

Perancangan dan Implementasi *Self-Checkout System* pada Toko Ritel menggunakan *Convolutional Neural Network (CNN)*

ANHAR, RAHMA ADI PUTRA

Jurusan Teknik Elektro, Universitas Riau, Pekanbaru
Email: anhar@lecturer.unri.ac.id

Received 8 Februari 2023 | *Revised* 15 Maret 2023 | *Accepted* 25 Maret 2023

ABSTRAK

Perkembangan teknologi self-checkout system meningkatkan efektivitas dalam melakukan proses pembayaran. Self-checkout system merupakan fasilitas yang memungkinkan konsumen untuk melakukan pembayaran melalui scanning beberapa produk sekaligus dalam satu waktu dan pengemasan secara mandiri. Penelitian ini mengusulkan self-checkout system menggunakan Convolutional Neural Network (CNN) dan arsitektur model MobileNetV2 dengan metode hamming loss. Perancangan sistem diimplementasikan pada GUI sebagai user interface penelitian. Dataset yang digunakan pada penelitian ini berjumlah 247 citra dengan resolusi 224 x 224 pixels terhadap tiga jenis produk Teh Botol, Indomie dan Chitato. Hasil pengujian menunjukkan nilai akurasi deteksi sebesar 88.8% dengan hamming loss 0.12%. Posisi produk dalam keadaan berjarak dapat meningkatkan nilai akurasi. Sistem GUI pada penelitian ini berhasil mendeteksi produk hanya dalam waktu 1 detik.

Kata kunci: *self-checkout system, convolutional neural network, MobileNetV2*

ABSTRACT

The development of self-checkout system technology increases the effectiveness of the payment process. Self-checkout system is a facility that allows consumers to make payments through scanning several products at the same time and packaging independently. This research proposes a self-checkout system using Convolutional Neural Network (CNN) and MobileNetV2 model architecture with hamming loss method. The system design is implemented on a GUI as a research user interface. The dataset used in this study amounted to 247 images with a resolution of 224 x 224 pixels of three types of bottled tea, Indomie and Chitato products. The test results show a detection accuracy value of 88.8% with a hamming loss of 0.12%. The position of the product in a spaced state can increase the accuracy value. The GUI system in this research successfully detects products in just 1 second.

Keywords: *self-checkout system, convolutional neural network, MobileNetV2*

1. PENDAHULUAN

Kebiasaan berbelanja di Toko Ritel telah menjadi rutinitas masyarakat untuk memenuhi kebutuhan sehari-hari. Dalam proses transaksi di Toko Ritel, layanan kasir digunakan untuk membantu konsumen dalam melakukan pembayaran. Namun, seringkali konsumen harus mengantre untuk mendapatkan pelayanan tersebut **(Tungadi, dkk, 2019)**. Umumnya, antrean di kasir disebabkan oleh dua faktor. Pertama, petugas kasir harus memindai *barcode* pada setiap produk yang dibeli oleh konsumen. Kedua, kasir harus menghitung uang yang dibayarkan oleh konsumen dan kembalian yang harus diberikan **(Sulaiman, dkk, 2016)**. Menurut hasil survei yang dilakukan oleh Capgemini pada Oktober 2019, sebagian besar konsumen merasa kesal terhadap antrean yang panjang di toko ritel. Kekecewaan ini dapat berdampak pada persepsi konsumen terhadap toko ritel dan mengurangi minat untuk belanja kembali di tempat tersebut.

Permasalahan antrean dapat diatasi dengan mengubah metode pembayaran dari sistem tunai menjadi sistem pembayaran mandiri atau *self-checkout system* **(Abana, dkk, 2020)**. *Self-checkout system* merupakan kegiatan berbelanja yang pembayarannya dapat dilakukan sendiri tanpa bantuan seorang kasir **(Suwarno & Lim, 2021)**. Teknologi *self-checkout* yang umum digunakan untuk memindai produk saat ini adalah sistem *Barcode* dan *Radio Frequency Identification* (RFID). Namun, kedua teknologi ini memiliki beberapa kelemahan. Salah satu kelemahan sistem *barcode* adalah proses identifikasi harus dilakukan secara satu persatu dan perangkat *reader* harus diarahkan secara tepat sehingga membutuhkan waktu lebih dan dapat menyebabkan antrean **(Saputra, 2019)**. Sementara, kelemahan RFID ada pada biaya komponen yang mahal serta rentan kehilangan tag RFID **(Hamim, 2018)**.

Beberapa peneliti telah mengembangkan teknologi *self-checkout system* dalam mengatasi permasalahan tersebut. Salah satu arsitektur yang sering digunakan dalam deteksi objek citra ialah *Convolutional Neural Network* (CNN) **(Pathak, dkk, 2018)**. Seperti penelitian oleh **(Bukhari, dkk, 2021)**, mengusulkan sistem *self-checkout system* yang dikembangkan menggunakan model arsitektur *Feedforward Convolutional Neural Network*. Hasil pengujian model CNN yang dikembangkan mencapai akurasi 91,7%. Penelitian lain oleh **(Rigner, 2019)**, membahas perbandingan model untuk deteksi objek pada toko ritel dengan teknik transfer *learning* dari model R-CNN, YOLO dan RetinaNet. Dari ketiga model tersebut, model RetinaNet dipilih karena memiliki akurasi dan waktu proses yang baik. Meskipun demikian, penelitian sebelumnya masih memiliki kekurangan seperti belum dilakukannya proses training model sehingga sistem tidak mampu mengenali atau memahami fitur-fitur dalam data input yang bersifat spasial.

