

Pengenalan Pola Dasar Angka berdasarkan Gerakan Tangan menggunakan *Machine Learning*

SYAFRIYADI NOR¹, MUHAMMAD AZIZ MUSLIM², MUHAMMAD ASWIN³

^{1,2,3}Departemen Teknik Elektro, Universitas Brawijaya, Indonesia

¹Jurusan Teknik Elektro, Politeknik Negeri Banjarmasin, Indonesia

Email : sapriyadi.yadi@gmail.com; syafriyadi@student.ub.ac.id

Received 4 Maret 2022 | Revised 23 Maret 2022 | Accepted 31 April 2022

ABSTRAK

Pengenalan gerakan tangan dianggap sebagai bagian penting dari interaksi manusia komputer, memungkinkan komputer untuk mengenali dan menafsirkan gerakan tangan dan menjalankan perintah. Penggunaan machine learning dimanfaatkan untuk mencari tren dan pola yang berbeda. Namun, tantangan untuk menerapkan machine learning menjadi bagaimana memilih di antara berbagai model berbeda digunakan untuk kumpulan data atau kasus berbeda. Tujuan dari penelitian ini adalah mengukur kinerja model machine learning yang diusulkan dengan pemilihan hyperparameter yang sesuai dalam mengenali 10 pola angka berdasarkan gerakan tangan di udara. Dalam makalah ini, model KNN, SVM, dan ANN-PSO diusulkan. Eksperimen dilakukan dengan mengumpulkan data gerakan yang berasal dari MPU-6050. Kinerja metode yang diusulkan dievaluasi menggunakan metrik standar seperti akurasi klasifikasi, presisi, recall, f1-score, dan AUC-ROC. Hasilnya menunjukkan bahwa akurasi rata-rata mencapai 87%.

Kata kunci: HCI, hand gesture recognition, machine learning, MPU-6050, pola

ABSTRACT

Hand gesture recognition is considered an essential part of human-computer interaction (HCI), enabling computers to recognize and interpret hand gestures and execute commands. The use of machine learning is utilized to look for different trends and patterns. However, the challenge for implementing machine learning becomes how to choose between different models used for different datasets or cases. This research aims to measure the performance of the proposed machine learning model by selecting the appropriate hyperparameters in recognizing 10 number patterns based on hand gestures in the air. In this paper, KNN, SVM, and ANN-PSO models are proposed. Experiments were carried by collecting gesture data from MPU-6050. The performance of the proposed method was evaluated using standard metrics such as classification accuracy, precision, recall, f1-score, and AUC-ROC. The results show that the average accuracy reaches 87%.

Keywords: HCI, hand gesture recognition, machine learning, MPU-6050, pattern

1. PENDAHULUAN

Gestur adalah gerakan tubuh yang ekspresif dan bermakna yang melibatkan gerakan fisik jari, tangan, lengan, kepala, wajah, atau tubuh dengan maksud menyampaikan informasi yang bermakna atau berinteraksi dengan lingkungan **(Anwar, dkk, 2019)**. Pengenalan gestur tangan memiliki banyak peran dalam komunikasi *non-verbal* dan interaksi komputer dengan manusia. Manusia cenderung secara alami menggunakan gerakan tangan dalam proses komunikasi untuk memperjelas maksud yang mereka ingin sampaikan, dan komputer mampu menangkap dan menafsirkan gerakan tangan, dan menjalankan perintah yang diberikan **(Yasen & Jusoh, 2019)**. Gerakan tangan dapat diklasifikasikan menjadi statis dan dinamis. Sesuai namanya, gerakan statis mengacu pada bentuk tangan yang tetap, sedangkan gerakan dinamis terdiri dari serangkaian gerakan tangan seperti melambaikan tangan **(Oudah, dkk, 2020)**. Sistem pengenalan gerakan (*gesture recognition*) memiliki dua proses : proses pertama yaitu akuisisi, yang mengubah gerakan fisik menjadi data numerik, dan proses kedua interpretasi, yang memberikan makna rangkaian simbol yang berasal dari proses keputusan **(Alleverd, dkk, 2006)**. Pengenalan gerakan dapat menciptakan sebuah sistem, yang dapat mengidentifikasi gerakan manusia tertentu dan menggunakannya untuk menyampaikan informasi atau untuk kontrol perangkat **(Trigueiros, dkk, 2012)**.

