
Studi Klasifikasi dengan KNN dan ANN Pada Sarung Tangan Penerjemah Angka dan Alfabet Bahasa Isyarat SIBI

Kevin Christopher Wungow^{1*}, Romy Budhi Widodo¹, Mochamad Subianto²

¹Pusat Studi Human-Machine Interaction, Teknik Informatika, Universitas Ma Chung,

²Teknik Informatika, Universitas Ma Chung, Jl. Villa Puncak Tidar Blok N-1, Malang, Indonesia 65151

Correspondence: Kevin Christopher Wungow (311810018@student.machung.ac.id)

Received: 23 07 22 – Revised: 01 08 22 - Accepted: 04 08 22 - Published: 09 09 22

Abstrak. Penyandang disabilitas di Indonesia khususnya pada jenis disabilitas tunawicara dan tunarungu pada tahun 2018, yaitu 0,15% untuk tuna wicara dan 0,11% untuk tuna rungu dari total masyarakat Indonesia. Berdasarkan jumlah penyandang tunawicara dan tunarungu yang ada di Indonesia jika menggunakan alat penerjemah bahasa isyarat tentu akan mempermudah komunikasi antara penyandang dengan non penyandang. Hal tersebut disebabkan oleh kurang pemahamannya orang tentang bahasa isyarat yaitu Sistem Bahasa Isyarat Indonesia dan Bahasa Isyarat Indonesia (BISINDO). Pada penelitian kali ini fokus pada bahasa Sistem Bahasa Isyarat Indonesia (SIBI) yang digunakan sebagai penerjemah angka dan alfabet dengan menggunakan sarung tangan yang dilengkapi dengan sensor tekuk. Penelitian ini bertujuan untuk mengklasifikasi angka 1 sampai 10 dan huruf A sampai Z, serta menggunakan metode K-Nearest Neighbors dan Artificial Neural Network, serta dilakukan pengujian secara real-time. Total data sampel yang digunakan pada penelitian ini sebanyak 288 buah. Pengambilan data dilakukan dengan cara subjek menggerakkan gestur angka 1 sampai 10 dan huruf A sampai Z sebanyak 2x jadi masing masing subjek memiliki data sampel sebanyak 72 buah. Pembuatan program klasifikasi ini dilakukan menggunakan bahasa pemrograman Python dan pustaka-pustaka Python lainnya. Tujuan yang ingin dicapai dalam penelitian ini telah berhasil dilakukan yaitu membuat program aplikasi yang dapat mengklasifikasi gestur tangan dengan menggunakan metode KNN serta mengujinya secara real time dan pembuatan model ANN. Hasil akurasi model ANN menghasilkan akurasi 93% dengan loss 0,2134. untuk model KNN dengan nilai $k = 7$ menghasilkan akurasi 99%. Pengujian real-time diuji menggunakan model KNN dengan nilai $k = 7$ mendapatkan akurasi rata-rata 75%.

Kata kunci: *artificial neural network (ann), flex point glove kit, flex sensor, k-nearest neighbors (knn), machine learning.*

Citation Format: Wungow, K.C., Widodo, R.M., & Subianto, M. (2022). Studi Klasifikasi dengan KNN dan ANN Pada Sarung Tangan Penerjemah Angka dan Alfabet Bahasa Isyarat SIBI. *Prosiding Seminar Nasional Universitas Ma Chung*, 60-74.

PENDAHULUAN

Penyandang disabilitas adalah orang yang mengalami keterbatasan fisik, intelektual, mental, atau sensorik dalam jangka waktu lama yang dalam berinteraksi dengan lingkungan masyarakat mengalami kesulitan untuk berpartisipasi aktif dengan masyarakat lainnya (Bahri, 2018). Adapun jenis dari penyandang disabilitas yaitu ketidakfungsian organ pendengaran atau fungsi berbicara dengan baik (tunarungu atau tunawicara). Dari jenis disabilitas diatas, penelitian ini berkontribusi pada penyandang tunarungu dan tunawicara dengan cara mengembangkan alat untuk memudahkan berkomunikasi antara penyandang dengan non penyandang.

Penyandang disabilitas di Indonesia pada umur 24 sampai 59 bulan di Indonesia khususnya pada jenis disabilitas tunawicara dan tunarungu pada tahun 2018, yaitu 0,15% untuk tuna wicara dan 0,11% untuk tuna rungu dari total populasi masyarakat Indonesia (Kementerian Kesehatan, 2018). Berdasarkan jumlah dari penyandang tunawicara dan tunarungu yang ada di Indonesia tersebut jika menggunakan alat penerjemah bahasa isyarat tentu akan mempermudah komunikasi antara penyandang dengan non penyandang. Hal tersebut disebabkan oleh kurang pahamnya kebanyakan orang tentang bahasa isyarat yaitu bahasa Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). Pada penelitian kali ini penulis fokus pada bahasa Sistem Bahasa Isyarat Indonesia (SIBI) yang digunakan sebagai penerjemah angka dan alfabet.

