Pengembangan Aplikasi Klasifikasi Alat Transportasi Berdasarkan Citra Digital untuk Pencatatan Aset Studi Kasus: PT. Pulo Mas Jaya

Muhammad Fikry

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav. 22, Jakarta 13210
Email: mhmmadfikry31@gmail.com

Abstract: *This research aims to develop desktop-based software that is useful for classifying digital images of transportation equipment for asset recording using the Convolutional Neural Network method to classify image data. The data used as training data is 15000 data which is divided into 3 data groups. While the test data used in this study were 3000 data. Theconvolutional neural network method is implemented using the TensorFlow software library. Inthis study, using the incremental model of software development. This incremental stage is divided into two stages, namely the first iteration focuses on making the classification model of transportation equipment and the second iteration focuses on making GUI-based applications. From the research results, iteration one produces a model with an accuracy value of 92.17%. for training data at 40 epochs and 92.16% on test data. Then, iteration two creates a desktop- based user interface built using the Tkinter framework.*
Keywords: *Classification, Convolutional Neural Networks, Accuracy, Transportation Equipment, tkinter.*

Abstrak: *Penelitian ini dibuat bertujuan untuk mengembangkan perangkat lunak berbasis desktop yang berguna untuk mengklasifikasikan alat transportasi berdasarkan citra digital untuk pencatatan aset menggunakan metode Jaringan Saraf Konvolusional untuk mengelompokkan data gambar. Data yang digunakan sebagai data latih sebanyak 15000 data yang terbagi dalam 3 kelompok data. Sedangkan data uji yang digunakan dalam penelitian ini sebanyak 3000 data. Metode jaringan saraf konvolusi diterapkan menggunakan pustaka perangkat lunak TensorFlow. Pada penelitian ini, menggunakan pengembangan perangkat lunak model inkremental. Tahapan inkremental ini dipecah menjadi dua tahapan, yaitu iterasi satu berfokus pada pembuatan model klasifikasi alat transportasi dan iterasi dua berfokus padapembuatan aplikasi berbasis GUI. Dari hasil penelitian, iterasi satu menghasilkan model dengan nilai akurasi sebesar 92,17%. untuk data latih pada 40 epoch dan 92,16% pada data uji.Kemudian, iterasi dua membuat antarmuka pengguna berbasis dekstop yang dibuatmenggunakan framework Tkinter.*
Kata kunci: *Klasifikasi, Jaringan Saraf Konvolusional, Akurasi, Alat Transportasi, tkinter.*

I. PENDAHULUAN

Saat ini perkembangan dan penggunaan teknologi ialah beralihnya ke data paperless yang dapat juga disebut transformasi Tentu saja, tujuan akhir dari transformasi digital adalah untuk memberikan layanan yang lebih efisien kepada pengguna dan menggantikan proses tradisional yang sekarang sudah usang. Salah satu contoh transformasi digital ada pada PT. Pulo Mas Jaya pada data aset inventaris sebuah perusahaan, aset yang dimaksud ialah alat transportasi yang digunakan pegawai perusahaan tersebut di mana perlunyadata kendaraan beserta citra fisiknya. Perlunya data alat transportasi ini merupakan bentuk proses pencatatan aset pada PT. Pulo Mas Jaya yang dilakukan sekali setiap tahunnya.

Proses pencatatan aset alat

transportasi pada PT. Pulo Mas Jaya dapat dikatakatan semi manual, dimana prosesnya pada departemen arsip melihat seluruh data aset inventaris alat transportasi dalam bentuk excel yang selanjutnya departemen arsip melakukan pencarian aset tersebut, pencarian aset dapat dilakukan dengan meminta gambar kepada pemilik asetatau dapat mengambil gambar secara mandiri dengan mencari aset tersebut ke lapangan langsung. Setelah gambar aset didapatkan yang setidaknya berjumlah puluhan aset alat transportasi tersebut, departemen arsip melakukan klasifikasi secara manual dimana melihat satu per satu dari seluruh gambar tersebut dan memisahkan gambar sesuai dengan jenis alat transportasi tersebut. Setelah klasifikasi secara manual dilakukan, tahap terakhir melakukan pembuatan *qr-code* untuk setiap gambar alattransportasi tersebut.

Melihat proses yang terjadi tersebut, dengan perkembangan teknologi padasaat ini memungkinkan melakukan klasifikasi alat transportasi berdasarkan citra digital dengan menggunakan komputer.

Machine learning adalah ilmu yang berfokus pada proses pembelajaran komputer agar berfungsi seperti otak manusia dengan belajar sendiri dari waktu ke waktu. Tugas utama *machine learning* adalah meningkatkan kemampuan perangkat untuk mempelajari petunjuk baru dari data dan memperluas kemampuan mesin untuk memecahkan masalah. Pemelajaran Dalam adalah bagian atau subset dari *Machine Learning*, yang pada dasarnya adalah Neural Network dengan tiga atau lebih lapisan. Pemelajaran dalam ini dapat dilakukan pada klasifikasi citra digital, di mana *image classification* merupakan salah satu metode dari Pemelajaran Dalam yang dapat dengan cepat mengenali suatu gambar [1].

Image classification sangat dibutuhkan khususnya dibidang informatika. Bagaimana caramenduplikasi selayaknya manusia dalam memahami informasi gambar sehingga komputer dapat mengenaliobjek dalam gambar. Kesulitan yang dihadapkan adalah perbedaan antar gambar, seperti sudut pandang yang berbeda, perbedaan skala, kondisi pencahayaan yang berbeda, dll.

Merujuk pada penelitian yang dilakukan oleh Gramandha Wega Intyanto dengan klasifikasi citra bunga dengan menggunakan pemelajaran dalam: CNN (Jaringan Saraf Konvolusional) mengatakan teknik yang digunakan oleh CNN adalah metode *backpropagation* (pembelajaran) dan metode *feedforward* (klasifikasi). CNN menggunakan dua tahap atau lapisan, lapisan ekstraksi fitur (*feature extraction layer*) dan klasifikasi (*fullyconnected layer*). Pada biasanya *feature extractionlayer* memiliki prosedur konvolusi dan *downsampling (pooling)*, dan *Fully Connected Layer* memiliki proses *flatten*, *hidden layer* dan *output layer* [2]. Pada *convolution*, fungsi aktivasidilakukan menerapkan *ReLU* dan tahap *downsampling* menerapkan *max pooling*.

Penelitian ini menggunakan algoritma Jaringan Saraf Konvolusional untuk mengklasifikasi alat transportasi berdasarkan citra digital untukpencatatan aset. Jaringan SarafKonvolusional bekerja sangat baik dibanding dengan algoritma lain. Jaringan Saraf Konvolusional merupakan salah satu algoritma dari Pemelajaran Dalam yang dapat disebut pengembangan *Multi Layer Perceptron* (MLP). Untuk klasifikasi citra, MLP tidak cocok dan menghasilkan hasil yang buruk karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel sebagai fitur independen [3].

