

Proposal Penelitian

PROPOSAL PERANCANGAN MEDIA MONITORING BAGI DAILYSOCIAL.ID

Proposal ini diajukan sebagai salah satu syarat untuk memenuhi Tugas Akhir Mata Kuliah
Pencarian Media Informasi Online



DIUSULKAN OLEH :

Sherly Santiadi	NIM: 2072025 / ANGKATAN: 2020
Grace Angelina Gunawan	NIM: 2072028 / ANGKATAN: 2020
Kathleen Felicia Annabel	NIM: 2072038 / ANGKATAN: 2020
Nisa Deviani Agustin Ruis	NIM: 2072051 / ANGKATAN: 2020

UNIVERSITAS KRISTEN MARANATHA
BANDUNG

2022

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa karena telah menolong kami dalam menyelesaikan Proposal Perancangan Media Monitoring Bagi DailySocial.id dengan tepat waktu. Dikarenakan tanpa bantuan daripada Tuhan Yang Maha Esa penulis tidak mungkin bisa untuk menyelesaikan Proposal Perancangan Media Monitoring Bagi DailySocial.id dengan baik.

Tidak lupa juga penulis mengucapkan terima kasih kepada:

1. Bapak Dr. Hapnes Toba, M.Sc., IPM selaku dosen pembimbing Mata Kuliah Pencarian Informasi Media Online.
2. Bapak Yohanes Adhi Nugraha selaku *Head of Engineering* di DailySocial.id.

Penulis berterima kasih atas dukungan materil dan moril sehingga penulis dapat menyelesaikan proposal ini dengan sebaik-baiknya. Penulis juga berterima kasih kepada seluruh rekan yang tidak dapat disebutkan satu per satu atas segala dukungan yang diberikan.

Penulis telah memaksimalkan segala usaha untuk tidak membuat kesalahan dalam menyusun proposal ini, akan tetapi penulis menyadari bahwa pada realitanya hal tersebut tidak dapat dihindari. Oleh karena itu, penulis berharap mendapatkan saran dan kritik yang membangun untuk penyempurnaan proposal selanjutnya. Penulis juga berharap proposal ini dapat bermanfaat bagi khalayak yang ingin menambah ilmu maupun kelompok-kelompok tertentu yang mungkin memiliki penelitian serupa.

Bandung, 28 Juni 2022

Penulis

PROPOSAL PERANCANGAN MEDIA MONITORING BAGI DAILYSOCIAL.ID

Sherly Santiadi¹⁾, Grace Angelina Gunawan²⁾ Kathleen Felicia Annabel³⁾ dan Nisa Deviani
Agustin Ruis⁴⁾

^{1, 2, 3, 4}Fakultas Teknologi Informasi, Universitas Kristen Maranatha

¹email: 2072025@maranatha.ac.id

Abstrak

Abstrak – Penyebaran informasi melalui media online sudah tidak dapat dihindari. Percepatan laju dari informasi yang mengalir di ruang sosial seperti sebuah arus yang begitu cepat dalam jumlah yang begitu banyak. Oleh karena itu, siapa saja dapat dengan leluasa mengakses informasi-informasi yang berada di media online, seperti Facebook, Twitter, Whatsapp, maupun berita yang didapatkan dari situs-situs tertentu. Informasi-informasi tersebut jika ditinjau lebih jauh sesungguhnya dienumerasikan berdasarkan *index* ataupun *rank* yang nantinya akan diurutkan dalam mesin pencarian seperti Google, Bing, Duck Duck Go, dan lain-lain. Apabila pengguna media online mengetikkan suatu kata kunci tertentu, tentu saja kata kunci-kata kunci yang muncul merupakan kata kunci yang sudah diurutkan berdasarkan *ranking* atau pencarian terbanyak dari seluruh pengguna media online. *Ranking* inilah yang sebenarnya merupakan konsep dasar dari *trending topics* yang akan dibahas pada proposal ini. Pada proposal ini bertujuan untuk menganalisis *trending topics* berdasarkan Tweet dari pengguna Twitter tertentu, situs-situs tertentu serta dataset internal yang telah disediakan oleh DailySocial.id. Kemudian dari hasil analisis tersebut akan dilakukan *forecasting* terkait topik-topik apa saja yang akan menjadi *hot topics* di masa yang akan datang. Dengan hasil *forecasting* tersebut diharapkan dapat dimanfaatkan sebagai referensi dalam penulisan artikel di situs DailySocial.id.

Kata kunci: *forecasting, clustering, crawling data*

DAFTAR ISI

KATA PENGANTAR	2
PROPOSAL PERANCANGAN MEDIA MONITORING BAGI DAILYSOCIAL.ID	3
DAFTAR ISI.....	4
DAFTAR GAMBAR	Error! Bookmark not defined.
DAFTAR TABEL.....	6
BAB I PENDAHULUAN.....	1
I.1 Latar Belakang	1
I.2 Rumusan Masalah	1
I.3 Tujuan Penelitian.....	1
I.4 Batasan Penelitian	2
BAB II KAJIAN LITERATUR	3
II.1 Penelitian Berlandaskan Twitter API.....	3
II.2 Konklusi Penelitian Berlandaskan Twitter API	3
II.3 Term Frequency-Inverse Document Frequency (TF-IDF).....	3
II.4 Konklusi Term Frequency-Inverse Document Frequency (TF-IDF).....	4
II.5 Latent Dirichlet Allocation.....	4
II.6 Konklusi Latent Dirichlet Allocation.....	4
II.7 Penelitian Berlandaskan <i>Website Crawling</i>	5
II.8 Konklusi Penelitian Berlandaskan <i>Website Crawling</i>	5
II.9 Beautiful Soup.....	5
II.10 Konklusi Beautiful Soup	5
BAB III ANALISIS DAN DESAIN.....	6
III.1 Analisis Kebutuhan Sistem	6
III.2 Analisis Langkah Kerja dan Riset.....	6
III.3 Flow Chart.....	7
III.4 Unified Modelling Language (UML).....	7
III.5 Entity Relationship Diagram (ERD)	8
III.6 Graphical User Interface (GUI).....	8
BAB IV IMPLEMENTASI	10
IV.1 DATASET TWITTER	10
IV.2 DATASET WEBSITE	20
IV.3 DATASET INTERNAL.....	31

BAB V UJI COBA SEDERHANA	38
V.1 DATASET TWITTER	38
V.2 DATASET WEBSITE	39
V.3 DATASET INTERNAL.....	41
BAB VI KESIMPULAN DAN SARAN	43
VI.1 KESIMPULAN	43
VI.2 SARAN	43
LAMPIRAN JADWAL Pengerjaan	44
LAMPIRAN KODE DAN VIDEO	49
DAFTAR PUSTAKA	50

DAFTAR GAMBAR

GAMBAR 1 FLOW CHART CRAWLING DATA DARI TWITTER API.....	7
GAMBAR 2 UML DARI TWITTER API	7
GAMBAR 3 ERD DARI TWITTER API	8
GAMBAR 4 GUI DASHBOARD DARI TWITTER DAN WEBSITE.....	8
GAMBAR 5 GUI FORECASTING DARI TWITTER DAN WEBSITE	9
GAMBAR 6 IMPLEMENTASI KODE FETCHING DATA DARI TWITTER API	10
GAMBAR 7 CENTER NODE MENJADI <i>HUB</i> BAGI NODE LAINNYA	11
GAMBAR 8 HASIL FETCHING DATA DARI TWITTER API.....	12
GAMBAR 9 IMPLEMENTASI KODE UNTUK TF-IDF.....	13
GAMBAR 10 PREPROCESSING DATASET	14
GAMBAR 11 FREKUENSI TWEET	15
GAMBAR 12 DISTRIBUSI TWEET	16
GAMBAR 13 WORD CLOUD	16
GAMBAR 14 PyLDAvis.....	17
GAMBAR 15 5 DIAGRAM TRENDING TOPIK.....	17
GAMBAR 16 HASIL FORECASTING.....	19
GAMBAR 17 IMPLEMENTASI KODE UNTUK WEB CRAWLER.....	20
GAMBAR 18 HASIL CRAWLING DARI WEBSITE.....	20
GAMBAR 19 DATA CSV UNTUK WEBSITE GIZMOLOGI	21
GAMBAR 20 DATA CSV YANG DIAMBIL DARI WEBSITE KUMPARAN	21
GAMBAR 21 BEBERAPA METODE PREPROCESSING WEBSITE	22
GAMBAR 22 HASIL TERM FREQUENCY WEBSITE.....	23
GAMBAR 23 HASIL TF-IDF UNTUK DATA WEBSITE	24
GAMBAR 24 HASIL WORD CLOUD WEBSITE KUMPARAN	25
GAMBAR 25 HASIL WORD CLOUD WEBSITE GIZMOLOGI	25
GAMBAR 26 PEMBUATAN LDA MODEL UNTUK WEBSITE KUMPARAN	26
GAMBAR 27 VISUALISASI PYLDAVIS UNTUK WEBSITE KUMPARAN	26
GAMBAR 28 DIAGRAM TOP 5 TOPIC WEBSITE KUMPARAN	27
GAMBAR 29 DIAGRAM TOP 5 TOPIC WEBSITE GIZMOLOGI.....	28
GAMBAR 30 DIAGRAM TOP 5 TOPIC WEBSITE IDNTIMES	29
GAMBAR 31 FLOW CHART CRAWLING DATA DARI WEBSITE	30
GAMBAR 32 UML DARI WEBSITE	30
GAMBAR 33 ERD DARI WEBSITE.....	31
GAMBAR 34 HASIL PENGAMBILAN DATASET INTERNAL DARI API DAILYSOCIAL.ID	31
GAMBAR 35 PREPROCESSING DATASET INTERNAL	32
GAMBAR 36 IMPLEMENTASI KODE UNTUK TF-IDF.....	33
GAMBAR 37 WORD CLOUD	34
GAMBAR 38 IMPLEMENTASI KODE UNTUK BAG OF WORDS	34
GAMBAR 39 IMPLEMENTASI KODE UNTUK LDA	35
GAMBAR 40 PyLDAvis.....	36
GAMBAR 41 BAR DIAGRAM.....	36
GAMBAR 42 TESTING MODEL DENGAN DOKUMEN BARU.....	38
GAMBAR 43 TESTING CLUSTERING BERDASARKAN KEDEKATAN INTERTOPIC	38

GAMBAR 44 HASIL FORECASTING.....	39
GAMBAR 45 TESTING MODEL DENGAN DATA BARU	39
GAMBAR 46 TESTING DENGAN PYLDAVIS.....	40
GAMBAR 47 TESTING MODEL DENGAN DOKUMEN BARU.....	41
GAMBAR 48 TESTING CLUSTERING BERDASARKAN KEDEKATAN INTERTOPIC	42

DAFTAR TABEL

TABEL 1 JADWAL TWITTER SEBELUM UTS	44
TABEL 2 JADWAL TWITTER SESUDAH UTS.....	45
TABEL 3 JADWAL WEB CRAWLER SEBELUM UTS.....	46
TABEL 4 JADWAL WEB CRAWLER SESUDAH UTS.....	47
TABEL 5 JADWAL DATASET INTERNAL SESUDAH UTS	48

BAB I

PENDAHULUAN

I.1 Latar Belakang

Media sosial semakin hari semakin banyak digunakan oleh berbagai individu. Dengan hadirnya media sosial maka akan memudahkan kita untuk berinteraksi dengan siapa pun dan di mana pun kita berada. Selain daripada itu, media sosial juga sudah sebagaimana pada umumnya digunakan untuk memediasi pertukaran informasi. Informasi tersebut dapat dibaca oleh pengguna media sosial melalui berbagai *platform* misalnya Twitter, Website, dan lain sebagainya. Oleh karena itu, kumpulan informasi yang tersebar di jejaring sosial diharapkan bisa dimanfaatkan untuk mencari kata-kata apa saja yang sering dibicarakan di jejaring sosial. Tentunya, dengan kata-kata tersebut dapat diolah menjadi sebuah kumpulan data yang nantinya dapat digunakan untuk memprediksi pada jangka waktu tertentu topik apa yang kira-kira akan diminati oleh pengguna. Hal ini tentu saja akan berguna bagi perusahaan DailySocial.id dalam menjadikan sebuah *platform* yang dapat menyajikan artikel-artikel yang terpercaya dan terakurat.

I.2 Rumusan Masalah

1. Bagaimana cara membangun *dataset* dengan metode *crawling* API Twitter dan website tertentu?
2. Bagaimana cara melakukan *forecasting* secara akurat dengan teknik *unsupervised learning*?
3. Bagaimana perbandingan keakuratan hasil *forecasting* melalui dataset yang dibangun melalui metode *crawling* dengan dataset internal yang telah disediakan oleh Daily Social?

I.3 Tujuan Penelitian

1. Untuk mengetahui cara membangun *dataset* dengan metode *crawling* API Twitter dan website tertentu.

2. Untuk mengetahui cara melakukan *forecasting* secara akurat dengan teknik *unsupervised learning*.
3. Untuk menguji perbandingan keakuratan hasil *forecasting* melalui dataset yang dibangun melalui metode *crawling* dengan dataset internal yang telah disediakan oleh Daily Social.

I.4 Batasan Penelitian

Batasan-batasan masalah pada penyusunan proposal ini adalah sebagai berikut :

1. *Tweet* yang diambil berbahasa Indonesia dari Izak Jennie, Budi Raharjo, dan juga Norman Sasono.
2. *Website* yang akan peneliti *crawling* diambil dari *kumparan.com*, *gizmologi.id*, dan *idntimes.id*.
3. Bahasa pemrograman yang digunakan dalam *fetching data Twitter* dan *website crawling* adalah *Python*.

BAB II

KAJIAN LITERATUR

II.1 Penelitian Berlandaskan Twitter API

Twitter menyediakan sebuah API yang bisa diakses pengguna untuk melakukan pencarian (*searching*) ataupun mengambil (*retrieving*) data yang nantinya data-data tersebut akan dijadikan sebagai dataset. Oleh karena itu, peneliti perlu membuat sebuah akun Twitter Developer agar bisa mengakses API tersebut. Setelah akun Twitter Developer terverifikasi, maka peneliti akan memiliki *Keys and Access Tokens*. *Keys and Access Tokens* tersebut perlu disimpan baik-baik sehingga tidak ada pengguna lain yang dapat menggunakan token tersebut. Peneliti juga dapat menekan tombol *generate* untuk membuat token baru jika diperlukan.

II.2 Konklusi Penelitian Berlandaskan Twitter API

Salah satu referensi yang digunakan ialah penelitian terkait *Twitter Sentiment Analysis* untuk mengklasifikasikan ulasan-ulasan film menggunakan pendekatan *Naive Bayes Classifier* dan *Support Vector Machine* yang dilakukan oleh Akshay Amonik, Niketan Jivane, Mahavir Bhadari, Dr. M. Venkatesan pada tahun 2015 (Rahutomo, Saputra and Fidyawan, 2018). Hal inilah yang dijadikan sebagai referensi bahwa peneliti dapat menggunakan Twitter API dalam pembangunan dataset untuk *media monitoring* apabila DailySocial.id tidak menyediakan dataset secara internal.

II.3 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency atau yang lebih dikenal sebagai TF-IDF merupakan sebuah algoritma yang akan menghitung bobot/*weight* pada setiap kata (token) yang tercantum pada sebuah dokumen (korpus). Algoritma ini berguna untuk mengetahui jumlah kata muncul dalam sebuah dokumen. Berdasarkan penelitian yang dilakukan oleh Zhi Liang Zhu, Jie Liang, De Yang Li, Hai Yu, dan Guo Qi Liu pada jurnal yang berjudul “Hot Topic Detection Based on a Refined TF-IDF Algorithm” (Zhu *et al.*, 2019), TF-IDF merupakan algoritma yang tepat untuk mencari *hot term* berdasarkan distribusi waktu tertentu. Pada jurnal tersebut dijelaskan setidaknya ada empat alasan utama dalam pemilihan algoritma TF-IDF. Pertama, banyak *hot term* merupakan *term* yang baru. Kedua, algoritma TF-IDF umumnya digunakan untuk pemilihan fitur teks (*text feature selection*). Ketiga, belum ada metode lain yang layak untuk digunakan dalam mencari *hot term* pada artikel berita. Keempat, apabila

menggunakan teknik deteksi *hot term* tradisional yaitu secara langsung menggunakan algoritma *clustering* dapat berkinerja buruk karena algoritma tersebut akan rentan terhadap *outlier*. Selain itu juga, banyaknya jumlah *non-hot term* dapat mengarah kepada banyaknya *outlier* pada proses *clustering*.

II.4 Konklusi Term Frequency-Inverse Document Frequency (TF-IDF)

Konklusi dari referensi dari artikel terkait TF-IDF adalah TF-IDF dapat digunakan untuk mencari *hot term* pada dokumen atau dalam hal ini berarti berkaitan langsung dengan dataset yang ada pada Twitter. Untuk pengimplementasiannya peneliti dapat mengimpor langsung dari *library Scikit-Learn*. Kemudian masukan dataset yang sudah dikumpulkan dan dilakukan *preprocessing* ke dalam bentuk *array*. Setelah itu pada *vectorizer* dapat digunakan atribut *stop_words* untuk menghapus kata-kata tidak berarti, seperti kata “yang”, “di”, “ke”, dan sebagainya. Adapun penggunaan *stemming* perlu diberlakukan pada algoritma TF-IDF yaitu proses pemetaan kata-kata berimbuhan menjadi kata dasar misalkan seperti kata “menulis” menjadi “tulis”.

II.5 Latent Dirichlet Allocation

Latent Dirichlet Allocation atau kerap dikenal sebagai LDA adalah salah satu pendekatan untuk mendeteksi topik-topik pada dokumen berdasarkan proporsi kemunculan topik tersebut. Ide dasar dari LDA sendiri yaitu meyakini bahwa sebuah dokumen pasti terdiri dari beberapa topik, oleh karena itu dengan model statistik kumpulan dari suatu dokumen akan direpresentasikan dalam sebuah *imaginary random process*. Kemudian setiap topik tersebut akan terdiri dari berbagai distribusi kata-kata.

II.6 Konklusi Latent Dirichlet Allocation

Pada penelitian ini, digunakan pendekatan Latent Dirichlet Allocation, karena ketika kita berhadapan dengan suatu data, misalkan data Twitter tentu saja satu pengguna bisa membuat *tweet* dalam kurun waktu terdekat misalkan 1 menit. Dalam waktu yang relatif singkat tersebut tentu saja berbagai macam *tweet* akan istilahnya bertumpuk menjadi satu apabila tidak ada pemisahan. Oleh karena itu untuk mengetahui distribusi topik-topik tersembunyi pada sekumpulan data misalkan Twitter, kita bisa menggunakan pendekatan LDA untuk membuat inferensi dari apa yang diperbincangkan dari kumpulan *tweet* tersebut.

II.7 Penelitian Berlandaskan Website Crawling

Dalam halaman suatu *website*, peneliti dapat melakukan *inspect* untuk melihat keseluruhan kode yang membentuk *website* tersebut. Kode tersebut dapat diambil kemudian diproses sehingga didapatkan seluruh *URL* yang terdapat dalam *website* dan nantinya akan dijadikan sebagai *dataset*. Proses ini dilakukan dengan menggunakan *python* yang akan dijalankan di *Jupyter Notebook*.

II.8 Konklusi Penelitian Berlandaskan Website Crawling

Dari *URL* yang telah diambil, *crawler* akan mengunduh semua halaman *web* yang dialamatkan dari *URL* tersebut sehingga nantinya dapat peneliti gunakan untuk mencari kata-kata yang sedang *trending*.

II.9 BeautifulSoup

Beautiful Soup adalah sebuah *database Python* berdasarkan mesin analitik *HTML/XML* yang digunakan untuk mengekstrak, menganalisis, dan mengedit informasi dari halaman *web*. Pengguna *Beautiful Soup* dapat menginstall mesin analitik *HTML/XML* tertentu. *Beautiful Soup* akan menganalisis dokumen apapun yang diberikan namun hanya dokumen *HTML* dan *XML* yang memenuhi spesifikasi yang akan dikonversi menjadi *DOM tree* lalu pengguna dapat menggunakan fungsi-fungsi *Beautiful Soup* untuk mengoperasikan sebuah *DOM tree*.

