
Deteksi Gerak Tangan sebagai Pengenal Bahasa Isyarat menggunakan Mediapipe dan Long-Short Term Memory

Prabowo Budi Utomo

Program Studi Administrasi Server dan Jaringan Komputer

Akademi Komunitas Negeri Putra Sang Fajar Blitar

Email: prabowo86@akb.ac.id

Risky Aswi Ramadhani

Fakultas Teknik, Program Studi Teknik Informatika

Universitas Nusantara PGRI Kediri

Email: ra.komo999@gmail.com

Hindra Kurniawan

Program Studi Operasionalisasi Komputer Perkantoran

Akademi Komunitas Negeri Putra Sang Fajar Blitar

Email: ra.komo999@gmail.com

ABSTRAK

Kehadiran orang disabilitas atau penyandang cacat seringkali dipandang sebelah mata oleh banyak orang dan mendapat perlakuan diskriminatif serta dianggap merepotkan, padahal jumlah penyandang cacat di Indonesia cukup banyak dan bervariasi. Sebagai salah satu kecacatan, tuli menjadi salah satu yg terbanyak diderita oleh orang Indonesia, dalam sebuah survey Nasional tahun 1994-1996 ada sekitar 18,5 % atau 40,5 juta jiwa yang mengalami gangguan pendengaran yang menggunakan bahasa isyarat berbasis gerak tangan sebagai media komunikasi. Dalam beberapa kondisi gerakan tangan ini cukup sulit untuk dipahami khususnya oleh yang bukan pengguna bahasa isyarat, bahkan dalam beberapa kasus formal hingga dibutuhkan juru bahasa isyarat untuk memahami maksud dari gerak tangan yang digunakan. Dalam penelitian ini menggunakan 36 jenis gerak tangan yang masing-masing gerak tangan direkam sebanyak 30 video *sequence* dengan setiap *sequence* akan merekam sebanyak 30 *frame*. Setiap frame yang diperoleh akan diekstraksi nilai *keypoint* dari wajah, pose dan gerak tangan yang direkam, untuk selanjutnya diproses dan dimodelkan menggunakan 3 metode *Long ShortTerm Memory* (LSTM) yaitu *single layers LSTM*, *multi layers LSTM* dan *Bidirectional LSTM*. Pemodelan LSTM yang telah dilakukan menghasilkan nilai akurasi sebesar 91% saat mendeteksi dan mengenali gerak tangan Bahasa isyarat BISINDO pada model yang diproses menggunakan metode *single layer LSTM epoch 500, hidden layers 32, batch size 64*.

Kata kunci: Tuli, LSTM, BISINDO, Pengenal Bahasa Isyarat

ABSTRACT

The presence of people with disabilities or people with disabilities is often underestimated by many people and receives discriminatory treatment and is considered a hassle, even though the number of people with disabilities in Indonesia is quite large and varied. As a disability, deafness is one of the most suffered by Indonesians. In a national survey in 1994-1996 there were around 18.5% or 40.5 million people who experienced hearing loss who used hand gesture-based sign language as a medium of communication. . In some conditions, these hand gestures are quite difficult to understand, especially for non-sign language users, even in some formal cases, an interpreter is required to understand the meaning of the hand gestures used. In this study, 36 types of hand movements were used, where each hand movement was recorded in 30 video sequences with each sequence recording 30 frames. Each frame obtained will be extracted keypoint values from recorded faces, poses and hand movements, to be further processed and modeled using 3 Long ShortTerm

Memory (LSTM) methods, namely single layer LSTM, multi layers LSTM and Bidirectional LSTM. The LSTM modeling that has been carried out produces an accuracy value of 91% when detecting and recognizing BISINDO's hand gestures in the model processed using the single layer LSTM epoch 500, hidden layers 32, batch size 64 method.

Keywords: Deaf, LSTM, BISINDO, Sign Language Recognition

1. PENDAHULUAN

Kehadiran orang disabilitas atau penyandang cacat seringkali dipandang sebelah mata oleh banyak orang, bahkan banyak yang mendapat perlakuan diskriminatif dan dianggap merepotkan, padahal melihat dari jumlah penyandang cacat di Indonesia yang cukup banyak tentu saja kondisi ini tidak boleh dibiarkan. Islam sendiri memandang para penyandang disabilitas sebagai sosok yang wajib mendapat perhatian yang didasari salah satunya pada rasa kemanusiaan, sebagaimana yang disampaikan Allah Subhanahu wa ta'ala didalam QS al-Israa [17] : 70 “Dan sesungguhnya telah Kami muliakan anak-anak Adam” [24]. Sebagai salah satu kecacatan, tuli menjadi salah satu yg terbanyak diderita oleh orang Indonesia, dalam sebuah survey Nasional tahun 1994-1996 ada sekitar 18.5 % atau 40.5 juta jiwa yang mengalami gangguan pendengaran [25]. Penyandang cacat menggunakan bahasa isyarat sebagai media komunikasi, khususnya bagi penderita tuli atau bisu. Mereka menggunakan gerakan tangan sebagai bahasa non-verbal dalam mengekspresikan keinginan atau emosi yang dirasakan. Di Indonesia sendiri terdapat dua bahasa isyarat yaitu Sistem Isyarat Bahasa Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) yang penggunaan disesuaikan dengan kondisi. SIBI dianggap bukan bahasa isyarat namun lebih seperti sistem dari suatu bahasa karena penggunaan yang sama dengan penggunaan Bahasa Indonesia, sedangkan BISINDO telah digunakan secara luas antara penderita tuli yang muncul secara alami dari interaksi dengan lingkungan sejak kecil dan telah menjadi budaya tuli di Indonesia

Namun dalam beberapa kondisi gerakan tangan ini cukup sulit untuk dipahami khususnya oleh yang bukan pengguna bahasa isyarat, bahkan dalam beberapa kasus formal hingga dibutuhkan juru bahasa isyarat untuk memahami maksud dari gerak tangan yang digunakan [1]. Tujuan dari penelitian ini untuk mengembangkan model komunikasi satu arah penyandang tuli dengan orang normal dengan memanfaatkan metode *deep learning* dalam mendeteksi dan mengenali bahasa isyarat dari gerak tangan untuk kemudian dikonversi menjadi teks [2].