Fokus dari penelitian ini adalah untuk mengembangkan alat dari penelitian sebelumnya yaitu penerjemah bahasa isyarat dengan menggunakan USB Glove kit yang terpasang dengan sensor tekuk yang datanya dapat diambil dan diolah dalam komputer. Pada penelitian sebelumnya hanya mampu menerjemahkan angka 1 sampai 10 dengan metode KNN secara real time. Untuk pengembangan yang ingin dilakukan penulis adalah penambahan kemampuan USB Glove kit untuk menerjemahkan angka dan alfabet serta membandingkan metode KNN dengan metode ANN (Widodo, et al., 2021).

Metode machine learning yang sesuai digunakan penelitian ini adalah k-nearest neighbor (KNN). Beberapa karakteristik KNN yang sesuai dengan sumber daya penelitian ini yaitu output target: diskrit, sifat data: tidak ada nilai data yang hilang, dimensi data terbatas, metodologi: supervised, computing resources: sedang, dan Akurasi: sedang (Widodo, et al., 2021).

MASALAH

Kurangnya pemahaman orang awam terhadap bahasa isyarat, oleh karena itu pembuatan sarung tangan penerjemah digunakan untuk mempermudah komunikasi antara penyandang disabilitas dengan non penyandang disabilitas.

METODE PELAKSANAAN

Seperti pada Gambar 1, alur sistem dimulai dari pembacaan data masukan sensor oleh USB Glove kit yang sudah terpasang dengan mikrokontroler dan sensor bend. Pergerakan tangan yang melakukan gestur bahasa isyarat akan diterima oleh sensor dan akan diteruskan ke komputer sebagai input data tersebut disimpan dalam bentuk file .csv. File dari hasil pengumpulan data dikelompokkan berdasarkan subjek dan gestur apa yang sedang digerakan contohnya pada penelitian ini dengan nama file “c10” yang berarti c adalah subjek ketiga dan 10 adalah gestur tangan yang diperagakan oleh subjek ketiga. Setelah semua data dikelompokkan untuk mempermudah pengolahan data selanjutnya dilakukan proses pembelajaran model machine learning, pada penelitian ini metode yang digunakan adalah metode ANN dan metode KNN. Setelah pembuatan model KNN telah selesai dilakukan maka langkah selanjutnya masuk ketahap klasifikasi secara real time dan hasilnya akan ditampilkan dilayar monitor menggunakan aplikasi dari Tkinter.

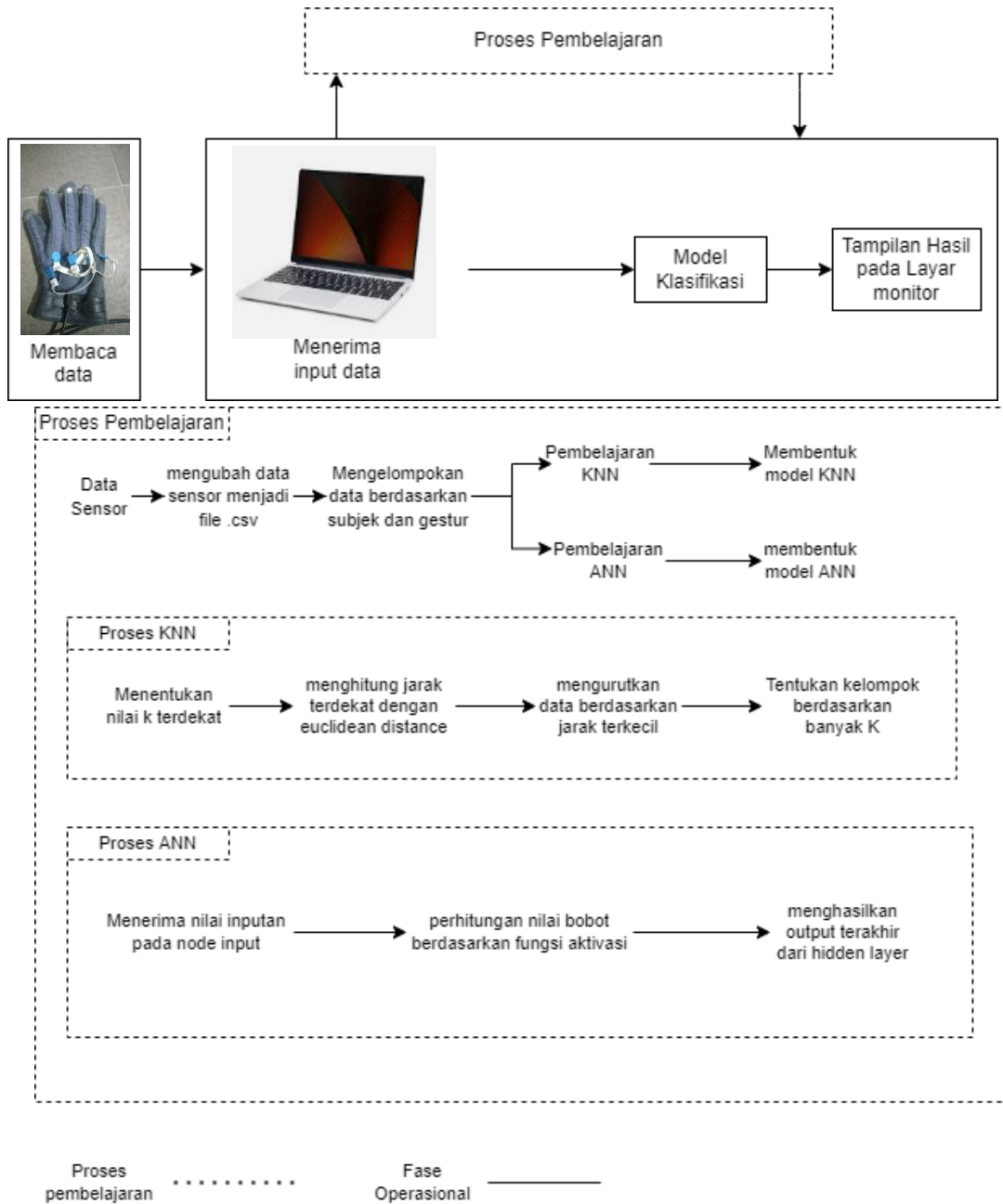
Perangkat USB glove kit yang terpasang dengan sensor bend memiliki banyak fitur berikut adalah penjelasan data apa saja yang dibaca oleh sensor-sensor yang ada pada perangkat tersebut,

