

## Aplikasi Pengecekan Dokumen Digital Tugas Mahasiswa Berbasis Website

Latius Hermawan<sup>1</sup>, Maria Bellaniar Ismiati<sup>2</sup>

Program Studi Informatika dan Sistem Informasi., Fakultas Sains dan Teknologi,  
Universitas Katolik Musi Charitas

Jl. Bangau No. 60 Palembang, 30113, Sumatera Selatan, Indonesia

Email: [1tiuz.hermawan@ukmc.ac.id](mailto:1tiuz.hermawan@ukmc.ac.id), [2bella@ukmc.ac.id](mailto:2bella@ukmc.ac.id)

**Abstract.** *Website-Based Application for Checking Students' Digital Assignment.* Nowadays, technology is not only about computers as it has advanced to smartphones and other things. In UKMC, technology has certainly helped the job. However, in this university, there is no application for checking the plagiarism of the students' digital assignments, whereas plagiarism is sometimes done by students when working on assignments from online sources. Students' assignments can be easily done by doing copy and paste without mentioning its reference because students tend to think practically when working on assignments. Plagiarism is strictly prohibited in education because it is not permitted. Therefore, a plagiarism detection application should be created. It applies a string-matching algorithm in text documents to search the common words between documents. By applying the string-matching method in document that match with other documents, an output that will provide information on how similar the text documents are can be generated. After testing, it is obtained that this application can help lecturers and students to reduce the level of plagiarism.

**Keywords:** *Application, Plagiarism, Digital, Assignment*

**Abstrak.** *Sekarang teknologi tidak hanya tentang computer karena kemajuannya telah merambah pada smartphone, dan hal-hal lainnya. Di UKMC, teknologi yang digunakan sudah sangat membantu pekerjaan. Namun di universitas ini, belum ada aplikasi yang dapat memeriksa plagiarisme dari tugas digital mahasiswa padahal plagiarisme terkadang dilakukan oleh mahasiswa saat mengerjakan tugas dari sumber online. Tugas mahasiswa dapat dengan mudah dibuat dengan cara copy-paste tanpa menyebutkan referensi, karena siswa cenderung berpikir praktis ketika mengerjakan tugas. Plagiarisme sangat dilarang dalam pendidikan karena tidak diizinkan. Oleh karena itu aplikasi pendeteksi plagiarisme perlu dibuat. Aplikasi ini menerapkan algoritma pencocokan string dalam dokumen teks untuk mencari kata-kata umum antar dokumen. Dengan metode pencocokan string pada dokumen yang cocok dengan beberapa dokumen lainnya dapat dihasilkan suatu keluaran yang akan memberikan informasi seberapa dekat antar dokumen teks tersebut. Setelah dilakukan pengujian, didapat hasil bahwa aplikasi ini dapat membantu dosen dan mahasiswa untuk mengurangi tingkat plagiarisme.*

**Kata Kunci:** *aplikasi, plagiarisme, tugas kuliah.*

### 1. Pendahuluan

#### 1.1. Latar Belakang

Seiring perkembangan teknologi dan informasi yang pesat dan setiap orang dengan mudah menerima dan mencari informasi yang dibutuhkan. Perkembangan teknologi dan informasi membawa dampak yang besar di dalam kehidupan manusia salah satunya dibidang pendidikan [1], dimana memungkinkan mahasiswa dan dosen dapat mencari segala informasi yang diperlukan, mulai dari buku, jurnal, tutorial dan ilmu teknis yang dapat digunakan. Dalam dunia pendidikan, informasi dari *e-book*, artikel ataupun jurnal baik yang berskala nasional hingga internasional dengan reputasi yang bervariasi dapat diakses melalui internet. Sayangnya dengan kemudahan yang diberikan memicu tindakan-tindakan yang menyimpang contohnya plagiarisme. Plagiarisme juga dapat diartikan berupa tulisan atau potongan tulisan maupun secara keseluruhan tidak dikutip dengan baik dan benar, baik tindakan tersebut sengaja ataupun tidak disengaja [2].

