

Perancangan Sistem Deteksi Kondisi *Microsleep* Berbasis Sensor MPU6050 dan *Support Vector Machine*

Achmad Bayhaqi Putra Adira¹, Denny Trias Utomo^{2*}

^{1,2} Teknik Informatika, Politeknik Negeri Jember, Jember, Indonesia

Article Info

Article history:

Received 6 Maret 2026

Revised 12 Maret 2026

Accepted 13 Maret 2026

Keywords:

Microsleep

Internet of Things

Support Vector Machine

Wearable Device

Sensor MPU6050

ABSTRACT

Kecelakaan lalu lintas di Indonesia didominasi oleh kendaraan roda dua, di mana faktor kesalahan manusia seperti kelelahan dan *microsleep* menjadi penyebab utama. Kondisi *microsleep* terjadi dalam durasi singkat dan sering kali tidak disadari, sehingga sangat berbahaya bagi pengendara motor. Penelitian ini bertujuan untuk merancang sebuah sistem peringatan dini kondisi kantuk berbasis perangkat *wearable* yang terintegrasi pada helm. Sistem ini dirancang menggunakan sensor *Inertial Measurement Unit* (IMU) tipe MPU6050 untuk mendeteksi pola pergerakan kepala secara *real-time*. Data akselerasi dan orientasi kepala diproses menggunakan mikrokontroler ESP32 dengan menerapkan algoritma *Machine Learning* yaitu *Support Vector Machine* (SVM) untuk mengklasifikasikan kondisi sadar dan *microsleep*. Jika sistem mendeteksi indikasi *microsleep*, aktuator berupa *buzzer* dan motor getar akan aktif untuk memberikan peringatan kepada pengendara. Pendekatan perancangan ini menawarkan solusi yang ergonomis dan portabel dibandingkan metode deteksi berbasis kamera yang kurang efektif bagi pengendara bermotor.

Copyright © 2026 Akiratech
Journal

All rights reserved

DOI : <https://doi.org/10.63935/akiratech.v3i1.287>

Corresponding Author:

Denny Trias Utomo,

Politeknik Negeri Jember, Jl. Mastrip No 146, Jember 68100, Indonesia

Email: denny.trias@polije.ac.id

1. PENDAHULUAN

Keselamatan lalu lintas di Indonesia masih menjadi perhatian utama mengingat tingginya angka kecelakaan yang terjadi setiap tahunnya. Berdasarkan data statistik laka lantas Pusiknas Polri periode 2023-2024, tercatat hampir 300.000 kasus kecelakaan, di mana kendaraan roda dua menjadi penyumbang terbesar insiden tersebut[1]. Analisis faktor penyebab menunjukkan bahwa dominasi kecelakaan, sekitar 95%, disebabkan oleh faktor kesalahan manusia (*human error*). Salah satu kondisi fisiologis yang paling berbahaya dan sering tidak disadari oleh pengendara adalah *microsleep*, yaitu hilangnya kesadaran sementara akibat kelelahan atau kantuk yang berlebihan. Bagi pengendara motor, kondisi ini sangat fatal karena hilangnya keseimbangan dalam hitungan detik dapat berujung pada kecelakaan serius.

Berbagai upaya penelitian telah dilakukan untuk mengembangkan sistem deteksi kantuk. Pendekatan yang umum digunakan adalah metode berbasis *computer vision* yang mendeteksi kelelahan melalui fitur wajah atau kedipan mata menggunakan metode facial landmark[2]. Penelitian lain juga telah mengimplementasikan algoritma *Support Vector Machine* (SVM) berbasis pengolahan citra untuk mendeteksi *microsleep* dengan tingkat akurasi yang tinggi[3]. Meskipun metode visual ini efektif untuk pengemudi mobil, penerapannya pada pengendara sepeda motor menghadapi kendala teknis yang signifikan. Penggunaan helm, masker pelindung, serta kondisi pencahayaan jalan yang dinamis membuat kamera gagal menangkap fitur wajah secara akurat dan konsisten.

