BAB 1
PENDAHULUAN

1.1 Latar Belakang

Kantuk merupakan kondisi biologis yang tampak sederhana, namun berimplikasi besar dalam aktivitas yang membutuhkan tingkat konsentrasi tinggi seperti mengemudi atau bekerja di lingkungan industri. Ketika seseorang mengemudi dalam kondisi mengantuk, fokus dan kewaspadaan akan terganggu, yang dapat berujung pada kecelakaan fatal. Berdasarkan pengamatan, sebagian besar kecelakaan di jalan raya disebabkan karena perilaku lalai pengemudi dalam keadaan mengantuk saat mengemudikan kendaraan. Berdasarkan laporan tahun 2021 kecelakaan lalu lintas di Indonesia khususnya di jalan ruas jalan toll yang disampaikan ditlantas polda metro jaya meningkat 6.8% dari tahun sebelumnya. Analisis laporan ini menunjukkan bahwa 78% dari total kecelakaan di jalan raya disebabkan oleh kurangnya perhatian pengemudi karena dalam kondisi mengantuk. Rasa kantuk dapat membuat konsentrasi pengemudi menjadi menurun, terutama ketika jalanan sedang padat karena kemacetan yang panjang. Banyak pengemudi yang lebih memilih mengabaikan rasa kantuknya dan meneruskan perjalanannya hingga sampai ketujuan, padahal hal ini dapat mengancam keselamatan pengemudi. Beberapa tahun terakhir ini, konsentrasi yang kurang saat sedang mengemudi menjadi salah satu penyebab terjadinya banyak kecelakaan (Cvahte Ojsteršek & Topolšek, 2019).

Mengantuk dapat didefinisikan sebagai hilangnya efisiensi pemrosesan kortikal secara progresif (Slater, 2008). Ketidakefisienan ini menyebabkan hilangnya fungsi kognitif secara bertahap. Menurut penelitian sebelumnya, 56% perawat shift malam melaporkan kurang tidur(Johnson dkk., 2014), menghasilkan kemungkinan kesalahan perawatan pasien yang jauh lebih tinggi. Studi lain yang menganalisis insiden cedera industri melaporkan risiko cedera 1,23 kali lebih tinggi pada shift malam dibandingkan pada shift pagi dalam sistem tiga shift (Smith dkk., 1994).

Electroencephalography (EEG) adalah metode yang digunakan untuk merekam aktivitas otak dengan mengukur fluktuasi tegangan yang dihasilkan dari arus ionik di dalam neuron otak(Sadaghiani dkk., 2022). Karena pola EEG yang muncul sesuai tahapan tidur(Rodenbeck dkk., 2006), EEG telah diadopsi dalam sistem deteksi kantuk dalam beberapa penelitian.

Pada penelitian sebelumnya oleh Yeo, Li, Shen, dan Wilder-Smith, Sinyal EEG direkam selama satu jam dari 20 subjek selama simulasi mengemudi (Yeo dkk., 2009). Data yang direkam secara manual diberi label sebagai "waspada" atau "mengantuk" dengan aturan khusus berdasarkan frekuensi kedipan mata dan aktivitas EEG, yang disegmentasi menjadi periode 10 detik. Untuk setiap zaman, delta, kekuatan pita frekuensi (BPs) theta, alpha, beta, dan gamma dihitung untuk ruang fitur dan diklasifikasikan dengan akurasi 99,30% dengan SVM.

Menghitung model autoregresif multivariat (MVAR) untuk mengekstrak fitur EEG(C. Zhao dkk., 2011). Sinyal EEG direkam selama 150 menit dari 13 subjek selama simulasi mengemudi. Koefisien model MVAR orde ketiga digunakan sebagai fitur EEG. Fitur-fitur ini diuji menggunakan berbagai strategi klasifikasi. Analisis komponen utama linier dan analisis komponen utama kernel (KPCA) diadopsi sebagai algoritma pengurangan fitur, dan jaringan fungsi basis radial (RBF) dan SVM dipilih untuk klasifikasi. Kombinasi KPCA dan SVM memiliki performa terbaik (81,64 %) dengan 25 fitur.

Analisis yang terlibat dalam menentukan tingkat kantuk berdasarkan karakteristik fisiologis bersifat intrusif. Karena sifatnya yang mengganggu ini, sejumlah peralatan yang memiliki banyak sensor dipasang di berbagai bagian tubuh pengemudi yang mampu mempelajari sinyal otak, impuls saraf, dan sebagainya. Dengan demikian, peralatan ini menimbulkan beban tambahan bagi pengemudi yang menghambat kelancaran berkendara. Oleh karena itu, tidak boleh ada keterikatan fisik antara sistem identifikasi dan pengemudi. Jadi, setelah memberikan hasil yang fenomenal, teknik ini tidak layak secara komersial. Untuk mengatasi masalah yang ditunjukkan oleh teknik deteksi kantuk berbasis fisiologis dan karakteristik kendaraan seperti yang dibahas dalam paragraf di atas, muncullah

teknik visi komputer. Di era sekarang, teknik ini menjadi lebih populer karena biaya pelaksanaannya yang rendah dan mudah dikonfigurasi dengan kendaraan serta sifatnya yang tidak mengganggu. Dari survei literatur, peneliti menemukan bahwa teknik komputer vision sebagian besar menggunakan ekspresi wajah dalam menentukan keadaan mengantuk karena mudah untuk mengidentifikasi pengemudi mengantuk atau waspada melalui ekspresi wajah(Jayasenan & Smitha, 2014; Sindhu, 2024). Bergasa dkk mengungkapkan bahwa frekuensi, amplitudo, durasi yang berkaitan dengan pembukaan dan penutupan mulut serta mata memainkan fungsi penting dalam mengidentifikasi keadaan kantuk pengemudi. Kerangka kerja berdasarkan teknik ini, terutama memeriksa area sekitar dan kondisi iris pada slot waktu tertentu untuk menghitung variabel-variabel ini yaitu frekuensi, amplitudo, durasi (Bergasa dkk., 2006). Teknik visi komputer untuk deteksi kelelahan melalui analisis ekspresi wajah dan kondisi mata memiliki kekurangan, Kerentanan terhadap kondisi pencahayaan, yang mempengaruhi akurasi model pelacakan wajah dan mata. Penghalangan dari kacamata atau masker wajah, yang membuat sulit untuk mendeteksi isyarat mata dengan akurat dan Variabilitas dalam pose kepala, yang sering mengurangi keandalan model ketika pengemudi tidak langsung menghadap kamera (Deng & Wu, 2019).

Transformer telah menjadi arsitektur dominan dalam berbagai tugas pemrosesan bahasa alami. Namun, adopsinya pada domain visi komputer menghadapi tantangan besar khususnya dalam menangani resolusi tinggi dan kebutuhan prediksi spasial yang padat. Menanggapi tantangan ini, (Z. Liu dkk., 2021) memperkenalkan Swin Transformer, sebuah model Vision Transformer yang dirancang secara hierarkis dan efisien secara komputasional untuk menangani skala objek yang bervariasi dalam citra. Vision Transformer (ViT) merupakan model berbasis self-attention yang unggul dalam memahami dependensi global antar piksel dalam citra. ViViT, sebagai perluasan dari ViT untuk video, membagi video menjadi patch spasial-temporal dan menerapkan mekanisme transformer untuk mempelajari representasi fitur jangka panjang. Dibandingkan model CNN konvensional atau 3D-CNN, ViViT tidak hanya mempertimbangkan hubungan

spasial antar piksel, tetapi juga menangkap perubahan dinamis dalam waktu, menjadikannya sangat cocok untuk mengenali pola-pola seperti kantuk yang berkembang secara bertahap dalam durasi waktu tertentu. Arsitektur ini juga bersifat modular, memungkinkan integrasi dengan preprocessing ringan dan augmentasi data dalam pipeline deteksi real-time (Arnab dkk., 2021).

Pendekatan visual non-intrusif berkembang pesat karena hanya memerlukan kamera untuk mengamati ekspresi wajah dan posisi kepala pengemudi. Pendekatan ini lebih praktis dan berpotensi diterapkan secara luas, tetapi metode visual konvensional masih menghadapi beberapa kendala, seperti sensitivitas terhadap pencahayaan, oklusi oleh kacamata, perubahan sudut pandang wajah, dan keterbatasan model dalam menangkap dinamika perubahan visual antar-frame video. Dalam konteks deteksi kantuk, kendala ini menjadi penting karena kondisi mata dan kemiringan kepala bukan hanya informasi spasial pada satu frame, melainkan juga pola temporal yang berkembang dari waktu ke waktu. Beberapa studi terbaru telah mengeksplorasi penggunaan Vision Transformer untuk pengenalan ekspresi wajah dan prediksi kondisi mental. Penelitian oleh (Adil Khan dkk., 2023) berhasil menggunakan ViViT untuk mengenali ekspresi wajah mikro (micro-expression recognition) dengan akurasi tinggi, menunjukkan bahwa ViViT mampu mendeteksi sinyal visual halus dalam urutan video. Penelitian lain oleh (C. Zhang dkk., 2023) menerapkan transformer untuk estimasi kemiringan kepala (*head pose estimation*), yang menjadi salah satu fitur utama dalam deteksi kantuk berbasis perilaku.

Perkembangan deep learning, khususnya transformer-based vision models, membuka peluang baru untuk mengatasi keterbatasan tersebut. Vision Transformer (ViT) menunjukkan kemampuan yang baik dalam menangkap hubungan global pada citra melalui mekanisme self-attention. Pengembangannya ke domain video menghasilkan arsitektur Video Vision Transformer (ViViT), yang dirancang untuk memodelkan informasi spasial dan temporal secara simultan. Karakteristik ini menjadikan ViViT relevan untuk deteksi kantuk, karena tanda-tanda kantuk muncul

sebagai kombinasi perubahan visual wajah pada tiap frame dan dinamika gerakan kepala serta mata antar-frame.

Penelitian tentang deteksi kantuk pengemudi berbasis video masih banyak didominasi oleh pendekatan CNN, *landmark-based methods* atau kombinasi CNN-LSTM. Penelitian yang secara spesifik memanfaatkan ViViT untuk mendeteksi kantuk pengemudi berdasarkan kombinasi kondisi mata, mulut menguap dan kemiringan kepala pada dataset dengan kondisi visual yang beragam masih terbatas. Selain itu, belum banyak penelitian yang menempatkan tiga indikator visual tersebut sebagai fokus utama dalam satu kerangka pemodelan spasial-temporal berbasis transformer.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan pengembangan model deteksi kantuk pengemudi berbasis Video Vision Transformer (ViViT) dengan memanfaatkan indikator visual utama, yaitu kondisi mata, mulut menguap dan kemiringan kepala. Model ini diharapkan mampu memberikan deteksi yang lebih adaptif terhadap variasi kondisi visual, memiliki performa yang baik dalam skenario nyata, serta berpotensi diimplementasikan sebagai sistem peringatan dini untuk meningkatkan keselamatan berkendara.

1.2 Identifikasi Masalah

Tingginya angka kecelakaan lalu lintas yang melibatkan pengemudi dalam kondisi mengantuk menunjukkan urgensi pengembangan sistem deteksi dini yang mampu mengantisipasi risiko tersebut. Data dari Ditlantas Polda Metro Jaya pada tahun 2021 menunjukkan bahwa angka kecelakaan di jalan tol meningkat sebesar 6,8% dibandingkan tahun sebelumnya, dan sebanyak 78% dari total kecelakaan tersebut disebabkan oleh kurangnya perhatian pengemudi yang dipicu oleh rasa kantuk (Cvahte Ojsteršek & Topolšek, 2019). Kondisi ini mempertegas perlunya pendekatan teknologi yang dapat mengenali gejala kantuk secara tepat waktu dan akurat. Berbagai metode telah dikembangkan, salah satunya adalah pendekatan berbasis sinyal fisiologis seperti elektroensefalografi (EEG), yang terbukti memiliki tingkat akurasi tinggi dalam mendeteksi perubahan aktivitas otak terkait

kantuk (Yeo dkk., 2009; C. Zhao dkk., 2011; Sadaghiani dkk., 2022). Namun, metode ini bersifat intrusif karena membutuhkan perangkat sensor yang harus dipasang langsung pada tubuh pengguna, sehingga menimbulkan ketidaknyamanan dan membatasi penerapannya dalam skenario kendaraan nyata.

Penelitian lain yang bersifat non-intrusif adalah pendekatan berbasis karakteristik kendaraan, seperti analisis kecepatan, akselerasi, atau pola manuver setir. Meskipun tidak mengganggu kenyamanan pengemudi, pendekatan ini tidak secara langsung mencerminkan kondisi kognitif atau fisiologis pengemudi, sehingga akurasinya dalam mendeteksi kantuk masih terbatas (Slater, 2008). Di sisi lain, pendekatan berbasis perilaku visual, yang memanfaatkan isyarat dari ekspresi wajah dan kondisi mata, dinilai lebih praktis dan dapat diterapkan secara langsung melalui kamera pemantau. Meskipun demikian, teknik ini masih menghadapi sejumlah tantangan teknis seperti sensitivitas terhadap kondisi pencahayaan, gangguan dari aksesoris wajah seperti kacamata atau masker, serta variabilitas arah pandang kepala pengemudi(Bergasa dkk., 2006; Deng & Wu, 2019).

Metode Video Vision Transformer (ViViT) mampu memberikan solusi yang dapat menangkap pola spasial dan temporal dari data video secara simultan. ViViT memanfaatkan arsitektur Transformer yang terbukti unggul dalam memahami hubungan global antar piksel pada citra, serta mampu mengenali dinamika ekspresi wajah dan gerakan kepala yang terjadi dalam durasi waktu tertentu (Arnab dkk., 2021). Penelitian terkini oleh Adil Khan dkk. (2023) menunjukkan efektivitas ViViT dalam mengenali mikro-ekspresi wajah, sedangkan studi oleh Zhang dkk. (2023) membuktikan keberhasilannya dalam estimasi kemiringan kepala. Selain itu, sebagian besar penelitian sebelumnya dalam deteksi kantuk berbasis visual masih menggunakan pendekatan yang berfokus pada analisis citra statis atau frame tunggal. Pendekatan tersebut belum sepenuhnya mampu memanfaatkan informasi temporal yang terdapat pada data video, padahal gejala kantuk pada pengemudi umumnya muncul sebagai perubahan perilaku visual yang berlangsung secara bertahap dari waktu ke waktu. Oleh karena itu, diperlukan pendekatan yang mampu

memodelkan hubungan spasial dan temporal secara simultan agar pola perubahan tersebut dapat dikenali dengan lebih akurat.

Perkembangan arsitektur *deep learning* berbasis *transformer* telah menunjukkan kemampuan yang signifikan dalam pengolahan data visual dan video karena mampu memodelkan hubungan spasial dan temporal secara lebih komprehensif melalui mekanisme *self-attention*. Salah satu arsitekturnya adalah Video Vision Transformer (ViViT), yang dirancang untuk mengekstraksi representasi visual tidak hanya dari setiap frame secara individual, tetapi juga dari dinamika antarframe pada suatu urutan video. Kemampuan ini menjadikan ViViT berpotensi tinggi untuk diterapkan pada sistem deteksi kantuk pengemudi, yang pada dasarnya memerlukan analisis perubahan visual secara berkelanjutan dari waktu ke waktu.

Penelitian terkait deteksi kantuk pengemudi masih didominasi oleh pendekatan yang berfokus pada indikator tunggal, seperti penutupan mata, frekuensi kedipan, atau ekspresi wajah tertentu, serta banyak yang masih bergantung pada arsitektur konvensional berbasis CNN atau CNN-LSTM. Pendekatan tersebut belum sepenuhnya mampu menangkap keterkaitan temporal yang kompleks antara beberapa indikator visual utama kantuk, seperti kondisi mata, mulut menguap, dan kemiringan kepala, yang muncul secara simultan dan saling melengkapi dalam merepresentasikan kondisi kantuk pengemudi. Penelitian yang secara khusus memanfaatkan ViViT untuk mengintegrasikan ketiga indikator visual tersebut dalam satu kerangka deteksi yang utuh masih relatif terbatas.

Berdasarkan kondisi tersebut, dapat diidentifikasi adanya penelitian yang belum ada model deteksi kantuk pengemudi yang mampu menggabungkan indikator visual utama secara non-intrusif, akurat, dan efisien, sekaligus memanfaatkan informasi spasial-temporal video secara optimal. Permasalahan ini menjadi penting karena sistem deteksi kantuk tidak cukup hanya akurat pada kondisi laboratorium, tetapi juga harus memiliki ketahanan terhadap variasi kondisi nyata, seperti perubahan pencahayaan, pose kepala, sudut wajah, dan dinamika gerakan pengemudi. Oleh karena itu, diperlukan pengembangan model berbasis

ViViT yang dirancang untuk mendeteksi kantuk pengemudi melalui integrasi indikator mata, mulut menguap, dan kemiringan kepala, sehingga mampu memberikan performa deteksi yang lebih andal serta memiliki potensi implementasi *real-time* sebagai sistem peringatan dini untuk meningkatkan keselamatan berkendara.

Berdasarkan kondisi tersebut, masalah pada penelitian ini adalah belum optimalnya model deteksi kantuk pengemudi yang mampu mengintegrasikan indikator visual utama, yaitu kondisi mata, mulut menguap, dan kemiringan kepala, secara non-intrusif dalam satu kerangka analisis spasial-temporal berbasis video, khususnya dengan memanfaatkan arsitektur Video Vision Transformer (ViViT) untuk kebutuhan deteksi yang akurat dan berpotensi diterapkan secara real-time.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, penelitian ini difokuskan pada penyelesaian masalah belum optimalnya model deteksi kantuk pengemudi berbasis video yang mampu mengintegrasikan indikator visual utama secara non-intrusif dalam satu kerangka analisis spasial-temporal. Permasalahan ini menjadi penting karena gejala kantuk tidak muncul sebagai peristiwa visual tunggal, melainkan sebagai rangkaian perubahan yang melibatkan kondisi mata, mulut menguap, dan kemiringan kepala secara simultan dan temporal. Oleh karena itu, diperlukan suatu model yang tidak hanya mampu merepresentasikan karakteristik visual setiap frame, tetapi juga memahami hubungan antarframe secara efektif. Arsitektur Video Vision Transformer (ViViT) dipandang memiliki potensi untuk menjawab kebutuhan tersebut, namun efektivitasnya dalam deteksi kantuk pengemudi, khususnya pada dataset NTHU-DDD dan dalam skenario implementasi *real-time*, masih perlu dibuktikan secara empiris. Dengan demikian, rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang model deteksi kantuk pengemudi berbasis Video Vision Transformer (ViViT) yang mampu memanfaatkan informasi spasial dan temporal dari data video secara efektif?

2. Bagaimana mengintegrasikan kondisi mata, mulut menguap dan kemiringan kepala sebagai indikator visual utama dalam proses deteksi kantuk pengemudi berbasis video?
3. Sejauh mana model ViViT yang diusulkan mampu membedakan kondisi waspada dan mengantuk secara akurat pada dataset NTHU Driver Drowsiness Detection (NTHU-DDD)?
4. Bagaimana merancang prototype sistem deteksi kantuk pengemudi yang non-intrusif dan real-time?

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah penelitian ini menghasilkan solusi dan aplikatif terhadap permasalahan deteksi kantuk pengemudi yang hingga saat ini masih menghadapi berbagai keterbatasan, baik dari sisi akurasi, kepraktisan implementasi, maupun kemampuan model dalam menangkap perubahan perilaku visual yang berlangsung secara bertahap. Tujuan penelitian ini adalah sebagai berikut :

1. Mengembangkan model deteksi kantuk pengemudi berbasis Video Vision Transformer (ViViT) pada data video wajah pengemudi.
2. Mengintegrasikan indikator visual kondisi mata, mulut menguap dan kemiringan kepala sebagai representasi utama dalam deteksi kantuk.
3. Mengevaluasi kinerja model yang diusulkan menggunakan metrik accuracy, precision, recall, F1-score, ROC-AUC, dan waktu inferensi.
4. Merancang prototype sistem deteksi kantuk non-intrusif dan real-time yang dapat digunakan untuk mendukung keselamatan berkendara.

1.5 Kebaruan (*Novelty*)

Novelty atau kebaruan dari model sistem informasi deteksi mata kantuk yang menggunakan metode Video Vision Transformer (VIVIT) adalah :

a. Kebaruan metodologi yaitu Pengembangan arsitektur Factorized Video Vision Transformer (ViViT) sebagai arsitektur utama untuk deteksi kantuk

pengemudi berbasis video, sehingga model tidak hanya menangkap fitur spasial pada tiap frame, tetapi juga dinamika temporal antar frame. Pada penelitian ini tidak mengklaim bahwa ViViT merupakan arsitektur baru namun kebaruan penelitian terletak pada adaptasi dan penerapan ViViT untuk deteksi kantuk pengemudi melalui integrasi tiga indikator visual utama, yaitu kondisi mata dan kemiringan kepala, dalam pipeline spasial-temporal yang dirancang untuk konteks keselamatan berkendara.

b. Kebaruan representasi fitur yaitu pemanfaatan tiga indikator visual utama secara terintegrasi, yakni kondisi mata, mulut menguap dan kemiringan kepala, sebagai dasar klasifikasi kondisi waspada dan kantuk dalam satu kerangka spasial-temporal.
c. Kebaruan implementatif yaitu penyusunan pipeline deteksi kantuk berbasis video yang mencakup ekstraksi frame, deteksi wajah dengan metode HRNet, normalisasi, tokenisasi patch spasial-temporal, dan klasifikasi berbasis transformer pada dataset driver drowsiness dengan variasi kondisi visual.
d. Kebaruan evaluatif yaitu penilaian performa model tidak hanya berdasarkan metrik klasifikasi, tetapi juga dengan mempertimbangkan waktu inferensi, sehingga relevansinya terhadap implementasi real-time.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini adalah :

1. Mampu memberikan pencegahan dini kecelakaan yang sering terjadi akibat pengemudi mengantuk.
2. Meningkatkan keamanan pengemudi dan penumpang dengan memberikan tindakan preventif. Peringatan dini dapat memungkinkan pengemudi untuk mengambil tindakan, seperti beristirahat, sebelum kondisi kantuk menjadi lebih serius.
3. Memberikan pemahaman yang lebih baik tentang tingkat kewaspadaan individu selama aktivitas yang memerlukan perhatian, seperti mengemudi. Hal

ini dapat membantu individu untuk mengelola waktu istirahat mereka dengan lebih baik.

