

TRANSFORMATIKA, Vol.17, No.2, January 2020, pp. 188 - 199

ISSN: 1693-3656,

journals.usm.ac.id/index.php/transformatika

■ page 188

Pembelajaran *Text Preprocessing* berbasis Simulator Untuk Mata Kuliah Information Retrieval

Latius Hermawan¹, Maria Bellanar Ismiati²

Jalan Bangau No.60, UNIKA Musi Charitas, e-mail: tiuz.hermawan@ukmc.ac.id

Jalan Bangau No.60, UNIKA Musi Charitas, e-mail: bella@ukmc.ac.id

ARTICLE INFO

Article history:

Received 13 November 2019

Received in revised form 15 January 2020

Accepted 27 January 2020

Available online 31 January 2020

ABSTRACT

Preprocessing is an important task and step in Information Retrieval. Information Retrieval (IR) is used to decide which documents in a collection must be taken to meet the user's information needs. The comparison is done in 4 steps in preprocessing, namely Case folding, Tokenizing, Filtering, Stemming.

All stages of the preprocessing are done manually during the teaching-learning process. The lecturer explained by writing and explaining one by one the correct words at each stage and students also manually reviewed each stage. Simulator is needed to help solve existing problems. The simulator has been successfully created by displaying 6 menus. The simulator has succeeded in outputting each stage according to the manual theory. From the results of the testing questionnaire, > 90% of respondents thought that using this simulator made it easier to work on the four stages of text pre-processing so that the time needed also became efficient.

Keywords : ir, simulator, text, preprocessing

1. Pendahuluan

Preprocessing adalah tugas dan langkah penting dalam *Text Mining*, *Natural Language Processing (NLP)* dan *Information Retrieval (IR)*. Di bidang *Text Mining*, data *preprocessing* digunakan untuk mengekstraksi pengetahuan yang menarik dan penting serta dari data teks yang tidak terstruktur. *Information Retrieval (IR)* digunakan untuk memutuskan dokumen mana dalam koleksi yang harus diambil untuk memenuhi kebutuhan pengguna akan informasi [1]. Oleh karena itu, keputusan pengambilan dibuat dengan membandingkan istilah kueri dengan istilah indeks (kata atau frasa penting) yang muncul dalam dokumen itu sendiri.

Perbandingan tersebut dilakukan dalam 4 langkah di *preprocessing*, yaitu *Case folding*, *Tokenizing*, *Filtering*, *Stemming* [2]. Dalam kaitannya dalam proses mengajar dalam perkuliahan, semua tahapan *preprocessing* tersebut dilakukan secara manual saat proses belajar-mengajar. Dosen menjelaskan dengan menulis dan menjabarkan satu per satu kata yang benar di setiap

Received November 13, 2019; Revised January 05, 2020; Accepted January 27, 2020

tahap dan mahasiswa pun melakukan review masing-masing tahapan secara manual pula. Selain itu, pembelajaran dengan cara manual tersebut menjadi monoton dan tidak interaktif antara dosen dan mahasiswa. Hasil dari kuesioner awal yang diberikan kepada mahasiswa yang pernah mendapat materi mengenai *text pre-processing* mengemukakan bahwa para mahasiswa mengalami kesulitan saat melakukan keempat tahapan *pre-processing* secara manual karena membutuhkan waktu yang lama untuk pengecekan secara berulang-ulang satu per satu dari masing-masing kata yang akan diproses. Salah satu penyebabnya adalah penulisan ulang setiap contoh kalimat yang ada pada setiap tahap *preprocessing text* yang membuat bosan para mahasiswa saat menganalisis teks.

Selain kebosanan dan suasana monoton yang dirasakan mahasiswa, hal yang sama juga dialami oleh dosen ketika menjadi pengampu mata kuliah *Information Retrieval*. Proses *pre-processing* yang dilakukan secara manual mengakibatkan mahasiswa dan dosen mengalami kendala untuk dapat melakukan *preprocessing text* secara lebih cepat, akibatnya proses belajar mengajar menjadi tidak efisien. Akibatnya, mahasiswa dan dosen seringkali kehabisan waktu pada saat proses belajar untuk mata kuliah tersebut. Kendala lain yang juga ditemui akibat masalah tersebut adalah penulisan ulang setiap contoh kalimat yang ada pada setiap tahap *preprocessing text*, dimana terdapat 4 tahapan yang ada yaitu *Case folding*, *Tokenization*, *Filtering*, *Stemming*.

