

**Pembobotan Korpus Twitter Tentang *Data Science*  
*Text Mining Text Retrieval* Menggunakan *Cosine*  
*Smilarity* dan *Latent Semantic Indexing***

**TUGAS KELOMPOK**

**Disusun Untuk Memenuhi Tugas Mata Kuliah Temu Kembali Informasi**

**Dosen Pengampu : Retnani Latifah, M.Kom**



**Disusun Oleh :**

<b>MUHAMMAD REZA</b>	<b>2019470055</b>
<b>SELAMET SAPUTRA</b>	<b>2019470069</b>
<b>SYECHAN AHMAD ZIDAN</b>	<b>2019470110</b>

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH JAKARTA  
2022**

## DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iii
DAFTAR TABEL .....	iv
LAMPIRAN.....	v
BAB I.....	1
1.1. Latar belakang masalah .....	1
1.2. Identifikasi Masalah .....	2
1.3. Rumusan Masalah .....	2
1.4. Batasan Masalah .....	3
BAB II .....	4
2.1. Data Acquisition.....	4
2.2. Data Exploration .....	5
2.3. <i>Preprocessing</i> .....	6
BAB III.....	8
3.1. <i>Term Frequency Inverse Document Frequency</i> .....	8
3.2. <i>Latent Semantic Indexing</i> .....	10
3.3. <i>Vector Space Model</i> .....	10
BAB IV .....	12
Kesimpulan.....	12
DAFTAR PUSTAKA .....	13

## **DAFTAR GAMBAR**

Gambar pengambilan data.....	4
Gambar korpus 1 .....	5
Gambar korpus 2 .....	5
Gambar perhitungan LSI .....	10

## DAFTAR TABEL

Tabel 2.1 Perubahan teks .....	5
Tabel 2.2 <i>stopwords</i> sering muncul .....	6
Tabel 2.3 hasil preprocessing.....	7
Tabel 3.1 <i>Term frequency</i> muncul di dokument .....	8
Tabel 3.2 <i>Term frequency</i> tidak muncul di dokument .....	8
Tabel 3.3 tf-idf <i>query</i> .....	9
Tabel 3.4 Hasil <i>Cosine Smiliarity</i> .....	11

## LAMPIRAN

Kode : [tugas-tki/Tugas\\_LATENT\\_SPACE\\_INDEX.ipynb at main · ackermanjayjay/tugas-tki \(github.com\)](https://github.com/ackermanjayjay/tugas-tki/blob/main/Tugas_LATENT_SPACE_INDEX.ipynb)

Data : [tugas-tki/data/data twitter/data about ai at main · ackermanjayjay/tugas-tki \(github.com\)](https://github.com/ackermanjayjay/tugas-tki/blob/main/data/twitter/data_about_ai)

## BAB I

### 1.1. Latar belakang masalah

Saat ini sedang gencarnya *data science*, *text mining*, dan *text retrieval*. *Data science* menurut David M. Blei dan Padhraic Smyth adalah turunan atau perhitungan menggunakan statistika untuk melakukan prediksi (Blei & Smyth, 2017). *Text mining* adalah pengolahan dari kumpulan document yang dipecah menjadi teks untuk mengetahui informasi yang bermanfaat dengan menggunakan perhitungan kalkulasi matematika (Sabrani et al., 2020). *Information retrieval* atau pengambilan informasi atau text retrieval adalah tugas untuk mengambil informasi yang sesuai atau relevan dari kumpulan korpus yang mewakili permintaan (kueri) (Djenouri et al., 2021).

*Text Mining* adanya *preprocessing* dan ekstraksi fitur, tahap *preprocessing* terdiri dari *case folding*, *stop word removal*, *stemming*, *word normalization* untuk mengatasi *overfitting* dari hasil *stemming* (Ma'rifah et al., 2020). ekstraksi fitur dalam text mining ada *term frequency* melihat setiap kata yang muncul didalam dokumen atau korpus, untuk menghitung *inverse document* diperlukan masing-masing kemunculan *term frequency* di setiap document atau korpus lalu dikalkulasi dengan rumus *Inverse Document Frequency* (IDF) .Maka dari hasil nilai ekstraksi fitur tf-idf ini digunakan untuk perhitungan similaritas, dan untuk beberapa metode dalam pendekatan statistika (Setyawan et al., 2021).