4. Menunjukkan kemampuan dan manfaat teknologi cerdas dalam memberikan solusi untuk masalah nyata. Sistem ini menciptakan lingkungan yang cerdas dan adaptif, memanfaatkan teknologi untuk meningkatkan keamanan dan kesejahteraan.

1.7 Ruang Lingkup Penelitian

Ruang lingkup penelitian untuk sistem deteksi mata kantuk berdasarkan analisis kondisi mata dengan metode *Video Vision Transformer* (VIVIT) bisa mencakup beberapa aspek penting. Tujuan dari ruang lingkup ini adalah untuk menentukan batasan dan fokus penelitian sehingga dapat terarah dengan baik. Berikut ini adalah beberapa komponen yang bisa menjadi bagian dari ruang lingkup penelitian tersebut:

1. Penelitian ini menggunakan dataset NTHU Drowsiness Detection Dataset. Variabilitas fitur: kondisi mata, mulut menguap dan kemiringan kepala. Pada penelitian hanya dilakukan untuk menangkap fitur dari wajah manusia.
2. Variabel dan parameter dalam penelitian ini meliputi :
    a. Kemiringan kepala yang menggambarkan posisi kepala pengemudi selama berkendaraan.
    b. Kondisi mata : meliputi kondisi terbuka dan tertutup mata pengemudi yang menjadi indikator mengantuk dalam mendeteksi kantuk.
    c. Mulut menguap : mulut menguap mencerminkan respons fisiologis yang berkaitan dengan kelelahan dan penurunan kewaspadaan, sehingga dapat digunakan sebagai indikator visual tambahan dalam deteksi kantuk pengemudi. Variabel ini direpresentasikan melalui perubahan spasial pada area mulut, khususnya peningkatan bukaan mulut yang teridentifikasi pada rangkaian frame video. Informasi ini kemudian dianalisis bersama indikator kondisi mata dan kemiringan kepala untuk meningkatkan akurasi dalam mengidentifikasi kondisi kantuk secara lebih komprehensif.

3. Metode yang digunakan untuk mengklasifikasi kondisi kantuk dan tidak mengantuk dengan menggunkan video vision transformer,dengan mengekstaksi fitur spasial dan temporal. Metode Video Vision Transformer (ViViT) memiliki kemampuan untuk menangkap hubungan temporal antar frame dalam satu urutan video. Hal ini dimungkinkan melalui penerapan *self-attention mechanism* yang menjadi inti dari arsitektur Transformer. Mekanisme ini memungkinkan ViViT untuk secara adaptif memperhatikan keterkaitan antar frame, sehingga mampu mengenali pola-pola sekuensial yang menunjukkan perubahan kondisi kantuk secara bertahap, seperti penutupan mata yang terjadi secara perlahan, perubahan arah pandangan, atau kecenderungan posisi kepala yang bergeser.
4. Untuk mengukur keakuratan dan kehandalan sistem deteksi kantuk yang dikembangkan, dilakukan proses evaluasi kinerja model secara komprehensif dengan menggunakan beberapa metrik evaluasi diterapkan dalam bidang machine learning :
   a. Akurasi *(accuracy)* sebagai ukuran utama untuk menghitung proporsi prediksi yang benar terhadap seluruh data pengujian. Akurasi memberikan gambaran umum mengenai tingkat ketepatan model secara keseluruhan.
   b. Precision dan Recall untuk mengukur kinerja model secara lebih mendetail. Precision mengindikasikan sejauh mana prediksi positif yang dihasilkan oleh model benar-benar sesuai dengan kondisi sebenarnya, sehingga berkaitan erat dengan kemampuan model dalam meminimalkan kesalahan positif (*false positive*). Sementara itu, recall menunjukkan kemampuan model dalam mendeteksi seluruh kasus positif yang sebenarnya terjadi, sehingga berhubungan dengan kemampuan model untuk mengurangi kesalahan negatif (*false negative*), yang pada konteks deteksi kantuk sangat penting untuk menghindari kegagalan sistem dalam mendeteksi kondisi kantuk yang berbahaya.
   c. F1-Score yang merepresentasikan harmoni antara kedua metrik tersebut. F1-Score menjadi metrik yang sangat relevan dalam kasus di mana

terdapat ketidakseimbangan kelas deteksi kantuk (*false negative*) harus ditekan semaksimal mungkin.

d. *Inference Time* diukur sebagai rata-rata waktu yang diperlukan model untuk memproses satu unit input, baik dalam bentuk frame maupun klip video, hingga menghasilkan output klasifikasi. Pengukuran ini mencerminkan kompleksitas komputasi model serta efisiensi arsitektur yang digunakan. Evaluasi terhadap inference time menjadi penting untuk memastikan bahwa model tidak hanya memiliki kinerja klasifikasi yang baik, tetapi juga mampu beroperasi secara efisien pada skenario implementasi nyata yang membutuhkan respons cepat dan berkelanjutan. Dalam penelitian ini, inference time digunakan untuk menilai kesiapan model dalam mendukung sistem deteksi kantuk berbasis video secara real-time.

1.8 Sistematika Penulisan

Struktur penulisan yang diterapkan dalam naskah disertasi ini mencakup beberapa pokok pembahasan sebagai berikut.

Bab I Pendahuluan

Bab ini dijelaskan tentang latar belakang masalah, identifikasi masalah, rumusan masalah, maksud dan tujuan penelitian, kebaruan penelitian, serta manfaat peneltiian, ruang lingkup penelitian, kontribusi penelitian serta sistematika penulisan.

Bab II Tinjauan Pustaka dan Landasan Teori

Bab ini membahas tentang tinjauan pustaka, keaslian penelitian, dan landasan teori yang digunakan.

Bab III Metodologi Penelitian

Bab ini menjelaskan alur pengembangan model deteksi kantuk diihat dari analisa kondisi mata dan kemiringan kepala dengan menggunakan metode vidio vision transformer.

BAB IV Hasil dan Pembahasan

Bab ini menjelaskan hasil penelitian yang dilakukan secara komprehensif meliputi pengujian, hasil, dan analisis pada setiap eksperimen skenario yang telah dilakukan.

Bab V Penutup

Bab ini merupakan pembahasan kesimpulan dari serangkaian hasil analisis serta evaluasi yang dilakukan dengan menjawab tujuan penelitian dan saran pengembangan pada penelitian selanjutnya.

keadaan matanya tertutup selama kurang lebih 1 detik (Khan, 2014). Berdasarkan KSS, skala tingkat 7 merupakan kondisi dimana manusia mulai merasakan kantuk tetapi tidak membutuhkan usaha yang berat untuk tetap terjaga. Kondisi terebut dapat dikatakan sebagai kondisi lelah. Manusia dalam kodisi lelah ditandai dengan menguap. Menguap adalah suatu keadaan diamana mata dalam keadaan tertutup atau setengah terbuka disertai dengan mulut yang terbuka lebar selama kurang lebih 1 detik (Azim dkk., 2009)

Penelitian ini diawali dengan upaya menilai tingkat kantuk pengemudi melalui pengamatan ekspresi wajah yang terekam, yang kemudian dinilai menggunakan skala kantuk standar. Peneliti mengeksplorasi indeks prediktor efektif seperti frekuensi kedipan, durasi penutupan mata, serta perubahan ekspresi wajah lain yang berkaitan dengan munculnya rasa kantuk. Data diperoleh melalui eksperimen simulasi mengemudi dalam kondisi terkontrol, memungkinkan pemantauan perilaku visual secara detail. Hasil penelitian menunjukkan bahwa indikator visual seperti peningkatan frekuensi kedipan dan durasi mata tertutup memiliki korelasi yang signifikan dengan tingkat kantuk pengemudi yang diukur secara subyektif, menjadikannya prediktor potensial untuk sistem deteksi kantuk otomatis. Kelebihan penelitian ini adalah fokusnya pada ekspresi wajah yang dapat dipantau secara non-invasif, namun keterbatasan utamanya adalah pengujian hanya dilakukan dalam skenario simulasi, sehingga diperlukan validasi lebih lanjut di lingkungan mengemudi nyata . Skala kantuk yang digunakan dibagi menjadi 5 tingkat (*point scale*) yang dinilai berdasarkan rekaman video wajah pengemudi (Kitajima dkk., 1997):

1. Tingkat 1 – Waspada (*alert)*, tidak ada tanda kantuk: mata terbuka normal, tidak ada penurunan frekuensi kedipan.
2. Tingkat 2 – Sedikit mengantuk: mulai menunjukkan peningkatan frekuensi kedipan, namun tetap responsif.
3. Tingkat 3 – Mengantuk sedang: kedipan mata melambat, durasi penutupan mata mulai terlihat lebih lama, wajah tampak kurang ekspresif.

4. Tingkat 4 – Sangat mengantuk: frekuensi kedipan tinggi, durasi mata tertutup >1 detik, terkadang kepala mulai terangguk.
5. Tingkat 5 – Tertidur sesaat (*microsleep*): mata tertutup lama, hilang kontak visual dengan jalan, kepala terangguk jelas.

Penelitian ini, tingkat kantuk pengemudi diklasifikasikan berdasarkan standar *Karolinska Sleepiness Scale* (KSS) yang telah dikelompokkan kembali guna menyesuaikan dengan karakteristik data visual. Skor KSS yang semula memiliki rentang satu hingga sembilan didistribusikan ke dalam empat kategori kelas utama berdasarkan kesamaan kondisi klinis dan perilaku yang ditunjukkan oleh subjek. Pendekatan pengelompokkan ini diadaptasi dari metode yang diusulkan oleh Ghoddoosian (Ghoddoosian dkk., 2019), namun dilakukan pengembangan lebih lanjut dengan menyertakan seluruh skala guna menangkap gradasi transisi kondisi pengemudi secara lebih detail. Pembagian empat kelas ini meliputi: kondisi terjaga penuh (Kelas 1), penurunan kewaspadaan atau fase transisi (Kelas 2), kondisi mengantuk (Kelas 3), hingga fase kantuk berat yang kritis (Kelas 4). Klasifikasi hierarkis ini bertujuan untuk melatih model ViViT agar mampu mengenali perubahan fitur spasio-temporal pada wajah pengemudi secara lebih presisi di setiap fasenya.

2.3.3 Pengolahan Citra Digital Wajah Dalam Sistem Komputer

Pengolahan citra wajah dimulai dengan deteksi wajah, yaitu proses otomatis yang melokalisasi area wajah dalam citra. Beberapa metode klasik, seperti algoritma Viola–Jones, menggunakan fitur Haar dan AdaBoost untuk mendeteksi wajah secara real-time pada citra (Viola & Jones, 2001) . Penelitian modern telah mengintegrasikan deep learning dalam tahap deteksi, seperti pendekatan Faster R-CNN yang diperkuat dengan PCA untuk meningkatkan akurasi pada dataset nyata(Muhammad dkk., 2021). Tahapan ini memiliki peran penting sebagai dasar untuk proses selanjutnya berupa prapemrosesan dan ekstraksi fitur. Tahap ekstraksi fitur menjadi inti sistem pengenalan wajah. Metode awal seperti PCA (eigenfaces) dan LDA (fisherfaces) berfokus pada reduksi dimensi dan representasi statistik wajah dalam ruang eigen, yang telah

diuji dalam berbagai dataset seperti ORL dan Yale-B(Eleyan & Demirel, 2006) . Teknik tekstur seperti LBP juga digunakan untuk menangkap pola lokal pada wajah, dan dikombinasikan dengan CNN untuk menambah kekokohan terhadap variasi pencahayaan dan ekspresi(J. Tang dkk., 2020) . Tren terkini mengandalkan arsitektur deep CNN, seperti FaceNet dan DeepFace, yang menghasilkan embedding vektor berdimensi tinggi (128 atau 4096) dengan teknik loss seperti triplet loss dan margin-based softmax, mencapai performa hingga ~99 % akurasi pada LFW dan YouTube Faces (William dkk., 2019) .

Setelah fitur diekstraksi tahap matching dan pengenalan wajah dilakukan dengan membandingkan embedding dengan basis data menggunakan metric seperti jarak Euclidean atau cosine similarity. Kualitas sistem sangat dipengaruhi oleh prapemrosesan seperti normalisasi pose (face alignment) dan super-resolution citra rendah . Selain itu, aspek keamanan seperti anti-spoofing menjadi penting untuk mencegah pemalsuan (misalnya menggunakan foto atau deepfake), yang ditangani melalui analisis tekstur serta pola frekuensi temporal oleh model berbasis CNN . Kombinasi komponen-komponen ini deteksi, praproses, ekstraksi fitur, matching, dan keamanan membentuk sistem pengenalan wajah modern yang andal dan adaptif terhadap tantangan dunia nyata. Tahap ekstraksi fitur merupakan inti dari pengolahan citra digital wajah. Pendekatan klasik seperti PCA (eigenface), LDA (fisherface), HOG, atau LBP menganalisis struktur statistik dan tekstur wajah dalam bentuk vektor berdimensi rendah . Namun, sejak tahun 2014, penggunaan deep learning, khususnya CNN, telah merevolusi ekstraksi fitur secara otomatis dan hierarkis, memberikan representasi wajah yang jauh lebih kaya dan efektif.

Teknologi digitalisasi dapat membantu dalam bidang transportasi modern mengoptimalkan kontribusi ekonomi, mengurangi pekerjaan, serta mengatasi solusi untuk area yang terisolasi. Sekarang ada metode yang mengurangi eksperimen kontak fisik dengan hewan untuk meningkatkan kenyamanan hewan dan menghindari wabah penyakit. Metode ini mulai banyak digunakan, menggunakan sensor, computer vision data video, big data,

dan teknologi blockchain untuk keuntungan bersama antara produsen ternak, konsumen, dan hewan ternak itu sendiri(Neethirajan & Kemp, 2021). Citra digital dikirimkan secara elektronik, file ditransfer ke media penyimpanan fisik, identifikasi obyek berdasarkan foto dan video adalah salah satu layanan yang bisa digunakan(Taoufik dkk., 2022). Proses ekstraksi citra digital ada beberapa tahap, diantaranya meliputi citra digital diambil setiap frame, kecepatan pengambilan adalah 30 frame per detik dan setiap frame dibaca oleh algoritma yang dikembangkan, rubah citra Red, Green, Blue (RGB) ke Hue Saturation Value (HSV). Citra yang ditangkap oleh webcam dalam format piksel RGB, Ambang citra HSV digunakan untuk mendapatkan ambang batas, merupakan metode untuk memisahkan daerah yang lebih tinggi atau lebih rendah dari nilai ambang batas yang telah ditetapkan berdasarkan nilai batas atas dan batas bawah. Ambang batas diterapkan untuk semua piksel citra dengan mengubah citra HSV menjadi citra biner. Citra biner merupakan satu set array yang memiliki nilai 0 dan 1, tanda bitwise sangat berguna saat mengekstrak bagian mana pun dari citra, yang termasuk operasi bitwise adalah and, or, not dan xor. Citra kontur dengan warna hijau sebagai ambang batas, kontur merupakan kurva yang menghubungkan semua titik secara kontinyu, yang memiliki warna atau intensitas yang sama. Dalam menghitung piksel per rasio matrik, untuk menentukan sifat geometris objek pada suatu citra, terlebih dahulu kita perlu melakukan kalibrasi menggunakan objek referensi. Objek referensi tidak lain adalah objek yang tinggi dan lebarnya akan dihitung (Gupta dkk., 2023)

Pengesahan wajah dimulai dengan tahap deteksi wajah, yakni identifikasi dan pelokalan area wajah dalam citra. Metode awal menggunakan teknik berbasis fitur seperti warna kulit, tepi, dan bentuk wajah, atau pola frekuensi seperti DCT dan morfologi matematika . Tahap ini bertujuan mengekstrak *region of interest* (ROI) yang menjadi input untuk serangkaian proses selanjutnya. Citra wajah yang telah dideteksi diproses pada fase prapemrosesan untuk memperbaiki kondisi citra. Teknik seperti normalisasi pencahayaan, pengurangan noise, dan perataan pose 2D–3D (frontalization)

kerap digunakan; contohnya metode 2D/3D alignment pada DeepFace yang mampu mengubah pose ke arah frontal secara akurat.

Dalam *face recognition*, CNN tidak hanya dipakai sebagai ekstraktor fitur, melainkan juga sebagai model end-to-end untuk klasifikasi wajah. Model seperti DeepFace dengan sembilan lapisan neural network menghasilkan vektor fitur (dimension 4096) yang selanjutnya dicocokkan menggunakan jarak Euclidean atau loss berbasis softmax terdiscriminatif(M. Wang & Deng, 2018) . Arsitektur dan fungsi loss modern semakin berkembang, misalnya margin-based softmax dan regularisasi, guna meningkatkan kapasitas diskriminatif sistem .

Kendala nyata dalam algoritma pengenalan wajah antara lain meliputi variasi pencahayaan, pose (sampai pandangan samping), ekspresi, occlusion, dan resolusi rendah. Survey terbaru menunjukkan adanya peningkatan algoritma super-resolution wajah (*face super-resolution*) untuk memperbaiki kualitas citra sebelum ekstraksi fitur . Selain itu, metode pose-invariant recognition dirancang agar sistem tetap efektif walau wajah tidak tampak frontal . Evaluasi kualitas citra (*face image quality assessment*) juga menjadi aspek penting karena kualitas data citra sangat mempengaruhi performa akhir sistem. Penelitian terbaru menyoroti kebutuhan model yang mampu menilai utilitas biometrik citra wajah secara otomatis dan integratif ke dalam pipeline pengenalan(Schlett dkk., 2022). Di samping itu, aspek anti-spoofing seperti pendeteksian wajah palsu atau manipulasi (*deepfake/forgery*) semakin mendapatkan perhatian. Teknik berbasis deep learning mendeteksi artefak tekstur, frekuensi, dan pola temporal untuk menghindari serangan identitas . Secara keseluruhan, rangkaian algoritma pengolahan citra wajah mulai deteksi, prapemrosesan, ekstraksi fitur, hingga matching dan keamanan menggunakan pendekatan klasik dan modern. Revolusi dimulai dari PCA/LDA dan berkembang pesat lewat CNN dan deep learning, memacu akurasi sistem ke level state-of-the-art, terutama di lingkungan dunia nyata yang menantang. Faktor-faktor seperti penyesuaian pose, kualitas citra rendah, dan ancaman

spoofing menjadi area riset aktif untuk meningkatkan keandalan sistem secara menyeluruh .

Pengolahan citra wajah dalam sistem komputer merupakan bagian dari bidang pengolahan citra digital yang berfokus pada analisis dan transformasi visual dari wajah manusia. Proses ini mencakup serangkaian langkah seperti peningkatan kualitas citra, penghilangan noise, normalisasi pencahayaan, dan koreksi geometri untuk menghasilkan citra wajah yang lebih representatif sebelum digunakan dalam sistem pengenalan atau analisis ekspresi wajah. Tahapan ini dapat dilakukan tanpa melibatkan identifikasi individu secara eksplisit, tetapi lebih ditujukan untuk meningkatkan kualitas visual data wajah itu sendiri agar sistem selanjutnya bekerja secara optimal (Jiang & Chen, 2008) .

2.3.4 Estimasi Kemiringan Kepala

Kemiringan kepala merupakan salah satu indikator visual penting yang digunakan dalam sistem deteksi kantuk karena secara fisiologis mencerminkan penurunan tingkat kewaspadaan seseorang. Dalam penelitian yang dilakukan oleh Vinay Shashidhar dan timnya (2022), kemiringan kepala dianalisis melalui sudut orientasinya terhadap sumbu vertikal dengan memanfaatkan 68 titik landmark wajah dari pustaka dlib. Penelitian ini secara khusus mengukur dua jenis kemiringan: *tilt*, yaitu kemiringan kepala ke samping kiri atau kanan, serta *nodding*, yaitu gerakan menunduk ke depan secara perlahan atau tiba-tiba. Kedua jenis gerakan ini dianggap sebagai gejala awal kantuk karena saat tingkat kewaspadaan menurun, kontrol otot leher melemah sehingga kepala secara tidak sadar cenderung terangguk atau condong ke satu sisi. Dalam sistem yang mereka rancang, diterapkan ambang batas sudut tertentu misalnya lebih dari 15 derajat untuk tilt dan lebih dari 20 derajat untuk nodding yang dijadikan parameter untuk mendeteksi kantuk. Ketika kepala melampaui sudut tersebut dalam durasi tertentu atau dengan frekuensi yang tinggi, sistem akan memberikan sinyal bahwa subjek berada dalam kondisi mengantuk. Pendekatan ini terbukti efektif dalam mendeteksi perubahan postur kepala secara real-time dan non-invasif, meskipun tetap memiliki keterbatasan

terhadap variabilitas posisi duduk, bentuk wajah, atau penggunaan aksesoris kepala yang dapat memengaruhi akurasi estimasi sudut (Shashidhar dkk., 2024) .

Sistem klasifikasi tahapan tidur berdasarkan sensor percepatan kepala (*head acceleration sensor*) tiga sumbu. Data dikumpulkan dari sensor yang dipasang di kepala pengguna selama tidur, lalu dianalisis menggunakan machine learning untuk mengklasifikasikan tahapan seperti bangun, *Rapid Eye Movement* (REM), tidur ringan, dan tidur dalam. Penelitian ini melakukan pendekatan berbasis getaran kepala (*ballistocardiogram*) yang ringan dan murah dibandingkan polisomnografi. Akan tetapi, metode ini kurang akurat dalam mendeteksi tahap REM, dan sensitif terhadap noise atau gerakan ekstrem kepala. Meskipun demikian, hasil penelitian cukup menjanjikan dengan akurasi keseluruhan mencapai 74,6%, membuka peluang aplikasi portable monitoring tidur di rumah (Yoshihi dkk., 2021).

