

## **IDENTIFICATION OF MENTAL ILLNESS FROM PATIENT DISEASES USING KNN AND LEVENSHTTEIN DISTANCE ALGORITHM**

Yustika Rahma\*<sup>1</sup>, Agi Prasetiadi<sup>2</sup>, Merlinda Wibowo<sup>3</sup>

<sup>1</sup>Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Indonesia  
Email: [yustikarahma79@gmail.com](mailto:yustikarahma79@gmail.com), [agi@ittelkom-pwt.ac.id](mailto:agi@ittelkom-pwt.ac.id), [merlinda@ittelkom-pwt.ac.id](mailto:merlinda@ittelkom-pwt.ac.id)

(Naskah masuk: 9 Juni 2022, Revisi: 18 Juni 2022, diterbitkan: 24 Oktober 2022)

### **Abstract**

According to WHO, in 2017, the estimated number of people with mental disorders worldwide was around 450 million people, including schizophrenia. Globally, for the condition of Southeast Asia alone, the number of people affected by mental disorders is 13.5%. Meanwhile, 13.4% of cases in Indonesia are affected by mental illness. The Association of Mental Medicine Specialists (PDSKJ) during October 2020 noted that 5661 people who did self-examination through the PDSKJ website came from 31 provinces and found that 32% of the population had psychological problems and 68% had no psychological issues.

Seeing that the level of mental illness in Indonesia is increasing, it is necessary to have a system to help the community with early prevention and treatment. With the growth of technology at its peak, Machine Learning technology can overcome the problem which is part of artificial intelligence. Furthermore, machine learning has an important role in improving the quality of health services because it is able to provide a medical diagnosis to predict disease. Therefore, the authors conducted a study to create a system to identify mental illness using the TF-IDF method. This method calculates the word weighting from a collection of complaints that the user gives. Then, these complaints will be classified using the KNN algorithm classification method and the Levenshtein Distance method to find the distance between the word inputted by the user and the word in the database and then calculate the number of differences between the two strings in the form of a matrix. The accuracy result of this machine learning classification is 0.928 or 93%, and will be visualized through web-based software using the Flask framework.

**Keywords:** Mental illness, Machine Learning, TF-IDF, KNN, Levenshtein Distance

## **IDENTIFIKASI PENYAKIT MENTAL DARI KELUHAN PASIEN MENGGUNAKAN ALGORITMA KNN DAN LEVENSHTTEIN DISTANCE**

### **Abstrak**

Menurut WHO pada tahun 2017 perkiraan jumlah penderita gangguan jiwa di dunia adalah sekitar 450 juta orang termasuk skizofrenia. Secara global, untuk kondisi Asia Tenggara sendiri tidak jumlah yang terkena gangguan mental yaitu 13,5%. Sedangkan untuk kasus di Indonesia sebesar 13,4% yang terkena penyakit mental. Perhimpunan Dokter Spesialis Kedokteran Jiwa (PDSKJ) selama Oktober 2020, mencatat masyarakat yang melakukan swaperiksa melalui web PDSKJ berjumlah 5661 kasus terdiri dari 31 Provinsi dan ditemukan sebanyak 32% masyarakat mengalami gangguan psikologis dan 68% tidak ada gangguan psikologis.

Melihat tingkat penyakit mental di Indonesia semakin meningkat, maka dibutuhkan sistem yang dapat membantu masyarakat untuk pecegahan dan pengobatan sejak dini. Dengan adanya Pertumbuhan teknologi yang sedang pada puncaknya, masalah tersebut bisa diatasi dengan teknologi *Machine Learning* yang menjadi bagian dari kecerdasan buatan. *Machine learning* berperan penting dalam meningkatkan kualitas layanan kesehatan karena mampu memberikan diagnosis medis untuk memprediksi penyakit. Oleh karena itu, penulis melakukan penelitian untuk membuat sebuah sistem yang dapat mengidentifikasi penyakit mental dengan menggunakan metode TF-IDF untuk pembobotan kata dari kumpulan keluhan-keluhan yang user berikan dan dari keluhan tersebut akan diklasifikasikan menggunakan metode klasifikasi algoritma KNN dan menggunakan metode *Levenshtein Distance* untuk mencari jarak antara kata yang diinputkan oleh *user* dengan kata pada database lalu menghitung jumlah perbedaan antar kedua string dalam bentuk matriks. Hasil akurasi dari klasifikasi *machine learning* ini yaitu sebesar 0,928 atau 93% dan akan divisualisasikan melalui perangkat lunak berbasis *web* menggunakan *framework* Flask.

