p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/informatika

Penerapan *Clustering* pada Aplikasi Pendeteksi Kemiripan Dokumen Teks Bahasa Indonesia

Gabriella E. I. Kambey, Rizal Sengkey, Agustinus Jacobus
Jurusan Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu, 95115, Indonesia
kambeygebby@gmail.com, rizalsengkey@gmail.com, a.jacobus@unsrat.ac.id
Diterima: 25 Juni 2020; direvisi: 27 Juni 2020; disetujui: 30 Juni 2020

Abstract* — *A text document is something written or printed that can be used as an explanation as well as to make certain information. The development of technology which is increasingly advanced and fast-paced, makes it easier for people to get information and is vulnerable to taking writing which is commonly known as an act of plagiarism. Plagiarism level information technology measuring a text document is related to information retrieval from large amounts of data. It takes a long time to process the resemblance of the entire contents of the text document. For this purpose, a similar document detection application was made by applying the clustering method. Based on the results of testing of applications that have been made, clustering applications can shorten the measurement time similarity, but the application of this technique causes a decrease in the level of accuracy compared to if not applying the clustering technique.

***Keywords* — *Clustering*; *K-Means*; *Text Document*; *Text Mining*.**

Abstrak — Dokumen teks adalah sesuatu yang tertulis atau tercetak yang dapat digunakan sebagai keterangan juga untuk membuat suatu informasi tertentu. Perkembangan teknologi yang semakin maju dan serba instan, mempermudah orang untuk mendapat informasi dan rentan terjadi pengambilan karya tulis yang biasa dikenal sebagai tindakan plagiarisme. Teknologi informasi pengukur tingkat plagiat suatu dokumen teks berhubungan dengan pencarian informasi dari data yang banyak. Butuh waktu yang lama untuk memproses hasil kemiripan dari seluruh isi dokumen teks. Untuk itu dibuat aplikasi pendeteksi kemiripan dokumen teks dengan menerapkan metode *clustering*. Berdasarkan hasil pengujian dari aplikasi yang telah dibuat, aplikasi *clustering* dapat mempersingkat waktu pengukuran kesamaan, tetapi penerapan teknik ini menyebabkan penurunan tingkat akurasi dibandingkan jika tidak menerapkan teknik *clustering*.

Kata kunci — *Clustering*; Dokumen Teks; *K-Means*; *Text Mining*.

I. PENDAHULUAN

Dokumen teks adalah sesuatu yang tertulis atau tercetak yang dapat digunakan sebagai keterangan juga untuk memuat suatu informasi tertentu. Karya-karya tulis yang sering kita dapati dalam bentuk digital bisa juga disebut dengan dokumen teks. Setiap karya tulis yang dituliskan oleh penulisnya sendiri memiliki hak cipta atas tiap tulisannya sendiri. Banyak orang yang mencari informasi untuk kebutuhan tugas, penelitian, maupun karya ilmiah dari karya tulis yang banyak dipublikasikan. Dengan banyaknya karya tulis yang dipublikasi secara bebas dan gratis membuat beberapa orang mengambil karya tulis tersebut secara bebas tanpa memikirkan hak cipta penulis tersebut. Tindakan mencuri karya tulis maupun pikiran atau gagasan penulis lain disebut plagiarisme. Tindakan plagiarisme ini sangat merugikan orang pertama yang mempunyai pikiran karya tulis atau pemikiran tersebut.

Perkembangan zaman sekarang yang semakin maju dan serba instan, membuat peranan teknologi semakin luas dan mencakup segala bidang. Teknologi informasi pengukur tingkat plagiat suatu dokumen teks berhubungan dengan pencarian informasi dari data yang banyak. *Text Mining* merupakan pengolahan data berupa teks dalam mencari suatu informasi dari banyaknya data. Tujuan dari *text mining* adalah untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antar dokumen lainnya. Salah satu metode dalam *text mining* adalah *clustering*, yang merupakan metode untuk mengelompokan data. Butuh waktu yang lama untuk memproses hasil kemiripan dokumen.

Berdasarkan latar belakang masalah yang ada maka perlu dikembangkan suatu aplikasi untuk mendeteksi kemiripan isi dokumen teks dengan memisahkan tiap kalimatnya menjadi data tersendiri dan mengelompokan tiap data kalimat yang ada pada isi dokumen teks untuk mempercepat waktu komputasi.

A. *Plagiarisme*

Plagiarisme atau sering disebut plagiat adalah penjiplakan atau pengambilan karangan, pendapat dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri.[1] Plagiat dapat dianggap sebagai tindak pidana karena mencuri hak cipta orang lain. Di dunia Pendidikan, pelaku plagiarism dapat mendapat hukuman berat seperti dikeluarkan dari sekolah/universitas. Pelaku plagiat disebut sebagai plagiator. Singkat kata, plagiat adalah pencurian karangan milik orang lain.[2] Dapat juga diartikan sebagai pengambilan karangan (pendapat dan sebagainya) orang lain yang kemudian dijadikan seolah-olah miliknya sendiri.[3] Setiap karangan yang asli dianggap sebagai hak milik si pengarang dan tidak boleh dicetak ulang tanpa izin yang mempunyai hak atau penerbit karangan tersebut.