II.10 Konklusi BeautifulSoup

Peneliti menggunakan *Beautiful Soup* untuk mengekstrak seluruh *URL* yang terdapat pada halaman *website* kompetitor dan nantinya hasil ekstraksi akan digunakan sebagai *dataset* untuk melihat kata-kata yang sedang *trending*. Untuk pengimplementasiannya peneliti dapat mengimpor *BeautifulSoup* langsung dari *library bs4*. Kemudian masukkan *URL website* yang akan peneliti *crawling* ke dalam fungsi yang telah diimpor. Gunakan *looping* untuk mendapatkan *URL* yang terdapat pada *website*.

BAB III

ANALISIS DAN DESAIN

III.1 Analisis Kebutuhan Sistem

Berdasarkan hasil penelitian yang telah dilakukan setidaknya ada 3 permasalahan utama yang bisa dijadikan sebagai landasan dalam menganalisis kebutuhan sistem:

- Perlunya sebuah sistem yang dapat mengestimasi topik yang akan populer di masa depan.
- Perlunya sebuah sistem yang dapat melakukan sumerisasi terhadap topik yang akan trending. Misalkan topik Metaverse akan terjadi peningkatan selama 3-4 hari ke depan.
- Perlunya sebuah sistem yang dapat mengimbangi atau bahkan melampaui kemampuan kompetitor yang ada.

III.2 Analisis Langkah Kerja dan Riset

Untuk analisis langkah kerja dan riset, ada beberapa hal yang dijadikan sebagai bahan pertimbangan yaitu untuk media monitoring direkomendasikan menggunakan Python 3.6.x - 2.8.x. Sedangkan untuk rekomendasi databasenya sendiri adalah PostgreSQL ≥ 11.0 .

Selanjutnya ada beberapa langkah kerja yang dipertimbangkan dalam melakukan media monitoring berdasarkan pendekatan teknik unsupervised:

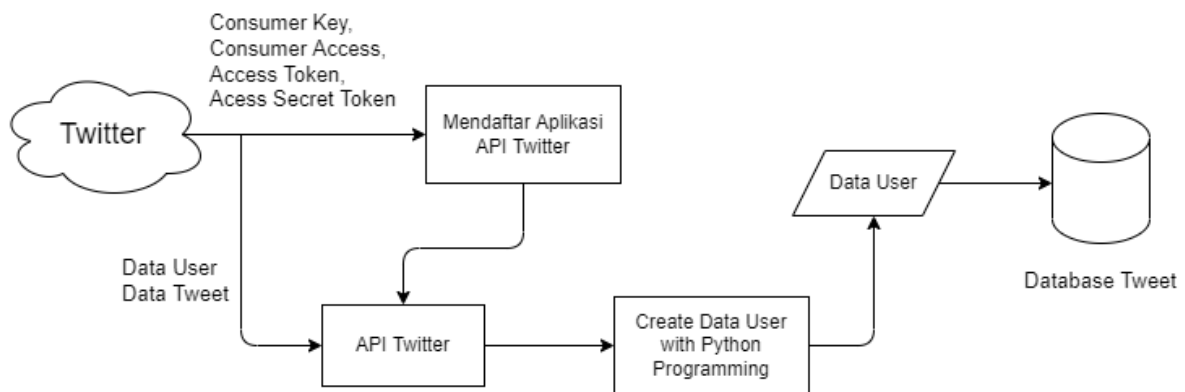
- Pembangunan Dataset

Pembangunan dataset ini dilakukan dengan cara mengumpulkan fitur-fitur penting yang dapat digunakan dalam memprediksi topik. Pembangunan dataset ini hanya berguna bagi dataset Twitter dan Website saja. Sedangkan untuk dataset internal sudah diberikan dari DailySocial.id.

- Relational Topic Modeling

Relational Topic Modeling sendiri merupakan cara *unsupervised* dalam teknik pembentukan kluster. Ada beberapa pendekatan yang nantinya digunakan yaitu TF-IDF, LDA sekaligus klasterisasi, setelah itu barulah dilakukan *forecasting*.

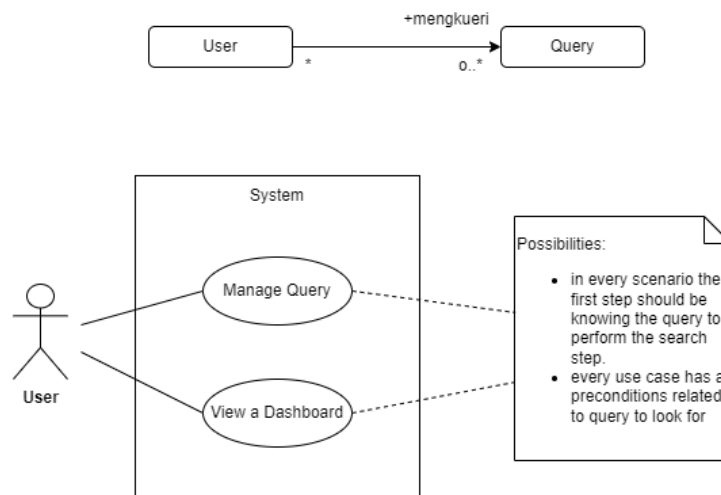
III.3 Flow Chart



GAMBAR 1 FLOW CHART CRAWLING DATA DARI TWITTER API

Proses crawling data dari Twitter API diawali dengan melakukan login pada akun Twitter. Setelah melakukan login kita harus mendaftar aplikasi untuk mendapatkan access tokens berupa consumer key, consumer access, access token dan access secret token. Setelah mendapat kode akses tadi, kita mendapatkan API twitter. Untuk bisa melakukan komunikasi antara token Access dan Twitter API, kita membuat script dengan Python Programming sebagai media untuk crawling. Script ini dapat melakukan pencarian data sesuai dengan query yang diinginkan. Ketika data user telah diperoleh maka data-data tersebut disimpan ke dalam database tweet.

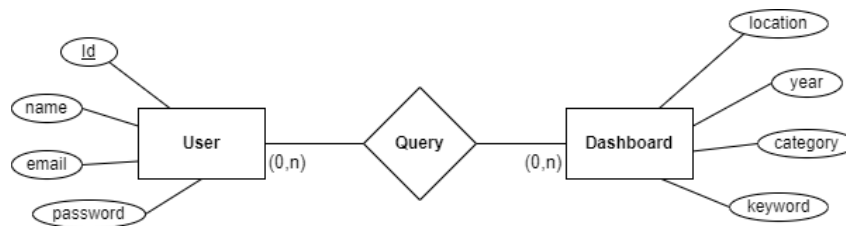
III.4 Unified Modelling Language (UML)



GAMBAR 2 UML DARI TWITTER API

Pada diagram di atas ditunjukkan bahwa user manage query dan dapat melihat dashboard di dalam sistem. Kemungkinannya di dalam setiap skenario sistem, langkah pertama yaitu sistem harus mengetahui query untuk melakukan langkah pencarian dari setiap kasus penggunaan yang memiliki prasyarat terkait dengan query yang harus dicari.

III.5 Entity Relationship Diagram (ERD)

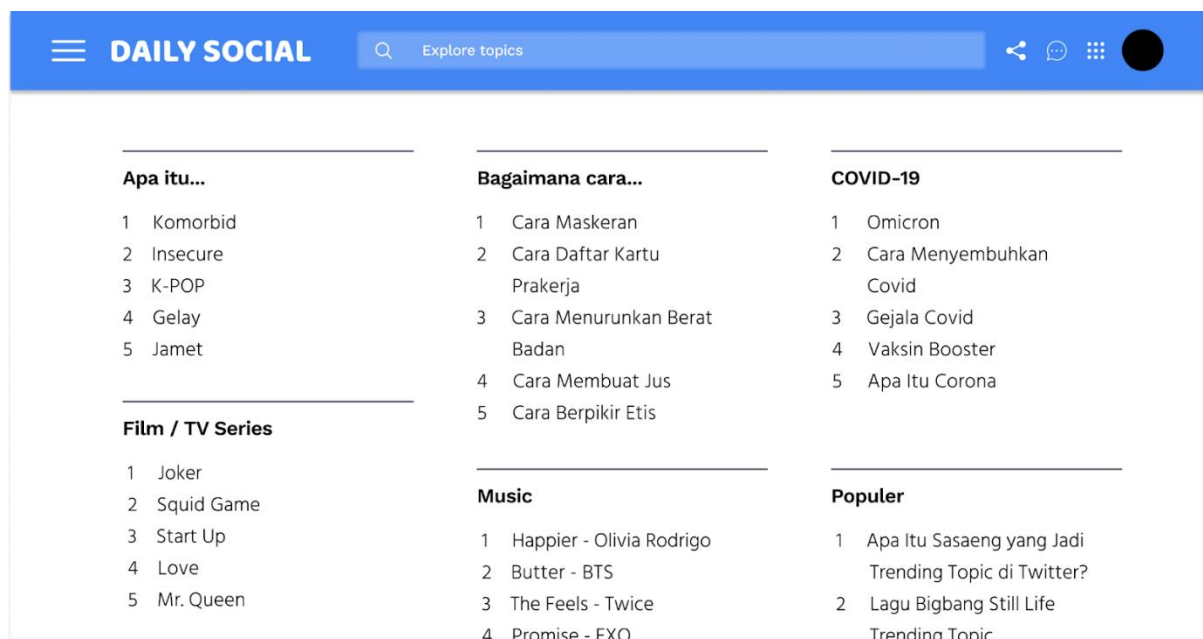


GAMBAR 3 ERD DARI TWITTER API

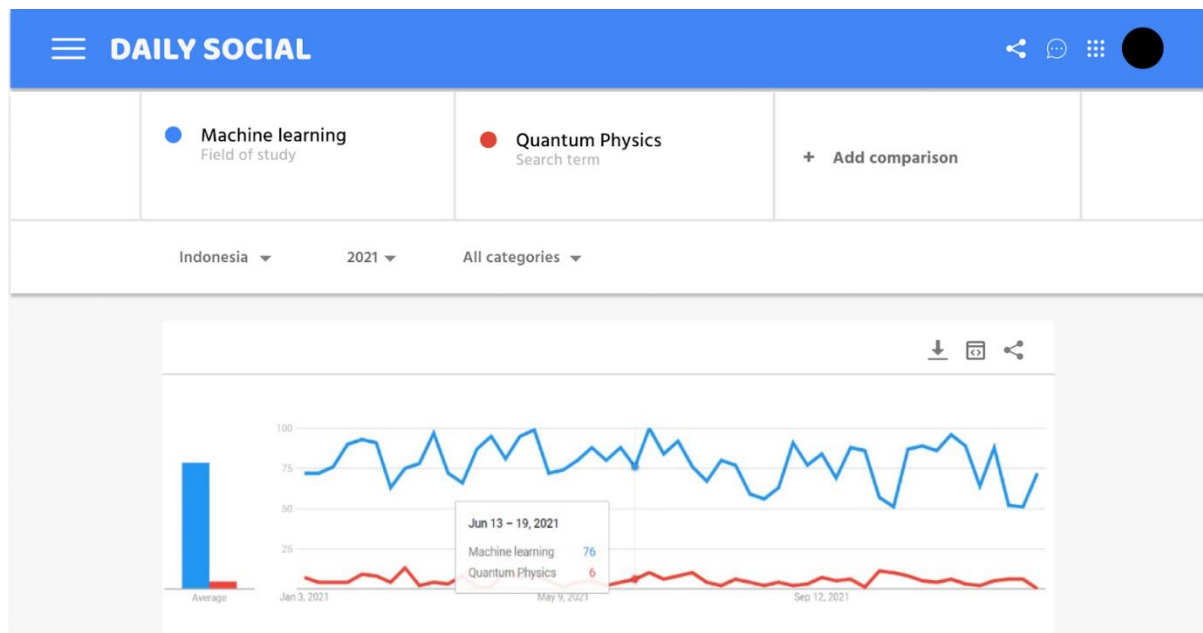
Gambar ERD di atas memiliki dua entitas yaitu entitas user dan dashboard. Entitas user ini memiliki atribut id, name, email dan password. Entitas user ini mengkueri entitas dashboard yang dimana entitas ini memiliki atribut location, year, category dan keyword.

III.6 Graphical User Interface (GUI)

Berikut adalah contoh gambar interface dari sistem. Di dalam gambar ini, user mendapatkan hak akses untuk melakukan pencarian query. Lalu user dapat melihat data dengan memfilter data berdasarkan dari lokasi, tahun dan berdasarkan kategorinya.



GAMBAR 4 GUI DASHBOARD DARI TWITTER DAN WEBSITE



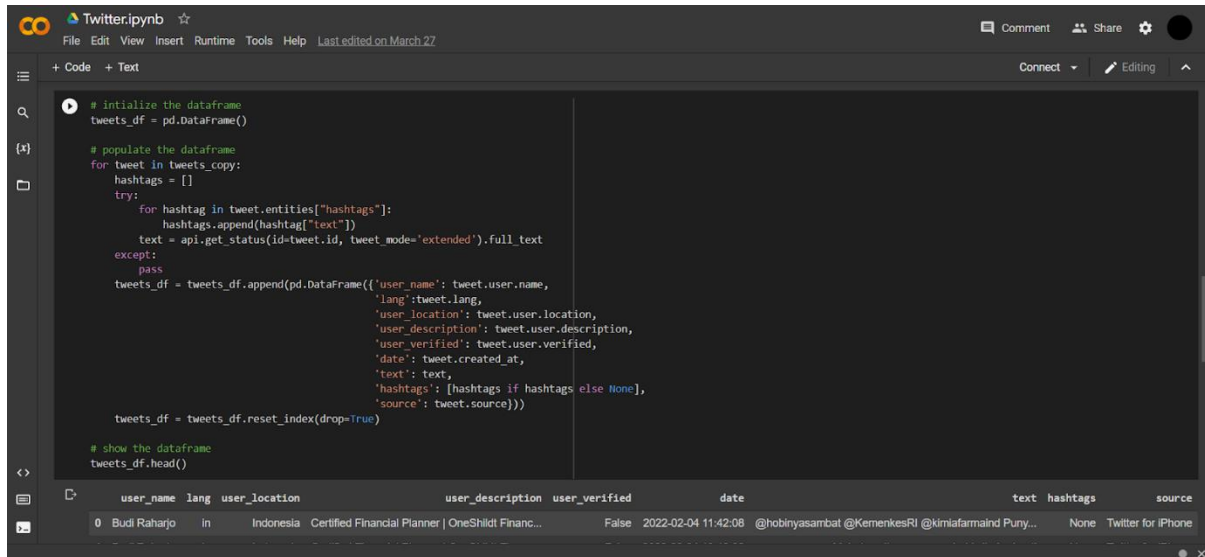
GAMBAR 5 GUI FORECASTING DARI TWITTER DAN WEBSITE

BAB IV

IMPLEMENTASI

IV.1 DATASET TWITTER

A. Contoh Implementasi Kode untuk Mengambil Data dari Twitter API



```
# initialize the dataframe
tweets_df = pd.DataFrame()

# populate the dataframe
for tweet in tweets_copy:
    hashtags = []
    try:
        for hashtag in tweet.entities["hashtags"]:
            hashtags.append(hashtag["text"])
        text = api.get_status(id=tweet.id, tweet_mode='extended').full_text
    except:
        pass
    tweets_df = tweets_df.append(pd.DataFrame({'user_name': tweet.user.name,
                                              'lang': tweet.lang,
                                              'user_location': tweet.user.location,
                                              'user_description': tweet.user.description,
                                              'user_verified': tweet.user.verified,
                                              'date': tweet.created_at,
                                              'text': text,
                                              'hashtags': [hashtags if hashtags else None],
                                              'source': tweet.source}))

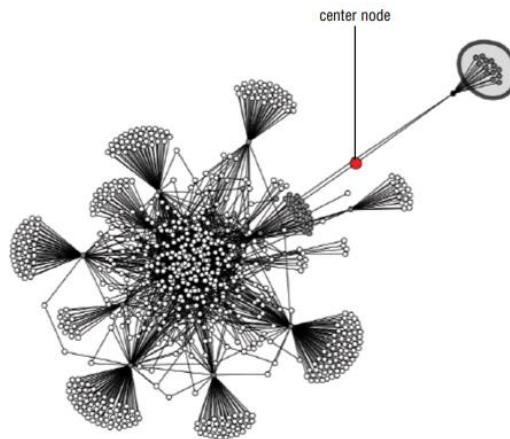
tweets_df = tweets_df.reset_index(drop=True)

# show the dataframe
tweets_df.head()
```

	user_name	lang	user_location	user_description	user_verified	date	text	hashtags	source
0	Budi Raharjo	in	Indonesia	Certified Financial Planner OneShildt Financ...	False	2022-02-04 11:42:08	@hobinyasambat @KemenkesRI @kimiafarmaind Puny...	None	Twitter for iPhone

GAMBAR 6 IMPLEMENTASI KODE FETCHING DATA DARI TWITTER API

Hal yang pertama kali perlu dilakukan dalam membangun sebuah algoritma *Machine Learning* adalah dataset. Dataset yang digunakan dapat berupa dataset internal yaitu kumpulan data yang sudah diberikan dari pihak DailySocial.id atau dapat juga berupa dataset yang dikumpulkan dari berbagai media sosial. Pada penelitian ini dataset yang dibangun berasal dari API Twitter. Setelah memiliki akun Twitter Developer, *Keys and Access Tokens* dapat langsung digunakan tidak hanya untuk melakukan *tweet*, *retweet*, tetapi juga *Keys and Access Tokens* tersebut dapat digunakan untuk melakukan *fetching data* yang ada pada Twitter. Bahasa pemrograman yang digunakan untuk melakukan *fetching data* pun beragam akan tetapi pada penelitian ini bahasa pemrograman yang digunakan adalah bahasa *Python*.



GAMBAR 7 CENTER NODE MENJADI *HUB* BAGI NODE LAINNYA

Selain itu untuk mengambil data-data yang ada di Twitter terdapat berbagai atribut yang bisa dicantumkan dalam implementasi kode, untuk penelitian ini atribut yang digunakan adalah *user_name*, *lang*, *user_location*, *user_description*, *user_verified*, *date*, *text*, *hashtags*, *source*. Pada gambar 2 hal yang ingin digambarkan adalah dikarenakan pada sebuah media sosial tertentu tidak semua pengguna-pengguna tersebut adalah pengguna aktif. Definisi pengguna aktif di sini adalah pengguna-pengguna yang secara aktif atau dengan interval waktu yang pendek akan menggunakan fasilitas yang diberikan media sosial misalnya saja melakukan *tweet*, *retweet*, *like*, *follow*, dan sebagainya. Oleh karena itu, salah satu strategi yang perlu digunakan adalah kita perlu mengambil secara acak seorang pengguna yang dapat dijadikan sebagai penghubung bagi pengguna-pengguna lainnya. Sehingga tidak perlu mengambil semua pengguna menjadi sample dalam mencari *trending topics* akan tetapi cukup untuk mengambil node-node yang dianggap representatif dalam pembuatan keputusan dalam hal ini keputusan dalam memprediksi *trending topics*.

