

PEMBANGUNAN APLIKASI *MOBILE* PENGENALAN OBJEK UNTUK PENDIDIKAN ANAK USIA DINI

Muhammad Fadhlan Supriadi¹, Ema Rachmawati^{*2}, Anditya Arifianto³

¹²³Fakultas Informatika, Universitas Telkom
Email: ¹fadhlanupriadi@students.telkomuniversity.ac.id, ²emarachmawati@telkomuniversity.ac.id,
³anditya@telkomuniversity.ac.id
^{*}Penulis Korespondensi

(Naskah masuk: 17 November 2020, diterima untuk diterbitkan: 22 Maret 2021)

Abstrak

Penggunaan ponsel sudah sangat erat dengan kehidupan anak usia dini sehingga menimbulkan beberapa dampak negatif bagi anak usia dini terutama berkurangnya interaksi dengan dunia sekitarnya. Salah satu teknologi yang dapat dikembangkan pada ponsel adalah *computer vision*. Salah satu penggunaan *computer vision* adalah *object recognition* yang memberikan solusi untuk membantu mengenali objek. Pada penelitian ini dibangun sistem pengenalan objek benda di dalam rumah yang diaplikasikan pada ponsel yang diharapkan membantu anak usia dini mengenali benda disekitarnya. *MobileNet* merupakan salah satu *feature extraction* yang memiliki kinerja yang baik dan ringan digunakan pada perangkat ponsel. Arsitektur *MobileNet* terdiri dari *layer depthwise convolution* dan *layer pointwise convolution* dalam mengekstraksi fitur. Percobaan ini juga menggunakan arsitektur *Single Shot Multibox Detector (SSD)* sebagai metode dalam mendeteksi objek. *Pre-trained model* dari dataset *COCO* digunakan pada eksperimen, untuk mengenali 20 jenis objek benda di dalam rumah. Dari hasil eksperimen, *MobileNetV2* menghasilkan nilai *mean Average Precision (mAP)* yang lebih baik dibandingkan dengan *MobileNetV1* dan *InceptionV2*, yaitu sebesar 99,34%.

Kata kunci: *MobileNet, SSD, computer vision, pengenalan objek*

THE DEVELOPMENT OF OBJECT RECOGNITION MOBILE APPLICATION FOR EARLY CHILDREN EDUCATION

Abstract

Mobile phone usage has been very close to early childhood life, so giving rise to some negative impact on early childhood, especially reduced interaction with the surrounding world. One of the technologies that can be developed on the cellphone is computer vision. One of the uses of computer vision is object recognition that provides solutions to help to recognize objects. This research builds a system for recognition objects inside in house that is developed on a cellphone that is expected to help early childhood recognize objects in the surrounding. MobileNet is one of feature extraction that has good performance and efficient use on a cellphone. MobileNet architecture consists of a depthwise convolution layer and pointwise convolution layer in extracting features. The experiment also uses the architecture of Single Shot Multibox Detector (SSD) as a method of detecting objects. We used MobileNet architecture as a pre-trained model that had previously been trained on COCO datasets, and implement transfer learning for 20 types of objects commonly found inside the house. The experimental result indicates that the mean Average Precision (mAP) of MobileNetV2 could exceed MobileNetV1 and InceptionV2 of 99.34%.

Keywords: *MobileNet, SSD, computer vision, object recognition*

1. PENDAHULUAN

Perkembangan teknologi ponsel berkembang sangat pesat diikuti dengan penggunaan ponsel sebagai *platform* untuk mengakses informasi. Namun penggunaan ponsel menimbulkan masalah yang lebih luas untuk anak usia dini dengan rentang umur 3 s.d. 6 tahun yang menyebabkan anak usia dini akan menghabiskan lebih banyak waktu berinteraksi dengan ponsel dibandingkan berinteraksi dengan

dunia sekitarnya (Haddon, 2013). *Computer vision* merupakan salah satu area penelitian yang berkaitan dengan teknologi ini yang mengajarkan mesin agar dapat memiliki kemampuan melihat dan mengenal objek (Dua & Cheng, 2014). Salah satu implementasi dari *computer vision* adalah *Object Recognition* atau pengenalan objek.

