

STUDI PEMROSESAN DATA PENGENALAN GESTUR TANGAN MENGUNAKAN METODE KNN

Romy Budhi Widodo^{1*)}, Windra Swastika²⁾, Hendry Setiawan²⁾, Mochamad Subianto³⁾

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

²Pusat Studi Artificial Intelligence for Digital Image dan Technopreneurship-Teknik Informatika,
Universitas Ma Chung, Malang

³Program Studi Teknik Informatika, Universitas Ma Chung, Malang

*Email Korespondensi: romy.budhi@machung.ac.id

ABSTRAK

Penelitian ini diawali oleh kebutuhan komunikasi antara penyandang tunarungu dan tunawicara dengan non-penyandang. Tujuan penelitian jangka panjang adalah penerjemah bahasa isyarat tangan menjadi teks. Sebuah sarung tangan dilengkapi sensor tekuk digunakan untuk akuisisi data. Data direkam dan akan digunakan pada tahapan desain model machine learning. Ada sepuluh langkah desain model machine learning yang dikerjakan dalam penelitian ini dan dijelaskan dalam artikel, yaitu import library, import dataset, exploratory data analysis, split data, data scrubbing, algoritma pre-model, penentuan algoritma machine learning, prediksi atau klasifikasi, optimasi, dan evaluasi. Dari enam belas fitur berhasil disederhanakan menjadi sebelas fitur dalam tahap pre-model algorithm. Pemisahan dataset menjadi tiga bagian yaitu data latih, data validasi, dan data uji; dengan proporsi 60:20:20. Pemilihan algoritma yang tepat adalah *k-nearest neighbor* dengan nilai $k = 3$. Akurasi yang dihasilkan dengan data uji adalah 98.9%.

Kata kunci: Pra pemrosesan data, exploratory data analysis, data scrubbing

ABSTRACT

This research was initiated by the need for communication between the deaf and speech impaired person with non-disabilities. The final research goal is hand sign language translators into text. A glove equipped with bending sensors, accelerometer, and magnetometer is used for data acquisition. The data is recorded and used at the design step of the machine learning model. Ten steps of machine learning model design is required, this study came to the sixth step. There are ten steps to the design of the machine learning model that were done in this study and described in the article, namely import library, import dataset, exploratory data analysis, split data, data scrubbing, pre-model algorithm, determination of machine learning algorithms, prediction or classification, optimization, and evaluation. In the pre-model algorithm step, the sixteen features was simplified to eleven features. In the split data step, separation of datasets into three parts namely training data, validation data, and test data with proportions of 60:20:20 accordingly. The correct algorithm selection is k-nearest neighbor with a value of $k = 3$. The accuracy of the systems using test data is 98.9%.

Keywords: Data pre-processing, exploratory data analysis, data scrubbing

PENDAHULUAN

Berdasarkan sensus 2012, terdapat 36.956 jiwa atau 7,87% dari total disabilitas di Indonesia adalah prevalensi tunarungu dan tunawicara [1]. Dari wawancara dengan Subjek tunarungu diperoleh keinginan suatu alat yang dapat membantu komunikasi antara penyandang tunarungu dengan masyarakat non-penyandang. Bahasa isyarat di Indonesia ada dua jenis seperti dijumpai dalam kehidupan sehari-hari, yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI memiliki reliabilitas tinggi dan standar pemerintah, SIBI diturunkan dari *American Sign Language* (ASL). Tetapi

BISINDO lebih fleksibel mengikuti suatu komunitas dan himpunan masyarakat di wilayah Indonesia [2]. Pada penelitian ini digunakan SIBI.

Penelitian terdahulu menggunakan sarung tangan untuk bahasa isyarat dilakukan menggunakan jaringan syaraf tiruan untuk klasifikasi angka 1, 2, dan 3 dilakukan Widodo dkk [3]. Hasil klasifikasi menunjukkan akurasi di atas 90%. Sedangkan penggunaan sensor flex yang merupakan sensor tekuk dipandang sebagai opsi yang tepat sebab tekukan tangan besarnya bisa terukur pada kelima jari. Demikian pula kemiringan tangan dapat diukur dengan menambahkan sensor inertial pada tangan. Penggunaan sensor flex, sensor inertial, dan sensor tekanan pada tangan telah dilakukan pula di dalam penelitian Lee dkk [4]. Lee dkk mengemukakan kelebihan penggunaan sarung tangan dibandingkan kamera dalam rekognisi gesture tangan.

