**DETEKSI PENGEMUDI KANTUK MENGGUNAKAN ALGORITMA YOLOv8**

**TUGAS AKHIR**

**Text

Description automatically generated**

**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**

**Text

Description automatically generated**

**PROGRAM STUDI TEKNIK KOMPUTER**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNOLOGI BATAM**

**2023**

# DAFTAR ISI

[DAFTAR ISI iii](#_Toc166783013)

[BAB 1 PENDAHULUAN 1](#_Toc166783014)

[1.1 Latar Belakang 1](#_Toc166783015)

[1.2 Perumusan Masalah 1](#_Toc166783016)

[1.3 Batasan Masalah 2](#_Toc166783017)

[1.4 Metode Penelitian 2](#_Toc166783018)

[1.5 Tujuan dan Manfaat Penelitian 2](#_Toc166783019)

[1.5.1 Tujuan 2](#_Toc166783020)

[1.5.2 Manfaat 3](#_Toc166783021)

[1.6 Sistematika Penulisan 3](#_Toc166783022)

[BAB 2 TINJAUAN PUSTAKA 4](#_Toc166783023)

# PENDAHULUAN

## 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.

## 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?

## 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

## 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:

## Tujuan dan Manfaat Penelitian

### Tujuan

Tujuan dari penelitian ini:

1. Mengenali bagaimana cara kerja metode *YOLO* yang popular 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.

### 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*.

## 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 ya ng 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.

# TINJAUAN PUSTAKA

## YOLOV8



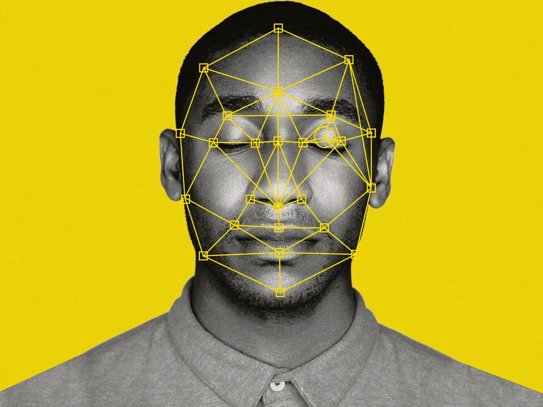
YOLO (*You Only Look Once*) adalah sebuah algoritma dalam bidang pengenalan objek pada citra yang sangat populer di bidang *computer vision* dan *machine learning*. Seri algoritma YOLO adalah salah satu algoritma dengan pertumbuhan tercepat dan terbaik hingga saat ini, terutama algoritma YOLOv8 yang 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(Lou dkk., 2023).

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*(Hussain, 2023). Kehadiran YOLO memberikan dampak yang signifikan dalam meningkatkan potensi pada *computer vision*(Hussain, 2023).

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(Hussain, 2023).
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(Hussain, 2023).
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(Hussain, 2023).
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(Hussain, 2023).
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(Hussain, 2023).
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(Hussain, 2023).
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(Hussain, 2023).
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(Hussain, 2023).

## *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(Lou dkk., 2023). *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(Hasan dkk., 2021).

*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(Hasan dkk., 2021) yang dapat membantu dalam melakukan deteksi kantuk pada pengemudi.

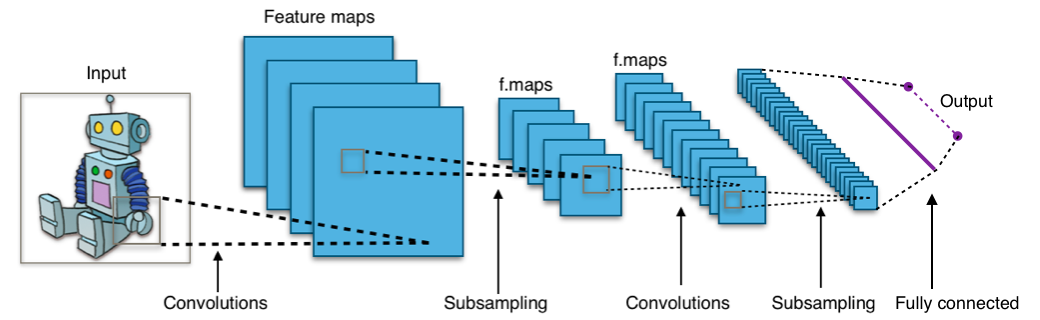
## *APPLE CORE ML*



*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​.

