

TUGAS AKHIR

Identifikasi Berita Hoax Menggunakan Kombinasi Metode K-Nearest Neighbor (KNN) dan TF-IDF Berbasis Web Dengan Menggunakan *Framework Codeigniter*



Oleh:

Danu Nur Irwanto

17053030

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS WIJAYA PUTRA

SURABAYA

2021

HALAMAN JUDUL

Identifikasi Berita Hoax Menggunakan Kombinasi Metode K-Nearest Neighbor (KNN) dan TF-IDF Berbasis Web Dengan Menggunakan Framework Codeigniter

Hoax News Identification Using a Combination of the K-Nearest Neighbor (KNN) Method and Web-Based TF-IDF Using the Codeigniter Framework

TUGAS AKHIR

**Diajukan sebagai salah satu syarat memperoleh
Gelar Sarjana Komputer pada Fakultas Teknik
Universitas Wijaya Putra Surabaya**



Oleh:

Danu Nur Irwanto

17053030

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS WIJAYA PUTRA

SURABAYA

2021

HALAMAN PERSETUJUAN

NAMA : Danu Nur Irwanto
NPM : 17053030
FAKULTAS : Teknik
PROGRAM STUDI : Teknik Informatika
JUDUL : Identifikasi Berita Hoax Menggunakan Kombinasi Metode
K-Nearest Neighbor (KNN) dan TF-IDF Berbasis Web
Dengan Menggunakan *Framework Codeigniter*

Surabaya,
Mengetahui,
Dekan Fakultas Teknik

Disetujui oleh :
Dosen Pembimbing

Slamet Riyadi, S.T., M.T
NIDN : 07119117101

M. harist Murdani, S.Kom., M.Sc.
NIDN : 0731078504

LEMBAR PENGESAHAN

Telah diakui dan disahkan oleh Tim Pemeriksa Tugas Akhir dan dinyatakan LULUS.
Dengan demikian, tugas akhir ini dinyatakan sah memenuhi prasyarat untuk mencapai sertifikasi Sarjana Teknik PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS WIJAYA PUTRA, Kepada :

NAMA : Danu Nur Irwanto

NPM : 17053030

JUDUL : Identifikasi Berita Hoax Menggunakan Kombinasi Metode
K-Nearest Neighbor (KNN) dan TF-IDF Berbasis Web
Dengan Menggunakan *Framework Codeigniter*

DEWAN PENGUJI TUGAS AKHIR

KETUA : Suryo Atmojo S.Kom, M.Kom (.....)
NIDN : 0709018901

ANGGOTA 1 : Isnaini Muhandhis, S.Kom., M.Kom (.....)
NIDN : 0716118803

Surabaya, 28 Agustus 2021

Mengetahui,

Dekan Fakultas Teknik

Ketua Program Studi Teknik Informatika

Slamet Riyadi, S.T., M.T

NIDN : 07119117101

Isnaini Muhandhis, S.Kom., M.Kom

NIDN : 0716118803

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak ada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu perguruan tinggi, dan juga tidak ada karya atau penilaian yang telah disusun atau didistribusikan oleh orang lain, kecuali yang dirujuk secara tertulis dalam tugas akhir ini. dan yang disebutkan didalam daftar referensi.

Surabaya, 28 Agustus 2021

Danu Nur Irwanto

NPM : 17053030

KATA PENGANTAR

Puji Syukur atas kehadiran ALLAH SWT, yang telah melimpahkan karunia, Rahmat dan Hidayah-Nya, sehingga kami dapat menyelesaikan tugas ini untuk memenuhi persyaratan mencapai derajat sarjana S1.

Dalam menyelesaikan tugas akhir ini penulis banyak mendapat bantuan serta arahan dari berbagai pihak, sehingga penyusunan tugas akhir ini berjalan dengan lancar. Dalam kesempatan kali ini, penulis banyak mengucapkan terima kasih kepada :

1. Bapak H.Budi Endarto, SH, M.Hum, selaku rektor Universitas Wijaya Putra Surabaya yang telah memberikan kesempatan menempuh pendidikan di Universitas Wijaya Putra Surabaya.
2. Bapak Slamet Riyadi, S.T.,M.T selaku Dekan Fakultas Teknik Universitas Wijaya Putra Surabaya.
3. Ibu Isnaini Muhandis, S.Kom., M.Kom. selaku ketua program studi Teknik Informatika Universitas Wijaya Putra.
4. Bapak M. Harist Murdani S.kom, M.Kom. selaku dosen pembimbing yang telah memberikan pengarahan dan bimbingan kepada penulis, sehingga tugas akhir ini dapat terselesaikan.
5. Bapak-ibu dosen yang telah bersedia memberikan masukan berupa kritik dan saran serta guna tentang tugas akhir ini.
6. Ayahanda dan ibunda tercinta beserta seluruh keluarga dan kerabat yang telah memberikan dukungan dan doanya.
7. Teman-teman seperjuangan Teknik Informatika, terima kasih atas dukungan dan bantuannya.
8. Dan semua pihak yang tidak bisa saya sebutkan satu persatu.

Kritik dan saran untuk perbaikan penulisan selanjutnya sangat penulis hargai. Idealnya Tugas Akhir ini dapat bermanfaat bagi semua pembaca.

Surabaya, 28 Agustus 2021

Danu Nur Irwanto
NPM : 17053030

LEMBAR KONSULTASI TUGAS AKHIR

Nama : Danu Nur Irwanto
Program Studi : Teknik Informatika
NPM : 17053030
Telp : 083856764585
Alamat : JALAN MADE BARAT RT.01/RW.02, MADE, SAMBIKEREP, SURABAYA
Judul tugas akhir : Identifikasi Berita Hoax Menggunakan Kombinasi Metode
K-Nearest Neighbor (KNN) dan TF-IDF Berbasis Web
Dengan Menggunakan *Framework Codeigniter*

Konsultasi Ke	URAIAN /MATERI BIMBINGAN	Bab Halaman	Hari Tanggal	Tanda Tangan Dosen Pembimbing
1	Latar belakang, Rumusan masalah, Batasan Penelitian	I	12-04-2021	
2	Tujuan Penelitian, Manfaat Penelitian	I	20-04-2021	
3	Kajian Pustaka	II	26-04-2021	
4	Landasan Teori	II	03-05-2021	
5	Gambaran Umum, perancangan Sistem	III	10-05-2021	
6	Perancangan Aplikasi	III	03-06-2021	
7	Implementasi Sistem	IV	14-06-2021	
8	Implementasi Tampilan, dan uji coba	IV	28-06-2021	
9	Kesimpulan	V	05-07-2021	
10	Saran	V	12-07-2021	

Surabaya, 28 Agustus 2021

Mengetahui,

Ketua Program Studi

Dosen pembimbing

Isnaini Muhandhis, S.Kom., M.Kom

NIDN : 0716118803

M. harist Murdani, S.Kom., M.Sc.

NIDN : 0731078504

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERSETUJUAN	iii
LEMBAR PENGESAHAN	iv
HALAMAN PERNYATAAN	v
KATA PENGANTAR.....	v
LEMBAR KONSULTASI TUGAS AKHIR.....	vi
DAFTAR ISI	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
INTISARI.....	xii
<i>ABSTRACT</i>	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Penelitian.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Kajian Pustaka	5
2.2 Landasan Teori	6
2.2.1 Text Mining	6
2.2.2 TF-IDF	7
2.2.3 K-Nearest Neighbor (KNN)	8
2.2.4 Framework	9
2.2.5 Codeigniter.....	9
2.2.6 PHP	10
2.2.7 Database.....	10
2.2.8 MySql.....	11

2.2.9	Berita	11
2.2.10	Berita HOAX.....	11
BAB III METODE PENELITIAN		12
3.1	Gambaran Umum.....	12
3.2	Alat dan Bahan Pembuatan Tugas Akhir.....	12
3.3	Tahap Perancangan Sistem	13
3.3.1	Pengumpulan Data.....	13
3.3.2	Pra-pemrosesan Data.....	14
3.3.3	Pembuatan Sistem Identifikasi Hoax	18
3.4	Tahap Perancangan Aplikasi	19
3.4.1	Desain Navigasi Menu Untuk User.....	19
3.4.2	Desain Navigasi Menu Untuk Admin	20
3.4.3	Context Diagram	20
3.4.4	Data Flow diagram level 1.....	21
3.4.5	Entity Relationship Diagram (ERD) Conceptual	21
3.4.6	Entity Relationship Diagram (ERD) Physic.....	22
3.4.7	Use Case Diagram Top Level	23
3.4.8	Use Case Diagram Admin	24
3.4.9	Use Case Diagram User	25
3.4.10	Perancangan Antar Muka (Interface).....	25
BAB IV HASIL DAN PEMBAHASAN		27
4.1	Implementasi Sistem Identifikasi Hoax.....	27
4.1.1	Implementasi Pra-pemrosesan data.....	27
4.1.2	Implementasi Perhitungan Jarak Terdekat	30
4.1.3	Implementasi hasil identifikasi.....	31
4.2	Implementasi Tampilan Identifikasi Hoax.....	32
4.2.1	Tampilan Registrasi	32
4.2.2	Tampilan Login	32
4.2.3	Tampilan Dashboard User.....	33
4.2.4	Tampilan Dashboard Admin.....	34
4.2.5	Tampilan Bobot TF-IDF.....	34

4.2.6	Tampilan Hasil KNN	37
4.2.7	Tampilan Hasil Klasifikasi	39
4.2.8	Tampilan Hasil Klasifikasi Berita Valid.....	40
4.2.9	Tampilan Hasil Klasifikasi Berita HOAX	40
4.2.10	Tampilan Hasil Klasifikasi Berita Dataset Tidak Ada	41
4.3	Uji Coba Sistem Identifikasi Hoax	42
4.3.1	Uji Coba Metode TF-IDF	42
4.3.2	Uji Coba Metode KNN	42
4.3.3	Uji Coba Login.....	43
4.3.4	Uji Coba Mengelola Data	43
BAB V	PENUTUP	44
5.1	Kesimpulan.....	44
5.2	Saran.....	44
DAFTAR PUSTAKA.....		44
LAMPIRAN		46
Lampiran I. Source Code.....		46
1)	Source Code Pre-Processing Teks.....	46
2)	Source Code Index Kata.....	49
3)	Source Code Hitung Bobot Kata	50
4)	Source Code Hitung Jarak Euclidean (KNN).....	50
5)	Source Code Jarak Terdekat K=3 (3 buah dataset terdekat).....	51
6)	Source Code Klasifikasi KNN	52
Lampiran II. Surat Pencatatan Ciptaan.....		53

DAFTAR GAMBAR

Gambar 3.1 Tahap Alir Perancangan Sistem	13
Gambar 3.2 Diagram Alir Pra-pemrosesan Data	14
Gambar 3.3 Tahap Tokenizing.....	15
Gambar 3.4 Tahap Case Folding.....	15
Gambar 3.5 Tahap Stopwords Removing	15
Gambar 3.6 Tahap Stemming	16
Gambar 3.7 Diagram Alir Perhitungan TF-IDF.....	16
Gambar 3.8 Diagram Alir system Identifikasi Hoax	18
Gambar 3.9 Navigasi Menu Untuk User.....	19
Gambar 3.10 Navigasi Menu Untuk Admin	20
Gambar 3.11 Context Diagram	20
Gambar 3.12 Data Flow diagram Level 1	21
Gambar 3.13 ERD Conceptual.....	22
Gambar 3.14 ERD Physic	23
Gambar 3.15 Use Case Top Level	24
Gambar 3.16 Use Case Admin.....	24
Gambar 3.17 Use Case User	25
Gambar 3.18 Tampilan Login	25
Gambar 3.19 Tampilan Halaman Utama Admin	26
Gambar 3.20 Tampilan Halaman Utama User	26
Gambar 4.1 Tampilan Registrasi User	32
Gambar 4.2 Tampilan Login Identifikasi Berita Hoax	32
Gambar 4.3 Tampilan Dashboard User.....	33
Gambar 4.4 Tampilan Input Teks Berita Identifikasi	34
Gambar 4.5 Tampilan Hasil Bobot TF-IDF	35
Gambar 4.6 Tampilan Hasil KNN	37
Gambar 4.7 Tampilan Hasil Klasifikasi.....	39
Gambar 4.8 Hasil Klasifikasi Berita Valid.....	40
Gambar 4.9 Hasil Klasifikasi Berita Hoax.....	41
Gambar 4.10 Hasil Klasifikasi Dataset Tidak Ada	41

DAFTAR TABEL

Tabel 2.1 Tabel Penelitian Terdahulu	5
Tabel 3.2 Tabel Hasil Pre-Processing	17
Tabel 3.3 Tabel Perhitungan TF-IDF	17
Tabel 4.1 Tabel Hasil Indexing.....	28
Tabel 4.2 Tabel Hasil Bobot TF-IDF	29
Tabel 4.3 Tabel Hasil Bobot Dokumen.....	30
Tabel 4. 4 Tabel Hasil Perhitungan Dan Pengurutan KNN	31
Tabel 4.5 Tabel Validasi Berita Dataset	31
Tabel 4.6 Tabel Hasil Identifikasi Berita	31
Tabel 4.7 Tabel Uji Coba TF-IDF.....	42
Tabel 4.8 Tabel Uji Coba Metode KNN	42
Tabel 4.9 Tabel Uji Coba Login	43
Tabel 4.10 Tabel Uji Coba Mengelola data	43

INTISARI
SISTEM IDENTIFIKASI BERITA HOAKS MENGGUNAKAN KOMBINASI METODE
TF-IDF & K-NN BERBASIS FRAMEWORK CODEIGNITER

OLEH

DANU NUR IRWANTO

17053030

Internet berkembang menjadi media informasi. Sangat disayangkan apabila informasi yang disampaikan tidak akurat atau bohong (hoax). Yakni berita yang memuat informasi palsu. Saat ini teknologi pendeteksi hoax masih jarang ditemukan.

Untuk mengidentifikasi semua berita hoax secara manual adalah hal yang sangat sulit. Oleh karena itu diperlukan pendekatan secara artificial intelligence (AI) atau kecerdasan buatan. Dimana sebuah mesin dilatih terlebih dahulu (machine learning) untuk menyelesaikan masalah tertentu. Dengan adanya kecerdasan buatan ini diharapkan dapat mengidentifikasi berita hoax lebih cepat dan lebih akurat.

Dalam ilmu komputer, untuk mengidentifikasi kemungkinan berita yang mengandung hoax atau tidak dilakukan dengan menggunakan pemodelan bahasa. Salah satunya adalah bahasa pemodelan dengan algoritma TF-IDF. Kemudian Metode K-Nearest Neighbor (KNN), dipakai untuk mengklasifikasikan berita yang sudah diidentifikasi sebagai berita palsu. Aplikasi ini dapat mengurangi penyebaran berita hoax. Sistem dapat mengidentifikasi berita hoax. Selama kategori berita ada dalam dataset. Untuk akurasi sistem identifikasi berita hoax dipengaruhi jumlah dataset yang ada dan pembahasan yang sesuai dengan topiknya.

Dapat dilakukan penambahan dataset berita hoax secara otomatis dari sumber berita yang sudah teridentifikasi, dari situs-situs resmi berita hoax dan valid yang sudah ada. Agar proses identifikasi menjadi lebih akurat.

Kata Kunci : TF-IDF, K-NN, BERITA, HOAKS, CODEIGNITER

ABSTRACT
HOAX NEWS IDENTIFICATION SYSTEM USING COMBINATION OF TF-IDF & K-NN
METHODS BASED ON CODEIGNITER FRAMEWORK

BY

DANU NUR IRWANTO

17053030

The Internet has evolved into a medium of information. It is unfortunate if the information conveyed is inaccurate or lies (hoax). It's news that contains false information. Today, hoax detection technology is still rarely found. To identify all hoax news manually is a very difficult thing.

Therefore, an artificial intelligence (AI) approach is needed. Where a machine is trained first (machine learning) to solve certain problems. With the existence of artificial intelligence is expected to identify hoax news faster and more accurately. In computer science, to identify possible news that contains hoaxes or is not done using language modeling One of them is a modeling language with TF-IDF algorithm.

Then the K-Nearest Neighbor (KNN) method, used to classify news that has been identified as fake news. This application can reduce the spread of hoax news. The system can identify hoax news. As long as the news category is in the dataset. For the accuracy of hoax news identification systems are influenced by the number of existing datasets and discussions that are in accordance with the topic.

It can be done to automatically add hoax news datasets from identified news sources, from official hoax and valid news sites that already exist. To make the identification process more accurate.

Keywords : TF-IDF, K-NN, NEWS, HOAKS, CODEIGNITER

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesatnya peningkatan inovasi teknologi mempermudah kita untuk mendapatkan informasi dengan cepat dimana saja dan kapan saja (Rozi & Sulistyawati, 2019). Sejujurnya, tidak ada informasi yang tidak dapat diperoleh dengan pandangan kita, ini menunjukkan bahwa semakin pesat laju peningkatan teknologi semakin cepat informasi yang dapat kita peroleh (Hakim, 2018). Kemajuan teknologi saat ini memberikan dampak yang positif, namun juga berdampak negatif. Penyampaian informasi menjadi sangat cepat, dimana setiap orang dengan mudahnya membuat informasi tersebut melalui internet.

