

**PERBANDINGAN PERFORMA ANTARA *FASTER R-CNN* DAN
SINGLE SHOT DETECTOR DALAM DETEKSI AKTIVITAS
MEROKOK DI LINGKUNGAN KAMPUS**

LAPORAN TUGAS AKHIR

Proposal ini disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Strata 1 (S1) pada Program Studi Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang



DISUSUN OLEH:

NAZLA ABAY DAUD ISTANTO

NIM 32602100103

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2025

FINAL PROJECT

***PERFORMANCE COMPARISON BETWEEN FASTER R-CNN AND
SINGLE SHOT DETECTOR IN SMOKING ACTIVITY DETECTION IN
CAMPUS ENVIRONMENT***

*Proposed to complete the requirement to obtain a bachelor's degree (S-1)
at Informatics Engineering Departement of Industrial Technology Faculty
Sultan Agung Islamic University*



ARRANGED BY:

NAZLA ABAY DAUD ISTANTO

NIM 32602100103

**MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG**

2025

**LEMBAR PENGESAHAN
TUGAS AKHIR**

Perbandingan Performa Antara *Faster R-CNN* dan *Single Shot Detector* dalam
Deteksi Aktivitas Merokok di lingkungan Kampus

**NAZLA ABAY DAUD ISTANTO
NIM 32602100103**

Telah dipertahankan di depan tim penguji ujian sarjana tugas akhir
Program Studi Teknik Informatika
Universitas Islam Sultan Agung
Pada tanggal : 2 Juni 2025....

TIM PENGUJI UJIAN SARJANA :

Dedy Kurniadi, S.T., M.Kom

NIK. 210615048

(Ketua Penguji)

(17-06-2025)

Moch. Taufik, S.T., M.IT

NIK. 210604034

(Anggota Penguji)

(16-06-2025)

Bagus SWP, S.Kom., M.Cs

NIK. 210616051

(Pembimbing)

(18-06-2025)

Semarang, 2 Juni 2025

Mengetahui,

Kaprodi Teknik Informatika
Universitas Islam Sultan Agung

Moch. Taufik, S.T., M.IT

NIK. 210604034

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Nazla Abay Daud Istanto

NIM : 32602100103

Judul Tugas Akhir : Perbandingan Performa Antara *Faster R-CNN* dan *Single Shot Detector* dalam Deteksi Aktivitas Merokok di lingkungan Kampus

Bahwa dengan ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

Semarang, 20 Juni 2025

Yang menyatakan,



Nazla Abay Daud Istanto

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertanda tangan di bawah ini:

Nama : Nazla Abay Daud Istanto

NIM : 32602100103

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Alamat Asal : Pati

Dengan ini menyatakan karya ilmiah berupa tugas akhir dengan judul : Perbandingan Performa Antara Faster R-CNN dan Single Shot Detector dalam Deteksi Aktivitas Merokok di lingkungan Kampus.

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan hak bebas royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap menyantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila di kemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan agung.

Semarang, 20 Juni 2025

Yang menyatakan,



Nazla Abay Daud Istanto

KATA PENGANTAR

Dengan mengucapkan syukur alhamdulillah atas kehadiran ALLAH SWT yang telah memberikan rahmat dan karunia-Nya kepada penulis, sehingga dapat menyelesaikan Tugas Akhir dengan judul “Perbandingan Performa *Antara Faster R-CNN* dan *Single Shot Detector* dalam Deteksi Aktivitas Merokok di lingkungan Kampus” dengan baik.

Tugas akhir ini disusun dan dibuat dengan adanya bantuan dari berbagai pihak, berupa materi maupun teknis, oleh karena itu saya selaku penulis mengucapkan terima kasih kepada:

1. Rektor Universitas Islam Sultan Agung Bapak Prof. Dr. H. Gunarto, S.H., M.H yang mengizinkan penulis menimba ilmu di kampus ini.
2. Dekan Fakultas Teknologi Industri Ibu Dr. Ir. Hj. Novi Marlyana, S.T., M.T., IPU., ASEAN Eng.
3. Kaprodi Teknik Informatika Bapak Moch, Taufik, S.T., M.IT.
4. Dosen pembimbing penulis Bapak Bagus Satrio Waluyo Poetro, S.Kom., M.Cs yang telah meluangkan waktu dan memberi ilmu dalam penyusunan tugas akhir.
5. Orang tua penulis tercinta Bapak Joko Istanto dan Ibu Rina Fatkhiyah yang telah memberikan doa serta dorongan baik moral maupun material.
6. Saudara penulis Milka Khaidzar Tsani yang telah memberikan dukungan selama masa studi.
7. Seseorang spesial yang telah memberikan semangat kepada penulis dalam menyelesaikan tugas akhir.
8. Teman seperjuangan yang telah memberikan dukungan moral, motivasi, serta semangat dalam proses penyelesaian tugas akhir.
9. Dan kepada semua pihak yang tidak dapat disebutkan satu persatu.

Semarang, 20 Juni 2025

Penulis

DAFTAR ISI

LAPORAN TUGAS AKHIR.....	i
LEMBAR PENGESAHAN TUGAS AKHIR	ii
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR.....	iii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
ABSTRAK	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat.....	3
1.6 Sistematika Penelitian	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori	9
2.2.1 Rokok.....	9
2.2.2 <i>Computer Vision</i>	9
2.2.3 <i>Object Detection</i>	10
2.2.4 <i>Faster regional convolutional neural network</i>	10
2.2.5 <i>Regional Proposal Network</i>	12
2.2.6 <i>Single Shot Detector</i>	13
BAB III METODE PENELITIAN	15
3.1 Studi Literatur.....	15
3.2 Pengumpulan <i>Dataset</i>	15
3.3 Arsitektur Model	16
3.3.1 <i>Faster R-CNN</i>	16

3.3.2	<i>Single Shot Detector</i>	17
3.4	Analisa Kebutuhan Fungsional	18
3.5	Analisa Kebutuhan Non-Fungsional	18
3.6	<i>Flowchart</i> Metode	20
3.7	Alur Sistem.....	21
3.8	Evaluasi dan analisis.....	22
BAB IV HASIL DAN ANALISIS PENELITIAN.....		23
4.1	Hasil.....	23
4.2	Analisa.....	29
BAB V KESIMPULAN DAN SARAN		50
5.1	Kesimpulan.....	50
5.2	Saran.....	52
DAFTAR PUSTAKA.....		53



DAFTAR TABEL

Tabel 1. 1 Sistematika penulisan.....	4
Tabel 4. 1 <i>Hyperparameter</i>	29
Tabel 4. 2 Hasil pengujian <i>faster r-cnn</i>	30
Tabel 4. 3 Hasil evaluasi <i>faster r-cnn</i>	38
Tabel 4. 4 Hasil pengujian <i>single shot detector</i>	40
Tabel 4. 5 Hasil evaluasi <i>single shot detector</i>	47



DAFTAR GAMBAR

Gambar 2. 1 Rokok	9
Gambar 2. 2 Arsitektur <i>faster r-cnn</i> (Charli dkk. 2020)	11
Gambar 2. 3 Arsitektur <i>SSD</i> (Cheng dkk. 2022)	13
Gambar 3. 1 Alur penelitian	15
Gambar 3. 2 <i>Dataset</i>	16
Gambar 3. 3 Alur <i>flowchart</i> model	20
Gambar 3. 4 Alur sistem	21
Gambar 4. 1 <i>Folder</i> awal	23
Gambar 4. 2 Gambar dan anotasi dalam satu <i>folder</i>	23
Gambar 4. 3 <i>Folder</i> baru	24
Gambar 4. 4 Pemanggilan dataset dan <i>unzip</i>	24
Gambar 4. 5 Label mapping	24
Gambar 4. 6 jumlah class pada model	25
Gambar 4. 7 source code <i>augmentasi</i>	25
Gambar 4. 8 Hasil <i>augmentasi</i>	25
Gambar 4. 9 Model <i>Faster Rcnn</i>	27
Gambar 4. 10 Model <i>SSD</i>	27
Gambar 4. 11 Grafik <i>loss training</i> dan <i>validation faster r-cnn</i>	27
Gambar 4. 12 Grafik <i>loss training</i> dan <i>validation single shot detector</i>	27
Gambar 4. 13 Tampilan mendeteksi rokok dengan model <i>faster r-cnn</i>	28
Gambar 4. 14 Tampilan mendeteksi rokok dengan model <i>ssd</i>	28

ABSTRAK

Penerapan pengaturan kawasan tanpa rokok di lingkungan kampus menghadapi tantangan dalam hal pengawasan dan penegakan. Penelitian ini bertujuan untuk membandingkan performa dua algoritma deteksi objek berbasis *deep learning*, yaitu *Faster R-CNN* dan *Single Shot Detector (SSD)*, dalam mendeteksi aktivitas merokok di area kampus Universitas Islam Sultan Agung (UNISSULA). Dataset yang digunakan terdiri dari 1935 gambar beranotasi aktivitas merokok yang diperoleh dari *Roboflow*, dengan pembagian data 85% pelatihan dan 15% validasi. Model dilatih menggunakan *Google Colab* dan diuji berdasarkan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil mengetahui bahwa *Faster R-CNN* mempunyai performa yang lebih unggul dengan nilai evaluasi terbaik mencapai 100% pada threshold 0.5, sedangkan SSD menunjukkan performa yang lebih rendah dengan akurasi maksimal sebesar 42,5% pada threshold yang sama. Temuan ini menyimpulkan bahwa *Faster R-CNN* lebih cocok untuk digunakan dalam sistem pendeteksian aktivitas merokok di lingkungan kampus dibandingkan SSD, terutama dalam konteks akurasi dan konsistensi deteksi.

Kata Kunci: Deteksi Merokok, *Faster R-CNN*, *Single Shot Detector*, *Computer Vision*, *Deep Learning*, Kampus Bebas Rokok

ABSTRACT

The implementation of no-smoking policies in campus environments faces challenges in monitoring and enforcement. This study aims to compare the performance of two deep learning-based object detection algorithms—*Faster R-CNN* and *Single Shot Detector (SSD)*—in detecting smoking activities on the campus of Sultan Agung Islamic University (UNISSULA). The dataset used consists of 1,935 annotated images of smoking activities obtained from *Roboflow*, with an 85% training and 15% validation split. The models were trained using *Google Colab* and evaluated using metrics such as *accuracy*, *precision*, *recall*, and *F1-score*. The results showed that *Faster R-CNN* outperformed *SSD*, achieving perfect evaluation metrics (100%) at a threshold of 0.5, whereas *SSD* reached a maximum accuracy of 42.5% at the same threshold. These findings suggest that *Faster R-CNN* is more suitable for implementing an automatic smoking detection system on campus, especially in terms of accuracy and detection consistency.

Keywords: Smoking Detection, *Faster R-CNN*, *Single Shot Detector*, *Computer Vision*, *Deep Learning*, Smoke-Free Campus

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebijakan larangan merokok diterapkan di berbagai kampus untuk membuat lingkungan yang sehat dan nyaman bagi seluruh civitas akademik. Banyak tulisan reklame berisi larangan merokok, baik untuk mahasiswa, karyawan, dan juga dosen saat berada di lingkungan UNISSULA (Sudarto 2020) dan sesuai surat edaran Rektor Nomor : 930/YBW-SA/IX/2024 tentang larangan merokok (UNISSULA 2024). Namun, kebijakan ini sering kali menemui kendala dalam hal pengawasan dan penegakan aturan karena kurangnya pengawasan yang memadai, keterbatasan petugas keamanan, serta ketidaksadaran terhadap dampak negatif dari kebiasaan tersebut terhadap kesehatan dan lingkungan.

Merokok di tempat umum, khususnya di lingkungan kampus, dapat menyebabkan berbagai dampak buruk, mengingat asap rokok mengandung campuran daun tembakau kering yang dicacah dan dicampur dengan bahan-bahan lain, seperti perekat dan senyawa kimia yang memiliki dampak buruk kesehatan perokok, termasuk perokok pasif yang tidak sengaja menghisap asap rokok (Ramadhani dkk. 2023). Selain itu, sampah puntung rokok yang dibuang sembarangan juga menambah masalah kebersihan di kampus.

Dengan laju perubahan teknologi yang semakin pesat dalam kehidupan masyarakat serta kemampuan komputasi komputer yang semakin tinggi. Salah satu area teknologi yang sedang naik daun saat ini adalah *machine learning* yang mana merupakan bagian dari ilmu kecerdasan buatan atau yang dikenal sebagai *artificial intelligence (AI)*. Salah satu cabang dari *AI* adalah *deep learning* (Ramadhanu dkk. 2023) dan dapat melakukan pembelajaran yang lebih mendalam sehingga menjadikan *deep learning* sebagai inti dari kemajuan *AI* serta memungkinkan untuk mendeteksi objek. Di bidang *deep learning*, menjadi populer dan metode ini tidak mengubah kualitas gambar tetapi mengubah dimensi. deteksi tambahan memakai *CNN* dengan metode

pembelajaran transfer. Keunggulan metode ini yaitu kemampuan untuk melakukan pra-pelatihan menggunakan informasi yang dipelajari dari model *CNN* yang telah dilatih (Poetro dkk. 2024). Penelitian terkait arsitektur *deep learning* untuk deteksi objek sangat penting untuk menghasilkan struktur yang dapat melakukan deteksi objek dengan lebih cepat.

