

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Tabel 2.1 Tabel Penelitian Terdahulu

Nama Jurnal	Judul Artikel	Penulis	Metode	Hasil
<i>Knowledge-Based Systems, 2022</i>	<i>Driver drowsiness detection in video sequences using hybrid selection deep features</i>	Bekhouche, S.E.	<i>Convolutional Neural Network (CNN)</i>	Penelitian ini menunjukkan bahwa CNN mampu mendeteksi kantuk secara efektif melalui analisis sekuens video wajah, serta memberikan performa tinggi pada identifikasi perubahan ekspresi terkait kelelahan [26].
<i>International Journal of Simulation Modelling, 2021</i>	<i>Real-time detection method of driver fatigue stage based on deep learning of face video</i>	Cui, Z.; Sun, H.M.; Yin, R.N.; Gao, L.	<i>Face Landmark Extraction + Threshold-Based Fatigue Detection</i>	Hasil penelitian menunjukkan bahwa metode <i>facial landmark</i> dapat mendeteksi kelelahan secara <i>real-time</i> dengan mengidentifikasi perubahan fitur wajah seperti kedipan dan penurunan posisi kepala [27].
<i>Materials Today: Proceedings, 2023</i>	<i>Facial Feature-Based Drowsiness Detection with Multi-Scale Convolutional Neural Network</i>	Vijaypriya, V.; Uma, M.	<i>CNN + Multilayer Perceptron (MLP)</i>	Penelitian ini menunjukkan bahwa kombinasi CNN dan MLP mampu meningkatkan akurasi deteksi kantuk dibanding model tunggal, dengan performa yang stabil pada

				kondisi visual berbeda [28].
<i>Sensors, 2025</i>	<i>Real-Time Driver Drowsiness Detection Using Facial Analysis and Machine Learning Techniques</i>	Essahraoui, S.; Lamaakal, I.; El Hamly, M.E.	<i>Face Feature Extraction + Classification Model (Non-DL)</i>	Sistem yang dikembangkan mampu menganalisis fitur wajah secara <i>real-time</i> dan menunjukkan kinerja yang baik dalam mengidentifikasi tanda kantuk seperti penurunan kewaspadaan dan penutupan mata [29].
<i>IEEE Access, 2021</i>	<i>Early Identification and Detection of Driver Drowsiness by Hybrid Machine Learning</i>	Altameem, A.; Kumar, A.; Poonia, R.; et al.	<i>Convolutional Neural Network (CNN)</i>	Penelitian ini menunjukkan bahwa CNN dapat digunakan secara efektif untuk mendeteksi kantuk pada tahap awal serta memberikan sistem peringatan dini guna mencegah <i>microsleep</i> , meningkatkan keamanan berkendara secara signifikan [30].
<i>Applied Computing and Informatics, 2024</i>	<i>A Real-Time Embedded System for Driver Drowsiness Detection Based on Visual Analysis of the Eyes and Mouth Using Convolutional Neural Network and Mouth Aspect Ratio</i>	Florez, R.; Palomino-Quispe, F.; Alvarez, N.; et al.	<i>Behavioral Feature Analysis (Eye Aspect Ratio, Head Pose)</i>	Hasil penelitian menunjukkan bahwa indikator perilaku seperti kedipan lambat dan gerakan kepala dapat digunakan secara efektif untuk mengidentifikasi kelelahan, serta menunjukkan performa baik pada sistem <i>embedded real-time</i> [31].

<i>Applied Intelligence, 2021</i>	<i>An efficient anomaly recognition framework using an attention residual lstm in surveillance videos</i>	Ullah, W.; Ullah, A.; Haq, I.U.; et al.	<i>CNN + Bidirectional LSTM (Bi-LSTM)</i>	Model <i>hybrid</i> CNN–BiLSTM menunjukkan akurasi tinggi dalam deteksi kantuk <i>real-time</i> , dan penelitian melaporkan bahwa integrasi fitur visual dengan pemodelan temporal menghasilkan peningkatan performa signifikan dibanding CNN saja [32].
<i>IEEE Sensors Journal, 2022</i>	<i>Driver drowsiness detection using facial thermal imaging in a driving simulator</i>	Tashakori, M.; Nahvi, A.; Ebrahimian, H.; et al.	<i>Thermal Imaging + Feature Extraction</i>	Penelitian ini menunjukkan bahwa analisis gambar termal dapat mengidentifikasi penurunan suhu wajah yang berkaitan dengan kantuk, memberikan alternatif non-visual yang efektif untuk deteksi kelelahan [33].
<i>Journal of King Saud University – Computer and Information Sciences, 2024</i>	<i>A CNN-based Deep Learning Framework for Driver's Drowsiness Detection</i>	Sohail, A.; Shah, A.A.; Ilyas, S.; et al.	<i>Hybrid CNN Architecture</i>	<i>Framework</i> CNN menunjukkan <i>robustnes</i> yang baik terhadap variasi lingkungan, dan penelitian melaporkan peningkatan akurasi melalui optimasi arsitektur CNN untuk mendeteksi penutupan mata dan perubahan ekspresi [34].

<i>Journal of Physics: Conference Series, 2022</i>	<i>Microsleep Detection of Automobile Drivers Using Electrocardiogram Signal</i>	Nor Shahrudin, N.S.; Sidek, K.A.; Jalaludin, S.N.	<i>CNN + Eye state analysis</i>	Penelitian ini menunjukkan bahwa CNN mampu mendeteksi <i>microsleep</i> berdasarkan durasi penutupan mata, serta memberikan akurasi tinggi dalam identifikasi <i>state</i> mata (<i>open/closed</i>) sebagai indikator utama <i>microsleep</i> [35].
--	--	---	---------------------------------	--

Penelitian-penelitian terdahulu yang tercantum pada Tabel 2.1 menunjukkan perkembangan signifikan dalam pemanfaatan teknik *deep learning*, terutama *Convolutional Neural Network* (CNN), untuk mendeteksi kelelahan dan potensi *microsleep* pada pengemudi. Berbagai studi berfokus pada analisis visual berbasis kamera, seperti pola kedipan mata, durasi penutupan kelopak mata, perubahan pose kepala, serta ekspresi wajah [26], [27], [29], [31], [33]. Hasil penelitian terdahulu secara umum menunjukkan bahwa CNN memiliki kemampuan yang kuat dalam mengekstraksi fitur spasial dari gambar atau video wajah, sehingga dapat mengidentifikasi tanda-tanda kantuk dengan stabil meskipun terjadi perubahan ekspresi atau kondisi visual tertentu [34]. Pendekatan lain yang mengandalkan *facial landmark* dan analisis video *real-time* juga terbukti efektif untuk mendeteksi perubahan mikro pada area mata dan kepala yang menjadi indikator awal kelelahan pengemudi.

