

## Sistem Pendeteksi Nama dan Harga Barang Pada Toko Sudiman Menggunakan Teachable Machine

Muhammad Rizky Thio<sup>1</sup>, Fuad Nur Hasan<sup>2</sup>

<sup>1,2</sup>Informatika/Illmu Komputer, Universitas Bina Sarana Informatika, Jl. Margonda No.8, Pondok Cina, Kecamatan Beji, Kota Depok, Jawa Barat, 16424

[rizkythio60@gmail.com](mailto:rizkythio60@gmail.com)<sup>1\*</sup>, [fuad.fnu@bsi.ac.id](mailto:fuad.fnu@bsi.ac.id)<sup>2</sup>

### Abstrak

*Toko Sudiman menghadapi masalah dalam identifikasi dan penentuan harga barang yang lambat dan kurang efisien, menyebabkan kesalahan harga dan menurunkan kepuasan pelanggan. Penelitian ini bertujuan mengembangkan sistem otomatis berbasis Pembelajaran Mesin untuk mendeteksi nama dan harga barang secara akurat. Menggunakan model dari Teachable Machine Google, sistem ini dibangun dengan arsitektur Convolutional Neural Network (CNN) untuk klasifikasi gambar, yang terbukti efektif. Penelitian ini menargetkan akurasi lebih dari 80% dan diujikan dalam kondisi ideal untuk memastikan performa optimal. Diharapkan sistem ini mampu meningkatkan efisiensi operasional, mengurangi kesalahan harga, dan meningkatkan kepuasan pelanggan di Toko Sudiman.*

**Kata kunci:** Teachable Machine, Pembelajaran Mesin, Pemindai

### A. Pendahuluan

Toko Sudiman menghadapi kendala dalam proses identifikasi harga barang yang lambat dan kurang efisien, yang berdampak pada kesalahan harga dan ketidakpuasan pelanggan. Teknologi menawarkan solusi inovatif, terutama melalui kecerdasan buatan (AI). Salah satu terobosan dalam AI adalah Machine Learning, di mana "teknologi Machine Learning memungkinkan komputer untuk belajar tanpa harus diprogram secara tradisional" (B. M. Wantania et al., 2020). Selain itu, "model Machine Learning berfungsi mendeteksi dan mengklasifikasi objek, yang masih menjadi tantangan utama dalam computer vision" (Ningsih et al., 2022). Penggunaan Teachable Machine dari Google juga memungkinkan penerapan Machine Learning dalam deteksi otomatis, seperti "mendeteksi tanaman Aglonema berbasis android" (Baihaqi et al., 2022), serta membantu "menerjemahkan bahasa isyarat ke huruf abjad" (Harditya, 2020).

Untuk mengatasi masalah kesalahan harga, penelitian ini berfokus pada pengembangan aplikasi dengan teknologi Machine Learning dan Convolutional Neural Network (CNN) untuk mengidentifikasi harga barang secara otomatis, yang diharapkan "meningkatkan efisiensi dan akurasi dalam menentukan harga" (Dacipta & Putra, 2022).

## B. Metode

### Teachable Machine

Dengan Teachable Machine, Anda dapat melatih komputer untuk mengenali gambar, suara, dan posisi tubuh. “Teachable Machine merupakan alat yang dapat digunakan untuk membuat sebuah model klasifikasi yang mudah digunakan untuk mengembangkan aplikasi machine learning” (Chazar & Rafsanjani, 2022).

### Machine Learning

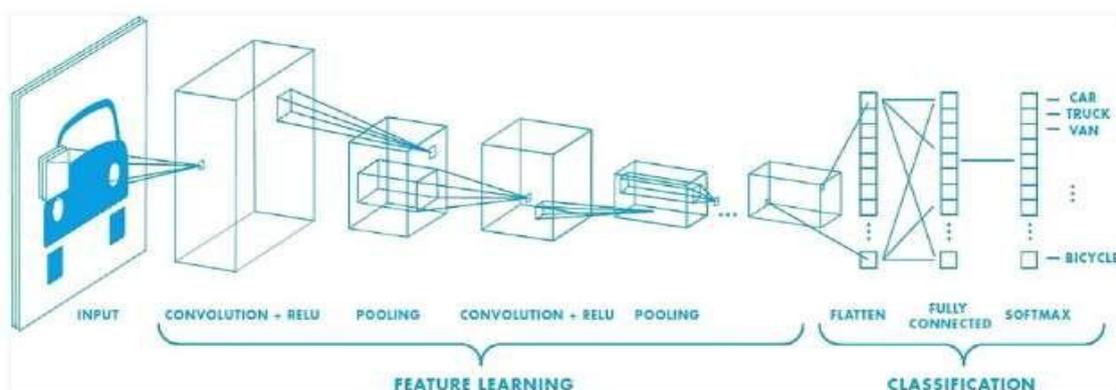
“Machine Learning is involved not only in robotics, but also in many other applications”(Géron & Russell, 2019). “Machine learning adalah bidang ilmu yang mengembangkan algoritma atau model yang dapat belajar dari data untuk menggali pengetahuan yang ada pada data tersebut seperti proses belajar pada manusia (Kharis & Zili, 2022).

### Tensorflow

“Tensorflow adalah sebuah framework komputasional untuk membuat model machine learning. TensorFlow menyediakan berbagai toolkit yang memungkinkan untuk membuat model pada tingkat abstraksi yang disukai dan dapat menjalankan grafik pada beberapa platform hardware, termasuk CPU, GPU, dan TPU” (Nasha Hikmatia A.E. & Zul, 2021).

### Convolutional Neural Networks (CNN)

“Convolutional neural networks or also known as CNN and are one of the most commonly used types of feed-forward neural networks in that the connectivity pattern between neurons is based on the organization of the common visual cortex system” (Millstein, 2018). CNN salah satu jenis algoritma deep learning yang dapat menentukan objek apa saja dalam sebuah gambar, mengenali dan membedakan antara satu gambar dengan yang lainnya. Bisa di lihat pada gambar di bawah ini, ada 3 Layer CNN yaitu: Convolution Layer, Polling Layer, Fully Connected Layer. Lalu ada tambahan sebuah node di akhir keluaran di setiap jaringan syarafnya yaitu Activation Function.



Gambar 1. The architecture of convolutional neural network.  
Sumber: (Penelitian, 2024)

## Dropout Regularization

Regularization adalah teknik yang digunakan untuk mengurangi overfitting atau noise. Adalah kondisi di mana sistem mampu belajar dengan baik dengan data pelatihan, namun tidak dapat menggeneralisasi dengan data uji.

“Dropout adalah teknik yang dapat digunakan untuk mencegah overfitting serta mempercepat proses pelatihan. Dropout menghilangkan neuron yang berupa hidden maupun layer yang terlihat pada jaringan. Neuron yang hilang akan dipilih secara acak oleh sistem dengan probabilitas dari nol hingga satu” (Eduardo Christianto, 2021).

