

Klasifikasi Jenis I

YOLO (*You Only Look Once*)

Zainal Abidin¹⁾ Heri Pratikno²⁾ Weny Indah Kusumawati³⁾

Program Studi/Jurusan Teknik Komputer

Fakultas Teknologi dan Informatika

Universitas Dinamika

Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email: 1)15410200053@dinamika.ac.id 2)heri@dinamika.ac.id, 3)weny@dinamika.ac.id

Abstrak: Jalan tol merupakan proyek yang digadang-gadang pemerintah dapat mengurai kemacetan sampai dapat menjadi sumber pemasukan kas negara. Saat ini Indonesia sudah mengandalkan jalan tol sebagai jalur transportasi antar daerah. Gerbang tol adalah tempat pelayanan transaksi tol bagi pemakai tol yang terdiri dari beberapa gardu dan sarana perlengkapan lainnya. Ketika terdapat gerbang tol, maka pasti terdapat kemacetan karena dalam proses menentukan golongan kendaraan masih dilakukan secara manual. Berdasarkan pokok permasalahan tersebut diatas, maka di dalam panalisaenelitian ini memberikan ide dan solusi yang berupa sebuah program untuk menklasifikasikan kendaraan dengan menggunakan kamera. Fungsi dari aplikasi ini adalah untuk menentukan jenis kendaraan berdasarkan golongan kendaraan pada data Jalan Tol agar dapat mempercepat proses pembayaran pada gerbang Tol, sehingga mengurangi kemacetan. Tingkat keberhasilan pada aplikasi klasifikasi jenis kendaraan adalah 93% dengan tingkat *error* 7%.

Kata kunci : Klasifikasi Jenis Kendaraan, Deteksi Kendaraan Dengan YOLO, Penghitung Kendaraan.

Pembangunan jalan tol merupakan salah satu bentuk usaha pemerintah dalam memudahkan masyarakat di Indonesia untuk dapat melakukan mobilitas mereka baik dalam hal ekonomi maupun sosial dengan baik dan cepat. Pembangunan dengan skala besar selain membutuhkan modal besar juga membutuhkan tanah untuk mendirikan bangunan tersebut. Jalan tol merupakan proyek yang digadang-gadang pemerintah dapat mengurai kemacetan sampai dapat menjadi sumber pemasukan kas negara. Saat ini Indonesia sudah mengandalkan jalan tol sebagai jalur transportasi antar daerah. Gerbang tol adalah tempat pelayanan transaksi tol bagi pemakai tol yang terdiri dari beberapa gardu dan sarana perlengkapan lainnya. Ketika terdapat gerbang tol, maka pasti terdapat kemacetan karena dalam proses menentukan golongan kendaraan masih dilakukan secara manual.

Berdasarkan pokok permasalahan tersebut diatas, maka di dalam Penelitian ini memberikan ide dan solusi yang berupa sebuah program untuk menklasifikasikan kendaraan dengan teknologi kamera. Fungsi dari aplikasi ini adalah untuk menentukan jenis kendaraan berdasarkan golongan

kendaraan pada data Jalan Tol agar dapat mempercepat proses pembayaran pada gerbang Tol, sehingga mengurangi kemacetan.

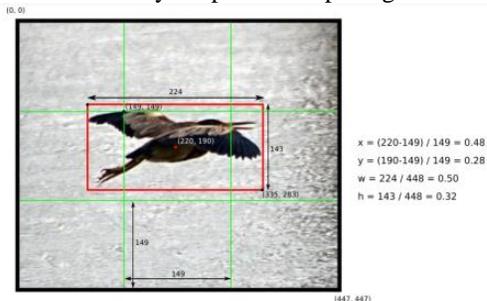
Pada saat ini telah banyak dilakukan penelitian terkait permasalahan ini, salah satunya adalah penelitian Dwi Ramadhani dari Universitas Telkom dengan aplikasi penghitung kendaraan dengan mendeteksi kendaraan menggunakan CVblob dan penampil Google Map (Ramadhani, 2014). Adapun penelitian lainnya dalam bentuk rancang bangun penghitung jumlah kendaraan roda empat dengan menggunakan mikrokontroler AT89S51 oleh Zulkurniawan dari Universitas Sriwijaya (Ecotipe, 2015).