Teknologi pengenalan objek berbasis *deep learning* telah menunjukkan kemajuan yang signifikan dalam berbagai bidang, termasuk deteksi aktivitas manusia. Dua arsitektur yang digunakan dalam deteksi objek adalah *Faster R-CNN* dan *Single Shot Detector*. *Faster-RCNN* merupakan suatu sistem untuk mendeteksi objek yang memiliki dua bagian. Bagian pertama yaitu *Fast Region-based*, sedangkan bagian yang kedua adalah *Regional Convolutional Neural Network* (Pardede dan Hardiansah 2022). Sementara itu, Algoritma *SSD* adalah salah satu cara yang sangat populer karena kemampuannya dalam menemukan objek secara langsung dengan tingkat ketelitian yang tinggi. *SSD* merupakan algoritma untuk mendeteksi objek yang mengandalkan pembelajaran mendalam, yang memungkinkan identifikasi objek dengan menggunakan ciri-ciri visual yang diambil dari gambar (Puspitasari dkk. 2025)

Meskipun kedua metode ini telah terbukti berhasil dalam berbagai deteksi, perbandingan performa mereka dalam konteks deteksi aktivitas merokok di lingkungan Unissula terbatas. Oleh karena itu, penelitian ini bertujuan guna membandingkan performa antara *Faster R-CNN* dan *Single Shot Detector* dalam mendeteksi aktivitas merokok, dengan harapan dapat menciptakan wawasan lebih baik mengenai metode yang lebih cocok untuk diterapkan untuk pengawasan kesehatan di lingkungan Unissula.

1.2 Rumusan Masalah

1. Bagaimana performa algoritma *Faster R-CNN* dan *Single Shot Detector (SSD)* dalam mendeteksi aktivitas merokok di lingkungan kampus?
2. Algoritma mana yang lebih efektif dan akurat dalam mendeteksi aktivitas merokok?
3. Apa kelebihan dan kekurangan masing-masing algoritma dalam konteks implementasi sistem deteksi otomatis?

1.3 Batasan Masalah

1. Deteksi hanya difokuskan pada aktivitas merokok yang terlihat jelas, seperti memegang atau mengisap rokok.
2. lingkup area penelitian terbatas pada zona larangan merokok di lingkungan kampus UNISSULA, seperti koridor, ruang terbuka, dan sekitar gedung akademik.
3. Pengujian dilakukan pada kondisi pencahayaan normal (siang hari) dan tidak mencakup malam hari atau kondisi minim cahaya.

1.4 Tujuan Penelitian

1. Membangun dan menguji model deteksi aktivitas merokok menggunakan algoritma *Faster R-CNN* dan *SSD*.
2. Membandingkan performa kedua algoritma berdasarkan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*.
3. Menentukan algoritma yang paling sesuai untuk mendukung pengawasan kawasan bebas rokok di kampus.

1.5 Manfaat

1. Penelitian ini memperlihatkan urgensi penerapan teknologi AI dan *computer vision* sebagai sarana pendukung kebijakan institusi, khususnya dalam penegakan aturan di lingkungan kampus.
2. Temuan dari penelitian ini berpotensi menjadi solusi inovatif yang dapat dimanfaatkan oleh institusi kampus untuk meningkatkan efektivitas

pengawasan terhadap pelanggaran aturan. Menjadi inovasi yang bisa digunakan oleh pihak kampus dalam menegakkan aturan.

3. Hasil penelitian ini dapat menjadi rujukan baru dalam pengembangan literatur di bidang *computer vision*, khususnya terkait model deteksi perilaku manusia seperti aktivitas merokok menggunakan pendekatan *Faster R-CNN* dan *Single Shot Detector*.
4. Memberikan evaluasi performa dua metode deteksi objek yang populer sebagai acuan untuk memilih pendekatan paling efektif dalam konteks pemantauan aktivitas merokok di lingkungan kampus.
5. Mendorong upaya mewujudkan kawasan kampus yang lebih bersih, sehat, dan terbebas dari paparan asap rokok melalui pemanfaatan teknologi deteksi otomatis.

1.6 Sistematika Penelitian

Sistematika penulisan yang digunakan oleh penulis dalam sebuah pembuatan laporan tugas akhir adalah sebagai berikut:

Tabel 1. 1 Sistematika penulisan

BAB I	PENDAHULUAN
	Bagian bab berisi latar belakang penggunaan judul, rumusan masalah, batasan masalah, tujuan penelitian, manfaat, serta sistematika penulisan.
BAB II	TINJAUAN PUSTAKA DAN DASAR TEORI
	Bagian bab memuat penelitian-penelitian sebelumnya dan dasar teori untuk membantu penulis memahami bagaimana teori yang berhubungan dengan <i>Single Shot Detector</i> dan pengimplementasian menggunakan <i>webcam</i> .
BAB III	METODE PENELITIAN
	Bagian bab mengungkapkan proses tahapan-tahapan penelitian dimulai dari mendapatkan data hingga proses pengolahan data yang ada.

BAB IV	HASIL DAN ANALISIS PENELITIAN
	Bagian Bab penulis mengungkapkan hasil penelitian seperti pengumpulan data, pengolahan data, pengujian model deteksi aktivitas merokok, dan juga perhitungan akurasi dari evaluasi model yang diterapkan
BAB V	KESIMPULAN DAN SARAN
	Bagian bab penulis memaparkan kesimpulan proses penelitian dari awal hingga akhir serta saran untuk penelitian selanjutnya.



BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada penelitian terdahulu sistem dibangun menggunakan metode *Faster R-CNN* Untuk menentukan posisi objek kereta, baik lokomotif maupun gerbong, menurut intensitas cahaya dan jarak ke objek tersebut. Sistem ini mencapai tingkat keakuratan dalam mendeteksi objek, dengan rata-rata *accuracy* deteksi untuk lokomotif 86,40%, dan *accuracy* deteksi untuk gerbong mencapai 97,23% (Pardede dan Hardiansah 2022).

Pada penelitian sebelumnya meneliti *accuracy* menggunakan model *Faster R-CNN* yang dilatih untuk mendeteksi senjata genggam kelas pisau. Hasil dari proses pengembangan model *Faster R-CNN* ini adalah model berhasil mendeteksi senjata genggam kelas pisau dengan *accuracy* sebanyak 87%, hasil akurasi diperoleh dari pengujian 475 gambar digital yang dilakukan di *Google Colab* (Pradana dkk. 2022).

Pada penelitian sebelumnya Pembuatan sistem untuk mendeteksi objek ini melibatkan proses pelatihan yang bertujuan untuk mengembangkan model *Faster R-CNN*. Berdasarkan hasil dari penerapan dan pengujian, dapat disimpulkan bahwa sistem ini mampu mengidentifikasi jenis hewan pemakan daging dan pemakan tumbuhan, dengan total model *loss* selama pelatihan mencapai 0.06 dan tingkat akurasi sebesar 89%. Hasil analisis *Recall* dan *Precision* diketahui bahwa kinerja dari sistem klasifikasi yang dibuat sangat memuaskan, dengan *Recall* 100% diperoleh pada kelompok gambar hewan seperti *Cheetah*, *Eagle*, *Komodo*, *Shark*, *Tiger*, *Bull*, *Guineapig*, dan *Zebra* (Putri dan Fahrurozi 2022).

Pada penelitian sebelumnya Sistem ini memakai permodelan *Faster Regional Convolutional Neural Network* yang dibuat pada *Tensorflow* dengan membuatnya di mini komputer atau Raspberry Pi 4B. Hasil akurasi pada sistem ini sebesar 0.9025 atau sebesar 90.25% dengan waktu

komputasi didalam *Raspberry Pi 4B* sebesar 7.638 detik per gambar (Audina dkk. 2021).

Pada penelitian sebelumnya *Faster R-CNN* merupakan pengembangan Jaringan Syaraf Konvolusional (*CNN*). Metode ini masih dalam proses pengembangan dan digunakan untuk mengenali objek atau pengenalan objek. Penelitian ini memanfaatkan kumpulan data burung *lovebird* dengan 808 gambar dan 8 jenis yaitu *Agapornis Personata*, *Agapornis Fischeri*, *Agapornis Lilianae*, *Agapornis Nigrigenis*, *Agapornis Roseicollis*, *Agapornis Taranta*, *Agapornis Pullarius*, *Agapornis Canus*. Hasil proses pembelajaran, diperoleh akurasi sebesar 96% dan kehilangan sebesar 0,2% (Charli dkk. 2020).

Pada penelitian sebelumnya Dalam studi ini, kami menerapkan metode *Single Shot Multibox Detector* yang terkenal untuk mengidentifikasi objek yang bergerak di dalam Aplikasi *Jupyter Notebook*. Data yang digunakan dalam penelitian ini bersifat publik dan diperoleh dari sumber dataset di situs *web Kaggle*, yang memiliki beragam jumlah gambar. Ada tujuh kategori merek mobil yang diteliti, yaitu Volkswagen, Hyundai, Lexus, Mercedes, Peugeot, Renault, dan Tesla. Untuk pengujian dalam penelitian ini, tersedia 1.240 gambar pada kategori data latih dan 270 gambar pada kategori data uji, di mana pengujian telah dilakukan dan menghasilkan angka evaluasi dengan tingkat akurasi terbaik sebesar 98.89% dan nilai loss sebesar 0.025%. (Zikri dkk. 2022).

Pada penelitian sebelumnya Sistem ini menerapkan metode *Single Shot Multibox Detector (SSD)*, Metode ini digunakan untuk mendeteksi gerakan huruf dalam Sistem Isyarat Bahasa Indonesia (SIBI) secara langsung. Cara kerjanya memanfaatkan kamera di laptop untuk menangkap pergerakan dan menampilkan hasilnya di situs web. Model yang diterapkan berhasil diuji dengan mengubah pengaturan hyperparameter. Hasil tes dengan nilai terbaik diperoleh dari dataset dengan rasio 90%:10%, *lr* 0,04, epoch 300, ukuran batch 4, dan 40000 step, dengan tingkat akurasi

mencapai 100%, hasil $mAP@.50IoU$ sebesar 100%, dan hasil $AR@100$ mencapai 91,79% (Apendi dkk. 2023).

Pada penelitian sebelumnya sebuah sistem pengenalan sapi menggunakan gambar dan video dari *UAV* dengan metode *SSD (Single Shot Detector)* bertujuan untuk mendukung peternakan dalam memantau hewan ternak, terutama sapi. Hasil yang diperoleh menunjukkan bahwa sistem ini mampu mengenali objek dengan menandai menggunakan bounding box dan memberikan label sapi. Selain itu, tingkat akurasi tertinggi dalam deteksi objek tercatat saat drone terbang pada ketinggian 5 meter, yaitu 94.45% untuk keadaan diam dan 100% saat bergerak (Fikri dan Setianingsih 2023).

Pada penelitian sebelumnya Proses ekstraksi fitur dilakukan dengan *deep learning* menggunakan *library tensorflow* dan pengolahan citra dengan metode *Single Shot MultiBox Detector (SSD)*. Pengujian dilakukan terhadap objek manusia, hewan (kucing), dan benda (kursi dan meja) untuk kondisi dalam dan luar ruangan dan didapatkan nilai akurasi sebesar 92% , sensitifitas 83%, dan spesifisitas 100% (Anggraeni dkk. 2020).

Pada penelitian sebelumnya Penelitian ini menunjukkan bahwa model *SSD MobileNet* dapat digunakan untuk mengenali berbagai jenis makanan, khususnya untuk kategori yang memiliki *mAP* tinggi. Model ini secara keseluruhan memiliki *mAP* sebesar 77.20%, yang menunjukkan kinerja yang baik dalam mengidentifikasi makanan. Beberapa kategori seperti kentang goreng, nasi goreng, dan tumis kangkung berhasil mencapai 100% *mAP*, yang berarti tingkat akurasi sangat tinggi. Namun, ada beberapa kategori seperti kerupuk putih (30%), tempe goreng (33%), dan terong balado (20%) yang menunjukkan hasil yang kurang memuaskan, menunjukkan kesulitan model dalam mengenali objek-objek tersebut (Puspitasari dan Pangaribuan 2025).