Metacarpophalangeal: tekukan jari yang terdapat pada telapak tangan

Proximal interphalangeal: tekukan jari setelah metacarpophalangeal

- 0 : tekukan metacarpophalangeal jari kelingking
- 1 : tekukan proximal interphalangeal jari kelingking
- 2 : tekukan metacarpophalangeal jari manis
- 3 : tekukan proximal interphalangeal jari manis
- 4 : tekukan metacarpophalangeal jari tengah
- 5 : tekukan proximal interphalangeal jari tengah
- 6 : tekukan metacarpophalangeal jari telunjuk
- 7 : tekukan proximal interphalangeal jari telunjuk
- 8 : tekukan metacarpophalangeal ibu jari
- 9 : tekukan proximal interphalangeal ibu jari
- 10 : sudut X pembacaan sensor akselerometer

- 11 : sudut Y pembacaan sensor akselerometer
- 12 : sudut Z pembacaan sensor akselerometer
- 13 : sudut X pembacaan sensor magnetometer
- 14 : sudut Y pembacaan sensor magnetometer
- 15 : sudut Z pembacaan sensor magnetometer



Gambar 1 Ilustrasi Proses Klasifikasi (Santoso, 2022)

HASIL DAN PEMBAHASAN

Pada bagian ini dilaporkan hasil beberapa tahapan dalam desain machine learning, dataset terdiri dari 14 kolom

Proses Pengambilan Data

Untuk Langkah awal adalah pengambilan data, pengambilan data dilakukan menggunakan kode yang sudah dibuat oleh peneliti terdahulu (Santoso, 2022). Kode yang sudah ada tersebut diubah sesuai dengan yang dibutuhkan pada penelitian ini seperti merubah penyaringan regular expression untuk menampilkan 13 fitur data yang dibutuhkan. Langkah awal adalah subjek menggunakan sarung tangan pada tangan kanan dan memastikan bahwa sensor bend sudah pada tempatnya dan sudah terhubung dengan baik. Kemudian melihat port yang masuk pada serial berapa dapat langsung di lihat pada device manager setelah selesai maka pengambilan data siap dilakukan dengan menggerakkan tangan sesuai dengan gestur pada penelitian ini yaitu angka dan huruf abjad bahasa isyarat SIBI. Setelah melakukan pergerakan maka data akan langsung disimpan dalam komputer dengan tipe file .csv. Untuk tampilan saat pengambilan data dapat dilihat pada Gambar 2 dan Gambar 3.



