

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Perkembangan zaman terus melaju cepat diiringi dengan pertumbuhan teknologi yang tumbuh begitu pesat dalam hal industri, komunikasi maupun informasi. Saat ini, kebutuhan manusia akan informasi sudah semakin meningkat, seiring dengan meningkatnya taraf kecerdasan manusia secara keseluruhan. Semakin cerdas manusia, semakin haus tuntutannya akan informasi yang tersaji secara akurat sesuai dengan kebutuhannya saat itu.

Text mining merupakan bagian ilmu yang mempelajari bagaimana menggali sebuah informasi dari data mentah, terutama data yang berupa data teks. Dalam proses mencari informasi pada data teks dilakukan beberapa tahapan, salah satunya adalah *Preprocessing*. Pada tahap *Preprocessing* dikenal sebuah proses yaitu *stemming*, yaitu proses mengenali kata dasar dengan cara menggabungkan atau memecahkan setiap varian-varian suatu kata[1]. Proses *stemming* ini dianggap mempermudah proses pengolahan data, dan juga mengurangi besar Tabel penyimpanan. Berbagai macam algoritma *stemming* yang sudah ada saat ini, memiliki kesalahan-kesalahan pada pembentukan kata dasar sehingga masih diperlukannya proses pemeriksaan kembali secara manual yang justru meningkatkan waktu kerja yang dibutuhkan dalam tahap *Preprocessing* [2]. Selain itu algoritma *stemming* yang ada saat ini kebanyakan dipakai untuk teks berbahasa

Inggris, dan lebih sulit diterapkan pada Bahasa Indonesia. Hal ini disebabkan karena Bahasa Indonesia memiliki tata Bahasa yang unik atau khas.

Terdapat beberapa macam algoritma *stemming* yang umumnya digunakan untuk Bahasa Indonesia dan Bahasa Inggris. Untuk proses *stemming* Bahasa Indonesia sebelumnya sudah pernah diteliti oleh beberapa peneliti diantaranya Algoritma *Stemming* Nazief Adriani, Algoritma *Stemming* Arifin Setiono, dan Algoritma Vega[3]. Algoritma Nazief Adriani dikembangkan berdasarkan aturan morfologi bahasa Indonesia yang mengelompokkan imbuhan menjadi awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*), dan gabungan awalan akhiran (*confixes*). Algoritma ini menggunakan kamus kata dasar dan mendukung recording, yakni penyusunan kembali kata-kata yang mengalami proses *stemming* berlebih. Kamus kata dasar tersebut dibutuhkan untuk memeriksa apakah kata dasar yang melalui proses *stemming* benar dan ditemukan pada kamus saat proses *stemming* dilakukan.

Beberapa algoritma *stemming* untuk bahasa Inggris, diantaranya adalah algoritma *Porter stemmer*, *Paice Husk stemmer*, *Lovins stemmer*, *Dawson stemmer*, dan *Krovetz stemmer* [4]. Algoritma *Paice Husk* merupakan salah satu algoritma untuk proses *stemming*. Algoritma *Paice Husk* dibuat di Departement Komputer Universitas Lancaster. Algoritma *Paice Husk* disebut kuat karena aturan-aturan *stemming* dapat diaplikasikan berulang kali dalam pemrosesan sebuah kata dan lebih akurat karena menggunakan kamus data untuk justifikasinya[5]. Saat ini algoritma *Paice Husk* masih digunakan untuk proses *stemming* pada Bahasa Inggris, sehingga untuk menghasilkan proses *stemming* yang baik pada Bahasa Indonesia menggunakan algoritma *Paice Husk* dibutuhkan implementasi aturan-aturan

morfologi Bahasa Indonesia pada algoritma. Algoritma *Paice Husk* ini dikenal berat, namun algoritma ini mudah diimplementasikan .

Setelah mengetahui beberapa permasalahan mengetahui kelebihan dan kekurangan algoritma Nazief Adriani dan algoritma *Paice Husk* yang diterapkan pada *stemming* kata berbahasa Indonesia, dan membandingkan hasil kerjanya. Setelah itu agar dapat mengetahui Algoritma manakah diantara Nazief Adrianai dan *Paice Husk* yang memiliki akurasi, kompleksitas, dan efektifitas waktu yang lebih baik.

Berdasarkan beberapa masalah yang telah dipaparkan sebelumnya maka diambil judul Skripsi yaitu ” **Perbandingan Algoritma Nazief Adriani dan *Paice-Husk* untuk Proses *Stemming* Teks Bahasa Indonesia** ”

1.2. Rumusan Masalah

Dari latar belakang yang telah dipaparkan sebelumnya terdapat beberapa permasalahan yang dapat di rumuskan, diantaranya:

1. Bagaimana mengimplementasikan teknik *stemming* dengan menggunakan Algoritma *Nazief Adriani* dan Algoritma *Paice Husk* pada teks Bahasa Indonesia?
2. Algoritma manakah diantara *Paice-Husk* dan Nazief Adriani yang memiliki akurasi, kompleksitas, dan efektifitas waktu yang lebih baik pada teknik *stemming* teks Bahasa Indonesia?

