# MULTI CHANNEL ACCESS – MDI (A)

**SCRAPING DATA, TEXT SUMMARIZATION, & TEXT ANALYSIS MENGGUNAKAN LIBRARY BEAUTIFULSOUP4, ALGORITMA MAXIMUM MARGINAL RELEVANCE (MMR), DAN MODEL LEXICON BASED PADA WEBSITE BERITA DI INDONEISA**



I Made Rismawan Nugraha   
NIM :1805551114

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS UDAYANA  
2021**

# Deskripsi

Kebutuhan akan informasi di jaman yang moderen ini merupakan suatu kebutuhan yang sangat penting dan mudah didapat karena adanya perkembangan teknologi komunikasi berdampak pada penggunaan internet untuk mempublikasi media digital. Media digital dapat berupa artikel, blog, maupun situs-situs berita yang berisi informasi seperti dokumen (*teks*). Sebuah media digital kebanyakan dibuat dengan penulisan kata yang berulang sehingga menemukan sebuah representasi yang tepat dari informasi tekstual tersebut merupakan sebuah hal yang penting untuk membuat pemahaman dari keseluruhan teks. Sebuah dokumen tersebut dapat diringkas oleh suatu sistem tanpa menghilangkan informasi yang penting maka pembaca dapat menghemat waktu, karena pembaca dapat memahami dan mengetahui informasi penting dari dokumen tersebut tanpa harus membaca isi dokumen secara keseluruhan.

Meringkas pada umumnya dapat dilakukan dengan mudah jika pembaca relative familiar terhadap teks yang sudah dibaca atau jika teks tersebut tidak terlalu panjang. Hasil rangkuman setiap individu yang bersifat subjektif sedangkan untuk menghasilkan sebuah rangkuman yang baik dan bagus perlu pertimbangan yang bersifat objektif. Meringkas dokumen secara manual oleh manusia, membutuhkan banyak biaya dan waktu apabila dokumen tersebut banyak dan panjang sehingga diperlukan sistem peringkas otomatis (*automatic summarization)* untuk mengatasi banyaknya biaya dan waktu tersebut ­­­(Mustaqhfiri, Abidin, & Kusumawati, 2012).

*Automatic Summarization* yang dirancang harus efisien terhadap waktu dan efektif terhadap ketepatan penyajian informasi. Ada beberapa metoda dan algortima yang digunakan dalam membuat engine *Automatic Summarization*. Algoritma yang digunakan untuk peringkasan artikel adalah algoritma TextRank. TextRank merupakan *graph-based ranking* *algorithm* (graf dengan model pemeringkatan) untuk pemrosesan teks. TextRank yang digunakan sebagai metode yang menghasilkan ekstraksi berupa kalimat (*TextRank for Sentence Extraction*). TextRank sangatlah fleksibel karena dapat digunakan pada berbagai bahasa tanpa mengubah algoritmanya. Hal ini dikarenakan TextRank tidak memerlukan data training untuk proses pengelolahan dokumen (Pragantha, & Tarumanagara, 2017).

# Tinjauan Pustaka

Tinjauan Pustaka merupaka bagian pendukung dari pembuatan engine ini. Penjelasan dari teori pendukung akan di jelaskan berdasarkan progress atau tahapan yang akan dilakukan. Progress berisikan tahapan dari pembuatan *Automatic Summarization* yang dapat dibagi menjadi 3 yaitu proses pembuatan *Scraping Data, Text Summarization,* dan *Text Analysis.* Penjelasan lebih lanjut mengenai progress akan dijabarkan sebagai berikut.

## Scraping Data

Scraping data adalah suatu teknik mengambil text atau secara umum obyek yang ada di situs lain yang nantinya dipergunakan untuk mendapatkan data dari tampilan sebuah website tersebut kemudian diletakkan ke website kita. Kesimpulannya jika terdapat perubahan data pada website parent yang kita lakukan grabbing tersebut maka otomatis data yang kita ambil juga pasti akan berubah tanpa harus merubahnya secara manual. Pada tahapan pembuatan engine Scraping data saya menggunakan *library open source* yang dapat digunakan yaitu library BeautifulSoup. Penjelasan terkait dengan library tersebut dapat dilihat pada penjelasan berikutnya.

### BeautifulSoup

BeautifulSoup merupakan salah satu *library open source* yang dapat digunakna pada Python yang memungkinkan untuk melakukan scraping dengan mudah dan cepat. BeautifulSoup hadir dengan metode-metode sederhana sehingga dapat dengan mudah melakukan proses navigasi, pencarian dan juga modifikasi struktur data situs yang di-scraping. BeautifulSoup dapat mengekstrak data, baik dari file XML maupun HTML dengan mengubah markup menjadi parse tree. Parsing HTML adalah suatu teknik untuk memisahkan suatu teks dari *tag-tag code* HTML sehingga menghasilkan sebuah data yang spesifik. Pengambilan data yang biasanya dilakukan secara manual dapat dilakukan secara otomatis dengan menggunakan BeautifulSoup. (Wesley, 2019).

|  |
| --- |
|  |

Gambar 2. 1 **Source Code Testing Grabing Data**

Gambar 2.1 merupakan tahapan awal di mana saya akun melakukan pengambilan data pada website. Jadi setelah meng-GET data pada URL yang telah di deklarasikan di awal, parse data tersebut menggunakan library dari beatifulsoup dan mengambil element html pada URL tersebut. Jadi di awal saya mengambil tanggal tayang dan juga nama bioskop yang dapat dilihat pada gambar di atas. Jadi pada dasarnya kita mengambil suatu informasi melalui tag-tag HTML yang ada pada sebuah *article* atau media digital yang ada.