Plagiarisme merupakan tindakan pengambilan alih informasi, data ataupun ilmu yang telah ada dan diakui sebagai karya pribadi, sehingga penemu yang asli tidak dikutip dalam informasi ataupun data yang dipakai [3]. Aplikasi plagiarisme dibutuhkan untuk mendukung perkuliahan mahasiswa karena

plagiarisme sendiri tidak diperkenankan dalam dunia akademik dan merupakan bentuk serius dari kesalahan ilmiah. Plagiarisme di dunia akademik dapat diatasi dengan pemanfaatan teknologi seperti menggunakan *tools* pendeteksi plagiarisme secara *online*. Sedangkan dalam bidang akademik, yang sering terjadi plagiarisme adalah tugas mahasiswa yang dikumpulkan. Plagiarisme yang dilakukan oleh mahasiswa saat mengerjakan tugas sering terjadi karena banyaknya informasi atau artikel yang tersedia secara *online*. Oleh karena itu, tugas mahasiswa dapat dengan mudah dibuat dengan melakukan *copy-paste* tanpa menyebutkan referensi karena mahasiswa cenderung berpikir praktis saat mengerjakan tugas-tugas dengan cara *copy paste* [4][13]. Selain itu, karena mahasiswa terbiasa membuat dan mengumpulkan tugas secara sembarang serta yang penting dapat kumpul tugas tepat waktu [5].

Berdasarkan wawancara dengan beberapa mahasiswa tentang plagiarisme tugas pada program studi Informatika, alasan mereka adalah sebagai berikut : terlalu banyak tugas yang harus dikumpulkan sehingga sering meminjam tugas temannya untuk mempersingkat waktu pengerjaan, begitu mudahnya pencarian data atau materi di internet sehingga mahasiswa semakin senang untuk melakukan copy paste tanpa melihat sumbernya terlebih dahulu, seringnya mahasiswa membuat tugas saat mendekati batas waktu pengumpulan sehingga harus mencari cara yang praktis dan cepat. Halhal tersebutlah yang menjadi alasan peneliti untuk membuat suatu aplikasi untuk mendeteksi plagiarisme tugas-tugas mahasiswa yang bentuknya *soft file*.

Aplikasi pendeteksi plagiarisme yang dibuat menerapkan algoritma pencocokan string dalam dokumen teks untuk mencari kata-kata umum di antara dokumen, dalam hal ini akan digunakan algoritma *TF-IDF* dan *Cosine Similarity Index*. Dengan adanya pencocokan string dalam dokumen yang dicocokkan dengan beberapa dokumen lain, dapat dihasilkan sebuah output yang akan memberikan informasi seberapa dekat dokumen teks yang dikumpulkan mahasiswa sebagai tugas kuliah dengan dokumen tugas kuliah yang lain. Berdasarkan latar belakang dan masalah yang sudah dijelaskan sebelumnya, peneliti mengambil judul “Simple Plagiarism Checker Application Untuk Memeriksa Dokumen Digital Tugas Mahasiswa”. Dengan adanya aplikasi ini diharapkan dapat membantu dosen dan mahasiswa untuk mengurangi tingkat plagiarisme yang sering terjadipada mahasiswa.

## 2. Tinjauan Pustaka

### 2.1. Aplikasi

Perangkat lunak aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah Microsoft Office dan Open Office.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna [6].

### 2.2. Plagiarisme

Menurut Kamus Besar Bahasa Indonesia (KBBI), plagiarisme adalah tindakan penjiplakan atau pengambilan karangan, pendapat, dan sebagainya dari orang lain dan menjadikannya seolah karangan dan pendapat sendiri [7]. Adapun beberapa contoh yang dianggap sebagai tindakan plagiarism sebagai berikut:

- (a) *Copy paste* (copas) pada artikel/tulisan/posting orang lain tanpa mencantumkan nama pemiliknya.
- (b) Mengganti nama pemilik karya tulis dengan nama sendiri.
- (c) Menyalin bersih tulisan orang lain kedalam karya tulis yang dibuat tanpa ada perbedaan kata.
- (d) Menggunakan ide orang lain baik berupa tulisan, gambar, video dan lainnya tanpa mencantumkan sumbernya.
- (e) Membeli karya tulis orang lain lalu menyebarkan atas nama pribadi.
- (f) Menulis hasil karya orang lain dengan

mengganti dengan kalimat sendiri tanpa mencantumkan sumber penelitian tersebut. (g) Mengubah hasil penelitian orang lain tanpa seizin dari pemilik karya ilmiah.

Suatu dokumen dapat dikatakan plagiat apabila hal-hal di bawah ini dilakukan oleh orang yang membuat karya ilmiah [8]: (a) Menyalin sebagian besar (>50%) dari satu penulis atau pengarang yang sama. (b) Menyalin sebagian tulisan (antara 20-50%) dari pengarang yang sama. (c) Menyalin kata perkata (hingga 20%). (d) Menyalin sebagian besar tanpa memberikan quote yang benar.

