

Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra-Terlatih

Prisky Ratna Aningtiyas, Agus Sumin dan Setia Wirawan

Sistem Informasi, Universitas Gunadarma

Jalan Margonda Raya No. 100, Depok

E-mail : priskyra@gmail.com, setia.gunadarma@gmail.com

Abstrak

Deteksi objek merupakan salah satu teknik untuk menemukan objek dalam gambar atau video. Salah satu metode untuk membuat deteksi objek adalah menggunakan *TensorFlow Object Detection API*. Metode tersebut menyediakan model pra – terlatih yang dapat dimanfaatkan dalam pembuatan aplikasi deteksi objek. Model yang digunakan dalam penelitian ini adalah SSD *Mobilenet V2*. Model tersebut dapat melakukan deteksi objek dengan menghasilkan akurasi dan area terdeteksi untuk keberadaan setiap kategori objek pada suatu gambar. Oleh karena itu, penelitian ini akan membuat aplikasi deteksi objek dengan memanfaatkan metode tersebut. Penelitian ini bertujuan untuk membuat aplikasi deteksi objek menggunakan *TensorFlow Object Detection API* dengan memanfaatkan SSD *Mobilenet V2* sebagai model pra – terlatih dalam penerapan ilmu *Deep Learning*. Aplikasi diharapkan dapat melakukan deteksi dan mengukur akurasi objek, yaitu *Camera*, *Handphone*, *Headphone*, *Laptop*, dan *Mouse* melalui input gambar. Dataset yang digunakan dibuat dengan mengumpulkan sebanyak 500 gambar dengan membagi menjadi tiga bagian, yaitu train set, validation set, dan test set dengan masing – masing perbandingan sebesar 70% : 20% : 10%. Pelatihan dilakukan dengan bantuan *Google Research Colaboratory* sebagai *virtual machine*. Penelitian ini menggunakan Python 3.6.8 sebagai bahasa pemrograman dan memakai beberapa *library* yang disediakan oleh Python. Dari uji coba yang dilakukan, aplikasi ini memiliki tingkat akurasi sebesar 93.02% pada *test set*.

Kata Kunci: Deep Learning, Deteksi Objek, Python, TensorFlow

Pendahuluan

Belajar adalah proses perubahan tingkah laku individu sebagai hasil dari pengalamannya dalam berinteraksi dengan lingkungan. Belajar bukan hanya sekedar menghafal, melainkan suatu proses mental yang terjadi dalam diri seseorang [1]. Pembelajaran adalah proses untuk menjadikan makhluk hidup belajar. Proses pembelajaran manusia didapat dari pengalaman yang diperoleh. Belajar yang dilakukan oleh manusia tidak sama dengan pembelajaran yang dilakukan oleh suatu mesin atau komputer. Komputer memiliki cara belajar yang berbeda yang tidak sama seperti manusia. Salah satu proses pembelajaran mesin, yaitu Supervised Learning. Supervised Learning adalah pembelajaran terarah atau terawasi. Supervised Learning dengan pembelajaran dengan mengajarkan model neural network pengetahuan. Tipe pembelajaran tersebut menyediakan data input dan output dalam proses pembelajaran.

Dalam dunia pembelajaran mesin dikenal istilah Machine Learning dan Deep Learning. Kedua istilah tersebut mengacu pada pengembangan sistem yang mampu belajar dengan menyediakan data sebagai bahan belajar (training) sebelum mengeluarkan output. Perbedaan dari Machine Learning dan Deep Learning adalah pada Deep Learning pembelajaran dilakukan dengan lebih mendalam dan kompleks.

Pengembangan lebih jauh dikenal istilah Computer Vision yang mengajarkan bagaimana komputer dapat melihat. Mesin diharapkan mampu untuk mengekstrak informasi dari data berupa gambar untuk menyelesaikan tugas tertentu. Salah satu teknik Computer Vision adalah dalam hal pendeteksian objek. Deteksi objek adalah teknik Computer Vision untuk dapat menemukan contoh objek dalam gambar atau video. Algoritma yang digunakan dalam mendeteksi objek memanfaatkan pembelajaran mesin mendalam untuk dapat menghasilkan hasil yang baik. Salah satu metode untuk

membuat deteksi objek adalah menggunakan TensorFlow Object Detection API yang menyediakan berbagai model pra – terlatih. Model pra – terlatih yang dapat digunakan dalam pembuatan deteksi objek adalah SSD Mobilenet V2. Model tersebut dapat melakukan deteksi objek dengan menghasilkan akurasi dan area terdeteksi untuk keberadaan setiap kategori objek pada suatu gambar. Model ini menghasilkan besar akurasi untuk keberadaan setiap kategori objek.

