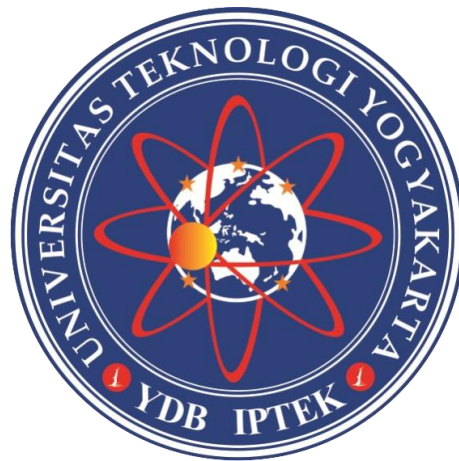


**RANCANG BANGUN SISTEM PENGENDALI LAMPU LALU
LINTAS BERDASARKAN PENGENALAN CITRA DIGITAL
KENDARAAN MENGGUNAKAN METODE FASTER R-CNN**

NASKAH PUBLIKASI TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Mencapai derajat Sarjana S-1 Program Studi Teknik Elektro



Disusun oleh
AHMAD MIQDAD MA'ALI
5150711015

**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

HALAMAN PENGESAHAN
NASKAH PUBLIKASI TUGAS AKHIR MAHASISWA

Judul Tugas Akhir

**RANCANG BANGUN SISTEM PENGENDALI LAMPU LALU LINTAS
BERDASARKAN PENGENALAN CITRA DIGITAL KENDARAAN
MENGUNAKAN METODE FASTER R-CNN**

Judul Naskah Publikasi

**RANCANG BANGUN SISTEM PENGENDALI LAMPU LALU LINTAS
BERDASARKAN PENGENALAN CITRA DIGITAL KENDARAAN
MENGUNAKAN METODE FASTER R-CNN**

Disusun oleh

AHMAD MIQDAD MA'ALI

5150711015

Mengetahui,

Nama	Jabatan	Tanda Tangan	Tanggal
M.S. Hendriyawan A, S.T., M.Eng	Pembimbing

Naskah Publikasi Tugas akhir ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana S-1 Program Studi Teknik Elektro.

Yogyakarta ,.....

Ketua Program Studi Teknik Elektro

Fakultas Teknologi Informasi dan Elektro, Universitas Teknologi Yogyakarta

M.S Hendriyawan Achmad,S.T.,M.Eng

NIK. 110810056

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini:

N a m a : Ahmad Miqdad Ma'ali

NIM : 5150711015

Program Studi : Teknik Elektro

**“Rancang Bangun Sistem Pengendali Lampu Lalu Lintas
Berdasarkan Pengenalan Citra Digital Kendaraan Menggunakan Metode
Faster R-Cnn”**

Menyatakan bahwa Naskah Publikasi ini hanya akan dipublikasikan di
JURNAL TeknoSAINS FTIE UTY, dan tidak dipublikasikan di jurnal yang lain.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta

Pada tanggal :

Yang menyatakan

Ahmad Miqdad Ma'ali

5150711015

Rancang Bangun Sistem Pengendali Lampu Lalu Lintas Berdasarkan Pengenalan Citra Digital Kendaraan Menggunakan Metode Faster R-Cnn

