



## Deteksi dan Penghitung Keramaian Menggunakan You Only Look Once 3 Tiny dan Raspberry Pi

Alauddin Maulana Hirzan<sup>1</sup>, Rastri Prathivi<sup>2</sup>, Mohammad Burhan Hanif<sup>3</sup>

<sup>1,2,3</sup> Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang, Semarang Indonesia

Email: <sup>1</sup>maulanahirzan@usm.ac.id, <sup>2</sup>vivi@usm.ac.id, <sup>3</sup>hanifburhan@usm.ac.id

### Abstrak

Keramaian adalah aspek sosial yang tidak bisa dipisahkan dari masyarakat. Baik untuk keperluan bersosialisasi hingga menyampaikan suara melalui demonstrasi, masyarakat akan membentuk keramaian untuk mencapai tujuan tersebut. Keramaian ini tentu saja memiliki dampak positif, namun tetap memiliki dampak negatif berupa kemungkinan terjadinya provokasi dan membentuk anarkisme. Oleh karena itu banyak penelitian yang memiliki fokus untuk melakukan deteksi keramaian. Namun sayangnya, penelitian yang telah dilakukan ini memiliki kelemahan di mana model yang dibuat tidak mampu melakukan deteksi jarak antar satu manusia dengan manusia lainnya. Untuk mengatasi permasalahan tersebut, penelitian ini mendesain sebuah model deteksi menggunakan YOLOv3-Tiny yang diimplementasikan ke dalam perangkat Internet of Things. Dari proses pengujian menggunakan 60 gambar dengan resolusi yang berbeda. Pengujian ini berhasil mendeteksi keramaian dan jarak antar manusianya. Model ini membutuhkan CPU sebanyak 76,22%. Untuk memori membutuhkan 454,78MB untuk proses, 405,61MB untuk data, dan 130,94MB untuk memori virtual. Dari hasil ini bisa disimpulkan bahwa model yang diusulkan mampu mendeteksi keramaian dengan baik tanpa mengalami kesalahan karena kurangnya kemampuan komputasi.

**Kata Kunci:** *Computer Vision, Internet of Things, Keramaian, YOLO 3*

### 1. PENDAHULUAN

Keramaian adalah aspek sosial masyarakat yang bertemu dalam satu lingkup area tertentu di suatu waktu untuk berbaur atau bersosialisasi maupun berbagi informasi terkini. Tetapi ada juga bentuk lain dari keramaian yang terbentuk secara insidental karena ada peristiwa khusus



yang tidak terjadi secara rutin. Sebagai contoh sederhana dari kasus ini adalah demonstrasi baik yang dilakukan oleh mahasiswa[1], buruh[2], maupun anggota masyarakat lainnya[3]. Keramaian yang terjadi di masyarakat, memiliki dampak positif dan negatif. Salah satu kemungkinan yang dapat terjadi dengan adanya kerumuman seperti demonstrasi adalah kerusakan maupun anarkisme. Berdasarkan artikel berita[1] yang sempat terjadi keriuhan sehingga memaksa pihak keamanan menggunakan gas air mata untuk mengamankan kerusakan yang terjadi[4]. Berdasarkan artikel-artikel yang dikutip tersebut, sudah bisa diambil titik utama permasalahan bahwa keramaian yang terjadi dimasyarakat dapat memberikan dampak negatif yang cukup signifikan apabila tidak dikendalikan. Hal yang dapat terjadi seperti terganggunya transportasi umum[5] yang juga dapat berdampak ke perputaran ekonomi daerah hingga hilangnya nyawa seorang warga[6]. Dalam aspek kesehatan, juga dapat meningkatkan resiko penularan suatu penyakit seperti COVID-19 [7].

Untuk membantu mengenali potensi keramaian, diusulkanlah berbagai macam model deteksi keramaian[8] baik menggunakan media foto maupun video. Berdasarkan artikel[9], model yang diusulkan oleh penelitian ini menggunakan teknologi Wi-Fi untuk mendeteksi keramaian. Penelitian berikutnya kemudian menerapkan teknologi *Wireless Fidelity* dan *Software Defined Network* (SDN) untuk mendeteksi keramaian di suatu area berdasarkan tingkat keramaian penggunaan[10]. Selain itu juga terdapat penelitian lain yang menerapkan *Single Shot multi-Detector* untuk melakukan deteksi orang dan melakukan penghitungan di frame gambar yang diambil oleh model. Dari deteksi yang dilakukan, model dapat mendeteksi orang dengan akurasi mencapai 95% baik dalam maupun luar ruangan [11], [12]. Pengembangan untuk mendeteksi keramaian juga dikembangkan untuk perangkat *Edge Computing* dengan menerapkan *MobileNet-SSD* untuk mendeteksi, menghitung, dan melacak orang. Dengan menggunakan perangkat *UpSquared2* dengan kemampuan *Vision Processing Unit*, proses deteksi memiliki kinerja yang cepat[13].

