

Deteksi Objek Berbahaya pada Video Animasi Kartun Anak Menggunakan YOLO26 Berbasis Computer Vision

Ridwan Widiyono^{*1}, Muslim Hidayat², Iman Ahmad Ihsannuddin³

^{1,2,3}Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Sains Al-Qur'an, Indonesia
Email: ¹ridwanwidiyono36@gmail.com, ²muslim_h@unsiq.ac.id, ³ahmadihsan@unsiq.ac.id

Abstrak

Perkembangan media digital meningkatkan akses anak terhadap tayangan animasi kartun yang tidak selalu aman, di mana beberapa tayangan diketahui mengandung objek berbahaya seperti pisau, senjata tajam, pistol, dan senapan yang berpotensi memberikan pengaruh negatif terhadap perkembangan anak. Penelitian ini bertujuan mengimplementasikan sistem deteksi objek berbahaya pada video animasi kartun anak menggunakan algoritma YOLO26 (*You Only Look Once*) untuk filtrasi konten. Kontribusi utama penelitian ini adalah pembangunan dataset anotasi khusus berisi 4.486 gambar dengan 5.060 *bounding box* yang dikumpulkan dari berbagai gaya animasi kartun, berbeda dari penelitian terdahulu yang umumnya menggunakan dataset domain nyata (*real-world*). Dataset terdiri atas empat kelas objek (pisau, senjata tajam, pistol, dan senapan) dan dibagi dengan rasio 80:10:10 untuk data latih, validasi, dan uji, dengan model YOLO26 di-*fine-tuning* dari bobot pra-latih COCO. Model YOLO26 dilatih dan dievaluasi menggunakan metrik *Precision*, *Recall*, *F1-Score*, *mAP@0.5*, dan *mAP@0.5:0.95* terhadap 449 gambar uji dengan 501 objek berbahaya. Hasil evaluasi menunjukkan *Precision* sebesar 0,945, *Recall* 0,938, *F1-Score* 0,941, *mAP@0.5* sebesar 0,953, dan *mAP@0.5:0.95* sebesar 0,763, dengan *mAP@0.5* di atas 0,88 pada seluruh kelas. Model diimplementasikan sebagai purwarupa aplikasi berbasis web untuk filtrasi konten yang memungkinkan pengguna memasukkan tautan video animasi dari repositori video aplikasi dan memperoleh hasil deteksi otomatis.

Kata kunci: Animasi Kartun, YOLO26, Deteksi Objek, Objek Berbahaya, Computer Vision

Abstract

The rapid growth of digital media has increased children's access to animated cartoon programs that are not always safe, with some containing dangerous objects such as knives, sharp weapons, pistols, and rifles that may negatively affect children's development. This study aims to implement a dangerous object detection system for children's animated cartoon videos using the YOLO26 (*You Only Look Once*) algorithm for content filtering. The main contribution is the construction of a custom annotated dataset comprising 4,486 images with 5,060 bounding boxes collected from various cartoon animation styles, distinguishing it from prior studies that mostly relied on real-world datasets. The dataset consists of four object classes (knife, sharp weapon, pistol, and rifle) and was split 80:10:10 for training, validation, and testing, with the YOLO26 model fine-tuned from COCO pre-trained weights. The YOLO26 model was trained and evaluated using *Precision*, *Recall*, *F1-Score*, *mAP@0.5*, and *mAP@0.5:0.95* on 449 test images containing 501 dangerous objects. Results showed a *Precision* of 0.945, *Recall* of 0.938, *F1-Score* of 0.941, *mAP@0.5* of 0.953, and *mAP@0.5:0.95* of 0.763**, with *mAP@0.5* above 0.88 across all classes**. The model was implemented as a prototype web-based application for content filtering that allows users to submit links to animated videos from the application's video repository and automatically obtain detection results.

Keywords: Cartoon Animation, YOLO26, Object Detection, Dangerous Object, Computer Vision

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)



1. PENDAHULUAN

Fenomena peningkatan konsumsi video animasi anak telah menjadi tren signifikan dalam ekosistem media digital saat ini. Berdasarkan laporan *Precise Advertiser Report - Kids (PARK)*, dominasi ini terlihat dari tingginya penggunaan platform berbagi video seperti YouTube yang mencapai 85% pada anak-anak, di mana 90% orang tua juga ikut menonton bersama mereka secara rutin [1]. Tingginya penetrasi ini sejalan dengan data *The Common Sense Census* yang mencatat bahwa menonton

video menyumbang 73% dari total waktu layar anak, dengan durasi tontonan online yang melonjak drastis hingga rata-rata 39 menit per hari [2]. Kondisi tersebut menjadikan video animasi kartun kini tidak lagi sekadar hiburan bagi anak. Konten yang dikonsumsi setiap hari turut membentuk cara berpikir, perilaku, dan perkembangan sosial mereka, sehingga keamanan konten menjadi hal yang tidak bisa diabaikan. Platform *streaming* memunculkan celah keamanan baru, di mana banyak tayangan berkedok animasi anak yang secara terselubung menyisipkan objek, adegan kekerasan, atau pesan tidak pantas [3]. Salah satu konten yang tersedia di platform *YouTube Kids*, yaitu video berjudul “*Tom and Jerry | Mega Compilation | Vol. 20 | Warner Classics*”. Pada bagian menit ke 38:10, terdapat adegan di mana karakter Tom mengayunkan kapak ke arah kepala karakter lain [4]. Meskipun adegan tersebut tidak memperlihatkan kontak langsung atau dampak kekerasan secara eksplisit, tindakan tersebut tetap merepresentasikan perilaku agresif yang berpotensi berbahaya jika ditiru oleh anak-anak. Kondisi ini sejalan dengan Pedoman Perilaku Penyiaran dan Standar Program Siaran (P3SPS) yang diterbitkan oleh Komisi Penyiaran Indonesia (KPI), yang secara eksplisit melarang penayangan konten kekerasan, senjata, dan adegan berbahaya pada program yang ditujukan untuk anak-anak [5]. Oleh karena itu, dibutuhkan metode yang andal untuk mendeteksi keberadaan objek berbahaya dalam kartun animasi anak secara otomatis, bukan hanya mengandalkan laporan pengguna atau moderasi manual.

Untuk menjawab tantangan tersebut, teknologi visi komputer (*computer vision*) yang dipadukan dengan pembelajaran mendalam (*deep learning*) menawarkan pendekatan komputasi yang sangat efektif untuk mengenali, melokalisasi, dan mengklasifikasikan entitas tertentu di dalam suatu tayangan visual [6]. Mengingat konten yang dianalisis berupa video animasi yang terdiri dari pergerakan puluhan bingkai gambar (*frame*) per detik, diperlukan algoritma yang mampu melakukan pemrosesan secara cepat dan *real-time*. Algoritma *You Only Look Once* (YOLO) hadir sebagai salah satu metode deteksi objek mutakhir yang memenuhi kriteria tersebut [7]. Keandalan arsitektur ini pun telah dikaji dan dibuktikan oleh berbagai penelitian terdahulu, yang berhasil mengimplementasikan algoritma serupa untuk mendeteksi objek secara spesifik dan akurat pada berbagai jenis pemrosesan citra maupun video [8], [9], [10], [11]. Deteksi objek berbahaya pada domain animasi tetap menghadapi masalah ilmiah tersendiri, yaitu karakteristik visual animasi (tekstur minim, garis tepi sederhana, proporsi objek bervariasi, dan warna tidak realistis) berbeda dari citra nyata, sehingga model yang dilatih pada citra nyata cenderung mengalami penurunan performa pada domain animasi [12], [13], [14].

Penelitian terdahulu yang relevan dengan masalah tersebut dapat dikelompokkan menjadi deteksi konten kekerasan, deteksi objek berbahaya, deteksi pada domain animasi, dan perkembangan arsitektur YOLO. Masing-masing kelompok memiliki keterbatasan yang diuraikan berikut.

Kelompok pertama adalah deteksi konten kekerasan. *Violence-YOLO*, yang paling dekat dengan penelitian ini, dikembangkan untuk mendeteksi kekerasan secara *real-time* menggunakan YOLO [8]. Pendekatan ini menunjukkan bahwa YOLO dapat dimanfaatkan untuk pemantauan konten secara cepat, sehingga relevan sebagai dasar sistem penyaringan konten. Model ini ditujukan untuk mendeteksi kekerasan secara umum, bukan melokalisasi objek berbahaya tertentu [8]. Akibatnya, hasil deteksinya tidak menunjukkan benda apa yang menjadi sumber bahaya dalam suatu adegan, padahal informasi tersebut dibutuhkan untuk penyaringan konten anak yang spesifik, seperti pada penelitian ini.

Kelompok kedua adalah deteksi objek berbahaya. Zeng dkk. mengusulkan model JRL-YOLO yang dikhususkan untuk deteksi objek berbahaya secara *real-time* dan berhasil mencapai nilai mAP sebesar 85,03%, namun model ini masih mengalami fenomena *missing detection* ketika dua objek target berada terlalu berdekatan, mengindikasikan keterbatasan dalam menangkap relasi spasial antar objek secara menyeluruh [9]. Model ini dikembangkan pada citra domain nyata dan tidak dievaluasi pada domain animasi [9]. Karena karakteristik visual animasi berbeda dari citra nyata, performa pendekatan semacam ini pada animasi kartun belum diketahui.

