

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

Ike Kurniati¹⁾, Heru Winarno²⁾, Dita Yuliyanti³⁾

^{1,2}Prodi Teknik Informatika, Fakultas Teknologi, ITB Swadharma Jakarta

³Prodi 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

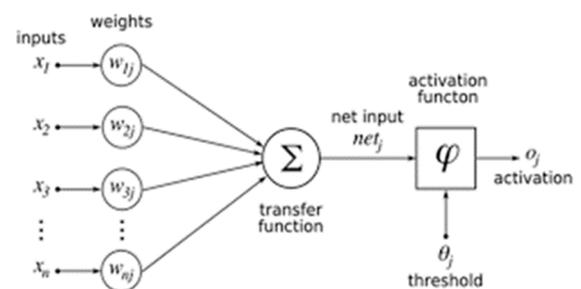
Metode penelitian yang digunakan adalah sebagai berikut: Observasi untuk mengumpulkan data-data yang berhubungan dengan faktor-faktor yang mempengaruhi angka buta huruf di Indonesia yang bersumber dari bps.go.id.

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.



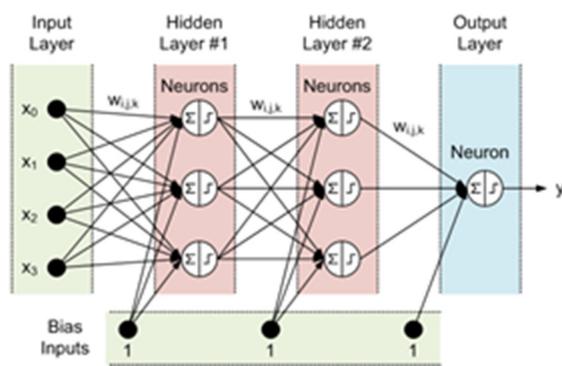
Gambar 1. *Artificial Neural Network*

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*.

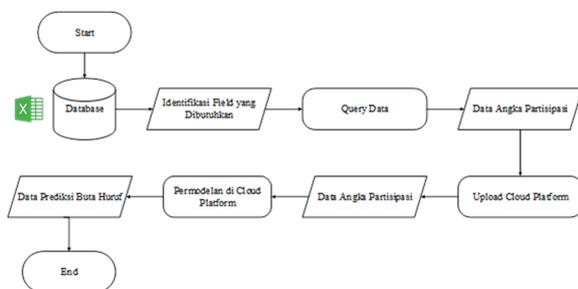
Output layer: lapisan terakhir dari neuron yang menghasilkan output system.



Gambar 2. Arsitektur *Neural Network*

Analisa Kebutuhan Informasi

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*. yang kemudian di *extract* dari sumber data dengan format *excel*. Mekanisme pengumpulan / penarikan data digambarkan dalam *flowchart* dibawah ini:



Gambar 3. *Flowchart* Pengumpulan Data

Dari penarikan data yang telah dilakukan, diketahui total keseluruhan data yang terekam selama 2011 – 2019 adalah sebanyak 315 data. Data yang sudah dikumpulkan tersebut kemudian dikelompokkan kembali menjadi dua jenis yaitu *data training* dan *data testing*. Data yang akan digunakan untuk melatih algoritma atau dijadikan sebagai *data training* adalah data dalam kurun waktu 2011 – 2014, dengan total data sebanyak 140 data

dan terdiri dari 15 *field* seperti pada gambar berikut :

```
[2] path_data_train="/content/drive/MyDrive/TA_dita_yuliyanti/data/data_training.xlsx"
```

```
load_data_train=pd.read_excel(path_data_train)
```

```
load_data_train
```

	Provinsi	tahun	apk_07	apk_10p	apk_10a	apk_10d	apk_10mp	apk_10s	apk_11	apk_11a	apk_11d	apk_11s	apk_12	apk_12a	apk_12d	apk_12s	betahuruf_11s	betahuruf_11a	betahuruf_11d	betahuruf_11s
0	ACEH	2011	100.67	97.15	79.29	92.51	74.87	61.37	90.99	94.37	72.14	27.68	4.37	1.29	12.55					
1	ACEH	2012	108.69	96.47	77.42	94.66	78.61	61.62	99.36	94.34	74.59	28.55	3.96	1.03	11.74					
2	ACEH	2013	110.71	94.39	79.09	97.09	82.57	63.43	99.66	95.23	74.70	29.16	3.34	0.67	10.49					
3	ACEH	2014	111.66	95.87	81.53	97.80	83.20	69.20	99.84	97.38	80.89	32.93	2.38	0.43	8.91					
4	SUMATERA UTARA	2011	104.76	89.19	78.97	91.61	68.08	57.48	98.34	89.00	67.10	16.94	3.22	1.64	6.80					

Gambar 4. Data Training

Keterangan Code :

- Code `path_data_train` digunakan untuk mentautkan file data train yang berada di dalam google drive agar terhubung dengan google colab guna untuk memproses data.
- Code `load_data_train` digunakan untuk membaca dan menampilkan data train yang telah ditautkan sebelumnya.

Dari gambar data diatas terdapat beberapa nama *field* berupa singkatan kata sebagai berikut:

Apk atau Angka Partisipasi Kasar adalah proporsi anak sekolah pada suatu jenjang pendidikan tertentu dalam kelompok umur yang sesuai dengan jenjang pendidikan tersebut. Semakin tinggi APK berarti semakin banyak anak usia sekolah yang bersekolah di suatu jenjang pendidikan pada suatu wilayah.

