

**RELEVANCE FEEDBACK PADA SISTEM TEMU KEMBALI
INFORMASI DOKUMEN *E-BOOK* BERBAHASA INDONESIA
MENGUNAKAN METODE BM25**

SKRIPSI

**Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

Disusun oleh:
Tasya Agiyola
NIM: 155150201111313



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019**

PENGESAHAN

RELEVANCE FEEDBACK PADA SISTEM TEMU KEMBALI INFORMASI
DOKUMEN E-BOOK BERBAHASA INDONESIA MENGGUNAKAN
METODE BM25

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Tasya Agiyola
NIM: 155150201111313

Skripsi ini telah diuji dan dinyatakan lulus pada
22 April 2019
Telah diperiksa dan disetujui oleh:

Pembimbing I



Indriati, S.T., M.Kom.
NIP: 19831013 201504 2

Pembimbing II



Bayu Rahayudi, S.T., M.T.
NIP: 19740712 200604 1

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 April 2019



Tasya Agiyola

NIM: 155150201111313

PRAKATA

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “*Relevance Feedback* pada Sistem Temu Kembali Informasi Dokumen *E-book* Berbahasa Indonesia menggunakan Metode BM25” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Ibu Indriati, S.T., M.Kom., selaku dosen Pembimbing I dan Bapak Bayu Rahayudi, S.T., M.T., selaku dosen Pembimbing II, yang telah dengan sabar membimbing dan memberikan arahan kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik,
2. Bapak Agus Wahyu Widodo, S.T., M.Cs., selaku Ketua Program Studi Teknik Informatika,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D., selaku Ketua Jurusan Teknik Informatika,
4. Bapak Randy Cahya Wihandika, S.ST., M.Kom., selaku dosen Penasehat Akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi,
5. Ayahanda Sufrianto dan Ibunda Suhartini, S.Pd., atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa memberikan doa dan semangat demi terselesaikannya skripsi ini,
6. Kedua kakak penulis Shopila Putri Sufty, S.Pd., dan Dwi Saras Mita Am.Keb., serta keluarga besar yang telah memberikan semangat yang tak henti-hentinya untuk segera menyelesaikan skripsi ini,
7. Sahabat-sahabat yang selalu mendukung penulis sejak SMA, Yulia Indriani, Syintiagung A. Siregar, Riska Pratiwi, Vivi Andini, dan Vindy Balqish R., terima kasih sudah selalu memberikan semangat dan dukungan selama ini. Semoga secepatnya kita bisa berkumpul lagi,
8. Sahabat-sahabat seperjuangan di Malang, Faradila Puspa W., Shafitri N. Putri, Ardia Regita P., Femilia N., dan Witriana S., yang selalu menjadi penyemangat, membantu segala kesulitan, dan selalu memberi kekuatan penuh kepada penulis selama 3,5 tahun di Malang. Semoga mimpi-mimpi kita semua bisa cepat tercapai,
9. Seluruh civitas academia Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 22 April 2019

Tasya Agiyola

Email: tasyaagi@student.ub.ac.id



ABSTRAK

Tasya Agiyola, Relevance Feedback pada Sistem Temu Kembali Informasi Dokumen *E-book* Berbahasa Indonesia menggunakan Metode BM25

Pembimbing: Indriati, S.T., M.Kom. dan Bayu Rahayudi, S.T., M.T.

Pemanfaatan *e-book* pada perkembangan teknologi digital memberikan kemudahan bagi pengguna untuk mendapatkan informasi lebih praktis daripada harus menggunakan buku cetak. Jumlah *e-book* yang tersebar di internet sangat banyak dan beragam, oleh karena itu diperlukan suatu sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia. Untuk meningkatkan relevansi hasil dokumen yang dikembalikan, dapat diterapkan teknik *relevance feedback*. *Relevance feedback* adalah teknik dimana pengguna dapat memberikan *feedback* terhadap hasil pencarian dokumen sebelumnya. Pengurutan jumlah dokumen yang dikembalikan berdasarkan *query* dihitung menggunakan metode BM25. Penelitian ini bertujuan untuk mengetahui hasil penerapan dan hasil pengujian dari *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25. Berdasarkan hasil pengujian, nilai AVP setelah *relevance feedback* mengalami penurunan. Pada pengujian berdasarkan nilai K, nilai AVP sebelum RF adalah 0,592, setelah RF(20) adalah 0,558, dan setelah RF(50) adalah 0,573. Sedangkan pada pengujian berdasarkan *expansion terms*, nilai AVP sebelum RF adalah 0,593, setelah RF(20) adalah 0,587, dan setelah RF(50) adalah 0,570.

Kata Kunci: *E-book*, sistem temu kembali informasi, *relevance feedback*, BM25, *precision@K*

ABSTRACT

Tasya Agiyola, *Relevance Feedback on Information Retrieval Systems Indonesian E-book Documents use the BM25 Method*

Supervisors: Indriati, S.T., M.Kom. and Bayu Rahayudi, S.T., M.T.

The use of e-books on the development of digital technology makes it easy for users to get more practical information than having to use printed books. The number of e-books spread on the internet is very numerous and varied, therefore a system for retrieving information on e-book documents in Indonesian is needed. To improve the relevance of the results of the returned documents, relevance feedback techniques can be applied. Relevance feedback is a technique where users can provide feedback on previous document search results. Sorting the number of documents returned based on queries is calculated using the BM25 method. This study aims to determine the results of the application and the results of testing of relevance feedback on the retrieval system of Indonesian e-book document information using the BM25 method. Based on the test results, the AVP value after relevance feedback has decreased. In testing based on K values, the AVP value before RF is 0.592, after RF(20) is 0,558, and after RF(50) is 0.573. While in testing based on the expansion terms, the AVP value before RF is 0.593, after RF(20) is 0,587 and after RF(50) is 0.570.

Keywords: *E-book, information retrieval, relevance feedback, BM25, precision@K*

DAFTAR ISI

| | |
|---|-----|
| PENGESAHAN..... | i |
| PERNYATAAN ORISINALITAS | ii |
| PRAKATA | iii |
| ABSTRAK..... | v |
| <i>ABSTRACT</i> | vi |
| DAFTAR ISI..... | vii |
| DAFTAR GAMBAR..... | ii |
| DAFTAR TABEL | iii |
| DAFTAR KODE PROGRAM | iv |
| DAFTAR LAMPIRAN | v |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 3 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan Masalah | 3 |
| 1.6 Sistematika Pembahasan | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 6 |
| 2.1 Kajian Pustaka | 6 |
| 2.2 <i>E-book</i> | 12 |
| 2.3 Sistem Temu Kembali Informasi | 12 |
| 2.3.1 <i>Preprocessing</i> | 13 |
| 2.4 <i>Relevance Feedback</i> | 13 |
| 2.4.1 <i>Manual – Relevance Feedback</i> | 14 |
| 2.4.2 Segmentasi Dokumen..... | 14 |
| 2.4.3 Penyeleksian <i>Expansion Terms</i> | 14 |
| 2.5 Metode BM25 | 15 |
| 2.6 Evaluasi | 16 |
| BAB 3 METODE PENELITIAN..... | 17 |
| 3.1 Studi Pustaka..... | 17 |
| 3.2 Peralatan Pendukung..... | 17 |
| 3.3 Tipe Penelitian | 17 |



| | | |
|----------------------------------|--|----|
| 3.4 | Strategi Penelitian | 18 |
| 3.5 | Pengumpulan Data..... | 18 |
| 3.6 | Teknik Penerapan Algoritme..... | 18 |
| 3.7 | Teknik Analisis Data | 19 |
| BAB 4 PERANCANGAN..... | | 20 |
| 4.1 | Deskripsi Umum Sistem | 20 |
| 4.2 | Perancangan Algoritme..... | 20 |
| 4.2.1 | <i>Preprocessing</i> | 21 |
| 4.2.2 | Pembobotan | 28 |
| 4.2.3 | Pengurutan Menggunakan Metode BM25..... | 32 |
| 4.2.4 | Penerapan <i>Relevance Feedback</i> | 33 |
| 4.3 | Manualisasi | 38 |
| 4.3.1 | Tahap <i>Preprocessing</i> | 39 |
| 4.3.2 | Tahap Pembobotan | 42 |
| 4.3.3 | Penerapan metode BM25 | 43 |
| 4.3.4 | Penerapan <i>Relevance Feedback</i> | 45 |
| 4.4 | Perancangan Pengujian..... | 48 |
| 4.4.1 | Perancangan Pengujian berdasarkan nilai <i>K</i> | 48 |
| 4.4.2 | Perancangan Pengujian berdasarkan <i>Expansion Terms</i> | 49 |
| 4.5 | Perancangan Antarmuka..... | 49 |
| BAB 5 IMPLEMENTASI..... | | 50 |
| 5.1 | Lingkungan Implementasi | 50 |
| 5.1.1 | Lingkungan Perangkat Keras | 50 |
| 5.1.2 | Lingkungan Perangkat Lunak..... | 50 |
| 5.2 | Batasan Implementasi..... | 50 |
| 5.3 | Implementasi Algoritme | 51 |
| 5.3.1 | <i>Preprocessing</i> | 51 |
| 5.3.2 | Pembobotan | 52 |
| 5.3.3 | Pengurutan Menggunakan Metode BM25..... | 53 |
| 5.3.4 | Penerapan <i>Relevance Feedback</i> | 54 |
| 5.4 | Implementasi Antarmuka | 56 |
| BAB 6 HASIL DAN PEMBAHASAN | | 57 |

| | | |
|------------------|--|----|
| 6.1 | Pengujian <i>Precision@K</i> | 57 |
| 6.1.1 | Pengujian berdasarkan Nilai <i>K</i> | 57 |
| 6.1.2 | Pengujian berdasarkan <i>Expansion Terms</i> | 63 |
| 6.2 | Analisis Hasil Pengujian..... | 68 |
| BAB 7 | PENUTUP | 70 |
| 7.1 | Kesimpulan..... | 70 |
| 7.2 | Saran | 70 |
| DAFTAR REFERENSI | | 72 |
| LAMPIRAN | | 74 |



DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2. 1 Proses <i>Manual – Relevance Feedback</i> | 14 |
| Gambar 4. 1 Diagram Alir Sistem | 21 |
| Gambar 4. 2 <i>Flowchart</i> tahapan Preprocessing | 22 |
| Gambar 4. 3 <i>Flowchart</i> Subproses <i>Case Folding</i> | 23 |
| Gambar 4. 4 <i>Flowchart</i> Subproses Tokenisasi | 24 |
| Gambar 4. 5 <i>Flowchart</i> Subproses <i>Stopword Removal</i> | 25 |
| Gambar 4. 6 <i>Flowchart</i> Subproses <i>Stemming</i> | 26 |
| Gambar 4. 7 <i>Flowchart</i> Subproses <i>Stemming</i> (Lanjutan) | 27 |
| Gambar 4. 8 <i>Flowchart</i> Tahapan Pembobotan | 28 |
| Gambar 4. 9 <i>Flowchart</i> Perhitungan TF | 29 |
| Gambar 4. 10 <i>Flowchart</i> Perhitungan DF | 30 |
| Gambar 4. 11 <i>Flowchart</i> Perhitungan IDF | 31 |
| Gambar 4. 12 <i>Flowchart</i> Tahap Pengurutan menggunakan metode BM25 | 32 |
| Gambar 4. 13 <i>Flowchart</i> Tahapan <i>Relevance Feedback</i> | 33 |
| Gambar 4. 14 <i>Flowchart</i> Pemilihan segmen | 35 |
| Gambar 4. 15 <i>Flowchart</i> Segmentasi Dokumen | 36 |
| Gambar 4. 16 <i>Flowchart</i> Seleksi Perluasan <i>Term</i> | 37 |
| Gambar 5. 1 <i>Source Code</i> Menyiapkan data latih | 51 |
| Gambar 5. 2 <i>Source code</i> Tokenisasi | 51 |
| Gambar 5. 3 <i>Source Code</i> Penghapusan <i>Stopword</i> | 51 |
| Gambar 5. 4 <i>Source Code</i> Proses <i>Stemming</i> | 52 |
| Gambar 5. 5 <i>Source code</i> Pembobotan TF | 52 |
| Gambar 5. 6 <i>Source code Document Frequency (DF)</i> | 52 |
| Gambar 5. 7 <i>Source Code</i> untuk <i>Invers Document Frequency (IDF)</i> | 53 |
| Gambar 5. 8 <i>Source Code</i> Pengurutan dokumen | 53 |
| Gambar 5. 9 <i>Source Code</i> untuk <i>Manual Relevance Feedback</i> | 54 |
| Gambar 5. 10 <i>Source Code</i> untuk Segmentasi | 54 |
| Gambar 5. 11 <i>Source Code</i> untuk Pembobotan Segmen | 55 |
| Gambar 5. 12 <i>Source Code</i> untuk Pemilihan segmen | 55 |
| Gambar 5. 13 <i>Source Code</i> untuk Penyeleksian <i>Term</i> | 55 |
| Gambar 5. 14 <i>Source Code</i> untuk Pencarian dokumen setelah <i>relevance feedback</i> | 56 |
| Gambar 5. 15 Tampilan Antarmuka | 56 |
| Gambar 6. 1 Grafik Perbandingan AVP pengujian berdasarkan nilai K (<i>User B</i>) | 62 |
| Gambar 6. 2 Grafik Perbandingan AVP pengujian berdasarkan nilai K (<i>User B</i>) | 63 |
| Gambar 6. 3 Grafik Perbandingan AVP pengujian berdasarkan <i>Expansion Terms (User A)</i> | 66 |
| Gambar 6. 4 Grafik Perbandingan AVP pengujian berdasarkan <i>Expansion Terms (User B)</i> | 67 |



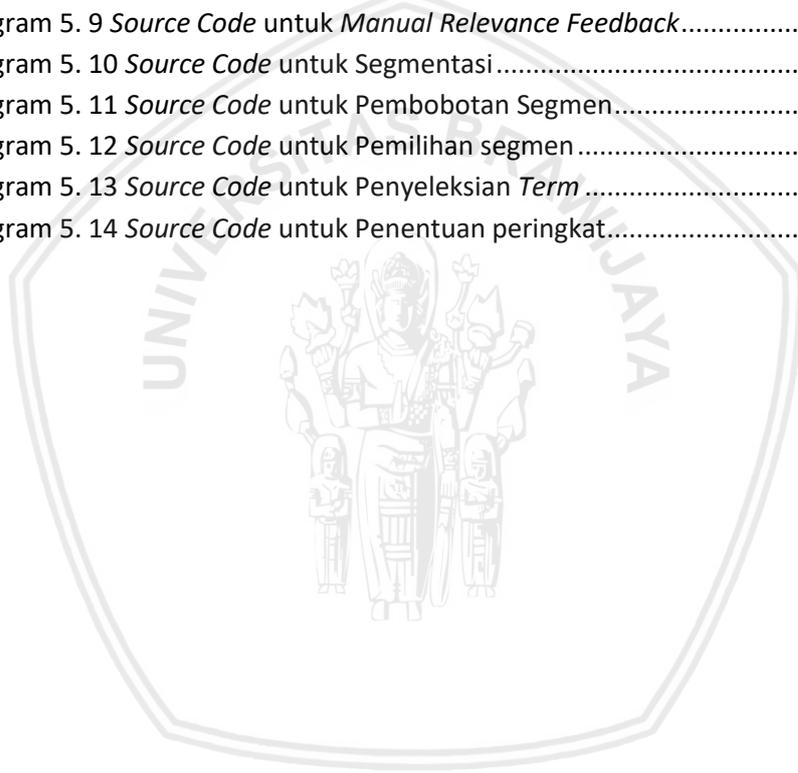
DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Kajian Pustaka | 10 |
| Tabel 4. 1 <i>Dataset</i> | 38 |
| Tabel 4. 2 <i>Case Folding</i> | 39 |
| Tabel 4. 3 Tokenisasi | 40 |
| Tabel 4. 4 <i>Stopword Removal</i> | 41 |
| Tabel 4. 5 <i>Stemming</i> | 41 |
| Tabel 4. 6 TF dan DF | 42 |
| Tabel 4. 7 Manualisasi Pembobotan Kata..... | 43 |
| Tabel 4. 8 Hasil Pembobotan <i>Query</i> | 44 |
| Tabel 4. 9 Hasil Segmentasi Dokumen | 45 |
| Tabel 4. 10 Pembobotan <i>Query</i> berdasarkan segmentasi | 46 |
| Tabel 4. 11 Nilai Relevansi Segmen dengan <i>Query</i> | 47 |
| Tabel 4. 12 Hasil nilai TSV | 47 |
| Tabel 4. 13 Hasil Pengurutan Dokumen setelah <i>relevance feedback</i> | 48 |
| Tabel 4. 14 Pengujian berdasarkan nilai <i>K</i> | 48 |
| Tabel 4. 15 Pengujian berdasarkan <i>Expansion Terms</i> | 49 |
| Tabel 6. 1 Hasil Percobaan 1 (<i>User A</i>) | 57 |
| Tabel 6. 2 Hasil Percobaan 1 (<i>User B</i>) | 58 |
| Tabel 6. 3 Hasil Percobaan 2 (<i>User A</i>) | 58 |
| Tabel 6. 4 Hasil Percobaan 2 (<i>User B</i>) | 58 |
| Tabel 6. 5 Hasil Percobaan 3 (<i>User A</i>) | 59 |
| Tabel 6. 6 Hasil Percobaan 3 (<i>User B</i>) | 59 |
| Tabel 6. 7 Hasil Percobaan 4 (<i>User A</i>) | 59 |
| Tabel 6. 8 Hasil Percobaan 4 (<i>User B</i>) | 60 |
| Tabel 6. 9 Hasil Percobaan 5 (<i>User A</i>) | 60 |
| Tabel 6. 10 Hasil Percobaan 5 (<i>User B</i>) | 60 |
| Tabel 6. 11 Hasil Percobaan 6 (<i>User A</i>) | 61 |
| Tabel 6. 12 Hasil Percobaan 6 (<i>User B</i>) | 61 |
| Tabel 6. 13 Perbandingan AVP pengujian berdasarkan nilai <i>K</i> (<i>User A</i>) | 61 |
| Tabel 6. 14 Perbandingan AVP pengujian berdasarkan nilai <i>K</i> (<i>User B</i>) | 62 |
| Tabel 6. 15 Hasil Percobaan 7 (<i>User A</i>) | 64 |
| Tabel 6. 16 Hasil Percobaan 7 (<i>User B</i>) | 64 |
| Tabel 6. 17 Hasil Percobaan 8 (<i>User A</i>) | 64 |
| Tabel 6. 18 Hasil Percobaan 8 (<i>User B</i>) | 65 |
| Tabel 6. 19 Hasil Percobaan 9 (<i>User A</i>) | 65 |
| Tabel 6. 20 Hasil Percobaan 9 (<i>User B</i>) | 65 |
| Tabel 6. 21 Perbandingan AVP pengujian berdasarkan nilai <i>K</i> (<i>User A</i>) | 66 |
| Tabel 6. 22 Perbandingan AVP pengujian berdasarkan nilai <i>K</i> (<i>User B</i>) | 67 |



DAFTAR KODE PROGRAM

| | |
|---|----|
| Kode Program 5. 1 <i>Source Code</i> Penyiapan data latih..... | 51 |
| Kode Program 5. 2 <i>Source code</i> Tokenisasi..... | 51 |
| Kode Program 5. 3 <i>Source Code</i> Penghapusan <i>Stopword</i> | 51 |
| Kode Program 5. 4 <i>Source Code</i> Proses <i>Stemming</i> | 52 |
| Kode Program 5. 5 <i>Source code</i> <i>Term Frequency (TF)</i> | 52 |
| Kode Program 5. 6 <i>Source code</i> <i>Document Frequency (DF)</i> | 52 |
| Kode Program 5. 7 <i>Source Code</i> untuk <i>Invers Document Frequency (IDF)</i> | 53 |
| Kode Program 5. 8 <i>Source Code</i> Pengurutan dokumen..... | 53 |
| Kode Program 5. 9 <i>Source Code</i> untuk <i>Manual Relevance Feedback</i> | 54 |
| Kode Program 5. 10 <i>Source Code</i> untuk Segmentasi..... | 54 |
| Kode Program 5. 11 <i>Source Code</i> untuk Pembobotan Segmen..... | 55 |
| Kode Program 5. 12 <i>Source Code</i> untuk Pemilihan segmen..... | 55 |
| Kode Program 5. 13 <i>Source Code</i> untuk Penyeleksian <i>Term</i> | 55 |
| Kode Program 5. 14 <i>Source Code</i> untuk Penentuan peringkat..... | 56 |



DAFTAR LAMPIRAN

| | |
|---|----|
| Lampiran A. Data Latih..... | 74 |
| Lampiran B. Data Uji | 74 |
| Lampiran C. Data <i>Query</i> Baru..... | 75 |
| Lampiran D. Dokumen Relevan..... | 76 |



BAB 1 PENDAHULUAN

Bab ini membahas tentang pendahuluan yang berisi latar belakang dibuatnya penelitian mengenai *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan dari penelitian ini.

1.1 Latar Belakang

Perkembangan teknologi yang semakin pesat membuat banyaknya teknologi baru bermunculan, sehingga akan memudahkan segala kegiatan manusia. Salah satu sektor di bidang teknologi yang berdampak pada perkembangan teknologi adalah media informasi. Media informasi menjadi suatu hal yang sangat dibutuhkan dalam segala aspek kehidupan masyarakat. Saat ini sumber informasi tidak hanya didapatkan melalui pencarian manual seperti mencari informasi dari buku cetak, majalah, ataupun koran. Pencarian informasi kini dapat dilakukan secara otomatis dengan memanfaatkan teknologi digital. Salah satu media informasi yang menggunakan teknologi digital adalah *e-book*. Menurut Putera (2011), *e-book* adalah salah satu teknologi yang memanfaatkan perangkat elektronik untuk menayangkan informasi multimedia dalam bentuk yang ringkas dan dinamis. Pemanfaatan *e-book* pada perkembangan teknologi digital memberikan kemudahan bagi pengguna yang ingin mendapatkan informasi dengan lebih praktis daripada harus menggunakan buku cetak. Dari segi penggunaan, pemanfaatan *e-book* memberikan dampak positif yaitu berkurangnya penggunaan kertas yang berlebihan. Sedangkan dari segi penyimpanan, *e-book* telah tersimpan di perangkat elektronik sehingga dapat dibaca dimana saja dan sangat praktis untuk dibawa kemana-mana.

Seiring berjalannya waktu, kemudahan dalam pengaksesan informasi membuat semakin bertambahnya jumlah *e-book* yang tersebar di internet, sehingga pengguna akan sulit mendapatkan *e-book* yang relevan sesuai dengan kebutuhannya. Berdasarkan masalah tersebut maka diberikan suatu usulan sistem temu kembali informasi (*information retrieval system*) yang dikhususkan pada pencarian dokumen *e-book* berbahasa Indonesia. Menurut Manning, Raghavan, dan Schutze (2009), sistem temu kembali informasi atau *information retrieval system* merupakan suatu proses penemuan informasi (biasanya dokumen) dari koleksi data yang sangat besar dan tidak terstruktur yang memenuhi permintaan atau kebutuhan pengguna berdasarkan *query* yang diinputkan oleh pengguna, sehingga menghasilkan dokumen yang relevan terhadap *query* pengguna. Subari dan Ferdinandus (2015) dalam penelitian sebelumnya membahas mengenai sistem temu kembali informasi untuk layanan kesehatan berobat dengan metode *vector space model*, pada penelitian tersebut didapatkan rata-rata nilai *precision* sebesar 62.14%.

Query masukkan pengguna sering kali ambigu atau tidak sesuai dengan yang dimaksud, sehingga informasi yang dihasilkan belum tentu relevan dengan yang dibutuhkan pengguna. Maka dari itu, untuk memberikan hasil dengan tingkat ketepatan informasi yang dibutuhkan pengguna menjadi lebih akurat, dapat digunakan teknik *Relevance Feedback*. *Relevance Feedback* merupakan salah satu teknik dalam sistem temu kembali informasi, dimana setelah sistem menampilkan hasil pencarian informasi pertama, pengguna dapat memberikan umpan balik seperti memilih atau menilai dokumen-dokumen yang relevan dengan kebutuhan pengguna, untuk selanjutnya digunakan kembali oleh sistem dalam mencari dokumen yang lebih relevan dari pencarian sebelumnya. Berdasarkan *feedback* dari pengguna, sistem kemudian melakukan reformulasi *query* baru. Dengan *query* yang telah dimodifikasi tersebut sistem akan mencari kembali dokumen yang relevan. Adisantoso, Ridha, dan Agusetyawan (2006) dalam penelitian sebelumnya mengenai *relevance feedback* pada temu kembali teks berbahasa Indonesia menggunakan metode *Ide-Dec-Hi* dan *Ide-Regular*. Penelitian tersebut menunjukkan bahwa teknik *relevance feedback* dapat meningkatkan kinerja sistem dengan memperoleh peningkatan sebesar 15.44% pada metode *Ide-Dec-Hi* dan peningkatan sebesar 14.54% metode *Ide-Regular*.

Pengembalian dokumen berdasarkan *query* pengguna dapat dilakukan dengan banyak metode, salah satunya dengan menerapkan metode BM25 (*Best Match 25*). Metode BM25 merupakan fungsi peringkat dokumen berdasarkan *query* yang diinputkan. Dokumen diurutkan berdasarkan nilai fungsi tertinggi. Metode BM25 dikembangkan dari metode *Binary Independence Model (BIM)*. *BIM* awalnya didesain untuk dokumen pendek sehingga dalam dokumen panjang, *BIM* belum mampu berfungsi dengan baik. Pardede, Husada, dan Riansyah (2005) dalam penelitian sebelumnya, membahas mengenai perbandingan metode BM25 dan metode PLSA (*Probabilistic Latent Semantic Analysis*) pada aplikasi *information retrieval*. Penelitian tersebut menunjukkan bahwa metode BM25 lebih mudah diterapkan karena memiliki nilai *precision* dan *F-Measure* yang lebih tinggi dibanding dengan metode PLSA.

Berdasarkan uraian di atas dan penelitian-penelitian sebelumnya, maka dikembangkan suatu sistem temu kembali informasi yang dikhususkan pada pencarian dokumen *e-book* berbahasa Indonesia. Sistem temu kembali ini menerapkan teknik *relevance feedback* untuk menghasilkan dokumen-dokumen relevan sesuai kebutuhan pengguna dengan metode BM25 sebagai fungsi pengurutan dari dokumen-dokumen yang dihasilkan sesuai *query* yang diinputkan pengguna.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan yang telah diuraikan, maka penulis dapat merumuskan permasalahan pada penelitian ini sebagai berikut.

1. Bagaimana hasil penerapan *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25 ?
2. Bagaimana hasil pengujian *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25 ?