### 2.3. Dokumen Digital

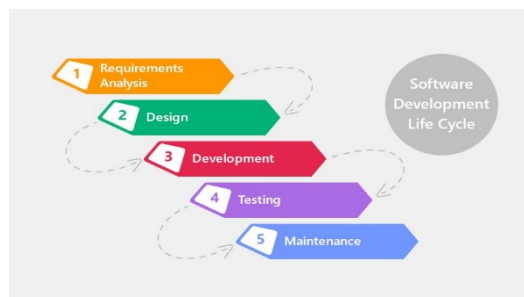
Dokumen adalah sebuah tulisan yang memuat data dan informasi. Biasanya, dokumen ditulis di kertas dan informasinya ditulis memakai tinta baik memakai tangan atau memakai media elektronik (seperti printer). Dokumen merupakan suatu sarana transformasi informasi dari satu orang ke orang lain atau dari suatu kelompok ke kelompok lain [9]. Dokumen meliputi berbagai kegiatan yang diawali dengan bagaimana suatu dokumen dibuat, dikendalikan, diproduksi, disimpan, didistribusikan, dan digandakan. Dokumen sangat penting, baik dalam kehidupan sehari-hari, organisasi, maupun bisnis. Dokumen digital merupakan setiap informasi elektronik yang dibuat, diteruskan, dikirimkan, diterima, atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal, atau sejenisnya, yang dapat dilihat, ditampilkan dan didengar melalui komputer atau sistem elektronik tetapi tidak terbatas pada tulisan, suara atau gambar, peta, rancangan, foto atau sejenisnya, huruf, tanda, angka, kode akses, simbol yang memiliki makna atau arti atau dapat dipahami oleh orang yang mampu memahaminya. Dokumen digital dapat dihasilkan dengan menggunakan aplikasi pengolah kata (*word processor*) seperti Microsoft Word, Notepad atau OpenOffice untuk menghasilkan sebuah berkas komputer dengan ekstension yang berbeda-beda sesuai dengan aplikasi pengolah kata yang digunakan.

### 2.4. Website

Pengertian *website* adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada website disebut dengan *web page* dan link dalam website memungkinkan pengguna bisa berpindah dari satu page ke page lain (*hyper text*), baik diantara page yang disimpan dalam server yang sama maupun server diseluruh dunia [10].

### 2.5. Waterfall Application Development

Model Sekuensial Linier atau sering disebut Model Pengembangan Air Terjun (*Waterfall*), merupakan paradigma model pengembangan perangkat lunak paling tua, dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian, dan pemeliharaan. Berikut Merupakan Tahapan – tahapan Pengembangan Model Sekuensial Linear / *Waterfall Development Model* seperti pada Gambar 1 berikut [11]:



Gambar 1. Model Proses Pengembangan Sistem *Waterfall*

## 2.6. PHP dan MySQL

PHP (HyperText Preprocessor) adalah sebuah bahasa utama script server-side yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop [12]. PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman script – script yang membuat dokumen HTML secara on the fly yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman server side.

## 2.7. Algoritma Cosine Similiarity dan TF-IDF

### 2.7.1. TF-IDF

*Term Frekuensi-Inverse Document Frekuensi* adalah salah satu perhitungan bobot dari frekuensi kemunculan sebuah term pada dokumen [13]. TF-IDF digunakan untuk mencari nilai bobot dari dokumen. Algoritma TF-IDF akan memeriksa kemunculan tiap kata pada isi dokumen dari hasil tokenisasi, filtering dari kemunculan tiap kata pada isi dokumen. Adapun rumus untuk perhitungan TF-IDF yaitu pada Persamaan 1 :

$$(1) \quad \begin{aligned} W_{ij} &= tf \times idf \\ W_{ij} &= tf_{ij} \times \left( \log \frac{N}{n} + 1 \right) \end{aligned}$$

Dimana :

- $W_{ij}$  : bobot kata *term* ke-j dan dokumen ke-i.
- $tf_{ij}$  : jumlah kemunculan kata / *term* ke-j dalam dokumen ke-i.
- $idf$  :  $\log N/n + 1$
- $N$  : jumlah semua dokumen yang ada.
- $n$  : jumlah dokumen yang mengandung term ke-j.

