

Karakteristik Metode Mobilenet-SSD Dengan *Pre-Trained* Model Mobilenet Untuk Objek Bergerak

Khalifah Falah¹, Milda Gustiana H², Ung Ungkawa³

^{1,2,3} Institut Teknologi Nasional

Email: khalifahfalaih@mhs.itenas.ac.id

Received DD MM YYYY | Revised DD MM YYYY | Accepted DD MM YYYY

ABSTRAK

Pada kemampuan yang dimiliki oleh manusia dalam mengenali sebuah objek dapat dikatakan luar biasa. Kemajuan teknologi saat ini tidak dapat dibendung seiring perkembangan penerapannya dalam kehidupan sehari-hari. Salah satunya adalah pengolahan citra di bidang teknologi computer vision. pada bidang computer vision memanfaatkan teknologi deep learning untuk mendeteksi sebuah objek. pada penelitian ini arsitektur CNN yang digunakan untuk mengenali sebuah objek bergerak adalah arsitektur SSD dengan menggunakan pre-trained model Mobilenet yang dilakukan analisis untuk mendeteksi objek dari input citra dua dimensi dan menggunakan memasukan video capture. Sistem melakukan deteksi objek terhadap 20 kelas objek dari tiga sisi yaitu depan, samping, dan belakang berdasarkan pre-trained model MobileNet. Hasil Pengujian dengan input citra dua dimensi memperoleh nilai rata-rata pada setiap kelas adalah sisi depan sebesar 91,5%, sisi samping sebesar 92,3%, sisi belakang sebesar 85,5%. Sedangkan hasil pengujian dengan input video capture memperoleh nilai rata-rata sebesar 72,8%.

Kata kunci: Pengenalan Objek, CNN, SSD, Mobile Net, Deteksi Objek

ABSTRACT

The ability possessed by humans to recognize an object can be said to be extraordinary. Advances in technology today can not be damned along with the development of its application in everyday life. One of them is image processing in the field of computer vision technology. in the field of computer vision utilizing deep learning technology to detect an object. In this study, the CNN architecture used to recognize a moving object is the SSD architecture using pre-trained Mobilenet models versions 1 which are analyzed to detect objects from two-dimensional image input and video capture input. The system detects 20 object classes from three sides, namely front, side, and back based on the pre-trained MobileNet model. Test results with two-dimensional image input obtained the average value in each class is the front side of 91.5%, the side of 92.3%, the back side of 85.5%. While the test results with video capture input obtained an average value of 72.8%.

Keywords: Object Identification, CNN, SSD, MobileNet, Object Detection

1. PENDAHULUAN

Kemajuan teknologi saat ini tidak dapat dibendung seiring perkembangan penerapannya dalam kehidupan sehari-hari. Salah satunya adalah pengolahan citra di bidang teknologi *computer vision*. Pemanfaatan *computer vision* ini sendiri pada pengolahan citra sudah banyak diterapkan antara lain adalah pengenalan wajah, jenis kendaraan, hingga ekstraksi informasi kompleks seperti pada *Google vision* (Popescu et al., 2017).

Selain itu, pada bidang *computer vision* memanfaatkan teknologi *deep learning* untuk mendeteksi sebuah objek. Teknologi tersebut merupakan salah satu bidang dari *Machine Learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. *Deep learning* memiliki dua metode yang paling populer digunakan yaitu *Convolutional Neural Network (CNN)* dan *Recurrent Neural Network (RNN)*. Perkembangan arsitektur *deep learning* mulai mengarah kepada arsitektur yang dapat mendeteksi lebih cepat serta sistem deteksi yang lebih efisien. Arsitektur yang memperhatikan hal tersebut diantaranya *Single Shot Multibox Detector (SSD)* dan *You Only Look Once (YOLO)* (Liu, W., 2015).

Berdasarkan penelitian yang sebelumnya dilakukan (F. E. Ramadhan., 2020) menguji pengaturan *neural network* pada *client-side* seperti *epoch* dan *learning rate* yang optimal, pengujian tersebut dilakukan penerapan *image classification* dengan *pre-trained* model *MobileNet*. Hasil dari penelitian adalah membuktikan bahwa *client-side machine learning* jauh lebih cepat pada saat *training time*, sementara *server-side machine learning* lebih unggul nilai *loss function* yang dihasilkannya.

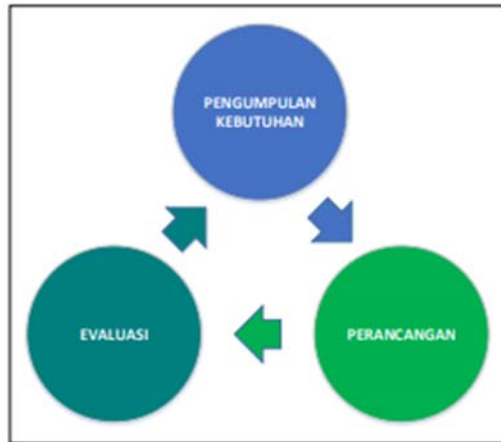
Pada penelitian lain sebelumnya dilakukan (F. Sindy., 2019) mendeteksi objek manusia secara *real time* menggunakan teknik *deep learning* dengan *MobileNet-SSD* sebagai model arsitektur jaringannya. Uji coba penelitian dilakukan dengan kondisi yang bervariasi dengan intensitas cahaya 50-550 *lux* dan jarak terhadap objek 1 – 10 meter. Didapatkan tingkat keakuratan sistem mampu mencapai 91,67%.