Kelompok ketiga adalah deteksi pada domain animasi. Tang mengembangkan sistem berbasis CNN bernama *AnimeNet* untuk mendeteksi konten kekerasan dan erotis pada video animasi kartun, namun model tersebut hanya mencapai akurasi sebesar 74,5%, dengan catatan bahwa penurunan performa tersebut disebabkan oleh kompleksitas tugas deteksi, karakteristik dataset yang terbatas, serta tantangan subjektivitas dalam menentukan konten yang tidak pantas [15]. Khalil dkk. [16] mengembangkan pendekatan berbasis fitur visual untuk mendeteksi kekerasan pada video kartun, sementara Taha dkk. [17] mengusulkan *InspectorNet* berbasis arsitektur transformer untuk tugas yang serupa, namun kedua pendekatan tersebut berfokus pada klasifikasi adegan kekerasan secara umum dan tidak melakukan

lokalisasi objek secara spesifik [16], [17]. Ketiga studi ini hanya mengklasifikasikan konten tanpa melokalisasi objek berbahaya [15], [16], [17].

Kelompok keempat adalah perkembangan arsitektur YOLO. Violence-YOLO menunjukkan bahwa pemilihan varian arsitektur YOLO secara signifikan memengaruhi performa deteksi [8], sehingga perbandingan arsitektur menjadi pertimbangan penting sebelum menentukan model yang digunakan. Perkembangan arsitektur YOLO telah berevolusi dari YOLOv5 hingga YOLOv10, mencakup penggunaan *backbone* CSPDarknet yang efisien, transisi ke model *anchor-free* pada YOLOv8, integrasi *Programmable Gradient Information* (PGI) dan *Generalized Efficient Layer Aggregation Network* (GELAN) pada YOLOv9, serta penghapusan *Non-Maximum Suppression* (NMS) pada YOLOv10 untuk meningkatkan kecepatan inferensi [11]. Evolusi arsitektural tersebut menjadi dasar pertimbangan dalam memilih varian YOLO yang sesuai dengan karakteristik domain animasi pada penelitian ini. Perkembangan arsitektur ini belum dikaji pada deteksi objek berbahaya di domain animasi kartun anak.

Selain perbandingan arsitektur, aspek transfer learning antar-domain juga menjadi pertimbangan penting dalam penelitian ini. *Transfer learning* merupakan strategi yang memanfaatkan pengetahuan dari domain sumber (*source domain*) yang memiliki data berlimpah untuk meningkatkan performa model pada domain target (*target domain*) yang memiliki data terbatas [18]. Beberapa penelitian menunjukkan bahwa proses *fine-tuning* model yang telah dilatih pada domain citra nyata seperti COCO dapat memberikan representasi fitur awal yang bermanfaat meskipun domain target berupa citra sintesis atau bergaya non-fotografis, sebab fitur tingkat rendah seperti tepi dan bentuk dasar relatif dapat ditransfer antar-domain, sementara fitur tingkat tinggi yang lebih spesifik terhadap tekstur dan warna umumnya memerlukan penyesuaian lebih lanjut melalui pelatihan ulang pada data domain target [19]. Atas dasar tersebut, penelitian ini memanfaatkan bobot pra-latih COCO sebagai titik awal pelatihan, kemudian dilakukan *fine-tuning* menggunakan dataset citra animasi kartun anak yang dikumpulkan secara khusus, sehingga model diharapkan dapat mempelajari karakteristik visual objek berbahaya pada domain animasi secara lebih efektif dibandingkan pelatihan dari awal (*training from scratch*).

Kesenjangan penelitian terletak pada belum adanya deteksi objek berbahaya yang terlokalisasi pada domain animasi. Penelitian deteksi objek berbahaya sebelumnya umumnya menggunakan dataset domain citra nyata [9], sedangkan penelitian pada domain animasi hanya mengklasifikasikan adegan kekerasan [15], [16], [17] tanpa melokalisasi objek berbahaya secara spesifik. Penerapan arsitektur YOLO terbaru pada domain non-fotografis seperti animasi kartun juga masih terbatas. Belum ada penelitian yang melokalisasi objek berbahaya (pisau, senjata tajam, pistol, dan senapan) pada animasi kartun anak menggunakan dataset anotasi khusus domain animasi, menurut sepengetahuan penulis. Kesenjangan tersebut menunjukkan kebutuhan nyata akan sistem deteksi yang tidak hanya mampu mencapai akurasi tinggi, tetapi juga efisien secara komputasi dan mencakup cakupan kelas objek berbahaya yang lebih luas dan beragam, khususnya dalam domain video animasi kartun anak [20].

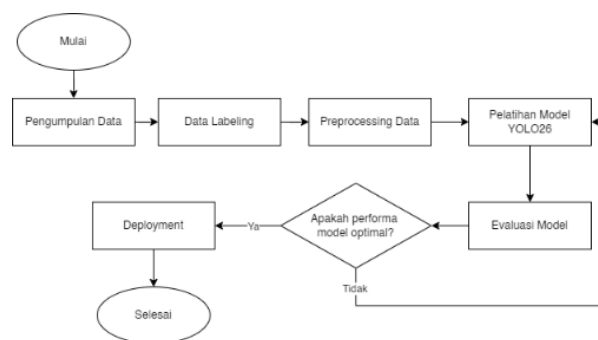
Untuk mengatasi keterbatasan tersebut, penelitian ini mengusulkan pendekatan deteksi objek berbahaya yang ditujukan secara spesifik untuk mengenali karakteristik visual pada animasi kartun anak. Karakteristik visual tersebut menjadi tantangan tersendiri bagi model deteksi objek, mengingat domain animasi memiliki distribusi fitur yang berbeda secara signifikan dibandingkan dengan citra fotografis nyata, sehingga model yang dilatih pada dataset dunia nyata cenderung mengalami penurunan performa ketika diterapkan pada domain animasi [12]. Secara lebih spesifik, citra animasi kartun umumnya hanya mempertahankan struktur kasar dan informasi warna dari suatu objek, berbeda dengan citra fotografis nyata yang memuat detail tekstur, pencahayaan, dan bayangan secara utuh [13]. Selain itu, ilustrasi pada animasi cenderung memiliki tekstur yang minim atau bahkan tidak ada, garis tepi yang lebih sederhana dan eksplisit, serta variasi proporsi bentuk objek antar judul animasi yang berbeda-beda, sehingga fitur visual yang dipelajari oleh model pada domain citra nyata sulit ditransfer secara langsung ke domain animasi [14]. Perbedaan karakteristik visual ini melandasi kebutuhan akan dataset khusus, alih-alih menggunakan dataset domain nyata yang umum pada penelitian deteksi objek berbahaya sebelumnya. Penelitian ini melatih model menggunakan dataset yang merangkum berbagai representasi dua dimensi dari objek berbahaya (seperti pisau, senjata tajam, pistol, dan senapan) yang diekstraksi langsung dari video animasi. Dengan pendekatan ini, model mampu mempelajari variasi spasial dan bentuk entitas berbahaya dalam lingkungan visual animasi yang memiliki gaya ilustrasi bervariasi, bentuk yang disederhanakan, serta warna yang tidak realistis.

Sistem ini memanfaatkan arsitektur *You Only Look Once* (YOLO), menggunakan pendekatan *single-stage detection* yang mampu memproses keseluruhan bingkai gambar (*frame*) secara serentak

untuk mempertahankan kecepatan komputasi tinggi (*real-time*) dan menangkap relasi antar objek secara komprehensif [10]. Kontribusi penelitian ini adalah (1) dataset anotasi khusus domain animasi kartun berisi 4.486 gambar dengan 5.060 bounding box dalam empat kelas, (2) penerapan YOLO26 melalui *fine-tuning* dari bobot COCO pada domain non-fotografis, dan (3) purwarupa aplikasi web yang menjalankan deteksi objek berbahaya pada video. Penelitian ini bertujuan mengembangkan dan mengevaluasi model deteksi objek berbahaya (pisau, senjata tajam, pistol, dan senapan) pada video animasi kartun anak menggunakan YOLO26, serta mengimplementasikannya sebagai purwarupa aplikasi web.

2. METODE PENELITIAN

Penelitian ini dirancang sebagai studi deteksi objek berbasis *deep learning* untuk mengidentifikasi objek berbahaya dalam video animasi kartun anak secara otomatis. Sistem yang dikembangkan memanfaatkan arsitektur *You Only Look Once* (YOLO) sebagai tulang punggung deteksi, didukung oleh pipeline pengolahan data berbasis *computer vision* yang mencakup pengumpulan data, anotasi, pra-proses, pelatihan model, dan evaluasi. Keseluruhan tahapan penelitian digambarkan dalam diagram alir pada Gambar 1.



Gambar. 1. Tahapan proses penelitian

2.1. Pengumpulan Data

Pengumpulan dataset dilakukan dengan memanfaatkan konten video animasi kartun anak yang tersedia secara publik pada platform YouTube. Proses pengumpulan data difokuskan pada video-video yang mengandung adegan dengan kategori objek berbahaya, meliputi pisau, senjata tajam (kapak dan pedang), pistol, dan senapan. Data bersumber dari 52 video pada 34 kanal YouTube dengan jenis animasi 2D, baik produksi klasik maupun modern, dengan total durasi 2 jam 57 menit (rata-rata sekitar 3 menit 25 detik per video). Sekitar tiga perempat video berasal dari seri Tom and Jerry dan Looney Tunes. Video dipilih berdasarkan ketersediaannya secara publik dan kemunculan adegan yang memuat salah satu dari empat kelas objek berbahaya. Video sumber dataset tidak didistribusikan ulang. Daftar kanal, tautan, durasi, dan tanggal akses video sumber disediakan pada repositori publik (<https://github.com/Mryourhead/sumber-video-deteksi-objek-berbahaya-yolo26>) sehingga video dapat ditelusuri langsung pada platform YouTube. Tahap awal pengumpulan data dilakukan dengan memotong bagian video yang memuat adegan objek berbahaya menjadi klip berdurasi maksimal 30 detik untuk mempermudah proses ekstraksi. *Frame* diekstraksi dari setiap klip menggunakan frame-extractor.com dengan laju 10 frame per detik sehingga satu klip menghasilkan hingga 300 gambar. Gambar yang memuat objek berbahaya dipilih secara manual dari hasil ekstraksi. Setiap gambar disimpan dalam format gambar untuk digunakan sebagai bagian dari dataset awal.

