Terakreditasi SINTA Peringkat 3

Surat Keputusan Direktur Jenderal Pendidikan Tinggi, Riset, dan Teknologi Nomor 225/E/KPT/2022
masa berlaku mulai Vol.7 No. 1 tahun 2022 s.d Vol. 11 No. 2 tahun 2026

Terbit online pada laman web jurnal:
http://publishing-widyagama.ac.id/ejournal-v2/index.php/jointecs

JOINTECS

JOINTS
(Journal of Information Technology and Computer Science)

Vol. 8 No. 3 (2023) 85 - 94

e-ISSN:2541-6448 p-ISSN:2541-3619

Prediksi Gerakan secara *Real-Time* untuk Pengenalan Bahasa Isyarat menggunakan LSTM

Vira Yolanda[1], Widya Silfianti[2]
Program Studi Magister Manajemen Sistem Informasi, Pascasarjana, Universitas Gunadarma
[1]viryolanda@staff.gunadarma.ac.id, [2]wsilfi@staff.gunadarma.ac.id

Abstract

Sign Language is a language used to communicate, especially for deaf and speech impaired people so that they can communicate with each other well. However, due to the lack of education in the community, sign language is rarely used, and many do not even know sign language. So we need a system to recognize sign language so that more people understand sign language. This study developed dynamic motion detection for 6-word sign language recognition in Indonesian sign language (SIBI) by utilizing real-time coordinate changes detected using MediaPipe Holistic and LSTM Neural Network. From the results of the tests that have been carried out, it can be concluded that the results of implementing the Sign Language Recognition System in real-time by utilizing coordinate changes using the MediaPipe Holistic and Long Short Term Memory (LSTM) methods obtained a model with an accuracy value of 96%. The accuracy obtained is 96%, meaning that the model is accurate in classifying movement classes. Each class has excellent precision, recall, f1-score, and support scores with an average of 96%.

Keywords: *artificial intelligence; deep learning; sign language; motion prediction; LSTM; mediapipe holistic.*

Abstrak

Bahasa Isyarat merupakan bahasa yang digunakan untuk berkomunikasi khususnya bagi penyandang tunarungu dan tuna wicara agar dapat saling berkomunikasi dengan baik. Namun karena minimnya pendidikan di masyarakat, bahasa isyarat jarang digunakan, bahkan banyak yang tidak mengetahui bahasa isyarat. Maka diperlukan suatu sistem untuk mengenali bahasa isyarat agar lebih banyak orang yang memahami bahasa isyarat. Dalam penelitian ini, kami mengembangkan deteksi gerakan dinamis untuk pengenalan 6 kata dalam bahasa isyarat Indonesia (SIBI) dengan memanfaatkan perubahan koordinat secara real-time yang dideteksi menggunakan MediaPipe Holistic dan model klasifikasi *LSTM Neural Network*. Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa hasil implementasi Sistem Pengenalan Bahasa Isyarat secara real-time dengan memanfaatkan perubahan koordinat menggunakan metode MediaPipe Holistic dan *Long Short Term Memory* (LSTM) diperoleh sebuah model dengan nilai akurasi 96%. Akurasi yang diperoleh sebesar 96% artinya model tersebut akurat dalam mengklasifikasikan kelas-kelas gerakan. Setiap kelas memiliki nilai presisi, *recall*, *f1-score* dan *support* yang sangat baik dengan rata-rata 96%.

Kata kunci: kecerdasan buatan; pembelajaran mendalam; bahasa isyarat; prediksi gerakan; LSTM; mediapipe holistic.

1. Pendahuluan

Komunikasi merupakan proses setiap manusia dapat berinteraksi sehingga dapat menyampaikan pesan satu sama lain. Namun, manusia yang dilahirkan dalam keadaan tuna rungu dan tuna wicara sulit berkomunikasi. Menurut Organisasi Kesehatan Dunia (WHO), sekitar 466 juta orang di seluruh dunia, yaitu sekitar 6% dari

Diterima Redaksi : 05-04-2023 | Selesai Revisi : 30-08-2023 | Diterbitkan Online : 30-09-2023

populasi penduduk di seluruh dunia memiliki gangguan pendengaran dan diperkirakan jumlah ini akan naik dua kali lipat pada tahun 2050 [1]. Solusi untuk mengatasi kekurangan ini adalah Bahasa Isyarat.

Bahasa Isyarat adalah bahasa yang digunakan untuk komunikasi terutama bagi penyandang tuna rungu dan tuna wicara sehingga mereka bisa saling berkomunikasi dengan baik. Bahasa isyarat menggunakan modalitas ketajaman visual, yang berarti kemampuan untuk melihat, mengungkapkan makna [2]. Namun karena kurangnya edukasi dalam masyarakat, bahasa isyarat jarang digunakan bahkan banyak masyarakat yang tidak mengetahui bahasa isyarat. Maka diperlukannya suatu sistem untuk mengenali bahasa isyarat sehingga semakin banyak masyarakat yang memahami bahasa isyarat. Sistem pengenalan gerakan tangan juga dapat membantu mencapai inklusi tunarungu pada masyarakat melalui peningkatan motivasi belajar dalam bidang pendidikan [3].

Human Computer Interaction (HCI) berkaitan dengan studi tentang antarmuka antara manusia dan komputer. Interaksi antara manusia dan komputer dibangun dengan mempertimbangkan perilaku manusia untuk membuat aplikasi yang dapat memuaskan kebutuhan konsumen [4]. Peran HCI sangat penting saat ini dalam berbagai bidang seperti bidang pendidikan, pertanian, kesehatan, dan lainnya. Pengenalan gerakan adalah salah satu bagian dari HCI yang memiliki peran besar dalam memperoleh informasi dari gerakan tangan dan mengidentifikasinya menjadi suatu informasi. Pengenalan gerakan tangan menjadi salah satu penelitian yang populer di komunitas riset untuk berbagai kebutuhan seperti bahasa isyarat, kontrol gerakan robot, *object tracking*, dan lainnya.

