

Deteksi Kecurangan Ujian Pada Ruang Tertutup Menggunakan Algoritma YOLOv8

Exam Cheating Detection in Closed Room Using YOLOv8 Algorithm

Afandi Nur Aziz Thohari^{*1}, Muttabik Fathul Lathief², Liliek Triyono³, Kuwat Santoso⁴
^{1,2}Program Studi Teknik Informatika, Politeknik Negeri Semarang, Semarang, Indonesia
^{3,4}Program Studi Teknologi Rekayasa Komputer, Politeknik Negeri Semarang, Semarang, Indonesia

E-mail: ¹afandi@polines.ac.id, ²muttabik.lathief@gmail.com, ³lilieki.triyono@gmail.com,
⁴kuwatsantoso@polines.ac.id

Received: 23-03-2025 | Revised: 23-04-2025 | Accepted: 24-04-2025

Abstrak

Salah satu tantangan pengawasan ujian di ruang kelas atau ruangan tertutup adalah keterbatasan mata pengawas yang biasanya mengalami kelelahan jika terus mengawasi dalam waktu yang cukup lama. Akibatnya banyak tindakan kecurangan siswa yang luput dari pengawasan. Salah satu solusi mengatasi permasalahan tersebut adalah menggunakan sistem pengawasan cerdas yang dapat mendeteksi kecurangan ujian yang dilakukan oleh siswa. Beberapa penelitian mengenai sistem pengawasan cerdas telah banyak dilakukan. Namun belum menghasilkan akurasi yang optimal dalam mendeteksi kecurangan ujian. Tujuan dari penelitian ini adalah membangun model deteksi kecurangan ujian pada ruangan tertutup dengan performa yang optimal. Metode yang digunakan untuk deteksi objek adalah YOLO versi 8 (YOLOv8). Sebelum dilatih menggunakan algoritma YOLOv8, dilakukan pemilihan parameter untuk mendapatkan performa model yang optimal. Hasil pengujian menunjukkan bahwa model YOLOv8s menghasilkan performa yang paling optimal dengan nilai *precision*, *recall*, *IoU-Score* dan *mAP50* sebesar 0.952, 0.966, 0.8977, dan 0.984. Pengujian pada lingkungan kerja menunjukkan bahwa model YOLOv8s dapat mendeteksi kecurangan ujian secara *real-time* dengan jumlah *frame per second* sebesar 28 fps. Meskipun sudah mendapatkan performa yang cukup optimal. Namun performa sistem pengawasan kecurangan ujian ini masih dapat terus ditingkatkan. Selain itu pada penelitian ini memiliki keterbatasan yaitu hanya dapat mendeteksi kecurangan di lokasi tempat diambilnya dataset.

Kata kunci: kecurangan ujian, pengawasan, ruang tertutup, YOLOv8

Abstract

One of the challenges in monitoring exams in the classroom or closed room is the limited eye of the tire supervisor if he/she continues to monitor for a long time. Therefore, many behaviors of students cheating over the escape. One solution to overcome this problem is to implement an smart monitoring system that capable of detecting student exam cheating. A number of studies on smart monitoring systems have been conducted. However, the studies have not achieved optimal accuracy in identifying exam cheating. The objective of this research is to develop a closed-room exam cheating detection model that operates at its optimal performance. The method used to detect the object is YOLO version 8 (Yolov8). Before training using the YOLOv8 method, hyperparameter tuning was made to generate best model performance. The test results have shown that the Yolov8s model has created the best performance with the precision, recall, IoU-Score and mAP50 values of 0.952, 0.966, 0.8977 and 0.984. Testing in the working environment

shows that the YOLOv8s model can detect exam cheating in real time at a frame rate of 28 fps. Although it has achieved quite optimal performance. However, the performance of this exam cheating monitoring system can still be improved. Furthermore, this study has limitations, specifically that it can only detect cheating in the place where the dataset was collected.

Keywords: Exam Cheating, Monitoring, Closed Room, YOLOv8

1. PENDAHULUAN

Pada era digital saat ini, penggunaan teknologi kecerdasan buatan sudah semakin masif digunakan di segala bidang. Tidak hanya bidang kesehatan dan ekonomi, penggunaan kecerdasan buatan juga digunakan pada bidang pendidikan. Salah satu implementasi kecerdasan buatan di bidang pendidikan adalah deteksi kecurangan ujian pada ruangan tertutup. Pengawasan ujian biasanya dilakukan oleh manusia secara konsisten dan terus menerus. Kondisi ini bisa menjadi kurang efektif karena terbatasnya kemampuan manusia dalam melakukan pengawasan secara konsisten. Mata manusia biasanya mengalami kelelahan jika terus mengawasi dalam waktu yang cukup lama. Akibatnya banyak tindakan kecurangan siswa yang luput dari pengawasan.