Beberapa penelitian yang membahas mengenai TensorFlow Object Detection API diantaranya adalah peneliti yang menggunakan TensorFlow Object Detection API dengan arsitektur SSD Mobilenet untuk mendeteksi lesi kulit. Penelitian tersebut, dilakukan lokalisasi objek untuk mendeteksi lesi kulit menggunakan model SSD-Mobilenet pada ISIC 2018 sebagai data pelatihan dan pengujian dataset. Untuk mengevaluasi kinerja deteksi, proses deteksi dilakukan dengan menggunakan dua metode yang berbeda, yaitu aplikasi ponsel real-time dari kamera Android (Galaxy S6), dan Jupyter Notebook dari TensorFlow Object Detection Application Program Interface (API). Total mAP adalah 96,04% dengan total loss 0,78. Hasil eksperimen mencapai 99% akurasi deteksi saat menggunakan Jupyter Notebook, sementara itu mencapai 100% dengan deteksi Android. Percobaan telah dilaksanakan pada sistem Ubuntu 16.04 LTS GTX1070 @ 2.80GHZ x8[2].

Selanjutnya, penelitian yang merupakan metode baru untuk mendeteksi tindakan yang terkait dengan kesehatan manusia (HHRA) dari urutan video menggunakan kamera Android. Penelitian ini, menggunakan dua pendekatan, yaitu model SSD Mobilenet dan Faster RCNN Resnet. Pendekatan yang diusulkan dievaluasi pada dataset NTU RGB + D, yang dikenal sebagai dataset pengakuan tindakan 3D yang dapat diakses publik terbesar saat ini. Dataset telah dibagi menjadi dataset pelatihan dan pengujian. Kualitas deteksi skor kepercayaan total (total mAP) untuk semua kelas tindakan adalah 95,8% berdasarkan pada model SSD-Mobilenet dan 93,8% berdasarkan pada model Faster-R-CNN-Resnet. Proses deteksi dicapai dengan menggunakan dua metode untuk mengevaluasi kinerja deteksi menggunakan kamera Android (Galaxy S6) dan menggunakan TensorFlow Object Detection Notebook dalam hal akurasi dan kecepatan deteksi. Hasil eksperimental telah menunjukkan peningkatan dalam hal akurasi deteksi dan efisiensi untuk identifikasi tindakan terkait kesehatan manusia. Eksperimen telah dijalankan pada sistem Ubuntu 16.04 LTS GTX1070 @ 2.80GHZ x8[3].

Penelitian yang merupakan penelitian dengan metode tradisional dalam Machine Learning untuk mendeteksi lampu lalu lintas dan klasifikasi dari metode deteksi objek pembelajaran dalam dengan keberhasilan membangun jaringan saraf convolutional (CNN), yang merupakan komponen dari

Deep Learning. Makalah ini menyajikan pendekatan Deep Learning untuk mendeteksi lampu lalu lintas dengan membandingkan dua model deteksi objek dan dengan mengevaluasi fleksibilitas TensorFlow Object Detection Framework untuk menyelesaikan masalah secara real-time. Termasuk Single Shot Multibox Detector (SSD) MobileNet V2 dan Faster-RCNN. Studi eksperimental menunjukkan bahwa Faster-RCNN menghasilkan 97,015%, yang mengungguli SSD sebesar 38,806% untuk model yang telah dilatih menggunakan 441 gambar[4].

Penelitian yang meningkatkan tugas deteksi objek untuk kendaraan self-driving dengan menggunakan TensorFlow API diikuti oleh jaringan saraf MobileNet. Efisiensi dalam pendeteksian objek adalah sekitar 85,18%, yang di atas rata-rata. Loss per langkah atau epoch adalah 2,73 (di bawah 3) yang mengawasi kehandalan model. Penelitian tersebut menguji model pada dataset yang disiapkan dari testbench[5].

Penelitian dengan jaringan deteksi objek kerangka kerja TensorFlow dilatih dan diuji untuk tugas pelokalan plat nomor otomatis. Pertama, dataset baru disiapkan untuk plat nomor Turki. Gambar-gambar dalam dataset diberi label dengan dua kelas yaitu mobil dan plat. Empat jaringan deteksi objek yang berbeda dikonfigurasi untuk dijalankan di lingkungan Colab Google. Konfigurasi jaringan ini adalah Single Shot MultiBox Detector (SSD) yang menggunakan fitur MobileNet dan fitur Resnet50, Faster Region Convolutional Neural Network (Faster R-CNN) menggunakan lapisan Inception untuk fitur, dan Region-based Fully Convolutional Networks (RFCN) dengan fitur Resnet101. Jaringan-jaringan ini dibandingkan untuk menentukan kinerja lokalisasi plat nomor. Berbagai jenis gambar input digunakan untuk menguji algoritma. Hasil pengujian menunjukkan bahwa solusi berbasis SSD dapat mendeteksi objek (mobil) yang lebih besar lebih baik daripada solusi berbasis R-CNN dan R-FCN yang lebih cepat. Tetapi SSD kurang baik mendeteksi beberapa plat nomor karena ukurannya yang relatif kecil. Selain itu, ditunjukkan dalam hasil pengujian bahwa plat nomor terdeteksi lebih baik daripada solusi berbasis SSD dengan Faster R-CNN dan R-FCN[6].

