PENGENALAN OBJEK MENGGUNAKAN METODE SINGLE SHOT MULTIBOX DETECTOR PADA BAHAN SEMBAKO

Henry Tanujaya [1)] Lina [2)]

[1) 2)] Teknik Informatika, FTI, Universitas Tarumanaraga
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
Email: henry.535180120@stu.untar.ac.id [1)], lina@fti.untar.ac.id [2)]

ABSTRAK

Bahan sembako adalah singkatan dari sembilan bahan pokok yang artinya diperlukan oleh masyarakat secara umum sebagai kebutuhan sehari – hari. Bahan sembako sangat beragam jenisnya seperti minyak, beras, susu, dan masih banyak lagi. Bahan sembako biasanya dapat ditemui di supermarket, toko eceran, maupun warung kecil. Supermarket, toko eceran, dan warung kecil menjadi penyedia banyak barang dan salah satunya bahan sembako untuk dibeli oleh masyarakat umum. Penyedia yang sangat memiliki banyak kebutuhan jenis bahan sembako biasanya terdapat di supermarket. Untuk supermarket dan toko eceran biasanya memiliki data stok barang masing – masing agar mengetahui jumlah barang mereka di rak penjualan. Pengecekan stok barang juga dilakukan untuk mengetahui tanggal kedaluwarsa, kualitas barang, dan lainnya. Metode Single Shot Multibox Detector sudah banyak digunakan untuk pengenalan objek atau pengenalan objek seperti aplikasi pengenalan benda, makhluk hidup, makanan, bahkan pengenalan wajah sekalipun. Kelebihan metode ini adalah kecepatan dan keamanan yang tidak kalah bagus dengan metode lain seperti YOLO dan Fast R-CNN. Jika dibandingkan, metode SSD dapat jauh lebih tinggi keakuratannya dan kecepatan proses pengenalan objek.

Kata kunci—

Pengenalan Objek, Supermarket, Bahan Sembako, SSD

1. Pendahuluan

Bahan sembako adalah singkatan dari sembilan bahan pokok yang artinya diperlukan oleh masyarakat secara umum sebagai kebutuhan sehari – hari. Bahan sembako sangat beragam jenisnya seperti minyak, beras, susu, dan masih banyak lagi. Bahan sembako biasanya dapat ditemui di supermarket, toko eceran, maupun warung kecil.

Supermarket, toko eceran, dan warung kecil menjadi penyedia banyak barang dan salah satunya bahan sembako untuk dibeli oleh masyarakat umum. Penyedia yang sangat memiliki banyak kebutuhan jenis bahan sembako biasanya terdapat di supermarket. Untuk supermarket dan toko eceran biasanya memiliki data stok barang masing – masing agar mengetahui jumlah barang mereka di rak penjualan. Pengecekan stok barang juga dilakukan untuk mengetahui tanggal kedaluwarsa, kualitas barang, dan lainnya.