## C. Hasil dan Pembahasan

### Proses dan Langkah Penelitian

Penelitian ini melibatkan beberapa komponen penting dalam pengembangan Sistem Deteksi Nama dan Harga Barang di Toko Sudiman. Berikut adalah rincian kebutuhan sistem dan langkah-langkah penelitian yang dilakukan:

#### Kebutuhan Sistem

##### Software:

- Android Studio
- Visual Studio Code
- Teachable Machine
- Web Browser
- Figma

##### Hardware

- Komputer

**Tabel 1. Spesifikasi Komputer.**

Processor	Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz 3.40 GHz
Memory	16000MB RAM
Storage	118GB SSD, 500GB HDD
OS	Windows 10 Pro 64-Bit
Network	Ethernet

Sumber: (Penelitian, 2024)

- Smartphone

**Tabel 2. Spesifikasi Smartphone.**

Processor	2.0ghZ Snapdragon 665 Octa-core
Memory	8 GB
Storage	128 GB
OS	Android versi 12

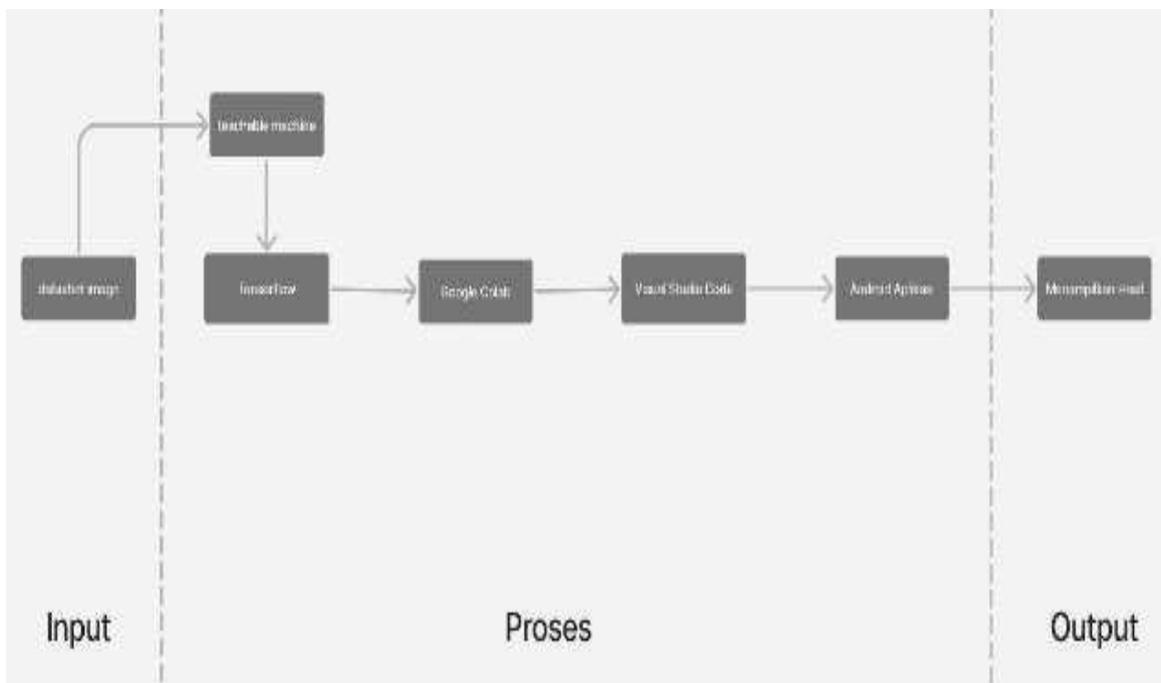
Sumber: (Penelitian, 2024)

### Rancangan Sistem

Di dalam perancangan sistem ini berguna untuk memberikan gambaran tentang fungsional sistem. Perancangan sistem ini meliputi blok diagram, *flowchart*, desain aplikasi.

### Blok Diagram

Sistem ini terdiri dari beberapa subsistem yang saling berinteraksi. Blok *input* berupa dataset gambar, proses dilakukan menggunakan Teachable Machine dan Google Colab, dan output berupa aplikasi Android untuk mendeteksi harga barang.

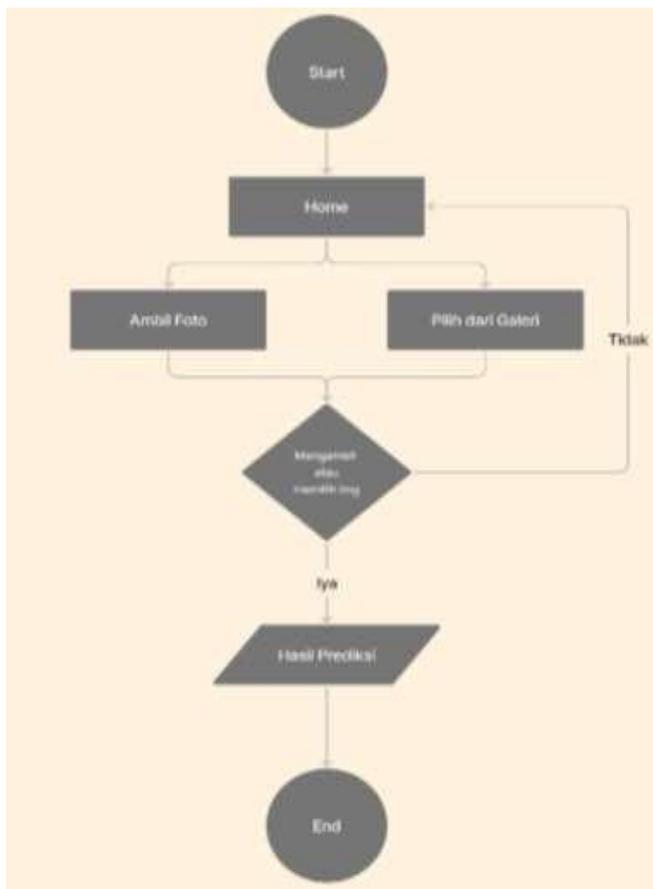


Gambar 2. Blok Diagram.

Sumber: (Penelitian, 2024)

### Flowchart

Flowchart sistem memandu alur kerja keseluruhan sistem, memudahkan perancangan dan pembangunan aplikasi. Langkah awal adalah melatih dataset menggunakan Teachable Machine, yang hasilnya diintegrasikan ke dalam aplikasi mobile melalui Android Studio atau Visual Studio Code, lalu diinstal pada *smartphone*. Flowchart menjelaskan proses aplikasi, mulai dari pembukaan, pengambilan gambar oleh kamera, perbandingan dengan dataset, hingga menampilkan prediksi nama dan harga barang di layar *smartphone*.

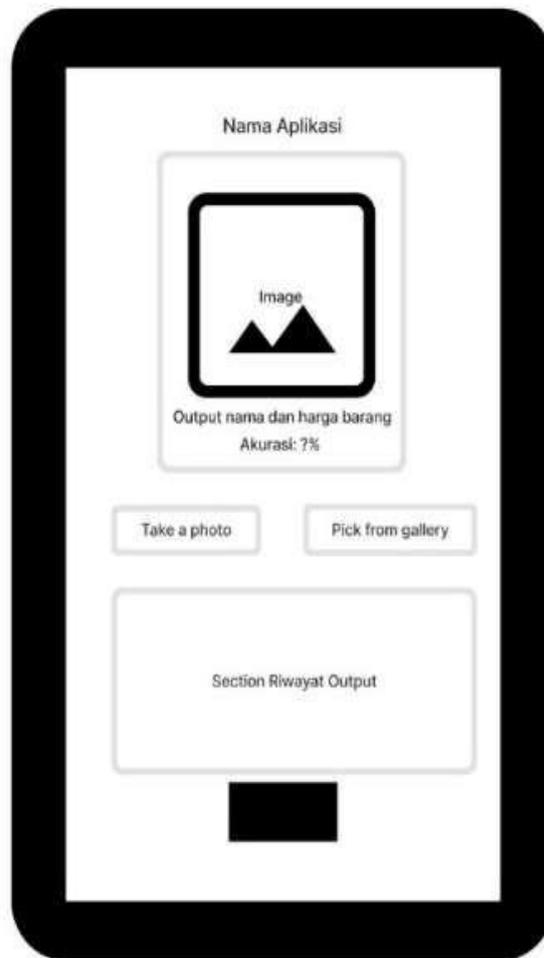


Gambar 3. Flowchart Sudiman Scanner.