Pada penelitian ini, penulis memberikan kontribusi melalui pembuatan sebuah aplikasi penghitung jumlah kendaraan berdasarkan jenisnya menggunakan metode YOLO (*You Only Look Once*). Adapun kelebihan metode YOLO adalah dapat mengidentifikasi objek dengan akurasi tinggi, selain itu juga dapat melakukan proses *tracking* dalam bentuk *bounding box* terhadap objek secara *realtime* dan memberikan label kendaraan sesuai dengan jenis kendaraannya.

METODE PENELITIAN

You Only Look Once (YOLO)

YOLO membagi suatu gambar yang dimasukkan menjadi suatu *grid* berukuran $S \times S$, dimana nilai S adalah 7 dengan *input* gambar berukuran 250×450 . Untuk mendapatkan *bounding box*, dilakukan konvolusi dari masukan gambar, sehingga hasil akhirnya mendapatkan ukuran *bounding box* sebesar $S \times S \times B * (5 + C)$ dimana B adalah banyaknya *bounding box* dalam 1 *grid* dan C adalah banyaknya class yang dapat diklasifikasi. Nilai B dikalikan dengan 5 karena sebuah *bounding box* memiliki 5 nilai yang perlu disimpan, koordinat x , koordinat y , lebar (*width*), tinggi (*height*) dan *confidence score* (nilai probabilitas *bounding box* yang bersangkutan memiliki sebuah objek). Untuk semua atribut pada *bounding box* dilakukan normalisasi, sehingga nilainya menjadi antara 0 hingga 1. Koordinat x dan y dinormalisasi menyesuaikan titik kiri atas dan *grid* yang bersangkutan. Dan tinggi dan lebar dinormalisasi sesuai dengan ukuran gambar. Nilai koordinat x dan y pada sebuah *bounding box* pada setiap *grid* merupakan titik tengah *grid* yang bersangkutan. Untuk ilustrasinya dapat dilihat pada gambar 1.



Gambar 1. Grid YOLO

Image Processing

Secara harfiah, citra (*image*) adalah gambar pada bidang dwimatra (dua- dimensi). Ditinjau dari sudut pandangan matematis, citra merupakan fungsi terus-menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian berkas cahaya tersebut, pantulan cahaya ini ditangkap oleh alat optik, sehingga bayangan objek yang disebut citra tersebut terekam. Citra digital dapat diartikan sebagai suatu fungsi dua dimensi $f(x,y)$, berukuran M baris dan N kolom, sedangkan x dan y adalah posisi koordinat spasial dan amplitudo f di titik koordinat (x,y) yang dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Nilai dari intensitas bentuknya adalah diskrit mulai dari 0 sampai 255, citra yang

ditangkap oleh kamera dan telah dikuantisasi dalam bentuk nilai diskrit disebut sebagai citra digital (*digital image*). Citra digital tersusun dari sejumlah nilai tingkat keabuan yang dikenal sebagai piksel (*pixel*). Untuk melakukan pemrosesan citra digital, maka citra analog harus dikonversi terlebih dahulu kedalam bentuk citra digital. Ada dua jenis citra digital, citra diam (*still image*) dan citra bergerak (*moving image*). Suatu citra (gambar) analog dengan ukuran panjang kali lebar, dapat didigitalisasi dengan mengambil *sampling* berupa matriks berukuran m kali n , dengan m adalah jumlah *sampling* untuk panjang, dan n adalah jumlah *sampling* untuk lebar.

Pengambilan citra dapat dilakukan oleh kamera atau alat lain yang dapat digunakan untuk mentransfer gambar. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra dipengaruhi oleh berbagai faktor yang mengakibatkan penampilan citra suatu benda tidak sama persis dengan bentuk aslinya. Faktor-faktor tersebut merupakan efek degradasi atau penurunan kualitas yang dapat berupa rentang kontras benda yang terlalu sempit atau terlalu lebar, distorsi, kekaburan (*blur*), kekaburan akibat objek citra dan sebagainya.

