

Implementasi Deep Learning untuk Object Detection Menggunakan Algoritma YOLO (You Only Look Once) pada Rambu Lalu Lintas di Indonesia

Adhy Wiranto Sudjana
Teknik Informatika
Universitas Pasundan
Bandung Indonesia
adhy_173040038@mail.unpas.ac.id

Handoko Supeno
Teknik Informatika
Universitas Pasundan
Bandung Indonesia
handoko@unpas.ac.id

Sejauh yang diamati, belum ada pustaka *dataset* yang memiliki *dataset* rambu lalu lintas di Indonesia secara komprehensif. Pustaka *dataset* diperlukan dalam pendeteksian objek rambu lalu lintas di Indonesia, pada penelitian ini bertujuan untuk pengumpulan *dataset* rambu lalu lintas khas Indonesia dan diteliti dengan menggunakan salah satu algoritma Deep Learning yang khusus untuk pendeteksian objek secara *real-time*.

Dataset yang dikumpulkan berupa gambar rambu lalu lintas yang melingkupi 21 jenis rambu dengan total gambar sebanyak 2100 buah gambar. Algoritma yang digunakan pada pendeteksian objek ini menggunakan YOLOv4 (You Only Look Once) karena algoritma ini merupakan salah satu yang paling unggul dalam pendeteksian objek secara *real-time* berdasarkan akurasi dan jumlah *frame per second*.

Pendeteksian objek menghasilkan akurasi model sebesar 95.63%, dimana model ini digunakan pada file video dan berhasil mendeteksi objek rambu lalu lintas dengan baik.

Kata Kunci— Object Detection, Deep Learning, YOLO, CNN, Rambu Lalu Lintas.

I. PENDAHULUAN

Untuk mewujudkan implementasi Computer Vision di Indonesia khususnya dalam penelitian *Autonomous Driving Car* dibutuhkan kontribusi dalam berbagai aspek. Salah satu kontribusi yang dilakukan pada penelitian ini adalah pengumpulan *dataset* rambu lalu lintas di Indonesia yang dapat dimanfaatkan untuk pendeteksian rambu lalu lintas di Indonesia. *Dataset* adalah kumpulan data yang telah diberikan label atau informasi dan telah dikelompokkan berdasarkan kelompok datanya [10].

Sejauh yang diamati, belum ada pustaka *dataset* yang memiliki *dataset* rambu lalu lintas di Indonesia secara komprehensif, ini juga merupakan alasan mengapa penelitian ini dilakukan. *Dataset* yang dikumpulkan berupa kumpulan gambar rambu lalu lintas khas Indonesia seperti rambu pemberhentian bus, rambu pemberitahuan lokasi putar balik, dan rambu peringatan pintu perlintasan kereta api. Penelitian ini juga meneliti bagaimana penggunaan *dataset* yang dikumpulkan terhadap pendeteksian objek rambu lalu lintas khas Indonesia menggunakan algoritma Deep Learning yaitu You Only Look Once versi 4.

II. PENELITIAN TERDAHULU

Berikut merupakan penelitian-penelitian yang pernah dilakukan sebelumnya untuk pendeteksian objek menggunakan algoritma YOLO :

Tabel 1 Penelitian Terdahulu

No	Penulis	Dataset	Keterangan
1	Juanola, M. S. [JUA19]	3 jenis Speed Traffic Sign yaitu 30km/jam, 60km/jam, dan 90km/jam sebanyak total 5900 citra.	Menghasilkan akurasi sebesar 86.89% untuk rambu 30km/jam, 88.09% untuk rambu 60km/jam, dan 100.00% untuk rambu 90km/jam.
2	Wisna H, J. S., Matulatan, T., dan Hayaty, N. [WIS20]	4 kelas kendaraan dengan total 200 citra.	Menghasilkan akurasi pendeteksian sebesar 83.3%
3	Melek, C. G., Sonmez, E. B., Albayrak, S. [MEL19]	200 citra minuman coca-cola. Lalu ada 10 kelas untuk merk rokok. Kedua hal tersebut dideteksi secara terpisah.	Untuk citra coca-cola menghasilkan loss sebesar 2.63% pada iterasi ke 2000. Untuk 10 kelas rokok, pada iterasi training sebesar 40000 menghasilkan total loss sebesar 8.22%.
4	He, Y. [HE16]	Dataset karya lukis Picasso dengan 218 citra.	Dalam pendeteksian objek pada karya lukis Picasso menghasilkan tingkat presisi/akurasi sebesar 40%.
5	Putra, M. H., Yussof, Z. M., Lim, K. C., Salim, S. I. [PUT18]	Dataset yang berisi 2 kelas manusia dan mobil dengan total 162 citra untuk training dan 162 citra untuk testing.	Menghasilkan mean Average Precision sebesar 59.2% dengan tingkat Average Precision untuk masing masing kelas yaitu kelas Manusia dengan AP 63.1% dan kelas Mobil yaitu 55.3%.

III. PENDEKATAN OBJECT DETECTION

A. Gambaran Umum Lalu Lintas di Indonesia

Undang-undang Republik Indonesia Nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan mendefinisikan sebagai gerak kendaraan dan orang di Ruang

Lalu Lintas Jalan, sedang yang dimaksud dengan Ruang Lalu Lintas Jalan adalah prasarana yang diperuntukkan bagi gerak pindah kendaraan, orang, dan/atau barang yang berupa jalan dan fasilitas pendukung [18].

