

**TESIS**

**SISTEM DETEKSI DAN PENGENALAN RAMBU LALU LINTAS DI  
INDONESIA MENGGUNAKAN ALGORITMA YOLOV4**

Disusun dan diajukan oleh

**REZKI WULANDARI ARIEF**

**D032181010**



**TEKNIK ELEKTRO  
FAKULTAS TEKNIK / SEKOLAH PASCASARJANA  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2021**

**SISTEM DETEKSI DAN PENGENALAN RAMBU LALU LINTAS DI  
INDONESIA MENGGUNAKAN ALGORITMA YOLOV4**

Tesis

Sebagai Salah Satu Syarat Untuk Mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun dan diajukan oleh

**REZKI WULANDARI ARIEF  
D032181010**

**Kepada**

**PROGRAM PASCASARJANA  
UNIVERSITAS HASANUDDIN  
MAKASSAR  
2021**

## LEMBAR PENGESAHAN TESIS

### SISTEM DETEKSI DAN PENGENALAN RAMBU LALU LINTAS DI INDONESIA MENGGUNAKAN ALGORITMA YOLOv4

Disusun dan diajukan oleh

**REZKI WULANDARI ARIEF**

**D032181010**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Magister Program Studi Teknik Elektro Fakultas Teknik Universitas Hasanuddin pada tanggal 19 Oktober dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

**Dr. Ir. Ingrid Nurtanio, M.T.**  
Nip. 196108131988112001

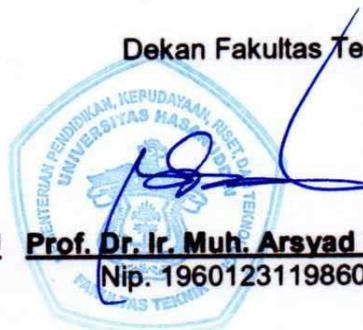
**Prof. Dr.-Ing. Faizal Arya Samman, S.T., M.T.**  
Nip. 197506052002121004

Ketua Program Studi,

Dekan Fakultas Teknik,



**Prof. Dr. Eng. Ir. Syafaruddin, S.T., M.Eng. IPU**  
Nip. 197405301999031003



**Prof. Dr. Ir. Muh. Arsyad Thaha, M.T.**  
Nip. 196012311986091001

## PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini:

Nama : Rezki Wulandari Arief  
Nomor Pokok : D032181010  
Program Studi : Teknik Elektro  
Konsentrasi : Elektro

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan tulisan atau pemikiran orang lain. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan tesis ini hasil karya orang lain, saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 6 Juni 2021

Yang menyatakan,



*Rezki Wulandari Arief*  
Rezki Wulandari Arief

## ABSTRAK

**Rezki Wulandari Arief.** Sistem Deteksi Dan Pengenalan Rambu Lalu Lintas Di Indonesia Menggunakan Algoritma YOLOv4. (dibimbing oleh Ingrid Nurtanio dan Faizal Arya Samman)

Rambu lalu lintas merupakan salah satu fasilitas perlengkapan jalan yang penting untuk menginformasikan kepada pengguna jalan tentang peraturan dan petunjuk secara visual. Saat ini sedang berkembang sistem otomatis *Traffic Sign Recognition* (TSR) yang diterapkan pada sistem pengemudi tingkat lanjut (ADAS) agar pengguna jalan dapat aman dan selamat saat di jalan raya. Oleh karena itu, makalah ini bertujuan untuk dapat mendeteksi dan mengenali rambu-rambu lalu lintas yang ada di jalan raya sehingga memberi informasi makna dari rambu lalu lintas tersebut secara otomatis. Pada penelitian ini menggunakan 35 kelas rambu yang terdiri dari rambu peringatan, larangan, perintah, dan petunjuk. Sistem ini diimplementasikan menggunakan *framework darknet* dengan model *You Only Look Once version 4* (YOLOv4). Investigasi yang dilakukan dalam penelitian ini adalah sistem yang mendeteksi dan mengenali rambu-rambu lalu lintas yang sudah dites secara *realtime*. Hasil mAP (*mean Average Precision*) pada sistem ini sebanyak 95.15%.

Kata kunci : Rambu Lalu Lintas, YOLOv4, Deteksi Objek

## **ABSTRACT**

**Rezki Wulandari Arief.** Traffic Sign Detection And Recognition System Using YOLOv4 Algorithm. (supervised by Ingrid Nurtanio and Faizal Arya Samman)

Traffic signs are one of the important road equipment facilities to inform road users about regulations and visual directions. Currently, an automatic Traffic Sign Recognition (TSR) system is being developed which is implemented in an advanced driver system (ADAS) so that road users can be safe and secure while on the road. Therefore, this paper aims to be able to detect and recognize traffic signs on the highway to provide information on the meaning of these traffic signs automatically. In this study, 35 classes of signs were used which consisted of warning signs, prohibitions signs, mandatory signs, and instructions signs. This system is implemented using the darknet framework with the You Only Look Once version 4 (YOLOv4) model. The investigation carried out in this study is a system that detects and recognizes traffic signs that have been tested in real time. The result of mAP (mean Average Precision) in this system is 95.15%.

Keywords : Traffic Signs, YOLOv4, Object detection

## KATA PENGANTAR

Segala puji selalu dipanjatkan kepada Allah SWT Yang Maha Kuasa yang telah memberikan rahmat, hidayah dan pertolongan-Nya dalam menyelesaikan tesis, yang berjudul **“SISTEM DETEKSI DAN PENGENALAN RAMBU LALU LINTAS DI INDONESIA MENGGUNAKAN ALGORITMA YOLOv4.”** Tak lupa pula shalawat dan salam kepada Nabi Muhammad SAW yang telah menyinari dunia ini dengan keindahan ilmu dan akhlak yang diajarkan kepada seluruh umatnya.

Ucapan terima kasih pun penulis hanturkan kepada kepada dosen pembimbing tesis Dr. Ir. Ingrid Nurtanio, M.T. dan Prof. Dr-Ing Faizal Arya Samman, S.T., M.T. yang telah meluangkan waktunya untuk membimbing dan berkonsultasi tentang materi dalam tesis ini dan juga kepada seluruh dosen dan staf Departemen Teknik Elektro, Universitas Hasanuddin yang telah membantu dalam hal keilmuan maupun administrasi pada tahap tesis ini. Penulis menyadari bahwa tesis ini masih belum sempurna. Dengan demikian, penulis tetap mengharapkan kritik dan saran dengan harapan tulisan ini bisa memberikan manfaat kepada seluruh pihak.

Makassar, 06 Juni 2021



Penulis

Rezki Wulandari Arief

## DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TESIS.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL .....	ix
DAFTAR GAMBAR .....	x
DAFTAR SINGKATAN .....	xi
BAB I PENDAHULUAN .....	1
A. LATAR BELAKANG .....	1
B. RUMUSAN MASALAH .....	4
C. TUJUAN PENELITIAN .....	5
D. MANFAAT PENELITIAN.....	5
E. BATASAN MASALAH.....	5
F. SISTEMATIKA PENULISAN.....	6
BAB II TINJAUAN PUSTAKA .....	8
A. LANDASAN TEORI .....	8
B. PENELITIAN TERKAIT .....	29
C. STATE OF THE ART.....	31
D. KERANGKA PIKIR .....	33
BAB III METODE PENELITIAN .....	35
A. TAHAPAN PENELITIAN.....	35
B. WAKTU DAN LOKASI PENELITIAN.....	36
C. INSTRUMEN PENELITIAN .....	37
D. SUMBER DATA .....	37
E. PERANCANGAN SISTEM.....	38
F. ANALISIS KERJA SISTEM.....	50
BAB IV HASIL DAN PEMBAHASAN.....	54
A. HASIL PENELITIAN .....	54
B. PEMBAHASAN.....	61
BAB V KESIMPULAN DAN SARAN .....	64
A. Kesimpulan .....	64
B. Saran .....	64
DAFTAR PUSTAKA .....	65

## DAFTAR TABEL

Tabel 1 State of The Art .....	31
Tabel 2. Jenis rambu-rambu yang digunakan pelabelan data.....	40
Tabel 3. Performa sistem deteksi dan pengenalan rambu lalu lintas .....	54
Tabel 4. Hasil Average Precision dan Confussion Matrix.....	56
Tabel 5. Kalkulasi mAP dan confussion matrix tiap video .....	58