Dari model-model penelitian yang telah diusulkan sebelumnya, terdapat kelemahan umum yang berkaitan dengan implementasi deteksi keramaian itu sendiri. Salah satu model yang diusulkan sebelumnya, hanya mengandalkan akses poin *WiFi* untuk mendeteksi keramaian. Tentu saja model ini tidak mengetahui apakah benar-benar ada orang atau tidak di area tersebut. Sehingga apabila ada satu (1) orang yang membawa dua (2)

perangkat gawai akan dihitung sebagai dua (2) orang. Sedangkan model-model lain yang menggunakan kecerdasan buatan diimplementasikan di perangkat komputer, sehingga model tidak bekerja optimal jika diimplementasikan di tingkat perangkat *Internet of Things* (IoT) dengan komputasi rendah[14]. Kelemahan lain yang dimiliki model-model sebelumnya adalah ketidakmampuan model untuk mendeteksi jarak antar orang.

Oleh karena itu, penelitian ini memiliki tujuan untuk mendesain sebuah model berbasis komputasi IoT yang mengimplementasikan metode *You Only Look Once version 3 Tiny* (YOLOv3-Tiny) untuk mendeteksi keramaian sekaligus menghitung dan mengukur jarak (berdasarkan 2 dimensi) masing-masing orang yang terdeteksi. Pengusulan metode ini didasarkan dari artikel[15] yang telah mampu menghitung larva udang dengan baik. Sehingga dapat diimplementasikan untuk mendeteksi manusia. Penggunaan metode YOLOv3 Tiny ini memiliki kinerja yang ringan jika dibandingkan dengan versi lainnya sehingga cocok untuk digunakan perangkat komputasi IoT nantinya[16]–[18].

## 2. METODOLOGI

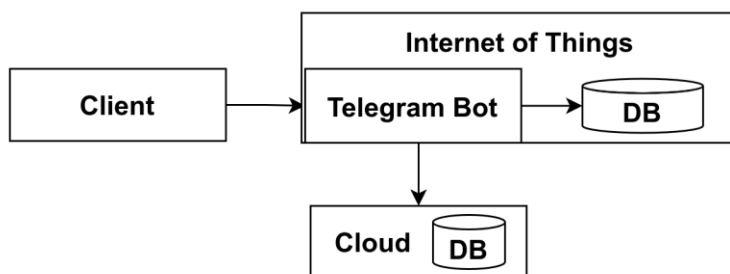
Penelitian ini menggunakan metode eksperimen untuk mendesain model yang diusulkan tersebut. Metode ini terdiri dari beberapa tahapan seperti: perancangan model, implementasi rangkaian model, dan Evaluasi. Berikut ini adalah tahapan yang akan dilakukan oleh peneliti untuk mendesain model deteksi dan penghitung keramaian:

### 2.1 Perancangan Model

Untuk dapat menyelesaikan masalah yang ditemukan. Penelitian ini merancang sebuah jaringan yang berisikan perangkat *Internet of Things* seperti *Raspberry Pi 3B Plus* yang terhubung dalam satu jaringan nirkabel utama. Karena perangkat ini sangat cocok digunakan untuk *Internet of Things*[19].

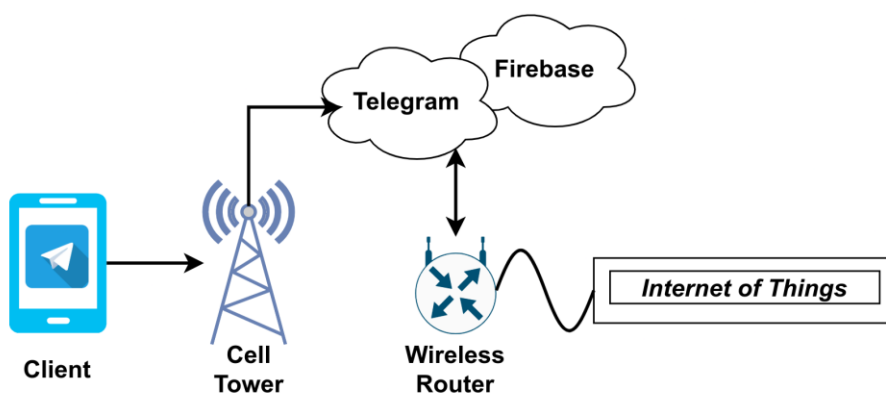
Untuk mendukung penyimpanan sementara di dalam perangkat IoT, penelitian ini menggunakan basis data *NoSQL* seperti *RethinkDB*[20]. Sedangkan untuk penyimpanan permanen di awan, penelitian ini menggunakan *Cloud Firestore* untuk menyimpan *metadata*[21] dan *Google Storage* untuk menyimpan data biner seperti gambar. Gambar 1 adalah

