Classification of Childhood Diseases with Fever Using Fuzzy K-Nearest Neighbor Method

Rizky Karunia Putra
Department of Informatics
Universitas Islam Indonesia
Yogyakarta, Indonesia
14523064@students.uii.ac.id

Sri Mulyati
Department of Informatics
Universitas Islam Indonesia
Yogyakarta, Indonesia
Mulya@uii.ac.id

***Abstract*— Fever or pyrexia is a condition when the body temperature rises above the average. This may occur due to viral or bacterial infection of the body. In addition, fever is the main symptom of diseases such as dengue fever, typhoid fever, diarrhea, gastroenteritis, measles, pneumonia, pharyngitis, and bronchitis. These diseases have similar symptoms, causing difficulty to distinguish them. In fact, the symptoms of diseases are usually recorded in a medical record document.

Medical records can be categorized in order to ease diagnosis. The technique to categorize based on certain characteristics to several classes is called classification. Classification can categorize textual data which are first converted into numerical data so that the classification process can generate results. Fuzzy K-Nearest Neighbor is one classification technique that measures the distance between training and testing data, which then put them into a fuzzy set. This study developed a classification system for childhood diseases with fever using Fuzzy K-Nearest Neighbor based on textual medical record documents.

The test results of the classification system showed an accuracy of 83.3% in the dengue fever and pneumonia data with a comparison of training and testing data of 80 : 20, K value of 10, and M value of 2. Thus, it can be concluded that Fuzzy K-Nearest Neighbor classification system can be used as a solution to the classification of childhood diseases with fever.

***Keywords*— fuzzy k-nearest neighbor, FKNN, classification of childhood diseases with fever, classification, machine learning**

1 INTRODUCTION

Fever is the phenomenon of an increase in the body temperature. This may occur due to viral or bacterial infection of the body [1]. In fact, fever may occur to anyone regardless of age and gender. However, children most commonly suffer from fever due to low immune system.

Fever is the main symptom of various diseases, especially infectious diseases. WHO defines infectious diseases as diseases caused by pathogenic microorganisms such as bacteria, viruses, parasites, and fungi that can spread either directly or indirectly from one person to another person [2]. Many diseases begin with fever, such as dengue fever, typhoid fever, diarrhea, gastroenteritis, measles, pneumonia, pharyngitis, and bronchitis.

Several types of diseases have similar symptoms. This causes a provisional diagnosis process to be difficult [3]. In fact, diagnosis process is very complex that not anyone can do it. There are important procedures that a physician shall undertake to make a diagnosis. A trained physician is able to dig up information from the existing anamnesis to classify the types of diseases that a patient suffers from [4].

Machine learning is one of many technologies that may assist a physician in making diagnosis. This aims to provide prediction and decision-making [5]. There are three kinds of techniques in developing machine learning and one of which is classification. Classification process can be carried out using multiple algorithms, such as Nearest Neighbor, Naïve Bayes, Fuzzy, and SVM.

Fuzzy K-Nearest Neighbor is a branch of Nearest Neighbor. This algorithm is combined Fuzzy Logic and K-Nearest Neighbor, which aims to find the nearest distance among neighbors. Unline K-Nearest Neighbor algorithm, this algorithm performs a fuzzy search by adding membership scores to avoid bias.

The implementation of Fuzzy K-Nearest Neighbor algorithm aims to assist physicians in making a provisional diagnosis based on medical record documents.