2.3.5 Computer Vision

Computer vision adalah bidang ilmu interdisipliner yang membahas bagaimana model komputasi dapat digunakan untuk mendapatkan pemahaman dari citra atau video, sehingga membangun suatu sistem seperti yang dapat dilakukan oleh sistem visual manusia (Choi dkk., 2018). Kombinasi machine learning dan deep learning dengan computer vision mampu meningkatkan pemahaman komputer, seperti menyederhanakan proses deteksi dan prediksi. Computer Vision telah dikembangkan untuk secara otomatis mengekstrak fitur kompleks dan terus menerus belajar dari data training (Aziz dkk., 2021).

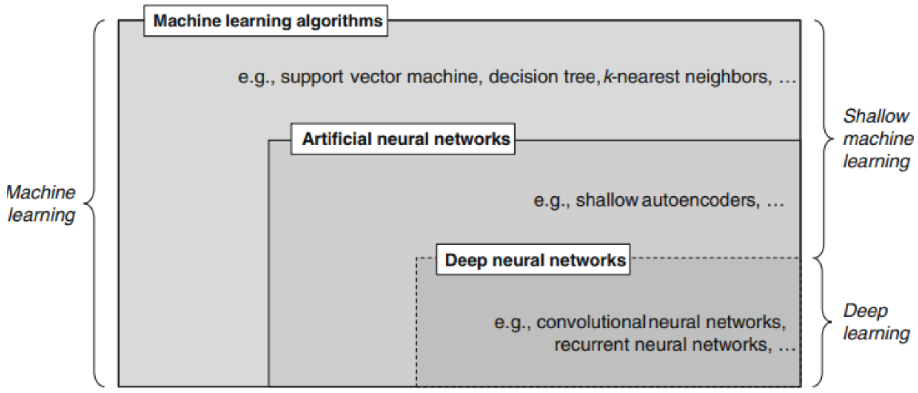
Keuntungan dari computer vision terletak pada pemantauan hewan noninvasif dan berbiaya rendah. Dengan demikian, sistem ini memungkinkan ekstraksi informasi dengan gangguan eksternal minimal (misalnya, penyesuaian sensor oleh manusia) dengan biaya yang terjangkau. Komponen utama dari sistem computer vision termasuk kamera, unit perekam, unit pemrosesan, dan model (Li, Huang, dkk., 2021).

Persepsi penglihatan manusia di adopsi dengan nama computer vision, persepsi visual pada dasarnya adalah tindakan mengamati dan mengenal pola serta 30objek melalui penglihatan atau input visual, secara sederhana sistem penglihatan manusia terdiri dari sensor atau mata untuk menangkap citra dan otak untuk memproses serta menafsirkan citra. Sistem yang ada di komputer merupakan refleksi dari sistem penglihatan visual yang ada pada manusia, untuk dapat meniru sistem tersebut di butuhkan dua komponen pokok yang pertama sensing device sebagai fungsi meniru mata pada manusia dan yang kedua interpreting device yang berfungsi sebagai otak.

2.3.6 Deep Learning

Deep Learning (DL), juga disebut pembelajaran hierarkis dan pembelajaran terstruktur dalam, adalah bagian dari machine learning yang mengeksploitasi penggunaan banyak lapisan pemrosesan informasi non-linear untuk ekstraksi dan transformasi fitur secara supervised maupun unsupervised, serta analisis pola dan klasifikasi (Purwins dkk., 2019).Dalam beberapa tahun terakhir, model Deep Learning telah mendapatkan momentum dalam masalah pencitraan seperti resolusi super gambar dan video, restorasi gambar atau pewarnaan ulang (López-Tapia dkk., 2021).

Deep Learning sangat berguna dalam domain dengan data berdimensi besar dan tinggi, itulah sebabnya jaringan saraf dalam mengungguli algoritma machine earning untuk sebagian besar aplikasi di mana data teks, gambar, video, ucapan, dan audio perlu diproses (Janiesch dkk., 2021). Konsep dan kelas deep learning dapat dilihat pada Gambar 2.3.



Gambar 2.1 Konsep dan Kelas *Deep Learning* (Janiesch dkk., 2021)

2.3.7 Transformer

Transformer merupakan arsitektur model yang pertama kali diperkenalkan dalam artikel Attention Is All You Need (Vaswani dkk., 2017). Model ini mengubah paradigma dalam pengolahan urutan data (*sequence processing),* terutama dalam penerjemahan mesin (*machine translation*) dan aplikasi Natural Language Processing (NLP). Sebelumnya, model-model berbasis Recurrent Neural Networks (RNN), seperti LSTM (Long Short-Term Memory) dan GRU (Gated Recurrent Unit), serta Convolutional Neural Networks (CNN) lebih dominan dalam pengolahan urutan data. Transformer memperkenalkan pendekatan yang sepenuhnya bergantung pada self-attention untuk menangkap ketergantungan dalam urutan data. Sebelum adanya Transformer, arsitektur utama yang digunakan dalam *sequence transduction* (mengubah satu urutan menjadi urutan lainnya) adalah model Recurrent Neural Networks (RNN). RNN menggunakan mekanisme rekursif di mana output dari setiap waktu bergantung pada input saat ini dan informasi tersembunyi dari waktu sebelumnya. Hal ini memungkinkan RNN untuk memproses urutan data secara bertahap, namun memiliki kelemahan dalam menangkap ketergantungan jarak jauh karena masalah vanishing gradient. Untuk mengatasi masalah ini, LSTM (Long Short-Term Memory) (Hochreiter & Schmidhuber, 1997). LSTM mengatasi masalah vanishing gradient dengan memperkenalkan gated mechanisms yang memungkinkan memori untuk mempertahankan informasi lebih lama. Meskipun LSTM banyak digunakan dalam aplikasi NLP

dan penerjemahan mesin, mereka masih memerlukan proses sekuensial, yang membatasi paralelisasi dalam pelatihan dan menyebabkan waktu pelatihan yang lebih lama.

Salah satu ide yang membawa kita lebih dekat ke Transformer adalah attention mechanism. Attention digunakan untuk memfokuskan model pada bagian tertentu dari input yang relevan, terlepas dari jarak posisional. Memungkinkan model untuk menangkap ketergantungan panjang lebih baik daripada model berbasis RNN(Israr dkk., 2023), yang memperkenalkan cara baru untuk menangkap ketergantungan dalam penerjemahan mesin. Meskipun model berbasis RNN + Attention memberikan hasil yang lebih baik daripada model murni RNN, masih ada kekurangan dalam hal paralelisasi dan efisiensi komputasi.

2.3.8 Vision Transformer

Transformer yang awalnya dikembangkan untuk tugas pemrosesan bahasa alami (NLP), telah terbukti sangat efektif dalam pengolahan data visual, seperti pengenalan citra dan deteksi objek. Konsep dasar dari Transformer adalah mekanisme *self-attention*, yang memungkinkan model untuk memproses setiap bagian data secara paralel dan menangkap hubungan jangka panjang antara elemen-elemen dalam data input. Ini berbeda dengan pendekatan berbasis konvolusi yang biasanya memproses data secara lokal dalam grid yang lebih kecil. Dalam konteks visual, kemampuan Transformer untuk menangkap dependensi global pada citra sangat berharga, terutama pada citra berresolusi tinggi dan dalam aplikasi yang membutuhkan pemahaman tentang konteks yang lebih luas dalam image.

Self-attention merupakan inti dari arsitektur Transformer. Mekanisme ini bekerja dengan memberi bobot pada setiap bagian input untuk menentukan seberapa penting informasi tersebut dalam konteks seluruh input. Sebagai contoh, dalam pemrosesan citra, Transformer memecah citra menjadi *patches* atau potongan-potongan kecil dan memperlakukan setiap *patch* sebagai token yang diolah dalam urutan. Setiap *patch* ini kemudian "berkomunikasi" dengan

patch lainnya melalui mekanisme self-attention untuk menghasilkan representasi yang lebih kompleks dari citra tersebut. Transformer dapat menangkap informasi lokal dan global dalam satu model, yang menjadikannya sangat unggul dibandingkan dengan model konvolusional (CNN) dalam tugas yang melibatkan pemahaman citra secara lebih mendalam (Pan dkk., 2021).

Vision Transformer (ViT) diperkenalkan sebagai penerapan langsung dari Transformer pada tugas pengolahan citra. ViT memecah citra menjadi *patches* yang lebih kecil dan memperlakukan mereka sebagai urutan token, mirip dengan bagaimana Transformer diterapkan pada teks. Keuntungan utama dari ViT adalah kemampuannya untuk menangkap dependensi global dalam citra yang lebih kompleks, yang sangat berguna untuk tugas-tugas seperti klasifikasi citra dan deteksi objek. Namun, penerapan ViT juga menghadapi tantangan, terutama dalam hal kebutuhan akan dataset besar untuk pelatihan. ViT membutuhkan lebih banyak data dan waktu pelatihan dibandingkan dengan CNN tradisional untuk dapat bersaing dalam akurasi, terutama pada dataset yang lebih kecil. Meski demikian, dengan pre-training pada dataset besar seperti ImageNet atau JFT-300M, ViT menunjukkan hasil yang sangat kompetitif, bahkan mengungguli CNN pada banyak tugas pengolahan citra (Bin Mohamad Azmi & Kamaru Zaman, 2024) .

Penggunaan Transformer dalam pengolahan data EEG untuk mendeteksi kelelahan pengemudi. Model yang diusulkan TFormer, menggabungkan kemampuan Transformer untuk menangkap pola global dalam domain waktu dan frekuensi. TFormer menggunakan tiga komponen utama: konvolusi untuk penyematan input, perhatian silang multi-head (TF-MCA) untuk mengintegrasikan pola domain waktu ke dalam titik frekuensi, dan perhatian diri untuk mempelajari pola global lebih lanjut. Hasil eksperimen menunjukkan keunggulan TFormer dibandingkan dengan metode yang ada dalam mendeteksi kelelahan pengemudi menggunakan data EEG, terutama dalam pengaturan lintas subjek (R. Li dkk., 2024). Metode berbasis Transformer untuk estimasi posisi kepala yang efisien. Pendekatan ini menggabungkan pembelajaran token orientasi yang dapat menangkap

hubungan antar bagian wajah dalam citra meskipun terjadi occlusion (penutupan sebagian wajah). Dengan menggunakan Transformer, model ini belajar hubungan geometris antar bagian wajah dan dapat mengatasi masalah perubahan besar dalam orientasi kepala. TokenHPE menghasilkan kinerja terbaik dalam mengklasifikasikan orientasi kepala di berbagai dataset dengan akurasi tinggi (C. Zhang dkk., 2023) .

Penggunaan Depth and Visual Concepts Aware Transformer (DEVICE) untuk tugas captioning citra berbasis OCR. DEVICE memanfaatkan informasi kedalaman untuk memperbaiki pemahaman tiga dimensi dalam pengenalan teks pada citra. Metode ini mengintegrasikan dua modul utama, yaitu Depth-enhanced Feature Updating yang memperbarui fitur OCR dengan kedalaman, dan Semantic-guided Alignment, yang mengarahkan penggunaan informasi visual untuk meningkatkan akurasi dalam menghasilkan deskripsi teks. Dengan menggabungkan kedalaman dan konsep visual, model ini dapat menghasilkan caption yang lebih lengkap dan akurat. DEVICE berhasil mengatasi kelemahan dalam captioning citra OCR yang hanya mengandalkan hubungan dua dimensi, seperti yang dilakukan model-model sebelumnya. Dengan menambahkan kedalaman dan pemahaman konsep visual, DEVICE mampu membangun hubungan spasial dan geometris tiga dimensi antara objek dan teks, yang meningkatkan akurasi caption. Pada pengujian menggunakan dataset TextCaps, DEVICE menunjukkan peningkatan signifikan dalam performa, mengungguli model-model sebelumnya dalam menghasilkan caption yang lebih tepat dan komprehensif (D. Xu dkk., 2025).

Metode FFTran suatu jaringan berbasis Transformer untuk tugas klasifikasi citra multi-label. FFTran memanfaatkan dua teknik utama: Multi-level Scale Information Integration Mechanism (MSIIM) untuk mengintegrasikan fitur dari berbagai skala dan Intra-Image Spatial-Channel Semantic Mining (ISCM) untuk mengekstrak informasi spasial dan kanal yang relevan. Dengan menggabungkan kedua teknik ini, FFTran dapat menangani objek kecil yang sulit dikenali, yang sering menjadi tantangan pada klasifikasi citra multi-label. Teknik MSIIM memungkinkan fusi fitur multi-skala, yang

membantu dalam menangani objek dengan ukuran yang sangat berbeda dalam citra. Sementara itu, ISCM membantu meningkatkan pemahaman model terhadap hubungan spasial dan kanal yang ada dalam citra, memungkinkan klasifikasi yang lebih akurat. FFTran menunjukkan peningkatan yang signifikan dalam akurasi klasifikasi citra, terutama pada objek-objek kecil, yang sering kali terabaikan oleh model-model klasifikasi tradisional. Pada dataset yang terkenal, seperti VOC2007 dan COCO2014, FFTran mengungguli model-model state-of-the-art yang ada, membuktikan keunggulannya dalam menangani klasifikasi multi-label yang kompleks. Model ini memperlihatkan kemampuan luar biasa dalam mengatasi variasi ukuran objek yang ada di citra, serta memperbaiki kinerja pada situasi di mana objek kecil atau tersembunyi sering kali tertinggal dalam klasifikasi (P. Liu dkk., 2025) .

Temporal-Channel Visual Transformer (TCVT), sebuah model yang dirancang untuk tugas video summarization. TCVT mengintegrasikan informasi spasial dan temporal melalui dua komponen utama: dual-stream embedding module dan inter-frame encoder. Dual-stream embedding module memungkinkan model untuk menggabungkan fitur visual dan gerakan optik, sedangkan inter-frame encoder menangani korelasi antar-frame dengan menggunakan beberapa modul perhatian temporal dan kanal. Pendekatan ini memungkinkan TCVT untuk memahami hubungan temporal dan spasial antara frame-frame dalam video, serta menangkap informasi penting dari video yang mungkin terlewat oleh metode konvensional. Eksperimen yang dilakukan pada dataset SumMe dan TVSum menunjukkan bahwa TCVT berhasil mengungguli model-model sebelumnya dalam hal mean Average Precision (mAP) dan menghasilkan ringkasan video yang lebih representatif. TCVT menunjukkan kinerja terbaik pada tugas video summarization, baik dalam hal akurasi maupun ukuran model, dibandingkan dengan teknik-teknik lain yang ada. Dengan menggunakan pendekatan temporal dan kanal yang disesuaikan, TCVT mampu menghasilkan ringkasan video yang lebih akurat, yang memperhitungkan baik perubahan visual maupun gerakan dalam video (Tian dkk., 2025) .

ASAFormer, sebuah metode pelacakan visual yang menggabungkan Vision Transformer (ViT) dengan Asymmetric Selective Attention (ASA) untuk meningkatkan efisiensi komputasi dalam pelacakan objek. ASAFormer mengatasi masalah redundansi dalam perhatian dengan memilih token representatif, yang mengurangi kompleksitas komputasi tanpa mengurangi akurasi pelacakan. Selain itu, ASAFormer menggunakan Dynamic Masked Attention (DMA) untuk menyesuaikan interaksi antara template objek dan cabang pencarian secara adaptif. Hal ini memungkinkan model untuk lebih fokus pada bagian-bagian penting dari citra, meningkatkan ketepatan pelacakan, dan mengurangi gangguan dari latar belakang yang tidak relevan. Dalam penelitian ini, ASAFormer diuji pada beberapa dataset benchmark pelacakan objek, termasuk TrackingNet dan UAV123. Hasil eksperimen menunjukkan bahwa ASAFormer memberikan kinerja pelacakan yang superior dibandingkan dengan metode-metode pelacakan visual lainnya, dengan akurasi yang lebih tinggi dan efisiensi komputasi yang lebih baik. Model ini berhasil menangani tantangan-tantangan seperti occlusion, perubahan skala, dan variasi bentuk objek dengan lebih efektif. Metode utama yang digunakan dalam ASAFormer adalah Asymmetric Selective Attention (ASA), yang memanfaatkan teknik pemilihan token representatif untuk mengurangi redundansi dalam operasi perhatian. Dengan mengurangi jumlah token yang terlibat dalam perhatian, model ini berhasil mengurangi beban komputasi. Selain itu, Dynamic Masked Attention (DMA) digunakan untuk menyesuaikan interaksi antara template dan cabang pencarian, memungkinkan model untuk berfokus pada area yang paling relevan dalam citra atau video yang meningkatkan akurasi pelacakan (Gong dkk., 2024).

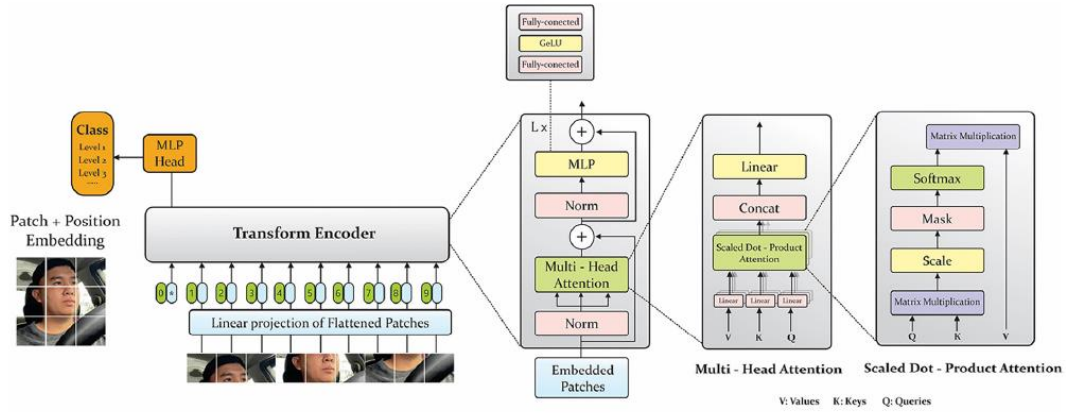
HuntFormer suatu metode pelacakan objek visual yang terinspirasi dari prinsip berburu alami, dengan menggunakan dua modul utama: Predictor dan Verifier. Predictor menggunakan particle filter untuk memprediksi lokasi objek dengan mengandalkan informasi pergerakan dan korelasi target, memungkinkan model untuk fokus pada objek yang terdeteksi dan mengurangi ruang pencarian. Verifier, di sisi lain, berfungsi untuk menilai keandalan hasil

pelacakan dengan menggunakan model ketidakpastian untuk memutuskan apakah template objek perlu diperbarui berdasarkan tingkat ketidakpastian yang terukur. Metode ini mengatasi masalah drift pelacakan dengan lebih efisien, memungkinkan HuntFormer untuk mendeteksi dan memperbaiki pelacakan yang hilang dengan cepat. Penelitian ini menunjukkan bahwa HuntFormer memiliki kinerja yang lebih unggul dibandingkan dengan model-model pelacakan lainnya pada dataset benchmark yang digunakan, termasuk pada kondisi pelacakan yang kompleks seperti occlusion dan objek yang keluar dari frame. Dalam eksperimen, HuntFormer berhasil mendeteksi kembali objek yang hilang lebih cepat dan lebih akurat daripada model pelacakan lainnya, yang menunjukkan keunggulannya dalam menghadapi perubahan besar dalam posisi objek. HuntFormer menggunakan metode berbasis particle filter untuk memperkirakan posisi objek dengan lebih baik, memanfaatkan motion trajectory sebagai pedoman untuk mendeteksi perubahan posisi objek. Verifier yang mengukur ketidakpastian dalam proses pelacakan membantu memperbarui template objek hanya jika diperlukan, meningkatkan efisiensi tanpa membebani komputasi. Metode ini menunjukkan bahwa menggabungkan prediksi berbasis filter dan pengecekan ketidakpastian dapat meningkatkan keandalan dan kecepatan dalam pelacakan objek (Z. Zhang dkk., 2024).

IAC-tracker merupakan pelacak objek visual berbasis Transformer yang berfokus pada penanganan perubahan penampilan objek secara langsung (*immediate appearance change*) menggunakan dua komponen utama: *Spatial Information Extractor* (SIE) dan *Temporal Information Extractor* (TIE). SIE bertanggung jawab untuk mengekstrak informasi spasial dari template objek dan wilayah pencarian, sementara TIE memproses informasi temporal dengan memperhatikan perubahan penampilan objek dalam urutan video. Gabungan dari kedua komponen ini memungkinkan model untuk lebih baik menangani perubahan skala dan deformasi objek, yang sering menjadi tantangan dalam pelacakan objek visual. Eksperimen yang dilakukan pada beberapa dataset benchmark, termasuk OTB100, menunjukkan bahwa IAC-tracker

mengungguli model pelacakan lainnya dalam hal *Area Under Curve* (AUC) dan mean Average Precision (mAP). Hasil ini menunjukkan bahwa IAC-tracker mampu menangani berbagai skenario pelacakan yang melibatkan objek dengan perubahan bentuk yang signifikan dan kondisi video yang kompleks seperti motion blur, occlusion, dan fast motion. Kelebihan lain dari IAC-tracker adalah kemampuannya untuk berjalan secara real-time, menjadikannya solusi yang efisien untuk aplikasi yang membutuhkan kecepatan dan akurasi. IAC-tracker menggunakan Transformer sebagai dasar arsitekturnya, yang memberikan kemampuan untuk menangkap ketergantungan jangka panjang antara objek dan latar belakang. Model ini menggabungkan SIE untuk ekstraksi fitur spasial dan TIE untuk menangkap dinamika temporal dalam video. Teknik spatial-temporal fusion digunakan untuk mengintegrasikan kedua informasi ini secara bersamaan, memungkinkan model untuk memahami dan melacak objek dengan lebih baik meskipun ada perubahan dalam penampilannya di berbagai frame video (Y. Li dkk., 2024).

