

Pengembangan Sistem Deteksi Hand Gesture untuk Mempermudah Menghafal Sandi Morse dengan Metode KNN

Development of Hand Gesture Detection System to Facilitate Memorizing Morse Code with KNN Method

Wahyuni^{1*}, Pitrasacha Adytia², Adha Trisna Lidya³

^{1,3}Teknik Informatika, STMIK Widya Cipta Dharma

²Sistem Informasi, STMIK Widya Cipta Dharma

¹wahyuni@wicida.ac.id, ²pitra@wicida.ac.id, ³adha@gmail.com

Abstract

Morse code is a unique communication technique that is still used in various contexts, such as emergency communication and amateur radio. Radio frequency controllers in Indonesia often face difficulties in memorizing Morse code. Current Morse code learning media are still limited to dots and lines that are difficult to memorize. This study develops a hand gesture detection system to facilitate memorization of Morse code. Hand gesture classification uses the K-Nearest Neighbors (KNN) method. This system utilizes hand movements such as clenching and opening the palm, to represent the combination of dots and lines in Morse code, with the hope of making the learning process more intuitive and interactive. The system implementation was carried out using a webcam, Mediapipe algorithm, OpenCV library, and Unity application. Then the model's performance was evaluated and its interface was tested with a black box. Total data was obtained from 50 people with 26 letters for each person, so that the total was 1300 data. The hand gesture detection system successfully identified letters of the alphabet based on hand movements with an accuracy of at least 60%. Further testing using KNN with a K-1 value showed an average accuracy of 81%. So the system is effective in detecting hand gestures to facilitate memorization of Morse code. Overall, with an average accuracy of 81%, this hand gesture detection system shows great potential in improving Morse code learning effectively and interestingly. The main obstacle in this study is the limited participant data, which results in variations in hand gestures and potential overlap between gesture classes. This study requires more data to improve accuracy and reduce errors in gesture detection. Thus, in further research, researchers are expected to increase the dataset used in hand gesture detection for Morse code.

Keywords: *Hand Gesture*, Morse Code, *Hand Gesture Detection*, KNN

Abstrak

Sandi morse adalah teknik komunikasi unik yang masih digunakan dalam berbagai konteks, seperti komunikasi darurat dan amatir radio. Pengendali frekuensi radio di Indonesia sering menghadapi kesulitan dalam menghafal sandi morse. Media pembelajaran sandi morse saat ini masih terbatas pada titik dan garis yang sulit untuk dihafalkan. Penelitian ini mengembangkan sistem deteksi *hand gesture* untuk mempermudah penghafalan sandi morse. Pengklasifikasian *hand gesture* menggunakan metode K-Nearest Neighbors (KNN). Sistem ini memanfaatkan gerakan tangan seperti mengepal dan membuka telapak tangan, untuk mewakili kombinasi titik dan garis dalam sandi morse, dengan harapan membuat proses belajar lebih intuitif dan interaktif. Implementasi sistem dilakukan dengan menggunakan webcam, algoritma Mediapipe, library OpenCV, dan aplikasi Unity. Kemudian model dievaluasi performanya dan serta antarmukanya diuji dengan blackbox. Data total diperoleh dari 50 orang dengan 26 huruf untuk setiap orangnya sehingga totalnya 1300 data. Sistem deteksi *hand gesture* berhasil mengidentifikasi huruf abjad berdasarkan gerakan tangan dengan akurasi minimal 60%. Pengujian lebih lanjut menggunakan KNN dengan nilai K-1, menunjukkan rata-rata akurasi sebesar 81%. Sehingga sistem efektif dalam mendeteksi gerakan tangan untuk mempermudah penghafalan sandi morse. Secara keseluruhan, dengan akurasi rata-rata 81%, sistem deteksi *hand gesture* ini menunjukkan potensi besar dalam meningkatkan pembelajaran sandi morse secara efektif dan menarik. Kendala utama dalam penelitian ini adalah terbatasnya data partisipan, yang mengakibatkan variasi dalam gerakan tangan dan potensi tumpang tindih antara kelas gerakan. Penelitian ini membutuhkan lebih banyak data untuk meningkatkan akurasi dan mengurangi kesalahan dalam deteksi gerakan. Sehingga, pada penelitian selanjutnya diharapkan peneliti memperbanyak dataset yang digunakan pada deteksi gerakan tangan untuk sandi morse.

Kata kunci: *Hand Gesture*, Sandi Morse, Deteksi *Hand Gesture*, KNN

1. Pendahuluan

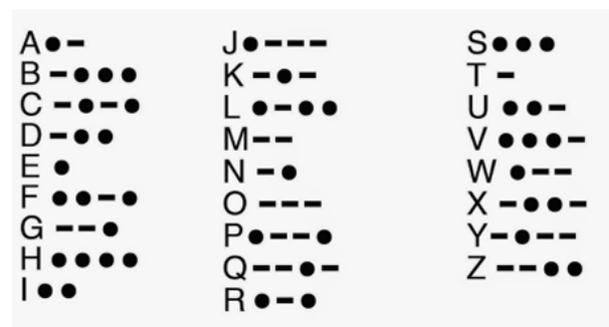
Sandi morse masih dipergunakan dalam berbagai konteks komunikasi, termasuk komunikasi darurat, navigasi kapal, dan amatir radio. Di Indonesia, sandi morse menjadi salah satu materi yang diujikan dalam Ujian Negara Amatir Radio (UNAR). Selain itu, penggunaan sandi morse sebagai media komunikasi juga sangat penting khususnya bagi pengendali frekuensi radio yang memiliki ruang lingkup tugas, tanggung-jawab, wewenang, dan hak untuk melakukan kegiatan di bidang pengendalian frekuensi radio. Seorang amatir radio dan pengendali frekuensi radio wajib menguasai sandi morse untuk dapat memberikan pertolongan saat dibutuhkan. Berdasarkan observasi yang telah dilakukan di Balai Monitor Spektrum Frekuensi Radio Kelas I Samarinda, mayoritas pengendali frekuensi radio mengalami kesulitan dalam menghafal sandi morse sehingga mempengaruhi performa kerja dalam kegiatan monitor yang melibatkan penggunaan sandi morse. Tantangan utama yang dihadapi individu dalam mempelajari, memahami dan menggunakan sandi morse adalah penghafalan kombinasi titik (dot) dan garis (dash) yang kompleks. Terlepas dari pentingnya sandi morse, pembelajaran yang ada masih terbatas berupa kombinasi titik dan garis berbentuk tabel yang cukup sulit untuk dihafalkan. Kombinasi titik (dot) dan garis (dash) yang digunakan dalam sandi ini memerlukan penghafalan yang cermat dan kadang membingungkan. Meskipun sebelumnya terdapat penelitian mengenai media interaktif sandi morse, penelitian tersebut masih menggunakan garis dan titik (simbol asli morse) sebagai bahan pembelajaran.

