

PENERAPAN DEEP LEARNING DALAM MENDETEKSI FOKUS SISWA DI MTS AL-MUKHTARIYAH MANDE

Achmad Rikza fauzain, Castaka Agus Sugianto

Teknik Informatika, Politeknik TEDC Bandung

Jalan Pasantren KM.2 Cibabat Cimahi Utara, Cimahi, Provinsi Jawa Barat, 40513

rikza41@gmail.com

ABSTRAK

Proses pembelajaran sering kali terkendala oleh kesulitan siswa dalam mempertahankan fokus, terutama di kelas dengan jumlah siswa yang banyak, sehingga observasi manual oleh guru di MTs Al-Mukhtariyah Mande menjadi tidak efisien dan subjektif. Permasalahan ini mendorong perlunya solusi teknologi untuk pemantauan yang objektif. Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi model deep learning berbasis algoritma Ultralytics YOLOv11 untuk mendeteksi tingkat fokus siswa secara real-time melalui analisis multi-kelas terhadap perilaku visual. Metode penelitian yang digunakan adalah metode eksperimental, yang mencakup pengumpulan data melalui perekaman video di kelas, pelabelan manual terhadap 546 frame gambar ke dalam lima kelas perilaku (membaca, memperhatikan, menoleh, menulis, dan menunduk), pelatihan model dengan strategi fine-tuning dari model pre-trained COCO, dan evaluasi performa. Hasil evaluasi menunjukkan model mencapai mean Average Precision (mAP) pada IoU 50% (mAP50) sebesar 0,635 secara keseluruhan. Performa model sangat baik dalam mendeteksi kelas 'menulis' (mAP50: 0,776) dan 'menunduk' (mAP50: 0,813), namun menunjukkan tantangan pada kelas 'menoleh' (mAP50: 0,483), yang mengindikasikan kesulitan dalam membedakan pose yang mirip secara visual. Model yang dikembangkan berhasil diimplementasikan dalam aplikasi web berbasis Flask untuk pemantauan real-time.

Kata kunci : Deep Learning, YOLOv11, Deteksi Fokus, Siswa, Pendidikan

1. PENDAHULUAN

Kemajuan teknologi di bidang kecerdasan buatan, khususnya *deep learning*, telah membuka peluang besar untuk meningkatkan kualitas pembelajaran. Penerapan AI mendorong inovasi dalam proses belajar-mengajar serta menciptakan lingkungan belajar yang lebih interaktif dan adaptif [1]. Salah satu aspek krusial dalam proses pembelajaran adalah mempertahankan fokus siswa, karena tingkat fokus sangat memengaruhi efektivitas penerimaan materi dan hasil belajar [2]. Fokus siswa didefinisikan sebagai tingkat keterlibatan perhatian siswa terhadap materi yang disampaikan. Indikator visual siswa yang fokus antara lain kontak mata yang tertuju ke guru atau papan tulis, postur tubuh yang tegap atau condong ke depan, serta gerakan tangan aktif seperti mengangkat tangan untuk bertanya sebagai tanda fokus tinggi [3]. Sebaliknya, tanda kurang fokus meliputi mengalihkan pandangan, menunduk tanpa membaca/menulis, atau terlibat dalam aktivitas tidak relevan seperti bermain ponsel [3].

Kemampuan memantau fokus siswa secara otomatis menjadi sangat penting untuk meningkatkan efisiensi pembelajaran. Pendekatan berbasis AI dapat mengidentifikasi tingkat perhatian siswa dengan menganalisis ekspresi wajah, arah pandangan, dan postur tubuh [1]. Dengan data tersebut, guru dapat memperoleh umpan balik real-time tentang keterlibatan siswa dan menyesuaikan strategi pengajaran secara dinamis [2]. Sebaliknya, metode konvensional bergantung pada observasi langsung guru yang bersifat subjektif, memakan waktu, dan sulit

diterapkan pada kelas besar. Pendekatan manual berpotensi menimbulkan bias penilaian serta menghambat intervensi tepat waktu bagi siswa dengan fokus menurun [4].

MTs Al-Mukhtariyah Mande, sebagai salah satu institusi pendidikan, juga menghadapi permasalahan serupa. Guru di sekolah tersebut umumnya mengandalkan pengamatan manual untuk menilai konsentrasi siswa, yang cenderung subjektif dan tidak efisien, terutama pada kelas dengan banyak siswa. Diperlukan solusi berbasis teknologi yang dapat secara otomatis mendeteksi fokus siswa secara *real-time* dengan memanfaatkan kecerdasan buatan. Solusi ini diharapkan membantu guru memperoleh umpan balik segera mengenai tingkat fokus siswa dan menyesuaikan metode pengajaran jika diperlukan demi meningkatkan kualitas pembelajaran [5].

Meskipun beberapa penelitian terdahulu telah menunjukkan potensi kecerdasan buatan untuk menilai keterlibatan siswa, penerapan spesifik algoritma YOLOv11 untuk mendeteksi fokus siswa secara *real-time* masih sangat terbatas. Oleh karena itu, penelitian ini bertujuan mengembangkan model deteksi perilaku siswa berbasis YOLOv11 yang mampu melakukan klasifikasi *multi-class* pada aktivitas siswa di kelas. Model ini dirancang untuk mendeteksi lima kategori perilaku siswa yang mencerminkan tingkat fokus, yaitu membaca, memperhatikan, menoleh, menulis, dan menunduk. Dengan model ini, diharapkan guru dapat lebih mudah mengidentifikasi siswa yang membutuhkan perhatian lebih dan menyesuaikan metode pengajaran berdasarkan tingkat fokus kelas secara objektif. Hasil penelitian ini diharapkan dapat

menjadi referensi bagi pengembangan sistem *smart classroom* berbasis kecerdasan buatan yang lebih adaptif dan efektif di masa depan.

