

SISTEM SELF-CHECKOUT PRODUK SEMBAKO MENGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORKS

Gloria Valerie Lao¹, Lina²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail: ¹gloria.535200049@stu.untar.ac.id ²lina@untar.ac.id

Abstrak

Terdapat banyak toko kelontong dari yang kecil hingga yang besar. Toko kelontong menjual berbagai produk mulai dari sayuran, buah-buahan, makanan instan, makanan ringan dan lain sebagainya. Toko kelontong yang besar atau dikenal dengan grocery store, dapat memiliki banyak pengguna pada saat yang bersamaan. Terdapat beberapa loket kasir yang disediakan untuk melayani pelanggan selama proses pembayaran. Bagi pelanggan yang memiliki banyak barang belanjaan akan memakan waktu yang lama saat dikasir dan dapat menyebabkan antrian yang panjang. Sistem akan menerima input citra yang didalamnya terdapat beberapa produk. Sistem ini akan melakukan pendeteksian dan pengenalan produk dengan menggunakan metode Faster R-CNN dan arsitektur ResNet-101. Setelah produk berhasil diidentifikasi, akan ditampilkan daftar dari produk beserta harga dan total harga keseluruhan. Pendeteksian dan pengenalan produk untuk 1 produk yang berada dalam 1 frame mendapatkan akurasi sebesar 84%, untuk 3 produk yang berada dalam 1 frame mendapatkan akurasi sebesar 44%, dan untuk 5 produk yang berada dalam 1 frame mendapatkan akurasi sebesar 12%.

Kata kunci— *Faster R-CNN, Pendeteksian, Produk belanjaan, Self-checkout, Toko kelontong*

Abstract

There are many grocery stores from small to large. Grocery stores sell a variety of products ranging from vegetables, fruits, instant foods, snacks and so on. A large grocery store or known as a grocery store, can have many users at the same time. There are several cashier counters provided to serve customers during the payment process. For customers who have a lot of groceries, it will take a long time at checkout and can cause long queues. The system will receive image input in which there are several products. This system will perform product detection and recognition using the Faster R-CNN method and the ResNet-101 architecture. Once the product has been identified, a list of products will be displayed along with the price and total price. Product detection and recognition for 1 product in 1 frame gets an accuracy of 84%, for 3 products in 1 frame gets an accuracy of 44%, and for 5 products in 1 frame gets an accuracy of 12%.

Keywords— *Faster R-CNN, Detection, Product groceries, Self-checkout, Grocery store*

1. PENDAHULUAN

Saat ini perkembangan toko kelontong berkembang sangat pesat, terdapat banyak sekali toko kelontong dari yang kecil hingga yang besar. Toko kelontong adalah pengembangan dari pasar yang bertempat di dalam sebuah bangunan yang menjual berbagai produk. Produk tersebut

dapat berupa sayuran, buah-buahan, makanan instan, makanan ringan, peralatan rumah tangga, obat-obatan dan lain sebagainya [1]. Toko kelontong sangat mudah ditemukan saat ini, baik diluar maupun di dalam pusat perbelanjaan, sehingga sangat mudah untuk diakses oleh pembeli.

Di toko kelontong yang besar atau yang lebih dikenal dengan *grocery store*, dapat memiliki banyak orang yang berbelanja pada saat yang bersamaan. Pada setiap kemasan produk terdapat sebuah barcode yang akan dipindai oleh kasir satu persatu menggunakan mesin pemindai, kemudian nama produk serta harga produk akan tercetak di mesin kasir. Untuk pembeli yang memiliki banyak barang belanjaan, proses pemindaian barang dapat memakan waktu yang lama. Hal tersebut dapat menyebabkan antrian panjang, dapat mengurangi kepuasan pelanggan dan membuat kurang efisien jika membeli langsung dari *grocery store*.

Terdapat beberapa toko yang telah memanfaatkan sistem *self-checkout* dengan berbagai cara seperti menggunakan teknologi RFID *barcode*, yang dimana pelanggan tinggal memasukkan barang kedalam kotak yang telah disediakan lalu proses scan berjalan secara otomatis seperti pada Uniqlo dan Decathlon. Terdapat juga penerapan sistem *self-checkout* yang diterapkan oleh Gramedia dengan cara menyusun produk terlebih dahulu pada kotak yang telah disediakan kemudian akan dilakukan proses identifikasi, setelah itu akan menampilkan *list* dari produk tersebut.

