



**KLASIFIKASI JENIS KENDARAAN PADA GERBANG TOL  
MENGUNAKAN METODE YOLO (*YOU ONLY LOOK ONCE*)**



**TUGAS AKHIR**

**Program Studi  
S1 TEKNIK KOMPUTER**

**UNIVERSITAS  
Dinamika**

**Oleh:**

**Zainal Abidin**

**15410200053**

---

---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2021**

**KLASIFIKASI JENIS KENDARAAN PADA GERBANG TOL  
MENGUNAKAN METODE YOLO (*YOU ONLY LOOK ONCE*)**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik**



Oleh :

Nama : Zainal Abidin  
NIM : 15410200053  
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**

**UNIVERSITAS DINAMIKA**

**2021**

**TUGAS AKHIR**  
**KLASIFIKASI JENIS KENDARAAN PADA GERBANG TOL**  
**MENGGUNAKAN METODE YOLO (*YOU ONLY LOOK ONCE*)**

Dipersiapkan dan disusun oleh

**Zainal Abidin**

**Nim : 15410200053**

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada : 21 Januari 2021

**Susunan Dewan Pembahas**

**Pembimbing :**

**I. Heri Pratikno, M.T., MTCNA., MTCRE.**  
NIDN. 0716117302



Digitally signed by  
Universitas Dinamika  
Date: 2021.01.29  
14:19:16 +07'00'

**II. Weny Indah Kusumawati, S.Kom., M.MT.**  
NIDN. 0721047201



Digitally signed by  
Universitas Dinamika  
Date: 2021.01.29  
14:44:31 +07'00'

**Pembahas :**

**Pauladie Susanto, S.Kom., M.T.**  
NIDN. 0729047501



Digitally signed by  
Universitas Dinamika  
Date: 2021.01.29  
14:56:41 +07'00'

Tugas akhir ini untuk diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana



**D. Jusak**

NIDN. 0708017101

Dekan Fakultas Teknologi dan Informatika  
UNIVERSITAS DINAMIKA

Digitally signed by  
Universitas  
Dinamika  
Date: 2021.02.11  
08:44:35 +07'00'



*“Makrifat”*  
UNIVERSITAS  
Dinamika

**SURAT PERNYATAAN  
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Zainal Abidin

NIM : 15410200053

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Tugas Akhir

Judul Karya: : **KLASIFIKASI JENIS KENDARAAN PADA GERBANG  
TOL MENGGUNAKAN METODE YOLO (YOU ONLY  
LOOK ONCE)**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan ilmu pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialih mediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kejarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 21 Januari 2021

Yang menyatakan



Zainal Abidin

NIM : 15410200053

## ABSTRAK

Pembangunan jalan tol merupakan salah satu bentuk usaha pemerintah dalam memudahkan masyarakat di Indonesia untuk bisa 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. Pada gerbang tol pasti terdapat kemacetan karena proses menentukan golongan kendaraan masih dilakukan secara manual.

Berdasarkan pokok permasalahan tersebut diatas maka pada Tugas Akhir ini memberikan ide dan solusi yang berupa sebuah program untuk mengklasifikasikan jenis kendaraan menggunakan kamera. Manfaat dari aplikasi ini adalah untuk menentukan jenis kendaraan berdasarkan golongan kendaraan berdasarkan data Jalan Tol secara otomatis untuk mempercepat proses pembayaran pada gerbang Tol sehingga mengurangi kemacetan. Tingkat keberhasilan pada aplikasi klasifikasi jenis kendaraan adalah 93% dengan tingkat *error*-nya 7%.

**Kata Kunci :** Klasifikasi Jenis Kendaraan, deteksi kendaraan, YOLO, Penghitung jumlah kendaraan.

## KATA PENGANTAR

Segala puji syukur bagi Allah SWT yang telah memberikan pertolongan dalam setiap kesulitan yang ada selama pelaksanaan Tugas Akhir ini. Atas berkat rahmat-NYA, pelaksanaan Tugas Akhir dapat terselesaikan dengan baik. Pelaksanaan Tugas Akhir ini merupakan sebuah syarat untuk menempuh wisuda pada program Studi S1 Teknik Komputer Universitas Dinamika Surabaya. Selama menyelesaikan penulisan laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dan dukungan dari berbagai pihak baik moral dan doa maupun materi. Oleh karena itu penulis mengucapkan terima kasih banyak kepada:

1. Allah SWT, karena memberikan kehidupan serta panca indra untuk dapat digunakan dalam menyelesaikan Tugas Akhir ini serta takdir kelulusan menempuh S1 Teknik Komputer bagi saya.
2. Orang Tua saya tercinta yang telah memberikan dukungan dan bantuan baik moral maupun doa, sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir maupun laporan ini.
3. Para Dosen Fakultas Teknik Komputer Universitas Dinamika yang telah sabar dalam membimbing dan mengajari saya selama waktu perkuliahan.
4. Teman-teman seperjuangan Teknik Komputer yang telah memberikan semangat dan bantuan selama ini.

Surabaya, 21 Januari 2021

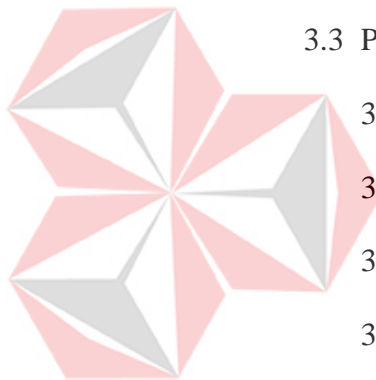
Penulis

## DAFTAR ISI

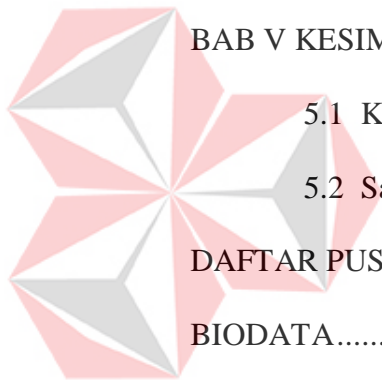
ABSTRAK .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	4
1.6 Metodologi Pengerjaan Tugas Akhir .....	4
1.7 Sistematika Penulisan .....	6
BAB II LANDASAN TEORI .....	7
2.1 <i>You Only Look Once (YOLO)</i> .....	7
2.2 <i>Image processing</i> .....	8
2.3 Bahasa Python .....	9
2.4 <i>OpenCV</i> .....	11
2.5 Jenis Mobil .....	12
2.2.1 Mobil Sedan .....	13
2.2.2 Mobil MPV .....	13
2.2.3 Mobil BUS .....	14



2.2.4 Mobil <i>Truck</i> .....	14
2.6 <i>Confusion Matrix</i> .....	15
2.7 PyQT .....	16
2.8 Microsoft Excel.....	17
2.9 Citra.....	18
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>19</b>
3.1 Studi Literatur .....	20
3.2 Perancangan Sistem.....	21
3.2.1 Blok Diagram Sistem.....	21
3.2.2 <i>Flowchart</i> Sistem.....	23
3.3 Proses Preprocessing .....	24
3.3.1 <i>Capture Frame</i> .....	24
3.3.2 RGB to <i>Greyscale</i> .....	25
3.3.3 <i>Greyscale</i> to <i>Binary</i> .....	26
3.3.4 <i>Grid Frame</i> .....	27
3.3.5 <i>Morphology</i> .....	27
3.3.6 <i>Masking</i> .....	28
3.3.7 <i>Bounding Box</i> dan <i>Labeling</i> .....	29
3.3.8 <i>Cropped</i> .....	29
3.3.9 <i>Counting</i> .....	30
3.3.10 Menyimpan data di Microsoft Excel.....	30
3.4 Desain Aplikasi.....	31
3.5 Pengujian Sistem.....	32
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	<b>33</b>



4.1 Analisis Kebutuhan Perangkat Lunak ( <i>Software</i> ).....	33
4.2 Analisis <i>Input</i> dan <i>Output</i> Sistem .....	34
4.2.1 <i>Form</i> pengolah <i>Image processing</i> .....	34
4.2.2 Tombol <i>Browse</i> .....	35
4.2.3 Tombol Proses .....	36
4.2.4 <i>Browse</i> .....	36
4.2.5 <i>Counting information</i> .....	37
4.3 Hasil Pengujian Metode YOLO pada Kendaraan .....	38
4.4 Hasil Pengujian <i>Counting information</i> .....	41
4.5 Hasil Pengujian Klasifikasi Kendaraan .....	42
BAB V KESIMPULAN DAN SARAN .....	44
5.1 Kesimpulan .....	44
5.2 Saran .....	44
DAFTAR PUSTAKA .....	46
BIODATA .....	467

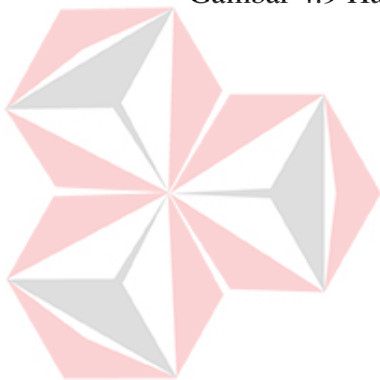


UNIVERSITAS  
Dinamika

## DAFTAR GAMBAR

Gambar 2.1 YOLO .....	8
Gambar 2.2 Python Logo .....	9
Gambar 2.3 <i>OpenCV</i> .....	11
Gambar 2.4 Mobil Sedan .....	13
Gambar 2.5 Mobil MPV .....	13
Gambar 2.6 Mobil BUS .....	14
Gambar 2.7 <i>Truck</i> .....	15
Gambar 2.8 Tampilan Qt Designer.....	17
Gambar 2.9 Microsoft Excel .....	18
Gambar 3.1 Metodologi Penelitian.....	19
Gambar 3.2 Blok diagram aplikasi penghitung kendaraan .....	22
Gambar 3.3 <i>Flowchart</i> sistem .....	23
Gambar 3.4 <i>Capture Frame</i> (RGB).....	25
Gambar 3.5 Citra <i>greyscale</i> .....	26
Gambar 3.6 Citra <i>binary</i> .....	27
Gambar 3.7 Hasil <i>grid</i> dan <i>morphology</i> .....	28
Gambar 3.8 Hasil <i>masking</i> .....	28
Gambar 3.9 Hasil <i>bounding box</i> .....	29
Gambar 3.10 Hasil <i>cropped</i> .....	29
Gambar 3.11 Hasil <i>counting</i> .....	30
Gambar 3.12 Hasil Microsoft Excel .....	30
Gambar 3.13 Desain aplikasi .....	31