Dalam beberapa tahun sebelumnya, arsitektur *Convolutional Neural Network (CNN)* pada

pengenalan objek telah mengalami perkembangan yang signifikan. *Inception* merupakan salah satu arsitektur model jaringan dasar yang memberikan teknik faktorisasi agar ukuran jaringan tidak menjadi terlalu besar sehingga dapat mengurangi waktu pengenalan objek dan mempertahankan akurasi (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016). Akan tetapi, kerumitan pada perangkat ponsel yang memiliki daya komputasi dan sumber daya memori yang terbatas mendorong peningkatan efektivitas model *Convolutional Neural Network (CNN)* agar dapat berfungsi dengan baik pada perangkat ponsel.

Salah satu model *Convolutional Neural Network (CNN)* yang dapat dijalankan pada perangkat ponsel dengan kinerja yang baik adalah *MobileNet*. *MobileNet* memiliki spesifikasi rendah, latensi rendah, dan daya konsumsi rendah sehingga sangat disarankan untuk diimplementasikan pada perangkat ponsel (Howard, 2017). Perbedaan mendasar antara arsitektur *MobileNet* dengan arsitektur CNN lainnya adalah penggunaan *layer* konvolusi dengan ketebalan filter yang sesuai dengan ketebalan dari masukan citra. *MobileNet* menerapkan arsitektur *depthwise separable convolution* yang membagi *layer* konvolusi menjadi *depthwise convolution* (Wang, Yuan, & Li, 2018) dan *pointwise convolution* (Hua, Tran, & Yeung, 2018) serta dalam pengembangannya telah ditambahkan arsitektur baru yaitu *linear bottleneck* dan *inverted residuals* (Sandler, Zhu, Zhmoginov, & Jan, 2018).

Pada penelitian ini diaplikasikan *transfer learning* pada model *MobileNet* yang sebelumnya telah dilatih pada *COCO datasets* (Lin dkk, 2014) untuk digunakan pada 20 objek benda. Selanjutnya implementasi pada perangkat ponsel menggunakan *ML-Kit Firebase* sebagai penghubung model dengan ponsel. Aplikasi yang dibangun diharapkan dapat mengedukasi penggunaan ponsel yang baik untuk anak usia dini dalam mengenali objek-objek disekitarnya.

Susunan makalah adalah sebagai berikut. Deskripsi sistem yang diusulkan terdapat pada bagian 2. Eksperimen dan analisis pada bagian 3 dan ditutup dengan kesimpulan pada bagian 4.

