

**PENERAPAN *TEXT MINING* UNTUK PENGELOLAAN IDENTIFIKASI  
KELUHAN PENGGUNA SECARA OTOMATIS DI POLITEKNIK**

**NEGERI INDRAMAYU**

**PROPOSAL TUGAS AKHIR**



**Oleh:**

**ELBA AYU KURNIA**

**NIM 1703056**

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI INDRAMAYU**

**APRIL 2021**

## HALAMAN PENGESAHAN

### **PENERAPAN *TEXT MINING* UNTUK PENGELOLAAN IDENTIFIKASI KELUHAN PENGGUNA SECARA OTOMATIS DI POLITEKNIK**

#### **NEGERI INDRAMAYU**

**Disusun oleh :**

**ELBA AYU KURNIA**

**NIM 1703056**

**Proposal Tugas Akhir disetujui oleh:**

Calon Pembimbing	:	<u>Alifia Puspaningrum, S.Pd., M.Kom.</u> NIP 199305282019032024	.....
---------------------	---	---	-------

Indramayu, 8 April 2021  
Koordinator Program Studi  
D3 Teknik Informatika

Fachrul P.B.M., S.ST., M.Kom.  
NIP 199204232018031001

## DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
DAFTAR ISI.....	iii
1. Latar Belakang Masalah.....	1
2. Rumusan Masalah.....	2
3. Batasan Masalah.....	2
4. Tujuan.....	2
5. Manfaat.....	3
6. Landasan Teori.....	3
7. Metode Pelaksanaan.....	6
8. Rencana Kegiatan.....	10
DAFTAR PUSTAKA.....	10

### **Latar Belakang Masalah**

Pada sebuah perusahaan jasa, kepuasan pelanggan adalah salah satu hal yang dibutuhkan untuk meningkatkan pendapatan perusahaan. Penanganan komplain saat ini dinilai masih kurang karena bersifat manual. Hal ini menyebabkan pelanggan mengeluh karena komplain mereka tidak diproses secara cepat oleh pihak yang bersangkutan. Sebelumnya komplain disampaikan langsung kepada pihak yang bersangkutan dan bagian tersebut akan menjawab apa yang ditanyakan.

Namun, dengan berkembangnya teknologi proses tersebut dapat dilakukan oleh komputer. Bukan hal sulit jika proses tersebut dilakukan oleh manusia, hanya saja memerlukan waktu yang tidak sedikit. Apabila dilakukan oleh komputer tentu saja akan terdapat masalah baru, yaitu dapatkah komputer menentukan kategori komplain serta jawaban yang tepat untuk komplain tersebut (Rudhi, 2018).

Oleh karena itu, dengan menggunakan *text mining* dalam tugas akhir ini diharapkan dapat membantu dalam memilah atau mengetahui kategori dari sebuah komplain dan memberikan jawaban yang akurat . Dengan *Text mining* dapat dicari kata-kata yang dapat mewakili isi dari komplain, lalu dianalisis apakah komplain tersebut masuk dalam kategori asuransi apa. Oleh karena itu, dengan menggunakan *text mining* dalam tugas akhir ini diharapkan dapat membantu dalam memilah atau mengetahui kategori dari sebuah komplain tersebut masuk dalam kategori asuransi apa.

Sedangkan pengukuran tingkat *similaritas* antar dokumen dilakukan dengan membandingkan suatu *keyword* dengan dokumen yang sudah dibuat sbelumnya di *database*. Agar hasil pengukuran tingkat *similaritas* dokumen dengan *keyword* mendapatkan hasil yang optimal maka digunakanlah algoritma TF-IDF atau *Term Frequency Inverse Document Frequency*. Algoritma ini digunakan karena paling baik dalam perolehan informasi (Rona, 2018).

## **Rumusan**

### **Masalah**

Berdasarkan latar belakang yang dijabarkan, maka diperoleh suatu rumusan permasalahan yang menjadi dasar pembuatan sistem tersebut, yakni sebagai berikut :

Bagaimana membuat sebuah aplikasi komplain yang dapat mengelompokkan komplain berdasarkan kategori?

Bagaimana algoritma TF-IDF (

*Term Frekuensi Inverse Document Frequency*

) dapat digunakan dalam proses penentuan tingkat

*similiritas*

yang sesuai dengan cara mengukur tingkat

*similiritas*

antar dokumen dengan membandingkan suatu

*keyword*

dengan

*template*

komplain yang sudah dibuat sebelumnya di

*database*

?