Bahasa Python

Saat ini pengembangan *Python* terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan *Python Software Foundation*. *Python Software Foundation* adalah sebuah organisasi *non-profit* yang dibentuk sebagai pemegang hak cipta intelektual *Python* sejak versi 2.1 dan dengan demikian mencegah *Python* dimiliki oleh perusahaan komersial. Saat ini distribusi *Python* sudah mencapai versi 2.6.1 dan versi 3.0. Nama *Python* dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi *Monty Python's Flying Circus*. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna *Python*. *Python* dapat dilihat pada gambar 2.



Gambar 2. Python logo

OpenCV

OpenCV (Open source computer vision) adalah pustaka/*library* dari fungsi pemrograman

yang terutama ditujukan untuk *computer vision* yang *realtime*. Pada awalnya *OpenCV* dikembangkan oleh Intel, kemudian didukung oleh Willow Garage lalu Itseez (yang kemudian diakuisisi oleh Intel). Perpustakaan/*library* dari *OpenCV* adalah lintas-*platform* dan gratis untuk digunakan di bawah lisensi BSD *open-source*. *OpenCV* juga mendukung kerangka pembelajaran yang terdapat pada *TensorFlow*, *Torch / PyTorch* dan *Caffe*. Logo *OpenCV* dapat dilihat pada gambar 3.



Gambar 3. Logo OpenCV

Jenis Mobil

Di Indonesia terdapat beberapa jenis mobil yang melintas pada jalan toll, sehingga pada Jalan toll menjadi golongan untuk menentukan tariff toll. Pada penelitian ini penulis mendeteksi 4 jenis mobil saja yaitu sedan, MPV, Bus dan Truk. Mobil jenis sedan adalah mobil yang didesain untuk 5 penumpang. Berbeda dengan jenis mobil lain yang berukuran lebih besar, sedan berukuran lebih kecil, sehingga mengurangi biaya pajak dan biaya keseluruhan dari jenis mobil lain yang ukurannya lebih besar. Berikut adalah contoh dari mobil Sedan. Mobil Sedan dapat dilihat pada gambar 4.



Gambar 4. Mobil Sedan

Mobil dengan jenis Multi Purpose Vehicle atau biasa disingkat MPV adalah mobil yang didesain untuk mengangkut jumlah penumpang yang lebih banyak dengan nyaman. Jumlah

penumpang yang dimaksud bervariasi dari 5 hingga 7 orang dengan ruang untuk penyimpanan bagasi tergantung dari model dan desain mobilnya. MPV biasa juga dikenal dengan sebutan minivan. Salah satu contoh mobil MPV dapat dilihat pada gambar 5.



Gambar 5. Mobil MPV

Bus adalah kendaraan kelas besar yang didesain untuk mengangkut penumpang dalam jumlah yang banyak dan memiliki jalur khusus. Bus awalnya diciptakan untuk menggantikan peran streetcar (semacam kereta kecil yang memiliki rel) sebagai kendaraan umum dengan rute khusus yang lebih fleksibel (lebih fleksibel karena bus dapat dioperasikan pada jalan umum tanpa menggunakan rel seperti streetcar). Contoh bus dapat dilihat pada gambar 6.



Gambar 6. Bus

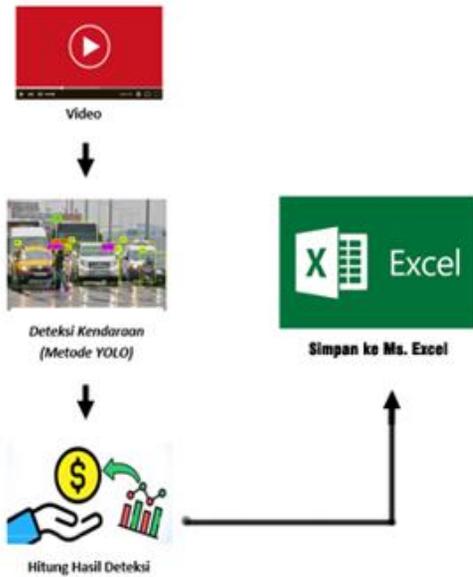
Dump truck merupakan alat berat yang berfungsi untuk mengangkut atau memindahkan material pada jarak menengah sampai jarak jauh (> 500m). Dump truck biasa digunakan untuk mengangkut material alam seperti tanah, pasir, batu split, dan juga material olahan seperti beton kering pada proyek konstruksi. Umumnya material yang dimuat pada dump truck oleh alat pemuat seperti excavator backhoe atau loader. Untuk membongkar muatan material bak dump truck dapat terbuka dengan bantuan sistem hidrolik.



