

Pengembangan Model Pendeteksian Gambar Rambu Lalu Lintas dengan Metode Faster R-CNN dengan Library Tensorflow

Christian Sanjaya¹⁾, Henri Septanto²⁾

Informatika, Fakultas Industri Kreatif Institut Teknologi dan Bisnis Kalbis
Jalan Pulomas Selatan Kav. 22, Jakarta 13210

¹⁾ Email: 2016102489@student.kalbis.ac.id

²⁾ Email: henri.septanto@kalbis.ac.id

Abstract: This study aims to develop a machine learning model to detect blocked traffic signals using the faster R-CNN method. Traffic signal detection is the most important part of a traffic unit. To solve this problem, the detection method relies on deep learning. The method used in this study is an image processing technique in which there is a faster R-CNN using additional stages of the search process. This research resulted in the development of a traffic light image detection model using the faster R-CNN method with a tensorflow library of traffic sign detection.

Keywords: machine learning, vehicle detection, computer vision, faster R-CNN

Abstrak: Penelitian ini bertujuan untuk mengembangkan model pembelajaran mesin terhadap pendeteksian rambu lalu lintas larangan menggunakan metode faster R-CNN. Deteksi rambu lalu lintas adalah bagian terpenting dalam kesatuan lalu lintas. Untuk mengatasi masalah ini, metode deteksi didasarkan pada pembelajaran yang mendalam. Metode yang digunakan pada penelitian ini adalah teknik pengolahan citra yang didalamnya terdapat faster R-CNN dengan menggunakan tahapan proses penelitian inkremental. Penelitian ini menghasilkan Pengembangan model pendeteksian gambar rambu lalu lintas dengan metode faster R-CNN dengan library tensorflow untuk pendeteksian rambu lalu lintas.

Kata kunci: machine learning, deteksi kendaraan, computer vision, faster R-CNN

I. PENDAHULUAN

Perkembangan teknologi yang terjadi saat ini adalah perkembangan teknologi yang terus berkembang di masyarakat seiring dengan majunya ilmu pengetahuan dalam masyarakat itu sendiri. Dengan adanya perkembangan teknologi ini akan terus memaksa agar manusia harus lebih berkembang dalam ilmu pengetahuan serta teknologi. Teknologi yang terus berkembang sampai saat ini sangat membantu dalam setiap kegiatan manusia, banyak faktor yang menjadi acuan terhadao semakin majunya perkembangan teknologi. Beberapa contoh penggunaan teknologi yang dibutuhkan untuk membantu manusia adalah dalam bidang pendidikan serta dalam bidang pekerjaan.

Seiring perkembangan teknologi tersebut, manusia sebagai tokoh utama dalam memajukan teknologi yang berjalan. Dalam memajukan teknologi dapat dilakukan dengan salah satu caranya yaitu membuat mesin dapat bekerja layaknya yang dilakukan seperti manusia. Mesin dapat bekerja layaknya manusia dengan menggunakan metode pembelajaran kepada mesin atau sering disebut *Machine Learning*. Metode tersebut dibuat untuk memudahkan pekerjaan manusia yang dapat dikerjakan oleh komputer dengan memberinya kecerdasan buatan atau *Artificial Intelligence*. Di dalam kecerdasan buatan terdapat metode pendeteksian objek, pendeteksian objek ini memiliki beberapa metode seperti *You Only Look Once (YOLO)*, *Faster*

Region-based Convolutional Neural Network (Faster R-CNN), K-Nearest Neighbor (KNN). Dalam beberapa metode tersebut memiliki kelebihan dan kekurangannya sendiri.

Pada penelitian ini menggunakan pembelajaran mesin yang digunakan dengan tujuan untuk melakukan pendeteksian terhadap rambu lalu lintas. Pendeteksian rambu lalu lintas dilakukan untuk memudahkan pembelajaran terhadap mesin serta sebagai pembelajaran kepada masyarakat terhadap arti dari setiap rambu lalu lintas. Dalam pendeteksian ini, dilakukan menggunakan algoritma *Faster R-CNN* untuk pendeteksian terhadap rambu lalu lintas.