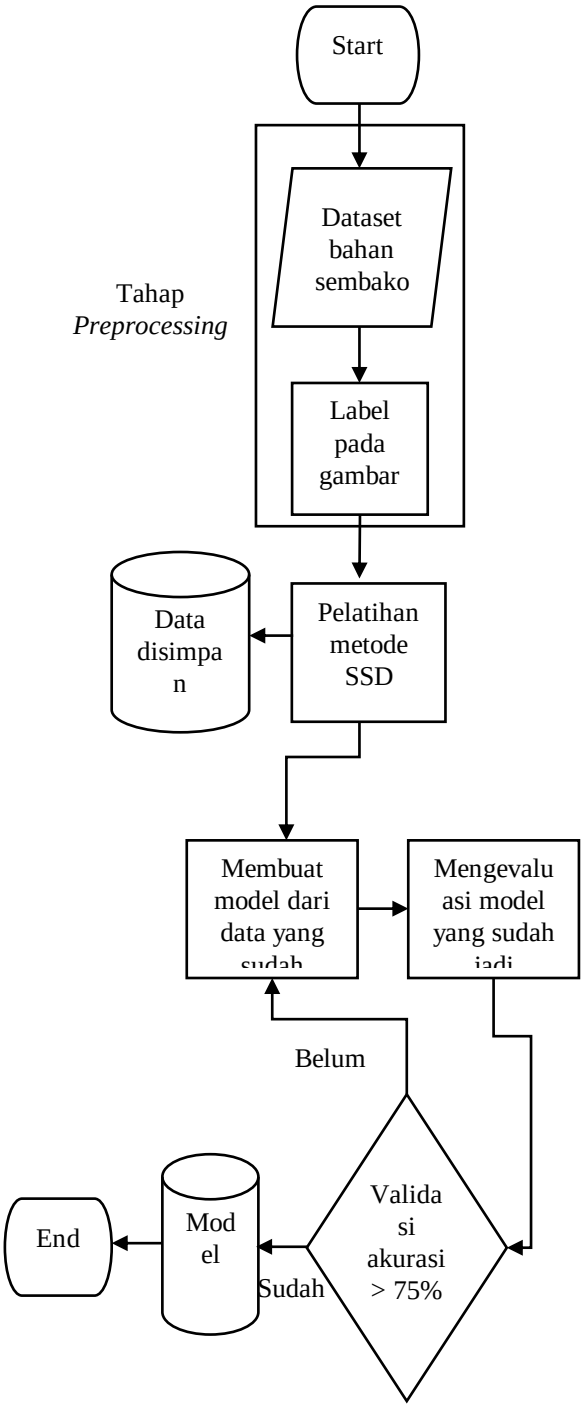
Pengenalan bahan sembako dapat membedakan jenis dari bentuk objeknya dan lebih mudah untuk melakukan pengelompokan berdasarkan jenisnya. Sistem ini pun dapat membantu seperti supermarket atau toko eceran untuk melakukan pendataan bahan sembako yang mereka jual. Saat ini pendataan stok terutama menghitung barang pada supermarket, toko eceran, atau warung kecil masih secara manual, sehingga jika barang yang sudah terlalu banyak terutama bahan sembako akan menjadi susah jika harus di cek secara manual oleh manusia. Maka dari itu sistem ini dapat membantu perhitungan stok barang dan mengenali jenis bahan sembako untuk pihak supermarket, toko eceran, maupun warung kecil.

Sistem yang akan dibangun menggunakan metode Single Shot Multibox Detector untuk mengenali berbagai objek kemasan dari bahan sembako dan membedakannya berdasarkan bentuk. Metode ini digunakan karena akurasinya tinggi dan masih menjadi rekomendasi untuk pengenalan objek. Selain itu sistem ini dapat menghitung barang otomatis dan menampilkannya saat mengenali jenis bahan sembako.

