Metode Cosine Similarity Untuk Mendeteksi Kemiripan Pada Dokumen Teks

Supiyanto*[1], Sriyono[2],)
[1] Program Studi Sistem Informasi FMIPA Universitas Cenderawasih
[2]Program Studi Ilmu Administrasi Publik, FISIP, Universitas Cenderawasih

E-mail: supi6976@gmail.com

Article Info
Article History
Received: 21/01/2023
Revised: 21/04/2023
Published: 21/08/2023

Keywords:
Cosine Similarity, Tokenizing, Folding, Pre-processing, Python

Abstract
The cosine similarity method is a method that can be used to calculate the similarity between two objects expressed in two vectors using keywords (keywords) of a document as a measure. This study aims to implement the cosine similarity method with Python programming to find document similarities. The text data used in this study is in the form of files with a text extension. Stages of research methods included literature studies, algorithm analysis, program design, and testing. The method used in this study is the cosine similarity method. At this stage, initially, the system will read data in the form of text. The system will do pre-processing, such as case folding, tokenizing, and so on, to produce maximum similarity between documents. Document similarity is done by comparing one document with another using the cosine similarity method. The result of the application is in the form of several similarities between documents in the form of percentages. The output of this research is creating an application that can be used to determine the similarity between documents.

Artikel Info
Sejarah Artikel
Diterima: 21/01/2023
Direvisi: 21/04/2023
Dipublikasi: 21/08/2023

Kata kunci:
Cosine Similarity, Tokenizing, Folding, Pre-processing, Kemiripan, Python

Abstrak
Metode cosine similarity merupakan metode yang dapa digunakan untuk menghitung kemiripan antara dua buah objek yang dinyatakan dalam dua buah vector dengan menggunakan keywords (kata kunci) dari sebuah dokumen sebagai ukuran. Tujuan dari penelitian ini yaitu mengimplementasikan metode cosine similarity dengan bahasa pemograman Python untuk mencari kemiripan dari dokumen. Data teks yang digunakan pada penelitian ini berupa file dengan berektensi txt. Tahapan metode penelitian yang dilakukan studi literatur, analisa algoritma, perancangan program dan pengujian. Metode yang digunakan pada penelitian ini yakni Metode cosine similarity. Tahapannya, semula sistem akan membaca data yang berupa teks, kemudian sistem akan melakukan pre-processing seperti case Folding, Tokenizing dan sebagainya guna menghasilkan kemiripan antar dokumen yang maksimal. Kemiripan dokumen dilakukan dengan cara membandingkan antara dokumen yang satu dengan yang lainya menggunakan metode cosine similarity. Hasil aplikasi berupa angka kemiripan antara dokumen dalam bentuk persentase. Luaran penelitian ini, terciptanya suatu aplikasi yang dapat digunakan untuk menentukan kemiripan antar suatu dokumen.

I.PENDAHULUAN

Dalam bidang pendidikan, salah satu dampak dari perkembangan dunia teknologi yaitu adanya jurnal online. Jurnal online merupakan dokumen digital yang sangat dibutuhkan dalam semua bidang, baik pendidikan, politik, ekonomi, dll. yang berisi koleksi dan terbitan atau transmisi mengenai berita dan hasil-hasil penelitian mengenai media. Jurnal online merupakan versi digital dari jurnal cetak yang sering dijumpai di perpustakaan.

Jurnal online memiliki beberapa keuntungan bagi pembaca, diantaranya yaitu mudah dibaca dimana saja tanpa membawa kertas cetakan. Selain memiliki keuntungan, jurnal online juga memiliki kekurangan yaitu pembaca sangat mudah untuk melakukan penjiplakan, mudah untuk di copy-paste tanpa membaca keseluruhan isi jurnal.

Perilaku penjiplakan atau biasa disebut plagiat sering terjadi dalam lingkungan akademisi baik

sekolah maupun di perguruan tinggi. Salah satu bentuk perilaku plagiat yang sering dilakukan yakni dengan meng-copy-paste-edit suatu isi jurnal online. Perilaku plagiat sendiri dapat ditemukan dalam bentuk kutipan pada sebuah dokumen (Firdaus,2003)

Cara mudah untuk mendeteksi plagiat yaitu dengan penggunaan search engine atau mesin pencarian dengan memasukkan kata kunci tema dokumen dan membiarkan mesin pencarian menemukan dokumen yang dijiplak (Firdaus,2003).

