

**DETEKSI PENGEMUDI KANTUK MENGGUNAKAN
ALGORITMA YOLOv8
TUGAS AKHIR**



ITEBA
Institut Teknologi Batam

OLEH:

MUHAMAD ARIE SETYA PUTRA PALA

1922021

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI BATAM**

2023

DETEKSI PENGEMUDI KANTUK MENGGUNAKAN ALGORITMA YOLOv8

TUGAS AKHIR

**DIAJUKAN SEBAGAI SALAH SATU SYARAT UNTUK MENDAPATKAN GELAR
SARJANA (STRATA-1) PADA PROGRAM STUDI TEKNIK KOMPUTER**



ITEBA
Institut Teknologi Batam

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI BATAM**

2023

DAFTAR ISI

DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Metode Penelitian	2
1.5 Tujuan dan Manfaat Penelitian	2
1.5.1 Tujuan	2
1.5.2 Manfaat	3
1.6 Sistematika Penulisan	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 YOLOV8.....	5
2.2 <i>FACE DETECTION</i>	7
2.3 <i>APPLE CORE ML</i>	7
2.4 DEEP LEARNING	9
2.5 <i>DATASET</i>	10
2.6 <i>Python</i>	12
2.7 Penelitian Terdahulu	13
BAB 3 Metodologi Penelitian.....	18
3.1 Metodologi Penelitian.....	18
3.1.1 Metode Kuantitatif.....	18
3.2 Kerangka Penelitian.....	19
3.2.1 Identifikasi Masalah.....	19
3.2.2 Studi Literatur	21
3.2.3 Pengumpulan Data	22
3.2.4 Pengolahan Data	22
3.2.5 Analisa Hasil.....	23
3.2.6 Implementasi.....	24
BAB 4 Analisa Perancangan.....	26
BAB 5 Hasil dan Pembahasan	26
BAB 6 Kesimpulan dan Saran	28
6.1 Kesimpulan	28
6.2 Saran	29
DAFTAR PUSTAKA	30

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pengenalan objek dalam citra dan video telah menjadi elemen kunci dalam berbagai aplikasi berbasis citra, termasuk di bidang keamanan, pengawasan lalu lintas, otomatisasi industri, kendaraan otonom, dan analisis visual. Identifikasi objek dan pelokalannya adalah tahap awal yang penting dalam pemrosesan informasi visual, dan keakuratan serta efisiensi deteksi objek menjadi faktor utama dalam keberhasilan aplikasi ini.

Metode YOLO (You Only Look Once) telah muncul sebagai salah satu pendekatan terkemuka dalam deteksi objek real-time. YOLO memiliki keunggulan yang signifikan karena mampu mengenali objek dalam satu proses inferensi, sehingga menjadikannya cepat dan efisien. Meskipun begitu, untuk mengintegrasikan YOLO ke dalam aplikasi berbasis iOS, diperlukan platform yang sesuai yang dapat menjalankan model machine learning dengan baik. Apple Core ML adalah salah satu solusi yang tepat untuk tugas ini. Core ML adalah framework yang dirancang khusus untuk memudahkan integrasi model machine learning ke dalam aplikasi iOS, sehingga membuka peluang besar bagi pengembang aplikasi iOS untuk meningkatkan kemampuan pengenalan objek dalam aplikasi mereka.

Penelitian ini bertujuan untuk mengintegrasikan model YOLO dalam lingkungan Apple Core ML untuk mengembangkan aplikasi iOS yang mampu mengenali objek dalam citra secara real-time. Integrasi ini akan memungkinkan pengguna iOS untuk mengakses kemampuan deteksi objek yang kuat dan akurat di dalam aplikasi mereka. Penelitian ini diharapkan dapat memberikan kontribusi penting dalam pengembangan aplikasi Mobile berbasis iOS yang cerdas dan efisien dalam pengenalan objek, yang dapat diaplikasikan dalam berbagai sektor seperti bisnis, industri, dan hiburan.

1.2 Perumusan Masalah

Adapun beberapa perumusan masalah berdasarkan latar belakang yang telah penulis sampaikan adalah sebagai berikut:

1. Bagaimana efektivitas dan efisiensi implementasi **Mengenal objek menggunakan metode YOLO** dalam situasi di mana koneksi internet mungkin terbatas atau tidak stabil?

2. Apa tantangan teknis utama yang dihadapi dalam mengembangkan dan menjalankan **Mengenal objek menggunakan metode YOLO**?
3. Bagaimana tingkat kualitas dan keakuratan dari respons **Mengenal objek menggunakan metode YOLO** dibandingkan dengan implementasi metode lain?
4. Apa saja perangkat lunak dan teknologi yang dapat digunakan untuk **Mengenal objek menggunakan metode YOLO** ini?

1.3 Batasan Masalah

1. Pengimplementasi tahap lanjutan dengan menjadikan penelitian dan aplikasi ini menjadi *API (Application Programmable Interface)*
2. *Interface* antar muka yang bagus, karena nantinya hanya sebatas *output* model yang sudah di *training*
3. Pertanggung jawaban keakuratan data dan jawaban

1.4 Metode Penelitian

Metode penelitian yang dipakai adalah **Studi Literatur**, yang merupakan pendekatan penelitian yang berfokus pada analisis, pengumpulan, dan sintesis informasi. Namun dalam penelitian ini penulis akan menjelaskan beberapa metode lain seperti:

1.5 Tujuan dan Manfaat Penelitian

1.5.1 Tujuan

Tujuan dari penelitian ini:

1. Mengetahui bagaimana cara kerja metode *YOLO* yang populer terhadap topik mengenal objek
2. Untuk menguji dan mengevaluasi model *YOLO* yang diintegrasikan dengan Apple Core ML. Hal ini melibatkan pengukuran kinerja model dalam hal akurasi, kecepatan, dan efisiensi.
3. Dapat mengembangkan API dari sistem *Mengenal objek menggunakan metode YOLO* yang sudah berjalan secara lokal untuk kebutuhan lainnya bahkan untuk dikomersialkan.

4. Dapat mengembangkan sebuah aplikasi iOS yang mampu mengenali objek secara *real-time* menggunakan kamera perangkat iOS bertujuan untuk kemudahan dalam penggunaan aplikasi dan memberikan pengenalan objek yang akurat.
5. Untuk mengintegrasikan model deteksi objek YOLO dengan *Apple Core ML* yang melibatkan pelatihan model YOLO dan konversi model tersebut agar dapat dijalankan di platform iOS dengan bantuan Core ML.

1.5.2 Manfaat

Adapun beberapa manfaat dari penelitian ini adalah sebagai berikut:

1. Kecepatan dan Akurasi: Integrasi YOLO dengan *Apple Core ML* berpotensi memberikan deteksi objek yang cepat dan akurat pada perangkat iOS. Hal ini akan menjadi keunggulan dalam aplikasi yang memerlukan deteksi objek *real-time*, seperti aplikasi pengawasan atau navigasi.
2. Penelitian ini juga akan membantu mempromosikan penggunaan *Apple Core ML* dalam pengembangan aplikasi iOS. Dengan demikian, akan ada peningkatan dalam pemanfaatan fitur machine learning dalam aplikasi *mobile*.

1.6 Sistematika Penulisan

Penulisan laporan tugas akhir terdiri dari 5 bab dengan penjelasan sebagai berikut:

BAB I: PENDAHULUAN

Berisi tentang latar belakang, rumusan dan batasan masalah, tujuan dan sistematika penulisan.

BAB II: LANDASAN TEORI

Berisi tentang teori-teori yang berhubungan dengan penelitian.

BAB III: METODOLOGI

Berisi tentang bagaimana penelitian dilakukan, mengumpulkan data, menganalisis data, dan mencapai tujuan penelitian.

BAB IV: HASIL DAN PEMBAHASAN

Berisi tentang hasil pengujian dan analisis sistem yang dibuat dan dibandingkan dengan dasar teori dan sistem lain yang dijadikan sebagai pembanding.

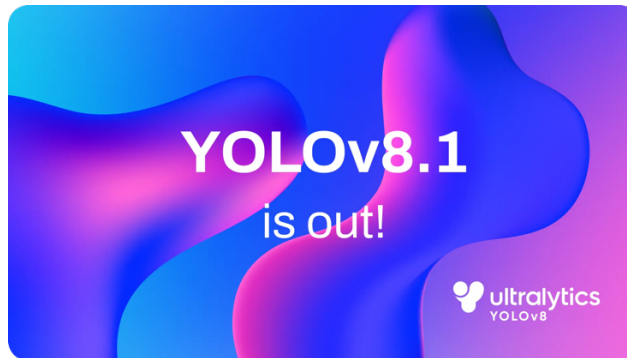
BAB V: PENUTUP

Bagian ini memuat kesimpulan dan saran-saran dari perancangan sistem, berikut juga keterbatasan yang ditemukan dan asumsi-asumsi dari penulis selama penelitian berlangsung.