Selain metode berbasis CNN tunggal, sejumlah penelitian mulai mengembangkan arsitektur yang lebih kompleks dengan mengombinasikan CNN dan *Multi-Layer Perceptron* (MLP) untuk meningkatkan performa klasifikasi. Pendekatan ini terbukti menghasilkan akurasi yang lebih baik dalam mendeteksi pola visual yang berkaitan dengan kantuk [28]. Ada pula penelitian yang menekankan analisis fitur wajah secara lebih mendalam untuk sistem deteksi *real-time* serta model yang memfokuskan deteksi pada indikator perilaku, seperti

frekuensi kedipan dan pergerakan kepala, yang kemudian diterapkan pada sistem *embedded*. Pendekatan-pendekatan ini menunjukkan bahwa analisis perilaku visual memiliki potensi besar dalam mendeteksi penurunan kewaspadaan secara akurat.

Beberapa penelitian lainnya mulai memanfaatkan model *hybrid* yang menggabungkan kemampuan ekstraksi spasial dari CNN dengan pemodelan temporal menggunakan jaringan seperti LSTM atau BiLSTM. Pendekatan *hybrid* ini memberikan hasil yang lebih baik karena mampu mengenali perubahan bertahap pada perilaku pengemudi, yang merupakan karakteristik utama dari *microsleep* [32]. Penelitian lain memanfaatkan data termal sebagai alternatif untuk mendeteksi perubahan fisiologis terkait kantuk, sementara studi yang melakukan optimasi pada CNN menunjukkan adanya peningkatan signifikan dalam mendeteksi penutupan mata dan kondisi mata (*open/closed*) sebagai indikator utama *microsleep* [30], [35]. Secara keseluruhan, penelitian terdahulu memberikan dasar penting bagi pengembangan sistem deteksi *microsleep*, sekaligus menunjukkan perlunya pendekatan yang mampu memanfaatkan informasi visual secara spasial dan temporal untuk meningkatkan akurasi deteksi dalam berbagai kondisi nyata.

Meskipun berbagai penelitian sebelumnya telah mengimplementasikan CNN dan metode analisis visual untuk mendeteksi kantuk atau kelelahan pengemudi, masih terdapat sejumlah celah penelitian yang belum sepenuhnya teratasi. Sebagian besar studi hanya memanfaatkan CNN untuk mengekstraksi fitur spasial dari gambar atau video wajah, padahal *microsleep* merupakan fenomena yang terjadi dalam hitungan detik dan memiliki pola temporal yang tidak dapat ditangkap hanya melalui analisis *frame* tunggal [26], [27], [29], [31], [33]. Selain itu, banyak penelitian menggunakan dataset dengan kondisi pencahayaan yang homogen dan variasi wajah yang terbatas, sehingga model yang dihasilkan kurang *robust* ketika dihadapkan pada kondisi lingkungan nyata yang beragam. Penelitian terdahulu juga cenderung berfokus pada deteksi *drowsiness* secara umum, bukan *microsleep* secara spesifik, meskipun *microsleep* merupakan fase kritis dengan risiko kecelakaan yang lebih tinggi. Lebih lanjut, hanya sedikit studi yang secara eksplisit menggabungkan CNN dan LSTM dalam satu arsitektur *hybrid* untuk

memanfaatkan keunggulan keduanya dalam mengenali pola spasial sekaligus temporal pada perilaku pengemudi [36], [37].

Secara keseluruhan, penelitian terdahulu menunjukkan bahwa pendekatan berbasis CNN mampu memberikan performa yang baik dalam mendeteksi indikator kantuk dan *microsleep* melalui ekstraksi fitur visual dari wajah pengemudi. Namun, pendekatan tersebut umumnya berfokus pada informasi spasial pada setiap *frame* sehingga belum sepenuhnya memanfaatkan hubungan temporal antar *frame*. Beberapa penelitian kemudian mengombinasikan CNN dengan jaringan temporal seperti LSTM dan BiLSTM untuk memanfaatkan informasi urutan waktu dalam proses klasifikasi. Temuan tersebut menunjukkan bahwa pemanfaatan informasi spasial dan temporal secara bersamaan berpotensi meningkatkan kemampuan model dalam mengenali perubahan perilaku pengemudi yang terjadi secara bertahap. Berdasarkan pertimbangan tersebut, penelitian ini menggunakan arsitektur *Hybrid CNN-LSTM* untuk mengintegrasikan kemampuan ekstraksi fitur visual dan pemodelan temporal dalam mendeteksi *microsleep*.

Research Gap

Berdasarkan celah tersebut, penelitian ini berkontribusi dengan mengembangkan model *hybrid CNN-LSTM* yang dirancang khusus untuk mendeteksi *microsleep* melalui analisis visual berbasis urutan waktu [38], [39], [40]. Penelitian ini berkontribusi dengan mengembangkan sistem deteksi *microsleep* menggunakan *hybrid CNN-LSTM* yang mengintegrasikan ekstraksi fitur spasial dan pemodelan temporal untuk meningkatkan kemampuan identifikasi kondisi kantuk berdasarkan perubahan visual wajah pengemudi. Berbeda dengan sebagian besar penelitian terdahulu yang menganalisis gambar secara independen atau hanya memanfaatkan informasi temporal secara terbatas, penelitian ini menerapkan pendekatan *sequence-based sliding window* untuk membentuk urutan *frame* sehingga hubungan temporal antar *frame* dapat dimanfaatkan dalam proses klasifikasi. Selain itu, model yang dikembangkan diimplementasikan dalam sistem deteksi *microsleep* secara *real-time* melalui *deployment* berbasis *webcam* menggunakan *Gradio* dan *Hugging Face*, sehingga mampu melakukan inferensi

langsung terhadap video dari kamera dengan menampilkan status deteksi beserta *confidence score*. Penelitian ini juga mengevaluasi performa model pada berbagai kondisi pencahayaan dan variasi atribut wajah untuk mengetahui ketahanan sistem pada skenario yang lebih mendekati kondisi berkendara nyata.

