

# Pengenalan Kode Semaphore Berbasis Raspberry Pi Menggunakan Metode CNN

1<sup>st</sup> Muhammad Alfarezi

Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

Muhammadbayualfarezi@student.telkomuniversity.ac.id

2<sup>nd</sup> Fazmah Arif Yulianto

Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

fazmaharif@telkomuniversity.ac.id

3<sup>rd</sup> Andrian Rakhmatsyah

Fakultas Informatika  
Universitas Telkom  
Bandung, Indonesia

kangadrian@telkomuniversity.ac.id

**Abstrak** - Sandi Semaphore merupakan sebuah metode komunikasi dengan cara mengirim sinyal menggunakan alat sederhana seperti bendera dan gerak tubuh. Namun dalam proses pembelajarannya, sandi semaphore mengalami kendala terkait minat dan ketertarikan siswa dalam mempelajari sandi tersebut. Penelitian ini bertujuan untuk menerapkan sebuah model Semaphore Code Recognition demi meningkatkan minat serta ketertarikan siswa dan masyarakat dalam mempelajari sandi semaphore. Sebuah model Semaphore Code Recognition menggunakan Raspberry Pi mikrokontroler, Convolutional Neural Network (CNN), OpenCV (Open Source Computer Vision) dan Library MediaPipe ditujukan sebagai sebuah media sarana baru bagi siswa dalam mempelajari sandi semaphore. Pengambilan gambar pada sistem ini tampak dari depan menggunakan Webcam yang terhubung ke Raspberry Pi. Tujuan dengan dibangunnya model ini dapat meningkatkan minat belajar bagi siswa

**Kata kunci**— Bendera Semaphore, Raspberry Pi, CNN, OpenCV, mikrokontroler

## I. PENDAHULUAN

Sebagai makhluk sosial, Manusia melakukan komunikasi antar sesama manusia, bahasa verbal adalah komunikasi yang biasa dilakukan oleh manusia dengan menggunakan kata-kata namun pada saat tertentu menggunakan bahasa verbal sangat tidak efektif sehingga digunakan bahasa nonverbal. Sandi *Semaphore* merupakan salah satu komunikasi nonverbal yaitu komunikasi yang dilakukan tanpa kata-kata. Sandi *semaphore* yang digunakan ada 27 sandi yaitu terdiri dari 26 alfabet [1]. Beberapa penerapan Sandi *semaphore* yang diterapkan saat ini salah satunya ada pada kemiliteran angkatan laut, dimana pada korps angkatan laut menggunakan isyarat semaphore sebagai keterampilan dalam mengirim, menerima, dan membaca berita komunikasi melalui isyarat semaphore. Isyarat semaphore juga dimaksudkan untuk meningkatkan profesionalisme, kemampuan, dan keterampilan prajurit korps angkatan laut.

Namun pada penerapannya saat ini, tidak banyak masyarakat serta siswa yang mempelajari sandi semaphore. Hal ini disebabkan rendahnya minat dan ketertarikan siswa dalam mempelajari sandi semaphore, media pembelajaran yang digunakan saat ini adalah hanya berupa gambar yang menjelaskan tentang sandi semaphore terkesan terlalu polos

dan membuat siswa menjadi jenuh. Oleh karena itu dibutuhkan sebuah media pembelajaran baru yang lebih interaktif untuk dapat menarik minat siswa dalam mempelajari sandi semaphore.

Pembangunan sistem Pengenalan Kode Semaphore Berbasis Raspberry Pi Menggunakan Metode CNN dibangun sebagai media pembelajaran baru dalam mempelajari sandi semaphore dengan menerapkan mikrokontroler Raspberry Pi dan metode CNN untuk membangun *Semaphore Code Recognition*. Penggunaan *Hand pose* adalah salah satu solusi memenuhi kebutuhan dalam manusia dan komputer (*Computer Human Interaction*) lebih cepat dan sejalan dengan fungsi anggota tubuh manusia tangan memiliki beragam pose dan setiap pose memiliki maksud dan makna tersendiri sesuai dengan kesepakatan yang melakukan komunikasi. *Hand pose* yang digunakan dalam tugas akhir ini untuk membantu menerjemahkan sandi *semaphore* menjadi huruf [2]. *Convolutional Neural Network* (CNN) adalah algoritma pembelajaran mendalam/*Deep Learning*. Kapasitas CNN dapat dikontrol dengan memvariasikan kedalaman dan luasnya, dan CNN juga membuat asumsi yang kuat dan sebagian besar benar tentang properti gambar [3]. Dalam beberapa tahun terakhir, CNN telah menjadi terancang untuk pengenalan objek computer vision. CNN memiliki potensi tinggi dalam deteksi objek dan segmentasi gambar [4].