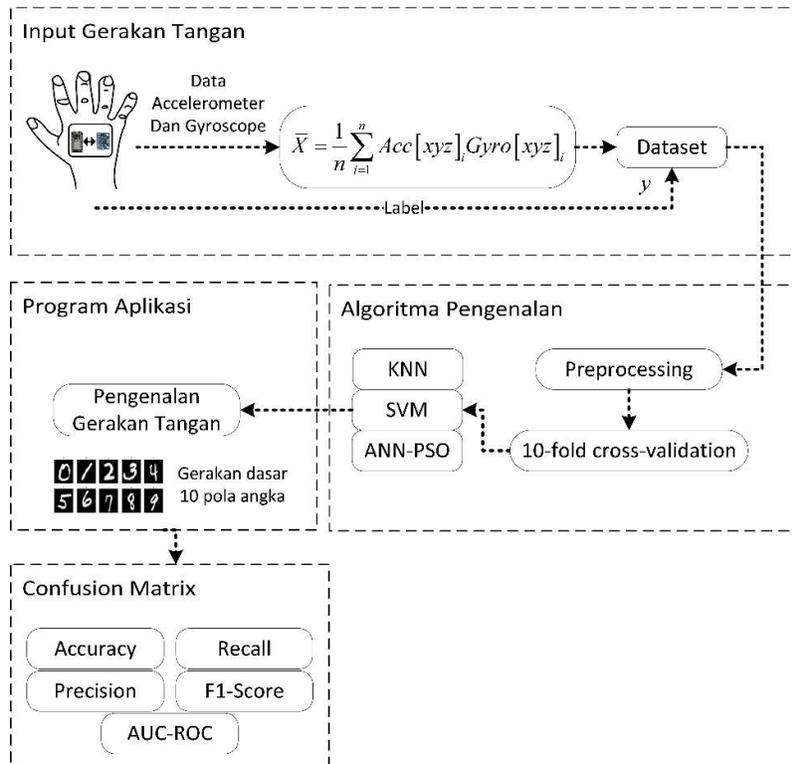
Human Computer Interaction (HCI) digunakan untuk interaksi antara manusia dan mesin. HCI umumnya menggunakan perangkat genggam seperti *mouse, keyboard, layer*, dan pemindai. Perangkat ini umumnya perlu dihubungkan ke komputer Namun demikian, dalam beberapa skenario seperti *remote command*, masih tidak nyaman untuk menggunakan perangkat yang disentuh oleh tangan ini. Dengan kemajuan komputer, pendekatan interaksi pengguna juga telah maju. Penggunaan tangan secara langsung sebagai perangkat *input* adalah metode yang menarik untuk menyediakan HCI yang alami. Sistem pengenalan gerakan tangan memiliki teknik berbasis *computer-vision* dan berbasis sensor yang dapat dikenakan. Terdapat penelitian pengenalan *hand gesture* pada Sistem Isyarat Bahasa Indonesia (SIBI) menggunakan *leap sensor* untuk membaca gerakan tangan dan jari. Namun, Gerakan pada pola angka terbatas pada *static gesture* dengan angka 5, klasifikasi yang digunakan adalah algoritma *Naive Bayes* dan mencapai akurasi 73.33%. Namun, Penggunaan sensor *leap motion* yang digunakan tergantung pada intensitas cahaya, karena dapat mengganggu pembacaan sensor infrared dan mempengaruhi akurasi **(Pramunanto, dkk, 2017)**. Terdapat penelitian yang sama pada Sistem Isyarat Bahasa Indonesia (SIBI) **(Rosalina, dkk, 2018)**, yaitu pengenalan huruf alfabet 'A'-'Z' dan angka 0-9 dengan *static gesture*. Metode yang dilakukan dengan mengambil gambar menggunakan kamera tunggal baik *eksternal* atau *built-in* untuk menangkap gambar secara *real-time* dengan deteksi tangan orang yang melakukan gerakan. Klasifikasi menggunakan Jaringan Syaraf Tiruan dengan total akurasi 90%. Namun, teknik berbasis *computer-vision* memerlukan pencahayaan yang konstan dan kompleks **(Zhang, dkk, 2019)** dan **(Gupta & Kumar, 2021)**, dan penempatan kamera yang sesuai. Teknik ini juga mahal untuk diimplementasikan **(Khomami & Shamekhi, 2021)**, **(Wu, dkk, 2016)**. Dalam penelitian ini memilih teknik berbasis sensor MPU-6050 yang terdiri dari *accelerometer* 3-sumbu, dan 3-sumbu *gyroscope* untuk mengukur percepatan gerakan, dan kecepatan sudut **(Abdullah, dkk, 2020)**. Selain itu, sensor IMU dipilih karena biaya yang murah, konsumsi daya rendah, dan dapat digunakan untuk melacak gerakan tangan secara keseluruhan **(Khomami & Shamekhi, 2021)** **(Filippeschi, dkk, 2017)**.

Algoritma *machine learning* dapat membantu mengklasifikasikan gerakan yang berbeda **(Krishnan, dkk, 2018)**. Sejumlah penelitian juga telah menunjukkan hasil yang baik dalam mengenali gerakan tangan menggunakan sensor IMU dengan bantuan *machine learning*. Namun, tidak semua algoritma *machine learning* memiliki kinerja yang baik dalam kumpulan

data atau kasus yang berbeda. Dalam penelitian ini pengukuran kinerja model *machine learning* dalam mengenali pola gerakan tangan di udara menggunakan sensor IMU dilakukan, dengan menyajikan hasil dalam bentuk matrik dan grafik dari model KNN, SVM, dan ANN-PSO yang diusulkan dengan pemilihan *hyperparameter*-nya. Pola yang digunakan sebagai masukan adalah sepuluh angka dasar [0-9] yang umum diketahui oleh banyak orang. Seluruh proses pengenalan gerakan tangan dibagi menjadi tiga fase: deteksi tangan, pelacakan tangan, dan pengenalan. Bagian ini mencakup ulasan tentang berbagai metode yang digunakan untuk pengenalan gerakan tangan. Fase pengenalan diklasifikasikan berdasarkan pengumpulan data fitur yang direkam menggunakan MPU-6050. Kinerja dari model dievaluasi ke dalam bentuk *confusion matrix* dan *AUC-ROC*.

2. METODE PENELITIAN

Gambar 1 menunjukkan diagram blok dari sistem pengenalan pola angka yang diusulkan. Perangkat yang digunakan dalam pengambilan data gerakan terdiri dari sensor MPU-6050 (3-axis accelerometer (*Acc*), dan 3-axis gyroscope (*Gyro*)) dan ESP32 sebagai mikrokontroler. Perangkat digunakan untuk merekam gerakan tangan yang membentuk pola angka dasar.



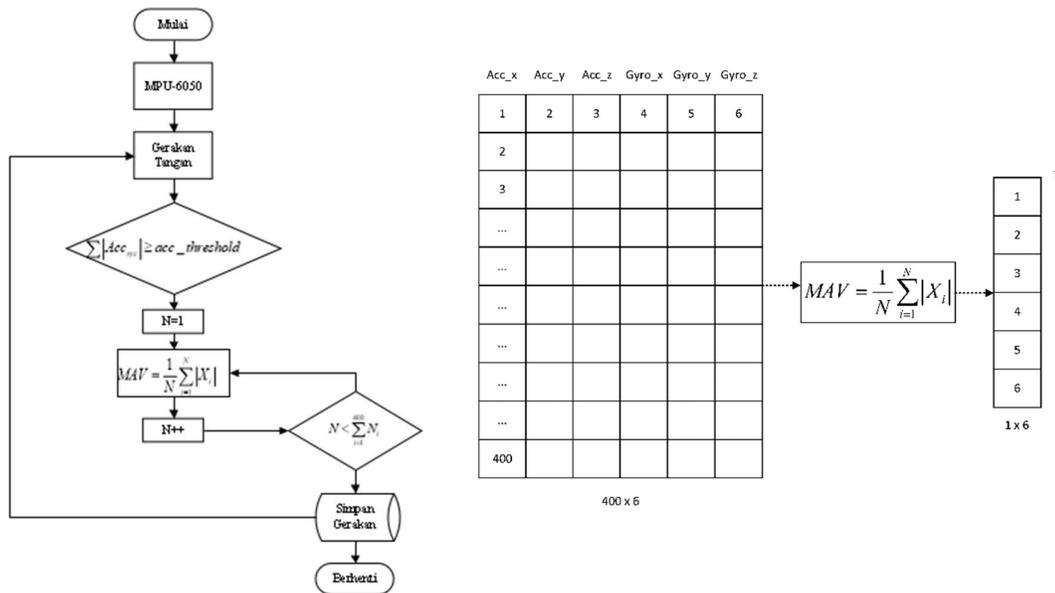
Gambar 1. Diagram Blok dari Sistem Pengenalan Pola Dasar Angka