alur sederhana proses penyimpanan gambar dari klien ke IoT dan diteruskan ke perangkat awan:



**Gambar 1.** Alur komunikasi dari klien ke komputasi awan

Untuk bisa melakukan deteksi keramaian dan mengkalkulasikan jaraknya, klien akan mengirimkan sampel gambar ke *Internet of Things*. Perangkat ini akan memroses gambar yang tersimpan di database sementara dan di *backup* ke komputasi awan secara permanen. Untuk rangkaian jaringan yang digunakan oleh penelitian ini adalah sebagai berikut.

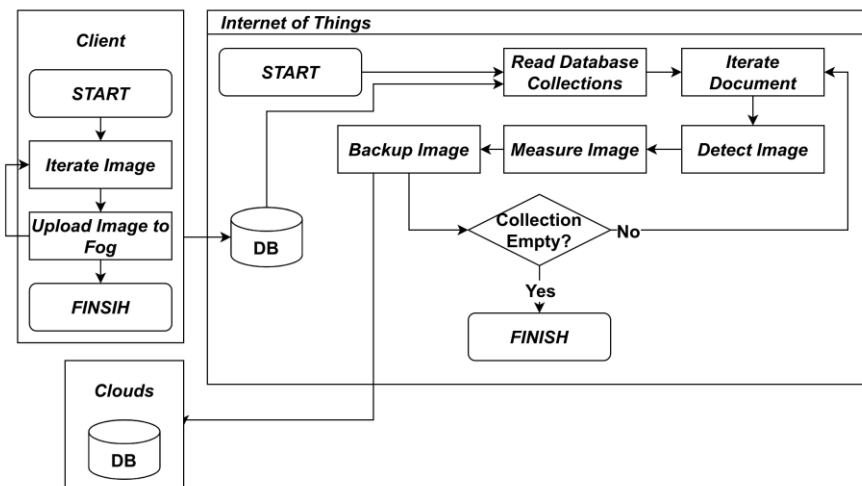


**Gambar 2.** Topologi Jaringan Model

Gambar 2 merupakan topologi jaringan yang digunakan oleh penelitian ini. Topologi jaringan ini terdiri dari klien yang terhubung ke jaringan seluler untuk mengunggah gambar ke Telegram. Sedangkan perangkat *Internet of Things* terhubung ke jaringan nirkabel untuk bisa membaca pesan dari klien, dan mengunggah hasilnya ke Telegram dan Firebase.

## 2.4 Implementasi Model

Perangkat *Internet of Things* ini menggunakan Raspberry Pi 3B + sebagai implementasi. Lalu masing-masing perangkat dihubungkan ke jaringan nirkabel melalui sebuah *router* yang juga terhubung ke internet. Di tahap ini, penataan perangkat keras komputasi IoT sudah selesai. Di tahap berikutnya adalah melakukan konfigurasi database di perangkat tersebut. Perangkat IoT maupun basis data awan harus dikonfigurasi terlebih dahulu. Untuk keperluan internal menggunakan basis data *RethinkDB*, dan untuk *backup* menggunakan *Cloud Firestore* milik *Google*. Pemilihan *Cloud Firestore* sebagai tempat *backup* karena memiliki kinerja dan latensi yang baik[22]. Untuk dapat menandai orang yang terdeteksi, menghitung jarak diantaranya dan melakukan proses *backup*, penelitian ini mendesain sebuah alur proses secara keseluruhan. Berikut ini adalah alur proses yang dilakukan oleh model yang diusulkan.



**Gambar 4.** Alur proses model yang diusulkan

Gambar 4 menjelaskan alur proses dari model yang diusulkan. Proses dimulai dengan klien melakukan pengunggahan data-data gambar yang sudah disiapkan oleh peneliti. Gambar-gambar ini kemudian disimpan di dalam basis data perangkat IoT. Tugas perangkat IoT adalah membaca basis data tersebut dan melakukan deteksi menggunakan *YOLOv3-Tiny*, melakukan pengukuran, dan memberikan tanda. Setelah tugas itu selesai, maka langkah terakhir untuk perangkat IoT adalah melakukan kopi ke

basis data komputasi awan secara permanen melalui Internet. Masing-masing perangkat IoT memiliki *Job ID* yang sudah ditentukan secara terprogram untuk memastikan tugas yang harus dijalankan terbagi secara adil dan merata.