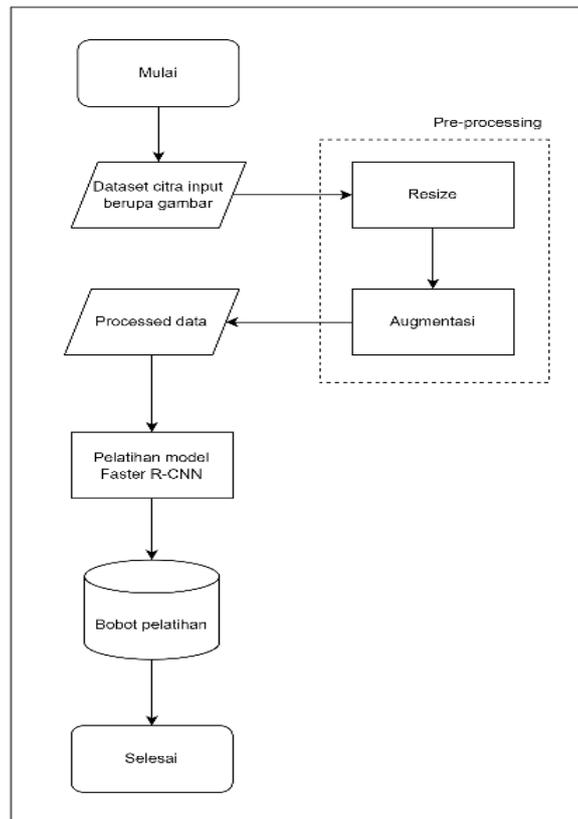
Pemilihan metode Faster R-CNN karena dapat mengekstraksi ciri penting dari setiap citra dengan otomatis [2], serta lebih efisien jika dibandingkan dengan metode lainnya. Dengan menggunakan metode ini, sistem akan dapat mengidentifikasi objek atau produk yang ditangkap oleh kamera.

2. METODE PENELITIAN

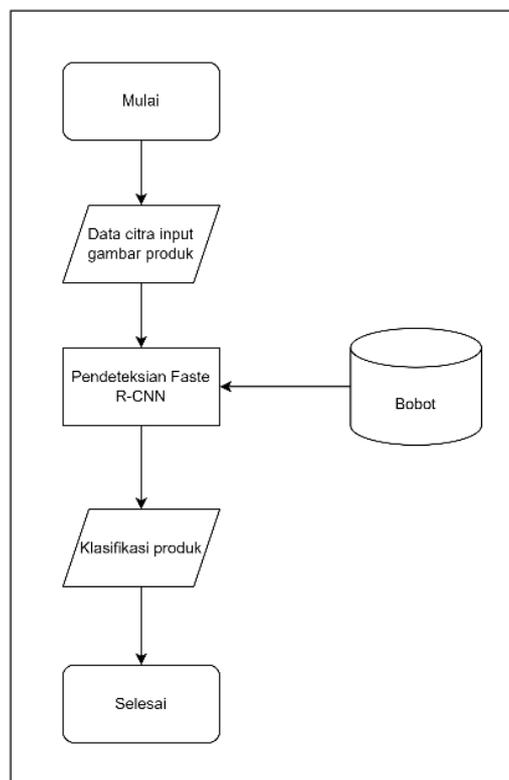
Sistem yang akan dirancang merupakan aplikasi pendeteksian jenis produk belanjaan berbasis desktop sehingga dapat dilakukannya *self-checkout*. Sistem ini menerapkan metode Faster R-CNN dengan menggunakan arsitektur ResNet-101. Sistem akan menerima input berupa citra produk yang telah diambil menggunakan kamera, maupun pengambilan gambar secara langsung. Tahapan pada proses pelatihan dapat dilihat pada Gambar 1, dan proses pengujian dapat dilihat pada Gambar 2.

Pada tahap pelatihan, *dataset* yang berupa input citra gambar akan melalui tahap *pre-processing* data. Tahap ini akan dilakukan proses *resizing* untuk mengubah ukuran citra, setelah *resize* maka akan dilakukan proses augmentasi, yang dimana akan dilakukan rotasi pada citra. Tahap selanjutnya akan dilakukan proses pelatihan menggunakan metode Faster R-CNN dan arsitektur ResNet-101.