Berdasarkan kedua penelitian tersebut maka diambil *MobileNet-SSD* sebagai model arsitektur jaringannya berdasarkan penelitian yang dilakukan (F. Sindy., 2019) apakah metode *MobileNet-SSD* dapat mendeteksi objek secara *real time* menggunakan *pre-trained* model dari penelitian (F. E. Ramadhan., 2020) yaitu *pre-trained* model *MobileNet*.

Oleh karena itu, berdasarkan latar belakang masalah maka pada penelitian ini arsitektur CNN yang digunakan adalah arsitektur *MobileNet-SSD* dengan menggunakan *pre-trained* model *MobileNet versi 1*. Sehingga penelitian ini bertujuan untuk mengukur performa arsitektur yang dipakai dengan *pre-trained* model *MobileNet* untuk mendeteksi objek bergerak.

2. METODE PENELITIAN

Tahapan pengembangan sistem mengimplementasikan metode *prototype*, metode ini dipilih agar sistem dapat dibuat secara terstruktur yang didalamnya memiliki beberapa tahapan yaitu pengumpulan kebutuhan, perancangan, dan evaluasi, seperti yang ditunjukkan pada Gambar 1.



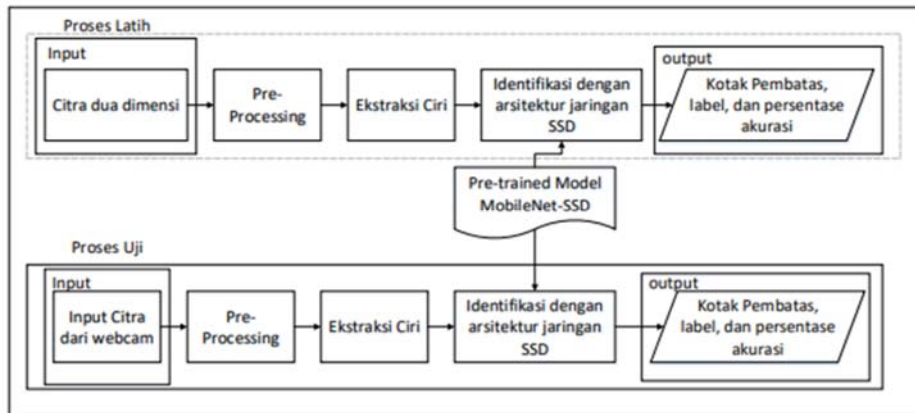
Gambar 1. Model Pengembangan Sistem Adaptasi dari Prototype

2.1. Identifikasi dan Analisa Kebutuhan

Perancangan sistem dilakukan dengan melakukan analisa kebutuhan untuk mendukung cara proses dan kinerja sistem berjalan dengan lancar. Kebutuhan secara keseluruhan dalam melaksanakan perancangan sistem terbagi menjadi dua yaitu kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*Software*).

2.2. Perancangan Umum

Pada bagian ini dijelaskan tentang pembangunan sistem secara keseluruhan dengan sederhana yang ditunjukkan pada Gambar 2.



Gambar 2. Blok Diagram Umum Sistem

Sistem yang dibangun adalah identifikasi objek bergerak terdiri dari beberapa langkah yaitu proses latih dan proses uji. Proses latih dilakukan dengan melakukan pelatihan kembali pada *pre-trained* model *MobileNet*; pelatihan dilakukan dengan melatih seluruh *train set* dengan menggunakan aplikasi *web framework* yang dibangun menggunakan *Flask*. *Flask* merupakan *web framework* yang ditulis menggunakan bahasa *Python* dan termasuk *micro framework* karena tidak memerlukan suatu alat atau pustaka dalam penggunaannya (M. Singh et al., 2019). Proses latih bertujuan untuk mencari data latih pada penelitian dengan menggunakan input citra dua dimensi yang didapat dari *Google Image Search*. selanjutnya adalah *pre-processing* adalah proses pengubahan ukuran citra yaitu 300x300. Lalu, citra masukan proses ekstraksi ciri menggunakan arsitektur *MobileNet* sebagai *feature extractor*. Hasil dari ekstraksi ciri

dilakukan klasifikasi dari *pre-trained* model *MobileNet* dan identifikasi dengan arsitektur jaringan SSD untuk dilakukan prediksi objek sehingga didapatkan *output* berupa kotak pembatas, label, dan persentase akurasi dari objek yang terdeteksi.

2.3. Pengumpulan Citra Latih

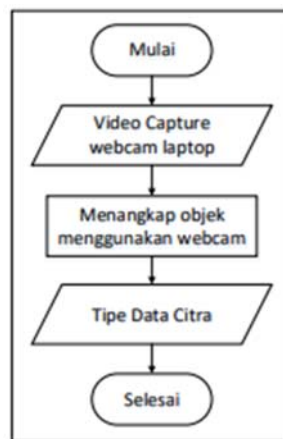
Pengumpulan citra latih dilakukan dengan mengumpulkan citra dua dimensi yang didapat dari *Google Image Search*, citra dua dimensi tersebut disimpan ke dalam satu direktori yang selanjutnya diproses untuk pencarian data latih citra menggunakan aplikasi *web framework* yang sudah dibuat sebelumnya. Citra dua dimensi yang dimaksud adalah citra yang di dalamnya terdapat objek dari semua yang ada pada *pre-trained* model *MobileNet*. Berikut adalah daftar objeknya ditunjukkan pada Tabel 3.