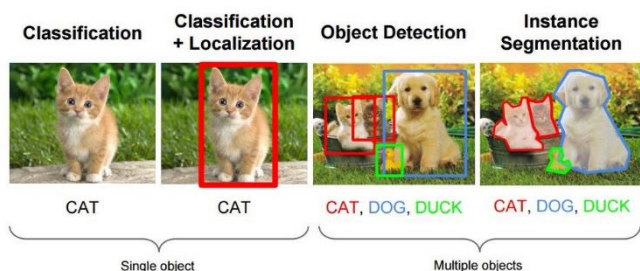
Berdasarkan Peraturan Menteri Perhubungan Republik Indonesia nomor PM 13 tahun 2014 Bab I pasal 1 tentang Ketentuan Umum menyatakan bahwa rambu lalu lintas adalah bagian perlengkapan jalan yang berupa lambang, huruf, angka, kalimat, dan/atau perpaduan yang berfungsi sebagai peringatan, larangan, perintah, atau petunjuk bagi pengguna jalan [8].

Pada Bab II tentang Spesifikasi Teknis Rambu Lalu Lintas bahwa rambu lalu lintas berdasarkan jenisnya terdiri atas 4 jenis [8] :

1. Rambu peringatan
2. Rambu larangan
3. Rambu perintah, dan
4. Rambu petunjuk

B. Pendekatan Computer Vision

Computer Vision melakukan suatu *Pattern Recognition* atau pengenalan pola pada suatu citra/gambar. Misalkan komputer dilatih dengan memberikan 100 buah gambar kucing dengan jenis yang berbeda-beda, namun kucing memiliki ciri khasnya tersendiri dimana kucing memiliki kumis, telinga yang berbentuk segitiga, dan berkaki empat. Dari ciri khas tersebut, komputer bertanggungjawab untuk “belajar” mengenal pola dari fitur/ciri yang dimiliki kucing tersebut yang nantinya jika komputer diberikan gambar kucing yang berbeda misalkan warna yang berbeda atau jenis yang belum pernah dilatih sebelumnya, komputer dapat dengan baik mengenal bahwa gambar tersebut adalah kucing. Jika komputer diberikan gambar anjing tentu komputer belum dapat mengenal hewan tersebut karena komputer belum belajar mengenal pola dari bentuk anjing itu sendiri, ini merupakan salah satu contoh agar pengenalan pola dari objek yang beragam dapat dilakukan maka komputer juga harus dilatih dengan data yang beragam.



Gambar 1 Pembagian Tugas Computer Vision

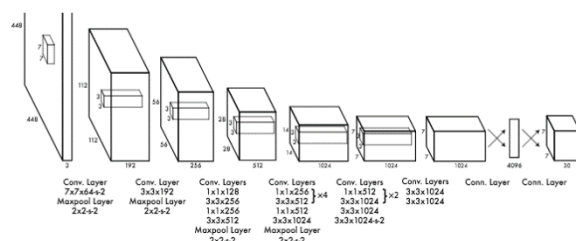
Pada Gambar 1 di atas menggambarkan tugas-tugas yang dilakukan oleh Computer Vision yang dibagi menjadi 4 yaitu Classification, Classification + Localization, Object Detection, dan Instance Segmentation. Algoritma Classification melakukan klasifikasi setiap gambar yang diberikan dimana setiap 1 gambar yang diberikan komputer akan mengklasifikasikannya dengan 1 kelas/informasi yang

paling sesuai dengan gambar tersebut. Classification + Localization mirip dengan Classification biasa yang diberikan lokalisasi berupa informasi tambahan yaitu *bounding box* yang memberi informasi letak dari objek yang diklasifikasikan. Lalu ada algoritma yang melakukan pendeteksian lebih dari 1 objek yaitu Object Detection. Object Detection dapat melakukan pendeteksian objek yang berada pada 1 gambar yang sama dan jenis objek yang berbeda-beda. Seperti gambar diatas terdapat gambar berisikan 2 anak kucing, 1 anjing, dan 1 mainan bebek, algoritma Object Detection mendeteksi objek apa yang ada pada gambar dan memberikan *bounding box* pada setiap objek tersebut. Lalu yang keempat adalah Instance Segmentation, algoritma jenis ini memiliki kelebihan dibandingkan dengan Object Detection yaitu dapat mengetahui bentuk dari objek itu sendiri.

Computer Vision sudah banyak diterapkan dalam kehidupan sehari-hari, misalkan pada aplikasi kamera *smartphone* terdapat fitur Face Recognition untuk *selfie* sehingga hasil foto menjadi lebih bagus, ataupun hal yang lebih baru misalkan mobil listrik yang dikeluarkan oleh perusahaan Tesla, memakai banyak teknologi salah satunya adalah Object Detection untuk mendeteksi objek di sekitarnya untuk menentukan berbagai hal seperti mengatur laju mobil, berhenti, belok ke arah kanan dan kiri, dan juga dapat memberi peringatan terhadap objek yang ada di depannya.