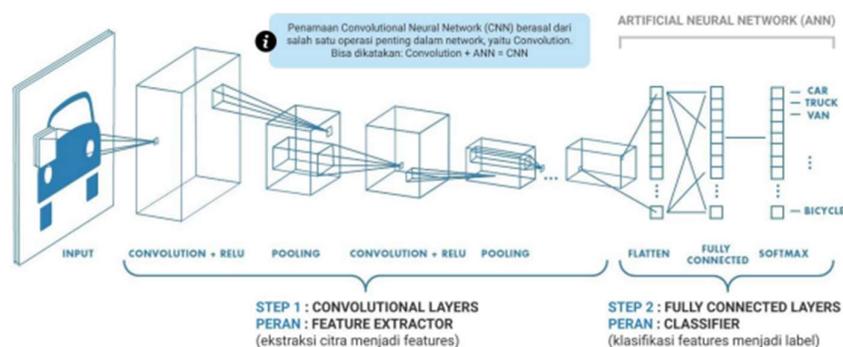
Pada penelitian ini diusulkan rancangan dan implementasi *Self-checkout System* pada Toko Ritel menggunakan *Convolutional Neural network* (CNN). Model deteksi objek pada penelitian ini dibangun menggunakan teknik *transfer learning* karena telah melakukan training model dengan dataset yang baru dan tidak terlalu banyak serta nilai akurasi yang jauh lebih baik **(Rochman & Junaedi, 2020)**. Arsitektur model CNN MobileNetV2 juga dikembangkan, hal ini dikarenakan akurasi yang lebih tinggi serta jumlah komputasi yang lebih rendah **(Sandler, dkk, 2018)**. Selain itu, penelitian ini menerapkan *Graphical User Interface* (GUI) dalam mengimplementasi hasil rancangan sistem. *Self-checkout system* yang dirancang pada penelitian ini didukung oleh Webcam sebagai media dalam mengambil gambar objek yang akan dideteksi. Penelitian ini menggunakan perangkat komputer sebagai media komputasi serta sebagai *user interface* untuk menampilkan data hasil deteksi.

1.1 Image Classification

Image classification merupakan salah satu aplikasi dari machine learning yang bertujuan untuk mengelompokkan citra ke dalam kelas tertentu berdasarkan fitur-fitur yang terdapat pada citra tersebut. Citra dapat dikelompokkan ke dalam kelas yang telah ditentukan sebelumnya, seperti binatang, tumbuhan, atau objek-objek lainnya. Dalam pengelompokan citra, terdapat dua metode yang sering digunakan yaitu *multi-label classification* dan *multi-class classification*. Perbedaan utama antara klasifikasi *multi-class* dan *multi-label* adalah cara mengelompokkan sampel ke dalam kelas-kelas. Pada klasifikasi *multi-class*, masing-masing kelas memiliki peran yang terpisah dan saling eksklusif. Contohnya, sebuah citra hanya dapat dikelompokkan ke dalam satu kelas, seperti binatang atau tumbuhan, tidak bisa keduanya sekaligus. Sedangkan pada klasifikasi *multi-label*, masing-masing label akan mewakili peran klasifikasi yang berbeda, namun peran tersebut masih saling terhubung satu sama lain sehingga akan menjadi satu kesimpulan yang sama. Contohnya, sebuah citra dapat dikelompokkan ke dalam kelas binatang dan tumbuhan secara bersamaan. Untuk menyelesaikan masalah klasifikasi *multi-class*, algoritma akan memprediksi kelas suatu sampel berdasarkan fitur-fitur yang terdapat pada sampel tersebut, kemudian mengelompokkannya ke dalam satu kelas yang telah ditentukan sebelumnya. Sedangkan untuk menyelesaikan masalah klasifikasi *multi-label*, algoritma akan memprediksi kelas-kelas suatu sampel berdasarkan fitur-fitur yang terdapat pada sampel tersebut, kemudian mengelompokkannya ke dalam lebih dari satu kelas yang telah ditentukan sebelumnya (Prawira, dkk, 2018).

1.2 Convolutional Neural Network

Convolutional neural network (CNN) adalah jenis *neural network* yang dirancang khusus untuk menangani masalah pengolahan citra. CNN merupakan metode yang efektif untuk mengklasifikasikan, mengidentifikasi, dan mengenali pola di dalam citra. CNN mampu memahami detail gambar dengan lebih baik karena memiliki arsitektur yang sesuai dengan cara otak manusia memproses informasi visual. Data yang digunakan pada CNN adalah data dua dimensi, seperti citra atau suara, dan menggunakan operasi konvolusi dalam matriks serta bobot yang berbentuk empat dimensi yang merupakan sekumpulan kernel konvolusi. Dengan sifat proses konvolusi tersebut, CNN hanya dapat digunakan pada data yang memiliki struktur data dua dimensi (Suartika, dkk, 2016).



Gambar 1. Arsitektur Dasar CNN (Prabhu, 2018)

Pada Gambar 1, secara sederhana ada dua bagian utama dalam operasi CNN yaitu pembelajaran fitur dan klasifikasi. Dalam bagian pembelajaran fitur, terdapat tiga proses yang dijalankan, yaitu:

1. Operasi konvolusi, yang merupakan proses utama dalam CNN. Tujuannya adalah untuk mengekstrak fitur dari citra masukan.

