

PEMODELAN APLIKASI PENDETEKSI PELAKSANAAN SOCIAL DISTANCING DALAM MEMBANTU PENGAWASAN PROGRAM 3M

Arnold Nasir¹⁾, Stefany Yunita Baralangi²⁾

^{1,2}Prodi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Atma Jaya Makassar
Alamat e-mail: arnold_nasir@outlook.com¹⁾, stefany_baralangi@lecturer.uajm.ac.id²⁾

ABSTRACT

The Coronavirus has made severe damage to the society around the world. Every aspect of human life was affected. As the result, the government propose the “new normal” policy to be applied as the new standards for regulating people in doing their activities. One of the rules within the new normal policy is to implement social distancing in a public space. In order to enforce the rule, the implementation of computer vision technology by using CNN becomes one of the solutions, especially in the university environment.

Keywords: *Convolutional Neural Network, CNN, detection, social distancing, tensorflow*

1. PENDAHULUAN

Novel coronavirus, atau penyakit menular yang dikenal sebagai coronavirus (COVID-19), telah mendatangkan malapetaka pada orang-orang di seluruh dunia. Tidak kurang dari 170 ekonomi negara di dunia diproyeksikan mengalami resesi sejak kemunculannya pada awal 2020[1], sehingga diperlukan upaya untuk mencegah dampak ekonomi yang lebih buruk di masa depan. Salah satu upaya yang dilakukan adalah penerapan kebijakan “new normal”, sebuah standar baru untuk mengatur cara hidup masyarakat pasca wabah korona.

Menurut Achmad Yurianto, Juru Bicara Penanganan COVID-19, new normal merupakan adaptasi masyarakat terhadap cara hidup baru, dan untuk meningkatkan produktivitas masyarakat yang mendukung dan mempertahankan produktivitas meskipun ada ancaman, masyarakat diharapkan dapat menjaga pola hidup bersih dan sehat selama virus corona di masyarakat masih ada[2]. Salah satu pola hidup sehat yang disebut pedoman new normal adalah dengan menerapkan *social distancing* saat berada di area umum.

Penerapan new normal diharapkan dapat menahan penyebaran virus corona. Namun, pemantauan penerapan new normal di masyarakat, seperti pemantauan menjaga jarak antar satu dengan yang lain, tidaklah mudah. Menurut Rezi Erdiansyah, banyak faktor yang membuat masyarakat lalai dalam

menjaga jarak atau yang juga dikenal dengan istilah *social distancing*. Meskipun keresahan masyarakat terhadap COVID-19 namun ikatan relasi sosial antar masyarakat masih lebih kuat dalam perspektif masyarakat, seperti contoh bentuk interaksionis simbolik yang dapat diamati secara langsung pada perilaku masyarakat. Adapun bentuk interaksi yang dimaksud seperti kebersamaan, kerja sama, solidaritas, dan sejenisnya sebagai bentuk dari interaksi sosial. Selain itu, bagi Sebagian masyarakat awam menganggap *social distancing* menyerupai menjaga jarak pada saat mengantri pada mesin ATM[3]. Oleh karena itu, perlu adanya mekanisme pemantauan yang dapat mendukung proses pemantauan *social distancing* di tempat-tempat umum seperti lingkungan kampus.

Salah satu teknik yang paling banyak digunakan saat ini adalah penggunaan teknologi computer vision, yang mengajarkan komputer untuk mengenali objek yang berbeda secara visual. Saat mengembangkan teknologi visi komputer yang sangat akurat, perlu untuk menggabungkannya dengan teknologi pembelajaran mesin. Salah satu platform yang paling banyak digunakan adalah TensorFlow.

Implementasi TensorFlow pada teknologi computer vision bertujuan untuk membuat aplikasi yang dapat mendukung proses pemantauan social distancing di lingkungan universitas. Selain itu, kehadiran

aplikasi pendeteksi masker wajah dapat mencegah penyebaran dan pembentukan klaster virus corona baru.