B. Contoh Hasil Pengambilan Data dari Twitter API

	user_name	lang	user_location	user_description	user_verified	date	text	hashtags	source
0	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-02-04 11:42:06	@hobinyasambat @KemenkesRI @kmliafa		Twitter for iPhone
1	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-02-04 10:19:08	Mak deg gitu gempanya habis itu berhenti		Twitter for iPhone
							HATI HATI		
							Bisa pusing kalau		
							Gaji kecil kurang..		
2	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-01-13 3:15:14	Gaji besar ga cukup		Twitter for iPhone
							Ada yang lebih serem daripada ujian CFP		
3	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-01-06 2:01:48	Karena ujian CFP ada persiapannya, kalau		Twitter for iPhone
4	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-01-05 5:04:28	Supaya tidak malu, cuma mau ngingetin..		Twitter for iPhone
5	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-01-02 11:26:17	Interview Live bareng @OfficialNewsTV s		Twitter for iPhone
6	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2022-01-01 3:15:23	Terima kasih 2021. Selamat datang 2022.		Twitter for iPhone
							Nemu kata-kata paling menggoda di akhir		
7	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-12-31 4:02:40	Beli sekarang, Bayar nanti		Twitter for iPhone
8	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-12-15 4:34:52	RT @cnbcindonesia: The Fed Bersiap Na		Twitter for Android
9	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-11-26 2:34:53	@feynance pake jalan kaki ke kantor pos g		Twitter Web App
10	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-11-26 1:34:22	Pagii.. yang dulu punya rekening Tabana:		Twitter for iPhone
11	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-11-25 8:17:22	Perkiraan cuacanya tepat		Twitter for iPhone
12	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-11-07 2:34:57	Sedih kalau lihat berita ada orang terjeba		Twitter for iPhone
13	Budi Raharjo	in	Indonesia	Certified Financial P	FALSE	2021-10-20 13:56:52	Jadi Generasi Sandwich dan memutuskan ma		Twitter for iPhone
							Free Seminar : KNOW YOUR NUMBERS, I		
							23 Oktober 2021 Jam 11.00 - 12.00 WIB		
14	Budi Raharjo	en	Indonesia	Certified Financial P	FALSE	2021-10-19 6:43:31	Registrasi https://t.co/fkISINPVs https://t.co/fkISINPVs		Twitter for iPhone

GAMBAR 8 HASIL FETCHING DATA DARI TWITTER API

Berdasarkan hasil diskusi dengan pihak DailySocial.id maka terdapat beberapa tokoh publik yang dianggap representatif dalam pengambilan keputusan diantaranya adalah Izak Jennie, Budi Raharjo, dan juga Norman Sasono. Pada gambar 8 peneliti memulai *fetching data* yang diambil dari akun Budi Raharjo (@raharjobudi). Adapun beberapa keuntungan dengan membangun data berlandaskan Twitter API diantaranya adalah:

- Peneliti dapat mengontrol *query* yang diinginkan dalam pengambilan data semisalnya saja *query* yang diambil bisa seluruh cuitan dari akun Twitter Budi Raharjo ataupun dengan kata kunci tertentu seperti ‘*metaverse*’ misalnya.

Akan tetapi, adapula kekurangan dalam pengambilan data dari Twitter API secara gratis diantaranya adalah:

- Adanya batasan dalam *fetching data* yaitu 2500 *tweet* untuk 1 kali *fetch*.
- Perlunya waktu yang lama dalam mengkonversi data ke dalam bentuk *Data Frame* dalam hal ini untuk 2500 *tweet* dibutuhkan waktu sekitar 40 menit (Pada penelitian ini digunakan Processor AMD Ryzen 3 3250U with Radeon Graphics, 2600 Mhz, 2 Core(s), 4 Logical Processor(s)).
- *Query* tidak bisa dikustomisasi untuk waktu tertentu misalnya saja peneliti ingin mengambil data dari tanggal ‘01-01-2020’ sampai dengan ‘01-06-2021’, hal ini tidak bisa diterapkan pada Twitter API secara gratis.

C. Contoh Implementasi Kode untuk Penerapan Algoritma TF-IDF

```
Score: 0.6983450651168823
Topic: 0.010*"wordl" + 0.004*"enhypenmemb" + 0.004*"enhypen" + 0.003*"look" + 0.003*"tweet" + 0.003*"heeseung" + 0.002*"level"

Score: 0.15721043944358826
Topic: 0.004*"cav" + 0.004*"happi" + 0.003*"birthday" + 0.003*"hope" + 0.003*"final" + 0.002*"amaz" + 0.002*"peopl" + 0.002*"s

Score: 0.011111128143966198
Topic: 0.003*"love" + 0.003*"manifest" + 0.003*"nigga" + 0.002*"crazi" + 0.002*"peopl" + 0.002*"team" + 0.002*"safe" + 0.002*"

Score: 0.011111123487353325
Topic: 0.002*"april" + 0.002*"go" + 0.002*"congratul" + 0.002*"fool" + 0.002*"strong" + 0.001*"mama" + 0.001*"excit" + 0.001*"

Score: 0.01111117899417877
Topic: 0.004*"cute" + 0.003*"sorri" + 0.002*"march" + 0.002*"guy" + 0.002*"that" + 0.002*"readi" + 0.002*"tire" + 0.002*"home"

Score: 0.01111116036772728
Topic: 0.008*"btstwt" + 0.004*"listen" + 0.003*"babi" + 0.003*"funni" + 0.003*"jungkook" + 0.002*"gonna" + 0.002*"btsbutter" +

Score: 0.01111113242805004
Topic: 0.003*"cool" + 0.002*"wanna" + 0.002*"fact" + 0.002*"video" + 0.002*"battl" + 0.002*"bore" + 0.002*"enjoy" + 0.002*"goo

Score: 0.0111111044883728
Topic: 0.004*"thank" + 0.002*"friend" + 0.002*"hello" + 0.002*"year" + 0.002*"absolut" + 0.002*"mood" + 0.002*"go" + 0.001*"st

Score: 0.0111111044883728
Topic: 0.003*"post" + 0.003*"photo" + 0.003*"cri" + 0.002*"pretti" + 0.002*"say" + 0.002*"lose" + 0.002*"exact" + 0.002*"shut"
```

GAMBAR 9 IMPLEMENTASI KODE UNTUK TF-IDF

Gambar 9 merupakan contoh implementasi algoritma TF-IDF, di mana pada sekumpulan topik tertentu akan diberikan bobot sesuai dengan kemunculan katanya. Misalkan, kita ambil contoh kata “btstwt” diberikan bobot sebesar 0.008 yang artinya lebih banyak kemunculannya dibandingkan kata “listen” yang hanya diberi bobot sebesar 0.004. Kemudian topik-topik tersebut diurutkan secara *ascending* di mana score TF-IDF paling tinggi berada pada urutan paling atas. Nilai skor di sini menunjukkan bahwa seberapa sering *keyword* tersebut muncul di sebuah dokumen yang akhirnya mengalokasikan bahwa *keyword* tersebut layak/penting berdasarkan kemunculan kata dalam dokumen. Semakin tinggi TF-IDF score maka semakin penting atau relevan term tersebut. Apabila skor TF-IDF mendekati nilai nol berarti term tersebut dapat dianggap kurang relevan.

D. Preprocessing Dataset

```
Expand Contraction

[ ] contractions_dict = {"ain't": "are not", "is'": "is", "aren't": "are not", "don't": "do not"}
# Regular expression for finding contractions
contractions_re=re.compile("(%s) %s" % '|'.join(contractions_dict.keys()))
def expand_contractions(text,contractions_dict=contractions_dict):
    def replace(match):
        return contractions_dict[match.group(0)]
    return contractions_re.sub(replace, text)

time: 5.67 ms (started: 2022-05-21 15:40:48 +00:00)

Remove Duplicates Value

[ ] def remove_duplicates(df):
    # dropping all duplicates value
    df.drop_duplicates(subset = "text",
                      keep = False, inplace = True)
    return df

time: 2.65 ms (started: 2022-05-21 15:40:48 +00:00)

Remove 2 Words Tweets

[ ] def remove_2_words_tweet(df):
    df = df['text'].apply(lambda x:np.NaN if len(x)<=2 else x)
    return df

time: 2.56 ms (started: 2022-05-21 15:40:48 +00:00)
```

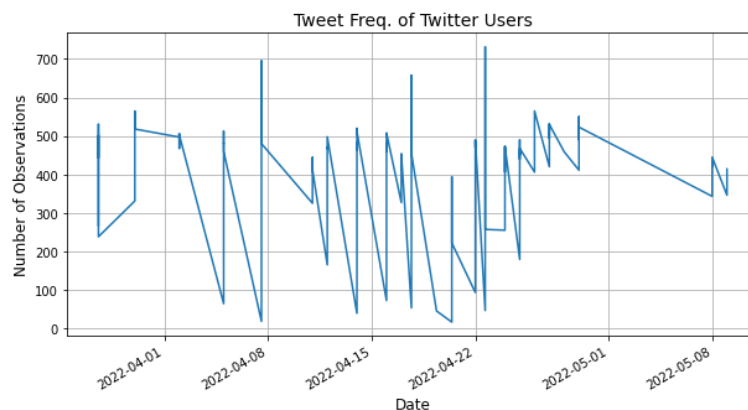
GAMBAR 10 PREPROCESSING DATASET

Gambar 10 merupakan contoh pembersihan data yang dilakukan. Dataset yang dikumpulkan secara manual melalui Twitter tentu saja tidak dapat langsung digunakan mengingat banyak kata-kata tertentu yang *meaningless* atau tidak bermakna. Hal tersebut dapat disebabkan oleh berbagai faktor misalkan saja karena kata tersebut merupakan kata penghubung, kata awalan yang tentu saja kemunculannya akan relatif lebih tinggi jika dibandingkan dengan kata-kata lain. Akan tetapi, kata tersebut sama sekali tidak dapat diinferensikan sebagai suatu kumpulan topik nantinya. Oleh karena itu, data-data yang sudah diambil perlu dilakukan pembersihan. Pembersihan data yang dilakukan pada penelitian ini meliputi:

- *Expand Contraction* → membuat kata-kata umum yang sering disingkat menjadi tidak disingkat, di sini karena tidak ada library yang bisa digunakan dari NTLK maupun *framework* lainnya, maka peneliti membuat *dictionary* atau kamus secara manual.
- *Remove Duplicate* → kata-kata yang terduplikasi dalam sebuah tweet akan dibuang karena akan menyebabkan kata tersebut memiliki kemunculan lebih sering daripada yang lainnya, padahal kata tersebut memiliki arti yang sama.
- *Remove 2 Words* → dari hasil penelitian yang dilakukan, tweet yang hanya mengandung 2 kata atau kurang biasanya tidak memiliki makna karena biasanya cuitan tersebut hanya sekedar sapaan seperti “Hello, Buddy!” yang tidak mungkin bisa dimanfaatkan untuk trending topics.
- *Remove punctuations* → tanda baca sering sekali muncul dalam tweet, akan tetapi tanda baca terkadang bisa menjadi rancu karena kemunculannya tidak bisa dimaknai apa-apa misalkan tanda seru (!) bisa saja merupakan peringatan, sapaan, perasaan kaget, tidak jelas arahnya, oleh karena itu di sini peneliti akan membuang tanda baca.

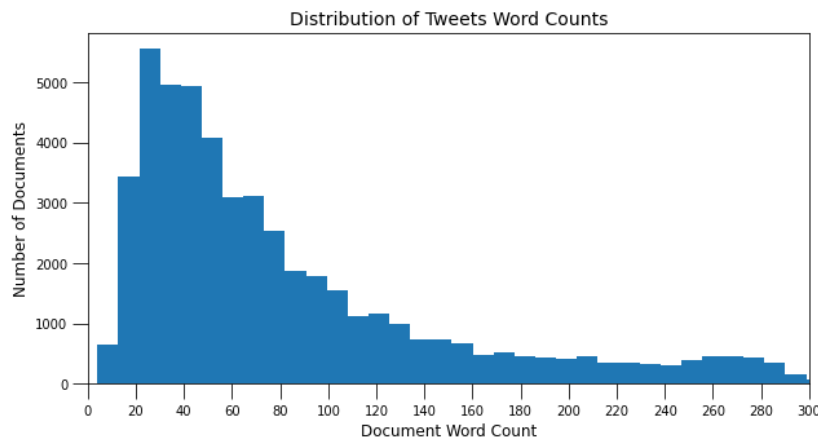
- *Remove number* → angka sering sekali muncul dalam tweet, akan tetapi sama seperti tanda baca, angka tidak bisa dijadikan sebagai acuan yang jelas dalam menentukan trending topics. Oleh karena itu, peneliti akan membuang angka yang muncul dengan regex.
- *Stopwords* → di sini peneliti menggunakan *library gensim* untuk membuang kata-kata yang sering muncul akan tetapi tidak memiliki makna seperti “the”, “a”, “an” dan lain sebagainya.
- Lematisasi → melihat dari kumpulan tweet banyak sekali kata yang sebenarnya bisa dikelompokkan menjadi satu kesatuan cluster seperti “listening” dengan “listen”, akan tetapi terpisah dari cluster karena model tidak bisa memahami kata-kata “listening” dan “listen” sebenarnya memiliki arti yang sama. Oleh karena itu, kata-kata tersebut akan diekstrak menjadi kata dasar, dengan harapan akan meningkatkan performa model.

E. Visualisasi Data



GAMBAR 11 FREKUENSI TWEET

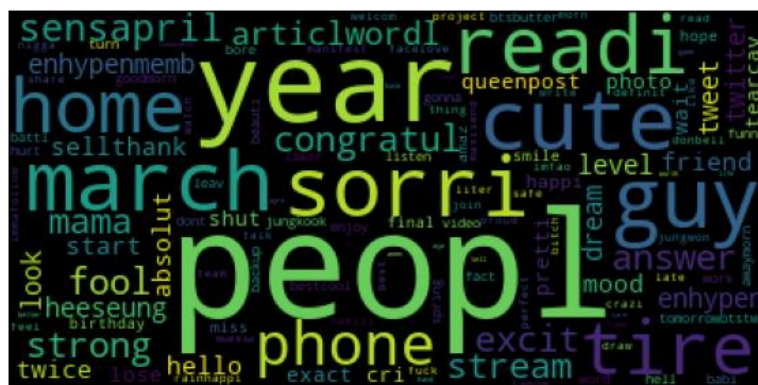
Gambar 11 merupakan contoh visualisasi dari seberapa banyak pengguna yang melakukan tweet pada waktu tertentu. Misalkan saja pada tanggal 01-04-2022 ada sekitar 500 user yang melakukan tweet. Akan tetapi setelah mencoba melakukan visualisasi data terhadap pengguna berdasarkan kurun waktu tertentu, hasil plot tersebut tidak dapat menggambarkan lebih mendalam apakah dataset yang peneliti bangun memiliki kecenderungan tertentu atau tidak. Oleh karena itu peneliti melakukan visualisasi data selanjutnya dengan histogram.



GAMBAR 12 DISTRIBUSI TWEET

Gambar 12 merupakan visualisasi data selanjutnya yang dilakukan oleh peneliti. Di sini lebih terlihat bahwa dataset yang dibangun memiliki kecenderungan ke arah kanan (*skewed to right*) yang berarti rata-rata pada dataset akan lebih besar dibandingkan mode di dataset. Artinya dataset yang dibangun bisa saja berdampak pada hasil akurasi model. Karena data tersebut tidak secara normal terdistribusi.

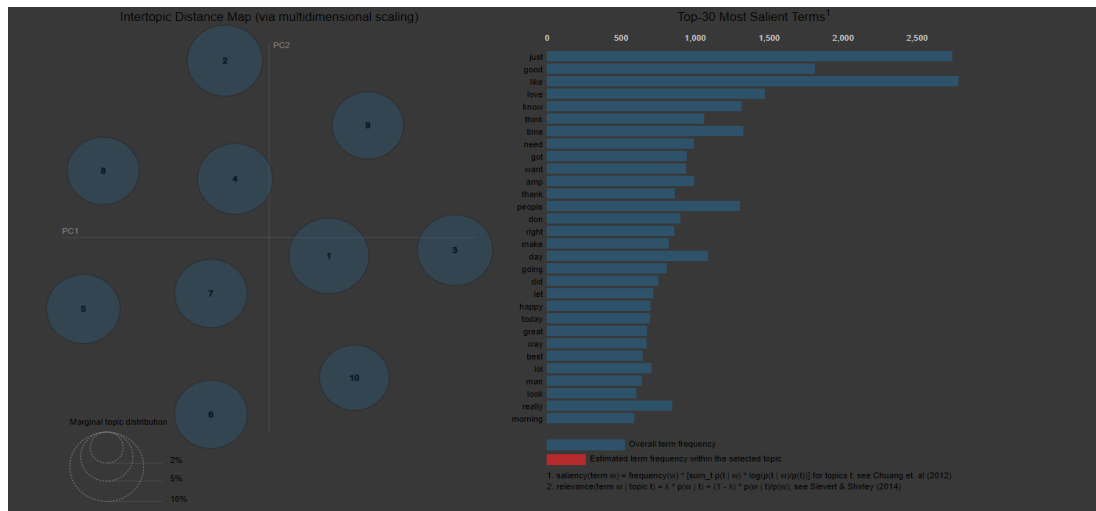
F. Word Cloud



GAMBAR 13 WORD CLOUD

Gambar 13 merupakan hasil visualisasi word cloud terhadap model LDA yang sudah dilakukan TF-IDF. Pada gambar ini menunjukkan kata apa saja yang kemunculannya yang relatif tinggi. Misalkan pada gambar ini kata “peopl” cenderung lebih banyak dibandingkan kata “queenpost” misalnya. Berapa banyak perbedaan yang terjadi? Apakah perbedaan signifikan? Bagaimana pendistribusian *topic modelling* tersebut? Hal ini tidak dapat dilakukan menggunakan word cloud. Oleh karena itu peneliti menggunakan pendekatan lain yaitu menggunakan pyLDAvis untuk melakukan visualisasi.

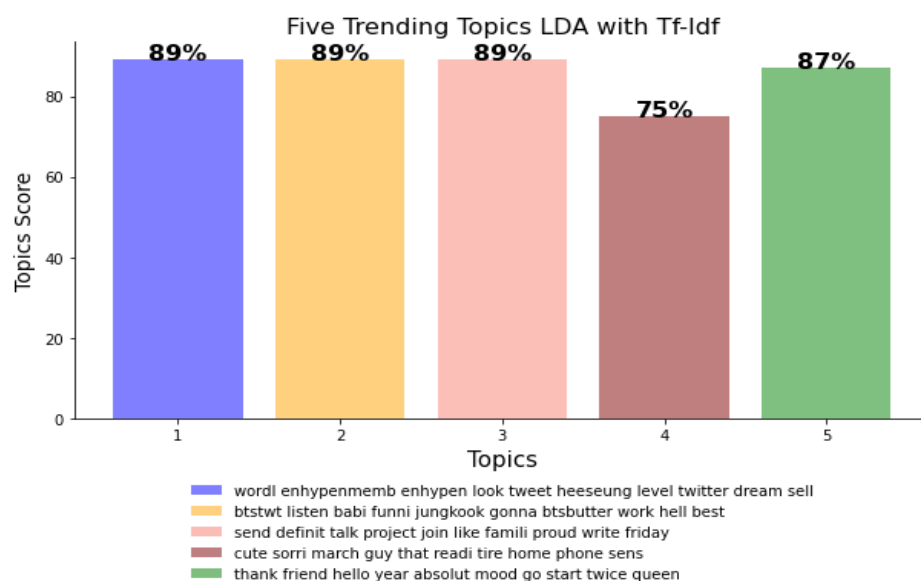
G. PyLDAvis



GAMBAR 14 PyLDAvis

Gambar 14 merupakan hasil visualisasi dengan pyLDAvis. Berdasarkan panel visualisasi, kemunculan term “like” menjadi kata yang paling sering muncul dalam sebuah korpus. Selain itu peneliti mencoba melakukan inferensi terkait *keyword-keyword* lainnya yang berkaitan dengan “like” yaitu misalkan kata “love”. Dalam PyLDAvis ini terdapat sebuah gambaran yang dijadikan *pre-assumption* di mana dokumen ini akan menghasilkan sebuah topik modeling yang berkaitan dengan suasana/perasaan seseorang. Oleh karena itu, peneliti selanjutnya membuat bar diagram terhadap 5 trending topik yang dihasilkan dengan pendekatan LDA menggunakan pembobotan TF-IDF.