## DAFTAR GAMBAR

Gambar 1. Ukuran Standar Rambu Peringatan .....	9
Gambar 2. Ukuran Standar Rambu Larangan .....	10
Gambar 3. Ukuran Standar Rambu Perintah .....	11
Gambar 4. Ukuran Standar Rambu Petunjuk .....	11
Gambar 5. Arsitektur YOLOv1 .....	15
Gambar 6. Darknet-53 .....	18
Gambar 7. Arsitektur YOLOv4 .....	20
Gambar 8. Ilustrasi DenseNet (a) dan CSP DensNet (b) .....	21
Gambar 9. Bounding Box .....	23
Gambar 10. Contoh Penggunaan Bounding Box .....	26
Gambar 11. Contoh Penggunaan Polygon Annotation .....	26
Gambar 12. Contoh Penggunaan Poin Annotation .....	27
Gambar 13. Contoh Penggunaan Semantic Annotation .....	28
Gambar 14. Tampilan <i>Notebook Google Colaboratory</i> .....	29
Gambar 15. Kerangka Pikir .....	33
Gambar 16. Diagram Tahapan Penelitian .....	35
Gambar 17. Desain Alur Sistem .....	39
Gambar 18. Flowchart proses labelling menggunakan LabelImg .....	41
Gambar 19. Pelabelan menggunakan LabelImg .....	42
Gambar 20. Perintah install requirement dan environment .....	43
Gambar 21. Perintah training .....	43
Gambar 22. Perintah preprocessing pada yolov4_custom.cfg .....	44
Gambar 23. Total layer CSP modul YOLOv4 .....	45
Gambar 24. SPP pada YOLOv4 .....	46
Gambar 25. Head YOLOv4 .....	47
Gambar 26. chart average loss-iterations .....	47
Gambar 27. Flowchart testing menggunakan YOLOv4 .....	48
Gambar 28. Contoh hasil deteksi .....	50
Gambar 29. Bagan Confussion Matrix .....	51
Gambar 30. Realtime deteksi rambu lalu lintas .....	59
Gambar 31. Kesalahan pada deteksi rambu lalu lintas .....	60

## DAFTAR SINGKATAN

ADAS	: <i>Advanced Driver Assistance System</i>
AI	: <i>Artificial Intelligent</i>
AP	: <i>Average Precision</i>
CNN	: <i>Convolutional Neural Network</i>
CPU	: <i>Central Processing Unit</i>
CSP	: <i>Cross Stage Partial</i>
CUDNN	: <i>Cuda Deep Neural Network</i>
FPN	: <i>Feature Pyramid Network</i>
GPU	: <i>Graphic Processing Unit</i>
HOG	: <i>Histogram Oriented Gradient</i>
IoU	: <i>Intersection over Union</i>
KTSD	: <i>Korean Traffic Sign Detection</i>
mAP	: <i>mean Average Precision</i>
OpenCV	: <i>Open source Computer Vision</i>
PaNet	: <i>Path Aggregation Network</i>
ResNet	: <i>Residual Neural Network</i>
SPP	: <i>Spatial Pyramid Pooling</i>
SVM	: <i>Support Vector Machine</i>
TSR	: <i>Technology Sign Recognition</i>
VGGNet	: <i>Visual Geometric Group Network</i>
YOLO	: <i>You Only Look Once</i>

# **BAB I**

## **PENDAHULUAN**

### **A. LATAR BELAKANG**

Fasilitas perlengkapan di jalan raya memberikan informasi kepada pengguna jalan tentang peraturan dan petunjuk yang diperlukan sehingga dapat meningkatkan keselamatan jalan dan menyediakan pergerakan yang teratur terhadap pengguna jalan agar mencapai arus lalu lintas yang selamat dan beroperasi dengan efisien. Fasilitas perlengkapan jalan yaitu terdiri dari rambu-rambu lalu lintas, alat pemberi isyarat lalu lintas, penerangan jalan, dan marka jalan (Yu et al., 2019). Khusus pada makalah ini berfokus untuk mendeteksi dan mengenali salah satu fasilitas perlengkapan jalan yaitu rambu-rambu lalu lintas.

Berdasarkan Keputusan Menteri Perhubungan No.61 Pasal 1 Tentang Rambu-Rambu Lalu Lintas, 1993, Rambu lalu lintas ialah berupa lambang, huruf, angka, kalimat, dan perpaduan diantaranya sebagai peringatan, larangan, perintah atau petunjuk bagi pemakai jalan (Minister of Transportation, 2014). Berdasarkan penjelasan tersebut, seharusnya rambu lalu lintas dapat memberikan informasi secara visual seperti mengemudi di jalur yang tepat, menghindari rintangan, dan mengetahui kondisi lalu lintas agar keselamatan pengemudi aman dan terhindar dari kecelakaan. Namun kenyataannya di Indonesia tidak semua pengguna

jalan memahami atau mengerti arti rambu-rambu lalu lintas tersebut karena rambu-rambu lalu lintas memiliki ratusan simbol dan bentuk.

Saat ini, telah berkembang *Technology Sign Recognition* (TSR) yaitu suatu teknologi yang mengembangkan sistem identifikasi objek rambu lalu lintas secara otomatis yang dapat menjadi salah satu alternatif untuk meningkatkan keselamatan berkendara. Hasil implementasi dari teknologi TSR adalah *Advanced Driver Assistance System* (ADAS). Dalam sistem bantuan pengemudi tingkat lanjut (ADAS), deteksi rambu lalu lintas berperan penting untuk menerima informasi jalan dan batas kecepatan terlebih dahulu.

Deteksi rambu lalu lintas merupakan masalah yang menantang karena oklusi dengan pohon, tiang, dan kendaraan besar. Selain itu, terdapat variasi yang besar dalam bentuk, warna, dan ukuran rambu lalu lintas karena perubahan pemandangan dan kondisi penerangan jalan (Manocha et al., 2019).

Terdapat berbagai penelitian tentang rambu lalu lintas dengan penggunaan teknologi yang semakin berkembang pesat, salah satunya penggunaan *computer vision*. Pada tahun 2017, Kun Qiao dkk melakukan penelitian tentang optimasi berdasarkan *Faster R-CNN* yang dikombinasikan melakukan percobaan berulang kali pada jaringan ZF yang menghasilkan nilai mAP (*mean Average Precision*) dengan optimasi sekitar 58%, dan jaringan VGG dengan optimasi sekitar 72% (Qiao et al., 2017). Pada tahun berikutnya 2018, Promlainak dkk menggunakan *Histogram*

*Oriented Gradient* (HOG) dan *Viola-Jones* untuk klasifikasi gambar lalu lintas dari inputan gambar dan *Support Vector Machine* (SVM) yang menghasilkan model terlatih untuk pengenalan rambu lalu lintas dengan gambar *testing* sebanyak 238 sampel dan 9 jenis rambu peringatan (Promlainak et al., 2018). Manocha dkk melakukan penelitian deteksi rambu lalu lintas dengan 3 jenis kelas yaitu rambu larangan, petunjuk, dan peringatan menggunakan *Korean Traffic Sign Detection* (KTSD) dataset berisi 3300 gambar jalan raya di Korea. Setiap gambar berada pada pemandangan jalan yang berbeda dengan memasukkan rambu lalu lintas ukuran kecil, sedang, dan besar dan terdiri 498 gambar untuk testing menghasilkan nilai mAP sebesar 86.61% yang dimana hasilnya jauh lebih baik daripada algoritma YOLOv3 dan *D-Patch* asli dalam hal nilai mAP dan waktu CPU (Manocha et al., 2019).

Masih di tahun yang sama, Wang Canyon meneliti menggunakan *deep learning* dan menjadikan fokus terhadap rambu batas kecepatan yang paling umum dan larangan rambu kiri dan kanan pada pengenalan rambu lalu lintas karena menurutnya rambu tersebut sangat penting bagi pengemudi untuk aman dalam mengemudi dengan tingkat akurasi bisa mencapai 96% (C. Wang, 2018). Penelitian selanjutnya dari Tarequl Islam pada 2019 tahun lalu. Ada total 28 rambu dari kategori rambu lalu lintas yang berbeda seperti rambu peringatan, rambu penenang lalu lintas, rambu batas kecepatan. Gambar diambil dari *database* citra rambu lalu lintas Inggris Raya namun citra jalan yang digunakan adalah Ukraina dan

Bangladesh. Gambar jalan Ukraina diambil dari *online* sedangkan jalan di Bangladesh diambil dengan telepon genggam dengan total hasil akurasi 90% (Islam, 2019).

Pada penelitian ini menggunakan algoritma versi ke 4 dari YOLO yaitu YOLOv4 untuk sistem deteksi dan pengenalan pada rambu-rambu lalu lintas. YOLOv4 menawarkan detektor canggih yang lebih cepat dan lebih akurat dari semua detektor alternatif yang tersedia (Bochkovskiy et al., 2020). Makalah ini memberikan kontribusi yaitu dapat mendeteksi 35 jenis rambu yang berbeda dari 4 kategori rambu-rambu lalu lintas yaitu rambu peringatan, larangan, perintah, dan petunjuk. Dengan demikian penelitian ini mengambil judul **"Deteksi dan Pengenalan Rambu-Rambu Lalu Lintas Di Indonesia Menggunakan Algoritma *You Only Look Once Version 4* (YOLOv4)."**

## B. RUMUSAN MASALAH

Berdasarkan pada latar belakang di atas, maka rumusan masalah pada penelitian ini adalah :

1. Bagaimana implementasi algoritma *You Only Look Once version 4* (YOLOv4) untuk mendeteksi dan mengenali rambu lalu lintas di Indonesia secara *realtime*?
2. Bagaimana akurasi yang dihasilkan algoritma *You Only Look Once version 4* (YOLOv4) untuk mendeteksi dan mengenali rambu lalu lintas di Indonesia secara *realtime* ?

