SKRIPSI

PEMBANGUNAN SISTEM PENCARIAN DAN TEMU KEMBALI INFORMASI DENGAN TERM REWEIGHTING RELEVANCE FEEDBACK

William Richard Suprayogi

NPM: 2015730044

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2021

UNDERGRADUATE THESIS

DEVELOPMENT OF INFORMATION RETRIEVAL SYSTEM WITH TERM REWEIGHTING RELEVANCE FEEDBACK

William Richard Suprayogi

NPM: 2015730044

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2021

LEMBAR PENGESAHAN

PEMBANGUNAN SISTEM PENCARIAN DAN TEMU KEMBALI INFORMASI DENGAN TERM REWEIGHTING RELEVANCE FEEDBACK

William Richard Suprayogi

NPM: 2015730044

Bandung, 01 Februari 2021

Menyetujui,

Pembimbing

Luciana Abednego, M.T.

Ketua Tim Penguji

Anggota Tim Penguji

Dr.rer.nat. Cecilia Esti Nugraheni

Husnul Hakim, M.T.

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMBANGUNAN SISTEM PENCARIAN DAN TEMU KEMBALI INFORMASI DENGAN TERM REWEIGHTING RELEVANCE FEEDBACK

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 01 Februari 2021

William Richard Suprayogi
NPM: 2015730044

ABSTRAK

Pencarian dan temu kembali informasi merupakan aktivitas untuk mencari suatu informasi/dokumen tertentu dalam suatu koleksi dokumen/koleksi informasi yang besar. Informasi yang biasa orang-orang cari dapat berupa data seperti teks,gambar,video, dan audio. Sistem pencarian akan menerima suatu kata kunci dari pengguna yang disebut *query*. Sistem nantinya akan mengambil semua dokumen/informasi yang berkaitan dengan *query* dengan cara membaca dokumen/informasi yang telah terindeks. Indeks merupakan daftar setiap kata yang muncul di koleksi dokumen beserta daftar dokumen yang mengandung salah satu dari setiap kata di indeks. Salah satu proses pada pengindeksan yaitu melakukan *term weighting.Term weighting* dilakukan untuk menilai suatu kata bernilai pada suatu kokeksi atau dokumen atau tidak. Semakin tinggi nilai dari bobot suatu kata berarti semakin sering digunakan pada koleksi dokumen.

Sistem pencarian banyak sekali dipakai orang untuk mencari informasi. Namun terdapat kemungkinan bahwa hasil yang diterima oleh pengguna dari sistem pencarian kurang memuaskan. Hasil pencarian tidak memuaskan apabila dokumen/informasi terkait dengan kata kunci tidak pada urutan atas. Terdapat solusi untuk mengatasi permasalahan tersebut yaitu dengan menggunakan fitur *relevance feedback*. *Relevance feedback* adalah fitur pada sistem pencarian yang nantinya akan mengembalikan hasil yang lebih relevan dari sebelumnya. Umpan balik yang diterima oleh sistem adalah dokumen-dokumen yang dianggap relevan oleh sistem atau pengguna. Ada dua cara *relevance feedback* yang bisa diimplementasikan pada sistem pencarian yaitu rochio *feedback* dan *probabilistic feedback*. *Rochio feedback* merupakan algoritma yang ditujukan untuk memaksimalkan nilai kemiripan antara *query* masukkan dengan dokumen yang relevan. *Probabilistic feedback* merupakan algoritma pengklasifikasian statistik dimana dapat memprediksi probabilitas suatu *query* dengan koleksi dokumen.

Terdapat dua cara pengukuran performa sistem pencarian yaitu *precision* dan *recall*. *Precision* merupakan hasil pembagian dari jumlah dokumen relevan yang ditemukan dengan jumlah dokumen yang ditemukan yang menggambarkan kemampuan sistem untuk tidak memanggil dokumen yang tidak relevan. Sedangkan recall merupakan hasil pembagian dari jumlah dokumen relevan yang ditemukan dengan jumlah dokumen relevan yang menggambarkan kemampuan sistem untuk mengambil dokumen yang relevan.

Telah dibuat suatu sistem pencarian dengan memanfaatkan kedua algoritma *relevance feedback*. Untuk membuat sistem pencarian diperlukan fitur untuk membaca dokumen, mengindeks dokumen-dokumen yang telah terbaca, dan melakukan pencarian pada indeks. Selain membuat sistem pencarian, nantinya akan dibuat sistem untuk menguji performa dari sistem pencarian tersebut. Untuk membangun kedua sistem tersebut, akan dimanfaatkan *library* perangkat lunak mesin pencarian *open-source* bernama Lucene. Lucene akan menyediakan alat-alat yang dapat membantu dalam pembangunan sistem mesin pencarian dan sistem pengujian performa mesin pencarian.