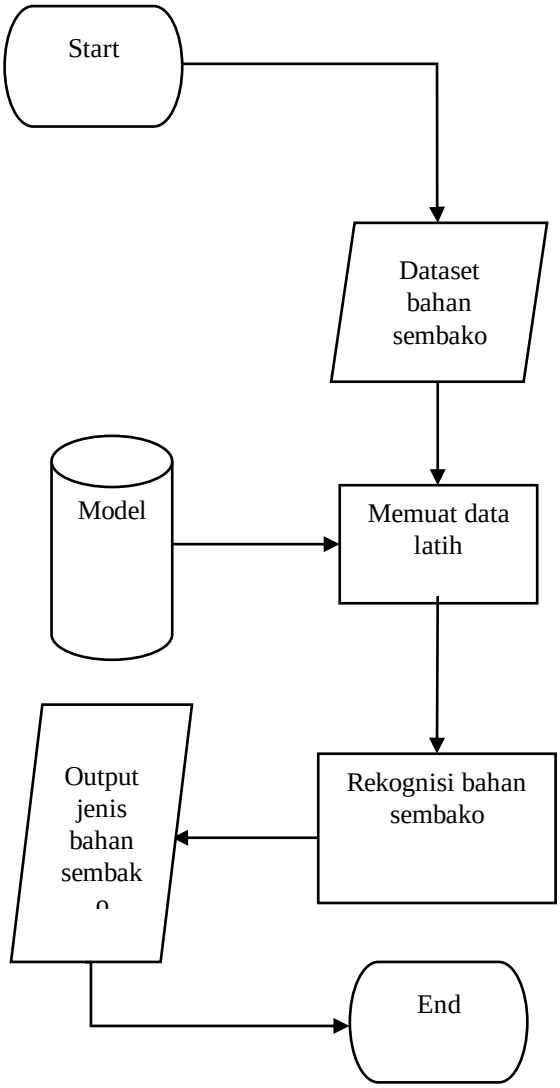
Metode Single Shot Multibox Detector sudah banyak digunakan untuk pengenalan objek atau pengenalan objek seperti aplikasi pengenalan benda, makhluk hidup, makanan, bahkan pengenalan wajah sekalipun. Kelebihan metode ini adalah kecepatan dan keamanan yang tidak kalah bagus dengan metode lain seperti YOLO dan Fast R-CNN. Jika dibandingkan, metode SSD dapat jauh lebih tinggi keakuratannya dan kecepatan proses pengenalan objek.

2. Metode Penelitian

Sistem yang dirancang merupakan program untuk toko eceran atau supermarket agar dapat membedakan jenis bahan sembako. Program ini selain dapat mengenali jenis bahan sembako, dapat menghitung otomatis bahan sembako setelah proses pengenalan jenis bahan sembako. Alur training dapat dilihat pada **Gambar 1** dan alur testing dapat dilihat pada **Gambar 2**.



Gambar 1. Flowchart Training



Gambar 2 Flowchart Testing

Pada gambar diatas untuk data training dimulai ketika tahap preprocessing sudah dilakukan. Tahap preprocessing adalah tahap pengumpulan dataset seperti mencari gambar di internet, lalu selanjutnya gambar yang sudah dikumpulkan akan diberi label agar menjadi dataset yang dapat dilatih. Setelah tahap preprocessing maka tahap selanjutnya adalah tahap data training yaitu menyimpan dataset dan melatih algoritma yang dipakai. Setelah data disimpan maka proses selanjutnya adalah data testing yaitu dari menginput dataset bahan sembako lalu memuat data yang sudah latih sebelumnya. Selanjutnya akan direkognisi lalu hasil dari rekognisi berupa jenis bahan sembako tersebut.

2.1. Pengumpulan Data

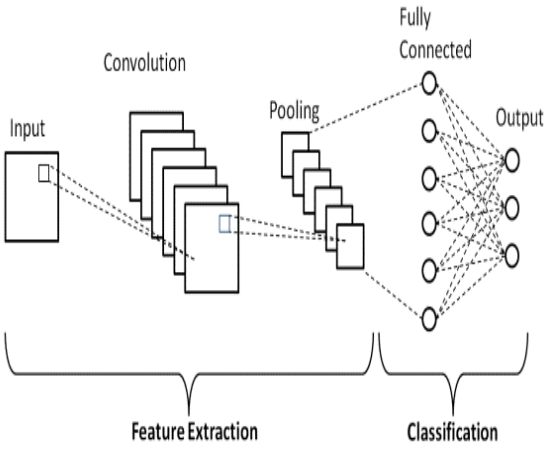
Dataset berupa 15 produk yaitu ultramilk, frisianflag, cimory, abccup, popmie, sedapcup, pristine,

aqua, leminerale, cocacola, milo, nescafe, indomie, sedap, dan sarimi. *Dataset* juga dikumpulkan dari video rak pribadi dan rak supermarket yang akan dipecah per frame agar menjadi gambar. Kelas yang digunakan ada 15 dengan masing – masing memiliki 100 gambar data train. Pengambilan dataset dilakukan dari depan dan lurus terhadap kamera. Untuk pencahayaan pada data gambar produk diambil dengan jelas agar bentuk dan warna produk juga dapat terlihat. Data yang digunakan untuk *training* sebanyak 2930, lalu untuk data *testing* sebanyak 2504 yang dimana masing – masing sudah termasuk data gambar dan data gambar yang sudah diberi label.