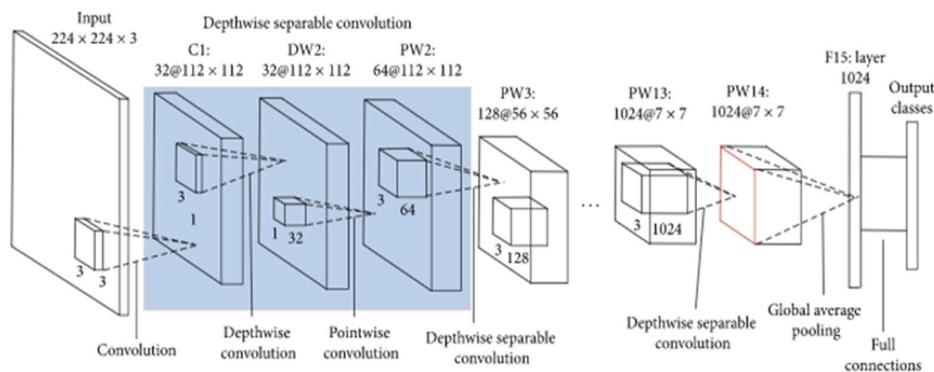
2. ReLu (*Rectified Linear Unit*), yaitu fungsi aktivasi yang memiliki output 0 jika input kurang dari 0. Artinya, jika $x \leq 0$ maka $x = 0$ dan jika $x > 0$ maka $x = x$.
3. *Pooling*, yang digunakan untuk mengurangi jumlah parameter pada peta fitur dan mengambil informasi yang paling penting dengan operasi *down-sampling*. Metode pooling yang umum digunakan adalah *max pooling* dan *average pooling*.

Pada bagian klasifikasi, terdapat tiga proses yaitu:

1. *Flatten*, *layer* ini mengubah output dari hasil *layer* konvolusi menjadi satu kolom *feature vector* untuk digunakan pada *layer* terhubung secara penuh.
2. *Fully connected layer*, *layer* ini mirip dengan *Artificial Neural Network* (ANN) untuk melakukan klasifikasi.
3. *Softmax*, untuk mendapatkan nilai probabilitas dari setiap kelas, nilai probabilitas kelas tertinggi merupakan output prediksi kelas yang diperoleh. Proses-proses tersebut merupakan operasi yang umum digunakan dalam berbagai model CNN.

1.3 MobileNetV2

MobileNetV2 adalah arsitektur *neural network* yang dikembangkan oleh Google untuk digunakan pada perangkat seluler atau ponsel pintar. MobileNetV2 merupakan versi terbaru dari arsitektur MobileNet yang lebih dulu dikembangkan. MobileNetV2 dirancang untuk memiliki ukuran yang kecil sehingga dapat dijalankan dengan cepat di perangkat seluler dengan spesifikasi rendah. MobileNetV2 menggunakan teknik yang disebut dengan *depthwise separable convolution* yang memungkinkan model ini memiliki ukuran yang lebih kecil dibandingkan dengan model CNN lainnya. MobileNetV2 memiliki beberapa fitur lain yang membuatnya lebih efisien dalam hal penggunaan daya seperti penggunaan *bottlenecks* dan penggunaan lapisan residual. MobileNetV2 juga dapat digunakan pada perangkat seluler untuk aplikasi *real-time* seperti pengenalan wajah, pengenalan suara, atau deteksi gerakan.



Gambar 2. Arsitektur MobileNetV2 (Wang, dkk, 2020)

Gambar 2 merupakan arsitektur dasar pada MobileNetV2. Terdiri dari lapisan konvolusi yang disebut dengan *depthwise separable convolution* dan lapisan *fully-connected* (FC). Selain lapisan tersebut, MobileNetV2 juga memiliki beberapa fitur lain yang membuatnya lebih efisien dalam hal penggunaan daya seperti penggunaan *bottlenecks* dan penggunaan lapisan residual (Howard, dkk, 2020).

1.4 Transfer Learning

Transfer learning adalah teknik yang digunakan dalam deep learning untuk mengambil model yang sudah terlatih dan menerapkannya pada masalah yang berbeda. Ia dapat membantu meningkatkan akurasi model karena model yang sudah terlatih telah teruji dan terbukti mampu menangani data yang besar dan umum dengan baik. Pada *transfer learning*, biasanya dilakukan penyesuaian pada lapisan bagian akhir dari model yang digunakan sebagai "bahan baku" untuk membangun model baru. Lapisan tersebut dapat dioptimalkan

sesuai dengan kebutuhan kita dengan cara mengubah beberapa parameter atau menambahkan lapisan baru. Namun, perlu diperhatikan bahwa *transfer learning* tidak selalu cocok untuk setiap dataset yang akan diolah. Pemilihan model yang akan digunakan sebagai "bahan baku" harus sesuai dengan dataset yang akan diolah agar dapat memberikan hasil yang baik (Rochman & Junaedi, 2020).

