusia dalam

mengkategorikan antara objek yang berbeda beda kemudian dapat membedakanya sekaligus mengelompokanya. Sedangkan prediksi atau regresi pada ML digunakan untuk mengestimasi keluaran berdasarkan data *input* yang telah dipelajari selama proses *training*. Beberapa metode ML yang populer yaitu *linear regression*, *decision trees*, *random forest*, *naïve bayes*, *Support Vector Machine (SVM)*, *K-nearest neighbour*, *K-means clustering* dan jaringan saraf tiruan[15].



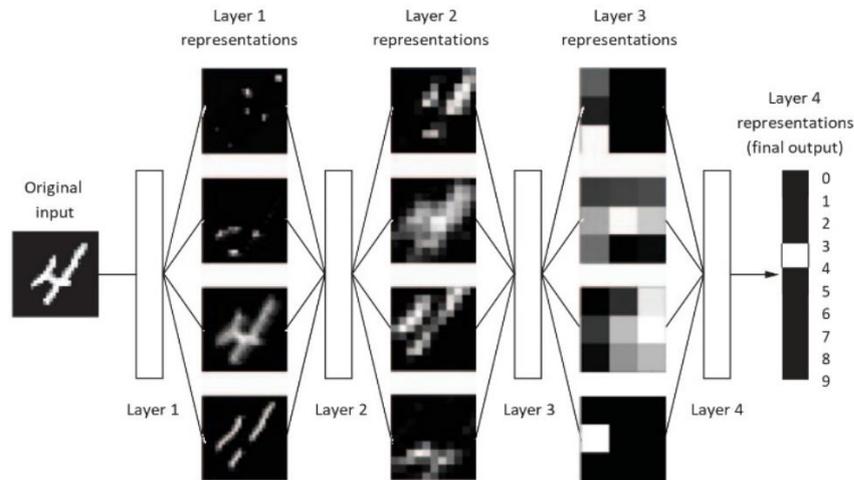
**Gambar 2.2 Perbandingan pemrograman konvensional dan ML[14]**

Pada Gambar 2.2 dijelaskan bahwa perbedaan pemrograman konvensional dengan ML yaitu pada pemrograman konvensional, *user* memasukan sebuah program atau aturan dan data yang akan diproses sesuai dengan aturan tersebut kemudian menghasilkan sebuah jawaban (*output*). Sedangkan pada ML, *user* memasukan data sekaligus jawaban (*output*) yang diharapkan dari data tersebut menghasilkan sebuah aturan yang nantinya bisa diterapkan pada data baru untuk menghasilkan jawaban sesuai yang diharapkan[9].

### 2.2.3 DL

DL adalah metode di dalam jaringan saraf atau “*neural network*” yang bermanfaat untuk meningkatkan kinerja pada proses pembelajaran jaringan saraf dengan banyak lapisan yaitu dengan memanfaatkan metode yang khusus seperti *Restricted Boltzmann Machine (RBM)*, yaitu lebih dari 7 lapisan[15]. Keunggulan dari DL ini adalah lebih efisien dan memberikan peningkatan akurasi yang tinggi sebanding dengan jumlah data yang dilatih. DL semakin cepat dan efisien setelah ditemukanya metode penerapan yang menggunakan *Graphic Processing Units (GPU)*. GPU adalah sebuah prosesor khusus untuk pengolahan grafis yang biasanya digunakan

pada *software games*. Proses pembelajaran DL dilakukan secara paralel dengan memanfaatkan GPU sehingga prosesnya lebih cepat dibanding hanya mengandalkan dari proses *Central Processing Unit* (CPU). Kekurangan DL yaitu membutuhkan data yang besar untuk dapat mempelajari suatu masalah yang lebih kompleks. Dalam DL juga memerlukan perangkat keras yang mumpuni sehingga dapat melatih model dengan cepat[16].



**Gambar 2.3 Representasi mendalam yang dipelajari pada model klasifikasi digit[9]**

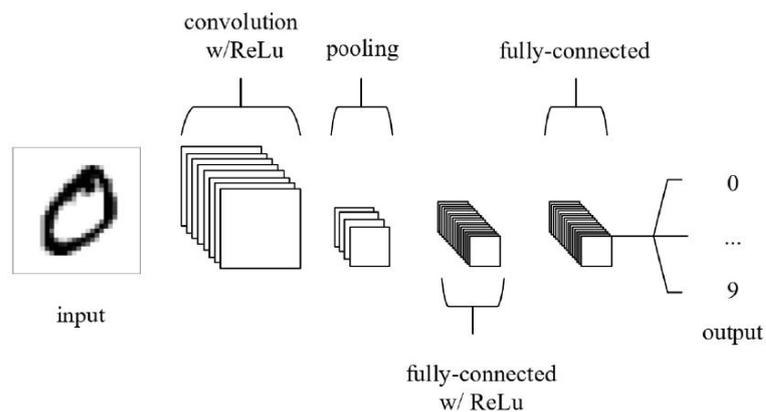
Pada Gambar 2.3 dijelaskan mengenai implementasi DL yaitu klasifikasi dari gambar digit dimana, jaringan mengubah gambar digit menjadi pencitraan yang semakin berbeda dari gambar aslinya dan semakin informatif. Jadi teknik tersebut yang digunakan dalam DL yaitu sebuah metode bertingkat untuk mempelajari representasi data[9].