Sebagai alternatif, pendekatan berbasis sensor gerak atau *Inertial Measurement Unit* (IMU) mulai dikembangkan. Penelitian sebelumnya telah merancang purwarupa helm pintar berbasis *Internet of Things*

(IoT) menggunakan mikrokontroler NodeMCU untuk memantau kondisi pengendara dengan mengirimkan data ke *platform* Firebase[4]. Meskipun demikian, pendekatan tersebut masih menyisakan celah riset yang krusial. Sistem yang ada sangat bergantung pada koneksi nirkabel dan pemrosesan berbasis *cloud computing*. Ketergantungan pada transmisi *cloud* ini mengakibatkan tingginya waktu tunda (*latency*) serta berisiko gagal merespons jika terjadi gangguan sinyal jaringan di jalan raya, sehingga kurang ideal untuk skenario keselamatan *real-time*. Untuk mengatasi kelemahan tersebut, deteksi yang akurat dan responsif memerlukan algoritma klasifikasi yang handal namun efisien untuk ditanamkan dan dieksekusi langsung secara mandiri pada perangkat mikrokontroler (*edge computing*). Integrasi pemrosesan data lokal dengan algoritma *Machine Learning* telah terbukti secara signifikan mampu memangkas waktu respons *real-time* dan menjaga keandalan sistem di berbagai sektor aplikasi pintar[5].

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan perancangan sistem peringatan dini *microsleep* yang terintegrasi langsung pada helm pengendara (helm pintar). Sistem ini dirancang menggunakan sensor MPU6050 untuk memantau akselerasi dan orientasi kepala, serta menerapkan algoritma *Machine Learning* yaitu *Support Vector Machine* (SVM) untuk mengklasifikasikan kondisi pengendara. Pemilihan SVM didasarkan pada kemampuannya dalam menangani klasifikasi data dengan margin pemisah yang optimal. Kontribusi utama dari perancangan ini adalah integrasi sistem deteksi berbasis gerak ke dalam perangkat *wearable* yang mampu memberikan notifikasi peringatan secara mandiri dan *real-time*, sehingga diharapkan dapat meminimalisir risiko kecelakaan akibat kelalaian manusia.

2. TINJAUAN PUSTAKA

2.1. *Microsleep*

Microsleep didefinisikan sebagai kondisi hilangnya kesadaran sementara yang terjadi secara mendadak dalam durasi yang sangat singkat, umumnya berkisar antara 1 hingga 15 detik. Fenomena ini sering kali muncul pada individu yang mengalami kelelahan fisik ekstrem atau sedang melakukan aktivitas yang bersifat monoton dalam jangka waktu lama, namun tetap memaksakan diri untuk terjaga. Secara neurologis, *microsleep* terkait erat dengan instabilitas struktur mikro tidur (*micro-sleep instability*) dan abnormalitas transisi gelombang otak (seperti yang dapat terekam pada sinyal *electroencephalographic* atau EEG), di mana otak memasuki fase istirahat sesaat meskipun secara fisik individu tersebut tampak terjaga dengan mata yang mungkin masih terbuka[6]. Dalam konteks keselamatan transportasi, kondisi ini dikategorikan sangat berbahaya karena penurunan kewaspadaan yang drastis dapat menyebabkan pengemudi kehilangan kendali atas kendaraannya, yang berkontribusi signifikan terhadap tingginya risiko kecelakaan lalu lintas[7].

2.2. Sensor MPU6050

MPU6050 merupakan modul sensor *Inertial Measurement Unit* (IMU) yang mengintegrasikan *accelerometer* 3-sumbu dan *gyroscope* 3-sumbu dalam satu *chip* berbasis teknologi *Micro-Electro-Mechanical Systems* (MEMS). Komponen *accelerometer* berfungsi mengukur percepatan linier berdasarkan perubahan gaya gravitasi untuk mendeteksi orientasi statis, sedangkan *gyroscope* mengukur kecepatan sudut (*angular velocity*) yang responsif terhadap gerakan rotasi dinamis. Keunggulan utama sensor ini adalah adanya *Digital Motion Processor* (DMP) yang mengurangi beban komputasi mikrokontroler, serta penggunaan antarmuka I2C yang efisien untuk integrasi sistem *wearable*[8]. Dalam penerapannya pada sistem pemantauan (*monitoring*), penelitian sebelumnya telah membuktikan keandalan sensor MPU6050 untuk mendeteksi kondisi *microsleep* pada pengendara sepeda motor melalui pengukuran perubahan sudut kemiringan kepala. Penggunaan sensor ini terbukti sangat presisi dalam membaca orientasi gerak, di mana pengujian klasifikasi kondisi kantuk pada rentang sudut kemiringan 10° hingga 30° dan kondisi normal pada rentang sudut 0° hingga 6° berhasil mencapai tingkat akurasi sebesar 100%[9].