Penelitian ini menetapkan empat kategori objek berbahaya, yaitu pisau, senjata tajam, pistol, dan senapan. Kriteria pembeda antarkelas didasarkan pada karakteristik bentuk visual objek, bukan pada fungsi atau tingkat ketajaman. Kelas pisau merujuk pada senjata tajam berukuran kecil yang digunakan dengan genggam satu tangan dan memiliki bilah pendek, seperti pisau dapur atau pisau saku. Kelas senjata tajam mencakup pedang dan kapak, yaitu objek tajam berukuran besar dengan tangkai atau bilah yang lebih panjang serta umumnya digunakan dengan kedua tangan atau ayunan penuh. Pemisahan ini dilakukan karena kedua kelompok objek tersebut memiliki proporsi bentuk, panjang bilah, dan pola

gerakan visual yang cukup berbeda pada *frame* animasi, sehingga dipertimbangkan sebagai dua kelas terpisah untuk meningkatkan presisi deteksi model. Sementara itu, kelas pistol dan senapan dibedakan berdasarkan ukuran dan cara penggunaannya, yaitu pistol sebagai senjata api genggam berukuran kecil dan senapan sebagai senjata api panjang yang umumnya dipegang dengan kedua tangan. Pedoman pembagian kelas ini diterapkan secara konsisten pada seluruh proses anotasi dataset. Gambar disertakan apabila objek berbahaya tampil utuh pada *frame* dan bentuk dasarnya dapat dikenali. Gambar dikeluarkan apabila objek terpotong oleh tepi *frame* atau bentuk dasarnya tidak dapat dikenali.

Total dataset yang berhasil dikumpulkan sebanyak 5.000 gambar yang terbagi ke dalam empat kategori objek berbahaya dengan masing-masing sebanyak 1.250 gambar per kategori, yaitu pisau, senjata tajam, pistol, dan senapan, sebagaimana diuraikan pada Tabel 1. Sebanyak 4.486 gambar memenuhi kriteria dan diikutkan dalam dataset, sedangkan 514 gambar dikeluarkan karena objek berbahaya terpotong pada tepi *frame*. Setelah proses pengumpulan data selesai, dataset kemudian memasuki tahap pelabelan objek menggunakan *bounding box* serta praproses data sebelum digunakan pada proses pelatihan model

Tabel 1. Hasil Pengumpulan Dataset (Gambar yang Dikumpulkan sebelum Penyaringan)

No.	Kategori	Jumlah Gambar
1	Pisau	1250
2	Senjata tajam	1250
3	Pistol	1250
4	Senapan	1250

2.2. Data labeling

Tahap *data labeling* dilakukan untuk memberikan anotasi pada setiap objek berbahaya yang terdapat pada dataset gambar. Proses anotasi dilakukan menggunakan *platform Roboflow*, yaitu *platform* berbasis *web* yang mendukung pengelolaan dataset dan anotasi objek dalam bidang *computer vision*[21]. Metode anotasi yang digunakan adalah *bounding box*, di mana setiap objek berbahaya dalam gambar diberi label sesuai kelas yang telah ditentukan, disertai koordinat kotak pembatas yang merepresentasikan posisi dan ukuran objek secara spasial. Format anotasi yang digunakan adalah format standar YOLO, di mana setiap label menyimpan informasi kelas, koordinat pusat, serta lebar dan tinggi kotak pembatas dalam bentuk nilai yang telah ternormalisasi terhadap dimensi gambar[22]. Pedoman pelabelan mengikuti definisi kelas pada Bagian 2.1. Pekerjaan dilakukan oleh dua anotator yang saling memvalidasi hasil kerja satu sama lain (validasi silang) dengan memeriksa apakah objek berbahaya tampil utuh, tidak terpotong oleh tepi *frame*, dan bentuk dasarnya masih dapat dikenali. Gambar yang tidak memenuhi kriteria tersebut tidak dimasukkan ke dalam dataset sehingga objek yang ambigu tidak diikutkan dalam pelatihan. Tingkat kesepakatan label antar-anotator tidak dihitung secara kuantitatif.

Dari proses pelabelan ini dihasilkan total sebanyak 5.060 *bounding box* dari keseluruhan 4.486 gambar teranotasi. Distribusi anotasi untuk setiap kelas dapat dilihat pada Tabel 2. Jumlah *bounding box* yang melebihi jumlah gambar disebabkan oleh adanya gambar yang memuat lebih dari satu objek berbahaya secara bersamaan dalam satu *frame*.

Tabel 2. Distribusi Anotasi Setiap Kelas

No.	Kategori	Jumlah Bounding Box
1	Pisau	1230
2	Senjata tajam	1300
3	Pistol	1300
4	Senapan	1230

Hasil proses anotasi menghasilkan sebanyak 1.230 *bounding box* pada kelas pisau dan senapan, serta 1.300 *bounding box* pada kelas pistol dan senjata tajam. Hasil labeling dengan *bounding box* tersebut dapat melebihi jumlah gambar dikarenakan setiap gambar memiliki lebih dari 1 objek berbahaya. Gambar yang telah melalui proses labeling selanjutnya digunakan pada tahap *preprocessing* dan pelatihan model deteksi objek.

2.3. Preprocessing Data

Seluruh gambar yang telah melalui proses anotasi kemudian masuk ke dalam tahap *preprocessing* data. Proses ini bertujuan untuk meningkatkan kualitas dataset, menyesuaikan format data dengan kebutuhan model, serta mendukung peningkatan performa deteksi objek. *Preprocessing* data dilakukan menggunakan platform Roboflow melalui beberapa tahapan utama, yaitu pembagian dataset (*dataset splitting*) dan *resize* gambar.

1. Data Splitting

Dataset splitting dilakukan dengan membagi seluruh hasil anotasi gambar menjadi *data training* sebesar 80%, *validation* sebesar 10%, dan *testing* sebesar 10%. Pembagian ini secara otomatis dilakukan melalui platform Roboflow setelah labeling atau anotasi selesai dilakukan. *Splitting* dataset menghasilkan total *train set* dari seluruh kelas kategori sebanyak 3.589 gambar, *valid set* 448 gambar dan *test set* 449 gambar. Jumlah ketiga *subset* (4.486 gambar) sama dengan jumlah gambar teranotasi, sedangkan 514 gambar yang dikeluarkan karena objeknya terpotong tepi *frame* (Bagian 2.1) tidak diikutkan dalam pembagian. *Frame* diekstraksi pada laju 10 *frame* per detik sehingga *frame* berurutan dari klip yang sama sangat mirip. Pembagian dilakukan secara acak per gambar sehingga *frame* yang mirip dapat berada pada subset latih, validasi, dan uji sekaligus. Hasil pada data uji berpotensi lebih optimistik dibandingkan performa pada video yang belum pernah dilihat model. Distribusi *bounding box* untuk setiap kelas pada masing-masing subset disajikan pada Tabel 3.

Tabel 3. Distribusi Bounding Box Tiap Kelas

Nama Kelas	Total Bounding Box	Data Training	Data Validation	Data Test
Pisau	1230	978	128	124
Senjata Tajam	1300	1040	137	123
Pistol	1300	1050	121	129
Senapan	1230	978	127	125
Total	5060	4046	513	501

2. Resize Gambar

Setelah proses pembagian dataset, seluruh gambar diubah ukurannya menjadi 640×640 piksel agar sesuai dengan ukuran input model YOLO26. Proses *resize* dilakukan menggunakan platform Roboflow sebelum dataset digunakan pada tahap pelatihan model.

2.4. Arsitektur Model YOLO26

Penelitian ini menggunakan YOLO26 sebagai algoritma deteksi objek utama. YOLO26 merupakan model terbaru dari keluarga YOLO yang dikembangkan oleh Ultralytics dan dirilis pada Februari 2026. Secara fundamental, YOLO bekerja berdasarkan pendekatan *single-stage detection*, yaitu memproses seluruh gambar dalam satu kali *forward pass* melalui jaringan saraf untuk menghasilkan prediksi kelas dan lokasi objek secara simultan. Pendekatan ini berbeda dari metode *two-stage detection* seperti Faster R-CNN yang memerlukan tahap proposal wilayah terpisah [23], sehingga YOLO unggul dalam kecepatan inferensi tanpa mengorbankan akurasi secara signifikan.