Pada penelitian [1] pengenalan sandi Morse menggunakan sensor taktil fleksibel berbasis material carbon nanotube (CNT) dan spons poliuretan (PUS) untuk mendeteksi input sandi Morse dari tekanan jari. Sensor ini dipasang pada sarung tangan dan dirancang untuk menangkap durasi dan intensitas tekanan (untuk membedakan titik dan garis). Data dari sensor kemudian diproses menggunakan model deep learning LSTM (Long Short-Term Memory) untuk mengenali pola-pola sandi Morse. Pada penelitian [2] bertujuan untuk mengklasifikasikan sinyal audio yang berisi sandi Morse menggunakan kombinasi CNN (Convolutional Neural Network) dan LSTM (Long Short-Term Memory). Penelitian ini cocok digunakan dalam sistem komunikasi alternatif, komunikasi darurat, dan aplikasi pendidikan untuk mengenalkan sandi Morse dengan pendekatan modern. Pada penelitian yang dilakukan Tsang dan Lu menunjukkan bahwa gerakan tangan atau mengunyah dapat meningkatkan daya ingat, karena gerakan tubuh mengaktifkan korteks frontal, yang sangat penting dalam proses ingatan [3]. Namun, bagaimana gerakan tangan memfasilitasi ingatan kata dalam aplikasi praktis dan apakah hal ini menghasilkan efek jangka panjang masih belum jelas. Penelitian ini

mengkaji dampak berbagai strategi terhadap daya ingat. Penelitian [4] memperkenalkan pendekatan inovatif untuk mendeteksi sandi Morse menggunakan kedipan mata, dirancang untuk memfasilitasi komunikasi non-verbal bagi individu dengan gangguan bicara. Sistem ini mengintegrasikan teknik penglihatan komputer dan machine learning untuk menginterpretasikan kedipan mata sukarela yang ditangkap melalui webcam sebagai sinyal sandi Morse, sambil membedakannya secara andal dari kedipan tidak disengaja.

Berdasarkan penelitian-penelitian yang telah diuraikan terdapat beberapa persamaan dalam penelitian, yaitu membuat sistem untuk belajar sandi morse dengan alat bantu atau simulasi untuk mempermudah pemahaman pengguna. Penelitian yang diusulkan saat ini memiliki perbedaan dengan penelitian terkait sebelumnya dalam hal pendekatan dalam mempelajari sandi morse. Penelitian terdahulu lebih banyak berfokus pada pengenalan sandi Morse dari sinyal audio, atau kedipan mata dan sentuhan menggunakan algoritma deep learning seperti CNN dan LSTM. Namun, penelitian ini menggunakan pendekatan yang secara eksplisit diarahkan untuk mempermudah proses pembelajaran atau penghafalan sandi Morse melalui gerakan tangan menggunakan algoritma yang ringan seperti KNN. Terdapat signifikansi bahwa gerakan tangan akan membantu peningkatan daya ingat dalam menghafal kosakata. Dengan memanfaatkan gerakan tangan, penelitian ini bermanfaat dalam meningkatkan kemampuan pengguna dalam menghafal sandi morse sehingga diharapkan dapat membantu pengendali frekuensi radio dalam menjalankan tugas yang berhubungan dengan sandi morse.

Sandi Morse menggunakan titik dan garis untuk membentuk suatu kalimat sesuai dengan ketentuan penggunaannya. Sandi morse biasanya dapat disampaikan melalui peluit, radio, asap, lampu, telegraf, serta arus listrik yang berfungsi untuk membedakan titik dengan strip. Sandi morse pada Gambar 1 yang sekarang kita kenal merupakan suatu ketetapan resmi dalam konferensi di Berlin tahun 1851[5].



Gambar 1. Sandi Morse [6]

OpenCV (Open Source Computer Vision Library) adalah pustaka open-source yang ditujukan untuk aplikasi computer vision dan machine learning. Pustaka

ini menyediakan lebih dari 2.500 algoritma yang efisien untuk pengolahan citra dan video, termasuk deteksi wajah, pengenalan objek, pelacakan gerakan, segmentasi, dan lainnya [7]

Unity merupakan Game engine yang dikembangkan oleh Unity Technologies. Unity merupakan alat bantu pengembang game dengan kemampuan rendering yang terintegrasi di dalamnya. Unity adalah sebuah bentuk teknologi terbaru yang meringankan dan memudahkan game pengembang membuat game [8]. Dengan menggunakan Unity sebagai antar muka dapat memberikan kemudahan interaksi dengan deteksi hand-gesture serta memantau gerakan tangan pada pengguna. Selain itu, Unity dapat digunakan untuk merekam data berupa titik-titik ruas jari yang kemudian disimpan di CSV yang nantinya digunakan untuk dataset deteksi menggunakan KNN.

Webcam adalah kependekan dari web camera yang merupakan perangkat kamera digital untuk dihubungkan ke komputer atau laptop. Dengan webcam, gambar Anda bisa tertangkap secara live kepada siapapun di berbagai penjuru dunia [9]. *Webcam* digunakan sebagai alat untuk menangkap gambar yang kemudian nantinya diproses menggunakan OpenCV untuk *image processing*. *Webcam* akan mengambil gambar tangan pengguna dari waktu ke waktu. Resolusi dan piksel *webcam* sendiri tidak terbatas karena semakin jelas gambar dan semakin kecil ukuran piksel akan membuat proses *image processing* semakin cepat.

Hand gesture recognition, merupakan teknologi yang mampu membaca gerak tangan kemudian dirubah menjadi teks dan atau suara. Gesture recognition merupakan topik dalam computer science dan language technology yang bertujuan agar komputer bisa memahami gerakan manusia yang umumnya berasal dari tangan atau wajah [10]. Selain itu, hand gesture juga dapat menggunakan webcam untuk mendeteksi landmark tangan dengan bantuan Open-CV dan algoritma mediapipe hand landmark. Mediapipe hand landmark merupakan algoritma deteksi landmark tangan yang terdiri dari 30.000 data gambar tangan pada dunia nyata dengan 21 koordinat dapat dilihat pada Gambar 2 [11].

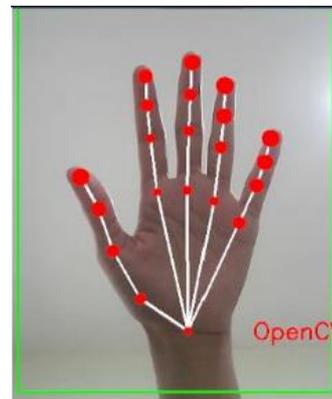


Gambar 2. Hand Landmarks

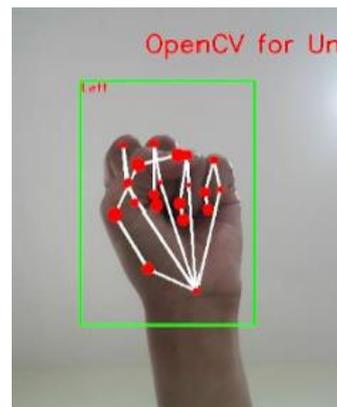
Cara kerja Metode MediaPipe Hand landmark menggunakan pipeline ML yang terdiri dari beberapa algoritma yang diintegrasikan untuk mendeteksi telapak tangan dari sebuah gambar utuh dan memberikan

feedback berupa kotak border atau pembatas tangan. Bentuk tangan yang digunakan sebagai citra gambar dan dipotong ditentukan oleh detektor telapak tangan dan memberikan titik-titik deteksi tangan 3D dengan presisi (ketepatan model) yang tinggi [12].

