

Rancang Bangun Sistem Peringatan Identifikasi Alat Pelindung Diri (APD) Menggunakan Metode You Only Look Once v4 (YOLOv4)

Muhammad Noor Rizki Isya¹, Joko Endrasmono², Agus Khumaidi³

^{1,2,3}Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal

Politeknik Perkapalan Negeri Surabaya

Jalan Teknik Kimia, Kampus ITS Sukolilo, Surabaya 60111

E-mail : misya17@student.ppns.ac.id

Abstrak

Berdasarkan data dari BPJAMSOSTEK angka klaim kecelakaan kerja pada semester I 2020 yakni dari Januari sampai dengan Juni 2020 meningkat 128 persen, angka ini naik dari sebelumnya hanya 85.109 kasus menjadi 105.753 kasus. Maka dari itu digunakanlah Alat Pelindung Diri (APD) untuk mencegah potensi bahaya dari kecelakaan kerja. Namun pada praktiknya seringkali dijumpai para pekerja yang melepaskan APD saat sedang bekerja. Hal ini disebabkan akibat kurangnya pengawasan dalam penggunaan APD saat bekerja. Berdasarkan permasalahan tersebut, maka penelitian ini dilakukan dengan pembuatan alat berupa sistem peringatan kelengkapan APD. *Input* dari sistem ini yaitu berupa rekaman citra pekerja yang menggunakan APD dan selanjutnya diproses menggunakan YOLOv4. Algoritma YOLOv4 akan mendeteksi pekerja beserta APD yang digunakan, apabila APD yang dilepaskan saat bekerja maka akan memberikan peringatan kepada pekerja. Sistem tersebut akan berjalan secara terus menerus atau yang bisa disebut dengan *Real-time*. Akurasi dari hasil pengujian data yang digunakan sebesar 94% untuk model YOLOv4, sedangkan untuk model YOLOv4-tiny mencapai 87%. Untuk percobaan sinkronisasi dengan *hardware* mencapai rata-rata 90% dan pengujian dengan tingkat kepatuhan mencapai 88%.

Kata Kunci: Identifikasi, Alat Pelindung Diri, *You Only Look Once v4, Real-time*.

1. PENDAHULUAN

Keselamatan dan kesehatan merupakan prioritas utama bagi para pekerja dan perusahaan. Kesehatan dan Keselemanan Kerja ini atau K3 juga diatur dalam Undang-undang Ketenagakerjaan. Peraturan ini tertuang dalam Peraturan Menteri Tenaga Kerja dan Transmigrasi No. Per.08/Men/VII/2010 tentang Alat Pelindung Diri. Alat Pelindung Diri (APD) adalah kelengkapan yang wajib digunakan saat bekerja sesuai bahaya dan risiko kerja untuk menjaga keselamatan pekerja itu sendiri dan orang di sekelilingnya.

Kecelakaan kerja pada industri banyak terjadi akibat para pekerja yang kurang disiplin dan kurangnya pengawasan dari perusahaan dalam penggunaan Alat Pelindung Diri (APD) ketika bekerja. Berdasarkan data dari BPJAMSOSTEK angka klaim kecelakaan kerja pada semester I 2020 yakni dari Januari sampai dengan Juni 2020 meningkat 128 persen, angka ini naik dari sebelumnya hanya 85.109 kasus menjadi 105.753 kasus. Oleh karena itu, dibutuhkan suatu alat yang dapat mendeteksi dan memastikan kelengkapan alat pelindung diri yang digunakan oleh pekerja. Pada penelitian sebelumnya [7] dan [8], sistem identifikasi kelengkapan alat pelindung diri

dilakukan sebelum melakukan pekerjaan yaitu dengan sistem absensi menggunakan *RFID*. Namun ketika bekerja banyak para pekerja yang merasa risih dan melepaskan alat pelindung diri. Maka dari itu, penelitian kali ini bertujuan untuk membuat alat yang dapat memberi peringatan ketika Alat Pelindung Diri (APD) dilepaskan saat bekerja.

Sistem ini akan menggunakan metode *You Only Look Once v4* (YOLOv4). YOLOv4 merupakan sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time*. YOLOv4 akan mendeteksi pekerja dan APD yang digunakan, apabila APD tersebut dilepas atau menjauh dari deteksi pekerja maka peringatan akan aktif.