Tugas akhir ini dibangun pada environment Raspberry Pi. Dimana Raspberry Pi adalah sebuah komputer kecil/microcontroller yang memiliki keunggulan dalam portabilitas dan konsumsi daya yang rendah. Hal ini memungkinkan sistem pengenalan kode semaphore untuk diimplementasikan di lapangan atau di lokasi yang jauh dari sumber daya listrik tetap. Disisi lain penggunaan aplikasi berbasis mobile memerlukan

Sebuah penelitian yang dilakukan oleh Q. Zhao, Y. Li, N. Yang, Y. Yang, and M. Zhu dengan judul “A convolutional neural network approach for semaphore flag signaling recognition” mengusulkan pendekatan pengenalan untuk Semaphore flag signaling (SFS) dengan menggunakan neural network convolutional yang ditingkatkan (CNN) untuk mengklasifikasi (SFS). Metode Improve CNN yang dipakai menghasilkan hasil dengan akurasi 99,95% dengan kekurangan sample error yang disebabkan oleh efek

pretreatment (efek segmentasi karakter tidak ideal) dan efek dari noise.

Dengan dibangunnya sistem model pada penelitian ini diharapkan dapat membantu meningkatkan minat belajar siswa dalam mempelajari sandi semaphore

II. SISTEM YANG DIBANGUN

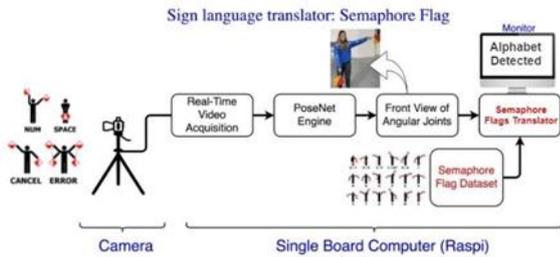


FIGURE 1'

Flowchart sistem yang dibangun

Pada Figure 1 menunjukkan bahwa proses pengambilan data dilakukan menggunakan Camera, lalu dilanjutkan dengan Real-Time Video Acquisition untuk memperoleh citra gambar secara real-time. Kemudian citra yang sudah didapatkan dimasukkan kedalam PoseNet Engine CNN untuk dilakukan prediksi sehingga mendapatkan hasil Front View of Angular Joints yang nantinya akan diterjemahkan menggunakan Sign Language Semaphore Flag Translator. Dilakukan pengujian terhadap model CNN yang dibangun yang akan diukur menggunakan matrik evaluasi Precision, Recall, F1-Score dan Akurasi untuk mengukur performansi model yang telah di bangun.

A. Real-Time Video Acquisition

Objek yang merupakan data gambar akan diambil secara realtime menggunakan Video Capture berupa kamera yang dipasang pada Raspberry PI. Data Video yang berupa gambar video realtime kemudian diolah menggunakan akuisisi data. yang selanjutnya akan diproses oleh Posenet Engine. Dalam proses capturing gambar dilakukan dengan menggunakan webcam/kamera Bahasa yang akan digunakan adalah python yang diaplikasikan menggunakan library OpenCV dan MediaPipe. Objek yang merupakan input data yang akan diolah, Kemudian dilakukan proses akuisisi untuk dapat menghasilkan gambar dan video realtime

B. MediaPipe

MediaPipe adalah sebuah library Python open-source yang dikembangkan oleh Google. Library ini menyediakan berbagai alat untuk analisis visual dalam waktu nyata, seperti deteksi pose tubuh manusia, deteksi wajah, pelacakan tangan, dan masih banyak lagi. MediaPipe memungkinkan pengembang untuk dengan mudah mengintegrasikan analisis visual yang kompleks ke dalam aplikasi mereka tanpa harus membangun semua komponennya dari awal.

MediaPipe diterapkan untuk mendeteksi postur tubuh selama proses training. MediaPipe Pose adalah solusi Machine Learning yang dapat mendeteksi pose tubuh secara presisi menggunakan 33 landmark. MediaPipe Pose menggunakan penelitian BlazePose [5].