Pada tahap pengujian, produk akan difoto sebagai input lalu akan dilakukan pendeteksian dan pengenalan produk dengan menggunakan metode Faster R-CNN. Untuk data pengujian akan diambil secara langsung menggunakan kamera.



Gambar 1 Proses pelatihan



Gambar 2 Tahap pengujian

2.1 Dataset

Jumlah data yang digunakan yaitu 2.500 gambar citra produk. Data yang digunakan berupa data dalam bentuk citra dua dimensi. Data dibagi menjadi 2 yaitu citra latih dan citra validasi. Citra latih adalah citra yang digunakan untuk melatih model Faster R-CNN. Citra validasi adalah citra yang digunakan untuk melakukan evaluasi model Faster R-CNN yang telah dilatih. Contoh dari data latih dan data uji dapat dilihat pada Gambar 3 dan Gambar 4.



Gambar 3 Contoh data latih



Gambar 4 Contoh data uji

Citra latih dan validasi terdiri dari 10 kelas dan setiap kelas memiliki jumlah 200 citra. Dataset diambil secara manual dengan mengambil gambar produk. Data yang digunakan dibagi menjadi 2 bagian yaitu data latih sebanyak 80%, dan data validasi sebanyak 20%. Untuk data uji terdapat 150 gambar citra. Rincian data latih dan data validasi dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1 Rincian data latih

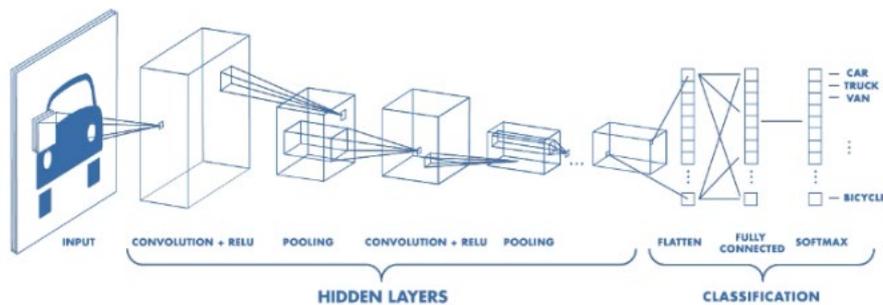
| Nama Produk | Jumlah |
|--------------------|---------------|
| Biskuit Selamat | 204 |
| Chitato Lite | 202 |
| Chitato Sapi | 202 |
| Fitbar | 202 |
| Nissin Cracker | 203 |
| Pocari | 204 |
| Pop Mie | 201 |
| Roma Kelapa | 203 |
| Teh Botol | 205 |
| Chiki Twist | 200 |
| Total | 2026 |

Tabel 2 Rincian data validasi

| Nama Produk | Jumlah |
|-----------------|------------|
| Biskuit Selamat | 50 |
| Chitato Lite | 50 |
| Chitato Sapi | 50 |
| Fitbar | 50 |
| Nissin Cracker | 50 |
| Pocari | 50 |
| Pop Mie | 50 |
| Roma Kelapa | 50 |
| Teh Botol | 50 |
| Chiki Twist | 50 |
| Total | 500 |

2.2 CNN

Convolutional Neural Network (CNN) merupakan salah satu model dari *Deep Neural Network* yang paling sering digunakan untuk menganalisis citra visual dengan menetapkan bobot serta bias objek yang berada dalam gambar input dan dapat membedakan satu dengan yang lain [3]. CNN sangat mirip dengan *Neural Network* biasa seperti, terdiri dari neuron yang memiliki bobot serta bias yang dapat dipelajari. Tahapan pre-processing yang dibutuhkan dalam CNN jauh lebih rendah dibandingkan dengan algoritma klasifikasi yang lainnya. Proses dari model CNN dapat dilihat pada Gambar 5.