YOLO26 dipilih karena menawarkan perbaikan arsitektural yang mengatasi keterbatasan versi-versi sebelumnya. Versi YOLOv8 hingga YOLOv10 menggunakan Distribution Focal Loss (DFL), sedangkan YOLOv8 dan YOLOv9 masih memerlukan NMS pada tahap pascaproses, dan YOLOv10 telah menghapus NMS [11]. YOLO26 menghapus DFL dan menghadirkan arsitektur end-to-end yang sepenuhnya bebas NMS [22]. Model ini mengganti DFL dengan mekanisme ProgLoss dan Small-Target-Aware Label Assignment (STAL) yang secara spesifik dirancang untuk meningkatkan akurasi pada objek berukuran kecil. YOLO26 juga memperkenalkan optimizer MuSGD sebagai optimizer default yang dirancang untuk menghasilkan konvergensi yang lebih stabil [22]. Kombinasi inovasi

arsitektural tersebut menjadikan YOLO26 relevan secara arsitektural untuk deteksi objek berbahaya berukuran relatif kecil pada *frame* video animasi kartun anak yang memiliki karakteristik visual unik, sedangkan kecepatan pemrosesan pada aplikasi dilaporkan pada Bagian 3.5.

2.5. Pelatihan Model

Pelatihan model dilakukan menggunakan pendekatan transfer learning dengan memanfaatkan bobot pra-latih dari dataset COCO. Pendekatan ini bertujuan mempercepat proses konvergensi dan meningkatkan kemampuan model dalam mengenali karakteristik objek pada dataset penelitian [18].

Tabel 4. Konfigurasi Hiperparameter Pelatihan Model

Hiperparameter	Nilai
Arsitektur Model	YOLO26
Epoch	100
Batch Size	16
Learning Rate Awal	0,001
Optimizer	AdamW
Ukuran Gambar Masukan	640 x 640 pixel
Transfer Learning	Bobot pra-latih COCO
Patience (Early Stopping)	15 epoch
Lingkungan Komputasi	Google Colab (GPU T4)

Model YOLO26 dilatih selama maksimal 100 *epoch* dengan mekanisme *early stopping* (*patience*=15), menggunakan ukuran citra 640×640 piksel dan batch size 16 pada lingkungan *Google Colab* dengan GPU T4. Proses pelatihan memanfaatkan transfer learning menggunakan bobot pra-latih COCO sebagai titik awal pelatihan pada dataset animasi kartun anak. Model terbaik dipilih berdasarkan nilai fitness Ultralytics (kombinasi berbobot mAP@0.5 dan mAP@0.5:0.95).

Proses pelatihan model dikonfigurasi menggunakan *optimizer* AdamW dengan *learning rate* awal (*lr0*) 0,001, *learning rate* akhir (*lrf*) 0,01 dari nilai awal, *weight decay* 0,0005, serta *warmup* selama 3 *epoch* pertama untuk menstabilkan pembelajaran awal. Batas maksimal pelatihan ditetapkan sebanyak 100 *epoch*, sejalan dengan rentang umum *fine-tuning* model YOLO pada dataset berskala kecil-menengah [24]. Sebagai langkah mitigasi *overfitting* dan optimalisasi efisiensi komputasi, mekanisme *early stopping* diterapkan dengan nilai *patience* 15 *epoch*. Pada praktiknya, proses pelatihan berjalan penuh hingga *epoch* ke-100 tanpa memicu penghentian dini. Hal ini mengindikasikan bahwa performa model pada data validasi tidak mengalami stagnasi lebih dari 15 *epoch* berturut-turut, yang juga diperkuat oleh tren penurunan *loss* yang relatif stabil pada hasil pelatihan (lihat Bagian 3.1).

2.6. Evaluasi Model

Evaluasi model dilakukan menggunakan data testing yang terpisah dari data training dan validation. Kinerja model diukur menggunakan metrik *Precision*, *Recall*, *F1-Score*, mAP@0.5, dan mAP@0.5:0.95. Sebuah prediksi dihitung sebagai *true positive* (TP) apabila kelasnya benar dan nilai *Intersection over Union* (IoU) dengan *bounding box* sebenarnya mencapai ambang yang ditetapkan. *False positive* (FP) adalah prediksi yang tidak memenuhi kondisi tersebut, dan *false negative* (FN) adalah objek sebenarnya yang tidak terdeteksi. Evaluasi menggunakan ambang IoU 0,5 untuk mAP@0.5 dan rentang 0,5 sampai 0,95 dengan kenaikan 0,05 untuk mAP@0.5:0.95, dengan ambang *confidence* 0,001 pada proses validasi. Nilai *Precision* dan *Recall* dilaporkan pada ambang *confidence* yang memberikan *F1-Score* tertinggi. *Precision* digunakan untuk mengukur ketepatan prediksi objek, sedangkan *Recall* mengukur kemampuan model dalam menemukan seluruh objek target. Selain itu, mAP digunakan untuk mengevaluasi akurasi deteksi secara keseluruhan pada berbagai tingkat *Intersection over Union* (IoU).

IoU dihitung dengan Persamaan (1):

$$IoU = \frac{Area_{overlap}}{Area_{union}} \quad (1)$$

Precision dihitung menggunakan Persamaan (2) [25]:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall dihitung dengan Persamaan (3) [25]:

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

F1-score dihitung dengan Persamaan (4):

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

Average Precision (AP) adalah luas area di bawah kurva *Precision-Recall* untuk satu kelas, dihitung dengan Persamaan (5), dan mAP adalah rata-rata AP seluruh kelas dengan Persamaan (6), dengan N adalah jumlah kelas:

$$AP = \int_0^1 P(R) dR \quad (5) \quad mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (6)$$

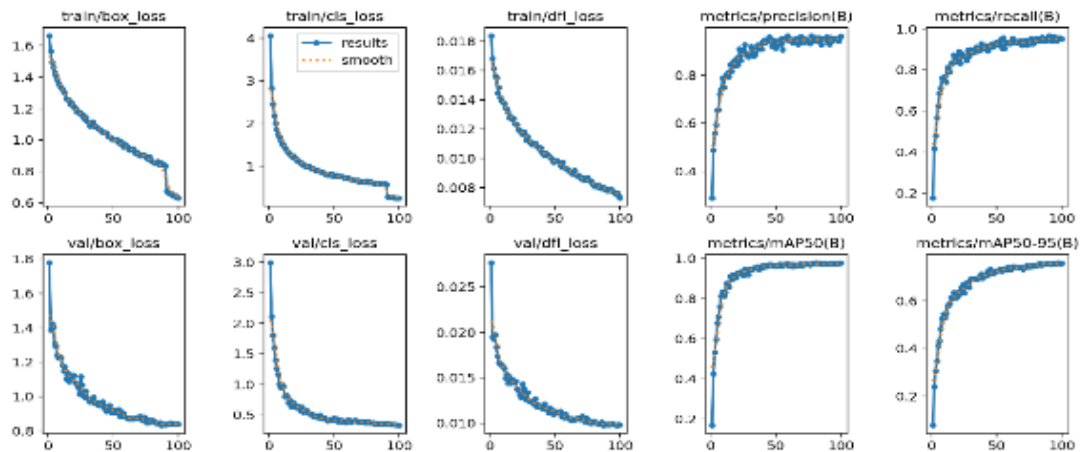
2.7. Deployment

Model YOLO26 terbaik hasil pelatihan diimplementasikan ke dalam aplikasi berbasis web menggunakan *framework Flask*. Sistem menerima masukan berupa tautan video yang merujuk ke repositori video pada aplikasi itu sendiri, sehingga tidak diperlukan proses pengunduhan dari platform eksternal, kemudian melakukan proses inferensi untuk mendeteksi objek berbahaya secara otomatis dan menghasilkan keluaran berupa video yang telah dianotasi dengan *bounding box*, label objek, serta nilai *confidence score*. Validasi implementasi dilakukan menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan. Pengujian difokuskan pada kesesuaian antara masukan dan keluaran sistem tanpa meninjau logika *internal program*.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Pelatihan Model YOLO26

Untuk menilai performa model selama proses pelatihan, dilakukan analisis melalui visualisasi grafik nilai *loss* dan metrik evaluasi pada data latih (*training*) maupun data validasi (*validation*). Grafik tersebut menampilkan beberapa komponen utama, yaitu *box loss*, *classification loss* (cls loss), dan *distribution focal loss* (dfl loss), serta metrik evaluasi berupa *precision*, *recall*, mAP@0.5, dan mAP@0.5:0.95. Melalui visualisasi ini, perkembangan performa model pada setiap epoch dapat diamati, termasuk untuk mengidentifikasi apakah model mengalami *overfitting* maupun *underfitting*. Gambar 2 berikut menunjukkan hasil pelatihan model selama 100 epoch.



Gambar. 2. Grafik proses *training* model

Berdasarkan grafik hasil pelatihan model, nilai *train box loss*, *train cls loss*, dan *train dfl loss* menunjukkan tren menurun secara konsisten dari awal hingga akhir *epoch*, yang menandakan bahwa model semakin mampu mempelajari pola objek pada data latih dengan tingkat kesalahan yang semakin kecil[25]. Hal serupa juga terlihat pada *validation loss* (val box loss, val cls loss, dan val dfl loss) yang mengalami penurunan stabil, menunjukkan bahwa model memiliki kemampuan generalisasi yang baik terhadap data validasi dan tidak mengalami *overfitting* yang signifikan.

3.2. Evaluasi Model

Evaluasi model dilakukan menggunakan data testing untuk mengukur kemampuan model dalam mendeteksi objek pada data yang belum pernah digunakan selama pelatihan. Kinerja model dianalisis menggunakan metrik *Precision*, *Recall*, *mAP@0.5*, *mAP@0.5:0.95*, dan *F1-Score* serta didukung oleh analisis *confusion matrix*.