Menurut (Faisal Rahutomo, Ingrid Yanuar Risca Pratiwi, & Diana Mayangsari Ramadhani, 2019) Internet telah berkembang menjadi media informasi yang terkenal tentang berbagai hal seperti pencarian berita, lagu, gambar, produk, film dan lain sebagainya. Semua disajikan dalam berbagai sumber seperti artikel, media sosial, dan blog. Sangat disayangkan jika informasi yang disampaikan merupakan informasi yang tidak akurat. Apalagi informasi tersebut merupakan informasi palsu (hoax) yaitu berita yang mengandung informasi palsu dan tidak bertanggung jawab serta bertujuan untuk meyakinkan netizen agar tertipu dan percaya pada berita hoax tersebut.

Menurut (Weddiningrum, 2018) beberapa informasi hoax dibuat oleh individu dan organisasi yang khusus membuat informasi penipuan kemudian menyebarkan ke masyarakat. Diungkapkan oleh situs CNN Indonesia, informasi yang diberikan oleh Kementerian Komunikasi dan Informatika mengungkapkan bahwa ada lebih dari 800.000 situs di Indonesia yang terbukti membuat berita palsu. Untuk mengatasi masalah tersebut, Pemerintah Indonesia mengarahkan Undang-Undang Pasal 28 ayat 1 No. 11 2008 tentang UU ITE. Dalam pasal ini disusun bahwa "Barangsiapa dengan sengaja dan tanpa hak membuat daan

menyebarkan berita bohong dan menipu, diancam dengan pidana kurungan penjara paling lama 6 tahun dan denda paling banyak Rp 1.000.000.000".

Menurut (Weddiningrum, 2018) Semakin maraknya trend hoax maka semakin banyak pula bermunculan ide-ide untuk mewaspadai penyebaran berita hoax tersebut. Banyak tips bermunculan agar tidak terjebak pemberitaan hoax, dan banyak juga media sosial yang memberikan layanan tambahan untuk mengeluhkan konten yang diduga mengandung unsur hoax. Untuk saat ini teknologi pencegahan hoax yang diterapkan dalam sistem deteksi hoax masih jarang ditemukan. Beberapa aplikasi penangkal hoax hanyalah situs web pelaporan berita hoax.

Menurut (Purba, 2019) dapat di lihat untuk mengidentifikasi berita hoax ini secara manual adalah hal yang sangat sulit dan merepotkan. Oleh karena itu dibutuhkan pemanfaatan teknologi seperti *Artificial Inteligent (AI)*. *AI* merupakan mesin (komputer) yang cerdas, dimana sebuah mesin dilatih lebih dahulu (*machine learning*) untuk menyelesaikan masalah tertentu. Kecerdasan buatan ini diharapkan dapat bekerja lebih cepat dan lebih akurat.

Dalam ilmu komputer, untuk mengidentifikasi kemungkinan berita yang mengandung hoax atau tidak dilakukan dengan menggunakan pemodelan bahasa (Purba, 2019). Salah satunya adalah bahasa pemodelan dengan algoritma TF-IDF. Algoritma TF-IDF akan digunakan untuk mengolah setiap kata, memisahkan dan membandingkannya dengan kata-kata yang ada. Nilai kesamaannya akan digunakan untuk mengidentifikasi apakah suatu berita mengandung hoax.

Metode K-Nearest Neighbor (KNN), dipakai untuk mengklasifikasikan berita yang sudah diidentifikasi sebagai berita palsu. K-Nearest Neighbor (KNN) akan menghitung jumlah frekuensi sebuah kata muncul dalam sebuah kalimat. Nilai frekuensi yang tertinggi merupakan klasifikasi kalimat tersebut.

Menurut (Mara Destiningrum & Qadhli Jafar Adrian, SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB, 2017) Dalam merancang identifikasi hoax ini akan dibuat aplikasi berbasis web dengan menggunakan *framework CodeIgniter*. *CodeIgniter* merupakan *framework* yang menggunakan *model, view, Controller* (MVC). CI memiliki banyak *library* untuk menjalankan aplikasi berbasis web, seperti validasi *form*, akses data sehingga kerangka kerja system tidak sulit dibangun. *CodeIgniter* memiliki dokumentasi lengkap.

Menurut (Mara Destiningrum & Qadhli Jafar Adrian, SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB, 2017) *CodeIgniter* memudahkan developer dalam membuat aplikasi web dengan PHP. Terlebih lagi, desain dan rancangan *codeigniter* membuat aplikasi menjadi terkoordinasi dan developer bisa dengan mudah membangun aplikasi. Kerangka web *framework codeigniter* digunakan untuk mengidentifikasi informasi palsu yang berupa teks. Aplikasi ini juga memudahkan para masyarakat dalam membedakan berita hoax atau bukan.

Berdasarkan latar belakang maraknya penyebaran informasi atau berita hoax di atas, maka akan dibuat sebuah tugas akhir dengan judul “Identifikasi Berita Hoax Menggunakan Kombinasi Metode K-Nearest Neighbor (KNN) dan TF-IDF Berbasis Web Dengan Menggunakan *Framework CodeIgniter*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan maka rumusan masalah yang diambil adalah, Bagaimana cara mengatasi penyebaran berita atau informasi hoax berupa teks dengan mengidentifikasi menggunakan kombinasi metode TF-IDF dan K-Nearest Neighbor (KNN) berbasis web dengan menggunakan *framework codeigniter* ? .

1.3 Batasan Penelitian

Tugas akhir memiliki batasan penelitian sebagai berikut :

1. Penelitian ini menggunakan data uji coba yang didapatkan melalui situs Web [“https://data.mendeley.com/datasets/p3hfqr5j3m/1”](https://data.mendeley.com/datasets/p3hfqr5j3m/1).
2. Penelitian ini mengombinasikan metode TF-IDF dan K-Nearest Neighbor (KNN) untuk mengidentifikasi informasi atau berita hoax berupa text.
3. Penelitian ini tidak menerapkan proses *stemming* pada Pra-pemrosesan data.

1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah, merumuskan dan mengimplementasikan sebuah aplikasi yang dapat mengidentifikasi informasi atau berita dalam bentuk teks, apakah mengandung hoax atau tidak dan mengkasifikasikannya dengan kombinasi metode TF-IDF dan K-Nearest Neighbor (KNN) berbasis web dengan menggunakan *framework codeigniter*.

1.5 Manfaat penelitian

Berikut manfaat yang diharapkan dari tugas akhir ini :

1. Mengkombinasikan metode TF-IDF dan K-Nearest Neighbor (KNN) untuk mendapatkan akurasi yang lebih baik, serta kecepatan proses yang lebih cepat.
2. Memberi kontribusi kepada masyarakat untuk mengidentifikasi apakah sebuah informasi atau berita tersebut hoax atau tidak. Masyarakat dapat mengakses melalui layanan web.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Dalam penyusunan skripsi ini, penulis mereferensi penelitian-penelitian terdahulu yang terkait dengan topik latar belakang masalah yang sejenis, sehingga dapat menjadi pendukung untuk pengimplementasian aplikasi sesuai latar belakang masalah skripsi. Adapun beberapa riset tersebut adalah :

Tabel 2.1 Tabel Penelitian Terdahulu

Peneliti	Judul	Metode	Pembahasan
AULIA AFRIZA & JULIO ADISANTOS O, 2018	Metode Klasifikasi Rocchio untuk Analisis Hoax	Klasifikasi Rocchio, TF-IDF	Hasil klasifikasi Rocchio akan dibandingkan dengan Multinomial Naive Bayes. Dimana hasil dari akurasi Rocchio sebesar 83.501% sedangkan Multinomial Naive Bayes sebesar 65.835%.
Weddingrum, 2018	Deteksi Konten Hoax Berbahasa Indonesia Pada Media Sosial Menggunakan Metode Levenshtein Distance	Levenshtein Distance, TF-IDF	metode Levenshtein memiliki akurasi sebesar 70%.
Faisal, Inggrid, & Diana, 2019	Eksperimen Naive Bayes pada Deteksi Berita Hoax Berbahasa Indonesia	Naive Bayes, TF	Berdasarkan hasil uji coba secara statis, metode ini menghasilkan akurasi sebesar 82,6% dan pengujian secara dinamis menghasilkan akurasi sebesar 68,33%.
Mulia Mahendra Alvanof & Rahmat	Analisis dan Deteksi Konten Hoax Pada Media Berita	Multilayer Perceptron (MLP), Naive	Hasil penelitian, metode Random Forest memiliki akurasi terbaik untuk klasifikasi berita, akurasi sebesar 75%.

Triandi, 2020	Indonesia Menggunakan Machine Learning	bayes, Support Vector Machine, Random Forest, TF- IDF	
------------------	---	---	--

Penelitian yang dilakukan adalah mengidentifikasi berita atau informasi hoax dalam bentuk teks. Identifikasi berita atau informasi hoax tersebut menggunakan kombinasi metode TF-IDF dan K-Nearest Neighbor (KNN). Tujuannya untuk mengetahui tingkat akurasi dari kombinasi metode TF-IDF dan K-Nearest Neighbor (KNN) dalam mengklasifikasi berita hoax berupa teks. Dengan dataset berita hoax tahun 2018 yang sudah tersedia dari situs "<https://data.mendeley.com/datasets/p3hfqr5j3m/1>" maka kombinasi metode TF-IDF dan K-Nearest Neighbor (KNN) ini diharapkan dapat memperoleh hasil yang lebih baik dibandingkan dengan penelitian sebelumnya yang dilakukan oleh (Faisal Rahutomo, Ingrid Yanuar Risca Pratiwi, & Diana Mayangsari Ramadhani, 2019).

2.2 Landasan Teori

2.2.1 Text Mining

Menurut (Mulia Mahendra Alvanof & Rahmat Triandi, 2020) Text mining merupakan data mining yang dikhususkan untuk menganalisis data yang berupa teks. Text mining sendiri bertujuan untuk mencari kata-kata yang dapat merepresentasikan dokumen.

1. Text Preprocessing

Menurut (Mulia Mahendra Alvanof & Rahmat Triandi, 2020) Text preprocessing adalah tahap awal dari data mining. Pada tahap ini dilakukan proses pengolahan data agar dokumen / data siap untuk proses klasifikasi. Berikut tahapan Text Preprocessing :

2. Tokenizing

Tokenizing adalah proses membagi atau memotong kalimat menjadi kata-kata dengan menggunakan spasi, contohnya: "Tugas akhir teknik informatika", akan dipecah menjadi: "Tugas", "akhir", "teknik", "informatika".

3. Case folding

Case folding adalah merubah semua huruf menjadi huruf kecil. Misalnya: "HOAX" akan diubah menjadi "hoax", "INFORMASI" menjadi kata "informasi", dan sebagainya.

4. Stopwords removing

Stopwords removing adalah menghapus kata yang tidak penting, seperti preposisi, konjungsi, kata keterangan dan pengganti. Contoh : "oleh","dari","pada", "sebuah", "ke","ini", "yang", "di",dll.

5. Stemming

Menurut (Weddiningrum, 2018) Stemming adalah proses pemisahan kata-kata yang mengandung prefiks atau sufiks untuk menghasilkan kata dasar. Proses stemming berguna untuk memperkecil struktur jumlah suku indeks .

2.2.2 TF-IDF

Menurut (Weddiningrum, 2018) TF-IDF (Term Frequency Inverse Document Frequency) adalah metode untuk menghitung bobot setiap kata pada dokumen dan juga sekumpulan dokumen, bobot tersebut menjadi presentasi penting untuk sebuah kata dalam dokumen. Makin besar bobotnya, menjadikan peran kata tersebut lebih penting dalam presentasi sebuah dokumen. Berikut perhitungan pembobotan kata :

a. Term Frequency (TF)

Merupakan banyaknya kemunculan kata/term di dalam dokumen. Berikut penjelasan persamaan rumus term frequency (Andre Rino Prasetyo, Indriati, & Putra Pandu, 2018).

$$tf(i) = \frac{freq(t_i)}{\sum freq(t)}$$

Ket :

$tf(i)$: nilai jumlah kemunculan kata dalam dokumen.

$freq(t_i)$: berapa kali sebuah kata muncul dalam dokumen.

b. Inverse Document Frequency (IDF)

(Andre Rino Prasetyo, Indriati, & Putra Pandu, 2018) Merupakan banyaknya dokumen yang mengandung kata/token/term t . Berikut penjelasan persamaan Rumus inverse document frequency.

$$idf(i) = \log \frac{|D|}{|\{d: t_i \in d\}|}$$

Ket :

$idf(i)$: nilai Inverse Document Frequency sebuah kata di seluruh dokumen.

$|D|$: Jumlah total dokumen.

$|\{d: t_i \in d\}|$: jumlah dokumen yang mengandung kata (t).

c. TF.IDF Weighting

Dari dua persamaan ini, akan menentukan bobot kata dalam sebuah maupun sekumpulan dokumen, caranya adalah menghitung perkalian antara dua persamaan tersebut. Berikut ini adalah persamaan untuk menentukan bobot (w) sebuah kata (Weddiningrum, 2018).

$$W_{t,d} = W \, tf_{t,d} \times idf_t$$

2.2.3 K-Nearest Neighbor (KNN)

Menurut (Asahar Johar T, Delfi Yanosma, & Kurnia Anggriani, 2016) Algoritma K-Nearest Neighbor (KNN) merupakan metode pengklasifikasian objek berdasarkan data pembelajaran yang paling dekat dengan objek tersebut. KNN adalah algoritma *supervised learning* dimana hasil dari *instance query* baru diklasifikasikan berdasarkan sebagian besar kategori dalam algoritma KNN. Kelas yang terdekat atau jarak terkecil adalah hasil klasifikasi. Untuk menentukan

kedekatan jarak metrik, digunakan rumus jarak *Euclidean*. Berikut persamaan Jarak *Euclidean*:

$$D_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Ket :

D : jarak tetangga terdekat

x : dataset

y : data uji

n : jumlah atribut yang dimiliki oleh data

f : kemiripan antara atribut i, untuk data X dan data Y

i : Atribut yang dimiliki oleh data

2.2.4 Framework

Menurut (Maulana Iqbal Tanjung, 2011) *Framework* adalah suatu kerangka kerja penulisan kode program agar lebih terstruktur. MVC (Model View Control) adalah konsep pemrograman yang memisahkan logika aplikasi pemrograman dari penyajiannya

2.2.5 Codeigniter

Menurut (Maulana Iqbal Tanjung, 2011) Codeigniter adalah streamwork untuk web yang dibuat dalam format PHP. Codeigniter mampu mempercepat pembuatan web, dikarenakan modul dan klas yang dibutuhkan sudah tersedia dan kita hanya perlu menggunakannya kembali pada aplikasi web yang kita buat. Dalam website resminya, menyatakan bahwa CodeIgniter adalah framework PHP yang kuat dan memiliki sedikit bug. Codeigniter dibuat untuk pengembang dengan bahasa pemrograman PHP untuk membuat web berfitur lengkap.

Menurut (Mara Destiningrum & Qadhli Jafar Adrian, Sistem Informasi Penjadwalan Dokter Berbasis WEB Dengan Menggunakan Framework Codeigniter, 2017) pengembang *Framework Codeigniter* adalah Rick Ellis, CEO Ellislab, Inc. Keunggulan *framework Codeigniter* adalah sebagai berikut:

1. *Framework Codeigniter* Gratis (*Open-Source*) dilisensikan di bawah Apache / BSD open-source jadi bersifat gratis.
2. Memiliki Ukuran kecil adalah keuntungan yang pasti jika dibandingkan dengan struktur lain yang sangat besar dalam perkiraan dan membutuhkan aset besar dan lebih jauh lagi dalam pelaksanaan dan kapasitas.
3. Memanfaatkan ide M-V-C Codeigniter adalah ide M-V-C (Model-View-Controller) yang memungkinkan pemisahan antara alasan aplikasi dan lapisan pertunjukan. Dengan PHP, query Mysql, Javascript dan CSS dapat saling dipisah-pisahkan sehingga lebih mudah dalam perbaikan kedepannya atau maintenance.

2.2.6 PHP

Menurut (Kurniawan, 2020) PHP adalah bahasa scripting server-side, artinya dijalankan di server, kemudian outputnya dikirimkan ke client (browser). PHP digunakan untuk membangun aplikasi web, php juga mendukung banyak database seperti Mysql, Oracle, Sybase, PostgreSQL, dll.).

2.2.7 Database

Menurut (Kurniawan, 2020) Database adalah kumpulan bidang terkait, hubungannya dapat ditunjukkan dengan kunci dari setiap file, satu database menunjukkan kumpulan data yang digunakan dalam lingkup perusahaan dan agensi. DB (*Database*) juga dapat didefinisikan sebagai kumpulan data terhubung yang disimpan bersama pada suatu media, diatur menurut skema atau struktur tertentu, dan dengan perangkat lunak untuk melakukan manipulasi untuk penggunaan tertentu.