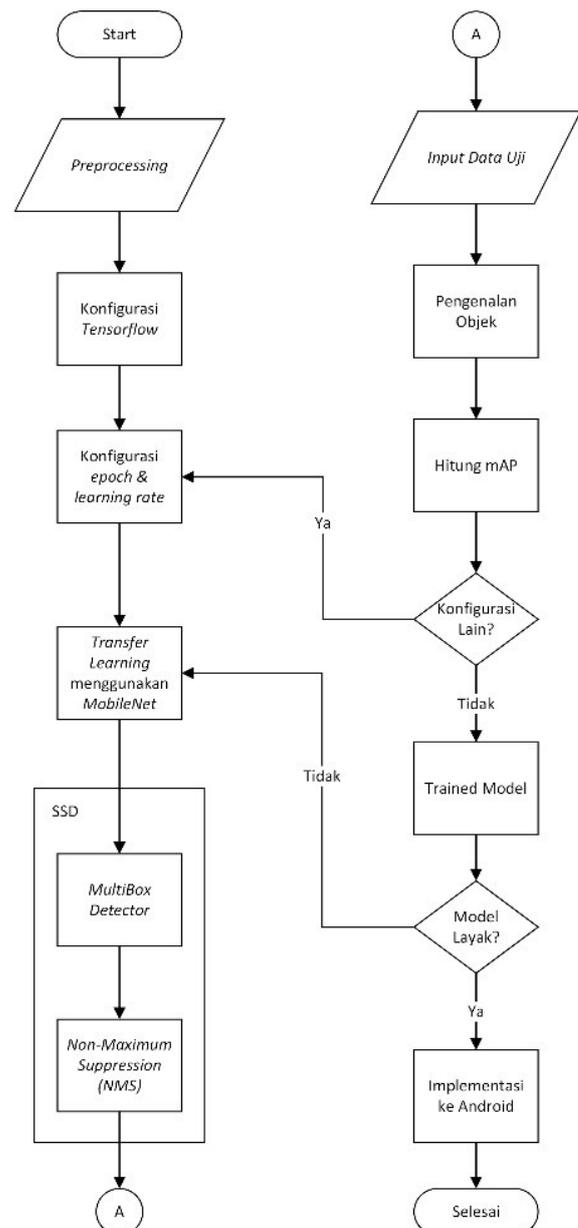
2. SISTEM YANG DIUSULKAN

Dalam penelitian ini dibuat rancangan sistem pengenalan objek dengan data masukan berupa citra. Sistem menggunakan *MobileNet* dengan mengaplikasikan *transfer learning* pada dataset baru dan *Single Shot Multibox Detector (SSD)* sebagai metode dalam mendeteksi objek. Secara umum penelitian ini memiliki skema seperti Gambar 1.

Citra masukan akan melalui tahap praproses terlebih dahulu berupa pemberian label *bounding box* yang akan digunakan pada tahap deteksi. Label ini berfungsi sebagai *ground truth bounding box* yang menjadi acuan apakah terdapat objek pada citra tersebut.

2.1. Penentuan Konfigurasi Sistem

Setelah proses *preprocessing* dilakukan maka terdapat data berupa citra dan label, namun data tersebut kemudian dikonversi terlebih dahulu sebelum dapat digunakan oleh *Tensorflow*. Citra yang telah dilabeli memiliki format file xml yang perlu dikonversi ke file berformat csv untuk menggabungkan seluruh file xml ke dalam 1 file pelatihan atau pengujian. Setelah file dikonversi ke format csv, selanjutnya file tersebut dikonversi lagi menjadi file TFRecord di mana file tersebut merupakan format dataset yang dapat digunakan pada *Tensorflow*. Selain itu membuat file baru *label map* yang berisi indeks dari 20 objek *dataset*.



Gambar 1. Skema umum pembangunan aplikasi

Pada penelitian ini digunakan konfigurasi berbeda-beda. Pada tahap konfigurasi, nilai *epoch* dan *learning rate* disetel untuk menguji kombinasi

parameter yang dapat memaksimalkan akurasi pengenalan objek.

2.2. Transfer Learning dengan MobileNet

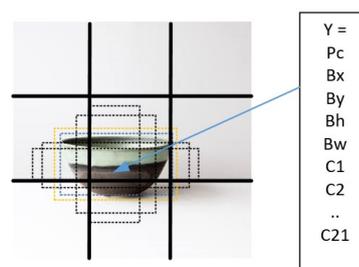
Pada arsitektur MobileNet (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) terdapat tiga *layer* konvolusi yang harus dilalui citra dalam mengekstraksi fitur. Batch normalization ditambahkan pada *layer* konvolusi yang berguna untuk meningkatkan kinerja dan stabilitas jaringan saraf tiruan. Fungsi aktivasi yang digunakan pada *MobileNet* adalah ReLU6 (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018). Dibandingkan dengan ReLU yang biasa, ReLU6 lebih kuat ketika menggunakan komputasi presisi yang rendah tetapi mencegah aktivasi yang terlalu besar.

Transfer learning dilakukan dengan cara menggunakan *knowledge* yang terdapat pada *pre-trained MobileNet COCO* yang memiliki 91 kelas objek dan 328.000 citra (Lin dkk, 2014). *Knowledge* tersebut disesuaikan dengan kelas objek yang digunakan saat pelatihan, yaitu 20 kelas objek.

Proses *transfer learning* dapat dilakukan dengan beberapa langkah, langkah awal dengan menghapus *fully-connected layer* terakhir dari *pre-trained model* yang berisi klasifikasi dari 91 kelas objek. Langkah selanjutnya, mengganti lapisan tersebut dengan *fully-connected layer* baru yang hanya memiliki klasifikasi 20 kelas objek ditambah dengan kelas *background*. Langkah terakhir, dilakukan *fine tuning* yang bertujuan melatih *linear classifier* sehingga dapat digunakan pada data yang lebih sedikit (Pan & Yang, 2010).