2 REVIEW OF SIMILAR RESEARCH

This study used several similar researches for comparison. Below is a list of some researches relevant to the present research, discussing the classification of diseases with fever and Fuzzy K-Nearest Neighbor classification technique.

a. By Sri Mulyati, Sri Kusumadewi, and Linda Rosita entitled "Model Sistem Pendukung Keputusan Untuk Diagnosis Penyakit Anak Dengan Gejala Demam Menggunakan Naive Bayesian Clasification" (Model of Decision Support System for Diagnosis of Childhood Disease with Fever Using Naive Bayesian Classification) [6].

b. By Fakihatin Wafiyah, Nurul Hidayat, Rizal Setya Perdana entitled "Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam" (Implementation of Modified K-Nearest Neighbor (MKNN) Algorithm for Fever Disease Classification)[3].

c. By Satria Dwi Nugraha, Rekyan Regasari Mardi Putri, Randy Cahya Wihandika entitled "Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Dalam Menentukan Status Gizi Balita" (Application of Fuzzy K-Nearest Neighbor (FK-NN) in Determining the Child Nutritional Status) [7].

d. By James M. Keller, Michael R. Gray, James A. Givens, JR entitled "A Fuzzy K-Nearest Neighbor Algorithm"[8].

978-1-5386-7422-2/18/$31.00 ©2018 IEEE

Some of the abovementioned researches can be seen in Table 1 Review of similar research.

Table 1 Review of Similar Research

Research	Development of System	Classification	Diseases with Fever	Childhood Diseases	FK-NN
Mulyati, Kusumadewi, & Rosita, (2012)	-	✓	✓	✓	-
Wafiyah, Hidayat, & Perdana, (2017)	✓	✓	✓	-	-
Nugraha, Putri, & Wihandika, (2017)	-	✓	-	-	✓
Keller, Gray, & Givens, (1985)	-	✓	-	-	✓

The review of the abovementioned researches concluded that there has not been any research that covers five research topics, namely system development, classification, diseases with fever, childhood diseases, and Fuzzy K-Nearest Neighbor. The present research, however, covers all these five topics.

3 METHODOLOGY

The methodology of this study consisted of data collection, system design, implementation, and system testing. The explanation of each stage is presented as follows

3.1 Data Collection

The data collection was conducted by collecting the medical records of patients who have clinically been declared to suffer from diseases with fever. There were eight diseases covered in this research as shown in Table 2. The data were in the form of secondary data obtained from Islamic Hospital Banjarmasin.

Table 2 Diseases

No	Diseases
1	Dengue Fever
2	Typhoid Fever
3	Measles
4	Pharyngitis
5	Bronchitis
6	Pneumonia
7	Diarrhea
8	Gastroenteritis

3.2 Needs Analysis

The needs analysis was conducted several stages, i.e. functionality analysis, input analysis, output analysis, software requirements analysis, and hardware requirements analysis. In the functional needs analysis, a search for functions that can be performed by a system was carried out, so as to answer the problem formulation. Input needs analysis contains data that were used as input. Output needs analysis is the results or the kinds of information that the system provides when used. Output is information to be presented on the user's system. Software analysis provides information about the software to be used for the development of system, while hardware analysis provides answer about the best and most effective hardware in the implementation of the system.

3.3 Design

Design was performed after system analysis was done. This stage resulted in an overview of the system being designed. This overview was obtained through several steps i.e.Use Case diagram, Activity diagram, interface design, and Flowchart

3.4 Methodology

In developing a Classification System for Childhood Diseases with Fever Using Fuzzy K-Nearest Neighbor Method, several methods were employed. These methods were used for preprocessing, classification and determining the accuracy. These methods were Term Frequency times Inverse Document Frequency (TF-IDF), Fuzzy K-Nearest Neighbor, and Confusion Matrix. TF-IDF was used to measure the weight of the medical record documents. Then the value obtained by TF-IDF was used in the next stage, i.e. classification. At this stage, Fuzzy K-Nearest Neighbor was used to create a model based on such TF-IDF values. Fuzzy K-Nearest Neighbor is a classification method by creating distance between documents that have been processed with fuzzy logic.

1. Term Frequency times Inverse Document Frequency (TF-IDF)

The flowchart of converting text to number from TF-IDF can be seen in Figure 1.