Penelitian ini bertujuan mendasari klasifikasi secara real time, dimana tahapannya sampai pada persiapan model machine learning (ML). Klasifikasi yang diharapkan adalah angka 1 sampai 20, seperti pada Gambar 1. Gambar 1 diambil dari [5], namun gerakannya bisa dilihat pada sebuah tutorial Youtube pada menit ke-1 detik ke-45 hingga menit ke-3 detik ke-10 dengan link <https://www.youtube.com/watch?v=h681dhezQyw&t=11s>.

METODE PENELITIAN

Langkah-langkah dalam perancangan machine learning mengacu pada saran dari Theobald dkk [6]. Disarikan terdapat sepuluh tahapan, yaitu: (1) import library, (2) import dataset, (3) exploratory data analysis, (4) data scrubbing, (5) algoritma pre-model, (6) split data, (7) menentukan algoritma machine learning, (8) prediksi, (9) evaluasi, dan (10) optimasi.

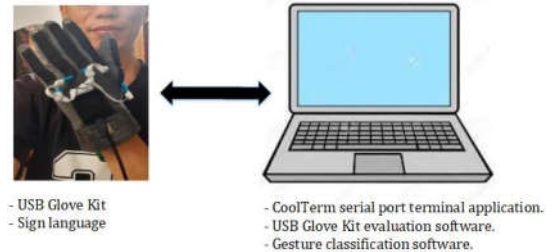
Tahap persiapan hardware dan penjelasan sensor telah dipublikasikan pada [3]. Klasifikasi yang diharapkan adalah angka 1-20. Jika diamati dari bahasa isyarat pada Gambar 1, isyarat angka belasan (11-20) membutuhkan ayunan punggung tangan ke bawah. Jika dibandingkan isyarat angka 1-9 yang statis dan tidak membutuhkan ayunan tangan. Maka di penelitian ini digunakan data sensor accelerometer selain sensor tekuk. Untuk klasifikasi bahasa isyarat angka, kelima sensor yang terpasang memiliki sepuluh fitur dominan dan enam fitur tambahan dari sensor accelerometer dan magnetometer [7]; sehingga total ada 16 fitur, 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
- x₁₄: Magnetometer sumbu x
- x₁₅: Magnetometer sumbu y
- x₁₆: Magnetometer sumbu z

Perangkat keras yang digunakan untuk pengambilan data sama dengan penelitian sebelumnya pada [3], seperti diilustrasikan pada Gambar 2.



Gambar 1. Angka 1-20 dalam Sistem Isyarat Bahasa Indonesia [5]



Gambar 2. Diagram pengambilan data gestur tangan [3]

Perangkat lunak yang digunakan adalah Python dengan notebook Jupyter. Instalasi Python menggunakan pre-packaged Python Distribution Anaconda. Adapun library Python yang digunakan adalah Pandas, NumPy, Sklearn (preprocessing, feature_selection, decomposition, model_selection)

HASIL DAN PEMBAHASAN

Pada bagian ini dilaporkan hasil beberapa tahapan dalam desain machine learning. Dataset terdiri atas 16 kolom (x_1 hingga x_{16}).

Tahap 1: *Import library/package*

Salah satu kelebihan Python untuk machine learning karena dilengkapi package yang lengkap. Gambar 3 menunjukkan potongan program import library.

```
import pandas as pd
import sklearn.preprocessing as skl
import numpy as np
import seaborn as seab
```

Gambar 3. *Import library*

Library Pandas banyak digunakan mengelola dan presentasi data termasuk tabulasi data. Dataset akan dipanggil dalam bentuk dataframe Pandas. Library NumPy digunakan untuk mengolah multi-dimensional array dan matrik, menghitung beberapa fungsi matematika. Library Scikit-learn (Sklearn) merupakan library inti pada machine learning. Sklearn tergolong library untuk *shallow algorithm* (model diprediksi langsung dari fitur input, berbeda dengan deep learning dimana output tergantung dari hasil layer sebelumnya). Didalam Sklearn memuat logistic regression, decision trees, linear regression, gradient boosting, dan lain-lain. Sklearn juga digunakan untuk evaluasi seperti menghitung mean absolute error, serta metode partisi data, yaitu split dan cross validation. Fungsi penting Sklearn dapat melakukan proses pelatihan (model train) dan menggunakan model hasil pelatihan untuk prediksi data tes. Masih ada library lain yang mendukung visualisasi data misalnya Matplotlib dan Seaborn. Keduanya menghasilkan visualisasi data yang berkualitas tinggi. Sedangkan library TensorFlow cocok untuk *deep learning* dan *artificial neural network*.