EAK MAU SIAPPP

BAB 2 TINJAUAN PUSTAKA

2.1 YOLOV8



YOLO (*You Only Look Once*) adalah sebuah algoritma dalam bidang pengenalan objek pada citra yang sangat populer di bidang *computer vision* dan *machine learning*. Seri algoritma YOLO adalah salah satu algoritma dengan pertumbuhan tercepat dan terbaik hingga saat ini, terutama algoritma YOLOv8 yang baru dirilis pada tahun 2023, yang telah mencapai tingkat akurasi tertinggi hingga saat ini. YOLOv8, yang diterbitkan pada tahun 2023, dirancang untuk menggabungkan keunggulan dari berbagai detektor objek *real-time*. YOLOv8 mendukung semua versi YOLO dan dapat beralih antara versi yang berbeda dengan mudah. Algoritma ini juga dapat berjalan di berbagai platform perangkat keras (CPU-GPU), memberikan fleksibilitas yang luar biasa[1].

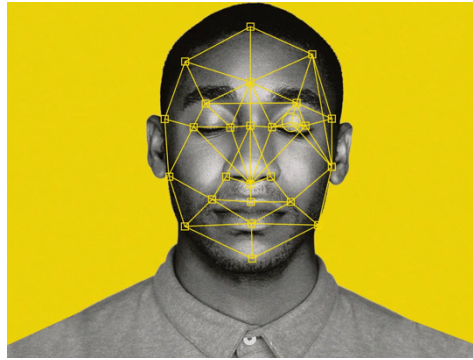
YOLO (*You Only Look Once*), sebuah *object detection model* dan *image segmentation* yang dikembangkan oleh Joseph Redmon dan Ali Farhadi di Universitas Washington, diperkenalkan pada tahun 2015. Model ini dengan cepat menjadi populer berkat kecepatannya yang tinggi dan tingkat akurasi yang sangat baik. Joseph dan Ali memperkenalkan pendekatan baru untuk deteksi objek dengan mengubahnya menjadi masalah regresi satu langkah. Dimulai dari *pixel* gambar, metode ini memprediksi *bounding box* dan *class probabilities* secara bersamaan. Konsep '*unified*' yang digunakan memungkinkan prediksi beberapa *bounding box* dan probabilitas kelas secara simultan, sehingga meningkatkan kecepatan dan akurasi *object detection*[2]. Kehadiran YOLO memberikan dampak yang signifikan dalam meningkatkan potensi pada *computer vision*[2].

Berikut sedikit ringkasan mengenai perkembangan YOLO pada setiap versi :

1. YOLO-v1 (2015): Arsitektur pertama yang menggabungkan prediksi *bounding box* dan klasifikasi dalam satu tugas, memungkinkan deteksi objek secara lebih efisien dalam satu langkah[2].

2. YOLO-v2 (2016): Memperkenalkan perbaikan arsitektur seperti normalisasi *batch* untuk stabilitas pelatihan yang lebih baik, resolusi gambar yang lebih tinggi untuk deteksi lebih akurat, dan penggunaan kotak jangkar (*anchor boxes*) untuk menangani berbagai ukuran objek[2].
3. YOLO-v3 (2018): Dilengkapi dengan skor *objectness* untuk memperbaiki prediksi *bounding box*, penambahan koneksi untuk lapisan *backbone* yang meningkatkan deteksi, dan kemampuan untuk menghasilkan prediksi pada tiga tingkat granularitas, meningkatkan kinerja pada objek kecil[2].
4. YOLO-v4 (2020): Dikembangkan oleh tim baru, menambahkan penggabungan fitur yang lebih baik, '*bag of freebies*' yang mencakup teknik gratis untuk meningkatkan performa, dan *mish activation* yang meningkatkan kinerja jaringan[2].
5. YOLO-v5 (2020): Varian pertama tanpa makalah ilmiah, dibangun dengan PyTorch dan memiliki *repository* GitHub yang aktif, membuat implementasi lebih mudah diakses oleh pengembang dan peneliti[2].
6. YOLO-v6 (2021): Fokus pada efisiensi perangkat keras dengan desain ulang modul *backbone* (*EfficientRep*) dan *neck* (Rep-PAN), serta pengenalan konsep *anchor-free* dan kepala yang terpisah untuk meningkatkan kinerja keseluruhan[2].
7. YOLO-v7 (2022): Memperkenalkan reformasi arsitektur dengan mempertimbangkan memori yang diperlukan untuk menjaga lapisan dalam memori dan jarak untuk *back-propagate* gradien, serta implementasi E-ELAN untuk agregasi lapisan yang lebih baik[2].
8. YOLO-v8 (2023): Dirilis oleh Ultralytics, menunjukkan kinerja *throughput* yang menjanjikan dibandingkan dengan pendahulunya, dengan parameter komputasi yang serupa. Ini mencakup berbagai peningkatan yang akan dijelaskan lebih lanjut dalam makalah yang akan datang[2].

2.2 FACE DETECTION



Saat ini, teknologi pengenalan objek menggunakan kamera adalah salah satu topik paling menantang, di mana akurasi dan performa *real-time* telah menjadi indikator utama yang digunakan dalam sensor kamera[1]. *Face detection* merupakan salah satu masalah paling populer dalam *computer vision* yang melibatkan identifikasi wajah dalam gambar digital. Baru-baru ini, teknik *face detection* telah berkembang dari metode *computer vision* tradisional ke pendekatan *machine learning* (ML) yang lebih canggih. Tahapan utama dalam teknologi *face detection* meliputi menemukan area dalam gambar di mana terdapat wajah atau beberapa wajah[3].

Face detection adalah masalah *computer vision* yang melibatkan pencarian wajah dalam gambar. Ini juga merupakan langkah awal untuk banyak teknologi terkait wajah, seperti verifikasi wajah, pemodelan wajah, pelacakan pose kepala, pengenalan jenis kelamin dan usia, pengenalan ekspresi wajah[3] yang dapat membantu dalam melakukan deteksi kantuk pada pengemudi.

2.3 APPLE CORE ML

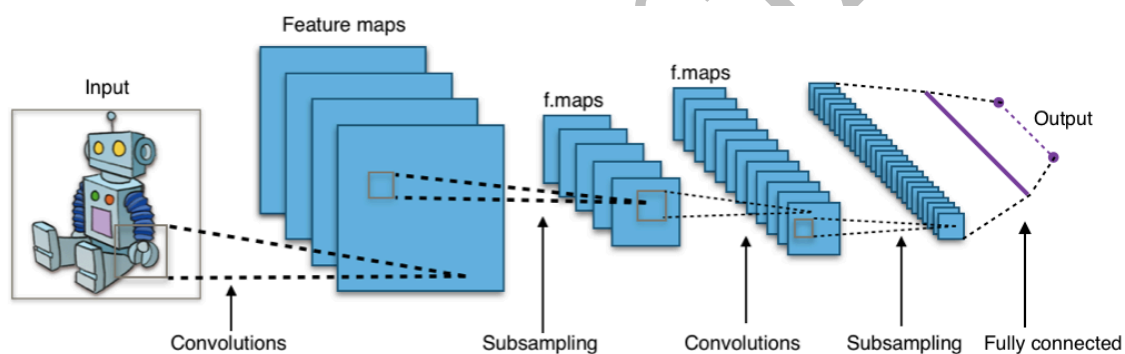


Apple Core ML, pertama kali diperkenalkan pada tahun 2017, merupakan *framework machine learning* yang memungkinkan pengembang untuk mengintegrasikan model *machine learning* ke dalam aplikasi mereka dengan lebih mudah dan efisien. *Core ML* dirancang untuk memanfaatkan perangkat keras dari perangkat *Apple*, termasuk CPU, GPU, dan *Neural Engine*,

guna memaksimalkan kinerja sambil meminimalkan penggunaan daya dan memori. Dengan *Core ML*, pengembang dapat membuat, melatih, dan mengonversi model *machine learning* dari berbagai pustaka pihak ketiga menggunakan paket *Python coremltools*, serta memanfaatkan *Xcode* untuk mengamankan dan mengoptimalkan model mereka.

Selain itu, *Core ML* menyediakan berbagai alat dan fitur untuk meningkatkan pengalaman pengembang dan pengguna. Alat-alat ini termasuk dukungan untuk model enkripsi, profil aplikasi untuk memantau panggilan API *Core ML*, serta *preview* langsung dari perilaku model pada data sampel atau dari kamera dan mikrofon perangkat. *Apple* juga menyediakan *Create ML*, alat yang memungkinkan pengembang untuk membangun dan melatih model *Core ML* langsung di *Mac* mereka tanpa perlu menulis kode. Dengan dukungan ekstensif dan dokumentasi yang lengkap, *Core ML* memungkinkan pengembang untuk dengan mudah mengintegrasikan teknologi *machine learning* yang canggih ke dalam aplikasi mereka, membuat aplikasi lebih pintar dan responsif terhadap kebutuhan pengguna.