2.2. Convolutional Neural-Network (CNN)

Convolutional Neural Network adalah bagian dari deep neural network dalam bidang deep learning, yakni artinya jenis jaringan saraf tiruan digunakan dalam berbagai tugas seperti gambar, audio, kata-kata. Tetapi secara umum jaringan saraf tiruan biasa digunakan untuk pengenalan dan pemrosesan gambar. Algoritma ini dirancang untuk memproses data piksel dan citra visual [1].

CNN memiliki arsitektur yang mencakup tiga lapisan, yaitu *convolution layers*, *pooling layers*, dan *fully connected layers*. Berikut gambaran arsitektur dari CNN pada **Gambar 3**.



Gambar 3. Arsitektur CNN

1. *Convolution layers*

Convolution layers atau lapisan konvolusi adalah komponen dasar dari arsitektur CNN, lapisan ini melakukan proses konvolusi pada suatu citra, lalu menghasilkan citra baru yang disebut feature map. Convolution layer juga memiliki fitur pendeteksi, yang disebut sebagai kernel atau filter yang berfungsi melintasi bidang reseptif gambar. Kernel atau filter ini diterapkan di seluruh input dan dihitung antara piksel input dan filter, input merupakan array angka atau dapat disebut tensor. Perhitungannya kurang lebih seperti perkalian sesame matriks, perhitungan akan terus berlanjut dengan bergeser satu langkah hingga seluruh array angka pada input gambar dihitung dengan kernel. Output dari perhitungan antara input gambar dan kernel disebut sebagai feature map, activation map, atau convelved feature.

2. *Pooling layers*

Proses setelah convolution layers adalah pooling layers atau lapisan penyatuan, pooling layers atau down sampling berfungsi untuk mengurangi jumlah dimensi, dan meminimalkan jumlah parameter dalam input. Cara kerja dari lapisan ini adalah dengan memindahkan filter ke seluruh input tetapi tidak membawa bobot apapun. Pada proses ini kernel menempatkan fungsi agregasi ke nilai di dalam bidang reseptif.

3. *Fully connected layers*

Setelah fitur yang diekstraksi oleh lapisan konvolusi dan dilakukan downsampling oleh lapisan penyatuan, maka proses terakhir adalah lapisan yang terhubung sepenuhnya atau fully connected layers yang bertujuan untuk mengklasifikasikan fitur – fitur yang dihasilkan dari berbagai filter.

2.3. Mobilenet – SSD

Mobilenet merupakan arsitektur Convolutional neural network (CNN) yang dapat mengurangi beban pemrosesan berlebih. Perbedaan antara arsitektur Mobilenet dan CNN adalah pada lapisan konvolusi dengan ketebalan filter yang sesuai dengan ketebalan input gambar. Mobilenet menggunakan depthwise separable convolution yang terdiri dari 2 konvolusi, yaitu konvolusi depthwise dan pointwise, dan digunakan untuk tujuan klasifikasi, atau sebagai fitur ekstraktor.

Mobilenet menggunakan konvolusi yang dapat dipisahkan secara mendalam untuk membangun jaringan saraf dalam yang lebih ringan. Pada lapisan konvolusi regular, kernel, atau filter konvolusi diterapkan ke semua saluran input gambar, lalu menjumlahkan bobot piksel input dengan filter kemudian meluncur ke input piksel berikutnya di seluruh gambar. Lapisan berikutnya adalah kombinasi dari konvolusi depthwise dan pointwise. Konvolusi mendalam melakukan konvolusi pada setiap saluran secara terpisah, konvolusi mendalam ini digunakan untuk menyaring input saluran, lalu untuk konvolusi pointwise yang memiliki filter 1x1 bertujuan untuk menggabungkan keluaran saluran dari konvolusi mendalam untuk membuat fitur baru. Dengan begitu, pemrosesan yang perlu dilakukan lebih sedikit daripada jaringan konvolusi biasa.

