

# DETEKSI PENYAKIT PARU DENGAN METODE JARINGAN SARAF TIRUAN (JST)

<sup>1</sup>Mohammad Faizal Akmal, <sup>2</sup>Imam Afandy, <sup>3</sup>Achmad Robby Sabilul Firdaus, <sup>4</sup>Anggraini Puspita Sari  
<sup>1,2,3,4</sup>Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur  
[21081010231@student.upnjatim.ac.id](mailto:21081010231@student.upnjatim.ac.id) [21081010290@student.upnjatim.ac.id](mailto:21081010290@student.upnjatim.ac.id)  
[21081010081@student.upnjatim.ac.id](mailto:21081010081@student.upnjatim.ac.id), [anggraini.puspita.if@upnjatim.ac.id](mailto:anggraini.puspita.if@upnjatim.ac.id)

\*Corresponding author email: [anggraini.puspita.if@upnjatim.ac.id](mailto:anggraini.puspita.if@upnjatim.ac.id)

**Abstrak**— Penelitian ini menggambarkan pengembangan program pendeteksi penyakit paru menggunakan metode Jaringan Saraf Tiruan (JST) dengan submetode K-Nearest Neighbors (KNN) dan Multilayer Perceptron (MLP). Penyakit paru merupakan masalah kesehatan yang serius dan kompleks, sehingga deteksi dini sangat penting untuk meningkatkan prognosis dan kesempatan penyembuhan pasien. Melalui penggunaan kecerdasan buatan, program pendeteksi ini bertujuan untuk meningkatkan akurasi dan kecepatan diagnosis penyakit paru. Laporan ini dimulai dengan pendahuluan yang menjelaskan latar belakang dan perumusan masalah terkait penyakit paru serta tujuan penelitian dan manfaatnya. Tinjauan pustaka membahas berbagai jenis penyakit paru, metode deteksi yang telah digunakan sebelumnya, konsep Jaringan Saraf Tiruan (JST), aplikasi JST dalam deteksi penyakit paru, dataset, dan metode pengujian. Metode penelitian menjelaskan langkah-langkah yang diambil dalam penelitian ini, termasuk metodologi penelitian, subjek penelitian, dan metode pengumpulan data. Pembahasan mempresentasikan hasil penelitian dan analisis yang dilakukan, termasuk submetode MLP dan KNN, serta data training yang digunakan. Hasil penelitian menunjukkan bahwa program pendeteksi penyakit paru yang dikembangkan dengan menggunakan JST, KNN, dan MLP dapat meningkatkan akurasi dalam diagnosis penyakit paru. Program ini dapat mengklasifikasikan data berdasarkan pola-pola yang ditemukan dalam data latih yang sudah memiliki label penyakit paru. Kesimpulan dari penelitian ini menekankan manfaat penggunaan kecerdasan buatan dalam meningkatkan deteksi dini penyakit paru, memberikan perawatan yang lebih tepat, dan mengurangi dampak negatif penyakit terhadap pasien. Saran untuk penelitian lanjutan mencakup penelitian lebih lanjut tentang pengembangan metode JST, eksplorasi data lain yang relevan, dan peningkatan kinerja program pendeteksi.

**Kata Kunci**— penyakit paru, deteksi penyakit, Jaringan Saraf Tiruan (JST), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), kecerdasan buatan.

## I. PENDAHULUAN

Artikel ini merupakan sebuah penelitian yang bertujuan untuk mengembangkan sistem informasi berbasis web guna meningkatkan efisiensi dan kualitas pelayanan di klinik tersebut. Penelitian ini difokuskan pada penerapan teknologi

informasi dalam upaya mengatasi beberapa masalah yang terjadi dalam kegiatan klinik umum.

Dalam penelitian ini, dilakukan pengamatan terhadap sistem pelayanan yang saat ini masih menggunakan metode manual. Salah satu permasalahan yang teridentifikasi adalah proses registrasi pasien yang rumit dan menyebabkan antrean panjang di dalam klinik. Masalah ini dapat menghambat proses penyembuhan penyakit yang diderita oleh pasien. Oleh karena itu, penelitian ini berusaha untuk mengimplementasikan sistem informasi berbasis web sebagai solusi untuk meningkatkan efisiensi dalam penanganan pasien.