C. Pendekatan Algoritma YOLO (You Only Look Once)

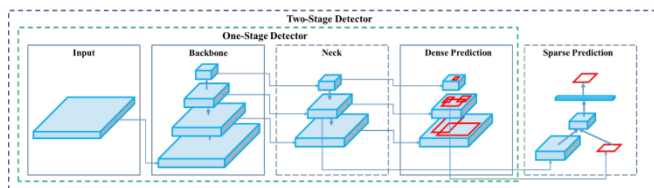
YOLO (You Only Look Once) merupakan algoritma yang dibangun berdasarkan arsitektur CNN (Convolutional Neural Network) berguna untuk pendeteksian objek secara *real-time*. Yolo pada awalnya dikembangkan oleh Joseph Redmon dan Ali Farhadi pada 2015. YOLO terinspirasi dari GoogleNet yang merupakan model yang digunakan untuk klasifikasi gambar [RED15]. YOLO versi 1 memiliki arsitektur jaringan yang terdiri dari 24 *convolutional layer* dan 2 *fully connected layer* [11]. Pendeteksian YOLO memisahkan gambar kedalam grid $S \times S$. Jika titik tengah dari suatu objek cocok dengan grid cell, maka grid cell tersebut harus mendeteksi objeknya. Berikut Gambar 2 di bawah ini merupakan arsitektur awal dari algoritma YOLO.



Gambar 2 Arsitektur YOLOv1

Versi terbaru saat laporan ini ditulis adalah versi 4 yang dikembangkan oleh 3 pengembang baru pada 24 april 2020 oleh Alexey Bochkovskiy, Chien-Yao Wang, dan Hong-Yuan Mark Liao. Joseph Redmon dan Ali Farhadi tidak melanjutkan pengembangan YOLO versi 4 dikarenakan

kekhawatiran mereka terhadap penggunaan YOLO oleh militer dan masalah privasi maka pengembangan selanjutnya digantikan oleh 3 author diatas. YOLOv4 memiliki peningkatan dalam hal kecepatan dan performance yang sangat signifikan jika dibandingkan dengan YOLOv3. YOLOv4 memiliki arsitektur jaringan yang berbeda dari versi sebelumnya. Arsitekturnya dibagi menjadi 3 bagian yaitu Backbone: CSPDarknet53, Neck: SPP, dan Head: YOLOv3 [1]. YOLO versi 4 merupakan bagian dari One-Stage Detector dimana jenis detector ini memiliki kelebihan dalam kecepatan dibandingkan algoritma dengan jenis Two-Stage Detector karena memiliki tahap tambahan yaitu Sparse Prediction sebelum suatu objek terdeteksi [2]. Berikut Gambar 3 di bawah ini adalah arsitektur detektor yang digunakan oleh YOLO versi 4.



Gambar 3 Arsitektur Detektor

Arsitektur YOLO memiliki 2 bagian utama yaitu Backbone dan Head. Pertama kali dilakukannya pendeteksian adalah memasukkan gambar atau video yang berada pada bagian input yang akan diteruskan ke bagian Backbone. Backbone merupakan “tulang belakang” yang berguna untuk meningkatkan akurasi sebelum dilakukannya pendeteksian di bagian Head. Lalu Neck merupakan perpanjangan dari Backbone yang digunakan untuk mengumpulkan feature maps dari stage yang berbeda setelah Backbone. Lalu bagian Head adalah bagian utama pada pendeteksian YOLO ini dimana bagian ini menggunakan YOLO versi 3. YOLO akan mengambil input berupa gambar lalu di-resize menjadi ukuran 416x416, gambar yang telah di-resize akan dibaca menjadi matriks lalu dilakukan operasi-operasi pada layer convolutional dan layer pooling. Layer Convolutional merupakan layer dimana gambar dalam bentuk matriks berisi nilai-nilai antara 0.00 sampai 1.00 akan dilakukan perhitungan matriks dengan filter 3x3 dimana hasilnya akan membentuk nilai-nilai yang baru, hal ini diteruskan sampai 106 kali, setiap 7 layer akan dilakukan downsampling dimana ukuran matriks akan dikurangi setengahnya hingga ukuran matriks menyusut menjadi 13x13. Matriks 13x13 akan dibagi menjadi 169 grid cell dimana setiap cell merupakan 1 bagian dari 169 titik. Cell yang telah dibagi masing-masing akan dilakukan pendeteksian dan akan menghasilkan bounding box beserta prediksi kelasnya.

Setelah membuat bounding box maka akan dilakukan kualifikasi dengan batas/threshold yang ditentukan, misalkan dari nilai 0.0 sampai 1.0 akan ditentukan threshold sebesar 0.6, maka bounding box dengan prediksi kelas yang memiliki nilai confidence dibawah 0.6 tidak akan ditampilkan. Hasil dari pendeteksian YOLO berupa bounding box (berbentuk persegi) dengan nilai confidence dan nama kelas yang terdeteksi. Bounding box terdiri dari variabel x, y, w, h, dan confidence, variabel x dan y untuk koordinat tengah dari

bounding box, w dan h adalah ukuran piksel dari suatu gambar, dan confidence adalah nilai Intersection Over Union (IOU) antara *predicted box* dengan *ground truth* atau dalam kata lain kelas gambar yang terdeteksi [12].