1.3 Tujuan

Berdasarkan rumusan masalah di atas, tujuan penelitian dapat diuraikan sebagai berikut.

1. Untuk mengetahui hasil penerapan *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25.
2. Untuk mengetahui hasil pengujian dari *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut.

Bagi Pengguna :

Memberikan kemudahan kepada pengguna dalam mencari dokumen *e-book* berbahasa Indonesia yang relevan berdasarkan query yang dimasukan sehingga menghindari munculnya dokumen yang tidak relevan.

Bagi Penulis :

Menambah pengetahuan dan pengalaman penulis dalam menerapkan *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25.

Bagi Pembaca :

Sebagai bahan literatur/ pustaka untuk penelitian selanjutnya yang berhubungan dengan sistem temu kembali informasi.

1.5 Batasan Masalah

Dalam tugas akhir ini, penulis membatasi penelitian dengan beberapa batasan yang didefinisikan di bawah ini.

1. Koleksi dokumen yang digunakan dalam penelitian ini adalah dokumen sinopsis *e-book* berbahasa Indonesia.

2. Dokumen *e-book* yang digunakan dalam penelitian ini berjumlah 260 dokumen yang dikumpulkan dari *Google Play Books*.
3. Mekanisme *relevance feedback* yang digunakan adalah *manual relevance feedback*.

1.6 Sistematika Pembahasan

Laporan Penelitian dibuat dengan sistematika sebagai uraian dan gambaran dari penyusunan skripsi yang terbagi menjadi beberapa bab sebagai berikut.

BAB 1 PENDAHULUAN

Bab ini membahas tentang pendahuluan yang berisi latar belakang dibuatnya penelitian mengenai *relevance feedback* pada sistem temu kembali dokumen *e-book* berbahasa Indonesia menggunakan metode BM25, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan dari penelitian ini.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan tentang landasan kepastakaan yang berisi kajian pustaka dari penelitian sebelumnya yang berhubungan dengan konsep sistem temu kembali informasi. Dasar teori-teori yang berkaitan dengan *information retrieval*, koleksi dokumen, teknik *relevance feedback*, dan metode BM25 yang mendukung dalam pelaksanaan penelitian dan penulisan skripsi ini.

BAB 3 METODOLOGI PENELITIAN

Bab ini membahas tentang serangkaian aturan dan langkah yang berkaitan dengan penelitian untuk menyelesaikan permasalahan. Terdiri atas studi literatur, tipe penelitian, pengumpulan data, perancangan sistem, implementasi sistem, pengujian sistem, pengambilan kesimpulan, dan jadwal penelitian.

BAB 4 PERANCANGAN

Bab ini membahas tentang analisis kebutuhan dan perencanaan rancangan *relevance feedback* pada sistem temu kembali informasi dokumen *e-book* menggunakan metode BM25, yang berupa deskripsi umum sistem, perancangan algoritme, pengujian, antarmuka.

BAB 5 IMPLEMENTASI

Bab ini membahas mengenai implementasi metode BM25 dalam pencarian dokumen *e-book* berbahasa Indonesia menggunakan teknik *relevance feedback*. Implementasi memiliki tahapan yaitu spesifikasi sistem, batasan implementasi, implementasi metode dan antarmuka.

BAB 6 HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang proses analisis dari hasil pengujian menggunakan metode BM25 yang telah diimplementasikan dengan menggunakan pengujian keakurasian data.

BAB 7 PENUTUP

Bab ini menjelaskan tentang kesimpulan dari penelitian yang dilakukan berdasarkan hasil pengujian yang telah didapatkan dari sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia dengan teknik *relevance feedback* menggunakan metode BM25, beserta saran-saran penulis kepada penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini menjelaskan tentang landasan kepastakaan yang berisi kajian pustaka dari penelitian-penelitian sebelumnya yang berhubungan dengan konsep sistem temu kembali informasi. Dasar teori-teori yang berkaitan dengan *information retrieval*, koleksi dokumen, teknik *relevance feedback*, dan metode BM25 yang mendukung dalam pelaksanaan penelitian dan penulisan skripsi ini.

2.1 Kajian Pustaka

Penelitian-penelitian yang telah dilakukan sebelumnya berkaitan dengan teknik *relevance feedback* pada temu kembali informasi dilakukan oleh (Adisantoso, Ridha, & Agusetyawan, 2006), (Erwin dan Mandala, 2004), dan (Yugianus, Dachlan, & Hasanah, 2013). Teknik *relevance feedback* adalah salah satu cara untuk meningkatkan hasil kinerja relevansi dokumen yang dihasilkan oleh sistem temu kembali informasi. *Relevance feedback* bertujuan untuk meningkatkan *recall* (kemiripan atau kecocokan antara permintaan informasi dengan jawaban permintaan) dan *precision* (proporsi jumlah dokumen yang ditemukan kembali oleh sistem) berdasarkan informasi dari dokumen-dokumen relevan yang diidentifikasi pengguna. Dari beberapa penelitian sebelumnya mengenai *relevance feedback*, yang membedakan dengan penelitian sebelumnya adalah pada penelitian ini teknik *relevance feedback* dilakukan untuk pencarian dokumen *e-book* berbahasa Indonesia dengan menggunakan metode BM25. Penelitian sebelumnya mengenai metode BM25 dilakukan oleh (Pardede, Husada, & Riansyah, 2005), (Sari dan Adriani, 2014), dan (Tjandra dan Widiasri, 2016). Metode BM25 adalah suatu fungsi pengurutan yang digunakan oleh *search engine* untuk mengurutkan dokumen yang relevan berdasarkan kata kunci yang dimasukkan pengguna. Yang membedakan dari penelitian sebelumnya, pada penelitian ini penerapan metode BM25 dilakukan untuk mencari relevansi tertinggi dalam pencarian dokumen *e-book* berbahasa Indonesia dengan menggunakan teknik *relevance feedback*.

Berdasarkan beberapa penelitian di atas, penulis mengusulkan untuk membuat sistem temu kembali informasi pada pencarian dokumen *e-book* berbahasa Indonesia dengan menerapkan teknik *relevance feedback* dan fungsi pengurutan menggunakan metode BM25. Untuk lebih memahami mengenai kajian-kajian pustaka yang digunakan dalam penelitian ini, penulis membuat tabel kajian pustaka dengan nama Tabel 2.1.

Tabel 2.1 Kajian Pustaka

| NO | JURNAL | OBJEK | METODE | HASIL |
|----|--|--|---|--|
| 1. | <i>Relevance Feedback</i> pada Temu-Kembali Teks Berbahasa Indonesia dengan Metode <i>IDE-DEC-HI</i> dan <i>IDE-REGULAR</i> (Adisantoso, J., Ridha A., & Agusetyawan A.W., 2006) | Koleksi dokumen berupa artikel-artikel pertanian berbahasa Indonesia dari berbagai situs media massa sebanyak 500 dokumen. | Metode <i>IDE-DEC-HI</i> dan <i>IDE-REGULAR</i> | Hasil pengujian pada penelitian tersebut adalah kinerja sistem tanpa <i>relevance feedback</i> adalah 0.447 sedangkan setelah <i>relevance feedback</i> , pada metode <i>Ide-Dec-Hi</i> mencapai 0.516, meningkat 15.44% dan metode <i>Ide-Regular</i> mencapai 0.512, meningkat sebanyak 14.54%. sehingga <i>relevance feedback</i> dapat digunakan dalam penelitian ini karena mengalami peningkatan kinerja yang cukup baik. |
| 2. | <i>Relevance Feedback</i> pada Temu Kembali Informasi Menggunakan Algoritma Genetika (Erwin A.H dan Mandala, 2004) | Koleksi dokumen yang diambil dari koleksi ADI, CISI, CACM, dan INSPEC. | Algoritma Genetika | Hasil pengujian pada penelitian tersebut adalah nilai rata-rata metode rochio untuk <i>recall</i> adalah 0.65 dan <i>recall</i> untuk metode genetika adalah 0.73. Nilai rata-rata metode rochio untuk <i>precision</i> adalah 0.041 dan <i>precision</i> untuk metode genetika adalah 0.042. Nilai rata-rata <i>Non Interpolated Average Precision (NAP)</i> untuk metode Rochio adalah 0.21 dan nilai NAP untuk metode genetika adalah 0.192. Sehingga dapat disimpulkan tingkat <i>recall</i> dan <i>precision</i> melebihi metode rochio, dengan peningkatan <i>recall</i> sebesar 12.30 persen dan <i>precision</i> 2.43 persen. Sedangkan untuk nilai <i>Non Interpolated Average Precision (NAP)</i> menurun sebesar 9.37 persen. |



| NO | JURNAL | OBJEK | METODE | HASIL |
|----|---|--|-----------------------------------|--|
| 3. | Implementasi dan Perbandingan Metode Okapi BM25 dan PLSA pada Aplikasi <i>Information Retrieval</i> (Pardede, J., Husada, M.G., & Riansyah R., 2005) | Dokumen-dokumen berbahasa Indonesia yang berformat *.doc, *.txt, *.docx, dan *.pdf. | Metode Okapi BM25 dan PLSA | Hasil pengujian dari penelitian tersebut diperoleh bahwa, metode BM25 memiliki kinerja yang lebih tinggi dalam melakukan perangkangan dengan nilai rata-rata <i>F-Measure</i> metode BM25 adalah 61.649 dan metode PLSA adalah 56.8877. |
| 4. | <i>Learning to Rank for Determining Relevant Document in Indonesian-English Cross Language Information Retrieval using BM25</i> (Sari, S., DAN Adriani, M., 2014) | Data diambil dari CLEF 2006 sebanyak 1240 dokumen | Metode BM25 | Hasil penelitian menunjukkan bahwa, metode BM25 menghasilkan kinerja yang baik sebagai fitur pengurutan kata pada proses ini. |
| 5. | Pengembangan Sistem Penelusuran Katalog Perpustakaan dengan Metode <i>Rocchio Relevance Feedback</i> (Yugianus, et al., 2013) | Koleksi dokumen katalog perpustakaan dengan kriteria data yaitu judul buku, deksripsi, jenis, penerbit, dan penulis. | <i>Rocchio Relevance Feedback</i> | Hasil penelitian ini adalah pengimplementasian metode <i>rocchio relevance feedback</i> dalam sistem penelusuran katalog perpustakaan menghasilkan nilai kemiripan yang tinggi diantara dokumen lain berdasarkan inputan <i>query</i> dari pengguna. |

| NO | JURNAL | OBJEK | METODE | HASIL |
|----|--|---|-----------------------------|---|
| 6. | Sistem Repositori Tugas Akhir Mahasiswa dengan Fungsi Peringkat Okapi BM25 (Tjandra, E., dan Widiasri, M., 2016) | Koleksi dokumen tugas akhir mahasiswa dengan kriteria judul, pengarang, pembimbing 1, pembimbing 2, dan status pengarang. | Fungsi Peringkat Okapi BM25 | Hasil penelitian ini adalah bahwa proses pencarian dokumen dalam sistem repositori tugas akhir mahasiswa ini menghasilkan kinerja yang lebih relevan dengan penggunaan fungsi peringkat Okapi BM25. |

2.2 E-book

Menurut Putera (2011), *e-book* adalah salah satu teknologi yang memanfaatkan komputer untuk menayangkan informasi multimedia dalam bentuk yang ringkas dan dinamis. *E-book* mampu mengintegrasikan tayangan suara, grafik, gambar, animasi, maupun *movie* sehingga informasi yang disajikan lebih kaya dibandingkan dengan buku konvensional. Dengan adanya *e-book*, pembaca tidak perlu untuk menyimpan buku-buku favoritnya dalam bentuk fisik, dan juga memudahkan bagi penulis dalam menyebarkan tulisan-tulisannya. File-file yang biasa digunakan untuk pengemasan *e-book* adalah format *pdf*, *exe*, *doc*, *ppt*, dan sebagainya. Akan tetapi tidak semua file dokumen yang berekstensi *pdf* atau *exe* disebut *e-book*.

2.3 Sistem Temu Kembali Informasi

Menurut Manning, Raghavan, dan Schutze (2009), sistem temu kembali informasi atau *information retrieval system* merupakan suatu proses penemuan informasi (biasanya dokumen) dari koleksi data yang sangat besar dan tidak terstruktur yang memenuhi permintaan atau kebutuhan pengguna berdasarkan *query* yang diinputkan oleh pengguna, sehingga menghasilkan dokumen yang relevan terhadap *query* pengguna. Prinsip kerja *information retrieval* adalah mengambil semua dokumen yang relevan dan mengabaikan dokumen yang tidak relevan ketika seorang pengguna menginputkan *query* untuk mencari dokumen yang sesuai kebutuhan pengguna. Selain itu menurut Hasugian (2003), sistem temu kembali informasi terkandung sejumlah kegiatan yang meliputi proses penyimpanan, penyediaan representasi, identifikasi serta pencarian atau penelusuran dokumen yang relevan pada suatu *database*, dalam rangka memenuhi kebutuhan informasi dari pengguna. Dari sekian banyak sistem temu kembali yang ada, salah satu diantaranya adalah sistem temu kembali informasi berbasis teks atau tekstual.

2.3.1 Preprocessing

Preprocessing merupakan tahapan pertama yang perlu dilakukan dalam pengurutan dokumen. *Preprocessing* dilakukan untuk mengekstrak isi dokumen menjadi kumpulan *terms*. *Preprocessing* terdiri dari beberapa proses, yaitu:

1. **Case Folding**

Case Folding adalah tahapan awal dari *preprocessing*. Tujuannya adalah untuk mengubah seluruh huruf besar dari setiap isi dokumen menjadi huruf kecil semua. Perubahan ini dilakukan agar setiap kalimat menjadi konsisten. Di tahap ini, dilakukan juga penghilangan karakter yang tidak penting.

2. **Tokenisasi**

Setelah dilakukan *case folding*, selanjutnya adalah Tokenisasi. Tokenisasi merupakan proses pemisahan suatu rangkaian kata yang dipisahkan berdasarkan spasi.

3. **Stopword Removal**

Pada tahap ini, seluruh kata-kata yang tidak penting yang termasuk dalam *stopword list* dihapus karena tidak memiliki arti dan pengaruh untuk proses selanjutnya. *Stopword list* yang digunakan dalam penelitian ini adalah *stopword list Sastrawi*.

4. **Stemming**

Stemming dilakukan untuk menghapus semua imbuhan yang terdapat pada kata dasar. Imbuhan terdiri dari imbuhan awalan, akhiran, sisipan, maupun kombinasi. Penelitian ini menggunakan *library Python Sastrawi* dengan menerapkan algoritma dari Algoritma Nazief dan Adriani.

2.4 Relevance Feedback

Relevance feedback adalah interaksi antara pengguna dan sistem untuk secara bersama-sama merundingkan masalah *query* yang tepat untuk menggambarkan kebutuhan informasi. Proses *relevance feedback* akan mengubah *query* awal menjadi *query* baru yang menggambarkan lebih jelas mengenai kebutuhan informasi yang dibutuhkan oleh pengguna. *Relevance feedback* bertujuan untuk meningkatkan *recall* dan *precision* berdasarkan informasi dari dokumen-dokumen relevan yang diidentifikasi pengguna (Adisantoso, Ridha, Agusetyawan, 2004).

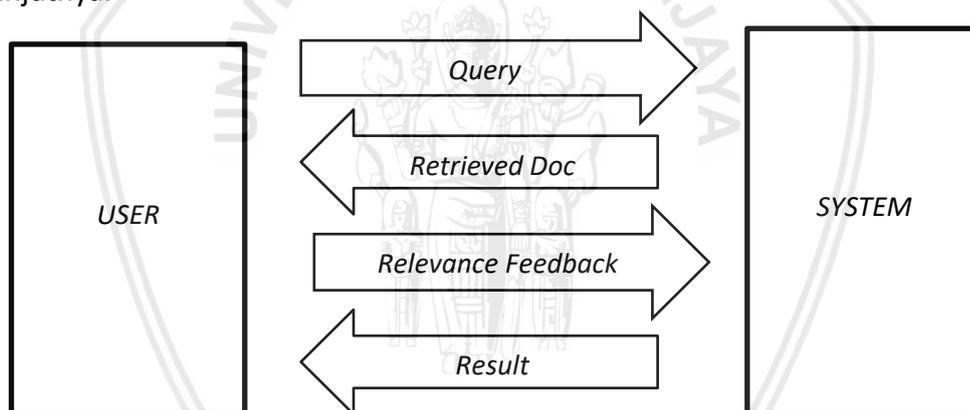
Dalam penentuan *relevance feedback* oleh pengguna dimaksudkan untuk mencari dokumen lanjut selain dari dokumen yang telah ditemukan tersebut. Apakah dengan proses ini akan ditemukan dokumen lain yang relevan atau tidak. Proses temu kembali informasi dengan proses *relevance feedback* yang baik akan menemukan dokumen-dokumen lain yang memiliki *relevance* dengan *query*. *Relevance Feedback* berguna untuk mendekatkan *query* dengan vektor rata-rata dokumen relevan dan menjauhkan dari vektor rata-rata dokumen tak relevan. Hal ini dapat dilakukan dengan menambahkan *term* pada *query* awal dan

pembobotan ulang, sehingga menggambarkan fungsi dalam membedakan dokumen relevan dan tidak relevan (Ruthven dan Lalmas, 2003).

Manning, Raghavan, dan Schutze (2009) mengatakan, ada beberapa metode untuk memodifikasi *query*, yaitu metode lokal dan metode global. *Relevance feedback* termasuk dalam metode lokal. Metode lokal terdiri dari 3 jenis, yaitu *Manual - Relevance Feedback* (umpan balik dilakukan oleh pengguna), *Pseudo - Relevance Feedback* (umpan balik dilakukan oleh sistem), dan *Indirect relevance feedback*.

2.4.1 Manual – Relevance Feedback

Manual – Relevance Feedback adalah jenis umpan balik dimana dilakukan secara manual oleh pengguna dengan memilih dokumen-dokumen yang relevan sesuai kebutuhan pengguna untuk diproses menghasilkan *query* baru yang akan digunakan dalam pencarian selanjutnya. Pada *manual-relevance feedback* sistem menampilkan hasil pada pencarian awal. Selanjutnya pengguna memberi tanda pada dokumen mana saja yang relevan, *feedback* tersebut kemudian diproses sistem untuk menghasilkan *query* baru yang akan digunakan pada pencarian selanjutnya.



Gambar 2. 1 Proses *Manual – Relevance Feedback*

Sumber: Mandala (2006)

2.4.2 Segmentasi Dokumen

Segmentasi dokumen merupakan tahapan yang dilakukan selanjutnya setelah proses *relevance feedback*. Dokumen dibagi menjadi beberapa segmen dengan masing-masing segmen berisi maksimal n terms. Segmen akan digunakan untuk memilih *term* sebagai *expansion terms*. Pemilihan segmen dilakukan berdasarkan peringkat teratas dari hasil pengurutan segmen. Segmen yang terpilih kemudian dilakukan penyeleksian *expansion terms*.

2.4.3 Penyeleksian *Expansion Terms*

Penyeleksian *expansion terms* dilakukan pada segmen n teratas. Berdasarkan penelitian oleh Yu, et al (2003), setiap *term* yang ada pada segmen

terpilih kecuali *terms* yang sama pada *query* awal diberi nilai bobot berdasarkan TSV (*Term Selection Value*) seperti persamaan (2.1):

$$TSV = IDF \times \frac{r}{R} \quad (2.1)$$

Keterangan:

IDF = *Invers Document Frequency*

r = Banyak dokumen/segmen yang terpilih yang mengandung *expansion terms i*

R = Banyaknya dokumen/segmen yang terpilih

Terms yang memiliki nilai TSV, *n* paling tinggi kemudian dijadikan *expansion query* yang digabungkan dengan *query* awal sehingga menjadi *query* baru untuk digunakan dalam pencarian selanjutnya.

2.5 Metode BM25

Metode BM25 banyak digunakan oleh mesin pencari untuk memberi peringkat sekumpulan dokumen sesuai dengan *query* yang diberikan. Dokumen yang memiliki nilai fungsi tertinggi merupakan dokumen yang paling sesuai dengan *query* yang diberikan. Fungsi BM25 menghitung 3 faktor dalam penentuan peringkat dokumen terhadap *query* yang diberikan. Faktor pertama adalah *TF* (*term frequency*), yaitu jumlah kemunculan suatu kata yang dicari dalam sebuah dokumen. Semakin besar nilai *TF* dalam sebuah dokumen, maka nilai fungsi peringkat suatu dokumen semakin besar. Faktor kedua adalah *IDF* (*invers document frequency of term*), yaitu nilai invers dari total dokumen yang berisi kata yang dicari. Semakin besar nilai *IDF* berarti semakin banyak dokumen yang memuat kata yang dicari. Artinya, semakin tinggi *IDF* maka kata yang dicari kurang baik digunakan untuk memberi peringkat, karena kata tersebut merupakan kata umum ditemukan pada banyak dokumen. Oleh karena itu untuk perhitungan ini digunakan nilai *inverse DF*. Faktor ketiga adalah panjang dokumen, yaitu banyaknya kata dalam sebuah dokumen tertentu. Dokumen yang di dalamnya terdapat kata yang dicari dan mempunyai jumlah kata lebih sedikit akan lebih baik peringkatnya daripada dokumen mempunyai jumlah kata lebih banyak (Russell & Norvig, 2010).

Fungsi metode BM25 untuk memberi peringkat dari *N* dokumen didefinisikan pada persamaan berikut (Russell & Norvig, 2010).

$$BM25(d_j, q_{1:N}) = \sum_{i=1}^N IDF(q_i) \cdot \frac{TF(q_i, d_j)(k+1)}{TF(q_i, d_j) + k(1-b + b \frac{|d_j|}{L})} \quad (2.2)$$

Dengan *L* dijabarkan pada persamaan berikut ini.

$$L = \frac{\sum_i |d_i|}{N} \quad (2.3)$$

Adapun fungsi *IDF* dapat dilihat pada persamaan berikut.

$$IDF(q_i) = \log \frac{N-DF(q_i)+0.5}{DF(q_i)+0.5} \quad (2.4)$$

Keterangan:

- q_i = kata yang dicari.
- $|d_j|$ = panjang dokumen d_j , banyaknya kata dalam dokumen d_j .
- $TF(q_i, d_j)$ = jumlah kemunculan kata q_i dalam sebuah dokumen d_j .
- $IDF(q_i)$ = nilai invers dari total dokumen yang berisi kata q_i .
- L = rata-rata panjang dokumen dari N dokumen.
- Terdapat dua parameter yang digunakan yaitu k dan b untuk proses evaluasi, nilai parameter yang biasa digunakan adalah $k = 2$ dan $b = 0.75$.

2.6 Evaluasi

Evaluasi yang digunakan pada penelitian ini adalah evaluasi temu kembali berperingkat berdasarkan nilai *precision@K*. Evaluasi ini bertujuan untuk mengetahui persentase dari hasil dokumen relevan diantara K dokumen teratas dengan mengabaikan dokumen yang urutannya di bawah K . Untuk menentukan nilai *precision@K*, dapat menggunakan persamaan 2.5.

$$precision@K = \frac{n}{K} \quad (2.5)$$

Keterangan:

- n adalah jumlah dokumen relevan pada peringkat K teratas
- K adalah nilai *threshold* peringkat

BAB 3 METODE PENELITIAN

Bab ini menjelaskan tentang langkah-langkah yang ditempuh dalam penelitian yang diusulkan. Adapun beberapa tahapan yang dilakukan dalam penelitian ini, yakni studi literatur, tipe penelitian, pengumpulan data, perancangan sistem, implementasi sistem, pengujian sistem, pengambilan keputusan, dan jadwal penelitian.

3.1 Studi Pustaka

Studi pustaka mempelajari mengenai dasar teori apa yang digunakan dalam menunjang penulisan dan pengerjaan proyek akhir. Teori-teori pendukung penulisan serta pemahaman tentang tugas akhir diperoleh dari buku, jurnal, *e-book*, penelitian sebelumnya yang memiliki topik yang berkaitan dengan proyek akhir ini, serta berupa bantuan dan mesin pencari internet. Referensi utama yang dibutuhkan sebagai penunjang penulisan ini adalah :

1. *E-book* berbahasa Indonesia
2. Sistem Temu Kembali Informasi
3. Teknik *Relevance Feedback*
4. Metode BM25

3.2 Peralatan Pendukung

Peralatan pendukung yang dibutuhkan dalam pembuatan sistem temu kembali informasi pencarian dokumen *e-book* dengan teknik *relevance feedback* menggunakan metode BM25 meliputi:

1. Kebutuhan *Hardware*, yaitu:
 - a. Intel® Core™ i5-7200U 2.5GHz
 - b. Memori 4GB DDR4
 - c. Harddisk kapasitas 1TB
 - d. Nvidia Geforce 940MX (2 GB DDR4)
2. Kebutuhan *Software*, yaitu:
 - a. Sistem operasi Windows 10 Pro 64-bit
 - b. Bahasa pemrograman Python
 - c. Spyder (Python 3.6)

3.3 Tipe Penelitian

Tipe yang digunakan dalam penelitian ini adalah tipe penelitian nonimplementatif yang menitikberatkan pada investigasi terhadap fenomena atau situasi tertentu, penganalisisan terhadap hubungan antar fenomena yang sedang dikaji untuk selanjutnya menghasilkan hasil berupa investigasi atau analisis ilmiah sebagai produk/artefak utamanya. Produk /artefak utama bisa dilakukan dengan melakukan survei, eksperimentasi, studi kasus, penelitian tindakan, studi etnografi, wawancara, kuisisioner, observasi, dan sebagainya.

Penelitian nonimplementatif memiliki dua pendekatan sesuai kegiatan penelitiannya, yaitu deskriptif dan analitik. Pada kegiatan penelitian ini, pendekatan dilakukan dengan analitik. Pendekatan penelitian secara analitik (*analytical/explanatory*) adalah sebuah kegiatan penelitian nonimplementatif yang dilakukan untuk menjelaskan derajat hubungan antar elemen dalam objek penelitian dengan fenomena/situasi tertentu yang sedang diteliti. Produk/artefak utama yang dihasilkan adalah hasil analisis. (Fakultas Ilmu Komputer Universitas Brawijaya, 2017).

3.4 Strategi Penelitian

Strategi penelitian adalah suatu strategi pengumpulan data yang menjadi objek, subjek, variabel, dan masalah yang diteliti agar data dapat sesuai dengan tujuan yang akan dicapai (Sukmadinata, 2009). Pada penelitian ini, strategi penelitian menggunakan metode kualitatif karena teori dikembangkan berdasarkan data yang dikumpulkan yaitu berupa dokumen sinopsis *e-book* untuk pengembangan teori *relevance feedback*. Penelitian kualitatif memiliki dua strategi yaitu interaktif dan noninteraktif, penelitian ini menggunakan strategi noninteraktif karena pengkajian dalam penelitian dilakukan berdasarkan analisis dokumen.

