

## Sistem Pengenalan Gerakan Tangan Berbasis ESP32 dan Machine Learning untuk Kontrol Slide Power Point

Gevan Putra Yudhistira<sup>1</sup>, Dahnial Syauqy<sup>2</sup>, Barlian Henryranu Prasetyo<sup>3</sup>

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>putra.gevan00@gmail.com, <sup>2</sup>dahnial87@ub.ac.id, <sup>3</sup>barlian@ub.ac.id

### Abstrak

Penggunaan Microsoft Office Power Point dalam presentasi dewasa ini masih berkecenderungan menggunakan perangkat keras seperti keyboard dan mouse sebagai kontrol sistem. Perangkat keyboard menjadi utama untuk berkomunikasi dengan sistem komputer dengan mengandalkan perangkat fisik seperti clicker atau bahkan keyboard untuk berpindah antar slide. Perkembangan teknologi interaksi manusia dan komputer mendorong munculnya alternatif kendali presentasi yang lebih intuitif, salah satunya melalui pengenalan gerakan tangan. Tipe penelitian skripsi ini adalah pengembangan implementatif, bertujuan merancang dan mengimplementasikan sistem pengenalan gestur tangan berbasis ESP32 dan Machine Learning untuk mengontrol perpindahan slide pada Microsoft PowerPoint secara nirkabel. Sistem ini menggunakan sensor MPU6050 untuk menangkap data akselerasi dan rotasi tangan, serta algoritma Random Forest sebagai model klasifikasi gerakan. Data sensor dikirim melalui komunikasi Bluetooth Low Energy (BLE) dari ESP32 ke komputer dan diterjemahkan menjadi perintah keyboard. Proses klasifikasi dilakukan langsung pada mikrokontroler secara real-time. Hasil pengujian menunjukkan bahwa model memiliki akurasi rata-rata sebesar 86,7% untuk mengenali tiga jenis gerakan (kanan, kiri, idle). Sistem ini mampu memberikan alternatif kendali presentasi yang efisien, serta berpotensi diadopsi oleh pengguna difabel sebagai solusi inklusif yang bebas perangkat genggam seperti clicker.

**Kata Kunci:** pengenalan gerakan tangan, ESP32, MPU6050, Random Forest, presentasi, sistem real-time

### Abstract

*The advancement of human-computer interaction technology has led to the emergence of more intuitive alternatives for presentation control, including hand gesture recognition. This study aims to design and implement a hand gesture recognition system based on ESP32 and Machine Learning to wirelessly control slide transitions in Microsoft PowerPoint. The system utilizes an MPU6050 sensor to capture acceleration and rotation data from hand movements, and employs the Random Forest algorithm for gesture classification. Sensor data is transmitted via Bluetooth Low Energy (BLE) from the ESP32 to a computer and converted into keyboard commands. The classification process is executed in real-time on the microcontroller. Testing results show that the model achieves an average accuracy of 86.7% in recognizing three gesture types (right, left, idle). This system provides an efficient alternative to conventional presentation tools, and has potential to be adopted by users with disabilities as an inclusive, hands-free solution.*

**Keywords:** hand gesture recognition, ESP32, MPU6050, Random Forest, presentation, real-time system

### 1. PENDAHULUAN

Perkembangan teknologi mengubah pola interaksi manusia–komputer dari perangkat input tradisional (keyboard, mouse) menuju antarmuka yang lebih alami seperti layar sentuh, perintah suara, dan gestur tangan. Untuk konteks presentasi, gestur tangan menarik karena membebaskan pengguna dari perangkat fisik

seperti clicker atau keyboard, sehingga mobilitas dan aliran komunikasi dengan audiens lebih terjaga. Ketergantungan pada perangkat fisik kerap mengganggu kelancaran presentasi—jangkauan terbatas, fokus terbagi, hingga hambatan ergonomi—sebagaimana disorot oleh laporan praktis dan temuan riset terkini yang menunjukkan sistem berbasis gestur dapat meningkatkan keluwesan gerak dan interaksi pembicara (Dy et al., 2024).