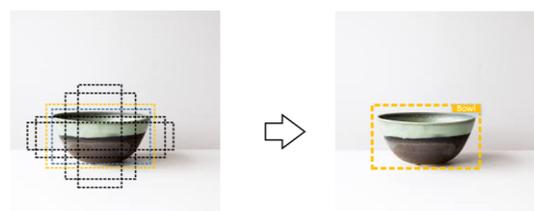
2.3. Single-Shot Multibox Detector (SSD)

Single Shot Multibox Detector (SSD) menggunakan metode *MultiBox Detector* untuk regresi *bounding box* dalam menentukan prediksi lokasi objek sekaligus mengklasifikasikannya pada suatu masukan citra. Cara kerja *MultiBox Detector* adalah dengan menyesuaikan *classifier* pada setiap *layer* konvolusi yang berukuran $m * n * b * (4 + c)$ dengan $m * n$ adalah ukuran *feature map* (contohnya 3x3 yang akan menghasilkan 9 sel atau grid), b adalah jumlah *predict bounding box* per grid, dan c adalah jumlah kelas ditambah 4 offset untuk ukuran setiap *predict bounding box*. Ilustrasi dapat dilihat pada Gambar 2. Selanjutnya SSD akan memeriksa apakah ada objek yang dikenali pada setiap grid pada *feature map*, ketika mendapatkan objek maka SSD akan membentuk *predict bounding box* baru dengan ukuran rasio yang berbeda-beda dengan jumlah maksimal 6 *bounding box* pada setiap grid. Adapun isi dari setiap *predict bounding box* adalah P_c untuk probabilitas softmax kelas target, B_x dan B_y adalah titik tengah pada grid tersebut sehingga tidak akan pernah melebihi nilai lebih dari 1, B_h & B_w adalah ukuran lebar dan tinggi dari *bounding box* tersebut, dan C_1 hingga C_{21} nilai skor prediksi tiap kelas (Liu dkk, 2016).



Gambar 2. *MultiBox Detector*

Metode *Non-Maximum Suppression (NMS)* membantu dalam mengurangi jumlah *predict bounding box* pada *feature map* agar dapat mendapatkan *bounding box* yang mendekati target kelas objek (Gambar 3). Di tahap awal, *Non-Maximum Suppression (NMS)* akan menghilangkan seluruh *bounding box* yang memiliki nilai P_c yang kurang dari sama dengan 0,6, lalu selanjutnya selama masih ada *bounding box* yang tersisa lakukan 2 tahap berikut: yaitu pilih *bounding box* dengan nilai P_c tertinggi dan hilangkan *bounding box* tersisa dengan nilai IoU lebih dari sama dengan 0,5 dengan *predict bounding box* sebelumnya. Tahap ini dilakukan sampai hanya tersisa satu *predict bounding box* untuk satu objek pada suatu citra (Hosang, Benenson, & Schiele, 2017).



Gambar 3. *Non-Maximum Suppression (NMS)*

2.4. Implementasi Model ke dalam Aplikasi Mobile

Implementasi model ke perangkat *android* menggunakan *Firebase ML KIT* yang menyediakan *mobile SDK machine learning*. *Firebase ML KIT* membantu agar dapat menghubungkan model yang telah dirancang sebelumnya ke aplikasi yang akan dibangun. Pada *Firebase ML KIT* dibuatkan *project* baru serta mengupload hasil ekspor model pada *custom model*. Selanjutnya pada *project android*, dibuatkan juga file label baru yang berisi nama-nama objek pada dataset yang disatukan beserta hasil ekspor model ke dalam folder "*asset*" pada *project android*.

Aplikasi *mobile* yang dibangun terdiri dari 3 (tiga) jenis soal berdasar tingkat kesulitan, yaitu:

1. *Easy*: Aplikasi memberitahu secara langsung label objek yang harus dicari.
2. *Medium*: Aplikasi memberikan deskripsi singkat tentang objek yang harus dicari.
3. *Hard*: Aplikasi memberikan gambar bentuk dari sebuah objek yang harus dicari.

Setiap jenis soal terdiri dari 10 soal objek berbeda dengan setiap soal memuat nilai skor 10 ketika

pengguna berhasil menebak objek tersebut. Setelah 10 soal objek telah dijawab maka aplikasi akan secara otomatis mengkalkulasikan skor akhir yang didapatkan.