**Kata kunci:** Penyakit Mental, Machine Learning, TF-IDF, KNN, Levenshtein Distance

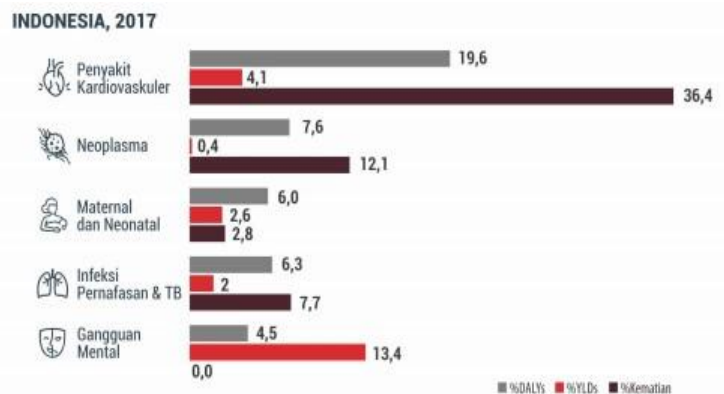
## 1. PENDAHULUAN

Pada awalnya, kesehatan mental hanya sebatas pada individu yang mempunyai gangguan mental atau kejiwaan saja dan tidak berlaku untuk populasi umum. Namun, seiring jalannya waktu pandangan tersebut bergeser sehingga kesehatan mental tidak hanya terpaku kepada individu yang memiliki gangguan mental atau jiwa saja, tetapi diperuntukan untuk individu yang mentalnya sehat. Yaitu bagaimana individu mengeksplorasi diri mereka sendiri dan bagaimana mereka berinteraksi dengan baik di lingkungan mereka [1].

Menurut Undang-Undang Nomor 18 Tahun 2014, Kesehatan jiwa adalah kondisi dimana seorang individu dapat berkembang secara fisik, mental, spiritual, dan sosial sehingga individu tersebut menyadari kemampuan sendiri, dapat mengatasi tekanan, dapat bekerja secara produktif, dan mampu memberikan kontribusi untuk komunitasnya. Kesehatan mental atau jiwa dapat merujuk pada bagaimana individu mampu menyesuaikan diri serta berinteraksi dengan baik dengan lingkungan sekitarnya[2]. Sedangkan Gangguan jiwa sendiri merupakan manifestasi dari bentuk penyimpangan perilaku akibat adanya distorsi emosi, sehingga ditemukannya ketidakwajaran dalam tingkah laku, hal ini terjadi karena menurunnya semua fungsi kejiwaan[3].

Gangguan mental bisa ringan hingga berat, yang dapat memengaruhi kemampuan seseorang dalam menjalani kehidupan sehari-hari. Faktanya, kebanyakan orang dengan gangguan jiwa masih bisa hidup seperti orang normal. Namun, jika kondisinya tidak stabil, seseorang mungkin memerlukan perawatan intensif di rumah sakit untuk mengobati kondisinya. Kondisi ini seringkali memicu keinginan untuk menyakiti diri sendiri atau mengakhiri hidup[4].

Perhimpunan Dokter Spesialis Kedokteran Jiwa (PDSKJ) selama Oktober 2020, mencatat masyarakat yang melakukan swaperiksa melalui web PDSKJ berjumlah 5661 kasus terdiri dari 31 Provinsi dan ditemukan sebanyak 32% masyarakat mengalami gangguan psikologis dan 68% tidak memiliki gangguan psikologis. Dr. Diah Setia Utami Sp. KJ, MARS selaku ketua umum PDSKJ menegaskan peningkatan gangguan kesehatan jiwa ini perlu ditindaklanjuti agar tidak menyebabkan gangguan yang lebih parah di kemudian hari. Semua komponen perlu bekerjasama dan memperhatikan tersebut.



Gambar 1 Beban Penyakit di Indonesia

Pada Gambar 1 menunjukkan bahwa menurut WHO pada tahun 2017, perkiraan jumlah orang dengan gangguan mental secara global adalah sekitar 450 juta orang termasuk skizofrenia, baik orang dewasa maupun anak-anak dapat terkena gangguan jiwa ini. Secara global, untuk kondisi di Asia Tenggara jumlah yang terkena gangguan mental yaitu 13,5%. Sedangkan untuk kasus di Indonesia sebesar 13,4% yang terkena penyakit mental[3].

Dengan adanya permasalahan diatas maka diperlukan sebuah teknologi yang dapat digunakan oleh masyarakat untuk mengetahui bagaimana keadaan psikologis atau mental masyarakat, apakah mengalami gangguan atau tidak. Penggunaan teknologi untuk memprediksi akan adanya potensi seseorang menderita gangguan mental akan lebih efektif menggunakan data-data penyakit mental yang dihimpun dari kasus-kasus yang pernah terjadi.

Melihat tingkat penyakit mental di Indonesia semakin meningkat, maka dibutuhkan sistem yang dapat membantu masyarakat untuk pencegahan dan pengobatan sejak dini. Selain itu akan membantu masyarakat dalam meminimalisir biaya yang dikeluarkan untuk konsultasi ke psikolog, karena membutuhkan biaya yang tidak sedikit. Masyarakat juga biasanya merasa malu untuk melakukan konsultasi ke psikolog, karena akan dikira atau disebut "orang gila" oleh masyarakat sekitar.