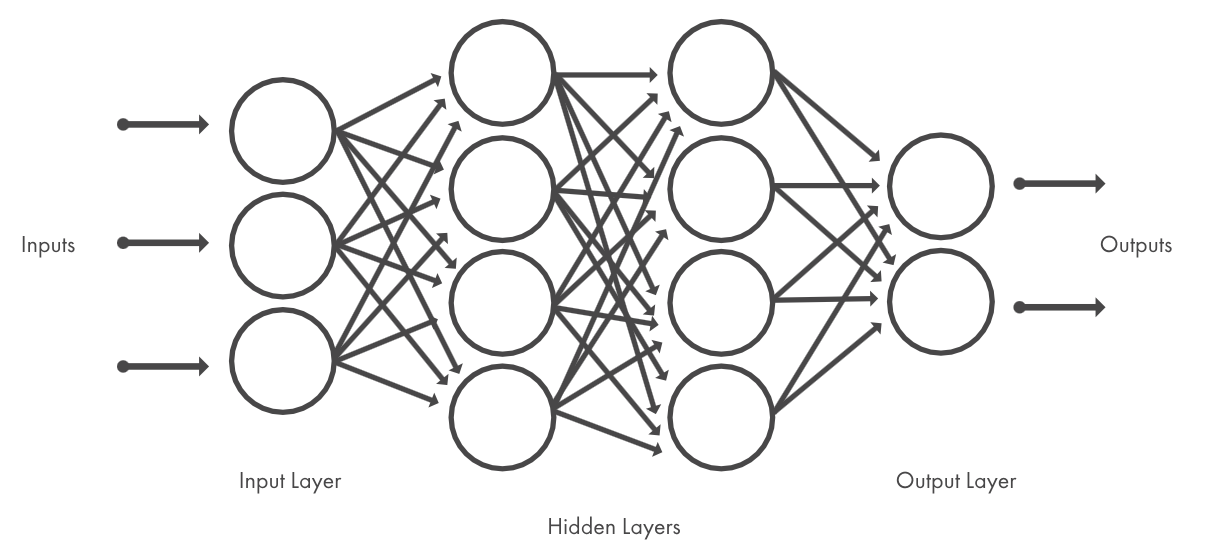
*Convolutional Neural Networks*



Selama sepuluh tahun terakhir, *Convolutional Neural Networks* (CNN) telah menjadi standar utama untuk berbagai operasi dalam bidang *computer vision* dan *machine learning*(Kiranyaz dkk., 2021). *Convolutional Neural Networks* (CNN) merupakan kelas dari jaringan saraf *deep feed-forward*, yang dibangun dengan meniru pola koneksi *neuron* di korteks visual manusia(Liu dkk., 2021).

*Convolutional Neural Networks* dapat secara otomatis mempelajari fitur-fitur yang mewakili variasi visual kompleks dari data pelatihan yang banyak.(Saragih & To, 2022). 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*(Saragih & To, 2022).

## DEEP LEARNING

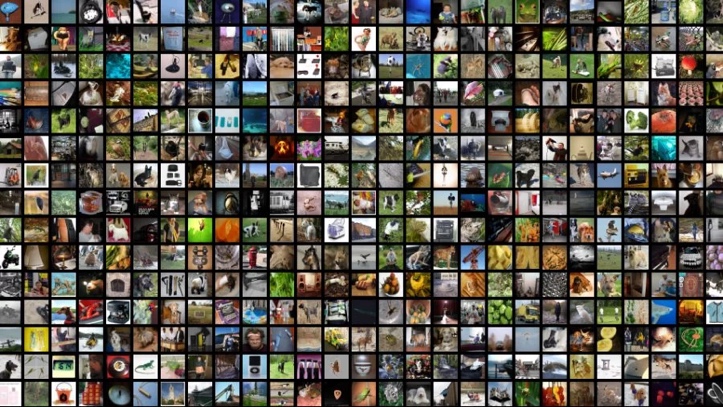


*Deep Learning* (DL) adalah pencapaian terbaru dalam era *machine learning*, di mana awalnya mampu menunjukkan kemampuan mendekati manusia, dan kini kemampuan yang melampaui manusia dalam banyak aplikasi termasuk terjemahan suara-ke-teks, deteksi dan pengenalan objek, deteksi anomali, pengenalan emosi dari rekaman audio atau video, dan sebagainya(Kiranyaz dkk., 2021).

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(Niu dkk., 2021). *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(Saxe dkk., 2021).

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(Saxe dkk., 2021).

## *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(Ding dkk., t.t.), *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 akademia dan industri(Ding dkk., t.t.).

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.

1. 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*.

1. 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.

1. *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.

1. 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.

1. 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.

1. Reproduksibilitas 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.

## *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(Dhruv dkk., t.t.).

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*)(Dhruv dkk., t.t.).

## Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Judul | Penulis | Metode | Hasil |
| 1 | *IoT-Based Non-Intrusive Automated Driver Drowsiness Monitoring Framework for Logistics and Public Transport Applications to Enhance Road Safety* | M. ADIL KHAN, TAHIR NAWAZ , UMAR S. KHAN , (Member, IEEE), AMIR HAMZA, AND NASIR RASHID | Kuantitatif | Jurnal ini mengusulkan kerangka kerja IoT otomatis *end-to-end* untuk memantau perilaku pengemudi, dirancang khusus untuk aplikasi logistik dan transportasi publik. Sistem ini terdiri dari Raspberry Pi dan Nvidia Jetson Nano untuk *edge computing* yang terhubung dengan *webcam* untuk menangkap gambar secara *real-time, cloud computing*, dan aplikasi *mobile*. Deteksi kantuk dilakukan dengan menganalisis aktivitas mata dan mulut, seperti tidur dan menguap, menggunakan teknik pemrosesan gambar berbasis *Deep Neural Network* (DNN) dengan *multilayer perceptron classifiers*. Sistem ini memantau empat parameter perilaku pengemudi: aktif, menguap, mata tertutup, dan terdistraksi, dengan akurasi deteksi sebesar 96%. Data yang dikumpulkan dikirim ke *database real-time* di *cloud*, 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 | *Drowsiness Detection Based On Driver Temporal Behavior Using a New Developed Dataset* | F. Faraji, F. Lotfi, J. Khorramdel, A. Najafi, A. Ghaffari K. N. Toosi University of Technology, Tehran, Iran. | Kuantitatif | Studi ini menggunakan pendekatan berbasis gambar dengan akurasi yang memadai dan proses yang cepat. Kami menerapkan YOLOv3 (*You Look Only Once-version*3) 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%. |
| 4 | Aplikasi *Warning Alert* Pendeteksi Kelelahan Ekspresi Wajah Pada Pengemudi Secara *Real-Time* Menggunakan Metode *You Only Look Once* Berbasis *Website* | Hafidh Ahmad Fauzan , Ari Kurniawan | Kuantitatif | penelitian yang menggunakan YOLOv4 untuk mendeteksi kelelahan pada pengemudi berdasarkan ekspresi wajah secara *real-time*. Algoritma pendeteksian diterapkan melalui platform *website* yang terhubung dengan Telegram, memungkinkan pengiriman pesan secara langsung mengenai status dan kondisi pengemudi dengan detail. Penulis memanfaatkan *dataset* yang tersedia di Kaggle, dipublikasikan oleh RAKIBUL.ECE.RUET, yang terdiri dari 4560 *file* per kelas, yaitu *active* dan *fatigue*. Selain itu, penulis juga menyiapkan *dataset* tambahan yang telah di-*augmentasi* menggunakan *photometric distortion* dan *geometric distortion* untuk meningkatkan variasi data. Melalui pendekatan ini, penulis berhasil mengembangkan metode baru yang mencapai akurasi sebesar 97,03%, menunjukkan efektivitas sistem dalam mendeteksi kelelahan pengemudi secara akurat dan cepat. |
| 5 | Implementasi Algoritma *Deep Learning* Untuk Sistem Deteksi Kantuk Pada Pengemudi MenggunakanYOLO | Mamta Anisa Bella | Kuantitatif | Dalam penelitian ini, pengemudi yang mengantuk diidentifikasi dengan label "*drowsy*", sementara pengemudi yang terjaga diberi label "*awake*". Metode yang digunakan memanfaatkan kombinasi YOLOv4 dan CNN, diimplementasikan melalui sebuah *website* yang terhubung langsung dengan layanan Telegram untuk mengirimkan status pengemudi secara *real-time*. *Dataset* yang digunakan diambil dari Kaggle, yang telah dipublikasikan oleh RAKIBUL.ECE.RUET, dengan masing-masing kelas memiliki 4560 *file*, serta *dataset* tambahan yang telah melalui proses *augmentasi*. Konfigurasi model termasuk penggunaan *batch* *size* 64, ukuran jaringan 416x416, subdivisi 16, *max* *batch* 4000, dan filter 21. Pembagian *dataset* 80%:20% dan *learning rate* 0.00261 memberikan nilai *Intersection* *over Union* (IoU) terbesar. Dengan metode ini, sistem berhasil mencapai akurasi sebesar 97,03%, menunjukkan efektivitasnya dalam mendeteksi kelelahan pengemudi dengan akurasi dan kecepatan yang tinggi. |
| 6 | *Driver's Drowsiness Detection System* | Aayush Bhetuwal, Siddanta K C | Kuantitatif | Penelitian ini mempersembahkan sistem deteksi kantuk pengemudi yang inovatif dengan memanfaatkan kemampuan canggih YOLOv5 dan CNN. Menggunakan *dataset* 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 *Intersection over Union* (IoU) sebesar 95,51%, meskipun hanya dilatih selama 10 *epoch*. Kemampuan deteksi *real-time* 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. |