Berdasarkan latar belakang diatas, penelitian ini akan menggunakan TensorFlow Object Detection API dengan memanfaatkan SSD Mobilenet V2 sebagai model pra - terlatih. Hasil pelatihan data akan diterapkan dalam pembuatan aplikasi yang mampu mendeteksi dan mengukur besar akurasi objek melalui input gambar. Aplikasi yang dibuat akan mendeteksi 5 objek, yaitu Camera, Handphone, Headphone, Laptop, dan Mouse.

Citra Digital, Pemrosesan dan Deteksi Objek

Citra digital merupakan suatu matriks di mana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar atau piksel) menyatakan tingkat keabuan pada titik tersebut. Untuk sebuah citra digital, setiap piksel memiliki nilai integer yakni gray level yang menunjukkan amplitudo atau intensitas dari piksel tersebut. Citra merupakan fungsi dua dimensi yang kedua variabelnya yaitu nilai amplitudo dan koordinatnya merupakan nilai integer [7].

Machine Learning

Istilah Machine Learning pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur Samuel, Machine Learning adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas. Machine Learning dapat didefinisikan sebagai metode komputasi berdasarkan pengalaman untuk meningkatkan performa atau membuat prediksi yang akurat. Definisi pengalaman disini ialah informasi sebelumnya yang telah tersedia dan bisa dijadikan data pembelajar [8].

Deep Learning

Deep Learning (Pembelajaran Dalam) atau sering dikenal dengan istilah Pembelajaran Struktural Mendalam (Deep Structured Learning) atau Pembelajaran Hierarki (Hierarchical Learning) adalah salah satu cabang dari ilmu pembelajaran mesin (Machine Learning) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam pembelajaran dalam dapat digunakan baik untuk kebutuhan pembelajaran terarah (Supervised Learning), pembelajaran tak terarah (Unsupervised Learning) dan semi-terarah (Semi-supervised Learning) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. Deep Learning disebut sebagai Deep (dalam) karena struktur dan jumlah jaringan saraf pada algoritmanya sangat banyak bisa mencapai hingga ratusan lapisan [9].

Computer Vision

Computer Vision adalah suatu proses transformasi atau perubahan dari data yang berasal dari kamera video maupun foto/gambar kedalam sebuah hasil keputusan ataupun sebuah presentasi yang baru, dimana hasil dari kegiatan transformasi tersebut memiliki kepentingan untuk mencapai suatu tujuan. Data yang dimasukan kedalam kegiatan

transformasi tersebut memungkinkan untuk memiliki beberapa informasi yang terkonstektual seperti halnya sebuah foto/gambar yang didalamnya terdapat berbagai objek. Dengan demikian akan didapatkan keputusan-keputusan yang akan diambil pada gambar, misalnya berupa “adakah telapak tangan seseorang pada gambar tersebut?” atau “siapa sajakah orang yang terdapat pada foto tersebut?”. Adapun perubahan kedalam presentasi yang baru seperti perubahan gambar menjadi grayscale atau juga pemotongan objek pada gambar [10].

Transfer Learning

Transfer Learning adalah suatu teknik atau metode yang memanfaatkan model yang sudah dilatih terhadap suatu dataset untuk menyelesaikan permasalahan lain yang serupa dengan cara menggunakannya sebagai starting point, memodifikasi dan mengupdate parameternya sehingga sesuai dengan dataset yang baru [11].

Untuk menerapkan Transfer Learning secara efektif, perlu menjawab tiga pertanyaan utama, yaitu :

1. Apa yang Harus Ditransfer Memahami pengetahuan umum antara sumber dan tugas target. Pengetahuan apa yang dapat ditransfer dari tugas sumber ke tugas target yang akan membantu meningkatkan kinerja tugas target.
2. Kapan Mentransfer Ketika sumber dan target domain tidak terkait sama sekali, maka tidak boleh mencoba menerapkan pembelajaran transfer. Jenis transfer ini disebut Transfer Negatif. Kita harus menerapkan pembelajaran Transfer hanya jika sumber domain / tugas terkait.
3. Cara Mentransfer Mengidentifikasi berbagai teknik untuk menerapkan Transfer Learning ketika sumber dan target domain / tugas terkait. Dapat menggunakan pembelajaran transfer induktif, pembelajaran transfer transduktif atau pembelajaran transfer tanpa pengawasan [12].