2. TINJAUAN PUSTAKA

2.1. Deep learning

Deep learning adalah cabang dari pembelajaran mesin (*machine learning*) yang menggunakan jaringan saraf tiruan dengan banyak lapisan (*deep neural networks*) untuk memproses data dan mengekstrak pola kompleks secara otomatis [6]. Dalam bidang *computer vision*, model *deep learning* (khususnya *Convolutional Neural Network* atau CNN) mampu mengekstraksi fitur citra secara otomatis terhadap variasi kondisi pencahayaan maupun pose. Bahkan, akurasi dapat melampaui kemampuan manusia dalam tugas-tugas pengenalan pola tertentu.

Tidak hanya pada bidang pendidikan, *deep learning* yang berbasis CNN juga telah menunjukkan performa sangat tinggi di bidang pertanian. Salah satu Penelitian berhasil menggunakan EfficientNetB0 untuk klasifikasi penyakit daun anggur dengan akurasi 99,36%, serta interpretasi hasil deteksi menggunakan Grad-CAM *visualization* untuk meningkatkan kepercayaan pengguna terhadap model [7]. Temuan ini menguatkan bahwa arsitektur CNN modern sangat potensial untuk berbagai aplikasi klasifikasi citra, termasuk analisis perilaku siswa di kelas.

2.2. You Only Look Once (YOLO)

You Only Look Once (YOLO) adalah algoritma deteksi objek satu tahap (*single-stage*) yang memproses seluruh gambar dalam satu kali *forward pass*, memprediksi koordinat *bounding box* dan probabilitas kelas secara simultan [8].

YOLO terus berkembang menjadi versi-versi baru. Tiap iterasi (YOLOv2 hingga YOLOv10) menambahkan fitur teknik terkini seperti *batch normalization*, *anchor-free detection*, dan mekanisme *attention*, serta peningkatan arsitektur *backbone* dan *neck* untuk akurasi dan kecepatan yang lebih baik [8].

YOLOv11, sebagai iterasi terbaru yang dikenalkan pada konferensi YOLO Vision 2024, memperkenalkan modul *arsitektural* seperti blok C3k2 (*Cross Stage Partial kernel size 2*), SPPF (*Spatial Pyramid Pooling – Fast*), dan C2PSA (*Convolutional block with Parallel Spatial Attention*) untuk peningkatan ekstraksi fitur [9]. Desain YOLOv11 yang inovatif menjaga jumlah parameter tetap ringan sambil meningkatkan akurasi dan kecepatan deteksi secara signifikan.

2.3. Python

Python adalah bahasa pemrograman tingkat tinggi yang sering digunakan dalam pengembangan aplikasi kecerdasan buatan (AI) dan *deep learning*. *Python* memiliki sintak yang sederhana dan mudah dipahami, menjadikannya pilihan utama dalam penelitian dan pengembangan AI [10]. Selain itu, *Python* memiliki ekosistem pustaka yang luas seperti

TensorFlow, *PyTorch*, dan *OpenCV*, yang sangat mendukung pengolahan citra dan analisis data [11].

2.4. OpenCV

OpenCV (Open Source Computer Vision Library) adalah pustaka sumber terbuka yang digunakan untuk pengolahan citra dan visi komputer. *OpenCV* menyediakan berbagai fungsi untuk manipulasi gambar, deteksi wajah, pengenalan ekspresi, serta analisis citra secara *real-time* [12].

2.5. Roboflow

Roboflow adalah sebuah platform berbasis web yang dirancang khusus untuk mempermudah dan mempercepat proses manajemen data *computer vision*, khususnya untuk proyek-proyek *machine learning* seperti deteksi objek (misalnya, dengan YOLO), segmentasi, dan klasifikasi gambar [13]. Dalam penggunaannya, *Roboflow* menyediakan fitur untuk mengelola, mengonversi, serta membagi data set ke dalam beberapa bagian, seperti data pelatihan, validasi, dan pengujian. Pembagian ini umumnya dilakukan secara acak untuk memastikan distribusi yang seimbang pada setiap kelas perilaku di dalam dataset. Praktik pembagian dataset ini menjadi salah satu standar dalam pengembangan model *machine learning*, karena dapat membantu memastikan kemampuan generalisasi model selama proses pelatihan serta mencegah terjadinya *overfitting* pada data pelatihan [11].

2.6. Flask

Flask adalah *framework* berbasis *Python* yang digunakan untuk membangun aplikasi web dan API (*Application Programming Interface*). *Flask* memiliki desain yang ringan dan fleksibel, sehingga cocok untuk mengembangkan sistem berbasis AI yang memerlukan komunikasi antara *backend* dan *frontend* secara *real-time* [14].

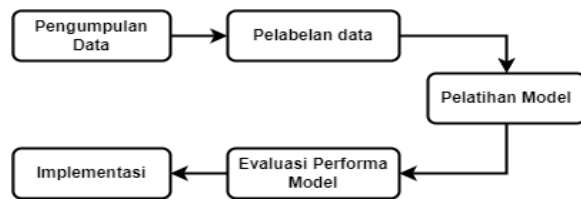
2.7. Studi Literatur

Sejumlah penelitian memanfaatkan *deep learning* untuk memantau perilaku atau emosi siswa dalam konteks pembelajaran. Misalnya, penggunaan algoritma YOLOv4 untuk mengklasifikasikan gambar siswa menjadi kelas “*engaged*” (terlibat) dan “*not engaged*” (tidak terlibat), dengan data kamera kelas sebagai sumber citra [15]. Penelitian lain di Indonesia, menerapkan *Face Emotion Recognition (FER)* menggunakan YOLOv8 untuk mendeteksi tingkat antusiasme siswa pada kelas daring [16].