2.2 Dasar Teori

2.2.1 Rokok

Rokok merupakan salah satu produk dari tembakau yang dimaksudkan untuk dibakar, dihisap dan/atau dihirup termasuk didalamnya adalah rokok kretek, rokok putih, cerutu atau pun bentuk lainnya yang dihasilkan dari tanaman *nicotiana tabacum*, *nicotiana rustica* dan spesies lainnya atau sintesis yang asapnya mengandung nikotin dan tar, dengan atau tanpa bahan tambahan (Rosmiati dkk. 2022).



Gambar 2. 1 Rokok

Kawasan-kawasan bebas rokok meliputi tempat untuk belajar dan mengajar yang dikenal sebagai kampus. Kampus adalah lokasi di mana sebuah perguruan tinggi atau Universitas. Umumnya, kampus mencakup perpustakaan, ruang kuliah, asrama, dan taman sebagai bagian dari tata letaknya. Penjelasan ini mencerminkan kumpulan bangunan yang dimiliki oleh lembaga tersebut, baik yang bersifat akademik maupun non-akademik. Kampus adalah area yang dilarang merokok karena khawatir aktivitas merokok dapat mengganggu proses belajar mengajar. Oleh karena itu, dalam upaya penertiban, pemerintah mengeluarkan peraturan mengenai kawasan tanpa rokok (Juweha 2019).

2.2.2 Computer Vision

Computer Vision merupakan bidang ilmu komputer yang memberi mesin kemampuan untuk melihat dan memahami gambar serta video. Dengan menggunakan algoritma canggih, komputer dapat mengidentifikasi wajah, objek, dan bahkan memahami adegan rumit layaknya manusia. Penggunaan

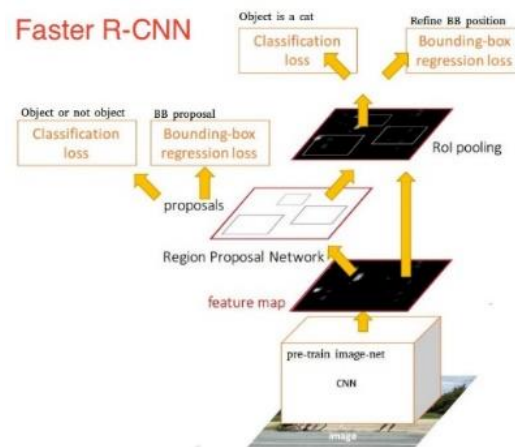
computer vision sangat beragam, mencakup keamanan, otomotif, medis, dan lingkungan. Secara ringkas, *computer vision* memungkinkan mesin untuk mendapatkan informasi penting dari dunia visual dan bertindak berdasarkan pemahaman tersebut (Sutisna dkk. 2024).

2.2.3 *Object Detection*

Deteksi objek atau Object Detection adalah aspek dari Penglihatan Komputer. Deteksi objek merujuk pada kemampuan perangkat komputer untuk mengenali berbagai objek dalam sebuah gambar. Proses ini melibatkan pengambilan fitur gambar seperti garis, sudut, kontur, dan warna dari suatu gambar. Deteksi objek juga merupakan bagian dari Pengenalan Objek atau Object Recognition. Jadi, bisa disimpulkan bahwa untuk melakukan deteksi objek, objek tersebut harus terlebih dahulu diidentifikasi. Dalam penelitian ini, akan dilakukan deteksi objek dan selanjutnya diterapkan untuk mengukur jarak serta menghitung jumlah objek (Indaryanto dkk. 2021).

2.2.4 *Faster regional convolutional neural network*

Faster regional convolutional neural network atau *Faster R-CNN* adalah sebuah metode yang memanfaatkan *convolutional neural network*. *Faster R-CNN* berfungsi sebagai sebuah teknik untuk mendeteksi objek yang terpantau oleh kamera, yang kemudian akan disimpan dalam *database* atau tidak, sehingga komputer dapat membuat kesimpulan tentang objek yang telah kita lacak. Terdiri atas lima komponen utama yang meliputi jaringan konvolusional, *RPN*, Lapisan ROI Pooling, serta Lapisan *full coneected*, box bounding, dan pengklasifikasi. Lewat jaringan usulan wilayah dan jaringan konvolusional dalam yang sepenuhnya, banyak objek diusulkan untuk menjadi proposisi dan kemudian dinormalisasi melalui Lapisan *ROI Pooling*. Selanjutnya, Lapisan Terkoneksi Penuh mengambil fitur-fitur relevan guna melakukan proses *clasification* dan *regretion* (Pardede dan Hardiansah 2022).



Gambar 2. 2 Arsitektur *faster r-cnn* (Charli dkk. 2020)

Faster-RCNN merupakan suatu sistem yang digunakan untuk identifikasi objek yang terdiri dari dua elemen utama. Elemen pertama adalah Jaringan Saraf Convolutional Regional, sementara elemen kedua adalah detektor dari Jaringan Saraf Konvolusional Regional Fast yang memanfaatkan wilayah yang diajukan. *Faster-RCNN* terbentuk dari lima komponen inti, jaringan konvolusional, *RPN*, Lapisan ROI Pooling, serta Lapisan *full connected*, box bounding, dan pengklasifikasi. Melalui lapisan konvolusi dan lapisan terhubung penuh, berbagai objek diusulkan sebagai saran, kemudian dinormalisasi dengan lapisan ROI Pooling. Selanjutnya, lapisan terhubung penuh mengambil fitur-fitur yang relevan untuk klasifikasi. Kebanyakan usulan tidak selalu sesuai dengan objek yang dituju, dan sistem berupaya mencari yang paling tepat. Akan tetapi, klasifikasi dilakukan berdasarkan saran itu sendiri, bukannya hanya berdasarkan yang paling tepat. Dalam aplikasinya di dunia nyata, ukuran objek bervariasi: seperti motor, hewan, dan plat. Ketika objek yang diincar berbeda, saran yang untuk ukuran besar dapat mencakup objek baik yang kecil maupun yang besar. Dalam keadaan ini, klasifikasi harus menghadapi tidak hanya variasi di dalam area objek, tetapi juga variasi di sekitar objek. Ini menjadi tidak efisien untuk klasifikasi dan kotak terbatas, serta dapat menghasilkan banyak hasil positif palsu dan negatif palsu saat menguji model (Pardede dan Hardiansah 2022).

2.2.5 Regional Proposal Network

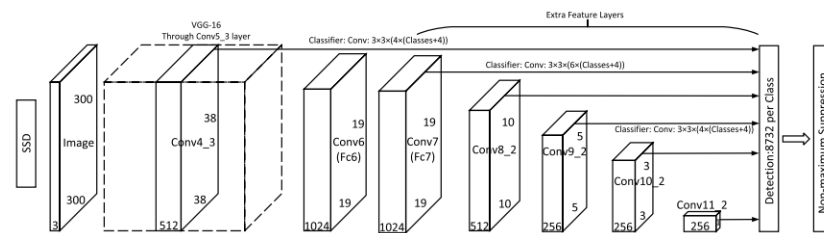
RPN digunakan untuk menghasilkan saran berdasarkan fitur yang ada. Peta fitur berukuran 512 diambil jaringan *convolutional* dasar menerima semua gambar menjadi input. Peta fitur yang didapat akan digunakan sebagai jendela geser berukuran 3x3, dengan setiap jendela yang digeser dipetakan menjadi vektor fitur berukuran 512. Vektor fitur itu kemudian di *input*-kan ke dua lapisan terhubung yaitu *full layer*, diikuti oleh lapisan regresi kotak dan deteksi kotak. pada konsep ini, prinsip *Anchor* diperkenalkan. *anchor* fokus ke jendela geser 3x3, di mana setiap jendela yang digeser mempunyai sembilan *anchor* dengan campuran tiga skala [128 2, 256 2, 512 2] dan tiga rasio aspek [1:1, 1:2, 2:1]. Lapisan klasifikasi kotak menentukan apakah *anchor* positif atau negatif, sedangkan lapisan regresi menghasilkan koordinat untuk pembatas. Untuk *training RPN*, fungsi kerugian dijabarkan seperti berikut.:

$$L = \frac{l}{N_{ds}} \sum L_{ds}(p, p^* + \lambda \frac{l}{N_{reg}}) \sum p^* L_{reg}(t, t^*) \quad (1)$$

Dimana indeks dari *Anchor* dalam *mini batch*,

- p adalah proposal objek,
- p^* adalah label dari proposal (ketika proposal adalah objek, $p^* = 1$, jika tidak $p^* = 0$),
- t dan t^* adalah koordinat dari *bounding box regression* dan *ground-truth* masing-masing,
- N_{cls} dan N_{reg} adalah parameter normalisasi,
- L_{cls} adalah kerugian deteksi, itu adalah kerugian log selama dua kelas,
- L_{reg} adalah defisit regresi, istilah $p^* L_{reg}$ berarti regresi dihidupkan guna untuk *anchor* positif (Pardede dan Hardiansah 2022).

2.2.6 Single Shot Detector



Gambar 2. 3 Arsitektur SSD (Cheng dkk. 2022)

Single Shot Detector (SSD) adalah metode deteksi objek yang menggunakan pendekatan *single deep neural network* untuk mengenali objek dalam sebuah gambar secara *real-time*. SSD bekerja dengan memanfaatkan kotak pembatas (*bounding box*), fitur multi-skala, dan kotak *default* untuk meningkatkan akurasi dan kecepatan deteksi, bahkan dengan resolusi gambar rendah. Algoritma ini dikembangkan untuk menjadi alternatif yang lebih cepat dibanding *Faster R-CNN* dengan akurasi yang sebanding. Arsitektur SSD didasarkan pada *Convolutional Neural Network (CNN)*, khususnya *VGG16* sebagai *backbone*. *VGG16* dipilih karena performanya yang baik dalam klasifikasi gambar serta mendukung transfer learning. Dalam SSD, *VGG16* dihubungkan dengan beberapa lapisan konvolusi tambahan yang bertugas mengekstrak fitur dari berbagai skala secara bertahap, yang memungkinkan deteksi objek pada ukuran yang berbeda-beda. Dalam penelitian ini digunakan varian *SSD MobileNetV2 FPNLite*, di mana backbone *VGG16* digantikan oleh *MobileNetV2*, yang lebih ringan dan hemat daya, cocok untuk perangkat dengan sumber daya terbatas. *MobileNetV2* memanfaatkan *depthwise* dan *pointwise convolution* serta fitur linear *bottleneck* dan *shortcut connection* untuk mempercepat pelatihan dan meningkatkan akurasi. Selain itu, ditambahkan komponen *Feature Pyramid Network (FPN)* untuk mendukung deteksi objek pada berbagai skala. *FPN* menggabungkan fitur dari berbagai tingkat lapisan konvolusi, membentuk struktur piramida yang menghasilkan peta fitur multi-skala berkualitas tinggi. Gabungan *MobileNetV2* dan *FPN* menjadikan *SSD MobileNetV2 FPNLite* sebagai arsitektur yang efisien namun tetap akurat dalam mendeteksi objek (Fikri dan Setianingsih 2023).

Berikut rumus *loss function* pada *ssd* :

$$L = (x, c, l, g) = \frac{1}{N} (Lconf(x, c) + \alpha Lloc(x, l, g)) \quad (2)$$

Keterangan :

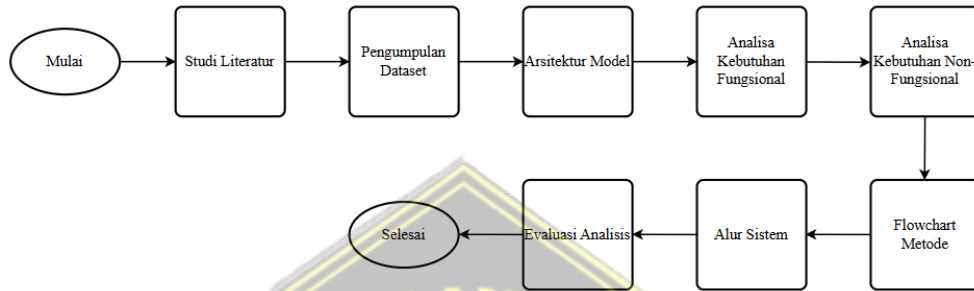
- L = Total *loss*
- $Lconf$ = *confidence loss* (biasanya *softmax loss*)
- $Lloc$ = *localization loss* (biasanya *smooth L1 loss*)
- x = indikator matching antara *default box* dan *ground truth box*
- c = kelas prediksi
- l = Prediksi lokasi
- g = *ground truth bounding box*
- N = jumlah *positive matches* (jumlah *default box* yang cocok dengan *ground truth*)
- α = bobot pengali untuk *balance* antara *confidence loss* dan *localization loss* (biasanya $\alpha = 1$)



BAB III

METODE PENELITIAN

Untuk metode penelitian yang akan dipakai dalam penelitian, adalah sebagai berikut :



Gambar 3. 1 Alur penelitian

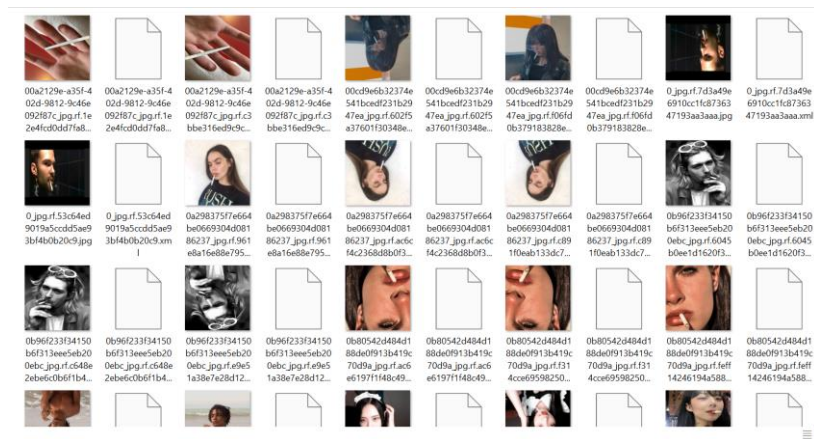
Pada gambar 3.1 memperlihatkan alur dari penelitian yang dilakukan pada saat penelitian, mulai dari studi literatur, pengumpulan *dataset*, arsitektur model, analisa kebutuhan fungsional, analisa kebutuhan non-fungsional, *flowchart* metode, alur sistem, dan evaluasi analisis.