B. *Text Mining*

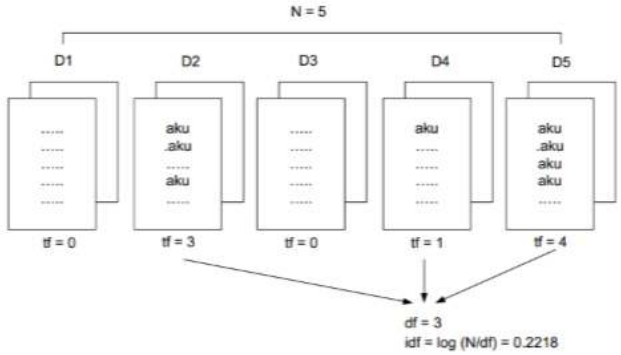
Text Mining adalah proses menemukan hal baru, yang sebelumnya tidak diketahui, mengenai informasi yang berpotensi untuk diambil manfaatnya dari sumber data yang tidak terstruktur mencakup dokumen bisnis, komentar customer, halaman web dan file XML.[4] *Text mining* hampir sama dengan data mining dalam hal tujuan dan proses, tapi pada *text mining* inputnya adalah file data tidak terstruktur seperti dokumen dalam bentuk word, PDF, XML, dan sebagainya.[5]

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 struktur. Adapun tugas khusus dari *text mining* antara lain, yaitu pengkategorisasian teks (*text categorization*) dan pengelompokan teks (*text clustering*).[6]

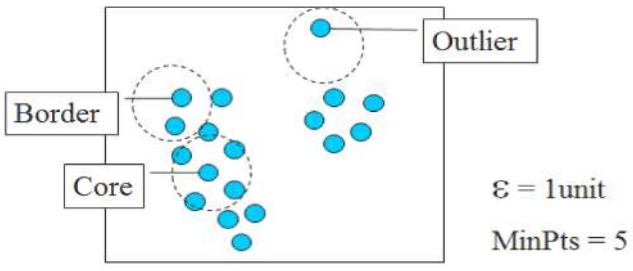
C. *Text Processing*

Text Processing berfungsi mengubah data tekstual yang tidak terstruktur ke dalam data terstruktur dan disimpan dalam basis data. Tahapan awal dari *text mining* adalah *text pre-processing* yang bertujuan untuk mempersiapkan teks menjadi data terstruktur dan dapat diproses pada tahap berikutnya.[7] Tahapan-tahapan dari *text pre-processing* adalah sebagai berikut:[8]

1) *Cleansing* adalah proses memperbaiki atau menghapus catatan yang rusak atau tidak akurat dari kumpulan dokumen, tabel, atau basis data dan mengacu pada pengidentifikasian bagian data yang tidak lengkap, salah, tidak akurat atau tidak relevan dari data dan kemudian mengganti, memodifikasi, atau menghapus data yang tidak diperlukan.
2) *Case folding* adalah proses mengubah mengubah seluruh huruf dari 'a' sampai dengan 'z' dalam dokumen menjadi huruf kecil.
3) *Tokenizing* merupakan tahap untuk memotong string input berdasarkan tiap kata yang menyusunnya.
4) *Filtering* adalah tahap mengambil kata-kata penting dari hasil tokenizing menggunakan algoritma stopword removal dengan membuang kata-kata yang kurang penting.
5) *Stemming* adalah tahap mencari roof kata dari tiap kata hasil filtering.



Gambar 1. Ilustrasi Metode TF



Gambar 2. *Core* dan *Border*

D. *Pembobotan Kata (Term Frequency)*

Term frequency merupakan salah satu metode untuk menghitung bobot tiap term dalam teks. Ilustrasi metode TF dapat dilihat pada gambar 1. Dalam metode ini, tiap term diasumsikan memiliki nilai kepentingan yang sebanding dengan jumlah kemunculan term tersebut pada teks (Mark Hall & Lloyd Smith, 1999).[9] Bobot sebuah term t pada sebuah teks dirumuskan dalam persamaan berikut:

$$W(d,t) = TF(d,t) \quad (1)$$

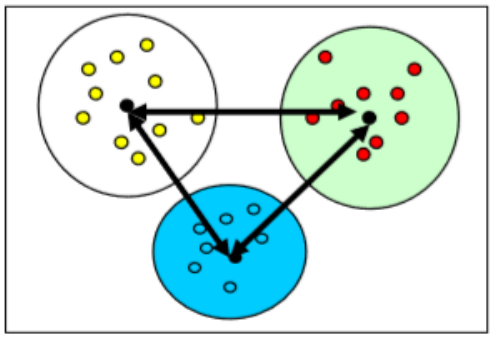
Dimana TF (d, t) adalah *term frequency* dari *term* t di teks d. *term frequency* dapat memperbaiki nilai *recall* pada *information retrieval*, tetapi tidak selalu memperbaiki nilai *precision*. Hal ini disebabkan *term* yang *frequent* cenderung muncul di banyak teks, sehingga term-term tersebut memiliki kekuatan diskriminatif/keunikan yang kecil. Untuk memperbaiki permasalahan ini, term dengan nilai frekuensi yang tinggi sebaiknya dibuang dari set *term*. Menemukan *threshold* yang optimal merupakan fokus dari metode ini.

E. *Density Based Spatial Clustering of Application with Noise (DBSCAN)*

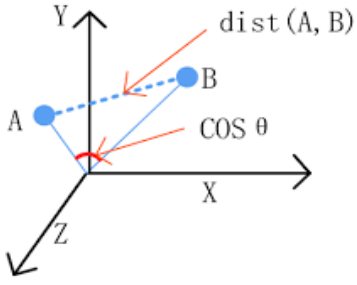
DBSCAN mendefinisikan *cluster* sebagai himpunan maksimum dari titik-titik kepadatan yang terkoneksi (*density-connected*). Semua objek yang tidak masuk ke dalam *cluster* manapun dianggap sebagai *noise*. [10] Gambaran proses pengelompokan dapat dilihat pada gambar 2.

DBSCAN menentukan sendiri jumlah *cluster* yang akan dihasilkan sehingga kita tidak perlu lagi untuk menentukan jumlah *cluster* yang diinginkan, tapi memerlukan 2 *input* lain, yaitu: [11]

MinPts : minimal banyak items dalam suatu cluster
Eps : nilai untuk jarak antar-items yang menjadi dasar pembentukan *neighborhood* dari suatu titik item



Gambar 3. *Clustering K-Means*



Gambar 4. *Cosine Similarity*

Adapun urutan algoritma dari DBSCAN secara umum memiliki 6 langkah, yaitu:

1) Pilih *point* p awal secara acak
2) Ambil semua point yang density reachable terhadap titik p
3) Jika p adalah *core point* maka cluster terbentuk
4) Jika p adalah *border point*, tidak ada yang merupakan hubungan *density-reachable* dari p dan DBSCAN akan mengunjungi *point* selanjutnya dari *database*
5) Lanjutkan proses sampai semua *point* telah diproses
6) Hasil yang didapatkan tidak tergantung dari urutan dari proses point yang diambil

F. *Artificial Intelligence*

Kecerdasan buatan (*Artificial Intelligence*) adalah kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah atau bisa disebut juga intelegensi artifisial, didefinisikan sebagai kecerdasan entitas ilmiah. Menurut Andreas Kaplan dan Michael Haenlein mendefinisikan kecerdasan buatan sebagai kemampuan sistem untuk menafsirkan data eksternal dengan benar, untuk belajar dari data tersebut dan menggunakan pembelajaran tersebut guna mencapai tujuan dan tuggas tertentu melalui adaptasi yang fleksibel. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia.