2.2.8 MySql

Menurut (Kurniawan, 2020) MySQL (My Structured Query Language) adalah: “Sebuah kerangka basis informasi sosial atau Relational Database Management System (RDBMS) yang dapat bekerja dengan cepat dan tidak sulit untuk digunakan. MySQL juga merupakan program penerimaan kumpulan data organisasi, sehingga dapat digunakan dengan baik untuk aplikasi.

2.2.9 Berita

Menurut (Munirul, Ula, Alvanof, & Triandi, 2020) Berita adalah laporan informasi tentang peristiwa dan pendapat yang aktual, penting serta menarik untuk disampaikan kepada publik dalam bentuk surat kabar, radio dan media-media online maupun offline. Salah satu syarat dari berita adalah bahwa berita tersebut harus didasarkan pada keadaan atau peristiwa yang benar-benar terjadi atau fakta yang sebenar-benarnya.

2.2.10 Berita HOAX

Menurut (Firmansyah, 2017) Berita palsu (hoax) adalah artikel berita yang sengaja dibuat untuk menyesatkan pembaca. Ada dua motivasi utama yang menyebabkan beredarnya berita palsu. Pertama adalah uang, artikel berita seolah-olah menjadi virus di media sosial yang dapat menarik pendapatan iklan yang signifikan saat pengguna mengeklik situs aslinya. Hal ini tampaknya menjadi motivasi utama sebagian besar produsen berita palsu yang identitasnya telah terungkap. Motivasi kedua adalah ideologis. Beberapa penyedia berita palsu berusaha untuk memajukan kandidat yang mereka sukai.

BAB III

METODE PENELITIAN

3.1 Gambaran Umum

Dalam pembuatan tugas akhir ini akan dikembangkan sistem yang mengimplementasikan konsep identifikasi berita hoax dengan kombinasi metode yaitu K-Nearest Neighbor (KNN) dan TF-IDF. Dalam sistem tersebut terdapat beberapa proses yang dimulai dari pengumpulan data, pre-processing data, K-Nearest Neighbor (KNN) dan TF-IDF, dan output berupa identifikasi berita hoax. Data penelitian ini diperoleh langsung dari dataset yang sudah tersedia di situs web "<https://data.mendeley.com/datasets/p3hfqr5j3m/1>".

3.2 Alat dan Bahan Pembuatan Tugas Akhir

- 1) Alat Laptop Lenovo G40-45 AMD A8
 - Sistem Operasi Windows 10 Pro 64-Bit
 - Memori Ram 8 GB
 - Hardisk Berkapasitas 500 GB
 - SSD 120 GB
- 2) Perangkat Lunak
 - Microsoft Visual Studio Code
 - XAMPP
 - Navicat Premium
 - Power Designer
- 3) Bahan
 - a) Jurnal referensi tentang Identifikasi Berita Hoax, metode pembobotan kata TF-IDF dan metode klasifikasi K-Nearest Neighbor (KNN).
 - b) Jurnal penelitian terdahulu dengan topik terkait.

3.3 Tahap Perancangan Sistem

Berikut adalah gambar tahap perancangan system :



Gambar 3.1 Tahap Alir Perancangan Sistem

3.3.1 Pengumpulan Data

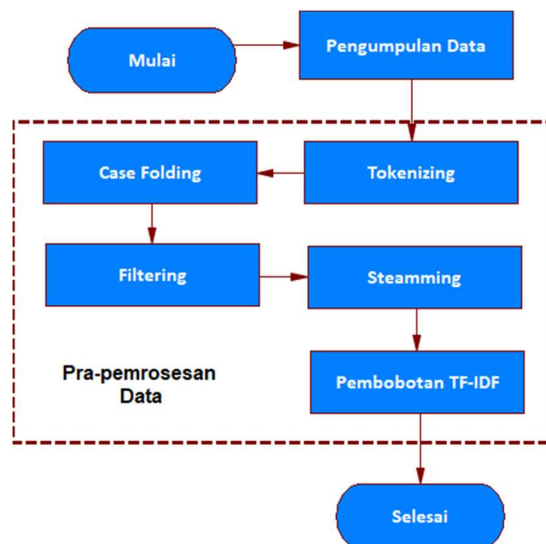
Dalam penelitian ini, data yang digunakan adalah dataset yang sudah tersedia di situs web "<https://data.mendeley.com/>". Dataset tersebut dikumpulkan secara manual melalui mesin pencarian google. Peneliti mengumpulkan dataset di portal-portal berita secara manual melalui internet, dimana menggunakan Google sebagai mesin pencari berita. Caranya adalah peneliti memasukkan kata kunci berita pada Google.

Kemudian Google akan mencari berita yang sama dengan kata kunci yang dimasukkan. Pakar juga harus mengetahui apakah berita yang diberikan oleh Google terinfeksi sebagai berita atau penilaian umum dengan cara identifikasi. Berdasarkan penelitian sebelumnya (Faisal Rahutomo, Inggrid Yanuar Risca Pratiwi, & Diana Mayangsari Ramadhani, 2019) terdapat situs berita yang divalidasi, diantaranya adalah merdeka.com, kompas.com, viva.co.id, metrotvnews.com, republika.co.id, tempo.co, dan tribunnews.com. Sehingga tolak ukur pada kevalidan berita didasarkan pada situs-situs berita tersebut. Dan jika yang ditemukan adalah berita, peneliti menyimpan konten berita tersebut dalam format teks dan pada database untuk kemudian dilakukan penandaan

berita secara manual. Dan Jika bukan berita, maka peneliti akan kembali ke tahap awal proses pengumpulan data.

3.3.2 Pra-pemrosesan Data

Berikut adalah gambar tahap pra-pemrosesan data :



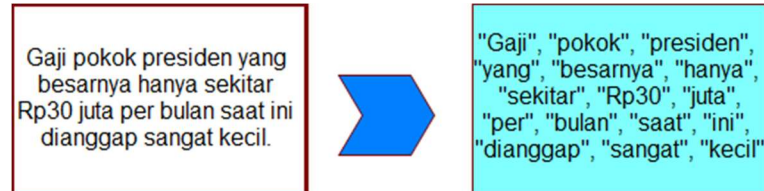
Gambar 3.2 Diagram Alir Pra-pemrosesan Data

Data yang sudah didapatkan merupakan dataset yang tersedia dari situs web "<https://data.mendeley.com/>". Data tersebut akan diolah untuk dijadikan data yang nantinya akan diproses kembali dalam perhitungan TF-TDF. Algoritma TF-TDF ini digunakan untuk memberikan bobot pada setiap kata dalam dokumen. Penggunaan TF-IDF juga digunakan sebagai penyeleksi fitur atau kata yang memegang informasi yang sangat mempengaruhi sebuah dokumen.

Pra-pemrosesan data secara umum memiliki beberapa proses utama yaitu Tokenizing, Case Folding, Filtering, Steaming dan yang terakhir adalah Pembobotan TF-TDF. Tahapan Pra-pemrosesan ini akan menghasilkan kata-kata berita hoax terseleksi dan disimpan dalam dokumen dengan format csv. Dokumen tersebut nantinya akan digunakan sebagai dataset dalam klasifikasi K-Nearest Neighbor (KNN), dengan mencari jarak terdekat antara berita hoax yang sudah di proses dengan berita hoax yang diuji.

1. Tokenizing

Berikut adalah gambar tahap tokenizing :

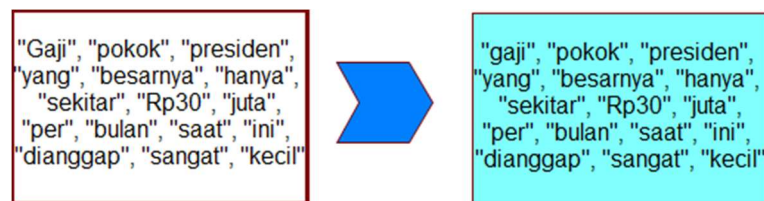


Gambar 3.3 Tahap Tokenizing

Proses membagi atau memotong kalimat menjadi kata-kata dengan menggunakan spasi.

2. Case Folding

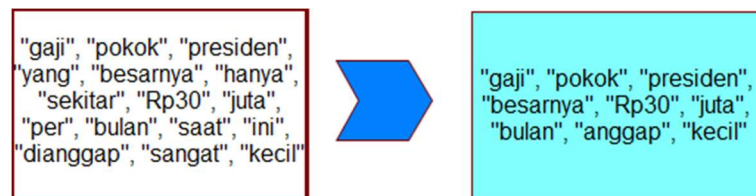
Berikut adalah gambar tahap case folding :



Gambar 3.4 Tahap Case Folding

Gambar diatas adalah contoh dari tahap Case folding, dimana semua karakter dalam teks diubah menjadi huruf kecil.

3. Stopwords removing



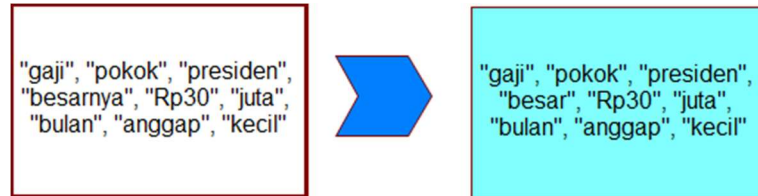
Gambar 3.5 Tahap Stopwords Removing

Berikut adalah gambar tahap stopwords removing :

Dalam langkah ini kata-kata yang bukan kata unik, atau kata-kata yang sering muncul tetapi tidak penting akan dihapus.

4. Stemming

Berikut adalah gambar tahap Stemming :

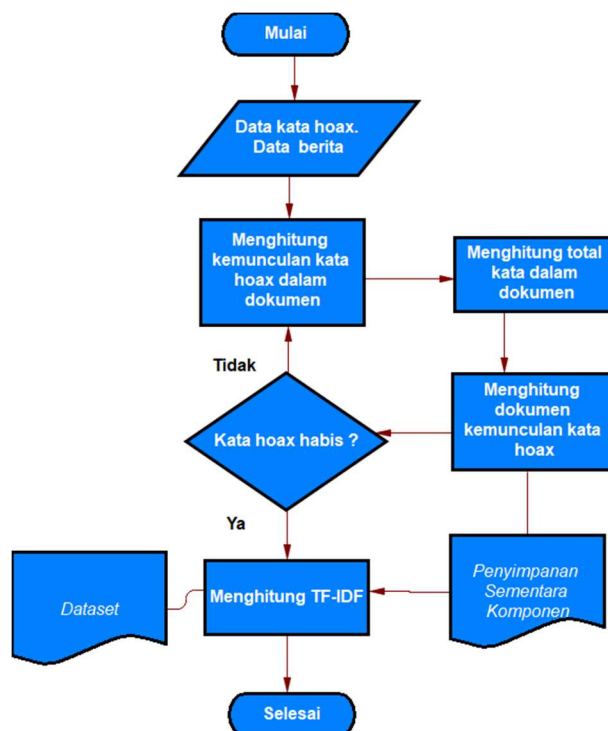


Gambar 3.6 Tahap Stemming

Proses pemisahan kata-kata yang mengandung prefiks atau sufiks untuk menghasilkan kata dasar. Proses stemming berfungsi untuk memperkecil ukuran struktur pengindeksan karena jumlah suku indeks yang berbeda berkurang.

5. Pembobotan TF-IDF

Berikut adalah gambar tahap pembobotan kata :



Gambar 3.7 Diagram Alir Perhitungan TF-IDF

Seperti yang sudah dijelaskan dalam BAB II, TF-IDF merupakan penggabungan dari dua rumus melalui perkalian yaitu rumus TF dan rumus IDF yang memberikan bobot pada sebuah kata dalam dokumen. Dalam proses ini, system akan menghitung banyaknya kata yang muncul dalam dokumen (t_i) dan total seluruh kata dalam dokumen (t) yang merupakan komponen pembentuk rumus Tf. Proses ini juga menghitung banyak dokumen dengan kata yang dicari nilai bobotnya ($d: t_i \in d$) yang merupakan komponen perhitungan Idf.

Laporan tersebut nantinya akan disiapkan kembali secara fisik menggunakan aplikasi Microsoft Excel untuk membuat bobot Tf-Idf dan disimpan bersama dengan kata-kata di arsip kata scam. Kata-kata fabrikasi yang disingkirkan dengan perhitungan Tf-Idf menghasilkan sebuah record dengan desain .csv yang nantinya akan digunakan sebagai informasi target dalam kerangka identifikasi berita bohong. Berikut adalah contoh perhitungan TF-IDF :

Tabel 3.2 Tabel Hasil Pre-Processing

Document	Token
D1	Teknik informatika
D2	Fakultas teknik universitas wijaya putra
D3	Mahasiswa jurusan informatika

Tabel 3.3 Tabel Perhitungan TF-IDF

Token	tf			df	D/df	IDF = Log(D/df)	W = tf * IDF		
	D1	D2	D3				D1	D2	D3
Jurusan	0	0	1	1	3	0.477	0	0	0.477
Teknik	1	1	0	2	1.5	0.176	0.176	0.176	0
informatika	1	0	1	2	1.5	0.176	0.176	0	0.176
Total							0.352	0.176	0.653

3.3.3 Pembuatan Sistem Identifikasi Hoax

Berikut adalah gambar tahap pembuatan sistem identifikasi hoax :



Gambar 3.8 Diagram Alir system Identifikasi Hoax

Setelah melalui tahap pra-proses dataset yang telah dijabarkan pada sub bab sebelumnya, maka tahap selanjutnya adalah membuat sebuah proses identifikasi konten berita yang akan diuji dan dibandingkan dengan kata-kata yang sudah disimpan melalui tahap pra- proses dataset. Dalam pengembangan kerangka ini, persiapan teks juga dilakukan untuk teks berita baru yang dimasukkan. Namun untuk penyimpanan data teks yang dimasukkan ini akan disimpan ke dalam penyimpanan yang berbeda dengan dataset. Karena data masukan baru ini tidak digunakan dalam jangka waktu yang lama seperti dataset, sehingga digunakan penyimpanan sementara yaitu di dalam Array php.

Sistem ini menggunakan hasil perhitungan Tf-Idf untuk menghitung bobot kata dan perhitungan K-Nearest Neighbor (KNN) untuk menghitung jarak antar kata yang dibandingkan. Hasil perhitungan Tf-Idf diperoleh melalui dataset. Untuk perhitungan jarak kata diperoleh dengan membandingkan dua kata yaitu kata pertama diperoleh melalui berita yang masuk dalam sistem dan selanjutnya

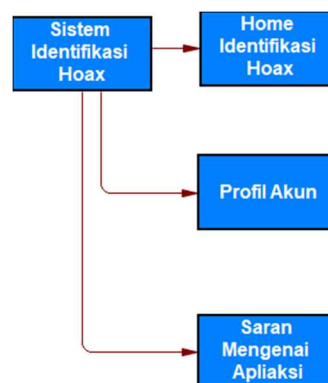
disebut sebagai kata sumber. Kata kedua diperoleh dari dataset yang selanjutnya disebut kata target.

Perbandingan kedua kata tersebut menggunakan perhitungan K-Nearest Neighbor (KNN). Ketika jarak antara kedua kata sudah ditemukan, sistem akan menghitung nilai kemiripan menggunakan rumus *similarity*. Nilai kemiripan kata tersebut kemudian akan dikalikan dengan nilai bobot (Tf-Idf) dan diperoleh hasil akhir dari kata sumber yang diolah. Hasil akhir dari sistem ini adalah hasil perhitungan rata-rata dari semua hasil akhir setiap kata sumber yang dibandingkan.