3.5 Pengumpulan Data

Pengumpulan data adalah tahapan yang dilakukan untuk memenuhi kebutuhan data yang diperlukan dalam pembangunan sistem temu kembali informasi. Data yang digunakan dalam penelitian ini merupakan data primer yaitu kumpulan dokumen sinopsis *e-book* berbahasa Indonesia yang dikumpulkan dari *Google Play* kategori buku sebanyak 260 dokumen. Teknik pengumpulan data menggunakan metode studi dokumentasi, yaitu pengumpulan data oleh peneliti dengan cara mengumpulkan dokumen-dokumen dari suatu sumber terpercaya. Selanjutnya, pengumpulan data dokumen *e-book* relevan berdasarkan inputan data uji (*query*) dilakukan oleh dua orang *user*, yaitu satu pakar dan satu mahasiswa. Pakar dalam penelitian ini adalah seorang Guru SMP dengan bidang studi Bahasa Indonesia.

3.6 Teknik Penerapan Algoritme

Penerapan algoritme dilakukan berdasarkan pada pengumpulan data dengan proses perancangan dan penerapan implementasi pencarian dokumen *e-book* menggunakan teknik *relevance feedback* dengan metode BM25. Penerapan algoritme metode BM25 digunakan untuk mencari hasil pengurutan dokumen berdasarkan *query* yang kemudian dilakukan *relevance feedback* sebagai tahap selanjutnya. Berikut adalah tahapan-tahapan yang akan dilakukan, yaitu:

1. Data masukkan berupa koleksi dokumen *e-book* berbahasa Indonesia dan *query* awal.
2. Melakukan tahapan *preprocessing* sebagai berikut:

- a. *Cleaning*, yaitu membersihkan karakter yang tidak penting seperti tanda baca, tag html, dsb.
 - b. *Case folding*, yaitu mengubah seluruh huruf besar menjadi huruf kecil.
 - c. Tokenisasi, yaitu memecah setiap dokumen menjadi kata berdasarkan spasi sebagai pemisahannya.
 - d. *Stopword Removal*, yaitu menghilangkan kata-kata *stopwords* yang tidak memiliki arti.
 - e. *Stemming*, yaitu menghilangkan setiap imbuhan yang terdapat pada kata dasar.
3. Melakukan tahapan pembobotan dengan mencari nilai *TF*, *DF*, dan *IDF*.
 4. Melakukan pengurutan dokumen berdasarkan *query* awal menggunakan metode BM25.
 5. Menerapkan teknik *relevance feedback* dengan memilih *n* dokumen yang relevan sesuai kebutuhan pengguna.
 6. Membagi setiap dokumen terpilih setelah *preprocessing* menjadi segmen-segmen dengan maksimal setiap segmen terdiri dari 50 kata.
 7. Perhitungan metode BM25 untuk memilih *n* segmen terbaik berdasarkan *query* awal
 8. Penyeleksian *expansion terms* dilakukan pada semua *terms* dengan segmen terpilih untuk mendapatkan *expansion query* yang selanjutnya digabungkan dengan *query* awal.
 9. Pencarian selanjutnya setelah *relevance feedback* dengan *query* baru untuk mendapatkan hasil pengurutan berupa dokumen-dokumen yang paling relevan.

3.7 Teknik Analisis Data

Analisis data dilakukan untuk menguji sistem apakah telah berjalan sesuai yang diharapkan. Pengujian sistem dilakukan berdasarkan evaluasi dari perhitungan nilai *precision@K*. Pengujian dilakukan dengan beberapa jenis percobaan untuk mendapatkan hasil evaluasi yang dapat dibandingkan. Pengujian ini dilakukan untuk memeriksa bagaimana pengaruh teknik *relevance feedback* dalam pencarian dokumen *e-book* berbahasa Indonesia menggunakan metode BM25.

BAB 4 PERANCANGAN

Pada bab ini akan membahas mengenai perancangan yang menggambarkan sistem temu kembali informasi seperti apa yang akan dibangun berdasarkan teori-teori dari referensi yang digunakan dalam penelitian ini. Perancangan pada bab ini terdiri dari deskripsi umum sistem, perancangan algoritme, perancangan pengujian dan perancangan antarmuka.

4.1 Deskripsi Umum Sistem

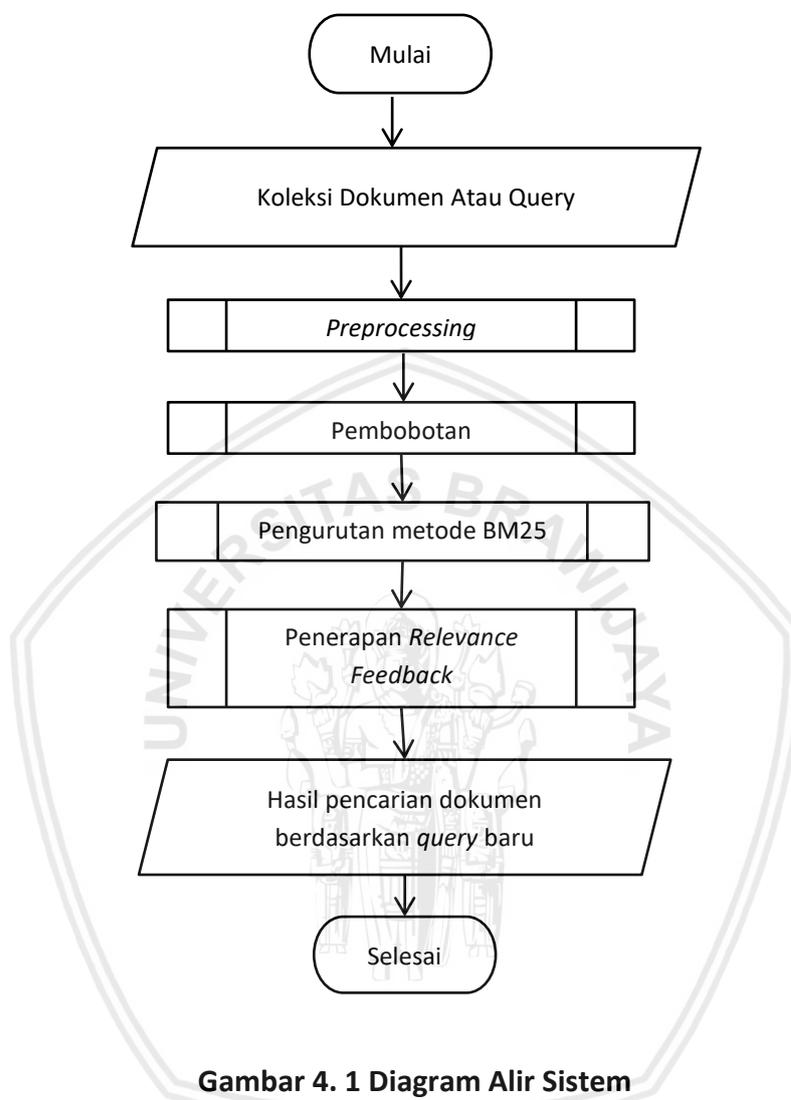
Sistem yang dibuat adalah sistem pencarian dokumen *e-book* dengan menggunakan teknik *relevance feedback*. Sistem pencarian akan menampilkan dokumen *e-book* secara berurut sesuai tingkat relevansi berdasarkan *query* yang dimasukkan oleh pengguna. Proses pengurutan dokumen *e-book* dilakukan dengan menggunakan metode BM25. Data yang digunakan dalam sistem adalah kumpulan sinopsis dari dokumen *e-book* yang tersebar di internet, sebanyak 260 data.

Terdapat tiga masukan pada sistem yaitu, *query* yang diinputkan oleh pengguna pada pencarian pertama, nilai K yang ditentukan pengguna, dan *feedback* yang diberikan pengguna untuk pencarian selanjutnya. *Feedback* yang diberikan adalah memilih dokumen-dokumen yang ditandai sebagai dokumen relevan atau sesuai dengan kebutuhan pengguna. Terdapat dua *output* pada sistem yaitu, hasil dokumen dengan peringkat K teratas berdasarkan *query* awal yang dimasukkan pengguna dan hasil dokumen dengan peringkat K teratas berdasarkan *feedback* yang diberikan pengguna pada pencarian sebelumnya.

4.2 Perancangan Algoritme

Perancangan algoritme menjabarkan mengenai tahapan-tahapan proses algoritme yang digunakan dalam menghasilkan tujuan yang ingin didapatkan dalam penelitian ini. Pada penelitian ini, tahapan proses algoritme dimulai dari tahap *preprocessing*. Pada tahap *preprocessing*, data diolah dengan menjalankan proses *case folding*, tokenisasi, menghapus *stopword*, dan *stemming*. Selanjutnya pada tahap pembobotan, setiap data *term* dicari nilai TF , DF , dan IDF -nya. Kemudian dilakukan pengurutan dokumen menggunakan metode BM25 berdasarkan pencarian *query* awal yang dimasukkan pengguna. Hasil pencarian awal tersebut menghasilkan dokumen-dokumen yang relevan berdasarkan nilai tertinggi dari perhitungan metode BM25. Pada tahap selanjutnya adalah penerapan teknik *relevance feedback*, dimana pengguna memberikan umpanbalik memilih N dokumen yang relevan menurut pengguna yang sesuai dengan kebutuhannya. Dokumen relevan yang terpilih tersebut selanjutnya dilakukan segmentasi, kemudian dilakukan *preprocessing* dan pembobotan. Nilai IDF yang didapatkan akan digunakan untuk menghitung nilai TSV yang merupakan nilai *expansion term*. *Top N terms* dengan nilai TSV tertinggi akan digunakan sebagai *query* baru untuk pencarian selanjutnya, sehingga akan

menghasilkan dokumen yang lebih *relevan* sesuai umpanbalik yang diberikan pengguna pada pencarian sebelumnya.

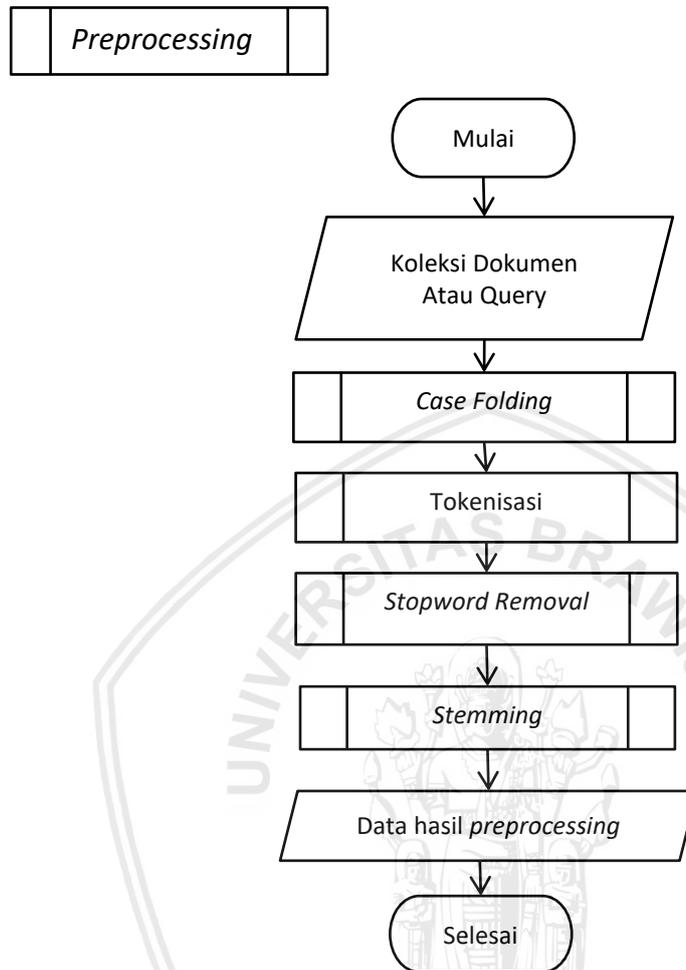


Gambar 4. 1 Diagram Alir Sistem

Berdasarkan diagram alir pada Gambar 4.1, diagram tersebut memiliki tahapan yang digambarkan secara umum. Setiap tahapan dalam diagram tersebut masih terdapat subproses-subproses yang harus dijalankan sebelum berpindah ke tahap selanjutnya. Tahapan-tahapan yang memiliki subproses adalah tahap *preprocessing*, pembobotan, pengurutan menggunakan metode BM25, dan yang terakhir penerapan *relevance feedback*.

4.2.1 Preprocessing

Pada tahap ini dilakukan dengan tujuan untuk menghasilkan data indeks berupa *term-term* yang didapat dari koleksi dokumen. Data indeks yang didapatkan akan digunakan untuk proses selanjutnya yaitu pembobotan data. Proses yang dilakukan dalam tahapan *preprocessing* dapat dilihat dari Gambar 4.2.

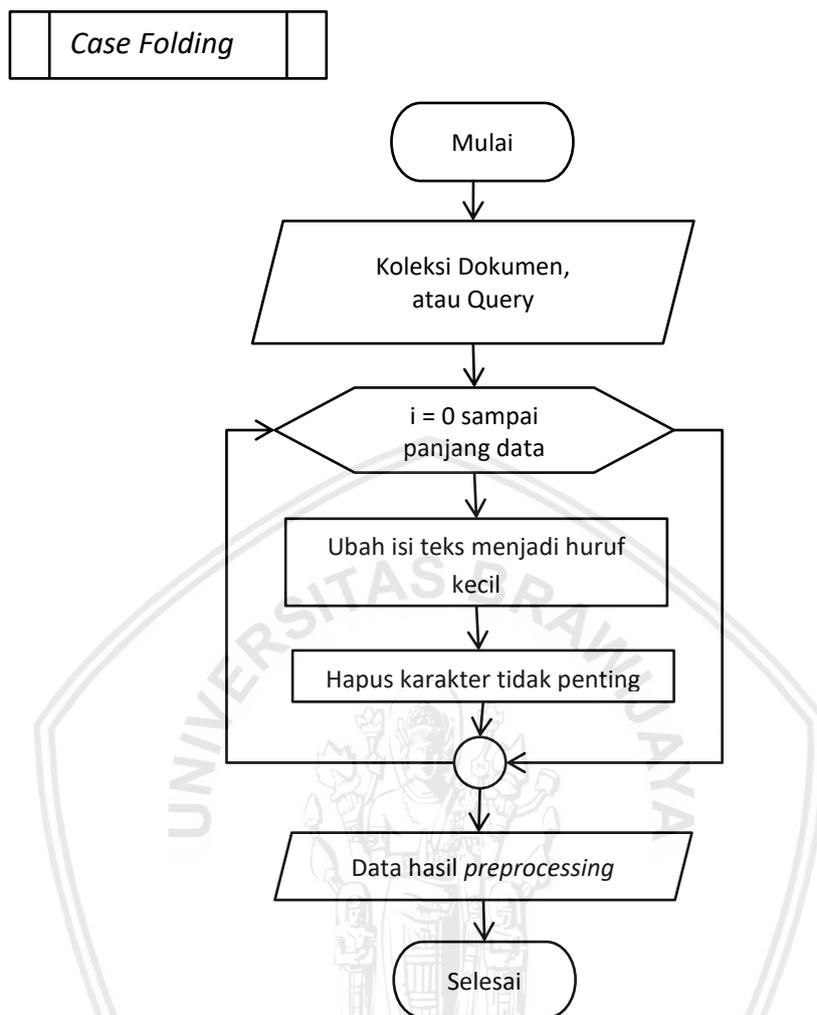


Gambar 4. 2 Flowchart tahapan Preprocessing

Pada Gambar 4.2 menggambarkan rangkaian proses dari tahapan *preprocessing* secara lebih detail. Adapun rangkaian proses dari tahapan *preprocessing* yang digunakan dalam penelitian ini adalah proses *case folding*, tokenisasi, penghapusan *stopword*, dan *stemming*. Pada tahap ini, koleksi dokumen atau *query* merupakan masukan dan data hasil *preprocessing* adalah keluarannya.

1. *Case Folding*

Dalam tahapan *preprocessing*, proses pertama yang dilakukan adalah *Case Folding*. *Case Folding* merupakan suatu proses dimana isi teks diubah menjadi huruf kecil semua. Proses yang dilakukan dalam *case folding* dapat dilihat dari Gambar 4.3.



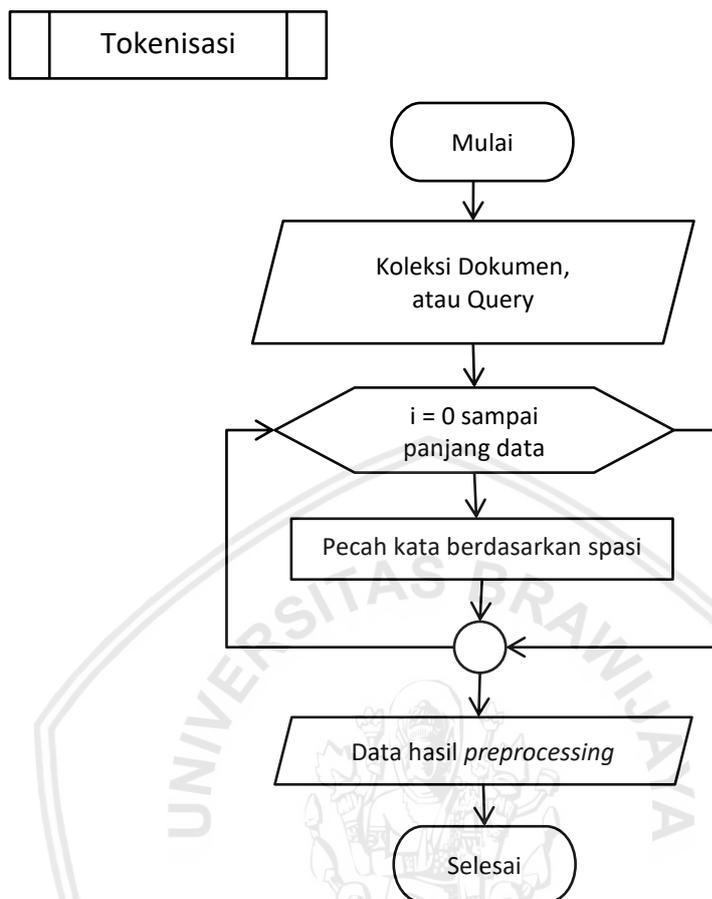
Gambar 4. 3 Flowchart Subproses Case Folding

Pada Gambar 4.3, proses *case folding* dapat dilakukan pada koleksi dokumen, *query*, dan segmen dokumen. Langkah pertama yang dilakukan adalah mengubah isi teks menjadi huruf kecil semua, selanjutnya adalah menghapus karakter yang tidak penting (angka, tanda baca, karakter khusus, *tag html*, *markup*, dan huruf yang tidak memiliki arti). Proses tersebut berulang sampai seluruh data telah ditokenisasi.

2. Tokenisasi

Tokenisasi merupakan suatu proses dimana data koleksi dokumen ataupun data *query* diolah dengan memecah rangkaian kata pada setiap isi dokumen berdasarkan spasi. Proses yang dilakukan dalam tokenisasi dapat dilihat dari Gambar 4.4.





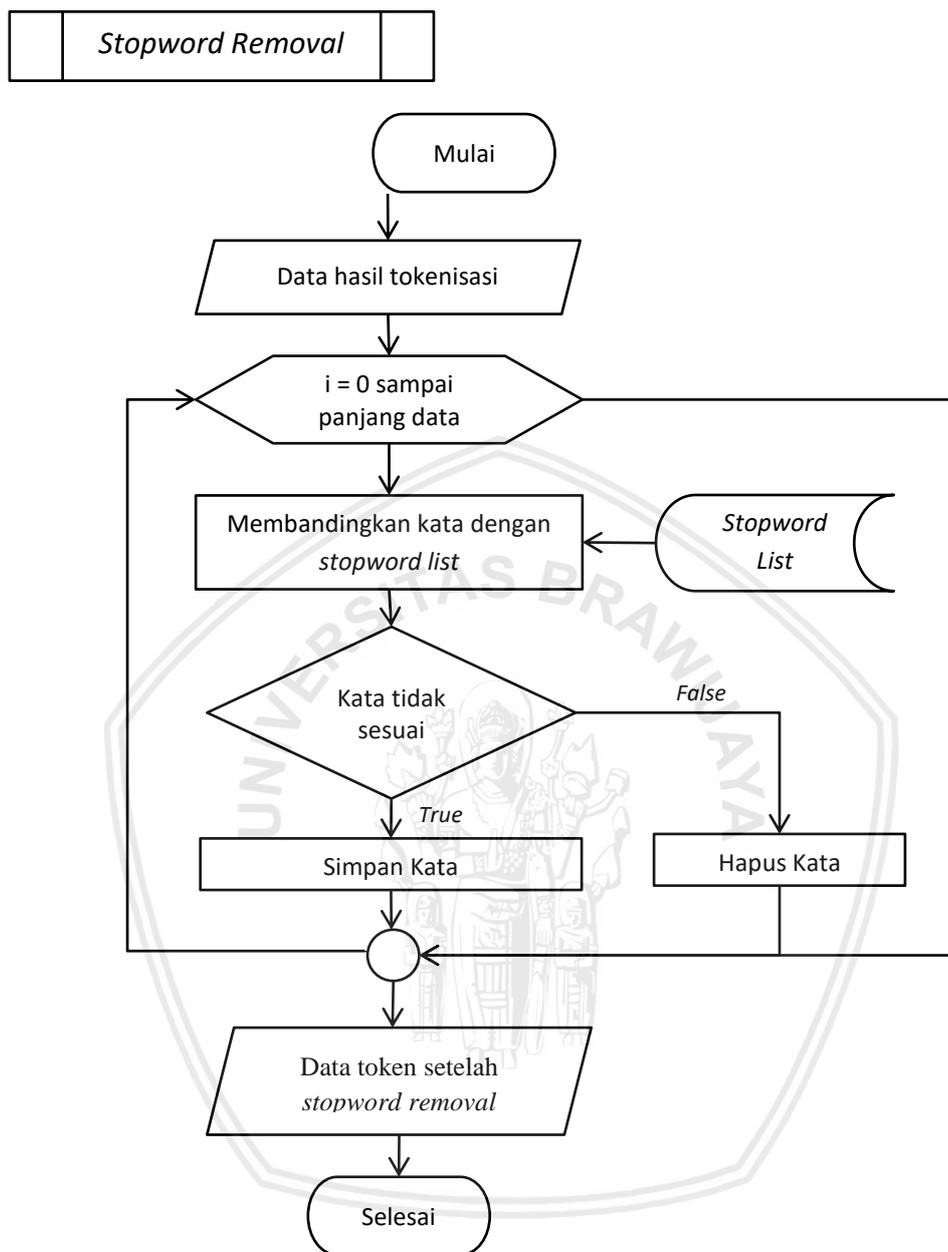
Gambar 4. 4 Flowchart Subproses Tokenisasi

Pada Gambar 4.4, proses tokenisasi dapat dilakukan pada koleksi dokumen, *query*, dan segmen dokumen. Langkah yang dilakukan adalah memecah setiap kata berdasarkan spasi yang menjadi pemisahannya. Proses tersebut berulang sampai seluruh data telah ditokenisasi.

3. *Stopword Removal*

Setelah tokenisasi, proses selanjutnya adalah menghapus *stopword*. *Stopword* merupakan kata-kata yang tidak berperan penting sehingga lebih baik dihapus. Kata-kata *stopword* yang ada di *stopword list* dihapus untuk proses selanjutnya. *Stopword list* yang digunakan dalam penelitian ini adalah *stopword list Sastrawi*. Proses yang dilakukan dalam *stopword removal* dapat dilihat dari Gambar 4.5.





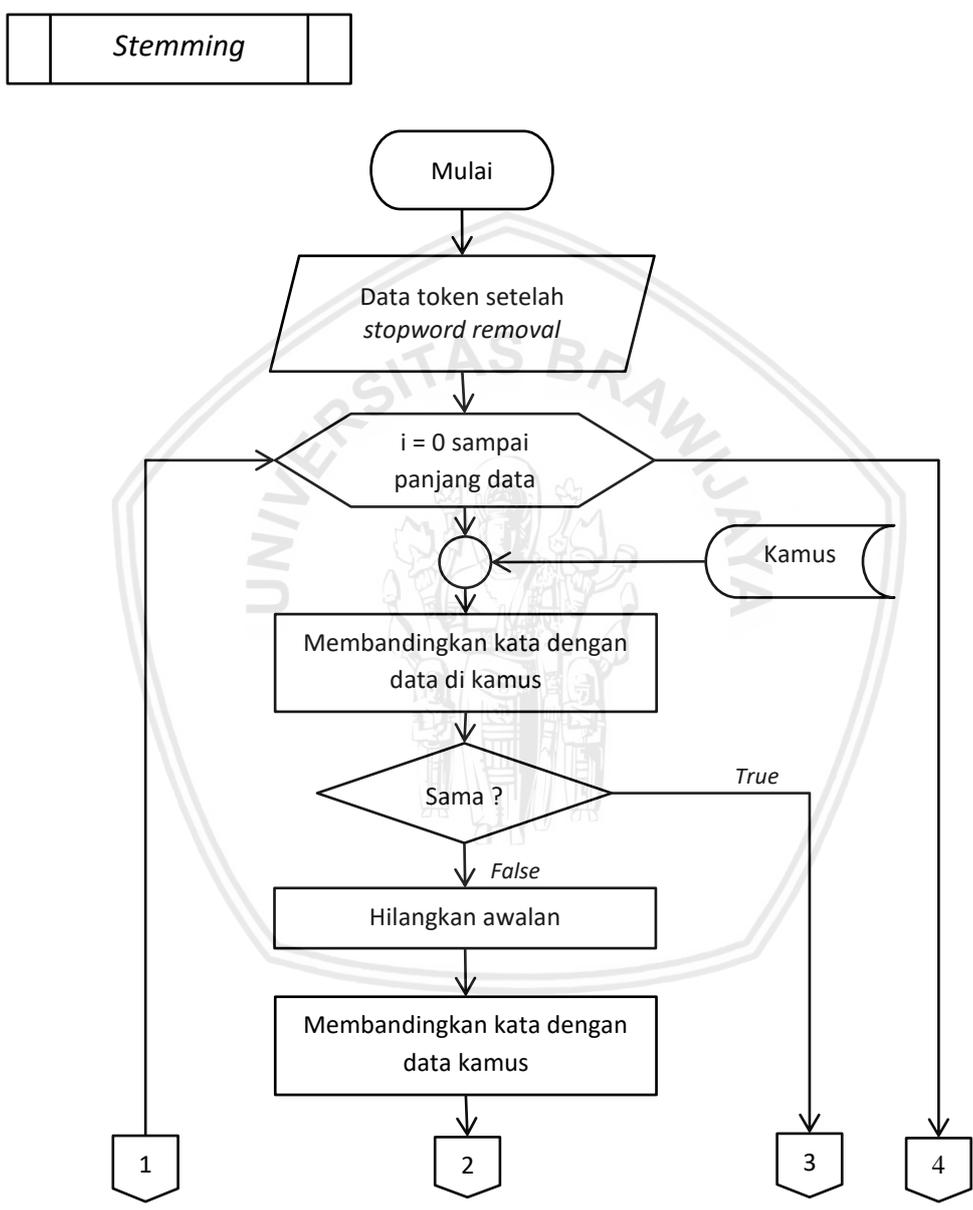
Gambar 4. 5 Flowchart Subproses Stopword Removal

Pada gambar 4.5, simbol *i* merupakan indeks dari setiap kata. Setiap kata dibandingkan dengan kata yang ada di *stopword list*, jika kata tidak ada di *stopword list* maka kata disimpan. Jika kata terdapat di *stopword list* maka kata harus dihapus. Proses tersebut terus berjalan sampai sebanyak jumlah kata telah dibandingkan seluruhnya.