Gambar 3.14 Hasil pengolah aplikasi .....	32
Gambar 4.1 Form pengolah <i>image processing</i> .....	35
Gambar 4.2 Tombol <i>browse</i> .....	35
Gambar 4.3 Tombol proses .....	36
Gambar 4.4 <i>Browse</i> .....	37
Gambar 4.5 <i>Counting information</i> .....	37
Gambar 4.6 <i>Grid 3x 3</i> .....	39
Gambar 4.7 Hasil prediksi.....	40
Gambar 4.8 Hasil <i>bounding box</i> .....	40
Gambar 4.9 Hasil pengujian klasifikasi kendaraan .....	42



UNIVERSITAS  
**Dinamika**

## DAFTAR TABEL

Tabel 4.1 Proses konvolusi .....	39
Tabel 4.2 Hasil pengujian pada video.....	41
Tabel 4.3 Hasil pengujian klasifikasi kendaraan .....	42



UNIVERSITAS  
**Dinamika**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pembangunan jalan tol merupakan salah satu bentuk usaha pemerintah dalam memudahkan masyarakat di Indonesia untuk bisa 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 Tugas Akhir ini memberikan ide dan solusi yang berupa sebuah program untuk mengklasifikasikan kendaraan dengan teknologi kamera. Fungsi dari aplikasi ini adalah untuk menentukan jenis kendaraan berdasarkan golongan kendaraan pada data Jalan Tol agar bisa 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 Tugas Akhir ini, penulis akan 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.

## 1.2 Rumusan Masalah

Berdasarkan dari latar belakang tersebut diatas terdapat beberapa rumusan masalah sebagai berikut:

1. Bagaimana implementasi proses deteksi kendaraan menggunakan metode YOLO ?
2. Bagaimana YOLO dapat mengetahui atau membedakan jenis kendaraan ?
3. Bagaimana cara melakukan proses perhitungan jumlah kendaraan berdasarkan jenisnya ?

### 1.3 Batasan Masalah

Dalam penerapan aplikasi penghitung jumlah kendaraan berdasarkan jenisnya menggunakan metode YOLO (*You Only Look Once*) terdapat beberapa batasan masalah sebagai berikut:

1. Pengujian menggunakan video yang sudah disiapkan.
2. Objek yang dideteksi adalah objek yang utuh dan tidak terdapat penghalang secara penuh.
3. Jenis kendaraan yang deteksi adalah truk, bis, dan sedan.
4. Proses deteksi dilakukan dengan pencahayaan yang cukup.
5. Menggunakan bahasa pemrograman Python.
6. Menggunakan library *OpenCV*.

### 1.4 Tujuan

Berdasarkan paparan rumusan masalah diatas, maka tujuan dari Tugas

Akhir ini adalah:

1. Dapat melakukan proses deteksi kendaraan menggunakan metode YOLO (*You Only Look Once*).
2. Aplikasi yang dibuat dengan metode YOLO dapat mengetahui atau membedakan jenis kendaraan.
3. Masing-masing jenis kendaraan yang terdeteksi dapat dilakukan perhitungan jumlahnya.



### 1.5 Manfaat

Manfaat yang diharapkan dari pelaksanaan projek Tugas Akhir ini adalah:

1. Menguji tentang kemampuan penerapan ilmu pengetahuan dan teknologi yang telah diperoleh selama dalam perkuliahan.
2. Sistem tersebut dapat digunakan untuk mempermudah dalam mengklasifikasi kendaraan dan menghitung kendaraan.

### 1.6 Metodologi Pengerjaan Tugas Akhir

Metodologi pengerjaan Tugas Akhir yang akan dikerjakan memiliki 5 tahap utama. Kelima tahap tersebut antara lain: tahap persiapan, tahap perencanaan, tahap pengumpulan data, tahap perancangan dan pembuatan alat, tahap analisis dan pengolahan data, dan tahap pemecahan masalah.

#### 1. Tahap Persiapan

Tahap persiapan merupakan rangkaian kegiatan sebelum memulai pengumpulan dan pengolahan data. Dalam tahap awal ini disusun hal-hal penting yang harus segera dilakukan dengan tujuan untuk mengefektifkan waktu dan pekerjaan. Tahap persiapan ini meliputi kegiatan-kegiatan sebagai berikut: menentukan judul Tugas Akhir, pembuatan proposal penyusunan Tugas Akhir, studi pustaka terhadap materi desain untuk menentukan garis besarnya, dan menentukan kebutuhan data. Perencanaan jadwal pembuatan desain persiapan diatas harus dilakukan secara cermat untuk menghindari pekerjaan yang berulang, sehingga tahap pengumpulan data menjadi lebih optimal.

## 2. Tahap Perencanaan

Tahap ini memberikan gambaran mengenai langkah awal sampai dengan akhir penyusunan laporan Tugas Akhir. Pengembangan penjelasannya dituangkan dalam bentuk diagram alir yang tersusun secara berurutan sesuai dengan proses kerjanya.

## 3. Pengumpulan Data

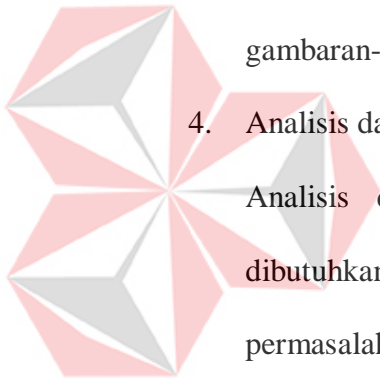
Metode pengumpulan data yang dilakukan yaitu sebagai berikut: Metode Literatur, yaitu mengumpulkan, mencari dan mengolah data tertulis serta metode kerja yang digunakan, Metode Observasi yang dilakukan dengan tujuan untuk dapat mengetahui kondisi riil di lapangan sehingga dapat diperoleh gambaran-gambaran sebagai pertimbangan dalam perencanaan desain struktur.

## 4. Analisis dan Pengolahan Data

Analisis dan pengolahan data dilakukan berdasarkan data-data yang dibutuhkan, selanjutnya dikelompokkan sesuai identifikasi tujuan permasalahan, sehingga diperoleh analisis pemecahan yang efektif dan terarah.

## 5. Pemecahan Masalah

Apabila hasil-hasil dari analisis dan pengolahan data sudah didapat, maka tahap pemecahan masalah bisa dilaksanakan, dengan tujuan mengetahui sejauhmana konstruksi yang sebenarnya di lapangan dan diproyeksikan terhadap kondisi riil berdasarkan peraturan-peraturan yang telah ditetapkan sebelumnya.



## 1.7 Sistematika Penulisan

Untuk mengetahui gambaran ringkas mengenai isi laporan Tugas Akhir serta untuk mempermudah pemahaman penelitian yang dilakukan, maka laporan Tugas Akhir ini dibagi menjadi beberapa bab, antara lain:

### **BAB I : PENDAHULUAN**

Bab ini berisi pendahuluan yang menjelaskan latar belakang permasalahan, perumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian serta sistematika penulisan.

### **BAB II : LANDASAN TEORI**

Bab ini berisi tentang teori-teori yang mendukung penelitian yang dilakukan. Teori tersebut merupakan teori yang sudah ada dari penelitian sebelumnya. Pengambilan teori tersebut dapat berasal dari buku, jurnal, maupun materi dari internet.

### **BAB III : METODE PENELITIAN**

Bab ini berisi perancangan sistem tentang penelitian yang akan dibuat. Yaitu pembuatan sistem klasifikasi kendaraan dalam kepentingan jalan tol

### **BAB IV : HASIL DAN PEMBAHASAN**

Bab ini berisi tentang analisis sistem baik dari segi kebutuhan *hardware* dan *software*, serta implementasi prototype dan pengujian prototype.

### **BAB V : KESIMPULAN DAN SARAN**

Bab ini menjelaskan tentang kesimpulan hasil uji coba yang diperoleh setelah menyelesaikan penelitian dan saran yang dapat digunakan sebagai masukan dalam pengembangan penelitian selanjutnya.

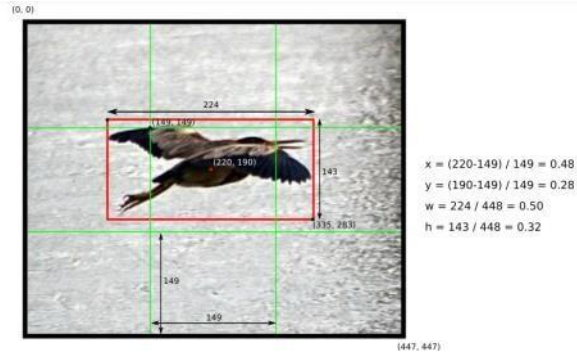
## BAB II

### LANDASAN TEORI

#### 2.1 *You Only Look Once (YOLO)*

YOLO akan membagi suatu gambar yang dimasukkan menjadi suatu *grid* berukuran  $S \times S$ , dimana nilai  $S$  adalah 7 dengan *input* gambar berukuran  $250 \times 450$ . Untuk mendapatkan *bounding box*, akan dilakukan konvolusi dari masukan gambar, sehingga hasil akhirnya akan 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* akan dilakukan normalisasi sehingga nilainya menjadi antara 0 hingga 1. Koordinat  $x$  dan  $y$  akan dinormalisasi menyesuaikan titik kiri atas dan *grid* yang bersangkutan. Dan tinggi dan lebar akan dinormalisasi sesuai dengan ukuran gambar. Nilai koordinat  $x$  dan  $y$  pada sebuah *bounding box* pada setiap *grid* merupakan titik tengah *grid* yang bersangkutan, ilustrasinya dapat dilihat pada Gambar 2.1



Gambar 2.1 YOLO

## 2.2 *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 bisa dilakukan oleh kamera atau alat lain yang bisa digunakan untuk mentransfer gambar. Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra akan 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.

### 2.3 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.



Gambar 2.2 Python Logo

Nama *Python* dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi *Monty Pythons Flying Circus*. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna *Python*.

