SISTEM DETEKSI *DRIVER DROWSINESS* MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK (CNN)*

SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan

Pendidikan Diploma Empat (D-4) Program Studi Teknik Komputer dan Jaringan

Jurusan Teknik Elektro

Politeknik Negeri Ujung Pandang

AURELIA VITANIA RUSLI

425 18 059

PROGRAM STUDI D4 TEKNIK KOMPUTER DAN JARINGAN

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI UJUNG PANDANG

MAKASSAR

2022

HALAMAN PENGESAHAN

Skripsi ini dengan judul "**Sistem Deteksi Driver Drowsiness Menggunakan Algoritma Convolutional Neural Network (CNN)**" oleh **Aurelia Vitania Rusli** NIM **425 18 059** telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D4-/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Makassar, 27 September 2022

Mengesahkan,

Pembimbing I,	Pembimbing II,
Meylanie Olivya, S.T., M.T.	Irmawati, S.T., M.T.
NIP. 198205032014042002	NIP. 197811242012122002

Mengetahui,
Koordinator Program Studi
Teknik Komputer dan Jaringan
Politeknik Negeri Ujung Pandang

Eddy Tungadi, S.T., M.T.
NIP 197908232010121001

HALAMAN PENERIMAAN

Pada hari ini, 27 September 2022, Tim Penguji Ujian Sidang Skripsi telah menerima dengan baik hasil ujian sidang skripsi oleh mahasiswa: **Aurelia Vitania Rusli NIM 425 18 059** dengan judul **SISTEM DETEKSI DRIVER DROWSINESS MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN).**

Makassar, 27 September 2022

Tim Penguji Ujian Sidang Skripsi:

1. Rini Nur, S.T., M.T.	Ketua	(.....)
2. Muhammad Nur Yasir Utomo,S.ST.,M.Eng	Sekertaris	(.....)
3. Sulaeman, S.T., M.T.	Anggota	(.....)
4. Eddy Tungadi, S.T., M.T.	Anggota	(.....)
5. Meylanie Olivya, S.T., M.T.	Anggota	(.....)
6. Irmawati, S.T., M.T.	Anggota	(.....)

KATA PENGANTAR

Puji dan syukur penulis kepada Tuhan Yang Maha Esa, oleh karena anugerah- Nya yang melimpah, kemurahan dan kasih setia yang besar menuntun penulis dalam mengerjakan skripsi ini yang berjudul "Sistem Deteksi *Driver Drowsiness* Menggunakan Algoritma *Convolutional Neural Network (CNN)*". Skripsi ini disusun guna memenuhi salah satu syarat untuk menyelesaikan studi serta dalam rangka memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan JurusanTeknik Elektro Politeknik Negeri Ujung Pandang.

Penulis tentu menyadari bahwa keberhasilan skripsi ini tidak lepas dari bantuan berbagai pihak baik secara langsung maupun tidak langsung. Oleh karenanya, penulis menyampaikan apresiasi dengan menghaturkan terimakasih sebesar- besarnya kepada:

1. Orang tua penulis yang sampai saat ini senantiasa memberikan semangat, dukungan, motivasi, dan dukungan lahir maupun batin serta doa yang tiada henti kepada penulis.
2. Bapak Prof. Ir.Muhammad Anshar, M. Si., Ph.D. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Eddy Tungadi, S.T., M.T. selaku Koordinator Program Studi Teknik Komputer dan Jaringan yang senantiasa memotivasi, menasihati, dan membantu penulis dalam proses belajar.
5. Ibu Meylanie Olivya, S.T., M.T., selaku dosen pembimbing I yang telah memberikan saran, kritik, bantuan dan arahan selama penulis menyusun dan menyelesaikan skripsi ini. Terima kasih atas waktu dan pikiran yang telah diberikan untuk membimbing penulis.
6. Ibu Irmawati ,S.T., M.T, selaku dosen pembimbing II yang telah dengan sabar meluangkan waktu untuk memberikan ilmu, motivasi, nasihat

   bantuan, dan bimbingan terbaik kepada penulis selama mengerjakan penelitian.
7. Seluruh dosen dan Staf Jurusan Teknik Elektro, khususnya Program Studi D4 Teknik Komputer dan Jaringan yang telah memberikan bekal ilmu kepada penulis.

8. Teman-teman seperjuangan di Program Studi Teknik Komputer dan Jaringan Angkatan 2018 yang telah menemani berjuang dalam meraih gelarS.Tr.T.
9. Saudari penulis yakni Anastasia Brigita yang telah dengan sangat sabar dalam memberikan bantuan, motivasi, nasihat, kritikan dan saran selama mengerjakan skripsi.
10. Semua pihak yang telah membantu penulis baik secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu.

Semoga Tuhan Yang Maha Esa senantiasa melimpahkan rahmat dan hidayah- Nya selalu yang telah diberikan kepada penulis menjadi berkat bagi kita semua. Penulis masih memiliki banyak kekurangan pengetahuan dan pengalaman pada topik yang diangkatdalam skripsi ini, begitu pula dalam penulisannya yang masih banyak terdapat kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang sifatnya membangun di masa mendatang. Semoga skripsi ini dapat bermanfaat bagi pembacamaupun bagi penulis sendiri.

Makassar, 27 September 2022

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Aurelia Vitania Rusli

NIM: 42518059

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul "**SISTEM DETEKSI DRIVER DROWSINESS MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)**" merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut diatas tidak benar, saya siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 27 September 2022

Aurelia Vitania Rusli

NIM. 42518059

SISTEM DETEKSI DRIVER DROWSINESS MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)

RINGKASAN

Setiap tahun kecelakaan lalu lintas meningkat secara terus- menerus. Salah satu penyebab kecelakaan adalah pengemudi yang mengatuk. Untuk meminimalisir terjadinya kecelakaan, dibutuhkan suatu sistem *driver drowsiness* yang menerapkan metode *machine learnin*g untuk memprediksi kondisi pengemudi. Kondisi tersebutdapat digunakan sebagaiinformasi yang akan meningkatkan keamanan saat berkendara. Penelitian ini menerapkan algoritma *Convolutional Neural Network* (CNN) untuk membuat model klasifikasi. Hasil dari penelitian ini di dapatkan performa akurasi setelah dilakukan validasi sebesar 94,51%. Jarak wajah terhadap *webcam* yang maksimal yaitu antara 40 cm sampai dengan 60 cm. kemudian waktu *delay* pembacaan mata pada jarak 40 cm memiliki nilai rata-rata yaitu 0.2682 detik. Pada jarak 50 cm memiliki nilai rata-rata yaitu 0.4662 detik. Sedangkanpada jarak60 cm memiliki nilai rata-rata yaitu 0.4662 detik.

Kata Kunci: *Drowsiness*, *Machine Learning*, CNN (*Convulutional NeuralNetwork*)

BAB I PENDAHULUAN

1.1 Latar Belakang

Setiap tahun banyak orang kehilangan nyawa karena kecelakaan lalu lintas yang fatal di seluruh dunia dan mengemudi dalam keadaan mengantuk adalah salah satu penyebab utama kecelakaan dan kematian di jalan. Kelelahan dan *micro sleep* yang ada pada pengemudi sering menjadi penyebab kecelakaan serius. Oleh karena itu, dibutuhkan suatu sistem untuk mendeteksi kelelahan pengemudi sebelum situasi kritis muncul (Mehta et al., 2019). Menurut data Kepolisian, di Indonesia, rata-rata tiga orang meninggal setiap jam akibat kecelakaan jalan. Data tersebut juga menyatakan bahwa besarnya jumlah kecelakaan tersebut disebabkan oleh beberapa hal, yaitu : 61 % kecelakaan disebabkan oleh faktor manusia yaitu yang terkait dengan kemampuan serta karakter pengemudi, 9 % disebabkan karena faktor kendaraan (terkait dengan pemenuhan persyaratan teknik laik jalan) dan 30 % disebabkan oleh faktor prasarana dan lingkungan (*Kementerian Komunikasi Dan Informatika*, n.d.)

Hampir setiap kendaraan modern baru di desain untuk kenyamanan pengemudi yang dapat menyebabkan meningkatnya rasa ngantuk pengemudi terutama pada rute yang panjang (Janjua & Safdar, 2021). Karena bahaya yang ditimbulkan akibat mengantuk di jalan, maka suatu metode perlu dikembangkan untuk mengurangi akibat yang ditimbulkan. Pengemudi yang tidak berhati-hati mungkin merupakan hasil dari kurangnya kewaspadaan saat mengemudi, karena pengemudi mengantuk dan tidak fokus (Saini & Saini, n.d.)

Dalam penelitian dan pengembangan, metode *machine learnin*g telah digunakan untuk memprediksi kondisi pengemudi. Kondisi tersebut dapat digunakan sebagai informasi yang akan meningkatkan keamanan jalan. *Machine learning* juga telah membawa perkembangan dalam pengolahan video yang memungkinkan gambar untuk dianalisis dengan akurasi. Untuk mendeteksi *drowsiness*, menggunakan *Convolution Neural Network* diterapkan untuk klasifikasi mata yang mendeteksi *driver drowsiness* dengan mempertimbangkan kedipan mata (Pachouly et al., 2020) dan sistem *OpenCV* dikembangkan untuk aplikasi penglihatan komputer secara *real-time* yang berfokus

pada gambar, video pengolahan dan analisis di wajah dan fitur deteksi objek (Saranya et al., 2021).

Berdasarkan uraian masalah diatas, pada penelitian ini bertujuan untuk membangun sebuah sistem deteksi dengan menggunakan penerapan algoritma *machine learning* dan beberapa penggunaan *tool* yaitu *Python*, *OpenCV* yang menggunakan model *deep learning* untuk mendeteksi wajah dan mata pengemudi , dan menggunakan model *Convolutional Neural Network* (CNN) untuk membuat model klasifikasi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, perumusan masalah yang didapat adalah bagaimana sistem ini akan memperingatkan pengemudi ketika kantuk terdeteksi.

1.3 Tujuan

Adapun tujuan penelitian ini sebagai berikut.