Tujuan dari penelitian ini ialah pengembangan aplikasi klasifikasi alat transportasi berdasarkan citra digitaluntuk pencatatan aset menggunakan algoritma CNN dengan pustaka perangkat lunak *TensorFlow* dan Mengetahui tingkat akurasi yang didapatkan dari hasil klasifikasi menggunakan algoritma CNN dengan pustaka perangkat lunak

TensorFlow.

Berdasarkan latar masalah dan penelitian sebelumnya yang telah dipaparkan, maka peneliti mengambil judul penelitian "PengembanganAplikasi Klasifikasi Alat TransportasiBerdasarkan Citra Digital UntukPencatatan Aset Studi Kasus: PT. Pulo Mas Jaya". Dengan adanya aplikasi tersebut, diharapkan identifikasi alat transportasi dapat lebih akurat dan efisien.

II. METODE PENELITIAN

A. Teori Pendukung

1. *Deep Learning*

Deep Learning (Pemelajaran Dalam) adalah bagian atau subset dari *Machine Learning*, yang pada dasarnya adalah*Neural Network* dengan tiga atau lebih lapisan. *Neural network* ini mencoba untuk mensimulasikan perilaku otak manusia, meskipun jauh dari menyamai kemampuannya (otak), memungkinkannya untuk belajar dari sejumlah data yang besar [1]. Sementara *Neural network* dengan satu lapisan masih dapat membuat perkiraan, *hidden layer* tambahan dapat membantu mengoptimalkan dan meningkatkan akurasi. Pemelajaran Dalammembedakan dirinya dengan *Machine Learning* berdasarkan jenis data yang digunakannya dan metode pembelajarannya [8]. Algoritma *machine learning* membuat prediksimenggunakan data terstruktur dan berlabel. Artinya, karakteristik tertentu ditentukan dari data input ke model dan diorganisasikan ke dalam tabel. Jika data tidak terstruktur, beberapa pra-pemrosesan biasanya dilakukan untuk mengatur data dalam format terstruktur. Algoritma Pemelajaran Dalam beradaptasi dan mencocokkan dengan tepat. Serta dapat memproses data tidak terstruktur dalam format mentah (teks, gambar, dll.) dan secara otomatis menentukan serangkaian fitur yang membedakan berbagai kategori data. Jeff Dean adalah Pakar *Google*, Anggota *Senior Grup Sistem* dan Infrastruktur *Google*, yang terlibat dalam penskalaan dan penerapan Pemelajaran Dalam di *Google*. Jeff dikenal terlibat dalamproyek *Google Brain* dan pengembangan perangkat lunak Pemelajaran Dalam skala besar seperti *DistBelief* dan kemudian *TensorFlow*. Kemudian ia juga berpendapat bahwa istilah Pemelajaran Dalam ialah dengan memikirkan jaringan saraf dalam yang besar. *Deep* biasanya mengacu pada jumlah lapisan dan karena itu merupakan istilah umum yang digunakan di media. Jeff menyimpulkan dengan menganggapnya sebagai *deep neural network.*

Neural Network datang dalam berbagai format, termasuk *recurrent neural networks*, jaringan saraf konvolusional, *artificial neuralnetworks* dan *feedforward neural networks*, masing-masing memilikimanfaat dengan kasus penggunaan tertentu. Namun, semuanya bekerja dengan cara yang sama, di mana memasukkan data dan membiarkan *model* mencari tahu sendiri apakah telah membuat interpretasi atau keputusan yang tepat tentang *elemen* data yang diberikan.

2. *Convolutional Neural Network*

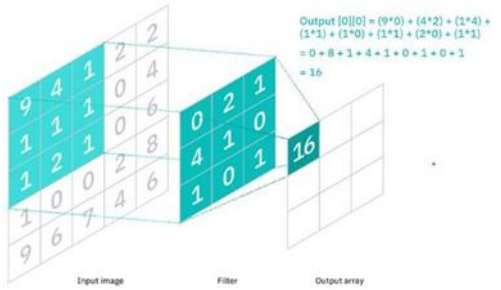
Convolutional Neural Network (Jaringan Saraf Konvolusional) adalah metode perkembangan dari *Artificial Neural Network* (ANN) yang mana mempelajari pola dengan stimulasi *neuron* untuk struktur *3D*. CNN dirancang untuk mengekstrak informasidari jumlah data yang lebih besar daripada ANN. Secara umum proses pembelajaran pada CNN sama dengan proses pada ANN, terbagi menjadi tiga lapisan utama yaitu lapisan masukkan, lapisan tersembunyi, dan lapisan luaran.Proses perubahan nilai bobot pada CNNsama dengan proses pada ANN. Artinya, ia menggunakan metode *backpropagation*. [5]

Setelah mendapatkan feature *vector* dari citra, citra dapat

dideskripsikan sebagai *vector of fixed length*, kemudian diperlukan pengklasifikasi untuk mengklasifikasikan feature *vector* tersebut [6]. CNN digunakan untukmengklasifikasikan data yang sudah diberi label. Metode CNN adalah pembelajaran *supervised*. Algoritma ini bertujuan untuk mengelompokkan data berdasarkan data yang ada. Arsitektur CNN yang khas terdiri dari*Convolutional Layer, Pooling Layer*, dan *Fully Connected Layer* [4].

3. Convolutional Layer

Convolutional layer adalah adalah komponen dari Jaringan Saraf Konvolusional yang sangat penting dimana mengekstrak fitur dari gambar dengan ukuran kernel konvolusi yang berbeda [4]. Setelah beberapa konvolusi, *set feature map* dapat diekstraksi dari gambar input. Tujuan dari konvolusi adalah untuk mengekstraksi citra masukan.



Gambar 1 Proses Convolution Layer

Menurut Gambar 1 setiap nilai keluaran pada feature map tidak harus terhubung ke setiap nilai piksel pada gambar masukan. Itu hanya perlu terhubung ke bidang reseptif, tempat filter diterapkan. Karena array keluaran tidak perlu memetakan secara langsung ke setiap nilai masukan, layer convolutional dan pooling biasanya disebut sebagai layer partially connected. Namun, karakteristik ini juga dapat digambarkan sebagai konektivitaslokal [7]

4. *Pooling Layer*

Secara umum, posisi *pooling layer* adalah setelah *convolutional layer*. Fungsi dari *pooling layer* adalah untuk mengurangi nilai sampel(*downsampling*), mempercepat komputasi, dan mengatasi masalah *overfitting* [4]. Ada dua *layer pooling* yang umum digunakan: *max pooling* dan *average pooling*. *Max pooling* mencari nilai maksimum dalam rentang tertentu, sedangkan *average pooling* mencari rentang karakteristik tertentu dengan menemukan rata-rata. Mirip dengan convolutional layer, proses pooling menyapu filter di seluruh masukkan, tetapi perbedaannya adalah bahwa filter ini tidak memiliki bobot. Sebagai gantinya, kernel menerapkan fungsi agregasi ke nilai dalam bidang reseptif, mengisi array keluaran [10].