2.5. Pengukuran Kinerja

Kinerja sistem diukur berdasarkan parameter mAP (*mean Average Precision*). Nilai mAP (*mean Average Precision*) menunjukkan tingkat pengenalan objek beserta lokalisasinya. Nilai ini menjadi parameter utama untuk mengukur kinerja suatu model yang dirancang dalam mengenali objek. Namun sebelum menghitung nilai mAP, *Average Precision (AP)* untuk setiap objek harus dihitung dulu pada dataset berdasarkan model yang digunakan. Dalam mendapatkan nilai *Average Precision (AP)* diperlukan juga mendefinisikan *True Positives* dan *False Positives* menggunakan *IoU* untuk mengidentifikasi apakah deteksi benar (*True*) atau salah (*False*). *Intersection over Union (IoU)* pada persamaan (1) merupakan rasio antara *intersection* dan *union* dari *ground-truth bounding boxes* (B_{gt}) dengan *predict bounding boxes* (B_p) (Rosenbrock, 2017).

$$\begin{aligned} \text{Intersection over Union (IoU)} \\ &= \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} \end{aligned}$$

Ambang batas *IoU* yang digunakan adalah 0,5, sehingga jika $\text{IoU} \geq 0,5$ maka akan dianggap *True Positive (TP)* begitu pula sebaliknya maka akan dianggap *False Positive (FP)*. Setiap citra memiliki data *ground truth* yang menunjukkan nilai akurasi prediksi dari objek tertentu pada citra tersebut. Maka dengan *IoU* dapat dihitung jumlah prediksi yang benar dan prediksi yang salah untuk menentukan *Average Precision (AP)* pada setiap *Class (c)* dengan persamaan (2).

$$\text{Average Precision (AP)} = \frac{TP(c)}{TP(c) + FP(c)}$$

Nilai *mean Average Precision (mAP)* selanjutnya dihitung menggunakan persamaan (3) (Henderson & Ferrari, 2017).

$$\begin{aligned} \text{mean Average Precision (mAP)} \\ &= \frac{1}{N} \sum_{c \in N} \frac{TP(c)}{TP(c) + FP(c)} \end{aligned}$$

3. EKSPERIMEN DAN ANALISIS

Pada bagian ini dipaparkan data yang digunakan pada eksperimen, pengujian *pre-trained model* pada beberapa arsitektur, hasil pengujian menggunakan beberapa parameter *epoch* dan *learning rate*, dan hasil

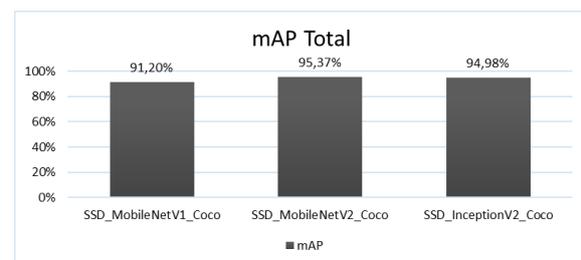
pengujian aplikasi yang dibangun dengan langsung melakukan simulasi langsung pada anak usia dini.

3.1. Data yang Digunakan

Data yang digunakan dalam penelitian ini adalah 20 jenis objek yang sering ditemukan di dalam rumah karena akan lebih mudah dan aman untuk ditemukan oleh anak usia dini. Adapun 20 jenis objek tersebut yaitu *bowl, broom, bucket, calculator, cap, clock, dustbin, fan, flashlight, hanger, iron, keyboard, laptop, remote, scissors, spoon, stapler, teapot, thermos, dan toothbrush*. Data dikumpulkan dari citra yang terdapat pada *Google* yang berukuran sekitar 255 x 255 dpi, dengan setiap objek terdiri dari 200 citra sehingga untuk keseluruhan data terdapat 4000 citra yang dibagi menjadi 3200 citra untuk data pelatihan dan 800 citra untuk data uji. Proses pelabelan dilakukan pada setiap citra yang berguna untuk mendapatkan *ground truth bounding box* yang menjadi acuan apakah terdapat objek pada citra tersebut.

3.2. Pemilihan Arsitektur

Parameter pada pengujian ini adalah *epoch* 50, *batch* 5, dan *learning rate* 0,004. Hasil pengujian *pre-trained model* untuk mengetahui model terbaik yang cocok dengan dataset dapat dilihat pada Gambar 4.