Tabel 1. Daftar Objek Citra Latih

No	Kelas	No	Kelas
1	Pesawat Terbang	11	Meja Makan
2	Sepeda	12	Anjing
3	Burung	13	Kuda
4	Perahu	14	Motor
5	Botol	15	Orang
6	Bus	16	Pot Bunga
7	Mobil	17	Domba
8	Kucing	18	Sofa
9	Kursi	19	Kereta
10	Sapi	20	Monitor

2.4. Pengambilan Citra Uji

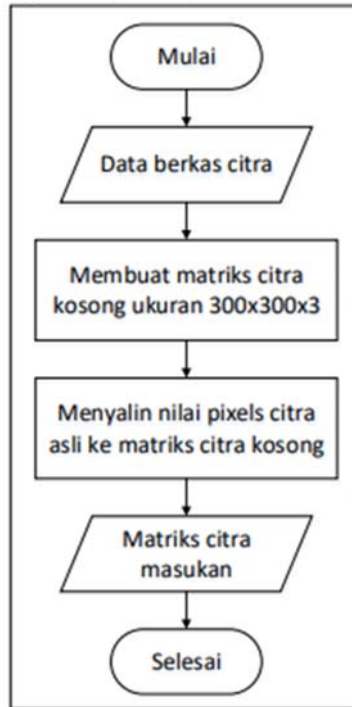
Pengambilan citra uji dilakukan menggunakan *webcam* laptop, pada keadaan *webcam* laptop yang sudah diaktifkan, sistem mengambil setiap objek yang tertangkap secara kontinu dan mengubahnya pada tipe data citra yaitu berisi nilai matriks citra yang digunakan untuk proses *pre-processing* yang dijelaskan pada subbab 2.5. berikut adalah diagram alir pengambilan citra uji ditunjukkan pada Gambar 3.



Gambar 3. Diagram Alir Pengambilan Citra Uji

2.5.Pre-Processing

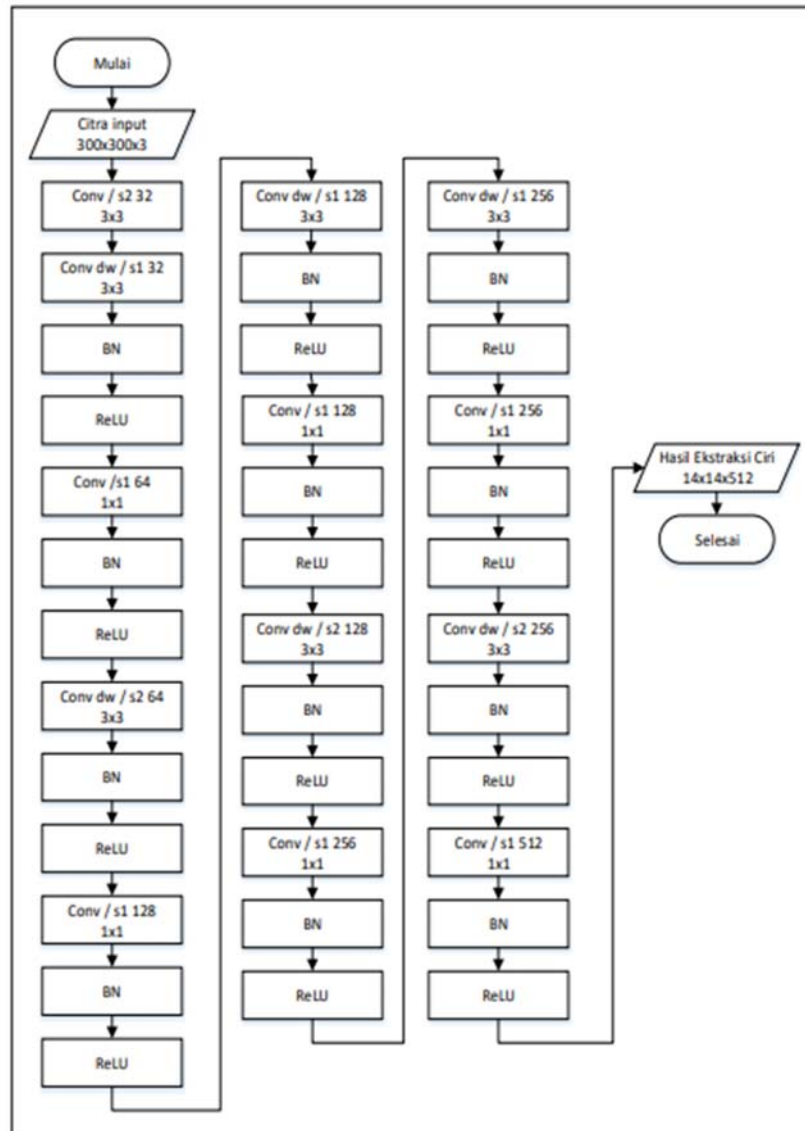
Pre-processing dilakukan dengan menggunakan masukan berupa data citra yang didapat dari pengambilan citra latih dan citra uji, tahapan ini bertujuan untuk menyesuaikan ukuran citra dengan arsitektur jaringan MobileNet-SSD. Tipe data berkas citra yang menyimpan data citra diubah menjadi matriks dimensi dengan jumlah *channel*/citra, pada implementasi ini digunakan citra dengan *channel RGB (Red, Green, Blue)*. Terlebih dahulu program membuat matriks citra kosong berukuran $300 \times 300 \times 3$, Nilai per *pixels* citra masukan asli ditempelkan pada array matriks sesuai dengan lokasi indeks masing-masing. Setelah itu didapatkan berkas masukan citra yang sudah sesuai dengan ukuran input citra pada arsitektur jaringan *MobileNet-SSD*. Berikut diagram alir *pre-processing* ditunjukkan pada Gambar 4.