Untuk melakukan penghitungan jarak, penelitian ini menggunakan kalibrasi kasar untuk dua dimensi (2D) dikarenakan batasan perlengkapan alat yang hanya mengandalkan kamera. Kalibrasi ini dilakukan dengan menggunakan alat ukur sederhana seperti penggaris, yang kemudian di foto dengan menggunakan kamera. Gambar yang dihasilkan oleh kamera tersebut diukur dengan rumus:

$$\text{kepadatan piksel (ppcm)} = \frac{\text{resolusi objek ukur (pixel)}}{\text{panjang objek ukur (cm)}} \quad (1)$$

Persamaan 1 ini nantinya akan mendapatkan tingkat densitas piksel dari suatu gambar. Jika terdapat kamera yang berbeda, maka tingkat densitasnya secara otomatis akan berbeda. Untuk berikutnya adalah menghitung jarak dunia nyata berdasarkan jarak yang dibuat oleh model. Maka untuk mendapatkan skala dari dunia nyata bisa dihitung dengan rumus berikut.

$$\text{skala jarak} = \frac{\text{perkiraan jarak asli (cm)}}{\left(\frac{\text{jarak hitungan model (pixel)}}{\text{densitas piksel (ppcm)}}\right)} = \frac{\text{perkiraan jarak asli (cm)}}{\text{jarak gambar (cm)}} \quad (2)$$

Dengan menggunakan Persamaan 2, penelitian ini bisa mendapatkan nilai skala jarak versi gambar dengan dunia nyata. Sehingga untuk mendapatkan perkiraan jarak dunia nyata bisa dihitung dengan rumus seperti berikut.

$$\begin{aligned} \text{perkiraan jarak (cm)} &= \left(\frac{\text{jarak hitungan model (pixel)}}{\text{kepadatan piksel (ppcm)}}\right) \\ &= \text{jarak gambar} * \text{skala jarak} \end{aligned} \quad (3)$$

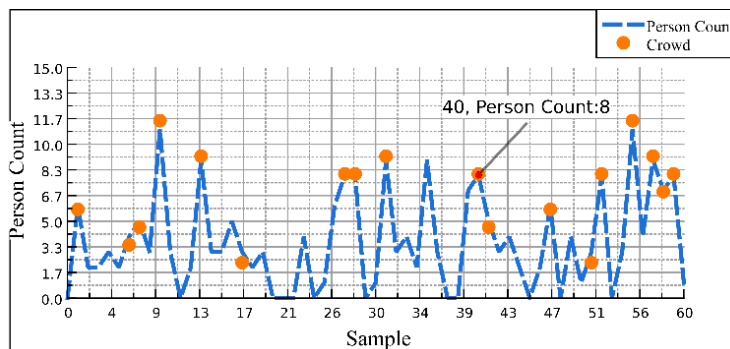
Dengan menggunakan Persamaan 3, hasil jarak perkiraan dalam ukuran centimeter didapatkan dan dilaporkan via Telegram Bot. Namun hasil yang didapatkan ini masih berupa estimasi perkiraan.

## 2.5 Evaluasi

Tahap ini merupakan tahap di mana perangkat IoT yang sudah diimplementasikan diuji dengan skenario yang sudah ditentukan. Dalam proses pengujian, penelitian ini menggunakan data gambar milik peneliti pribadi (dokumentasi pribadi) berjumlah 60 (enam puluh) foto dengan tiga (3) ukuran resolusi yang berbeda. Foto jenis pertama memiliki resolusi yang rendah (1200-1600 piksel), jenis kedua memiliki resolusi sedang (3264 piksel) dan jenis ketiga beresolusi tinggi (5500 piksel). Selain itu, konten yang terdapat di dalam gambar tersebut memiliki kemungkinan ada dan tidaknya keramaian. Sehingga proses pengujian menjadi *blind test*. Berikutnya adalah memastikan pembagian tugas untuk mendeteksi gambar-gambar ini tersebar secara merata, penelitian ini menggunakan konsep *round robin* dengan nilai maksimal lima (5) perangkat. Untuk proses pengujiannya sendiri, ada beberapa faktor penilaian yang diambil seperti: deteksi keramaian beserta jumlahnya, penggunaan CPU perangkat, penggunaan memori. Faktor-faktor tersebut nantinya akan menentukan bagaimana kinerja *YOLOv3 Tiny* yang diimplementasikan dalam komputasi IoT untuk mendeteksi keramaian.