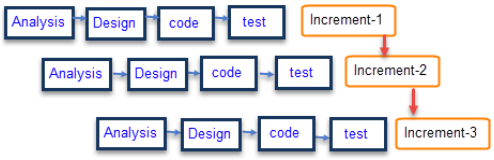
5. *Fully-Connected Layer*

Nilai piksel dari gambar masukkan tidak terhubung langsung ke lapisan keluaran di layer *partially connected layers*. Namun, pada lapisan *fully- connected*, setiap simpul di lapisan keluaran terhubung langsung ke simpul di lapisan sebelumnya. Lapisan ini melakukan tugas klasifikasi berdasarkan fitur yang diekstraksi melalui lapisan sebelumnya dan filter yang berbeda. Sementara lapisan *convolutional* dan *pooling* cenderung menggunakan fungsi *ReLu*, lapisan *fully-connected* biasanya memanfaatkan fungsi aktivasi *softmax* untuk mengklasifikasikan input dengan tepat, menghasilkan probabilitas dari 0 hingga 1 [7].

6. Metode Pengembangan Perangkat Lunak

Pada penelitian ini, proses pengembangan perangkat lunak mengikuti metodologi *Software Development Life Cycle* (SDLC) dengan menggunakan model inkremental. SDLC adalah serangkaian proses yang bertujuan untuk membuat perangkat lunak. Alasan memilih model inkremental adalah karena

memiliki kelebihan sebagai berikut: Proses pengujian dan debugging yang mudah, kemungkinan kegagalan proyek yang rendah, pembuatan perangkat lunak sesuai permintaan yang cepat [9]. Berikut tahapan yang akan dilakukanpada model inkremental ini yaitu: analisis, desain, implementasi danpengujian.

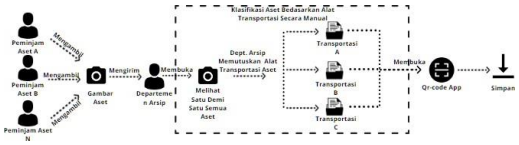


Gambar 2 Proses Metode Inkremental

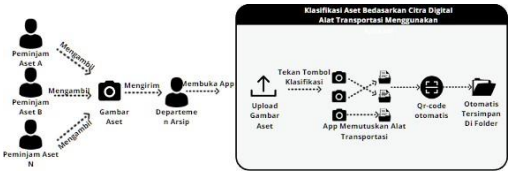
Model Inkremental adalah proses pengembangan perangkat lunak di mana persyaratan dipecah menjadi beberapa modul mandiri dari siklus pengembangan perangkat lunak [10]. Berdasarkan Gambar 2 Analisis adalah analisis kebutuhan dan spesifikasi data, kebutuhan perangkat lunak dan perangkat keras. Desain berarti merancang proses dan fungsi agar tujuan yang ingin dicapai sesuai dengan harapan. Kode adalah proses implementasi dari tahap desain dalam bahasa pemrograman. *Test* untuk menguji program memastikan bahwa program telah berjalan dengan baik.

B. Kronologis Penelitian

Sebagai langkah awal dalam pengembangan aplikasi, kerangka pemikiran menjadi model konseptual dari teori-teori terkait dari berbagaifaktor yang diidentifikasi sebagai isu penting yang dapat digambarkan pada Gambar 3 dan Gambar 4.



Gambar 3 Sebelum Menggunakan Aplikasi



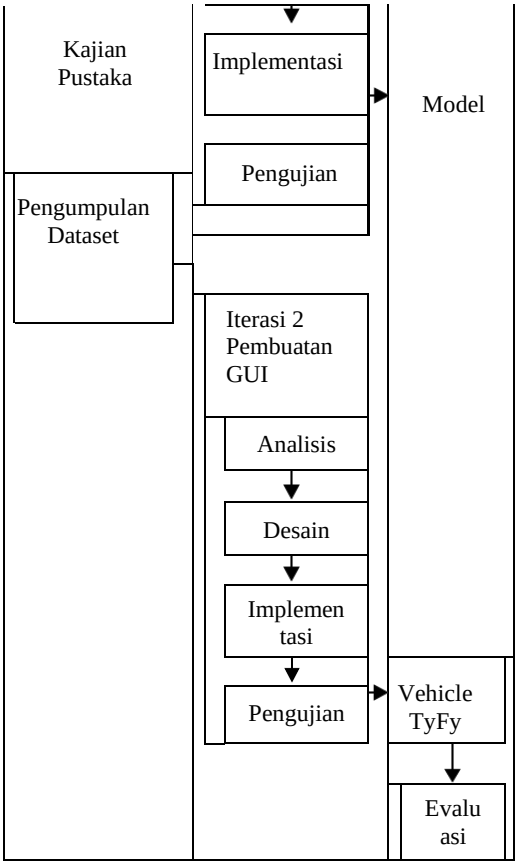
Gambar 4 Sesudah Menggunakan Aplikasi

Pada Gambar 3 adalah bentuk alur dari sebuah kerangka proses mengelompokkan alat transportasi yang sederhana, dimulai dari peminjam aset mengambil foto dari aset yang dimiliki, kemudian diberikan kepada bagian departemen arsip yang selanjutnya melakukan klasifikasi alat transportasi berdasarkan citra digital untukpencatatan aset sesuai dengan alat transportasi dengan melihat satu demi satu gambar aset yang telah dikirim kemudian memutuskan alat transportasi aset tersebut, setelah citra digitaldiklasifikasi, citra digital dimasukkan kedalam *qr-code* app untuk mendapatkan *qr-code* sesuai dengan citra digital dan terakhir menyimpan hasil generator *qr-code*.

Pada Gambar 4 adalah bentuk alur dari kerangka pemikiran klasifikasi alat transportasi berdasarkan citra digital untuk pencatatan aset menggunakan aplikasi, dimulai dari peminjam aset mengambil foto dari aset yang dimiliki, kemudian diberikan kepada bagian departemen arsip yang selanjutnya bagian departemen arsip menggunakan aplikasi yang dibuat pada penelitian ini.di mana dept. Arsip mengupload gambaraset lalu menekan tombol klasifikasi akan otomatis menentukan alat transportasi aset tersebut beserta pembuatan *qr-code* yang akan tersimpandi direktori. Desain Penelitian

C. Desain Penelitian

Pra-Penelitian	Penelitian	Hasil Penelitian
Observasi	Iterasi 1 Pembuatan Model	
Identifikasi Masalah	Analisis	
	Desain	



Gambar 5 Tahapan Penelitian

Seperti terlihat pada Gambar 5 Tahapan Penelitian, tahapan ini dimulai dengan 3 tahapan utama: tahapan pra-penelitian, tahapan penelitian, dantahapan hasil penelitian. Ketiga proses tersebut dapat dijelaskan satu demi satu sebagai berikut:

Fase pertama pra-penelitian dimulai dengan observasi dilakukan untuk mengumpulkan data yang dibutuhkan untuk penelitian dan pengembangan aplikasi dan lalu mengidentifikasi masalah untuk mengetahui batasan atau poin apa saja yang menjadi landasan untuk diuraikan yang selanjutnya melakukan kajian pustaka atau studi literatur mengenai pengolahan citra serta metode dalam melakukan klasifikasi gambar dan melakukan pengumpulan dataset.