Salah satu cara untuk mengatasi permasalahan tersebut adalah dengan merancang model pembelajaran yang baik dan interaktif. Tao *et al* (2011) menyebutkan bahwa alat bantu pembelajaran yang dirancang dengan baik, terbukti secara efektif mengurangi kesulitan dalam mempelajari topik-topik yang kompleks [3]. Dalam penelitiannya [4] menemukan bahwa penggunaan *CrypTool* sebagai alat bantu pembelajaran kriptografi meningkatkan nilai (*grade*) mahasiswa hingga 10% dibandingkan dengan pembelajaran tanpa bantuan *CrypTool* [4]. Temuan serupa juga diungkapkan oleh Katz (2015), bahwa penggunaan alat bantu (simulator) dalam pembelajaran kriptografi dapat meningkatkan kemampuan mahasiswa untuk menyelesaikan kasus-kasus kriptografi dibandingkan dengan cara manual [5]. Berdasarkan penjelasan di atas, maka peneliti membuat suatu simulator untuk membantu menyelesaikan masalah yang ada dengan judul penelitian Pembelajaran *Text Preprocessing* berbasis Simulator Untuk Mata Kuliah *Information Retrieval*.

1.1 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

Bagaimana membuat pembelajaran *text pre-processing* berbasis simulator untuk mata kuliah *Information Retrieval* agar waktu yang dibutuhkan untuk proses belajar-mengajar menjadi lebih efisien ?

1.2. Tujuan

Adapun tujuan Penelitian ini adalah sebagai berikut

Untuk membuat simulator pembelajaran *text pre-processing* di mata kuliah *Information Retrieval* agar waktu yang dibutuhkan untuk proses belajar-mengajar menjadi lebih efisien.

1.3. Studi Pustaka

1. *Information Retrieval*

Information Retrieval (IR) adalah proses menemukan bahan (biasanya dokumen) dari bersifat terstruktur (biasanya teks) yang memenuhi kebutuhan informasi dari dalam koleksi besar yang disimpan di computer [2]. *Information Retrieval System* (IRS) merupakan sistem yang digunakan untuk menemukan kembali dokumen yang relevan terhadap kebutuhan *user* dari suatu kumpulan informasi berdasarkan kata kunci atau *keywords* atau *query* dari *user*. Suatu dokumen yang memiliki rangking yang lebih tinggi daripada dokumen lain akan dianggap lebih relevan terhadap *query*.

1. *Preprocessing*

Berdasarkan ketidak teraturan struktur data teks, maka proses sistem temu kembali informasi ataupun *text mining* memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur. Tahap *Text Preprocessing* adalah tahapan dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Proses *preprocessing* ini meliputi (1) *case folding*, (2) *tokenizing*, (3) *filtering*, dan (4) *stemming* [2].

2. Simulator

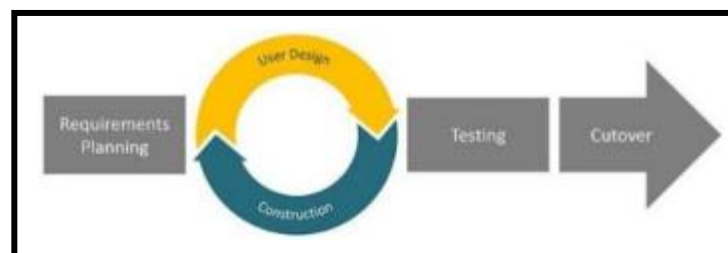
Simulasi merupakan proses perancangan model matematis atau logis dari sistem nyata, melakukan eksperimen terhadap model dengan menggunakan komputer untuk menggambarkan, menjelaskan dan memprediksi perilaku sistem [8]. Sedangkan [9] simulasi didefinisikan sebagai sekumpulan metode dan aplikasi untuk menirukan atau merepresentasikan perilaku dari suatu sistem nyata, yang biasanya dilakukan pada komputer dengan menggunakan perangkat lunak tertentu.

3. PHP dan My SQL

PHP (*HyperText Preprocessor*) adalah sebuah bahasa utama *script server-side* yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop. Menurut [6], dalam bukunya yang berjudul Pemrograman Web Dengan PHP, menyebutkan bahwa : "PHP merupakan secara umum dikenal dengan sebagai bahasa pemrograman script – script yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side*".