Tahap 2: Import dataset

Folder kerja yang digunakan secara default pada C:/users/./; pada folder kerja disiapkan .csv file. File dengan format csv (comma-separated values) memiliki format data dalam teks dan separatornya koma, lain dengan file excel dimana format dalam tabular. Bentuk file csv pada data gestur tangan seperti pada Gambar 4. Terdapat 17 fitur dalam satu baris, fitur ke-1 hingga ke-16 berasal dari 16 sensor, sedangkan data ke-17 adalah kelas yang diberikan oleh peneliti. Kelas 1 artinya bahasa isyarat 1, kelas 20 berarti gestur tangan untuk bahasa isyarat 20.

```
11001,7893,2554,11935,1322,12992,21177,7970,6217,1785,13040,-  
480,6560,160,2176,388,1  
11073,8064,2554,11721,1028,12695,20553,8038,6105,1785,13040,-  
480,6560,160,2176,388,1  
:  
:  
8446,3723,2554,7801,280,9528,13428,3851,1871,3993,12752,5120,6144,-  
404,2416,296,20
```

Gambar 4. Cuplikan data csv gestur tangan

Library Pandas memungkinkan import csv file dan melakukan manipulasi atau pengolahan data csv tanpa merubah file asli, untuk itu Pandas merubahnya menjadi dataframe sehingga dapat dikelola dalam *development environment*. Cuplikan pada Gambar 5 adalah tahapan import dataset, set presisi 2 digit, menampilkan dimensi dataframe, dan menampilkan contoh dua baris dari dataframe.

```
#library pada Gambar 3 di sini ....  
path = r"~/data_Python_study/gabunganAngka-train-label.csv"  
  
#membuat header kolom sebab data tidak memiliki header  
headernames =  
['mcpPinky','pipPinky','mcpRing','pipRing','mcpMid','pipMid','mcpIndex','pipIndex',  
 'x',  
 'mcpThumb','pipThumb','accX','accY','accZ','magX','magY','magZ','output']  
  
dataframe = pd.read_csv(path, names=headernames)  
pd.set_option('precision',2)  
  
print(dataframe.shape) #menampilkan dimensi data  
print(dataframe.head(n=2)) #menampilkan 2 baris data pertama
```

Gambar 5. Cuplikan import dataset, set presisi, dimensi data, dan tampilan isi dataframe

Sedangkan Gambar 6 menunjukkan hasil eksekusi Gambar 5.

```
(45147, 17)  
mcpPinky pipPinky mcpRing pipRing mcpMid pipMid mcpIndex pipIndex \  
0 11001 7893 2554 11935 1322 12992 21177 7970  
1 11073 8064 2554 11721 1028 12695 20553 8038  
  
mcpThumb pipThumb accX accY accZ magX magY magZ output  
0 6217 1785 13040 -480 6560 160 2176 388 1  
1 6105 1785 13040 -480 6560 160 2176 388 1
```

Gambar 6. Hasil import dataset berupa dimensi data dan tampilan isi dataframe

Tahap 3: Exploratory data analysis (EDA)

Tahap EDA bertujuan mempersiapkan dataset untuk pemrosesan dan analisis lebih lanjut. Didalamnya terdapat pemahaman bentuk dan distribusi data dan scan *missing value*, analisis korelasi untuk melihat fitur mana yang relevan. Untuk mengamati jumlah data yang kosong (null) digunakan perintah `.isnull().sum()`. Gambar 7 menunjukkan hasil eksekusi pengecekan *missing value*, diperoleh tidak ada data yang hilang pada semua fitur.

