Implementasi Metode *Yolo Object* Detector untuk Klasifikasi Jenis Kendaraan yang Melintas di Ruas Jalan

Rico Aditya Utama[1] dan Lussiana ETP[2]

[1]Sistem Informasi, Universitas Gunadarma
[1]Jl. Margonda Raya No.100, Depok, Jawa Barat 16424
[2]Magister Teknologi Informasi, STMIK Jakarta STI&K
[2]Jl. BRI No.17, Radio Dalam Kebayoran Baru Jakarta Selatan 12140
E-mail : ricoaditya59@gmail.com, lussiana.etp@gmail.com

Abstrak

Intelligent Traffic System (ITS) adalah upaya untuk mengintegrasikan teknologi telekomunikasi, elektronik, dan informasi dengan rekayasa transportasi untuk merencanakan, merancang, mengoperasikan, memantau dan mengelola sistem transportasi jalan. Salah satu pendukung ITS dengan memanfaatkan CCTV yang berfungsi untuk pengawasan dan pemantauan kondisi jalan. Permasalahan yang timbul antara lain perangkat CCTV yang saat ini ada pada kebanyakan ruas jalan raya hanya digunakan untuk merekam video namun belum difungsikan sebagai alat pendeteksi dan pengenalan jenis kendaraan. Berkaitan dengan hal tersebut penelitian ini bertujuan mengimplementasikan metode YOLO (You Only Look Once) Object Detectors untuk mendeteksi dan mengenali jenis kendaraan pada ruas jalan raya. Tahapan yang dilakukan antara lain akuisisi data citra pada ruas jalan, labelling citra, pembagian data latih dan data uji, augmentasi citra, training model YOLO, dan melakukan pengujian. Berdasarkan hasil pengujian, model mampu mendeteksi 3 jenis kendaraan yaitu mobil, motor, dan bus serta menghasilkan nilai akurasi untuk seluruh jenis kendaraan sebesar 85%, nilai presisi sebesar 93.5%, dan nilai mAP (Mean Average Precision) sebesar 97.07%..

Kata kunci : Deteksi Kendaraan, klasifikasi, YOLO, *Intelligent Traffic System*

Pendahuluan

Saat ini sudah banyak pemanfaatan CCTV di berbagai jalan raya yang digunakan untuk memantau kondisi lalu lintas jalan raya, baik terhadap berbagai pelanggaran pengguna jalan raya maupun kemungkinan kemacetan yang terjadi. Berdasarkan hal tersebut mulai banyak dikembangkan suatu sistem pemantauan canggih, yang dikenal dengan nama Intelligent Traffic System (ITS).

Intelligent Traffic System (ITS) merupakan proses pengintegrasian dari teknologi telekomunikasi, elektronik, dan informasi atau disingkat telematika dengan rekayasa transportasi, untuk merencanakan, merancang, mengoperasikan, memantau, memelihara dan mengelola sistem transportasi jalan [1].

Berkaitan dengan pemasangan dan pemanfaatan CCTV, salah satu fungsinya adalah untuk mengetahui jenis kendaraan yang melintas di jalan raya. Permasalahan yang muncul antara lain bagaimana membangun sistem yang dapat diterapkan dan dapat mengidentifikasi kendaraan secara akurat. Berbagai metode identifikasi atau pendeteksian kendaraan telah banyak dilakukan antara lain: Metode klasifikasi kendaraan menggunakan SVM (*Sup-*

DOI : http://dx.doi.org/10.32409/jikstik.20.4.2875

port Vector Machine) telah dilakukan [2]. Pada penelitian tersebut hal yang pertama kali dilakukan adalah melakukan fitur ekstraksi menggunakan deteksi tepi, deteksi sudut, dan color transform. Kemudian semua fitur tersebut dijadikan input algoritma SVM untuk mengklasifikasikan kendaraan atau bukan. Hasilnya sistem yang diusulkan mampu mendeteksi kendaraan dengan baik tetapi klasifikasinya masih belum sampai pada tipe kendaraan.

