DETEKSI PRODUK RITEL PADA SISTEM SELF-CHECKOUT MENGGUNAKAN EFFICIENTDET

Stefi Lauren [1] Kent Dylan Sanjaya [2] Steven [3]
Christabella Jocelynne Chandra [4] Angela Chow [5] Ravelino Radissey [6]

[1) 2) 3)] Teknik Informatika Universitas Tarumanagara
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
email : [1)] stefi.535230150@stu.untar.ac.id, [2)] kent.535220063@stu.untar.ac.id,
[3)]steven.535220091@stu.untar.ac.id,
[4)] christabella.535220166@stu.untar.ac.id, [5)] angela.535220170@stu.untar.ac.id,
[6)]ravelino.535220181@stu.untar.ac.id

ABSTRAK

Sistem self-checkout menawarkan solusi inovatif untuk mengurangi antrean dan biaya operasional di sektor ritel. Dengan memanfaatkan deep learning, sistem ini memungkinkan identifikasi produk secara otomatis tanpa interaksi kasir, sehingga meningkatkan efisiensi dan kenyamanan pelanggan. Penelitian ini mengusulkan sistem self-checkout menggunakan model EfficientDet, yaitu metode deteksi objek yang dikenal memiliki keseimbangan antara akurasi dan efisiensi komputasi. Dataset yang terdiri atas 10 jenis produk makanan dan minuman dikumpulkan serta digunakan untuk pelatihan, validasi, dan pengujian. Model dilatih dengan variasi batch size dan epoch untuk menentukan konfigurasi terbaik. Hasil optimal diperoleh pada batch size 16 dan 800 epoch, dengan akurasi pelatihan sebesar 97% dan akurasi validasi sebesar 99,79%. Kinerja sistem kemudian dievaluasi melalui tiga skenario pengujian. Pada citra dengan satu produk, sistem mencapai akurasi deteksi 100% dan akurasi pengenalan 92%. Pada citra dengan tiga produk, akurasi deteksi dan pengenalan masing-masing mencapai 72%, sedangkan pada lima produk mencapai 42% dan 30%. Hasil ini menunjukkan bahwa EfficientDet layak digunakan dalam pengembangan sistem self-checkout yang efisien dan praktis, sekaligus menyoroti tantangan dalam deteksi multiobjek.

Key words

deteksi objek, EfficientDet, self-checkout, deep learning, deteksi produk

1. Pendahuluan

Perkembangan teknologi digital telah memberikan dampak yang signifikan di berbagai sektor, termasuk industri ritel. Salah satu inovasi yang semakin populer adalah *self-checkout*, yaitu sistem yang memungkinkan pelanggan memindai dan membayar produk tanpa interaksi dengan kasir. Sistem ini awalnya dikembangkan untuk mengatasi berbagai tantangan dalam sektor ritel, seperti antrean panjang di kasir, tingginya biaya operasional, serta kebutuhan akan layanan yang cepat dan efisien. Dalam praktiknya, *self-checkout* dirancang untuk memberikan pengalaman berbelanja yang lebih mandiri bagi pelanggan. Setelah memilih produk, pelanggan dapat memindai barang dan melakukan pembayaran secara langsung tanpa bantuan kasir. Inovasi ini mengurangi ketergantungan terhadap tenaga kerja manusia, sehingga berpotensi menurunkan biaya operasional toko. Selain itu, *self-checkout* dapat menghemat waktu berbelanja karena proses pembayaran menjadi lebih cepat dan efisien.

Secara konvensional, *self-checkout* masih bergantung pada pemindaian kode batang (barcode) atau kode QR untuk mengidentifikasi produk [1]. Namun, metode ini memiliki keterbatasan, seperti proses pembayaran yang lebih lambat akibat pemindaian dan risiko kesalahan ketika kode rusak atau tertutup. Untuk mengatasi keterbatasan tersebut, diperlukan penerapan teknologi *computer vision* dan *deep learning* dalam mendeteksi serta mengenali objek secara otomatis, sehingga pengalaman *self-checkout* menjadi lebih efisien dan praktis. Oleh karena itu, pengembangan teknologi deteksi objek sangat diperlukan untuk meningkatkan pengalaman konsumen dalam sistem *self-checkout*. Salah satu pendekatan yang menjanjikan adalah metode *EfficientDet*, yaitu algoritma deteksi objek yang dikenal karena kemampuan pelatihannya yang cepat serta akurasi tinggi dalam waktu pelatihan yang relatif singkat. EfficientDet telah terbukti memiliki efisiensi komputasi yang lebih baik dibandingkan *YOLO* dan *AmoebaNet*, menjadikannya salah satu algoritma deteksi objek yang paling akurat dan efisien [2]. Berdasarkan keunggulan tersebut, beberapa penelitian sebelumnya telah mengeksplorasi penerapan EfficientDet untuk berbagai tujuan deteksi objek. Misalnya, EfficientDet telah digunakan untuk mengembangkan sistem penentuan