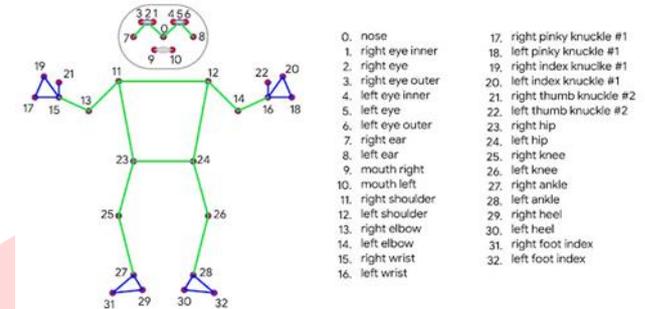
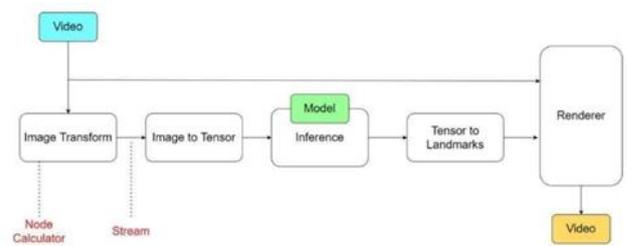


FIGURE 1  
MediaPipe Graph Solution

Pada penelitian ini, tidak seluruh landmarks hasil MediaPipe Pose Landmark yang akan digunakan. Hanya beberapa landmarks terkait pose semaphore saja yang akan digunakan. Beberapa landmark terpilih yang digunakan pada penelitian ini adalah sebagai berikut : [ Landmark 15, Landmark 13, Landmark 11, Landmark 12, Landmark 14, Landmark 16, Landmark 23, dan Landmark 24 ]. Daftar Landmark tersebut dipilih karena berkaitan langsung dengan pose semaphore dimana pose semaphore hanya membutuhkan dan merekam fitur lengan serta badan untuk dapat memperagakan gestur nya

C. Gesture Recognition

Sejarah Hand Gesture Recognition untuk kontrol komputer dimulai dengan penemuan antarmuka kontrol berbasis sarung tangan. Para peneliti menyadari bahwa gerakan yang terinspirasi oleh bahasa isyarat dapat digunakan untuk menawarkan perintah sederhana untuk antarmuka komputer. Penemuan ini berangsur-angsur berkembang dengan pengembangan accelerometer yang jauh lebih akurat, kamera inframerah dan bahkan sensor serat optik yang dapat ditekan (goniometer optik). Beberapa perkembangan dalam sistem berbasis sarung tangan akhirnya menawarkan kemampuan untuk mewujudkan pengenalan berbasis computer vision tanpa sensor yang melekat pada sarung tangan. Yakni penggunaan sarung tangan berwarna atau sarung tangan yang menggunakan warna unik (color marker) untuk kemampuan pelacakan jari. Selama 25 tahun terakhir, evolusi ini telah menghasilkan banyak banyak produk sukses yang menawarkan koneksi nirkabel total dengan resistensi paling sedikit terhadap pemakainya [5].

Metodologi Hand Gesture Recognition secara umum bias dikelompokkan kedalam 2 kelompok pendekatan [6] yakni :

1. Pendekatan berbasis Visual (Vision Based Approach)
2. Pendekatan berbasis Sensor (Sensor Based Approach)

Data gambar diambil menggunakan berbagai optic device dan kamera. Metode ini berfokus pada gambar yang diambil dari gesture, mengekstraksi fitur gambar tersebut dan mengenalnya. Color marker adalah salah satu metode pendekatan ini. Terdapat beberapa batasan dalam metode color marker. Karena itu gerakan yang dilakukan

menggunakan tangan kosong lebih disukai daripada menggunakan color marker. Digunakan berbagai algoritma deteksi kulit dalam sistem pengenalan gerakan tangan kosong untuk deteksi tangan. Masalah yang dihadapi oleh jenis sistem ini adalah kondisi latar belakang yang tidak tetap dan dinamis sehingga mempengaruhi hasil pengenalan. Selain itu sistem ini lebih kompleks dan mahal karena menggunakan perangkat kamera untuk pengenalan dan resource komputasi yang besar untuk memproses gambar gesture tersebut.

Sensor Based Approach memanfaatkan berbagai sensor untuk mengumpulkan data gerakan yang dilakukan. Data ini kemudian dianalisis dan kesimpulan diambil sesuai dengan model pengenalan. Dalam hal pengenalan isyarat tangan, berbagai sensor ditempatkan di tangan dan ketika tangan melakukan gerakan apa pun, data dicatat dan dianalisis lebih lanjut. Penemuan Data glove pertama dilakukan pada tahun 1977 [6].