Vision Transformer (ViT) diperkenalkan oleh sekelompok peneliti (Dosovitskiy dkk., 2020). Arsitektur ini merupakan jaringan saraf intensif yang dirancang untuk tugas pengenalan citra, dan dibangun berdasarkan arsitektur Transformer (Vaswani dkk., 2017). Gagasan utamanya adalah memperlakukan potongan-potongan citra (*image patches*) sebagai urutan token yang dapat digunakan sebagai masukan ke dalam model Transformer. Arsitektur Transformer telah berhasil diadaptasi dan diterapkan secara efektif dalam berbagai tugas, seperti restorasi citra dan deteksi objek (Y. Liu dkk., 2019). Dalam Vision Transformer, sebuah citra dibagi menjadi grid patch yang tidak saling tumpang tindih, di mana setiap patch kemudian diproyeksikan secara linear. Selanjutnya, urutan hasil embedding linear dari patch tersebut diberikan sebagai input ke encoder Transformer standar. Proses ini memungkinkan model untuk mempelajari informasi global maupun lokal dari suatu *image*.



Gambar 2.2 Vision Transformer Model (T.-C. Phan dkk., 2025a)

Gambar 2.2 menunjukkan gambaran umum dari model ViT, yang terdiri atas lapisan embedding, lapisan encoder, dan lapisan klasifikasi. Langkah pertama adalah membagi citra $\mathcal{X} \in \mathbb{R}^{CxHxW}$ dari dataset pelatihan menjadi patch $x_p \in \mathbb{R}^{Nx(CxPxp)}$, di mana $(H\ x\ W)$ adalah resolusi dari citra x, c adalah jumlah saluran channel, N adalah jumlah total patch, dan $(P\ x\ P)$ (Ayana & Choe, 2022) adalah resolusi dari tiap patch χp. Ukuran patch p umumnya ditetapkan sebesar 32 × 32 atau 16 × 16. Semakin kecil ukuran patch, maka performa model cenderung semakin baik, namun juga meningkatkan beban komputasi. Oleh karena itu, ukuran patch 16 × 16 dipilih karena dianggap mampu memberikan performa yang baik tanpa meningkatkan kompleksitas komputasi secara signifikan.

Transformer melakukan setiap patch sebagai token individu dan memetakannya ke dalam dimensi D menggunakan proyeksi linear E yang dapat dilatih. Proyeksi embedding ini dikaitkan dengan token kelas (*class token*) X_{class} yang dapat dilatih, yang berperan penting dalam menyelesaikan proses klasifikasi, serupa dengan yang dilakukan pada model BERT. Untuk melacak informasi lokasi, digunakan embedding posisi E_{pos} guna mempertahankan susunan spasial patch sesuai dengan posisi aslinya dalam citra. Hubungan terenkripsi dari patch ke token awal Z_0 direpresentasikan dalam suatu persamaan (Eq).

$$Z_0 = \left[x_{class}; X_p^1\ E; \ldots; X_p^N E\right] + Epos \qquad (2.1)$$

dimana :

Z_0 = Matriks input pertama (initial embedding)

Epos = Position Embedding

Vision Transformer terdiri atas *L* blok pengkode (*encoding blocks*) dan setiap blok dapat dibagi menjadi dua sub-komponen yang terdiri dari blok multi-head self-attention (MHA) dan blok multilayer perceptron (MLP). Blok MLP bertugas untuk mengalirkan nilai z_0 ke dalam pengkode transformer. Di dalam encoder MHA merupakan komponen utama dari transformer yang mencakup lapisan self-attention dan lapisan *concatenation* (penggabungan).

Self-attention nilai masukan $X = x_1, x^2, \ldots xn$ transformer melakukan operasi atensi secara simultan pada himpunan query set (Q) terhadap seluruh key (K) dan value (V) seperti ditunjukkan pada Persamaan (2.2). Matriks bobot W^Q,W^K,W^V adalah matriks bobot yang dapat dipelajari, yang didasarkan pada perhitungan hasil dot product antara query dan seluruh key, kemudian diskalakan dengan akar kuadrat dari dimensi D dan selanjutnya diterapkan fungsi softmax.

$$Atten(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{D}}\right)V \qquad (2.2)$$

Dimana :

Atten = *Attention*

Q (Query) = Matriks yang mewakili model pada setiap posisi dalam video

K (Key) = Matriks yang berisi indeks atau informasi kunci dari setiap posisi video untuk dicocokkan dengan *Query*

V (Value) = Matriks yang berisi informasi fitur yang sebenarnya

Softmax = Fungsi aktivasi yang mengubah skor kecocokan antara Q dan K menjadi bobot

Lapisan concatenation merupakan kerangka kerja transformer menjalankan mekanisme Multi-Head melalui scaled dot-product attention secara berulang kali dengan bobot pembelajaran yang berbeda-beda. Selanjutnya, semua kepala atensi (*attention heads*) digabungkan untuk

menghasilkan keluaran akhir, seperti ditunjukkan pada Persamaan (2.3), di mana W_i^Q, W_i^K, W_i^V, W^o adalah matriks parameter. Keluaran akhir pada kelas ι dari blok MHA ditentukan oleh Persamaan (2.4).

$$MHA(Q,K,V) = Concat(Atten1, Atten\ 2, \dots, Atten\ h)W^0 \quad (2.3)$$

dimana :

MHA = *Multi-Head Attention*

h = jumlah *head* atau kepala perhatian dalam model

W^0 = Matriks bobot proyeksi akhir

$$Z_l^{\backprime} = MHA\ (LN_{(z\iota-1)} + Z_\iota - 1, dimana\ l\ = 1, \dots, L \quad (2.4)$$

dimana :

Z_l^1 = Hasil keluaran (*output*)

$LN_{(z\iota-1)}$ = Layer Normalization

$Z_\iota - 1$ = Input dari lapisan sebelumnya

$L =$ Indeks Lapisan

MLP terdiri atas dua lapisan fully connected dan satu aktivasi GeLU di tengahnya, dengan keluaran pada kelas *l* didefinisikan dalam Persamaan (2.5). Elemen pertama dari Z_l^0 diambil dan dikirimkan ke *head classifier* untuk memprediksi kelas akhir dari encoder terhadap label sebagaimana didefinisikan dalam Persamaan (2.6).

$$z_1 = MLP\ (LN_{(z\iota-1)} + Z_l^{\backprime} - 1\ , dimanal = 1, \dots, L \quad (2.5)$$

dimana :

MLP = Multilayer Perceptron

z_1 = Representasi fitur akhir dari lapisan ke-l

$$y = LN(Z_l^0) \quad (2.6)$$

dimana :

y = Vektor fitur final

LN = Layer Normalization

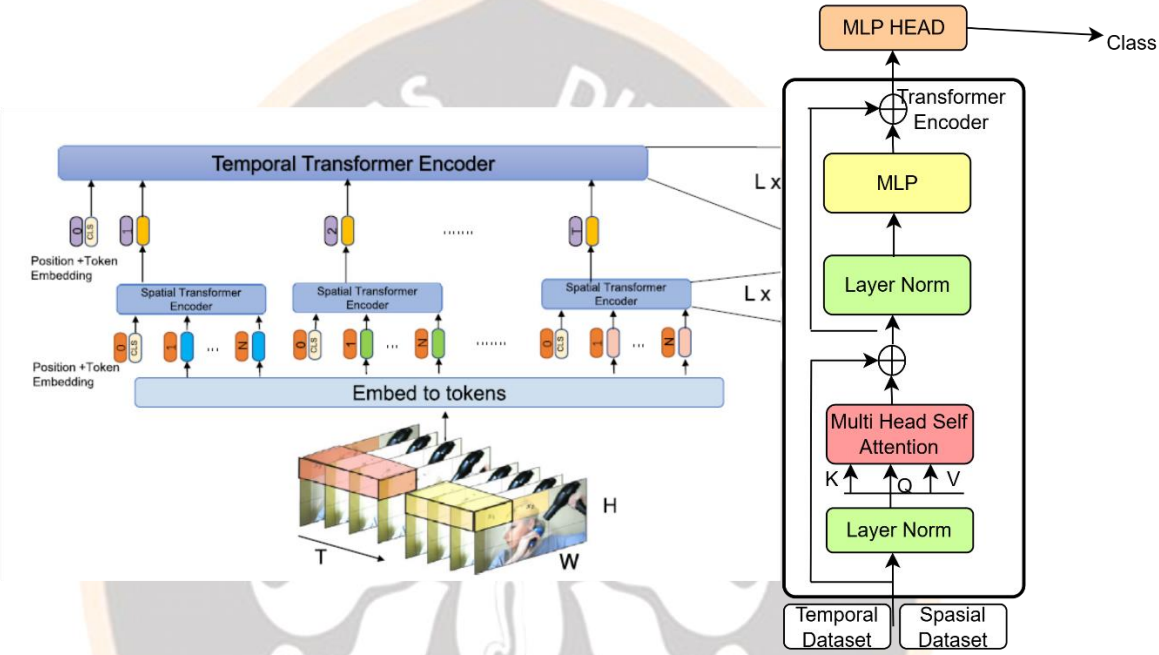
Z_l^0 = token pertama indeks ke-0

2.3.9 Video Vision Transformer (ViViT)

Sebagai salah satu terobosan dalam bidang pengenalan citra, Vision Transformer (ViT) telah memperkenalkan pendekatan baru yang mengadaptasi mekanisme perhatian (*attention mechanism*) dari arsitektur Transformer yang awalnya dikembangkan untuk pemrosesan bahasa alami ke dalam domain pengolahan citra. Tidak seperti Convolutional Neural network (CNN) yang mengandalkan operasi konvolusi lokal, Vision Transformer (ViT) membagi citra menjadi patch-patch kecil yang diperlakukan sebagai token, mirip seperti kata dalam kalimat, dan kemudian memprosesnya secara global menggunakan self-attention. Pendekatan ini memungkinkan model untuk menangkap hubungan spasial yang lebih luas dan kompleks antarpixel tanpa tergantung pada struktur lokal. Keunggulan ViT dalam memahami representasi visual inilah yang menjadi dasar bagi berbagai pengembangan arsitektur berbasis transformer, termasuk Video Vision Transformer (ViViT), yang diperluas untuk menangani data video yang mengandung informasi spasial dan temporal secara simultan.

Video Vision Transformer (ViViT) adalah sebuah arsitektur deep learning berbasis Transformer yang dirancang untuk memahami informasi spasial dan temporal dalam data video secara simultan. Tidak seperti Vision Transformer (ViT) yang hanya menangani data citra statis, ViViT memperluas prinsip Transformer ke domain video dengan membagi video menjadi potongan tiga dimensi yang disebut *tubelets*, yaitu patch citra yang mencakup dimensi ruang dan waktu. Setiap tubelet diubah menjadi representasi vektor dan diproses melalui rangkaian blok Transformer, yang mengandalkan mekanisme multi-head self-attention untuk menangkap hubungan antar patch secara fleksibel, tanpa ketergantungan pada operasi konvolusi atau jaringan berulang. ViViT mendukung berbagai pendekatan arsitektur, seperti *joint spatio-temporal attention* maupun *factorized attention* (pemrosesan spasial dan temporal dilakukan secara terpisah), yang masing-masing memiliki keunggulan dalam hal akurasi dan efisiensi komputasi. Berkat desainnya yang

modular dan skalabel, ViViT terbukti unggul dalam berbagai tugas pemahaman video, termasuk klasifikasi aksi, deteksi aktivitas, dan aplikasi real-time lainnya seperti deteksi kantuk berbasis video (Arnab dkk., 2021). Berikut merupakan gambaran ViViT .



Gambar 2.3 Arsitektur Video Vision Transfomer (Sun dkk., 2024)

Berbeda dengan ViT pada ViViT terjadi dari penyisipan klip video. Perluasan penyisipan ViT ke dalam video adalah ViViT, yang mengekstrak "tubelet" spasio-temporal yang tidak tumpang tindih dari video input dan melakukan proyeksi linear. Dibandingkan dengan frame n_t sampel seragam dari video, dimensi penyisipan tubelet

$$t\ x\ h\ x\ w,\ n_t = \left[\frac{T}{t}\right],\ n_h = \left[\frac{H}{h}\right],\ n_w = \left[\frac{W}{w}\right] \qquad (2.7)$$

Token-token yang diperoleh dari dimensi temporal, tinggi, dan lebar diproses secara terpisah. Kemudian, total token sebanyak $nt \cdot nh \cdot nw$ akan diteruskan ke dalam encoder Transformer. Dengan kata lain, penyematan tubelet dapat dianggap sebagai membangun blok 3D, informasi spatio-temporal digabungkan selama tokenisasi, seperti yang ditunjukkan di sisi kiri Gambar 2.3. *Multi-Headed Self-Attention* (MSA) dimasukkan dalam penyematan yang telah diubah pada model ViT dan ViViT setelah blok layer

normalization (LN). Blok multilayer perceptron (MLP) juga diperlukan dalam encoder Transformer setelah blok LN dan terdiri dari dua proyeksi linear yang dipisahkan oleh non-linearitas GELU.

Metode Video Vision Transformer (ViViT) yang digunakan dalam penelitian ini merupakan arsitektur yang berdiri di atas fondasi bertingkat, mulai dari konsep Deep Learning, arsitektur Transformer, hingga Vision Transformer (ViT). Sebagai bagian dari rumpun Deep Learning, ViViT memanfaatkan jaringan saraf tiruan yang dalam untuk mengekstraksi fitur kompleks dari data visual tanpa intervensi manual. Arsitektur ini mengadopsi mekanisme *self-attention* dari model Transformer, yang memungkinkan sistem memberikan bobot perhatian berbeda pada setiap bagian informasi. ViViT merupakan pengembangan langsung dari Vision Transformer (ViT), jika ViT mengadaptasi Transformer untuk pemrosesan citra spasial, maka ViViT memperluas kapabilitas tersebut ke dalam dimensi temporal. Dengan demikian, ViViT mengintegrasikan kekuatan ketiga unsur tersebut untuk menangkap hubungan spasio-temporal dalam video, sehingga perubahan kecil pada kondisi mata dan gerakan kepala pengemudi dapat dideteksi secara lebih akurat sebagai representasi dari kondisi kantuk.

2.3.10 Evaluasi Performa Model Deteksi Kantuk

Penelitian ini bertujuan untuk mengevaluasi performa model deteksi kantuk pada pengemudi dengan menggunakan pendekatan Brain-Computer Interface (BCI) yang berbasis sinyal EEG (*Electroencephalography*). Para penulis mengusulkan sistem klasifikasi kantuk yang tidak bergantung pada pengolahan citra wajah, melainkan menggunakan fitur neurofisiologis yang ditangkap langsung dari aktivitas otak. Pendekatan ini sangat relevan dalam konteks pengembangan sistem deteksi kantuk yang lebih objektif dan tidak terpengaruh oleh pencahayaan atau ekspresi wajah. Dataset yang digunakan berasal dari eksperimen mengemudi simulatif di mana peserta menjalani sesi panjang mengemudi monoton sambil dipantau oleh perangkat EEG. Setiap sesi diberi label berdasarkan kondisi sadar (alert) dan mengantuk (drowsy), yang dikonfirmasi secara subjektif maupun objektif melalui pengamatan gelombang

otak (terutama alpha dan theta) dan validasi perilaku. Hasil penelitian menunjukkan bahwa fitur-fitur EEG tertentu, seperti band daya pada frekuensi alpha dan theta, sangat relevan untuk deteksi kondisi kantuk. Model klasifikasi terbaik yang dihasilkan dalam studi ini adalah Random Forest, yang mampu mencapai F1-score hingga 79%, sementara SVM juga menunjukkan performa yang kompetitif. Selain itu, kombinasi kanal EEG tertentu (seperti O1 dan Fp2) menghasilkan akurasi yang lebih tinggi dibandingkan penggunaan semua kanal secara serempak, yang menunjukkan pentingnya pemilihan fitur kanal yang selektif(Hidalgo Rogel dkk., 2024) .

Beberapa penelitian menggabungkan pendekatan berbasis pengenalan citra wajah dengan algoritma machine learning untuk mendeteksi tanda-tanda kantuk seperti seringnya mata tertutup, frekuensi kedipan yang tinggi, serta ekspresi wajah yang menunjukkan kelelahan. Penelitian ini mengimplementasikan beberapa teknik klasifikasi seperti K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), dan algoritma deep learning seperti Convolutional Neural Network (CNN) dan YOLOv5. Sebagai perbandingan, model juga diuji menggunakan Faster R-CNN dan YOLOv8 untuk mendeteksi wajah dan bagian mata secara presisi. Data yang digunakan dalam studi ini berasal dari tiga dataset publik yang sangat dikenal dalam penelitian deteksi kantuk, yaitu NTHU-DDD (National Tsing Hua University Drowsy Driver Detection), YawDD (Yawning Detection Dataset), dan UTA-RLDD (University of Texas at Arlington Real-Life Drowsiness Dataset). Ketiga dataset ini mencakup berbagai ekspresi wajah, kondisi pencahayaan, dan variasi gerakan kepala yang meniru kondisi nyata dalam berkendara, sehingga memungkinkan pengujian sistem secara lebih komprehensif. Hasil penelitian menunjukkan bahwa pendekatan berbasis deep learning, khususnya kombinasi CNN dan YOLOv5, mampu memberikan performa terbaik dalam mendeteksi kantuk dengan akurasi mencapai lebih dari 98% dalam pengujian pada dataset NTHU-DDD. YOLOv5 terbukti unggul dalam mendeteksi fitur wajah secara cepat dan akurat dalam skenario real-time. Selain itu, integrasi sistem klasifikasi berdasarkan pola mata dan kepala

menghasilkan peningkatan signifikan pada sensitivitas sistem dalam mengenali fase awal kantuk (Essahraui dkk., 2025).

Penelitian ini membahas pengembangan serta evaluasi performa dan penerimaan pengguna terhadap sistem deteksi kantuk berbasis wearable device, yang dirancang untuk memberikan solusi praktis dan nyaman bagi pengemudi dalam mendeteksi kelelahan selama berkendara. Sistem ini menggunakan perangkat pintar yang dikenakan di tubuh untuk memantau indikator fisiologis yang berkaitan dengan kondisi kantuk, seperti denyut jantung dan tingkat aktivitas tubuh. Fokus penelitian tidak hanya pada akurasi teknis sistem, tetapi juga pada bagaimana sistem ini diterima dan dirasakan oleh penggunanya dalam situasi nyata. Metode penelitian yang diterapkan mencakup dua tahap utama, yaitu pengujian teknis terhadap performa sistem dan survei evaluasi terhadap persepsi pengguna. Sinyal denyut jantung dan akselerasi gerakan digunakan sebagai parameter utama dalam mendeteksi peralihan dari kondisi waspada ke kondisi mengantuk. Data yang digunakan dalam penelitian ini dikumpulkan dari sejumlah peserta yang diminta menjalani sesi mengemudi panjang di lingkungan yang terkendali, baik dalam skenario mengemudi nyata maupun menggunakan simulator. Hasil penelitian menunjukkan bahwa sistem deteksi kantuk berbasis wearable ini mampu mencapai tingkat akurasi yang cukup tinggi dalam mengidentifikasi kondisi kantuk berdasarkan kombinasi sinyal denyut jantung dan gerakan tubuh. Dalam uji coba menggunakan 30 peserta dengan rentang usia 20–25 tahun dan 65–70 tahun, sistem deteksi kondisi kantuk berbasis deteksi denyut jantung (heart rate) berhasil mencapai akurasi keseluruhan 82,72% (Kundinger dkk., 2021).

Arsitektur Transformer yang diusulkan dalam FAViT (Feature-level Augmentation Vision Transformer) dirancang untuk mengatasi tantangan dalam tugas *video-based person re-identification* (ReID), khususnya dalam konteks efisiensi komputasi dan ketahanan terhadap gangguan visual. Penelitian ini bertujuan membangun model Vision Transformer yang mampu secara efektif mengekstraksi dan menggabungkan informasi spasial serta

temporal dari video, sembari menjaga kompleksitas model tetap rendah dibandingkan arsitektur Transformer konvensional yang cenderung sangat besar dan mahal secara komputasi. Dibandingkan dengan Vision Transformer konvensional, FAViT memiliki sejumlah keunggulan penting. Pertama, input token pada FAViT mencakup token latar belakang tambahan, yang tidak tersedia dalam ViT klasik. Kedua, mekanisme perhatian dalam FAViT dibedakan berdasarkan jenis token, sehingga model dapat memfokuskan perhatian hanya pada bagian-bagian penting dari video. Ketiga, augmentasi dilakukan pada tingkat fitur (bukan gambar), yang memungkinkan pelatihan model menjadi lebih fleksibel dan efisien. Selain itu, FAViT memperkenalkan dua sub-tugas pelatihan tambahan yang secara signifikan meningkatkan kemampuan generalisasi model. Yang terpenting, FAViT mampu mencapai performa tinggi dengan jumlah parameter yang jauh lebih rendah (~78 juta), sekitar 65% lebih ringan dibandingkan model-model Transformer sebelumnya(Kim dkk., 2025) .