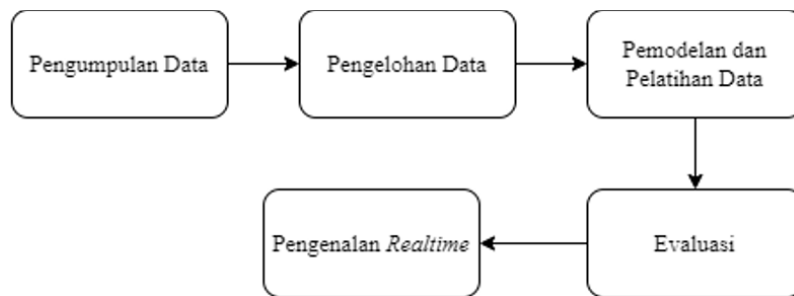
Dalam penelitian sebelumnya A. Mittal dkk. mengusulkan metode untuk mendeteksi dan pengenalan bahasa isyarat India (ISL) berdasar *dataset* yang diperoleh dari 6 orang berbeda, dimana setiap orang mengumpulkan data dari 35 kata yang diulang setidaknya 15 kali untuk setiap kata untuk kemudian *dataset* dilatih menggunakan metode LSTM yang telah dimodifikasi sehingga diperoleh model untuk pengenalan tanda dan kalimat bahasa isyarat [3]. sedangkan K. Gomase dkk menggunakan *mediapipe* dan metode KNN dalam deteksi dan pengenalan bahasa isyarat India yang memperoleh rata-rata akurasi 86% hingga 91% pada video dengan 17 FPS [4]. dalam penelitian lain yang dilakukan Halder dkk. yang menggunakan *mediapipe* dan *machine learning* untuk memperoleh model pengenalan yang ringan dan bisa disesuaikan dengan perangkat pintar dengan bahasa isyarat Amerika, India dan Turki sebagai *dataset*, dimana akurasi model yang diperoleh diperkirakan mencapai 90% [5]. Dalam penelitian lain juga digunakan algoritma K-Mean dan *mediapipe* sebagai teknologi pengenalan gerak secara *realtime* [6]. Mengacu pada penelitian lain yang disampaikan diatas, maka dalam deteksi dan pengenalan bahasa isyarat dibutuhkan semacam model berdasar *dataset*, dimana *dataset* diperoleh melalui pengumpulan data yang diolah menggunakan *mediapipe*, sedangkan untuk model diperoleh dengan melatih *dataset* menggunakan metode LSTM.

Penelitian ini akan menggunakan *dataset* gerak tangan bahasa isyarat untuk huruf dan kata dengan berdasar ketentuan BISINDO (Bahasa Isyarat Indonesia). Gerak tangan akan dideteksi dan direkam secara *realtime* dengan memanfaatkan *library opencv* dan *mediapipe* [7], untuk selanjutnya dilakukan beberapa proses dan disimpan sebagai *dataset*, *dataset* yang diperoleh akan diproses lebih lanjut menggunakan metode *Long-Short Term Memory*. *Long-Short Term Memory* (LSTM) merupakan salah satu jenis Machine Learning berbasis pendekatan *Recurrent Neural Network* yang

dapat mengenali kondisi saat ini dengan menggunakan *dataset* skala besar [8], sejauh pengetahuan penulis di Indonesia belum banyak yang menerapkan *mediapipe* dalam deteksi gerak tangan atau *Long Short-Term Memory* dalam mengenali isyarat gerak tangan.

2. METODOLOGI PENELITIAN

Penelitian dilaksanakan dalam beberapa tahapan yang difokuskan untuk melakukan pengenalan Bahasa isyarat dengan memanfaatkan *library OpenCV* dan *mediapipe* untuk mendeteksi dan mengeskraksi *keypoints* dari *frame* video gerak tangan yang selanjutnya diproses menggunakan algoritma *Long-Short Term Memory* sehingga diperoleh model untuk melakukan pengenalan BISINDO (Bahasa Isyarat Indonesia). Tahapan yang dilaksanakan dalam penelitian ini meliputi tahap pengumpulan data, pengolahan data, pemodelan dan pelatihan data, evaluasi dan pengenalan *Realtime*. Detail tentang tahapan-tahapan yang dilaksanakan dalam penelitian ini sebagaimana yang ditunjukkan pada Gambar 1.



Gambar 1. Alur Metode Penelitian

2.1. Pengumpulan Data

Sumber Data yang diperggunakan dalam penelitian ini bersumber dari video gerak tangan yang direkam menggunakan kamera yang dibuat berdasarkan gerak isyarat BISINDO (Bahasa Isyarat Indonesia), dimana data yang dikumpulkan sebanyak 35 jenis gerak isyarat yang bersumber dari huruf abjad A – Z (26 gerak isyarat) atau kosakata isyarat yang digunakan sehari-hari seperti kata ‘Tolong’, kata ‘Terima kasih’, kata ‘Hebat’, kata ‘Maaf’, kata ‘Halo’, kata ‘Oke’, kata ‘Iyo’, kata ‘Ora’, kata ‘aku malu’.

Video gerak tangan akan direkam dari kamera dengan memanfaatkan *library opencv* untuk mendeteksi dan merekam tangan, badan dan wajah, dan *library mediapipe* untuk mendapatkan model *landmark* dan penempatan 25 *keypoints* pada tangan, 2 *keypoints* pada badan dan 9 *keypoints* pada wajah.



Gambar 2. Pengumpulan Gambar Gerak Tangan

2.2. Pengolahan Data

Terdapat tiga tahapan utama yang dilakukan dalam pengolahan data, tahapan pertama yaitu klasifikasi video gerak tangan berdasar jenis gerak isyarat yang dilakukan, dimana dalam perekaman

gerak tangan telah ditentukan akan direkam sebanyak minimal 30 video *sequence* untuk masing-masing jenis gerak isyarat, dengan setiap setiap *sequence* akan merekam sebanyak 30 *frame*.

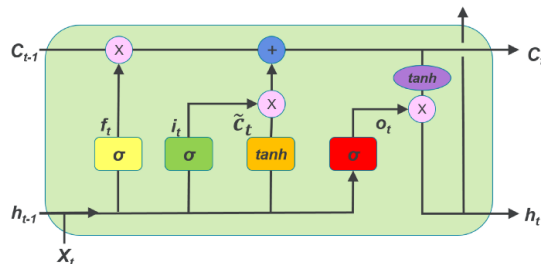
Pada tahapan kedua dilakukan ekstraksi *keypoints* dari setiap pola gerak tangan untuk disimpan dalam bentuk *array* menggunakan library *numpy*, sehingga nanti akan dihasilkan data file berekstensi *npz* dari setiap pola gerak tangan yang memuat informasi *keypoints* dari setiap *frame*.

Tahapan ketiga dilaksanakan bersamaan dengan tahapan kedua, dimana data file hasil ekstraksi *keypoints* dilakukan pada masing-masing video gerak tangan yang direkam kemudian secara otomatis tersimpan dalam *folder* yang sudah dibuat sebelumnya [9].