Pendeteksian kendaraan juga dikembangkan menggunakan OpenCV oleh [3]. Pada penelitian tersebut dilakukan proses deteksi dan pengenalan tipe kendaraan. Pada proses pendeteksian kendaraan menggunakan metode haar-like feature kemudian dilakukan pengujian sistem usulan dengan 3 skenario kondisi jalan yaitu kondisi padat, sepi, dan normal. Hasil pengujian diperoleh tingkat akurasi pada skenario jalan padat, normal dan sepi secara berurut adalah 28.2%, 47.5%, dan 77.8%.

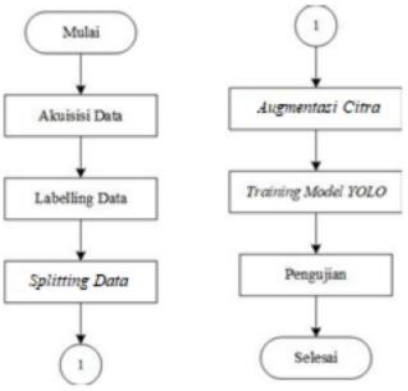
Selanjutnya dikembangkan model pendeteksian kendaraan dengan klasifikasi tipe kendaraan telah dilakukan oleh [4] dan [5]. Pada penelitian tersebut terdapat 3 tahap yang dilakukan yaitu segmentasi citra, klasifikasi kendaraan dan penghitung kendaraan. Tahap segmentasi yang dilakukan diawali dengan rotasi, *background subtraction, image filtering* dan *contour closure.* Selanjutnya pada tahap klasifikasi kendaraan menggunakan metode pengukuran dimensi kendaraan. Hasilnya pada sistem usulan tingkat akurasi deteksinya mencapai 98.7%.

Penelitian deteksi tipe kendaraan dengan melakukan klasifikasi bedasarkan deteksi tepi telah dilakukan oleh [6]. Pada penelitian tersebut tahapan awal yang dilakukan adalah akusisi data ruas jalan raya berupa video yang selanjutnya dilakukan deteksi tepi menggunakan canny. Hasil yang diperoleh didapatkan nilai presentase error deteksi sebesar 2 – 26.75%.

Seiring dengan berkembang pesatnya ilmu deep learning untuk diterapkan di berbagai aspek juga mempengaruhi lahirnya metode baru dalam bidang deteksi objek yaitu dengan object detection algorithm. Salah satu object detection algorithm YOLO (*You Only Look Once*) yang dikembangkan pada tahun 2016 [7]. Tujuan penelitian ini menerapkan algoritma YOLO (*You Only Look Once*) untuk mendeteksi dan mengidentifikasi kendaraan yang melintas di ruas jalan raya.

Metode Penelitian

Tahapan penelitian terlihat pada gambar 1.



Gambar 1: Tahapan Penelitian

Gambar 1 merupakan skema tahapan penelitian yang dilakukan. Secara garis besar terdiri dari identifikasi masalah, akuisisi data, labelling data, splitting data, Augmentasi citra, training YOLO, dan pengujian. Dalam penelitian ini diawali dengan akuisisi data. Dari data citra hasil akuisisi tersebut dilakukan proses labelling pada objek kendaraan yang terlihat. Selanjutnya dilakukan proses augmentasi citra untuk menghasilkan citra yang memiliki sudut pandang baru dari citra yang sudah ada, sehingga dataset memiliki lebih banyak variasi citra tambahan untuk proses training.

Perangkat yang diperlukan

Untuk menunjang kelancaran penelitian, diperlukan pendukung penelitian seperti perangkat keras dan perangkat lunak. Berikut perangkat keras dan perangkat lunak yang digunakan selama penelitian: Processor Intel Core i5 Gen 8, GPU Nvidia Cuda V10.1.243, Sistem Operasi Windows 10, Memori RAM 8 GB, Python 3.6.7, OpenCV 3.4, dan Google Colab.

Akuisisi Data

Tahap akuisisi data dilakukan dengan mengambil citra kendaraan yang melintas di jalan raya, dalam hal ini Jl. Jend. Sudirman Jakarta Selatan. Untuk mendapatkan citra yang dibutuhkan, proses pengambilan dilakukan dari ketinggian sekitar 15 m, sehingga didapatkan citra

berbagai kendaraan berikut ruas jalan yang dilalui. Pada penelitian ini digunakan kamera handphone Xiaomi Mi A1 untuk pengambilan citra, dan untuk kebutuhan pengujan jumlah data citra yang diambil sebanyak 5 citra seperti Gambar 2.