## **Batasan**

### **Masalah**

Untuk memfokuskan pembahasan, dapat diperoleh beberapa batasan masalah, di antaranya:

Data pengujian

diperoleh

dari

*public*

*dataset*

.

Penggunaan algoritma TF-IDF dalam menentukan tingkat

*similiritas*

dengan komplain untuk memperoleh kelompok komplain yang serupa.

Sistem menggunakan bahasa pemrograman

Python

dan  
*database*  
MySQL.  
Proses  
*stemming*  
menggunakan algoritma  
*Porter Stemmer*  
dalam bahasa  
Indonesia.

## **Tujuan**

Tujuan dari pembuatan sistem yang dibuat meliputi :

Merancang dan membangun aplikasi

*Automatic Complaint Management System*

, untuk mempermudah pencatatan komplain dari pelanggan sampai penanganan terhadap komplain.

Mengimplementasikan algoritma TF-IDF untuk menentukan jawaban yang sesuai dengan *keyword*

.

## **Manfaat**

Manfaat yang dapat diperoleh dengan dibuatnya sistem ini adalah :

Memberikan masukan dan solusi dalam hal ini

bagian komplain maupun

*customer service*

dari suatu instansi atau perusahaan

untuk penanganan komplain yang akurat.

Mengetahui tingkat pemahaman pelanggan

akan

suatu

produk

atau layanan

yang dihasilkan dan dipasarkan

suatu

perusahaan

.

Tugas akhir (TA)

ini diharapkan dapat memberikan kontribusi untuk

pengembangan literatur dalam penelitian yang berhubungan dengan kepuasan konsumen.

## **Landasan**

### **Teori**

#### ***Text Mining***

*Text mining* adalah salah satu bidang khusus dari *data mining*. Sesuai dengan buku *The Text mining Handbook*, *text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah peringkatan dokumen. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *Text mining* antara lain yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*).

Permasalahan yang dihadapi pada *text mining* sama dengan permasalahan yang terdapat pada *data mining*, yaitu jumlah data yang besar, dimensi yang tinggi, dan struktur data yang terus berubah, serta data *noise*. Perbedaan di antara keduanya adalah pada data yang digunakan. Pada *data mining*, data yang digunakan adalah *structured data*, sedangkan pada *text mining*, data yang digunakan *text mining* pada umumnya adalah *unstructured data*, atau minimal *semistructured*. Hal ini menyebabkan adanya tantangan tambahan pada *text mining* yaitu struktur teks yang *complex* dan tidak lengkap, arti yang tidak jelas dan tidak *standard*, dan bahasa yang berbeda ditambah translasi yang tidak akurat.

*Text mining* sendiri memiliki beberapa tipe antara lain:

*Search and Information Retrieval*

Menyimpan dan menemukan kembali dokumen teks, termasuk mesin pencari dan kata kunci pencarian.

*Document*

*Clustering*

Pengelompokan dan pengkategorian istilah, potongan, paragraf, atau dokumen menggunakan metode mining

.

*Document Classification*

Pengelompokan dan pengkategorian istilah, potongan, paragraf, atau dokumen menggunakan metode *document classification*

## Web Mining

Data dan *text mining* pada internet yang fokus pada skala dan antar hubungan pada *website*.  
*Information Extraction*

Identifikasi dan ekstraksi fakta yang relevan.  
*Natural Language Processing*

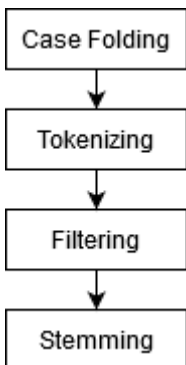
Pemrosesan bahasa tingkat rendah yang biasanya digunakan untuk bahasa komputasi.  
*Concept Extraction*

Pengelompokan kata dan frase dalam grup yang sama.

### Ekstraksi Dokumen

Teks yang akan dilakukan proses *text mining*, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi huruf yang lebih tinggi, terdapat *noise* pada data, dan terdapat struktur teks yang kurang baik. Cara yang digunakan dalam mempelajari suatu teks data adalah dengan menentukan fitur – fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen. Sebelum menentukan fitur – fitur yang mewakili, diperlukan tahap *preprocessing*.

Yang secara umum dilakukan dalam *text mining* pada dokumen, yaitu *case folding*, *tokenizing*, *filtering*, *stemming*, *tagging* dan *analyzing*.