### C. TUJUAN PENELITIAN

Berdasarkan rumusan masalah yang telah diuraikan, tujuan penelitian yang diusulkan adalah :

1. Untuk membangun sistem rambu lalu lintas di Indonesia menggunakan algoritma *YOLOv4* secara *realtime*.
2. Untuk menghitung akurasi yang dihasilkan menggunakan algoritma *YOLOv4* untuk mendeteksi dan pengenalan rambu lalu lintas di Indonesia secara *realtime*.

### D. MANFAAT PENELITIAN

Adapun manfaat dari penelitian ini adalah diharapkan dapat memberikan pengetahuan tentang pengimplementasian algoritma *YOLOv4* untuk mendeteksi dan mengenali rambu lalu lintas di Indonesia. Dan dapat digunakan sebagai referensi untuk pengembangan *Traffic Sign Recognition* yang diterapkan untuk kemudi otomatis (*auto pilot*) sebagai *Drive Assistance System (DAS)* pada mobil.

### E. BATASAN MASALAH

Mengingat luasnya ruang lingkup penelitian ini, maka penelitian ini dibatasi pada :

1. Sistem yang dibuat untuk mendeteksi rambu-rambu lalu lintas.
2. Proses pengambilan video dilakukan pada kondisi cerah atau cukup pencahayaan.
3. Pemrosesan secara *realtime* di sistem ini menggunakan CPU (*Central Processing Unit*).

## **F. SISTEMATIKA PENULISAN**

Sistematika penulisan pada penelitian ini adalah :

### **Bab I Pendahuluan**

Bab I berisi penjelasan tentang latar belakang penelitian tentang deteksi dan rekognisi rambu lalu lintas di Indonesia dengan algoritma YOLOv4, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah serta sistematika penulisan.

### **Bab II Tinjauan Pustaka**

Bab II berisi penjelasan tentang tinjauan pustaka yang merupakan penjelasan tentang hasil-hasil penelitian lainnya yang berkaitan dengan penelitian yang dilakukan tentang sumber acuan terbaru dari pustaka primer seperti buku, artikel, jurnal, dan tulisan asli lainnya untuk mengetahui penelitian relevan dengan usulan penulis terkait deteksi dan pengenalan rambu lalu lintas dengan algoritma YOLOv4. Dalam bab ini juga diuraikan tentang kerangka pikir tentang masalah, metode, pengukuran dan hasil dari penggunaan metode yang sesuai dengan objek dalam penelitian yang diusulkan.

### **Bab III Metode Penelitian**

Bab III berisi penjelasan tentang metode penelitian, penentuan masalah, penentuan pendekatan komputasi, juga penjelasan bagaimana pengembangan dan penerapan sistem objek penelitian. Selain itu, bab III menjelaskan metode analisis data, metode pengukuran penelitian, penerapan pada masalah penelitian, serta analisis kerja sistem.

#### **BAB IV Hasil Dan Pembahasan**

Bab IV merupakan penjelasan mengenai hasil yang didapatkan pada saat proses pengujian dilakukan.

#### **BAB V Kesimpulan Dan Saran**

Bab V berisi kesimpulan terhadap hasil yang didapatkan dalam penelitian ini dan juga berisi saran-saran kepada pembaca atau peneliti untuk dapat menyempurnakan kekurangan-kekurangan atau capaian-capaian yang belum tercapai pada penelitian ini agar kedepannya bisa mendapatkan hasil yang jauh lebih baik.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **A. LANDASAN TEORI**

##### **1. Rambu Lalu Lintas**

Rambu adalah alat yang utama dalam mengatur, memberi peringatan dan mengarahkan lalu lintas.

Rambu yang efektif harus memenuhi hal-hal berikut:

- a. Menarik perhatian dan mendapat respek pengguna jalan.
- b. Memberikan pesan yang sederhana dan mudah dimengerti.
- c. Menyediakan waktu cukup kepada pengguna jalan dalam memberikan respon.

Untuk memenuhi kebutuhan tersebut, pertimbangan-pertimbangan yang harus diperhatikan dalam perencanaan dan pemasangan rambu adalah:

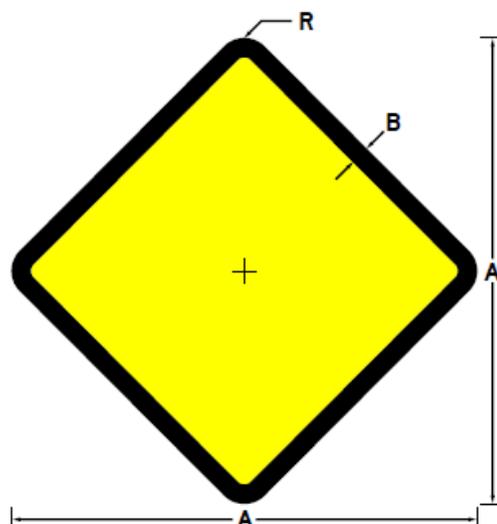
- a. Keseragaman bentuk dan ukuran rambu.
- b. Keseragaman dalam alat kontrol lalu lintas memudahkan tugas pengemudi untuk mengenal, memahami dan memberikan respon. Konsistensi dalam penerapan bentuk dan ukuran rambu akan menghasilkan konsistensi persepsi dan respon pengemudi.

Desain rambu yaitu warna, bentuk, ukuran, dan tingkat retrorefleksi yang memenuhi standar akan menarik perhatian pengguna jalan, mudah dipahami dan memberikan waktu yang cukup bagi pengemudi dalam memberikan respon. Selain itu, berdasarkan BAB II tentang Spesifikasi

Teknis Rambu Lalu Lintas Pasal 5 pada Peraturan Menteri Perhubungan Republik Indonesia Nomor 13 Tahun 2004 Ayat 3 “Setiap rambu wajib dipasang logo perhubungan berupa stiker dibagian depan sebelah bawah” dan Ayat 4 yang berbunyi “Stiker logo perhubungan sebagaimana dimaksud pada ayat (3) diterbitkan oleh Direktur Jenderal, gubernur, atau bupati/walikota sesuai dengan kewenangan” sehingga apabila rambu yang tidak memiliki stiker logo perhubungan, maka rambu tersebut palsu/tidak sesuai standar.

Lokasi rambu berhubungan dengan pengemudi sehingga pengemudi yang berjalan dengan kecepatan normal dapat memiliki waktu yang cukup dalam memberikan respon. Rambu yang benar pada lokasi yang tepat harus memenuhi kebutuhan lalu lintas dan diperlukan pelayanan yang konsisten dengan memasang rambu yang sesuai kebutuhan (Rahmad et al., 2018).

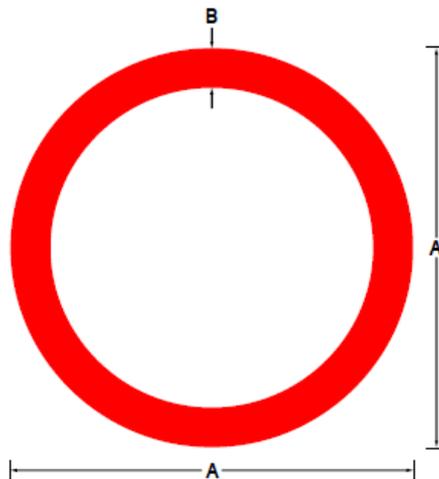
### 1.1 Ukuran Daun Rambu Peringatan Ukuran Standar



Gambar 1. Ukuran Standar Rambu Peringatan (Sumber: Minister of Transportation, 2014)

Jenis Ukuran	A	B	R
Kecil	450	25	37
Sedang	600	25	37
Besar	750	31	47
Sangat Besar	900	38	56

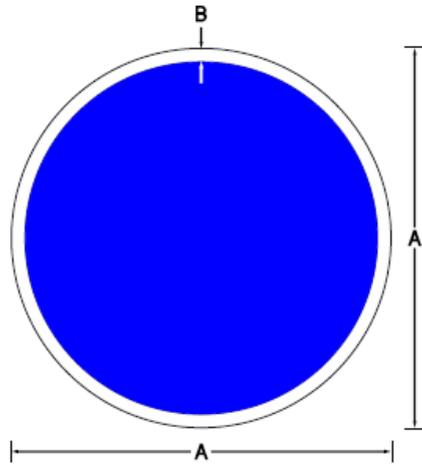
### 1.2 Ukuran Daun Rambu Larangan Ukuran Standar



Gambar 2. Ukuran Standar Rambu Larangan (Sumber:Minister of Transportation, 2014)

Jenis Ukuran	A	B
Kecil	450	45
Sedang	600	60
Besar	750	75
Sangat Besar	900	90

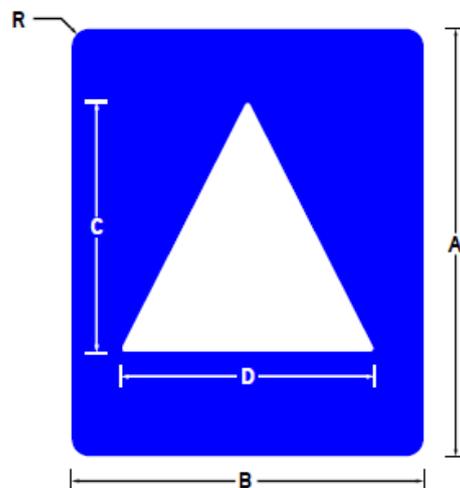
### 1.3 Ukuran Daun Rambu Perintah Ukuran Standar



