

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 koleksi 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 masukan selama proses pembuatan skripsi.
- Novia, Gisel, Anhela, 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

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xxi
DAFTAR TABEL	xxv
DAFTAR KODE PROGRAM	xxvii
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.6 Sistematika Pembahasan	3
2 LANDASAN TEORI	5
2.1 Pencarian dan Temu Kembali Informasi	5
2.2 Indexing	6
2.2.1 Proses <i>Textual Processing</i>	6
2.2.2 Struktur Data <i>Indexing</i>	8
2.2.3 <i>Inverted Index</i>	8
2.3 Searching	8
2.3.1 <i>Boolean Model</i>	9
2.3.2 Vector space model	9
2.3.3 Term Weighting	9
2.3.4 Tf-idf	10
2.3.5 Pengukuran similaritas pada <i>Vector Space Model</i>	11
2.4 <i>Relevance Feedback</i>	11
2.4.1 Pengertian <i>Relevance Feedback</i>	11
2.4.2 Algoritma <i>Rocchio</i>	12
2.4.3 Algoritma <i>Probabilistic</i>	14
2.5 Penilaian Peforma Pencarian	14
2.6 <i>Lucene</i>	15
2.6.1 <i>Indexing</i> pada <i>Lucene</i>	15
2.6.2 <i>Searching</i> pada <i>Lucene</i>	16
3 ANALISIS	19
3.1 Analisis <i>Inverted Index</i>	19
3.2 Analisis penggunaan tf-idf	20
3.3 Analisis Penerapan Similaritas <i>Query</i> terhadap dokumen	21

3.4	Analisis penerapan <i>Rocchio Relevance Feedback</i>	24
3.5	Analisis penerapan <i>Probabilistic Relevance Feedback</i>	25
3.6	Analisis penerapan penilaian evaluasi sistem	29
3.7	Analisis Lucene	30
3.8	Analisis <i>Use Case</i>	31
3.9	Analisis Masukan dan Keluaran Program	34
3.9.1	Perangkat Lunak Pengindeksan	34
3.9.2	Perangkat Lunak Mesin Pencarian	35
3.9.3	Perangkat Lunak Uji	35
3.10	Diagram Aktivitas	38
3.11	Diagram Kelas Sederhana	41
4	PERANCANGAN	45
4.1	Rancangan Antarmuka	45
4.1.1	Antarmuka Sistem <i>Indexing</i>	45
4.1.2	Antarmuka Sistem Pencarian <i>Rocchio Feedback</i>	46
4.1.3	Antarmuka Sistem Pencarian <i>Probabilistic Feedback</i>	47
4.1.4	Antarmuka Sistem Pengujian <i>Rocchio Feedback</i>	47
4.1.5	Antarmuka Sistem Pengujian <i>Probabilistic Feedback</i>	48
4.1.6	Antarmuka Grafik <i>Precision</i> dan <i>Recall</i>	50
4.2	Diagram Kelas Rinci	50
4.2.1	Diagram Kelas <i>Search Engine</i>	50
4.2.2	Diagram kelas Testing	61
4.3	<i>Pseudocode</i>	68
4.3.1	<i>Pseudocode</i> untuk Algoritma Pengindeksan	69
4.3.2	<i>Pseudocode</i> untuk Algoritma Pencarian	69
4.3.3	<i>Pseudocode</i> untuk Algoritma <i>Relevance Feedback</i>	70
5	IMPLEMENTASI DAN PENGUJIAN	75
5.1	Implementasi	75
5.1.1	Lingkungan Implementasi	75
5.1.2	Implementasi Perangkat Lunak	75
5.1.3	Potongan Kode Program	86
5.2	Pengujian	93
5.2.1	Pengujian Fungsional	93
5.2.2	Pengujian Eksperimental	109
6	KESIMPULAN DAN SARAN	165
6.1	Kesimpulan	165
6.2	Saran	165
	DAFTAR REFERENSI	167
A	KODE PROGRAM FITUR <i>Indexing</i>	169
A.1	Package <i>cranfield_collection</i>	169
A.2	Package <i>Indexer</i>	170
B	KODE PROGRAM FITUR <i>Searching</i> DAN <i>Feedback</i>	173
B.1	Package <i>searcher</i>	173
B.2	Package <i>Relevance Feedback</i>	176
B.3	Package <i>Weighting Term</i>	179
C	KODE PROGRAM FITUR PENGUJIAN	183