2.2 Tinjauan Teori

2.2.1 Pengertian, Faktor, dan Tanda-tanda *Microsleep*

Microsleep adalah kondisi di mana seseorang mengalami kehilangan kesadaran yang singkat, biasanya hanya beberapa detik hingga satu menit. Meskipun waktunya singkat, kondisi ini sangat berbahaya, terutama bagi orang yang sedang mengemudi karena dapat menyebabkan kecelakaan fatal [41]. *Microsleep* sering terjadi dalam situasi stres yang berkepanjangan, kurang tidur, atau kelelahan fisik atau mental yang ekstrim. Karena tubuh tetap terjaga selama *microsleep*, pengemudi tidak menyadari penurunan kesadaran [42].

Penyebab utama dari *microsleep* adalah kurang tidur atau gangguan tidur lainnya. Individu yang memiliki kebiasaan tidur yang buruk, seperti tidur yang terputus atau tidak cukup lama, berisiko lebih besar untuk mengalami *microsleep* [43], [44]. Selain itu, faktor lingkungan juga turut berperan, seperti berkendara pada malam hari atau dalam perjalanan panjang yang monoton. Keadaan tubuh yang lelah atau stres berat, serta penggunaan obat-obatan tertentu, juga dapat memperburuk kondisi ini.

Microsleep seringkali sulit dideteksi, terutama bagi pengemudi. Namun, ada beberapa gejala yang mungkin menjadi tanda bahwa seseorang mengalami *microsleep*. Salah satunya adalah penutupan mata yang lebih lama dari normal. Selain itu, kedip mata yang lebih jarang atau lambat dapat menunjukkan penurunan tingkat kesadaran [45]. Selain itu, pengemudi yang mengalami *microsleep* seringkali kehilangan kemampuan untuk mengontrol posisi kepala mereka, yang menyebabkan kepala jatuh atau bergerak secara tiba-tiba. Perubahan fisik yang dapat dilihat, seperti mata yang terpejam dan kepala yang terkulai, penurunan konsentrasi adalah tanda lain dari *microsleep*. Pengemudi mungkin kehilangan fokus, melamun, atau tidak menyadari

perubahan situasi di sekitarnya, seperti lampu lalu lintas yang berubah atau kendaraan lain yang mendekat [46], [47]. Gejala ini dapat sangat berbahaya karena meningkatkan risiko terjadinya kecelakaan. Deteksi dini tanda-tanda *microsleep* sangat penting untuk mencegah terjadinya kecelakaan lalu lintas dan dapat menjadi kunci untuk mengambil tindakan preventif.

2.2.2 Kamera untuk Mendeteksi *Microsleep* Pada Pengendara Mobil

Deteksi *microsleep* pada pengemudi kendaraan merupakan tantangan besar karena kondisi ini sering terjadi secara singkat dan tanpa disadari oleh pengemudi. Teknologi berbasis kamera dan *computer vision* telah menjadi salah satu pendekatan utama dalam mengidentifikasi tanda-tanda *microsleep* secara otomatis dan *real-time* [48]. Penggunaan kamera dalam sistem deteksi ini sangat bergantung pada analisis wajah dan mata, karena perubahan fisik yang terkait dengan *microsleep* dapat dipantau melalui gambar visual [27], [49].

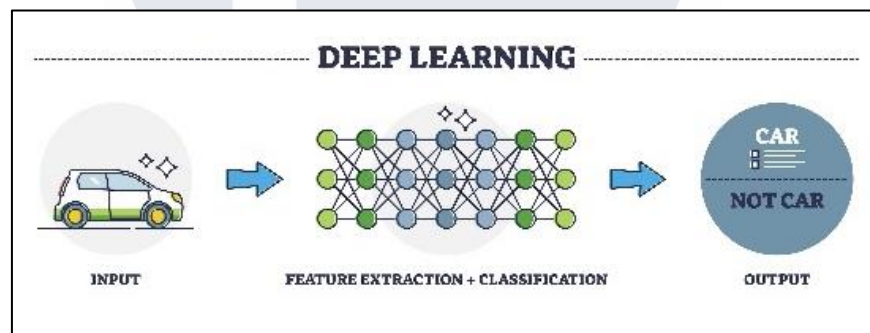
Sistem deteksi berbasis kamera, algoritma pengolahan gambar seperti *OpenCV* sering digunakan untuk mendeteksi dan menganalisis pola kedipan mata pengemudi dalam video secara langsung [50]. Dengan algoritma ini, komputer dapat mengenali ketika mata pengemudi tertutup lebih lama daripada batas normal, yang merupakan salah satu gejala utama *microsleep*. Selain itu, teknologi kamera dapat mendeteksi pergerakan kepala ini dan memberi peringatan kepada pengemudi atau sistem kendaraan untuk mencegah kecelakaan [51]. Penggunaan sensor berbasis video memungkinkan pemantauan terus-menerus yang dapat digunakan sebagai input untuk hybrid CNN-LSTM untuk meningkatkan akurasi deteksi.

Sistem berbasis *computer vision* ini juga menggabungkan analisis ekspresi wajah dan gerakan mata untuk mendeteksi tanda-tanda *microsleep* [52], [53]. Deteksi *microsleep* dengan menggunakan kamera juga memberikan keuntungan karena tidak memerlukan interaksi langsung dengan pengemudi, yang membuatnya lebih aman dan efisien untuk diterapkan dalam kendaraan yang sedang berjalan. Selain itu, dengan berkembangnya teknologi pemrosesan

gambar dan peningkatan kemampuan algoritma dalam memproses data video secara *real-time*, sistem deteksi *microsleep* berbasis kamera kini dapat beroperasi dengan tingkat kecepatan dan akurasi yang lebih tinggi.

2.2.3 *Deep Learning* untuk Pemrosesan Data Gambar

Deep learning merupakan salah satu cabang dari *machine learning* yang menggunakan jaringan saraf buatan berskala besar (*deep neural networks*) untuk mempelajari representasi data secara otomatis [54], [55]. Dalam pemrosesan data gambar, deep learning menjadi pendekatan yang sangat populer karena kemampuannya mengekstraksi fitur visual secara hierarkis, mulai dari pola sederhana hingga struktur kompleks. Jaringan saraf dalam mampu mengenali hubungan antar piksel, bentuk objek, pola tekstur, hingga karakteristik visual abstrak tanpa memerlukan perancangan fitur manual oleh manusia.