Dengan adanya Pertumbuhan teknologi yang sedang pada puncaknya, masalah tersebut bisa diatasi dengan teknologi Machine Learning yang menjadi bagian dari kecerdasan buatan. Machine Learning memiliki peran penting dalam meningkatkan kualitas layanan kesehatan karena mampu menyajikan diagnosis medis hingga memprediksi penyakit[6]. Metode yang digunakan pada penelitian ini adalah algoritma TF-IDF atau pembobotan kata untuk menghitung bobot setiap kata yang umum digunakan, metode KNN yang berguna untuk mengklasifikasi suatu data penyakit mental berbentuk teks dan algoritma *Levenshtein Distance* untuk mencari jarak terdekat teks setelah dilakukannya klasifikasi. Algoritma K-nearest

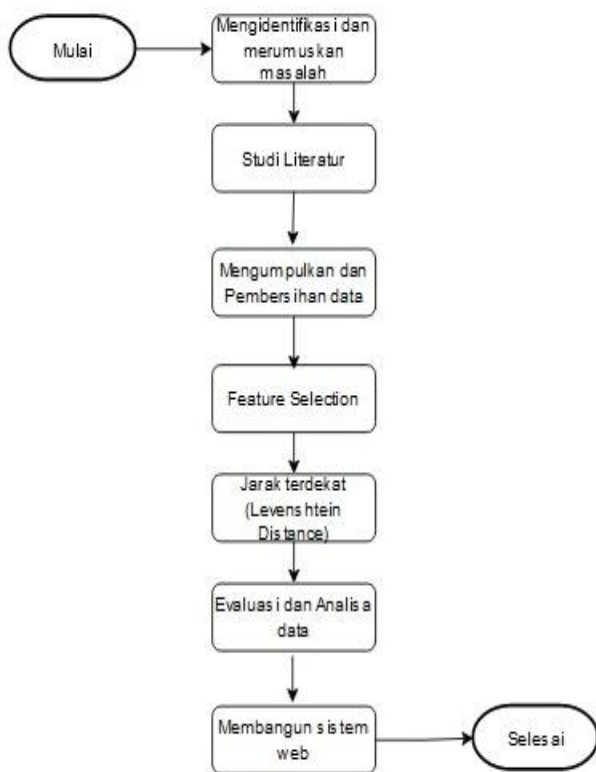
Neighbor atau biasa disebut KNN ialah algoritma klasifikasi data mining yang digunakan untuk klasifikasi objek baru berdasarkan objek terdekatnya, dan berdasarkan data pembelajaran yang diberi label[7]. Dan sedangkan algoritma Levenshtein Distance bekerja dengan cara mencari jarak antara kata yang diinputkan oleh user dengan kata pada *database* lalu menghitung jumlah perbedaan antar kedua *string* dalam bentuk matriks[8].

Sehingga penelitian ini akan melakukan identifikasi penyakit mental dengan algoritma klasifikasi KNN dan algoritma Levenshtein Distance yang akan dibangun berbasiskan sistem web yang menggunakan *micro-framework* Flask sehingga pengembangan lebih efisien. Flask juga merupakan *micro-framework* berbasis bahasa pemrograman python, yang tidak memiliki terlalu banyak *tools* dan *library*, sehingga lebih ringan dan tidak memakan banyak resource[5]. Sistem *web* juga bisa digunakan siapa saja yang ingin mengetahui apakah memiliki potensi penyakit gangguan mental sehingga dapat dilakukan pencegahan dan pengobatan sejak dini.

## 2. METODE PENELITIAN

### 2.1 Diagram Penelitian

Diagram alir penelitian atau proses penelitian keseluruhan dan langkah-langkah dalam menyelesaikan penelitian ini ditunjukkan pada Gambar 2.



Gambar 2 Diagram Alir Penelitian

### 2.2 Mengidentifikasi dan Merumuskan Masalah

Langkah awal yang dilakukan yaitu mengidentifikasi masalah dan merumuskan masalah terkait penelitian yang akan diteliti. Menetapkan bidang, topik, masalah juga metode yang akan digunakan untuk menyelesaikan permasalahan tersebut.

### 2.3 Studi Literatur

Pada proses ini penulis melakukan pengumpulan data-data yang berhubungan dengan penelitian yang akan diteliti. Dengan cara melakukan riset kepustakaan, mencari referensi dari berbagai jurnal, buku, internet, dan media elektronik yang berkaitan dengan klasifikasi data, algoritma TF-IDF, algoritma KNN, algoritma *Levenshtein Distance*, flask dan ilmu tentang kejiwaan. Studi literatur ini bertujuan untuk memperkuat dasar pada penelitian ini.

### 2.4 Mengumpulkan Data

Tahap selanjutnya yaitu mengumpulkan data dan pembersihan data. Pengambilan data-data diambil dari beberapa buku kesehatan psikiatri yang berhubungan dengan penyakit mental. Kemudian pembersihan dilakukan dengan hanya mengambil data yang akan digunakan untuk penelitian yaitu macam-macam penyakit mental dan gejala-gejalanya.

Tabel 1 Data Penyakit Mental Psikosis

No	Penyakit	Gejala 1	Gejala 2	Gejala 3	Gejala 4
1	Bipolar	Pesimistik	Optimis berlebih	Mudah Lelah	Mudah Marah
2	Anxiety	Cemas	Depresi	Gelisah	Mudah Marah
3	OCD	Melakukan Aktivitas Berulang	Perfeksionis	Punya Pikiran Sendiri	Cemas
4	Skizofrenia	Halusinasi	Gelisah	Cemas	Apatis

Pada Tabel 1 berisikan data-data penyakit mental jenis psikosis yang akan digunakan pada penelitian ini. Berikut ada 4 macam penyakit dan 12 macam gejala.

### 2.5 Proses Text Processing

Tahap ini melakukan proses *Text Processing* dengan proses *Stemming* dari kata yang diinputkan oleh user atau pasien. Tahap *Stemming* adalah tahap mengembalikan kata-kata yang diperoleh dari *Filtering*, kemudian dihilangkan kata imbuhan awal dan imbuhan akhirnya[9].