Gambar 3. Ukuran Standar Rambu Perintah (Sumber:Minister of Transportation, 2014)

Jenis Ukuran	A	B
Kecil	450	20
Sedang	600	20
Besar	750	25
Sangat Besar	900	30

### 1.4 Ukuran Daun Rambu Petunjuk Ukuran Standar



Gambar 4. Ukuran Standar Rambu Petunjuk (Sumber: Minister of Transportation, 2014)

Jenis Ukuran	A	B	C	D	R
Kecil	500	400	260	230	37
Sedang	600	500	350	350	37
Besar	750	650	430	460	47
Sangat Besar	900	700	520	580	56

## 2. Visi Komputer

Visi komputer adalah bidang kecerdasan buatan yang melatih komputer menafsirkan dan memahami dunia visual. Visi komputer adalah cabang *Artificial Intelligent (AI)* yang menggunakan gambar digital dari kamera dan video serta model pembelajaran mendalam, mesin-mesin dapat dengan akurat mengidentifikasi dan mengklasifikasikan objek. Visi komputer mengimplementasikan beberapa kemampuan visual manusia yang diteruskan menuju otak seperti deteksi benda, pengenalan wajah dan mengenali bahaya.

Pada visi komputer, *deep Learning* sering digunakan untuk pengenalan dan deteksi objek. Proses *deep learning* pada visi komputer memanfaatkan piksel pada citra untuk ekstrasi pola atau atribut dari citra yang ingin dideteksi. Biasanya, untuk setiap aplikasi yang diberikan, keseluruhan tugas tidak dapat dilaksanakan pada sebuah tahapan tunggal. *Computer vision* terdiri dari tahapan-tahapan seperti perolehan citra, *preprocessing*, pengekstraksian fitur, penyimpanan objek secara asosiatif, pengaksesan suatu basis pengetahuan, dan pengenalan (B.Fisher et al., 2013).

### 3. Pengolahan Citra

Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka pengolahan citra tidak dapat dilepaskan dengan bidang visi computer. Kebanyakan teknik pemrosesan gambar dan visi komputer diimplementasikan dalam perangkat lunak komputer. Seringkali, hanya teknik paling sederhana yang bermigrasi ke perangkat keras. Melalui teknik pengkodean untuk memaksimalkan efisiensi dalam transmisi gambar merupakan kepentingan komersial yang cukup bahwa mereka telah menjamin pengembangan perangkat keras yang ekstensif dan sangat canggih. Sistem tersebut mencakup format pengkodean gambar *Joint Photographic Expert Group* (JPEG) dan *Moving Picture Expert Group* (MPEG). *C*, *C ++*, *Python*, dan *Java* sekarang adalah bahasa yang paling populer untuk implementasi sistem vision karena kekuatannya dalam mengintegrasikan fungsi tingkat tinggi dan rendah, dan ketersediaan kompiler yang baik. Ketika sistem menjadi lebih kompleks, *C ++*, *Python*, dan *Java* menjadi lebih menarik ketika enkapsulasi dan polimorfisme dapat dieksploitasi. Menurut (Nixon & Aguado, 2019), banyak orang menggunakan *Python* dan *Java* sebagai bahasa pengembangan sebagian karena kemandirian platform tetapi juga karena kemudahan dalam implementasi (meskipun beberapa mengklaim

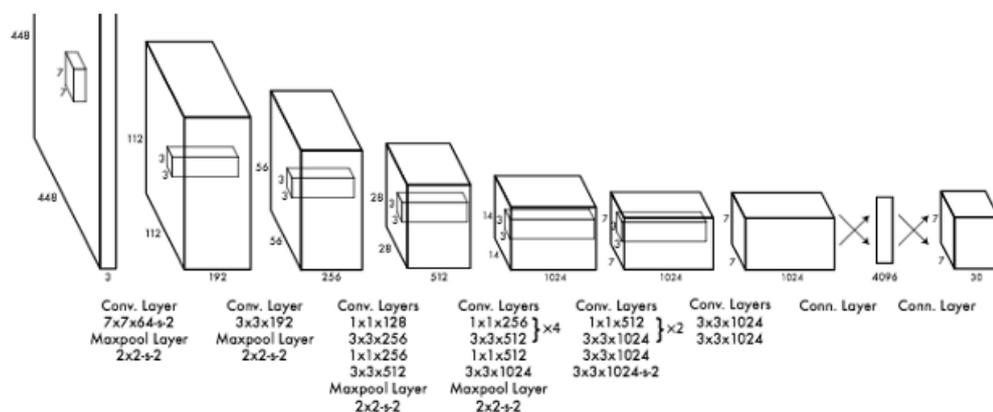
bahwa kecepatan / efisiensi lebih baik dalam C / C ++). *Python* saat ini merupakan bahasa pilihan, bukan karena alasan tertentu. Beberapa buku teks menawarkan sistem pemrosesan gambar yang diimplementasikan dalam bahasa-bahasa ini. Juga, ada banyak paket komersial yang tersedia, meskipun ini seringkali terbatas pada teknik dasar. Dalam hal paket perangkat lunak, yang paling populer adalah *OpenCV (Open Source Computer Vision)* yang filosofinya adalah untuk membantu penggunaan komersial visi komputer dalam antarmuka manusia-komputer, robotika, pemantauan, biometrik, dan keamanan dengan menyediakan infrastruktur gratis dan terbuka di mana upaya komunitas visi yang didistribusikan dapat dikonsolidasikan dan dioptimalkan kinerja.

#### **4. You Only Look Once (YOLO) version 1**

YOLO pertama kali diciptakan oleh Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. YOLO membagi gambar atau video yang diinput menjadi  $S \times S$  *grid*. Jika titik tengah koordinat pada GT (*Ground Truth*) suatu objek jatuh ke dalam grid, maka grid tersebut bertanggung jawab untuk mendeteksi suatu objek. Inovasi dari YOLO adalah mereformasi *framework* dari *region proposal detection*: seri R-CNN perlu menghasilkan *region proposal* dimana untuk melengkapi proses klasifikasi dan regresi tetapi ada *overlap* (tumpang tindih) dimana *region proposal* akan membawa banyak proses yang berulang. Namun, YOLO memprediksi *bounding box* dari objek

yang berada di semua *grid*, kemungkinan lokasi dan probabilitas dari semua *class* pada satu waktu (Redmon et al., 2016).

Jaringan YOLO menggunakan 1 x 1 *convolutional layer* (untuk integrasi antar *channel*) + 3 x 3 *convolutional layer* sebagai pengganti permulaan modul. Struktur jaringan pada YOLOv1 mengandung 24 *convolutional layers* dan 2 *full connection layers* seperti pada Gambar 5.



Gambar 5. Arsitektur YOLOv1 (Sumber:Redmon & Farhadi, 2017)

Algoritma ini tidak hanya mengklasifikasi sebuah *input* citra ke dalam suatu kategori, akan tetapi juga bisa mendeteksi berbagai objek yang berada didalam citra. Algoritma ini mengaplikasikan *single neural network* ke sebuah citra secara keseluruhan. Artinya, *network* tersebut membagi citra ke berbagai *regions* dan memprediksi *bounding boxes* dan probabilitas tiap *region*. Setiap *bounding box* yang ada dihitung berdasarkan probabilitas prediksinya.

## 5. You Only Look Once version 2 (YOLOv2)

YOLOv2 berfokus pada meningkatkan lokalisasi pada objek dan meningkatkan akurasi dari proses klasifikasi. Joseph Redmond dan Ali

Farmadi pada YOLOv2 ini, ingin lebih akurat dimana proses pendeteksian menjadi lebih cepat. Untuk mengatasi masalah kompleksitas dan akurasi penulis mengusulkan model klasifikasi yang disebut *Darknet-19* untuk digunakan sebagai *backbone* untuk YOLOv2. *Darknet-19* memiliki 19 lapisan *convolutional* dan 5 lapisan *maxpooling*. Ini mencapai akurasi 91,2% top-5 di *ImageNet* yang lebih baik daripada jaringan VGG (90%) dan YOLO (88%). Bentuk keluaran YOLOv2 adalah  $13 \times 13 \times (k \times (1+4+20))$  di mana  $k$  adalah jumlah *anchor box*, 20 adalah jumlah kelas. Jadi bentuk outputnya adalah  $13 \times 13 \times 125$ .