3.4 Tahap Perancangan Aplikasi

3.4.1 Desain Navigasi Menu Untuk User

Berikut adalah menu untuk User pada system :



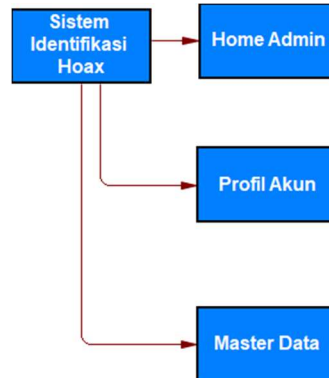
Gambar 3.9 Navigasi Menu Untuk User

Dalam sistem Identifikasi hoax web terdapat beberapa halaman menu untuk user, diantaranya adalah :

- 1) Menu Home Identifikasi Hoax, halaman home untuk identifikasi berita hoax
- 2) Menu Profil Akun, berisi informasi detail mengenai akun
- 3) Menu Saran Mengenai Aplikasi, halaman saran untuk pengembangan aplikasi

3.4.2 Desain Navigasi Menu Untuk Admin

Berikut adalah menu untuk Admin pada system :



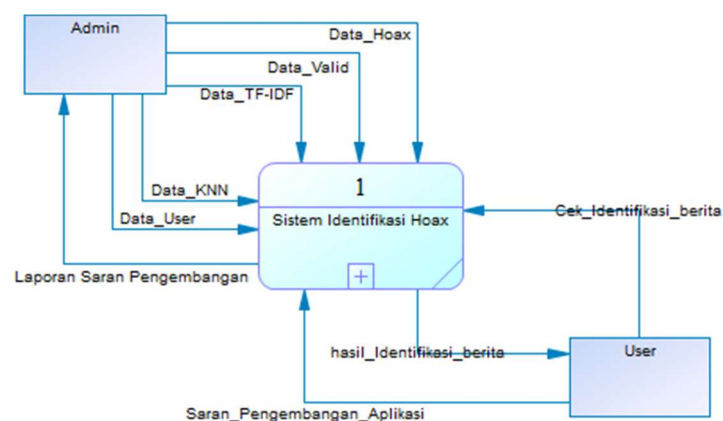
Gambar 3.10 Navigasi Menu Untuk Admin

Dalam sistem Identifikasi hoax web terdapat beberapa halaman menu untuk Admin, diantaranya adalah :

- 1) Menu Home Admin, berisi halaman home untuk admin
- 2) Menu Profil Akun, berisi informasi detail mengenai akun
- 3) Menu Master Data, berisi *create, read, update, delete* data berita hoax dan saran dari user

3.4.3 Context Diagram

Berikut adalah Context Diagram sistem identifikasi hoax :

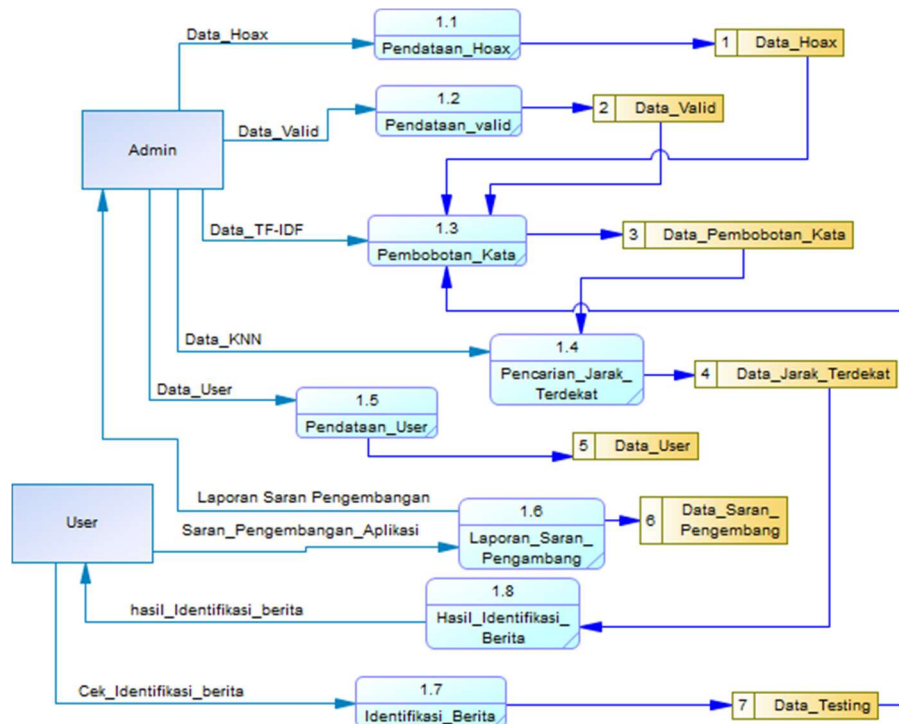


Gambar 3.11 Context Diagram

Pada gambar 3.11 adalah context diagram yang menunjukkan alur dari admin dan user bagaimana sistem tersebut berjalan.

3.4.4 Data Flow diagram level 1

Berikut adalah data flow diagram level 1 :

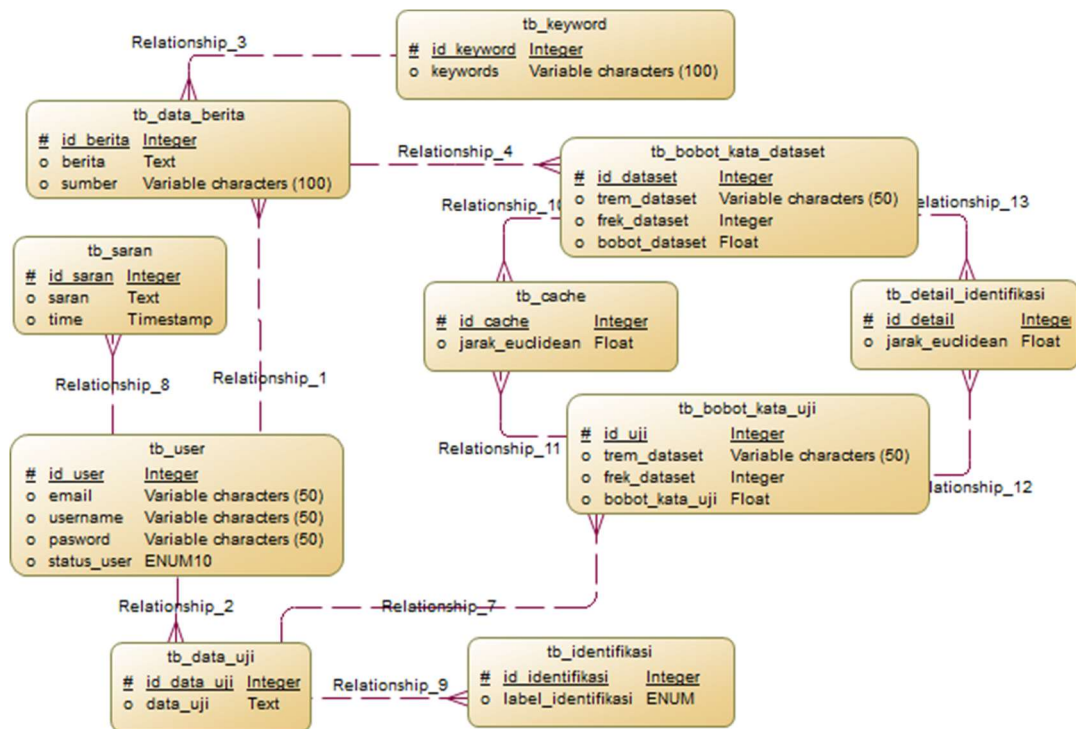


Gambar 3.12 Data Flow diagram Level 1

Pada gambar 3.12 menunjukkan dfd level 1 proses Identifikasi Hoax yg pertama sistem berjalan dimulai dari pendataan hoax, pendataan valid, pembobotan kata, dan perhitungan jarak terdekat antara berita yang di uji dengan dataset, hingga masuk ke hasil identifikasi berita.

3.4.5 Entity Relationship Diagram (ERD) Conceptual

Berikut adalah ERD pada sistem identifikasi hoax :

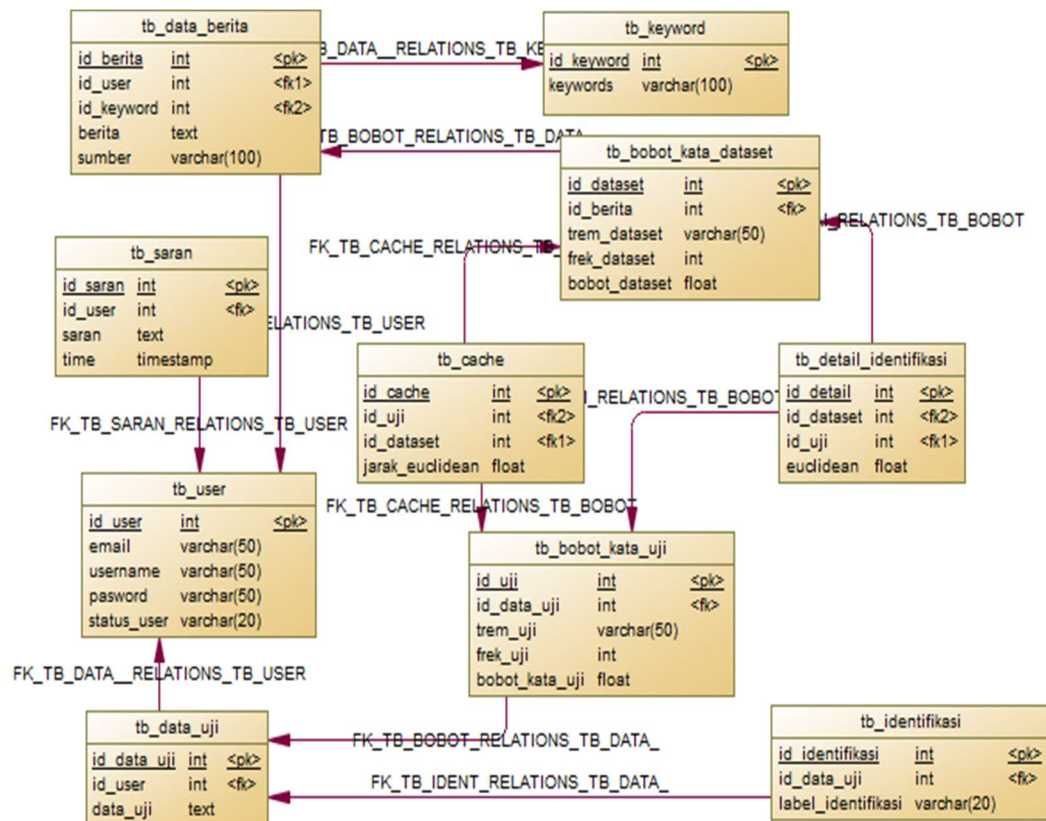


Gambar 3.13 ERD Conceptual

Pada gambar 3.13 menunjukkan erd conceptual proses identifikasi hoax dimulai dari user terdapat id user sebagai primary key tipe data integer, nama lengkap tipe data varchar dengan panjang karakter 20, mail tipe data varchar dengan panjang karakter 20, Username tipe data varchar dengan panjang karakter 20, pswd tipe data varchar dengan panjang karakter 05.

3.4.6 Entity Relationship Diagram (ERD) Physic

Berikut adalah ERD physic :

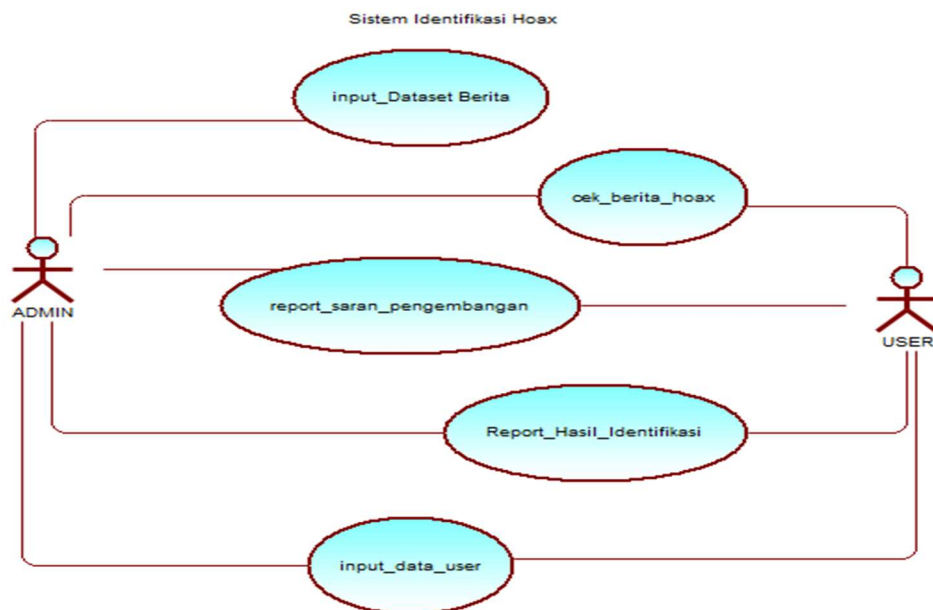


Gambar 3.14 ERD Physic

Pada gambar 3.14 menunjukkan erd physic antara proses identifikasi hoax antara alur pertama sistem berjalan dimana mulai dari pendataan berita untuk dataset , input berita uji dan perhitungan bobot kata dimana menunjukkan sistem database tersebut.

3.4.7 Use Case Diagram Top Level

Berikut adalah use case diagram top level :

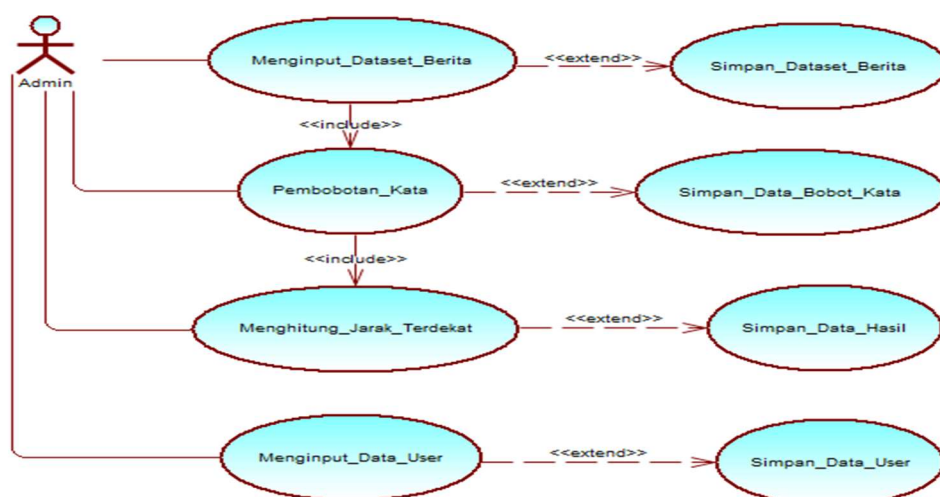


Gambar 3.15 Use Case Top Level

Pada gambar 3.15 menunjukkan top level use case proses identifikasi berita hoax yg pertama sistem berjalan dimulai dari input dataset berita, input data uji, input data user, report hasil identifikasi, hingga report saran pengembang.

3.4.8 Use Case Diagram Admin

Berikut adalah use case diagram admin :

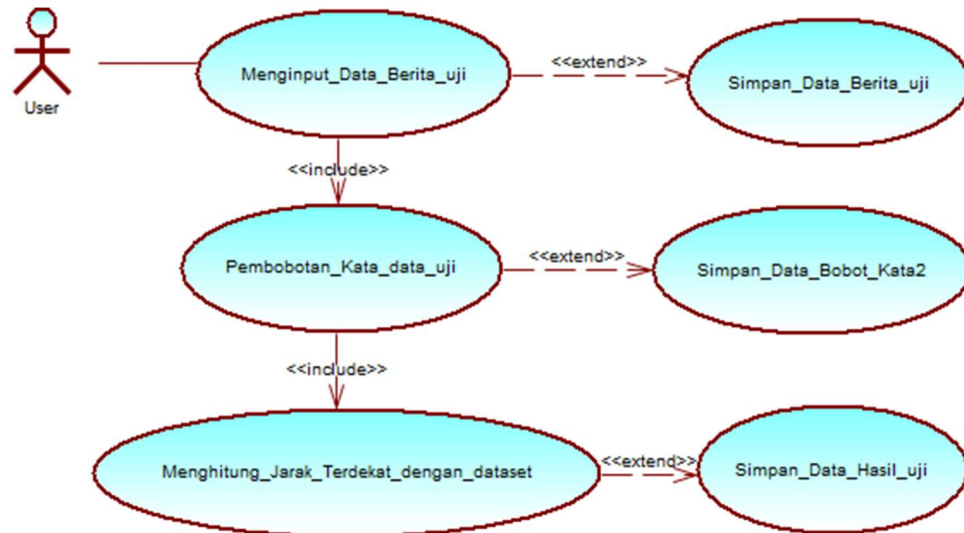


Gambar 3.16 Use Case Admin

Pada gambar 3.16 menunjukkan use case admin mulai dari input dataset berita hingga mengatur data user.

3.4.9 Use Case Diagram User

Berikut adalah use case diagram User :

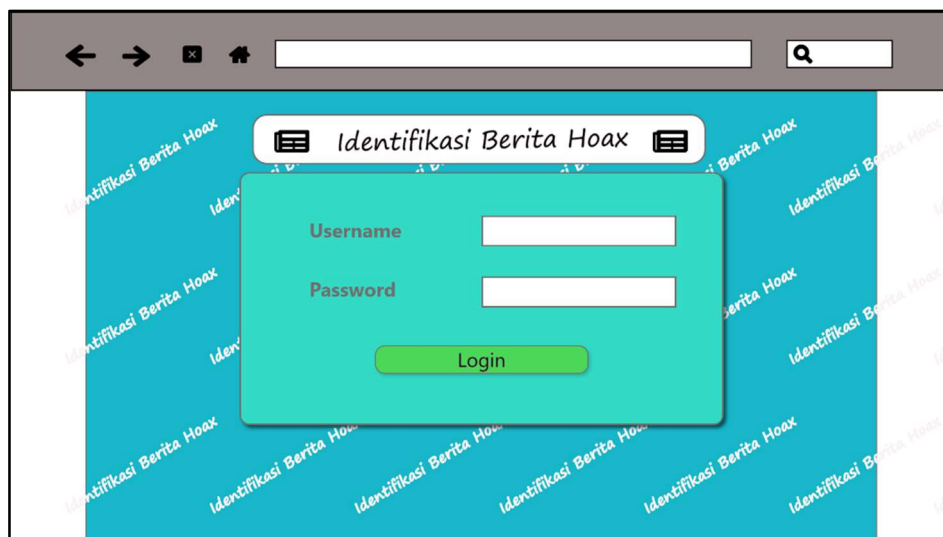


Gambar 3.17 Use Case User

Pada gambar 3.17 menunjukkan use case user mulai dari input berita testing hingga mengirim feedback atau saran pengembangan aplikasi.

