BAB I
PENDAHULUAN

1.1 Latar Belakang

Keselamatan pengendara merupakan aspek krusial dalam mengurangi angka kecelakaan lalu lintas yang masih tinggi secara global. Kecelakaan lalu lintas disebabkan oleh beberapa faktor seperti faktor individu, faktor kondisi mesin kendaraan, bahkan ada juga pemicu eksternal lainnya seperti kondisi sarana dan prasarana yaitu jalanan yang dilalui pengendara yang kadang tidak aman untuk dilalui. Menurut laporan tahunan *World Health Organization (WHO)* menyatakan keselamatan berkendara pada tahun 2023 menunjukkan tentang jumlah kecelakaan lalu lintas mencapai diangka 1,19 juta laporan (*Global Status Report on Road Safety* 2023). Hal ini menunjukkan betapa pentingnya upaya untuk meningkatkan keselamatan jalan raya sehingga memberikan pengurangan yang signifikan akan kecelakaan lalu lintas.

Kantuk merupakan salah satu faktor yang menyebabkan kecelakaan lalu lintas dari segi kesalahan manusia atau *human error.* Faktor tersebut termasuk kedalam 10% penyebab kecelakan lalu lintas dan memakan korban jiwa untuk beberapa kasus (*American Automobile Association*, 2018). Fakta tersebut menunjukkan bahwa mengantuk saat berkendara adalah hal serius yang sudah sepatutnya tidak boleh untuk diabaikan. Akan tetapi, pengendara biasanya tetap memaksakan diri untuk tetap mengendarai kendaraannya walaupun sudah ada tanda-tanda mengantuk yang dapat membahayakan dirinya selama berkendara. Hal tersebut disebabkan pengemudi biasanya tidak dapat menilai atau mendeteksi bahwa dirinya tersebut telah mengalami kantuk pada tahap kantuk tertentu. Tidur dapat diklasifikasikan dalam 3 tahap dan salah satu tahapnya adalah *Non-Rapid Eye Movement Sleep* atau disingkat NREM (Sahayadhas et al., 2012). Fase tersebut dapat memiliki beberapa fase lagi yaitu fase mengantuk, tidur ringan, dan tidur berat (Brodbeck et al.,2012). Berdasarkan *American Sleep Association* (AAA), terdapat beberapa faktor yang menyebabkan kantuk pada siang hari yaitu lingkungan saat ingin tidur tidak nyaman pada saat malam hari, durasi tidur yang tidak ideal dimana seorang individu dianjurkan untuk mendapatkan 8 jam tidur setiap harinya agar dapat mengurangi rasa kantuk disiang hari sekaligus rehabilitasi bagi sel sel tubuh yang rusak, efek obat juga salah satu faktor yang menyebabkan kantuk saat siang hari karena dalam komponen obat memiliki pemicu untuk membuat sel tubuh menjadi *rileks* tetapi disatu sisi meningkatkan rasa kantuk secara signifikan.

Dengan perkembangan peradaban dan teknologi, hal tersebut bisa diatasi dengan berbagai macam cara dan pendekatan. Terdapat beberapa metode untuk mendeteksi kantuk yang mungkin dialami oleh pengendara dimana dapat melalui beberapa aspek untuk menjadi parameter yaitu *vehicle-based measures, behavioral measure, dan physiological measure* (Fan et al., 2007). Aspek tersebut dapat dikombinasikan sehingga dapat memberikan pengalaman lebik baik kepada calon pengguna. Kombinasi tersebut adalah akan menggabungkan antara aspek behavioral measure dan physiological measure dimana dengan kombinasi ini, meningkatkan akurasi untuk mendeteksi kantuk yang dialami oleh pengendara berdasarkan intensitas pengendara menguap ataupun
padukan dengan monitoring vital pengendara berupa detak jantung
innya (Sahayadhas et. al., 2012).

parameter tersebut, dibutuhkan teknologi *wearable* sehingga dapat
ometrik dan diolah agar mendapatkan keluaran yang ditargetkan
kantuk pengendara (Misbhauddin, 2019). Oleh karena itu,
atch berperan penting dalam penelitian ini karena melalui integrasi

dengan *Apple HealthKit*, parameter fisiologis tersebut dapat dimanfaatkan sebagai indikator awal kondisi kantuk pengendara (Jaworski & Park, 2023).

Hingga saat ini, banyak penelitian yang telah membahas pendeteksian kantuk agar membantu pengendara menjadi lebih aman dan nyaman saat berkendara. Tetapi banyak keterbatasan jika melihat dari keluaran penelitiannya yaitu penelitian yang mendeteksi nya tidak secara real-time, penelitian berfokus hanya mendeteksi detak jantung pengendara, dan lainnya (Choi et al., 2018). Meskipun telah banyak penelitian yang mengangkat deteksi kantuk dengan pendekatan visual atau fisiologis, belum ditemukan studi yang secara spesifik mengembangkan sistem berbasis ekosistem *Apple* yang mengintegrasikan data vital dari *HealthKit* dan *machine learning*. Padahal, potensi dari integrasi ini cukup besar untuk memberikan solusi real-time yang efisien dan terintegrasi langsung dalam perangkat yang digunakan sehari-hari oleh pengguna.

Machine learning bawaan apple disebut dengan *createML*, aplikasi ini membantu dalam mengolah data yang sudah dikumpulkan sebelumnya berupa data dari fisiologis dan behavioral pengendara. Aplikasi ini memiliki *interface* sederhana dan sangat efisien untuk digunakan tetapi *createML* memiliki limitasi diaspek tertentu yang cukup krusial dalam membangun model machine learning yang sesuai (Fuller et al., 2022). Keterbatasan tersebut ada di bagian *fine tunning* model yang dibangun sehingga model tidak fleksible untuk di eksplorasi lebih lanjut (Thakkar, 2019). Oleh karena itu dibutuhkan alternatif lain agar bisa mengatasi keterbatan tersebut, *createML* akan dikombinasikan dengan *framework* lainnya dimana dengan hal tersebut, penelitian ini akan lebih fleksibel dan dapat dikustomisasi secara struktural untuk memenuhi kebutuhan pemodelan lanjutan dibagian tertentu itu. Alasan lain pemilihan framework dukungan menjadi alat untuk membangun model adalah karena penelitian ini dibangun berbasis ekosistem *apple*, maka dibutuhkan ekstensi *.mlmodel* agar model *machine learningnya* dapat diimplementasikan dalam aplikasi yang dibangun (Thakkar, 2019). Dengan penggunaan *tersebut*, setelah model mencapai target yang diinginkan maka akan dilakukan konversi agar *keras* dapat diubah menjadi *.mlmodel* sehingga dapat diintegrasikan ke dalam aplikasi berbasis iOS

Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sistem deteksi kantuk saat berkendara berbasis ekosistem Apple melalui pemanfaatan data vital dari *HealthKit* dan pengolahan data menggunakan *machine learning*. Sistem ini diharapkan dapat menjadi solusi preventif berbasis teknologi yang membantu menurunkan angka kecelakaan lalu lintas akibat kantuk (Singh et al., 2023). Penelitian ini diharapkan tidak hanya berkontribusi dalam pengembangan teknologi deteksi kantuk, tetapi juga meningkatkan keselamatan pengendara secara real-time melalui sistem berbasis ekosistem Apple yang aman dan efisien (Fathima et al., 2024).

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dikemukakan, terbentuklah beberapa rumusan masalah berikut :

1. Bagaimana merancang sistem deteksi kantuk berbasis perangkat Apple (*Apple* *iPhone*) dengan memanfaatkan data vital yang diperoleh melalui Kit?
   nengintegrasikan data vital dari *Apple HealthKit* dengan model *ning* untuk mendeteksi kantuk secara real-time?
   nengevaluasi efektivitas sistem dalam mendeteksi kantuk dan peringatan kepada pengendara melalui notifikasi ?

1.3 Batasan Masalah

Batasan masalah merupakan indikator pembatas yang bertujuan untuk menjaga penelitian tetap berada dalam lingkup objektif yang ingin dicapai, berdasarkan fenomena yang terjadi, yaitu kantuk yang dialami pengendara dan berpotensi membahayakan keselamatan saat berlalu lintas. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem deteksi kantuk hanya dibangun untuk dijalankan pada perangkat berbasis ekosistem *Apple* yaitu *Apple Watch* dan *iPhone* dengan dukungan *HealthKit* dan *machine learning*.
2. Parameter fisiologis yang digunakan terbatas pada data vital yang tersedia melalui *HealthKit*.
3. Evaluasi sistem difokuskan pada akurasi klasifikasi kantuk serta respon sistem memberikan peringatan berupa getaran ataupun notifikasi, bukan pada aspek medis ataupun diagnosis klinis. Selain itu, evaluasi yang digunakan untuk melihat kriteria sukses dari rancang bangun ini adalah *user acceptance testing* (UAT).

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem deteksi kantuk saat berkendara berbasis ekosistem *Apple* dengan pendekatan fisiologis melalui data vital *healthKit* dengan integrasi *machine learning*. Adapun tujuan khusus dari penelitian ini adalah sebagai berikut:

1. Merancang sistem deteksi kantuk berbasis perangkat *Apple* (*Apple Watch* dan *iPhone*) dengan pemanfaatan data vital yang diperoleh melalui *Apple HealthKit.*
2. Mengintegrasikan data vital dari *Apple HealthKit* dengan model *machine learning* untuk mendeteksi kantuk secara *real-time*.
3. Mengevaluasi efektifitas sistem dalam mendeteksi kantuk dan memberikan peringatan kepada pengendara melalui notifikasi

1.5 Landasan Teori

1.5.1 Teori Dasar Deteksi Kantuk

Mengantuk adalah kondisi seorang individu yang ditandai dengan berkurangnya kewaspadaan dan fungsi kognitif dan dapat berpengaruh pada kinerja dan meningkatkan resiko kecelakaan lalu lintas. Kondisi tersebut dapat didefinisikan sebagai kondisi indvidu mengalami penurunan kecepatan pemrosesan ataupun respon yang dapat diberikan kepada suatu kejadian dan salah satunya adalah pada saat berkendara (Dutta et al., 2020) (Strohl, 2012). Meskpun rasa kantuk dianggap sepele bagi sebagian orang tetapi implikasinya dapat berdampak sangat besar dan berisiko apalagi di lingkungan yang membutuhkan tingkat fokus yang tinggi seperti mengemudi (Holsmann, 2008).

Penelitian menunjukkan bahwa tanda tanda fisiologis ataupun data vital seperti detak jantung, suhu tubuh, dan variabilitas rata rata detak jantung dapat menjadi parameter yang efektif dalam memprediksi tingkat kantuk (Garcia-Perez et. al, 2022).
elevan karena tanda tanda vital ataupun fisiologis ini terjadi sebelum
kantuk seperti menguap ataupun mata tertutup sehingga
eteksi secara non-intrusif dan objektif. Penurunan variabilitas detak
jukkan peningkatan kelelahan dan penurunan kesiapan tubuh yang
rasa kantuk. Akan tetapi, terdapat beberapa pendekatan lainnya
isual yang mengamati ekspresi wajah pengendara. Indikator yang
puti kedipan mata pengendara yang mengecek intensitas menutup
Selain itu menghitung durasi penutupan mata (*PERCLOS*) adalah

indikator lainnya yang bersifat visual yang dapat dijadikan salah satu cara yang membantu mendeteksi kantuk saat berkendara.

Penggabungan pendekatan visual dan fisiologis merupakan pendekatan yang dipilih dalam penelitian ini. Sistem yang menggabungkan kedua pendekatan tersebut diharapkan mampu meningkatkan akurasi deteksi kantuk hingga 20-30% dibanding sistem yang berbasis satu jenis data saja (Garzia-Perez et. al. 2022)

1.5.2 Sistem Sensor Pada Ekosistem *Apple*

Ekostem *Apple* terdiri atas beberapa platfrom yaitu *iPhone, Apple Watch, iPad, Mac* dan ekosistem ini bisa saling berhubungan satu sama lain untuk memudahkan pengguna dalam meningkatkan pengalamanya. Jika membahas terkait sistem sensor itu sendiri, platfrom yang condong akan hal tersebut adalah *Apple Watch*. Perangkat *wearable* resmi dari *Apple* yang dilengkapi dengan berbagai sistem sensor yang canggih yang dapat membantu monitoring data vital ataupun fisiologis pengguna atau pengendara secara *real-time*. Salah satu sensor utama dari perangkat ini adalah *Optical Heart Sensor* yang menggunakan teknologi *photoplethysmography (PPG)*. Sensor ini dapat memancarkan cahaya hijau dan inframerah ke kulit pergelangan tangan dan mengukur perubahan pantulan cahaya yang terjadi akibat perubahan volume darah saat jantung berdenyut. Dengan cara ini, *Apple Watch* dapat merekam detak jantung secara akurat dan kontinu (Apple Watch Paper, 2024).

Data dari sensor tersebut dapat diintegrasikan melalui platfrom *HealthKit* yang merupakan *framework Apple* untuk mengelola dan menyimpan semua data data vital secara terpusat dan aman. *Framework* tersebut memungkinkan untuk aplikasi yang akan dibangun dapat mengakses data vital seperti detak jantung, HRV, dan lainnya sehingga membantu dalam pengembangan sistem deteksi kantuk bagi pengendara.

Keunggulan Sistem sensor dari ekosistem *Apple* berupa *iPhone* dan *Apple Watch* terletak pada kemampuannya untuk menggabungkan data fisiologis dan data visual secara *realtime* dengan akurasi yang memadai dan sangat mendukung rancang bangun untuk mendeteksi kantuk secara *non intrusive* dan berkelanjutan. *Apple Watch* dan *HealthKit* menyediakan fondasi teknologi yang kuat dalam pengembangan sistem deteksi kantuk berbasis data vital dalam ekosistem *Apple*

1.5.3 *Machine learning* untuk Deteksi Kantuk

Machine learning merupakan metode yang sangat efektif untuk mengklasifikasikan kondisi kantuk pengendara berdasarkan data fisiologis dan perilaku pengemudi. Algoritma *machine learning* dapat digunakan untuk mengenali pola-pola dari fitur yang diekstraksi dari data sensor dan di kombinasikan dengan data visual seperti kedipan mata dan durasi penutupan mata pengemudi.