1. Dapat membuat sistem deteksi *driver drowsiness* dengan menerapkan algoritma CNN.
2. Dapat mengetahui performa akurasi dari algoritma yang digunakan.
3. Dapat mengetahui jarak maksimal wajah dan pembacaan waktu delay mata terhadap *webcam.*

1.4 Ruang Lingkup Penelitian

Penelitian ini memiliki batasan-batasan sebagai berikut:

1. Penelitian ini hanya membahas cara membangun sistem deteksi kantuk menggunakan penerapan algoritma *machine learning*.
2. Sistem hanya mendeteksi pada objek wajah
3. Sistem pendeteksi hanya mendeteksi kantuk di tempat yang terang.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Bagi Pengguna, dapat mengurangi terjadinya kecelakaan akibat pengemudi mengantuk.
2. Bagi Peneliti, dapat digunakan untuk menambah wawasan dan referensi terkait sistem deteksi *driver drowsiness*
3. Bagi Institusi Pendidikan, dapat digunakan sebagai bahan acuan atau referensi untuk melakukan penelitian yang berkaitan dengan teknologi *machine learning*.

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam pembuatan sebuah sistem deteksi driver drowsiness dengan menggunakan penerapan algoritma machine learning, penulis menggunakan beberapa teori atau penelitian penunjang sebagai berikut.

Tabel 2. 1 Penelitian Terkait

Judul Penelitian	Deskripsi
Deteksi Mata Mengantuk Berbasis Citra Digital Menggunakan Metode *Haar Classifier* Secara *Real Time* (Alfari, 2013)	Penelitian ini menjelaskan tentang perangkat lunak berbasis pengolahan citra yang dapat dijadikan alat untuk pendeteksian mata mengantuk secara *real-time* berdasarkan mata dalam keadaan tertutup. Metode pengolahan citra yang digunakan untuk pendeteksian ini adalah metode *Haar Cascade Classifier* yang menggabungkan empat konsep utama yaitu *training data, Haar like feature, integral image*, dan *cascade classifer*.

Judul Penelitian	Deskripsi
Fuzzy Based Method for Driver Drowsiness Detection (Rigane et al., 2018)	Penelitian ini menjelaskan pendekatan baru untuk intelligent driver drowsiness detection system menggunakan perilaku visual pengemudi. Estimasi kewaspadaan pengemudi berhasil dibuat dengan menggabungkan gejala wajah dan mata menggunakan pengontrol fuzzy logic. Hasil percobaan menggunakan simulasi fuzzy logic di Matlab menunjukkan kinerja pendekatan yang dikembangkan dalam istilah kekuatan dan keandalan.
Development of Vehicle Driver Drowsiness Detection System Using Electrooculogram (EOG) (Chieh et al., 2005)	Penelitian ini menjelaskan tentang penggunaan *electrooculogram* (EOG) sebagai alternatif untuk sistem berbasis video dalam mendeteksi aktivitas mata yang disebabkan oleh mengantuk dievaluasi. EOG yang merupakansinyal listrik yang dihasilkan oleh gerakan mata, diperoleh oleh modul kendali biosignal *mobile* dan diproses secara *offline* menggunakan komputer pribadi. Perbedaan sinyal digital dan teknik *fuzzy* informasi sederhana digunakan untuk mendeteksi tanda- tanda penenggelaman dalam sinyal EOG. Berdasarkan pada *Personal Digital Assistant* (PDA)

Judul Penelitian	Deskripsi
Driver Drowsiness Detection System Based on Binary Eyes Image Data (Kahlon & Ganesan, 2018)	Penelitian ini menjelaskan tentang penggunaan *Matlab* sebagai alat pemrosesan gambar yang digunakan untuk memproses *image* yang disediakan oleh kamera dan membuat objek sistem menggunakan algoritma *Viola_Jones* untuk mendeteksi benda-benda seperti hidung, mulut atau tubuh bagian atas. Setelah menangkap gambar, daerah mata pergei panjang disesuaikan untuk mengurangi kebisingan. *Image-type* filter median digunakan untuk mengurangi kebisingan dan kemudian gambar *smoothened.* Deteksi mengantuk dilakukan berdasarkan kondisi seperti rasio hitam sampai putih piksel, jumlah piksel di kolom lebih besar dari nilai ambang batas dan bentuk mata. Cahaya dan posisi pengemudi memainkan peran penting. Sistem dapat ditata ke self-learn saat awal mula untuk menetapkan nilai *threshold.*

Judul Penelitian	Deskripsi
Driver Drowsiness Detection Using ANN Image Processing (Vesselenyi et al., 2017)	Penelitian ini menjelaskan mengenai kemungkinan untuk mengembangkan sistem deteksi kantuk bagi pengemudi mobil berdasarkan tiga jenis metode: pemrosesan sinyal EEG dan EOG dan analisa gambar pengemudi serta mempelajari kemungkinan untuk mendeteksi keadaaan mengantuk atau waspada pengemudi berdasarkan gambar yang diambil selama mengemudi dan dengan menganalisis keadaan mata pengemudi: terbuka, setengah terbuka dan tertutup. Untuk tujuan ini dua jenis jaringan saraf buatan digunakan: satu jaringan lapisan teersembunyi dan jaringan *autoencoder*.

Judul Penelitian	Deskripsi
Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection (Arefnezhad et al., 2019)	Penelitian ini menjelaskan sebuah metode pemilihan fitur baru untuk merancang sistem deteksi kantuk pengemudi non-invasif berdasarkan data roda kemudi. Pemilih fitur yang diusulkan dapat memilih fitur yang paling terkait ke tingkat kantuk untuk meningkatkan akurasi klasifikasi. Metode ini didasarkan pada kombinasi dari filter dan *wrapper* fitur seleksi algoritma menggunakan *adaptive neuro-fuzzy inference system* (ANFIS). Dalam metode ini pertama, empat indeks filter yang berbeda diterapkan pada fitur yang diekstrak dari data roda kemudi. Setelah itu, nilai keluaran dari setiap indeks filter diimpor sebagai masukan ke *fuzzy inference system* untuk menentukan tingkat penting dari setiap fitur dan memilih fitur yang paling penting. Kemudian, fitur yang dipilih diimpor ke *support vector machine* (SVM) untuk klasifikasi biner untuk mengklasifikasikan kondisi mengemudi dalam dua kelas mengantuk dan terjaga. Akhirnya, akurasi klasifikasi dieksploitasi untuk menyesuaikan parameter dari sistem fuzzy adaptif menggunakan *Particle Swarm Optimization* (PSO) algoritma.

Judul Penelitian	Deskripsi
Intelligent Driver Drowsiness Detection Sysyem Using Uncorrelated Fuzzy Locality Preserving Analysis (Khushaba et al., 2011)	Penelitian ini menjelaskan tentang metode *Uncorrelated Fuzzy Local Preserving Analysis* (UFLPA*).* UFLPA yang diusulkan memanfaatkan perubahan perilaku pengemudi, melalui sinyal *Electroencephalogram* (EEG), *Electrooculogram* (EOG) dan *Electrocardiogram (ECG)* untuk mengekstrak serangkaian fitur yang dapat sangat mendeskriminasikan antara tingkat kantuk yang berbeda. Tidak seperti metode yang ada, UFLPA yang diusulkan mempertimbangkan sifat *fuzzy* dari pengukuran input sementara melestarikan lokal diskriminatif dan *manifold* struktur data. Selain itu, UFLPA juga memanfaatkan *Singular Value Decomposition* (SVD) untuk menghidari masalah singularitas dan menghasilkan satu set fitur yang tidak terkait.

Judul Penelitian	Deskripsi
Driver Drowsiness Detection Using Condition-Adaptive Representation Learning Framework (Yu et al., 2019)	Penelitian ini menjelaskan tentang *condition-adaptive representation learning framework* untuk deteksi kantuk pengemudi berdasarkan *3D-Deep Convolutional Neural Network*. *Framework* yang diusulkan terdiri dari empat model: *spatio-temporal representation learning* mengekstrak fitur yang dapat menggambarkan gerakan dan penampilan dalam video secara bersamaan. *Scene condition understanding* menggolongkan kondisi adegan yang berkaitan dengan berbagai kondisi tentang pengemudi dan situasi mengemudi, seperti kepala, mata, dan mulut. *Feature fusion* menghasilkan kondisi adaptif representasi menggunakan dua fitur diekstrak dari model diatas. Model *drowsiness detection* mengenali status kantuk pengemudi menggunakan *condition-adaptive representation. Condition-adaptive representation learning framework* dapat mengekstrak fitur yang lebih diskriminatif berfokus pada setiap *scene condition* daripada representasi umum sehingga metode deteksi kantuk dapat memberikan hasil yang lebih akurat untuk berbagai situasi mengemudi.

Dalam melakukan perancangan sistem deteksi driver drowsiness, penelitian ini menggunakan OpenCV sebagai pengumpulan gambar dari *webcam*, memasukkan ke dalam model *deep learning* yang akan mengklasifikan apakah mata pengemudi „terbuka‟ atau „tertutup. Model yang digunakan yaitu dibuat dengan Keras menggunakan *Convolutional Neural Network* (CNN). *Convolutional Neural Network* adalah jenis kusus dari deep neural network yang berjalan sangat baik untuk tujuan klasifikasi gambar. CNN pada dasarnya terdiri dari lapisan masukan, lapisan keluaran dan lapisan tersembunyi yang dapat memiliki beberapa nomor lapisan. Operasi pembauran dilakukan pada lapisan-lapisan ini menggunakan filter yang melakukan perkalian 2D lapis dan filter.

2.2 Dasar Teori

2.2.1 Drowsiness

Drowsiness atau *sleepiness* adalah masalah serius yang perlu ditangani untuk peningkatan keselamatan berkendara di jalan (Ahmad Kamran et al., 2019) .Ada beberapa hal yang menjadi penyebab seseorang mengalami *drowsy driving*, diantara yaitu: (*Mengenal Drowsy Driving Plus Berbagai Penyebabnya - Otomotif Tempo.Co*, n.d.)

a. Kondisi tubuh tidak sehat

Saat tubuh tidak sehat berbagai kemungkinan bisa terjadi. Mulai dari pusing bahkan pingsan secara tiba-tiba. Tak hanya itu, kondisi pun akan semakin buruhk jika seseorang berada di bawah pengaruh obat. Karena itu, sebaiknya waspada terhadap kondisi tubuh saat ingin berkendara.

b. Pengaruh minuman keras

Hal ini yang bisa menyebabkan rasa kantuk aadalah adanya pengaruh minuman keras. Pengaruh alkohol atau minuman keras pada setiap orang berbeda-beda, namun pada umunya minuman keras bisa menyebabkan seseorang hilang kesadaran secara tiba-tiba karena mengantuk. Karena itulah, sebaiknya seseorang tidak mengemudi saat di bawah pengaruh minuman keras.