D. Matrix Evaluation

Matrik Evaluasi merupakan instrument yang digunakan untuk melakukan evaluasi pada performa dari suatu model klasifikasi yang dihasilkan sebelumnya. Hasil yang telah diprediksi akan dibandingkan dengan data yang sebenarnya. Berikut adalah beberapa *Evaluation Matrix* yang digunakan pada penelitian ini :

1. Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

2. Precision

$$Precision = \frac{TP}{TP + FP}$$

3. Recall

$$Recall = \frac{TP}{TP + FN}$$

III. HASIL DAN EKPRERIMEN

A. Dataset Dan Preprocessing

hasil penelitian yang dimulai dari deskripsi yang berhubungan dengan data penelitian (meliputi gambaran umum citra gambar, variabel penelitian dan uji kualitas). Hasil pengujian dan pembahasan terhadap uji hipotesis yang diuji dengan menggunakan program pengolahan data serta metode klasifikasi data.

B. Analisa Data

Pengolahan data yang digunakan pada penelitian ini adalah, berupa data citra gambar yang dikumpulkan dari beberapa relawan melalui survey yang dilakukan oleh peneliti secara langsung serta beberapa data yang didapatkan dari website <https://data.mendeley.com/datasets/tc5tnchrs2/1>. Ini adalah Dataset Sinyal Bendera Semaphore yang disiapkan oleh Midshipmen Akademi Angkatan Laut. Ini mencakup 20

contoh untuk setiap bendera semaphore dalam alfabet, diambil oleh 4 orang yang berbeda.

TABLE 1  
Ekspreimen

| No | Tujuan Eksperimen / Simulasi                       | Parameter yang diukur / diamati                      |
|----|----------------------------------------------------|------------------------------------------------------|
| 1  | Mendapatkan Hasil Input Pose                       | Video Input = diamati                                |
| 2  | Mendapatkan Hasil Output Pose (Feature Extraction) | Hand Pose Landmark = diukur                          |
| 3  | Menguji Model Classifier terhadap Output Pose      | Akurasi = diukur<br>Probabilitas = diukur            |
| 4  | Menguji Konektivitas Kamera Raspberry Pi           | PiCamera = diamati dapat menangkap gambar atau tidak |
| 5  | Menguji Pose Detection pada Model CNN              | CNN Model Prediction = prediksi sesuai atau tidak    |

Data citra diambil menggunakan library OpenCV yang kemudian dilakukan ekstraksi fitur melalui library Mediapipe untuk menghasilkan Hand Pose Landmark Dalam bentuk tabel koordinat Hand Pose Landmark yang disimpan ke dalam file CSV (CommaSeparated Values). Total berjumlah 2204 gambar. Kemudian dataset dilakukan proses data split menjadi data latih dan data uji dengan rasio data sebesar 70% sebagai data latih dan 30% sebagai data uji.

TABLE 2  
Ekstraksi Fitur

|      | 0        | 1        | 2        | ... | 96       | 97       | 98       | 99  |
|------|----------|----------|----------|-----|----------|----------|----------|-----|
| 0    | 0.527358 | 0.227569 | 0.999137 | ... | 0.438396 | 1.032487 | 0.916136 | A   |
| 1    | 0.529825 | 0.268963 | 0.999221 | ... | 0.479991 | 1.032734 | 0.915032 | A   |
| 2    | 0.520153 | 0.378262 | 0.999296 | ... | 0.477915 | 1.081104 | 0.911794 | A   |
| 3    | 0.476760 | 0.280844 | 0.999295 | ... | 0.409470 | 0.939981 | 0.916475 | A   |
| 4    | 0.439209 | 0.277267 | 0.999321 | ... | 0.373389 | 0.930445 | 0.918191 | A   |
| ...  | ...      | ...      | ...      | ... | ...      | ...      | ...      | ... |
| 2199 | 0.465665 | 0.361464 | 0.999492 | ... | 0.420065 | 1.068548 | 0.898052 | Z   |
| 2200 | 0.516227 | 0.407231 | 0.998066 | ... | 0.537574 | 1.070643 | 0.856574 | Z   |
| 2201 | 0.441961 | 0.399586 | 0.998193 | ... | 0.427602 | 1.003137 | 0.868906 | Z   |
| 2202 | 0.446164 | 0.366405 | 0.998369 | ... | 0.432307 | 1.073727 | 0.863123 | Z   |
| 2203 | 0.449977 | 0.368886 | 0.998507 | ... | 0.425405 | 1.013519 | 0.874375 | Z   |