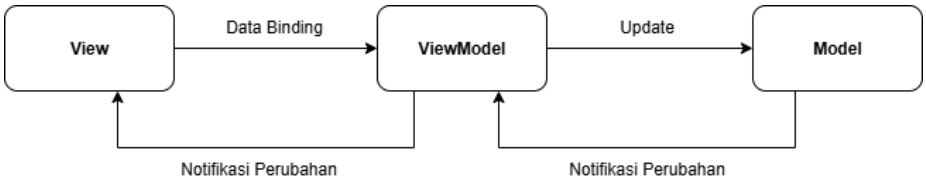
API berbasis *Python* untuk membangun dan melatih model dan banyak digunakan dalam pengembangan sistem deteksi kantuk karena kemudahan penggunaan dan dukungan berbagai jenis *CNN*. Model *CNN* dapat menjadi salah satu opsi untuk mendeteksi tanda-tanda kantuk seperti frekuensi kedipan mata. *CNN* mampu
ari gambar secara otomatis tanpa perlu ekstraksi fitur manual,
an performa klasifikasi dengan mempertimbangkan matriks yang
egi akurasi, *f1 score*, *recall*, dan *precission* agar model yang
gan objektif target yang diinginkan.

model *multilayer perceptron* (*MLP*) juga dapat digunakan untuk
erbasis data fisiologis atau data vital. Studi menunjukkan bahwa
ada data fisiologis dapat mencapai akurasi hingga 86% dalam

membedakan pengendara dalam kondisi kantuk dan terjaga (Wulandari, 2021). Hal ini memungkinkan integrasi dengan input data lebih dari satu (*multimodal*) seperti data vital dan data visual. Selain itu, *CreateML* mendukung Teknik *augmentasi* data dan *fine tunning* data sehingga dapat mengatasi *output* model yang *overfitting* ataupun *underfitting* pada model dimana hal ini tidak didapatkan jika menggunakan *createML* bawaan dari *Apple*.

1.5.4 SwiftUI

SwiftUI adalah *framework* deklaratif yang dikembangkan oleh *Apple* untuk membangun antarmuka pengguna pada platfrom *Apple* seperti *iOS, iPadOS, macOS, watchOS, dan tvOS. Framework* ini pertama kali diperkenalkan pada tahun 2019 dimana memungkinkan pengembang untuk membuat tampilan secara efisien dengan menuliskan kode yang lebih ringkas dan mudah dipahami menggunakan bahasa pemrograman *Swift*. Selain itu, *SwiftUI* menyediakan fitur *preview real-time* yang memudahkan pengembang dalam melihat hasil desain secara langsung selama proses pengembangan (Wiratama & Santoso, 2019).



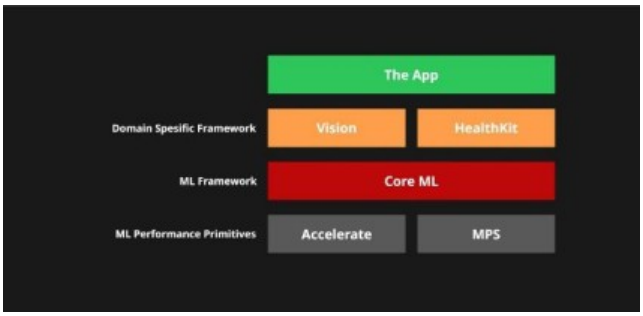
Gambar 1 Arsitektur MVVM

Pada **Gambar 1 Arsitektur MVVM** Selain aspek antarmuka yang deklaratif, *SwiftUI* diorgansasikan dengan pola arsitektur *Model-View-ViewModel (MVVM)* untuk memisahkan logika dari tampilan. Arsitektur bagian model merepresentasikan data mentah dari aplikasi dimana untuk penelitian ini akan berkaitan dengan data dari detak jantung, HRV, dan lainnya yang diambil dari HealthKit. Selanjutnya *ViewModel* berfungsi sebagai lapisan penghubung yang menyiapkan data siap pakai dimana dalam ViewModel biasanya terdapat fungsi yang mengolah data dari model sebelum ditampilkan kedalam *view* di *swiftUI* melalui *property wrapping binding, state*, dan sebagainya (*Worldwide Developers Conference*, 2025). Pendekatan MVVM meningkatkan pemeliharan kode menjadi lebih mudah dan lebih tertata sehingga dapat melakukan optimisasi kode secara tidak langsung. Kombinasi SwiftUI dengan MVVM memungkinkan pembangunan sistem deteksi kantuk yang modular, responsif serta mudah dikembangkan lebih lanjut.

1.5.5 Core ML

Core ML adalah sebuah *machine learning framework* yang dikembangkan oleh *Apple* Inc. dan pertama kali diperkenalkan pada tahun 2017 dimana bertujuan untuk *machine learning* ke dalam sebuah aplikasi *Apple* dengan berbagai *e ML* memiliki banyak fitur yang dapat dimanfaatkan berdasarkan ng ingin dibangun. Hal tersebut dapat berupa *image classification, object detection, tabular data, regression*, dan lain sebagainya. berbagai macam pendekatan tersebut agar pengembang dapat i dan menciptakan aplikasi yang diinginkan dan memecahkan alaupun *Core ML* dapat diimplementasikan secara eksplisit dalam

suatu *text editor* tetapi terdapat aplikasi *interface* yang memisahkan jika ingin membangun model *machine learning* yaitu *CreateML* dimana dengan antarmuka sederhana diharapkan pengembang dapat dengan mudah mengembangkan model *machine learning* dengan berbagai macam jenis pendekatan yang telah disebutkan (Thakkar, 2019).



Gambar 2 Apple ML Stack

Pada **Gambar 2 Apple ML Stack** *Framework* ini juga memiliki teknologi turunan lain dimana berhubungan erat dengan *Core ML* yaitu *framework speech* yang secara signifikan meningkatkan pengalaman pengembang dalam membangun aplikasi berbasis *speech* atau *voice* dimana dengan eksplorasi lebih lanjut, seorang pengembang dapat menciptakan keluaran yang berbeda beda dengan *framework* tersebut (Sahin, 2021). Dengan demikian, *Core ML* adalah kemajuan yang signifikan dalam mendukung pembuatan *machine learning* model yang dapat diakses dan efisien bagi pengembang.

1.5.6 Vision

Vision juga salah satu *framework* yang dikembangkan oleh *Apple* dimana *framework* ini dirancang dengan *inputan* sebuah gambar dan menganalisisnya. Pengembang yang membuat sistem pengenalan visual, akan sangat terbantu dengan adanya *framework* ini. *Vision* terintegrasi langsung dengan *Core ML* dimana memungkinkan untuk menciptakan model yang lebih baik dengan kemampuan pemrosesan gambar (Marques, 2020).

Misalnya, dalam aplikasi pertanian, seperti mengklasifikasikan varietas apel, *framework* ini dapat menggunakan algoritma seperti *k-nearest neighbor* untuk menganalisis karakteristik fisik dari apel. Dan untuk matriks tertentu dapat mencapai ditingkat 84-97% berdasarkan warna dan bentuk (Nasution & Amrullah, 2022). Selain itu, *Vision framework* berkontribusi cukup signifikan dalam dunia robotik karena respon dari setiap analisis *input* gambar yang diterima adalah salah satu aspek robotik yang terjadi di perkembangan teknologi masa kini. Salah satu contohnya adalah robot pemetik apel yang membantu pengenalan target dan menyortir yang perlu dipetik sehingga berdampak pada kinerja dan efektifitas biaya (Liu et at., 2023).

Dengan dasar *framework* ini, *Apple* sendiri mengembangkan implementasi
ar lebih membantu setiap calon penggunanya diberbagai bidang.
tasi nyata dari *framework* ini adalah terciptanya salah satu produk
ıaan nama dengan *framework* yaitu *Apple Vision Pro* dimana
man yang lebih dekat dengan dunia nyata seorang pengguna.

1.5.7 Xcode

Sama halnya seperti *Visual Studio Code*, aplikasi *Xcode* ini ekslusif untuk mengembangkan aplikasi di platform *apple* dimana umumnya menggunakan bahasa pemrograman *swift*. *Xcode* ini merupakan salah satu *text editor* dalam dunia pengembang yang cukup populer. Aplikasi ini memfasilitasi pengkodean dan pembuatan aplikasi yang efisien melalui alat alat seperti *interface builder* dan fitur fitur *debugging* tingkat lanjut (Knot, 2018). Secara arsitektural, Xcode menyediakan *single window workflow* yang terhubung erat dengan *framework Cocoa/Cocoa Touch* serta dilengkapi *runtime* dan *profiler (Instruments)* untuk analisis performa, manajemen memori, dan *footprint energy*. Kehadiran *LLDB debugger*, *refactoring otomatis*, dan fitur *code completion* berbasis indeks *clang* terbukti meningkatkan produktivitas serta menurunkan jumlah *defect* pada fase awal pengembangan. Evolusi *Xcode* versi terbaru menambahkan integrasi *source control* berbasis *Github*, dukungan *Swift Concurrency* serta *XcodeCloud* sebagai layanan CI/CD yang terotomasi di lingkungan *Apple*, sehingga siklus rilis aplikasi dapat dilakukan dengan lebih terukur. Dengan kombinasi fitur tersebut, *Xcode* tidak hanya dipandang sebagai *text editor* melainkan ekosistem pengembangan komprehensif yang mendukung praktik industri pada platform *Apple* secara berkelanjutan.

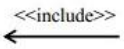
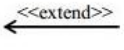
1.5.8 TestFlight

Seorang pengembang di platform *Apple* harus memastikan aplikasi telah memenuhi standar yang sesuai agar memastikan pengalaman untuk calon penggunanya mencapai di titik yang ditargetkan diawal. TestFlight adalah alat penting bagi seorang pengembang dimana memfasilitasi pengujian *beta* aplikasi dengan memungkinkan pengembang mengundang tester untuk menguji aplikasi pada versi prarilis sebelum benar benar *deploy* kedalam app store. Proses ini melibatkan pembuatan tautan undangan yang dapat diakses dengan mudah oleh tester, meningkatkan efisiensi pengujian dengan menghilangkan kebutuhan untuk entri kode manual atau navigasi email (Zhiying et al., 2019). Selain itu, strategi pengujian yang efektif, termasuk otomatisasi UI, sangat penting untuk mengidentifikasi bug dan memastikan fungsionalitas aplikasi, yang dapat dicapai dengan menggunakan alat bawaan *Apple* (Penn, 2013). Seiring berkembangnya dunia pengembang, kompleksitas aplikasi juga semakin meningkat, pendekatan pengujian komprehensif menjadi semakin penting dimana kemanan dan kinerja sangat penting dalam suatu aplikasi atau sistem (Kulesovs et at., 2018).

1.5.9 Unified Modelling Language (UML)

UML atau disebut dengan Unified Modelling Language adalah bahasa visual semiformal yang biasanya digunakan saat ingin membangun suatu sistem, produk, ataupun pengembangan perangkat lunak dimana menyederhanakan sistem yang miliki *complexity* yang cukup tinggi menjadi visualisasi yang merepresentasikan hal tersebut dalam bentuk yang lebih sederhana. Bahasa ini digunakan sebagai alat komunikasi
ebih mudah dipahami walaupun divisi atau rekan tim tidak memiliki
yang memadai. UML ini memvisualisasikan berbagai macam
am berbagai jenis diagram termasuk diagram *use case, sequence,*
a (Schroeder, 2022). Sebagian besar dalam pengembangan
enggunakan UML dimana hal tersebut diakui secara luas dan
ak alat dan menjadikannya aset penting dalam pengembangan
nodel (Selic, 2008).

Diagram *use case* (UCD) merupakan komponen krusial dari *Unified Modeling Language (UML)* yang secara visual merepresentasikan interaksi antara pengguna (aktor) dan fungsionalitas sistem, secara efektif menguraikan apa yang seharusnya dicapai sistem tanpa perlu mendalami detail implementasi (Seidl dkk., 2015). UCD memfasilitasi pemahaman perilaku sistem, yang penting untuk pengembangan dan pemeliharaan perangkat lunak, terutama saat merekayasa balik sistem yang ada untuk mengekstrak persyaratan (Andriyani dkk., 2022). Kemajuan terbaru mencakup pembuatan UCD secara otomatis dari cerita pengguna, yang meningkatkan akurasi dan efisiensi dalam menangkap persyaratan perangkat lunak (Fathiyah & Widyani, 2024). Lebih lanjut, perangkat metodologis seperti *UCCheck* membantu dalam pembuatan dan peninjauan UCD, mendorong kepatuhan terhadap aturan formal dan meningkatkan kualitas elaborasi use case (Aquino dkk., 2020). Namun, masih terdapat tantangan dalam menetapkan semantik yang tepat untuk UCD, yang vital untuk membedakan berbagai diagram dan maknanya (Kautz dkk., 2022).

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*
	*Use case* : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

Gambar 3 Use Case Diagram

Pada **Gambar 3 Use Case Diagram** *Activity diagram*, komponen kunci dari *anguage (UML),* berfungsi untuk memodelkan alur tindakan dalam menangkap aliran kontrol dan data antar berbagai aktivitas. revolusi secara signifikan dari UML 1 ke UML 2, dengan tambahan eningkatkan penerapannya di berbagai bidang, termasuk proses atika (Siebenhaller & Kaufmann, 2006) (Seidl dkk., 2015). Diagram untuk merepresentasikan aspek prosedural dan dapat memodelkan objek maupun non-objek (Seidl dkk., 2015). Namun, diagram ini

bukannya tanpa tantangan; masalah seperti *livelock* dan *deadlock* dapat muncul karena kurangnya aturan penggambaran formal dan inkonsistensi semantik (Sugunnasil, 2017) (Sugunnasil, 2016). Pada **Gambar 4 Activity Diagram** Penelitian terbaru telah mengusulkan metode formal, termasuk ekspresi proses dan automata, untuk mendeteksi kesalahan perilaku ini, sehingga meningkatkan keandalan diagram aktivitas (Sugunnasil, 2017) (Sugunnasil, 2016). Selain itu, perangkat otomatis telah dikembangkan untuk memfasilitasi transformasi diagram sekuens menjadi diagram aktivitas, yang selanjutnya menyederhanakan proses pemodelan (Kulkarni & Srinivasa, 2021).