## 3. HASIL DAN PEMBAHASAN

Penelitian ini telah berhasil mengimplementasikan model deteksi keramaian dengan menggunakan *YOLOv3-Tiny* dalam bentuk sebuah perangkat komputasi IoT. Setelah berhasil mengimplementasikan model tersebut, penelitian ini juga melakukan pengujian terhadap model tersebut. Untuk detail keseluruhan hasil deteksi ditampilkan dalam diagram seperti berikut.



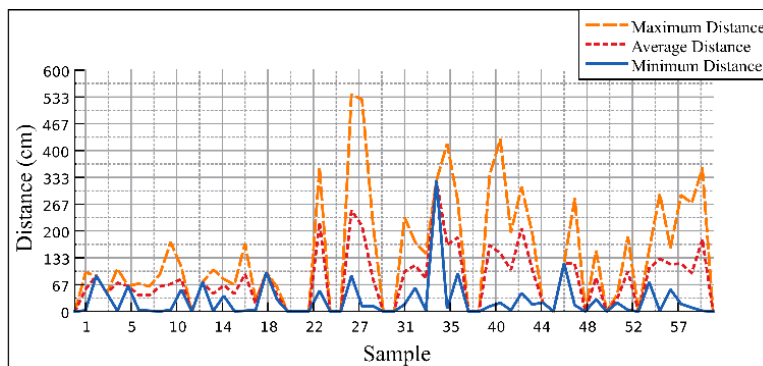
Gambar 5. Hasil deteksi keseluruhan data

Gambar 5 adalah hasil deteksi dengan menggunakan seluruh data yang digunakan selama pengujian. Dari 60 data gambar yang diberikan ke model, sebanyak 18 gambar terdeteksi memiliki keramaian di dalamnya. Sebagai contoh adalah data sampel ke 40 yang memiliki jumlah orang sebanyak delapan (8) dan terdeteksi sebagai keramaian. Gambar 6 adalah gambar dari sampel 40 yang terdeteksi memiliki keramaian:



**Gambar 6.** Hasil deteksi gambar nomor 40

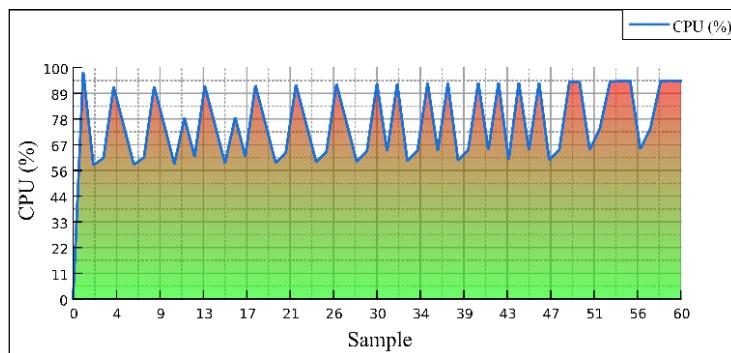
Berdasarkan hasil sampel gambar nomor 40, model yang diusulkan ini berhasil mendeteksi keramaian dengan jumlah total orang adalah delapan (8) orang. Model ini terpicu karena ada lebih dari tiga orang yang berdekatan titiknya satu sama lainnya. Sehingga gambar ini terdeteksi memiliki keramaian meskipun tidak berjumlah banyak. Hasil pengujian yang dilakukan ini juga menghasilkan pengukuran jarak yang bisa dilihat di Gambar 7.



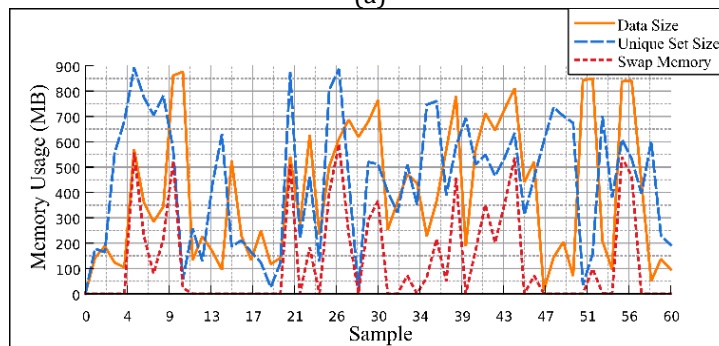
**Gambar 7.** Hasil pengukuran jarak oleh model