Gambar 2: Data citra Ruas Jalan

Gambar 2 merupakan citra hasil tahap akuisisi data yang disimpan dalam file dengan ekstensi JPG, tampak berbagai kendaraan yang melintas, seperti: kendaraan roda dua, roda empat, kendaraan besar dan badan ruas jalan yang dilalui.

Labelling Data

Tahap ini bertujuan untuk memberi label dan kotak pembatas (posisi objek) pada objek kendaraan yang ingin dideteksi. Dengan pemberian label pada tiap kendaraan maka diperoleh posisi tiap kendaraan sehingga dapat diperoleh posisi lebar dan tinggi kendaraan. Proses labelling ini dilakukan dengan menggunakan software LabelImg.

Gambar 3: Proses Labelling

Gambar 3 merupakan contoh citra yang sedang dalam proses pelabelan (ditunjukkan dengan arah panah). Hal yang pertama dilakukan dalam proses labelling adalah membuat batas berupa kotak di area objek kendaraan dengan cara men-drag kursor pada area objek lalu memberinya label. Hasil dari proses labelling adalah sebuah file annotation yang berisi informasi terkait objek tersebut, seperti pada Gambar 4.

Label	xmin	xmax	ymin	ymax	h	w
Car	1691	2181	1044	1685	641	490

Gambar 4: Hasil labelling

Pada Gambar 4 terdapat 7 atribut objek yaitu label, xmin, xmax, ymin, ymax, h, dan w. Label adalah kelas objek yang didefinisikan dan merupakan atribut yang diidentifikasi. Xmin dan xmax adalah 2 titik pada sumbu x yang digunakan untuk menentukan lebar (w) kotak, sedangkan ymin dan ymax adalah 2 titik pada sumbu y yang digunakan untuk menentukan tinggi (h) kotak. Untuk menentukan lebar kotak dilakukan dengan melakukan pengurangan antara xmax dengan xmin, sedangkan untuk menentukan tinggi kotak dilakukan dengan pengurangan antara ymax dengan ymin. Proses ini diulang sampai semua objek yang terlihat pada citra tersebut mendapat label. Kumpulan data citra setelah proses labelling ini selanjutnya disebut dengan dataset.

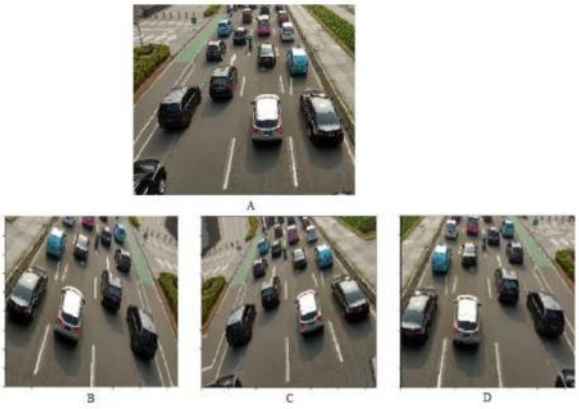
Splitting Data

Splitting data atau pembagian data dilakukan untuk memisahkan antara data yang digunakan untuk proses training dan data yang digunakan untuk pengujian, dengan demikian terdapat data latih dan data uji.

Augmentasi Citra

Tahap ini merupakan proses mengubah atau memodifikasi citra dengan sedemikian rupa sehingga komputer mendeteksi bahwa citra yang sudah diubah tersebut adalah citra yang berbeda [9]. Proses ini bertujuan untuk memperkaya dataset yang dihasilkan dari proses akuisisi data, sehingga dapat meningkatkan tingkat akurasi dan presisi model YOLO yang

dilatih dikarenakan model mendapatkan data-data tambahan yang berguna untuk model melakukan prediksi dengan lebih baik. Proses augmentasi yang dilakukan pada penelitian ini adalah dengan melakukan rotasi acak dengan derajat maksimal 30o, zoom-in secara acak dengan zoom maksimal sebesar 20%, dan membalikkan gambar secara horizontal.