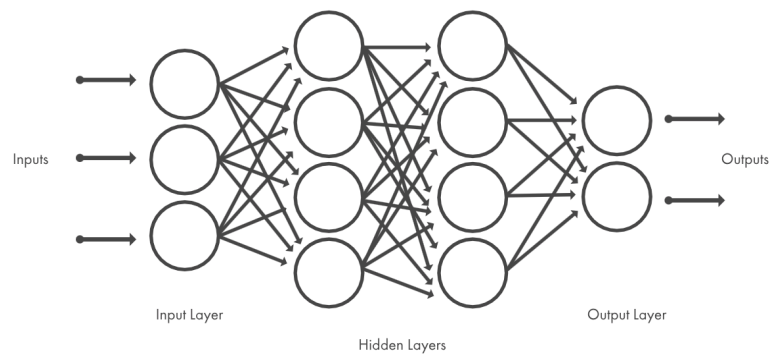
Convolutional Neural Networks



Selama sepuluh tahun terakhir, *Convolutional Neural Networks* (CNN) telah menjadi standar utama untuk berbagai operasi dalam bidang *computer vision* dan *machine learning*[4]. *Convolutional Neural Networks* (CNN) merupakan kelas dari jaringan saraf *deep feed-forward*, yang dibangun dengan meniru pola koneksi *neuron* di korteks visual manusia[5].

Convolutional Neural Networks dapat secara otomatis mempelajari fitur-fitur yang mewakili variasi visual kompleks dari data pelatihan yang banyak.[6]. Salah satu keberhasilan *Convolutional Neural Networks* adalah dalam pengenalan wajah. Pengenalan wajah oleh komputer adalah teknik yang memungkinkan komputer secara otomatis mengenali wajah dalam gambar. Berbagai peneliti telah melakukan penelitian terkait *face detection*[6].

2.4 DEEP LEARNING



Deep Learning (DL) adalah pencapaian terbaru dalam era *machine learning*, di mana awalnya mampu menunjukkan kemampuan mendekati manusia, dan kini kemampuan yang melampaui manusia dalam banyak aplikasi termasuk terjemahan suara-ke-teks, deteksi dan pengenalan objek, deteksi anomali, pengenalan emosi dari rekaman audio atau video, dan sebagainya[4].

Konsep paling penting dalam bidang *deep learning* terinspirasi oleh sistem biologis manusia yang cenderung fokus pada bagian-bagian khas saat memproses sejumlah informasi yang besar. Dengan perkembangan *deep neural networks*, perhatian mengenai mekanisme ini telah banyak digunakan dalam berbagai domain aplikasi yang sangat luas[7]. *Deep learning* adalah istilah yang digunakan untuk menyebut seperangkat metode dalam membangun jaringan saraf berlapis-lapis (atau '*deep*') yang mampu menyelesaikan masalah-masalah menantang dalam *supervised classification*, pemodelan generatif, atau pembelajaran mendalam[8].

Model *deep learning* memiliki banyak potensi pada ilmu jaringan (*neuroscience*). Salah satu yang paling menarik adalah kemampuannya melampaui perancangan fungsi secara manual dan memahami bagaimana komputasi berkembang dari pengalaman. *Neuroscientists* sudah menyadari peluang ini, namun pemanfaatannya baru saja dimulai[8].

2.5 DATASET



Dataset adalah kumpulan data yang terorganisir dan biasanya disimpan dalam bentuk tabel atau *database*, yang digunakan untuk analisis, pemrosesan, dan penelitian. Setiap *dataset* terdiri dari elemen-elemen yang dikenal sebagai '*data points*' atau '*records*', yang masing-masing memiliki sejumlah atribut atau variabel yang menjelaskan karakteristik data tersebut. Sebagai contoh, dalam *dataset* kesehatan, setiap rekaman mungkin berisi informasi tentang pasien seperti usia, jenis kelamin, diagnosis, dan hasil tes medis. *Dataset* dapat dikumpulkan dari berbagai sumber seperti eksperimen ilmiah, survei, sensor, transaksi bisnis, dan media sosial, dan sering kali dilengkapi dengan *metadata* yang menjelaskan struktur, asal, dan konteks data tersebut. Pengelolaan dan analisis *dataset* memainkan peran kunci dalam berbagai disiplin ilmu, termasuk ilmu data, *machine learning*, statistik, dan informatika, membantu peneliti dan profesional dalam membuat keputusan yang didukung oleh data dan mengembangkan wawasan baru.

Dataset telah muncul dalam ratusan makalah penelitian, di mana *dataset* tersebut digunakan sebagai dasar untuk pengembangan dan perbandingan banyak algoritma[9], *Datasets* sangat penting dalam ekosistem *machine learning*. Selain menyediakan data pelatihan dan pengujian bagi pembuat model, *datasets* juga merumuskan masalah, mengorganisir komunitas, dan menjembatani antara akademika dan industri[9].

Berikut adalah daftar peran dataset pada penelitian ini :

1. Sumber Data Utama:

Dataset yang Anda miliki, yang terdiri dari foto-foto pengemudi saat mengantuk dan sadar, merupakan sumber data utama yang digunakan untuk melatih dan menguji algoritma YOLOv8. Tanpa dataset ini, model deteksi kantuk tidak dapat dibuat atau diuji.

2. Pelatihan Model:

Dataset digunakan untuk melatih model YOLOv8 agar dapat mengenali dan mendeteksi tanda-tanda kantuk pada pengemudi. Dalam proses pelatihan, model akan mempelajari pola-pola visual yang terkait dengan kondisi mengantuk dan sadar berdasarkan gambar-gambar yang ada dalam *dataset*.

3. Validasi dan Pengujian:

Setelah model dilatih, *dataset* yang berbeda digunakan untuk validasi dan pengujian. Hal ini penting untuk mengevaluasi kinerja model dan memastikan bahwa model dapat mendeteksi kantuk dengan akurasi yang tinggi pada gambar-gambar yang belum pernah dilihat sebelumnya.

4. *Preprocessing* dan *Augmentasi*:

Dataset juga memainkan peran dalam tahap *preprocessing*, di mana gambar-gambar mungkin perlu diubah ukurannya, dinormalisasi, atau dilakukan *augmentasi* (seperti rotasi, *flipping*, atau perubahan kecerahan) untuk meningkatkan kualitas dan variasi data yang tersedia untuk pelatihan model.

5. Pemetaan Kelas:

Dataset harus diberi label dengan benar untuk memastikan bahwa setiap gambar dikategorikan dengan tepat sebagai "mengantuk" atau "sadar." Label ini sangat penting agar model dapat belajar membedakan antara kedua kondisi tersebut.

6. Evaluasi Model:

Kualitas dan keberagaman *dataset* mempengaruhi evaluasi model. *Dataset* yang baik harus mencakup berbagai kondisi pencahayaan, sudut pandang, dan variasi wajah untuk memastikan model yang dihasilkan dapat diandalkan dalam situasi dunia nyata.

7. Reproduksi Penelitian:

Dengan mendokumentasikan dan mungkin membagikan *dataset* (dengan memperhatikan privasi dan etika), penelitian Anda dapat direproduksi oleh peneliti lain, yang meningkatkan validitas dan kontribusi ilmiah dari skripsi Anda.

2.6 Python



Pada tahun 1991, Guido van Rossum menciptakan bahasa pemrograman *Python*. Ternyata, ada cerita menarik di balik penamaan "*Python*". Saat itu, Guido sedang membaca naskah seri BBC "*Monty Python's Flying Circus*". Terinspirasi oleh buku tersebut, ia memutuskan untuk memberi nama bahasa pemrogramannya "*Python*" karena terdengar singkat dan unik. *Python* adalah bahasa pemrograman yang berorientasi objek, diinterpretasikan, dan interaktif. Bahasa ini menawarkan berbagai struktur data tingkat tinggi seperti *list*, *tuple*, *set*, dan *array* asosiatif (disebut *dictionary*). *Python* juga mendukung pengetikan dan pengikatan dinamis, serta menyediakan modul, kelas, pengecualian, dan manajemen memori otomatis[10].

Selain itu, *Python* digunakan dalam sistem komputasi paralel dan terkenal dengan sintaksnya yang sederhana dan mudah dipahami, namun tetap kuat dan fleksibel. *Python* juga memiliki interpreter untuk Java yang dikenal sebagai *JPython*, mirip dengan interpreter untuk bahasa C. Salah satu keunggulan *Python* adalah ketersediaan berbagai pustaka yang luas, yang bisa mengurangi jumlah kode yang harus ditulis oleh *programmer* hingga sepertiga. Inilah salah satu alasan mengapa *Python* telah mencapai puncak popularitas dalam bidang Pembelajaran Mesin (*Machine Learning*)[10].