### 2.7.2. Cosine Similarity

Cosine Similarity digunakan untuk melakukan perhitungan kesamaan dari dokumen. Rumus yang digunakan oleh cosine similarity adalah pada Persamaan 2:

$$(2) \quad \textit{Similarity Value (SV)} = \cos \theta \frac{AB}{|A| |B|}$$

Dimana:

- A : vektor A
- B : vektor B
- $|A|$  : panjang vektor A,  $|B|$  : panjang vektor B

## 2.8. Penelitian Terdahulu

Terdapat beberapa penelitian yang dilakukan sebelumnya. Misalnya penelitian [14] menjelaskan pencarian semantik dengan algoritma weighted tree similarity dapat diterapkan pada pencarian dokumen dan pencarian halaman web. Pada penelitian [15] membahas masalah pendeteksian kemiripan dokumen teks menggunakan algoritma lavenshtein distance dengan mengambil data dari berita. Algoritma lavenshtein distance juga mampu digunakan untuk pencarian E-Boarding House untuk mempermudah mahasiswa mendapatkan tempat kos [16]. Salah satu penelitian yang telah dilakukan oleh [17] untuk mendeteksi plagiat dari penelitian ini hanya membahas masalah sistem pendeteksian kemiripan judul skripsi menggunakan algoritma winnowing, dan hasilnya sistem ini mampu melakukan proses penentuan persentase pendeteksian kemiripan judul skripsi menjadi lebih cepat dan akurat.

### 3. Metodologi Penelitian

#### 3.1. Model Pengembangan perangkat Lunak

Kegiatan penelitian ini dilakukan dalam waktu satu semester. Secara keseluruhan, kegiatan penelitian terbagi dalam lima tahap penelitian. Pada tahap pertama, dilakukan identifikasi kebutuhan secara lebih mendalam. Hasil yang diharapkan dari tahapan ini adalah daftar identifikasi, baik kebutuhan fungsional maupun kebutuhan nonfungsional. Berdasarkan identifikasi kebutuhan yang telah dibuat, dilakukan analisis perangkat lunak yang dibangun pada tahap kedua. Hasil analisis berupa pemodelan perangkat lunak yang siap dirancang. Pada tahap ketiga, model perangkat lunak yang telah dibuat pada tahap kedua, diterjemahkan dalam bentuk desain perangkat lunak yang siap dikodekan dengan bahasa pemrograman *HTML*, *CSS*, *PHP*, *MySQL*, dan *Javascript*. Selanjutnya, pada tahap keempat, pengkodean perangkat lunak dilakukan. Hasil dari kegiatan ini adalah sebuah perangkat lunak berupa website. Pada tahap kelima, dilakukan pengujian terhadap website yang sudah dibuat apakah sudah sesuai untuk menyelesaikan masalah yang dihadapi.

#### 3.2. Alat Penelitian

Alat penelitian yang digunakan dalam pembuatan penelitian ini adalah Laptop dan koneksi internet dengan kecepatan 10Mbps agar proses percobaan, *upload data and download data* dapat berjalan baik pada website. Semua alat penelitian ini berupa *hardware* yang saling mendukung satu sama lain.

#### 3.3 Alat Analisis

Alat analisis yang digunakan dalam penelitian ini adalah menggunakan aplikasi deteksi plagiarisme sendiri karena peneliti harus menyiapkan beberapa dataset untuk tugas kuliah mahasiswa untuk dicek plagiarismenya. Dengan adanya beberapa dataset tugas kuliah mahasiswa, maka dapat dilihat apakah ada penurunan dari persentase plagiarisme yang didapatkan oleh masing-masing tugas mahasiswa. Analisis dilakukan untuk mengetahui apakah dengan adanya aplikasi deteksi plagiarisme ini dapat membantu untuk mengurangi tingkat plagiarisme yang dilakukan oleh mahasiswa saat mengerjakan tugas. Selain itu, analisis dilakukan juga dengan menjalankan tombol untuk cek plagiarisme. Di tombol ini sudah ditanamkan algoritma yang digunakan untuk membandingkan antar kata yang ada di tugas kuliah tersebut.

#### 3.4. Kebutuhan Data

Data yang digunakan dalam penelitian ini adalah tugas-tugas mahasiswa yang pernah dikumpulkan dalam bentuk *soft file* dengan jumlah dataset 79 file dan data uji sebanyak 86 data. Sehingga terdapat 165 *file* tugas mahasiswa yang digunakan. Tugas-tugas ini harus mempunyai soal yang sama dan mata kuliah yang sama.

