Pengenalan Video Peraga Menggunakan Transformer untuk Pembelajaran Bahasa Isyarat

Adam Hadi Prasetyo, Nanik Suciati, dan Anny Yuniarti

Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember

Jln. Arief Rahman Hakim, Surabaya 60111, Indonesia

*e-mail*: adam.hp31@gmail.com, nanik@if.its.ac.id, anny@if.its.ac.id

***Abstrak****—* ***Dalam kehidupan sehari-hari, komunikasi penting dan bahasa isyarat menjadi alternatif komunikasi untuk individu dengan keterbatasan fisik. Standar Isyarat Bahasa Indonesia (SIBI) memberikan panduan dalam bentuk buku, namun pembelajaran dengan media buku kurang efektif dan tidak praktis untuk dibawa. Bahasa isyarat dapat dikenali oleh komputer melalui pengenalan gambar (image recognition). Sebelum komputer mampu mengenali gambar yang diberikan, terdapat model pengenalan gambar yang harus dilatih dengan machine learning. Transformer adalah salah satu metode machine learning yang dapat digunakan dalam melatih model.*** ***Model memiliki 4 multihead, 1 encoder layer dan dilatih selama 50 epoch. Dari pengaturan model tersebut dihasilkan akurasi test sebesar 100% dengan presisi test 100 %. Setelah model dilatih kemudian akan digunakan dalam aplikasi untuk mengenali gestur bahasa isyarat. Selain itu, aplikasi pembelajaran bahasa isyarat juga dilengkapi dengan contoh video peraga bahasa isyarat, untuk menunjang pembelajaran.***

***Kata kunci****—****: Bahasa Isyarat, SIBI, Image Recognition, Machine Learning, Transformer, Self-Attention, Encoder Only***

M

I. PENDAHULUAN

anusia merupakan makhluk ciptaan tuhan yang berakal. Dalam kehidupan sehari-hari, manusia berkomunikasi satu sama lain untuk menyampaikan informasi. Komunikasi umumnya disampaikan melalui lisan, tulisan, serta visual. Akan tetapi, manusia tidak sempurna, beberapa memiliki kekurangan yang menghambat untuk berkomunikasi dengan sesama. Untuk mengatasi hambatan tersebut diciptakan metode lain untuk menggantikan cara komunikasi umum seperti dengan menggunakan bahasa isyarat. Masyarakat pada umumnya kesulitan dalam menggunakan bahasa isyarat karena tidak terbiasa. Untuk mengatasi permasalahan tersebut diperlukan adanya media pembelajaran bahasa isyarat.

Bahasa isyarat adalah bahasa yang menggunakan gestur anggota badan – pada umumnya tangan – untuk menyampaikan maksud. Bahasa isyarat sendiri memiliki standar yang berbeda, sesuai dengan bahasa daerah sekitarnya. Di Indonesia, standar baku bahasa isyarat tercantum dalam Standar Isyarat Bahasa Indonesia (SIBI). Pembakuan bahasa isyarat bertujuan untuk mempermudah penggunaan bahasa isyarat sehingga makna yang disampaikan tepat. SIBI memberikan panduan dan petunjuk sistematis dalam menggunakan bahasa isyarat yang baku dalam bentuk buku. Namun pembelajaran dengan media buku kurang efektif karena gambar pada buku tidak dapat memberikan contoh riil seperti praktek langsung. Selain itu buku kurang praktis untuk dibawa karena memiliki halaman tebal.

Untuk itu diperlukan sebuah sarana aplikasi yang dapat membantu dan mewadahi pembelajaran bahasa isyarat bagi masyarakat umum atau penyandang tunarungu. Diperlukan juga pembelajaran yang interaktif sehingga peserta belajar lebih mudah dalam memahami. Salah satu sarana yang memenuhi syarat tersebut adalah gim, yang menyediakan interaksi dengan peserta ajar dan memberikan umpan balik secara *real-time*.