H. Diagram 5 Trending Topik



GAMBAR 15 5 DIAGRAM TRENDING TOPIK

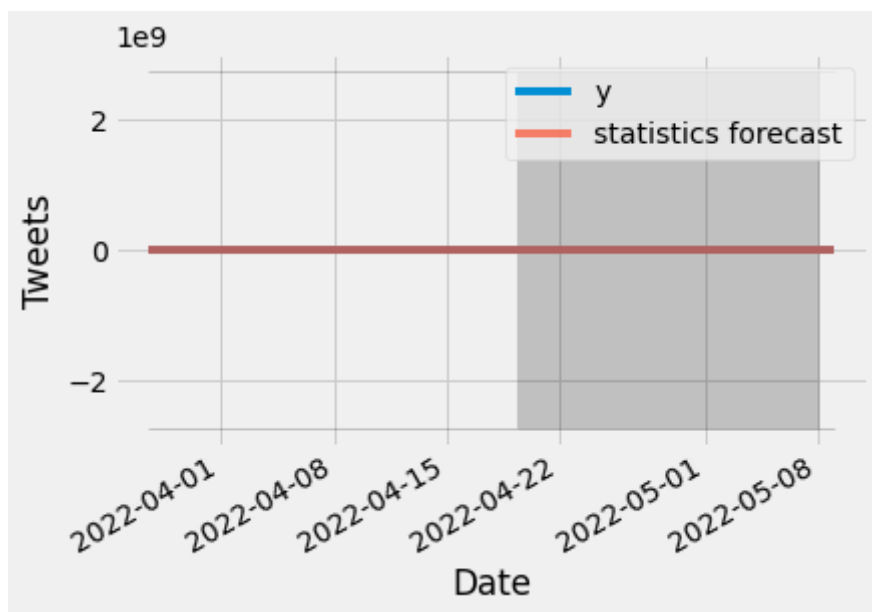
Gambar 15 ini merupakan hasil dari *plotting* model LDA dengan TF-IDF. Topik yang dihasilkan sebenarnya cukup banyak, akan tetapi di sini peneliti mengambil 5 topik teratas. Pada gambar ini dapat dilakukan interpretasi yaitu:

- Topik #1 (Boyband Korea) : wordl → setelah didiskusikan bersama kemungkinan kata wordl ini mengacu pada kata “wordle bts” yang merupakan permainan tebak kata BTS . Selanjutnya ada kata “enhypenmemb” mengacu pada kata “enhypen member” dan “enhypen” yang merupakan anggota dari *boyband* di Korea Selatan. Kemudian kata “look” di sini bukan merupakan outlier, karena ketika dicari melalui dataset original, kata “look” ini ternyata mengacu pada sebuah lagu BTS yaitu “look here”. Kemudian ada kata “tweet” jika dicari berdasarkan dataset original “tweet” dan “twitter” ini mengacu pada *hashtag* di mana *hash tag* tersebut menyajikan sebuah konten seperti meme terkait BTS. Selanjutnya ada kata “heeseung” yang merupakan nama member dari *boyband* Enhypen. Kemudian ada kata “level” → mengacu pada permainan level/tingkatan permainan wordle bts. Kemudian ada kata “dream” → mengacu pada lagu pertama BTS yaitu no more dream. Sedangkan untuk kata “sell” mengacu pada penjualan lagu/merchandise *boyband* Korea.
- Topik #2 (BTS) : btstwt → mengacu pada akun official milik BTS yaitu @bts_twt. kemudian kata “listen” di sini agak rancu mendengarkan apa. Akan tetapi jika melihat dari kedekatan kata di cluster ini kemungkinan kata “listen” ini kuat korelasinya dengan mendengarkan lagu-lagu *boyband* Korea. Kemudian kata “babi” di sini jika dilihat dari dataset original, kata ini mengacu pada “baby of BTS” yaitu Jungkook. Sama hal seperti kata “funi” yang merujuk pada kata “funny of Jungkook”. Selanjutnya kata “Jungkook” sendiri merupakan anggota BTS. Selanjutnya “btsbutter” mengenai lagu BTS yaitu Butter. Kemudian “best” di sini mengacu pada album “The Best of BTS”. Untuk kata “work” dan “hell” belum diketahui merujuk pada kata apa.
- Topik #3 (-) : Pada topik 3 ini, walaupun secara skornya cukup tinggi, akan tetapi, karena kata-kata yang digunakan cukup general. Peneliti tidak dapat memaknai apa yang dimaksud dalam cluster ini.
- Topik #4 (E Sens Rapper Korea) : Mengacu pada kata “march” → pada dataset original kata ini banyak digunakan terkait lagu-lagu e sens yang dirilis pada bulan Maret. Sedangkan kata “phone” di sini ternyata secara tidak langsung ada beberapa orang yang mengaitkan kata “e sens” kepada merk telepon genggam. Ciri khas pada topik ini adalah skornya yang relatif rendah dibandingkan topik lain dapat menjadikan sebuah makna

bahwa topik ini tingkat relevansinya cukup rendah dibandingkan dengan topik-topik lain.

- Topik #5 (Lagu Girlband Twice) : “Twice” mengacu pada girlband Korea. Selanjutnya kata “Queen” merupakan salah satu lagu Twice itu sendiri. Kemudian untuk kata “mood” mengacu pada sebuah gambar lelucon (meme) terkait mood/ekspresi wajah dari girlband Twice. Kemudian kata “thank” → mengacu pada judul lagu Twice yaitu “Thank you, Family”. Kemudian kata “Hello” mengacu pada salah satu lagu Twice. “Year” → mengacu pada lagu Twice “The Year of Yes”.

Secara garis besar, topik-topik yang dihasilkan jika kita tarik secara general, hampir sebagian besar topik modeling yang dihasilkan berkaitan langsung dengan grup band Korea Selatan. Hal ini dapat divalidasi mengingat pada faktanya memang saat ini tingkat popularitas dari grup band Korea Selatan terbilang cukup tinggi di kalangan masyarakat.



GAMBAR 16 HASIL FORECASTING

Gambar 16 merupakan hasil *forecasting* yang sudah dilakukan, walaupun pada gambar ini terlihat masih belum dapat menampilkan hasil yang baik dalam mencoba memprediksi berdasarkan *test set* yang diberikan. Kemungkinan hal ini dikarenakan pendekatan *clustering* yang belum tepat. Hasil prediksi dapat dikatakan baik apabila garis ‘y’ dan garis ‘statistics forecast’ tidak bertumpuk dengan tepat dan juga bisa dengan menggunakan perhitungan MSE semakin kecil MSE maka hasil prediksi dapat dikatakan semakin baik.

IV.2 DATASET WEBSITE

A. Contoh Implementasi Kode untuk Mengambil URL dari Halaman Web

```
In [1]: from urllib.request import urlopen
        from bs4 import BeautifulSoup as soup
        import re

In [2]: idn = urlopen("https://www.idntimes.com/")
        bsobj = soup(idn.read())

In [3]: for link in bsobj.findAll('a'):
        if 'href' in link.attrs:
            print(link.attrs['href'])
```

GAMBAR 17 IMPLEMENTASI KODE UNTUK WEB CRAWLER

Hal pertama yang perlu dilakukan adalah mengimpor *urlopen* dari *urllib.request* untuk membuka halaman website, dan juga *BeautifulSoup* dari *bs4* untuk mengekstrak URL dari halaman *website* yang akan peneliti *crawling*. Untuk menampilkan seluruh URL yang terdapat pada *website*, gunakan *looping for* untuk memeriksa setiap *href* yang ada. Pada penelitian ini, URL yang telah diekstrak akan menjadi dataset. Bahasa pemrograman yang peneliti gunakan untuk melakukan ekstraksi data adalah *Python*.

B. Hasil Pengambilan Data URL dari Web Crawling

```
https://www.idntimes.com
#search-modal
https://community.idntimes.com/login
https://ramadan.idntimes.com
https://www.idntimes.com/quiz
https://www.idntimes.com/news
https://www.idntimes.com/business
https://www.idntimes.com/sport
https://www.idntimes.com/tech
https://www.idntimes.com/hype
https://www.idntimes.com/korea
https://www.idntimes.com/life
https://www.idntimes.com/health
https://community.idntimes.com
#
https://www.idntimes.com
https://jabar.idntimes.com
https://banten.idntimes.com
https://jateng.idntimes.com
```

GAMBAR 18 HASIL CRAWLING DARI WEBSITE

Berdasarkan hasil diskusi dengan pihak DailySocial.id, terdapat beberapa *website* kompetitor. Beberapa di antaranya yaitu *kumparan.com*, *gizmologi.id*, dan *idntimes.com*. Pada gambar XXX, peneliti mengambil URL yang terhubung dengan *website idntimes.com*.

Namun dikarenakan seperti yang bisa dilihat pada gambar, dengan menggunakan *beautiful soup* baru saja mendapatkan seluruh link yang ada dan itupun belum tersaring hanya yang beritanya saja sesuai yang penulis butuhkan, maka dari link-link yang ada tersebut, *web crawling* dilanjutkan dengan *software* pihak ketiga sehingga menghasilkan CSV seperti ini (berikut merupakan contoh data berita untuk *gizmologi*).

Title	Title_URL	Image	daymonth	Author_URL	Author	Comment	View
Xiaomi Mi Band 7 Janjian Layar Lebih Besar 25%, https://gizmologi.id/news/xiaomi-mi-band-7-janjian-layar-lebih-besar/ ,	21 Mei	https://gizmologi.id/author/sarifah/	Siti Sarifah A,0	101			
Penampakan Tablet realme Pad X 5G, Ada Pensilnya, https://gizmologi.id/news/penampakan-tablet-realme-pad-x-5g-ada-pen/ ,	21 Mei	https://gizmologi.id/author/sarifah/	Siti Sarifah A,0	107			
Google Menghapus Jumlah Pengguna Android 127, https://gizmologi.id/news/pengguna-android-127/ ,	21 Mei	https://gizmologi.id/author/chawir/	Chandra Wirawan,0	107			
Kantor Pusat Nikkei Group Asia di Singapura Dapat Serangan Siber, https://gizmologi.id/news/serangan-siber-singapura/ ,	21 Mei	https://gizmologi.id/author/chawir/	Chandra Wirawan,0	117			
Berlisensi Bappeti, Pluang Tawarkan Investasi Emas Digital, https://gizmologi.id/news/pluang-tawarkan-investasi-emas-digital/ ,	20 Mei	https://gizmologi.id/author/aditya/	Aditya Fajar,0	151			
Perkuat Ekosistem Digital, Huawei Tingkatkan Kontribusi di Indonesia, https://gizmologi.id/news/huawei-perkuat-ekosistem-digital/ ,	20 Mei	https://gizmologi.id/author/aditya/	Aditya Fajar,0	142			
Mastercard Gandeng Ayocconnect Hadirkan Open Banking, Limit Hingga Rp30 Juta, https://gizmologi.id/news/mastercard-ayoconnect-open-banking/ ,	20 Mei	https://gizmologi.id/author/aditya/	Aditya Fajar,0	164			
Telkomsel dan Kredivo Hadirkan Layanan Utang PayLater, Limit Hingga Rp30 Juta, https://gizmologi.id/news/telco/kredivo-telkomsel-paylater/ ,	19 Mei	https://gizmologi.id/author/mrbambang/	Bambang Dwi Atmoko,0	161			
Pelanggan IndiHome Berkesempatan Nonton NCT Dream & Red Velvet di Allo Bank Festival, https://gizmologi.id/news/pelanggan-indihome-allo-bank-festival/ ,	19 Mei	https://gizmologi.id/author/mrbambang/	Bambang Dwi Atmoko,0	184			
Masuk ke Indonesia, EdgeConnex Bangun Pusat Data Hyperscale di Jakarta, https://gizmologi.id/news/edgeconnex-pusat-data-hyperscale-di-jakarta/ ,	19 Mei	https://gizmologi.id/author/aditya/	Aditya Fajar,0	183			
Hore! Pengguna OPPO Find XS Pro Bakal Dapatkan Update Android 13 Beta 1, https://gizmologi.id/news/oppo-find-xs-pro-dapatkan-update-android-13-beta-1/ ,	19 Mei	https://gizmologi.id/wp-content/uploads/2022/05/OSC09706-Copy-390x220.jpg	19 Mei	https://gizmologi.id/author/aditya/			
Sekarang Beli Paket Data Smartfren Bisa Lewat Facebook, https://gizmologi.id/news/telco/beli-paket-data-smartfren-lewat-facebook/ ,	19 Mei	https://gizmologi.id/wp-content/uploads/2021/03/Smartfren-Unlimited.jpg	18 Mei	https://gizmologi.id/author/aditya/			
Grab Tekan Mol! dengan BSSN, Perkuat Keamanan Siber Mitra dan Merchant, https://gizmologi.id/news/grab-mou-bssn-siber-mitra-dan-merchant/ ,	18 Mei	https://gizmologi.id/wp-content/uploads/2020/12/Grab-Bike.jpg	18 Mei	https://gizmologi.id/author/aditya/			
Rebranding, PINTAR Berdayakan Angkatan Kerja Melalui Platform Pendidikan, https://gizmologi.id/news/startup/pintar-platform-pendidikan-kerja/ ,	18 Mei	https://gizmologi.id/wp-content/uploads/2022/05/Pintar-390x220.jpg	18 Mei	https://gizmologi.id/author/mr			
Daftar Merchant ShopeeFood & ShopeePay Kini Bisa Langsung dari Aplikasi!, https://gizmologi.id/news/registrasi-mandiri-shopeefood-shopeepay/ ,	18 Mei	https://gizmologi.id/wp-content/uploads/2021/08/ShopeePay-17.8-Semangat-UMKM-Lokal-1.jpg	18 Mei	https://gizmologi.id/author/mr			
Indonesia Baru Mulai 5G, Samsung Sudah Mau Geber Teknologi 6G, https://gizmologi.id/news/indonesia-baru-5g-samsung-geber-teknologi-6g/ ,	17 Mei	https://gizmologi.id/wp-content/uploads/2022/05/6G-Forum-main2-390x220.jpg	17 Mei	https://gizmologi.id/author/mr			
Bertemu Jokowi di Space X, Elon Musk Siap ke Indonesia Bahas Proyek Masa Depan, https://gizmologi.id/news/elon-musk-jokowi-proyek-masa-depan-indonesia/ ,	17 Mei	https://gizmologi.id/wp-content/uploads/2022/05/Elon-Musk-Jokowi-390x220.jpeg	17 Mei	https://gizmologi.id/author/mr			
ipSCAPE Perkuat Kehadiran di Asia, Tunjuk Erik Meijer sebagai Komisaris ipSCAPE, https://gizmologi.id/news/enterprise/erik-meijer-komisaris-ipscap/ ,	17 Mei	https://gizmologi.id/wp-content/uploads/2022/05/Erik-Meijer-ok-390x220.jpg	17 Mei	https://gizmologi.id/author/mr			
iOS 15.5 Hadir Bawa Fitur Baru, Berikut Cara Mengunduhnya, https://gizmologi.id/news/ios-15-5-dirilis/ ,	17 Mei	https://gizmologi.id/wp-content/uploads/2021/11/Apple-iPhone-8-0010.jpg	17 Mei	https://gizmologi.id/author/pras/			
Huawei Donasikan Laptop Lewat Program Young Genius, Bantu Proses Belajar Siswa, https://gizmologi.id/news/huawei-donasikan-laptop-program-young-genius/ ,	17 Mei	https://gizmologi.id/wp-content/uploads/2022/05/Huawei-Matebook-Young-Genius-390x220		https://gizmologi.id/author/mr			
OPPO A16k Tawarkan Performa dan Kualitas Mantap Harga Rp2 Jutaan, https://gizmologi.id/news/oppo-a16k-harga-2jutaan/ ,	17 Mei	https://gizmologi.id/wp-content/uploads/2022/05/oppo-16k-indonesia-390x220.webp	17 Mei	https://gizmologi.id/author/mr			
Booyah! Tim Free Fire Indonesia Sumbang Emas dan Perak di SEA Games Vietnam, https://gizmologi.id/news/tim-free-fire-indonesia-di-sea-games-vietnam/ ,	15 Mei	https://gizmologi.id/wp-content/uploads/2022/05/WhatsApp-Image-2022-05-15-at-9.50.37-PM-3		https://gizmologi.id/author/mr			
Lebaran 2022, Lonjakan Trafik Internet Smartfren di Atas 10%, https://gizmologi.id/news/telco/smartfren-traffic-lebaran-2022/ ,	15 Mei	https://gizmologi.id/wp-content/uploads/2021/12/BTS-Smartfren-di-BSD-Tangerang.jpg	15 Mei	https://gizmologi.id/author/mr			
Mengenal Glance, Unicorn India yang Merambah Pasar Indonesia Melalui Platform Lock-screen, https://gizmologi.id/news/glance-platform-lock-screen/ ,	15 Mei	https://gizmologi.id/wp-content/uploads/2022/05/Glance-ok-390x220.jpg	15 Mei	https://gizmologi.id/author/mr			
Lazada: Perdaazanan Digital Jadi Pendorong Pertumbuhan Ritel Q2 2022, https://gizmologi.id/news/digital-commerce-confidence-index-lazada-2022/ ,	15 Mei	https://gizmologi.id/wp-content/uploads/2021/08/Ilustrasi-atm-e-commerce-belanja-online-pexels-anna					

GAMBAR 19 DATA CSV UNTUK WEBSITE GIZMOLOGI

C. Menampilkan Data CSV dari masing-masing Web Kompetitor

Import Data

```
In [1]: import pandas as pd
import numpy as np
```

```
In [2]: data = pd.read_csv("idntimes.csv")
```

```
In [3]: data
```

```
Out[3]:
```

	Title	Title_URL	Image	Date
0	Joe Biden Setujui Penjualan Senjata ke Mesir S...	https://www.idntimes.com/news/world/zidan-patr...	NaN	\n \n ...
1	[UPDATE] Kasus COVID-19 Dunia Naik 598 Ribu, K...	https://www.idntimes.com/news/world/vadhia-ild...	NaN	\n \n ...
2	Imbas Krisis Pangan, Milisi Tigray Bebaskan 4....	https://www.idntimes.com/news/world/pri-145/im...	NaN	\n \n ...
3	Banyak Penolakan, Bintang Film Dewasa Miyabi B...	https://www.idntimes.com/news/indonesia/irfanf...	NaN	\n \n ...
4	66 Jam yang Menegangkan: Kronologi Jelang Soeh...	https://www.idntimes.com/news/indonesia/gregor...	NaN	\n \n ...
...
1214	Raja Salman dan Putra Mahkota Salat Idul Fitri...	https://www.idntimes.com/news/world/sonya-mich...	https://www.idntimes.com/assets/img/placeholde...	\n \n 02...
1215	Hari Buruh Momen Evaluasi Isu Kekerasan Peremp...	https://www.idntimes.com/news/indonesia/ila-hu...	https://www.idntimes.com/assets/img/placeholde...	\n \n 02...
1216	Prabowo Silaturahmi ke Jokowi di Yogya, Disugu...	https://www.idntimes.com/news/indonesia/muhamm...	https://www.idntimes.com/assets/img/placeholde...	\n \n 02...
1217	Ketegangan di Semenanjung Korea, China Dorong ...	https://www.idntimes.com/news/world/sonya-mich...	https://www.idntimes.com/assets/img/placeholde...	\n \n 02...
1218	Ratusan Narapidana Rutan Kelas I Depok Dapat R...	https://www.idntimes.com/news/indonesia/dicky-...	https://www.idntimes.com/assets/img/placeholde...	\n \n 02...

1219 rows x 4 columns

GAMBAR 20 DATA CSV YANG DIAMBIL DARI WEBSITE KUMPARAN

Data *csv* yang penulis gunakan dari ketiga website kompetitor memiliki jumlah dan judul kolom yang berbeda. Namun di sini, yang akan digunakan untuk penelitian adalah kolom *Title* yang merupakan judul dari suatu berita. Gambar di atas merupakan contoh data yang diambil dari *kumparan.com*.