2.7 Penelitian Terdahulu

No	Judul	Penulis	Metode	Hasil
1	<i>IoT-Based Non-Intrusive Automated Driver Drowsiness Monitoring Framework for Logistics and Public Transport Applications to Enhance Road Safety</i>	M. ADIL KHAN, TAHIR NAWAZ , UMAR S. KHAN , (Member, IEEE), AMIR HAMZA, AND NASIR RASHID	Kuantitatif	Jurnal ini mengusulkan kerangka kerja IoT otomatis <i>end-to-end</i> untuk memantau perilaku pengemudi, dirancang khusus untuk aplikasi logistik dan transportasi publik. Sistem ini terdiri dari Raspberry Pi dan Nvidia Jetson Nano untuk <i>edge computing</i> yang terhubung dengan <i>webcam</i> untuk menangkap gambar secara <i>real-time</i> , <i>cloud computing</i> , dan aplikasi <i>mobile</i> . Deteksi kantuk dilakukan dengan menganalisis aktivitas mata dan mulut, seperti tidur dan menguap, menggunakan teknik pemrosesan gambar berbasis <i>Deep Neural Network</i> (DNN) dengan <i>multilayer perceptron classifiers</i> . Sistem ini memantau empat parameter perilaku pengemudi: aktif, menguap, mata tertutup, dan terdistraksi, dengan akurasi deteksi sebesar 96%. Data yang dikumpulkan dikirim ke <i>database real-time</i> di <i>cloud</i> , memungkinkan pemantauan jarak jauh melalui aplikasi Android interaktif, di mana admin dapat menambahkan beberapa pengemudi dan menerima notifikasi kantuk serta informasi terkait lainnya untuk evaluasi pengemudi.
2	<i>Drowsiness Detection Based On Driver</i>	F. Faraji, F. Lotfi, J. Khorramdel, A. Najafi, A.	Kuantitatif	Studi ini menggunakan pendekatan berbasis gambar dengan akurasi yang memadai dan proses yang cepat. Kami menerapkan

Temporal Behavior Using a New Developed Dataset Ghaffari K. N. Toosi
University of Technology,
Tehran, Iran.

YOLOv3 (*You Look Only Once-version3*) CNN (*Convolutional Neural Network*) untuk mengekstrak fitur wajah secara otomatis. Selanjutnya, jaringan neural LSTM (*Long-Short Term Memory*) digunakan untuk mempelajari perilaku temporal pengemudi termasuk periode waktu mengantuk dan kedipan mata serta klasifikasi urutan. Untuk melatih YOLOv3, kami menggunakan *dataset* yang dikumpulkan bersama metode *transfer learning*. Selain itu, dataset untuk proses pelatihan LSTM dihasilkan oleh CNN yang disebutkan dan diformat sebagai urutan dua dimensi yang terdiri dari durasi waktu kedipan mata dan mengantuk. *Dataset* yang dikembangkan mempertimbangkan gangguan seperti pencahayaan dan postur kepala pengemudi. Untuk eksperimen *real-time*, sebuah kerangka *multi thread* dikembangkan untuk menjalankan kedua CNN dan LSTM secara paralel. Hasilnya menunjukkan kemampuan hibrida CNN dan LSTM dalam mendeteksi kantuk serta efektivitas metode yang diusulkan.

3 *Vision Transformers and YoloV5 based Driver Drowsiness Detection Framework* Ghanta Sai Krishna,
Kundrapu Supriya, Jai
Vardhan and Mallikharjuna
Rao K

Kuantitatif

Metode yang digunakan dalam penelitian ini adalah penggunaan *framework* berbasis *vision transformers* dan arsitektur YOLOv5 untuk pengenalan kantuk pengemudi. Mereka mengusulkan arsitektur khusus *pre-trained* YOLOv5 untuk ekstraksi wajah

dengan tujuan mengekstraksi *Region of Interest* (ROI). Selain itu, karena keterbatasan arsitektur sebelumnya, mereka memperkenalkan *vision transformers* untuk klasifikasi gambar biner yang dilatih dan divalidasi pada *dataset* publik UTA-RLDD. Model ini mencapai akurasi pelatihan sebesar 96,2% dan akurasi validasi sebesar 97,4%. Selama pelatihan, teknik *augmentasi* gambar digunakan untuk menghasilkan variasi gambar yang dapat meningkatkan kinerja *framework* ViT. Dengan demikian, metode yang dicapai adalah akurasi validasi sebesar 97.4%.

- | | | | |
|--|---|--------------------|--|
| <p>4 Aplikasi <i>Warning</i>
 <i>Alert</i> Pendeteksi
 Kelelahan
 Ekspresi Wajah Pada
 Pengemudi Secara
 <i>Real-Time</i>
 Menggunakan Metode
 <i>You Only Look Once</i>
 Berbasis <i>Website</i></p> | <p>Hafidh Ahmad Fauzan
 , Ari Kurniawan</p> | <p>Kuantitatif</p> | <p>Penelitian yang menggunakan YOLOv4 untuk mendeteksi kelelahan pada pengemudi berdasarkan ekspresi wajah secara <i>real-time</i>. Algoritma pendeteksian diterapkan melalui platform <i>website</i> yang terhubung dengan Telegram, memungkinkan pengiriman pesan secara langsung mengenai status dan kondisi pengemudi dengan detail. Penulis memanfaatkan <i>dataset</i> yang tersedia di Kaggle, dipublikasikan oleh RAKIBUL.ECE.RUET, yang terdiri dari 4560 <i>file</i> per kelas, yaitu <i>active</i> dan <i>fatigue</i>. Selain itu, penulis juga menyiapkan <i>dataset</i> tambahan yang telah di-<i>augmentasi</i> menggunakan <i>photometric distortion</i> dan <i>geometric distortion</i> untuk meningkatkan variasi data. Melalui</p> |
|--|---|--------------------|--|

5	Implementasi Algoritma <i>Deep Learning</i> Untuk Sistem Deteksi Kantuk Pada Pengemudi Menggunakan YOLO	Mamta Anisa Bella	Kuantitatif	<p>pendekatan ini, penulis berhasil mengembangkan metode baru yang mencapai akurasi sebesar 97,03%, menunjukkan efektivitas sistem dalam mendeteksi kelelahan pengemudi secara akurat dan cepat.</p> <p>Dalam penelitian ini, pengemudi yang mengantuk diidentifikasi dengan label "<i>drowsy</i>", sementara pengemudi yang terjaga diberi label "<i>awake</i>". Metode yang digunakan memanfaatkan kombinasi YOLOv4 dan CNN, diimplementasikan melalui sebuah <i>website</i> yang terhubung langsung dengan layanan Telegram untuk mengirimkan status pengemudi secara <i>real-time</i>. <i>Dataset</i> yang digunakan diambil dari Kaggle, yang telah dipublikasikan oleh RAKIBUL.ECE.RUET, dengan masing-masing kelas memiliki 4560 <i>file</i>, serta <i>dataset</i> tambahan yang telah melalui proses <i>augmentasi</i>. Konfigurasi model termasuk penggunaan <i>batch size</i> 64, ukuran jaringan 416x416, subdivisi 16, <i>max batch</i> 4000, dan filter 21. Pembagian <i>dataset</i> 80%:20% dan <i>learning rate</i> 0.00261 memberikan nilai <i>Intersection over Union</i> (IoU) terbesar. Dengan metode ini, sistem berhasil mencapai akurasi sebesar 97,03%, menunjukkan efektivitasnya dalam mendeteksi kelelahan pengemudi dengan akurasi dan kecepatan yang tinggi.</p>
---	---	-------------------	-------------	--

6	<i>Driver's Drowsiness Detection System</i>	Aayush Bhetuwal, Siddanta K C	Kuantitatif	<p>Penelitian ini mempersembahkan sistem deteksi kantuk pengemudi yang inovatif dengan memanfaatkan kemampuan canggih YOLOv5 dan CNN. Menggunakan <i>dataset</i> yang terdiri dari 4560 gambar aktif dan 4560 gambar kelelahan yang dikumpulkan dari Kaggle, sistem ini menunjukkan kinerja yang luar biasa. Model YOLOv5s mencapai akurasi mengesankan sebesar 93,01% dan <i>Intersection over Union</i> (IoU) sebesar 95,51%, meskipun hanya dilatih selama 10 <i>epoch</i>. Kemampuan deteksi <i>real-time</i> dari sistem ini, dikombinasikan dengan tingkat akurasinya yang tinggi, menunjukkan potensi signifikan untuk meningkatkan keselamatan di jalan raya dengan memantau dan merespons kelelahan pengemudi secara efektif.</p>
---	---	-------------------------------	-------------	--

BAB 3

Metodologi Penelitian

3.1 Metodologi Penelitian

Dalam penelitian ini, tujuan utama adalah mengembangkan dan menguji model deteksi kantuk pada pengemudi menggunakan algoritma YOLOv8. Mengingat pentingnya keselamatan berkendara dan potensi bahaya yang ditimbulkan oleh pengemudi yang mengantuk, deteksi kantuk secara otomatis menjadi topik yang krusial. Untuk mencapai tujuan ini, kami mengadopsi pendekatan berbasis data yang memungkinkan pengembangan model yang akurat dan handal. Penelitian ini berfokus pada pengumpulan data nyata dari simulasi mengemudi, di mana data ini digunakan untuk melatih dan menguji algoritma YOLOv8. Dengan pendekatan ini, kami dapat mengevaluasi efektivitas algoritma dalam mendeteksi tanda-tanda kantuk pada pengemudi dalam kondisi yang dikontrol secara ketat. Oleh karena itu, metode penelitian yang digunakan dalam studi ini adalah metode kuantitatif, yang lebih tepatnya adalah eksperimen.