G. *Machine Learning*

Pembelajaran mesin (*machine learning*) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*). *Machine Learning* adalah disiplin ilmu yang mencakup perancangan dan pengembangan algoritma yang memungkinkan komputer untuk mengembangkan perilaku yang didasarkan pada data empiris, seperti dari sensor data basis data. Sistem pembelajar dapat memanfaatkan contoh (data) untuk menangkap ciri yang diperlukan dari probabilitas yang mendasarinya (yang tidak diketahui). Data dapat dilihat sebagai contoh yang menggambarkan hubungan antara variabel yang diamati. Fokus besar penelitian pembelajaran mesin adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Pada tahun 1959, Arthur Samuel mendefinisikan bahwa pembelajaran mesin adalah bidang studi yang memberikan kemampuan untuk belajar tanpa diprogram secara eksplisit. Kemampuan belajar yang menjadi dominan ditentukan oleh kemampuan perangkat lunak atau algoritmanya. Pembelajaran mesin dapat berfungsi untuk beradaptasi dengan suatu keadaan yang baru, serta untuk mendeteksi dan memperkirakan suatu pola. Algoritma dalam pembelajaran mesin dapat dikelompokkan berdasarkan masukan dan keluaran yang diharapkan dari algoritma, yaitu pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*), pembelajaran semi-terarah (*semi-supervised learning*), *reinforcement learning*.

H. *Deep Learning*

Deep learning yang dikenal dengan istilah pembelajaran struktur mendalam atau pembelajaran hierarki adalah salah satu cabang dari ilmu pembelajaran mesin (*machine learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam *deep learning* dapat digunakan baik untuk kebutuhan pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi-terarah (*semi-supervised learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya. Model pada *deep learning* pada dasarnya dibangun berdasarkan jaringan saraf tiruan, yang risetnya sudah berlangsung sejak era 80-an namun baru-baru ini kembali bangkit dengan adanya komputer yang semakin cepat apalagi ditambah dengan kemampuan kartu grafis modern yang mampu melakukan kalkulasi berbasis matriks secara simultan.

I. *K-Means Clustering*

Data *Clustering* merupakan salah satu metode data mining yang bersifat tanpa arahan (*unsupervised*). Ada dua jenis data *clustering* yang sering dipergunakan dalam proses pengelompokan data, yaitu *hierarchical* data *clustering* (hirarki) dan *non-hierarchical* data *clustering* (non hirarki). *K-Means* merupakan salah satu metode data *clustering non-hierarchical* yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih cluster.[12]

Metode ini mempartisi data ke dalam *cluster* sehingga data yang memiliki karakteristik yang sama dikelompokan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokan ke dalam kelompok yang lain seperti pada gambar 3. Adapun tujuan dari data *clustering* ini adalah untuk meminimalisir variasi antar *cluster*. Manfaat *clustering* adalah sebagai Identifikasi *Object* (*Recognition*) misalnya dalam bidang *Image Processing*, *Computer Vision* atau robot *vision*. Selain itu adalah sebagai Sistem Pendukung Keputusan dan *Data Mining* seperti segmentasi pasar, pemetaan wilayah, manajemen marketing, dll. [12]

Data *clustering* menggunakan metode *K-Means* ini secara umum dilakukan dengan algoritma dasar sebagai berikut (Yudi Agusta, 2007):

1) Tentukan jumlah cluster
2) Alokasikan data ke dalam cluster secara random
3) Hitung centroid/rata-rata dari data yang ada di masing-masing cluster
4) Alokasikan masing-masing data ke centroid/ rata-rata terdekat
5) Kembali ke tahap 3, apabila masih ada data yang berpindah cluster atau apabila perubahan nilai *centroid*, ada yang di atas nilai *threshold* yang ditentukan atau apabila perubahan nilai pada *objective function* yang digunakan di atas nilai *threshold* yang ditentukan.

J. *Cosine Similarity*

Metode *cosine similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antara dua buah objek. Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure* seperti pada gambar 4. Metode *cosine similarity* ini menghitung similarity antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah vektor dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran.[13]

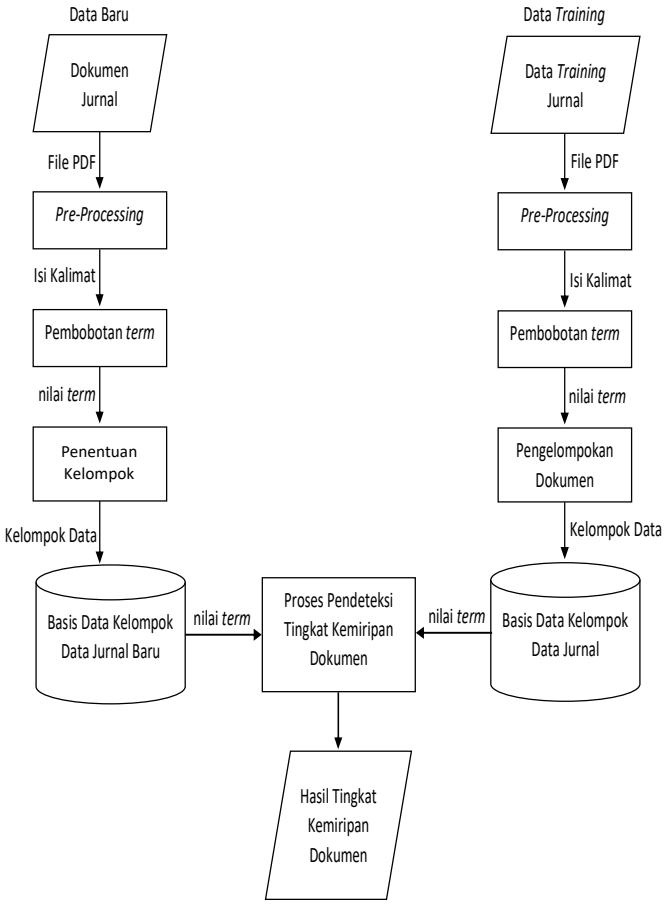
$$CosSim(d_i, q_i) = \frac{q_i . d_i}{|q_i||d_i|} = \frac{\sum_j^t=(q_{ij} . d_{ij})}{\sqrt{\sum_j^t=1(q_{ij})^2 . \sum_j^t=1(d_{ij})^2}} \quad (2)$$