Ahmad Miqdad Ma'ali, M.S. Hendiyawan Achmad

Program Studi Teknik Elektro, Fakultas Teknologi Informasi dan Elektro

Universitas Teknologi Yogyakarta

Jl. Ringroad Utara Jombor Sleman Yogyakarta

E-mail : miqdadmq@gmail.com

ABSTRAK

Kemacetan merupakan hal yang umum terjadi saat ini, terutama di kota-kota besar. Salah satu penyebab kemacetan adalah meningkatnya jumlah kendaraan yang ada di jalan. Sehingga hal tersebut menyebabkan kemacetan, khususnya pada area perempatan jalan. Dimana kondisi kepadatan kendaraan pada tiap jalur yang tidak sama. Misalnya, pada salah satu jalur, padat oleh kendaraan sedangkan di jalur lain lengang, sehingga jalur yang padat akan semakin padat dan jalur yang lengang akan semakin lengang dikarenakan durasi lampu hijau yang sama, oleh karena itu untuk mengatur kelancaran lalu lintas adalah dengan lampu lalu lintas adaptif berdasarkan tingkat kepadatan kendaraan tiap jalur. Saat ini pengaturan lampu lalu lintas adaptif dilakukan berdasarkan waktu, hal tersebut masih kurang efektif dikarenakan waktu atau jam macet tidak selalu sama setiap hari. Solusi yang ditawarkan dalam penelitian ini untuk penerapan lampu lalu lintas adaptif adalah menggunakan computer vision dan pengolahan citra digital agar lampu lalu lintas disesuaikan secara real time melalui visual. Dengan menggunakan metode Faster R-CNN, maka sistem dapat mendeteksi objek kendaraan secara real-time di jalan raya. Sistem dapat mendeteksi melalui dataset kendaraan yang telah dilakukan pelatihan sebelumnya. Kemudian sistem akan menghitung jumlah kendaraan yang ada di jalan untuk menentukan durasi lampu hijau dan merah sesuai dengan tingkat kepadatannya. Hasil dari penelitian ini adalah mendeteksi dan menghitung jumlah kendaraan pada simulasi perempatan jalan dengan menggunakan dataset berupa mainan berbentuk kendaraan untuk menentukan durasi lampu lalu lintas. Hasil akurasi dari dataset yang telah dilakukan pelatihan adalah 97.027% serta persentase error perhitungan jumlah kendaraan adalah sebesar 2.188%.

Kata kunci : Computer Vision, Pengolahan Citra Digital, Dataset, Deep Learning, Faster R-CNN

1. PENDAHULUAN

Kemacetan merupakan hal yang umum terjadi di jalan seiring dengan meningkatnya tingkat penggunaan kendaraan pribadi seperti mobil. Oleh karena itu, untuk mengatur kelancaran lalu lintas adalah dengan lampu lalu lintas adaptif berdasarkan tingkat kepadatan kendaraan tiap jalur. Salah satu cara solusi dalam penerapan lampu lalu lintas adaptif adalah menggunakan waktu, dimana lampu lalu lintas diatur sesuai waktu umumnya terjadi kemacetan, solusi lainnya dalam mengatur lampu lalu lintas agar adaptif yang akan dibahas dalam penelitian ini adalah melalui tingkat kepadatan kendaraan dengan menggunakan *computer vision* dan *deep learning*. Kepadatan kendaraan dapat ditentukan dengan menggunakan deep learning dan pengolahan citra, yaitu deteksi objek dan klasifikasi objek. Dalam penelitian lain pernah dilakukan deteksi kendaraan berbasis pengolahan citra digital dan *deep learning*. Penelitian yang dilakukan oleh Lazaro, A. et al (2017) Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV. Metode yang diterapkan dalam penelitian tersebut adalah *Haar-like Features*, yaitu dengan membuat sebuah *classifier* yang di-*training* dengan sejumlah sampel

citra dari suatu objek dengan hasil akurasi rata-rata 77.8% untuk kondisi jalan sepi, 47.5% untuk kondisi jalan normal, dan 28.2% untuk kondisi jalan padat. Kemudian Plemakova, V (2018) dalam *Vehicle Detection Based on Convolutional Neural Network*, mendeteksi dan mengklasifikasi objek kendaraan menggunakan metode *Convolutional Neural Network*. Dalam penelitian ini ditawarkan metode *Faster R-CNN* untuk deteksi dan klasifikasi objek kendaraan yang kemudian akan dihitung jumlah kendaraan berdasarkan klasifikasinya. Kemudian setelah diketahui jumlah kendaraan berdasarkan klasifikasinya dapat ditentukan durasi lampu hijau dan lampu merah sesuai dengan tingkat kepadatannya. Apabila kendaraan semakin padat maka durasi lampu hijau pada sisi jalan tersebut akan semakin lama, namun pada sisi lain yang jumlah kendaraannya tidak terlalu padat durasi lampu hijau akan tetap sama. Metode *Faster R-CNN* dikembangkan oleh Ren, S. et al (2015) dimana dikenalkan *Region Proposal Network* (RPN) sebagai cara untuk membuat dataset agar dapat mendeteksi objek pada metode *Faster R-CNN*, dengan cara pengajuan atau pengecekan gambar ke sistem apakah gambar yang terekam berupa objek atau background. Hasil dari penelitian ini adalah dengan

