



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sejak awal perkembangan *web*, orang-orang mulai bergantung pada *search engine* untuk mencari berbagai macam informasi. Menurut Metaxas (2009), ketika ukuran *web* masih sangat kecil, direktori *web* dibangun dan dikelola menggunakan model sistem pakar manusia untuk menampilkan dan mengkategorikan halaman berdasarkan karakteristiknya. Namun pada pertengahan tahun 1990, sangat jelas sekali bahwa model ini sudah tidak lagi relevan. Maka lahirlah *search engine* yang pertama dan terus berkembang hingga saat ini.

*Search engine* yang ada sekarang ini merupakan hasil studi dari *information retrieval*, yang di dalamnya membahas mengenai bagaimana memproses sekumpulan data agar dapat ditampilkan, disimpan, dan dicari untuk tujuan pengetahuan dan sebagai respon terhadap permintaan pengguna (Alhenshiri, 2007).

Untuk menyimpan setiap informasi yang ada di dalam *web*, *search engine* membutuhkan suatu repositori data, salah satunya berupa *relational database*. Namun tidak semua pengguna memahami konsep *schema* dan *query language* untuk mendapatkan informasi tersebut (Termehchy, 2009). Tentunya hal ini merupakan halangan terbesar bagi pengguna awam. Menurut Hariyono (2005), dengan adanya *search engine* pengguna dapat memasukkan *query* yang bebas

dalam arti kata *query* yang sesuai dengan bahasa manusia dan sistem dapat menemukan dokumen yang sesuai dengan *query* yang ditulis.

Sekarang ini aplikasi *search engine* sudah banyak digunakan dalam berbagai bidang, salah satunya adalah jurnal ilmiah. Menurut Belt (2011), jurnal ilmiah dapat diartikan sebagai media publikasi mengenai suatu hasil penelitian yang ditujukan kepada komunitas ilmiah. Seiring dengan perkembangan teknologi, jurnal ilmiah mulai dipublikasikan menggunakan media *online* atau sering juga disebut dengan jurnal *online*. Banyaknya jumlah jurnal yang muncul tiap tahun, menyulitkan pengguna ketika harus melakukan pencarian di dalam basis data.

Berdasarkan penelitian yang telah dilakukan oleh James (1998), 80% pengguna layanan *search engine* cenderung melihat hanya pada 2 jawaban pertama yang ditampilkan oleh sistem. Oleh karena itu, kita membutuhkan suatu algoritma *ranking* untuk menghasilkan jawaban yang paling relevan (Alhenshiri, 2007). *Answer Graph Generation* adalah suatu algoritma yang digunakan untuk menghasilkan jawaban berupa *tuple tree* dari *relational database* (Liu, 2006). *Tuple tree* ini terbentuk sesuai dengan *query* yang pengguna berikan. Selain itu dengan adanya perhitungan dengan *vector based model*, maka setiap jawaban akan diberikan peringkat sesuai dengan relevansinya antara *query* dengan jawaban.

Pada umumnya informasi mengenai suatu dokumen atau teks akan disimpan oleh *search engine* dalam bentuk indeks. Indeks tersebut berupa kumpulan istilah-istilah yang sebelumnya telah melalui beberapa tahap

pemrosesan seperti tokenisasi, pembuangan *stopwords*, dan *stemming*. Indeks inilah yang nantinya akan menjadi representasi dari suatu dokumen (Budi, 2006).

Dari beberapa tahapan pemrosesan teks diatas, *stemming* menjadi salah satu isu penting dalam *Information Retrieval*. *Stemming* adalah suatu proses dimana kata-kata yang ada di dalam suatu dokumen ditranformasikan menjadi kata-kata dasar (Keke, 2012). Hal ini menyebabkan berkurangnya ukuran indeks sehingga dapat membuat proses *retrieval* suatu dokumen menjadi lebih cepat karena jumlah indeks yang dicari menjadi lebih sedikit (Budi, 2006).