Keterangan:
q_{ij} = bobot istilah *j* pada dokumen $i = tf_{ij} \, . idf_j$
d_{ij} = bobot istilah *j* pada dokumen $i = tf_{ij} \, . idf_j$

Hasil kemiripan yang berkisaran -1 berarti sangat berlawanan, 1 berarti sama persis, dan 0 menunjukkan ortogonal (tegak lurus), sedangkan nilai diantaranya menunjukkan kesamaan atau ketidaksamaan sedang. Untuk persamaan teks, vektor atribut d dan q biasanya merupakan term frekuensi dari dokumen. *Cosine similarity* dapat digunakan sebagai metode normalisasi dokumen yang panjang selama perbandingan berjalan. Dalam pengambilan informasi, *cosine similarity* dari 2 dokumen akan berkisar dari 0 sampai 1, karena *term frequency* (menggunakan pembobotan tf-idf) tidak boleh negatif. Sudut antara dua vektor *term frequency* tidak boleh lebih besar dari 90.

K. Penelitian Terkait

Dalam Publikasi (Ayu 2016) Pengukur Kemiripan Dokumen Teks Bahasa Indonesia menggunakan Metode *Cosine Similarity*. Hasil akhir dari penelitian ini adalah tingkat kemiripan seluruh isi dokumen dengan dokumen-dokumen yang sudah ada. Persamaan dengan penelitian ini menggunakan metode *cosine similarity* dalam menghitung kemiripan dokumen. Perbedaan dengan penelitian tidak menggunakan metode *clustering* dan hanya mengukur kemiripan dari seluruh isi dokumen bukan tiap kalimat.



Gambar 5. Blok Diagram Sistem

Dalam Publikasi (Adinugroho 2018) Implementasi Metode *Text Mining* dan *K-Means Clustering* untuk Pengelompokan Dokumen Skripsi (Studi Kasus: Universitas Brawijaya). Penelitian ini merupakan pengelompokan dokumen skripsi menggunakan sistem dengan metode *text mining* sebagai *preprocessing*-nya dan *K-Means* menjadi metode dalam pengelompokannya. Persamaan dengan penelitian ini adalah menggunakan metode *clustering* yang sama, yaitu *K-Means* dan menggunakan algoritma *cosine similarity* dalam menemukan kesamaan antar dokumen. Perbedaannya adalah penelitian ini hanya menggunakan *K-Means* sebagai metode *clustering* dan mengelompokan berdasarkan keseluruhan dokumen.

Dalam Publikasi (Indriyanto 2017) Klasterisasi Dokumen Tugas Akhir Menggunakan *K-Means Clustering* sebagai Analisa Penerapan Sistem Temu Kembali. Penelitian ini membahas pengelompokan dokumen hasil pencarian berdasarkan kategori yang dapat menggambarkan isi dari suatu dokumen. Persamaan dengan penelitian ini, yaitu sama-sama menerapkan *clustering* dan menggunakan algoritma *K-Means*. Perbedaan dengan penelitian ini, hanya mengelompokan dokumen berdasarkan judul dari dokumen tugas akhir.

II. METODE PENELITIAN

A. Observasi dan Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah dokumen jurnal Program Studi Informatika Fakultas Teknik Universitas Sam Ratulangi yang diunggah melalui situs e-journal UNSRAT. Dokumen jurnal yang diunggah merupakan data yang digunakan sebagai data *training* maupun data *test* dalam mengukur tingkat kemiripan dokumen.

TABEL I
JUMLAH DOKUMEN JURNAL

Tahun Publikasi	Jumlah Dokumen
Tahun 2012	9 Dokumen Jurnal
Tahun 2013	11 Dokumen Jurnal
Tahun 2014	10 Dokumen Jurnal
Tahun 2015	10 Dokumen Jurnal
Tahun 2016	15 Dokumen Jurnal
Tahun 2017	15 Dokumen Jurnal
Tahun 2018	10 Dokumen Jurnal
Tahun 2019	20 Dokumen Jurnal
TOTAL	**100 Dokumen Jurnal**

Dokumen jurnal yang diunggah merupakan dokumen jurnal tahun publikasi 2012 sampai dengan tahun 2019. Banyaknya dokumen jurnal yang diunggah dapat dilihat pada tabel I. Dari 100 dokumen jurnal yang sudah diunggah, diproses dalam sistem dan hasilnya disimpan dalam basis data. Data tersebut (data *training*) akan digunakan sebagai data pembanding terhadap dokumen jurnal baru yang akan diukur tingkat kemiripannya.

B. Perancangan dan Pembuatan aplikasi

Perancangan dan pembuatan aplikasi perlu dilakukan untuk menjaga agar proses kerja aplikasi dapat berjalan dengan lancar dan teratur sehingga menghasilkan informasi yang benar. Prinsip kerja aplikasi bertujuan untuk memperlihatkan proses kerja aplikasi secara umum, bagaimana proses sistem berjalan dan mendapatkan hasil yang dibutuhkan. Prinsip kerja aplikasi seperti pada gambar 5 yang secara garis besar memperlihatkan proses kerja aplikasi bagaimana sistem mendapatkan hasil yang dibutuhkan.