Gambar 4. Diagram Alir *Pre-processing*

2.6.Ekstraksi Ciri

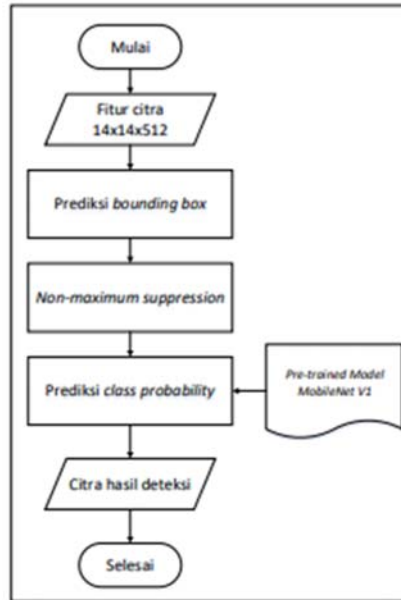
Arsitektur *MobileNet* digunakan sebagai *feature extractor*, pada tahapan ini citra masukan akan melewati lapisan-lapisan konvolusi sebanyak n filter. Konvolusi pada arsitektur *MobileNet* dibagi menjadi dua yaitu *depthwise* dan *pointwise convolution*, *depthwise convolution* adalah lapisan konvolusi dengan ukuran kernel 3×3 dengan jumlah perpindahan atau *stride* setiap kalkulasi konvolusi sebanyak 2 *pixels* dan 1 *pixel*. Sedangkan *pointwise convolution* memiliki ukuran kernel 1×1 dengan jumlah perpindahan atau *stride* 1 *pixel* (Howard et al., 2017). Arsitektur *MobileNet* memanfaatkan *Batch Normalization (BN)* dan *Rectified Linear Unit (ReLU)*, *BN* digunakan untuk menormalkan kontribusi ke lapisan pada setiap kelompok mini, *ReLU* merupakan lapisan aktivasi CNN dengan mengaplikasikan fungsi $f(x) = \max(0, x)$, apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. Hasil akhir dari proses ini adalah fitur citra berukuran 14×14 sebanyak 512 filter. Berikut adalah diagram alir proses ekstraksi ciri ditunjukkan pada Gambar 5.



Gambar 5. Diagram Alir Ekstraksi Ciri

2.7. Deteksi Objek

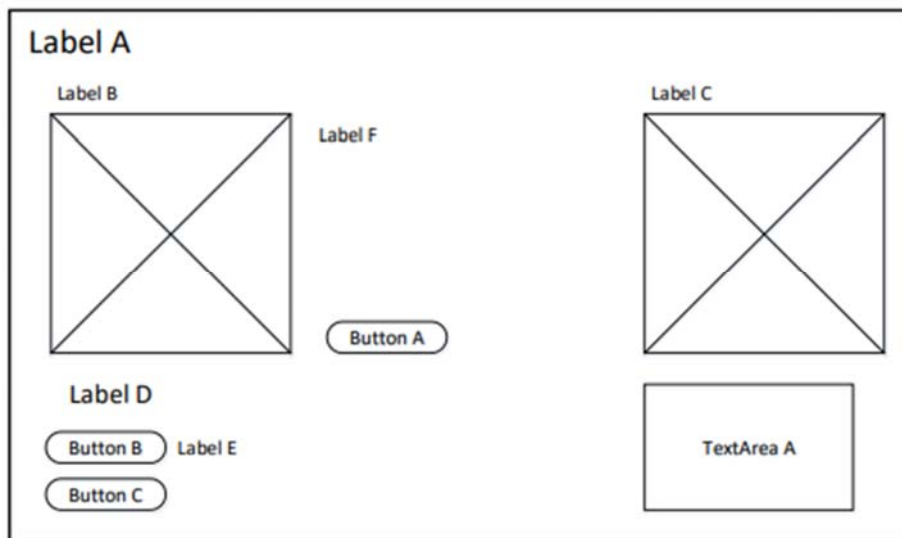
Pada tahapan deteksi objek dilakukan klasifikasi dan lokalisasi secara bersamaan. Fitur citra hasil ekstraksi ciri dilakukan prediksi objek terhadap citra, deteksi objek dilakukan menggunakan *pre-trained* model *MobileNet*, *pre-trained* akan dapat mengenali jenis objek yang ada pada citra tersebut. Deteksi objek dilakukan pada setiap *input* citra berukuran 300x300 dan dilakukan ekstraksi ciri dengan melewati 13 lapisan konvolusi sehingga didapat hasil akhir adalah 14x14. Prediksi *bounding box* dilakukan oleh arsitektur SSD yang prediksinya bisa mencapai 8732 prediksi, dari banyaknya prediksi tersebut dilakukan metode *non-maximum suppression* bertujuan untuk menghilangkan hasil prediksi yang memiliki nilai kebenaran rendah sesuai nilai *threshold* yang diterapkan yaitu 0.45. Kemudian nilai-nilai yang didapat dari prediksi adalah koordinat *bounding box*, nilai *confidence bounding box*, *class object* dan nilai *confidence class object*. Hasil akhir dari deteksi objek adalah citra masukan yang memiliki hasil deteksi dengan *bounding box*, *class confidence*, dan *class object*. Berikut adalah diagram alir deteksi objek yang ditunjukkan pada Gambar 6.