### 2.6 Proses Feature Selection

Tahap selanjutnya akan dilakukan proses *feature selection* atau seleksi fitur menggunakan pembobotan TF-IDF[10]. Tahap ini bertujuan untuk mengetahui berapa banyak kemunculan data pada dokumen dan untuk meningkatkan nilai akurasi klasifikasi[11].

**2.7 Membangun Klasifikasi Model**

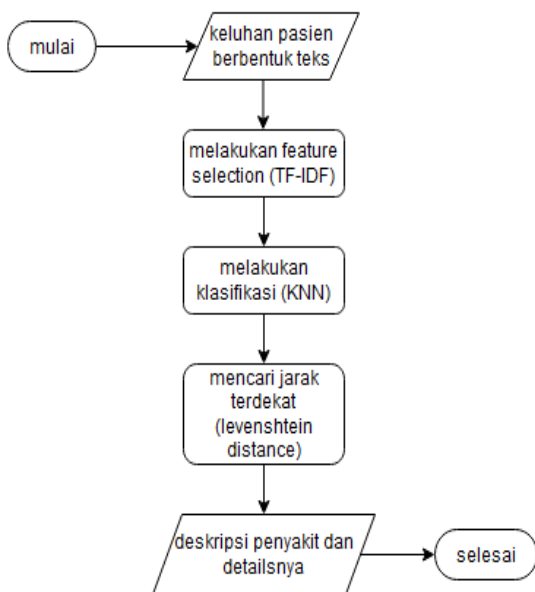
Setelah melakukan pembobotan kata tahap selanjutnya yaitu membangun klasifikasi model dengan metode *Levenshtein Distance*. Dalam tahap klasifikasi memiliki 2 proses dataset, yaitu proses *training* dan proses *testing*. Dalam analisis ini data akan dibagi menjadi 80% dan 20%. Sebanyak 80% dataset digunakan untuk proses training, yaitu proses melatih mesin tentang pengetahuan dataset yang sudah diberi label sebelumnya. Kemudian pada proses *testing* menggunakan 20% dari total dataset untuk mengetahui tingkat akurasi pada klasifikasinya.

**2.8 Evaluasi dan Analisa Performa Klasifikasi**

Pada tahap ini melakukan evaluasi dan analisa performa aplikasi apakah saat mengklasifikasikan data penyakit dengan baik atau tidak dengan menggunakan metode *Confusion Matrix*[12].

**2.9 Menerapkan Model Klasifikasi dalam Machine Learning**

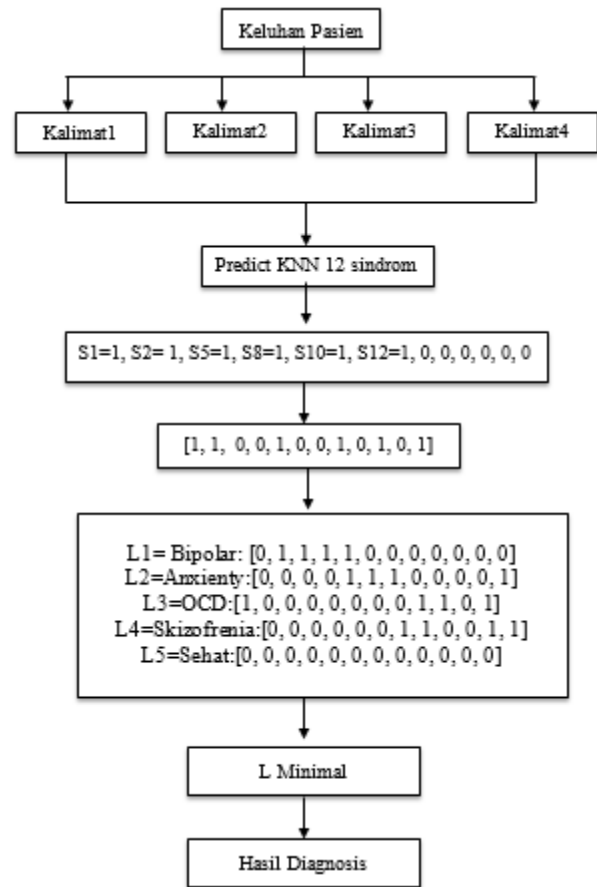
Pada tahap terakhir yaitu membuat *machine learning* berbasis *web* menggunakan *micro-framework flask*. Alur rancangan secara umum dapat dilihat pada Gambar 3.



Gambar 3 Diagram klasifikasi machine learning

Pada Gambar 3 menjelaskan tentang alur rancangan sistemnya secara umum yaitu ada tahap dimana pasien memberikan keluhan berbentuk teks, lalu diproses dengan pembobotan kata, kemudian kalimat tersebut diklasifikasikan dengan algoritma KNN dan dicari jarak terdekatnya antar kalimat dengan algoritma Levenshtein Distance sehingga mendapatkan hasil deskripsi penyakitnya. Untuk alur lebih

detailnya dalam mengklasifikasikan penyakit mental pada *machine learning* dapat dilihat pada Gambar 4.