Sejumlah arsitektur DL banyak digunakan dan banyak dijelaskan dalam literatur lainnya, seperti *Recurrent Neural Network* (RNN), *Deep Trust Networks* (DBN), CNN, *Deep Neural Networks* (DNN), *Deep Autoencoders* (DA), *Deep Boltzmann Machines* (DBM)[17].

#### 2.2.4 CNN

Metode DL mulai merambah ke dalam bidang pengenalan objek, seperti ekstraksi fitur dan klasifikasi atau pengelompokan menggunakan CNN. CNN adalah salah satu kelas dari DL yang menggunakan teknik *backpropagation* (pembelajaran) dan klasifikasi dalam implemenetasinya[1]. Metode CNN ini

memiliki akurasi *output* yang relatif tinggi terbukti metode ini sering digunakan dalam klasifikasi objek dan mendapatkan akurasi yang tinggi. Jaringan saraf konvolusional bekerja dengan menerima masukan berupa gambar sehingga banyak digunakan dalam pengenalan citra. *Input* tersebut kemudian dilatih dengan beberapa lapisan seperti “*softmax*” untuk menghasilkan suatu *output* yang dapat mengklasifikasikan objek dari *input* tersebut[18]. Contoh konfigurasi CNN yang populer dan umum digunakan adalah AlexNet, VGGNet (*Visual Geometry Group*), dan GoogLeNet, ResNet, LeNet-5, ZFNet dan sebagainya[17].



**Gambar 2.4** Arsitektur CNN[9]

Pada Gambar 2.4 merupakan algoritma dari CNN. Algoritma pada CNN ini sangat sederhana terdiri dari *input layer*, *output layer*, dan *hidden layers*. Dalam *hidden layers* mempunyai beberapa lapisan yaitu ada :

1) *Convolutional layer*

*Convolutional layer* adalah inti dari CNN. Lapisan ini menggunakan kernel CONV sebagai sebuah filter untuk menggeser gambar asli. Setiap nilai piksel dari data berkorelasi lokal di dalam filter yang kemudian akan dikalikan dan ditambahkan untuk menghasilkan hasil konvolusi. Parameter pada lapisan ini diantaranya meliputi ukuran kernel, *depth*, *zero-padding*, dan jumlah filter. Perhitungan konvolusi secara matematis ditulis dengan persamaan :

$$h(x) = f(x) * g(x) \tag{2.1}$$

dimana  $h(x)$  merupakan parameter yang dihasilkan dari konvolusi citra *input* berupa  $f(x)$  dan  $g(x)$ . Perhitungan konvolusi pada gambar didapatkan hasil yang berbeda beda tergantung pada *padding* dan *stride* yang digunakan[19].

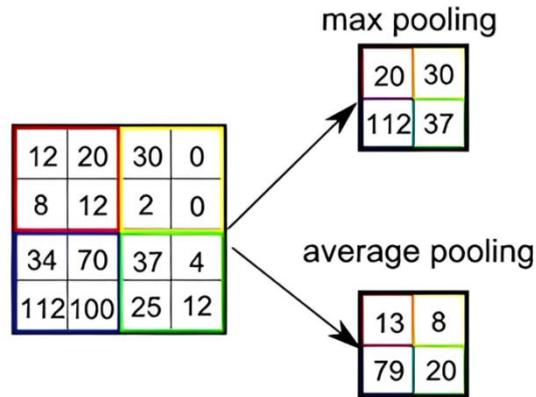
*Convolutional layer* menggunakan filter yaitu kernel yang digunakan untuk mengekstraksi sebuah objek dari sebuah *inputan*. Kernel berisi bobot yang digunakan untuk mengidentifikasi objek atau karakter dari objeknya. Operasi konvolusi digunakan untuk menghasilkan transformasi linear dari *inputan* yang mencocokkan informasi yang terdapat dalam data. Bobot dalam lapisan tersebut mencakup kernel konvolusi yang digunakan, yang memungkinkan pelatihan data berdasarkan *input* dalam CNN[20].

*Convolutional Layer* memiliki beberapa *hyperparameter* diantaranya :

- a) *Depth*, pada *depth* volume dari *output* yang dihasilkan oleh *convolutional layer* dapat diatur secara manual melalui jumlah neuron di dalam *layer* ke wilayah yang sama dengan *input*
- b) *Stride*, *stride* digunakan untuk mengatur kedalaman kedalaman disekitar dimensi yang di masukan untuk menempatkan bidang reseptif
- c) *Zero Padding*, pada parameter ini digunakan untuk mengisi batas *input*, dan merupakan metode yang efektif untuk memberikan kontrol yang lebih lanjut mengenai dimensi *volume* pada *output*[21].