Gambar 6. Diagram Alir Deteksi Objek

2.8. Membangun Prototype

Membangun *prototype* merupakan tahapan yang memuat perancangan sementara yang menjadi dasar penelitian dan dilakukan pengerjaannya sampai tahap implementasi dan pengujian sistem. Salah satunya adalah membuat perancangan antar muka yang digunakan untuk mencari data latih penelitian dengan masukan citra dua dimensi yang dilakukan deteksi objek terhadap citra. berikut adalah perancangan antar muka yang ditunjukkan pada Gambar 7.

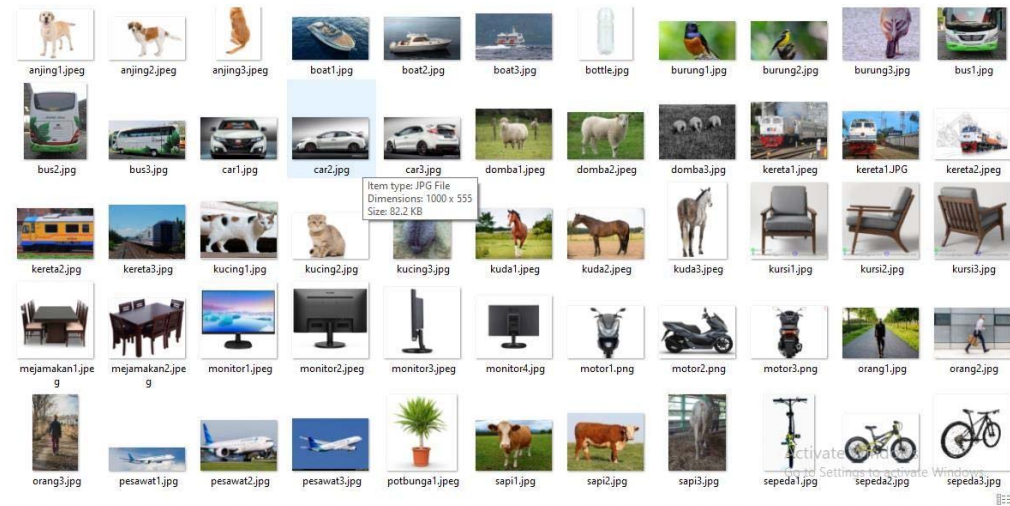


Gambar 7. Perancangan Antar Muka

3. HASIL DAN PEMBAHASAN

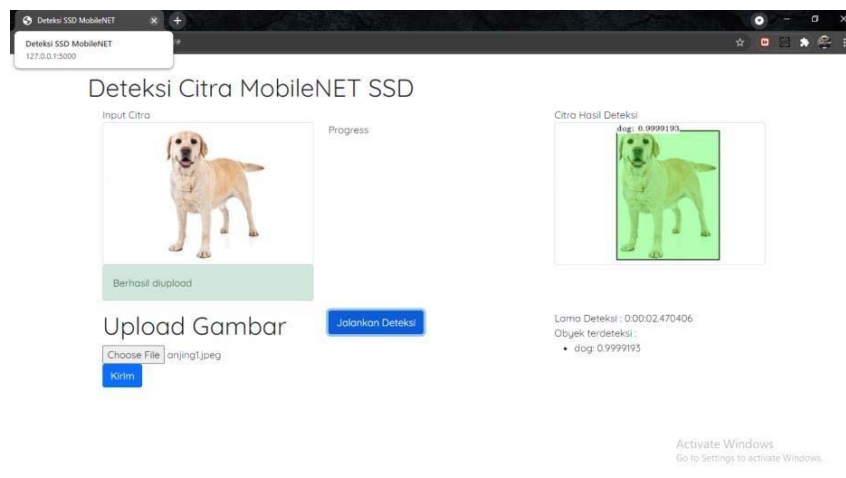
3.1. Pengujian Input Citra

Pada pengujian dilakukan menggunakan *input* citra dua dimensi dari *Google Image Search*, didapatkan 58 citra dua dimensi dan disimpan pada satu direktori, citra dua dimensi tersebut adalah citra yang di dalamnya terdapat objek, setiap objek yang memiliki sisi berbeda dilakukan pengujian pada setiap sisinya, yaitu sisi depan, samping, dan belakang. Berikut kumpulan citra dua dimensi yang didapat ditunjukkan pada Gambar 8.



Gambar 8. Kumpulan Citra Dua Dimensi

Setiap citra dua dimensi tersebut dilakukan deteksi objek menggunakan aplikasi *web framework* yang telah dibangun. Berikut adalah aplikasi *web framework* ditunjukkan pada Gambar 9.