Oleh karena itu teknologi kecerdasan buatan dapat menjadi alternatif untuk membantu pengawasan tindak kecurangan yang dilakukan oleh siswa. Teknis pengawasan menggunakan kamera yang terhubung ke perangkat komputer. Pada perangkat komputer tersebut telah ditanam model kecerdasan buatan yang mampu mendeteksi adanya perilaku kecurangan dalam bentuk pose gerakan. Tujuan dari penelitian ini adalah mendapatkan model deteksi kecurangan ujian dengan Tingkat presisi dan akurasi yang optimal. Kemudian *men-deploy* model tersebut ke dalam mini komputer agar dapat mendeteksi kecurangan siswa secara *real-time*.

Sebelumnya telah terdapat beberapa penelitian terkait deteksi kecurangan ujian menggunakan cabang teknologi kecerdasan buatan, yaitu visi komputer. Penelitian pertama membangun model kecerdasan buatan yang mampu mendeteksi pose objek dihasilkan melalui teknologi *deep learning* seperti *Convolutional Neural Network* (CNN), *Regional Convolution Neural Network* (RCNN), *You Only Look Once* (YOLO), dll. Salah satu penelitian deteksi kecurangan menggunakan CNN menghasilkan akurasi sebesar 80% [1]. Terdapat pula penelitian dengan algoritma CNN menggunakan arsitektur MobileNet menghasilkan akurasi 84,52% untuk mendeteksi kecurangan ujian melalui kamera [2].

Penelitian deteksi kecurangan ujian menggunakan algoritma YOLO versi 4 sudah pernah dilakukan [3]. Terdapat 6 jenis kecurangan yang dapat dideteksi yaitu menggunakan ponsel, melihat ke kiri melihat ke kanan, melihat ke belakang, melihat ke bawah, dan kerjasama. Sistem dapat bekerja dengan baik untuk dapat mendeteksi perilaku kecurangan ujian dengan nilai *Average Precision* keseluruhan adalah 85,67%. Kekurangan dari sistem adalah tidak dapat mendeteksi kecurangan secara *real-time*. Pengguna harus memasukan gambar yang ingin di deteksi. Penelitian berikutnya menggunakan algoritma YOLO versi 5 [4]. Jumlah perilaku kecurangan yang dideteksi ada 5 kelas yaitu normal, cheating, exchange paper, giving code, and looking friend. Hasil pengujian menunjukkan arsitektur YOLOv5s menghasilkan nilai mAP terbaik yaitu 0,555.

Selanjutnya penelitian menggunakan YOLOv3 dan OpenPose untuk deteksi potensi siswa melakukan kecurangan [5]. Hasil pengujian menunjukkan akurasi sebesar 96,2% dengan *Frame Rate* sebesar 25 FPS. Penggunaan algoritma YOLO masif dilakukan untuk mendeteksi objek untuk prediksi kecurangan ujian. Pada penelitian [6] menggunakan YOLOv7 untuk mendeteksi objek manusia, kemudian menggunakan arsitektur CNN-RNN yaitu inceptionV3 untuk melakukan klasifikasi tindakan kecurangan. Akurasi yang dihasilkan oleh model CNN-RNN adalah 90,67%. Algoritma YOLO dapat digunakan untuk mendeteksi pose tubuh manusia, kemudian kumpulan pose yang telah didapatkan dilatih menggunakan algoritma pembelajaran

mesin untuk klasifikasi *cheating* dan *no cheating* [7]. Hasil pengujian menunjukkan bahwa model yang dibangun memiliki akurasi lebih tinggi dari pada algoritma machine learning lain, yaitu sebesar 94%.

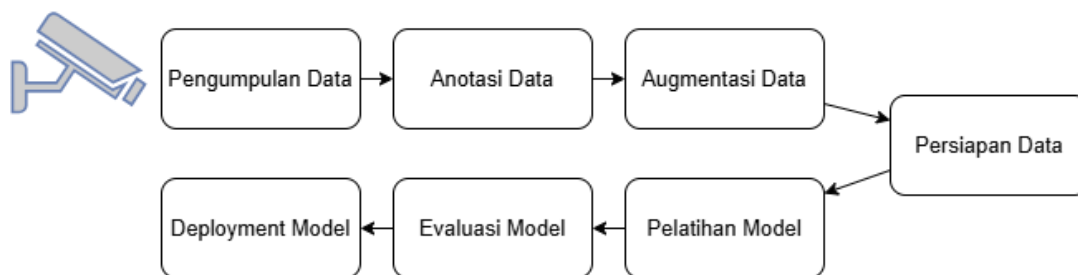
Pada penelitian ini akan dibangun model untuk deteksi kecurangan ujian melalui pose tubuh siswa. Pose yang didapat dikenali adalah menengok ke kanan, ke kiri, dan ke belakang. Terdapat dua kelas pada model yaitu *cheating* and *no cheating*. Dataset yang dipakai bersumber dari dokumentasi pribadi dengan mengambil foto siswa saat melakukan simulasi ujian pada ruangan tertutup. Perbandingan penelitian ini dengan penelitian sebelumnya adalah menggunakan YOLOv8 untuk mendeteksi objek. Pada algoritma YOLOv8 juga memiliki fitur deteksi pose tubuh sehingga dapat mendeteksi posisi tubuh siswa yang sedang menyontek. Selain fitur deteksi pose, alasan lain menggunakan YOLOv8 adalah lebih akurat dan cepat dalam mendeteksi objek dibandingkan YOLO versi sebelumnya [8],[9]. Kemudian YOLOv8 memiliki keadaan lebih stabil dan lebih bisa diandalkan dibandingkan versi yang lebih baru seperti YOLOv9 dan YOLOv10 [10].