IV. PERSIAPAN DAN PELATIHAN DATASET





A. Persiapan Dataset

Rambu yang dikumpulkan merupakan rambu-rambu lalu lintas yang ada di Indonesia meliputi 4 kelompok rambu dengan jenis rambu setiap kelompoknya ditambahkan lampu pemberhentian, total jenis rambu yang dikumpulkan adalah sebanyak 21 jenis dengan rincian sebagai berikut:

Tabel 2 Rincian Rambu Lalu Lintas

No	Kelompok Rambu	Nama Rambu	Jumlah	Gambar
1		Larangan Parkir	100 Buah	
2		Larangan Berhenti	100 Buah	
3		Larangan Masuk Bagi Kendaraan Bermotor dan Tidak Bermotor	100 Buah	
4	Rambu Larangan	Larangan Memutar Balik	100 Buah	
5		Larangan Belok Kanan	100 Buah	
6		Larangan Belok Kiri	100 Buah	
7		Larangan Berjalan Terus Wajib Berhenti Sesaat	100 Buah	

No .	Kelompok Rambu	Nama Rambu	Jumlah	Gambar
8	Rambu Peringatan	Peringatan Banyak Pejalan Kaki Menggunakan Zebra Cross	100 Buah	
9		Peringatan Alat Pemberi Isyarat Lalu Lintas	100 Buah	
10		Peringatan Pintu Perlintasan Kereta Api	100 Buah	
11		Peringatan Simpang Tiga Sisi Kiri	100 Buah	
12		Peringatan Penegasan Rambu Tambahan	100 Buah	
13	Rambu Perintah	Perintah Masuk Jalur Kiri	100 Buah	
14		Perintah Pilihan Memasuki Salah Satu Jalur	100 Buah	
15	Rambu Petunjuk	Petunjuk Lokasi Putar Balik	100 Buah	
16		Petunjuk Penyeberangan Pejalan Kaki	100 Buah	
17		Petunjuk Lokasi Pemberhentian Bus	100 Buah	

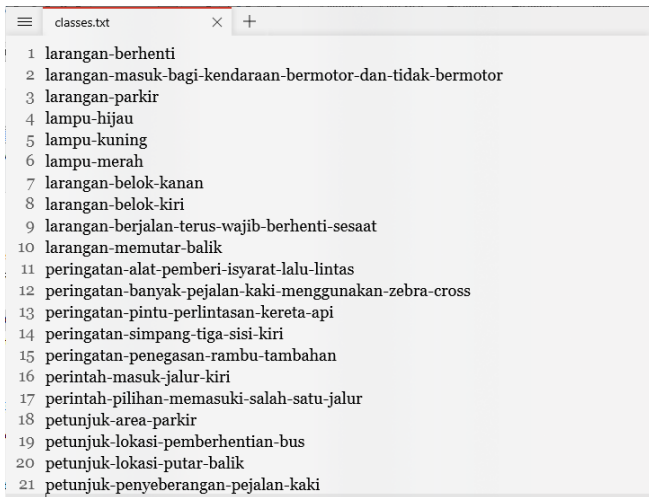
No .	Kelompok Rambu	Nama Rambu	Jumlah	Gambar
18		Petunjuk Area Parkir	100 Buah	
19	Lampu Pemberhentian	Lampu Merah	100 Buah	
20		Lampu Kuning	100 Buah	
21		Lampu Hijau	100 Buah	

Dataset rambu yang dikumpulkan adalah foto-foto rambu lalu lintas yang diambil melalui beberapa sumber, mulai dari *Google Maps*, *Mapillary*, dan pengambilan gambar melalui kamera smartphone.

Setelah 2100 buah gambar rambu telah dikumpulkan, tahap selanjutnya adalah melakukan pengelompokan per jenis rambu, setiap gambar dikelompokkan untuk setiap jenisnya di dalam folder, total ada 21 folder dimana masing-masing folder berisi 100 buah gambar rambu. Gambar rambu yang telah dikelompokkan harus dibagi lagi menjadi 2 kelompok untuk *training* dan *validation*, masing-masing kelompok ini berisi 21 jenis rambu dengan rasio pembagian sebesar 70:30 gambar per jenis rambu, jadi jumlah akhirnya adalah sebanyak 1470 buah gambar untuk *training* dan 630 buah gambar untuk *validation*. Kelompok pertama dengan folder **obj** digunakan pada saat pelatihan untuk membuat model lalu proses pelatihan dilanjutkan dengan mengevaluasi model yang dihasilkan pada proses *training* menggunakan kelompok gambar dengan folder **test**.

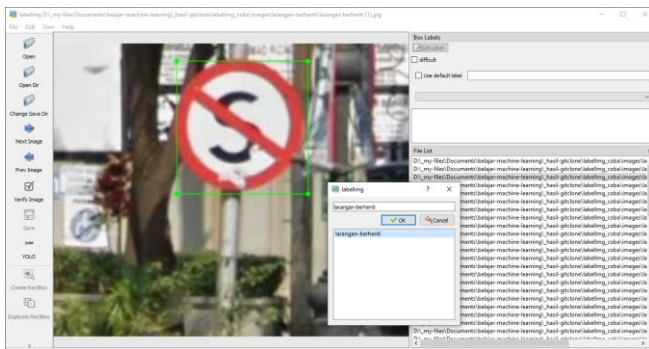
B. Pemberian Label pada Dataset

Sebelum dapat diberikan label, dibutuhkan perangkat lunak yang bernama `labelImg`. `labelImg` membutuhkan `file` bernama `classes.txt` yang berisikan daftar jenis rambu dimana `file` ini akan digunakan sebagai referensi `labelImg` untuk melakukan pemberian label dataset. Gambar 4 di bawah ini menggambarkan isi dari `file classes.txt`.



