

# Penerapan Algoritma Text Mining dan TF-IDF Untuk Pengelompokan Topik Skripsi Pada Aplikasi Repository STMIK Budi Darma

Herlina Sari, Guidio Leonarde Ginting, Taronisokhi Zebua, Mesran

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: herlinasari0198@gmail.com

**Abstrak**—Skripsi adalah suatu karya ilmiah yang wajib ditulis oleh mahasiswa sebagai syarat untuk tugas akhir pendidikan. Bagi mahasiswa yang ingin menyusun skripsi, contohnya mahasiswa pada kampus STMIK Budi Darma diharuskan terlebih dahulu mencari topik untuk judul yang akan diajukan. Adapun cara untuk mencari referensi topik skripsi dapat dilakukan dengan cara mengakses aplikasi repository. Judul skripsi memiliki topik yang berbeda-beda, maka dibutuhkan sebuah pengelompokan topik skripsi. Pengklasifikasian atau pengelompokan judul skripsi pada aplikasi repository sangat penting dilakukan, karena dengan adanya pengelompokan terhadap judul skripsi maka akan mempermudah dalam mencari informasi topik skripsi yang bisa dijadikan sebagai referensi pada penelitian selanjutnya. Maka dari itu, penelitian ini bertujuan untuk membuat sebuah aplikasi repository yang mampu mengelompokkan skripsi. Penelitian ini menggunakan tiga metode yaitu Text Mining, TF-IDF, dan cosine similarity. Data abstrak skripsi akan diproses oleh Text Mining untuk menghasilkan kalimat yang mewakili skripsi, kemudian diberi bobot menggunakan TF-IDF dan cari tingkat kemiripan menggunakan cosine similarity. Berdasarkan penelitian yang telah dilakukan, abstrak skripsi yang sesuai dengan kategori adalah 34 abstrak dari 50 abstrak skripsi yang diolah. Maka jika dipersentasekan hanya 73%.

**Kata Kunci:** Text Mining; TF-IDF; Cosine Similarity; Repository

**Abstract**—Thesis is a scientific work that must be written by students as a requirement for the final project of education. For students who want to write a thesis, for example, students on the STMIK Budi Darma campus are required to first find a topic for the title to be submitted. The way to find thesis topic references can be done by accessing the repository application. The title of the thesis has different topics, so it takes a grouping of thesis topics. Classification or grouping of thesis titles in the repository application is very important, because with the grouping of thesis titles it will make it easier to find thesis topic information that can be used as a reference in further research. Therefore, this study aims to create a repository application that is able to group theses. This research uses three methods, namely Text Mining, TF-IDF, and cosine similarity. The thesis abstract data will be processed by Text Mining to produce sentences that represent the thesis, then weighted using TF-IDF and find the level of similarity using cosine similarity. processed. So if the percentage is only 73%.

**Keywords:** Text Mining; TF-IDF; Cosine Similarity; Repository

## 1. PENDAHULUAN

Skripsi adalah istilah yang digunakan didalam pendidikan untuk menggambarkan suatu karya ilmiah berupa tulisan hasil dari penelitian sarjana S1 yang membahas tentang suatu permasalahan dalam bidang ilmu tertentu. Skripsi adalah suatu karya ilmiah yang wajib ditulis oleh mahasiswa sebagai syarat untuk tugas akhir pendidikan. Bagi mahasiswa yang ingin menyusun skripsi, diharuskan terlebih dahulu mencari topik untuk judul yang akan diajukan. Penentuan topik tentunya memerlukan referensi penelitian terdahulu yang telah diterima. Tujuannya agar tidak ada penggunaan topik yang sama dan judul yang hampir sama.

Topik skripsi merupakan pokok atau inti pembicaraan yang juga dapat diartikan sebagai permasalahan yang akan dibahas atau diuraikan didalam suatu karya ilmiah. Adapun cara untuk mencari referensi topik skripsi dapat dilakukan dengan cara mengakses aplikasi *repository*. Aplikasi *repository* adalah sebuah aplikasi penyimpanan yang bisa digunakan untuk manajemen data dalam informasi, atau penggunaan kumpulan jurnal atau karya ilmiah dari civitas akademik dalam suatu kampus atau universitas, salah satu contohnya adalah kampus STMIK Budi Darma. Saat ini STMIK Budi Darma masih belum memiliki sebuah aplikasi *repository* yang telah memuat fungsi pengklasifikasian terhadap judul skripsi, sehingga pada saat dilakukan penelusuran/pencarian judul skripsi, maka membutuhkan waktu yang lama untuk menemukan referensi topik skripsi yang ingin dicari. Pengklasifikasian atau pengelompokan judul skripsi pada aplikasi *repository* sangat penting dilakukan, karena dengan adanya pengelompokan terhadap judul skripsi maka akan mempermudah dalam mencari informasi topik skripsi yang bisa dijadikan sebagai referensi pada penelitian selanjutnya.

Teknik klasifikasi memiliki beberapa algoritma yang dapat digunakan untuk mengimplementasikannya. Salah satu algoritmanya adalah algoritma *text mining*. *Text mining* merupakan suatu konsep atau langkah yang diterapkan untuk melakukan suatu proses penyelidikan data tersembunyi dalam bentuk data teks. Adapun teknik yang dimiliki *text mining* adalah pengelompokan atau pengklasifikasi yang memiliki fungsi untuk membentuk beberapa kelompok yang memiliki kemiripan atau kesamaan dalam data setiap kelompoknya. *Text mining* pada umumnya adalah *unstructured data*, atau minimal *semi structured*[1]. TF-IDF merupakan metode perhitungan bobot setiap kata yang paling umum digunakan. Metode ini merupakan algoritma yang melakukan penggabungan dua metode yaitu konsep frekuensi kemunculan *term* dalam sebuah dokumen dan *inverse* frekuensi dokumen yang mengandung kata tersebut, sehingga proposisi jumlah dokumen yang ditemukan kembali dan dianggap relevan dengan *query* akan meningkat[2].

Penulis memilih algoritma *text mining* dan algoritma TF-IDF untuk mengelompokkan topik skripsi berdasarkan penelitian terdahulu oleh Rizki Tri Wahyuni, Dhidik Prastiyanto, dan Eko Suprpto dengan judul “Penerapan Algoritma *Cosine Similarity* dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi” menyimpulkan bahwa hasil penelitian tersebut menunjukkan bahwa persentase tingkat ketetapan klasifikasi sistem adalah sebesar 98% [3]. Dan



penelitian yang dilakukan oleh Moh. Mahdi Sya'bani dan Reni Umilasari dengan judul “Penerapan Metode *Cosine Similarity* dan Pembobotan TF/IDF pada Sistem Klasifikasi Sinopsis Buku di Perpustakaan Kejaksaan Negeri Jember” menyimpulkan bahwa pengujian aplikasi terdapat 120 dokumen sinopsis dengan 10 kategori dan menghasilkan nilai *precision* sebesar 90,91% pada *threshold* 0,1 dan nilai *recall* sebesar 100% pada *threshold* 0,1 dan 0,2. Ketepatan akurasi pada sistem aplikasi yang diuji adalah 80,83% [4].