*Vector Space Model* (VSM). Sebuah model yang digunakan untuk mengukur sebuah kueri antara suatu dokumen dengan suatu kata kunci atau *keyword* (Susanti et al., 2020). *Vector space* adalah geometri berdimensi besar, ruang yang batas-batasnya ditentukan oleh *vector*. Konsep dasar *vector space model* adalah menghitung jarak vector antara dokumen dengan kata kunci yang dimasukkan kemudian mengurutkan berdasarkan tingkat kedekatannya (Susanti et al., 2020). Salah satu model *Vector Space Model* adalah *Cosine Smiliarity* untuk memodelkan *document* text sebagai vector kata, dengan menggunakan kesamaan antara dua dokumen (Ma'rifah et al., 2020).

*Latent semantic indexing* (LSI). Adalah sebuah Teknik information retrieval di mana sekumpulan kata digunakan untuk mengidentifikasi sekumpulan dokumen yang paling relevan. Kueri terhadap sekumpulan dokumen yang telah menjalani LSI akan mengembalikan hasil yang secara konseptual mirip dengan kriteria pencarian. Matriks dihitung di mana baris sesuai dengan dokumen dan kolom sesuai dengan istilah. Matriks ini kemudian direduksi menggunakan teknik *singular value decomposition* (SVD) untuk menemukan kumpulan dokumen yang paling penting. Setelah mendapatkan perkiraan peringkat rendah dari matriks term-dokumen menggunakan SVD, matriks yang dihitung digunakan untuk memproyeksikan setiap vektor dalam matriks kueri dalam ruang yang diperkecil (Parajuli & Shakya, 2018)

### 1.2. Identifikasi Masalah

Berdasarkan permasalahan di latar belakang, permasalahan tersebut dilakukan identifikasi sebagai berikut :

1. Melakukan *preprocessing document* teks
2. Melakukan ekstraksi fitur teks menggunakan *Term Frequency Inverse Document Frequency*
3. Menghitung jarak teks dokumen menggunakan *Cosine Smilarity* dan *Latent semantic indexing*

### 1.3. Rumusan Masalah

Berdasarkan permasalahan diatas, akan dilakukan perumusan atau kajian sebagai berikut :

1. Bagaimana melakukan preprocessing data teks *document* ?
2. Bagaimana cara melakukan ekstraksi fitur text dari *document* menggunakan *Term Frequency Inverse Document Frequency* ?
4. Bagaimana cara menghitung jarak teks *document* twitter menggunakan *Cosine Smilarity* dan *Latent semantic indexing* ?

#### **1.4. Batasan Masalah**

1. Mengolah teks *document* dari twiiter tentang *data science text mining text retrieval*
2. Menggunakan bantuan *library* regex,NLTK, Pandas, Numpy untuk mengolah text data dan menghitung hasil jarak kedekatan dokument
3. Menggunakan data dari *twitter* berjumlah 84 dokument hanya bahasa inggris
4. Mengetahui hasil kedekatan jarak antara *document*



## BAB II

### 2.1. Data Acquisition

Data yang digunakan dalam laporan kali ini adalah data atau korpus yang diambil berasal dari twitter dengan cara *scrapping* menggunakan apify, korpus yang diambil berjumlah 84 dokument

< All tasks

### Twitter Scraper Task

orchid\_spoon/twitter-scraper-task

Actor quacker/twitter-scraper

▸ Input Information Runs Integrations Issues Settings

[Switch to JSON editor](#)

You can scrape Twitter either by search term or by URL. To try it out, just type in your search term and click ▸ **Start** to start scraping. If you need any guidance, just [follow this tutorial](#).