Pendekatan ini juga memberi nilai inklusivitas: pengguna dengan keterbatasan motorik halus dapat mengendalikan presentasi tanpa perlu menggenggam atau menekan perangkat kecil. Di sisi teknis, ketersediaan mikrokontroler modern seperti **ESP32**—berprosesor ganda, hemat daya, serta dilengkapi Wi-Fi dan **Bluetooth**—memungkinkan pemrosesan *real-time* di tepi (Oner, 2021). Digabung dengan sensor **IMU MPU6050** (akselerometer + giroskop tiga sumbu, antarmuka I<sup>2</sup>C), sistem dapat menangkap orientasi/gerak tangan secara akurat dalam paket yang ringkas dan mudah diintegrasikan (World Scientific News, 2025). Data gerak yang diperoleh dapat diproses di ESP32 dan dikirim ke komputer/piranti presentasi via Bluetooth, sehingga cocok untuk pengendalian slide bebas clicker sekaligus relevan bagi aplikasi lain seperti kendali drone, VR/AR, dan akses kontrol.

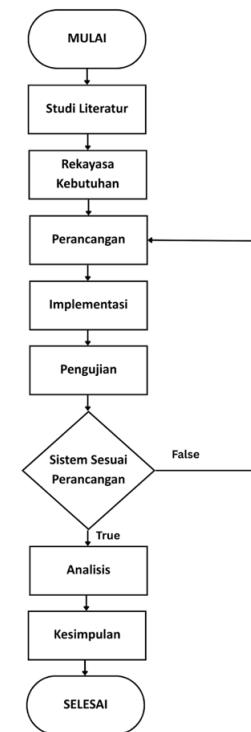
Untuk klasifikasi gestur, **Random Forest** dipilih karena stabil, tangguh terhadap *noise*, dan relatif tahan *overfitting* pada data sensor yang variatif; sebagai metode *ensemble*, ia menggabungkan banyak *decision tree* untuk meningkatkan akurasi dan generalisasi (Religia et al., 2021).

Berdasarkan latar tersebut, penelitian ini bertujuan merancang dan mengimplementasikan sistem pengenalan gerakan tangan berbasis **ESP32** dan **MPU6050** untuk mengendalikan perpindahan slide pada Microsoft PowerPoint. Diharapkan sistem meningkatkan kenyamanan dan mobilitas narasumber, menyediakan alternatif yang lebih inklusif bagi berbagai kondisi pengguna, serta menjadi landasan pengembangan aplikasi sejenis di domain lain.

## 2. METODOLOGI

Tipe penelitian ini adalah pengembangan implementatif, Sistem Pengenalan Gerakan Tangan Berbasis ESP32 dan Machine Learning sebagai kendali slide PowerPoint berbasis gestur tangan agar interaktif dan intuitif.

Penelitian ini dimulai dengan melakukan studi literatur dan analisa kebutuhan, perancangan dan implementasi sistem, pengumpulan data, pengolahan dan training model, evaluasi dan penyusunan kesimpulan dan saran. Gambar 1 ini adalah alur penelitian.



### 2.1 Presentasi

Sistem dirancang sebagai pengendali presentasi bebas-genggam: pengguna mengenakan sarung tangan berisi sensor IMU yang membaca gerak tangan, ESP32 memproses sinyal dan mengklasifikasikannya menjadi kanan, kiri, atau idle, lalu mengirim perintah keyboard panah melalui Bluetooth (BLE) untuk memindah slide di PowerPoint. Alur kendali mengikuti FSM (IDLE → WAIT\_FOR\_INPUT → PROCESSING → ACTION\_EXECUTED) agar response konsisten, sementara buzzer memberi umpan balik cepat ketika gestur dikenali. Kinerja dievaluasi dengan akurasi per kelas dan keseluruhan, latensi inferensi di ESP32, serta keandalan transmisi BLE saat sesi presentasi.

### 2.2 Teknik Pengumpulan Data

Data primer dikumpulkan dengan menempatkan **MPU6050** di punggung tangan. Subjek diminta melakukan gestur **kanan**, **kiri**, dan **idle** secara berulang pada durasi/jeda terkontrol. Sinyal akselerometer dan giroskop direkam setiap 25 ms dan dikemas per jendela 20 sampel ( $\pm 0,5$  s). Data dikirim *real-time* ke komputer dan disimpan (CSV), lalu **dilabeli manual** sesuai gestur yang sedang dilakukan; segmen ber-*noise* disaring. Setiap jendela dihitung 18 fitur statistik (min, maks, rata-rata

untuk 6 kanal: ax, ay, az, gx, gy, gz) sebagai representasi pola gerakan.