Gambar 2.1 Alur Kerja *Deep Learning*

Sumber: *Deep Learning* [56]

Perbedaan utama antara *machine learning* tradisional dan *deep learning* dapat dilihat pada proses *feature extraction*. Pada *machine learning* konvensional, fitur harus ditentukan secara manual melalui teknik seperti *edge detection*, *corner detection*, atau *histogram-based features*. Hal ini membuat kualitas model sangat bergantung pada kemampuan manusia dalam merumuskan fitur yang relevan. Gambar 2.1 menggambarkan bahwa pada *machine learning*, proses dimulai dari input kemudian dilanjutkan dengan ekstraksi fitur oleh manusia sebelum memasuki tahap klasifikasi [56].

Sebaliknya, pada *deep learning*, ekstraksi fitur dan klasifikasi dilakukan secara *end-to-end* di dalam satu model. Jaringan saraf dalam secara otomatis mempelajari fitur yang paling relevan dari data mentah melalui banyak lapisan belajar. Lapisan awal biasanya mendeteksi pola dasar seperti garis dan tepi, sedangkan lapisan yang lebih dalam menangkap struktur yang lebih kompleks seperti bentuk objek atau pola visual tertentu [57]. Gambar pada bagian *deep learning* menunjukkan bahwa seluruh proses, mulai dari input hingga keluaran, ditangani langsung oleh jaringan saraf tanpa memerlukan intervensi manual pada tahap ekstraksi fitur [58].

Kemampuan *deep learning* untuk belajar langsung dari data mentah memberikan keuntungan dalam hal akurasi, generalisasi, dan kemampuan menangani variasi data visual. Selain itu, penggunaan arsitektur jaringan yang semakin dalam, optimisasi GPU, serta teknik regularisasi modern membuat *deep learning* mampu memproses gambar dalam jumlah besar dan menghasilkan performa yang lebih baik dibandingkan pendekatan *machine learning* tradisional [59]. Dengan keunggulan ini, *deep learning* menjadi fondasi utama dalam berbagai aplikasi pemrosesan gambar modern, seperti *image classification*, *object detection*, *face recognition*, dan *video analysis*.

2.3 Kerangka Kerja dan Algoritma

2.3.1 Kerangka Kerja

Penelitian ini membutuhkan metode *data mining* yang mampu mengidentifikasi pola, hubungan, maupun struktur tersembunyi pada data yang digunakan. Dalam konteks *big data analytics*, terdapat beberapa kerangka kerja (*framework*) yang umum dijadikan acuan dalam proses *data mining*. Tiga di antaranya yang paling banyak digunakan adalah CRISP-DM (*Cross-Industry Standard Process for Data Mining*), KDD (*Knowledge Discovery in Databases*), dan SEMMA (*Sample, Explore, Modify, Model, Assess*).

Masing-masing kerangka kerja memiliki tahapan, pendekatan, fokus analisis, serta karakteristik yang berbeda. Perbedaan ini dapat memengaruhi proses penelitian mulai dari persiapan data, pemodelan, evaluasi, hingga

implementasi hasil. Oleh karena itu, pemilihan metode yang tepat perlu disesuaikan dengan kebutuhan, tujuan, dan karakteristik data dalam penelitian yang dilakukan.

Tabel 2.2 Perbandingan Kerangka Kerja

Aspek	Kerangka Kerja <i>Data Mining</i>		
	CRISP-DM[60]	KDD [61]	SEMMA [62]
Tahapan	Terdiri dari 6 tahapan, yaitu: <i>Business understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment</i>	Melalui 5 tahapan, yaitu: <i>Selection, Preprocessing, Transformation, Data Mining, dan Evaluation</i>	Melewati 5 tahapan, yaitu: <i>Sample, Explore, Modify, Model, Assess.</i>
Fokus	CRISP-DM berfokus dari pemahaman bisnis hingga penerapan model secara lengkap.	KDD berfokus pada eksplorasi data dan penemuan dalam penelitian.	SEMMA berfokus pada pemilihan, eksplorasi, permodelan, dan evaluasi model.
Tingkat Detail	Detail dan lebih terstruktur, dengan penekanan pada pengelolaan proyek dan pemahaman bisnis.	Lebih berorientasi pada langkah-langkah teknis data mining.	Cukup terstruktur dengan fokus pada tahapan praktis dalam analisis data.
Fleksibilitas	Fleksibel, dapat disesuaikan dengan kebutuhan proyek yang berbeda.	Kurang fleksibel, lebih terfokus pada data mining.	Fleksibel, tetapi lebih teknis dan fokus pada model.
Tingkat Penggunaan	Sangat populer dalam berbagai aplikasi data mining dan analisis data.	Lebih digunakan di akademik dan penelitian.	Lebih populer dalam industri, terutama untuk aplikasi data mining praktis.

Berdasarkan perbandingan pada tabel 2.2, metode CRISP-DM dipilih karena memiliki tahapan yang lebih lengkap, terstruktur, dan sistematis dibandingkan KDD maupun SEMMA. CRISP-DM tidak hanya berfokus pada proses teknis *data mining*, tetapi juga menekankan pentingnya pemahaman bisnis (*business understanding*) serta tahap *deployment*, yang tidak dibahas sedetail pada dua framework lainnya. Struktur enam tahapannya mulai dari

bandingkan KDD yang lebih berorientasi pada aspek lebih fokus pada pemodelan praktis, CRISP-DM antara kebutuhan analitis dan kebutuhan implementasi. CRISP-DM menjadi pilihan yang paling sesuai untuk dalam menyelesaikan permasalahan deteksi *misclassification* yang membutuhkan alur kerja yang jelas, terstruktur secara langsung pada skenario nyata.

Industry Standard Process for Data Mining adalah satu kerangka kerja yang digunakan dalam berbagai analisis data [63]. CRISP-DM dikembangkan bertujuan untuk menyediakan panduan terstruktur bagi berbagai industri untuk mengeksplorasi dan menganalisis data. CRISP-DM menawarkan langkah-langkah yang dapat diulang dan diperbaiki selama proses.