2.3. Mikrokontroler ESP32

ESP32 adalah mikrokontroler jenis *System-on-Chip* (SoC) yang dirancang untuk efisiensi tinggi dalam aplikasi *Internet of Things* (IoT). Perangkat ini ditenagai oleh prosesor *dual-core* Xtensa 32-bit LX6 yang mampu beroperasi hingga kecepatan 240 MHz, memungkinkan pemrosesan data sensor dan eksekusi algoritma klasifikasi dilakukan secara paralel dengan cepat. Selain performa komputasi yang tinggi, ESP32 memiliki fitur konektivitas nirkabel terintegrasi (Wi-Fi dan Bluetooth) serta manajemen daya yang efisien, menjadikannya platform yang ideal untuk perangkat *wearable* bertenaga baterai yang membutuhkan kemampuan komunikasi data dan pengolahan sinyal secara *real-time*[10]. Penggunaan mikrokontroler nirkabel semacam ini untuk pemrosesan data sensor secara berkelanjutan juga telah terbukti andal dan sangat efisien dalam mendukung keakuratan sistem otomatis cerdas[11].

2.4. Support Vector Machine (SVM).

Support Vector Machine (SVM) merupakan algoritma *supervised learning* yang dirancang untuk menyelesaikan masalah klasifikasi dengan memetakan data *input* ke *output* berdasarkan pola yang dipelajari selama fase pelatihan[12]. Prinsip kerja utama SVM adalah menemukan *hyperplane* atau bidang pemisah optimal yang memiliki margin terluas antar kelas data. Lebih lanjut, algoritma ini memiliki keunggulan signifikan dalam memecahkan masalah klasifikasi yang sangat non-linier melalui penerapan teknik *kernel* (*kernel trick*) yang mampu mengeksplorasi geometri pada data[13].

Selain memiliki stabilitas yang baik terhadap *overfitting* serta kemampuan menangani data berdimensi tinggi[12], SVM juga secara luas diakui sebagai metode klasifikasi yang efisien dan andal. Hal ini dibuktikan dengan kemampuannya menghasilkan tingkat akurasi yang tinggi pada berbagai aplikasi sistem deteksi dan pengenalan pola yang kompleks[14]. Karakteristik komprehensif ini menjadikan SVM sebagai metode yang sangat efektif untuk mengklasifikasikan fluktuasi pola sinyal sensor gerak guna membedakan kondisi sadar dan *microsleep* pada pengemudi secara presisi.

3. METODE PENELITIAN

Penelitian ini menerapkan metode eksperimental dengan pendekatan prototyping untuk mengembangkan sistem peringatan dini *microsleep* berbasis perangkat wearable. Pendekatan prototyping ini diadopsi karena memfasilitasi pengujian iteratif yang cepat, sehingga fungsionalitas dan keandalan sistem dapat divalidasi dan disesuaikan secara efektif di lingkungan nyata[15]. Fokus utama penelitian adalah merancang bangun sistem tertanam (*embedded system*) yang mengintegrasikan sensor gerak dan algoritma *Machine Learning* agar dapat beroperasi secara *real-time*. Tahapan pelaksanaan penelitian disusun secara sistematis yang mencakup empat fase utama, yaitu: (1) perancangan perangkat keras, (2) akuisisi serta pra-pemrosesan data untuk pembentukan *dataset*, (3) implementasi algoritma klasifikasi *Support Vector Machine* (SVM) ke dalam mikrokontroler, dan (4) perancangan pengujian performa sistem dalam mendeteksi kondisi kantuk.