## 2) *Pooling layers*

Pada lapisan ini digunakan untuk mengurangi jumlah dari hasil dimensi dari layer CONV yang terletak di antara 2 lapisan CONV. Ada 3 jenis dari *pooling layer* yaitu *general pooling*, *overlapping pooling*, dan *spatial pyramid pooling*. Metode yang paling umum digunakan adalah *max* dan *average pooling*. Untuk mengatur dimensi gambar agar sama maka harus dilakukan perubahan konvolusi gambar menjadi dimensi yang sama dengan *spatial pyramid pooling*. *Max pooling* mengambil nilai maksimum dari setiap patch kecil dari gambar, yang membantu menyoroti fitur paling dominan dan mengurangi dimensi secara signifikan. *Average pooling*, digunakan untuk menghitung rata-rata nilai dari setiap *patch* kecil, yang membantu mengurangi *noise* dan mempertahankan informasi konteks. *Spatial pyramid pooling* memungkinkan model untuk menerima *input* gambar dengan ukuran yang bervariasi dan mengubahnya menjadi ukuran tetap tanpa perlu *cropping* atau *warping*, sehingga lebih fleksibel dan dapat mengakomodasi berbagai ukuran *input*[19].



**Gambar 2.5 Jenis dari *Pooling layer*[22]**

Pada Gambar 2.5 merupakan teknik pada *pooling layer*, yaitu *max pooling* yang digunakan untuk memilih nilai tertinggi dari area gambar yang dicakup oleh kernel, sedangkan *average pooling* digunakan untuk menghitung rata-rata dari semua nilai dalam area gambar yang dicakup oleh kernel. Gambar di sebelah kiri menunjukkan matriks *input* 4x4. *Max Pooling* mengambil nilai maksimum dari setiap area 2x2 yang dicakup oleh kernel dan menghasilkan matriks yang lebih kecil berukuran 2x2. Nilai maksimum dari setiap area adalah area berwarna merah 20, area berwarna hijau 30, area berwarna biru 112, area berwarna kuning 37. Sehingga didapatkan hasil *max pooling* seperti pada gambar 2.5. Selanjutnya pada *average pooling* gambar di sebelah kiri menunjukkan matriks *input* yang sama berukuran 4x4. *Average pooling* mengambil nilai rata-rata dari setiap area 2x2 yang dicakup oleh kernel dan menghasilkan matriks yang lebih kecil berukuran 2x2. Nilai rata-rata dari setiap area adalah pada area merah 13, area hijau 8, area biru 79, dan area kuning 20. Sehingga didapatkan untuk *average pooling* seperti pada gambar 2.5[22].

3) *Normalization layer*

Lapisan normalisasi berguna untuk mengatasi variasi nilai yang signifikan pada *inputan*. Namun, lapis ini sudah jarang digunakan karena hasilnya yang relatif kecil bahkan tidak berdampak sama sekali ke dalam model pembelajarannya.

4) *ReLU layer*

ReLU atau *Rectified Linier Units* menerapkan fungsi :

$$f(x) = \max(0, x) \quad (2.2)$$

dimana nilai  $X$  merupakan *input* pada *activation* yang berarti aktivasi elemen dilakukan pada saat *pixel* berbentuk matriks diambang batas 0. Fungsi tersebut berguna untuk meningkatkan sifat ketidaklinearan fungsi keputusan dan keseluruhan jaringan tanpa memengaruhi *convolutional layer* Keuntungan dari ReLu adalah efisiensi dalam hal komputasi[16].

#### 5) *Fully connected layers*

*Fully connected layer* adalah lapisan terakhir pada CNN, cara kerjanya dengan mengirimkan data ke *output* lalu menyederhanakannya dan mempercepat proses perhitungan data. Persamaan *fully connected layers* dapat dituliskan dengan rumus

$$h(x) = g \tag{2.3}$$

dimana  $g$  merupakan fungsi aktivasi yang digunakan untuk menambahkan sifat non linearitas pada model sedangkan  $b$  merupakan bias atau nilai konstan yang dimasukan ke dalam  $\sum_i w_i x_i$ . Sedangkan  $x_i$  merupakan nilai dari fitur  $w_i$  sebagai bobot koneksi saraf,  $b$  sebagai bias dan  $g$  sebagai fungsi aktifasi[19].

#### 6) *Loss Layer*

*Loss layer* merupakan lapisan yang terakhir pada CNN. Lapisan ini digunakan untuk menentukan bagaimana hasil *training* memberikan hukuman dari penyimpangan antara hasil prediksi dan label. Ada beberapa variasi dari *loss function* di antaranya *softmax loss* digunakan untuk memprediksi salah satu dari keseluruhan kelas yang eksklusif, *sif-gmoid cross-entropy loss* digunakan untuk memprediksi dari jumlah nilai probabilitas dalam rentang [0,1], dan *Euclidean loss* digunakan sebagai regresi nilai berkelanjutan[16].

### 2.2.5 DETEKSI OBJEK (*OBJECT DETECTION*)

*Object Detection* merupakan bagian dari CNN yang digunakan dalam pendeteksian objek secara *realtime* oleh komputer setelah dilakukan proses *training* data yang besar. *Object detection* adalah proses mendeteksi dan mengenali berbagai objek dalam suatu gambar atau video dengan memilih posisi objek menggunakan

kotak pembatas. Pada *object detection* terdapat dua informasi mengenai klasifikasi, *confident rate* dan lokasi objek yang disebut dengan *localization*.