Hingga saat ini sudah ada beberapa aplikasi yang dikembangkan untuk mewadahi pembelajaran bahasa isyarat. Salah satunya adalah Aplikasi Bahasa Isyarat untuk Tuna Rungu yang menggunakan platform android. Aplikasi ini menyediakan materi dan evaluasi pembelajaran bahasa isyarat dengan media ponsel. Adapun materi yang disajikan dalam bentuk gambar dan video. Untuk kategori materi bahasa isyarat yang tersedia antara lain abjad, angka, benda-benda umum, serta kelompok kata. Modul evaluasi pada aplikasi berupa tebak gambar dan tebak video, yang mana pengguna aplikasi akan memilih gambar atau video yang sesuai dengan soal yang diberikan, atau sebaliknya. [1]

Aplikasi Pembelajaran Bahasa Isyarat (PemBais), adalah aplikasi pembelajaran bahasa isyarat serupa yang memberikan materi gambar dan video penggunaan bahasa isyarat juga telah dikembangkan. Materi pada PemBais memiliki beberapa kategori seperti kosa kata, angka, abjad, serta video gerakan bahasa isyarat. Aplikasi ini tidak menyajikan modul evaluasi bagi pengguna.[2]

Dari contoh yang diberikan, aplikasi pembelajaran bahasa isyarat dapat menyampaikan pembelajaran melalui gambar dan video. Namun pembelajaran bahasa isyarat tersebut belum dapat memberikan evaluasi bagi pengguna tentang ketepatan penggunaan bahasa isyarat. Salah satu cara untuk mengevaluasi ketepatan tersebut adalah dengan memberikan aplikasi pembelajaran kemampuan untuk mengenali bahasa isyarat.

Dalam mengenali bahasa isyarat, solusi yang dapat digunakan adalah *deep learning*. *Convolutional Neural Network* (CNN) merupakan salah satu metode *deep learning* yang umum digunakan ketika *dataset* berupa gambar statis [4]. Jika *dataset* berupa video atau urutan gambar yang saling terhubung, *Recurrent Neural Network* (RNN) dapat menghasilkan akurasi yang lebih baik.[3]

Pada pengembangan aplikasi pembelajaran bahasa isyarat ini, metode *deep learning* yang akan digunakan untuk mengenali bahasa isyarat adalah arsitektur transformer. Arsitektur tersebut memiliki kelebihan dalam mengolah data sekuensial seperti RNN. Namun, transformer dapat mengolah semua *input* dalam satu waktu yang bersamaan.

II. TINJAUAN PUSTAKA

1. *Deep Learning*

*Deep learning* atau *deep structured learning* adalah bagian dari metode *machine learning* dengan basis *neural network* (jaringan saraf) buatan. *Deep learning* memungkinkan model yang terdiri dari beberapa lapisan pemrosesan untuk mempelajari representasi data dengan berbagai tingkat abstraksi. Metode ini menemukan struktur rumit dalam *data set* besar dengan menggunakan algoritma *backpropagation* untuk menunjukkan bagaimana mesin harus mengubah parameter yang digunakan untuk menghitung representasi di setiap lapisan dari representasi di lapisan sebelumnya [7]. Arsitektur *deep learning* seperti *deep neural network* (DNN), *recurrent neural network* (RNN), *convolutional neural* network (CNN), dan *transformer* telah diterapkan di berbagai bidang seperti citra computer, pengenalan suara, serta *natural language processing* (NLP).

A picture containing diagram, line, design

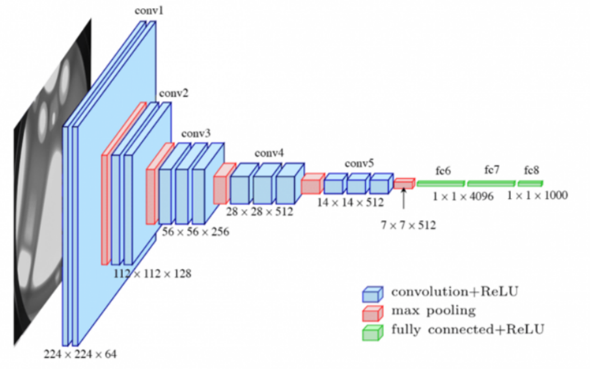
Description automatically generated

**Gambar 1.** Arsitektur Deep Learning dengan  
Multilayer Neural Network [7]

1. *CNN*

CNN (*Convolutional Neural Network* atau *ConvNet*) adalah salah satu arsitektur *deep learning*. CNN merupakan versi regular dari *multilayer perceptron* yang merupakan jaringan yang terhubung penuh. Pada *multilayer perceptron* setiap *neuron* dalam satu lapisan terhubung dengan semua *neuron* yang ada di lapisan berikutnya. [8]