Fase kedua yaitu penelitian dimulai dengan menggunakan model pengembangan perangkat lunak *inkremental* untuk mendukung proses pengembangan perangkat lunak yang dilakukan. Model *inkremental* dipilih karena mudah dikelola, memiliki risiko pengembangan sistem yang rendah, dan mengutamakan fitur-fitur utama sistem, sehingga tidak perlu menunggu semua persyaratan. Pada penelitian ini menggunakan dua proses inkremen (iterasi), di mana proses iterasi pertama melakukan pengolahan data dan pembuatan model, kemudian prosesiterasi kedua melakukan pembuatan tampilan sederhana aplikasi *GUI* untuk *user*.

Fase ketiga yaitu hasil penelitian yang berisi model, aplikasi dan evaluasi dari proses iterasi yang telah dilakukan sebelumnya.

D. Iterasi Satu

Pada tahap iterasi satu membahas tentang pembuatan model jaringan saraf konvolusional, di mana dengan harapan dapat melakukan prediksi sebagai inti perangkat lunak. Tahap iterasi satuterdiri dari analisis kebutuhan perangkat penelitian, desain pembuatan model, implementasi desain ke dalam bentuk *code*, dan terakhir pengujian terhadap model aplikasi yang dibuat.

Pada proses ini dilakukan untuk analisis kebutuhan data dan metode penelitian yang akan digunakan. Data yang digunakan pada penelitian ini adalah dalam bentuk citra, berdasarkan studi literatur yang telah dilakukan peneliti menggunakan *dataset Cifar-10*. Peneliti melakukan *subset dataset* di mana peneliti dapat memilih class yang dibutuhkan, *dataset* telah dipisah akan dilakukan *pre-processing.* Tahapan *pre-processing* ini dilakukan untukmengubah ukuran gambar dan normalisasi nilai *array*. Setelah tahap *pre-processing*, maka didapatkannya fitur-fitur citranya yang akan digunakan pada proses pengklasifikasian. Metode pengklasifikasian yang digunakan pada penelitian ini adalah metode jaringan saraf konvolusional.

Setelah hasil model klasifikasi

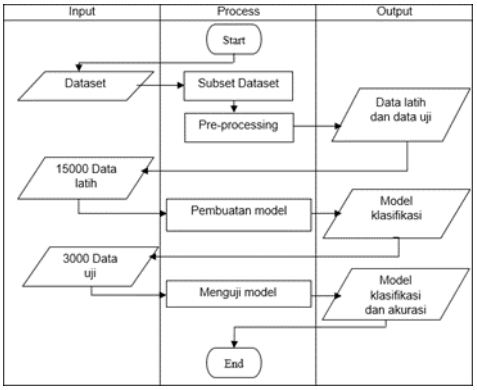
dari *library TensorFlow Keras* yang menerapkan algoritma klasifikasi Jaringan Saraf Konvolusional didapatkan, tahap selanjutnya peneliti menguji model klasifikasi. Setelah model diuji, peneliti selanjutnya menghitung nilai akurasi. Nilai akurasi diambil dari hasil model klasifikasi.

Dalam melakukan penelitian ini, penelitian menggunakan perangkat yang mendukung pengembangan aplikasi. Perangkat yang digunakan dalam proses pengembangan aplikasi ini ditunjukkan pada Tabel 1.

Tabel 1 Perangkat Pengembangan Aplikasi

Perangkat Lunak	***Perangkat Keras***
Google Colaboratory	*8 GB RAM Memory*
Windows 10 64-bit	*AMD Radeon 7700 HD Seroes*
Python 3.7.12	*i3-4130 ~3.4GHz*
NumPy 1.21.5	*HDD 256 GB*
Matplotlib 3.2.2	
Tensorflow 2.8.0	
Keras 2.8.0	

Pada proses ini dilakukan untuk analisis kebutuhan data dan metode penelitian yang akan digunakan. Data yang digunakan pada penelitian ini adalah dalam bentuk citra, berdasarkan studi literatur yang telah dilakukan peneliti menggunakan *dataset Cifar-10*. Peneliti melakukan *subset dataset* di mana peneliti dapat memilih class yang dibutuhkan, *dataset* telah dipisah akanTahap selanjutnya dilakukan untuk membuat desain pembuatan model dari analisis iterasi satu. Tahapan pertama memasukkan *dataset* yang sudah didapatkan dari hasil studi literatur, kemudian *dataset* tersebut memasuki tahap subset *dataset* yaitu memilih *class* yang diinginkan, kemudian melakukan *pre-processing*. Pada tahap ini, data latih di *resize* dan normalisasi nilai *array*.Data uji tersebut akan memasuki tahap *pre-processing* seperti data latih. *Output*yang dihasilkan kemudian dilakukan pembuatan model menggunakan metode CNN. Setelah mendapat model klasifikasi, data latih dan *testing* akan mendapatkan hasil akurasi dari model klasifikasi.



Gambar 6 Desain Pembuatan Model

Tahap selanjutnya merupakan prosespenejermahan kerangka pemikiran yangtelah disusun pada tahap sebelumnya kedalam bahasa koding. Penelitian ini menggunakan bahasa pemrograman *Python.* Menggunakan library*TensorFlow Keras* untuk menerapkanalgoritma CNN, library *Numpy* digunakan untuk melakukan operasi matematika, terutama saat melakukan perhitungan yang melibatkan vektor*;* matriks; dan *tensor*, dan library *Matplotlib* untuk visualisasi grafik vektor. Berikut adalah kode yang berperan sebagai *library dataset* yang digunakan dalam penelitian ini.

```
#Load Dataset
from keras.datasets import cifar10

#Komputasi Numerik
import numpy as np

# Untuk Visualisasi Hasil
import matplotlib.pyplot as plt
(X_train,Y_train),(X_test,Y_test)=cifar10.load_data()
```

Gambar 7 Kode Program Inisialisasi LibraryUntuk Dataset

Pada Gambar 7 Kode Program Inisialisasi Library untuk *Dataset*, menggunakan library *Keras* untuk mengambil dataset dan library *Matplotlib* untuk visualisasi contoh citra digital di

dataset. Pada Gambar 8 merupakan kode program untuk subset *dataset* yaitu memilih class yang diinginkan peneliti. Pada Gambar 9 digunakan untuk mengetahui jumlah data latih dan testing pada *dataset.*

```
X_train = X_train[np.isin(Y_train, [1,8,9]).flatten()]
Y_train = Y_train[np.isin(Y_train, [1,8,9]).flatten()]
X_test = X_test[np.isin(Y_test, [1,8,9]).flatten()]
Y_test = Y_test[np.isin(Y_test, [1,8,9]).flatten()]
```

Gambar 8 Kode Program *Subset Dataset*

```
print(X_train.shape, Y_train.shape, X_test.shape, Y_test.shape)
```

Gambar 9 Kode Program Pengecekkan JumlahIsi *Dataset*

```
plt.imshow(X_train[1])
```

Gambar 10 Kode Program Menampilkan Contoh Isi *Dataset*

Gambar 10 Kode Program Menampilkan Contoh Isi *Dataset*, melalui plot gambar dari *dataset* untuk memvisualisasikan *dataset*.