4. Rapid Application Development (RAD)

RAD atau pengembangan aplikasi cepat adalah pendekatan berorientasi objek untuk pengembangan system yang meliputi metode pengembangan serta perangkat lunak [7]. RAD dilihat sebagai metodologi pengembangan sistem yang berusaha untuk mengatasi perubahan persyaratan kebutuhan *user* dan merekomendasikan RAD untuk mengembangkan aplikasi berbasis web. RAD merupakan model proses perangkat lunak yang menekankan pada daur pengembangan hidup yang singkat. Gambar model proses RAD dapat dilihat pada gambar 5 seperti di bawah ini.



Gambar 1. *Rapid Application Development* (Sumber : Pressman, 2010)

1.4. Penelitian Terdahulu

Penelitian yang dilakukan oleh [10] menggunakan pemrograman Basic mengemukakan bahwa Pada perkuliahan komputer, kriptografi menjadi salah satu mata kuliah yang diajarkan. Pembelajaran kriptografi menjadi sangat penting bagi mahasiswa

komputer agar mengetahui cara-cara mengamankan data. Proses komputasi dan perhitungan yang rumit membuat mahasiswa mengalami kesulitan menemukan hasil dari enkripsi yang dilakukan. Oleh karena itu penulis akan membuat aplikasi pembelajaran bagi mahasiswa khususnya pembelajaran kriptografi kunci publik ElGamal. Hasil yang diberikan dari penelitian ini yaitu dengan adanya aplikasi ini, mahasiswa dapat mengetahui proses enkripsi dan dekripsi pada algoritma ElGamal secara detail dalam hal proses matematika yang dilakukan. Algoritma ElGamal juga bisa dimanfaatkan untuk mengirimkan sebuah pesan rahasia yang sangat rahasia, yaitu kunci dari sebuah kriptografi simetris. Pada algoritma ElGamal suatu plaintext yang sama akan dienkripsi menjadi ciphertext yang berbeda-beda.

Sedangkan pada penelitian [11] mengatakan dalam penelitiannya menggunakan Java sebagai tool membuat aplikasi bahwa pertumbuhan sosial media yang sangat pesat tidak membuat twitter ditinggal oleh penggunanya. Twitter merupakan sebuah sosial media yang dimanfaatkan oleh penggunanya untuk berbagi informasi. Tidak banyak karakter yang dapat dimasukkan pada komentar di twitter. Keterbatasan karakter tersebut membuat para peneliti memakai data tersebut untuk penelitiannya. Dari uraian tersebut maka diperlukan adanya penanganan yang ekstra pada saat tahap pre-processing atau tahap persiapan data. Pada penelitian kali ini akan membahas beberapa teknik penanganan data komentar dari twitter untuk proses data mining. Hasilnya yaitu setiap proses pada tahap pre-processing data *text* dari twitter yang telah dibuat, memiliki hasil yang sangat baik, sehingga dapat digunakan untuk penelitian lebih lanjut tentang data mining atau *text* mining.

Selanjutnya oleh [12] dalam penelitiannya yang berbasis web simulator mengatakan pembelajaran Bahasa Indonesia bagi siswa sekolah dasar amatlah sulit, khususnya mulai kelas satu sampai kelas empat dalam hal pemahaman penalaran kalimat dalam Bahasa Indonesia. Dibutuhkan suatu metode yang dapat mempermudah pemahaman kalimat bahasa Indonesia menggunakan metode maximum marginal relevance. Metode ini dapat mengurangi redundansi dalam perangkaian kalimat pada dokumen dan memiliki lima tahap dalam pencarian *text preprocessing* yaitu pemecahan kalimat, *case folding*, *tokenizing*, *filtering*, dan *stemming*. Proses selanjutnya menghitung bobot *tf-idf*, bobot *query relevance* dan bobot *similarity*. Aplikasi ini telah di ujicoba secara acak pada siswa sekolah dasar kelas satu sampai kelas empat. Hasilnya yaitu Aplikasi pembelajaran yang dibuat oleh penulis telah di ujicoba pada 10 siswa sekolah dasar kelas satu sampai kelas empat secara random dengan kesimpulan yaitu: siswa dapat lebih memahami penalaran pelajaran Bahasa Indonesia dengan mudah terutama pada soal essay.