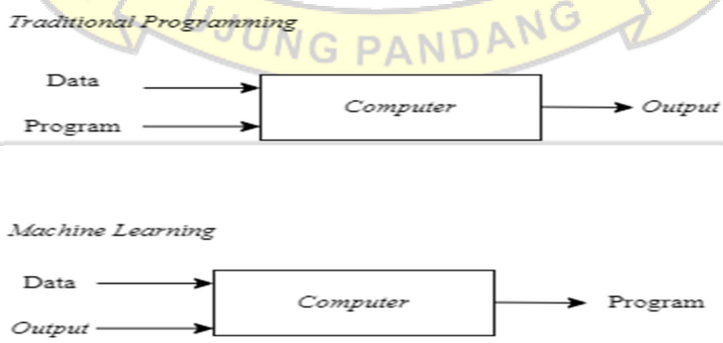
c. Jalan panjang dan lurus

Selain faktor dari tubuh, faktor luar pun berpengaruh, salah satunya saat mengemudi di jalanan yang panjang dan lurus. Jalan di daerah perkotaan tentu akan berbeda dengan jalan di daerah terpencil maupun jalan tol yang sepi. Jalanan seperti ini akan membuat pengemudi mengantuk.

2.2.2 Machine Learning

Machine Learning (ML) atau pembelajaran mesin merupakan pendekatan dalam AI yang banyak digunakan untuk menggantukan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, ML mencoba menirukan bagaimana proses manusia atau mahkluk cerdas belajar dan mengeneralisasi. Setidaknya ada dua aplikasi utama dalam ML yaitu klasifikasi dan prediksi. Ciri khas dari ML adalah adanya proses pelatihan, pembelajaran, atau *training* (Ahmad, n.d.).

Machine learning membantu menangani dan memprediksi data yang sangat besar dengan cara merepresentasikan data-data tersebut dengan algoritma pembelajaran. Machine learning dapat membantu komputer memprogram diri mereka sendiri. Jika pemrograman adalah pekerjaan untuk membuat otomatis, maka *machine learning* mengotomatisasi proses otomatis. Pada dasarnya *machine learning* membiarkan data melakukan pekerjaan. Berikut gambaran umum machine learning dibandingkan dengan pemrograman secara tradisional.



Gambar 2. 1 Perbandingan Pemrograman Tradisional dan *Machine Learning*

Pada gambar diatas dapat diketahui bahwa pemrograman dengan cara tradisional data dan program komputer untuk menghasilakn suatu output. Sedangkan, dengan menggunakan teknik machine learning data dan output dijalankan dengan komputer untuk membuat sebuah prgoram (Putri, 2018).

Pada *machine learning* ada cara yang berbeda untuk mengekstrak informasi dari data, tergantung pada bagaimana algoritmanya dibangun. Umumnya, prosesnya membutuhkan sejumlah besar data yang memberikan respon yang diharapkan diberikan *input* tertentu. Setiap pasangan *input* mewakili sebuah contoh dan lebih banyak contoh memudahkan algoritma untuk dipelajari. Itu karena setiap pasangan *input* cocok dengan garis, *cluster*, atau representasi statistik lain yang mendefinisikan domain masalah. (*Kenali Algoritma Machine Learning Dan Jenisnya - Coding Studio*, n.d.)

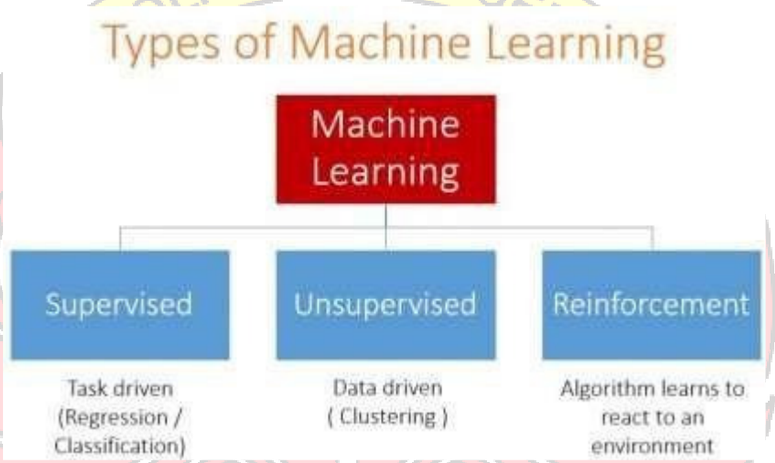
Gambar 2. 2 Cara Kerja Machine Learning

Pada gambar diatas terdapat penjelasan mengenai cara kerja *machine learning* sebagai berikut:

a. *Decision Process*: Secara umum, algoritma *machine learning* digunakan untuk membuat *prediction* atau *classification*. Berdasarkan beberapa data *input*, yang dapat diberi label atau tidak, algoritma akan menghasilkan perkiraan tentang suatu pola dalam data.
b. *Error Function*: Ini berfungsi untuk mengevaluasi prediksi model. Jika ada contoh yang diketahui, *error function* dapat membuat perbandingan untuk menilai keakuratan model.

c. *Model Optimization Process*: Jika model dapat lebih cocok dengan data *point* dalam *training set*, maka bobot disesuaikan untuk mengurangi perbedaan antara contoh yang diketahui dan estimasi model. Algoritma akan mengulangi proses evaluasi dan pengoptimalan ini, memperbarui bobot secara mandiri hingga ambang batas akurasi terpenuhi.

Dalam pembelajaran *machine learning*, terdapat beberapa jenis algoritma *machine learning*:(*Yuk Kenali Macam-Macam Algoritma Machine Learning!*, n.d.)



Gambar 2. 3 Jenis-Jenis Machine Learning (*Mengenal Machine Learning*, n.d.)

1. *Supervised Learning/ Predictive*

Algoritma ini digunakan untuk memprediksi hasil masa depan berdasarkan data historis. Model prediksi biasanya diberi instruksi yang jelas sejak awal seperti apa yang perlu dipelajari dan bagaimana itu perlu dipelajari. Data-data yang digunakan dalam algoritma ini merupakan data berlabel, artinya algoritma mengidentifikasi fitursecara eksplisit dan melakukan prediksi atau klasifikasi yang sesuai. Semakin banyak pelatihan pada algoritma, maka algoritma dapat mengidentifikasi hubungan antara dua variabel sehingga kita dapat memperoleh hasil yang baru dengan data baru. Beberapa jenis algoritma supervised learning tergantung dari tujuan terbentuknya:

a. Regresi Linier

Algoritma ini digunakan untuk mengukur hubungan linier antara dua atau lebih dari dua variabel.

b. Random Forest

Metode pembelajaran ansambel untuk melakukan klasifikasi, regresi, dan tugas lain melalui pembentukan pohon keputusan dan output yang dihasilkan berupa kelas yang merupakan modus atau mean dari pohon individu.

c. Artificial Neural Network

Artificial neural network diibaratkan seperti otak manusia dan mempelajari data dari waktu ke waktu.

2. Unsupervised Learning

Algortima ini tidak menggunakan data label karena algoritma ini mampu belajar dari data dengan menemukan pola implisit. Algoritma *unsupervised learning* mengidentifikasi data berdasarkan kepadatan, struktur, segmen serupa, dan fitur serupa lainnya. Beberapa algoritma yang digunakan dalam *unsupervised learning* yaitu:

a. Clustering

Clustering adalah teknik pengelompokan kumpulan objek serupa dalam grup yang sama yang berbeda dari objek grup lainnya. Beberapa teknik clustering adalah K-Means, DBSCAN, dan hierarki clustering.

b. Deteksi anomali

Deteksi anomali merupakan teknik untuk mendeteksi pencilan dalam data tanpa label dengan asumsi bahwa sebagian besar sampel data berdistribusi normal dengan mengamati instance yang sesuai dengan kumpulan data lainnya.

c. Deep belief network.

Deep belief network adalah model grafis generatif yang juga termasuk ke dalam algoritma *neural network* yang dirancang untuk *unsupervised learning.*

3. Reinforcement Learning

Reinforcement learning adalah algoritma yang memungkinkan mesin untuk berinteraksi dengan lingkungan dinamis untuk mencapai target atau tujuan. Dengan algoritma ini, mesin dan agen *software* dapat mengevaluasi perilaku ideal dalam suatu kasus. Dengan bantuan *reward*, agen akan mempelajari perilaku dan memperbaiki model yang telah dihasilkan. Umpan balik dari *reward* ini dikenal dengan sinyal penguatan.

2.2.3 Deep Learning

Deep Learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan saraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *deep learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *machine learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi(Jimmy Pujoseno, 2018).

Belakangan ini *deep learning* menjadi sorotan dalam pengembangan *machine learning.* Alasannya yaitu karena *deep learning* telah mencapai hasil yang luar biasa dalam visi komputer (Krizhevsky, A.,Sutskever I., 2012). *Deep learning* memungkinkan model komputasi yang terdiri dari beberapa lapisan pemrosesan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. Metode-metode ini telah secara dramatis meningkatkan *state-of-the-art* dalam *speech recognition, visual object recognition, object detection* dan banyak domain lainnya seperti penemuan obat

dan genomik. *Deep learning* menemukan struktur rumit dalam kumpulan data besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah parameter internalnya yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya. *Deep convolutional nets* telah menghasilkan terobosan dalam *processing images, video, speech* dan audio, sedangkan jaring berulang telah menyoroti data berurutan seperti *text* dan *speech* (Lecun et al., 2015).