menggunakan metode ini dapat dihasilkan tingkat akurasi sebesar 73.2 % dan 70.4 % berdasarkan 300 pengujian pada tiap gambar.

Maka untuk mewujudkan sistem tersebut diperlukan computer vision sebagai object recognition yang bertujuan untuk mendeteksi atau mengenali objek-objek tertentu yang kemudian dapat diamati *secara real time*, sehingga dapat membantu manusia dalam mengklasifikasi dan menghitung jumlah kendaraan. Metode yang digunakan dalam penelitian ini untuk mendeteksi objek kendaraan adalah metode *faster r-cnn*.

2. METODOLOGI PENELITIAN

2.1. Alat dan Bahan

a. Open CV

OpenCV (Open Source Computer Vision Library) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time, yang pemngembangannya diawali oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. OpenCV dirilis dibawah lisensi permisif BSD yang lebih bebas dari pada GPL, dan memberikan kebebasan sepenuhnya untuk dimanfaatkan secara komersil tanpa perlu mengungkapkan kode sumbernya. Ia juga memiliki antar muka yang mendukung bahasa pemrograman C++, C, Python dan Java, termasuk untuk sistem operasi Windows, Linux, Mac OS, iOS dan Android. OpenCV didisain untuk efisiensi dalam komputasi dan difokuskan pada aplikasi real-time.

b. Tensorflow

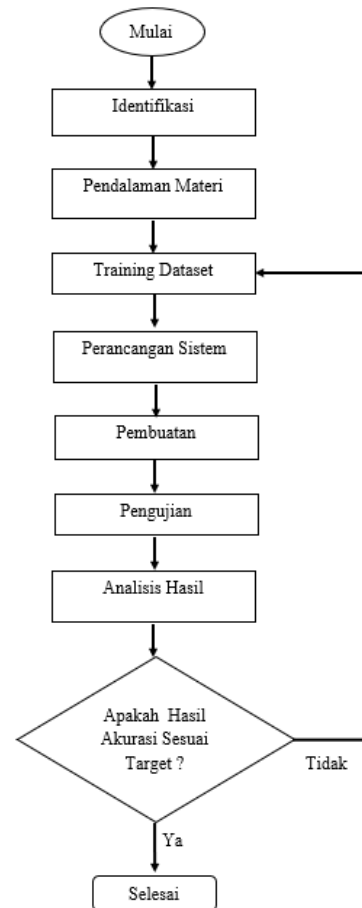
TensorFlow merupakan kerangka dasar yang digunakan dalam proses machine learning. TensorFlow merupakan library khusus untuk machine learning yang dikembangkan oleh google. TensorFlow telah dimanfaatkan terhadap berbagai hal, salah satunya adalah digunakan untuk mencegah kebutaan. Caranya, dengan membantu dokter melakukan filter terhadap retinopati diabetes. TensorFlow juga dimanfaatkan untuk membantu penyelamatan hutan, dengan memberi peringatan dini terhadap potensi aktivitas deforestasi ilegal.

c. Webcam

Webcam adalah sebutan bagi kamera waktu-nyata yang gambarnya bisa dilihat melalui www (World Wide Web), program pengolah pesan cepat, atau aplikasi pemanggilan video. Kamera web dapat diartikan juga sebagai sebuah kamera video digital kecil yang dihubungkan ke komputer melalui port USB, port COM atau dengan jaringan Ethernet atau Wi-Fi.