Sebuah *convolutional neural network* terdiri atas *input layer, hidden layer,* dan *output layer*. Pada *feed-forward neural network*, setiap lapisan tengah disebut sebagai lapisan tersembunyi (*hidden layer*) karena *input* dan *output* ditutupi oleh fungsi aktivasi dan konvolusi akhir. Dalam *convolutional neural network,*  lapisan tersembunyi berisi lapisan yang melakukan konvolusi. Umumnya, lapisan ini mencakup lapisan yang melakukan *dot product* dari kernel konvolusi dengan *input matrix* lapisan. Saat kernel konvolusi berjalan di sepanjang *input matrix* pada lapisan, operasi konvolusi menghasilkan peta fitur, yang kemudian berkontribusi pada *input layer* berikutnya. Hal ini diikuti oleh lapisan lain seperti *pooling layer*, *fully connected layer*, dan *normalization layer.*



**Gambar 2.** Arsitektur CNN VGG-16 [11]

1. *RNN*

*Recurrent Neural Network* adalah bagian dari *deep learning* yang memiliki *self-connected hidden layer*. Salah satu kelebihan dari RNN adalah "memori" atau ingatan dari *input* di masa lalu tersimpan dalam *network internal state*. Dari "memori" tersebut RNN dapat memanggil atau mendapatkan konteks *input* masa lalu. Kelebihan lain yang dimiliki RNN, tingkat perubahan (*rate of change*) dari *internal state* dapat diatur melalui *recurrent weights*. Tingkat perubahan yang modular tersebut memberikan kekokohan terhadap distorsi lokal dari *input*. [9]

Graphical user interface, application

Description automatically generated

**Gambar 1.** Arsitektur Umum RNN Terkompresi (kiri) dan Tidak Terkompresi (kanan)

1. *Transformer*

Transformer merupakan implementasi mekanisme *self-attention* pada sebuah model *deep learning*, yang dapat memfokuskan bagian *input* yang diberikan. Model transformer memiliki kesamaan dengan *Recurrent Neural Network* (RNN) yang didesain untuk memproses *input* data sekuensial. Perbedaan yang dimiliki oleh transformer dengan model RNN adalah model transformer dapat memproses semua *input* secara bersamaan. Mekanisme *attention* yang diberikan pada transformer memungkinkan model untuk mendapatkan konteks dari setiap bagian *input*. Sebagai contoh, ketika model diberikan *input* berupa sebuah kalimat bahasa alami, transformer akan memproses seluruh kalimat secara utuh. [10]

A picture containing diagram, sketch, plan, text

Description automatically generated**Gambar 4.** Arsitektur Transformer

1. *TensorFlow*

TensorFlow merupakan *library* yang difokuskan untuk mengerjakan kegiatan yang berhubungan dengan *machine learning* dan *artificial intelligence*. Pada awalnya, TensorFlow dikembangkan oleh tim Google Brain untuk keperluan riset dan produksi internal Google. Pada tahun 2017, versi 1.0.0 TensorFlow dirilis untuk umum. *Library* ini tersedia dalam beberapa bahasa pemrograman seperti Python, C++, JavaScript, dan Java. Dalam penggunaannya, TensorFlow menyediakan API yang menggunakan Keras. API tersebut memungkinkan pengguna untuk membuat model *machine learning* sesuai dengan kebutuhan. TensorFlow dapat memanfaatkan beberapa CPU dan GPU yang tersedia pada komputer untuk menjalankan *model implementation*.

1. *MediaPipe*

MediaPipe adalah *framework* *open-source* yang digunakan untuk mendeteksi objek secara *real-time*. MediaPipe menyediakan solusi untuk berbagai masalah deteksi objek seperti *tracking*, *segmentation*, serta *object detection*. Salah satu solusi MediaPipe yakni Holistic, memungkinkan pengguna untuk menggabungkan beberapa model *tracking* menjadi kesatuan yang utuh seperti kerangka. MediaPipe Holistic dapat digunakan untuk memprediksi pose, dengan batasan resolusi video *input* 256x256 pixel.

III. METODOLOGI PENELITIAN

*A. Deskripsi Data*

*Dataset* yang digunakan untuk melatih model berupa video diambil dalam posisi *landscape* dengan resolusi 1920x1080 *pixel* dan *frame rate* 30fps. Detail *dataset* yang digunakan yaitu:

1. 50 *class*,
2. 20 video dalam setiap *class*,
3. Durasi video 4 detik,
4. 3 pemeraga, dan
5. 6 variasi posisi pemeraga.