Gambar 2. Subjek melakukan gestur tangan

```

Enter Gesture Name :
a1
Press Enter to Start Recording..
['322', '94', '490', '1218', '556', '0', '0', '156', '1299', '312', '3488', '-656', '16368']
['322', '94', '490', '1218', '556', '0', '0', '156', '1140', '312', '3488', '-656', '16368']
['161', '94', '490', '1218', '556', '0', '0', '156', '1140', '312', '3584', '-1296', '15968']
['161', '94', '490', '1218', '494', '0', '0', '0', '1140', '208', '3584', '-1296', '15968']
['161', '0', '490', '1218', '556', '0', '0', '0', '1140', '208', '3760', '-880', '16224']
['161', '94', '490', '1218', '494', '0', '0', '0', '1140', '312', '3760', '-880', '16224']
['161', '0', '490', '1218', '494', '0', '0', '0', '1140', '312', '3632', '-896', '16448']
['161', '0', '490', '1218', '494', '0', '0', '0', '1140', '208', '3632', '-896', '16448']
['322', '94', '490', '1218', '556', '51', '0', '156', '1140', '312', '3632', '-1072', '16192']
['161', '0', '490', '1218', '494', '0', '0', '0', '1140', '208', '3632', '-1072', '16192']
['161', '94', '490', '1218', '556', '51', '0', '156', '1140', '312', '3632', '-1072', '16192']
['161', '0', '490', '1218', '494', '51', '0', '0', '1140', '208', '3456', '-1008', '16192']
['322', '94', '490', '1218', '556', '51', '0', '156', '1140', '312', '3456', '-1008', '16192']
['161', '94', '490', '1218', '494', '51', '0', '0', '1140', '208', '3664', '-736', '16336']
['322', '94', '490', '1218', '556', '51', '0', '156', '1299', '312', '3664', '-736', '16336']
['161', '94', '490', '1218', '556', '51', '0', '156', '1140', '312', '3504', '-960', '16176']
['161', '94', '490', '1218', '556', '51', '0', '0', '1140', '312', '3504', '-960', '16176']
['161', '0', '490', '1218', '494', '51', '0', '0', '1140', '208', '3552', '-784', '16240']
['161', '0', '490', '1218', '494', '0', '0', '0', '1140', '208', '3552', '-784', '16240']
['161', '94', '490', '1218', '494', '51', '0', '0', '1140', '208', '3520', '-688', '16496']
['161', '0', '490', '1218', '494', '51', '0', '0', '1140', '208', '3520', '-688', '16496']
['161', '94', '490', '1218', '556', '51', '0', '156', '1140', '312', '3712', '-928', '16384']
['322', '94', '490', '1218', '556', '51', '0', '156', '1140', '208', '3712', '-928', '16384']
['161', '94', '490', '1096', '494', '51', '227', '0', '1140', '208', '3440', '-784', '16384']
['322', '94', '490', '1218', '556', '51', '227', '156', '1140', '312', '3440', '-784', '16384']
['161', '94', '490', '1218', '556', '51', '227', '0', '1140', '312', '3344', '-848', '16336']
['161', '94', '490', '1218', '556', '51', '227', '156', '1140', '312', '3344', '-848', '16336']

```

Gambar 3. Tampilan proses pengambilan data

Proses Pengolahan Data

Pengolahan data menggunakan google colab untuk mempermudah pengolahan data. Langkah awal data yang sudah diambil maka akan dimasukan kedalam google colab dan akan diberi nilai output untuk memudahkan dalam klasifikasi, karena pada data mentah nilai sensor tidak memiliki klasifikasi output. Setelah selesai pada pemberian klasifikasi output maka selanjutnya data akan disatukan semuanya dan diberi label, karena pada data mentah nilai sensor belum memiliki label. Pemberian label dapat mempermudah saat pengolahan data. Label yang digunakan pada data ini adalah sebagai berikut:

- mcpPinky : tekukan metacarpo-phalangeal jari kelingking
- pipPinky : tekukan proximal interphalangeal jari kelingking
- mcpRing : tekukan metacarpo-phalangeal jari manis
- pipRing : tekukan proximal interphalangeal jari manis
- mcpMid: tekukan metacarpo-phalangeal jari tengah
- pipMid : tekukan proximal interphalangeal jari tengah
- mcpIndex : tekukan metacarpo-phalangeal jari telunjuk
- pipIndex : tekukan proximal interphalangeal jari telunjuk
- mcpThumb : tekukan metacarpo-phalangeal ibu jari
- pipThumb : tekukan proximal interphalangeal ibu jari
- accX : sudut X pembacaan sensor akselerometer

accY	: sudut Y pembacaan sensor akselerometer
accZ	: sudut Z pembacaan sensor akselerometer
Output	: Klasifikasi output sensor

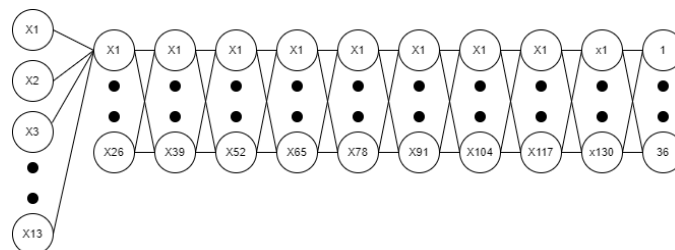
Untuk klasifikasi output sensor, pada penelitian ini menggunakan encoding untuk mempermudah pada saat pengolahan data untuk daftar encoding pada nilai output adalah sebagai berikut:

1	: 1	19	: I
2	: 2	20	: J
3	: 3	21	: K
4	: 4	22	: L
5	: 5	23	: M
6	: 6	24	: N
7	: 7	25	: O
8	: 8	26	: P
9	: 9	27	: Q
10	: 10	28	: R
11	: A	29	: S
12	: B	30	: T
13	: C	31	: U
14	: D	32	: V
15	: E	33	: W
16	: F	34	: X
17	: G	35	: Y
18	: H	36	: Z

Setelah selesai pemberian label langkah selanjutnya adalah pengecekan kembali apakah data terdapat nilai 0 atau missing value, setelah data semua sudah selesai maka akan masuk ketahap selanjutnya yaitu split data.