Gambar 4 Diagram Lengkap Klasifikasi Machine Learning

Pada Gambar 4 yang menunjukkan diagram lengkap proses klasifikasi machine learning pendeteksi penyakit mental didapatkan bahwa pasien menginputkan keluhannya berbentuk teks yang dipisahkan dengan tanda titik (.) sebanyak 4 kalimat, lalu kalimat tersebut akan diproses prediksi dengan klasifikasi algoritma KNN, setelah di prediksi akan dihasilkan prediksi sindrom-sindrom sebanyak 12 sindrom berbentuk list, lalu setelah didapatkan list tersebut dilakukan pencarian jarak terdekat dari seluruh list berdasarkan 4 penyakit yaitu Bipolar, Anxiety, OCD, dan Skizofrenia. Kemudian jika list tersebut tidak cocok dengan ke-4 penyakit tersebut maka pasien termasuk ke kategori sehat.

**3. HASIL DAN PEMBAHASAN**

**3.1 Pelabelan Data**

Pada tahap ini setelah dilakukannya pengumpulan data, selanjutnya melakukan pelabelan data set ciri gejala penyakit berdasarkan diagnosis pasien. Setelah melakukan pengumpulan data, data tersebut

masih bersifat mentah yang dimana belum dapat diklasifikasikan. Maka dilakukan pelabelan data supaya data set dapat diklasifikasi berdasarkan ciri gejalanya. Proses pelabelan data dilakukan melalui beberapa proses yaitu:

1. Pelabelan dilakukan berdasarkan data set gejala penyakit tersebut.
2. Selanjutnya memberi label pada seluruh data set gejala penyakit yang sudah ditentukan yaitu label 1 untuk kalimat yang termasuk ciri gejala, dan label 0 untuk kalimat yang tidak termasuk pada ciri gejala tersebut.

### 3.2 Proses Stemming Data

Sebelum melakukan pembobotan menggunakan algoritma TF-IDF, data set perlu di *stemming* atau dipisahkan kata dasarnya dari kata yang berimbuhan. Implementasi *stemming* data dengan pemrograman python menggunakan pengujian *library sastrawi.stemmer.stemmerfactory*.

### 3.3 Pembobotan TF-IDF

Setelah data set dilakukan stemming langkah selanjutnya data dilakukan pembobotan menggunakan algoritma TF-IDF untuk mencari peluang kemunculan kata pada suatu dokumen dan meningkatkan nilai akurasi[13].

Perhitungan TF-IDF dengan cara manual menggunakan beberapa sample data diantaranya:

**Data1: saya mudah marah**

**Data2: saya mudah lelah**

Langkah untuk melakukan perhitungan TF-IDF secara manual yaitu:

- a. Menentukan term unik atau fitur  
Langkah awal pada perhitungan ini mencari term unik atau fitur yaitu dari data yang sudah ada. berikut contoh implementasi kemunculan term yang ada pada semua dokumen di Tabel 2.

No	Term/Fitur
1	Saya
2	Mudah
3	Marah
4	Saya
5	Mudah
6	Lelah

- b. Menghitung Pembobotan TF dan DF

Setelah diperoleh term pada semua dokumen, langkah selanjutnya mencari nilai TF atau term frekuensi dari masing-masing term dan melakukan perhitungan DF atau jumlah dokumen yang terdapat dalam suatu term. Penjelasan detailnya terdapat pada Tabel 3.

Tabel 3 Perhitungan TF dan DF

No	Term	TF		DF
		Data1	Data2	
1	Saya	1	1	2
2	Mudah	1	1	2
3	Marah	1	0	1
4	Lelah	0	1	1

### c. Menghitung IDF

Setelah melakukan perhitungan TF dan DF selanjutnya menghitung IDF atau invers dari DF dengan rumus:

$$idf_i = \log \frac{N}{df_i}$$

Dimana:

$idf_i$  = Inverse Document Frequency

N = jumlah dokumen yang terambil oleh sistem

$df_i$  = banyaknya dokumen dalam koleksi dimana term muncul didalamnya / kata yang dicari.

### d. Menghitung TF-IDF

Tahap terakhir pada perhitungan TF-IDF yaitu melakukan perkalian dari hasil TF dan IDF dari masing-masing term[14]. Pada Tabel 4 merupakan detail hasil perhitungan TF-IDF.