Tahap pertama pada prinsip kerja aplikasi ini, yaitu mengolah data *training* dengan memasukkan dokumen jurnal ke dalam sistem dimana file jurnal yang dari format PDF dikonversi ke dalam format TXT dengan menggunakan fungsi dari *library* pdfminer. Kemudian masuk pada proses pertama, yaitu *preprocessing* dimana pada proses ini seluruh dokumen jurnal yang dimasukan sebagai data *training* hanya diambil karakter (*string*) yang dapat diolah atau berguna pada proses selanjutnya. Pada *preprocessing* ini terdapat beberapa sub prosesnya, yaitu proses *cleansing* dimana dokumen jurnal dibersihkan dari segala tanda baca atau karakter yang tidak diperlukan, kemudian mengubah semua huruf kapital menjadi huruf kecil pada proses *case folding*, dan setelah itu setiap dokumen dipecah menjadi kalimat-kalimat yang berdiri sendiri dengan menggunakan fungsi *split*, yaitu memisahkan karakter berdasarkan tanda baca titik (.) pada proses *sentence tokenizing*. Setelah kalimat didapat lalu dihilangkan imbuhan-imbuhan sehingga tersisa kata dasarnya (proses *stemming*), kata-kata umum yang sering muncul juga dihilangkan (proses *stopword removal*) dan kedua proses ini menggunakan fungsi *stemmer* dan *stopword removal* pada *library* sastrawi. Proses selanjutnya adalah pembobotan *term* (*term frequency*) dimana pada proses ini dihitung jumlah kemunculan kata pada tiap-tiap kalimat yang ada menggunakan fungsi *tokenize* pada *library* NLTK. Kemudian pada proses pengelompokan dokumen jurnal data *training* ini menggunakan nilai *term frequency* dari tiap kalimat yang ada untuk menentukan kelompok data menerapkan algoritma *DBSCAN* pada metode *clustering* dengan menjalankan fungsi *cluster DBSCAN* pada *library scikit-learn*. Algoritma *DBSCAN* digunakan pada pengelompokan data *training* karena dapat menentukan sendiri jumlah kelompoknya.

Tahap kedua adalah mengolah data baru dengan memasukan satu dokumen jurnal yang akan diukur tingkat kemiripannya berdasarkan perbandingan dengan data *training*. Proses mengolah data baru dan data *training* tidak jauh berbeda hanya saja pada proses penentuan kelompok data baru, sistem akan memprediksi dan menentukan kelompok data berdasarkan kelompok yang sudah dibentuk saat mengolah data *training*. Penentuan kelompok data baru ini menerapkan algoritma *K-means* dengan menjalankan fungsi *cluster K-means* pada *library scikit-learn*. Kemudian data baru yang sudah ditentukan kelompok diukur nilai kemiripannya dengan membandingkan data *training* yang terdapat pada kelompok data yang sama berdasarkan nilai *term frequency* tiap-tiap kalimat. Proses menghitung nilai kemiripan dokumen tersebut menerapkan algoritma *cosine similarity* dengan menjalankan fungsi *cosine similarity* pada *library scikit-learn*. Hasil akhir yang didapat dari sistem adalah berupa banyaknya kalimat yang dinyatakan mirip dan kalimat yang tidak mirip dengan standar nilai kemiripan kalimat bernilai 50%.

C. Evaluasi Sistem

Evaluasi sistem bertujuan untuk memeriksa dan menilai jika hasil dari aplikasi yang telah dirancang dan dibuat sesuai dengan tujuan utama pembuatan aplikasi. Penerapan *clustering* pada aplikasi pendeteksi tingkat kemiripan dokumen teks ini menggunakan algoritma *DBSCAN* dalam menentukan jumlah kelompok yang dibutuhkan oleh algoritma *K-Means* dalam menentukan kelompok data. Untuk menentukan jumlah kelompok dengan menggunakan algoritma *DBSCAN* dibutuhkan parameter nilai *epsilon* dan *minpts*. Tiap parameter yang dimasukan dalam algoritma *DBSCAN* ini menghasilkan jumlah kelompok yang berbeda-beda. Untuk itu dilakukan beberapa perbandingan pengujian menggunakan parameter *epsilon* 1 *minpts* 5, *epsilon* 2 *minpts* 5, *epsilon* 3 *minpts* 5, *epsilon* 4 *minpts* 5, dan *epsilon* 5 *minpts* 5. Dari tiap-tiap parameter yang ada dihitung dan disimpan nilai akurasi dan waktu komputasinya.

Dalam melakukan perhitungan untuk mencari nilai akurasi data yang telah dihasilkan oleh sistem dilakukan pengujian menggunakan *confusion matrix*. *Confusion matrix* adalah tabel yang sering digunakan untuk menggambarkan kinerja model klasifikasi pada data uji yang nilai sebenarnya diketahui. Terdapat 4 istilah sebagai representasi hasil proses klasifikasi, yaitu: *True Positive* (TP): data positif yang terdeteksi benar, *True Negative* (TN): data negatif yang terdeteksi salah, *False Positive* (FP): data negatif namun terdeteksi data positif, *False Negative* (FN): data positif namun terdeteksi data negatif. Untuk menggunakan metode ini perlu dilakukan beberapa perhitungan dalam mencari nilai akurasi, presisi, dan *recall*. Akurasi adalah persentase dari total data yang diidentifikasi dan dinilai. Berikut adalah rumus dari akurasi:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN}\ x\ 100\% \qquad (3)$$

Dalam penerapan *clustering* pada aplikasi pendeteksi kemiripan dokumen teks ini dilakukan perbandingan hasil akhir ketika menerapkan *clustering* dan tidak menerapkan *clustering*. Perbandingan hasil akhir yang menjadi bahan pengujian adalah persentase akurasi dari hasil kemiripan dokumen teks dan waktu komputasi yang diperoleh sistem saat menjalankan proses untuk mendapat nilai kemiripan dokumen teks. Lama waktu komputasi atau waktu proses kerja aplikasi diukur mulai dari proses pertama dokumen teks dimasukkan sampai aplikasi berhasil mendapat hasil akhirnya yaitu mendapat nilai kemiripan dokumen teks. Hal ini diperlukan untuk mengetahui jika menerapkan metode ini akan lebih akurat atau tidak.