3.4.10 Perancangan Antar Muka (Interface)

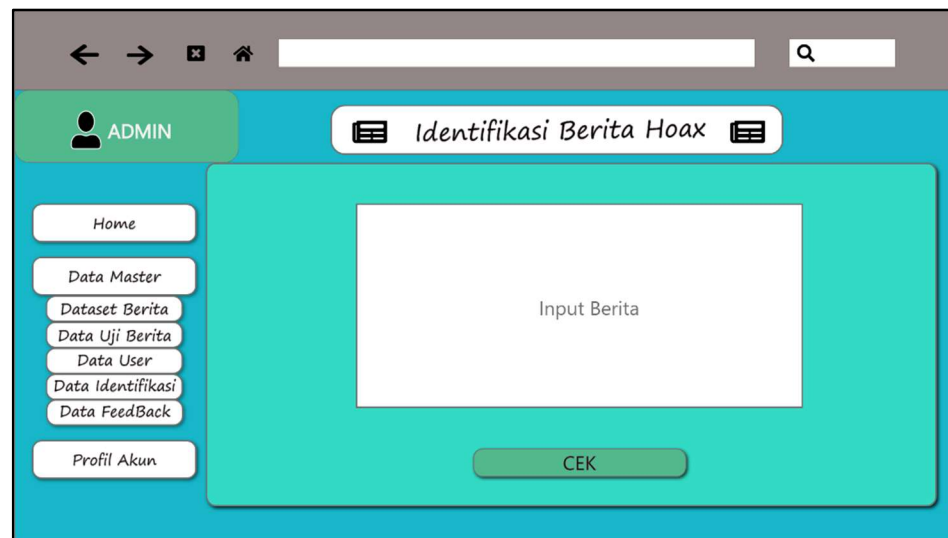
1) Menu Login



Gambar 3.18 Tampilan Login

Pada gambar 3.18 menunjukkan perancangan antar muka tampilan halaman Login sistem Identifikasi Hoax.

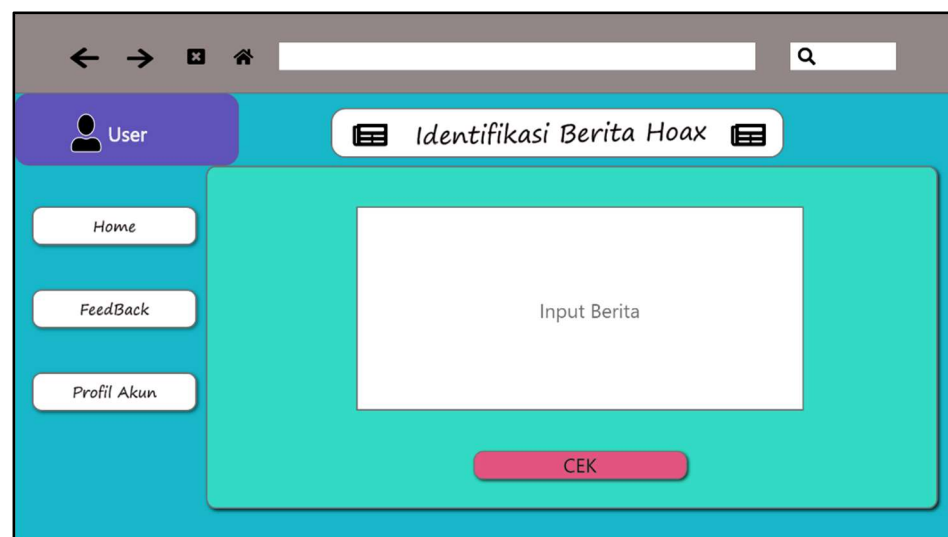
2) Menu Halaman Admin



Gambar 3.19 Tampilan Halaman Utama Admin

Pada gambar 3.19 menunjukkan perancangan antar muka tampilan halaman utama admin sistem identifikasi hoax.

3) Menu Halaman User



Gambar 3.26 Tampilan Halaman Utama User

Pada gambar 3.20 menunjukkan perancangan antar muka tampilan halaman utama user sistem identifikasi hoax.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang implementasi system yaitu pada desain tampilan dan sistem yang telah dirancang pada bab sebelumnya. Implementasi sistem ini meliputi implementasi program.

4.1 Implementasi Sistem Identifikasi Hoax

4.1.1 Implementasi Pra-pemrosesan data

Dataset dan data uji akan diproses terlebih dahulu agar data tersebut bisa diidentifikasi menjadi berita hoax atau valid. Tahap pertama dalam pemrosesan data adalah dengan membuat index kata penting dari dataset dan data uji. Kemudian setelah data index kata penting sudah dibuat maka tahap berikutnya menghitung bobot kata yang sudah di index.

1) Membuat *Indexing* Kata Dataset Dan Data Uji

Pada tahap ini dataset dan data uji berita akan melalui 4 proses. Pertama proses *Tokenizing*, memotong kalimat menjadi kata-kata menggunakan spasi. Ke-dua proses *Case Folding*, semua huruf diubah menjadi huruf kecil. Ke-tiga proses *Stopwords Removing*, kata yang tidak penting atau tidak berhubungan dengan berita akan dihapus. Dan ke-empat proses *Stemming*, merubah semua kata menjadi kata dasar. Berikut adalah tabel hasil indexing :

Tabel 4.1 Tabel Hasil Indexing

DATA UJI	DATASET 1	DATASET 2	DATASET 3	DATASET 4	DATASET 5
lele	iphone	sesuap	pakar	menelusuri	cara
mengandung	memiliki	lele	bantah	lele	liru
kanker	layar	mengandung	ikan	pemicu	menolong
pembudidaya	melengkung	kanker	lele	kanker	stroke
tersinggung	berukuran	hoax	kandung	banjarmasin	detikhealth
kompas	by	fakta	kanker	post	jakarta
com	imam	isu	medan	yayu	masyarakat
wib	baihaki	jakarta	wol	fathilal	menusuk
komentar	nov	jejaring	isu	aneka	jarum
foto	smartphone	sosial	mengandung	menu	menyilet
akibat	galaxy	beredar	kanker	ikan	dikit
bak	round	menyebut	jejaring	rumah	kulit
buah	flex	ikan	sosial	makan	darah
truk	belum	jorok	berita	pecel	konon
terbuka	hasil	suap	mulut	lela	lancar
ngaja	signifikan	daging	menyebarkan	fathilal	aliran
kilogram	pasaran	terkandung	dampak		tersumbat
ikan	harga	julukan	enggan	bahan	akibat
tumpah	laporkan	merujuk	membeli	makanan	tusuk
jalan	menghentikan	sifat		populer	ujung
guizhou	aan	doyan	online	indonesia	jari
china	handphone	mengonsumsi	berhasil	tersebar	telinga
shang	terbaru	limbah	merangkum	kabar	mu
ist	milik	buah	komentar	mengonsumsi	tindakan
penulis	apple	viral	konsumsi	berbahaya	menarik
kontr	kabarkan	internet	rutin	hatan	narik
tor	tinggalan	kotoran	lauk	tubuh	pasien
surabaya	buah	manusia	beredar	kabarkan	rangan
achmad	laporan	jadikan	hidupan	mengandung	mulut
faizal	tulis	pakan	hidup	ratusan	mencong
kanker	bloomberg	budidaya	jorok	kanker	sayang
editorglori	hadir	china	julukan	mari	belum
wadrianto	upgrade	mentara	merujuk	dikit	bukti
suap	bandingkan	habitat	sifat	mengenai	cenderung
daging	s	asli	doyan	beluk	melukai
terkandung	sumber	catfish	mengonsumsi	clarias	penderita
judul	desain	spesies	limbah	spp	empiris
beber	nsor	tangguh	buah	hidup	kuping

2) Menghitung Bobot Kata Dataset dan Data Uji

Pada tahap ini data kata penting pada dataset dan data uji yang sudah melalui proses *indexing* akan dihitung bobot katanya. Proses perhitungan bobot yang digunakan pada tahap ini adalah proses perhitungan TF-IDF. Proses ini digunakan agar dapat dihitung jarak antara dataset dengan data uji. Untuk proses perhitungan TF-IDF sudah dijelaskan pada bab III. Berikut adalah tabel hasil pembobotan TF-IDF :

Tabel 4.2 Tabel Hasil Bobot TF-IDF

TOKEN	DATA UJI	DATASET 1	DATASET 2	DATASET 3	DATASET 4	DATASET 5
lele	23,7894	0	1,6902	3,38039	1,21307	0
mengandung	0,690196	0	15,8596	0,912045	39,649	0
kanker	2,73613	0	0,690196	21,807	2,77833	0
pembudidaya	5,07059	0	5,47227	26,7631	1,38917	0
tersinggung	3,38039	0	0,912045	1,6902	5,07059	0
kompas	2,77833	0	1,94641	1,38917	3,38039	0
com	0,919494	0	5,88763	9,72416	3,38039	0
wib	0,690196	0	0,656936	3,38039	1,6902	0
komentar	1,08814	0	0,845098	5,50974	1,6902	0
foto	0,690196	0	0,367977	2,76078	1,6902	0
akibat	0,991226	0	0,61997	8,2084	23,7894	0
bak	0	0	1,47191	0,845098	0,648803	0
buah	0,869847	0	9,91226	0,735954	1,38917	0
truk	0	0	2,17627	0,690196	1,08814	0
terbuka	0	0	1,08814	4,35254	1,6902	0
ngaja	3,38039	0	0,991226	0,787106	1,6902	0
kilogram	0	0	1,08814	2,42615	17,6629	0
ikan	8,92103	0	1,21307	1,6902	1,08814	0
tumpah	0	0	0,912045	4,1675	1,21307	0
jalanan	0,787106	0	0,912045	9,5674	1,08814	0
guizhou	0	0	1,08814	1,21307	0,411442	0
china	2,42615	0	1,08814	1,08814	1,38917	0
shang	0	0	2,17627	1,6902	0,486076	0
ist	0	0	0,869847	1,08814	4,35254	0
penulis	1,38917	0	0,912045	1,98245	3,67977	0
kontr	0	0	1,21307	1,38917	1,98245	0
tor	0	0	0,991226	1,6902	2,94381	0

surabaya	0	0	0,845098	0,309985	0,912045	0
achmad	0	0	1,21307	1,6902	3,45098	0
faizal	0	0	1,98245	1,6902	1,08814	0
editorglori	0,845098	0	1,08814	2,17627	3,64818	0
wadrianto	0	0	1,21307	1,21307	1,38917	0
suap	0	0	0,735954	0,912045	0,991226	0
daging	1,08814	0	1,38917	0,912045	1,08814	0
terkandung	0,991226	0	0,991226	1,08814	1,6902	0
judul	1,08814	0	1,38917	1,08814	1,6902	0
beber	1,6902	0	1,38917	1,08814	1,6902	0
pembicaraan	0,514105	0	1,38917	0,869847	4,22549	0

4.1.2 Implementasi Perhitungan Jarak Terdekat

Tahap ini merupakan tahap untuk menghitung dan menentukan jarak terdekat antara dataset dengan data uji. Pada proses pertama dalam tahap ini adalah menghitung bobot dokumen atau berita. Kemudian menghitung jarak antara dataset dan data uji dan mengurutkan dataset dari yang terdekat sampai yang terjauh.

1) Menghitung bobot dokumen

Perhitungan bobot dokumen digunakan agar dapat menghitung jarak terdekat. Menghitung bobot dokumen dilakukan dengan cara menghitung total bobot kata yang sudah ter indexing pada tahap sebelumnya. Berikut tabel hasil perhitungan bobot dokumen :

Tabel 4.3 Tabel Hasil Bobot Dokumen

DATA BERITA	BOBOT DOKUMEN
DATA UJI	66,81489
DATASET 1	0
DATASET 2	152,413456
DATASET 3	278,831049
DATASET 4	406,873218
DATASET 5	0

2) Menghitung dan mengurutkan jarak dokumen

Proses ini digunakan untuk mengetahui hasil identifikasi berita. Caranya dengan menggunakan perhitungan KNN. Kemudian hasil perhitungan diurutkan dari yang terdekat sampai yang terjauh. Berikut tabel hasil perhitungan dan pengurutan KNN :

Tabel 4. 4 Tabel Hasil Perhitungan Dan Pengurutan KNN

NO	DATA BERITA	JARAK EUCLIDEAN
1	DATASET 2	102,052
2	DATASET 3	116,782
3	DATASET 4	131,718
4	DATASET 1	209,06
5	DATASET 5	209,06

4.1.3 Implementasi hasil identifikasi

Tahap ini merupakan tahap terakhir dalam implementasi system identifikasi berita hoax. Pada tahap ini menampilkan hasil identifikasi dari data uji berita. Hasil identifikasi didapatkan melalui keterangan validasi berita terdekat. Ada 3 hasil identifikasi yaitu HOAX, VALID, UNIDENTIFIED. Berikut tabel hasil identifikasi berita :

Tabel 4.5 Tabel Validasi Berita Dataset

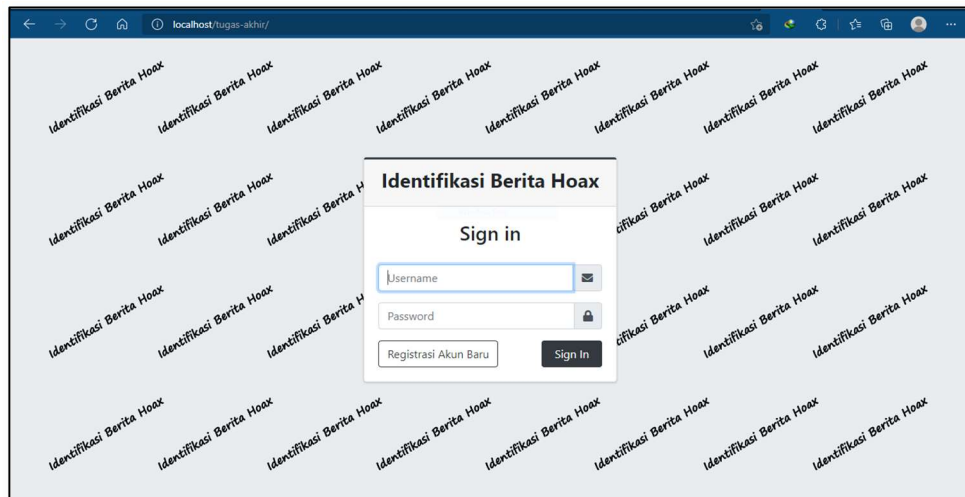
NO	DATASET BERITA	VALIDASI BERITA
1	DATASET 2	HOAX
2	DATASET 3	VALID
3	DATASET 4	HOAX
4	DATASET 1	VALID
5	DATASET 5	VALID

Tabel 4.6 Tabel Hasil Identifikasi Berita

NO	DATA UJI BERITA	HASIL IDENTIFIKASI
1	DATA UJI	HOAX

4.2 Implementasi Tampilan Identifikasi Hoax

4.2.1 Tampilan Registrasi



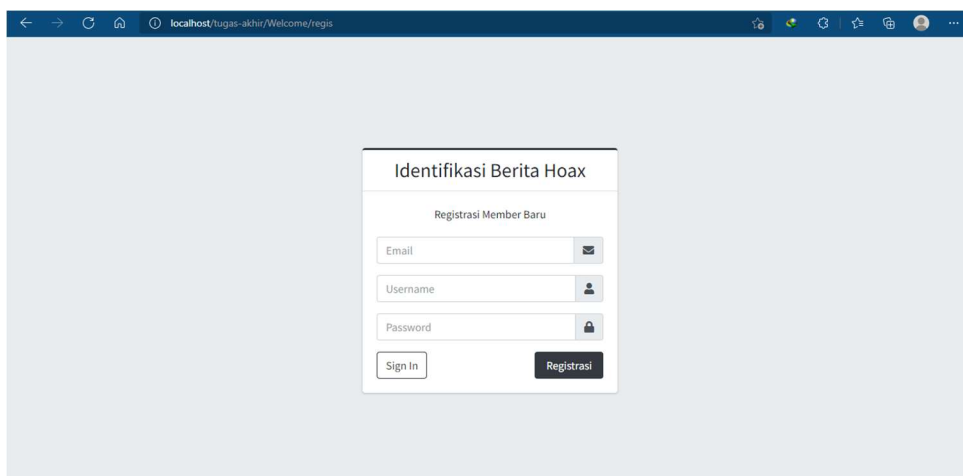
Gambar 4.1 Tampilan Registrasi User

Berikut adalah tampilan Registrasi User :

Pada gambar 4.1 merupakan tampilan registrasi user. Disini user melakukan registrasi agar bisa menggunakan sistem identifikasi berita hoax. Untuk mendaftar user memasukkan email, username, dan password. Kemudian menekan tombol registrasi. Registrasi berhasil apabila setiap kolom sudah terisi dengan benar, dan username yang sudah di inputkan tidak ada yang sama dalam database.