Sisi utama yang membedakan *Python* dengan bahasa lain adalah dalam hal aturan penulisan kode program. Bagi para programmer di luar *Python* siap-siap dibingungkan dengan aturan indentasi, tipe data, *tuple*, dan *dictionary*. *Python* memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa *programmer* menyukai *Python*. Selain itu *Python* merupakan salah satu produk yang *opensource*, *free*, dan *multiplatform*.

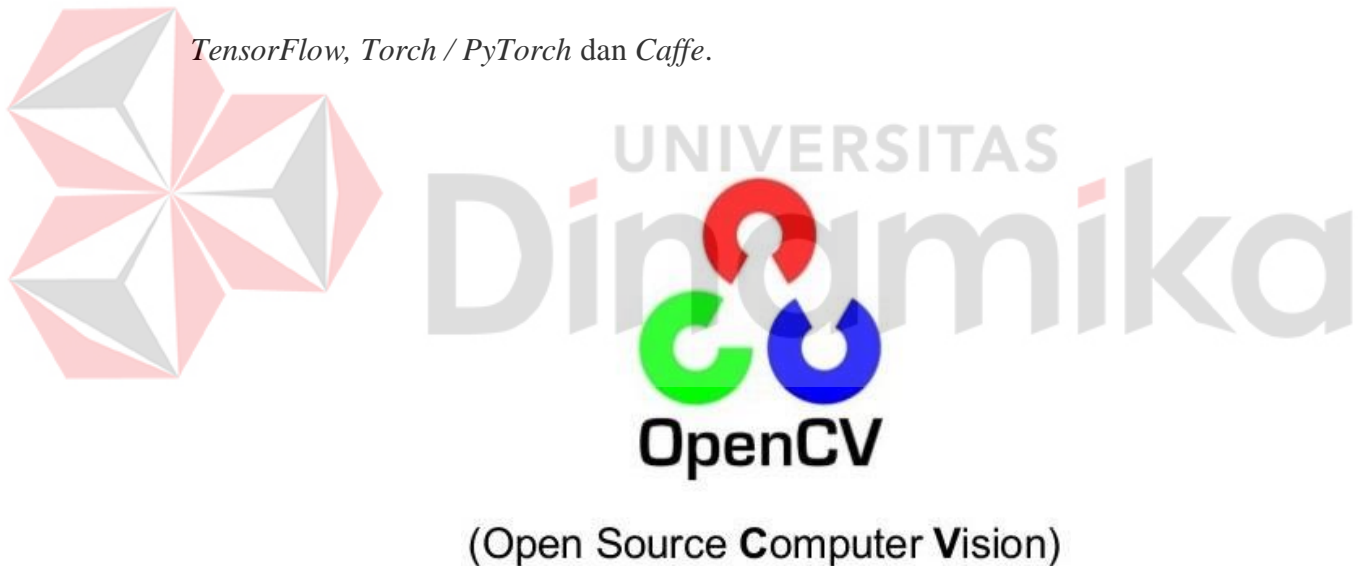
Beberapa fitur yang dimiliki *Python* adalah:

- a) Memiliki kepustakaan yang luas; dalam distribusi *Python* telah disediakan modul-modul siap pakai untuk berbagai keperluan.
- b) Memiliki tata bahasa yang jernih dan mudah dipelajari.
- c) Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber.
- d) Berorientasi obyek.
- e) Memiliki sistem pengelolaan memori otomatis (*garbage collection*, seperti java).
- f) Modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa *Python* maupun C/C++.
- g) Memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman Java, *Python* memiliki fasilitas pengaturan penggunaan ingatan

komputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan komputer secara langsung.

#### 2.4 *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*.



Gambar 2.3 *OpenCV*

*OpenCV (Open Source Computer Vision Library)* adalah *open source* visi komputer dan perpustakaan perangkat lunak pembelajaran mesin. *OpenCV* dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi alat berat pada produk komersial.



Menjadi produk berlisensi BSD, *OpenCV* memudahkan bisnis untuk memanfaatkan dan memodifikasi kode.

Memiliki *library* lebih dari 2500 algoritma yang dioptimalkan, yang mencakup serangkaian komprehensif visi komputer dan algoritma mesin dan pembelajaran mesin yang mutakhir. Algoritma ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak gerakan kamera, melacak objek bergerak, mengekstraksi model objek 3D, menghasilkan 3D point *clouds* dari kamera stereo, menjahit gambar bersama untuk menghasilkan resolusi tinggi gambar seluruh adegan, menemukan gambar serupa dari basis data gambar, menghapus *red eye* dari gambar yang diambil menggunakan *flash*, mengikuti gerakan mata, mengenali pemandangan dan membuat *marker* untuk menambahkannya dengan *augmented reality*, dan sebagainya. *OpenCV* memiliki lebih dari 47 ribu orang pengguna/komunitas dan perkiraan jumlah unduhan melebihi 18 juta. *Library* digunakan secara luas di perusahaan, kelompok penelitian dan oleh badan pemerintah.

## 2.5 Jenis Mobil

Di Indonesia terdapat beberapa jenis mobil yang melintas pada jalan tol. Sehingga pada Jalan tol menjadi golongan untuk menentukan tariff tol. Pada penelitian ini penulis mendeteksi 4 jenis mobil saja yaitu sedan, MPV, Bus dan *Truck*. Jenis mobil ini dapat dilihat pada Sub Bab berikut.

### 2.2.1 Mobil Sedan

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.



Gambar 2.4 Mobil Sedan

### 2.2.2 Mobil MPV

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 2.5.



Gambar 2.5 Mobil MPV

### 2.2.3 Mobil BUS

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 2.6.



Gambar 2.6 Mobil BUS

### 2.2.4 Mobil Truck

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 2.7 *Truck*

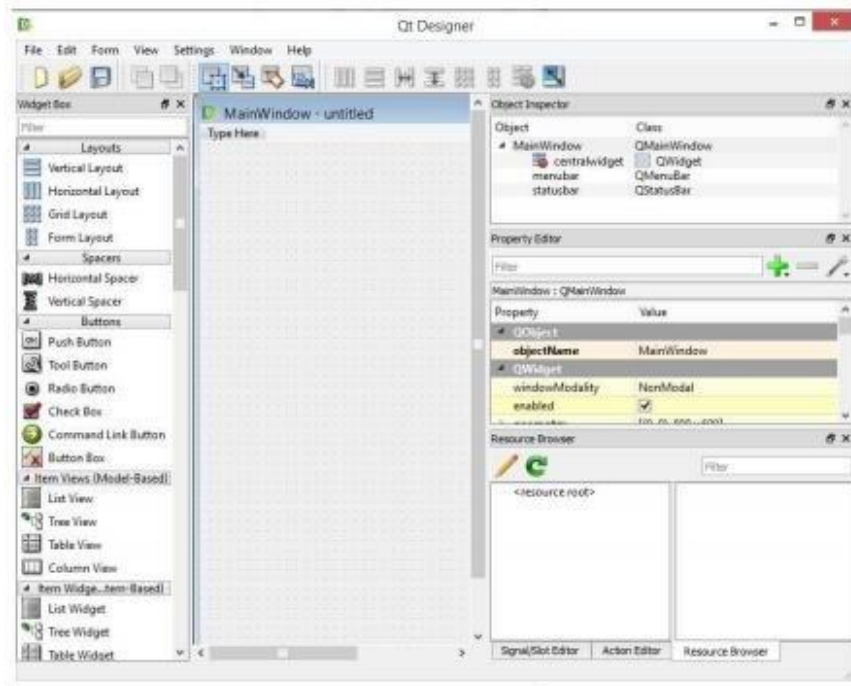
## 2.6 *Confusion Matrix*

*Confusion matrix* adalah sebuah tabel yang menunjukkan kinerja dari sebuah model klasifikasi yang memiliki data jawaban benar. Dari tabel yang didapatkan, untuk model klasifikasi yang dimiliki dapat dihitung akurasi, presisi, F-Score dan masih banyak lagi variabel yang dapat dihitung berdasarkan kondisi data yang diprediksi atau diklasifikasikan. Isi dari tabel confusion matrix ada 4, yaitu

1. *True Positive (TP)*, kondisi dimana model mengklasifikasikan data sebagai ya (TRUE) dan jawaban aktualnya adalah ya (TRUE),
2. *True Negative (TN)*, kondisi dimana model mengklasifikasikan data sebagai tidak (FALSE) dan jawaban aktualnya adalah tidak (FALSE),
3. *False Positive (FP)*, kondisi dimana model mengklasifikasikan data sebagai ya (TRUE) dan jawaban aktualnya adalah tidak (FALSE), dan
4. *False Negative (FN)*, kondisi dimana model mengklasifikasikan data sebagai tidak (FALSE) dan jawaban aktualnya adalah ya (TRUE).

## 2.7 PyQT

PyQt adalah binding untuk aplikasi Qt buatan Digia yang dapat digunakan pada berbagai sistem operasi, termasuk Windows, MacOS, dan Linux. Kelas pada Qt menggunakan mekanisme sinyal komunikasi antar objek yang aman namun berpasangan secara longgar, sehingga komponen piranti lunak yang dapat digunakan berulang-ulang dapat dengan mudah dibuat. Qt juga mencakup Qt Designer, yaitu sebuah aplikasi yang untuk mendesain antar muka grafis. Qt Designer memungkinkan pembuatan widget, dialog, dan main window lengkap dengan hanya melakukan drag-and-drop pada form. Dengan menggunakan Qt Designer, tiap perubahan pada halaman yang telah dibuat dapat dilihat secara langsung. Berkas yang dibuat dalam Qt Designer dapat diubah menjadi berkas Python. Kontrol terhadap antar muka grafis yang dibuat dapat ditambahkan dalam bentuk kode Python. Hal ini menjadi suatu kelebihan dari PyQt, dimana kemudahan desain yang disediakan oleh Qt dapat dimanipulasi dengan bahasa Python yang cenderung mudah dipelajari. Tampilan Qt Designer dapat dilihat pada Gambar dibawah ini.