Sistem yang diusulkan menggunakan hand-pose estimator dari OpenCV yang menggunakan deep-learning sebagai deteksinya. Sistem melakukan deteksi gerakan untuk sandi morse berdasarkan karakteristik dari titik ruas jari. Terdapat 15 titik ruas jari yang digunakan pada penelitian ini yaitu 3 titik jari untuk ibu jari dan 3 titik ruas jari untuk 4 jari lainnya. Titik ruas jari ini mewakili posisi jari dan bentuk jari yang dilakukan oleh pengguna. Kemudian, posisi jari tersebut diukur jaraknya dan diskalakan agar lebar tangan semua gerakan antar pengguna dapat memiliki jarak nilai yang sama. Pada setiap gerakan membuka dan menutup, terdapat perubahan dari setiap kemiringan ruas jari. Posisi jari membuka, dan menutup dapat dilihat sebagai berikut. Pose gerakan tangan saat membuka dan menutup dapat dilihat Gambar 3 dan 4.



Gambar 3. Pose Gerakan Tangan Bentuk Membuka (Semua Jari Membuka)



Gambar 4. Pose Gerakan Tangan Bentuk Menutup (Semua Jari Menutup)

Setiap melakukan gerakan, akan diawali dengan posisi tangan setengah membuka untuk menandakan bahwa data akan diambil. Dataset yang telah divalidasi kemudian digunakan sebagai acuan untuk

mengintegrasikan aplikasi antarmuka sehingga menjadi sistem deteksi hand-gesture untuk menghafal sandi morse.

Metode *Min-Max* adalah metode yang digunakan untuk normalisasi data karena model data berupa vektor fitur tidak konsisten dengan skala. Kelebihan metode ini adalah keseimbangan antara nilai-nilai data perbandingan. Tidak ada data yang dominan dibandingkan data lainnya. Proses ini dilakukan karena model data fitur mempunyai perbedaan yang signifikan. Normalisasi min-max digunakan untuk menyesuaikan rentang data agar sama antara satu data dengan data lainnya. Proses normalisasi dilakukan dengan mencari nilai maksimum dan minimum dari setiap baris data [13].

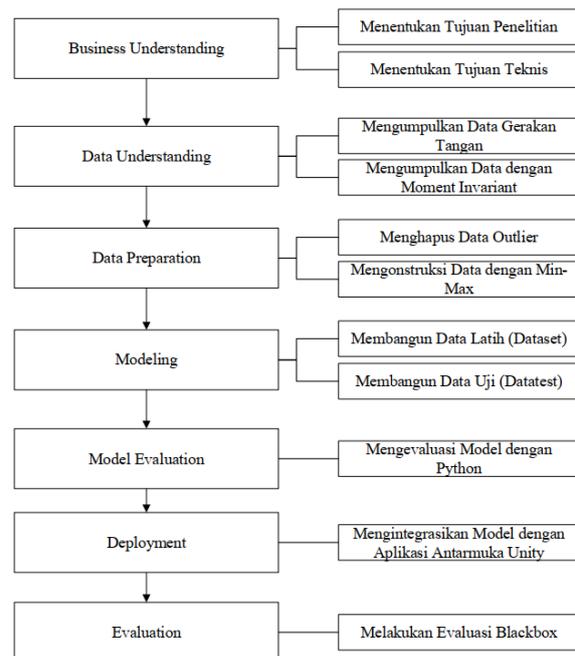
File Comma Separated Values (CSV) adalah jenis file teks yang umumnya berisi daftar data dan dapat juga dikenal dengan sebutan Character Separated Values atau Comma Delimited files. File-file ini sering digunakan untuk melakukan pertukaran data antara berbagai aplikasi. Meskipun karakter koma biasanya digunakan sebagai pemisah (delimiter) antara data, terkadang karakter lain, seperti titik koma, juga digunakan dalam format file CSV. File CSV dapat diintegrasikan pada Unity untuk menyimpan data [14]. Untuk menyimpan data rekaman dari titik-titik ruas jari untuk setiap huruf yang diperagakan oleh pengguna, maka digunakan CSV sebagai media penyimpanan. Selain kemudahan penulisan data, CSV memberikan kita kemudahan ketika kita ingin membuat grafik atau histogram untuk mewakili nilai titik ruas20 tertentu sebagai visualiasi. CSV ini nantinya akan digunakan sebagai input data pada python baik untuk dataset dan data-test serta untuk dataset ketika mengimplementasikan deteksi sistem pada aplikasi antarmuka.

Algoritma K-Nearest Neighbors (KNN) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Prinsip dasar dari algoritma KNN adalah menentukan kelas atau nilai dari sebuah titik data baru berdasarkan kelas atau nilai dari "tetangga terdekat" yang sudah diketahui. KNN bekerja dengan cara menghitung jarak antara titik data baru dengan titik-titik data yang ada, lalu memilih K jumlah tetangga terdekat (dengan K adalah bilangan positif yang ditentukan sebelumnya), kemudian menentukan klasifikasi atau prediksi berdasarkan mayoritas atau rata-rata dari tetangga-tetangga tersebut [15], [16]. KNN merupakan algoritma *machine learning* yang bersifat *non-parametric* dan *lazy learning*. Metode yang bersifat *nonparametric* memiliki makna bahwa metode tersebut tidak membuat asumsi apa pun tentang distribusi data yang mendasarinya. Dengan kata lain, tidak ada jumlah parameter atau estimasi parameter yang tetap dalam model, terlepas data tersebut berukuran kecil ataupun besar [17].

Tujuan dari penelitian ini adalah untuk mengembangkan sistem deteksi hand gesture sebagai alat bantu dalam mempermudah menghafal sistem sandi morse serta mengintegrasikan teknologi hand gesture untuk mengoreksi gerakan tangan sandi morse. Diharapkan penelitian ini dapat membantu pengendali frekuensi radio untuk mempermudah penghafalan sandi garis dan titik. Hal ini sangat penting untuk penerjemahan data pada komunikasi radio dalam berbagai konteks. Penelitian ini juga akan menghasilkan pengembangan sistem deteksi handgesture yang inovatif dan dapat diterapkan dalam penghafalan sandi morse. Dan sistem yang dikembangkan diharapkan dapat membantu meningkatkan efektivitas menghafal sandi morse dan mengubah pengenalan Gerakan tangan menjadi alat bantu yang lebih menarik dan seru.