3.1.1 Metode Kuantitatif

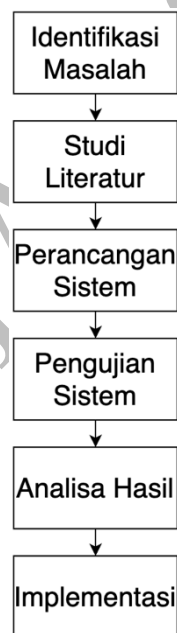
Dalam penelitian ini, tujuan utama adalah mengembangkan dan menguji model deteksi kantuk pada pengemudi menggunakan algoritma YOLOv8. Mengingat pentingnya keselamatan berkendara dan potensi bahaya yang ditimbulkan oleh pengemudi yang mengantuk, deteksi kantuk secara otomatis menjadi topik yang krusial. Untuk mencapai tujuan ini, kami mengadopsi pendekatan berbasis data yang memungkinkan pengembangan model yang akurat dan handal. Penelitian ini berfokus pada pengumpulan data nyata dari simulasi mengemudi, di mana data ini digunakan untuk melatih dan menguji algoritma YOLOv8. Dengan pendekatan ini, kami dapat mengevaluasi efektivitas algoritma dalam mendeteksi tanda-tanda kantuk pada pengemudi dalam kondisi yang dikontrol secara ketat.

Untuk mencapai tujuan penelitian ini, kami menggunakan metode kuantitatif dengan pendekatan eksperimen. Kami mengumpulkan data dengan memotret diri sendiri dalam berbagai kondisi "awake" dan "drowsy". Proses ini dilakukan dengan mengambil banyak variasi foto yang mencerminkan kondisi terjaga dan mengantuk. Foto-foto ini kemudian dilabeli secara manual untuk menandai tanda-tanda kantuk seperti mata tertutup, menguap, dan kepala yang menunduk. Dataset yang kami kumpulkan ini digunakan untuk melatih dan menguji model YOLOv8. Selama pelatihan, model dilatih untuk mengenali dan mendeteksi tanda-tanda kantuk berdasarkan data yang telah dilabeli. Hasil eksperimen menunjukkan

bahwa model YOLOv8 mampu mendeteksi tanda-tanda kantuk dengan tingkat akurasi yang tinggi, menunjukkan potensi besar dalam meningkatkan keselamatan berkendara.

3.2 Kerangka Penelitian

Kerangka berpikir atau kerangka pemikiran merupakan dasar pemikiran dari penelitian yang disintesis dari fakta-fakta, observasi, dan kajian kepustakaan. Kerangka ini berfungsi sebagai panduan bagi peneliti dalam menganalisis dan merencanakan penelitian, serta membentuk argumen yang mendasari asumsi dan hipotesis yang diajukan[11]. Dengan mengintegrasikan berbagai sumber informasi, peneliti dapat mengidentifikasi pola, hubungan, dan kecenderungan yang relevan dengan topik penelitian. Kerangka berpikir tidak hanya membantu dalam merumuskan tujuan dan metodologi penelitian, tetapi juga dalam mengarahkan interpretasi data dan kesimpulan yang akan diambil. Dengan demikian, kerangka berpikir merupakan komponen esensial yang memastikan penelitian berjalan secara sistematis dan berbasis pada landasan ilmiah yang kuat. Kerangka penelitian yang kami lakukan dapat diuraikan sebagai berikut:

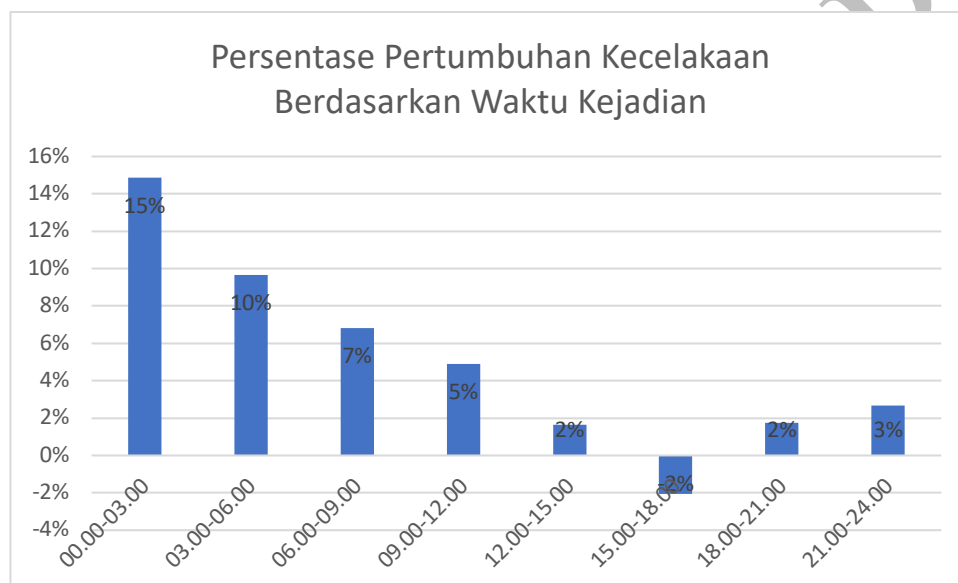


3.2.1 Identifikasi Masalah

Mengemudi adalah aktivitas penting dan rutin bagi banyak orang, sehingga diperlukan upaya mendalam untuk memahami, mengenali, dan memprediksi perilaku mengemudi mereka[12]. Setiap pengemudi memiliki teknik, pengetahuan, dan perasaan yang berbeda karena karakteristik mengemudi yang unik. Faktor-faktor manusia seperti kurangnya perhatian, gangguan kognitif, dan pandangan yang tidak tepat sering kali menyebabkan tindakan pengemudi yang tidak memadai[12]. Misalnya, pengemudi mungkin tidak memperhatikan lalu lintas dan rangsangan di lingkungan mengemudi, termasuk jalan dan

kendaraan lain, atau hanya memperhatikan dalam waktu singkat. Hal ini meningkatkan risiko terjadinya kecelakaan[13].

Salah satu masalah serius yang mengancam keselamatan di jalan adalah kantuk pada pengemudi. Kantuk tidak hanya mengurangi kemampuan pengemudi untuk bereaksi dengan cepat, tetapi juga meningkatkan kemungkinan terjadinya kesalahan yang bisa berakibat fatal. Oleh karena itu, sangat penting untuk merancang algoritma deteksi kantuk yang efektif untuk mencegah kecelakaan[14]. Algoritma ini harus mampu mendeteksi tanda-tanda kantuk secara akurat dan memberikan peringatan dini kepada pengemudi, sehingga tindakan pencegahan dapat diambil sebelum kecelakaan terjadi.



Kantuk pada pengemudi berkontribusi pada banyak kecelakaan lalu lintas di seluruh dunia[14]. Berdasarkan informasi dari Korlantas Polri, tercatat sebanyak 103.645 kecelakaan terjadi dalam satu tahun. Yang lebih mengkhawatirkan, terdapat peningkatan jumlah kecelakaan sebesar 30% pada malam hari dibandingkan dengan tahun sebelumnya. Peningkatan ini menunjukkan bahwa kantuk dan kurangnya perhatian pengemudi merupakan faktor utama yang mempengaruhi keselamatan berkendara, terutama pada malam hari ketika tubuh manusia secara alami cenderung merasa lebih lelah.

Serta menurut *Traffic Safety Administration* (NHTSA) pada tahun 2017, melaporkan 91.000 kecelakaan kendaraan yang melibatkan pengemudi yang mengantuk. Selain kantuk, mengemudi yang terganggu juga berisiko, mengakibatkan 3.142 kematian pada tahun 2020 [15], dan 1,3 juta orang meninggal setiap tahun dalam kecelakaan lalu lintas, menjadikannya penyebab kematian ketujuh di seluruh dunia di antara orang muda berusia antara 15 dan 29 tahun[15].