Terdapat beberapa kontribusi yang ditawarkan pada penelitian ini, antara lain:

1. Menghasilkan model dengan tingkat akurasi yang lebih tinggi dibanding model dari penelitian lain untuk mendeteksi kecurangan ujian pada ruang tertutup.
2. Meningkatkan kecepatan *Frame Per Second* (FPS) pengawasan kecurangan ujian
3. Menghasilkan aplikasi web yang dapat mendeteksi kecurangan ujian dan menampilkan bukti kecurangan dengan cara meng-*capture* wajah siswa yang mencontek.

2. METODE PENELITIAN

Penelitian ini mengusulkan metode untuk mengatasi perilaku kecurangan pada ruang ujian menggunakan algoritma YOLOv8. Alasan menggunakan YOLOv8 adalah akurasi yang tinggi dalam mendeteksi objek, memiliki kecepatan deteksi objek yang tinggi, dan pada versi 8 mendukung segmentasi objek, dan klasifikasi gambar [11]. Sebelum dilatih menggunakan YOLO, dilakukan anotasi objek pada citra untuk memberi label pada objek yang terindikasi menyontek. Adapun langkah-langkah untuk pembuatan model deteksi menyontek ditunjukkan pada Gambar 1.



Gambar 1. Metode Penelitian yang Diusulkan

2.1 Pengumpulan Data

Tahap pertama yang dilakukan adalah pengumpulan data. Data yang dipakai adalah citra hasil pengawasan ujian pada ruangan tertutup. Jumlah citra yang berhasil dikumpulkan sebanyak 653 citra. Terdapat beberapa posisi atau sudut pengambilan gambar. Siswa melakukan pose atau *gesture* saat menyontek jawaban dari siswa lain. Pose yang biasa digunakan untuk menyontek adalah gerakan kepala pada saat menengok ke kanan atau kiri untuk melihat hasil pekerjaan siswa lain. Oleh karena itu pada penelitian ini proses anotasi dilakukan saat ada indikasi kecurangan pada saat ujian seperti menoleh ke kanan atau ke kiri melihat jawaban ujian. Lebih jelas mengenai

dataset pengawasan kecurangan ujian dapat dilihat pada Gambar 2.



Gambar 2. Gambar Dataset yang Digunakan

2.2 Anotasi Data

Setelah data gambar terkumpul, selanjutnya dilakukan labelling image. Terdapat dua label yang digunakan yaitu “*cheating*” dan “*no cheating*”. Label “*cheating*” akan ditambahkan apabila pose kepala siswa menoleh ke kanan, ke kiri, ke depan, dan ke belakang. Sedangkan label “*no cheating*” akan ditambahkan saat pose kepala siswa fokus menatap soal ujian atau lembar jawaban. Terdapat 653 gambar yang harus di anotasi. Dalam satu gambar terdapat empat objek siswa yang melakukan simulasi ujian. Setiap objek siswa harus di anotasi untuk mendapatkan akurasi model yang tinggi.

2.3 Augmentasi Data

Salah satu cara untuk menambah jumlah dataset adalah dengan melakukan augmentasi data. Tujuan augmentasi data adalah untuk menambah pola gambar dan mempertajam karakteristik pada objek [12]. Pada penelitian ini digunakan 5 teknik augmentasi gambar antara lain Hue and Saturation, Flip, Shear, Scale, dan Mixup. Masing-masing teknik augmentasi memiliki parameter nilai yang menentukan besar kecilnya perubahan pada gambar. Nilai dari masing-masing parameter augmentasi dapat dilihat pada Tabel 1.