Pelatihan untuk klasifikasi menggunakan kumpulan data klasifikasi kelas *ImageNet* 1000 untuk 160 epoch menggunakan penurunan gradien stokastik dengan tingkat *learning rate* 0,1, peluruhan tingkat polinomial dengan kekuatan 4, peluruhan bobot 0,0005 dan momentum 0,9 menggunakan kerangka jaringan saraf *Darknet*. Selama pelatihan, menggunakan trik augmentasi data standar termasuk tanaman acak, rotasi, dan rona, saturasi, dan pergeseran eksposur. YOLOv2 awalnya melatih model pada gambar pada  $224 \times 224$ , kemudian mereka melakukan *fine tuning* pada jaringan klasifikasi pada resolusi  $448 \times 448$  penuh selama 10 epoch di *Image Net* sebelum pelatihan untuk deteksi. Jadi jaringan punya waktu untuk menyesuaikan filternya agar bekerja lebih baik pada input resolusi lebih tinggi. Ini meningkatkan mAP sebesar 4%. Pelatihan untuk deteksi, memodifikasi jaringan untuk deteksi dengan menghapus lapisan konvolusi terakhir dan sebagai gantinya menambahkan tiga lapisan

konvolusi  $3 \times 3$  dengan 1024 filter masing-masing diikuti oleh lapisan konvolusi  $1 \times 1$  akhir dengan jumlah *output* yang dibutuhkan untuk deteksi (Redmon & Farhadi, 2017).

Untuk memprediksi *k-bounding box* YOLOv2 menggunakan ide *anchor box*. YOLO memprediksi koordinat *bounding box* secara langsung menggunakan *fully connected layers* diatas ekstraktor fitur konvolusi. Di YOLOv2 *fully connected layers* dihapus dan menggunakan *anchor box* untuk memprediksi kotak pembatas.

## **6. You Only Look Once version 3 (YOLOv3)**

YOLOv3 merupakan salah satu *object detector* yang memiliki tingkat akurasi yang relatif baik dan mampu mendeteksi objek secara *real-time*. YOLOv3 menggunakan metode *anchor-based detector* dimana metode ini menggunakan kotak jangkar (*anchor box*) pada data gambar yang masuk ke dalam *Convolutional Neural Network* atau jaringan syaraf buatan. Jika menggunakan metode *anchor-based detector*, maka proses komputasi akan sangat signifikan menurun karena *anchor box* akan di-generate dalam sekali proses untuk setiap *frame*. *Anchor-based detector* pada YOLOv3 memilih *anchor box* yang tepat sebagai area objek yang dideteksi berdasarkan nilai *confidence score* dan IoU (*Intersection over Union*) yang memenuhi nilai *threshold* yang sudah ditentukan. Dengan metode ini YOLOv3 memperoleh *Mean Average Precision* pada COCO dataset sebesar 55,3% dengan kecepatan deteksi sebesar 35 *Frame per Second* menggunakan GPU Pascal Titan X.

Di YOLOv3 digunakan jaringan yang jauh lebih dalam yaitu Darknet-53 dengan 53 lapisan konvolusi. Darknet-53 lebih baik daripada ResNet-101 dan 1,5x lebih cepat. Darknet-53 memiliki kinerja yang mirip dengan ResNet-152 dan 2x lebih cepat.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 6. Darknet-53 (Sumber: Redmon & Farhadi, 2018)

Seperti yang dilihat pada Gambar 6, seluruh jaringan adalah rantai beberapa blok dengan beberapa langkah 2 lapisan konvolusional diantaranya untuk mengurangi dimensi. Di dalam blok, hanya ada struktur *bottleneck* (1x1 diikuti oleh 3x3) ditambah koneksi lewati. Jika tujuannya adalah untuk melakukan klasifikasi multi-kelas seperti yang dilakukan

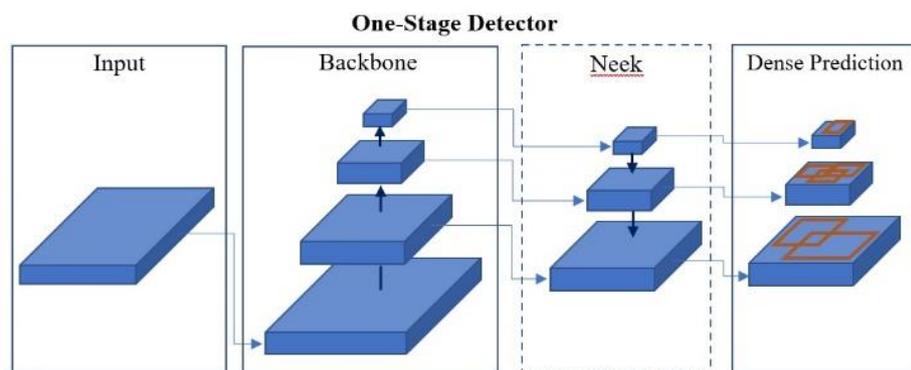
ImageNet, penyatuan rata-rata dan 1000 cara lapisan yang terhubung penuh ditambah aktivasi softmax akan ditambahkan.

YOLOv3 dirancang untuk menjadi detektor multi-skala, juga membutuhkan fitur dari beberapa skala. Oleh karena itu, fitur dari tiga blok sisa terakhir semuanya digunakan dalam deteksi selanjutnya. Setiap jaringan dilatih dengan pengaturan yang identik dan diuji pada 256x256, akurasi pemotongan tunggal. Waktu berjalan diukur pada Titan X pada 256x256. Dengan demikian, Darknet-53 bekerja setara dengan pengklasifikasi canggih tetapi dengan operasi *floating point* yang lebih sedikit dan kecepatan yang lebih tinggi. Darknet-53 juga mencapai operasi *floating point* terukur tertinggi per detik. Ini berarti struktur jaringan memanfaatkan GPU dengan lebih baik, membuatnya lebih efisien untuk dievaluasi dan dengan demikian lebih cepat. Itu sebagian besar karena ResNets memiliki terlalu banyak lapisan dan tidak terlalu efisien (Redmon & Farhadi, 2018).

## **7. You Only Look Once version 4 (YOLOv4)**

Pada versi ke 4 dikembangkan oleh Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao yang dirilis pada 24 April 2020. Dalam (Chethan Kumar et al., 2020) dijelaskan bahwa *You Only Look Once version 4* (YOLOv4) berjalan 2 kali lebih cepat dibandingkan *EfficientDet* dengan performa yang sama dan meningkatkan AP dan FPS sebesar 10% dan 12% dibandingkan YOLOv3. Algoritma model YOLOv4 berhasil

mendeteksi beberapa kelas objek dalam sebuah video dengan mendeteksi setiap objek di setiap *frame* demi *frame* hingga seluruh *frame* video tertutup, kelas tersebut dilambangkan dalam *bounding box*. Arsitektur dari YOLOv4 ditunjukkan pada Gambar 7. Terlihat pada gambar, bahwa YOLO merupakan *one-stage detector* yaitu melihat dan memproses sebuah gambar hanya dengan sekali proses sesuai dengan namanya *You Only Look Once*. Arsitektur YOLOv4 menggunakan *backbone*: CSPDarknet53 , *neck*: SPP dan PaNet, dan *head*: YOLOv3 yang juga merupakan dari *Dense Prediction*.

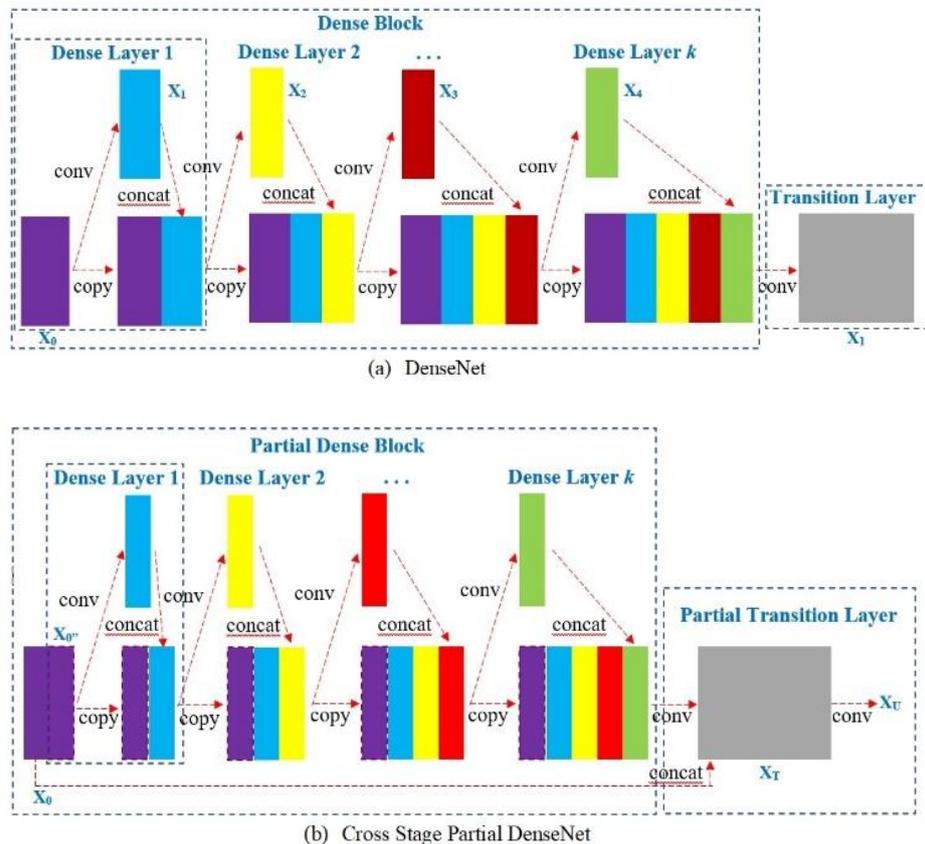


Gambar 7. Arsitektur YOLOv4 (Sumber: Bochkovskiy et al., 2020)

### 7.1 Cross Stage Partial Network (CSP) Darknet53

YOLOv4 menggunakan koneksi CSP di atas dengan Darknet-53 di bawah sebagai backbone dalam ekstraksi fitur. Model CSPDarknet53 memiliki akurasi yang lebih tinggi dalam deteksi objek dibandingkan dengan desain berbasis *ResNet* meskipun mereka memiliki kinerja klasifikasi yang lebih baik (C. Y. Wang et al., 2020). CSPDarknet53 didasarkan pada *DenseNet*. *DenseNet* dirancang untuk menghubungkan lapisan dalam jaringan neural konvolusional sebagai berikut: untuk

mengatasi masalah gradien yang menghilang, untuk mendukung penyebaran fitur, mendorong jaringan untuk menggunakan kembali fitur, dan mengurangi jumlah parameter jaringan (Huang et al., 2017).