Penelitian ini menguraikan tentang bagaimana mengklasifikasikan topik skripsi melalui isi abstrak dari masing-masing skripsi Teknik Informatika STMIK Budi Darma. Kata-kata dari isi abstrak skripsi akan diseleksi berdasarkan algoritma *text mining*, sehingga menghasilkan suatu kalimat berupa topik. Proses yang dilakukan *text mining* adalah *tokenizing* (menghilangkan tanda baca), *filtering* (menghilangkan kata tidak penting) dan *stemming* (mengubah kata turunan menjadi kata dasar). Proses penentuan bobot pada kalimat topik yaitu dengan menggunakan algoritma TF-IDF (*Term Frequency – Inverse Document Frequency*). Proses yang dilakukan TF-IDF adalah membandingkan setiap kalimat topik yang telah dihasilkan dari proses *text mining* dengan menghitung bobotnya berdasarkan algoritma TD-IDF. Nilai bobot yang didapatkan akan digunakan untuk menghitung tingkat kemiripan isi skripsi berdasarkan metode *cosine similarity*.

## 2. METODOLOGI PENELITIAN

### 2.1 Text Mining

Ilmu *text mining* digunakan untuk mengubah kumpulan teks menjadi numerik sehingga dapat dikomputasikan. *Text mining* adalah salah satu bidang khusus dari *datamining*. 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 *datamining* yang salah satunya adalah peringkatan dokumen. Fungsi dari *text mining* biasanya digunakan dalam klasifikasi dokumen tekstual dimana dokumen-dokumen tersebut akan diklasifikasikan sesuai dengan topik dokumen tersebut. Adanya bantuan dari *text mining*, maka suatu artikel dapat diketahui jenis kategorinya melalui kata-kata yang terdapat pada artikel tersebut. Sehingga dengan adanya *text mining* dapat membantu melakukan pengelompokkan suatu dokumen dengan waktu yang singkat.

### 2.2 Algoritma TF-IDF

Pembobotan TF/IDF adalah suatu hubungan kata (*term*) yang berada pada dokumen yang akan diberikan suatu nilai bobot[4]. 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[5].

*Term Frequency* dan *Inverse Document Frequency* (TF-IDF) merupakan pembobotan yang sering digunakan dalam penelusuran informasi dan *text mining*[6].

Cara kerja dalam mencari nilai *term-frequency* melalui beberapa persamaan yaitu :

$$Tf_{t,d} = 1 + \log tf \dots\dots\dots(1)$$

dimana setiap variabel dijelaskan sebagai berikut :

tf : *term* frekuensi atau banyaknya kata pada dokumen

Tft,d : *term* frekuensi atau banyaknya kata t pada dokumen d atau pembobotan local

Mencari nilai *inverse document-frequency* melalui persamaan 2 :

$$Idf_t = \log n/df_t \dots\dots\dots(2)$$

Idf<sub>t</sub> : *inverse document-frequency* atau pembobotan global

n : banyaknya dokumen

df<sub>t</sub> : banyaknya dokumen yang memiliki kata t.

Berdasarkan persamaan 1 dan persamaan 2 baru dapat ditentukan nilai bobotnya (W<sub>t,d</sub>) dengan mengalikan kedua persamaan sehingga menjadi persamaan 3.

$$W_{t,d} = Tf_{t,d} \times Idf_t \dots\dots\dots(3)$$

dimana:

Tf<sub>t,d</sub> : *term* frekuensi atau banyaknya kata pada dokumen atau pembobotan local

Idf<sub>t</sub> : *inverse document frequency* atau pembobotan global

W<sub>t,d</sub> : nilai bobot akhir kata

Kemudian lakukan normalisasi pada bobot yang telah didapatkan dengan menggunakan persamaan 4.



$$\frac{W_{t,d}}{\sqrt{\sum_{t=1}^n (W_{t,d})^2}} \dots \dots \dots (4)$$

Dimana :

n : banyaknya kata

t : iterasi kata ke-

W<sub>t,d</sub> : nilai bobot akhir kata

Pembobotan tf-idf digunakan untuk mengukur seberapa penting suatu kata dalam suatu dokumen[6].

### 2.3 Algoritma Cosine Similarity

*Cosine similarity* adalah salah satu ukuran kemiripan paling populer yang diterapkan pada dokumen teks, seperti dalam berbagai aplikasi pengambilan dalam pembentukan dan pengelompokan[7]. Metode *cosine similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum perhitungan metode ini didasarkan pada *vector space similarity measure*. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah vektor dengan menggunakan kata kunci dari sebuah dokumen sebagai ukuran.

Rumus perhitungan *cosine similarity* :

$$\text{Cos Sim}(d_i, q_i) = \frac{q_i \cdot d_i}{|q_i| |d_i|} = \frac{\sum_j^t (q_{ij} \cdot d_{ij})}{\sqrt{\sum_j^t (q_{ij})^2} \cdot \sqrt{\sum_j^t (d_{ij})^2}} \dots \dots \dots (5)$$

keterangan :

q<sub>ij</sub> : term ke-i untuk dokumen ke-jq

d<sub>ij</sub> : term ke-i untuk query ke-j(keyword term)

t : jumlah istilah j pada q atau d

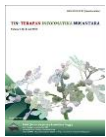
Model ruang vektor dan pembobotan TF-IDF digunakan untuk merepresentasikan nilai numerik dokumen sehingga kemudian dapat dihitung kedekatan antar dokumen. Kemiripan antar dokumen dihitung menggunakan suatu fungsi ukuran kemiripan (*similarity measure*). Semakin besar hasil fungsi *similarity*, maka kedua objek yang dievaluasi semakin mirip, demikian pula sebaliknya. Ukuran ini memungkinkan perbandingan dokumen sesuai dengan kemiripan (relevansi)nya terhadap *query*. Kualitas hasil dari dokumen yang didapatkan sangat tergantung pada fungsi *similarity* yang digunakan[8].

## 3. HASIL DAN PEMBAHASAN

Setiap mahasiswa tingkat akhir tentunya akan menyelesaikan satu mata kuliah yang menjadi akhir dari perkuliahan yaitu skripsi. Skripsi disusun oleh mahasiswa secara perorangan dengan tujuan agar mampu menganalisis, merangkum dan menerapkan atau mengaplikasikan pengalaman yang didapatkan selama pendidikan yang di tuangkan kedalam sebuah karya ilmiah atau skripsi. Ketika ingin menyusun skripsi tentunya mahasiswa harus mencari topik lebih dulu agar dapat dijadikan sebuah judul skripsi. Namun, terkhusus mahasiswa STMIK Budi Darma masih memiliki kendala dalam mencari topik skripsi, hal itu terjadi karena tidak tersedia sebuah aplikasi yang mendukung mahasiswa dalam mencari topik skripsi. STMIK Budi Darma hanya menyediakan dalam bentuk dokumen kumpulan-kumpulan judul skripsi yang dapat diperoleh dari portal kampus STMIK Budi Darma. Sehingga mahasiswa sangat membutuhkan sebuah aplikasi yang dapat membantu dalam mencari topik skripsi, misalnya aplikasi perpustakaan atau yang sering disebut dengan aplikasi *repository* yang di dalamnya terdapat berbagai macam judul skripsi dari berbagai bidang keilmuan. Oleh sebab itu, mahasiswa membutuhkan satu sistem yang mampu membantu dalam mencari topik skripsi dengan mudah yaitu *repository* atau perpustakaan digital yang juga STMIK Budi Darma.