Dengan total data sejumlah **1000 data video**.

*B. Perancangan Model* *Pengenalan Bahasa Isyarat*

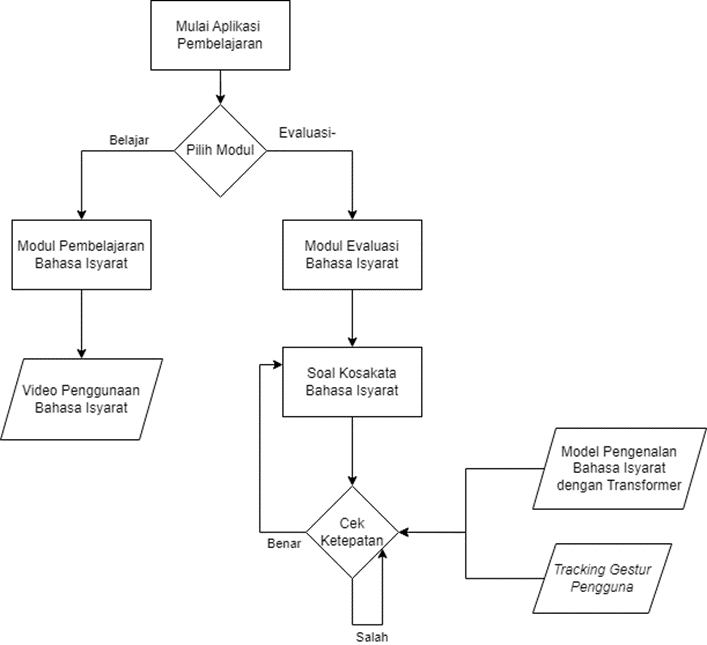
Tugas akhir ini akan menggunakan dan memodifikasi model transformer. Modifikasi yang dilakukan adalah menggunakan bagian *encoder* saja pada transformer. Arsitektur yang digunakan pada model pengenalan bahasa isyarat tidak memerlukan generasi output. Sebagai gantinya, arsitektur tersebut berfokus pada pemrosesan dan pengenalan urutan gerakan atau isyarat dalam bahasa isyarat.

Langkah dalam melatih model dapat diringkas sebagai berikut:

|  |  |
| --- | --- |
| **Langkah 1** | Melakukan *preprocessing* pada *dataset* yang digunakan dengan memberikan label pada video |
| **Langkah 2** | Melakukan ekstraksi fitur dengan mengambil *keypoint* setiap *frame* menggunakan MediaPipe |
| **Langkah 3** | Normalisasi data *keypoints* untuk meningkatkan akurasi model. |
| **Langkah 4** | Augmentasi data dengan menambahkan data yang menggeser posisi *keypoints*. Data digeser ke arah atas, bawah, kiri dan kanan untuk setiap *keypoints* yang telah dinormalisasi. |
| **Langkah 5** | *Split* data dengan rasio *train : validation : test* sebesar 70:15:15. |
| **Langkah 6** | *Train* model dengan arsitektur transformer *encoder only*. |
| **Langkah 7** | Evaluasi akurasi dan loss pada model. |
| **Langkah 8** | *Export* model yang telah dievaluasi untuk digunakan pada aplikasi pembelajaran bahasa isyarat. |

*C. Perancangan Aplikasi Pembelajaran Bahasa Isyarat*

Pada Gambar 5 menunjukkan rancangan aplikasi pembelajaran bahasa isyarat. Saat memulai aplikasi, akan terdapat dua pilihan modul, Belajar dan Evaluasi. Pada modul belajar terdapat video yang memperagakan penggunaan bahasa isyarat dengan Standar Isyarat Bahasa Indonesia (SIBI). Pada modul evaluasi, terdapat ujian soal kosakata bahasa isyarat yang dapat diselesaikan. Untuk melakukan pengecekan ketepatan dalam penggunaan, aplikasi akan melakukan *tracking* gestur menggunakan kamera. Gestur akan dievaluasi dengan model yang telah dilatih menggunakan transformer. Hasil evaluasi akan ditampilkan setelah proses cek ketepatan. Modul evaluasi juga menyediakan tempat untuk latihan peraga bahasa isyarat. Pada bagian latihan, diberikan kebebasan memperagakan bahasa isyarat tanpa adanya soal kosakata bahasa isyarat. Hasil deteksi peraga pada latihan akan ditampilkan secara *real time*.