### 2.3 Random Forest

Model **Random Forest** dipilih karena stabil pada data sensor yang bervariasi dan berlatensi rendah di mikrokontroler. Pelatihan dilakukan **offline** (scikit-learn) menggunakan fitur terstandarisasi dan pemisahan latih–uji; hiperparameter dipilih ringkas untuk menjaga ukuran model. Model terlatih kemudian **dieksport ke C/C++** (mis. via *emlearn*) dan dipanggil di firmware ESP32 untuk inferensi **on-device**. Hasil klasifikasi dipetakan ke tombol panah kanan/kiri; prediksi *idle* tidak mengirim perintah.

### 2.4 Sensor

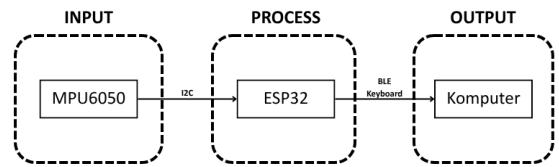
MPU6050 menggabungkan akselerometer dan giroskop tiga sumbu dengan antarmuka I<sup>2</sup>C, sehingga mudah diintegrasikan dengan ESP32. Pemasangan pada sarung tangan menjaga konsistensi orientasi sensor terhadap gerakan. Untuk meningkatkan ketahanan terhadap *noise*, sinyal dibaca periodik (25 ms), dikemas per jendela, dan diringkas menjadi fitur statistik yang lebih stabil untuk klasifikasi.

### 2.5 Mikrokontroler pada Sistem Tertanam

ESP32 berperan sebagai pusat komputasi tepi: membaca IMU, menyiapkan *buffer* jendela, menghitung fitur, menjalankan inferensi Random Forest, dan mengirim BLE ke host sebagai perintah keyboard. Implementasi difokuskan pada efisiensi memori (penggunaan array statis, pengurangan variabel sementara) dan latensi milidetik agar respons gestur terasa instan. Integrasi FSM menyederhanakan kontrol status dan *debounce* keputusan, sedangkan modul BLE memastikan kompatibilitas luas dengan perangkat presentasi tanpa perangkat tambahan.

## 3. Perancangan dan Implementasi

Untuk menjelaskan sebuah sistem diperlukan diagram blok yang menyangkut input, proses dan output. Diagram blok ini dapat memberikan gambaran secara umum bagaimana komponen utama dalam sistem saling terhubung dan berinteraksi.



Gambar 1. Diagram Blok Sistem

Sistem ini mempunyai beberapa komponen utama, yaitu sensor MPU6050 dan mikrokontroler ESP32. Perangkat-perangkat tersebut terletak di punggung tangan kanan. Sensor gerak MPU6050 akan mengirim data yang terhubung menggunakan I2C ke mikrokontroler ESP32. Di dalam ESP32, data-data tersebut akan diolah sesuai dengan logika program, lalu hasilnya dikirim sebagai tekanan tombol virtual via Bluetooth Low Energy (BLE) ke komputer tujuan dan buzzer. Dengan begitu, setiap perubahan Gerakan pada sensor MPU6050 bisa langsung diklasifikasi menjadi input keyboard di komputer ditandai dengan buzzer aktif sebagai umpan balik pengguna ketika gesture dikenali atau saat sistem masuk ke mode siap.

