

**QUERY EXPANSION PADA SISTEM TEMU KEMBALI  
INFORMASI DOKUMEN JURNAL BERBAHASA INDONESIA  
MENGUNAKAN METODE BM25**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Faradila Puspa Wardani  
NIM: 155150207111151



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

# PENGESAHAN

QUERY EXPANSION PADA SISTEM TEMU KEMBALI INFORMASI DOKUMEN  
JURNAL BERBAHASA INDONESIA MENGGUNAKAN METODE BM25

SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Faradila Puspa Wardani  
NIM: 155150207111151

Skripsi ini telah diuji dan dinyatakan lulus pada  
2 Januari 2019  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Indriati, S.T, M.Kom  
NIP: 19831013 201504 2 002



Bayu Rahayudi, S.T, M.T  
NIP: 19740712 200604 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Komtawan, 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 pustaka.

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

Malang, 2 Januari 2019



Faradila Puspa Wardani

NIM: 155150207111151

## PRAKATA

Puji syukur penulis panjatkan atas kehadiran Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian yang berjudul “*Query Expansion Pada Sistem Temu Kembali Informasi Dokumen Jurnal Berbahasa Indonesia Menggunakan Metode BM25*”.

Penelitian ini merupakan syarat untuk memenuhi sebagian persyaratan kurikulum pada Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya. Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada pihak-pihak yang telah membantu selama penyusunan laporan penelitian ini, diantaranya:

1. Allah SWT yang telah memberikan segalanya, sehingga penulis berkesempatan untuk dapat menempuh pendidikan dan menyelesaikan skripsi penulis.
2. Keluarga penulis, mamah, bapak, mas isal, mbak yaya, dek bela yang dengan sepenuh hati membantu penulis dalam segala hal, serta selalu memberikan doa dan motivasi kepada penulis dalam menyelesaikan skripsi ini.
3. Ibu Indriati, S.T., M.Kom. selaku Dosen Pembimbing I yang dengan tulus ikhlas membimbing dan membantu penulis dalam menyelesaikan skripsi ini.
4. Bapak Bayu Rahayudi, S.T, M.T. selaku Dosen Pembimbing II yang dengan sabar membimbing dan mengarahkan penulis dalam penyusunan laporan skripsi ini.
5. BTS, yang telah menginspirasi, memotivasi dan menjadi penyemangat penulis selama proses penyelesaian skripsi ini.
6. Femi, Nyenye, Ardia, Sasa, Wiwit, yang telah mengisi hari-hari dengan penuh tawa, dan menjadi penyemangat penulis selama diperantauan.
7. Lilis, Ika, Suryani, Yasmin, yang telah mengisi hari-hari penulis dan menjadi keluarga pertama selama diperantauan.
8. Nabila, Nissa, Mutya, Nadin, Alma, Ayu, Nana, yang telah memberikan semangat kepada penulis dalam menyelesaikan skripsi ini.
9. Teman-teman angkatan 2015 Fakultas Ilmu Komputer Universitas Brawijaya, atas bantuannya dalam bentuk apapun.

Penulis menyadari penelitian ini masih banyak kekurangan dan jauh dari kata sempurna, untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun. Semoga penelitian ini dapat bermanfaat bagi pihak yang membutuhkan.

Malang, 2 Januari 2019

Penulis

faradilapw@gmail.com

## ABSTRAK

**Faradila Puspa Wardani, *Query Expansion* Pada Sistem Temu Kembali Informasi Dokumen Jurnal Berbahasa Indonesia Dengan Menggunakan Metode BM25**