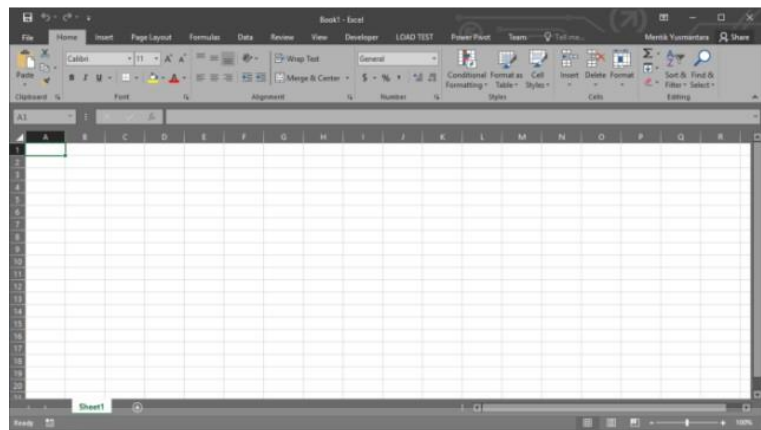
Gambar 2.8 Tampilan Qt Designer

## 2.8 Microsoft Excel

Pengertian Microsoft Excel adalah sebuah program atau aplikasi yang merupakan bagian dari paket instalasi Microsoft Office, berfungsi untuk mengolah angka menggunakan spreadsheet yang terdiri dari baris dan kolom untuk mengeksekusi perintah. Microsoft Excel telah menjadi *software* pengolah data / angka terbaik di dunia, selain itu Microsoft Excel telah didistribusikan secara multi-platform. Microsoft Excel tidak hanya tersedia dalam *platform* Windows, Microsoft Excel juga tersedia di MacOS, Android dan Apple.

Microsoft Excel secara fundamental menggunakan *spreadsheet* untuk manajemen data serta melakukan fungsi-fungsi Excel yang lebih dikenal dengan formula Excel. Excel merupakan program *spreadsheet* elektronik. *Spreadsheet* adalah kumpulan dari sel yang terdiri atas baris dan kolom tempat memasukkan

angka pada Microsoft Excel. Jumlah sel Microsoft Excel 2016 terdiri dari 1.048.576 Baris dan 16.384 Kolom atau 17.179.869.184 Sel. Tampilan Microsoft Excel dapat dilihat pada Gambar 2.9 dibawah ini. Pada penelitian ini Microsoft Excel digunakan untuk menyimpan database dari hasil proses penghitungan jenis kendaraan.



Gambar 2.9 Microsoft Excel



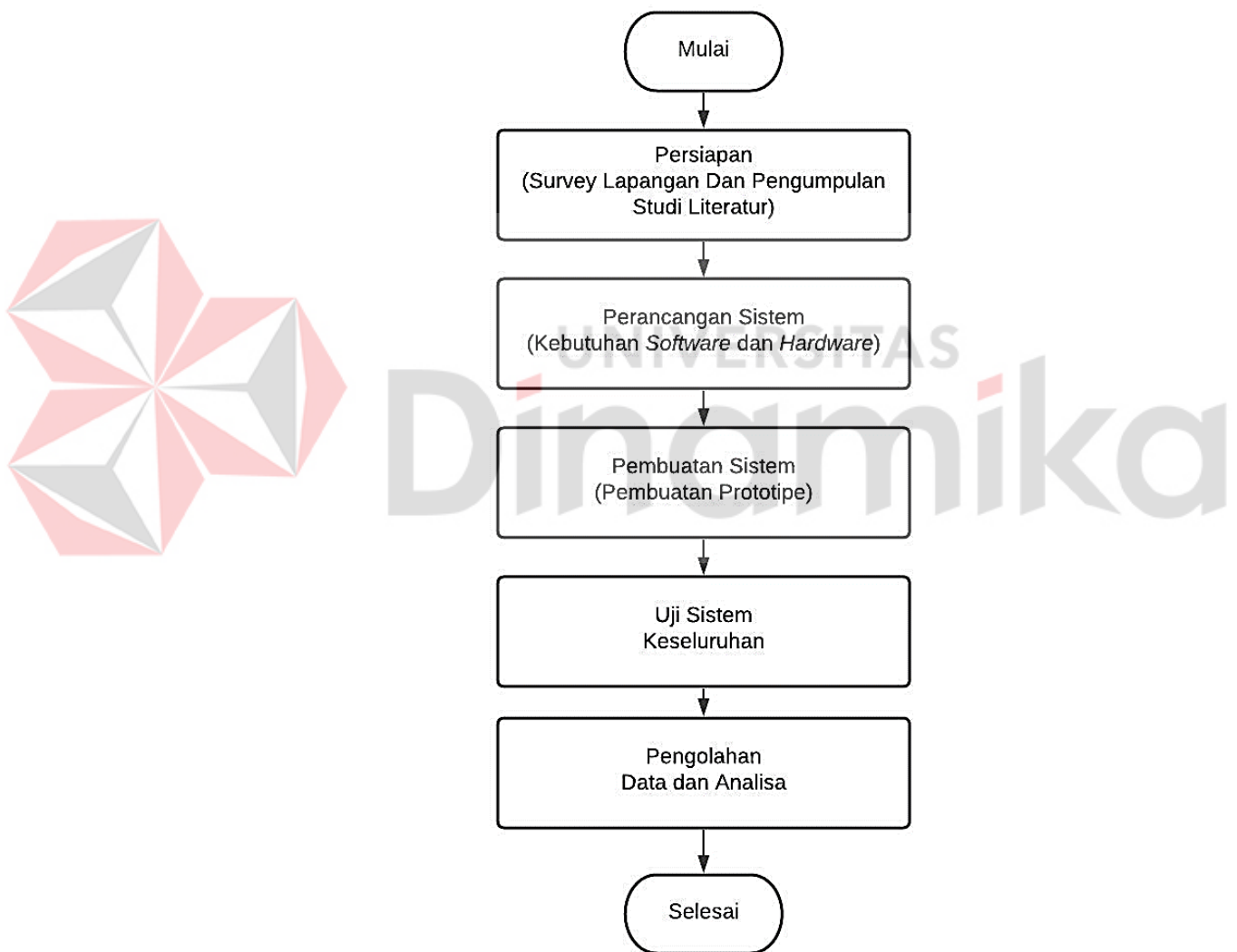
## 2.9 Citra

*Image* atau disebut citra adalah sebutan untuk gambar yang diambil, di-copy dan disimpan pada perangkat elektronik. Sebuah *image* memiliki banyak jenis yang dikenal sebagai ekstensi *image*. Berdasarkan ekstensi tersebut sebuah *image* memiliki cara penyimpanan nilai-nilai yang merepresentasikan gambar *image* pada layar perangkat elektronik. Dalam penelitian ini jenis ekstensi yang akan digunakan adalah JPEG (Joint Photographic Experts Group). Jenis ekstensi JPEG sebuah *file* grafik *image* yang dikembangkan oleh sebuah grup atau organisasi ahli dengan tujuan mengcompress algoritma untuk keperluan komputasi image pada komputer. *File Image* dengan jenis ekstensi JPEG biasanya berakhiran .jpg.

### BAB III

## METODOLOGI PENELITIAN

Dalam penelitian ini ada beberapa tahapan yang harus dilakukan dalam proses membuat sistem. Berikut ini adalah *flowchart* dari metodologi penelitian sebagaimana yang ditunjukkan pada Gambar 3.1 dibawah.



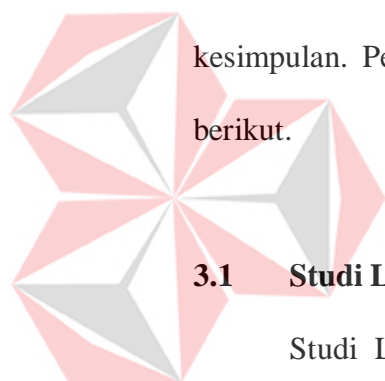
Gambar 3.1 Metodologi Penelitian



Dari Gambar 3.1 dapat dijelaskan bahwa pengerjaan penelitian ini diawali dengan tahapan persiapan yaitu tahapan pencarian data lapangan dengan melakukan *survey* secara langsung dan pengumpulan data melalui studi literatur. Tahapan kedua setelah semua data persiapan di dapatkan adalah tahap perancangan sistem yaitu tahap yang berisikan perancangan kebutuhan *software* dan *hardware*. Tahapan berikutnya adalah tahapan pembuatan sistem yaitu tahapan inti dimana penulis mulai membuat prototipe dan melakukan pemrograman *software*. Setelah prototipe dan *software* tercipta langkah selanjutnya adalah pengujian sistem secara keseluruhan mulai dari percobaan skala kecil hingga skala besar. Terakhir adalah pengolahan data hasil dari pengujian dan ditutup dengan analisis untuk mendapatkan kesimpulan. Penjelasan pada Metodologi Penelitian dapat dilihat pada Sub bab berikut.

### 3.1 Studi Literatur

Studi Literatur ini berfungsi untuk mempelajari teori penunjang yang mendukung dalam pengerjaan tugas akhir, seperti pemilihan komponen yang akan digunakan, penggunaan *software* pendukung, serta pengintegrasian seluruh sistem. Sehingga penulis akan lebih paham mengenai bagaimana cara mengerjakan dan dari mana kah penulis harus memulai. Sumber Literatur tidak hanya berasal dari Buku, namun terdapat dalam bentuk *Journal*, *Paper*, Referensi dari *web internet* dan lain sebagainya.