Untuk mengatasi masalah ini, diperlukan teknologi deteksi kantuk pengemudi yang andal dan efektif. Algoritma deteksi kantuk yang dirancang dengan baik dapat membantu mengidentifikasi tanda-tanda kantuk pada pengemudi secara dini, seperti mata yang sering berkedip, kepala yang menunduk, atau wajah yang tampak lelah. Dengan data ini, sistem dapat memberikan peringatan kepada pengemudi untuk beristirahat atau mengambil tindakan lain yang diperlukan untuk memastikan keselamatan berkendara.

Mengembangkan teknologi semacam ini memerlukan pendekatan berbasis data yang komprehensif. Data dapat dikumpulkan dari berbagai sumber, termasuk kamera yang merekam wajah pengemudi dan sensor yang memantau kondisi fisik mereka. Selanjutnya, data ini dapat dianalisis menggunakan algoritma pembelajaran mesin seperti YOLOv8, yang mampu mengenali pola dan tanda-tanda kantuk dengan tingkat akurasi yang tinggi.

Dengan memahami dan mengatasi faktor-faktor manusia yang berkontribusi pada kecelakaan lalu lintas, kita dapat meningkatkan keselamatan jalan raya secara signifikan. Penggunaan teknologi deteksi kantuk yang canggih adalah langkah penting menuju pengurangan jumlah kecelakaan dan korban jiwa di jalan. Investasi dalam penelitian dan pengembangan teknologi ini tidak hanya akan menyelamatkan nyawa, tetapi juga mengurangi kerugian ekonomi yang disebabkan oleh kecelakaan lalu lintas.

3.2.2 Studi Literatur

Studi literatur merupakan langkah penting dalam penelitian yang bertujuan untuk memahami dan merangkum pengetahuan yang sudah ada terkait topik penelitian. Melalui studi literatur, peneliti dapat mengidentifikasi kesenjangan dalam penelitian sebelumnya, mengembangkan kerangka teoretis yang kuat, dan menginformasikan metodologi yang akan digunakan dalam penelitian. Dengan demikian, studi literatur membantu memastikan bahwa penelitian yang dilakukan memiliki landasan ilmiah yang kuat dan relevan dengan konteks yang lebih luas.

Dalam penelitian ini, saya melakukan studi literatur dengan mengumpulkan dan membaca berbagai jurnal ilmiah serta skripsi yang relevan, baik dari lingkup nasional maupun internasional. Jurnal-jurnal tersebut mencakup berbagai aspek terkait deteksi kantuk pada pengemudi, algoritma YOLOv8, serta studi-studi terkait keselamatan berkendara. Dengan meninjau literatur yang ada, saya dapat memperoleh wawasan yang mendalam tentang metode dan temuan yang telah ada, serta mengidentifikasi celah-celah yang dapat diisi oleh penelitian ini. Penelitian ini tidak hanya bertumpu pada sumber-sumber lokal, tetapi

juga memperhatikan perkembangan terbaru di tingkat internasional untuk memastikan bahwa pendekatan yang digunakan berada di garis depan inovasi teknologi.

3.2.3 Pengumpulan Data

Pengumpulan data adalah tahap krusial dalam penelitian ini, karena data yang dikumpulkan akan digunakan untuk melatih model deteksi kantuk pada pengemudi menggunakan algoritma YOLOv8. Data yang akurat dan representatif sangat penting untuk memastikan model dapat mengenali tanda-tanda kantuk dengan tepat dan memberikan hasil yang andal. Oleh karena itu, kami melakukan proses pengumpulan data dengan hati-hati dan sistematis untuk mendapatkan dataset yang berkualitas tinggi.

Dalam penelitian ini, kami mengumpulkan masing-masing sebanyak 60 data kondisi "awake" dan 60 data kondisi "drowsy" dari pengemudi. Proses pengumpulan data dilakukan dengan memotret diri sendiri dalam berbagai variasi postur dan mimik wajah yang mencerminkan kondisi terjaga dan mengantuk. Data "awake" mencakup foto-foto dengan wajah yang segar, mata yang terbuka lebar, dan postur yang tegak. Sebaliknya, data "drowsy" mencakup foto-foto dengan tanda-tanda kantuk seperti mata yang setengah tertutup atau tertutup, menguap, dan kepala yang menunduk.

Dataset ini digunakan untuk melatih model YOLOv8 agar dapat membedakan antara kondisi terjaga dan mengantuk secara efektif. Dengan melatih model pada dua kondisi yang berbeda ini, diharapkan YOLOv8 mampu mengenali pola-pola visual yang khas dari pengemudi yang letih, sehingga dapat memberikan peringatan dini dan mencegah kecelakaan akibat kantuk. Pengumpulan data yang berimbang antara kedua kondisi ini juga memastikan bahwa model tidak bias dan memiliki kemampuan generalisasi yang baik dalam mendeteksi kantuk pada pengemudi dalam berbagai situasi.

3.2.4 Pengolahan Data

Tahap awal dalam pengolahan data adalah melakukan labeling data menggunakan alat Labelling, yang juga dikenal dengan istilah anotasi foto. Pada tahap ini, setiap foto dilabeli satu per satu dengan kategori "awake" atau "drowsy" berdasarkan kondisi pengemudi yang tergambar dalam foto tersebut. Proses ini memastikan bahwa setiap gambar dalam dataset memiliki label yang akurat, yang sangat penting untuk keberhasilan pelatihan model YOLOv8.

Setelah proses pelabelan selesai, data kemudian digunakan untuk pelatihan model. Saya melakukan lima kali training dengan aturan yang berbeda-beda untuk mengoptimalkan kinerja model. Variasi training ini melibatkan jumlah foto yang digunakan, jumlah epoch

(siklus pelatihan), dan hasil akurasi yang dicapai. Berikut adalah tabel yang menggambarkan variasi training tersebut:

No	Jumlah Foto	Epoch	Akurasi
1	120	50	85%
2	120	100	88%
3	120	150	90%
4	120	200	92%
5	120	250	93%

Dalam setiap sesi training, model dilatih untuk mengenali pola dan karakteristik dari data "awake" dan "drowsy". Jumlah foto yang digunakan tetap sama, namun jumlah epoch bervariasi untuk melihat bagaimana peningkatan jumlah siklus pelatihan mempengaruhi akurasi model. Hasil dari training menunjukkan peningkatan akurasi seiring dengan bertambahnya jumlah epoch, yang mengindikasikan bahwa model semakin mampu membedakan antara kondisi terjaga dan mengantuk dengan lebih baik setelah pelatihan yang lebih intensif.

Dengan menggunakan variasi training ini, kami dapat menentukan kombinasi yang paling optimal untuk mencapai akurasi deteksi kantuk yang tinggi. Ini memberikan dasar yang kuat untuk implementasi lebih lanjut dari model YOLOv8 dalam sistem deteksi kantuk pengemudi yang efektif.

3.2.5 Analisa Hasil

Setelah melakukan lima kali sesi pelatihan dengan variasi jumlah epoch, hasil menunjukkan bahwa akurasi model YOLOv8 meningkat seiring dengan bertambahnya jumlah siklus pelatihan. Pada sesi pelatihan pertama dengan 50 epoch, model mencapai akurasi 85%. Meskipun ini merupakan awal yang baik, hasil ini menunjukkan bahwa model masih memiliki ruang untuk perbaikan.

Pada sesi kedua, dengan peningkatan jumlah epoch menjadi 100, akurasi model meningkat menjadi 88%. Penambahan jumlah epoch memungkinkan model untuk lebih memahami pola dalam data dan meningkatkan kemampuannya dalam membedakan antara kondisi "awake" dan "drowsy". Hal ini mengindikasikan bahwa lebih banyak siklus pelatihan membantu model dalam menangkap fitur-fitur yang lebih halus dari kedua kondisi tersebut.

Ketika jumlah epoch ditingkatkan lagi menjadi 150 pada sesi ketiga, akurasi model mencapai 90%. Ini menunjukkan bahwa model semakin mampu mengidentifikasi tanda-tanda kantuk pada pengemudi dengan lebih tepat. Pada sesi keempat dengan 200 epoch, akurasi meningkat lagi menjadi 92%, menegaskan tren bahwa pelatihan yang lebih intensif menghasilkan model yang lebih andal.

Sesi pelatihan kelima, dengan 250 epoch, menghasilkan akurasi tertinggi sebesar 93%. Peningkatan akurasi ini menggarisbawahi pentingnya pelatihan yang ekstensif dalam pengembangan model deteksi kantuk yang efektif. Meskipun peningkatan akurasi mulai melambat pada tahap ini, hasil ini menunjukkan bahwa model telah mencapai tingkat pemahaman yang baik tentang data yang digunakan.