Gambar 4. Hasil Pengujian *Pre-trained Model* berdasarkan mAP Total

Hasil data pengujian pada Gambar 4 menunjukkan bahwa nilai *pre-trained model* SSD_MobileNetV2_Coco mendapatkan nilai mAP paling besar (yaitu 95,37%) dibandingkan dengan SSD_InceptionV2_Coco yang memiliki nilai mAP 94,98% dan SSD_MobileNetV1_Coco yang memiliki nilai mAP 91,20%. *MobileNetV2* memiliki nilai mAP yang lebih baik daripada *InceptionV2* dikarenakan *MobileNet* menggunakan konvolusi *depthwise separable* yang memiliki jaringan konvolusi yang lebih mendalam dibandingkan *InceptionV2* yang menggunakan konvolusi standar sehingga ini menghasilkan jumlah parameter yang lebih sedikit pada *MobileNetV2* dibandingkan *InceptionV2*.

Sementara itu, *MobileNetV1* memiliki nilai mAP yang rendah dibandingkan *MobileNetV2* dikarenakan pada *MobileNetV2* memiliki lapisan *inverted residual bottleneck* tambahan yang membuat *MobileNetV2* akan memiliki lapisan konvolusi yang lebih banyak dibandingkan *MobileNetV1*, sehingga pada pelatihan *MobileNetV2* membutuhkan waktu yang lebih lama

dibandingkan *MobileNetV1* bahkan waktu pelatihan *InceptionV2* juga masih lebih lama dan hampir setara waktu pelatihannya dengan *MobileNetV2* dibandingkan *MobileNetV1* yang lebih cepat.

Tabel 1. Hasil Pengujian Parameter *Epoch* dan *Learning rate*

No	Epoch	Learning Rate	mAP (%)
1	50	0,04	83,99
2	50	0,004	95,37
3	50	0,0004	98,63
4	100	0,04	97,73
5	100	0,004	98,10
6	100	0,0004	99,34

3.3. Pemilihan Parameter *Epoch* dan *Learning Rate*

Berdasarkan hasil tahap pengujian pada subbagian 3.2, maka *pre-trained model* yang digunakan pada tahap pengujian ini adalah *SSD_MobileNetV2_Coco*. Selanjutnya dilakukan pengujian parameter *epoch* dan *learning rate*. Pengujian yang dilakukan menggunakan parameter *epoch* 50 dan 100, sedangkan *learning rate* menggunakan 0,04, 0,004, dan 0,0004 serta masih menggunakan *batch* 5. Hasil pengujian parameter *epoch* dan *learning rate* terhadap nilai mAP dapat dilihat pada Tabel 1.

Hasil eksperimen menunjukkan bahwa nilai *epoch* dan *learning rate* dapat berpengaruh pada tingkat akurasi. Maka dapat disimpulkan bahwa semakin besar nilai *epoch* maka semakin baik pula tingkat akurasinya, sedangkan jika semakin besar *learning rate* maka semakin buruk pula tingkat akurasinya dikarenakan ketelitian jaringan semakin berkurang yang disebabkan jaringan tersebut tidak stabil dan menyebabkan nilai *error* yang berulang pada nilai tertentu sehingga mencegah mencapai target yang diharapkan. Maka selanjutnya hasil pelatihan dari *SSD_MobileNetV2_Coco* dengan parameter *epoch* 100 dan *learning rate* 0,0004 digunakan pada implementasi model ke aplikasi *android*.

3.4. Tahap Pengujian

Pre-trained model yang digunakan pada tahap pengujian ini adalah *SSD_MobileNetV2_Coco* dengan parameter *epoch* 100 dan *learning rate* 0,0004. Aplikasi berbasis *android* yang dibangun bertujuan untuk mengetahui hasil deteksi objek dengan berbagai sudut arah kamera yaitu depan, atas, kanan, dan kiri dengan latar putih. Sepuluh objek yang diujikan adalah *bowl*, *clock*, *keyboard*, *remote*, *spoon*, *calculator*, *laptop*, *cap*, *stapler*, dan *thermos*. Hasil pengujian pada aplikasi dapat dilihat pada Tabel 2.