Model Transformer berbasis perhatian berbobot tetangga terdekat (k-NN) yang disebut k-ViViT (k-NN Video Vision Transformer) untuk meningkatkan akurasi dan efisiensi dalam tugas pengenalan aksi pada video. Berangkat dari kelemahan model Vision Transformer (ViT) dan turunannya, seperti ViViT, yang menggunakan mekanisme self-attention sepenuhnya terhubung (fully-connected), model ini mencoba mengurangi kompleksitas komputasi dan dampak dari token yang tidak relevan atau mengandung noise. Untuk itu, penulis mengintegrasikan mekanisme k-NN attention, yang hanya memilih top-k token paling relevan, dalam arsitektur Transformer. Pendekatan ini memungkinkan proses pelatihan yang lebih cepat serta meningkatkan akurasi model dengan meminimalkan overfitting terhadap informasi tidak penting dalam frame video. Arsitektur yang dikembangkan terbagi menjadi dua varian: Uk-ViViT (Undecomposed) dan Dk-ViViT (Decomposed). Model Uk-ViViT mempertahankan arsitektur ViViT konvensional dengan integrasi k-NN attention secara langsung pada keseluruhan token spatio-temporal. Sementara itu, model Dk-ViViT memisahkan pemrosesan spasial dan temporal ke dalam

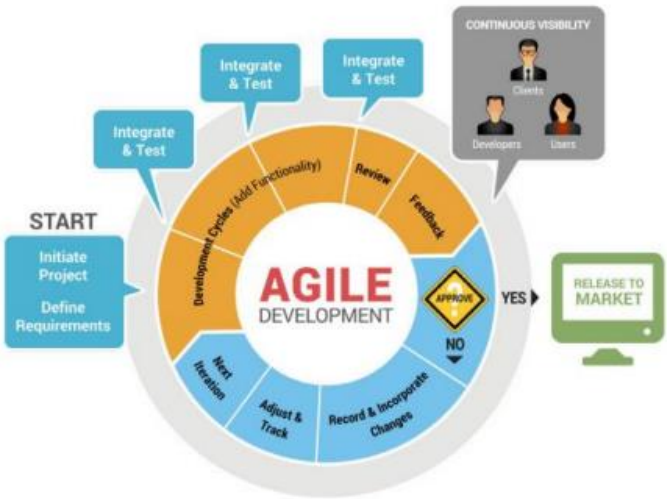
dua encoder Transformer yang berbeda, masing-masing dilengkapi dengan k-NN attention. Pemisahan ini memungkinkan interaksi antar-token yang lebih fokus dan mengurangi pencampuran informasi spasial dan temporal yang tidak berkaitan. Eksperimen dilakukan pada dua benchmark standar, yaitu UCF101 dan HMDB51, dengan menunjukkan peningkatan akurasi yang signifikan. Model Dk-ViViT mencatat Top-1 accuracy sebesar 94.2% pada UCF101 dan 82.5% pada HMDB51, mengungguli varian ViViT orisinal maupun model Uk-ViViT. Hasil ini menunjukkan peningkatan akurasi sebesar 7.5% dan 15.4% dibanding ViViT pada kedua dataset tersebut. Lebih lanjut, studi ablation menunjukkan bahwa penerapan k-NN attention secara terpisah pada domain spasial dan temporal masing-masing juga meningkatkan kinerja, meskipun implementasi kombinatif pada keduanya (Dk-ViViT) tetap menjadi yang paling unggul(Sun dkk., 2024).

Pendekatan Vision Transformer (ViT) yang diperkuat dengan modul Depth-Wise Convolutions (DWConv) sebagai jalur bypass untuk mengatasi kelemahan ViT dalam menangkap informasi lokal, khususnya pada dataset kecil. DWConv dirancang sebagai shortcut ringan yang memotong seluruh blok Transformer untuk mengembalikan representasi lokal tanpa menambah beban komputasi secara berarti, dan dapat diintegrasikan secara fleksibel tanpa mengubah struktur dasar model. Metode ini terdiri dari dua varian, yaitu bypass beberapa blok Transformer untuk efisiensi dan penggunaan paralel DWConv dengan berbagai kernel untuk menangkap detail lokal yang kaya. Implementasinya pada ViT-Tiny, ViT-Small, CaiT-xxs, dan Swin-Tiny menunjukkan peningkatan akurasi signifikan, seperti pada CIFAR-10 (dari 94.01% ke 96.41%) dan CIFAR-100 (dari 73.68% ke 78.05%), bahkan mengungguli model ViT yang lebih besar dengan parameter dan FLOPs lebih rendah. Selain itu, DWConv juga meningkatkan performa pada Tiny-ImageNet, ImageNet-1K, serta tugas deteksi dan segmentasi objek di COCO, dengan peningkatan skor mAP untuk Mask R-CNN dari 28.2 menjadi 30.6. Metode ini menunjukkan konvergensi pelatihan lebih cepat (~100 epoch dibanding 300 epoch), mampu menangkap objek kecil secara lebih akurat

(terlihat pada visualisasi Grad-CAM), dan menambah beban parameter serta komputasi <0.5%. Dengan demikian, DWConv menjadi solusi yang efisien dan efektif untuk meningkatkan performa ViT, terutama pada skenario pelatihan dari awal di dataset terbatas, tanpa memerlukan modifikasi besar pada arsitektur (T. Zhang dkk., 2025).

2.3.11 Agile Software Development Lifecycle

Pengembangan rekayasa perangkat lunak ditawarkan untuk mengatasi keterbatasan dan mengurangi biaya dengen memberikan fleksibilitas perubahan di setiap tahap (Fitzgerald dkk., 2019). Trend terkini mengadopsi pengembangan cloud computing, big data, dan koordinasi di setiap tahapan. Ada banyak metode dan pendekatan pengembangan perangkat lunak tradisional, seperti pendekatan waterfall, pendekatan iteratif dan inkremental, pendekatan spiral dan pendekatan evolusioner. Metode pengembangan perangkat lunak tradisional membutuhkan proses yang sangat berat. *Agile software development lifecycle* (ASDL) merupakan metode pengembangan perangkat lunak modern, metode pengembangan sistem ini adalah kerangka kerja konseptual untuk rekayasa perangkat lunak yang dimulai dengan fase perencanaan awal dan mengikuti jalan menuju fase pengembangan dengan interaksi berulang dan inkremental sepanjang siklus hidup sistem. Nilai-nilai dan prinsip-prinsip ini memberikan dasar untuk memandu proses pengembangan perangkat lunak dan untuk memberikan karakteristik yang dapat dibedakan untuk metode apa pun dengan fitur kecepatan(Al-Saqqa dkk., 2020). Tahapan agile dapat dilihat pada Gambar 2.4, gambar tersebut menjelaskan setiap tahapan development *life cycle agile*.



Gambar 2.4 Agile Software Development Lifecycle (ASDL) (Fitzgerald dkk., 2019)

Tahapan Agile Development Software :

1. Konsep (*Consept)*

Perencanaan adalah tahap kunci dalam metodologi Agile yang memungkinkan tim untuk memahami tujuan proyek, mengidentifikasi kebutuhan, dan merencanakan langkah-langkah untuk mencapai hasil yang diinginkan. Dalam tahap perencanaan peneliti mengidentifikasi fitur-fitur yang akan dikembangkan dan menentukan prioritasnya. Pada Tahapan ini juga melibatkan pembuatan rencana kerja yang mencakup estimasi waktu dan sumber daya yang diperlukan untuk mengembangkan setiap fitur.

2. Lahir (*Inseption*)

Tahap perancangan dalam metodologi Agile adalah momen ketika peneliti mengembangkan rancangan rinci untuk produk yang akan dikembangkan. Desain ini mencakup aspek visual, antarmuka pengguna, dan struktur keseluruhan produk. Dalam tahap perancangan, peneliti menggabungkan kebutuhan pengguna dengan visi produk untuk menciptakan desain yang baik dan fungsional. Desain ini dapat dibagi menjadi iterasi yang lebih kecil untuk memastikan bahwa desain terus berkembang seiring berjalannya waktu.

3. Pengulangan (*Iteration*)

Pengulangan dilakukan secara teratur dalam siklus Agile dan dapat melibatkan pengujian produk, pemeriksaan kode, atau evaluasi desain. Umpan balik yang diterima selama tahap ini membantu peneliti untuk melakukan perbaikan dan penyesuaian yang diperlukan sebelum melanjutkan ke tahap selanjutnya.

4. Peluncuran (*Release*)

   Tahap peluncuran adalah saat produk akhirnya siap untuk dirilis ke pengguna akhir. Setelah melalui berbagai tahap pengembangan, pengujian, dan perbaikan, produk dianggap telah mencapai kualitas yang memadai untuk digunakan oleh pengguna. Peluncuran produk bisa bersifat perlahan atau sekaligus, tergantung pada preferensi tim pengembang dan jenis produk yang dibangun. Dalam tahap ini, produk telah melewati semua proses pengembangan dan siap untuk memberikan nilai kepada pengguna.

5. Pemeliharaan (*Maintance*)

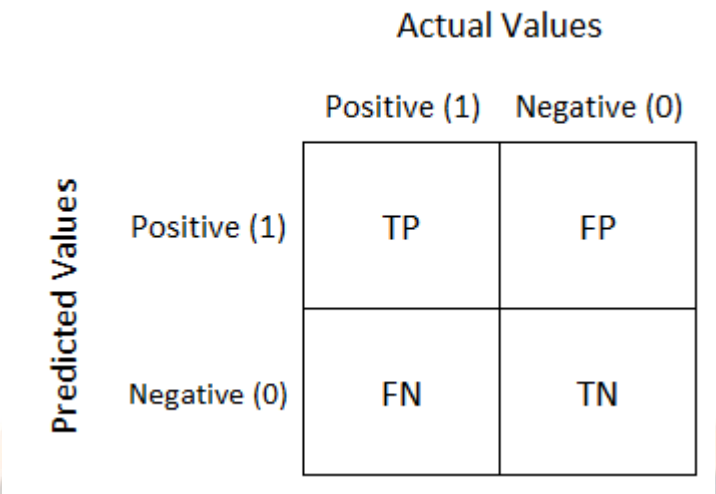
   Setelah produk dirilis dan digunakan oleh pengguna, tahap pemeliharaan dimulai. Tahap ini melibatkan pemantauan produk secara terus-menerus untuk memastikan kinerja yang baik dan mengatasi masalah yang mungkin muncul.

6. Pensiun (*Retired*)

   Tahap pensiun adalah akhir dari siklus hidup produk dalam metodologi Agile. Penting untuk mengelola pensiun produk dengan baik dan memberikan informasi kepada pengguna tentang perubahan yang akan terjadi. Dalam beberapa kasus, produk yang pensiun dapat digantikan dengan produk yang lebih baru dan lebih baik sesuai dengan kebutuhan pengguna dan pasar.

2.3.12 Metode Hasil Evaluasi

Matriks evaluasi diperlukan untuk memastikan bahwa model yang dikembangkan benar-benar mampu menyelesaikan permasalahan dalam mendeteksi tingkat kantuk secara akurat. Gambar 2.4 mengilustrasikan evaluasi model yang disebut dengan *confusion matrix*.



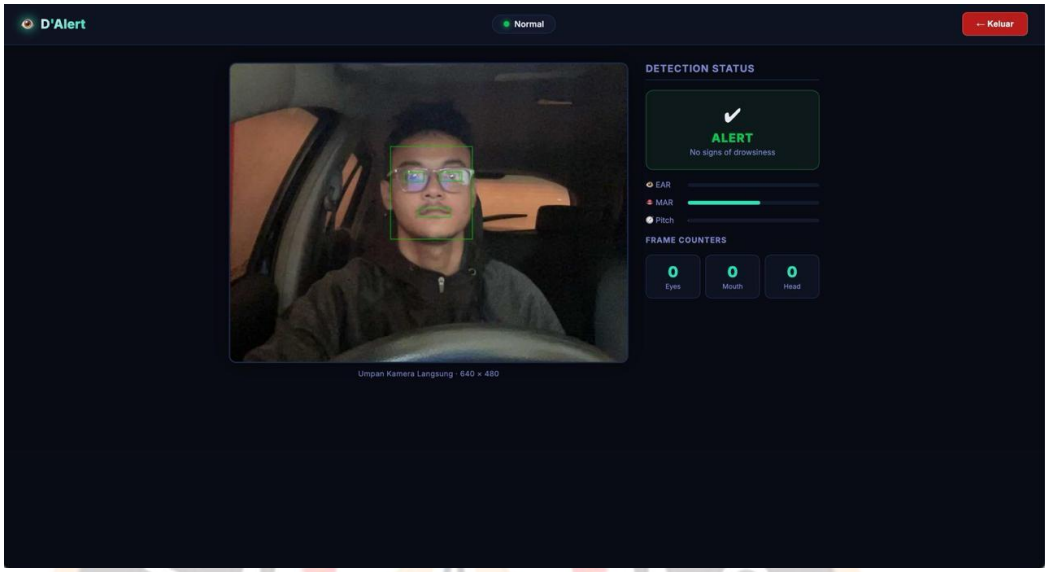
Gambar 2.5 Ilustrasi Confusion Matrix

Keterangan komponen pada *confusion matrix*:

- TP (True Positif): Kondisi kantuk yang berhasil terdeteksi dengan benar sesuai kenyataan.
- FP (False Positif): Model memprediksi kondisi kantuk padahal sebenarnya tidak terjadi.
- FN (False Negatif): Model gagal mendeteksi kondisi kantuk yang sebenarnya ada.
- TN (True Negatif): Model dengan benar memprediksi tidak ada kondisi kantuk sesuai kenyataan.

Dalam penelitian ini, analisis deteksi kantuk dilakukan berdasarkan empat metrik evaluasi utama untuk mengukur kinerja model, yaitu:

Analisis kumpulan data deteksi kantuk didasarkan pada empat metrik evaluasi utama yang digunakan untuk mengukur kinerja model secara menyeluruh. Metrik yang digunakan untuk analisis model deteksi ini meliputi Accuracy, Recall, Precision, dan F1-Score. Sedangkan metrik yang digunakan, yaitu : Inference Time (s)

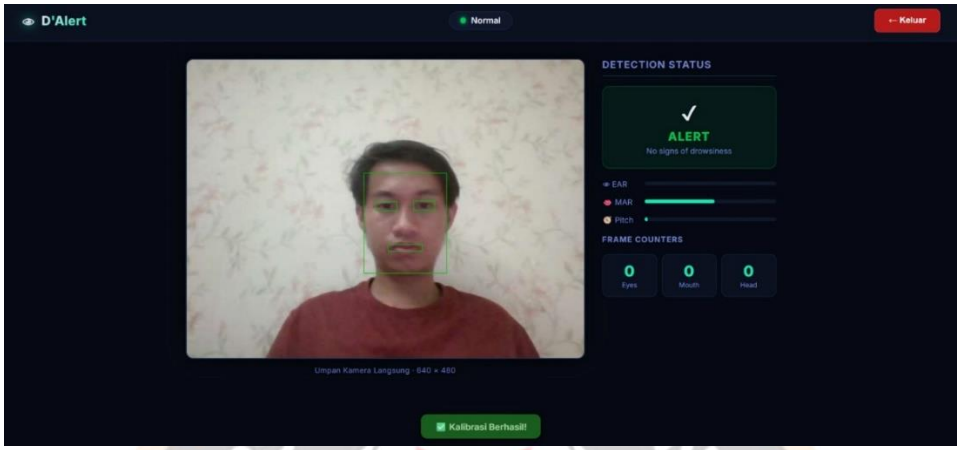


Gambar (b)

Gambar 4.13 Proses Inisiasi Deteksi Kantuk

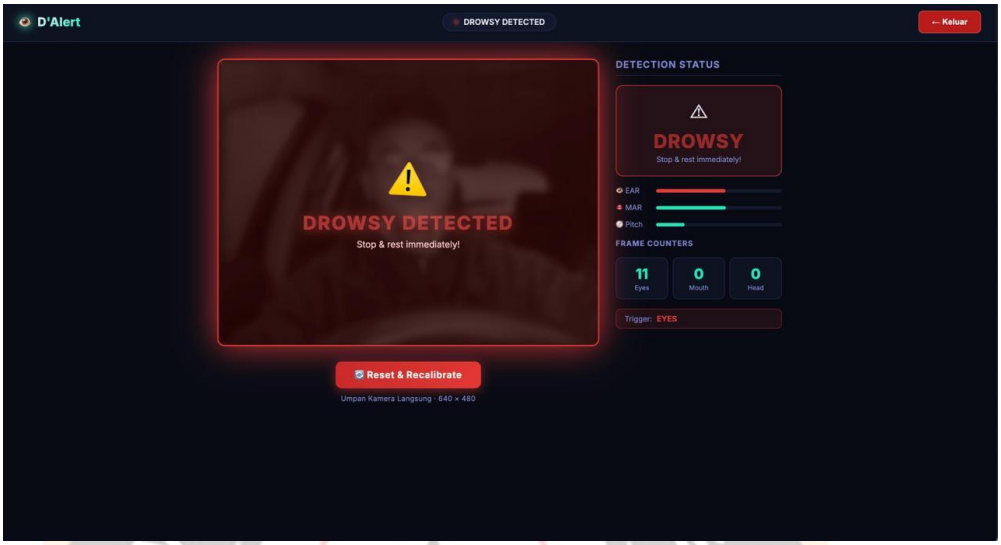
Pada Gambar 4.13 merupakan tampilanpil dasbor pemantauan aktif dari sistem deteksi kantuk bernama D'Alert yang sedang berjalan pada sebuah peramban web. Di sisi kiri antarmuka, terdapat jendela umpan kamera langsung (*Live Feed*) beresolusi 640 x 480 piksel yang menangkap wajah pengguna secara *real-time*. Sistem secara cerdas telah memetakan area wajah menggunakan kotak pembatas (*bounding box*) berwarna hijau, serta melakukan pelacakan titik-titik utama pada mata dan mulut untuk menganalisis aktivitas fisik subjek secara presisi. Gambar 4.13 (a) merupakan kondisi dengan skenario pengujian diruangan lab menggunakan kacamata. Pada Gambar 4.13 (b) merupakan kondisi dengan skenario dalam mobil menggunakan kacmata pada kondisi pencahayaan kurang.

Pada panel bagian kanan, sistem menyajikan informasi deteksi yang sangat detail melalui status utama bertuliskan ALERT dengan latar hijau, yang menandakan bahwa subjek saat ini berada dalam kondisi terjaga tanpa tanda-tanda kelelahan. Di bawah status tersebut, terdapat indikator teknis berupa grafik batang untuk tiga parameter utama: *Eye Aspect Ratio* (EAR) untuk mendeteksi kedipan, *Mouth Aspect Ratio* (MAR) untuk mendeteksi frekuensi menguap, dan Pitch untuk memantau kemiringan kepala. Saat ini, metrik menunjukkan bahwa semua indikator masih berada dalam batas normal.



Gambar 4.14 Proses Deteksi Kantuk Tidak Menggunakan Kacamata

Pada Gambar 4.13, proses inisialisasi kamera dimulai setelah pengguna menekan tombol *Start Detecting*. Wajah pengguna mulai terlihat di area deteksi, sementara status menampilkan keterangan *warming_up* yang berarti sistem sedang mempersiapkan buffer data awal sebelum melakukan analisis. Proses ini menandakan bahwa sistem sudah berhasil menerima input dari kamera namun belum melakukan pendeteksian kantuk. Sesuai dengan yang sudah dijelaskan pada penggunaan kamera dengan spesifikasi Webcam 720 p dengan dimens 1280 x 720. Resolusi ini sudah sangat mencukupi untuk mendeteksi *facial landmarks* (seperti kelopak mata dan kemiringan kepala) tanpa membebani komputasi secara berlebihan. Menggunakan 720p jauh lebih efisien untuk proses *real-time* dibandingkan 1080p, karena jumlah data yang diproses per *frame* lebih sedikit, sehingga *latency* alarm bisa ditekan.



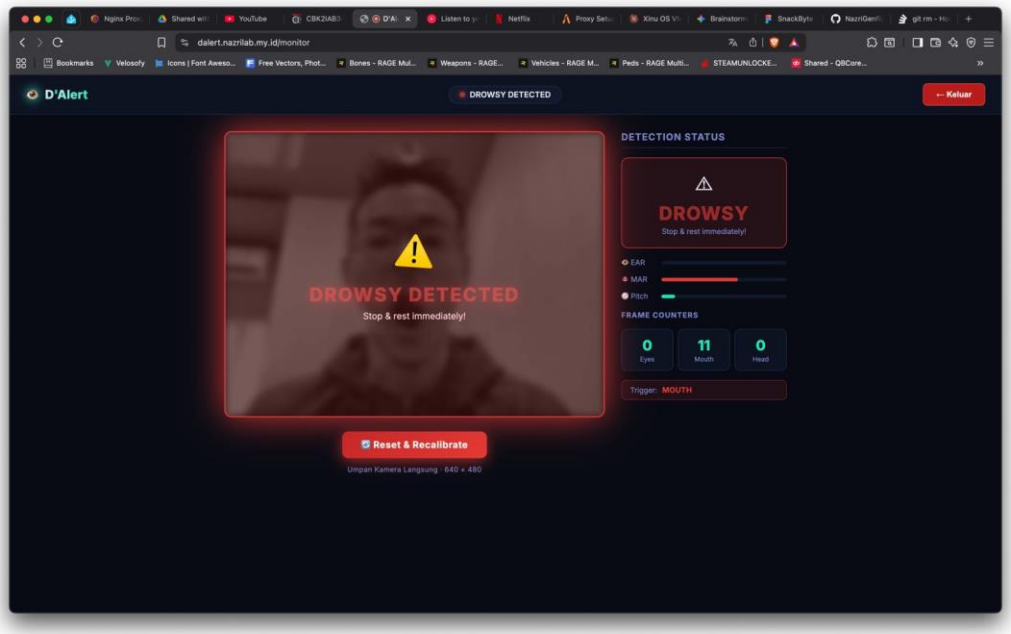
Gambar 4.5 Sistem Mendeteksi Kantuk

Pada Gambar 4.15 menunjukkan kondisi kritis di mana sistem D'Alert berhasil mendeteksi tanda-tanda kantuk yang signifikan pada pengguna. Antarmuka aplikasi berubah secara drastis dengan munculnya hamparan (*overlay*) berwarna merah menyala dan ikon peringatan kuning di tengah layar yang bertuliskan DROWSY DETECTED. Pesan peringatan tersebut instruktif dan tegas, meminta pengguna untuk segera berhenti dan beristirahat (*Stop & rest immediately!*), guna mencegah potensi kecelakaan yang fatal akibat penurunan kesadaran.