Gambar 7. Truck

Skema Sistem

Skema sitem dibuat berdasarkan pengkabelan pada masing masing perangkat. Perancangan dapat dilihat pada gambar 8.

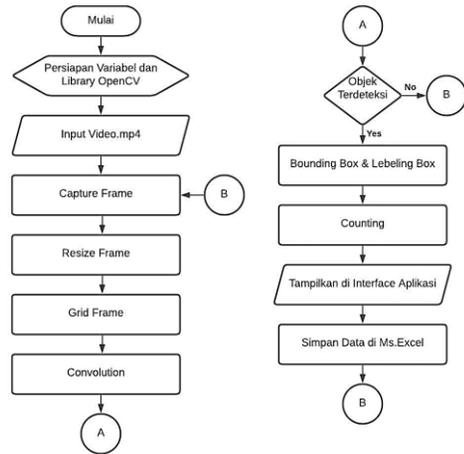


Gambar 8. Diagram sistem

Saat suatu objek memiliki tingkat probabilitas yang linier dengan *dataset*, maka YOLO memberi sebuah *bounding box* terhadap objek dan memberi sebuah *labeling* sesuai dengan objek yang terdeteksi. Di dalam sebuah citra video terdapat berbagai jenis objek kendaraan. Sistem melakukan klarifikasi dan menghitung setiap objek berdasarkan jenis kendaraan. Hasil dari komputasi tersebut disimpan kedalam *database Ms. Excel* yang nantinya data ini dilanjutkan pengiriman ke *client* yang merupakan aplikasi *desktop*. Aplikasi *desktop* menampilkan sebuah data berupa grafik atau tabel sesuai dari hasil komputasi program.

Flowchart Sistem

Flowchart sistem adalah alur dari sebuah pemrograman dimana pemrograman mempunyai library pada sistem yang digunakan. *Flowchart* dapat dilihat pada gambar 9.

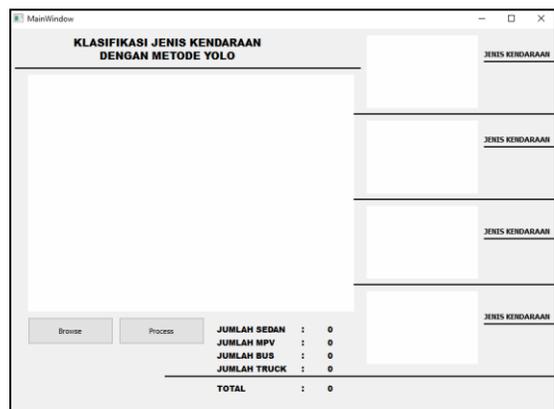


Gambar 9. Flowchart sistem

Pada gambar 9 terdapat persiapan variabel dan library *OpenCV* pada awal program kemudian dilanjutkan dengan *input Video.mp4* dimana format video harus mp4. Setelah itu masuk pada *Capture frame* dimana pada langkah ini adalah mengambil gambar pada video yang telah diinput. Hasil dari *capture frame* tersebut adalah berupa ruang warna RGB kemudian dilakukan *resize frame* untuk memperkecil *pixel*. Hasil dari *resize frame* tersebut dibentuk *grid*, sehingga dapat melakukan proses konvolusi. Semua proses tersebut adalah YOLO Detection. Jika objek terdeteksi, maka dilanjutkan dengan *bounding box* dan labeling tetapi jika objek tidak terdeteksi, maka dilakukan *capture frame* ulang. Hasil dari *bounding box* dan labeling ditampilkan pada aplikasi *desktop* dan disimpan pada Microsoft *Excel*.