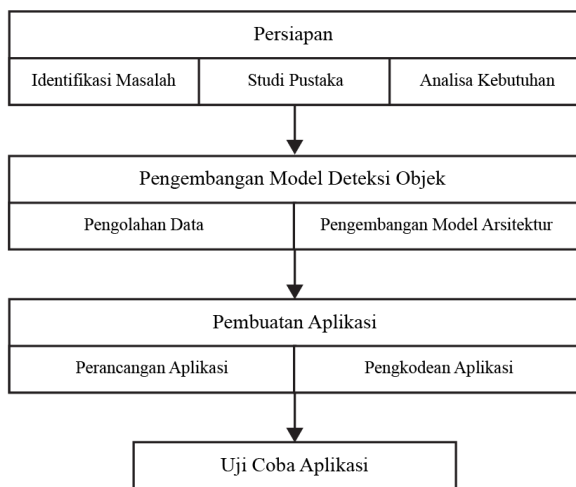
Deteksi Objek

Deteksi objek (Object Detection) adalah teknik visi komputer untuk menemukan contoh objek dalam gambar atau video. Algoritma deteksi objek biasanya memanfaatkan pembelajaran mesin atau pembelajaran mendalam untuk menghasilkan hasil yang bermakna. Ketika manusia melihat gambar atau video, manusia dapat mengenali dan menemukan objek dalam beberapa saat berbeda dengan komputer yang memerlukan komputasi yang kompleks. Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek adalah deteksi objek menempatkan keberadaan objek dalam gambar dan menggambar kotak pembatas di sekitar objek itu. Ini biasanya melibatkan dua proses, yaitu mengklasifikasikan jenis objek, dan kemudian menggambar kotak di sek-

itar objek itu. Klasifikasi gambar dan skenario deteksi objek terlihat serupa. Secara umum, klasifikasi adalah mengklasifikasikan gambar ke dalam kategori tertentu. Sedangkan objek deteksi adalah mengidentifikasi lokasi objek dalam gambar, dan misalnya menghitung jumlah instance suatu objek [13].

Metode Penelitian

Penelitian ini dibuat secara bertahap sesuai dengan tahapannya. Berikut ini adalah gambaran umum tahapan penelitian yang dilakukan pada penelitian ini, dapat dilihat pada Gambar 1.



Gambar 1: Metode Penelitian

Pada tahap pertama yang dilakukan adalah tahap persiapan. Dilakukan dengan mengidentifikasi masalah, melakukan studi pustaka yang diperlukan untuk referensi pembuatan aplikasi deteksi objek. Selanjutnya analisis kebutuhan mengenai perangkat keras dan perangkat lunak yang akan dipakai.

Pada tahap pengembangan deteksi objek, dilakukan pengembangan dalam membuat model aplikasi deteksi objek. Tahap ini berisi pengolahan data, pengembangan arsitektur, dan pelatihan data.

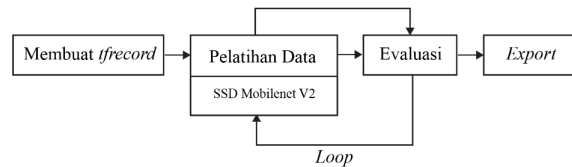
Pada tahap pembuatan aplikasi meliputi perancangan aplikasi, dan pengkodean aplikasi. Pembuatan aplikasi berupa perancangan aplikasi yang terdiri dari perancangan struktur navigasi, perancangan flowchart, dan perancangan tampilan.

Setelah diperoleh model hasil dari proses pelatihan data, model tersebut diterapkan pada aplikasi deteksi objek. Kemudian akan dilakukan uji coba aplikasi pada test set yang telah dipersiapkan sebelumnya.

Uji coba aplikasi akan dilakukan pada seluruh data test set yang berjumlah 50 gambar dengan masing – masing kelas objek berjumlah 10 gambar. Hasil dari uji coba aplikasi akan menghasilkan akurasi dari aplikasi deteksi objek.

Pengembangan Model Arsitektur

Pengembangan model arsitektur pada pembuatan aplikasi deteksi objek menggunakan TensorFlow Object Detection API. Untuk itu, berikut adalah skenario pengembangan arsitektur menggunakan TensorFlow Object Detection API dapat dilihat, pada Gambar 2.



Gambar 2: Skenario Pengembangan Arsitektur

Membuat tfrecord

Pembuatan tfrecord dilakukan pada Train Set yang telah dilakukan pengolahan data. File gambar yang telah dilakukan labeling dan menghasilkan format xml akan diubah menjadi tfrecord. Perubahan dilakukan sebelum pelatihan data. Untuk mengubah xml menjadi tfrecord akan dilakukan pada Google Colaboratory dengan menuliskan program pada notebook.