Pada panel statistik di sisi kanan, sistem memberikan rincian data yang menjelaskan alasan munculnya peringatan tersebut. Indikator EAR (*Eye Aspect Ratio*) menunjukkan batang berwarna merah yang sangat rendah, menandakan bahwa mata pengguna telah tertutup atau berkedip terlalu lambat dalam durasi yang tidak wajar. Status Trigger: EYES diaktifkan karene terdeteksi dari mata yang memiliki tanda mengantuk. Sementara itu, metrik untuk mulut (*MAR*) dan posisi kepala (*Pitch*) terlihat masih dalam kondisi stabil, menunjukkan bahwa pada momen ini, mata adalah indikator utama kegagalan fokus pengguna.

Sebagai respons terhadap deteksi kantuk ini, sistem menyediakan tombol darurat berwarna merah di bawah jendela kamera yang bertuliskan Reset & Recalibrate. Fitur ini memungkinkan pengguna untuk mengatur ulang sensor setelah beristirahat atau memastikan posisi wajah kembali optimal sebelum melanjutkan perjalanan. Di bagian atas layar, status koneksi juga berubah menjadi

merah dengan teks DROWSY DETECTED, memberikan informasi berlapis yang mudah terlihat. Secara keseluruhan, tampilan ini dirancang untuk memberikan efek urgensi yang tinggi agar pengguna segera mengambil tindakan preventif demi keselamatan.

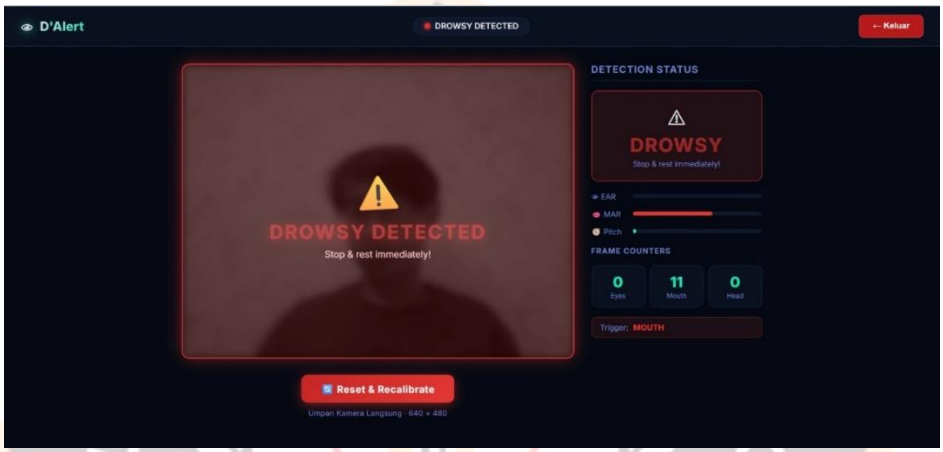


Gambar 4.16 Sistem Mendeteksi Kantuk Kondisi Kacamata

Pada Gambar 4.16 menunjukkan momen ketika sistem berhasil mengidentifikasi tanda kelelahan melalui aktivitas mulut pengguna. Terlihat jelas pada jendela kamera utama bahwa subjek sedang menguap, yang memicu munculnya *overlay* peringatan berwarna merah bertuliskan "DROWSY DETECTED". Peringatan visual ini disertai dengan instruksi tegas agar pengguna segera berhenti dan beristirahat. Perubahan warna antarmuka menjadi serba merah, termasuk pada bilah status di bagian atas, bertujuan untuk memberikan efek kejut dan meningkatkan kewaspadaan pengguna secara instan terhadap kondisi kantuk yang terdeteksi.

Pada panel statistik di sisi kanan, indikator keberhasilan deteksi ini terlihat pada metrik MAR (*Mouth Aspect Ratio*) yang menunjukkan batang berwarna merah memanjang, menandakan bahwa bukaan mulut telah melewati ambang batas normal. Status Trigger: MOUTH diaktifkan karena tanda mengantuk terdeteksi dari mulut yang terbuka (menguap). Berbeda dengan deteksi sebelumnya, pada kondisi

ini indikator mata (*EAR*) dan posisi kepala (*Pitch*) terlihat tetap stabil, yang membuktikan bahwa sistem D'Alert mampu bekerja secara spesifik dalam mengenali berbagai jenis indikator kantuk secara terpisah maupun bersamaan demi keamanan pengguna.



Gambar 4.7 Sistem Mendeteksi Status Drowsy

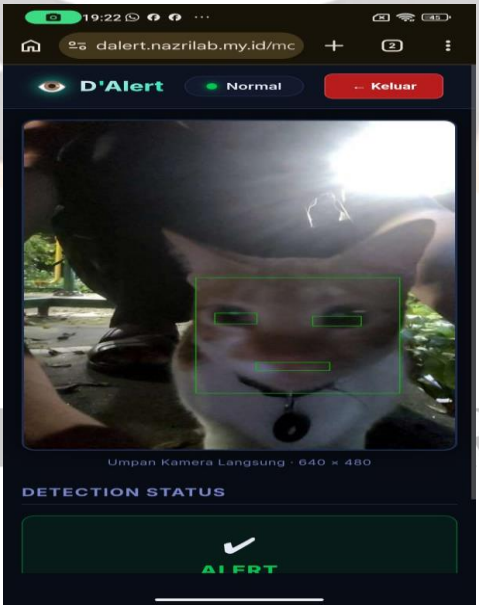
Gambar 4.17 menunjukkan kondisi kantuk Gambar tersebut menunjukkan sistem D'Alert yang sedang berada dalam status waspada tinggi setelah mendeteksi gejala kantuk pada pengguna. Fokus utama pada tangkapan layar ini adalah deteksi perilaku menguap, yang secara otomatis memicu munculnya pesan peringatan berwarna merah bertuliskan DROWSY DETECTED disertai instruksi untuk segera berhenti dan beristirahat. Seluruh bingkai kamera dan elemen antarmuka berubah menjadi warna merah terang sebagai sinyal peringatan visual yang kuat untuk menarik perhatian pengguna agar segera menyadari kondisi kelelahan yang dialami.

Pada bagian panel statistik di sisi kanan, terlihat bahwa pemicu utama peringatan ini adalah nilai MAR (*Mouth Aspect Ratio*) yang melonjak drastis, ditandai dengan batang merah yang panjang. status Trigger: MOUTH diakatifkan karene tanda mengantuk terlihat dari mulut. Sementara itu, indikator mata (*EAR*) dan posisi kepala (*Pitch*) tetap berada pada level aman, menunjukkan bahwa sistem ini sangat sensitif dan spesifik dalam membedakan berbagai jenis tanda fisik kantuk secara akurat untuk meminimalisir risiko kecelakaan.

Pada pengembangan sistem deteksi kantuk pengemudi, variasi persentase ambang batas (*threshold*) yang muncul yaitu 70%, 50%, 86%, dan 96% merupakan

representasi dari skor probabilitas atau tingkat keyakinan model (*confidence score*) dalam mengklasifikasikan kondisi pengemudi secara *real-time*. Fluktuasi angka ini terjadi karena sistem melakukan penyesuaian adaptif terhadap parameter lingkungan yang dinamis, seperti perubahan intensitas cahaya (siang dan malam) serta penggunaan aksesori seperti kacamata yang terdapat pada dataset NTHU-DDD. Angka 50% berfungsi sebagai batas sensitivitas minimum untuk memberikan peringatan dini pada kondisi visual yang sulit, sementara angka yang lebih tinggi hingga 96% merepresentasikan tingkat kepastian absolut model saat mendeteksi indikator kantuk berat seperti *microsleep* atau penutupan mata yang lama. Kedepannya, penentuan ambang batas ini akan distandarisasi melalui analisis kurva akurasi dan *recall* guna menyeimbangkan antara kecepatan respons alarm dan minimalisasi *false alarm* demi menjamin keamanan serta kenyamanan pengemudi secara optimal.

Pada tahap evaluasi model deteksi kantuk berbasis Video Vision Transformer (ViViT), dilakukan uji coba unik untuk mengukur fleksibilitas dan batasan *feature extractor* dalam mengenali fitur wajah secara *real-time*. Pengujian kali ini melibatkan objek hewan, yaitu seekor kucing, untuk melihat apakah parameter deteksi mata dan mulut yang telah dilatih pada dataset manusia tetap aktif atau mengalami redundansi.



Gambar 4. 18 Pengujian Pada Hewan Kucing

Penelitian ini juga melakukan uji coba eksploratif terhadap objek non manusia seperti pada Gambar 4.18, dalam hal ini seekor kucing untuk mengevaluasi fleksibilitas dan kemampuan generalisasi arsitektur *Video Vision Transformer* (ViViT) yang diusulkan. Pengujian ini bertujuan untuk mengukur sejauh mana *feature extractor* pada model mampu mengenali pola morfologi wajah yang berbeda namun memiliki kemiripan struktur secara spasial. Hasil pengujian sistem D'Alert menunjukkan bahwa model mampu melokalisasi *bounding box* pada area wajah serta mengidentifikasi *Region of Interest* (RoI) pada bagian mata dan mulut secara presisi. Model deteksi yang dikembangkan dalam penelitian ini difokuskan untuk mengenali kondisi wajah manusia. Meskipun pada Gambar 4.18 sistem menunjukkan kemampuan untuk melokalisasi fitur wajah dan memberikan status 'ALERT' pada objek non-manusia (kucing), pengujian tersebut tidak dimaksudkan untuk memperluas cakupan fungsionalitas metode pada hewan, namun hal ini digunakan sebagai parameter keberhasilan mekanisme *feature extractor* pada arsitektur *Video Vision Transformer* (ViViT) dalam mengidentifikasi geometri wajah yang kompleks secara presisi. Kemampuan sistem mendeteksi fitur pada objek di luar data utama yaitu Dataset NTHU DDD menjadi validasi algoritma untuk mengenali *Region of Interest* (RoI) mata dan mulut sistem dioptimalkan hanya untuk memvalidasi status *alert* pengemudi manusia.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan analisis yang telah dilakukan terhadap model deteksi kantuk pengemudi berbasis Video Vision Transformer (ViViT), dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Penelitian ini berhasil merancang model deteksi kantuk dengan memanfaatkan arsitektur ViViT yang mampu mengekstraksi fitur spasial (area wajah) dan dinamika temporal (perubahan antar-frame) secara simultan melalui mekanisme *self-attention*. Penggunaan *patch-based tokenization* terbukti efektif dalam menangkap pola perilaku kantuk yang bersifat sekuensial dalam data video.
2. Model berhasil mendeteksi kantuk menggunakan tiga fitur tanda mengantuk yaitu kondisi mata, gerakan mulut menguap dan kemiringan kepala dengan melakukan pendekatan Video Vision Transformer dibuktikan dengan nilai loss sebesar 0,603 yang artinya model tidak mengalami overfitting.
3. Pengujian pada dataset benchmark NTHU DDD membuktikan bahwa model berbasis ViViT ini memiliki tingkat akurasi, presisi, dan *recall* yang baik untuk membedakan kondisi alert dan mengantuk, dibuktikan dengan hasil pengujian *Precision* 94%, *Recall* 90% seta *F1 Score* 93%.
4. Penelitian ini menghasilkan prototype sistem deteksi kantuk non-intrusif dan real-time yang telah berhasil diterapkan dan berpotensi menjadi solusi pendukung keselamatan berkendara.
5. Telah berhasil diimplementasikan sistem deteksi kantuk yang dapat bekerja secara *real-time* dibuktikan dnegan hasil inference time sebesar 28.9ms/frame dan tidak mengganggu kenyamanan fisik pengemudi.

5.2 Saran

Pada hasil evaluasi sistem yang telah dilakukan, penelitian di masa depan disarankan untuk memperbanyak variasi dataset agar model deteksi kantuk memiliki kemampuan generalisasi yang lebih baik di berbagai situasi berkendara.

Sebagai tindak lanjut dari penelitian ini, disarankan agar pengembangan model deteksi kantuk pengemudi berbasis Video Vision Transformer (ViViT) dapat diarahkan pada beberapa aspek penting, seperti :

1. Penelitian lanjutan perlu mengeksplorasi optimasi arsitektur model agar lebih ringan dan efisien, sehingga lebih sesuai untuk implementasi pada perangkat dengan keterbatasan sumber daya, seperti sistem tertanam di kendaraan.
2. Integrasi dengan sensor multimodal, misalnya detak jantung atau deteksi grip setir, dapat dipertimbangkan untuk meningkatkan reliabilitas deteksi.
3. Evaluasi lebih lanjut terkait aspek user experience dan penerimaan pengguna di dunia nyata juga menjadi penting, agar sistem yang dikembangkan tidak hanya akurat secara teknis, tetapi juga praktis dan dapat diterima dengan baik oleh pengemudi.

DAFTAR PUSTAKA

Adi, K., Widodo, C. E., Widodo, A. P., & Aristia, H. N. (2020). Monitoring system of drowsiness and lost focused driver using raspberry pi. *Iranian Journal of Public Health*, *49*(9). https://doi.org/10.18502/ijph.v49i9.4084

Adil Khan, M., Nawaz, T., Khan, U. S., Hamza, A., & Rashid, N. (2023). IoT-Based Non-Intrusive Automated Driver Drowsiness Monitoring Framework for Logistics and Public Transport Applications to Enhance Road Safety. *IEEE Access*, *11*, 14385–14397. https://doi.org/10.1109/ACCESS.2023.3244008

Al-Saqqa, S., Sawalha, S., & Abdelnabi, H. (2020). Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, *14*(11), 246–270. https://doi.org/10.3991/ijim.v14i11.13269

Arif, S., Munawar, S., & Ali, H. (2023). Driving drowsiness detection using spectral signatures of EEG-based neurophysiology. *Frontiers in Physiology*, *14*. https://doi.org/10.3389/fphys.2023.1153268

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A Video Vision Transformer. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV48922.2021.00676

Arslanoglu, M. C., Albayrak, A., & Acar, H. (2025). Vision Transformers Versus Convolutional Neural Networks: Comparing Robustness by Exploiting Varying Local Features. *IEEE Access*, *13*. https://doi.org/10.1109/ACCESS.2025.3559794

Asdyo, B., Kanigoro, B., & Rojali. (2023). Drowsy Detection System by Facial Landmark and Light Gradient Boosting Machine Method. *Procedia Computer Science*, *227*, 500–507. https://doi.org/10.1016/j.procs.2023.10.551

Ayana, G., & Choe, S. W. (2022). BUViTNet: Breast Ultrasound Detection via Vision Transformers. *Diagnostics*, *12*(11). https://doi.org/10.3390/diagnostics12112654

Azim, T., Jaffar, M. A., & Mirza, A. M. (2009). Automatic fatigue detection of drivers through pupil detection and yawning analysis. *2009 4th International Conference on Innovative Computing, Information and Control, ICICIC 2009*. https://doi.org/10.1109/ICICIC.2009.119

Azmi, M. M. B. M., & Zaman, F. H. K. (2024). Driver Drowsiness Detection Using Vision Transformer. *2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 329–336. https://doi.org/10.1109/ISCAIE61308.2024.10576317

Bai, J., Yu, W., Xiao, Z., Havyarimana, V., Regan, A. C., Jiang, H., & Jiao, L. (2022). Two-Stream Spatial-Temporal Graph Convolutional Networks for Driver Drowsiness Detection. *IEEE Transactions on Cybernetics*, *52*(12). https://doi.org/10.1109/TCYB.2021.3110813

Bergasa, Luis Miguel, & dan kawan-kawan. (2006). Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems 7.1*, 63–67.

Bin Mohamad Azmi, M. M., & Kamaru Zaman, F. H. (2024). Driver Drowsiness Detection Using Vision Transformer. *2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 329–336. https://doi.org/10.1109/ISCAIE61308.2024.10576317

Cabot, N., Lamouchi, D., Yaddaden, Y., & Cherif, R. (2024). Drowsiness Detection Through Yawning and Eye Blinking Models Using Convolutional Neural Networks and Transfer Learning. *2024 Sixth International Conference on Intelligent Computing in Data Sciences (ICDS)*, 1–8. https://doi.org/10.1109/ICDS62089.2024.10756434

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). Pre-trained image processing transformer. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR46437.2021.01212

Chen, K., Zhu, T., Li, S., & Shi, Y. (2021). Real-Time Yawning Detection Based on Machine Learning Algorithm and Time Series Classification using Facial Feature Points. *2021 International Conference on High Performance Big Data and Intelligent Systems, HPBD and IS 2021*. https://doi.org/10.1109/HPBDIS53214.2021.9658473

Cvahte Ojsteršek, T., & Topolšek, D. (2019). Influence of drivers' visual and cognitive attention on their perception of changes in the traffic environment. *European Transport Research Review*, *11*(1). https://doi.org/10.1186/s12544-019-0384-2

Dawson, D., Ian Noy, Y., Härmä, M., Kerstedt, T., & Belenky, G. (2011). Modelling fatigue and the use of fatigue models in work settings. Dalam *Accident Analysis and Prevention* (Vol. 43, Nomor 2). https://doi.org/10.1016/j.aap.2009.12.030

Deng, W., & Wu, R. (2019). Real-Time Driver-Drowsiness Detection System Using Facial Features. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2936663

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., &

Houlsby, N. (2020). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. https://doi.org/10.48550/arXiv.2010.11929

Dua, M., Shakshi, Singla, R., Raj, S., & Jangra, A. (2021). Deep CNN models-based ensemble approach to driver drowsiness detection. *Neural Computing and Applications*, *33*(8). https://doi.org/10.1007/s00521-020-05209-7

Eleyan, A., & Demirel, H. (2006). PCA and LDA based face recognition using feedforward neural network classifier. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4105 LNCS*. https://doi.org/10.1007/11848035_28

Essahraui, S., Lamaakal, I., El Hamly, I., Maleh, Y., Ouahbi, I., El Makkaoui, K., Filali Bouami, M., Pławiak, P., Alfarraj, O., & Abd El-Latif, A. A. (2025). Real-Time Driver Drowsiness Detection Using Facial Analysis and Machine Learning Techniques. *Sensors*, *25*(3), 812. https://doi.org/10.3390/s25030812

Fitzgerald, K., Browne, L. M., & Butler, R. F. (2019). Using the Agile software development lifecycle to develop a standalone application for generating colour magnitude diagrams. *Astronomy and Computing*, *28*. https://doi.org/10.1016/j.ascom.2019.05.001

Florez, R., Palomino-Quispe, F., Coaquira-Castillo, R. J., Herrera-Levano, J. C., Paixão, T., & Alvarez, A. B. (2023). A CNN-Based Approach for Driver Drowsiness Detection by Real-Time Eye State Identification. *Applied Sciences (Switzerland)*, *13*(13). https://doi.org/10.3390/app13137849

Gangadharan K, S., & Vinod, A. P. (2022). Drowsiness detection using portable wireless EEG. *Computer Methods and Programs in Biomedicine*, *214*. https://doi.org/10.1016/j.cmpb.2021.106535

Ghoddoosian, R., Galib, M., & Athitsos, V. (2019). *A Realistic Dataset and Baseline Temporal Model for Early Drowsiness Detection*. http://arxiv.org/abs/1904.07312

Gomaa, M., Mahmoud, R., & Sarhan, A. (2022). A CNN-LSTM-based Deep Learning Approach for Driver Drowsiness Prediction. *Journal of Engineering Research*, *0*(0). https://doi.org/10.21608/erjeng.2022.141514.1067

Gong, X., Zhang, Y., & Hu, S. (2024). ASAFormer: Visual tracking with convolutional vision transformer and asymmetric selective attention. *Knowledge-Based Systems*, *291*. https://doi.org/10.1016/j.knosys.2024.111562

Gupta, A., Mazumdar, B. D., Mishra, M., Shinde, P. P., Srivastava, S., & Deepak, A. (2023). Role of cloud computing in management and education. *Materials Today: Proceedings*, *80*. https://doi.org/10.1016/j.matpr.2021.07.370

Hasan, M. M., Watling, C. N., & Larue, G. S. (2022). Physiological signal-based drowsiness detection using machine learning: Singular and hybrid signal approaches. *Journal of Safety Research*, *80*, 215–225. https://doi.org/10.1016/j.jsr.2021.12.001

Heo, B., Yun, S., Han, D., Chun, S., Choe, J., & Oh, S. J. (2021). *Rethinking Spatial Dimensions of Vision Transformers*. https://doi.org/https://doi.org/10.48550/arXiv.2103.16302

Hidalgo Rogel, J. M., Martínez Beltrán, E. T., Quiles Pérez, M., López Bernal, S., Martínez Pérez, G., & Huertas Celdrán, A. (2024). Studying Drowsiness Detection Performance While Driving Through Scalable Machine Learning Models Using Electroencephalography. *Cognitive Computation*, *16*(3). https://doi.org/10.1007/s12559-023-10233-5

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Israr, H., Khan, S. A., Tahir, M. A., Shahzad, M. K., Ahmad, M., & Zain, J. M. (2023). Neural Machine Translation Models with Attention-Based Dropout Layer. *Computers, Materials and Continua*, *75*(2). https://doi.org/10.32604/cmc.2023.035814

Jayasenan J. S, & Smitha, P. S. Mrs. (2014). Driver Drowsiness Detection System. *IOSR journal of VLSI and Signal Processing*, *4*(1), 34–37. https://doi.org/10.9790/4200-04113437

Jiang, X., & Chen, Y. F. (2008). Facial image processing. *Studies in Computational Intelligence*, *91*. https://doi.org/10.1007/978-3-540-76831-9_2

Johnson, A. L., Jung, L., Brown, K. C., Weaver, M. T., & Richards, K. C. (2014). Sleep deprivation and error in nurses who work the night shift. *Journal of Nursing Administration*, *44*(1). https://doi.org/10.1097/NNA.0000000000000016

Kang, N., Han, S., Kim, S., Kwon, S. J., Choi, Y., Lee, Y. T., & Lee, S. I. (2022). Driver Drowsiness Detection based on 3D Convolution Neural Network with Optimized Window Size. *International Conference on ICT Convergence*, *2022-October*. https://doi.org/10.1109/ICTC55196.2022.9952988

Khan, I. (2014). *DRIVER'S FATIGUE DETECTION SYSTEM BASED ON FACIAL FEATURES CORE View metadata, citation and similar papers at core.ac.uk provided by UTHM Institutional Repository*.