2204 rows x 100 columns

C. DATASET SPLIT

Dataset yang sudah Untuk menguji Model yang dibangun dapat menggunakan uji performansi klasifikasi yang dilakukan dengan melakukan prediksi data untuk mengamati hasil akurasi dari hasil prediksi terhadap input citra terhadap model CNN yang dibangun dengan mengamati hasil Classification Report dari model. Model CNN yang digunakan pada penelitian ini dilatih sebanyak 10 iterasi. dengan ukuran batch\_size data pada data training sebesar 32. Berikut adalah hasil yang didapatkan

```

Epoch 1/10 [-----] - 14s 17ms/step - loss: 3.2590 - accuracy: 0.0612 - val_loss: 3.2249 - val_accuracy: 0.1125
Epoch 2/10 [-----] - 0s 6ms/step - loss: 2.9688 - accuracy: 0.1616 - val_loss: 2.4866 - val_accuracy: 0.3040
Epoch 3/10 [-----] - 0s 6ms/step - loss: 1.8463 - accuracy: 0.4326 - val_loss: 1.2941 - val_accuracy: 0.5280
Epoch 4/10 [-----] - 0s 6ms/step - loss: 1.0722 - accuracy: 0.6779 - val_loss: 0.7945 - val_accuracy: 0.7812
Epoch 5/10 [-----] - 0s 6ms/step - loss: 0.7770 - accuracy: 0.7915 - val_loss: 0.7387 - val_accuracy: 0.7872
Epoch 6/10 [-----] - 0s 6ms/step - loss: 0.6763 - accuracy: 0.8235 - val_loss: 0.6996 - val_accuracy: 0.7795
Epoch 7/10 [-----] - 0s 6ms/step - loss: 0.5833 - accuracy: 0.8593 - val_loss: 0.5751 - val_accuracy: 0.8281
Epoch 8/10 [-----] - 0s 6ms/step - loss: 0.5479 - accuracy: 0.8691 - val_loss: 0.4963 - val_accuracy: 0.8845
Epoch 9/10 [-----] - 0s 6ms/step - loss: 0.4588 - accuracy: 0.8951 - val_loss: 0.4520 - val_accuracy: 0.9164
Epoch 10/10 [-----] - 0s 6ms/step - loss: 0.3998 - accuracy: 0.9121 - val_loss: 0.4163 - val_accuracy: 0.9170
    
```

FIGURE 3 Hasil Training CNN



FIGURE 4 Gesture Landmark

dapat dilihat bahwa model hasil training dengan iterasi 10 epoch sudah mendapatkan besaran akurasi sebesar 91.21%. berdasarkan hasil uji tersebut dapat disimpulkan bahwa penggunaan 10 epoch sudah mencukupi dalam proses latih model, hal ini bertujuan untuk menghindari overfitting dalam proses training dengan jumlah iterasi epoch yang terlalu besar. 91.21%. berdasarkan hasil uji tersebut dapat disimpulkan bahwa penggunaan 10 epoch sudah mencukupi dalam

TABLE 1 Deskripsi Citra Hasil Prediksi

| Citra Gambar | Label Sebenarnya | Label Prediksi | Confidence Level |
|--------------|------------------|----------------|------------------|
|--------------|------------------|----------------|------------------|