C.1	Package dataQuery	183
C.2	Package query_reader	186
C.3	Package TestingPerformanceRochio	187
C.4	Package TestingPerformanceProbabilistic	192
D	KODE PROGRAM UI	199
E	HASIL EKSPERIMEN	211
E.1	Pengujian Rochio <i>Feedback</i>	211
E.2	Pengujian <i>Probabilistic Feedback</i>	214

DAFTAR GAMBAR

2.1	Diagram PTKI	6
2.2	Diagram <i>Text Processing</i>	8
2.3	Vector Space Model	9
2.4	Nilai optimal <i>rochio query</i> untuk memisahkan dokumen yang relevan dan tidak relevan	13
2.5	Hasil algoritma <i>rochio</i> yang sudah dimodifikasi	13
2.6	<i>Index Processing</i> pada <i>Lucene</i>	16
2.7	<i>Searching Proses</i> pada <i>Lucene</i>	17
3.1	Diagram <i>Use Case Diagram</i> Perangkat Lunak	32
3.2	Contoh Struktur Dokumen Cranfield	36
3.3	Contoh Struktur Query Cranfield	37
3.4	Contoh Isi <i>File qrel</i>	37
3.5	Diagram Aktivitas untuk Proses Pengindeksan Dokumen	38
3.6	Diagram Aktivitas untuk Proses Pencarian dengan <i>Rochio Feedback</i>	39
3.7	Diagram Aktivitas untuk Proses Pencarian dengan <i>Probabilistic Feedback</i>	40
3.8	Diagram Aktivitas untuk Proses Pengujian	41
3.9	Kelas Diagram <i>Search Engine</i>	42
3.10	Kelas Diagram <i>Testing</i>	43
4.1	Rancangan Antarmuka Sistem <i>Indexing</i>	45
4.2	Rancangan Antarmuka Sistem Pencarian dengan <i>rochio Feedback</i>	46
4.3	Rancangan Antarmuka Sistem Pencarian dengan <i>Probabilistic Feedback</i>	47
4.4	Rancangan Antarmuka Pengujian Rochio	48
4.5	Rancangan Antarmuka Pengujian Probabilistic	49
4.6	Rancangan Antarmuka Grafik Precision dan Recall	50
4.7	Kelas Diagram Indexer	51
4.8	Kelas Cranfield Document	51
4.9	Kelas Cranfield <i>Text Collection</i>	51
4.10	Kelas <i>Handler Document</i>	52
4.11	Kelas Cranfield Indexer	52
4.12	Kelas <i>Read Document Index</i>	52
4.13	Kelas Diagram Searching dan Feedback	54
4.14	Kelas Searcher Rochio	54
4.15	Kelas Searcher Probabilistic	54
4.16	Kelas Search Cranfield Controller	55
4.17	Kelas Search Probabilistic Controller	55
4.18	Kelas <i>Rochio Feedback</i>	57
4.19	Kelas <i>Probabilistic Feedback</i>	57
4.20	Kelas <i>TfidfWeight</i>	59
4.21	Kelas <i>DocumentResultRochio</i>	60
4.22	Kelas <i>DocumentResultProbabilistic</i>	61
4.23	Kelas <i>PathDocument</i>	61
4.24	Kelas Diagram Testing	61