2. TINJAUAN PUSTAKA

2.1 Kajian Penelitian Sejenis

Adapun beberapa kajian penelitian sejenis yang digunakan sebagai acuan dalam kegiatan penelitian ini, diantaranya:

1. Srishti Verma dan Prashant Kumar Jain dengan judul penelitian: “*COVID-19: Automatic Social Distancing Rule Violation Detection using PP-Yolo & Tensorflow in OpenCV*” [4]. Penelitian ini berfokus pengembangan metode deteksi pelanggaran aturan terkait pelaksanaan *social distancing* pada area umum seperti stasiun kereta api, mall, pasar, dan bandar udara. Metode yang digunakan dalam penelitian ini meliputi implementasi PP-Yolo & Tensorflow pada OpenCV sehingga menghasilkan sebuah sistem yang dapat memiliki level akurasi yang tinggi serta tingkat kesalahan deteksi yang rendah.
2. Shesang Degadwala, Dhairya Vyas, Harsh Dave, dan Arpana Mahajan dengan judul penelitian: “*Visual Social Distance Alert System Using Computer Vision & Deep Learning*” [5]. Penelitian ini berfokus pada proses pengembangan sistem dengan menerapkan computer vision dan *deep learning* dalam mempelajari penerapan *social distancing* berdasarkan video pengamatan. Hasil yang didapatkan dari penelitian ini adalah kemampuan sistem dalam melakukan proses screening yang cepat dan akurat dalam memetakan potensi penyebaran wilayah pandemi berdasarkan aktivitas sosial masyarakat misalnya dalam berinteraksi dengan jarak yang cukup dekat dan tidak mematuhi peraturan pemerintah yang disarankan.

2.2 Convolutional Neural Network

Salah satu metode *deep learning* yang saat ini banyak digunakan dalam penelitian terkait *computer vision* adalah *Convolutional Neural Network* (CNN). CNN terdiri dari banyak lapisan yang tersembunyi, dimana tiap-tiap lapisan melakukan proses

perhitungan matematika dengan input dan output yang nantinya output pada setiap lapisan menjadi inputan pada lapisan berikutnya. CNN sendiri merupakan metode yang umum digunakan dalam proses *image classification*, misalnya R-CNN atau sering disebut *Regional Convolutional Neural Network*. Prinsip kerja dari R-CNN adalah dengan membuat kotak pembatas pada sebuah gambar, kemudian area yang berada didalam kotak tersebut akan diklasifikasikan. Selain itu, ada pula metode lain dalam proses *image processing* yaitu *You Only Look Once* (YOLO). YOLO merupakan metode yang memiliki proses deteksi objek yang cepat dan banyak digunakan dalam proses deteksi objek dikarenakan metodenya yang membaca gambar secara keseluruhan selama proses pelatihan dan waktu tes sehingga menghasilkan informasi yang kontekstual.

Pada dasarnya lapisan pada CNN dapat dikategorikan dalam dua kategori yaitu:

- a. Lapisan ekstraksi fitur, dimana lapisan terletak pada bagian awal arsitektur dimana lapisan ini berfungsi untuk menerima input gambar lalu diproses oleh neuron yang terkoneksi pada area lokal pada lapisan sebelumnya.
- b. Lapisan klasifikasi, dimana lapisan ini tersusun atas neuron yang terhubung secara penuh dengan lapisan lainnya. Lapisan ini mendapatkan inputan dari output lapisan ekstraksi fitur berupa vektor yang diproses dan ditransformasikan dengan tambahan beberapa lapisan tersembunyi atau juga dikenal dengan istilah *hidden layer*. Output dari lapisan ini berupa penilaian kelas terhadap hasil klasifikasi.

2.3 TensorFlow

TensorFlow merupakan salah satu open-source machine-learning library yang umum digunakan dalam melakukan proses analitik berskala besar dan komputasi machine learning. TensorFlow dikembangkan oleh tim peneliti dan arsitek Google Brain dari Google's Machine Learning Research Organisation dan bertujuan untuk membantu kegiatan penelitian di bidang machine learning dan deep neural network. Beberapa contoh penggunaan TensorFlow antara lain melakukan partisi digit tulisan tangan, identifikasi gambar, pencocokan kata, natural

language processing (NLP), dan sequence-to-sequence method untuk interpretasi mesin.