D. Preprocessing Dataset

Case Folding

```
In [4]: data['Title'] = data['Title'].str.lower()
print(data['Title'])

0      cek tkp: kelonggaran penggunaan masker saat cf...
1      menguak fakta ade firmansyah, sang pengemudi b...
2      ada 54 titik pedulilindungi di cfd sudirman-th...
3      jokowi di rakernas projo: bicara persoalan ban...
4      foto: petani brasil kehilangan mata pencaharia...
...
245     pns di bali curi ponsel anak saudara tetangga,...
246     mobil fortuner terjerumus ke parit sedalam 3 m...
247     pria di bali jadi korban pembacokan di indekos...
248     jelang lebaran, 135 preman di medan ditangkap ...
249     jokowi serahkan bansos di gedung pos: jangan d...
Name: Title, Length: 250, dtype: object
```

Split Words

```
In [5]: data['Title'] = data['Title'].str.split()
print(data['Title'])

0      [cek, tkp:, kelonggaran, penggunaan, masker, s...
1      [menguak, fakta, ade, firmansyah,, sang, penge...
2      [ada, 54, titik, pedulilindungi, di, cfd, sudi...
3      [jokowi, di, rakernas, projo:, bicara, persoal...
4      [foto:, petani, brasil, kehilangan, mata, penc...
...
245     [pns, di, bali, curi, ponsel, anak, saudara, t...
246     [mobil, fortuner, terjerumus, ke, parit, sedal...
247     [pria, di, bali, jadi, korban, pembacokan, di,...
248     [jelang, lebaran,, 135, preman, di, medan, dit...
249     [jokowi, serahkan, bansos, di, gedung, pos:, j...
Name: Title, Length: 250, dtype: object
```

Removing Symbol

```
In [6]: symbol_list = "!\"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\\n\\,0123456789"
for i in range(250):
    for c in symbol_list:
        data['Title'][i] = np.char.replace(data['Title'][i], c, "")
print(data['Title'])
```

GAMBAR 21 BEBERAPA METODE PREPROCESSING WEBSITE

Data pada kolom *title* yang akan penulis teliti memang sudah menggunakan kata baku, namun bukan berarti data tersebut sudah dapat digunakan untuk penelitian. Hal ini dikarenakan masih terdapat banyak kata-kata yang tidak memiliki makna namun muncul secara sering, kata-kata ini harus dihilangkan agar hasil penelitian lebih akurat. Selain itu juga, masih banyak kata yang masih memiliki imbuhan atau yang tercampur dengan simbol-simbol tertentu, walaupun kata-kata tersebut sering muncul nantinya tidak akan terdeteksi jika tidak dimurnikan. Oleh karena itu, berikut ini hal-hal yang penulis lakukan untuk menambah kemurnian dari data yang akan digunakan :

- *Case Folding* → mengubah semua kata menjadi hanya terdiri dari huruf kecil saja. Pada saat analisis data dilakukan, satu kata yang sama jika memiliki perbedaan pada penggunaan huruf kapital akan terdeteksi menjadi kata yang berbeda. Oleh karena itu,

data harus terlebih dahulu disamakan dengan diubah bentuknya menjadi *lowercase* atau tidak ada lagi yang menggunakan huruf kapital.

- *Split Words* → memisahkan data menjadi per kata kemudian menaruhnya dalam sebuah array. Awalnya, judul membentuk suatu kalimat. Namun karena penulis akan meneliti kata yang sering muncul, maka kalimat tersebut dipisahkan menjadi kumpulan kata yang disimpan dalam sebuah array untuk satu judulnya.
- *Removing Symbol* → membuang semua karakter non huruf dari data. Sebelumnya, data yang ada belum murni dari karakter-karakter yang bukan huruf. Masih terdapat tanda baca, simbol-simbol tidak bermakna lainnya, dan juga terdapat angka. Semua karakter tersebut akan dihilangkan dengan cara mendeklarasikan karakter apa saja yang akan dihilangkan, kemudian jika menemukan karakter tersebut pada data maka akan dihapus.
- *Remove Stopwords* → membuang kata-kata tidak bermakna yang sering muncul. Untuk daftar kata apa saja yang sebaiknya dibuang, penulis mengambilnya dari library NLTK yang berbahasa Indonesia, kemudian mencari dan membuangnya menggunakan iterasi.

Semua proses untuk membersihkan data ini dilakukan pada ketiga data dari masing-masing web kompetitor.

E. Implementasi Kode untuk Penerapan Algoritma TF-IDF

Term Frequency

```
from sklearn.feature_extraction.text import CountVectorizer

a = len(data['Title'])

# Create a vectorizer object
vectorizer = CountVectorizer()
vectorizer.fit(data['Title'])

# Printing the identified unique words along with their indices
print("Vocabulary: ", vectorizer.vocabulary_)

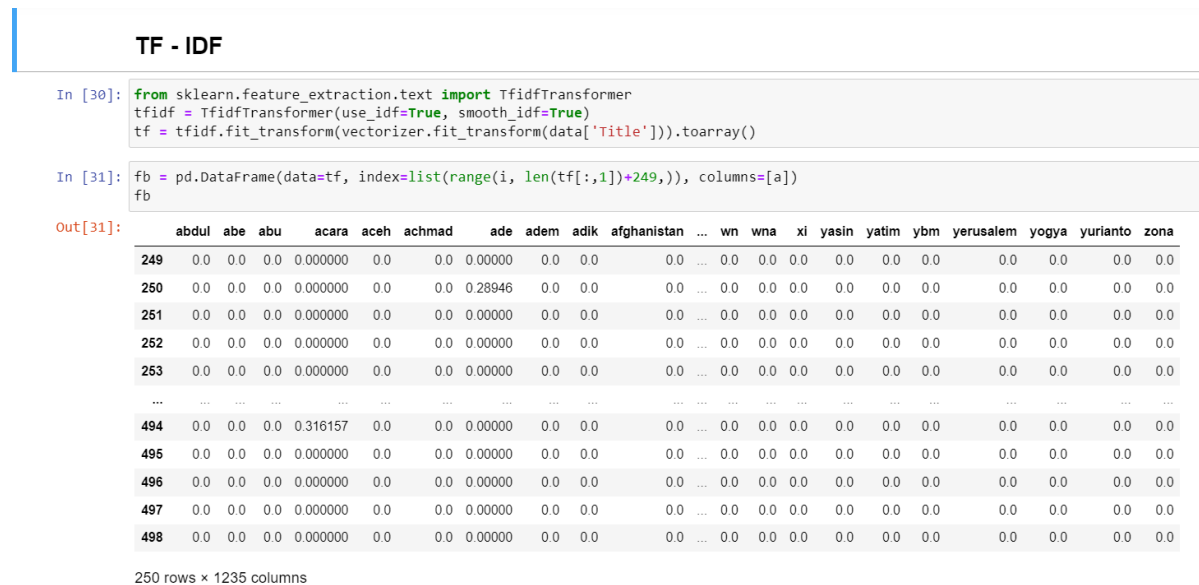
# Encode the data
vector = vectorizer.transform(data['Title'])

# Summarizing the Encoded Texts
print("Encoded document is:")
print(vector.toarray())
```

Vocabulary: {'cek': 173, 'tkp': 1141, 'kelonggaran': 492, 'penggunaan': 808, 'masker': 638, 'cfd': 178, 'bundaran': 159, 'hi': 376, 'menguak': 666, 'fakta': 315, 'ade': 6, 'firmansyah': 320, 'sang': 967, 'pengemudi': 805, 'bus': 164, 'maut': 64, 'tol': 1144, 'mojokerto': 689, 'titik': 1140, 'peduliilindungi': 768, 'sudirmanthamrin': 1058, 'warga': 1211, 'diimbau': 241, 'scan': 980, 'jokowi': 443, 'rakernas': 916, 'projo': 898, 'bicara': 137, 'bangsa': 81, 'pemilu': 785, 'foto': 326, 'petani': 855, 'brasil': 150, 'kehilangan': 487, 'mata': 643, 'pencarian': 792, 'akibat': 17, 'banjir': 82, 'senang': 995, 'dibuka': 230, 'langkah': 584, 'kehidupan': 486, 'normal': 718, 'suasana': 1055, 'ramai': 919, 'olahraga': 724, 'mayat': 64, 'korban': 552, 'terowongan': 1123, 'runtuh': 952, 'khasmir': 527, 'india': 399, 'ditemukan': 275, 'polisi': 872, 'tutup': 1162, 'jalan': 422, 'sudirmanmh': 1057, 'thamrin': 1131, 'masyarakat': 642, 'ketum': 526, 'pbnu': 764, 'bendera': 110, 'lgbt': 599, 'kedubes': 482, 'inggris': 404, 'silakan': 1025, 'urusan': 1182, 'kondisi': 547, 'tabrak': 1073, 'rumah': 951, 'ciamis': 181, 'sebabkan': 983, 'orang': 728, 'tewas': 1129, 'din': 252, 'syamsuddin': 1071, 'pidato': 857, 'kazan': 475, 'kolaborasi': 542, 'rusiaislam': 954, 'jembatan': 434, 'peradaban': 826, 'hujan': 385, 'angin': 39, 'pekanbaru': 771, 'mat': 644, 'lampu': 581, 'jam': 424, 'papan': 750, 'reklame': 928, 'timpa': 1136, 'kedai': 480, 'populer': 879, 'achmad': 5, 'yurianto': 1233, 'meninggal': 670, 'dunia': 298, 'anggota': 38, 'brimob': 152, 'letuskan': 598, 'tembakan': 1089, 'indahny': 397, 'lukisan': 612, 'pantai': 749, 'pacitan': 734, 'karya': 472, 'sby': 979, 'hasil': 371, 'melukis': 655, 'ribuan': 9, 'siswa': 1034, 'tradisi': 1149, 'sanad': 965, 'alquran': 28, 'pesantren': 853, 'daqu': 202, 'hilang': 380, 'dijemput': 242, 'gadis': 328, 'melang': 626, 'diduga': 235, 'trafficking': 1150, 'prof': 896, 'wiku': 1219, 'kenang': 508, 'garda': 33, 'terdepan': 1105, 'ri': 934, 'dilanda': 249, 'covid': 189, 'sinyal': 1030, 'dukung': 296, 'ganjar': 330, 'pranowo': 885, 'dapatkah': 201, 'uu': 1188, 'tpks': 1148, 'kawin': 474, 'anak': 36, 'swiss': 1069, 'laporkan': 590, 'cacar': 165, 'monyet': 691, 'terinfeksi': 1109, 'negeri': 712, 'viryan': 1196, 'aziz': 67, 'terapi': 1097, 'cuci': 190, 'otak': 731, 'dsa': 29}

GAMBAR 22 HASIL TERM FREQUENCY WEBSITE

Sebelum masuk ke TF-IDF, berikut ini merupakan implementasi dari teknik Term Frequency menggunakan metode *CountVectorizer* dari *sklearn*. Hasil dari metode tersebut adalah akan ditampilkan seluruh kata yang digunakan beserta nilai Term Frequency nya. Semakin besar nilainya, semakin sering kata tersebut muncul.



GAMBAR 23 HASIL TF-IDF UNTUK DATA WEBSITE

Kemudian selanjutnya untuk nilai dari TF-IDF akan dihitung menggunakan metode *TfidfTransformer*. Hasilnya akan muncul kata yang digunakan dari a sampai dengan z beserta nilai TF-IDF untuk setiap barisnya. Sama seperti nilai *term frequency* tadi, semakin besar nilainya, maka kata tersebut semakin sering muncul. Jika nilainya 0 berarti kata tersebut tidak muncul sama sekali pada baris yang dimaksud.

F. Implementasi Word Count untuk Melihat Kata yang Sering Muncul

Untuk selanjutnya, dilakukan visualisasi data dalam bentuk *word cloud* yang bertujuan untuk mengetahui kata apa yang paling sering muncul dan seberapa sering suatu kata muncul. Semakin besar ukuran katanya, maka semakin sering kata tersebut muncul.

G. Implementasi Latent Dirichlet Allocation dan Visualisasi PyLDAvis

```
# Create the object for LDA model using gensim library
Lda = gensim.models.ldamodel.LdaModel

total_topics = 10
number_words = 10

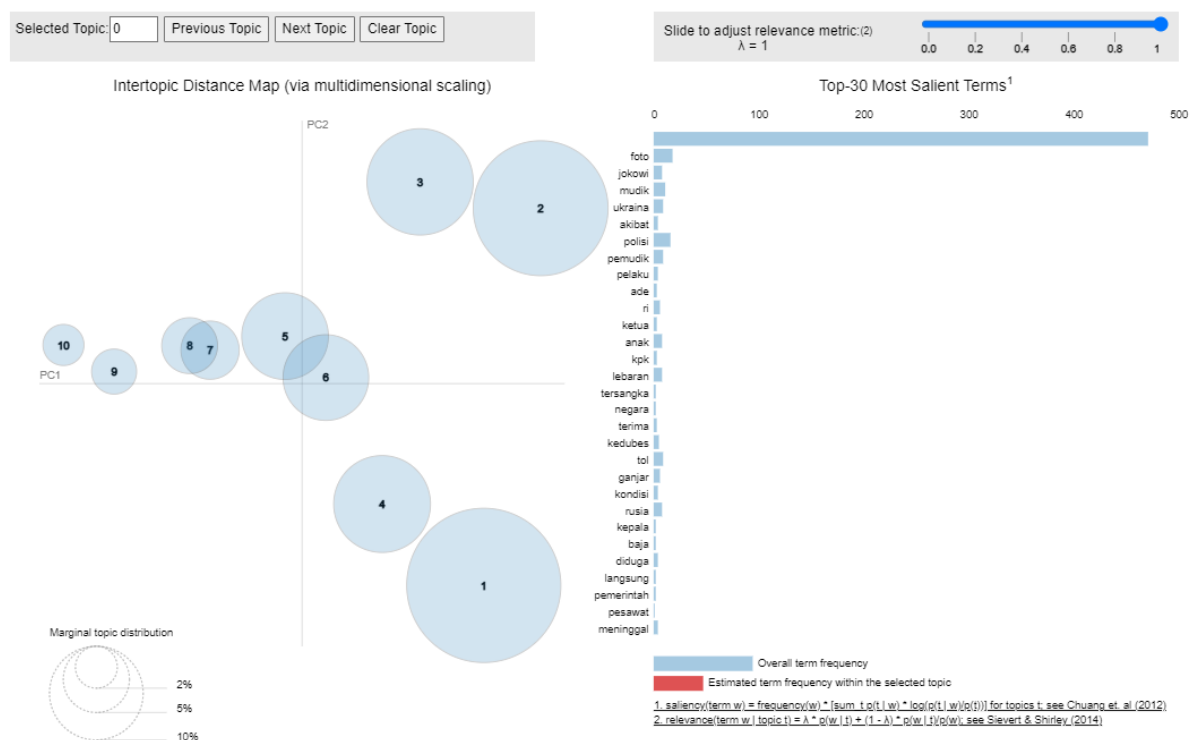
# Running and Training LDA model on the document term matrix
lda_model = Lda(doc_term_matrix, num_topics=total_topics, id2word = dictionary, passes = 50)

lda_model.show_topics(num_topics=total_topics, num_words=number_words)

[(0,
 '0.151*** + 0.018**polisi" + 0.013**tol" + 0.008**cfd" + 0.008**ade" + 0.007**warga" + 0.005**kota" + 0.005**one" + 0.005**way" + 0.005**lebaran'),
 (1,
 '0.148*** + 0.012**anak" + 0.006**bandung" + 0.006**pesantren" + 0.006**brimob" + 0.006**minyak" + 0.006**barang" + 0.006**ramadhan" + 0.006**temukan" + 0.006**mudik'),
 (2,
 '0.055*** + 0.010**mudik" + 0.010**jokowi" + 0.010**foto" + 0.005**lebaran" + 0.005**jakarta" + 0.005**bali" + 0.005**arus" + 0.005**ribuan" + 0.005**orang'),
 (3,
 '0.131*** + 0.012**foto" + 0.009**mudik" + 0.006**lebaran" + 0.006**warga" + 0.006**pemudik" + 0.006**jelang" + 0.006**corona" + 0.006**bawa" + 0.006**polisi'),
 (4,
 '0.218*** + 0.010**pemudik" + 0.009**kedubes" + 0.009**foto" + 0.007**ganjar" + 0.007**mudik" + 0.007**ukraina" + 0.006**tewas" + 0.006**singapura" + 0.006**demo'),
 (5,
 '0.253*** + 0.006**orang" + 0.006**israel" + 0.006**jokowi" + 0.006**polisi" + 0.005**indonesia" + 0.005**ri" + 0.005**jelang" + 0.004**g" + 0.004**terkait'),
 (6,
 '0.025*** + 0.007**kpk" + 0.007**mudik" + 0.007**pemerintah" + 0.007**ketua" + 0.007**kunjungi" + 0.007**terima" + 0.007**dukungan" + 0.007**ri" + 0.007**ade'),
 (7,
 '0.170*** + 0.017**foto" + 0.013**ukraina" + 0.011**rusia" + 0.009**kondisi" + 0.007**covid" + 0.007**presiden" + 0.005**perang" + 0.005**uas" + 0.005**diduga'),
 (8,
 '0.086*** + 0.010**foto" + 0.010**akibat" + 0.010**pelaku" + 0.005**korban" + 0.005**kerja" + 0.005**dunia" + 0.005**lebaran" + 0.005**kpk" + 0.005**covid'),
 (9,
 '0.014**pesawat" + 0.007**jokowi" + 0.007**kepala" + 0.007**baja" + 0.007**korupsi" + 0.007**tersangka" + 0.007**negara" + 0.007**tinjau" + 0.007**langsung" + 0.007**bentu
 k'')]
```

GAMBAR 26 PEMBUATAN LDA MODEL UNTUK WEBSITE KUMPARAN

Sebelum melakukan visualisasi LDA menggunakan PyLDAvis, dibuat model terlebih dahulu. LDA ini bertujuan untuk membagi kata-kata sesuai dengan kategorinya. Di sini, penulis menggunakan 10 jenis kategori dengan jumlah kata untuk masing-masing kategorinya juga adalah 10.



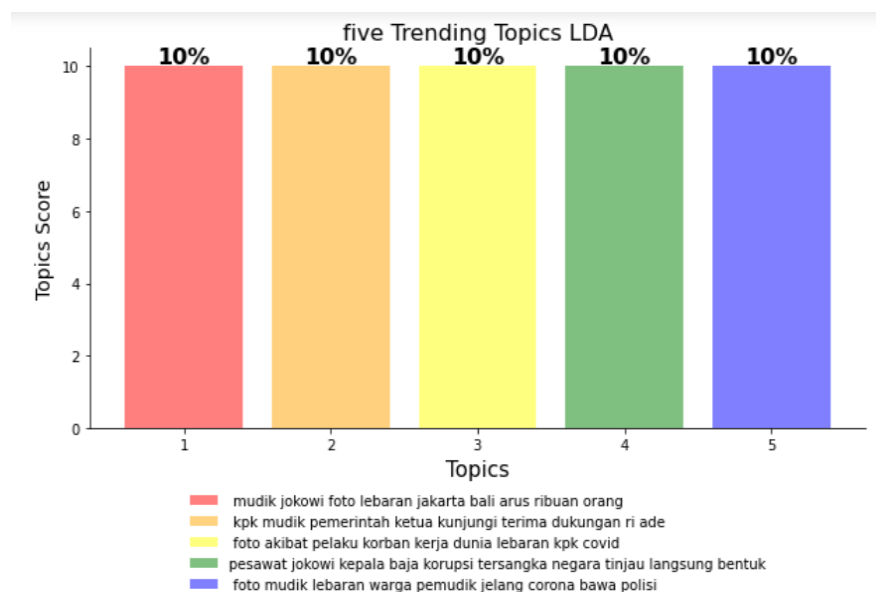
GAMBAR 27 VISUALISASI PYLDAVIS UNTUK WEBSITE KUMPARAN

Berikut merupakan hasil visualisasi PyLDAvis untuk website kumparan. LDAvis memungkinkan penulis untuk dapat melihat 30 kata paling relevan secara umum, 30 kata paling

relevan secara khusus untuk masing-masing topik, dan juga menentukan untuk suatu kata tertentu akan lebih condong masuk ke topik yang mana (semakin besar ukuran lingkaran topik, maka kata semakin cocok dengan topik tersebut). Dapat dilihat bahwa kata yang memiliki *overall term frequency* yang tinggi memang kata yang muncul dengan ukuran besar pada *word cloud*.

H. Top 5 Topik oleh Bar Diagram

Seluruh bar menunjukkan hasil yang merata yang berarti seluruh topik yang berada di top 5 muncul dengan merata.