Gambar 7 merupakan hasil pengukuran jarak yang didapatkan melalui kalibrasi kasar untuk mengetahui jarak antara satu orang dengan orang lainnya berdasarkan titik tengah deteksi (Lihat Gambar 6). Terdapat tiga (3) indikator jarak yang didapatkan oleh model berupa: jarak minimal, maksimal, dan rata-rata. Jarak minimal memperkirakan jarak terdekat. Jarak maksimal memperkirakan jarak terjauh, dan jarak rata-rata memperkirakan jarak pertengahan di antara kedua indikator. Dari hasil deteksi yang dilakukan, model ini mendapatkan rata-rata jarak terdekat adalah 27,53 cm. Jarak terjauh adalah 143,21cm, dan jarak rata-rata adalah 79,64cm. Hasil pengukuran ini bersifat kasar sehingga memiliki kadar kesalahan yang cukup tinggi karena mengabaikan aspek tiga dimensi yang tidak dipunyai oleh kamera aset penelitian ini. Pengujian berikutnya adalah pengujian kinerja model untuk memroses masing-masing gambar. Berikut ini adalah hasil penggunaan CPU dan memori yang diatur dalam bentuk sampel data gambar.



(a)



(b)

**Gambar 8.** Tingkat penggunaan CPU (a) dan memori (b) untuk deteksi keramaian

Gambar 8 (a) memperlihatkan kebutuhan CPU yang ada di dalam perangkat komputasi IoT untuk mendeteksi data gambar. Hasil *benchmark* yang dihasilkan oleh penelitian ini tersimpan dalam skala 400% karena dihitung dari jumlah inti prosesor yang ada di dalam IoT (4 inti x 100%). Sehingga penelitian ini melakukan menurun skala hasil *benchmark* dari 400% ke 100% agar lebih mudah untuk dipahami. Dari hasil deteksi 60 gambar, model yang diusulkan ini menggunakan CPU dengan rata-rata 304,9% dalam skala 400% atau 76,225% dalam skala 100%. Hal ini memiliki arti bahwa model yang diusulkan memerlukan komputasi sebanyak tiga inti prosesor untuk mendeteksi keramaian di gambar. Berdasarkan Gambar 8 (b), penggunaan memori untuk menjalankan proses (diindikasikan sebagai *Unique Set Size*) membutuhkan rata-rata 454,78MB, dan untuk menyimpan data yang akan diolah (dengan indikasi *Data Size*) sebanyak 405,61MB. Jika jumlah total antara *Unique Set Size* dan *Data Size* melebihi total RAM yang dimiliki perangkat komputasi IoT (yaitu 909MB) maka secara otomatis sistem operasi akan menggunakan memori virtual dengan istilah *Swap Memory*. Dalam penelitian ini, model hanya membutuhkan rata-rata 130,94MB saja untuk menompang proses deteksi 60 gambar. Penggunaan memori yang cukup rendah (jika dibandingkan dengan komputer umum atau *edge*) ini disebabkan oleh model deteksi yang lebih kecil dibandingkan algoritma *YOLOv3*. Sehingga *YOLOv3-Tiny* yang diimplementasikan di perangkat IoT ini lebih efisien untuk digunakan daripada algoritma sebelumnya.

Dari hasil pengujian yang telah dilakukan, penelitian ini mengambil poin-poin penting dan membahas hasil yang didapatkan dari penelitian. Pembahasan pertama berkaitan dengan hasil deteksi yang diperlihatkan di Gambar 6, model berhasil mendeteksi beberapa orang. Dari gambar tersebut, model berhasil mengukur jarak dengan minimal 27,53 cm dan maksimal 143,21cm. Hasil ini didapatkan oleh model dengan melakukan kalkulasi jarak gambar berdasarkan densitas pixel dan jarak di gambar (*in-picture distance*), sehingga tidak 100% mewakili jarak aslinya. Pembahasan berikutnya adalah kinerja dari model komputasi tersebut. Jika dilihat dari aspek penggunaan CPU yang mencapai 304,9% (atau 76,225% dengan skala 100%), perangkat ini masih memiliki satu inti tersisa untuk menjalankan kebutuhan sistem lainnya. Sehingga model tidak sepenuhnya mengganggu ketersediaan CPU untuk sistem. Pembahasan selanjutnya adalah mengenai penggunaan memori yang menjadi hal krusial. Jika dilihat dari hasil kinerjanya, model terkadang membutuhkan memori melebihi ketersediaan RAM mencapai 909MB.