Gambar 5 CNN [4]

Terdapat tiga jenis lapisan pada CNN yang sering digunakan yaitu, lapisan *convolutional* yang merupakan dasar dari arsitektur CNN yang melakukan ekstraksi fitur, yang terdiri dari *convolutional operation* dan *activation function*. Yang kedua merupakan lapisan *pooling*, yang dimana lapisan ini dimasukkan secara berkala di antara lapisan *convolutional* berturut-turut. Lapisan ini berfungsi untuk mengurangi ukuran spasial representasi secara progresif agar jumlah parameter dapat dikurangi dan perhitungan dalam jaringan, serta dapat menghindari *overfitting*. Yang terakhir merupakan lapisan *fully-connected*, yang dimana *output* dari *feature map* akan diubah menjadi *array* angka satu dimensi dan terhubung ke satu atau lapisan *fully-connected* lainnya, yang dimana setiap input terhubung ke setiap *output* dengan bobot yang dapat dipelajari

2.2 Faster R-CNN

Faster Region Convolutional Neural Network (Faster R-CNN) adalah salah satu algoritma deep learning yang digunakan untuk mendeteksi objek pada citra. Faster R-CNN merupakan perkembangan dari *Region Based Convolutional Neural Networks* (R-CNN) dan *Fast Region-based Convolutional Network* (Fast R-CNN).

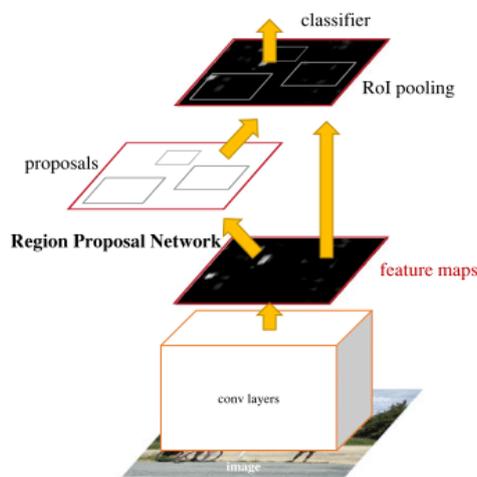
Metode Faster R-CNN menggunakan ide baru, yaitu *Region Proposal Network* (RPN) yang menggantikan metode *selective search* yang digunakan pada metode R-CNN dan Fast R-CNN. *Region Proposal Network* akan menerima input berupa sebuah gambar dengan ukuran berapa pun dan menghasilkan output berupa wilayah-wilayah yang diajukan dan nilai *objectness* dari wilayah yang diajukan. Nilai *objectness* (*objectness score*) merupakan nilai yang menunjukkan seberapa yakin wilayah yang diajukan di dalamnya terdapat sebuah objek. Arsitektur Faster R-CNN dapat dilihat pada Gambar 6.

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|------------|-------------|---|---|---|--|--|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| | | 3×3 max pool, stride 2 | | | | |
| conv2.x | 56×56 | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ |
| conv3.x | 28×28 | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$ |
| conv4.x | 14×14 | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$ |
| conv5.x | 7×7 | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | 1.8×10^9 | 3.6×10^9 | 3.8×10^9 | 7.6×10^9 | 11.3×10^9 |

Gambar 6 Arsitektur Faster R-CNN [5]

2.2 ResNet

ResNet (*Residual Network*) merupakan Jaringan Saraf Tiruan (JST) dari jenis yang dibangun pada konstruksi yang dikenal dari sel piramidal dalam korteks serebral [6]. ResNet memiliki 5 jenis *layer* dengan dengan jumlah masing-masing, yaitu ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152. Dapat dilihat arsitektur dalam lapisan ResNet pada Gambar 7.



Gambar 7 Arsitektur ResNet [7]

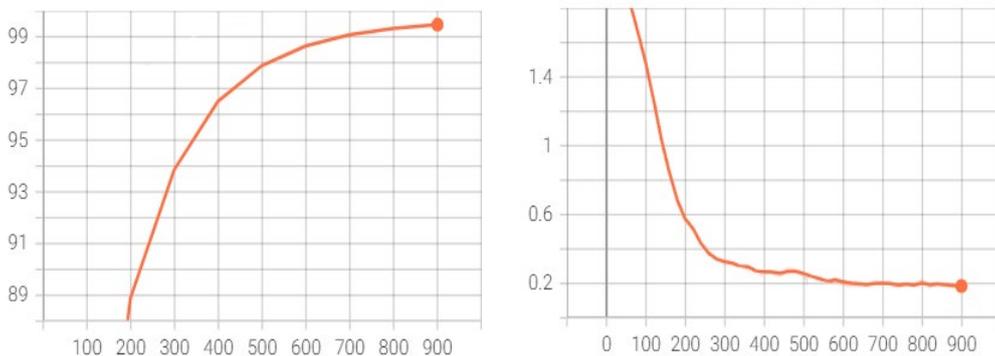
3. HASIL DAN PEMBAHASAN

Pengujian dilakukan terhadap model Faster R-CNN yang telah dilatih. Pelatihan dilakukan menggunakan data latih yang telah dikumpulkan sendiri. Pengujian dilakukan dengan menggunakan 4 buah model yang jumlah *epoch* serta *batch size* telah divariasikan. Tabel hasil pengujian model Faster R-CNN dapat dilihat pada Tabel 3.