3.1. Perancangan Perangkat Keras

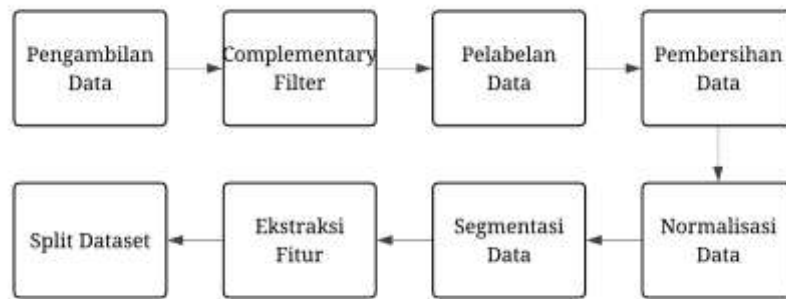
Perancangan sistem perangkat keras berfokus pada integrasi mikrokontroler ESP32 sebagai unit pemroses utama dengan sensor dan aktuator untuk membentuk perangkat *wearable*. Sensor MPU6050 dihubungkan ke ESP32 menggunakan antarmuka komunikasi I2C sebagai input data gerak. Untuk memberikan notifikasi peringatan kepada pengguna, sistem menggunakan *buzzer* aktif dan motor getar sebagai media keluaran. Seluruh komponen elektronik ini ditenagai oleh baterai Li-Ion 3.7V berkapasitas 1000mAh, yang dilengkapi dengan modul pengisi daya TP4056 dan saklar utama untuk manajemen daya.

3.2. Akuisisi dan Pra-pemrosesan Data

Data penelitian diperoleh melalui eksperimen pengambilan data gerak kepala menggunakan sensor MPU6050 yang terpasang pada helm. Sensor ini merekam enam parameter utama secara *real-time*, yaitu akselerasi linier dan kecepatan sudut. Data tersebut dikirimkan dari mikrokontroler ESP32 ke komputer melalui komunikasi serial dan disimpan dalam format *Comma-Separated Values* (CSV) untuk pembentukan *dataset*.

Proses akuisisi data ini melibatkan 3 orang subjek eksperimen. Untuk memastikan resolusi pergerakan kepala terekam secara presisi tanpa membebani komputasi mikrokontroler, sensor MPU6050 dikonfigurasi dengan tingkat pengambilan sampel (*sampling rate*) sebesar 10 Hz. Pengambilan data untuk setiap subjek dilakukan dalam dua skenario utama guna mendapatkan variasi pola gerakan yang representatif: (1) sesi perekaman kondisi sadar saat melakukan simulasi berkendara normal dengan durasi 15 menit per subjek, dan (2) sesi perekaman simulasi kondisi mengantuk (*microsleep*) dengan durasi 2 menit per subjek.

Setelah data mentah terkumpul, dilakukan tahapan pra-pemrosesan untuk meningkatkan kualitas data sebelum digunakan dalam pelatihan model. Alur lengkap proses ini ditunjukkan pada Gambar 2.



Gambar 1. Alur Pelabelan dan Pra-Pemrosesan Data

Tahap pertama pra-pemrosesan adalah penerapan *Complementary Filter* untuk menggabungkan data akselerometer dan *gyroscope*. Metode ini bertujuan mengurangi *noise* dari akselerometer dan *drift* dari *gyroscope*, sehingga menghasilkan estimasi sudut orientasi (seperti *pitch* dan *roll*) yang lebih stabil dan akurat. Selanjutnya, dilakukan pelabelan data secara manual di mana data diberi label '0' untuk kondisi normal dan '1' untuk kondisi *microsleep* berdasarkan catatan observasi saat eksperimen.

Data yang telah dilabeli kemudian dibersihkan dari nilai yang hilang (*missing values*) atau anomali. Untuk memastikan setiap fitur memiliki kontribusi yang setara dalam proses klasifikasi, dilakukan normalisasi menggunakan metode *Min-Max Scaling*. Terakhir, data disegmentasi menggunakan teknik *sliding window* dengan interval waktu tertentu (misalnya 2 detik) dan dilakukan ekstraksi fitur statistik (seperti rata-rata, standar deviasi, dan rentang nilai) pada setiap segmen. Hasil ekstraksi inilah yang menjadi input akhir untuk algoritma klasifikasi.