kelayakan penerima bantuan tunai langsung berdasarkan citra rumah [2], serta deteksi patung dengan mengombinasikan *Gaussian Smoothing Filter* dan EfficientDet untuk meningkatkan akurasi deteksi [3]. Selain itu, ada juga penelitian yang berfokus pada peningkatan metode deteksi objek menggunakan pendekatan *Findcontour* yang dikombinasikan dengan logika fuzzy untuk mendeteksi aksara Bali pada daun lontar [4].

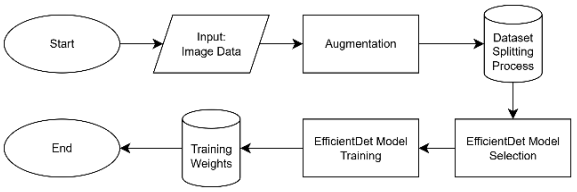
Meskipun penelitian-penelitian tersebut menunjukkan hasil yang menjanjikan, sebagian besar belum berfokus pada penerapan EfficientDet dalam sistem self-checkout untuk mendeteksi produk ritel. Penelitian ini merancang sistem *self-checkout* berbasis EfficientDet yang terdiri atas beberapa modul utama. Pertama, modul Home menyediakan antarmuka utama dengan tombol unggah foto yang digunakan untuk memilih citra input berupa gambar produk. Setelah citra input dimasukkan, sistem akan menggunakan model EfficientDet yang telah dibuat untuk mendeteksi produk. Modul Help berisi panduan penggunaan langkah demi langkah agar pengguna dapat memahami cara mendeteksi produk dengan benar, serta informasi mengenai tombol-tombol lain yang ada. Modul About menampilkan informasi mengenai aplikasi dan pengembangnya, sedangkan modul Result menampilkan hasil deteksi berupa daftar produk dan jumlahnya.

Sistem ini menerima masukan berupa citra yang berisi produk, kemudian melakukan proses deteksi dan pengenalan menggunakan metode EfficientDet. Dataset yang digunakan dalam penelitian ini dikumpulkan secara langsung dari 10 jenis produk makanan dan minuman, yaitu Aqua, Teh Kotak, Good Day, Kopi Kenangan, Fanta, Coca-Cola, Sprite, Chitato, Taro, dan Qtela. Sistem self-checkout ini memungkinkan pengguna mengambil gambar produk, kemudian sistem akan menampilkan daftar produk melalui antarmuka yang telah disediakan. Penelitian ini menyajikan prototipe *self-checkout* berbasis *computer vision* serta evaluasi terhadap kinerjanya, yang dapat menjadi acuan bagi pengembangan sistem yang lebih efisien dan praktis di masa mendatang.

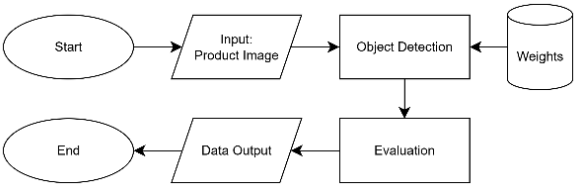
2. Metode Penelitian

2.1. Pelatihan dan Implementasi Sistem

Proses pengembangan sistem ini terdiri dari dua tahap utama, yaitu pelatihan model dan implementasi sistem. Pada tahap pelatihan, dilakukan augmentasi terhadap citra produk, yang kemudian dibagi menjadi subset pelatihan, validasi, serta pengujian. Selanjutnya, model EfficientDet dipilih dan dilatih untuk menghasilkan bobot deteksi yang teroptimasi, seperti yang ditunjukkan pada Gambar 1. Pada tahap implementasi, bobot tersebut diintegrasikan ke dalam sistem untuk memungkinkan deteksi secara *real-time*. Pertama, citra produk dimasukkan dan diproses oleh model EfficientDet untuk melakukan deteksi dan klasifikasi objek, kemudian hasilnya dievaluasi sebelum ditampilkan kepada pengguna. Diagram proses ini dapat dilihat pada Gambar 2. Pengembangan dua tahap ini memastikan kinerja deteksi yang baik serta penerapan yang praktis dalam prototipe sistem *self-checkout* untuk industri ritel.



