**TEXT SUMMARIZATION MENGGUNAKAN ALGORITMA MAXIMUM MARGINAL RELEVANCE (MMR)**

**MULTI CHANNEL ACCESS - MDI (A)**

Diajukan guna memenuhi sebagian persyaratan dalam rangka menyelesaikan Pendidikan Sarjana Strata Satu (S1) Program Studi Teknologi Informasi



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).

# Metode Pembuatan

Bagian Progress akan menjelaskan progress atau tahapan yang akan dilakukan penulis. Progress berisikan tahapan dari pembuatan *Automatic Summarization* yang dapat dibagi menjadi 2 yaitu proses pembuatan *engine* *Automatic Summarization* dan *engine Grabbing Data.* Penjelasan lebih lanjut mengenai progress akan dijabarkan sebagai berikut.

## Grabbing Data

Grabbing 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 Grabbing 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. 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. 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.

# Progress

Bagian progress ini menjelaskan tentang progress pembuatan *engine* yang sudah dikerjakan. Adapun penjelasan progress akan dibagi menjadi 2 bagian pengerjaan yaitu Grabing Data dan Text Summarization yang akan dijelaskan sebagai berikut.

## Grabbing Data

Grabbing data pada pembuatan engine tersebut menggunakan library dari BeautifulSoup untuk mengambil data pada sebuah website. Setelah mendalami mempelajari library tersebut hanya dapat mengambil data berdasarkan tag html yang terdapat pada domain tersebut, sehingga jika ingin mengambil data scraping dari berbeda domain maka diperlukannya analisa *source code* yang terdapat pada domain tersebut. Progress yang telah dibuat kali ini mengambil data scraping pada salah satu website berita CNBC Indonesia berdasarkan berita terbaru atau kata kunci yang ingin dimasukan oleh user. Penjelasan lebih lanjutnya akan dijelaskan pada gambar berikutnya.

### Fungsi Main Engine Grabbing Data

Fungsi main dari *engine grabbing* data ini merupakan inti dari *engine* tersebut dimana fungsinya berguna dalam memanggil fungsi-fungsi yang telah dibuat pada *engine* tersebut. Fungsi main dari *engine grabbing* data ini dapat dilihat pada Gambar 3.1 sebagai berikut.

|  |  |
| --- | --- |
|  |  |

Gambar 3. Fungsi Engine Utama Grabbing Data

Gambar 3.1 merupakan tampilan dari kode program yang melakukan pemanggilan fungsi yang telah dibuat diawal. Fungsi dari engine utama ini berguna dalam memanggil fungsi-fungsi yang telah dibuat. getAllPageWebsite (url, search) berguna dalam mengambil semua halam yang terdapat pada website berita cbncindonesia dimana itu nantinya bisa digunakan pada pengambilan data urlDetail disetiap page yang ada, parameter yang digunakan pengguna hanya memasukan pencarian berita pada variable search saja dan jika memang user tidak ingin memasukan pencarian maka default dari pencarian berita adalah berita yang terbaru. getUrlInPage(url,search,page) fungsi ini berguna dalam mengambil data url detail menuju berita tersebut, berapakah ada total berita pada setiap halam, dengan menggunakan parameter search sebagai pencarian dan *page* itu bisa dengan memasukan *var* AllPage tadi. scarpingDataDetail(urlDetail,mydb) setelah mendapatkan urldetail dari masing-masing berita maka akan dilakukan pengambilan data yang akan diambil. Penjelasan lebih lanjut mengenai fungsi yang telah dibuat akan dijelaskan pada penjelasan beritkutnya.

### Fungsi Simpan dan Validasi Database

Fungsi *database* dari *engine grabbing* data adalah melakukan penyimpanan data dan melakukan validasi data pada *database*. Fungsi *database* ini dapat dilihat pada Gambar 3.2 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Simpan Data dan Validasi Database

Gambar 3.2 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 tersebut 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 getAllPageWebsite

Fungsi getAllPageWebsite dari engine grabbing data adalah melakukan pengambilan data page yang terdapat pada pencarian berita tersebut. Fungsi getAllPageWebsite ini dapat dilihat pada Gambar 3.3 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Mengambil Jumlah Page pada Domain Website CNBC

Gambar 3.3 merupakan proses pengambilan data halaman pada domain CNCB Indonesia dan juga anda dapat mengatur configurasi kata kunci dari berita yang akan dicari pada website tersebut. Nah untuk mencoba mengambil data dari lain website nantinya rencanannya akan dibuatkan validasi pencarian website tersebut. Mengambil semua data halaman tersebut berguna untuk mengambil semua data yang berkatian dari kata kunci tersebut, karena biasanya dia user hanya mengambil data pada page pertama saja. Hasil pemanggilan dari fungsi tersebut berupa integer dari jumlah page yang terdapat pada pencarian website tersebut.