Jika pada Gambar 7, dijumpai ada data yang hilang maka beberapa teknik bisa digunakan untuk proses selanjutnya pada Tahap 4 (*data scrubbing*), yaitu:

- ```
#library pada Gambar 3 di sini
#list program import dataset pada Gambar 5 di sini....

dataframe.isNull().sum()

#hasil eksekusi pada dataframe
mcpPinky 0
pipPinky 0
mcpRing 0
pipRing 0
mcpMid 0
pipMid 0
mcpIndex 0
pipIndex 0
mcpThumb 0
pipThumb 0
accX 0
accY 0
accZ 0
magX 0
magY 0
magZ 0
output 0
dtype: int64
```

```
#library pada Gambar 3 di sini
#list program import dataset pada Gambar 5 di sini....

#jika menggunakan Pearson's correlation
df_corr = dataframe.corr(method='pearson')

#membuat heatmap dan print hasil korelasi
seab.heatmap(df_corr, annot=True, cmap='coolwarm')
print(df_corr)

#hasil eksekusi heatmap (tampilan terlalu kecil sebab ada 16 fitur)
```

281

#### Tahap 4: Split data

Klasifikasi tergolong *supervised machine learning* dimana input data set dan target diberikan bersama-sama. Dalam klasifikasi, kita perlu menyediakan minimal dua set data, yaitu: data latih dan data uji. Data latih digunakan untuk membangun model prediksi. Sedangkan data uji digunakan untuk menentukan tingkat akurasi model yang sudah dihasilkan. Supaya model valid, maka kuncinya adalah data uji tidak boleh dipakai untuk proses latih, membangun model, dan mengoptimasi model.

Selain kedua jenis data tersebut, beberapa peneliti memiliki sekelompok data yang disebut data validasi. Namun umumnya keberadaan data validasi bukanlah suatu keharusan dalam proses *machine learning*. Urutannya menjadi sebagai berikut: 1) Model awal dibangun dengan data latih; 2) Data validasi dicobakan pada model, digunakan untuk memberi masukan dan mengoptimasi parameter model; 3) Data uji digunakan untuk menghitung error model dalam memprediksi luaran.

Dalam implementasi klasifikasi, langkah pertama adalah melatih classifier. Sehingga diperlukan data latih. Porsi ketiga jenis data dalam penelitian ini adalah 60% untuk data latih, 20% data validasi, dan 20% data uji. Teknik split data, mula-mula data dibagi 60%:40%, 60% data akan menjadi data latih. Data yang 40% akan dibagi menjadi 20%:20% untuk data validasi dan data uji. Gambar 9a adalah proses split data menggunakan class **sklearn.model\_selection.train\_test\_split()**.

```
#library pada Gambar 3 di sini
import sklearn.model_selection as sklms
#list program import dataset pada Gambar 5 di sini....

#memisahkan tahap I, 60%-40% untuk training dan testing data I

X_train, X_test1, y_train, y_test1 = sklms.train_test_split(X, y, test_size = 0.4,
random_state = 8, shuffle = True) #menggunakan seed number 8

print("ukuran X train: ", X_train.shape)
print("ukuran y train: ", y_train.shape)

print("ukuran X test sementara/tahap 1: ", X_test1.shape)
print("ukuran y test sementara/tahap 1: ", y_test1.shape)

#menyimpan ke excel file
df_test_excel=X_test1.to_excel("df_test_excel.xlsx") #menyimpan X_test1 ke excel

#-----
#memisahkan tahap II, 50%-50% untuk validation dan testing data

X_test, X_validation, y_test, y_validation = sklms.train_test_split(X_test1, y_test1,
test_size = 0.5, random_state = 8, shuffle = True) #menggunakan seed number 8

print("ukuran X validation: ", X_validation.shape)
print("ukuran y validation: ", y_validation.shape)

print("ukuran X test: ", X_test.shape)
print("ukuran y test: ", y_test.shape)
#-----
```

Gambar 9a. Proses split data menjadi data latih, data validasi, dan data uji.

```
ukuran X train: (27088, 16)
ukuran y train: (27088, 1)

ukuran X test sementara/tahap 1: (18059, 16)
ukuran y test sementara/tahap 1: (18059, 1)

ukuran X validation: (9030, 16)
ukuran y validation: (9030, 1)

ukuran X test: (9029, 16)
ukuran y test: (9029, 1)
```

Gambar 9b. Hasil eksekusi program Gambar 9a.

#### Tahap 5: Data scrubbing

Data scrubbing adalah istilah umum yang mewakili “manipulasi data untuk keperluan analisis, termasuk mempersiapkan jenis data.” Beberapa algoritma memerlukan jenis data yang tepat, misalnya regresi linier menggunakan variabel kontinu, Gradient Descent dan k-Nearest Neighbor membutuhkan data yang sudah di-*rescaling*.