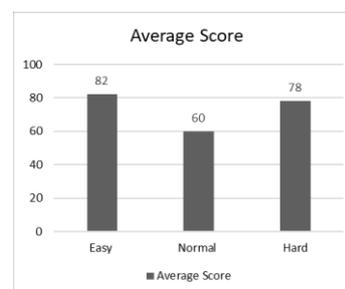
Hasil data pengujian pada Tabel 2 menunjukkan bahwa sudut arah kamera dari depan, atas, kiri dan kanan dapat mendeteksi hampir seluruh objek dengan latar putih. Maka berdasarkan hasil pengujian ini dapat dianalisis bahwa sudut arah kamera dari depan dapat mendeteksi seluruh objek dengan akurasi yang paling

tinggi dibandingkan sudut arah kamera dari atas, kanan, dan kiri. Objek *remote* dan *calculator* memiliki karakteristik bentuk yang memiliki banyak tombol didepannya sehingga akan lebih sulit dideteksi selain dari arah depan sedangkan untuk objek *spoon* juga sulit dideteksi dari ketiga arah ini dikarenakan karakteristik bentuknya yang tipis dibandingkan dengan objek lainnya. Selain itu objek *stapler* tidak dapat dideteksi dari arah atas dikarenakan karakteristik bentuknya yang hampir sama hanya pada bagian depan, kanan dan kiri saja. Sehingga berdasarkan analisis pengujian tahap ini, dapat disimpulkan bahwa sudut pandang objek yang bervariasi pada dataset akan mempengaruhi hasil kinerja model dalam mengenali objek pada perangkat ponsel.

Tabel 2. Hasil Pengujian Object Detection Real-time

No	Objek	Deteksi dari Depan (Score)	Deteksi dari Atas (Score)	Deteksi dari Kanan (Score)	Deteksi dari Kiri (Score)
1	<i>Bowl</i>	Ya (0.96)	Ya (0.80)	Ya (0.89)	Ya (0.87)
2	<i>Spoon</i>	Ya (0.93)	Ya (0.79)	Ya (0.70)	Ya (0.78)
3	<i>Clock</i>	Ya (1.00)	Ya (0.74)	Ya (0.96)	Ya (0.98)
4	<i>Laptop</i>	Ya (1.00)	Ya (0.99)	Ya (1.00)	Ya (1.00)
5	<i>Keyboard</i>	Ya (0.99)	Ya (0.86)	Ya (0.99)	Ya (0.99)
6	<i>Remote</i>	Ya (0.97)	Ya (0.87)	Ya (0.68)	Ya (0.73)
7	<i>Stapler</i>	Ya (0,84)	Tidak	Ya (0.89)	Ya (0.83)
8	<i>Thermos</i>	Ya (0,99)	Ya (0,98)	Ya (0,96)	Ya (0,91)
9	<i>Calculator</i>	Ya (0,94)	Ya (0,87)	Ya (0,85)	Ya (0,89)
10	<i>Hanger</i>	Ya (0,99)	Ya (0,88)	Ya (0,99)	Ya (0,99)

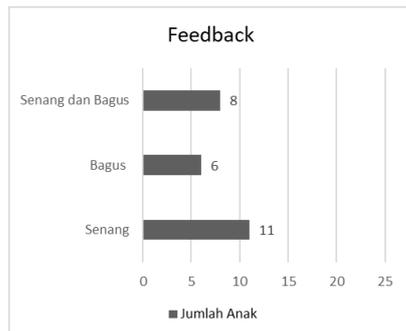
3.5. Pengujian Aplikasi Edukasi Anak Usia Dini dalam Mengenali Objek



Gambar 5. Hasil rata-rata skor pengujian aplikasi pada anak-anak

Pengujian aplikasi edukasi anak usia dini dalam mengenali objek dilakukan dengan penggunaan aplikasi secara langsung pada 25 anak usia dini. Hasil pengujian aplikasi edukasi anak usia dini dalam mengenali objek dapat dilihat pada Gambar 5 dan Gambar 6. Sepuluh orang anak mencoba tipe soal *easy*, 5 (lima) anak mencoba tipe soal *normal*, dan 11 anak mencoba tipe soal *hard*. Rata-rata skor untuk tipe

soal *easy* adalah 82, untuk tipe soal normal adalah 60, dan untuk tipe soal *hard* adalah 78.