1.3. Tujuan dan Manfaat

1.3.1. Tujuan

Berdasarkan rumusan masalah yang telah dibuat, maka tujuan dari penelitian ini, yaitu :

1. Mengetahui cara mengimplementasikan teknik *stemming* dengan menggunakan Algoritma *Paice Husk* dan Algoritma Nazief Adriani pada teks Bahasa Indonesia.
2. Mengetahui Algoritma manakah diantara *Paice Husk* dan Nazief Adriani yang memiliki akurasi, kompleksitas, dan efektifitas waktu yang lebih baik pada teknik *stemming* teks Bahasa Indonesia.

1.3.2. Manfaat

Manfaat yang akan diperoleh dari penelitian ini yaitu dapat membantu memperluas wawasan mengenai kajian *text mining* terutama dalam mencari informasi pada data teks yang salah satu tahapannya adalah *Preprocessing* dan prosesnya ada yang disebut *stemming*.

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Kamus yang digunakan pada proses *stemming* merupakan Kamus Besar Bahasa Indonesia.
2. Proses *stemming* berdasarkan aturan bentuk kata kerja Bahasa Indonesia yang sesuai EYD (Eja'an Yang Disempurnakan).

3. Tidak dapat melakukan perbaikan pada kesalahan penulisan kata.
4. Bahasa pemrograman yang digunakan adalah PHP.

1.5. Metodologi Penelitian

1.5.1. Teknik Pengumpulan Data

Adapun teknik pengumpulan data yang dilakukan terdiri dari beberapa tahapan yaitu :

1. Observasi

Pada metode ini penulis melakukan pengamatan baik secara langsung maupun melalui media pendukung seperti *internet* untuk mencatat fenomena yang muncul dan mempertimbangkan hubungan antar aspek dalam fenomena tersebut.

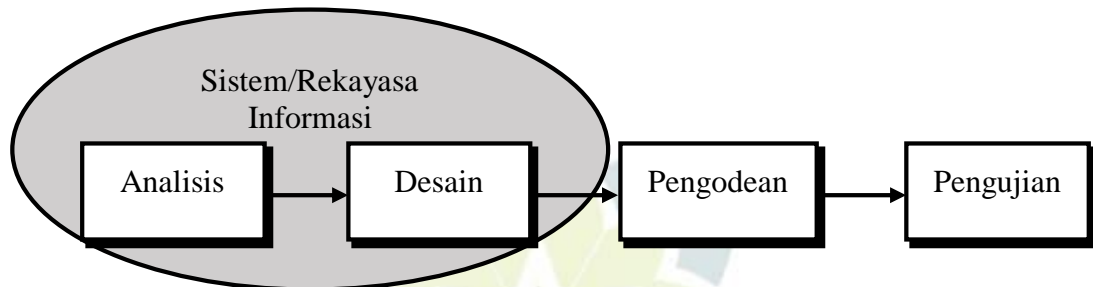
2. Studi Literatur

Melakukan pendalaman materi dari buku-buku yang terkait dengan materi judul dan juga merujuk kepada beberapa jurnal yang terkait dengan materi, lalu melakukan penyaduran atau pengambilan informasi beberapa poin dari sumber-sumber tersebut.

1.5.2. Metode Pengembangan Perangkat Lunak

Model proses pengembangan perangkat lunak yang digunakan dalam pembuatan sistem pada penelitian ini yaitu menggunakan model Air Terjun (*Waterfall*). Model SDLC air terjun (*waterfall*) sering disebut model sekuensial

linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)[6]. Gambar 1.1 model air terjun:



Gambar 1.1 Metode Waterfall

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspezifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

b. Desain

Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain

perangkat lunak yang dihasilkan pada tahap ini juga perlu di dokumentasikan.

c. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.

1.6. Sistematika Penulisan

Penulisan skripsi ini dibagi kedalam lima bab, yang disusun berdasarkan sistematika sebagai berikut :

Bab I : Pendahuluan

Dalam bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, kerangka pemikiran, metodologi penelitian dan sistematika penulisan.

Bab II : Studi Pustaka

Dalam bab ini akan dituliskan mengenai teori-teori yang akan digunakan dan relevan dengan penelitian ini serta penjelasan mengenai penelitian – penelitian sebelumnya terkait dengan materi-materi yang sudah ada.

Bab III : Perancangan Sistem

Dalam bab ini akan dijelaskan mengenai analisis dan perancangan terhadap penelitian yang akan dilakukan. Dimulai dari analisis sistem, analisis kebutuhan fungsional, analisis kebutuhan perangkat lunak maupun perangkat keras, analisis arsitektur sistem, serta perancangan dalam pembuatan sistem tersebut meliputi *UML* serta *user interface*.

Bab IV : Implementasi Sistem

Dalam bab ini akan membahas tentang implementasi klasifikasi emosi data tekstual yang dilakukan, skenario pengujian dan hasil pengujian tentang penelitian yang dilakukan serta perhitungan akurasi dari metode yang digunakan pada penelitian.

Bab V : Penutup

Dalam bab ini akan dibahas mengenai kesimpulan dan saran yang didapatkan dari serangkaian proses penelitian yang telah dilakukan.