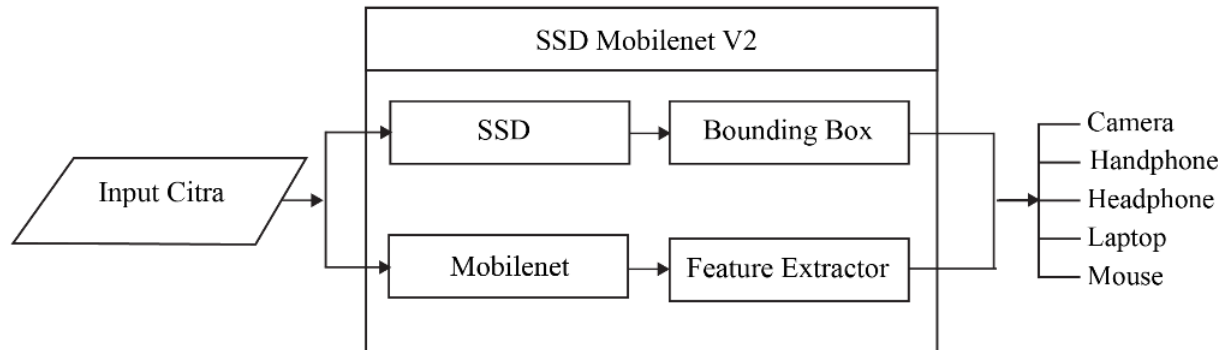
Pelatihan Data (Re – train)

Pelatihan data dilakukan dengan melakukan pelatihan kembali pada model pra – terlatih pada dataset yang telah selesai dibuat. Pelatihan dilakukan dengan melatih seluruh Train Set dengan bantuan Repository Github sebagai tempat penyimpanan dataset dan Google Colaboratory sebagai Virtual Machine selama melakukan proses pelatihan data. Skenario pelatihan data, dapat dilihat, pada Gambar 3.

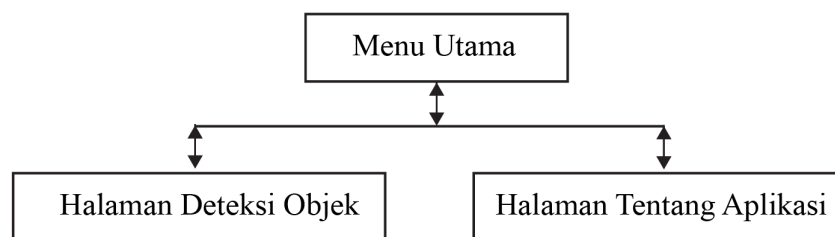
Pelatihan data dilakukan dengan menginput file tfrecord yang akan digunakan. Selain itu seluruh data yang diperlukan akan tersimpan ke dalam file repository Github. Untuk memanggil file tersebut dilakukan pemanggilan bersamaan dengan konfigurasi hyperparameter. Pada konfigurasi hyperparameter ditentukan jumlah steps untuk pelatihan data dan pemanggilan model pra – terlatih. Selanjutnya, dilakukan input file tfrecord agar dapat dilakukan pelatihan. Proses pelatihan akan memanfaatkan fasilitas GPU yang tersedia pada Google Colaboratory. Hasil pelatihan akan dapat di-export untuk diterapkan pada aplikasi. Pada penelitian kali ini, model pra – terlatih yang dipilih adalah model SSD Mobilenet V2. Model tersebut terdapat pada TensorFlow Object Detection API. Pemanggilan akan dilakukan pada model `ssd_mobilenet_v2_coco`. Berikut merupakan skenario pengembangan model secara sederhana dapat dilihat, pada Gambar 4.



Gambar 3: Pelatihan Data



Gambar 4: Skenario Penggunaan Model Pra – Terlatih



Gambar 5: Struktur Navigasi Aplikasi Deteksi Objek

SSD Mobilenet terdiri dari SSD yang berperan sebagai base model, dan Mobilenet sebagai Network Model. SSD akan mengatur pendeteksian objek dengan membuat Bonding Box. Mobilenet akan bekerja untuk mengekstrak fitur yang akan nantinya diklasifikasi. Penggabungan SSD dan Mobilenet akan membantu dalam proses pembuatan aplikasi deteksi objek. Dalam aplikasi deteksi objek dibutuhkan SSD untuk membuat lokalisasi gambar untuk menentukan posisi objek. Sedangkan Mobilenet akan dibutuhkan untuk membantu mengklasifikasi objek yang terdapat dalam suatu gambar. Klasifikasi akan menghasilkan kategori untuk masing – masing objek yaitu, Camera, Handphone, Headphone, Laptop, dan Mouse.