4.25	Kelas Query Reader Cranfield	62
4.26	Kelas Query Reader Relevance Cranfield	62
4.27	Kelas Searcher Rochio Tester	62
4.28	Kelas Rochio Feedback Tester	63
4.29	Kelas TfIdf Tester	63
4.30	Kelas Search Rochio Tester	63
4.31	Kelas Rochio Value Tester	64
4.32	Kelas Searcher Probabilistic Tester	66
4.33	Kelas Probabilistic Feedback Tester	66
4.34	Kelas Search Probabilistic Test	67
4.35	Kelas Probabilistic Value Tester	67
5.1	Antarmuka <i>Indexing</i> Ketika Perangkat Lunak Baru Dibuka	76
5.2	Antarmuka <i>Indexing</i> Ketika Sedang Dijalankan	77
5.3	Antarmuka <i>Indexing</i> Ketika Selesai Dijalankan	77
5.4	Antarmuka Mesin Pencarian dengan Rochio <i>Feedback</i>	78
5.5	Antarmuka Mesin Pencarian Rochio Ketika Melakukan Pencarian Awal	79
5.6	Antarmuka Mesin Pencarian Rochio Ketika Dokumen Ditandai	79
5.7	Antarmuka Mesin Pencarian Rochio Setelah <i>Feedback</i>	80
5.8	Antarmuka Mesin Pencarian <i>Probabilistic Feedback</i>	80
5.9	Antarmuka Mesin Pencarian <i>Probabilistic Feedback</i> Ketika Melakukan Pencarian Awal	81
5.10	Antarmuka Mesin Pencarian Probabilistic Setelah Feedback	82
5.11	Antarmuka Pengujian Rochio	82
5.12	Antarmuka Hasil Pengujian Rochio	83
5.13	Antarmuka Grafik <i>Precision</i> Pengujian Rochio <i>feedback</i>	84
5.14	Antarmuka Grafik <i>Recall</i> Pengujian Rochio <i>feedback</i>	84
5.15	Antarmuka Pengujian <i>Probabilistic Feedback</i>	85
5.16	Antarmuka Hasil Pengujian <i>Probabilistic Feedback</i>	85
5.17	Antarmuka Hasil Grafik <i>Precision Probabilistic Feedback</i>	86
5.18	Antarmuka Hasil Grafik <i>Recall Probabilistic Feedback</i>	86
5.19	Pengujian Fitur <i>Indexing</i> Nomor 1	94
5.20	Pengujian Fitur <i>Indexing</i> Nomor 2	94
5.21	Pengujian Fitur <i>Indexing</i> Nomor 3	95
5.22	Pengujian Fitur <i>Indexing</i> Nomor 4	95
5.23	Pengujian Fitur <i>Searching</i> Rochio Nomor 1	96
5.24	Pengujian Fitur <i>Searching</i> Rochio Nomor 2	97
5.25	Pengujian Fitur <i>Searching</i> Rochio Nomor 3	97
5.26	Pengujian Fitur <i>Searching</i> Rochio Nomor 4	98
5.27	Pengujian Fitur <i>Searching</i> Rochio Nomor 5	98
5.28	Pengujian Fitur <i>Searching</i> Rochio Nomor 6	99
5.29	Pengujian Fitur <i>Searching</i> Rochio Nomor 7	99
5.30	Pengujian Fitur <i>Searching Probabilistic</i> Nomor 1	100
5.31	Pengujian Fitur <i>Searching Probabilistic</i> Nomor 2	101
5.32	Pengujian Fitur <i>Searching Probabilistic</i> Nomor 3	101
5.33	Pengujian Fitur <i>Searching Probabilistic</i> Nomor 4	102
5.34	Pengujian Fitur <i>Searching Probabilistic</i> Nomor 5	102
5.35	Pengujian Fungsional <i>Testing</i> Rochio Nomor 1	103
5.36	Pengujian Fungsional <i>Testing</i> Rochio Nomor 2	104
5.37	Pengujian Fungsional <i>Testing</i> Rochio Nomor 3	104
5.38	Pengujian Fungsional <i>Testing</i> Rochio Nomor 4	105
5.39	Pengujian Fungsional <i>Testing</i> Rochio Nomor 5	105
5.40	Pengujian Fungsional <i>Testing</i> Rochio Nomor 6	106