Dalam proses merekam gerakan tangan. Pertama adalah dengan menempatkan perangkat pada bagian tangan. Kemudian *gyroscope* dan *accelerometer* sepanjang tiga sumbu (x, y, z) yang berasal dari sensor IMU digunakan untuk membuat pola gerakan angka. Untuk itu, jumlah rekaman yang tetap (jumlah sampel) mulai dari deteksi gerakan pada *star point* sampai *end point* akan menggunakan informasi dari sensor IMU. Ini berarti vektor fitur yang digunakan akan berdimensi $6 \times$ jumlah sampel, yang bisa menjadi terlalu besar untuk dimuat di memori ESP-32. Dalam penelitian ini, panjang data gerakan (jumlah sampel) ditentukan

sebanyak 400 sampel, dengan waktu tiap sampelnya adalah $t=0.3 \mu s$. Untuk mereduksi dimensi data gerakan dari keluaran MPU-6050, maka menggunakan persamaan *Mean Absolute Value* (MAV), yang ditunjukkan pada Persamaan (1) dan (2). MAV adalah estimasi dari penjumlahan nilai absolut dari pembacaan MPU-6050, di mana N adalah jumlah sampel.

$$MAV(Acc) = \frac{1}{N} \sum_{i=1}^N |A_i| \quad (1)$$

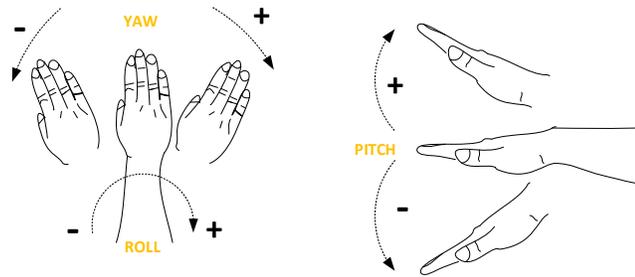
$$MAV(Gyro) = \frac{1}{N} \sum_{i=1}^N |G_i| \quad (2)$$



Gambar 2. Deteksi Gerakan dan Pengolahan Data Mentah dari MPU-6050

Dalam penelitian ini, titik awal dan akhir gerakan terdeteksi secara otomatis oleh perubahan *accelerometer*. Dimana batas percepatan yang ditentukan adalah sebesar 25g dari penjumlahan nilai *accelerometer* sepanjang sumbu (x, y, z) yang nilainya diabsolutkan. Kemudian Acc dan Gyro diperoleh untuk proses perekaman sebanyak 400 sampel untuk sekali gerakan. Data yang disimpan merupakan hasil rata-rata dari Acc dan Gyro sepanjang sumbu (x, y, z) dari Persamaan (1) dan (2). Untuk melakukan gerakan dengan pola dasar angka, orientasi gerakan tangan menyesuaikan dengan arah sumbu rotasi pada MPU-6050.

Enam fitur dipilih dari IMU. Pada Gambar 2, Acc_x, Acc_y, dan Acc_z masing-masing mewakili pembacaan *accelerometer* sepanjang sumbu x, sumbu y, dan sumbu z. Demikian pula, Gyro_x, Gyro_y, dan Gyro_z masing-masing adalah pembacaan *gyroscope* sepanjang sumbu x, sumbu y, dan sumbu z.



Gambar 3. Ilustrasi Orientasi Gerakan Tangan (Fernandez, dkk, 2016)

Pada Gambar 3 orientasi gerakan tangan meliputi sumbu X (*ROLL*), Y (*PITCH*), dan Z (*YAW*). Untuk memulai suatu gerakan akan ditentukan nilai batas dari *accelerometer* sepanjang sumbu (x, y, z)

Tahap *preprocessing* pada Gambar 1 meliputi pemilihan fitur yang datanya berasal dari MPU-6050, yang memiliki pengukuran *gyroscope* dan *accelerometer*, penggunaannya dapat mengukur rotasi dan percepatan di sepanjang sumbu (x, y, z). Kombinasi kedua pengukuran tersebut digunakan untuk mendapatkan informasi yang akurat tentang orientasi suatu gerakan. Normalisasi data fitur diperlukan dalam tahap ini. Tujuan utama dari normalisasi adalah untuk membuat data fitur homogen secara keseluruhan, ini untuk membantu dalam penskalaan dan meningkatkan kualitas data. Persamaan (3) digunakan untuk normalisasi data fitur dengan *Min-Max scaling*. Sehingga rentang data fitur berkisar antara [0-1].

$$\bar{X} = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (3)$$

Untuk memastikan bahwa model *machine learning* mengenali pola yang akurat dari data baru, dan memiliki bias yang rendah. Oleh karena itu, dalam penelitian ini menggunakan teknik *k-fold cross-validation*. Pemilihan nilai k=10 dipilih menjadi *10-fold cross-validation* yang sama diterapkan pada (Khomami & Shamekhi, 2021).

Algoritma *machine learning* diterapkan untuk mengenali pola gerakan angka setelah tahap *preprocessing*, kinerja model *machine learning* terhadap fitur (Acc, dan Gyro) menjadi masukan pada tiga algoritma *machine learning* yang diusulkan. Yaitu; *k-nearest neighbour*, *support-vector machine*, dan *artificial neural network* yang dioptimasi dengan pemilihan bobot dan bias dengan algoritma *particle swarm optimization*.

Evaluasi terhadap kinerja model *machine learning* dalam penelitian ini menggunakan teknik *confusion matrix*. Jumlah prediksi yang benar dan salah dirangkum dengan nilai hitungan dan dipecah oleh masing-masing kelas. Dalam *confusion matrix* untuk mengukur akurasi klasifikasi, *precision*, *recall*, dan *F1-score* dengan menggunakan Persamaan (4) sampai (7).