Dalam melakukan proses analitik, TensorFlow menggunakan grafik aliran data yang bertujuan untuk merepresentasikan komputasi, shared state, dan perhitungan operasi yang mempengaruhi keadaan tertentu. Sementara itu, untuk proses komputasi TensorFlow menggabungkan beberapa model dan algoritma *machine learning* dan *neural networks*.

3. METODOLOGI PENELITIAN

Dalam melaksanakan kegiatan penelitian ini, peneliti memilih menerapkan *Rapid Application Development* (RAD) sebagai metode pengembangan aplikasi guna mendapatkan hasil yang lebih cepat dibandingkan dengan menerapkan metode pengembangan aplikasi tradisional seperti metode *Waterfall*. Hal ini dapat tercapai berkat siklus RAD yang terbilang singkat (90 hari) serta metode ini juga menerapkan penggunaan teknik yang berulang dalam pengembangan dari *prototype* hingga menjadi *final product* seperti yang dinyatakan oleh Murch [6]. Selain itu, James Martin berpendapat bahwa RAD tidak hanya membutuhkan waktu yang singkat dalam mengembangkan sebuah software tetapi juga memberikan kualitas end-product yang lebih baik bila dibandingkan metodologi tradisional seperti metode *Waterfall*.

Menurut Watkins [7], RAD memiliki kerangka waktu yang tetap dalam mengembangkan sebuah aplikasi ataupun sistem, mulai dari 30, 60, hingga 90 hari. Dengan demikian, apabila peneliti ingin mengembangkan *software* atau aplikasi yang ternyata waktu penyelesaiannya melebihi kerangka waktu terlama yang dimiliki RAD maka proyek tersebut dipastikan tidak akan selesai dengan hasil yang diharapkan.



Gambar 1. Bagan Metode RAD

Gambar 1 menunjukkan fase-fase pengembangan yang terdapat dalam RAD, antara lain:

1. Analisa

Pada fase ini peneliti mengumpulkan data yang diperlukan dalam mengembangkan aplikasi yang dirancang. Observasi serta studi literatur merupakan instrumen penelitian yang dipilih untuk digunakan dalam mengumpulkan data. Dari kegiatan observasi diperoleh data yang diperlukan dalam membantu proses kegiatan analisa perbandingan kedua metode *distance classifier*. Setelah data-data telah dikumpulkan, kemudian dilakukan analisa terkait waktu yang dibutuhkan agar penelitian ini dapat diselesaikan sesuai dengan target waktu yang telah ditentukan.

2. Siklus Prototyping

Pada fase ini, peneliti mulai mengembangkan aplikasi secara bertahap berdasarkan hasil analisa terhadap data-data yang telah dikumpulkan. Proses pengembangan dimulai dengan pemodelan terhadap fitur yang didukung oleh aplikasi yang dirancang menggunakan diagram *use case* maupun diagram aktivitas. Selanjutnya, dengan menggunakan sistem yang telah ada sebagai pedoman dasar dalam merancang aplikasi yang diinginkan, peneliti kemudian mengubah atau menambahkan beberapa fitur baru ke dalam sistem yang akan menghasilkan sebuah prototipe. Dengan menggunakan prototipe sebagai acuan dalam pengembangan aplikasi proses pengembangan dapat berjalan lebih singkat. Hal ini dikarenakan peneliti hanya melakukan perubahan maupun perbaikan pada prototipe sehingga tidak perlu merubah rancangan desain aplikasi secara keseluruhan. Setelah peneliti selesai membangun dan menyempurnakan prototipe, maka selanjutnya prototipe tersebut akan didemonstrasikan. Pada tahapan ini, prototipe akan diuji oleh beberapa pengguna, kemudian pengguna memberikan evaluasi terhadap prototipe, dengan memberikan beberapa masukan berupa perbaikan-perbaikan kecil maupun besar. Namun, jika pengguna berpendapat bahwa prototipe tidak memenuhi ekspektasi calon pengguna, maka peneliti harus mengulang proses mengembangkan prototipe.