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

Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) merupakan kumpulan tulisan karya ilmiah dari penelitian mahasiswa/i Fakultas Ilmu Komputer Universitas Brawijaya. Pada setiap jurnal terdapat bagian abstrak yang merupakan ringkasan dari keseluruhan isi dokumen karya ilmiah, yang mengandung konsep, pernyataan masalah, pendekatan, dan kesimpulan yang dirangkai. Jumlah jurnal yang terdapat pada J-PTIHK sangat banyak, maka dari itu diperlukan suatu sistem pencarian dokumen untuk mempermudah pengguna dalam menemukan dokumen jurnal, yaitu sistem pencarian yang menerapkan Sistem Temu Kembali Informasi (STKI) atau *Information Retrieval* (IR). Untuk mengurangi hasil pengembalian dokumen yang tidak relevan pada STKI, maka dapat diterapkan teknik perbaikan *query* yaitu teknik *query expansion*. Penerapan *query expansion* dapat dilakukan dengan menggunakan metode BM25. BM25 adalah salah satu metode dalam STKI yang digunakan untuk pengurutan hasil relevansi dokumen terhadap *query* yang ingin dicari. Penelitian ini dilakukan dengan bertujuan untuk melihat hasil implementasi *query expansion* pada sistem temu kembali informasi jurnal berbahasa Indonesia menggunakan metode BM25. Hasilnya pada saat penambahan kata untuk *query* sebanyak 4 kata nilai *Precision@K* mengalami kenaikan rata-rata sebesar 0.492.

Kata kunci: jurnal, abstrak, sistem temu kembali informasi, *query expansion*, BM25.

## ABSTRACT

**Faradila Puspa Wardani, Query Expansion in Information Retrieval of Indonesian Journal Using BM25 Method**

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

Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTI IK) is a collection of scientific papers from the research Faculty of Computer Science's students, Universitas Brawijaya. In each journal there is an abstract section which is a summary of the entire contents of the scientific work document, which contains concepts, problem statements, approaches, and conclusions that are arranged. The number of journals contained in J-PTI IK is very large, therefore a document search system is needed to facilitate users in finding journal documents, a search system that implements an Information Retrieval (IR). To minimize the results of returning documents that are not relevant to the IR, query techniques can be applied, called query expansion techniques. Application of query expansion can be done by using BM25 method. BM25 is an IR that is used to sort the results of the relevance documents to the query that user want to search. This research was conducted by aiming to see the results of query expansion in the information retrieval of Indonesian journal using BM25 method. The result, when adding words for queries as many as 4 words, the Precision @ K value increases by an average of 0.492.

**Keywords:** journal, abstract, information retrieval, query expansion, BM25.

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR <i>SOURCE CODE</i> .....	xii
DAFTAR LAMPIRAN .....	xiii
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 Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer.....	9
2.3 Sistem Temu Kembali Informasi.....	9
2.3.1 <i>Case Folding</i> .....	9
2.3.2 <i>Filtering</i> .....	10
2.3.3 <i>Tokenization</i> .....	10
2.3.4 <i>Stemming</i> .....	11
2.4 Query Expansion.....	11
2.5 Metode BM25.....	12
2.6 Evaluasi Sistem .....	13



BAB 3 METODOLOGI .....	14
3.1 Tipe Penelitian .....	14
3.2 Strategi Penelitian .....	14
3.3 Lokasi Penelitian .....	14
3.4 Teknik Pengumpulan Data.....	14
3.5 Perancangan Algoritma .....	15
3.6 Teknik Penerapan Algoritma .....	15
3.7 Peralatan Pendukung .....	16
3.8 Teknik Analisis Data.....	16
BAB 4 PERANCANGAN.....	17
4.1 Deskripsi Umum Sistem.....	17
4.2 Diagram Alir Sistem .....	17
4.2.1 <i>Case Folding</i> .....	19
4.2.2 <i>Filtering</i> .....	20
4.2.3 <i>Tokenization</i> .....	21
4.2.4 <i>Stemming</i> .....	22
4.2.5 Perhitungan TF .....	24
4.2.6 Perhitungan IDF.....	25
4.2.7 Pemingkatan dengan Metode BM25.....	26
4.2.8 Penentuan <i>Query</i> Baru .....	28
4.3 Manualisasi.....	29
4.3.1 <i>Dataset</i> .....	29
4.3.2 <i>Preprocessing</i> .....	30
4.3.3 Pembobotan TF IDF.....	34
4.3.4 Perhitungan Metode BM25 .....	35
4.3.5 Penentuan <i>Query</i> baru .....	36
4.4 Perancangan Pengujian .....	38
4.5 Perancangan Tampilan Antarmuka .....	39
BAB 5 IMPLEMENTASI .....	40
5.1 Lingkungan Implementasi .....	40
5.1.1 Lingkungan Perangkat Lunak .....	40