Evaluasi

Evaluasi model dilakukan pada data Validation Set selama proses pengembangan model. Evaluasi bertujuan untuk menemukan parameter yang tepat. Sehingga, tahap ini dilakukan secara berulang hingga menemukan parameter yang tepat dalam proses pembuatan model. Evaluasi akan dilakukan pada 100 gambar pada Validation Set.

Export

Selanjutnya, apabila parameter yang tepat telah ditemukan dan model telah dapat dibuat, maka model tersebut akan di-export dan disimpan menjadi file protobuf (.pb) dan label map (.pbtxt) yang nantinya akan digunakan pada pembuatan aplikasi deteksi objek.

Perancangan Struktur Navigasi Aplikasi

Perancangan struktur navigasi bertujuan untuk menjelaskan alur dari suatu aplikasi. Pada aplikasi deteksi objek, dibuat struktur navigasi hierarki. Struktur navigasi aplikasi deteksi objek dapat dilihat, seperti pada Gambar 5.

Perancangan Tampilan

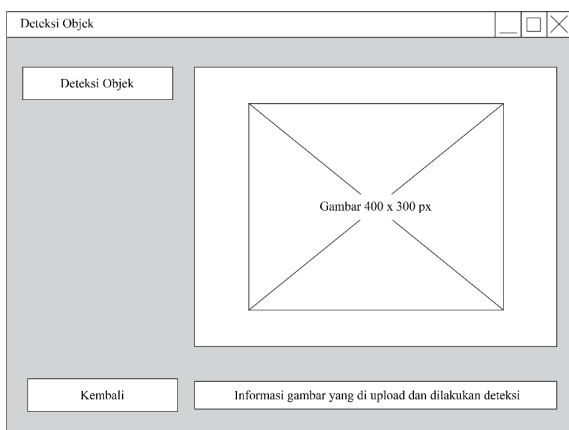
Perancangan tampilan pada aplikasi pendeteksian objek yang terdiri dari tampilan tampilan menu utama, tampilan halaman deteksi objek, dan tampilan halaman tentang aplikasi. Perancangan tampilan halaman dilakukan agar hasil output dari aplikasi memiliki tampilan yang diharapkan. Tampi-

lan rancangan halaman menu utama aplikasi dapat dilihat, seperti pada Gambar 6.



Gambar 6: Rancangan Tampilan Halaman Menu Utama

Tampilan rancangan halaman Deteksi Objek dapat dilihat, seperti pada Gambar 7.



Gambar 7: Rancangan Tampilan Halaman Deteksi Objek

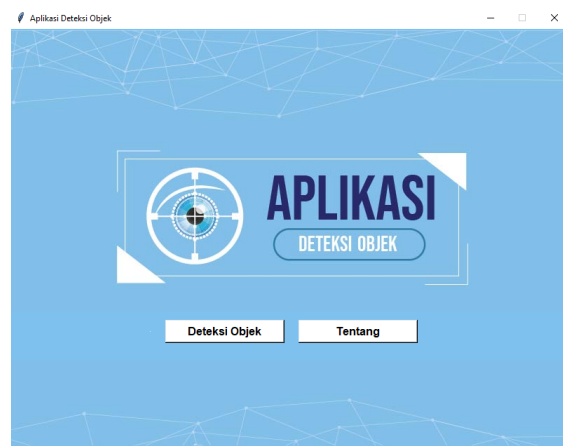
Tampilan rancangan halaman Tentang Aplikasi dapat dilihat, seperti pada Gambar 8.



Gambar 8: Rancangan Tampilan Halaman Tentang Aplikasi

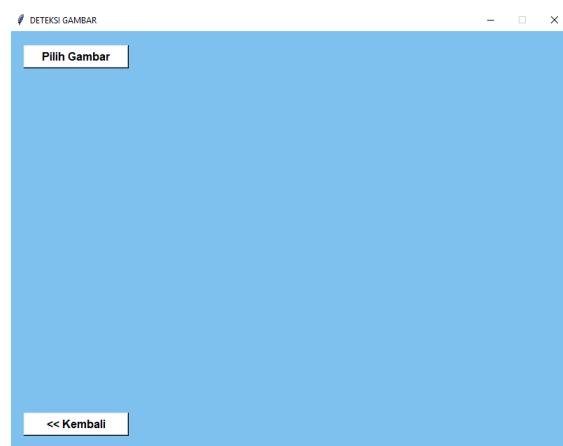
Hasil dan Pembahasan

Setelah dilakukan perancangan, kemudian dilakukan proses pembuatan aplikasi hingga selesai. Untuk menjalankan aplikasi ini, adalah dengan membuka Python IDLE, kemudian menuju direktori menyimpan file python aplikasi deteksi objek. Aplikasi dapat dijalankan dengan menekan tombol F5 pada keyboard. Uji coba aplikasi dilakukan dengan menampilkan halaman yang telah dirancang sebelumnya. Pada aplikasi deteksi objek terdapat 3 tampilan utama, yaitu halaman Menu Utama, halaman Deteksi Objek, dan halaman Tentang Aplikasi. Berikut halaman Menu Utama Aplikasi dapat dilihat pada Gambar 9.