**Gambar 5.** Diagram Alir Aplikasi Pembelajaran Bahasa Isyarat

*D. Implementasi Aplikasi Pembelajaran Bahasa Isyarat*

Aplikasi pembelajaran bahasa isyarat menggunakan bahasa pemrograman Python. Aplikasi ini menggunakan *library* Tkinter, OpenCV, dan MediaPipe. Terdapat dua modul utama pada aplikasi pembelajaran bahasa isyarat, yakni modul Belajar dan modul Evaluasi.

Dalam implementasinya, aplikasi ini menggunakan *framework* Tkinter sebagai antarmuka grafis, OpenCV untuk memanipulasi dan memutar video, serta MediaPipe untuk mendeteksi pose tubuh dan tangan. Modul Belajar memiliki fitur untuk memutar video peragaan bahasa isyarat. Sedangkan pada modul Evaluasi, terdapat fitur untuk memfasilitasi pengguna dalam mempraktekan bahasa isyarat. Praktek bahasa isyarat tersebut memiliki soal, dimana pengguna harus mencocokkan bahasa isyarat yang diperagakan dengan soal yang diberikan.

*E. Skenario Uji Coba Model Pengenalan Bahasa Isyarat*

Dalam mengimplementasikan model, terdapat beberapa skenario uji coba dengan mengganti beberapa parameter untuk menemukan model dengan akurasi yang terbaik. Parameter yang diubah pada uji coba adalah jumlah *epoch*, jumlah *multihead*, dan jumlah *encoder layer*. Pada skenario normal, parameter yang digunakan adalah jumlah *epoch* dengan variasi 50 dan 40 *epoch*. Skenario penambahan *multihead* menggunakan parameter *epoch* 50 dan 40, serta *multihead* sejumlah 8. Skenario penambahan *encoder layer* terdapat peningkatan pada jumlah *encoder layer* menjadi 2, sementara parameter *epoch* tetap menggunakan 50 dan 40. Terakhir, skenario penambahan *multihead* dan *encoder layer* menggabungkan perubahan pada kedua skenario sebelumnya, dengan *multihead* sejumlah 8 dan *encoder layer* sejumlah 2, serta parameter *epoch* 50 dan 40.

**Tabel 1.** Skenario Uji Coba Model

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No. Skenario** | ***Epoch*** | | ***Multihead*** | | ***Encoder Layer*** | |
| 1 | | 50 | | 4 | | 1 | |
| 2 | | 40 | | 4 | | 1 | |
| 3 | | 50 | | 8 | | 1 | |
| 4 | | 40 | | 8 | | 1 | |
| 5 | | 50 | | 4 | | 2 | |
| 6 | | 40 | | 4 | | 2 | |
| 7 | | 50 | | 8 | | 2 | |
| 8 | | 40 | | 8 | | 2 | |

IV. HASIL DAN PEMBAHASAN

*A. Hasil Uji Coba Model Pembelajaran Bahasa Isyarat*

Dari skenario uji coba model yang telah dilakukan, dhasil dapat dilihat pada Tabel 2.

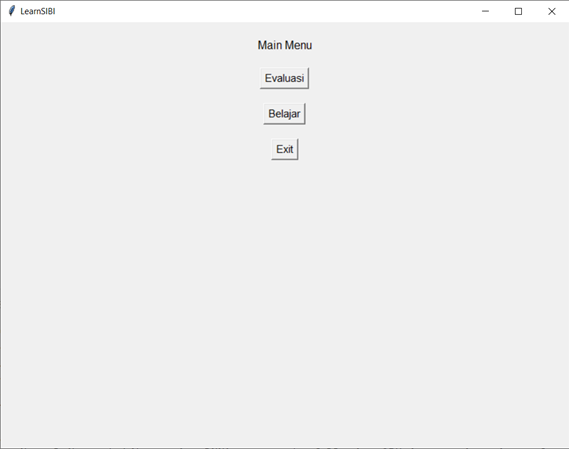
**Tabel 2**. Hasil Uji Coba Model

|  |  |  |
| --- | --- | --- |
| **No. Skenario** | ***Test Accuracy  (%)*** | ***Test Precision (%)*** |
| 1 | 100,00 | 100,00 |
| 2 | 98,67 | 100,00 |
| 3 | 93,33 | 95,24 |
| 4 | 96,00 | 96,64 |
| 5 | 97,33 | 97,33 |
| 6 | 96,67 | 96,67 |
| 7 | 86,00 | 87,16 |
| 8 | 88,67 | 90,48 |