2.3. Pemodelan dan Pelatihan Data

Dalam tahapan ini data yang diproses dalam tahapan sebelumnya akan dilatih menjadi beberapa model menggunakan 3 jenis arsitektur LSTM yaitu *Single layer LSTM*, *multi layer LSTM* dan *Bidirectional LSTM* sebagaimana yang diusulkan oleh alim & Rakun [10], perbedaan arsitektur dan hasil prediksi pada masing-masing model akan digunakan untuk menentukan model yang lebih efektif [11] dalam mengenali dan menerjemahkan gerak tangan. *Long-Short Term Memory* sendiri merupakan tipe *neural network* yang termasuk dari *Recurrent Neural network* (RNN), dimana RNN menggunakan n umpan balik dari *output layers* kembali ke *input layers*, dimana setiap koneksi umpan balik dapat digunakan sebagai *tim-delay gate* [8], disamping itu RNN mampu mewakili secara eksplisit pengaruh nilai *output* masa lalu pada perhitungan *output* saat ini.

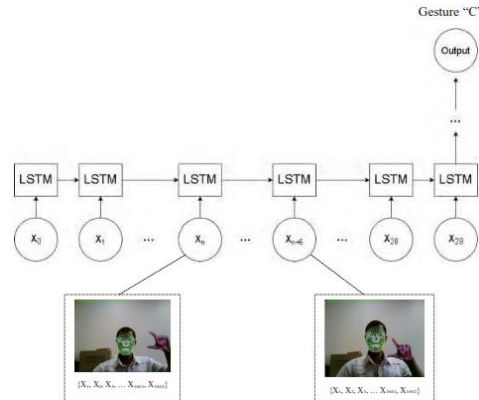
Sebagai varian dari RNN, LSTM memiliki kemampuan dalam mempelajari data yang harus digunakan dengan tujuan untuk memperbaiki permasalahan pada *gradient vanishing* dan *exploding RNN*, LSTM menggantikan vektor RNN dengan blok memori yang dilengkapi gate sehingga dapat mempertahankan memori jangka panjang yang pada prinsipnya dapat melatih bobot gatling yang tepat sehingga berguna dalam mencapai stae-of-art untuk berbagai masalah salah satunya pengenalan ucapan [12]. Arsitektur dalam LSTM secara umum terdiri dari *memory cell*, *input gate*, *output gate* dan *forget gate* [13], setiap unit LSTM (gambar 3) akan memiliki sel memori dan status yang dalam waktu t yang direpresentasikan dengan c_t , yang dibaca dan dimodifikasi serta dikendalikan oleh fungsi *sigmoid* dan berpengaruh pada gerbang masukan i_t , gerbang *forget* f_t dan gerbang keluaran o_t , dua sumber eksternal (h_{t-1} dan x_t). Status tersembunyi h_t dihitung oleh vektor masukan x_t yang diterima jaringan pada waktu t dan status tersembunyi sebelumnya h_{t-1} . Pada saat menghitung status simpul lapisan tersembunyi, gerbang masukan, keluaran, *forget* dan x akan secara bersamaan mempengaruhi keadaan node. Selain itu, setiap gerbang memiliki sumber internal, yaitu, status sel c_{t-1} dari blok selnya. Tautan antara cell dan gerbang sendiri dirujuk ke koneksi *peephole* [14].



Gambar 3. Perulangan Modul LSTM[15]

2.3.1. Single layer LSTM

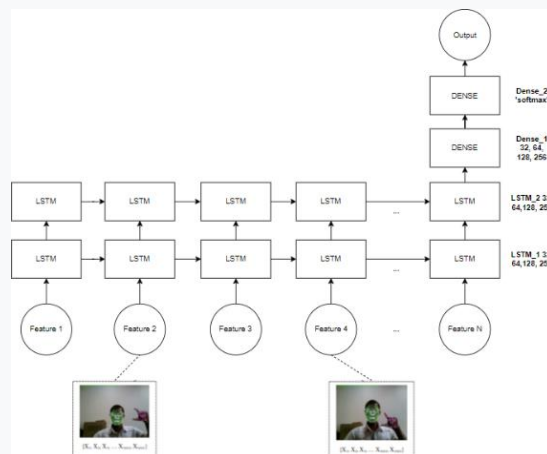
Single layer LSTM memiliki jaringan dasar yang terdiri dari satu lapisan LSTM antara lapisan *input* dan *output* [9] yang memiliki tiga gerbang dan komponen sel memori. *Output* dari lapisan terakhir pada langkah waktu terakhir dari setiap urutan harus diberi label yang sama sehingga dapat digunakan untuk input pada lapisan selanjutnya, arsitektur *single layer LSTM* dapat dilihat pada gambar 4.



Gambar 4. Single layer LSTM[16]

2.3.2. Multilayer LSTM

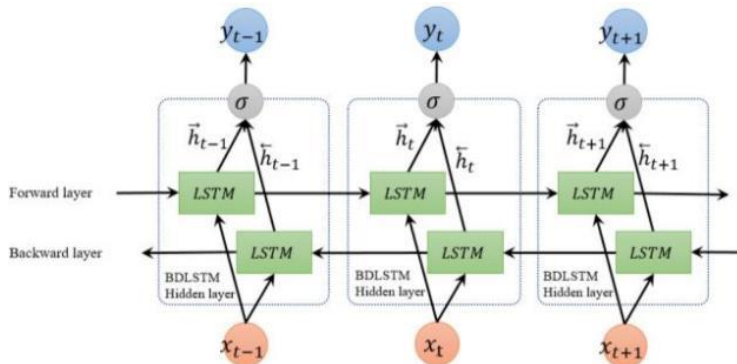
Multilayer LSTM merupakan beberapa lapis layer LSTM yang digabungkan menjadi satu lapisan LSTM dimana setiap lapis LSTM akan melakukan perhitungan pada satu indeks waktu dan mentransfer *output* ke LSTM berikutnya. Jumlah lapisan LSTM yang digabungkan menunjukkan jumlah pengamatan data yang dipertimbangkan sebelum membuat prediksi. Dalam penelitian ini setelah mempertimbangkan data yang telah diproses, maka akan digunakan 2 lapis *layer LSTM* dan 3 lapis *layer LTSM* dalam pelatihan model.