1. Hasil Pengujian Data Testing

Pengujian dilakukan menggunakan model terbaik hasil pelatihan berdasarkan nilai fitness Ultralytics (kombinasi berbobot *mAP@0.5* dan *mAP@0.5:0.95*). Hasil pengujian model pada data testing disajikan pada Tabel 5

Tabel 5. Hasil Pengujian Model

Class	Images	Instance	Precision(P)	Recall(R)	F1-Score	mAP50	mAP50-95
All	449	501	0,945	0,938	0,941	0,953	0,763
Pisau	121	124	0,930	0,958	0,944	0,978	0,785
Senjata Tajam	113	123	0,908	0,886	0,897	0,885	0,695
Pistol	94	129	0,984	0,939	0,961	0,971	0,771
Senapan	122	125	0,960	0,971	0,966	0,979	0,802

Berdasarkan hasil pengujian pada 449 gambar dengan total 501 objek, model memperoleh nilai *Precision* sebesar 0,945, *Recall* sebesar 0,938, *F1-Score* sebesar 0,941, *mAP@0.5* sebesar 0,953, dan *mAP@0.5:0.95* sebesar 0,763. Jumlah gambar per kelas pada Tabel 5 berjumlah 450, lebih satu dari total 449 gambar uji, karena kolom *Images* menghitung gambar yang memuat kelas tersebut, sehingga gambar yang memuat objek dari dua kelas dihitung pada kedua kelas. Hasil tersebut menunjukkan bahwa model mampu mendeteksi objek berbahaya dengan tingkat akurasi yang tinggi pada data testing. Berdasarkan evaluasi per kelas, kategori senapan menunjukkan performa terbaik dengan nilai *mAP@0.5:0.95* sebesar 0,802, sedangkan kategori senjata tajam memperoleh nilai terendah sebesar 0,695 dan juga memiliki jumlah *false negative* tertinggi pada *confusion matrix* (lihat Gambar 3).

Perbedaan performa antar kelas menunjukkan bahwa karakteristik visual objek masih memengaruhi kemampuan model dalam melakukan deteksi secara konsisten. Namun secara keseluruhan, seluruh kelas berhasil mencapai nilai $mAP@0.5$ di atas 0,88 sehingga model dapat dikatakan memiliki performa deteksi yang baik. Selisih antara $mAP@0.5$ (0,953) dan $mAP@0.5:0.95$ (0,763) menunjukkan bahwa objek umumnya ditemukan, tetapi ketepatan kotak pembatas menurun pada ambang IoU yang lebih ketat. Hal ini diduga berkaitan dengan bentuk objek animasi yang disederhanakan atau digambar pada sudut yang tidak lazim, tetapi dugaan tersebut belum diuji secara khusus.

Untuk memperoleh gambaran performa model secara lebih objektif, dilakukan eksperimen tambahan dengan melatih model pembandingan pada dataset yang identik dengan penelitian ini, sehingga perbedaan performa yang teramati dapat diatribusikan secara langsung pada perbedaan arsitektur, bukan pada variasi dataset atau kondisi pengujian.

Tabel 6. Perbandingan Performa YOLO26 dan YOLOv8 pada Dataset yang Sama

Model	Kelas	Precision	Recall	F1-Score	$mAP@0.5$	$mAP@0.5:0.95$
YOLO26	All	0,945	0,938	0,941	0,953	0,763
YOLO26	Pisau	0,930	0,958	0,944	0,978	0,785
YOLO26	Senjata Tajam	0,908	0,886	0,897	0,885	0,695
YOLO26	Pistol	0,984	0,939	0,961	0,971	0,771
YOLO26	Senapan	0,960	0,971	0,966	0,979	0,802
YOLOv8	All	0,970	0,940	0,955	0,941	0,754
YOLOv8	Pisau	0,976	0,966	0,971	0,966	0,773
YOLOv8	Senjata Tajam	0,938	0,870	0,903	0,861	0,669
YOLOv8	Pistol	0,992	0,955	0,973	0,964	0,768
YOLOv8	Senapan	0,976	0,971	0,973	0,972	0,803

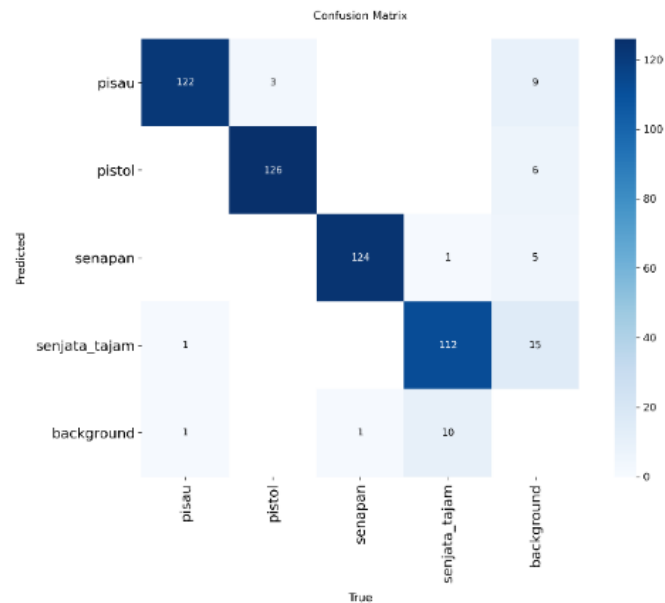
Berdasarkan Tabel 6, secara keseluruhan performa kedua model relatif setara. YOLOv8 unggul pada metrik *Precision* (0,970 vs 0,945), dan *F1-Score* (0,955 vs 0,941), sedangkan YOLO26 sedikit lebih baik pada $mAP@0.5$ (0,953 vs 0,941) dan $mAP@0.5:0.95$ (0,763 vs 0,754). *Recall* kedua model hampir sama (0,940 pada YOLOv8 dan 0,938 pada YOLO26). Apabila ditelaah per kelas, YOLO26 memperoleh $mAP@0.5$ yang lebih tinggi pada keempat kelas, dengan selisih terbesar pada kelas senjata tajam (0,885 berbanding 0,861). Pada $mAP@0.5:0.95$, YOLO26 lebih tinggi pada kelas pisau, senjata tajam, dan pistol, sedangkan pada kelas senapan keduanya hampir sama (0,802 berbanding 0,803). YOLOv8 memperoleh *Precision* dan *F1-Score* yang lebih tinggi pada seluruh kelas. Temuan yang menarik adalah senjata tajam tetap menjadi kelas dengan performa terendah pada kedua model ($mAP@0.5:0.95$ sebesar 0,695 pada YOLO26 dan 0,669 pada YOLOv8). Hal ini sejalan dengan analisis *confusion matrix* YOLOv8 yang mencatat *false negative* tertinggi pada kelas ini (13 dari 123 objek) disertai 17 *false positive* pada kelas yang sama. Pola yang sama terlihat pada YOLO26 (10 dari 123 objek, Gambar 3). Konsistensi kesulitan ini mengindikasikan bahwa tantangan deteksi diduga lebih dipengaruhi oleh karakteristik visual dataset domain animasi seperti variasi rasio aspek dan sudut pandang dibandingkan keterbatasan salah satu dari kedua arsitektur yang diuji.

Perbandingan langsung pada dataset yang sama ini tidak menunjukkan keunggulan salah satu model pada seluruh metrik. Selisih mAP antarmodel kecil (maksimal 0,026 pada tingkat kelas) dan berasal dari satu kali pelatihan per model, sehingga keunggulan salah satu model tidak disimpulkan signifikan. Pada subset uji dengan GPU Tesla T4, waktu inferensi tercatat 5,8 ms per gambar untuk YOLO26 dan 10,1 ms untuk YOLOv8, dengan waktu pascaproses 0,4 ms dan 1,6 ms. Perbedaan ini belum dapat diatribusikan sepenuhnya pada arsitektur karena ukuran kedua model berbeda (YOLO26n dengan 2,4 juta parameter dan YOLOv8s dengan 11,1 juta parameter) dan pengukuran dilakukan pada mode validasi, bukan pada aliran video. Pemilihan YOLO26 pada penelitian ini didasarkan pada

rancangan arsitekturnya (proses end-to-end tanpa NMS) sebagaimana diuraikan pada bagian metode penelitian, sedangkan keunggulan arsitektural tersebut belum dapat disimpulkan dari perbandingan ini.