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

$$Precision = \frac{TP}{TP+} \quad (5)$$

$$Recall = \frac{TP}{TP+} \quad (6)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recal} \quad (7)$$

$TP = True\ Positive$

$TN = True\ Negatif$

$FP = False\ Positive$

$FN = False\ Negative$

Kurva ROC (*Receiver Operating Characteristic*) biasanya digunakan dalam klasifikasi biner. Untuk memperluas kurva ROC dan area ROC ke klasifikasi multi-label, *output* perlu dibinerisasi. Penggunaan *micro* dan *macro averaging* untuk mengevaluasi semua kinerja semua label. Di dalam *micro-averaging* perlu dihitung nilai *precision* dari setiap kelas secara individu berupa TP, TN, FP, dan FN dari k-label.

$$Precision_{micro} = \frac{TP_1 + \dots + TP_k}{TP_1 + \dots + TN_k + FP_1 + \dots + FP_k} \quad (8)$$

Dan di dalam *macro-averaging* dihitung nilai rata-rata dari setiap label

$$Precision_{macro} = \frac{Precision_1 + \dots + Precision_k}{k} \quad (9)$$

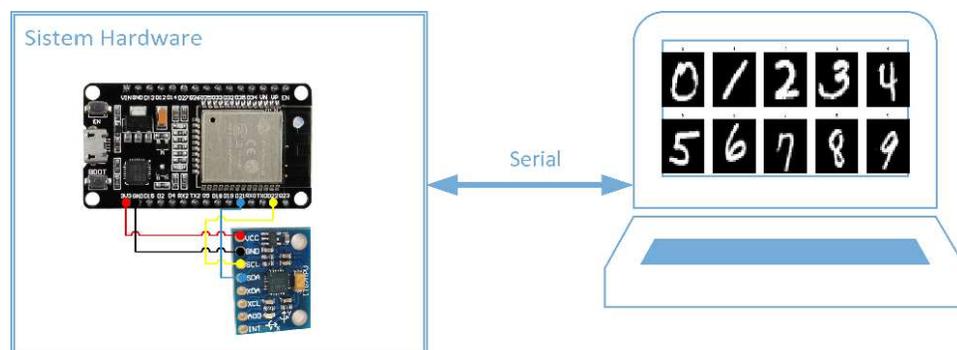
2.1 Deskripsi Perangkat keras

a. ESP-32

ESP-32 adalah *chip* tunggal yang memiliki fitur Wi-Fi dan *Bluetooth* 2,4 GHz yang dirancang dengan teknologi 40 nm *ultra-low-power* TSMC (**Espressif, 2019**). ESP32 kaya akan fitur dengan Wi-Fi terintegrasi dan konektivitas *bluetooth* untuk jangkauan aplikasi yang luas. ESP32 dirancang untuk aplikasi seluler, perangkat elektronik yang dapat dikenakan, dan *Internet-of-Things* (IoT).

b. MPU-6050

MPU-6050 adalah sensor IMU dengan *6-Degree of freedom* (DOF) yang terdiri dari tiga sumbu *gyroscope*, dan tiga sumbu *accelerometer*. Dalam penelitian ini, rentang pengukuran *accelerometer* dan *gyroscope* diatur ke $\pm 2g$ dan ± 250 yang cukup untuk menangkap gerakan yang terkait dengan gerakan pola angka. MPU-6050 menggunakan protokol serial I2C untuk berkomunikasi dengan mikrokontroler ESP-32 melalui *library* MPU-6050 untuk membaca data *accelerometer* dan *gyroscope*.



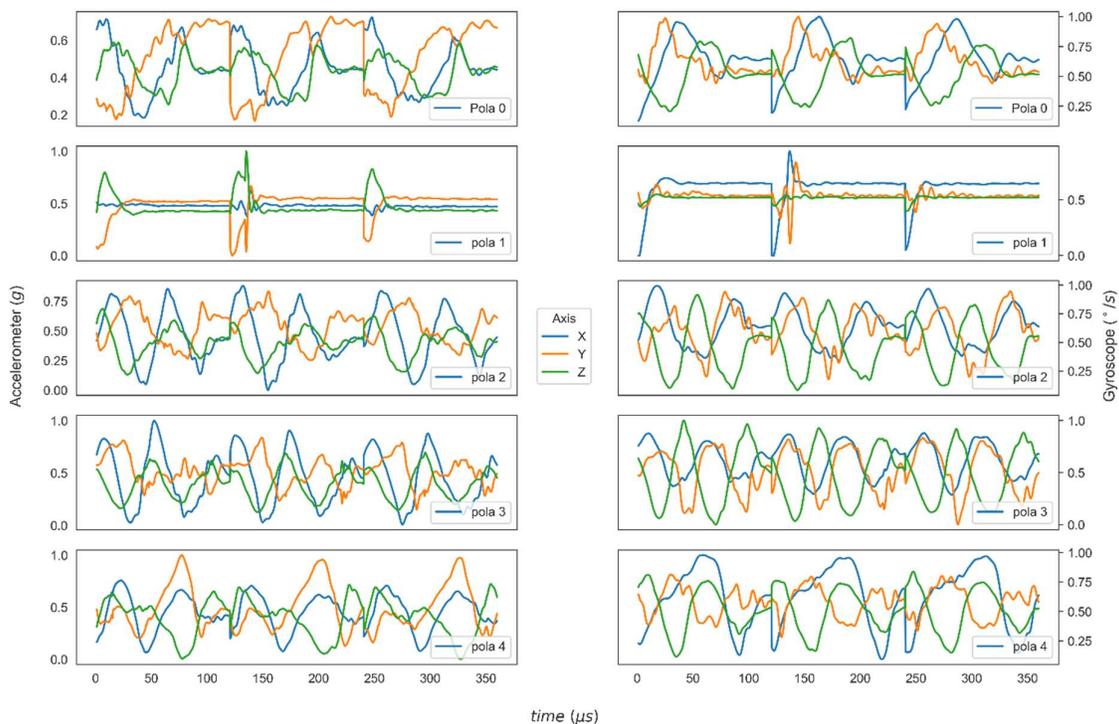
Gambar 4. Perangkat Keras yang Digunakan untuk Akuisisi Data Gerakan

2.2 Pengumpulan dan Pengolahan Data Gerakan

Secara total, ada sepuluh pola gerakan angka yang umum digunakan dalam pengenalan angka 0-9 dipilih dalam penelitian ini. Setiap subjek duduk di kursi atau berdiri di depan monitor laptop yang memiliki antarmuka dalam perekaman gerakan yang diletakkan di atas meja. Perangkat ditempatkan di tangan kanan menggunakan sarung tangan yang terhubung melalui *port* USB laptop. Subjek diminta untuk mengulangi gerakan pola angka 0-9. Secara total, ada 100 gerakan dengan setiap satu pola gerakan sebanyak sepuluh kali, dan sepuluh sukarelawan/subjek dan tidak kidal direkrut untuk pengumpulan data. Secara keseluruhan deskripsi dataset terdapat pada Tabel 1 yang terdiri dari enam kolom fitur dan satu kolom label dengan total 1000 pola gerakan.