Tabel 1. Teknik Augmentasi Data untuk Deteksi Kecurangan Ujian

Augmentasi Data	Perintah YOLOv8	Nilai
Hue and Saturation	hsv_h	0.015
	hsv_s	0.8
	hsv_v	0.1
Flip	flipud	0
	fliplr	0.5
Shear	shear	25
Scale	scale	0.7
Mixup	mixup	1

2.4 Persiapan Data

Sebelum di training, dataset akan dibagi menjadi data latihan, data validasi, dan data uji. Pada penelitian ini perbandingan data latih, validasi, dan uji adalah 80:10:10. Apabila jumlah total data adalah 653 gambar, maka jumlah data latih adalah 521 gambar, data validation dan data uji masing-masing 66 gambar. Jumlah data training 80% dari total dataset bertujuan agar mendapatkan model dengan akurasi yang tinggi. Sebab semakin banyak data yang dilatih, maka semakin banyak pula pola yang dipelajari oleh mesin.

2.5 Melatih Model

Tahap training menentukan performa model yang dihasilkan. Oleh karena itu sebelum di latih, terdapat penambahan parameter-parameter untuk mengoptimalkan proses pelatihan. Pada

penelitian ini digunakan beberapa parameter yang penting untuk meningkatkan akurasi model. Parameter yang digunakan pada penelitian ini adalah epoch, patience, batch size, learning rate. Nilai dari masing-masing parameter untuk pelatihan model pada penelitian ini ditunjukkan pada Tabel 2.

Tabel 2. Nilai Parameter untuk Meningkatkan Akurasi Model

Parameter	Perintah YOLOv8	Nilai
Epoch	epochs	25, 50, 100
patience	patience	50
Learning Rate	lr0	0.01
	lr1	0.01
Batch Size	batch	64

Penelitian ini menggunakan parameter-parameter yang berpengaruh untuk meningkatkan akurasi deteksi objek. Table 2 menunjukkan bahwa epoch yang digunakan ada 3 yaitu 25, 50, 100. Ketiga nilai epoch tersebut akan diuji untuk mengetahui nilai epoch ideal untuk menghasilkan performa model terbaik. Selanjutnya ada parameter patience yang merupakan jumlah epoch yang digunakan untuk mengamati pelatihan. Nilai patience terlalu kecil dapat mengakibatkan model menghentikan pelatihan terlalu dini. Sedangkan nilai patience terlalu besar mengakibatkan model akan terus berlatih. Berdasarkan beberapa referensi, diketahui nilai patience yang ideal adalah 50 [13].

Parameter lainnya yang mempengaruhi akurasi adalah batch size. Pada penelitian ini digunakan jumlah batch size 64 karena telah terbukti meningkatkan performa model [14], [15]. Parameter learning rate digunakan untuk mengatur seberapa besar langkah atau perubahan weights yang diambil model pada setiap iterasi selama proses backpropagation dalam rangka meminimalkan loss function [16]. Learning rate tidak boleh terlalu besar karena dapat menyebabkan loncatan yang terlalu besar dan membuat optimisasi meleset. Tidak boleh juga terlalu kecil karena dapat menyebabkan optimisasi menjadi lambat. Berdasarkan pengujian pada penelitian sebelumnya, nilai learning rate yang ideal adalah 0.01 [17].

2.6 Evaluasi Model

Evaluasi bertujuan untuk mengetahui performa model yang dihasilkan dari proses pelatihan. Terdapat beberapa metrik evaluasi untuk mengukur performa model yang dihasilkan dari proses pelatihan menggunakan algoritma YOLO.

a) Precision

Precision adalah rasio antara jumlah *True Positives* (prediksi yang benar) dengan total *Positive Predictions* (jumlah prediksi positif yang benar dan salah). *Precision* menunjukkan seberapa akurat model dalam mendeteksi objek dibandingkan dengan jumlah total deteksi yang dilakukan. Perhitungan nilai *precision* diperoleh melalui persamaan 1.

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \quad (1)$$

b) Recall

Recall adalah rasio antara jumlah *True Positives* dengan total *Actual Positives* (jumlah semua objek yang sebenarnya ada di gambar). *Recall* menggambarkan kemampuan model untuk menemukan semua objek yang ada di gambar. Perhitungan nilai *Recall* diperoleh melalui persamaan 2.