4.2.2 Tampilan Login

Berikut adalah tampilan Login :

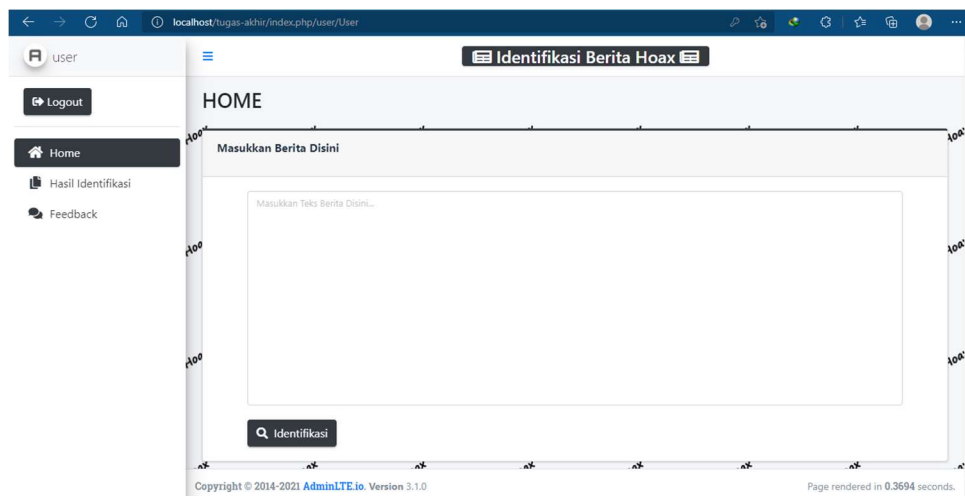


Gambar 4.2 Tampilan Login Identifikasi Berita Hoax

Pada gambar 4.2 merupakan tampilan login identifikasi berita hoax. Ini merupakan tampilan utama pada aplikasi identifikasi berita hoax. Disini user dapat login sebagai member atau admin tergantung dari status user tersebut. Setelah user melakukan registrasi maka suser bisa login, dan user dapat mengidentifikasi berita hoax.

4.2.3 Tampilan Dashboard User

Berikut merupakan tampilan dashboard User :

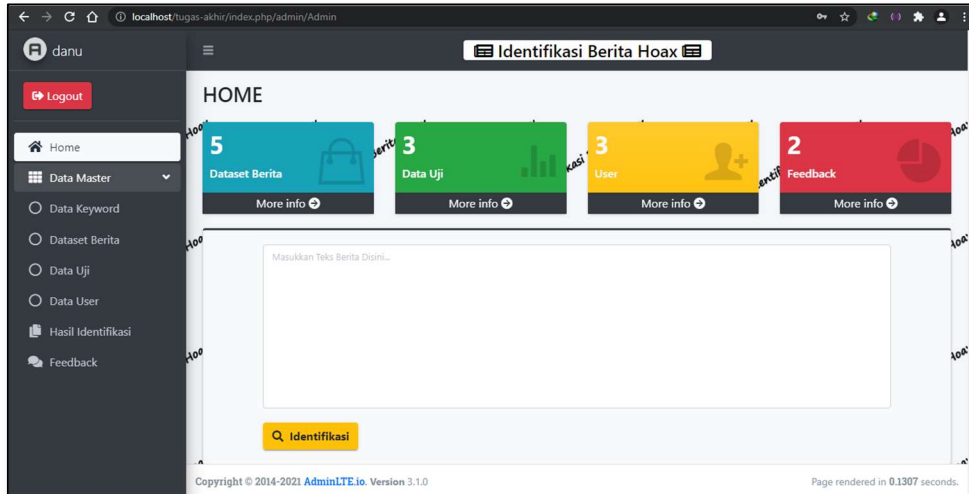


Gambar 4.3 Tampilan Dashboard User

Pada gambar 4.3 merupakan tampilan dashboard untuk user. Disini user dapat menginputkan berita hoax yang akan di identifikasi. Berita tersebut dapat diinputkan ke dalam kolom yang sudah tersedia. Setelah diinputkan user akan mendapatkan hasil identifikasi dari berita tersebut.

4.2.4 Tampilan Dashboard Admin

Berikut adalah tampilan input berita yang akan diidentifikasi :



Gambar 4.4 Tampilan Input Teks Berita Identifikasi

Pada gambar 4.1 merupakan tampilan input berita untuk identifikasi. Di sini admin bisa menginputkan berita yang mau diidentifikasi untuk mengetahui berita tersebut merupakan berita hoax atau bukan. Admin dapat menginputkan berita tersebut pada kolom input, setelah di input admin bisa menekan tombol Identifikasi untuk mengetahui hasilnya.

4.2.5 Tampilan Bobot TF-IDF

Berikut adalah tampilan hasil bobot TF-IDF :

No	Term	Frekuensi	Bobot
1	sesuap	1	0.69897
2	lele	16	0
3	mengandung	1	0
4	kanker	6	0
5	hoax	1	0.69897
6	fakta	3	1.19382
7		8	0

Gambar 4.5 Tampilan Hasil Bobot TF-IDF

Pada gambar 4.2 merupakan tampilan halaman hasil proses TF-IDF. Disini setiap berita yang sudah diinputkan akan dilakukan pra-pemrosesan data. Yaitu Tokenizing, Case folding, Filtering, dan Steaming. Kemudian dilanjutkan proses pembobotan TF-IDF. Hal ini dilakukan untuk inisialisasi berita dengan cara menghitung jumlah kata yang sama (*Trem Frekuensi*), dan dihitung bobotnya. Berikut adalah source codenya :

1) Source code Pra-Pemrosesan data

```
|1. public function buatIndexDataset(){
|2.     $this->M_Kata_Dataset->del_data();
|3.     $berita = $this->M_Dataset->getAll();
|4.     foreach($berita as $b){
|5.         $idDoc = $b['ID_BERITA'];
|6.         $isiBerita = $b['BERITA'];
|7.         // menerapkan preprocessing teks
|8.         $teksBerita = $this->M_Metode->preProcessing($isiBerita);
```

Pada source code diatas baris ke-2 menghapus semua data index di tabel bobot kata untuk memperbarui data index kata yang akan dihitung. Baris ke-3 memanggil semua data berita yang akan di proses. Baris ke-8 adalah tahap pra-pemrosesan data.

2) Source code menghitung jumlah kata (*Trem Frekuensi*).

```
|1. $arberita = explode(" ", trim($teksBerita));
|2. $jumlah = array_count_values($arberita);
|3. foreach($jumlah as $j => $j_value){
|4.     if($j != ""){
|5.         $row = $this->M_Kata_Dataset->getAnd($idDoc,$j);
|6.         $row_count = count($row->result_array());
|7.         $data = array(
|8.             'ID_BERITA' => $idDoc,
|9.             'TREM_DATASET' => $j,
|10.            'FREK_DATASET' => $j_value);
|11.         if($row_count > 0){
|12.             $this->M_Kata_Dataset->updateData($data);
|13.         }else{
|14.             $this->M_Kata_Dataset->add($data); }}}
```

Pada source code di atas adalah tahap awal dari metode kata TF-IDF yaitu menghitung frekuensi kata yang sama. Baris ke-1 berita yang sudah di proses dirubah menjadi array, baris ke-2 menghitung jumlah kata yang sama, baris ke-11 mengecek data yang sama jika ada yang sama maka akan di update jika tidak ada yang sama maka akan ditambah data baru.

3) Source code menghitung bobot kata.

```
|1. public function hitungBobotDataset(){
|2. $bobot = $this->M_Kata_Dataset->getAll();
|3.     foreach($bobot as $b){
|4.         $idKata = $b['ID_KATA_DATASET'];
|5.         $tremData = $b['TREM_DATASET'];
|6.         $TF = $b['FREK_DATASET'];
|7.         $D = $this->M_Kata_Dataset->getNDoc();
|8.         $df = $this->M_Kata_Dataset->getDF($tremData);
|9.         $W = $TF * log10($D/$df);
|10.        $data2 = array(
|11.            'ID_KATA_DATASET' => $idKata,
|12.            'BOBOT_KATA_DATASET' => $W
|13.        );
|14.        $this->M_Kata_Dataset->updateBobot($data2);
|15.    }
|16. }
```

Pada source kode diatas adalah tahap selanjutnya dari metode TF-Idf yaitu menghitung bobot kata. Baris ke-2 memanggil semua data yang sudah diproses pada tahap awal. Baris ke-7 menghitung jumlah dokumen, baris ke-8 menghitung jumlah dokumen dengan trem yang sama. Baris ke-9 menghitung bobot kata TF-IDF ($W = TF \times \log_{10}(D/df)$). Setelah proses hitung selesai kemudian data akan di update.

4.2.6 Tampilan Hasil KNN

Berikut adalah tampilan hasil hitung jarak KNN :

No	Data Training	Status Data	Euclidean Distance
1	Sesup Lele Mengandung 3.000 S....	VALID	4
2	Menelusuri Kebenaran Lele Pemi....	VALID	4.67788
3	Konsumsi Ikan Lele Memicu Kank....	HOAX	5.46682
4	Benarkah Sesup Lele Mengandun....	HOAX	6.41567
5	Pakar Bantah Ikan Lele Kandung....	VALID	8.56289

Gambar 4.6 Tampilan Hasil KNN

Pada gambar 4.3 merupakan tampilan halaman hasil KNN. Di sini akan diambil 5 buah dataset untuk klasifikasi hasil akhir identifikasi atau $K = 5$. Proses pengambilan dataset tersebut ditentukan melalui hasil perhitungan KNN yaitu *Euclidean Distance*. Perhitungan tersebut digunakan untuk menentukan jarak terdekat yaitu hasil nilai terkecil. Proses ini nanti akan digunakan untuk klasifikasi KNN. Berikut adalah source codenya :

1) Source code hitung jarak Euclidean (KNN).

```

1. public function hitungKNN($id){
2. $this->M_Chace->del_data();
3. if($id == null){ $docUji = $this->M_Bobot_Uji->getDistinct();
4. }else{ $docUji = $this->M_Bobot_Uji->getID($id); }
5. $docDistinct = $this->M_Chace->getDistinct();
6.     foreach($docUji as $dUji){
7.         $idD = $dUji['ID_DATA_UJI'];
8.         foreach($docDistinct as $DD){
9.             $docID = $DD['ID_BERITA'];
10.            $tremUji = $this->M_Chace->getAnd($idD);
11.            $wUji = 0;
12.            $wDataset = 0;
13.            foreach($tremUji as $t){
14.                $arTrem = $t['TREM_UJI'];
15.                $uji = $t['BOBOT_KATA_UJI'];
16.                $wUji = $wUji + $uji;
17.                $data = array(
18.                    'ID_BERITA' => $docID,

```

```

|19.             'TREM_UJI' => $arTrem );
|20.     $tremData = $this->M_Chace->getBobotTrem($data);
|21.     $set = $tremData['BOBOT_KATA_DATASET'];
|22.     $wDataset = $wDataset + $set; }
|23.         $temp = $wDataset - $wUji;
|24.         $pangkat = pow($temp,2);
|25.         $p_vektor = sqrt($pangkat);
|26.         $data2 = array(
|27.             'ID_DATA_UJI' => $idD,
|28.             'ID_BERITA' => $docID,
|29.             'JARAK_EUCLIDEAN' => $p_vektor );
|30.         $this->M_Chace->add($data2);
|31.     }}
|32.     $this->M_Metode->lanjutKNN($id); }

```

Pada source code di atas baris ke 3-4 memanggil data testing yang akan diidentifikasi. Baris ke-5 memanggil semua dataset yang sudah melalui proses metode TF-IDF. Baris ke 10-16 menghitung bobot dokumen data testing. Baris ke 20-22 menghitung semua bobot dataset dengan trem yang sama dengan data testing. Baris ke 23-25 hitung jarak *euclidean* $\sqrt{\sum (dataset - data\ training)^2}$.

2) Source code mengurutkan jarak tetangga terdekat.

```

|1. public function lanjutKNN($id){
|2.     $this->M_Identifikasi_detail->del_data();
|3.     if($id == null){
|4.         $idTest = $this->M_Bobot_Uji->getDistinct();
|5.     }else{
|6.         $idTest = $this->M_Bobot_Uji->getID($id); }
|7.     foreach($idTest as $t){
|8.         $idDocUji = $t['ID_DATA_UJI'];
|9.         $cache = $this->M_Chace->getLimit($idDocUji);
|10.        foreach($cache as $c){
|11.            $data = array(
|12.                'ID_DATA_UJI' => $idDocUji,
|13.                'ID_BERITA' => $c['ID_BERITA'],
|14.                'JARAK_EUCLIDEAN' => $c['JARAK_EUCLIDEAN'] );
|15.        $cek = $this->M_Identifikasi_detail->cekData($data);
|16.        if($cek > 0){ $this->M_Identifikasi_detail->updateData($data);
|17.        }else{$this->M_Identifikasi_detail->add($data); }}}
|18.    $this->M_Chace->del_data();
|19.    $this->M_Metode->knnClass(); }

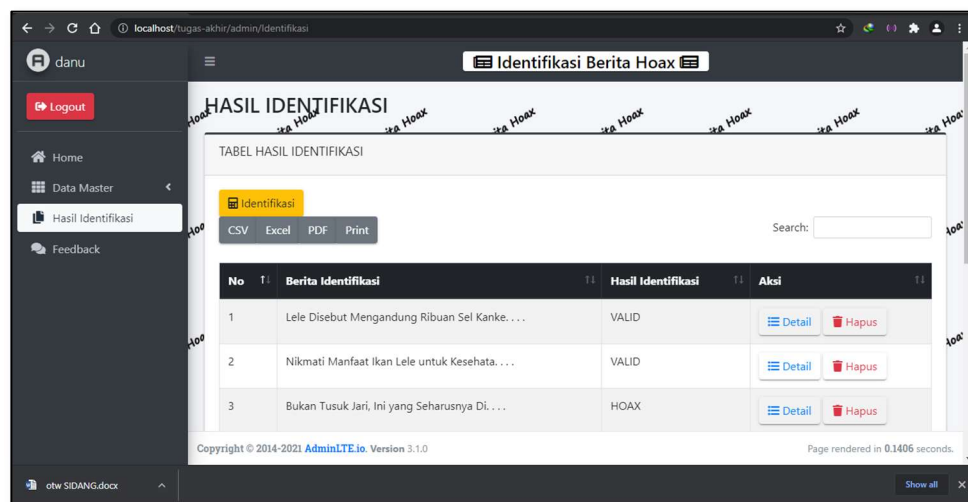
```

Pada source code diatas merupakan lanjutan dari metode KNN yaitu mengambil tetangga terdekat. Setelah proses hitung jarak *euclidean*, maka akan diambil

sejumlah tetangga terdekat sebanyak 5 buah dataset. Pada baris ke 3-6 memanggil semua dataset yang sudah melalui proses hitung jarak euclidean (KNN). Baris ke-9 memanggil 5 tetangga terdekat dari dataset.

4.2.7 Tampilan Hasil Klasifikasi

Berikut adalah tampilan hasil Klasifikasi KNN :



Gambar 4.7 Tampilan Hasil Klasifikasi

Pada gambar 4.6 merupakan tampilan halaman hasil klasifikasi KNN. Di sini 5 buah dataset yang sudah diproses akan digunakan untuk klasifikasi. Proses klasifikasi ini menggunakan dataset terdekat atau nilai terkecil untuk menentukan hasil identifikasi. Hasil ini akan menentukan data testing yang diinputkan merupakan berita hoax atau bukan. Berikut adalah source codenya :

3) Source code klasifikasi KNN.

```

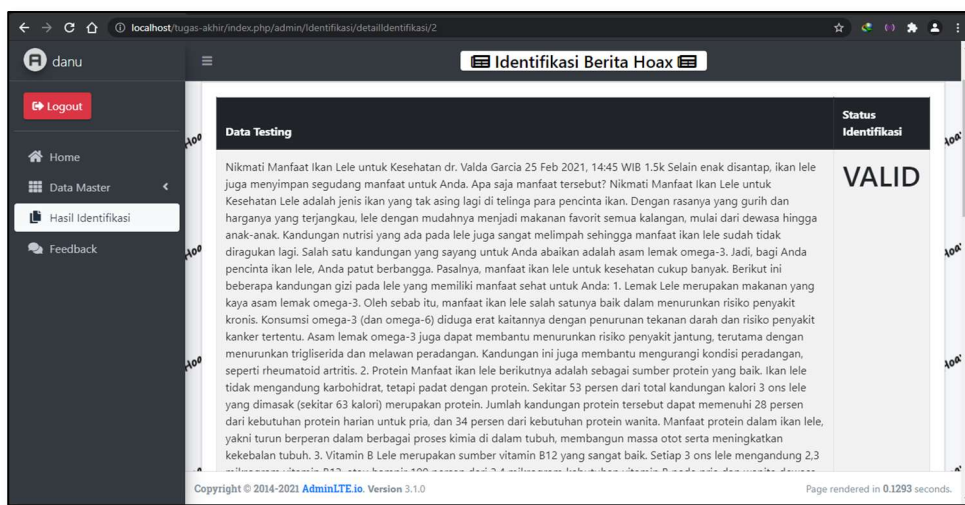
1. public function knnClass(){
2.     $detail = $this->M_Identifikasi_detail->getDistinct();
3.     foreach ($detail as $det) {
4.         $id = $det['ID_DATA_UJI'];
5.         $labH = $this->M_Identifikasi_detail->getById($id);
6.
7.         $data = array(
8.             'ID_DATA_UJI' => $id,
9.             'LABEL_IDENTIFIKASI' => $labH['STATUS_BERITA'] );
10.        $cek = $this->M_Identifikasi->cekData($data);
11.        if($cek > 0){ $this->M_Identifikasi->updateData($data);
12.        }else{ $this->M_Identifikasi->add($data); }}}

```

Source code di atas merupakan tahap akhir yaitu klasifikasi KNN. Baris ke-2 memanggil semua 5 tetangga terdekat dari proses sebelumnya. Baris ke 5 melakukan klasifikasi dengan mengambail jarak terdekat atau terkecil. Baris ke 9 adalah hasil status berita dari hasil terdekat.