Studi-studi ini menunjukkan tren pemanfaatan YOLO dalam mengklasifikasi perilaku atau ekspresi peserta didik. Semua literatur ini menegaskan bahwa teknologi *computer vision*, khususnya varian YOLO, sangat potensial untuk aplikasi pemantauan kelas yang bertujuan meningkatkan keterlibatan dan hasil belajar siswa.

3. METODE PENELITIAN

Penelitian ini mengadopsi metode eksperimental sebagai pendekatan utama untuk mencapai tujuan pengembangan dan evaluasi model deteksi *multi-class* perilaku siswa berbasis YOLOv11.



Gambar 1. Alur penelitian

Seluruh proses penelitian terdiri dari beberapa tahapan utama seperti yang terlihat pada gambar 1, yaitu pengumpulan data, pelabelan data, pelatihan

model, evaluasi performa model, dan implementasi model.

3.1. Pengumpulan Data

Pengumpulan data pada penelitian ini diperoleh dari video pembelajaran yang direkam di kelas MTs Al-Mukhtariyah Mande. Perekaman dilakukan selama beberapa sesi belajar mengajar tanpa adanya intervensi khusus dari peneliti, agar data yang terkumpul mencerminkan kondisi kelas yang nyata. Setelah data video terkumpul, kemudian *frame* di ekstrak secara berkala untuk dijadikan *dataset*.

3.2. Pelabelan Data

Selanjutnya adalah proses pelabelan atau anotasi, setiap objek siswa diberi label perilaku sesuai kategori yang telah ditetapkan, yaitu: membaca, memperhatikan, menoleh, menulis, dan menunduk. Kriteria pelabelan dibuat berdasarkan ciri-ciri visual utama yang dapat dilihat pada tabel 1.

Tabel 1. Kriteria pelabelan

Kriteria	Ciri-ciri
Membaca	Siswa menunduk dengan pandangan tertuju pada buku atau kertas
Memperhatikan	Siswa menatap ke arah guru atau papan tulis dengan ekspresi wajah fokus
Menoleh	Kepala atau mata siswa terlihat mengarah ke samping, menjauh dari pusat perhatian pembelajaran
Menulis	Siswa terlihat memegang alat tulis dan melakukan gerakan menulis di atas meja
Menunduk	Kepala siswa menunduk tanpa aktivitas membaca atau menulis yang jelas

Data set hasil anotasi atau pelabelan, kemudian dibagi menjadi tiga bagian: data latih sebanyak 70%, data validasi sebanyak 20%, dan data uji sebanyak 10%. Pembagian ini dilakukan secara acak untuk memastikan distribusi yang seimbang pada setiap kelas perilaku.

3.3. Pelatihan Model

Proses pelatihan model dilakukan menggunakan YOLOv11 dengan lingkungan pemrograman *Python* dan *library PyTorch*. Model YOLOv11 yang digunakan merupakan model *pre-trained* dari data set COCO yang selanjutnya di *fine-tune* menggunakan data perilaku siswa agar model lebih adaptif pada karakteristik data lokal [8], [17].

3.4. Evaluasi Performa Model

Selanjutnya dilakukan evaluasi performa model untuk mengukur kemampuan generalisasi terhadap data yang belum terlihat. Metrik yang digunakan meliputi *Accuracy*, *Precision*, *Recall*, *F1-score*, dan *mean Average Precision* (mAP) untuk setiap kelas perilaku siswa. Konsep dasar evaluasi melibatkan *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) [11].

Berdasarkan konsep tersebut, metrik didefinisikan sebagai berikut:

- a. *Accuracy*: Proporsi total *True Positives*.

$$Accuracy = \frac{TP}{TP+FN}$$

(1)

- b. *Precision*: Proporsi prediksi *True Positives* dari semua prediksi positif model.

$$Precision = \frac{TP}{TP + FP}$$

(2)

- c. *Recall*: Proporsi kasus positif aktual yang teridentifikasi model.

$$Recall = \frac{TP}{TP+FN}$$

(3)

- d. *F1-score*: Rata-rata harmonis *Precision* dan *Recall*.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

(4)

- e. *Mean Average Precision* (mAP): Metrik utama deteksi objek seperti YOLOv11. mAP menghitung rata-rata *Average Precision* (AP) untuk setiap kelas, di mana AP adalah area di bawah kurva *Precision-Recall*.

Selain metrik numerik, *confusion matrix* dihasilkan untuk visualisasi dan analisis kesalahan klasifikasi [11]. Hasil deteksi juga diamati secara visual pada *frame-frame* video guna memastikan kemampuan model membedakan perilaku di kondisi nyata.

3.5. Implementasi

Model deteksi perilaku siswa yang telah terlatih diimplementasikan dalam sebuah aplikasi web interaktif. Aplikasi ini dikembangkan menggunakan *framework Python Flask*, yang berfungsi sebagai

antarmuka pengguna untuk memanfaatkan model YOLOv11 secara praktis. Implementasi web ini bertujuan untuk menyajikan demonstrasi fungsional model serta potensi aplikasinya sebagai alat bantu bagi pengajar dalam memantau dan mengevaluasi partisipasi siswa di kelas.

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Data di ambil dari perekaman video kegiatan belajar mengajar di MTs Al-Mukhtariyah Mande pada beberapa kelas dan mata pelajaran berbeda. Proses perekaman berlangsung secara natural, tanpa ada pengarahan khusus, agar perilaku yang terekam benar-benar merepresentasikan suasana kelas yang otentik. Setelah seluruh sesi perekaman selesai, peneliti melakukan ekstraksi *frame* dari video menggunakan perangkat lunak *OpenCV*. Dari total video yang direkam, diperoleh 546 *frame* berkualitas baik yang memperlihatkan wajah siswa secara jelas seperti yang terlihat pada gambar 2.