Gambar 5: Contoh Hasil Augmentasi Citra

Gambar 5 menunjukkan hasil proses augmentasi citra dari sebuah sampel dataset citra. Citra A adalah citra hasil akuisisi data, sedangkan citra B, C, dan D adalah hasil dari proses augmentasi. Pada citra B dilakukan operasi mirroring secara horizontal dan rotasi ke arah kiri, dan untuk citra C dilakukan operasi mirroring horizontal lalu rotasi ke arah kanan, sedangkan untuk citra D hanya dilakukan operasi mirroring secara horizontal.

Training Model YOLO

Tahap training model dilakukan dengan langkah-langkah sebagai berikut. [8]

1. Mempersiapkan dataset dalam 1 folder yang berisikan file citra bersama file annotation.
2. Melakukan konfigurasi YOLO training config file untuk mengatur parameter training seperti jumlah epochs, batch size, dan classes.
3. Input data training.
4. Menghitung mAP (Mean Average Precision) untuk mendapatkan pengukuran performa model dari nilai presisi rata-rata per bulk epochs. Berikut rumus dari mAP. [7]

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \qquad (1)$$

5. Menghitung multi-part loss function sebagai average loss pada model dengan target output.
6. Kembali ke langkah dua sebanyak epoch yang telah ditentukan.

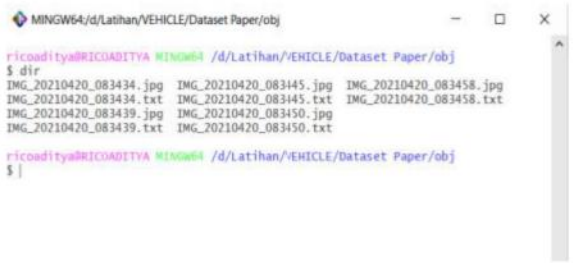
Pengujian

Pengujian ini dilakukan dengan mengambil data uji dan citra baru untuk melihat output prediksi yang dihasilkan oleh model yang sudah dilatih sebelumnya.

Hasil dan Pembahasan

Dataset

Dataset diperoleh dari hasil proses labelling. Berikut ini adalah struktur dataset yang telah siap untuk digunakan.



Gambar 6: Struktur Dataset

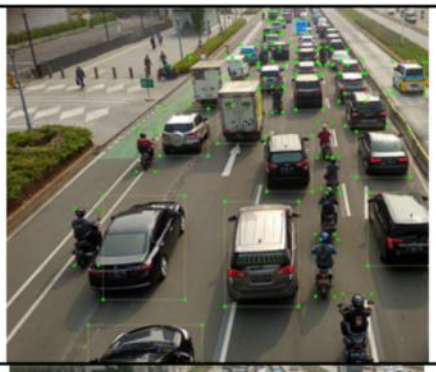
Pada Gambar 6 terlihat struktur folder dari dataset yang telah selesai dari proses labelling. Hasil akhir proses labelling menghasilkan file annotation berformat txt yang berisikan informasi-informasi tentang class object dan dimensi kotak pembatas object. Tabel 1 adalah hasil labelling beserta annotationnya.

Tabel 1: Dataset

Citra Hasil Labelling	Annotation (.txt)

No	Label	xmin	xmax	ymin	ymax	h	w
1	Mobil	1686	2193	1661	2193	532	507
2	Mobil	1119	1524	1173	1524	351	405
3	Mobil	2336	3067	1730	3067	1337	731
4	Mobil	2193	2560	844	2560	1716	367
5	Mobil	1774	2084	763	2084	1321	310
6	Mobil	365	1008	925	1482	557	643
7	Mobil	1060	1334	675	1334	659	274
8	Mobil	1772	2003	370	2003	1633	231
9	Mobil	2088	2355	435	2355	1920	267
10	Mobil	1210	1429	368	1429	1061	219
11	Mobil	1446	1638	494	1638	1144	192
12	Mobil	1558	1746	232	1746	1514	188
13	Motor	1650	1731	575	1731	1156	81
14	Motor	1337	1382	100	1382	1282	45
15	Motor	1393	1430	146	1430	1284	37
16	Mobil	11	462	1790	2240	450	451
17	Mobil	2070	2253	202	2253	2051	183
18	Mobil	1814	1989	104	1989	1885	175
19	Mobil	1649	1786	38	1786	1748	137
20	Mobil	2037	2178	62	2178	2116	141