4.2.8 Tampilan Hasil Klasifikasi Berita Valid

Berikut adalah tampilan hasil identifikasi berita Valid :

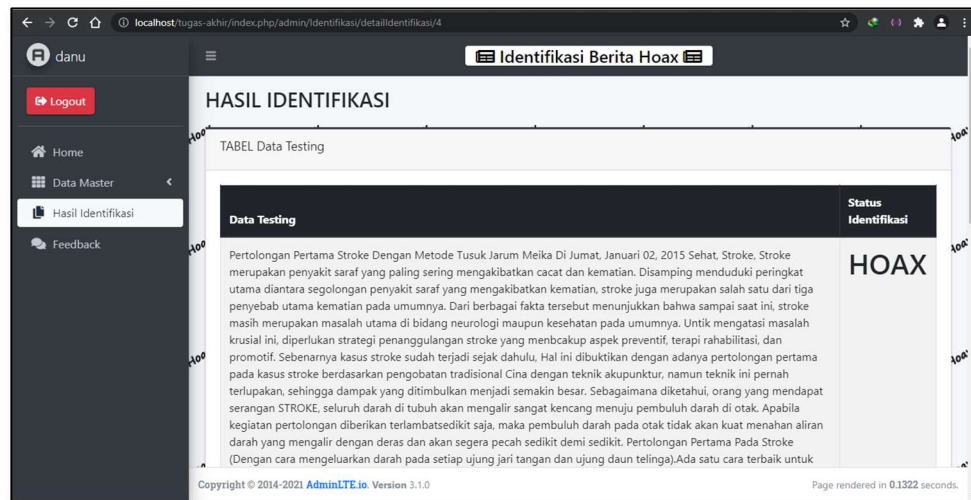


Gambar 4.8 Hasil Klasifikasi Berita Valid

Gambar diatas merupakan berita hasil identifikasi. Berita tersebut sudah melewati proses yang dijabarkan sebelumnya. Dan hasil dari identifikasi berita tersebut adalah VALID.

4.2.9 Tampilan Hasil Klasifikasi Berita HOAX

Berikut adalah tampilan hasil identifikasi berita Hoax :

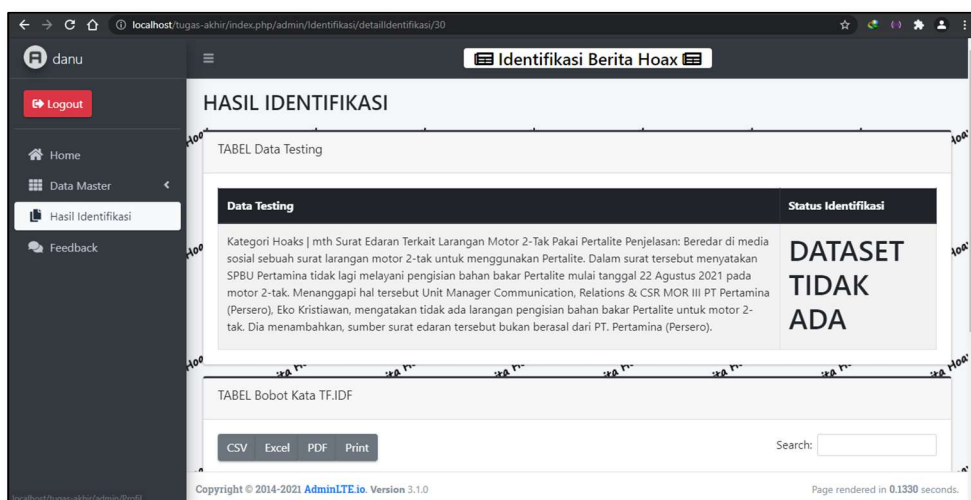


Gambar 4.9 Hasil Klasifikasi Berita Hoax

Gambar diatas merupakan berita hasil identifikasi. Berita tersebut sudah melewati proses yang dijabarkan sebelumnya. Dan hasil dari identifikasi berita tersebut adalah HOAX.

4.2.10 Tampilan Hasil Klasifikasi Berita Dataset Tidak Ada

Berikut adalah tampilan hasil identifikasi berita Dataset tidak ada :



Gambar 4.10 Hasil Klasifikasi Dataset Tidak Ada

Gambar diatas merupakan berita hasil identifikasi. Berita tersebut sudah melewati proses yang dijabarkan sebelumnya. Dan hasil dari identifikasi berita tersebut adalah DATASET TIDAK ADA. Hasil tersebut bisa terjadi Karena, belum ada dataset untuk mengidentifikasi berita tersebut.

4.3 Uji Coba Sistem Identifikasi Hoax

4.3.1 Uji Coba Metode TF-IDF

Hasil pengujian metode TF-IDF dapat di lihat pada tabel 4.1

Tabel 4.7 Tabel Uji Coba TF-IDF

Uji Coba	Hasil Perbandingan antara hasil pada sistem dan perhitungan manual
Mengambil kata penting	Hasil Sesuai
Menghitung jumlah kata yang sama	Hasil Sesuai
Pembobotan kata	Hasil Sesuai
Pembobotan dokumen	Hasil Sesuai

4.3.2 Uji Coba Metode KNN

Hasil pengujian metode KNN dapat di lihat pada tabel 4.2

Tabel 4.8 Tabel Uji Coba Metode KNN

Uji Coba	Hasil Perbandingan antara hasil pada sistem dan perhitungan manual
Menghitung jarak euclidean (KNN)	Hasil Sesuai
Menentukan jarak terdekat	Hasil Sesuai
Klasifikasi KNN	Hasil Sesuai

4.3.3 Uji Coba Login

Hasil pengujian login dapat dilihat pada tabel 4.3

Tabel 4.9 Tabel Uji Coba Login

Skenario	Yang Diharapkan	Pengamatan	Kesimpulan
User masuk menggunakan username dan password	User dapat masuk sesuai username dan password	User dapat masuk sesuai username dan password	Berhasil
Admin mengakses menu admin	Pengguna dapat mengakses sesuai rolenya	Hanya admin yang dapat mengakses menu admin	Berhasil

4.3.4 Uji Coba Mengelola Data

Hasil mengelola data dapat dilihat pada tabel 4.4

Tabel 4.10 Tabel Uji Coba Mengelola data

Skenario	Yang Diharapkan	Pengamatan	Kesimpulan
Melakukan Input dataset	Hanya admin yang dapat menginputkan dataset	Tidak ada yang bisa input dataset kecuali admin	Berhasil
Melakukan input data Training atau uji identifikasi	Semua user bisa melakukan pengujian identifikasi	Semua user bisa melakukan pengujian identifikasi	Berhasil
Hitung otomatis TF-IDF, dan KNN	Admin tidak perlu melakukan hitung manual	Hasil hitung dapat dilakukan otomatis	Berhasil

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan dari penelitian yang sudah dilakukan, dapat disimpulkan hasilnya sebagai berikut :

1. Aplikasi ini dapat mengurangi penyebaran berita hoax. Dengan mengkombinasikan metode KNN dan TF-IDF, system yang dibuat dapat mengklasifikasikan berita hoax dan valid.
2. Sistem dapat mengidentifikasi berita hoax. Selama kategori berita ada dalam dataset.
3. Untuk akurasi sistem identifikasi berita hoax dipengaruhi jumlah dataset yang ada dan pembahasan yang sesuai dengan topiknya. Topik yang sesuai adalah isi dari berita dengan pembahasan yang sama tapi memiliki tujuan yang berbeda, contoh topik berita tentang ikan lele yang menyebabkan kangker. Untuk isi berita yang valid akan menjelaskan kenapa lele bisa menyebabkan kangker dan kenapa hal tersebut bisa terjadi, sedangkan untuk isi berita yang Hoax akan menjelaskan bahwa lele memang dapat menyebabkan kangker.

5.2 Saran

Dari hasil implementasi aplikasi dan analisa yang telah dilakukan, dapat diberikan saran perbaikan dan pengembangan dalam penelitian ini :

1. Dapat dilakukan pengembangan dan perbaikan kecepatan proses yang terlalu lama.
2. Dapat menambahkan proses stemming untuk pre-processing data.
3. Dapat dilakukan penambahan dataset berita hoax secara otomatis dari sumber berita yang sudah teridentifikasi, dari situs-situs resmi berita hoax dan valid yang sudah ada. Agar proses identifikasi menjadi lebih akurat.

DAFTAR PUSTAKA

- Andre Rino Prasetyo, Indriati, & Putra Pandu. (2018). Klasifikasi Hoax Pada Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Modified K-Nearest Neighbor. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*.
- Asahar Johar T, Delfi Yanosma, & Kurnia Anggriani. (2016). Implementasi Metode K-Nearest Neighbor (KNN) dan Simple Additive Weighting (SAW) Dalam Pengambilan Keputusan Seleksi Penerimaan Anggota Paskibraka. *Jurnal Pseudocode*.
- AULIA AFRIZA, & JULIO ADISANTOSO. (2018). Metode Klasifikasi Rocchio untuk Analisis Hoax Rocchio Classification Method for Hoax Analysis. *Jurnal Ilmu Komputer Agri-Informatika*.
- Faisal Rahutomo, Ingrid Yanuar Risca Pratiwi, & Diana Mayangsari Ramadhani. (2019). EKSPERIMEN NAÏVE BAYES PADA DETEKSI BERITA HOAX BERBAHASA INDONESIA. *Jurnal Penelitian Komunikasi dan Opini Publik*, 1-15.
- Firmansyah, R. (2017). Web Klarifikasi Berita Untuk Meminimalisir Penyebaran Berita Hoax. *Jurnal Informatika*, 230-235.
- Hakim, I. (2018). FRAMING KOMPAS.COM TERHADAP BERITA KASUS PERNYATAAN HOAX RATNA SARUMPAET. 1.
- Jesri Hotman Tridayana Purba. (2019). IDENTIFIKASI DAN KLASIFIKASI SMS HOAX DENGAN KOMBINASI METODE N-GRAM DAN NAÏVE BAYES.
- Kurniawan, A. (2020). SISTEM PENDUKUNG KEPUTUSAN PENJADWALAN BESERTA RUTE PENGIRIMAN BARANG DENGAN METODE CLUSTERING K-MEANS DAN METODE DIJKSTRA . *Teknik*.
- Mara Destiningrum, & Qadhli Jafar Adrian. (2017). SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB SISTEM INFORMASI PENJADWALAN DOKTER BERBASIS WEB. *jurnal TEKNOINFO*.
- Maulana Iqbal Tanjung. (2011). Analisis dan Perancangan Sistem Informasi Berbasis Website Menggunakan Framework Codeigniter.
- Mulia Mahendra Alvanof, & Rahmat Triandi. (2020). ANALISA DAN DETEKSI KONTEN HOAX PADA MEDIA BERITA INDONESIA MENGGUNAKAN MACHINE LEARNING. *Jurnal Teknologi Terapan & Sains*.
- Munirul, Ula, Alvanof, M. M., & Triandi, R. (2020). ANALISA DAN DETEKSI KONTEN HOAX PADA MEDIA BERITA INDONESIA MENGGUNAKAN MACHINE LEARNING. *Jurnal Teknologi Terapan and Sains*, 1.

Purba, J. H. (2019). IDENTIFIKASI DAN KLASIFIKASI SMS HOAX DENGAN KOMBINASI METODE N-GRAM DAN NAÏVE BAYES.

Rozi, F. N., & Sulistyawati, D. H. (2019). Klasifikasi Berita Hoax Pilpres Menggunakan Metode Modified K-Nearest Neighbor dan Pembobotan Menggunakan TF-IDF. *KONVERGENSI*, 1.

Weddiningrum, F. G. (2018). DETEKSI KONTEN HOAX BERBAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN METODE LEVENSHTAIN DISTANCE.

LAMPIRAN

Lampiran I. Source Code

1) Source Code Pre-Processing Teks

```
|1.      public function preProcessing($teks){
|2.      $filterr = array("'",'"',"-","_","-
|      ",")",("","\'","/","=", "+", ".",":",";","!", "?", "\n");
|3.      $teks = str_replace($filterr," ", $teks);
|4.      $teks = strtolower(trim($teks));
|5.      $astoplist = array(
|6.      ' di',' ke',' se',' ada ',' adalah',' adanya',' adapun','
|      agak ',' agaknya',' agar ',' akan ',' akankah',' akhir',' lah','
|      kah',' akhiri',' akhirnya',' aku ',' akulah',' amat',' amatlah','
|      anda',' andalah',' antar',' antara',' antaranya',' apa',' apaan'
|      , 'apabila',' apakah',' apalagi',' apatah',' arti',' artinya',' as
|      al',
|      'asalkan',' atas',' atau',' ataukah',' ataupun',' awal',' awalnya
|      ',' bagai',' bagaikan',' bagaimana',' bagaimanakah',' bagaimanap
|      un',' bagainamakah',' bagi',' bagian',' bahkan',' bahwa',' bahwas
|      annya',' bahwasanya',' baik ',' belas',' baiklah',' bakal',' bak
|      alan',' balik',' banyak',' bapak',' beberapa',' begini',' beginia
|      n',' beginikah',' beginilah',' begitu',' begitukah',' begitulah'
|      , 'begitupun',' belakangan',' belumlah',' benarkah',' benarlah',
|      'berada',' berakhir',' berakhirilah',' berakhirnya',' berapa',' b
|      erapakah',' berapalah',' berapapun',' berarti',' berawal',' berb
|      agai',' berdatangan',' berikan',' berikut',' berikutnya',' berju
|      mlah',' berkali -
|      kali',' berkata',' berkehendak',' berkeinginan',' berkenaan',' b
|      erlainan',' berlalu',' berlangsung',' berlebihan',' bermacam','
|      bermacam-macam',' bermaksud',' bermula',' bersama',' bersama-
|      sama',' bersiap',' bersiap-siap',' bertanya',' bertanya-
|      tanya',' berturut',' berturut-
|      turut',' bertutur',' berujar',' berupa',' besar',' betul',' betul
|      kah',' biasa',' biasanya',' bila',' bilakah',' bisa',' bisakah','
|      boleh',' bolehkah',' bolehlah',' buat',' bukan',' bukankah',' buk
|      anlah',' bukannya',' bulan',' bung',' caranya',' cukup',' cukupka
|      h',' cukuplah',' cuma',' dahulu',' dalam',' dan ',' dapat',' dari
|      ',
|      'daripada',' datang',' dekat ',' demi ',' demikian',' demikian
|      lah',' dengan',' depan',' dia',' diakhiri',' diakhirinya',' diala
|      h',' diantara',' diantaranya',' diberi',' diberikan',' diberikan
|      nya',' dibuat',' dibuatnya',' didapat',' didapatkan',' digunaka
|      n',' diibaratkan',' diibaratkannya',' diingat',' diingatkan',' d
|      iinginkan',' dijawab',' dijelaskan',' dijelaskannya',' dikarena
|      kan',
|      'dikatakan',' dikatakannya',' dikerjakan',' diketahui',' diketa
|      huinya',' dikira',' dilakukan',' dilalui',' dilihat',' dimaksud'
```