2. Metode Penelitian

Metode pengembangan sistem dilakukan dengan menerapkan langkah-langkah yang terdapat pada SKKNI yaitu Business Understanding, Data Understanding, Data Preparation, Modeling, Model Evaluation, Deployment dan Evaluation [18] dapat dilihat pada Gambar 5.



Gambar 5. Metode Pengembangan Sistem

2.1 Business Understanding

Pada tahap *Business Understanding*, tujuan yang dicapai dari penelitian ini adalah untuk membuat sebuah sistem hand-gesture untuk mempermudah menghafal sandi morse. Beberapa tahap yang akan dilakukan antara lain: (1) Webcam akan mengambil gerakan tangan yang kemudian dideteksi pose

tangganya menggunakan OpenCV dan Media Pipe Hand Landmarks untuk mendapatkan titik setiap ruas jari. (2) Tugas analitik yang digunakan adalah klasifikasi jenis gerakan tangan ke dalam huruf abjad sandi morse menggunakan metode KNN. (3) Lama waktu pada saat melakukan gerakan adalah dalam batas waktu yang ditentukan yaitu 5 detik untuk menjaga panjang data sama antara satu sama lain. (4) Proses deteksi gerakan pengguna dilakukan menggunakan tombol Mulai untuk mendapatkan awal dan akhir gerakan sehingga tidak terdapat noise gerakan. (5) Performa akan diukur berdasarkan presisi, akurasi dan recall dari sistem. (6) Data baik dataset dan datatest direkam dan diambil di Balai Monitor Spektrum Frekuensi Radio Kelas I Samarinda.

2.2 Data Understanding

Pada tahap *Data Understanding*, Pada penelitian ini, data yang telah dikumpulkan adalah berupa titik ruas jari yang kemudian pada setiap titiknya, data akan diproses sebagaimana berikut: (1) Data total diperoleh dari 50 orang dengan 26 huruf untuk setiap orangnya sehingga totalnya 1300 data. (2) Struktur data yang digunakan pada data ini yaitu berupa titik setiap ruas jari yang berjumlah 15 berupa x dan y yang diambil dalam 5 detik. Sehingga untuk setiap huruf memiliki 30 data kolom dikali baris dari jumlah fps dikali 5. (3) Setiap rekaman titik ruas jari diwakili menggunakan Moment Invariant sebagai bentuk distribusi data. (4) Setelah itu beberapa atribut ruas akan diseleksi kembali untuk mengeliminasi atribut yang tidak berdampak signifikan. (5) Data akan ditulis pada sebuah file CSV beserta label class (huruf A-Z) serta nama kolom dari tiap data tersebut (titik ruas ibu jari ke-1 sampai titik ruas jari kelingking ke-4).

2.3 Data Preparation

Pada tahapan *Data Preparation*, Sebelum data digunakan, data disiapkan dengan beberapa tahap yaitu membersihkan data dari noise atau gangguan pada data. Untuk menangani perbedaan skala data karena perbedaan posisi maka diperlukan feature engineering. Data distandarisasi dengan tujuan menjaga skala antar data tetap sama dengan menggunakan metode min-max.

2.4 Modeling

Selanjutnya pada tahapan *Modeling*, pada penelitian ini, algoritma yang digunakan adalah K-Nearest Neighbor (KNN) terhadap data yang telah didapatkan. Data dibagi menjadi dua bagian yaitu data latih (dataset - 80%) serta data uji (datatest-20%) sebagaimana pembagian ideal pada machine learning yang umumnya digunakan. Karena data yang dikumpulkan adalah sebanyak 1300 data (26 huruf x 50 orang). Maka dataset yang digunakan sebanyak 1040 dan datatest sebanyak 260. Setelah itu data dievaluasi performanya menggunakan confusion matrix untuk mendapatkan data berupa presisi, akurasi dan recall dari sistem.

2.5 Model Evaluation

Pada tahapan *Model Evaluation*, melakukan evaluasi performa dari sistem deteksi berdasarkan dataset dan data-test. Pengujian performa atau evaluasi performa ini ditujukan untuk mendapatkan tingkat akurasi, presisi, dan recall dari setiap gerakan huruf abjad sehingga sistem dapat dibangun dengan baik. Pada penelitian ini, Python digunakan sebagai alat evaluasi awal dari performa sistem sebelum dataset diterapkan dan diintegrasikan pada Aplikasi dari sistem deteksi *hand-gesture* sandi morse yang dibangun.

2.6 Deployment

Pada tahapan *Deployment*, sistem perekaman data perlu dibangun karena kebutuhan pengambilan data berupa titik setiap ruas jari yang mana digunakan sebagai acuan trajectory atau lintasan dari gerakan tangan pengguna. Sebuah antarmuka dengan Unity3D berupa tampilan kamera dengan titik deteksi tiap ruas tangan pengguna serta tombol untuk memulai perekaman data yang digunakan saat pengambilan data. Selain itu, sistem akhir juga dibangun setelah melakukan analisa performa menggunakan Python. Adapun bagian-bagian yang akan dibangun dalam sistem ini adalah sebagai berikut: (1) Pemrograman sistem, pemrograman yang akan dibangun menggunakan Unity3D dengan menggunakan Bahasa C#. (2) Pengolahan Citra, pengolahan deteksi objek tangan yang akan menggunakan algoritma Mediapipe Hand Landmark dengan Bahasa C# dan library OpenCV. (3) Sistem deteksi hand-gesture sandi morse menggunakan Unity3D, webcam yang terhubung dengan komputer dan ditampilkan pada antarmuka aplikasi yang dibuat di Unity3D. Data gambar yang ditangkap, kemudian diolah oleh OpenCV dan algoritma *Mediapipe Hand Landmark* dan dideteksi pola gerakannya berdasarkan titik ruas jari terhadap label *class* (huruf A-Z) yang ada di dataset menggunakan KNN.

2.7 Evaluation

Pada tahapan *Evaluation*, model yang telah diterapkan pada sistem akan dievaluasi menggunakan metode black box. Pengujian kinerja berupa waktu respons sistem terhadap setiap gestur untuk memastikan deteksi yang cepat dan responsif. (1) Pengujian fungsionalitas dengan memeriksa apakah sistem dapat mengenali dan mengklasifikasikan gestur tangan sesuai dengan sandi Morse yang diintegrasikan serta mengevaluasi respons sistem terhadap variasi input seperti gerakan tangan yang lambat atau cepat. (2) Pengujian integrasi berupa memastikan bahwa sistem KNN terintegrasi secara baik dengan komponen-komponen lain dalam sistem deteksi *hand-gesture* serta menguji elemen-elemen seperti antarmuka pengguna dan sistem penyimpanan data.