2. Analisis Metrik Evaluasi

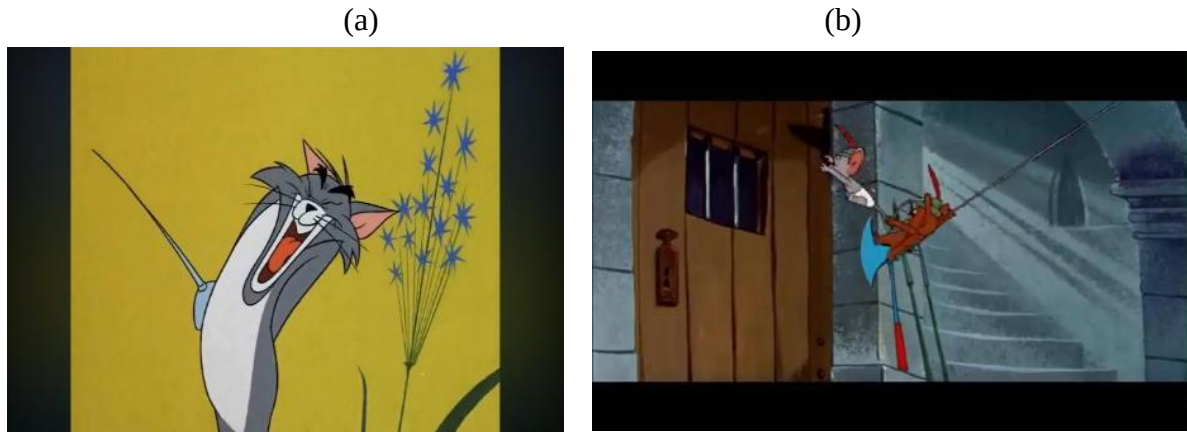
Confusion matrix digunakan untuk menganalisis hasil deteksi model secara lebih rinci dengan membandingkan label aktual dan hasil prediksi pada setiap kelas objek. Pada matriks ini, nilai yang berada pada diagonal utama menunjukkan jumlah prediksi benar (*true positive*), sedangkan nilai di luar diagonal menunjukkan adanya kesalahan klasifikasi antarkelas maupun objek yang tidak terdeteksi (*false negative*). Gambar 3 berikut menampilkan hasil *confusion matrix* model pada data pengujian.



Gambar. 3. *Confusion Matrix*

Berdasarkan *confusion matrix* pada Gambar 3, model berhasil mengklasifikasikan sebagian besar objek dengan benar. Kelas pisau menunjukkan 122 prediksi benar dengan 1 objek yang salah diklasifikasikan sebagai kelas senjata tajam dan 1 objek yang tidak terdeteksi (*background*). Kelas pistol memperoleh 126 prediksi benar tanpa objek yang tidak terdeteksi, tetapi dengan 3 objek yang salah diklasifikasikan sebagai kelas pisau. Kelas senapan menghasilkan 124 prediksi benar dengan 1 objek yang tidak terdeteksi. Kelas senjata tajam menunjukkan 112 prediksi benar dengan 1 objek yang salah diklasifikasikan sebagai kelas senapan dan 10 objek yang tidak terdeteksi, yang menunjukkan bahwa model masih mengalami kesulitan dalam mendeteksi beberapa objek pada kelas tersebut. Selain itu, terdapat sejumlah prediksi *false positive* yang berasal dari objek *background*, yaitu 9 pada kelas pisau, 6 pada kelas pistol, 5 pada kelas senapan, dan 15 pada kelas senjata tajam.

Jumlah *false negative* tertinggi pada kelas senjata tajam, yaitu 10 dari 123 objek, secara proporsional jauh lebih besar dibandingkan kelas lainnya. Kelas ini mencakup pedang dan kapak yang memiliki bilah dan tangkai lebih panjang dibandingkan pisau. Pada animasi kartun, objek memanjang sering ditampilkan miring, tergenggam sebagian, atau hanya terlihat ujungnya saat diayunkan (contoh pada Gambar 4), sehingga area piksel objek diduga menjadi lebih kecil dan tidak proporsional. Kemiripan visual antarkelas juga masih terjadi pada kondisi tertentu, yaitu satu pisau diprediksi sebagai senjata tajam dan satu senjata tajam diprediksi sebagai senapan. Kombinasi faktor tersebut diduga berkontribusi terhadap tingginya *false negative*, sejalan dengan keterbatasan umum model YOLO pada objek kecil atau berasio aspek ekstrem yang menjadi salah satu motivasi STAL pada YOLO26. Kontribusi STAL terhadap hasil ini belum dapat dipastikan karena studi ablasi belum dilakukan. Kelas ini juga menghasilkan *false positive* dari *background* terbanyak (15), dan penyebabnya belum dianalisis secara visual.



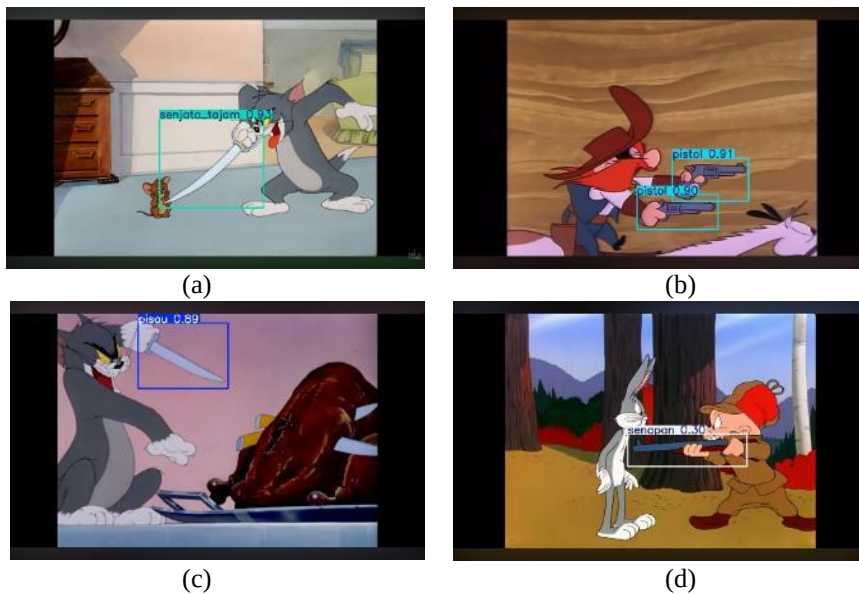
Gambar 4. Contoh Kasus False Negative pada Kelas Senjata Tajam, (a) Pedang Degen yang Tidak Terdeteksi, (b) Kapak yang Tidak Terdeteksi pada Posisi Miring dan Tertutup Sebagian

Gambar 4 mengilustrasikan dua kasus *false negative* pada kelas senjata tajam. Pedang degen pada Gambar 4(a) tidak terdeteksi; bilahnya sangat tipis dan berposisi diagonal sehingga rasio aspeknya ekstrem dan area pikselnya kecil terhadap keseluruhan *frame*. Kapak pada Gambar 4(b) tidak terdeteksi; objek miring dan sebagian tertutup tali serta elemen latar belakang sehingga bentuk utuhnya tidak jelas. Kedua contoh ini sejalan dengan dugaan bahwa variasi sudut pandang, oklusi, dan rasio aspek ekstrem pada penggambaran animasi berkontribusi menyulitkan deteksi kelas senjata tajam, tetapi dua contoh belum cukup untuk menyimpulkan penyebab utamanya.

Performa yang tinggi pada domain animasi ini sejalan dengan teori *transfer learning*, yaitu fitur tingkat rendah seperti tepi, sudut, dan kontur dasar relatif dapat ditransfer antar-domain, sedangkan fitur tingkat tinggi yang spesifik terhadap tekstur dan warna memerlukan penyesuaian pada data domain target [19]. Citra animasi umumnya hanya mempertahankan struktur kasar dan warna objek dengan tekstur minim [13], [14], sehingga *fine-tuning* pada dataset animasi diduga berperan dalam menyesuaikan fitur dari bobot COCO terhadap domain ini [12]. Penelitian ini tidak membandingkan bobot COCO dengan pelatihan dari awal, sehingga kontribusi *transfer learning* belum dapat dipisahkan dari kontribusi dataset dan arsitektur. YOLO26 memiliki mekanisme *self-attention* pada blok C2PSA dan STAL yang secara teoretis dirancang untuk konteks global dan objek kecil, tetapi YOLOv8 yang tidak memiliki kedua komponen tersebut memperoleh performa yang relatif setara (Tabel 6). Kontribusi masing-masing komponen karena itu tidak dapat disimpulkan dari data ini dan memerlukan studi ablatasi.

3. Contoh Kualitatif Deteksi pada Data Uji Baru

Bagian ini menyajikan contoh kualitatif untuk mengilustrasikan keluaran model pada citra yang berada di luar data pelatihan maupun validasi. Pada tahap ini, model diuji menggunakan 4 citra baru dan menghasilkan keluaran berupa prediksi objek dalam bentuk *bounding box* beserta label kelas yang terdeteksi. Gambar 5 menyajikan contoh hasil pengujian model terhadap data uji baru dan Tabel 7 merangkum kesesuaian hasilnya.



Gambar. 5. Contoh gambar hasil uji validitas model, (a) hasil deteksi senjata tajam, (b) hasil deteksi pistol, (c) hasil deteksi pisau, (d) hasil deteksi senapan

Hasil pengujian terhadap keempat citra baru menunjukkan bahwa seluruh data uji memperoleh kategori valid, yaitu label dan posisi *bounding box* sesuai dengan kondisi objek sebenarnya pada setiap gambar (Tabel 7). Jumlah contoh yang terbatas membuat hasil ini bersifat ilustratif dan tidak dapat dijadikan bukti kuantitatif kemampuan generalisasi model. Evaluasi kuantitatif ditunjukkan oleh hasil pada data testing (Tabel 5).