Gerakan tangan dibagi menjadi dua jenis, yakni statis dan dinamis. Gerakan tangan statis adalah gerakan tangan yang memanfaatkan ejaan dari jari atau biasa disebut finger spelling, sedangkan gerakan dinamis adalah gerakan pola tangan yang berubah setiap saat [5]. Sistem pengenalan gerakan memiliki beberapa teknik dalam pengambilan data yaitu vision-based dan data-gloves. Data vision-based diambil dengan kamera yang menggunakan beberapa property seperti warna dan tekstur untuk mengenali suatu gerakan. Namun cara ini memiliki banyak tantangan seperti pencahayaan, latar belakang dan kemiripan warna dengan data gerakan tersebut [6]. Data dengan pendekatan data-gloves menggunakan sensor untuk menangkap posisi koordinat dari tubuh dan menangkap gerakan. Data diambil menggunakan sensor pengambilan gambar yang diperlukan seperti kamera, sensor inframerah/stereo-IR atau sensor kedalaman. Namun, solusi ini tidak cukup ringan untuk dijalankan secara real-time pada perangkat biasa. Dengan demikian solusi tersebut masih memiliki kekurangan yaitu terbatas pada platform yang dilengkapi dengan prosesor yang kuat. Solusi terbaru yang dikembangkan adalah sistem pengenalan gerakan menggunakan nilai ekstraksi titik koordinat dari tangan, wajah dan pose menggunakan *MediaPipe Holistic*. Solusi ini memiliki kinerja yang hampir sama dengan metode kedua namun dengan menggunakan *MediaPipe Holistic* maka tidak perlu perangkat deteksi seperti kamera kedalaman, sensor *infrared*, dan prosesor yang kuat. *MediaPipe Holistic* dapat bekerja secara *real-time* dalam mengambil nilai *keypoint* menggunakan input format RGB melalui kamera webcam.

MediaPipe Holistic merupakan *machine learning pipeline* yang menggunakan kombinasi nilai titik koordinat dari keberadaan tubuh manusia seperti tangan, pose, dan wajah [7]. Pelacakan gerakan tangan pada penelitian ini menggunakan *MediaPipe Holistic*. *MediaPipe Holistic* menggunakan arsitektur dari *MediaPipe Pose* yang ditambahkan dengan *MediaPipe Face Mesh* dan *MediaPipe Hand* yang diintegrasikan dengan model yang berbeda.

MediaPipe pose digunakan untuk melacak pose tubuh dengan ketelitian tinggi secara *real-time* dan membangun segmentasi latar belakang pada frame video dengan representasi warna RGB yang memanfaatkan model BlazePose [8]. Model BlazePose menghasilkan 33 *keypoint* tubuh untuk satu orang dan berjalan pada lebih dari 30 *frame* per detik. Pada daerah pose terdeteksi 33 *keypoint* dan dibangun landmark pada titik tersebut. Nilai dari titik koordinat pose memiliki shape (33,4) yaitu memiliki 33 *keypoint* dengan 4 koordinat yaitu x, y, z dan visibility.

MediaPipe hands adalah solusi pelacakan tangan dan jari dengan ketelitian tinggi secara *pipeline* sehingga dapat bekerja dengan sangat bagus secara *real-time*. MediaPipe hands menggunakan machine learning pipeline untuk mendapatkan 21 *keypoint* dari tangan pada setiap *frame*. Model *BlazePalm* pada MediaPipe hands berguna untuk mendeteksi telapak tangan (*palm detector*) selama kotak pembatas (*bounding box*) mendeteksi objek yang mirip dengan telapak tangan sehingga sistem akan selalu mendeteksi objek tangan [9].

MediaPipe face mesh digunakan untuk melacak 468 *keypoint* dari geometri wajah dan membangun landmark wajah 3D secara *real-time*. *MediaPipe face mesh* menggunakan *machine learning pipeline* untuk menyimpulkan geometri permukaan 3D dengan hanya menggunakan kamera tanpa memerlukan sensor kedalaman/*depth sensor*. *MediaPipe face mesh* menggunakan model *BlazeFace* dan model face landmark yang dikembangkan oleh Google Research. Platform ini memungkinkan memiliki mesh tiga dimensi dari wajah di dalam gambar [10].

Beberapa penelitian serupa telah dikembangkan dalam pembuatan sistem pengenalan Bahasa Isyarat. Terdapat beberapa jurnal yang telah dikaji satu persatu sebagai referensi yang digunakan sebagai acuan dalam penelitian ini, dengan melihat penggunaan beberapa

jenis metode dan algoritma yang digunakan dalam penelitian yang serupa. Sehingga penelitian ini mengacu dan mengembangkan dari beberapa penelitian terdahulu yang terkait

Penelitian dari Saggio [11] mengembangkan pengenalan gerakan tangan bahasa Italia menggunakan sarung tangan sensorik dan unit pengukuran inersia untuk mengumpulkan gerakan jari, pergelangan tangan, dan lengan bawah tangan. Metode klasifikasi yang digunakan adalah K-Nearest Neighbors dengan dynamic time warping dan Convolutional Neural Network (CNN). Pengujian dilakukan oleh tujuh orang yaitu 5 orang laki-laki dan 2 orang perempuan yang berusia 29–54 tahun. Performa metode klasifikasi mendapatkan akurasi 96,6% 3,4 (SD) untuk k-Nearest Neighbors dengan Dynamic Time Warping dan 98,0% 2.0 (SD) untuk Convolutional Neural Networks. Sistem ini berkinerja sangat baik.

Penelitian yang dilakukan oleh Liao [12] mengembangkan aplikasi gerakan tangan dinamis menggunakan jaringan Residual ConvNet dan LSTM Bi-directional 3-dimensi yang disebut sebagai BLSTM-3D Residual Network (B3D ResNet). Metode ini terdiri dari tiga bagian utama. Pertama, objek tangan dilokalisasi dalam bingkai video untuk mengurangi kompleksitas waktu dan kompleksitas ruang perhitungan jaringan. Kemudian, B3D ResNet mengekstrak fitur spatiotemporal dari urutan video, dan menetapkan skor menengah yang sesuai dengan setiap tindakan dalam urutan video menjadi analisis fitur. Pengujian pada penelitian ini dilakukan pada dataset DEVISIGN_D dan SLR_Dataset. Hasilnya menunjukkan bahwa metode yang memperoleh akurasi 89,8% pada dataset DEVISIGN_D dan 86,9% pada dataset SLR_Dataset. Selain itu, B3D ResNet dapat secara efektif mengenali gerakan tangan yang kompleks melalui data urutan video yang lebih besar, dan memperoleh akurasi pengenalan yang tinggi untuk 500 kosakata dari bahasa isyarat tangan Cina.