Sedangkan SSD merupakan algoritma deep learning yang mendiskritasi ruang output dari kotak pembatas menjadi satu set kotak standar pada berbagai rasio dan skala aspek per lokasi peta fitur. Metode ini

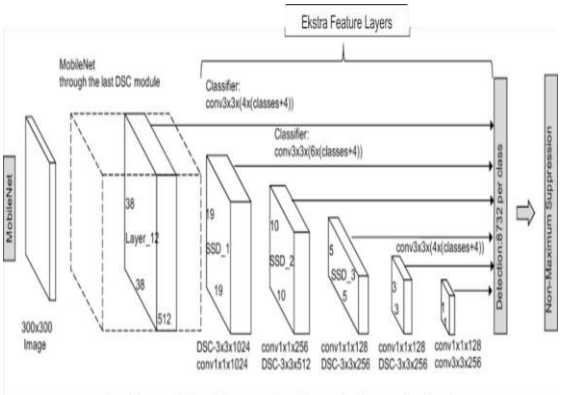
menerapkan fitur kotak pembatas untuk memperkirakan lokasi objek yang dideteksi [2]. Untuk pemrosesan dan nilai deteksi kecepatan pada metode SSD memiliki hasil yang tinggi sehingga metode ini cocok untuk diterapkan pada deteksi program secara real-time.

SSD juga menambahkan lapisan fitur konvolusional di ujung base network untuk memprediksi rasio aspek yang berbeda. Pada extra feature layers, arsitektur jaringan yang dapat digunakan dapat berupa inception, mobilenet, atau resnet.

SSD mempercepat proses dengan menghilangkan kebutuhan akan jaringan proposal wilayah, berbeda dengan metode Faster R-CNN yang lebih cepat menggunakan jaringan proposal wilayah untuk membuat kotak batas dan menggunakan kotak tersebut untuk mengklasifikasikan objek. Lalu untuk meningkatkan akurasi pada SSD, maka diterapkan beberapa peningkatan termasuk fitur multi-skala dan kotak default. Peningkatan tersebut memungkinkan SSD untuk mencocokkan akurasi Faster R-CNN menggunakan gambar beresolusi lebih rendah, yang selanjutnya meningkatkan kecepatan lebih tinggi.

Metode MobileNet menggunakan deep convolutional layer sebagai lapisan dasar jaringan, yang mengoptimalkan delay dengan mempertimbangkan ukuran model. Mobilenet menggunakan konvolusi yang dapat dipisahkan secara mendalam untuk membangun jaringan saraf dalam yang lebih ringan dan dapat meningkatkan kecepatan deteksi. Menggabungkan algoritma MobileNet dengan algoritma SSD dapat secara efektif meningkatkan kecepatan deteksi model [3]. Untuk arsitektur Mobilenet-SSD dapat dilihat pada **Gambar 4**.

Cara kerja Mobilenet-SSD adalah:



Gambar 4 Arsitektur Faster R-CNN

1. Mobilenet melakukan konvolusi *depthwise* dan *pointwise*,
2. Selanjutnya membuat feature map dengan skala yang berbeda – beda untuk gambar yang sudah masuk,
3. Selanjutnya adalah tahap prediksi untuk default box dengan ground-truth box dan mengklasifikasikan sebagai kecocokan positif atau negatif,
4. Selanjutnya adalah tahap non-maximum suppression yang bertugas untuk menghapus duplikat prediksi yang menunjuk ke objek yang sama.

2.4. *Confusion Matrix*

Pengujian dilakukan menggunakan *confusion matrix*. *Confusion matrix* merupakan teknik yang digunakan untuk melihat hasil performa dari sebuah algoritma yang dipakai. Selain itu ada juga hasil perhitungan seperti akurasi, presisi, recall, dan f1-score.