Beberapa teknik data scrubbing seperti pernah diuraikan sedikit di Tahap 3, yaitu 1) penghapusan data, 2) penanganan *missing value*. Teknik lain adalah 3) *one-hot encoding*, digunakan untuk merubah variabel kategorikal menjadi bentuk biner, jika data kategorikal tersebut tidak dapat dipakai untuk algoritma clustering dan regresi, misalnya pengubahan gender dan nama kota menjadi biner.

Jenis data scrubbing yang lain adalah: 4) Scaling, 5) Normalisasi, 6) Binerisasi, 7) Standardisasi, 8) Data labelling. **Scaling**, secara umum digunakan untuk rescaling data

menjadi 0 sampai 1. Gradient Descent dan k-Nearest Neighbor membutuhkan scaling. Gambar 10, menunjukkan proses rescaling dengan MinMaxScaler pada data gestur tangan.

**Normalisasi**, tahap ini bermanfaat jika data tergolong Sparse dataset, artinya data banyak yang bernilai nol. Tujuan normalisasi data: menormalisasi masing-masing data ke unit norm, untuk rescaling tiap baris data supaya panjangnya = 1. Normalisasi banyak digunakan untuk klasifikasi dan clustering teks. Normalisasi menggunakan library scikit-learn preprocessing yang dipakai adalah "Normalizer" class. Ada dua teknik normalisasi: L1 normalization dan L2 normalization

```
library pada Gambar 3 di sini
list program import dataset pada Gambar 5 di sini....
list program split data pada Gambar 9 di sini....

membuat penskala 0 sampai 1
data_scaler = skl.MinMaxScaler(feature_range=(0,1))
np.set_printoptions(precision=1)

print("X train sebelum scaling:")
print(X_train[0:3])

X_train = data_scaler.fit_transform(X_train)
X_validation = data_scaler.fit_transform(X_validation)
X_test = data_scaler.fit_transform(X_test)

print("X train setelah scaling:")
print(X_train[0:3])

hasil eksekusi proses rescaling (tampilan 3 baris pertama saja)
```

```
X train sebelum scaling:
mcpPinky pipPinky mcpRing pipRing mcpMid pipMid mcpIndex \
30994 2650 360 178 983 93 203 558
14626 9660 3286 3747 16261 1601 14458 20250
32947 12115 7674 950 11331 280 7229 12777

pipIndex mcpThumb pipThumb accX accY accZ magX magY magZ
30994 308 2108 191 8976 4880 9888 -736 2304 392
14626 11409 4592 6152 13840 128 6256 40 2328 280
32947 9023 3880 1800 11568 4224 6176 -456 2336 396
```

```
X train setelah scaling:
[[0.1 0. 0. 0. 0. 0. 0. 0. 0.1 0. 0.7 0.4 0.6 0.4 0.8 0.8]
 [0.3 0.1 0.1 0.5 0. 0.4 0.6 0.3 0.1 0.2 0.7 0.3 0.5 0.8 0.8 0.7]
 [0.4 0.2 0. 0.3 0. 0.2 0.4 0.3 0.1 0.1 0.7 0.4 0.5 0.6 0.8 0.8]]
```

Gambar 10. Proses rescaling pada data gestur tangan

**Binerisasi/Thresholding**, konsep binerisasi data adalah menjadikan data 0 jika dibawah nilai threshold. Data menjadi 1 jika nilai data sama dengan atau diatas threshold. Penggunaan preprocessing ini lebih kepada tujuannya untuk mendeteksi ada atau tidak suatu fitur, bukan kepada nilainya yang dipentingkan. Binerisasi menggunakan library scikit-learn preprocessing, yang dipakai adalah "Binarizer" class.