Gambar 2. Frame berkualitas baik

Frame yang buram, gelap, atau wajah tertutup objek seperti pada gambar 3 dieliminasi dari data set.



Gambar 3. Frame berkualitas buruk

4.2. Pelabelan Data

Setelah data *frame* berhasil dikumpulkan, tahap selanjutnya adalah proses pelabelan (anotasi) pada setiap wajah siswa yang tampak di *frame*. Pelabelan data dilakukan secara manual menggunakan platform *Roboflow*, yang memudahkan proses penandaan *bounding box* pada objek wajah siswa seperti yang terlihat pada gambar 4.



Gambar 4. Penandaan *bounding box* pada objek wajah siswa

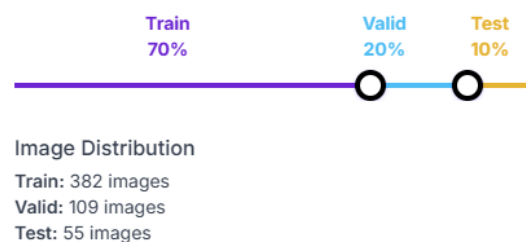
Setiap wajah siswa yang terdeteksi dalam *frame* diberikan label perilaku sesuai kategori yang telah ditentukan, yaitu: membaca, memperhatikan, menoleh, menulis, dan menunduk. Dihasilkan jumlah untuk tiap kategori perilaku siswa yang bisa di liat di tabel 2.

Tabel 2. Jumlah label tiap kelas

Label	Jumlah
Membaca	871
Memperhatikan	1052
Menoleh	569
Menoleh	220
Menunduk	336

Proses pelabelan memerlukan ketelitian ekstra, terutama untuk membedakan perilaku yang memiliki kemiripan secara visual, misalnya antara memperhatikan dan menoleh atau antara menulis dan membaca. Untuk memastikan konsistensi, peneliti menggunakan panduan label visual yang telah dibuat pada tahap perancangan penelitian.

Data set hasil anotasi, yang terdiri dari 546 gambar, kemudian dibagi menjadi tiga bagian: data latih sebanyak 70%, data validasi sebanyak 20%, dan data uji sebanyak 10% yang dapat dilihat pada gambar 5.



Gambar 5. Pembagian dataset

4.3. Pelatihan Model

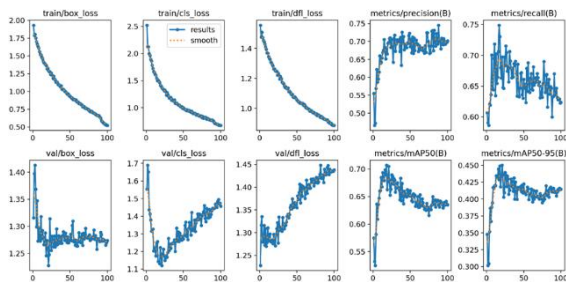
Pelatihan model YOLOv11 dilakukan untuk mengoptimalkan kemampuan model dalam mendeteksi dan mengklasifikasikan perilaku siswa secara otomatis dari citra kelas yang terekam.

Agar proses pelatihan berjalan optimal, digunakan PC dengan spesifikasi sebagai berikut:

Tabel 3. Spesifikasi PC

Komponen	Spesifikasi
Processor	Intel Core i5-12400F
RAM	32 GB DDR4
GPU	NVIDIA GeForce RTX 3060 Super (12GB VRAM)
Penyimpanan	SSD 1 TB
Sistem Operasi	Windows 11 64-bit
Software	Python 3.9, YOLOv11, PyTorch, CUDA Toolkit

Tahapan pelatihan model dimulai dengan inisialisasi model YOLOv11 menggunakan konfigurasi yang telah disesuaikan dengan kebutuhan deteksi perilaku siswa. Selanjutnya, model dilatih selama 100 *epoch* dengan parameter *batch size* 16, ukuran gambar 640×640 piksel, *learning rate* 0.001, serta *optimizer* Adam untuk mempercepat proses konvergensi. Pada setiap *epoch*, model melakukan proses *forward pass* untuk memprediksi *bounding box* dan kelas perilaku siswa pada setiap gambar. Hasil prediksi tersebut kemudian dievaluasi menggunakan fungsi *loss* gabungan yang terdiri dari *box loss*, *classification loss*, dan *DFL loss*. Proses *backpropagation* dijalankan untuk memperbarui bobot model berdasarkan hasil perhitungan *loss* tersebut sehingga model dapat terus belajar dan menyesuaikan parameter terhadap data yang diberikan..



Gambar 6. Hasil pelatihan model

Selama proses pelatihan, performa model dipantau secara berkala melalui pencatatan nilai *loss*, *precision*, *recall*, dan *mean Average Precision* (mAP) pada data validasi di setiap *epoch*. Hasil *monitoring* ditampilkan pada Gambar 6, yang menunjukkan tren penurunan *loss* yang konsisten dan peningkatan mAP pada data validasi, menandakan bahwa model berhasil mengalami konvergensi dan belajar secara efektif. Model terbaik, yakni model dengan nilai mAP tertinggi pada data validasi, kemudian disimpan untuk

digunakan pada tahap evaluasi dan implementasi model deteksi perilaku siswa.

4.4. Skenario Pengujian

Skenario pengujian pada penelitian ini dirancang untuk mengevaluasi kemampuan model YOLOv11 dalam mendeteksi dan mengklasifikasikan perilaku siswa pada data yang benar-benar baru dan belum pernah digunakan selama pelatihan. Pengujian dilakukan menggunakan data uji yang telah dipisahkan secara acak sebanyak 10% dari total dataset utama, sehingga proses evaluasi dapat mengukur kemampuan generalisasi model secara objektif terhadap berbagai kategori perilaku siswa.