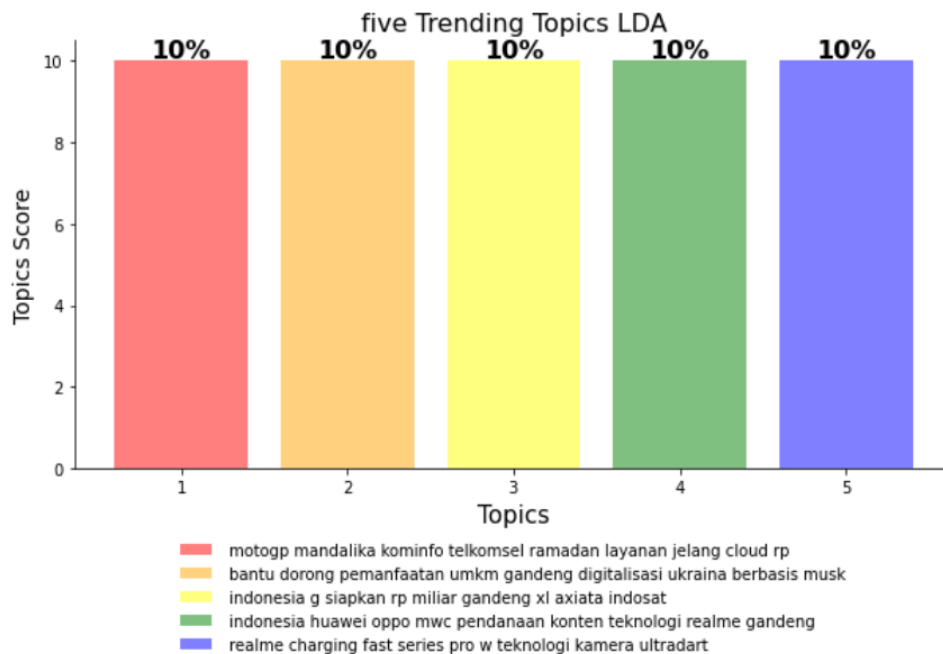
GAMBAR 28 DIAGRAM TOP 5 TOPIC WEBSITE KUMPARAN

Walaupun hasilnya masih kurang baik karena dalam satu topik tidak semuanya saling menyambung satu sama lain (biasanya satu kata dengan kata lainnya, kemudian kata lainnya tersebut dengan kata yang lain lagi), namun dapat terlihat korelasi antar topik.

- Topik #1 → Mengacu kepada berita Pak Jokowi memberi lampu hijau untuk mudik, dan ribuan orang diperkirakan akan menyeberang dari Bali ke Jawa pada arus mudik lebaran tersebut.
- Topik #2 → Mengacu kepada berita tentang Ketua DPR RI, Ade Komarudin, yang diperiksa KPK sebagai saksi tersangka tindak korupsi
- Topik #3 → Mengacu kepada berita yang menunjukkan akibat *covid* dalam dunia kerja dan juga berita mengenai pelaku dan korban korupsi bansos *covid*.
- Topik #4 → Mengacu kepada berita pesawat Jokowi sempat berputar-putar di Turki dan juga mengenai pengungkapan tersangka kasus dugaan korupsi penyewaan pesawat

di PT Garuda Indonesia.

- Topik #5 → Mengacu kepada berita seputar mudik lebaran, seperti wagub DKI yang mengingatkan warga untuk taat protokol pada saat mudik lebaran, foto mudik, hingga upaya yang dilakukan oleh polda agar warga tak mudik.

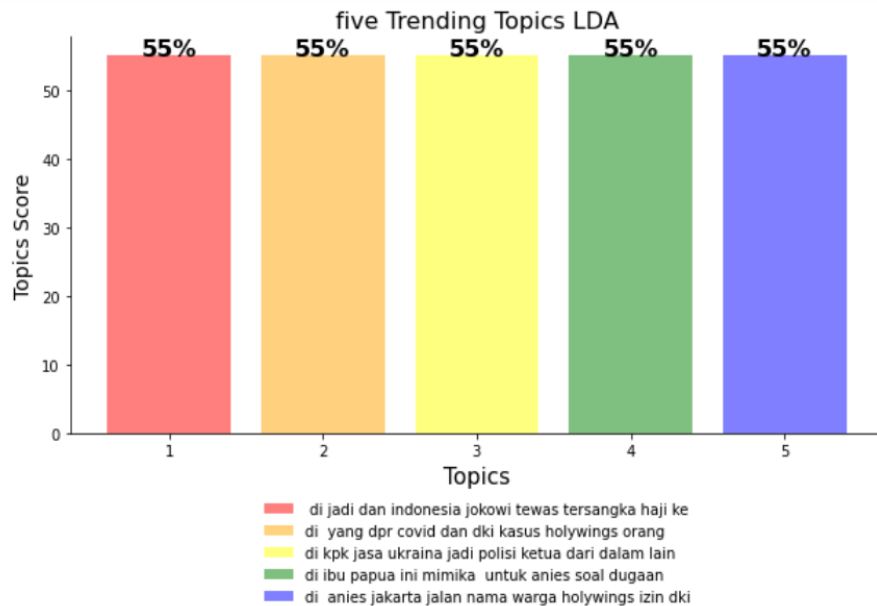


GAMBAR 29 DIAGRAM TOP 5 TOPIC WEBSITE GIZMOLOGI

Pada website ini, walaupun terlihat hasilnya sama, namun korelasi antar topiknya lebih kuat dibandingkan website lainnya. Berikut merupakan korelasi dari 5 pembagian topik teratas pada *website Gizmologi* :

- Topik #1 → Mengacu pada berita Telkomsel menyediakan layanan 5G di MotoGP Mandalika, hal ini dilakukan bersama Kominfo. Selain itu juga, terdapat pameran seperti Cloud Gaming.
- Topik #2 → Mengacu pada berita kominfo mencatat tantangan yang berkaitan dengan UMKM, pemerintah dorong kolaborasi untuk digitalisasi UMKM, hingga berita mengenai kegiatan bantu UMKM menggunakan digitalisasi yang sudah diterapkan.
- Topik #3 → Mengacu pada berita mengenai XL Axiata yang kembali gandeng Juniper Networks dalam Pengembangan 5G di Indonesia, dan juga berita mengenai XL Axiata serta Indosat lainnya.
- Topik #4 → Mengacu pada berita Kominfo menggandeng huawei dan menyiapkan infrastruktur 5G di IKN, serta berita yang berkaitan dengan smartphone dengan merek oppo dan realme.

- Topik #5 → Mengacu pada berita tentang UltraDart Realme yang memiliki teknologi FastCharging, ini mencakup spesifikasinya dan keunggulannya.



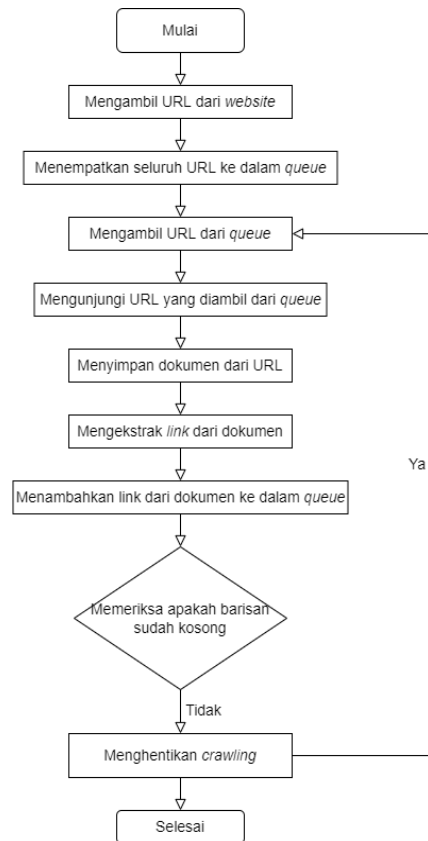
GAMBAR 30 DIAGRAM TOP 5 TOPIC WEBSITE IDNTIMES

Untuk website idntimes juga hasilnya sama, tidak ada topik yang lebih unggul daripada yang lainnya. Namun, untuk data yang ini memang masih terdapat data yang belum berhasil dibersihkan, dikarenakan jenis datanya sulit untuk dianalisis menggunakan library-library tertentu. Adapun untuk korelasi antara kata-kata dalam tiap topiknya adalah sebagai berikut :

- Topik #1 → Tidak mengacu pada berita tertentu, namun beberapa berita yang muncul di antaranya : 6 orang jadi tersangka kasus anak 12 tahun curi hp, kunjungan Jokowi ke Ukraina-Rusia, total 9 calon haji Indonesia meninggal dunia di Arab Saudi
- Topik #2 → Mengacu pada berita mengenai promo miras di Holywings (yang dikenal dengan kasus Holywings). Pemprov DKI berikan teguran keras akan kasus tersebut.
- Topik #3 → Tidak mengacu pada berita tertentu, namun beberapa berita yang muncul di antaranya : Polri kuasai institusi dari BIN, KPK, hingga Kemendagri, Nasdem puji keberanian Presiden Jokowi kunjungi Ukraina, dan KPK bakal panggil kembali Lasmi Indaryani
- Topik #4 → Tidak mengacu pada berita tertentu, namun mengacu pada berita dengan topik tertentu, yaitu yang berlokasi atau berkaitan dengan Mimika, salah satu Kabupaten di Papua. Salah satu contoh beritanya adalah Ibu Kota Papua Tengah di Nabire, Warga Ancam Tutup Freeport.
- Topik #5 → Mengacu pada berita Anies Baswedan selaku Gubernur DKI mencabut izin usaha seluruh kafe Holywings di Jakarta, ada beberapa berita yang muncul, mulai

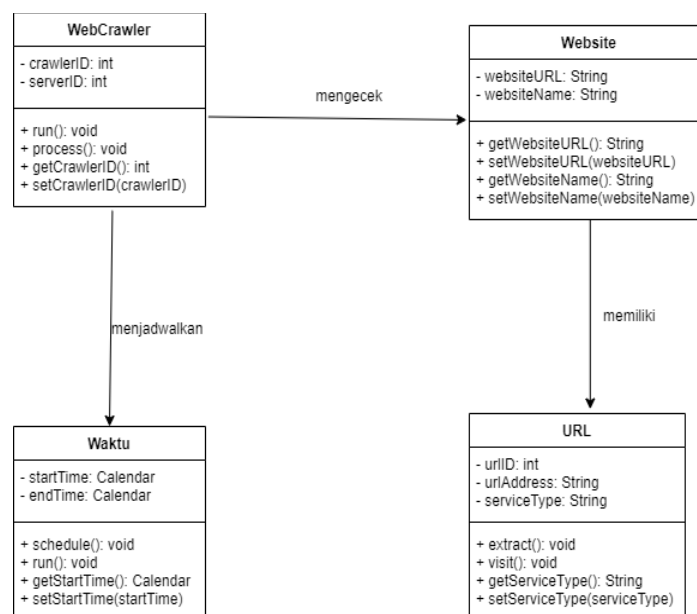
dari Anies banjir pujian, hingga alasan Anies mencabut izin tersebut.

I. Flow Chart



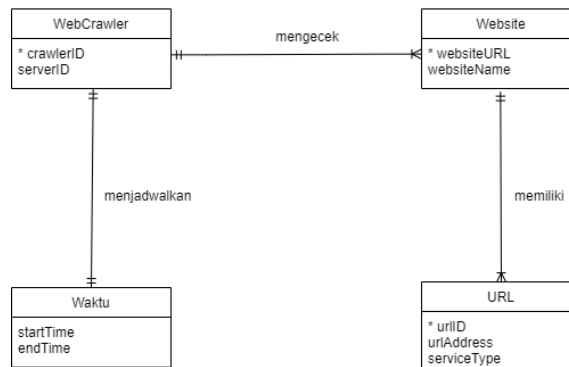
GAMBAR 31 FLOW CHART CRAWLING DATA DARI WEBSITE

J. Unified Modelling Language (UML)



GAMBAR 32 UML DARI WEBSITE

K. Entity Relationship Diagram (ERD)



GAMBAR 33 ERD DARI WEBSITE

GAMBAR 15

IV.3 DATASET INTERNAL

A. Hasil Pengambilan Data dari Dataset Internal DailySocial.id

JSON	Raw Data	Headers
Save	Copy	Collapse All Expand All (slow) Filter JSON
success:	true	
count:	42874	
data:		
0:		
title:	"Cara Mendaftarkan Diri sebagai Agen di Aplikasi Super"	
subtitle:	"Super menghadirkan harga sembako yang terjangkau di seluruh pelosok Indonesia. Simak cara menjadi Super Agen berikut ini"	
id:	460225	
type:	"post"	
slug:	"daftar-agen-di-super"	
url:	"https://dailysocial.id/post/daftar-agen-di-super"	
status:	"publish"	
date:	"2022-06-28 19:35:07+00:00"	
is_premium:	false	
tier_pricing:	null	
author:	(-)	
categories:	(-)	
tags:	(-)	
image:	(-)	
1:	(-)	
2:	(-)	
3:	(-)	
4:	(-)	
5:	(-)	
6:	(-)	
7:	(-)	
8:	(-)	
9:	(-)	

GAMBAR 34 HASIL PENGAMBILAN DATASET INTERNAL DARI API DAILYSOCIAL.ID

Pihak *DailySocial.id* sendiri memiliki sebuah API yang wajib dijaga kerahasiaannya. Peneliti diwajibkan untuk memahami konten NDA (*Non Disclosure Agreement*) sebagai perjanjian untuk menjaga kerahasiaan data. Setelah itu diberikan sebuah *link* API yang akan dijadikan sebagai bahan dari dataset internal. Peneliti membatasi pengambilan data sebanyak 500 buah dengan perkiraan pengambilan data dari bulan Januari hingga April 2022. Melalui dataset internal ini, peneliti akan mencoba menganalisis JSON yang ada di dalamnya dan

memanfaatkannya sebagai salah satu hal untuk media monitoring dan menjadi pembanding dengan *website* kompetitor dan juga Twitter API.

B. Preprocessing Dataset Internal

```
In [5]: title = np.char.lower(title)

In [6]: import re
marks = "!\"#$%&'()*+,-./:;<=>@[\\]^_`{|}~\n"
for x in marks:
    title = np.char.replace(title, x, "")

In [7]: title = repr(title)
title = re.sub(r"^[a-zA-Z]", " ", title)

In [8]: # remove the whitespaces or specific characters from the string at the beginning and end of the string
title = title.strip()

In [9]: title = re.sub(r"\s+", " ", title)

In [10]: from nltk.tokenize import word_tokenize
title_tokens = word_tokenize(title)
print(title_tokens)

['array', 'ide', 'peluang', 'usaha', 'di', 'daerah', 'pedesaan', 'semakin', 'lengkap', 'bank', 'digital', 'jenius', 'mudahka', 'n', 'pengguna', 'bayar', 'zakat', 'dan', 'sedekah', 'online', 'iprice', 'bantu', 'umkm', 'asia', 'tenggara', 'go', 'online', 'melalui', 'program', 'iprice', 'sellers', 'club', 'saturdays', 'umumkan', 'pendanaan', 'seri', 'a', 'dipimpin', 'oleh', 'alt', 'ara', 'ventures', 'potensi', 'layanan', 'ecommerce', 'dukung', 'pemulihan', 'ekonomi', 'di', 'indonesia', 'bukalapak', 'kanto', 'ngi', 'laba', 'bersih', 'triliun', 'rupiah', 'di', 'kuartal', 'pertama', 'traveloka', 'terus', 'perluas', 'kerja', 'sama', 'd', 'agan', 'pembankan', 'laba', 'sama', 'talker', 'budidkan', 'laksanasi', 'digitalisasi', 'indonesia', 'hai', 'gandang', 'sa']
```

GAMBAR 35 PREPROCESSING DATASET INTERNAL

Gambar 35 merupakan contoh *preprocessing* data yang dilakukan. Dataset yang berasal dari *website DailySocial.id* masih harus dibersihkan karena meskipun kebanyakan sudah memakai kata-kata baku seperti yang umum ditemukan dalam website yang memuat kumpulan artikel, namun masih ada beberapa kata seperti kata penghubung juga imbuhan awalan dan akhiran yang tidak memiliki makna penting tetapi tingkat kemunculannya tinggi. Oleh karena itu, beberapa pembersihan data yang dilakukan meliputi:

- *Lowercase* → selama pemrosesan teks, setiap kalimat terbagi ke dalam kata-kata dan tiap kata dianggap sebagai token setelah *preprocessing*. Bahasa pemrograman menganggap data teks sebagai data yang sensitif, yang berarti kata yang sama namun dengan penulisan huruf besar dan huruf kecil yang berbeda dianggap tidak sama. Proses ini membuat setiap kata yang memiliki huruf besar menjadi huruf kecil dengan menggunakan *numpy*.
- *Remove Punctuation* → tanda baca merupakan simbol yang tak berarti dalam *corpus document*. Oleh karena itu peneliti menyimpan setiap tanda baca yang mungkin muncul ke dalam suatu variabel dan melakukan iterasi untuk menghilangkan tanda baca tersebut.
- *Remove Number and Whitespaces* → sama seperti tanda baca, angka dan *whitespace* pun tak memiliki arti penting dalam suatu dokumen. Disini peneliti menggunakan *regex* untuk menghilangkan angka dan juga *whitespace* yang terdapat dalam dokumen.

- Tokenisasi → tokenisasi merupakan pembagian teks ke dalam *token* atau *unit* yang lebih kecil. Hal ini dilakukan sebagai dasar agar peneliti dapat mengembangkan model yang baik dan membantu pemahaman mengenai teks yang dimiliki. Peneliti menggunakan *library* NLTK untuk melakukan tokenisasi.
- *Stopwords* → disini peneliti menggunakan *library* NLTK untuk menghilangkan kata-kata yang tidak bermakna namun sering ditemukan dalam dokumen. Daftar dari *stopwords* yang digunakan diambil dari Bahasa Indonesia, contohnya kata-kata seperti ‘dan’, ‘aku’, ‘dari’, ‘adalah’, dan lain-lain.

C. Contoh Implementasi Kode untuk Penerapan Algoritma TF-IDF

Out[14]:

	term	rank
338	pendanaan	79.0
53	bisnis	69.0
434	startup	60.0
173	indonesia	57.0
407	rupiah	54.0
...
353	perkenalkan	2.0
354	perkreditan	2.0
358	peroleh	2.0
359	perolehan	2.0
0	ac	2.0

500 rows × 2 columns

GAMBAR 36 IMPLEMENTASI KODE UNTUK TF-IDF

Gambar 36 merupakan contoh implementasi algoritma TF-IDF. Disini sebanyak 500 *term* akan diurutkan mulai dari yang memiliki *term frequency* terbesar hingga terkecil untuk mendapatkan top 500 *term* dengan *term frequency* terbesar. Sebagai contoh kata “pendanaan” memiliki *term frequency* terbesar yaitu 79.0 yang berarti kemunculan kata ini lebih banyak dibandingkan dengan kata lainnya. Semakin sering *term* tertentu muncul, maka artinya akan semakin penting atau relevan *term* tersebut.

D. Word Cloud



GAMBAR 37 WORD CLOUD

Gambar 37 menunjukkan hasil visualisasi dari word cloud. Word cloud sendiri berfungsi sebagai representasi visual dari kata-kata yang sering digunakan di dalam dokumen. Kata yang paling besar menunjukkan bahwa semakin sering kata itu digunakan. Sebagai contoh kata 'bisnis', 'indonesia', dan 'startup' merupakan kata yang lebih besar ukurannya dibanding dengan kata-kata lain seperti misalnya 'bank', 'aplikasi', 'laba', dll. Hal ini menunjukkan bahwa kata-kata seperti 'bisnis', 'indonesia', dan 'startup' lebih sering digunakan dibanding dengan yang lain.