1.5 Hamming Loss

Hamming loss merupakan sebuah metode untuk melakukan evaluasi dan digunakan untuk menentukan performansi dari model yang dibuat pada jenis klasifikasi *multi-label*. Dalam metode *hamming loss*, perhitungan yang dilakukan adalah banyaknya jumlah kesalahan klasifikasi terhadap data yang diuji. Performansi dalam metoda ini ditandai dengan representasi nilai dari loss, idealnya nilai *hamming loss* pada sistem adalah nol, yang berarti semakin kecil atau mendekati nol nilai maka semakin bagus performansi dari model yang dibangun. Untuk menghitung *hamming loss* dengan Persamaan (1) (Wiraguna, dkk, 2019).

$$hloss(h) = \frac{1}{p} \sum_{i=1}^p \frac{1}{Q} |h(x_i) \Delta Y_i| \quad (1)$$

Keterangan:

P = jumlah banyaknya data

Q = jumlah banyaknya kelas

$|h(x_i) \Delta Y_i|$ = jumlah banyaknya kesalahan dalam klasifikasi yang terjadi

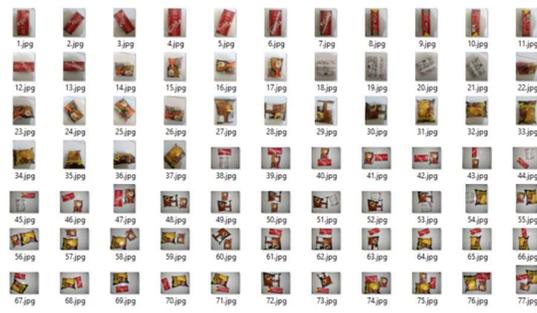
2. METODOLOGI

2.1 Perancangan Model

Model deteksi objek akan dibangun dengan teknik *transfer learning* menggunakan pre-trained model MobileNetV2. Arsitektur MobileNetV2 dari CNN dipilih karena model ini mampu memiliki nilai akurasi yang lebih tinggi namun dengan jumlah komputasi yang lebih rendah, sehingga sangat tepat diaplikasikan pada perangkat mobile atau yang memiliki sumber daya terbatas. Dalam perancangan model ada beberapa tahapan yang dilakukan yaitu sebagai berikut:

2.1.1 Pengumpulan Dataset

Dataset yang digunakan adalah sekumpulan data gambar yang diambil secara langsung menggunakan kamera *smartphone*. Dataset tersebut terdiri dari tiga kelas, yaitu teh botol, Indomie, dan Chitato. Pengambilan data gambar harus memiliki kualitas yang bagus dengan kondisi pencahayaan yang baik, serta pengambilan gambar diambil dari berbagai sudut dan harus terlihat semua. Total jumlah data gambar yang dikumpulkan yaitu 247 gambar dengan format gambar *.jpg.



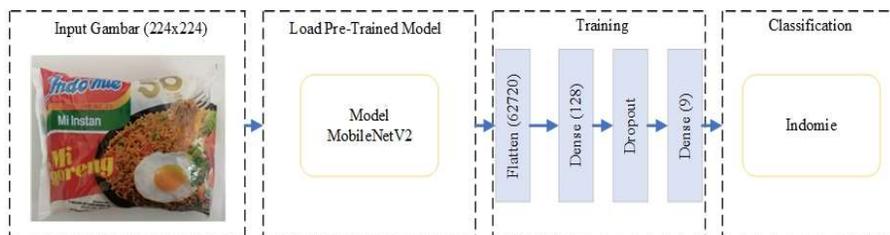
Gambar 3. Persiapan Dataset

2.1.2 Preprocessing

Preprocessing merupakan tahap mengolah data gambar yang telah dikumpulkan sebelumnya untuk dilakukan penyesuaian. Pada *preprocessing* terdapat beberapa tahap yaitu: pelabelan, *resizing* atau mengubah ukuran gambar, dan *split dataset*.

2.1.3 Training Model

Rancangan arsitektur yang digunakan dalam pelatihan model ini terdapat pada Gambar 4.

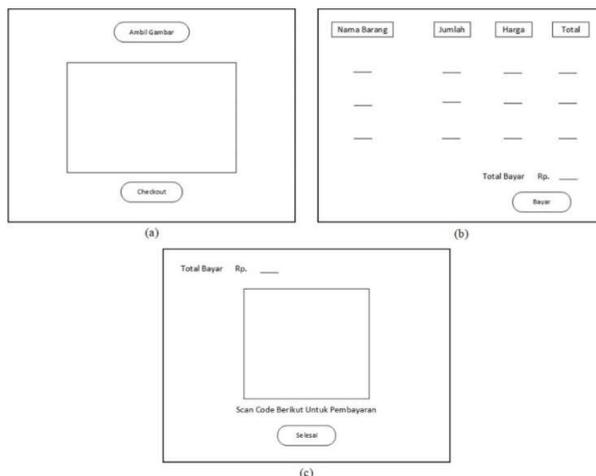


Gambar 4. Rancangan Training Model

Rancangan *training* model pada Gambar 4 menggunakan *transfer learning* dengan arsitektur model MobileNetV2. Input gambar yang digunakan pada dataset berukuran sesuai dengan input MobileNetV2 yaitu 224x224 pixel. Setiap data gambar akan memasuki tahap *training* untuk proses *feature mapping* atau *feature extractor* yang berfungsi untuk mengambil fitur-fitur yang ada pada data citra. Ekstraksi fitur ini menggunakan arsitektur model MobileNetV2 yang sudah terlatih sebelumnya. Sehingga pada tahap ini, setiap citra yang akan diproses akan mengikuti arsitektur MobileNetV2 dalam melakukan operasi konvolusi. Setiap *convolution* akan memiliki konfigurasi yang berbeda-beda, seperti *padding*, *filter*, *channel*, dan *stride*. Proses ini merupakan tahap terpenting dalam pengolahan citra, karena pada tahap ini sebuah model akan dilatih untuk mempelajari pola-pola yang terdapat pada data citra, sehingga diharapkan dapat menghasilkan pengenalan objek yang akurat dengan tingkat akurasi yang tinggi.

