

## QUEUE DISTANCING BASED ON IMAGE PROCESSING

Riski Anandita Basyuni<sup>1)</sup> dan Budi Sugandi<sup>2)</sup>

<sup>1)</sup>Jurusan Teknik Elektro, Politeknik Negeri Batam

<sup>2)</sup>Jurusan Teknik Elektro, Politeknik Negeri Batam

E-mail: budi\_sugandi@polibatam.ac.id

### Abstract

This article aims to propose a system to detect queue distancing based on You Only Look Once (YOLO) algorithm as real-time image processing. YOLO perform single neural network to whole image. The networks divide an image into some pieces and predict a bounding box and probability to all of image parts. The method could detect and classify the object simultaneously. Our proposed system is evaluated in real time lift queue within queue distance 1 meter. The system alarm is given to the queue distance below than 1 meter. Sistem diuji secara real time di suatu antrian lift dengan jarak antar objek 1 m. The system performed succesfully queue distancing detection accuracy 100% in 10 times experiment.

**Keywords:** *Queue distancing, deteksi objek, yolo, real time*

## PENDAHULUAN

Antrean adalah kumpulan orang yang berdiri deretan memanjang yang menunggu untuk mendapatkan giliran dilayani dan lain sebagainya. Antrean akan diproses sesuai urutan yang pertama kali akan dilayani terlebih dahulu. Banyaknya kumpulan orang dalam sebuah antrean dapat mengakibatkan antrean yang tidak teratur. Sehingga beberapa pelayanan public menerapkan batas maksimal antrean. Pada masa pandemi covid, untuk menjaga penyebaran penyakit telah diterapkan pula *physical distancing* atau pembatasan jarak fisik. Menurut spesialis penyakit menular Stevan Gordo, MD dari Claveland menjelaskan bahwa *pyhsical distancing* yang dianjurkan ialah minimal 1 meter (Kompas.com, 2021). Penerapan batasan antrean ini membuat sistem pada beberapa pelayanan public menerapkan sistem untuk mengawasi jarak antrean. Beberapa penelitian terkait pengawasan jaarak antrian pun berkembang dan banyak diusulkan untuk diterapkan di masa pandemi ini. Penelitan ini terkait dengan pendeteksian keberadaan objek dan jarak antar objek dalam antrian.

Penelitian terkait pendeteksian objek telah dilakukan oleh beberapa peneliti. Salah satunya dengan menggunakan metode *Histogram of Oriented Gradient*. Metode ini menggunakan orientasi gradien untuk mendeteksi objek (Sariningsih et al., 2017).

Dalam penelitian lainnya, proses pendeteksian objek menggunakan pengolahan warna model normalisasi RGB. Metode ini menghitung presentase warna RGB pada sebuah citra (Kusumanto & Tomponu, 2011). Metode *background subtraction* telah diusulkan pula untuk mendeteksi objek bergerak (Umam & Negara, 2016). Penerapan pengolahan citra juga telah dilakukan untuk mengetahui pergerakan menggunakan kamera dengan metode *adaptive motion detection algorithm* (Antonius et al., 2015). Penggunaan algoritma *Haar Cascade Classifier* juga telah digunakan beberapa penelitian (Setjo et al., 2017) dan (Pratama et al., 2020).

Metode *deep learning* merupakan metode pendeteksian yang lebih akurat untuk objek yang kompleks dan telah banyak diusulkan oleh beberapa peneliti. Meskipun masih tetap menyisakan kekurangan dan tantangan untuk lebih diperbaiki dari sisi akurasi dan kecepatan pengolahan data (Syarifah, 2018) dan (Mulyawan et al., 2013)..

Metode *You Only Look Once* (Yolo) merupakan metode yang banyak dipakai dalam proses pendeteksian objek. Yolo menggunakan seluruh gambar dan membaginya untuk kemudian dilakukan pendeteksian (Jupiyandi et al., 2019) dan (Redmon et al., 2016).