Gambar 5. Multilayer LSTM[17]

2.3.3. Bidirectional LSTM

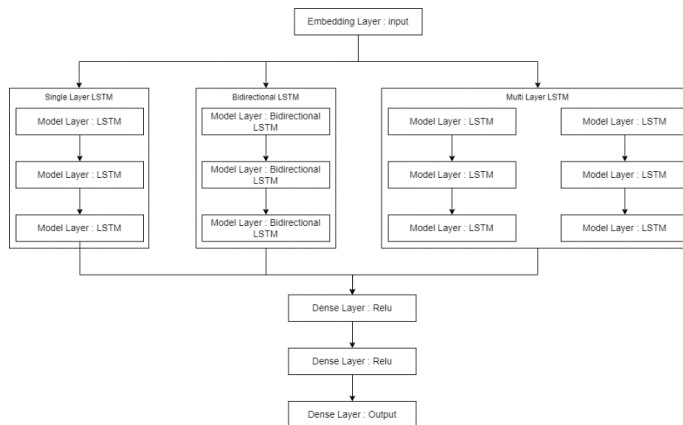
LSTM memiliki kelemahan dalam memproses hanya satu arah saja, yaitu dari awal sampai akhir, dengan mempertimbangkan mekanisme yang hanya satu arah ini mendorong lahirnya algoritma *Bidirectional LSTM*, dimana secara teknis algoritma ini menjalankan dua unit algoritma LSTM dalam satu waktu untuk membaca secara dua arah yaitu awal sampai akhir/*forward* dan akhir sampai awal/*backward* [18]. Arsitektus *Bidirectional LSTM* seperti pada gambar 6 memiliki 2 lapisan LSTM yang terpisah satu untuk maju/*forward* dan satu untuk mundur/*backward*, kemudian setiap *output* yang dihasilkan dari lapisan LSTM akan digabungkan menggunakan fungsi *merge* [19].



Gambar 6. Arsitektur Bidirectional LSTM[20]

2.4. Implementasi Model

Dalam penelitian ini akan dimanfaatkan tiga model algoritma LSTM yaitu single layer LSTM, multi layer LSTM dan *Bidirectional* LSTM dalam mendeteksi dan mengenali gerak tangan. Ketiga model algoritma ini akan menggunakan satu *embedding layer* sebagai *input layer*, tiga model *layer* LSTM, dua *dense layer* sebagai *hidden layer* dan satu *dense* sebagai *output layer*, penggunaan layer yang cukup banyak ditujukan untuk memperoleh model dengan akurasi tinggi [21]. Detail dair penggunaan *layer* pada implementasi model ditunjukkan gambar 7 berikut.



Gambar 7. Layer model yang dipergunakan

Dalam proses *training* data dipergunakan beberapa parameter seperti jumlah *batch size* dengan nilai 32, 64, 128, 256 dengan *hidden layer* dengan nilai 32 dan 64. Data juga akan *ditraining* pada masing-masing algoritma LSTM pada *epoch* dengan nilai 100, 250, 500 dan 1000. Proses *training* dilakukan menggunakan *hardware* komputer dengan spesifikasi seperti pada tabel 1.

Tabel 1. Spesifikasi Hardware

Komponen	Spesifikasi
Processor	Intel(R) Core (TM) i7-3520M CPU @ 2.90GHz 2.90 GHz
RAM (DDR4)	8.00 GB
HDD	700 GB
GPU Model	Intel ® HD Graphics 4000

Spesifikasi hardware dalam tabel 1 dipergunakan sebagai acuan estimasi lama waktu proses training dari masing-masing model, proses *training* dapat berlangsung cepat maupun lambat bilamana diproses dalam perangkat keras dengan spesifikasi berbeda [22].

2.5. Evaluasi

Setelah semua model dilatih, langkah selanjutnya adalah mengevaluasi model, evaluasi model dilakukan dengan memperhatikan nilai *akurasi* dan *loss* dari setiap model menggunakan visualisasi dari *tensorboard*. Nilai *akurasi* dan *loss* akan dilihat dan dibandingkan antar model, parameter lain juga akan digunakan untuk mengukur performa model seperti akurasi, recall, presisi dan skor f1 seperti yang terdapat didalam confusion matrix. *Confusion matrix* merupakan metode yang digunakan untuk perhitungan *akurasi*, *recall*, *precision* dan *error date* yang digunakan dalam evaluasi permasalahan klasifikasi dalam *machine learning* dimana outputnya berupa dua atau lebih kelas.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 8. Confusion Matrix

Confusion Matrix merupakan tabel kombinasi yang berbeda antara prediksi dan nilai aktual, yang ditunjukkan dengan TP, TN, FP dan FN.

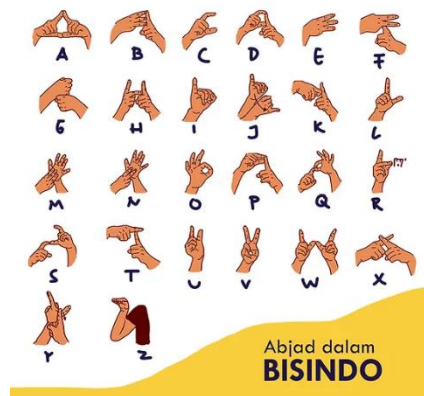
- TP (*True Positive*) : Model diprediksi positif dan itu benar.
- TN (*True Negative*) : Model diprediksi negatif dan itu benar.
- FP (*False Positive*) : Model diprediksi positif dan itu salah.
- FN (*False Negative*) : Model diprediksi negatif dan itu salah

Dari nilai-nilai yang didapatkan maka dapat dilakukan pengukuran turunan lainnya seperti *recall*, *precision*, *accuracy* dan *F-measure*.

$$\begin{aligned}
 Accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\
 Recall &= \frac{TP}{TP+FN} \\
 Precision &= \frac{TP}{TP+FP} \\
 F1 - score &= \frac{2 \times Recall \times Precision}{Recall+Precision}
 \end{aligned}
 \tag{16}$$

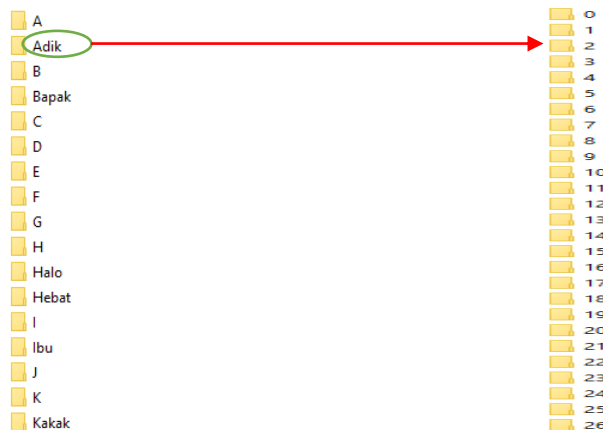
3. HASIL DAN PEMBAHASAN

Sebagaimana yang disampaikan pada bahasan diatas, bahwa penelitian ini dilakukan untuk mendapatkan model yang dipergunakan dalam mendeteksi dan mengenali gerak tangan bahasa isyarat BISINDO. Data gerakan yang digunakan dalam penelitian ini bersumber dari gerak isyarat BISINDO (Bahasa Isyarat Indonesia), yang terdiri dari gerak huruf abjad A – Z (26 gerak isyarat) dan kosakata isyarat yang digunakan sehari-hari seperti yang ditunjukkan dalam gambar 9.