Tabel 4 Perhitungan TF-IDF

No	Term	Data1	Data2
1	Saya	0	0
2	Mudah	0	0
3	Marah	0.301	0
4	Lelah	0	0.301

## 3.4 Hasil Proses Klasifikasi

### 3.4.1 Perbandingan Algoritma Klasifikasi

Sebelum dilakukannya klasifikasi data kalimat keluhan pasien pada gejala penyakit, penulis melakukan perbandingan 5 algoritma klasifikasi yaitu Random Forest, SVM (Support Vektor Machine), Naive Bayes, KNN (K-Nearest Neighbor), dan Perceptron untuk membandingkan mana yang memiliki akurasi tertinggi untuk mengklasifikasikan kalimat keluhan pasien terhadap gejala penyakitnya. Implementasi perbandingan algoritma ini menggunakan pemrograman python dengan library `sklearn.metrics.model_classification_report`, `confusion_matrix`, `accuracy_score`. Hasil dari contoh hasil implementasi perbandingan 5 Algoritma terhadap 12 gejala penyakit dapat dilihat pada Gambar 6.

```

knn
[[107  0]
 [ 11  3]]
precision  recall  f1-score  support
   0      0.91    1.00    0.95    107
   1      1.00    0.21    0.35     14

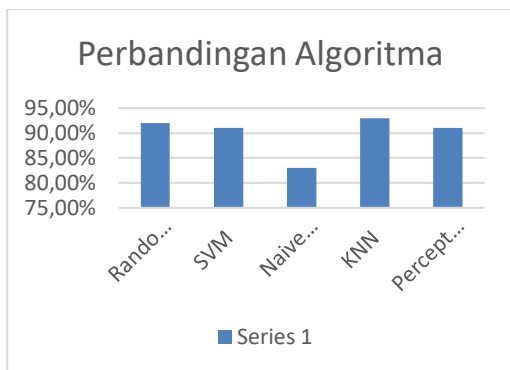
accuracy          0.91    121
macro avg         0.95    0.61    0.65    121
weighted avg      0.92    0.91    0.88    121
0.9090909090909091
    
```

Berikut hasil dari keseluruhan 10 kali percobaan pada setiap algoritma terhadap 12 gejala penyakit tersebut disajikan pada Tabel 5.

Gambar 6 Contoh Hasil Akurasi

Tabel 5 Perbandingan Klasifikasi

Gejala	Random Forest	SVM	NaiveB ayes	KNN	Perceptron
Punya pikiran sendiri	0.932	0.921	0.78	<b>0.945</b>	0.912
Pesimistik	0.905	<b>0.907</b>	0.771	0.902	0.885
Optimis	0.91	0.894	0.83	<b>0.916</b>	0.889
berlebih					
Mudah lelah	0.915	0.916	0.845	<b>0.922</b>	0.907
Mudah marah	0.907	0.901	0.819	0.912	<b>0.913</b>
Depresi	0.906	0.908	0.76	<b>0.925</b>	0.863
Gelisah	0.943	0.907	0.852	<b>0.949</b>	0.93
Apatis	0.937	0.918	0.83	<b>0.955</b>	0.929
Melakukan aktivitas berulang	0.954	0.919	0.877	<b>0.955</b>	0.955
Perfeksionis	0.92	0.919	0.822	<b>0.927</b>	0.922
Halusinasi delusi	<b>0.918</b>	0.904	0.87	0.916	0.913
Cemas	0.917	0.911	0.857	<b>0.92</b>	0.901
<b>Rata-rata</b>	<b>0.92</b>	<b>0.91</b>	<b>0.83</b>	<b>0.928</b>	<b>0.91</b>



Gambar 8 Grafik Rata-Rata Nilai Algoritma

Pada Gambar 8 merupakan grafik rata-rata nilai algoritma keseluruhan dari 12 gejala yang diuji pada 5 algoritma. Dan dapat

dilihat bahwa algoritma KNN tetap menjadi algoritma yang memiliki nilai tertinggi dari seluruh algoritma yaitu dengan nilai 93%. Sehingga dapat disimpulkan bahwa algoritma KNN merupakan algoritma yang memiliki nilai tertinggi dari pengujian pergejala dan memiliki nilai tertinggi dari rata-rata keseluruhan juga. Maka dari itu penulis menggunakan algoritma KNN untuk melakukan klasifikasi pada penyakit mental.

### 3.5 Implementasi Levenshtein Distance

Setelah data keluhan pasien diklasifikasikan menggunakan algoritma KNN langkah selanjutnya yaitu mencari jarak terdekat antara keluhan pasien yang sudah diklasifikasikan dengan ciri-ciri penyakit. Implementasi algoritma levenshtein

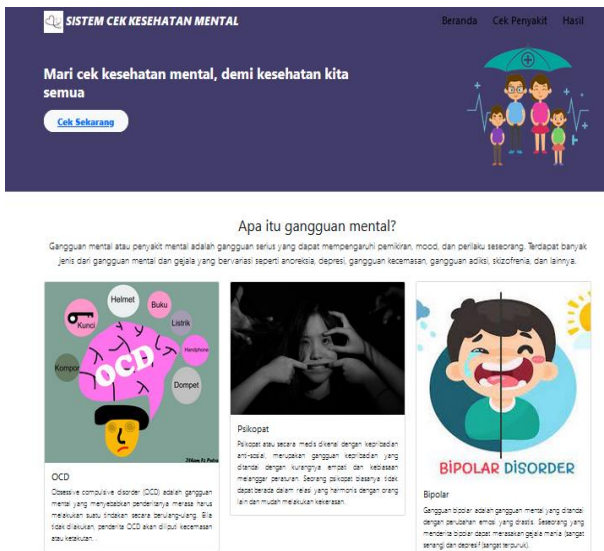


distance dengan pemrograman python menggunakan *library sklearn.feature\_extraction.text*. Berikut hasil dari implementasi algoritma levenshtein distance yang memperoleh hasil kalimat keluhan pasien yang sudah di klasifikasi terhadap gejala penyakit dapat di sorting dan dicari jarak terdekat yang paling mendekati gejala penyakit tersebut[15]. Sehingga dapat diperoleh hasil akhir penyakit tersebut. Pada tabel 5 berikut menunjukkan hasil dari proses mencari jarak terdekat menggunakan Levenshtein Distein dimana ada proses sebelum dilakukannya sorting dan setelah dilakukannya sorting. Hasil diagnosis dapat dilihat dari hasil akhir setelah disorting.