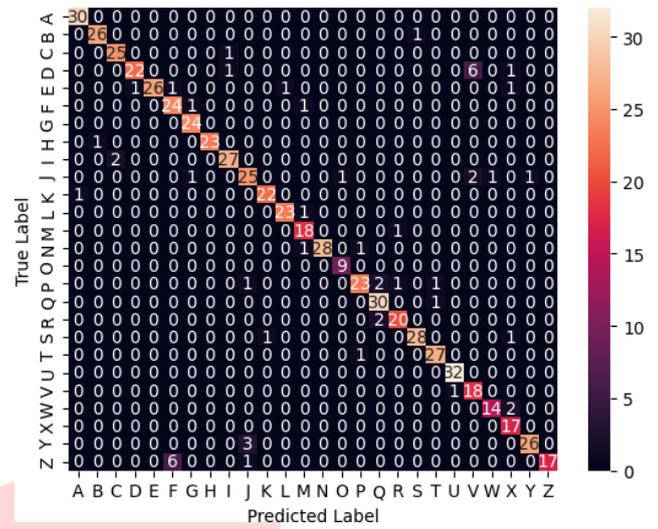
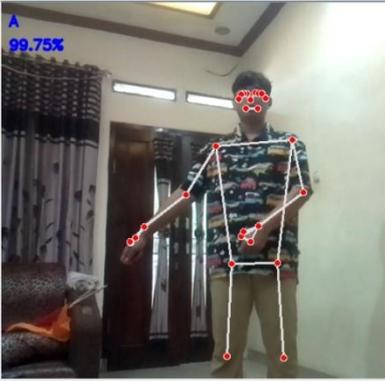
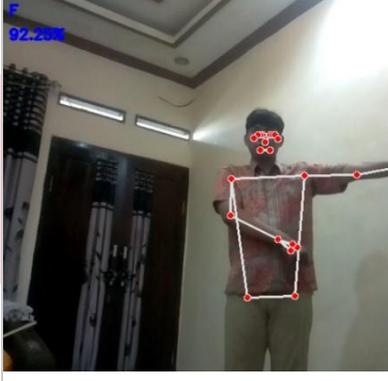
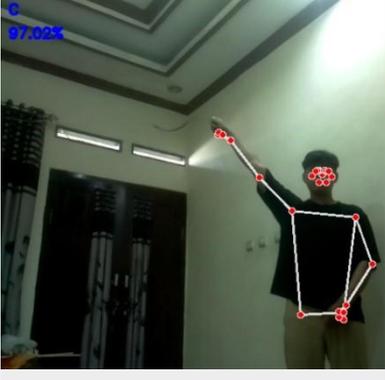
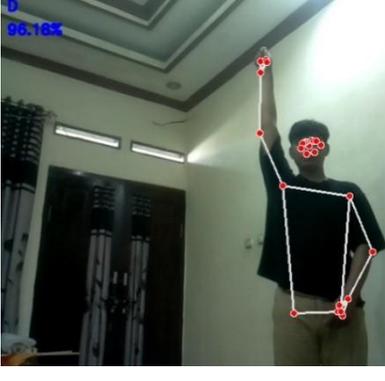
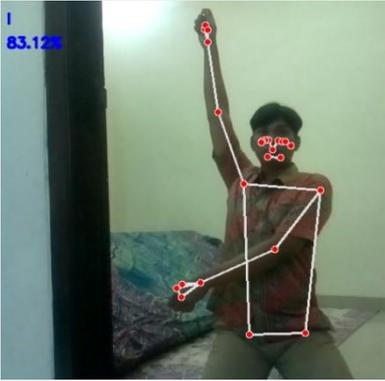
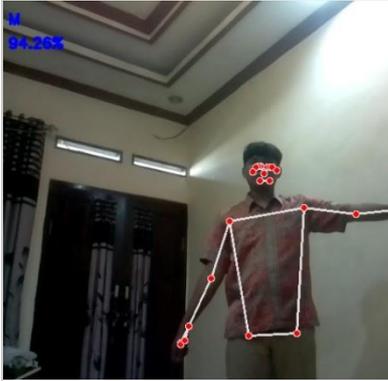
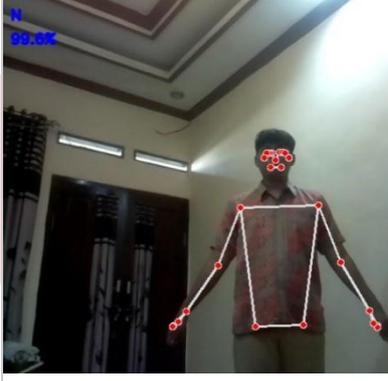
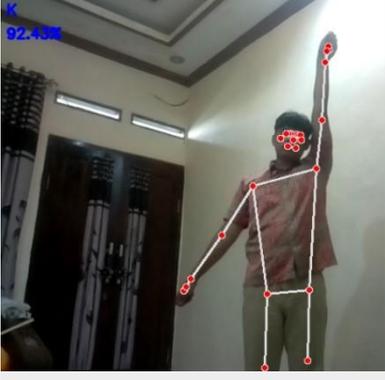
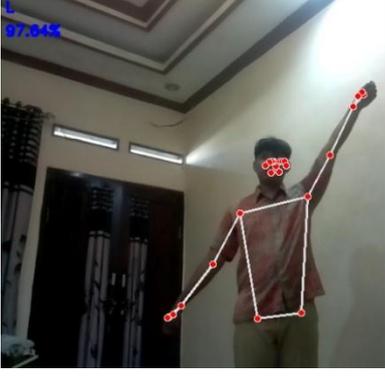