Gambar1. Proses Ekstraksi Dokumen (Sumber: \*)

### Algoritma

#### TF/IDF

Metode TF/IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu yang disebut *Term Frequency* (TF) dan *inverse* frekuensi dokumen yang mengandung kata yang disebut *Inverse Document Frequency* (IDF). Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut di dalam dokumen. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut akan

rendah pada kumpulan dokumen. Rumusan umum TF-IDF dituliskan pada persamaan 1 berikut.

Rumus umum untuk TF-IDF :

$$W_{dt} = TF_{dt} * IDF_t (1)$$

Keterangan:

d = dokumen ke-d

t = kata ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = banyaknya kata yang dicari pada sebuah dokumen IDF = *Inverse Dokument Frequency*

D= total dokumen

df = banyak dokumen yang mengandung kata yang dicari

*Term* yang sering muncul pada dokumen tapi jarang muncul pada kumpulan dokumen memberikan nilai bobot yang tinggi. W akan meningkat dengan jumlah kemunculan *term* pada dokumen dan berkurang dengan jumlah *term* yang muncul pada dokumen. Setelah bobot (W) masing-masing dokumen diketahui, maka dilakukan proses pengurutan dimana semakin besar nilai W, semakin besar tingkat kecocokan dokumen tersebut terhadap kata kunci, demikian sebaliknya.

## Metode Pelaksanaan

### Metode Pengembangan Perangkat Lunak

Peneliti menggunakan metode *waterfall*, *waterfall* adalah pendekatan *System Development Life Cycle* (SDLC) paling awal yang digunakan untuk pengembangan perangkat lunak. Disebut juga sebagai model SDLC linear-sekuensial. Pendekatan *Waterfall* digunakan secara luas dalam pengembangan sistem, step-step nya terdiri dari:

#### *Requirement Gathering and analysis*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun dengan mengumpulkan contoh *dataset* yang akan diolah kemudian masuk pada tahap klasifikasi.

#### *System Design*

Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap. Merancang sistem dimulai dari *mock up interfaces*, *database*, gambar atau *icon* aplikasi.

#### *Implementation*

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit. Mulai mengimplementasikan sistem yang sudah dirancang dan dianalisis ke dalam kode program. Setiap fungsi yang dibuat akan langsung dites apakah terjadi *error* atau tidak.

#### *Integration and Testing*

Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*). Melakukan tes semua fungsi atau fitur yang terdapat pada aplikasi apakah terjadi *error* atau tidak.

#### *Deployment of system*

Mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan



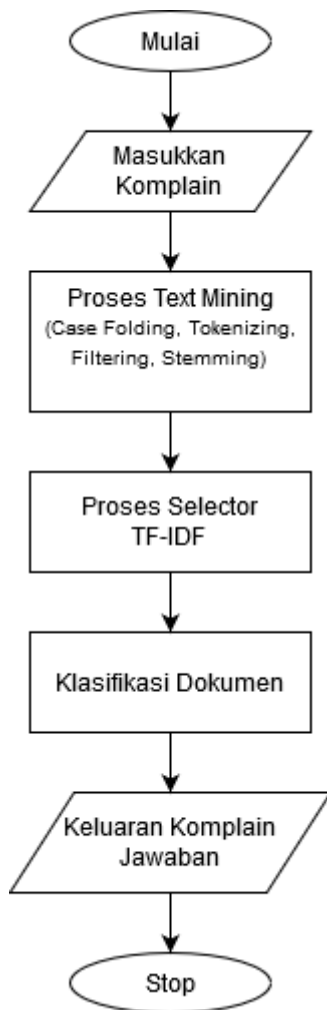
karena adaptasi dengan situasi sebenarnya dengan menerapkan aplikasi pada instansi atau perusahaan yang membutuhkan.

#### Maintenance

Proses pemeliharaan sistem yang sudah dibangun. Pemeliharaan dilakukan selama aplikasi masih digunakan pada institusi atau perusahaan terkait.