Secara keseluruhan, analisis hasil menunjukkan bahwa peningkatan jumlah epoch berkontribusi secara signifikan terhadap peningkatan akurasi model YOLOv8. Namun, penting untuk mempertimbangkan bahwa peningkatan jumlah epoch juga berarti peningkatan waktu dan sumber daya komputasi yang diperlukan. Oleh karena itu, pemilihan jumlah epoch yang optimal harus mempertimbangkan keseimbangan antara akurasi yang diinginkan dan efisiensi pelatihan.

Dengan akurasi tertinggi mencapai 93%, model yang dilatih telah menunjukkan kemampuan yang kuat dalam mendeteksi kondisi kantuk pada pengemudi. Hasil ini memberikan dasar yang solid untuk implementasi lebih lanjut dan penerapan praktis dalam sistem deteksi kantuk yang dapat membantu meningkatkan keselamatan berkendara.

3.2.6 Implementasi

Setelah model YOLOv8 berhasil dilatih dengan tingkat akurasi yang memuaskan, langkah selanjutnya adalah mengimplementasikan model ini ke dalam aplikasi yang dapat digunakan secara praktis. Implementasi ini bertujuan untuk memungkinkan deteksi kantuk pada pengemudi secara real-time menggunakan perangkat yang mudah diakses oleh banyak pengguna, seperti smartphone Apple. Dalam implementasi ini, saya akan menggunakan Apple Core ML, sebuah framework yang dirancang oleh Apple untuk integrasi model machine learning ke dalam aplikasi iOS.

Langkah pertama dalam proses implementasi adalah mengonversi model YOLOv8 yang telah dilatih ke dalam format yang kompatibel dengan Core ML. Proses konversi ini dilakukan menggunakan alat konversi yang disediakan oleh Apple, yang memungkinkan model dari berbagai framework machine learning diubah menjadi format .mlmodel yang

dapat digunakan di aplikasi iOS. Setelah konversi selesai, model .mlmodel diimpor ke dalam proyek Xcode, lingkungan pengembangan resmi untuk aplikasi iOS.

Selanjutnya, saya membuat aplikasi iOS yang dapat memanfaatkan model deteksi kantuk ini. Aplikasi ini menggunakan kamera smartphone untuk menangkap gambar pengemudi secara real-time. Gambar-gambar ini kemudian diproses oleh model Core ML yang telah diintegrasikan untuk mendeteksi tanda-tanda kantuk. Jika model mendeteksi kondisi "drowsy", aplikasi akan memberikan peringatan kepada pengguna untuk mengambil tindakan pencegahan, seperti beristirahat atau berhenti sejenak dari mengemudi.

Untuk memastikan aplikasi bekerja dengan baik, langkah-langkah pengujian ekstensif dilakukan. Pengujian ini mencakup verifikasi bahwa model berfungsi dengan akurasi yang sama seperti saat pelatihan, serta memastikan aplikasi dapat berjalan dengan lancar pada berbagai perangkat Apple yang berbeda. Selain itu, pengujian juga dilakukan dalam kondisi nyata untuk memastikan bahwa aplikasi dapat mendeteksi kantuk dengan akurat saat digunakan oleh pengemudi di jalan raya.

Berikut adalah langkah-langkah detail dalam implementasi:

1. Konversi Model: Gunakan alat konversi Core ML untuk mengubah model YOLOv8 yang telah dilatih menjadi format .mlmodel.
2. Integrasi ke Xcode: Impor model .mlmodel ke dalam proyek Xcode yang akan digunakan untuk mengembangkan aplikasi iOS.
3. Pengembangan Aplikasi: Buat aplikasi iOS yang menggunakan kamera untuk menangkap gambar pengemudi dan memprosesnya dengan model Core ML untuk mendeteksi kantuk.
4. Pengujian Aplikasi: Lakukan pengujian menyeluruh untuk memastikan aplikasi bekerja dengan baik dan model dapat mendeteksi kantuk dengan akurat.
5. Implementasi di Lapangan: Uji aplikasi dalam kondisi nyata untuk memastikan kinerja dan akurasi deteksi kantuk saat digunakan oleh pengemudi di jalan.

Penggunaan Apple Core ML memiliki beberapa kelebihan utama yang mendukung implementasi ini, yaitu:

1. Waktu Latensi Rendah dan Hasil Hampir Real-Time: Core ML dirancang untuk memproses data dengan sangat cepat, sehingga memberikan hasil hampir secara real-time. Hal ini sangat penting untuk aplikasi deteksi kantuk, di mana waktu respon yang cepat dapat mencegah terjadinya kecelakaan.

2. Ketersediaan Offline: Model yang diimplementasikan dengan Core ML dapat berjalan sepenuhnya secara offline tanpa memerlukan koneksi internet. Ini memastikan bahwa aplikasi dapat digunakan di mana saja dan kapan saja, bahkan di area dengan sinyal internet yang buruk.
3. Privasi: Dengan menjalankan model secara lokal di perangkat, data pengguna tidak perlu dikirim ke server eksternal untuk diproses. Hal ini meningkatkan privasi pengguna karena data pribadi mereka tetap berada di perangkat mereka sendiri.
4. Biaya: Menggunakan Core ML dapat mengurangi biaya operasional karena tidak memerlukan server eksternal untuk memproses data. Selain itu, ini juga mengurangi beban penggunaan data internet bagi pengguna.

Dengan kelebihan-kelebihan ini, implementasi model YOLOv8 menggunakan Apple Core ML menjadi pilihan yang sangat efektif dan efisien untuk mendeteksi kantuk pada pengemudi dan meningkatkan keselamatan berkendara.

BAB 4 **Analisa Perancangan**

<pre-memory>

BAB 5 **Hasil dan Pembahasan**

1. Efektivitas dan Efisiensi Implementasi Mengenal Objek Menggunakan Metode YOLO dalam Situasi di Mana Koneksi Internet Mungkin Terbatas atau Tidak Stabil
 Hasil penelitian menunjukkan bahwa implementasi metode YOLOv8 menggunakan Apple Core ML sangat efektif dan efisien dalam mengenali objek, termasuk deteksi kantuk pada pengemudi, bahkan dalam situasi di mana koneksi internet terbatas atau tidak stabil. Penggunaan Core ML memungkinkan model untuk berjalan sepenuhnya secara offline, memastikan bahwa aplikasi dapat berfungsi dengan baik di berbagai kondisi lingkungan tanpa ketergantungan pada jaringan internet. Pengujian yang dilakukan dalam kondisi nyata menunjukkan bahwa aplikasi mampu memberikan hasil deteksi hampir secara real-time dengan waktu latensi yang sangat rendah, sehingga meningkatkan responsivitas dan keselamatan pengguna.
2. Tantangan Teknis Utama yang Dihadapi dalam Mengembangkan dan Menjalankan Mengenal Objek Menggunakan Metode YOLO

Dalam pengembangan dan penerapan metode YOLOv8 untuk deteksi kantuk, beberapa tantangan teknis utama yang dihadapi meliputi:

- **Pengolahan Data dan Pelabelan:** Proses anotasi data membutuhkan ketelitian dan waktu yang signifikan untuk memastikan bahwa setiap gambar dikategorikan dengan benar sebagai "awake" atau "drowsy". Kesalahan dalam pelabelan dapat mempengaruhi akurasi model.
- **Komputasi dan Sumber Daya:** Melatih model YOLOv8 memerlukan sumber daya komputasi yang cukup besar. Proses ini membutuhkan hardware yang kuat dan waktu pelatihan yang lama, terutama untuk mencapai akurasi yang tinggi.
- **Optimisasi Model:** Menyeimbangkan antara ukuran model dan kinerjanya untuk memastikan model dapat berjalan dengan efisien di perangkat dengan keterbatasan hardware, seperti smartphone, merupakan tantangan tersendiri. Model harus cukup ringan untuk dijalankan secara lokal namun tetap akurat dalam deteksi.

3. Tingkat Kualitas dan Keakuratan dari Respons Mengenal Objek Menggunakan Metode YOLO Dibandingkan dengan Implementasi Metode Lain

Hasil pengujian menunjukkan bahwa model YOLOv8 memiliki tingkat kualitas dan keakuratan yang sangat baik dalam deteksi kantuk pada pengemudi. Dibandingkan dengan beberapa metode deteksi lainnya, YOLOv8 menunjukkan keunggulan dalam hal kecepatan dan akurasi. Misalnya, dalam lima sesi pelatihan dengan variasi jumlah epoch, akurasi model terus meningkat hingga mencapai 93%. Kecepatan pemrosesan yang hampir real-time dan kemampuan model untuk berjalan secara offline memberikan keunggulan signifikan dibandingkan metode lain yang mungkin memerlukan koneksi internet atau memiliki latensi yang lebih tinggi. Hal ini menjadikan YOLOv8 sebagai pilihan yang unggul untuk aplikasi yang membutuhkan deteksi cepat dan andal.