3. Pengujian

Setelah prototipe telah memenuhi kriteria yang telah ditetapkan, maka prototipe akan masuk dalam fase pengujian. Pada fase ini, prototipe akan diuji dengan serangkaian uji kelayakan yang telah ditentukan oleh peneliti untuk mengetahui apakah masih terdapat kecacatan yang mungkin tidak nampak pada saat proses pengembangan. Hasil dari setiap uji kelayakan akan dicatat dan dirangkum sehingga dapat digunakan untuk pengembangan prototipe yang serupa dikemudian hari.

4. Implementasi

Fase implementasi merupakan fase terakhir dari proses pengembangan sebuah aplikasi menggunakan metode RAD. Setelah prototipe berhasil melewati fase pengujian maka prototipe telah siap sepenuhnya untuk digunakan atau dioperasikan.

4. HASIL DAN PEMBAHASAN

4.1 Analisa Kebutuhan

4.1.1 Observasi

Kegiatan penelitian ini berfokus pada area publik yang memiliki fasilitas CCTV sehingga memudahkan dalam proses pengambilan gambar maupun video yang dibutuhkan dalam pengembangan dataset. Adapun dalam penelitian ini mengambil wilayah area terbuka pada universitas dikarenakan wilayah area terbuka memiliki potensi yang lebih tinggi dalam mendapatkan lebih dari tiga objek manusia yang terekam dalam satu frame. Kamera diatur sedemikian rupa sehingga menangkap pada sudut tetap dan tampilan bingkai video diubah menjadi tampilan *bird-view* 2D untuk memperkirakan jarak antara setiap orang secara akurat sehingga gambar manusia di dalam bingkai diratakan pada bidang horizontal. Kemudian, empat titik dari bidang horizontal dipilih dan diubah menjadi *bird-eye view*.

Dari hasil observasi yang dilakukan didapatkan bahwa area seperti koridor, taman, dan kantin yang merupakan tempat banyaknya mahasiswa berkumpul menjadi area yang paling banyak dikunjungi oleh para mahasiswa di area universitas.

Jarak antara objek manusia dapat diperkirakan, diskalakan, dan diukur dengan menghitung jarak euclidean antara titik tengah objek.

4.2 Pembentukan dan Pelatihan Dataset

Proses pembentukan dataset yang digunakan dalam kegiatan penelitian ini bersumber dari dua sumber, yaitu dari dataset yang didapatkan dari cocodataset.org dan yang kedua didapatkan dari data gambar hasil *splicing* video CCTV yang didapatkan.

Data berupa gambar-gambar baik individu maupun kelompok orang dalam suatu keramaian, diperoleh baik dengan pengumpulan data manual maupun dari internet. Data yang didapat adalah 935. Data tersebut dibagi menjadi 653 data latih dan 282 data uji.

Gambar yang didapatkan dari cocodataset.org akan langsung masuk ke tahapan berikutnya yaitu proses pemberian label dimana proses ini penting dalam melatih *machine learning* yang nanti akan digunakan untuk mengenali objek manusia. Sementara itu, untuk gambar hasil *splicing* video CCTV tentunya melalui tahapan verifikasi, diantaranya:

- Memastikan video CCTV yang dijadikan sumber memiliki kualitas resolusi sekurang-kurangnya 720p dan 60 fps.
- Video CCTV yang lolos pada tahapan diatas nantinya akan dilakukan proses *splicing* secara manual menggunakan aplikasi Adobe Premiere guna mendapatkan hasil *export* yang memadai serta untuk memudahkan peneliti dalam melakukan pemberian label.

Setelah dataset gambar dikumpulkan maka proses berikutnya adalah dengan memberikan label. Proses pemberian label ini juga dikenal dengan sebutan *annotation* dimana proses ini peneliti memberikan informasi spesifik terkait objek yang nantinya dideteksi sebagai objek manusia pada sebuah gambar dataset.



Gambar 2. Ilustrasi IoU dan *ground-truth bounding box* vs *predicted bounding box*

Pada proses *annotation* dilakukan pula penentuan koordinat *ground-truth bounding box*, kemudian nantinya dibandingkan

predicted bounding box sehingga menghasilkan nilai *Intersection over Union* yang berperan dalam memvisualisasikan hasil deteksi *machine learning*.