Gambar 8. Ilustrasi DenseNet (a) dan CSP DenseNet (b) (Sumber: C. Y. Wang et al., 2020)

Pada Gambar 8 diatas merupakan ilustrasi *DenseNet* dan *Cross-Stage-Partial DenseNet (CSPDenseNet)*. Lapisan dasar yang diubah menjadi 2 bagian oleh *CSPNet* yaitu bagian pertama yang dilalui oleh blok padat dan lapisan transisi, sedangkan bagian yang satunya digabungkan dan dikirimkan oleh peta fitur ke tahap berikutnya (C. Y. Wang et al., 2020).

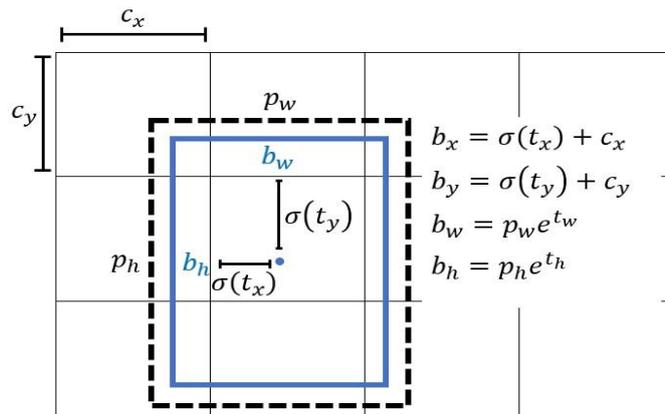
## 7.2 *Spatial Pyramid Pooling (SPP)* dan *Path Aggregation Network (PaNet)*

SPP dan PaNet merupakan *neck* yang terhubung ke *head*. Blok *neck* memiliki tujuan tertentu yaitu memberi lapisan ekstra diantara *head* dan *backbone* pada blok *dense prediction*. Fungsi *neck* ialah meningkatkan ketahanan jaringan dan meningkatkan akurasi dan kecepatan deteksi (Nixon & Aguado, 2019).

Blok SPP ditambahkan melalui CSPDarknet53, karena secara signifikan meningkatkan bidang reseptif, memisahkan fitur konteks yang paling signifikan dan menyebabkan hampir tidak ada pengurangan kecepatan operasi jaringan. PaNet sebagai metode agregasi parameter dari *backbone* yang berbeda untuk tingkat detektor yang berbeda, sedangkan di YOLOv3 menggunakan FPN (C. Y. Wang et al., 2020).

## 7.3 Prediksi Kotak Pembatas

YOLOv3 (*anchor based*) *head* sebagai arsitektur YOLOv4 yang memiliki proses yang sama. Peran *head* dalam kasus *one-stage detector* adalah melakukan *dense prediction*. *Dense prediction* adalah prediksi akhir yang terdiri dari vektor yang berisi koordinat *bounding box* yang diprediksi ialah (pusat, tinggi, lebar), prediksi *confidence score* dan label.



Gambar 9. Bounding Box (Sumber: Redmon & Farhadi, 2018)

*Anchor* adalah semacam *bounding box* prioritas yang memprediksi lebar dan tinggi kotak sebagai *offset* menggunakan sentroid *cluster*. Rumus di Gambar 9 menguraikan bagaimana mengubah jaringan keluaran untuk mendapatkan prediksi *bounding box*.  $t_x$ ,  $t_y$ ,  $t_w$ ,  $t_h$  adalah *output* jaringan.  $b_x$ ,  $b_y$ ,  $b_w$ ,  $b_h$  adalah titik pusat.  $c_x$  dan  $c_y$  adalah koordinat kiri atas *grid*.  $p_w$  dan  $p_h$  adalah dimensi *anchor* untuk kotak pembatas.  $x$ ,  $y$ , tinggi dan lebar merupakan prediksi yang dimasukkan (Redmon & Farhadi, 2018).

Selain menggunakan arsitektur diatas, YOLOv4 juga menggunakan fungsi aktivasi yang bernama *Mish*. Fungsi aktivasi memainkan peran penting dalam kinerja dan dinamika pelatihan di jaringan saraf. Fungsi aktivasi adalah fungsi titik-bijaksana non-linier yang bertanggung jawab untuk memperkenalkan nonlinier ke *input* transformasi linier dalam lapisan jaringan saraf. *Mish* adalah fungsi aktivasi non-monotonik yang halus, kontinu, teratur sendiri, yang secara matematis didefinisikan sebagai:

$$f(x) = x \tanh(\text{softplus}(x)) = x \tanh(\ln(1 + e^x)) \quad (1)$$

Mirip dengan *Swish*, *Mish* terbatas di bawah dan tidak terbatas di atas dengan kisaran  $[\approx -0,31, \infty)$ . Turunan pertama dari *Mish* dapat didefinisikan sebagai :

$$f'(x) = \frac{e^x \omega}{\delta^2} \quad (2)$$

Dimana,  $\omega = 4(x+1) + 4e^{2x} + e^{3x} + e^x(4x + 6)$  dan  $\delta = 2e^x + e^{2x} + 2$ . Terinspirasi oleh *Swish*, *Mish* menggunakan properti *Self-Gating* dimana input yang tidak termodulasi dikalikan dengan output dari fungsi input non-linear. Karena pelestarian sejumlah kecil informasi negatif, *Mish* menghilangkan dengan merancang prasyarat yang diperlukan untuk fenomena *Dying ReLU*. Properti ini membantu dalam ekspresivitas dan aliran informasi yang lebih baik. Menjadi tidak terbatas di atas, *Mish* menghindari saturasi, yang umumnya menyebabkan pelatihan melambat karena gradien mendekati nol secara drastis (Misra, 2020). Dibatasi di bawah juga menguntungkan karena menghasilkan efek regularisasi yang kuat.

## **8. Image Annotation**

Performa kecerdasan buatan sangat bergantung pada keakuratan data pelatihannya. Anotasi gambar adalah teknik utama yang digunakan untuk membuat data pelatihan untuk visi komputer. Agar mesin dapat melihat objek di sekitarnya, gambar beranotasi diperlukan untuk melatih algoritma pembelajaran mesin untuk belajar melihat dunia seperti yang kita

lakukan (Low, 2020). Anotasi gambar merupakan proses menentukan wilayah secara manual dalam gambar dan membuat deskripsi berbasis teks dari wilayah tersebut. Ini adalah langkah pertama yang penting dalam membangun kebenaran dasar untuk melatih model visi komputer. Ada berbagai kasus penggunaan untuk anotasi gambar, seperti *computer vision* pada penelitian ini yaitu deteksi dan mengenali rambu lalu lintas otomatis (Morikawa, 2019).

Anotasi dalam *machine learning* pada dasarnya adalah proses pelabelan data dalam berbagai media gambar, teks, atau video. Label biasanya ditentukan sebelumnya dan dipilih untuk memberikan informasi model visi komputer pada objek yang digambarkan dalam gambar. Algoritma kemudian akan menggunakan data beranotasi untuk mempelajari dan mengenali pola serupa saat disajikan dengan data baru. Bergantung pada sifat proyek, industri yang berbeda memerlukan bentuk anotasi yang berbeda.

Ada beberapa jenis *image annotation* yang digunakan pada *machine learning* antara lain :

a. *Bounding Box*

Jenis anotasi gambar yang paling umum digunakan dan paling sederhana adalah kotak pembatas. Bentuk anotasi ini mengharuskan labeller menggambar kotak sedekat mungkin dengan tepi objek utama dalam gambar. Penggunaan kotak

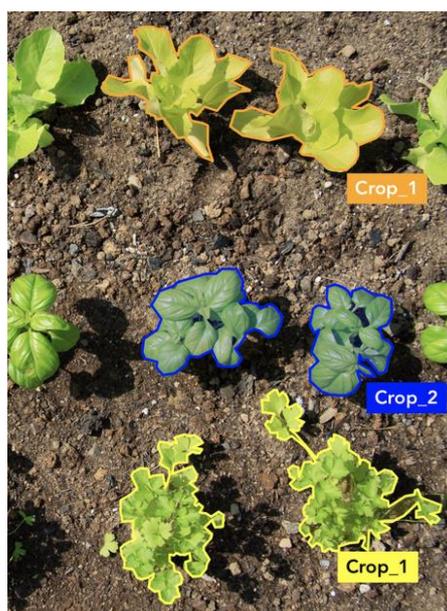
pembatas 2D sering ditemukan dalam klasifikasi objek, pelokalan, dan deteksi untuk berbagai industri.