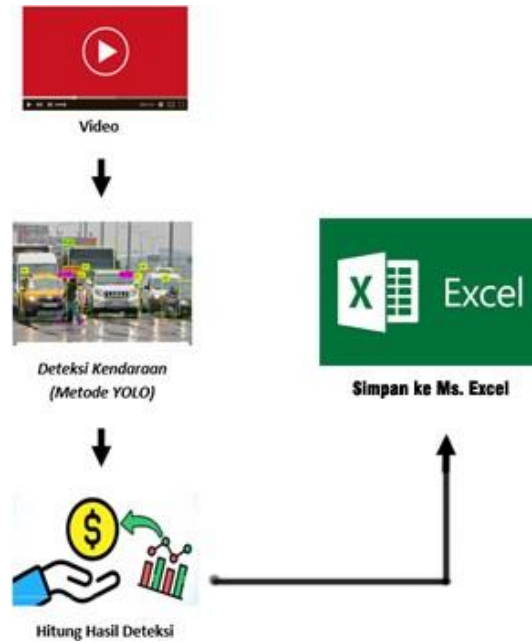


## 3.2 Perancangan Sistem

Perancangan merupakan gambaran langkah-langkah dari pembuatan aplikasi. Sedangkan pembuatan adalah tindakan dari perencanaan tersebut. Sehingga perancangan dan sistematika akan memberikan kemudahan dalam pembuatan aplikasi

### 3.2.1 Blok Diagram Sistem

Pada Tugas Akhir ini perancangan program dilakukan dengan lebih berfokus kepada sistem dari sebuah proses dibandingkan dengan perancangan terhadap perangkat keras. Dalam proses pembuatan aplikasi ini *input* utama dalam memulai menjalankan aplikasi ini adalah sebuah citra video terkait kondisi lalu lintas dengan objek kendaraan yang beragam atau jenis kendaraan pada umumnya yang terdapat di jalan raya. Citra ini akan diproses oleh algoritma YOLO dengan melakukan sebuah deteksi berdasarkan dengan data dari *dataset* yang telah dibuat dan di *training* sebelumnya. Hasil dari *training dataset* ini akan melakukan sebuah proses algoritma untuk menentukan tingkat kesesuaian objek dari *dataset* dengan objek dari video melalui sebuah prediksi geometri.

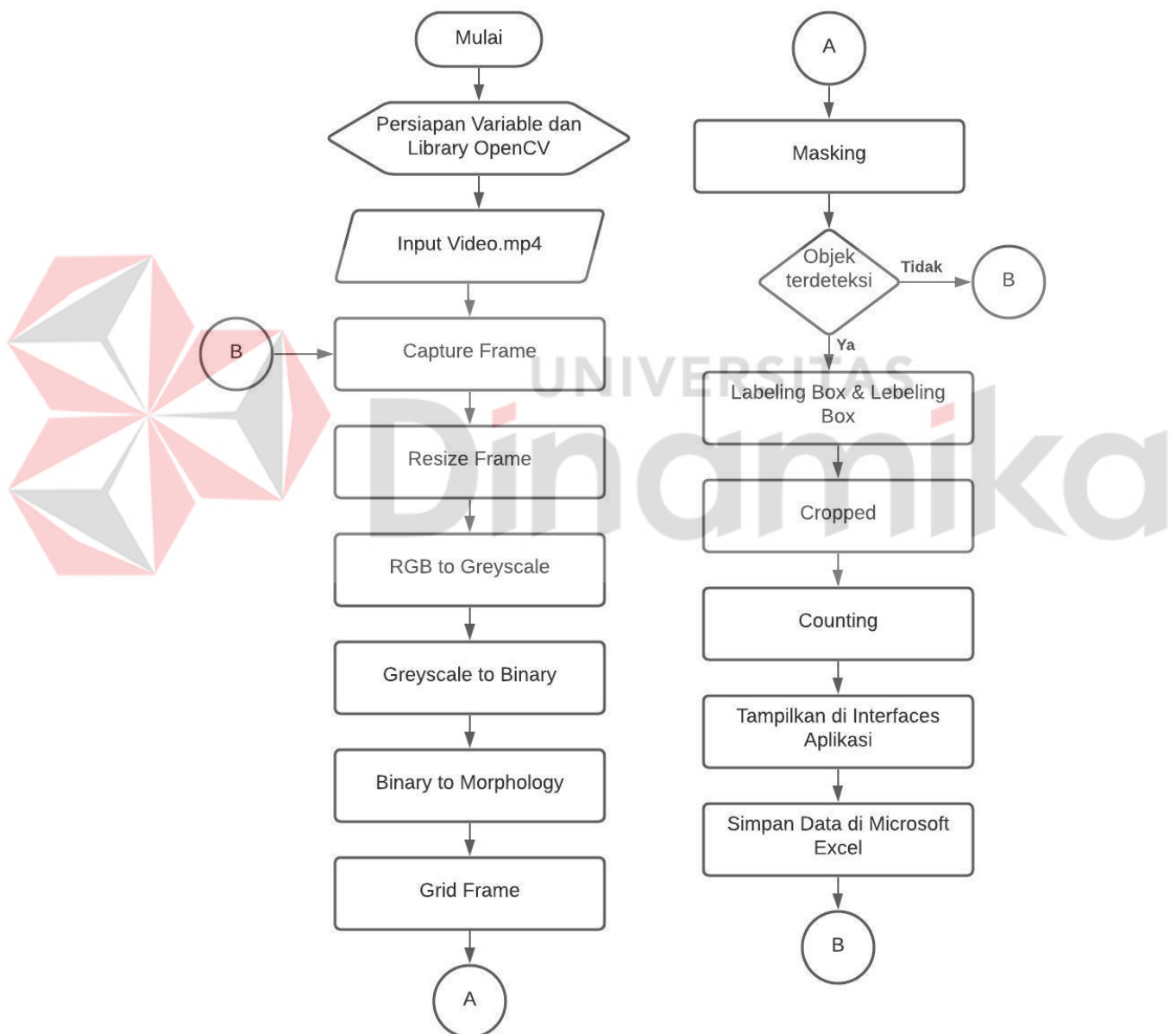


Gambar 3.2 Blok diagram aplikasi penghitung kendaraan

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

### 3.2.2 Flowchart Sistem

*Flowchart* sistem adalah alur dari sebuah pemrograman dimana pemrograman mempunyai *library* pada sistem yang digunakan. *Flowchart* dapat dilihat pada Gambar 3.3 dibawah ini.



Gambar 3.3 *Flowchart* sistem

Pada Gambar 3.3 terdapat persiapan *variable* 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* dirubah dalam bentuk Greyscale kemudian diproses untuk mendapatkan citra binary. Hasil dari binary tersebut di proses dengan morfologi untuk menghilangkan noise kemudian hasil dari proses tersebut masuk diproses lagi dengan *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 akan dilakukan *capture frame* ulang. Hasil dari *bounding box* dan labeling ditampilkan pada aplikasi desktop dan disimpan pada Microsoft excel. Penjelasan pada Proses Pre-Processing seperti pada alur Flowchart tersebut dapat dilihat pada Sub Bab berikut.

### **3.3 Proses Preprocessing**

Pada proses preprocessing adalah proses mengkonversi citra dari pembacaan *frame* hingga mendapatkan hasil berupa klasifikasi dan *counting* kendaraan. Preprocessing dapat dijelaskan pada sub bab berikut.

#### **3.3.1 Capture Frame**

*Capture Frame* didapatkan dari input video dengan format mp4 dimana untuk memproses frame dapat di capture frame satu per satu, hasil dari capture

frame adalah berupa RGB. RGB mempunyai tiga ruang warna sehingga harus diproses dengan fitur selanjutnya untuk menjadikan satu ruang warna. Hasil *Capture Frame* dapat dilihat pada Gambar 3.4.



Gambar 3.4 *Capture Frame* (RGB)

### 3.3.2 RGB to Greyscale

RGB to Greyscale adalah proses merubah tiga ruang warna menjadi satu ruang warna sehingga citra yang awalnya mempunyai warna akan berubah menjadi citra abu-abu. Proses konversi dapat menggunakan rumus:

$$\text{Greyscale} = 0.333R + 0.333G + 0.333B \quad (3.1)$$

Pada rumus 3.1 tersebut dapat diaplikasikan untuk mengkonversi citra menjadi greyscale yang dapat dilihat pada Gambar 3.5.



Gambar 3.5 Citra *Greyscale*

### 3.3.3 *Greyscale to Binary*

*Greyscale to Binary* adalah proses merubah citra *greyscale* ke dalam citra *binary*. Citra *binary* adalah citra yang hanya memiliki nilai 1 dan 0. Untuk mengonversi citra *greyscale* ke citra *binary* dapat menggunakan rumus

$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases} \quad (3.2)$$

Pada rumus 3.2 tersebut terdapat variable T yang artinya adalah *Threshold* dimana *Threshold* dapat menggunakan 127. Nilai 127 adalah nilai tengah dari pixel dimana pixel mempunyai *range* nilai 0-255 sehingga dapat diambil nilai tengah yaitu 127, hasil dari konversi dapat dilihat pada Gambar 3.6.



Gambar 3.6 Citra *Binary*

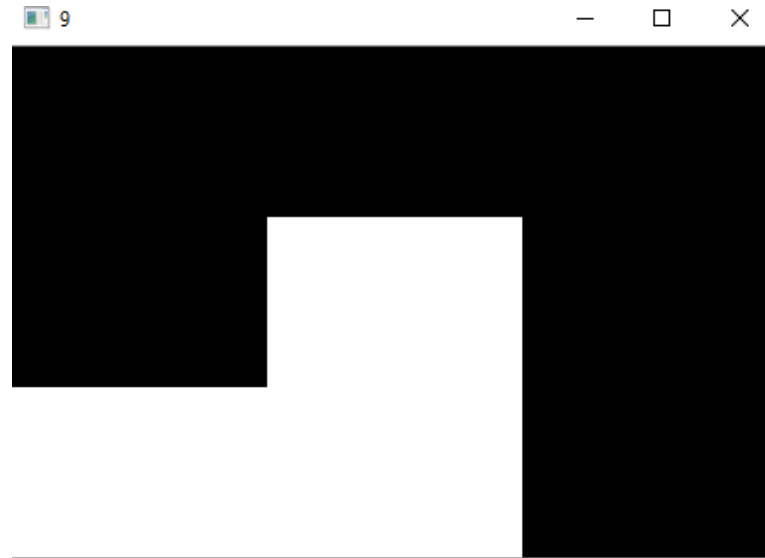
### 3.3.4 *Grid Frame*

*Grid Frame* adalah proses untuk menghilangkan *background* pada citra. Proses menghilangkan *background* dapat menggunakan *grid* 3x3 sampai objek mobil terdeteksi. *Grid* tersebut dapat dilakukan secara berulang hingga objek dapat terdeteksi dan dapat diproses kedalam proses selanjutnya. Proses *grid frame* ini dapat menentukan proses deteksi dari objek. Hasil *grid* tersebut dapat di-*morphology* sehingga dapat mendeteksi objek yang terdeteksi dan *background* yang tidak diperlukan.

### 3.3.5 *Morphology*

Pada proses *Morphology* adalah proses untuk merata-rata *pixel* sehingga hasil dari *grid* dapat diproses untuk mendapatkan *pixel* yang berisi objek. Hasil dari *morphology* dapat dilihat pada Gambar 3.7.