3.1 Studi Literatur

Peneliti mengumpulkan beberapa *e-book*, artikel, jurnal, skripsi dan tesis hasil penelitian yang dilakukan oleh peneliti sebelumnya, dan beberapa laman di internet sebagai media untuk peneliti mempelajari terkait *Machine Learning*, *Deep Learning*, *Computer vision*, *Object detection*, *Faster R-CNN*, dan *Single Shot Detector* untuk membantu peneliti menyelesaikan penelitian.

3.2 Pengumpulan *Dataset*

Dataset yang digunakan dalam penelitian ini didapat dari sumber *Roboflow* dengan jenis data *Pascal VOC (Visual Object Classes)* sejumlah 1935 data. Informasi yang terdapat pada gambar berupa anotasi, label, dan ukuran, dalam *dataset* ini disimpan dalam *file xml*. *Dataset* akan diproses untuk digunakan dalam tahapan *training* dan *validation* dengan perbandingan jumlah gambar *training* 85:15 *validation*.

Gambar 3.2 *Dataset*

Pada gambar 3.2 merupakan contoh dari gambar yang ada di dalam *dataset* yang akan digunakan. Setelah diperoleh, *dataset* akan dilakukan *preprocessing* data untuk menyesuaikan gambar yang dibutuhkan dalam pembuatan model deteksi ini.

3.3 Arsitektur Model

3.3.1 *Faster R-CNN*

Pada tahapan model *Faster R-CNN*, gambar pertama-tama akan diproses oleh *convolutional neural network (CNN)*. *CNN* akan mencari pola-pola atau fitur seperti tepi, sudut, tekstur, atau bentuk di dalam *dataset*. *CNN* akan menghasilkan sebuah "peta fitur" (*feature map*), yang bisa dianggap sebagai gambaran kasar dari gambar, namun dengan informasi yang lebih fokus pada objek yang ada di dalamnya. Setelah gambar diubah menjadi peta fitur, ada bagian penting lain yang disebut *Region Proposal Network (RPN)*. Fungsi *RPN* adalah untuk mencari kemungkinan lokasi objek dalam gambar. Artinya, *RPN* akan mencoba untuk menemukan area-area di dalam gambar yang berpotensi berisi objek rokok. *RPN* bekerja dengan cara membuat banyak *bounding box* (kotak pembatas) di seluruh gambar dan menilai mana yang paling mungkin berisi objek yang ingin dideteksi. Setelah *RPN* menentukan beberapa area yang mungkin berisi objek, langkah berikutnya adalah *RoI Pooling*. Di bagian ini, gambar yang telah diproses oleh *RPN* akan dipotong-potong sesuai dengan area yang dianggap penting. *RoI Pooling* mengambil potongan-potongan gambar ini dan mengubahnya menjadi ukuran

yang sama (misalnya 7x7 piksel). Dengan cara ini, model bisa memproses setiap area (potongan gambar) yang berisi objek dengan cara yang sama, sehingga model bisa lebih efisien dalam membuat prediksi. Pada tahap terakhir, *Faster R-CNN* menggunakan dua Langkah yaitu *Klasifikasi* dan *Regresi*. *Klasifikasi*, model akan menentukan apakah setiap area (potongan gambar) yang diambil oleh *RPN* berisi objek rokok atau tidak. Ini adalah bagian dari proses pengenalan objek. *Regresi*, selain mendeteksi objek, model juga akan memperbaiki lokasi objek dalam gambar. Ini dilakukan dengan menghitung ulang posisi dan ukuran *bounding box* untuk mendapatkan prediksi lokasi objek yang lebih akurat.

3.3.2 *Single Shot Detector*

SSD (Single Shot MultiBox Detector) adalah algoritma deteksi objek yang dirancang untuk mendeteksi berbagai objek dalam gambar secara cepat dan efisien dalam satu proses langsung, tanpa memerlukan tahap *proposal region* seperti pada *R-CNN*. Proses deteksi dimulai dengan *backbone* berbasis *convolutional neural network (CNN)*, seperti *VGG-16* atau *ResNet*, yang bertugas mengekstraksi fitur-fitur penting dari gambar, seperti tepi, pola, tekstur, dan bentuk objek. Hasil ekstraksi ini berupa *feature map* yang merepresentasikan informasi visual dari berbagai tingkatan. *SSD* kemudian memanfaatkan beberapa *layer* konvolusional tambahan yang secara bertahap mengecilkan resolusi *feature map*, sehingga memungkinkan deteksi objek pada berbagai skala. Hal ini menjadikan *SSD* mampu mendeteksi objek besar di *layer* awal dan objek kecil di *layer* yang lebih dalam. Pada setiap *layer*, *SSD* memprediksi sejumlah *default boxes (anchor box)* dengan berbagai ukuran dan rasio aspek. Setiap kotak default digunakan untuk memprediksi keberadaan objek beserta kelasnya melalui dua tugas utama, yaitu klasifikasi dan regresi. Klasifikasi bertugas menentukan jenis objek seperti rokok atau bukan, sedangkan regresi bertugas memprediksi lokasi dan ukuran *bounding box* yang akurat. *SSD* membagi gambar menjadi *grid* dan melakukan prediksi di setiap *grid* berdasarkan kotak *default* yang sesuai. Keunggulan utama *SSD* adalah efisiensinya dalam melakukan deteksi dalam satu tahap, dengan tetap

mempertahankan akurasi yang kompetitif. *Output* akhir dari *SSD* mencakup koordinat bounding box, label kelas objek, dan *confidence score* sebagai indikator keyakinan model terhadap prediksi.

3.4 Analisa Kebutuhan Fungsional

Dalam penelitian ini, kebutuhan fungsional sistem disusun berdasarkan tahapan metode *CRISP-DM*, yang merupakan kerangka kerja standar untuk pengembangan sistem berbasis data mining dan pembelajaran mesin. Sistem yang dibangun ditujukan untuk mendeteksi aktivitas merokok secara otomatis dengan akurasi tinggi, sebagai bagian dari dukungan terhadap penerapan kebijakan kawasan tanpa rokok di lingkungan kampus. Pada tahap *business understanding*, sistem dirancang agar mampu menjawab kebutuhan institusi kampus dalam mendeteksi pelanggaran terhadap aturan larangan merokok. Sistem ini harus mampu menghasilkan keluaran yang dapat digunakan sebagai alat bantu pengawasan dan evaluasi kebijakan. Tahap *data understanding* menuntut sistem untuk dapat mengimpor dan membaca dataset dari sumber eksternal, dalam hal ini *Roboflow*, serta memahami struktur data seperti jumlah gambar, format anotasi, dan label yang digunakan. Pemahaman data ini penting agar proses pelatihan model dapat berjalan secara optimal. Selanjutnya, dalam tahap *data preparation*, sistem perlu mampu melakukan berbagai proses pra-pemrosesan data. Hal ini mencakup augmentasi gambar untuk memperkaya variasi data, pemetaan ulang label untuk menyederhanakan klasifikasi menjadi dua kelas (merokok dan tidak merokok), serta pembagian dataset ke dalam data latih dan validasi dengan proporsi yang tepat. Selain itu, sistem harus mampu memastikan bahwa seluruh dataset memiliki struktur folder dan format file yang sesuai dengan kebutuhan model, seperti format *Pascal VOC* (.xml). Pada tahap *modeling*, sistem harus memiliki kemampuan untuk membangun dan melatih dua model deteksi objek berbasis *deep learning*, yaitu *Faster R-CNN* dan *Single Shot Detector (SSD)*. Sistem ini juga harus menyediakan pengaturan parameter pelatihan seperti nilai *batch size*, *learning rate*, jumlah *epoch*, serta jenis *optimizer* yang digunakan. Selama proses pelatihan, sistem perlu

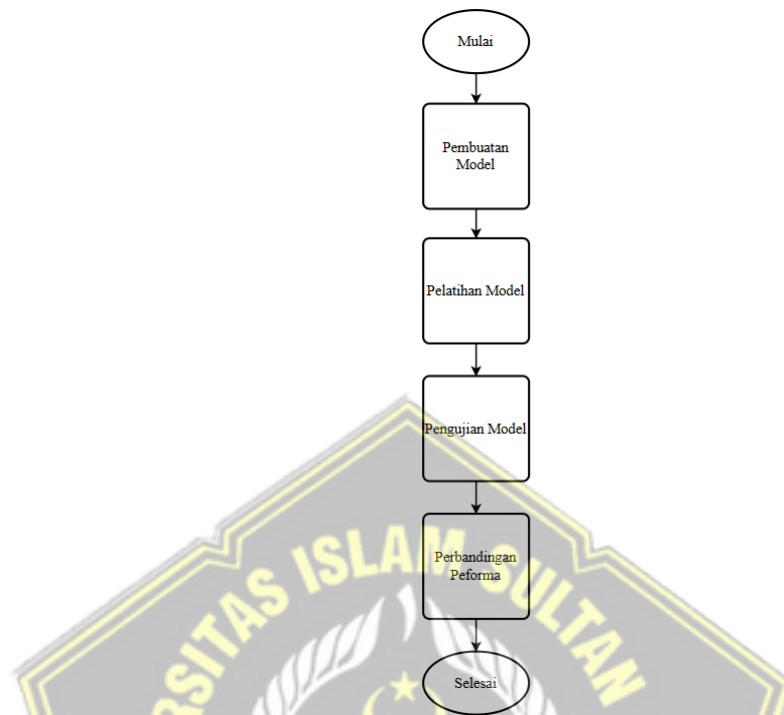
menampilkan grafik *loss* untuk menunjukkan proses belajar model dari waktu ke waktu. Tahap *evaluation* menuntut agar sistem dapat menghitung metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score* secara otomatis setelah model diuji. Selain itu, sistem harus mampu menampilkan hasil evaluasi tersebut dalam bentuk tabel serta grafik perbandingan performa kedua model. Output visual berupa gambar deteksi yang dilengkapi bounding box, label klasifikasi, dan confidence score juga menjadi bagian dari kebutuhan fungsional untuk memvalidasi keberhasilan model secara visual. Meskipun tahap *deployment* belum menjadi fokus utama dalam penelitian ini, sistem telah dirancang agar hasil model dapat disimpan dalam format yang mendukung penerapan lebih lanjut, seperti file model berformat *.pth*. Hal ini memungkinkan sistem untuk diintegrasikan ke dalam sistem pemantauan video real-time di masa mendatang sebagai pengembangan lanjutan. Dengan demikian, seluruh kebutuhan fungsional dalam penelitian ini disesuaikan untuk mendukung proses pengolahan data, pelatihan model, evaluasi performa, serta potensi implementasi sistem deteksi aktivitas merokok berbasis deep learning secara komprehensif dan terstruktur.

3.5 Analisa Kebutuhan Non-Fungsional

Pada tahap ini kebutuhan non fungsional adalah kebutuhan yang dimiliki setiap sistem yang membatasi pengembangan ataupun standarisasi sistem yang akan dibuat. Kebutuhan tersebut terdiri dari :

1. Perangkat Keras (*Hardware*) yang digunakan dalam pembuatan aplikasi sebagai berikut :
 - a. Laptop Toshiba C655, Prosesor : Intel(R) Core (TM) i3, RAM : 4,00 GB, dan SSD : 256 GB
2. Perangkat Lunak (*Software*) Software yang digunakan dalam perancangan dan pembuatan aplikasi ini sebagai berikut:
 - a. OS : Windows 10
 - b. Bahasa program : Python
 - c. Compile : Google Colab
 - d. Perencanaan : Visual Paradigm.