Machine intelligence berguna dalam beberapa banyak situasi dan setara atau lebih baik daripada human experts dalam beberapa kasus, artinya *deep learning* dapat menjadi solusi untuk permasalahan berikut :(Alzubaidi et al., 2021)

1. Kasus dimana *human experts* tidak tersedia.
2. Kasus dimana *human* tidak mampu menjelaskan keputusan yang diambil menggunakan keahliannya (*language understanding, medical decisions, and speech recognition*).
3. Kasus dimana solusi masalah diperbaharui seiring waktu (*price prediction, stock preferernce, weather prediction, and tracking*).
4. Kasus dimana solusi memerlukan adaptasi berdasarkan kasus tertentu (*personalization, biometrics*).
5. Kasus dimna ukuran masalahnya sangat besar dan melebihi kemampuan penalaran yang tidak memadai (*sentiment analysis, matching ads to Facebook, calculation webpage ranks*).

Beberapa fitur kinerja yang dimiliki *deep learning* antara lain:

1. *Universsal Learning Approach* : Karena *deep learning* mempunyai kemampuan untuk bekerja di hampir semua domain aplikasi.
2. *Robustness*: Secara umum, fitur yang dirancang secara presisi tidak diperlukan dalam teknik *deep learning*. Sebaliknya, fitur yang dioptimalkan dipelajari secara otomastis terkait dengan tugas yang sedang dipertimbangkan. Dengan demikian, ketahanan terhadap perubahan biasa pada data masukan dapat dicapai.

3. *Generalization*: Tipe data yang berbeda atau aplikasi yang berbeda dapat menggunakan teknik *deep learning* yang sama, pendekatan yang sering disebut sebagai *transfer learning*(TL) merupakan pendekatan yang berguna dalam permasalahan dimana data tidak mencukupi.
4. *Scalability*: *Deep learning* sangat *scalable*. ResNet yang ditemukan oleh *Microsoft*, terdiri dari 1202 lapisan dan sering diterapkan pada *supercomputing scale*. *Lawrence Livermore National Laboratory*(LLNL), perusahaan besar yang bekerja pada pengembangan *framework*, mengadopsi pendekatan serupa dimana ribuan *node* dapat diimplementasikan.

2.2.4 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*/gambar. CNN digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image* dan untuk membaca, mendeteksi dan mengenali sebuah objek pada sebuah citra digital (data berupa gambar) yaitu dengan cara mengubah struktur gambar menjadi matrix array atau pixel, maksudnya data gambar yang dibaca akan diubah oleh neural network menjadi sebuah matriks dengan nilai pixel tergantung pada intensitas pixel pada setiap titik.*Convolutional Neural Network* (CNN) merupakan salah satu metode *Deep Learning* (DL) yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital (BELLA, 2021).

Adapun manfaat menggunakan CNN sebagai berikut: (Alzubaidi et al., 2021)

1. Fitur pembagian bobot yang mengurangi jumlah parameter jaringan yang dapat dilatih dan membantu jaringan untuk meningkatkan generalisasi dan menghindari overfitting.
2. Secara bersamaan mempelajari lapisan ekstraksi fitur dan lapisan klasifikasi menyebabkan keluaran model menjadi sangat terorganisir dan sangat bergantung pada fitur yang diekstraksi.
3. Implementasi jaringan skala besar jauh lebih mudah dengan CNN dibandingkan dengan jaringan saraf lainnya.

2.2.5 OpenCV

OpenCV (*Open Computer Vision Library*) merupakan library perangkat lunak open source yang memiliki lisensi BSD-licensed product. OpenCV memiliki lebih dari 2500 algoritma yang telah di optimasi disediakan untuk menangani hal mengenai computer vision dan machine learning. Algoritma yang ada dapat digunakan untuk mendeteksi wajah, mengenali wajah, mengidentifikasi objek, dan lain-lain.

OpenCV dimulai di Intel pada tahun 1999 oleh Gary Bradsky dan rilis pertama keluar pada tahun 2000.Vadim Pisarevsky bergabung dengan Gary Bradsky untuk mengelola tim OpenCV perangkat lunak Rusia Intel.Pada tahun 2005, OpenCV digunakan pada Stanley, kendaraan yang memenangkan DARPA Grand Challenge 2005.Kemudian pengembangan aktifnya berlanjut di bawah dukungan Willow Garage, dengan Gary Bradsky dan Vadim Pisarevsky memimpin proyek.Saat ini, OpenCV mendukung banyak algoritma yang terkait dengan Visi Komputer dan Pembelajaran Mesin dan memperluas hari demi hari.Saat ini OpenCV mendukung berbagai bahasa pemrograman seperti C , Python, Java dll dan tersedia di berbagai platform termasuk Windows, Linux, OS X, Android, iOS dll.Selain itu, antarmuka berdasarkan CUDA dan OpenCL juga sedang dikembangkan secara aktif untuk operasi GPU berkecepatan tinggi. Kombinasi terbaik untuk dapat melakukan operasi berkecepatan tinggi tersebut adalah dengan perpaduan antara OpenCV, C++ API dan bahasa pemrograman Python (Mordvintsev, 2017).

Computer Vision adalah kemampuan mesin/komputer dalam melihat hingga mampu mengekstrak informasi dari sebuah gambar. Salah satu bidang yang berkaitan dengan *Computer Vision* adalah pengolahan citra atau yang biasa disebut *Image Processing* dan *Video Processing*. (Susanti & Fadillah, 2019)

OpenCV memiliki banyak fitur, antara lain: pengenalan wajah, pelacakan wajah, deteksi wajah, kalman filtering, dan berbagai jenis metode AI (*Aritificial Intelegence*) (Puspaningrum & Saputra, 2018).

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di kampus 1 Politeknik Negeri Ujung Pandang, Jl. Perintis Kemerdekaan KM 10 Makassar, Sulawesi Selatan. Penelitian ini dilakukan pada bulan April 2022 sampai dengan bulan Agustus 2022.

3.2 Alat dan Bahan

Untuk mendukung proses pengerjaan penelitian ini dibutuhkan perangkat yang mampu menjalankan sistem dengan baik sehingga tujuan penelitian dapat tercapai. Perangkat penelitian dikategorikan menjadi perangkat keras (*hardware*) dan perangkat lunak (*software*).

1. Perangkat Keras (*Hardware*)

Tabel 3. 1 Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi	Kegunaan
1.	Sebuah Laptop	• Processor Intel® CORE™ i5-1035G1 CPU @ 1.00GHz 1.19 GHz • Windows 10 • RAM 12 GB	Untuk membangun dan menjalankan sistem deteksi kantuk beserta perangkat lunak pendukungnya.
2.	Sebuah *smartphone*	Samsung M21	Digunakan sebagai penangkap dan pendeteksian wajah

3.	Sebuah alat pengukur intensitas cahaya	Lux meter	Dapat digunakan sebagai pengukuran intensitas cahaya.

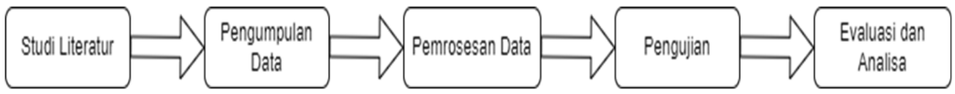
2. Perangkat Lunak (*Software*)

Tabel 3. 2 Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Deskripsi	Kegunaan
1.	*Programming Language*	*Python 3*	Digunakan untuk menjalankan kode program dari sistem yang akan dibuat.
2.	*Text Editor*	*Jupyter Notebook*	Digunakan untuk menulis kode program dari sistem yang akan dibuat.
3.	*Webcam*	Aplikasi *DroidCam*	Digunakan sebagai penangkap dan pendeteksian wajah

3.3 Prosuder penelitian

Agar penelitian yang dilakukan dapat berjalan dengan baik dan terstruktur diperlukan sebuah metode perancangan sehingga hasil yang diperoleh sesuai dengan tujuan penelitian. Penelitian ini akan merancang suatu sistem yang bertujuan untuk mendeteksi keberadaan user dengan memanfaatkan perangkat *mobile* kemudian diuji coba sehingga menghasilkan data dan informasi untuk dapat diambil keputusan yang digunakan dalam membuat dokumentasi penelitian. Penjelasan mengenai tahapan proses penelitian yang dilakukan digambarkan pada Gambar 3.1



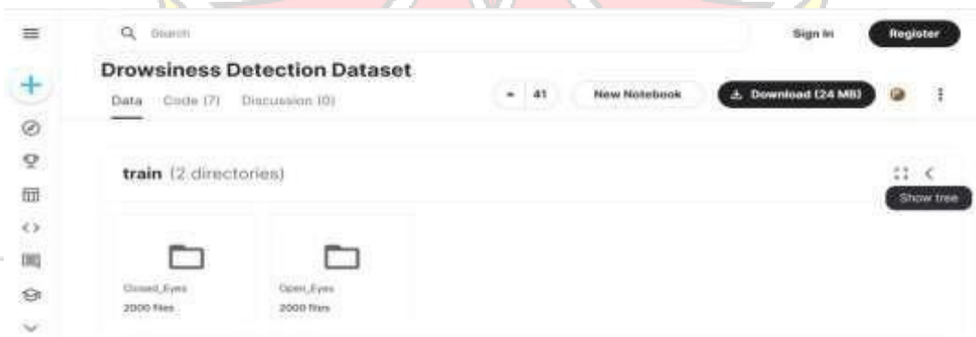
Gambar 3. 1 Tahapan Proses Penelitian

3.3.1 Studi Literatur

Studi literatur dilakukan untuk mengumpulkan dan memperoleh data yang tepat dalam penelitian dengan cara mempelajari, membaca, dan mencatat literatur baik itu berupa jurnal ilmiah atau artikel ilmiah yang tersedia melalui halaman web di internet yang berkaitan dengan masalah penelitian.