Penelitian dari Seunghyeok Shin dan Whoi-Yul Kim [13] yaitu mengembangkan sistem pengenalan gerakan tangan dengan teknologi sensor tiga dimensi. Sensor ini tidak hanya mendeteksi bentuk dan gerakan tangan, tetapi juga kerangka tangan. Sehingga perubahan pose tangan untuk gerakan tangan dinamis dapat diprediksi dengan mudah dengan mengekstraksi fitur geometris dari data kerangka tangan, membagi fitur menjadi beberapa bagian, dan melakukan klasifikasi menggunakan metode GRU-RNN (PB-GRU-RNN). Hasil akhir yang mereka dapatkan bahwa metode ini memberikan kinerja yang serupa dengan metode lainnya walaupun menggunakan lebih sedikit parameter. Cara ini masih memiliki banyak kekurangan diantaranya rumit, harga yang mahal untuk perangkat sensor, dan perlu menggunakan perangkat tambahan dalam penggunaannya sehingga gerakan tangan menjadi sedikit sulit dan sulit dilakukan oleh orang lain.

Penelitian dari Mohamad Ali [14] bertujuan untuk mengambangkan sistem pengenalan gerakan tangan statis untuk bahasa arab. Sistem ini menggunakan dua gabungan model jaringan saraf konvolusi yaitu DenseNet121 dan VGG16. Untuk mengevaluasi kinerja digunakan dataset yang terdiri dari berbagai tanda statis. Dataset itu terdiri dari 220.000 gambar untuk 44 kategori yaitu terdapat 32 huruf, 11 angka (0:10), dan 1 untuk tidak ada. Untuk masing-masing tanda-tanda statis, ada 5000 gambar yang dikumpulkan dari sukarelawan yang berbeda. Hasilnya menunjukkan akurasi yang didapatkan sangat bergantung pada jumlah gambar tanda pengenalan yang salah dalam pelatihan, validasi dan menguji dataset. Model jaringan saraf convolutional (CNN) terbaik untuk ekstraksi fitur dan klasifikasi bahasa isyarat Arab adalah DenseNet121 untuk penggunaan model tunggal dan DenseNet121 & VGG16 untuk penggunaan multi-model.

Pada penelitian ini dilakukan penyempurnaan dari penelitian sebelumnya dengan melakukan pengenalan gerakan tangan dinamis melalui nilai ekstraksi *keypoints* yang dapat mendeteksi gerakan. Dimana perubahan koordinat titik gerakan tubuh atau lengan tangan yang dideteksi oleh sistem. Pengenalan gerakan tangan dinamis ini memberikan hasil lebih baik karena dapat mendeteksi variasi kecepatan dan orientasi dari tangan pengguna. Sehingga kendala seperti pencahayaan, latar belakang, dan kecepatan gerakan dapat diatasi dengan metode ini. Lalu klasifikasi pada sistem pengenalan bahasa isyarat menggunakan model *Long Short Term Memory* (LSTM) neural network.

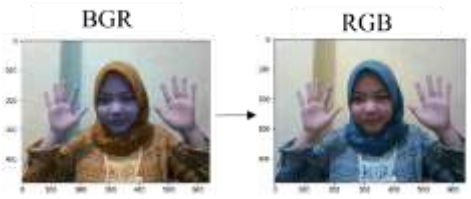
2. Metode Penelitian

Penelitian ini bertujuan untuk mengimplementasi deteksi gerakan dinamis untuk pengenalan 6 kata dalam bahasa isyarat Indonesia (SIBI). Sistem ini dapat melakukan pengenalan gerakan dengan memanfaatkan perubahan koordinat yang dideteksi menggunakan *MediaPipe Holistic* secara *real-time*. Pada penelitian ini terdapat 6 tahap dalam membangun sistem pengenalan gerakan tangan.

2.1. Melacak titik koordinat tubuh

Pelacakan titik koordinat pada daerah wajah, tangan, dan pose merupakan komponen penting untuk memberikan nilai masukkan dalam sistem pengenalan gerakan. Penelitian ini melacak koordinat dari 3 daerah tersebut menggunakan library pada bahasa pemrograman python yaitu *MediaPipe Holistic*. Koordinat ini berguna untuk membangun orientasi kerangka pengguna yang akurat agar sistem dapat mendeteksi gerakan melalui perubahan koordinat yang terdeteksi.

Tahap pertama adalah mengakses kamera webcam secara *real-time* menggunakan library OpenCV. Format warna default dalam library OpenCV adalah BGR (*Blue Green Red*), sedangkan library *MediaPipe Holistic* dapat mendeteksi titik koordinat daerah tubuh berupa *input* dengan format warna RGB (*Red Green Blue)*. Maka,



Gambar 1. Hasil konversi dari format warna BGR ke RGB

Gambar 2. Gambaran implementasi MediaPipe Holistic pada sebuah gambar

pada tahap ini format warna pada video webcam dikonversi dari representasi warna BGR ke RGB menggunakan cv2.COLOR_BGR2HSV pada library OpenCV. Berikut pada Gambar 1 merupakan gambaran hasil konversi dari format warna BGR ke RGB pada sebuah gambar.

Tahap selanjutnya dilakukan pengambilan titik koordinat dari daerah tangan, wajah dan pose dari masukkan melalui kamera webcam dengan representasi warna RGB menggunakan library *MediaPipe Holistic*. *MediaPipe Holistic* bisa diakses dengan mengimport library mediapipe pada bahasa pemrograman python. *MediaPipe Holistic* dapat menggambar landmark dan mendeteksi *keypoint* pada daerah wajah, pose, dan tangan. Berikut pada Gambar 2 merupakan gambaran implementasi dari MediaPipe Holistic.