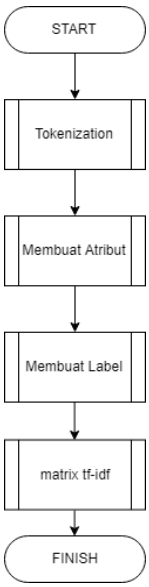

Figure 1 TF-IDF Processes

a. Tokenization

In tokenization, the characters "_" in any documents were separated into terms. This step was performed on all the documents both in the training and testing data [i].

b. Creating Attributes

In this step, attributes from the results of tokenization were created. This step was performed using looping and branching, aimed to create attributes that are free from duplication.

c. Creating Label

In this step, labeling of the classes in the documents of the training data was performed.

d. Creating TF-IDF Matrix

In this step, weighting of the textual documents in both the training and testing data was performed [i]. TF-IDF is composed of two words, i.e. TF and IDF. TF measures the number of words that appear in a document. IDF measures the extend to which a word is important. IDF can be measured using equation(1). After obtaining the values of TF and IDF, then the value of TF-IDF can be obtained using equation(2).

$$IDF(t) = \log(total\ dokumen / total\ kata\ t\ pada\ semua\ dokumen) \quad (1)$$

$$TFIDF(t) = TF(t) . IDF(t) \quad (2)$$

2. Fuzzy K-Nearest Neighbor

The algorithm performs classification based on the distance between the data that will be evaluated using K neighbors in training data. There are various methods to calculate the distance between data, one of which is Cosine Similarity. Cosine Similarity is highly suitable for textual data. The calculation using Cosine Similarity may use the formula in equation(3).

$$\cos(\Theta_{ij}) = \frac{\sum_k (d_{ik} d_{jk})}{\sqrt{\sum_k d_{ik}^2} \sqrt{\sum_k d_{jk}^2}} \quad (3)$$

d_{ik} = weight of term in i-th data
d_{jk} = weight of term in j-th data

Cosine Similarity requires the value of weight per term in a text on the data. Cosine Similarity can be combined with TF-IDF to obtain weight. TF-IDF is a combination of Term Frequency (TF) and Inverse Document Frequency (IDF) to provide weight of term by considering how frequent the term appears in a document.

The algorithm of Fuzzy K-Nearest Neighbor is as follows [8].

a. Determine parameter K
b. Measure the distance between the data that will be evaluated in all the training data.
c. Sort the distances (ascending) and determine the nearest distance until K-th sequence.
d. Calculate Ui (membership score of all classes on x) using the formula in equation.

$$U_i(x) = \frac{\sum_{j=1}^{k} U_{ij} \left(1/\|x - x_j\|^{2/(m-1)}\right)}{\sum_{j=1}^{k} \left(1/\|x - x_j\|^{2/(m-1)}\right)} \quad (4)$$

U_{ij} = Fuzzy membership score
k = Nearest neighbor score
j = Variable of membership data in testing data
m = Fuzzy strength of m > 1

To calculate the membership score in Fuzzy K-NN, the membership score was firstly calculated using the following equation [7].

$$u_{ij} = \begin{cases} 0.51 + \left(\frac{n_j}{n}\right) * 0.49, & jika\ j = 1 \\ \left(\frac{n_j}{n}\right) * 0.49, & jika\ j \neq 1 \end{cases} \quad (5)$$

n_j = the number of neighbors in j-th class in training data n
n = the number of training data
j = data class

3. Confusion Matrix

Confusion Matrix is a table used to describe the performance of a classification model[9]. Confusion Matrix table consists of N rows and columns. Confusion Matrix contains correct and incorrect predicted values to be compared with the actual testing data.

Confusion Matrix is highly useful for measuring the level of Recall, Precision and Accuracy. Precision can be calculated using equation(6). Recall can be calculated using equation(7). Accuracy can be calculated using equation(8).

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$Akurasi = \frac{TP + FN}{P + N} \quad (8)$$

4 Result and Implementation

Implementation is the next stage of the design planned in the previous stage. This stage showed whether the design could run smoothly as planned. In implementation, several planned stages were performed, namely: data retrieval, voice tag, changing file type, taking MFCC features, training, testing and data display.