3. Hasil dan Pembahasan

Tahap perancangan sistem dimulai dengan menentukan algoritma yang digunakan. Pertama-tama, tahap perekaman data dilakukan sebagaimana berikut: (1) Pendeteksian landmark ruas jari tangan menggunakan *MediaPipe Hand Landmark* seperti terlihat pada Gambar 6 - 8. (2) Titik-titik ruas jari tangan diambil nilai x dan y untuk 15 ruas jari (setiap jari memakai 3 ruas jari). (3) Sistem perekaman dibuat dengan Unity untuk mengambil data x dan y dari 15 ruas selama 5 detik. (4) Data tersebut kemudian diubah skalanya menggunakan MinMax (5) Data MinMax kemudian diproses menggunakan Moment Invariant agar data menjadi 1 baris data. (6) Data kemudian ditulis ke csv berupa Raw data, MinMax Data, dan MI(Moment Invariant Data).

Lalu dilanjutkan dengan tahapan pembuatan sistem deteksi *hand-gesture* untuk membantu menghafal sandi morse. Proses sistem deteksi dilakukan sebagaimana langkah berikut: (1) Menggunakan algoritma yang telah dibuat berupa deteksi landmark. (2) Sistem merekam gerakan tangan selama 5 detik. (3) Nilai x dan y diambil dari hand landmark. (4) Konversi nilai x dan y menggunakan MinMax untuk skala data. (5) Data diekstraksi menggunakan Moment Invariant untuk menjadikan data menjadi 1 baris. (6) Data Moment Invariant yang telah diekstraksi kemudian dikirimkan ke Python untuk dicocokkan dengan model KNN. (7) Data tersebut kemudian dibandingkan dan mendapatkan label huruf sesuai jarak terdekat (KNN). (8) Label huruf yang sesuai ditampilkan pada antarmuka.

Setelah itu, penjelasan setiap algoritma yang digunakan pada system akan dijelaskan menggunakan alur SKKNI. Pada tahap persiapan dan pengambilan data, data berupa gerakan tangan direkam berdasarkan titik-titik dari ruas jari tangan berupa X dan Y. Data direkam dalam 5 detik dimana gerakan yang diperagakan sesuai dengan titik dan garis sandi morse. Gerakan titik diwakili oleh tangan menggenggam sedangkan gerakan garis diwakili oleh tangan membuka. Setiap gerakan diawali dengan pose tangan setengah membuka. Setiap 1 huruf untuk 1 orang direkam selama 5 detik sehingga data tersebut memiliki sekitar 30 sampai dengan 50 baris data (bergantung pada FPS) yang disebut "Raw Data". Data tersebut kemudian diproses menggunakan metode MinMax untuk mengubah skala data menjadi range (0-1) yang disebut dengan "MinMax Data". Dikarenakan jumlah data yang terlalu banyak (30-50 baris data hanya untuk 1 huruf dan memiliki 30 fitur), data kemudian diekstrak berdasarkan nilainya menggunakan pendekatan Moment Invariant (Mean, Median, Standar Deviasi, dan Skewness) yang disebut MI Data. Fitur dengan nilai x dan y memiliki titik 0 hanya 1 yaitu mengikuti posisi ruangan Unity (tengah area $y=0$, tengah area $x=0$ berbeda dengan tengah layer kamera).

Karena objek tangan tidak selalu berada di tengah layar dan dapat memiliki ukuran berbeda sehingga mempengaruhi nilai X dan Y dari objek tersebut maka pada pemrosesan data atau feature engineering, dilakukan MinMax untuk membuat setiap titik memiliki minimum dan maximum sesuai dengan pergerakan ruas jari tersebut.

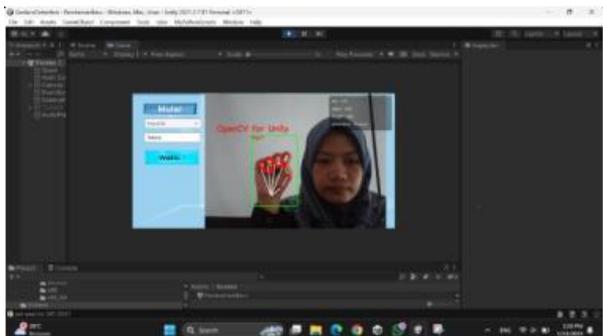


Gambar 6. Tangan Setengah Membuka

Landmark tangan ditampilkan berdasarkan input dari tangan yang ditangkap oleh kamera dan ditampilkan pada Unity, kemudian menggunakan OpenCV dan *MediaPipe Hand Landmark* sebagai algoritmanya. Kode dan antarmuka untuk mendeteksi Landmark ruas jari dan mengambil rekaman data serta menampilkannya pada Unity adalah sebagai berikut pada *script MediaPipe HandPose Landmarks*.



Gambar 7. Tangan Membuka



Gambar 8. Tangan Menggenggam

Ruas jari tangan dideteksi menggunakan algoritma *MediaPipe Hand Landmark*. Dari landmark yang dideteksi, kemudian dilakukan konversi nilai menjadi

variabel x dan y untuk ruas jari 2,3,4,6,7,8,10,11,12,14,15,16,18,19, dan 20. Perekaman data dibuat menggunakan Unity C# untuk merekam, menulis data gerakan tangan menjadi beberapa data yaitu Raw Data, MinMax Data, dan MI. Filecode akan mendapatkan nilai sesuai input nama pengguna yang kemudian akan menjalankan fungsi rekam data. Pada setiap frame, data dari Dnn berupa ruas jari x dan y yang telah dibuat kemudian dibaca dan ditambahkan pada listJari. Kemudian, dilakukan pemanggilan fungsi WaktuRekamData untuk menghentikan perekaman kemudian memanggil fungsi TulisRawData, TulisMinMaxData. Fungsi TulisRawData dipanggil untuk melakukan pembuatan string dengan delimiter “,” untuk setiap x dan y ruas jari sehingga menjadi format yang sesuai50 dengan file csv. Data yang telah ditulis menggunakan kode akan menjadi sebuah file dengan nama orang dan gerakan huruf yang dilakukan seperti pada Tabel 1. Setiap kolom data mewakili fitur data sedangkan baris adalah data dari setiap framenya.

Tabel 1. Hasil Rekaman Data Raw

No.	Ruas2x	Ruas2y	Ruas3y	Ruas4
1	452.2279	151.3161	464.0583	125.7671
2	453.0695	150.9761	465.1988	125.8302
3	452.5858	150.5861	464.1926	124.629
4	452.8391	150.3271	465.4369	124.9696
5	451.3401	150.386	462.6458	125.4935
6	448.9966	150.3305	459.6321	123.5527

Fungsi TulisMomentInvariant dipanggil untuk melakukan skala data menjadi 0-1 yang setiap x dan y ruas jari sehingga semua data memiliki format yang sama. Data setiap fitur kemudian digabung menggunakan delimiter “,” sebagai tring agar menjadi format yang sesuai dengan file csv. Data string kemudian ditulis pada file CSV. Setelah dilakukan penulisan data MinMax, data akan menjadi 1 file csv seperti dapat dilihat pada Tabel 2.