Peneliti [14] juga melakukan penelitian tentang simulasi *stemming* terhadap kata porno yang berbasis Java yaitu bermula dari informasi yang mengarah pada isi yang mengandung porno saat ini sangat mudah diterima oleh anak yang masih belum siap menerimanya dari berbagai media diantaranya adalah CD, majalah, televisi, internet, dan lain sebagainya, bahkan di dunia pendidikan baik sengaja atau tidak sengaja mendapat informasi yang mengarah pada pornografi. Maka dalam penelitian ini dikembangkan sebuah aplikasi yang mampu mencegah konten porno pada soal ujian berbahasa Indonesia. Aplikasi yang dikembangkan terdiri dari 5 (lima) modul yaitu *Tokenizing*, *Affix Porn Filtering*, *Stemming*, *Root Porn Filtering*, and *Phrase Porn Filtering*. Hasil yang didapat yaitu Dalam penelitian ini telah berhasil dibuat sebuah aplikasi *text filtering* untuk mencegah konten porno pada soal berbahasa Indonesia berbasis keyword pada pertanyaan-pertanyaan yang dibuat untuk soal ujian.

2. Metode Penelitian

2.1. Tahapan-tahapan penelitian yang dilakukan disesuaikan dengan model *Rapid Application Development* (RAD) yang terdiri dari beberapa fase, yaitu :

- a. *Requirements Planning* (Perencanaan Persyaratan)
 1. Penentuan masalah di tempat penelitian
 2. Penjabaran lebih detail mengenai *requirement* di tempat penelitian
 3. Penyebaran kuisioner untuk mendukung masalah yang dijabarkan di latar belakang penelitian
 4. Studi pustaka dan studi literatur sebagai bahan referensi yang dapat mendukung penelitian ini
- b. *User Design*
 1. Penentuan tools yang digunakan dalam penelitian ini
 2. Perancangan interface untuk Simulator *Text Pre-processing*
- c. *Konstruksi*
 1. Proses coding untuk Simulator *Text Pre-processing*
 2. Pengecekan bug dan error yang masih ada
- d. *Testing / Evaluasi*
 1. Pengenalan simulator ke beberapa mahasiswa (36 orang) yang sudah pernah mendapat materi *text pre-processing*
 2. Pengujian simulator ke beberapa mahasiswa (36 orang) yang sudah pernah mendapat materi *text pre-processing*
 3. Pengujian simulator ke beberapa mahasiswa (36 orang) yang sudah pernah mendapat materi *text pre-processing*
 4. Penyebaran dan Pengisian Kuisioner ke responden setelah responden mencoba simulatornya
 5. Analisis data hasil kuisioner akhir
 6. Pembuatan Laporan Penelitian beserta kesimpulan terhadap hasil penelitian dan pengujian
- e. *Cutover*

Tahap ini merupakan tahap dimana aplikasi sudah siap dipakai saat proses belajar mengajar dilangsungkan sehingga saat proses tidak lagi menggunakan cara manual tetapi menggunakan simulator yang dapat menghasilkan *output* secara otomatis agar prosesnya menjadi lebih efisien.

2.2. Prosedur perancangan yang dilakukan adalah dengan membuat *interface* dari Simulator *Text Pre-processing*. Proses-proses yang dilakukan dalam perancangan *interface* simulator ini adalah:

1. Proses pembuatan rancangan menu-menu simulator
2. Proses pengumpulan materi-materi yang akan dimasukkan ke dalam simulator *text pre-processing*
3. Proses *coding interface* dan fungsionalitas sistem/simulatornya
4. Proses implementasi, yaitu uji coba oleh peneliti untuk mengecek *bug* dan *error* sebelum diberikan kepada pengguna dan publik
5. Proses pengujian kepada *end user* merupakan tahap terakhir yang dilakukan untuk mendapatkan *feedback* dari pengguna mengenai simulator yang telah dibuat. Pengujian dilakukan kepada beberapa mahasiswa yang sudah pernah mendapat materi mengenai *text pre-processing*.
6. Setelah *end user* mencoba simulator, dilakukan proses penyebaran dan pengisian kuisioner ke masing-masing responden

2.3. Metode Pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

- a. Pembagian kuisioner untuk mendukung masalah di latar belakang penelitian

- b. Analisis data hasil kuisioner awal
- c. Tinjauan Pustaka (Landasan Teori dan Studi Literatur)

2.4. Analisis data dalam penelitian ini dilakukan dengan perhitungan statistika deskriptif untuk mendapatkan hasil pengujian simulator ini ke 36 mahasiswa UKMC yang sudah pernah mendapat materi mengenai *text pre-processing*.

3. Hasil dan Pembahasan

3.1. Tampilan Simulator *Text Pre-processing*

Alamat *website* simulator ini adalah [http://text-preprocessing .indoweb.xyz/](http://text-preprocessing.indoweb.xyz/). Simulator ini mempunyai 6 menu, yaitu beranda, bantuan, materi, *pre test*, simulator, dan *post test*. Masing-masing menu dibuat untuk lebih menjelaskan materi *pre-processing* dengan lebih detail. *Tampilan / interface* Simulator *Text Pre-processing* akan dijabarkan dengan lebih jelas dan lengkap di bawah ini.