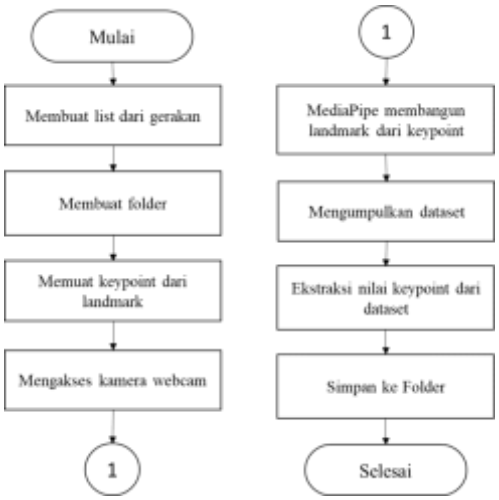
2.2. Ekstraksi Fitur

Tahap ini dilakukan ekstraksi fitur yaitu dengan menggabungkan seluruh *keypoint* dari daerah wajah, telapak tangan, dan pose yang telah dideteksi menggunakan *MediaPipe Holistic*. Himpunan titik koordinat dari seluruh daerah digabungkan agar dapat mendeteksi koordinat landmark menjadi satu vektor yang kemudian digunakan sebagai masukan ke dalam model klasifikasi LSTM. Pada daerah pose terdeteksi 33 *keypoint* dan dibangun landmark pada titik tersebut. Nilai titik koordinat dari pose memiliki shape (33,4) yaitu memiliki 33 *keypoint* dengan 4 koordinat yaitu x, y, z dan visibility. Pada daerah tangan terdeteksi 21 *keypoint* pada tangan kanan dan 21 *keypoint* pada tangan kiri. Nilai titik koordinat dari telapak tangan memiliki shape (21,3) pada tangan kanan dan (21,3) pada tangan kiri yaitu memiliki 21 *keypoint* dengan 3 koordinat yaitu x, y, dan z. Nilai dari wilayah wajah memiliki shape (468,3) yaitu memiliki 468 *keypoint* dengan 3 koordinat yaitu x, y, dan z. Data 2 dimensi yang diperoleh tersebut

```
Extract Keypoint
Input:
  keypoint_pose = [x1,y1,z1,vis1,x2,y2,z2,…]
  keypoint_handleft = [x1,y1,z1,x2,y2,z2,x3…]
  keypoint_phandright = [x1,y1,z1,x2,y2,z2,…]
  keypoint_face = [x1,y1,z1,x2,y2,z2,x3,y3,…]

Output: 1-d array for all keypoints
  Result = [x1,y1,z1,vis1,x2,y2,vis2,…]

  Def extract_keypoint(result)
    //POSE
    Get keypoint_pose
      pose = []
      for i = 1 to keypoint_pose do
        if keypoint_pose
          pose = keypoint_pose.flatten()
        else
          np.zeros(33*4)
      end for

    //HAND LEFT
    Get keypoint_handleft
      hl = []
      for i = 1 to keypoint_handleft do
        if keypoint_handleft
          hl = keypoint_handleft.flatten()
        else
          np.zeros(21*3)
      end for

    //HAND RIGHT
      hr = []
      for i = 1 to keypoint_handright do
        if keypoint_handright
          hl = keypoint_handright.flatten()
        else
          np.zeros(21*3)
      end for

    //FACE
      face = []
      for i = 1 to keypoint_face do
        if keypoint_face
          hl = keypoint_face.flatten()
        else
          np.zeros(468*3)
      end for

    return concatenate([pose,hl,hr,face])
```

dapat menimbulkan gangguan dan mengandung kesalahan jika tubuh bergerak sangat cepat dan menyebabkan kehilangan pelacakan koordinat. Oleh karena itu, semua *keypoint* 2 dimensi diproses terlebih dahulu sebelum digunakan lebih lanjut dengan mengubah nilai tersebut menjadi bentuk array 1 dimensi. Himpunan titik koordinat dari seluruh daerah digabungkan agar dapat mendeteksi koordinat landmark menjadi array 1 dimensi dan mempertimbangkan bahwa beberapa koordinat bisa hilang dari kerangka yang ditangkap ketika seluruh daerah tubuh pada pengguna tidak berada dalam jangkauan. Berikut algoritma dari proses ekstraksi nilai *keypoint*.

Nilai *keypoint* dari semua daerah diproses terlebih dahulu sebelum digunakan lebih lanjut dengan mengubah nilai tersebut menjadi bentuk array 1 dimensi menggunakan function flatten() pada library NumPy dan mempertimbangkan bahwa beberapa *keypoint* bisa hilang dari kerangka yang ditangkap ketika pengguna seluruh daerah tubuhnya tidak berada dalam jangkauan kamera atau *MediaPipe* tidak dapat menangkap beberapa dari landmark. Sehingga pada tahap ini seluruh



Gambar 3. Flowchart proses pembuatan folder dan dataset

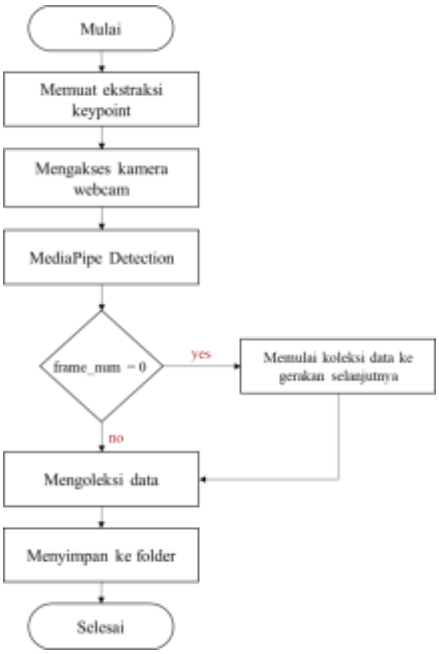
Gambar 4. Gerakan yang akandideteksi

nilai yang tidak terdeteksi diekstraksi dengan mengisi array menggunakan interpolasi linier 1 dimensi sederhana dengan nilai 0. Tahap ini dilakukan menggunakan library NumPy dengan menggunakan function zeros(). Tahap ini dilakukan agar sistem tetap dapat dijalankan walaupun ada beberapa titik landmark yang tidak terdeteksi pada *frame*.

Setelah seluruh nilai *keypoint* di ekstraksi, maka selanjutnya nilai *keypoint* dari setiap daerah akan dijumlahkan seluruhnya menggunakan function concatenate() pada library Numpy. Tahap ini dilakukan agar mendapatkan jumlah keseluruhan *keypoint* yang terdeteksi maupun yang tidak terdeteksi untuk dilanjutkan ke proses selanjutnya. Sehingga didapatkan nilai 1662 *keypoint* yang dideteksi oleh sistem pada setiap *frame* dengan bentuk array 1 dimensi pada setiap gerakan.