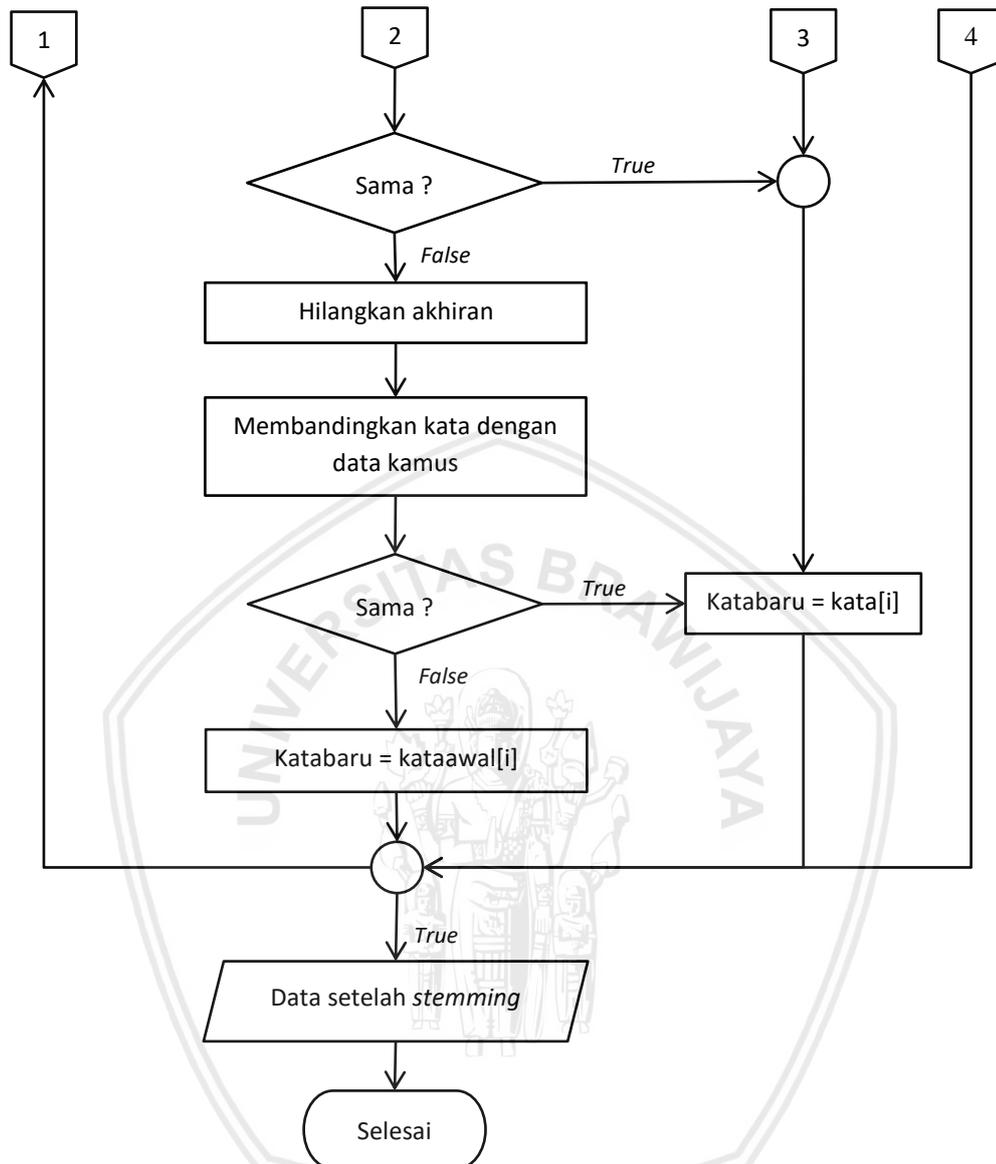
4. Stemming

Proses *stemming* adalah proses dimana setiap kata yang memiliki imbuhan di-awalan, di-akhiran, ataupun di-awalan dan di-akhiran harus dihapus.

Contoh kata yang memiliki imbuhan di-awalan dan di-akhiran adalah me-ragu-kan, 'me' adalah imbuhan di-awalan dan 'kan' adalah imbuhan di-akhiran dari kata dasar 'ragu'. Proses yang dilakukan dalam *stemming* dapat dilihat dari Gambar 4.6 dan Gambar 4.7.



Gambar 4. 6 Flowchart Subproses Stemming

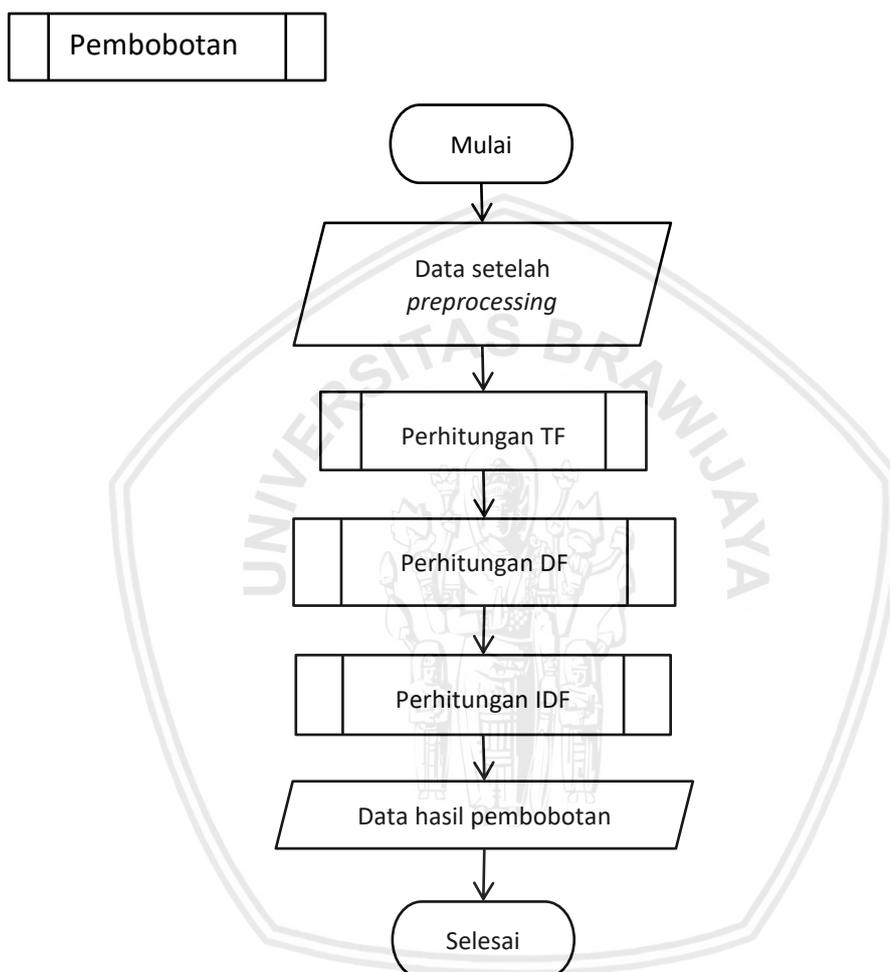


Gambar 4. 7 Flowchart Subproses Stemming (Lanjutan)

Pada gambar 4.6, setiap kata dibandingkan dengan data kamus yang berisi kata-kata dasar tanpa imbuhan. Jika kata sesuai dengan data kamus, maka kata disimpan sebagai kata baru. Jika kata tidak ada dikamus, maka hilangkan imbuhan awalan (me, mem, ke, be, ber, di, meng, dsb.) dari kata tersebut. Selanjutnya pada gambar 4.7, kata yang imbuhan awalnya telah dihilangkan dibandingkan lagi dengan data kamus. Jika kata sesuai dengan di kamus, maka kata disimpan sebagai katabaru. Jika tidak sama, hilangkan imbuhan akhiran (i, kan, dsb) dari kata tersebut. Selanjutnya kata dibandingkan kembali, jika kata sesuai dengan kamus maka kata disimpan sebagai katabaru. Jika kata tidak sesuai, maka kata kembali ke kata awal dan disimpan sebagai katabaru. Proses terus berjalan sampai indeks i sama dengan banyaknya kata.

4.2.2 Pembobotan

Setelah tahapan *preprocessing* selesai, selanjutnya adalah proses pembobotan. Pembobotan dilakukan untuk menghitung TF (*term frequency*), DF (*document frequency*), dan IDF (*invers document frequency*). Proses yang dilakukan dalam tahapan pembobotan dapat dilihat dari Gambar 4.8.



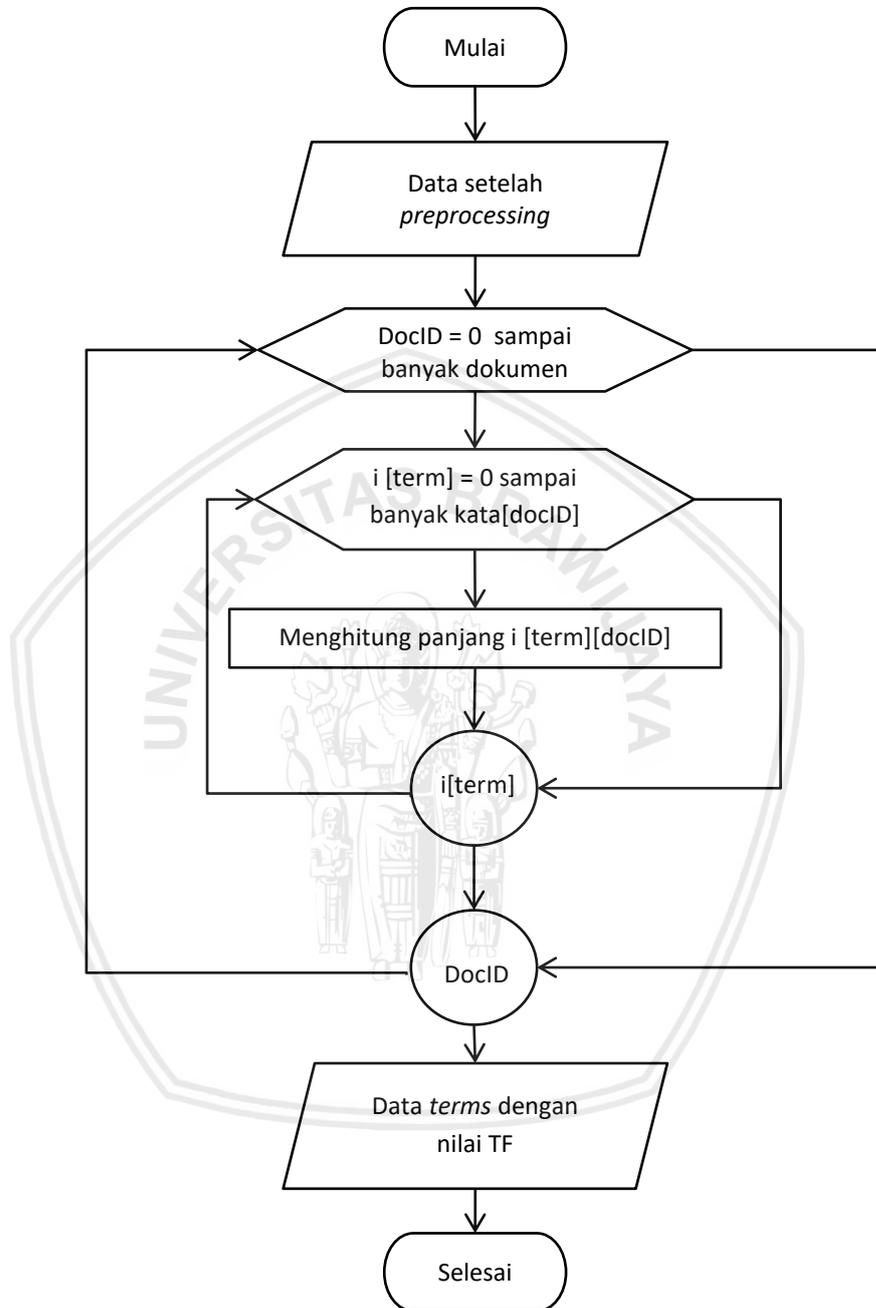
Gambar 4. 8 Flowchart Tahapan Pembobotan

Pada Gambar 4.8, menunjukkan rangkaian proses dari tahapan pembobotan secara lebih detail. Adapun rangkaian proses dari tahapan pembobotan yang digunakan dalam penelitian ini adalah proses perhitungan TF, perhitungan DF, dan perhitungan IDF. Pada tahap ini, data hasil *preprocessing* adalah masukan dan data hasil pembobotan adalah keluarannya.

1. Perhitungan TF

Nilai TF dihitung berdasarkan banyaknya kemunculan term pada setiap dokumen. Proses perhitungan TF dilihat pada Gambar 4.9.

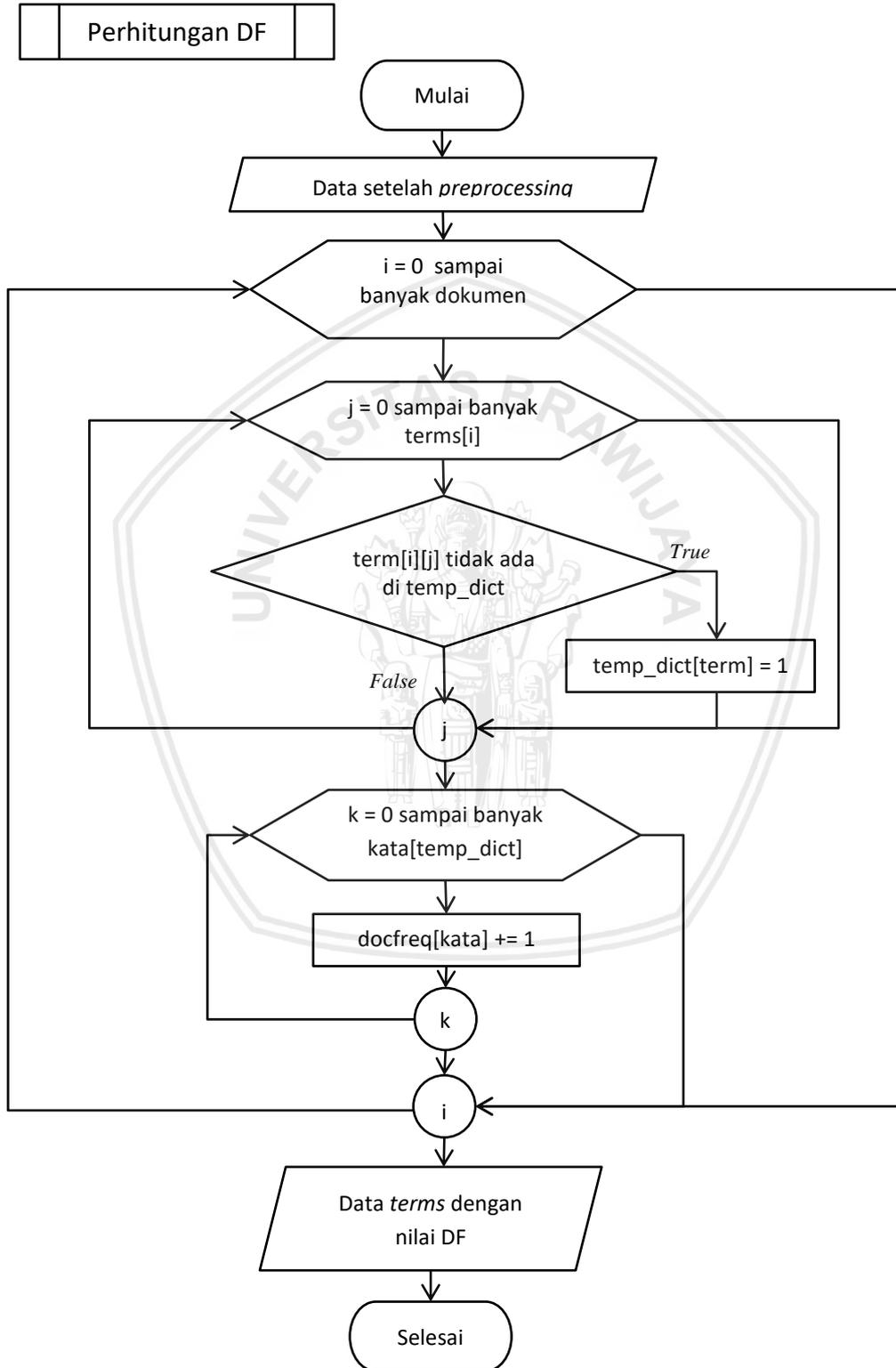
Perhitungan TF



Gambar 4. 9 Flowchart Perhitungan TF

Pada Gambar 4.9, data setelah *preprocessing* sebagai masukan untuk dilakukan perhitungan frekuensi term. Pada proses ini menggunakan perulangan *for* dengan *docID* sebagai indeks untuk banyaknya dokumen dan *i[term]* sebagai indeks untuk banyaknya term pada *docID*. Selanjutnya panjang *i* dari indeks *[term][docID]* dihitung untuk mendapatkan hasil dari nilai TF.

2. Perhitungan DF
 Nilai DF dihitung berdasarkan banyaknya dokumen yang terdapat term.
 Proses perhitungan DF dapat dilihat pada Gambar 4.10.

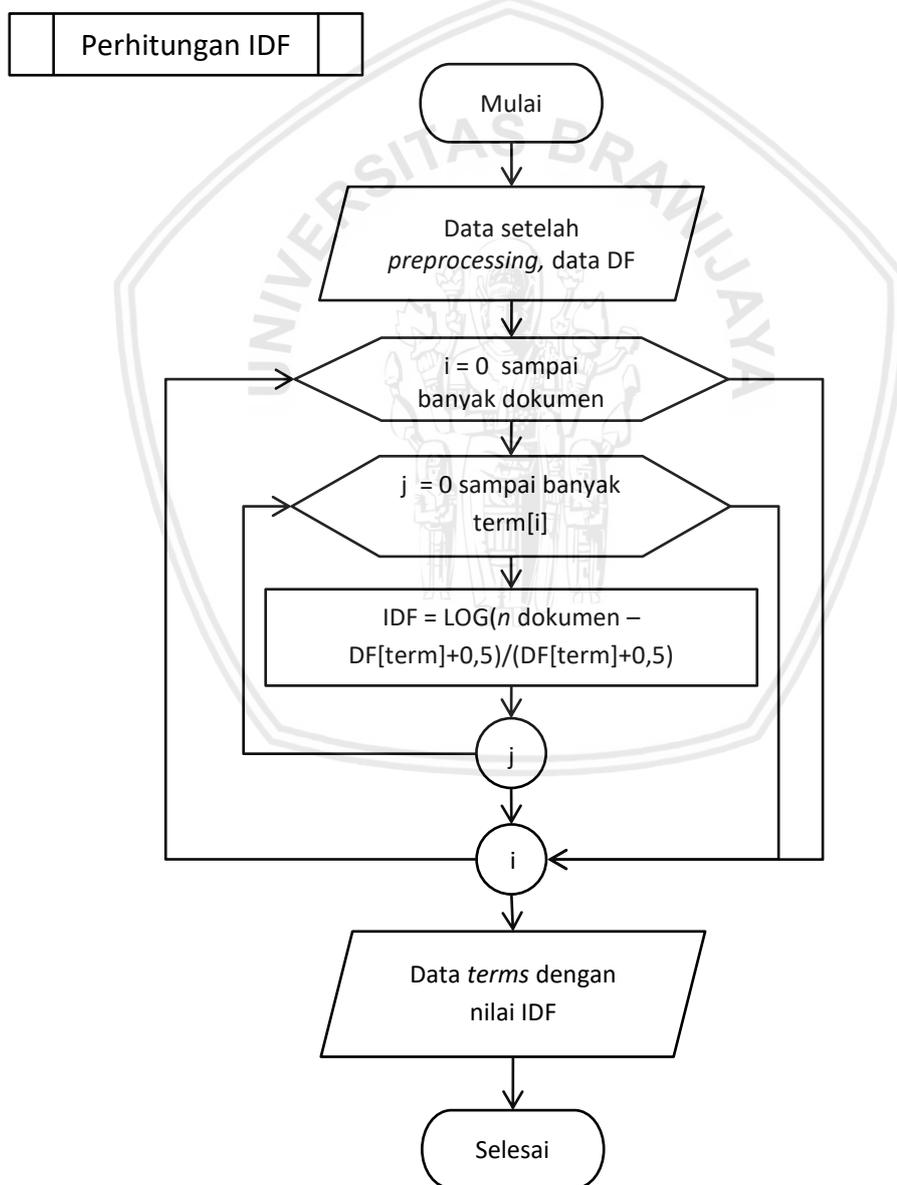


Gambar 4. 10 Flowchart Perhitungan DF

Pada Gambar 4.10, data setelah *preprocessing* sebagai masukan untuk dilakukan perhitungan frekuensi dokumen. Proses dilakukan menggunakan perulangan *for* dengan [i] adalah indeks dari dokumen dan [j] adalah indeks dari term. Jika term[i][j] tidak ada di kamus temp_dict, maka term diberi nilai 1. Proses terus berlanjut sampai sebanyak term pada dok[i]. Selanjutnya pada indeks [k], kata yang ada di kamus temp_dict setiap perulangan dok[i] ditambah 1, sehingga nilai DF akan diketahui.

3. Perhitungan IDF

Nilai IDF dihitung berdasarkan nilai invers dari banyaknya dokumen yang terdapat term. Proses perhitungan IDF dapat dilihat pada Gambar 4.11.



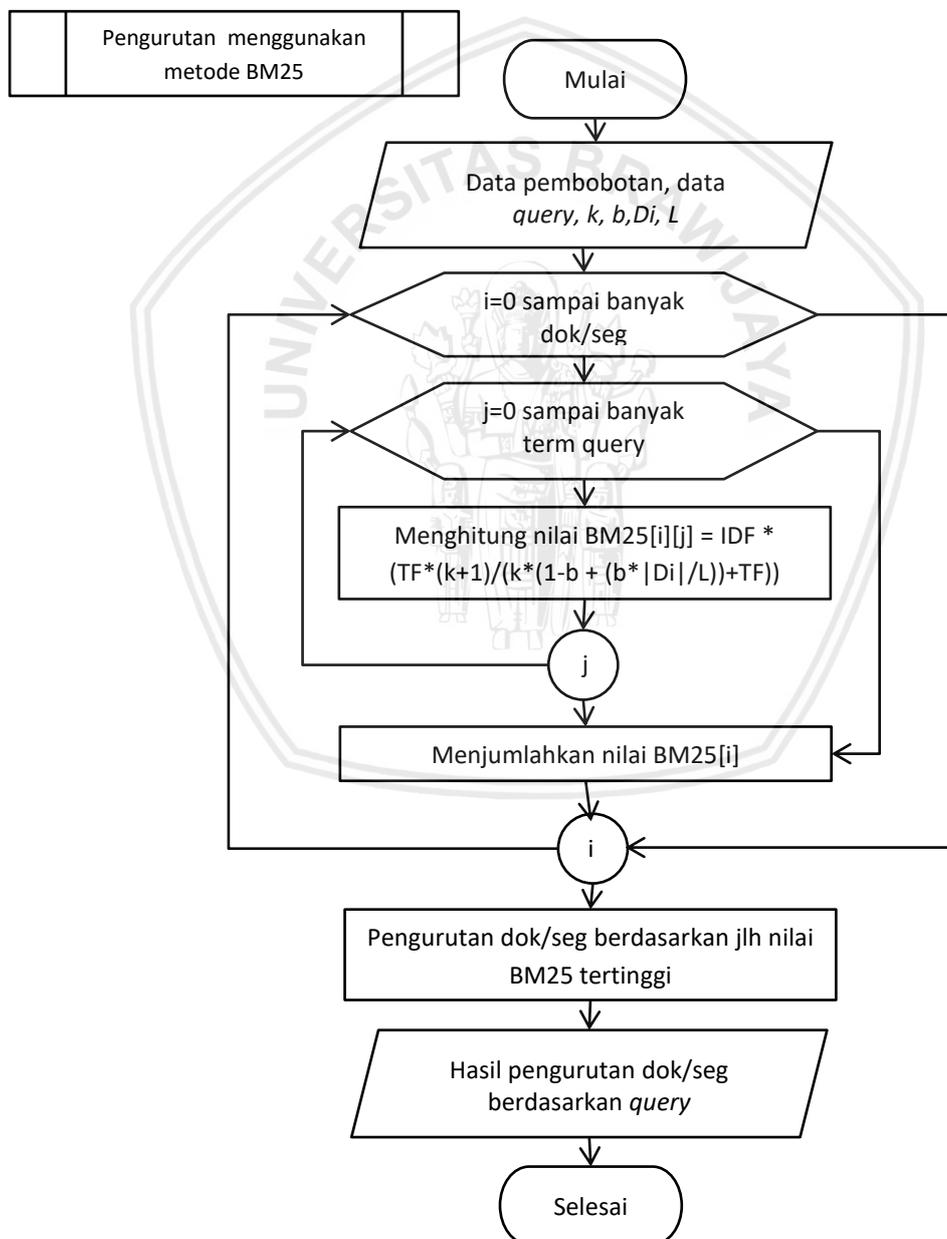
Gambar 4. 11 Flowchart Perhitungan IDF



Pada Gambar 4.11, data setelah *preprocessing* dan data nilai DF digunakan untuk proses perhitungan IDF. Proses dilakukan menggunakan perulangan *for* dengan [i] adalah indeks untuk dokumen dan [j] adalah indeks untuk term[i]. Setiap term[i][j] dilakukan perhitungan nilai IDF menggunakan persamaan 2.4.

4.2.3 Pengurutan Menggunakan Metode BM25

Tahapan yang dilakukan selanjutnya adalah mengurutkan dokumen berdasarkan *query* yang dimasukkan pengguna dengan menggunakan metode BM25. Proses pengurutan dapat dilihat pada Gambar 4.12.