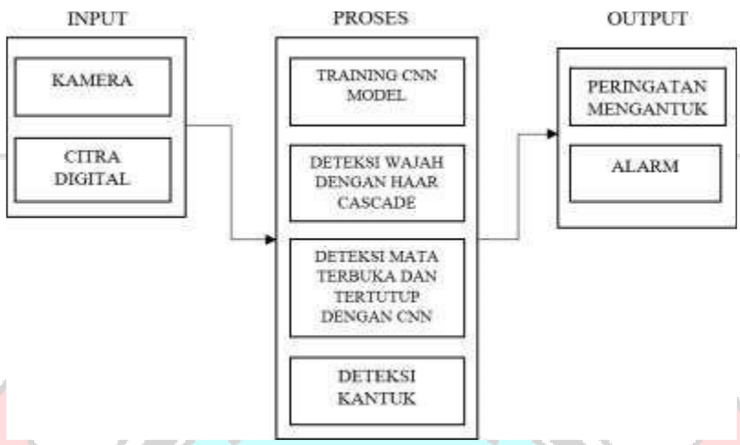
3.3.2 Pengumpulan Data

Pada tahapan ini, pengumplan data dilakukan dengan cara mengambil dataset dari *website Kaggle.com (*https://www.kaggle.com/datasets/prasadvpatil/mrl-dataset?resource=download) seperti pada gambar 3.2 yang berisi 4000 *file* mencakup gambar normal dan gambar pengemudi sedang menutup mata dengan kondisi pencahayaan yang berbeda. Tujuan pengambilan dataset ini digunakan sebagai *data preparation, model training* dan data percobaan. *Data preparation* dilakukan untuk memastikan keakuratan dan konsisten data mentah yang disiapkan untuk pemrosesan dan analisis data. *Model training* dilakukan untuk membuat atau melatih model *machine learning* dan data percobaan digunakan untuk menguji performa atau akurasi dari model yang telah di-*training*.



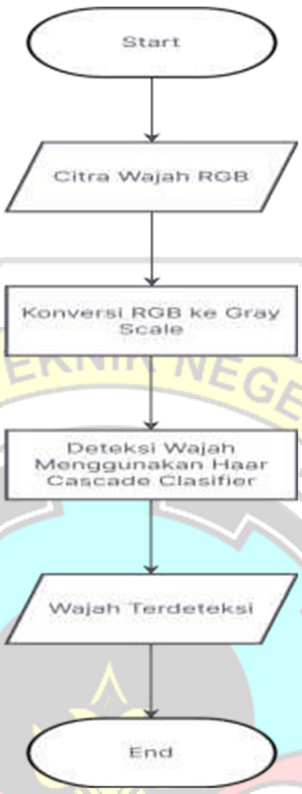
Gambar 3. 2 Lokasi Pengambilan Dataset

3.3.3 Pemrosesan Data



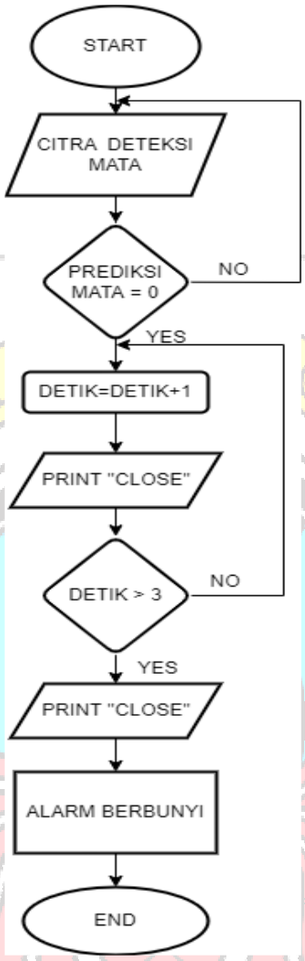
Gambar 3. 3 Diagram Blok Sistem

Pada gambar 3.3 merupakan diagram blok sistem yang terdiri tiga bagian yaitu input, proses, dan output. Pada tahap input yang dimulai dari kamera yang akan menangkap citra. Kemudian, tahap proses yang terdiri dari empat bagian yaitu melakukan training CNN model dengan data gambar yang terdiri dari gambar sepasang mata terbuka dan sepasang mata tertutup, mendeteksi keberadaan wajah dengan menggunakan *Haar Cascade*, mendeteksi mata dalam kondisi terbuka dan tertutup menggunakan CNN, dan mendeteksi kantuk yang dari lamanya durasi mata terpejam. Selanjutnya, pada tahap output akan menghasilkan berupa tulisan peringatanmengantuk yang akan tampil pada monitor pada saat mata dalam kondisi tertutup yangdurasinya selama 3 detik dan lebih dari 3 detik yang terdapat suara alarm saat *user* terindikasi mengantuk dan terdapat keterangan apakah mata dalam kondisi terbuka atau tertutup, juga terdapat durasi atau waktu yang akan mulai terhitung saat mata dalam kondisi tertutup. Durasi yang menjadi acuan untuk pendeteksian kantuk berasal dari teori *microsleep* yang memiliki durasi sekitar 3 sampai 5 detik, bahkan ada yang sampai 10 detik.(*Biar Tidak Terserang Microsleep Saat Mudik? Ini Saran Dokter*, n.d.). Jika setelah kondisi tertutup, kemudian mata terbuka lagi, maka proses perhitungan waktunya akan berhitung mundur lagi dan kembali ke angka 0.



Gambar 3. 4 Flowchart Deteksi Wajah

Pada Gambar 3.4 merupakan flowchart deteksi wajah. Gambar 3.4 dijelaskan bahwa proses menginput berupa citra wajah RGB yang didapat dari kamera, kemudian memproses konversi ke *gray scale* untuk mempermudah proses, setelah itu dilakukan proses pendeteksian wajah menggunakan *haar cascade classifier* yang dikhususkan untuk wajah bagian depan dengan menghasilkan output wajah yang terdeteksi.



Gambar 3. 5 Flowchart Deteksi Kantuk

Pada gambar 3.5 merupakan flowchart deteksi kantuk. Pada gambar 3.5 dapat dijelaskan bahwa input berupa citra dari hasil deteksi mata yang terbuka dan tertutup dilakukan pemilihan kondisi untuk nilai dari prediksi mata. Jika waktu durasi mata tertutup lebih dari 3 detik maka ditampilkan tulisan bahaya serta alarm peringatan akan berbunyi.

3.3.4 Pengujian

Pada tahap ini dilakukan pengujian dan percobaan terhadap sistem sesuai dengan spesifikasi yang telah ditentukan sebelumnya serta memastikan program yang telah dibuat dapat berjalan dengan baik sesuai dengan yang diharapkan.

1. Pengujian Akurasi

Pada tahap teknik pengujian akurasi digunakan untuk sebagai seberapa kedekatan nilai yang dihasilkan dari pengujian dengan nilai yang sebenarnya (Aryani et al., 2017). Adapun formulasi yang dapat dilihat pada persamaan beriku ini.

$$Accuracy\ (\%) = \frac{Correct\ Answer}{Total\ Testing} * 100$$

Correct Answer adalah satuan yang menyatakan jumlah hasil benar luaran yang dihasilkan sedangkan total testing adalah dari keseluruhan data yang diproses atau diujikan pada sistem (I. P. A. E. D. U. Udayana and P. G. S. C. Nugraha, 2020)

2. Pengujian Sistem

Pada tahap ini dilakukan proses pengujian sistem dengan menggunakan pengujian akurasi sistem yang berfungsi untuk menilai komparasi penggunaan metode peningkatan kualitas citra dalam mengenali objek pada wajah (Udayana, I. P. A. E. D., & Supartha, 2021). Pengujian ini menggunakan *Haar Cascade* untuk mengetahui jarak minimal dan maksimal yang dapat diterapkan untuk melakukan pendeteksian wajah. Pengujian sistem ini dilakukan sebanyak 5 *sample* untuk setiap tiga kondisi cuaca dengan jarak penglihatan yang berbeda.

3.3.5 Evaluasi dan Analisa

Pada tahapan ini dilakukan evaluasi dan analisis yang dijadikan sebagai bahan referensi agar sesuai dengan tujuan yang ingin dicapai dari penelitian ini. Data atau informasi hasil pengujian yang dievaluasi akan dijadikan sebagai bahan dalam pembuatan dokumen sebagai produk akhir penelitian. Masalah yang ditemukan dalam proses penelitian turut dijadikan sebagai data hasil penelitian agar memudahkan pembaca untuk mengembangkan penelitian lebih lanjut.

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem bertujuan untuk menjelaskan tahapan - tahapan implementasi tersebut berupa implementasi perangkat lunak, implementasi perangkat keras dan hasil dari pengujian.

4.1.1 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan proses instalasi perangkat lunak, sehingga dapat beroperasi dengan benar. Proses instalasi perangkat lunak dapat dilihat pada tabel 4.1.

Tabel 4. 1 Implementasi Perangkat Lunak

Perangkat Lunak	Keterangan
Microsoft Windows 10 Pro	*Operating System*
Jupyter Notebook	Aplikasi Pembangun *System*
DroidCam	Aplikasi *Webcam*

4.1.2 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan realisasi dari analisis dan perancangan kebutuhan perangkat keras. Implementasi perangkat keras yangdilakukan meliputi perangkat keras yang digunakan dalam pembanguan sistem dan perangkat keras yang diperuntukan bagi komputer yang akan menggunakan program ini. Berikut ini merupakan spesifikasi perangkat keras untuk komputer dan juga webcam yang digunakan dalam pembangunan sistem sebagai berikut:

Tabel 4. 2 Implementasi Perangkat Keras

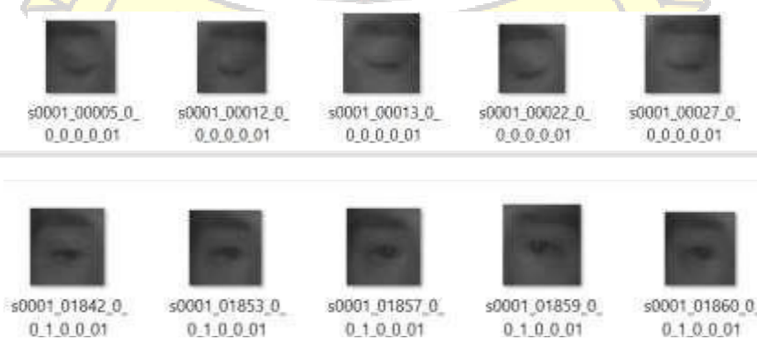
Perangkat Keras	Keterangan
Laptop	• Sistem Operasi Windows 10 Pro 64 bit • Processor : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz • RAM: 12 GB
Smartphone	• Merek: Samsung M21 • RAM 4GB + ROM 64 GB • Camera : Single 20 MP, f/2.0, 26mm (wide), Resolusi video 1080p@30fps
Lux Meter	• Satuan metrik ukuran cahaya pada suatu permukaan : lux

4.1.3 Perancangan Sistem

Implementasi program ini merupakan sebuah langkah penggunaan program sehingga dapat berjalan dengan tahap perancangan.