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \quad (2)$$

c) *Intersection over Union (IoU)*

IoU adalah metrik yang mengukur seberapa baik prediksi *bounding box* terhadap *ground truth bounding box*. IoU dihitung dengan membagi area intersection antara prediksi dan ground truth dengan area union dari keduanya. IoU biasanya digunakan untuk menentukan apakah prediksi dianggap sebagai *True Positive* (jika IoU melebihi ambang batas, misalnya 0.5). Perhitungan IoU ditunjukkan pada persamaan 3.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (3)$$

d) *Mean Average Precision (mAP)*

Mean Average Precision (mAP) adalah metrik kinerja keseluruhan untuk deteksi objek pada algoritma YOLO. mAP menggabungkan precision dan recall pada berbagai ambang batas IoU. Proses perhitungannya mAP ditunjukkan pada persamaan 4.

$$mAP = \frac{1}{c} \sum_{i=1}^c AP_i \quad (4)$$

dimana, c adalah jumlah kelas dan AP_i adalah nilai *Average Precision* untuk kelas ke- i

2.7 Deployment Model

Setelah mendapatkan model dengan performa terbaik, selanjutnya dilakukan proses deployment. Pada tahap ini dibangun aplikasi web yang dapat mendeteksi kecurangan pada saat ujian. Ketika siswa terdeteksi melakukan kecurangan, maka sistem akan meng-capture wajah mereka. Sehingga pengawas dapat melihat secara jelas wajah siswa yang terindikasi melakukan kecurangan.

3. HASIL DAN PEMBAHASAN

Setelah melalui tahap pelatihan, langkah terakhir adalah mengevaluasi model. Algoritma YOLO memiliki beberapa varian model yaitu n, s, m, l, and x [18]. YOLO dengan varian n dan s memiliki arsitektur dengan kompleksitasnya rendah. Model YOLO varian n dan s memiliki ukuran yang ringan dan kecil. Waktu pelatihan model varian n dan s sangat cepat, namun akurasi yang dihasilkan rendah. Sedangkan YOLO varian m, l, dan x memiliki arsitektur yang kompleks dan kecepatan training yang lama. Keunggulan dari model varian m, l, dan x adalah memiliki akurasi tinggi. Pada penelitian ini digunakan semua varian model YOLOv8 untuk mendapatkan performa model yang terbaik dalam mendeteksi kecurangan ujian.

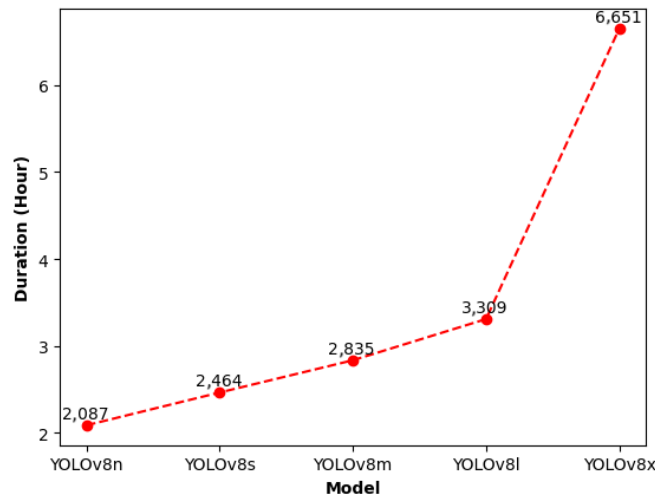
Parameter yang dipakai pada saat pelatihan mengacu pada Tabel 2. Berdasarkan pengujian diperoleh hasil bahwa jumlah epoch 50 memiliki performa yang paling baik dalam mendeteksi kecurangan ujian. Oleh karena itu digunakan epoch dengan jumlah sebesar 50. Hasil pengujian dari masing-masing model ditunjukkan pada Tabel 3.

Table 3. Hasil Evaluasi dari Model YOLOv8 Model untuk Deteksi Kecurangan Ujian

Model	Precision	Recall	IoU-Score	mAP50
-------	-----------	--------	-----------	-------

YOLOv8n	0.942	0.96	0.8912	0.974
YOLOv8s	0.952	0.966	0.8977	0.984
YOLOv8m	0.934	0.943	0.8858	0.979
YOLOv8l	0.956	0.929	0.8876	0.973
YOLOv8x	0.952	0.957	0.8793	0.981

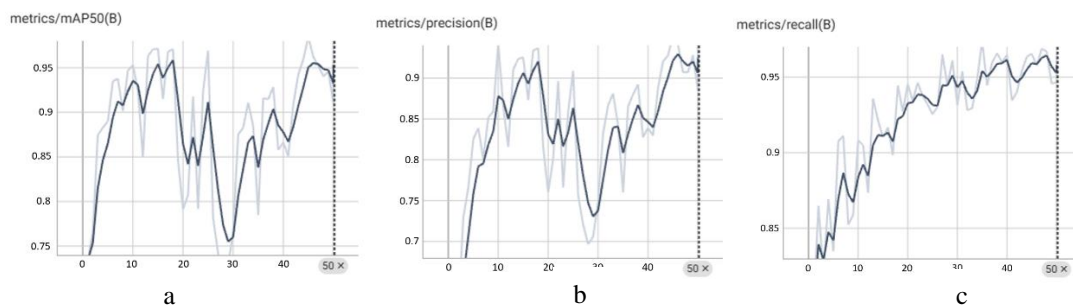
Berdasarkan hasil pengujian, diketahui bahwa model YOLOv8s memiliki performa yang paling baik dengan nilai Recall sebesar 0.966 dan mAP50 sebesar 0.984. Proses pelatihan menggunakan personal komputer dengan spesifikasi Intel Core i7 13th Generation 2.10 Ghz, 16 Gb of RAM, and using NVIDIA GeForce RTX 3050 OEM 16 Gb Memory Size. Proses komputasi dilakukan pada GPU agar lebih cepat [19]. Hasil pengujian waktu pelatihan menggunakan GPU ditunjukkan pada Gambar 3.