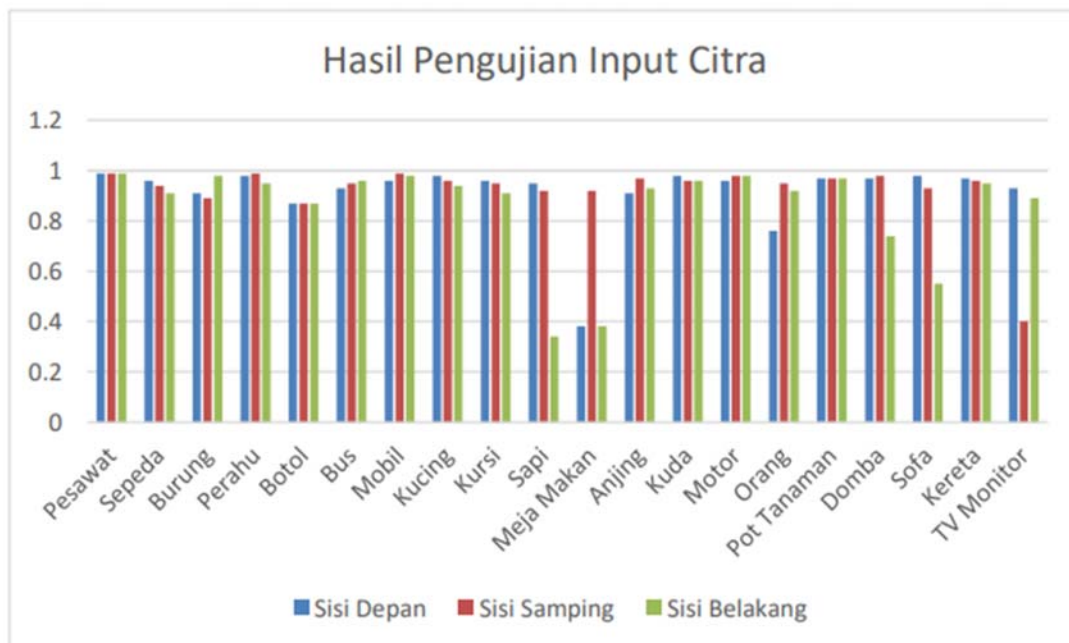


Gambar 9. Aplikasi *Web Framework* Deteksi Objek

Pengujian dilakukan pada 20 kelas sesuai dengan *pre-trained* model *MobileNet-SSD*, hasil deteksi objek pada setiap kelas ditunjukkan pada Tabel 2 dan Gambar 10.

Tabel 2. Hasil Pengujian Input Citra

Objek	Sisi			Rata-rata Waktu (detik)
	Depan	Samping	Belakang	
Pesawat	0.99	0.99	0.99	0.22456
Sepeda	0.96	0.94	0.91	0.21103
Burung	0.91	0.89	0.98	0.39932
Perahu	0.98	0.99	0.95	0.24638
Botol	0.87	0.87	0.87	0.30332
Bus	0.93	0.95	0.96	0.25784
Mobil	0.96	0.99	0.98	0.21236
Kucing	0.98	0.96	0.94	0.28642
Kursi	0.96	0.95	0.91	0.26728
Sapi	0.95	0.92	0.34	0.38632
Meja Makan	0.38	0.92	0.38	0.44298
Anjing	0.91	0.97	0.93	0.20032
Kuda	0.98	0.96	0.96	0.23467
Motor	0.96	0.98	0.98	0.21102
Orang	0.76	0.95	0.92	0.33618
Pot Tanaman	0.97	0.97	0.97	0.23962
Domba	0.97	0.98	0.74	0.36374
Sofa	0.98	0.93	0.55	0.41327
Kereta	0.97	0.96	0.95	0.23281
TV Monitor	0.93	0.4	0.89	0.34193



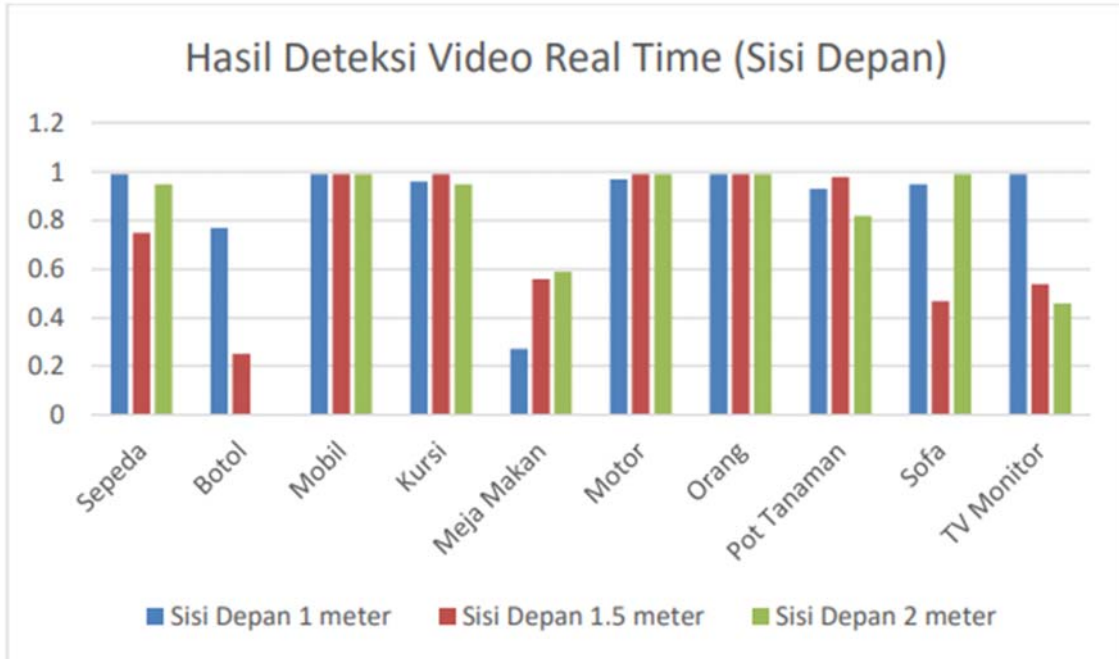
Gambar 10. Grafik Hasil Pengujian Input Citra