Gambar 1. Diagram alir pelatihan model



Gambar 2. Diagram alir pengujian model

2.2. Dataset

Dataset yang digunakan dalam penelitian ini dikumpulkan melalui pengambilan citra langsung terhadap produk-produk ritel yang sudah disebutkan. Setiap produk difoto dari berbagai sudut, mulai dari 0° hingga 360°, serta di bawah kondisi pencahayaan yang bervariasi untuk memastikan ketahanan sistem terhadap perubahan orientasi dan pencahayaan. Secara keseluruhan, sebanyak 2026 citra disiapkan untuk pelatihan, dengan minimal 200 citra per jenis produk, serta 500 citra untuk validasi, dengan 50 citra per jenis produk. Dataset ini terdiri atas 10 jenis produk makanan dan minuman yang umum tersedia di toko ritel, sebagaimana telah disebutkan pada bagian pendahuluan. Representasi jenis-jenis produk yang digunakan dalam proses deteksi dapat dilihat pada Gambar 3.

Untuk evaluasi model, disusun dataset pengujian independen yang terdiri dari 150 citra. Citra-citra ini dibagi ke dalam tiga skenario:

1. 50 citra yang berisi satu produk per *frame*
2. 50 citra yang berisi tiga produk per *frame*
3. 50 citra yang berisi lima produk per *frame*

Pemisahan ini memastikan bahwa proses pelatihan dan validasi dilakukan pada data yang berbeda dari tahap pengujian, sehingga evaluasi terhadap kinerja deteksi dan pengenalan dapat dilakukan dengan lebih baik. Contoh dari ketiga skenario pengujian tersebut ditunjukkan pada Gambar 4, yang memperlihatkan *frame* berisi satu produk, tiga produk, dan lima produk.

Gambar 3. Contoh citra dari sepuluh kategori produk dalam dataset

Gambar 4. Contoh citra skenario dataset pengujian: satu, tiga, dan lima produk per *frame*.

2.3. Model EfficientDet

EfficientDet merupakan algoritma deteksi objek yang dikembangkan oleh Google Research, yang menyeimbangkan antara akurasi dan efisiensi komputasi [2], sehingga cocok digunakan untuk aplikasi *real-time* dan penerapan pada perangkat dengan kemampuan komputasi terbatas. Representasi sederhana dari arsitektur EfficientDet ditunjukkan pada Gambar 5, dengan fokus pada komponen BiFPN dan *prediction heads*. Arsitektur ini terdiri atas tiga komponen utama, yaitu *backbone*, *neck*, dan *head*.

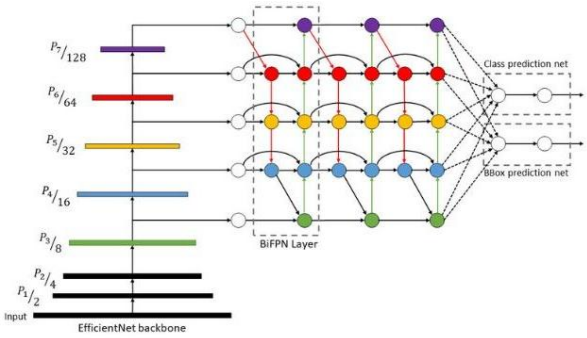
Komponen *EfficientBackbone* berfungsi untuk mengekstraksi fitur dari citra masukan. Komponen ini didasarkan pada arsitektur EfficientNet, yang menerapkan *compound sccaling* untuk menyeimbangkan kedalaman, lebar, dan resolusi jaringan, seperti yang ditunjukkan pada Gambar 6. Pendekatan *scaling* tersebut dirumuskan sebagai berikut:

$$d = \alpha^{\phi} \quad (1)$$
$$w = \beta^{\phi} \quad (2)$$
$$r = \gamma^{\phi} \quad (3)$$

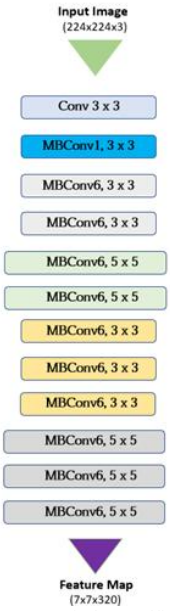
Di mana parameter α, β, dan γ merupakan *hyperparameter* yang menentukan skala untuk kedalaman, lebar, dan resolusi jaringan secara berturut-turut, sedangkan ϕ merupakan koefisien yang mengatur skala model (misalnya EfficientNet-B0, B1, dan seterusnya). Mekanisme ini memungkinkan EfficientNet memproses citra secara efisien dan menghasilkan *feature map* multi-skala. Pada *EfficientBackbone*, tingkat resolusi diatur dari yang terendah (P1) hingga tertinggi (P7), yang menyediakan representasi hierarkis yang penting untuk mendeteksi objek dengan berbagai ukuran [2].