5.41	Pengujian Fungsional <i>Testing Probabilistic</i> Nomor 1	107
5.42	Pengujian Fungsional <i>Testing Probabilistic</i> Nomor 2	107
5.43	Pengujian Fungsional <i>Testing Probabilistic</i> Nomor 3	108
5.44	Pengujian Fungsional <i>Testing Probabilistic</i> Nomor 4	108
5.45	Pengujian Fungsional <i>Testing Probabilistic</i> Nomor 5	109
5.46	Pengujian Fungsional <i>Testing Probabilistic</i> Nomor 6	109
5.47	Grafik <i>Precision</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 1-50 . .	113
5.48	Grafik <i>Recall</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 1-50 . . .	113
5.49	Grafik <i>Precision</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 51-100	115
5.50	Grafik <i>Recall</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 51-100 . .	116
5.51	Grafik <i>Precision</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 101-150	118
5.52	Grafik <i>Recall</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 101-150 . .	118
5.53	Grafik <i>Precision</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 151-225	121
5.54	Grafik <i>Recall</i> dengan <i>Term Limit</i> 25 dan <i>Factor Tf-Idf</i> 15 untuk <i>Query</i> 151-225 . .	121
5.55	Grafik <i>Precision</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 1-50 . .	123
5.56	Grafik <i>Recall</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 1-50 . . .	124
5.57	Grafik <i>Precision</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 51-100	126
5.58	Grafik <i>Recall</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 51-100 . .	126
5.59	Grafik <i>Precision</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 101-150	128
5.60	Grafik <i>Recall</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 101-150 . .	129
5.61	Grafik <i>Precision</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 151-225	131
5.62	Grafik <i>Recall</i> dengan <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 151-225 . .	132
5.63	Grafik <i>Precision</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 1-50 . .	134
5.64	Grafik <i>Recall</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 1-50 . . .	134
5.65	Grafik <i>Precision</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 51-100	136
5.66	Grafik <i>Recall</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 51-100 . .	137
5.67	Grafik <i>Precision</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 101-150	139
5.68	Grafik <i>Recall</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 101-150 . .	139
5.69	Grafik <i>Precision</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 151-225	142
5.70	Grafik <i>Recall</i> dengan <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 151-225 . .	142
5.71	Grafik <i>Precision</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 1-50	145
5.72	Grafik <i>Recall</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 1-50	145
5.73	Grafik <i>Precision</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 51-100	147
5.74	Grafik <i>Recall</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 51-100	148
5.75	Grafik <i>Precision</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 101-150	150
5.76	Grafik <i>Recall</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 101-150	150
5.77	Grafik <i>Precision</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 151-225	153
5.78	Grafik <i>Recall</i> dengan <i>MU</i> 2000.0 dan <i>Factor</i> 25 <i>Query</i> 151-225	153
5.79	Grafik <i>Precision</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 1-50	155
5.80	Grafik <i>Recall</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 1-50	156
5.81	Grafik <i>Precision</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 51-100	158
5.82	Grafik <i>Recall</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 51-100	158
5.83	Grafik <i>Precision</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 101-150	160
5.84	Grafik <i>Recall</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 101-150	161
5.85	Grafik <i>Precision</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 151-225	163
5.86	Grafik <i>Recall</i> dengan <i>MU</i> 3000.0 dan <i>Factor</i> 50 <i>Query</i> 151-225	164
E.1	Hasil Pengujian Precision Rochio dengan <i>Boost Query</i> 0.3 <i>Term Limit</i> 25 dan <i>Factor</i> Tf-Idf 15	211
E.2	Hasil Pengujian Recall Rochio dengan <i>Boost Query</i> 0.3 <i>Term Limit</i> 25 dan <i>Factor</i> Tf-Idf 15	211