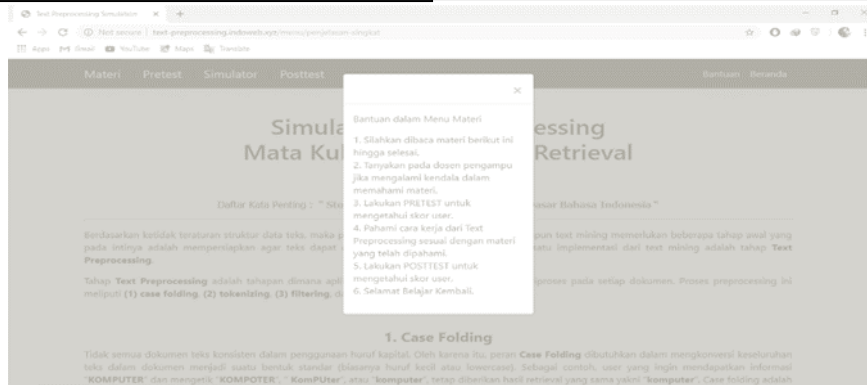
1. Tampilan awal / Menu Beranda



Gambar 6. Tampilan Menu Awal / Beranda

Alamat *website* dari simulator ini adalah <http://text-preprocessing .indoweb.xyz/> dengan menggunakan *domain* yang tidak berbayar. Tampilan awal berisi judul simulator, logo UKMC, logo RistekDikti, tombol Mulai Simulator, dan nama peneliti. Setelah *user* mengetahui maksud dan tujuan dibuatnya aplikasi ini, *user* dapat menekan tombol Mulai Simulator agar dapat langsung masuk ke menu utama, yaitu menu materi untuk mempelajari lebih detail mengenai *text pre-processing*. Sebelum masuk ke menu materi, sebelumnya akan dijelaskan lebih detail mengenai menu bantuan yang seharusnya dibuka/dibaca terlebih dahulu sebelum memulai menu-menu yang lain.

2. Tampilan Menu Bantuan

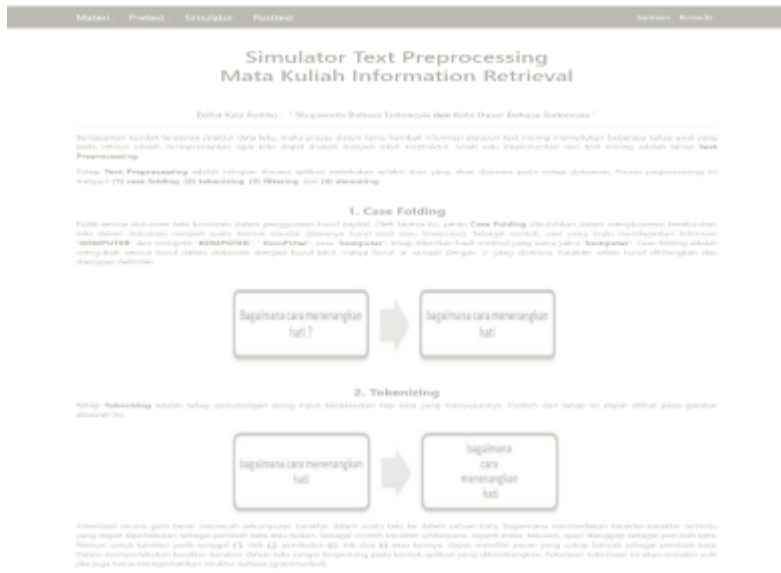


Gambar 7. Tampilan Menu Bantuan

Menu bantuan di atas merupakan menu yang dibuat agar *user* tidak mengalami kendala saat mengakses simulator *text pre-processing*. Di dalam menu ini berisi beberapa poin yang menjelaskan menu-menu lain seperti penjelasan *pre test* dan *post test*. Oleh karena itu, *user* sebaiknya mengakses dan membaca menu bantuan ini terlebih dahulu sebelum beralih ke menu-menu lainnya dalam simulator ini.