II. METODE PENELITIAN

A. Metode

Penelitian ini menggunakan beberapa teori yang berhubungan dengan keseluruhan proses yang berjalan selama penelitian dilakukan. Teori-teori ini digunakan sebagai pendukung setiap proses yang berhubungan dengan proses pendeteksian rambu lalu lintas.

1. Rambu Lalu Lintas

Rambu lalu lintas merupakan salah satu dari perlengkapan jalan yang berupa lambang, huruf, angka, kalimat atau perpaduan di antaranya yang berfungsi sebagai peringatan, larangan, perintah atau petunjuk bagi pengguna jalan. [1]

2. Computer Vision

Computer Vision merupakan ilmu yang mempelajari bagaimana komputer dapat mengenali objek yang akan diamati. Hal ini dilakukan untuk memvisualisasikan pengamatan dari manusia yang akan diaplikasikan

kedalam komputer. *Computer Vision* terdiri dari 2 bidang ilmu, yaitu:

1. Pengolahan Citra (*Image Processing*)

Pengolahan citra / *Image Processing* merupakan ilmu yang mempelajari tentang kualitas citra agar menghasilkan citra yang baik dan mudah dimengerti oleh manusia dan mesin.

2. Pengenalan Pola (*Pattern Recognition*)

Pengenalan pola/*Pattern Recognition* merupakan proses pengelompokan data numerik secara otomatis, yaitu bertujuan untuk dapat diidentifikasi sebuah objek pada suatu citra. [2]

3. *Faster R-CNN*

Faster R-CNN adalah modifikasi dari *fast R-CNN*. Perbedaan di antara keduanya adalah bahwa jaringan *fast R-CNN* menggunakan pencarian selektif untuk menghasilkan bidang yang diinginkan, sedangkan jaringan *faster R-CNN* menggunakan jaringan "*Region Proposal Network*" atau dikenal juga sebagai RPN. RPN mengambil peta fitur citra sebagai input dan menghasilkan seperangkat objek, masing-masing dengan nilai objek sebagai output. [3]

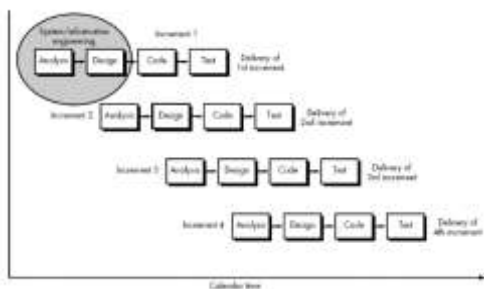
4. Inkremental

Model inkremental adalah metode pengembangan perangkat lunak yang terdiri dari beberapa tahapan perkembangan secara sekuensial linear. Proses pengembangan perangkat lunak dilakukan secara bertahap dan hasil dari setiap produk pada setiap tahapannya akan mengalami peningkatan.

Pada model inkremental, tahapan inkremental pertama adalah pembuatan inti dalam suatu program. Maka program yang dihasilkan pada tahapan inkremental pertama adalah program

yang memiliki fungsi utama dari perangkat lunak yang akan dibuat. Hasil yang terdapat pada inkremental pertama akan dievaluasi dan akan dilanjutkan untuk perencanaan tahapan inkremental berikutnya. Tahapan berikutnya tersebut yang disebut tahapan inkremental dua, dimana produk ini merupakan tahapan peningkatan dari inkremental pertama.