Bagian *neck* diimplementasikan menggunakan *Bidirectional Feature Pyramid Network* (BiFPN), yang merupakan pengembangan dari *Feature Pyramid Network* (FPN) tradisional, seperti yang ditunjukkan pada Gambar 7. BiFPN memperkuat penggabungan fitur multi-skala dengan mengombinasikan feature map dalam dua arah: *top-down*, di mana fitur tingkat tinggi diturunkan dan digabungkan dengan fitur tingkat rendah, serta *bottom-up*, di mana fitur detail tingkat rendah ditingkatkan dan diintegrasikan ke dalam *feature map* tingkat tinggi. Setiap *fusion node* pada BiFPN secara adaptif memberikan bobot pada fitur masukan, sehingga jaringan dapat mempelajari fitur mana yang paling berkontribusi terhadap deteksi yang akurat. Proses penggabungan iteratif ini memungkinkan representasi objek yang lebih kuat pada berbagai ukuran.

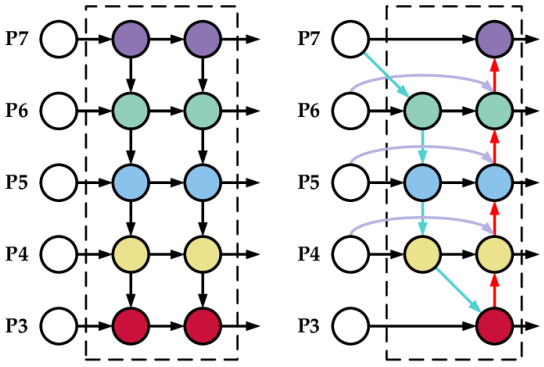
Komponen *EfficientHead* merupakan tahap akhir dari EfficientDet yang terdiri dari lapisan konvolusional dan lapisan *output* yang menghasilkan koordinat *bounding box* serta prediksi kelas objek. Bagian *backbone* menggunakan menggunakan EfficientNet untuk penskalaan citra, sedangkan BiFPN pada bagian *neck* menggunakan fitur multi-skala dari level P3 hingga P7 untuk melakukan prediksi objek. EfficientDet diimplementasikan dalam platform *TensorFlow*. Model ini mendukung berbagai model seperti *CenterNet*, *MobileNet*, *ResNet*, dan *Faster R-CNN*. Keunggulan utama EfficientDet terletak pada kemampuannya mencapai kinerja yang lebih tinggi dibandingkan model-model tersebut, dengan teteap mempertahankan efisiensi komputasi. Selain itu, EfficientDet memiliki ukuran model yang ringan, sehingga ideal untuk diterapkan pada perangkat dengan sumber daya terbatas seperti perangkat *mobile* [3].



Gambar 5. Struktur BiFPN dan *Efficienthead* pada EfficientDet.



Gambar 6. Arsitektur EfficientNet



Gambar 7. Perbandingan FPN (kiri) dan BiFPN (kanan) pada EfficientDet.

2.4. Evaluasi

Untuk menilai kinerja model yang diusulkan, digunakan *Mean Average Precision* (mAP) dan metrik berbasis *confusion matrix*. Metrik mAP merupakan metrik evaluasi standar untuk deteksi objek, yang dihitung sebagai rata-rata dari *Average Precision* (AP) untuk seluruh kelas objek, seperti yang ditunjukkan pada Persamaan (4) [5].

$$mAP = \frac{1}{N}\sum_{i=1}^{N} \quad AP(i) \times 100\% \qquad (4)$$

Dengan N merupakan jumlah total kelas objek dalam dataset, dan $AP(i)$ menunjukkan nilai AP untuk kelas objek ke-i.

Kinerja klasifikasi juga dievaluasi menggunakan *confusion matrix*, yang terdiri dari *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) [6], seperti yang ditampilkan pada Tabel 1. *Confusion matrix* merupakan metode perhitungan yang membandingkan hasil klasifikasi dengan data sebenarnya [7]. Matriks ini menunjukkan tingkat akurasi dalam bentuk persentase dan berfungsi sebagai acuan yang berguna untuk menilai kinerja suatu algoritma klasifikasi.