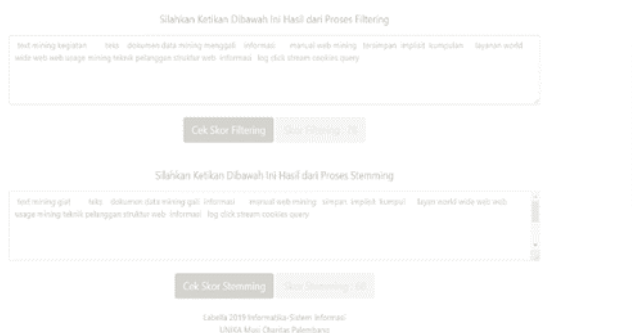
3 Tampilan Menu Materi (Tampilan ini sama dengan tampilan ketika *user* menekan tombol Mulai Simulator)

Pada menu ini, *user* diberikan penjelasan mengenai 4 tahapan *text pre-processing*, yaitu *case folding*, *tokenizing*, *filtering*, dan *stemming*. Selain berupa tulisan dan contoh, pada bagian ini diberikan pula gambar-gambar yang mendukung untuk masing-masing tahapan. Hal ini dilakukan agar *user* yang membaca menu materi ini tidak hanya mendapat teori tetapi juga langkah-langkah pastinya dengan menggunakan contoh soal. Selain berupa materi yang dijelaskan secara detail, pada menu ini diberikan pula link untuk melihat daftar kata-kata yang masuk ke dalam *stopword* dan daftar kata-kata dasar. Link *stopword* digunakan untuk tahapan *filtering* sedangkan daftar kata dasar untuk tahap *stemming*.



Gambar 8. Tampilan Menu Materi di Simulator



Gambar 13. Skor *Case folding* dan *Tokenizing*Gambar 14. Skor *Filtering* dan *Stemming*

5 Tampilan Menu Simulator

Pada Gambar 15 di bawah ini merupakan tampilan menu Simulator yang dapat berulang kali *user* gunakan dengan kalimat yang bisa berganti-ganti. *User* memasukkan kalimatnya di bagian kotak putih di bawah tulisan Silahkan Masukkan *Text* Percobaan. Selanjutnya *user* dapat menekan tombol *case folding*, *tokenizing*, *filtering*, dan *stemming* untuk mendapatkan hasil akhir dari *text pre-processing*.

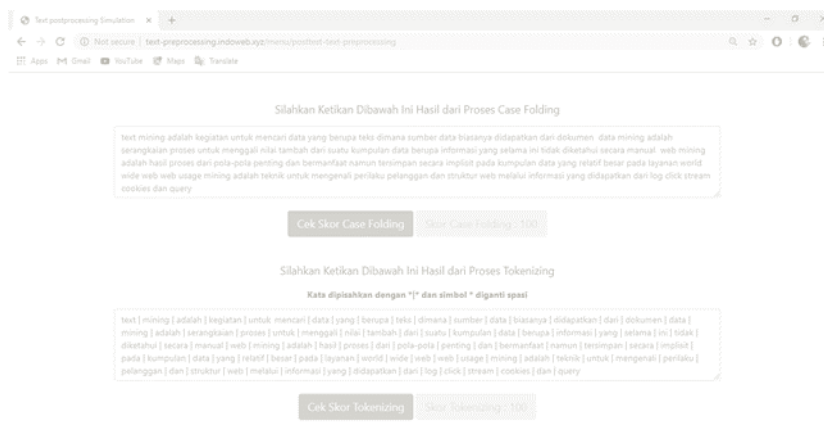
Gambar 15. Tampilan Simulator *Text Pre-processing* beserta Contohnya

Proses tahapan *text pre-processing* di simulator ini berjalan dengan baik, yaitu ketika ada suatu kalimat utuh, simulator ini dapat menghasilkan *output* untuk *text pre-processing* yang sudah melewati 4 tahapan di atas. Saat *user* menekan tombol *case folding*, tampilan kalimatnya sudah dapat merubah semua karakter huruf menjadi huruf kecil dan semua karakter lain seperti tanda baca dan tanda hubung juga sudah dapat dihilangkan oleh simulator ini. Selanjutnya saat *user* menekan tombol *tokenizing*, tampilan kalimatnya sudah dibagi-bagi per token/kata yang dipisahkan dengan tanda garis lurus (|). Sampai pada tahap ini, tidak ada kata yang dihilangkan.