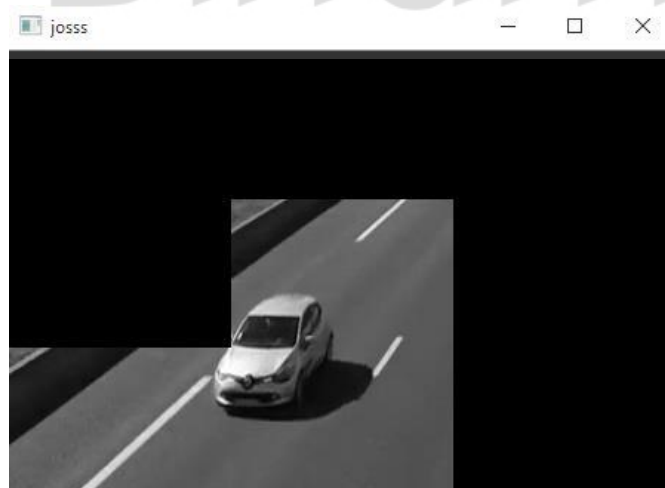




Gambar 3.7 Hasil *grid* dan *morphology*

### 3.3.6 Masking

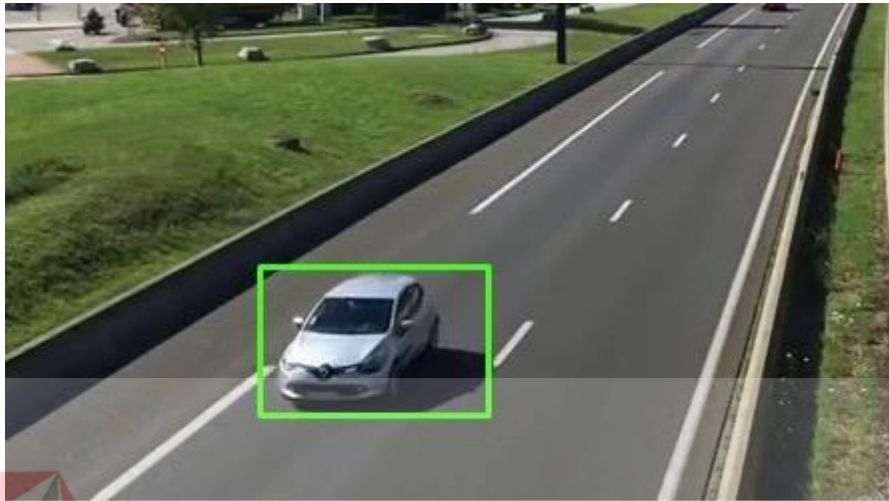
Proses *masking* adalah proses untuk menampilkan hasil deteksi dari *grid* ke dalam citra yang berwarna sehingga dapat melanjutkan ke *bounding box* dan *labeling*. Proses *masking* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Hasil *masking*

### 3.3.7 Bounding Box dan Labeling

Hasil dari proses *masking* dapat dilanjutkan dengan proses *bounding box* dan *labeling* sehingga objek dapat diperjelas dengan kotak hijau. Hasil dari *bounding box* dan *labeling* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Hasil *bounding box*

### 3.3.8 Cropped

Hasil dari *bounding box* dapat di *cropped* sehingga hasil dapat ditampilkan dalam *interfaces* aplikasi. Proses ini digunakan untuk mempercantik *interface* aplikasi. Sehingga dapat memudahkan pengguna dalam melihat proses lebih detail. Hasil dari *cropped* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Hasil *cropped*

### 3.3.9 Counting

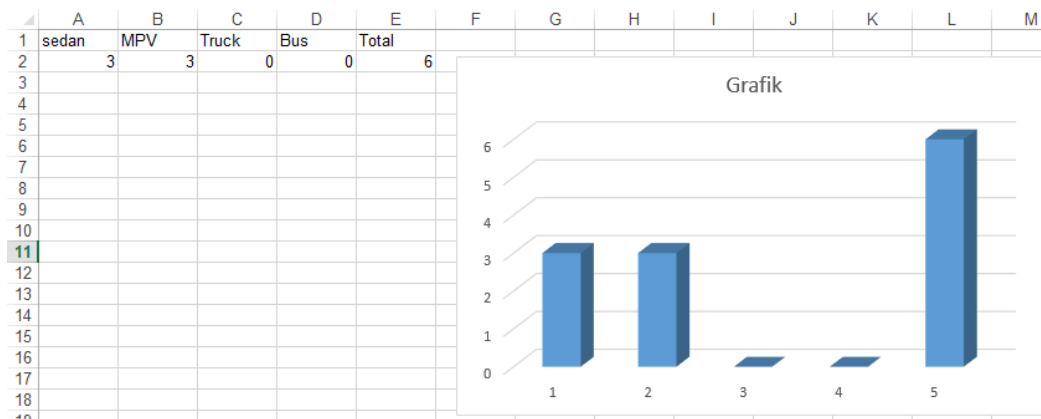
*Counting* pada sistem ini adalah digunakan untuk menghitung kendaraan setelah melalui proses klasifikasi. Proses dari *counting* hanya menggunakan penjumlahan setelah klasifikasi terdeteksi. Hasil dari *counting* dapat dilihat pada Gambar 3.11.

<b>JUMLAH SEDAN</b>	<b>:</b>	<b>3</b>
<b>JUMLAH MPV</b>	<b>:</b>	<b>3</b>
<b>JUMLAH BUS</b>	<b>:</b>	<b>0</b>
<b>JUMLAH TRUCK</b>	<b>:</b>	<b>0</b>
<hr/>		
<b>TOTAL</b>	<b>:</b>	<b>6</b>

Gambar 3.11 Hasil *counting*

### 3.3.10 Menyimpan data di Microsoft Excel

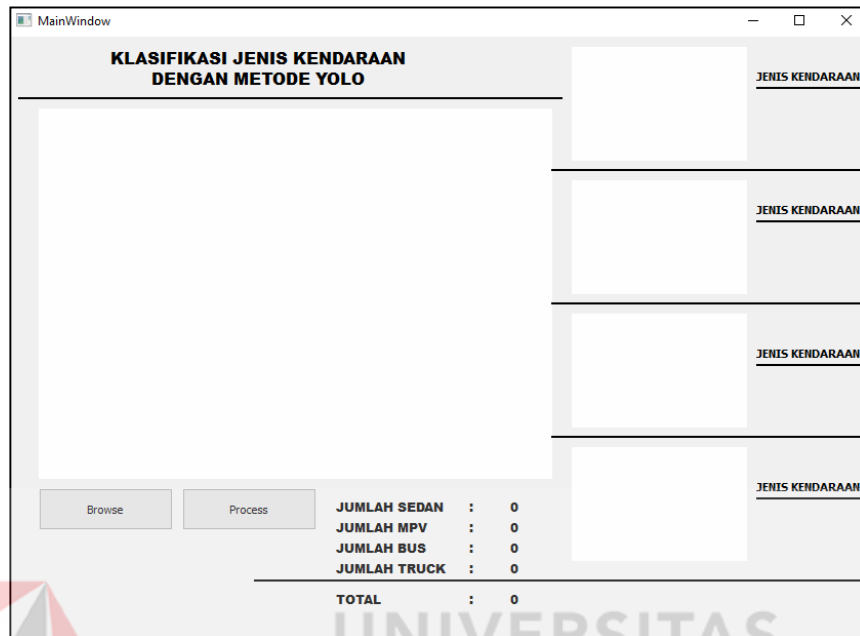
Hasil dari *counting* dapat disimpan dalam Microsoft Excel dimana dapat mempermudah proses pengembangan dalam sistem ini. Contohnya dapat dihitung dengan harga atau dapat ditampilkan dalam grafik bulanan atau tahunan. Hasil Penyimpanan Microsoft Excel dapat dilihat pada Gambar 3.12.



Gambar 3.12 Hasil Microsoft Excel

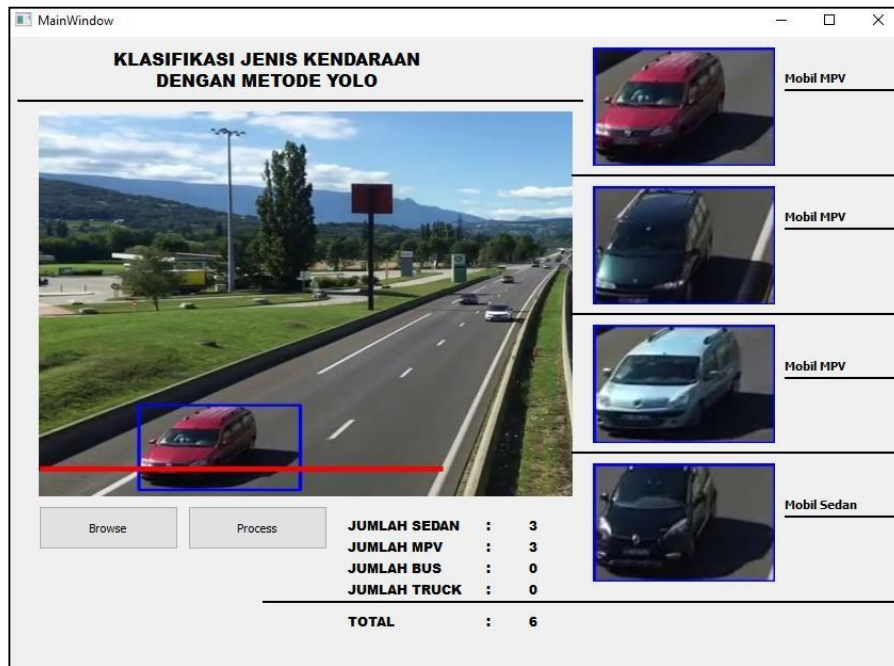
### 3.4 Desain Aplikasi

Pada Desain aplikasi ini adalah purwarupa dari aplikasi *desktop* yang telah dibuat oleh penulis. Desain aplikasi dapat dilihat pada Gambar dibawah 3.13.



Gambar 3.13 Desain Aplikasi

Pada Gambar 3.13 tersebut adalah desain aplikasi dengan terdapat tombol *Browse* dan *Proses* serta terdapat frame utama untuk menampilkan video. Sedangkan pada bagian kanan terdapat 4 *frame* digunakan untuk menampilkan mobil yang terdeteksi. Hasil pada pengolahan desain tersebut dapat dilihat pada Gambar dibawah 3.14.