Tabel 1. *Confusion Matrix*

Kelas	Prediksi Positif	Prediksi Negatif
Aktual Positif	*True Positive* (TP)	*False Negative* (FN)
Aktual Negatif	*False Positive* (FP)	*True Negative* (TN)

Berdasarkan nilai-nilai tersebut, diperoleh ukuran evaluasi umum, yaitu akurasi (*accuracy),*yang mengukur tingkat ketepatan keseluruhan dari sistem klasifikasi, sebagaimana ditunjukkan pada Persamaan (5).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (5)$$

2.5. Perancangan Sistem dan Metode Pengujian

Sistem *self-checkout* yang dikembangkan mengintegrasikan model EfficientDet ke dalam antarmuka berbasis web. Perancangan sistem difokuskan pada integrasi antara proses deteksi objek dan tampilan antarmuka pengguna agar hasil deteksi dapat ditampilkan secara real-time dan mudah dipahami. Untuk menguji fungsionalitas dan kinerja sistem, digunakan metode *black-box testing*. Pengujian ini menilai keluaran sistem berdasarkan data masukan tanpa memeriksa kode sumber secara internal. Pendekatan ini dipilih karena efektif untuk mengevaluasi keakuratan fungsionalitas, keandalan hasil deteksi, serta konsistensi perilaku sistem dari perspektif pengguna akhir. Pengujian dilakukan pada tingkat modul maupun sistem secara keseluruhan untuk memastikan seluruh komponen berjalan sesuai rancangan.

3. Hasil dan Pembahasan

Implementasi sistem *self-checkout* dilakukan menggunakan laptop ASUS A409JF yang dilengkapi prosesor Intel Core i7 generasi ke-8 dan RAM sebesar 8 GB. Lingkungan pengembangan yang digunakan meliputi sistem operasi Windows 10, bahasa pemrograman Python sebagai bahasa utama, *Visual Studio Code* sebagai *code editor*, *Google Colab* untuk pelatihan model, serta *Amazon Web Services* (AWS) sebagai dukungan untuk proses *deployment*.

Pelatihan model dilakukan di *Google Colab* dengan memanfaatkan akselerasi GPU untuk meningkatkan kecepatan komputasi. Selain itu, AWS digunakan untuk mendukung proses *deployment* dan menyediakan akses terhadap sumber daya berbasis *cloud*, sehingga sistem dapat diuji dan diimplementasikan secara fleksibel di luar lingkungan lokal.

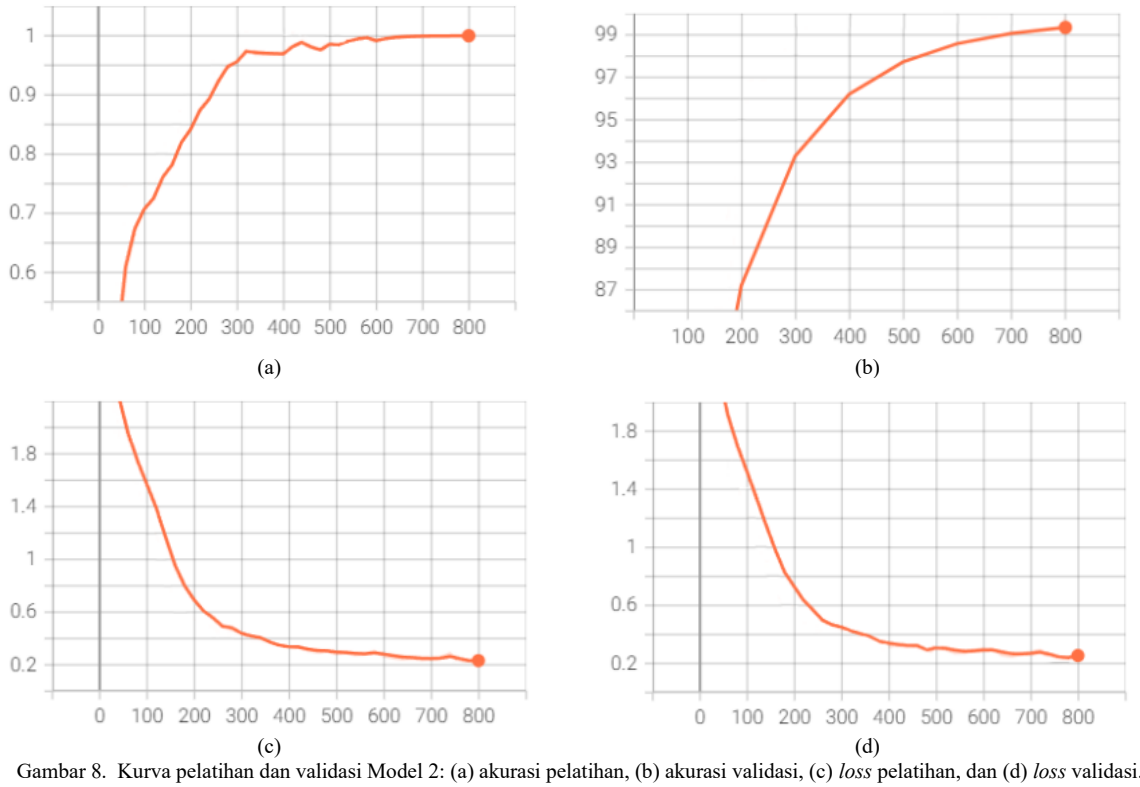
Evaluasi dilakukan terhadap model EfficientDet yang telah dilatih menggunakan dataset yang dikumpulkan. Empat model diuji dengan memvariasikan ukuran *batch* (*batch size*), sementara jumlah *epoch* dipertahankan tetap sebanyak 800, seperti dirangkum