2.3. Dataset

Pembuatan dataset pada penelitian ini dilakukan dengan menggunakan serangkaian kode pada bahasa pemrograman python untuk mengambil dataset secara otomatis dengan mengambil nilai perubahan titik koordinat pada setiap gerakan yang terdeteksi pada *frame*. Dataset diambil menggunakan kamera webcam yang dapat diakses menggunakan salah satu library OpenCV pada bahasa pemrograman python yaitu fungsi cv2.VideoCapture(). Berikut flowchart dari seluruh proses untuk mengambil data secara otomatis secara *real-time* dapat dilihat pada Gambar 3.

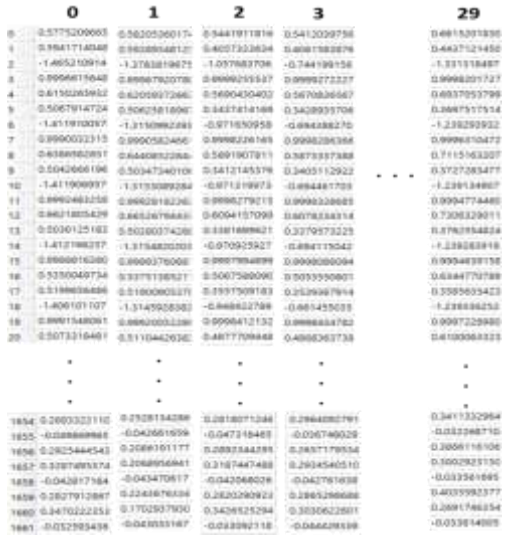


Gambar 5. Database Mirroring Architecture

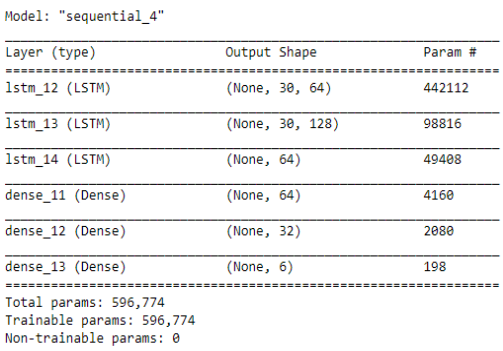
Tahap pertama adalah membuat folder untuk data yang akan diambil. Folder akan dibuat sesuai dengan banyaknya gerakan yang akan dideteksi oleh sistem. Gerakan isyarat yang dideteksi pada sistem ini terdiri dari 6 gerakan dinamis yaitu gerakan terima kasih, tolong, maaf, sehat, saya, dan terkejut. Beberapa kata ini diambil dari bahasa isyarat SIBI (Sistem Bahasa Isyarat Indonesia) yang memiliki karakteristik gerakan yang sangat berbeda satu sama lain. Berikut pada Gambar 4 merupakan gerakan dari bahasa isyarat Indonesia yang dideteksi pada penelitian ini.

Tahap selanjutnya adalah melakukan pengambilan dataset yang akan digunakan untuk *training* dan testing pada penelitian ini. Dataset yang akan diambil berupa nilai *keypoint* dari gerakan yang terdeteksi pada setiap frame. Pada setiap gerakan terdapat 30 frame yang diambil secara otomatis menggunakan kamera webcam. Berikut pada Gambar 5 merupakan flowchart proses pengambilan dataset.

Tahap pertama yaitu mengambil nilai ekstraksi *keypoint* yang telah disimpan pada file bernama 0.npy. Tahap kedua yaitu mengakses kamera webcam menggunakan library OpenCV pada bahasa pemrograman python yaitu fungsi cv2.VideoCapture(). Lalu sistem akan otomatis menggambar landmark yang terdeteksi dari hasil ekstraksi atau joint *keypoint* yang sudah diambil pada proses sebelumnya. Tahap ketiga yaitu pengambilan data. Data yang diambil pada sistem adalah nilai *keypoint* setiap gerakan pada satu frame dan disimpan pada file dengan format .npy atau numpy array. Format file .npy merupakan format file biner standar untuk mempertahankan satu array NumPy yang berisi *array record* dari perubahan nilai koordinat *keypoint* yang sudah diambil dalam suatu frame video.



Gambar 6. Contoh nilai koordinat keypoint yang diambil pada setiap frame



```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_12 (LSTM)               (None, 30, 64)            442112
_________________________________________________________________
lstm_13 (LSTM)               (None, 30, 128)           98816
_________________________________________________________________
lstm_14 (LSTM)               (None, 64)                49408
_________________________________________________________________
dense_11 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_12 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_13 (Dense)             (None, 6)                 198
=================================================================
Total params: 596,774
Trainable params: 596,774
Non-trainable params: 0
```

Gambar 7. Model LSTM

Pada Gambar 6 merupakan contoh perubahan nilai koordinat yang akan dikumpulkan dalam satu frame. Satu frame akan mengambil 30 urutan perubahan *keypoint* setiap detiknya. Setiap gerakan memiliki 30 frame yaitu dari frame ke-0 sampai frame ke-29. Jika frame ke-0 maka sistem akan otomatis mengisi data ke folder gerakan berikutnya dan mengambil data gerakan baru.

2.4. Labeling dan Pre-Processing

Tahap ini dilakukan labelling dan preprocessing pada data agar dapat digunakan pada proses selanjutnya. Pada tahap ini menggunakan library sklearn yaitu train_test_split yang digunakan untuk melakukan partisi data sehingga data akan dibagi untuk *training* dan *testing*. Tahap pertama adalah membuat label dari data yang telah dikumpulkan. Label dibuat dari banyaknya kelas dari gerakan yang dideteksi. Label 0 untuk kelas saya, label 1 untuk kelas terima kasih, label 2 untuk kelas tolong, label 3 kelas untuk sehat, label 4 untuk kelas terkejut, dan label 5 untuk kelas maaf.

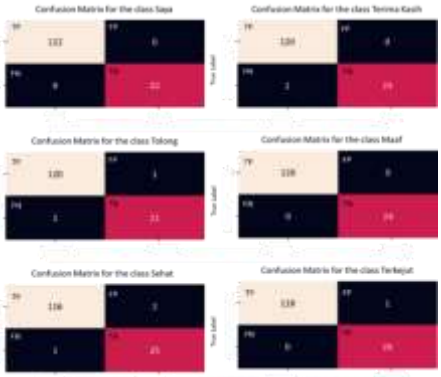
Tahap kedua yaitu melakukan preprocessing dan membangun fitur dari data yang telah dikumpulkan. Tahap ini dilakukan untuk menggabungkan seluruh nilai array dari urutan data yang tersedia sehingga menjadi suatu fitur pada setiap gerakan. Sehingga data akan memiliki 180 array yang memiliki 30 frame di setiap array dengan 1662 *keypoint* yang dideteksi. Setiap array didefinisikan satu gerakan sehingga label terdiri dari 180 nilai. Tahap selanjutnya adalah membagi data untuk *training* dan untuk *testing* menggunakan library sklearn yaitu train_test_split. Untuk data *testing* dibagi menjadi 20% dari banyak keseluruhan data sehingga untuk *training* menjadi 144 data array dan untuk *testing* 36 data array dari keseluruhan yaitu 180 data array.