CRISP-DM menawarkan langkah-langkah yang dapat diulang dan diperbaiki selama proses.

```

graph TD
    BU[Business Understanding] --> DU[Data Understanding]
    DU --> DP[Data Preparation]
    DP --> M[Modeling]
    M --> E[Evaluation]
    E --> D[Deployment]
    D --> BU
    E --> BU
    E --> DU
    Data[(Data)] --- DU
    Data --- DP
    Data --- M
    Data --- E
    
```

19

Sumber: DMME *CRISP-DM* model [64]

Gambar 2.2 merupakan penjelasan dari 6 fase *CRISP-DM* yang meliputi tahap *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, dan *deployment*.

2.3.2 Algoritma

1) Perbandingan Algoritma

Dalam penelitian deteksi *microsleep* berbasis *computer vision*, pemilihan algoritma *deep learning* yang tepat merupakan langkah penting untuk memastikan akurasi tinggi dan kemampuan model dalam mengenali pola visual secara konsisten. Oleh karena itu, dilakukan perbandingan antara tiga algoritma yang umum digunakan dalam pengolahan data visual, yaitu *Convolutional Neural Network* (CNN), *Artificial Neural Network* (ANN), dan *K-Nearest Neighbor* (KNN). Masing-masing algoritma memiliki karakteristik, kompleksitas, dan kemampuan ekstraksi fitur yang berbeda, sehingga analisis komparatif diperlukan untuk menentukan algoritma yang paling sesuai dengan kebutuhan penelitian, khususnya dalam mendeteksi pola perilaku pengemudi yang mengarah pada *microsleep*.

Tabel 2.3 Perbandingan Algoritma

Aspek	Algoritma <i>Deep learning</i>		
	CNN [65]	ANN [66]	KNN [67]
Jenis <i>Deep learning</i>	<i>Deep learning</i> (untuk data gambar dan spasial)	<i>Deep learning (feedforward neural networks)</i>	Machine Learning (<i>non-deep learning</i>)
Parameter model	Banyak parameter (lapisan, filter, kernel)	Parameter: lapisan, neuron, fungsi aktivasi	Tidak ada pelatihan eksplisit (parameter k, jarak)
Skalabilitas	Sangat skalabel untuk data besar	Skalabel, tapi lebih lambat pada data besar	Tidak skalabel untuk dataset besar
Kompleksitas Waktu	Waktu pelatihan tinggi, cepat	Waktu pelatihan sedang,	Waktu prediksi

Aspek	Algoritma Deep learning		
	CNN [65]	ANN [66]	KNN [67]
	setelah pelatihan	tergantung kedalaman	lambat pada data besar
Kemampuan Ekstraksi Fitur	Ekstraksi otomatis dari gambar	Memerlukan ekstraksi fitur manual	Tidak ada ekstraksi fitur, berbasis jarak
Overfitting	Rentan, tapi dapat diatasi dengan regulasi	Rentan, bisa diatasi dengan regulasi	Risiko rendah, sensitif terhadap noise
Robustness terhadap Noise	Tahan terhadap noise jika dilatih dengan baik	Sensitif terhadap noise tanpa regulasi	Sangat sensitif terhadap noise

Berdasarkan hasil perbandingan pada tabel 2.3, dapat disimpulkan bahwa CNN merupakan algoritma yang paling tepat untuk penelitian ini. CNN memiliki kemampuan ekstraksi fitur otomatis pada data visual, sangat skalabel untuk dataset besar, serta mampu menghasilkan representasi spasial yang lebih akurat dibanding ANN dan KNN. Selain itu, CNN terbukti lebih *robust* terhadap *noise* ketika dilatih dengan baik, menjadikannya ideal untuk proses deteksi *microsleep* yang membutuhkan analisis frame per frame dari video pengemudi. Dengan karakteristik tersebut, CNN menjadi fondasi utama dalam arsitektur *Hybrid CNN-LSTM* yang digunakan dalam penelitian ini untuk memahami baik fitur visual maupun dinamika temporal perilaku pengemudi.

Dalam penelitian yang berfokus pada deteksi *microsleep* berbasis *computer vision*, pemilihan algoritma *hybrid* menjadi bagian penting untuk memastikan model tidak hanya mampu mengekstraksi fitur visual secara efektif tetapi juga memahami pola temporal dari pergerakan mata atau wajah pengemudi. CNN berperan sebagai ekstraktor fitur spasial dari gambar, sehingga diperlukan algoritma pendamping yang mampu memproses dinamika waktu secara optimal. Terdapat beberapa algoritma yang umum diintegrasikan dengan CNN, seperti LSTM, GRU, dan 1D-CNN Temporal, yang masing-masing memiliki karakteristik, kelebihan, serta keunggulan teknis yang berbeda. Oleh karena itu, tabel berikut

disusun untuk memberikan gambaran perbandingan ketiga algoritma tersebut sebagai dasar pemilihan metode hybrid yang paling sesuai untuk mendukung tujuan penelitian ini.

Tabel 2.4 Perbandingan Algoritma *Hybrid* Berbasis CNN

Aspek	Algoritma <i>Hybrid</i> Berbasis CNN		
	CNN-LSTM[68]	CNN-GRU [69]	CNN-BiLSTM [70]
Jenis Arsitektur	<i>Hybrid</i> CNN untuk ekstraksi fitur + LSTM untuk pemrosesan urutan (<i>sequence</i>).	<i>Hybrid</i> CNN + GRU (varian LSTM yang lebih ringan).	<i>Hybrid</i> CNN + BiLSTM yang membaca urutan dua arah.
Kemampuan Sequence Learning	Baik, mampu menangkap pola waktu jangka panjang.	Cukup baik, tetapi sedikit kurang akurat dibanding LSTM pada pola kompleks.	Sangat baik, karena memproses <i>sequence</i> dari forward dan backward.
Kompleksitas Model	Kompleks, memerlukan waktu training menengah-tinggi.	Lebih sederhana dari LSTM (lebih sedikit <i>gate</i>).	Paling kompleks karena dua arah (dua LSTM).
Kebutuhan Komputasi	Tinggi, tetapi efisien untuk data video/temporal.	Lebih rendah dibanding LSTM.	Tertinggi karena dua kali pemrosesan <i>sequence</i> .
Kemampuan	Baik, stabil untuk berbagai jenis data temporal.	Baik, tetapi rentan kehilangan	Sangat baik, tetapi berisiko <i>overfitting</i> .