Proses Pembuatan Model ANN

Pembuatan model ANN pada penelitian ini menggunakan Google Colab dengan bahasa python. Untuk dataset yang digunakan saat pembuatan model ANN adalah dengan menggunakan file data latih dan data uji yang sudah disiapkan diambil dari total file yaitu 288 file yang sudah disatukan. Data ini juga merupakan data yang sama digunakan untuk pembuatan model KNN. Pertama yang dilakukan adalah melakukan import library Tensor Flow dan Keras yang diperlukan pada saat pembuatan model. Setelah library berhasil untuk dimasukkan maka langkah pertama adalah membaca hasil dataset yang sudah diberi label class. Selanjutnya dilakukan transform data dengan menggunakan one hot encoding. One hot encoding disini digunakan untuk membuat kolom dari output yang awalnya dari single kolom menjadi 36 kolom sesuai dengan klasifikasi pada penelitian ini. Setelah itu kolom disatukan kembali pada tabel utama. Kemudian langkah selanjutnya adalah menentukan X dan Y pada dataset untuk X adalah 13 fitur dari sensor dan Y adalah 36 klasifikasi output yang sudah di encoding menjadi 36 kolom selanjutnya membagi data menjadi data uji dan data latih, untuk data uji pada model ANN adalah 30% dan data latih 70%. Hasil split dari library Scikit-Learn kemudian disimpan dalam bentuk dataframe dan diubah ke bentuk csv. Kemudian 4 buah csv yaitu xtraining, xtest, ytraining, dan ytest, disimpan untuk nantinya akan digunakan juga untuk proses pembelajaran KNN pada subbab selanjutnya. Kemudian dilakukan pembuatan model ANN dengan menentukan node, input layer, hidden layer, output layer dan fungsi aktivasi. Untuk arsitektur ANN pada penelitian ini dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur ANN

Dapat dilihat pada Gambar 4 arsitektur ANN memiliki 13 node pada layer input yaitu sesuai dengan fitur masukan pada sensor yaitu:

x1: MCP (metacarpo-phalangeal) kelingking

x2: PIP (proximal interphalangeal) kelingking

x3: MCP jari manis

x4: PIP jari manis

x5: MCP jari tengah

x6: PIP jari tengah

x7: MCP jari telunjuk

x8: PIP jari telunjuk

x9: MCP ibu jari

x10: PIP ibu jari

x11: Accelerometer sumbu x

x12: Accelerometer sumbu y

x13: Accelerometer sumbu z

Sedangkan untuk hidden layer terdapat 9 layer dengan masing masing tiap nodenya adalah 26, 39, 52, 65, 78, 91, 104, 117, 130 dengan menggunakan fungsi aktivasi ReLu. Pada layer terakhir yaitu layer output pada penelitian ini menggunakan 36 sesuai dengan klasifikasi angka yang berjumlah 10 dan huruf abjad yang berjumlah 26 dengan menggunakan fungsi aktivasi softmax. Kemudian setelah itu dilakukan compile dengan parameter optimizer adam, loss categorical crossentropy dan metrics accuracy.

Setelah dilakukan compile maka tahap selanjutnya adalah dilakukan training model dengan batch size 32 dan 100 epoch untuk tampilan Gambar 5 merupakan contoh dari proses Training pada ANN. Dengan validasi X uji dan Y uji. Setelah model berhasil dibuat dan dirasa memiliki akurasi dan loss yang cukup baik maka dilakukan save model. Kemudian model yang sudah disave dilakukan load model untuk dilakukan uji coba pada data uji. Sebelum melakukan uji model dilakukan inverse transform encoding pada klasifikasi output agar hasil output mudah diamati, untuk hasil dari uji model dapat dilihat pada Gambar 6 Hasil dari model ANN yang dibuat memiliki jumlah loss 0,1669 dan akurasi training 0,9541.

```
[7.2822309e-13 2.1943779e-12 9.7238916e-16 1.4349627e-10 8.9216629e-35
1.1981744e-21 2.2786142e-37 3.7295334e-26 3.4180622e-29 6.6442962e-09
8.4556291e-09 1.3111612e-07 7.1100708e-34 1.8104069e-09 2.7275262e-09
1.2240193e-07 3.1550506e-08 1.2507395e-07 8.7445778e-06 1.4688414e-20
2.3144942e-04 3.0727648e-07 6.7157909e-02 9.3104064e-01 1.5287226e-03
6.3134231e-09 3.8010364e-09 5.1069077e-13 2.3992359e-06 1.5109347e-05
1.4343397e-05 3.5351071e-34 1.0992002e-14 3.6591927e-10 1.3127275e-11
1.7628087e-08]
```