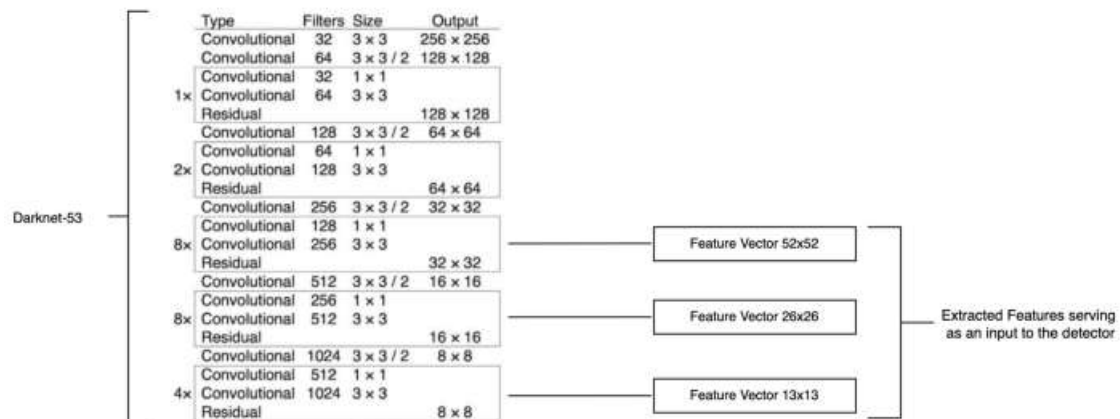
Artikel ini bertujuan untuk melakukan pendeteksian jarak antrian dengan menerapkan algoritma Yolo. Deteksi jarak antrian dilakukan dengan mendeteksi objek dan mengukur jarak antar objek. Jarak antar objek ditentukan dengan mengukur perbandingan piksel yang terdeteksi dengan jarak objek dengan kamera.

Sistematika artikel ini disajikan mulai dari pendahuluan, algoritma yang digunakan dalam artikel, kemudian diikuti oleh hasil dan pembahasan terkait implementasi dari algoritma yang digunakan. Artikel diakhiri oleh simpulan yang didapat dari implementasi algoritma dan hasil yang didapatkan.

## **METODE PENELITIAN**

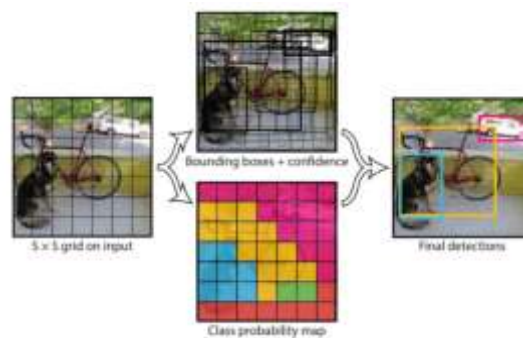
Artikel ini menggunakan Yolo-v3 dalam proses pendeteksian objek. Yolo-v3 adalah algoritma yang digunakan untuk mendeteksi objek yang menggunakan darknet-53 sebagai *feature framework*. Pada awalnya YOLO-v3 mempunyai 53 layer CNN kemudian saat pendeteksian 53 layer lagi ditumpuk di atasnya sehingga menjadi 106 layer. YOLO-v3 menggunakan *multi-label classification* untuk memprediksi kelas pada tiap *bounding box*. Hal ini karena setiap *bounding box* mungkin mempunyai lebih dari

satu kelas ataupun tidak mempunyai kelas sama sekali. Untuk input citra berdimensi 416x416 digunakan 3 skala pada *multi-label classification* yaitu 52x52 untuk objek yang kecil, 26x26 untuk objek menengah dan 13x13 untuk objek yang besar seperti yang ditunjukkan pada gambar 1.