1. Pengumpulan *Dataset*

Dataset gambar mata ini memiliki total gambar sebanyak 4000 gambar yang terbagi menjadi dua bagian yaitu training data yang berjumlah 3200 gambar memliki 2 kelas dan test data yang berjumlah 800 gambar memiliki 2 kelas



Gambar 4. 1 Dataset

2. *Preprocessing Dataset*

Pada tahap proses pengumpulan data, gambar-gambar tersebut di *training* mengunakan *convolutional neural network* sehingga menghasilkan gambar yang kelihatan kasar. Gambar – gambar tersebut diubah ukurannya menjadi bentuk persegi yang berukuran 256 x 256 piksel.

3. Arsitektur

Pada tahap ini dirancang untuk mengenali beberapa gambar mata seperti Lapisan konvolusi yang digunakan diikuti oleh *pooling layer* dan *dense layer* digunakan untuk merancang model ini (Risdin, F., Mondal, P. K., & Hassan, 2020). Pada lapisan pertama dan kedua adalah lapisan konvolusi yang memiliki ukuran filter 32. Lapisan pertama dianggap sebagai lapisan input yang meminta saluran RGB ukuran 32x32 yang menggunakan *padding* yang sama. Kedua layer menggunakan aktivasi ReLU (1) dan memiliki properti yang sama untuk *padding*. Output dari layer kedua terhubung dengan *max pooling layer* yang memiliki *pool size* 2. Lapisan ketiga yang terhubung ke lapisan konvolusi 4 dari filter 64 dan ukuran kernel 3. Lapisan keempat memiliki filter 128 dan ukuran kernel 3 .

$$\mathrm{ReLU}(X) = \mathrm{MAX}(0, X)$$

Gambar 4. 2 Aktivasi ReLu (Risdin, F., Mondal, P. K., & Hassan, 2020) Fungsi

categorical cross entropy (4) digunakan untuk menghitung kesalahan. Eksplorasi yang sedang berlangsung menunjukkan bahwa cross-entropy menunjukkan beberapa kualitas daripada fungsi lain seperti kesalahan klasifikasi (Risdin, F., Mondal, P. K., & Hassan, 2020).

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \quad \text{for } j=1,\ldots,k$$

Gambar 4. 3 Categorical Cross Entropy Function (Risdin, F., Mondal, P. K., & Hassan, 2020)

4. *Training the model*

Pada tahap ini, model di *training* pada dataset yang telah dikumpulkan dengan menggunakan *batch size* yaitu 128 dan dilakukan *epoch* sebanyak 5.

5. Proses Pengambilan Gambar sebagai Input dari *Webcam*

Pada gambar 4.4 merupakan *script* yang digunakan untuk pengambilan gambar sebagai input dari *webcam* dan juga digunakan untuk mengembalikan video dari *webcam* pertama di komputer.

```
cap = cv2.VideoCapture(0)
```

Gambar 4. 4 Pengambilan Gambar Sebagai Input Dari Webcam

Pada gambar 4.5 merupakan *script* merupakan proses *infinite loop* yang digunakan untuk untuk mengakses kamera dan mengatur *capture object*. *cap.read()* akan membaca setiap frame dan menyimpan gambar dalam *variabel frame*.

```
while True:
    ret, frame = cap.read()
```

Gambar 4. 5 Pembuatan Infinite Loop

6. Pendeteksian Wajah dalam Gambar dan Pembuatan *Region of Interest* (ROI)

Pada tahap ini digunakan untuk mendeteksi wajah dalam citra. Hal yang perlu dilakukan dalam mendeteksi wajah yaitu mengubah citra menjadi skala abu-abu/*grayscale*. Pada gambar 4.6 merupakan *script* yang menggunakan pengklasifikasi *haar cascade* untuk mendeteksi wajah yang digunakan untuk mengatur *classifier*.

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
```

Gambar 4. 6 Pengaturan CascadeClassifier Wajah

Selanjutnya, pada gambar 4.7 merupakan *script MultiScale* mendeteksi wajah yang digunakan untuk mendeteksi wajah. Script ini akan mengembalikan persegi panjang dengan koordinat (x,y,w,h) di sekitar wajah yang terdeteksi.

```
faces= face_cascade.detectMultiScale(gray, scaleFactor= 1.2, minNeighbors=3)
```

Gambar 4. 7 MultiScale Mendeteksi Wajah

7. Pendeteksian Mata dari ROI dan Pemberian *Classifier*

Pada tahap ini memiliki prosedur yang sama untuk mendeteksi wajah yaitu digunakan untuk mendeteksi mata. Hal yang perlu dilakukan dalam mendeteksi mata yaitu mengubah citra menjadi skala abu-abu/*grayscale*. Pada gambar 4.8 merupakan *script* yang menggunakan pengklasifikasi *haar cascade* untuk mendeteksi mata yang digunakan untuk mengatur *classifier*.

```
eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
```

Gambar 4. 8 Pengaturan CascadeClassifier Mata

Selanjutnya, pada gambar 4.9 merupakan *script MultiScale* mendeteksi mata yang digunakan untuk mendeteksi wajah. *Script* ini akan mengembalikan persegi panjang dengan koordinat (x,y,w,h) di sekitar wajah yang terdeteksi.

```
eyes= eye_cascade.detectMultiScale(gray, scaleFactor= 1.2, minNeighbors=3)
```

Gambar 4. 9 MultiScale Mendeteksi Mata

Kemudian, mengekstrak mata dari gambar penuh dengan menggunakan *script* yang tertera pada gambar 4.10. *Script* ini dapat dicapai dengan mengekstraksi kotak batas mata dan dapat menarik gambar mata dari bingkai dengan kode ini yang kemudian akan dimasukkan ke dalam pengklasifikasi CNN untuk memprediksi apakah mata terbuka atau tertutup.

```
eye= frame[ey:ey+eh,ex:ex+ew]
```

Gambar 4. 10 Mengekstrak Mata

8. Pengklasifikasi dalam Mengkategorikan apakah Mata Terbuka atau Tertutup Pada tahap ini merupakan pengklasifikasi CNN untuk memprediksi status mata. Untuk memasukkan gambar ke dalam model, diperlukan melakukan operasi tertentu karena model membutuhkan dimensi yang benar untuk memulai. Hal yang pertama dilakukan yaitu mengubah citra berwarna menjadi *grayscale* seperti pada gambar 4.11.

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

Gambar 4. 11 Converting to Gray-scale

Pada gambar 4.12 merupakan *script* yang digunakan untuk mengubah ukuran gambar menjadi 80*80 piksel karena model yang dilatih pada gambar 80*80.

```
eye= cv2.resize(eye,(80,80))
```

Gambar 4. 12 Mengubah Ukuran Gambar

Selanjutnya, menormalkan data untuk konvergensi yang lebih baik seperti pada gambar 4.13 dimana semua nilai akan berada di antara 0-1.

```
eye= eye/255
```

Gambar 4. 13 Menormalkan Data

Untuk memuat model menggunakan *script* yang tertera pada gambar 4.14.

```
model = load_model(r'C:\Users\LENOVO\Documents\drowsiness\models\model.h5')
```

Gambar 4. 14 Memuat Model

Untuk memprediksi setiap mata dengan model menggunakan *script* seperti pada gambar 4.15 . Jika nilai prediksi bernilai 1 maka menyatakan mata terbuka, jika nilai prediksi bernilai 0 maka menyatakan mata tertutup.

```
prediction= model.predict(eye)
```

Gambar 4. 15 Prediksi Mata

9. Menghitung Skor untuk Memeriksa apakah Orang Mengantuk

Skor pada dasarnya adalah nilai yang akan digunakan untuk menentukan berapa lama seseorang telah menutup matanya. Jadi jika kedua mata tertutup, maka akan terus meningkatkan skor dan ketika mata terbuka, maka akan terus mengurangi skor. Untuk melihat skor di layar menggunakan script yang tertera pada gambar 4.16 dan gambar 4.17 yang akan menampilkan status *realtime* orang tersebut.

```
cv2.putText(frame,'closed',(10,height-20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,fontScale=1,color=(255,255,255),
            thickness=1,lineType=cv2.LINE_AA)
cv2.putText(frame,'Score'+str(Score),(100,height-20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,
            fontScale=1,color=(255,255,255),thickness=1,lineType=cv2.LINE_AA)
```

Gambar 4. 16 Menampilkan Status Mata Tertutup