Tabel 7. Contoh Hasil Deteksi pada Data Uji Baru

No.	Objek Aktual	Hasil Deteksi	Keterangan
1	Pisau	Pisau	Valid
2	Senjata Tajam	Senjata Tajam	Valid
3	Pistol	Pistol	Valid
4	Senapan	Senapan	Valid

3.3. Deployment

Model YOLO26 yang telah dilatih berhasil diimplementasikan ke dalam aplikasi berbasis web menggunakan *framework* Flask. Sistem menerima tautan video dari repositori video aplikasi dan menghasilkan keluaran berupa video hasil deteksi yang dilengkapi *bounding box*, label objek, dan nilai *confidence score*. Kotak pembatas ditampilkan pada deteksi dengan *confidence* di atas 0,37, sedangkan tangkapan layar bukti disimpan pada *confidence* 0,77 ke atas, satu kali untuk setiap kelas yang terdeteksi. Aplikasi web yang dibangun terdiri dari dua halaman utama. Pertama, halaman *index* yang berfungsi sebagai halaman awal aplikasi dan menyediakan form masukan tautan video beserta tombol "Proses Video" untuk memulai proses deteksi. Kedua, halaman *result* yang menampilkan hasil deteksi berupa video yang telah diproses, *bounding box* pada objek berbahaya yang terdeteksi, label nama objek, kesimpulan deteksi, serta nilai *confidence score* sebagai tingkat keyakinan model terhadap hasil prediksi. Antarmuka aplikasi terdiri atas halaman input dan halaman hasil deteksi sebagaimana ditunjukkan pada Gambar 6. Kecepatan pemrosesan aplikasi dibahas pada Bagian 3.5.

3.4. Pengujian Fungsional Aplikasi (Black Box Testing)

Setelah aplikasi selesai dikembangkan, dilakukan pengujian fungsional menggunakan metode black box testing untuk memastikan seluruh fungsi sistem berjalan sesuai dengan kebutuhan yang telah dirancang. Pengujian ini mengevaluasi fungsi aplikasi, bukan akurasi model deteksi, sehingga hasilnya terpisah dari evaluasi performa model pada Bagian 3.2. Tabel 8 menyajikan hasil pengujian teknis aplikasi yang telah dilakukan.

Tabel 8. Black Box Testing

No.	Fitur yang di Uji	Langkah Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Submit link Video	User memasukkan link video valid lalu klik "Proses Video"	Sistem menerima link dan memproses video	Berhasil
2	Validasi input kosong	Pengguna menekan Proses Video tanpa memasukkan link	Sistem menampilkan pesan peringatan	Berhasil
3	Proses deteksi video	Sistem memproses video yang diinput	Video berhasil dianalisis oleh model YOLO26	Berhasil
4	Menampilkan hasil deteksi	Sistem menampilkan video, gambar screenshot, label dan Kesimpulan dari hasil deteksi	Sistem menampilkan video, gambar screenshot, label dan Kesimpulan dari hasil deteksi	Berhasil
5	<i>Bounding box</i> objek	Video mengandung objek berbahaya	Bounding box muncul pada objek yang terdeteksi	Berhasil
6	Label objek dan <i>Confidence Score</i>	Sistem mendeteksi objek	Nama objek berbahaya dan nilai <i>confidence score</i> tampil pada hasil deteksi	Berhasil
7	Video tanpa objek berbahaya	Input video tanpa objek berbahaya	Sistem tidak menampilkan <i>bounding box</i> dan menampilkan "benda berbahaya tidak terdeteksi"	Berhasil
8	Tombol Kembali atau input ulang	Pengguna ke halaman awal	Sistem kembali ke halaman index tanpa error	Berhasil

Berdasarkan hasil pengujian yang terdapat Tabel 8 di atas, diperoleh hasil bahwa seluruh fungsi sistem dapat berjalan sesuai dengan kebutuhan yang telah dirancang. Pengujian menunjukkan bahwa setiap fitur berhasil dieksekusi tanpa ditemukan kesalahan fungsional, sehingga dapat disimpulkan bahwa aplikasi telah berhasil diimplementasikan dan seluruh fungsinya berjalan sesuai rancangan. Implementasi sistem pada penelitian ini masih terbatas pada tahap purwarupa (*prototipe*). Sistem belum diuji maupun dioptimasi untuk menangani beban kerja dalam lingkungan skala produksi sehingga diperlukan pengembangan lebih lanjut sebelum dapat diterapkan pada penggunaan yang lebih luas.

3.5. Pengukuran Kecepatan Pemrosesan Aplikasi

Kecepatan pemrosesan aplikasi diukur pada tiga video berdurasi sekitar 40 detik (30 FPS) dari repositori aplikasi, menggunakan GPU NVIDIA GeForce GTX 1650 dengan model YOLO26n dan ukuran masukan 640 piksel, tanpa menghitung 10 *frame* awal. Setiap video diproses tiga kali (Tabel 9), dan pemrosesan dikategorikan setara real-time apabila FPS *end-to-end* tidak lebih rendah dari FPS video. Waktu model rata-rata berkisar 15,4 sampai 27,4 ms per *frame*, dan pemrosesan *end-to-end*

selesai dalam 23,1 sampai 39,2 detik atau setara 30,6 sampai 51,9 FPS, di atas laju video pada kesembilan pengukuran. Rata-rata FPS *end-to-end* per video adalah $39,2 \pm 11,1$ (Tom and Jerry 1), $32,9 \pm 3,5$ (Tom and Jerry 2), dan $44,4 \pm 3,6$ (Looney Tunes 1). Video Tom and Jerry 2 memiliki selisih terkecil dari laju video. Persentil ke-95 waktu model berkisar 21,6 sampai 57,3 ms, sehingga sebagian *frame* diproses lebih lambat daripada *interval frame* video (sekitar 33 ms). Hasil antar-pengulangan pada video yang sama dapat berbeda cukup besar (Tom and Jerry 1: 23,1 sampai 37,8 detik). Waktu baca *frame*, penggambaran kotak, dan penulisan video masing-masing di bawah 1 ms per *frame* pada seluruh pengukuran. Log inferensi per frame ditampilkan di terminal selama pengukuran. Satu pengukuran awal pada video Looney Tunes 1 dikeluarkan karena pencatatan waktu pada versi kode tersebut belum lengkap. Pengukuran ini dilakukan pada satu perangkat dan tiga video dari repositori yang sama, sehingga kemampuan pemrosesan real-time belum dapat disimpulkan secara umum.

Tabel 9. Kecepatan Pemrosesan Video pada Aplikasi dengan 3 Pengulangan

Video	Ulangan	Model ms/frame	Median/P95 (ms)	FPS model	End to end (detik)	FPS end to end
Tom and Jerry 1	1	15,4	14,4 / 21,6	64,8	23,1	51,9
Tom and Jerry 1	2	18,8	15,7 / 41,1	53,1	37,8	31,8
Tom and Jerry 1	3	25,1	17,1 / 50,8	39,8	35,5	33,8
Tom and Jerry 2	1	27,4	18,6 / 54,9	36,5	39,2	30,6
Tom and Jerry 2	2	21,7	16,1 / 51,9	46,1	32,5	36,9
Tom and Jerry 2	3	26,2	16,9 / 57,3	38,1	38,5	31,2
Looney Tunes 1	1	16,5	14,9 / 22,8	60,8	25,7	46,7
Looney Tunes 1	2	16,4	14,7 / 23,3	61,0	26,0	46,2
Looney Tunes 1	3	19,7	15,2 / 48,8	50,8	30,0	40,2

Hasil penelitian ini perlu dibaca dengan keterbatasan berikut. Dataset terbatas pada 4.486 gambar teranotasi yang bersumber dari 52 video pada 34 kanal dengan dominasi seri Tom and Jerry dan Looney Tunes, sehingga variasi gaya animasi, tokoh, dan latar yang tercakup terbatas. Data dibagi secara acak per gambar sehingga kemiripan antar-frame dari adegan yang sama tidak dapat dikesampingkan dan belum diukur. Performa pada animasi yang belum pernah dilihat model juga belum diketahui. Gambar dengan objek terpotong oleh tepi *frame* dikeluarkan dari dataset sehingga performa pada kondisi tersebut belum diketahui. Tingkat kesepakatan label antar-anotator tidak dihitung secara kuantitatif. Dataset gambar tidak dipublikasikan karena bersumber dari video berhak cipta, dan replikasi penuh dapat terbatas apabila video sumber dihapus atau dijadikan privat. Sistem hanya mendeteksi keberadaan objek dan belum mengenali aksi (*action recognition*), sehingga belum dapat membedakan objek yang digunakan secara berbahaya dari objek yang tidak, dan *false positive* dari background masih terjadi pada seluruh kelas. Studi ablasi komponen YOLO26 serta perbandingan dengan pelatihan dari awal juga belum dilakukan. Model dan aplikasi karena itu belum dapat dinyatakan siap digunakan secara luas tanpa evaluasi pada sumber animasi yang lebih beragam, pengukuran kecepatan pada aliran video langsung dan perangkat yang lebih beragam, dan pengujian efektivitas filtrasi konten secara *end-to-end*.

4. KESIMPULAN

4.1. Kesimpulan

Penelitian ini menghasilkan tiga kontribusi utama dalam upaya deteksi objek berbahaya pada video animasi kartun anak. Kontribusi pertama adalah dataset citra animasi kartun beranotasi khusus untuk objek berbahaya, terdiri dari 4.486 gambar dengan 5.060 bounding box dalam empat kelas, yaitu pisau, senjata tajam, pistol, dan senapan. Kontribusi kedua adalah model YOLO26 hasil fine-tuning dari bobot pra-latih COCO pada dataset tersebut, yang memperoleh *Precision* 0,945, *Recall* 0,938, *F1-Score* 0,941, *mAP@0.5* 0,953, dan *mAP@0.5:0.95* 0,763 pada 449 gambar uji dengan 501 objek. Kelas senjata tajam memiliki performa terendah (*mAP@0.5:0.95* sebesar 0,695), dan performa keseluruhan model setara dengan YOLOv8 pada dataset yang sama. Kontribusi ketiga adalah purwarupa aplikasi web yang seluruh fungsinya berjalan sesuai rancangan berdasarkan *Black Box Testing*.