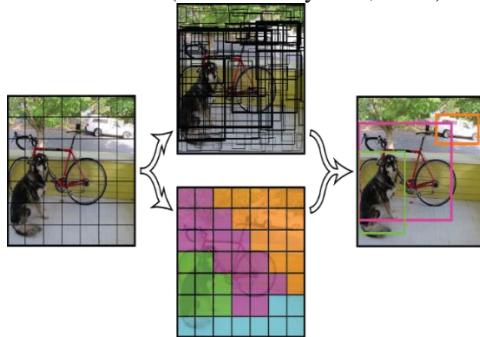
2. METODOLOGI

2.1 You Only Look Once v4 (YOLOv4)

You Only Look Once (YOLO) adalah sebuah algoritma yang dikembangkan untuk melakukan deteksi secara *real-time*. YOLO menggunakan pendekatan yang sangat berbeda dengan algoritma sebelumnya, yakni menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi kotak pembatas dan

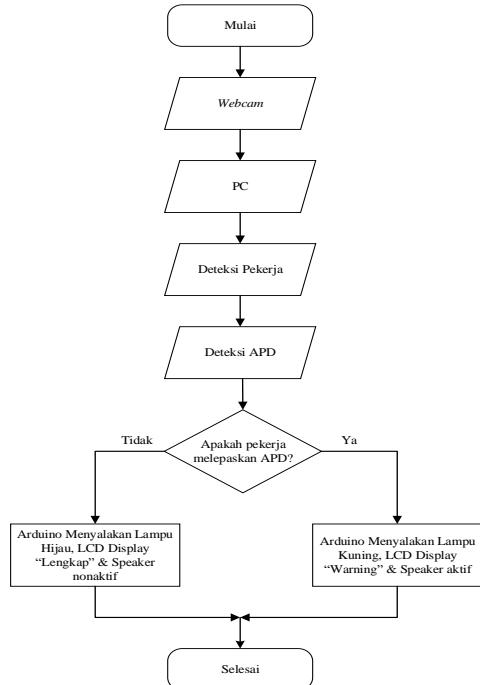
probabilitas, untuk setiap kotak wilayah pembatas ditimbang probabilitasnya untuk mengklasifikasikan sebagai objek atau bukan (Yanuar, 2018)

Dalam Tugas Akhir ini, versi YOLO yang digunakan adalah YOLOv4 dikarenakan deteksi yang lebih cepat dan akurat dibandingkan dengan versi YOLO sebelumnya dan *object detector* lainnya, YOLOv4 dikatakan juga bekerja sangat cepat baik di CPU maupun GPU sehingga cocok digunakan untuk *real-time* (Bochkovskiy dkk., 2020).



Gambar 2.1 Deteksi Objek YOLO

2.2 Perancangan Sistem

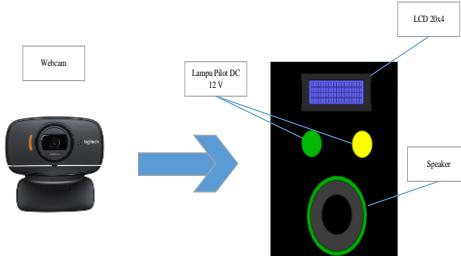


Gambar 2.2 Flowchart Sistem

Pada **Gambar 2.2** *webcam* bertindak sebagai alat yang merekam ruangan sekitar layaknya sebagai sensor. Hasil dari rekaman tersebut selanjutnya di proses pada *PC* dengan metode YOLOv4 yang sudah di *training*. YOLOv4 akan mendeteksi pekerja serta APD

yang digunakan. Apabila APD yang digunakan tidak dilepas maka *Arduino* akan menyalaikan lampu hijau, LCD menampilkan “Lengkap”, dan *speaker* nonaktif. Apabila APD dilepas maka *Arduino* akan menyalaikan lampu kuning, LCD menampilkan “Warning”, dan mengaktifkan *speaker*.

2.3 Desain Hardware



Gambar 2.3 Desain Hardware



Gambar 2.4 Desain Hardware

Pada **Gambar 2.4** merupakan salah satu contoh penerapan *hardware* pada tempat kerja dimana pekerja merupakan objek yang akan di deteksi beserta artibut APD yang dikenakan, apabila pekerja tersebut melepas APD maka akan mengaktifkan sistem peringatan, apabila pekerja tidak melepas APD maka sistem peringatan tidak aktif.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini terdapat pengujian dataset hasil *training* menggunakan model YOLOv4 dan YOLOv4-tiny dengan 10 kali percobaan pada setiap *class*, percobaan sinkronisasi dengan *hardware* dengan cara melepas APD, dan pengujian sistem dengan tingkat kepatuhan.

3.1 Dataset

Proses pembuatan dataset dengan mengambil gambar sampel objek lalu membaginya menjadi 13 *class* objek dengan cara anotasi data. Anotasi data dilakukan untuk mendapat nilai (*class id*, *x*, *y*, *width*, *height*).