## Text Summarization

Text SummarizationPeringkasan teks otomatis (*text summarization*) adalah pendekatan yang bisa digunakan untuk meringkas aritkel yang panjang menjadi lebih pendek dan ringkas, sehingga hasil rangkuman teks yang relative pendek bisa mewakilkan teks yang panjang tersebut. Pada dasarnya menggunakan prinsip *Natural Language Processing* dan algoritma untuk membuat sebauh engine tersebut. Pada penelitian ini saya menggunakan algoritma dari TextRank yang nantinya akan dijelaskan lebih detail pada penjelasan berikutnya.

### Maximum Marginal Relevance (MMR)

Algoritma MMR merupakan metode ekstraksi ringkasan yang digunakan untuk meringkas dokumen tunggal maupun multi dokumen MMR meringkas dokumen dengan melakukan perhitungan kesamaan antara bagian teks. Pada peringkasan dokumen dilakukan proses segmentasi dokumen menjadi kalimat dan dilakukan pengelompokan sesuai dengan gender kalimat tersebut. MMR digunakan dengan mengkombinasikan kalimat yang diberikan pengguna. Ide dasar dari MMR ini yaitu memberikan penambahan nilai bagi kalimat yang relevan dan memberikan pengurangan nilai redundansi informasi antara kalimat tersebut dengan kalimat lain yang telah terpilih. Sebuah kalimat dikatakan memiliki marginal relevance yang tinggi jika kalimat tersebut relevan terhadap isi dari kalimat dan mempunyai kesamaan bobot term maksimum dibandingkan dengan query. Peringkasan kalimat dengan tipe ekstraktif, nilai akhir diberikan pada kalimat Si dalam MMR dihitung dengan persamaan

|  |
| --- |
|  |

Gambar 2. 2 **Perhitungan MMR**

Gambar 2.2 merupakan perhitungan dari MMR yang menjelaskan tentang himpunan kalimat relevan yang dipilih, Sim1 dan Sim2 adalah matriks dari kesamaan kalimat. Untuk mendapatkan kumpulan kalimat yang relevan ini, kita memberikan peringkat pertama pada kalimat yang relevan dengan memberikan presentase tertentu dari kalimat tersebut.

## Text Analysis Sentiment­

Text Analysis adalah proses menganalisis tulisan *online*untuk menentukan nada emosional dari penulisnya. Analisa sentimen (*sentiment analysis*) merupakan proses penambangan kontekstual teks yang mengidentifikasi dan mengekstrak informasi subjektif dalam sumber yang dapat membantu kita untuk menilai sentimen sesuai dengan yang kita inginkan. Karena engine yang dibuat di khususkan untuk mengambil berita Indonesia jadi sentiment analisis pada penelitian ini menggunakan model lexicon. Model Lexicon akan dijelaskan pada penjelasan berikut ini.

### Model Lexicon

Lexicon based merupakan metode yang digunakan untuk mengelompokkan opini masyarakat ke dalam 3 kelas yaitu, sentimen positif, negatif atau netral. Model Lexicon ini sendiri didasarkan pada orientasi kontekstual sentimen pada jumlah orientasi sentimen pada setiap kata atau kalimat. *Lexicon based* umumnya menggunakan kamus untuk mendukung klasifikasi sentimen yaitu SentiWordNet. SentiWordNet merupakan hasil anotasi otomatis dari semua synset pada WordNet.

# Kebutuhan Pembuatan Engine

Kebutuhan Pembuatan Engine dalam penelitian ini dengan menggunakan beberapa *library* yang membantu dalam pembuatan. Adapun penjelasan mengenai *library* yang digunakan akan dijelaskan sebagai berikut.

Table 1 Library Engine Scraping dan Summarization

|  |  |  |
| --- | --- | --- |
| **Library** | **Version** | **Kegunaan** |
| mysql.connector | 2.2.9 | Berguna dalam melakukan pengolahan data dan penyimpanan data pada database |
| Nltk | 3.5 | library yang sangat powerfull untuk digunakan dalam pemrosessan human language data. Memberikan kemudahan interfacing ke lebih dari 50 corpora dan lexial resources |
| import\_ipynb | 0.1.3 | Berguna agar dapat mengakses sebauh class yang diakses pada lain folder engine |
| regex | 2020.10.15 | Regex adalah singkatan dari Regular Expresion. Regex merupakan sebuah teks (string) yang mendefinisikan sebuah pola pencarian sehingga dapat membantu kita untuk melakukan matching (pencocokan), locate (pencarian), dan manipulasi teks |
| sastrawi | 1.0.1 | untuk mengembalikan kata kebentuk dasarnya |
| spacy | 3.2.0 | spaCy dan Python untuk mendeteksi persamaan antar kalimat |
| stopwords | 1.0.0 | Stop words adalah kata umum (common words) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna |
| termcolor | 1.1.0 | Untuk melakukan visualisasi pada data summary. |
| dateTime | 4.3 | Berguna dalam pengolahan waktu yang dimasukan oleh pengguna. |
| csv | 3.8.9 | untuk membaca file csv yang dan membuat file csv |
| beautifulsoup4 | 4.9.3 | Berguna dalam mengambil semua data HTML berdasarkan url yang dimasukan. User dapat mengambil data pada website dengan mendeklarasi Tag HTML dan Class yang digunakan. |
| requests | 2.24.0 | Requests adalah library yang mudah digunakan dengan banyak fitur mulai ... Hal ini akan berguna ketika kamu berurusan dengan file media |

Table 1 merupakan library yang digunakan dalam membuat sebuah engine scrpaing dan summarization. Version dan penjelasan kegunaan dari library tersebut juga sudah dijelaskan dengan lengkap.