**Gambar 2.6 Hasil prediksi dan label pada gambar[21]**

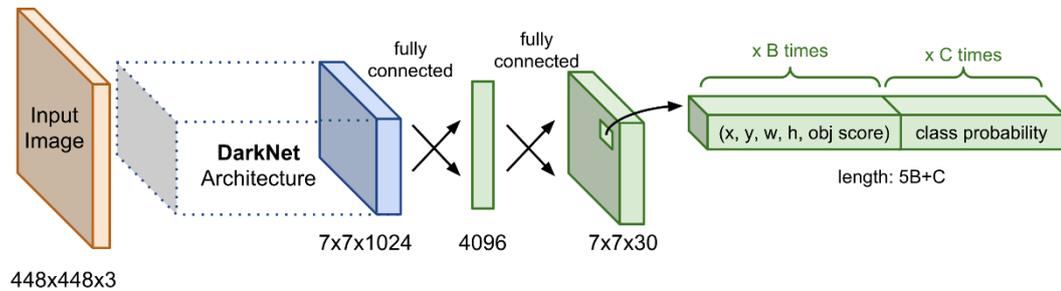
Pada Gambar 2.6 dalam mengaplikasikan *localization*, deteksi objek menggunakan *bounding box* yang pada setiap objek mempunyai label. Dalam *bounding box* terdapat 4 buah nilai yaitu informasi titik kordinat (x, y), dan ukuran dari *bounding box* (panjang dan lebar)[21]. Cara kerja pendeteksian ini menggunakan nilai *Intersection over Union* (IoU). IoU membandingkan nilai *actual* dan nilai *predicted*. Pada model objek deteksi YOLOV5 yang dibuat menggunakan nilai IoU sebesar 0.5, yang artinya apabila nilai threshold yang yang dihasilkan  $>0.5$  maka objek akan terdeteksi benar atau true positive. *Object detection* awalnya merupakan dari klasifikasi citra yang kemudian dikembangkan lebih lanjut sebagai pedeteksian objek. *Object detection* digunakan dalam penelitian ini, setiap model *object detection* memiliki kelebihan dan kekurangannya masing masing bergantung pada jenis data dan banyaknya data yang diproses[23].

Metode pendeteksian objek terbagi menjadi dua jenis, yakni metode konvensional dan metode pembelajaran mendalam. Beberapa tantangan pada saat pemrosesan gambar yaitu gambar yang diambil buram karena ketidakstabilan kamera, gambar tidak jelas karena adanya penghalang yang menutupi objek tersebut, kualitas gambar mungkin buruk akibat cahaya yang berlebihan atau resolusi dari kamera yang rendah[24].

### **2.2.6 YOLO**

Kemampuan dalam melakukan pendeteksian secara otomatis menggunakan AI sangat diperlukan. YOLO yang berarti “*You Only Look Once*” berarti sebuah gambar bisa diprediksi atau deteksi objek serta letaknya[19]. YOLO pertama kali

diciptakan pada tahun 2015 oleh Redmon[5]. YOLO merupakan metode pendeteksian objek yang populer karena karakteristik sistem deteksinya dan dapat digunakan secara *real-time*. Cara kerja YOLO yaitu hanya perlu memasukan gambar yang sudah ditandai dengan kotak pembatas ke dalam jaringan untuk mendapatkan hasil deteksi akhir[25]. YOLO hanya memiliki 1 buah CNN untuk melakukan *localization (bounding box)* dan klasifikasi objek[21].