Gambar 6. Hasil *feedback* pengujian aplikasi pada anak-anak

Feedback dari pengguna terkait aplikasi dikumpulkan dengan memberikan 2 (dua) pertanyaan yaitu (1) apakah pengguna merasa senang saat menggunakan aplikasi dan (2) apakah pengguna merasa aplikasi ini bagus. Hasilnya menunjukkan bahwa 8 pengguna merasa senang dan menilai aplikasinya bagus, 11 merasa hanya senang, dan 6 merasa aplikasi hanya bagus. Maka berdasarkan hasil pengujian ini dapat dianalisis bahwa anak usia dini sangat tertarik dengan tipe soal yang menampilkan gambar dibandingkan tulisan dan anak usia dini dapat mendapatkan hasil score antara 80 atau 90 untuk tipe soal bergambar dibandingkan dengan tipe soal tulisan. Sedangkan ketertarikan anak usia dini dengan aplikasi edukasi ini menampilkan reaksi yang antusias selama simulasi di mana *feedback* dari anak usia dini memberikan respon yang positif yaitu merasa senang ataupun bagus terkait aplikasinya. Berdasarkan eksperimen yang dilakukan, dapat ditunjukkan bahwa aplikasi yang dibangun dapat diaplikasikan pada penggunaan yang lebih luas, yaitu untuk memberikan edukasi penggunaan ponsel yang baik kepada anak usia dini dengan cara mengenalkan objek-objek yang terdapat di lingkungan sekitarnya.

4. KESIMPULAN

Pada penelitian ini, kami mencoba mengimplementasikan *model MobileNet* yang telah dilatih pada *COCO dataset* yang memiliki 91 kelas objek dan diaplikasikan *transfer learning* pada dataset baru yaitu 20 kelas objek. Dari eksperimen yang dilakukan, kami mendapatkan hasil nilai mAP sebesar 99,34% menggunakan *pre-trained model* pada arsitektur *SSD MobileNetV2_Coco* dengan parameter *epoch* 100, *learning rate* 0,0004, dan *batch* 5. Sedangkan hasil simulasi aplikasi edukasi ini menunjukkan bahwa anak usia dini antusias dengan kehadiran aplikasi ini dengan memberikan *feedback* yang positif dikarenakan aplikasi ini memberikan metode pembelajaran yang lebih interaktif apalagi menggunakan media ponsel yang digemari anak usia dini pada saat ini.

Namun pada aplikasi edukasi ini masih memiliki kekurangan dalam mengenali objek terutama pada

faktor *background environment* dan *viewpoint* yang menyebabkan aplikasi mengalami kesulitan dalam mengenali objek dengan kedua faktor tersebut. Oleh karena itu, untuk penelitian lebih lanjut disarankan agar menambah dataset yang lebih bervariasi dengan latar objek yang berbeda ataupun sudut pandang objek yang berbeda pada setiap citra agar dapat mengatasi objek dengan *background environment* dan *viewpoint* yang berbeda-beda. Dan juga dapat mencoba *pre-trained model* yang lebih besar seperti dataset *ImageNet* yang memiliki 1000 kelas sehingga sistem dapat digunakan untuk mengenali berbagai macam kelas objek selain 91 kelas objek yang ada pada *COCO dataset*.

DAFTAR PUSTAKA

- DU, C.J., CHENG, Q., 2014. Computer Vision. In: O'Donnell C., Fagan C., Cullen P. (eds) *Process Analytical Technology for the Food Industry*. Food Engineering Series. Springer, New York, NY.
- HADDON, L., 2013. Mobile media and children. *Mob. Media Commun.*
- HENDERSON, P. dan FERRARI, V. 2017. End-to-end training of object class detectors for mean average precision. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*
- HOWARD, A. G. dkk., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv*.
- HOSANG, J., BENENSON, R., AND SCHIELE, B., 2017. Learning non-maximum suppression. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.
- HUA, B., TRAN, M.-K., AND YEUNG, S.-K., 2018. Pointwise Convolutional Neural Networks. *Cvpr*.
- LIN, T. Y. dkk., 2014. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*
- LIU, W. dkk., 2016. SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.
- PAN, S. J. dan YANG, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*.
- ROSENBROCK, A., 2017. Intersection over Union (IoU) for object detection. *Pyimagesearch.com*.
- SANDLER, M., ZHU, M., ZHMOGINOV, A. dan JAN, C.V., 2018. Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation.

CVPR.

- SANDLER, M., HOWARD, A., ZHU, M., ZHMOGINOV, A. dan CHEN, L. C., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
- SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J., dan WOJNA, Z., 2016. Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2818-2826
- WANG, G., YUAN, G., LI, T. dkk, 2018. An multi-scale learning network with depthwise separable convolutions. *IP SJ T Comput Vis Appl* 10 (11).

Halaman ini sengaja dikosongkan