

## PEMANFAATAN ALGORITMA *NEURAL NETWORK* UNTUK *PREDICTIVE ANALYTIC* ANGKA BUTA HURUF DI INDONESIA

Ike Kurniati<sup>1)</sup>, Heru Winarno<sup>2)</sup>, Dita Yuliyanti<sup>3)</sup>

<sup>1,2</sup>Prodi Teknik Informatika, Fakultas Teknologi, ITB Swadharma Jakarta

<sup>3</sup>Prodi Sistem Informasi, Fakultas Teknologi, ITB Swadharma Jakarta

Correspondence author: I. Kurniati, ikekurniati@swadharma.ac.id, Jakarta, Indonesia

### Abstract

The government of Indonesia has always been aware of the necessary role of education in the development and progress of the nation, so it included it in Article 31 of the 1945 Constitution. Based on data from the Central Statistics Agency, the population aged over ten years who were illiterate in 2017 amounted 4.08% of the total population aged over ten years. Predictive analytics is a method of analysis used to assess risk, and future business trends. The research is to find strategic points that can be utilized and used as references in implementing efforts to reduce illiteracy in Indonesia. The data for the research using secondary data from bps.go.id, namely data on the illiteracy rate for the 2011 – 2019 period. The predictions were made using the Artificial Neural Network Algorithm to produce an output accuracy of 69%.

**Keywords** : predictive analytics, illiteracy, artificial neural network

### Abstrak

Pemerintah sejak dulu menyadari akan peran pendidikan yang begitu penting bagi pembangunan dan kemajuan bangsa, sehingga mencantumkan dalam pasal 31 Undang-Undang Dasar 1945. Berdasarkan data Badan Pusat Statistik penduduk usia diatas 10 tahun yang buta huruf pada 2017 mencapai 4,08% dari total populasi penduduk usia di atas 10 tahun. *Predictive analytics* merupakan sebuah metode dalam analisis yang digunakan untuk menilai risiko, tren bisnis di masa depan, hingga prediksi kapan perawatan peralatan akan dilakukan. Tujuan dari penelitian ini untuk menemukan poin-poin strategis yang dapat dimanfaatkan dan digunakan sebagai referensi dalam rangka implementasi upaya-upaya menekan jumlah buta huruf di Indonesia. Penelitian dilakukan dengan data sekunder dari bps.go.id, yaitu data angka buta huruf dalam kurun waktu 2011 – 2019. Prediksi yang dilakukan menggunakan algoritma jaringan syaraf tiruan menghasilkan ouput akurasi sebesar 69 %.

**Kata Kunci** : jaringan syaraf tiruan, *predictive analytics*, buta huruf

### A. PENDAHULUAN

Pendidikan merupakan salah satu aspek penting yang sangat berperan dalam pembangunan suatu bangsa (Simbolon et al.,

2018). Pemerintah yang sejak dulu menyadari akan peran pendidikan yang begitu penting bagi pembangunan dan kemajuan bangsa, mencantumkan pasal 31 dalam Undang-Undang Dasar 1945 tentang

pendidikan yang berbunyi (1) Tiap-tiap warga negara berhak mendapat pengajaran; (2) Pemerintah mengusahakan dan menyelenggarakan satu sistem pengajaran nasional, yang diatur dengan undang-undang (Sekretariat Negara, 1945). Berdasarkan apa yang telah ditetapkan dalam Undang-Undang Dasar 1945 di atas, maka pendidikan merupakan hal yang sangat vital sehingga semua warga negara tanpa pandang bulu berhak mendapatkannya. Oleh karena itu, pendidikan perlu diperhatikan dan dibangun sedemikian rupa agar sumber daya manusia di negara ini menjadi lebih berkualitas.

Buta huruf dalam arti buta bahasa Indonesia (Sucahyo, Nur; Kurniati, Ike; Hidayatullah, 2020), buta pengetahuan dasar yang dapat menunjang kehidupan sehari-hari, buta aksara dan angka, buta akan informasi kemajuan teknologi, merupakan beban berat untuk mengembangkan sumberdaya manusia yang berkualitas dalam arti mampu menggali dan memanfaatkan peluang yang ada di lingkungannya. Selain itu buta huruf (buta aksara) adalah orang yang tidak memiliki kemampuan membaca, menulis dan berhitung serta penerapannya dalam kehidupan sehari-hari.

Berdasarkan data Badan Pusat Statistik penduduk usia di atas 10 tahun yang buta huruf pada 2017 mencapai 4,08% dari total populasi penduduk usia di atas 10 tahun. Angka ini lebih rendah dari tahun sebelumnya 4,19%. Sementara penduduk usia 15 tahun ke atas yang buta huruf 4,5% dan penduduk usia 15-45 tahun yang tidak bisa membaca dan menulis 0,94%. Adapun penduduk usia di atas 45 tahun yang buta huruf mencapai 11,08% (Simbolon et al., 2018).

Salah satu hal yang dapat dilakukan untuk memprediksi angka buta huruf di Indonesia adalah dengan cara menemukan signifikan faktor yang menyebabkan tingginya angka buta huruf (Andiyono et al., 2013). Agar setelah menemukan faktor yang paling mempengaruhi tersebut dapat dilakukan langkah-langkah selanjutnya

dalam upaya mengurangi angka buta huruf yang terjadi di Indonesia. Faktor signifikan untuk menentukan *predictive analytic* (Bukhsh & Stipanovic, 2020) angka buta huruf di Indonesia dengan menggunakan algoritma neural network.

*Predictive analytics* merupakan sebuah metode dalam analisis yang digunakan untuk menilai risiko, tren bisnis di masa depan, hingga prediksi kapan maintenance akan dilakukan (Sabbeh, 2018).

Algoritma *Neural Network* atau Jaringan syaraf tiruan (JST) adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel saraf biologi (Septiana et al., 2020). *Neural network* memiliki beberapa properti yang membuat mereka populer untuk *clustering*. Pertama, *neural network* adalah arsitektur pengolahan inheren paralel dan terdistribusi (Sucahyo, Nur; Kurniati, Ike; Hidayatullah, 2020). Kedua, *neural network* belajar dengan menyesuaikan bobot interkoneksi dengan data, Hal ini memungkinkan neural network untuk "menormalkan" pola dan bertindak sebagai fitur (atribut) *extractors* untuk kelompok yang berbeda. Ketiga, *neural network* memproses vektor numerik dan membutuhkan pola objek untuk diwakili oleh fitur kuantitatif saja (Sucahyo, Nur; Kurniati, Ike; Hidayatullah, 2020).

Pada penelitian ini dipelajari juga hasil dari penelitian terdahulu yaitu penelitian dilakukan oleh Arius Satoni Kurniawansyah dalam memprediksi hasil ujian kompetensi kebidanan Akbid Dehasen Bengkulu menggunakan metode Artificial Neural Network. Penelitian ini menentukan beberapa variabel yang menjadi kriteria kelulusan mahasiswa. Data yang diambil adalah data hasil Ujian Kompetensi Kebidanan Akbid Dehasen Bengkulu tahun 2014 dan 2015 (Kurniawan, 2018).

Penelitian yang dilakukan oleh Bakhtiar Rifai untuk prediksi penyakit jantung. Model yang dihasilkan diuji untuk mendapatkan nilai *accuracy*, *precision*, *recall* dan AUC dari algoritma sehingga didapat pengujian

dengan menggunakan *neural network* didapat nilai *accuracy* adalah 91.45 % dengan nilai *precision* 92.79 % dan nilai AUC adalah 0.937. Dengan demikian dari hasil pengujian model diatas dapat disimpulkan bawa *neural network* memberikan pemecahan untuk permasalahan penyakit jantung lebih akurat (Rifai, 2013).