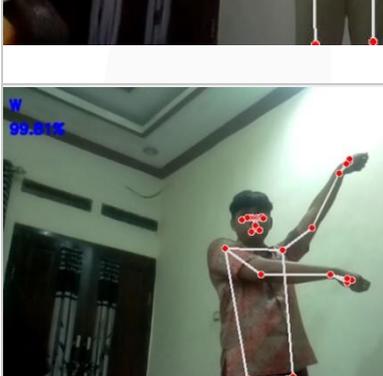
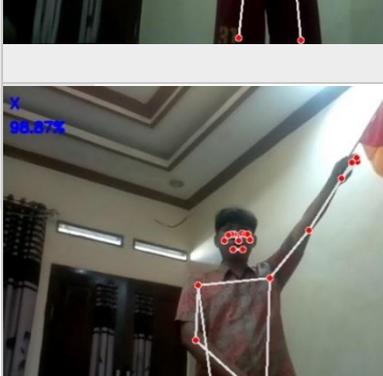
FIGURE 2 Confusion matrix CNN

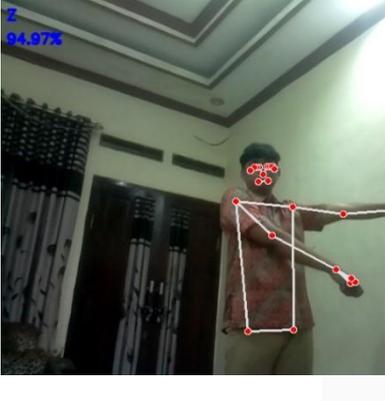
|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| A            | 0.97      | 1.00   | 0.98     | 30      |
| B            | 0.96      | 0.96   | 0.96     | 27      |
| C            | 0.93      | 0.96   | 0.94     | 26      |
| D            | 0.96      | 0.73   | 0.83     | 30      |
| E            | 1.00      | 0.87   | 0.93     | 30      |
| F            | 0.77      | 0.92   | 0.84     | 26      |
| G            | 0.92      | 1.00   | 0.96     | 24      |
| H            | 1.00      | 0.96   | 0.98     | 24      |
| I            | 0.93      | 0.93   | 0.93     | 29      |
| J            | 0.83      | 0.81   | 0.82     | 31      |
| K            | 0.96      | 0.96   | 0.96     | 23      |
| L            | 0.96      | 0.96   | 0.96     | 24      |
| M            | 0.86      | 0.95   | 0.90     | 19      |
| N            | 1.00      | 0.93   | 0.97     | 30      |
| O            | 0.90      | 1.00   | 0.95     | 9       |
| P            | 0.92      | 0.82   | 0.87     | 28      |
| Q            | 0.88      | 0.97   | 0.92     | 31      |
| R            | 0.91      | 0.91   | 0.91     | 22      |
| S            | 0.97      | 0.93   | 0.95     | 30      |
| T            | 0.93      | 0.96   | 0.95     | 28      |
| U            | 0.97      | 1.00   | 0.98     | 32      |
| V            | 0.69      | 0.95   | 0.80     | 19      |
| W            | 0.93      | 0.88   | 0.90     | 16      |
| X            | 0.77      | 1.00   | 0.87     | 17      |
| Y            | 0.96      | 0.90   | 0.93     | 29      |
| Z            | 1.00      | 0.71   | 0.83     | 24      |
| accuracy     |           |        | 0.92     | 658     |
| macro avg    | 0.92      | 0.92   | 0.92     | 658     |
| weighted avg | 0.92      | 0.92   | 0.92     | 658     |

FIGURE 5 Classification Report Model CNN

|                                                                                                         |   |   |        |                                                                                                          |   |   |        |
|---------------------------------------------------------------------------------------------------------|---|---|--------|----------------------------------------------------------------------------------------------------------|---|---|--------|
|  <p>A<br/>99.75%</p>   | A | A | 99.75% |  <p>E<br/>99.1%</p>    | E | E | 99.1%  |
|  <p>B<br/>98.53%</p>   | B | B | 99.53% |  <p>F<br/>92.25%</p>   | F | F | 92.25% |
|  <p>C<br/>97.02%</p>  | C | C | 97.02% |  <p>G<br/>90.61%</p>  | G | G | 90.61% |
|  <p>D<br/>96.18%</p> | D | D | 96.18% |  <p>H<br/>94.65%</p> | H | H | 94.65% |

|                                                                                                         |   |   |            |                                                                                                          |   |   |            |
|---------------------------------------------------------------------------------------------------------|---|---|------------|----------------------------------------------------------------------------------------------------------|---|---|------------|
|  <p>I<br/>83.12%</p>   | I | I | 83.12<br>% |  <p>M<br/>94.26%</p>   | M | M | 94.26<br>% |
|  <p>J<br/>100.0%</p>   | J | J | 100%       |  <p>N<br/>99.6%</p>    | N | N | 99.6%      |
|  <p>K<br/>92.43%</p>  | K | K | 92.43<br>% |  <p>O<br/>99.01%</p>  | O | O | 99.01<br>% |
|  <p>L<br/>97.64%</p> | L | L | 97.64<br>% |  <p>P<br/>99.91%</p> | P | P | 99.91<br>% |