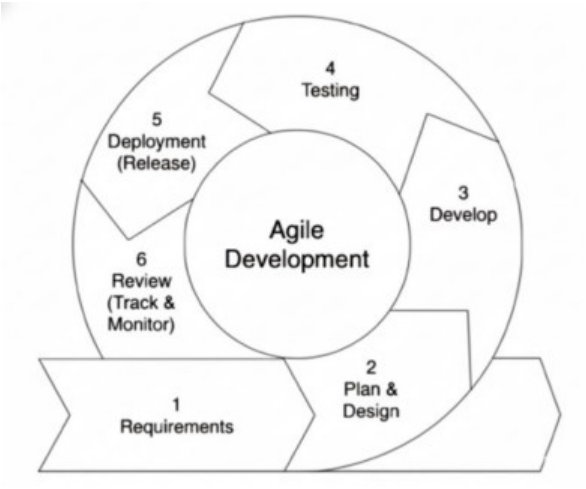
Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem,aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 4 Activity Diagram

1.5.10 Agile

Pada **Gambar 5.** Metodologi *Agile* sudah lazim muncul sebagai kerangka kerja penting dalam manajemen suatu proyek dimana hal ini juga digunakan dalam pengembangan perangkat lunak dan karena keefektifannya dalam implementasinya, metodologi ini sudah mulai banyak diadopsi di berbagai industri karena fleksibilitas yang melekat dan fokus yang berpusat pada pengguna atau pelanggan (Daraojimba et al., 2024).

ji ini, termasuk Scrum dan Kanban, meningkatkan kinerja proyek ng motivasi pengerjaan dan kemampuan beradaptasi terhadap enting dalam dunia teknologi yang berkembang pesat saat ini.



Gambar 5 Agile Model

1.5.11 Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada evaluasi fungsionalitas aplikasi dari perspektif pengguna tanpa mempertimbangkan struktur kode internal. Pendekatan ini banyak digunakan di berbagai aplikasi untuk memastikan perangkat lunak memenuhi persyaratan pengguna dan berfungsi sesuai harapan (Mumtaz dkk., 2024). Metode ini juga mencakup beberapa teknik seperti *Equivalence Partitioning, Boundary Value Analysis, dan State Transition Testing*, yang diterapkan dalam berbagai skenario. Oleh karena itu, *Black box testing* merupakan metode yang serbaguna dan penting untuk memverifikasi fungsionalitas dan keandalan perangkat lunak di berbagai aplikasi.

1.5.12 User Acceptance Testing

User Acceptance Testing (UAT) adalah salah satu tahap pengujian aplikasi dimana berfokus pada pendapat dari calon pengguna nantinya. Dimana tujuannya adalah untuk memastikan produk atau sistem yang dibangun dapat memenuhi kebutuhan dan dapat mencapai kinerja sesuai dengan spesifikasi yang ditargetkan di awal fase pengembangan. UAT sangat efektif untuk memastikan kualitas dengan mempertimbangkan waktu dan biaya perangkat lunak dan mendukung kemistri terhadap calon pengguna dan meningkatkan kepercayaan pengguna terhadap komunitas yang akan dibangun.

1.6 Penelitian Terkait

Berdasarkan **Tabel 1 Literatur Penelitian Terdahulu** Pada penelitian yang berjudul "Deteksi Rasa Kantuk Pengendara Kendaraan Bermotor Menggunakan *Image* Setiawan & Marsiska Ariesta, 2021). Peneliti melihat dan erapa masalah dimana pengendara yang kurang disiplin dan dalam berkendara yang mengakibatkan kecelakaan akibat hal metode pendekatan dalam penelitian ini adalah memanfaatkan ra dimana akan merepresentasikan objek dalam bentuk citra digital an memiliki keluaran tertentu. Selain itu, ada berbagai citra yang njadi parameter dalam penelitian ini yaitu citra warna, citra

grayscale, dan citra ekstraksi iris mata. Ada beberapa tahapan yang dialami sebelum mendapatkan keluaran dimana mendeteksi pengendara dalam kondisi kantuk atau terjaga yaitu dimulai proses pemotongan atau *cropping* lalu citra diolah dan dikonversi menjadi citra *grayscale* agar lebih mudah dideteksi lalu citra tersebut diekstrak untuk memisahkan bagian pupil lalu diubah menjadi citra *threshold* dan ini memudahkan dalam menghitung dari citra tersebut dan mendeteksi kantuk. Disatu sisi keluaran penelitian ini masih sampai di tahap pengolahan citra dan mendeteksi kantuk tetapi belum diimplementasikan secara *real-time* sesuai dengan calon pengguna yaitu pengendara jadi belum ada yang dapat memastikan keefektifannya. Akan tetapi, disatu sisi keakurasian sistem ini dengan menggunakan metode *bwarea* dapat mencapai 99% dimana ini angka yang cukup tinggi tapi cukup berisiko karena berpotensi *overfitting* untuk deteksi nya. Selain itu ada juga limitasi lainnya dari penelitian ini yaitu dari segi kondisi pupil mata saat pengolahan citra harus bebas dari gangguan pencahayaan agar tidak mempengaruhi proses identifikasi.

Selanjutnya, Penelitian berjudul "*Drowsiness detection in drivers with a smartwatch*" (Sonia Diaz & Pino Caballero, 2022). Penelitian ini berfokus pada teknologi dari *smartwatch* itu sendiri dimana adanya sensor bawaan dari *smartwatch* yang membantu dalam mendeteksi kantuk terhadap pengendara. Parameter yang digunakan untuk mengecek hal tersebut adalah dengan Variabilitas Detak Jantung (HRV). Sensor yang digunakan dalam penelitian ini adalah *PPG sensor* yang menggunakan *light-based technology* untuk mendeteksi aliran darah yang dikontrol oleh jantung. Penelitian ini menggunakan *hardware* untuk mencapai tujuan penelitiannya dan salah satunya adalah *Samsung galaxy watch 4 LTE* dimana membantu untuk memonitoring Kesehatan setiap hari nya dan dengan adanya sensor *BioActive* dapat mengukur *ECG* dan tekanan darah secara *real-time*. Selain dari mengukur data fisiologis pengguna degan *hardware* tersebut, *Samsung galaxy watch* juga memiliki sensor gerakan berupa *accelerometer* dan gyroscope. Penelitian ini memiliki rancangan berbasis *android* dan menggunakan *android studio* sebagai *text editor* pengembangannya dan bahasa pemrograman yang digunakan adalah *Kotlin* dan *java* karena dari *smartwatch* itu agar dapat diakses data nya lebih detail maka dapat diakses melalui *android*. *Health platfrom* bawaan Samsung juga dimaksimalkan untuk mendapatkan akses data kesehatan pengguna yang lebih detail dan lebih terstruktur. Untuk tahap pengolahan data, hal tersebut di analisis melalui dua tipe pendekatan yaitu *supervised* dan *unsupervised* dimana untuk *supervised learning* dimulai dengan melakukan *labeling* data agar atribut dapat di prediksi dengan tepat dan terbantu. Sedangkan disatu sisi, *unsupervised learning* dimulai dengan data tanpa *label* dimana ini memberikan komparasi antara dua pendekatan tersebut sehingga dapat memilih pendekatan yang terbaik dalam membangun rancang bangunnya nantinya.

Penelitian berikutnya berjudul "*Variation of the Heartbeat and Activity as an Indicator of Drowsiness at the Wheel Using a Smartwatch*" (Sergio et. at, 2015). Penelitian ini memiliki kemiripan seperti penelitian sebelumnya dimana memanfaatkan smartwatch dalam mendapatkan data berdasarkan sensor yang detak jantung yaitu *PPG sensor* dan ini menjadi salah satu sensor yang digunakan untuk menangkap data nya. Penelitian ini diawal akan mengumpulkan data yang telah ditangkap oleh sensor tadi dan selanjutnya anjut. Peneliti melakukan pengumpulan data dengan mengundang nografi antara pria dan wanita yang seimbang serta ada juga beragam. Pengumpulan data tersebut akan melakukan simulasi eberapa kondisi dari relawan yang telah diundang, para relawan kendara dengan menggunakan device pada saat kondisi normal engendarai motor dan melakukan perjalanan kurang lebih 30 menit lah satu kondisi perjalanan yang cukup singkat dan ingin diketahui

hasil dari para relawan yang melakukannya, selanjutnya diharapkan mereka mendapatkan data diman kondisi pengendara melakukan perjalanan yang cukup Panjang dengan roda empat dan melakukan istirahat setiap 2 jam perjalanan. Selanjutnya didapatkan beberapa parameter dari data atau *feature* yang akan diolah yaitu *accelerometer, heart rate monitor, pedometer, gyroscope* lalu peneliti menggunakan *FFT-Fast Fourier Transform*. Dengan *FTT*, terdapat beberapa tipe dan plot dimana peneliti menemukan pola saat pengendara mengalami kondisi kantuk. Dalam penelitian ini mendapatkan keluaran dimana fungsional aplikasi adalah memberikan peringatan kepada pengendara untuk istirahat setiap 2 jam sekali jika berkendara dalam jangka waktu yang cukup Panjang, lalu fungsionalitas lainnya adalah memberikan sebuah notifikasi yang berarti pengendara tersebut dalam bahaya saat berkendara akibat adanya indikasi kantuk yang membahayakan pengendara tersebut maupun orang lainnya.

Selanjutnya, Penelitian berjudul "Pengembangan Aplikasi Pendeteksi Kantuk Pada pengendara Kendaraan Bermotor Dengan Menggunakan Sensor Detak Jantung Pada *Smartwatch*" (Imam, Agi, Sutrisno, 2019). Peneliti juga melihat adanya urgensi dalam masalah ini dan menawarkan sebuah sistem dalam bentuk aplikasi berbasis android dimana menggunakan data detak jantung yang dimana memanfaatkan sensor detak jantung pada *smartwatch* yang dikenakan oleh pengendara. Dalam perancangannya, sistem ini dibangun dengan pendekatan *Object Oriented Design (OOD)* dan dalam implementasinya menggunakan metode *Object Oriented Programming (OOP).* Setelah itu, saat aplikasi atau sistemnya telah selesai, Peneliti melakukan beberapa pengujian untuk memastikan aplikasinya sesuai dengan standar yang ditentukan sebelumnya dan diharapkan dapat benar benar bermanfaat untuk mengatasi masalah yang sudah ada. Peneliti menggunakan pengujian *black box*, akurasi dan *usability*. Ini adalah beberapa pengujian yang dapat mewakili sebuah kualitas dari produk atau sistem itu sendiri dimana setiap pengujiannya mendapatkan hasil yang cukup memuaskan dan sudah cukup untuk digunakan oleh pengendara secara langsung. Oleh karena itu, hasil atau keluaran dari penelitiannya yaitu berupa sebuah aplikasi. Berdasarkan analisis kebutuhan dari calon pengguna, aplikasinya memiliki empat kebutuhan fungsional yaitu alarm peringatan, mengirimkan pesat darurat, menyimpan nomor darurat, dan melihat histori. Disatu sisi, aplikasi ini juga memiliki dua kebutuhan non fungsional yaitu *reliability* dan *usability*. Implementasi aplikasi tersebut dibangun dengan menggunakan bahasa pemrograman *Kotlin* dengan implementasi metode pendekatan tadi yaitu *Object Oriented Programming (OOP)*. Aplikasi ini juga diberikan pengujian dengan mengundang responden dan menggunakan secara *real-time* dan hasilnya cukup memuaskan dimana dari jumlah pengujian dengan total 22 pengujian, terdapat jumalh valid 19 sedangkan tidak valid hanya berjumlah 3 saja. Ini menunjukkan hasil yang cukup bagus yaitu sebesar 86,3%.

Terakhir adalah sebuah penelitian berjudul "Deteksi Kantuk Pengendara Roda Empat Menggunakan *Haar Cascade Classifier dan Convolutional Neural Network*" (Cahya, Danang, 2021). Penelitian ini memiliki fokus yang cukup spesifik dimana menganalisis sebuah kejadian mengantuk dalam waktu beberapa detik yaitu *microsleep*. Penelitian ini melakukan pengembangan dalam pengolahan citra digital dalam
ebut pada pengendara roda empat dengan motode pengenalan
e Clasifier dan klasifikasi menggunakan *Convolutional Neural*
ıg digunakan dalam sistem ini memanfaatkan sebuah perangkat
ı yang akan menangkap wajah pengendara selama mengendarai
Keluaran dari sistem adalah adanya suara alarm peringatan jika
si kantuk. Sistem dapat mendeteksi berbagai jenis mata dengan
100%. Dimana akurasi rata-rata yang didapatkan dengan jarak

kamera dengan wajah yaitu sekitar 30-50cm adalah 95%. Pengujian dari sistem ini dilaksanakan mengugnakan sampel sebanyak 1 dengan memasang posisi kamera dengan variasi jarak dengan pengendara yaitu 30-50cm. Sistem akan mendeteksi jika durasi mata tertutup lebih dari 3 detik maka alarm akan berbunyi tetapi jika setelah kondisi mata tersebut tertutup kemudian terbuka kembali maka perhitungan durasi akan kembali di tahap awal. Hasilnya dari 11 kali sampel dengan variasi jarak tersebut hanya terdapat 2 kali kondisi gagal dalam mendeteksi dan alarm tidak berbunyi dari total percobaan.