3.6 Flowchart Metode



Gambar 3. 3 Alur *flowchart* model

Pada gambar 3.3 menunjukkan *flowchart* model dari penelitian yang akan dijalankan oleh peneliti, dimulai dengan pembuatan model, pelatihan model, pengujian model, dan perbandingan peforma.

1. Pembuatan model

Pada tahap pembuatan model, model deteksi berbasis *Faster R-CNN* dan *Single Shot Detector* dirancang dan dikembangkan. Proses ini melibatkan pengumpulan *dataset* berupa gambar atau video yang relevan dengan aktivitas merokok, dilanjutkan dengan anotasi dataset untuk menandai area-area yang mengindikasikan aktivitas merokok. Pembuatan model ini bertujuan agar algoritma dapat mengenali objek rokok, asap, dan aktivitas merokok dengan akurasi tinggi.

2. Pelatihan model

Setelah model selesai dirancang, tahap berikutnya adalah melatih algoritma menggunakan *dataset* yang telah disiapkan. Proses pelatihan melibatkan pemberian data *input* kepada algoritma untuk mengajarkan bagaimana mendeteksi pola-pola tertentu yang terkait dengan aktivitas

merokok. Pada tahap ini, berbagai parameter seperti *learning rate*, jumlah *epoch*, dan ukuran *batch* dioptimalkan untuk memastikan algoritma dapat belajar dengan baik. *Faster R-CNN* dan *Single Shot Detector* dilatih secara terpisah untuk membandingkan performanya.

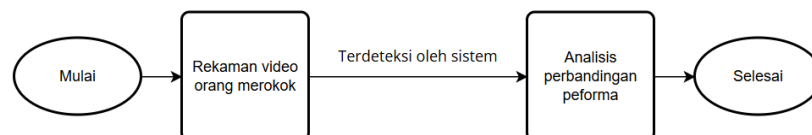
3. Pengujian model

Tahap pengujian dilakukan setelah proses pelatihan selesai. Tahap ini, model yang dilatih diuji dengan data uji yang belum dilihat sebelumnya oleh algoritma. Tujuannya adalah untuk mengevaluasi kemampuan model dalam mendeteksi aktivitas merokok di berbagai kondisi, seperti pencahayaan, sudut kamera, dan tingkat keramaian. Hasil pengujian mencakup metrik seperti *accuracy*, *precision*, *recall*, dan *f1-score*.

4. Perbandingan peforma

Hasil dari pengujian kedua algoritma dibandingkan untuk menganalisis kelebihan dan kekurangannya masing-masing. *Faster R-CNN* biasanya lebih unggul dalam hal akurasi deteksi, tetapi memiliki waktu *inferensi* yang lebih lama karena arsitekturnya yang kompleks. Sebaliknya, *SSD (Single Shot MultiBox Detector)* dikenal dengan kecepatan deteksinya yang tinggi, meskipun akurasinya mungkin sedikit lebih rendah dibandingkan *Faster R-CNN*. Analisis ini bertujuan untuk menentukan algoritma yang paling sesuai untuk diterapkan di lingkungan Unissula, terutama untuk mendukung upaya penegakan kebijakan zona larangan merokok.

3.7 Alur Sistem



Gambar 3. 4 Alur sistem

Gambar 3.4 menjelaskan alur kerja sistem yang dimulai dengan tahap inisialisasi proses. Data utama berupa rekaman video aktivitas merokok di lingkungan Unissula digunakan sebagai input sistem. Video tersebut

diproses oleh dua algoritma, yaitu *Faster R-CNN* dan *Single Shot Detection*, untuk mendeteksi aktivitas merokok. Hasil deteksi dari kedua algoritma ini kemudian dibandingkan melalui analisis performa yang mencakup *accuracy*, *precision*, *recall*, dan *f1-score*. Setelah proses analisis selesai, hasil penelitian dirangkum untuk menentukan algoritma yang lebih unggul, dan penelitian pun diakhiri.

3.8 Evaluasi dan analisis

Pada tahapan evaluasi dan analisis, peneliti akan menggunakan beberapa matrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *f1-score*. Untuk rumus dari setiap metrik adalah berikut:

1. *Accuracy* merupakan proses untuk mengukur seringnya model membuat prediksi benar dibandingkan oleh semua prediksi yang dikerjakan.

$$\frac{TP+TN}{TP+FP+TN+FN} \quad (3)$$

2. *Precision* adalah persentase deteksi positif yang benar (benar terdeteksi sebagai aktivitas merokok) dibandingkan dengan semua deteksi positif.

$$\frac{TP}{TP+FP} \quad (4)$$

3. *Recall* merupakan persentase aktivitas merokok yang berhasil terdeteksi dibandingkan dengan semua aktivitas merokok yang ada dalam *dataset*.

$$\frac{TP}{TP+FN} \quad (5)$$

4. *F1-Score* adalah metrik yang digunakan untuk mengukur keseimbangan dengan *precision* dan *recall* dalam satu nilai. Semakin tinggi nilainya, maka semakin baik performa model secara keseluruhan dalam mendeteksi dan tidak salah deteksi.

$$2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1 Hasil

Data yang dikumpulkan dalam studi ini terdiri dari 1935 gambar yang diambil dari situs web *Roboflow*, dimana data tersebut dibagi untuk tahap *training*, *validation*, dan *testing*. Data yang didapatkan sudah dilengkapi dengan *bounding box* yang tersimpan dalam *file* berformat *.xml* dengan kategori “*cigarette*”. Gambar 4.1 dan 4.2 menggambarkan kondisi awal dari dataset yang diambil dari Roboflow.



Gambar 4. 1 Folder awal



Gambar 4. 2 Gambar dan anotasi dalam satu folder

Dalam proses merancang model, kita akan memanfaatkan dataset dengan pembagian 85:15 untuk *training* dan *validation*, yang berarti semua data yang ada akan digabungkan. Data tersebut akan dihimpun dalam satu *folder*, lengkap dengan *subfolder* anotasi yang menyimpan file anotasi serta gambar yang menyimpan seluruh gambar sesuai dengan gambar 4.3 untuk digunakan dalam tahap pengembangan model. Tata letak *folder* ini memiliki peranan yang signifikan dalam menjamin bahwa alur pemrosesan data, proses pelatihan, serta evaluasi model dapat dilakukan dengan lancar tanpa ada kesalahan. Dengan mengorganisir data ke dalam format *folder* yang tertata,

kegiatan *augmentasi*, pembacaan *file*, serta pencocokan gambar dengan anotasi akan berlangsung lebih efisien dan mengurangi kemungkinan kesalahan saat dijalankan dalam kode dan diupload di *Google drive*.

 anotasi	18/05/2025 13:18	File folder
 gambar	18/05/2025 13:18	File folder

Gambar 4. 3 Folder baru

Pada gambar 4.4 adalah proses pemanggilan dataset yang di-*upload* dari *Google drive* dan proses *unzip* dari dataset yang akan kita gunakan.

```

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] !unzip '/content/drive/MyDrive/dataset2.zip'

Archive: /content/drive/MyDrive/dataset2.zip
creating: dataset2/
creating: dataset2/anotasi/
inflating: dataset2/anotasi/000509.jpg.rf.54f9ca7d2e529acace7d4b7707913caf.xml
inflating: dataset2/anotasi/000509.jpg.rf.a2d8b46a314a45be3540502ec5622ca4.xml
inflating: dataset2/anotasi/00a2129e-a35f-402d-9812-9c46e092f87c.jpg.rf.1e2e4fcd0dd7fa866b94e38462fa3c71.xml
inflating: dataset2/anotasi/00a2129e-a35f-402d-9812-9c46e092f87c.jpg.rf.c3bbe316ed9c9c3892945fa473d89fea.xml
inflating: dataset2/anotasi/00d9e6b32374e541bcd7231b2947ea.jpg.rf.602f5a37601f30348e7bd618ce83c451.xml

```

Gambar 4. 4 Pemanggilan dataset dan *unzip*

Sebelum data dibagi, akan dilakukan pemetaan label yang bertujuan untuk menyederhanakan dan menyamakan berbagai jenis label yang ada di dataset agar lebih konsisten dan mudah diproses oleh model. Dalam proses pembuatan model untuk mendeteksi aktivitas merokok, terdapat label “*cigarette*” yang mana mewakili satu kategori, yaitu “Merokok” seperti pada gambar 4.5 Label diwakili dengan angka satu karena disini peneliti hanya menggunakan dua *class* yaitu “Merokok” dan “Tidak merokok” seperti pada gambar 4.6 sehingga selain angka satu di ibaratkan nol alias tidak merokok.

```

# mapping label
label_mapping = {
    "cigarette": 1
}

```

Gambar 4. 5 Label mapping

```
#label 0 is fixed for background
classes=["tidak merokok","merokok"]

num_classes=2
```

Gambar 4. 6 jumlah class pada model

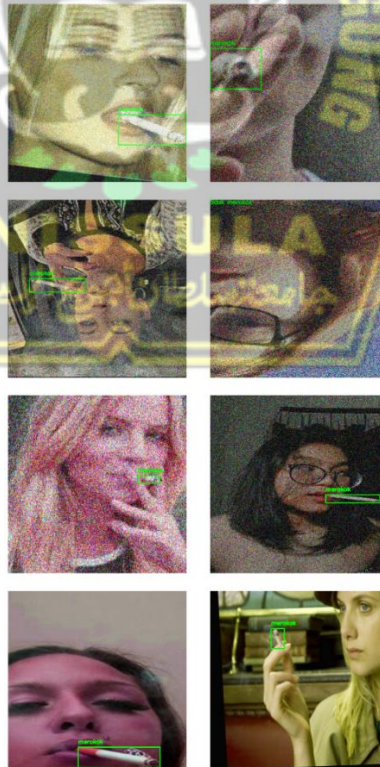
Data tersebut kemudian dilakukan *augmentasi* untuk memperkaya keragaman data menggunakan berbagai transformasi dengan hasil seperti gambar 4.7 dan hasil dari augmentasi tersebut ditunjukkan oleh gambar 4.8.

```
[ ] train_transform=A.Compose([A.HorizontalFlip(),
    A.ShiftScaleRotate(rotate_limit=15,value=0,
        border_mode=cv2.BORDER_CONSTANT),

    A.OneOf([
        A.CLAHE(),
        A.RandomBrightnessContrast(),
        A.HueSaturationValue(p=1),
        A.GaussNoise(),
        A.RandomResizedCrop(height=480, width=480, size=(480,480))], # Added size parameter as a tuple
        bbox_params=A.BboxParams(format='pascal_voc',min_visibility=0.15,
            label_fields=['labels']))

val_transform=A.Compose([A.Resize(height=480,width=480)],
    bbox_params=A.BboxParams(format='pascal_voc',min_visibility=0.15,
        label_fields=['labels']))

/usr/local/lib/python3.11/dist-packages/segmentation/core/validation.py:111: UserWarning: ShiftScaleRotate is a special case of Affine transform. Pl
original_init(self, **validated_kwargs)
<ipython-input-14-49baf828c36e>:2: UserWarning: Argument(s) 'value' are not valid for transform ShiftScaleRotate
A.ShiftScaleRotate(rotate_limit=15,value=0,
<ipython-input-14-49baf828c36e>:10: UserWarning: Argument(s) 'height, width' are not valid for transform RandomResizedCrop
A.RandomResizedCrop(height=480, width=480, size=(480,480))], # Added size parameter as a tuple
```

Gambar 4. 7 source code *augmentasi*Gambar 4. 8 Hasil *augmentasi*

Pada gambar 4.9 dan gambar 4.10 merupakan pemanggilan model *pretained* yang akan digunakan dalam training. Hasil pembuatan model yang dilakukan dengan tahapan *training* dan *validation* menggunakan dataset yang menghasilkan grafik *loss* seperti gambar 4.11. Grafik *Loss Progression* dari hasil pelatihan model *Faster R-CNN* menunjukkan penurunan yang konsisten pada *training loss* dan *validation loss* seiring bertambahnya jumlah *epoch*, yang menandakan bahwa model berhasil belajar dan beradaptasi terhadap pola-pola dalam data secara benar. Pada awal pelatihan (*epoch* ke-0), nilai *loss* masih cukup tinggi dengan *training loss* berada di atas 0.10 dan *validation loss* di sekitar 0.075, namun keduanya secara bertahap mengalami penurunan yang stabil hingga mencapai nilai mendekati 0.045 pada akhir pelatihan di *epoch* ke-24. Hal ini menunjukkan bahwa model mengalami perbaikan dalam meminimalkan kesalahan prediksi baik pada data latih maupun data validasi. Selain itu, grafik ini juga memperlihatkan bahwa tidak terdapat gejala *overfitting* yang signifikan, karena nilai *validation loss* tetap berada dalam tren yang sama dengan *training loss* dan tidak menunjukkan lonjakan atau penyimpangan tajam. Dengan demikian, grafik ini memberikan indikasi kuat bahwa model *Faster R-CNN* yang digunakan memiliki performa pelatihan yang baik dan generalisasi yang cukup stabil terhadap data validasi.