Penelitian yang dilakukan oleh Nur Sucahyo, Ike Kurniati, Syarif Hidayatullah dimana algoritma *Neural Network* digunakan dalam Penyusunan Strategi Pemberantasan Buta Aksara Dan Berhitung Di Negara-Negara Asean. Data yang digunakan dalam penelitian adalah data primer dan data sekunder dengan rentang waktu tahun 2004 sampai dengan tahun 2019 dengan cakupan data negara-negara di wilayah ASEAN (Sucahyo, Nur; Kurniati, Ike; Hidayatullah, 2020).

Tujuan dari penelitian ini untuk menemukan poin-poin strategis yang dapat dimanfaatkan dan digunakan sebagai referensi dalam rangka implementasi upaya-upaya menekan jumlah buta huruf di Indonesia.

## B. METODE PENELITIAN

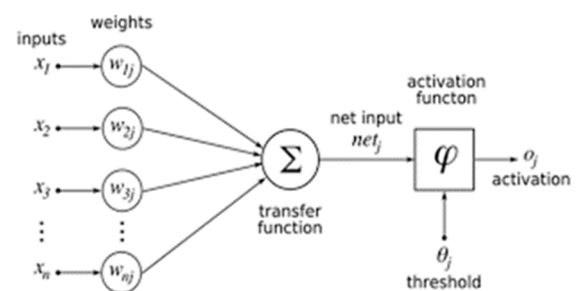
Metode penelitian yang digunakan adalah sebagai berikut: Observasi untuk mengumpulkan data-data yang berhubungan dengan faktor-faktor yang mempengaruhi angka buta huruf di Indonesia yang bersumber dari bps.go.id.

Studi Pustaka untuk mempelajari, meneliti, dan mengimplementasikan topik penelitian berdasarkan paper, jurnal ilmiah serta artikel-artikel yang berkaitan dengan permasalahan yang dibahas pada penelitian ini. informasi dan wawasan secara lebih mendalam.

Data yang digunakan dalam penelitian ini merupakan data sekunder yang dimiliki oleh Badan Pusat Statistik. Pengumpulan data dilakukan dengan cara mengunduh data dari website Badan Pusat Statistik. Data yang ditarik merupakan data dalam kurun waktu 2011 – 2019 dengan format Microsoft Excel.

## C. HASIL DAN PEMBAHASAN

*Artificial Neural Network* (ANN) atau Jaringan Syaraf Tiruan merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Cara kerja *Neural Network* dapat dianalogikan sebagaimana halnya manusia belajar dengan menggunakan contoh atau yang disebut sebagai supervised learning.



Gambar 1. *Artificial Neural Network*

*Neural Network* memproses informasi berdasarkan cara kerja otak manusia. Dalam hal ini *Neural Network* terdiri dari sejumlah besar elemen pemrosesan yang saling terhubung dan bekerja secara paralel untuk memecahkan suatu masalah tertentu. ANN adalah sekumpulan neuron yang terorganisir dalam lapisan-lapisan (*layers*), diantaranya:

**Input layer:** lapisan yang membawa data masuk kedalam system untuk kemudian di proses pada layer selanjutnya.

**Hidden layer:** lapisan antara input layer dan output layer, dimana artificial neuron yang memiliki sekumpulan input pembobot 'weight' dan prosedur untuk menghasilkan output neuron melalui *activation function*.

**Output layer:** lapisan terakhir dari neuron yang menghasilkan output system.





tinggi APS berarti makin banyak usia sekolah yang bersekolah di suatu daerah.

**Buta huruf** adalah persentase penduduk buta huruf di Indonesia yang dikelompokkan berdasarkan kelompok umur.

Dari gambar di atas dapat diketahui jika data training yang digunakan terdiri dari beberapa field, yaitu:

Tabel 1. Kamus Data

no	nama field	keterangan
1	Provinsi	lokasi atau wilayah dimana data tersebut diambil
2	tahun	tahun pengambilan data
3	apk_sd	angka partisipasi kasar pada jenjang pendidikan sekolah dasar
4	apk_smp	angka partisipasi kasar pada jenjang pendidikan sekolah menengah pertama
5	apk_sma	angka partisipasi kasar pada jenjang pendidikan sekolah menengah atas
6	apm_sd	angka partisipasi murni pada jenjang pendidikan sekolah dasar
7	apm_smp	angka partisipasi murni pada jenjang pendidikan sekolah menengah pertama
8	apm_sma	angka partisipasi murni pada jenjang pendidikan sekolah menengah atas
9	aps_7_12	angka partisipasi sekolah pada kelompok umur 7 sampai 12 tahun
10	aps_13_15	angka partisipasi sekolah pada kelompok umur 13 sampai 15 tahun
11	aps_16_18	angka partisipasi sekolah pada kelompok umur 16 sampai 18 tahun
12	aps_19_24	angka partisipasi sekolah pada kelompok umur 19 sampai 24 tahun
13	butahuruf_15	angka persentase buta huruf kelompok umur kurang dari 15 tahun

no	nama field	keterangan
14	butahuruf_15_44	angka persentase buta huruf kelompok umur 15 sampai 44 tahun
15	butahuruf_45	angka persentase buta huruf kelompok umur lebih dari 45 tahun

Adapun tipe data dari masing-masing *field* yang ada pada *data training* adalah sebagai berikut :

```
load_data_train.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 140 entries, 0 to 139
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Provinsi               140 non-null    object
1   tahun                 140 non-null    int64
2   apk_sd                140 non-null    float64
3   apk_smp               140 non-null    float64
4   apk_sma               140 non-null    float64
5   apm_sd                140 non-null    float64
6   apm_smp               140 non-null    float64
7   apm_sma               140 non-null    float64
8   aps_7-12              140 non-null    float64
9   aps_13-15             140 non-null    float64
10  aps_16-18              140 non-null    float64
11  aps_19-24              140 non-null    float64
12  butahuruf_<15         140 non-null    float64
13  butahuruf_15-44      140 non-null    float64
14  butahuruf_>45         140 non-null    float64
dtypes: float64(13), int64(1), object(1)
memory usage: 16.5+ KB
```

Gambar 5. Tipe Data dari *Data Training*

Keterangan Code: `load_data_train.info()` digunakan untuk menampilkan data training mulai dari jumlah kolom dan baris, nama field dan tipe data field.

*Data testing* adalah bagian dari dataset yang digunakan untuk mengetahui performa algoritma yang sudah dilatih sebelumnya dengan menggunakan *data training*. Adapun data yang digunakan menjadi *data testing* adalah data dalam kurun waktu 2015 – 2019, dengan total data sebanyak 175 data dan terdiri dari 15 *field* seperti pada gambar berikut :

```
[6] path_data_test = "/content/drive/MyDrive/TA_dita_yuliyanti/data/data_testing.xlsx"
[7] load_data_test = pd.read_excel(path_data_test)
load_data_test
```

Gambar 6. Data Testing

Keterangan Code: Code `path_data_test` digunakan untuk mentautkan file data testing yang berada di dalam google drive agar terhubung dengan google colab guna untuk memproses data.

Code `load_data_test` digunakan untuk membaca dan menampilkan data testing yang telah ditautkan sebelumnya.