Tahapan pada inkremental akan terus berulang sampai menghasilkan program akhir yang lengkap sesuai dengan fungsi pada perangkat lunak yang ingin dibuat. Kelebihan dari inkremental adalah setiap tahapan yang terdapat didalam inkremental memiliki hasil akhir, walaupun hasil tersebut dapat dikembangkan kembali pada tahapan inkremental berikutnya. [4]



Gambar 1 Tahapan Inkremental

5. Tensorflow

Tensorflow adalah sebuah platform yang digunakan untuk machine learning yang bersifat end-to-end open source. Tensorflow mempunyai ekosistem tools, library, dan resources community yang konferensif dan fleksibel yang memungkinkan para penulis untuk meningkatkan teknologi terkini pada machine learning dan memudahkan para developers dalam menciptakan aplikasi berbasis machine learning. [5]

6. Epoch

Epoch adalah ketika seluruh dataset sudah melalui proses training pada Neural Network sampai dikembalikan ke awal untuk sekali putaran, karena

satu Epoch terlalu besar untuk dimasukkan (feeding) kedalam komputer maka dari itu kita perlu membaginya kedalam satuan kecil (batches). [6]

7. Residual Network (ResNet)

Resnet atau Residual Network merupakan jaringan residual yang memiliki jaringan yang dalam. Jaringan terdalam yang terdapat dalam resnet adalah berjumlah 152 lapisan. Jaringan ini memiliki 8 kali lebih dalam dari jaringan VGG namun kompleksitasnya masih lebih rendah dari jaringan VGG.

Peningkatan kedalaman jaringan pada deep convolutional neural network tidak selalu menghasilkan akurasi pelatihan yang lebih baik. Pada beberapa kasus terdapat penurunan akurasi pelatihan. Hal tersebut disebabkan oleh degradasi yang terjadi pada jaringan karena tidak semua jaringan mudah dioptimisasi. Solusi untuk mengurangi degradasi pada jaringan yang lebih dalam adalah dengan cara menambahkan pemetaan identitas (identity mapping) dan lapisan lainnya yang disalin dengan model yang lebih dangkal maka dibuatlah jaringan ResNet. [7]

layer name	output size	16-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112	7x7, 64, stride 2				
		3x3 max pool, stride 2				
conv2_x	56x56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28x28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14x14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7x7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1x1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	5.8×10^9	7.6×10^9	11.3×10^9

Gambar 2 Arsitektur dan jumlah lapisan pada ResNet

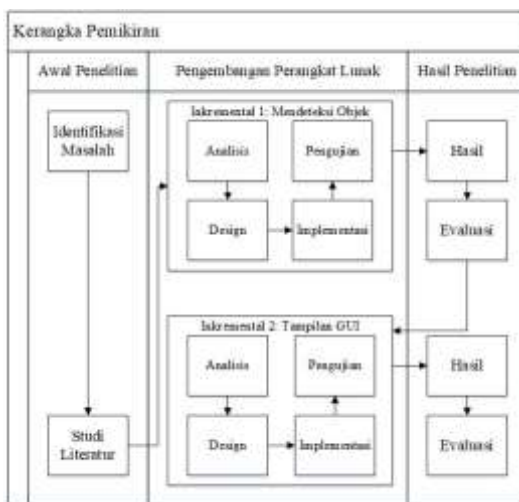
8. Metode Pengujian Perangkat Lunak

Penelitian ini dilakukan pengujian fungsionalitas yang terdiri dari pengujian white-box dan black-box. Pengujian white-box adalah pengujian yang didasarkan dalam kedetailan prosedur dan alur logika kode program.

Pengujian *black-box* adalah pengujian yang didasarkan pada detail yang terdapat di dalam aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan bisnis proses. [8]