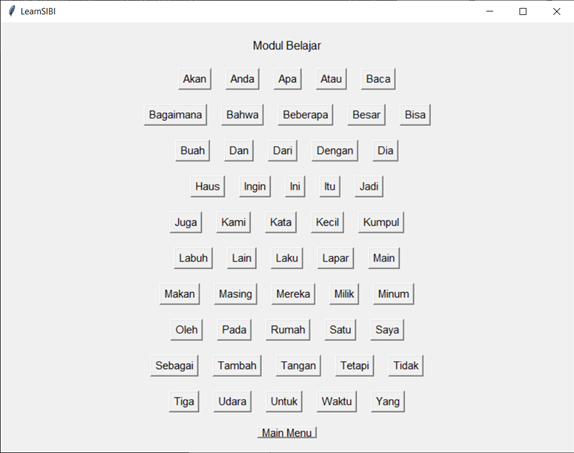
Dari uji coba yang telah dilakukan, didapatkan bahwa skenario ke-1 memiliki hasil terbaik dengan akurasi tes sebesar 100,00% dan presisi tes 100,00%. Skenario tersebut memiliki parameter *epoch* sebanyak 50 dengan 4 *multihead* dan 1 *encoder layer*. Sedangkan untuk hasil terburuk terdapat pada skenario ketujuh yang memiliki akurasi tes 86,00% dan presisi test 87,16%. Parameter pada skenario tersebut adalah 50 *epoch*, 8 *multihead* dan 2 *encoder* layer.

*B. Hasil Implementasi Aplikasi Pembelajaran Bahasa Isyarat*

Gambar 6 adalah antarmuka untuk menu utama pada aplikasi pembelajaran bahasa isyarat. Menu utama ini berfungsi sebagai penghubung modul evaluasi dan modul belajar. Terdapat *title* menu dengan judul *Main Menu*, serta tombol Evaluasi dan Belajar. Ketika tombol Evaluasi dan Belajar dipencet, pengguna diarahkan ke menu untuk modul evaluasi atau belajar. Aplikasi akan ditutup ketika tombol Exit dipilih.



**Gambar 6.** Tampilan Antarmuka Menu Utama Aplikasi Pembelajaran Bahasa Isyarat

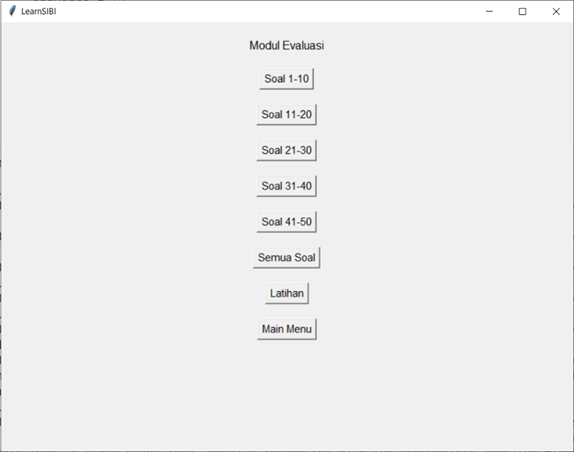


**Gambar 7.** Tampilan Antarmuka Menu Modul Belajar Aplikasi Pembelajaran Bahasa Isyarat



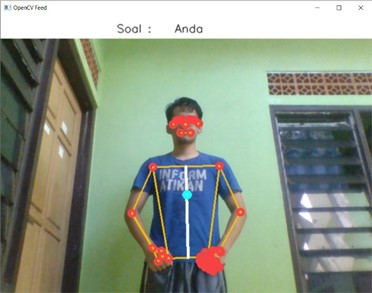
**Gambar 8.** Jendela Pemutar Video Peragaan Bahasa Isyarat

Gambar 7 adalah antarmuka untuk menu utama modul belajar. Menu ini berfungsi sebagai menu untuk memutar video peragaan bahasa isyarat. Terdapat *title* menu dengan judul *Modul Belajar*, serta tombol untuk memutar video. Setiap tombol yang ada pada modul merupakan *class* bahasa isyarat yang digunakan dalam aplikasi pembelajaran. Ketika salah satu tombol dipencet, muncul jendela baru yang memutarkan video seperti pada Gambar 8. Jendela pemutaran video ditutup secara otomatis ketika video selesai atau ketika tombol ‘q’ dipencet. Pengguna dapat kembali ke menu utama dengan tombol Main Menu.