Seluruh data uji tersebut diberikan ke model deteksi untuk menghasilkan prediksi *bounding box* dan label perilaku pada setiap wajah siswa dalam gambar. Kinerja model dievaluasi menggunakan sejumlah metrik, yaitu *precision*, *recall*, *mean Average Precision* (mAP50 dan mAP50-95), serta *confusion matrix* untuk tiap kategori perilaku. Selain evaluasi kuantitatif, dilakukan juga evaluasi kualitatif dengan cara meninjau hasil visualisasi prediksi model pada frame data uji, guna memastikan bahwa model mampu membedakan masing-masing kategori perilaku siswa dalam situasi kelas nyata.

4.5. Evaluasi Performa Model

Model YOLOv11 diuji pada lima kelas perilaku siswa, yaitu membaca, memperhatikan, menoleh, menulis, dan menunduk. Beberapa metrik utama yang digunakan dalam evaluasi adalah *precision* (presisi), *recall* (sensitivitas), serta *mean Average Precision* pada dua ambang *IoU*, yaitu mAP50 (*IoU* 50%) dan mAP50-95 (*IoU* 50%–95%). Secara umum, hasil evaluasi model dapat dilihat pada tabel 4.

Tabel 4. Hasil evaluasi model yolov11 secara umum

Metrik	Nilai
<i>Precision</i>	0,701 (70,1%)
<i>Recall</i>	0,624 (62,4%)
mAP50	0,635 (63,5%)
mAP50-95	0,415 (41,5%)

Nilai tersebut menunjukkan bahwa model memiliki tingkat ketepatan yang baik dalam mendeteksi perilaku siswa di kelas, meskipun masih terdapat beberapa kekeliruan deteksi pada kelas tertentu.

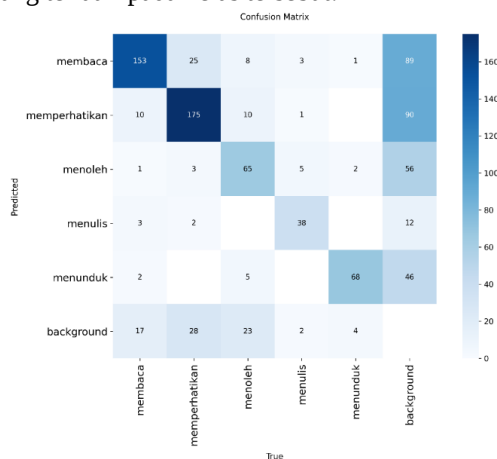
Tabel 5 merangkum performa model pada masing-masing kelas perilaku siswa:

Tabel 5. Hasil Evaluasi Model yolov11 Per-Kelas

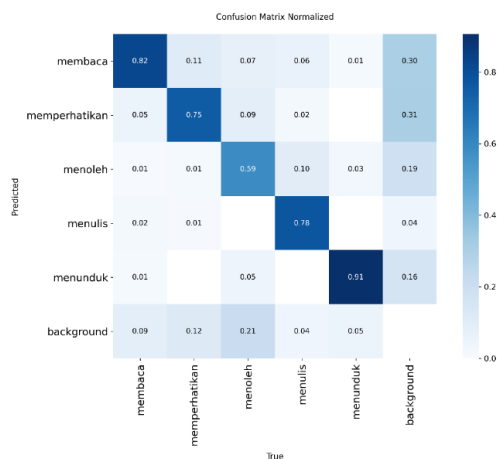
Kelas	<i>Precision</i>	<i>Recall</i>	mAP50	mAP50-95
Membaca	0,656	0,729	0,714	0,471
Memperhatikan	0,753	0,665	0,743	0,473
Menoleh	0,565	0,514	0,483	0,298
Menulis	0,804	0,735	0,776	0,508
Menunduk	0,713	0,840	0,813	0,503

Dari tabel di atas, dapat dilihat bahwa kelas menulis memperoleh nilai *Precision* tertinggi (0,804), menunjukkan bahwa prediksi menulis oleh model umumnya sangat akurat dan jarang keliru. Sementara itu, kelas menunduk memiliki nilai *recall* tertinggi (0,840), artinya sebagian besar perilaku menunduk pada siswa berhasil dideteksi oleh model. Kedua kelas ini juga mendapatkan skor mAP50 yang paling tinggi, yaitu 0,776 untuk menulis dan 0,813 untuk menunduk, menandakan model sangat andal dalam mengenali aktivitas tersebut di dalam kelas.

Kelas memperhatikan dan membaca juga menunjukkan performa yang baik, dengan skor mAP50 sekitar 0,74 dan 0,71, sehingga sistem cukup andal dalam mengenali siswa yang memperhatikan guru atau sedang membaca buku. Namun, pada kelas menoleh, nilai *Precision* dan *recall* berada pada posisi terendah (0,565 dan 0,514), demikian juga nilai mAP50 (0,483). Hal ini menunjukkan bahwa model sering mengalami kesulitan membedakan perilaku menoleh dari perilaku lain yang secara visual mirip, seperti memperhatikan atau menunduk. Hal ini dapat disebabkan oleh kemiripan visual (misal, sudut pandang kepala yang hanya sedikit berubah), variasi pose yang sangat beragam, atau jumlah data menoleh yang lebih sedikit dalam dataset sehingga model kurang terlatih pada kelas tersebut.