No	Label	xmin	xmax	ymin	ymax	h	w
1	Mobil	25	896	1740	2247	507	871
2	Mobil	1749	2208	1044	1613	569	459
3	Mobil	2432	3292	1494	2235	741	860
4	Mobil	904	1263	618	1035	417	359
5	Mobil	1787	2082	525	799	274	295
6	Mobil	2251	2680	678	1035	357	429
7	Mobil	1132	1382	380	599	219	250
8	Mobil	1449	1694	349	587	238	245
9	Mobil	1870	2030	273	432	159	160
10	Mobil	2132	2375	301	532	231	243
11	Mobil	1368	1539	182	335	153	171
12	Mobil	1606	1751	109	261	152	145
13	Mobil	2096	2239	140	266	126	143
14	Mobil	1832	1996	97	251	154	164
15	Motor	1373	1425	73	168	95	52
16	Mobil	1461	1589	75	180	105	128
17	Mobil	1656	1763	20	99	79	107
18	Mobil	1865	1982	30	90	60	117
19	Mobil	2030	2134	4	47	43	104
20	Motor	1789	1820	54	161	107	31

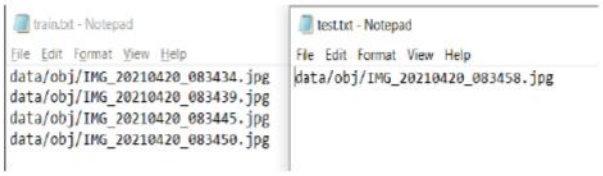


No	Label	xmin	xmax	ymin	ymax	h	w
1	Mobil	21	711	1911	2247	336	690
2	Mobil	1288	1997	1554	2247	693	709
3	Mobil	2121	2992	1518	2247	729	871
4	Mobil	652	1078	1020	1515	495	426
5	Mobil	1411	1743	722	1078	356	331
6	Mobil	1928	2264	930	1225	295	436
7	Mobil	1027	1257	411	652	241	230
8	Mobil	1457	1669	297	580	283	213
9	Mobil	1728	1928	297	592	295	200
10	Mobil	1728	1897	299	428	129	169
11	Mobil	1499	1659	261	282	121	160
12	Mobil	1209	1440	220	354	134	231
13	Mobil	1240	1323	161	256	95	83
14	Mobil	1704	1821	182	285	103	117
15	Mobil	1516	1635	187	256	69	119
16	Mobil	1264	1459	143	230	88	95
17	Mobil	1661	1752	128	206	78	91
18	Mobil	1272	1326	109	170	61	53
19	Mobil	1495	1571	146	225	81	76
20	Mobil	1414	1472	51	128	77	59
21	Mobil	1207	1272	56	116	62	66
22	Mobil	1645	1702	97	159	62	57
23	Mobil	1525	1607	112	166	53	72
24	Mobil	1509	1561	25	82	57	52
25	Mobil	1607	1657	22	85	63	50
26	Mobil	1228	1276	16	54	38	48
27	Motor	1147	1195	216	311	95	48
28	Motor	1182	1228	156	247	91	46

No	Label	xmin	xmax	ymin	ymax	h	w
1	Mobil	494	1278	1490	2237	747	784
2	Mobil	1623	2256	1470	2235	765	633
3	Mobil	2430	3170	1751	2247	496	740
4	Mobil	2270	2911	994	1544	550	641
5	Mobil	1775	2104	761	1128	367	329
6	Mobil	997	1382	851	1237	386	385
7	Motor	892	1037	711	944	233	145
8	Mobil	1261	1466	425	678	253	205
9	Mobil	1799	2049	447	666	219	250
10	Mobil	2166	2430	461	737	276	264
11	Mobil	1380	1542	254	406	152	162
12	Mobil	1787	1937	263	413	150	150
13	Mobil	2073	2249	230	387	157	176
14	Mobil	1630	1766	166	299	133	136
15	Mobil	1516	1642	125	228	103	126
16	Mobil	1882	1999	159	287	128	117
17	Mobil	2018	2151	132	249	117	133
18	Mobil	1854	1956	101	201	100	102
19	Mobil	2025	2128	68	154	86	103
20	Mobil	1573	1680	40	144	104	107
21	Mobil	1713	1801	23	113	90	88
22	Mobil	1842	1925	4	87	83	83
23	Mobil	1970	2051	9	90	81	81