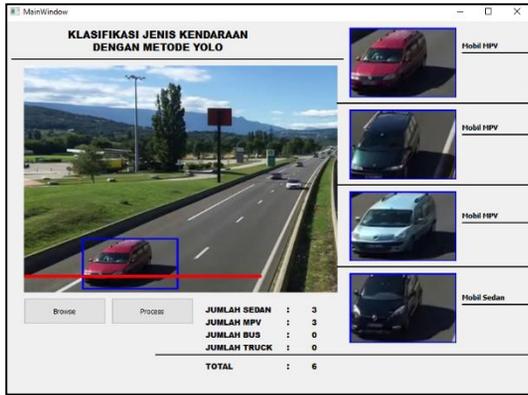
Desain Aplikasi

Pada Desain aplikasi ini adalah purwarupa dari aplikasi *desktop* yang telah dibuat oleh penulis. Desain aplikasi dapat dilihat pada gambar 10.



Gambar 10. Desain aplikasi

Pada gambar 10 tersebut adalah desain aplikasi dengan terdapat tombol *Browse* dan *Proses* serta terdapat frame utama untuk menampilkan video. Pada bagian kanan terdapat 4 frame digunakan untuk menampilkan mobil yang terdeteksi. Hasil pada pengolahan desain tersebut dapat dilihat pada gambar dibawah ini.



Gambar 11. Hasil aplikasi

HASIL DAN PEMBAHASAN

Pengujian Metode YOLO pada Kendaraan

Hasil pengujian metode YOLO pada kendaraan digunakan untuk mengetahui cara konsep dari metode YOLO itu sendiri. Langkah pertama adalah membuat daset pada Microsoft *Excel*. *Dataset* tersebut dapat membantu YOLO dalam proses konvolusi. Setelah semua *dataset* gambar diberi label langkah selanjutnya adalah melakukan training untuk menghasilkan model yang dipakai untuk pendeteksian objek. Lamanya proses training tergantung dari seberapa besar *dataset* yang dibuat. Semakin banyak jumlah gambar semakin lama juga proses trainingnya, tetapi akurasi yang dihasilkan semakin baik. Sebelum proses training perlu untuk mengkonfigurasi proses konvolusi, yaitu metode CNN yang digunakan YOLO. Konfigurasi yang perlu disesuaikan ada dibagian konvolusi dan region yaitu:

a. Filters

Rumus untuk nilai filters adalah:

$$(5 + \text{Jumlah Kelas}) \times 5$$

Penulis menggunakan 4 kelas untuk pendeteksian jadi konfigurasinya adalah
 $\text{Filters} = (5 + 4) \times 5 = 45$

b. Classes

Penulis menggunakan 4 kelas untuk pendeteksian objek jadi classes diisi dengan nilai 4.

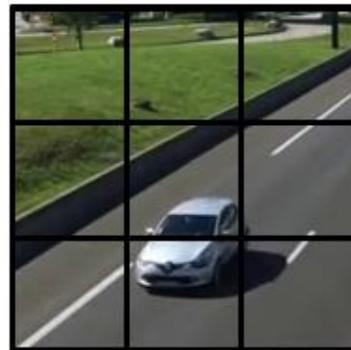
Konfigurasi diatas menghasilkan proses konvolusi sebagai berikut:

Tabel 1. Proses Konvolusi

Layer	Size	Stride	Output
Input	-	-	416 x 416 x 3
Conv 1	3 x 3	1	416 x 416 x 16
Max Pool 1	2 x 2	2	208 x 208 x 16
Conv 2	3 x 3	1	208 x 208 x 32
Max Pool 2	2 x 2	2	104 x 104 x 32
Conv 3	3 x 3	1	104 x 104 x 64
Max Pool 3	2 x 2	2	52 x 52 x 64
Conv 4	3 x 3	1	52 x 52 x 128
Max Pool 4	2 x 2	2	26 x 26 x 128
Conv 5	3 x 3	1	26 x 26 x 256
Max Pool 5	2 x 2	2	13 x 13 x 256
Conv 6	3 x 3	1	13 x 13 x 512
Max Pool 6	2 x 2	1	13 x 13 x 512
Conv 7	3 x 3	1	13 x 13 x 1024
Conv 8	3 x 3	1	13 x 13 x 1024
Conv 9	1 x 1	1	13 x 13 x 45