Adapun tipe data dari masing-masing *field* yang ada pada *data testing* adalah sebagai berikut:

```
load_data_test.info()

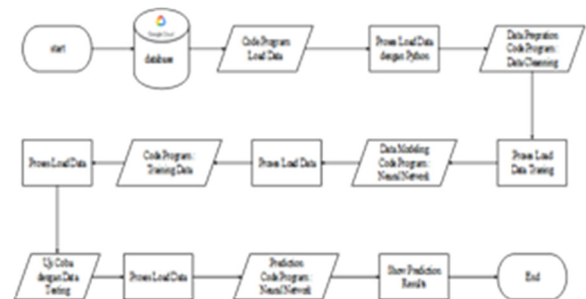
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 175 entries, 0 to 174
Data columns (total 15 columns):
 #   Column                Non-Null Count  Dtype
---  ---                ---
 0   Provinsi              175 non-null   object
 1   tahun                175 non-null   int64
 2   apk_sd               175 non-null   float64
 3   apk_smp              175 non-null   float64
 4   apk_sma              175 non-null   float64
 5   apm_sd               175 non-null   float64
 6   apm_smp              175 non-null   float64
 7   apm_sma              175 non-null   float64
 8   aps_7-12             175 non-null   float64
 9   aps_13-15            175 non-null   float64
10  aps_16-18            175 non-null   float64
11  aps_19-24            175 non-null   float64
12  butahuruf_<15        175 non-null   float64
13  butahuruf_15-44     175 non-null   float64
14  butahuruf_>45       175 non-null   float64
dtypes: float64(13), int64(1), object(1)
memory usage: 20.6+ KB
```

Gambar 7. Tipe Data dari *Data Testing*

Keterangan code: `load_data_test.info()` digunakan untuk menampilkan data testing mulai dari jumlah kolom dan baris, nama field dan tipe data field.

Pengolahan data untuk menghasilkan informasi atau menghasilkan pengetahuan dari data mentah (*raw data*). Data historis penjualan yang telah dikumpulkan akan melalui beberapa tahapan pengolahan data

sebelum akhirnya dapat menghasilkan informasi yang ditargetkan. Langkah-langkah pengolahan data digambarkan pada *flowchart* dibawah ini.



Gambar 8. *Flowchart* Pengolahan Data

Dari *flowchart* diatas dimulai dengan import data dari google cloud lalu ditampilkan dalam google colab dan di proses dengan python. Setelah itu data yang sudah ditampilkan di lakukan proses *cleaning* untuk memeriksa data apakah ada yang tidak sesuai dan setelah dilakukan proses *cleaning* lalu ditampilkan data yang sudah melewati proses *cleaning*. Lalu dilakukan pemodelan data *training* dengan menggunakan algoritma *neural network* lalu ditampilkan hasil pemodelan. Selajutnya memproses data *testing* seperti data *training* sebelumnya, ketika sudah sesuai maka dapat dilakukan prediksi menggunakan algoritma *neural network* lalu menampilkan hasil proses prediksi. *Preprocessing data* tahapan dalam mengubah data mentah (*raw data*) menjadi data berkualitas (*input* yang baik) yang dapat digunakan untuk membuat permodelan data. *Preprocessing data* dilakukan dalam beberapa langkah dan dengan memanfaatkan *library* dan modul yang ada dalam pemrograman *python*.

### Menyimpan Data ke dalam *Google Cloud*

Tahap awal dalam *preprocessing data* adalah menyimpan data dalam bentuk *excel* ke dalam *Google Cloud*. Penyimpanan data ke dalam *Google Cloud* dilakukan agar data dapat diproses di dalam *Google Colab*.

Berikut adalah gambar penyimpanan data di dalam *Google Cloud*.



Gambar 9. Penyimpanan *Dataset* ke dalam *Google Cloud*

**Import Library** Setelah menyimpan data ke dalam *Google Cloud* langkah selanjutnya yang dilakukan adalah melakukan *import library* dan modul-modul yang dibutuhkan untuk melakukan pemrosesan data. Berikut adalah *library* dan modul yang digunakan untuk persiapan data dan pengaplikasian metode *Neural Network* :

```
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import seaborn as sns
import sklearn
from sklearn.neural_network import MLPRegressor

# Import necessary modules
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from math import sqrt
from sklearn.metrics import r2_score
```

Gambar 10. Library Python

Mengubah *data training* yang sudah dikumpulkan ke dalam bentuk *DataFrame*. Tujuan dari proses ini adalah agar data yang sudah dikumpulkan dapat *ter-capture* ke dalam *environment python* karena sebagian besar *library* dalam *python* hanya dapat memproses data yang sudah berbentuk *DataFrame* tujuan lain dari dilakukannya proses ini adalah untuk meringankan kinerja mesin dalam memproses data. Proses pengubahan *data training* ke dalam *DataFrame* adalah sebagai berikut:

	Provinsi	aps_7-12	aps_13-15	aps_16-18	aps_19-24	butahuruf_<15	butahuruf_15-44	butahuruf_>45
0	ACEH	10.07	97.15	79.28	92.31	74.07	81.27	88.89
1	ACEH	10.08	96.47	77.02	94.08	75.01	81.02	86.38
2	ACEH	10.71	96.29	75.08	97.09	82.07	82.40	89.88
3	ACEH	11.06	97.07	81.03	97.00	82.03	89.34	97.28
4	SUMATERA	10.76	89.18	79.07	91.01	89.08	87.46	90.34
18	INDONESIA	10.07	97.15	79.28	92.31	74.07	81.27	88.89
19	INDONESIA	10.08	96.47	77.02	94.08	75.01	81.02	86.38
20	INDONESIA	10.71	96.29	75.08	97.09	82.07	82.40	89.88
21	INDONESIA	11.06	97.07	81.03	97.00	82.03	89.34	97.28
22	INDONESIA	10.76	89.18	79.07	91.01	89.08	87.46	90.34

## 11. Data Frame Training

### *Data Cleaning* (Pembersihan Data)

Data yang telah diubah ke dalam bentuk *DataFrame* kemudian akan memasuki tahap *data cleaning* (pembersihan data). Pada tahap ini akan dilakukan pengecekan akan kesalahan yang mungkin ada pada data seperti data kosong (*missing value*). Proses *data cleaning* (pembersihan data) adalah sebagai berikut:

```
[15] df_train.isna().any()
Provinsi      False
tahun         False
apk_sd        False
apk_smp        False
apk_sma        False
apm_sd         False
apm_smp        False
apm_sma        False
aps_7-12       False
aps_13-15      False
aps_16-18      False
aps_19-24      False
butahuruf_<15  False
butahuruf_15-44 False
butahuruf_>45  False
dtype: bool

[14] df_train.isnull().sum()
Provinsi      0
tahun         0
apk_sd        0
apk_smp        0
apk_sma        0
apm_sd         0
apm_smp        0
apm_sma        0
aps_7-12       0
aps_13-15      0
aps_16-18      0
aps_19-24      0
butahuruf_<15  0
butahuruf_15-44 0
butahuruf_>45  0
dtype: int64
```

Gambar 12. Data Cleaning

Keterangan code: `df_train.isna().any()` adalah mengembalikan nilai boolean untuk setiap kolom. Jika setidaknya ada satu nilai yang hilang di kolom tersebut, hasilnya adalah `True`. `df_train.isnull().sum()` adalah untuk mengetahui ada atau tidaknya nilai atau isi kolom yang hilang. Dari gambar diketahui jika data training tidak memiliki data kosong (*missing value*). Maka dapat disimpulkan jika data training yang akan digunakan merupakan data yang berkualitas dan siap untuk diteruskan untuk proses selanjutnya.