2.2. Langkah Penelitian

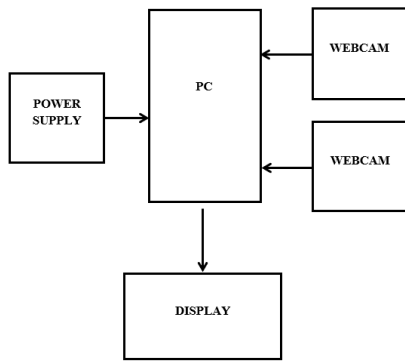
Proses penelitian ini melibatkan beberapa langkah yang harus dilakukan, seperti yang ditunjukkan pada gambar 3.1.



Gambar 1: Langkah penelitian

2.3. Deskripsi Sistem

Sistem terdiri dari PC atau laptop dan juga kamera webcam. Sistem akan mendeteksi jenis dan jumlah kendaraan melalui gambar yang terekam oleh kamera webcam. Kemudian sistem akan mengatur durasi lampu lalu lintas sesuai dengan jumlah kendaraan yang ada. pemasangan sistem dilakukan pada perempatan jalan, untuk mendeteksi jenis kendaraan dan menghitung jumlah kendaraan. Sistem bekerja sesuai dengan jumlah kendaraan yang ada pada suatu jalan, jika jumlah kendaraan terlalu padat, maka lampu hijau akan menyala lebih lama dari biasanya. Untuk mendeteksi jenis kendaraan digunakan dataset khusus yang terdiri dari mobil, motor, bus, truk, sepeda dan manusia. Gambar 2 berikut ini menunjukkan diagram blok sistem :



Gambar 2: Diagram blok penelitian

2.4. Manajemen Lampu Lalu Lintas

Menurut (Underwood) manajemen lalu lintas adalah pengorganisasian, perencanaan, pemberian arah, dan pemantauan keadaan pergerakan lalu lintas, termasuk pejalan kaki, pesepeda, dan semua tipe kendaraan (dalam Risdiyanto, 2014). Ada beberapa hal yang perlu diperhatikan dalam mengatur lampu lalu lintas:

a. Fase Sinyal

Fase adalah rangkaian yang diberlakukan untuk satu atau lebih arus lalu lintas, dimana didalam siklus menerima perlakuan yang sama. Jumlah fase yang baik adalah fase yang menghasilkan kapasitas besar dan rata-rata tundaan lebih rendah.

b. Waktu Merah Semua

Waktu merah semua yang diperlukan untuk pengosongan pada akhir setiap fase harus memberi kesempatan bagi kendaraan terakhir (melewati garis henti pada akhir sinyal kuning) berangkat dari titik konflik sebelum kedatangan kendaraan yang datang pertama dari fase berikutnya (melewati garis henti pada awal sinyal hijau pada titik yang sama). Jadi merah semua merupakan fungsi dari kecepatan dan jarak dari kendaraan yang berangkat dan yang datang dari garis henti sampai ke titik konflik dan panjang dari kendaraan yang berangkat. Panjang waktu kuning pada sinyal lalu lintas perkotaan di Indonesia biasanya adalah 3,0 detik.

c. Waktu Siklus

Waktu siklus adalah waktu untuk urutan lengkap dan indikasi sinyal. Waktu siklus untuk fase, dapat dihitung dengan rumus atau gambar di bawah ini. Waktu siklus hasil penghitungan ini merupakan waktu siklus optimum, di mana akan menghasilkan tundaan terkecil.

$$C_o = \frac{(1.5 \times LT + 5)}{(1-IFR)} \dots\dots\dots(1)$$

Dimana :

C_o = Waktu siklus sinyal (detik)
 LT = Total waktu hilang per siklus (detik)
 IFR = Perbandingan arus persimpangan S

Berikut ini adalah waktu siklus yang ideal untuk fase yang berbeda-beda:

Table 1: Waktu siklus.