Pada saat penentuan koordinat *ground-truth bounding box*, titik tengah pada *bounding box* menjadi acuan untuk melacak pergerakan objek pada gambar. Proses ini dilakukan oleh *mobile robot* dimana *mobile robot* bergerak berdasarkan dua titik tengah: titik tengah relatif terhadap sel; dan titik tengah relatif terhadap *bounding box*.

Untuk menghitung sudut, pertama-tama menentukan ukuran bingkai untuk mendeteksi objek. Dalam persamaan 3 dan 4, ukuran layar ditentukan oleh variabel l_x dan l_y . Koordinat pusat relatif terhadap sisi sel dihitung dengan menambahkan t_x dan t_w untuk mendapatkan nilai C_x , dan menambahkan t_y dan t_h untuk mendapatkan nilai C_y .

$$C_x = \frac{t_x + t_w}{2} \quad (1)$$

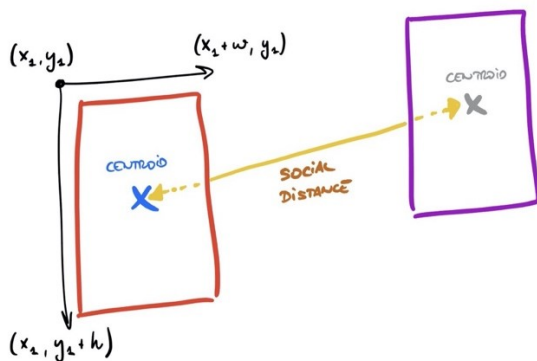
$$C_y = \frac{t_y + t_h}{2} \quad (2)$$

$$S_x = \left(\frac{l_x}{2} - C_x \right) + 1 \quad (3)$$

$$S_y = \left(l_y - C_y \right) + 1 \quad (4)$$

$$Angle = \frac{180 \times \arctan \frac{S_x}{S_y}}{\pi} \quad (5)$$

Dataset yang telah melewati proses *annotation* selanjutnya akan melalui proses *training* dimana menghasilkan *weight* atau bobot. Bobot sendiri merupakan variabel yang dipakai untuk mendeteksi objek pada sebuah gambar. Hasil dari proses deteksi gambar berupa *confidence* atau tingkat keyakinan, nama kelas, dan koordinat *bounding box*.



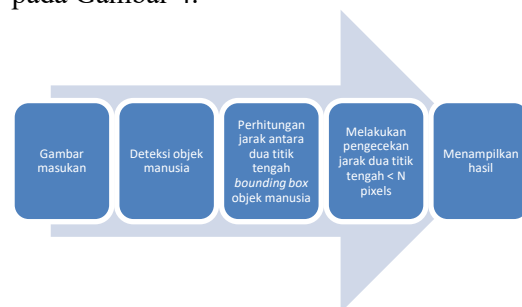
Gambar 3. Perhitungan jarak tengah antar *bounding box*

Proses deteksi objek manusia secara singkat dapat dijelaskan sebagai berikut:

- Pertama-tama gambar akan dimasukkan kedalam aplikasi.
- Setelah gambar dimasukkan maka aplikasi melakukan *feature extraction*.
- Setelah melalui proses *feature extraction* maka selanjutnya aplikasi akan melakukan proses *object classification* untuk memetakan objek yang terdapat pada gambar.
- Proses terakhir adalah *output* dimana aplikasi menghasilkan file gambar yang telah dilengkapi dengan label dari setiap objek yang berhasil dideteksi oleh aplikasi. Jumlah objek yang berhasil dideteksi bergantung pada berapa banyak kelas objek yang diberikan label oleh peneliti. Misalnya, peneliti tidak hanya membuat label terhadap objek manusia, tetapi juga memberikan label terhadap objek umum (*common object*) seperti sofa, bola, daun, dsb sehingga aplikasi tidak hanya dapat mendeteksi objek manusia saja tetapi juga objek-objek lain yang telah diberikan label.

Tujuan dari pelatihan ini adalah agar aplikasi mampu untuk mengisolasi proses deteksi sehingga hanya menampilkan kelas objek manusia saja dan bukan kelas dari objek lain.