E. Bag of Words

```
import gensim
import gensim.corpora as corpora

# Membuat Dictionary
dictionary = corpora.Dictionary([title])

# Membuat Corpus
titles = title

# Term Document Frequency
corpus = [dictionary.doc2bow(title) for text in titles]

# print
print(corpus[1][0][:30])

[(0, 1), (1, 19), (2, 1), (3, 1), (4, 2), (5, 1), (6, 1), (7, 4), (8, 1), (9, 1), (10, 2), (11, 1), (12, 1), (13, 16), (14, 2), (15, 1), (16, 2), (17, 1), (18, 1), (19, 2), (20, 4), (21, 3), (22, 1), (23, 2), (24, 1), (25, 1), (26, 1), (27, 1), (28, 1), (29, 3)]
```

GAMBAR 38 IMPLEMENTASI KODE UNTUK BAG OF WORDS

Objek yang sudah ditokenisasi dalam preprocessing dataset diubah menjadi *corpus* dan *dictionary*. Dictionary diperlukan untuk membuat sebuah corpus, dimana corpus ini termasuk ke dalam metode bag of words yang merupakan salah satu cara untuk melakukan pembobotan kata yang sering muncul dalam dokumen. Caranya yaitu adalah dengan mengubah data teks menjadi angka atau vektor. Metode lainnya yg juga telah digunakan di dalam project ini yaitu TF-IDF. Disini peneliti melakukan print data *corpus* sebanyak 30 buah untuk mengetahui beberapa hasil pembobotan kata yang sering muncul dalam dokumen.

F. Implementasi LDA

LDA

```
from pprint import pprint
# number of topics
num_topics = 10

# Build LDA model
lda_model = gensim.models.LdaMulticore(corpus=corpus,
                                       id2word=dictionary,
                                       num_topics=num_topics)

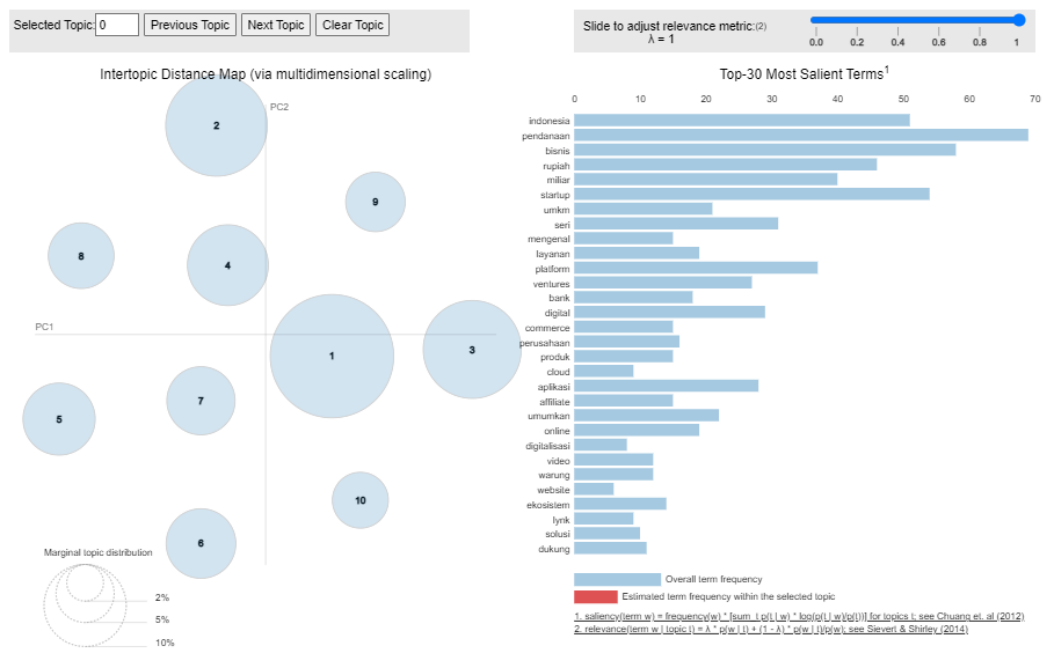
# Print the Keyword in the 10 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.019*"indonesia" + 0.019*"bisnis" + 0.018*"pendanaan" + 0.015*"rupiah" + '
  '0.014*"startup" + 0.014*"miliar" + 0.011*"platform" + 0.010*"digital" + '
  '0.009*"seri" + 0.008*"ventures"'),
 (1,
  '0.023*"pendanaan" + 0.017*"bisnis" + 0.015*"indonesia" + 0.014*"startup" + '
  '0.014*"miliar" + 0.012*"platform" + 0.011*"rupiah" + 0.010*"aplikasi" + '
  '0.008*"seri" + 0.008*"digital"'),
 (2,
  '0.021*"startup" + 0.021*"bisnis" + 0.019*"pendanaan" + 0.013*"indonesia" + '
  '0.013*"miliar" + 0.012*"rupiah" + 0.012*"platform" + 0.010*"digital" + '
  '0.008*"b" + 0.008*"umumkan"'),
 (3,
  '0.020*"pendanaan" + 0.020*"bisnis" + 0.014*"indonesia" + 0.014*"startup" + '
  '0.013*"platform" + 0.012*"miliar" + 0.012*"rupiah" + 0.010*"digital" + '
  '0.008*"seri" + 0.007*"aplikasi"'),
 (4,
  '0.021*"pendanaan" + 0.017*"bisnis" + 0.016*"startup" + 0.013*"miliar" + '
  '0.013*"rupiah" + 0.012*"indonesia" + 0.011*"digital" + 0.011*"seri" + '
  '0.011*"platform" + 0.009*"ventures"'),
 (5,
  '0.019*"indonesia" + 0.019*"pendanaan" + 0.017*"bisnis" + 0.016*"startup" + '
  '0.014*"rupiah" + 0.011*"platform" + 0.010*"miliar" + 0.009*"seri" + '
  '0.009*"ventures" + 0.009*"aplikasi"'),
 (6,
  '0.022*"pendanaan" + 0.020*"bisnis" + 0.019*"rupiah" + 0.017*"startup" + '
  '0.013*"indonesia" + 0.012*"miliar" + 0.010*"platform" + 0.010*"seri" + '
  '0.008*"digital" + 0.007*"aplikasi"'),
 (7,
  '0.022*"pendanaan" + 0.019*"bisnis" + 0.016*"startup" + 0.014*"indonesia" + '
  '0.014*"miliar" + 0.011*"rupiah" + 0.011*"platform" + 0.008*"b" + '
  '0.008*"ventures" + 0.008*"umumkan"'),
 (8,
  '0.021*"pendanaan" + 0.018*"bisnis" + 0.016*"rupiah" + 0.016*"indonesia" + '
  '0.014*"startup" + 0.014*"miliar" + 0.011*"platform" + 0.011*"aplikasi" + '
  '0.009*"digital" + 0.008*"umkm"'),
 (9,
  '0.026*"pendanaan" + 0.017*"bisnis" + 0.016*"startup" + 0.015*"rupiah" + '
  '0.014*"indonesia" + 0.012*"platform" + 0.010*"digital" + 0.010*"miliar" + '
  '0.008*"seri" + 0.008*"aplikasi"')]
```

GAMBAR 39 IMPLEMENTASI KODE UNTUK LDA

LDA (*Latent Dirichlet Allocation*) merupakan sebuah contoh dari topic model dan digunakan untuk mengklasifikasi teks dalam sebuah dokumen ke dalam suatu topik khusus. LDA akan membangun sebuah topik per model dokumen dan kata-kata per *topic model*. Disini penulis mengambil 10 topik dan memasukkannya ke dalam parameter. Masing-masing *keyword* dari 10 topik tersebut akan memiliki bobot tertentu dalam suatu topik. Semakin tinggi bobot dari suatu kata dalam suatu topik, maka lebih besar relevansi kata tersebut dalam dokumen atau topik yang bersangkutan.

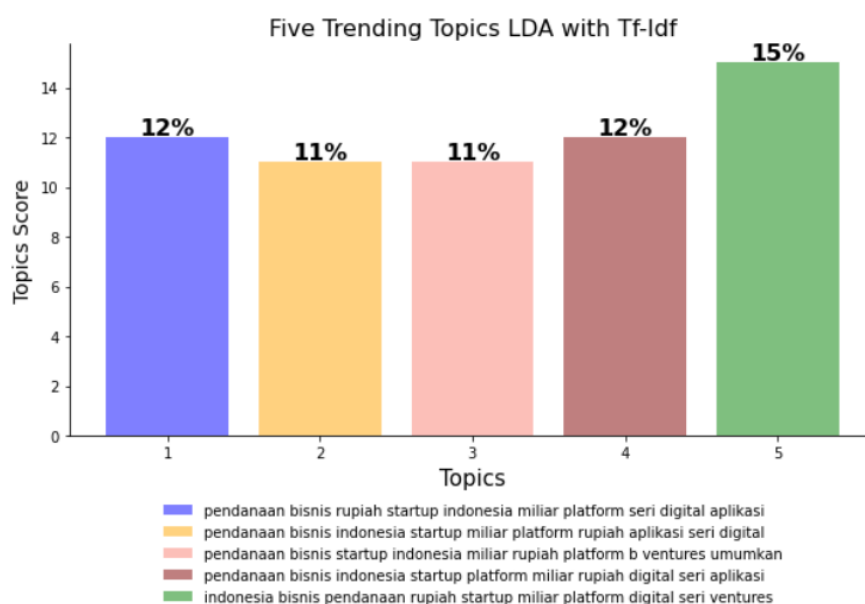
G. Visualisasi LDA dengan PyLDAvis



GAMBAR 40 PyLDAvis

Visualisasi dari LDA dilakukan untuk memvisualisasikan topik agar interpretasi terhadap topik dapat lebih mudah dilakukan. Modul dari pyLDAvis dapat digunakan untuk lebih memahami topik individual dan juga relasi antar topik-topiknya. Semakin tinggi *term frequency* pada suatu kata, maka semakin relevan kata itu dalam suatu topik. Jika jarak antar lingkaran semakin dekat, maka ada kemungkinan topik-topik tersebut akan semakin berhubungan.

H. Bar Diagram



GAMBAR 41 BAR DIAGRAM

Bar diagram ini digunakan untuk melihat hasil visualisasi pendistribusian topik dari LDA. Disini peneliti mengambil 5 buah topik dan hasilnya menunjukkan bahwa kelima topik yang didapat memiliki hasil interpretasi yang mirip yaitu mengenai pendanaan bisnis *startup* di Indonesia.

BAB V

UJI COBA SEDERHANA

V.1 DATASET TWITTER

```

Testing Model With Unseen Document

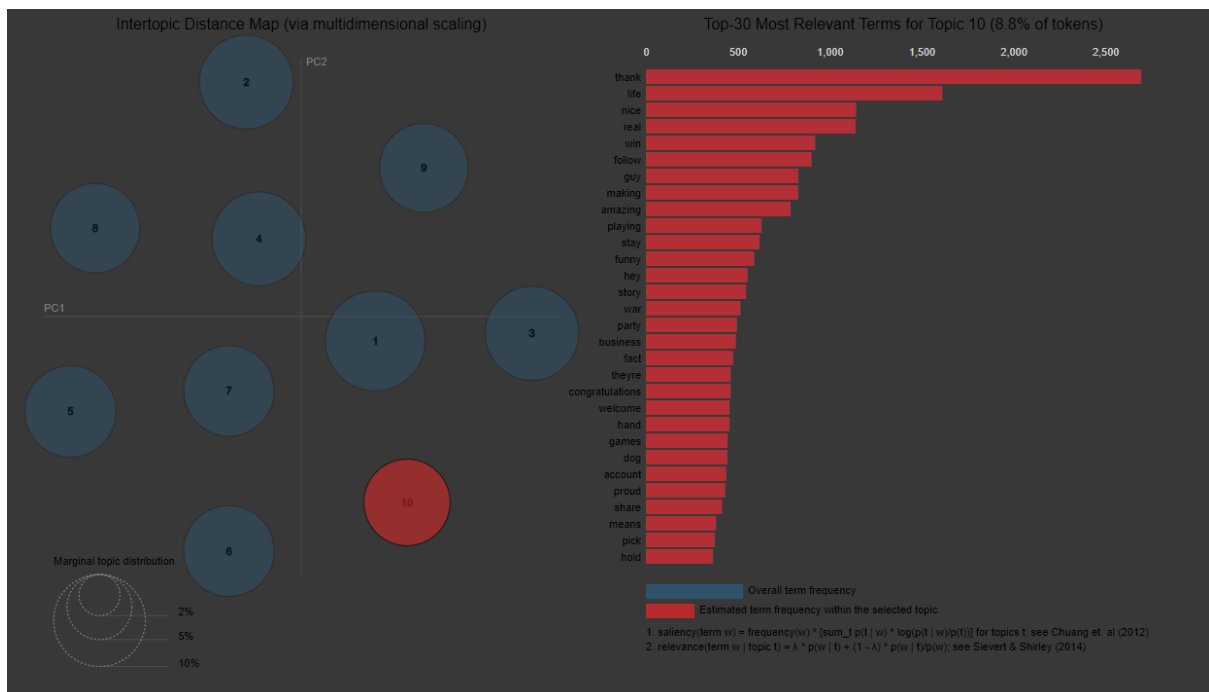
[458] unseen_document = 'How machine learning is used in healthcare'
bow_vector = dictionary.doc2bow(preprocess(unseen_document))
for index, score in sorted(lda_model_tfidf[bow_vector], key=lambda tup: -1*tup[1]):
    print("Score: {} \t Topic: {}".format(score, lda_model_tfidf.print_topic(index, 5)))

Score: 0.4887419044971466      Topic: 0.003*cool + 0.002*wanna + 0.002*battl + 0.002*fact + 0.002*video
Score: 0.2945913076480757      Topic: 0.002*send + 0.002*definit + 0.002*join + 0.002*project + 0.002*like
Score: 0.0166667178273281      Topic: 0.005*happi + 0.004*fuck + 0.003*morn + 0.003*best + 0.003*thing
Score: 0.016666695475578388     Topic: 0.004*cute + 0.003*sorri + 0.002*guy + 0.002*that + 0.002*march
Score: 0.01666668882499771      Topic: 0.003*april + 0.002*go + 0.002*congratul + 0.002*fool + 0.002*strong
Score: 0.01666668574417114      Topic: 0.004*happi + 0.004*cav + 0.003*birthday + 0.003*hope + 0.003*final
Score: 0.016666678711771965     Topic: 0.003*love + 0.002*manifest + 0.002*nigga + 0.002*crazi + 0.002*peopl
Score: 0.016666676849126816     Topic: 0.003*shit + 0.003*sleep + 0.003*life + 0.003*game + 0.002*mate
Score: 0.016666673123836517     Topic: 0.010*wordl + 0.004*look + 0.003*enhyphen + 0.003*enhyphenmemb + 0.003*tweet
Score: 0.016666673123836517     Topic: 0.003*post + 0.003*cri + 0.003*photo + 0.003*pretti + 0.002*say
Score: 0.016666673123836517     Topic: 0.003*follow + 0.002*bitch + 0.002*read + 0.002*feel + 0.002*welcom
Score: 0.016666671261191368     Topic: 0.004*thank + 0.002*friend + 0.002*yearn + 0.002*absolut + 0.002*mood
Score: 0.016666671261191368     Topic: 0.003*stop + 0.003*care + 0.003*better + 0.002*amen + 0.002*worth
Score: 0.01666666753598107      Topic: 0.009*brstw + 0.004*listen + 0.003*babl + 0.003*funni + 0.003*jungkook
Score: 0.01666666753598107      Topic: 0.012*morn + 0.004*beautil + 0.003*leav + 0.002*dont + 0.002*turn
time: 89.1 ms (started: 2022-06-30 17:03:28 +00:00)

```

GAMBAR 42 TESTING MODEL DENGAN DOKUMEN BARU

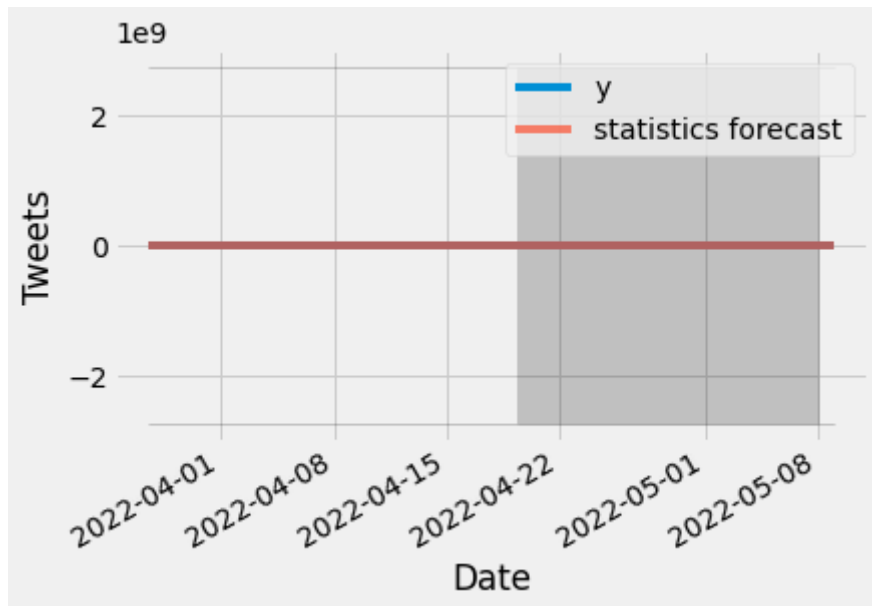
Gambar 42 merupakan hasil testing model menggunakan dokumen yang belum pernah dilihat. Pengujian ini dilakukan untuk mencoba menguji apakah model tersebut dapat dengan benar-benar melakukan pembobotan (TF-IDF) pada sebuah keyword yang diberikan. Pada kasus ini keyword yang diberikan adalah “How machine learning is used in healthcare” yang akan diberikan bobot sesuai dengan masing-masing kata.



GAMBAR 43 TESTING CLUSTERING BERDASARKAN KEDEKATAN INTERTOPIC

Gambar 43 merupakan hasil testing program mencoba mencari kedekatan intertopic (topik-topik yang ada di dalam setiap cluster) terhadap seluruh dokumen yang diberikan. Blok

berwarna merah merupakan estimasi term frequency dengan topik yang dipilih, sedangkan blok berwarna biru merupakan term frequency keseluruhan dokumen.



GAMBAR 44 HASIL FORECASTING

Gambar 44 merupakan hasil *forecasting* yang berhasil dilakukan, dimulai dari tanggal tanggal 8 Mei 2022. Pendekatan model yang dilakukan menggunakan SARIMAX. Walaupun pada penelitian ini hasil *forecasting* masih belum tepat akan tetapi program sudah berjalan dengan baik. Kemungkinan hal ini dikarenakan pendekatan *clustering* yang belum tepat.