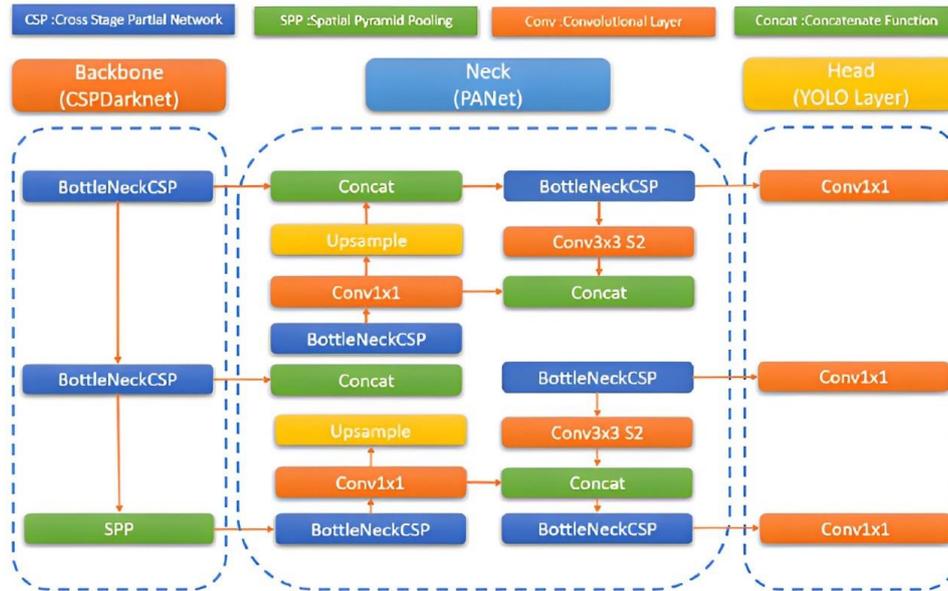


**Gambar 2.7** Arsitektur YOLO [19]

Pada Gambar 2.7 dijelaskan mengenai arsitektur dari YOLO pada *layer* pertama merupakan *input* gambar yang berukuran 448x448 pixel. Selanjutnya gambar tersebut diproses melalui beberapa lapisan konvolusi dan *pooling* yang sudah dikustomisasi dengan YOLO sehingga dapat digunakan dalam mengekstrak fitur-fitur pada gambar. Selanjutnya melewati lapisan *fully connected* yang berfungsi untuk menghasilkan prediksi deteksi dari sebuah objek. Selanjutnya diterapkan algoritma *Non Maximum Supressions* (NMS) yang digunakan untuk menghilangkan kotak yang *overlap* atau tumpang tindih sehingga hanya mempertahankan nilai deteksi yang tertinggi.

### 2.2.7 YOLOV5

YOLOv5 merupakan arsitektur dari object detection yang memiliki nilai akurasi yang lebih baik dari versi sebelumnya. YOLOv5 merupakan bagian dari CNN untuk object detection. CNN menggunakan arsitektur tiga dimensi, yaitu width, height, dan depth. YOLOv5 menggabungkan berbagai teknik modern seperti data augmentation, mosaic augmentation, dan auto-learning bounding box anchors untuk meningkatkan performa deteksi objek. Arsitektur ini juga lebih ringan dan lebih cepat dalam melakukan inferensi, menjadikannya ideal untuk aplikasi *real-time*[26].



**Gambar 2.8** Arsitektur YOLOV5[26]

Pada Gambar 2.8 dijelaskan mengenai arsitektur dari YOLOV5, Dimana terdiri dari 3 bagian yaitu *backbone* yang menggunakan *CSPDarknet* digunakan untuk meningkatkan efisiensi pemrosesan dan menurunkan kompleksitas model dengan memisahkan fitur yang diproses secara parsial dan menyatukannya kembali, *Neck* (PANet / *Path Aggregation Network*) digunakan untuk membantu dalam mempertahankan informasi yang lebih baik dari berbagai fitur dan menggabungkannya untuk meningkatkan akurasi deteksi, dan *Head* (YOLO layer) yang berfungsi untuk menghasilkan prediksi deteksi

Pada YOLOV5 menggunakan beberapa parameter performansi dalam *training* data diantaranya :

A. Akurasi

Keakuratan merupakan tolok ukur keandalan suatu sistem dalam mengkategorikan objek. Parameter ini mengukur ketepatan sistem dalam mengidentifikasi dalam data pengujian. Perhitungan akurasi dapat dilakukan dengan menggunakan persamaan

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.5)$$

B. Presisi

Presisi yaitu hasil dari jumlah objek yang dideteksi benar (*true positive*) dibandingkan dengan keseluruhan hasil deteksi yang terprediksi positif. Jadi semakin tinggi deteksi berupa *false positive* maka nilai presisi yang dihasilkan

akan semakin rendah, sebaliknya apabila semakin rendah nilai *false positive* maka nilai presisi semakin tinggi.

$$Precision = \frac{TP}{TP+FP} \quad (2.6)$$

### C. Recall

Recall yaitu nilai dari jumlah gambar yang terdeteksi benar (*true positive*) dibandingkan dengan seluruh data positif. Semakin tinggi nilai yang dihasilkan dari *recall* maka semakin bagus sistem dapat mengklasifikasikan suatu kelas dari objek yang terdeteksi.

$$Recall = \frac{TP}{TP+FN} \quad (2.7)$$

### D. Confidence Score

*Confidence* adalah representasi dari Intersection over Union (IoU) antara kotak prediksi dan kotak ground truth. Untuk mendapatkan prediksi final, faktor penentu utamanya adalah *class confidence score*, yang diperoleh berdasarkan probabilitas kondisional kelas dan *box confidence score*. *Class confidence score* mengukur tingkat keyakinan pada klasifikasi dan lokalisasi objek. *Class confidence score* memberikan nilai keyakinan untuk setiap kotak, yang mengkodekan kemungkinan kelas yang muncul di dalam kotak dan seberapa cocok kotak prediksi dengan objek. Jika tidak ada objek yang terdeteksi, maka nilai *confidence* adalah nol[27].

$$Confidence\ Score = Pr(Class) \times IoU \quad (2.8)$$

### E. mean Average Precision (mAP)

mAP merupakan nilai rata rata yang didapat dari *average precision* atau AP. *Metrics* inilah yang menilai kinerja dari suatu model objek deteksi. Sebelum mengukur mAP diperlukan adanya penyesuaian pada *threshold* pada IoU (*Intersection Over Union*). Nilai IoU pada metrik mAP model YOLOV5 dibuat sebesar 0.5 sehingga nilai pada metrik yang dihasilkan yaitu mAP@0.5[28].