2.2 Perancangan *Graphical user interface* (GUI)

Antarmuka pengguna atau GUI dibuat dengan bahasa pemrograman python dan terdiri dari tiga slide utama.



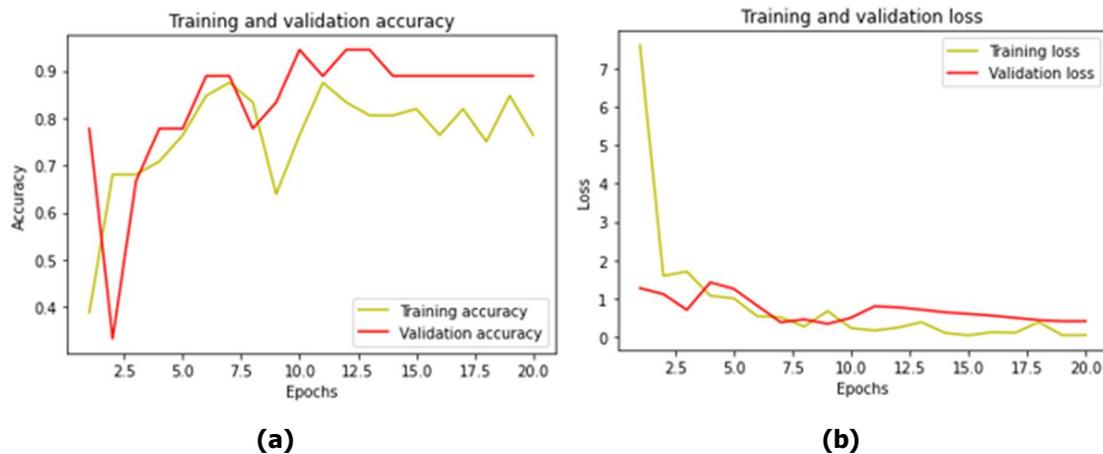
Gambar 5. Rancangan GUI, (a) Slide Pertama, (b) Slide Kedua, (c) Slide Ketiga

Dalam Gambar 5 terdiri dari tiga slide, slide pertama terdiri dari tombol ambil gambar dan tombol *checkout*. Pada slide kedua berfungsi untuk menampilkan hasil scan belanjaan dan tombol pembayaran. Selanjutnya pada slide ketiga untuk menampilkan total hasil belanjaan dan kode pembayaran serta tombol selesai untuk menyelesaikan proses *checkout*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil dan Pembahasan Model

Hasil *training* pada jurnal ini menggunakan learning rate 0.0001 dengan *epoch* 20 dan untuk *batch size* berukuran 32. Dari hasil *training* yang dilakukan, tingkat pembelajaran pada model yang dibangun menghasilkan nilai akurasi sebesar 88.8 %.



Gambar 6. Grafik Akurasi dan Loss

Grafik nilai akurasi pada Gambar 6 (a) mengalami fluktuasi pada saat di *training*, hal ini dikarenakan model sedang mempelajari *feature* atau data yang ada pada citra. Nilai akurasi pada awal *training* sangat rendah yaitu karena pada tahap awal pelatihan, model tersebut belum menemukan pola yang tepat untuk memprediksi hasil. Karena itu, model tersebut masih sering salah dalam memprediksi hasil, sehingga akurasinya akan rendah. Namun, setelah beberapa waktu (*epoch*), model tersebut mulai menemukan pola yang tepat dan kemampuannya untuk memprediksi hasil semakin baik, sehingga akurasinya mulai meningkat. Namun, setelah beberapa waktu, akurasi akan mencapai puncaknya dan kemudian mulai menurun. Ini disebabkan karena model tersebut telah menemukan pola yang tepat, dan model tidak akan dapat menemukan pola yang lebih baik lagi, sehingga akurasinya akan mulai menurun dan stabil. Nilai loss akurasi dan validasi yang baik adalah yang mendekati nol. Semakin kecil nilai loss akurasi dan validasi, semakin baik model tersebut dalam memprediksi hasil yang benar. Namun, nilai loss akurasi dan validasi pada Gambar 6 (a) dan Gambar 6 (b) yang terlalu kecil juga dapat menandakan bahwa model tersebut mungkin mengalami *overfitting*, yaitu keadaan di mana model tersebut terlalu spesifik untuk data pelatihan dan tidak dapat menangani data baru dengan baik.

3.2 Hasil dan Pembahasan Evaluasi Model

Dalam evaluasi dengan metode *hamming loss*, perhitungan dilakukan dengan melihat banyaknya kesalahan prediksi atau klasifikasi terhadap data citra yang diuji. Sebagai contoh, jika seharusnya sebuah data dengan objek Indomie seharusnya diklasifikasikan sebagai label Indomie, namun ketika model memprediksi data tersebut diklasifikasikan sebagai Chitato.