1. Data Collection

The data collection was carried out at Islamic Hospital Banjarmasin, resulting in eight types of diseases

recommended by this hospital. There were 106 data of medical record documents collected from Indonesian Islamic Hospital, divided into eight categories of diseases as follows.

1. 9 documents for Bronchitis.
2. 10 documents for Measles.
3. 18 documents for Dengue Fever.
4. 17 documents for Typhoid Fever.
5. 13 documents for Diarrhea.
6. 10 documents for Pharyngitis.
7. 17 documents for Gastroenteritis.
8. 12 documents for Pneumonia.

These medical records were then put into a table saved in a csv file format. The data from the medical record documents that were used were data of pulse rate, respiration rate, temperature, pain, weight, height, and positive anamnesis. All the documents were put together into a sorted dataset table in one file.

2. Data Reading

The process is to read the data put into the system. The data were composed of file, the value of K, the value of M, the training and testing comparison values and class selection. The value of K was filled out with a value of 10, the value of M with 2, the training and testing comparison value with 80, while the class selections were dengue fever and pneumonia. The results of this process can be seen in figure 2.

```
Banyaknya Dataset :  106
Banyaknya Dataset Yang Terfilter :  30
Banyaknya Data Training :  24
Banyaknya Data Testing :  6
```

Figure 2 Results of Data Reading

3. Tokenization

In this process, string was trimmed from the training and testing data documents. Tokenization was done to cut the strings that contained character "_", so as to ease the next step. This process used several functions in python to simplify the process.

4. Creating Attributes

In this process, any terms identified to be duplicated based on the tokenization results were erased. This process only used looping and branching.

5. TF Matrix

This process aimed to measure the number of terms in the documents. The size of this matrix was length of document x length of attributes. The program codes used to fill out the matrix value were a combination of nested loop and branching.

6. IDF Matrix

This process measured the number of the appearances of value > 0 for all the documents on all the attributes in the TF matrix. The size of this matrix was the length of attribute x 1. This matrix was created using equation as explained in the previous part.

7. TF-IDF Matrix

This process aimed to obtain the weight of the document data. In this process, scalar multiplication between the TF and IDF matrices was performed. This process used a NumPy library to facilitate the scalar multiplication. The result can be seen in figure 3 TF-IDF Matrik.

```
Nilai TFIDF :
[[0.         0.         0.92081875 ... 0.         0.         0.92081875]
 [1.39794001 0.         0.         ... 0.         0.         0.        ]
 [0.         0.         0.         ... 0.         0.         0.        ]
 ...
 [0.         0.         0.         ... 0.         0.69897    0.        ]
 [0.         0.16749109 0.         ... 0.         0.16749109 0.16749109]
 [0.49485002 0.         0.49485002 ... 0.49485002 0.         0.        ]]
```

Figure 3 TF-IDF Matrix

8. Cosine Similarity

This stage measured the distance between the testing and training data. By using equation (3) where d_{ik} is the value of TF-IDF from the i-th data training document and d_{jk} is the value of TF-IDF from the j-th document. The results of the calculation were then sorted from large to small. This process used a number of libraries in Python. The calculation results can be seen in figure 4. Cossine similarity results.

```
Nilai Cosine Similarity :
[[0.24534877591719567, 'dengue'], [0.1895887908977689, 'dengue'],
[0.15191213309254606, 'dengue'], [0.11946181672557761, 'dengue'],
[0.10889181606711107, 'dengue'], [0.10269018163134175, 'dengue'],
[0.09377827238877195, 'dengue'], [0.08194385102881893, 'dengue'],
[0.08177818838631745, 'dengue'], [0.06107312718732327, 'pnemonia'],
[0.060177657243732655, 'dengue'], [0.05426025696441193, 'pnemonia'],
[0.05167984666335912, 'dengue'], [0.04099326544628275, 'dengue'],
[0.03968636818938708, 'dengue'], [0.01418116817505414, 'pnemonia'],
[0.012375659771556963, 'pnemonia'], [0.012282504046235094, 'pnemonia'],
[0.01209606866276496, 'pnemonia'], [0.01074499100958838, 'pnemonia'],
[0.008881778849310174, 'pnemonia'], [0.008314027439731171, 'pnemonia'],
[0.002900221286945139, 'dengue'], [0.0028886041520608647, 'dengue']]
```