3.3. Implementasi Algoritma Klasifikasi

Implementasi algoritma dilakukan melalui dua tahap utama: pelatihan model (*training*) dan penanaman model (*deployment*) pada mikrokontroler. Pada tahap pelatihan, *dataset* yang telah diproses dibagi menjadi data latih (70%) dan data uji (30%) untuk memvalidasi kemampuan generalisasi model. Pelatihan dilakukan menggunakan bahasa pemrograman Python dengan pustaka *Scikit-Learn*. Algoritma *Support Vector Machine* (SVM) dikonfigurasi menggunakan teknik *Grid Search* dengan validasi silang (*Cross-Validation*) untuk menemukan kombinasi *hyperparameter* optimal, seperti jenis kernel, nilai C, dan Gamma, yang menghasilkan akurasi terbaik dalam membedakan kelas sadar dan *microsleep*.

Setelah model optimal diperoleh, tahap selanjutnya adalah konversi model untuk lingkungan *embedded*. Model yang tersimpan dalam format Python (.pkl) dikonversi menjadi format array C atau *header file* (.h) agar kompatibel dengan memori mikrokontroler. Kode program klasifikasi kemudian ditulis menggunakan Arduino IDE, di mana array model SVM diintegrasikan ke dalam *firmware* ESP32. Dalam implementasinya, mikrokontroler diprogram untuk melakukan ekstraksi fitur dari data sensor secara *real-time*, menghitung prediksi menggunakan bobot model yang telah ditanam, dan menghasilkan sinyal keluaran untuk mengontrol aktuator tanpa memerlukan koneksi ke komputer eksternal.

3.4. Rancangan Skenario Pengujian

Sebagai tahap lanjutan untuk memvalidasi fungsionalitas dan akurasi perangkat *wearable* yang telah dirancang, penelitian ini mengusulkan sebuah skenario pengujian di lingkungan nyata. Pengujian ini nantinya direncanakan untuk melibatkan subjek yang mengenakan helm purwarupa dan melakukan simulasi berkendara dalam dua kondisi skenario: kondisi sadar (normal) dan simulasi kondisi mengantuk (*microsleep*).

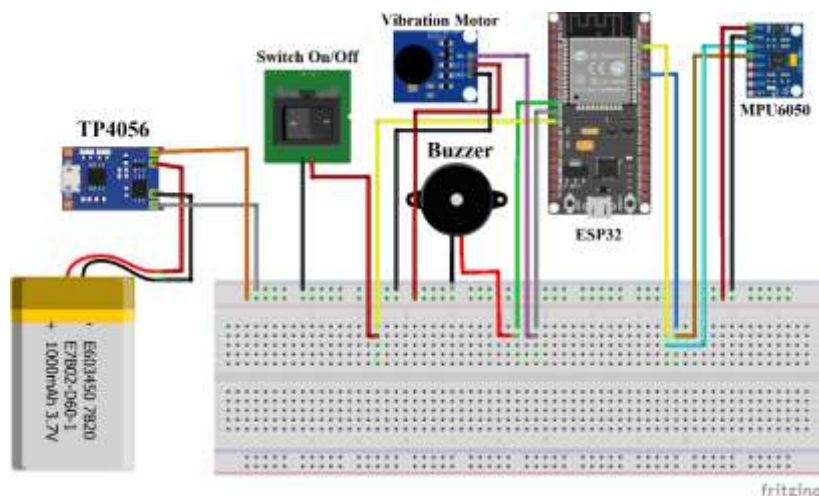
Selama pengujian, sistem dirancang untuk melakukan klasifikasi secara kontinu. Data hasil prediksi yang dikeluarkan oleh ESP32 akan dicatat dan dibandingkan dengan label kondisi aktual pengguna (*ground truth*). Hasil perbandingan antara prediksi sistem dan kondisi aktual tersebut nantinya akan dihitung menggunakan *Confusion Matrix* untuk mendapatkan parameter evaluasi performa kuantitatif, yang meliputi Akurasi, Presisi, Recall, dan F1-Score pada tahap pengembangan selanjutnya.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Implementasi sistem perangkat keras menghasilkan unit *wearable* yang mengintegrasikan mikrokontroler, sensor, dan aktuator dalam satu kesatuan rangkaian. Fokus utama implementasi adalah