Tabel 1. Deskripsi Dataset Gerakan Tangan

Kelas	10
Sampel per kelas	100
Total sampel	1000
Dimensi	6
Fitur	P, positif



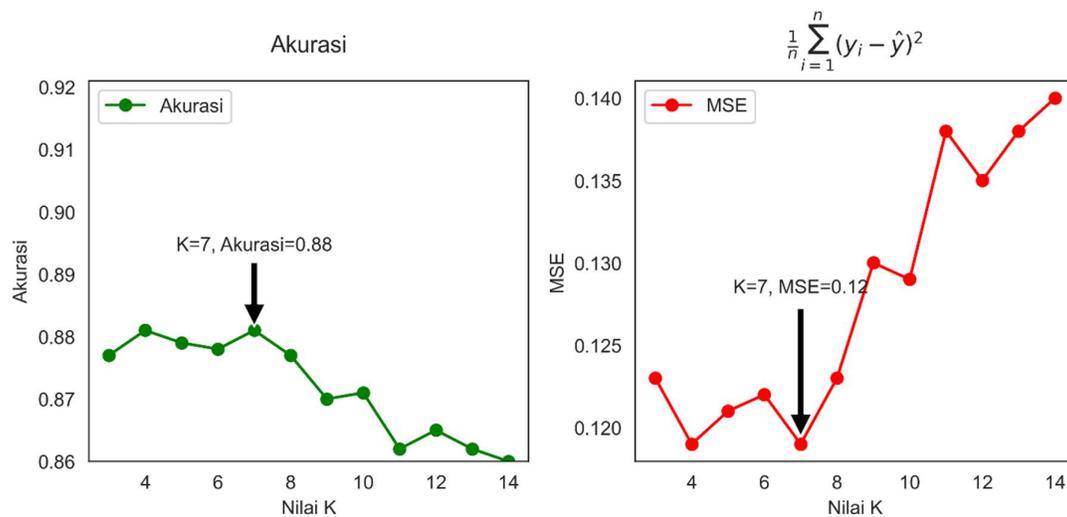
Gambar 5. Sinyal Accelerometer, dan Gyroscope dari Pola "nol", "satu", "dua", "tiga", dan "empat" dengan Tiga Pengulangan yang Direkam pada Saat Akuisisi Data Gerakan

2.3 Pemilihan *Hyperparameter*

Untuk menyesuaikan model *machine learning* ke dalam masalah yang berbeda, *hyperparameter* harus disetel. Memilih konfigurasi *hyperparameter* terbaik untuk model *machine learning* memiliki dampak langsung pada performa model (Yang & Shami, 2020). Jika penyetelan *hyperparameter* yang tepat dari pengklasifikasi model *machine learning*

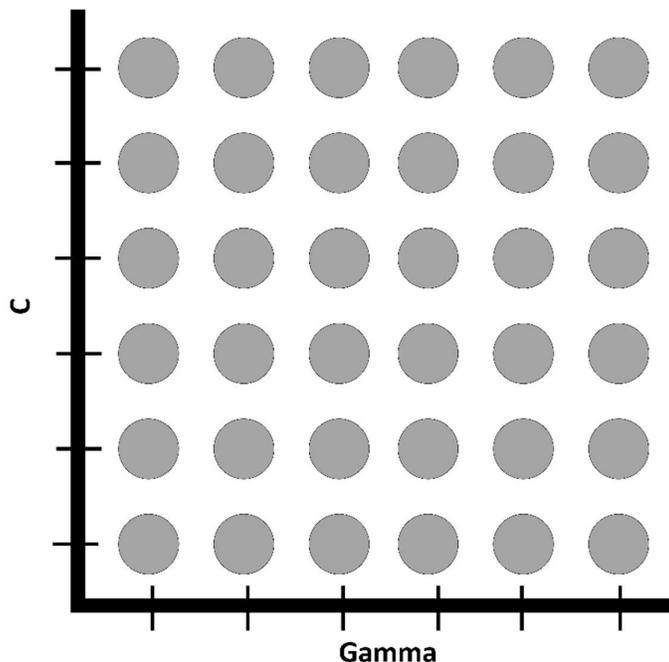
dilakukan, secara signifikan akurasi yang lebih tinggi dapat diperoleh (**Elgeldawi, dkk, 2021**). Dalam penelitian ini, model SVM, KNN, dan ANN-PSO diusulkan. Pada model KNN jumlah tetangga terdekat k diatur ke rentang [3-15]. Cara ini dilakukan karena algoritma KNN tidak memiliki proses *training* data yang membutuhkan penyesuaian bobot maupun bias. Sedangkan pada model SVM, fungsi kernel RBF (*Radial Basis Function*) dipilih untuk SVM dan *hyperparameter* C dan γ terbaik disetel menggunakan teknik *gridsearch*. Sedangkan pada ANN metode pembelajaran menggunakan algoritma *Particle Swarm Optimization*. Algoritma yang sering digunakan untuk melatih ANN adalah *Backpropagation* bersama dengan *Gradient Descent*. Namun, dapat juga menggunakan algoritma pengoptimalan yang lain. seperti melatih model ANN menggunakan *Particle Swarm Optimization*. Teknik ini dipilih karena dibandingkan dengan GA (*Genetic Algorithm*), lebih mudah untuk mengimplementasikan PSO, karena PSO tidak memiliki operasi tambahan tertentu seperti *crossover* dan mutasi (**Yang & Shami, 2020**).

Pada algoritma KNN, nilai k ditentukan untuk menentukan data uji masuk ke dalam kelas mana yang memiliki jarak terdekat dengan kumpulan data latih. Berdasarkan hasil pengujian menggunakan iterasi nilai k dengan rentang 3-15 ke dalam sebuah model KNN, kemudian mengembalikan nilai akurasi rata-rata berdasarkan teknik *cross validation*. Pada Gambar 6 hasil dengan $k=7$ memiliki rata-rata akurasi sebesar 88% dengan nilai MSE 0.12.



Gambar 6. Pencarian Nilai K [3-15] pada Model KNN

Pada Gambar 7, pemilihan *hyperparameter* C dan γ pada model SVM menggunakan *gridsearch*. Dalam metode ini, penentuan nilai grid untuk *hyperparameter* C dan γ [0.001, 0.01, 0.1, 1, 10, 100], beserta dengan penentuan fungsi kernel adalah dengan RBF, kemudian model dievaluasi untuk setiap kombinasi. Himpunan *hyperparameter* yang memberikan akurasi tertinggi dianggap sebagai yang terbaik.