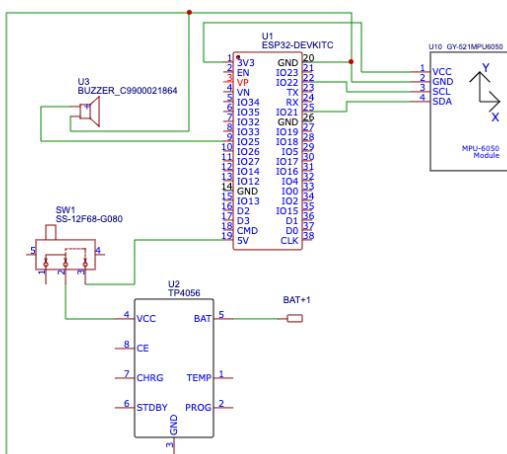
### 3.1. Perancangan Perangkat Keras



Gambar 2. Ilustrasi Perancangan Perangkat Keras

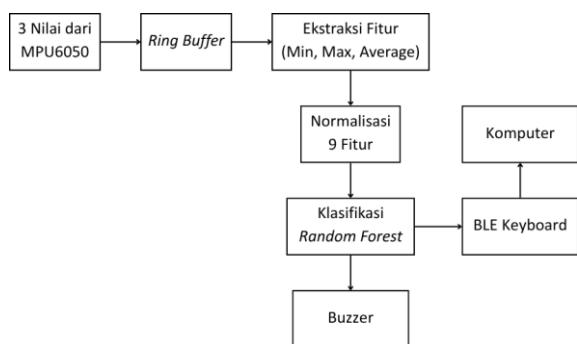
Perangkat keras dirancang sebagai sistem *wearable* berbasis sarung tangan kanan agar gerakan pengguna tetap natural dan posisi sensor konsisten. Arsitektur fisiknya terdiri atas sensor MPU6050 sebagai IMU, ESP32-DevKitC sebagai unit komputasi tepi dan antarmuka Bluetooth Low Energy (BLE) Keyboard, modul pengisian TP4056 dan baterai

18650 sebagai catu daya, serta buzzer aktif sebagai umpan balik auditori. MPU6050 ditempatkan di punggung tangan tepat di bawah ruas jari tengah untuk menangkap dinamika pergelangan dan telapak, ESP32 diposisikan tepat di bawahnya sebagai pusat pemrosesan, TP4056 dan baterai ditempatkan di sisi pergelangan agar distribusi bobot seimbang, sementara buzzer diletakkan di bawah ESP32 agar mudah dihubungkan dan tidak mengganggu gerakan. Jalur komunikasi antarperangkat menggunakan I<sup>2</sup>C dengan IO21 (SDA) dan IO22 (SCL) pada ESP32; sensor mendapat suplai 3V3 dari pin regulasi ESP32, dan buzzer dikendalikan melalui GPIO32. Tata letak kabel dibuat sesingkat mungkin untuk menjaga kerapian, meminimalkan interferensi mekanis, dan memudahkan perawatan serta pengisian daya.



Gambar 3. Diagram skematik mikrokontroler, Input/Output dan daya

### 3.2. Perancangan Perangkat Lunak



Gambar 4. Diagram proses secara umum

Perangkat lunak dirancang sebagai pipa pemrosesan dari sensor hingga keluaran BLE dengan kontrol berbasis Finite State Machine (FSM). Pada sisi pengumpulan data dan pelatihan, ESP32 mengekstrak fitur ringkas dari sinyal IMU—setiap jendela 20 sampel (interval

baca ±25 ms) menghasilkan 18 fitur statistik (minimum, maksimum, dan rata-rata untuk ax, ay, az, gx, gy, gz). Fitur dikirim dalam format CSV ke komputer, dilabeli (kanan, kiri, idle), kemudian dinormalisasi dengan Z-score; parameter mean dan standar deviasi disimpan untuk dipakai ulang saat inferensi. Model Random Forest dilatih (*offline*, scikit-learn) dengan *split* latih–uji terstratifikasi (mis. 80:20), lalu diekspor ke header C/C++ menggunakan emlearn agar dapat dijalankan di ESP32.

Pada sisi *runtime*, ESP32 menginisialisasi BLE Keyboard, IMU, dan buzzer, menerapkan *low-pass filter* sederhana ( $\alpha \approx 0,1$ ), serta mengelola alur melalui FSM IDLE → WAIT\_FOR\_INPUT → PROCESSING → ACTION\_EXECUTED. Deteksi posisi siap berbasis ambang akselerasi memicu *countdown* 3 detik, dilanjutkan akuisisi hingga 20 sampel atau batas waktu 5 detik. Jendela data dihitung fiturnya, dinormalisasi dengan parameter pelatihan, lalu diklasifikasikan oleh model tertanam. Hasil kanan/kiri dipetakan ke tombol panah BLE, sementara idle tidak mengirim perintah. Pola bunyi buzzer dibedakan untuk menandai siap, peringatan *countdown*, sukses, dan *idle/timeout*.

### 3.3. Implementasi Perangkat Keras



Gambar 5. Sarung Tangan

Implementasi fisik direalisasikan pada

sarung tangan hitam dengan papan PCB lubang sebagai dudukan komponen agar kokoh namun tetap ringan. MPU6050 direkatkan pada punggung tangan untuk menjaga orientasi tetap, ESP32 dan TP4056 dipasang rapat untuk meminimalkan jejak, baterai 18650 ditempatkan di sisi pergelangan agar keseimbangan tetap baik saat tangan berayun, dan buzzer dipasang dekat ESP32 untuk jalur singkat ke GPIO32. Koneksi listrik mengikuti rancangan: 3V3 → VCC MPU6050, GND bersama, IO21 → SDA, IO22 → SCL, VIN/OUT+ TP4056 → sakelar → ESP32, serta GPIO32 → buzzer. Setelah verifikasi sambungan dan penataan kabel, seluruh rangkaian dirapikan sehingga nyaman dipakai dan mudah diisi ulang tanpa melepas modul.