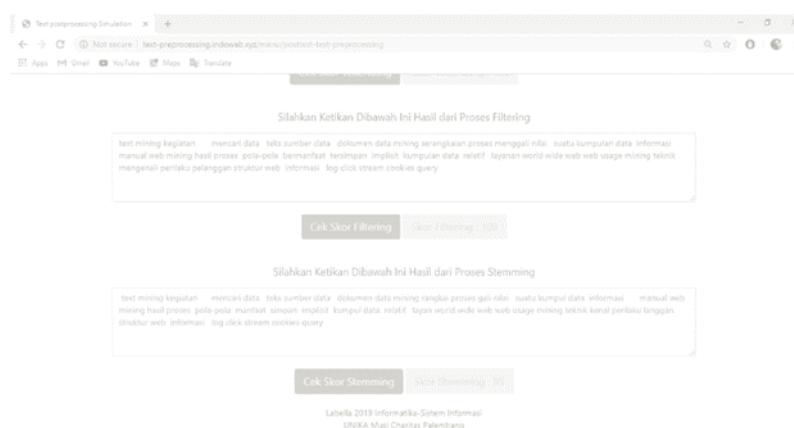
Tahap selanjutnya adalah *filtering* yaitu menghilangkan beberapa kata yang dianggap kurang/tidak penting berdasarkan *stopwords* Bahasa Indonesia. Isi dari *stopwords* contohnya adalah kata ganti orang, kata hubung, dan masih banyak lagi. Apabila kata yang di dalam kalimat tersebut ada di dalam daftar *stopwords*, maka kata tersebut harus dihilangkan karena sudah dianggap kurang/tidak penting. Pada tahap ini sudah mulai menghilangkan beberapa kata dari keseluruhan kalimat utuh yang dijadikan contoh. Simulator ini sudah berhasil mencocokkan masing-masing kata yang sudah dipisahkan dari *tokenizing* dengan kata-kata yang ada di dalam *stopwords*.

Tahap terakhir adalah *stemming* yaitu tahap menghilangkan semua imbuhan dan akhiran yang melekat di beberapa kata hasil dari *filtering*. Semua hasil dari *stemming*, harus berupa kata dasar dan tidak ada lagi imbuhan maupun akhiran yang masih melekat. Prosesnya adalah simulator ini membuang imbuhan dan akhiran dari suatu kata. Setelah itu dibandingkan dengan kata dasar di *link* kata dasar Bahasa Indonesia sehingga didapatkan kata dasar aslinya. Simulator ini sudah berhasil menampilkan kata dasar dari beberapa kata hasil *filtering* sehingga hasil akhirnya sudah lebih singkat dan berupa kata dasar.

6 Tampilan Menu *Post Test* beserta Skor untuk masing-masing Tahapan



Gambar 19. Skor *Case folding* dan *Tokenizing* untuk *Post Test*



Gambar 20. Skor *Case folding* dan *Tokenizing* untuk *Post Test*

3.2. Hasil Pengujian Akhir (Kuisisioner)

1. Berdasarkan pengolahan kuisisioner terlihat bahwa seluruh mahasiswa tertarik untuk menggunakan simulator edukasi ini dengan persentase sebanyak 100%. Jumlah mahasiswa yang mengisi kuisisioner tersebut sebanyak 36 mahasiswa. Hal tersebut menunjukkan bahwa seluruh mahasiswa yang mengisi kuisisioner ini tertarik menggunakan simulator ini.

2. Berdasarkan pengolahan kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa tampilan simulator ini menarik dengan persentase sebanyak 72% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa tampilan *game* edukasi ini tidak menarik dengan persentase sebanyak 28%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa tampilan simulator edukasi ini menarik.
3. Berdasarkan pengolahan kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa simulator *text pre-processing* ini mudah digunakan dengan persentase sebanyak 92% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa simulator *text pre-processing* ini tidak mudah digunakan dengan persentase sebanyak 8%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa simulator *text pre-processing* ini mudah digunakan.
4. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa *fitur* pada Simulator ini lengkap dengan persentase sebanyak 75% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa *fitur* pada Simulator ini tidak lengkap dengan persentase sebanyak 25%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa *fitur* pada Simulator ini lengkap.
5. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa tampilan informasi dengan topik pada alur aplikasi simulator telah sesuai dengan persentase sebanyak 97% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa tampilan informasi dengan topik pada alur aplikasi simulator tidak sesuai dengan persentase sebanyak 3%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa tampilan informasi dengan topik pada alur aplikasi simulator telah sesuai.
6. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa informasi yang disajikan telah sesuai dengan topik dengan persentase sebanyak 94% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa informasi yang disajikan tidak sesuai dengan topik dengan persentase sebanyak 6%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa informasi yang disajikan telah sesuai dengan topik.
7. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa bahasa yang digunakan dalam simulator ini mudah dipahami dengan persentase sebanyak 83% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa bahasa yang digunakan dalam simulator ini tidak mudah dipahami dengan persentase sebanyak 17%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa bahasa yang digunakan dalam simulator ini mudah dipahami.
8. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa simulator membuat lebih mudah dalam meningkatkan pengetahuan mengenai *text pre-processing* dengan persentase sebanyak 97% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa simulator tidak membuat lebih mudah dalam meningkatkan pengetahuan mengenai *text pre-processing* dengan persentase sebanyak 3%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa simulator membuat lebih mudah dalam meningkatkan pengetahuan mengenai *text pre-processing*.
9. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa dengan menggunakan simulator dapat lebih terdorong untuk belajar *text pre-processing* dengan persentase sebanyak 97% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa dengan menggunakan simulator, responden tidak dapat lebih terdorong untuk belajar *text pre-processing* dengan persentase sebanyak 3%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa dengan menggunakan simulator dapat lebih terdorong untuk belajar *text pre-processing*.
10. Berdasarkan pengolahan hasil kuisioner terlihat bahwa lebih banyak mahasiswa yang berpendapat bahwa dengan menggunakan simulator mampu memberikan tambahan pengetahuan