V.2 DATASET WEBSITE

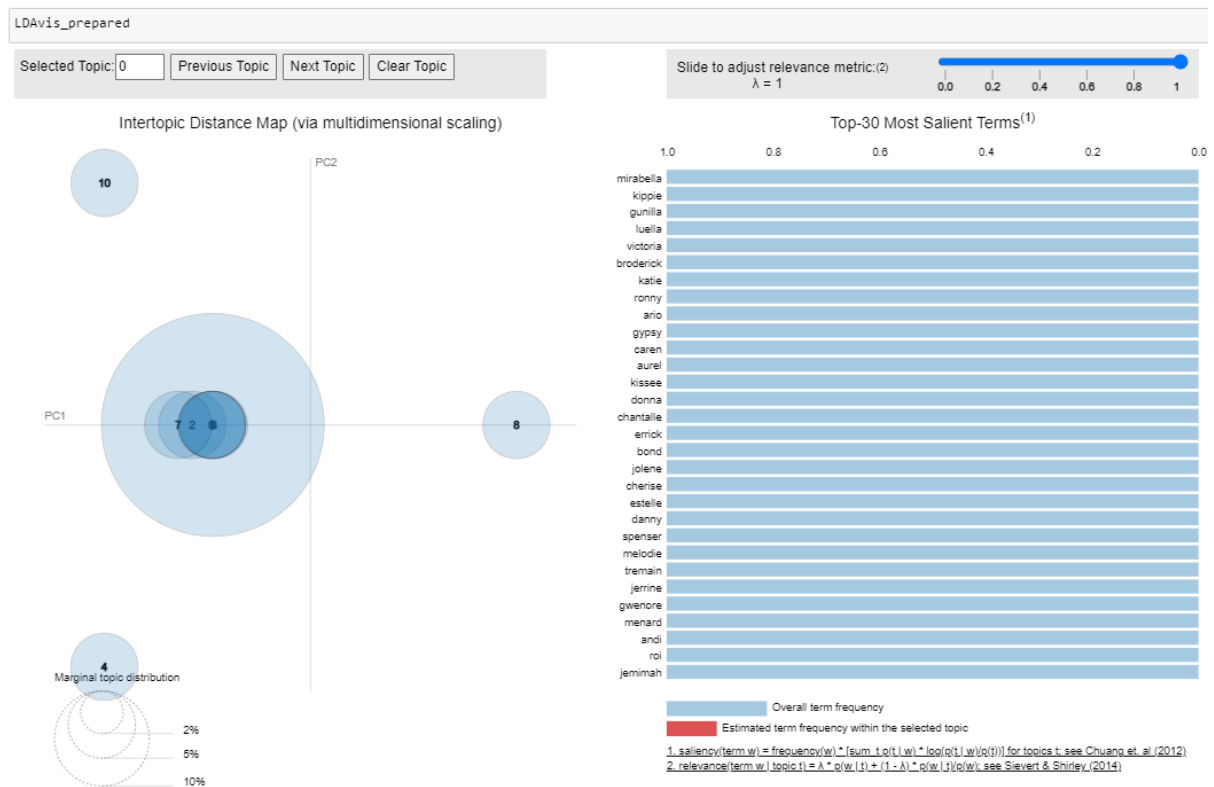
```
# Running and Training LDA model on the document term matrix
lda_model = Lda(doc_term_matrix, num_topics=total_topics, id2word = dictionary, passes = 50)
lda_model.show_topics(num_topics=total_topics, num_words=number_words)

[(0,
  '0.021*luella' + 0.021*gunilla' + 0.011*heddi' + 0.011*antoni' + 0.011*catharine' + 0.011*tana' + 0.011*lincol' + 0.011*suzanne' + 0.011*yelena' + 0.011*enni
s'''),
 (1,
  '0.012*eadar' + 0.012*jolene' + 0.012*estelle' + 0.012*ky' + 0.012*avictor' + 0.012*siana' + 0.012*gwenore' + 0.012*ario' + 0.012*duff' + 0.012*regan'''),
 (2,
  '0.011*marc' + 0.011*meyer' + 0.011*chere' + 0.011*toiboid' + 0.011*cate' + 0.011*clare' + 0.011*sasha' + 0.011*bard' + 0.011*renata' + 0.011*dorita'''),
 (3,
  '0.011*lvia' + 0.011*jill' + 0.011*rae' + 0.011*arvy' + 0.011*fernandina' + 0.011*onfroi' + 0.011*art' + 0.011*cece' + 0.011*deanne' + 0.011*kalila'''),
 (4,
  '0.011*malinde' + 0.011*chrisie' + 0.011*sandye' + 0.011*gabriele' + 0.011*cheslie' + 0.011*itch' + 0.011*aigneis' + 0.011*hannis' + 0.011*frieda' + 0.011*ervi
n'''),
 (5,
  '0.020*victoria' + 0.011*corri' + 0.011*melany' + 0.011*merralee' + 0.011*hyman' + 0.011*emelen' + 0.011*noel' + 0.011*kimble' + 0.011*nollie' + 0.011*dominiqu
e'''),
 (6,
  '0.020*broderick' + 0.020*katie' + 0.011*kenyon' + 0.011*kyllila' + 0.011*tedi' + 0.011*dory' + 0.011*dame' + 0.011*erasmus' + 0.011*thomas' + 0.011*mirilla'''),
 (7,
  '0.022*miraella' + 0.022*kipie' + 0.011*ingemar' + 0.011*clarabelle' + 0.011*virginia' + 0.011*prue' + 0.011*marylee' + 0.011*joey' + 0.011*goldie' + 0.011*aud
re'''),
 (8,
  '0.022*olivero' + 0.022*cy' + 0.012*ruthie' + 0.012*elbertine' + 0.012*clyve' + 0.012*odella' + 0.012*cass' + 0.012*fonzie' + 0.012*chandler' + 0.012*cherily
n'''),
 (9,
  '0.011*hamish' + 0.011*rosanne' + 0.011*adina' + 0.011*ivan' + 0.011*gui' + 0.011*bernadene' + 0.011*cornelia' + 0.011*ruperto' + 0.011*hurleigh' + 0.011*amelit
a'')]
```

GAMBAR 45 TESTING MODEL DENGAN DATA BARU

Gambar tersebut merupakan hasil testing dengan dokumen baru. Dokumen yang penulis gunakan adalah dokumen mengenai daftar id dan nama siswa. Kolom yang penulis gunakan di sini adalah bagian kolom *first_name*. Nama-nama tersebut sudah dibagi ke 10 topik yang

masingnya terdiri dari 10 data. Sebenarnya data ini bukanlah data yang baik untuk diteliti dikarenakan data nama biasanya merupakan sesuatu yang unik bukan seperti data berita yang memiliki kesinambungan satu dengan yang lainnya, namun itulah yang ingin penulis lihat, bagaimana model tersebut bekerja jika datanya sangat unik.



GAMBAR 46 TESTING DENGAN PYLDAVIS

Pada gambar di atas dapat dilihat, jika kita menggunakan data yang tidak baik, yaitu data yang sepenuhnya unik, maka akan menghasilkan pembobotan yang sama persis untuk setiap katanya dan juga visualisasi pembagian topiknya tidak menunjukkan hasil yang baik.

V.3 DATASET INTERNAL

```

from pprint import pprint
# number of topics
num_topics = 10

# Build LDA model
lda_model = gensim.models.LdaMulticore(corpus=corpus,
                                       id2word=dictionary,
                                       num_topics=num_topics)

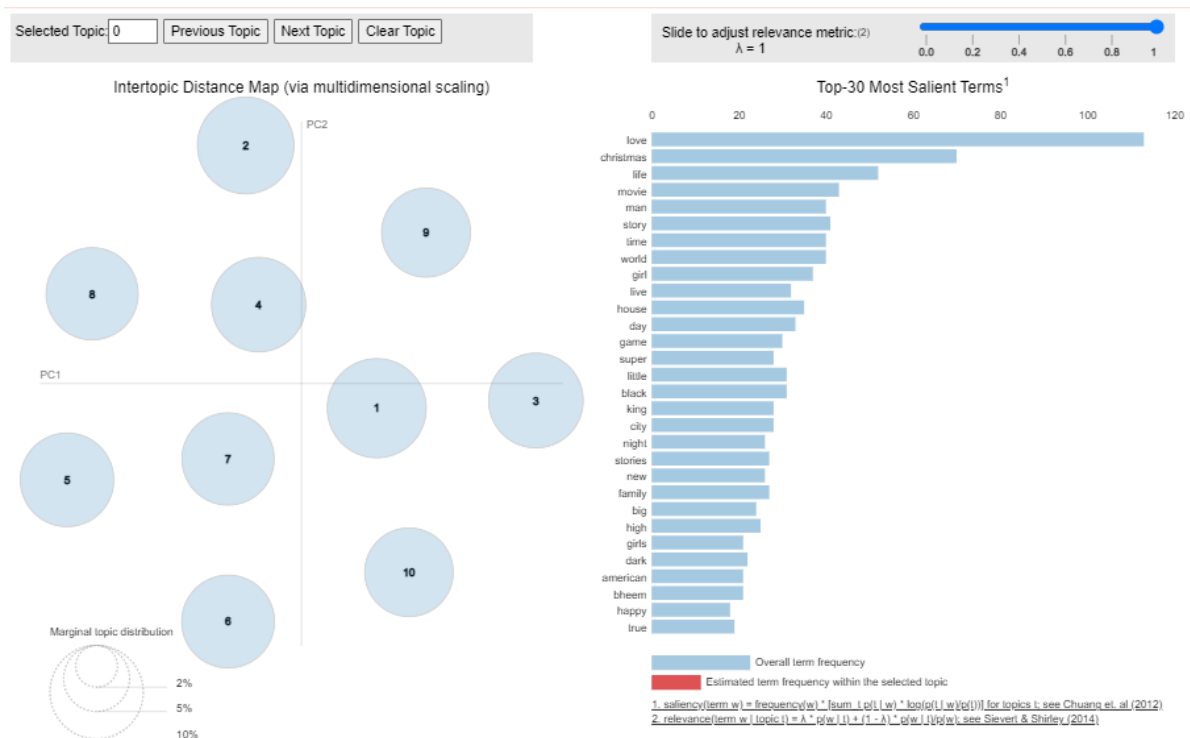
# Print the Keyword in the 10 topics
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]

[(0,
  '0.045*"taxi" + 0.045*"mighty" + 0.045*"parties" + 0.045*"festival" + '
  '0.045*"came" + 0.045*"kite" + 0.045*"five" + 0.045*"little" + '
  '0.045*"driver" + 0.045*"shadow"'),
 (1,
  '0.050*"little" + 0.049*"grail" + 0.049*"festival" + 0.048*"shadow" + '
  '0.048*"five" + 0.048*"bheem" + 0.048*"driver" + 0.048*"taxi" + '
  '0.048*"python" + 0.048*"holy"'),
 (2,
  '0.045*"dtype" + 0.045*"holy" + 0.045*"festival" + 0.045*"monty" + '
  '0.045*"came" + 0.045*"five" + 0.045*"parties" + 0.045*"grail" + '
  '0.045*"driver" + 0.045*"shadow"'),
 (3,
  '0.053*"taxi" + 0.051*"five" + 0.051*"U" + 0.050*"monty" + 0.050*"holy" + '
  '0.050*"came" + 0.049*"clash" + 0.048*"festival" + 0.048*"kite" + '
  '0.047*"array"'),
 (4,
  '0.048*"holy" + 0.048*"grail" + 0.047*"reference" + 0.047*"little" + '
  '0.046*"five" + 0.046*"taxi" + 0.046*"festival" + 0.046*"parties" + '
  '0.046*"bheem" + 0.046*"python"'),
 (5,
  '0.053*"back" + 0.052*"films" + 0.050*"python" + 0.049*"reference" + '
  '0.047*"shadow" + 0.046*"mighty" + 0.046*"U" + 0.046*"dtype" + 0.046*"taxi" + '
  '0.046*"kite"'),
 (6,
  '0.049*"films" + 0.048*"back" + 0.048*"mighty" + 0.047*"driver" + '
  '0.047*"kite" + 0.047*"U" + 0.047*"came" + 0.046*"array" + 0.046*"python" + '
  '0.046*"dtype"'),
 (7,
  '0.050*"festival" + 0.048*"bheem" + 0.048*"U" + 0.046*"mighty" + '
  '0.046*"python" + 0.046*"reference" + 0.046*"little" + 0.046*"came" + '
  '0.046*"five" + 0.045*"array"'),
 (8,
  '0.050*"grail" + 0.049*"driver" + 0.049*"parties" + 0.048*"little" + '
  '0.048*"bheem" + 0.046*"came" + 0.046*"festival" + 0.046*"shadow" + '
  '0.046*"holy" + 0.046*"mighty"'),
 (9,
  '0.050*"holy" + 0.050*"clash" + 0.050*"grail" + 0.049*"parties" + '
  '0.047*"monty" + 0.046*"shadow" + 0.046*"reference" + 0.046*"little" + '
  '0.046*"array" + 0.046*"bheem"')]

```

GAMBAR 47 TESTING MODEL DENGAN DOKUMEN BARU

Gambar di atas merupakan hasil *testing* dengan dokumen baru, disini peneliti mengambil dokumen mengenai tayangan *film* dan *TV show* di *Netflix*. Kolom yang digunakan yaitu *title*, yang berisikan judul-judul dari *film*. Pengujian dilakukan dengan mengambil 10 topik dan dapat dilihat bahwa pembobotan setiap term pada tiap dokumen telah berjalan dengan baik.



GAMBAR 48 TESTING CLUSTERING BERDASARKAN KEDEKATAN INTERTOPIC

Gambar ini menunjukkan *testing* berdasarkan kedekatan antar topik dan relevansi beberapa *term* yang sering ditemukan dalam masing-masing topik. Bar di sebelah kanan menunjukkan seberapa sering suatu kata muncul dalam suatu dokumen.

BAB VI

KESIMPULAN DAN SARAN

VI.1 KESIMPULAN

Pada penelitian ini, telah dilakukan pengumpulan dataset dari Twitter, Website kompetitor, dan dataset internal. Kemudian, setelah dataset terkumpul, kumpulan data tersebut diberlakukan pembersihan data seperti data-data yang tidak bermakna, dan data-data yang bernilai duplikat maupun kosong. Selanjutnya data-data tersebut dilakukan pembobotan kata dengan menggunakan TF-IDF agar bisa mengetahui seberapa penting suatu term atau kata tersebut dalam *corpus* atau dokumen. Selanjutnya dengan membangun pipeline untuk dilakukan klasterisasi menggunakan pendekatan *Latent Dirichlet Allocation* dan juga K-Means. Setelah menjadi klaster untuk setiap topik-topik yang mempunyai kedekatan kata, dilakukan analisis setiap topik yang terdapat pada masing-masing kluster beserta memperhatikan *scoring* pada setiap kluster untuk mengecek seberapa dominan kata tersebut dalam sebuah dokumen. Selanjutnya dilakukan forecasting dengan pendekatan SARIMAX.

VI.2 SARAN

Pada penelitian ini masih terdapat beberapa kekurangan, oleh karena itu beberapa saran bagi peneliti selanjutnya:

1. Untuk menentukan jumlah cluster yang ada pada penelitian ini diambil secara acak, artinya jumlah cluster yang dihasilkan belum tentu optimal, oleh karena itu alangkah lebih baik untuk menentukan jumlah cluster dapat menggunakan pendekatan *elbow method* ataupun *silhouette*.
2. Apabila melihat hasil dari *topic modeling* masih terdapat kata-kata yang *meaningless*, oleh karena itu alangkah baiknya untuk melakukan pembersihan data lebih baik lagi, misalnya dengan memanfaatkan regex.
3. Data yang terbentuk masih belum berdistribusi normal, oleh karena itu alangkah baiknya untuk membentuk dan melakukan pendekatan statistika sehingga data tersebut dapat berdistribusi normal, atau bisa dengan mengumpulkan data lebih banyak dan berkala.
4. *Forecasting* yang dihasilkan masih belum baik, oleh karena itu alangkah baiknya untuk mencoba menggunakan pendekatan lainnya agar akurasi dapat meningkat lebih baik.

LAMPIRAN

JADWAL Pengerjaan

TABEL 1 JADWAL TWITTER SEBELUM UTS

Pertemuan	Kegiatan	Fitur
1	Membuat grup sekaligus membahas topik mana yang akan dipilih	-
2	Identifikasi kebutuhan seperti data, perangkat, sistem	-
3	Pendekatan umum yang dirancang	-
4	Studi literatur	-
5	Studi literatur	-
6	Menunggu <i>approve</i> akun <i>Twitter Developer</i>	-
7	<i>Fetch data</i> dari Twitter dan mengumpulkan dalam .csv	-Tweepy Fetch -Filter by date -List to dataframe converter -Dataframe to csv converter
8	Membangun proposal	-

TABEL 2 JADWAL TWITTER SESUDAH UTS

Pertemuan	Kegiatan	Fitur
9	<i>Fetch data</i> dari Twitter dan mengumpulkan dalam .csv	-Tweepy Fetch -Filter by date -List to dataframe converter -Dataframe to csv converter
10	Preprocessing data	- <i>Scikit Learn</i>
11	Preprocessing data	- <i>Scikit Learn</i>
12	Clustering data	-sklearn.cluster.AgglomerativeClustering
13	TF-IDF Vectorization	-sklearn.feature_extraction.text.TfidfVectorizer
14	LDA Topic Modelling	- sklearn.discriminant_analysis.LinearDiscriminantAnalysis
15	Membangun dashboard dan melakukan forecasting	-
16	Membangun proposal	-

TABEL 3 JADWAL WEB CRAWLER SEBELUM UTS

Pertemuan	Kegiatan	Fitur
1	Membuat grup sekaligus membahas topik mana yang akan dipilih	-
2	Identifikasi kebutuhan seperti data, perangkat, sistem	-
3	Pendekatan umum yang dirancang	-
4	Studi literatur	-
5	Studi literatur	-
6	Studi literatur	-
Pertemuan	Kegiatan	Fitur
7	Mengambil <i>URL</i> dari website <i>idntimes.id</i>	-Jupyter Notebook -Beautiful Soup -Urlopenlib
8	Membangun proposal	-

TABEL 4 JADWAL WEB CRAWLER SESUDAH UTS

Pertemuan	Kegiatan	Fitur
9	Mengambil <i>URL</i> dari website <i>kumparan.com</i> dan <i>gizmologi.id</i>	-
10	Melakukan web crawling dari <i>URL</i> yang sudah diambil	-
11	Ekstraksi data untuk mengambil keyword	- <i>Scikit Learn</i>
12	Clustering data	-sklearn.cluster.AgglomerativeClustering
13	TF-IDF Vectorization	-sklearn.feature_extraction.text.TfidfVectorizer
14	LDA Topic Modelling	- sklearn.discriminant_analysis.LinearDiscriminantAnalysis
15	Membangun dashboard dan melakukan forecasting	-
16	Membangun proposal	-

TABEL 5 JADWAL DATASET INTERNAL SESUDAH UTS

Pertemuan	Kegiatan	Fitur
9	Melakukan pengambilan dataset internal dari DailySocial.id dan menyimpannya dalam .csv	-
10	Preprocessing data	- <i>Scikit Learn</i>
11	Preprocessing data	- <i>Scikit Learn</i>
12	Clustering data	-sklearn.cluster.AgglomerativeClustering
13	TF-IDF Vectorization	-sklearn.feature_extraction.text.TfidfVectorizer
14	LDA Topic Modelling	- sklearn.discriminant_analysis.LinearDiscriminantAnalysis
15	Membangun dashboard dan melakukan forecasting	-
16	Membangun proposal	-

LAMPIRAN KODE DAN VIDEO

A. TAUTAN KODE

Berikut merupakan tautan untuk melihat kode program pada penelitian ini:

<https://github.com/sntdshrly/Information-Retrieval>

B. TAUTAN VIDEO

Berikut merupakan tautan untuk melihat video penjelasan pada penelitian ini:

<https://youtu.be/7i-h-tY7orE>

DAFTAR PUSTAKA

Rahutomo, F., Saputra, P. Y. and Fidyawan, M. A. (2018) 'Implementasi Twitter Sentiment Analysis Untuk Review Film Menggunakan Algoritma Support Vector Machine', *Jurnal Informatika Polinema*, 4(2), p. 93. doi: 10.33795/jip.v4i2.152.

Zhu, Z. *et al.* (2019) 'Hot Topic Detection Based on a Refined TF-IDF Algorithm', *IEEE Access*, 7, pp. 26996–27007. doi: 10.1109/ACCESS.2019.2893980.

Rofiqi, M. A. *et al.* (2019) 'Implementasi Term-Frequency Inverse Document Frequency (TF-IDF) Untuk Mencari Relevansi Dokumen Berdasarkan Query', *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 1(2), pp. 58–64. doi: 10.28926/ilkomnika.v1i2.18.

Annisa, A., Munarko, Y. and Azhar, Y. (2016) 'Peringkasan Tweet Berdasarkan Trending Topic Twitter Dengan Pembobotan TF-IDF dan Single Linkage Angglomerative Hierarchical Clustering', *Kinetik*, 1(1). doi: 10.22219/kinetik.v1i1.7.

Destarani, A. R., Slamet, I. and Subanti, S. (2019) 'Trend Topic Analysis using Latent Dirichlet Allocation (LDA) (Study Case: Denpasar People's Complaints Online Website)', *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 5(1), pp. 50–58. doi: 10.26555/jiteki.v5i1.13088.

Javed Mehedi Shamrat, F.M. *et al.* (2020) 'An effective implementation of web crawling technology to retrieve data from the world wide web (Www)', *International Journal of Scientific and Technology Research*, 9(1), pp. 1252–1256.

Li, W., Wang, S. and Bhatia, V. (2016) 'PolarHub: A large-scale web crawling engine for OGC service discovery in cyberinfrastructure', *Computers, Environment and Urban Systems*, 59, pp. 195–207. doi:10.1016/j.compenvurbsys.2016.07.004.

et al. (2015) 'A Study of Web Information Extraction Technology Based on Beautiful Soup', *Journal of Computers*, 10(6), pp. 381–387. doi:10.17706/jcp.10.6.381-387