Begitu juga dengan gambar 4.12. Berdasarkan grafik *Loss Progression* selama proses pelatihan model *Single Shot Detector (SSD)*, terlihat bahwa nilai *training loss* dan *validation loss* secara konsisten mengalami penurunan seiring bertambahnya jumlah *epoch*, yang menunjukkan bahwa model berhasil belajar dari data pelatihan dan mampu melakukan generalisasi dengan cukup baik terhadap data validasi. Pada awal pelatihan (*epoch* ke-0), *training loss* berada di angka sekitar 1.95 dan *validation loss* di kisaran 1.33, namun keduanya menunjukkan tren penurunan yang stabil hingga mencapai sekitar 0.83 pada akhir pelatihan di *epoch* ke-24. Tidak terdapat indikasi *overfitting* yang signifikan, karena *validation loss* cenderung mengikuti pola penurunan yang serupa dengan *training loss* dan bahkan menunjukkan nilai yang sedikit lebih rendah pada beberapa *epoch*, yang mengindikasikan model

tidak hanya mengetahui data pelatihan, tetapi mampu mengetahui pola pada data yang belum pernah dilihat sebelumnya. Secara keseluruhan, grafik ini menunjukkan proses training berjalan baik.

```
[ ] model = fasterrcnn_resnet50_fpn(pretrained=True)

in_features = model.roi_heads.box_predictor.cls_score.in_features
model.roi_heads.box_predictor = torchvision.models.detection.faster_rcnn.FastRCNNPredictor(in_features, num_classes)
device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
model.to(device)

(bn2): FrozenBatchNorm2d(256, eps=0.0)
(conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
(bn3): FrozenBatchNorm2d(1024, eps=0.0)
(relu): ReLU(inplace=True)
```