Salah satu metode yang digunakan peneliti dalam mesin pencarian yaitu metode Cosine Similarity. Cosine Similarity merupakan metode yang digunakan untuk menghitung tingkat kemiripan (similarity) antar dua buah objek yang berbobot. (Sugiyamta, 2015).

Metode TF-ID Cosine Similarity bisa digunakan untuk menganalisa kesamaan ataukemiripan suatu dokumen teks dengan dokumen lainnya. Hal ini bisa digunakanuntuk membandingkan suatu karya tulis, apakah plagiat atau bukan. Dan seberapa persen kemiripannya dengan karya tulis yang lain. (Nuramijaya,2020)

II.METODE PENELITIAN

Pada Metode Cosine Similarity, teks masukan terlebih dahulu diolah menjadi data numerik agar dapat dilakukan pengolahan lebih lanjut. sehngga dalam text mining ada istilah pre-processing data, yaitu proses pendahuluan yang diterapkan pada data teks dengan tujuan menghasilkan data yang siap diolah baik itu data berupa teks maupun secara numerik.

Ada 2 tahap metode penelitian yang digunakan pada penelitian kemiripan teks yakni Pre-processing data dan Cosine Similarity.

Pre-processing

Beberapa pre-processing data yang diguna-kan dalam penelitian ini antara lain: Case Folding, Tokenizing, Stopwords/Filtering, Stemming, Distribusi Frekuensi. Tahap pre-processing data yang digunakan dalam penelitian ini antara lain :

Case folding

Case folding adalah salah satu bentuk text pre-processing yang bertujuan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Beberapa hal yang dapat dilakukan dalam tahap case folding, diantaranya, mengubah text menjadi lowercase, menghapus angka, menghapus tanda baca dan menghapus whitepace (karakter kosong)

Tokenizing

Tokenizing merupakan proses pemisahan teks menjadi potongan-potongan kata, angka, simbol, tanda baca dan entitas penting lainnya. Contoh, Misalnya diberikan kalimat "rumah idaman adalah rumah yang bersih." Dengan proses Tokenizing akan dihasilkan kata-kata : 'rumah', 'idaman', 'adalah', 'rumah', 'yang', 'bersih', '.'

Filtering (Stopword Removal)

Filtering adalah tahap mengambil kata-kata penting dari hasil token dengan menggunakan algoritma stoplist (membuang kata kurang penting) atau wordlist (menyimpan kata penting).

Stopword adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Contoh stopword dalam bahasa Indonesia adalah “yang”, “dan”, “di”, “dari”, dll. Makna di balik penggunaan stopword yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, sehingga didapat kata-kata penting sebagai gantinya. (Nugroho, 2019).

Misalnya diberikan kalimat "Andi kerap melakukan transaksi rutin secara daring atau online. Menurut Andi belanja online lebih praktis & murah." Maka hasil proses filtering yakni : 'andi', 'kerap', 'transaksi', 'rutin', 'daring', 'online', 'andi', 'belanja', 'online', 'praktis', 'murah'

Stemming

Stemming adalah teknik yang digunakan untuk mengekstrak bentuk dasar kata dengan menghilangkan imbuhan dari kata tersebut. Ini seperti menebang dahan pohon ke batangnya. Misalnya, akar kata eating, eats, eaten adalah eat.

Sastrawi

Python Sastrawi merupakan library sederhana yang dapat mengubah kata berimbuhan berbahasa Indonesia menjadi bentuk dasarnya. Misalnya diberikan kalimat " Mereka meniru-nirukannya". Hasil proses Sastrawi. Stemmer yakni: "mereka tiru"

Frequency Distribution

Menghitung frekuensi kemunculan setiap tokens (kata) dalam teks. Misalnya diberikan kalimat "Andi kerap melakukan transaksi rutin secara daring atau online. Menurut Andi belanja online lebih praktis & murah."

Hasil proses Frequency Distribution yakni: ('andi', 2), ('online', 2), ('kerap', 1), ('melakukan', 1), ('transaksi', 1), ('rutin', 1), ('secara', 1), ('daring', 1), ('atau', 1), ('menurut', 1), ('belanja', 1), ('lebih', 1), ('praktis', 1), ('murah', 1)

Cosine Similarity

Cosine Similarity merupakan metode yang digunakan untuk menghitung tingkat kemiripan antar dua buah objek (Rizki Tri Wahyuni, Dhidik Prastiyanto, 2017).. Metode Cosine Similarity digunakan untuk menghitung nilai cosinus sudut antara dua vektor dan mengukur kemiripan antar dua dokumen dengan menggambarkan suatu kesamaan antara vektor query dan vektor dokumen yang menghasilkan sudut cosinus x diantara dua vektor tersebut. Nilai sudut cosinus antara dua vektor menentukan kesamaan dua buah objek yang dibandingkan dimana nilai terkecil adalah 0 dan nilai terbesar adalah 1. Nilai 0 menandakan bahwa dokumen yang dibandingkan tidak ada kemiripan, dan semakin mendekati nilai 1 maka dokumen tersebut memiliki tingkat kemiripan yang besar. (Pratama, 2018).