Tabel 2. Hasil Rekaman MinMax Data

No.	Ruas2x	Ruas2y	Ruas3y	Ruas4
1	0.680009	1	0.723789	0.826092
2	0.73002	0.972481	0.743091	0.831907
3	0.701277	0.940915	0.726062	0.722363
4	0.716329	0.919951	0.747121	0.753406
5	0.627253	0.924719	0.699884	0.801156
6	0.487993	0.920227	0.648879	0.624267

Setelah dilakukan MinMax, data kemudian akan diekstraksi menggunakan Moment Invariant (Mean, Median, Stdev, Skewness) agar data yang awalnya berbentuk 1 file menjadi 1 baris.

Pada tahapan *Feature Engineering*, dilakukan pemilihan fitur yang digunakan untuk deteksi sandi morse. Terdapat 15 titik (warna biru) atau fitur yang dipakai dari total 21 fitur untuk setiap bagian tangan sehingga jika diambil nilai x dan y dari titik tersebut, maka total fitur tersebut akan menjadi 30 fitur. Untuk mengurangi jumlah fitur yang digunakan, dilakukan

observasi perubahan nilai titik-titik saat gerakan membuka menutup. Perubahan posisi yang tidak terlalu signifikan adalah pada titik: 0, 1, 5, 9, 13, dan 17. Oleh karena itu, sejak awal perekaman data titik tersebut tidak dijadikan fitur dalam proses pengenalan gerakan sandi morse. Setelah itu dilakukan correlation matrix untuk mengecek fitur-fitur terpenting yang memiliki korelasi satu sama lain. Selanjutnya, pada tahap feature engineering, dilakukan proses Scaling data menggunakan MinMax untuk mengatasi ukuran tangan yang berbeda antar pengguna.

Sebelum melakukan evaluasi dari klasifikasi data, data dibagi menjadi data test 20% dan data set 80% berdasarkan pembagian ideal datapada machine learning. Selain itu, dilakukan penentuan K yang hasilnya dapat dilihat pada Tabel 3.

Tabel 3. Hasil Evaluasi Performa K-1

	Precision	Recall	F1-Score	Support
A	0.90	0.82	0.86	11.00
B	0.67	0.83	0.74	12.00
C	0.60	0.67	0.63	9.00
D	0.67	0.80	0.73	10.00
E	0.89	1.00	0.94	8.00
F	0.56	0.83	0.67	6.00
G	0.91	0.83	0.87	12.00
H	0.78	0.58	0.67	12.00
I	1.00	0.67	0.80	6.00
J	0.78	0.70	0.74	10.00
K	0.67	0.75	0.71	8.00
L	0.83	0.56	0.67	9.00
M	0.92	1.00	0.96	12.00
N	1.00	1.00	1.00	11.00
O	1.00	0.88	0.93	8.00
P	0.83	0.83	0.83	12.00
Q	0.91	0.83	0.87	12.00
R	1.00	0.89	0.94	9.00
S	0.73	0.92	0.81	12.00
T	1.00	1.00	1.00	8.00
U	0.78	0.88	0.82	8.00
V	0.83	0.83	0.83	12.00
W	0.67	0.60	0.63	10.00
X	0.80	0.80	0.80	15.00
Y	0.89	0.89	0.89	9.00
Z	0.88	0.78	0.82	9.00
Accuracy			0.82	260
Macro AVG	0.83	0.81	0.81	260
Weighted AVG	0.83	0.82	0.82	260

Sebagai tambahan, untuk mendapatkan pemahaman lebih jelas, dilakukan pengujian confusion matrix untuk melihat seberapa banyak huruf terprediksi benar. Selain itu, model juga dibuat menggunakan joblib supaya model dapat digunakan sebagai deteksi nantinya.

Precision mengukur seberapa akurat prediksi positif model. Dengan kata lain, ini menunjukkan persentase dari prediksi positif yang benar-benar positif[19]. Ini dihitung sebagai: $TP / (TP + FP)$, di mana TP adalah True Positive (positif yang benar) dan FP adalah False Positive (positif yang salah).

Recall (Sensitivitas): Recall mengukur seberapa baik model menangkap semua kasus positif dari dataset. Ini menunjukkan persentase kasus positif yang benar-benar berhasil terdeteksi [19]. Ini dihitung sebagai: $TP / (TP + FN)$, di mana TP adalah True Positive dan FN adalah False Negative (negatif yang salah). F1-Score: F1-Score adalah rata-rata harmonik dari precision dan recall. Ini memberikan keseimbangan antara precision dan recall. F1-Score adalah rata-rata harmonis dari precision dan recall. F1-Score lebih tepat digunakan ketika kita membutuhkan keseimbangan antara precision dan recall, terutama jika distribusi kelas tidak seimbang [19].

Support menunjukkan jumlah aktual kasus dari setiap kelas dalam dataset. Ini tidak mempengaruhi kinerja model secara langsung, tetapi menunjukkan berapa banyak sampel dari setiap kelas yang digunakan untuk menghitung metrik [19]. Accuracy: Accuracy adalah proporsi dari total prediksi yang benar (positif dan negatif) terhadap jumlah total contoh. Ini dihitung sebagai: $(TP + TN) / (TP + TN + FP + FN)$, di mana TN adalah True Negative (negatif yang benar). Sebelum melakukan implementasi KNN menggunakan sebuah nilai K, nilai K harus ditentukan terlebih dahulu. Penentuan nilai K terbaik, dilakukan dengan membandingkan nilai akurasi K. Angka yang dipakai adalah angka ganjil agar tidak menimbulkan kesalahan klasifikasi jika data terdeteksi sebagai 2 huruf dengan jarak K yang sama. Nilai akurasi tiap K diambil sebanyak 5 kali dan dirata-rata akurasinya. Berdasarkan percobaan, dapat diketahui bahwa nilai akurasi terbesar adalah nilai K-1. sehingga K-1 dapat digunakan untuk implementasi.