What search terms do you want to scrape? (optional) ⓘ

1 text mining ×

2 data science ×

3 Text Retrieval ×

[+ Add](#) [Bulk edit](#) [Remove empty fields](#)

Gambar pengambilan data

```

0 I think the message in Data Science needs to b...
1 Python libraries for:\n\n- Machine Learning\n-...
2 Free Data Science PDF Books \n📖:
3 Top tech skills for a #DataEngineer in 2022 🤖...
4 💡¿Se puede crear gráficos espectaculares que i...
...
59 Excellent retrieval skills in #BusheyHeathRead...
60 A novel adapter-based method for parameter-eff...
61 On @jhuc1sp YouTube: Changes in Tweet Geolocat...
62 Yes, I am looking for a summer 2023 research i...
63 trec: TREC collection (2010). A bipartite netw...
Name: full_text, Length: 64, dtype: object

```

Gambar korpus 1

```

0 Every story in the world has one of 6 basic pl...
1 SoLA invites you to a lecture on "Text Mining ...
2 Check out our events happening this week! \n\n...
3 The RuMOR team is growing! Thanks to @SSHRC_CR...
4 I'm doing a lot of preaching right now to coll...
5 Why my #Geosis package is simply the most robu...
6 Text Mining and Analytics #TextMining https://...
7 Let's speed up my booming Twitter career! Here...
8 Are you after a course that will teach you the...
9 Fundamentals of Predictive Text Mining (Texts ...
10 Awesome strategies for our humanities courses ...
11 meaning of life is number 42
12 Brisbane Data, Power BI and AI Bootcamp speake...
13 this is a possible tweet
14 It's a Tweet!
15 this is an example tweet
16 this is your next tweet
17 or, maybe, a possible badger
18 Python Text Mining: Perform Text Processing, W...
19 and now for something completely different
Name: full_text, dtype: object

```

Gambar korpus 2

## 2.2 Data Exploration

Karena dokument terpisah maka dilakukan penggabungan antara korpus 1 dan korpus 2, maka total data yang digabungkan berjumlah 84 dokument. Dokument 4 terdapat kalimat bukan bahasa inggris maka dilakukan perbaikan agar kalimat tersebut menjadi bahasa inggris.

Tabel 2.1 Perubahan teks

Bukan bahasa inggris	Bahasa Inggris
Se puede crear gráficos espectaculares	Can you create spectacular graphics

que incluyan los resultados de las pruebas estadísticas con rigor? statsplot lo vuelve simple y listo para publicar	that include the results of rigorous statistical tests?
---------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------

Pada tabel 2.1 kalimat diubah menjadi bahasa inggris, agar pada teks *preprocessing* tidak perlu memusingkan stemming bahasa non inggris karena bahasa sebelum dilakukan perubahan adalah bahasa spanyol.

Dokument yang sudah digabungkan memiliki stopwords yang semuanya bahasa inggris dapat dilihat pada tabel 2.2

Tabel 2.2 *stopwords* sering muncul

<i>Stopwords</i>	<i>Frequensi</i>
to	53
in	46
and	46
a	44
the	38
of	33
is	26
for	24
you	19
on	18

### 2.3 *Preprocessing*

Tahap *preprocessing* terdiri dari tahap :

1. *Casefolding*

Adalah tahap untuk mengecilkan huruf yang sebelumnya kapital, penghapusan tanda baca, penghapusan nomor dan link

2. *Stopword removal*

Adalah tahap untuk menghapus kata penghubung

### 3. *Stemming*

Adalah tahap untuk mereduksi kata menjadi kata dasar

### 4. *Word normalization*

5. Adalah tahap opsional yang digunakan untuk mencegah hasil overfitt pada saat tahap *Stemming*

Setelah tahap sudah ditentukan maha buat *pipeline* agar data dokument dapat di *preprocessing* yang dapat dilihat pada tabel 2.3

Tabel 2.3 hasil preprocessing

Sebelum <i>preprocessing</i>	Sesudah <i>preprocessing</i>
I think the message in Data Science needs to be: Don't believe everything you read. 🧵 #stats #datascience <a href="https://t.co/4jGMgmX8Nw">https://t.co/4jGMgmX8Nw</a>	think message data science needs believe everything read stats datascience jgmngx nw

Hasil dari preprocessing pada Tabel 2.3 dapat dilihat bahwa kata seperti “I” dihapus pada saat preprocessing, serta link seperti “<https://t.co/4jGMgmX8Nw>” juga dihapus karena ingin mengambil teksnya saja, untuk dilakukan pencarian informasi yang bermanfaat.