III. HASIL DAN PEMBAHASAN

A. Algoritma

Secara bahasa alami, berikut ini algoritma atau langkah-langkah yang digunakan untuk menentukan kemiripan teks menggunakan Cosine Similarity.

Langkah 1. Tentukan teks dokumen yang akan dicari kemiripannya, misalnya T1, T2 dst.
Langkah 2. Tentukan juga dokumen teks yang akan digunakan sebagai pembanding untuk dicari kemiripannya, misalnya T0
Langkah 3. Lakukan proses tokenisasi dan stemming atau biasa disebut preprocessing untuk semua dokumen, untuk menghilangkan kata sambung dan imbuhan, sehingga tersisa kata dasar saja dan tanpa tanda baca.
Langkah 4. Lakukan ekstrak semua dokumen (T0, T1, T2) untuk memperoleh kata-kata yang ada di dalam dokumen.
Langkah 5. Menghitung Frekuensi Kata dari setiap Dokumen (TF)
Langkah 6. Lakukan perkalian antara Frekuensi dokumen pembanding T0 dengan frekuensi dokumen yang akan dicari kemiripannya (T1, T2)
Langkah 7. Kemudian langkah terakhir mencari nilai similarity antara A dan B menggunakan rumus :$Cosine\cos(A_k, B_k) = \frac{\sum_k (A_k * B_k)}{\sqrt{\sum_k {A_k}^2} * \sqrt{\sum_k {B_k}^2}}$
Langkah 8. Tampilkan persentase kemiripan antar dokumen

B. Implementasi coding

Penggalan coding yang digunakan untuk mendekteksi kemiripan dokumen menggunakan bahasa pemogrman Python.

i. Import Library yang digunakan

```
import nltk
import re
import math
from collections import Counter
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```

ii. Fungsi untuk menentukan kemiripan teks

```
def get_cosine(vec1, vec2):
intersection = set(vec1.keys()) & set(vec2.keys())
numerator = sum([vec1[x] * vec2[x] for x in intersection])

denominator = math.sqrt(sum1) * math.sqrt(sum2)
if not denominator:
return 0.0
else:
return float(numerator)/denominator
```

iii.Fungsi untuk merubah teks menjadi vektor

```
def text_to_vector(text):
word = WORD.findall(text)
return Counter(word)
```

iv.Coding untuk membaca file

```
i=0
for x in file:
   f0=open(path + file[i],'r')
   i+=1
print(f'\n')
```

v.Coding Untuk pre-processing

```
f[k]=file[k].lower()
tokens[k] = nltk.tokenize.word_tokenize(f[k])
freq_tokens[k] = nltk.FreqDist(tokens[k])


list_stopwords = set(stopwords.words('indonesian'))

tokens_without_stopword[k] = [word for word in freq_tokens[k] if not word in list_stopwords]

factory = StemmerFactory()

stemmer = factory.create_stemmer()

list_tokens[k] = tokens_without_stopword[k]

output[k] = [stemmer.stem(token) for token in list_tokens[k]]
```

vi.Coding mengubah teks menjadi vektor

```
for i in range(jml):
   f[i]=text_to_vector(t[i])
```

C. Hasil Pengujian

Sebagai pengujian terhadap aplikasi yang dibuat berikut ini kami sampaikan hasil pengujian. Pada pengujian ini kami gunakan empat (4) file ber-ekstensi txt dengan isi filenya sebagai berikut.