Sebelum merancang sebuah sistem perlu dilakukan analisis terhadap skripsi terlebih dahulu. Tujuan dibuatnya aplikasi *repository* adalah untuk memudahkan mahasiswa mencari topik skripsi agar mampu membuat sebuah judul skripsi tanpa harus mengunjungi perpustakaan kampus atau perpustakaan daerah terdekat. *Repository* menyediakan berbagai skripsi berdasarkan bidang-bidang keilmuan. Namun, masalah yang terjadi dalam pengelompokan skripsi ini adalah kekeliruan dari pengelompokan skripsi, dimana hal tersebut terjadi karena keterbatasan pemahaman dari koleksi skripsi.

Untuk dapat menentukan kategori skripsi secara otomatis penulis menerapkan algoritma *text mining* dan algoritma *Term Frequency – Inverst Document Frequency* (TF-IDF). Penerapan algoritma tersebut terbagi menjadi dua yaitu untuk algoritma *text mining* digunakan sebagai pengolahan teks agar kata yang ditampilkan merupakan akar dari kalimat. Kedua penerapan algoritma *Term Frequency – Inverst Document Frequency* (TF-IDF) untuk mengolah kata hasil *text mining* dengan cara membandingkan terhadap data target, dalam hal ini penulis menentukan data target sebanyak 50 judul skripsi. Hasil dari algoritma *Term Frequency – Inverst Document Frequency* (TF-IDF) tersebut akan menentukan judul skripsi yang sudah diketahui kategori skripsi berdasarkan nilai dari bobot terbesar dari algoritma TF-IDF. Selanjutnya, dilakukan proses mengukur tingkat kemiripan dari masing-masing dokumen menggunakan algoritma *Cosine Similarity*. Hasil dari *Cosine Similarity* akan menentukan pengelompokan bidang.



*Data resource* digunakan sebagai *keyword* atau kata kunci dalam proses hitung kemiripan yang berupa kata dasar. *Keyword* atau kata kunci didapat dari perbandingan masing-masing abstrak sesuai kategori, nilai bobot yang paling tinggi akan dijadikan sebagai *keyword* atau kata kunci untuk dapat mewakili masing-masing kategori skripsi. *Data resource* dapat dilihat pada tabel 1.

**Tabel 1.** Data Resource

No	Kategori skripsi	Kata kunci / keyword
1	Kriptografi	citra, digital, rentan, sadap, curi, data, tanggung, jaga, aman, hasil, usg, manfaat, teknik, kriptografi, sandi, enkripsi, bentuk, paham, algoritma, camellia, salah, andal, wujud, tingkat, proses, rumit, sulit, kriptanalisis, akses, peneliti, bit, deskripsi, tahap, chipper, perlu, ronde, dimana, masuk, fungsi, $f$ , $f^{-1}$ , subkunci, kw, kkl, urai, dasar, erti, orang, upaya, minimalisir, tindak, salahguna
2	Kompresi	butuh, kapasitas, simpan, muncul, teknik, kompresi, ukur, data, akibat, boros, memori, lambat, proses, pindah, tujuan, hemat, peneliti, analisis, algoritma, fibonacci, code, file, audio, ekstensi, mp, parameter, banding, ratio, compression, space, savings, bit, kirim, penuh, sistem, harap, ulang, masyarakat, milik, jenis, lossy, dekompresi
3	Pengolahan Citra	citra, mata, alat, diagnostik, luas, kumpul, informasi, pasien, retinopathy, dokter, deteksi, erti, kenal, lesion, pembuluh, kapiler, saraf, optik, kamerafundus, milik, kontras, greylevel, jangkau, dinamis, dampak, serius, prosedur, diagnosis, hasil, darah, wilayah, fov, field, view, spesialis, tingkat, kualitas, tugas, kembang, metode, contrast, stretch, image, enhancement, proses, mudah, interpretasi, manusia, human, visual, system, hvs, salah, preprocess, manipulasi, domain, frekuensi, dynamic, range, digital, ubah, nilai, pixel, awal, dasar, target, gambar
4	Sistem Pendukung Keputusan (SPK)	lkp, karya, prima, salah, lembaga, kursus, latih, perusahaan, sedia, instruktur, kualitas, selesai, soal, rancang, aplikasi, sistem, dukung, putus, bantu, manajemen, seleksi, prestasi, spk, alternatif, ambil, proses, kelas, penuh, kriteria, multiple, criteria, decision, make, mcdm, topsis, terap, metode, mudah
5	Data Mining	tambang, data, upaya, gali, informasi, harga, database, datamining, knowledge, discovery, singkat, kkd, salah, algoritma, populer, teknik, apriori, temu, pola, kombinasi, hubung, item, set, association, rules, atur, asosiasi, implementasi, bidang, bisnis, dagang, didik, telekomunikasi, hasil, bantu, bijak, ambil, putus, prediksi, sewa, alat, berat, sistem, perusahaan, jenis, prioritas, utama, stok,antisipasi, kosong, barang, minim, pengaruh, layan, konsumen, iring, cipta, lingkungan, inteligen, hadap, tingkat, saing,

### 3.1 Penerapan Algoritma TF-IDF

Algoritma *text mining* digunakan pada aplikasi *repository* dimana *text mining* adalah salah satu cara untuk melakukan pengklasifikasian dengan cara menemukan pola yang berbeda di dalam data tekstual yang berjumlah besar. Berikut ini adalah pengelompokan topik skripsi berdasarkan perbandingan satu abstrak dengan 5 katagori yang telah disediakan. Contoh abstrak skripsi yang diambil yaitu atas nama Neti Rusri Yanti tahun 2018 dengan judul “Implementasi Algoritma Camellia Pada Penyandian Record Database”. Proses *text mining* dapat dilihat pada tabel berikut ini :

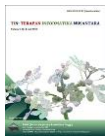
**Tabel 2.** Contoh Teks Abstrak

Abstrak
<i>Record database</i> dalam bentuk teks sangat banyak digunakan, sehingga sangat rentan terhadap pencurian data oleh pihak-pihak yang tidak berhak. Demi menjaga keamanan <i>record database</i> dapat dilakukan dengan pemanfaatan teknik kriptografi. Teknik kriptografi dapat menyandikan <i>record database</i> dengan mengenkripsikannya ke dalam bentuk sandi-sandi yang tidak dipahami. Algoritma Camellia adalah salah satu algoritma yang dapat diandalkan dalam mewujudkan teknik kriptografi. Algoritma simetri ini akan menghasilkan tingkat keamanan yang lebih tinggi terhadap <i>record database</i> karena dapat menyandikannya ke bentuk sandi dengan proses yang cukup rumit sehingga akan mempersulit kriptanalisis untuk mengakses <i>database</i> tersebut. Penelitian ini akan menggunakan Algoritma Camellia 128 bit untuk proses enkripsi dan dekripsinya, sehingga dalam prosesnya perlu melalui beberapa tahap yang panjang agar dapat menghasilkan <i>cipher</i> akhirnya. Algoritma ini memerlukan 18 ronde dimana setiap enam ronde harus memasuki Fungsi F dan Fungsi $FL^{-1}$ dan juga memerlukan sebanyak 26 subkunci yaitu kw, k dan kl. Penelitian ini menguraikan proses pengamanan <i>record database</i> dengan menyandikannya berdasarkan algoritma Camellia, sehingga dihasilkan teks <i>record database</i> dalam bentuk sandi yang sulit dipahami dan dimengerti oleh orang lain. Hal ini dilakukan sebagai upaya untuk meminimalisir tindakan-tindakan penyalahgunaan <i>record database</i> .