Berikut adalah empat nilai dari *confusion matrix* yaitu:

1. *True Positive* (TP): kelas A dideteksi sebagai kelas A.
2. *True Negative* (TN): kelas B dideteksi sebagai kelas B.
3. *False Negative* (FN): kelas A dideteksi sebagai kelas B.
4. *False Positive* (FP): kelas B dideteksi sebagai kelas A.

Berdasarkan nilai TP, TN, FN, dan FP; dapat dihitung nilai-nilai *accuracy*, *precision*, *recall*, dan F1 *score*:

1. *Accuracy*

Mewakili jumlah contoh data yang diklasifikasikan dengan benar berdasarkan jumlah total data. Cara menghitung akurasi:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

2. *Precision*

Merupakan perbandingan dari *true positives* dan keseluruhan data.

$$Precision = \frac{TTP}{TP + FP}$$

3. *Recall*

Merupakan perbandingan dari *true positives* kepada seluruh *positives*.

$$Recall = \frac{TP}{TP + FN}$$

4. F1-Score

Merupakan nilai rata-rata dari *precision* dan *recall*.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

3. Hasil Percobaan

Cara pengujian ini dilakukan menggunakan rak di sebuah toko lalu beberapa skenario untuk produk yang supermarket, dan kamera pada aplikasi (realtime). Sekanrio dibuat agar dapat membantu menyerupai produk yang ditaruh pada rak supermarket.

Lalu untuk pengujian akan dicoba menggunakan 150 epoch, 300 epoch, dan 500 epoch, selanjutnya dibuat confusion matrix untuk masing – masing epoch yang dilakukan untuk melihat kinerja model tersebut.

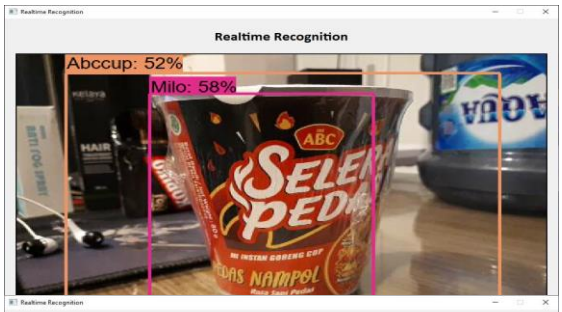
Dari hasil pengujian model sebelumnya dapat dilihat bahwa pengujian dengan 150 epoch menghasilkan akurasi lebih bagus dari pada epoch 300 dan 500. Jadi untuk pengujian menggunakan data test dengan skenario (2 produk, 3 produk, dan 5 produk), rak supermarket, dan juga realtime akan menggunakan 150 epoch.

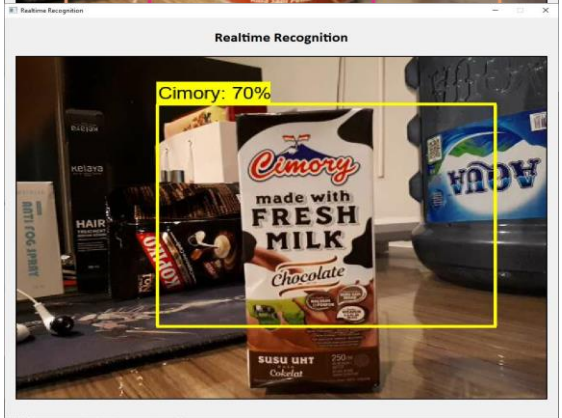
Tabel 1 Perbandingan epoch 150, 300, dan 500

	Akuras	Presisi	Recall	F1-Score
Epoch 150	86.08%	0.89	0.87	0.86
Epoch 300	80.38%	0.83	0.818	0.81
Epoch 500	80.92%	0.79	0.82	0.79

Hasil pendeteksian masih lebih bagus ditujukan kepada salah satu barang atau per barang dibandingkan dengan skenario 2, 3, 5 barang pada pendeteksian. Hal ini dapat dikarenakan data train yang kurang, warna dan bentuk pada objek yang mirip, dan epoch yang kurang.