memastikan jalur komunikasi data dan distribusi daya berjalan sesuai spesifikasi desain. Skema lengkap rangkaian sistem yang telah diimplementasikan ditunjukkan pada Gambar 2.



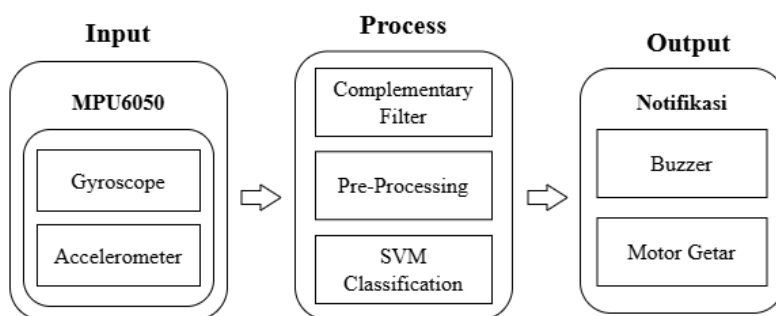
Gambar 2. Skema Implementasi Perangkat Keras

Berdasarkan skema di atas, sensor MPU6050 terhubung ke ESP32 melalui jalur I2C, di mana pin SDA dihubungkan ke GPIO 21 dan pin SCL ke GPIO 22. Konfigurasi ini memungkinkan transfer data akselerasi dan *gyroscope* dengan kecepatan tinggi. Untuk mekanisme peringatan, *buzzer* dihubungkan ke pin GPIO 25 sedangkan motor getar dihubungkan ke pin GPIO 26. Kedua aktuator ini dikonfigurasi sebagai *output* digital aktif-tinggi.

Selain itu, sebuah tombol fisik (*push button*) diintegrasikan pada pin GPIO 27 yang berfungsi sebagai pemicu aktivasi sistem atau kalibrasi awal. Untuk kebutuhan daya, tegangan dari baterai Li-Ion 3.7V disalurkan melalui modul TP4056 menuju pin VIN dan GND pada ESP32, memastikan suplai tegangan stabil 5V ke mikrokontroler serta memfasilitasi pengisian ulang baterai yang aman.

4.2. Logika Pemrosesan Data

Logika pemrosesan data yang ditanamkan pada mikrokontroler ESP32 dirancang untuk bekerja secara siklik dan *real-time*. Alur kerja sistem mengikuti prinsip *Input-Process-Output* (IPO) untuk memastikan setiap data gerakan yang masuk segera dianalisis dan direspons. Diagram alur logika sistem secara lengkap ditunjukkan pada Gambar 3.



Gambar 3. Diagram Logika Pemrosesan Data (IPO)

Pada tahap Input, sistem secara kontinu membaca data mentah akselerasi dan kecepatan sudut dari sensor MPU6050. Data ini kemudian masuk ke tahap Process, yang diawali dengan penerapan *Complementary Filter* untuk mendapatkan nilai orientasi (*pitch* dan *roll*) yang stabil dan minim *noise*. Nilai orientasi tersebut kemudian diekstraksi fiturnya dan dijadikan masukan bagi algoritma klasifikasi SVM. Model SVM yang telah tertanam di dalam memori mikrokontroler akan melakukan perhitungan matematis (*dot product*) untuk memprediksi kondisi pengendara: Kelas 0 (Sadar) atau Kelas 1 (*Microsleep*).