Terdapat beberapa langkah dalam penerapan *text mining* pada pengklasifikasian judul skripsi berdasarkan abstrak. Berikut langkah-langkah dari *text mining* :

#### 1. Tokenizing

*Tokenizing* merupakan langkah awal dalam penguraian deskripsi yang berupa suatu kalimat menjadi potongan kata. Berikut proses *tokenizing* pada kalimat-kalimat yang terdapat didalam abstrak.



**Tabel 3.** Hasil *Tokenizing*

Abstrak	Teks Input	Hasil Tokenizing
1	<p>record database dalam bentuk teks sangat banyak digunakan sehingga sangat rentan terhadap pencurian data oleh pihak pihak yang tidak berhak demi menjaga keamanan record database dapat dilakukan dengan pemanfaatan teknik kriptografi teknik kriptografi dapat menyandikan record database dengan mengenkripsikannya ke dalam bentuk sandi sandi yang tidak dipahami algoritma camellia adalah salah satu algoritma yang dapat diandalkan dalam mewujudkan teknik kriptografi algoritma simetri ini akan menghasilkan tingkat keamanan yang lebih tinggi terhadap record database karena dapat menyandikannya ke bentuk sandi dengan proses yang cukup rumit sehingga akan mempersulit kriptanalis untuk mengakses database tersebut penelitian ini akan menggunakan algoritma camellia bit untuk proses enkripsi dan dekripsinya sehingga dalam prosesnya perlu melalui beberapa tahap yang panjang agar dapat menghasilkan cipher akhirnya algoritma ini memerlukan ronde dimana setiap enam ronde harus memasuki fungsi <math>f</math> dan fungsi <math>f^{-1}</math> dan juga memerlukan sebanyak subkunci yaitu <math>k_w</math> <math>k</math> dan <math>k_l</math> penelitian ini menguraikan proses pengamanan record database dengan menyandikannya berdasarkan algoritma camellia sehingga dihasilkan teks record database dalam bentuk sandi yang sulit dipahami dan dimengerti oleh orang lain hal ini dilakukan sebagai upaya untuk meminimalisir tindakan tindakan penyalahgunaan record database.</p>	<p>record database dalam bentuk teks sangat banyak digunakan Sehingga sangat rentan terhadap pencurian data oleh pihak pihak yang tidak berhak demi menjaga keamanan record database dapat dilakukan dengan pemanfaatan teknik kriptografi teknik kriptografi dapat menyandikan record database dengan mengenkripsikannya ke dalam bentuk sandi sandi yang tidak dipahami algoritma camellia adalah salah satu dengan proses yang cukup rumit sehingga akan</p>



Abstrak	Teks Input	Hasil Tokenizing
		mempersulit kriptanalis untuk mengakses database tersebut penelitian ini akan menggunakan algoritma Yang dapat diandalkan dalam mewujudkan teknik kriptografi algoritma simetri ini akan menghasilkan tingkat keamanan yang lebih tinggi terhadap record database karena dapat menyandikannya ke bentuk sandi algoritma camellia bit untuk proses enkripsi dan dekripsinya sehingga dalam prosesnya perlu melalui beberapa tahap yang panjang agar dapat menghasilkan cipher akhirnya algoritma



Abstrak	Teks Input	Hasil Tokenizing
		ini
		memerlukan
		ronde
		dimana
		setiap
		enam
		ronde
		harus
		memasuki
		fungsi
		f
		dan
		fungsi
		f <sup>-1</sup>
		dan
		juga
		memerlukan
		sebanyak
		subkunci
		yaitu
		kw
		k
		dan
		kl
		penelitian
		ini
		menguraikan
		proses
		pengamanan
		record
		database
		dengan
		menyandikannya
		berdasarkan
		algoritma
		camellia
		sehingga
		dihasilkan
		teks
		record
		database
		dalam
		bentuk
		sandi
		yang
		sulit
		dipahami
		dan
		dimengerti
		oleh
		orang
		lain
		hal
		ini
		dilakukan
		sebagai
		upaya
		untuk
		meminimalisir
		tindakan
		tindakan



Abstrak	Teks Input	Hasil Tokenizing penyalahgunaan record database.
---------	------------	---

2. *Filtering (wordlist/stoplist)*

Tahap *filtering* merupakan tahap menghilangkan kata yang tidak penting (*stoplist*) atau menyimpan kata yang penting (*wordlist*). *Stoplist/stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan *bag-of-words*. Contoh *stopwords* adalah “yang”, “dan”, “di”, “dari” dan seterusnya. Data *stopword* dapat diambil dari suatukamus kata-kata yang menyimpan kata-kata yang bisa dihilangkan atau dengan kata lain kata-kata yang tidak penting disebut dengan kamus tala.

**Tabel 4.** Hasil *Filtering*

Abstrak	Hasil Tokenizing	Hasil Filtering
1	record database <del>dalam</del> bentuk teks <del>sangat</del> <del>banyak</del> <del>digunakan</del> <del>sehingga</del> <del>sangat</del> rentan <del>terhadap</del> pencurian data <del>oleh</del> <del>pihak</del> <del>pihak</del> yang <del>tidak</del> berhak <del>demi</del> menjaga keamanan record database <del>dapat</del> <del>dilakukan</del> <del>dengan</del> pemanfaatan teknik kriptografi teknik kriptografi <del>dapat</del> menyandikan record database <del>dengan</del> mengkripsikannya <del>ke</del> <del>dalam</del> bentuk sandi sandi yang <del>tidak</del> dipahami algoritma	record database bentuk teks rentan pencurian data berhak Menjaga keamanan record database pemanfaatan teknik kriptografi teknik kriptografi menyandikan record database mengkripsikannya bentuk sandi sandi dipahami algoritma camellia salah algoritma diandalkan mewujudkan teknik kriptografi algoritma simetri menghasilkan tingkat keamanan record database menyandikannya bentuk sandi proses rumit mempersulit kriptanalisis mengakses



Abstrak	Hasil Tokenizing	Hasil Filtering
	camellia	database
	<del>adalah</del>	penelitian
	salah	algoritma
	<del>satu</del>	camellia
	algoritma	bit
	<del>yang</del>	proses
	<del>dapat</del>	enkripsi
	diandalkan	dekripsinya
	<del>dalam</del>	prosesnya
	mewujudkan	tahap
	teknik	menghasilkan
	kriptografi	cipher
	algoritma	algoritma
	simetri	memerlukan
	<del>ini</del>	ronde
	<del>akan</del>	dimana
	menghasilkan	ronde
	tingkat	memasuki
	keamanan	fungsi
	<del>yang</del>	f
	<del>lebih</del>	fungsi
	<del>tinggi</del>	$f^{-1}$
	<del>terhadap</del>	memerlukan
	record	subkunci
	database	kw
	<del>karena</del>	k
	<del>dapat</del>	kl
	menyandikannya	penelitian
	ke	menguraikan
	bentuk	proses
	<del>beberapa</del>	pengamanan
	tahap	record
	<del>yang</del>	database
	<del>panjang</del>	menyandikannya
	<del>agar</del>	berdasarkan
	<del>dapat</del>	algoritma
	menghasilkan	camellia
	sandi	dihasilkan
	<del>dengan</del>	teks
	proses	record
	<del>yang</del>	Database
	<del>eukup</del>	bentuk
	rumit	sandi
	<del>sehingga</del>	sulit
	<del>akan</del>	dipahami
	mempersulit	dimengerti
	kriptanalisis	orang
	<del>untuk</del>	upaya
	mengakses	meminimalisir
	database	tindakan
	<del>tersebut</del>	tindakan
	penelitian	penyalahgunaan
	<del>ini</del>	record
	<del>akan</del>	database
	<del>menggunakan</del>	
	algoritma	
	camellia	
	bit	
	<del>untuk</del> proses	
	enkripsi	