Figure 4 Cosine Similarity Results

9. Neighbors

In this process, the results of cosine similarity calculation equal to the value of K were taken. This process only used looping. The results of the process can be seen in figure 5 Neighbors

```
Tetangga :
[[0.24534877591719567, 'dengue'], [0.1895887908977689, 'dengue'],
[0.15191213309254606, 'dengue'], [0.11946181672557761, 'dengue'],
[0.10889181606711107, 'dengue'], [0.10269018163134175, 'dengue'],
[0.09377827238877195, 'dengue'], [0.08194385102881893, 'dengue'],
[0.08177818838631745, 'dengue'], [0.06107312718732327, 'pnemonia']]
```

Figure 5 Neighbor

10. Counting Votes

This process aimed to find out about the number of votes obtained on each label in the neighbors. This process used the counter library to count the number of votes on the label.

11. Calculating Membership

This stage aimed to calculate the membership scores to the labels of the existing neighbors. This process used the formula in equation (5) in the form of program code. The final result was in the form of list filled with dictionary which contained key of the label and value of the calculation results. The calculation results can be seen in Figure 6.

```
Nilai Membership :
[{'dengue': 0.9510000000000001, 'pnemonia': 0.049}, {'dengue':
0.9510000000000001, 'pnemonia': 0.049}, {'dengue': 0.9510000000000001,
'pnemonia': 0.049}, {'dengue': 0.9510000000000001, 'pnemonia': 0.049},
{'dengue': 0.9510000000000001, 'pnemonia': 0.049}, {'dengue':
0.9510000000000001, 'pnemonia': 0.049}, {'dengue': 0.9510000000000001,
'pnemonia': 0.049}, {'dengue': 0.9510000000000001, 'pnemonia': 0.049},
{'dengue': 0.9510000000000001, 'pnemonia': 0.049}, {'dengue': 0.441,
'pnemonia': 0.559}]
```

Figure 6 Membership Results

12. Calculating Fuzzy

This stage aimed to calculate to obtain fuzzy set. The formula in equation (4) was implemented into program code. U_{ij} is the value obtained from the calculation of membership, while x - x_j is euclidian distance formula which was not used in this study because this study used cosine similarity.

After the calculation, the value of the fuzzy set was obtained by calculating the values of the sets of each label on all the neighbors. After that, the maximum value of the fuzzy set was calculated. Finally, label was obtained based on the maximum value. The calculation result can be seen in figure 7.

```
Matrix Fuzzy :
[[0.01553023 0.00080019]
 [0.0260088  0.0013401 ]
 [0.04050985 0.00208726]
 [0.06550695 0.00337523]
 [0.07884154 0.00406229]
 [0.08865184 0.00456776]
 [0.10630192 0.00547718]
 [0.13922359 0.00717346]
 [0.13978823 0.00720255]
 [0.11622601 0.14732504]]

Total Fuzzy :
[0.81658896 0.18341104]

Nilai Maksimum Fuzzy :
0.8165889646392839

Hasil Klasifikasi :  dengue

Hasil Sebenarnya :  dengue
```

Figure 7 Classification Results

13. Testing

The last stage was to test the classification model using Confusion Matrix.