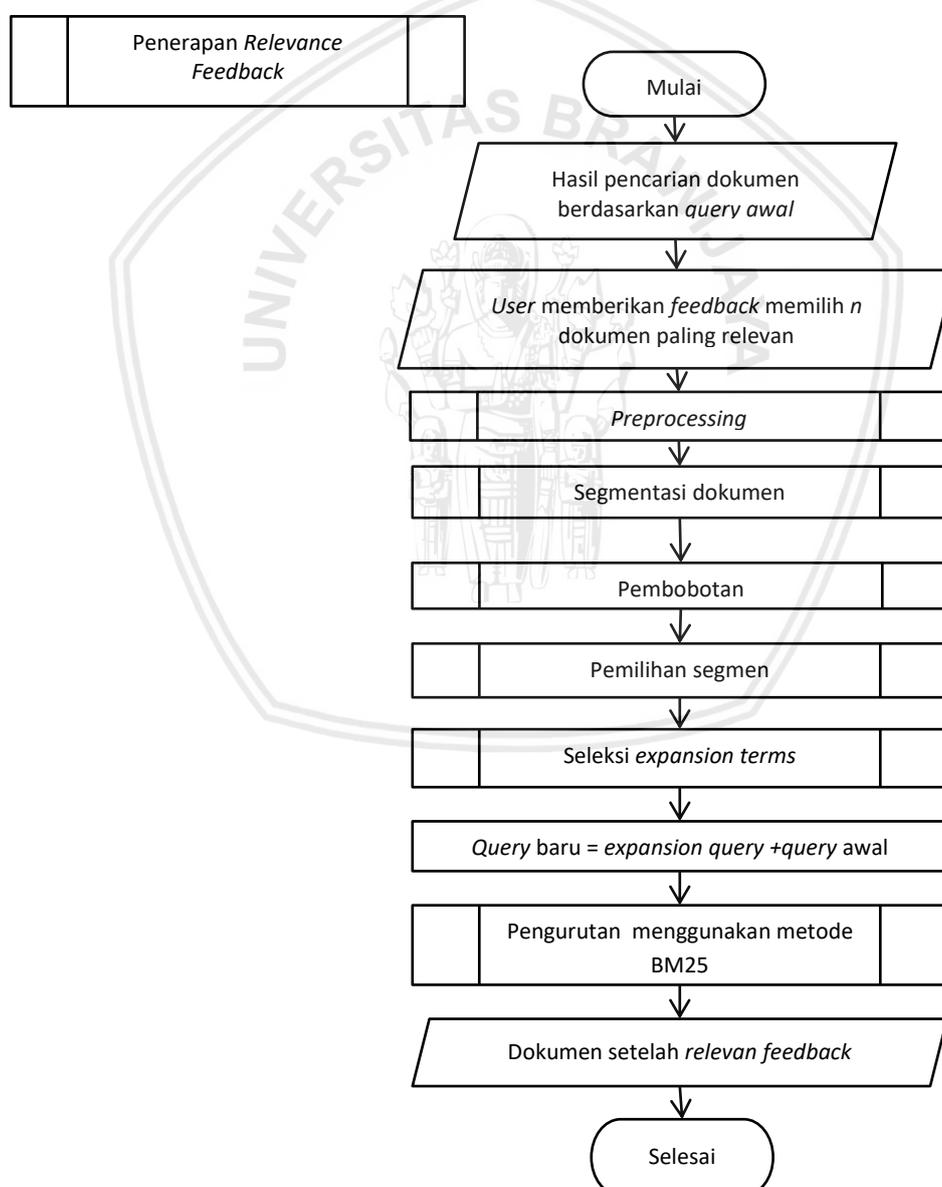


Gambar 4. 12 Flowchart Tahap Pengurutan menggunakan metode BM25

Pada Gambar 4.12, Untuk menghitung nilai BM25, dibutuhkan nilai TF dan IDF dari setiap *query term* yang dimasukkan. Dengan diketahui nilai k bernilai 1,2 dan nilai b bernilai 0,75. $|D_i|$ adalah panjang dokumen dari dokumen i . Selanjutnya untuk mencari nilai L dihitung dari jumlah semua panjang dokumen dibagi banyaknya dokumen. Nilai BM25 dihitung pada masing-masing dokumen berdasarkan *query term*, jika semua *term* sudah dihitung selanjutnya dijumlahkan seluruh nilai term pada setiap dokumen. Jumlah nilai BM25 dari setiap dokumen diurutkan dari nilai tertinggi.

4.2.4 Penerapan Relevance Feedback

Tahapan selanjutnya setelah pengurutan dokumen adalah menerapkan teknik *relevance feedback*. Proses pengurutan dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Flowchart Tahapan Relevance Feedback

Pada Gambar 4.13, Setelah sistem menampilkan hasil dari pencarian *query* awal, selanjutnya adalah pengguna memberikan umpanbalik (*feedback*) dengan memilih N dokumen yang paling relevan sesuai kebutuhan pengguna. Dokumen-dokumen yang terpilih ini akan diproses dengan setiap dokumen akan dibagi menjadi segmen-segmen. Pembagian segmen dilakukan setelah proses *preprocessing* dengan masing-masing segmen terdiri dari 50 kata. Kemudian setiap *term* pada masing-masing segmen dihitung nilai TF, DF, dan IDF.

Pemilihan segmen dilakukan dengan menghitung nilai BM25 berdasarkan *query* awal pada pencarian sebelumnya. Segmen dengan nilai tertinggi akan dipilih untuk diproses dalam penyeleksian perluasan *terms*. Setiap *term* pada segmen yang terpilih kemudian dihitung nilai TSV (*Term Selection Value*) dengan mengalikan nilai IDF dengan segmen yang terpilih dibagikan dengan jumlah seluruh segmen. Selanjutnya dipilih *top N term* dengan nilai TSV paling tinggi yang akan digunakan sebagai *query* baru. Pencarian dokumen selanjutnya berdasarkan *query* lama ditambah *query* baru dilakukan dengan menghitung kembali nilai BM25 dari semua *query* untuk kemudian didapatkan *ranking* dokumen yang paling relevan sesuai *query* baru.

1. *Preprocessing*

Dokumen-dokumen yang terpilih pada saat *relevance feedback* akan diproses dengan membagi setiap dokumen menjadi segmen-segmen. Sebelum dilakukan segmentasi, setiap dokumen diproses dahulu dengan menjalankan seluruh proses ditahapan *preprocessing*. Proses *preprocessing* dapat dilihat pada Gambar 4.2.

2. Segmentasi Dokumen

Untuk mendapatkan *expansion terms*, segmentasi dokumen dilakukan dalam mempercepat kinerja sistem dengan membagi dokumen menjadi segmen-segmen. Setelah setiap dokumen telah melakukan *preprocessing*, dokumen kemudian dibagi menjadi beberapa segmen dengan ketentuan setiap segmen terdiri dari 50 kata. Proses segmentasi dokumen dapat dilihat pada Gambar 4.15.

3. Pembobotan

Pembobotan segmentasi dilakukan untuk mencari nilai TF, DF, dan IDF pada term disetiap segmen. Nilai TF dihitung berdasarkan frekuensi term pada setiap segmen, nilai DF dihitung berdasarkan frekuensi segmen munculnya suatu term, dan nilai IDF dihitung berdasarkan nilai invers suatu term dari setiap frekuensi segmen. Proses pembobotan dapat dilihat pada Gambar 4.8.

4. Pemilihan Segmen

Pada Gambar 4.14, Pemilihan segmen dilakukan dengan memilih n segmen dengan urutan paling tinggi menggunakan perhitungan metode BM25 berdasarkan *query* awal. Proses pengurutan segmen menggunakan metode BM25 dapat dilihat pada Gambar 4.12.

5. Seleksi Perluasan *Terms*

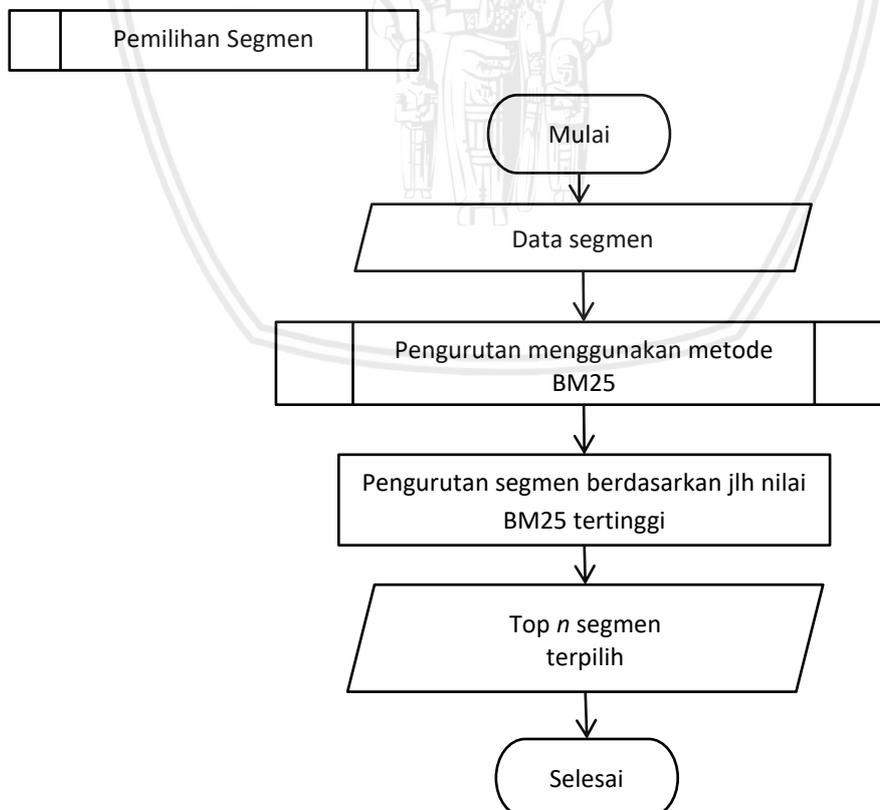
Pada Gambar 4.16, segmen yang telah terpilih kemudian dilakukan penyeleksian *term* untuk diambil sebagai *expansion terms*. Setiap *term* dihitung nilai TSV (*term selection value*). Nilai TSV dihitung berdasarkan nilai IDF pada masing-masing *term*. *Term* dengan nilai TSV paling tinggi akan digunakan sebagai perluasan *term* untuk diambil sebagai tambahan *query* baru. *Term* yang diambil adalah *term* yang berada diperingkat *top-n term*.

6. Menggabungkan *expansion terms* dengan *query* awal

Expansion terms yang sudah didapatkan kemudian digabungkan dengan *query* awal pada pencarian sebelumnya. *Query* baru selanjutnya digunakan dalam pencarian selanjutnya, dimana dengan menggunakan *query* baru ini dapat menghasilkan dokumen-dokumen yang paling relevan.

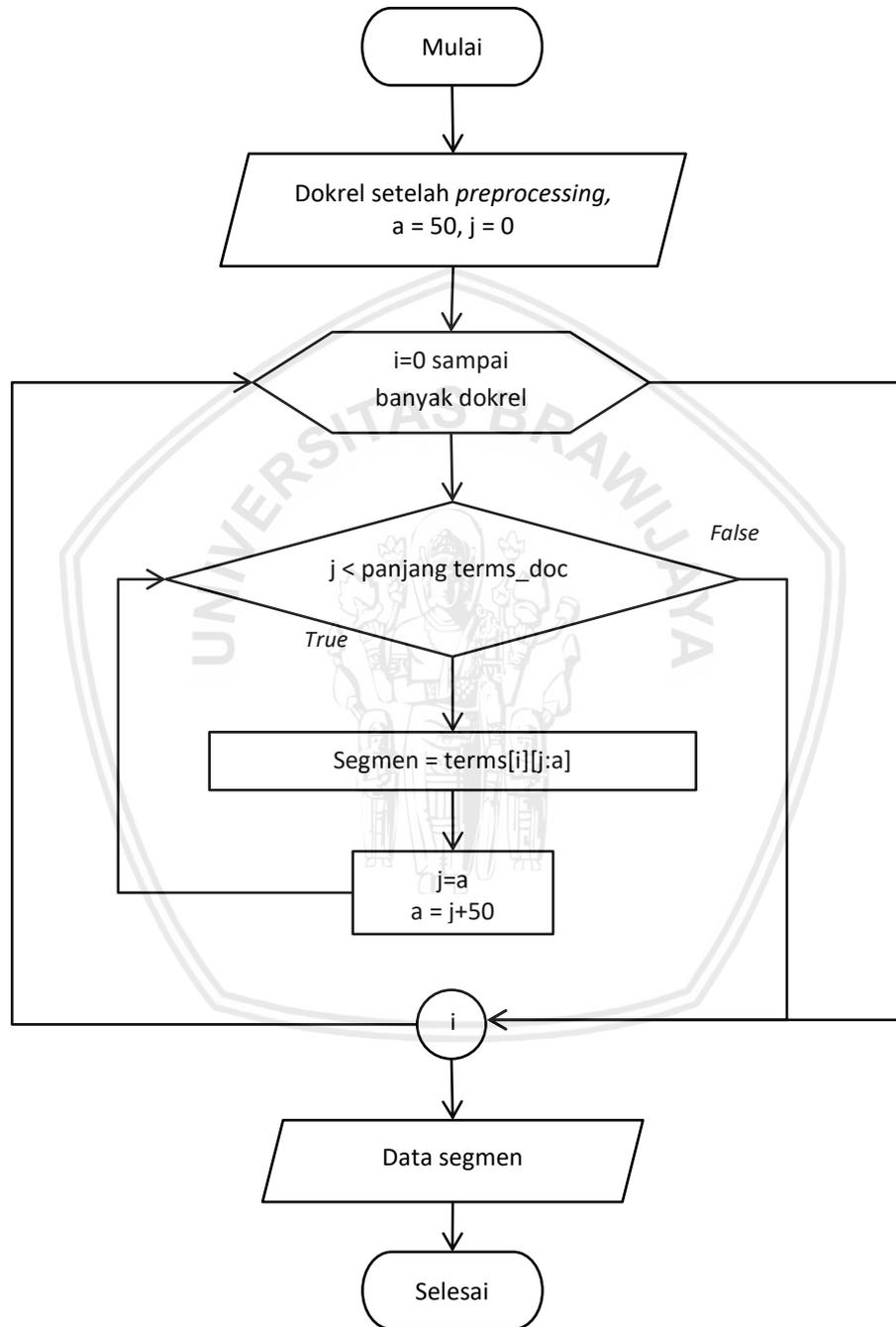
7. Pengurutan dokumen menggunakan metode BM25

Langkah selanjutnya adalah menghitung nilai BM25 untuk mendapatkan urutan dokumen dengan tingkat relevansi yang paling tinggi. Perhitungan nilai BM25 dilakukan berdasarkan *query* baru yang telah didapatkan sebelumnya. Dokumen-dokumen dengan nilai relevansi paling tinggi akan ditampilkan pada pencarian selanjutnya. Proses pengurutan dokumen menggunakan metode BM25 dapat dilihat pada Gambar 4.12.

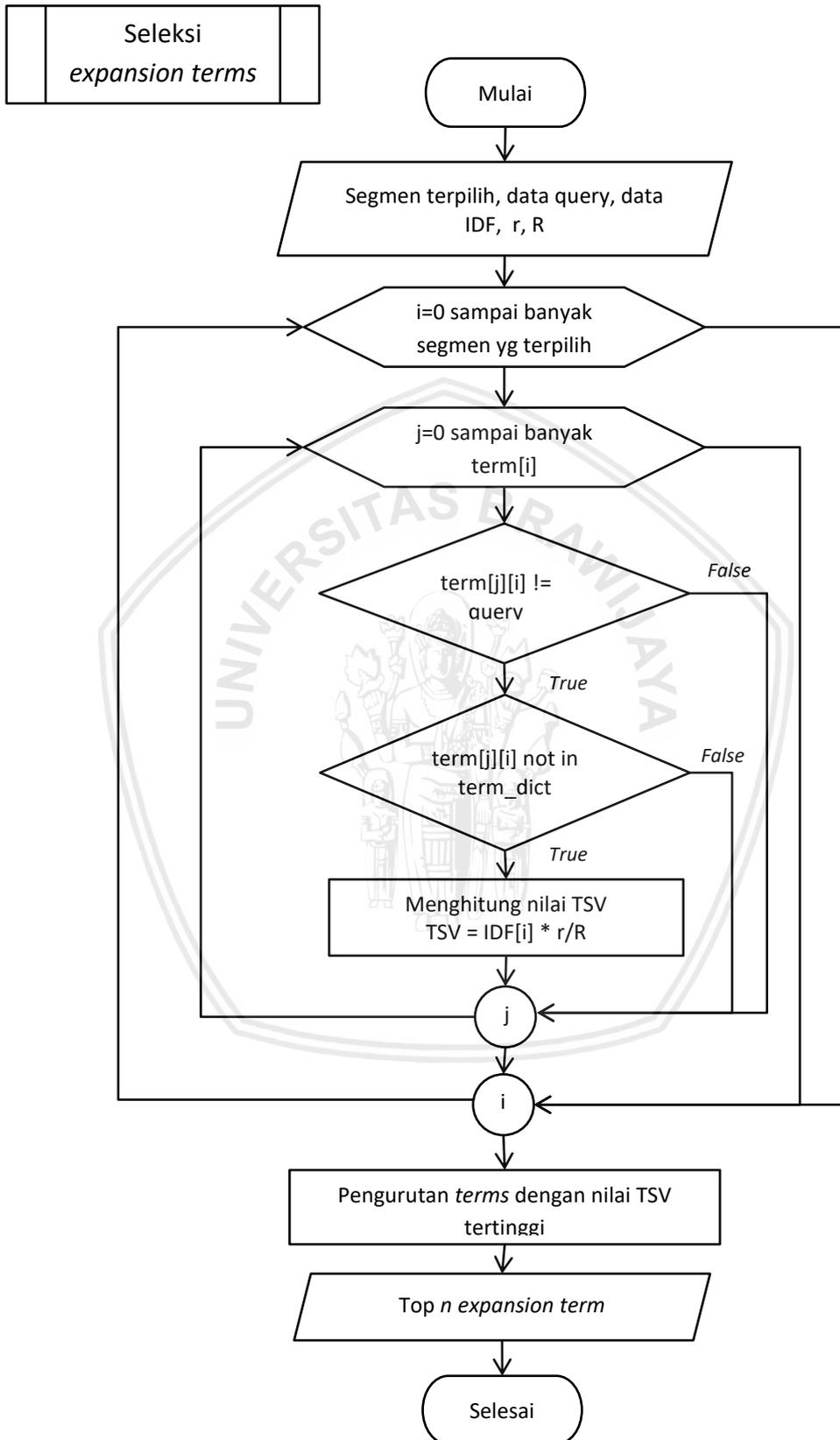


Gambar 4. 14 Flowchart Pemilihan segmen

| | | |
|--|-----------------------|--|
| | Segmentasi Dokumen | |
|--|-----------------------|--|



Gambar 4. 15 Flowchart Segmentasi Dokumen



Gambar 4. 16 Flowchart Seleksi Perluasan Term

4.3 Manualisasi

Manualisasi untuk sistem pencarian dokumen *e-book* dilakukan dengan menggunakan data latih berupa 5 dokumen terpilih yang berisi sinopsis dari masing-masing *e-book*. Selanjutnya yang menjadi data uji adalah *query* awal dan *query* baru, dengan ditentukan *query* awalnya adalah “Buku cara cerdas dan sukses”. *Query* baru akan ditentukan pada saat teknik *relevance feedback* diterapkan dalam penyeleksian *expansion terms*. Contoh data latih yang digunakan ditunjukkan pada Tabel 4.1.

Tabel 4. 1 Dataset

| Data | Isi Teks |
|------------|--|
| Dok1 | Seni Rupa Indonesia Dalam Perspektif Sejarah Buku Seni Rupa Indonesia dalam Perspektif Sejarah ini khusus dirancang untuk membantu kelancaran perkuliahan mahasiswa di Fakultas Seni Rupa dan Desain terutama jurusan yang ada mata kuliah Sejarah Seni Rupa Indonesia. <i>Dan seterusnya...</i> |
| Dok2 | Jayaning Majapahit: Kisah Para Kesatria Penjaga Samudra Inilah sepenggal kisah dari Kerajaan Majapahit. Setelah Gajah Mada mengucapkan Sumpah Palapa untuk menyatukan nusantara, salah satu langkah pertamanya adalah memperkuat armada laut. Seorang laksamana bernama Nala ia perintahkan untuk membangun kapal-kapal jung, memperkuat benteng istana, juga sungai dan laut. <i>Dan seterusnya...</i> |
| Dok3 | Pintar Tes IQ Untuk Anak 4-6 Tahun Orangtua sebagai guru yang pertama pasti menginginkan anak berhasil dalam pendidikannya. Dan kita sebagai orangtua tentu sangat mengetahui bahwa pendidikan sejak dini sangat penting untuk perkembangan otaknya. IQ yang berarti inteligensi adalah keterampilan berpikir dan kemampuan beradaptasi dan belajar dari pengalaman hidup sehari-hari. <i>Dan seterusnya...</i> |
| Dok4 | Kunci Sukses Meyakinkan & Memengaruhi Orang Lain dalam Berbagai Urusan Setiap manusia yang normal menginginkan kesuksesan dan kebahagiaan. Apakah Anda pernah memikirkan tentang kenyataan bahwa orang lain memainkan peranan penting dalam kesuksesan atau kebahagiaan Anda? Tujuan buku ini mengajarkan kepada Anda pengetahuan berdasarkan sifat dan tindakan manusia. <i>Dan seterusnya...</i> |
| Dok5 | Cerdik Berbicara Cerdas Menguasai Suasana "Kepiawaian bicara secara langsung mendukung kesuksesan kita di bidang apa pun. Dengan amat menarik, buku ini memaparkan taktik dan seni berbicara, yang akan memudahkan Anda memenangi perkara dan menguasai susana. <i>Dan seterusnya...</i> |
| Query Awal | Buku cara cerdas dan sukses |

4.3.1 Tahap *Preprocessing*

Pada tahap *preprocessing*, terdapat beberapa proses yang meliputi *case folding*, tokenisasi, *Stopword Removal*, dan *stemming*.

Pada Tabel 4.2 merupakan hasil dari proses *case folding*, yaitu mengubah semua huruf pada teks menjadi huruf kecil. Selanjutnya data dibersihkan dari karakter-karakter yang tidak penting seperti tanda baca, tag html, dsb. Tabel 4.2 menampilkan *query* awal dan 2 dokumen sebagai contoh hasil dari proses *case folding*.

Tabel 4. 2 Case Folding

| Data | Teks |
|-------------------|--|
| Dok1 | seni rupa indonesia dalam perspektif sejarah buku seni rupa indonesia dalam perspektif sejarah ini khusus dirancang untuk membantu kelancaran perkuliahan mahasiswa di fakultas seni rupa dan desain terutama jurusan yang ada mata kuliah sejarah seni rupa indonesia di samping itu buku ini hadir dalam rangka untuk membantu mahasiswa agar lebih memahami tentang seni rupa indonesia dalam perspektif sejarah yang meliputi masa prasejarah sampai masa hindu budha di indonesia dengan demikian buku ini diharapkan dapat menjadikan salah satu referensi bagi perkuliahan mahasiswa untuk mata kuliah sejarah seni rupa indonesia |
| Dok2 | jayaning majapahit kisah para kesatria penjaga samudra inilah sepenggal kisah dari kerajaan majapahit setelah gajah mada mengucapkan sumpah palapa untuk menyatukan nusantara salah satu langkah pertamanya adalah memperkuat armada laut seorang laksamana bernama nala ia perintahkan untuk membangun kapal kapal jung memperkuat benteng istana juga sungai dan laut cerita pertempuran raden wijaya dengan raja kediri yang dibantu panglima dari mongol memberi wawasan bagi nala untuk menjalankan strateginya para kesatria yang berperang demi mempertahankan tanah dan laut mereka juga dikisahkan dengan seru di dalam buku ini jayaning majapahit kejayaan di majapahit menceritakan kembali sejarah kebesaran majapahit yang dimulai dari pembangunan armada lautnya |
| <i>Query</i> Awal | buku cara cerdas dan sukses |

Tahap berikutnya adalah tokenisasi, setiap dokumen dipecah berdasarkan spasi sebagai pembatasnya, sehingga setiap dokumen akan berisi kata-kata tunggal. Pada tabel 4.3 menampilkan dokumen 1 dan 2, juga *query* awal sebagai contoh dalam proses tokenisasi.

Tabel 4. 3 Tokenisasi

| Data | Teks |
|------------|--|
| Dok1 | seni // rupa // indonesia // dalam // perspektif // sejarah // buku // seni // rupa // indonesia // dalam // perspektif // sejarah // ini // khusus // dirancang // untuk // membantu // kelancaran // perkuliahan // mahasiswa // di // fakultas // seni // rupa // dan // desain // terutama // jurusan // yang // ada // mata // kuliah // sejarah // seni // rupa // indonesia // di // samping // itu // buku // ini // hadir // dalam // rangka // untuk // membantu // mahasiswa // agar // lebih // memahami // tentang // seni // rupa // indonesia // dalam // perspektif // sejarah // yang // meliputi // masa // prasejarah // sampai // masa // hindu // budha // di // indonesia // dengan // demikian // buku // ini // diharapkan // dapat // menjadikan // salah // satu // referensi // bagi // perkuliahan // mahasiswa // untuk // mata // kuliah // sejarah // seni // rupa // indonesia |
| Dok2 | jayaning // majapahit // kisah // para // kesatria // penjaga // samudra // inilah // sepenggal // kisah // dari // kerajaan // majapahit // setelah // gajah // mada // mengucapkan // sumpah // palapa // untuk // menyatukan // nusantara // salah // satu // langkah // pertamanya // adalah // memperkuat // armada // laut // seorang // laksamana // bernama // nala // ia // perintahkan // untuk // membangun // kapal // kapal // jung // memperkuat // benteng // istana // juga // sungai // dan // laut // cerita // pertempuran // raden // wijaya // dengan // raja // kediri // yang // dibantu // panglima // dari // mongol // memberi // wawasan // bagi // nala // untuk // menjalankan // strateginya // para // kesatria // yang // berperang // demi // mempertahankan // tanah // dan // laut // mereka // juga // dikisahkan // dengan // seru // di // dalam // buku // ini // jayaning // majapahit // kejayaan // di // majapahit // menceritakan // kembali // sejarah // kebesaran // majapahit // yang // dimulai // dari // pembangunan // armada // lautnya |
| Query Awal | buku // cara // cerdas // dan // sukses |

Setelah data sudah dilakukan tokenisasi, kemudian dilakukan proses penghapusan *stopword*. Semua kata-kata yang tidak memiliki peran penting yang terdapat dalam *stopword list* dihilangkan. Tabel 4.4 merupakan hasil dari proses *stopword removal* dari contoh dokumen 1, dokumen 2, dan *query* awal.