Tipe Kontrol	Waktu Siklus
2 fase	40 – 80
3 fase	50 – 100
4 fase	80 – 130

d. Waktu Hijau

Waktu hijau adalah waktu nyala hijau dalam suatu jalur. Penghitungan waktu hijau untuk tiap fase dijelaskan dengan rumus di bawah ini.

$$g_i = (c - LT) \times PR_i \dots\dots\dots(2)$$

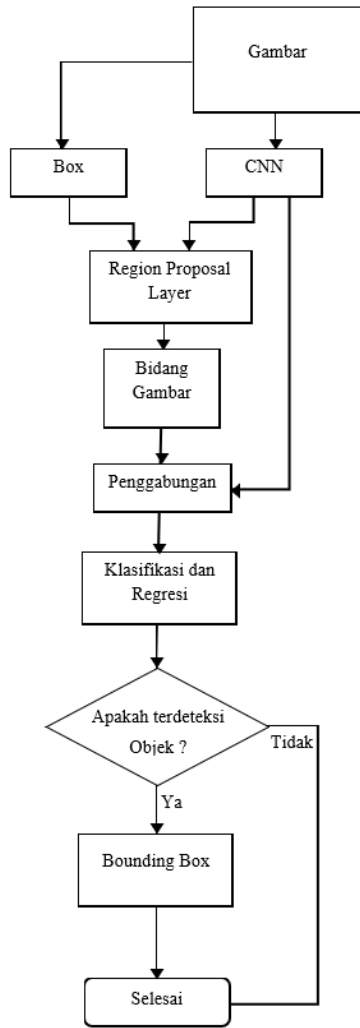
dengan :

g_i = Waktu hijau dalam fase-I (detik)
 c = Waktu siklus yang ditentukan (detik)
 LT = Total waktu hilang per siklus (detik)
 Pri = Perbandingan fase

2.5. Faster R-CNN

Penggunaan region proposal method seperti Selective Search masih menjadi kendala dalam proses deteksi objek menggunakan R-CNN, karena metode seperti ini membutuhkan waktu yang relatif lama untuk menghasilkan regions. Untuk menyelesaikan permasalahan diatas dikembangkan Faster R-CNN. Faster R-CNN menggunakan RPN (Region Proposal Network). RPN merupakan pengembangan dari region proposal yang telah dilatih secara optimal untuk menghasilkan region proposal dengan kualitas tinggi. RPN akan memprediksi ukuran objek dan nilai keakuratan pada tiap objek. Dengan menggunakan RPN, dapat melakukan pengajuan lebih cepat, yaitu 10 ms per gambar daripada region proposal dengan Selective Search yang hanya dapat melakukan pengajuan atau proposal 0.2 detik per gambar (Ren, He, Girshick, Sun, 2016). RPN adalah sebuah neural network yang menggantikan peran Selective Search untuk mengajukan region (bagian-bagian mana dari sebuah gambar yang perlu dilihat lebih jauh). RPN menghasilkan beberapa bounding box, setiap box memiliki 2 skor probabilitas apakah pada lokasi tersebut terdapat objek atau tidak. Penggunaan RPN untuk mengganti Selective Search ini mengurangi kebutuhan komputasi yang cukup signifikan, dan membuat keseluruhan model dapat di-train secara end-to-end. Selain itu, Faster R-CNN juga menghasilkan performa yang lebih cepat dan lebih akurat bila dibandingkan dengan Fast R-CNN, dan saat ini telah menjadi pilihan model yang umum digunakan untuk deteksi objek dengan solusi

berbasis deep learning. Berikut ini adalah diagram alir dari Faster R-CNN.