### 3.3. Implementasi Perangkat Lunak

Implementasi perangkat lunak terdiri dari tiga bagian yang saling terhubung. Pertama, akuisisi sensor dan ekstraksi fitur di ESP32: inisialisasi Wire.h dan pustaka IMU, pembacaan ax, ay, az, gx, gy, gz setiap ±25 ms, penerapan low-pass filter, ring buffer 20×6, lalu perhitungan min/maks/rata-rata per kanal untuk membentuk vektor fitur berdimensi 18; vektor dapat dicetak ke serial (CSV) untuk pembentukan dataset. Kedua, pelatihan model Random Forest di Python: penggabungan CSV, pelabelan konsisten, normalisasi (StandardScaler), split latih–uji terstratifikasi, pelatihan RandomForestClassifier, evaluasi dasar, penyimpanan mean/std ke berkas, dan konversi model ke header dengan emlearn (disertai eml\_common.h, eml\_trees.h, eml\_log.h). Ketiga, klasifikasi dan FSM di ESP32: pada runtime fitur dinormalisasi dengan array mean/std yang ditanam, diprediksi melalui fungsi predict dari model emlearn, kemudian hasil kanan/kiri dipetakan menjadi KEY\_RIGHT\_ARROW/KEY\_LEFT\_ARROW melalui pustaka BLEKeyboard (dengan pemeriksaan isConnected()), sedangkan idle diabaikan. FSM memastikan transisi yang deterministik (siap → countdown → akuisisi → proses → eksekusi), cooldown singkat mencegah eksekusi beruntun, dan buzzer memberikan umpan balik instan untuk status siap, peringatan satu detik sebelum akhir countdown, konfirmasi keberhasilan, maupun indikasi idle/timeout. Logging serial diaktifkan untuk memantau nilai terfilter, status FSM, dan hasil klasifikasi selama uji coba.

## 4. PENGUJIAN DAN ANALISIS

Bab ini memaparkan pengujian sistem serta analisis hasil untuk menjawab dua pertanyaan utama: (1) akurasi Random Forest dalam mengenali gestur tangan (*kanan, kiri, idle*) pada ESP32 secara real-time; (2) waktu komputasi inferensi model di ESP32.

### 4.1. Metodologi Pengujian

Tujuan & skenario. Model Random Forest (hasil pelatihan berbasis 18 fitur statistik: *min–max–mean* untuk ax, ay, az, gx, gy, gz) telah dikonversi ke *header C* melalui *emlearn* dan dijalankan on-device pada ESP32 tanpa komputer host. Pengujian akurasi dilakukan real-time pada dua kondisi penggunaan (duduk & berdiri), masing-masing 30 ulangan per kelas (*kanan, kiri, idle*) → total 90 percobaan.

Prosedur ringkas.

1. Nyalakan ESP32 & hubungkan BLE; pastikan sensor dan buzzer aktif.
2. Deteksi posisi siap → jalankan hitung mundur → lakukan gestur sesuai perintah.
3. Catat keluaran klasifikasi dari Serial Monitor dan respon tombol BLE.
4. Hitung akurasi per kelas & keseluruhan
5. Bandingkan hasil real-time dengan evaluasi berbasis data latih

### 4.2. Hasil Akurasi Klasifikasi – Real-Time (ESP32)

Tabel 1. Confusion Matrix Pengujian Secara Real-Time

Actual \ Predict	Kanan	Kiri	Idle	Total	Akurasi Kelas
Kanan	28	0	2	30	93.3%
Kiri	3	26	1	30	86.7%
Idle	1	5	24	30	80.0%
Jumlah	32	31	27	90	86.7% (total)

Metri kunci (per kelas):

1. Precision: Kanan 87.5%, Kiri 83.9%, Idle 88.9%
2. Recall: Kanan 93.3%, Kiri 86.7%, Idle 80.0%

3. F1-score: Kanan 90.3%, Kiri 85.2%, Idle 84.2%

Catatan: Gestur kanan paling stabil dikenali; idle paling menantang (sering tertukar ketika gerak sangat lambat).