|                                                                                                         |   |   |        |                                                                                                          |   |   |        |
|---------------------------------------------------------------------------------------------------------|---|---|--------|----------------------------------------------------------------------------------------------------------|---|---|--------|
|  <p>Q<br/>90.9%</p>    | Q | Q | 90.9%  |  <p>U<br/>92.22%</p>   | U | U | 92.22% |
|  <p>R<br/>84.87%</p>   | R | R | 84.8%  |  <p>V<br/>95.05%</p>   | V | V | 95.05% |
|  <p>S<br/>90.43%</p>  | S | S | 90.43% |  <p>W<br/>99.81%</p>  | W | W | 99.81% |
|  <p>T<br/>96.58%</p> | T | T | 96.58% |  <p>X<br/>98.87%</p> | X | X | 98.87% |

|                                                                                                        |   |   |        |
|--------------------------------------------------------------------------------------------------------|---|---|--------|
|  <p>Y<br/>91.81%</p>  | Y | Y | 91.81% |
|  <p>Z<br/>94.97%</p> | Z | Z | 94.97% |

#### D. PEMBAHASAN

Implementasi sistem Semaphore Code Recognition yang dibangun menggunakan Raspberry Pi dan PiCamera tidak mengalami masalah serta kendala dalam penerapannya. Terlepas dari kapasitas serta spesifikasi Raspberry Pi yang tergolong rendah tidak mempengaruhi pengujian dalam segi pemrosesan.

Sistem pengenalan Gesture menunjukkan hasil yang baik serta menjanjikan selama langkah pengujian, Tabel 2 menunjukkan kinerja hasil model yang sangat baik pada setiap label. Sistem dapat dengan benar dan baik mengenali pose tubuh yang berbeda-beda sesuai dengan result yang diharapkan, meskipun terdapat beberapa noise, distorsi, kondisi cahaya yang rendah, serta ukuran foto yang berbeda sistem tetap dapat melakukan prediksi dengan sangat baik. Selain itu, peneliti memperoleh respon yang cukup cepat pada proses deteksi serta klasifikasi pose tubuh pada pengujiannya

#### IV. KESIMPULAN

Pada penelitian yang telah dilakukan, peneliti memperkenalkan implementasi sistem Semaphore Code Recognition menggunakan Raspberry Pi. Peneliti memenuhi tujuan utama yaitu sistem dapat mengenali berbagai bentuk pose tubuh sesuai dengan data yang telah diberikan. Sistem juga tetap dapat mendapatkan hasil prediksi yang baik dengan besaran akurasi sebesar 92% terlepas dari adanya faktor seperti noise pada Citra, perbedaan ukuran Citra, serta perbedaan intensitas cahaya.

Untuk penelitian lebih lanjut, dimungkinkan untuk menggunakan metode dan/atau modular tambahan untuk dapat meningkatkan performa serta efisiensi CNN pada Raspberry Pi dalam proses pengenalan gestur tubuh untuk mendeteksi Sandi Semaphore.

#### REFERENSI

- [1] I. Suryana, E. Paulus, B. Subartini, and A. Utama, "Sistem Penerjemah Sandi Semaphore Dengan," no. November, pp. 7–11, 2015..
- [2] D. B. Utami and M. Ichwan, "Pengenalan Pose Tangan Menggunakan HuMoment," *J. Infotel*, vol. 9, no. 1, p. 100, 2017, doi: 10.20895/infotel.v9i1.177.
- [3] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Cnn实际训练的," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2012..
- [4] D. N. Fernandez, "Development of a hand pose recognition system on an embedded computer using Artificial Intelligence," *Proc. 2019 IEEE 26th Int. Conf. Electron. Electr. Eng. Comput. INTERCON 2019*, pp. 1–4, 2019, doi: 10.1109/INTERCON.2019.8853573..
- [5] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose tracking," 2020, [Online]. Available: <http://arxiv.org/abs/2006.10204>.
- [6] L. Triyono, E. H. Pratisto, S. A. T. Bawono, F. A. Purnomo, Y. Yudhanto, and B. Raharjo, "Sign Language Translator Application Using OpenCV," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 333, no. 1, 2018, doi: 10.1088/1757-899X/333/1/012109.