5.1.2	Lingkungan Perangkat Keras .....	40
5.2	Implementasi Sistem .....	40
5.2.1	<i>Case Folding</i> .....	41
5.2.2	<i>Filtering</i> .....	41
5.2.3	<i>Tokenization</i> .....	41
5.2.4	<i>Stemming</i> .....	42
5.2.5	Perhitungan Nilai TF .....	42
5.2.6	Perhitungan Nilai IDF .....	43
5.2.7	Pemeringkatan Metode BM25 .....	43
5.2.8	Penentuan <i>Query</i> Baru .....	44
5.3	Implementasi Tampilan Antarmuka .....	45
BAB 6	HASIL DAN PEMBAHASAN .....	46
6.1	Pengujian <i>Precision@K</i> .....	46
6.1.1	Skenario 1 .....	46
6.1.2	Skenario 2 .....	47
6.1.3	Skenario 3 .....	48
6.1.4	Skenario 4 .....	50
6.2	Pembahasan Hasil Pengujian .....	51
BAB 7	PENUTUP .....	52
7.1	Kesimpulan .....	52
7.2	Saran .....	52
DAFTAR	REFERENSI .....	53
LAMPIRAN	.....	54



## DAFTAR TABEL

Tabel 2.1 Kajian pustaka .....	7
Tabel 4.1 Dataset .....	29
Tabel 4.2 Manualisasi Case Folding .....	30
Tabel 4.3 Manualisasi Tokenization .....	31
Tabel 4.4 Manualisasi Filtering.....	33
Tabel 4.5 Manualisasi Stemming .....	34
Tabel 4.6 Hasil Perhitungan TF IDF .....	34
Tabel 4.7 Hasil Perhitungan BM25.....	35
Tabel 4.8 Hasil Pengurutan Dokumen.....	35
Tabel 4.9 Hasil Penentuan Query Baru .....	36
Tabel 4.10 Hasil Rekomendasi Kata Query baru .....	37
Tabel 4.11 Perancangan Pengujian Penerapan Query Expansion .....	39
Tabel 6.1 Hasil Pengujian Skenario 1 .....	46
Tabel 6.2 Hasil Pengujian Skenario 2 .....	47
Tabel 6.3 Hasil Pengujian Skenario 3 .....	49
Tabel 6.4 Hasil Pengujian Skenario 4 .....	50