Gambar 3. Durasi Pelatihan Model YOLOv8

Proses komputasi algoritma YOLO sangat kompleks sehingga membutuhkan perangkat komputer dengan spesifikasi *high end*. Gambar 3 menunjukkan durasi pembuatan model YOLOv8 dengan jumlah epoch sebanyak 50. Varian YOLOv8 tipe n, s, dan m memiliki waktu training yang lebih cepat karena memiliki arsitektur sederhana. Sedangkan YOLOv8 varian l dan x membutuhkan waktu pelatihan yang cukup lama karena memiliki arsitektur yang kompleks. YOLOv8 varian x memiliki waktu training yang paling lama karena memiliki arsitektur yang paling kompleks diantara varian model YOLO yang lain dan ukuran model yang besar yaitu 2x lipat ukuran model YOLOv8l.

Model YOLOv8s memiliki performa terbaik untuk mendeteksi posisi menyontek siswa di ruang ujian. Pada saat dilakukan training dengan jumlah epoch sebanyak 25, akurasi yang diperoleh tidak terlalu tinggi. Hal ini dikarenakan model yang dihasilkan tidak mencapai konvergensi. Sedangkan pada saat dilatih dengan jumlah epoch sebanyak 100 terjadi overfitting model. Jumlah epoch yang ideal dan menghasilkan akurasi paling tinggi adalah 50 epoch. Hasil pelatihan model dengan jumlah epoch sebesar 50 ditunjukkan pada Grafik Gambar 4.



Gambar 4. Grafik hasil Penelitian (a) mAP50; (b) Precision; (c) Recall

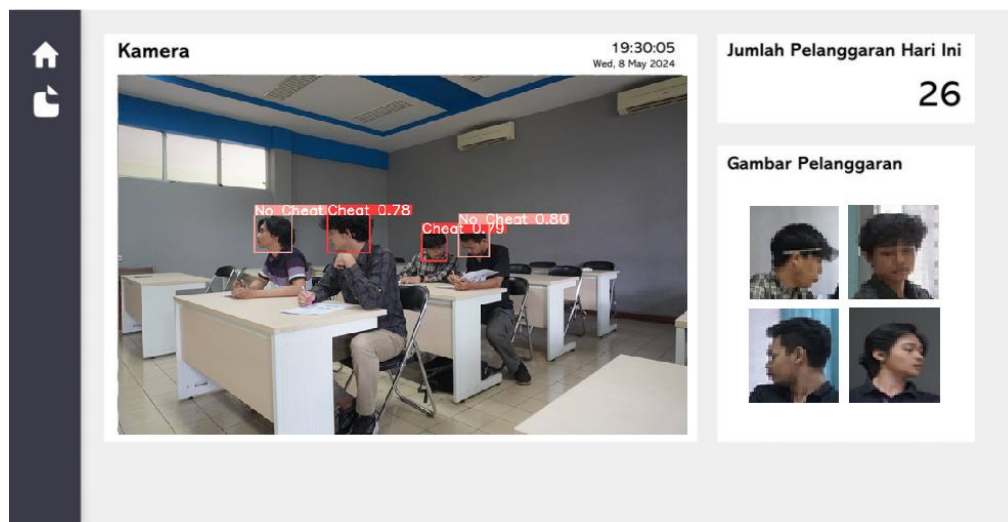
Grafik pada Gambar 4 menunjukkan nilai mAP50, precision, dan recall dari setiap epoch yang diambil dari Tensorboard. Perubahan nilai mAP50, precision, dan recall terjadi karena YOLO akan selalu mengupdate nilai bobot setiap kali epoch bertambah. Nilai metrik evaluasi akan terus bertambah seiring dengan bertambahnya jumlah epoch. Hal ini ditunjukkan pada Grafik Gambar 4 (b), nilai precision terus meningkat saat nilai epoch bertambah. Hasil pengujian model menggunakan citra uji ditunjukkan pada Gambar 5.