Tabel 1 Literatur Penelitian Terdahulu

No	Author	Tahun	Judul Penelitian	Hasil Penelitian	GAP Penelitian
1	Iwan Setiawan & Marsiska Ariesta	2021	Deteksi Rasa Kantuk Pengendara Kendaraan Bermotor Menggunakan *Image Processing*	Penggunaan metode pengolahan citra (*image processing*) untuk mendeteksi kondisi mata (ngantuk/tidak) menghasilkan tingkat akurasi cukup tinggi menggunakan metode *bwarea*	Belum membahas integrasi dengan sensor biometrik atau perangkat *wearable* untuk meningkatkan deteksi
2	Sonia Diaz & Pino Caballero	2022	*Drowsiness Detection in Drivers with a Smartwatch*	Sistem aplikasi mengumpulkan variabel fisiologis penting (dari smartwatch) pada saat mengemudi dan menghasilkan *alert*	Kurang membahas perbandingan efektivitas sensor fisik (misal kamera) dengan fisiologis, dan belum diuji secara luas di lingkungan nyata
3	Sergio Ríos Aguilar, dkk	2015	*Variation of the Heartbeat and Activity as an Indicator of Drowsiness at the Wheel Using a Smartwatch*	Menggunakan variasi detak jantung dan aktivitas fisik yang diukur dengan smartwatch (sensor HRV dan akselerometer)	Belum terintegrasi real-time dengan sistem peringatan lain seperti *image processing*; sampel terbatas

No	Author	Tahun	Judul Penelitian	Hasil Penelitian	GAP Penelitian
4	Imam, Agi, Sutrisno	2019	Pengembangan Aplikasi Pendeteksi Kantuk Pada Pengendara Kendaraan Bermotor Dengan Menggunakan Sensor Detak Jantung Pada *Smartwatch*	Aplikasi *Android* mendeteksi kantuk lewat data detak jantung yang diperoleh dari *smartwatch*, tingkat validitas sistem 100%, akurasi pengujian pada 5 responden sebesar 86,3%, *usability* "*Excellent*"	Fokus hanya pada detak jantung, tidak menggabungkan parameter lain seperti aktivitas fisik atau deteksi citra mata
5	Cahya Aji Saputra, Danang Prasetyo	2021	Deteksi Kantuk Pengendara Roda Empat Menggunakan *Haar Cascade Classifier dan CNN*	Metode *Haar Cascade Classifier* (deteksi mata ROI) dan *CNN* (state mata) mencapai keberhasilan deteksi 100% untuk klasifikasi mata terbuka/tertutup, akurasi kantuk 93.9%, waktu komputasi <0.11 detik	Belum mengintegrasikan hasil deteksi dengan data biometrik lain, serta tidak diuji pada beragam kondisi pencahayaan

BAB II
METODE PENELITIAN

2.1 Waktu dan Lokasi Penelitian

Pada **Tabel 2 Waktu Penelitian** Sebagai titik awal penulis dalam membangun sistem yang diajukan sebelumnya. Maka penulis melalui tahap yang cukup krusial adalah penyusunan waktu dan lokasi penelitian dimana ini menjadi acuan penulis yang bertujuan agar sistem rancang bangun dapat diselesaikan tepat waktu dan mencapai minimum standar yang diharapkan. Penulis memulai penelitiannya pertama kali pada bulan Juni 2025 dan ada beberapa fase yang akan di lalui mulai dari investigasi penulis untuk membangun sistem rancang bangun yang sesuai dengan kebutuhan calon pengguna atau *user,* fase dimana penulis memulai mengolah data yang didapatkan, lalu fase saat penulis memulai membangun rancang bangun yang berdasarkan hasil data yang telah dikumpulkan dan diolah sebelumnya, hingga fase saat penulis melakukan *testing* dan menyelesaikan sistem rancang bangun deteksi kantuk tersebut dimana direncanakan akan selesai pada bulan November 2025.

Tabel 2 Waktu Penelitian

No	Kegiatan Penelitian	Juni	Juli	Agustus	September	Oktober	November
1	Pengajuan Judul						
2	Penyusunan Proposal						
3	Studi Literatur & Observasi Sistem Terkait						
4	Perancangan Sistem						
5	Persiapan Model						
6	Implementasi Sistem						
7	Pengujian Sistem						

pulan Data

ah salah satu aspek pendekatan yang memanfaatkan data dari
ya yang telah dikumpulkan sehingga penulis bisa melakukan

pembaharuan dan melakukan pendekatan lainnya terhadap data yang tersedia. Metode ini cukup populer karena ketersediaan data yang cukup luas dan dapat diakses di berbagai repositori seperti perpustakaan, museum, bahkan secara daring. Pengumpulan data sekunder ini menjadi salah satu pendekatan alternatif yang dapat menghemat biaya maupun waktu penulis. Tetapi disatu sisi, relevansi data terhadap penelitian yang dilakukan oleh penulis mungkin saja kurang sesuai dan harus ada pengolahan lebih lanjut terkait data yang telah dikumpulkan. Oleh karena itu metode yang diterapkan dalam pengumpulan data sekunder adalah melalui studi literatur yang akan berfokus pada apa yang menjadi data yang disediakan oleh penelitian terkait dan selain itu pengumpulan *dataset* yang akan berfungsi untuk digunakan dalam pembangunan model *machine learning* yang dimana diharapkan untuk *datasetnya* sendiri adalah berupa *open-source* sehingga peneliti dapat secara langsung mengolah *dataset* tersebut dan melakukan pendekatan yang sesuai dan relevan dengan penelitian yang dilakukan saat ini (Trinh, 2017).

Salah satu metrik yang menjadi acuan seseorang pengendara dalam kondisi kantuk adalah menghitung metrik *PERCLOS* (persentase mata tertutup dalam periode tertentu). Oleh sebab itu, penulis akan menggunakan beberapa pendekatan yaitu melalui dataset dalam format video atau foto. Hal ini bertujuan untuk mengetahui pendekatan terbaik untuk penyusunan *model machine learning*.

Pada **Tabel 3 Deskripsi variabel dataset NTHU-DDD** *Dataset* yang menjadi potensial pertama adalah dataset NTHU-DD yang dikembangkan oleh *National Tsing Hua University*. *Dataset* ini berisi rekaman video seorang pengemudi mobil dalam berbagai macam kondisi lingkungan maupun kondisi pengendara agar membantu model machine learning dapat mendeteksi kantuk pada saat berkendara. Jumlah data dari dataset ini adalah 550 video yang terdiri atas 22 partisipan yang dilengkapi dengan 5 skenario serta untuk durasi video nya adalah 30 detik sebagai fondasi awal dari dataset yang disediakan.

Tabel 3 Deskripsi variabel dataset NTHU-DDD

No	Variabel	Tipe Data	Deskripsi
1	*subject_id*	*int*	Id untuk tiap partisipan
2	*scenario*	*String*	Kondisi cahaya & atribut
3	*frame_sequence*	*Image Seq*	Urutan gambar/frame video
4	*eye_state*	*Categorical*	*Open/closed*
5	*blink_type*	*Categorical*	*Slow/fast/none*
6	*action_label*	*Categorical*	*Normal/drowsy/yawn/blink slow/blink fast*

Selanjutnya Pada **Tabel 4 Deskripsi variabel dataset RT-BENE** sebagai dataset pembandingan adalah dataset dalam bentuk foto. Dataset tersebut Bernama RT-*l Time Blink Estimation in Natural Environments*). *Dataset* ini a yang telah dilabel dalam opsi open, closed, maupun uncertain. **ıset Value RT-BENE** *Dataset* ini juga *dataset* yang potensial untuk sistem rancang bangun dalam mendeteksi kantuk. Jumlah data dataset ini cukup besar yaitu 243.714 gambar dan Memiliki format engan resolusi 60*36 px.

Tabel 4 Deskripsi variabel dataset RT-BENE

No	Variabel	Tipe Data	Deskripsi
1	*image_id*	*String*	Nama file gambar
2	*subject_id*	*Int*	ID unik
3	*eye_patch*	*Image*	Gambar area mata
4	*eye_state*	*Categorical*	*Open/closed/uncertain*

Gambar 6 Dataset Value RT-BENE

2.3 Metode Pengembangan Sistem

Metode Pengembangan ataupun penelitian yang menjadi acuan dalam membangun sistem yang telah diajukan dalam penelitian ini adalah melakukan observasi dan studi literatur dimana Hal tersebut bertujuan untuk mengetahui sistem yang serupa dan melakukan penyesuaian sesuai dengan standar yang telah ditetapkan. Dengan berdasarkan studi literatur dari penelitian sebelumnya, penulis bisa memaksimalkan sumber daya yang ada dan penulis bisa berharap dapat menyelesaikan rumusan masalah yang telah ada (Nelson et al. 2020).

Metode yang secara teknis digunakan dalam pengembangan sistem rancang bangun adalah adalah metode *agile* karena dengan metode ini yang merupakan transformasi dari kolaborasi antara pengembangan sistem rancang bangun maupun penelitian ilmiah. Metode ini sesuai dengan sistem yang diajukan karena memberikan kesan fleksibilitas dan iterasi yang cukup cepat dan tinggi. Sistem rancang bangun ini diharapkan bisa memenuhi standar karena sistem ini akan berdasarkan oleh prinsip *customer data driven* dimana akan berfokus pada feedback untuk melihat kebutuhan pengguna terhadap sistem yang akan dibangun (Zhen, 2024). Benefit yang ditawarkan dengan menerapkan *agile methodology* adalah memungkin penulis dapat lebih adaptif acam perubahan saat ada dibeberapa fase yang ada dalam karena itu, hal ini merupakan pendekatan yang baik oleh penulis sistem yang telah diajukan. Fase yang ada dalam metodologi ahap dan biasanya dibagi dalam bentuk *sprint* untuk melihat ada atau diberikan terkait dengan sistem rancang bangun tersebut. g akan dilalui dalam pengembangan sistem rancang bangun ini dalah *requirements* dimana akan berfokus pada analisis kebutuhan

pengguna yang akan dikumpulkan melalui observasi maupun interview singkat, selanjutnya adalah tahap *design* yang akan berfokus pada membangun sistem antarmuka maupun interaksi pengguna terhadap aplikasi atau sistem rancang bangun tersebut, Tahap berikutnya tahap *development* yaitu tahap yang cukup krusial karena akan mengimplementasi dari tahap design seperti sebelumnya menjadi sebuah prototype awal untuk dilanjutkan ke tahap selanjutnya, Setelah tahap *development* adalah tahap *testing* dimana akan berfokus pada feedback terkait *prototype* awal yang telah dibuat, Selanjutnya tahap yang akan memanfaatkan *testflight* yang akan mempersiapkan *prototype* nya di *deploy*, Terakhir adalah tahap *review* yang akan ditulis dalam bentuk laporan yang membahas terkait semua proses yang telah dilalui dan melihat hasil output yang telah dibangun.

2.3.1 Analisis Kebutuhan

Dalam metode agile, Pada **Tabel 5 Pertanyaan Interview** salah satu tahapan penting adalah menganalisis kebutuhan pengguna dalam merancang konsep sistem deteksi kantuk ini. Metode yang diterapkan untuk menganalisis kebutuhan tersebut adalah dengan interview beberapa responden berjumlah 10 orang yang memiliki pola sama yaitu orang orang yang memiliki mobilitias tinggi dengan menggunakan mobil serta pengalaman *microsleep* saat berkendara. Adapun penyusunan pertanyaan interview disajikan dalam bentuk tabel yaitu sebagai berikut

Tabel 5 Pertanyaan Interview

No	Pertanyaan	Tujuan
1	Seberapa sering Anda mengalami rasa kantuk atau microsleep saat berkendara	Mengetahui tingkat kejadian kantuk atau microsleep pada pengendara.
2	Dalam kondisi seperti apa Anda biasanya mulai merasa kantuk di perjalanan?	Mengidentifikasi faktor pemicu kantuk, seperti waktu, kondisi fisik, atau lingkungan.
3	Apakah Anda sudah pernah menggunakan perangkat wearable saat berkendara?	Mengetahui tingkat adopsi pengguna terhadap perangkat Apple sebagai alat bantu.
4	Data vital apa yang menurut Anda paling relevan untuk mendeteksi kantuk ?	Menggali persepsi pengguna tentang indikator fisiologis penting untuk deteksi kantuk.
5	Bagaimana menurut Anda bentuk peringatan yang paling efektif untuk mengingatkan pengendara saat kantuk terdeteksi?	Menentukan preferensi pengguna terhadap mekanisme alert seperti getaran, suara, atau notifikasi.
	enting menurut Anda eksi kantuk dalam keselamatan	Mengukur persepsi manfaat sistem terhadap keselamatan pengguna.
	lebih suka sistem ini atis di latar belakang?	Mengetahui preferensi pengguna terhadap kemudahan dan kontrol penggunaan aplikasi.

No	Pertanyaan	Tujuan
8	Bagaimana pengalaman Anda menggunakan aplikasi berbasis kesehatan sebelumnya (jika ada)?	Memahami pengalaman dan ekspektasi pengguna terhadap aplikasi serupa.
9	Apa saran Anda agar sistem deteksi kantuk ini lebih efektif dan mudah digunakan?	Mendapatkan masukan langsung untuk perbaikan rancangan sistem.

2.3.2 Pengembangan Model

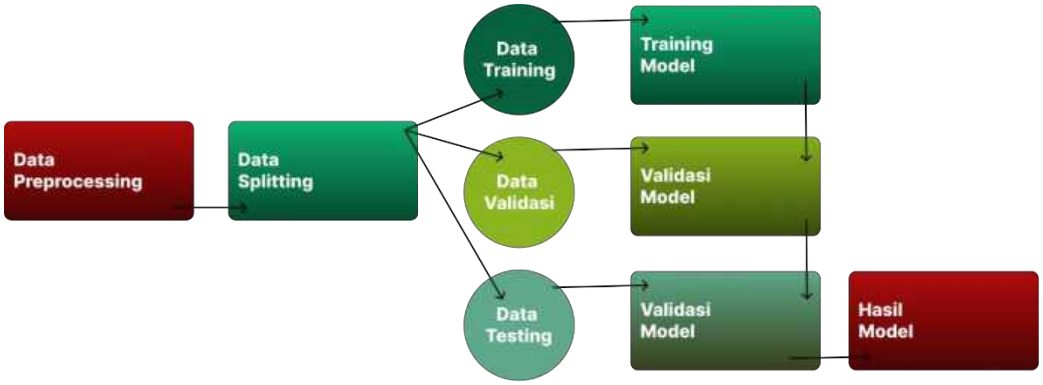
2.3.2.1 Alur Pengembangan Model

Machine learning digunakan dalam inputan kamera untuk memastikan kondisi perubahan pada wajah pengguna dan berfokus pada mata karena sistem ini secara objektif ingin monitoring level kantuk pengguna hingga ke level berbahaya yaitu *microsleep* dimana kondisi tersebut sangat membahayakan pengendara di jalan raya. *Microsleep* ini kadang tidak disadari pengendara tetapi ini merupakan kondisi yang sangat krusial dan membahayakan orang lain. Penulis melakukan beberapa iterasi dalam pelatihan model karena ingin memastikan pendekatan terbaik yang dapat dilihat dari beberapa metrik kesuksesan dari model.

Iterasi pertama adalah penulis menggunakan *dataset* publik dan selanjutnya dilakukan implementasi menggunakan *image classifier* dengan *createML*. Lebih lanjut, *model dataset* ini juga menggunakan dua label yaitu terbuka dan tertutup untuk memastikan kondisi mata pengguna. Pada iterasi pertama ini mengunakan *Image Feature Print V1* dengan menggunakan iterasi sebanyak 80 kali secara standar dan menerapkan beberapa augmentasi seperti menambahkan *noise* dan *expose* serta yang lainnya agar mendapatkan data yang lebih bervariasi.