Pemodelan data adalah proses menciptakan model data untuk sistem

informasi dengan menerapkan algoritma tertentu. Proses permodelan dengan menggunakan metode Neural Network terdiri dari beberapa langkah berikut ini :

- a. Membaca Data dan Pemeriksaan Data  
 Baris kode pertama membaca data sebagai pandas dataframe, sedangkan baris kedua mencetak jumlah baris dan kolom pada data. Baris ketiga memberikan ringkasan statistik yang dialihkan dari variabel.

```
[ ] df_load_data_train = pd.read_excel(path_data_train)
print(df_load_data_train.shape)
df_load_data_train.describe().transpose()

(140, 15)
```

	count	mean	std	min	25%	50%	75%	max
tahun	140.0	2012.500000	1.122048	2011.0	2011.7500	2012.5000	2013.2500	2014.00
apk_sd	140.0	103.365000	18.450312	0.0	103.9925	106.9300	109.5600	114.68
apk_smp	140.0	84.706214	15.753781	0.0	84.2550	88.0550	90.8075	97.59
apk_sma	140.0	69.294571	15.020136	0.0	64.1250	71.3300	78.1125	94.62
apm_sd	140.0	90.582214	16.224054	0.0	91.6250	93.5100	96.1500	98.98
apm_smp	140.0	68.022786	13.958064	0.0	64.5200	70.1300	74.9775	85.20
apm_sma	140.0	52.329500	11.934278	0.0	48.4775	53.3050	59.2750	70.83
aps_7_12	140.0	94.881143	16.788629	0.0	97.7175	98.3700	99.0500	99.96
aps_13_15	140.0	87.943000	15.963480	0.0	87.5650	90.9000	94.3400	99.48
aps_16_18	140.0	63.705214	13.383216	0.0	59.6875	65.4450	70.3350	86.44
aps_19_24	140.0	19.909929	7.940267	0.0	15.3100	19.3000	23.5975	49.08
butahuruf_15	140.0	5.913500	5.949914	0.0	2.5500	4.0500	8.1600	35.47
butahuruf_15_44	140.0	2.511571	5.288935	0.0	0.6575	1.1700	2.2800	34.55
butahuruf_45	140.0	13.368786	9.518560	0.0	6.6850	10.5700	20.0150	42.55

Gambar 13. Load Data Training

- b. Menentukan Variabel Target Prediksi  
 Variabel target adalah sebuah variabel yang menjadi goal dari sebuah prediksi. Adapun variabel target dalam prediksi ini adalah butahuruf\_15, butahuruf\_15\_44, butahuruf\_45

```
[ ] target_column = ['butahuruf_15', 'butahuruf_15_44', 'butahuruf_45']
```

Gambar 14. Variabel target

- c. Membuat Variabel Input  
 Variabel input adalah sebuah variabel yang dibuat setelah menentukan variabel target untuk mengurangi variabel target dari data yang digunakan.

```
#input variabel
input = list(set(list(df_load_data_train.columns))-set(target_column))

input
```

Gambar 15. Variabel input

- d. Membuat Dataset Training dan Testing  
 Baris kode pertama membuat array variabel independen (X) dan dependen (y). Baris ketiga membagi data menjadi Dataset Training dan Testing, dan baris keempat mencetak bentuk Dataset Training dan Testing.

```
[ ] X = df_load_data_train[input].values
y = df_load_data_train[target_column].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=48)
print(X_train.shape); print(X_test.shape)

(98, 12)
(42, 12)
```

Gambar 16. Dataset Training dan Testing

- e. Membuat, Memprediksi dan Mengevaluasi Model Neural Network.  
 Kode pertama digunakan untuk mengimport dan membangun scikit-learn library estimator Multi-Layer Perceptron Classifier.  
 Kode StandardScaler untuk menghilangkan mean (terpusat pada 0) dan menskalakan ke variansi (deviasi standar = 1), dengan asumsi data terdistribusi normal (gauss) untuk semua fitur.

```
[14] from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score

[15] #untuk membuat data fit dalam kurva distribusi normal
scalar = StandardScaler()
scalar.fit(X_train)
# transform data kedalam distribusi normal sebagai mean dan stv.dev
X_train = scalar.transform(X_train)

X_test = scalar.transform(X_test)
```

Gambar 17. Kode scikit-learn library

Kode dibawah ini untuk membuat model dengan argumen 'hidden\_layer\_sizes' yang disetel ke tiga lapisan, yang memiliki jumlah neuron yang sama dengan jumlah fitur dalam data.

```
[ ] mlp = MLPRegressor(hidden_layer_sizes=(3,3,3), activation='relu', solver='lbfgs', max_iter=100)
mlp.fit(X_train,y_train)

UserWarning: /usr/local/lib/python3.11/site-packages/sklearn/neural_network/multilayer_perceptron.py:476: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Decrease the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/auto_examples/neural_networks/mlp_parameters_tuning.html
self._iter_ = _check_optimize_result("lbfgs", opt_res, self._max_iter)
MLPRegressor(activation='relu', alpha=0.0001, batch_size='auto', beta=1e-05,
data_init=0.001, early_stopping=False, epsilon=0.01,
hidden_layer_sizes=(3, 3, 3), learning_rate='constant',
learning_rate_init=0.001, max_fun=10000, max_iter=100,
momentum=0.1, n_iter_no_change=10, nesterov_momentum=True,
power_t=0.5, random_state=None, shuffle=True, solver='lbfgs',
tol=0.0001, validation_fraction=0.1, verbose=False,
warm_start=False)
```

Gambar 18. Hidden layer



Kode dibawah ini menggunakan model terlatih untuk menghasilkan prediksi pada dataset pelatihan dan pengujian.

```
[17] prediction=mlp.predict(X_test)

hasil_prediksi=(X_test, prediction)
hasil_prediksi

(array([[ -3.66150347e-01,  2.68575443e-01,  1.18357483e+00,
         1.33743223e-01,  2.30868508e-01, -4.43606975e-01,
         3.91877771e-01,  3.77638466e-01,  8.94837687e-02,
         1.49674270e-01,  1.84188808e-01,  4.11283710e-01],
        [-5.31163253e-01, -3.73726610e-01,  1.72244920e-01,
         -3.26641818e-01, -6.72625279e-02, -4.43606975e-01,
         1.21116356e+00, -4.00926612e-02, -3.96363687e-02,
         -2.13877393e-01,  8.39700983e-02, -2.54568433e-01,
        [-2.13094898e-01, -7.70554809e-01, -1.00021465e+00,
         1.80144235e-01, -6.15778259e-02, -1.33082093e+00,
         -4.27408022e-01, -8.63513885e-01,  6.43770753e-02,
         -4.13758386e-01,  1.45214866e-01, -8.53500763e-01],
        [-7.41614495e-01, -7.00601120e-01, -1.25787359e+00,
         -2.52699205e-01,  2.50449148e-01, -4.43606975e-01,
         -5.29818747e-01, -1.41184388e+00,  1.13873128e-01,
         -2.34879381e-01,  6.93548698e-02, -7.49775052e-01],
        [ 5.27470270e-01,  3.33000325e+00,  1.52439747e+00,
         -1.70763418e-01,  2.71924689e-01,  4.43606975e-01,
         -1.34910454e+00,  1.12788727e+00,  5.31361572e-01,
         5.81301343e-01,  2.99022747e-01,  1.44854082e+00],
        [ 5.39427726e-01,  6.81938151e-01,  1.00313932e+00,
         3.67198314e-01,  1.86022526e-01,  4.43606975e-01,
         -2.22586574e-01,  9.47271821e-01,  3.16878677e-01,
```

Gambar 19. Hasil Prediksi Data Train

- f. Menemukan Akurasi Prediksi  
Selanjutnya ke tahap evaluasi model, Untuk mengevaluasi model mlp dengan mencetak akurasi dan melihat statistik pengujian seperti Weights dan intercepts.

```
#menemukan akurasi
print('Weights')
print(mlp.coefs_)
print('intercepts-----')
print(mlp.intercepts_)
print('akurasi model')
mlp.score(X_test,y_test)

akurasi model
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:434:
  "multioutput='uniform_average'."), FutureWarning)
0.691150783023893
```

Gambar 20. Akurasi Model

Dari laporan tersebut, dapat dilihat hasil akurasi model sebesar 0,69 atau 69 % yang berarti pemodelan data berjalan dengan baik meskipun hasil akurasi tidak terlalu bagus.