Penelitian ini memiliki beberapa keterbatasan. Pengukuran awal pada satu GPU (GTX 1650) menunjukkan pemrosesan *end-to-end* berada di atas laju video pada kesembilan pengukuran (30,6 sampai 51,9 FPS pada video 30 FPS), tetapi dilakukan pada satu perangkat dan tiga video dari repositori yang sama, sehingga kemampuan *real-time* belum dapat disimpulkan secara umum. Sistem masih berupa purwarupa yang belum diuji pada beban skala produksi. Penelitian hanya mengevaluasi deteksi objek berbahaya, sehingga efektivitas filtrasi konten secara *end-to-end* belum dievaluasi. Dataset juga terbatas pada empat kelas objek dan gaya animasi yang dikumpulkan.

4.2. Saran

Berdasarkan temuan dan keterbatasan penelitian ini, beberapa rekomendasi untuk penelitian selanjutnya adalah sebagai berikut:

1. Mengembangkan deteksi tindakan atau aksi (*action recognition*), tidak hanya sebatas deteksi keberadaan objek berbahaya secara statis, mengingat tingkat ancaman suatu objek dalam konteks animasi turut ditentukan oleh aksi yang menyertainya, misalnya membedakan antara pisau yang sedang digunakan untuk memotong bahan makanan dengan pisau yang diarahkan untuk menyerang karakter lain.
2. Menggabungkan hasil deteksi objek dengan analisis naratif atau konteks adegan (*scene context analysis*) untuk mengurangi *false positive* pada objek yang secara visual menyerupai senjata namun tidak digunakan dalam konteks berbahaya.
3. Melakukan studi ablasi untuk mengukur kontribusi masing-masing komponen arsitektural YOLO26, seperti mekanisme C2PSA dan *Small-Target-Aware Label Assignment* (STAL), secara terpisah terhadap performa deteksi pada domain animasi.
4. Memperluas dataset dengan menambahkan variasi gaya ilustrasi animasi dan kategori objek berbahaya lain di luar empat kategori yang digunakan pada penelitian ini.
5. Mengukur efisiensi *real-time* pada aliran video secara langsung, meliputi FPS, waktu inferensi per *frame*, dan latensi, serta membandingkannya antarmodel pada perangkat keras yang sama.
6. Membandingkan *fine-tuning* dari bobot pra-latih COCO dengan pelatihan dari awal (*training from scratch*) untuk mengukur kontribusi transfer learning pada domain animasi.
7. Mengevaluasi efektivitas filtrasi konten secara *end-to-end* serta menguji dan mengoptimalkan purwarupa pada beban skala produksi.

DAFTAR PUSTAKA

- [1] P. T. Giraffe Insights, "PARK: Precise Advertiser Report - Kids.," Giraffe Insights & Precise TV, America (USA), Market Research Report, Jun 2022. [Daring]. Tersedia pada: https://content.precisify.com/hubfs/PARK_%20Precise%20Advertiser%20Report%20-%20Kids%20%20.pdf
- [2] M. B. R. Victoria Rideout, M.A. Ph. D., "The Common Sense census: Media use by kids age zero to eight, 2020," Common Sense Media, San Francisco, CA, 2020. [Daring]. Tersedia pada:

- https://www.common sense media.org/sites/default/files/research/report/2020_zero_to_eight_census_final_web.pdf
- [3] M. Gkolemi, P. Papadopoulos, E. P. Markatos, dan N. Kourtellis, “YouTubers Not madeForKids: Detecting Channels Sharing Inappropriate Videos Targeting Children,” dalam *14th ACM Web Science Conference 2022*, Jun 2022, hlm. 370–381. doi: 10.1145/3501247.3531556.
 - [4] Beverly Hills, California, America. *Tom and Jerry | Mega Compilation | Vol. 20 | Warner Classics | Sufferin’ Cats*, (1943). [Daring Video]. Tersedia pada: <https://www.youtube.com/watch?v=LNAEv9t5HWI>
 - [5] “Peraturan Komisi Penyiaran Indonesia Nomor 03 Tahun 2007 tentang Standar Program Siaran,” Komisi Penyiaran Indonesia, Jakarta, 2007. [Daring]. Tersedia pada: <https://www.kpi.go.id/download/regulasi/Standar%20Program%20Siaran.pdf>
 - [6] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, dan M. Parmar, “A review of convolutional neural networks in computer vision,” *Artif. Intell. Rev.*, vol. 57, no. 4, hlm. 99, Mar 2024, doi: 10.1007/s10462-024-10721-6.
 - [7] R. Sapkota dkk., “YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series,” *Artif. Intell. Rev.*, vol. 58, no. 9, hlm. 274, Jun 2025, doi: 10.1007/s10462-025-11253-3.
 - [8] W. Xu, D. Zhu, R. Deng, K. Yung, dan A. W. H. Ip, “Violence-YOLO: Enhanced GELAN Algorithm for Violence Detection,” *Appl. Sci.*, vol. 14, no. 15, hlm. 6712, Agu 2024, doi: 10.3390/app14156712.
 - [9] Y. Zeng, L. Zhang, J. Zhao, J. Lan, dan B. Li, “JRL-YOLO: A Novel Jump-Join Repetitious Learning Structure for Real-Time Dangerous Object Detection,” *Comput. Intell. Neurosci.*, vol. 2021, no. 1, hlm. 5536152, Jan 2021, doi: 10.1155/2021/5536152.
 - [10] G. Lavanya dan S. D. Pande, “Enhancing Real-time Object Detection with YOLO Algorithm,” *EAI Endorsed Trans. Internet Things*, vol. 10, Des 2023, doi: 10.4108/eetiot.4541.
 - [11] M. A. R. Alif dan M. Hussain, “YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain,” 14 Juni 2024, *arXiv*: arXiv:2406.10139. doi: 10.48550/arXiv.2406.10139.
 - [12] L. Jiao dkk., “A Survey of Deep Learning-Based Object Detection,” *IEEE Access*, vol. 7, hlm. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.
 - [13] X. Mao dkk., “COCO-O: A Benchmark for Object Detectors under Natural Distribution Shifts,” 2 Agustus 2023, *arXiv*: arXiv:2307.12730. doi: 10.48550/arXiv.2307.12730.
 - [14] D. Kadish, S. Risi, dan A. S. Løvlie, “Improving Object Detection in Art Images Using Only Style Transfer,” 4 Mei 2021, *arXiv*: arXiv:2102.06529. doi: 10.48550/arXiv.2102.06529.
 - [15] Y. Tang, “AnimeNet: A Deep Learning Approach for Detecting Violence and Eroticism in Animated Content,” *Comput. Mater. Contin.*, vol. 77, no. 1, hlm. 867–891, 2023, doi: 10.32604/cmc.2023.041550.
 - [16] T. Khalil, S. Jan, R. M. Ghoniem, dan M. I. K. Khalil, “Multi-CNN Fusion Framework for Predictive Violence Detection in Animated Media,” *Comput. Mater. Contin.*, vol. 86, no. 2, hlm. 1–20, 2026, doi: 10.32604/cmc.2025.072655.
 - [17] M. Taha, A. Alsammak, dan A. Zaky, “InspectorNet: Transformer network for violence detection in animated cartoon,” *Eng. Res. J. - Fac. Eng. Shoubra*, vol. 52, no. 2, hlm. 114–119, Apr 2023, doi: 10.21608/erjsh.2023.181713.1119.
 - [18] F. Zhuang dkk., “A Comprehensive Survey on Transfer Learning,” 23 Juni 2020, *arXiv*: arXiv:1911.02685. doi: 10.48550/arXiv.1911.02685.
 - [19] J. Yosinski, J. Clune, Y. Bengio, dan H. Lipson, “How transferable are features in deep neural networks?,” 6 November 2014, *arXiv*: arXiv:1411.1792. doi: 10.48550/arXiv.1411.1792.
 - [20] K. Yousaf dan T. Nawaz, “A Deep Learning-Based Approach for Inappropriate Content Detection and Classification of YouTube Videos,” *IEEE Access*, vol. 10, hlm. 16283–16298, 2022, doi: 10.1109/ACCESS.2022.3147519.
 - [21] K. Abhinand dkk., “Malayalam Sign Language Identification using Finetuned YOLOv8 and Computer Vision Techniques,” 8 Mei 2024, *arXiv*: arXiv:2405.06702. doi: 10.48550/arXiv.2405.06702.
 - [22] P. Hidayatullah dan R. Tubagus, “YOLO26: A Comprehensive Architecture Overview and Key Improvements,” 2026, doi: <https://doi.org/10.48550/arXiv.2602.14582>.

- [23] S. Ren, K. He, R. Girshick, dan J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," 6 Januari 2016, *arXiv*: arXiv:1506.01497. doi: 10.48550/arXiv.1506.01497.
- [24] Amresh Kumar, "Fine-tuning YOLO for Custom Object Detection: A Complete Guide," Medium. [Daring]. Tersedia pada: <https://medium.com/@amresh.kumar11/fine-tuning-yolo-for-custom-object-detection-a-complete-guide-6ce3724ce9f1>
- [25] Aston Zhang, Zachary C. Lipton, dan A. J. S. Mu Li, "Dive into Deep Learning," hlm. 1185, 2023, doi: arXiv:2106.11342.