Untuk mengatasi permasalahan kurangnya memori, penelitian ini telah menambahkan memori virtual tambahan berukuran satu *Gigabyte* untuk menompang proses deteksi. Dari hasil pembahasan ini bisa disimpulkan bahwa model yang diusulkan ini mampu mendeteksi keramaian beserta perkiraan jarak di antara orang-orangnya, dan memiliki tingkat komputasi yang cukup untuk perangkat IoT. Sehingga perangkat yang digunakan dapat bekerja dengan baik mendeteksi, menghitung keramaian, dan melakukan *backup* data deteksi ke perangkat awan dengan baik.

#### 4. KESIMPULAN

Keramaian adalah aspek sosial yang memiliki manfaat yang baik dan buruk. Kegiatan tersebut merupakan salah satu bentuk cara bersosialisasi satu dengan sama lainnya. Namun juga dapat berdampak buruk apabila keramaian berubah menjadi demonstrasi anarkisme. Untuk mendukung mendeteksi keramaian, terdapat beberapa penelitian yang difokuskan untuk mendesain model deteksi. Namun sayangnya, penelitian yang telah dilakukan tersebut memiliki kelemahan dalam implementasinya. Oleh karena itu penelitian ini memiliki tujuan untuk mendesain sebuah model yang mampu berjalan di atas perangkat komputasi IoT seperti *Raspberry Pi 3B plus* dan mampu melakukan deteksi dan pengukuran jarak manusia. Dari hasil pengujian yang telah dilakukan, model mampu mendeteksi keramaian. Sedangkan untuk hasil *benchmark* yang dilakukan, model menggunakan CPU sebanyak 76,22% dari total semua CPU perangkat IoT. Untuk penggunaan memori memiliki rata-rata 454,78MB untuk *Unique Set Size*, 405,61MB untuk pengolahan *Data*, dan 130,94MB untuk memori virtual. Untuk dapat menompang kinerja lebih jauh, sangat disarankan untuk menggunakan *Raspberry Pi* dengan versi yang lebih tinggi.

#### REFERENSI

- [1] I. Fathurohman, "Massa BEM SI Aksi Rempang dan Seruyan Dihadang Polisi di Kejagung," *IDN Times*, Jakarta, Oct. 13, 2023. Accessed: Oct. 13, 2023. [Online]. Available: <https://www.idntimes.com>
- [2] F. Halim and F. Peace Simbolon, "Demo Buruh di Patung Kuda, Waspada Macet: Arus Lalu Lintas Dialihkan," *VIVAnews*, Jakarta, Oct. 02, 2023. Accessed: Oct. 13, 2023. [Online]. Available: <https://www.viva.co.id>

- [3] A. Hamapu, "Warga Demo Tolak Relokasi saat Menteri Bahlil ke Rempang Batam," *detikNews*, Kepulauan Riau, Oct. 06, 2023. Accessed: Oct. 13, 2023. [Online]. Available: <https://www.detik.com>
- [4] I. Fathurohman, "Polisi Bersenjata Gas Air Mata Hadang Mahasiswa di Depan Kejangung," *IDN Times*, Jakarta, Oct. 13, 2023. Accessed: Oct. 13, 2023. [Online]. Available: <https://www.idntimes.com>
- [5] H. B. Alexander, "Macet Panjang 3 Kilometer di GT Jatikarya, Pengelola: Dampak Demo Warga Ahli Waris Lahan," *Kompas*, Jakarta, 2022-10-07. Accessed: Oct. 14, 2023. [Online]. Available: <https://www.kompas.com>
- [6] E. Dyah Fitriani, "Kecam Demo Ricuh di Seruyan Tewaskan Warga, Komisi III DPR: Polisi Gegabah," *detikNews*, Jakarta, Oct. 10, 2023. Accessed: Oct. 13, 2023. [Online]. Available: <https://news.detik.com>
- [7] C. Liu *et al.*, "The impact of crowd gatherings on the spread of COVID-19," *Environ. Res.*, vol. 213, p. 113604, 2022, doi: <https://doi.org/10.1016/j.envres.2022.113604>.
- [8] F. H. Utkarsh Singh Jean-François Determe and P. D. Doncker, "Crowd Monitoring: State-of-the-Art and Future Directions," *IETE Tech. Rev.*, vol. 38, no. 6, pp. 578–594, 2021, doi: [10.1080/02564602.2020.1803152](https://doi.org/10.1080/02564602.2020.1803152).
- [9] J. R. Santana, L. Sánchez, P. Sotres, J. Lanza, T. Llorente, and L. Muñoz, "A Privacy-Aware Crowd Management System for Smart Cities and Smart Buildings," *IEEE Access*, vol. 8, pp. 135394–135405, 2020, doi: [10.1109/ACCESS.2020.3010609](https://doi.org/10.1109/ACCESS.2020.3010609).
- [10] M. Mu, "WiFi-based Crowd Monitoring and Workspace Planning for COVID-19 Recovery." arXiv, 2020. doi: [10.48550/ARXIV.2007.12250](https://doi.org/10.48550/ARXIV.2007.12250).
- [11] M. Ahmad, I. Ahmed, K. Ullah, and M. Ahmad, "A Deep Neural Network Approach for Top View People Detection and Counting," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Oct. 2019, pp. 1082–1088. doi: [10.1109/UEMCON47517.2019.8993109](https://doi.org/10.1109/UEMCON47517.2019.8993109).
- [12] I. Ahmed, M. Ahmad, A. Ahmad, and G. Jeon, "IoT-based crowd monitoring system: Using SSD with transfer learning," *Comput. Electr. Eng.*, vol. 93, p. 107226, 2021, doi: <https://doi.org/10.1016/j.compeleceng.2021.107226>.
- [13] A. C. Cob-Parro, C. Losada-Gutiérrez, M. Marrón-Romera, A. Gardel-Vicente, and I. Bravo-Muñoz, "Smart Video Surveillance System Based on Edge Computing," *Sensors*, vol. 21, no. 9, 2021, doi: [10.3390/s21092958](https://doi.org/10.3390/s21092958).