Mesin learning mampu mendeteksi dan memilih masalah yang ada di dalam programnya, termasuk pengumpulan data data asli dapat berupa excel, csv, ms acces, dan lain sebagainya.

```
#load data untuk prediksi
path_data_test='/content/drive/MyDrive/TA_dita_yuliyanti/data/data_testing.xlsx'

load_data_test = pd.read_excel(path_data_test)

df_data_test=pd.DataFrame(load_data_test)

df_data_test["Provinsi"] = df_data_test["Provinsi"].astype('category')

df_data_test["Provinsi"] = df_data_test["Provinsi"].cat.codes

input = list(set(list(df_data_test.columns)))
```

Gambar 21. Proses Load Data

Keterangan code :

- path\_data\_test digunakan untuk mengimport data dari google drive ke dalam google colab.
- load\_data\_test digunakan untuk membaca data yang telah di import dari google drive.
- df\_data\_test=pd.DataFrame(load\_data\_test) adalah agar data yang sudah dikumpulkan dapat ter-capture ke dalam environment python karena sebagian besar library dalam python hanya dapat memproses data yang sudah berbentuk DataFrame tujuan lain dari dilakukannya proses ini adalah untuk meringankan kinerja mesin dalam memproses data.
- df\_data\_test["Provinsi"]=df\_data\_test["Provinsi"].astype('category') digunakan untuk mengubah tipe data kolom provinsi menjadi tipe data kategori guna meminimalisir kesalahan proses prediksi.
- input=list(set(list(df\_data\_test.columns))) digunakan untuk menentukan variabel input dalam proses prediksi.
- Setelah variabel input dibuat maka dijalankan kode dibawah ini untuk menemukan values, apakah ada kesalahan atau tidak pada data yang telah di proses sebelumnya. jika data benar maka tampilan yang muncul dari kode tersebut adalah array-array dari data yang telah diproses.

```
input_predict = df_data_test[input].values
input_predict

array([[97.99, 81.43, 83.33, ..., 99.9, 33.07, 85.55],
       [96.47, 76.23, 84.14, ..., 99.35, 25.16, 78.48],
       [98.12, 82.53, 81.42, ..., 99.44, 33.13, 76. ],
       ...,
       [93.76, 81.49, 96.41, ..., 97.68, 31.48, 69.92],
       [79.19, 63.5, 76.33, ..., 82.67, 22.91, 57.19],
       [97.64, 72.36, 83.98, ..., 99.24, 25.21, 79.4 ]])
```

```
input_predict.shape

(175, 12)
```

Gambar 22. Variabel Input Predict

Setelah variabel input dibuat maka selanjutnya dilakukan implementasi model dengan menggunakan data baru.

```
y_testing_pred = mlp.predict(input_predict)

y_testing_pred

array([[146.05409646, -1.46388827, 461.87317496],
       [110.08232314, -0.86819108, 347.91766944],
       [102.99518757, -0.7508272, 325.46622927],
       [122.81146326, -1.07898727, 388.24249815],
       [157.60325974, -1.6551439, 498.45993781],
       [125.59224312, -1.12503735, 397.05177122],
       [154.31535413, -1.60069576, 488.04413366],
       [136.472661, -1.30521847, 431.52000579],
       [163.84859452, -1.75856745, 518.24462498],
       [149.48957833, -1.5207803, 472.75648809],
       [167.8613767, -1.82501964, 530.95677618],
       [167.3965979, -1.81732285, 529.48439665],
       [160.37693307, -1.70107629, 507.24669801],
       [116.69336662, -0.97767081, 368.86089055],
       [157.4730022, -1.65298682, 498.04729304],
       [170.03874889, -1.86107721, 537.85450531],
       [160.77680516, -1.70769823, 508.51345863],
       [132.44679174, -1.23854956, 418.76639583],
       [117.21248786, -0.98626752, 370.5054223 ],
       [153.82674215, -1.59260429, 486.49625264],
       [143.27120057, -1.41780316, 453.05719846],
       [164.39101264, -1.76754997, 519.96295925],
       [128.7143486, -1.17673982, 406.94233472],
```