Berdasarkan penelitian yang telah dilakukan oleh Agusta (2009) yaitu membandingkan algoritma *Stemming Porter Indonesia* dengan *Nazief-Adriani*, dapat disimpulkan bahwa algoritma *Nazief-Adriani* memiliki presisi yang lebih tinggi dalam melakukan *Stemming* kata-kata berbahasa indonesia, namun memiliki kelemahan dari sisi waktu pemrosesan. Beberapa hal yang menyebabkan lamanya waktu pemrosesan adalah algoritma *Nazief-Adriani* harus melakukan pencocokan kata dengan data yang ada di dalam basis data dan bahasa indonesia itu sendiri memiliki morfologi kata yang lebih kompleks dibandingkan dengan bahasa lainnya .

Dari permasalahan tersebut semakin terbuka luas peluang bagi peneliti untuk melakukan penelitian lebih lanjut mengenai algoritma *Stemming Nazief-Adriani*. Salah satu solusi yang dapat ditawarkan adalah dengan menggunakan algoritma *Bloom Filter* sebagai pengganti pencocokan kata yang ada di dalam basis data. Algoritma ini merupakan *space-efficient probabilistic data structure* yang memungkinkan untuk melakukan *query* kepada suatu himpunan data

(Tarkoma, 2010). Keunggulan algoritma ini adalah pencocokan kata menjadi lebih cepat karena penggunaan operasi matematik dalam pencarian suatu elemen relatif lebih cepat dibandingkan dengan *query* basis data.

## 1.2 Rumusan Masalah

Dari permasalahan yang telah dijelaskan sebelumnya, maka rumusan masalah penelitian ini dapat dibagi menjadi beberapa bagian sebagai berikut.

1. Apakah algoritma *Answer Graph Generation* dapat menghasilkan jawaban yang relevan kepada pengguna?
2. Apakah algoritma *Stemming Nazief-Adriani* dapat mengoptimalkan fitur pencarian?
3. Apakah algoritma *Bloom Filter* dapat mengoptimalkan performa *Stemming Nazief-Adriani*?

## 1.3 Batasan Masalah

Beberapa batasan yang harus ditentukan dalam penelitian ini yaitu sebagai berikut.

1. Penelitian menggunakan media penyimpanan data dalam bentuk *relational database* (MySQL).
2. Penelitian membahas seputar penerapan konsep relevansi jawaban, optimasi *query* dan optimasi *indexing*.
3. Penelitian tidak membahas penggunaan *boolean operator* dalam fitur pencarian.

4. Penelitian tidak membahas pencarian berdasarkan frase kata.
5. Algoritma *Answer Graph Generation* diimplementasikan pada fitur pencarian *simple search*.
6. Penelitian diimplementasikan dengan menggunakan *framework* Codeigniter.
7. Studi kasus pada jurnal Universitas Multimedia Nusantara.

#### **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah mengimplementasikan algoritma *Answer Graph Generation* dan melakukan optimasi *indexing* untuk pencarian kata kunci pada jurnal *online*.

#### **1.5 Manfaat Penelitian**

Penelitian ini bermanfaat untuk menghasilkan suatu fitur *search engine* yang optimal dari sisi *indexing* dan relevan dari sisi jawaban yang diberikan kepada pengguna.

#### **1.6 Sistematika Penulisan**

Sistematika penulisan laporan skripsi ini dapat dijelaskan sebagai berikut.

1. Bab I berisi pendahuluan yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

2. Bab II berisi tinjauan pustaka yang terdiri dari landasan teori mengenai *Information Retrieval*, *Answer Graph Generation*, *Tuples Tree Ranking*, *Nazief-Adriani Stemmer*, *Bloom Filter*, *Index Construction* dan *CodeIgniter*.
3. Bab III berisi analisis dan perancangan sistem yang terdiri dari spesifikasi umum kebutuhan sistem dan desain sistem.
4. Bab IV berisi implementasi dan uji coba yang terdiri dari penjelasan mengenai implementasi dan hasil uji coba sistem.
5. Bab V berisi simpulan dan saran yang terdiri dari kesimpulan penelitian dan saran untuk penelitian selanjutnya.

UMMN