Kielty, P., Dilmaghani, M. S., Shariff, W., Ryan, C., Lemley, J., & Corcoran, P. (2023). Neuromorphic Driver Monitoring Systems: A Proof-of-Concept for

Yawn Detection and Seatbelt State Detection Using an Event Camera. *IEEE Access*, *11*. https://doi.org/10.1109/ACCESS.2023.3312190

Kim, M., Cho, M., Lee, H., & Lee, S. (2025). Spatio-temporal Feature-level Augmentation Vision Transformer for video-based person re-identification. *Pattern Recognition*, *168*, 111813. https://doi.org/10.1016/j.patcog.2025.111813

Kitajima, H., Numata, N., Yamamoto, K., & Goi, Y. (1997). Prediction of automobile driver sleepiness (1st report, rating of sleepiness based on facial expression and examination of effective predictor indexes of sleepiness). *Nippon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C*, *63*(613), 3059–3066. https://doi.org/10.1299/kikaic.63.3059

Krishna, G. S., Supriya, K., Vardhan, J., & K, M. R. (2022a). *Vision Transformers and YoloV5 based Driver Drowsiness Detection Framework*. https://doi.org/https://doi.org/10.48550/arXiv.2209.01401

Krishna, G. S., Supriya, K., Vardhan, J., & K, M. R. (2022b). *Vision Transformers and YoloV5 based Driver Drowsiness Detection Framework*. https://doi.org/https://doi.org/10.48550/arXiv.2209.01401

Kundinger, T., Riener, A., & Bhat, R. (2021). Performance and acceptance evaluation of a driver drowsiness detection system based on smart wearables. *Proceedings - 13th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications, AutomotiveUI 2021*. https://doi.org/10.1145/3409118.3475141

Lee, C., & An, J. (2023). LSTM-CNN model of drowsiness detection from multiple consciousness states acquired by EEG. *Expert Systems with Applications*, *213*. https://doi.org/10.1016/j.eswa.2022.119032

Li, R., Hu, M., Gao, R., Wang, L., Suganthan, P. N., & Sourina, O. (2024). TFormer: A time–frequency Transformer with batch normalization for driver fatigue recognition. *Advanced Engineering Informatics*, *62*, 102575. https://doi.org/10.1016/j.aei.2024.102575

Li, Y., Liu, X., Yuan, D., Wang, J., Wu, P., & Liu, J. (2024). A Transformer-based visual object tracker via learning immediate appearance change. *Pattern Recognition*, *155*, 110705. https://doi.org/10.1016/j.patcog.2024.110705

Liu, P., Qian, W., Huang, J., Tu, Y., & Cheung, Y.-M. (2025). Transformer-driven feature fusion network and visual feature coding for multi-label image classification. *Pattern Recognition*, *164*, 111584. https://doi.org/10.1016/j.patcog.2025.111584

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *Proceedings of the IEEE International Conference on Computer Vision*. https://doi.org/10.1109/ICCV48922.2021.00986

López-Tapia, S., Molina, R., & Katsaggelos, A. K. (2021). Deep learning approaches to inverse problems in imaging: Past, present and future. *Digital Signal Processing: A Review Journal*, *119*. https://doi.org/10.1016/j.dsp.2021.103285

Muhammad, H. I., Musa, K. I., Abdulrahman, M. L., Abubakar, A., Umar, K., & Ishola, A. (2021). Enhancing Detection Performance of Face Recognition Algorithm Using PCA-Faster R-CNN. *European Journal of Electrical Engineering and Computer Science*, *5*(3). https://doi.org/10.24018/ejece.2021.5.3.321

Neethirajan, S., & Kemp, B. (2021). Digital Livestock Farming. Dalam *Sensing and Bio-Sensing Research* (Vol. 32). https://doi.org/10.1016/j.sbsr.2021.100408

Nurnaningsih, D., Adi, K., & Surarso, B. (2025). Real-Time Driver Drowsiness Detection Using ViViT for In-Vehicle Monitoring Systems Under Diverse Driving Conditions. *International Journal of Transport Development and Integration*, *9*(4), 688–699. https://doi.org/10.56578/ijtdi090401

Pan, Z., Zhuang, B., He, H., Liu, J., & Cai, J. (2021). *Less is More: Pay Less Attention in Vision Transformers*. https://doi.org/https://doi.org/10.48550/arXiv.2105.14217

Pandey, N. N., & Muppalaneni, N. B. (2021). Real-Time Drowsiness Identification based on Eye State Analysis. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, 1182–1187. https://doi.org/10.1109/ICAIS50930.2021.9395975

Park, S., Kim, B. K., & Dong, S. Y. (2022). Self-Supervised RGB-NIR Fusion Video Vision Transformer Framework for rPPG Estimation. *IEEE Transactions on Instrumentation and Measurement*, *71*. https://doi.org/10.1109/TIM.2022.3217867

Pauly, L., & Sankar, D. (2016). Detection of drowsiness based on HOG features and SVM classifiers. *Proceedings of 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks, ICRCICN 2015*. https://doi.org/10.1109/ICRCICN.2015.7434232

Phan, A. C., Nguyen, N. H. Q., Trieu, T. N., & Phan, T. C. (2021). An efficient approach for detecting driver drowsiness based on deep learning. *Applied Sciences (Switzerland)*, *11*(18). https://doi.org/10.3390/app11188441

Phan, T.-C., Phan, A.-C., & Nguyen, N.-H.-Q. (2025a). A novel approach of drowsiness levels detection using Vis-Net combined with facial emotion. *Systems and Soft Computing*, *7*, 200288. https://doi.org/10.1016/j.sasc.2025.200288

Phan, T.-C., Phan, A.-C., & Nguyen, N.-H.-Q. (2025b). A novel approach of drowsiness levels detection using Vis-Net combined with facial emotion. *Systems and Soft Computing*, *7*, 200288. https://doi.org/10.1016/j.sasc.2025.200288

Prabaswara, S. (2013). *"Studi Kelelahan Dalam Aktivitas Mengemudi Berdurasi Panjang*. Institut Teknologi Bandung. .

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S. Y., & Sainath, T. (2019). Deep Learning for Audio Signal Processing. *IEEE Journal on Selected Topics in Signal Processing*, *13*(2). https://doi.org/10.1109/JSTSP.2019.2908700

Puspasari, M. A., Syaifullah, D. H., Iqbal, B. M., Afranovka, V. A., Madani, S. T., Susetyo, A. K., & Arista, S. A. (2023). Prediction of drowsiness using EEG signals in young Indonesian drivers. *Heliyon*, *9*(9). https://doi.org/10.1016/j.heliyon.2023.e19499

Qianyi Jiang, Q. J., Qianyi Jiang, H. X., & Huahu Xu, C. C. (2023). Two-Stream Spatial–Temporal Transformer Networks For Driver Drowsiness Detection. *電腦學刊*, *34*(5). https://doi.org/10.53106/199115992023103405008

Rimal, R., Herceg, M., Vranjes, M., & Grbic, R. (2023). Driver Monitoring System using an Embedded Computer Platform. *2023 31st International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2023*. https://doi.org/10.23919/SoftCOM58365.2023.10271657

Rodenbeck, A., Binder, R., Geisler, P., Danker-Hopfe, H., Lund, R., Raschke, F., Weeß, H. G., & Schulz, H. (2006). A review of sleep EEG patterns. Part I: A compilation of amended rules for their visual recognition according to Rechtschaffen and Kales. Dalam *Somnologie* (Vol. 10, Nomor 4). https://doi.org/10.1111/j.1439-054X.2006.00101.x

Sadaghiani, S., Brookes, M. J., & Baillet, S. (2022). Connectomics of human electrophysiology. *NeuroImage*, *247*. https://doi.org/10.1016/j.neuroimage.2021.118788

Saravanaraj Sathasivam, & Abd Kadir Mahamad. (2020). 2020 IEEE Student Conference on Research and Development (SCOReD) : Johor, Malaysia, 27-

28 September 2020. *IEEE Malaysia Section, Institute of Electrical and Electronics Engineers*.

Satyanarayana, M. S., Aruna, T. M., & Guruprasad, Y. K. (2021). Continuous monitoring and identification of driver drowsiness alert system. *Global Transitions Proceedings*, *2*(1), 123–127. https://doi.org/10.1016/j.gltp.2021.01.017

Schlett, T., Rathgeb, C., Henniger, O., Galbally, J., Fierrez, J., & Busch, C. (2022). Face Image Quality Assessment: A Literature Survey. *ACM Computing Surveys*, *54*(10). https://doi.org/10.1145/3507901

Shashidhar, V., Ahmed, M., M, R., K S, R., & J, B. (2024). A Novel Approach to Advancing Drowsiness Detection Using Head-Angle and Eye-Aspect Ratio. *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 1–5. https://doi.org/10.1109/CSITSS64042.2024.10816922

Sindhu N R. (2024). *REAL TIME DROWSINESS IDENTIFICATION BASED ON EYE STATE ANALYSIS* (Vol. 10, Nomor 2). www.ijariie.com1144

Singh, P., Mahim, S., & Prakash, S. (2022). Real-Time EAR Based Drowsiness Detection Model for Driver Assistant System. *2022 International Conference on Signal and Information Processing, IConSIP 2022*. https://doi.org/10.1109/ICoNSIP49665.2022.10007514

Slater, J. D. (2008). A definition of drowsiness: One purpose for sleep? *Medical Hypotheses*, *71*(5), 641–644. https://doi.org/10.1016/j.mehy.2008.05.035

Smith, L., Folkard, S., & Poole, C. J. M. (1994). Increased injuries on night shift. *The Lancet*, *344*(8930). https://doi.org/10.1016/S0140-6736(94)90636-X

Stancin, I., Cifrek, M., & Jovic, A. (2021). A review of eeg signal features and their application in driver drowsiness detection systems. Dalam *Sensors* (Vol. 21, Nomor 11). https://doi.org/10.3390/s21113786

Sun, W., Ma, Y., & Wang, R. (2024). k-NN attention-based video vision transformer for action recognition. *Neurocomputing*, *574*, 1–8. https://doi.org/10.1016/j.neucom.2024.127256

Sunagawa, M., Shikii, S. ichi, Beck, A., Kek, K. J., & Yoshioka, M. (2023). Analysis of the effect of thermal comfort on driver drowsiness progress with Predicted Mean Vote: An experiment using real highway driving conditions. *Transportation Research Part F: Traffic Psychology and Behaviour*, *94*, 517–527. https://doi.org/10.1016/j.trf.2023.03.009

Tang, J., Su, Q., Su, B., Fong, S., Cao, W., & Gong, X. (2020). Parallel ensemble learning of convolutional neural networks and local binary patterns for face

recognition. *Computer Methods and Programs in Biomedicine*, *197*. https://doi.org/10.1016/j.cmpb.2020.105622

Tang, Y., Qi, L., Xie, F., Li, X., Ma, C., & Yang, M.-H. (2025). *Video Prediction Transformers without Recurrence or Convolution*. https://doi.org/https://doi.org/10.48550/arXiv.2410.04733

Taoufik, N., Boumya, W., Achak, M., Chennouk, H., Dewil, R., & Barka, N. (2022). The state of art on the prediction of efficiency and modeling of the processes of pollutants removal based on machine learning. Dalam *Science of the Total Environment* (Vol. 807). https://doi.org/10.1016/j.scitotenv.2021.150554

Tian, X., Jin, Y., Zhang, Z., Liu, P., & Tang, X. (2025). Video summarization with temporal-channel visual transformer. *Pattern Recognition*, *165*, 111631. https://doi.org/10.1016/j.patcog.2025.111631

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. ukasz, & Polosukhin, I. (2017). Attention is All you Need. Dalam I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Ed.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc. https://doi.org/https://doi.org/10.48550/arXiv.1706.03762

Venkateswarlu, M., & Chirra, V. R. R. (2025). CNN-ViT: A multi-feature learning based approach for driver drowsiness detection. *Array*, *27*. https://doi.org/10.1016/j.array.2025.100425

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*. https://doi.org/10.1109/cvpr.2001.990517

Wang, J., Yang, X., Li, H., Liu, L., Wu, Z., & Jiang, Y.-G. (2021). *Efficient Video Transformers with Spatial-Temporal Token Selection*. https://doi.org/https://doi.org/10.48550/arXiv.2111.11591

Wang, M., & Deng, W. (2018). *Deep Face Recognition: A Survey*. https://doi.org/10.1016/j.neucom.2020.10.081

Wang, Z., & Liu, Y. (2024). *STAA: Spatio-Temporal Attention Attribution for Real-Time Interpreting Transformer-based Video Models*. https://doi.org/https://doi.org/10.48550/arXiv.2411.00630

William, I., Ignatius Moses Setiadi, D. R., Rachmawanto, E. H., Santoso, H. A., & Sari, C. A. (2019). Face Recognition using FaceNet (Survey, Performance Test, and Comparison). *Proceedings of 2019 4th International Conference on*

Informatics and Computing, ICIC 2019. https://doi.org/10.1109/ICIC47613.2019.8985786

Xu, D., Huang, Q., Zhang, X., Cheng, H., Shuang, F., & Cai, Y. (2025). DEVICE: Depth and Visual Concepts Aware Transformer for OCR-based image captioning. *Pattern Recognition*, *164*, 111522. https://doi.org/10.1016/j.patcog.2025.111522

Xu, W., Xu, Y., Chang, T., & Tu, Z. (2021). *Co-Scale Conv-Attentional Image Transformers*. https://doi.org/https://doi.org/10.48550/arXiv.2104.06399

Yang, H., Liu, L., Min, W., Yang, X., & Xiong, X. (2021). Driver Yawning Detection Based on Subtle Facial Action Recognition. *IEEE Transactions on Multimedia*, *23*. https://doi.org/10.1109/TMM.2020.2985536

Yeo, M. V. M., Li, X., Shen, K., & Wilder-Smith, E. P. V. (2009). Can SVM be used for automatic EEG detection of drowsiness during car driving? *Safety Science*, *47*(1). https://doi.org/10.1016/j.ssci.2008.01.007

Yoshihi, M., Okada, S., Wang, T., Kitajima, T., & Makikawa, M. (2021). Estimating sleep stages using a head acceleration sensor. *Sensors (Switzerland)*, *21*(3). https://doi.org/10.3390/s21030952

Zhang, C., Liu, H., Deng, Y., Xie, B., & Li, Y. (2023). *TokenHPE: Learning Orientation Tokens for Efficient Head Pose Estimation via Transformers*. https://doi.org/10.1109/cvpr52729.2023.00859

Zhang, T., Xu, W., Luo, B., & Wang, G. (2025). Depth-Wise Convolutions in Vision Transformers for efficient training on small datasets. *Neurocomputing*, *617*, 128998. https://doi.org/10.1016/j.neucom.2024.128998

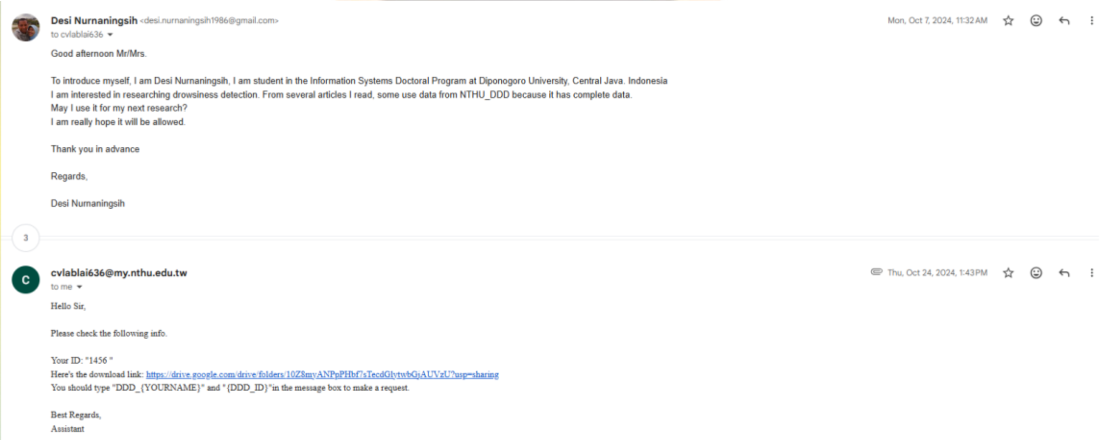
Zhang, Z., Xue, W., Zhou, Y., Zhang, K., & Chen, S. (2024). Hunt-inspired Transformer for visual object tracking. *Pattern Recognition*, *156*, 110703. https://doi.org/10.1016/j.patcog.2024.110703

Zhao, C., Zheng, C., Zhao, M., Tu, Y., & Liu, J. (2011). Multivariate autoregressive models and kernel learning algorithms for classifying driving mental fatigue based on electroencephalographic. *Expert Systems with Applications*, *38*(3). https://doi.org/10.1016/j.eswa.2010.07.115

Zhao, J., Zhang, F., Cai, Y., Tian, G., Mu, W., Ye, C., & Feng, T. (2024). *Learning Sequence Descriptor based on Spatio-Temporal Attention for Visual Place Recognition*. https://doi.org/10.1109/LRA.2024.3354627

LAMPIRAN

1. Surel Meminta Pengambilan Data



Desi Nurnaningsih <desi.nurnaningsih1986@gmail.com> — Mon, Oct 7, 2024, 11:32 AM
to cvlablai636

Good afternoon Mr/Mrs.

To introduce myself, I am Desi Nurnaningsih, I am student in the Information Systems Doctoral Program at Diponogoro University, Central Java. Indonesia
I am interested in researching drowsiness detection. From several articles I read, some use data from NTHU_DDD because it has complete data.
May I use it for my next research?
I am really hope it will be allowed.

Thank you in advance

Regards,

Desi Nurnaningsih

3

cvlablai636@my.nthu.edu.tw — Thu, Oct 24, 2024, 1:43 PM
to me

Hello Sir,

Please check the following info.

Your ID: "1456 "
Here's the download link: https://drive.google.com/drive/folders/10Z8myANPpPHbf7sTecdGlytwbGjAUVzU?usp=sharing
You should type "DDD_{YOURNAME}" and "{DDD_ID}"in the message box to make a request.