Pada skripsi ini telah dilakukan pengujian untuk sistem pencarian dengan memanfaatkan kedua algoritma *relevance feedback*. Berdasarkan pengujian yang dilakukan, ditemukan bahwa cara *relevance feedback* yang paling terbaik ada pada rochio *relevance feedback* jika sistem diukur dengan *precision* dan *recall*. Selain itu, ditemukan bahwa dalam 1 detik pengindeksan dapat mengindeks lebih dari 1000 dokumen.

Kata-kata kunci: *Query*, sistem pencarian, *relevance feedback*, indeks, *precision*, *recall*, Lucene

ABSTRACT

Retrieval of information is an activity to find certain information/documents in a large document collection/information collection. The information that people usually look for can be in the form of data such as text, images, videos and audio. The search system will receive a keyword from the user called query. The system will be retrieve all documents / information related to the query by reading the indexed document/information. Index is a list of every word that appears in the document collection along with a list of documents containing one of each word in the index.

Search systems are widely used by people to find information. However, it is possible that the results received by users from the search system are not satisfactory. The search results are not satisfactory if the documents/information related to the user are not in the top order. There is a solution to overcome these problems, namely by using the relevance feedback feature. Relevance feedback is a feature on search engines that will return more relevant results than before. The feedback received by the system are documents that are deemed relevant by the system or users. There are two ways relevance feedback that can be implemented in search system, namely rochio feedback and probabilistic feedback.

There are two ways to measure the performance of the search system, namely precision and recall. Precision is a measurement related to the accuracy of the search system in providing relevant documents. Recall relates to the search system's ability to return all documents relevant to the query.

A search system will be created using both relevance feedback algorithms. To create a search system, features are needed to read documents, index documents that have been read, and perform searches on the index. In addition to creating a search system, a system will be created to test the performance of the search system. To build the two systems, we will use an open-source search engine software library called Lucene. Lucene will provide tools that can assist in building search engine systems and search engine performance testing systems.

This thesis has been tested for the search system using both relevance feedback algorithms. Based on the tests carried out, it was found that the best way of relevance feedback is rochio relevance feedback if the system is measured with precision and recall. In addition, it was found that in 1 second indexing can index more than 1000 documents.

A search system has been developed using both relevance feedback algorithms. To create a search system, features are needed to read documents, index documents that have been read, and perform searches on the index. In addition to creating a search system, a system will be created to test the performance of the search system. To build the two systems, the library search engine software open-source software named Lucene will be utilized. Lucene will provide tools that can assist in building search engine systems and search engine performance testing systems.

Keywords: Query, search system, relevance feedback, index, *precision*, *recall*, Lucene

Untuk diri sendiri, keluarga, teman-teman seperjuangan, dan semua yang telah mendukung

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena dengan rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul "Pembangunan Sistem Pencarian dan Temu Kembali Informasi dengan *Term Reweighting Relevance Feedback*". Penulis berharap skripsi ini dapat berguna bagi orang yang membutuhkan dan juga dapat membantu bagi orang yang akan melanjutkan penelitian ini untuk selanjutnya. Pada kesempatan kali ini, penulis mengucapkan terima kasih kepada:

- Ibu Luciana Abednego, M.T., yang telah membantu membimbing penulis dalam proses pembuatan skripsi beserta perangkat lunak.
- Ibu saya yang telah menyemangati saya dalam proses pembuatan skripsi.
- Teman-teman discord yang selalu mendukung dan memberi masukkan selama proses pembuatan skripsi.
- Novia, Gisel, Andhela, dan teman-temand discord yang telah mendengarkan keluh kesah saya selama pembuatan skripsi.
- Pihak-pihak lain yang tidak bisa disebutkan satu per satu.

Bandung, Februari 2021

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE PROGRAM

BAB 1

PENDAHULUAN

Bab ini menguraikan latar belakang skripsi, rumusan masalah, tujuan yang berdasarkan rumusan masalah, batasan masalah skripsi, metodologi pengerjaan skripsi, dan sistematika pembahasan pada dokumen skripsi.

1.1 Latar Belakang

Pencarian dan temu kembali informasi merupakan suatu aktivitas untuk mencari suatu informasi/dokumen yang berada dalam suatu koleksi teks (corpus)/koleksi informasi dalam ukuran yang besar. Informasi yang bisa didapatkan biasanya bisa berupa data seperti teks, gambar, video, dan audio. Tujuan dari pencarian dan temu kembali informasi ini adalah untuk mengembalikan suatu informasi/dokumen yang relevan terhadap kebutuhan pengguna.