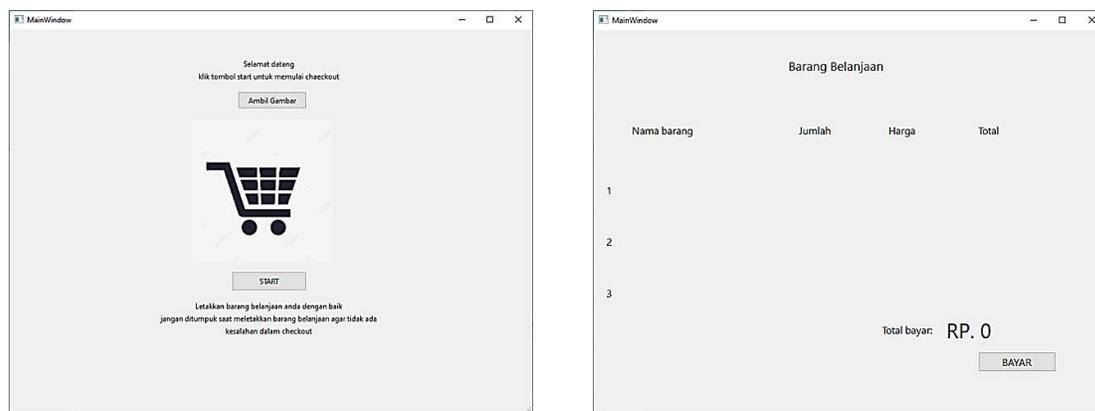
Maka hal tersebut yang diperhitungkan dalam penggunaan metode *hamming loss*. Hasil evaluasi *hamming loss* sebagai berikut.

$$\begin{aligned} \text{Hamming loss} &= \frac{\text{Total Prediksi benar}}{\text{Jumlah data}} \\ &= \frac{44}{50} = 0.88 \end{aligned}$$

Berdasarkan hasil evaluasi dengan *hamming loss*, dari 50 data yang diuji, diperoleh nilai *hamming score* 0,88 sehingga nilai *hamming loss* yang didapatkan yaitu 0,12. Nilai *hamming loss* yang bagus adalah yang mendekati nol. Semakin kecil nilai *hamming loss*, semakin baik model tersebut dalam memprediksi hasil yang benar. Namun, nilai *hamming loss* yang terlalu kecil juga dapat menandakan bahwa model tersebut mungkin mengalami *overfitting*, yaitu keadaan di mana model tersebut terlalu spesifik untuk data pelatihan dan tidak dapat menangani data baru dengan baik. Tetapi jika nilai *hamming loss*nya tinggi, maka menunjukkan bahwa model tersebut tidak mampu memprediksi dengan benar, atau dengan kata lain, model tersebut memiliki tingkat kesalahan yang tinggi. Ini dapat disebabkan oleh beberapa faktor, seperti data yang tidak terlalu baik, model yang tidak cocok dengan data, atau model yang tidak terlatih dengan baik.

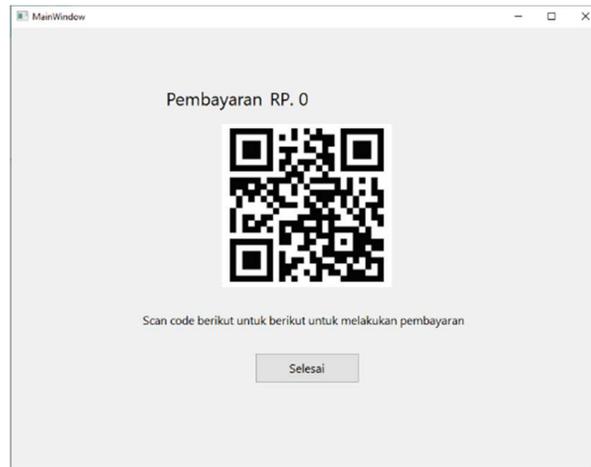
3.3 Hasil dan Pembahasan GUI

Hasil GUI yang telah berhasil dibuat pada penelitian ini ditunjukkan melalui Gambar 7 dan Gambar 8 dalam *self-checkout system*.



(a) (b)
Gambar 7. Tampilan (a) Halaman Awal (b) Halaman Hasil

Gambar 7 (a) merupakan tampilan pada halaman awal yang terdiri dari dua tombol, yaitu tombol *Ambil Gambar* dan tombol *Start*. Pada halaman ini juga terdapat peringatan agar pengguna dapat meletakkan barang belanjanya dengan baik dan tidak saling tumpang tindih, agar tidak ada kesalahan dalam melakukan *checkout*. Setelah barang belanjaan ditempatkan pada keranjang yang telah ditentukan, selanjutnya pengguna memilih tombol *Ambil Gambar* berfungsi untuk mengambil gambar barang yang telah ditempatkan di keranjang. Selanjutnya pengguna dapat memulai *checkout* dengan menekan tombol *Start*. Pada Gambar 7 (b) merupakan halaman hasil yang berguna untuk menampilkan barang belanjaan yang telah di *checkout* sebelumnya. Pada halaman ini akan ditampilkan nama barang belanjaan, jumlah, harga satuan dan total harga keseluruhan. Selanjutnya untuk memulai pembayaran pengguna harus menekan tombol *Bayar*.



Gambar 8. Tampilan Halaman Pembayaran

Dalam Gambar 8, halaman ini akan menampilkan total harga yang harus dibayarkan serta *QR Code* untuk pembayaran. Pengguna dapat menekan tombol Selesai jika transaksi telah diselesaikan dan GUI akan otomatis berpindah kehalaman awal

3.4 Hasil dan Pembahasan Pengujian

Sistem yang dikembangkan pada penelitian ini dapat dilakukan pemindaian terhadap objek apa saja tanpa spesifikasi tertentu. Pada penelitian ini, sampel objek yang digunakan adalah tiga jenis objek meliputi Indomie, Teh Botol dan Chitato. Sebelum objek tersebut dilakukan pengujian, system terlebih dahulu menginput dataset yang tertera pada produk. Pada pengujian ini, peneliti melakukan pengujian dengan teknik peletakan posisi objek dalam dua kondisi. Kondisi pertama dalam keadaan produk berdempetan dan kondisi kedua dalam keadaan produk di beri jarak. Proses pengujian dalam penelitian ini dapat dilihat melalui Gambar 9 (a) dan 9 (b) di bawah ini.