Pada **Gambar 7 Alur Training Model** Iterasi kedua dilakukan untuk membandingkan *model* yang lebih baik untuk diterapkan. Iterasi kedua ini menggunakan metode yang cukup berbeda dengan iterasi kedua yaitu menggunakan *model vision* yang ada dari *apple* sehingga bisa digunakan secara langsung dalam sistem. *model vision* ini berfokus pada *face landmark* yang memastikan koordinat titik mata dan mengetahui perubahan gerakan pada koordinat tersebut. Pada saat implementasi iterasi kedua ini, penulis menambahkan fitur mendeteksi wajah pada kamera, dan ini merupakan pemantik *model* deteksi kantuk terjadi jika terdeteksi wajah. Jika wajah tidak terdeteksi maka model deteksi kantuknya juga tidak akan bekerja. Penulis juga melakukan beberapa iterasi lanjutan untuk melihat perbedaan yang signifikan.

Tahap penyusunan *model* ini diawali dengan data preprocessing dimana menambahkan *noise* dan *expose* dari data nya sehingga mendapat ragam data yang lebih banyak saat melatih model, setelah itu diikuti pembagian data menjadi tiga yaitu data training, data testing dan data validation dan melihat konsistensi metrik yang ada. metrik yang diperhatikan untuk model ini adalah dari sisi akurasinya karena memastikan pengguna bisa terdeteksi dengan baik saat berkendara secara langsung.



Gambar 7 Alur Training Model

2.3.2.2 Spesifikasi *Model*

Model ini dikembangkan dan diuji menggunakan *macOS* dengan bantuan *CreateML* dan *Core ML*, serta diimplementasikan ke dalam aplikasi berbasis *Swift*. Proses *deteksi* wajah dan pemotongan area mata dilakukan menggunakan *Vision Framework*, sedangkan prediksi kondisi mata dilakukan secara langsung oleh *model machine learning* yang telah dilatih sebelumnya. Dengan kombinasi antara data visual dan fisiologis, *model* ini mampu menghasilkan sistem deteksi kantuk yang efisien, ringan, dan dapat diimplementasikan pada perangkat *Apple* secara *real-time*. Spesifikasi *model* bertujuan untuk menjelaskan konfigurasi teknis dan komponen pendukung dari sistem deteksi kantuk yang dikembangkan. Model utama pada penelitian ini dibangun menggunakan *framework CreateML* dari *Apple*, dengan jenis model *Image Classifier* yang mengimplementasikan arsitektur *Image Feature Print V1*. Arsitektur ini merupakan bentuk *Convolutional Neural Network (CNN)* yang telah dioptimalkan untuk perangkat Apple, sehingga mampu melakukan proses inferensi secara cepat dan efisien dengan ukuran model yang relatif kecil. Model ini menerima input berupa citra RGB berukuran 360×360 piksel, yang telah melalui proses deteksi wajah dan pemotongan area mata menggunakan *Vision Framework*. Setiap citra diklasifikasikan ke dalam dua kelas, yaitu *open* (mata terbuka) dan *closed* (mata tertutup). Proses pelatihan dilakukan menggunakan *Create ML Image Classifier* dengan pembagian *dataset* sebesar 80% untuk data pelatihan dan 20% untuk data pengujian.

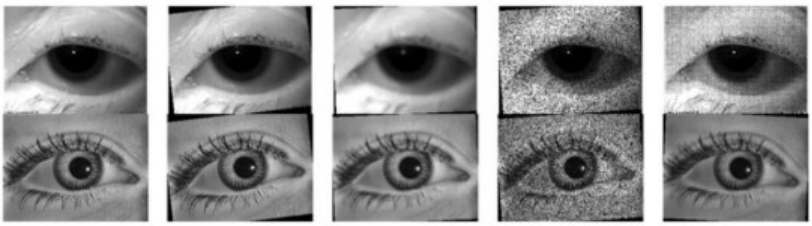
Tabel 6 Spesifikasi Model

Komponen	Spesifikasi
Framework	CreateML
Arsitektur	Image Feature Print
Jenis Input	Citra RGB
	360 * 360 piksel
	2 (Open, Closed)
	Label (Open/Closed)

2.3.2.3 Preprocessing dan Augmentasi Data

Tahap *preprocessing dan augmentasi data* dilakukan untuk meningkatkan kualitas serta variasi data citra yang digunakan dalam proses pelatihan model. Tujuan utama dari tahap ini adalah agar model dapat mengenali pola secara lebih baik, tidak mudah overfitting, serta memiliki kemampuan generalisasi yang baik terhadap kondisi nyata seperti pencahayaan berbeda, *noise* kamera, maupun variasi posisi wajah pengguna. Selanjutnya, dilakukan proses *augmentasi* data untuk memperbanyak variasi citra tanpa menambah jumlah data asli. *Augmentasi* yang digunakan pada penelitian ini meliputi dua teknik utama, yaitu penambahan *noise (add noise) dan penyesuaian pencahayaan (expose adjustment).*

1. *Add Noise* dilakukan dengan menambahkan derau acak pada citra untuk mensimulasikan kondisi nyata seperti gangguan sensor kamera, pencahayaan rendah, atau getaran saat pengambilan gambar.
2. *Expose Adjustment* dilakukan dengan mengubah tingkat kecerahan (exposure) citra secara acak untuk meniru variasi kondisi pencahayaan, misalnya siang hari, malam hari, atau pencahayaan dalam kabin kendaraan.



Gambar 8 Contoh Implementasi Augmentasi

Pada **Gambar 8 Contoh Implementasi Augmentasi** Kedua teknik augmentasi ini secara signifikan membantu meningkatkan *robustness* model terhadap variasi visual di dunia nyata. Setelah proses *augmentasi*, seluruh citra disimpan kembali dalam format 360×360 piksel dan digunakan sebagai dataset pelatihan (*training set*) dan pengujian (*testing set*) pada *Create ML Image Classifier*.

2.3.2.4 Data Splitting

Pada **Tabel 7 Penyebaran Data Pelatihan** Tahap *data splitting* merupakan proses pembagian *dataset* menjadi beberapa *subset* dengan tujuan untuk melatih, memvalidasi, dan menguji performa *model machine learning* secara objektif. Pembagian data dilakukan agar model tidak hanya mampu mengenali pola dari data pelatihan, tetapi juga dapat menggeneralisasi dengan baik terhadap data baru yang belum pernah dilihat sebelumnya. Pada penelitian ini, data citra dibagi menjadi tiga subset dengan proporsi 80% untuk data pelatihan (*training set*), 10% untuk data validasi (*validation set*), dan 10% untuk data pengujian (*testing set*). Proporsi 80:10:10 dipilih karena memberikan a ketersediaan data untuk pelatihan dan cukupnya data untuk an. Dengan pembagian ini, *model* dapat dilatih dengan data yang tetap menjaga keakuratan evaluasi performa secara independen.

Tabel 7 Penyebaran Data Pelatihan

No	Subset Data	Persentase	Jumlah Data
1	*Training Set*	80%	4.354 Gambar
2	*Validation Set*	10%	252 Gambar
3	*Testing Set*	10%	240 Gambar

2.3.2.5 Pelatihan Model

Tahap pelatihan model merupakan proses utama dalam pengembangan sistem deteksi kantuk ini, di mana model *machine learning* dilatih untuk mengenali pola visual antara kondisi mata terbuka (*open*) dan mata tertutup (*closed*). Proses pelatihan dilakukan menggunakan *framework Create ML* dari *Apple* dengan jenis model *Image Classifier*, yang secara internal menerapkan arsitektur *Convolutional Neural Network (CNN)*. Proses pelatihan berlangsung secara iteratif melalui beberapa *epoch*, di mana setiap *epoch* merepresentasikan satu siklus penuh pembelajaran terhadap seluruh data pelatihan. Selama pelatihan, model secara bertahap memperbarui *bobot* (*weights*) untuk meminimalkan *loss function* dan meningkatkan nilai akurasi. Data validasi digunakan pada setiap iterasi untuk memantau performa model terhadap data yang tidak ikut dilatih, sehingga dapat mencegah terjadinya *overfitting*.

Selain itu, *Create ML* juga menerapkan proses *automatic hyperparameter tuning*, yaitu pemilihan parameter optimal seperti *learning rate* dan *batch size* untuk mencapai hasil terbaik. Hasil pelatihan menunjukkan bahwa model berhasil mencapai akurasi sebesar 94% pada data pengujian, dengan *loss value* yang stabil setelah beberapa epoch. Ukuran *model* akhir yang dihasilkan adalah 7 KB, menjadikannya efisien dan mudah diimplementasikan pada perangkat berbasis iOS atau macOS. *Model* yang telah dilatih kemudian diekspor dalam format .mlmodel, yang dapat langsung diintegrasikan ke dalam aplikasi menggunakan Core ML. Dengan integrasi tersebut, sistem dapat melakukan prediksi kondisi mata secara *real-time* melalui kamera perangkat, sekaligus menjadi komponen utama dalam proses deteksi kantuk berbasis visual.

2.3.2.6 Evaluasi Model

Tahap evaluasi model dilakukan untuk menilai kinerja model machine learning dalam mengenali kondisi mata terbuka dan tertutup secara akurat. Evaluasi ini bertujuan untuk memastikan bahwa model tidak hanya memiliki performa tinggi pada data pelatihan, tetapi juga mampu melakukan generalisasi dengan baik terhadap data baru.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Dimana :

itive) : Data positif yang diprediksi benar
sitive) : Data negatif namun diprediksi sebagai data positif

naan 1. Berdasarkan hasil pengujian menggunakan dataset uji a pelatihan, *model* menunjukkan tingkat akurasi sebesar 94%, yang puan klasifikasi yang sangat baik. Nilai *loss* yang stabil pada setiap

epoch memperlihatkan bahwa proses pembelajaran berlangsung optimal tanpa indikasi overfitting yang signifikan.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Dimana :

- TP (*True Positive) :* Data positif yang diprediksi benar
- FN (*False Negative*) : Data positif namun diprediksi sebagai data negatif

Pada **Persamaan 2.** Selain metrik akurasi, model juga dievaluasi menggunakan parameter *precision, recall, dan F1-score* untuk memastikan keseimbangan antara deteksi benar terhadap kondisi mata terbuka dan tertutup. Hasil evaluasi menunjukkan nilai *precision* sebesar 0.93, recall sebesar 0.94, dan *F1-score* sebesar 0.93, menandakan performa *model* yang konsisten dan dapat diandalkan dalam mendeteksi perubahan kondisi mata pengguna secara *real-time*.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

Dimana :

- *Recall* : Mengukur seberapa banyak kasus positif dari semua data yang aktualnya positif
- *Precision* : Mengukur semua prediksi pada label tertentu dan melihat persentase prediksi benar

Pada **Persamaan 3.** Visualisasi hasil evaluasi berupa confusion matrix memperlihatkan bahwa tingkat kesalahan klasifikasi sangat rendah, dengan sebagian besar prediksi berada pada kategori yang benar. Hal ini menunjukkan bahwa model mampu mempelajari perbedaan fitur visual antara mata terbuka dan tertutup secara efektif.

$$Accuracy = \frac{TP+TN}{TP + FP + FN+TN} \quad (4)$$

Dimana :

- TP (*True Positive) :* Data positif yang diprediksi benar
- FP (*False Positive*) : Data negatif namun diprediksi sebagai data positif
- FN (*False Negative*) : Data positif namun diprediksi sebagai data negatif
- TN (*True Negative*) : Data negative yang diprediksi benar

ıaan 4. Dengan ukuran akhir model sebesar 7 KB, model terbukti ‹ı waktu inferensi yang cepat ketika diimplementasikan dalam OS. Secara keseluruhan, hasil evaluasi ini menunjukkan bahwa memiliki performa tinggi, efisien, dan siap digunakan dalam sistem sis *machine learning* dan integrasi kamera *real-time*.

2.4 Instrumen Penelitian

Pada **Tabel 8 Instrumen Penelitian Perangkat Lunak** Penelitian ini memiliki beberapa instrumen yang akan mendukung penulis untuk mengimplementasikan rancang bangun yang direncanakan sebelumnya. Instrumen tersebut akan berkaitan dengan perangkat keras dan perangkat lunak yang saling berkorelasi satu sama lain dimana diharapkan dengan instrumen tersebut akan meningkatkan standar minimun dari produk atau rancang bangun yang dibentuk.