**Gambar 9.** Tampilan Antarmuka Menu Modul Evaluasi Aplikasi Pembelajaran Bahasa Isyarat

Gambar 9 adalah antarmuka untuk menu utama modul evaluasi. Menu ini berfungsi sebagai menu untuk memilih soal yang diujikan pada proses evaluasi. Terdapat *title* menu dengan judul *Modul Evaluasi*, serta tombol memilih soal evaluasi. Ketika salah satu tombol soal dipencet, muncul jendela baru untuk proses evaluasi bahasa isyarat. Pengguna dapat kembali ke menu utama dengan tombol Main Menu.



**Gambar 10.** Tampilan Jendela Proses Evaluasi Aplikasi Pembelajaran Bahasa Isyarat

Pada Gambar 10 terdapat tampilan video *real time* pengguna. Di bagian atas, terdapat antarmuka untuk soal. Di awal mulainya proses evaluasi, terdapat waktu jeda 2,5 detik untuk pengguna bersiap-siap. Ketika data *keypoints* dari 120 *frame* terkumpul, jendela proses evaluasi akan ditutup, lalu data *keypoints* digunakan untuk memprediksi bahasa isyarat. Hasil evaluasi ditampilkan pada jendela baru seperti pada Gambar 11.



**Gambar 11.** Tampilan Jendela Hasil Evaluasi Aplikasi Pembelajaran Bahasa Isyarat

*C. Evaluasi*

Dari 8 skenario uji coba yang dilakukan pada tahap pelatihan model tidak ditemukan kendala yang mengganggu jalannya proses. Pelatihan pada uji coba ini menggunakan CPU bawaan dari laptop. Waktu pelatihan tersebut dapat dikurangi dengan mengganti hardware pelatihan menggunakan GPU.

Secara keseluruhan, aplikasi sudah berjalan lancar dan sesuai dengan rancangan awal. Model pengenalan bahasa isyarat yang akan digunakan adalah model dari skenario latihan ke-1 dengan akurasi sebesar 100%. Model tersebut dapat digunakan dalam pengenalan bahasa isyarat oleh modul evaluasi aplikasi pembelajaran tanpa kendala. Kemudian pada modul belajar, video yang diputarkan tidak memiliki masalah apapun. Masalah kecil yang dijumpai pada proses evaluasi adalah *frame* *rate* video. Seharusnya *frame rate* video berada pada 30fps, sedangkan pada proses evaluasi video tidak mencapai 30fps. Salah satu faktor yang menyebabkan turunnya *frame rate* adalah proses evaluasi yang berjalan sebagai satu proses utuh. Masalah tersebut tidak memengaruhi performa atau menyebabkan error pada aplikasi pembelajaran bahasa isyarat.

V. KESIMPULAN DAN SARAN

*A. Kesimpulan*

Berdasarkan hasil perancangan dan uji coba pada tugas akhir ini, didapatkan bahwa arsitektur transformer dapat mengenali data bahasa isyarat dalam bentuk video dengan baik. Aplikasi pembelajaran bahasa isyarat yang memanfaatkan arsitektur transformer dapat mengenali video peragaan bahasa isyarat secara *real time* sebagai sarana evaluasi. Selain itu Aplikasi pembelajaran bahasa isyarat dapat memberikan sarana pembelajaran melalui pemutaran video peraga. Dari hasil perancangan dan uji coba pada tugas akhir ini dapat disimpulkan bahwa:

* Model pengenalan bahasa isyarat menggunakan arsitektur Transformer dengan variasi Transformer *encoder only*. Model ini memiliki 4 *multihead* dan 2 *encoder layer*, serta dilatih selama 40 *epoch*.
* Aplikasi pembelajaran bahasa isyarat memiliki menu utama yang menghubungkan modul belajar dan modul evaluasi. Modul belajar menyediakan video peraga bahasa isyarat sebagai contoh, sedangkan modul evaluasi memungkinkan pengguna menguji kemampuan mereka dalam menggunakan bahasa isyarat.
* Modul evaluasi menggunakan model pengenalan bahasa isyarat dengan arsitektur Transformer *encoder only* untuk mengenali gestur bahasa isyarat. Selama proses evaluasi, data keypoints dari setiap *frame* yang diambil secara *real time* oleh kamera dikumpulkan. Setelah terkumpul sebanyak 120 *frame*, data tersebut digunakan untuk memprediksi gestur bahasa isyarat. Gestur dianggap benar jika prediksi gestur sesuai dengan soal yang diberikan.
* Modul belajar memutar video peraga bahasa isyarat sesuai kata yang dipilih. Ketika video diputar, jendela baru akan muncul untuk memutar video peraga tersebut. Setelah selesai diputar, jendela akan ditutup secara otomatis.
* Aplikasi pembelajaran bahasa isyarat dievaluasi berdasarkan performa model pengenalan bahasa isyarat dan fitur aplikasi yang mendukung pembelajaran bahasa isyarat. Evaluasi menunjukkan bahwa model dapat mengenali bahasa isyarat yang diperagakan, dan fitur aplikasi untuk memutar video peraga dan evaluasi gestur telah berjalan dengan baik.

*B. Saran*

Dalam pengerjaan tugas akhir ini, terdapat banyak kekurangan yang dapat diperbaiki dan dikembangkan untuk meningkatkan kinerja aplikasi dan model. Pada bagian model, dapat dilakukan perubahan pada arsitektur atau mengganti *hardware* pelatihan model dengan GPU untuk memangkas waktu pelatihan model. Saran berikutnya yang dapat dilakukan untuk mengembangkan model, arsitektur model dapat diubah dan di-*tuning* untuk menemukan model yang lebih optimal. *Dataset* yang digunakan pada model dapat ditambahkan lagi untuk *train, test* dan *validation*. Di bagian aplikasi, *user interface* (antarmuka) dapat dibenahi tata letak, bentuk, serta tampilannya. Aplikasi dapat dikembangkan lagi dengan bahasa pemrograman dan *platform* yang berbeda. Sebagai contoh dengan menggunakan C# untuk *platform* Android.

DAFTAR PUSTAKA

[1] P. Mandarani and Y. Putra, "Aplikasi Bahasa Isyarat untuk Tuna Rungu Menggunakan Platform Android," Institut Teknologi Padang, 2020. [Online]. Available: https://teknoif.itp.ac.id/index.php/teknoif/article/view/40/780.

[2] M. H. Pradikja, H. Tolle, and K. C. Brata, "Pengembangan Aplikasi Pembelajaran Bahasa Isyarat Berbasis Android Tablet," Universitas Brawijaya, 2019. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1705/645.

[3] C. K. M. Lee, K. K. H. Ng, C. Chen, H. C. W. Lau, S. Y. Chung, and T. Tsoi, "American Sign Language Recognition and Training Method with Recurrent Neural Network," [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420310745.

[4] S. Suharjito, R. Anderson, F. Wiryana, and M. C. Ariesta, "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output," [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917320720.

[5] S. Suharjito, H. T. Gunawan, N. Thiracitta, and A. Nugroho, "Sign Language Recognition Using Modified Convolutional Neural Network Model," IEEE, 2018. DOI: 10.1109/INAPR.2018.8627014.

[6] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian Sign Language Recognition using YOLO Method," Institut Teknologi Sepuluh Nopember, 2021. [Online]. Available: https://www.researchgate.net/publication/350082064\_Indonesian\_Sign\_Language\_Recognition\_using\_YOLO\_Method/fulltext/608b697e458515d315e6bd95/Indonesian-Sign-Language-Recognition-using-YOLO-Method.pdf.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[8] M. V. Valueva et al., "Application of The Residue Number System to Reduce Hardware Costs of The Convolutional Neural Network Implementation," Mathematics and Computers in Simulation, Elsevier BV, vol. 177, pp. 232-243, 2020. doi:10.1016/j.matcom.2020.04.031.N. Balakrishnan, “Modified Vogel’s approximation method for the unbalanced transportation problem,” *Appl. Math. Lett.*, vol. 3, no. 2, pp. 9–11, 1990, doi: 10.1016/0893-9659(90)90003-T.

[9] A. Graves et al., "A Novel Connectionist System for Improved Unconstrained Handwriting Recognition," in Proceedings of the 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 2009, pp. 5-8.

[10] A. Vaswan et al., "Attention is All You Need," [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[11] K. Le, "An overview of VGG16 and NiN models," Medium, 2017. [Online]. Available: https://medium.com/mlearning-ai/an-overview-of-vgg16-and-nin-models-96e4bf398484. Accessed on Dec. 6, 2022.