# Langkah Pengerjaan

Pembahasan mengenai pembuatan engine yang sudah dikerjakan akan dibagi menjadi 3 bagian yaitu proses Scraping Data, Text Summarization, dan Analysis Sentiment. Penjelasan lebih lanjut terkait 3 bagian pembuatan engine tersebut akan dijelaskan pada sub bab di bawah ini.

## Scraping Data

Scraping Data pada pembuatan engine tersebut menggunakan library dari BeautifulSoup untuk mengambil data pada sebuah website. BeautifulSoup4 hanya bisa mengambil data tersebut dengan mempelajari sebuah halam website yang akan dilakukan Scraping. Cara kerja dari BeautifulSoup4 ini sendiri yaitu dengan mepelajari halaman website yang akan dilakukan Scraping, sehingga jika ingin mengambil Data Scraping dari berbeda halaman website maka diperlukannya analisa *source code* yang terdapat pada *domain* atau website tersebut. Pengambilan Scraping Data berita yang akan dilakukan pada tugas ini yaitu dengan menganalisa *source code* yang terdapat pada website berita Indonesia yaitu CNCBIndonesia, Detik.com, Kompas.com, Kumparan, Liputan6 dan TribunNews. Gambaran alur proses pengguna menggu engine scraping akan dijelaskan pada Gambar 4. 1 di bawah ini.

|  |
| --- |
| C:\Users\risma\Desktop\FILE VSION MCA.jpg |

Gambar 4. 1Alur Proses User Menggunakan Engine Scraping

Gambar 4. 1 merupakan Alur Proses User Menggunakan Engine Scraping yang telah dibuat. Pengguna hanya tinggal memasukan parameter yang tersedia yaitu Tanggal Berita dan Pencarian Berita yang akan dicari. Engine nantinya akan mencari berdasarkan parameter yang dimasukan oleh user di Class yang telah dibuat diatas yaitu terdapat 6 Class Berita yaitu Class CNCBIndonesia, Detik.com, Kompas.com, Kumparan, Liputan6 dan TribunNews. Data yang telah diambil dari 6 website berita tersebut nantinya akan disimpan kedalam *database.* Proses Engine dari masing-masing class akan dijelaskan pada Gambar 4. 2di bawah ini.

|  |
| --- |
| C:\Users\risma\Desktop\AlurKerjaClass.jpg |

Gambar 4. 2 Alur Kerja Engine Class Berita

Gambar 4. 2 merupakan alur kerja dari engine class scraping berita secara umum dilakukan. Pembuatan engine dengan mengimport library BeautifulSoup4, mysql.connector, Exception, dan Datetime. Parameter yang didapatkan dari engine tersebut yaitu url berita, Tanggal Berita, dan kata kunci pencarian berita tersebut. Selanjutnya dengan menggunakan library dari BeautifulSoup4 mengambil semua content yang terdapat pada url tersebut. Analisa Tag HTML pada website ini yaitu mencari data-data yang akan diambil pada website tersebut dengan parameter pengambilan tag yaitu tag html dan class yang digunakan. Pada website berita itu hasil yang didapatkan yaitu berupa per halaman dari berita yang ditampilkan, jadi pengambilan semua url detail dari berita tersebut dan di simpan pada variable DataURLDetail. Selanjutnya dilakukan perulangan scraping pada proses Scraping Data dan engine akan menvalidasi data yang akan diambil, apakah sudah pernah di scraping atau belum. Jika sudah maka dilakukan ke pengecekan apakah DataURLDetail masih tersedia, jika masih maka akan dilakukan scraping lagi hingga data pada DataURLDetail itu habis. Data yang berhasil di dapatkan akan disimpan kedalam database. Penjelasan selanjutnya akan menjelaskan source code dari engine Scraping Data.

### Fungsi Main Engine Scraping Data

Fungsi main dari *engine Scraping Data* ini merupakan inti dari *engine* .Main Engine berguna dalam memanggil fungsi-fungsi atau Class yang telah dibuat. Fungsi main dari *engine Scraping Data* ini dapat dilihat pada Gambar 4. 3 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 3 Engine Master Scraping Data

Gambar 4. 3 merupakan *source code* dari Engine Master Scraping Data. Engine Master karena setiap fungsi dan Class akan dipanggil semuanya di source code tersebut. Kernel pertama adalah bagian import semua Class yang telah dibuat berdasarkan website berita di Indonesia. Pada bagian kernel ketiga atau yang bergambarkan kotak merah tersebutlah pengguna dapet mengganti parameter yang diinginkan. Parameter yang dimasukan adalah Keyword dan Tanggal Berita itu terbit. Parameter tersebut nantinnya digunakan pada fungsi scrapingData, dimana fungsi tersebut berguna dalam melakukan scraping pada tiap-tiap Class pada kernel bagian 2 yaitu def scrapingData. Penjelasan secara umum dari Class akan dijelaskan pada gambar di bawah ini. Data url yang didapatkan pada masing-masing Class tersebut dibuat dengan pengambilan data pada *database*.

|  |
| --- |
|  |

Gambar 4. 4 Parameter Pemanggilan Setiap Class

Gambar 4. 4 merupakan parameter yang digunakan dalam pemanggilan setiap class tersebut. Pengaturan parameter pencarian berita dan tanggal berita dibuat pada engine secara lansung agar pengguna dapat secara fleksibel menggunakan atau mengganti isi dari parameter dengan mudah.