Tabel 8 Instrumen Penelitian Perangkat Lunak

No	Perangkat Lunak	Kegunaan
1	*Xcode* 16.0 (Build 16A2409), rilis 2024, *macOS Sequoia* 15.0	Digunakan untuk merancang sistem dan integrasi machine learning serta dalam pengelolaan kode
2	*Figma* *Desktop App* v116.14.3 (64-bit)	Digunakan dalam merancang antarmuka sistem dalam bentuk *low fidelity* hingga *high fidelity*
3	*HealthKit* SDK *iOS* 18.0 (Apple Health Framework),	Digunakan sebagai parameter tambahan untuk deteksi kantuk
4	*CreateML* *Version 1.5, macOS Native App, Update*	Merancang machine learning model dari sisi wajah pengendara hingga mendeteksi kantuk
5	*Yolo* *YOLOv8.2.43 (Ultralytics), Python 3.11*	Digunakan untuk Mendeteksi adanya orang
6	*Swift* *Swift 5.10.1, Compiler di Xcode 16, Update 2024-05*	*Framework* yang diimpelemtasi dalam pengelolan kode
7	*Github* *Version 3.3.8 (64-bit), Update 2024-08-10*	Pengelolaan dan dokumentasi kode secara sistematis
8	*Draw.io*	Menggambarkan visualisasi diagram dan interaksi pengguna dengan sistem
9	*Vision* *Apple Vision Framework, iOS 18.0 SDK*	Integrasi machine learning dengan sistem rancang bangunn

Instrumen Penelitian Perangkat Keras Selain perangkat lunak, merlukan perangkat keras yang cukup khusus karena sistem akan g lingkup apple sehingga memerlukan perangkat yang mendukung

Tabel 9 Instrumen Penelitian Perangkat Keras

No	Perangkat Keras
1	*MacBook M4 Pro* *Apple M4 Pro Chip 12-core CPU, 18-core GPU, 16-core Neural Engine, RAM 16 GB LPDDR5, SSD 512 GB, Layar Liquid Retina XDR 3024×1964, ProMotion 120Hz, macOS Sequoia 15.0*
2	*Iphone 15* *Chip A17 Pro 6-core CPU, 6-core GPU, Neural Engine 16-core, RAM 8 GB, Storage 128 GB, Layar Super Retina XDR OLED 6,1", iOS 18.0*
3	*Apple Watch 10* *Chip S10 SiP, RAM 2 GB, Storage 32 GB, Layar LTPO OLED Always-On, watchOS 11.0*

2.5 Tahapan Penelitian

Metode *agile* adalah metode yang akan menjadi fokus penulis dalam rancang bangun sistem deteksi kantuk ini yang disebabkan oleh beberapa pertimbangan yang telah dibahas sebelumnya. Adapun tahapan penelitiannya yaitu:

1. *Requirements*
   Pada tahap ini akan berfokus pada analisis kebutuhan dari sistem yang akan dibangun, dimana metode dalam analisis kebutuhannya dapat berupa observasi dan *interview* serta menerapkan *framework AIUEO* yang membantu penulis menerjemahkan kejadian tertentu saat observasi. Tahap ini diharapkan mendapat informasi penting berupa data yang akan dimanfaatkan di tahap pengembangan sistem.
2. *Design*
   Tahap *design* adalah tahap awal setelah pengumpulan data yang akan berfokus pada merancang desain interaksi pengguna terhadap aplikasi yang akan membantu dalam merancang antarmuka sistem yang sesuai sehingga tidak terjadi kebingungan pengguna saat menggunakan sistem yang dibangun. Hal ini juga bertujuan menentukan design system yang sesuai dengan *user persona* yang ditargetkan agar tidak salah sasaran dalam penggunaannya.
3. *Development*
   Tahap ini akan cukup krusial karena akan menciptakan *prototype* awal sistem yang dapat digunakan karena fungsionalnya diharapkan telah dapat berfungsi dengan baik. Tahapan ini akan mengimplementasikan hasil desain antarmuka yang telah dibuat sebelumnya lalu akan berfokus pada *hierarchy design* agar pengguna tidak kebingungan. Tahap Development ini akan berfokus pada pengolahan data yang berfokus pada membangun model *machine learning* dalam bentuk dua *layer* gkatkan pengalaman pengguna dalam menggunakan aplikasi. udah *training* dengan baik, selanjutnya adalah tahapan integrasi *ntend* serta model kedalam sebuah *prototype* awal tersebut.

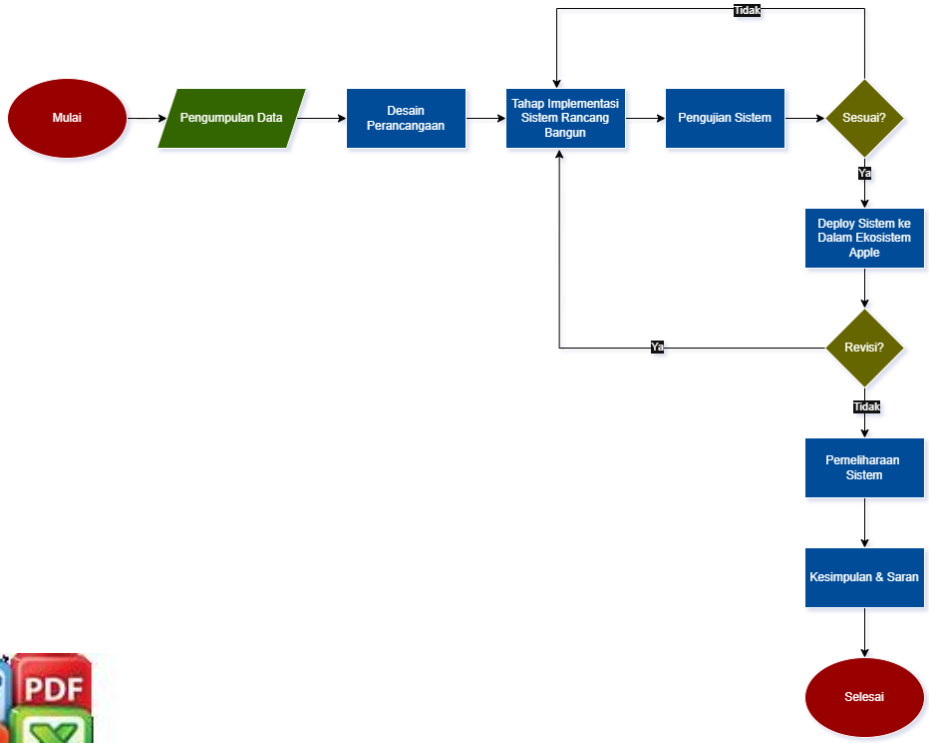
prototyping awal telah dibuat secara fungsionalitas, selanjutnya engundang beberapa narasumber untuk melakukan pengetesan yang dibangun dan akan mendapatkan beberapa *feedback* untuk a atau tahap pengembangan selanjutnya. Bentuk *testing* yang

diterapkan berupa *user acceptance testing* yang akan berfokus pada pengalaman pengguna serta melihat keinginan pengguna kedepannya terkait sistem yang ditawarkan.

5. *Deploy*
   Proses *deploy* akan memanfaatkan aplikasi *ekosistem apple* yang akan dibantu oleh tim *apple* untuk prosesnya dan akan diakurasi lebih lanjut oleh *tester* dari tim *apple* sehingga aplikasi ini bisa digunakan secara publik di *app store*. Dalam tahap ini, penulis akan dihadapkan oleh beberapa *rejected* oleh *tester* tetapi ini menjadi salah satu alasan juga penerapan *agile* karena penulis akan langsung adaptif untuk melakukan perubahan dan mengikuti standar yang ditetapkan tim *apple* untuk aplikasi atau sistem yang ingin dipublikasi di ekosistem *apple*.
6. *Review*
   Setelah berbagai macam tahapan yang dilalui, selanjutnya adalah pemeliharaan aplikasi jangka panjang dengan melihat review singkat yang mungkin dilontarkan oleh pengguna di *platfrom* daring tersebut yaitu *app store* sehingga bisa melakukan pengembangan lebih lanjut terkait dari sistemnya.

2.6 Alur Penelitian

Pada **Gambar 9. Alur Penelitian** Berdasarkan tahapan *agile methodology* sebelumnya, terbentuklah alur penelitian yang dirancang penulis dan akan dilalui selama tahapan penelitian. Alur penelitian ini dapat berubah sesuai dengan kebutuhan penulis berkaitan dari segi efektifitas dan dari segi efisiensi.

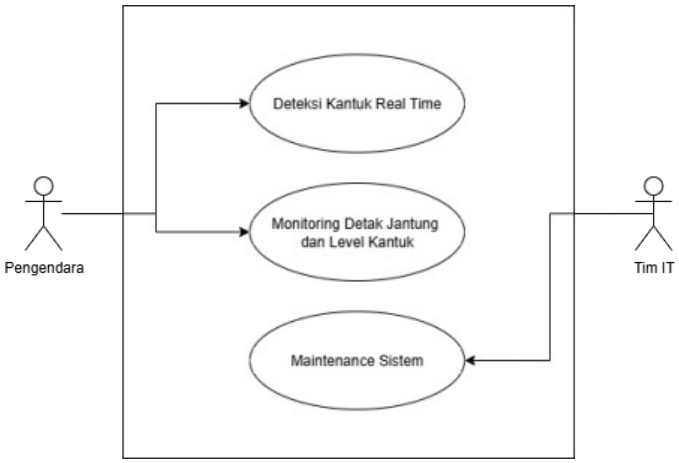


Gambar 9 Alur Penelitian

2.7 Diagram Sistem

2.7.1 Use Case Diagram

Diagram ini merupakan salah satu bagian dari *Unified Modelling Language* dimana *use case diagram* ini merepresentasikan aktor yang berperan dalam fungsionalitas aplikasi sehingga memberikan bayangan yang cukup jelas terkait batasan fitur yang disajikan dam dapat disesuaikan dengan waktu dan tenaga yang akan digunakan oleh penulis. Diagram ini dapat berupa interaksi *user* dan ini dapat membantu stakeholder maupun penulis dalam memaksimalkan sistem dan cara sistem bekerja (Seidl et al., 2015).

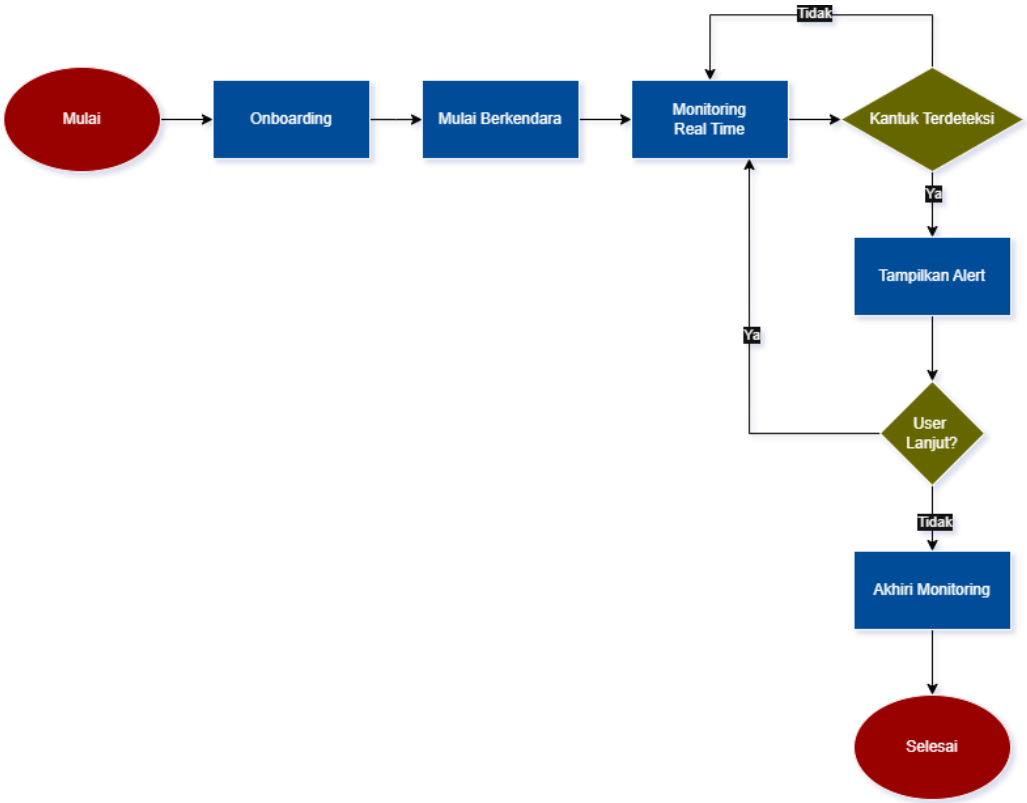


Gambar 10 Use Case Diagram

Pada **Gambar 10 Use Case Diagram** Disajikan bentuk *use case* sistem rancang bangun dimana terdapat dua aktor yaitu pengendara sebagai aktor utama dan Tim IT sebagai aktor pendukung. Tetapi disatu sisi, sistem rancang bangun tidak memerlukan login untuk memastikan *role* dari user karena untuk tim IT akan fokus kedalam source code yang digunakan untuk membangun sistem aplikasi ini.

1. Pengendara
   Aktor utama ini mendapat semua fungsionalitas yang ditawarkan kecuali maintenance sistem dimana pengendara dapat menggunakan fitur utama yaitu deteksi kantuk *real time* selama user tersebut berkendara sehingga dapat mendapat *alert* ataupun *tracking* kantuk nya selama berkendara sehingga membantu pengendara untuk mengecek Kesehatan atau rasa kantuk user persona.
2. Tim IT
   Aktor pendukung ini merupakan salah satu tim yang membantu meningkatkan pengalaman pengguna nantinya dalam maintenance sistem. Tetapi disatu sisi tim IT hanya bisa mengakses data data dari user dan menganalisis user *behaviour* untuk pengembangan lanjutan

2.7.2 Activity Diagram



Gambar 11 Activity Diagram

Pada **Gambar 11. Activity Diagram** Proses dimulai dengan tahap *onboarding*, di mana pengguna melakukan inisialisasi aplikasi atau sistem sebelum berkendara. Pada tahap ini, sistem menyiapkan kamera atau sensor yang akan digunakan untuk memantau wajah dan perilaku pengemudi. Setelah proses onboarding selesai, pengguna dapat memilih untuk memulai berkendara. Sistem kemudian masuk ke tahap monitoring real-time, di mana data dari kamera dianalisis secara kontinu untuk mendeteksi tanda-tanda kantuk, seperti penutupan mata, menundukkan kepala, atau hilangnya fokus pandangan.

Ketika sistem mendeteksi adanya indikasi kantuk, proses berpindah ke tahap tampilkan alert. Pada tahap ini, sistem memberikan peringatan kepada pengemudi melalui notifikasi suara, visual, atau getaran, dengan tujuan agar pengemudi segera menyadari kondisi mengantuknya. Setelah peringatan muncul, sistem menunggu respon dari pengguna dengan memeriksa apakah pengguna akan melanjutkan berkendara atau tidak. Jika pengguna memilih untuk beristirahat, sistem tetap berada dalam kondisi siaga dan menunggu hingga pengguna siap untuk melanjutkan perjalanan.

jguna memutuskan untuk melanjutkan berkendara, sistem akan nonitoring real-time untuk terus memantau kondisi pengemudi. a memilih untuk mengakhiri perjalanan, maka sistem akan masuk toring, di mana semua proses pemantauan dihentikan dan data npan untuk keperluan evaluasi atau analisis lebih lanjut.

2.8 Rancangan Antarmuka

Rancangan antarmuka adalah tahap awal dalam pengembangan sistem rancang bangun dimana terdapat beberapa antarmuka yang akan buat sebagai acuan awal dalam tahap *development* kedepannya. Walaupun demikian, perubahan antarmuka dalam tahap implementasi masih memungkinkan dengan pertimbangan tertentu sehingga rancangan antarmuka awal ini hanya menjadi acuan awal agar tata letak komponennya dapat terlihat dengan jelas.