Gambar 3.14 Hasil pengolahan aplikasi

### 3.5 Pengujian Sistem

Evaluasi dan pengujian dilakukan untuk mendapatkan data dan kesimpulan terhadap sistem yang diterapkan pada Tugas Akhir ini. Adapun beberapa proses pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian *Input* dan *Output* Aplikasi.
2. Pengujian YOLO.
3. Pengujian *Counting Information*.
4. Pengujian Klasifikasi Kendaraan.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Analisis Kebutuhan Perangkat Lunak (*Software*)

Fungsi dari perangkat lunak pada sistem ini adalah digunakan untuk mendukung proses pemrograman. Kebutuhan ini meliputi kebutuhan perangkat lunak yang digunakan pada pembuatan Aplikasi. Adapun kebutuhan *Software* yang diperlukan untuk merancang aplikasi ini adalah sebagai berikut:

1. Sistem Operasi Windows 10 Pro

Sistem operasi ini digunakan menghubungkan antara aplikasi *desktop* dengan laptop/komputer. Sistem operasi yang digunakan adalah Windows 10 Pro dimana sistem operasi ini dapat mendukung berjalannya aplikasi yang dipakai.

2. Aplikasi Python 3.9.0

Aplikasi Python ini digunakan untuk memprogram sistem yang dibuat. Bahasa yang digunakan dalam aplikasi ini adalah Python, sedangkan versi yang dipakai adalah 3.9.0. Aplikasi Python digunakan dalam penelitian ini karena aplikasi ini bersifat *open source*.

3. *OpenCV*

*OpenCV* digunakan sebagai *library* untuk mendukung proses *image processing* sehingga formula atau rumus sudah tercantum pada *library* ini.

4. *QT Designer*

*QT Designer* digunakan untuk kepentingan GUI (*Graphic User Interface*) atau antarmuka pengguna aplikasi. Karena aplikasi Python sendiri tidak tersedia GUI sehingga pada penelitian ini menggunakan *QT Designer*.

## 5. Microsoft Excel

Microsoft Excel pada penelitian ini digunakan sebagai penyimpan data latih atau disebut juga dengan database. Aplikasi Ms Excel tersebut dapat dipanggil oleh program pada aplikasi python sehingga Ms Excel digunakan untuk mendukung sistem dari penulis.

## 6. Microsoft Word

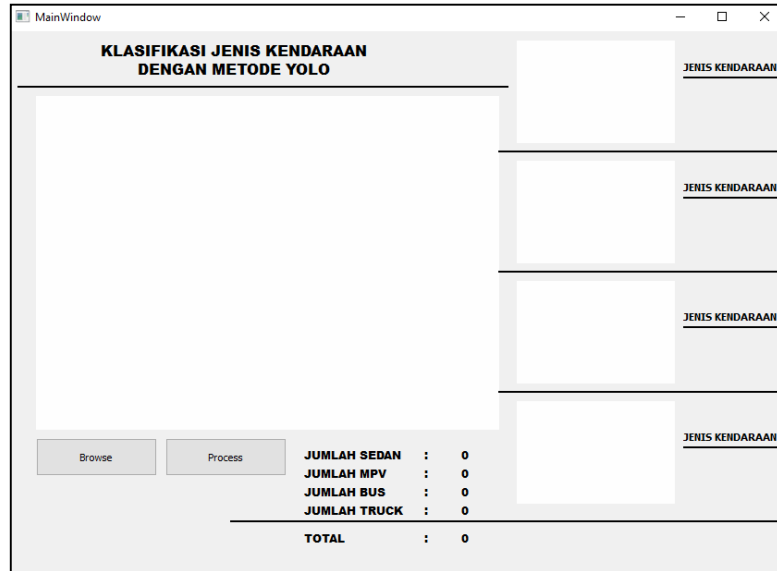
Microsoft Word pada penelitian ini digunakan untuk kepentingan pembuatan laporan penelitian. Sehingga dapat mendukung dalam penyelesaian penelitian ini

### 4.2 Analisis *Input* dan *Output* Sistem

Analisis *input* dan *output* merupakan implementasi dari desain antarmuka *input* dan *output* yang telah dibuat. Aplikasi identifikasi daun ini memiliki 1 menu utama yaitu form pengolah *image processing*.

#### 4.2.1 Form pengolah *Image processing*

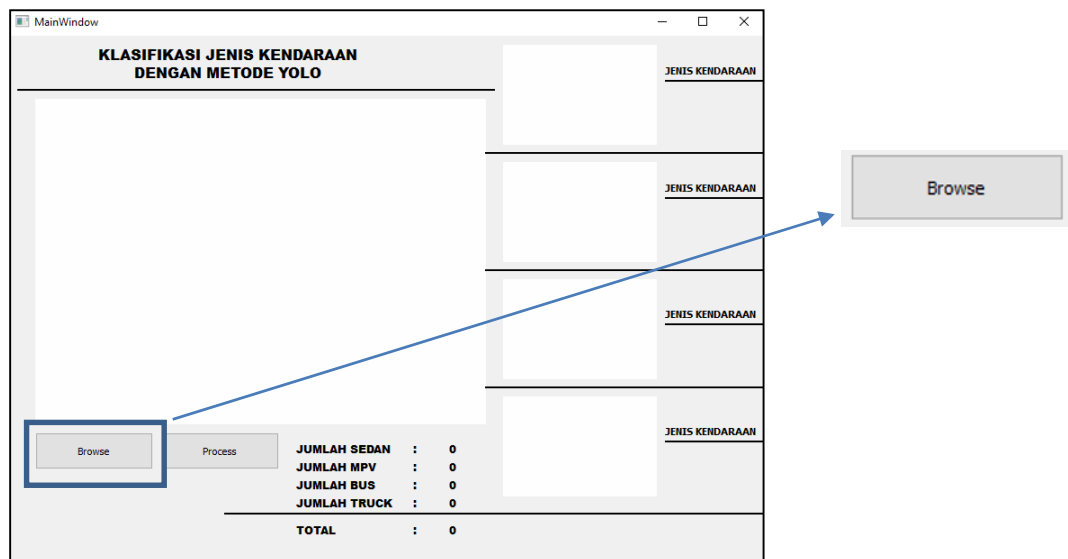
Pada *form* ini terdapat Tombol *Browse* dan Tombol *Process*. Sedangkan untuk *pre-processing* memiliki 4 *frame* untuk penampil hasil *pre-processing*. Untuk hasil ditampilkan dalam bentuk *text* pada bagian bawah aplikasi. *Form* pengolah *Image processing* dapat dilihat pada Gambar berikut 4.1.



Gambar 4.1 *Form* pengolah *image processing*

#### 4.2.2 Tombol *Browse*

Tombol *Browse* digunakan untuk memilih contoh gambar yang diuji pada aplikasi. Ketika ditekan tombol ini maka secara otomatis memunculkan jendela baru. Pengguna dapat memilih video yang diproses pada aplikasi tersebut. Tombol *browse* dapat dilihat pada Gambar dibawah 4.2 ini.

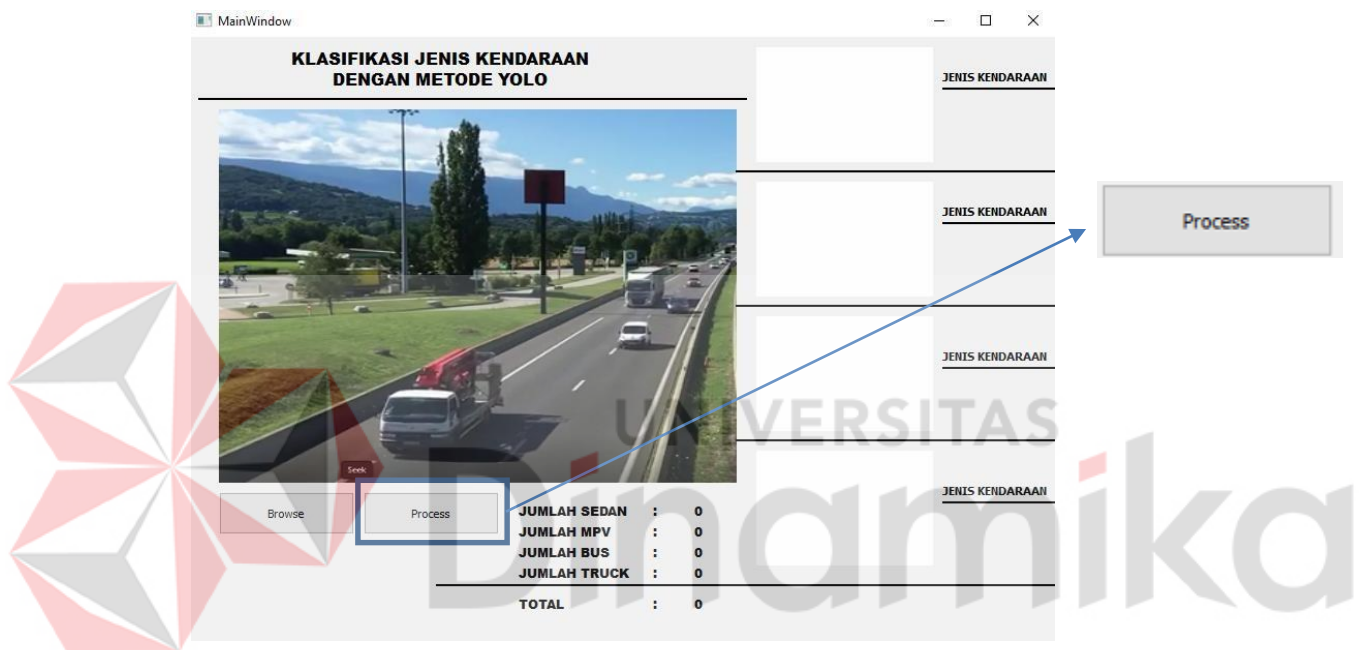


Gambar 4.2 Tombol *Browse*



### 4.2.3 Tombol Proses

Pada tombol proses adalah digunakan untuk memproses video yang telah dipilih oleh pengguna sehingga dapat mengetahui hasil dari *image processing* tersebut. Tombol proses memicu aplikasi untuk masuk pada deteksi kendaraan. Tombol proses dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 Tombol Proses

### 4.2.4 Browse

Pada *browse* digunakan untuk menampilkan hasil dari *image processing* yang telah diproses pada sistem ini. *Browse* terdapat 4 urutan kendaraan yang tertera lebeling pada setiap kendaraannya. *Browse* dapat dilihat pada Gambar 4.4 dibawah ini.