2.5. Long Short Term Memory (LSTM)

Tahap ini melakukan perancangan arsitektur *neural network* yaitu menggunakan *Long Short Term Memory* (LSTM). Lapisan LSTM dapat secara efisien mengklasifikasikan data yang terungkap dalam waktu. LSTM terdiri dari sel dengan input gate, output gate dan forget gate. Dengan unit tersebut, LSTM dapat menghapus status temporal dan memungkinkan yang baru untuk disimpan atau mempertahankan status yang sama dari sebelumnya [15]. Hal ini sangat penting untuk mengklasifikasikan gerakan dinamis melalui perubahan koordinat sehingga penelitian ini menggunakan lapisan LSTM untuk modelnya.

Pembuatan arsitektur neural network dilakukan dengan bahasa pemrograman python menggunakan TensorFlow versi 2.4.1 dan library tensorflow-keras-layers untuk dapat mengimport LSTM layer dan dense layer. Pembuatan arsitektur dilakukan tahap demi tahap secara sequential dengan menggunakan "add" untuk menambahkan layer per layer. Arsitektur terdiri dari 6 layer yaitu 3 lapisan LSTM dan 3 lapisan dense layer. Lapisan yang pertama adalah LSTM layer yang memiliki 64 unit dan terdiri dari 3 argumen yaitu return sequences bernilai true artinya akan mengembalikan nilai sequences karena layer selanjutnya masih membutuhkannya. Memiliki ReLu Activation dan input shape (30,1662) yang artinya setiap prediksi akan memiliki 30 frame dengan 1662 *keypoint* yang terdeteksi. Untuk lapisan kedua dan lapisan ketiga masing-masing berisi 128 dan 64 unit dan memiliki ReLu Activation. Lapisan ketiga memiliki return_sequences bernilai false artinya nilai sequences tidak dikembalikan lagi karena layer selanjutnya yaitu dense layer tidak memerlukannya. Selanjutnya terdapat 3 lapisan dense layer atau fully connected neural network neuron dengan masing-masing berisi 64, 32 dan action shape[0] artinya akan mengembalikan semua nilai yang berada dalam model sehingga akan memprediksi gerakan.

Pada Gambar 7 dapat dilihat model memiliki total parameter 596.774. Jumlah parameter tergantung pada pendekatan fitur menggunakan 1662 *keypoint* landmark. Nilai neuron yang didapatkan merupakan hasil eksperimen yang memiliki kinerja terbaik sebelum memilih model yang disajikan.



Gambar 8. Nilai confusion Matrix dari 6 kelas gerakan

	precision	recall	f1-score	support
Saya	1.00	1.00	1.00	27
Terima Kasih	1.00	0.91	0.95	23
Tolong	0.92	0.92	0.92	25
Sehat	0.91	0.91	0.91	23
Terkejut	0.96	1.00	0.98	22
Maaf	0.96	1.00	0.98	24
accuracy			0.96	144
macro avg	0.96	0.96	0.96	144
weighted avg	0.96	0.96	0.96	144

Gambar 9. Hasil akurasi 6 kelas gerakan

2.6. Training

Pada tahap ini melakukan pelatihan terhadap data menggunakan arsitektur LSTM yang sudah dibuat. Ukuran sampel *training* adalah 144 dan ukuran sampel validasi adalah 36. Untuk mencegah overfitting maka digunakan metode Early Stopping. Metode early stopping adalah menghentikan *training* ketika nilai akurasi sudah bagus [16]. Maka, *training* berlangsung selama 42 epoch dengan menggunakan metode early stopping. Hasil pelatihan model *training* dan label data selanjutnya disimpan ke dalam file dengan ekstensi .h5 untuk digunakan dalam proses testing model.

3. Hasil dan Pembahasan

Tahap ini dilakukan untuk mengevaluasi kinerja model dalam melakukan prediksi pada sistem pengenalan gerakan tangan dinamis dan kinerja sistem jika diuji secara *real-time*. Tahap ini terdiri dari pengujian performa model yang dilakukan dengan menganalisis nilai confusion matrix yang diperoleh dari model. Selanjutnya menganalisis performa sistem dalam mengenali gerakan dengan pengujian secara real-time.

3.1. Evaluasi Performa Model

Tahap ini mengevaluasi kinerja model dalam melakukan prediksi pada sistem pengenalan gerakan tangan dinamis menggunakan library sklearn.metrics dalam pembuatan confusion matrix dan menghasilkan akurasi dari model yang telah di *training*. *Confusion matrix* berfungsi untuk membandingkan nilai aktual dengan nilai prediksi yang

Tabel 1. Tabel Pengujian secara Real-Time

No	Gerakan	Keluaran
1	Saya Terkejut Sehat Terima kasih	Saya Terkejut Terima Kasih Sehat Terima Kasih
2	Tolong Terkejut Sehat Saya	Terima Kasih Sehat Tolong Terkejut Saya Terima Kasih Sehat Tolong Terkejut Saya
3	Terima kasih Maaf Saya Tolong	Terima Kasih Saya Maaf Saya Terima Kasih Sehat Tolong
4	Terkejut Saya Sehat Terima kasih	Terkejut Saya Terima Kasih Sehat Terima Kasih
5	Saya Terkejut Sehat Tolong	Saya Terkejut Terima Kasih Sehat Tolong

berfungsi untuk mengevaluasi apa yang terdeteksi sebagai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Normalisasi matriks kebingungan memberikan informasi lebih lanjut tentang hubungan antara kelas dan tipe kesalahan klasifikasi [17]. Nilai ini digunakan untuk mendapatkan hasil akurasi dari model. Berikut Gambar 10 merupakan nilai *confusion matrix* dari setiap kelas.