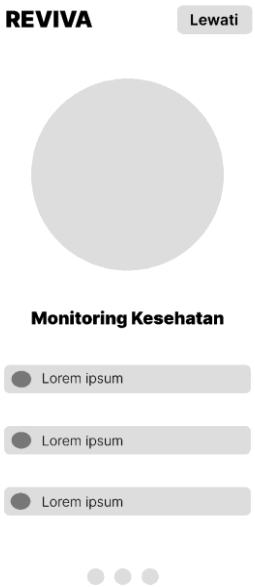
2.8.1 Onboarding – Deteksi Kantuk

Halaman ini dibuat untuk memperkenalkan fitur-fitur utama kepada pengguna baru sebelum mereka benar-benar mulai memakai aplikasi. Secara keseluruhan, tampilannya cukup simpel dan langsung pada intinya, jadi pengguna bisa langsung paham apa yang mau disampaikan. Pada **Gambar 12 Onboarding – Deteksi Kantuk** Di bagian atas, ada logo Reviva di pojok kiri yang jadi identitas dari aplikasi ini, dan di sebelah kanannya ada tombol Lewati yang bisa digunakan kalau pengguna ingin skip pengenalan dan langsung masuk ke aplikasi utama. Bagian tengahnya sendiri didominasi oleh lingkaran besar berwarna abu-abu muda. Lingkaran ini sepertinya dirancang sebagai tempat untuk menampilkan ilustrasi atau mungkin animasi singkat yang menjelaskan tentang fitur Deteksi Kantuk Real Time. Karena ukurannya yang cukup besar dan letaknya pas di tengah, otomatis perhatian pengguna akan tertuju ke sini dulu untuk memahami fitur utama yang ditawarkan. Sementara itu, di bagian bawah halaman terdapat tiga buah bar horizontal yang masing-masing dilengkapi dengan ikon berbentuk lingkaran kecil dan teks penjelasan. Bar-bar ini kemungkinan besar digunakan untuk menjelaskan beberapa poin penting terkait fitur atau keunggulan utama dari aplikasi Reviva secara lebih detail.

Gambar 12 Onboarding – Deteksi Kantuk

2.8.2 Onboarding – Monitoring Kesehatan

Dari segi tampilan, desainnya dibuat konsisten dengan halaman sebelumnya supaya terlihat seragam dan gampang dikenali pengguna. Di bagian paling atas, masih ada logo Reviva yang terletak di sebelah kiri sebagai penanda aplikasi, dan tombol Lewati di sebelah kanan yang bisa dipencet kalau pengguna mau langsung masuk ke halaman utama tanpa harus melalui pengenalan fitur ini. Untuk bagian tengahnya, sama seperti halaman pertama, ada lingkaran besar dengan warna abu-abu muda yang jadi fokus utama. Lingkaran ini sepertinya bakal menampilkan ilustrasi, animasi, atau ikon yang berhubungan dengan konsep monitoring kesehatan. Penempatannya yang di tengah layar memang sengaja dibuat seperti itu biar pengguna langsung fokus ke sana dan paham gambaran dari fitur yang lagi dijelaskan. Tepat di bawah lingkaran tersebut, ada tulisan Monitoring Kesehatan yang memperjelas bahwa fitur ini memang dirancang untuk memantau kondisi kesehatan pengguna secara real-time saat sedang berkendara, misalnya seperti detak jantung, tingkat stres, atau tingkat fokus mereka. Lalu di bagian bawahnya lagi, ada tiga bar horizontal yang masing-masing punya ikon bulat dan teks penjelasan. Bagian ini kemungkinan dipakai untuk menjabarkan fitur-fitur pendukung atau manfaat-manfaat yang bisa didapat dari sistem Monitoring Kesehatan ini seperti pada **Gambar 13 Onboarding Monitoring Kesehatan**

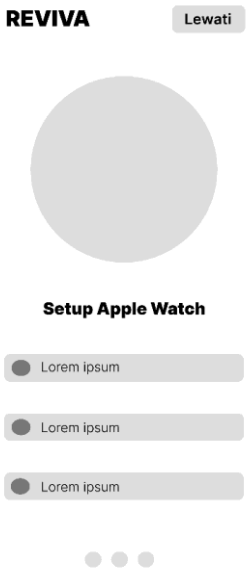


Gambar 13 Onboarding Monitoring Kesehatan

2.8.3 Onboarding – Setup Apple Watch

boarding – Setup Apple Watch menampilkan halaman ketiga atau engenalan aplikasi Reviva, yang kali ini membahas tentang fitur Tampilannya masih konsisten dengan dua halaman sebelumnya ana, dan tertata rapi supaya pengguna nyaman dari awal pakai s, logo Reviva tetap ada untuk menunjukkan identitas aplikasi dan ang profesional. Di sebelah kanannya juga masih tersedia tombol rikan kebebasan bagi pengguna yang ingin langsung masuk ke

aplikasi tanpa harus menyelesaikan semua tahap pengenalan ini. Untuk bagian tengah, sama seperti dua halaman sebelumnya, ada lingkaran besar berwarna abu-abu muda yang kemungkinan besar akan diisi dengan ilustrasi atau animasi yang menjelaskan cara menghubungkan aplikasi dengan Apple Watch. Elemen visual ini memang sengaja ditekankan karena fitur di halaman ini fokus pada integrasi dengan perangkat wearable untuk mendukung sistem monitoring kesehatan dan deteksi kantuk. Di bagian bawahnya, ada tiga bar horizontal yang dilengkapi ikon lingkaran dan teks penjelasan. Bar-bar ini berfungsi untuk menjabarkan langkah-langkah atau keuntungan dari fitur setup Apple Watch ini. Yang menarik, di bagian paling bawah ada indikator berupa tiga titik yang menandakan bahwa ini adalah halaman terakhir dari rangkaian onboarding yang bisa digeser atau di-*scroll*.



Gambar 14 Onboarding – Setup Apple Watch

2.8.4 Homepage

Desain antarmuka ini dibuat dengan tujuan untuk memberikan tampilan yang sederhana, intuitif, dan mudah dipahami oleh pengguna. Pada bagian atas, terdapat dua komponen utama yaitu *PERCLOS* dan *heart rate*, yang masing-masing merepresentasikan indikator utama dalam proses deteksi kantuk. *PERCLOS* (Percentage of Eye Closure) berfungsi untuk menampilkan tingkat persentase keterbukaan mata pengguna yang diukur melalui kamera,

Pada **Gambar 15 Homepage** *heart rate* menampilkan data detak jantung pengguna yang diambil dari sensor eksternal atau perangkat *wearable*. Di bawah kedua indikator tersebut terdapat elemen *risk level*, yang menampilkan hasil analisis tingkat ıa berdasarkan kombinasi data dari *PERCLOS* dan detak jantung.

Bagian ini menjadi output utama yang membantu pengguna memahami kondisi kewaspadaan mereka secara cepat dan jelas. Pada bagian bawah antarmuka, terdapat dua tombol utama, yaitu *Start* dan *Reset*. Tombol Start digunakan untuk memulai proses pengambilan data dan pemantauan secara realtime, sedangkan tombol Reset berfungsi untuk mengatur ulang tampilan atau menghapus data hasil pengukuran sebelumnya.

Gambar 15 Homepage

2.8.5 Halaman Deteksi Kantuk

Desain ini menampilkan berbagai komponen utama yang saling terintegrasi untuk memberikan informasi secara langsung kepada pengguna. Pada bagian atas, terdapat dua indikator utama, yaitu *PERCLOS* dan *heart rate*. Komponen *PERCLOS* berfungsi untuk menampilkan nilai persentase keterbukaan mata pengendara yang digunakan dalam mendeteksi tanda-tanda kantuk berdasarkan pergerakan mata. Sementara itu, komponen *heart rate* menampilkan detak jantung pengendara yang diperoleh melalui sensor untuk membantu sistem menilai kondisi fisiologis pengguna secara lebih akurat. Di bawah kedua indikator tersebut terdapat bagian *risk level*, yang berfungsi sebagai hasil analisis dari sistem berdasarkan data yang diperoleh dari kedua indikator sebelumnya. Bagian ini menampilkan tingkat risiko kantuk pengendara, seperti rendah, sedang, atau tinggi, yang dihasilkan dari perhitungan dan *model* deteksi yang diterapkan. Selanjutnya, terdapat elemen *Preview* Pengendara, yang menampilkan umpan balik visual secara langsung dari kamera. Fitur ini memungkinkan sistem dan pengguna untuk memantau kondisi wajah atau ekspresi secara *realtime*, sehingga proses deteksi dapat berlangsung secara transparan dan interaktif. Pada bagian paling bawah antarmuka, l utama, yaitu *Quit* dan *Reset*. Tombol *Quit* digunakan untuk ı proses deteksi dan keluar dari program, sedangkan tombol *Reset* gatur ulang data atau memulai kembali proses pemantauan dari **16 Halaman Deteksi Kantuk** Desain antarmuka ini menonjolkan

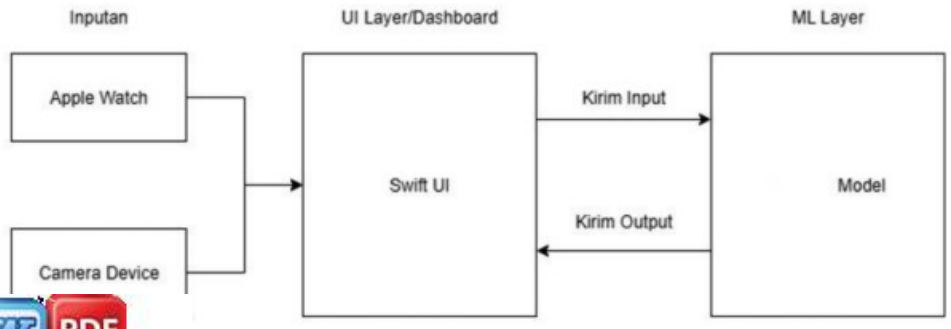
kesederhanaan dan kejelasan visual agar pengguna dapat dengan mudah memahami informasi yang ditampilkan tanpa terganggu oleh elemen visual yang berlebihan.

Gambar 16 Halaman Deteksi Kantuk

2.9 Arsitektur Sistem

Penulis ingin memastikan perancangan sistem yang akan dibangun dapat dilakukan dengan baik maka disajikan arsitektur dari sisi client maupun backend ataupun *model machine learningnya* agar sistem rancang bangun dapat dijadikan pedoman dan memudahkan dari segi integrasi antara antarmuka yang akan dibangun dengan *model machine learning* yang bekerja dengan target metrik tertentu sebagai metrik sukses dari rancang bangun tersebut.

2.9.1 Arsitektur Sistem Rancang Bangun

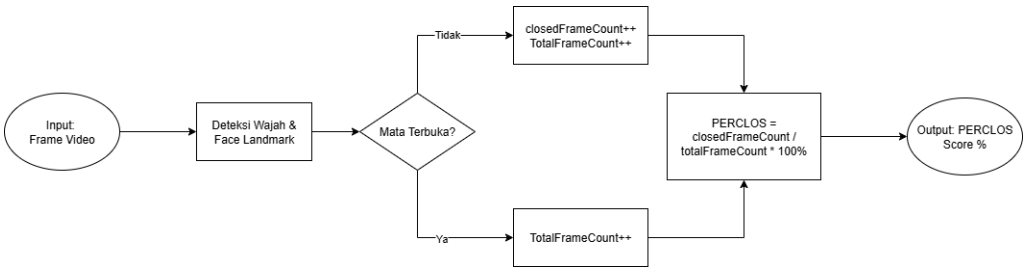


Gambar 17 Arsitektur Sistem Rancang Bangun

Pada **Gambar 17 Arsitektur Sistem Rancang Bangun** Memperlihatkan arsitektur aplikasi yang membagi inputan hingga mendapat output yang ditampilkan di antarmuka pengguna. Inputannya terdapat dua hal yaitu inputan fisiologis atau data vital dan kondisi mata dan menghitung durasi mata berkedip. Setelah mendapatkan kedua inputan tersebut, maka akan dikirimkan ke *UI layer* terlebih dahulu sebagai data monitoring agar pengguna juga dapat melihat kondisi tubuhnya. Selain itu data tersebut akan lanjut dikirimkan ke *ML layer* untuk diolah lebih lanjut agar mendapatkan output akhir dan dikirimkan kembali ke *UI Layer*

PERCLOS merupakan salah satu metrik yang paling *reliabel* untuk mendeteksi kantuk pada pengendara, yang mengukur persentase waktu mata tertutup dalam periode tertentu. Sistem ini mengimplementasikan perhitungan *PERCLOS* secara *real-time* menggunakan pendekatan berbasis *computer vision* dan *face landmark detection*.

2.9.2 Alur Mekanisme Sistem



Gambar 18 Alur Perhitungan PERCLOS Score

Pada **Gambar 18 Alur Perhitungan PERCLOS Score.** *PERCLOS* (Percentage of Eye Closure) merupakan metrik yang digunakan untuk mengukur persentase waktu di mana mata pengguna tertutup dalam periode pengamatan tertentu. Nilai ini dihitung berdasarkan jumlah *frame* video yang menunjukkan kondisi mata tertutup dibandingkan dengan total seluruh *frame* yang diamati. Secara matematis, rumus *PERCLOS* dapat dituliskan sebagai berikut:

$$PERCLOS = \left(\frac{closedFrameCount}{totalFrameCount}\right) \times 100\% \quad (5)$$

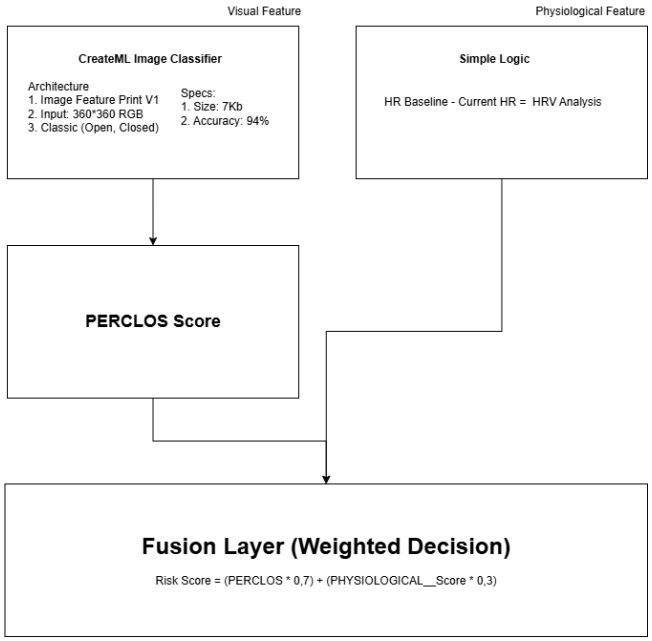
Dimana :

- *ClosedFrameCount* : jumlah *frame* video dimana mata terdeteksi dalam keadaan tertutup
- *totalFrameCount* : jumlah keseluruhan *frame* yang dianalisis

maan 5. Rumus tersebut menunjukkan bahwa semakin tinggi *ameCount*, maka semakin besar kemungkinan pengguna berada ntuk. Sebagai contoh, jika dari 200 *frame* video, terdapat 40 frame ata tertutup, maka nilai PERCLOS dihitung sebagai berikut:

$$\left(\frac{40}{200}\right) \times 100\% = 20\% \quad (6)$$

Pada **Persamaan. 6.** Artinya, selama interval pengamatan tersebut, mata pengguna tertutup selama 20% dari total waktu, yang dapat diinterpretasikan sebagai tanda awal munculnya kantuk. Nilai ambang batas tertentu kemudian digunakan dalam sistem untuk menentukan apakah kondisi pengguna sudah termasuk dalam kategori kantuk ringan, kantuk berat, atau masih normal.