E.3	Hasil Pengujian Precision Rochio dengan <i>Boost Query 0.5 Term Limit 50</i> dan <i>Factor Tf-Idf 30</i>	212
E.4	Hasil Pengujian Recall Rochio dengan <i>Boost Query 0.5 Term Limit 50</i> dan <i>Factor Tf-Idf 30</i>	212
E.5	Hasil Pengujian Rochio dengan <i>Boost Query 2 Term Limit 75</i> dan <i>Factor Tf-Idf 50</i>	213
E.6	Hasil Pengujian Recall Rochio dengan <i>Boost Query 2 Term Limit 75</i> dan <i>Factor Tf-Idf 50</i>	213
E.7	Hasil Pengujian Precision Probabilitisic dengan MU 2000 dan <i>Factor 25</i>	214
E.8	Hasil Pengujian Recall Probabilitisic dengan MU 2000 dan <i>Factor 25</i>	214
E.9	Hasil Pengujian Precision Probabilitisic dengan MU 3000 dan <i>Factor 50</i>	215
E.10	Hasil Pengujian Recall Probabilitisic dengan MU 3000 dan <i>Factor 50</i>	215
E.11	Hasil Pengujian Recall Rochio dengan <i>Boost Query 2 Term Limit 75</i> dan <i>Factor Tf-Idf 50</i>	215

DAFTAR TABEL

2.1	Matrix Lancaster	15
3.1	Tabel <i>Term</i>	19
3.2	Tabel Frekuensi untuk Satu Dokumen	20
3.3	Tabel Frekuensi Dokumen untuk Setiap Kata	20
3.4	Tabel Hasil Perhitungan idf Setiap Kata	20
3.5	Tabel Nilai Tf-Idf untuk Setiap Kata	21
3.6	Tabel nilai TF keseluruhan kata	21
3.7	Tabel <i>document frequency</i> dan idf setiap kata	22
3.8	Tabel tf-idf keseluruhan kata	23
3.9	Tabel nilai <i>similarity</i> dengan <i>Query</i>	24
3.10	Tabel koleksi dokumen yang dikembalikan oleh sistem	24
3.11	Tabel Frekuensi Kata pada Setiap Dokumen	26
3.12	Tabel Ukuran Setiap Dokumen	26
3.13	Tabel <i>Score</i> Probabilitas <i>Term</i>	27
3.14	Tabel Score C1 dan C2	28
3.15	Tabel Score P_{Doc} dan P_{ref}	28
3.16	Tabel Probabilitas Term terhadap dokumen1	28
3.17	Tabel Score Probabilitas Query Terhadap Dokumen	29
3.18	Tabel sampel data dokumen	29
5.1	Tabel Pengujian Fungsional Fitur <i>Indexing</i>	93
5.2	Tabel Pengujian Fungsional Fitur <i>Searching</i> Rochio	96
5.3	Tabel Pengujian Fungsional Fitur <i>Searching</i> Probabilistic	100
5.4	Tabel Pengujian Fungsional Fitur Pengujian Rochio <i>feedback</i>	103
5.5	Tabel Pengujian Fungsional Fitur Pengujian Probabilistic <i>feedback</i>	106
5.6	Tabel Pengujian Kecepatan <i>Indexing</i> untuk Dokumen Cranfield dengan Satuan Dokumen per Milidetik	110
5.7	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.3 Term Limit 25 dan Factor Tf-Idf 15 untuk Query 1-50	111
5.8	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.3 Term Limit 25 dan Factor Tf-Idf 15 untuk Query 51-100	114
5.9	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.3 Term Limit 25 dan Factor Tf-Idf 15 untuk Query 101-150	116
5.10	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.3 Term Limit 25 dan Factor Tf-Idf 15 untuk Query 151-225	119
5.11	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.5 Term Limit 50 dan Factor Tf-Idf 30 untuk Query 1-50	122
5.12	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.5 Term Limit 50 dan Factor Tf-Idf 30 untuk Query 51-100	124
5.13	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan Boost Query 0.5 Term Limit 50 dan Factor Tf-Idf 30 untuk Query 101-150	127