Gambar 10. Contoh Penggunaan Bounding Box (Sumber: Low, 2020)

b. *Polygon Annotation*

Anotasi poligon penting karena tidak setiap objek dapat masuk secara tepat dalam kotak pembatas. Biasanya digunakan untuk anotasi yang lebih tepat untuk item yang bentuknya tidak beraturan, misalnya objek non-simetris pada gambar udara seperti buah, pohon, bangunan terkenal, atau rumah. Anotasi poligon biasanya membutuhkan tingkat presisi yang tinggi.



Gambar 11. Contoh Penggunaan Polygon Annotation (Sumber: Low, 2020)

c. *Poin Annotation*

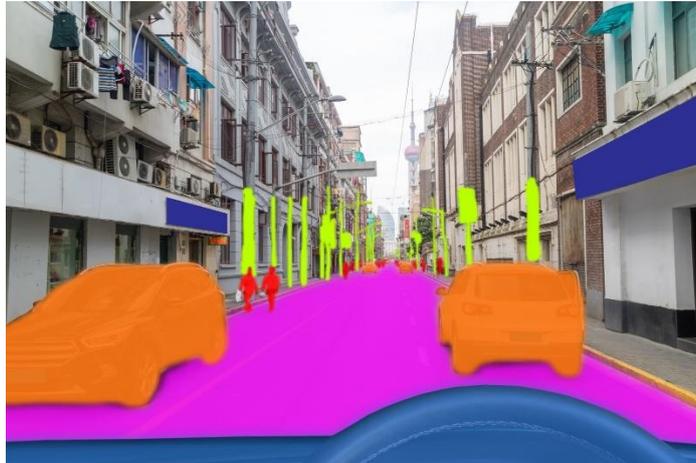
Anotasi titik melibatkan plotting titik-titik utama yang akurat di lokasi tertentu pada gambar. Bentuk anotasi ini paling umum digunakan untuk pengenalan wajah dan analisis sentimen. Dengan mengidentifikasi dan mengikuti pergerakan titik tengah pada ekspresi wajah, algoritme pembelajaran mesin dapat mendeteksi emosi melalui pembacaan prediktif.



Gambar 12. Contoh Penggunaan Poin Annotation (Sumber: Low, 2020)

d. *Semantic Annotation*

Segmentasi semantik biasanya digunakan untuk deteksi dan lokalisasi objek tertentu. Penerapan pemahaman gambar yang terperinci seperti itu biasanya dapat ditemukan di berbagai industri, dan ini sangat populer di industri kendaraan otonom, karena mobil yang dapat mengemudi sendiri memerlukan pemahaman yang mendalam tentang lingkungannya. Sedangkan di Agritech digunakan untuk analisis ladang tanaman untuk mendeteksi penyakit dan pertumbuhan yang tidak normal.

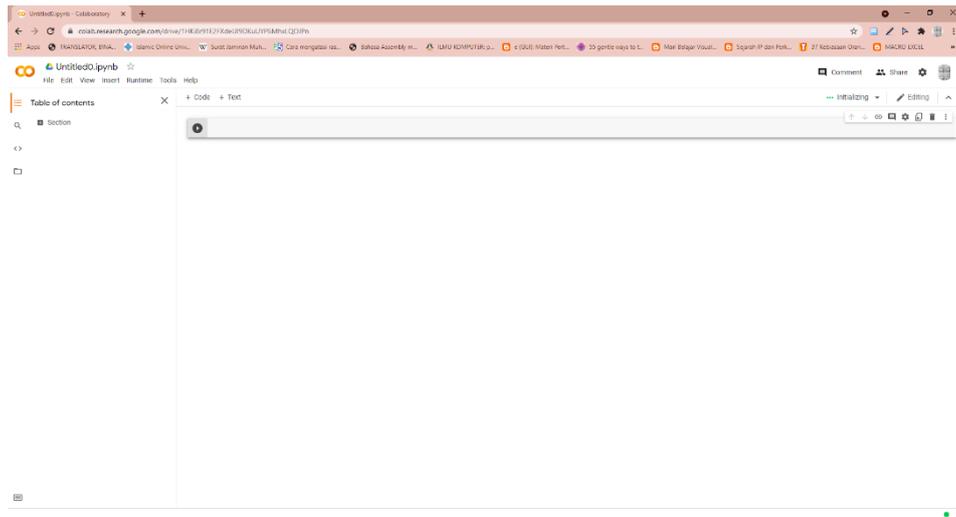


Gambar 13. Contoh Penggunaan Semantic Annotation (Sumber: Low, 2020)

## 9. Google Colaboratory

*Colaboratory*, atau “Colab” untuk lebih singkatnya merupakan produk dari *Google Research*. Colab memungkinkan siapa saja menulis dan mengeksekusi kode *python* melalui *browser* serta sangat cocok untuk *machine learning*, analisis data, serta pendidikan (Google, 2020). Secara lebih teknis, Colab merupakan layanan *notebook Jupyter* yang dihosting dan dapat digunakan tanpa penyiapan, serta menyediakan akses gratis ke *resource* komputasi termasuk GPU. Colab dapat digunakan secara gratis. Colab memungkinkan untuk digunakan dan dibagikan ke *notebook Jupyter* dengan pengguna lain tanpa perlu mendownload, menginstal, atau menjalankan apapun. *Notebook Colab* disimpan di *Google Drive*, atau dapat dimuat dari *GitHub*. *Notebook Colab* dapat dibagikan dengan cara seperti yang biasa dilakukan dengan Google Dokumen atau *Spreadsheet*. Dapat menelusuri *notebook Colab* menggunakan *Google Drive*. *Notebook Colab* dijalankan dengan menghubungkan ke mesin virtual yang memiliki masa penggunaan maksimum hingga 12 jam. *Notebook Colab* juga akan

terputus dari *Virtual Machine* (VM) jika terlalu lama tidak ada aktivitas. Masa penggunaan VM maksimum dan waktu tunggu nonaktif dapat berubah dari waktu ke waktu, atau berdasarkan penggunaan. Tampilan awal *notebook colab* dapat dilihat pada Gambar 14 berikut.



Gambar 14. Tampilan *Notebook Google Colaboratory*

## B. PENELITIAN TERKAIT

Beberapa penelitian terkait sistem deteksi dan pengenalan rambu lalu lintas menggunakan YOLOv4 diantaranya sebagai berikut:

1. Kun Qiao dkk, pada tahun 2017 melakukan penelitian tentang optimasi berdasarkan Faster R-CNN yang dikombinasikan melakukan percobaan berulang kali pada jaringan ZF yang menghasilkan nilai mAP (*mean Average Precision*) dengan optimasi sekitar 58%, dan jaringan VGG dengan optimasi sekitar 72%.
2. Promlainak dkk, pada tahun 2018 menggunakan *Histogram Oriented Gradient* (HOG) dan *Viola-Jones* untuk klasifikasi gambar lalu lintas

dari inputan gambar dan *Support Vector Machine* (SVM) yang menghasilkan model terlatih untuk pengenalan rambu lalu lintas dengan gambar testing sebanyak 238 sampel dan 9 jenis rambu peringatan.

3. Manocha dkk, melakukan penelitian deteksi rambu lalu lintas dengan 3 jenis kelas yaitu rambu larangan, petunjuk, dan peringatan menggunakan *Korean Traffic Sign Detection* (KTSD) dataset berisi 3300 gambar jalan raya di Korea pada tahun 2018. Setiap gambar berada pada pemandangan jalan yang berbeda dengan memasukkan rambu lalu lintas ukuran kecil, sedang, dan besar dan terdiri dari 498 gambar untuk testing menghasilkan nilai mAP sebesar 86.61% yang dimana hasilnya jauh lebih baik daripada algoritma YOLOv3 dan D-Patch asli dalam hal nilai mAP dan waktu CPU.
4. Wang Canyon juga melakukan penelitian pada tahun 2018 menggunakan deep learning dengan menjadikan fokus terhadap rambu batas kecepatan yang paling umum dan larangan rambu kiri dan kanan pada pengenalan rambu lalu lintas karena menurutnya rambu tersebut sangat penting bagi pengemudi untuk aman dalam mengemudi dengan tingkat akurasi bisa mencapai 96%.
5. Tarequl Islam pada 2019. Ada total 28 rambu dari kategori rambu lalu lintas yang berbeda seperti rambu peringatan, rambu penenang lalu lintas, rambu batas kecepatan, dll. Gambar diambil dari *database* citra rambu lalu lintas Inggris Raya namun citra jalan yang digunakan

adalah Ukraina dan Bangladesh. Gambar jalan Ukraina diambil dari online sedangkan jalan di Bangladesh diambil dengan telepon genggam dengan total hasil akurasi 90%.