- [14] M. Laroui, B. Nour, H. Moun gla, M. A. Cherif, H. Afifi, and M. Guizani, "Edge and fog computing for IoT: A survey on current research activities & future directions," *Comput. Commun.*, vol. 180, pp. 210–231, 2021, doi: <https://doi.org/10.1016/j.comcom.2021.09.003>.
- [15] S. Armalivia, Z. Zainuddin, A. Achmad, and Muh. A. Wicaksono, "Automatic Counting Shrimp Larvae Based You Only Look Once (YOLO)," in *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, Apr. 2021, pp. 1–4. doi: [10.1109/AIMS52415.2021.9466058](https://doi.org/10.1109/AIMS52415.2021.9466058).
- [16] Z.-F. Xu, R.-S. Jia, Y.-B. Liu, C.-Y. Zhao, and H.-M. Sun, "Fast Method of Detecting Tomatoes in a Complex Scene for Picking Robots," *IEEE Access*, vol. 8, pp. 55289–55299, 2020, doi: [10.1109/ACCESS.2020.2981823](https://doi.org/10.1109/ACCESS.2020.2981823).
- [17] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application," *IEEE Access*, vol. 8, pp. 9102–9114, 2020, doi: [10.1109/ACCESS.2020.2964608](https://doi.org/10.1109/ACCESS.2020.2964608).
- [18] K. M and K. P. R, "Comparative Analysis of YOLOv3-320 and YOLOv3-tiny for the Optimised Real-Time Object Detection System," in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, Apr. 2022, pp. 495–500. doi: [10.1109/ICIEM54221.2022.9853186](https://doi.org/10.1109/ICIEM54221.2022.9853186).
- [19] Q. Xu and J. Zhang, "piFogBed: A Fog Computing Testbed Based on Raspberry Pi," in *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, Oct. 2019, pp. 1–8. doi: [10.1109/IPCCC47392.2019.8958741](https://doi.org/10.1109/IPCCC47392.2019.8958741).
- [20] W. Wingerath, N. Ritter, and F. Gessert, "Real-Time Databases," in *Real-Time & Stream Data Management: Push-Based Data in Research & Practice*, W. Wingerath, N. Ritter, and F. Gessert, Eds., Cham: Springer International Publishing, 2019, pp. 21–41. doi: [10.1007/978-3-030-10555-6\\_3](https://doi.org/10.1007/978-3-030-10555-6_3).
- [21] R. Kesavan, D. Gay, D. Thevessen, J. Shah, and C. Mohan, "Firestore: The NoSQL Serverless Database for the Application Developer," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 2023, pp. 3367–3379.
- [22] Y. Sukmana and Y. Rosmansyah, "The Use of Cloud Firestore For Handling Real-time Data Updates: An Empirical Study of Gamified Online Quiz," in *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, Dec. 2021, pp. 1239–1244. doi: [10.1109/CECIT53797.2021.00220](https://doi.org/10.1109/CECIT53797.2021.00220).