Abstrak	Hasil Tokenizing	Hasil Filtering
	<del>dan</del> dekripsinya <del>sehingga</del> <del>dalam</del> prosesnya perlu melalui cipher akhirnya algoritma <del>ini</del> memerlukan ronde dimana setiap enam ronde harus memasuki fungsi f <del>dan</del> fungsi f <sup>-1</sup> <del>dan</del> juga memerlukan sebanyak subkunci yaitu kw k <del>dan</del> kl penelitian <del>ini</del> menguraikan proses pengamanan record database dengan menyandikannya berdasarkan algoritma camellia <del>sehingga</del> dihasilkan teks Record database <del>dalam</del> bentuk sandi yang sulit dipahami <del>dan</del> dimengerti <del>oleh</del>	



Abstrak	Hasil Tokenizing	Hasil Filtering
	orang	
	<del>lain</del>	
	<del>hal</del>	
	<del>ini</del>	
	<del>dilakukan</del>	
	sebagai	
	upaya	
	<del>untuk</del>	
	meminimalisir	
	tindakan	
	tindakan	
	penyalahgunaan	
	record	
	. database	

3. *Stemming*

Tahap *stemming* merupakan tahap mengubah kata imbuhan menjadi kata dasar atau dengan kata lain adalah mengembalikan kata ke dalam bentuk aslinya.

**Tabel 5.** Hasil *Stemming*

Abstrak	Hasil Filtering	Hasil Stemming
1	record	record
	database	database
	bentuk	bentuk
	teks	teks
	rentan	rentan
	<del>pencurian</del>	curi
	data	data
	<del>berhak</del>	hak
	<del>Menjaga</del>	jaga
	<del>keamanan</del>	aman
	record	record
	database	database
	<del>pemanfaatan</del>	manfaat
	teknik	teknik
	kriptografi	kriptografi
	teknik	teknik
	kriptografi	kriptografi
	<del>menyandikan</del>	sandi
	record	record
	database	database
	<del>mengkripsikannya</del>	enkripsi
	bentuk	bentuk
	sandi	sandi
	sandi	sandi
	<del>dipahami</del>	pahami
	algoritma	
	camellia	
	salah	
	algoritma	
	<del>diandalkan</del>	
	<del>Mewujudkan</del>	
	Teknik	
	kriptografi	algoritma
	algoritma	camellia
	simetri	salah
	<del>menghasilkan</del>	algoritma
	tingkat	andal
	<del>keamanan</del>	Wujud
	Record	teknik
	database	kriptografi



Abstrak	Hasil Filtering	Hasil Stemming
	<del>menyandikannya</del>	algoritma
	bentuk	simetri
	sandi	hasil
	proses	tingkat
	rumit	aman
	<del>mempersulit</del>	record
	kriptanalisis	database
	<del>mengakses</del>	sandi
	database	bentuk
	penelitian	sandi
	algoritma	proses
	camellia	rumit
	bit	sulit
	proses	kriptanalisis
	enkripsi	akses
	<del>dekripsinya</del>	database
	<del>prosesnya</del>	penelitian
	tahap	algoritma
	<del>menghasilkan</del>	camellia
	cipher	bit
	algoritma	proses
	<del>memerlukan</del>	enkripsi
	ronde	dekripsi
	dimana	proses
	ronde	tahap
	<del>memasuki</del>	hasil
	fungsi	cipher
		algoritma
		perlu
		ronde
		dimana
		ronde
		masuk
		fungsi
	f	f
	fungsi	fungsi
	$f^{-1}$	$f^{-1}$
	<del>memerlukan</del>	perlu
	subkunci	subkunci
	kw	kw
	k	k
	Kl	Kl
	penelitian	peneliti
	<del>menguraikan</del>	urai
	proses	proses
	<del>pengamanan</del>	aman
	record	record
	database	database
	<del>menyandikannya</del>	sandi
	<del>berdasarkan</del>	dasar
	algoritma	algoritma
	camellia	camellia
	<del>dihasilkan</del>	hasil
	teks	teks
	record	record
	database	database
	bentuk	bentuk
	sandi	sandi
	sulit	sulit
	<del>dipahami</del>	pahami



Abstrak	Hasil Filtering	Hasil Stemming
	<del>dimengerti</del>	erti
	orang	orang
	upaya	upaya
	<del>mminimalisir</del>	minimalisir
	<del>tindakan</del>	tindak
	<del>tindakan</del>	tindak
	<del>penyalahgunaan</del>	salahguna
	record	record
	database	database

4. *Tagging*

*Tagging* merupakan tahap untuk mencari bentuk awal/root dari tiap kata lampau atau hasil dari proses *stemming*. *Tagging* berfungsi untuk merubah kata dasar menjadi kata lampau. Proses ini pada umumnya dilakukan pada kata yang berbahasa inggris.

**Tabel 6.** Hasil *Tagging*

Abstrak	Hasil Stemming	Hasil Tagging
	record	record
	database	database
	bentuk	bentuk
	teks	teks
	rentan	rentan
	curi	curi
	data	data
	hak	hak
	jaga	jaga
	aman	aman
	record	record
	database	database
	manfaat	manfaat
	teknik	teknik
	kriptografi	kriptografi
	teknik	teknik
	kriptografi	kriptografi
	sandi	sandi
	record	record
	database	database
	enkripsi	enkripsi
	bentuk	bentuk
	sandi	sandi
	sandi	sandi
	pahami	pahami
	algoritma	algoritma
	camellia	camellia
	salah	salah
	algoritma	algoritma
	andal	andal
	Wujud	Wujud
	teknik	teknik
	kriptografi	kriptografi
	algoritma	algoritma
	simetri	simetri
	hasil	hasil
	tingkat	tingkat
	aman	aman
	record	record
	database	database
	sandi	sandi
	bentuk	bentuk
	sandi	sandi
	proses	proses



Abstrak	Hasil Stemming	Hasil Tagging
	rumit	rumit
	sulit	sulit
	kriptanalisis	kriptanalisis
	Akses	Akses
	database	database
	penelitian	penelitian
	algoritma	algoritma
	camellia	camellia
	bit	bit
	proses	proses
	enkripsi	enkripsi
	dekripsi	dekripsi
	proses	proses
	tahap	tahap
	hasil	hasil
	cipher	cipher
	algoritma	algoritma
	perlu	perlu
	ronde	ronde
	dimana	dimana
	ronde	ronde
	masuk	masuk
	fungsi	fungsi
	f	f
	fungsi	fungsi
	$f^{-1}$	$f^{-1}$
	perlu	perlu
	subkunci	subkunci
	kw	kw
	k	k
	Kl	Kl
	peneliti	peneliti
	urai	urai
	proses	proses
	aman	aman
	record	record
	database	database
		sandi
		dasar
		algoritma
		camellia
	sandi	hasil
	dasar	teks
	algoritma	record
	camellia	database
	hasil	bentuk
	teks	sandi
	record	sulit
	database	paham
	bentuk	erti
	sandi	orang
	sulit	upaya
	paham	minimalisir
	erti	tindak
	orang	tindak
	upaya	salahguna
	minimalisir	record
	tindak	database
	tindak	
	salahguna	