#### 3.5. Tools yang digunakan

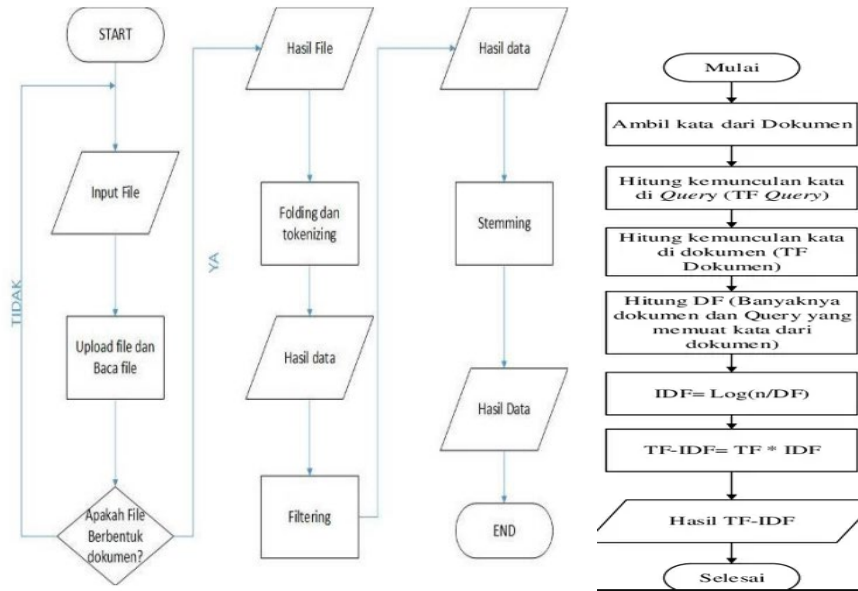
*Tools* yang digunakan dalam mendukung penelitian ini adalah *Microsoft Word*, bahasa pemrograman (*PHP* dan *Javascript*), *HTML*, *CSS*, *MySQL*, *sublime*, dan *xampp*.

### 4. Hasil dan Diskusi

#### 4.1. Proses pencocokan kata

Proses pencocokan kata yang dilakukan oleh sistem (sesuai pada Gambar 2) dijelaskan dengan lebih detail sebagai berikut. Saat dilakukan proses *upload* tugas kuliah, maka sistem akan mendeteksi semua kata yang ada. Proses *text preprocessing* akan langsung dilakukan ketika dosen telah berhasil meng-*upload* tugas kuliah yang akan dicek tingkat plagiarismenya. Proses *Case Folding-Tokenizing-Filtering-Stemming* akan membuat kata penting yang terdeteksi dapat langsung di-*index* menggunakan algoritma TF-IDF. Selanjutnya sistem akan mengambil data *query* dan akan melakukan perulangan sebanyak data *query*. Jika data *query* ada, maka prosesnya adalah menghilangkan kata yang sama, lalu sistem akan menampilkan data *query*. Jika data *query* tidak ada, maka proses selesai. Proses *real*-nya adalah mengambil data tugas kuliah lalu lakukan

perulangan sebanyak data tugas kuliah. Jika data tugas kuliah ada, maka akan dilakukan proses untuk menghilangkan kata yang sama, lalu tampilkan data tugas kuliah. Jika data tugas kuliah tidak ada, maka proses selesai. Pesan variabel adalah lakukan perulangan jika jumlah data lebih kecil sama dengan jumlah data tugas kuliah, maka gabungkan nilai *array* pada tiap data tugas kuliah, lalu tampilkan data array, hilangkan nilai *array* yang sama, hitung nilai *array*.