```
cv2.putText(frame,'open',(10,height-20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,fontScale=1,color=(255,255,255),
            thickness=1,lineType=cv2.LINE_AA)
cv2.putText(frame,'Score'+str(Score),(100,height-20),fontFace=cv2.FONT_HERSHEY_COMPLEX_SMALL,
            fontScale=1,color=(255,255,255),thickness=1,lineType=cv2.LINE_AA)
```

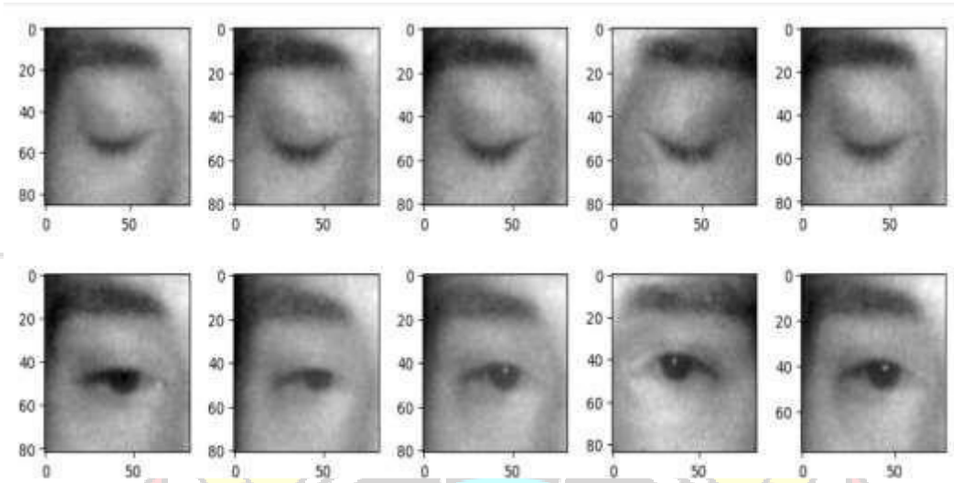
Gambar 4. 17 Menampilkan Status Mata Terbuka

4.2 Pengujian dan Hasil Pengujian

Pengujian dilakukan dengan serangkaian percobaan-percobaan dalam kondisi-kondisi tertentu yang dapat mempengaruhi keefektifan kinerja sistem pendeteksian mata mengantuk.

4.2.1 Training, Testing and the Validation of the model

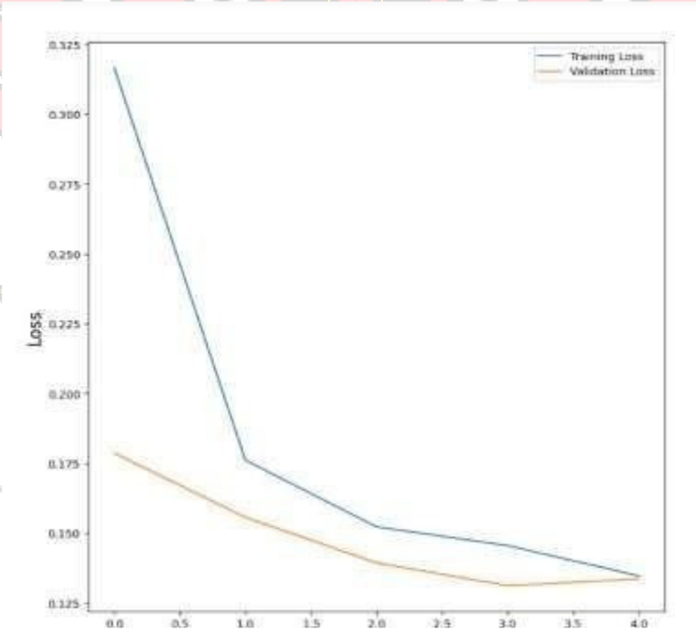
Pada gambar 4.18 merupakan hasil pengujian model memprediksi data milik kelas mana. Selama pengujian jenis ini, semua gambar data validasi dipotong dalam ukuran 256 x 256 piksel.



Gambar 4. 18 *Computer Vision* dari data Validasi

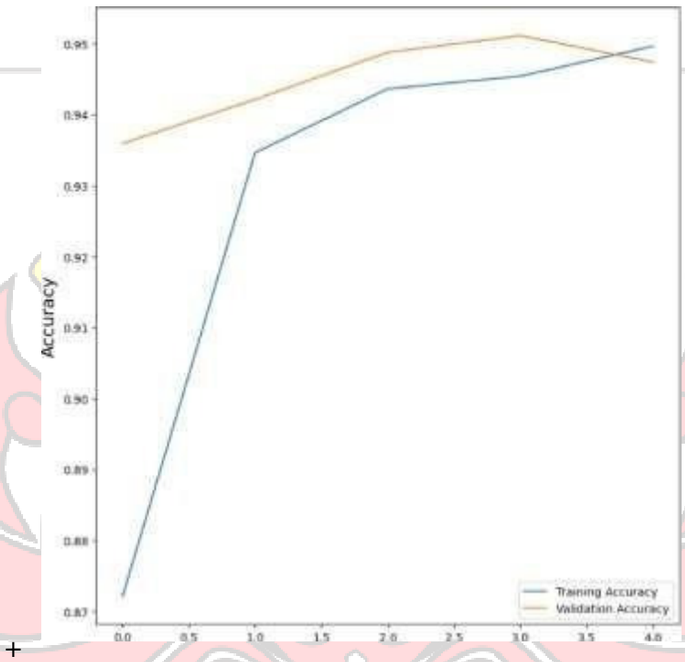
4.2.2 Model Performance

Pada gambar 4.19 merupakan hasil *loss* dari *training* dan *validation* dataset yang menjalankan 5 epoch model memperoleh *loss* 18,5% untuk *training dataset* yang dibuat dan 14,76% pada dataset validasi.



Gambar 4. 19 *Loss* dari *Training* dan *Validation*

Pada gambar 4.20 merupakan hasil akurasi training dataset dan dataset validasi yang menjalankan 5 epoch model memperoleh akurasi 92,73% untuk *training dataset* yang dibuat dan 94,51% pada dataset validasi.



Gambar 4. 20 Akurasi dari *Training* dan *Validation*

4.2.3 Model Summary

Pada gambar 4.21 merupakan visualisasi rangkuman. Gambar tersebut menunjukkan arsitektur model yang diusulkan yang mencakup banyak lapisan untuk mengimplementasikan model. Lapisan *convolution* dan *max-pooling* digunakan di bagian ekstraksi fitur dan lapisan padat dan soft-max digunakan sebagai lapisan yang terhubung penuh.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 256, 256, 32)      320

 max_pooling2d (MaxPooling2D  (None, 128, 128, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 128, 128, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 64, 64, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 64, 64, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 32, 32, 128)      0
 2D)

 flatten (Flatten)           (None, 131072)            0

 dense (Dense)               (None, 64)                8388672

 dense_1 (Dense)             (None, 2)                 130

=================================================================
Total params: 8,481,474
Trainable params: 8,481,474
Non-trainable params: 0
_________________________________________________________________
None
```

Gambar 4. 21 Visualisasi Rangkuman Model

4.2.4 Pengujian Jarak Wajah terhadap Webcam

Pengujian ini dilakukan dengan menggunakan aplikasi *DroidCam* dimana kamera *smartphone* dihubungkan ke laptop sebagai pengganti *webcam* yang ada di laptop. Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam membaca dan mendeteksi citra wajah pengemudi dengan jarak yang berbeda-beda serta mengetahui berapa jarak efektif yang dapat dicapai sistem pada saat bekerja. Pengujian ini dilakukan pada saat mata keadaan tertutup.

Tabel 4. 3 Hasil Pengujian Jarak Wajah terhadap Webcam

Percobaan	Jarak Wajah (cm)		
	40	50	60
1	√	√	√
2	√	√	√
3	√	√	√
4	√	√	√
5	√	√	√

Keterangan:

√ = sistem mendeteksi mata

× = sistem tidak mendeteksi mata

Berikut ini persentase data yang telah diuji:

1. Untuk jarak 40 cm

Persentase data = $\frac{5}{5} \times 100\%$

= 100%

2. Untuk jarak 50 cm

Persentase data = $\frac{5}{5} \times 100\%$

= 100%

3. Untuk jarak 60 cm

$$\text{Persentase data} = \frac{5}{5} \times 100\%$$

$$= 100\%$$

Dari pengujian diatas dapat disimpulkan bahwa semakin jauh jarak wajah dan semakin dekat jarak wajah ke kamera maka sistem deteksi mata akan bekerja lebih baik apabila dilakukan pada jarak yang agak jauh dari posisi kamera. Nilai optimum dari pengujian deteksi objek berdasarkan jarak kamera adalah 40-60 cm.

4.2.5 Pengujian *Delay* Pembacaan Mata

Tujuan pengujian ini adalah untuk mengetahui sistem dalam membaca dan mendeteksi wajah pengemudi bekerja dengan baik. Pengujian ini dilakukan dengan jarak uji yaitu 40 cm, 50 cm, dan 60 cm dengan posisi lurus menghadap depan *webcam*.

Tabel 4. 4 Hasil Pengujian Delay Pembacaan Mata Pada Jarak 40 cm

Percobaan	Pembacaan Wajah	Delay (s)
1	Terdeteksi	0.257
2	Terdeteksi	0.276
3	Terdeteksi	0.285
4	Terdeteksi	0.270
5	Terdeteksi	0.253

Tabel 4. 5 Hasil Pengujian Delay Pembacaan Mata Pada Jarak 50 cm

Percobaan	Pembacaan Wajah	Delay (s)
1	Terdeteksi	0.361
2	Terdeteksi	0.399
3	Terdeteksi	0.386
4	Terdeteksi	0.366
5	Terdeteksi	0.379

Tabel 4. 6 Hasil Pengujian Delay Pembacaan Mata Pada Jarak 60 cm

Percobaan	Pembacaan Wajah	Delay (s)
1	Terdeteksi	0.446
2	Terdeteksi	0.479
3	Terdeteksi	0.465
4	Terdeteksi	0.484
5	Terdeteksi	0.457

Berikut ini nilai rata-rata data yang telah diuji:

1. Untuk jarak 40 cm

Nilai rata-rata data $= \frac{0.257 + 0.276 + 0.285 + 0.270 + 0.253}{5}$

$= \frac{1.341}{5}$

= 0.2682 detik

2. Untuk jarak 50 cm

$$\text{Nilai rata-rata data} = \frac{0.361 + 0.399 + 0.386 + 0.366 + 0.379}{5}$$

$$= \frac{1.891}{5}$$

$$= 0.3782 \text{ detik}$$

3. Untuk jarak 60 cm

$$\text{Nilai rata-rata data} = \frac{0.446 + 0.479 + 0.465 + 0.484 + 0.457}{5}$$

$$= \frac{2.331}{5}$$

$$= 0.4662 \text{ detik}$$

Pada tabel diatas, hasil pengujian untuk nilai delay pembacaan mata pada jarak 40 cm memiliki nilai rata-rata 0.2682 detik. Pada jarak 50 cm, nilai delay pembacaan mata memiliki nilai rata-rata 0.3782 detik. Pada jarak 60 cm, nilai delay pembacaan mata memiliki nilai rata-rata 0.4662 detik . Pada pengujian ini dapat disimpulkan bahwa semakin jauh jarak deteksi kantuk maka semakin besar delay pembacaan mata yang terdeteksi.

4.2.6 Intensitas Cahaya terhadap Deteksi Mata Mengantuk

Pada pengujian ini bertujuan untuk mengetahui pengaruh cahaya terhadap proses deteksi mata mengantuk. Untuk mengetahui nilai intensitas cahaya dapat menggunakan alat ukur pencahayaan yaitu lux meter . Lux meter memiliki satuan lux yang didefinisikan sebagai satuan metrik ukuran cahaya pada suatu permukaan.lux meter memiliki range intensitas cahaya antara 1-100.000 Lux. Dalam pengujian ini dilakukan pada tiga kondisi yaitu pada pagi hari, siang hari dan malam hari dengan jarak 50 cm dari *webcam*.