Table 3 Classification Results

Classification Result	Testing Data	Status
Dengue	Pnemonia	Incorrect
Dengue	Dengue	Correct
Dengue	Dengue	Correct
Pnemonia	Pnemonia	Correct
Dengue	Dengue	Correct
Dengue	Dengue	Correct

Table 3 classification results provides information needed to measure the level of accuracy. The information is as follows:

1. Four Dengue testing data classified as Dengue.
2. Zero Dengue testing data classified as Pnemonia.
3. One Pneumonia testing data classified as Dengue.
4. One Pneumonia testing data classified as Pnemonia.

Based on the information, the confusion matrix can be constructed as follows.

Table 4 Confusion Matrix

Class	Classified as Dengue	Classified as Pnemonia
Dengue	4	0
Pnemonia	1	1

After developing the confusion matrix, the values of precision, recall, and accuracy were calculated using equation(6) and (7).

Table 5 Precision and Recall

Class	Precision	Recall
Dengue	0.8	1
Measles	1	0.5
Mean	0.9	0.75

Based on table 5 precesion and recall, the value of precision according to the classification sample is 0.9 or 90%. The value of recall is 0.75 or 75%.

The Classification System for Childhood Diseases with Fever Using Fuzzy K-Nearest Neighbor with an input file consisting of 80% training data, K value of 10, and M value of 2, and two classes i.e. dengue and pneumonia, resulted in a level of accuracy of 0.834 or, if converted into a percent, 83.4%.

In evaluating the classification results using Fuzzy K-Nearest Neighbor, it is necessary to use ROC AUC. ROC (Receiver Operating Characteristic) aims to determine the extend to which the developed model is able to distinguish between one disease from the other. The method for the evaluation process is by calculating the value of the area under curve of ROC using a scale as in table 6 scale.

Table 6 AUC Scale

0.5 – 0.6	Fail
0.6 – 0.7	Poor
0.7 – 0.8	Fair
0.8 – 0.9	Good
0.9 – 1.0	Excellent

14. Result of System Development

The result of developing a Classification System for Childhood Diseases with Fever Using Fuzzy K-Nearest Neighbor Method with python and Django framework can be seen in Figure 8 main page, figure 9 Page of file, figure 10 page of inserting single, and figure 11.