**Standardisasi**, kegunaan standardization adalah untuk mentransformasi atribut data ke distribusi normal (normally distributed Gaussian). Dimana mean=0 dan SD=1. Preprocessing ini bermanfaat untuk algoritma Machine Learning seperti Regresi Linier dan Regresi Logistik; dimana data diasumsikan memiliki distribusi normal. Sehingga bila data sudah di-rescale dengan teknik ini, maka harapannya metode ML tersebut menghasilkan performansi yang tinggi. Untuk standarisasi ke mean=0 dan SD=1, digunakan library scikit-learn preprocessing, yang dipakai adalah "StandardScaler" class. Perintah yang digunakan adalah

```
data_scaler = skl.StandardScaler()
X_train = data_scaler.fit_transform(X_train)
```

**Data Labelling**, data yang dikirim ke ML harus sesuai dengan tipe data yang diharapkan oleh metode ML itu sendiri dan class yang digunakan. Sebagian besar fungsi pada scikit learn membutuhkan label angka. Jika data masih memiliki label dalam words (kata/string) maka perlu proses "label encoding" untuk merubah menjadi label angka. Untuk data labeling digunakan library scikit-learn preprocessing, yang dipakai adalah "LabelEncoder" class.

## Tahap 6: Algoritma pre-model

Kadang tidak semua fitur digunakan. Fitur yang paling relevan untuk prediksi akan dipilih dalam proses ini. Proses memilih fitur yang akan dipakai untuk ML dinamakan Feature Selection atau Attribute Selection atau algoritma pre-model. Kelebihan penggunaan feature selection sebelum pemodelan data dengan ML adalah 1) mengurangi overfitting, 2) meningkatkan akurasi ML utamanya yang menggunakan regresi linier dan logistik, 3) mengurangi waktu pelatihan data.

Beberapa teknik feature selection: 1) menggunakan automatic feature: univariate selection, 2) menggunakan recursive feature elimination, 3) menggunakan feature importance, 4) menggunakan unsupervised learning, yaitu PCA (Principal Component Analysis) dan k-Means Clustering.

Pada kesempatan ini digunakan *feature importance*, teknik *feature importance* digunakan untuk memilih fitur yang penting. Pemilihannya berdasarkan skor, semakin tinggi skor maka atribut tersebut tergolong penting. Ekstrak fitur menggunakan class **sklearn.ensemble.ExtraTreesClassifier**. Gambar 11 menunjukkan prosesnya.

Dari hasil *feature importance* diperoleh 10 fitur penting yaitu: fitur 2 (PIP kelingking), 10 (PIP ibu jari), 14 (Magnetometer x), 3 (MCP jari manis), 4 (PIP jari manis), 6 (PIP jari tengah), 5 (MCP jari tengah), 8 (PIP jari telunjuk), 9 (MCP ibu jari), 16 (Magnetometer z).

Terlihat bahwa sensor pada kelima jari merupakan fitur penting yang tidak dapat disederhanakan. Berikutnya adalah data dari magnetometer x dan z yang merupakan sebagian dari 10 fitur penting. Dalam praktik data magnetometer akan mudah berubah sesuai arah Subjek saat percobaan dilakukan, sehingga fitur magnetometer dalam praktik akan dihilangkan dan digantikan fitur accelerometer. Fitur accelerometer ( $x_{11}$ ,  $x_{12}$ , dan  $x_{13}$ ) digunakan sebab angka 10 hingga 20 menggunakan anggukan tangan yang dapat dideteksi oleh sensor tersebut.

```
library pada Gambar 3 di sini
import sklearn.ensemble as skle
list program import dataset pada Gambar 5 di sini....

array = dataframe.values #two dimension tabular data
#memisahkan INPUT dan OUTPUT
X = array[:, 0:16] #mengambil indeks ke-0 sampai 15
Y = array[:, 16] #mengambil indeks ke-16
print("n X size: 'n', X.shape)
print("n Y size: 'n', Y.shape)
#-----

Extract feature dari dataset
model = skle.ExtraTreesClassifier()
model.fit(X,Y)
print("nScores for each attributes:'n", model.feature_importances_)

#hasil eksekusi

X size:
(45147, 16)

Y size:
(45147,)

Scores for each attributes:
[0.05971364 0.10022071 0.07325942 0.07230683 0.06274054 0.07278559
 0.0554361 0.06361204 0.06370255 0.08837127 0.02399304 0.04091566
 0.02997139 0.07177275 0.06101359 0.06018487]
```

Gambar 11. Proses pemilihan fitur dengan *feature importance*

Berdasarkan analisis di atas, fitur yang akan digunakan dalam klasifikasi adalah fitur 2 (PIP kelingking), 10 (PIP ibu jari), 3 (MCP jari manis), 4 (PIP jari manis), 6 (PIP jari tengah), 5 (MCP jari tengah), 8 (PIP jari telunjuk), 9 (MCP ibu jari), 11 (Accelerometer x), 12 (Accelerometer y), dan 13 (Accelerometer z).