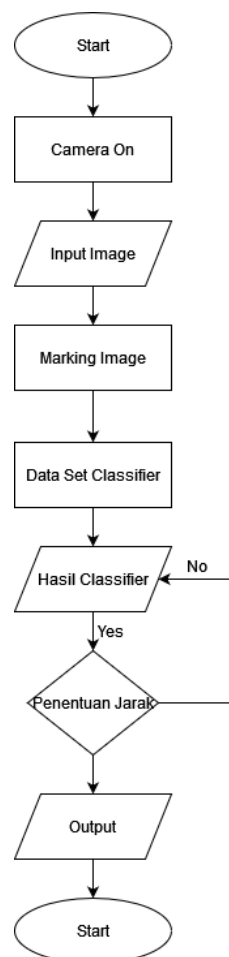


Gambar 1. *Multi-Scale Feature Extraction* untuk Gambar 416x416

Proses awal pendeteksian menggunakan Yolo adalah dengan membagi menjadi grid  $S \times S$ . Pada setiap sel grid yang berfungsi sebagai memprediksi objek didalamnya. Setiap sel grid bertugas memprediksi kotak pembatas  $B$  dan nilai probabilitas kelas  $C$  pada grid. Prediksi pada kotak  $B$  memiliki 5 koordinat penting dalam proses pendeteksian yaitu  $x$ ,  $y$ ,  $w$ ,  $h$ , dan *confidence score*. Koordinat  $x$  dan  $y$  merupakan pusat dari *bounding box* yang mengarah terhadap batas kotak grid, koordinat ini dinormalisasi antara 0 dan 1. Koordinat  $w$  dan  $h$  adalah ukuran gambar. Sementara *confidence score* menggambarkan seberapa yakin kotak pembatas  $B$  berisi objek (Redmon et al., 2016). Setiap *bounding box* memprediksi probabilitas nilai  $C$  pada gambar sebagai bahan untuk pendeteksian. Jaringan ini akan membagi gambar menjadi wilayah-wilayah kemudian memprediksi kotak pembatas dan probabilitas untuk setiap kotak wilayah pembatas probabilitasnya untuk mengklasifikasi sebagai objek. Proses pendeteksian dengan Yolo digambarkan pada gambar 2.



Gambar 2. Ilustrasi proses pendeteksian YOLO



Gambar 3. Diagram alir sistem

Gambar 3 menunjukkan diagram alir sistem yang dikembangkan untuk sistem training dan testing pendeteksian objek antrian. Kondisi awal dari sistem adalah pengambilan data citra untuk data set yang objeknya adalah pergerakan manusia. Kemudian citra tersebut diberi dilabel berupa *bounding box* untuk tiap objek yang

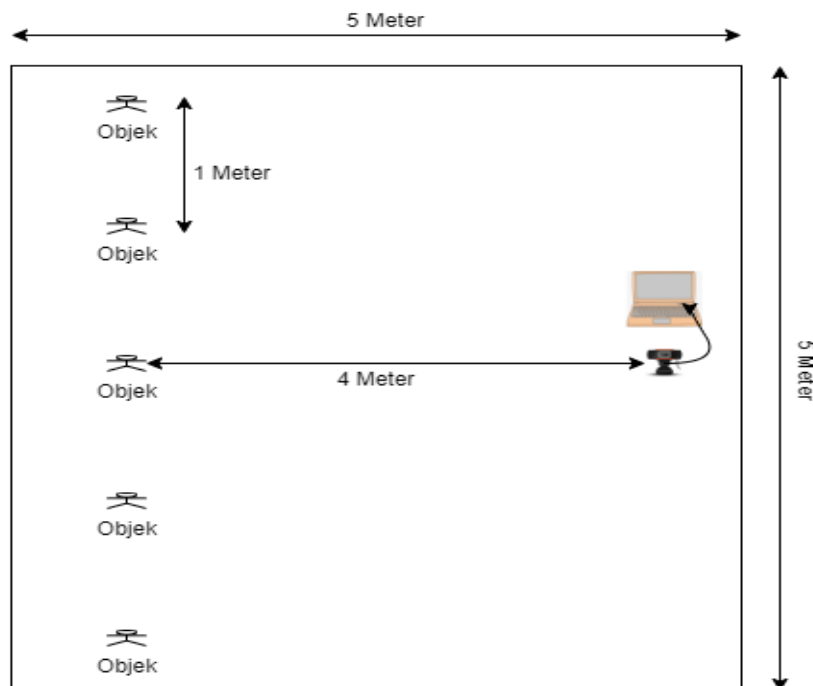
dideteksi. Proses pelabelan menggunakan *Yolo-Mark* yang merupakan *framework* dari *neural network darknet*. Setelah semua dataset gambar dilabel, dataset akan di teruskan ke proses *training*. Proses *training* bertujuan untuk melatih komputer dengan cara mengolah gambar dan pelabelan objek yang telah dibuat sehingga membentuk pola dari karakteristik kelas. Karakteristik tersebut akan menjadi bahan pertimbangan komputer dalam mencapai prediksi atau keputusan. Setelah *training* selesai, data hasil *training* akan di lakukan pengujian atau testing. Proses testing dimulai dengan input berupa video secara *real time*, saat sistem mendeteksi antrean yang sesuai dengan yang telah dilabel maka sistem akan menampilkan informasi berupa *bounding box*. Sistem akan memberikan alarm jika jarak antara 1 orang ke orang lainnya kurang dari 1 meter. Dengan mengatur jarak antara kamera dengan objek diperoleh bahwasannya jarak 1 meter ekuivalen dengan 120 piksel pada data citra. Dengan menggunakan perbandingan referensi jarak tersebut akan diketahui jarak antar objek.