III. HASIL DAN PEMBAHASAN

A. Implementasi Aplikasi

Implementasi aplikasi merupakan tahap penerapan hasil dari setiap proses yang ada ke dalam perancangan sistem menggunakan Bahasa pemrograman. Antarmuka sistem aplikasi pendeteksi tingkat kemiripan dokumen teks ini dibuat dengan menggunakan kode HTML dalam *web framework Flask* dalam mempermudah pengguna menggunakan aplikasi.

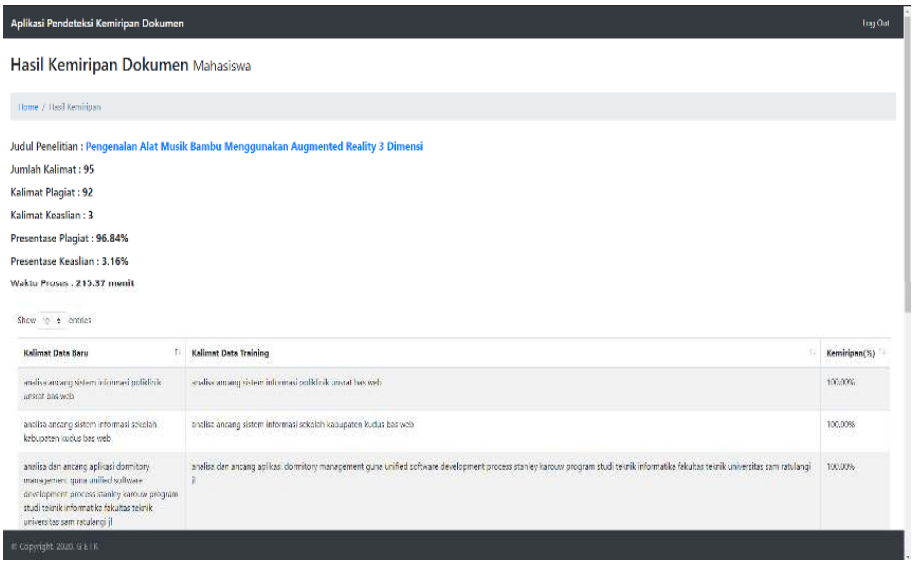
1) Gambar 6
Merupakan tampilan hasil kemiripan dokumen jurnal yang telah diunggah oleh pengguna (mahasiswa) ke dalam aplikasi. Pada halaman ini terdapat informasi hasil kemiripan secara rinci seperti judul dokumen, banyaknya kalimat yang mirip dan tidak, persentase kemiripan dan keaslian.

2) Gambar 7
Merupakan tampilan utama untuk pengguna (pengolah jurnal) yang terdapat hasil kemiripan dokumen yang telah diunggah. Pengolah jurnal mendapat fitur lebih seperti menghapus dan menerima data jurnal, melihat rincian hasil kemiripan, dan mengunggah dokumen jurnal.

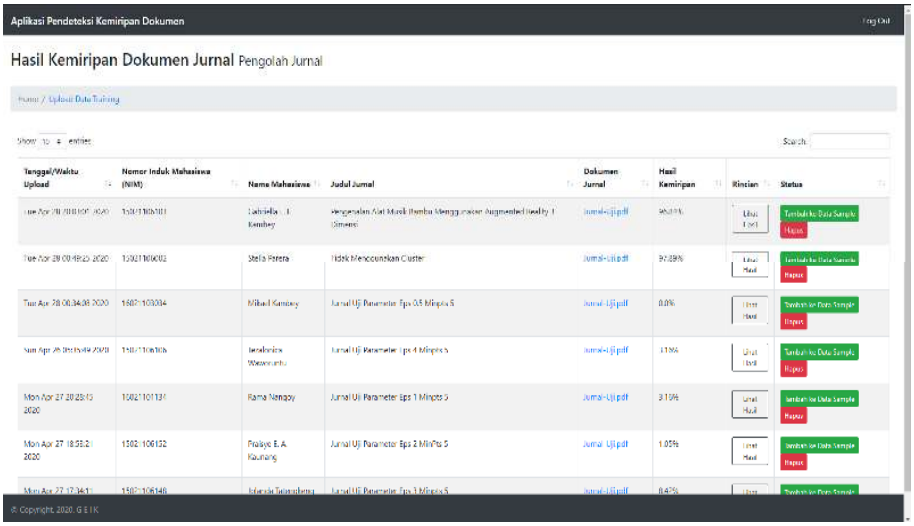
3) Gambar 8
Merupakan tampilan untuk pengguna (pengolah jurnal) melihat hasil terperinci dari hasil kemiripan dokumen jurnal yang sudah ada pada halaman utama. Halaman ini sama dengan halaman hasil kemiripan dokumen jurnal mahasiswa. Informasi yang ditampilkan juga sama hanya saja tidak ditampilkan judul penelitian pada halaman ini.

B. Hasil Pengujian

Pengujian dilakukan pada data jurnal dengan menerapkan metode *clustering* dari beberapa parameter yang digunakan sebagai perbandingan. Pengujian juga dilakukan tanpa menerapkan metode *clustering* dengan tidak melakukan proses pengelompokan pada sistem. Sistem hanya mengukur nilai kemiripan berdasarkan keseluruhan dokumen jurnal bukan berdasarkan tiap-tiap kelompoknya. Diperhatikan nilai akurasi dari hasil akhir aplikasi saat menerapkan *clustering* dan tidak menerapkan *clustering* juga waktu komputasinya.