(a)
(b)
Gambar 9. Pengujian dengan Posisi (a) Berdempetan (b) Berjarak

Pada Gambar 9 merupakan proses pengujian dengan posisi produk saling berdempetan dan posisi berjarak. Posisi produk saling berdempetan menunjukkan sisi produk sedikit tumpang tindih, posisi produk teh botol berada diantara produk chitato dan indomie. Sementara, posisi produk berjarak diberi jarak $\pm 3 - 5$ cm. Pemberian jarak dilakukan agar hasil gambar tiap produk yang dihasilkan lebih maksimal, sehingga diharapkan tidak akan terjadi kesalahan deteksi oleh sistem. Rincian hasil pengujian ditunjukkan oleh Tabel 1 berikut.

Tabel 1. Hasil Pengujian dengan Posisi Berdempetan

No	Nama Barang	Deteksi					
		Posisi Berdempetan			Posisi Berjarak		
1	Indomie	1	0		1	0	
2	Chitato	1	0		1	0	
3	Teh Botol	1	0		1	0	
4	Indomie, Indomie	1	1		1	1	
5	Chitato, Chitato	1	1		1	1	
6	Teh Botol, Teh Botol	1	0		1	1	
7	Indomie, Chitato	1	1		1	1	
8	Indomie, Teh Botol	1	1		1	1	
9	Chitato, Teh Botol	1	1		1	1	
10	Indomie, Teh Botol, Chitato	1	0	1	1	1	1
11	Indomie, Indomie, Chitato	1	1	1	1	1	1
12	Indomie, Indomie, Teh Botol	1	0	1	1	1	1
13	Chitato, Chitato, Indomie	1	1	1	1	1	1
14	Chitato, Chitato, Teh Botol	1	1	0	1	1	1
15	Teh Botol, Teh Botol, Indomie	1	0	1	1	1	1
16	Teh Botol, Teh Botol, Chitato	1	0	1	1	1	1
17	Indomie, Indomie, Indomie	1	1	1	1	1	1
18	Chitato, Chitato, Chitato	1	1	1	1	1	1
19	Teh Botol, Teh Botol, Teh Botol	1	1	0	1	1	1

Hasil pengujian pada Tabel 1 diatas menunjukkan bahwa pengujian terhadap posisi produk berdempetan dengan 2 barang mendapati 1 kesalahan. Kesalahan lebih bayak terjadi pada pengujian dengan jumlah 3 barang. Rata-rata kesalahan banyak terjadi pada produk teh botol. Hal ini dikarenakan produk teh botol kemasannya lebih kecil dibandingkan chitato dan indomie, sehingga pada saat produk diletakkan secara berdempetan produk teh botol sedikit tertutup oleh produk disebelahnya. Oleh karena itu hasil pengambilan gambar menjadi kurang maksimal dan menyebabkan kesalahan deteksi oleh sistem. Hasil pengujian saat penempatan produk diberi jarak menunjukkan seluruh barang dapat dideteksi dengan benar tanpa adanya kesalahan. Sistem dapat mendeteksi dengan benar karena sisi keseluruhan produk dapat terlihat jelas oleh sistem pada saat pengambilan gambar. Sehingga dapat diartikan pemberian jarak antar produk sangat mempengaruhi keberhasilan hasil deteksi. Oleh karena itu, posisi penempatan harus diperhatikan pada saat pengujian dengan 2 dan 3 barang. Penelitian ini dilakukan terhadap tiga kategori objek dengan spesifikasi tertentu. Sistem yang diusulkan pada penelitian ini dapat beroperasi dengan menginput data objek sebelum di lakukan pendeteksian.

3.5 Hasil dan Pembahasan Terhadap Pengujian Waktu

Penelitian ini melaukan pengujian waktu untuk mengetahui efektivitas kinerja dari sistem yang diusulkan. Adapun waktu yang diperlukan oleh sistem untuk memproses dan mendeteksi barang yang di *checkout* terdapat pada Gambar 10 berikut.

1/1	[=====]	- 1s 844ms/step
1/1	[=====]	- 1s 798ms/step
1/1	[=====]	- 1s 799ms/step
1/1	[=====]	- 1s 800ms/step
1/1	[=====]	- 1s 817ms/step
1/1	[=====]	- 1s 804ms/step
1/1	[=====]	- 1s 819ms/step
1/1	[=====]	- 1s 792ms/step
1/1	[=====]	- 1s 804ms/step
1/1	[=====]	- 1s 803ms/step
1/1	[=====]	- 1s 828ms/step
1/1	[=====]	- 1s 814ms/step
1/1	[=====]	- 1s 821ms/step
1/1	[=====]	- 1s 809ms/step
1/1	[=====]	- 1s 805ms/step
1/1	[=====]	- 1s 801ms/step
1/1	[=====]	- 1s 814ms/step

Gambar 10. Waktu Proses Deteksi

Berdasarkan hasil pengujian keseluruhan, pada Gambar 10 menunjukkan bahwa waktu yang dibutuhkan oleh sistem untuk memproses data dan mendeteksi barang yang dicheckout. Rata-rata sistem membutuhkan waktu 1 detik untuk memproses dan menampilkan hasil deteksi pada halaman hasil. Tidak ada perbedaan yang signifikan untuk memproses dari jumlah 1 barang hingga 3 barang. Hasil ini sudah sangat baik karena sudah sesuai dengan yang diinginkan yaitu kurang dari 5 detik untuk waktu prosesnya. Namun waktu proses yang dibutuhkan mungkin akan berbeda-beda jika dijalankan pada perangkat komputer lain, karena spesifikasi dan sistem operasi yang digunakan mungkin saja berbeda sehingga mempengaruhi kinerja program.