Tahap Output dieksekusi berdasarkan hasil prediksi tersebut. Jika logika sistem mendeteksi keluaran Kelas 1 (*Microsleep*), mikrokontroler akan segera mengirimkan sinyal digital *HIGH* ke pin GPIO 25 dan GPIO

26 untuk mengaktifkan *buzzer* dan motor getar secara bersamaan. Sebaliknya, jika terdeteksi Kelas 0, sistem akan mempertahankan status *LOW* (diam) pada kedua aktuator dan kembali melakukan pembacaan data sensor siklus berikutnya.

4.3. Pembahasan

Desain sistem yang diusulkan menawarkan solusi alternatif terhadap tantangan deteksi kantuk yang selama ini didominasi oleh pendekatan *computer vision*. Berbeda dengan metode berbasis kamera yang sangat bergantung pada kondisi pencahayaan dan visibilitas wajah, penggunaan sensor inersia (MPU6050) pada desain ini memberikan keunggulan signifikan dalam hal reliabilitas lingkungan. Sistem tidak terpengaruh oleh intensitas cahaya maupun obstruksi fisik seperti penggunaan masker atau kaca helm yang gelap, yang sering menjadi kendala utama pada metode deteksi wajah atau kedipan mata bagi pengendara motor.

Dari sisi efisiensi komputasi, penerapan algoritma *Support Vector Machine* (SVM) pada mikrokontroler ESP32 menunjukkan pendekatan yang lebih efisien dibandingkan penggunaan *Deep Learning* seperti *Convolutional Neural Network* (CNN). SVM memiliki kompleksitas komputasi yang lebih rendah namun tetap efektif menangani data berdimensi tinggi, sehingga memungkinkan proses klasifikasi dilakukan secara *real-time* langsung pada perangkat (*edge computing*) tanpa memerlukan prosesor berspesifikasi tinggi atau konsumsi daya yang besar. Hal ini menjadikan sistem lebih hemat energi dan responsif dalam memberikan peringatan dini.

Keunggulan lain dari rancangan ini adalah bentuk implementasinya sebagai perangkat *wearable* yang terintegrasi langsung pada helm (*smart helmet*). Desain ini menjawab keterbatasan sistem deteksi kantuk konvensional yang sering kali tidak praktis untuk kendaraan roda dua. Dengan menempatkan seluruh komponen (sensor, pemroses, dan aktuator) pada helm, sistem dapat memantau kondisi pengendara secara kontinu tanpa mengganggu kenyamanan atau konsentrasi berkendara. Integrasi ini menjadikan sistem bersifat portabel dan mandiri, sehingga dapat berfungsi optimal sebagai fitur keselamatan aktif bagi pengendara sepeda motor.

5. KESIMPULAN

Penelitian ini berhasil merancang bangun arsitektur sebuah sistem peringatan dini kondisi *microsleep* bagi pengendara sepeda motor yang terintegrasi penuh dalam perangkat *wearable* berupa helm. Sistem dirancang menggunakan sensor MPU6050 untuk mengakuisisi data pergerakan kepala dan mikrokontroler ESP32 sebagai unit pemrosesan utama yang menjalankan algoritma klasifikasi *Support Vector Machine* (SVM). Berdasarkan hasil implementasi rancangan, purwarupa sistem ini secara fungsional telah siap untuk beroperasi secara mandiri dalam memberikan respons peringatan *real-time* melalui notifikasi suara dan getaran tanpa ketergantungan pada komputer eksternal. Adapun evaluasi kuantitatif untuk mengukur performa algoritma secara pasti meliputi tingkat akurasi, presisi, recall, dan F1-score akan menjadi fokus pada tahapan riset lanjutan (*future work*) ketika purwarupa diuji secara komprehensif pada skenario berkendara yang sesungguhnya.

Ucapan Terimakasih

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan dalam penyusunan artikel ini sehingga penelitian ini dapat diselesaikan dengan baik.