B. Implementasi

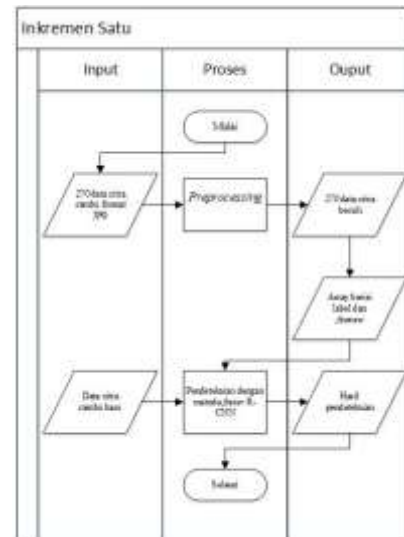
Pada tahap proses penelitian, peneliti membuat sebuah aplikasi menggunakan metode inkremental untuk diterapkan dalam pengembangan aplikasi. Model inkremental dipilih karena pengerjaan dapat dilakukan secara bertahap dan pada setiap tahapan dapat menghasilkan penambahan kualitas pada hasil aplikasi. Metode inkremental pada tahapan pertama merupakan tahapan inti yang sesuai dengan teori pengembangan perangkat lunak menggunakan metode inkremental. Tahap penelitian ini terdiri dari dua tahapan, yaitu tahapan inkremental satu dan tahapan inkremental dua.



Gambar 3 Proses Penelitian

1. Inkremen Satu

Tahap inkremen satu dilakukan untuk mendeteksi objek, melatih data agar sistem dapat mendeteksi objek yang akan dideteksi dan agar data tersebut dapat di klasifikasikan berdasarkan bentuk dan nama objek tersebut.



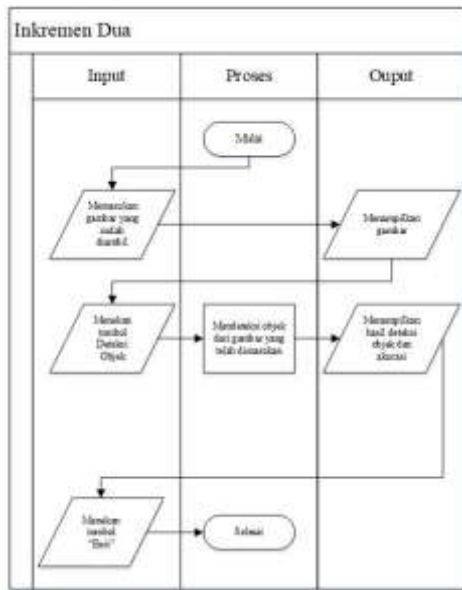
Gambar 4 Tahapan Inkremen Satu

Tabel 1 Skenario Pengujian Inkremen Satu

No	Tahap Uji	Input Data	Harapan
1	Ekstrasi citra rambu rambu lalu lintas	Folder training,	Menghasilkan file lable.txt
2	Training	File label.txt	Menghasilkan file train.py
3	Testing model	File gambar testing	Menghasilkan gambar yang sudah dideteksi

2. Inkremen Dua

Tahap inkremen dua dilakukan untuk pembuatan tampilan gui agar pengguna dapat melakukan pendeteksian objek rambu lalu lintas larangan dengan mudah.



Gambar 5 Tahapan Inkremen Dua

Tabel 2 Skenario Pengujian Inkremen Dua

No	Tahap Uji	Input data	Harapan
1	Mengambil gambar objek	Menekan tombol "Pilih Gambar"	Memproses data
2	Mendeteksi objek dari gambar yang sudah dipilih	Mengambil objek yang sudah diambil sebelumnya	Menampilkan hasil pendeteksian objek
3	Menghapus data gambar	Menekan tombol "Reset"	Menghapus data
4	Keluar	Menekan tombol "Keluar"	GUI keluar dengan menekan tombol keluar

III. HASIL DAN PEMBAHASAN

A. Inkremen Satu

Proses pengembangan algoritma terhadap pendeteksian rambu lalu lintas diawali dengan pemberian label kepada gambar rambu lalu lintas yang akan di training. Selain pemberian label, proses pertama juga diawali dengan menentukan lokasi *Bounding Box* yang digunakan untuk dilakukannya proses pendeteksian. Kemudian proses yang dilakukan setelah pemberian label dan adalah mengekstraksi file gambar

kedalam bentuk *file* (.CSV) dan kemudian dari *file* gambar yang telah berbentuk (.CSV) tersebut di *convert* kembali menjadi format (.TXT) dengan nama *Annotate.txt* yang akan digunakan untuk proses training yang dimasukan kedalam *code* training. Pada penelitian ini, proses training dilakukan dengan menggunakan *epoch* sebanyak 300 kali dengan *Batch Size* sebanyak 300 kali. Training tersebut akan menghasilkan dua buah *file* baru yaitu file model *frcnn.hdf5* dan *config.pickle*. Kedua *file* tersebut akan digunakan kembali saat melakukan proses tahapan *testing*. Dari hasil *testing* tersebut akan menghasilkan gambar baru yang sudah memiliki *Bounding Box* dan label sesuai dengan gambar yang sudah di training sebelumnya.