```
SECARA HARFIAH, CITRA (IMAGE) ADAL
AH GAMBAR PADA BIDANG DWIMATRA (DU
A-DIMENSI). DITINJAU DARI SUDUT PA
NDANGAN MATEMATIS, CITRA MERUPAKAN
 FUNGSI MENERUS (CONTINUE) DARI IN
TENSITAS CAHAYA PADA BIDANG DWIMAT
RA. SUMBER CAHAYA MENERANGI OBJEK,
 OBJEK MEMANTULKAN KEMBALI SEBAGIA
N BERKAS CAHAYA TERSEBUT, PANTULAN
 CAHAYA INI DITANGKAP OLEH ALAT OP
TIK SEHINGGA BAYANGAN OBJEK YANG D
ISEBUT CITRA TERSEBUT TEREKAM (MUN
IR, 2004). DAN MENURUT PUTRA (2013
), CITRA DIGITAL DAPAT DIARTIKAN S
EBAGAI SUATU FUNGSI DUA DIMENSI F(
X.Y), BERUKURAN M BARIS DAN N KOLO
M SEDANGKAN X DAN Y ADALAH POSISI
KOORDINAT SPASIAL DAN AMPLITUDOF D
I TITIK KOORDINAT (X,Y) YANG DINAM
AKAN INTENSITAS ATAU TINGKAT KEABU
AN DARI CITRA PADA TITIK TERSEBUT
```

Isi dari file T1.txt