mengenai *text pre-processing* dengan persentase sebanyak 97% sedangkan masih ada beberapa mahasiswa yang berpendapat bahwa dengan menggunakan simulator tidak mampu memberikan tambahan pengetahuan mengenai *text pre-processing* dengan persentase sebanyak 3%. Hal tersebut menunjukkan bahwa hampir seluruh mahasiswa yang mengisi kuisioner ini berpendapat bahwa dengan menggunakan simulator mampu memberikan tambahan pengetahuan mengenai *text pre-processing*.

4. Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian ini setelah peneliti berhasil membuat Simulator *Text Pre-processing* adalah :

Simulator untuk pembelajaran *text pre-processing* telah berhasil dibuat dengan menampilkan 6 menu, yaitu menu beranda, bantuan, materi, *pre test*, simulator, dan *post test*. Simulator ini juga telah berjalan dengan baik karena telah berhasil mengeluarkan *output* untuk masing-masing tahapan yang sesuai dengan teori manualnya. Dari hasil kuisioner pengujian, > 90% responden berpendapat bahwa dengan menggunakan simulator ini lebih mempermudah pengerjaan keempat tahapan *text pre-processing* sehingga waktu yang dibutuhkan juga menjadi efisien.

References

- [1] Gurusamy, V., (2014). *Preprocessing Techniques for Text Mining*. Conference Paper RTRICS, India
- [2] Manning, D. (2008). *Introduction to Information Retrieval*. Cambridge University Press New York, NY, USA
- [3] Tao, J., Ma, J., Mayo, J., Shene, C.-K., & Keranen, M. (2011). *DESvisual: A Visualization Tool for the DES Cipher*. Journal of Computing Sciences in Colleges, 27(1), 81-89
- [4] Adamovic, S., Branovic, I., (2011). *Teaching interactive cryptography: the case for CrypTool*. IEEE Conference, ICEST 2011
- [5] Zelvina, A., (2012), *Perancangan Aplikasi Pembelajaran Kriptografi Kunci Publik ElGamal Untuk Mahasiswa*. Jurnal Dunia Teknologi Informasi Vol. 1, No. 1, (2012) 56- 62
- [6] Betha Sidik. (2012). *Pemrograman Web dengan PHP*. Informatika,. Bandung.
- [7] Pressman, Roger S. (2010). *Software Engineering : A Practitioner's Approach. (7th. Edition)*. New York: McGraw-Hill Higher Education.
- [8] Harrel, C., Ghosh, B.K., dan Bowden, R.O., (2004), *Simulation Using Promodel Second Edition*, McGraww-Hill; New York.
- [9] Hoover, S., dan Perry, R., (1990), *Simulation: A Problem Solving Approach*, Addison-Wesley; Reading-MA
- [10] Sulhan, M., (2014)., *Metode Stemming Sebagai Preprocessing Pada Filter Kata Porno Melalui Aspek Pendidikan*. Seminar Nasional Teknologi Informasi dan Komunikasi, UAJY, Yogyakarta
- [11] Mujilahwati, S., (2016)., *Pre-Processing Text Mining Pada Data Twitter*. Seminar Nasional Teknologi Informasi dan Komunikasi, UAJY, Yogyakarta
- [12] Marlinda., L., (2013)., *Pembelajaran Bahasa Indonesia Berbasis Web Menggunakan Metode Maximum Marginal Relevance*. Seminar Nasional Sistem Informasi Indonesia, Jakarta.