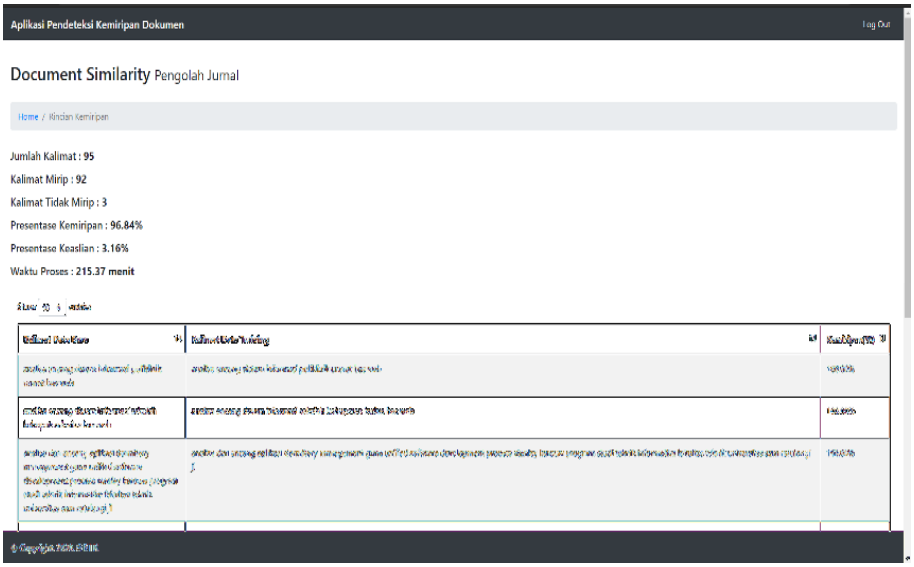


Gambar 6. Halaman Hasil Kemiripan Dokumen

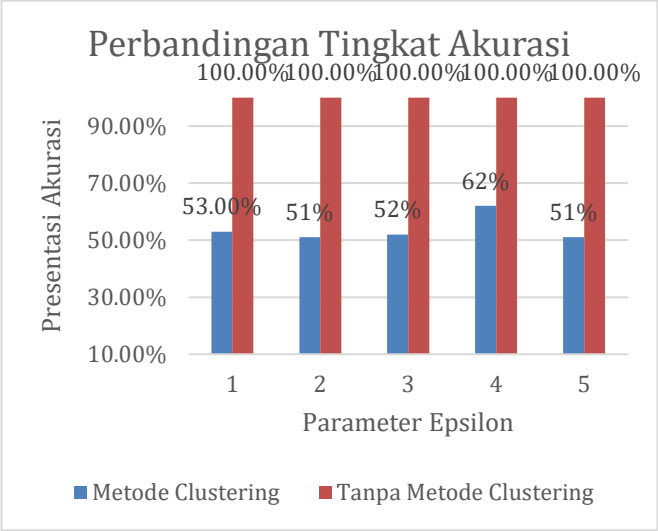


Gambar 7. Halaman Utama Pengolah Jurnal

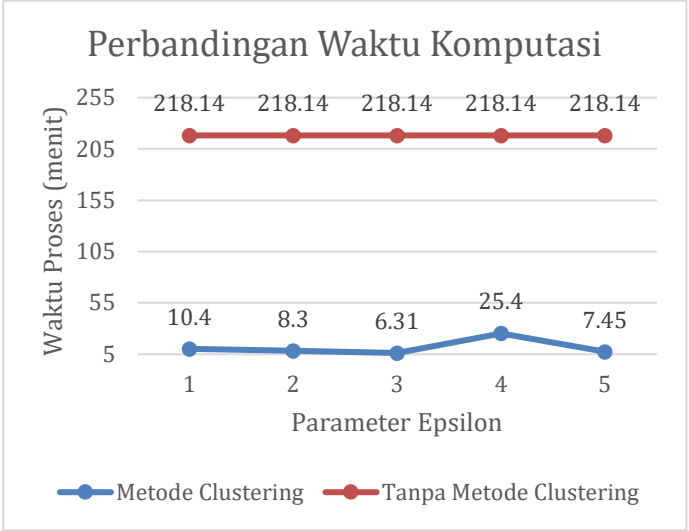
p-ISSN : 2301-8402, e-ISSN : 2685-368X ,dapat diakses melalui https://ejournal.unsrat.ac.id/index.php/informatika



Gambar 8. Halaman *Detail* Hasil Kemiripan



Gambar 9. Grafik Perbandingan Akurasi Tanpa Penerapan *Clustering* & Penerapan *Clustering*



Gambar 10. Grafik Perbandingan Waktu Komputasi Tanpa Penerapan *Clustering* & Penerapan *Clustering*

TABEL II
HASIL WAKTU KOMPUTASI

Parameter	Pembersihan data	Pengelompokan	Mengukur Kemiripan	Jumlah Dokumen
Epsilon 1 *Minpts* 5	3,52 menit	6 detik	6,6 menit	**10,4 menit**
Epsilon 2 *Minpts* 5	3,34 menit	7 detik	4,49 menit	**8,30 menit**
Epsilon 3 *Minpts* 5	3,18 menit	5 detik	3,8 menit	**6,31menit**
Epsilon 4 *Minpts* 5	3,1 menit	5 detik	21,58 menit	**25,4 menit**
Epsilon 5 *Minpts* 5	2,44 menit	5 detik	4,56 menit	**7,45 menit**
Tanpa Metode *Clustering*	52 detik	-	217,22 menit	**218,14 menit**

1) Perbandingan Akurasi Tanpa Penerapan *Clustering* & Penerapan *Clustering*

Pada gambar 9 dapat dilihat perbandingan tingkat persentase akurasi dari hasil performansi sistem saat menerapkan *clustering* dengan tidak menerapkan *clustering*. Nilai akurasi saat menerapkan metode *clustering* berbeda jauh dengan tidak menerapkan *clustering* dengan nilai tertinggi hanya mencapai 62%. Hal ini terjadi karena data yang dibandingkan saat menerapkan *clustering* hanya berdasarkan kelompok data yang sama. Sedangkan saat tidak menerapkan *clustering* data yang dibandingkan secara keseluruhan sehingga mendapat nilai persentase akurasi tinggi, yaitu mencapai 100%.