Gambar 7. Metode Pencarian *Hyperparameter* [C dan Gamma] pada Model SVM menggunakan *Gridsearch*

Dalam makalah ini, pengaturan parameter ANN dan PSO ditunjukkan pada Tabel 2 dan 3:

Tabel 2. Parameter ANN

Input Layer	Bias	Hidden Layer	Output Layer
6	2	20	10

Tabel 3. Parameter *Particle Swarm Optimization*

w	C1	C2	Dimensi
0.9	0.5	0.5	163

3. HASIL DAN PEMBAHASAN

Untuk menunjukkan hasil metode yang diusulkan, sejumlah percobaan dilakukan dan hasilnya disediakan menggunakan beberapa grafik dan metrik standar seperti akurasi klasifikasi, presisi, *recall*, dan *F1-score*. Seperti yang ditunjukkan pada Tabel 4. Kami memeriksa persentasenya dari setiap gerakan tangan yang dikenali dengan benar oleh model KNN, SVM, dan ANN-PSO menggunakan *confusion matrix*, kami mendapat hasil klasifikasi yang cukup baik pada setiap gerakan, kecuali pada pola gerakan "Empat" dan "Lima" pada model KNN. Gambar 8 memperlihatkan bahwa pola gerakan "Empat" memiliki kesalahan prediksi sebesar 20%, dan pola yang sering diprediksi adalah pola gerakan "Lima", begitu juga dengan pola gerakan "Lima" yang memiliki kesalahan prediksi sebesar 21%, pola yang paling banyak diprediksi salah adalah pola gerakan "Delapan". Kedua pola gerakan ini sangat mempengaruhi akurasi rata-rata klasifikasi. Ini karena kedua pola gerakan ini memiliki gerakan yang tidak menyambung dan titik awal dan akhir gerakan yang tidak konsisten.

Nor, dkk

Aktual	Noi	89 8.90%	0 0.0%	1 0.10%	0 0.0%	0 0.0%	0 0.0%	7 0.70%	0 0.0%	0 0.0%	3 0.30%	100 10.00%
	Satu	0 0.0%	100 10.00%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 10.00%
	Dua	0 0.0%	0 0.0%	83 8.30%	11 1.10%	0 0.0%	1 0.10%	1 0.10%	3 0.30%	1 0.10%	0 0.0%	100 10.00%
	Tiga	0 0.0%	0 0.0%	6 0.60%	88 8.80%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	2 0.20%	3 0.30%	100 10.00%
	Empat	1 0.10%	0 0.0%	0 0.0%	0 0.0%	80 8.00%	13 1.30%	3 0.30%	0 0.0%	0 0.0%	3 0.30%	100 10.00%
	Lima	0 0.0%	1 0.10%	0 0.0%	1 0.10%	5 0.50%	79 7.90%	0 0.0%	0 0.0%	13 1.30%	1 0.10%	100 10.00%
	Enam	2 0.20%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	95 9.50%	0 0.0%	0 0.0%	2 0.20%	100 10.00%
	Tujuh	2 0.20%	0 0.0%	0 0.0%	1 0.10%	0 0.0%	1 0.10%	1 0.10%	93 9.30%	0 0.0%	2 0.20%	100 10.00%
	Delapan	0 0.0%	0 0.0%	0 0.0%	3 0.30%	3 0.30%	11 1.10%	0 0.0%	0 0.0%	83 8.30%	0 0.0%	100 10.00%
	Sembilan	4 0.40%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	2 0.20%	1 0.10%	0 0.0%	92 9.20%	100 10.00%
$\sum_{i=1}^{10}$ Prediksi	98 9.80%	101 10.10%	90 9.00%	104 10.40%	88 8.80%	107 10.70%	109 10.90%	98 9.80%	99 9.90%	106 10.60%	1000 100.00%	
	Noi	Satu	Dua	Tiga	Empat	Lima	Enam	Tujuh	Delapan	Sembilan	$\sum_{i=1}^{10}$ Aktual	
	Prediksi											

Gambar 8. Tabel *Confusion Matrix* dari Model KNN (k=7)

Aktual	Noi	94 9.40%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 0.40%	1 0.10%	0 0.0%	1 0.10%	100 10.00%
	Satu	0 0.0%	100 10.00%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 10.00%
	Dua	0 0.0%	0 0.0%	91 9.10%	3 0.30%	0 0.0%	2 0.20%	0 0.0%	2 0.20%	2 0.20%	0 0.0%	100 10.00%
	Tiga	0 0.0%	0 0.0%	8 0.80%	88 8.80%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	2 0.20%	1 0.10%	100 10.00%
	Empat	0 0.0%	0 0.0%	0 0.0%	0 0.0%	88 8.80%	4 0.40%	2 0.20%	1 0.10%	3 0.30%	2 0.20%	100 10.00%
	Lima	0 0.0%	1 0.10%	0 0.0%	0 0.0%	2 0.20%	81 8.10%	0 0.0%	0 0.0%	16 1.60%	0 0.0%	100 10.00%
	Enam	2 0.20%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	95 9.50%	0 0.0%	0 0.0%	2 0.20%	100 10.00%
	Tujuh	1 0.10%	0 0.0%	1 0.10%	2 0.20%	1 0.10%	0 0.0%	0 0.0%	93 9.30%	0 0.0%	2 0.20%	100 10.00%
	Delapan	0 0.0%	0 0.0%	0 0.0%	3 0.30%	3 0.30%	18 1.80%	0 0.0%	0 0.0%	76 7.60%	0 0.0%	100 10.00%
	Sembilan	2 0.20%	0 0.0%	0 0.0%	0 0.0%	1 0.10%	1 0.10%	3 0.30%	2 0.20%	0 0.0%	91 9.10%	100 10.00%
$\sum_{i=1}^{10}$ Prediksi	99 9.90%	101 10.10%	100 10.00%	96 9.60%	95 9.50%	107 10.70%	104 10.40%	100 10.00%	99 9.90%	99 9.90%	1000 100.00%	
	Noi	Satu	Dua	Tiga	Empat	Lima	Enam	Tujuh	Delapan	Sembilan	$\sum_{i=1}^{10}$ Aktual	
	Prediksi											

Gambar 9. Tabel *Confusion Matrix* dari Model SVM (Fungsi Kernel RBF, C=10, Gamma=10)