Tabel 4.7 Pengujian Intensitas Cahaya terhadap Deteksi Mata Mengantuk

Kondisi	Lux Meter	Pembacaan Mata Mengantuk	Keterangan
Pagi (Cerah)	(900 lux) 		Terbaca Mata Mengantuk

Pagi (Mendung)	(150 lux) 		Terbaca Mata Mengantuk
Siang (Cerah)	(1000 lux) 		Terbaca Mata Mengantuk

Kondisi	Lux Meter	Pembacaan Mata Mengantuk	Keterangan
Malam (Jalanan terang)	(100 lux) 		Terbaca Mata Mengantuk
Malam (Jalanan Gelap)	(0 lux) 		Tidak Terbaca Mata Mengantuk

Pada pengujian ini, inputan objek (wajah) sangat dipengaruhi terhadap intensitas cahaya. Semakin tinggi ukuran yang dihasilkan lux meter maka semakin mudah sistem mendeteksi objek. Pengukuran intensitas cahaya dilakukan pada 3 kondisi waktu yaitu pagi hari, siang hari, dan malam hari. Berdasarkan tabel 4.7 objek lebih mudah dideteksi pada waktu pagi hari dan siang hari dikarenakan sinar matahari lebih banyak langsung menerpa wajah pengemudi sehingga intensitas cahaya yang dihasilkan semakin tinggi. Pada waktu malam hari, sistem tidak bisa membaca objek dengan baik dikarenakan intensitas cahaya yang didapatkan sedikit sehingga mempengaruhi sistem dalam mendeteksi objek.

BAB V PENUTUP

a. Kesimpulan

Dari proses perancangan program hingga pengujian yang telah dilakukan , maka dapat disimpulkan sebagai berikut:

1. Penelitian ini berhasil membuat sistem Driver Drowsiness dengan menerapkan algoritma CNN dengan performa akurasi sebesar 94,51%.
2. Penelitian ini di dapatkan jarak maksimal wajah antara 40 cm sampai 60 cm terhadap webcam.
3. Waktu delay pembacaan mata terhadap webcam pada jarak 40 cm di perolah rata-rata 0.2682 detik, jarak 50 cm di perolah rata-rata 0.3782 detik , sedangkan jarak 60 cm di perolah rata-tara 0.4662 detik.
4. Hasil pengujian intensitas cahaya terhadap deteksi mata mengantuk memiliki nilai maksimal antara 0-900 lux.

b. Saran

Pada penelitian ini masih terdapat kekurangan, sehingga diharapkan penelitian ini dapat dilakukan pengembangan diantaranya:

1. Penerapan pada kasus lingkungan yang berbeda.
2. Kondisi pencahayaan mempengaruhi keberhasilan dari pengujian.
3. Penerapan algortima *machine learning* yang digunakan lebih bervariasi dan akurat.

DAFTAR PUSTAKA

Ahmad, A. (n.d.). *Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning*. Retrieved December 28, 2021, from www.teknoindonesia.com

Ahmad Kamran, M., Mannan, M. M. N., & Jeong, M. Y. (2019). Drowsiness, Fatigue and Poor Sleep‟s Causes and Detection: A Comprehensive Study. *IEEE Access*, *7*, 167172–167186.

Alfari, A. Z. (2013). *DETEKSI MATA MENGANTUK BERBASIS CITRA DIGITAL MENGGUNAKAN METODE HAAR CLASSIFIER SECARA REAL TIME*. http://repository.unpad.ac.id/frontdoor/index/index/docId/8064

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data 2021 8:1*, *8*(1), 1–74. https://doi.org/10.1186/S40537-021-00444-8

Arefnezhad, S., Samiee, S., Eichberger, A., & Nahvi, A. (2019). Driver Drowsiness Detection Based on Steering Wheel Data Applying Adaptive Neuro-Fuzzy Feature Selection. *Sensors 2019, Vol. 19, Page 943*, *19*(4), 943. https://doi.org/10.3390/S19040943

Aryani, K. A., Divayana, D. G. H., & Wirawan, I. M. A. (2017). Sistem Pakar Diagnosis Penyakit Jerawat di Wajah dengan Metode Certainty Factor. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, *6*(2), 96. https://doi.org/10.23887/janapati.v6i2.11496

BELLA, M. A. (2021). *Implementasi Algoritma Deep Learning Untuk Sistem Deteksi Kantuk Pada Pengemudi Menggunakan Yolo*.

Biar Tidak Terserang Microsleep Saat Mudik? Ini Saran Dokter. (n.d.). Retrieved September 19, 2022, from https://health.detik.com/berita-detikhealth/d-

3539619/biar-tidak-terserang-microsleep-saat-mudik-ini-saran-dokter

Chieh, T. C., Mustafa, M. M., Hussain, A., Hendi, S. F., & Majlis, B. Y. (2005). Development of vehicle driver drowsiness detection system using electrooculogram (EOG). *2005 1st International Conference on Computers, Communications and Signal Processing with Special Track on Biomedical Engineering, CCSP 2005*, 165–168. https://doi.org/10.1109/CCSP.2005.4977181

I. P. A. E. D. U. Udayana and P. G. S. C. Nugraha. (2020). Prediksi Citra Makanan Menggunakan Convolutional Neural Network Untuk Menentukan Besaran Kalori Makanan. *J. Teknol. Inf. Dan Komput.*, *6*(1), 30–38.

Janjua, M. S., & Safdar, I. (2021). *Artificial Intelligence based Driver Drowsiness Application*. https://doi.org/10.6084/m9.figshare.14905440.v1

Jimmy Pujoseno, 14611160. (2018). *IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI ALAT TULIS*. https://dspace.uii.ac.id/handle/123456789/6478

Kahlon, M., & Ganesan, S. (2018). Driver Drowsiness Detection System Based on Binary Eyes Image Data. *IEEE International Conference on Electro Information Technology*, *2018-May*, 209–215. https://doi.org/10.1109/EIT.2018.8500272

Kementerian Komunikasi dan Informatika. (n.d.). Retrieved December 28, 2021, from https://kominfo.go.id/index.php/content/detail/10368/rata-rata-tiga-orang-meninggal-setiap-jam-akibat-kecelakaan-jalan/0/artikel_gpr

Kenali Algoritma Machine Learning dan Jenisnya - Coding Studio. (n.d.). Retrieved September 19, 2022, from https://codingstudio.id/algoritma-machine-learning/

Khushaba, R. N., Kodagoda, S., Lal, S., & Dissanayake, G. (2011). *Intelligent driver drowsiness detection system using Uncorrelated Fuzzy Locality Preserving Analysis*. 4608–4614. https://doi.org/10.1109/IROS.2011.6094405

Krizhevsky, A.,Sutskever I., &Hinton G. E. (2012). ImageNet Classification

withDeep Convolutional Neural Networks. *InProceedings of NIPS*.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature 2015 521:7553*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Mehta, S., Dadhich, S., Gumber, S., & Jadhav Bhatt, A. (2019). Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3356401

Mengenal Drowsy Driving Plus Berbagai Penyebabnya - Otomotif Tempo.co. (n.d.). Retrieved December 28, 2021, from https://otomotif.tempo.co/read/1523002/mengenal-drowsy-driving-plus-berbagai-penyebabnya

Mengenal Machine Learning. (n.d.). Retrieved September 19, 2022, from https://inixindojogja.co.id/mengenal-machine-learning/

Mordvintsev, A. (2017). *OpenCV-Python Tutorials Documentation Release 1*.

Pachouly, S., Bhondve, N., Dalvi, A., Dhande, V., Bhamare, N., Professor, A., & Student, B. E. (2020). *Driver Drowsiness Detection using Machine Learning*. *8*. https://doi.org/10.22214/ijraset.2020.6324

Puspaningrum, E. Y., & Saputra, W. S. J. (2018). DETEKSI WAJAH DENGAN BOOSTED CASCADE CLASSIFIER. *Scan : Jurnal Teknologi Informasi Dan Komunikasi*, *13*(3), 15–18. https://doi.org/10.33005/SCAN.V13I3.1367

Putri, R. K. S. C. (2018). *IMPLEMENTASI DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK UNTUK KLASIFIKASI GAMBAR (Studi Kasus: Klasifikasi Gambar Pada Tanaman Anggrek Bulan Putih, Anggrek Dendrobium, dan Anggrek Ekor Tupai)*.

Rigane, O., Abbes, K., Abdelmoula, C., & Masmoudi, M. (2018). A fuzzy based method for driver drowsiness detection. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, *2017-October*, 143–147. https://doi.org/10.1109/AICCSA.2017.131

Risdin, F., Mondal, P. K., & Hassan, K. M. (2020). Convolutional neural networks (CNN) for detecting fruit information using machine learning techniques. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *22*(2), 01–13.

Saini, V., & Saini, R. (n.d.). *Driver Drowsiness Detection System and Techniques: A Review*. Retrieved August 7, 2021, from www.ijcsit.com

Saranya, L., Hariharan, P., Gopinath, A., & V, D. K. (2021). *Drowsiness Detection and Alert System Using Open Cv*. *25*(5), 3608–3615.

Susanti, R., & Fadillah, N. (2019). Deteksi Wajah Secara Real Time Menggunakan Metode Camshift. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *3*(2), 133. https://doi.org/10.30865/MIB.V3I2.1113

Udayana, I. P. A. E. D., & Supartha, I. K. D. G. (2021). Implementasi Kombinasi Metode Mean Denoising dan Convolutional Neural Network pada Facial Landmark Detection. *Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI, 10*(1), 1–10.

Vesselenyi, T., Moca, S., Rus, A., Mitran, T., & Tătaru, B. (2017). Driver drowsiness detection using ANN image processing. *IOP Conference Series: Materials Science and Engineering*, *252*(1), 012097. https://doi.org/10.1088/1757-899X/252/1/012097

Yu, J., Park, S., Lee, S., & Jeon, M. (2019). Driver Drowsiness Detection Using Condition-Adaptive Representation Learning Framework. *IEEE Transactions on Intelligent Transportation Systems*, *20*(11), 4206–4218. https://doi.org/10.1109/TITS.2018.2883823

Yuk Kenali Macam-Macam Algoritma Machine Learning! (n.d.). Retrieved September 19, 2022, from https://dqlab.id/kenali-macam-algoritma-machine-learning#heading-conten-hero-0