Gambar 4.4 *Browse*

#### 4.2.5 Counting Information

Pada *counting information* tersebut menggunakan tipe *text* sehingga hasil tersebut ditampilkan dalam bentuk teks. *Counting information* tersebut dimulai dengan jumlah pada masing-masing jenis mobil dan kemudian total semua mobil. Gambar dari *Counting information* dapat dilihat pada Gambar 4.5.

Gambar 4.5 *Counting information*

### 4.3 Hasil Pengujian Metode YOLO pada Kendaraan

Hasil pegujian metode YOLO pada kendaraan digunakan untuk mengetahui cara konsep dari metode YOLO itu sendiri. Langkah pertama adalah membuat *dataset* 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 akan 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 akan semakin baik. Sebelum proses training perlu untuk mengkonfigurasi proses konvolusi, yaitu metode CNN yang akan 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.

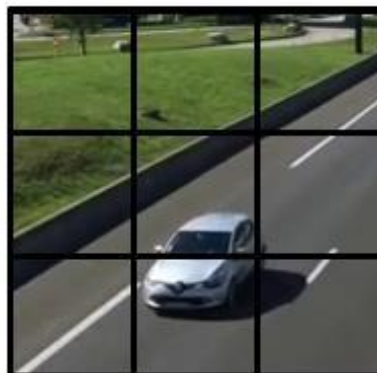
Konfigurasi diatas akan menghasilkan proses konvolusi sebagai berikut:

Tabel 4.1 Proses konvolusi

Layer	Size	Stride	Output
<i>Input</i>	-	-	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:

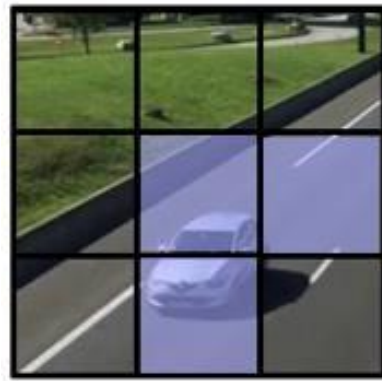
1. *Input* gambar akan diubah menjadi 416 x 416 kemudian dibagi menjadi *grid* sesuai konfigurasi yang ditentukan yaitu  $S \times S$  dimana  $S = 3$  seperti Gambar 4.6 berikut:



Gambar 4.6 *Grid* 3x 3

2. Kemudian setiap *grid* akan 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 4.7 dan Gambar 4.8 berikut:



Gambar 4.7 Hasil prediksi



Gambar 4.8 Hasil *bounding box*

Hasil prediksi *Bounding box* akan 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.

#### 4.4 Hasil Pengujian *Counting information*

Pada hasil pengujian *counting* ini digunakan untuk menentukan tingkat keberhasilan dan kelayakan aplikasi dalam menghitung kendaraan. Skenario dalam pengujian ini adalah dengan cara menguji aplikasi pada 1 video yang berdurasi 2 menit. Hasil pengujian dapat dilihat pada dan Tabel 4.1.

Tabel 4.2 Hasil pengujian pada video

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
<b>Total Kendaraan</b>		<b>136</b>	<b>145</b>	<b>9</b>

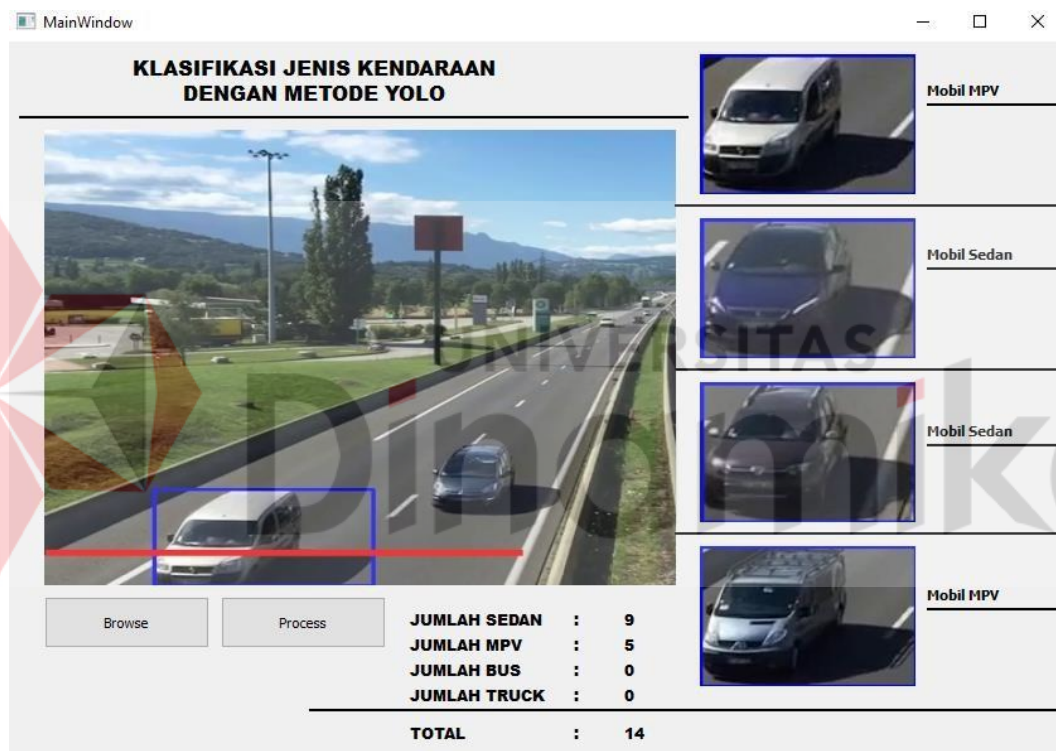
Pengambilan data pada Tabel 4.2 tersebut dengan menguji keakuratan penghitungan deteksi aplikasi dan penghitungan manual dapat disimpulkan bahwa masih terdapat banyak selisih atau disebut juga dengan *error*. Karena data keberhasilan sudah ditemukan maka dapat melakukan proses penghitungan keakuratan deteksi aplikasi dengan cara.

$$\begin{aligned}
 \text{Keakuratan Data} &= 100\% - (\text{selisih/perhitungan manual} \times 100\%) \\
 &= 93.8\%
 \end{aligned}$$

Pada keakuratan penghitung kendaraan memiliki tingkat keberhasilan yaitu 93%.

#### 4.5 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 kendaraan apa yang dideteksi oleh sistem. Pada pengujian kali ini memiliki skenario yaitu pengujian dengan menggunakan 1 video berdurasi 2 menit. Hasil dari pengujian klasifikasi kendaraan dapat dilihat pada Gambar 4.9.



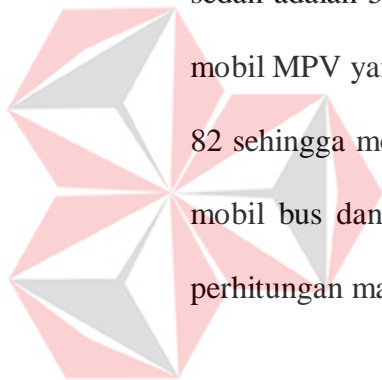
Gambar 4.9 Hasil pengujian klasifikasi kendaraan

Tabel 4.3 Hasil pengujian klasifikasi kendaraan

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

No	Detik	Jumlah Kendaraan Terdeteksi				Jumlah Kendaraan Manual			
		Sedan	MPV	Bus	Truck	Sedan	MPV	Bus	Truck
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
<b>Total</b>		<b>52</b>	<b>77</b>	<b>2</b>	<b>5</b>	<b>56</b>	<b>82</b>	<b>2</b>	<b>5</b>

Pada Tabel 4.3 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 bisa dideteksi adalah 77 sedangkan dalam kondisi manual adalah 82 sehingga mobil MPV memiliki selisih yaitu 5 mobil MPV, sedangkan untuk mobil bus dan *truck* tidak memiliki selisih karena proses deteksi sama dengan perhitungan manual.





## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil pengujian yang dilakukan pada penelitian ini, dapat diambil beberapa kesimpulan 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 bisa dideteksi adalah 77 sedangkan dalam kondisi manual adalah 82 sehingga mobil MPV memiliki selisih yaitu 5 mobil MPV, sedangkan untuk mobil bus dan *truck* tidak memiliki selisih karena proses deteksi sama dengan perhitungan manual.

#### 5.2 Saran

Berikut ini adalah saran yang dapat diberikan setelah melakukan perancangan sistem dan mendapatkan kesimpulan dari uji coba yang digunakan. Saran ini dibutuhkan untuk proses pengembangan sistem yang akan dikembangkan nanti. Saran dapat dilihat pada keterangan berikut:

1. Sangat baik jika *pixel* pada video lebih tinggi, sehingga dapat lebih mudah terdeteksi.

2. Lebih baik menggunakan kamera yang mempunyai *pixel* tinggi, sehingga dapat melakukan *live stream*.



UNIVERSITAS  
**Dinamika**

## DAFTAR PUSTAKA

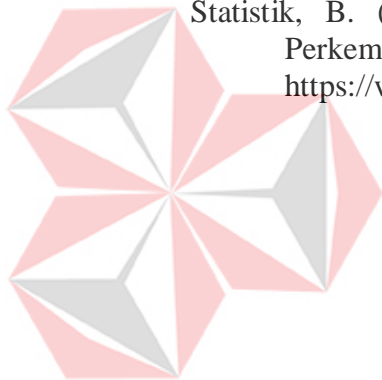
Directory, S. (2018 , Februari 27). *Web Application*. Retrieved from Desktop Applications Vs. Web Applications: <https://www.streetdirectory.com/>

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.

Statistik, B. (2018, Desember 16). *Badan Pusat Statistik*. Retrieved from Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 1949-2018: <https://www.bps.go.id/linkTableDinamis/view/id/1133>



UNIVERSITAS  
**Dinamika**