## HASIL DAN PEMBAHASAN

Pada sistem ini ruangan yang dipergunakan seluas 5 x 5 meter. Objek yang akan dideteksi akan berjumlah 5 dan objek akan berbaris antrean berjarak 1 meter sesuai garis referensi yang telah dibuat sebagai acuan garis antrean. Kamera diletakan di tengah pada jarak 2.5 meter. Jarak kamera dengan objek terjauh adalah 4 meter. Pengambilan dan pengujian data dilakukan pada arah samping pada antrean, ini dilakukan karena agar objek terlihat dengan jelas pada 1 baris antrean. Gambaran detail setting pengujian ditunjukkan pada gambar 4. Spesifikasi laptop yang digunakan ditunjukkan pada tabel 1. Hasil pengujian sistem ditunjukkan pada gambar 5.

Tabel 1  
Spesifikasi laptop yang digunakan

<i>Processor</i>	11th Gen Intel® Core™ i5-11300H @ 3.10GHz
<i>RAM</i>	8 GB
<i>Storage</i>	SSD 512 GB
<i>Graphics Card</i>	NVIDIA® GeForce® GTX 1650 with 4GB of dedicated GDDR6
<i>Operating System</i>	Ubuntu 18.04 LTS 64-bit

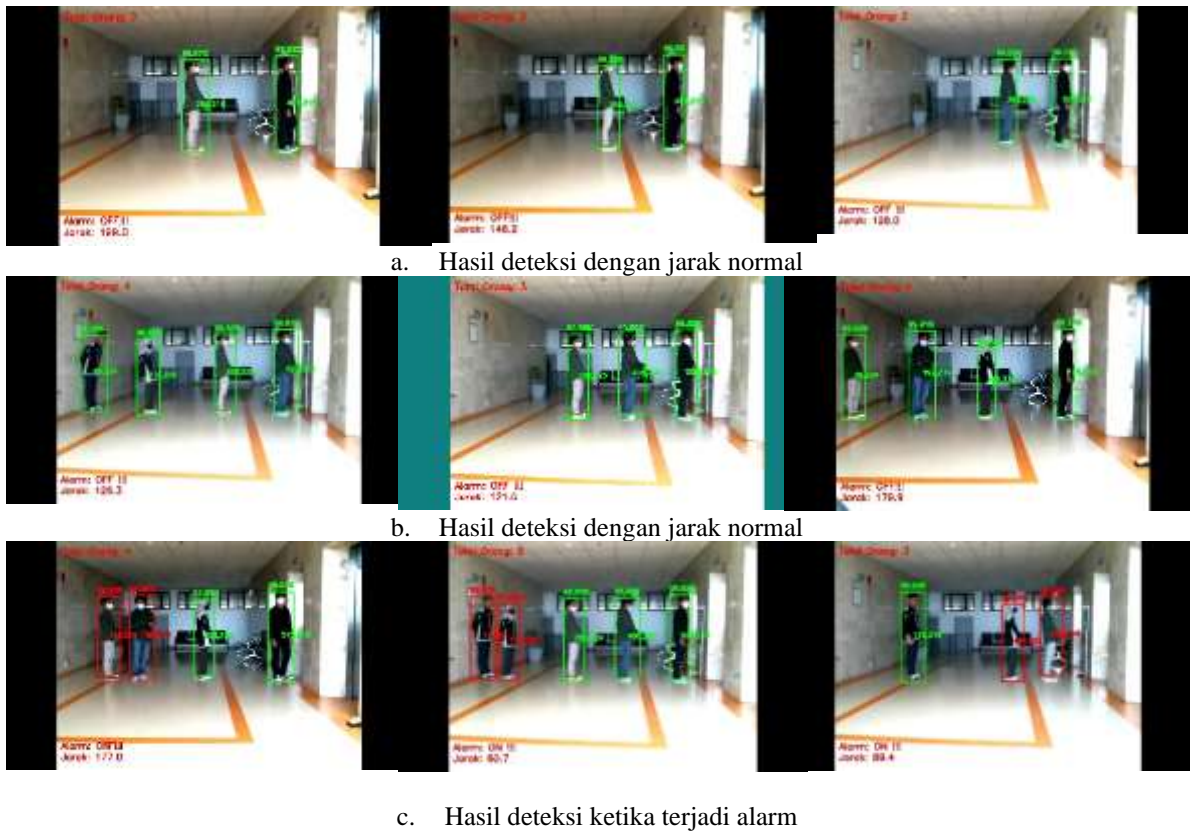


Gambar 4. Setting pengujian

Tabel 2  
Tabel Pengujian

Frame	Jarak aktual (cm)	Jarak pengujian (pixel)	Alarm (ON/OFF)
Frame 1	170	199	OFF
Frame 2	120	146	OFF
Frame 3	150	176	OFF
Frame 4	100	121	OFF
Frame 5	50	177	ON
Frame 6	150	186	OFF
Frame 7	130	153	OFF
Frame 8	70	85	ON
Frame 9	100	128	OFF
Frame 10	30	60	ON
Frame 11	100	126	OFF
Frame 12	70	89	OFF