, 'dimaksudkan', 'dimaksudkannya', 'dimaksudnya', 'diminta', 'dimintai', 'dimisalkan', 'dimulai', 'dimulailah', 'dimulainya', 'dimungkinkan', 'dini', 'dipastikan', 'diperbuat', 'diperbuatnya', 'dipergunakan', 'diperkirakan', 'diperlihatkan', 'diperlukan', 'diperlukannya', 'dipersoalkan', 'dipertanyakan', 'dipunyai', 'diri', 'dirinya', 'disampaikan', 'disebut', 'disebutkan', 'disebutkannya', 'disini', 'disinilah', 'ditambahkan', 'ditandaskan', 'ditanya', 'ditanyai', 'ditanyakan', 'ditegaskan', 'ditujukan', 'ditunjuk', 'ditunjuki', 'ditunjukkan', 'ditunjukkannya', 'ditunjuknya', 'dituturkan', 'dituturkannya', 'diucapkan', 'diucapkannya', 'diungkapkan', 'dong', 'dua', 'dulu', 'empat', 'enak', 'enggak', 'enggaknya', 'entah', 'entahlah', 'guna', 'gunakan', 'hada p', 'hai', 'hal', 'halo', 'hallo', 'hampir', 'hanya', 'hanyalah', 'hari', 'harus', 'haruslah', 'harusnya', 'helo', 'hello', 'hendak', 'hendaklah', 'hendaknya', 'hingga', 'ia', 'ialah', 'ibarat', 'ibaratkan', 'ibaratnya', 'ibu', 'ikut', 'ingat', 'ingat-ingat', 'ingin', 'inginkah', 'inginkan', 'ini', 'inikah', 'inilah', 'itu', 'itukah', 'itulah', 'jadi', 'jadilah', 'jadinya', 'jangan', 'janganakan', 'janganlah', 'jauh', 'jawab', 'jawaban', 'jawabnya', 'jelas', 'jelaskan', 'jelaslah', 'jelasnya', 'jika', 'jikalau', 'juga', 'jumlah', 'jumlahnya', 'justru', 'kadar', 'kala', 'kalau', 'kalaulah', 'kalaupun', 'kali', 'kalian', 'kami', 'kamilah', 'kamu', 'kamulah', 'kapan', 'kapankah', 'kapanpun', 'karena', 'karenanya', 'kasus', 'kata', 'katakan', 'katakanlah', 'katanya', 'keadaan', 'kebetulan', 'kecil', 'kedua', 'keduanya', 'keinginan', 'kelamaan', 'kelihatan', 'kelihatannya', 'kelima', 'keluar', 'kembali', 'kemudian', 'kemungkinan', 'kemungkinannya', 'kena', 'kenapa', 'kepada', 'kepadanya', 'kerja', 'kesampaian', 'keseluruhan', 'keseluruhannya', 'keterlaluhan', 'ketika', 'khusus', 'khususnya', 'kini', 'kinilah', 'kira', 'kira-kira', 'kiranya', 'kita', 'kitalah', 'kok', 'kurang', 'lagi', 'lagian', 'lain', 'lainnya', 'laku', 'lalu', 'lama', 'lamanya', 'langsung', 'lanjut', 'lanjutnya', 'lebih', 'lewat', 'lihat', 'lima', 'luar', 'macam', 'maka', 'makanya', 'makin', 'maksud', 'malah', 'malahan', 'mampu', 'mampukah', 'mana', 'manakala', 'manalagi', 'masa', 'masalah', 'masalahnya', 'masih', 'masihkah', 'masing', 'masing-masing', 'masuk', 'mata', 'mau', 'maupun', 'melainkan', 'melakukan', 'melalui', 'melihat', 'melihatnya', 'memang', 'memastikan', 'memberi', 'memberikan', 'membuat', 'memerlukan', 'memihak', 'meminta', 'memintakan', 'memisalkan', 'memperbuat', 'mempergunakan', 'memperkirakan', 'memperlihatkan', 'mempersiapkan', 'mempersoalkan', 'mempertanyakan', 'mempunyai', 'memulai', 'memungkinkan', 'menaiki', 'menambahkan', 'menandaskan', 'menanti', 'menanti-nanti', 'menantikan', 'menanya', 'menanyai', 'menanyakan', 'mendapat', 'mendapatkan', 'mendatang', 'mendatangi', 'mendatangkan', 'menegaskan', 'mengakhiri', 'mengapa', 'mengatakan', 'mengatakannya', 'mengenai', 'mengerjakan', 'mengetahui', 'menggunakan',

'menghendaki', 'mengibaratkan', 'mengibaratkannya', 'mengingat', 'mengingat', 'menginginkan', 'mengira', 'mengucapkan', 'mengucapkannya', 'mengungkapkan', 'menjadi', 'menjawab', 'menjelaskan', 'menuju', 'menunjuk', 'menunjuki', 'menunjukkan', 'menunjuknya', 'menurut', 'menuturkan', 'menyampaikan', 'menyangkut', 'menyatakan', 'menyebutkan', 'menyeluruh', 'merasa', 'mereka', 'merekalah', 'merupakan', 'meski', 'meskipun', 'meyakini', 'meyakinkan', 'minta', 'mirip', 'misal', 'misalkan', 'misalnya', 'mohon', 'mula', 'muat', 'mulai', 'mulailah', 'mulanya', 'mungkin', 'mungkinkah', 'naik', 'namun', 'nanti', 'nantinya', 'nya', 'nyaris', 'nyata', 'nyatanya', 'oleh', 'olehnya', 'orang', 'pada', 'pada hal', 'padanya', 'pak', 'paling', 'panjang', 'pantas', 'para', 'pasti', 'pastilah', 'penting', 'pentingnya', 'per', 'percuma', 'perlu', 'perlukah', 'perlunya', 'pernah', 'persoalan', 'pertama', 'pertama-
tama', 'pertanyaan', 'pertanyakan', 'pihak', 'pihaknya', 'pukul', 'pula', 'pun', 'punya', 'rasa', 'rasanya', 'rupa', 'rupanya', 'saat', 'saatnya', 'saja', 'sajalah', 'salam', 'saling', 'sama', 'sama-sama', 'sambil', 'sampai', 'sampai-
sampai', 'sampaikan', 'sana', 'sangat', 'sangatlah', 'sangkut', 'satu', 'saya', 'sayalah', 'sebab', 'sebabnya', 'sebagai', 'sebagaimana', 'sebagainya', 'sebagian', 'sebaik', 'sebaik-
baiknya', 'sebaiknya', 'sebaliknya', 'sebanyak', 'sebegini', 'sebegini', 'sebelum', 'sebelumnya', 'sebenarnya', 'seberapa', 'sebesar', 'sebetulnya', 'sebisanya', 'sebuah', 'sebut', 'sebutlah', 'sebutnya', 'secara', 'secukupnya', 'sedang', 'sedangkan', 'sedemikian', 'sedikit', 'sedikitnya', 'seenaknya', 'segala', 'segalanya', 'segera', 'seharusnya', 'sehingga', 'seingat', 'sejak', 'sejauh', 'sejenak', 'sejumlah', 'sekadar', 'sekadarnya', 'sekali', 'sekalikali', 'sekalian', 'sekaligus', 'sekalipun', 'sekarang', 'sekaranglah', 'sekecil', 'seketika', 'sekiranya', 'sekitar', 'sekitarnya', 'sekurang-
kurangnya', 'sekurangnya', 'sela', 'selain', 'selaku', 'selalu', 'selama', 'selama-
lamanya', 'selamanya', 'selanjutnya', 'seluruh', 'seluruhnya', 'semacam', 'semakin', 'semampu', 'semampunya', 'semasa', 'semasih', 'semata', 'semata-
mata', 'semaunya', 'sementara', 'semisal', 'semisalnya', 'sempat', 'semua', 'semuanya', 'semula', 'sendiri', 'sendirian', 'sendirinya', 'seolah', 'seolah-
olah', 'seorang', 'sepanjang', 'sepantasnya', 'sepantasnyalah', 'seperlunya', 'seperti', 'sepertinya', 'sepihak', 'sering', 'seringnya', 'serta', 'serupa', 'sesaat', 'sesama', 'sesampai', 'sesegera', 'sesekali', 'seseorang', 'sesuatu', 'sesuatunya', 'sesudah', 'sesudahnya', 'setelah', 'setempat', 'setengah', 'seterusnya', 'setiap', 'setiba', 'setibanya', 'setidak-
tidaknya', 'setidaknya', 'setinggi', 'seusai', 'sewaktu', 'siap', 'siapa', 'siapakah', 'siapapun', 'sini', 'sinilah', 'soal', 'soa

lnya','suatu','sudah','sudahkah','sudahlah','supaya','tadi','tadinya','tahu','tak','tambah','tambahnya','tampak','tampaknya','tandas','tandasnya','tanpa','tanya','tanyakan','tanyanya','tapi','tegas','tegasnya','telah','tempat','tentang','tentu','tentulah','tentunya','tepat','terakhir','terasa','terbanyak','terdahulu','terdapat','terdiri','terhadap','terhadapnya','teringat','teringat-ingat','terjadi','terjadilah','terjadinya','terkira','terlalu','terlebih','terlihat','termasuk','ternyata','tersampaikan','tersebut','tersebutlah','tertentu','tertuju','terus','terutama','tetap','tetapi','tiap','tiba','tiba-tiba','tidak','tidakkah','tidaklah','tiga','toh','tuju','tunjuk','turut','tutur','tuturnya','ucap','ucapnya','ujar','ujarnya','umumnya','ungkap','ungkapnya','untuk','usah','usai','wadu','wah','wahai','waktunya','walau','walaupun','wong','ya','yaitu','yakin','yakni','ng','sa','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z','ja','jak','ta','jum','meng','kan','no','si','mua','nah','nal','airan','kel','gala','kou','san','ber','hu','dr','spb','onk','him','ti','apkli','gkiri','te','kg','bps','menc','kkp','yang','mesti','pilih','bersungut','pudar','ndalikan','dradjat','informasi','benar','jenis','kota','produksi','badan','mi','but','tara','me','angg','rta','but','ter','trnas','pe','ring','kit','wasp','la','pisu','tiong','br','lu','mem','â€','nin','inst','ipb','ms','â€œberita','ma','tika','gkas','an','tr','nners','su','luk','nali','1','2','3','4','5','6','7','8','9','0');

```
|7.     foreach ($astoplist as $i => $arr) {
|8.         $teks = str_replace($astoplist[$i], " ", $teks);
|9.     }
|10.        $teks = strtolower(trim($teks));
|11.        return $teks;
|12.    }
```

2) Source Code Index Kata

```
|1.     public function buatIndexDataset(){
|2.         $this->M_Kata_Dataset->del_data();
|3.         $berita = $this->M_Dataset->getAll();
|4.         foreach($berita as $b){
|5.             $idDoc = $b['ID_BERITA'];
|6.             $isiBerita = $b['BERITA'];
|7.             $teksBerita = $this->M_Metode-
>preProcessing($isiBerita);
|8.             $arberita = explode(" ", trim($teksBerita));
```



```

|9.         $jumlah = array_count_values($arberita);
|10.    foreach($jumlah as $j => $j_value){
|11.        if($j != ""){
|12.            $row_count = count($this->M_Kata_Dataset-
>getAnd($idDoc,$j)
|13.                ->result_array());
|14.            $data = array(
|15.                'ID_BERITA' => $idDoc,
|16.                'TREM_DATASET' => $j,
|17.                'FREK_DATASET' => $j_value
|18.                if($row_count > 0){
|19.            $this->M_Kata_Dataset->updateData($data);
|20.                }else{
|21.                    $this->M_Kata_Dataset-
>add($data);
|22.                }
|23.            }
|24.        }
|25.    }
|26. }

```

3) Source Code Hitung Bobot Kata

```

|1.    public function hitungBobotDataset(){
|2.        $bobot = $this->M_Kata_Dataset->getAll();
|3.        foreach($bobot as $b){
|4.            $idKata = $b['ID_KATA_DATASET'];
|5.            $tremData = $b['TREM_DATASET'];
|6.            $TF = $b['FREK_DATASET'];
|7.            $D = $this->M_Kata_Dataset->getNDoc();
|8.            $df = $this->M_Kata_Dataset-
>getDF($tremData);
|9.            $W = $TF * log10($D/$df);
|10.           $data2 = array(
|11.               'ID_KATA_DATASET' => $idKata,
|12.               'BOBOT_KATA_DATASET' => $W);
|13.           $this->M_Kata_Dataset-
>updateBobot($data2);
|14.       }
|15.   }

```

4) Source Code Hitung Jarak Euclidean (KNN)

```

|1.    public function hitungKNN($id) {
|2.        $this->M_Chace->del_data();
|3.        if($id == null){ $docUji = $this->M_Bobot_Uji-
>getDistinct();
|4.        }else{$docUji = $this->M_Bobot_Uji->getID($id); }
|5.        $docDistinct = $this->M_Chace->getDistinct();
|6.        foreach($docUji as $dUji){

```

```

|7.          $idD = $dUji['ID_DATA_UJI'];
|8.          foreach($docDistinct as $DD){
|9.              $docID = $DD['ID_BERITA'];
|10.             $tremUji = $this->M_Chace-
>getAnd($idD);
|11.             $wUji = 0;
|12.             $wDataset = 0;
|13.             foreach($tremUji as $t){
|14.                 $arTrem = $t['TREM_UJI'];
|15.                 $uji = $t['BOBOT_KATA_UJI'];
|16.                 $wUji = $wUji + $uji;
|17.                 $data = array(
|18.                     'ID_BERITA' => $docID,
|19.                     'TREM_UJI' => $arTrem
|20.                 );
|21.             $tremData = $this->M_Chace->getBobotTrem($data);
|22.             $set = $tremData['BOBOT_KATA_DATA
SET'];
|23.             $wDataset = $wDataset + $set;
|24.         }
|25.         $temp = $wDataset - $wUji;
|26.         $pangkat = pow($temp,2);
|27.         $p_vektor = sqrt($pangkat);
|28.         $data2 = array(
|29.             'ID_DATA_UJI' => $idD,
|30.             'ID_BERITA' => $docID,
|31.             'JARAK_EUCLIDEAN' => $p_vektor);
|32.         $this->M_Chace->add($data2);
|33.     }}$this->M_Metode->lanjutKNN($id); }

```

5) Source Code Jarak Terdekat K=3 (3 buah dataset terdekat)

```

|1.     public function lanjutKNN($id) {
|2.         $this->M_Identifikasi_detail->del_data();
|3.         if($id == null){ $idTest = $this->M_Bobot_Uji-
>getDistinct();
|4.         }else{$idTest = $this->M_Bobot_Uji->getID($id); }
|5.         foreach($idTest as $t){
|6.             $idDocUji = $t['ID_DATA_UJI'];
|7.             $cache = $this->M_Chace-
>getLimit($idDocUji);
|8.             foreach($cache as $c){
|9.                 $data = array(
|10.                    'ID_DATA_UJI' => $idDocUji,
|11.                    'ID_BERITA' => $c['ID_BERITA'],
|12.                    'JARAK_EUCLIDEAN' => $c['JARAK_EUCLIDEAN']
);
|13.             $cek = $this->M_Identifikasi_detail->cekData($data);
|14.             if($cek > 0){

```

```

|15.      $this->M_Identifikasi_detail->updateData($data);
|16.      }else{$this->M_Identifikasi_detail->add($data);
|17.      }
|18.      }
|19.      }
|20.      $this->M_Chace->del_data();
|21.      $this->M_Metode->knnClass();}

```

6) Source Code Klasifikasi KNN

```

|1.      public function knnClass()
|2.      {
|3.          $detail = $this->M_Identifikasi_detail-
|4.          >getDistinct();
|5.          foreach ($detail as $det) {
|6.              $id = $det['ID_DATA_UJI'];
|7.              $lab1 = 'HOAX';
|8.              $lab2 = 'VALID';
|9.              $labH = $this->M_Identifikasi_detail-
|10.             >getId($id,$lab1);
|11.             $labV = $this->M_Identifikasi_detail-
|12.             >getId($id,$lab2);
|13.             if($labH > $labV){ $label = 'HOAX';
|14.             }else{$label = 'VALID';
|15.                 $data = array(
|16.                     'ID_DATA_UJI' => $id,
|17.                     'LABEL_IDENTIFIKASI' => $label);
|18.                 $cek = $this->M_Identifikasi-
|19.                 >cekData($data);
|20.                 if($cek > 0){ $this->M_Identifikasi-
|21.                 >updateData($data);
|22.                 }else{$this->M_Identifikasi->add($data); }
|23.             }}

```

Lampiran II. Surat Pencatatan Ciptaan

 REPUBLIK INDONESIA KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA	
SURAT PENCATATAN CIPTAAN	
Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:	
Nomor dan tanggal permohonan	: EC00202209894, 9 Februari 2022
Pencipta	
Nama	: DANU NUR IRWANTO dan MUHAMMAD HARIST MURDANI
Alamat	: JALAN MADE BARAT, RT.01, RW.02, KELURAHAN MADE, KECAMATAN SAMBIKEREK, SURABAYA, JAWA TIMUR, 60219
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT UNIVERSITAS WIJAYA PUTRA
Alamat	: JALAN RAYA BENOWO NO.1-3, SURABAYA, JAWA TIMUR, 60197
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Identifikasi Berita Hoax Menggunakan Kombinasi Metode K-Nearest Neighbor (KNN) Dan TF-IDF Berbasis Web Dengan Menggunakan Framework Codeigniter
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 9 Februari 2022, di SURABAYA
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan	: 000325198
adalah benar berdasarkan keterangan yang diberikan oleh Pemohon. Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.	
	a.n Menteri Hukum dan Hak Asasi Manusia Direktur Jenderal Kekayaan Intelektual u.b. Direktur Hak Cipta dan Desain Industri
 Dr. Syarifuddin, S.T., M.H. NIP.197112182002121001	
Disclaimer: Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.	