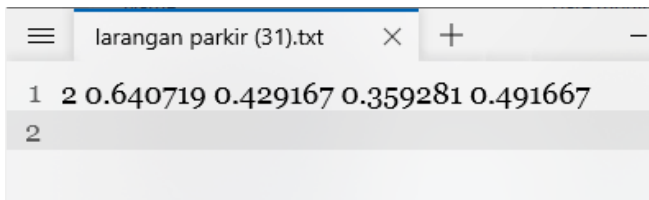
Gambar 4 Daftar Kelas

Untuk melakukan pemberian label yaitu dengan cara memberikan bounding box pada objek di dalam gambar lalu diberikan kelas. Gambar 5 di bawah ini menampilkan cara pemberian label dengan *tool* labelImg.



Gambar 5 Pemberian Label

Hasil dari proses pemberian label akan menghasilkan file berekstensi *.txt* yang berisi informasi nomor indeks jenis rambu yang merferensikan pada file *classes.txt*, koordinat dari bounding box, dan ukuran dari bounding box. Berikut Gambar 6 di bawah ini merupakan contoh hasil dari pelabelan dataset.



Gambar 6 Isi Label Dataset

Agar pelatihan dapat dilakukan, dibutuhkan 2 file berekstensi *.txt* dengan nama *train.txt* dan *test.txt* yang masing-masing berisi lokasi file gambar dataset untuk

kelompok *training* dan *validation* sehingga saat proses pelatihan model, file gambar dapat dibaca oleh algoritma YOLO berdasarkan 2 file tersebut.

C. Persiapan Pelatihan Dataset

Tahap selanjutnya adalah melakukan persiapan untuk pelatihan, ada 7 file yang perlu dipersiapkan sebelum melakukan prosedur pelatihan dataset dimana akan diupload ke Google Drive yang selanjutnya akan digunakan pada Google Colab.

Tabel 3 Daftar File Persiapan Dataset

No.	Nama File	Keterangan
1	Obj.zip	File yang berisi <i>dataset</i> untuk <i>training</i>
2	Test.zip	File yang berisi <i>dataset</i> untuk <i>validation</i>
3	Yolov4-obj.cfg	File konfigurasi algoritma YOLOv4
4	Obj.names	File yang berisikan daftar jenis rambu, isi sama dengan file <i>classes.txt</i>
5	Obj.data	File yang berisi informasi mengenai jumlah kelas, lokasi file daftar dataset (<i>train.txt</i> dan <i>test.txt</i>), lokasi file <i>obj.names</i> , dan lokasi backup hasil pelatihan <i>dataset</i> pada Google Drive
6	Train.txt	File yang berisi daftar <i>dataset</i> untuk <i>training</i>
7	Test.txt	File yang berisi daftar <i>dataset</i> untuk <i>validation</i>

D. Pelatihan Dataset

Selanjutnya proses pelatihan dataset dapat dilakukan dengan menggunakan file *obj.data*, file konfigurasi yolo, dan file pre-trained weight. File *obj.data* digunakan sebagai referensi jumlah kelas yang dideteksi, lokasi file dataset, dan lokasi penyimpanan file weight hasil pelatihan, file konfigurasi yolo digunakan sebagai konfigurasi pelatihan, dan file *yolov4.conv.137* digunakan untuk meningkatkan hasil pelatihan sehingga meningkatkan akurasi pendeteksian objek. Berikut Gambar 7 di bawah ini merupakan *capture* dari proses pelatihan dataset.

```

CUDA-version: 10010 (11020), cudNN: 7.6.5, CUDNN_HALF=1, GPU count: 1
CUDNN_HALF=1
OpenCV version: 3.2.0
yolov4-obj
  0 : compute_capability = 750, cudnn_half = 1, GPU: Tesla T4
net.optimized_memory = 0
mini_batch = 4, batch = 64, time_steps = 1, train = 1
layer  filters  size/strd(dil)  input  output
0 conv  32  3 x 3/ 1  416 x 416 x 3 -> 416 x 416 x 32 0.299 BF
1 conv  64  3 x 3/ 2  416 x 416 x 32 -> 208 x 208 x 64 1.595 BF
2 conv  64  1 x 1/ 1  208 x 208 x 64 -> 208 x 208 x 64 0.354 BF
3 route  1
4 conv  64  1 x 1/ 1  208 x 208 x 64 -> 208 x 208 x 64 0.354 BF
5 conv  32  1 x 1/ 1  208 x 208 x 64 -> 208 x 208 x 32 0.177 BF
6 conv  64  3 x 3/ 1  208 x 208 x 32 -> 208 x 208 x 64 1.595 BF
7 Shortcut Layer: 4, wt = 0, wn = 0, outputs: 208 x 208 x 64 0.003 BF
8 conv  64  1 x 1/ 1  208 x 208 x 64 -> 208 x 208 x 64 0.354 BF
9 route  8 2
10 conv  64  1 x 1/ 1  208 x 208 x 128 -> 208 x 208 x 64 0.789 BF
11 conv  128  3 x 3/ 2  208 x 208 x 64 -> 104 x 104 x 128 1.595 BF
12 conv  64  1 x 1/ 1  104 x 104 x 128 -> 104 x 104 x 64 0.177 BF
13 route  11
14 conv  64  1 x 1/ 1  104 x 104 x 128 -> 104 x 104 x 64 0.177 BF
15 conv  64  1 x 1/ 1  104 x 104 x 64 -> 104 x 104 x 64 0.089 BF
16 conv  64  3 x 3/ 1  104 x 104 x 64 -> 104 x 104 x 64 0.797 BF

```

Gambar 7 Proses Pelatihan

E. Hasil Eksperimen

Alasan mengapa penelitian ini menggunakan algoritma YOLOv4 dibandingkan YOLOv3 adalah selain versinya lebih baru juga tingkat pendeteksiannya lebih baik. Berikut adalah diagram eksperimen yang membandingkan kemampuan pendeteksian YOLOv3 dengan YOLOv4 pada *trained-weight* iterasi ke 10000 dan 20000.