Gambar 7. Confusion matrix

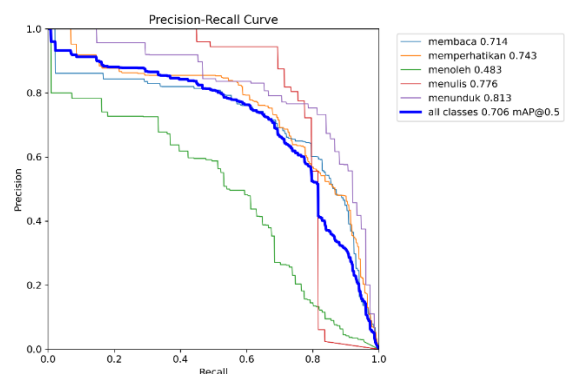


Gambar 8. Confusion matrix normalized

Confusion matrix yang dihasilkan seperti yang terlihat pada gambar 7 dan gambar 8 memperlihatkan bahwa prediksi benar (*true positive*) paling banyak berada di diagonal, khususnya untuk kelas membaca, memperhatikan, menulis, dan menunduk. Hal ini sejalan dengan nilai presisi dan *recall* yang tinggi pada kelas tersebut.

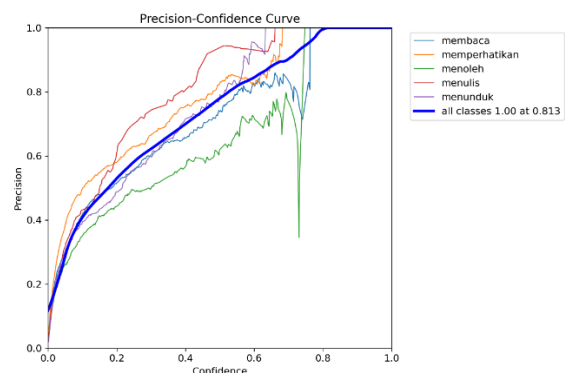
Namun, pada kelas menoleh, terdapat kecenderungan model melakukan kesalahan klasifikasi ke kelas memperhatikan dan menunduk. Kesalahan ini umumnya terjadi karena dalam beberapa kasus siswa yang hanya sedikit memalingkan wajah masih dapat dikira sebagai memperhatikan, atau jika kepala sedikit menunduk saat menoleh, dapat terklasifikasi sebagai menunduk.

Selain *confusion matrix*, performa model juga dievaluasi menggunakan berbagai kurva metrik, seperti *Precision-Recall* (PR), *Precision*, *Recall*, dan *F1-Score*.

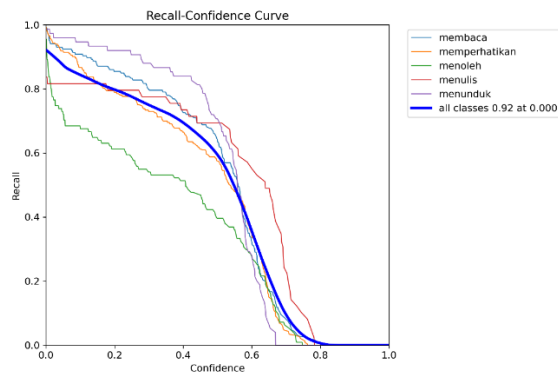


Gambar 9. Precision-Recall

Berdasarkan analisis kurva Precision-Recall pada gambar 9, kelas menunduk dan menulis memiliki area di bawah kurva (*average precision/AP*) yang paling besar, yang sejalan dengan nilai mAP yang tinggi pada dua kelas tersebut. Sementara itu, kurva *Precision-Recall* untuk kelas menoleh tampak lebih rendah, menandakan bahwa model mengalami kompromi lebih besar antara presisi dan *recall* pada kelas tersebut dibanding kelas lainnya.

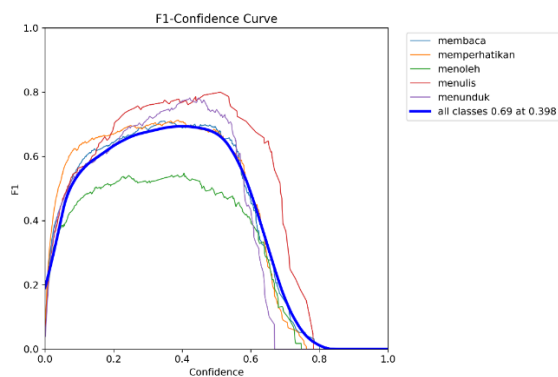


Gambar 10. Precision



Gambar 11. Recall

Kurva Precision pada gambar 10 memperlihatkan bahwa semakin tinggi ambang kepercayaan (*confidence threshold*), nilai presisi model juga meningkat. Hal ini terjadi karena prediksi yang dihasilkan pada *confidence* tinggi umumnya lebih akurat sehingga jumlah *false positive* menurun. Sebaliknya, kurva *Recall* pada gambar 11 cenderung menurun seiring meningkatnya *threshold*, karena prediksi menjadi semakin selektif dan lebih banyak objek aktual yang terlewat (*false negative* bertambah).

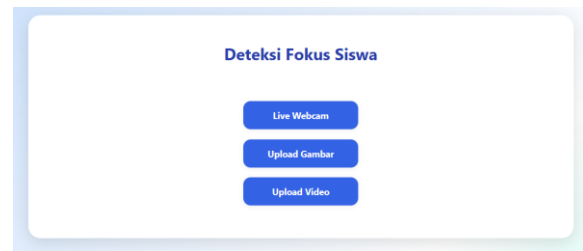


Gambar 12. F1-score

Sementara itu, kurva *F1-Score* pada gambar 12 menunjukkan adanya titik optimal pada *threshold confidence* sekitar 0,25 hingga 0,30. Pada rentang nilai *threshold* tersebut, model mencapai keseimbangan terbaik antara presisi dan *recall*, sehingga dapat dijadikan acuan dalam menentukan *threshold* operasi model untuk aplikasi di lingkungan nyata.