Gambar 5. Hasil Pengujian Model YOLOv8 untuk Ujian di Ruangan Tertutup

Berdasarkan hasil pengujian Gambar 5, diketahui bahwa model dapat mendeteksi pose siswa saat menyontek di ujian. Model dapat mendeteksi gerakan wajah siswa saat menoleh ke kanan, kiri, depan, dan belakang untuk mencontok hasil pekerjaan temannya. Pengujian juga dilakukan di beberapa angle atau sudut ruangan untuk memastikan model dapat mendeteksi tindakan kecurangan yang dilakukan siswa. Pengujian secara langsung yang dilakukan mendapatkan nilai *frame per second* sebesar 28 fps. Nilai ini sudah cukup untuk mendapatkan hasil video yang *smooth* untuk proses pengawasan.

Model yang telah diuji dan memiliki performa terbaik akan di *deploy* pada aplikasi web. Tujuan dilakukan *deployment* adalah untuk memudahkan pengawas dalam mengawasi siswa saat mengerjakan ujian. Model YOLO akan ditanamkan pada web kemudian saat program dijalankan sistem sudah dapat meng-*capture* aktivitas yang dilakukan siswa. Apabila ada siswa yang terindikasi menyontek, maka sistem akan meng-*capture* wajah dari siswa dan menampilkan nya pada sidebar website. Kemudian website juga dapat mencatat jumlah pelanggaran yang dilakukan oleh siswa selama mengerjakan ujian. Adapun tampilan dari website deteksi kecurangan ujian ditunjukkan pada Gambar 6.



Gambar 6. Tampilan Website Hasil *Deployment* Model YOLOv8

4. KESIMPULAN

Penelitian ini bertujuan untuk membangun sistem pengawasan pintar yang dapat mendeteksi kecurangan ujian pada ruangan tertutup. Deteksi kecurangan ujian menggunakan algoritma YOLOv8 dengan penambahan *hyperparameter*. Berdasarkan hasil pengujian menggunakan 5 varian YOLO, didapatkan kesimpulan bahwa model YOLOv8s memiliki performa yang paling optimal. Hasil pengujian menunjukkan model YOLOv8s memiliki nilai precision, recall, IoU-Score dan mAP50 sebesar 0.952, 0.966, 0.8977, dan 0.984. Kemudian hasil pengujian pada lingkungan kerja menunjukkan sistem dapat mendeteksi secara *real-time* dengan kecepatan video sebesar 28 fps. Hal ini sudah sesuai dengan standar pengawasan video. Model YOLOv8s telah berhasil di-*deploy* pada website dan berhasil mengidentifikasi pelanggaran yang dilakukan oleh siswa di ruang ujian. Meskipun memiliki performa yang baik, namun sistem pengawasan yang dibangun memiliki keterbatasan yaitu hanya mampu mendeteksi kecurangan di lokasi tempat dataset tersebut diambil. Pengembangan lebih lanjut terkait sistem pengawasan kecurangan ujian adalah meningkatkan performa model agar lebih akurat. Pengembangan lainnya adalah menambah jenis pelanggaran yang dilakukan oleh siswa, seperti pelanggaran menggunakan ponsel atau catatan, pelanggaran menunjukkan jawaban ke peserta lain, dll.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada Pusat Penelitian dan Pengabdian Kepada Masyarakat (P3M) Politeknik Negeri Semarang yang telah memberikan dana DIPA Penelitian Terapan Kompetitif, Sehingga penelitian ini dapat berjalan dengan lancar.

DAFTAR PUSTAKA

- [1] M. P. Pangestu, S. Wiyono, and D. I. Af'idah, "Platform Ujian Online Berbasis Pendeteksi Gerakan Kecurangan Menggunakan Kamera," *Infomatek*, vol. 26, no. 1, pp. 55–62, 2024, doi: 10.23969/infomatek.v26i1.11208.
- [2] F. A. Hariz, I. N. Yulita, and I. Suryana, "Human Activity Recognition Berdasarkan Tangkapan Webcam Menggunakan Metode Convolutional Neural Network (CNN)