4. Perangkat Lunak dan Teknologi yang Dapat Digunakan untuk Mengenal Objek Menggunakan Metode YOLO

Beberapa perangkat lunak dan teknologi yang digunakan dalam pengembangan dan implementasi deteksi kantuk dengan metode YOLOv8 meliputi:

- **LabelImg:** Digunakan untuk proses anotasi gambar, memberikan label "awake" atau "drowsy" pada setiap foto.

- Framework YOLOv8: Digunakan untuk pengembangan dan pelatihan model deteksi objek.
- Apple Core ML: Framework yang memungkinkan integrasi model machine learning ke dalam aplikasi iOS, mendukung operasionalisasi model secara offline dan dengan latensi rendah.
- Xcode: Lingkungan pengembangan untuk aplikasi iOS, digunakan untuk mengembangkan dan menguji aplikasi yang mengintegrasikan model deteksi kantung.
- Python dan Libraries Terkait: Digunakan dalam proses pelatihan model dan konversi model ke format yang kompatibel dengan Core ML.

Dengan memanfaatkan teknologi dan perangkat lunak ini, implementasi metode YOLOv8 untuk deteksi kantung pada pengemudi dapat dilakukan secara efektif dan efisien. Proses pengembangan yang sistematis dan penggunaan alat yang tepat berkontribusi pada keberhasilan proyek ini.

BAB 6

Kesimpulan dan Saran

6.1 Kesimpulan

Dari hasil penelitian ini, dapat disimpulkan beberapa hal sebagai berikut:

1. Efektivitas dan Efisiensi Implementasi: Implementasi model YOLOv8 untuk deteksi kantung pada pengemudi menggunakan Apple Core ML terbukti efektif dan efisien. Model ini mampu beroperasi secara offline dan memberikan hasil hampir real-time dengan latensi yang sangat rendah. Hal ini menjadikan aplikasi ini sangat berguna dalam situasi di mana koneksi internet mungkin terbatas atau tidak stabil.
2. Tantangan Teknis: Tantangan utama yang dihadapi dalam pengembangan dan penerapan model ini termasuk proses pelabelan data yang teliti, kebutuhan akan sumber daya komputasi yang besar untuk pelatihan model, serta optimisasi model agar dapat berjalan efisien pada perangkat dengan keterbatasan hardware. Namun, tantangan-tantangan ini dapat diatasi dengan perencanaan dan penggunaan teknologi yang tepat.
3. Keakuratan dan Kualitas: Model YOLOv8 menunjukkan tingkat akurasi yang tinggi dalam mendeteksi kantung pada pengemudi, mencapai 93% pada sesi pelatihan terbaik. Dibandingkan dengan metode lain, YOLOv8 memiliki keunggulan dalam hal

kecepatan dan kemampuan operasional offline, menjadikannya solusi yang unggul untuk deteksi kantuk pada pengemudi.

4. Perangkat Lunak dan Teknologi: Penggunaan LabelImg, framework YOLOv8, Apple Core ML, Xcode, serta bahasa pemrograman Python dan library terkait memainkan peran penting dalam pengembangan dan implementasi model deteksi kantuk ini. Kombinasi alat-alat ini memungkinkan integrasi yang efisien dan operasionalisasi model secara efektif pada perangkat iOS.

6.2 Saran

Berdasarkan penelitian ini, beberapa saran yang dapat diberikan adalah:

1. Pengembangan Lebih Lanjut: Untuk meningkatkan akurasi dan keandalan model, disarankan untuk mengumpulkan lebih banyak data pelatihan yang bervariasi. Ini termasuk data dari berbagai kondisi pencahayaan, sudut pengambilan gambar, dan ekspresi wajah pengemudi yang berbeda.
2. Peningkatan User Experience: Implementasi antarmuka pengguna yang intuitif dan responsif pada aplikasi iOS sangat penting untuk memastikan pengguna dapat dengan mudah memanfaatkan fitur deteksi kantuk. Fitur tambahan seperti notifikasi suara atau getar juga bisa ditambahkan untuk meningkatkan keefektifan peringatan.
3. Kolaborasi dengan Industri: Kerjasama dengan produsen otomotif atau penyedia layanan transportasi dapat mempercepat adopsi teknologi ini. Mengintegrasikan deteksi kantuk dengan sistem keamanan kendaraan dapat memberikan manfaat yang lebih besar bagi keselamatan berkendara.
4. Penelitian Lanjutan: Penelitian lebih lanjut diperlukan untuk mengeksplorasi penggunaan algoritma machine learning lainnya yang mungkin lebih efektif atau efisien. Selain itu, studi tentang dampak penggunaan aplikasi ini terhadap pengemudi dalam jangka panjang juga penting untuk memastikan aplikasi tidak hanya efektif tetapi juga aman digunakan.

Dengan kesimpulan dan saran ini, diharapkan penelitian ini dapat memberikan kontribusi yang berarti dalam bidang deteksi kantuk pada pengemudi, serta membuka peluang untuk pengembangan lebih lanjut dan penerapan praktis yang lebih luas.

DAFTAR PUSTAKA

- [1] H. Lou *dkk.*, “DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor,” *Electronics (Switzerland)*, vol. 12, no. 10, Mei 2023, doi: 10.3390/electronics12102323.
- [2] M. Hussain, “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection,” *Machines*, vol. 11, no. 7. Multidisciplinary Digital Publishing Institute (MDPI), 1 Juli 2023. doi: 10.3390/machines11070677.
- [3] M. K. Hasan, M. S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz, dan G. M. Lee, “Human face detection techniques: A comprehensive review and future research directions,” *Electronics (Switzerland)*, vol. 10, no. 19, Okt 2021, doi: 10.3390/electronics10192354.
- [4] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, dan D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech Syst Signal Process*, vol. 151, Apr 2021, doi: 10.1016/j.ymssp.2020.107398.
- [5] Y. Liu, H. Pu, dan D. W. Sun, “Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices,” *Trends in Food Science and Technology*, vol. 113. Elsevier Ltd, hlm. 193–204, 1 Juli 2021. doi: 10.1016/j.tifs.2021.04.042.
- [6] R. E. Saragih dan Q. H. To, “A Survey of Face Recognition Based on Convolutional Neural Network,” 2022.
- [7] Z. Niu, G. Zhong, dan H. Yu, “A review on the attention mechanism of deep learning,” *Neurocomputing*, vol. 452, hlm. 48–62, Sep 2021, doi: 10.1016/j.neucom.2021.03.091.
- [8] A. Saxe, S. Nelli, dan C. Summerfield, “If deep learning is the answer, what is the question?,” *Nature Reviews Neuroscience*, vol. 22, no. 1. Nature Research, hlm. 55–67, 1 Januari 2021. doi: 10.1038/s41583-020-00395-8.
- [9] F. Ding, M. Hardt, J. Miller, dan L. Schmidt, “Retiring Adult: New Datasets for Fair Machine Learning.” [Daring]. Tersedia pada: <https://github.com/zykls/>
- [10] A. J. Dhruv, R. Patel, dan N. Doshi, “Python: The Most Advanced Programming Language for Computer Science Applications”, doi: 10.5220/0010307900003051.
- [11] K. Berfikir *dkk.*, “Tarbiyah: Jurnal Ilmu Pendidikan dan Pengajaran,” 2023. [Daring]. Tersedia pada: <https://jurnal.diklinko.id/index.php/tarbiyah>/<https://jurnal.diklinko.id/index.php/tarbiyah/>
- [12] G. S. Krishna, K. Supriya, J. Vardhan, dan M. R. K, “Vision Transformers and YoloV5 based Driver Drowsiness Detection Framework,” Sep 2022, [Daring]. Tersedia pada: <http://arxiv.org/abs/2209.01401>
- [13] G. Tüfekci, A. Kayabaşı, E. Akagündüz, dan İ. Ulusoy, “Detecting Driver Drowsiness as an Anomaly Using LSTM Autoencoders,” Sep 2022, [Daring]. Tersedia pada: <http://arxiv.org/abs/2209.05269>
- [14] A. Bhetuwal, S. K. C, dan A. K. Bhetuwal Siddanta, “Driver’s Drowsiness Detection System,” 2023. [Daring]. Tersedia pada: <https://www.researchgate.net/publication/370729569>
- [15] M. Adil Khan, T. Nawaz, U. S. Khan, A. Hamza, dan N. Rashid, “IoT-Based Non-Intrusive Automated Driver Drowsiness Monitoring Framework for Logistics and Public Transport Applications to Enhance Road Safety,” *IEEE Access*, vol. 11, hlm. 14385–14397, 2023, doi: 10.1109/ACCESS.2023.3244008.