Sumber: (Penelitian, 2024)

### Desain Aplikasi

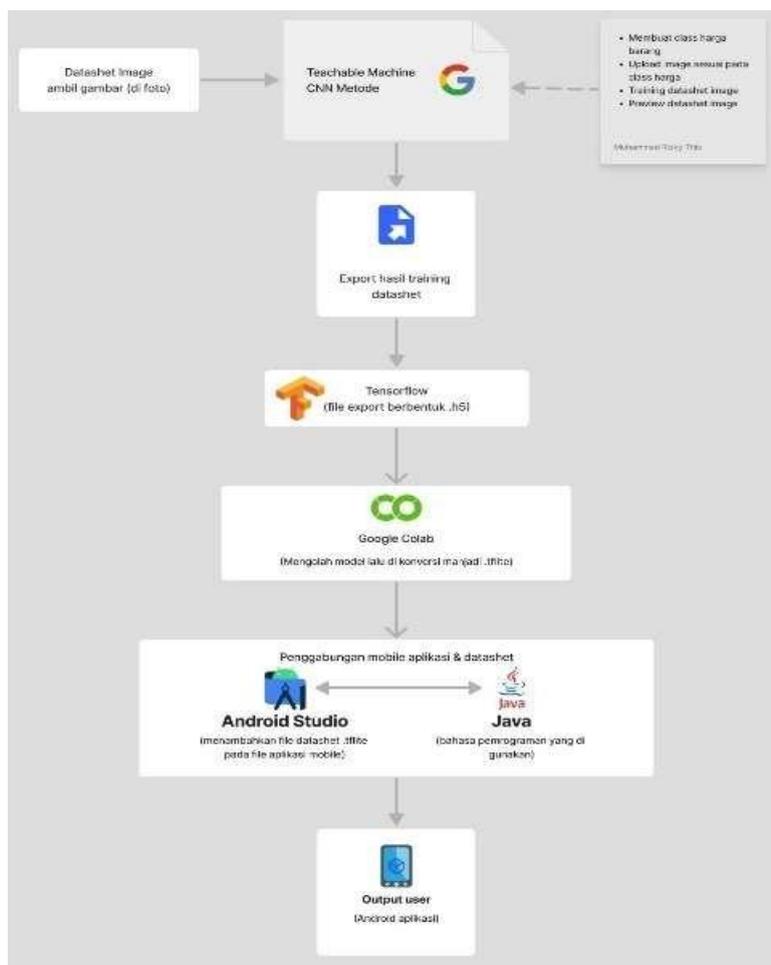
Untuk desain aplikasi nya sangat sederhana hanya berisi nama aplikasi headernya, lalu ada output gambar yang kita input bisa menggunakan langsung dari kamera atau bisa di input langsung di penyimpanan *smartphone*, setelah di *input* akan muncul nama dan harga barang beserta berapa akurasi nya. Untuk desain nya bisa di lihat di gambar di bawah ini.



Gambar 4. Desain Aplikasi.  
Sumber: (Penelitian, 2024)

### Rancangan Sistem Aplikasi

Langkah awal pengembangan aplikasi ini adalah membuat dataset atau model untuk klasifikasi harga barang. Proses ini menggunakan Teachable Machine, platform dari Google yang memudahkan pembuatan model pembelajaran mesin tanpa perlu keahlian pemrograman.



Gambar 5. Rancangan sistem aplikasi.

Sumber: (Penelitian, 2024)

## Metode Pengolahan dan Analisis Data

### 1. Pengumpulan Data

Pengumpulan data dilakukan setelah perancangan sistem, desain *flowchart*, dan desain aplikasi selesai. Data dikumpulkan melalui observasi lapangan di Toko Sudiman dengan mengambil foto barang dalam format JPG untuk digunakan sebagai dataset. Observasi ini bertujuan untuk memahami proses identifikasi dan penetapan harga barang, serta mengidentifikasi kendala yang muncul. Data yang terkumpul dianalisis untuk mendukung pengembangan sistem identifikasi harga berbasis *Machine Learning*.



Gambar 6. Foto barang-barang yang di gunakan untuk sampel.

Sumber: (Penelitian, 2024)

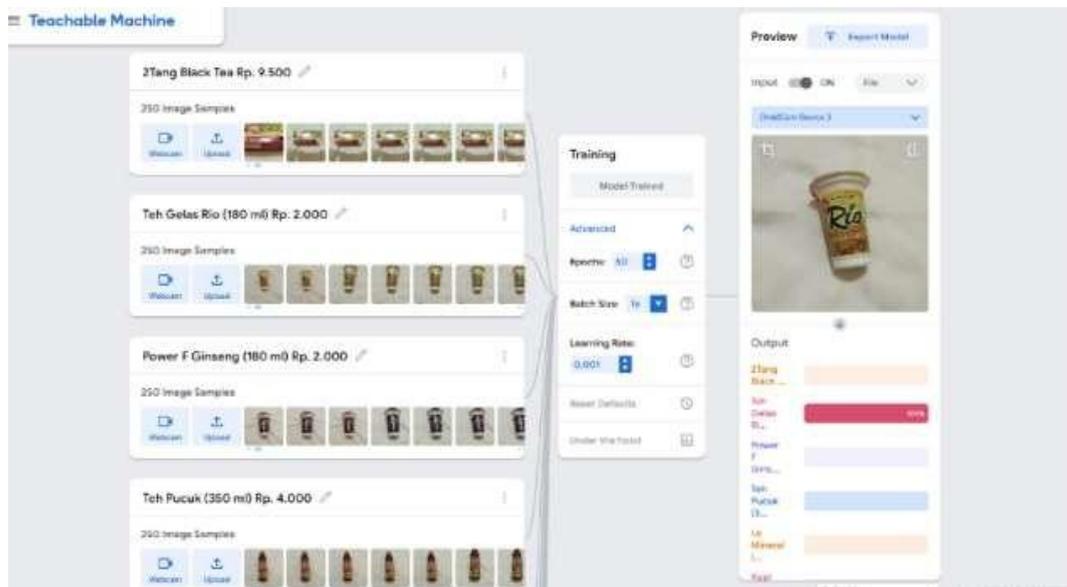
Dalam proses pengumpulan data ini, setiap barang yang diperlukan difoto satu per satu dari berbagai sudut yang berbeda agar dataset yang dihasilkan mencakup variasi yang maksimal. Setelah itu, foto-foto barang yang serupa disatukan dalam folder yang sama untuk memudahkan proses pengolahan dataset.

Name	Date modified	Type
2Tang Black Tea Rp. 9.500	11/06/2024 10:32	File folder
Aqua (600 ml) Rp. 4.000	11/06/2024 10:37	File folder
Cleo (220 ml) Rp. 1.500	11/06/2024 10:37	File folder
Geikikara Ramen Hot Spicy (120g) Rp. 7.000	11/06/2024 10:35	File folder
Gekikara Ramen Hot Carbo (120g) Rp. 7.000	11/06/2024 10:35	File folder
Indomie Goreng (85g) Rp. 3.500	11/06/2024 10:36	File folder
Indomie Rasa Soto (70g) Rp. 3.000	11/06/2024 10:36	File folder
Kopi Kapal Api Special Mix 23g Rp. 1.000	11/06/2024 10:34	File folder
Le Mineral (500 ml) Rp. 5.000	11/06/2024 10:38	File folder
Power F Ginseng (180 ml) Rp. 2.000	11/06/2024 10:39	File folder
Teh Gelas Rio (180 ml) Rp. 2.000	11/06/2024 10:39	File folder
Teh Pucuk (350 ml) Rp. 4.000	11/06/2024 10:39	File folder

Gambar 7. Pengelompokan gambar.  
Sumber: (Penelitian, 2024)

## 2. Pembuatan Dataset

Di dalam penelitian ini pembuatan datasetnya di dalam Teachable Machine untuk membuat dataset sekaligus training model nya di Teachable Machine. Di dalam Teachable Machine nya bisa langsung masukan gambar-gambar yang sudah di kelompokkan susai *classnya* dan training modelnya.