Gambar 6 Hasil objek deteksi



Pada pendeteksian ini juga menghasilkan nilai akurasi yang terdapat pada hasil deteksi dan dapat dilihat dalam *Command Prompt*. Pada halaman *Command Prompt* juga memiliki waktu yang dibutuhkan untuk sebuah model melakukan pemrosesan pendeteksian suatu objek yang akan dideteksi.

Tabel 3 Hasil pengujian inkremen satu

No	Nama Proses	Skenario	Proses
1	Input Dataset	Mengambil seluruh gambar dan dikumpulkan kedalam satu folder dan setiap gambar akan disimpan ke dalam satu variabel	Gambar masuk
2	Membuat Array	Membuat array untuk label dan dataset	Data gambar akan dikelompokkan kedalam label yang dibedakan berdasarkan jenis rambu
3	Model Faster R-CNN	Membuat model Faster R-CNN dengan epoch tertentu	Membuat model dengan cara melakukan pelatihan kepada data training gambar
4	Deteksi gambar	Mendeteksi beberapa jenis rambu lalu lintas larangan	Mendeteksi gambar dengan model yang sudah dibuat

B. Inkremen Dua

Pada tahapan inkremen dua penelitian difokuskan kepada tampilan terhadap GUI dengan tujuan agar pengguna dapat mudah dalam melakukan pendeteksian objek tanpa harus menjalankan pendeteksian di *Command Prompt*. Pada tahapan penelitian ini digunakan terhadap membuat tampilan serta fungsi-fungsi yang dapat dijalankan didalam tampilan tersebut. Pembuatan tampilan GUI dapat dibuat menggunakan bantuan *library* yang telah disediakan oleh Python yakni dengan menggunakan *PyQt5* serta *PyQt5 Tools* agar pembuatan tampilan GUI dapat dilakukan dengan mudah dan cepat.

Setelah membuat tampilan GUI, selanjutnya penelitian dilanjutkan dengan memberikan fungsi pada setiap tombol yang terdapat didalam tampilan GUI menggunakan *code* yang dapat di

tambahkan dengan pemrograman Python. Sebelum fungsi tombol di berikan, tampilan yang sudah dibuat harus di *convert* dari bentuk tampilan GUI menjadi bentuk *code* dengan cara menggunakan *Command Prompt* dengan *code pyuic5 -x program.ui -o program.py*. *Code* program.ui digunakan untuk memanggil program yang sudah dibuat sebelumnya dalam bentuk tampilan GUI dan di ubah kedalam bentuk *Code* program dengan nama program.py.

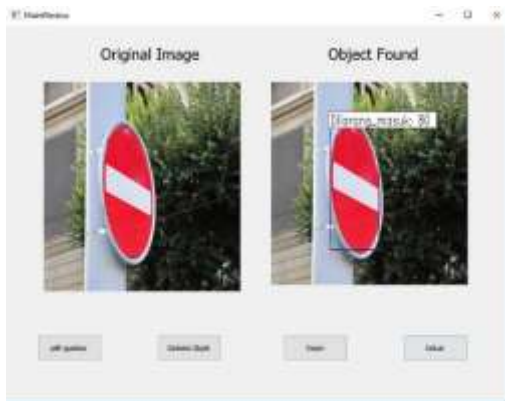
Membuka tampilan GUI pendeteksian rambu ini dapat dibuka menggunakan *Command Prompt* dengan *code* python program.py , dimana *code* tersebut dapat membuka tampilan dari *file* yang sudah dibuat sebelumnya didalam program.py.