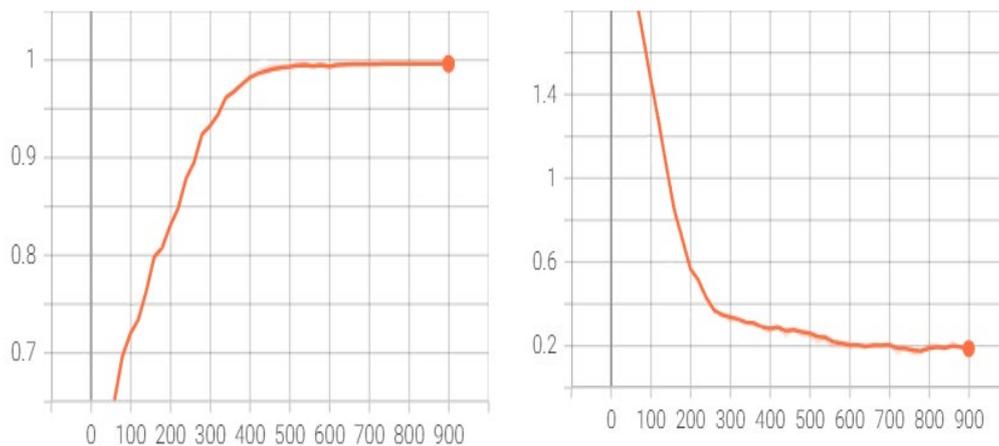
Tabel 3 Hasil pengujian model

| No. | Model | Batch Size | Epoch | Akurasi Latih | Loss Latih | Akurasi Validasi | Loss Validasi |
|-----|---------|------------|-------|---------------|------------|------------------|---------------|
| 1. | Model 1 | 8 | 800 | 97% | 0,027 | 99,76% | 0,231 |
| 2. | Model 2 | 16 | 800 | 97% | 0,196 | 99,64% | 0,222 |
| 3. | Model 3 | 64 | 800 | 96% | 0,191 | 99,62%% | 0,173 |
| 4. | Model 4 | 64 | 900 | 98% | 0,029 | 81,32% | 0,024 |

Adapun hasil terbaik diantara ke-4 model yang telah dilatih serta dilakukan validasi dan yang akan dilakukan uji coba kepada data set pengujian adalah dengan menggunakan model 4 yaitu dengan *batch size* berukuran 64, *epoch* 900 dengan tingkat akurasi latih sebesar 98% dan akurasi validasi sebesar 81,32%. Untuk grafik akurasi serta loss latih dapat dilihat pada Gambar 8, sedangkan untuk grafik akurasi dan latih validasi dapat dilihat pada Gambar 9.



Gambar 8 Grafik akurasi dan loss latih model 4



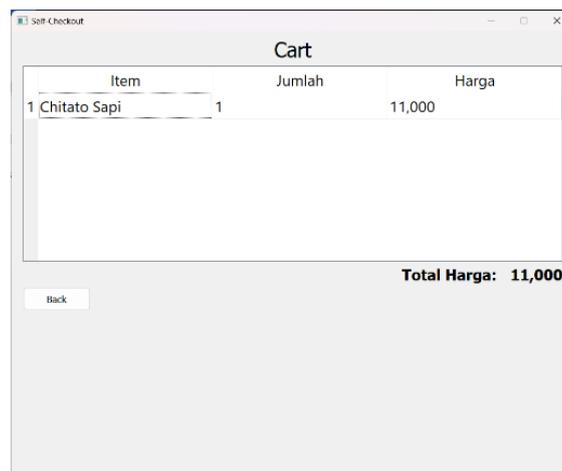
Gambar 9 Grafik akurasi dan loss validasi model 4

Pengujian terhadap hasil keluaran program ditujukan untuk mengetahui akurasi dari sistem self-checkout dari metode yang digunakan. Pada dataset pengujian terdapat 150 citra uji yang terbagi menjadi 3 skenario. Untuk skenario pertama terdapat 1 produk pada citra, pada skenario kedua terdapat 3 produk pada citra dan skenario ketiga terdapat 5 produk pada citra dengan jumlah dataset uji sebanyak 50 citra setiap skenario.