Gambar 23. Array Hasil Prediksi

Hasil yang ditampilkan diatas adalah berupa array-array yang tak bisa dibaca maka agar bisa dibaca dan diketahui hasilnya perlu selanjutnya dilakukan penggabungan data testing dan data hasil prediksi yang telah dilakukan sebelumnya.

```
df_result_predict = pd.concat([df_data_test, df_testing_pred], sort=False, axis=1)

df_result_predict

Provincial Tahun  apl_01  apl_02  apl_03  apl_04  apl_05  apl_06  apl_07  apl_08  apl_09  apl_10  apl_11  apl_12  apl_13  apl_14  apl_15  apl_16  apl_17  apl_18  apl_19  apl_20  apl_21  apl_22  apl_23  apl_24  apl_25  apl_26  apl_27  apl_28  apl_29  apl_30  apl_31  apl_32  apl_33  apl_34  apl_35  apl_36  apl_37  apl_38  apl_39  apl_40  apl_41  apl_42  apl_43  apl_44  apl_45  apl_46  apl_47  apl_48  apl_49  apl_50  apl_51  apl_52  apl_53  apl_54  apl_55  apl_56  apl_57  apl_58  apl_59  apl_60  apl_61  apl_62  apl_63  apl_64  apl_65  apl_66  apl_67  apl_68  apl_69  apl_70  apl_71  apl_72  apl_73  apl_74  apl_75  apl_76  apl_77  apl_78  apl_79  apl_80  apl_81  apl_82  apl_83  apl_84  apl_85  apl_86  apl_87  apl_88  apl_89  apl_90  apl_91  apl_92  apl_93  apl_94  apl_95  apl_96  apl_97  apl_98  apl_99  apl_100  apl_101  apl_102  apl_103  apl_104  apl_105  apl_106  apl_107  apl_108  apl_109  apl_110  apl_111  apl_112  apl_113  apl_114  apl_115  apl_116  apl_117  apl_118  apl_119  apl_120  apl_121  apl_122  apl_123  apl_124  apl_125  apl_126  apl_127  apl_128  apl_129  apl_130  apl_131  apl_132  apl_133  apl_134  apl_135  apl_136  apl_137  apl_138  apl_139  apl_140  apl_141  apl_142  apl_143  apl_144  apl_145  apl_146  apl_147  apl_148  apl_149  apl_150  apl_151  apl_152  apl_153  apl_154  apl_155  apl_156  apl_157  apl_158  apl_159  apl_160  apl_161  apl_162  apl_163  apl_164  apl_165  apl_166  apl_167  apl_168  apl_169  apl_170  apl_171  apl_172  apl_173  apl_174  apl_175  apl_176  apl_177  apl_178  apl_179  apl_180  apl_181  apl_182  apl_183  apl_184  apl_185  apl_186  apl_187  apl_188  apl_189  apl_190  apl_191  apl_192  apl_193  apl_194  apl_195  apl_196  apl_197  apl_198  apl_199  apl_200  apl_201  apl_202  apl_203  apl_204  apl_205  apl_206  apl_207  apl_208  apl_209  apl_210  apl_211  apl_212  apl_213  apl_214  apl_215  apl_216  apl_217  apl_218  apl_219  apl_220  apl_221  apl_222  apl_223  apl_224  apl_225  apl_226  apl_227  apl_228  apl_229  apl_230  apl_231  apl_232  apl_233  apl_234  apl_235  apl_236  apl_237  apl_238  apl_239  apl_240  apl_241  apl_242  apl_243  apl_244  apl_245  apl_246  apl_247  apl_248  apl_249  apl_250  apl_251  apl_252  apl_253  apl_254  apl_255  apl_256  apl_257  apl_258  apl_259  apl_260  apl_261  apl_262  apl_263  apl_264  apl_265  apl_266  apl_267  apl_268  apl_269  apl_270  apl_271  apl_272  apl_273  apl_274  apl_275  apl_276  apl_277  apl_278  apl_279  apl_280  apl_281  apl_282  apl_283  apl_284  apl_285  apl_286  apl_287  apl_288  apl_289  apl_290  apl_291  apl_292  apl_293  apl_294  apl_295  apl_296  apl_297  apl_298  apl_299  apl_300  apl_301  apl_302  apl_303  apl_304  apl_305  apl_306  apl_307  apl_308  apl_309  apl_310  apl_311  apl_312  apl_313  apl_314  apl_315  apl_316  apl_317  apl_318  apl_319  apl_320  apl_321  apl_322  apl_323  apl_324  apl_325  apl_326  apl_327  apl_328  apl_329  apl_330  apl_331  apl_332  apl_333  apl_334  apl_335  apl_336  apl_337  apl_338  apl_339  apl_340  apl_341  apl_342  apl_343  apl_344  apl_345  apl_346  apl_347  apl_348  apl_349  apl_350  apl_351  apl_352  apl_353  apl_354  apl_355  apl_356  apl_357  apl_358  apl_359  apl_360  apl_361  apl_362  apl_363  apl_364  apl_365  apl_366  apl_367  apl_368  apl_369  apl_370  apl_371  apl_372  apl_373  apl_374  apl_375  apl_376  apl_377  apl_378  apl_379  apl_380  apl_381  apl_382  apl_383  apl_384  apl_385  apl_386  apl_387  apl_388  apl_389  apl_390  apl_391  apl_392  apl_393  apl_394  apl_395  apl_396  apl_397  apl_398  apl_399  apl_400  apl_401  apl_402  apl_403  apl_404  apl_405  apl_406  apl_407  apl_408  apl_409  apl_410  apl_411  apl_412  apl_413  apl_414  apl_415  apl_416  apl_417  apl_418  apl_419  apl_420  apl_421  apl_422  apl_423  apl_424  apl_425  apl_426  apl_427  apl_428  apl_429  apl_430  apl_431  apl_432  apl_433  apl_434  apl_435  apl_436  apl_437  apl_438  apl_439  apl_440  apl_441  apl_442  apl_443  apl_444  apl_445  apl_446  apl_447  apl_448  apl_449  apl_450  apl_451  apl_452  apl_453  apl_454  apl_455  apl_456  apl_457  apl_458  apl_459  apl_460  apl_461  apl_462  apl_463  apl_464  apl_465  apl_466  apl_467  apl_468  apl_469  apl_470  apl_471  apl_472  apl_473  apl_474  apl_475  apl_476  apl_477  apl_478  apl_479  apl_480  apl_481  apl_482  apl_483  apl_484  apl_485  apl_486  apl_487  apl_488  apl_489  apl_490  apl_491  apl_492  apl_493  apl_494  apl_495  apl_496  apl_497  apl_498  apl_499  apl_500  apl_501  apl_502  apl_503  apl_504  apl_505  apl_506  apl_507  apl_508  apl_509  apl_510  apl_511  apl_512  apl_513  apl_514  apl_515  apl_516  apl_517  apl_518  apl_519  apl_520  apl_521  apl_522  apl_523  apl_524  apl_525  apl_526  apl_527  apl_528  apl_529  apl_530  apl_531  apl_532  apl_533  apl_534  apl_535  apl_536  apl_537  apl_538  apl_539  apl_540  apl_541  apl_542  apl_543  apl_544  apl_545  apl_546  apl_547  apl_548  apl_549  apl_550  apl_551  apl_552  apl_553  apl_554  apl_555  apl_556  apl_557  apl_558  apl_559  apl_560  apl_561  apl_562  apl_563  apl_564  apl_565  apl_566  apl_567  apl_568  apl_569  apl_570  apl_571  apl_572  apl_573  apl_574  apl_575  apl_576  apl_577  apl_578  apl_579  apl_580  apl_581  apl_582  apl_583  apl_584  apl_585  apl_586  apl_587  apl_588  apl_589  apl_590  apl_591  apl_592  apl_593  apl_594  apl_595  apl_596  apl_597  apl_598  apl_599  apl_600  apl_601  apl_602  apl_603  apl_604  apl_605  apl_606  apl_607  apl_608  apl_609  apl_610  apl_611  apl_612  apl_613  apl_614  apl_615  apl_616  apl_617  apl_618  apl_619  apl_620  apl_621  apl_622  apl_623  apl_624  apl_625  apl_626  apl_627  apl_628  apl_629  apl_630  apl_631  apl_632  apl_633  apl_634  apl_635  apl_636  apl_637  apl_638  apl_639  apl_640  apl_641  apl_642  apl_643  apl_644  apl_645  apl_646  apl_647  apl_648  apl_649  apl_650  apl_651  apl_652  apl_653  apl_654  apl_655  apl_656  apl_657  apl_658  apl_659  apl_660  apl_661  apl_662  apl_663  apl_664  apl_665  apl_666  apl_667  apl_668  apl_669  apl_670  apl_671  apl_672  apl_673  apl_674  apl_675  apl_676  apl_677  apl_678  apl_679  apl_680  apl_681  apl_682  apl_683  apl_684  apl_685  apl_686  apl_687  apl_688  apl_689  apl_690  apl_691  apl_692  apl_693  apl_694  apl_695  apl_696  apl_697  apl_698  apl_699  apl_700  apl_701  apl_702  apl_703  apl_704  apl_705  apl_706  apl_707  apl_708  apl_709  apl_710  apl_711  apl_712  apl_713  apl_714  apl_715  apl_716  apl_717  apl_718  apl_719  apl_720  apl_721  apl_722  apl_723  apl_724  apl_725  apl_726  apl_727  apl_728  apl_729  apl_730  apl_731  apl_732  apl_733  apl_734  apl_735  apl_736  apl_737  apl_738  apl_739  apl_740  apl_741  apl_742  apl_743  apl_744  apl_745  apl_746  apl_747  apl_748  apl_749  apl_750  apl_751  apl_752  apl_753  apl_754  apl_755  apl_756  apl_757  apl_758  apl_759  apl_760  apl_761  apl_762  apl_763  apl_764  apl_765  apl_766  apl_767  apl_768  apl_769  apl_770  apl_771  apl_772  apl_773  apl_774  apl_775  apl_776  apl_777  apl_778  apl_779  apl_780  apl_781  apl_782  apl_783  apl_784  apl_785  apl_786  apl_787  apl_788  apl_789  apl_790  apl_791  apl_792  apl_793  apl_794  apl_795  apl_796  apl_797  apl_798  apl_799  apl_800  apl_801  apl_802  apl_803  apl_804  apl_805  apl_806  apl_807  apl_808  apl_809  apl_810  apl_811  apl_812  apl_813  apl_814  apl_815  apl_816  apl_817  apl_818  apl_819  apl_820  apl_821  apl_822  apl_823  apl_824  apl_825  apl_826  apl_827  apl_828  apl_829  apl_830  apl_831  apl_832  apl_833  apl_834  apl_835  apl_836  apl_837  apl_838  apl_839  apl_840  apl_841  apl_842  apl_843  apl_844  apl_845  apl_846  apl_847  apl_848  apl_849  apl_850  apl_851  apl_852  apl_853  