Contoh hasil untuk pendeteksian *realtime* dapat dilihat pada **Gambar 5.**

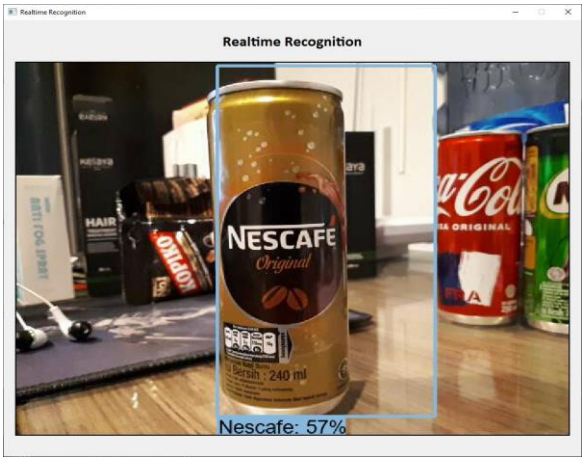




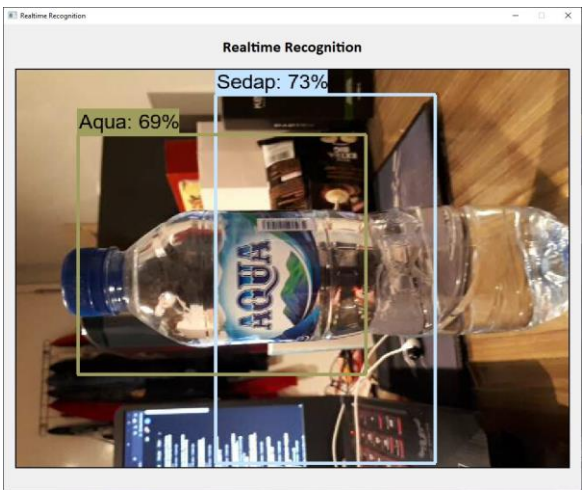
dibuat agar menyerupai susunan seperti di supermarket. Pengujian dilakukan untuk melihat apakah aplikasi yang sudah dibuat sudah berjalan dengan sesuai atau tidak.

Pengujian dilakukan menggunakan model Mobilenet SSD dengan menggunakan skenario yang dipakai berupa 2 produk, 3 produk, 5 produk, produk di rak

Gambar 5. Contoh hasil pendeteksian *realtime*



Gambar 5. (Lanjutan)



4. Kesimpulan

Berdasarkan hasil percobaan dalam proses pembuatan program sistem presensi berdasarkan pengenalan wajah dengan masker, kesimpulan yang dapat diambil adalah:

1. Pada pengujian diatas dapat diartikan bahwa semakin besar epoch yang penyelesaiannya maka akan semakin kecil hasil akurasinya, maka didapat lah epoch 150 dengan akurasi 86.08%,
2. Hasil dari validasi data train sudah cukup bagus tetapi hasil dari validasi data test tidak sesuai dengan barangnya dan bahkan ada yang tidak terdeteksi,
3. Hasil deteksi dari kamera realtime juga kurang bagus dan hanya beberapa yang dapat terdeteksi

REFERENSI

[1] Yamashita, R; Nishio, M; Do, R. K. G; Togashi, K. “Convolutional neural networks: an overview and application in radiology”. Insights into Imaging. 22 Juni 2018

[2] Hui, J. SSD object detection: Single Shot MultiBox Detector for real-time processing. https://jonathan-hui.medium.com/ssd-object-detection-single_shot-multibox-detector-for-real-time-processing-9bd8deac0e06. 14 Maret 2018

[3] Palwankar, T; dan Kothari, K. “Real Time Object Detection using SSD and MobileNet”. Ijraset Journal For Research in Applied Science and Engineering Technology. Volume 10, Nomor 3. 11 Maret 2022