pada Tabel 2. Di antara model yang diuji, Model 2 menunjukkan kinerja terbaik dengan akurasi pelatihan sebesar 97% dan akurasi validasi sebesar 99,76%. Meskipun nilai validation loss-nya sedikit lebih tinggi dibandingkan Model 1 dan Model 3, kombinasi antara akurasi validasi yang tinggi dan kinerja pelatihan yang stabil menjadikan Model 2 kandidat paling baik untuk pengujian lebih lanjut pada dataset uji. Untuk memberikan gambaran lebih lanjut, kurva pelatihan dan validasi Model 2 ditunjukkan pada Gambar 6, yang terdiri dari empat subplot (a) akurasi pelatihan, (b) akurasi validasi, (c) *loss* pelatihan, dan (d) *loss* validasi. Plot akurasi menunjukkan bahwa model mengalami konvergensi secara stabil dan mempertahankan kinerja tinggi sepanjang *epoch* pelatihan. Sementara itu, plot *loss* menunjukkan penurunan yang konsisten baik pada data pelatihan maupun validasi tanpa adanya perebedaan signifikan di antara keduanya. Hasil ini mengindikasikan bahwa model mampu melakukan generalisasi dengan baik tanpa mengalami *overfitting*, sehingga mendukung pemilihannya sebagai kandidat akhir untuk evaluasi pada data yang belum pernah digunakan sebelumnya.

Hasil tersebut menunjukkan bahwa EfficientDet mampu mencapai tingkat akurasi yang tinggi baik pada data pelatihan maupun validasi, sehingga membuktikan kelayakannya untuk digunakan pada tugas deteksi objek dalam sistem *self-checkout* yang diusulkan. Selain itu, konsistensi akurasi pada berbagai ukuran batch menunjukkan ketahanaan model, sementara variasi kecil pada nilai *loss* mengindikasikan bahwa penyesuaian *hyperparameter* lebih lanjut masih dapat dilakukan untuk memperoleh hasil yang lebih optimal.

Tabel 2. Hasil Pelatihan dan Validasi Berbagai Model EfficientDet

No.	Model	*Batch Size*	*Epoch*	Akurasi Pelatihan	*Loss Pelatihan*	Akurasi Validasi	*Loss* Validasi
1	Model 1	8	800	96%	0,181	99,61%	0,163
2	Model 2	16	800	97%	0,027	99,76%	0,231
3	Model 3	32	800	97%	0,171	99,67%	0,181
4	Model 4	64	800	96%	0,196	99,64%	0,242



Gambar 8. Kurva pelatihan dan validasi Model 2: (a) akurasi pelatihan, (b) akurasi validasi, (c) *loss* pelatihan, dan (d) *loss* validasi.