Abstrak	Hasil Stemming	Hasil Tagging
	record	
	database	

Pembobotan TF/IDF adalah suatu hubungan kata (*term*) yang berada pada dokumen yang akan diberikan suatu nilai bobot. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata di dalam sebuah dokumen hasil dari pemrosesan teks abstrak yang dibandingkan. Frekuensi kemunculan sebuah kata dalam sebuah dokumen disebut dengan *Term Frequency* (TF) dan *inverse* frekuensi dokumen yang mengandung kata yang disebut *Inverse Document Frequency* (IDF). Tujuan dari algoritma ini adalah untuk melihat seberapa penting kata (*term*) tersebut di dalam dokumen abstrak.

Setelah dilakukannya proses *text mining* dan menghasilkan kalimat yang memiliki peran penting didalam perhitungan, selanjutnya yaitu perhitungan TF-IDF. Proses menghitung nilai bobot, terlebih dahulu hitung jumlah kemunculan *term* pada setiap dokumen, kemudian menjumlahkan setiap kemunculan kata yang sama pada semua dokumen letakkan pada kolom DF. Selanjutnya, mencari IDF yaitu hitung jumlah dokumen dibagi DF kemudian hasil pembagian di-log-kan. Langkah selanjutnya, mencari WDT yaitu nilai TF dikalikan nilai IDF sehingga memperoleh nilai bobot *term*. Perhitungan untuk mendapatkan nilai TF, IDF dan wdt adalah sebagai berikut :

Langkah pertama yaitu mencari nilai TF

*Term* : record

D1 : 7

D1 bernilai 7 karena *term* record pada dokumen absrak terdapat 7 *term* yang sama.

Df = jumlah setiap kemunculan kata yang sama pada semua dokumen

Mencari nilai IDF :

n = jumlah dokumen

df = jumlah kemunculan *term* pada setiap dokumen abstrak

$$IDF = \text{Log} (n/df) + 1$$

$$= \text{Log} (50/1) + 1$$

$$= 2,698970004$$

Mencari nilai wdt :

Wdt = tf.idf

D1 :

$$Wdt = 7 \times 2,698970004$$

$$= 18,89279003$$

Berdasarkan diatas juga dilakukan untuk mencari *term* yang lainnya, berikut hasil keseluruhan proses dari perhitungan TF-IDF yang disajikan pada tabel 6.

**Tabel 7.** Hasil Pembobotan kata

Term	Tf	Df	N	Idf $\frac{1}{\log(n/df) + 1}$	Wdt = Tf.Idf
record	7	1	50	2,698970004	18,89279003
Database	8	4	50	2,096910013	16,7752801
bentuk	4	10	50	1,698970004	6,795880017
teks	2	6	50	1,920818754	3,841637508
rentan	1	2	50	2,397940009	2,397940009
Curi	1	4	50	2,096910013	2,096910013
data	1	26	50	1,283996656	1,283996656
Hak	1	3	50	2,22184875	2,22184875
Jaga	1	4	50	2,096910013	2,096910013
Teknik	3	13	50	1,585026652	4,755079956
kriptografi	3	3	50	2,22184875	6,665546249
Sandi	7	5	50	2	14
Record	7	1	50	2,698970004	18,89279003
database	8	4	50	2,096910013	16,7752801
Enkripsi	2	2	50	2,397940009	4,795880017
bentuk	4	10	50	1,698970004	6,795880017
sandi	7	5	50	2	14
sandi	7	5	50	2	14
Paham	2	4	50	2,096910013	4,193820026
algoritma	6	20	50	1,397940009	8,387640052
camellia	3	2	50	2,397940009	7,193820026
salah	1	23	50	1,337242168	1,337242168
Algoritma	6	20	50	1,397940009	8,387640052



Term	Tf	Df	N	$\frac{Idf}{\log(n/df) + 1}$	Wdt = Tf.Idf
Andal	1	3	50	2,22184875	2,22184875
Wujud	1	2	50	2,397940009	2,397940009
teknik	3	13	50	1,585026652	4,755079956
kriptografi	3	3	50	2,22184875	6,665546249
algoritma	6	20	50	1,397940009	8,387640052
simetri	1	1	50	2,698970004	2,698970004
Hasil	3	24	50	1,318758763	3,956276288
dimana	1	10	50	1,698970004	1,698970004
ronde	2	2	50	2,397940009	4,795880017
masuk	1	6	50	1,920818754	1,920818754
fungsi	2	3	50	2,22184875	4,443697499
f	1	2	50	2,397940009	2,397940009
Fungsi	2	3	50	2,22184875	4,443697499
Fl	1	2	50	2,397940009	2,397940009
subkunci	1	2	50	2,397940009	2,397940009
Kw	1	2	50	2,397940009	2,397940009
k	1	2	50	2,397940009	2,397940009
urai	1	2	50	2,397940009	2,397940009
proses	4	29	50	1,236572006	4,946288026
Aman	3	6	50	1,920818754	5,762456262
record	7	1	50	2,698970004	18,89279003
database	8	4	50	2,096910013	16,7752801
Sandi	7	5	50	2	14
dasar	1	16	50	1,494850022	1,494850022
algoritma	6	20	50	1,397940009	8,387640052
camellia	3	2	50	2,397940009	7,193820026
Hasil	3	24	50	1,318758763	3,956276288
teks	2	6	50	1,920818754	3,841637508
record	7	1	50	2,698970004	18,89279003
Database	8	4	50	2,096910013	16,7752801
bentuk	4	10	50	1,698970004	6,795880017
sandi	7	5	50	2	14
Sulit	2	5	50	2	4
Paham	2	4	50	2,096910013	4,193820026
erti	1	5	50	2	2
orang	1	11	50	1,657577319	1,657577319
upaya	1	6	50	1,920818754	1,920818754
minimalisir	1	2	50	2,397940009	2,397940009
tindak	2	2	50	2,397940009	4,795880017
tindak	2	2	50	2,397940009	4,795880017
Salahguna	1	16	50	1,494850022	1,494850022
record	7	1	50	2,698970004	18,89279003
database	8	4	50	2,096910013	16,7752801

**Tabel 8.** Hasil Pengelompokan Kategori Skripsi

Hasil Klasifikasi							
No	Term	TF-IDF	Kompresi	Pengolahan Citra	SPK	Kriptografi	Data Mining
1	<i>record</i>	18,89	0	0	0	0	0
2	<i>database</i>	16,77	0	0	0	0	16,77528
5	rentan	2,397	0	0	0	2,39794	0
6	curi	2,096	0	0	0	2,09691	0
7	data	1,284	1,283997	0	0	1,2839967	1,283996
8	hak	2,221	0	0	0	0	0
9	jaga	2,096	0	0	0	2,09691	0
10	aman	5,762	0	0	0	5,7624563	0
11	manfaat	2,397	0	0	0	2,39794	0
12	teknik	4,755	4,75508	0	0	4,75508	4,7550799
13	kriptografi	6,665	0	0	0	6,6655462	0
14	sandi	14	0	0	0	14	0