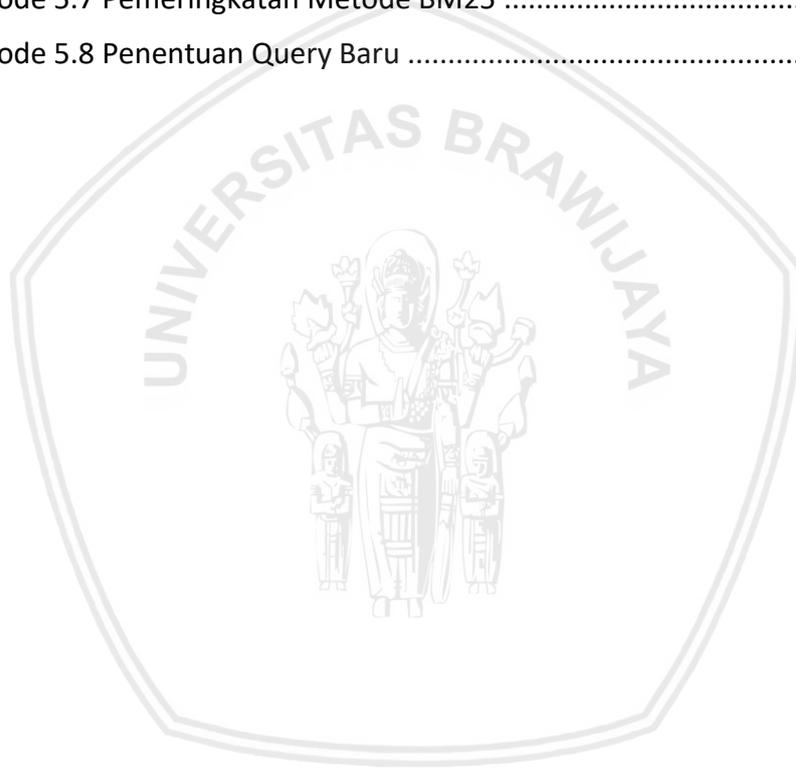
## DAFTAR GAMBAR

Gambar 2.1 Case Folding .....	10
Gambar 2.2 Filtering .....	10
Gambar 2.3 Tokenization .....	10
Gambar 2.4 Stemming .....	11
Gambar 4.1 Diagram Alir Sistem .....	18
Gambar 4.2 Case Folding .....	19
Gambar 4.3 Filtering .....	20
Gambar 4.4 Tokenization .....	21
Gambar 4.5 Stemming .....	23
Gambar 4.6 Perhitungan TF .....	24
Gambar 4.7 Perhitungan IDF .....	25
Gambar 4.8 Pemeringkatan dengan Metode BM25 .....	27
Gambar 4.9 Penentuan query baru .....	28
Gambar 5.1 Tampilan Antarmuka .....	45
Gambar 6.1 Grafik Skenario 1 .....	47
Gambar 6.2 Grafik Skenario 2 .....	48
Gambar 6.3 Grafik Skenario 3 .....	49
Gambar 6.4 Grafik Skenario 4 .....	51



## DAFTAR SOURCE CODE

Source Code 5.1 Case Folding .....	41
Source Code 5.2 Filtering .....	41
Source Code 5.3 Kode Program Tokenization.....	42
Source Code 5.4 Stemming .....	42
Source Code 5.5 Perhitungan Nilai TF.....	43
Source Code 5.6 Perhitungan Nilai IDF .....	43
Source Code 5.7 Pemeringkatan Metode BM25 .....	44
Source Code 5.8 Penentuan Query Baru .....	45



## DAFTAR LAMPIRAN

Lampiran A.1 Data Uji .....	54
Lampiran A.2 Pengujian .....	55
Lampiran A.3 Pengujian Dengan Pakar .....	75



## BAB 1 PENDAHULUAN

Bab 1 menjelaskan tentang pendahuluan yang berisi latar belakang penulis melakukan penelitian mengenai *query expansion* pada sistem temu kembali informasi pencarian jurnal berbahasa Indonesia menggunakan metode BM25, rumusan masalah, tujuan, manfaat, batasan masalah, sistematika pembahasan.

### 1.1 Latar Belakang

Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya merupakan kumpulan tulisan mahasiswa/i Fakultas Ilmu komputer Universitas Brawijaya yang berupa karya ilmiah dari penelitian yang telah dilakukan. J-PTIHK Universitas Brawijaya memiliki 3 vol arsip jurnal, vol 1 diterbitkan pada tahun 2017, vol 2 diterbitkan pada tahun 2018, dan vol 3 diterbitkan pada tahun 2019, setiap vol bernomor hingga 12. Pada setiap jurnal terdapat bagian abstrak yang merupakan ringkasan dari keseluruhan isi dokumen karya ilmiah, yang mengandung konsep, pernyataan masalah, pendekatan, dan kesimpulan yang dirangkai sehingga dapat menggambarkan keseluruhan isi tulisan, dengan panjang pada umumnya sebanyak 200 kata, tanpa menghilangkan informasi yang ada (Nasution, 2017).