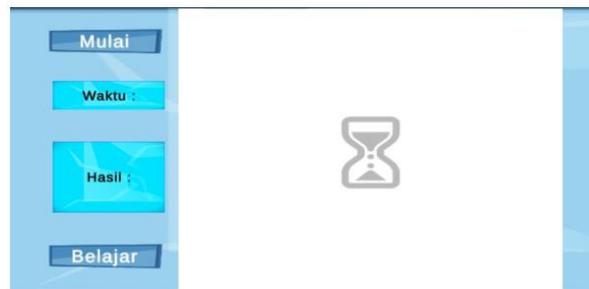
Macro average menghitung rata-rata metrik untuk setiap kelas, lalu merata-ratakannya tanpa mempertimbangkan jumlah contoh dalam setiap kelas. Dengan kata lain, setiap kelas memiliki bobot yang sama dalam perhitungan, tidak peduli seberapa besar atau kecil jumlah data dalam kelas tersebut [20]. menghitung rata-rata metrik untuk setiap kelas, tetapi memberikan bobot sesuai dengan jumlah contoh di setiap kelas. Artinya, kelas yang memiliki lebih banyak sampel akan memberikan kontribusi yang lebih besar terhadap rata-rata akhir[20]. Ini berguna saat kelas memiliki distribusi yang tidak seimbang. K-1 menjadi nilai optimal untuk akurasi karena keterbatasan data yang penulis kumpulkan yaitu 50 orang dengan 26 huruf. Variasi data orang yang berjumlah 50 dapat mempengaruhi sensitivitas akurasi ketika nilai K semakin besar yang membuat huruf atau data masuk ke dalam klasifikasi huruf lainnya. Rata-rata akurasi yang didapatkan memiliki nilai lebih dari 70%. Hal ini menunjukkan, meskipun dengan keterbatasan data yang dimiliki, deteksi sandi morse menggunakan landmark (titik ruas jari) tangan, dengan pendekatan Moment Invariant (Mean, Median Standar Deviasi, Skewness)

untuk gerakan dinamis menunjukkan hasil yang cukup bagus.

Hasil perbandingan menunjukkan bahwa K-1 memiliki nilai tertinggi rata-rata akurasi adapat dilihat pada Tabel 4.

Tabel 4. Perbandingan Nilai K

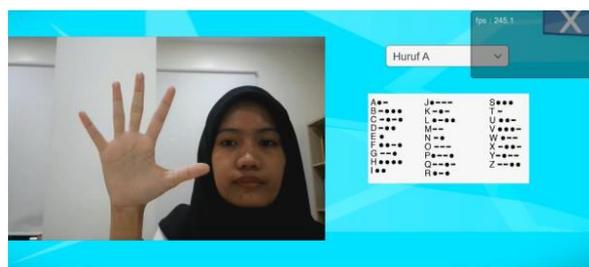
Nilai K	Repetisi 1	Repetisi 2	Repetisi 3	Repetisi 4	Repetisi 5	Rata-Rata
K-1	80%	80%	79%	82%	84%	81%
K-3	82%	74%	79%	80%	82%	79%
K-5	82%	75%	78%	80%	79%	79%
K-7	74%	83%	78%	78%	77%	78%
K-9	75%	82%	83%	78%	78%	79%
K-11	75%	79%	83%	79%	75%	78%
K-13	75%	74%	78%	76%	81%	77%



Gambar 9. Antarmuka Sistem Deteksi Sandi Morse

Implementasi deteksi gerakan tangan untuk Sandi Morse kemudian diterapkan pada Aplikasi dengan Unity3D dan Python. Unity3D digunakan untuk mendesain antarmuka agar terlihat bagus dan menarik untuk belajar sandi morse yang memiliki tombol mulai, waktu, hasil dan tombol belajar dapat dilihat pada Gambar 9.

Pada tombol belajar akan menampilkan antarmuka belajar gerakan sandi morse berupa video gerakan sandi morse sesuai huruf yang dipilih dan ingin dipelajari seperti pada Gambar 10.



Gambar 10. Antarmuka Belajar Gerakan Sandi Morse

Evaluasi sistem dilakukan untuk menentukan apakah sistem yang telah dibuat dapat berjalan sebagaimana semestinya. Pada evaluasi kali ini, dilakukan BlackBox Testing. Hasil menunjukkan bahwa pengujian antarmuka berhasil dengan baik. Setelah itu, setiap huruf dilakukan pengecekan apakah setiap huruf dapat mendeteksi dengan benar atau tidak. Setiap huruf dicek

degan 5 orang berbeda untuk menentukan apakah huruf terdeteksi dengan benar dapat dilihat pada Tabel 5.

Tabel 5. Pengujian Deteksi Huruf

No	Huruf	Tes 1	Tes 2	Tes 3	Tes 4	Tes 5	Akurasi
1	A	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
2	B	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
3	C	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
4	D	Gagal	Berhasil	Berhasil	Berhasil	Berhasil	80
5	E	Berhasil	Berhasil	Gagal	Berhasil	Berhasil	80
6	F	Berhasil	Gagal	Berhasil	Gagal	Berhasil	60
7	G	Berhasil	Gagal	Berhasil	Berhasil	Berhasil	80
8	H	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
9	I	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
10	J	Berhasil	Gagal	Berhasil	Berhasil	Berhasil	80
11	K	Berhasil	Berhasil	Gagal	Berhasil	Berhasil	80
12	L	Berhasil	Gagal	Gagal	Berhasil	Berhasil	60
13	M	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
14	N	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
15	O	Gagal	Gagal	Berhasil	Berhasil	Berhasil	60
16	P	Berhasil	Gagal	Berhasil	Berhasil	Berhasil	80
17	Q	Berhasil	Gagal	Berhasil	Gagal	Berhasil	60
18	R	Berhasil	Berhasil	Berhasil	Gagal	Berhasil	80
19	S	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
20	T	Berhasil	Berhasil	Berhasil	Berhasil	Berhasil	100
21	U	Gagal	Berhasil	Berhasil	Berhasil	Berhasil	80
22	V	Berhasil	Gagal	Gagal	Berhasil	Berhasil	60
23	W	Berhasil	Berhasil	Berhasil	Gagal	Berhasil	80
24	X	Berhasil	Berhasil	Gagal	Gagal	Berhasil	60
25	Y	Berhasil	Berhasil	Gagal	Gagal	Berhasil	60
26	Z	Gagal	Gagal	Berhasil	Berhasil	Berhasil	80

Hasil pengujian blackbox menunjukkan bahwa semua huruf berhasil terdeteksi, meskipun tidak semua individu berhasil melakukannya. Pada percobaan dilakukan penambahan waktu tunggu sebelum merekam berupa angka hitung mundur selama 3 detik. Penambahan waktu tunggu ini memberikan dampak positif pada individu yang baru pertama kali mencoba sistem sehingga mampu bersiap untuk melakukan gerakan meskipun menyebabkan waktu lebih lama dari seharusnya (dari 5 detik waktu rekam ditambah 3 detik waktu siap-siap). Meskipun ada beberapa huruf yang memiliki gerakan rumit dengan 4 kode, kebanyakan huruf menunjukkan akurasi setidaknya 60% dan lebih dari 60%.