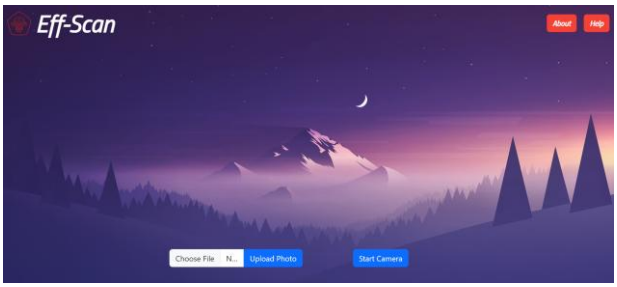
Model yang telah dilatih diuji pada tiga skenario pengujian, dan hasilnya dirangkum pada Tabel 3. Sistem menunjukkan kinerja terbaik pada skenario dengan satu produk, dengan akurasi deteksi sebesar 100% dan akurasi pengenalan sebesar 92%. Pada skenario tiga produk, akurasi deteksi dan pengenalan menurun hingga 72%. Kinerja terendah diperoleh pada skenario lima produk, di mana akurasi deteksi hanya mencapai 42% dan akurasi pengenalan turun hingga 30%. Berdasarkan pengamatan lebih lanjut, penurunan kinerja ini sebagian besar disebabkan oleh kesalahan klasifikasi pada beberapa produk, seperti Nissin Crackers dan Roma Kelapa. Kemiripan visual pada kemasan produk- produk tersebut dengan produk lain dalam satu *frame* meningkatkan kemungkinan terjadinya kesalahan pengenalan dan deteksi palsu.

Tabel 3. Hasil Pengujian Model EfficientDet

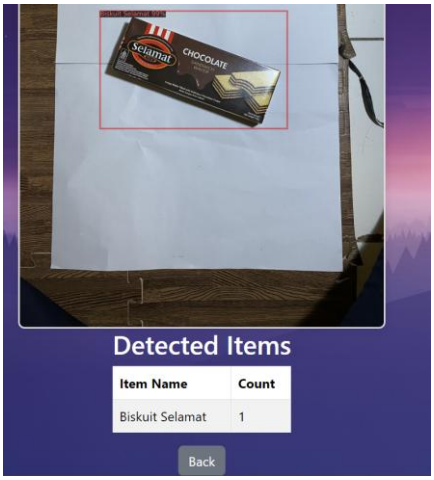
No.	Skenario Pengujian	Jumlah Objek Terdeteksi	Akurasi Deteksi	Jumlah Objek Dikenali	Akurasi Pengenalan
1	Satu produk per *frame*	50	100%	46	92%
2	Tiga produk per *frame*	36	72%	36	72%
3	Lima produk per *frame*	21	42%	15	30%

Penelitian ini memiliki beberapa keterbatasan yang dapat menjadi peluang untuk pengembangan di masa mendatang. Sistem menunjukkan penurunan akurasi ketika menangani *frame* dengan jumlah produk yang lebih banyak, serta sering salah mengklasifikasikan produk dngan kemiripan visual yang tinggi. Selain itu, evaluasi dilakukan menggunakan dataset uji yang relatif kecil, yaitu sebanyak 150 citra, sehingga belum sepenuhnya merepresentasikan keragaman kondisi ritel di dunia nyata. Penelitian lanjutan dapat mengatasi keterbatasan tersebut dengan memperluas dataset menggunakan sampel yang lebih beragam, serta menguji arsitektur deteksi objek lainnya seperti YOLO, SSD, atau Faster R-CNN. Pendekatan lanjutan seperti peningkatan fitur (*feature enhancement*) atau penerapan mekanisme perhatian (*attention mechanism*) juga dapat dieksplorasi untuk meningkatkan kemampuan model dalam membedakan produk dengan kemiripan visual yang tinggi. Selain itu, penerapan sistem ini pada prototipe *self-checkout* di lingkungan nyata akan memberikan wawasan berharga terkait kinerja dan tingkat kegunaannya secara praktis.

Prototipe *self-checkout* yang dikembangkan berhasil diimplementasikan dalam antarmuka berbasis web yang terdiri atas empat modul. Setiap modul telah diuji dari segi fungsionalitas dan kemudahan penggunaannya. Gambar 9 menampilkan modul Home, yang berfungsi dengan baik dalam menampilkan empat tombol navigasi, termasuk fitur unggah untuk citra produk. Gambar 10 menunjukkan modul Result, yang menampilkan daftar produk terdeteksi beserta jumlahnya. Selain itu, modul Help menyediakan panduan penggunaan yang jelas, sedangkan modul About memuat informasi mengenai program serta pengembangnya. Secara keseluruhan, hasil implementasi menunjukkan bahwa seluruh modul yang dirancang berfungsi sesuai dengan tujuan dan mampu memberikan antarmuka yang ramah pengguna.