#### 4.3. Hasil Evaluasi Berbasis Data Latih (Offline) & Perbandingan

Tabel 2. Confusion Matrix Pengujian Berdasarkan Data Latih

Actual \ Predict	Kanan	Kiri	Idle	Total	Akurasi Kelas
Kanan	28	0	2	30	93.3%
Kiri	3	26	1	30	86.7%
Idle	1	5	24	30	80.0%
Jumlah	32	31	27	90	86.7% (total)

Metri kunci (per kelas):

1. Precision: Kanan 89.2%, Kiri 85.7%, Idle 64.8%
2. F1-score: Kanan 83.8%, Kiri 81.5%, Idle 73.1%

Perbandingan & interpretasi.

1. Akurasi real-time 86.7% > akurasi offline 79.84%.
2. Peningkatan real-time kemungkinan dipengaruhi (i) konsistensi subjek saat eksekusi gestur, (ii) *thresholding* posisi siap dan *filtering* di firmware yang menstabilkan sinyal input.
3. Idle konsisten menjadi sumber salah klasifikasi (kemiripan fitur dengan gestur beramplitudo kecil). Perluas variasi data *idle*, tambahkan *debounce* keputusan, atau terapkan ambang kepercayaan untuk menahan keluaran meragukan.

#### 4.4. Pengukuran Waktu Komputasi (ESP32)

Tabel 3. Hasil Pengukuran Waktu

Statistik	Nilai
Rata-rata	1,409 μs
Minimum	1,197 μs

Maksimum	1,546 μs
Ulangan	30 kali

Inferensi di bawah 2 ms menunjukkan *pipeline* sangat responsif untuk interaksi real-time. Variasi sempit menegaskan efisiensi eksekusi model hasil konversi *emlearn* di perangkat berdaya terbatas

#### 4.5. Ringkasan Temuan & Implikasi

1. Sistem mencapai akurasi real-time 86.7% (Kanan 93.3%, Kiri 86.7%, Idle 80.0%) pada ESP32 dengan pengenalan *on-device*.
2. Waktu inferensi ≈ 1.41 ms cukup untuk kendali slide tanpa jeda yang terasa.
3. Kelas idle paling rawan salah klasifikasi → disarankan: perluasan data *idle*, *smoothing* tambahan, serta ambang kepercayaan (menahan output saat skor rendah).
4. Hasil menunjukkan kelayakan kendali presentasi bebas-genggam yang praktis, portabel, dan inklusif.

#### 5. KESIMPULAN

Pengujian sistem secara langsung dalam kondisi posisi duduk menunjukkan bahwa tingkat akurasi klasifikasi real-time ESP32 mencapai 86,7%, dengan akurasi tertinggi pada gesture kanan (93,3%), diikuti kiri (86,7%), dan idle (80%). Hal ini menunjukkan bahwa sistem dapat bekerja dengan baik dan konsisten untuk aplikasi gesture sederhana.

Waktu komputasi sistem ESP32 dalam melakukan klasifikasi gesture tercatat rata-rata sebesar 1409 mikrodetik (sekitar 1,41 milidetik). Nilai ini menunjukkan bahwa sistem mampu memberikan respons dalam waktu yang sangat cepat, cocok untuk kebutuhan interaksi real-time seperti kontrol slide atau presentasi.

#### 6. DAFTAR PUSTAKA

- Bhatele, S., & Bedekar, M. (2024). Sensor fusion for human activity recognition using accelerometer and gyroscope. Journal of Intelligent Sensor Systems, 10(1), 34–45.
- Cantisano, N., Rossi, S., et al. (2025). Wearable-based gesture detection for human-machine interaction: Review and challenges. ACM Transactions on Interactive Intelligent Systems, 15(1), 1–27.