### Pembuatan Class Scraping

Class Scraping merupakan bagian source code yang digunakan dalam Scraping Data berdasarkan url website berita. Terdapat 6 Class yang digunakan dalam melakukan scraping. 6 Class tersebut sebelumnya sudah dianalisa berdasarkan isi tag HTML yang digunakan pada website tersebut. 6 Class tersebut sesuai dengan pengambilan list berita yang akan diambil yaitu CNCBIndonesia, Detik.com, Kompas.com, Kumparan, Liputan6 dan TribunNews.

|  |
| --- |
|  |

Gambar 4. 5 Class Scraping pada Engine

Gambar 4. 5 merupakan Class yang dibuat dalam melakukan Scraping pada website berita. Terdapat 6 Class yang terdiri dari CNBC, Detik, Kompas, Kumparan, Lipuitan6, dan Tribunnews. Dari ke 6 Class tersebut terdapat scraping data yang sama tetapi pemanggilan Tag , Class , dan Id pada masing-masing Class tersebut berbeda. Struktur dari semua Class yang dibuat semuanya mirip dengan file yang lainnya,tetapi dalam tidak bisa di pergunakan untuk semua website dikarenakan setiap website memiliki karakter tag atau class HTML yang berbeda-beda. Jika ingin mengambil data dari website yang lain, terlebih dahulu kalian harus menganalisa website tersebut. Penjelasan lebih dalam mengenai isi dari Class akan dijelaskan secara umum menggunakan Class CNBC.

#### Import Library Class

Import Library Class merupakan tahapan yang terpenting sebelum melakuakan Scraping Data. Import Library memudahkan developer dalam menggunakan fungsi-fungsi yang sudah ada sesuai dengan keperluan dari library yang digunakan. Penjelasan lebih lanjut terkait dengan library yang digunakan dalam Class Scraping ini akan dijelaskan sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 6 Improt Library File Class Berita

Gambar 4. 6 merupakan library yang digunakan pada Class Scraping Data. Penjelasan dari masing-masing library akan dijelaskan sebagai berikut.

1. Exception : Berguna dalam menghandle error yang terjadi pada engine agar ketika terdapat error pada engine, engine tidak langsung berhenti berjalan.
2. BeautifulSoup : Berguna dalam mengambil semua data HTML berdasarkan url yang dimasukan. User dapat mengambil data pada website dengan mendeklarasi Tag HTML dan Class yang digunakan.
3. Datetime : Berguna dalam pengolahan waktu yang dimasukan oleh pengguna.
4. Get : Mengecek response yang terdapat pada url tersebut benar adanya.
5. Mysql.connector : Berguna dalam melakukan pengolahan data dan penyimpanan data pada database.

#### Constructor Class Scraping Berita

Constructor merupakan suatu method yang akan memberikan nilai awal pada saat suatu class yang dibuat saat melakukan pemanggilan. Penjelasan lebih lanjut terkait dengan Constructor Class Scraping Berita akan dijelaskan sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 7 Constructors Class Scraping Berita

Gambar 4. 7 merupakan Constructor pada class scraping berita. Pemanggilan sebuah class scraping berita tersebut harus menggunakan 3 parameter yang digunakan yaitu urlberita, pencarian berita, dan tanggal tersebut berita tersebut. Method constructor itu juga mendeklarasikan sebuah objek dengan koneksi database yang digunakan pada engine. Terdapat kondisi apabila tanggal berita yang dicari kosong, maka engine akan menggunakan tanggal sekarang untuk mencari sebuah berita.

#### Fungsi Simpan dan Validasi Database

Fungsi *database* dari *engine scraping data* adalah melakukan penyimpanan data dan melakukan validasi data pada *database*. Method pengolahan *database* ini dapat dilihat pada Gambar 4. 8 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 8 Fungsi Pengolahan Database (Validasi dan Simpan Data)

Gambar 4. 8 meperlihatkan terdapat 2 fungsi yaitu simpanDatabase dan validasiDatabase. Fungsi dari simpanDatabase itu berguna dalam melakukan penyimpanan data setelah melakukan grabbing data yang telah dilakukan. Dengan memanggil fungsi tersebut data yang akan diinput berupa type data tuple sebagai parameter pemanggil fungsi simpanDatabase() dan juga *connection* database yang diambil dari deklarasi connection database diawal. Fungsi dari validasiDatabase() ini berguna dalam melakukan pengecekan terhadap *database* apakah data yang akan discraping sudah terinput atau belum pada *database*. Fungsi tersebut berguna dalam menghindari terjadinya data double pada database. Parameter yang digunakan pada fungsi tersebut adalah *connection* database dan urlDetail dari database tersebut, apakah sudah terdapat urlDetail tersebut pada database dan jika belum maka akan dilakukan input data menggunakan fungsi simpanDatabase().