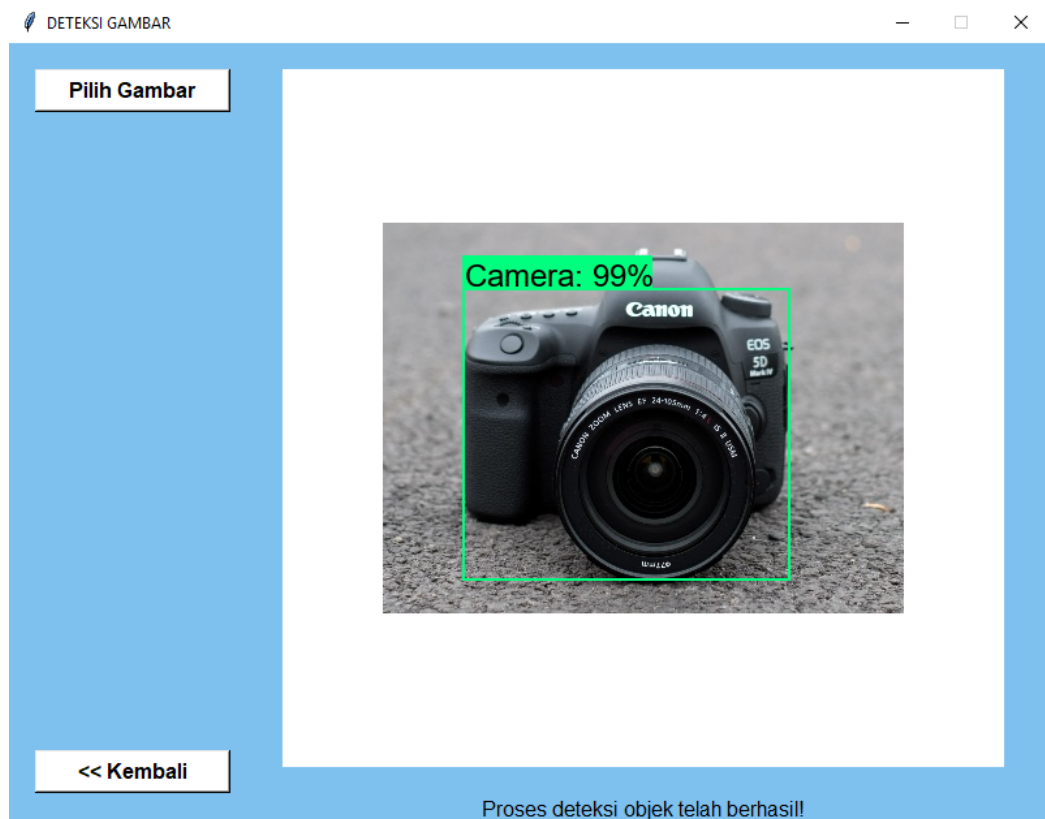
Gambar 9: Tampilan Halaman Menu Utama

Selanjutnya adalah halaman Deteksi Objek, halaman ini dapat digunakan untuk melakukan deteksi objek. Berikut merupakan halaman Deteksi Objek dapat dilihat, pada Gambar 10.



Gambar 10: Tampilan Halaman Deteksi Objek

Untuk dapat mendeteksi objek, terdapat button Upload Gambar untuk memilih gambar yang ingin dilakukan pendeteksian. Apabila objek dapat terdeteksi akan dapat dilihat pada area yang telah tersedia. Berikut merupakan tampilan gambar yang berhasil terdeteksi dapat dilihat, pada Gambar 11.



















































Gambar 11: Gambar yang Berhasil Dilakukan Deteksi



Gambar 12: Tampilan Halaman Tentang Aplikasi

Tabel 1: Hasil Pengujian Aplikasi Deteksi Objek

No.	Nama Objek	Hasil Deteksi	Keterangan	Benar/Salah	Akurasi (%)	Rata – rata akurasi (%)
1.	Camera		Terdeteksi	Benar	99	99
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
2.	Handphone		Terdeteksi	Benar	99	89.1
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Tidak Terdeteksi	-	-	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
3.	Headphone		Terdeteksi	Benar	99	89.1
			Terdeteksi	Salah	-	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
4.	Laptop		Tidak Terdeteksi	-	-	89.1
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
5.	Mouse		Terdeteksi	Benar	99	98.8
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	97	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
			Terdeteksi	Benar	99	
	Rata – rata Akurasi Keberhasilan Pendeteksian					93.02

Terakhir, adalah halaman Tentang Aplikasi yang memuat informasi mengenai aplikasi deteksi objek. Berikut merupakan halaman Tentang Aplikasi dapat dilihat, pada Gambar 12.