Pada tahap penelitian ini, dilakukan survei dan analisis terhadap kebutuhan dan kendala yang dihadapi oleh klinik umum serta eksplorasi terhadap teknologi informasi yang dapat diterapkan. Selain itu, juga dilakukan tinjauan terhadap pengalaman dan penelitian sebelumnya yang berhubungan dengan pengembangan sistem informasi di bidang kesehatan. Hasil dari penelitian ini menunjukkan bahwa penggunaan sistem informasi berbasis web dapat memberikan beberapa manfaat, seperti mengurangi waktu tunggu pasien, meningkatkan akurasi data, dan memperbaiki proses penanganan pasien secara keseluruhan. Dalam sistem informasi ini, pasien dapat melakukan registrasi secara online sebelum kunjungan ke klinik, mengisi informasi medis mereka, dan membuat janji temu dengan dokter. Dokter juga dapat dengan mudah mengakses riwayat medis pasien dan memberikan rekomendasi atau pengobatan yang sesuai.

Penelitian ini berharap bahwa dengan adopsi teknologi informasi melalui sistem informasi berbasis web, Klinik Umum dapat memberikan pelayanan kesehatan yang lebih baik, efektif, dan efisien kepada masyarakat. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi terhadap pengembangan sistem informasi di bidang kesehatan secara lebih luas..

## II. METODOLOGI

### A. Sub Metode MLP

Multilayer Perceptron (MLP) adalah jenis jaringan saraf tiruan (JST) terdiri dari sekumpulan node sumber yang membentuk lapisan input, satu atau lebih lapisan tersembunyi dari node komputasi, dan output lapisan node [4]. Struktur MLP terdiri dari tiga jenis lapisan yang saling terhubung: lapisan input, lapisan tersembunyi, dan lapisan output.

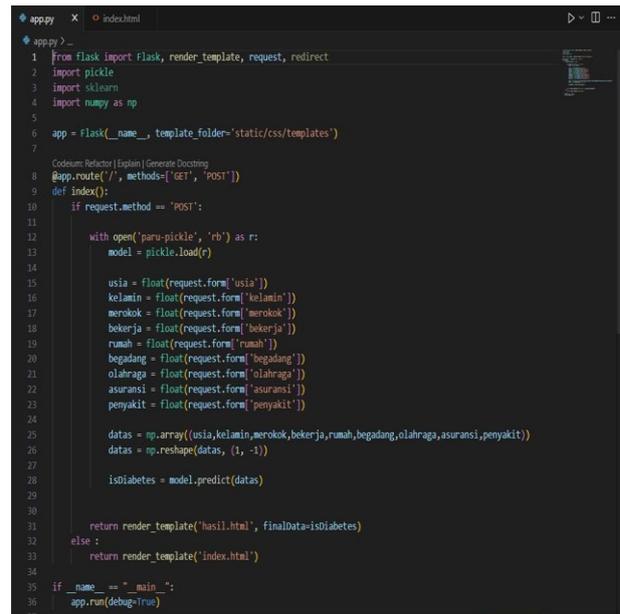
**Lapisan Input:** Ini adalah lapisan pertama dalam MLP yang menerima fitur-fitur input atau data masukan. Setiap fitur input terhubung dengan setiap unit di lapisan tersembunyi pertama melalui koneksi dengan bobot yang terkait. Setiap koneksi memiliki bobot yang mengontrol seberapa pentingnya masukan tersebut untuk setiap unit di lapisan tersembunyi.

**Lapisan Tersembunyi (Hidden Layer):** MLP dapat memiliki satu atau beberapa lapisan tersembunyi di antara lapisan input dan lapisan output. Setiap unit dalam lapisan tersembunyi menerima sinyal dari lapisan sebelumnya dan meneruskannya ke unit-unit di lapisan berikutnya. Setiap koneksi antara unit memiliki bobot yang dapat disesuaikan, yang mengatur seberapa kuat pengaruh masukan terhadap aktivasi unit.