### Fungi Mengambil URL Detail Berita (getUrlInPage)

Fungsi getUrlInPage dari engine grabbing data adalah melakukan pengambilan data url dari masing-masing berita yang ada pada jumlah page yang telah di tentukan. Fungsi getUrlInPage ini dapat dilihat pada Gambar 3.4 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Mengambil URL Berita Detail

Gambar 3.4 merupakan penjelasan pembuatan fungsi dari menjelaskan tentang pengambilan url dari headline berita yang ada pada CNBC Indonesia. 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.

### Fungsi Mengambial 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 3.4 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Mengambil Semua Data Pada URL Detail yang diSimpan

Gambar 3.5 menjelaskan terkait dengan pengambilan semua data pada CNBC Indonesia. Skema pengambilan data yang telah dibuat saya buatkan validasi pada *database* agar tidak terdapat data *double* yang disimpan pada database. Sebagai parameter pembeda saya menggunakan link dari detail berita tersebut untuk menentukan apakah sebuah data tersebut sudah ada pada database. Ketika tidak ada maka 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 3.6 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Hasil Penyimpanan Pada Database

Gambar 3.6 merupakan hasil dari *grabbing* *database* dan melakukan penyimpanan kedalam *database* mysql yang masih berada dilocal computer saya. Data yang diambil itu berupa judul berita, label berita, penulis, kapan terbitnya, url berita tersebut, dan content dari berita tersebut.

## 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.

### Deklarasi Library dan Class Algoritma

Deklarasi library dan Class Algoritma merupakan tahapan awal dalam membuat sebuah engine dalam menggunakan fungsi pengolahan suatu data. Deklarasi library ini dapat dilihat pada Gambar 3.7 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Kode Program Deklarasi Library dan sebuah Class

Gambar 3.7 merupakan tahapan awal dalam membuat sebuah engine, yang dimana terdapat banyak library yang digunakan untuk membuat engine tersebut. Library mysql.connector berfungis sebagai pengolahan data agar terhubung langsung dengan sebuah database. Library NLTK sebagai pengolahan sebuah paragraph dengan melakukan tokenize sebuah paragraph tersebut menjadi tokenize perkalimat. Library re sebagai membersihkan kalimat yang masih ada beberapa elemet tag HTML yang ada. Library Sastrawi sebagai proses steaming data yaitu berguna dalam menjadikan sebuah kata menjadi kata dasarnya. Library pandas sebagai pengolahan data dengan menampilkan visualisasi tabel. Library stopword yaitu berguna dalam melakukan kata yang tidak penting pada suatu paragraph. Library String sebagai mebersihkan data dari beberapa tanda baca. Library colored sebagai mewarnakan sebuah text sesuai dengan parameter yang dimasukan.

### Deklarasi Connection Database dan Fungsi Steaming Data

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. Deklarasi connection database dan steaming data ini dapat dilihat pada Gambar 3.8 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Deklarasi Connection Database dan Fungsi Steaming Data

Gambar 3.8 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.

### Mengambil Data pada Database

Mengambil data pada database yang digunakan nantinya untuk melakukan summary. Pengambilan data pada database ini dapat dilihat pada Gambar 3.9 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Mengambil data pada Database

Gambar 3.9 merupakan proses dari pengambilan data pada database yang nantinya akan digunakan untuk proses summarization.

### Fungsi Load Stopword

Fungsi loadstopword berguna dalam menghilangkan kata-kata yang tidak penting seperti kata hubung sebuah kalimat. Fungsi loadstopword ini dapat dilihat pada Gambar 3.10 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Fungsi LoadStopword

Gambar 3.10 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.

### Fungsi Simpan Summary ke Database

Fungsi simpanSummary berguna dalam menyimpan data yang telah diringkas kedalam kolom field summary. Query digunakan disini adalah update data sesuai dengan text yang di summarization. Fungsi simpanSummary ini dapat dilihat pada Gambar 3.10 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Simpan Data Summary

Gambar 3.11 merupakan tampilan dari fungsi simpan data summary. Ringkasan yang telah dihasilkan nantinya akan diupdate pada database dengan parameter update menggunakan id dari berita tersebut.

### 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 3.12. Penjelasan lebih lanjut mengenai beberap fungsi yang telah dibuat tersebut akan dijelaskan sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Fungsi Text Preprocessing Data

Gambar 3.12 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 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 3.13 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Hasil Summarization Data

Gambar 3.13 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.

### 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 3.14 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Class Algoritma MMR

Gambar 3.14 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. Caculate MMR tersebut menerpakan rumus yang dapat dilihat pada Gambar 2.2. Hasil dari calculateMMR tesebut berupa rentetan kalimat yang penting pada suatu paragraph.

### Fungsi Utama Pemanggilan Semua Fungsi yang Ada

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 3.15 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Kode Program Fungsi Utama Engine Summarization

Gambar 3.15 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 3.2.10 sebagai berikut.

### Hasil Summarization

Hasil summarization menggunakan algorimta MMR akan ditampilkan pada Gambar 3.16 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Hasil Summarization Algoritma MMR

Gambar 3.16 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 3.17 sebagai berikut.

|  |
| --- |
|  |

Gambar 3. Data Tersimpan di Database

Gambar 3.17 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.