- Cao, Y., & Qian, K. (2021). Sliding window approach for real-time gesture recognition using IMU sensors. *Sensors*, 21(4), 1120.
- DQLab. (2023). Random Forest untuk Pemula. Retrieved from <https://dqlab.id>
- Dasgupta, S., Krishnamurthy, A., et al. (2018). Real-time processing on embedded devices using ring buffer structures. *Embedded Systems Journal*, 12(3), 211–225.
- Dy, G., Lim, S., & Chen, R. (2024). Improving presentation flow with wearable input devices. *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, 1–8.
- Fadel, D. M., & Kareem, H. M. (2023). A comparative study of vision and sensor-based hand gesture recognition. *International Journal of Advanced Computer Science and Applications*, 14(2), 56–63.
- Han, J., Kamber, M., & Pei, J. (2011). *Data mining: Concepts and techniques* (3rd ed.). Elsevier.
- Jiang, Y., Hu, L., et al. (2020). Improving gesture recognition with overlapping sliding windows and sensor fusion. *Sensors*, 20(15), 4321.
- Khan, W., Lee, S., et al. (2021). Efficient hand motion recognition using IMU sensors and statistical features. *IEEE Sensors Journal*, 21(20), 22845–22856.
- Kwon, D., Kim, J., et al. (2019). Comparative evaluation of classification algorithms for IMU-based gesture recognition. *IEEE Transactions on Human-Machine Systems*, 49(5), 459–470.
- Lee, H., Zhang, Q., et al. (2022). Auditory feedback systems for wearable devices. *ACM Transactions on Interactive Systems*, 12(4), 28–40.
- Leo, J. (2024). *Embedded Systems: An Introduction*. Malang: UB Press.
- Microsoft Learn. (2024). *PowerPoint Object Model Reference*. Retrieved from <https://learn.microsoft.com/en-us/office/vba/api/overview/powerpoint>
- Oner, E. (2021). ESP32: Architecture, Features, and Applications. *Journal of Embedded Systems and Applications*, 14(2), 55–66.
- Pahl, G., Beitz, W., et al. (2007). *Engineering Design: A Systematic Approach*. Springer.
- Prabowo, R. A., Wibowo, S., & Hadi, F. (2021). Gesture classification using statistical features and random forest. *Journal of Intelligent Systems*, 6(2), 134–142.
- Pratama, R. A., Syauqy, D., & Budi, A. S. (2023). Sistem Monitoring Gerakan Sit-up berbasis Data Sensor Accelerometer dan Gyroscope menggunakan Algoritma K-Nearest Neighbor pada NodeMCU ESP8266. *Jurnal Ilmu Komputer dan Sistem Informasi*, 9(2), 102–110.
- Rollout. (n.d.). *PowerPoint Automation Overview*. Retrieved from <https://rollout.io>
- Sappa, A., Vintimilla, B., et al. (2020). Real-time hand gesture recognition using machine learning on low-power devices. *Pattern Recognition Letters*, 130, 27–35.
- Škulj, A., Novak, D., & Kalin, M. (2021). Inertial Measurement Units in Motion Analysis: A Review. *Measurement*, 172, 108654.
- T-vK. (2020). *BLEKeyboard Arduino Library*. Retrieved from <https://github.com/T-vK/ESP32-BLE-Keyboard>
- Utomo, H. (2021). Penerapan IMU dalam Robotika dan Navigasi. *Jurnal Teknik Elektro*, 10(1), 77–84.
- World Scientific News. (2025). *MPU6050 Sensor Review and Applications*. WSN, 167, 45–59.
- ylax. (2019). Automating Slide Presentations with VBA and PowerPoint Object Model. Retrieved from <https://stackoverflow.com>
- Yundra, H. (2020). Pemanfaatan Data Sensor untuk Aktivitas Manusia Menggunakan Machine Learning. *Jurnal Teknologi Informasi dan Komunikasi*, 4(2), 81–88.
- Zhang, H., Zhang, Y., et al. (2023). Vision-Based Gesture Recognition in Low-Light Environments. *Computer Vision and Image Understanding*, 227, 103661.
- Zhang, M., Yang, S., et al. (2021). Tuning MPU6050 for gesture recognition systems. *Sensors and Actuators A: Physical*, 319, 112593.

- Zhou, B., Liu, W., et al. (2019). Feature extraction strategies for IMU-based gesture recognition. *International Journal of Advanced Robotic Systems*, 16(1), 1–12.
- Zhou, H., Liu, J., et al. (2021). Design principles for wearable interaction systems. *ACM Computing Surveys*, 54(2), 1–35.
- Emlearn Contributors. (2020). emlearn: Machine Learning inference engine for microcontrollers and embedded systems [Computer software]. GitHub. <https://github.com/emlearn/emlearn>
- Tillaart, R. (2021). Statistic library for Arduino [Computer software]. GitHub. <https://github.com/RobTillaart/Statistic>