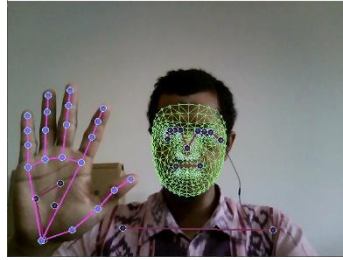
Gambar 9. Gerak tangan huruf abjad Bisindo[26]

Data yang diperggunakan dalam penelitian ini telah mengalami beberapa penyesuaian bentuk selama prose pengumpulan data dan preprocessing. Dalam proses pengumpulan data tekebih dahulu akan dilakukan setting *directory* penyimpanan berupa *folder* sebanyak jenis gerak isyarat yang direkam, setting *directory* dibuat memanfaatkan Bahasa python. Dalam setting *directory* akan dibuat 36 folder sesuai jumlah jenis gerak isyarat yang akan direkam (gerak isyarat huruf abjad A – Z dan kosakata isyarat yang digunakan sehari-hari seperti kata ‘Tolong’, kata ‘Terima kasih’, kata ‘Hebat’, kata ‘Maaf’, kata ‘Halo, kata ‘Oke’, kata ‘Iyo’, kata ‘Ora’, kata ‘aku malu’ menggunakan BISINDO). Pada masing-masing folder akan dibuat lagi 30 folder (0 – 29) yang didasarkan pada jumlah video/gambar masing-masing yang dikumpulkan sebanyak 30 kali.



Gambar 10. Struktur folder data

Setelah dilakukan pembuatan folder selanjutnya dilakukan pengumpulan data memanfaatkan perangkat kamera webcam yang dilakukan dengan menggunakan library python yaitu OpenCV yang dikombinasikan dengan Numpy dan Mediapipe. Dalam perekaman video gerak tangan untuk 36 jenis gerak isyarat yang telah ditentukan akan direkam sebanyak 30 video *sequence* untuk masing-masing jenis gerak isyarat sehingga akan diperoleh sebanyak 1080 video gerak tangan. Disetiap video *sequence* terdapat 30 *frame* yang diperggunakan sebagai gerak isyarat. Dari setiap video yang telah direkam disamping diperoleh pola gerak tangan juga diperoleh *keypoints* dari setiap *frame*, dengan memanfaatkan library Mediapipe Holistic dapat dibuat ekstraksi *keypoints* dari wajah, pose dan kedua tangan, gambar landmark dan deteksi dari setiap frame [23].

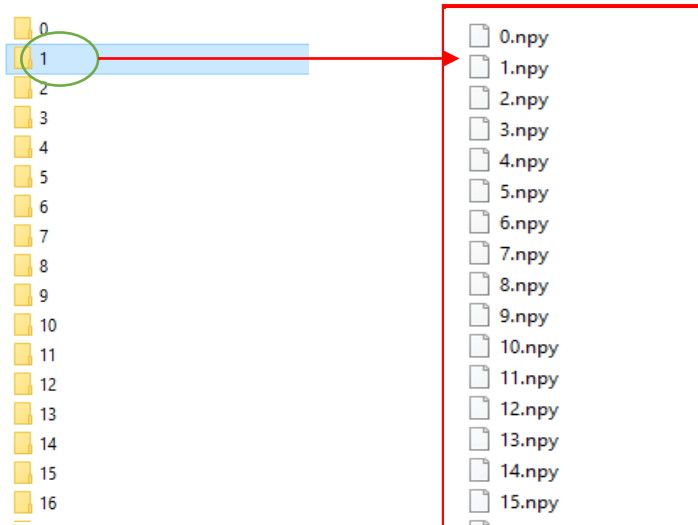


Gambar 11. Deteksi gerak tangan dan pengambilan *landmark*

Keypoint yang diperoleh dari data deteksi gerak tangan dan gambar *landmark* dalam setiap frame selanjutnya diekstraksi dan disimpan dalam bentuk *array* menggunakan *library* *numpy* dan disimpan menggunakan *ekstensi* *np*.

```
array([ 0.52969038,  0.73114651, -1.06150305, ...,  0.11933031,  
        0.74062216, -0.05882541])
```

Gambar 12. Ekstraksi *Keypoints*.



Gambar 13. Menyimpan *keypoints* menjadi file *np*

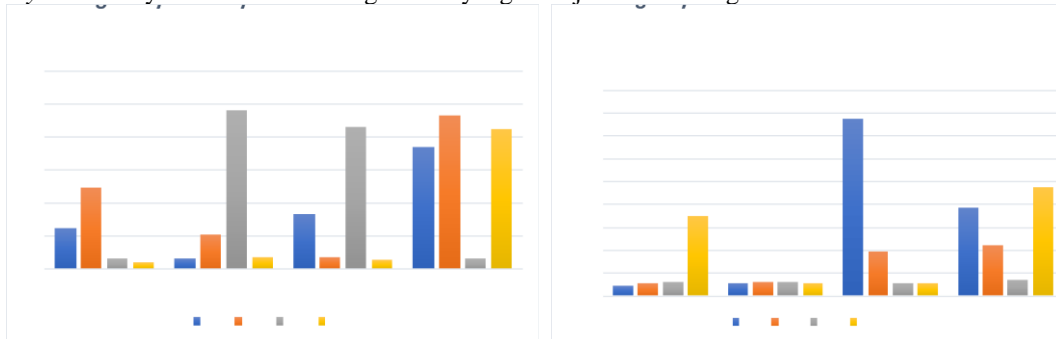
Data hasil *keypoints* yang telah disimpan selanjutnya dalam tahap preprocessing akan digabungkan menjadi satu baris *array Numpy* multi dimensi dengan bentuk (1080,32,1662). *Array* ini dipergunakan untuk menyimpan *variable independen* X yang selanjutnya dipetakan ke *array Numpy* lainnya dari label yang telah dibuat yang menjadi *variable dependen* y, menggunakan *library* Keras, *variable* y yang dikonversi dalam *array biner* yaitu [1, 0, 0, ..., 0, 0, 0]. Setelah mendapatkan *array* untuk *variable* X dan y langkah selanjutnya adalah membagi data menjadi data *train* dan data *test*. Memanfaatkan *library* *scikit-learn* dataset yang telah diperoleh dibagi dengan rasio 8:2 untuk data *train* dan data *test*, dengan jumlah data gerak tangan sebanyak 1080 data, sehingga pembagian dataset menjadi 864 data *train* dan 216 data *test*. Data yang telah dibagi selanjutnya diproses menggunakan metode LSTM.