Gambar 8. Pengumpulan dataset.

Sumber: (Penelitian, 2024)

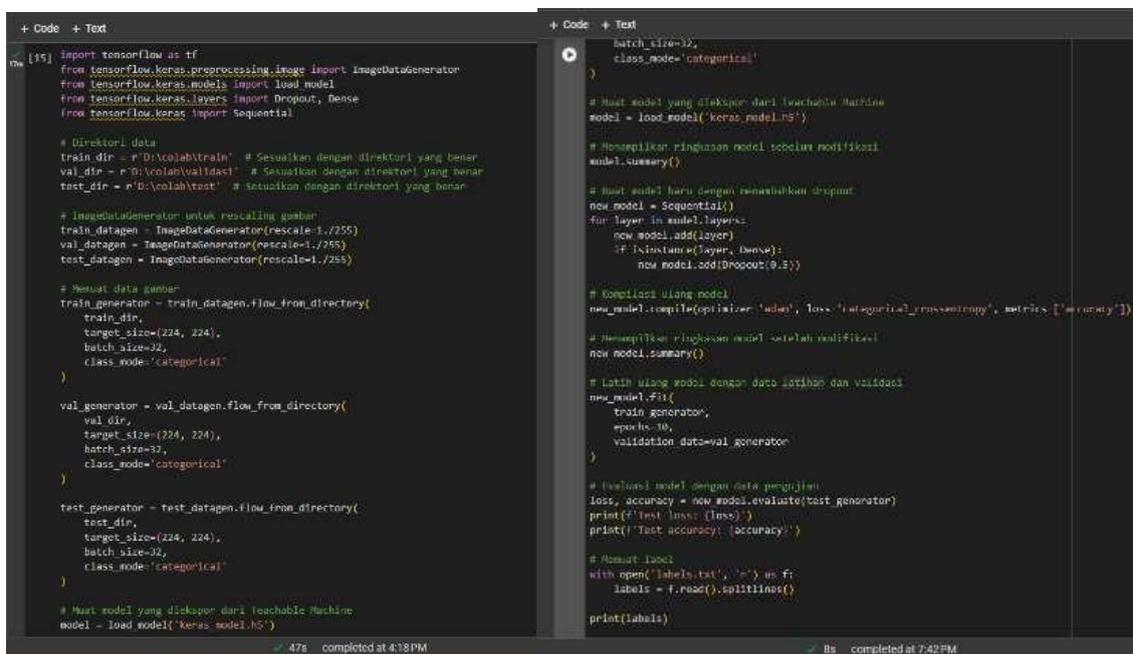
Tabel 3. Class dan jumlah gambar.

Class	Jumlah Image
2Tang Black Tea Rp. 9.500	150
Teh Gelas Rio (180 ml) Rp. 2.000	150
Power F Ginseng (180 ml) Rp. 2.000	150
Teh Pucuk (350 ml) Rp. 4.000	150
Le Mineral (500 ml) Rp. 5.000	150
Kopi Kapal Api Special Mix 23g Rp. 1.000	150
Gekikara Ramen Hot Carbo (120g) Rp. 7.000	150
Geikikara Ramen Hot Spicy (120g) Rp. 7.000	150
Indomie Goreng (85g) Rp. 3.500	150
Indomie Rasa Soto (70g) Rp. 3.000	150
Aqua (600 ml) Rp. 4.000	150
Cleo (220 ml) Rp. 1.500	150

Sumber: (Penelitian, 2024)

### 3. Penambahan Metode Pengolahan

Di dalam penelitian ini ada penambahan metode pengolahan dengan menggunakan metode *Dropout regularization*, yang dimana metode ini berguna jika model yang di gunakan mengalami *overfitting* dan memperbagus Akurasi.



```
+ Code + Text
[15]
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import Dropout, Dense
from tensorflow.keras import Sequential

# Direktori data
train_dir = 'D:\colab\train' # Sesuaikan dengan direktori yang benar
val_dir = 'D:\colab\validasi' # Sesuaikan dengan direktori yang benar
test_dir = 'D:\colab\test' # Sesuaikan dengan direktori yang benar

# ImageDataGenerator untuk rescaling gambar
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

# Memuat data gambar
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)

# Muat model yang di ekspor dari Teachable Machine
model = load_model('keras_model.h5')

batch_size=32,
class_mode='categorical'
)

# Muat model yang di ekspor dari Teachable Machine
model = load_model('keras_model.h5')

# Menampilkan ringkasan model sebelum modifikasi
model.summary()

# Muat model baru dengan menambahkan dropout
new_model = Sequential()
for layer in model.layers:
    new_model.add(layer)
    if isinstance(layer, Dense):
        new_model.add(Dropout(0.5))

# Kompilasi ulang model
new_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Menampilkan ringkasan model setelah modifikasi
new_model.summary()

# Latih ulang model dengan data latihan dan validasi
new_model.fit(
    train_generator,
    epochs=10,
    validation_data=val_generator
)

# Evaluasi model dengan data pengujian
loss, accuracy = new_model.evaluate(test_generator)
print('Test loss: {loss}')
print('Test accuracy: {accuracy}')

# Mount label
with open('labels.txt', 'w') as f:
    labels = f.read().splitlines()
print(labels)

478 completed at 4:38 PM
Bs completed at 7:42 PM
```

Gambar 9. Tampilan Google Colab metode Dropout Regularization.

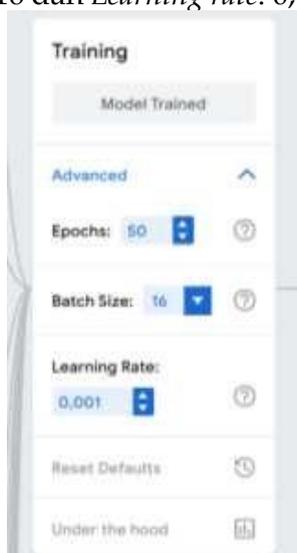
Sumber: (Penelitian, 2024)

## Hasil Penelitian

Pada penelitian ini akan dibahas proses, hasil, dan pengujian yang mendukung hipotesis dan rumusan masalah. Sistem pendeteksi nama dan harga barang di Toko Sudiman menggunakan Teachable Machine. Pengembangan proyek ini melalui beberapa tahapan pengolahan data, yang akan dijelaskan mulai dari pengolahan hingga menjadi aplikasi mobile.

### 1. Output dari Teachable Machine

Pengumpulan datanya lalu di dalam pembuatan dataset ini perlu *training* model nya menggunakan *epoch: 50, Batch Size:16 dan Learning rate: 0,001*. Bisa di lihat di bawah ini.



Gambar 10. Pengaturan Teachable Machine.

Sumber: (Penelitian, 2024)

### Accuracy Per Class

Adapun hasil akurasi per *class* yang di tunjukan oleh Teachable Machine nya yang meliputi jenis barang nya yang di antaranya meliputi: *Class*, *Accuracy* nya per *class* sekiat 1.00 atau 100% dan juga *Samples*.