4.6. Implementasi Model

Setelah proses pelatihan dan evaluasi selesai, model YOLOv11 diimplementasikan dalam sistem *real-time* menggunakan aplikasi web berbasis *Flask*. Sistem ini memungkinkan deteksi fokus siswa dilakukan secara langsung di kelas dan hasilnya dapat diakses oleh guru melalui *dashboard* web interaktif yang dapat dilihat pada gambar 13.



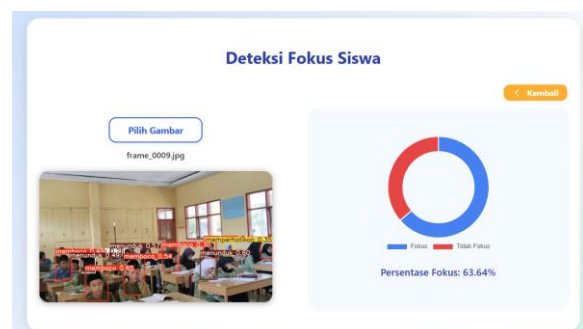
Gambar 13. Tampilan menu utama deteksi fokus siswa pada aplikasi web

Proses deteksi perilaku siswa dalam aplikasi ini dilakukan secara *real-time* dengan memanfaatkan keunggulan YOLOv11, baik pada mode *live webcam*, *upload* gambar, maupun *upload* video. Setiap *frame* yang masuk ke sistem akan diproses oleh model untuk mendeteksi wajah siswa, kemudian mengklasifikasikan masing-masing ke dalam lima perilaku utama, yaitu membaca, memperhatikan, menoleh, menulis, dan menunduk. Hasil deteksi divisualisasikan secara langsung dalam bentuk *bounding box* dan label perilaku pada wajah siswa, sehingga guru atau pengamat dapat memantau kondisi fokus kelas secara aktual.



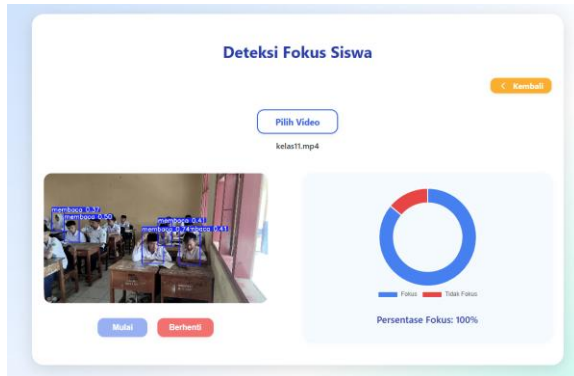
Gambar 14. Tampilan deteksi webcam

Pada mode *Live Webcam* (Gambar 14), sistem memproses *input* video dari kamera secara kontinu dan menampilkan hasil deteksi secara langsung. Guru dapat memantau perilaku siswa di kelas secara *real-time*, serta melihat perbandingan jumlah siswa yang fokus dan tidak fokus melalui diagram yang ditampilkan di sisi *dashboard*.



Gambar 15. Tampilan deteksi gambar

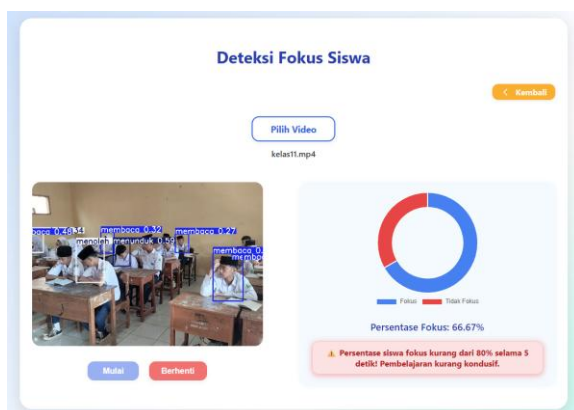
Pada mode *Upload Gambar* (Gambar 15), aplikasi memungkinkan pengguna mengunggah foto kelas, kemudian sistem akan memproses citra tersebut untuk mendeteksi perilaku setiap siswa. Hasil deteksi berupa *bounding box* dan label perilaku langsung divisualisasikan di atas gambar, sementara persentase fokus siswa juga diperbarui secara otomatis.



Gambar 16. Tampilan deteksi video

Pada mode *Upload Video* (Gambar 16), aplikasi dapat menganalisis perilaku siswa dalam rekaman pembelajaran. Setiap *frame* video akan diproses secara bertahap sehingga distribusi perilaku siswa dapat diamati sepanjang durasi video. Sistem juga menghitung persentase fokus siswa secara kumulatif dan menampilkannya dalam bentuk *pie chart*.

Fitur analisis statistik menjadi nilai tambah yang signifikan pada sistem ini. Diagram *pie chart* menyajikan proporsi siswa dengan status “fokus” (membaca, memperhatikan, menulis) dan “tidak fokus” (menoleh, menunduk), sehingga guru bisa segera mengidentifikasi situasi kelas secara kuantitatif. Ketika persentase siswa yang tidak fokus melampaui ambang batas tertentu, aplikasi secara otomatis memunculkan notifikasi suara sebagai peringatan seperti yang terlihat di gambar 17, memungkinkan intervensi lebih cepat oleh guru.