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
```

Gambar 11 Program Inisialisasi Library Untuk Proses Pemodelan CNN

Pada Gambar 11 merupakan kode program untuk menginisialisasi library yang dibutuhkan untuk melakukan pelatihan model, di mana menggunakan library *TensorFlow Keras*. *Library* tersebut akan digunakan untuk memudahkan dalam tahapan pengembangan aplikasi.

```
#Pre-Processing
#Convert dari integer ke floats
X_train=X_train.astype('float32')
X_test=X_test.astype('float32')

#Normaliasi
X_train=X_train/255.0
X_test=X_test/255.0
```

Gambar 12 Kode Program *Pre-Processing* Data

Gambar 12 merupakan kode programuntuk melakukan *pre-processing.* Gambar akan dilakukan resize menjadi ukuran 32 x 32. Selanjutnya, nilai array gambar akan dinormalisasi dengan mengkonversi nilai sehingga beradadalam kisaran 0 dan 1.

```
#One-hot encoding
Y_train=np_utils.to_categorical(Y_train)
Y_test=np_utils.to_categorical(Y_test)

num_classes=Y_test.shape[1]
```

Gambar 13 Kode Program One-Hot Encoding

Setelah pre-processing gambar dilakukan, Gambar 13 merupakan kode program untuk *pre-processing* label, di mana perlunya convert *categorical data*ke *numerical data*. Penelitian menggunakan teknik *one-hot encoding* yang melakukan variabel yang dikodekan integer dihapus dan variabel *biner* baru ditambahkan untuk setiap nilai integer unik.

```
#Nembuat struktur model CNN
model=Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(32, 32, 3)))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D((2, 2)))
#Flatten
model.add(Flatten())
model.add(Dropout(0.8))
model.add(Dense(512,activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Gambar 14 Kode Program Pembuatan Arsitektur CNN Data Pelatihan

```
#Konfigurasi Optimizer
sgd=SGD(lr=0.01,momentum=0.9,decay=(0.01/25),nesterov=False)

#Compile Model
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

Gambar 15 Kode Program Untuk Konfigurasidan Compile Model

```
#Deksripsi model
model.summary()
```

Gambar 16 Kode Program MenampilkanDeskripsi Model

```
#Proses Training
x = model.fit(X_train,Y_train,
    validation_data=(X_test,Y_test),
    epochs=40)
```

Gambar 17 Kode Program Pembelajaran Model Klasifikasi Data Pelatihan

Data latih akan digunakan melatih algoritma pembelajaran CNN. Proses pelatihan berlangsung selama 40 *epoch.* CNN yang sudah dilatih dengan data*traning* menghasilkan model prediksi. Kode program *training* dan pembuatan arsitektur model dapat dilihat pada Gambar 14, 15, 16 dan 17.

```
_,acc=model.evaluate(X_test,Y_test)
print(acc*100)
```

Gambar 18 Kode Program Untuk Mengetahui Akurasi Model

```
model.save("model_klasifikasi_alat_transportasi.h5")
```

Gambar 19 Kode Program Menyimpan Model

Gambar 19 merupakan kode programsetelah model selesai ditraining, hasil model disimpan dengan nama file *"model_klasifikasi_jenis_kendaraan.h5 ",* Gambar 18 merupakan kode program untuk pengujian model dengan data uji yang sudah dibagi sebelumnya. Hasil pengujian ini berupa akurasi model tersebut.

```
#Evaluasi Model
plt.figure(figsize=[8,6])
plt.plot(x.history['loss'], 'r', linewidth = 3.0)
plt.legend(['Training Loss'], fontsize = 18)
plt.xlabel('Epochs', fontsize = 16)
plt.ylabel('Loss', fontsize = 16)
plt.title('Loss Curves', fontsize = 16)
plt.show()

plt.figure(figsize=[8,6])
plt.plot(x.history['accuracy'], 'r', linewidth = 3.0)
plt.legend(['Training Accuracy'], fontsize = 18)
plt.xlabel('Epochs', fontsize = 16)
plt.ylabel('Accuracy', fontsize = 16)
plt.title('Accuracy Curves', fontsize = 16)
plt.show()
```

Gambar 20 Kode Program Evaluasi Model

Pada Gambar 20 merupakan kode program untuk evaluasi model, di mana akan mengetahui visualisasi *loss* dan *accuracy* hasil pelatihan dalam bentuk grafik melalui library *Matplotlib.*