Saat ini, mesin pencarian banyak sekali dipakai untuk mencari informasi. Mesin pencarian informasi ini akan relevan sesuai dengan apa yang dicari pengguna. Misalnya saja ketika pengguna memasukkan kata *apple* pada mesin pencarian maka akan dikembalikan dokumen atau informasi yang relevan dengan kata *apple*. Namun pada kenyataannya, masih terdapat sistem pencarian yang tidak menghasilkan suatu dokumen yang relevan sesuai dengan masukan *query* oleh pengguna. Misal ketika pengguna memasukkan kata *apple* pada mesin pencarian maka sistem tersebut akan mengembalikan dokumen/informasi yang tidak terlalu relevan dengan query *apple*. Pengguna baru akan menemukan dokumen/informasi yang relevan pada baris-baris pada halaman tertentu.

Untuk menemukan suatu kata kunci terdapat dalam dokumen atau tidak, bisa dilakukan dengan membaca seluruh isi dokumen. Nantinya mesin pencarian akan mengembalikan dokumen apabila ada kata yang mirip dengan kata kunci yang dimasukkan. Cara tersebut lebih cocok jika diterapkan pada koleksi dokumen yang lebih kecil dibandingkan dengan koleksi dokumen yang besar. Pada koleksi dokumen yang besar akan memakan waktu yang lama untuk mencari dokumen yang relevan sesuai dengan masukan kata kunci.

Selain itu, cara lain untuk mencari kata kunci yang terdapat dalam koleksi dokumen yaitu dengan membaca indeks kata-kata yang muncul yang terdapat dalam dokumen. Cara ini paling efisien dibandingkan dengan cara pertama karena mesin tidak perlu membaca koleksi dokumen secara keseluruhan. Namun, indeks harus dibangun terlebih dahulu sebelum dibaca oleh mesin. Berikut adalah cara membuat indeks kata-kata yang muncul yang terdapat dalam dokumen:

1. Kumpulkan semua dokumen yang akan diindeks
2. Tokenisasi teks yang terdapat dalam dokumen. Tokenisasi merupakan pemecahan kata pada kalimat. Misal kalimat 'Ibu pergi ke pasar', maka *token-token* dari kalimat tersebut adalah 'ibu', 'pergi', dan 'pasar'.
3. Normalisasi kata-kata yang sudah ditokenisasi agar tidak ada dua kata atau lebih yang memiliki kata dasar yang sama.
4. Buat indeks berdasarkan token yang sudah dinormalisasi.

Kata-kata yang telah terindeks mempunyai bobot sendiri. Bobot kata dapat diartikan sebagai nilai kata tersebut apakah berharga pada suatu koleksi dokumen atau tidak. Semakin besar bobot dari suatu kata maka dapat diartikan kata tersebut sering di pakai pada suatu koleksi dokumen. Sedangkan jika bobot suatu kata kecil maka kata tersebut tidak sering di pakai pada suatu koleksi dokumen tersebut. Misal kata "sejarah" pada koleksi dokumen mempunyai nilai 30 sedangkan "Indonesia" memiliki nilai 20. Bisa diartikan kata "sejarah" lebih bernilai pada suatu koleksi dokumen dibandingkan dengan kata "Indonesia".

Ada berbagai cara agar hasil pencarian yang dilakukan relevan dengan apa yang dicari pengguna. Salah satunya yaitu dengan menggunakan teknik pengukuran relevansi dokumen dengan metode *relevance feedback*. Metode ini akan menunggu *feedback* dari pengguna berupa dokumen yang sudah ditandai relevan oleh pengguna. Hasil dari feedback tersebut akan dikirim ke mesin pencarian untuk melakukan pencarian ulang. Hasil pencarian tersebut akan mirip dengan pencarian awal pengguna namun dengan lebih banyak dokumen yang relevan. Selain cara penandaan dokumen, ada cara lain untuk mendapatkan dokumen yang lebih relevan dengan menghitung probabilistas *query* terhadap dokumen. Sistem nantinya akan mengembalikan dokumen berdasarkan nilai probabilitas tertinggi sampai dengan probabilitas terendah. Pembobotan ulang perlu dilakukan pada metode *relevance feedback* dengan alasan untuk menghasilkan dokumen yang benar benar relevan dengan pengguna. Pembobotan ulang dalam metode *relevance feedback* bisa disebut sebagai *term reweighting*.

Untuk melakukan *relevance feedback*, terdapat dua algoritma yang bisa diimplementasikan yaitu *Rochio algorithm* dan *Probabilistic algorithm*. *Rocchio algorithm* merupakan algoritma yang memanfaatkan *vector space model* dimana algoritma tersebut akan mencari suatu *query* vektor memaksimalkan kemiripan dengan dokumen yang relevan dan meminimalkan kemiripan dengan dokumen yang tidak relevan. Sedangkan *probabilistic algorithm* merupakan algoritma yang memanfaatkan model *Naive Bayes* dimana nantinya akan dibuat suatu *classifier* untuk membedakan dokumen yang relevan dan non-relevan.

