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

**TUGAS KELOMPOK**

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

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

**Disusun Oleh :**

|  |  |
| --- | --- |
| **MUHAMMAD REZA** | **2019470055** |
| **SELAMET SAPUTRA** | **2019470069** |
| **SYECHAN AHMAD ZIDAN** | **2019470110** |

PROGRAM STUDI TEKNIK INFORMATIKAFAKULTAS TEKNIKUNIVERSITAS MUHAMMADIYAH JAKARTA2022

# **DAFTAR ISI**

[**DAFTAR ISI** ii](#_Toc121892662)

[**DAFTAR GAMBAR** iii](#_Toc121892663)

[**DAFTAR TABEL** iv](#_Toc121892664)

[**LAMPIRAN** v](#_Toc121892665)

[**BAB I** 1](#_Toc121892666)

[**1.1.** **Latar belakang masalah** 1](#_Toc121892667)

[**1.2.** **Identifikasi Masalah** 2](#_Toc121892668)

[**1.3.** **Rumusan Masalah** 2](#_Toc121892669)

[**1.4.** **Batasan Masalah** 3](#_Toc121892670)

[**BAB II** 4](#_Toc121892671)

[**2.1.** **Data Acquisition** 4](#_Toc121892672)

[**2.2** **Data Exploration** 5](#_Toc121892673)

[**2.3** ***Preprocessing*** 6](#_Toc121892674)

[**BAB III** 8](#_Toc121892675)

[**3.1.** ***Term Frequency Inverse Document Frequency*** 8](#_Toc121892676)

[**3.2.** ***Latent Semantic Indexing*** 10](#_Toc121892677)

[**3.3.** ***Vector Space Model*** 10](#_Toc121892678)

[**BAB IV** 12](#_Toc121892679)

[**Kesimpulan** 12](#_Toc121892680)

[**DAFTAR PUSTAKA** 13](#_Toc121892681)

# **DAFTAR GAMBAR**

[Gambar pengambilan data 4](#_Toc121892651)

[Gambar korpus 1 5](#_Toc121892652)

[Gambar korpus 2 5](file:///E:\COLLEGE\SEMESTER%207\TEMU%20KEMBALI%20INFORMASI\1%20TUGAS%20KELOMPOK\tugas-tki\LATENT%20SPACE%20INDEX\LAPORAN\EDLINK\Laporan_Kelompok_TKI.docx#_Toc121892653)

[Gambar perhitungan LSI 10](#_Toc121892654)

# **DAFTAR TABEL**

[Tabel 2.1 Perubahan teks 5](#_Toc121892644)

[Tabel 2.2 *stopwords* sering muncul 6](#_Toc121892645)

[Tabel 2.3 hasil preprocessing 7](#_Toc121892646)

[Tabel 3.1 *Term frequency* muncul di dokument 8](#_Toc121892647)

[Tabel 3.2 *Term frequency* tidakmuncul di dokument 8](#_Toc121892648)

[Tabel 3.3 tf-idf *query* 9](file:///E:\COLLEGE\SEMESTER%207\TEMU%20KEMBALI%20INFORMASI\1%20TUGAS%20KELOMPOK\tugas-tki\LATENT%20SPACE%20INDEX\LAPORAN\EDLINK\Laporan_Kelompok_TKI.docx#_Toc121892649)

[Tabel 3.4 Hasil *Cosine Smiliarity* 11](#_Toc121892650)

# **LAMPIRAN**

Kode : [tugas-tki/Tugas\_LATENT\_SPACE\_INDEX.ipynb at main · ackermanjayjay/tugas-tki (github.com)](https://github.com/ackermanjayjay/tugas-tki/blob/main/LATENT%20SPACE%20INDEX/code/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/tree/main/data/data%20twitter/data%20about%20ai)

# **BAB I**

## **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). I*nformation 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). ekstrasi 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 ekstrasi fitur tf-idf ini digunakan untunk perhitung 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 pencarianMatriks 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)

## **Identifikasi Masalah**

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

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

## **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* ?
3. Bagaimana cara menghitung jarak teks *document* twitter menggunakan *Cosine Smiliarity*  dan *Latent semantic indexing* ?

## **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* mengguanakan apify, korpus yang diambi berjumlah 84 dokument

Graphical user interface, text, application, email

Description automatically generated

Gambar pengambilan data

Text

Description automatically generated

Gambar korpus 1

Text

Description automatically generated

Gambar korpus 2

## **2.2 Data Exploration**

Karena dokument terpisah maka dilakukan penggabungan antara korpus 1 dan korpus 2, maka total data yang digabungkan berjulam 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 que incluyan los resultados de las pruebas estadísticas con rigor? statsplot lo vuelve simple y listo para publicar | Can you create spectacular graphics 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

|  |  |
| --- | --- |
| *Stopwords* | *Frequensi* |
| 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

1. *Stopword removal*

Adalah tahap untuk menghapus kata pengubung

1. *Stemming*

Adalah tahap untuk mereduksi kata menjadi kata dasar

1. *Word normalization*
2. 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 *preprocessing* | Sesudah *preprocessing* |
| I think the message in Data Science needs to be: Don't believe everything you read. 🧵  #stats #datascience https://t.co/4jGMgmX8Nw | think message data science needs believe everything read stats datascience jgmgmx nw |

Hasil dari preprocessing pada Tabel 2.3 dapat dilihat bahwa kata seperti “I” dihapus pada saat preprocessing, serta link seperti “<https://t.co> “ 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

|  |  |  |
| --- | --- | --- |
| *Term*(kata) | *Dokument frequency* | *Frequency in dokument* |
| data | 23 | 1 |
| text | 44 | 1 |
| mining | 27 | 1 |
| retrieval | 18 | 1 |
| python | 11 | 1 |

### Tabel 3.2 *Term frequency* tidakmuncul di dokument

|  |  |
| --- | --- |
| *Term*(kata) | Frequensi |
| data | 0 |
| text | 0 |
| mining | 0 |
| retrieval | 0 |
| python | 0 |

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

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

= 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:

*idf(term)\** FQ

*idf(data)\*FQ*

*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 | 3.111111111111111 | 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 :

Text

Description automatically generated

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:

= bobot tf-idf *term i* sampai *j*

= bobot tf-idf *term i* sampai *q*

### Tabel 3.4 Hasil *Cosine Smiliarity*

|  |  |  |
| --- | --- | --- |
| *Query* | 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 *casefiolding, stemming* menggunakan algoritma porter*, remove stopwords* menggunakan NLTK*, word normalization,* melakukan ekstraksi menggunakan tf-idf dengan *library scikit learn* agar dapat dilakukan kalkulasi menghitung reduksi singular dekomposisi serta dilakukan *vector space model*  menghasilkan dokumen paling terkuat adalah dokumen 8.

# **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