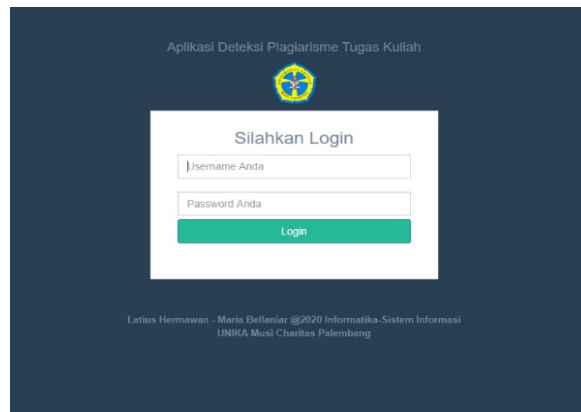
Gambar 2. Flowchart dari Aplikasi dan Algoritma

Ambil jumlah dokumen tugas kuliah, jika jumlah nilai *array* lebih besar dari nol, maka hitung nilai D/DF dan IDF, lakukan perulangan sebanyak nilai *array* yang telah di filter. Jika jumlah data lebih besar dari nol, tampung nilai ke dalam variabel, hitung nilai *tf\*idf*, lakukan perulangan sebanyak nilai *array* pada setiap dokumen tugas kuliah. Pesan variabel, lakukan perulangan jika nilai *array* pada setiap dokumen tugas kuliah lebih besar dari nol, maka hitung jarak tiap masing-masing dokumen, lakukan perulangan sebanyak nilai *array* pada setiap dokumen tugas kuliah, tampilkan hasil dan tampung nilai ke variabel. Jika jumlah *query* lebih besar dari nol, hitung nilai panjang vector dengan *query*, lakukan perulangan sebanyak jumlah *query*, lalu tampilkan hasil. Lakukan perulangan jika jumlah data lebih besar dari satu, maka hitung nilai *Q\*Di* sebanyak jumlah data tugas kuliah, lakukan *string matching* untuk mengukur kemiripan dokumen, lalu simpan data ke *database*, proses selesai.

## 4.2. Antar muka Aplikasi

### 4.2.1. Tampilan Login Simple Plagiarism Checker Application

Halaman ini fungsinya adalah untuk Login di Aplikasi Deteksi Plagiarisme Tugas Kuliah. Hak akses untuk aplikasi ini diberikan kepada dosen selaku pengampu mata kuliah yang ingin mengetahui apakah para mahasiswa yang membuat tugas kuliah saling *copy-paste* satu sama lain atau tidak. Tampilan Login dapat dilihat pada Gambar 3 di bawah ini.



Gambar 3. Tampilan Login Aplikasi Deteksi Plagiarisme Tugas Kuliah

#### 4.2.2. Halaman Upload Tugas Mahasiswa

Pada halaman ini, dosen dapat mengunggah *file* tugas-tugas mahasiswa dengan memberikan keterangan nama mahasiswa kemudian memilih *file* dan pilih pula *cover* untuk *file* tugas mata kuliah tersebut. Dosen dapat mengunggah banyak *file* tugas di halaman ini seperti yang ditampilkan pada Gambar 4.

Gambar 4. Tampilan Halaman Upload Tugas Mahasiswa

#### 4.2.3. Tampilan Halaman Hasil Plagiarisme (Terdeteksi)

Gambar 5 menampilkan hasil persentase yang terdeteksi plagiarisme dari tugas-tugas kuliah mahasiswa. Sistem ini menampilkan hasil terdeteksi plagiat apabila persentasenya lebih dari 50%. Persentase ini dapat diubah sesuai dengan kebijakan dosen pengampu masing-masing.

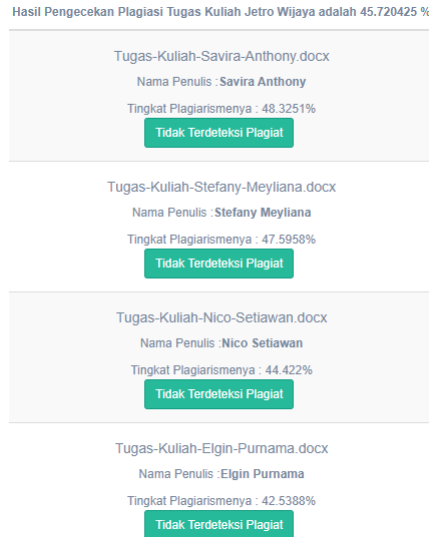
Hasil Pengecekan Plagiasi Tugas Kuliah Jetro Wijaya adalah 55.7618 %

Tugas-Kuliah-Stefany-Meyliana.docx Nama Penulis : Stefany Meyliana Tingkat Plagiarismenya : 57.8413% Terdeteksi Plagiat
Tugas-Kuliah-Savira-Anthony.docx Nama Penulis : Savira Anthony Tingkat Plagiarismenya : 57.7796% Terdeteksi Plagiat
Tugas-Kuliah-Nico-Setiawan.docx Nama Penulis : Nico Setiawan Tingkat Plagiarismenya : 54.6423% Terdeteksi Plagiat
Tugas-Kuliah-Elgin-Purnama.docx Nama Penulis : Elgin Purnama Tingkat Plagiarismenya : 52.784% Terdeteksi Plagiat