6. Agus Mulyanto dkk, di tahun 2020 melakukan penelitian untuk mengenali pola rambu lalu lintas menggunakan YOLOv4 menggunakan dataset *Indonesia Traffic Signs* (ITS). Dataset ITS terdiri dari empat kategori (peringatan, larangan, wajib dan arahan) dengan dua puluh enam tanda. Model *deep learning* YOLOv4 berbasis *backbone CSP-DarkNet53* telah menunjukkan kinerja yang baik dengan Average Precision (mAP@0.5) sebesar 74.91% untuk 26 rambu rambu lalu lintas Indonesia (Mulyanto et al., 2020).

### C. STATE OF THE ART

Berikut tabel *State of The Art* dari penelitian yang berhubungan dengan sistem deteksi dan pengenalan rambu lalu lintas menggunakan YOLOv4:

Tabel 1 State of The Art

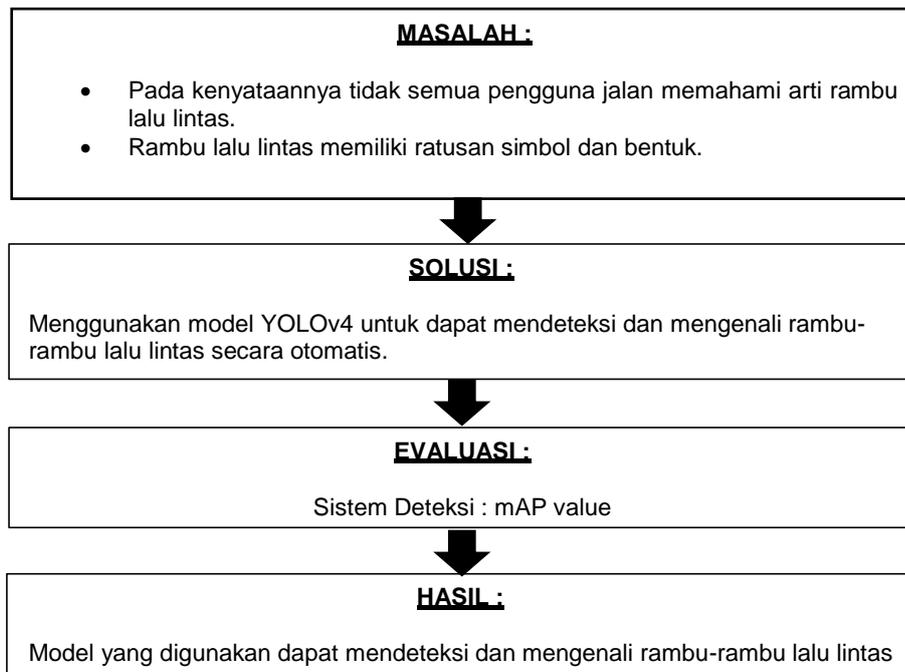
No	Penulis / Tahun	Judul	Metode/deskripsi	Hasil
1	Kun Qiao et.al/ 2017	Optimization of Traffic Sign Detection and Classification Based on Faster R-CNN	Faster-RCNN/ Dalam tulisan ini, peneliti mengusulkan mengoptimasi deteksi dan klasifikasi rambu-rambu lalu lintas berdasarkan Faster R-CNN yang dikombinasikan melakukan percobaan berulang kali pada jaringan ZF	Hasil eksperimen menunjukkan bahwa metode yang diusulkan dapat menghasilkan nilai mAP ( <i>mean Average Precision</i> ) dengan optimasi sekitar 58%, dan jaringan VGG dengan optimasi sekitar 72%

No	Penulis / Tahun	Judul	Metode/deskripsi	Hasil
2	Promlainak et.al / 2018	Thai Traffic Sign Detection and Recognition for Driver Assistance	<i>Histogram Oriented Gradient (HOG)</i> dan <i>Viola-Jones</i> / penelitian tentang klasifikasi gambar lalu lintas dari inputan gambar dan <i>Support Vector Machine (SVM)</i>	Hasil eksperimen menunjukkan bahwa metode yang diusulkan dapat menghasilkan model terlatih untuk pengenalan rambu lalu lintas dengan gambar testing sebanyak 238 sampel dan 9 jenis rambu peringatan.
3	Manocha et.al / 2018	Korean Traffic Sign Detection Using Deep Learning	Optimasi YOLOv3/ melakukan penelitian deteksi rambu lalu lintas dengan 3 jenis kelas yaitu rambu larangan, petunjuk, dan peringatan menggunakan <i>Korean Traffic Sign Detection (KTSD)</i> dataset berisi 3300 gambar jalan raya di Korea. Setiap gambar berada pada pemandangan jalan yang berbeda dengan memasukkan rambu lalu lintas ukuran kecil, sedang, dan besar dan terdiri dari 498 gambar untuk testing	Algoritma yang dikembangkan menghasilkan nilai mAP sebesar 86.61% yang dimana hasilnya jauh lebih baik daripada algoritma YOLOv3 dan D-Patch asli dalam hal nilai mAP dan waktu CPU.
4	Wang Canyong/2018	Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning	VGG-16, Convolutional neural network(CNN) / melakukan penelitian menggunakan deep learning dengan menjadikan fokus terhadap rambu batas kecepatan yang paling umum dan larangan rambu kiri dan kanan pada pengenalan rambu lalu lintas karena menurutnya rambu tersebut sangat penting bagi pengemudi untuk aman dalam mengemudi.	Hasil penelitiannya yaitu setelah melakukan 20.000 iterasi, tingkat akurasi bisa mencapai 96%.
5	Tarequl Islam / 2019	Traffic sign detection and recognition based on convolutional neural networks	Convolutional neural network(CNN) / Peneliti menggunakan total 28 rambu dari kategori rambu lalu lintas yang berbeda seperti rambu peringatan, rambu penenang lalu lintas, rambu batas kecepatan, dll. Gambar diambil dari <i>database</i> citra rambu lalu lintas Inggris Raya namun citra jalan yang digunakan adalah Ukraina dan Bangladesh. Gambar jalan Ukraina diambil dari online sedangkan jalan di Bangladesh diambil dengan telepon genggam.	Hasil eksperimen menunjukkan bahwa tingkat akurasi metode ini dengan total hasil akurasi 90%.

No	Penulis / Tahun	Judul	Metode/deskripsi	Hasil
6	Agus Mulyanto et.al/ 2020	Indonesian Traffic Sign Recognition For Advanced Driver Assistent (ADAS) Using YOLOv4	You Only Look Once version 4 (YOLOv4) / melakukan penelitian untuk mengenali pola rambu lalu lintas menggunakan YOLOv4 menggunakan dataset <i>Indonesia Traffic Signs</i> (ITS). Dataset ITS terdiri dari empat kategori (peringatan, larangan, wajib dan petunjuk) dengan dua puluh enam tanda.	Hasil eksperimen menunjukkan bahwa Model <i>deep learning</i> YOLOv4 berbasis <i>backbone CSP-DarkNet53</i> telah menunjukkan kinerja yang baik dengan Average Precision (mAP@0.5) sebesar 74.91% untuk 26 rambu rambu lalu lintas Indonesia.

Perbedaan penelitian ini dengan penelitian sebelumnya adalah fokus pada jenis dan total rambu lalu lintas yang digunakan lebih banyak dan beragam untuk mengetahui kemampuan YOLOv4 dalam mendeteksi dan mengenali dengan skala banyak. Selain itu sistem pada penelitian sudah tidak lagi dalam bentuk gambar maupun video tapi sudah secara *real time*.

#### D. KERANGKA PIKIR



Gambar 15. Kerangka Pikir

Pada Gambar 15 menunjukkan kerangka pikir pada penelitian ini. Pada tahap pertama menjelaskan permasalahan banyak pengguna jalan tidak memahami arti rambu lalu lintas padahal seharusnya rambu-rambu lalu lintas memberikan informasi secara visual kepada para pengguna jalan agar berkendara secara aman selain itu rambu-rambu lalu lintas memiliki ratusan simbol dan bentuk yang total berjumlah 265 rambu yang terdiri 97 rambu peringatan, 58 rambu larangan, 28 rambu perintah dan 82 rambu petunjuk yang telah ditetapkan pada Peraturan Menteri Perhubungan No.13 Tahun 2014. Kemudian, solusi yang ditawarkan ialah dengan menggunakan sebuah sistem berbasis deteksi objek yang dapat mendeteksi dan mengenali rambu-rambu lalu lintas secara otomatis menggunakan metode *deep learning* yaitu YOLOv4. Evaluasi dilakukan dengan menghitung nilai persentase *mean Average Precision* (mAP), *precision*, *recall*, *f1-score* untuk mengetahui keakuratan sistem yang dibuat dalam penelitian ini. Adapun penjelasan cara menghitung untuk evaluasi sistem pada bab 3 bagian analisis kerja sistem.