24

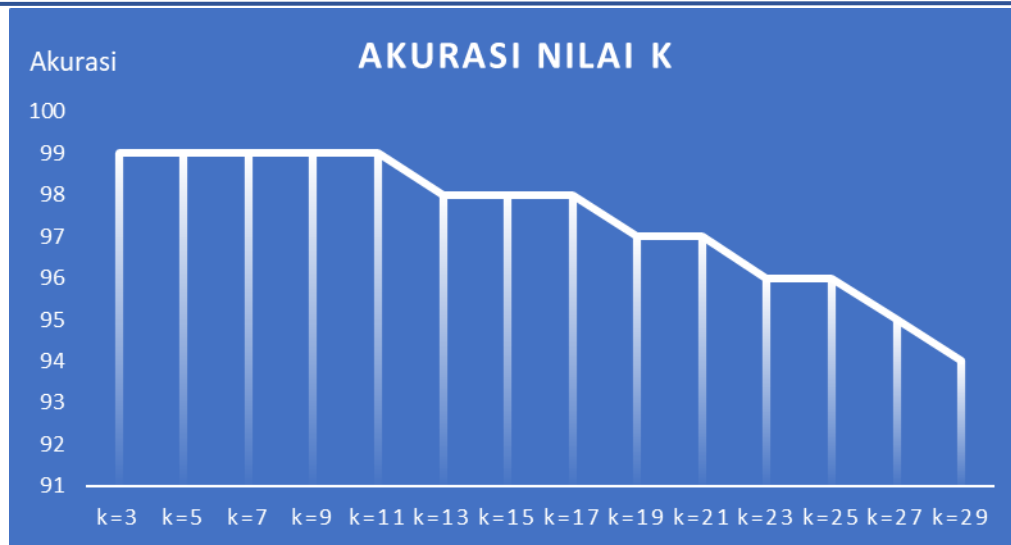
Gambar 5. Array hasil dari node output softmax

```
Epoch 95/100
630/630 [=====] - 3s 5ms/step - loss: 0.2556 - accuracy: 0.9256 - val_loss: 0.2118 - val_accuracy: 0.9331
Epoch 96/100
630/630 [=====] - 2s 4ms/step - loss: 0.2713 - accuracy: 0.9198 - val_loss: 0.3008 - val_accuracy: 0.9286
Epoch 97/100
630/630 [=====] - 3s 4ms/step - loss: 0.4451 - accuracy: 0.8822 - val_loss: 0.6973 - val_accuracy: 0.7928
Epoch 98/100
630/630 [=====] - 3s 4ms/step - loss: 0.3710 - accuracy: 0.8940 - val_loss: 0.2844 - val_accuracy: 0.9162
Epoch 99/100
630/630 [=====] - 3s 5ms/step - loss: 0.2519 - accuracy: 0.9247 - val_loss: 0.2558 - val_accuracy: 0.9160
Epoch 100/100
630/630 [=====] - 3s 5ms/step - loss: 0.1711 - accuracy: 0.9461 - val_loss: 0.1669 - val_accuracy: 0.9541
```

Gambar 6. Proses training ANN

Proses Pembuatan Model KNN

Pembuatan model KNN pada penelitian ini menggunakan bantuan Google Colab dengan bahasa python. Untuk dataset yang digunakan saat pembuatan model KNN adalah dengan menggunakan 4 buah file csv pada proses pembelajaran ANN pada subbab sebelumnya yaitu xtraining, xtest, ytraining, dan ytest. Hal ini menjamin kesamaan data latih yang digunakan antara dua metode. Langkah awal yang dilakukan adalah melakukan import library yang digunakan yaitu Scikit-Learn untuk proses pembelajaran KNN, setelah import library sudah dilakukan maka langkah selanjutnya adalah dengan membaca dataset 4 buah file csv. Setelah dataset berhasil terbaca oleh google colab maka dataset akan siap untuk dibuat model KNN. Langkah selanjutnya adalah menentukan nilai X dan Y, untuk nilai X adalah 13 fitur data masukan dari sensor sedangkan nilai Y adalah 1 kolom output dengan memiliki 36 class klasifikasi yaitu angka 1 – 10 adalah mengklasifikasikan angka 1 sampai 10 dan angka 11 – 36 adalah mengklasifikasikan huruf abjad A sampai Z. kemudian dilakukan split data, pada penelitian ini untuk metode KNN memiliki data uji 30% dan data latih 70%. Kemudian setelah split data dilakukan scaler transform menggunakan MinMaxScaler. Setelah penentuan scaler maka selanjutnya adalah penentuan nilai k pada KNN. Pada penelitian ini diputuskan untuk memilih k dengan jumlah 3 karena akurasi yang baik yaitu 99% dan jumlah data yang banyak memerlukan nilai k yang tinggi juga. Untuk melihat perbandingan dengan nilai k yang lain dapat dilihat pada Gambar 7.