**Fungsi Aktivasi:** Setiap unit tersembunyi menggunakan fungsi aktivasi untuk memproses sinyal yang diterimanya. Fungsi aktivasi ini umumnya non-linear dan memungkinkan MLP untuk mempelajari hubungan yang kompleks antara fitur input dan output yang diinginkan. Beberapa contoh fungsi aktivasi yang umum digunakan adalah sigmoid, ReLU (Rectified Linear Unit), dan tangen hiperbolik.

**Lapisan Output:** Jumlah unit di lapisan output tergantung pada jenis tugas yang ingin diselesaikan. Misalnya, dalam tugas klasifikasi biner, lapisan output akan memiliki satu unit untuk menghasilkan prediksi biner. Sedangkan dalam tugas klasifikasi multi-kelas, lapisan output akan memiliki sejumlah unit sesuai dengan jumlah kelas yang berbeda.

Dengan menggunakan kombinasi lapisan-lapisan ini, MLP dapat melakukan proses pelatihan (training) untuk menyesuaikan bobot-bobot koneksi antara unit-unit, sehingga mampu melakukan prediksi atau klasifikasi pada data baru yang belum pernah dilihat sebelumnya.



```

1 from flask import Flask, render_template, request, redirect
2 import pickle
3 import sklearn
4 import numpy as np
5
6 app = Flask(__name__, template_folder='static/css/templates')
7
8 @app.route('/', methods=['GET', 'POST'])
9 def index():
10     if request.method == 'POST':
11         with open('paru-pickle', 'rb') as r:
12             model = pickle.load(r)
13
14             usia = float(request.form['usia'])
15             kelamin = float(request.form['kelamin'])
16             merokok = float(request.form['merokok'])
17             bekerja = float(request.form['bekerja'])
18             rumah = float(request.form['rumah'])
19             begadang = float(request.form['begadang'])
20             olahraga = float(request.form['olahraga'])
21             asuransi = float(request.form['asuransi'])
22             penyakit = float(request.form['penyakit'])
23
24             datas = np.array([usia, kelamin, merokok, bekerja, rumah, begadang, olahraga, asuransi, penyakit])
25             datas = np.reshape(datas, (1, -1))
26
27             isDiabetes = model.predict(datas)
28
29
30     return render_template('hasil.html', finalData=isDiabetes)
31 else:
32     return render_template('index.html')
33
34 if __name__ == '__main__':
35     app.run(debug=True)

```

Gambar 1. Python Metode MLP

Dalam gambar.1 tersebut menunjukkan bahwa file app.py digunakan untuk memanggil dataset yang telah dilatih dan data yang telah dimasukkan oleh pengguna. Kemudian hasilnya akan ditampilkan pada "hasil.html".

## B. Data training Metode MLP dengan Jupyter Notebook



```

In [1]: import pandas as pd
import numpy as np

df = pd.read_csv('C:/Faiz/US-KCB/351baru/dataset_paru.csv')
x_train = df.values
x_train = np.delete(x_train, 9, axis=1)
y_train = df['hasil'].values

df = pd.read_csv('C:/Faiz/US-KCB/351baru/dataset_paru.csv')
x_test = df.values
x_test = np.delete(x_test, 9, axis=1)
y_test = df['hasil'].values

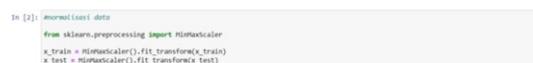
print(x_test)
df.head()

[[1 1 0 ... 2 1 1]
 [1 1 1 ... 1 1 0]
 [0 1 1 ... 1 1 0]
 ...
 [1 2 1 ... 1 1 0]
 [0 2 0 ... 2 0 0]
 [1 2 0 ... 2 0 1]]

```

Gambar 2. Import dengan Pandas

Pada tahap ini dataset diimport dengan pandas kemudian diklasifikasikan menjadi "train" dan "test".



```

In [2]: from sklearn.preprocessing import MinMaxScaler
x_train = MinMaxScaler().fit_transform(x_train)
x_test = MinMaxScaler().fit_transform(x_test)