```
from PIL import Image
import numpy as np
im=Image.open("test5.jpg")
im=im.resize((32,32))
im=np.expand_dims(im,axis=0)
im=np.array(im)
pred=model.predict([im])[0]
if(pred[9]):
  print('Truck')
elif(pred[8]):
  print('Kapal Layar')
elif(pred[1]):
  print('Mobil')
else :
  print('Tidak Diketahui')
```

Gambar 21 Kode Program Pengujian Model

Kemudian model digunakan untukmemprediksi suatu citra digital yang dimasukkan yang hasilnya akan berupa cetakan prediksi alat transportasi yang tepat dengan menggunakan kode program dari Gambar 21.

Setelah itu di tahap pengujian akan diuji fungsi-fungsi yang terdapat dalam logika pemrogramman dengan metode *White-Box.*

E. Iterasi Dua

Pada tahap ini dilakukan pembuatan tampilan aplikasi GUI berbasis *dekstop* guna mempermudah pengguna dalammembuat *qr-code* yang akan otomatis tersimpan pada direktori aplikasi *user*.

Tampilan *user iterface* yang dibuat adalah tampilan untuk fungsi untuk memuat citra, fungsi otomatis *qr-code* berisi hasil prediksi; tanggal; jam, dan fungsi untuk memprediksi data citra tersebut

Tabel 2 Perangkat Pembuatan GUI

Perangkat Lunak	***Perangkat Keras***
Windows 10 64-bit	*8 GB RAM Memory*
Python 3.10.2	*AMD Radeon 7700 HD Seroes*
Tkinter 8.6	*i3-4130 ~3.4GHz*
PIL 9.0.1	*HDD 256 GB*
Pyqrcode 1.2.1	
NumPy 1.22.2	

Pada Tabel 2, pada perangkat lunak penelitian menggunakan *Operating System windows 10 64-bit*, bahasa pemrograman *Python* dan *library Tkinter*

untuk pembuatan *Graphical User Interface* (GUI).

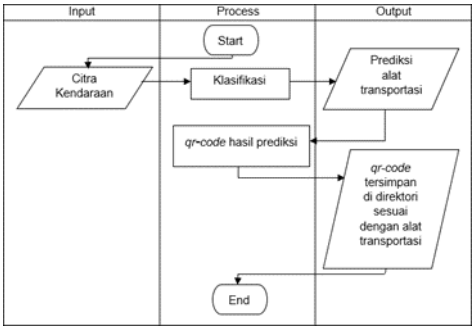
Selain perangkat pembuatan GUI yang digunakan penelitian yang disebutkan pada Tabel 2, penelitian telah menentukan persyaratan target yang diperlukan agar aplikasi dapat berjalan dengan baik pada Tabel 3. Di bawah ini adalah perangkat yang telah diputuskan oleh penelitian untuk memungkinkan aplikasi berjalan dengan benar.

Tabel 3 Perangkat Minimum UntukMenjalankan Aplikasi

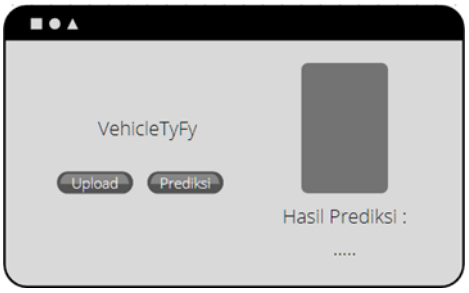
Perangkat Lunak	***Perangkat Keras***
Windows 10 atau lebih baru	*4 GB RAM Memory*
	Intel HD Graphics
	Processor i3
	HDD 120 GB

Tahap selanjutnya dilakukan perancangan *graphic user interface* untuk memberikan gambaran program menggunakan aplikasi ini. Pada iterasi kedua penelitian berfokus pada aplikasi untuk bagian tampilan. Tampilan *user interface* akan menyesuaikan dengan kebutuhan penelitian.

Tahap analisis ini dilakukan pada fitur yang akan digunakan pada GUI, di mana aplikasi berguna untuk melakukan klasifikasi, menampilkan citra yang dimasukkan, kemudian menampilkan hasil nama alat transportasi berdasarkan citra digital masukkan dan terakhirTahap selanjutnya dilakukan perancangan *graphic user interface* untuk yang akan dibuat menggunakan *tkinter*. Dari tahapan analisis, fungsi-fungsi yang akan dibuat ialah fungsi untuk memuat citra, fungsi otomatis *qr-code* berisi hasil prediksi; tanggal; jam, dan membuat fungsi untuk memprediksidata citra tersebut.



Gambar 22 Desain Pembuatan GUI



Gambar 23 Purwarupa Aplikasi

Pada tahap ini merealisasikan fungsi-fungsi yang sudah dianalisis dan di desain pada tahap sebelumnya ke dalambentuk kode program. Penelitian ini membuat 4 fungsi utama dalam GUI ini,yaitu fungsi load file citra digital, fungsimemprediksi citra digital dengan model yang telah dibuat, fungsi membuat *qr- code* pada hasil prediksi dan fungsi menyimpan *qr-code* kedalam folder direktori.

```
#Import Library Pembuatan Aplikasi GUI
import tkinter as tk
from tkinter import filedialog
from tkinter import *

#Library untuk citra
from PIL import ImageTk, Image

#Library untuk array
import numpy

#Untuk load model
from keras.models import load_model
model = load_model('model_klasifikasi_alat_transportasi.h5')

#Library untuk qrcode
from datetime import date,datetime
import pyqrcode
```

Gambar 24 Kode Program Inisialisasi Library

Pada Gambar 24 merupakan kode program untuk inisialisasi library yang akan digunakan untuk pengembangan GUI dan *load model* yang telah disimpan.

```
#Inisialisasi GUI
root=tk.Tk()
root.geometry('800x550') #Ukuran window
root.title('Klasifikasi Citra Alat Transportasi')
label=Label(root,font=('Arial',20))
sign_image = Label(root)
heading = Label(root, text="VehicleTyFy", font=('Arial',20,'bold'))
heading.place(x=100,y=110)
catatan = Label(root, text="QR code akan otomatis tersimpan di Folder Alat Transportasi!",
font=('Arial',10,'italic'))
catatan.place(x=200,y=530)
```

Gambar 25 Kode Program Inisialisasi Awal GUI

```
def klasifikasi(file_path):
    global label_packed
    image = Image.open(file_path)
    image = image.resize((32,32))
    image = numpy.expand_dims(image, axis=0)
    image = numpy.array(image)
    pred = model.predict([image])[0]
    today = date.today()
    now = datetime.now()
    current_time = now.strftime("%H.%M.%S")
    Jam = now.strftime("%H:%M:%S")
    d1 = today.strftime("%d-%m-%Y")
    if(pred[9]):
        label.configure(text='Truck',  font=('Arial',14,'bold'))
        teks_qr=("Aset Alat Transportasi adalah Truck"+'\nPada Tanggal: '+d1+'\nJam: '+Jam)
        #QR-Code
        im = pyqrcode.QRCode(teks_qr)
        im.png('test.png',scale=10)
        im = Image.open('test.png')
        im = im.convert("RGB")
        logo = Image.open(file_path)
        box = (220,220,340,340)
        im.crop(box)
        region = logo
        region = region.resize((box[2] - box[0], box[3] - box[1]))
        im.paste(region,box)
        im.save('Alat Transportasi/Truck/Hasil_QrCode_' + current_time + '.jpg')
    elif(pred[8]):
        label.configure(text='Kapal Layar',  font=('Arial',14,'bold'))
        teks_qr=("Aset Alat Transportasi adalah Kapal"+'\nPada Tanggal: '+d1+'\nJam: '+Jam)
        #QR-Code
        im = pyqrcode.QRCode(teks_qr)
        im.png('test.png',scale=10)
        im = Image.open('test.png')
        im = im.convert("RGB")
        logo = Image.open(file_path)
        box = (220,220,340,340)
        im.crop(box)
        region = logo
        region = region.resize((box[2] - box[0], box[3] - box[1]))
        im.paste(region,box)
        im.save('Alat Transportasi/Kapal/Hasil_QrCode_' + current_time + '.jpg')
        label.place(x=550,y=410)
```

Gambar 26 Kode Program Desain GUI