Tabel 4. 4 Stopword Removal

| Data | Teks |
|------------|---|
| Dok1 | seni // rupa // indonesia // perspektif // sejarah // buku // seni // rupa // indonesia // perspektif // sejarah // khusus // dirancang // membantu // kelancaran // perkuliahan // mahasiswa // fakultas // seni // rupa // desain // terutama // jurusan // mata // kuliah // sejarah // seni // rupa // indonesia // samping // buku // hadir // rangka // membantu // mahasiswa // memahami // seni // rupa // indonesia // perspektif // sejarah // meliputi // masa // prasejarah // masa // hindu // budha // indonesia // buku // diharapkan // salah // satu // referensi // perkuliahan // mahasiswa // mata // kuliah // sejarah // seni // rupa // indonesia |
| Dok2 | jayaning // majapahit // kisah // kesatria // penjaga // samudra // sepenggal // kisah // kerajaan // majapahit // gajah // mada // mengucapkan // sumpah // palapa // menyatukan // nusantara // salah // satu // langkah // pertamanya // memperkuat // armada // laut // seorang // laksamana // bernama // nala // perintahkan // membangun // kapal // kapal // jung // memperkuat // benteng // istana // sungai // laut // cerita // pertempuran // raden // wijaya // raja // kediri // dibantu // panglima // mongol // memberi // wawasan // nala // menjalankan // strateginya // kesatria // berperang // mempertahankan // tanah // laut // dikisahkan // seru // buku // jayaning // majapahit // kejayaan // majapahit // menceritakan // sejarah // kebesaran // majapahit // dimulai // pembangunan // armada // lautnya |
| Query Awal | buku // cara // cerdas // sukses |

Langkah terakhir dalam tahapan *preprocessing* adalah *stemming*. *stemming* merupakan proses untuk menghilangkan imbuhan-imbuhan dari suatu kata dasar. Penelitian ini menggunakan *library stemming Sastrawi*. Pada Tabel 4.5 menunjukkan hasil proses *stemming* dari contoh dokumen 1, dokumen 2, dan *query* awal.

Tabel 4. 5 Stemming

| Data | Teks |
|------|---|
| Dok1 | seni // rupa // indonesia // perspektif // sejarah // buku // seni // rupa // indonesia // perspektif // sejarah // khusus // rancang // bantu // lancar // kuliah // mahasiswa // fakultas // seni // rupa // desain // utama // jurusan // mata // kuliah // sejarah // seni // rupa // indonesia // samping // buku // hadir // rangka // bantu // mahasiswa // paham // seni // rupa // indonesia // perspektif // sejarah // liput // masa // sejarah // masa // hindu // budha // indonesia // buku // harap // salah // satu // referensi // kuliah // mahasiswa // mata // kuliah // sejarah // seni // rupa // indonesia |

Tabel 4. 5 Stemming (Lanjutan)

| Data | Teks |
|------------|---|
| Dok2 | jayaning // majapahit // kisah // kesatria // jaga // samudra // penggal // kisah // raja // majapahit // gajah // mada // ucap // sumpah // palapa // satu // nusantara // salah // satu // langkah // pertama // kuat // armada // laut // orang // laksamana // nama // nala // perintah // bangun // kapal // kapal // jung // kuat // benteng // istana // sungai // laut // cerita // tempur // raden // wijaya // raja // kediri // bantu // panglima // mongol // beri // wawasan // nala // jalan // strategi // kesatria // perang // tahan // tanah // laut // kisah // seru // buku // jayaning // majapahit // jaya // majapahit // cerita // sejarah // besar // majapahit // mulai // bangun // armada // laut |
| Query Awal | buku // cara // cerdas // sukses |

4.3.2 Tahap Pembobotan

Pembobotan dilakukan untuk menghitung nilai TF, DF, dan, IDF. Pada Tabel 4.6 menampilkan nilai TF suatu kata dari setiap dokumen beserta nilai DF dengan mengambil 20 contoh data *term*.

Tabel 4. 6 TF dan DF

| NO | Kata | TF | | | | | DF |
|----|------------|----|----|----|----|----|----|
| | | D1 | D2 | D3 | D4 | D5 | |
| 1 | seni | 6 | 0 | 0 | 1 | 1 | 3 |
| 2 | rupa | 6 | 0 | 0 | 0 | 0 | 1 |
| 3 | indonesia | 6 | 0 | 0 | 0 | 0 | 1 |
| 4 | perspektif | 3 | 0 | 0 | 0 | 0 | 1 |
| 5 | sejarah | 6 | 1 | 0 | 0 | 0 | 2 |
| 6 | buku | 3 | 1 | 1 | 4 | 1 | 5 |
| 7 | khusus | 1 | 0 | 0 | 0 | 0 | 1 |
| 8 | rancang | 1 | 0 | 0 | 0 | 1 | 2 |
| 9 | bantu | 2 | 1 | 0 | 0 | 0 | 2 |
| 10 | lancar | 1 | 0 | 0 | 0 | 0 | 1 |
| 11 | cerdas | 0 | 0 | 1 | 0 | 1 | 2 |
| 12 | masuk | 0 | 0 | 1 | 0 | 0 | 1 |
| 13 | jenjang | 0 | 0 | 1 | 0 | 0 | 1 |
| 14 | tinggi | 0 | 0 | 1 | 0 | 0 | 1 |
| 15 | kunci | 0 | 0 | 0 | 1 | 0 | 1 |
| 16 | sukses | 0 | 0 | 0 | 3 | 1 | 2 |
| 17 | cara | 0 | 0 | 0 | 1 | 8 | 2 |
| 18 | milik | 0 | 0 | 0 | 1 | 0 | 1 |
| 19 | cerdik | 0 | 0 | 0 | 0 | 1 | 1 |
| 20 | bicara | 0 | 0 | 0 | 0 | 3 | 1 |

Selanjutnya untuk menghitung nilai IDF dapat menggunakan persamaan (2.4). Contoh perhitungan nilai IDF dari term “buku” dengan nilai DF diketahui adalah 5. Nilai IDF term “buku” menggunakan persamaan (2.4) adalah:

$$IDF = \log \frac{5 - 5 + 0.5}{5 + 0.5}$$

$$= -1,041392685$$

Pada Tabel 4.7 merupakan hasil manualisasi dari tahapan pembobotan pada setiap term.

Tabel 4. 7 Manualisasi Pembobotan Kata

| NO | Kata | TF | | | | | DF | IDF |
|----|------------|----|----|----|----|----|----|--------------|
| | | D1 | D2 | D3 | D4 | D5 | | |
| 1 | seni | 6 | 0 | 0 | 1 | 1 | 3 | -0,146128036 |
| 2 | rupa | 6 | 0 | 0 | 0 | 0 | 1 | 0,477121255 |
| 3 | indonesia | 6 | 0 | 0 | 0 | 0 | 1 | 0,477121255 |
| 4 | perspektif | 3 | 0 | 0 | 0 | 0 | 1 | 0,477121255 |
| 5 | sejarah | 6 | 1 | 0 | 0 | 0 | 2 | 0,146128036 |
| 6 | buku | 3 | 1 | 1 | 4 | 1 | 5 | -1,041392685 |
| 7 | khusus | 1 | 0 | 0 | 0 | 0 | 1 | 0,477121255 |
| 8 | rancang | 1 | 0 | 0 | 0 | 1 | 2 | 0,146128036 |
| 9 | bantu | 2 | 1 | 0 | 0 | 0 | 2 | 0,146128036 |
| 10 | lancar | 1 | 0 | 0 | 0 | 0 | 1 | 0,477121255 |
| 11 | cerdas | 0 | 0 | 1 | 0 | 1 | 2 | 0,146128036 |
| 12 | masuk | 0 | 0 | 1 | 0 | 0 | 1 | 0,477121255 |
| 13 | jenjang | 0 | 0 | 1 | 0 | 0 | 1 | 0,477121255 |
| 14 | tinggi | 0 | 0 | 1 | 0 | 0 | 1 | 0,477121255 |
| 15 | kunci | 0 | 0 | 0 | 1 | 0 | 1 | 0,477121255 |
| 16 | sukses | 0 | 0 | 0 | 3 | 1 | 2 | 0,146128036 |
| 17 | cara | 0 | 0 | 0 | 1 | 8 | 2 | 0,146128036 |
| 18 | milik | 0 | 0 | 0 | 1 | 0 | 1 | 0,477121255 |
| 19 | cerdik | 0 | 0 | 0 | 0 | 1 | 1 | 0,477121255 |
| 20 | bicara | 0 | 0 | 0 | 0 | 3 | 1 | 0,477121255 |

4.3.3 Penerapan metode BM25

Setelah didapatkan nilai TF dan IDF, kemudian melakukan perhitungan metode BM25 untuk meranking dokumen-dokumen yang relevan berdasarkan query awal yang dimasukkan. Perhitungan dilakukan dengan menggunakan persamaan (2.2). Adapun langkah-langkah yang harus dilakukan, yaitu:

- a. Menginisialisasi parameter $k = 1,2$ dan $b = 0,75$;
- b. Menghitung banyaknya kata pada setiap dokumen $|D_j|$
 $|D_1| = 61, |D_2| = 72, |D_3| = 54, |D_4| = 66, |D_5| = 56,$
- c. Menghitung panjang rata-rata (L) dari seluruh data dokumen,

$$L = \frac{61 + 72 + 54 + 66 + 56}{5}$$

$$L = 61,8$$

- d. Perhitungan *Retrieval status value (RSV)* berdasarkan *query* awal yaitu "Buku cara cerdas dan sukses" yang telah dilakukan *preprocessing* dan pembobotan seperti Tabel 4.8.

Tabel 4. 8 Hasil Pembobotan Query

| QUERY | TF | | | | | DF | IDF(qi) |
|--------|----|----|----|----|----|----|----------|
| | D1 | D2 | D3 | D4 | D5 | | |
| buku | 3 | 1 | 1 | 4 | 1 | 5 | -1,04139 |
| cara | 0 | 0 | 0 | 1 | 8 | 2 | 0,146128 |
| cerdas | 0 | 0 | 1 | 0 | 1 | 2 | 0,146128 |
| sukses | 0 | 0 | 0 | 3 | 1 | 2 | 0,146128 |

Perhitungan untuk masing-masing dokumen:

$$\begin{aligned}
 BM25(D_1) &= -1,04139 \cdot \frac{(3)(1,2+1)}{(3)+1,2(1-0,75+0,75\frac{61}{61,8})} + 0,146128 \cdot \frac{(0)(1,2+1)}{(0)+1,2(1-0,75+0,75\frac{61}{61,8})} \\
 &\quad + 0,146128 \cdot \frac{(0)(1,2+1)}{(0)+1,2(1-0,75+0,75\frac{61}{61,8})} + 0,146128 \cdot \frac{(0)(1,2+1)}{(0)+1,2(1-0,75+0,75\frac{61}{61,8})} \\
 &= -1,641026304
 \end{aligned}$$

$$BM25(D_2) = -0,975525351$$

$$BM25(D_3) = -0,944006373$$

$$BM25(D_4) = -1,373359225$$

$$BM25(D_5) = -0,496907436$$

Sehingga jika diurutkan berdasarkan nilai terbesar maka urutan dokumen dari dokumen paling relevan dimulai dari dok5, dok3, dok2, dok4, dan dok1.



4.3.4 Penerapan *Relevance Feedback*

Pencarian pertama dokumen berdasarkan *query* awal memberikan hasil berupa dokumen-dokumen relevan sesuai *query* masukkan pengguna. Dokumen-dokumen tersebut diurutkan berdasarkan tingkat relevansi paling tinggi menggunakan perhitungan metode BM25. Selanjutnya untuk menerapkan teknik *relevance feedback* dapat mengikuti langkah-langkah sebagai berikut.

1. Pengguna memilih n dokumen paling relevan dari hasil pencarian awal yang digunakan sebagai umpan balik (*feedback*).
Berdasarkan contoh hasil pada pencarian sebelumnya, dimisalkan pengguna memilih 2 dokumen yang dijadikan *feedback* yaitu **dok3** dan **dok5**.
2. Segmentasi dokumen
Langkah selanjutnya adalah membagi setiap dokumen menjadi segmen-segmen. Pada manualisasi, diambil contoh setiap segmen terdiri dari 20 kata. Jumlah segmen pada setiap dokumen tidak ditentukan karna berdasarkan jumlah kata dari dokumen tersebut. Tabel 4.10 merupakan hasil dari segmentasi setiap dokumen.

Tabel 4. 9 Hasil Segmentasi Dokumen

| Dok | Segmen | kata |
|------|--------|--|
| Dok3 | 1 | pintar tes iq anak tahun orangtua guru pertama anak hasil didik orang tua didik dini penting kembang otak iq |
| | 2 | arti inteligensi terampil pikir mampu adaptasi ajar alam hidup hari hari arti anak mampu alam mampu pecah masalah hidup hari |
| | 3 | hari buku susun orang tua tahu tingkat cerdas anak dini masuk jenjang didik tinggi |
| Dok5 | 1 | cerdik bicara cerdas kuasa suasana piawai bicara cara langsung dukung sukses bidang amat tarik buku papar taktik seni bicara mudah |
| | 2 | menang perkara kuasa suasana tahu cara jitu sampai kabar buruk cara bijak fakta keras cara lembut rumit cara sederhana berita |
| | 3 | panas cara dingin tolak berat cara ringan lewat cara rancang tangan gosip desas desus dusta yakin |

3. Pembobotan segmen
Pembobotan segmentasi dilakukan untuk menghitung nilai relevansi segmen berdasarkan *query* awal yaitu "Buku cara cerdas dan sukses" dengan menggunakan metode BM25. Tabel 4.11 menampilkan hasil pembobotan berupa nilai tf , df , dan idf dari *query* awal.

Tabel 4. 10 Pembobotan Query berdasarkan segmentasi

| No | Kata | dok | | | | | | df | idf |
|----|--------|--------|---|---|---|---|---|----|-------------|
| | | 3 | | | 5 | | | | |
| | | segmen | | | | | | | |
| | | 1 | 2 | 3 | 1 | 2 | 3 | | |
| 1 | buku | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 0,14612804 |
| 2 | cara | 0 | 0 | 0 | 1 | 4 | 3 | 3 | -0,14612804 |
| 3 | cerdas | 0 | 0 | 1 | 1 | 0 | 0 | 2 | 0,14612804 |
| 4 | sukses | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0,47712125 |

4. Pemilihan segmen

Pemilihan segmen dilakukan dengan memilih satu segmen yang berada paling tinggi urutannya berdasarkan hasil perhitungan nilai relevansi metode BM25 dengan *query* awal yaitu “Buku cara cerdas dan sukses”. Perhitungan dilakukan berdasarkan persamaan (2.2) dengan langkah-langkah sebagai berikut.

- a. Menginisialisasi parameter $k = 1,2$ dan $b = 0,75$;
- b. Menghitung banyaknya kata pada setiap segmen D_j

$$|Seg_1D_3| = 20, |Seg_2D_3| = 20, |Seg_3D_3| = 14, |Seg_1D_5| = 20, |Seg_2D_5| = 20, |Seg_3D_5| = 16$$

- c. Menghitung panjang rata-rata (L) dari seluruh data dokumen berdasarkan persamaan (2.3) sehingga diketahui nilai L adalah,

$$L = 18,33$$

- d. Perhitungan nilai relevansi dari setiap segmen berdasarkan *query* awal dengan pembobotan *query* pada Tabel 4.11 sebagai berikut.

$$BM25(Seg_1D_5) =$$

$$0,146 \cdot \frac{(1)(1,2+1)}{(1)+1,2(1-0,75+0,75\frac{20}{18,33})} + 0,146 \cdot \frac{(1)(1,2+1)}{(1)+1,2(1-0,75+0,75\frac{20}{18,33})}$$

$$+ 0,146 \cdot \frac{(1)(1,2+1)}{(1)+1,2(1-0,75+0,75\frac{20}{18,33})} + 0,477 \cdot \frac{(1)(1,2+1)}{(1)+1,2(1-0,75+0,75\frac{20}{18,33})}$$

$$= 0,6009$$

Nilai relevansi untuk semua segmen terdapat pada Tabel 4.12.



Tabel 4. 11 Nilai Relevansi Segmen dengan Query

| Segmen | Nilai Relevansi |
|-------------|-----------------|
| Dok5segmen1 | 0,600901659 |
| Dok3segmen3 | 0,323540553 |
| Dok3segmen1 | 0 |
| Dok3segmen2 | 0 |
| Dok5segmen5 | -0,23606798 |
| Dok5segmen2 | -0,243462881 |

Sehingga dok5segmen1 terpilih karena memiliki nilai relevansi paling tinggi.

5. Seleksi *expansion terms*
Berdasarkan persamaan (2.1) dilakukan seleksi *expansion terms* pada semua *terms* pada segmen yang terpilih kecuali *terms* yang sama dengan *terms* pada *query* awal.

Tabel 4. 12 Hasil nilai TSV

| DOK5SEG1 | TF | IDF | TSV |
|----------|----|---------|------------|
| cerdik | 1 | 0,47712 | 0,07952021 |
| bicara | 3 | 0,47712 | 0,07952021 |
| piawai | 1 | 0,47712 | 0,07952021 |
| langsung | 1 | 0,47712 | 0,07952021 |
| dukung | 1 | 0,47712 | 0,07952021 |
| bidang | 1 | 0,47712 | 0,07952021 |
| amat | 1 | 0,47712 | 0,07952021 |
| tarik | 1 | 0,47712 | 0,07952021 |
| papar | 1 | 0,47712 | 0,07952021 |
| taktik | 1 | 0,47712 | 0,07952021 |
| seni | 1 | 0,47712 | 0,07952021 |
| mudah | 1 | 0,47712 | 0,07952021 |
| kuasa | 1 | 0,14613 | 0,02435467 |
| suasana | 1 | 0,14613 | 0,02435467 |

Pada Tabel 4.13 diambil 5 peringkat teratas dari *terms* untuk digunakan sebagai *expansion query* yang digabungkan dengan *query* awal untuk membentuk suatu *query* baru yang akan digunakan pada pencarian selanjutnya. *Expansion query* yang diambil adalah “cerdik bicara piawai langsung dukung”.

6. Pengurutan dokumen menggunakan metode BM25

Pencarian selanjutnya dilakukan setelah menemukan *query* baru hasil dari *relevance feedback* dengan penyeleksian *expansion terms*. *Query* baru merupakan gabungan dari *query* awal dan *expansion query*, yaitu “buku cara cerdas sukses cerdas bicara piawai langsung dukung”.

Selanjutnya dilakukan kembali perhitungan metode BM25 menggunakan persamaan (2.2) untuk menentukan *ranking* dari setiap dokumen berdasarkan *query* baru. Tabel 4.14 merupakan hasil dari pengurutan dokumen berdasarkan *query* baru setelah melakukan *relevance feedback*.

Tabel 4. 13 Hasil Pengurutan Dokumen setelah *relevance feedback*

| RANKING | BM25 |
|---------|-------------|
| D5 | 2,252926715 |
| D3 | -0,94400637 |
| D2 | -0,97552535 |
| D4 | -1,37335922 |
| D1 | -1,6410263 |

Dari Tabel 4.14 didapatkan nilai relevansi perhitungan metode BM25 menggunakan *query* baru, sehingga hasil pencarian dokumen *e-book* berdasarkan *query* baru “buku cara cerdas sukses cerdas bicara piawai langsung dukung” menampilkan dokumen dengan urutan secara menurun diawali dengan dokumen **dok5**, **dok3**, **dok2**, **dok4**, dan **dok1**.

4.4 Perancangan Pengujian

Pengujian dilakukan menggunakan evaluasi temu kembali informasi berdasarkan $precision@K$ untuk mengukur kinerja sistem. Terdapat 2 jenis pengujian yang dilakukan yaitu, berdasarkan nilai K dan berdasarkan *expansion terms*.

4.4.1 Perancangan Pengujian berdasarkan nilai K

Pengujian ini dilakukan untuk mengetahui pengaruh pengambilan dokumen teratas dengan nilai K sebagai peringkat banyaknya dokumen teratas yang diambil. Percobaan dilakukan sebanyak 6 kali pada masing-masing data uji dengan nilai K adalah 10, 15, 20, 25, 30, dan 35. Pada pengujian ini, *expansion terms* yang ditambahkan sebanyak 1 kata dengan menggunakan 20 kata/segmen dan 50 kata/segmen.

Tabel 4. 14 Pengujian berdasarkan nilai K

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

4.4.2 Perancangan Pengujian berdasarkan *Expansion Terms*

Pengujian ini dilakukan untuk mengetahui pengaruh banyaknya penambahan kata (*term*) untuk perluasan *query*. *Term* dengan peringkat y teratas akan ditambahkan pada *query* baru. Percobaan dilakukan sebanyak 3 kali pada masing-masing data uji dengan y adalah 1, 2, dan 3 kata. Pada pengujian ini, dokumen yang diambil sebanyak 20 dokumen dengan menggunakan 20 kata/segmen dan 50 kata/segmen.

Tabel 4. 15 Pengujian berdasarkan *Expansion Terms*

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |

4.5 Perancangan Antarmuka

Sistem temu kembali informasi dokumen *e-book* dirancang dengan antarmuka berupa *Command Line Interface (CLI)*. *Interface* ini berinteraksi dengan sistem operasi dengan menggunakan *text terminal*. *Terminal* dalam peralatan khusus pada sebuah mesin komputer disebut sebagai *console*, yang berfungsi sebagai perantara komunikasi antar perangkat keras.

BAB 5 IMPLEMENTASI

Pada bab ini akan membahas mengenai implementasi sistem temu kembali informasi yang dibangun. Implementasi pada bab ini terdiri dari lingkungan implementasi yang terbagi atas lingkungan perangkat keras dan lingkungan perangkat lunak, selanjutnya terdapat batasan implementasi, implementasi algoritme, dan yang terakhir implementasi antarmuka.

5.1 Lingkungan Implementasi

Lingkungan implementasi menjelaskan tentang kebutuhan sistem dalam proses implementasi yang digunakan penulis pada penelitian ini. Lingkungan implementasi terdiri dari lingkungan perangkat keras (*hardware*) dan lingkungan perangkat lunak (*software*).

5.1.1 Lingkungan Perangkat Keras

Sistem yang dibuat adalah *Relevance Feedback* pada Sistem Temu Kembali Informasi Dokumen *E-book* Berbahasa Indonesia menggunakan Metode BM25. Dalam penerapannya, sistem ini menggunakan perangkat keras komputer dengan spesifikasi seperti berikut ini:

1. Intel® Core™ i5-7200U 2.5GHz
2. Memori 4GB DDR4
3. Harddisk kapasitas 1TB
4. Nvidia Geforce 940MX (2 GB DDR4)

5.1.2 Lingkungan Perangkat Lunak

Pengembangan *Relevance Feedback* pada Sistem Temu Kembali Informasi Dokumen *E-book* Berbahasa Indonesia menggunakan Metode BM25 memanfaatkan perangkat lunak seperti berikut ini:

1. Sistem operasi Windows 10 Pro 64-bit
2. Bahasa pemrograman Python
3. Alat pemrograman Spyder (Python 3.6)

5.2 Batasan Implementasi

Batasan implementasi merupakan batasan-batasan yang telah ditentukan dalam pembuatan sistem berdasarkan pada tahap perancangan sistem. Beberapa batasan-batasan dalam proses implementasi *Relevance Feedback* pada Sistem Temu Kembali Informasi Dokumen *E-book* Berbahasa Indonesia menggunakan Metode BM25 adalah sebagai berikut:

1. Dokumen yang digunakan dalam penelitian ini adalah dokumen sinopsis *e-book* berbahasa Indonesia.
2. Dokumen *e-book* yang digunakan dalam penelitian ini berjumlah 260 dokumen yang berasal dari aplikasi *Google Play Books*.
3. Mekanisme *relevance feedback* yang digunakan adalah *manual relevance feedback*.

5.3 Implementasi Algoritme

Pada implementasi algoritme terdapat beberapa proses yang dimulai dari *preprocessing*, pembobotan, pengurutan metode BM25, dan penerapan *relevance feedback*.

Data yang digunakan sebanyak 260 dokumen sinopsis *e-book* yang dikumpulkan dari *Google Play* dengan kategori buku. Kode Program 5.1 adalah *source code* untuk memproses semua dokumen agar dapat dilakukan tahap *preprocessing*.

| Algoritma 1: Penyiapan data latih | |
|-----------------------------------|---|
| 1 | def openfile(file): |
| 2 | isi = [] |
| 3 | for i in file: |
| 4 | f = open(docs_input + '\\\ ' + i, 'r+') |
| 5 | data=f.read() |
| 6 | isi.append(data) |
| 7 | return isi |

Kode Program 5. 1 Source Code Penyiapan data latih

5.3.1 Preprocessing

Pada Kode Program 5.2 merupakan *source code* tahap tokenisasi untuk memecah isi dokumen menjadi token dengan menghilangkan karakter yang tidak penting dan mengubah huruf kapital menjadi huruf kecil.

| Algoritma 2: Proses Tokenisasi | |
|--------------------------------|--|
| 1 | def tokenisasi(file): |
| 2 | hasil = [] |
| 3 | for i in file: |
| 4 | data = i.lower() |
| 5 | clean = re.findall(r'\b[a-z]{4,15}\b', data) |
| 6 | hasil.append(clean) |
| 7 | return hasil |

Kode Program 5. 2 Source code Tokenisasi

Setelah dilakukan tokenisasi, kemudian dilakukan tahapan *stopword removal* yaitu menghapus *stopword*. Pada Kode Program 5.3 merupakan *source code* untuk menghilangkan *stopword* atau kata-kata yang tidak memiliki peran yang terdapat di *stopword list Sastrawi*.

| Algoritma 3: Proses Stopword Removal | |
|--------------------------------------|----------------------------------|
| 1 | def stopremoval(file): |
| 2 | konten = [] |
| 3 | for word in file: |
| 4 | str1 = ' '.join(word) |
| 5 | filter1 = stopwords.remove(str1) |
| 6 | data = filter1.split(' ') |
| 7 | konten.append(data) |
| 8 | # print(konten) |
| 9 | return konten |

Kode Program 5. 3 Source Code Penghapusan Stopword

Langkah terakhir dalam tahap *preprocessing* adalah *stemming*, yaitu proses untuk menghilangkan imbuhan-imbuhan dari suatu kata dasar. Proses

stemming menggunakan *library stemmer Sastrawi*. *Source code* untuk proses *stemming* terdapat pada Kode Program 5.4.

| Algoritma 4: Proses Stemming | |
|------------------------------|--------------------------------|
| 1 | def stemming(file): |
| 2 | tess = [] |
| 3 | for word in file: |
| 4 | str2 = ' '.join(word) |
| 5 | katadasar = stemmer.stem(str2) |
| 6 | data = katadasar.split(' ') |
| 7 | tess.append(data) |
| 8 | return tess |

Kode Program 5. 4 Source Code Proses Stemming

5.3.2 Pembobotan

Pada tahap pembobotan terdapat 3 proses, yaitu mencari nilai *Term Frequency (TF)*, *Document Frequency (DF)*, dan *Invers Document Frequency (IDF)*. Kode Program 5.5 merupakan *source code* untuk menghitung *term frequency (TF)* yaitu jumlah kemunculan *term* pada setiap dokumen id.

| Algoritma 5: Perhitungan TF | |
|-----------------------------|--|
| 1 | def term_frequency(file,word,tfdict): |
| 2 | for docID in file: |
| 3 | TF = defaultdict(list) |
| 4 | for i, term in enumerate(file[docID]): |
| 5 | TF[term].append(i) |
| 6 | a = len(TF[word]) |
| 7 | # print(TF) |
| 8 | print(docID+1,TF[word],a,word) |
| 9 | tfdict[word].append(a) |
| 10 | return tfdict |

Kode Program 5. 5 Source code Term Frequency (TF)

Selanjutnya pada Kode Program 5.6 merupakan proses untuk menghitung *DF* yaitu *Document Frequency*.

| Algoritma 6: Perhitungan DF | |
|-----------------------------|------------------------------|
| 1 | def docfreq(docs, freqdict): |
| 2 | for tokens in docs: |
| 3 | temp_dict = defaultdict(int) |
| 4 | for tok in docs[tokens]: |
| 5 | if tok not in temp_dict: |
| 6 | temp_dict[tok] = 1 |
| 7 | for word in temp_dict: |
| 8 | freqdict[word] += 1 |
| 9 | # print(freq_dict) |
| 10 | return freqdict |

Kode Program 5. 6 Source code Document Frequency (DF)

Untuk proses selanjutnya yaitu penghitungan nilai *Invers Document Frequency (IDF)* dengan *source code* dapat dilihat pada Kode Program 5.7.