```
Secara umum, pengolahan citra digi
tal menunjuk pada pemrosesan gamba
r dua dimensi menggunakan komputer
. Dalam konteks yang lebih luas, p
engolahan citra digital mengacu pa
da pemprosesan setiap data dua dim
ensi. Citra digital merupakan sebu
ah larik (array) yang berisi nilai
-nilai real maupun kompleks yang d
irepresentasikan dengan deretan bi
t tertentu. Citra digital dapat di
definisikan secara matematis sebag
ai fungsi intensitas dalam 2 varia
ble x dan y, yang dapat dituliskan
 f(x, y), dimana (x, y) merepresen
tasikan koordinat spasial pada bid
ang 2 dimensi dan f(x, y) merupaka
n intensitas cahaya pada koordinat
 tersebut. Citra digital merupakan
 representasi citra asal yang bers
ifat kontinyu. Untuk mengubah citr
a yang bersifat kontinu diperlukan
 sebuah cara untuk mengubahnya dal
am bentuk data digital. Komputer m
enggunakan sistem bilangan biner u
ntuk memecahkan masalah ini. Denga
n menggunakan sistem bilangan bine
r ini, citra dapat diproses dalam
komputer dengan sebelumnya mengeks
trak informasi citra analog asli d
an mengirimnya ke komputer dalam b
entuk biner. Proses ini disebut de
ngan digitalisasi
```

Isi dari file T2.txt

PENGOLAHAN CITRA (IMAGE PROCESSING) ADALAH PEMROSESAN CITRA, KHUSUSNYA DENGAN MENGGUNAKAN COMPUTER, MENJADI CITRA YANG KUALITASNYA LEBIH BAIK. PENGOLAHAN CITRA INI SANGAN DIPERLUKAN KARENA WALAUPUN CITRA SANGAT KAYA DENGAN INFORMASI, NAMUN SERINGKALI CITRA MENGALAMI PENURUNAN MUTU (DEGRADASI), MISALNYA MENGANDUNG CACAT ATAU DERAU (NOISE), WARNANYA TERLALU KONTRAS, KURANG TAJAM, KABUR (BLURRING), DAN SEBAGAINYA. TENTU SAJA CITRA SEMACAM INI MENJADI LEBIH SULIT DIINTERPRETASI KARENA INFORMASI YANG DISAMPAIKAN OLEH CITRA TERSEBUT MENJADI BERKURANG
Isi dari file T3.txt

Citra merupakan suatu representasi (gambaran), image, kemiripan, atau imitasi dari suatu objek dalam bidang dua dimensi. Sebagai suatu sistem keluaran dalam suatu perekaman data, citra dapat berupa optik, analog, maupun digital. Bersifat optik misalnya adalah foto, bersifat analog misalnya dapat berupa sinyal-sinyal video seperti pada monitor televisi, dan dapat bersifat digital dimana citra dapat secara langsung disimpan pada media penyimpanan. Citra yang berasal dari penglihatan manusia terdiri dari dua komponen, yaitu iluminasi dan refleksi. Iluminasi [i(x, y)] merupakan jumlah cahaya dari sumber cahaya yang mengenai objek. Sedangkan reflektansi [r(x,y)] merupakan jumlah cahaya yang dipantulkan oleh objek ke mata. Nilai iluminasi dipengaruhi sumber cahaya, dan reflektansi ditentukan oleh karakteristik objek yang ditangkap. Dimana reflektansi bernilai 0 sampai 1. 0 apabila objek menyerap cahaya, dan 1 apabila objek memantulkan cahaya secara sempurna
Isi dari file T4.txt

Setelah mengalami Pre-processing, berikut tampilan teks dalam bentuk vektor dan frekuensi kemunculan kata.

'x': 2, 'y': 2, 'dimensi': 2, 'pantul': 2, 'citra': 1, 'cahaya': 1, 'objek': 1, 'bidang': 1, 'dwimatra': 1, 'fungsi': 1, 'intensitas': 1, 'koordinat': 1, 'titik': 1, 'harfiah': 1, 'image': 1, 'gambar': 1, 'dua': 1, 'tinjau': 1, 'sudut': 1, 'pandang': 1, 'matematis': 1, 'terus': 1, 'continue': 1, 'sumber': 1, 'rang': 1, 'berkas': 1, 'tangkap': 1, 'alat': 1, 'optik': 1, 'bayang': 1, 'rekam': 1, 'munir': 1, '2004': 1, 'putra': 1, '2013': 1, 'digital': 1, 'arti': 1, 'f': 1, 'ukur': 1, 'm': 1, 'baris': 1, 'n': 1, 'kolom': 1, 'posisi': 1, 'spasial': 1, 'amplitudof': 1, 'nama': 1, 'tingkat': 1, 'abu': 1
T1.txt setelah dipre-precessing dalam bentuk text to vektor

'representasi': 3, 'ubah': 2, 'proses': 2, 'citra': 1, 'digital': 1, 'komputer': 1, 'x': 1, 'y': 1, 'dimensi': 1, 'biner': 1, 'olah': 1, 'data': 1, 'intensitas': 1, '2': 1, 'f': 1, 'koordinat': 1, 'sifat': 1, 'bentuk': 1, 'sistem': 1, 'bilang': 1, 'pemrosesan': 1, 'gambar': 1, 'konteks': 1, 'luas': 1, 'acu': 1, 'pemprosesan': 1, 'larik': 1, 'array': 1, 'isi': 1, 'nilai': 1, 'real': 1, 'kompleks': 1, 'deret': 1, 'bit': 1, 'definisi': 1, 'matematis': 1, 'fungsi': 1, 'variable': 1, 'tulis': 1, 'mana': 1, 'spasial': 1, 'bidang': 1, 'cahaya': 1, 'kontinyu': 1, 'kontinu': 1, 'pecah': 1, 'ekstrak': 1, 'informasi': 1, 'analog': 1, 'asli': 1, 'kirim': 1, 'digitalisasi': 1
T2.txt setelah dipre-precessing dalam bentuk text to vektor

'citra': 1, 'olah': 1, 'informasi': 1, 'image': 1, 'processing': 1, 'pemrosesan': 1, 'computer': 1, 'kualitas': 1, 'sangan': 1, 'kaya': 1, 'seringkali': 1, 'alami': 1, 'turun': 1, 'mutu': 1, 'degradasi': 1, 'kandung': 1, 'cacat': 1, 'derau': 1, 'noise': 1, 'warna': 1, 'kontras': 1, 'tajam': 1, 'kabur': 1, 'blurring': 1, 'sulit': 1, 'interpretasi': 1, 'kurang': 1