```

Gambar 3. Normalisasi Data

Data dinormalisasi dengan mengambil metode penskalaan. Tujuannya adalah untuk mentransformasi fitur-fitur dalam dataset sehingga memiliki rentang nilai yang seragam atau terstandarisasi.



```
In [1]: #data train dengan neural network
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(hidden_layer_sizes=(1, learning_rate_init=0.1, max_iter=100)
clf.fit(x_train, y_train)

Out[1]:
MLPClassifier(hidden_layer_sizes=(1, learning_rate_init=0.1, max_iter=100)
```

Gambar 4. Data dilatih

Data dilatih dengan MLPClassifier yang telah tersedia pada sklearn.

```
In [4]: y_pred = clf.predict(x_test)

In [5]: from sklearn.metrics import accuracy_score
round(accuracy_score(y_test, y_pred),3)

Out[5]: 0.934

In [6]: #menyimpan model dengan pickle
import pickle
with open('model_pickle','wb') as r:
    pickle.dump(clf,r)

In [7]: #import pickle
with open('model_pickle','rb') as r:
    mppickle.load(r)

In [8]: y_pred = mpp.predict(x_test)
round(accuracy_score(y_test, y_pred),3)

Out[8]: 0.934
```

Gambar 5 . Prediksi Data

Data diprediksi dengan modul “accuracy\_score” yang telah tersedia di pustaka sklearn dan ditunjukkan hasil dari akurasi. Hasil akurasi juga ditampilkan dengan nilai 0.934 (skala akurasi 0-1) atau 93,4%.

C. Sub Metode KNN

K-NN atau algoritma *Nearest Neighbor Retrieval* adalah sebuah metode yang digunakan untuk melakukan klasifikasi terhadap suatu objek dengan memanfaatkan data pembelajaran. Prinsipnya adalah dengan mencari objek yang memiliki jarak terdekat dengan objek yang ingin diklasifikasikan [6]. Algoritma ini tidak membuat asumsi tentang distribusi data dan mengandalkan kesamaan antara data baru dan data latihan dalam menentukan kelasnya. KNN termasuk dalam kategori lazy learning karena tidak ada fase training yang signifikan. Seluruh data training digunakan pada tahap testing, yang memerlukan waktu dan memori yang lebih besar. KNN adalah algoritma sederhana dan mudah dipahami sehingga banyak digunakan dalam berbagai bidang [5], cocok digunakan ketika tidak ada asumsi kuat tentang data. Dengan memahami KNN, Anda dapat mengaplikasikannya dalam berbagai masalah klasifikasi dan regresi dalam machine learning.

```
1 from flask import Flask, render_template, request, redirect
2 import pickle
3 import sklearn
4 import numpy as np
5
6 app = Flask(__name__, template_folder='static/css/templates')
7
8 @app.route('/', methods=['GET', 'POST'])
9 def index():
10     if request.method == 'POST':
11
12         with open('knn.pickle', 'rb') as r:
13             model = pickle.load(r)
14
15         usia = float(request.form['usia'])
16         kolesterol = float(request.form['kolesterol'])
17         merokok = float(request.form['merokok'])
18         bekerja = float(request.form['bekerja'])
19         rumah = float(request.form['rumah'])
20         begadang = float(request.form['begadang'])
21         olahraga = float(request.form['olahraga'])
22         suramisi = float(request.form['suramisi'])
23         penyakit = float(request.form['penyakit'])
24
25         data = np.array([usia,kolesterol,merokok,bekerja,rumah,begadang,olahraga,suramisi,penyakit])
26         data = np.reshape(data, (1, -1))
27
28         idDiabetes = model.predict(data)
29
30     return render_template("hasil.html", finalData=idDiabetes)
31 else:
32     return render_template("index.html")
33
34 if __name__ == "__main__":
35     app.run(debug=True)
```