Aspek	Algoritma <i>Hybrid</i> Berbasis CNN		
	CNN-LSTM[68]	CNN-GRU [69]	CNN-BiLSTM [70]
Generalisasi		detail urutan panjang.	
Akurasi pada Data Gambar	Tinggi, terbukti efektif di banyak studi deteksi perilaku/gestur.	Tinggi, namun biasanya sedikit di bawah LSTM.	Tinggi, sering lebih tinggi dari LSTM, tetapi bergantung dataset.

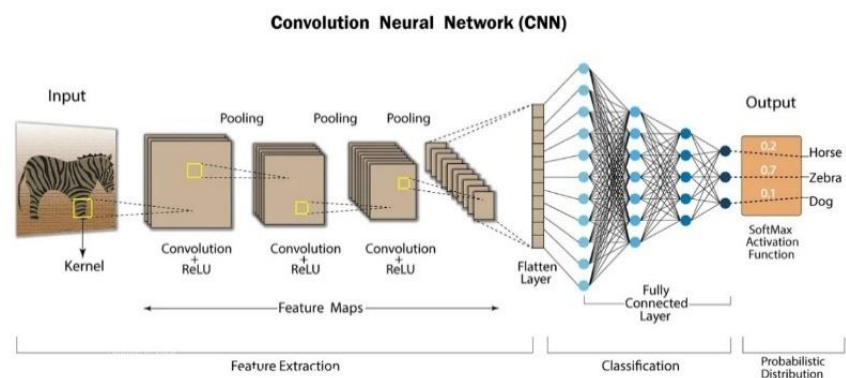
Berdasarkan tabel perbandingan 2.4, CNN-LSTM menjadi algoritma *hybrid* yang paling seimbang dari aspek akurasi, kemampuan menangkap pola temporal, serta efisiensi komputasi. CNN memiliki kemampuan ekstraksi fitur visual yang optimal untuk mendeteksi karakteristik wajah, sedangkan LSTM sangat efektif dalam mempelajari perubahan ekspresi dari waktu ke waktu termasuk pola *microsleep* yang terjadi secara cepat dan bertahap. Dibandingkan CNN-GRU, model CNN-LSTM lebih mampu menangkap urutan jangka panjang yang penting dalam mendeteksi penutupan mata yang progresif. Sementara CNN-BiLSTM memang menawarkan akurasi tinggi, namun kompleksitas dan kebutuhan komputasinya jauh lebih besar, sehingga kurang efisien untuk implementasi *real-time*. Dengan pertimbangan tersebut, CNN-LSTM menjadi pilihan yang optimal untuk penelitian ini.

Dengan mempertimbangkan keseimbangan antara performa, kompleksitas, dan kemampuan dalam mempelajari pola temporal, CNN-LSTM dipilih sebagai algoritma *hybrid* yang digunakan dalam penelitian skripsi ini. Arsitektur ini dinilai paling sesuai untuk mendeteksi perubahan kecil pada ekspresi wajah dan pola pergerakan mata yang menjadi indikator *microsleep*. Pemilihan ini sejalan dengan tujuan penelitian untuk menghasilkan sistem deteksi *microsleep* yang akurat,

responsif, dan dapat diterapkan pada skenario dunia nyata bagi pengendara mobil.

a) *Convolutional Neural Network*

Convolutional Neural Network (CNN) adalah jenis jaringan saraf tiruan yang populer dan efektif untuk pemrosesan data gambar atau video [71]. CNN bertujuan untuk mengenali pola visual dengan meniru cara sistem visual manusia bekerja. Algoritma ini menggunakan mekanisme *konvolusi* untuk mengekstrak fitur seperti tepi, sudut, tekstur, dan pola kompleks lainnya dari data yang dimasukkan [72].



Gambar 2.3 Arsitektur CNN

Sumber: CNN Architecture [73]

Gambar 2.3 di atas menggambarkan arsitektur umum CNN yang terdiri dari dua proses utama: *feature extraction* dan *classification*. Berikut adalah penjelasan setiap bagian:

a. *Input Layer*

Layer ini menerima data mentah, seperti gambar input berupa gambar zebra. Dalam konteks CNN, gambar diproses sebagai matriks piksel yang merepresentasikan intensitas warna atau nilai RGB. Kernel (filter) digunakan untuk memindai gambar input dalam proses konvolusi.

b. *Convolution Layer*

Lapisan ini berfungsi untuk mendeteksi fitur lokal dengan menggunakan kernel. Dot product antara kernel dan bagian gambar dihitung dan diekstrak menjadi informasi penting. Aktivasi fungsi seperti ReLU (*Rectified Linear Unit*) digunakan untuk meningkatkan *non-linearitas*. ReLU didefinisikan sebagai berikut:

$$f(x) = \max(0, x) \quad (2.1)$$

Fungsi ini memetakan nilai negatif ke nol dan mempertahankan nilai positif.

c. *Pooling Layer*

Layer ini diterapkan untuk mengurangi dimensi data (*down-sampling*) dengan mempertahankan fitur utama sambil mengurangi jumlah parameter, sehingga mengurangi kompleksitas komputasi. Dua jenis *pooling* umum adalah *max pooling* (mengambil nilai maksimum) dan *average pooling* (mengambil rata-rata nilai). Proses matematisnya adalah:

$$P(i, j) = \max\{O(x, y)\} \quad (2.2)$$

untuk semua $(x, y) \in \text{area pooling}$

Dimana:

$P(i, j)$: nilai *output* pada lokasi *pooling* i, j .

$O(x, y)$: nilai elemen pada area *pooling* di *feature map*.

d. *Fully Connected Layer*

Lapisan ini bertindak seperti *multilayer perceptron* biasa, menghubungkan semua neuron dari lapisan sebelumnya untuk menghasilkan keputusan. Proses ini mengintegrasikan fitur yang diekstraksi untuk menghasilkan output yang mendekati probabilitas prediksi. Nilai output dihitung sebagai:

$$z_j = \sum_{i=1}^n w_{ij} \cdot x_i + b_j \quad (2.3)$$

z_j : nilai *output* neuron ke- j .

w_{ij} : bobot yang menghubungkan neuron i ke neuron j .

x_i : nilai *input* ke neuron ke- i .

b_j : bias untuk neuron ke- j .

n : jumlah input.