Gambar 3: Diagram alir faster r-cnn

3. HASIL DAN PEMBAHASAN

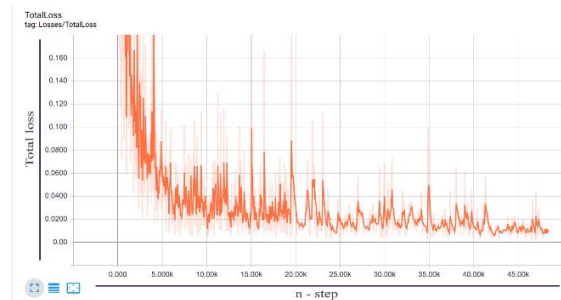
3.1. Hasil Training Dataset

Sumber dataset yang digunakan merupakan gambar mainan berbentuk kendaraan, jumlah gambar mainan yang digunakan adalah 153 buah gambar. Dalam pelatihan dataset terdapat total loss yang dijadikan acuan untuk mengetahui tingkat akurasi dari dataset tersebut. Total loss merupakan total dari keseluruhan loss atau error yang terjadi selama proses training dataset. Dalam proses training total loss dimulai pada nilai 4.1823, menurut Ren, dkk (2016) total loss harus berada pada nilai yang sangat kecil, yaitu kurang dari 0.05 agar objek dapat terdeteksi secara akurat. Dalam training dataset yang saya lakukan total loss mencapai nilai 0.0033 pada langkah ke 48160. Berikut ini adalah hasil dan grafik total loss yang ditunjukkan pada gambar 3 dan 4.

```

Anaconda Prompt: python train.py --logtostderr --train_dir=training --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config
INFO:tensorflow:global step 48158: loss = 0.0085 (1.244 sec/step)
INFO:tensorflow:global step 48159: loss = 0.0087 (1.255 sec/step)
INFO:tensorflow:global step 48160: loss = 0.0087 (1.255 sec/step)
INFO:tensorflow:global step 48161: loss = 0.0033 (1.219 sec/step)
INFO:tensorflow:global step 48162: loss = 0.0033 (1.219 sec/step)
INFO:tensorflow:global step 48163: loss = 0.0362 (1.234 sec/step)
INFO:tensorflow:global step 48164: loss = 0.0095 (1.241 sec/step)
INFO:tensorflow:global step 48165: loss = 0.0095 (1.241 sec/step)
INFO:tensorflow:global step 48166: loss = 0.0364 (1.250 sec/step)
INFO:tensorflow:global step 48167: loss = 0.0094 (1.219 sec/step)
INFO:tensorflow:global step 48168: loss = 0.0094 (1.219 sec/step)
INFO:tensorflow:global step 48169: loss = 0.0084 (1.234 sec/step)
INFO:tensorflow:global step 48170: loss = 0.0084 (1.234 sec/step)
INFO:tensorflow:global step 48171: loss = 0.0221 (1.255 sec/step)
INFO:tensorflow:global step 48172: loss = 0.0221 (1.256 sec/step)
INFO:tensorflow:global step 48173: loss = 0.0133 (1.250 sec/step)
INFO:tensorflow:global step 48174: loss = 0.0133 (1.250 sec/step)
INFO:tensorflow:global step 48175: loss = 0.0109 (1.405 sec/step)
INFO:tensorflow:global step 48176: loss = 0.0109 (1.405 sec/step)
INFO:tensorflow:Recording summary at step 48169.
INFO:tensorflow:Recording summary at step 48169.
INFO:tensorflow:global step 48170: loss = 0.0385 (1.294 sec/step)
INFO:tensorflow:global step 48171: loss = 0.0185 (1.284 sec/step)
INFO:tensorflow:global step 48172: loss = 0.0052 (1.237 sec/step)
INFO:tensorflow:global step 48173: loss = 0.0092 (1.237 sec/step)
  