Setelah aplikasi telah mengenali objek manusia, maka selanjutnya peneliti merancang model analisis jarak antara dua objek manusia pada sebuah frame gambar. Secara umum proses analisa *social distancing* pada aplikasi yang dirancang dapat dilihat pada Gambar 4.



Gambar 4. Tahapan model analisis jarak antara dua objek manusia

4.3 Hasil Pengujian

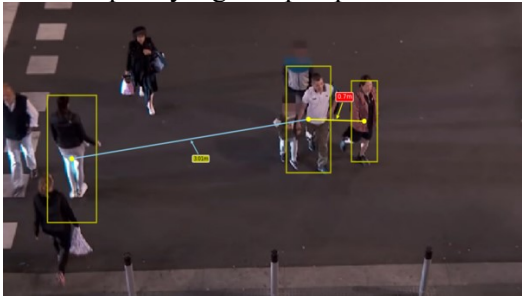
Setelah prototipe melalui proses penyempurnaan maka selanjutnya prototipe akan diuji untuk melihat kemampuannya dalam mendeteksi pelaksanaan *social distancing*.

Dari hasil pengujian mendapatkan tingkat akurasi prototipe dalam objek manusia mencapai 90%. Adapun setelah dilakukan pengujian stress testing, aplikasi mampu berjalan dengan mengkonsumsi *resource* RAM paling tinggi 4 GB.



Gambar 4. Hasil pengujian deteksi objek manusia dari dataset

Setelah dilakukan pengujian terhadap proses deteksi maka peneliti melakukan pengujian terhadap gambar hasil *splicing* dan didapatkan bahwa aplikasi mampu untuk menghitung jarak antar dua *bouding box*. Adapun indikator yang digunakan untuk menandakan bahwa *social distancing* dengan jarak aman diberi warna garis biru, sementara apabila jarak *bouding box* antar dua objek manusia kurang dari jarak yang telah ditentukan maka akan diberikan warna garis merah seperti yang nampak pada Gambar 5.



Gambar 5. Proses pengujian perhitungan jarak antar *bouding box*

5. KESIMPULAN

Dari hasil kegiatan penelitian ini dapat disimpulkan sebagai berikut:

1. Implementasi TensorFlow.js dapat digunakan dalam membantu proses deteksi objek manusia.
2. Aplikasi yang dirancang berbasis web sehingga ringan untuk dijalankan.
3. Pencahayaan tetap memegang peranan penting dalam perhitungan akurasi deteksi objek manusia. Pada kondisi pencahayaan ruangan yang kurang terang, akurasi deteksi dapat berkurang.

6. DAFTAR PUSTAKA

- [1] Liputan6. Dampak Corona, Pertumbuhan Ekonomi 170 Negara Diprediksi Negatif. Dapat diakses pada: <https://www.liputan6.com/bisnis/read/4225298/dampak-corona-pertumbuhan-ekonomi-170-negara-diprediksi-negatif>.
- [2] AyoBandung.com.2022. [online] Available at: <https://www.ayobandung.com/netizen/pr-79682821/masalah-dari-adanya-penerapan-social-distancing> [Accessed 15 Juli 2022].
- [3] Kompas.com, 2022. Social Distancing dan Hambatannya dalam Sosio-kultural Indonesia Halaman all - Kompas.com. [online] KOMPAS.com. Available at: <https://www.kompas.com/tren/read/2020/03/30/142329065/social-distancing-dan-hambatannya-dalam-sosio-kultural-indonesia?page=all> [Accessed 15 Juli 2022].
- [4] Verma, S. and Jain, P., 2022. COVID-19: Automatic Social Distancing Rule Voilation Detection using PP-Yolo & Tensorflow in OpenCV. 2022 International Conference for Advancement in Technology (ICONAT).
- [5] Degadwala, S., Vyas, D., Dave, H. and Mahajan, A., 2020. Visual Social Distance Alert System Using Computer Vision & Deep Learning. 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA).
- [6] Murch, R.2012, Project Management: Best Practices for IT Professionals, Ed.1, Prentice Hall, New Jersey.
- [7] Watkins, J. 2009, Agile Testing: How To Succeed in an Extreme Testing Environment, Cambirdge University Press, Cambridge