#### Fungsi Mengambil Halaman pada Website Berita

Fungsi getAllPageWebsite dari engine *scraping data* adalah melakukan pengambilan jumlah halaman yang terdapat pada pencarian berita tersebut. Website berita dalam melakukan pencarian sebuah berita, pasti menyajikan daftar berita itu dalam bentuk list daftar berita yang disajikan dalam page atau halam website. Untuk mengambil semau berita yang terdapat pada semua page tersebut, dengan menganalisa dan mengambil halam terkahir dari berita dan melakukan looping pada url yang telah di deklarasikan untuk mendapatkan urlDetail semua berita yang dicari. Fungsi getAllPageWebsite ini dapat dilihat pada Gambar 4. 9 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 9 Fungsi getAllPageWebsite pada Class Scraping Berita

Gambar 4.8 merupakan fungsi yang dibuat untuk mengambil jumlah halaman yang terdapat pada website berita tersebut. Return yang diberikan dari pemanggilan fungsi ini sendiri adalah integer jumlah page. Terdapat kondisi apabila terdapat kata kunci pada pencarian maka dia akan masuk ke website index berita dengan tanggal yang dimasukan pengguna ataupun default pengguna telah di atur pada fungsi constractor. Kenapa proses ini diperlukan karena dalam melakukan scraping seorang biasanya hanya mengambil halaman pertama saja, jadi disini tujuannya mengambil jumalah halaman yang berguna dalam mengambil semua data berita yang ada.

#### Fungsi Mengambil URL Detail Berita (getULRinPage)

Fungsi getUrlInPage dari engine scraping data adalah melakukan pengambilan data url dari masing-masing berita yang ada. Pengambilan urlDetail dengan melakukan looping pada jumlah *page* yang didapatkan pada fungsi getAllPageWebsite. Fungsi getUrlInPage ini dapat dilihat pada Gambar 4. 10 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 10 Fungsi Mengambil Semua URL Berita

Gambar 4. 10 merupakan penjelasan dari Fungsi Mengambil Semua URL Berita. Parameter pemanggilan fungsi tersebut adalah page yang didapatkan dari fungsi getAllPageWesbsite. Maka nantinya akan diambil setiap *url* berita list menggunakan looping dengan url pada page ke berapa. Karena untuk mendapatkan content dari berita tersebut mempunyai harus mempunyai link yang berbeda-beda setiap beritanya, jadi link dari detail berita tersebut akan disimpan kedalam Array dan akan di *looping* untuk mengambil *content* dari detail berita tersebut. Array tersebut nantinya akan digunakan untuk mengambil scraping lebih dalam seperti menggambil data judul,penulis, dan isi berita. Return dari fungsi ini adalah Arrya yang berisikan list-list urldetail berita yang akan digunakan untuk scraping data detail.

#### Fungsi Mengambil Detail Data Berita (scrapingDataDetail)

Fungsi scrapingDataDetail dari engine grabbing data adalah melakukan pengambilan data detail yang terdapat pada urlDetail berita yang ada. Pengambilan data yang diambil berupa judul berita, label berita, penulis, kapan terbitnya, url berita tersebut, dan content dari berita tersebut. Fungsi scrapingDataDetail ini dapat dilihat pada Gambar 4. 11 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 11 Fungsi Mengambil Semua Data Pada URL Detail

Gambar 4. 11 merupakan sebuah fungsi utama dari sebuah Class Scraping Berita. Pada fungsi ini lah semua fungsi yang telah dibuat sebelumnya yaitu getUrlInPage, getAllPageWebsite, validasiDataDatabase, dan simpanDatabase. Alur pengambilan data dimulai dari pengambilan getUrlInPage yaitu mengambil semua url berita yang terdapat pada halaman website, dengan parameter yang digunakan adalah jumlah page yang didapatkan dari *return* fungsi getAllPageWebsite. Alur selanjutnya yaitu melakukan validasi pada database agar tidak melakukan scraping kemabli yang membuat performa engine melambat dan data double.Validasi tersebut menggunakan fungsi validasiDataDatabase dengan parameter urldetail yang didapatkan. Selanjutnya engine akan mencoba mengambil menggunakan *source* *try catch* untuk menanggulangi terjadinya data missing saat mescraping data. Data yang diambil berupa data judul berita, label berita, penulis, kapan di publishnya sebuah berita, url dan content dari berita tersebut. Selanjutnya data akan disimpan kedalam *database* sesuai dengan fungsi yang dibuat diatas tersebut. Hasil dari grabbing data tersebut dapat dilihat pada Gambar 4. 12 sebagai berikut.

#### Hasil Running Engine Scraping Data (scrapingDataDetail)

Hasil Running Engine Scraping Data dari ke 6 website berita yang telah di analisa tersebut dapat dilihat pada Gambar 4. 12 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 12 Hasil Scraping Data dari Engine

Gambar 4. 12 merupakan hasil dari setelah engine scrapingData dijalankan. Table yang tidak terisi hanya bagian field summary dan analisa sentiment yang nanti akan dilanjutkan pada engine summarization. Pengambilan dari 6 website berita yang ada di Indonesia itu engine kurang lebih membutuhkan waktu 20 menit dengan total data yang didapatkan 1148 data. Pengambilan data tersebut tergantung dengan katagori pencarian dan tanggal yang ditentukan. Pada pencarian tersebut saya menggunakan kata kunci dari Vaksin dan tanggal 27 December 2021.