```

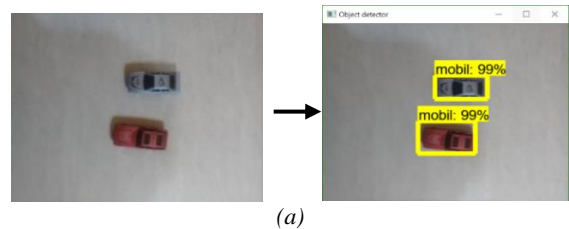
Gambar 4: Hasil training dataset



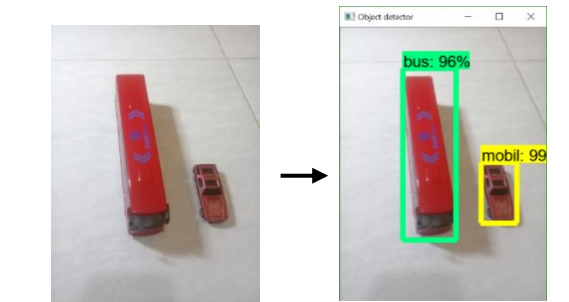
Gambar 5: Hasil total loss training dataset

3.2. Hasil Pengujian Akurasi

Gambar 5 dan 6 berikut ini adalah beberapa contoh pengujian akurasi deteksi objek.



(a)



(b)

Gambar 6: Contoh deteksi pada dua mobil(a) dan contoh deteksi mobil dan bus(b).

Berdasarkan dari tabel pengujian deteksi objek kendaraan terhadap 15 sampel gambar, persentase akurasinya adalah 97.027%. Sehingga, akurasi mencapai target yang diinginkan yaitu > 70%.

3.3. Pengujian Perhitungan

Pengujian yang dilakukan adalah berupa mendeteksi 20 jenis sampel gambar berbeda yang pada masing-masing gambar terdapat beberapa kendaraan, kemudian tingkat akurasi perhitungan akan ditentukan dari apakah seluruh kendaraan dalam gambar terdeteksi atau tidak. Dari pengujian ini dapat diketahui apakah sistem dapat menghitung kendaraan dengan benar, baik berdasarkan jenis kendaraan ataupun jumlah atau total kendaraan yang ada pada gambar tersebut. Gambar berikut ini adalah beberapa contoh hasil perhitungan yang dilakukan.



(a)



(b)

Gambar 7: Contoh perhitungan kondisi ramai(a) dan contoh perhitungan kondisi lengang(b).

Berdasarkan dari pengujian perhitungan objek kendaraan terhadap 20 sampel gambar yang berbeda, persentase terjadinya error adalah 2.188%. Sehingga, dataset bisa dianggap baik karena error yang relative kecil.

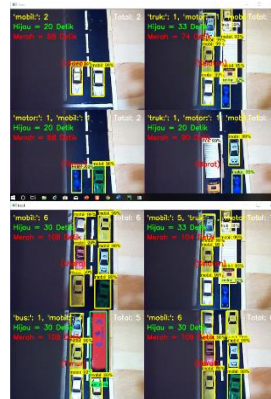
3.4. Pengujian Sistem

Pengujian sistem dilakukan pada simulasi jalan secara *real-time* dimana menggunakan 2 buah kamera webcam serta pengujian melalui input gambar berupa 4 buah jalur pada perempatan, dimana ditentukan jumlah objek atau kendaraan untuk mencapai kondisi padat adalah 5 buah objek atau kendaraan.. Berikut ini adalah gambar dalam pengujian secara *real-time* menggunakan 2 buah kamera usb.



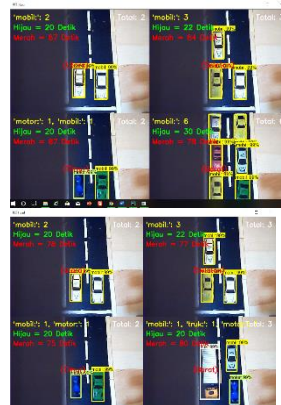
Gambar 8: Pengujian *real-time*

Kemudian dilakukan pengujian terhadap 4 input gambar yang berbeda-beda untuk setiap jalur. Kemudian lampu hijau dan merah akan menyesuaikan sesuai dengan tingkat kepadatan pada setiap jalur. Berikut ini adalah beberapa contoh pengujian yang dilakukan.