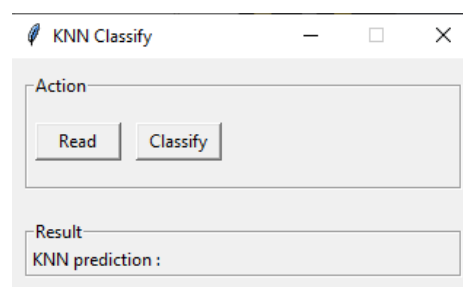


Gambar 7. Akurasi 99% pada k=3

Setelah k ditentukan dengan baik maka selanjutnya adalah untuk menyimpan model KNN dalam bentuk .pkl dan siap digunakan pada real time.

Antarmuka Tampilan Aplikasi

Antarmuka tampilan aplikasi realtime pada penelitian ini masih memakai code atau aplikasi pada peneliti terdahulu (Santoso, 2022). Aplikasi tersebut memiliki tampilan seperti pada Gambar 8



Gambar 8. Tampilan antarmuka aplikasi

Pada Gambar 8 dapat dilihat aplikasi memiliki 2 tombol yaitu “READ” dan “CLASSIFY”. Tombol “READ” berfungsi sebagai membaca data yang ada pada sensor sedangkan tombol “CLASSIFY” berfungsi sebagai mengambil 100 data terakhir yang telah dibaca pada tombol “READ” setelah itu dilakukan load model KNN dan melakukan prediksi dan menampilkan hasilnya pada kolom “KNN prediction: “. Akan tetapi pada saat pertama kali dijalankan aplikasi memiliki delay yang cukup ringan untuk memprediksi.

Hasil Pengujian Real Time

Pengujian secara real time dengan menggunakan metode KNN dilakukan secara langsung kepada 6 subjek. Masing – masing subjek melakukan pengujian 1 kali terhadap 36 gestur yaitu 10 gestur untuk gestur angka 1 sampai 10 dan 26 gestur untuk huruf abjad A sampai Z. subjek melakukan gestur sesuai dengan gestur tangan bahasa isyarat sibi, akan tetapi sebelum menuju ke gestur selanjutnya harus dilakukan reset. Reset disini dimaksudkan adalah dengan menggerakkan tangan menggenggam dan membuka telapak tangan untuk melakukan cek apakah tampilan pembacaan sensor sudah sesuai dengan gerakan tangan subjek. Karena setelah melakukan gestur terdapat delay sehingga diperlukannya reset. Untuk hasil pengujian realtime pada subjek dapat dilihat pada Tabel 1.

Tabel 1. Hasil pengujian tiap subjek

Subyek	Klasifikasi		Akurasi
	Benar	Salah	
1	31	5	86,11
2	29	7	80,55
3	26	10	72,22
4	28	8	77,77
5	27	9	75,00
6	24	12	66,66
Average			75,00

Dapat dilihat pada Tabel 1 bahwa hasil pada tiap subjek yang tertinggi adalah 86% sedangkan yang terendah adalah 66%, dengan akurasi rata – rata 75%

Diskusi Hasil

Berdasarkan hasil pengujian secara real time dengan menggunakan metode KNN dengan k bernilai 3 yang telah dilakukan mendapatkan hasil yang cukup baik bagi setiap subjeknya dengan rata rata akurasi 75% kepada 6 subjek. Untuk nilai tertinggi pada subjek yaitu 86% dengan jumlah benar 31 dan salah 5 sedangkan terendah yaitu 66% dengan jumlah benar 24 dan salah 12. Untuk hasil pada setiap gestur menunjukkan beberapa gestur memiliki tingkat akurasi yang tinggi yaitu 100% dan paling rendah adalah 16% dengan jumlah benar 1 dan salah 5. Namun pada beberapa gestur huruf memiliki akurasi yang rendah dibawah 50% yaitu huruf A dengan akurasi 33%, huruf R dengan akurasi 16%, huruf X dengan akurasi 33%. Hal tersebut dapat terjadi karena adanya kemiripan gestur

huruf A dengan huruf M, huruf N, huruf S, dan E. kemudian untuk gestur huruf R memiliki kemiripan dengan gestur angka 2, huruf K, huruf U, huruf V. kemudian untuk gestur huruf X memiliki kemiripan dengan gestur huruf T, huruf D, angka 1 dan huruf Z.

Hal ini disebabkan dengan adanya perbedaan panjang dan dimensi dari masing – masing tangan pada subjek. Selain itu juga dipengaruhi oleh kemampuan subjek untuk melakukan gerakan gestur yang sesuai dengan baik dan benar. Dari beberapa subjek yang diuji, ada beberapa diantaranya tidak dapat dengan maksimal untuk menekukkan jari sesuai dengan gestur tangan yang tersedia. Hal ini tentu menimbulkan kurang maksimalnya akurasi yang didapatkan pada beberapa gestur. Selain itu terdapat masalah pada penempatan sensor tekuk yang tidak pas pada tangan subjek karena setiap subjek memiliki letak tekukan metacarpo phalangeal dan proximal phalangeal yang berbeda. Namun untuk meminimalisir keadaan tersebut pada penelitian ini sensor diberikan selotip pada ujung atas sensor dengan sarung tangan agar tidak lepas atau tertarik pada saat melakukan gerakan gestur.