Gambar 9. Antarmuka modul Home



Gambar 10. Antarmuka modul Result

4. Kesimpulan

Berdasarkan hasil dan pembahasan, dapat disimpulkan bahwa sistem *self-checkout* berbasis EfficientDet berfungsi secara efektif sesuai dengan spesifikasi perancangan, dengan seluruh modul berjalan dengan baik sebagaimana telah diverifikasi melalui pengujian. Sistem menunjukkan tingkat akurasi yang tinggi dalam mendeteksi dan mengenali produk tunggal pada setiap bingkai, yaitu dengan akurasi deteksi sebesar 100% dan akurasi pengenalan sebesar 92%. Namun, kinerja sistem menurun seiring dengan meningkatnya jumlah produk dalam satu *frame*, di mana pada skenario tiga produk diperoleh akurasi deteksi dan pengenalan sebesar 72%, sedangkan pada skenario lima produk akurasi deteksi turun menjadi 42% dan pengenalan menjadi 30%. Penurunan ini terutama disebabkan oleh kesalahan klasifikasi pada produk dengan kemiripan warna atau fitur visual yang serupa.

Dari empat model EfficientDet yang dilatih, Model 2 dengan *batch size* sebesar 16 dan 800 *epoch* menunjukkan keseimbangan terbaik antara akurasi pelatihan (97%) dan akurasi validasi (99,76%), dengan konvergensi yang stabil serta kemampuan generalisasi yang baik. Hasil tersebut mengonfirmasi bahwa EfficientDet layak digunakan untuk tugas deteksi dan pengenalan objek pada sistem *self-checkout*, serta mampu mengenali produk secara akurat dalam tingkat kompleksitas yang bervariasi. Selain itu, hasil pengujian seluruh modul menggunakan metode *black-box testing* menunjukkan kinerja yang baik, di mana semua fungsi dan fitur dalam program telah berjalan sesuai dengan spesifikasi perancangan. Secara keseluruhan, penelitian ini menegaskan potensi EfficientDet sebagai dasar yang

kuat untuk pengembangan aplikasi ritel otomatis di masa mendatang.

REFERENSI

[1] F. Falah, L. Halim and N. Saputro, "Perancangan Awal Sistem Automatic Self-Checkout Untuk Produk Buah Berbasis CNN dan Sensor Berat Loadcell," *Jurnal Teknik,* vol. 21, no. 1, pp. 1-6, 2023.

[2] M. F. R. Akbari, B. Rahayudi and L. Muflikhah, "Implementasi Deep Learning menggunakan Algoritma EfficientDet untuk Sistem Deteksi Kelayakan Penerima Bantuan Langsung Tunai berdasarkan Citra Rumah di Wilayah Kabupaten Kediri," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 7, no. 4, pp. 1817-1825, 2023.

[3] M. A. Saleh, Z. S. Ameen, C. Altrjman and F. Al-Turjman, "Computer-Vision-Based Statue Detection with Gaussian Smoothing Filter and EfficientDet," *Sustainability,* vol. 14, no. 18, pp. 1-10, 2022.

[4] I. P. Manuaba and K. A. T. Indah, "Perbaikan Deteksi Objek Metode Findcontour Menggunakan Logika Fuzzy untuk Mendeteksi Objek Aksara Bali pada Daun Lontar," *Technomedia Journal,* vol. 7, no. 3, pp. 314-322, 2023.

[5] H. R. Yudistira and Lina, "Human Activity Recognition dari Rekaman Video Pengawas dengan Metode YOLO," *Jurnal Ilmu Komputer dan Sistem Informasi,* pp. 1-5, 2022.

[6] R. L. Hasanah, M. Hasan, W. E. Pangesti and F. F. Wati, "Klasifikasi Penerima Dana Bantuan Desa Menggunakan Metode KNN (K-Nearest Neighbor)," *Jurnal Techno Nusa Mandiri,* vol. 16, no. 1, pp. 1-6, 2019.

[7] Ainurrohmah, "Akurasi Algoritma Klasifikasi pada Software Rapidminer dan Weka," *PRISMA, Prosiding Seminar Nasional Matematika,* vol. 4, pp. 493-499, 2021.

[8] W. E. Nurjanah, R. S. Perdana and M. A. Fauzi, "Analisis Sentimen Terhadap Tayangan Televisi Berdasarkan Opini Masyarakat pada Media Sosial Twitter menggunakan Metode K-Nearest Neighbor dan Pembobotan Jumlah Retweet," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 1, no. 12, pp. 1750-1757, 2017.

Stefi Lauren, sedang menempuh pendidikan pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta.

Kent Dylan Sanjaya, sedang menempuh pendidikan pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta.

Steven, sedang menempuh pendidikan pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta.

Christabella Jocelynne Chandra, sedang menempuh pendidikan pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta.

Angela Chow, sedang menempuh pendidikan pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta.

Ravelino Radissey, sedang menempuh pendidikan pada Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, Jakarta