(a)

(b)



(c)

(d)

Gambar 9: Kondisi salah satu jalan padat (a) dan (c), kondisi semua jalan padat (b), kondisi semua jalan sepi (d)

4. PENUTUP

4.1. Kesimpulan

Berikut ini adalah kesimpulan dalam penelitian ini :

- a. Dataset dilatih dengan menggunakan framework tensorflow dengan akurasi deteksi 97.027%, dimana persentase akurasi tersebut lebih dari target yang ditentukan, yaitu lebih dari 70%.
- b. Persentase error kalkulasi kendaraan yang relative kecil yaitu 2.188%, sehingga sistem dapat mendeteksi jenis dari objek yang diinginkan berdasarkan masukan citra digital dengan baik

4.2. Saran

Adapun saran untuk penelitian yang selanjutnya adalah sebagai berikut:

- a. Dalam proses pelatihan dataset, akan lebih cepat apabila menggunakan gpu dengan tingkat compute capability lebih dari 6.0.
- b. Menggunakan lebih banyak foto sebagai sumber pelatihan dataset, agar didapatkan hasil yang lebih akurat dalam mendeteksi objek.

PERSEMBAHAN

Puji syukur dipanjatkan atas kehadiran Allah SWT, karena dengan limpahan karunia-Nya penulis dapat menyelesaikan penelitian ini.

Terima kasih kepada bapak M.S. Hendriyawan Achmad.,S.,M.Eng selaku Ketua Program Studi Teknik Elektro Fakultas Teknologi Informasi dan Elektro dan dosen pembimbing, serta seluruh dosen Program Studi Teknik Elektro, Universitas Teknologi Yogyakarta.

Terima kasih kepada berbagai pihak yang serta-merta ikut membantu dalam penyusunan penelitian ini.

DAFTAR PUSTAKA

- [1] Alamsyah, D. dan Pratama, D. (2018), Deteksi Ujung Jari menggunakan Faster-RCNN dengan Arsitektur Inception v2 pada Citra Derau, Jurnal Sistem & Teknologi Informasi Komunikasi, 2(1)..
- [2] Binkot, D. (1997), Manual Kapasitas Jalan Indonesia (MKJI), P.T BINA KARYA (PERSERO)..
- [3] Deng, L. (2014), Deep Learning: Methods and Applications, Foundations and Trends® in Signal Processing, 7(3–4), 197–387.
- [4] Fukushima, K. (1980), Biological Cybernetics, Biological Cybernetics, 202(36), 193–202.
- [5] Girshick, R. (2015), Fast R-CNN, Microsoft Research.
- [6] Girshick, R., Donahue, J., Darrell, T., Malik, J. dan Mercan, E. (2014), R-CNN for Object Detection, IEEE Conference.
- [7] Kusumanto, R.D. dan Tompunu, A.N. (2011), PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB, , 2011(Semantik)..
- [8] Lazaro, A., Buliali, J.L. dan Amaliah, B. (2017), Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV, Jurnal Teknik ITS, 6(2).
- [9] Lecun, Y., Bengio, Y. dan Hinton, G. (2015), Deep learning..
- [10] Malik, J., Arbeláez, P., Carreira, J., Fragkiadaki, K., Girshick, R., Gkioxari, G., ... Tulsiani, S. (2016), *The three R ' s of computer vision : Recognition, reconstruction, and reorganization*, 72, 4–14.
- [11] P, I.W.S.E., Wijaya, A.Y. dan Soelaiman, R. (2016), Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101, , 5(1).
- [12] Plemakova, V. (2018), *Vehicle Detection Based on Convolutional Neural Networks*, University of Tartu..
- [13] Ren, S., He, K. dan Girshick, R. (2015), Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks, , 1–9.
- [14] Risdiyanto (2014), *Rekayasa & Manajemen Lalu Lintas Teori dan Aplikasi*, Andayani, Ed. Yogyakarta: LeutikaPrio.
- [15] Yilmaz, A.A., Guzel, M.S., Askerbeyli, I. dan Bostanci, E. (2018), A Vehicle Detection Approach using Deep Learning Methodologies, Computer Engineering Department.