Gambar 3.1 Pembuatan Dataset

3.2 Training

Proses ini bertujuan untuk mendapatkan bobot yang optimal dengan cara mengolah gambar dan anotasi yang telah dibuat sehingga terbentuk pola atau karakteristik dari masing-masing *class* yang akan menjadi bahan pertimbangan komputer dalam mencapai sebuah keputusan dalam memprediksi objek. Proses *training* dilakukan menggunakan CPU atau GPU

```
(Next mAP calculation at 1000 iterations)
2: 2283.846436, 2287.456684 avg loss, 0.000000 rate, 402.387715 seconds, 128 images, 2838.146112 hours left
Loaded: 0.000019 seconds
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.349513), count: 27, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.385788), count: 26, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.356411), count: 8, class
total bbox = 4205, rewritten bbox = 0.142687
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.381142), count: 28, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.336065), count: 23, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.305334), count: 6, class)
```

Gambar 3.1 Training menggunakan CPU

```
(Next mAP calculation at 1000 iterations)
2: 2286.967646, 2287.577881 avg loss, 0.000000 rate, 4.239488 seconds, 128 images, 75.742751 hours left
Loaded: 0.000027 seconds
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.373506), count: 28, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.334603), count: 27, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.265520), count: 10, class
total bbox = 4566, rewritten bbox = 0.131464 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.314281), count: 45, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.379703), count: 46, class
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.355612), count: 17, class)
```

Gambar 3.2 Training menggunakan GPU

Berdasarkan gambar tersebut dapat dilihat bahwa *training* menggunakan CPU memerlukan waktu 2838 jam sedangkan pada GPU memerlukan waktu 75 jam, sehingga pada penelitian ini digunakan GPU sebagai tempat komputasi *training*.

3.3 Pengujian Dataset



Gambar 3.3 Pengujian Dataset

Tabel 3.1 Hasil Pengujian Data Setiap *Class*

Class	YOLOv4		YOLOv4-tiny	
	Akurasi	FPS	Akurasi	FPS
Lengkap	64,2%	6,83	49%	38,69
Tidak lengkap	100%	6,44	100%	38,64
Safety helmet	100%	6,89	95%	37,62
Safety earmuff	95%	6,88	83%	39,1
Safety glasses	98%	6,90	68%	38,06
Safety industry mask	97%	6,87	92%	38,52
Safety mask	98%	6,87	87%	38,22
Safety gloves	100%	6,86	100%	38,95
Without Safety Gloves	99%	6,59	100%	38,52
Without Safety Mask	88%	6,86	96%	34,89
Without Safety Earmuff	96%	6,87	98%	38,95
Without Safety Glasses	93%	6,89	98%	38,22
Without Safety Helm	95%	6,88	92%	38,35
Rata-rata	94%	6,79	87%	38,12

Berdasarkan **Tabel 3.1** diketahui bahwa akurasi YOLOv4 lebih tinggi sedangkan pada FPS YOLOv4-tiny lebih cepat. Sehingga pada sistem akan menggunakan YOLOv4-tiny sebagai model objek deteksi dengan alasan mencari FPS yang lebih cepat meskipun mengorbankan sedikit akurasi.

3.4 Percobaan Sinkronisasi dengan Hardware



Gambar 3.4 Percobaan Sinkronisasi dengan *Hardware*

Tabel 3.2 Hasil Percobaan Sinkronisasi dengan *Hardware*

Percobaan	Aktif	Tidak Aktif	Persentase
Pelepasan APD Helm	9	1	90%
Pelepasan APD Penutup Telinga	8	2	80%
Pelepasan APD Kacamata	10	0	100%
Pelepasan APD Masker	9	1	90%
Pelepasan APD Sarung Tangan	9	1	90%
Rata-rata		90%	

Berdasarkan 10 kali percobaan pelepasan pada masing-masing APD, terdapat 5 kali sistem peringatan tidak aktif dengan rata-rata keberhasilan 90% ketika APD dilepas. Sistem peringatan tidak aktif disebabkan karena tidak terdeteksinya objek bagian anggota tubuh yang dilepas APD.

3.5 Pengujian Sistem dengan Tingkat Kepatuhan

Pengujian sistem dengan tingkat kepatuhan dilakukan melibatkan satu kelompok eksperimen dengan jumlah 5 orang dengan cara memanipulasi suatu keadaan sesuai prosedur yang diberikan berupa pelepasan APD pada setiap percobaan dengan tujuan untuk mengetahui apakah sistem peringatan aktif atau tidak.