Apm atau Angka Partisipasi Murni adalah proporsi anak sekolah pada suatu kelompok tertentu yang bersekolah pada tingkat yang sesuai dengan kelompok umurnya. APM selalu lebih rendah dibanding APK karena pembilangnya lebih kecil sementara penyebutnya sama.

Aps atau Angka Partisipasi Sekolah adalah perbandingan antara jumlah murid kelompok usia sekolah tertentu yang bersekolah pada berbagai jenjang pendidikan dengan penduduk kelompok usia sekolah yang sesuai dan dinyatakan dalam persentase. Makin

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	61.27	88.39
1	ACEH	10.08	96.47	77.02	94.08	75.01	61.02	86.36
2	ACEH	10.71	94.39	75.09	97.09	62.07	62.40	89.66
3	ACEH	11.06	92.07	61.03	97.00	62.03	60.34	97.38
4	INDONESIA	10.71	94.76	89.19	79.07	91.01	69.08	67.46
18	INDONESIA	10.71	92.07	89.00	94.00	91.07	69.00	67.00
19	INDONESIA	10.71	92.07	89.00	94.00	91.07	69.00	67.00
20	INDONESIA	10.71	92.07	89.00	94.00	91.07	69.00	67.00
21	INDONESIA	10.71	92.07	89.00	94.00	91.07	69.00	67.00

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=0.0, 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_warmup=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,
         -2.52690205e-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 agh_01 agh_02 agh_03 agh_04 agh_05 agh_06 agh_07 agh_08 agh_09 agh_10 agh_11 agh_12 agh_13 agh_14 agh_15 agh_16 agh_17 agh_18 agh_19 agh_20 a 1 2
0 0 2015 113.36 97.88 83.33 97.99 85.55 69.82 99.90 97.71 81.43 33.07 146.054096 -1.463888 461.873175
1 34 2015 112.62 95.41 84.14 96.47 78.48 66.69 99.35 96.34 76.23 25.16 110.082323 -0.868191 347.917669
2 32 2015 113.25 90.94 81.42 98.12 76.00 66.90 99.44 95.98 82.53 33.13 102.995188 -0.750827 325.466229
3 26 2015 110.40 94.14 76.16 96.63 78.22 62.60 98.79 94.48 75.07 24.85 122.811463 -1.078987 388.242498
4 8 2015 111.99 91.78 76.12 97.68 77.94 59.41 99.55 95.06 70.75 22.22 157.603260 -1.655144 498.459938
...
170 20 2019 112.19 90.40 95.29 96.21 74.68 64.23 99.61 97.29 79.65 38.58 115.932008 -0.951316 363.819602
171 21 2019 109.73 86.53 93.17 97.09 76.20 63.95 98.97 96.97 76.41 31.23 131.938496 -1.230132 417.150603
172 25 2019 111.24 88.49 90.41 93.76 69.92 63.15 97.68 96.58 81.49 31.48 119.963411 -0.894411 353.893803
173 24 2019 91.94 78.11 76.33 79.19 57.19 44.32 82.67 80.13 63.50 22.91 124.952441 -1.108115 393.832605
174 7 2019 107.48 90.57 83.98 97.64 79.40 60.84 99.24 95.91 72.36 25.21 159.779866 -1.691172 505.302072
175 rows x 15 columns
```

Gambar 24. hasil akhir prediksi

Terakhir, dilakukan import data hasil prediksi dari google colab ke google drive dalam bentuk excel agar lebih mudah membaca data selanjutnya,

```
!export as download if in colab
!rm google.colab import files

xlsx = df_result_predict.to_excel('content/drive/MyDrive/TA_dita_vullyanti/data/hasil_prediksi_batahuvf.xlsx', sheet_name='Sheet1')
```

Gambar 25. Import Hasil Prediksi

Analisis kelayakan sistem dilakukan untuk menguji apakah usulan sistem / proyek akan dapat memberikan keuntungan dan dapat dilanjutkan. Uji kelayakan dilakukan dengan mengamati tiga aspek meliputi aspek teknologi, operasional, dan hukum.

Kelayakan Teknologi dibutuhkan beberapa faktor pendukung yaitu perangkat keras dalam pembuatan prediksi adalah komputer PC dengan spesifikasi Core i3, RAM 8 GB, Jaringan Komputer. Sedangkan perangkat lunak yang dibutuhkan dalam pembuatan prediksi adalah sebagai berikut: Sistem operasi *Windows 10*, Web browser *Google Chrome*, *Google Platform* sebagai media penyimpanan data, *Google Colab* sebagai media pemrosesan data,

Kelayakan operasional dinilai dari segi penggunaan layak karena hanya memerlukan alat-alat yang standar seperti *browser*, akun *gmail*, dan *google colab* yang merupakan perangkat yang sudah akrab digunakan oleh banyak orang sehingga pengoperasian dapat dilakukan dengan baik.

Kelayakan hukum dari Prediksi ini dibuat dengan memanfaatkan *environment google cloud platform* yang disediakan secara gratis, dan code yang diinput tidak mengandung unsur-unsur yang dapat memabahayakan perangkat keras maupun perangkat lunak pengguna, data prediksi yang dibuat juga dinilai tidak melanggar hukum yang berlaku. Sehingga dapat disimpulkan proyek pembuatan data prediksi ini layak secara hukum.

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., Bekti, 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>