- Dengan Arsitektur MobileNet,” *J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 4, pp. 103–115, 2022, doi: 10.30630/jitsi.3.4.97.
- [3] T. Nur, Huzaeni, and M. Khadafi, “Implementasi Metode Object Detection Dengan Algoritma Yolo (You Only Look Once) Untuk,” *J. Teknol. Rekayasa Inf. dan Komput.*, vol. 6, no. 2, pp. 28–33, 2023.
- [4] F. Bimantoro, I. G. Pasek, S. Wijaya, and M. R. Aohana, “Pendeteksian Kecurangan Ujian Melalui CCTV Menggunakan Algoritma YOLOv5,” in *Prosiding Seminar Nasional Teknologi dan Sains Tahun 2024*, Kediri: Universitas Nusantara PGRI Kediri, 2024, pp. 109–117.
- [5] Z. Wan, X. Li, B. Xia, and Z. Luo, “Recognition of Cheating Behavior in Examination Room Based on Deep Learning,” *Proc. - 2021 Int. Conf. Comput. Eng. Appl. ICCEA 2021*, pp. 204–208, 2021, doi: 10.1109/ICCEA53728.2021.00048.
- [6] A. Zaffar, M. Jawad, and M. Shabbir, “A Novel CNN-RNN Model for E-Cheating Detection Based on Video Surveillance,” *UW J. C. Sci.*, vol. 5, no. 1, pp. 1–13, 2023, [Online]. Available: <https://uwjcs.org.pk/index.php/ojs/article/view/64>
- [7] N. Tran, M. Nguyen, T. Le, T. Huynh, T. Nguyen, and T. Nguyen, “Exploring the potential of skeleton and machine learning in classroom cheating detection,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 32, no. 3, pp. 1533–1544, 2023, doi: 10.11591/IJEECS.V32.I3.PP1533-1544.
- [8] R. S. Wijaya, A. Wibisana, and E. R. Jamzuri, “Comparative Study of YOLOv5, YOLOv7 and YOLOv8 for Robust Outdoor Detection,” *J. Appl. Electr. Eng.*, vol. 8, no. 1, pp. 37–43, 2024.
- [9] N. Affes, J. Ktari, N. Ben Amor, T. Frikha, and H. Hamam, “Comparison of YOLOV5, YOLOV6, YOLOV7 and YOLOV8 for Intelligent Video Surveillance,” *J. Inf. Assur. Secur.*, vol. 18, no. 5, p. 147, 2023.
- [10] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, “A Review on YOLOv8 and Its Advancements,” I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, Eds., Singapore: Springer Nature Singapore, 2024, pp. 529–545.
- [11] H. Li, J. Huang, Z. Gu, D. He, J. Huang, and C. Wang, “Positioning of mango picking point using an improved YOLOv8 architecture with object detection and instance segmentation,” *Biosyst. Eng.*, vol. 247, no. April, pp. 202–220, 2024, doi: 10.1016/j.biosystemseng.2024.09.015.
- [12] P. Kaur, B. S. Khehra, and E. B. S. Mavi, “Data Augmentation for Object Detection: A Review,” in *Midwest Symposium on Circuits and Systems (MWSCAS)*, Lansing, MI, USA: IEEE, 2021, pp. 537–543. doi: 10.1109/MWSCAS47672.2021.9531849.
- [13] Suvarna Patil, Soham Waghule, Siddhesh Waje, Prasad Pawar, and Shreyash Domb, “Efficient Object Detection with YOLO: A Comprehensive Guide,” *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 5, pp. 519–531, 2024, doi: 10.48175/ijarsct-18483.
- [14] L. Ramos, E. Casas, E. Bendek, C. Romero, and F. Rivas-Echeverría, “Hyperparameter optimization of YOLOv8 for smoke and wildfire detection: Implications for agricultural and environmental safety,” *Artif. Intell. Agric.*, vol. 12, pp. 109–126, 2024, doi: <https://doi.org/10.1016/j.aiia.2024.05.003>.
- [15] Z. S. Hidayat, Y. A. Wijaya, and R. Kurniawan, “Optimizing YOLOv8 for Autonomous Driving: Batch Size for Best Mean Average Precision (mAP),” *J. Tek. Inform.*, vol. 5, no. 4, pp. 1147–1153, 2024, doi: <https://doi.org/10.52436/1.jutif.2024.5.4.1626>.
- [16] Z. Lin, S. Zhang, Y. Zhou, H. Wang, and S. Wang, “Learning rate burst for superior SGDM and AdamW integration,” *J. Intell. Fuzzy Syst.*, vol. Preprint, pp. 1–11, 2024, doi: 10.3233/JIFS-239157.
- [17] Rashmi, U. Ghose, and M. Gupta, “Comparative Design Analysis of Optimized Learning Rate for Convolutional Neural Network,” *Lecture Notes on Data Engineering and Communications Technologies*. Springer Singapore, Singapore, pp. 339–352, 2021.

-
- [18] M. Hussain, "YOLOv1 to v8: Unveiling Each Variant-A Comprehensive Review of YOLO," *IEEE Access*, vol. 12, no. February, pp. 42816–42833, 2024, doi: 10.1109/ACCESS.2024.3378568.
- [19] A. Jayasimhan and P. Pabitha, "A comparison between CPU and GPU for image classification using Convolutional Neural Networks," in *International Conference on Communication, Computing and Internet of Things (IC3IoT)*, Chennai, India, 2022, pp. 1–4. doi: 10.1109/IC3IoT53935.2022.9767990.