Rumus AP terbagi menjadi dua bagian :

#### 1. Menghitung Presisi Interpolasi (P inter)

Rumus pertama digunakan untuk menghitung presisi interpolasi (P inter) pada nilai recall tertentu. Presisi interpolasi adalah nilai presisi maksimum yang dapat dicapai pada nilai recall tertentu. Rumus tersebut adalah :

$$P \text{ interpolasi}(r_{n+i}) = \max p(r'); r' \geq (rn + i) \quad (2.9)$$

Dimana :

$P \text{ inter}(rn+1)$  : Presisi interpolasi pada nilai recall  $rn+1$

$\max p(r')$  : Nilai presisi maksimum pada nilai recall  $r'$

$r'$  : Nilai recall yang lebih besar dari atau sama dengan  $rn+1$

Rumus ini menghitung nilai presisi interpolasi dengan mengambil nilai presisi maksimum yang dapat dicapai pada nilai recall yang lebih besar dari atau sama dengan nilai recall yang diberikan. Nilai ini digunakan untuk menghaluskan kurva presisi-recall yang berbentuk zig-zag.

## 2. Menghitung rata-rata Presisi

Rumus kedua digunakan untuk menghitung rata-rata presisi rata-rata (AP). AP merupakan metrik yang mengukur kinerja keseluruhan algoritma klasifikasi dalam memprediksi kelas objek. Rumus tersebut adalah :

$$AP = \frac{\sum (rn + i - rn) p \text{ inter}(rn + i)}{\sum (rn + i - rn)} \quad (2.10)$$

Dimana

AP : Rata-rata presisi rata-rata

$\sum (rn + i - rn)$  : Simbol penjumlahan

$(rn + i - rn)$  : Selisih antara nilai recall  $rn+1$  dan  $r'n$

$p \text{ inter}(rn + i)$  : Presisi interpolasi pada nilai recall  $rn+1$

Selanjutnya untuk menghitung nilai mAP yaitu dengan menjumlahkan hasil perkalian antara selisih nilai recall dan presisi interpolasi pada setiap nilai recall yang unik. Nilai recall yang unik adalah nilai recall di mana nilai presisi maksimum jatuh. Hasil penjumlahan ini kemudian dibagi dengan jumlah nilai recall yang unik.

$$mAP = \frac{\sum_{i=1}^N AP(i)}{N} \times 100\% \quad (2.11)$$

Dimana

mAP : Rata-rata mAP

$\sum$  : Simbol penjumlahan

AP(i) : Rata-rata presisi rata-rata (AP) untuk kelas ke-i

N : Jumlah kelas

Rumus ini menghitung mAP dengan menjumlahkan nilai AP untuk setiap kelas dan kemudian membaginya dengan jumlah kelas. mAP merupakan

metrik yang mengukur kinerja keseluruhan algoritma klasifikasi dalam memprediksi kelas objek pada banyak kelas[29].

### 2.2.8 CONFUSIONS MATRIX

Performansi model klasifikasi dapat dinilai melalui beberapa parameter, seperti *precision*, *recall*, mAP dan akurasi. Untuk menghitung nilai *precision*, *recall*, dan akurasi, digunakan beberapa istilah seperti *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Istilah-istilah ini kemudian disusun dalam sebuah matriks yang dikenal sebagai *confusion matrix*[20]. Penilaian akurasi biasanya tidak cukup hanya dengan menentukan model yang dirancang dengan cukup baik dan memiliki akurasi yang tinggi, akan tetapi akurasi pada sistem pengujianya harus bisa menghasilkan prediksi yang akurat berdasarkan klasifikasinya.

Confusion Matrix		Predicted			False Negative (FN)	Recall
		Class 1	Class 2	Class 3		
Actual	Class 1	A	B	C	B + C	$A/(A + B + C)$
	Class 2	D	E	F	D + F	$E/(D + E + F)$
	Class 3	G	H	I	G + H	$I/(G + H + I)$
	False Positive (FP)	D + G	B + H	C + F	Overall Accuracy = $A + E + I / (\text{Sum of red and green squares})$	
	Precision	$A/(A + D + G)$	$E/(B + E + H)$	$I/(C + F + I)$		

■ True positives ■ True Negatives ■ Misclassified cases ■ False Positives ■ False Negatives.

**Gambar 2.9 Confusions Matrix Multi Class**[30]

Pada Gambar 2.9 merupakan gambar dari *confusions matrix multi class* yang digunakan untuk menghitung keseluruhan akurasi dari model yang dibuat. Dari 3 kelas tersebut, masing masing kelas mempunyai karakteristik yang berbeda warna pada masing masing kelas mencerminkan nilai TP, TN, FP, FN, *misclassified cases*[30].