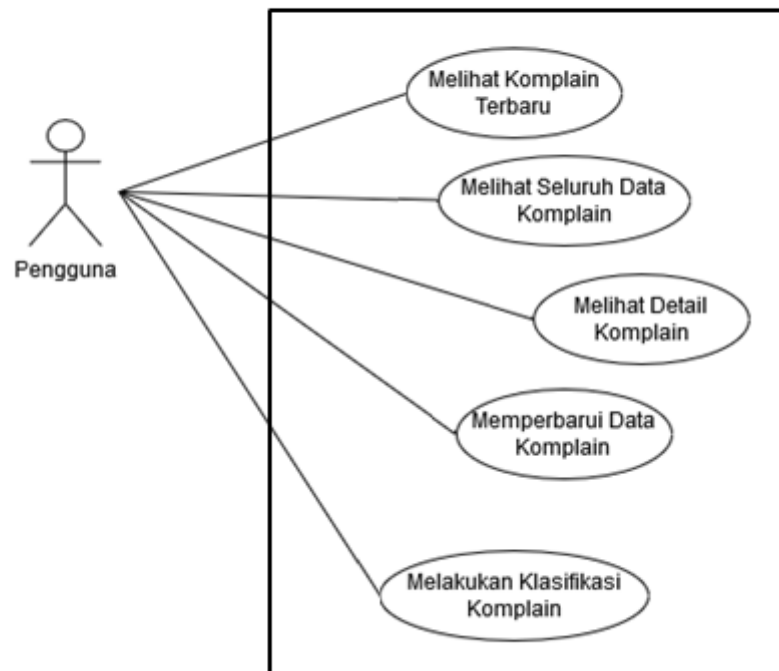
## Analisis Sistem

Tugas akhir ini merupakan usaha dalam upaya mendapatkan penyelesaian permasalahan aplikasi *Automatic Complaint Management System* menggunakan Algoritma TF- IDF untuk memberikan jawaban yang akurat terhadap masukan dari pelanggan. Bagi perusahaan aplikasi ini akan mambantu meningkatkan pelayanan yang lebih baik terhadap pelanggan.



Gambar 2. *Flowchart Proses Automatic Complaint Management System*

PENERAPAN TEXT MINING UNTUK PENGELOLAAN IDENTIFIKASI  
KELUHAN PENGGUNA SECARA OTOMATIS DI  
POLITEKNIK NEGERI INDRAMAYU



Gambar 3. *Use Case Proses Automatic Complaint Management System*

Proses pertama dimulai dari pelanggan yang menginputkan dokumen yang berupa teks. Masukan yang berupa teks dari pelanggan akan di *filtering* terhadap kata-kata yang tidak penting seperti: yang, dan, yaitu, dan lain sebagainya. kemudian dilakukan seleksi fitur dengan menggunakan pembobotan algoritma TF-IDF. Klasifikasi dokumen berfungsi memilah atau mengetahui kategori dari sebuah komplain. Jawaban diperoleh dengan membandingkan suatu *keyword* dengan dokumen yang sudah dibuat.

## Rencana Kegiatan

Tabel 1. Rencana Kegiatan

NO	KEGIATAN	April 2020				Mei 2020				Juni 2020				Juli 2020				Agustus 2020			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1	Identifikasi Masalah																				
2	Analisis Kebutuhan Sistem																				
3	Studi Literatur																				
4	Membuat Rancangan Sistem																				
5	Implementasi Program																				
6	Uji Coba Program (testing)																				
7	Revisi Konsep, Desain Rancangan, Code Program																				
8	Penyusunan Laporan Penulisan Tugas Akhir/Skripsi																				
9	Pelaksanaan Sidang Tugas Akhir/Skripsi																				
10	Pelaksanaan Revisi Tugas Akhir/Skripsi																				

## Daftar

## Pustaka

Dellia, P. (2017). *Kombinasi TF-IDF dan Pemilihan Fitur Markov Random Field (MRF) Dalam Perbaikan Kinerja Pengelompokan Twitter Keluhan Pajak* (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).

Qaiser, S., & Ali, R. (2018). Text mining: use of TF-IDF to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1), 25-29.

Rona, Nesya Dwi. (2018). *Model Text Mining Untuk Identifikasi Keluhan Pelanggan Produk Perusahaan Perangkat Lunak*. Yogyakarta : Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia.

Rudhi, Ardhi Sasmita. (2018). *Pemanfaatan TF/IDF Untuk Sistem Informasi e-Complaint Handling*. Surabaya: Jurusan Sistem Informasi Universitas Narotama Surabaya.

Saputra, Iwan. (2011). *Penggunaan Algoritma TFIDF Dalam Proses Hierarchical Template Matching*. Bali : Jurnal Konferensi Nasional Sistem dan Informatika.