Gambar 5. Tampilan Halaman Hasil Plagiarisme (Terdeteksi)

#### 4.2.4. Tampilan Halaman Hasil Plagiarisme (Tidak terdeteksi)

Gambar 6 menampilkan halaman *report* tugas-tugas mata kuliah yang tidak terdeteksi plagiat antara satu dengan yang lainnya. Hasil persentase berada di bawah 50% sehingga tugas-tugas tersebut tidak terdeteksi plagiat.



**Gambar 6. Tampilan Halaman Hasil Plagiarisme (Tidak Terdeteksi)**

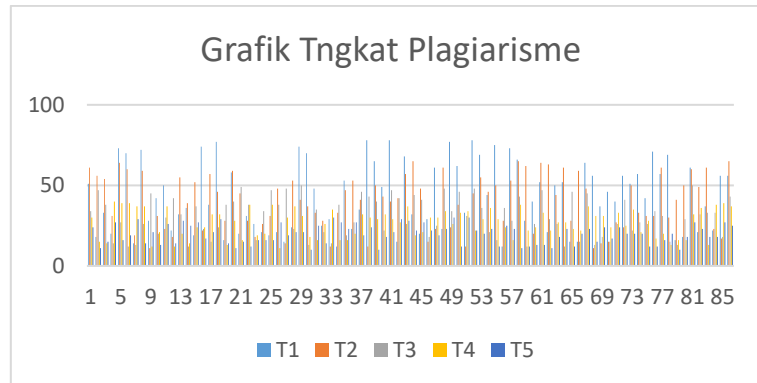
#### 4.3. Hasil Percobaan Dokumen dan Mahasiswa

Percobaan dilakukan untuk mengukur seberapa efektifnya penggunaan aplikasi ini untuk menurunkan tingkat plagiarisme yang dilakukan mahasiswa pada tugas digital yang mereka kumpulkan selama satu semester. Dokumen setiap mahasiswa akan dibandingkan dengan seluruh dokumen yang ada untuk pengecekan kemiripannya. Beberapa data yang diuji dapat dilihat pada Tabel 1 dengan keterangan DT = Dokumen Tugas. Total data mahasiswa berjumlah 86 data. Grafik pengecekan dokumen mahasiswa selama satu semester (tugas 1-tugas 5) ditampilkan pada Gambar 7.

**Tabel 1. Hasil Pengecekan Dokumen**

Mahasiswa	DT <sub>1</sub>	DT <sub>2</sub>	DT <sub>3</sub>	DT <sub>4</sub>	Hasil
JW	55%	45%	42%	35%	<b>Menurun</b>
SA	47%	45%	37%	33%	<b>Menurun</b>
NS	48%	50%	49%	51%	<b>Meningkat</b>
SM	50%	47%	43%	39%	<b>Menurun</b>
EP	55%	55%	59%	65%	<b>Meningkat</b>
FN	60%	45%	45%	39%	<b>Menurun</b>
CA	48%	43%	41%	31%	<b>Menurun</b>
GB	55%	41%	39%	32%	<b>Menurun</b>
MM	40%	42%	45%	48%	<b>Meningkat</b>
...	...	...	...	...	...
JR	49%	38%	29%	22%	<b>Menurun</b>





**Gambar 7. Grafik Pengecekan Dokumen Mahasiswa selama Satu Semester (Tugas 1-Tugas 5)**

## 5. Kesimpulan dan Saran

Kesimpulan yang didapatkan dari penelitian ini adalah aplikasi ini dapat berfungsi sebagaimana mestinya yaitu dapat menampilkan persentase plagiarisme beserta statusnya apakah terdeteksi plagiat atau tidak. Aplikasi ini dapat menampilkan hal tersebut karena algoritma TF-IDF dan Cosine Similarity yang dimasukkan ke aplikasi ini berjalan dengan baik dan runut. Dari hasil pengujian dokumen tugas, didapatkan kesimpulan bahwa para mahasiswa belajar untuk menekan persentase plagiarisme menjadi lebih kecil. Dalam beberapa tugas walaupun pernah terdeteksi plagiat, mahasiswa mampu belajar bahwa dengan adanya aplikasi ini, mahasiswa belajar bertanggungjawab dalam mengerjakan tugas kuliah.