Untuk menilai kinerja dari sistem pencari dan temu kembali informasi, ada dua nilai yang bisa dipakai yaitu *precision* dan *recall*. *Precision* merupakan hasil pembagian dari jumlah dokumen relevan yang ditemukan dengan jumlah dokumen yang ditemukan yang berkaitan dengan penilaian kemampuan sistem untuk tidak mengambil dokumen yang tidak relevan. Sedangkan *recall* merupakan hasil pembagian dari jumlah dokumen relevan yang ditemukan dengan jumlah dokumen relevan yang berkaitan dengan penilaian kemampuan sistem dalam mengambil dokumen yang relevan.

Pada skripsi ini, dibuat sebuah sistem pencarian yang memanfaatkan dua algoritma *relevance feedback* secara *offline*. Untuk membuat sistem ini diperlukan sistem pembacaan dokumen yang nantinya dokumen-dokumen tersebut akan melalui tahap pengindeksan. Pada Dokumen yang sudah diindeks nantinya akan bisa dilakukan proses pencarian sesuai dengan masukkan yang diberikan oleh pengguna. Perangkat lunak yang dibuat pada skripsi ini akan dibangun dengan menggunakan bahasa Java yang akan dibantu dengan *library Lucene*. Lucene merupakan *library* berbasis bahasa Java dimana *libary* ini akan membantu dalam proses *indexing* dan juga *searching* pada sistem yang dibuat.

1.2 Rumusan Masalah

Rumusan masalah untuk skripsi ini adalah:

1. Bagaimana cara membuat sistem pencarian dan temu kembali untuk mengembalikan suatu teks secara *offline*?

2. Bagaimana cara mengimplementasikan dua algoritma *Term Reweighting Relevance Feedback* pada sistem pencarian?

3. Bagaimana cara membandingkan algoritma rochio dan algoritma probabilistic tersebut?

1.3 Tujuan

Penelitian ini dilakukan dengan tujuan sebagai berikut:

1. Membuat sistem pencarian dan temu kembali untuk mengembalikan suatu dokumen secara *offline*.
2. Mengimplementasikan dua algoritma *term reweighting relevance feedback* pada sistem pencarian.
3. Membandingkan kedua algoritma *relevance feedback* dengan mengukur kinerja pada sistem pencarian.

1.4 Batasan Masalah

Batasan masalah untuk skripsi ini adalah:

1. Mesin pencarian hanya melakukan pencarian halaman teks secara *offline*.
2. Teks yang dicari dalam bahasa Inggris. Alasannya karena *library* Lucene hanya dapat menangani bahasa Inggris dan isi dokumen keseluruhan memakai bahasa Inggris.

1.5 Metodologi

Metodologi untuk pengerjaan skripsi adalah:

1. Melakukan studi literatur tentang teori-teori yang digunakan untuk membangun sistem pencarian dan temu kembali informasi berbasis offline, pembangunan mesin pencarian menggunakan Lucene, teknik *term reweighting relevance feedback*, dan pembangunan mesin pencari secara *offline*.
2. Merancang dan membuat program untuk melakukan *indexing* menggunakan Lucene.
3. Membuat mesin pencari yang sangat sederhana menggunakan Lucene untuk memahami cara membuat mesin pencari menggunakan Lucene.
4. Merancang dan menganalisis algoritma *relevance feedback* untuk dimplementasi pada mesin pencari.
5. Membuat mesin pencari berdasarkan rancangan yang telah dibuat.
6. Melakukan pengujian dan evaluasi terhadap mesin pencari.
7. Menyusun dokumen skripsi.

1.6 Sistematika Pembahasan

- Bab 1 Pendahuluan

  Bab 1 menjelaskan latar belakang skripsi, rumusan masalah, batasan masalah, metodologi, dan sistematika pembahasan. Pada bagian latar belakang dijelaskan dasar tentang *information retreival*, *relevance feedback*, perangkat lunak yang akan dibuat, serta *tools* yang digunakan untuk membuat aplikasi.

- Bab 2 Landasan Teori

  Pada bab 2 akan dijelaskan landasan teori yang akan dipakai pada skripsi ini seperti *information retreival*, teknik *relevance feeback*, teknik evaluasi penilaian sistem, dan *Lucene*.

- Bab 3 Analisis

  Pada bab 3 akan dijelaskan analisis algoritma, analisis perangkat lunak, dan analisis diagram kelas sederhana.

- Bab 4 Perancangan

  Pada bab 4 berisi perancangan antarmuka, diagram aktivitas, diagram kelas, dan *pseudocode* untuk perancangan perangkat lunak.

- Bab 5 Implementasi dan Pengujian

  Pada bab 5 berisi implementasi kode program dan pengujian.

- Bab 6 Kesimpulan dan Saran

  Pada bab 6 berisi kesimpulan setelah melakukan pengujian dan saran-saran untuk mengembangkan skripsi ini.