4. KESIMPULAN

Penelitian ini mengusulkan rancangan dan implementasi *self-checkout system* pada toko ritel menggunakan *Convolutional Neural Network (CNN)* menggunakan metode *transfer learning* CNN dari arsitektur model MobileNetV2. Dataset yang digunakan pada penelitian ini berjumlah 247 citra dengan resolusi 224 x 224 pixels terhadap tiga jenis produk yang telah ditentukan berupa Teh Botol, Indomie dan Chitato. Berdasarkan hasil pengujian yang telah dilakukan, diperoleh bahwa tingkat akurasi yang dihasilkan mencapai persentase 88.8 % dengan total 247 data gambar dataset. Hasil evaluasi model berdasarkan Tabel *hamming loss* mendapatkan nilai *loss* sebesar 0.12 % dan nilai *hamming score* atau akurasinya sebesar 0.88 %. Berdasarkan hasil tersebut model yang dikembangkan terbukti dapat berjalan cukup ringan pada perangkat pengujian dan berjalan sudah sangat baik. Hasil pengujian menunjukkan bahwa pada sistem GUI waktu yang dibutuhkan untuk proses deteksi objek pada saat *checkout* hanya memerlukan waktu 1 detik. Hasil pengujian juga menunjukkan bahwa posisi penempatan produk dapat memengaruhi tingkat akurasi dalam pendeteksian. Hasil pendeteksian terhadap produk dengan posisi berjarak memiliki tingkat akurasi yang lebih tinggi dibandingkan produk dengan posisi berdempetan. *Self-checkout system* dengan CNN yang dikembangkan pada penelitian ini berhasil dijalankan sesuai dengan hasil yang diharapkan.

DAFTAR RUJUKAN

- Abana, E. C., Daña, T. B., Alan, C., Martin, J. M., Buraga, R., & Balagtas, C. (2019). Self-Service Checkout System for Groceries. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(4), 1815–1818. <https://doi.org/10.35940/ijrte.c6245.118419>
- Bukhari, S. T., Amin, A. W., Naveed, M. A., & Abbas, M. R. (2021). *ARC: A Vision-based Automatic Retail Checkout System*.
- Hamim, M. (2018). Penggunaan Teknologi Berbasis RFID untuk Security System. *Indonesian Journal Of Academic Librarianship*, 2(2), 13–20.
- Howard, A. G., Chen, B., & Wang, W. (2020). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. October.
- Pathak, A. R., Pandey, M., & Rautaray, S. (2018). Application of Deep Learning for Object Detection. *Procedia Computer Science*, 132(Iccids), 1706–1717. <https://doi.org/10.1016/j.procs.2018.05.144>
- Prabhu. (2018). *Understanding of Convolutional Neural Network (CNN) — Deep Learning*. Medium.
- Prawira, I. M. R., Adiwijaya, A., & Mubarak, M. S. (2018). Klasifikasi Multi-Label Pada Topik Berita Berbahasa Indonesia Menggunakan Multinomial Naive Bayes. *eProceedings of Engineering*, 5(3), 7774–7781.
- Rigner, A. (n.d.). AI-based machine vision for retail self-checkout system. *GrosseryCheckout*.
- Rochman, F., & Junaedi, H. (2020). Implementasi Transfer Learning Untuk Identifikasi Ordo Tumbuhan Melalui Daun. *Jurnal Syntax Admiration*, 1(6), 672–679.
- Sandler, M., Zhu, M., Zhmoginov, A., & Mar, C. V. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520.
- Saputra, A. R. (2019). Simulasi Sistem Point of Sale Menggunakan Radio Frequency Identification Pada Perusahaan Ritel. *JURNAL TEMATIKA*, 7(1), 11–18.
- Suartika, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). *Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101*. 5(1).
- Sulaiman, A. W., Susanto, E., & Sunarya, U. (2016). Perancangan dan implementasi sistem faktur dan pembayaran otomatis pada toko swalayan berbasis rfid. *Jurnal Penelitian dan Pengembangan Telekomunikasi, Kendali, Komputer, Elektrik, dan Elektronika (TEKTRIKA)*, 1(1), 98–102.
- Suwarno, & Lim, R. R. (2021). Perancangan Sistem Kasir Layanan Mandiri Berbasis Web

- Pada Supermarket Bless Dengan Metode Sdlc. *Conference on Management, Business, Innovation, Education and Social Sciences (CoMBInES)*, 1(1), 860–873.
- Tungadi, A. L., Lisangan, E. A., & Saputra, A. R. (2019). Simulasi Sistem Point of Sale Menggunakan Radio Frequency Identification Pada Perusahaan Ritel. *TEKNOMATIKA, Jurnal Informatika dan Komputer*, 12(1), 1–7.
- Wang, W., Li, Y., Zou, T., Wang, X., You, J., & Luo, Y. (2020). A novel image classification approach via dense-mobilenet models. *Mobile Information Systems*, 2020. <https://doi.org/10.1155/2020/7602384>
- Wiraguna, A., Faraby, S. Al, & Adiwijaya. (2019). Klasifikasi Topik Multi Label pada Hadis Bukhari dalam Terjemahan Bahasa Indonesia Menggunakan Random Forest. *e-Proceeding of Engineering*, 6(1), 2144–2153.