Jurnal yang tersedia pada J-PTIHK berjumlah sangat banyak, maka diperlukan suatu sistem pencarian dokumen untuk mempermudah pengguna dalam menemukan dokumen jurnal, yaitu sistem pencarian yang menerapkan Sistem Temu Kembali Informasi (STKI) atau *information Retrieval* (IR). Pada penelitian sebelumnya terkait STKI oleh Irmawati (2017) dengan objek 5 link dokumen *online* yang didapatkan dari proses pencarian pada [www.google.com](http://www.google.com), dengan menggunakan metode *Vector Space Model*. Hasil dari penelitian tersebut adalah rata-rata nilai *recall* hampir 100% dan nilai rata-rata *precision* sekitar 73.6%.

Sistem Temu Kembali Informasi mengembalikan dokumen sesuai dengan *query* yang telah diinputkan pada sistem oleh pengguna, namun tidak semua dokumen yang dikembalikan relevan dan sesuai dengan kebutuhan informasi pengguna. Faktanya, *query* yang dimasukkan pengguna ke dalam sistem juga berupa kata yang tidak jelas atau tidak spesifik, dan pengguna sering mengalami kesulitan untuk mendeskripsikan informasi yang sebenarnya dibutuhkan ke dalam bentuk kata. Untuk mengurangi hasil pengembalian dokumen yang tidak relevan dan untuk melakukan perbaikan *query*, maka dapat diterapkan teknik perbaikan *query* pada sistem temu kembali informasi yaitu teknik *query expansion*. Penelitian terkait *query expansion* telah

dilakukan oleh Tanjung (2017). Pada penelitian Tanjung (2017) dengan objek *tweet* pada Twitter Kompas dan Detik, dengan menerapkan metode *query expansion* berbasis apriori dengan *Fuzzy K-Nearest Neighbour*, hasil yang didapatkan adalah nilai akurasi sebesar 76% jika menggunakan *preprocessing* dan akurasi sebesar 82% jika tanpa menggunakan *preprocessing*. Selain itu juga didapatkan nilai akurasi sebesar 76% jika menggunakan *query expansion* dan nilai akurasi sebesar 58% jika tanpa menggunakan *query expansion*.

Penerapan *query expansion* dapat dilakukan dengan menggunakan metode BM25 (*Best Matching*). BM25 adalah metode pada sistem temu kembali informasi yang digunakan untuk pengurutan hasil relevansi dokumen terhadap *query* yang ingin dicari (Manning, Raghavan, & Schutze, 2009). Penelitian mengenai metode BM25 telah dilakukan oleh Pardede (2015) dengan objek pada penelitian ini berupa dokumen berbahasa Indonesia yang terdapat pada komputer dengan format \*.doc, \*.txt, \*.docx, dan \*.pdf. Sedangkan metode yang digunakan adalah metode BM25 dan PLSA, dengan hasil penelitian yang didapat berupa nilai rata-rata *f-measure* pada metode BM25 adalah 61.649, sedangkan pada metode PLSA adalah 56.8877.

Berdasarkan permasalahan yang telah dijelaskan, maka penulis akan mengembangkan sebuah sistem temu kembali informasi untuk dokumen jurnal pada J-PTIHK yang memiliki tingkat relevansi tinggi berdasarkan *query* yang dimasukkan, mengingat sistem pencarian pada J-PTIHK belum menerapkan teknik *query expansion* dan pengembalian dokumen hanya dengan cara pencocokan *query* dengan kata yang terdapat pada dokumen dan tidak ada pemeringkatan dokumen relevan. Pengembangan sistem tersebut dengan cara menerapkan *query expansion* pada sistem temu kembali informasi menggunakan metode BM25, yang terdapat beberapa tahap, tahap yang pertama adalah melakukan proses *preprocessing* terhadap data latih dan *query*, *preprocessing* tersebut antara lain *case folding*, *tokenization*, *filtering*, dan *stemming*. Tahap selanjutnya dilakukan pembobotan setiap kata pada *query* dengan rumus TF IDF, lalu dilakukan pemeringkatan dokumen dengan menggunakan metode BM25. Selanjutnya setiap kata dari seluruh dokumen yang relevan dilakukan perhitungan bobot kata untuk menentukan pilihan kata yang digunakan untuk proses penambahan kata pada *query expansion*.