Accuracy per class

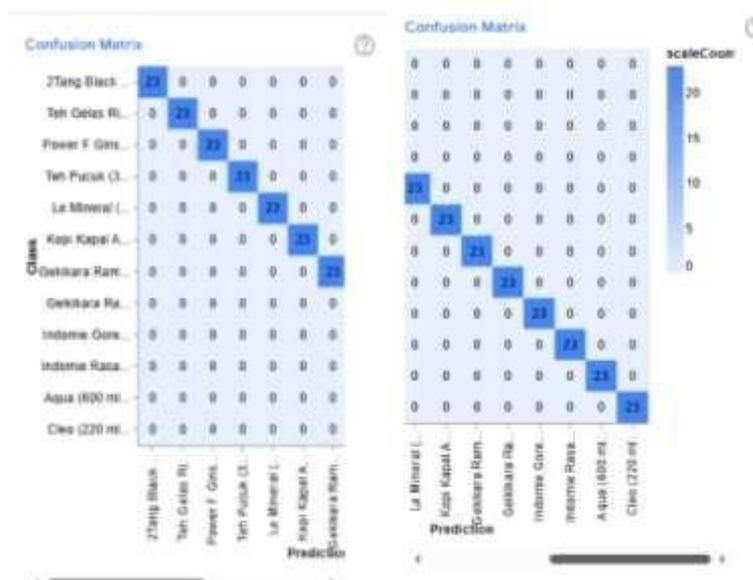
CLASS	ACCURACY	# SAMPLES
2Tang Black ...	1.00	23
Teh Gelas Ri...	1.00	23
Power F Gins...	1.00	23
Teh Pucuk (3...	1.00	23
Le Mineral (...)	1.00	23
Kopi Kapal A...	1.00	23
Gekikara Ram...	1.00	23
Geikikara Ra...	1.00	23
Indomie Gore...	1.00	23
Indomie Rasa...	1.00	23
Aqua (600 ml...	1.00	23
Cleo (220 ml...	1.00	23

Gambar 11. Accuracy per class.

Sumber: (Penelitian, 2024)

### Confusion Matrix

*Confusion Matrix* di bawah ini menunjukkan bahwa model mengklasifikasikan setiap gambar dari semua produk dengan benar sebanyak 23 kali tanpa kesalahan, menunjukkan akurasi 100% untuk semua kelas yang diuji.

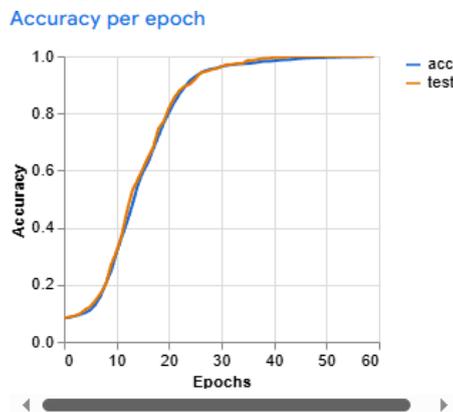


Gambar 12. Confusion Matrix.

Sumber: (Penelitian, 2024)

### Accuracy Per Epoch

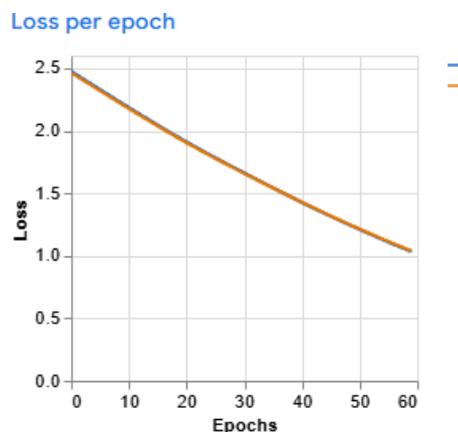
Adapun *Accuracy per epoch* di bawah ini grafik ini menunjukkan peningkatan akurasi model seiring dengan bertambahnya jumlah *epoch* selama proses pelatihan, baik untuk data pelatihan maupun data pengujian.



Gambar 13. *Accuracy per epoch*.  
Sumber: (Penelitian, 2024)

### Accuracy Per Loss

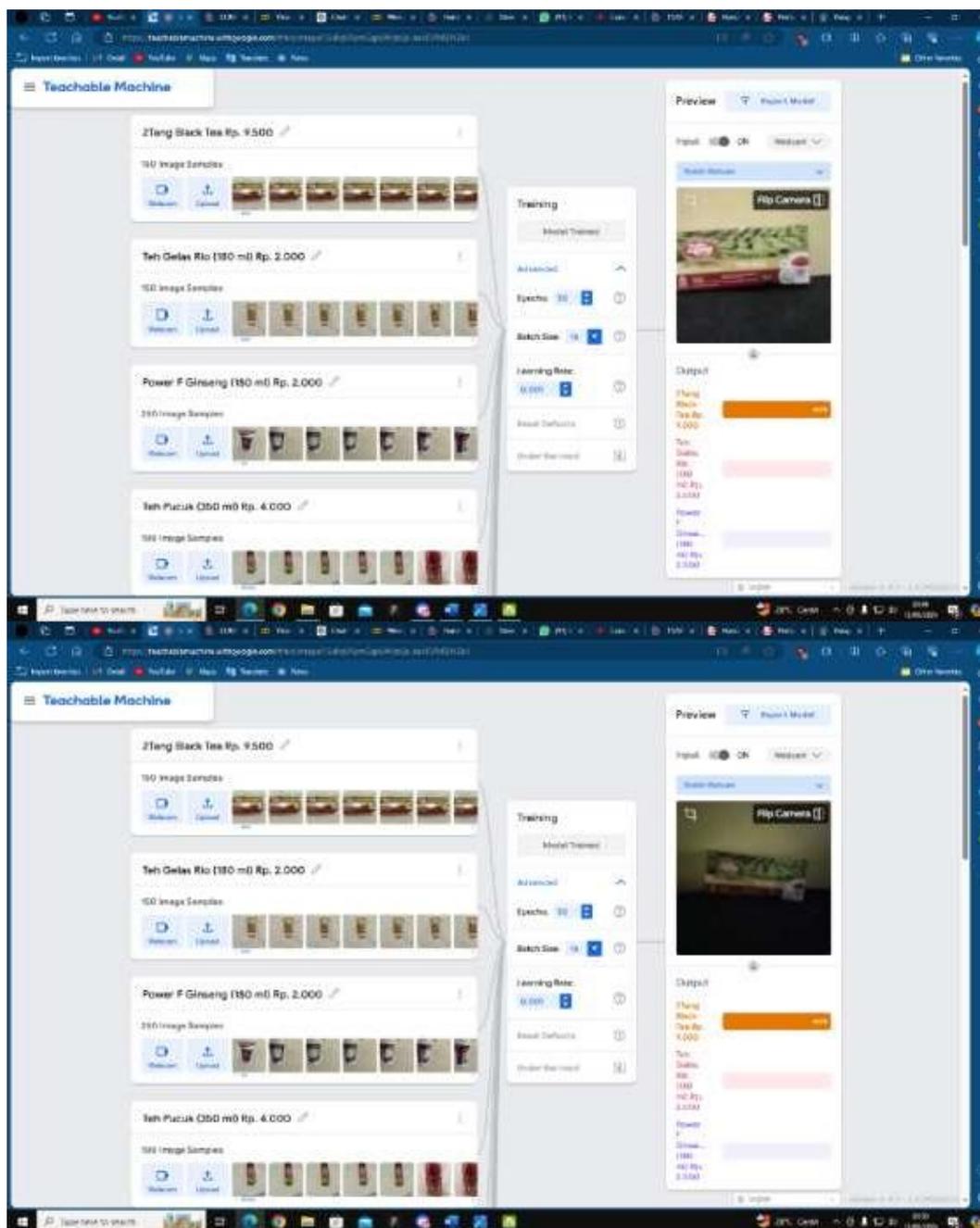
Dan ada juga *loss per epoch* yang menunjukkan kerugian model seiring bertambah jumlah *epoch* nya selama proses *taining* model.