No	Label	xmin	xmax	ymin	ymax	h	w
1	Mobil	25	896	1740	2247	507	871
2	Mobil	1749	2208	1044	1613	569	459
3	Mobil	2432	3292	1494	2235	741	860
4	Mobil	904	1263	618	1035	417	359
5	Mobil	1787	2082	525	799	274	295
6	Mobil	2251	2680	678	1035	357	429
7	Mobil	1132	1382	380	599	219	250
8	Mobil	1449	1694	349	587	238	245
9	Mobil	1870	2030	273	432	159	160
10	Mobil	2132	2375	301	532	231	243
11	Mobil	1368	1539	182	335	153	171
12	Mobil	1606	1751	109	261	152	145
13	Mobil	2096	2239	140	266	126	143
14	Mobil	1832	1996	97	251	154	164
15	Motor	1373	1425	73	168	95	52
16	Mobil	1461	1589	75	180	105	128
17	Mobil	1656	1763	20	99	79	107
18	Mobil	1865	1982	30	90	60	117
19	Mobil	2030	2134	4	47	43	104
20	Motor	1789	1820	54	161	107	31

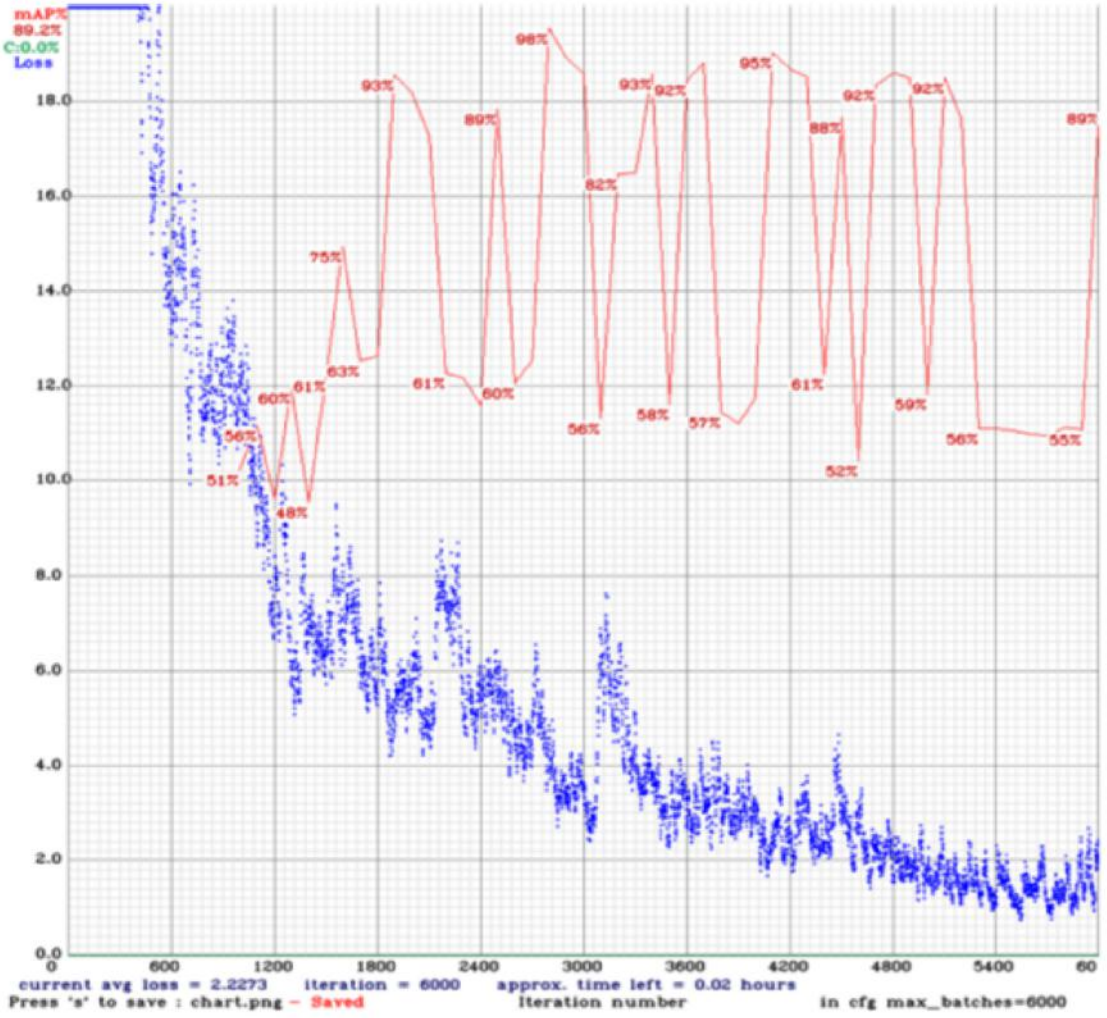
Selanjutnya dari dataset tersebut akan dibagi menjadi data latih dan data uji. Pembagian dataset ini menggunakan presentasi 80:20 yang artinya 80% dari keseluruhan dataset digunakan untuk data latih dan 20% dari keseluruhan dataset digunakan untuk data uji. Berikut ini adalah komposisi antara data latih dan data uji disajikan pada Gambar 7.