Fungsi aktivasi, seperti *Softmax*, kemudian digunakan untuk mengubah *output* menjadi probabilitas.

e. *Output Layer*

Lapisan akhir ini menghasilkan prediksi probabilitas untuk setiap kelas, seperti pada gambar, yang menunjukkan kemungkinan untuk zebra, kuda, dan anjing menggunakan fungsi aktivasi *Softmax* untuk mengubah skor menjadi distribusi probabilitas. Fungsi *Softmax* menghitung distribusi probabilitas dari *output*:

$$P(y = k | x) = \frac{e^{z_k}}{\sum_{j=1}^n e^{z_j}} \quad (2.4)$$

Dimana:

z_k : skor logit untuk kelas ke- k .

$P(y = k | x)$: probabilitas prediksi untuk kelas ke- k .

n : jumlah kelas.

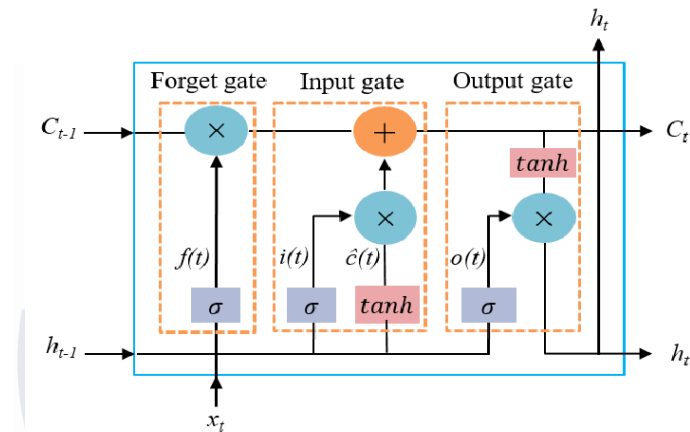
Softmax memastikan semua nilai probabilitas berada dalam rentang $[0, 1]$ dan totalnya sama dengan 1.

CNN menggunakan operasi matematis konvolusi, aktivasi, *pooling*, dan klasifikasi untuk memproses data *input* hingga menghasilkan prediksi akhir yang didesain agar dapat mengekstraksi fitur lokal, mengurangi dimensi data, dan memprediksi hasil dengan tingkat akurasi tinggi.

b) *Long Short-Term Memory*

Long Short-Term Memory (LSTM) adalah jenis *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi masalah *vanishing gradient* dan *long-term dependency*. Dalam penelitian deteksi *microsleep*, LSTM digunakan untuk memahami urutan *frame video*, yaitu bagaimana pola kantuk berkembang dari waktu ke waktu—hal

yang tidak dapat dilakukan CNN saja. Untuk mencapai kemampuan ini, LSTM menggunakan struktur khusus berupa *cell state* dan tiga gerbang utama, yaitu *Forget Gate*, *Input Gate*, dan *Output Gate*. Setiap *gate* bekerja menggunakan fungsi *sigmoid* (σ) dan operasi titik (*element-wise*).



Gambar 2.4 Arsitektur LSTM

Sumber: LSTM Architecture [74]

a. *Forget Gate*

Forget Gate bertugas memutuskan informasi mana yang akan dibuang dari *cell state* sebelumnya (C_{t-1}). *Gate* ini penting untuk “melupakan” informasi yang tidak relevan sehingga model tidak menyimpan pola yang salah.

b. *Input Gate*

Input Gate menentukan informasi baru apa yang perlu disimpan ke *cell state*.

Input Gate memiliki dua bagian:

- *Gate sigmoid* (i_t) $\rightarrow$ menentukan yang boleh masuk
- *Candidate state* ($\hat{C}_t$) $\rightarrow$ berisi informasi baru calon memori

c. *Output Gate*

Output Gate menentukan apa yang akan dikirim sebagai *output* pada waktu t dan menjadi *hidden state* berikutnya.

c) **CNN-LSTM Hybrid Model**

Model CNN–LSTM merupakan arsitektur *hybrid* yang menggabungkan dua pendekatan deep learning, yaitu *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM), yang masing-masing memiliki keunggulan dalam memproses jenis informasi yang berbeda [75]. CNN digunakan untuk mengekstraksi fitur visual dari setiap *frame* video, seperti bentuk mata, frekuensi kedipan, arah pandangan, atau perubahan posisi kepala. Sifat CNN yang unggul dalam menangkap pola spasial dari gambar membuatnya sangat ideal untuk mendeteksi perubahan visual yang merupakan indikator awal *microsleep*. Hasil ekstraksi fitur dari CNN kemudian tidak langsung digunakan sebagai prediksi akhir, tetapi menjadi input bagi LSTM untuk dianalisis secara berurutan.

Di sisi lain, LSTM berfungsi untuk memodelkan pola temporal atau hubungan antar *frame* dari waktu ke waktu. Hal ini penting karena *microsleep* bukan hanya terlihat dari satu *frame* tunggal, tetapi dari rangkaian pergerakan wajah yang menunjukkan kelelahan progresif, seperti mata yang menutup lebih lama, kelopak mata yang berat, atau kepala yang perlahan turun. Dengan menggunakan memori jangka panjang dan mekanisme gerbang (*gates*), LSTM mampu mengingat atau melupakan informasi tertentu secara selektif, sehingga model dapat lebih akurat dalam membedakan antara kedipan normal dan tanda-tanda *microsleep* yang lebih berbahaya. Kombinasi kedua algoritma ini memungkinkan sistem untuk tidak hanya melihat apa yang terjadi pada wajah pengemudi, tetapi juga bagaimana perubahannya berlangsung dari waktu ke waktu.

Secara keseluruhan, arsitektur *hybrid* CNN–LSTM memberikan kemampuan analisis visual yang lebih komprehensif dibandingkan penggunaan CNN atau LSTM secara terpisah. CNN memastikan fitur spasial diekstraksi secara optimal, sementara LSTM memastikan bahwa dinamika temporal dipertimbangkan dengan baik. Pendekatan ini sesuai dengan tujuan penelitian, yaitu mendeteksi *microsleep* dari rekaman video *real-time* secara akurat dan responsif. Dengan memanfaatkan

kekuatan kedua model ini, sistem dapat memberikan deteksi lebih dini dan lebih tepat terhadap kondisi *microsleep*, yang pada akhirnya bertujuan untuk meningkatkan keselamatan pengemudi dan mengurangi risiko kecelakaan akibat kelelahan.