Algoritma 7: Perhitungan IDF

```

1 def invers_df(file, doc_freqs, word, idfdict):
2     b = []
3     for docID in file:
4         IDF = defaultdict(list)
5         for i, term in enumerate(file[docID]):
6             idf = math.log10(((len(file))-
7                 doc_freqs[term]+0.5) /
8                 (doc_freqs[term]+0.5))
9             # print(idf, term)
10            IDF[term].append(idf)
11            if word == term:
12                a = IDF[word]
13                if a == []:
14                    del a
15                else: b.append(a)
16            else: a = 0
17        if b == []:
18            b = [[0]]
19            c = b[0]
20        else: c = b[0]
21        idfdict[word].append(c[0])
22        # print(word, idfdict[word])
23    return idfdict[word][0]

```

Kode Program 5. 7 Source Code untuk *Invers Document Frequency (IDF)*

5.3.3 Pengurutan Menggunakan Metode BM25

Setelah nilai TF, DF, dan IDF didapatkan, selanjutnya adalah menghitung nilai BM25 untuk mencari dokumen-dokumen atau segmen-segmen yang relevan berdasarkan urutan nilai BM25 paling tinggi. Pada Kode Program 5.8 merupakan *source code* untuk menghitung nilai BM25.

Algoritma 8: Pengurutan menggunakan metode BM25

```

1 def BM25(docid, tf, idf, doclength, docs, queries, rsv):
2     k = 1.2
3     b = 0.75
4     j1Dj = sum(doc_length)
5     L = j1Dj/docid
6     j1_BM25 = []
7
8     for docID in docs:
9         BM25 = defaultdict(list)
10        for i in queries:
11            inv = idf[i]
12            TF = float(tf[i][docID])
13            IDF = float(inv[0])
14            DL = (doclength[docID])
15            bm25 = IDF* ((TF) * (k+1) / (k* (1-
16                b+ (b*DL/L) )+TF) )
17            BM25[i].append(bm25)
18        # print(docID+1, i, BM25[i])
19        j1_BM25.append(BM25[i][0])
20        a = sum(j1_BM25)
21        # print("Jumlah: ", a)
22        j1_BM25.clear()
23    rsv[docID+1].append(a)

```

Kode Program 5. 8 Source Code Pengurutan dokumen

5.3.4 Penerapan *Relevance Feedback*

Terdapat dua jenis teknik *relevance feedback* berdasarkan pemberian *feedback*nya. Ada *manual relevance feedback* yaitu *feedback* diberikan langsung oleh *user* dan ada *pseudo relevance feedback* yaitu *feedback* diambil dari K urutan tertinggi dokumen yang relevan dengan K ditentukan terlebih dahulu. Pada Kode Program 5.9 merupakan *source code* untuk *manual relevance feedback* karena pada penelitian ini menggunakan *manual relevance feedback*.

| Algoritma 9: Proses <i>Relevance Feedback</i> | |
|---|---------------------------------------|
| 1 | rf = input("User feedback [docid]: ") |
| 2 | a = rf.split(',') |
| 3 | for i in a: |
| 4 | x = i |
| 5 | docid = int(x) |
| 6 | for j, docid in enumerate(doc_rank): |
| 7 | if docid == docid: |
| 8 | docid |
| 9 | for f, d in enumerate(doc_id): |
| 10 | if f+1 == docid: |
| 11 | docid |
| 12 | print(docid, doc_id[f]) |
| 13 | for q, w in enumerate(token_doc): |
| 14 | if q+1 == docid: |
| 15 | docid |
| 16 | print(docid, token_doc[q]) |
| 17 | print('=====') |

Kode Program 5. 9 Source Code untuk Manual Relevance Feedback

Setelah pemberian *feedback*, setiap dokumen yang terpilih kemudian dilakukan proses segmentasi. Proses ini dilakukan dengan membagi setiap dokumen menjadi segmen-segmen. setiap segmen memiliki maksimal 50 *term*. Kode Program 5.10 merupakan *source code* untuk proses segmentasi.

| Algoritma 10: Proses segmentasi dokumen | |
|---|-------------------------------|
| 1 | e = len(token_doc[q]) |
| 2 | i = 0 |
| 3 | a = 50 |
| 4 | while i < e: |
| 5 | seg = token_doc[q][i:a] |
| 6 | seglist.append(seg) |
| 7 | print(seg) |
| 8 | segmen_doc[docid].append(seg) |
| 9 | i = a |
| 10 | a = i+50 |

Kode Program 5. 10 Source Code untuk Segmentasi

Kemudian setiap segmen dilakukan pembobotan berdasarkan *query*. Pada Kode Program 5.11 merupakan *source code* untuk proses penghitungan TF, DF, dan IDF.

| Algoritma 11: Proses pembobotan segmen | |
|--|---|
| 1 | for q,i in enumerate(seglist): |
| 2 | j1 = q+1 |
| 3 | segmen_dict[q] = i |
| 4 | segf = index1.docfreq(segmen_dict,sf_dict) |
| 5 | seg1 = index1.doclength(seglist,seg_length) |
| 6 | index1.docid(seglist,seg_id) |
| 7 | sid = index1.doclist(seg_id) |
| 8 | seg_query = |
| 9 | index1.query(queries,segmen_dict, |
| 10 | sf dict,tfs dict,idfs dict) |

Kode Program 5. 11 Source Code untuk Pembobotan Segmen

Selanjutnya pemilihan segmen dilakukan dengan menghitung nilai BM25. Segmen dengan urutan 1 teratas akan dipilih untuk proses selanjutnya. Kode Program 5.12 merupakan *source code* untuk pemilihan segmen.

| Algoritma 12: Proses pemilihan segmen | |
|---------------------------------------|--|
| 1 | index1.BM25(sid,tfs_dict,idfs_dict,seg_length,seg_id,seg_query,RSVs) |
| 2 | segsort_RSV = sorted(RSVs.items(), |
| 3 | key=operator.itemgetter(1),reverse=True) |
| 4 | print("Jumlah Segmen: ",len(segsort_RSV)) |
| 5 | segm = 1 |
| 6 | for (segid,rsv) in segsort_RSV[:segm]: |
| 7 | print("SEGID: ",segid,"==> RSV: ",rsv) |
| 8 | seg_rank[segid].append(rsv) |
| 9 | seg_rank[segid].append(rsv) |

Kode Program 5. 12 Source Code untuk Pemilihan segmen

Setelah segmen terpilih, langkah selanjutnya adalah menyeleksi perluasan *term* pada semua *term* yang terdapat pada segmen yang terpilih dan *term* terpilih akan ditambahkan menjadi *query* baru. Penyeleksian dilakukan dengan menghitung nilai *term status value (TSV)* berdasarkan persamaan (2.1). Kode Program 5.13 merupakan *source code* untuk penyeleksian *term*.

| Algoritma 13: Proses perluasan term | |
|-------------------------------------|--|
| 1 | for f, d in enumerate(seg_id): |
| 2 | if f+1 == segid: |
| 3 | qelist.append(seg_id[f]) |
| 4 | dict1 = defaultdict(list) |
| 5 | for i in qelist: |
| 6 | for j in i: |
| 7 | if j not in dict1: |
| 8 | idf = index1.invers_df(segmen_dict, |
| 9 | sf_dict,j,dict1) |
| 10 | IDF = float(idf) |
| 11 | tsv = IDF*1/segm |
| 12 | TSV[j].append(tsv) |
| 13 | # print("TSV: ",tsv) |
| 14 | TSV_sort = sorted(TSV.items(), |
| 15 | key=operator.itemgetter(1),reverse=True) |
| 16 | for (query,tsv) in TSV_sort[:1]: |
| 17 | # PENAMBAHAN QUERY BARU |
| 18 | print("Query baru: ",query,tsv) |
| 19 | queries[0].append(query) |
| 20 | print('====QUERY EXPANSION====') |
| 21 | print(queries) |

Kode Program 5. 13 Source Code untuk Penyeleksian Term

Selanjutnya dengan menggunakan *query* baru, dilakukan kembali pencarian dokumen berdasarkan urutan tertinggi dari nilai BM25. Pada Kode Program 5.14 merupakan *source code* untuk menentukan *ranking* dokumen berdasarkan relevansinya terhadap *query*.

| | |
|--|--|
| Algoritma 14: Penentuan peringkat dokumen setelah RF | |
| 1 | qnew = |
| 2 | indexl.query(queries, token_doc, df_dict, tfd_dict, idfd_dict) |
| 3 | indexl.BM25(ab, tfd_dict, idfd_dict, |
| 4 | doc_length, doc_id, qnew, RSVd) |
| 5 | sort_RSV = sorted(RSVd.items(), |
| 6 | key=operator.itemgetter(1), reverse=True) |
| 7 | for (docid, rsv) in sort_RSV[:int(k)]: |
| 8 | print("DOCID: ", docid, "=> RSV: ", rsv) |
| 9 | doc_rank[docid].append(rsv) |

Kode Program 5. 14 Source Code untuk Penentuan peringkat dokumen setelah *relevance feedback*

5.4 Implementasi Antarmuka

Tampilan antarmuka sistem menggunakan *console line interface*. Pada Gambar 5.15 dapat dilihat bahwa percobaan sistem dilakukan dengan memberikan *query* : buku motivasi. Nilai K yang diambil sebanyak 10 dokumen teratas dan penambahan *query* baru sebanyak 1 kata.

```

Query: pelajaran sekolah
@k dokumen: 10
=====
DOCID: 46 RSV: [2.228677977955068]
DOCID: 152 RSV: [2.2114235333814176]
DOCID: 250 RSV: [2.1544712995891535]
DOCID: 154 RSV: [1.9418679178099223]
DOCID: 126 RSV: [1.8648770364084157]
DOCID: 13 RSV: [1.8624100893991187]
DOCID: 98 RSV: [1.8624100893991187]
DOCID: 176 RSV: [1.7638091406662217]
DOCID: 232 RSV: [1.6244836634276352]
DOCID: 231 RSV: [1.611262393957078]
Hasil Dokumen : 10
Dokumen Relevan : 4
[250, 126, 232, 231]
Precision @10 : 0.4
=====
---RELEVANCE FEEDBACK---
User feedback [docid]: 250, 126, 232, 231
=====
Jumlah Segmen: 5
SEGID: 2 ==> RSV: [-0.32939162037447256]
=====
Query baru: umum
Query baru: isi
Query baru: trik
=====
-----QUERY EXPANSION-----
[['pelajaran', 'sekolah', 'umum', 'isi', 'trik']]
=====
DOCID: 250 RSV: [5.841326829057092]
DOCID: 75 RSV: [4.256652499130837]
DOCID: 44 RSV: [2.850886600110366]
DOCID: 152 RSV: [2.8175426671652866]
DOCID: 63 RSV: [2.70230003737592]
DOCID: 236 RSV: [2.526012088912119]
DOCID: 114 RSV: [2.3905700917589003]
DOCID: 46 RSV: [2.228677977955068]
DOCID: 21 RSV: [2.1844707756934008]
DOCID: 24 RSV: [2.16696436824146]
=====
Hasil Dokumen setelah RF: 10
Dokumen Relevan setelah RF: 5
[250, 44, 63, 236, 24]
Precision @10 RF : 0.5

```

Gambar 5. 1 Tampilan Antarmuka

BAB 6 HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil pengujian yang dilakukan pada sistem temu kembali informasi pencarian *e-book* berbahasa Indonesia menggunakan metode BM25 dengan teknik *relevance feedback* beserta pembahasannya.

6.1 Pengujian *Precision@K*

Sistem diuji berdasarkan nilai *precision @K* yang dibandingkan sebelum *relevance feedback* dan sesudah *relevance feedback*. Terdapat 2 jenis pengujian yang dilakukan, yaitu pengujian berdasarkan nilai *K* dan pengujian berdasarkan *expansion terms*. Data uji yang digunakan sebanyak 5 *query* yaitu, 'belajar bahasa', 'buku anak', 'sejarah politik', 'buku motivasi', dan 'pelajaran sekolah'. Data latih yang digunakan berupa dokumen *e-book* sebanyak 260 dokumen. Pengujian dilakukan dengan menggunakan dua *user*.

6.1.1 Pengujian berdasarkan Nilai *K*

Pengujian ini dilakukan untuk mengetahui pengaruh pengambilan dokumen teratas dengan nilai *K* sebagai peringkat banyaknya dokumen teratas yang diambil. Percobaan dilakukan sebanyak 6 kali pada masing-masing data uji dengan nilai *K* adalah 10, 15, 20, 25, 30, dan 35. Pada pengujian ini, *expansion terms* yang ditambahkan sebanyak 1 kata.

1. Percobaan 1

Percobaan pertama dilakukan dengan nilai *K* adalah 10. Dokumen yang diambil sebanyak 10 dokumen teratas yang dihasilkan dari perhitungan menggunakan metode BM25. Hasil pengujian pada percobaan 1 dapat dilihat pada Tabel 6.1 untuk *user A* dan Tabel 6.2 untuk *user B*.

a. *User A*

Tabel 6. 1 Hasil Percobaan 1 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,600 | 0,600 | 0,600 |
| 2 | 0,900 | 0,900 | 0,700 |
| 3 | 0,800 | 0,800 | 0,700 |
| 4 | 0,800 | 0,700 | 0,700 |
| 5 | 0,400 | 0,300 | 0,400 |

b. *User B*Tabel 6. 2 Hasil Percobaan 1 (*User B*)

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | 0,700 | 0,800 | 0,700 |
| 2 | 0,800 | 0,800 | 0,600 |
| 3 | 0,900 | 0,800 | 0,900 |
| 4 | 0,900 | 0,600 | 0,900 |
| 5 | 0,500 | 0,500 | 0,600 |

2. Percobaan 2

Percobaan kedua dilakukan dengan nilai K adalah 15. Dokumen yang diambil sebanyak 15 dokumen teratas yang dihasilkan dari perhitungan menggunakan metode BM25. Hasil pengujian pada percobaan 2 dapat dilihat pada Tabel 6.3 untuk *user A* dan Tabel 6.4 untuk *user B*.

a. *User A*Tabel 6. 3 Hasil Percobaan 2 (*User A*)

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | 0,467 | 0,467 | 0,400 |
| 2 | 0,800 | 0,800 | 0,800 |
| 3 | 0,800 | 0,733 | 0,733 |
| 4 | 0,800 | 0,667 | 0,800 |
| 5 | 0,300 | 0,267 | 0,333 |

b. *User B*Tabel 6. 4 Hasil Percobaan 2 (*User B*)

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | 0,533 | 0,667 | 0,533 |
| 2 | 0,600 | 0,667 | 0,667 |
| 3 | 0,733 | 0,467 | 0,733 |
| 4 | 0,600 | 0,667 | 0,600 |
| 5 | 0,533 | 0,467 | 0,667 |

3. Percobaan 3

Percobaan ketiga dilakukan dengan nilai K adalah 20. Dokumen yang diambil sebanyak 20 dokumen teratas yang dihasilkan dari perhitungan

menggunakan metode BM25. Hasil pengujian pada percobaan 3 dapat dilihat pada Tabel 6.5 untuk *user A* dan Tabel 6.6 untuk *user B*.

a. *User A*

Tabel 6. 5 Hasil Percobaan 3 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,350 | 0,350 | 0,350 |
| 2 | 0,700 | 0,700 | 0,700 |
| 3 | 0,700 | 0,800 | 0,700 |
| 4 | 0,650 | 0,600 | 0,700 |
| 5 | 0,300 | 0,300 | 0,300 |

b. *User B*

Tabel 6. 6 Hasil Percobaan 3 (*User B*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,550 | 0,450 | 0,450 |
| 2 | 0,700 | 0,800 | 0,700 |
| 3 | 0,750 | 0,900 | 0,800 |
| 4 | 0,750 | 0,750 | 0,600 |
| 5 | 0,450 | 0,400 | 0,450 |

4. **Percobaan 4**

Percobaan keempat dilakukan dengan nilai K adalah 25. Dokumen yang diambil sebanyak 25 dokumen teratas yang dihasilkan dari perhitungan menggunakan metode BM25. Hasil pengujian pada percobaan 4 dapat dilihat pada Tabel 6.7 untuk *user A* dan Tabel 6.8 untuk *user B*.

a. *User A*

Tabel 6. 7 Hasil Percobaan 4 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,320 | 0,320 | 0,320 |
| 2 | 0,600 | 0,600 | 0,600 |
| 3 | 0,720 | 0,760 | 0,640 |
| 4 | 0,560 | 0,600 | 0,720 |
| 5 | 0,320 | 0,280 | 0,320 |

b. *User B*Tabel 6. 8 Hasil Percobaan 4 (*User B*)

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | 0,520 | 0,360 | 0,560 |
| 2 | 0,600 | 0,640 | 0,400 |
| 3 | 0,680 | 0,800 | 0,720 |
| 4 | 0,760 | 0,600 | 0,640 |
| 5 | 0,560 | 0,360 | 0,480 |

5. Percobaan 5

Percobaan kelima dilakukan dengan nilai K adalah 30. Dokumen yang diambil sebanyak 30 dokumen teratas yang dihasilkan dari perhitungan menggunakan metode BM25. Hasil pengujian pada percobaan 5 dapat dilihat pada Tabel 6.9 untuk *user A* dan Tabel 6.10 untuk *user B*.

a. *User A*Tabel 6. 9 Hasil Percobaan 5 (*User A*)

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | 0,267 | 0,267 | 0,267 |
| 2 | 0,567 | 0,533 | 0,500 |
| 3 | 0,733 | 0,700 | 0,633 |
| 4 | 0,600 | 0,500 | 0,600 |
| 5 | 0,367 | 0,267 | 0,367 |

b. *User B*Tabel 6. 10 Hasil Percobaan 5 (*User B*)

| Query | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-------|------------|----------------|----------------|
| 1 | 0,433 | 0,267 | 0,433 |
| 2 | 0,600 | 0,533 | 0,633 |
| 3 | 0,833 | 0,833 | 0,733 |
| 4 | 0,633 | 0,467 | 0,600 |
| 5 | 0,533 | 0,467 | 0,533 |

6. Percobaan 6

Percobaan keenam dilakukan dengan nilai K adalah 35. Dokumen yang diambil sebanyak 35 dokumen teratas yang dihasilkan dari perhitungan menggunakan metode BM25. Hasil pengujian pada

percobaan 6 dapat dilihat pada Tabel 6.11 untuk *user A* dan Tabel 6.12 untuk *user B*.

a. *User A*

Tabel 6. 11 Hasil Percobaan 6 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,229 | 0,229 | 0,229 |
| 2 | 0,514 | 0,514 | 0,514 |
| 3 | 0,657 | 0,657 | 0,629 |
| 4 | 0,514 | 0,543 | 0,543 |
| 5 | 0,314 | 0,343 | 0,314 |

b. *User B*

Tabel 6. 12 Hasil Percobaan 6 (*User B*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,429 | 0,229 | 0,457 |
| 2 | 0,486 | 0,486 | 0,486 |
| 3 | 0,771 | 0,629 | 0,686 |
| 4 | 0,571 | 0,543 | 0,571 |
| 5 | 0,429 | 0,429 | 0,429 |

Berdasarkan keenam percobaan pada setiap *user* didapatkan hasil rata-rata nilai *precision* pada setiap nilai *K*. Perbandingan hasil *average precision* (AVP) yang diperoleh pada saat sebelum melakukan *relevance feedback* dan setelah melakukan *relevance feedback* ditunjukkan pada Tabel 6.13 dan Gambar 6.1 untuk *user A*, Tabel 6.14 dan Gambar 6.2 untuk *user B*.

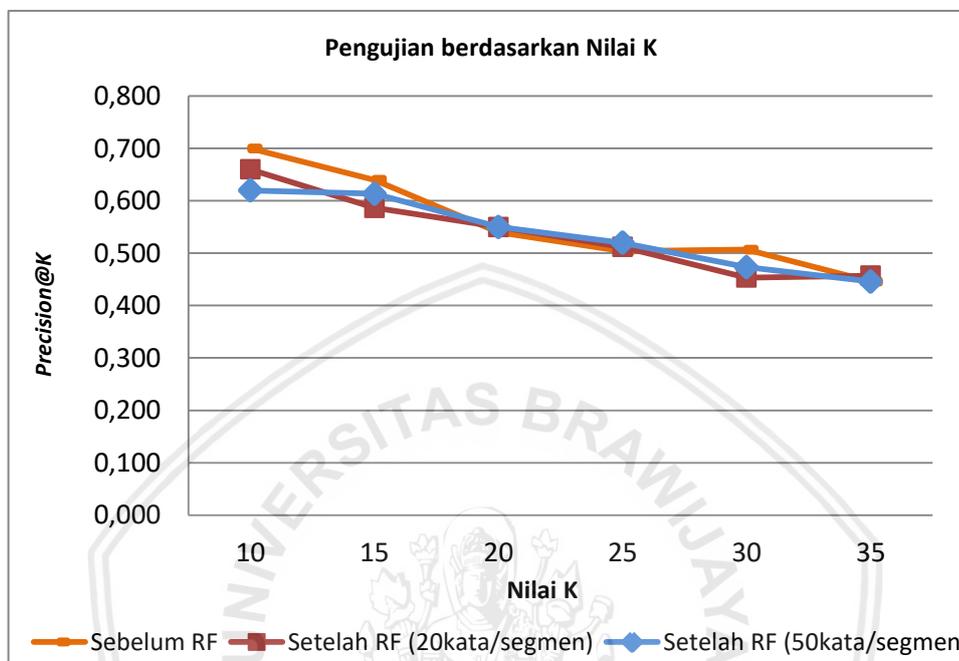
a. *User A*

Tabel 6. 13 Perbandingan AVP pengujian berdasarkan nilai *K* (*User A*)

| Percobaan | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-----------|------------|----------------|----------------|
| 1 | 0,700 | 0,660 | 0,620 |
| 2 | 0,640 | 0,587 | 0,613 |
| 3 | 0,540 | 0,550 | 0,550 |
| 4 | 0,504 | 0,512 | 0,520 |
| 5 | 0,507 | 0,453 | 0,473 |
| 6 | 0,446 | 0,457 | 0,446 |
| Rata-rata | 0,556 | 0,533 | 0,537 |

Pada Tabel 6.13, dapat dilihat bahwa Percobaan 1 oleh *user A* memiliki rata-rata nilai *precision@K* yang paling baik pada saat sebelum dan setelah *relevance feedback*, dengan nilai *K* adalah 10. Tetapi, jika dilihat berdasarkan

kenaikan rata-rata nilai *precision@K* sebelum dan setelah *relevance feedback*, percobaan 4 adalah yang paling baik, dengan nilai *K* adalah 25. Selanjutnya jika dibandingkan antara setelah RF(20) dan setelah RF(50), nilai *precision* pada saat setelah RF(50) cenderung lebih baik daripada setelah RF(20).



Gambar 6. 1 Grafik Perbandingan AVP pengujian berdasarkan nilai K (User B)

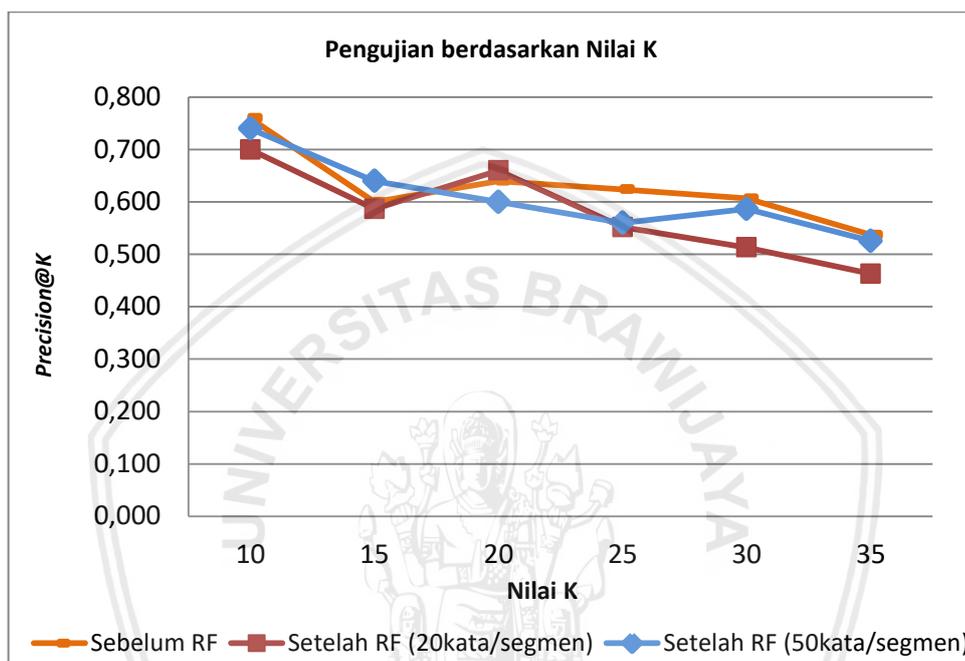
Pada Gambar 6.1, grafik menunjukkan bahwa perbandingan berdasarkan nilai *K* menghasilkan nilai *precision* yang apabila semakin besar nilai *K*, maka nilai *precision@K* semakin rendah. Selanjutnya, jika dibandingkan berdasarkan hasil sebelum, setelah *relevance feedback* (20) dan setelah *relevance feedback* (50), nilai *precision@K* setelah *relevance feedback* cenderung menurun dengan perbandingan antara setelah RF (20) dan setelah RF (50) tidak begitu signifikan bedanya.

b. User B

Tabel 6. 14 Perbandingan AVP pengujian berdasarkan nilai K (User B)

| Percobaan | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-----------|------------|----------------|----------------|
| 1 | 0,760 | 0,700 | 0,740 |
| 2 | 0,600 | 0,587 | 0,640 |
| 3 | 0,640 | 0,660 | 0,600 |
| 4 | 0,624 | 0,552 | 0,560 |
| 5 | 0,607 | 0,513 | 0,587 |
| 6 | 0,537 | 0,463 | 0,526 |
| Rata-rata | 0,628 | 0,579 | 0,609 |

Pada Tabel 6.14, dapat dilihat bahwa Percobaan 1 memiliki rata-rata nilai *precision@K* yang paling baik pada saat sebelum dan setelah *relevance feedback*, dengan nilai *K* adalah 10. Tetapi, jika dilihat berdasarkan kenaikan rata-rata nilai *precision@K* sebelum dan setelah *relevance feedback*, percobaan 3 adalah yang paling baik, dengan nilai *K* adalah 20. Selanjutnya jika dibandingkan antara setelah RF(20) dan setelah RF(50), nilai *precision* pada saat setelah RF(50) cenderung lebih baik daripada setelah RF(20).