## Text Summarization

Text Summarization adalah pembuatan bentuk yang lebih singkat dari suatu teks dengan memanfaatkan suatu algoritma tertentu. Alur proses dari melakukan text summarization melalui beberapa tahapan yaitu Persiapan Data, melakukan Text Preprocessing, Cleansing Data, Case Folding, Tokenizing Data, dan Ssteamming data tersebut agar mendapatkan hasil yang maximal pada perhitungan Similarity kata. Proses Text Summarization ini akan dijelaskan lebih lanjut sebagai berikut. Alur dari engine Text Summarization dapat dilihat pada Gambar 4. 13 sebagai berikut.

|  |
| --- |
| C:\Users\risma\Desktop\textsum.jpg |

Gambar 4. 13 Alur Engine Text Summarization

Gambar 4. 13 merupakan alur proses kerja dari engine Text Summarization. Dimulai dari pemanggilan data yang belum tersummary. Selanjutnya dengan melakukan atau memisahkan data dari paragraph ke kalimat yang berguna dalam nantinya agar gampang meranking sebuah kalimat. Proses selanjutnya melakukan Text Preprocessing pada kata yang telah di tokenize denga tahapan yang harus dilalui Text Preprocessing yaitu Steaming, Cleansing, dan Stopword kalimat. Hasil setelah melalui proses tersebut adalah Text Cleansing yang berisikan kata dasar dari suatu kalimat dan pengurangan sebuah kata yang tidak penting. Pemanggilan Class MMR yang bertujuan sebagai proses summarization pada Engine dengan melakukan hal pertama yaitu proses pengecekan similarity pada kalimat lalu dengan proses selanjutnya yaitu melakuakn ranking pada sebuah kalimat berita.Output tersebut berupa data text yang telah diringkas berdasarkan pengurutan algoritma dan text berita. Setelah mendapatkan output tersebut dilakukan proses analisis sentiment menggunakan model Lexicon dan menghasilkan output Labeling dan Score yang didapatkan.

### Import Library Engine Text Summarization

Import Library dan Class merupakan tahapan yang terpenting sebelum melakuakan Text Summarizartion. Import Library memudahkan developer dalam menggunakan fungsi-fungsi yang sudah ada sesuai dengan keperluan dari library yang digunakan. Penjelasan lebih lanjut terkait dengan library yang digunakan dalam engine Text Summarizartion ini akan dijelaskan sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 14 Import Library pada Engine Text Summarization

Gambar 4. 14 merupakan library yang digunakan pada engine Text Summarization. Penjelasan dari masing-masing library akan dijelaskan sebagai berikut.

1. Mysql.connector : Berguna dalam melakukan pengolahan data dan penyimpanan data pada database.
2. sent\_tokenize : Berguna dalam melakukan pemenggalan pada suatu kalimat.
3. Import\_ipynb : Berguna agar dapat mengakses sebauh class yang diakses pada lain folder engine.
4. re : Berguna dalam melakuakn cleansing data atau cara kerjanya yaitu mereplace sebuah kata.
5. StemmerFactory : Merubah sebuah kata menjadi kata dasar dan membuat sebuah kalimat menjadikan lower atau huruf kecil semua.
6. Pd : utnuk pengolahan data yang masuk pada database
7. MMR : Berguna dalam melakukan summary data berita.
8. Stopword : Berguna dalam menghilangkan kata-kata yang dianggap tidak penting agar dapat lebih baik dalam menganalisa suatu kalimat.
9. String : Menghilangkan kalimat tanda baca pada suatu kalimat
10. Colored : Untuk melakukan visualisasi pada data summary.
11. Datetime : Berguna dalam pengolahan waktu yang dimasukan oleh pengguna.
12. Csv : untuk membaca file csv yang dan membuat file csv
13. NLTK.Tokenize : Berguna dalam merubah paragraph menjadi kalimat dan fungsi secara umumnya yaitu pengolahan data pada string atau text.

### Pembuatan Fungsi Connection Database dan Steaming

Deklarasi conncetion database dan fungsi dari steaming data berguna dalam nantinya dalam melakukan pengolahan data pada database dan pengolahan data pada texts dengan memanggil fungsi steming data yang telah dibuat ini. Pembuatan connection database dan steaming data ini dapat dilihat pada Gambar 4. 15 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 15 Pembuatan Connection Database dan Fungsi Steaming Data

Gambar 4. 15 merupakan tahapan membuat koneksi database dan juga membuat fungsi dari stemmerFactory yang berguna dalam melakukan steaming sebuah kalimat atau kata. Penggunaan fungsi tersebut akan digunakan pada kode program berikutnya.

### Proses Text Preprocessing

Masuk ketahapan text preprocessing yang meliputi tahapan T*okenize, Cleansing. , Case Folding, Steaming,* dan *Stopword.* Proses Text Preprocessing tersebut dapat dilihat pada Gambar 4. 16 sebagai berikut. Penjelasan lebih lanjut mengenai beberap fungsi yang telah dibuat tersebut akan dijelaskan sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 16 Fungsi Text Preprocessing Data

Gambar 4. 16 merupakan tahapan dari proses Text Preprocessing Data. Dalam tahapan tersebut terdapat beberapa Fungsi yang telah dibuat. Adapun penjelsan dari masing-masing fungsi akan dijelaskan sebagai berikut.