No	Nama Rambu	YOLOv3		YOLOv4	
		10000	20000	10000	20000
1	larangan-berhenti	✗	✓	✓	✓
2	larangan-masuk-bagi-kendaraan-bermotor-dan-tidak-bermotor	✗	✗	✓	✗
3	larangan-parkir	✓	✓	✓	✓
4	lampu-hijau	✓	✓	✓	✓
5	lampu-kuning	✓	✓	✓	✓
6	lampu-merah	✓	✓	✓	✓
7	larangan-belok-kanan	✓	✓	✓	✓
8	larangan-belok-kiri	✓	✓	✓	✓
9	larangan-berjalan-terus-wajib-berhenti-sesaat	✓	✓	✓	✓
10	larangan-memutar-balik	✓	✓	✓	✓
11	peringatan-alat-pemberi-isyarat-lalu-lintas	✓	✓	✓	✓
12	peringatan-banyak-pejalan-kaki-menggunakan-zebra-cross	✗	✗	✗	✗
13	peringatan-pintu-perlintasan-kereta-api	✓	✓	✓	✓
14	peringatan-simpang-tiga-sisi-kiri	✓	✓	✓	✓
15	peringatan-penegasan-rambu-tambahan	✓	✓	✓	✓
16	perintah-masuk-jalur-kiri	✗	✗	✗	✗
17	perintah-pilihan-memasuki-salah-satu-jalur	✓	✓	✓	✓
18	petunjuk-area-parkir	✓	✓	✗	✗
19	petunjuk-lokasi-pemberhentian-bus	✓	✓	✓	✓
20	petunjuk-lokasi-putar-balik	✗	✗	✗	✗
21	petunjuk-penyeberangan-pejalan-kaki	✗	✗	✗	✗
Jumlah Terdeteksi		14	14	13	14

Keterangan

- ✓ Terdeteksi
- ✗ Salah mendeteksi
- ✗ Tidak terdeteksi

Gambar 8 Kemampuan Pendeteksian Algoritma

Sebagai perbandingan, 21 gambar baru di luar dataset diuji coba terhadap algoritma. Ukuran ruang deteksi YOLOv3 diatur sebesar 608x608 sedangkan YOLOv4 sebesar 416x416. Dapat dilihat pada Gambar 8 di atas, kemampuan pendeteksian YOLOv4 sedikit lebih baik dalam jumlah total, pada pendeteksian objek berlaku bahwa kemampuan pendeteksian akan meningkat jika ruang deteksinya semakin besar, dengan ruang deteksi yang lebih kecil YOLOv4 mampu menyaingi YOLOv3. Untuk hasil akhir pelatihan dataset dengan YOLOv4 dijelaskan pada Gambar 9 di bawah ini,

No	Nama rambu	AP (%)				
		Batch 10000	Batch 20000	Batch 30000	Batch 42000	Best Batch
1	larangan-berhenti	99,70	99,7	96,67	96,67	96,67
2	larangan-masuk-bagi-kendaraan-bermotor-dan-tidak-bermotor	100,00	100,00	100,00	100,00	100,00
3	larangan-parkir	84,96	89,23	82,41	88,56	92,93
4	lampu-hijau	93,38	93,77	94,85	94,1	93,85
5	lampu-kuning	82,72	72,66	65,21	73,27	73,78
6	lampu-merah	74,17	71,26	70,61	75,98	76,26
7	larangan-belok-kanan	93,87	92,22	90,32	92,66	92,66
8	larangan-belok-kiri	93,55	98,32	99,25	98,16	98,27
9	larangan-berjalan-terus-wajib-berhenti-sesaat	94,12	93,94	96,89	94,03	94,03
10	larangan-memutar-balik	96,67	94,89	96,33	96,45	96,45
11	peringatan-alat-pemberi-isyarat-lalu-lintas	100,00	100,00	99,89	100,00	100,00
12	peringatan-banyak-pejalan-kaki-menggunakan-zebra-cross	100,00	100,00	100,00	100,00	100,00
13	peringatan-pintu-perlintasan-kereta-api	100,00	100,00	100,00	100,00	100,00
14	peringatan-simpang-tiga-sisi-kiri	100,00	100,00	99,89	100,00	100,00
15	peringatan-penegasan-rambu-tambahan	100,00	100,00	100,00	100,00	100,00
16	perintah-masuk-jalur-kiri	100,00	100,00	100,00	100,00	100,00
17	perintah-pilihan-memasuki-salah-satu-jalur	96,67	96,67	96,67	96,67	96,67
18	petunjuk-area-parkir	100,00	100,00	100,00	100,00	100,00
19	petunjuk-lokasi-pemberhentian-bus	100,00	100,00	100,00	100,00	100,00
20	petunjuk-lokasi-putar-balik	96,67	96,67	96,67	96,67	96,67
21	petunjuk-penyeberangan-pejalan-kaki	100,00	100,00	100,00	100,00	100,00
mAP (%)		95,64	95,21	94,56	95,39	95,63