## BAB III

### 3.1. *Term Frequency Inverse Document Frequency*

Pada Tahap ini dokument yang sudah dilakukan *preprocessing* dipecah menjadi per kata, serta pada masing-masing kata yang muncul di dokument diberikan nilai 1, akan tetapi jika tidak muncul di dokument diberikan nilai 0. Pada tabel 3.1 dan tabel 3.2 adalah salah satu sampel term yang muncul di dokumen dan tidak muncul di dokumen.

Tabel 3.1 *Term frequency* muncul di dokument

<i>Term(kata)</i>	<i>Dokument frequency</i>	<i>Frequency in dokument</i>
data	23	1
text	44	1
mining	27	1
retrieval	18	1
python	11	1

Tabel 3.2 *Term frequency* tidak muncul di dokument

<i>Term(kata)</i>	Frequensi
data	0
text	0
mining	0
retrieval	0
python	0

Tahap *term* frekuensi sudah dilakukan maka dilakukan *Inverse Document Frequency*(IDF) dengan rumus :

$$N/df$$

$$idf = \log_{10}\left(\frac{N}{Df}\right)$$

N = Jumlah dokumen

*Document frequency*(DF)= jumlah kemunculan term yang muncul di dokument

*Frequency Qorpus*(FQ)

Maka perhitungannya dengan menggunakan sampel dari tabel 3.1

$$idf(data) = \log_{10}\left(\frac{84}{23}\right) = 0,563$$

Pada tahap *term frequency inverse document query* adalah tahap untuk mencari bobot berdasar kueri yang ingin ditentukan dilakukan dengan hasil *Inverse Document Frequency*(IDF), semisal kueri “data” dilakukan pencarian apakah data ada di *term* jika ada maka bernilai 1, sebagai berikut cara perhitungannya:

$$tfidfq = idf(term) * FQ$$

$$tfidfq(data) = idf(data) * FQ$$

$$tfidfq(data) = 0,563 * 1$$

Pada tabel *term frequency document inverse query*, dengan kueri masukkan “data”, “text”, “mining” menghasilkan hasil pengujian yang sudah dilakukan preprocessing, bahwa term “data” menghasilkan *inverse document frequency* yang paling besar diantara ketiga term.

Tabel 3.3 tf-idf query

Term	DF	N/df	IDF
data	23	3.652173913043478	0, 5625514500442887
text	44	1.9090909090909092	0.2808266095756942
mining	27	3.1111111111111111	0.49291552190289434
retrieval	18	4.666666666666667	0.6690067809585756
python	11	7.636363636363637	0.8828866009036567

### 3.2. Latent Semantic Indexing

Setelah tahap *Term Frequency Inverse Document Frequency* dilakukan hasil bobot dari perhitungan *Term Frequency Inverse Document Frequency* dijadikan *vector* lalu dilakukan reduksi *Singular Value Decomposition* untuk mengetahui relasi antara *term* dan *dokumen*, metode *Latent semantic indexing* menggunakan perhitungan sebagai berikut :

$$M = U \cdot \Sigma \cdot V^T$$

where

$$M = \begin{pmatrix} 0.8828866009036567 \\ 0.49291552190289434 \\ 0.2808266095756942 \\ 0.6690067809585756 \\ 0.5625514500442887 \end{pmatrix}$$

$$U = \begin{pmatrix} 0.6464344571581995 & -0.3609043080957560 & -0.2056164367324579 & -0.4898353138913518 & -0.4118905427500316 \\ 0.3609043080957560 & 0.9208884878254458 & -0.04507185664719853 & -0.1073736487184314 & -0.09028787674850156 \\ 0.2056164367324579 & -0.04507185664719853 & 0.9743214077725716 & -0.06117352038532266 & -0.05143931807054129 \\ 0.4898353138913518 & -0.1073736487184314 & -0.06117352038532266 & 0.8542677276390454 & -0.1225427057965508 \\ 0.4118905427500316 & -0.09028787674850156 & -0.05143931807054129 & -0.1225427057965508 & 0.8969568339211366 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 1.365778991399883 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$V = ( 1.0000000000000000 )$$