Gambar 3.5 Pengujian Sistem dengan Tingkat Kepatuhan

Tabel 3.3 Hasil Pengujian Sistem dengan Tingkat Kepatuhan

Jumlah Orang	Persentase Kepatuhan	Status Peringatan	Keterangan
1	100%	Tidak Aktif	Benar
1	80%	Tidak Aktif	Salah
1	60%	Aktif	Benar
2	100%	Tidak Aktif	Benar
2	80%	Tidak Aktif	Salah
2	60%	Aktif	Benar
3	100%	Tidak Aktif	Benar
3	80%	Aktif	Benar
3	67%	Aktif	Benar
4	100%	Tidak Aktif	Benar
4	80%	Aktif	Benar
4	80%	Aktif	Benar
5	100%	Tidak Aktif	Benar
5	80%	Aktif	Benar
5	80%	Aktif	Benar
Rata-rata Kebenaran			87%

Berdasarkan **Tabel 3.3** pengujian dengan tingkat kepatuhan pada masing-masing APD, terdapat 2 kali sistem peringatan tidak aktif yang seharusnya aktif ketika tingkat kepatuhan dibawah 100%.

4. PENUTUP

4.1 Kesimpulan

Dari hasil percobaan dan pengujian yang telah dilakukan pada tugas akhir ini dapat disimpulkan sebagai berikut. Pertama, pada

proses *training* data dapat disimpulkan bahwa menggunakan GPU untuk tempat komputasi jauh lebih cepat dibandingkan menggunakan CPU sebesar 97%. Hal ini disebabkan karena GPU memang ditugaskan untuk memproses pengolahan tampilan grafik dan memiliki VRAM, dimana CPU tidak memilikinya sehingga memerlukan waktu yang lama untuk proses *training*. Kedua, pada pengujian data akurasi yang dihasilkan oleh model YOLOv4 adalah 94% dengan nilai FPS 6,79 sedangkan model YOLOv4-tiny memiliki akurasi 87% dengan nilai FPS 38,12. Hal ini menunjukkan bahwa model YOLOv4-tiny bekerja lebih cepat dibandingkan model YOLOv4 meskipun mengorbankan sedikit akurasi, maka dari itu digunakanlah model YOLOv4-tiny pada sistem dengan tujuan performa yang lebih cepat.

Untuk penelitian selanjutnya terdapat beberapa saran yaitu, memperbanyak dataset setiap *class* untuk meningkatkan akurasi deteksi yang lebih baik dan menggunakan PC/laptop dengan spesifikasi yang lebih baik agar proses *training* atau menjalankan proses deteksi bisa lebih cepat dan maksimal terutama apabila ingin menggunakan model YOLOv4.

5. DAFTAR PUSTAKA

- [1] Anwar, A., Pengertian dan Sejarah Bahasa Pemrograman Python, Indonesia: ilmuti.org, 2019.
- [2] Bochkovskiy, A., Wang, C. Y. and Liao, H. Y. M., “YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv, 2020.
- [3] Digital 16bit Serial Output Type Ambient Light Sensor IC, BH1750FVC, ROHM, Kyoto, Japan: 2011.
- [4] Eko Supriyatno, S., “Pemodelan Sistem Audio Secara Wireles Transmitter Menggunakan Laser Pointer”, Jurnal Teknik Mesin (JTM), 05, 2015.
- [5] Huang Gao, Liu Zhuang, van der Maaten Laurens, Weinberger Kilian, “Densely Connected Convolutional Networks”. 10.1109/CVPR.2017.243, 2017.
- [6] Maier, J., Humenberger, M. (2013), “Movement detection based on dense optical flow for unmanned aerial vehicles”, *International Journal of Advanced Robotic Systems*, 10, pp. 1–11. doi: 10.5772/52764.
- [7] Menteri Tenaga Kerja dan Transmigrasi, “Peraturan Menteri Tenaga Kerja dan Transmigrasi Republik Indonesia”, Peraturan Menteri tenaga Kerja dan Transmigrasi, VII(8), 2010.
- [8] Pham, M. T., Courtai, Luc, Friguet, Chloe, Levefre, Sebastian, Baussard, Alexandre. (2020) ‘YOLO-fine: One-stage detector of small objects under various backgrounds in remote sensing images’, *Remote Sensing*, 12(15), pp. 1–26. doi: 10.3390/RS12152501.
- [9] Pradana, R. D. W., “Rancang Bangun Sistem Identifikasi Kelengkapan Alat Pelindung Diri (APD) Menggunakan Metode Convolutional Neural Network (CNN)”, Surabaya: Politeknik Perkapalan Negeri Surabaya, 2019.
- [10] Pratama, T. A. K., “Pengembangan Sistem Identifikasi Alat Pelindung Diri (APD) Menggunakan Metode Convolutional Neural Network (CNN)”, Surabaya: Politeknik Perkapalan Negeri Surabaya, 2020.
- [11] Redmon, Joseph, Divvala, Santosh, Girshick, Ross, Farhadi, Ali, ‘You only look once: Unified, real-time object detection’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 779–788. doi: 10.1109/CVPR.2016.91, 2016.
- [12] Toleubay Yeldar, James Alex, “Getting Started with TensorFlow Deep Learning”, 10.1007/978-3-030-14524-8_4, 2020.