5.14	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan <i>Boost Query</i> 0.5 <i>Term Limit</i> 50 dan <i>Factor Tf-Idf</i> 30 untuk <i>Query</i> 151-225	129
5.15	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan <i>Boost Query</i> 2 <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 1-50	132
5.16	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan <i>Boost Query</i> 2 <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 51-100	135
5.17	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan <i>Boost Query</i> 2 <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 101-150	137
5.18	Tabel Hasil Pengujian Rochio <i>feedback</i> dengan <i>Boost Query</i> 2 <i>Term Limit</i> 75 dan <i>Factor Tf-Idf</i> 50 untuk <i>Query</i> 151-225	140
5.19	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 2000 dan <i>Factor</i> 25 untuk <i>Query</i> 1-50	143
5.20	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 2000 dan <i>Factor</i> 25 untuk <i>Query</i> 51-100	146
5.21	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 2000 dan <i>Factor</i> 25 untuk <i>Query</i> 101-150	148
5.22	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 2000 dan <i>Factor</i> 25 untuk <i>Query</i> 151-225	151
5.23	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 3000 dan <i>Factor</i> 50 untuk <i>Query</i> 1-50	154
5.24	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 3000 dan <i>Factor</i> 50 untuk <i>Query</i> 51-100	156
5.25	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 3000 dan <i>Factor</i> 50 untuk <i>Query</i> 101-150	159
5.26	Tabel Hasil Pengujian <i>Probabilistic feedback</i> dengan MU 3000 dan <i>Factor</i> 50 untuk <i>Query</i> 151-225	161

DAFTAR KODE PROGRAM

5.1	CranfieldTexCollection.java	87
5.2	CranfieldHandlerDocument.java	88
5.3	CranfieldIndexer.java	89
5.4	SearchRochio.java	89
5.5	RochioFeedback.java	90
5.6	ProbabilisticFeedback.java	91
A.1	CranfieldDocument.java	169
A.2	CranfieldTextCollection.java	169
A.3	CranfieldHandlerDocument.java	170
A.4	CranfieldIndexer.java	171
A.5	ReadDocumentIndex.java	172
B.1	SearcherRochio.java	173
B.2	SearcherProbabilistic.java	174
B.3	SearchCranfieldRochioController.java	175
B.4	SearchCranfieldProbabilisticController.java	176
B.5	RochioFeedback.java	176
B.6	ProbabilisticFeedback.java	177
B.7	TfIdfWeight.java	179
C.1	CranfieldQuery.java	183
C.2	CranfieldRelevanceQuery.java	185
C.3	QueryReaderCranfield.java	186
C.4	QueryReaderRelevanceCranfield.java	186
C.5	SearchRochioTester.java	187
C.6	RochioFeedbackTester.java	188
C.7	TfIdfTester.java	189
C.8	SearcherRochioTester.java	191
C.9	RochioValueTester.java	192
C.10	SearchProbabilisticTest.java	192
C.11	ProbabilisticFeedbackTester.java	193
C.12	SearcherProbabilisticTester.java	195
C.13	ProbabilisticValueTester.java	197
D.1	RelevanceFeedbackDocumnet.java	199
D.2	FXMLDocumentController.java	199
D.3	FXMLChartPrecisionRochioController.java	207
D.4	FXMLChartRecallRochioController.java	208
D.5	FXMLChartPrecisionProbabilisticController.java	209
D.6	FXMLChartRecallProbabilisticController.java	210

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 probabilitas *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 *Rocchio 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 *library* 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 diimplementasi 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 retrieval*, *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 retrieval*, teknik *relevance feedback*, 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.