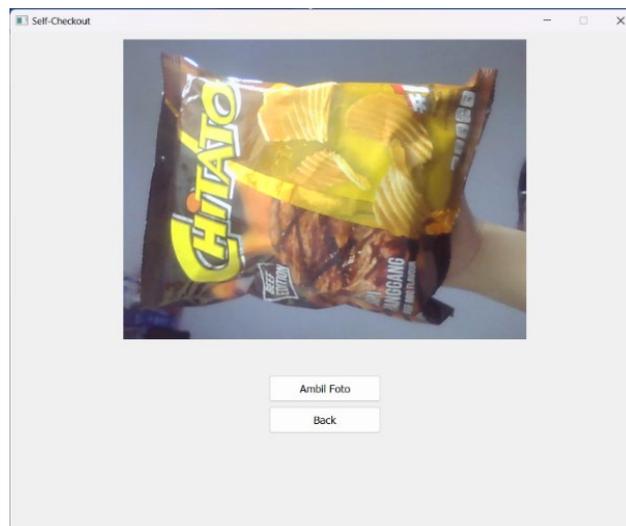
Tabel 4 Hasil pengujian keluaran program

| No. | Skenario Tes | Jumlah Data | Jumlah Benar | Akurasi |
|-----|------------------------------------|-------------|--------------|---------|
| 1. | Satu objek dalam satu <i>frame</i> | 50 | 42 | 84% |
| 2. | Tiga objek dalam satu <i>frame</i> | 50 | 22 | 44% |
| 3. | Lima objek dalam satu <i>frame</i> | 50 | 6 | 12% |

Pada Tabel 4 diatas menunjukkan bahwa untuk satu objek yang berada dalam satu frame didapatkan akurasi sebesar 84%, untuk tiga objek yang berada dalam satu frame didapatkan akurasi sebesar 44%, dan untuk lima objek yang berada dalam satu frame didapatkan akurasi sebesar 12%. Contoh *input* serta ouput dari program dapat dilihat pada Gambar 10 dan Gambar 11.



Gambar 10 Contoh *input* program



Gambar 11 Contoh *output* program

4. KESIMPULAN

Kesimpulan yang didapatkan dari hasil pengujian yang dilakukan terhadap program sistem *self-checkout* dari citra produk dengan menggunakan metode Faster R-CNN adalah sebagai berikut:

1. Akurasi tertinggi didapatkan pada model 4 dengan menggunakan *batch size* 64 dan 900 *epoch*.
2. Untuk pengujian 1 produk yang terdapat dalam 1 frame mendapatkan akurasi sebesar 84%, sedangkan pengujian 3 produk yang terdapat dalam 1 frame mendapatkan akurasi sebesar 44%, dan untuk pengujian 5 produk yang terdapat dalam 1 frame mendapatkan akurasi sebesar 12%.

DAFTAR PUSTAKA

- [1] Suciati, 2012, Jenis gerai atau toko, *Jurnal Pendidikan Kesejahteraan Keluarga*.
- [2] Herlambang, M. F., 2020, TA: PENGENALAN KARAKTER HURUF BRAILLE DENGAN METODE CONVOLUTIONAL NEURAL NETWORK, *Tesis*, Fakultas Teknik Informatika, Institut Teknologi Nasional Bandung.
- [3] Mandal, M., 2021, Introduction to Convolutional Neural Networks (CNN), <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>, diakses tanggal 25 Agustus 2023.
- [4] Saha, S., 2018, A Guide to Convolutional Neural Networks — the ELI5 way, <https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/>, diakses tanggal 25 Agustus 2023.
- [5] Ren, S., He. K., Girshick, R., Sun, J., 2015, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Advances in neural information processing systems*, Vol. 28.
- [6] Harahap, M., Laia, E. M., Sitanggang, L. S., Sinaga, M., Sihombing, D. F., Husein, A. M., 2022, Deteksi Penyakit Covid-19 Pada Citra X-Ray Dengan Pendekatan Convolutional Neural Network (CNN), *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, Vol. 6(1), hal 70-77.
- [7] He, K., Zhang, X., Ren, S., Sun, J., 2016, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, Hal 770-778.