Tabel 2 menunjukkan hasil pendeteksian jarak antrean dibandingkan jarak sebenarnya dan proses pendeteksian jarak yang menyebabkan alarm berbunyi. Dari hasil diatas terlihat bahwa sistem berhasil memberikan alarm peringatan batasan jarak antrean sebesar 1 m, di mana jika pendeteksian jarak antrean kurang dari 1 m maka alarm akan berbunyi.



Gambar 5. Hasil Deteksi

## SIMPULAN

Dari hasil pengujian sistem yang telah dilakukan pada antean sejumlah orang, sistem telah berhasil mendeteksi objek dan jarak antar masing-masing antean. Sistem berhasil mengukur jarak antar objek dan memberikan alarm jika jarak lebih kecil dari jarak antean yang diperbolehkan. Referensi jarak yang digunakan merupakan konversi dari jumlah pixel yang dideteksi dengan menggunakan jumlah pixel 120. Jarak ini dihitung pada kondisi jarak kamera yang tetap. Hasil pengujian menunjukkan semua kondisi objek telah dideteksi dengan baik dan menghasilkan alarm yang sesuai dengan jarak yang ditentukan.

Untuk pengerjaan ke depan, perlu ditambahkan dengan jumlah antean lebih dari 1. Dan juga mencoba mengubah setting kamera tidak hanya diletakkan di samping, juga diletakkan di atas objek. Ini menjadi topik lanjutan di prodi selanjutnya.

**DAFTAR PUSTAKA**

- Antonius, A., Triyanto, D., & Ruslianto, I. (2015). Penerapan Pengolahan Citra dengan Metode Adaptive Motion Detection Algorithm pada Sistem Kamera Keamanan dengan Push Notification ke Smartphone Android. *Jurnal Coding Sistem Komputer Untan*, 3(2), 54–65.
- Jupiyandi, S., Saniputra, F. R., Pratama, Y., Dharmawan, M. R., & Cholissodin, I. (2019). Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(4), 413. <https://doi.org/10.25126/jtiik.2019641275>
- Kompas.com. (2021). *Berapa Jarak Aman untuk Cegah Penularan Covid-19?* <https://lifestyle.kompas.com/read/2020/06/05/094239020/berapa-jarak-aman-untuk-cegah-penularan-covid-19>
- Kusumanto, R., & Tomponu, N. (2011). Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi Rgb. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan 2011 (Semantik 2011)*, 17(C), 329–332.
- Mulyawan, H., Samsono, M. Z. H., & Setiawardhana. (2013). Identifikasi Dan Tracking Objek Berbasis Image Processing Secara Real Time. *Politeknik Elektronika Negeri Surabaya, Institut Teknologi Sepuluh Nopember, (ITS) Surabaya.*, 1–5. [http://repo.pens.ac.id/1324/1/Paper\\_TA\\_MBAH.pdf](http://repo.pens.ac.id/1324/1/Paper_TA_MBAH.pdf)
- Pratama, M. R., Rizal, & Sumaryo, S. (2020). Desain Sistem Deteksi Objek Real Time Dengan Metode Haar Cascade Classifier. *E-Proceeding of Engineering*, 26–34.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Sariningsih, N. O., Suratman, F. Y., & Wibawa, I. P. D. (2017). Perancangan Dan Implementasi Tracking Objek Manusia Untuk Autonomous Car. *EProceedings of Engineering*, 4(1).
- Setjo, C. H., Achmad, B., & Faridah. (2017). Thermal image human detection using Haar-cascade classifier. *Proceedings - 2017 7th International Annual Engineering Seminar, InAES 2017*. <https://doi.org/10.1109/INAES.2017.8068554>
- Syarifah, R. D. (2018). Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network. *Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network*.
- Umam, K., & Negara, B. S. (2016). Deteksi Obyek Manusia Pada Basis Data Video Menggunakan Metode Background Subtraction Dan Operasi Morfologi. *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer Dan Teknologi Informasi*, 2(2), 31. <https://doi.org/10.24014/coreit.v2i2.2391>