Pada Gambar 8 dapat dilihat hasil confusion matrix yang menunjukkan performa model dalam memprediksi gerakan pada setiap kelas. Seperti untuk kelas "Saya", model memiliki nilai True Positive 120 menunjukkan performa model sudah bagus yaitu dapat memprediksi gerakan "Saya" secara tepat walaupun model memiliki nilai True Negative 22 yaitu model masih kemungkinan mendeteksi gerakan lain menjadi kelas gerakan "Saya". Secara keseluruhan dari nilai confusion matrix, performa model sudah sangat baik. Berikut Gambar 9 menunjukkan hasil akurasi pengenalan untuk 6 gerakan pada kata-kata dalam SIBI (Sistem Bahasa Isyarat Indonesia).

Pada Gambar 9 dapat dilihat nilai akurasi serta nilai performance metrics dari model yaitu precision, recall, f1-score, dan support pada setiap kelas. Nilai akurasi adalah perbandingan antara data yang terklasifikasi benar dengan jumlah keseluruhan data sehingga menunjukkan seberapa akurat model dalam mengklasifikasi kelas gerakan [18]. Akurasi yang didapatkan adalah 96% artinya bahwa model sudah akurat dalam mengklasifikasi kelas gerakan. Pada setiap kelas memiliki nilai precision, recall, f1-score dan support yang sangat baik dengan rata-rata 96%.

Nilai precision menggambarkan kecocokan antara permintaan informasi dengan jawaban terhadap informasi tersebut yang diberikan oleh model [19]. Nilai recall menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. Nilai F1 Score menggambarkan perbandingan rata-rata dari nilai precision dan recall sebagai acuan performa algoritma.

Gambar 10. Gerakan yang dideteksi pada sistem pengenalan gerakan tangan dinamis

Gambar 11. Gambaran kesalahan deteksi karena interaksi tangan

Gambar 12. Gambaran kesalahan deteksi karena interaksi tangan-wajah

Sehingga dapat disimpulkan dari hasil keseluruhan nilai bahwa model sudah akurat dalam pengujian ini.

3.2. Pengujian secara Real-Time

Setelah model dilatih maka tahap selanjutnya adalah melakukan testing secara *real-time* untuk melihat kinerja dari sistem dalam pengenalan gerakan tangan secara dinamis. Sistem yang dibuat hanya dapat mengenali gerakan secara individual dan tidak mampu mengenali kalimat. Sistem ini berjalan rata-rata selama 30 frame per detik (0,033 detik per setiap *frame*). Pada Tabel 1 diuji beberapa contoh keluaran sistem untuk setiap masukan.

Pada Tabel 1 dapat dilihat bahwa seluruh gerakan dapat terdeteksi dengan baik namun selalu ada beberapa gerakan yang salah sebelum terdeteksi. Hal ini dikarenakan sistem mendeteksi beberapa koordinat yang memiliki kemiripan dengan gerakan lain. Sehingga saat sistem belum mencapai 30 frame, maka sistem menampilkan output yang mendekati nilai koordinat terdekat. Namun, setelah 30 frame terdeteksi, maka sistem menampilkan output terakhir yang terdeteksi. Berikut Gambar 10 merupakan 6 kelas gerakan yang dideteksi oleh sistem pengenalan gerakan tangan dinamis.

3.3. Pembahasan

Pengenalan gerakan melalui deteksi pergerakan koordinat *keypoint* ini memungkinkan untuk menguji sistem secara kualitatif menggunakan visualisasi. Pengujian secara manual dilakukan untuk menemukan kasus kegagalan model dan mengkategorikannya menjadi dua kategori yaitu interaksi tangan dan interaksi tangan-wajah.Ketika ada interaksi antara kedua tangan

Gambar 13. Kesalahan deteksi gerakan sehat dan terima kasih

Gambar 14. Gerakan "Maaf" yang mengalami kesalahan deteksi

atau satu tangan yang menghalangi tangan yang lain dari pandangan kamera, sistem sering gagal memperkirakan pose salah satu tangan. Gambaran kesalahan dapat dilihat pada Gambar 11.

Pada Gambar 11 adalah gambaran kesalahan untuk mendeteksi salah satu tangan yaitu tangan kanan karena tumpang tindih dengan tangan kiri. Sehingga *keypoint* untuk tangan kanan justru terdeteksi di belakang tangan kiri. Keterangan warna pada Gambar 13 adalah warna merah untuk pose, warna hijau untuk tangan kiri, warna biru untuk tangan kanan dan warna merah tua untuk wajah. Ketika ada interaksi antara tangan dan wajah atau tangan tumpang tindih dengan wajah dari sudut kamera, sistem sering gagal memperkirakan pose wajah yang berinteraksi. Gambaran kesalahan dapat dilihat pada Gambar 12.

Kasus-kasus interaksi yang terlewatkan antara bagian-bagian tubuh yang berbeda ini sering kali kehilangan esensi *keypoint*, di mana saat interaksi terdapat *keypoint* yang tertutup kamera. Esensi *keypoint* merupakan proses yang memungkinkan sistem visi mesin untuk mendeteksi daerah yang paling penting dan informatif dalam gambar [20]. Sehingga, esensi *keypoint* ini akan menjadi pembeda utama fitur untuk gerakan ini sehingga jika terdapat *keypoint* yang tertutup maka dapat menghalangi kemampuan model untuk mengekstrak informasi dari gerakan tersebut. Setelah dilakukan pengujian secara *real-time*, sistem memiliki beberapa kesulitan untuk membedakan beberapa gerakan yang mirip, seperti pada ada gerakan "Sehat" sering terdeteksi

menjadi gerakan "Terima Kasih" atau sebaliknya seperti pada Gambar 13.

Pada Gambar 13 merupakan salah satu kesalahan gerakan karena beberapa posisi koordinat tangan pada kedua gerakan yang mirip di awal frame sehingga sering terjadi kesalahan deteksi. Hal ini juga terjadi pada gerakan "Maaf" yang sering tidak terdeteksi dan justru mendeteksi gerakan "Saya". Pada Tabel 1 dapat dilihat bahwa Maaf tidak terdeteksi 2 kali dan justru mendeteksi gerakan "Saya". Berikut Gambar 14 merupakan gambaran gerakan "Maaf" tidak terdeteksi dan justru mendeteksi gerakan "Saya".