Data yang telah dibagi selanjutnya diproses dan diklasifikasi menggunakan metode LSTM menjadi model *training* untuk dipergunakan dalam pengenalan gerak tangan. Metode LSTM yang digunakan akan membuat tiga model training yang didasarkan pada tiga model utama LSTM yaitu *single layers* LSTM, *multilayers* LSTM dan *Bidirectional* LSTM. Dalam setiap lapisan LSTM

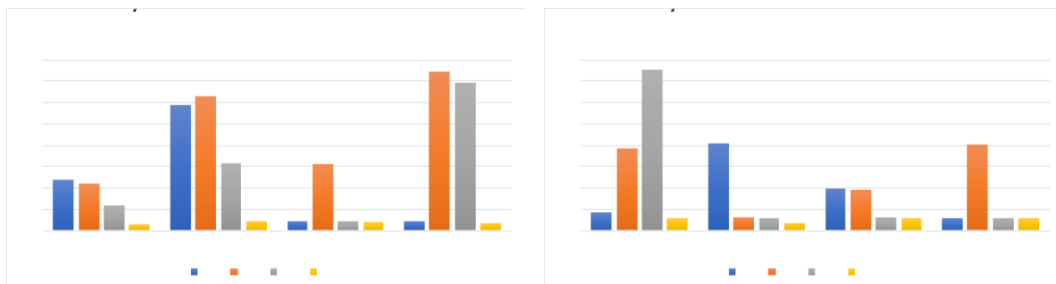
disusun dari 2 *hidden layer* LSTM yaitu 32 dan 64, yang akan ditraining dengan jumlah *batch size* yaitu 32, 64, 128 dan 256, mekanisme *optimizer* adalah Adam, dengan *epoch* senilai 100, 250, 500 dan 1000 dan penentuan *loss* menggunakan *categorical crossentropy*. Dengan menggunakan nilai akurasi dan *loss* yang diperoleh dari proses *training* data yang dilakukan, maka akan ditentukan model deteksi sesuai masing-masing model LSTM yg digunakan.

3.1 Perbandingan Kinerja Model

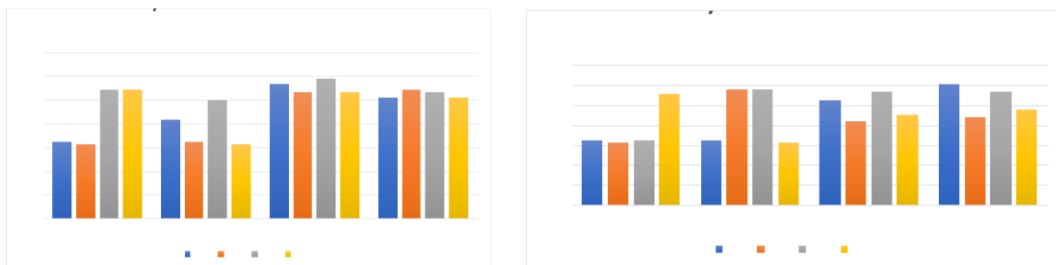
Sebagaimana diketahui bahwa mekanisme evaluasi dari ketiga model didasarkan pada empat matriks evaluasi yaitu *recall*, *precision*, *accuracy* dan *F-measure*. Selama proses *training* data yang telah dilakukan diperoleh nilai akurasi dari masing-masing model yang disusun berdasar 2 *hidden layer* LSTM yaitu 32 dan 64 sebagaimana yang ditunjukkan dalam grafik berikut.



Gambar 14. Nilai akurasi LSTM Single layers



Gambar 15. Nilai akurasi LSTM Multi layers



Gambar 16. Nilai akurasi Bidirectional LSTM

Jika melihat dari gambar 14 diketahui bahwa penggunaan algoritma LSTM *singlelayers* menunjukkan *gap* yang cukup besar pada nilai akurasi antar *batch size*, yang secara tidak langsung menunjukkan kinerja yang belum sesuai standar dalam pengujian. Dalam *training* data menggunakan algoritma LSTM *singlelayer* diperoleh nilai akurasi yang cukup tinggi (diatas 90%) dalam *hidden layers* 32, *batch size* 64 dengan *epoch* 500, nilai akurasi yang cukup tinggi menjadi bukti adanya kesesuaian antara data validasi dengan data *training*.

Dalam gambar 15 hal yang hampir sama ditunjukkan dalam nilai akurasi yang diperoleh pada training data menggunakan algoritma LSTM multilayer, *gap* yang cukup besar ditunjukkan antar *hidden layers* khususnya pada *batch size* 32, 64 dan 128. Besarnya *gap* yang diperoleh secara tidak langsung menunjukkan ketidaksesuaian model LSTM pada *batch size* tersebut dalam pengujian, namun nilai akurasi yang cukup tinggi (diatas 72%) diperoleh pada *hidden layers* 64, *batch size* 32 dengan *epoch* sebesar 500, yang secara tidak langsung menunjukkan kesesuaian model yang digunakan dengan data validasi dengan data *training*.

Sementara itu hal yang berbeda ditunjukkan pada model dengan metode *Bidirectional LSTM*, dalam gambar 16 ditunjukkan grafik akurasi yang diperoleh dengan *gap* yang tidak terlalu besar namun dengan nilai akurasi yang cukup kecil (dibawah 10%). Hal menunjukkan bahwa pengujian data training dan data validasi menunjukkan kesesuaian yang cukup kecil serta kurang menunjukkan korelasi, mekanisme inputan yang hanya menggunakan satu *embedding layer* sebagai *input layer* diduga menjadi salah satu penyebab rendahnya nilai akurasi yang diperoleh sebab sebagaimana yang ditunjukkan dalam gambar 7 diatas *Bidirectional LSTM* mempergunakan 2 lapisan terpisah untuk memperoleh output yang secara tidak langsung juga membutuhkan 2 *input layer*. Nilai akurasi yang cukup baik (diatas 5,9%) dalam model ini diperoleh pada *hidden layers* 32, *batch size* 128 dengan nilai *epoch* 500.

3.2 Pengujian

Model yang telah diperoleh didapatkan dari proses *training* dan *testing* merupakan model LSTM yang ditentukan berdasar nilai akurasi yang ditunjukkan pada gambar 13, 14 dan 15 diatas. Namun disebabkan jumlah deteksi dan pengenalan yang dilakukan cukup banyak, maka penulis hanya menampilkan ilustrasi deteksi dan pengenalan gerak tangan pada model dengan nilai akurasi terbaik. Untuk mendapatkan keakuratan hasil deteksi dan pengenalan secara *real-time*, pada penelitian ini penulis melakukan pendeteksian dan pengenalan gerak tangan sebanyak 6 kali percobaan pada masing-masing model LSTM dengan nilai akurasi terbaik, sebagaimana yang ditunjukkan pada tabel 1, tabel 2 dan tabel 3 berikut.