Hasil Klasifikasi							
No	Term	TF-IDF	Kompresi	Pengolahan Citra	SPK	Kriptografi	Data Mining
15	enkripsi	4,795	0	0	0	4,79588	0
16	paham	4,193	0	0	0	4,19382	0
17	algoritma	8,3876	8,38764	0	0	8,3876401	8,38764005
18	camellia	7,1938	0	0	0	7,19382	0
19	salah	1,3372	0	1,337242168	1,337	1,3372422	1,33724217
20	andal	2,2218	0	0	0	2,2218487	0
21	wujud	2,3979	0	0	0	2,39794	0
22	simetri	2,699	0	0	0	0	0
23	hasil	3,9563	0	3,956276288	0	3,9562763	3,95627629
24	tingkat	1,4202	0	1,420216403	0	1,4202164	1,4202164
25	proses	4,9463	4,946288	4,946288026	0	4,946288	0
26	rumit	2,2218	0	0	0	2,2218487	0
27	sulit	4	0	0	0	4	0
28	kriptanalisis	2,3979	0	0	0	2,39794	0
29	akses	2,2218	0	0	0	2,2218487	0
30	peneliti	3,1701	3,170053	0	0	3,1700533	0
31	bit	1,7959	1,79588	0	0	1,79588	0
32	dekripsi	2,3979	0	0	0	2,39794	0
33	tahap	2,2218	0	0	0	2,2218487	0
34	cipher	2,3979	0	0	0	2,39794	0
35	ronde	4,7959	0	0	0	4,79588	0
36	dimana	1,699	0	0	0	1,69897	0
37	masuk	1,9208	0	0	0	1,9208188	0
39	f	2,3979	0	0	0	2,39794	0
40	Fl	2,3979	0	0	0	2,39794	0
41	subkunci	2,3979	0	0	0	2,39794	0
42	Kw	2,3979	0	0	0	2,39794	0
43	k	2,3979	0	0	0	2,39794	0
44	Kl	2,3979	0	0	0	2,39794	0
45	urai	2,3979	0	0	0	2,39794	0
46	dasar	1,4949	0	1,494850022	0	1,49485	0
47	erti	2	0	2	0	2	0
48	orang	1,6576	0	0	0	1,6575773	0
49	upaya	1,9208	0	0	0	1,9208188	1,92081875
50	minimalisir	2,3979	0	0	0	2,39794	0
51	tindak	4,7959	0	0	0	4,79588	0
52	salahguna	1,4949	0	0	0	1,49485	0
Total			24,33894	15,15487291	1,337	157,33579	39,8365504

Dari hasil tabel 8. dapat dilihat bahwa pengklasifikasian kategori dari dokumen abstrak skripsi pada contoh kasus memiliki nilai tertinggi pada kategori kriptografi, maka dokumen abstrak tersebut tergolong pada kategori kriptografi. Berikut hasil pengelompokan dari keseluruhan dokumen abstrak skripsi kampus STMIK Budi Darma:

**Tabel 9.** Daftar Judul Skripsi dan kelompok

No	Judul	Tahun	Kelompok
1	Implementasi Algoritma Camellia Pada Penyandian Record Database	2018	Kriptografi
2	Implementasi Algoritma Camellia Pada Penyandian Citra Usg	2018	Kriptografi
3	Penerapan Algoritma Golom Coding Pada Aplikasi Kompresi Short Message Service (Sms)	2019	Kompresi
4	Implementasi Algoritma Prediction By Partial Matching Pada Kompresi File Teks Terenkripsi Elgamal	2017	Kompresi
5	Perancangan Aplikasi Kompresi Adiktif Citra Usg Menggunakan Algoritma C-Mean	2019	Pengolahan Citra
6	Implementasi Algoritma Base64 Untuk Mengamankan File Secara Online	2019	Kompresi
7	Implementasi Algoritma Quadtree Pada Game Bubble Match	2019	Data Mining
8	Implementasi Algoritma Hybrid Case Based Pemecahan Masalah PC Mesin Potocopy Konika Minolta (Studi Kasus : CV.Perdana Mandiri)	2019	Pengolahan Citra
9	Data Mining Pengarsipan Dokumen Karyawan Ybilaris Dengan Metode Alphabetic Filing System ( Studi Kasus : Pt. Perkebunan Nusantara Iv ( Ptpn Iv Kebun Adolina)	2019	Pengolahan Citra



No	Judul	Tahun	Kelompok
10	Penerapan Algoritma Rough Set Untuk Mengidentifikasi Faktor - Faktor Resignnya Karyawan ( Study Kasus Pt. Sumber Alfaria Trijaya, Tbk )	2019	Data mining
11	Perancangan Aplikasi Pengendalian Bahan Baku Pada Crystal Jade Restaurant Dengan Menggunakan Metode Material Requirement Planning (Mrp)	2019	SPK
12	Implementasi Data Mining Algoritma C 5.0 Dalam Sppt-Sni Air Minum Dalam Kemasan (Studi Kasus : Baristand)	2019	Data Mining
13	Perancangan Aplikasi Optimalisasi Jumlah Persediaan Bumbu Penyedap Masakan Dengan Menggunakan Metode Simpleks (Studi Kasus : Pt. Ajinomoto Sales Indonesia)	2018	SPK
14	Implementasi Data Mining Untung Mendukung Perkerasan Jalan Di Kota Medan Dengan Metode Internasional Roughness Index	2018	Data Mining
15	Implementasi Data Mining Untuk Pola Penjualan Produk Makanan Ringan Menggunakan Assosiation Rule Pada Swalayan Maju Bersama	2018	Data Mining
16	Implementasi Metode Internal Rate Of Return (Irr) Dalam Menentukan Nilai Investasi Emas	2019	Data Mining
17	Analisa Perbandingan Algoritma Fp-Growth Dan Algoritma Cut Both Ways Dalam Menentukan Pola Penjualan Handphone	2019	Kompresi
18	Optimalisasi Biaya Transportasi Pengiriman Barang Dengan Menggunakan Metode Potensial Pada Pt. Bintang Mutiara Cemerlang	2019	Data Mining
19	Penyelesaian Sistem Linear Untuk Menghitung Keuntungan Maksimum Menggunakan Metode Interpolasi Polinomial Newton Pada Pt. Intra Cakrawala Teknindo	2017	Kriptografi
20	Penerapan Data Mining Identifikasi Tingkat Trauma Korban Bencana Alam Letusan Gunung Sinabung Menggunakan Algoritma Apriori	2017	Data Mining
21	Penerapan Metode Distribution Requirement Planning (DRP) Pada Sistem Informasi Penmdistribusian Barang PT. Singer	2018	Kompresi
22	Implementasi Algoritma Apriori Untuk Memprediksi Penyewaan Alat Berat Pada Pt. Karya Nusantara	2017	Data Mining
23	Penerapan Algoritma Aac Adaptif Arithmetic Coding Pada Aplikasi Kamus Teknologi Informasi Berbasis Android	2019	Kompresi
24	Penerapan Algoritma Fibonacci Codes Pada Kompresi Aplikasi Audio Mp3 Berbasis Dekstop	2019	Kompresi
25	Penerapan Metode Symmetric Codes Dalam Pengompresian File Docx	2018	Kompresi
26	Perancangan Aplikasi Konversi File Image Hasil Scan Menjadi File Text Menggunakan Metode Feature Extraction	2018	Kompresi
27	Penerapan Er Mapper Pada Citra Digital Untuk Menentukan Kualitas Minyak Pada Biji Sawit	2019	Pengolahan Citra
28	Implementasi Logarithmic Image Processing Pada Color Constancy Citra Mozaik	2019	Pengolahan Citra
29	Perbaikan Citra Rgb Dengan Menggunakan Metode Homomorphic Filtering Menggunakan Butterworth Filter	2019	Pengolahan Citra
30	Implementasi Perbaikan Kualitas Citra Dengan Metode Geometric Median Filter Dan Penyisipan Teks Pada Gambar Dengan Metode LSB (Least Significant Bit)	2018	Pengolahan Citra
31	Perancangan Aplikasi Perbaikan Citra Mata Dengan Menggunakan Metode Contrast Streching	2017	Pengolahan Citra
32	Implementasi Algoritma Clipping Cohen Sutherland Pada Pematongan Objek Pada Citra	2018	Pengolahan Citra
33	Perancangan Aplikasi Penyembunyian File Teks Pada Citra Gif Dengan Menerapkan Algoritma Eztego	2017	Kriptografi
34	Perancangan Sistem Pendukung Keputusan Pengujian Dan Sertifikasi Mutu Barang Hasil Perkebunan Dengan Metode Fuzzy Mamdani	2019	Data Mining
35	Sistem Pendukung Keputusan Pemilihan Mr. Celebrity Dan Mrs Celebrity Sumut Dengan Menerapkan Metode Waspa (Studi Kasus: Rumah Bakat Trend Management)	2019	SPK
36	Sistem Pendukung Keputusan Murabahah Emas Islamic Bank (Ib) Dengan Menerapkan Metode Ahp (Studi Kasus : Pt. Bank Bni Syariah Medan )	2019	Kompresi
37	Sistem Pendukung Keputusan Pemilihan Sekolah Smp Negeri Terbaik Dalam Bidang Akademis Pada Kantor Dinas Pendidikan Kabupaten Deli Serdang Menerapkan Metode Multifactor Evaluation Process (Mfep)	2019	SPK