4. Kesimpulan

Berdasarkan uraian-uraian yang telah dijelaskan dan pembahasan mengenai sistem deteksi *hand-gesture* untuk mempermudah menghafal sandi morse, dapat ditarik kesimpulan sebagai berikut: Berdasarkan percobaan nilai K pada KNN mulai dari K-1 sampai K-13, rata-rata akurasi 77% hingga 81% yang menunjukkan bahwa evaluasi model dapat digunakan untuk proses deteksi dengan baik. Dari hasil evaluasi performa model sistem deteksi, disimpulkan bahwa nilai K terbaik yaitu K-1. Hasil deteksi huruf sandi morse dengan gerakan tangan yang dievaluasi dengan blackbox menunjukkan akurasi setidaknya 60% sampai 100%. Pengujian antarmuka perekaman dataset dan antarmuka belajar sandi morse menunjukkan bahwa

sistem berjalan dengan baik 100%. Kendala yang dihadapi pada penelitian ini yaitu terbatasnya partisipan (50 orang) yang membuat akurasi beberapa pendeteksian huruf seperti huruf V, X, dan Y bernilai 60 karena kemiripan gerakan dan gabungan dari 4 kombinasi gerakan.

Selain itu, pengguna masih kesulitan memperagakan gerakan membuka dan menutup tangan yang benar dan konsisten saat menggunakan sistem. Sistem deteksi *hand-gesture* untuk mempermudah mempelajari sandi morse berhasil dibuat dengan akurasi sistem dan antarmuka aplikasi yang baik sehingga dapat digunakan sebagaimana semestinya.

Pada penelitian selanjutnya diharapkan untuk mengambil lebih banyak variasi data dari partisipan atau orang agar dapat meningkatkan akurasi pendeteksian huruf dan menghindari kesalahan deteksi pada huruf yang memiliki Gerakan yang mirip maupun kombinasi 4 gerakan tangan. Untuk penelitian selanjutnya diharapkan adanya umpan balik apakah lebar membuka dan menutup tangan telah sesuai dengan yang ditentukan. Untuk meningkatkan akurasi, menggabungkan pendekatan Moment Invarian dengan pendekatan statistik lain sangat disarankan pada penelitian selanjutnya. Pada penelitian selanjutnya, diharapkan peneliti dapat bekerjasama dengan seluruh pengendali frekuensi radio di Balai Monitor Spektrum Frekuensi Radio Kelas I Samarinda agar sistem dapat dengan mudah diterapkan dan mendapat umpan balik secara langsung dari penelitian ini.

Daftar Rujukan

- [1] F. Wang, A. Hu, Y. Song, W. Zhang, J. Zhu, and M. Liu, "Morse Code Recognition Based on a Flexible Tactile Sensor with Carbon Nanotube/Polyurethane Sponge Material by the Long Short-Term Memory Model," *Micromachines (Basel)*, vol. 15, no. 7, p. 864, Jun. 2024, doi: 10.3390/mi15070864.
- [2] G. Bhavania, G. A. Kumar, Kumarm J. Sharat, and K. Tarun, "Morse Code Decoding Using Sound Waves in Machine Learning," *International Journal of Research Publication and Reviews*, vol. 5, no. 11, pp. 6196–6201, Nov. 2024.
- [3] T. W. Tsang and H. J. Lu, "Hand movement improves word memory of Grade 1 students," *Int J Sch Educ Psychol*, vol. 10, no. 3, pp. 408–417, Jul. 2022, doi: 10.1080/21683603.2020.1862724.
- [4] A. Fendar, S. Lokhande, C. Karpe, and A. Ingle, "MORSE CODE DETECTION USING MACHINE LEARNING," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 7, no. 4, pp. 5023–5028, Apr. 2025.
- [5] PusbangJusinfo, "Sandi Morse dan Cara Cepat Menghafalnya," <https://pramukadiy.or.id/sandi-morse-dan-cara-cepat-menghafalnya/>.
- [6] D. Savitri, "Kode Morse: Pengertian, Sejarah dan Cara Menghafalnya Baca artikel detikedu, 'Kode Morse: Pengertian, Sejarah dan Cara Menghafalnya,'" <https://www.detik.com/edu/detikpedia/d-6469196/kode-morse-pengertian-sejarah-dan-cara-menghafalnya>.
- [7] OpenCV.org, "About OpenCV," 2023.
- [8] N. F. Ramadhanti, M. Lamada, and M. Riska, "Pengembangan Aplikasi Game Edukasi 3D 'Finding Geometry' Berbasis Unity Sebagai Media Pembelajaran Bangun Ruang Matematika," *Jurnal MediaTIK*, vol. 4, no. 2, May 2021.
- [9] R. Apriansyah, Rusidi, and D. Pujiyanto, "Bangun Alat Deteksi Penggunaan Masker Menggunakan Arduino Pada Layanan Penerimaan Mahasiswa Baru Universitas Mahakarya Asia," *Jurnal Informatika dan Komputer (JIK)*, vol. 13, no. 2, pp. 16–22, Dec. 2022.
- [10] A. Darmawan, "Aplikasi Hand Gesture Recognition Sebagai Media Penerjemah Bahasa Isyarat Berbasis Android," Universitas Komputer Indonesia, Bandung, 2021.
- [11] MediaPipe Hands, "MediaPipe Solutions," <https://mediapipe.readthedocs.io/en/latest/solutions/hands.html>.
- [12] F. Zhang *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking," 2020. [Online]. Available: <https://arxiv.org/abs/2006.10214>
- [13] A. Rahagiyanto, A. Basuki, and R. Sigit, "Moment Invariant Features Extraction for Hand Gesture Recognition of Sign Language based on SIBI," *EMITTER International Journal of Engineering Technology*, vol. 5, no. 1, pp. 119–138, Jul. 2017, doi: 10.24003/emitter.v5i1.173.
- [14] Unity Technologies, "Importing and exporting CSV files," <https://docs.unity3d.com/Packages/com.unity.localization@1.4/manual/CSV.html>.
- [15] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Ann Transl Med*, vol. 4, no. 11, pp. 218–218, Jun. 2016, doi: 10.21037/atm.2016.03.37.
- [16] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York, NY: Springer US, 2021. doi: 10.1007/978-1-0716-1418-1.
- [17] adminlp2m, "Algoritma K-Nearest Neighbors (KNN) – Pengertian dan Penerapan," <https://lp2m.uma.ac.id/2023/02/16/algoritma-k-nearest-neighbors-knn-pengertian-dan-penerapan/>.
- [18] Kementerian Ketenagakerjaan, *Penetapan Standar Kompetensi Kerja Nasional Indonesia Kategori Informasi dan Komunikasi Golongan Pokok Aktivitas Pemrograman, Konsultasi Komputer, dan Kegiatan yang Berhubungan Dengan Itu (ybdi) Bidang Keahlian Artificial Intelligence Subbidang Data Science*. Indonesia: Keputusan Menteri Ketenagakerjaan (Kepmenaker), 2020.
- [19] C. C. Aggarwal, *Machine Learning for Text*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-96623-2.
- [20] *Data Mining*. Elsevier, 2012. doi: 10.1016/C2009-0-61819-5.