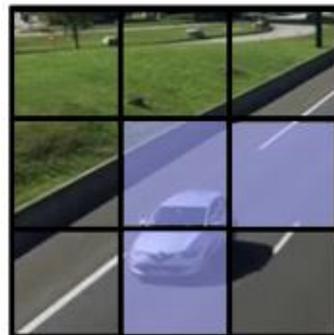
Berdasarkan arsitektur yang digunakan YOLO berikut proses prediksi terhadap *dataset* yang digunakan dengan konfigurasi diatas:

1. *Input* gambar diubah menjadi 416 x 416 kemudian dibagi menjadi *grid* sesuai konfigurasi yang ditentukan yaitu $S \times S$ dimana $S = 3$ seperti gambar 12 berikut:

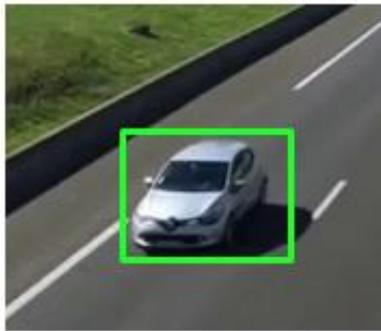


Gambar 12. Grid 3x 3

2. Kemudian setiap *grid* memprediksi *B Bounding box* dan *C class probabilities* dimana nilai $B = 2$ dan $C = 4$ (Jumlah kelas yang digunakan). Prediksi *bounding box* ditunjukkan pada gambar 13 dan gambar 14 berikut:



Gambar 13. Hasil prediksi



Gambar 14. Hasil *Bounding Box*

Hasil prediksi *Bounding box* menghasilkan 5 komponen yaitu x, y, w, h dan confidence. x dan y adalah koordinat yang ada ditengah box dan bersifat relatif terhadap ukuran gambar.

Hasil Pengujian *Counting information*

Pada hasil pegujian *counting* ini digunakan untuk menentukan tingkat keberhasilan dan kelayakan aplikasi dalam menghitung kendaraan. Scenario dalam pengujian ini adalah dengan cara menguji aplikasi pada 1 video yang berdurasi 2 menit. Hasil pengujian dapat dilihat pada dan tabel 2.

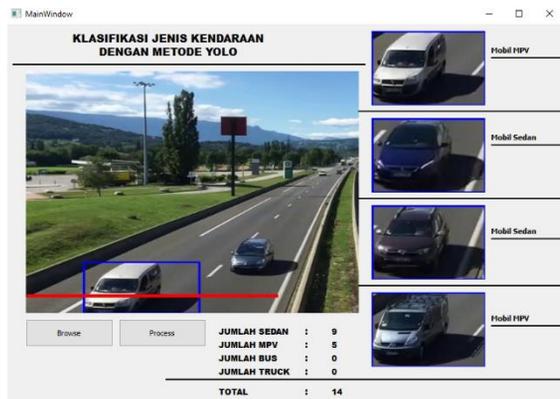
Tabel 2. *Counting information*

Nomor	Detik	Jumlah Kendaraan		
		Terdeteksi	Manual	Selisih
1	ke-10	5	5	0
2	ke-20	12	13	1
3	ke-30	25	29	4
4	ke-40	33	37	4
5	ke-50	46	50	4
6	ke-60	58	62	4
7	ke-70	65	70	5
8	ke-80	82	87	5
9	ke-90	104	111	7
10	ke-100	116	123	7
11	ke-110	128	135	7
12	ke-120	136	145	9
Total Kendaraan		136	145	9

Pengambilan data pada tabel 2 tersebut dengan menguji keakuratan penghitungan detksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak selisih atau disebut juga dengan *error*. Pada keakuratan penghitung kendaraan memiliki tingkat keberhasilan yaitu 93%.