REFERENSI

- [1] Pusiknas Polri, "Data Kecelakaan Lalu Lintas 2023-2024, Kepolisian Negara Republik Indonesia," https://pusiknas.polri.go.id/laka_lantas.
- [2] I. K. Y. T. Pranata, C. Ramadhani, and G. Wahyu Wiriasto, "Menggunakan Metoda Facial Landmark Detection Berbasis Raspberry PI 3 Modul B," *Dielektrika - Department of Electrical Engineering University of Mataram*, vol. 10, no. 2, pp. 100–111, 2023.
- [3] Sugeng and H. Praminiarto, "Detection of Drowsiness in Drivers Using Image Processing and Support Vector Machine (SVM) Classification," *Jurnal Online Informatika*, vol. 9, no. 2, pp. 238–248, Aug. 2024, doi: 10.15575/join.v9i2.1076.
- [4] P. Prasetyawan, S. Samsugi, and R. Prabowo, "Internet of Thing Menggunakan Firebase dan Nodemcu untuk Helm Pintar," *Jurnal ELTIKOM*, vol. 5, no. 1, pp. 32–39, Mar. 2021, doi: 10.31961/eltikom.v5i1.239.
- [5] D. T. Utomo *et al.*, "Development of IS-Candling (intelligent system for candling) web application for IoT-based hatch fertility detection," *BIO Web Conf.*, vol. 210, p. 05003, 2026, doi: 10.1051/bioconf/202621005003.
- [6] M. Cesari *et al.*, "A data-driven system to identify REM sleep behavior disorder and to predict its progression from the prodromal stage in Parkinson's disease," *Sleep Med.*, vol. 77, pp. 238–248, Jan. 2021, doi: 10.1016/j.sleep.2020.04.010.
- [7] S. H. Zaleha *et al.*, "Microsleep Accident Prevention for SMART Vehicle via Image Processing Integrated with Artificial Intelligent," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Dec. 2021. doi: 10.1088/1742-6596/2129/1/012082.

- [8] J. M. Sultan, N. H. Zani, M. Azuani, S. Z. Ibrahim, and A. Md Yusop, "Analysis of Inertial Measurement Accuracy using Complementary Filter for MPU6050 Sensor," *Jurnal Kejuruteraan*, vol. 34, no. 5, pp. 959–964, Sep. 2022, doi: 10.17576/jkukm-2022-34(5)-24.
- [9] S. A. Maqdis, A. Adiwilaga, and M. Munawir, "DESIGN OF MICROSLEEP DETECTION SYSTEM IN 32-BIT MICROCONTROLLER-BASED MOTORISTS WITH RANDOM FOREST METHOD," *JIKO (Jurnal Informatika dan Komputer)*, vol. 7, no. 2, pp. 131–137, Aug. 2024, doi: 10.33387/jiko.v7i2.8539.
- [10] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and Implementation of ESP32-Based IoT Devices," *Sensors*, vol. 23, no. 15, Aug. 2023, doi: 10.3390/s23156739.
- [11] E. Rosdiana, F. Y. Ali, E. R. Pristiwaningsih, R. Purbaningtyas, D. R. Hartadi, and D. T. Utomo, "IoT for Agricultural Innovation: Enhancing Robusta Coffee Seedling Growth in Controlled Environments with Intelligent Air Quality Control System (IAQCS)," *International Journal of Technology, Food and Agriculture*, vol. 2, no. 1, pp. 1–9, Dec. 2024, doi: 10.25047/tefa.v2i1.5752.
- [12] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning: Methods and Applications to Brain Disorders*, Elsevier, 2020, pp. 101–121. doi: 10.1016/B978-0-12-815739-8.00006-7.
- [13] D. Valkenborg, A. J. Rousseau, M. Geubbelmans, and T. Burzykowski, "Support vector machines," Nov. 01, 2023, *Elsevier Inc.* doi: 10.1016/j.ajodo.2023.08.003.
- [14] D. M. Abdullah and A. M. Abdulazeez, "Machine Learning Applications based on SVM Classification: A Review," *Qubahan Academic Journal*, vol. 1, pp. 81–90, Apr. 2021, doi: <https://doi.org/10.48161/qaj.v1n2a50>.
- [15] W. P. P. Witra, I. G. Anugrah, S. U. Nuha, N. S. Wibowo, and D. T. Utomo, "RAD-UCD Implementation in Project Management Information System," *Journal of Development Research*, vol. 8, no. 2, p. Process, Nov. 2024, doi: 10.28926/jdr.v8i2.398.