4. Kesimpulan

Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa hasil implementasi Sistem Pengenalan Bahasa Isyarat secara *real-time* dengan memanfaatkan perubahan koordinat menggunakan *MediaPipe Holistic* dan metode *Long Short Term Memory* (LSTM) diperoleh nilai akurasi model sebesar 96%. Dengan melakukan pengujian secara *real-time* terdapat satu huruf yang belum konsisten benar saat dikenali yaitu gerakan "Maaf" karena gerakan tersebut perubahan koordinatnya memiliki tingkat kemiripan tinggi dengan gerakan "Saya". Pengembangan lebih lanjut dapat dilakukan untuk menyempurnakan penelitian sistem pengenalan bahasa isyarat untuk pengenalan kalimat. Sistem ini hanya dapat mengenali beberapa kata dalam bahasa isyarat, sedangkan bahasa isyarat memiliki pola kalimat yang sangat cepat gerakannya sehingga sistem yang dibuat harus memiliki kecepatan deteksi yang signifikan. Diharapkan penelitian dikembangkan lagi sehingga kekurangan sistem dapat mengenali pergerakan kalimat serta mengembangkan sistem agar dapat huruf yang memiliki kemiripan yang tinggi dapat dideteksi dengan baik.

Daftar Pustaka

[1] H. Luqman, E. S. M. El-Alfy, and G. M. BinMakhashen, "Joint Space Representation And Recognition Of Sign Language Fingerspelling Using Gabor Filter And Convolutional Neural Network," *Multimed. Tools Appl.*, vol. 80, no. 7, pp. 10213–10234, 2021, doi: 10.1007/s11042-020-09994-0.

[2] G. Malathi, "Indian Sign Language Using Holistic Pose Detection," vol. 32, no. 3, pp. 19746–19752.

[3] C. K. M. Lee, K. K. H. Ng, C. H. Chen, H. C. W. Lau, S. Y. Chung, and T. Tsoi, "American Sign Language Recognition And Training Method With Recurrent Neural Network," *Expert Syst. Appl.*, vol. 167, no. April, 2021, doi: 10.1016/j.eswa.2020.114403.

[4] Y. Che, C. B. Sivaparthipan, and J. Alfred Daniel, "Human-Computer Interaction on IoT-Based College Physical Education," *Arab. J. Sci. Eng.*, no. 0123456789, 2021, doi: 10.1007/s13369-021-05895-y.

[5] A. Rahagiyanto, "Identifikasi Ekstraksi Fitur untuk Gerakan Tangan dalam Bahasa Isyarat (SIBI) Menggunakan Sensor MYO Armband," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 19, no. 1, pp. 127–137, 2019, doi: 10.30812/matrik.v19i1.510.

[6] 沙洁, "A Review of Vision Based Dynamic Hand Gestures Recognition," *Comput. Sci. Appl.*, vol. 10, no. 05, pp. 990–1001, 2020, doi: 10.12677/csa.2020.105102.

[7] A. Moryossef, I. Tsochantaridis, R. Aharoni, S. Ebling, and S. Narayanan, "Real-Time Sign Language Detection using Human Pose Estimation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12536 LNCS, pp. 237–248, 2020, doi: 10.1007/978-3-030-66096-3_17.

[8] Y. Wu *et al.*, "A Computer Vision-Based Yoga Pose Grading Approach Using Contrastive Skeleton Feature Representations," *Healthc.*, vol. 10, no. 1, pp. 1–12, 2022, doi: 10.3390/healthcare10010036.

[9] A. Halder and A. Tayade, "International Journal of Research Publication and Reviews Real-time Vernacular Sign Language Recognition Using MediaPipe And Machine Learning," no. 2, pp. 9–17, 2021.

[10] M. Mohammed and K. Ibrahim, "Eye Blinking For Command Generation Based on Deep Learning," vol. 13, no. 4, pp. 22–29, 2022.

[11] G. Saggio, P. Cavallo, M. Ricci, V. Errico, J. Zea, and M. E. Benalcázar, "Sign Language Recognition Using Wearable Electronics: Implementing K-nearest Neighbors With Dynamic Time Warping And Convolutional Neural Network Algorithms," *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–14, 2020, doi: 10.3390/s20143879.

[12] Y. Liao, P. Xiong, W. Min, W. Min, and J. Lu, "Dynamic Sign Language Recognition Based on Video Sequence With BLSTM-3D Residual Networks," *IEEE Access*, vol. 7, no. c, pp. 38044–38054, 2019, doi: 10.1109/ACCESS.2019.2904749.

[13] S. Shin and W. Y. Kim, "Skeleton-Based Dynamic Hand Gesture Recognition Using a Part-Based GRU-RNN for Gesture-Based Interface," *IEEE Access*, vol. 8, pp. 50236–50243, 2020, doi: 10.1109/ACCESS.2020.2980128.

[14] M. H. Ismail, S. A. Dawwd, and F. H. Ali, "Static Hand Gesture Recognition Of Arabic Sign Language By Using Deep CNNs," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 24, no. 1, pp. 178–188, 2021, doi: 10.11591/ijeecs.v24.i1.pp178-188.

[15] S. Mittal and O. P. Sangwan, "Big Data Analytics

Using Deep LSTM Networks: A Case Study For Weather Prediction," *Adv. Sci. Technol. Eng. Syst.*, vol. 5, no. 2, pp. 133–137, 2020, doi: 10.25046/aj050217.

[16] Y. Bin Wang, Z. H. You, S. Yang, H. C. Yi, Z. H. Chen, and K. Zheng, "A deep learning-based method for drug-target interaction prediction based on long short-term memory neural network," *BMC Med. Inform. Decis. Mak.*, vol. 20, Mar. 2020, doi: 10.1186/s12911-020-1052-0.

[17] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label Classifier Performance Evaluation with Confusion Matrix," pp. 01–14, 2020, doi: 10.5121/csit.2020.100801.

[18] R. AGUSTINA, R. MAGDALENA, and N. K. C. PRATIWI, "Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 2, p. 446, 2022, doi: 10.26760/elkomika.v10i2.446.

[19] D. Cahyanti, A. Rahmayani, and S. A. Husniar, "Analisis performa metode Knn pada Dataset pasien pengidap Kanker Payudara," *Indones. J. Data Sci.*, vol. 1, no. 2, pp. 39–43, 2020, doi: 10.33096/ijodas.v1i2.13.

[20] V. Mousavi, M. Varshosaz, and F. Remondino, "Using information content to select keypoints for uav image matching," *Remote Sens.*, vol. 13, no. 7, 2021, doi: 10.3390/rs13071302.