Pengenalan Pola Dasar Angka berdasarkan Gerakan Tangan menggunakan *Machine Learning*



Gambar 10. Tabel *Confusion Matrix* dari Model ANN-PSO (Tabel 2 dan 3)

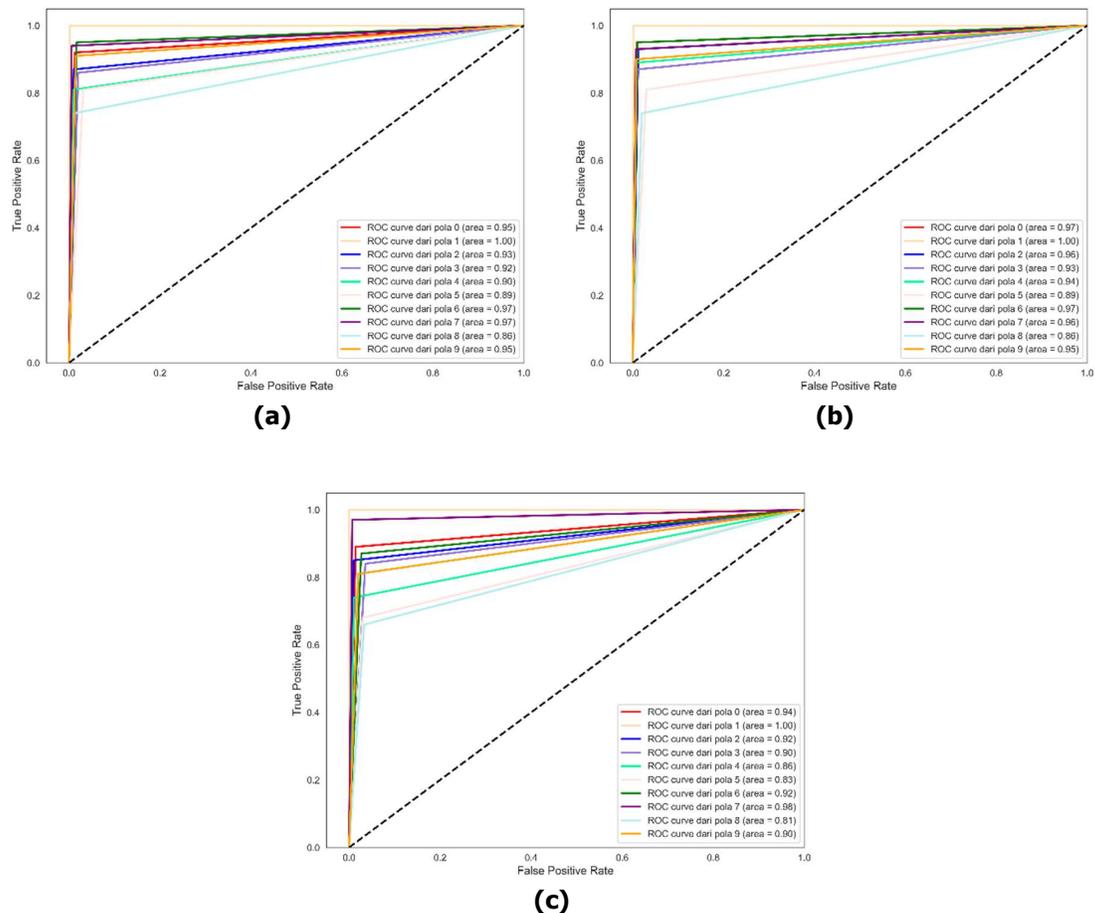
Tabel 4. Hasil Eksperimen dari Metode yang Diusulkan dalam Mengukur Akurasi, *Precision*, *Recall* dan *F1-score*

Pola	KNN			SVM			ANN-PSO		
	precision	Recall	f1-score	precision	Recall	f1-score	precision	Re-call	f1-score
Nol	0,91	0,89	0,90	0,95	0,94	0,94	0,87	0,89	0,88
Satu	0,99	1	1	0,99	1	1	1	1	1
Dua	0,92	0,83	0,87	0,91	0,91	0,91	0,9	0,85	0,88
Tiga	0,85	0,88	0,86	0,92	0,88	0,9	0,72	0,84	0,78
Empat	0,91	0,80	0,85	0,93	0,88	0,9	0,88	0,74	0,8
Lima	0,74	0,79	0,76	0,76	0,81	0,78	0,72	0,68	0,7
Enam	0,87	0,95	0,91	0,91	0,95	0,93	0,78	0,87	0,82
Tujuh	0,95	0,93	0,94	0,93	0,93	0,93	0,94	0,97	0,96
Delapan	0,84	0,83	0,83	0,77	0,76	0,76	0,69	0,66	0,67
Sembilan	0,87	0,92	0,89	0,92	0,91	0,91	0,82	0,81	0,81
Akurasi	0,88			0,89			0,83		

Pada Gambar 9, pola gerakan “Delapan” memiliki kesalahan prediksi 24%, karena pola ini paling banyak memprediksi pola “Lima” dari pola lainnya. Sebaliknya pada pola gerakan “Satu” memiliki tingkat akurasi mencapai 100%, ini dikarenakan gerakan yang hanya naik atau turun secara vertikal lebih mudah untuk dilakukan. Tidak seperti model KNN dan SVM yang memiliki akurasi yang hampir sama, model ANN-PSO memiliki akurasi hanya mencapai 83%. Gambar

10 memperlihatkan pola "Satu" sama dengan model KNN dan SVM yang memiliki pengenalan pola angka mencapai 100%. Namun pengenalan pola "Lima" dan "Delapan" kesalahan prediksi lebih besar mencapai lebih dari 30%. Sehingga dari ketiga model *machine learning* yang digunakan, bahwa ketiganya memiliki kinerja yang baik dalam mengenali pola gerakan tangan di udara. Dan dari ketiganya pula, pola gerakan "Lima" dan "Delapan" sama-sama memiliki tingkat pengenalan pola yang rendah.

Tabel 4 menggambarkan perbandingan kinerja di antara berbagai model klasifikasi seperti KNN, SVM, dan ANN dengan metode mengoptimasi bobot dan bias menggunakan algoritma PSO. Untuk memperkirakan hasil metode yang diusulkan, analisis komparatif diperlukan. Selain itu, menarik untuk menganalisis hasil semua model dalam hal AUC-ROC pada Gambar 11(a) hingga Gambar 11(c). Dibandingkan dengan semua model yang diusulkan. Baik nilai kurva ROC rata-rata mikro dan makro adalah 87,3 dalam model *machine learning* yang diusulkan dan memiliki kinerja yang baik dalam mengklasifikasi pola angka berdasarkan gerakan tangan di udara.