## Tahap 7: Menentukan *algoritma Machine Learning*

Langkah ini meskipun dari awal sudah kita rencanakan, utamanya pada saat melakukan Tahap 4 (data scrubbing) di bagian scaling; perlu adanya pemeriksaan ulang. Adapun

beberapa algoritma machine learning menurut [6] yang sesuai digunakan dalam penelitian ini adalah *k-nearest neighbor* (KNN). Beberapa karakteristik KNN yang sesuai dengan sumber daya penelitian ini yaitu: 1) Output target: diskrit; 2) Sifat data: tidak ada nilai data yang hilang, dimensi data terbatas; 3) Metodologi: supervised; 4) *Computing resources*: sedang (*medium*); dan 5) Akurasi: sedang.

### Tahap 8: Prediksi Atau Klasifikasi

Algoritma KNN dapat digunakan untuk kasus prediksi (*regressor*) maupun klasifikasi. Pada penelitian ini kasusnya adalah klasifikasi dengan 20 kelas. Langkah awal adalah menentukan nilai *k* pada algoritma KNN. Tabel 1 menunjukkan hasil percobaan penggunaan beberapa kandidat nilai *k* (ganjil) yang sering digunakan dalam berbagai penelitian.

Tabel 1. Hasil pemilihan nilai *k* pada algoritma KNN

| <i>k</i> | <i>False Positive</i> | <i>False Negative</i> | Total | Akurasi |
|----------|-----------------------|-----------------------|-------|---------|
| 3        | 106                   | 106                   | 212   | 0.988   |
| 5        | 127                   | 127                   | 254   | 0.986   |
| 7        | 148                   | 148                   | 296   | 0.984   |

Langkah berikutnya akan menggunakan *k* = 3 dengan pertimbangan akurasi yang lebih tinggi dan jumlah *false* paling kecil diantara ketiga uji coba, yaitu bernilai 212. Gambar 12 memperlihatkan proses mendapatkan matrik konfusi untuk evaluasi nilai *k*, data yang digunakan adalah data validasi.

### Tahap 9: Optimasi

Langkah optimasi digunakan untuk *tuning* hiperparameter. Penggunaan data validasi pada tahap sebelumnya

### Tahap 10: Evaluasi

Langkah evaluasi model dilakukan dengan menerapkan model pada data uji (data tes). Kemudian menguji klasifikasi, misalnya 10 baris pertama dari data uji, dilihat hasil klasifikasinya. Gambar 13 adalah program yang sama dengan Gambar 12 namun perhitungan confusion matrix, classification report, dan accuracy score menggunakan *y\_test*. Bagian akhir dari program di Gambar 13 adalah cara menggunakan model yang telah dihasilkan untuk klasifikasi.

```

a)
Classification report:
precision recall f1-score support

 1 1.00 1.00 1.00 351
 2 0.99 1.00 0.99 343
 3 1.00 0.99 0.99 358
 4 1.00 0.99 0.99 404
 5 0.98 0.99 0.99 349
 6 1.00 0.99 1.00 478
 7 1.00 1.00 1.00 427
 8 0.98 0.99 0.99 390
 9 0.98 1.00 0.99 409
 10 1.00 0.99 0.99 402
 11 1.00 1.00 1.00 470
 12 1.00 1.00 1.00 449
 13 0.99 0.98 0.99 584
 14 1.00 0.99 1.00 581
 15 0.98 0.99 0.99 550
 16 0.97 0.96 0.97 523
 17 0.94 0.96 0.95 442
 18 0.99 0.99 0.99 493
 19 1.00 1.00 1.00 544
 20 1.00 1.00 1.00 482

 accuracy 0.99 9029
 macro avg 0.99 9029
weighted avg 0.99 9029

Accuracy:
0.9895891017831432

b)
sepuluh baris data uji:
[[0. 0.7 0. 0. 0. 0. 0. 0. 0. 0.1 0.8 0.2 0.4 0.7 0.4 0.9]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.1 0.1 0.8 0.4 0.4 0.6 0.9]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.5 0.3 0.7 0.2 0.6 0.7]
 [0.1 0.1 0. 0.1 0. 0. 0.2 0.1 0. 0.1 0.8 0.4 0.5 0.5 0.8]
 [0.7 0.3 0.1 0.4 0. 0.5 0.5 0.5 0.6 0. 0.8 0.2 0.4 0.9 0.5 0.8]
 [0.6 0.4 0.2 0.2 0. 0.8 0.8 0.4 0.3 0.4 0.8 0.2 0.4 0.8 0.4 0.9]
 [0.3 0.9 0.2 0.2 0. 0. 0. 0. 0. 0.7 0.3 0.5 0.7 0.7 0.8]
 [0.3 0.1 0.1 0.4 0.1 0.6 0.5 0.3 0.1 0.3 0.8 0.3 0.5 0.7 0.8 0.8]
 [0. 0.1 0. 0. 0. 0. 0. 0. 0. 0.7 0.3 0.4 0.5 0.6 0.9]
 [0.4 0.2 0.1 0.5 0. 0.2 0.3 0.4 0.2 0.1 0.7 0.4 0.5 0.4 0.8 0.7]]

Hasil klasifikasi sesuai dengan kelas sebenarnya, sebagai berikut:
[6 14 15 19 4 2 3 7 13 15]