Setelah aplikasi berhasil dijalankan, dilakukan pengujian dengan melakukan uji coba pada data Test Set, yang telah dipersiapkan sebelumnya. Terdapat 50 gambar yang akan dilakukan untuk menguji aplikasi deteksi objek. Terdapat 5 kelas dengan masing-masing kelas berjumlah 10 gambar. Hasil pengujian objek, dapat dilihat pada Tabel 1.

Penutup

Berdasarkan penelitian yang telah dilakukan, berhasil dibuat aplikasi untuk mendeteksi objek dengan menggunakan TensorFlow Object Detection API dengan memanfaatkan SSD Mobilenet V2 sebagai model pra – terlatih. Program dapat mendeteksi 5 kategori kelas objek, yaitu Camera, Handphone, Headphone, Laptop, dan Mouse. Aplikasi dapat menampilkan tingkat pengukuran akurasi masing-masing objek. Hasil pengujian yang dilakukan pada 50 test set dengan jumlah masing – masing kelas objek adalah 10 gambar. Gambar objek Camera mendapatkan rata – rata persentase sebesar 99%, Handphone sebesar 89.1%, Headphone sebesar 89.1%, Laptop sebesar 89.1%, dan Mouse sebesar 98.8%. sehingga diperoleh rata – rata akurasi keberhasilan pendeteksian aplikasi deteksi objek adalah 93.02%.

Aplikasi ini dapat dikembangkan dengan menggunakan metode lain yang serupa. Pengembangan diharapkan dapat membuat aplikasi dapat mendeteksi lebih dari satu objek dalam satu gambar. Selain itu, dapat dilakukan penambahan fitur real time image processing dan pengembangan aplikasi yang dapat dijalankan pada berbagai platform seperti pada android maupun website.

Daftar Pustaka

- [1] Rusman, “Model-model Pembelajaran Mengembangkan Profesionalisme Guru Edisi Kedua”, Jakarta: PT RajaGrafindo Persada, 2012.
- [2] A.M. Taqi , et al., “Skin Lesion Detection by Android Camera based on SSD- Mobilenet and TensorFlow Object Detection API”, American Journal of Advanced Research, 2019.
- [3] F. Al-Azzo , A.M. Taqi dan M. Milanova , “Human Related-Health Actions Detection using Android Camera based on TensorFlow Object Detection API”, International Journal of Advanced Computer Science and Applications Vol.9 No. 10, 2018.
- [4] T.V. Janahiraman and M.S.M. Subuhan, “Traffic Light Detection Using Tensorflow Object Detection Framework”, International Conference on System Engineering and Technology (ICSET), 2019.
- [5] S. Howal , A. Jadhav and C. Arthshi, “Object Detection for Autonomous Vehicle Using TensorFlow”, International Conference on Intelligent Computing, Information and Control Systems, 2019.
- [6] M. Peker, “Comparison of Tensorflow Object Detection Networks for Licence Plate Localization”, Global Power, Energy and Communication Conference (IEEE GPECOM2019), 2019.
- [7] A.F. Fikriya, M.I. Irawan dan Soetrisno, “Implementasi Extreme Learning Machine untuk Pengenalan Objek Citra Digital”, Jurnal Sains dan Seni ITS. Vol 6, No.1, 2017.
- [8] J. Pujoseno, “Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Alat Tulis. Studi Kasus : Gambar alat tulis (Ballpoint, Penghapus, dan Penggaris)”, Skripsi, UII, Yogyakarta, 2018.
- [9] W. Dadang, “Memahami Kecerdasan Buatan berupa Deep Learning dan Machine Learning”, diakses daring pada <https://warstek.com/2018/02/06/deepmachinelearning/>, 2018.
- [10] R.A Safitri, “Implementasi Metode Viola and Jones untuk Mengenali Isyarat Jari Sebagai Sarana Navigasi Pada Aplikasi Pemutar Musik”, Skripsi, Universitas Komputer Indonesia, Bandung, 2015.
- [11] S. Sena, “Pengenalan Deep Learning Part 8 : Gender Classification using Pre – Trained Network (Transfer Learning)”, diakses daring https://medium.com/@sa_muelsena/pengenalan-deep-learning-part-8-gender-classification-using-pre-trained-network-transfer-37ac910500d1, 2018.
- [12] R. Khandelwal, “Deep Learning Using Transfer Learning”, diakses daring pada <https://medium.com/datadriveninvestor/deep-learning-using-transfer-learning-cfbce1578659>, 2019.
- [13] A. Ouaknine, “Review of Deep Learning Algorithms for Object Detection” , diakses daring pada <https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-clf3d437b852>, 2019.

Halaman ini sengaja dikosongkan.