Tabel 5 Hasil Levenshtein Distance

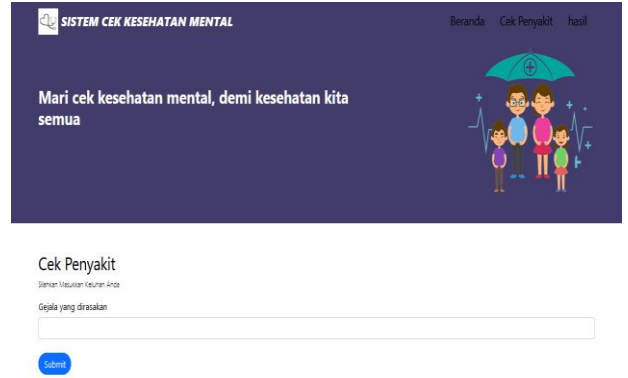
Sebelum di sorting	Sesudah di sorting
['Bipolar',4]	Bipolar
['Anxiety',6]	
['OCD',9]	
['Skizofrenia',5]	

3.6 Implementasi *Machine Learning* pada *Website* menggunakan *Flask*  
 Berikut hasil tampilan *website* Pendeteksi Penyakit Mental yang berhasil dijalankan diflask melalui alamat <http://127.0.0.1:5000/>, memiliki 3 halaman yaitu halaman beranda, halaman cek penyakit dan halaman hasil.



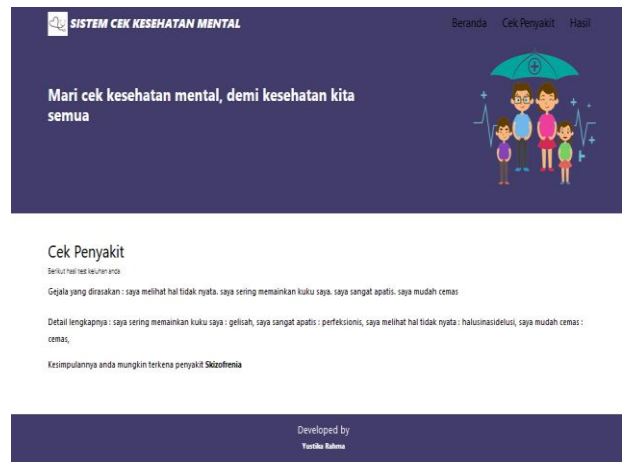
Gambar 9 Halaman Beranda

Pada Gambar 9 halaman beranda yaitu halaman awal pada *website* yang berisikan informasi terkait gangguan mental dan macam-macamnya.



Gambar 10 Halaman Cek Penyakit

Pada Gambar 10 halaman cek penyakit user diminta untuk memasukkan keluhan terkait gejala yang dirasakan berupa teks panjang.



Gambar 11 Halaman Hasil

Pada Gambar 11 halaman hasil user akan mengetahui jenis penyakit mental apa sesuai dengan gejala yang diinputkan dan detail terkait gejala yang dirasakan.

#### 4. DISCUSSION

Pada penelitian oleh Rizki Tri Wahyuni, Dkk (2017) melakukan penelitian bagaimana cara mencari klasifikasi dokumen skripsi dengan metode Cosine similarity dan Levenshtein distance. Kelemahannya penelitian ini yaitu masih dibutuhkannya penggunaan term yang unik, karena masih banyak kata yang sama dengan kata kunci sehingga sistem memilih nilai tertinggi dari cosine similarity. Kesimpulannya hasil rata-rata presentasi kelayakan sebesar 88,3% dan ketepatan klasifikasi mencapai 98%. Sehingga layak digunakannya algoritma Levenshtein Distace untuk pencarian jarak terdekat pada penelitian ini. Penelitian ini juga telah dilakukan perbandingan 5 algoritma klasifikasi untuk menemukan algoritma klasifikasi terbaik. Dan dihasilkan bahwa algoritma Random Forest memiliki

akurasi sebanyak 92%, algoritma SVM memiliki akurasi sebanyak 91%, algoritma Naive Bayes memiliki akurasi sebanyak 83%, KNN memiliki akurasi sebanyak 93% dan algoritma Perceptron sebanyak 91%. Dapat ditarik kesimpulan bahwa dari ke 5 algoritma tersebut KNN merupakan algoritma klasifikasi yang memiliki akurasi tertinggi dan layak untuk digunakan dalam penelitian mengidentifikasi penyakit mental ini.