* **ToktokenizeParagraftoKalimat** adalah fungsi agar seuatu paragraph itu dibagi menjadi beberapa kalimat dengan menjadikan beberapa potong kalimat yang terdapat pada paragraph.
* **cleansingData** adalah fungsi dalam mebersihkan data berita yang terdapat pada database. Data yang diambil tidak terlalu banyak kata kotornya, jadi dengan menggunakan 2 fungsi regex dan string sebagai penghilan tanda baca pada suatu kalimat saja sudah cukup.
* **steamingData** berfungsi sebagai menjadikan kata menjadi lower atau huruf kecil sekaligus juga menjadikan seuatu kalimat menjadi kalimat dasarnya,
* **indoStopWord** berguna dalam menghilangkan sebuah kata sambung yang tidak penting pada kalimat agar nantinya dapat dengan maksimal melakukan perhitungan similarity sebuah kalimat tersebut. pada Stopword tersebut menggunakna library NLTK dan text Eksternal.

### Fungsi Stopword Data

Fungsi Stopword Data berguna dalam menghilangkan kata-kata yang tidak penting seperti kata hubung sebuah kalimat. Fungsi Stopword ini dibuat menjadi 2 bagian yaitu load\_stopWord dengan menggunakan file reverensi dan menggunakan library NLTK. Pembuatan fungsi ini dapat dilihat pada Gambar 4. 17 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 17 Fungsi Stopword Data Use Library dan File External

Gambar 4. 17 merupakan fungsi dari pengambilan data loadstop word yang menggunakan kernel pada bagian atas menggunakan file.txt yang didapatkan pada internet dan yang kedua adalah stopword menggunakan library NLTK. Komperasi kedua stopword tersebut digunakan untuk mendapatkan hasil stopword yang lebih maksimal. Pada Library NLTK terdapat 789 total kata dan kalau dari file.txt terdapat 917 kata.

### Class Algoritma MMR

Pembuatan clas algoritma MMR ini berguna dalam memudahkan melakukan kalkulasi peringkasan dengan mudah. Pada Class tersebut lah inti dari perigkasan suatau kata, dimana pada class tersebut melakukan proses perhitungan similarity kata dan kalkulasi MMR .Class MMR dapat dilihat pada Gambar 4. 18 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 18Class Algoritma MMR

Gambar 4. 18 merupakan implementasi darri algoritma MMR yang digunakan untuk peringkasan suatu data. Pertama terdapat fungsi yang digunakan untuk menghitung seberapa kalimat tersebut terdapat kemiripannya menggunaakan libraru dari matrix. Fungsi yang pertama tersebut nantinya dipanggil pada perhitungan similarity score yang tedapat pada fungsi kedua yaitu similarityScore(). Fungsi kedua tersebut berguna untuk menentukan berapa tingkat kesamaan dari semua kalimat yang ada dan nantinya akan dibandingkan dengan semua klimat tersebut. Fungsi ketiga caculateMMR disini menghasikan kalimat mana saja yang terpilih pada teknik perningkasan MMR tersebut. Calculate MMR tersebut menerpakan rumus yang dapat dilihat pada Gambar 2.2. Hasil dari Calculate MMR tesebut berupa rentetan kalimat yang telah diurutkan berdasarkan algoritma MMR.

### Fungsi Menampilkan Summarization

Fungsi menampilkan summarization tersebut berguna dalam menampilkan visualisasi data yang tealh diringkas sehingga pengguna dapat lebih mudah dimengerti kalimat mana saja yang digunakan dalam meringkas data. Kode Program dari menampilkan hasil peringkasan data dapat dilihat pada Gambar 4. 19 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 19 Fungsi Menampilkan Hasil Summary

Gambar 4. 19 merupakan fungsi yang dibuat untuk menampilkan data ringkasan menggunakan algortima MMR tersebut. Parameter yang digunakan pun hasil dari perhitungan MMR pada pemanggilan sebuah class MMR. Terdapat 2 fungsi yang berbeda dimana fungsi pertama untuk menampilkan hasil ringkasannya saja dan fungsi kedua sebagai menampilakn visualisasi data ringkasan dengan kalimat asli dari data tersebut.

### Fungsi Simpan Data ke Database

Fungsi simpanSummary berguna dalam menyimpan data yang telah diringkas kedalam kolom field summary dan analisa sentiment. Query digunakan disini adalah update data sesuai dengan text yang di summarization menggunakan parameter id. Fungsi simpanSummary ini dapat dilihat pada Gambar 4. 20 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 20 Fungsi Simpan Data ke Database

Gambar 4. 20 merupakan tampilan dari fungsi simpan data summary. Ringkasan yang telah dihasilkan nantinya akan diupdate pada database dengan parameter update menggunakan id dari berita tersebut.

### Fungsi Summarization Engine

Fungsi utama merupakan disini pemrosesan semua teks yang ada mulai dari text preprocessing hingga pemanggilan class MMR yang dilakukan untuk summary. Kode program Fungsi utama tersebut dapat dilihat pada Gambar 4. 21 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 21 Fungsi Utama Engine Text Summarization

Gambar 4. 21 meperlihatkan proses summarization yang dilakukan dengan memanggil data dari database pada loop df[‘article] tersebut. Alur dari summarization dimulai dari pemanggilan content lalu melakukan tokenize dari pargraf ke kalimat dan data tokenizeKalimat tersebut dilakukan looping untuk memproses data tersebut dengan memanggil fungsi tanda panah kedua yaitu steamingData dengan paprameter input yaitu hasil dari cleansing data. Setelah mendapatkan data bersih dan menyimpan data asli tersebut maka akan dilakukan pemanggilan class MMR untuk menghitung score similarity dan penentuan kalimat yang mana saja akan digunakan. JIka sudah maka hasil dari summary tersebut akan disimpan ke database dengan parameter hasil summary dan id dari data tersebut. Hasil summarization akan dijelaskan pada sub bab di bawah ini.