### 2.2.9 GOOGLE COLLABORATORY

*Google Collab* adalah sebuah layanan *cloud* gratis yang mendukung GPU yang memungkinkan pengguna untuk menjalankan sebuah pemrograman *python* dengan *zero configuration*, akses gratis ke GPU, dan berbagi dengan mudah.

*Google Collab* merupakan antarmuka yang interaktif yang memungkinkan pengguna menulis dan menjalankan kode programnya[31].



**Gambar 2.10** *Google Collab*olatory[31]

*Google Collab*olatory dari sisi *software*, menyediakan Sebagian besar *library* yang dibutuhkan, contohnya adalah *library TensorFlow, NumPy, Pandas*, dan lainnya. Dari sisi *hardware*, *google collab* menyediakan akses ke memori yang terhubung dengan *google drive*, berbagai jenis *prosesor* termasuk CPU, GPU, dan TPU, serta kapasitas RAM. *Google colab* memiliki jaminan server yang stabil sehingga tidak ada hambatan dalam pemrosesan data selama koneksi internetnya lancar[32].

#### **2.2.10** *OPEN COMPUTER VISION LIBRARY (OPENCV)*

*OpenCV* adalah sebuah *library* program yang bisa diakses secara bebas dan *open source*. Program ini digunakan untuk pengolahan citra yang bersifat dinamis dan dilakukan secara waktu nyata yang mempunyai lebih dari 2500 algoritma yang dioptimalkan.



**Gambar 2.11** *Library OpenCV*[33]

Pada gambar 2.11 merupakan *library OpenCV*. *Library* ini menggunakan metode yang memungkinkan suatu komputer dapat melihat suatu objek sama seperti manusia sehingga dapat melakukan identifikasi dan pengambilan keputusan berdasarkan objek yang sudah terdeteksi. Kelebihan dari *OpenCV* yaitu memiliki

*library* dokumen yang banyak, bekerja secara cepat untuk komputer yang menggunakan prosesor intel, proses komputasi lebih ringan, dan dapat bekerja secara *real-time*[33].

### 2.2.11 TENSORFLOW

*TensorFlow* adalah sebuah *library* perangkat lunak yang disediakan secara bebas dan *opensource* untuk pembelajaran mesin. Meskipun *library* ini dapat diterapkan dalam berbagai tugas, namun focus utamanya terletak pada pelatihan dan inferensi jaringan *neural*. *TensorFlow* mewakili sebuah *library* matematika yang simbolis yang berbasis pada *dataflow* dan pemrograman[34].

**Tabel 2.2 Struktur *toolkit TensorFlow***

Komponen API			Penjelasan
<i>Tensorflow Estimator</i>			API tingkat tinggi yang berorientasi objek
<i>tf.layers, tf.losses, tf.metrics</i>			Library yang dapat digunakan kembali sebagai komponen model umum
<i>Python TensorFlow</i>			Menyediakan OP, yang membungkus oleh kernel C++
<i>C++ TensorFlow</i>			Memaksimalkan efisiensi dan performa <i>Tensorflow</i>
CPU	GPU	TPU	Kernel yang berfungsi pada satu platform atau lebih

Tabel 2.2 merupakan struktur *toolkit* dari *tensorflow* yang menggambarkan komponen API dari *tensorflow* dan memiliki peran untuk mengeksekusi model dari *tensorflow* tersebut. *Tensorflow estimator* berfungsi untuk menyederhanakan proses membangun, melatih, dan mengevaluasi model *machine learning*. Selanjutnya *library tf.layers* digunakan untuk membuat lapisan *neural network*, *tf.losses* digunakan untuk menentukan fungsi *loss* yang digunakan selama pelatihan model, *tf.metrics* digunakan untuk mengukur performa model. *Python TensorFlow* berfungsi untuk mendefinisikan dan menjalankan operasi di *tensorflow*. *C++* berfungsi untuk memaksimalkan efisiensi dan performansi dari *tensorflow*. Sedangkan CPU, GPU, dan TPU merupakan *platform* perangkat keras yang digunakan pada *tensorflow*[34].

### 2.2.12 PYTORCH

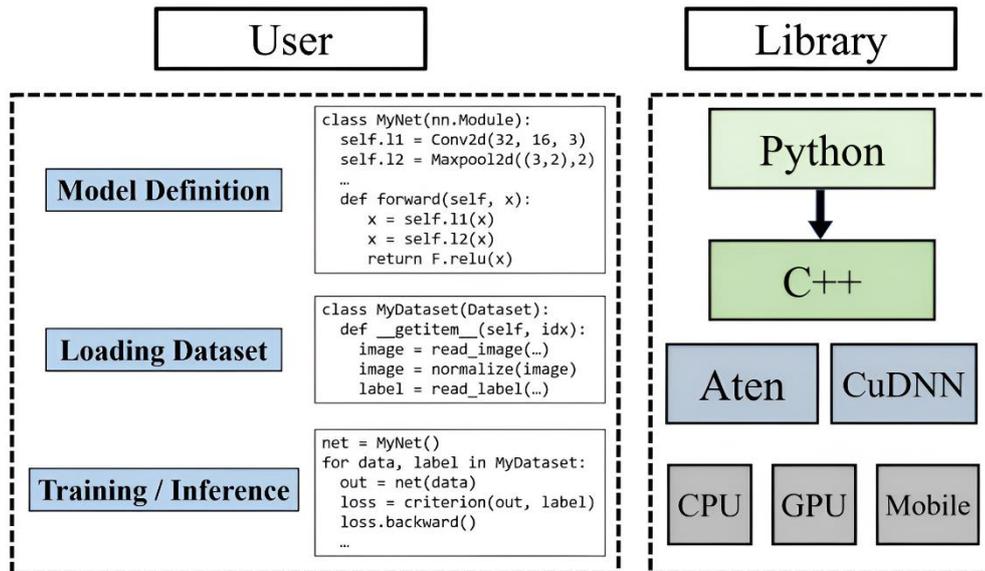
*Pytorch* adalah *library* dalam Bahasa pemrograman *python* yang digunakan untuk melakukan komputasi dalam DL. *User* dapat melakukan komputasi pada DL