Berdasarkan pembuatan Model ANN maka pada penelitian ini untuk model ANN masih belum dapat dilakukan secara real time. Hasil yang didapat pada model ANN sudah cukup baik yaitu memiliki jumlah akurasi 95 % dan loss 0,1669.

Berdasarkan hasil pengujian menggunakan data uji pada model KNN dan ANN. Didapatkan akurasi pada confusion matrix yaitu 95% untuk ANN dan 99% untuk KNN. Hasil akurasi yang rendah disebabkan oleh jumlah loss yang tinggi sehingga Ketika data diuji dengan data diluar training maka hasilnya rendah.

Hasil Pengujian Model KNN dan ANN

Model KNN dan ANN yang sudah dibuat kemudian diuji menggunakan data 288 file yang disatukan kemudian data uji diambil dari hasil split data menggunakan Scikit-Learn yang kemudian disimpan dan digunakan untuk kedua metode yaitu KNN dan ANN. Kemudian untuk Classification report ANN menunjukkan hasil total akurasi 95% dengan menampilkan nilai precision, recall, dan f1-score pada masing masing gesturnya, begitu juga dengan classification report untuk KNN yang menunjukkan hasil total akurasi 99% lebih baik daripada ANN.

KESIMPULAN

Bagian Berdasarkan hasil dari pembuatan model pada kedua metode dapat dibandingkan sebagai berikut:

1. ANN memiliki waktu yang cukup lama pada saat training data dan load model dibandingkan dengan KNN
2. Berdasarkan pengujian menggunakan data uji, KNN memiliki akurasi yang lebih baik dibandingkan dengan ANN dengan nilai akurasi KNN adalah 99% dan nilai akurasi ANN adalah 95%
3. ANN memiliki nilai precision, recall dan f1-score yang lebih rendah dibandingkan dengan KNN.

Bedasarkan hasil pengujian maka pada penelitian ini metode KNN lebih baik daripada metode ANN dalam klasifikasi angka dan alfabet untuk bahasa SIBI

UCAPAN TERIMA KASIH

Mengucapkan terima kasih atas dukungan dan bantuan yang diberikan untuk penelitian ini disampaikan kepada:

1. Tuhan Yang Maha Esa atas berkat dan rahmat yang diberikan hingga terselesainya penelitian ini
2. Orang Tua serta keluarga yang senantiasa memberi dukungan dalam segala hal yang tidak dapat tergantikan
3. Bapak Dr. Eng. Romy Budhi, ST, MT. dan Bapak Mochamad Subianto, S.Kom., M.Cs. Dosen prodi Teknik Informatika selaku pembimbing Penelitian. Bapak Hendry Setiawan, ST, M.Kom Dosen prodi Teknik Informatika selaku penguji Penelitian.
4. Ibu Dr.Kestrialia Rega Prilianti, M.Si, Bapak Windra Swastika, Ph.D, Bapak Ir. Oesman Hendra Kelana, M.Div, M.Cs, Bapak Paulus Lucky Tirma Irawan, S.Kom., MT. Ronald Dwi Nompunu, M.T. yang telah menyalurkan ilmunya selama perkuliahan.

DAFTAR PUSTAKA

- Kusumah, A. P. J., Rizal, A. & Rahmawati, D., 2021. Bend Sensor untuk penerjemah bahasa isyarat tangan. *e-Proceeding of Engineering*, 8(2), p. 930.
- Santoso, D. A., 2022. Rancang Bangun Aplikasi Klasifikasi Gestur Tangan Menggunakan Metode K - Nearest Neighbors.
- Saputra, M. Z., Basjaruddin, N. C. & E. S., 2019. Pengembangan Sarung Tangan Elektronik. *POLBAN*.
- Widodo, R. B., Swastika, W. & Haryasena, A. B., 2020. Studi Sensor dan Akuisisi Data Hand Gesture dengan Sarung Tangan. *Prosiding Seminar Nasional CIASTECH 2021 Universitas Widyagama Malang*.
- Widodo, R. B., Swastika, W., Setiawan, H. & Subianto, M., 2021. Studi Pemrosesan Data Pengenalan Gestur Tangan menggunakan metode KNN. *Prosiding Seminar Nasional CIASTECH 2021 Universitas Widyagama Malang*.
- Zakaria, Firmansyah, R. A. & Prabowo, Y. A., 2019. Rancang Bangun Flex Sensor Gloves untuk Penerjemah Bahasa. *Prosiding Seminar Nasional Sains dan Teknologi Terapan Institut Teknologi Adhi Tama Surabaya*.



© 2022 by authors. Content on this article is licensed under a Creative Commons Attribution 4.0 International license. (<http://creativecommons.org/licenses/by/4.0/>).