Gambar 6. Python Metode KNN

Dalam gambar 6 tersebut menunjukkan bahwa file app.py digunakan untuk memanggil dataset yang telah dilatih dan data yang telah dimasukkan oleh pengguna. Kemudian hasilnya akan ditampilkan pada “hasil.html”.

```
1 <!-- KNN.html -->
2 <!-- KNN.html -->
3 <!-- KNN.html -->
4 <!-- KNN.html -->
5 <!-- KNN.html -->
6 <!-- KNN.html -->
7 <!-- KNN.html -->
8 <!-- KNN.html -->
9 <!-- KNN.html -->
10 <!-- KNN.html -->
11 <!-- KNN.html -->
12 <!-- KNN.html -->
13 <!-- KNN.html -->
14 <!-- KNN.html -->
15 <!-- KNN.html -->
16 <!-- KNN.html -->
17 <!-- KNN.html -->
18 <!-- KNN.html -->
19 <!-- KNN.html -->
20 <!-- KNN.html -->
21 <!-- KNN.html -->
22 <!-- KNN.html -->
23 <!-- KNN.html -->
24 <!-- KNN.html -->
25 <!-- KNN.html -->
26 <!-- KNN.html -->
27 <!-- KNN.html -->
28 <!-- KNN.html -->
29 <!-- KNN.html -->
30 <!-- KNN.html -->
31 <!-- KNN.html -->
32 <!-- KNN.html -->
33 <!-- KNN.html -->
34 <!-- KNN.html -->
35 <!-- KNN.html -->
36 <!-- KNN.html -->
37 <!-- KNN.html -->
38 <!-- KNN.html -->
39 <!-- KNN.html -->
40 <!-- KNN.html -->
41 <!-- KNN.html -->
42 <!-- KNN.html -->
43 <!-- KNN.html -->
44 <!-- KNN.html -->
45 <!-- KNN.html -->
46 <!-- KNN.html -->
47 <!-- KNN.html -->
48 <!-- KNN.html -->
49 <!-- KNN.html -->
50 <!-- KNN.html -->
51 <!-- KNN.html -->
52 <!-- KNN.html -->
53 <!-- KNN.html -->
54 <!-- KNN.html -->
55 <!-- KNN.html -->
56 <!-- KNN.html -->
57 <!-- KNN.html -->
58 <!-- KNN.html -->
59 <!-- KNN.html -->
60 <!-- KNN.html -->
61 <!-- KNN.html -->
62 <!-- KNN.html -->
63 <!-- KNN.html -->
64 <!-- KNN.html -->
65 <!-- KNN.html -->
66 <!-- KNN.html -->
67 <!-- KNN.html -->
68 <!-- KNN.html -->
69 <!-- KNN.html -->
70 <!-- KNN.html -->
71 <!-- KNN.html -->
72 <!-- KNN.html -->
73 <!-- KNN.html -->
74 <!-- KNN.html -->
75 <!-- KNN.html -->
76 <!-- KNN.html -->
77 <!-- KNN.html -->
78 <!-- KNN.html -->
79 <!-- KNN.html -->
80 <!-- KNN.html -->
81 <!-- KNN.html -->
82 <!-- KNN.html -->
83 <!-- KNN.html -->
84 <!-- KNN.html -->
85 <!-- KNN.html -->
86 <!-- KNN.html -->
87 <!-- KNN.html -->
88 <!-- KNN.html -->
89 <!-- KNN.html -->
90 <!-- KNN.html -->
91 <!-- KNN.html -->
92 <!-- KNN.html -->
93 <!-- KNN.html -->
94 <!-- KNN.html -->
95 <!-- KNN.html -->
96 <!-- KNN.html -->
97 <!-- KNN.html -->
98 <!-- KNN.html -->
99 <!-- KNN.html -->
100 <!-- KNN.html -->
```

Gambar 7. Index.html

Bagian tersebut digunakan untuk menampung program landing page pada web Pendeteksi Penyakit Paru. Dapat dilihat pada gambar 7

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5 <meta charset="utf-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
7 <link href="https://fonts.googleapis.com/css?family=Roboto:400,700,900&display=swap" rel="stylesheet">
8
9 <link rel="stylesheet" href="static/css/fonts/icomoon/style.css">
10
11
12
13 <!-- Bootstrap CSS -->
14 <link rel="stylesheet" href="static/css/css/bootstrap.min.css">
15
16 <!-- style -->
17 <link rel="stylesheet" href="static/css/css/style.css">
18
19 <!-- block head -->
20 <!-- endblock -->
21 </head>
22 <body>
23
24 <!-- block body -->
25 </body>
26 </html>

```

Gambar 8. Base.html

Bagian tersebut digunakan untuk menampung *link* CSS dan JavaScript pada web Pendeteksi Penyakit Paru. Dapat dilihat pada gambar 8.