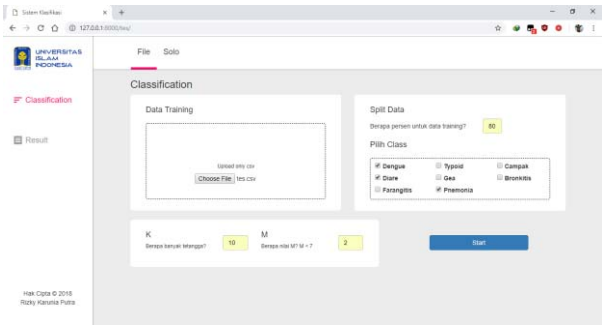

Figure 8 Main Page

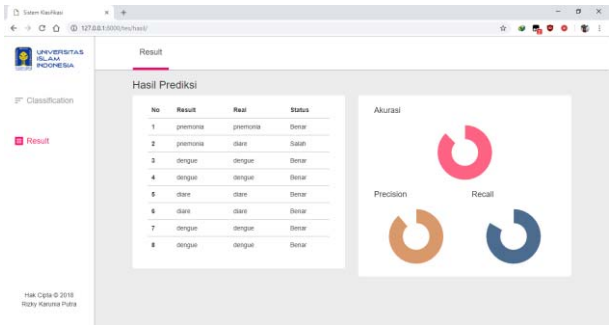

Figure 9 Page of File Classification Results

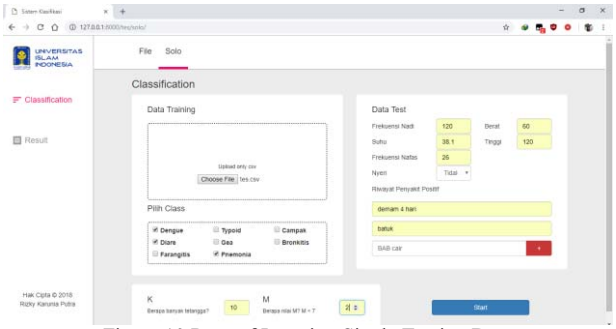

Figure 10 Page of Inserting Single Testing Data

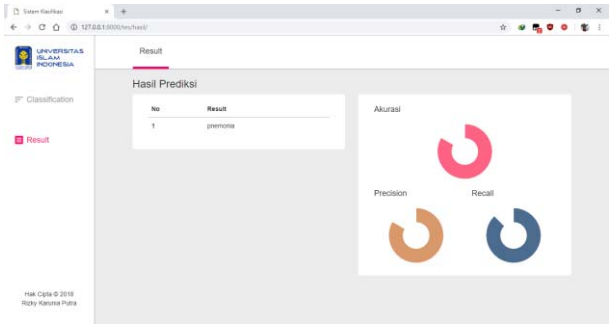

Figure 11 Page of Single Testing Data Classification Results

5 Conclusions and Suggestions

Based on the results of development and testing carried out on the Classification System for Childhood Diseases with Fever Using Fuzzy K-Nearest Neighbor Method, the conclusions are as follows:

a. This system is able to classify medical records in the form of textual documents.
b. This system is able to receive testing data input from a collection of documents or from one document.
c. This system is able to provide a level of accuracy generated from the model.
d. The accuracy is higher than 80%, so the model is good.

Based on these conclusions, this system has already generated an output in accordance with the problem formulation. However, this speech to text system still has many disadvantages such as: limited datasets and limited attributes in the dataset. The limited attributes make it difficult for the system to distinguish between one disease and the other.

After conducting an analysis of the system, the authors realized that this system still has some disadvantages. Thus, the following suggestions are given for further research in order to cover the disadvantages of this system.

a. Add more datasets.
b. Analyze the most suitable attributes to distinguish between one disease from the other.
c. Automatically determine the values of K and M that are suitable to be used in a classification process.

References

[1] A. I. Jamil and K. R. Ucu, "Memasuki Pancaroba, Anak-Anak Rentan Terserang Demam," *Republika*, 2016. [Online]. Available: https://www.republika.co.id/berita/nasional/umum/16/03/10/o3sml3282-memasuki-pancaroba-anakanak-rentan-terserang-demam.

[2] A. Baerheim, "The diagnostic process in general practice : has it a two-phase structure ?," *Fam. Pract.*, vol. 18, no. 3, pp. 243–245, 2001.

[3] F. Wafiyah, N. Hidayat, and R. S. Perdana, "Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam," *J. Pengemb. Teknol. Inf. dan Ilmu Komun.*, vol. 1, no. 10, pp. 1210–1219, 2017.

[4] J. M. Keller, M. R. Gray, and J. Givens, James A, "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-16, no. 4, pp. 580–585, 1985.

[5] K. Markhan, "Simple Guide to Confusion Matrix Terminology," *dataschool.io*, 2014. [Online]. Available: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/.

[6] SAS, "Machine Learning What it is and why it matters," *SAS*, 2016. [Online]. Available: https://www.sas.com/en_sg/insights/analytics/machine-learning.html.

[7] S. Mulyati, S. Kusumadewi, and L. Rosita, "Model Sistem Pendukung Keputusan Untuk Diagnosis Penyakit Anak Dengan Gejala Demam Menggunakan Naive Bayes," in *Seminar Nasional Informatika Medis III (SNIMed III)*, 2012, no. 3, pp. 50–54.

[8] S. D. Nugraha, R. Regasari, M. Putri, and R. C. Wihandika, "Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Dalam Menentukan Status Gizi Balita," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 9, pp. 925–932, 2017.

[9] WHO, "Infectious diseases," *WHO.int*, 2015. [Online]. Available: http://www.who.int/topics/infectious_diseases/en/.