No	Judul	Tahun	Kelompok
38	Analisa Efisiensi Metode Moora Pada Penentuan Karyawan Terbaik Menerapkan Algoritma Big O Notation	2019	Kriptografi
39	Sistem Pendukung Keputusan Seleksi Instruktur Pada Lkp Prima Menggunakan Metode Topsis	2019	SPK
40	Aplikasi Sistem Pendukung Keputusan Penentuan Penerina Dana Ppa Dengan Menerapkan Metode Electre (Studi Kasus : Stmik Royal Kisaran)	2019	SPK
41	Sistem Pendukung Keputusan Pemilihan Customer Service Pada Pt. Global Media Nusantara Menggunakan Metode Ahp (Analytical Hierarchy Process)	2017	SPK
42	Perancangan Aplikasi Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik Pada Cv. Multi Engineering Dengan Menggunakan Analytical Hierarchy Process	2017	Data Mining
43	Sistem Pendukung Keputusan Dalam Pemilihan Dokter Terbaik Menggunakan Metode Analyst Network Process (Anp) Berbasis Android (Studi Kasus : Rsu Bhakti Medan)	2017	Pengolahan Citra
44	Sistem Pendukung Keputusan Dalam Pemilihan Mahasiswa Terbaik Untuk Program Student Exchange Menggunakan Metode Electre (Studi Kasus : Universitas Hkbp Nommensen)	2018	SPK
45	Sistem Pendukung Keputusan Penyeleksian Colour Guard Pada Marching Band GINADA Dengan Menggunakan Metode VIKOR Dan Borda N	2018	SPK
46	Sistem Pendukung Keputusan Pemilihan Workshop Leader Di Gereja Dengan Menggunakan Metode Simple Multy Attribute Rating Technique (Studi Kasus : Gereja Pentakosta Indonesia Sidang Marindal Garu Viii Medan)	2017	SPK
47	Sistem Pakar Diagnosa Diseksi Aorta Menerapkan Metode Fuzzy Sugeno	2019	Data Mining
48	Prediksi Tingkat Peredaran Kosmetik Palsu Menggunakan Metode Naive Bayes (Studi Kasus : Bpom Kota Medan)	2019	Data Mining
49	Sistem Pakar Diagnosa Iskemia Menerapkan Metode Fuzzy Sugeno	2019	Pengolahan Citra
50	Sistem Pakar Mendiagnosa Penyakit Lupus Menggunakan Metode Constraint Satisfaction Problem (Csp)	2019	SPK

Berdasarkan tabel diatas memperlihatkan hasil pengelompokkan menggunakan penerapan *text mining*, tf-idf dan *cosine similarity*. Hasil dari ketiga proses tersebut tidak semua skripsi masuk kedalam kategori yang sesuai.

#### 4. KESIMPULAN

Dari berbagai penjelasan yang telah diuraikan dalam laporan, maka dapat disimpulkan berbagai hal Algoritma *text mining* dan term *frequency – inverse document frequency* (TF-IDF) mampu melakukan klasifikasi skripsi. Adanya proses *cosine similarity* dikarenakan dibutuhkan untuk membandingkan antar abstrak. Semakin banyak keyword yang terdapat pada abstrak maka semakin baik untuk mengelompokkan

#### REFERENCES

- [1] M. Nurjannah and I. F. Astuti, "PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY ( TF-IDF ) UNTUK TEXT MINING," vol. 8, no. 3, pp. 110–113, 2013.
- [2] M. A. Ariyanti, A. P. Wibawa, and U. Pujiyanto, "Metode term frequency - invers document frequency pada mekanisme pencarian judul skripsi," *Tekno*, vol. 28, no. 2, p. 177, 2019.
- [3] R. T. Wahyuni, D. Prastiyanto, and E. Suprpto, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi," vol. 9, no. 1, 2017.
- [4] M. Mahdi, "Penerapan Metode Cosine Similarity dan Pembobotan TF / IDF pada Sistem Klasifikasi Sinopsis Buku di Perpustakaan Kejaksaan Negeri Jember," pp. 31–42.
- [5] R. A. Sasmita, A. Z. Falani, F. I. Komputer, U. N. Surabaya, and T. Mining, "Pemanfaatan algoritma tf/idf pada sistem informasi ecomplaint handling," vol. 27, no. 1, pp. 27–33, 2018.
- [6] N. Agusvina and N. Santoso, "Pengelompokan Artikel Berbahasa Indonesia Dengan Menggunakan Reduksi Fitur Information Gain Thresholding Dan K-Means," vol. 2, no. 10, pp. 3822–3828, 2018.
- [7] L. M. Bening Herwijayanti<sup>1</sup>, Dian Eka Ratnawati<sup>2</sup>, "Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity," p. 308, 2018.
- [8] C. S. Sitti Munifah, Abdul Syukur, "PENGELOMPOKAN ARSIP UNIVERSITAS MENGGUNAKAN ALGORITMA K-MEANS DENGAN FEATURE SELECTION CHI SQUARE," *J. Teknol. Inf.*, vol. 11, pp. 160–171, 2015.