```

Gambar 14. a) Tampilan hasil perhitungan confusion matrix dan akurasi terhadap seluruh data uji.  
b) Percobaan klasifikasi terhadap 10 data.

Hasil pengujian model KNN terhadap semua data uji memiliki akurasi 0,989 dengan false positive=94 dan false negative=94; dimana nilai ketiganya lebih baik dibandingkan hasil validasi di Tabel 1. Sedangkan hasil percobaan klasifikasi dengan 10 data masukan, diperoleh akurasi 100%. Gambar 14 merupakan hasil perhitungan confusion matrix dan klasifikasi 10 data masukan.

## KESIMPULAN

Penelitian ini mendasari tahap klasifikasi gestur tangan yang akan dilakukan pada penelitian lanjutan. Dari tahapan ini telah dihasilkan 11 fitur dari 16 fitur sebelumnya melalui proses *feature importance*, yang dapat menjadi pertimbangan untuk memperkecil *resource* komputasi. Data telah dibagi menjadi data latih, data validasi, dan data uji, dengan perbandingan 60%:20%:20%. Hasil akurasi menggunakan data uji adalah 98.9%. Tulisan ini mencoba menuliskan selengkapnya proses penelitian yang dilakukan, tentunya banyak kekurangan sehingga penulis bersedia berdiskusi lebih lanjut.

## UCAPAN TERIMA KASIH

Mengucapkan terima kasih kepada LPPM Universitas Ma Chung atas skema penelitian mandiri. Ucapan terima kasih diucapkan kepada semua pihak yang mendukung penelitian ini sehingga dapat berjalan dengan baik.

## REFERENSI

- [1] F.A. Prasetyo, "Disability and Health Issues: Evolution Concepts, Human Rights, Complexity of Problems, and Challenges (in Indonesian)," Jakarta, 2014. doi: 10.1007/s13398-014-0173-7.2.
- [2] N. Sugianto and F. Samopa, "Analisa Manfaat Dan Penerimaan Terhadap Implementasi Bahasa Isyarat Indonesia Pada Latar Belakang Komplek Menggunakan Kinect Dan Jaringan Syaraf Tiruan ( Studi Kasus SLB Karya Mulia 1 )," *Juisi*, vol. 01, no. 01, pp. 56–72, 2015.
- [3] R. B. Widodo, W. Swastika, and A. B. Haryasena, "Studi Sensor dan Akuisisi Data Hand Gesture dengan Sarung Tangan," in *Conference on Innovation and Application of Science and Technology (CIASTECH )*, 2020, no. Ciastech, pp. 561–568, [Online]. Available: <http://publishing-widyagama.ac.id/ejournal-v2/index.php/ciastech/article/view/1949>.
- [4] B. G. Lee and S. M. Lee, "Smart Wearable Hand Device for Sign Language Interpretation System with Sensors Fusion," *IEEE Sens. J.*, vol. 18, no. 3, 2018, doi: 10.1109/JSEN.2017.2779466.
- [5] Anonymous, *Kamus Sistem Isyarat Bahasa Indonesia & Bahasa Murni*. 2006.
- [6] O. Theobald, *Machine Learning with Python: A Practical Beginners' Guide*. Scatterplot Press, 2019.
- [7] F. S. Systems, "Bend Sensor ® USB Glove Kit User Guide," 2016.