3.1. Pengujian Input Video *Real Time*

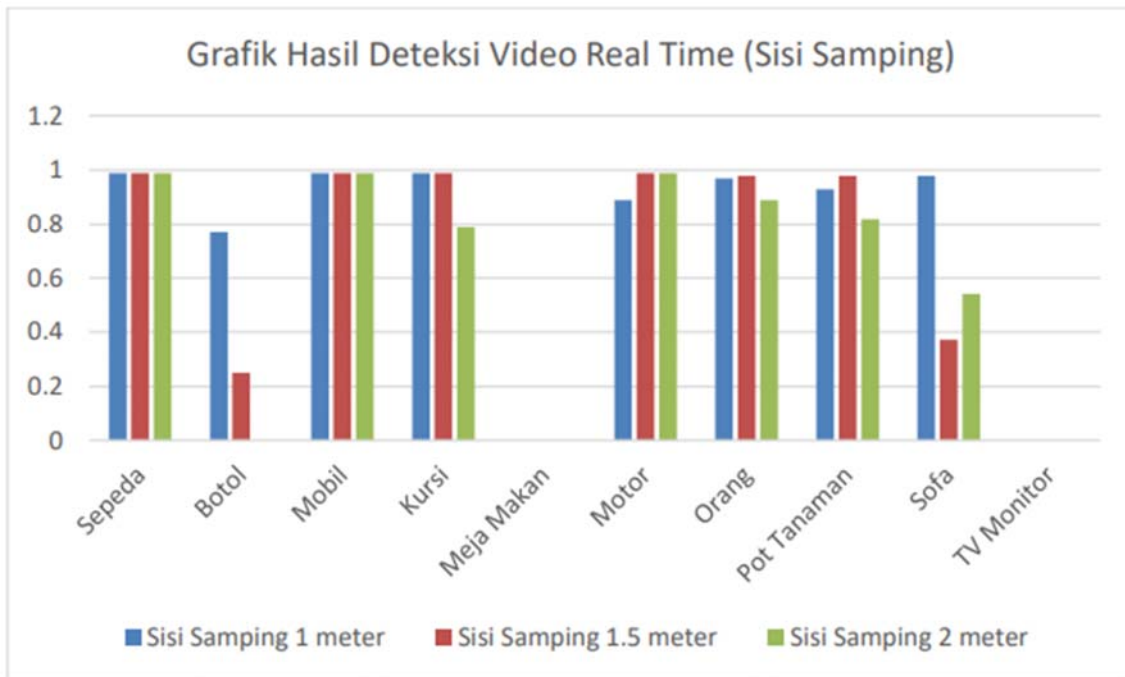
Pengujian input video dilakukan dengan menggunakan *webcam* laptop dengan beberapa objek yang dapat ditemukan pada sekitar lokasi pengujian sesuai dengan *pre-trained* model *MobileNet-SSD*. Berikut adalah hasil pengujian dari *input video real time* ditunjukkan pada Tabel 3, Gambar 11, Gambar 12, dan Gambar 13.