## 1.2 Rumusan Masalah

Rumusan permasalahan yang akan diselesaikan pada penelitian ini sesuai dengan penjelasan latar belakang masalah diatas adalah sebagai berikut.

1. Bagaimana menggunakan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia dengan metode BM25?
2. Bagaimana hasil menggunakan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia dengan metode BM25?

## 1.3 Tujuan

Tujuan dari penelitian penerapan *query expansion* untuk sistem temu kembali informasi dokumen jurnal berbahasa Indonesia menggunakan metode BM25 ini adalah sebagai berikut.

1. Menggunakan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia dengan metode BM25.
2. Menguji hasil *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia dengan metode BM25.

## 1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut.

1. Bagi penulis, menambah pengetahuan dalam menerapkan *query expansion* untuk sistem temu kembali informasi pencarian jurnal berbahasa Indonesia menggunakan metode BM25.
2. Bagi mahasiswa Fakultas Ilmu Komputer, memberikan hasil pencarian jurnal yang lebih relevan sesuai dengan informasi yang diinginkan.
3. Bagi Fakultas Ilmu Komputer, memberikan sistem temu kembali informasi pencarian jurnal berbahasa Indonesia yang lebih akurat.

## 1.5 Batasan Masalah

Untuk memfokuskan permasalahan dalam penelitian ini, maka permasalahan yang ada dibatasi sebagai berikut.

1. Dokumen yang digunakan pada penelitian ini adalah abstrak dari jurnal pada J-PTIUK Universitas Brawijaya, dari arsip jurnal vol 2 no 12 dan vol 3 no 1.
2. Abstrak yang digunakan adalah abstrak berbahasa Indonesia.
3. Dokumen yang digunakan sebanyak 350 dokumen.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan dalam pembuatan laporan dibuat untuk memberikan penjelasan dari bab-bab yang terdapat pada laporan ini, dengan sistematika sebagai berikut.

### BAB 1: PENDAHULUAN

Bab 1 menjelaskan tentang latar belakang dilakukannya penelitian terhadap objek dokumen Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTI IK) Universitas Brawijaya menggunakan penerapan *query expansion* dengan metode BM25 pada sistem temu kembali informasi, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah penelitian, dan sistematika pembahasan pada penelitian.

### BAB 2: LANDASAN KEPUSTAKAAN

Bab 2 menjelaskan tentang kajian pustaka yang berisi penelitian-penelitian sebelumnya terkait dengan penelitian ini, dan juga terdapat dasar teori yang berisi teori berkaitan dengan penelitian ini yaitu J-PTI IK, sistem temu kembali informasi, *query expansion*, dan metode BM25.

### BAB 3: METODOLOGI

Bab 3 menjelaskan tentang serangkaian langkah yang dilakukan untuk menyelesaikan permasalahan dalam objek dokumen J-PTI IK dengan penerapan *query expansion* menggunakan metode BM25 pada sistem temu kembali informasi.

### BAB 4: PERANCANGAN

Bab 4 menjelaskan tentang perancangan rancangan sistem, antara lain rancangan algoritma, rancangan pengujian, serta perhitungan manualisasi dalam penerapan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia menggunakan metode BM25.

### BAB 5: IMPLEMENTASI

Bab 5 menjelaskan tentang bagaimana penerapan metode *query expansion* pada sistem temu kembali informasi jurnal berbahasa Indonesia menggunakan metode BM25, berdasar pada perancangan yang telah dibuat.

### BAB 6: HASIL DAN PEMBAHASAN

Bab 6 menjelaskan tentang pembahasan dan hasil dari pengujian sistem berdasarkan hasil implementasi yang telah dilakukan terhadap hasil *query expansion* pada sistem temu kembali dokumen jurnal bahasa Indonesia menggunakan metode BM25.