T3.txt setelah dipre-precessing dalam bentuk text to vektor

'simpan': 2, 'pantul': 2, 'objek': 1, 'cahaya': 1, 'citra': 1, 'sifat': 1, 'iluminasi': 1, 'reflektansi': 1, 'optik': 1, 'analog': 1, 'digital': 1, 'mana': 1, 'x': 1, 'y': 1, 'sumber': 1, '0': 1, '1': 1, 'representasi': 1, 'gambar': 1, 'image': 1, 'mirip': 1, 'imitasi': 1, 'bidang': 1, 'dimensi': 1, 'sistem': 1, 'keluar': 1, 'rekam': 1, 'data': 1, 'foto': 1, 'sinyal': 1, 'video': 1, 'monitor': 1, 'televisi'

: 1, 'langsung': 1, 'media': 1, 'asal': 1, 'lihat': 1, 'manusia': 1, 'komponen': 1, 'refleksi': 1, 'i': 1, 'r': 1, 'mata': 1, 'nilai': 1, 'pengaruh': 1, 'tentu': 1, 'karakteristik': 1, 'tangkap': 1, 'nila': 1, 'serap': 1, 'sempurna': 1
T4.txt setelah dipre-precessing dalam bentuk text to vektor

Adapun berikut masing-masing irisan kata yang ada di f1 dengan f2; f1 dengan f3 dan f1 dengan f4.

{'gambar', 'spasial', 'digital', 'koordinat', 'dimensi', 'intensitas', 'x', 'citra', 'matematis', 'cahaya', 'bidang', 'y', 'f', 'fungsi'}
Irian antara file T1 dan T2

{'image', 'citra'}
Irian antara file T1 dan T3

{'gambar', 'pantul', 'tangkap', 'digital', 'dimensi', 'x', 'citra', 'image', 'cahaya', 'bidang', 'y', 'rekam', 'sumber', 'optik', 'objek'}
Irian antara file T1 dan T3

Adapun kemiripan dokumen diantara T1 terhadap T2, T3 dan T4 sebagai berikut :

- Kemiripan antara file[1] dan file[2] = 26.79 %
- Kemiripan antara file[1] dan file[3] = 4.93 %
- Kemiripan antara file[1] dan file[4] = 35.61 %

IV. KESIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil pembahasan pada pengujian aplikasi yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Aplikasi dirancang untuk mendekteksi kemiripan dokumen menggunakan algoritma cosine similarity berjalan sangat baik dan digunakan untuk mengetahui besaran kemiripan dokumen.

2. Pengujian terhadap dua dokumen yang sama persis redaksinya dan penulisannya, aplikasi mendeteksi kemiripannya 100%

3. Penulisan kalimat yang hanya merubah dari kalimat aktif menjadi pasif atau sebaliknya, aplikasi mendeteksi kemiripan dokumen di atas 70% untuk data yang tidak dilakukan pre-processing dan diatas 80% untuk data yang dilakukan pre-processing.

4. Dokumen yang diuji kemiripan tanpa pre-processing mempunyai nilai kemiripan lebih kecil dibandingkan dengan dokumen yang dilakukan pre-processing terlebih dahlu lalu diuji kemiripannya.

B. Saran

Adapun saran-saran yang dapat dilakukan penelitian ataupun pengembangan selanjutnya adalah sebagai berikut:

1. Dokumen yang digunakan untuk mendeteksi kemiripan antara dokumen pada aplikasi ini masih dalam bentuk file yang ber-ektensi txt oleh karena masih perlu untuk diuji untuk dokumen jenis lain seperti pdf, word atau yang lain.
2. Aplikasi ini masih sangat terbuka untuk dikembangkan sehingga dapat suatu saat nanti dapat digunakan sebagai aplikasi alternatif untuk mendeteksi kemiripan antar dokumen atau plagiarisme
3. Sistem dapat dikembangkan dengan menggunakan GUI, sehingga lebih menarik dan familiar bagi user.

DAFTAR RUJUKAN

Firdaus, H. B. (2003). Algoritma Rabin-Karp. *Ilmu Komputer Dan Teknologi Informasi III*, *III*, 1–5.

Ismail, & Yunarso Eka, W. (2015). Aplikasi Berbasis

Web Pendeteksi Plagiarisme Menggunakan Algoritma Himpunan Kata. *Jurnal Informatika,Telekomunikasi Dan Elektronika*, *6*(2), 2–7. https://doi.org/10.20895/infotel.v6i2.79

Nugroho, K. S. (2019a). *basic-text-preprocessing*. Dasar Text Preprocessing Dengan Python. https://ksnugroho.medium.com/

Nugroho, K. S. (2019b). *Dasar text preprocessing-dengan python*. https://github.com/ksnugroho/basic-text-preprocessing/blob/master/text-preprocessing.ipynb

Nuramijaya. (2020). *Menghitung Kemiripan Dokumen dengan TF-IDF Cosine Similarity*.

Perdana, K. (2014). Pencarian dan perangkingan Obat Tradisional berdasarkan Gejala Penyakit Menggunakan Metode Cosine Similarity. *Skripsi*.

Pratama, R. P. (2018). *Aplikasi deteksi plagiarisme menggunakan metode cosine similarity*. http://etheses.uin-malang.ac.id/id/eprint/13894%0Ahttp://etheses.uin-malang.ac.id/13894/1/14650044.pdf

Rizki Tri Wahyuni, Dhidik Prastiyanto, E. S. (2017). Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi. *Jurnal Teknik Elektro*, *Vol. 9 No.*, 18–23.

Sugiyamta. (2015). Sistem Deteksi kemiripan Dokumen Dengan Algoritma Cosine Similarity dan Single Pass Clustering. *Dinamika Informatika*, *7*.

Wibowo, A. (2012). Mencegah dan menanggulangi plagiarisme di dunia. *Departemen Administrasi Dan Kebijakan Kesehatan Fakultas Kesehatan Masyarakat Universitas Indonesia*.