Gambar 7: Komposisi data latih dan data uji

Visualisasi Hasil Training

Gambar 8 menunjukkan visualisasi hasil training dengan iterasi sebanyak 6000 kali. Sumbu y sebagai parameter nilai loss sedangkan sumbu x adalah jumlah iterasi training yang dilakukan. Grafik berwarna Biru menunjukkan nilai loss selama proses training. Berdasarkan pada grafik, dapat dinyatakan bahwa semakin banyak iterasi maka nilai loss semakin kecil, sedangkan grafik yang berwarna merah menujukkan nilai presisi rata-rata dan dapat dinyatakan bahwa semakin banyak iterasi maka nilainya semakin besar. Pada iterasi ke-6000 nilai loss dapat ditekan sampai dengan 2.2273 dan nilai presisi mencapai angka 89% ini menunjukkan bahwa semakin lama model dilatih akan semakin baik dalam mengenali objek kendaraan.



Gambar 8: Visualisasi hasil training

Tahap Klasifikasi

Tahap ini dilakukan setelah tahap training untuk data latih. Pada pengujian dilakukan klasifikasi tiga jenis kendaraan, yaitu: mobil, motor, dan bus. Klasifikasi dilakukan dengan mengikuti pengukuran seperti pada Tabel 2.

Tabel 2: Nilai Prediksi

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	TP *(True Positive) Correct Result*	FP *(False Positive) Unexpected Result*
	False	FN *(False Negative) Missing Result*	TN *(True Negative) Correct Absense of Result*

Dimana,

- *True Positive* (TP): Sistem berhasil mengidentifikasi kendaraan dengan benar.
- *False Positive* (FP): Sistem salah mengidentifikasi kendaraan.
- *False Negative* (FN): Objek kendaraan tidak di deteksi sistem.
- *True Negative* (TN): Sistem mendeteksi selain kendaraan.

Tabel 3, Tabel 4 dan Tabel 5 berisi hasil klasifikasi sistem per kelas dari proses pengujian sistem menggunakan data uji.

Tabel 3: Hasil prediksi kelas mobil

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	19	1
	False	0	0

Tabel 3 menunjukkan hasil prediksi model terhadap kelas mobil. Nilai True Positif sebesar 19 kali yang artinya model berhasil mengidentifikasi mobil secara benar dan hanya 1 kali model salah mengidentifikasi objek mobil.

Tabel 4: Hasil prediksi kelas motor

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	9	2
	False	0	0

Tabel 4 menunjukkan hasil prediksi model terhadap kelas motor. *Nilai True Positif* sebesar 9 yang artinya model berhasil mengidentifikasi motor secara benar sebanyak 9 kali dan hanya 2 kali model salah mengidentifikasi objek motor.