apl_854  apl_855  apl_856  apl_857  apl_858  apl_859  apl_860  apl_861  apl_862  apl_863  apl_864  apl_865  apl_866  apl_867  apl_868  apl_869  apl_870  apl_871  apl_872  apl_873  apl_874  apl_875  apl_876  apl_877  apl_878  apl_879  apl_880  apl_881  apl_882  apl_883  apl_884  apl_885  apl_886  apl_887  apl_888  apl_889  apl_890  apl_891  apl_892  apl_893  apl_894  apl_895  apl_896  apl_897  apl_898  apl_899  apl_900  apl_901  apl_902  apl_903  apl_904  apl_905  apl_906  apl_907  apl_908  apl_909  apl_910  apl_911  apl_912  apl_913  apl_914  apl_915  apl_916  apl_917  apl_918  apl_919  apl_920  apl_921  apl_922  apl_923  apl_924  apl_925  apl_926  apl_927  apl_928  apl_929  apl_930  apl_931  apl_932  apl_933  apl_934  apl_935  apl_936  apl_937  apl_938  apl_939  apl_940  apl_941  apl_942  apl_943  apl_944  apl_945  apl_946  apl_947  apl_948  apl_949  apl_950  apl_951  apl_952  apl_953  apl_954  apl_955  apl_956  apl_957  apl_958  apl_959  apl_960  apl_961  apl_962  apl_963  apl_964  apl_965  apl_966  apl_967  apl_968  apl_969  apl_970  apl_971  apl_972  apl_973  apl_974  apl_975  apl_976  apl_977  apl_978  apl_979  apl_980  apl_981  apl_982  apl_983  apl_984  apl_985  apl_986  apl_987  apl_988  apl_989  apl_990  apl_991  apl_992  apl_993  apl_994  apl_995  apl_996  apl_997  apl_998  apl_999  apl_1000  apl_1001  apl_1002  apl_1003  apl_1004  apl_1005  apl_1006  apl_1007  apl_1008  apl_1009  apl_1010  apl_1011  apl_1012  apl_1013  apl_1014  apl_1015  apl_1016  apl_1017  apl_1018  apl_1019  apl_1020  apl_1021  apl_1022  apl_1023  apl_1024  apl_1025  apl_1026  apl_1027  apl_1028  apl_1029  apl_1030  apl_1031  apl_1032  apl_1033  apl_1034  apl_1035  apl_1036  apl_1037  apl_1038  apl_1039  apl_1040  apl_1041  apl_1042  apl_1043  apl_1044  apl_1045  apl_1046  apl_1047  apl_1048  apl_1049  apl_1050  apl_1051  apl_1052  apl_1053  apl_1054  apl_1055  apl_1056  apl_1057  apl_1058  apl_1059  apl_1060  apl_1061  apl_1062  apl_1063  apl_1064  apl_1065  apl_1066  apl_1067  apl_1068  apl_1069  apl_1070  apl_1071  apl_1072  apl_1073  apl_1074  apl_1075  apl_1076  apl_1077  apl_1078  apl_1079  apl_1080  apl_1081  apl_1082  apl_1083  apl_1084  apl_1085  apl_1086  apl_1087  apl_1088  apl_1089  apl_1090  apl_1091  apl_1092  apl_1093  apl_1094  apl_1095  apl_1096  apl_1097  apl_1098  apl_1099  apl_1100  apl_1101  apl_1102  apl_1103  apl_1104  apl_1105  apl_1106  apl_1107  apl_1108  apl_1109  apl_1110  apl_1111  apl_1112  apl_1113  apl_1114  apl_1115  apl_1116  apl_1117  apl_1118  apl_1119  apl_1120  apl_1121  apl_1122  apl_1123  apl_1124  apl_1125  apl_1126  apl_1127  apl_1128  apl_1129  apl_1130  apl_1131  apl_1132  apl_1133  apl_1134  apl_1135  apl_1136  apl_1137  apl_1138  apl_1139  apl_1140  apl_1141  apl_1142  apl_1143  apl_1144  apl_1145  apl_1146  apl_1147  apl_1148  apl_1149  apl_1150  apl_1151  apl_1152  apl_1153  apl_1154  apl_1155  apl_1156  apl_1157  apl_1158  apl_1159  apl_1160  apl_1161  apl_1162  apl_1163  apl_1164  apl_1165  apl_1166  apl_1167  apl_1168  apl_1169  apl_1170  apl_1171  apl_1172  apl_1173  apl_1174  apl_1175  apl_1176  apl_1177  apl_1178  apl_1179  apl_1180  apl_1181  apl_1182  apl_1183  apl_1184  apl_1185  apl_1186  apl_1187  apl_1188  apl_1189  apl_1190  apl_1191  apl_1192  apl_1193  apl_1194  apl_1195  apl_1196  apl_1197  apl_1198  apl_1199  apl_1200  apl_1201  apl_1202  apl_1203  apl_1204  apl_1205  apl_1206  apl_1207  apl_1208  apl_1209  apl_1210  apl_1211  apl_1212  apl_1213  apl_1214  apl_1215  apl_1216  apl_1217  apl_1218  apl_1219  apl_1220  apl_1221  apl_1222  apl_1223  apl_1224  apl_1225  apl_1226  apl_1227  apl_1228  apl_1229  apl_1230  apl_1231  apl_1232  apl_1233  apl_1234  apl_1235  apl_1236  apl_1237  apl_1238  apl_1239  apl_1240  apl_1241  apl_1242  apl_1243  apl_1244  apl_1245  apl_1246  apl_1247  apl_1248  apl_1249  apl_1250  apl_1251  apl_1252  apl_1253  apl_1254  apl_1255  apl_1256  apl_1257  apl_1258  apl_1259  apl_1260  apl_1261  apl_1262  apl_1263  apl_1264  apl_1265  apl_1266  apl_1267  apl_1268  apl_1269  apl_1270  apl_1271  apl_1272  apl_1273  apl_1274  apl_1275  apl_1276  apl_1277  apl_1278  apl_1279  apl_1280  apl_1281  apl_1282  apl_1283  apl_1284  apl_1285  apl_1286  apl_1287  apl_1288  apl_1289  apl_1290  apl_1291  apl_1292  apl_1293  apl_1294  apl_1295  apl_1296  apl_1297  apl_1298  apl_1299  apl_1300  apl_1301  apl_1302  apl_1303  apl_1304  apl_1305  apl_1306  apl_1307  apl_1308  apl_1309  apl_1310  apl_1311  apl_1312  apl_1313  apl_1314  apl_1315  apl_1316  apl_1317  apl_1318  apl_1319  apl_1320  apl_1321  apl_1322  apl_1323  apl_1324  apl_1325  apl_1326  apl_1327  apl_1328  apl_1329  apl_1330  apl_1331  apl_1332  apl_1333  apl_1334  apl_1335  apl_1336  apl_1337  apl_1338  apl_1339  apl_1340  apl_1341  apl_1342  apl_1343  apl_1344  apl_1345  apl_1346  apl_1347  apl_1348  apl_1349  apl_1350  apl_1351  apl_1352  apl_1353  apl_1354  apl_1355  apl_1356  apl_1357  apl_1358  apl_1359  apl_1360  apl_1361  apl_1362  apl_1363  apl_1364  apl_1365  apl_1366  apl_1367  apl_1368  apl_1369  apl_1370  apl_1371  apl_1372  apl_1373  apl_1374  apl_1375  apl_1376  apl_1377  apl_1378  apl_1379  apl_1380  apl_1381  apl_1382  apl_1383  apl_1384  apl_1385  apl_1386  apl_1387  apl_1388  apl_1389  apl_1390  apl_1391  apl_1392  apl_1393  apl_1394  apl_1395  apl_1396  apl_1397  apl_1398  apl_1399  apl_1400  apl_1401  apl_1402  apl_1403  apl_1404  apl_1405  apl_1406  apl_1407  apl_1408  apl_1409  apl_1410  apl_1411  apl_1412  apl_1413  apl_1414  apl_1415  apl_1416  apl_1417  apl_1418  apl_1419  apl_1420  apl_1421  apl_1422  apl_1423  apl_1424  apl_1425  apl_1426  apl_1427  apl_1428  apl_1429  apl_1430  apl_1431  apl_1432  apl_1433  apl_1434  apl_1435  apl_1436  apl_1437  apl_1438  apl_1439  apl_1440  apl_1441  apl_1442  apl_1443  apl_1444  apl_1445  apl_1446  apl_1447  apl_1448  apl_1449  apl_1450  apl_1451  apl_1452  apl_1453  apl_1454  apl_1455  apl_1456  apl_1457  apl_1458  apl_1459  apl_1460  apl_1461  apl_1462  apl_1463  apl_1464  apl_1465  apl_1466  apl_1467  apl_1468  apl_1469  apl_1470  apl_1471  apl_1472  apl_1473  apl_1474  apl_1475  apl_1476  apl_1477  apl_1478  apl_1479  apl_1480  apl_1481  apl_1482  apl_1483  apl_1484  apl_1485  apl_1486  apl_1487  apl_1488  apl_1489  apl_1490  apl_1491  apl_1492  apl_1493  apl_1494  apl_1495  apl_1496  apl_1497  apl_1498  apl_1499  apl_1500  apl_1501  apl_1502  apl_1503  apl_1504  apl_1505  apl_1506  apl_1507  apl_1508  apl_1509  apl_1510  apl_1511  apl_1512  apl_1513  apl_1514  apl_1515  apl_1516  apl_1517  apl_1518  apl_1519  apl_1520  apl_1521  apl_1522  apl_1523  apl_1524  apl_1525  apl_1526  apl_1527  apl_1528  apl_1529  apl_1530  apl_1531  apl_1532  apl_1533  apl_1534  apl_1535  apl_1536  apl_1537  apl_1538  apl_1539  apl_1540  apl_1541  apl_1542  apl_1543  apl_1544  apl_1545  apl_1546  apl_1547  apl_1548  apl_1549  apl_1550  apl_1551  apl_1552  apl_1553  apl_1554  apl_1555  apl_1556  apl_1557  apl_1558  apl_1559  apl_1560  apl_1561  apl_1562  apl_1563  apl_1564  apl_1565  apl_1566  apl_1567  apl_1568  apl_1569  apl_1570  apl_1571  apl_1572  apl_1573  apl_1574  apl_1575  apl_1576  apl_1577  apl_1578  apl_1579  apl_1580  apl_1581  apl_1582  apl_1583  apl_1584  apl_1585  apl_1586  apl_1587  apl_1588  apl_1589  apl_1590  apl_1591  apl_1592  apl_1593  apl_1594  apl_1595  apl_1596  apl_1597  apl_1598  apl_1599  apl_1600  apl_1601  apl_1602  apl_1603  apl_1604  apl_1605  apl_1606  apl_1607  apl_1608  apl_1609  apl_1610  apl_1611  apl_1612  apl_1613  apl_1614  apl_1615  apl_1616  apl_1617  apl_1618  apl_1619  apl_1620  apl_1621  apl_1622  apl_1623  apl_1624  apl_1625  apl_1626  apl_1627  apl_1628  apl_1629  apl_1630  apl_1631  apl_1632  apl_1633  apl_1634  apl_1635  apl_1636  apl_1637  apl_1638  apl_1639  apl_1640  apl_1641  apl_1642  apl_1643  apl_1644  apl_1645  apl_1646  apl_1647  apl_1648  apl_1649  apl_1650  apl_1651  apl_1652  apl_1653  apl_1654  apl_1655  apl_1656  apl_1657  apl_1658  apl_1659  apl_1660  apl_1661  apl_1662  apl_1663  apl_1664  apl_
```