Gambar 7 Tampilan GUI

Tampilan GUI yang digunakan pada penelitian ini memiliki 4 tombol yang digunakan untuk fungsi yang berbeda pada setiap tombolnya. Tombol pertama merupakan tombol “pilih gambar” yang digunakan untuk mengambil gambar yang terdapat didalam komputer. Tombol kedua merupakan tombol “Deteksi Objek” yang digunakan untuk mendeteksi gambar yang sudah dipilih sebelumnya dengan menjalankan pendeteksian secara otomatis di dalam *Command Prompt* dan akan keluar hasilnya di

dalam tampilan GUI setelah proses di *Command Prompt* telah selesai dilakukan. Tombol berikutnya adalah tombol “Reset” yang digunakan untuk menghapus gambar original dan gambar yang sudah di deteksi didalam tampilan GUI. Tombol yang terakhir adalah tombol “keluar”, tombol ini digunakan untuk keluar dari tampilan GUI.



Gambar 8 Hasil pendeteksian tampilan GUI

Tabel 4 Hasil pengujian inkremen dua

No	Nama Tombol	Fungsi	Hasil
1	Pilih gambar	Memasukkan gambar yang berada pada suatu folder untuk ditampilkan kedalam tampilan GUI.	VALID
2	Deteksi objek	Melakukan pendeteksian objek dengan memanggil file test_frcnn.py dan memakai model_frcnn.hdf5 dan menampilkan hasil objek kedalam tampilan GUI.	VALID
3	Reset	Menghapus gambar pada original image dan object found yang terdapat pada tampilan gui.	VALID
4	Keluar	Keluar dari tampilan GUI.	VALID

IV. SIMPULAN

Berdasarkan hasil penelitian dan uji coba yang dilakukan oleh peneliti, maka penelitian ini dapat diambil kesimpulan sebagai berikut:

1. Menghasilkan model yang di implementasikan kedalam tampilan GUI yang dapat mendeteksi dan melakukan perhitungan terhadap keluar atau masuknya manusia menggunakan metode YOLO.
2. Pengenalan rambu lalu lintas larangan akan lebih efektif setelah dilakukan tahapan *preprocessing* dimana gambar dipotong terlebih dahulu dan diberi label agar lebih mudah pada saat pendeteksian.
3. Saat melakukan training data, jumlah dataset dan nilai epoch sangat berpengaruh dalam tingkat keakuratan terhadap pendeteksian objek.

DAFTAR RUJUKAN

- [1] R. Cahya, I.F. Rahmah, R.A. Asmara, "Deteksi Dan Pengenalan Rambu Lalu Lintasa Di Indonesia Menggunakan RGBN Dan GABOR", vol.3, 2017.
- [2] R. Kusumanti, W.S. Pambudi and A. N. Tomponu, "Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang," *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*, 2012.
- [3] L. Zhang, L. Lin, X. Liang, dan K. He. "Is faster R-CNN doing well for pedestrian detection?" *ECCV*, 2016.
- [4] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, New York: MxGraw-Hill, 2010.
- [5] "Introduction to Tensorflow", Tensorflow[online]. Tersedia: <https://www.tensorflow.org/> [Diakses 20 Juni 2020]
- [6] "Memahami Epoch Batch Size Dan Iteration", imam.digmi.id, 2018. [Online]. Available: <https://imam.digmi.id/post/memahami-epoch-batch-size-dan-iteration/> [Accessed: 08-Ags-2020].
- [7] He, K. Zhang, X. Ren, S. & Sun, J. (2015) Deep Residual Learning for Image Recognition Kaiming. *IEEE*, 3 428-429. <https://doi.org/10.1246/cl/2003.428>.
- [8] R. AS and M. Shalahudin, *Rekayasa Perangkat Lunak : Terstruktur dan berorientasi objek*, Bandung: Informatika, 2014.