2.4 Alat Penelitian

1) *Python*

Dalam proses pengembangan sistem deteksi *microsleep* berbasis *computer vision* dan *deep learning*, pemilihan *tools* pemrograman memiliki peran penting dalam memastikan sistem dapat dibangun secara efisien, fleksibel, dan akurat. Terdapat dua *tools* yang umum digunakan dalam analisis data dan pengembangan model *machine learning*, yaitu Python dan R. Keduanya memiliki karakteristik dan fokus penggunaan yang berbeda, sehingga diperlukan perbandingan untuk menentukan *tools* yang paling sesuai dengan kebutuhan penelitian ini. Tabel 2.5 berikut merupakan perbandingan antara Python dan R berdasarkan beberapa aspek penting.

Tabel 2.5 Perbandingan Alat Penelitian

Aspek	Tools	
	<i>Python</i> [76]	R [77]
Definisi & Tujuan	Bahasa pemrograman serbaguna untuk pengembangan aplikasi, data science, dan AI.	Bahasa pemrograman khusus untuk analisis statistik dan visualisasi data.
Sintaks & Tingkat Kesulitan	Sintaks sederhana dan mudah dipahami, cocok untuk pemula.	Sintaks lebih teknis dan kompleks, lebih sulit dipahami bagi pemula.
Objektif penggunaan	Digunakan untuk pengembangan aplikasi, machine learning, <i>deep learning</i> , dan automasi.	Digunakan untuk analisis statistik, pemodelan data, dan visualisasi data.
Kinerja	Cukup cepat dengan <i>library</i> optimasi seperti NumPy dan Pandas, tetapi lebih lambat daripada C++.	Cepat dalam analisis statistik, namun bisa lebih lambat pada dataset besar dibandingkan <i>Python</i> .
Integrasi	Mudah diintegrasikan dengan berbagai <i>library</i> eksternal dan sistem lain.	Terfokus pada analisis data dan visualisasi, dengan integrasi terbatas pada aplikasi non-statistik.

Berdasarkan perbandingan di atas, *Python* dipilih sebagai *tools* utama dalam penelitian ini karena memiliki sintaks yang sederhana, mudah dipahami, serta menawarkan ekosistem *library* yang sangat lengkap untuk pengembangan deep learning seperti TensorFlow, Keras, dan PyTorch. *Python* juga lebih unggul dalam integrasi dengan sistem *computer vision* melalui *library* seperti *OpenCV*, menjadikannya ideal untuk membangun model *hybrid* CNN-LSTM yang digunakan dalam mendeteksi *microsleep* pada pengemudi mobil. Selain itu, kemampuannya dalam menangani *dataset* berukuran besar dan fleksibilitas dalam pengembangan aplikasi membuat *Python* menjadi pilihan yang paling tepat untuk mendukung kebutuhan penelitian ini.

Python adalah bahasa pemrograman tingkat tinggi yang banyak digunakan dalam penelitian dan pengembangan sistem berbasis *machine learning* serta *deep learning* [78]. Popularitas *Python* didukung oleh sintaksnya yang sederhana dan mudah dipahami, sehingga meningkatkan keterbacaan kode dan memudahkan proses pengembangan. Bahasa ini juga mendukung berbagai paradigma pemrograman seperti prosedural, imperatif, objektif, dan fungsional, sehingga memberikan fleksibilitas bagi pengembang untuk memilih pendekatan yang paling sesuai dengan kebutuhan sistem. Kemampuan ini menjadikan *Python* sangat efektif dalam membangun solusi kompleks dengan struktur kode yang tetap ringkas, efisien, dan mudah dikelola.

Dalam bidang *image processing*, *Python* telah menjadi standar utama karena menyediakan berbagai *library* yang kuat dan komprehensif, seperti *OpenCV*, *Pillow*, dan *scikit-image* [79]. Ketiga *library* tersebut memungkinkan manipulasi dan analisis gambar secara langsung, mulai dari operasi dasar seperti *resize*, *cropping*, dan konversi warna, hingga pemrosesan lanjutan seperti *edge detection*, *filtering*, *object tracking*, dan *feature extraction*. *OpenCV* merupakan *library* yang paling luas digunakan dan menyediakan fungsionalitas untuk pengenalan wajah, deteksi titik wajah, transformasi gambar, segmentasi objek, dan pengolahan video *real-time*. Kemampuan ini

sangat relevan dalam penelitian yang berkaitan dengan *computer vision* dan pemantauan perilaku pengemudi [80].

Python juga memiliki keunggulan melalui integrasinya dengan ekosistem *machine learning* dan *deep learning* yang sangat matang. *Library* seperti *TensorFlow*, *Keras*, dan *PyTorch* mendukung pembangunan arsitektur *neural network*, termasuk model *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM). Dengan adanya *library* tersebut, peneliti dapat merancang pipeline yang lengkap mulai dari *pre-processing* gambar, ekstraksi fitur, pembangunan model, pelatihan, hingga evaluasi performa, dalam satu lingkungan kerja yang konsisten. *Python* turut menyediakan *library* pendukung seperti *NumPy* dan *Pandas* untuk pengolahan data numerik serta *Matplotlib* dan *Seaborn* untuk visualisasi hasil pelatihan dan evaluasi model.

Selain itu, *Python* mampu berintegrasi dengan akselerasi perangkat keras seperti GPU dan TPU melalui *library* seperti *CUDA*, *cuDNN*, dan *TensorFlow-GPU*. Dukungan tersebut memastikan proses pelatihan model *deep learning* dapat berlangsung lebih cepat, terutama ketika bekerja dengan dataset berukuran besar atau arsitektur yang kompleks. Dengan seluruh ekosistem yang tersedia, *Python* menjadi alat penelitian yang sangat kuat dan komprehensif. *Python* banyak digunakan dalam pengembangan sistem deteksi objek, klasifikasi gambar, segmentasi gambar, dan deteksi *microsleep* berbasis *computer vision* karena kemampuannya dalam menangani pengolahan visual yang intensif dan pemodelan *deep learning* yang mendalam.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A