Saran untuk penelitian selanjutnya adalah fungsionalitas aplikasi ini dapat lebih dikembangkan lagi seperti dapat mengecek plagiarisme secara otomatis untuk semua tugas mahasiswa, tidak lagi satu per satu.

## 6. Ucapan Terima Kasih

Terima Kasih kepada Pihak Civitas Akademik UNIKA Musi Charitas Palembang yang telah memberi dukungan finansial dan upaya lainnya sehingga penelitian ini dapat diselesaikan dan dipublikasikan dalam bentuk Jurnal.

## Referensi

- [1] Pranajaya, "Analisis dan pencegahan plagiarisme di kalangan mahasiswa: studi kasus di fakultas teknologi informasi Universitas Yarsi," in *Proc. SENDI\_U.*, Jul. 2017, pp. 108-116.
- [2] H. Soelistryo, *Plagiarisme: Pelanggaran Hak Cipta dan Etika*. Yogyakarta, Indonesia: Kanisius, 2011.
- [3] A. Rachmanidah, "Analisis tingkat plagiarisme karya ilmiah jurnal mahasiswa ilmu perpustakaan undip tahun 2015-2016 dengan software turnitin," *Jurnal Ilmu Perpustakaan*, vol. 7, no. 3, pp. 11-20, Aug. 2018.
- [4] D. Purwitasari, "Deteksi keberadaan kalimat sama sebagai indikasi penjiplakan dengan algoritma hashing berbasis n-gram," *Jurnal Ilmiah KURSOR*, vol. 6, no. 1, pp. 37-44, Mar. 2017.
- [5] Ismail, "Aplikasi berbasis web pendeteksi plagiarisme menggunakan algoritma himpunan kata," *Jurnal Infotel*, vol. 6, no. 2, pp. 95-100, Nov. 2014.
- [6] N. Safaat, *Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android*, 1st ed., Bandung, Indonesia: Informatika, 2012.
- [7] H. Soelistryo, *Plagiarisme: Pelanggaran Hak Cipta dan Etika*. Yogyakarta, Indonesia: Kanisius, 2011.
- [8] *General Frequently Asked Questions: A Plagiarism FAQ*. IEEE. 2017. [Online]. Available: <https://www.ieee.org/publications/rights/copyright-faqs.html>

- [9] S. Bakri, "Dokumentasi digital: aplikasi penyimpanan dan pengambilan dokumen digital tenaga pendidik," *SEMNASSTEKNOMEDIA*, vol. 5, no. 1, 2017.
- [10] H. Fatta, *Analisis dan Perancangan Sistem Informasi*. Yogyakarta, Indonesia: Andi, 2007.
- [11] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. 7th ed., New York, USA: McGraw-Hill Higher Education, 2015.
- [12] D. Utama, "Rancang bangun sistem informasi pemesanan custom furniture berbasis website," *J-SAKTI*, vol. 3, no 1, pp. 53-67, Mar. 2019.
- [13] S. N. Lolyta, R. Y. Dillak, and F. E. Laumal, "Sistem deteksi plagiarisme lintas bahasa menggunakan algoritma tf-idf," *Jurnal Ilmiah Flash*, vol. 5, no. 1, pp. 29-32, Jun. 2019.
- [14] R. Sarno and F. Rahutomo, "Penerapan algoritma weighted tree similarity untuk pencarian semantik," *JUTI*, vol. 7, no. 1, pp. 35-42, Jan. 2008.
- [15] N. H. Ariyani, Sutardi, and R. Ramadhan, "Aplikasi pendeteksi kemiripan isi teks dokumen menggunakan metode Levenshtein distance," *Jurnal SemanTIK*, vol. 2, no. 1, pp. 279-286, Jan. 2016.
- [16] A. Iqbal and Nurdin, "Implementasi aplikasi E-Boarding house di Kota Lhokseumawe menggunakan algoritma Levenshtein Distance," *Jurnal Sistem Informasi*, vol. 1, no. 1, pp. 139-167, May 2017.
- [17] Nurdin and A. Munthoha, "Sistem pendeteksian kemiripan judul skripsi menggunakan algoritma Winnowing," *Jurnal InfoTekJar*, vol. 2, no. 1, pp. 90-97, Sept. 2017.