Gambar 14. *Accuracy per loss*.  
Sumber: (Penelitian, 2024)

- **Hasil Prediksi Teachable Machine**

Bawah ini adalah data *testing* dari hasil *training* model di Teachable Machine, dengan mengarahkan kamera ke barangnya atau meng *import file* ke Teachable Machine.



Gambar 15. Tampilan Teachable Machine uji data *training*.  
Sumber: (Penelitian, 2024)

## 2. Pengujian Hipotesis

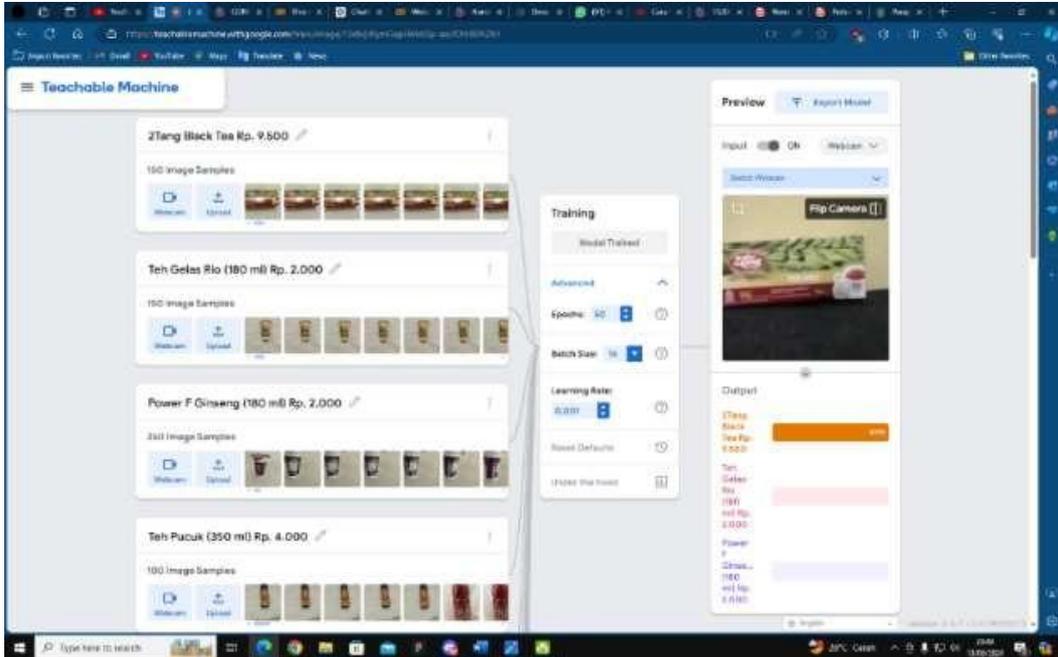
Di dalam penelitian ini menguji hipotesis-hipotesis apakah hipotesisnya di terima atau tidak.

$H_0$ : Sistem pendeteksi nama dan harga barang yang dikembangkan menggunakan Teachable Machine akan mampu mendeteksi nama dan harga barang dengan akurasi lebih dari 80%.

Jawaban:

Diterima: Pada gambar di bawah, hasil pendeteksian menunjukkan bahwasistem berhasil mengidentifikasi "2Tang Black Tea" dengan akurasi 100%.

Ini menunjukkan bahwa sistem pendeteksi mampu mencapai akurasi lebih dari 80%, sehingga hipotesis  $H_0$  diterima.



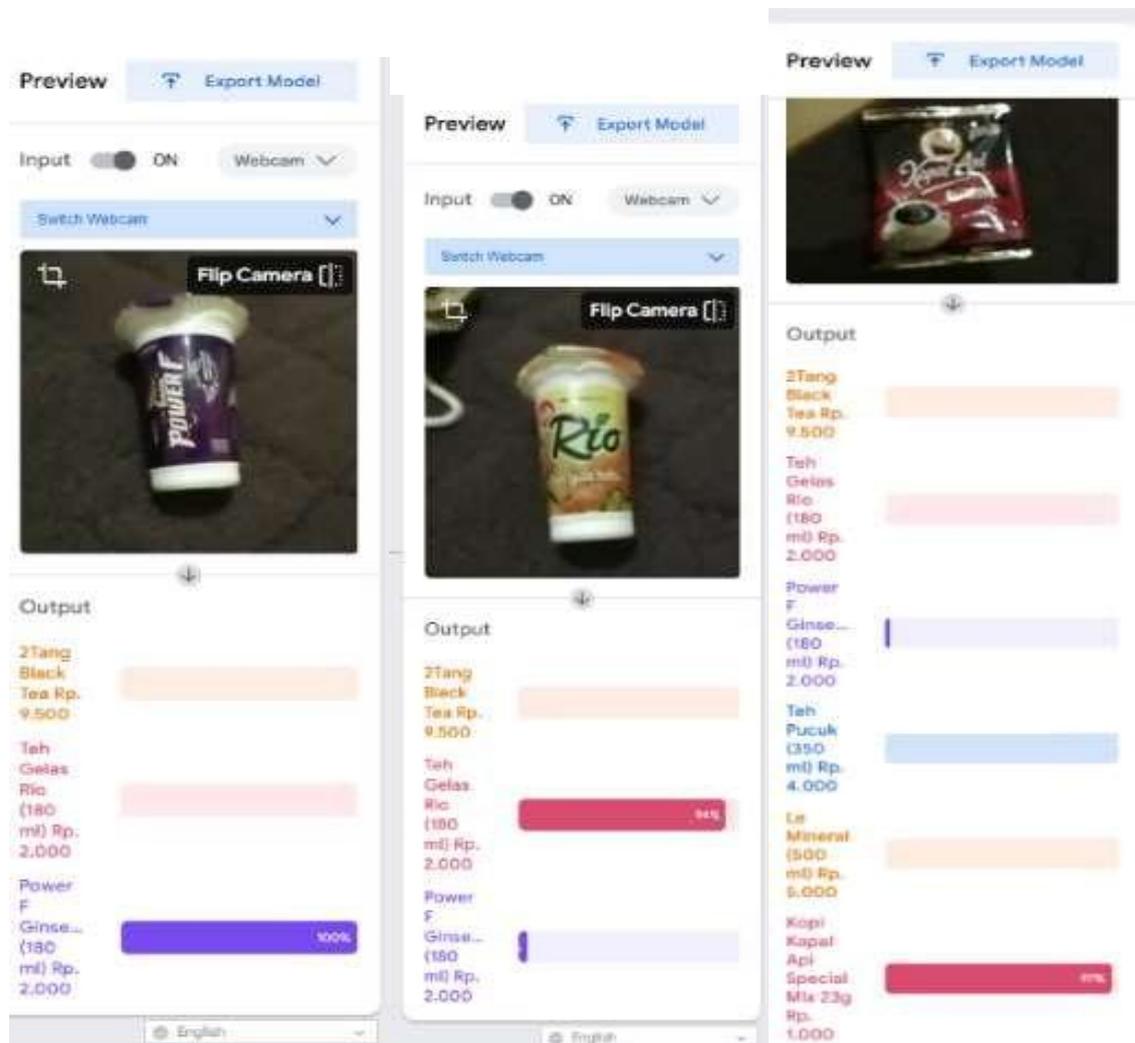
Gambar 16. Tampilan Teachable Machine uji data *training*.  
Sumber: (Penelitian, 2024)

$H_1$ : Akurasi sistem pendeteksi akan bervariasi berdasarkan jenis barang yang dideteksi, dengan beberapa barang lebih mudah dikenali daripada yang lain.

Jawaban:

Diterima: Berdasarkan gambar ini bawah ini dan gambar-gambar sebelumnya menyatakan bahwa  $H_1$  diterima di karenakan ada variasi akurasi berdasarkan jenis barang dan juga ada beberapa barang yang lebih mudah di kenali oleh sistem, dalam catatan bervariasi tergantung pengambilan gambarnya jadi bisa bervariasi akurasinya.