2) Perbandingan Waktu Komputasi Tanpa Penerapan *Clustering* & Penerapan *Clustering*

Saat sistem mengukur dan menghasilkan tingkat kemiripan dokumen teks membutuhkan waktu komputasi yang berbeda saat menerapkan *clustering* maupun tanpa menerapkannya.

Seperti yang dapat dilihat pada tabel II, saat menerapkan *clustering* waktu yang dibutuhkan oleh sistem untuk tiap parameter juga berbeda. Perbedaan waktu komputasi saat menerapkan *clustering* dan tanpa menerapkan *clustering* dapat dilihat grafiknya pada gambar 10.

Dapat dilihat grafik waktu komputasi yang sangat jauh perbandingannya dimana saat menerapkan *clustering* hanya memakan waktu sekitar 5-25 menit sedangkan saat tidak menerapkan *clustering* waktu komputasi terdapat pada titik 218 menit. Hal ini dapat membuktikan bahwa saat menerapkan metode *clustering* lebih mempercepat sistem dalam menghasilkan tingkat kemiripan dokumen teks. Waktu proses yang cepat dapat terjadi karena pada saat menerapkan *clustering* sistem hanya membandingkan data yang terdapat pada kelompok yang sama. Itulah sebabnya waktu komputasi lebih cepat saat menerapkan metode *clustering*.

IV. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa aplikasi pendeteksi tingkat kemiripan mendapat hasil yang tidak tepat atau kurang akurasi ketika menerapkan metode *clustering* dibandingkan saat tidak menerapkan metode tersebut. Tetapi waktu yang dibutuhkan aplikasi untuk proses mendapat hasil kemiripan lebih sedikit atau lebih cepat waktu komputasinya ketika menerapkan metode *clustering* dibandingkan saat tidak menerapkan metode tersebut.

B. *Saran*

Saran untuk pengembangan penelitian lebih lanjut dari penelitian ini dalam pengembangan aplikasi setelah pengguna mengunggah file perlu ditambah fitur yang dapat memberi pemberitahuan kepada pengguna ketika hasil kemiripan dokumen teks yang telah didapat. Pengembangan penelitian ini juga bisa menggunakan metode atau algoritma *clustering* yang lain untuk mendapatkan hasil kemiripan yang lebih akurat.

V. KUTIPAN

[1] V. Stepchyshyn and R. S. Nelson, *Library plagiarism policies*. Chicago : College Library Information Packet Committee, College Libraries Section, Association of College and Research Libraries, 2007, 2007.

[2] H. Shadily, "Ensiklopedi Indonesia." Ichtiar Baru van Hoeve, 1980.

[3] "plagiat," *artikata.com*, 2014. [Online]. Available: http://artikata.com/arti-345419-plagiat.html. [Accessed: 23-Jun-2014].

[4] D. Delen and M. D. Crossland, "Seeding the survey and analysis of research literature with text mining," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1707–1720, 2008.

[5] D. Turban, E. Sharda, R. Dele, *Decision Support and Business Intelligence Systems*. .

[6] HENDRO NINDITO, "TEORI TEXT MINING DAN WEB MINING," *15 December 2016*, 2016. [Online]. Available: https://sis.binus.ac.id/2016/12/15/teori-text-mining-dan-web-mining/.

[7] E. Rahmawati, L, Sihwi, SW, Suryani, "ANALISA CLUSTERING MENGGUNAKAN METODE K-MEANS DAN HIERARCHICAL CLUSTERING (STUDI KASUS : DOKUMEN SKRIPSI JURUSAN KIMIA , FMIPA , 2 . 3 Term Weighting dengan Term Frequency."

[8] INFORMATIKALOGi, "Text Preprocessing," *NOVEMBER 27, 2016 · UPDATED JULY 11, 2017*, 2016. [Online]. Available: https://informatikalogi.com/text-preprocessing/.

[9] M. A. Hall and L. A. Smith, "Feature Subset Selection : A Correlation Based Filter Approach," pp. 1–4.

[10] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, Jun. 1998.

[11] I MADE SUWIJA PUTRA, "ALGORITMA DBSCAN (DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE) DAN CONTOH PERHITUNGANNYA," UNIVERSITAS UDAYANA DENPASAR, 2018.

[12] D. A. W. H. J. Y. S. Purwanto, "RANCANG BANGUN APLIKASI CLUSTERING DATA MINING," vol. vol 7. no, p. 7, 2018.

[13] G. A. Pradnyana, "Perancangan Dan Implementasi Automated Document Integration Dengan Menggunakan Algoritma Complete Linkage Agglomerative Hierarchical Clustering," *J. Ilmu Komput.*, Vol. 5, Jan. 2012.

Gabriella Eyrene Iriani Kambey, lahir di Jayapura 31 Oktober 1997. Penulis merupakan anak ke-1 dari 3 bersaudara. Penulis mulai menempuh Pendidikan di Sekolah Dasar Swasta Antonius Medan (2003-2009). Penulis lalu melanjutkan ke Sekolah Menengah Pertama Swasta Putri Cahaya Medan (2009-2010) lalu pindah di Sekolah Menengah Pertama Katolik Santa Theresia Manado (2010-2012). Kemudian penulis melanjutkan Sekolah Menengah Atas Negeri 9 Manado (2012-2015). Pada tahun 2015 penulis melanjurkan studi ke Perguruan Tinggi Negeri di Universitas Sam Ratulangi Manado dengan mengambil Program Studi S-1 Teknik Informatika di Jurusan Teknik Elektro Fakultas Teknik. Pada bulan November tahun 2018 Penulis mengajukan proposal Skripsi untuk memenuhi syarat meraih gelar sarjana (S1) dengan judul Penerapan *Clustering* pada Aplikasi Pendeteksi Kemiripan Dokumen Teks Bahasa Indonesia yang kemudian disetujui dan melanjutkan pembuatan penelitian skripsi. Pembuatan skripsi ini dibimbing oleh dua dosen pembimbing, yaitu Rizal Sengkey, ST, MT dan Agustinus Jacobus, ST, M.Cs. Pada 22 Mei 2020, penulis resmi menyelesaikan skripsi dengan menyandang gelar sarjana komputer dengan predikat sangat memuaskan.