Gambar 6. 2 Grafik Perbandingan AVP pengujian berdasarkan nilai K (User B)

Pada Gambar 6.2, grafik menunjukkan bahwa perbandingan berdasarkan nilai *K* menghasilkan nilai *precision* yang apabila semakin besar nilai *K*, maka nilai *precision@K* semakin rendah. Selanjutnya, jika dibandingkan berdasarkan hasil sebelum RF, setelah RF (20), dan setelah RF (50), nilai *precision@K* setelah *relevance feedback* cenderung menurun dengan perbandingan antara setelah RF (20) dan setelah RF (50) dapat dilihat bahwa pada nilai *precision* setelah RF(50) cenderung meningkat daripada setelah RF(20).

6.1.2 Pengujian berdasarkan *Expansion Terms*

Pengujian ini dilakukan untuk mengetahui pengaruh banyaknya penambahan kata (*term*) untuk perluasan *query*. *Term* dengan peringkat *y* teratas akan ditambahkan pada *query* baru. Percobaan dilakukan sebanyak 3 kali pada masing-masing data uji dengan *y* adalah 1, 2, dan 3 kata. Pada pengujian ini, dokumen yang diambil sebanyak 20 dokumen teratas.

1. Percobaan 7

Percobaan ketujuh dilakukan dengan nilai *y* adalah 1. Kata yang ditambah sebanyak 1 kata teratas yang dihasilkan dari perhitungan pada

persamaan 2.1 . Hasil pengujian pada percobaan 7 dapat dilihat pada Tabel 6.15 untuk *user A* dan Tabel 6.16 untuk *user B*.

a. *User A*

Tabel 6. 15 Hasil Percobaan 7 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,350 | 0,350 | 0,350 |
| 2 | 0,700 | 0,700 | 0,700 |
| 3 | 0,700 | 0,800 | 0,700 |
| 4 | 0,650 | 0,600 | 0,700 |
| 5 | 0,300 | 0,300 | 0,300 |

b. *User B*

Tabel 6. 16 Hasil Percobaan 7 (*User B*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,550 | 0,450 | 0,450 |
| 2 | 0,700 | 0,800 | 0,700 |
| 3 | 0,750 | 0,900 | 0,800 |
| 4 | 0,750 | 0,750 | 0,600 |
| 5 | 0,450 | 0,400 | 0,450 |

2. Percobaan 8

Percobaan kedelapan dilakukan dengan nilai y adalah 2. Kata yang ditambah sebanyak 2 kata teratas yang dihasilkan dari perhitungan pada persamaan 2.1 . Hasil pengujian pada percobaan 8 dapat dilihat pada Tabel 6.17 untuk *user A* dan Tabel 6.18 untuk *user B*.

a. *User A*

Tabel 6. 17 Hasil Percobaan 8 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,350 | 0,350 | 0,350 |
| 2 | 0,700 | 0,700 | 0,700 |
| 3 | 0,700 | 0,800 | 0,700 |
| 4 | 0,650 | 0,500 | 0,750 |
| 5 | 0,300 | 0,350 | 0,350 |

b. *User B*Tabel 6. 18 Hasil Percobaan 8 (*User B*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,550 | 0,550 | 0,650 |
| 2 | 0,700 | 0,700 | 0,500 |
| 3 | 0,850 | 0,850 | 0,800 |
| 4 | 0,900 | 0,800 | 0,800 |
| 5 | 0,450 | 0,450 | 0,400 |

3. Percobaan 9

Percobaan kesembilan dilakukan dengan nilai y adalah 3. Kata yang ditambah sebanyak 3 kata teratas yang dihasilkan dari perhitungan pada persamaan 2.1. Hasil pengujian pada percobaan 9 dapat dilihat pada Tabel 6.19 untuk *user A* dan Tabel 6.20 untuk *user B*.

a. *User A*Tabel 6. 19 Hasil Percobaan 9 (*User A*)

| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,350 | 0,300 | 0,350 |
| 2 | 0,700 | 0,650 | 0,700 |
| 3 | 0,700 | 0,750 | 0,750 |
| 4 | 0,650 | 0,500 | 0,750 |
| 5 | 0,300 | 0,350 | 0,350 |

b. *User B*Tabel 6. 20 Hasil Percobaan 9 (*User B*)

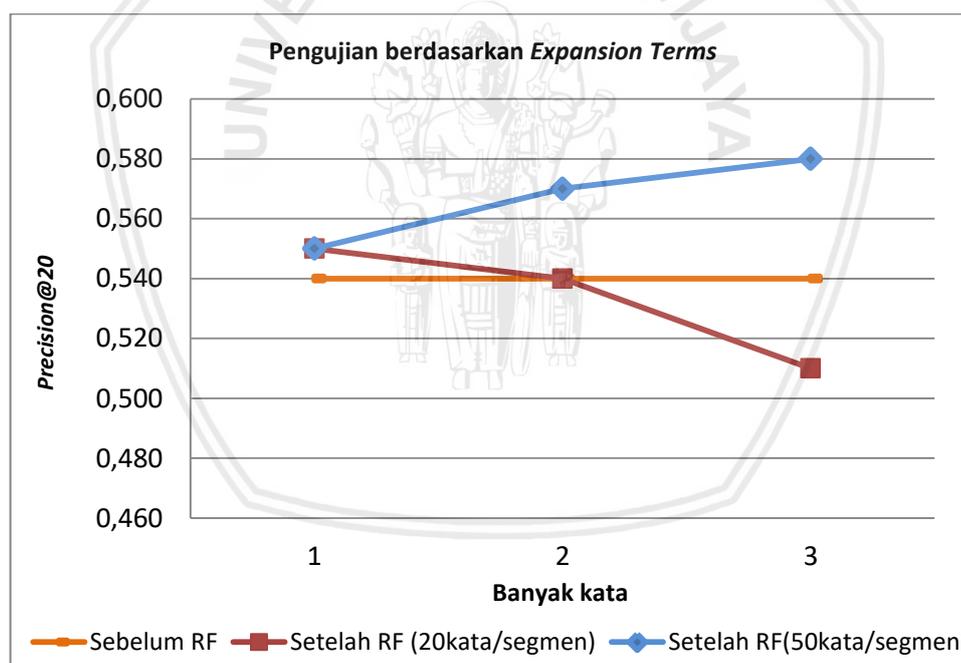
| <i>Query</i> | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|--------------|------------|----------------|----------------|
| 1 | 0,450 | 0,350 | 0,350 |
| 2 | 0,750 | 0,550 | 0,550 |
| 3 | 0,800 | 0,850 | 0,650 |
| 4 | 0,650 | 0,800 | 0,550 |
| 5 | 0,400 | 0,400 | 0,350 |

Berdasarkan ketiga percobaan didapatkan hasil rata-rata nilai *precision* pada setiap nilai y yang merupakan peringkat teratas kata yang dipilih untuk ditambahkan sebagai *query* baru. Perbandingan hasil *average precision* (AVP) yang diperoleh pada saat sebelum melakukan *relevance feedback* dan setelah melakukan *relevance feedback* ditunjukkan pada Tabel 6.21 dan Gambar 6.3 untuk *user A*, Tabel 6.22 dan Gambar 6.4 untuk *user B*.

a. *User A*Tabel 6. 21 Perbandingan AVP pengujian berdasarkan *expansion terms* (*User A*)

| Percobaan | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-----------|------------|----------------|----------------|
| 7 | 0,540 | 0,550 | 0,550 |
| 8 | 0,540 | 0,540 | 0,570 |
| 9 | 0,540 | 0,510 | 0,580 |
| Rata-rata | 0,540 | 0,533 | 0,567 |

Pada Tabel 6.21, dapat dilihat bahwa Percobaan 7 oleh *user A* memiliki kenaikan nilai *precision@K* yang paling baik setelah RF(20), yaitu pada saat *expansion terms* sebanyak 2 kata. Sedangkan percobaan 9 oleh *user A* memiliki kenaikan nilai *precision@K* yang paling baik setelah RF(50). Selanjutnya jika dilihat berdasarkan kenaikan rata-rata nilai *precision@K*, rata-rata nilai *precision* setelah RF(50) lebih baik dibandingkan setelah RF(20) dan sebelum RF, dan rata-rata nilai *precision* sebelum RF lebih baik dibandingkan setelah RF(20).

Gambar 6. 3 Grafik Perbandingan AVP pengujian berdasarkan *Expansion Terms* (*User A*)

Pada Gambar 6.3, grafik menunjukkan bahwa perbandingan berdasarkan *expansion terms*, menghasilkan nilai *precision* yang apabila semakin banyak kata yang ditambahkan, maka nilai *precision@K* cenderung meningkat. Namun hanya nilai *precision@K* setelah RF(50) saja yang meningkat, sedangkan pada nilai *precision@K* setelah RF(20) cenderung menurun, dan pada saat sebelum *relevance feedback* cenderung tetap. Selanjutnya, jika perbandingan berdasarkan hasil sebelum dan setelah *relevance feedback*, nilai *precision@K* setelah

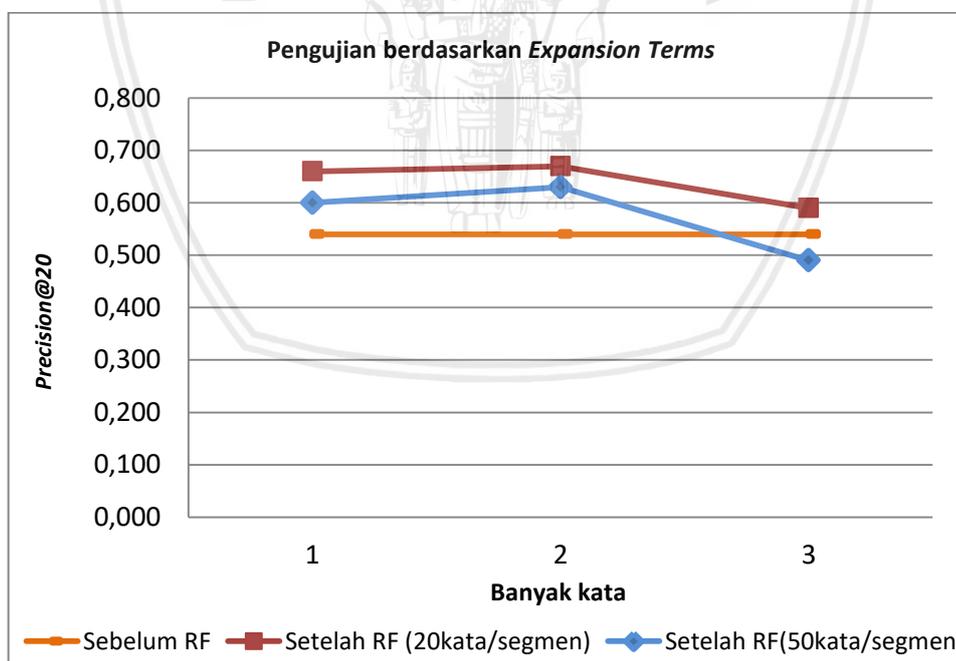
relevance feedback cenderung meningkat. Kenaikan dan penurunan nilai *precision* tergantung oleh kata baru yang ditambahkan setelah RF. Jika kata tidak sesuai pada topik *query* awal maka nilai *precision* akan mengalami penurunan, sedangkan jika kata yang ditambahkan sesuai pada topik *query* awal, maka nilai *precision* akan cenderung meningkat.

b. User B

Tabel 6. 22 Perbandingan AVP pengujian berdasarkan *expansion terms* (User B)

| Percobaan | Sebelum RF | Setelah RF(20) | Setelah RF(50) |
|-----------|------------|----------------|----------------|
| 7 | 0,640 | 0,660 | 0,600 |
| 8 | 0,690 | 0,670 | 0,630 |
| 9 | 0,610 | 0,590 | 0,490 |
| Rata-rata | 0,647 | 0,640 | 0,573 |

Pada Tabel 6.22, dapat dilihat bahwa Percobaan 7 oleh *user B* memiliki kenaikan nilai *precision@K* yang paling baik pada saat setelah RF(20), yaitu pada saat *expansion terms* sebanyak 1 kata. Sedangkan pada saat setelah RF(50), tidak ada percobaan yang mengalami kenaikan nilai *precision*. Selanjutnya jika dilihat berdasarkan kenaikan rata-rata nilai *precision@K* sebelum RF, setelah RF(20), dan setelah RF(50), nilai *precision@K* sebelum RF lebih tinggi. Namun penurunan nilai *precision* paling signifikan terjadi pada saat setelah RF(50).



Gambar 6. 4 Grafik Perbandingan AVP pengujian berdasarkan *Expansion Terms* (User B)

Pada Gambar 6.4, grafik menunjukkan bahwa perbandingan berdasarkan *expansion terms*, menghasilkan nilai *precision* yang semakin banyak kata yang ditambahkan, maka nilai *precision@K* cenderung menurun. Selanjutnya, jika

perbandingan berdasarkan hasil sebelum dan setelah *relevance feedback*, nilai *precision@K* setelah *relevance feedback* cenderung meningkat. Perbandingan antara setelah RF(20) dengan setelah RF(50) jika dilihat pada grafik pada Gambar 6.4, kenaikan nilai *precision* setelah RF(20) lebih baik dibandingkan dengan setelah RF(50). Kenaikan dan penurunan nilai *precision* tergantung oleh kata baru yang ditambahkan setelah RF. Jika kata tidak sesuai pada topik *query* awal maka nilai *precision* akan mengalami penurunan, sedangkan jika kata yang ditambahkan sesuai pada topik *query* awal, maka nilai *precision* akan cenderung meningkat.

6.2 Analisis Hasil Pengujian

Pada kedua pengujian, didapatkan bahwa rata-rata nilai *precision@k* sebelum *relevance feedback* cenderung lebih tinggi daripada rata-rata nilai *precision@k* setelah *relevance feedback*. Hal itu dikarenakan, penggunaan *feedback* dalam perluasan *query* tersebut menghasilkan penambahan kata yang sebagian besar tidak berhubungan dengan topik pada *query* awal. Sehingga peringkat dari dokumen relevan menjadi menurun. Perluasan *query* didapat dari kata-kata dengan nilai *TSV* tertinggi yang dihitung dari persamaan 2.1. Pada pengujian ini, nilai *TSV* pada peringkat pertama menghasilkan banyak kata yang memiliki nilai *TSV* yang sama. Sehingga sistem dengan otomatis memilih secara acak berdasarkan kemunculan pertama kata yang memiliki nilai *TSV* tertinggi pada suatu segmen. Itulah yang menyebabkan *expansion terms* sebagian besar tidak sesuai dengan topik pada *query* awal.

Pada penelitian ini, terdapat 2 jenis pengujian setelah RF, yaitu setelah RF(20) dengan mengambil minimal 20 kata pada suatu segmen dan setelah RF(50) dengan mengambil minimal 50 kata pada suatu segmen. Tidak terdapat perbedaan yang signifikan pada keduanya, namun jika dilihat berdasarkan kenaikan nilai *precisionnya*, pada pengujian setelah RF(50) memiliki nilai yang lebih baik dibandingkan dengan setelah RF(20), hal itu karena jumlah kata setelah RF(50) yang diproses lebih banyak dibandingkan setelah RF(20). Sehingga dari jumlah kata yang lebih banyak, sistem akan lebih mudah mencari kata yang paling sesuai dengan topik pada *query* awal yang memiliki nilai *TSV* tertinggi.

Berdasarkan pengujian yang dilakukan oleh 2 *user*, dapat dilihat bahwa pada *user B* menghasilkan nilai *precision* yang lebih tinggi dibandingkan dengan *user A*. Hal itu karena terdapat perbedaan dalam penginputan dokumen *feedback* pada kedua *user*. Pada *user A*, penginputan *feedback* dilakukan dengan menginputkan semua dokumen relevan berdasarkan *query* pada pencarian awal, sedangkan pada *user B*, penginputan *feedback* dilakukan dengan menginputkan beberapa dokumen relevan dengan jumlah dokumen yang diinputkan sesuai kebutuhan *user*, sehingga *feedback* yang kemudian direformulasikan oleh sistem menjadi *query* baru berpeluang menghasilkan kata yang ditambahkan sesuai dengan *query* awal.

Pada Gambar 6.1 dan Gambar 6.2, dapat dilihat bahwa pengaruh pengambilan dokumen berdasarkan nilai K menghasilkan nilai *precision*, yang semakin besar nilai K maka nilai *precision*-nya makin rendah. Hal itu karena semakin banyak dokumen yang diambil, maka perbandingan antara dokumen yang dihasilkan dengan dokumen yang relevan semakin jauh. Selain itu, jumlah *term* yang akan diproses menjadi lebih banyak. Pada pengujian ini, *query* baru yang dihasilkan pada setiap nilai K berbeda-beda. Hal itu juga dapat memengaruhi banyaknya dokumen relevan yang dihasilkan.

Pada Gambar 6.3 dan Gambar 6.4, dapat dilihat bahwa nilai *precision* cenderung mengalami kenaikan setelah *relevance feedback*. Kenaikan tersebut terjadi jika kata-kata yang ditambahkan sesuai dengan topik pada *query* awal. Namun jika dilihat berdasarkan banyak kata yang ditambahkan, dapat dilihat bahwa semakin banyak kata yang ditambahkan maka nilai *precision* semakin menurun, hal itu karena kata-kata yang ditambahkan merupakan kata yang tidak sesuai dengan topik pada *query* awal. Sehingga membuat dokumen relevan mengalami penurunan peringkat, akibatnya banyak dokumen relevan yang tidak ikut diambil.

Misalkan, pada segmen 1 terdapat kata “meja”, “piawai”, “bicara”, “dukung”, “bidang”, “tarik”, “taktik”, “seni”, “kuasa”, “suasana”. Kata-kata tersebut memiliki nilai *TSV* yang sama, yaitu senilai 0,079. Karena kata dengan kemunculan pertama pada segmen 1 yang memiliki nilai *TSV* tertinggi adalah kata “meja”, maka kata “meja” akan digunakan untuk penambahan kata pada *query* “buku cara cerdas sukses” menjadi “buku cara cerdas sukses meja”. Penambahan kata “meja” dalam percobaan ini membuat nilai *precision* akan menurun karena kata “meja” tidak berhubungan dengan topik pada *query* awal yaitu “buku cara cerdas sukses”.

BAB 7 PENUTUP

Bab ini menjelaskan tentang penutup yang berupa kesimpulan dari hasil penelitian yang didapatkan dari pengujian, serta saran-saran yang diberikan oleh penulis untuk penelitian selanjutnya mengenai sistem temu kembali informasi menggunakan teknik *relevance feedback*.

7.1 Kesimpulan

Berdasarkan hasil pengujian, kesimpulan dari penelitian *relevance feedback* sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25 adalah sebagai berikut.

1. Penerapan *relevance feedback* sistem temu kembali informasi dokumen *e-book* berbahasa Indonesia menggunakan metode BM25 menghasilkan kinerja sistem yang cukup baik. Meskipun kinerja sistem sebelum *relevance feedback* lebih baik dibandingkan setelah *relevance feedback*, tetapi tidak ditemukan perbedaan yang signifikan.
2. Penelitian ini diuji berdasarkan 2 jenis pengujian. Pada pengujian berdasarkan nilai K , rata-rata nilai *precision@K* sebelum RF adalah 0,592, setelah RF(20) adalah 0,573, dan setelah RF(50) adalah 0,558. Pada pengujian berdasarkan *expansion terms*, rata-rata nilai *precision* sebelum RF adalah 0,593, setelah RF(20) adalah 0,570, dan setelah RF(50) adalah 0,587.
3. Dari kedua pengujian, didapatkan bahwa rata-rata nilai *precision@K* sebelum *relevance feedback* lebih tinggi daripada rata-rata nilai *precision@K* setelah *relevance feedback*. Hal itu karena, penggunaan *feedback* dalam perluasan *query* menghasilkan penambahan kata yang sebagian besar tidak berhubungan dengan topik pada *query* awal. Sehingga peringkat dari dokumen relevan menjadi menurun. Pada penelitian ini, banyak kata yang memiliki nilai *TSV* tertinggi yang sama. Sehingga sistem memilih secara urut berdasarkan kemunculan pertama kata yang memiliki nilai *TSV* tertinggi pada suatu segmen. Itulah yang menyebabkan perluasan *query* menghasilkan penambahan kata yang sebagian besar tidak sesuai dengan topik pada *query* awal.

7.2 Saran

Adapun saran penulis untuk penelitian selanjutnya adalah sebagai berikut.

1. Untuk mendapatkan hasil yang lebih baik, sebaiknya koleksi dokumen yang digunakan lebih banyak. Selain itu, kesalahan penulisan pada

dokumen lebih diperhatikan agar kata-kata yang tidak bermakna tidak muncul setelah proses *preprocessing*.

2. Untuk mendapatkan *query* baru yang berhubungan dengan *query* awal, sebaiknya pengguna dapat memilih kata yang akan ditambahkan dari kata-kata terbaik yang telah disediakan sistem.



DAFTAR REFERENSI

- Adisantoso, J., Ridha A., & Agusetyawan, A.W., 2004. *Relevance Feedback Pada Temu-Kembali Teks Berbahasa Indonesia Dengan Metode IDE-DEC-HI Dan IDE-REGULAR*. Bogor: Institut Pertanian Bogor.
- Anbiana, E.D., 2009. *Pseudo-Relevance Feedback pada Temu-Kembali menggunakan Segmentasi Dokumen*. S1. Institut Pertanian Bogor. Tersedia di <<https://repository.ipb.ac.id/handle/123456789/12603>> [Diakses 28 Oktober 2018].
- Erwin A.H, M. & Mandala R., 2004. *Relevance Feedback pada Temu Kembali Informasi Menggunakan Algoritma Genetika*. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi.
- Fakultas Ilmu Komputer Universitas Brawijaya, 2017. *Panduan Skripsi Fakultas Ilmu Komputer Universitas Brawijaya 2017*, edisi 2. [pdf]. Fakultas Ilmu Komputer Universitas Brawijaya. Tersedia di: <<http://file.filkom.ub.ac.id/fileupload/assets/upload/filemanager/Skripsi/PanduanSkripsiFilkom-v2.pdf>> [Diakses 31 Agustus 2018]
- Hasugian, J., 2003. *Penggunaan Bahasa Alamiah dan Kosa Kata Terkontrol dalam Sistem Temu Kembali Informasi Berbasis Teks*. Medan: Universitas Sumatera Utara.
- Indriyani, W., 2011. *Pseudo-Relevance Feedback pada Temu-Kembali menggunakan Segmentasi Kalimat*. S1. Institut Pertanian Bogor. Tersedia di <<https://repository.ipb.ac.id/handle/123456789/51257>> [Diakses 28 Oktober 2018]
- Manning, C.D., Ragnavan, P., & Schutze, H., 2008. *Introduction to Information Retrieval*. [pdf]. Cambridge University Press. Tersedia di: <<https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf>> [Diakses 31 Agustus 2018]
- Mandala, R., 2006. *Evaluasi Efektifitas Metode Machine-Learning Pada Search-Engine*. Bandung: Institut Teknologi Bandung.
- Pardede, J., Husada, M.G., & Riansyah, R., 2015. *Implementasi dan Perbandingan Metode Okapi BM25 dan PLSA pada Aplikasi Information Retrieval*. Bandung: Itenas Bandung.
- Putera, P. B., 2011. *Ebook dan Pasar Perbukuan Kini*. [online]. Tersedia di: <<http://www.buku-e.lipi.go.id/utama.cgi?artikel&1321295564&&1>> [Diakses 31 Agustus 2018].

- Ruthven, I, & Lalmas M., 2003. *A survey on the Use of Relevance Feedback for Information Access Systems*. *Knowledge Engineering Review*, 18(2), 95-145.
- Russel, S.J., & Norvig, P., 2010. *Artificial Intelligence: A Modern Approach*, 3rd Edition. [pdf]. New Jersey: Prentice Hall. Tersedia di <<https://faculty.psau.edu.sa/filedownload/doc7pdfa154ffbcec538a4161a406abf62f5b76-original.pdf>> [Diakses 10 September].
- Sari, S., & Adriani, M., 2014. *Learning to Rank for Determine Relevant Document in Indonesian-English Cross Language Information Retrieval using BM25*. Depok: University of Indonesia.
- Subari, & Ferdinandus, 2015. *Sistem Information Retrieval Layanan Kesehatan untuk Berobat dengan Metode Vector Space Model (VSM) Berbasis Webgis*. *SNATIKA* 2015. 3(1), 202-212.
- Tjandra, E., & Widiastri, M., 2016. *Sistem Repositori Tugas Akhir Mahasiswa dengan Fungsi Peringkat Okapi BM25*. Surabaya: Universitas Surabaya.
- Yugianus, P., Dachlan, H.S., & Hasanah, R.N., 2013. *Pengembangan Sistem Penelusuran Katalog Perpustakaan Dengan Metode Rocchio Relevance Feedback*. *Jurnal EECCIS*. 7(1), 47-52.
- Yu, S., et al., 2003. *Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation*. In *The Fifth Asia Pacific Web Conference*.