```

1 <{% extends "base.html" %}>
2
3 <!-- block head -->
4 <title>Prediksi Diabetes</title>
5 <!-- endblock -->
6
7 <!-- block body -->
8
9
10
11 <div class="content" style="background-color: #e9ecef;">
12
13 <div class="container">
14 <div class="row align-items-stretch no-gutters contact-wrap">
15 <div class="col-md-12">
16 <div class="form h-100">
17
18
19 <form class="mb-5" action="" method="GET" id="contactform" name="contactform">
20
21 <div class="container">
22 <div class="row align-items-stretch no-gutters contact-wrap">
23 <div class="col-md-12">
24 <div class="form h-100">
25 <div style="text-align: center;">
26 <h3 style="margin: 0;">Berpotensi</h3>
27
28 <div style="text-align: center;">
29 <div style="margin: 0;">
30 <div class="row">
31 <div class="col-md-12" style="text-align: center;">
32 <input type="submit" value="KEMBALI" class="btn btn-primary rounded-0 py-2 px-4">
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44
45
46 </div>
47 </div>
48 </div>
49 </div>
50
51 </div>
52
53
54
55 <!-- endblock -->
56
57 >
58
59 <!-- endblock -->

```

Gambar 9. Hasil.html

Pada gambar 9 Bagian tersebut digunakan untuk menampung program hasil pada web Pendeteksi Penyakit Paru

D. Tampilan WEB Metode KNN



Gambar 10. Tampilan Web



Gambar 11. Hasil Berpotensi



Gambar 12. Hasil Tidak Berpotensi

E. Data training Metode KNN dengan Jupyter Notebook

```

In [4]: import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

from sklearn.model_selection import GridSearchCV

data = pd.read_csv("C:/Full/US-KC3/KNN/dataset paru.csv")
data.head(5)