```
elif(pred[1]):
    label.configure(text='Mobil',  font=('Arial',14,'bold'))
    teks_qr=("Aset Alat Transportasi adalah Mobil"+'\nPada Tanggal: '+d1+'\nJam: '+Jam)
    #QR-Code
    im = pyqrcode.QRCode(teks_qr)
    im.png('test.png',scale=10)
    im = Image.open('test.png')
    im = im.convert("RGB")
    logo = Image.open(file_path)
    box = (220,220,340,340)
    im.crop(box)
    region = logo
    region = region.resize((box[2] - box[0], box[3] - box[1]))
    im.paste(region,box)
    im.save('Alat Transportasi/Mobil/Hasil_QrCode_' + current_time + '.jpg')
```

Gambar 27 Kode Program Desain GUI

```
else:
    label.configure(text='Tidak Diketahui',  font=('Arial',14,'bold'))
    label.place(x=540,y=410)
```

Gambar 28 Kode Program Desain GUI

Pada Gambar 25, 26, 27, 28 merupakan kode program desain GUI diawali dengan ukuran window, title aplikasi, lalu label nama aplikasi, label catatan, kemudian fungsi klasifikasi dan disertai fungsi *qr-code* yang akan otomatis tersimpan pada folder direktori masing-masing alat transportasi.

```
#Membuat button prediksi
def tampilkan_klasifikasi_button(file_path):
    klasifikasi_b=Button(root,text="Prediksi", font=('Arial',13),command=lambda: klasifikasi
    (file_path),padx=10,pady=10)
    klasifikasi_b.place(x=200,y=250)
```

Gambar 29 Kode Program Desain GUI

Gambar 30 Kode Program Desain GUI

Gambar 31 Kode Program Desain GUI

Gambar 29 merupakan kode program membuat tombol prediksi dan posisi tombol. Gambar 30 merupakan kode program membuat function upload citradigital dan label untuk menampung gambar citra digital yang dimasukkan. Gambar 31 merupakan kode program membuat tombol upload dan posisi tombol.

```
#posisi image yang diupload
sign_image.place(x=450,y=30,width=300,height=300)

#posisi label prediksi
label.place(x=590,y=410)
```

Gambar 32 Kode Program Desain GUI

```
#hasil prediksi txt
hasil_txt = tk.Label(root, text="Hasil Prediksi :" , font=('Arial',14))
hasil_txt.place(x=500,y=310,width=230,height=60)

root.mainloop()
```

Gambar 33 Kode Program Desain GUI

Pada Gambar 32 dan 33 merupakan kode program untuk mengatur posisi dari citra digital yang di input, posisi label prediksi dan membuat label hasil prediksi serta mengatur posisinya.

Setelah itu di tahap pengujian akan diuji fungsi-fungsi yang terdapat dalam logika pemrogramman dengan metode *Black-Box*.

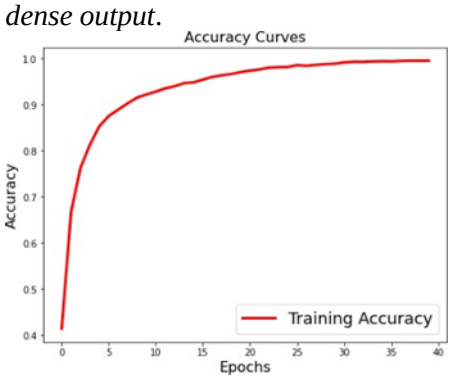
III. HASIL DAN PEMBAHASAN

A. Iterasi Satu

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        896

 conv2d_1 (Conv2D)           (None, 32, 32, 32)        9248

 max_pooling2d (MaxPooling2D  (None, 16, 16, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 16, 16, 64)        18496

 conv2d_3 (Conv2D)           (None, 16, 16, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 conv2d_4 (Conv2D)           (None, 8, 8, 128)         73856

 conv2d_5 (Conv2D)           (None, 8, 8, 128)         147584

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 128)        0
 2D)

 dropout (Dropout)           (None, 4, 4, 128)         0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dense_1 (Dense)             (None, 10)                5130

=================================================================
Total params: 1,341,226
Trainable params: 1,341,226
Non-trainable params: 0
_________________________________________________________________
```

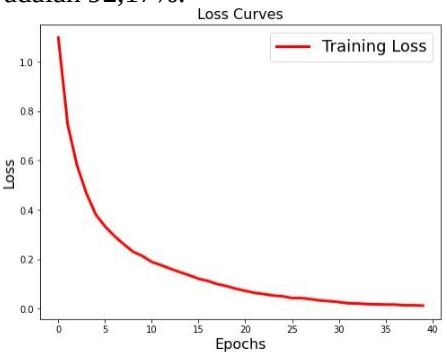
Gambar 34 Desain Model

Pada Gambar 34 merupakan deskripsi arsitektur *model* untuk menguji tahapan ini berjalan dengan baik atau tidak, di mana dimulai denganmelakukan konvolusi citra digital, citra digital ini akan kita dapatkan nilai nilai hasil dari konvolusi, kemudiandilakukan konvolusi kembali untuk mendapatkan jumlah parameter yang akan dilatih, setiap konvolusi memiliki nilai fungsi aktivasi, pada penelitian ini menggunakan fungsi aktivasi *ReLu* di mana jika ada nilai negatif nilai itu akan dibuat menjadi 0, selanjutnya dilakukan *pooling* menggunakan *max pooling*. Proses tersebut diulangi sebanyak 2x di mana tahap ini biasa disebut *feature learning* atau *feature extraction*. Kemudian memasuki tahap f*latten* berguna untuk mengubah *pixel* menjadi 1 dimensi yang sebelumnya masih terbentuk matriks. Lalu dilakukan *dropout* untuk mengurangi *overvitting.* Kemudian memasuki tahap fully connected menggunakan *dense hidden layer* dan *dense output*.



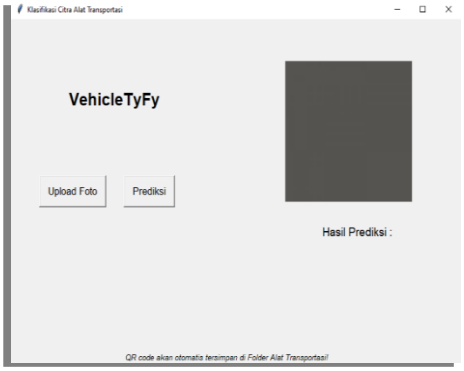
Gambar 35 Grafik Akurasi Proses Pelatihan

Pada Gambar 35, grafik tersebut memiliki sumbu x yang mewakili *epoch* dan sumbu y yang mewakili nilai akurasi. Semakin tinggi nilai akurasi pelatihan, semakin baik model untuk memprediksi. Pada grafik *loss* proses pelatihan dapat dilihat pada Gambar 36,di mana sumbu x adalah yang mewakili*epoch* dan sumbu y mewakili nilai *loss*. Semakin rendah nilai *loss* yang dimilikioleh sebuah model akan akan semakin baik model tersebut. Jadi nilai akurasi data latih pada 40 *epoch* adalah 92,17%.