Tabel 5: Hasil prediksi kelas bus

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	1	0
	False	0	0

Tabel 5 menunjukkan hasil prediksi model terhadap kelas bus. Nilai *True Positif* sebesar 1 yang artinya model berhasil mengidentifikasi bus secara benar sebanyak 1 kali. Dari data matriks per kelas pada Tabel 3, Tabel 4 dan Tabel 5 didapatkan nilai average precision seperti Gambar 9.

```
class_id = 0, name = car, ap = 96.42%
class_id = 1, name = motor, ap = 94.78%
class_id = 2, name = bus, ap = 100.00%
```

Gambar 9: Average Precision

Selanjutnya dilakukan penghitungan nilai presisi menggunakan persamaan (1) dan diperoleh nilai presisi rata-rata atau *mean average precision* (mAP) terbaik selama proses training adalah sebagai berikut.

$$mAP = \frac{96.42 + 94.78 + 100}{3} = 97.07\%$$

Tabel 6 merupakan tabel hasil klasifikasi dari semua jenis kendaraan dengan menggunakan data uji.

Tabel 6: Hasil prediksi semua kelas

		Nilai Sebenarnya	
		True	***False***
Nilai Prediksi	***True***	29	2
	False	3	0

Dari Tabel 6 terdapat 2 parameter yang berpengaruh pada tingkat performa model yaitu *accuracy* dan *precision*. Nilai accuracy mengukur seberapa tepat model tersebut dapat memprediksi dengan benar sedangkan nilai *precision* mengukur seberapa banyak model memprediksi objek dengan benar. Kedua parameter tersebut dapat dihitung dengan menggunakan persamaan: [10]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + TN} \quad (3)$$

Dengan mengimplementasikan persamaan (2) dan (3) didapatkan nilai *accuracy* dan *precision* dari hasil pengujian model YOLO adalah sebagai berikut.

$$Accuracy = \frac{29 + 0}{20 + 0 + 2 + 3} x100\% = 85\%$$

$$Precision = \frac{29}{29 + 2} x100\% = 93.5\%$$

Dari hasil pengujian yang telah dilakukan, maka didapatkan hasil keseluruhan. Perhitungan dari Tabel 6 menunjukkan hasil akurasi sebesar 85% dan nilai presisi sebesar 93.5% serta nilai mAP sebesar 97.07% untuk hasil persentase seluruh jumlah citra ruas jalan yang termasuk ke dalam data uji menggunakan model YOLO.

Penutup

Bedasarkan hasil pengujian, implementasi deteksi kendaraan pada citra ruas jalan menggunakan model YOLO Object Detection mampu menghasilkan nilai akurasi seluruh kelas sebesar 85%, presisi sebesar 93.5% dan presisi rata-rata sebesar 97.07%.

Daftar Pustaka

[1] Erwin Kusnandar, "ITS Untuk Indonesia", Kementerian Pekerjaan Umum Republik Indonesia, Jakarta, Desember 2011.

[2] C. Antony and E. Konguvel, "Vehicle Detection Based On Feature Extraction and SVM Classification", in Journal of Computer Engineering, pp 42-47, 2014

[3] Alvin Lazaro, Joko Lianto Buliali dan Bilqis Amaliah, "Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV", Jurnal Teknik ITS, 6(2), 2337-3520, 2017.

[4] Z. Maoutakki, I. M. Ouloul, K. Afdel and A. Amghar, "Real-time System Based On Feature Extraction for Vehicle Detection and Classification", Transport and Telecommunication, pp 93-102, 2018.

[5] P.M. Daigavane and P.R. Bajaj, "Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways", International Journal of Computer Science and Network Security", 10(8), 2010.

[6] Bagus Pribadi dan Muchammad Nasser, "Sistem Klasifikasi Jenis Kendaraan Melalui Teknik Olah Citra Digital", SETRUM, vol. 3, no. 2, hal. 35-39, 2014.

[7] Redmon Joseph., Santosh Divvala., Ross Girshick and Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, doi: 10.1109/CVPR.2016.91, 2016.

[8] Adrian Rosebrock, "Deep Learning for Computer Vision with Python", Pyimagesearch, September 2017.

[9] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning", CoRR, abs/1712.04621, http://arxiv.org/abs/1712.04621, 2017.

[10] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Inf. Process. Manag, vol. 45, no. 4, hal. 427–437, 2009.