Tabel 2. . hasil deteksi real-time pada model 13 kelas dengan *singlelayer LSTM epoch 500, hidden layers 32, batch size 64.*

Tes	Kosakata Isyarat												
	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
1	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
2	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
3	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
4	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
5	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
6	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Kosakata isyarat yang berhasil terdeteksi}}{\text{Total keseluruhan percobaan pendeteksi}} \times 100\% & [13] \\
 &= \frac{71}{78} \times 100\% \\
 &= 0.9103
 \end{aligned}$$

Tabel 3. hasil deteksi real-time pada model 13 kelas dengan *multilayers LSTM epoch 500, hidden layers 64, batch size 32.*

Test	Kosakata Isyarat												
	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
1	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
2	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
3	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
4	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
5	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
6	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Kosakata isyarat yang berhasil terdeteksi}}{\text{Total keseluruhan percobaan pendeteksi}} \times 100\% & [13] \\
 &= \frac{67}{78} \times 100\% \\
 &= 0.8589
 \end{aligned}$$

Tabel 2 dan tabel 3 menunjukkan hasil deteksi secara *real time* yang dilakukan sebanyak 6 kali percobaan dengan 13 kelas menggunakan model *single layer LSTM epoch 500, hidden layers 32, batch size 64* dan model *multilayers LSTM epoch 500, hidden layers 64, batch size 32*. Dalam pengujian yang dilakukan menggunakan model *single layer LSTM epoch 500, hidden layers 32, batch size 64* mendapatkan nilai akurasi sebanyak 91% sedangkan dalam model *multilayers LSTM epoch 500, hidden layers 64, batch size 32* mendapatkan nilai akurasi sebanyak 86%. Keberhasilan dan kegagalan pengklasifikasian gestur kosakata isyarat dipengaruhi oleh beberapa faktor seperti gestur kosakata isyarat yang dideteksi apakah memiliki kesamaan dengan gestur kosakata isyarat yang di-*training*, pose tubuh saat pendeteksian, posisi tangan, jarak kamera serta pencahayaan, disamping model yang dipergunakan.

Tabel 4. hasil deteksi real-time pada model 12 kelas dengan *bidirectional LSTM epoch 500, hidden layers 32, batch size 128*.

Test	Kosakata Isyarat												
	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
1	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
2	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
3	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
4	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
5	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih
6	Halo	Selamat_Pagi	Hati-hati	Baik_hati	Maaf	Oke	Sabar	Tolong	Bapak	Ibu	A	B	Terima_kasih

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Kosakata isyarat yang berhasil terdeteksi}}{\text{Total keseluruhan percobaan pendeteksi}} \times 100\% & [13] \\
 &= \frac{42}{78} \times 100\% \\
 &= 0.5385
 \end{aligned}$$

Sedangkan dalam tabel 4 menunjukkan hasil deteksi secara *real time* yang dilakukan sebanyak 6 kali percobaan dengan 13 kelas pada model *bidirectional LSTM epoch 500, hidden layers 32,*

batch size 128. Kegagalan pengklasifikasian gestur kosakata isyarat pada model ini lebih banyak dipengaruhi oleh nilai akurasi model yang cukup rendah, sehingga berpengaruh dalam pengenalan gesture dan pose tubuh, disamping faktor lain seperti pencahayaan yang kurang merata. Dalam pengujian yang dilakukan dengan 6 kali percobaan mendapatkan nilai akurasi sebanyak 54%.

Tabel 5. Beberapa hasil deteksi real time yang dilakukan dalam percobaan ini

Gambar	Hasil Deteksi
	Berhasil mendeteksi kosakata isyarat "Oke"
	Berhasil mendeteksi kosakata isyarat "Maaf", namun gesture "Maaf" tertumpuk dengan gesture "Selamat pagi" dikarenakan posisi tangan yang sama dengan posisi perekaman gesture "Selamat pagi"
	Gesture "Selamat pagi" dideteksi sebagai "Halo", disebabkan posisi tangan yang sama dengan posisi perekaman gesture "Halo"

Berdasarkan deteksi yang dilakukan diperoleh model terbaik dalam pengujian ini merupakan model *single layer LSTM*. Nilai akurasi yang diperoleh model ini cukup tinggi yaitu 91%, sebanding dengan nilai akurasi yang diperoleh selama proses pembentukan model yang memiliki nilai akurasi lebih tinggi dari model yang lainnya. Namun dalam deteksi yang dilakukan terdapat beberapa gesture kosakata yang cukup sulit untuk dideteksi sistem dengan kemungkinan deteksi yang rendah, seperti gesture "maaf" dan "tolong". Hal ini disebabkan oleh gesture yang memiliki kesamaan dengan gesture lainnya, posisi tangan yang sama dengan posisi gesture lainnya serta nilai akurasi model yang kurang baik.

Menyangkut kualitas, model terbaik yang diperoleh dalam penelitian ini belum cukup andal dalam melakukan deteksi gesture berbentuk kalimat, disebabkan kurangnya data dan kurangnya variasi data yang digunakan sehingga semakin model yang diperoleh hanya mampu mendeteksi gesture kosakata. Untuk alasan ini maka dalam penelitian lanjutan perlu dipertimbangkan penggunaan data dalam bentuk kalimat serta penambahan variasi data *training*.

4. KESIMPULAN

Penelitian ini dilakukan untuk mendapatkan model yang mampu mengenali kosakata bahasa isyarat BISINDO. Dengan memanfaatkan *library MediaPipe Holistic* dalam mengekstraksi *keypoint*

pada wajah, pose dan tangan, untuk selanjutnya diolah menggunakan metode LSTM. Dari hasil percobaan yang telah dilakukan dapat diketahui bahwa dari 3 model yang diperoleh dari 3 metode LSTM yang digunakan memiliki nilai akurasi yang berbeda-beda, dimana model *single layer* LSTM *epoch* 500, *hidden layers* 32, *batch size* 64 yang memiliki akurasi terbesar mencapai 91%. Namun model yang diperoleh ini masih jauh dari sempurna, dimana model ini belum cukup handal dalam melakukan deteksi bahasa isyarat dalam bentuk kalimat, mengingat data yang dipergunakan merupakan data kosakata. Dalam penelitian selanjutnya diharapkan adanya peneliti yang menggunakan metode serupa namun dengan data dalam bentuk kalimat dan lebih variatif.