Tabel 3. Hasil Pengujian Input Video *Real Time*

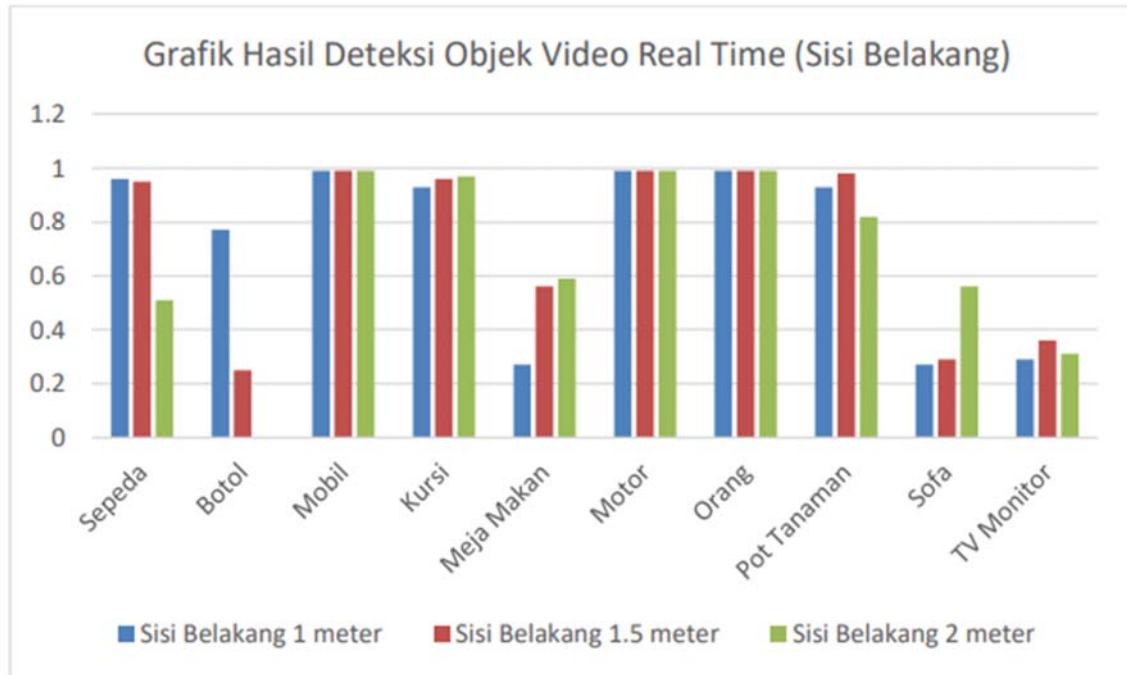
Objek	Sisi								
	Depan			Samping			Belakang		
	1 m	1.5 m	2 m	1 m	1.5 m	2 m	1 m	1.5 m	2 m
Pesawat	-	-	-	-	-	-	-	-	-
Sepeda	0.99	0.75	0.95	0.99	0.99	0.99	0.96	0.95	0.51
Burung	-	-	-	-	-	-	-	-	-
Perahu	-	-	-	-	-	-	-	-	-
Botol	0.77	0.25	0	0.77	0.25	0	0.77	0.25	0
Bus	-	-	-	-	-	-	-	-	-
Mobil	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Kucing	-	-	-	-	-	-	-	-	-
Kursi	0.96	0.99	0.95	0.99	0.99	0.79	0.93	0.96	0.97
Sapi	-	-	-	-	-	-	-	-	-
Meja Makan	0.27	0.56	0.59	0	0	0	0.27	0.56	0.59
Anjing	-	-	-	-	-	-	-	-	-
Kuda	-	-	-	-	-	-	-	-	-
Motor	0.97	0.99	0.99	0.89	0.99	0.99	0.99	0.99	0.99
Orang	0.99	0.99	0.99	0.97	0.98	0.89	0.99	0.99	0.99
Pot Tanaman	0.93	0.98	0.82	0.93	0.98	0.82	0.93	0.98	0.82
Domba	-	-	-	-	-	-	-	-	-
Sofa	0.95	0.47	0.99	0.98	0.37	0.54	0.27	0.29	0.56
Kereta	-	-	-	-	-	-	-	-	-
TV Monitor	0.99	0.54	0.46	0	0	0	0.29	0.36	0.31



Gambar 11. Grafik Hasil Deteksi Video Real Time Sisi Depan



Gambar 12. Grafik Hasil Deteksi Video Real Time Sisi Samping



Gambar 13. Grafik Hasil Deteksi Video Real Time Sisi Belakang

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan yaitu karakteristik metode *MobileNet-SSD* menggunakan *pre-trained* model *MobileNet* dapat diambil beberapa kesimpulan. Pada pengujian yang dilakukan dengan input citra dua dimensi, objek pesawat mendapat hasil deteksi yang tinggi dari sisi depan, samping maupun belakang dengan persentase mencapai 99%, dan objek meja makan mendapat nilai prediksi yang rendah dari sisi depan dan belakang dengan persentase adalah 38%.

Pengujian menggunakan input video real time tidak dilakukan pada setiap kelas objek yang ada pada *pre-trained* model *MobileNet* karena tidak terdapat objek-objek tersebut di sekitar lokasi ketika pengujian berlangsung. Terdapat beberapa objek yang tidak dapat terdeteksi pada sisi samping objek diantaranya adalah meja makan dan monitor. Jarak antara kamera dan objek mempengaruhi hasil deteksi pada objek yang berukuran besar pada jarak 1 meter nilai hasil deteksi di bawah 50%, tetapi pada jarak 2 meter nilai hasil deteksi di atas 50%, sebaliknya pada hasil deteksi objek yang berukuran kecil.

DAFTAR PUSTAKA

- Popescu, D. A., Zilberman, N., Moore, A. W. (2017). *Characterizing the impact of network latency on cloud-based applications performance*. University of Cambridge, United Kingdom.
- Ramadhan, F. E. (2020). *Penerapan Image Classification Dengan Pre-trained Model MobileNet dalam Client-side Machine Learning*. Universitas Islam Negeri Syarif Hidayatullah. Jakarta.
- Shindy, F. (2019). *Pendeteksian Objek Manusia Secara Real Time dengan Metode MobileNet-SSD Menggunakan Movidius Neural Compute Stick pada Raspberry Pi*. Universitas Sumatera Utara. Medan.
- Singh, M., Verma, A., Parasher, A., Chauhan, N., & Budhiraja, G. (2019). *Implementation of Database Using Python Flask Framework*. Vol. 8, no. 12, pp. 24894-24899.
- A. G. Howard., M. Zhu., B. Chen., D. Kalenichenko., W. Wang., T. Weyand., M. Andreetto., H. Adam. (2017). *Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. dan Berg, A.C., (2015). *SSD: Single Shot Multibox Detector, International Journal of European Conference on Computer Vision*, Vol 5