- Teh 2Tang black Tea: 100%
- Power F Ginseng: 100%
- Teh Gelas Rio: 94%
- Kopi Kapal Api: 97%



Gambar 17. Tampilan Teachable Machine uji data *training*.  
 Sumber: (Penelitian, 2024)

## Hasil Pengujian

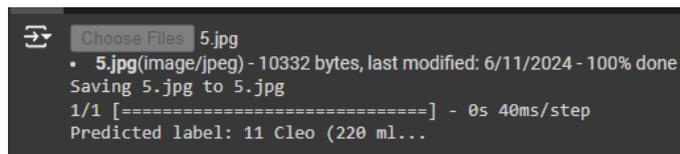
Di dalam penelitian ini akan menampilkan pengujian model di Google Colab dan di aplikasi mobile.

### 1. Uji Model

Sehabis meng *export* model di Teachable Machine disini akan saya uji kinerja modelnya di Google Colab dengan menggunakan *Inferensi Model*.



Gambar 18. Tampilan *output inferensi model*.  
 Sumber: (Penelitian, 2024)



Gambar 19. Tampilan *output interferensi* model.  
 Sumber: (Penelitian, 2024)

Dari Gambar 18 dan 19, terlihat bahwa model masih kurang akurat dalam prediksi. Beberapa barang yang tidak ada di dataset, seperti "botol minum," dikenali oleh sistem sebagai barang lain yang ada di dataset, seperti minuman Power F Ginseng.

## 2. Penggunaan Metode *Dropout Regularization*

Selanjutnya penelitian ini akan menggunakan metode *Dropout regularization* untuk meningkatkan akurasi model saya agar dapat mendapatkan hasil yang memuaskan, untuk hasil dari *Dropout Regularization* nya bisa di lihat di gambar di bawah ini.

```

[15] .....
Total params: 570508 (2.86 MB)
Trainable params: 525420 (2.80 MB)
Non-trainable params: 45088 (55.00 KB)
model: "Sequential_"
-----
Layer (type)                Output Shape              Param #
-----
sequential_26 (Sequential)  (None, 1024)              418208
sequential_27 (Sequential)  (None, 10)                125760
-----
Total params: 570508 (2.86 MB)
Trainable params: 525420 (2.80 MB)
Non-trainable params: 45088 (55.00 KB)
-----
Epoch 1/20
10/20 [====] - 131s 3s/step - loss: 0.2791 - accuracy: 0.9167 - val_loss: 0.3761 - val_accuracy: 0.9084
Epoch 2/20
10/20 [====] - 98s 3s/step - loss: 5.5805e-04 - accuracy: 1.0000 - val_loss: 0.2458 - val_accuracy: 0.9891
Epoch 3/20
10/20 [====] - 107s 3s/step - loss: 7.1851e-04 - accuracy: 1.0000 - val_loss: 0.2417 - val_accuracy: 0.9110
Epoch 4/20
10/20 [====] - 98s 3s/step - loss: 1.5996e-04 - accuracy: 1.0000 - val_loss: 0.2514 - val_accuracy: 0.9128
Epoch 5/20
10/20 [====] - 70s 3s/step - loss: 0.1089e-05 - accuracy: 1.0000 - val_loss: 0.2213 - val_accuracy: 0.9127
Epoch 6/20
10/20 [====] - 395s 3s/step - loss: 0.4383e-05 - accuracy: 1.0000 - val_loss: 0.1941 - val_accuracy: 0.9120
Epoch 7/20
10/20 [====] - 310s 3s/step - loss: 7.0169e-05 - accuracy: 1.0000 - val_loss: 0.1861 - val_accuracy: 0.9178
Epoch 8/20
10/20 [====] - 96s 3s/step - loss: 1.0517e-04 - accuracy: 1.0000 - val_loss: 0.2167 - val_accuracy: 0.9131
Epoch 9/20
10/20 [====] - 388s 3s/step - loss: 0.6483e-05 - accuracy: 1.0000 - val_loss: 0.1169 - val_accuracy: 0.9195
Epoch 10/20
10/20 [====] - 381s 3s/step - loss: 0.5807e-05 - accuracy: 1.0000 - val_loss: 0.1531 - val_accuracy: 0.9342
Epoch 11/20
10/20 [====] - 32s 920ms/step - loss: 0.1359 - accuracy: 0.9381
Test loss: 0.1108/394444/1s
Test accuracy: 0.9381/394444/1s
17:20:00 [Info] 1: 100 Galaxi (1...), 2: Power F Ginseng (1...), 3: Teh Pucuk (1...), 4: Air Mineral (1...), 5: Gula Kapas (1...), 6: Gula Gula (1...), 7: Gula
    
```

Gambar 20. Tampilan Google Colab hasil dari *Dropout Regularization*.  
 Sumber: (Penelitian, 2024)

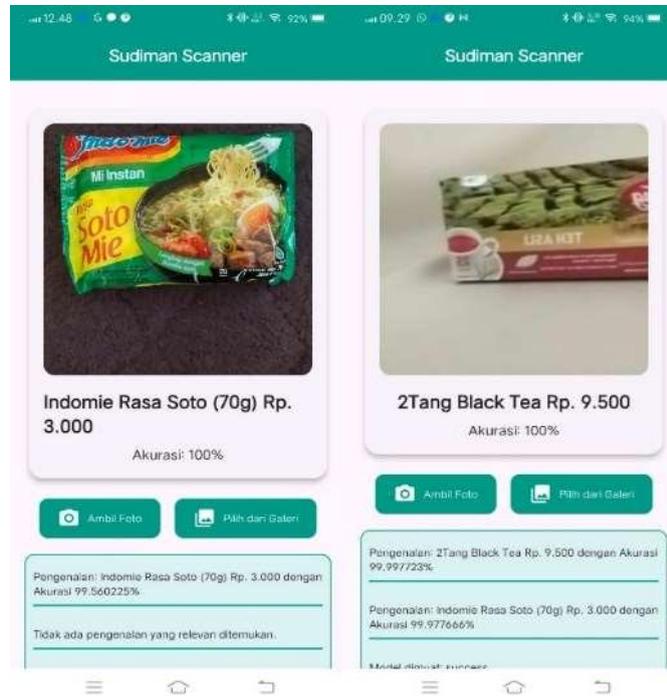
Model yang dikembangkan mencapai akurasi 93.81%, melampaui target 80% yang ditetapkan di BAB I. Ini menunjukkan bahwa sistem berhasil mendeteksi nama dan harga barang dengan akurasi tinggi dan dapat diimplementasikan pada perangkat mobile. Hasil ini mendukung kesimpulan bahwa tujuan penelitian telah tercapai.

## 3. Pengujian Sampel Menggunakan Aplikasi Mobile

Untuk pengujian terakhir yaitu pengujian model pada aplikasi mobilnya apakah model nya berjalan dengan baik atau tidak.