Gambar 19 Mekanisme Perhitungan Sistem Deteksi Kantuk

Pada **Gambar 19 Mekanisme Perhitungan Sistem Deteksi Kantuk** Arsitektur model yang dikembangkan pada penelitian ini terdiri atas beberapa komponen utama yang saling terintegrasi, yaitu *Create ML Image Classifier*, *Simple Logic (HRV Analysis)*, *PERCLOS Score Calculation*, serta *Fusion Layer (Weighted Decision)* sebagai tahap akhir pengambilan keputusan.

Pada tahap pertama, sistem menggunakan *Create ML Image Classifier* yang dibangun menggunakan *framework Create ML* dari *Apple*. *Model* ini menggunakan arsitektur *Image Feature Print V1* dengan masukan citra berukuran 360×360 piksel dalam format RGB. Kelas yang digunakan adalah dua kategori utama, yaitu *open* (mata terbuka) dan *closed* (mata tertutup). *Model* yang dihasilkan memiliki ukuran sebesar 7 KB dengan tingkat akurasi mencapai 94%. Keluaran dari *model* ini digunakan untuk mendeteksi kondisi mata pengguna secara *real-time* sebagai dasar perhitungan tingkat kantuk.

sistem juga memanfaatkan analisis sinyal fisiologis melalui *Simple Rate Variability (HRV)*. Nilai *HRV* dihitung dari selisih antara detak *Baseline*) dan detak jantung saat ini (*Current HR*), yang si kelelahan atau menurunnya kewaspadaan pengguna. Hasil dari diolah lebih lanjut menjadi *PERCLOS Score* (*Percentage of Eye* gambarkan persentase waktu mata tertutup dalam periode tertentu.

PERCLOS dikenal sebagai salah satu indikator fisiologis paling kuat dalam mendeteksi kantuk karena berkorelasi langsung dengan penurunan tingkat kewaspadaan visual.

Proses perhitungan *Risk Score* dalam sistem deteksi kantuk ini mengombinasikan dua komponen utama, yaitu fitur visual dan fitur fisiologis, melalui pendekatan *fusion layer* berbobot. Pada sisi visual, model *CreateML Image Classifier* digunakan untuk mengenali kondisi mata pengguna (*open/closed*) dan menghasilkan nilai *PERCLOS Score*, yang merepresentasikan persentase waktu mata tertutup selama periode observasi. Sementara pada sisi fisiologis, sistem menggunakan analisis detak jantung (*Heart Rate Variability / HRV*) yang diperoleh dari perangkat *Apple Watch* melalui integrasi *HealthKit*. Nilai fisiologis dihitung dengan mengukur selisih antara detak jantung baseline pengguna. Selisih ini (ΔHR) berfungsi sebagai indikator perubahan tingkat kewaspadaan. Penurunan signifikan dari nilai *HRcurrent* terhadap *HRbaseline* menandakan adanya penurunan aktivitas sistem saraf simpatik, yang secara fisiologis berkorelasi dengan munculnya rasa kantuk.

Tahap akhir adalah *Fusion Layer (Weighted Decision),* yang berfungsi menggabungkan dua sumber informasi visual dan fisiologis menggunakan pendekatan *weighted decision*. Bobot $\propto$ diberikan pada komponen *PERCLOS* karena kontribusi visual dianggap lebih dominan dalam mendeteksi kantuk, sedangkan sinyal fisiologis memiliki bobot $1-\propto$ untuk memperkuat akurasi keputusan akhir. Hasil akhir berupa skor risiko kantuk yang dapat dijadikan indikator untuk memberikan peringatan dini kepada pengguna. *Risk score* ini memiliki beberapa rentang dan kategori yaitu dimulai dari kategori aman dari rentang 0 hingga 40, lalu dilanjutkan kategori waspada dari 40 hingga 60, apabila skor nya lebih dari 60 maka akan masuk ke kategori berbahaya.

Penurunan nilai HRV umumnya menunjukkan kondisi kelelahan atau menurunnya kewaspadaan seseorang, sehingga dapat dijadikan indikator pendukung dalam sistem ini. Kedua nilai tersebut kemudian digabungkan dalam lapisan Fusion Layer, dengan rumus penggabungan sebagai berikut:

$$Risk\ Score = (PERCLOS \times \propto) + (Physiological \times (1-\propto)) \quad (7)$$

Dimana :

- *PERCLOS* : indikator visual dari tingkat kantuk seseorang
- *Physiological* : indikator fisiologis seperti detak jantung seseorang
- *α (alpha)* : koefisien pembobotan antara 0 dan 1 yang menentukan seberapa besar kontribusi masing masing komponen terhadap total skor resiko

Pada **Persamaan 7.** Bobot sebesar α diberikan pada komponen visual karena indikator kondisi mata secara langsung mencerminkan tingkat kantuk pengguna, sedangkan bobot (1 − α) diberikan pada komponen fisiologis sebagai pelengkap untuk meningkatkan reliabilitas hasil deteksi. Komponen *PERCLOS* (*Percentage of Eye Closure*) merupakan parameter visual yang mengukur persentase waktu mata pengguna dalam kondisi tertutup selama interval waktu tertentu. Nilai *PERCLOS* yang tinggi ...pengguna sering menutup mata dalam durasi pendek, yang secara ...berkorelasi kuat dengan peningkatan tingkat kantuk. Parameter ini ...dari analisis citra wajah secara *real-time* melalui *model machine* ...atau perubahan kondisi mata.

...komponen psikologis mencakup indikator fisiologis seperti detak ...an *baseline-nya* yang diambil secara *real-time* dari perangkat *Apple* ...asi HealthKit. Penurunan detak jantung atau deviasi signifikan dari

baseline normal dapat menjadi indikator awal kelelahan atau kantuk pada pengguna. Nilai *Risk Score* yang dihasilkan dari kombinasi kedua komponen ini digunakan untuk menentukan status akhir sistem, apakah pengguna berada dalam kondisi normal, waspada, atau mengantuk. Melalui pendekatan *fusion logic* ini, sistem tidak hanya bergantung pada satu jenis data, tetapi mengombinasikan data visual dan data fisiologis yang saling melengkapi. Dengan demikian, tingkat akurasi dan ketahanan sistem terhadap variasi kondisi lingkungan maupun perbedaan individu dapat ditingkatkan, menghasilkan deteksi kantuk yang lebih stabil, akurat, dan kontekstual.

Penulis melakukan implementasi ke dalam aplikasi dan integrasi setelah *model* telah dibuat dan siap digunakan. *Model* tersebut diekspor dalam format *MLModel* agar dapat diimplementasikan ke dalam aplikasi berbasis iOS lalu selanjutnya dalam sistem tersebut dilakukan integrasi berupa memanggil *function* yang akan menggunakan *MLModel* yang tersedia lalu akan dilempar ke dalam *controller* sehingga bisa digunakan dan ditampilkan di antarmuka yang akan dilihat oleh pengguna dan bisa memaksimalkan model yang telah dibuat oleh penulis.

2.10 Skenario Pengujian

2.10.1 Skenario Pengujian Fungsional

Pada **Tabel 10 Skenario Pengujian Fungsional** Metode yang digunakan dalam pengujian fungsional ini adalah *black box testing* dimana pengujian ini berfokus pada fungsionalitas tanpa perlu mengetahui struktur kode yang telah dibangun dalam sistem deteksi kantuk ini. Fokus skenario ini adalah kesesuaian antara harapan output dan respon yang terjadi. Jadi ini menjadi salah satu metrik pengujian dari sistem rancang bangun ini.

Tabel 10 Skenario Pengujian Fungsional

No	Skenario Pengujian	Input Data	Hasil yang diharapkan
TC-01	Pengujian Alur Onboarding Awal	Pengguna membuka aplikasi untuk pertama kali	Aplikasi menampilkan layar sambutan dengan deskripsi fitur yang disediakan
TC-02	Pengujian Deteksi Kantuk (Positif)	Pengguna menekan tombol dan mulai merekam wajah. Pengguna menutup mata secara perlahan dan berulang	Aplikasi menampilkan peringatan visual dan peringatan alarm. Parameter *PERCLOS* meningkat secara signifikan dan Risk Level berubah dari aman menjadi waspada
	ı Deteksi Negatif)	Pengguna menekan tombol dan mulai merekam dengan kondisi mata terbuka	Aplikasi tidak menampilkan peringatan apa pun. Parameter *PERCLOS* tetap rendah

No	Skenario Pengujian	Input Data	Hasil yang diharapkan
TC-04	Pengujian Deteksi dengan *PERCLOS* Rendah	Pengguna menekan tombol *Start*, tetapi tingkat *PERCLOS*	Aplikasi tidak menampilkan peringatan kantuk. Risk Level tetap aman
TC-05	Pengujian Tampilan Data *Real-Time*	Aplikasi dalam mode *Start*	Parameter *PERCLOS* (%) dan *Heart Rate* (BPM) diperbarui secara real-time di layar utama. Angka *Risk Level* juga diperbarui sesuai kondisi.
TC-06	Pengujian Tombol *Quit* dan *Reset*	Pengguna menekan tombol *Quit* atau *Reset* saat deteksi sedang berjalan	Aplikasi kembali ke tampilan awal sebelum *Start* dengan semua parameter direset ke nilai default (misal, *PERCLOS* 0.0%, *Risk Level* aman).
TC-07	Pengujian Notifikasi Suara	Sistem mendeteksi kantuk dari wajah dan *PERCLOS* mencapai ambang batas.	Alarm suara seperti yang terdengar harus berbunyi.
TC-08	Pengujian Notifikasi Visual	Sistem mendeteksi kantuk	Sebuah label Peringatan: Terdeteksi Kantuk! berwarna merah muncul di bagian bawah layar

2.10.2 Skenario Pengujian *User Acceptance Testing*

Pada **Tabel 11 Skenario User Acceptance Testing** Pengujian sistem dengan calon pengguna adalah hal yang sangat penting dikarenakan ini menjadi salah satu bentuk umpan balik langsung dari calon pengguna yang akan menggunakan sistem deteksi kantuk tersebut. Melalui pengujian ini, dapat diketahui apakah sistem telah berjalan sesuai dengan kebutuhan pengguna, mudah digunakan, serta mampu memberikan hasil yang akurat dan dapat dipahami. Selain itu, User Acceptance Testing juga bertujuan untuk memastikan bahwa fitur-fitur yang dikembangkan telah memenuhi ekspektasi pengguna sebelum sistem diimplementasikan secara penuh.

Tabel 11 Skenario User Acceptance Testing

No	Skenario Pengujian	Langkah Pengguna	Pertanyaan Pendukung	Kriteria Kelulusan
UAT-01	Kejelasan Onboarding	Pengguna Menginstal dan mengikuti alur onboarding	1. Apakah kamu dapat menginstal aplikasi tanpa kesulitan? 2. Apakah alur onboarding mudah dipahami tanpa perlu bantuan tambahan? 3. Apakah instruksi pada layar onboarding jelas dan informatif? 4. Apakah ada bagian dari onboarding yang membingungkan atau tidak relevan?	Pengguna dapat menginstal dan memahami alur onboarding tanpa bantuan.
UAT-02	Kejelasan Tampilan Data	Mulai deteksi dan amati tampilan data *PERCLOS* dan *Heart Rate.*	1. Apakah kamu dapat memahami arti dari parameter *PERCLOS* dan Heart Rate yang ditampilkan? 2. Apakah tampilan data mudah dibaca dan informatif? 3. Apakah ada elemen data yang membingungkan atau tidak diperlukan? 4. Apakah kamu merasa visualisasi data membantu memahami kondisimu?	Pengguna dapat memahami arti dari parameter yang ditampilkan.

No	Skenario Pengujian	Langkah Langkah Pengguna	Pertanyaan Pendukung	Kriteria Kelulusan
UAT-03	Respons Peringatan	Pura-pura mengantuk hingga alarm berbunyi	1. Apakah kamu dapat mendengar atau merasakan alarm peringatan dengan jelas? 2. Apakah peringatan (suara, getaran, tampilan visual) terasa cukup efektif untuk menarik perhatianmu? 3. Apakah peringatan tersebut tidak mengganggu kenyamanan berkendara? 4. Menurutmu, apakah waktu munculnya alarm sudah tepat?	Pengguna menganggap respons aplikasi (alarm suara, getaran) cukup efektif dan tidak mengganggu
UAT-04	Kemudahan Interaksi	Tekan tombol *Quit* atau *Reset*	1. Apakah tombol *Quit* dan *Reset* mudah ditemukan? 2. Apakah kamu dapat menghentikan atau mereset aplikasi tanpa kesulitan? 3. Apakah interaksi dengan tombol berjalan lancar tanpa bug atau delay?	Pengguna dapat menghentikan atau mereset aplikasi dengan mudah.

No	Skenario Pengujian	Langkah Langkah Pengguna	Pertanyaan Pendukung	Kriteria Kelulusan
UAT-05	Perasaan Aman dan nyaman	Gunakan aplikasi selama beberapa menit	1. Apakah kamu merasa lebih aman saat berkendara dengan aplikasi ini aktif? 2. Apakah tampilan dan respons aplikasi membuatmu merasa tenang? 3. Apakah ada aspek dari aplikasi yang justru membuatmu tidak nyaman? 4. Secara keseluruhan, apakah kamu bersedia menggunakan aplikasi ini lagi di masa depan?	Pengguna merasa lebih aman dan nyaman saat berkendara dengan aplikasi ini