Gambar 17. Tampilan deteksi video ketika presentasi fokus kurang dari seharusnya

Fitur-fitur ini membuat sistem deteksi fokus siswa menjadi tidak hanya informatif, tetapi juga sangat aplikatif dalam mendukung aktivitas belajar

mengajar secara langsung. Implementasi web ini telah diuji secara nyata di kelas dan berjalan stabil, dengan hasil deteksi yang sesuai dengan kondisi aktual di lapangan.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penerapan *deep learning* dengan model YOLOv11 mampu menghasilkan sistem deteksi perilaku siswa yang akurat dan berjalan secara *real-time* di lingkungan kelas, dengan kemampuan mengenali lima kategori aktivitas utama yaitu membaca, memperhatikan, menoleh, menulis, dan menunduk serta rata-rata mAP50 di atas 70% pada sebagian besar kelas perilaku. Integrasi model ke dalam aplikasi web berbasis *Flask* terbukti efektif dalam mendukung pemantauan fokus siswa secara interaktif oleh guru. Namun, masih ditemukan tantangan pada deteksi perilaku menoleh yang memiliki akurasi relatif lebih rendah, sehingga disarankan untuk meningkatkan jumlah data pelatihan pada kategori tersebut, melakukan *augmentasi* data yang lebih beragam, serta mengoptimalkan arsitektur model dan menambah fitur notifikasi otomatis guna meningkatkan keandalan sistem. Secara umum, penelitian ini menunjukkan bahwa teknologi deteksi objek berbasis *deep learning* sangat potensial diterapkan di bidang pendidikan, dan diharapkan dapat menjadi landasan bagi pengembangan *smart classroom* yang lebih efektif dan adaptif di masa depan.

DAFTAR PUSTAKA

- [1] A. Abdullah and S. Yahya, “KAJIAN PEMANFAATAN DEEP LEARNING DALAM PEMBELAJARAN,” *transformasi*, vol. 7, no. 1 SE-Articles, pp. 25–41, Jul. 2025, [Online]. Available: <https://transformasi.kemenag.go.id/index.php/journal/article/view/333>
- [2] A. Andriana, “Model Pembelajaran Berbasis Deep Learning Bagi Siswa Inklusi di Pendidikan Vokasi,” *J. Tiarsie; Vol 18 No 4 J. TIARSIE 18.4*, 2021, doi: 10.32816/tiarsie.v18i4.129.
- [3] P. Lusiana, “UPAYA MENINGKATKAN KONSENTRASI BELAJAR SISWA MELALUI METODE PROBLEM BASED LEARNING DALAM LAYANAN BIMBINGAN KLASIKAL DI KELAS XII SMAN 1 GEDUNG MENENG,” *Semin. Nas. Pendidik. Profesi Guru FKIP UAD*, vol. 1, no. 1, Dec. 2021, Accessed: Jul. 09, 2025. [Online]. Available: <http://seminar.uad.ac.id/index.php/SemNasPPG/article/view/12064>
- [4] F. Bimantoro, I. G. Pasek, and S. Wijaya, “DETEKSI FOKUS MAHASISWA MENGGUNAKAN ALGORITMA YOU,” vol. x, no. December, 2023.
- [5] S. T. Rasmana, J. Ahmad Wali Bahari, and A. Y. Wicaksono, “Analisa Suasana Belajar Kelas

- Berdasarkan Deteksi Ekspresi Wajah Menggunakan Deepface,” *J. Edukasi dan Penelit. Inform.*, vol. 10, no. 3, pp. 401–410, 2024.
- [6] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, “The Understanding of Deep Learning: A Comprehensive Review,” *Math. Probl. Eng.*, vol. 2021, no. 1, p. 5548884, Jan. 2021, doi: 10.1155/2021/5548884.
- [7] C. A. Sugianto, D. Rohmayani, J. Fredricka, and M. Doheir, “Interpretable Deep Learning Model for Grape Leaf Disease Classification Based on EfficientNet with Grad- CAM Visualization,” *J. Innov. Inf. Technol. Appl.*, vol. 7, no. 1, pp. 143–152, 2025.
- [8] R. Khanam and M. Hussain, “YOLOv11: An Overview of the Key Architectural Enhancements,” vol. 2024, pp. 1–9, 2024, [Online]. Available: <http://arxiv.org/abs/2410.17725>
- [9] L. He, Y. Zhou, L. Liu, and J. Ma, “Research and Application of YOLOv11-Based Object Segmentation in Intelligent Recognition at Construction Sites,” *Build. 2024, Vol. 14, Page 3777*, vol. 14, no. 12, p. 3777, Nov. 2024, doi: 10.3390/BUILDINGS14123777.
- [10] S. Maesaroh *et al.*, *Bahasa Pemrograman Python*. 2024.
- [11] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc, 2022.
- [12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. “O’Reilly Media, Inc.,” 2008.
- [13] “Roboflow: Computer vision tools for developers and enterprises.” Accessed: Jul. 06, 2025. [Online]. Available: <https://roboflow.com/>
- [14] H. Walingkas and P. Saian, “Penerapan Framework Flask pada Pembangunan Sistem Informasi Pemasok Barang,” *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 7, pp. 227–234, Apr. 2023, doi: 10.35870/jtik.v7i2.729.
- [15] A. S. Pillai, “Student Engagement Detection in Classrooms through Computer Vision and Deep Learning: A Novel Approach Using YOLOv4,” *Sage Sci. Rev. Educ. Technol.*, vol. 5, pp. 87–97, Aug. 2022, doi: 10.2139/SSRN.5187374.
- [16] K. Salma and S. Hidayat, “Deteksi Antusiasme Siswa dengan Algoritma Yolov8 pada Proses Pembelajaran Daring,” *J. Indones. Manaj. Inform. dan Komun.*, vol. 5, no. 2, pp. 1611–1618, May 2024, doi: 10.35870/JIMIK.V5I2.716.
- [17] Z. He, K. Wang, T. Fang, L. Su, R. Chen, and X. Fei, “Comprehensive Performance Evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on Object Detection of Power Equipment,” Nov. 2024, Accessed: Feb. 07, 2025. [Online]. Available: <http://arxiv.org/abs/2411.18871>