Best Regards,
Assistant

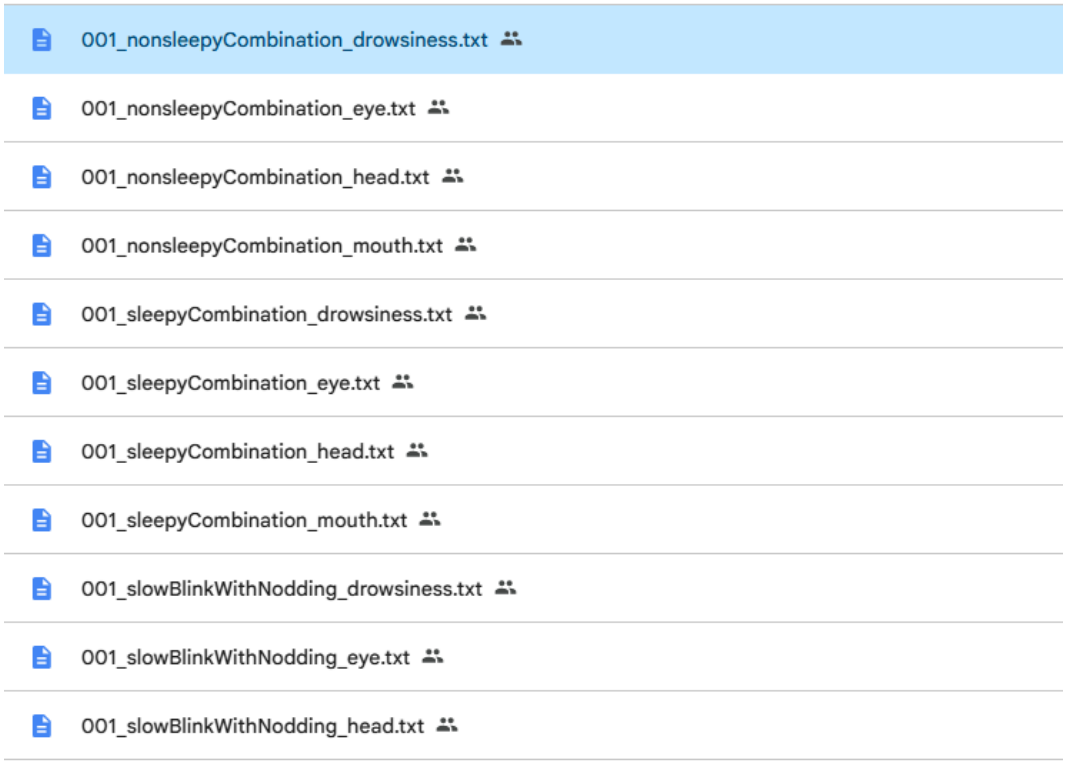
2. Bentuk Data Video

nonsleepyCombination	28/06/2016 16:15	AVI File	27.199 KB
sleepyCombination	28/06/2016 16:15	AVI File	22.423 KB
slowBlinkWithNodding	28/06/2016 16:15	AVI File	15.848 KB
yawning	28/06/2016 16:15	AVI File	15.099 KB

Video

Length	00:01:51
Frame width	640
Frame height	480
Data rate	2322kbps
Total bitrate	2322kbps
Frame rate	30.00 frames/second

3. Bentuk Data Text



4. Contoh Kodingan Program Aplikasi

```
<!DOCTYPE html>

<html lang="id">

<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Visualisasi Deteksi Kantuk (ViT + GRU)</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;600;700&
display=swap" rel="stylesheet">
    <!-- MediaPipe Scripts -->
    <script src="https://cdn.jsdelivr.net/npm/@mediapipe/face_mesh/face_mesh.js" cros
sorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/@mediapipe/drawing_utils/drawing_utils.js"
 crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/@mediapipe/camera_utils/camera_utils.js" c
rossorigin="anonymous"></script>
    <style>
        body {
            font-family: 'Inter', sans-serif;
            background-color: #f0f4f8;
        }
        .step-card {
            transition: all 0.3s ease-in-out;
            border-left-width: 4px;
        }
        .step-card.active {
            transform: scale(1.03);
            box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1), 0 4px 6px -2px rgba(0, 0, 0, 0.05);
            border-color: #3b82f6; /* blue-500 */
        }
        .step-card.inactive {
```

```
        filter: grayscale(80%);
        opacity: 0.7;
      }
      .btn-running {
        cursor: not-allowed;
      }
      #recommendationBox {
        transition: all 0.3s ease-in-out;
      }
    </style>
</head>
<body class="bg-slate-100 text-slate-800">
    <div class="container mx-auto p-4 md:p-8">
      <header class="text-center mb-8">
        <h1 class="text-3xl md:text-4xl font-bold text-slate-900">Pengembangan Model Deteksi Kantuk</h1>
        <p class="text-lg text-slate-600 mt-2">Visualisasi Interaktif menggunakan Metode <span class="font-semibold text-blue-600">Video Vision Transformer + GRU</span></p>
      </header>

      <div class="grid grid-cols-1 lg:grid-cols-2 gap-8 items-start">
        <!-- Kolom Kiri: Simulasi Visual -->
        <div class="bg-white p-6 rounded-2xl shadow-lg sticky top-8">
          <h2 class="text-2xl font-bold mb-4 text-center">Simulasi & Analisis Real-time</h2>
          <div class="relative w-full aspect-video bg-slate-800 rounded-lg overflow-hidden mb-4">
            <img id="driverImage" src="https://placehold.co/640x360/1e293b/ffffff?text=Pilih+Mode" alt="Driver" class="w-full h-full object-cover">
```

```
            <video id="videoFeed" playsinline class="hidden w-full h-full object-cover transform scaleX(-1)"></video>
            <canvas id="overlayCanvas" class="absolute top-0 left-0 w-full h-full"></canvas>
        </div>
        <div class="flex justify-center gap-4">
            <button id="simulationButton" class="bg-blue-600 text-white font-bold py-3 px-6 rounded-lg shadow-md hover:bg-blue-700 transition-all duration-300 transform hover:scale-105">
                Mulai Simulasi
            </button>
            <button id="cameraButton" class="bg-teal-600 text-white font-bold py-3 px-6 rounded-lg shadow-md hover:bg-teal-700 transition-all duration-300 transform hover:scale-105">
                Gunakan Kamera Live
            </button>
            <button id="resetButton" class="hidden bg-red-600 text-white font-bold py-3 px-6 rounded-lg shadow-md hover:bg-red-700 transition-all duration-300 transform hover:scale-105">
                Reset
            </button>
        </div>
        <div id="statusOutput" class="mt-6 text-center">
            <p class="text-xl font-semibold">Status Pengemudi:</p>
            <p id="drowsinessStatus" class="text-3xl font-bold text-gray-400 transition-colors duration-300">- Belum Ada Data -</p>
        </div>
        <!-- Bagian Rekomendasi -->
        <div id="recommendationBox" class="mt-4 p-4 rounded-lg text-center hidden">
            <h4 class="font-bold text-lg mb-2">Rekomendasi Sistem</h4>
            <p id="recommendationText" class="text-md"></p>
        </div>
```

```
        </div>


        <!-- Kolom Kanan: Penjelasan Algoritma -->
        <div class="space-y-6">
          <!-- Visualisasi Proses Training & Testing -->
          <div class="bg-white p-5 rounded-lg shadow">
            <h2 class="text-xl font-bold mb-4 text-center text-slate-900 border-b pb-2">Proses Training & Testing Model</h2>
            <div class="my-4 text-center">
              <button id="loadDatasetButton" class="bg-indigo-600 text-white font-semibold py-2 px-6 rounded-lg shadow-sm hover:bg-indigo-700 transition-all duration-300">
                Input Dataset (NTHU-DDD)
              </button>
              <p id="datasetStatus" class="text-sm text-green-600 mt-2 font-semibold hidden">Dataset NTHU-DDD berhasil dimuat!</p>
            </div>
          </div>

          <h2 class="text-2xl font-bold mb-4 text-center text-slate-900">Tahapan Proses Algoritma (Inference)</h2>

          <div id="step-1" class="step-card inactive bg-white p-5 rounded-lg shadow">
            <h3 class="text-xl font-semibold text-slate-800 mb-2">1. Input & Deteksi Wajah</h3>
            <p class="text-slate-600 mb-3">Sistem menerima frame video dan mendeteksi lokasi wajah pengemudi.</p>
            <div class="text-sm font-medium">Frame saat ini: <span id="frameCounter" class="font-bold">0</span></div>
          </div>
```

```
            <div id="step-2" class="step-card inactive bg-white p-5 rounded-lg shadow">
                <h3 class="text-xl font-semibold text-slate-800 mb-2">2. Ekstraksi Fitur Kunci</h3>
                <p class="text-slate-600 mb-4">Sistem mengidentifikasi <i>facial landmarks</i> untuk mengekstrak fitur.</p>
                <div class="space-y-3">
                    <div>
                        <label class="font-medium text-sm">Kondisi Mata (EAR)</label>
                        <div class="w-full bg-slate-200 rounded-full h-2.5"><div id="eye-progress" class="bg-blue-500 h-2.5 rounded-full" style="width: 0%"></div></div>
                    </div>
                    <div>
                        <label class="font-medium text-sm">Kondisi Mulut (MAR)</label>
                        <div class="w-full bg-slate-200 rounded-full h-2.5"><div id="mouth-progress" class="bg-green-500 h-2.5 rounded-full" style="width: 0%"></div></div>
                    </div>
                    <div>
                        <label class="font-medium text-sm">Kemiringan Kepala (°)</label>
                        <div class="w-full bg-slate-200 rounded-full h-2.5"><div id="head-progress" class="bg-purple-500 h-2.5 rounded-full" style="width: 50%"></div></div>
                    </div>
                </div>
            </div>

            <div id="step-3" class="step-card inactive bg-white p-5 rounded-lg shadow">
                <h3 class="text-xl font-semibold text-slate-800 mb-2">3. Analisis Temporal (ViT + GRU)</h3>
                <p class="text-slate-600">Model ViT +
GRU menganalisis urutan (sekuens) fitur dari beberapa frame untuk memahami konteks waktu dan perubahan perilaku.</p>
                <div id="gru-animation" class="flex space-x-2 mt-4 h-8 items-center">
```

```
                <span class="text-sm font-medium">Frames:</span>
              </div>
            </div>

            <div id="step-4" class="step-card inactive bg-white p-5 rounded-lg shadow">
              <h3 class="text-xl font-semibold text-slate-800 mb-2">4. Output Klasifikasi</h3>
              <p class="text-slate-600">Berdasarkan analisis,
model menghasilkan output akhir yang mengklasifikasikan kondisi pengemudi.</p>
              <div id="output-final" class="mt-3 text-center text-lg font-bold text-gray-400">- Menunggu Hasil -</div>
            </div>
          </div>
        </div>
      </div>

    <script type="module">
      const simulationButton = document.getElementById('simulationButton');
      const cameraButton = document.getElementById('cameraButton');
      const resetButton = document.getElementById('resetButton');
      const driverImage = document.getElementById('driverImage');
      const videoFeed = document.getElementById('videoFeed');
      const canvas = document.getElementById('overlayCanvas');
      const ctx = canvas.getContext('2d');
      const statusText = document.getElementById('drowsinessStatus');
      const loadDatasetButton = document.getElementById('loadDatasetButton');
      const datasetStatus = document.getElementById('datasetStatus');
      const recommendationBox = document.getElementById('recommendationBox');
      const recommendationText = document.getElementById('recommendationText');
      const frameCounter = document.getElementById('frameCounter');
```

```
const eyeProgress = document.getElementById('eye-progress');
const mouthProgress = document.getElementById('mouth-progress');
const headProgress = document.getElementById('head-progress');
const gruAnimationContainer = document.getElementById('gru-animation');
const outputFinal = document.getElementById('output-final');
const steps = [
    document.getElementById('step-1'),
    document.getElementById('step-2'),
    document.getElementById('step-3'),
    document.getElementById('step-4'),
];

let audioCtx;
let alarmBeeping = false;
let currentMode = 'idle'; // 'idle', 'simulation', 'live'
let simulationInterval;
let frameCount = 0;
let camera;
let faceMesh;

// --- REFINED DETECTION LOGIC VARIABLES ---
let eyeClosedCounter = 0;
let yawnCounter = 0;
const EAR_THRESHOLD = 0.22; // Eyes are considered closed if EAR is below this
const EAR_CONSECUTIVE_FRAMES = 15; // Trigger drowsy if eyes are closed for ~0.5s
const MAR_THRESHOLD = 0.6; // Mouth is considered yawning if MAR is above this
const YAWN_CONSECUTIVE_FRAMES = 25; // Trigger drowsy if yawning for ~0.8s
```

```
    // --- SIMULATION DATA ---
    const simulationData = [
                [1, 0.35, 0.1,          -2, 'alert_1.png'],          [2, 0.33, 0.1, 0, 'alert_2.png'],
[3, 0.36, 0.1, 1, 'alert_1.png'],
                [4, 0.34, 0.1,          -1, 'alert_2.png'],          [5, 0.25, 0.15, 5, 'drowsy_1.png'],
[6, 0.20, 0.2, 8, 'drowsy_2.png'],
                [7, 0.15, 0.25, 12, 'drowsy_1.png'],          [8, 0.28, 0.6, 5, 'yawning_1.png'],
[9, 0.25, 0.8, 2, 'yawning_2.png'],
                [10, 0.29, 0.5, 0, 'yawning_1.png'],          [11, 0.10, 0.2, 15, 'sleepy_1.png'],
[12, 0.08, 0.2, 18, 'sleepy_2.png'],
                [13, 0.07, 0.2, 20, 'sleepy_1.png'],          [14, 0.09, 0.18, 17, 'sleepy_2.png'],
[15, 0.12, 0.15, 15, 'sleepy_1.png'],
    ];
    const imageUrls = {
        'alert_1.png': 'https://placehold.co/640x360/1e293b/a3e635?text=Waspada', 'ale
rt_2.png': 'https://placehold.co/640x360/1e293b/86efac?text=Waspada',
        'drowsy_1.png': 'https://placehold.co/640x360/1e293b/facc15?text=Mulai+Lelah'
, 'drowsy_2.png': 'https://placehold.co/640x360/1e293b/fbbf24?text=Mulai+Lelah',
        'yawning_1.png': 'https://placehold.co/640x360/1e293b/f97316?text=Menguap',
'yawning_2.png': 'https://placehold.co/640x360/1e293b/fb923c?text=Menguap',
        'sleepy_1.png': 'https://placehold.co/640x360/1e293b/f87171?text=Mengantuk',
'sleepy_2.png': 'https://placehold.co/640x360/1e293b/ef4444?text=Sangat+Mengantuk'
,
    };

    // --- CORE FUNCTIONS ---
    function playAlarm() {
        if (!audioCtx || alarmBeeping) return;
        alarmBeeping = true;
        let beepCount = 0;
        const maxBeeps = 3;
        function beep() {
```

```
            if (beepCount >= maxBeeps) { alarmBeeping = false; return; }
            const oscillator = audioCtx.createOscillator();
            const gainNode = audioCtx.createGain();
            oscillator.connect(gainNode); gainNode.connect(audioCtx.destination);
            oscillator.type = 'sine'; oscillator.frequency.setValueAtTime(900, audioCtx.curren
tTime);
            gainNode.gain.setValueAtTime(0.5, audioCtx.currentTime);
            oscillator.start(); oscillator.stop(audioCtx.currentTime + 0.2);
            beepCount++; setTimeout(beep, 400);
        }
        beep();
    }

    function updateUIWithFeatures(eyeRatio, mouthRatio, headTilt) {
        frameCount++;
        frameCounter.textContent = frameCount;

        // --- REFINED DETECTION LOGIC ---
        if (eyeRatio < EAR_THRESHOLD) {
            eyeClosedCounter++;
        } else {
            eyeClosedCounter = 0;
        }

        if (mouthRatio > MAR_THRESHOLD) {
            yawnCounter++;
        } else {
            yawnCounter = 0;
        }
```

```
        // Step 2: Update Progress Bars
        const eyeClosedness = 100 - ((eyeRatio - 0.1) / 0.3) * 100;
        eyeProgress.style.width = `${Math.min(100, Math.max(0, eyeClosedness))}%`;
        const mouthOpenness = (mouthRatio / 0.8) * 100;
        mouthProgress.style.width = `${Math.min(100, Math.max(0, mouthOpenness))}%`
;
        const tiltPercentage = 50 + (headTilt * 2);
        headProgress.style.width = `${Math.min(100, Math.max(0, tiltPercentage))}%`;

        // Step 3: Animate GRU
        if (frameCount % 10 === 0) {
            const animBlock = document.createElement('div');
            animBlock.className = 'w-6 h-6 rounded bg-slate-300 animate-pulse';
            gruAnimationContainer.appendChild(animBlock);
            if (gruAnimationContainer.children.length > 8) {
                gruAnimationContainer.children[1].remove();
            }
        }

        // Step 4: Determine Drowsiness Status & Recommendation
        let currentStatus, statusColor, outputColor, recText, recClass;

        if (eyeClosedCounter > EAR_CONSECUTIVE_FRAMES || yawnCounter > YAWN_CO
NSECUTIVE_FRAMES || Math.abs(headTilt) > 25) {
            [currentStatus, statusColor, outputColor, recText, recClass] = ['Mengantuk', 'text-
red-500', 'text-red-500', 'BAHAYA! Segera menepi dan istirahat.      Alarm diaktifkan.', 'bg-
red-100 text-red-800'];
            playAlarm();
```

```
                } else if (eyeClosedCounter > 5 || yawnCounter > 5 || Math.abs(headTilt)
> 15 || mouthRatio > 0.4) {
            [currentStatus, statusColor, outputColor, recText, recClass] = ['Mulai Lelah', 'text-
yellow-500', 'text-yellow-500', 'Disarankan untuk mencari tempat istirahat terdekat.', 'bg-
yellow-100 text-yellow-800'];
        } else {
            [currentStatus, statusColor, outputColor, recText, recClass] = ['Waspada', 'text-
green-500', 'text-green-500', 'Kondisi Anda baik.
Terus fokus dan berkendara dengan aman.', 'bg-green-100 text-green-800'];
        }
        statusText.textContent = currentStatus;
        statusText.className = `text-3xl font-bold ${statusColor} transition-
colors duration-300`;
        outputFinal.textContent = currentStatus;
        outputFinal.className = `mt-3 text-center text-lg font-bold ${outputColor}`;
        recommendationText.textContent = recText;
        recommendationBox.className = `mt-4 p-4 rounded-lg text-center ${recClass}`;
        recommendationBox.classList.remove('hidden');
    }

    function resetAll() {
        if (currentMode === 'simulation') clearInterval(simulationInterval);
        if (currentMode === 'live' && camera) camera.stop();
        currentMode = 'idle';
        frameCount = 0;
        eyeClosedCounter = 0;
        yawnCounter = 0;

        ctx.clearRect(0, 0, canvas.width, canvas.height);
        statusText.textContent = '- Belum Ada Data -';
        statusText.className = 'text-3xl font-bold text-gray-400';
```

```
        outputFinal.textContent = '- Menunggu Hasil -';
        outputFinal.className = 'mt-3 text-center text-lg font-bold text-gray-400';
        driverImage.src = 'https://placehold.co/640x360/1e293b/ffffff?text=Pilih+Mode';
        driverImage.classList.remove('hidden');
        videoFeed.classList.add('hidden');

        simulationButton.classList.remove('hidden');
        cameraButton.classList.remove('hidden');
        resetButton.classList.add('hidden');

        gruAnimationContainer.innerHTML = '<span class="text-sm font-medium">Frames:</span>';
        steps.forEach(step => {
            step.classList.add('inactive');
            step.classList.remove('active');
        });
        eyeProgress.style.width = '0%';
        mouthProgress.style.width = '0%';
        headProgress.style.width = '50%';
        recommendationBox.classList.add('hidden');
    }

    // --- SIMULATION MODE ---
    function runSimulationFrame() {
        if (frameCount >= simulationData.length) {
            resetAll();
            return;
        }
```

```
        const [frame, eyeRatio, mouthRatio, headTilt, imageName]
= simulationData[frameCount];
        driverImage.src = imageUrls[imageName];
        steps.forEach(s => s.classList.remove('inactive'));
        activateStep(frameCount % 4);
        // In simulation, we use the predefined data, not the counter logic for status
        const drowsyScore =  ((0.35 - eyeRatio)  * 5)  +  (mouthRatio * 1.5)  +
(Math.abs(headTilt) / 15);
        updateUIWithFeatures(eyeRatio, mouthRatio, headTilt, drowsyScore > 1.8, drowsy
Score > 1.0);
    }

    function startSimulationMode() {
        if (currentMode !== 'idle') resetAll();
        currentMode = 'simulation';
        frameCount = 0;

        simulationButton.classList.add('hidden');
        cameraButton.classList.add('hidden');
        resetButton.classList.remove('hidden');

        runSimulationFrame();
        simulationInterval = setInterval(runSimulationFrame, 1500);
    }

    // --- LIVE CAMERA MODE ---
    function onResults(results) {
        ctx.clearRect(0, 0, canvas.width, canvas.height);
        if (results.multiFaceLandmarks && results.multiFaceLandmarks[0]) {
            const landmarks = results.multiFaceLandmarks[0];
```

```
            drawConnectors(ctx,                          landmarks, FACEMESH_TESSELATION,
{color: '#C0C0C070', lineWidth: 1});


            const L_EYE_INDICES = [33, 160, 158, 133, 153, 144];
            const R_EYE_INDICES = [362, 385, 387, 263, 373, 380];

            const lEye = L_EYE_INDICES.map(i => landmarks[i]);
            const rEye = R_EYE_INDICES.map(i => landmarks[i]);
            const ear = (getEAR(lEye) + getEAR(rEye)) / 2;

            const MOUTH_INDICES = [61, 291, 0, 17];
            const mouth = MOUTH_INDICES.map(i => landmarks[i]);
            const mar = getMAR(mouth);

            const nose = landmarks[1];
            const chin = landmarks[152];
            let headTilt = 0;
            if (nose && chin) {
                headTilt = -Math.atan2(chin.y - nose.y, chin.x - nose.x) * (180 / Math.PI) + 90;
            }

            updateUIWithFeatures(ear, mar, headTilt);
        } else {
            updateUIWithFeatures(0.35, 0.1, 0); // Waspada state if no face
        }
    }

    function getEAR(eye) {
        const p1 = eye[0], p2 = eye[1], p3 = eye[2], p4 = eye[3], p5 = eye[4], p6 = eye[5];
```

```
    const dist = (pA, pB) => Math.hypot(pA.x - pB.x, pA.y - pB.y, pA.z - pB.z);
    return (dist(p2, p6) + dist(p3, p5)) / (2 * dist(p1, p4));
}

function getMAR(mouth) {
    const p1 = mouth[0], p2 = mouth[1], p3 = mouth[2], p4 = mouth[3];
    const dist = (pA, pB) => Math.hypot(pA.x - pB.x, pA.y - pB.y, pA.z - pB.z);
    return dist(p3,p4) / dist(p1,p2);
}

function activateStep(index) {
    steps.forEach((s, i) => s.classList.toggle('active', i === index));
}

function startLiveMode() {
    if (currentMode !== 'idle') resetAll();
    currentMode = 'live';

    driverImage.classList.add('hidden');
    videoFeed.classList.remove('hidden');
    simulationButton.classList.add('hidden');
    cameraButton.classList.add('hidden');
    resetButton.classList.remove('hidden');

    faceMesh = new FaceMesh({locateFile: (file) => {
        return `https://cdn.jsdelivr.net/npm/@mediapipe/face_mesh/${file}`;
    }});
    faceMesh.setOptions({
        maxNumFaces: 1,
```

```
        refineLandmarks: true,
        minDetectionConfidence: 0.5,
        minTrackingConfidence: 0.5
    });
    faceMesh.onResults(onResults);

    camera = new Camera(videoFeed, {
        onFrame: async () => {
            await faceMesh.send({image: videoFeed});
        },
        width: 640,
        height: 360
    });
    camera.start();
}

// --- EVENT LISTENERS & INITIALIZATION ---
window.addEventListener('resize', () => {
    const container = driverImage.parentElement;
    canvas.width = container.clientWidth;
    canvas.height = container.clientHeight;
});

loadDatasetButton.addEventListener('click', () => {
    datasetStatus.classList.remove('hidden');
    loadDatasetButton.disabled = true;
    loadDatasetButton.textContent = 'Dataset Dimuat';
    loadDatasetButton.classList.add('bg-gray-400', 'cursor-not-allowed');
});
```

```
    simulationButton.onclick = () => {
      if (!audioCtx) audioCtx = new (window.AudioContext || window.webkitAudioCont
ext)();
      startSimulationMode();
    };

    cameraButton.onclick = () => {
      if (!audioCtx) audioCtx = new (window.AudioContext || window.webkitAudioCont
ext)();
      startLiveMode();
    };

    resetButton.onclick = resetAll;

    // Initial setup
    const container = driverImage.parentElement;
    canvas.width = container.clientWidth;
    canvas.height = container.clientHeight;
    resetAll();
  </script>
</body>
</html>
```