Gambar 4. 9 Model *Faster Rcn*

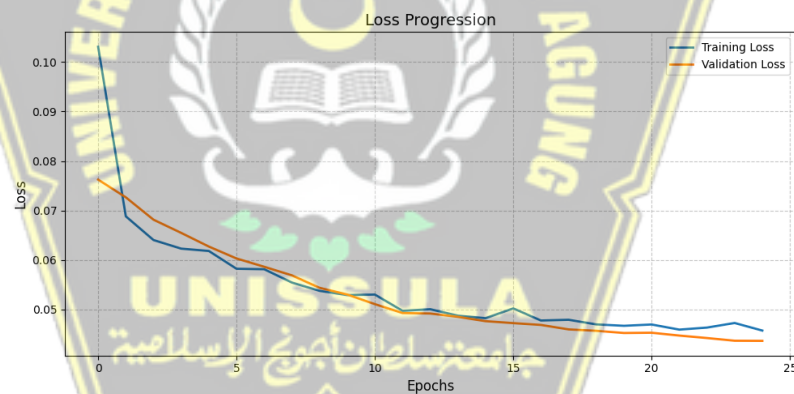
```
[ ] model = ssd300_vgg16(weights=SSD300_VGG16_Weights.DEFAULT)

in_channels = det_utils.retrieve_out_channels(model.backbone, (480, 480))
num_anchors = model.anchor_generator.num_anchors_per_location()
model.head = SSDHead(in_channels=in_channels, num_anchors=num_anchors,
                      num_classes=num_classes)

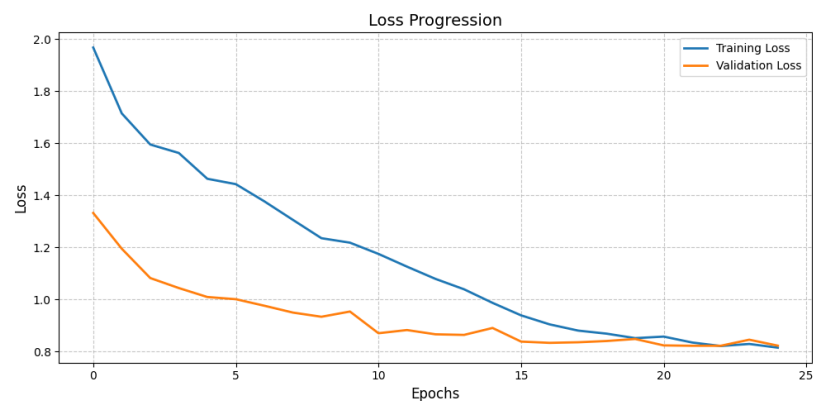
model.to(device)

Downloading: "https://download.pytorch.org/models/ssd300_vgg16_coco-b556d3b4.pth" to /
100% 136M/136M [00:01<00:00, 79.9MB/s]
SSD(
  (backbone): SSDFeatureExtractorVGG(
    (features): Sequential(
      (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
      (1): ReLU(inplace=True)
      (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Gambar 4. 10 Model *SSD*

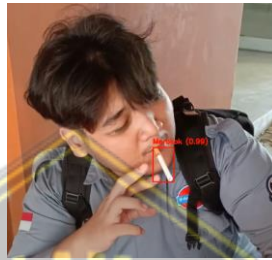


Gambar 4. 11 Grafik *loss training dan validation faster r-cnn*



Gambar 4. 12 Grafik *loss training dan validation single shot detector*

Model kemudian diunduh dalam bentuk *.pth* dan dilakukan pengujian menggunakan *google colab*. Adapun cara menjalankannya, pertama memastikan model yang sudah dilatih sebelumnya melalui tahapan proses *training* dan *validation*, serta memanggil *file* model untuk dijalankan pada kode program sistem deteksi aktivitas merokok. Apabila sudah dipastikan dalam posisi yang sesuai, maka program bisa dijalankan.



Gambar 4. 13 Tampilan mendeteksi rokok dengan model *faster r-cnn*



Gambar 4. 14 Tampilan mendeteksi rokok dengan model *ssd*

Gambar 4.9 dan gambar 4.10 menunjukkan hasil pengujian dari sistem deteksi aktivitas merokok yang telah dikembangkan, di mana sistem mampu mengenali keberadaan seseorang yang sedang merokok dengan cukup akurat. Pada tampilan hasil tersebut, rokok yang sedang dipegang oleh seseorang dalam gambar berhasil terdeteksi dan secara otomatis diberi penanda berupa kotak (*bounding box*) pada area yang mengindikasikan objek rokok. Selain kotak deteksi, *output* sistem juga menampilkan label berupa tulisan "Merokok" yang menunjukkan klasifikasi objek, serta dilengkapi dengan nilai *confidence score* yang merepresentasikan tingkat keyakinan model terhadap deteksi tersebut. Nilai *confidence score* ini berfungsi sebagai indikator sejauh mana model yakin bahwa objek yang terdeteksi memang merupakan rokok, sehingga mendukung validasi hasil deteksi yang ditampilkan pada layar.

4.2 Analisa

Pada model deteksi aktivitas merokok ini dibuat dengan beberapa parameter yang sebelumnya sudah ditentukan nilainya, seperti pada tabel 4.1 berikut:

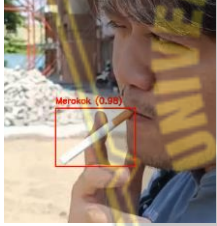
Tabel 4. 1 *Hyperparameter*

Parameter	Nilai
Batch Size	2
Epoch	25
Learning Rate	0.00003
Optimizer	SGD

Tabel 4.1 menampilkan konfigurasi *hyperparameter* yang digunakan dalam proses pelatihan model deteksi aktivitas merokok. *Batch size* ditetapkan sebesar 2, yang berarti model memproses dua gambar sekaligus dalam satu iterasi selama pelatihan, membantu dalam stabilisasi pembaruan bobot meskipun dengan sumber daya terbatas. Proses pelatihan dilakukan selama 25 *epoch*, di mana satu epoch merepresentasikan satu kali siklus penuh melalui seluruh dataset pelatihan, memberikan model kesempatan yang cukup untuk belajar dari data. *Learning rate* yang digunakan adalah sebesar 0.00003, nilai yang kecil ini bertujuan agar proses pembaruan bobot berlangsung secara perlahan dan stabil, sehingga dapat menghindari resiko *overshooting* atau tidak *konvergen*. Adapun algoritma optimisasi yang digunakan adalah *SGD (Stochastic Gradient Descent)*, yang merupakan optimizer paling umum digunakan untuk melatih model *machine learning* dan *deep learning*, terutama *neural network*.

Dengan menggunakan parameter dengan nilai tersebut, diperoleh hasil pengujian sebagaimana dalam tabel 4.2 dan tabel 4.4 di bawah ini:

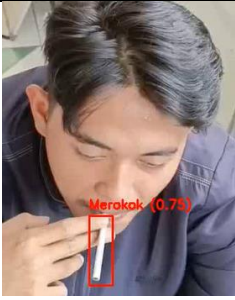
Tabel 4. 2 Hasil pengujian *faster r-cnn*

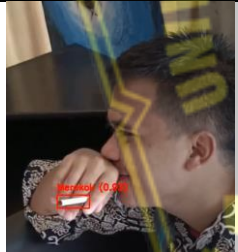
No	Hasil	Pred	Act	Conf. Score	Threshold				
					0.5	0.6	0.7	0.8	0.9
1		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
2		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
3		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
4		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
5		Merokok	Merokok	0.72	TP	TP	TP	FN	FN

6		Merokok	Merokok	0.85	TP	TP	TP	TP	FN
7		Merokok	Merokok	0.89	TP	TP	TP	TP	FN
8		Merokok	Merokok	0.82	TP	TP	TP	TP	FN
9		Merokok	Merokok	0.52	TP	FN	FN	FN	FN
10		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
11		Merokok	Merokok	0.51	TP	FN	TN	FN	FN

12		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
13		Merokok	Merokok	0.95	TP	TP	TP	TP	TP
14		Merokok	Merokok	0.92	TP	TP	TP	TP	TP
15		Merokok	Merokok	0.85	TP	TP	TP	TP	FN
16		Merokok	Merokok	0.98	TP	TP	TP	TP	TP
17		Merokok	Merokok	0.97	TP	TP	TP	TP	TP

18		Merokok	Merokok	0.76	TP	TP	TP	FN	FN
19		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
20		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
21		Merokok	Merokok	0.96	TP	TP	TP	TP	TP
22		Merokok	Merokok	0.82	TP	TP	TP	TP	FN
23		Merokok	Merokok	0.94	TP	TP	TP	TP	TP

24		Merokok	Merokok	0.75	TP	TP	TP	FN	FN
25		Merokok	Merokok	0.53	TP	FN	TN	TN	FN
26		Merokok	Merokok	0.94	TP	TP	TP	TP	TP
27		Merokok	Merokok	0.96	TP	TP	TP	TP	TP
28		Merokok	Merokok	0.98	TP	TP	TP	TP	TP

29		Merokok	Merokok	0.94	TP	TP	TP	TP	TP
30		Merokok	Merokok	0.95	TP	TP	TP	TP	TP
31		Merokok	Merokok	0.93	TP	TP	TP	TP	TP
32		Merokok	Merokok	0.93	TP	TP	TP	TP	TP
33		Merokok	Merokok	0.79	TP	TP	TP	FN	FN
34		Merokok	Merokok	0.78	TP	TP	TP	FN	FN

35		Merokok	Merokok	0.78	TP	TP	TP	FN	FN
36		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
37		Merokok	Merokok	0.96	TP	TP	TP	TP	TP
38		Merokok	Merokok	0.98	TP	FP	TP	TP	TP
39		Merokok	Merokok	0.71	TP	TP	TP	FN	FN

40		Merokok	Merokok	0.97	TP	TP	TP	TP	TP
----	---	---------	---------	------	----	----	----	----	----

Adapun hasil dari Evaluasi model *faster r-cnn* pengujian yang dilakukan diperoleh hasil berikut:

1. Nilai *threshold* 0.5 membuat sebanyak 40 *true positive*, 0 *false positive*, 0 *true negative*, dan 0 *false negative*. Maka, diperoleh *accuracy* 100%, *precision* 100%, *recall* 100%, dan *f1-score* 100%.
2. Nilai *threshold* 0.6 membuat sebanyak 36 *true positive*, 1 *false positive*, 0 *true negative*, dan 3 *false negative*. Maka, diperoleh nilai *accuracy* 90%, *precision* 97,2%, *recall* 92,3%, dan *f1-score* 94,6%.
3. Nilai *threshold* 0.7 membuat sebanyak 37 *true positive*, 0 *false positive*, 0 *true negative*, dan 3 *false negative*. Maka, diperoleh nilai *accuracy* 92,5%, *precision* 100%, *recall* 92,5%, dan *f1-score* 96,1%.
4. Nilai *threshold* 0.8 membuat sebanyak 30 *true positive*, 0 *false positive*, 0 *true negative*, dan 10 *false negative*. Maka, diperoleh nilai *accuracy* 72,5%, *precision* 100%, *recall* 72,5%, dan *f1-score* 84,1%.
5. Nilai *threshold* 0.9 membuat sebanyak 25 *true positive*, 0 *false positive*, 0 *true negative*, 15 *false negative*. Maka, diperoleh nilai *accuracy* 60%, *precision* 100%, *recall* 60%, dan *f1-score* 75%.

Tabel 4. 3 Hasil evaluasi *faster r-cnn*

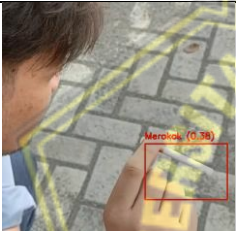
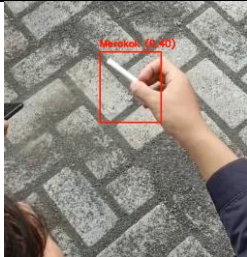
Nilai	Threshold				
	0.5	0.6	0.7	0.8	0.9
<i>True Positive</i>	40	36	37	30	25
<i>False Positive</i>	0	1	0	0	0
<i>True Negative</i>	0	0	0	0	0
<i>False Negative</i>	0	3	3	10	15
<i>Accuracy</i>	1	0.90	0.925	0.725	0.6
<i>Precision</i>	1	0.972	1	1	1
<i>Recall</i>	1	0.923	0.925	0.725	0.6
<i>F1-Score</i>	1	0.946	0.961	0.841	0.75

Berdasarkan tabel 4.3 hasil evaluasi performa model pada berbagai nilai *threshold* (0.5 hingga 0.9), dapat dilakukan analisis mendalam terhadap performa model dalam mendeteksi. Tabel ini mencakup metrik evaluasi seperti *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, *False Negative (FN)*, *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Pada *threshold* 0.5, model menunjukkan performa sempurna dengan nilai *True Positive* sebanyak 40, tanpa adanya *False Positive* maupun *False Negative*, serta nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* semuanya bernilai 1.0. Ini menunjukkan bahwa pada *threshold* ini, model mampu mengidentifikasi seluruh data positif dengan sangat baik tanpa melakukan kesalahan prediksi. Namun demikian, tidak adanya *True Negative* maupun *False Positive* di semua *threshold* menunjukkan bahwa data uji mungkin hanya berisi sampel positif, atau model tidak pernah mendeteksi data sebagai negatif, sehingga analisis terhadap performa dalam membedakan kelas negatif tidak dapat dilakukan. Ketika *threshold* ditingkatkan menjadi 0.6, terlihat adanya penurunan *True Positive* menjadi 36 dan kemunculan 3 *False Negative*, yang menyebabkan *Recall* menurun menjadi 0.923. Meskipun demikian, *Precision* masih tinggi pada 0.972, dan *F1-Score* juga relatif tinggi di angka 0.946. Hal ini menunjukkan bahwa meskipun beberapa prediksi positif mulai terlewat

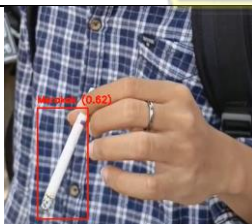
(menjadi *FN*), sebagian besar prediksi positif masih benar (*TP*), dan secara keseluruhan performa model masih baik. Pada threshold 0.7, terdapat sedikit peningkatan pada *True Positive* (menjadi 37), namun *False Negative* tetap 3, menghasilkan *Recall* 0.925 dan *Precision* 1.0. Nilai *F1-Score* tertinggi dicapai di *threshold* ini, yaitu 0.961, yang menunjukkan bahwa keseimbangan antara *Precision* dan *Recall* sangat baik. Ini bisa menjadi indikasi bahwa *threshold* 0.7 adalah nilai optimal dalam konteks ini. Ketika *threshold* dinaikkan lebih tinggi ke 0.8 dan 0.9, performa model semakin menurun. Pada *threshold* 0.8, *True Positive* turun drastis menjadi 29, dan *False Negative* naik menjadi 11. Hal ini menyebabkan *Recall* turun ke 0.725 dan *F1-Score* ke 0.841, meskipun *Precision* masih tetap 1.0. Hal ini menunjukkan bahwa model menjadi terlalu ketat dalam menentukan prediksi positif, sehingga banyak data positif tidak dikenali (*FN* meningkat). Penurunan lebih lanjut terlihat pada *threshold* 0.9, dengan hanya 24 *True Positive* dan 16 *False Negative*, sehingga *Recall* menurun signifikan menjadi 0.6 dan *F1-Score* menjadi hanya 0.75. Secara keseluruhan, analisis ini menunjukkan bahwa nilai *threshold* sangat mempengaruhi performa model. *Threshold* rendah seperti 0.5 memberikan performa yang sangat tinggi pada dataset ini, sedangkan *threshold* tinggi menyebabkan model kehilangan banyak prediksi positif yang benar. Nilai *threshold* 0.7 dapat dianggap sebagai kompromi terbaik antara *Precision* dan *Recall*, sebagaimana terlihat dari nilai *F1-Score* yang tertinggi.

Tabel 4. 4 Hasil pengujian *single shot detector*

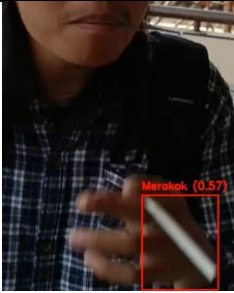
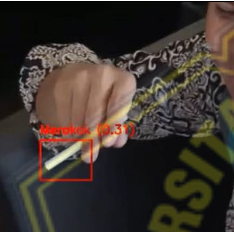
No	Hasil	Pred	Act	Conf. Score	Threshold				
					0.5	0.6	0.7	0.8	0.9
1		Merokok	Merokok	0.33	FN	FN	FN	FN	FN
2		Merokok	Merokok	0.60	TP	TP	FN	FN	FN
3		Merokok	Merokok	0.52	TP	FN	FN	FN	FN
4		Merokok	Merokok	0.48	FN	FN	FN	FN	FN
5		Merokok	Merokok	0.30	FN	FN	FN	FN	FN

6		Merokok	Merokok	0.36	FN	FN	FN	FN	FN
7		Merokok	Merokok	0.63	TP	TP	FN	FN	FN
8		Merokok	Merokok	0.36	FN	FN	FN	FN	FN
9		Merokok	Merokok	0.68	TP	TP	FN	FN	FN
10		Merokok	Merokok	0.64	TP	TP	FN	FN	FN
11		Merokok	Merokok	0.40	FN	FN	FN	FN	FN

12		Merokok	Merokok	0.63	TP	TP	FN	FN	FN
13		Merokok	Merokok	0.31	FN	FN	FN	FN	FN
14		Merokok	Merokok	0.74	TP	TP	TP	FN	FN
15		Merokok	Merokok	0.30	FN	FN	FN	FN	FN
16		Merokok	Merokok	0.38	FN	FN	FN	FN	FN
17		Merokok	Merokok	0.31	FN	FN	FN	FN	FN

18		Merokok	Merokok	0.34	FN	FN	FN	FN	FN
19		Merokok	Merokok	0	FN	FN	FN	FN	FN
20		Merokok	Merokok	0.60	TP	TP	FN	FN	FN
21		Merokok	Merokok	0.60	TP	TP	FN	FN	FN
22		Merokok	Merokok	0.62	TP	TP	FN	FN	FN
23		Merokok	Merokok	0.62	TP	TP	FN	FN	FN

24		Merokok	Merokok	0	FN	FN	FN	FN	FN
25		Merokok	Merokok	0.30	FN	FN	FN	FN	FN
26		Merokok	Merokok	0.72	TP	TP	FN	FN	FN
27		Merokok	Merokok	0	FN	FN	FN	FN	FN
28		Merokok	Merokok	0.36	FN	FN	FN	FN	FN
29		Merokok	Merokok	0.41	FN	FN	FN	FN	FN

30		Merokok	Merokok	0.57	TP	FN	FN	FN	FN
31		Merokok	Merokok	0	FN	FN	FN	FN	FN
32		Merokok	Merokok	0.31	FN	FN	FN	FN	FN
33		Merokok	Merokok	0.79	TP	TP	TP	FN	FN
34		Merokok	Merokok	0	FN	FN	FN	FN	FN
35		Merokok	Merokok	0.35	FN	FN	FN	FN	FN

36		Merokok	Merokok	0.49	FN	FN	FN	FN	FN
37		Merokok	Merokok	0.89	TP	TP	TP	TP	FN
38		Merokok	Merokok	0.38	FN	FN	FN	FN	FN
39		Merokok	Merokok	0.66	TP	TP	FN	FN	FN
40		Merokok	Merokok	0.64	TP	TP	FN	FN	FN

Adapun dari hasil pengujian *single shot detector* diperoleh hasil berikut:

1. Nilai *threshold* 0.5 membuat 17 *true positive*, 0 *false positive*, 0 *true negative*, dan 23 *false negative*. Maka, diperoleh nilai *accuracy* 42,5%, *precision* 100%, *recall* 42,5%, dan *f1-score* 59,6%.
2. Nilai *threshold* 0.6 membuat 15 *true positive*, 0 *false positive*, 0 *true negative*, dan 25 *false negative*. Maka, diperoleh nilai *accuracy* 37,5%, *precision* 100%, *recall* 37,5%, dan *f1-score* 54,5%.
3. Nilai *threshold* 0.7 membuat 3 *true positive*, 0 *false positive*, 0 *true negative*, dan 37 *false negative*. Maka, diperoleh nilai *accuracy* 7,5%, *precision* 100%, *recall* 7,5%, dan *f1-score* 13,9%.