Gambar perhitungan LSI

### 3.3. Vector Space Model

Setelah tahap *Latent semantic indexing* dilakukan hasil bobot dari perhitungan tahap LSI dari masing-masing kueri di kalkulasikan menggunakan metode *Cosine Smiliarity* untuk mengetahui jarak *term* dari masing-masing document, metode *Cosine Smiliarity* menggunakan perhitungan sebagai berikut:

$$Cossim(q,j) = \frac{\sum_{i=1}^t (w_{ij} * w_{iq})}{\sqrt{\sum_{i=1}^t w_{ij}^2 * \sum_{i=1}^t w_{iq}^2}}$$

$w_{ij}$  = bobot tf-idf *term i* sampai *j*

$w_{iq}$  = bobot tf-idf *term i* sampai *q*

Tabel 3.4 Hasil *Cosine Smiliarity*

<i>Query</i>	Dokument terkuat	Hasil smiliaritas
data text mining retrieval python	8	0.9998938571906955



## BAB IV

### Kesimpulan

Kesimpulan mengenai data korpus yang digunakan, yaitu data dari media sosial twitter, yang kemudian digabungkan menggunakan *library pandas*, lalu dilakukan *preprocessing* terdiri dari *casefolding*, *stemming* menggunakan algoritma porter, *remove stopwords* menggunakan NLTK, *word normalization*, melakukan ekstraksi menggunakan tf-idf dengan *library scikit learn* agar dapat dilakukan kalkulasi menghitung jarak *term* dari masing-masing dokumen menggunakan *vector space model* metode *cosine smiliarity*, serta menghasilkan dokumen paling terkait adalah dokumen 20.

## DAFTAR PUSTAKA

- Blei, D. M., & Smyth, P. (2017). Science and data science. In *Proceedings of the National Academy of Sciences of the United States of America* (Vol. 114, Issue 33, pp. 8689–8692). National Academy of Sciences.  
<https://doi.org/10.1073/pnas.1702076114>
- Djenouri, Y., Belhadi, A., Djenouri, D., & Lin, J. C. W. (2021). Cluster-based information retrieval using pattern mining. *Applied Intelligence*, 51(4), 1888–1903. <https://doi.org/10.1007/s10489-020-01922-x>
- Ma'rifah, H., Wibawa, A. P., & Akbar, M. I. (2020). Klasifikasi Artikel Ilmiah Dengan Berbagai Skenario Preprocessing. *Sains, Aplikasi, Komputasi Dan Teknologi Informasi*, 2(2), 70. <https://doi.org/10.30872/jsakti.v2i2.2681>
- Parajuli, S., & Shakya, S. (2018). Malware Detection and Classification Using Latent Semantic Indexing. *Journal of Advanced College of Engineering and Management*, 4. <https://doi.org/10.3126/jacem.v4i0.23205>
- Sabrani, A., Wedashwara W., I. G. W., & Bimantoro, F. (2020). Multinomial Naïve Bayes untuk Klasifikasi Artikel Online tentang Gempa di Indonesia. *Jurnal Teknologi Informasi, Komputer, Dan Aplikasinya (JTika)*, 2(1), 89–100. <https://doi.org/10.29303/jtika.v2i1.87>
- Setyawan, C., Benarkah, N., & Prasetyo, V. R. (2021). Automatic Text Summarization Berdasarkan Pendekatan Statistika pada Dokumen Berbahasa Indonesia. *KELUWIH: Jurnal Sains Dan Teknologi*, 2(1).  
<https://doi.org/10.24123/saintek.v2i1.4045>
- Susanti, S., Azmi, M., Ali, E., Rahmaddeni, R., & Saputra Wijaya, Y. (2020). Perbandingan Boolean Model Dan Vector Space Model Dalam Pencarian Dokumen Teks. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 11(2), 268–277. <https://doi.org/10.31849/digitalzone.v11i2.4168>