dengan objek seperti gambar dan video menggunakan *pytorch* dikombinasikan dengan *torchvision*.



**Gambar 2.12 Library Pytorch**[35]

Pada gambar 2.12 merupakan *library pytorch* yang sering digunakan dalam analisis citra pada CNN. *Library* ini memanfaatkan *neuron kortik* dalam melakukan analisisnya. *Torchvision* mendukung berbagai arsitektur CNN, sehingga sangat bagus digunakan untuk pemodelan berjenis CNN[35].



**Gambar 2.13 Library DL berbasis Pytorch**[36]

Paada Gambar 2.13 merupakan *library DL* berbasis *pytorch*. Pengembang dalam bidang DL biasanya menggunakan *library* dalam mendukung *training* dan inferensi pada DNN. Secara teoritis DNN terdiri dari beberapa lapisan. Pada bagian definisi model, *Conv2d* dan *Maxpool2d* adalah API yang dipanggil untuk menambahkan dua lapisan ke dalam contoh DNN. Kemudian fungsi *forward* menjelaskan bagaimana data *input* harus mengalir dalam lapisan yang telah ditentukan. Sebelum pelatihan dan inferensi yang sebenarnya, dataset juga harus dimuat pada awal pemrosesan yang diperlukan, misalnya *torchvision.transforms.normalize* adalah langkah penting dalam awal pemrosesan data, yang bertujuan untuk mengubah skala nilai variabel *input* dan target untuk

kinerja yang lebih baik. *Training* adalah proses bagi DNN untuk mempelajari cara melakukan tugas. Sebagai contoh, pada klasifikasi gambar, dengan memberi *input* pada DNN dengan data yang diketahui dan label yang sesuai, kita dapat memperoleh model DL yang terlatih[36].

### **2.2.13 KARDUS BERJENIS *CORRUGATED BOX***

Kardus merupakan wadah atau pelindung yang biasanya digunakan pada kegiatan industri. Kardus berjenis *corrugated box* ini terbuat dari bahan karton yang bergelombang dan memiliki lapisan tipis sebagai permukaannya. Karena bobotnya yang ringan, praktis dan nyaman digunakan, maka kardus dengan jenis *corrugated box* ini banyak digunakan untuk wadah atau *packaging* dalam dunia industri dan logistik.



**Gambar 2. 14 Kardus Berjenis *Corrugated Box***[37]

Pada Gambar 2.14 merupakan kardus kemasan berjenis *corrugated box*. Kualitas kardus dapat dilihat dari visualnya, baik permukaannya maupun teksturnya. Kardus dikatakan tidak rusak apabila mempunyai permukaan yang sempurna tanpa cacat, sedangkan kardus dikatakan rusak apabila memiliki cacat fisik pada bagian kardus seperti basah/terkena cairan, berlubang, dan terjadi penekukan. Kardus dikatakan [38].

### **2.2.14 LOGISTIK**

Logistik adalah proses pengelolaan aliran barang atau jasa dari sumber hingga tujuan, mencakup perencanaan, pelaksanaan, dan pengendalian yang efisien dan efektif dari barang atau jasa serta informasi terkait, dari titik asal hingga titik penggunaan, untuk memenuhi kebutuhan pelanggan. Di era revolusi industri 4.0, teknologi informasi memainkan peran yang sangat penting dalam mengubah tata cara dalam industri dan bisnis. Terutama dalam cara pelaku bisnis menawarkan produk dan jasa mereka. Industri logistik diperkirakan akan terus berkembang

dalam beberapa tahun mendatang berkat pertumbuhan e-commerce. Teknologi AI diharapkan dapat dikembangkan dalam berbagai aspek pada dunia logistik. Salah satu penerapan teknologi AI dalam dunia logistik yaitu dengan menerapkan *computer vision* untuk melakukan proses deteksi pada kualitas kemasan berjenis kardus pada konsumen sehingga selain menerima barang, konsumen juga dapat melakukan *checking* pada kualitas kemasannya, hal ini akan memiliki dampak positif sehingga pihak penjual maupun jasa kirim lebih berhati-hati terhadap pengelolaan barang[39].