DAFTAR PUSTAKA

- [1] N. Abe, Institute of Electrical and Electronics Engineers, and IEEE Computer Society, 2018 *IEEE International Conference on Big Data : proceedings : Dec 10 - Dec 13, 2018, Seattle, WA, USA*.
- [2] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A. B. Gil-González, and J. M. Corchado, "Deepsign: Sign Language Detection and Recognition Using Deep Learning," *Electronics (Switzerland)*, vol. 11, no. 11, Jun. 2022, doi: 10.3390/electronics11111780.
- [3] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion," *IEEE Sens J*, vol. 19, no. 16, pp. 7056–7063, Aug. 2019, doi: 10.1109/JSEN.2019.2909837.
- [4] K. Gomase, A. Dhanawade, P. Gurav, and S. Lokare, "Sign Language Recognition using Mediapipe," *International Research Journal of Engineering and Technology*, 2021, [Online]. Available: www.irjet.net
- [5] A. Ojha, A. Pandey, S. Maurya, and A. Thakur, "Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network." [Online]. Available: www.ijert.org
- [6] S. Chakraborty, S. Banerjee, N. Bandyopadhyay, Z. Sarkar, and P. Chakraverty, "Indian Sign Language Classification (ISL) using Machine Learning." [Online]. Available: www.ajec.smartsociety.org
- [7] S. Kim, K. Yun, J. Park, and J. Y. Choi, "Skeleton-based action recognition of people handling objects," in *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Institute of Electrical and Electronics Engineers Inc.*, Mar. 2019, pp. 61–70. doi: 10.1109/WACV.2019.00014.
- [8] M. Kamal Wisyaldin, G. Maya Luciana, H. Pariaman, and P. Pembangunan Jawa Bali, "Pendekatan Long Short-Term Memory untuk Memprediksi Kondisi Motor 10 kV pada PLTU Batubara," vol. 9, no. 2, doi: 10.33322/kilat.v9i2.997.
- [9] H. Moetia Putri and W. Fuadi, "PENDETEKSIAN BAHASA ISYARAT INDONESIA SECARA REAL-TIME MENGGUNAKAN LONG SHORT-TERM MEMORY (LSTM)."
- [10] K. Halim and E. Rakun, "Sign language system for Bahasa Indonesia (Known as SIBI) recognizer using TensorFlow and long short-term memory," in *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*,

-
- Institute of Electrical and Electronics Engineers Inc., Jan. 2019, pp. 403–407. doi: 10.1109/ICACSSIS.2018.8618134.
- [11] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, “Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting,” in *Procedia Computer Science*, Elsevier B.V., 2018, pp. 89–98. doi: 10.1016/j.procs.2018.08.153.
- [12] C.-C. Chiu et al., “State-of-the-art Speech Recognition With Sequence-to-Sequence Models,” Dec. 2017, [Online]. Available: <http://arxiv.org/abs/1712.01769>
- [13] D. Deb and V. E. Balas, “Rajeeb Dey Editors Innovations in Infrastructure Proceedings of ICIIF,” 2018. [Online]. Available: <http://www.springer.com/series/11156>
- [14] G. Budiprasetyo, M. Hani’ah, and D. Z. Aflah, “Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM),” *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 3, pp. 164–172, Jan. 2023, doi: 10.25077/teknosi.v8i3.2022.164-172.
- [15] S. Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti, W. Kurnia Sari, D. Palupi Rini, R. Firsandaya Malik, and I. B. Saladin Azhar, “Terakreditasi SINTA Peringkat 2 Klasifikasi Teks Multilabel pada Artikel Berita Menggunakan Long Short-Term Memory dengan Word2Vec,” *masa berlaku mulai*, vol. 1, no. 3, pp. 276–285, 2017.
- [16] R. Auziqni, “BISINDO INDONESIAN SIGN LANGUAGE RECOGNITION USING MEDIPIPE HOLISTIC AND LSTM DEEP LEARNING MODEL Thesis Report.” [Online]. Available: <http://digilib.mercubuana.ac.id/>
- [17] H. D. Trinh, L. Giupponi, and P. Dini, “Mobile Traffic Prediction from Raw Data Using LSTM Networks,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 1827–1832. doi: 10.1109/PIMRC.2018.8581000.
- [18] H. Elfaik and E. H. Nfaoui, “Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text,” *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 395–412, Jan. 2021, doi: 10.1515/jisys-2020-0021.
- [19] D. R. Alghifari, M. Edi, and L. Firmansyah, “Implementasi Bidirectional LSTM untuk Analisis Sentimen Terhadap Layanan Grab Indonesia,” *Jurnal Manajemen Informatika (JAMIKA)*, vol. 12, no. 2, pp. 89–99, Sep. 2022, doi: 10.34010/jamika.v12i2.7764.
- [20] H. F. Fadli and A. F. Hidayatullah, “Identifikasi Cyberbullying pada Media Sosial Twitter Menggunakan Metode LSTM dan BiLSTM.”
- [21] Utomo, P. B., Wahyudi, D., & Nalendra, A. K. (2024). Implementasi Convolution-Augmented Transformer Berbasis Kecerdasan Buatan dalam Analisis Sentimen Teks Hasil Konversi Suara ke Teks. *Generation Journal*, 8(1), 63-71. <https://doi.org/10.29407/gj.v8i1.22202>
- [22] J. Khatib Sulaiman, A. Fikri Hanif, T. Bayu Sasongko, A. Dwi Laksito, and U. Amikom Yogyakarta, “Perbandingan Kinerja LSTM, Bi-LSTM, dan GRU pada Klasifikasi Judul

- Berita Clickbait,” Indonesian Journal of Computer Science Attribution, vol. 12, no. 4, p. 2136.
- [23] W. Zhu et al., “Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks,” Mar. 2016, [Online]. Available: <http://arxiv.org/abs/1603.07772>
- [24] <https://www.cnnindonesia.com/gaya-hidup/20170323180138-255-202335/masih-ada-405-juta-orang-indonesia-berjuang-lawan-ketulian> (diakses pada 10 Januari 2023)
- [25] <https://www.republika.co.id/berita/pfwt1n313/islam-beri-perhatian-penyandang-disabilitas> (diakses pada 10 Januari 2023)
- [26] [https://www.klobility.id/post/perbedaan-bisindo-dan-sibi#:~:text=BISINDO%20\(Bahasa%20Isyarat%20Indonesia\)&text=BISINDO%20dibentuk%20oleh%20kelompok%20Tuli,disampaikan%20dengan%20gerakan%20dua%20tangan](https://www.klobility.id/post/perbedaan-bisindo-dan-sibi#:~:text=BISINDO%20(Bahasa%20Isyarat%20Indonesia)&text=BISINDO%20dibentuk%20oleh%20kelompok%20Tuli,disampaikan%20dengan%20gerakan%20dua%20tangan). Diakses pada 17 Februari 2023