Gambar 36 Grafik Loss Proses Pelatihan

B. Iterasi Dua

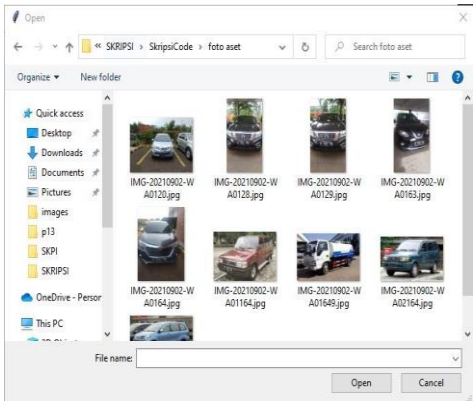


Gambar 37 Tampilan Awal GUI

Hasil dari iterasi dua adalah hasil tampilan desain aplikasi yang menggunakan *library Tkinter* untuk membuat apa yang disebut dengan antarmuka pengguna atau GUI *(graphical user interface*). Gambar 37 menunjukkan tampilan awal aplikasi.

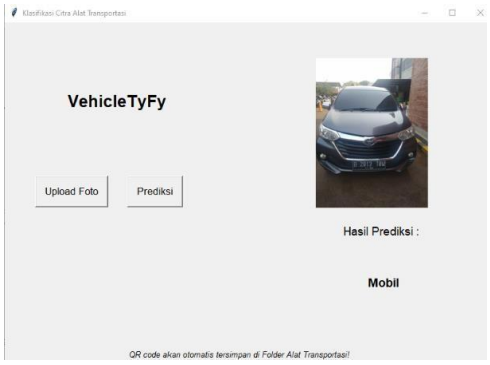
Selanjutnya tombol "Upload

Foto" digunakan untuk menjalankan fungsi pemilihan file data gambar. Tampilan pemilihan file data gambar ditunjukkan pada Gambar 38.



Gambar 38 Pemilihan Berkas Data Gambar

Setelah memilih data gambar, gambar akan ditampilkan pada bingkai di GUI dan tombol "Prediksi" akan otomatis muncul disamping tombol "Upload Foto". Selanjutnya tekan tombol "Prediksi" untuk melakukan prediksi data gambar yang telah dipilih sebelumnya. Hasil akhir pada tampilan GUI atau aplikasi dapat terlihat pada Gambar 39. Setelah hasil prediksi dihasilkan, fungsi membuat *qr-code*akan dijalankan dan hasil pembuatan *qr-code* akan tersimpan di folder sesuai dengan alat transportasi yang diprediksi. Hasil tampilan *qr-code* dapat dilihat pada Gambar 40.



Gambar 39 Tampilan Hasil Akhir Aplikasi

Gambar 40 Tampilan Hasil qr-code

IV. SIMPULAN

Setelah melakukan berbagai upaya untuk memecahkan masalah, yaitu mengidentifikasi masalah, membangun kerangka berpikir dan mengembangkan aplikasi. Berikut adalah beberapa poin penting yang dapat disimpulkan dari proses penelitian sebagai berikut:

1. Berdasarkan tujuan dari penelitian ini yaitu, pengembangan aplikasi klasifikasi alat transportasi berdasarkan citra digital untuk pencatatan aset menggunakan algoritma CNN dengan pustaka perangkat lunak TensorFlow.
2. Model pembelajaran mesin sebagaiinti dari aplikasi VehicleTyFy dikembangkan menggunakan pustaka perangkat lunak tensorflow yang menerapkan konsep pembelajaran yang diawasi.
3. Data terbagi menjadi data latih dandata uji. Hasil akurasi dari algoritma Jaringan Saraf Konvolusional yang telah dibuat pada 40 epoch untuk data latih bernilai 92,17% dan 92,16% pada data uji.
4. Hasil pengujian model pada iterasi satu menggunakan white box sudahberjalan sesuai harapan dimana fungsi-fungsi sudahterimplementasi dengan baik dan mendapat model terbaik.
5. Hasil pengujian GUI menggunakan black-box sudah berjalan sesuai

harapan dimana fitur-fitur yang
6. sebelumnya dirancang sudah terimplementasi dengan baik dan berfungsi dengan semestinya.

Berdasarkan hasil penelitian yang telah dilakukan, peneliti memiliki saran sebagai berikut:

1. Menambah kategori yang lebihbanyak lagi seperti merek dan harga alat transportasi.
2. Membuat pengembangan aplikasiyang serupa berbasis *web* dan *realtime object detection.*
3. Memperbaiki tampilan-tampilan yang ada pada aplikasi, tampilan tersebut harus dibuat secaramenarik.

DAFTAR RUJUKAN

[1] G. Marcus, "Deep Learning: A Critical Appraisal," *arXiv preprint arXiv:1801.00631,* pp. 1-27, 2018.

[2] G. W. Intyanto, "Klasifikasi Citra Bunga dengan Menggunakan Deep Learning: CNN (Convolution Neural Network)," *Jurnal Arus Elektro Indonesia,* vol. 7(3), pp. 80-83, 2021.

[3] I. W. S. E. Putra, "Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101," *J.Tek. ITS,* vol. 5, no. 1, pp. 65-69, 2016.

[4] M. Sholihin, M. R. Zamroni dan B. , "Identifikasi Kesegaran Ikan Berdasarkan Citra Insang Dengan Metode Convolution Neural Network," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi),* vol. 8.3, pp. 1352-1360, 2021.

[5] A. Parlys, A. A. Zahra dan A. Hidayanto, "Penggolongan Lagu Berdasarkan Spektogram dengan Convolution Neural Network," *Transient: Jurnal Ilmiah Teknik Elektro* , vol. 7.1, pp. 28-33, 2018.

[6] M. Xin dan Y. Wang, "Research on image classification model based on deep convolution neural network," *EURASIP Journal on Image and Video Processing,* pp. 1-11, 2019.

[7] I. C. Education, "Convolutional Neural Networks," IBM, 20 October 2020. [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks. [Diakses 16 August 2022].

[8] P. Winardi dan E. Setyati, "Identifikasi Jenis Daging Menggunakan Algoritma Convolution Neural Network," *Journal of Information System, Graphics, Hospitality and Technology,* vol. 3.02, pp. 82-88, 2021.

[9] T. Augusta, M. A. Pangestu, D. A. Mara dan T. Indriyani, "Implementasi ModelIncrementalPada Sistem Informasi Klinik Nurani JayaBerbasis Desktop," *Prosiding Seminar Nasional TeknikElektro, Sistem Informasi, dan Teknik Informatika (SNESTIK),* vol. 1.1, pp. 275- 282, 2021.

[10] M. Martin, "Incremental Model in SDLC: Use, Advantage & Disadvantage," Guru99, 25 June 2022. [Online]. Available: https://www.guru99.com/what-is-incremental-model-in-sdlc-advantages-disadvantages.html. [Diakses August 18 2022].