#### D. PENUTUP

Penelitian dilakukan dengan mengumpulkan data sekunder yang didapatkan dari bps.go.id.

Adapun strategi yang dilakukan untuk meminimalisir kelemahan yang dimiliki adalah dengan membuat sebuah data prediksi angka buta huruf yang diharapkan dapat memberikan gambaran kepada pemerintah mengenai buta huruf di masa mendatang.

Prediksi yang dilakukan dengan mengaplikasikan Algoritma Artificial Neural Network dengan memanfaatkan library python dan mesin learning. Pengumpulan data yang akan dijadikan input dan akan diproses adalah data angka buta huruf dalam kurun waktu 2011 – 2019. Prediksi yang dilakukan menggunakan Algoritma Artificial Neural Network menghasilkan output akurasi sebesar 69 %.

Disarankan agar penelitian kedepannya menggunakan metode lain agar bisa dibandingkan dengan metode Algoritma Artificial Neural Network sehingga hasil prediksi yang didapat memiliki nilai akurasi yang lebih baik.

#### E. DAFTAR PUSTAKA

- Andiyono, A., Bakti, R. D., & Irwansyah, E. (2013). Analisis Faktor yang Mempengaruhi Angka Buta Huruf Melalui Geographically Weighted Regression: Studi Kasus Propinsi Jawa Timur. *ComTech: Computer, Mathematics and Engineering Applications*, 4(1), 443. <https://doi.org/10.21512/comtech.v4i1.2788>
- Bukhsh, Z. A., & Stipanovic, I. (2020). Predictive Maintenance for Infrastructure Asset Management. *IT Professional*, 22(5), 40–45. <https://doi.org/10.1109/MITP.2020.2975736>
- Kurniawan, A. S. (2018). Implementasi Metode Artificial Neural Network Dalam Memprediksi Hasil Ujian Kompetensi Kebidanan (Studi Kasus Di Akademi Kebidanan Dehasen Bengkulu). *Pseudocode*, 5(1), 37–44. <https://doi.org/10.33369/pseudocode.5.1.37-44>
- Rifai, B. (2013). Algoritma Neural Network Untuk Prediksi. *Techno Nusa Mandiri*, IX(1), 1–9.
- Sabbeh, S. F. (2018). Machine-learning techniques for customer retention: A comparative study. *International Journal of Advanced Computer Science and Applications*, 9(2), 273–281. <https://doi.org/10.14569/IJACSA.2018.090238>
- Sekretariat Negara. (1945). *Undang-Undang Dasar 1945* (Undang-Undang Dasar 1945).
- Septiana, T., Puspita, N., Fikih, M. Al, & Setyawan, N. (2020). Face Mask Detection Covid-19 Using Convolutional Neural Network ( Cnn ). *Seminar Nasional Teknologi Dan Rekayasa (SENTRA) 2020*, 3, 27–32.
- Simbolon, I. A. R., Yatussa'ada, F., & Wanto, A. (2018). Penerapan Algoritma Blackpropagation Dalam Memprediksi Persentase Penduduk Buta Huruf di Indonesia. *Jurnal Informatika UPGRIS*, 4(2), 163. <https://doi.org/10.21512/comtech.v4i1.2788>
- Sucahyo, Nur; Kurniati, Ike; Hidayatullah, S. (2020). Pendekatan Data Science Dengan Algoritma Neural Network Dalam Penyusunan Strategi Pemberantasan Buta Aksara dan Berhitung di Negara-Negara ASEAN. *Jurnal Teknologi Informatika Dan Komputer MH Thamrin2*, 6(2), 1–11. <https://doi.org/https://doi.org/10.37012/jtik.v6i2.293>