Hasil Pengujian Klasifikasi Kendaraan

Pada pengujian klasifikasi kendaraan digunakan untuk mengetahui apakah proses klasifikasi dapat berjalan dengan baik atau tidak. Proses klasifikasi ini adalah menentukan jenis kendaan apa yang dideteksi oleh sistem. Pada pengujian kali ini memiliki scenario yaitu pengujian dengan menggunakan 1 video berdurasi 2 menit. Hasil dari pengujian klasifikasi kendaraan dapat dilihat pada gambar 15.



Gambar 15. Hasil pengujian klasifikasi kendaraan

Tabel 3. Hasil pengujian klasifikasi

No	Detik	Jumlah Kendaraan Terdeteksi				Jumlah Kendaraan Manual			
		Sedan	MPV	Bus	Truck	Sedan	MPV	Bus	Truck
1	ke-10	4	1	0	0	4	1	0	0
2	ke-20	7	4	0	1	7	5	0	1
3	ke-30	12	9	1	3	12	13	1	3
4	ke-40	15	14	1	3	15	17	1	3
5	ke-50	19	23	1	3	19	27	1	3
6	ke-60	25	29	1	3	25	33	1	3
7	ke-70	31	30	1	3	34	36	1	3
8	ke-80	38	38	2	4	38	38	2	4
9	ke-90	42	55	2	5	42	55	2	5
10	ke-100	45	64	2	5	45	64	2	5
11	ke-110	50	71	2	5	53	77	2	5
12	ke-120	52	77	2	5	56	82	2	5
Total		52	77	2	5	56	82	2	5

Pada tabel 3 tersebut terdapat proses klasifikasi kendaraan dengan kendaraan terdeteksi pada mobil sedan adalah 52, sedangkan deteksi manual pada sedan adalah 56, sehingga selisih pada mobil sedan adalah 4 mobil sedan. Untuk mobil MPV yang dapat dideteksi adalah 77, sedangkan dalam kondisi manual adalah 82, sehingga mobil MPV memiliki selisih yaitu 5 mobil MPV. Untuk mobil bus dan truk tidak memiliki selisih karena proses deteksi sama dengan perhitungn manual.

KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, dapat diambil beberapa kesimpulan. Kesimpulan tersebut diantaranya:

1. Metode YOLO dapat digunakan untuk mengklasifikasi jenis kendaraan dan menghitung jenis kendaraan dengan cukup baik
2. Pada pengujian *counting information* dapat mendeteksi 93% dan memiliki tingkat *error* 7%.
3. Pada pengujian klasifikasi proses klasifikasi kendaraan dengan kendaraan terdeteksi pada mobil sedan adalah 52, sedangkan deteksi manual pada sedan adalah 56, sehingga selisih pada mobil sedan adalah 4 mobil sedan. Untuk mobil MPV yang dapat dideteksi adalah 77, sedangkan dalam kondisi manual adalah 82, sehingga mobil MPV memiliki selisih yaitu 5 mobil MPV. Untuk mobil bus dan truk tidak memiliki selisih karena proses deteksi sama dengan perhitungan manual.

DAFTAR PUSTAKA

- Directory, S. (2018 , Februari 27). *Web Application*. Retrieved from *Desktop Applications Vs. Web Applications*: <https://www.streetdirectory.com/>
- Javanet, M. (2017, April 21). *Javanet Media*. Retrieved from *Pengertian dan Fungsi SQL Server* : <http://www.javanetmedia.com/2017/04/pengertian-dan-fungsi-sql-server.html>
- Nishad, G. (2019, Maret 9). *You Only Look Once(YOLO)*. Retrieved from *Implementing YOLO in less than 30 lines of Python Code*: <https://towardsdatascience.com/you-only-look-once-yolo-implementing-yolo-in-less-than-30-lines-of-python-code-97fb9835bfd2>
- Ramadhani, D. (2014). *Aplikasi Penghitung Kendaraan Dengan Deteksi Kendaraan Menggunakan Cvblob Dan Penampil Google Map B Erbasis Android*. *Openlibrary-Telkom*, 23.