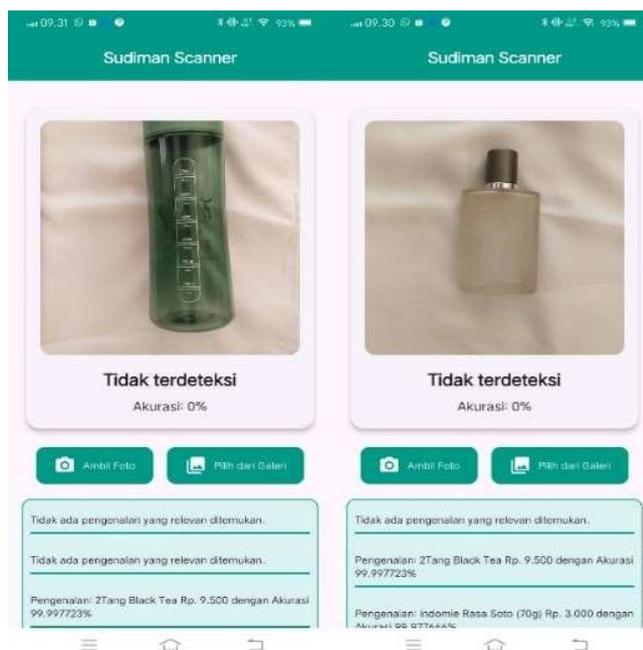
- Pengujian model dengan menggunakan "Pilih dari Galeri" yaitu meng *input* gambar

yang berada di dalam dataset, dengan contoh sample Indomie Rasa Soto dan 2Tang Black Tea:



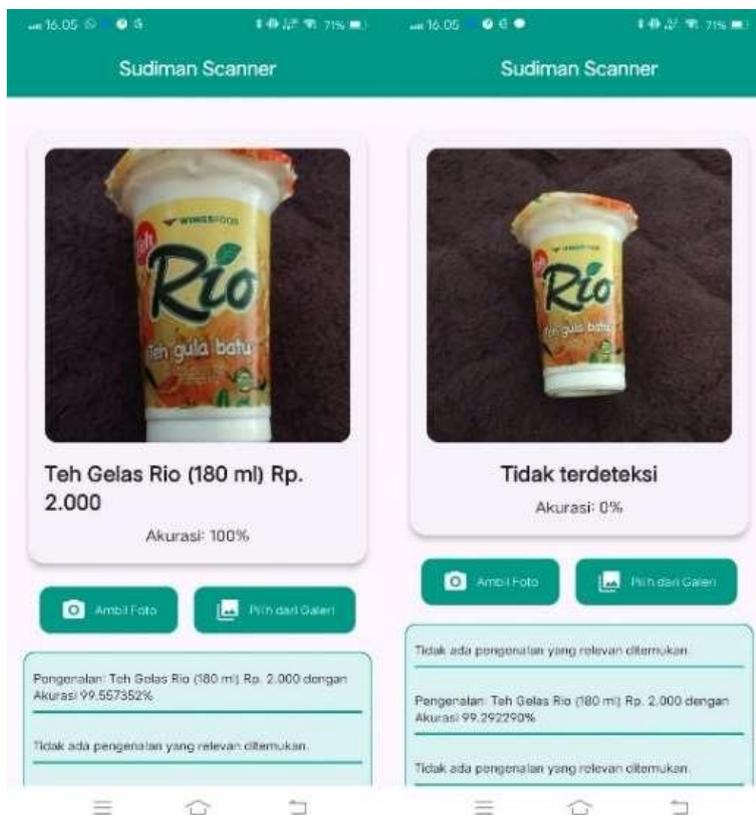
Gambar 21. Tampilan hasil Sudiman Scanner data testing proyek.  
Sumber: (Penelitian, 2024)

- Pengujian model dengan barang yang tidak ada di dalam dataset, dengan contoh barang yaitu botol minum dan botol parfum:



Gambar 22. Tampilan hasil Sudiman Scanner data testing proyek.  
Sumber: (Penelitian, 2024)

Pengujian menggunakan “Ambil Foto” pengambilan foto langsung dari kamera smartphone:



Gambar 23. Tampilan hasil Sudiman Scanner data testing proyek.  
Sumber: (Penelitian, 2024)

Dengan demikian adalah pengujian model apakah berjalan dengan baik atau tidak di aplikasi mobile nya. Bisa di lihat di gambar di atas bahwa barang yang ada di dalam model bisa di kenali oleh sistem dan barang yang tidak ada di model tidak bisa di kenali oleh sistem, dengan catatan terkadang barang yang dapat di kenali oleh sistem dapat tidak terdeteksi dari bagaimana cara pengambilan gambarnya jika menggunakan “Ambil Foto” bisa di lihat dari Gambar di atas jadi untuk proyek ini masih ada kekurangannya dan akurasi nya masih bervariasi.

#### D. Kesimpulan

Penelitian ini berhasil mengembangkan sistem pendeteksi nama dan harga barang di Toko Sudiman menggunakan Teachable Machine dengan hasil yang memuaskan. Sistem ini menunjukkan akurasi tinggi dalam mengenali barang di dataset, dengan beberapa kelas mencapai akurasi 100%. Uji hipotesis membuktikan kemampuan sistem mendeteksi barang dengan akurasi di atas 80%, bahkan dalam pencahayaan kurang ideal. Model yang diuji di Google Colab mencapai akurasi 93.81% dan berfungsi dengan baik dalam aplikasi mobile, meskipun masih terdapat kekurangan yang bisa dikembangkan lebih lanjut.

## Daftar Pustaka

- B. M. Wantania, B., R.U.A. Sompie, S., & D. Kambey, F. (2020). Penerapan Pendeteksian Manusia Dan Objek Dalam Keranjang Belanja Pada Antrian Di Kasir. *Jurnal Teknik Informatika*, 15(2).
- Baihaqi, M. B., Litianianda, Y., & Triyanto, A. (2022). IMPLEMENTASI TENSOR FLOW LITE PADA TEACHABLE UNTUK IDENTIFIKASI TANAMAN AGLONEMA BERBASIS ANDROID. *KOMPUTEK*, 6(1). <https://doi.org/10.24269/jkt.v6i1.1143>
- Chazar, C., & Rafsanjani, M. H. (2022). Penerapan Teachable Machine Pada Klasifikasi Machine Learning Untuk Identifikasi Bibit Tanaman. *Prosiding Seminar Nasional Inovasi Dan Adopsi Teknologi (INOTEK)*, 2(1). <https://doi.org/10.35969/inotek.v2i1.207>
- Dacipta, P. N., & Putra, R. E. (2022). Sistem Klasifikasi Limbah Menggunakan Metode Convolutional Neural Network (CNN) Pada Webservice Berbasis Framework Flask. *Journal of Informatics and Computer Science (JINACS)*, 3(04). <https://doi.org/10.26740/jinacs.v3n04.p394-402>
- Eduardo Christianto, A. M. S. F. D. K. (2021). Implementation of Convolutional Neural Network on Images for Starlings Classification. *Jurnal Teknik Informatika*, ?(?), 1–8.
- Géron, A., & Russell, Rudolph. (2019). Machine learning step-by-step guide to implement machine learning algorithms with Python. In *O'Reilly Media, Inc* (p. 7).
- Harditya, A. (2020). *Indonesian Sign Language (BISINDO) As Means to Visualize Basic Graphic Shapes Using Teachable Machine*. <https://doi.org/10.2991/assehr.k.201202.045>
- Kharis, S. A. A., & Zili, A. H. A. (2022). Learning Analytics dan Educational Data Mining pada Data Pendidikan. *JURNAL RISET PEMBELAJARAN MATEMATIKA SEKOLAH*, 6(1). <https://doi.org/10.21009/jrpms.061.02>
- Millstein, F. (2018). Deep Learning with Keras Beginner's Guide To Deep Learning With Keras. In *CreateSpace Independent Publishing Platform* (p. 16).
- Nasha Hikmatia A.E., & Zul, M. I. (2021). Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow. *Jurnal Komputer Terapan*, 7(1). <https://doi.org/10.35143/jkt.v7i1.4629>
- Ningsih, P. T. S., Gusvarizon, M., & Hermawan, R. (2022). Analisis Sistem Pendeteksi Penipuan Transaksi Kartu Kredit dengan Algoritma Machine Learning. *Jurnal Teknologi Informatika Dan Komputer*, 8(2). <https://doi.org/10.37012/jtik.v8i2.1306>