## 5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, menghasilkan bentuk aplikasi pendeteksi penyakit mental berdasarkan keluhan pasien menggunakan algoritma KNN dan *Levenshtein Distance*. Aplikasi ini telah berhasil dibuat dalam bentuk *machine learning* yang secara otomatis menghasilkan hasil klasifikasi berdasarkan keluhan pasien. Oleh karena itu, hal ini dapat membantu meminimalisir konsultasi biaya ke psikolog. Cara medeteksi penyakit mental digunakan 2 algoritma yaitu KNN dan *Levenshtein Distance*. Algoritma ini berhasil diimplementasikan ke dalam *Machine Learning* yang telah berhasil dibangun untuk melakukan deteksi penyakit mental berdasarkan data-data keluhan pasien. Perbandingan algoritma dengan pembagian data *training* dan data *testing* yaitu 80 : 20 menunjukkan algoritma Random Forest memiliki akurasi sebanyak 92%, algoritma SVM memiliki akurasi sebanyak 91%, algoritma Naive Bayes memiliki akurasi sebanyak 83%, KNN memiliki akurasi sebanyak 93% dan algoritma Perceptron sebanyak 91%. Dapat ditarik kesimpulan bahwa dari ke 5 algoritma tersebut KNN merupakan algoritma klasifikasi yang memiliki akurasi tertinggi dan layak untuk digunakan dalam penelitian mengidentifikasi penyakit mental ini.

## DAFTAR PUSTAKA

- [1] Fakhriyani Diana Vidya, *kesehatan mental*. 1st ed. Jawa timur: CV. Duta Media, 2017.
- [2] F. A. Saputra, Y. Y. Ranimpi, and R. T. Pilakoannu, "Kesehatan Mental dan Koping Strategi di Kudangan, Kecamatan Delang, Kabupaten Lamandau Kalimantan Tengah: Suatu Studi Sosiodemograf," *Humanit. (Jurnal Psikologi)*, vol. 2, no. 1, pp. 63–74, 2018, doi: 10.28932/humanitas.v2i1.1046.
- [3] Indrayani yoeyoen aryantin, *Situasi Kesehatan Jiwa di Indonesia*. 1st ed. Jakarta: kemkes.go.id, 2019.
- [4] B. A. Fundrika. (2020, Oct 14). "Persen Masyarakat Indonesia Alami Masalah Gangguan Jiwa Selama Pandemi[Online]". Available : <https://www.suara.com/health/2020/10/14/171948/68-persen-masyarakat-indonesia-alami-masalah-gangguan-jiwa-selama-pandemi>.
- [5] N. Aziz, "Implementasi algoritma knn untuk memprediksi potensi penyakit jantung dengan python flask," Skripsi Teknik Informatika. Fakultas Komunikasi dan Informatika. Universitas Muhammadiyah Surakarta. Surakarta. Indonesia. 2021.
- [6] I. Sulistiani, E. Mufida, P. M. Yasser, and L. Alamsyah, "Systematic Literature Review : Bankruptcy Prediction Menggunakan Teknik Machine Learning dan Deep Learning," *Jurnal Intech*. vol. 2, no. 1, pp. 13–18, 2021.
- [7] N. M. Putry, B. N. Sari, M. Kom, T. Informatika, and U. S. Karawang, "Komparasi Algoritma Knn Dan Naive Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Melitus," *Jurnal Sains dan Manajemen*. vol. 10, no. 1, pp.47-48, 2022.
- [8] H. T. Saidah, M. S. N. Ishlah, and N. N. Rokhmah, "Autocorrect pada Modul Pencarian Drugs e-Dictionary Menggunakan Levenshtein Distance," *Rekayasa Sist. dan Teknol. Inf.*, vol. 4, no. 1, pp. 64–69, 2020.
- [9] I.Nur Irmasnyah, N.Chafid and Mujiyanto, "Penerapan Filter Kata Menggunakan Metode Stemming Pada Aplikasi Chatting Berbasis Web," *Webinar Nasional Cendekiawan*. vol. 1, no. 1, pp. 1–9, 2020.
- [10] R. T. Wahyuni, D. Prastiyanto, and E. Suprptono, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," *J. Tek. Elektro Univ. Negeri Semarang*, vol. 9, no. 1, pp. 18–23, 2017, [Online]. Available: <https://journal.unnes.ac.id/nju/index.php/jte/article/download/10955/6659>.
- [11] R. I. Saputri, "Perbandingan Metode Naive Bayes Classifier Dan Support Vector Machine Untuk Klasifikasi Cyber Harassment Pada Twitter," Skripsi Teknik Informatika. Fakultas Informatika. IT Telkom Purwokerto. Purwokerto. Indonesia. 2020.
- [12] A. Amrin and H. Saiyar, "Aplikasi Diagnosa Penyakit Tuberculosis Menggunakan Algoritma Naive Bayes," *Jurikom*, vol.5, no. no.5, pp. 498–502, 2018.
- [13] D. S. Hormansyah and Y. P. Utama, "Aplikasi Chatbot Berbasis Web Pada Sistem Informasi Layanan Publik Kesehatan Di Malang Dengan Menggunakan Metode Tf-Idf," *J. Inform. Polinema*, vol. 4, no. 3, p. 224, 2018, doi: 10.33795/jip.v4i3.211.



- [14] R. R. A. Siregar, F. A. Sinaga, and R. Arianto, "Aplikasi Penentuan Dosen Penguji Skripsi Menggunakan Metode TF-IDF dan Vector Space Model," *Comput. J. Comput. Sci. Inf. Syst.*, vol. 1, no. 2, p. 171, 2017, doi: 10.24912/computatio.v1i2.1014.
- [15] Indrayani, D. Sugianti, and M. A. Al Karomi, "Optimasi Parameter K pada Algoritma K-Nearest Neighbour untuk Klasifikasi Penyakit Diabetes Mellitus," *Pros. SNATIF ke-6 Tahun 2019*, no. 2007, pp. 96–101, 2019.