Gambar 9 Akurasi Pendeteksian YOLOv4

F. Hasil Pelatihan

Tahap pelatihan dataset memakan waktu kira-kira selama 70 jam dengan melatih dataset berupa rambu lalu lintas sebanyak 2100 buah. Hasil yang didapat dari tahap pelatihan berupa model yang berbentuk file berekstensi `.weights` dimana akan digunakan untuk tahap pendeteksian objek. Akurasi untuk setiap jenis rambu bervariasi dengan akurasi terkecil sebesar 73.78% untuk lampu kuning dan akurasi terbesar adalah 100% untuk rambu petunjuk area parkir, dan akurasi keseluruhan adalah sebesar 95.63%.

V. PENGUJIAN DATASET

Setelah dilakukannya proses pelatihan dataset, hasil berupa file `.weights` digunakan untuk tahap pengujian *dataset*. pengujian pertama dilakukan terhadap file foto dan dilanjutkan dengan file video.

A. Pengujian dengan File Gambar

Contoh pengujian menggunakan file gambar yang berisi objek rambu petunjuk lokasi putar balik, berikut hasilnya dengan tingkat akurasi pendeteksian sebesar 100%. Gambar 8 di bawah ini adalah hasil pengujian menggunakan file gambar.



Gambar 10 Hasil Pengujian File Gambar

B. Pengujian dengan File Video

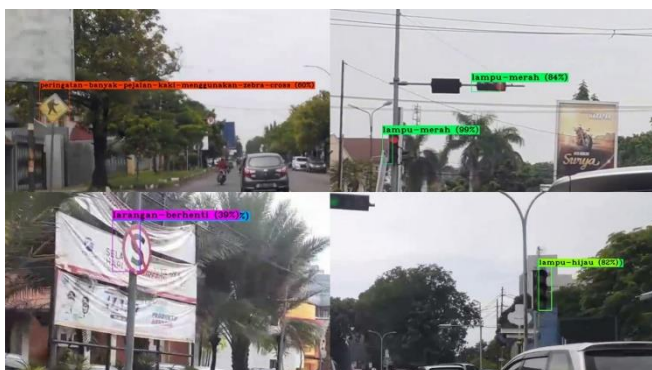
Pengujian selanjutnya adalah dengan menggunakan file video, File video yang disiapkan akan digunakan untuk menguji algoritma, hasil dari pengujian akan menghasilkan video dengan nama video_hasil_pengujian.avi yang telah diberikan bounding box hasil pendeteksian objek. Berikut hasil *screenshot* dari file video yang telah diuji.



Gambar 11 Hasil Pengujian File Video

Pada Gambar 9 di atas terdapat contoh hasil *scene* pendeteksian objek pada file video yang memperlihatkan beberapa rambu lalu lintas dengan posisi yang berbeda-beda. Untuk rambu larangan berhenti salah dideteksi sebagai larangan belok kiri, dan lampu hijau juga terdapat ketidaktepatan pendeteksian dimana terdeteksi lampu merah.

Dengan menggunakan file konfigurasi yang telah dibuat sebelumnya, hasil pendeteksian objek pada video masih kurang baik. Maka dari itu konfigurasi algoritma dirubah sedikit agar kemampuan pendeteksian objek yolo terhadap rambu lalu lintas menjadi lebih baik. Hiperparameter width dan height yang sebelumnya sebesar 416x416 dibesarkan ukurannya menjadi 608x608 sehingga pendeteksian menjadi lebih baik, hiperparameter adalah parameter yang ada pada algoritma pendeteksian YOLO yang berguna sebagai konfigurasi untuk pelatihan dan pendeteksian, contoh dari hiperparameter seperti width dan height, max_batches, dan filter. Pada Gambar 10 di bawah ini adalah hasil dari perubahan nilai hiperparameter width dan height pada pendeteksian objek rambu lalu lintas,



Gambar 12 Hasil Pengujian File Video Setelah Dikonfigurasi Ulang

Setelah dikonfigurasi ulang, pada Gambar 10 di atas memperlihatkan *scene* yang sama dengan gambar sebelumnya, dimana pendeteksian rambu larangan berhenti dan lampu hijau terdeteksi dengan baik.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan yang dapat diambil dalam penelitian mengenai Implementasi Deep Learning untuk Object Detection Menggunakan Algoritma YOLO (You Only Look Once) pada Rambu Lalu Lintas Di Indonesia adalah sebagai berikut:

1. Kontribusi untuk melengkapi pustaka dataset sangat berguna untuk berbagai hal, seperti penelitian dan pembelajaran.
2. Selain menggunakan *dataset* yang sudah ada untuk pendeteksian objek, *dataset custom* juga dapat dibuat sehingga bermanfaat untuk kebutuhan yang spesifik misalkan untuk pendeteksian rambu lalu lintas di Indonesia.
3. Proses pengumpulan *dataset* adalah kegiatan yang menantang sekaligus memberikan pemahaman karena untuk bisa mengumpulkan *dataset* dengan kuantitas dan kualitas yang dibutuhkan perlu didapat dari berbagai sumber seperti aplikasi pemberi layanan map online (Google Maps dan Mapillary) dan memfoto objek rambu secara langsung di tempat.
4. Cara membuat *dataset* untuk keperluan yang spesifik adalah dengan mengumpulkan data dari objek yang dibutuhkan lalu diberikan label masing-masing datanya sehingga terbentuklah *dataset* yang utuh.
5. Algoritma YOLOv4 dipilih untuk pendeteksian objek rambu lalu lintas di Indonesia karena *performance* pendeteksian cepat dan akurat dengan Average Precision sebesar 43.5% di 65 FPS menggunakan GPU Tesla V100 dimana lebih baik dibandingkan dengan EfficientDet dengan AP 49.5% namun dengan jumlah FPS jauh dibawah YOLOv4.
6. Proses pengumpulan dataset dan pemberian label pada dataset memakan waktu yang lebih lama dibandingkan dengan proses pendeteksian objek
7. *Tuning* algoritma merupakan hal yang harus diperhatikan agar prosedur pendeteksian objek menjadi lebih baik dan bermanfaat.

B. Saran

Berdasarkan penelitian yang telah dilakukan dimulai dari pengumpulan *dataset* sampai dengan hasil pendeteksian, berikut adalah saran yang dapat dilakukan pada penelitian-penelitian selanjutnya:

1. Penambahan jumlah *dataset* dapat dilakukan untuk meningkatkan kemampuan pendeteksian algoritma dan memperluas jangkauan kelasnya.

2. Meningkatkan kualitas *dataset* juga bermanfaat untuk meningkatkan kemampuan pendeteksian algoritma.

REFERENSI

- [1] Alfariasi, H. M. (2020, Mei 26). *Yolov4: Teknologi Terbaru dalam Perkembangan Algoritma Object Detection*. Diterima dari <https://medium.com/@haiqalmuhamadalfarisi/yolov4-teknologi-terbaru-dalam-perkembangan-algoritma-object-detection-78031aad4f16>.
- [2] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. arXiv:2004.1093v1.
- [3] Brownlee, J. (2019, Maret 19). *A Gentle Introduction to Computer Vision*. <https://machinelearningmastery.com/what-is-computer-vision/>. Diakses pada 24 Mei 2021.
- [4] He, Y. (2016). *Object Detection with YOLO on Artwork Dataset*. Spring CS281B Advanced Computer Vision.
- [5] Juanola, M. S. (2019). *Speed Traffic Sign Detection on the CARLA simulator using YOLO*. Spanyol: Universitas Pompeu Fabra.
- [6] Melek, C. G., Sonmez, E. B., Albayrak, S. (2019). *Object Detection in Shelf Images with YOLO*. IEEE EUROCON 2019 – 18th International Conference on Smart Technologies.
- [7] Olafenwa, M. (2019, Agustus 1). *Object Detection Training – Preparing Your Custom Dataset*. Diterima dari <https://medium.com/deepquestai/object-detection-training-preparing-your-custom-dataset-6248679f0d1d>
- [8] Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 13 Tahun 2014. Rambu Lalu Lintas. Menteri Perhubungan Republik Indonesia. Jakarta.
- [9] Peters, J. F. (2017). *Foundations of Computer Vision: Computational Geometry, Visual Image Structures, and Object Shape Detection*. Canada: Springer International Publishing.
- [10] Purnama, B. (2019). *Pengantar Machine Learning: Konsep dan Praktikum dengan Contoh Latihan Berbasis R dan Python*. Bandung: Penerbit Informatika.
- [11] Putra, M. H., Yussof, Z. M., Lim, K. C., Salim, S. I. (2018). *Convolutional Neural Network for Person and Car Detection using YOLO Framework*. Journal of Telecommunication, Electronic, and Computer Engineering, 10, 67-71.
- [12] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2015). *You Only Look Once: Unified, Real-Time Object Detection*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788
- [13] Saini, J. (2020, April 26). *Here's a Quick Way to Learn About Pip in Python*. Diterima dari <https://medium.com/swlh/heres-a-quick-way-to-learn-about-pip-in-python-18617d466c59>
- [14] Sarkar, D. (2018, November 15). *A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning*. Diterima dari <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- [15] Solem, J. E. (2012, Juli 10). *Programming Computer Vision with Python*. Boston: O'Reilly Media.
- [16] Suyanto, Ramadhani, K. N., dan Mandala, S. (2019). *Deep Learning: Modernisasi Machine Learning Untuk Big Data*. Bandung: Penerbit Informatika.
- [17] Szelisky, R. (2010, Oktober 19). *Computer Vision: Algorithms and Applications*. New York City: Springer.
- [18] Undang-Undang Republik Indonesia Nomor 22 Tahun 2009. Lalu Lintas dan Angkutan Jalan. Jakarta.
- [19] Wisna H, J. S., Matulatan, T., dan Hayaty, N. (2020). *Deteksi Kendaraan Secara Real Time Menggunakan Metode YOLO Berbasis Android*. Jurnal Sustainable: Jurnal Hasil Penelitian dan Industri Terapan, 9(1), 8-14
- [20] Zhang, J., Huang, M., Jin, X., Li, X. (2017, November 16). *A Real-Time Chinese Traffic Sign Detection Algorithm Based on Modified YOLOv2*. MDPI Algorithms.
- [21] Zulkhaidi, T. C. A., Maria, E., dan Yulianto. (2019). *Pengenalan Pola Bentuk Wajah dengan OpenCV*. Jurti, 3(2), 182