4. Nilai *threshold* 0.8 membuat 1 *true positive*, 0 *false positive*, 0 *true negative*, dan 49 *false negative*. Maka, diperoleh nilai *accuracy* 2,5%, *precision* 100%, *recall* 2,5%, dan *f1-score* 4,8%.
5. Nilai *threshold* 0.9 membuat 0 *true positive*, 0 *false positive*, 0 *true negative*, 40 *false negative*. Maka, diperoleh nilai *accuracy* 0%, *precision* 0%, *recall* 0%, dan *f1-score* 0%.

Tabel 4. 5 Hasil evaluasi *single shot detector*

Nilai	Threshold				
	0.5	0.6	0.7	0.8	0.9
<i>True Positive</i>	17	15	3	1	0
<i>False Positive</i>	0	0	0	0	0
<i>True Negative</i>	0	0	0	0	0
<i>False Negative</i>	23	25	37	39	40
<i>Accuracy</i>	0.425	0.375	0.075	0.025	0
<i>Precision</i>	1	1	1	1	0
<i>Recall</i>	0.425	0.375	0.075	0.025	0
<i>F1-Score</i>	0.596	0.545	0.139	0.048	0

Berdasarkan tabel 4.5 evaluasi performa model pada berbagai nilai *threshold* (0.5 hingga 0.9), terlihat adanya penurunan signifikan dalam kemampuan model untuk mendeteksi kelas positif seiring meningkatnya nilai *threshold*. Pada *threshold* 0.5, model mampu mendeteksi 17 data sebagai *True Positive* dengan 23 *False Negative*, tanpa adanya *False Positive* maupun *True Negative*. Hal ini menghasilkan *Accuracy* sebesar 0.425, *Precision* sempurna (1.0), *Recall* sebesar 0.425, dan *F1-Score* sebesar 0.596. Walaupun nilai *Precision* tetap 1.0 di semua *threshold* (hingga *threshold* 0.8), hal ini tidak memberikan gambaran menyeluruh karena tidak disertai dengan peningkatan performa lainnya, seperti *Recall* atau *Accuracy*. Nilai *Precision* yang sempurna justru terjadi karena tidak adanya *False Positive*, bukan karena kemampuan deteksi model yang benar-benar akurat secara keseluruhan. Ketika *threshold* dinaikkan menjadi 0.6, jumlah *True Positive* menurun menjadi 15 dan *False Negative* meningkat menjadi 25. Hal ini menunjukkan bahwa semakin tinggi *threshold*, semakin banyak kasus positif yang gagal dikenali oleh model. Akibatnya, *Recall* menurun menjadi 0.375 dan *F1-Score* turun menjadi 0.545. Penurunan performa semakin tajam pada *threshold* 0.7, di mana *True Positive* hanya 3, sedangkan *False Negative* melonjak menjadi 37. *Accuracy* pun turun drastis menjadi 0.075, *Recall* tinggal 0.075, dan *F1-Score* anjlok ke angka 0.139. Pada *threshold* 0.8 dan 0.9, performa model menjadi sangat buruk, dengan hanya 1 *True Positive* dan kemudian tidak ada sama sekali (*True Positive* = 0 pada *threshold* 0.9), sementara *False Negative* mencapai angka maksimal yaitu 40. Ini menyebabkan metrik performa seperti *Accuracy*, *Recall*, dan *F1-Score* turun ke titik terendah, masing-masing menjadi 0.025, 0.025, dan 0.048 pada *threshold* 0.8, dan semuanya menjadi 0 pada *threshold* 0.9. Secara umum, hasil ini menunjukkan bahwa model memiliki sensitivitas yang sangat rendah terhadap kelas positif ketika *threshold* dinaikkan, yang berdampak pada penurunan drastis *Recall* dan *F1-Score*. Meskipun *Precision* tetap tinggi karena tidak adanya *False Positive*, hal ini menyesatkan karena model sebenarnya gagal dalam mendeteksi sebagian besar data positif. Idealnya, model yang baik harus memiliki

keseimbangan antara *Precision* dan *Recall*, namun dalam kasus ini, performa model sangat tidak seimbang, terutama pada threshold di atas 0.6. *Threshold* optimal tampaknya berada di angka 0.5, meskipun performanya pun belum sepenuhnya baik, karena setengah dari data positif masih terlewat ($FN = 23$).



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini memfokuskan pada perbandingan dua algoritma deteksi objek yang populer, yakni *Faster R-CNN* dan *Single Shot Detector (SSD)*, untuk mengetahui mana yang lebih optimal dalam konteks tersebut. Dataset yang digunakan dalam penelitian ini diperoleh dari *Roboflow*, dengan total 1935 gambar beranotasi aktivitas merokok. Data ini kemudian dibagi menjadi data pelatihan dan validasi dengan rasio 85:15. Proses pelatihan model dilakukan di *Google Colab*, menggunakan *framework deep learning* berbasis *Python*. Setiap model diuji berdasarkan beberapa metrik evaluasi utama, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*, untuk memberikan gambaran yang komprehensif terhadap performa deteksi masing-masing algoritma.

Hasil pelatihan menunjukkan bahwa model *Faster R-CNN* memiliki performa yang sangat baik. Pada nilai *threshold* 0.5, model ini mencapai skor sempurna di semua metrik evaluasi, yaitu 100% pada *accuracy*, *precision*, *recall*, dan *f1-score*. Bahkan ketika nilai *threshold* ditingkatkan menjadi 0.7, performanya masih tergolong sangat tinggi, dengan *f1-score* mencapai 96,1%. Ini mengindikasikan bahwa *Faster R-CNN* tidak hanya mampu mendeteksi objek rokok secara akurat, tetapi juga konsisten dalam berbagai skenario. Penurunan performa baru terlihat signifikan ketika *threshold* dinaikkan ke 0.8 atau 0.9, yang menandakan bahwa model menjadi terlalu selektif dan mulai kehilangan banyak deteksi positif.

Sebaliknya, *SSD* menunjukkan hasil yang jauh lebih rendah dalam hal akurasi dan konsistensi deteksi. Pada *threshold* 0.5, *SSD* hanya mampu mencapai akurasi sebesar 42,5% dengan *recall* yang sama, meskipun *precision* tetap tinggi (100%) karena tidak ada deteksi positif yang salah (*false positive*). Namun, nilai *recall* yang rendah menunjukkan bahwa model gagal mendeteksi sebagian besar kasus aktivitas merokok yang sebenarnya ada dalam data uji. Kinerja *SSD* semakin menurun drastis ketika *threshold*

dinaikkan, hingga mencapai akurasi 0% pada *threshold* 0.9. Hal ini menunjukkan bahwa *SSD* sangat sensitif terhadap pengaturan ambang batas dan kurang stabil dalam mendeteksi objek dalam konteks yang kompleks seperti aktivitas merokok.

Perbedaan performa antara kedua model ini dapat dijelaskan dari segi arsitektur. *Faster R-CNN* memiliki dua tahap dalam proses deteksinya, yakni *Region Proposal Network (RPN)* untuk menghasilkan kandidat lokasi objek dan klasifikasi akhir. Ini membuatnya lebih teliti dan akurat, meskipun dengan waktu komputasi yang sedikit lebih lama. Sebaliknya, *SSD* adalah model satu tahap yang lebih cepat karena langsung memprediksi *bounding box* dan kelas dalam satu proses. Namun, efisiensinya mengorbankan akurasi, terutama pada objek kecil dan kompleks seperti rokok yang mungkin tersembunyi sebagian atau tampak samar di gambar.

Secara keseluruhan, dapat disimpulkan bahwa *Faster R-CNN* lebih unggul secara signifikan dibandingkan *SSD* dalam hal mendeteksi aktivitas merokok di lingkungan kampus. Akurasi yang tinggi, konsistensi performa, dan ketahanannya terhadap variasi *threshold* menjadikan *Faster R-CNN* pilihan yang lebih cocok untuk diimplementasikan dalam sistem pendeteksian otomatis untuk mendukung kawasan bebas rokok. Sementara itu, meskipun *SSD* menjanjikan kecepatan, keterbatasannya dalam akurasi membuatnya kurang dapat diandalkan dalam konteks pengawasan kebijakan kampus yang membutuhkan ketelitian.

Sebagai rekomendasi, pengembangan sistem deteksi aktivitas merokok di masa depan dapat mempertimbangkan penggunaan *Faster R-CNN* sebagai model utama. Namun, optimalisasi dapat dilakukan untuk mempercepat proses inferensi tanpa mengorbankan akurasi, misalnya dengan memanfaatkan perangkat keras yang lebih mumpuni atau mengintegrasikan model dengan teknologi *edge computing*. Selain itu, pengumpulan *dataset* yang lebih besar dan beragam, serta pengujian di kondisi pencahayaan dan sudut kamera yang berbeda, juga penting untuk meningkatkan performa model dalam kondisi nyata.

5.2 Saran

Untuk pendeteksi rokok dilingkungan kampus yang mengutamakan keakuratan dalam mendeteksi objek positif, disarankan menggunakan *Faster R-CNN*, terutama pada *threshold* 0.7 yang menunjukkan performa paling optimal. Selain itu, disarankan untuk mengevaluasi dataset lebih lanjut dengan menambahkan data negatif agar *True Negative (TN)* dan *Accuracy* dapat dihitung secara lebih representatif. Apabila kecepatan inferensi menjadi faktor penting, *SSD* dapat dipertimbangkan dengan modifikasi arsitektur atau pelatihan lanjutan (*fine-tuning*) agar sensitivitas terhadap objek positif meningkat. Namun secara umum, prioritas tetap sebaiknya diberikan kepada model dengan keseimbangan metrik yang baik untuk hasil yang lebih andal. Untuk pengembangan ke depan, disarankan untuk mengeksplorasi penggunaan backbone lain guna meningkatkan akurasi sekaligus efisiensi komputasi. Selain itu, penambahan variasi data latih. Penggunaan teknik *augmentasi* data yang lebih beragam juga penting untuk memperkuat kemampuan generalisasi model. Evaluasi model sebaiknya diperluas dengan menggunakan dataset yang lebih banyak dan lengkap. Dengan menerapkan rekomendasi-rekomendasi tersebut, diharapkan sistem pendeteksi ini bisa terus diperbaiki kinerjanya dan memberikan keuntungan yang besar bagi masyarakat, terutama dalam mengurangi jumlah pelanggaran terkait aktivitas merokok di tempat-tempat yang dilarang.

DAFTAR PUSTAKA

- Apendi, Siroojuddin, Casi Setianingsih, Marisa W. Paryasto, Sistem Deteksi, Bahasa Isyarat, Deep Learning, Real Time, A. Bahasa Isyarat, B. Sistem Isyarat, and Bahasa Indonesia. 2023. "Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector." 10(1):249–55.
- Audina, Mela Tri, Fitri Utaminingrum, and Dahnia Syauqi. 2021. "Sistem Deteksi Dan Klasifikasi Jenis Kendaraan Berbasis Citra Dengan Menggunakan Metode Faster-RCNN Pada Raspberry Pi 4B." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* 5(2):814–19.
- Charli, Fino, Hadi Syaputra, Muhammad Akbar, Siti Sauda, and Febriyanti Panjaitan. 2020. "Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird." *Journal of Information Technology Ampera* 1(3):185–97. doi: 10.51519/journalita.volume1.issuue3.year2020.page185-197.
- Cheng, Lu, Yicai Ji, Chao Li, Xiaojun Liu, and Guangyou Fang. 2022. "Improved SSD Network for Fast Concealed Object Detection and Recognition in Passive Terahertz Security Images." *Scientific Reports* 1–16. doi: 10.1038/s41598-022-16208-0.
- Fikri, Fidra Irfan, and Casi Setianingsih. 2023. "Sistem Pendeteksi Sapi Pada Peternakan Dengan Citra Dan Video UAV Menggunakan Metode Single Shot Detector." 10(5):4600–4607.
- Indaryanto, Faizal, Anan Nugroho, and Alfa Faridh Suni. 2021. "Aplikasi Penghitung Jarak Dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19." *Edu Komputika Journal* 8(1):31–38. doi: 10.15294/edukomputika.v8i1.47837.
- Juweha. 2019. "Pengaruh Larangan Merokok Terhadap Kepatuhan Mahasiswa Untuk Tidak Merokok Di Areal Kampus Universitas Muhammadiyah Mataram." *Journal of Chemical Information and Modeling* 53(9):1–58.
- Muhdar, Grace Tedy Tulak, Bangu, Rosmiati, Iis Afrianty. 2022. "Edukasi Dampak

- Rokok Terhadap Kesehatan Remaja Di Kabupaten Kolaka.” 33(1):1–12.
- Mulyana, Dadang Iskandar, and Muhammad Zikri. 2022. “Optimasi Mendeteksi Klasifikasi Citra Digital Logo Mobil Indonesia Dengan Metode Single Shot Multibox Detector.” 13(2):88–94.
- PARDEDE, JASMAN, and HENDRI HARDIANSAH. 2022. “Deteksi Objek Kereta Api Menggunakan Metode Faster R-CNN Dengan Arsitektur VGG 16.” *MIND Journal* 7(1):21–36. doi: 10.26760/mindjournal.v7i1.21-36.
- Poetro, Bagus Satrio Waluyo, Sri Mulyono, and Vani Aulia Pramesti. 2024. “Prediksi Penyakit Batu Ginjal Dengan Menerapkan Convolutional Neural Network.” 153–62.
- Pradana, Ivandi Christiani, Eko Mulyanto, and Reza Fuad Rachmadi. 2022. “Deteksi Senjata Genggam Menggunakan Faster R-CNN Inception V2.” *Jurnal Teknik ITS* 11(2). doi: 10.12962/j23373539.v11i2.86587.
- Puspitasari, Fifin Ayu, and Hotma Pangaribuan. 2025. “DETEKSI KALORI PADA CITRA MAKANAN DENGAN ALGORITMA SINGLE SHOT MULTIBOX DETECTOR.” 02.
- Putri, Sherien Trisnawaty Eka, and Achmad Fahrurrozi. 2022. “Pendeteksian Objek Pada Citra Hewan Karnivora Dan Herbivora Menggunakan Faster R-Cnn.” *Jurnal Ilmiah Informatika Komputer* 27(1):32–42. doi: 10.35760/ik.2022.v27i1.5858.
- Ramadhani, Tivany, Usna Aulia, and Winda Amelia Putri. 2023. “Bahaya Merokok Pada Remaja.” *Jurnal Ilmiah Kedokteran Dan Kesehatan* 3(1):185–95. doi: 10.55606/klinik.v3i1.2285.
- Ramadhanu, Agung, Raja Ayu Mahessya, Muhammad Raihan Zaky, Mokti Isra, Sistem Informasi, and Universitas Putra Indonesia YPTK Padang. 2023. “Penerapan Teknologi Machine Learning Dengan Metode Vader Pada Aplikasi Sentimen Tamu Di Hotel Dymens.” *JOISIE Journal Of Information System And Informatics Engineering* 7(1):165–73.
- Samratul Fuady, Nehru, and Gina Anggraeni. 2020. “Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera.” 3(2):39–43. doi: 10.33087/jepca.v3i2.38.

- Sudarto, Sudarto. 2020. "Budaya Akademik Islami Di Universitas Islam Sultan Agung Semarang Dalam Perspektif Islamisasi Ilmu." *Ta'dibuna: Jurnal Pendidikan Islam* 9(2):267. doi: 10.32832/tadibuna.v9i2.3526.
- Sutisna, Tori, Agung Rachmat Raharja, Eko Hariyadi, and Vito Hafizh Cahaya Putra. 2024. "Penggunaan Computer Vision Untuk Menghitung Jumlah Kendaraan Dengan Menggunakan Metode SSD (Single Shoot Detector)." *INNOVATIVE: Journal Of Social Science Research* 4:6060–67.
- UNISSULA. 2024. *Sultan Agung (Unissula)*. Vol. 6583584.