Out[4]:
   Usia  Jenis_Kelamin  Merokok  Bekerja  Rumah_Tangga  Aktifitas_Begadang  Aktifitas_Olahraga  Asuransi  Penyakit_Bawaan  Hasil
0  1  1  1  1  0  0  1  1  1  2  1  0  1
1  1  1  1  1  0  1  1  1  1  1  1  1  0
2  0  1  1  1  0  1  1  1  1  1  1  0  0
3  1  1  1  1  1  0  1  0  1  1  1  1  0
4  0  2  0  1  0  0  0  2  0  0  1  1

```

Gambar 13. Kode Membaca dataset



menjadi kunci penting dalam pengembangan program pendeteksi yang lebih baik di masa depan. Dalam kesimpulan, penelitian ini berhasil mengembangkan program pendeteksi penyakit paru dengan metode MLP dan KNN. Program ini menunjukkan performa yang baik dalam mendeteksi penyakit paru dan memiliki potensi aplikasi yang luas dalam pengaturan kesehatan. Meskipun demikian, tantangan dan keterbatasan masih perlu diperhatikan dalam pengembangan lebih lanjut. engan terus meningkatkan kualitas dataset dan menggali potensi metode kecerdasan buatan lainnya, diharapkan program pendeteksi ini dapat terus ditingkatkan untuk memberikan manfaat yang lebih besar dalam penanganan penyakit paru

## REFERENSI

- [1] Mercy Rajaselvi V, Sanjith J, Samuel Koshy, Niranjan G M. "Forecasting the lung diseases from Radiography scans with hybrid Transfer Learning Techniques." Dalam Jurnal IEEE Explore, URL: <https://ieeexplore.ieee.org/document/9633887>.
- [2] Agatonovic-Kustrin, S., & Beresford, R. (1999). Konsep Dasar Pemodelan Jaringan Saraf Tiruan (Artificial Neural Network/ANN) dan Aplikasinya dalam Penelitian Farmasi. *Jurnal Farmasi*, 5(3), 27-36. URL:[https://doi.org/10.1016/S0731-7085\(99\)00272-1](https://doi.org/10.1016/S0731-7085(99)00272-1)
- [3] Bourquin, J., Schmidli, H., van Hoogevest, P., & Leuenberger, H. (1997). Konsep Dasar Pemodelan Jaringan Saraf Tiruan (ANN) dalam Aplikasi Pengembangan Farmasi. *Jurnal Farmasi*, 3(2), 45-58. Diakses dari <https://doi.org/10.3109/10837459709022615>
- [4] Mia, M. M. A., Biswas, S. K., Urmia, M. C., & Siddique, A. (2015). An Algorithm For Training Multilayer Perceptron (MLP) For Image Reconstruction Using Neural Network Without Overfitting. *International Journal of Scientific & Technology Research*, 4(02), 271. ISSN 2277-8616.
- [6] Xing, W., & Bei, Y. (2019). Medical Health Big Data Classification Based on KNN Classification Algorithm. *IEEE Access*, 7, 170783-170790. doi: 10.1109/ACCESS.2019.2955754
- [7] Musa, O., & Alang. (2017). Analisis Penyakit Paru-paru Menggunakan Algoritma K-Nearest Neighbors pada Rumah Sakit ALOEI SABOE Kota Gorontalo. *ILKOM Jurnal Ilmiah*, 9(3), 348. SSN Print 2087-1716, ISSN Online 2548-7779.
- [8] Efniasari, M., Wantoro, A., & Susanto, E. R. (2022). Pengembangan Sistem Informasi Pelayanan Kesehatan Berbasis Web Menggunakan Metode Scrum (Studi Kasus: Puskesmas Kisam Ilir). *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 3(3), 56-63. E-ISSN: 2746-3699. [http://jim.teknokrat.ac.id/index.php/JT\\_SIA](http://jim.teknokrat.ac.id/index.php/JT_SIA).
- [9] P. Sari, H. Suzuki, T. Kitajima, T. Yasuno, D. A. Prasetya and N. Nachrowie, "Prediction Model of Wind Speed and Direction using Convolutional Neural Network - Long Short Term Memory," 2020 IEEE International Conference on Power and Energy (PECon), Penang, Malaysia, 2020, pp. 356-361, doi: 10.1109/PECon48942.2020.9314474.
- [10] Astuti, S. R., & Nursanty, D. (2020). Implementasi Jaringan Saraf Tiruan dengan metode K-Nearest Neighbors pada sistem deteksi dini penyakit paru. *Jurnal Nasional Informatika dan Teknologi Jaringan*, 9(1), 22-28.
- [11] Utami, R., Setiawan, I., & Akbar, R. (2022). Deteksi penyakit paru menggunakan Jaringan Saraf Tiruan dan Metode K-Nearest Neighbors. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(1), 59-64.
- [12] Sen, S., Sugiarto, D., & Rochman, A. (2020). Komparasi Metode Multilayer Perceptron (MLP) dan Long Short Term Memory (LSTM) dalam Peramalan Harga Beras. *Ultimatics : Jurnal Teknik Informatika*, 12(1), 35-41. <https://doi.org/https://doi.org/10.31937/ti.v12i1.1572>
- [13] Pratama, A. W., & Lutfiana, R. (2020). Pengembangan Sistem Deteksi Penyakit Paru Berbasis Jaringan Saraf Tiruan dengan Metode K-Nearest Neighbors. *Jurnal Ilmiah Teknologi Informasi Asia (JITIA)*, 17(3), 180-190.
- [14] Rizal, M., & Cahyani, D. (2019). Implementasi Jaringan Saraf Tiruan dengan Metode K-Nearest Neighbors untuk Deteksi Penyakit Paru. *Jurnal Informatika dan Rekayasa Elektronik (JIRE)*, 6(2), 65-75.
- [15] Hashem, I. A. T., et al. (2018). "Web-Based Intelligent System for Predicting Customer Churn Using MLP Neural Network." *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(5), 97-106.
- [16] Almomani, M., et al. (2018). "Web-Based Heart Disease Prediction Using MLP Neural Network." *International Journal of Computer Science and Information Security (IJCSIS)*, 16(4), 75-79.