Gambar 11. Analisis AUC-ROC untuk (a) KNN, (b) SVM, dan (c) ANN-PSO

Dari hasil pengukuran dengan AUC-ROC pada Gambar 11, menunjukkan bahwa baik KNN, SVM, dan ANN-PSO layak untuk penelitian lebih lanjut tentang masalah pengenalan pola gerakan angka dasar ini. Algoritma *machine learning* adalah *tool* yang ampuh untuk berbagai aplikasi dan banyak digunakan dalam beberapa penelitian sebelumnya dalam aplikasi berbasis komputer. Namun, penting untuk menggunakannya dengan cara yang benar. Untuk aplikasi

yang berbeda, fitur yang berbeda dan pengklasifikasi yang berbeda mungkin memiliki kinerja yang berbeda secara signifikan. Disarankan untuk mencoba pendekatan yang berbeda untuk menentukan model yang terbaik. Poin lainnya adalah bahwa *hyperparameter* harus disetel dengan baik untuk mendapatkan model yang optimum. Perlu dicatat bahwa, terkadang, nilai *default* bekerja sangat baik untuk beberapa model *machine learning* dan *dataset* yang ditentukan, namun hal ini tidak selalu terjadi saat mengubah model dan kumpulan data yang digunakan. Dalam pendekatan kami, jika kami hanya menggunakan *hyperparameter* secara *default* untuk model SVM, maka hanya akurasi 83% yang dapat dicapai. Selain itu, penyetelan pada *hyperparameter* yang berbeda pada model ANN-PSO yang tertera pada Tabel 2 dan 3 masih memungkinkan model dapat mencapai akurasi yang lebih tinggi lagi, dan memberikan hasil yang baik untuk kasus lain jika nilai parameter PSO ditetapkan dengan benar dan jumlah iterasi yang cukup.

4. KESIMPULAN

Dalam penelitian ini, kami menyajikan hasil pengukuran ke dalam bentuk *confusion matrix* dan *AUC-ROC* dari pengenalan gerakan tangan yang mampu mengenali pola gerakan tangan di udara. Kami mengevaluasi sistem pengenalan gerakan tangan dengan 10 pola angka dasar dengan menerapkan algoritma *machine learning* untuk mengklasifikasikan pola gerakan tersebut beserta pemilihan *hyperparameter*-nya. Sistem ini terdiri dari sensor MPU-6050 dan mikrokontroler ESP32 yang ditempatkan pada sarung tangan. KNN, SVM, dan ANN-PSO dipilih sebagai model pengklasifikasi. Hasilnya menunjukkan dan mengindikasikan bahwa pada setiap model yang diusulkan dalam mengenali pola angka berdasarkan gerakan tangan mampu diklasifikasi dengan baik.

DAFTAR RUJUKAN

- Allevard, T., Benoit, E., & Foulloy, L. (2006). Hand posture recognition with the fuzzy glove. *Modern Information Processing*, 417 - 427.
- Abdullah, A., Abdul-Kadir, N. A., & Che Harun, F. K. (2020). An Optimization of IMU Sensors-Based Approach for Malaysian Sign Language Recognition, *International Conference on Computing Engineering and Design (ICCED)*, (pp. 1 - 4).
- Anwar, S., Sinha, S. K., Vivek, S., & Ashank, V. (2019). Hand gesture recognition: A survey. *Lecture Notes in Electrical Engineering*, 511(3), 365–371.
- Elgeldawi, E., Sayed, A., Galal, A. R., dan Zaki, A. M. (2021). Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*, 8(4), 1–21.
- Espressif. (2021, Desember 23). *ESP32 Series Datasheet*. Retrieved from www.espressif.com.
- Fernandez, R. A. S., Sanchez-Lopez, J. L., Sampedro, C., Bavle, H., Molina, M., & Campoy, P. (2016). Natural user interfaces for human-drone multi-modal interaction. *International Conference on Unmanned Aircraft Systems (ICUAS)*, (pp. 1013–1022).
- Filippeschi, A., Schmitz, N., Miezal, M., Bleser, G., Ruffaldi, E., dan Stricker, D. (2017). Survey

- of motion tracking methods based on inertial sensors: A focus on upper limb human motion. *Sensors (Switzerland)*, 17(6), 1 - 40.
- Gupta, R., & Kumar, A. (2021). Indian sign language recognition using wearable sensors and multi-label classification. *Computers and Electrical Engineering*, 90(October 2020), 106898.
- Khomami, S. A., & Shamekhi, S. (2021). Persian sign language recognition using IMU and surface EMG sensors. *Measurement: Journal of the International Measurement Confederation*, 168(September 2020), 108471.
- Krishnan, K. S., Saha, A., Ramachandran, S., & Kumar, S. (2018). Recognition of human arm gestures using Myo armband for the game of hand cricket. *IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, (pp. 389 - 394).
- Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *Journal of Imaging*, 6(8), 1 - 29.
- Pramunanto, E., Sumpeno, S., & Legowo, R. S. (2017). Classification of hand gesture in Indonesian sign language system using naive bayes. *International Seminar on Sensor, Instrumentation, Measurement and Metrology: Innovation for the Advancement and Competitiveness of the Nation (ISSIMM)*, (pp. 187 - 191).
- Rosalina, Yusnita, L., Hadisukmana, N., Wahyu, R. B., Roestam, R., & Wahyu, Y. (2018). Implementation of real-time static hand gesture recognition using artificial neural network. *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, (pp. 1 - 6).
- Trigueiros, P., Ribeiro, F., & Reis, L. P. (2012). A comparison of machine learning algorithms applied to hand gesture recognition. *7th Iberian Conference on Information Systems and Technologies (CISTI)*, (pp. 1 - 6).
- Wu, J., Sun, L., & Jafari, R. (2016). A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors. *IEEE Journal of Biomedical and Health Informatics*, 20(5), 1281 - 1290.
- Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415, 295 - 316.
- Yasen, M., & Jusoh, S. (2019). A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*, 2019(9), 1 - 30.
- Zhang, X., Yang, Z., Chen, T., Chen, D., & Huang, M. C. (2019). Cooperative Sensing and Wearable Computing for Sequential Hand Gesture Recognition. *IEEE Sensors Journal*, 19(14), 5775 - 5783.