### Fungsi Main Engine Text Summarization

Fungsi Main Engine merupakan fungsi utama dari engine Text Summarization. Pemanggilan data yang belum mempunyai summary akan dipanggil dan dilakukan peringkasan dengan memanggil fungsi dari summarization() dengan parameter row data. Penjelasan lebih lanjut dapat dilihat pada Gambar 4. 22 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 22 Fungsi Main Engine Text Summarization

Gambar 4. 22 merupakan fungsi main engine. Pemanggilan fungsi summarization() dengan parameter data yang dipanggil dari database. Hasil dari jalannya sebuah engine ini nantiya dapat dilihat di bawah ini.

### Hasil Text Summarization

Hasil summarization dari pemanggilan fungsi summarization() yang dilakukan diatas. Penjelasan lebih lanjut dapat dilihat pada Gambar 4. 23Gambar 4. 23 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 23 Hasil Summarization

Gambar 4. 23 memperlihatkan hasil dari summarization dari algoritma MMR. Pada bagian atas menampilkan hasil dari kalimat summary saja dan bagian yang kedua teks tersebut merupakan visualisasi dari teks originalnya tetapi teks yang berwarna merah tersebutlah yang digunakan untuk bahan summarization. Hasil summarization telah disimpan kedalam database yang dapat dilihat pada Gambar 4. 24 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 24 Penyimpanan Data Summary Pada Database

Gambar 4. 24 merupakan hasil summary yang telah dihasilkan dari perhitungan diatas dan akan disimpan kedalam database dengan cara mengupdate field pada data tersebut dengan parameter update menggunakan ID berita tersebut.

## Analisis Sentiment

*sentiment analysis* adalah proses memahami dan mengelompokkan emosi (positif, negatif, dan netral) yang terdapat dalam tulisan menggunakan teknik analisis teks. Sentiment analysis adalah proses penggunaan text analytics untuk mendapatkan berbagai sumber data dari internet dan beragam platform media sosial. Tujuannya adalah untuk memperoleh opini dari pengguna yang terdapat pada platform tersebut. Analisis sentiment disini menggunakan model dari lexicon. Alur kerja dari analisis sentiment ini dapat dilihat pada sebagai Gambar 4. 25 berikut.

|  |
| --- |
| C:\Users\risma\Desktop\text analisis.jpg |

Gambar 4. 25 Alur Text Analisis Sentiment

Gambar 4. 25 memperlihatkan Alur Text Analisis Sentiment. Dimulai dari pengambilan basis pengetahuan pada database berupa link path tempat basis pengathuan itu berada, selanjutnya dengan mencocokan kata yang terdapat pada kalimat text yang akan dianalisis apakah ada pada basis pengetahu Lexicon, jika ada maka akan ditambahkan sesuai score dari kata tersebut hingga hasil akhir dari proses tersebut adalah jika score tersebut kurang dari 0 maka negative,jika sama dengan 0 maka netral, dan jika lebih besar dari 0 maka positif. Output dari proses text analisis ini yaitu labling paragraph dan score yang didapatkan. Nantinya data tersebut akan disimpan pada database.

### Fungsi Analisis Sentiment

Fungsi Analisis Sentiment ini berada pada engine summarization. Pemanggilan fungsi ini nantinya berada pada fungsi Main Engine dengan pemanggilan 2 kali yaitu untuk text yang sudah disummary dan text asli dari berita tersebut. Tujuannya agar dapat membandingkan bahwa apabila setelah di summary akan merbuah sentiment tersebut.

|  |
| --- |
|  |

Gambar 4. 26 Fungsi Analisis Sentiment

Gambar 4. 26 merupakan fungis yang digunakan untuk melakukan proses sentiment analisis. Tahapan awal yaitu meload basis pengetahuan dari lexicon yang berisikan kata-kata positive maupun negative beserta dengan score dari kata tersebut. Selanjutnya kata yang akan dianalisis akan dihitung berapa jumlah terdapat kata yang mengandung positive dan negative tersebut. score tersebut kurang dari 0 maka negative,jika sama dengan 0 maka netral, dan jika lebih besar dari 0 maka positif. Hasil dari sentiment tersebut dapat dilihat pada penjelasan berikutnya.

### Hasil Analisis Sentiment

Proses analisis sentiment itu nantinya akan disimpan pada database engine. Hasil Analisis Sentiment dapat dilihat pada Gambar 4. 27 sebagai berikut.

|  |
| --- |
|  |

Gambar 4. 27 Hasil Analisis Sentiment

Gambar 4. 27 merupakan hasil dari analisis sentiment yang telah dilakukan sebelumnya. Maka dapat disimpulkan perbedaan secara signifikan yang terjadi sesudah sebuah berita di summary. Kebanyakan kata setelah disummary ini hasilnya positive.

# Kesimpulan