## BAB 7: PENUTUP

Bab 7 menjelaskan tentang penutup laporan yang berupa kesimpulan terhadap akurasi dan hasil sistem dari penelitian yang telah dilakukan beserta dengan saran-saran untuk pengembangan sistem yang lebih lanjut.



## BAB 2 LANDASAN KEPUSTAKAAN

Bab 2 menjelaskan tentang kajian pustaka yang berisi penelitian-penelitian terdahulu terkait dengan penelitian ini, dan juga terdapat dasar teori mengenai Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, sistem temu kembali informasi, *query expansion*, dan metode BM25.

### 2.1 Kajian Pustaka

Pada sub bab ini berisi penelitian terdahulu terkait dengan objek dan metode yang digunakan pada penelitian ini, yang bertujuan untuk mendukung penelitian yang diusulkan. Penelitian terdahulu terkait Sistem temu Kembali Informasi telah dilakukan oleh (Pertiwi, 2017), dan (Irmawati, 2017). Pada penelitian yang pertama, peneliti melakukan penelitian terhadap pengembangan sistem temu kembali informasi terhadap objek makalah penelitian yang ada pada <http://ejournal.bsi.ac.id/ejurnal>, dengan menggunakan metode *Latent Semantic Indexing* (LSI). Berdasarkan hasil pengujian yang dilakukan pada *website* <http://ejournal.bsi.ac.id/ejurnal> dengan menggunakan 10 kata kunci, menghasilkan nilai rata-rata presisi sebesar 1.53% dan nilai rata-rata *recall* sebesar 10.29% (Pertiwi, 2017).

Pada penelitian kedua terkait sistem temu kembali informasi dilakukan dengan objek 5 *link* dokumen *online* yang didapatkan dari proses pencarian pada [www.google.com](http://www.google.com), dengan *size* yang berbeda-beda pada setiap objeknya. Pada penelitian ini menggunakan metode *Vector Space Model*. Tujuan dari penelitian ini adalah mempermudah pengguna dalam mendapatkan dokumen sesuai dengan kebutuhan informasinya. Hasil dari penelitian tersebut adalah rata-rata nilai *recall* hampir 100% dan nilai rata-rata *precision* sekitar 73.6%, yang berarti 73.6% dokumen yang berhasil dikembalikan adalah dokumen relevan dengan kata kuncinya (Irmawati, 2017).

Di sisi lain juga terdapat penelitian terkait *query expansion* yang telah dilakukan sebelumnya oleh (Haryanto, 2018), dan (Tanjung, 2017). Pada penelitian pertama, penerapan *query expansion* dilakukan dengan menggunakan metode *Support Vector Machine*, dengan objek penelitian adalah 400 data dari situs [tokopedia.com](http://tokopedia.com), yang terdiri dari 200 komentar positif, dan 200 komentar negatif. Pada penelitian ini *query expansion* dilakukan untuk menganalisis sentimen dari komentar yang ada pada situs [tokopedia.com](http://tokopedia.com). Hasil dari penelitian adalah nilai rata-rata akurasi sebesar 96.25% dengan menggunakan SVM dengan *query expansion*, sedangkan jika

tanpa *query expansion* nilai rata-rata akurasi sebesar 94.75% (Haryanto, 2018).

Penelitian selanjutnya terkait *query expansion* juga dilakukan dengan objek *tweet* pada Twitter Kompas dan detik. Metode yang digunakan adalah *query expansion* berbasis apriori dengan *Fuzzy K-Nearest Neighbour*, dengan hasil yang didapatkan adalah nilai akurasi sebesar 76% jika menggunakan *preprocessing* dan akurasi sebesar 82% jika tanpa menggunakan *preprocessing*, maka dapat disimpulkan bahwa proses *preprocessing* tidak memberikan pengaruh signifikan pada proses klasifikasi teks. Selain itu juga didapatkan nilai akurasi sebesar 76% jika menggunakan *query expansion* dan nilai akurasi sebesar 58% jika tanpa menggunakan *query expansion*, dari hasil tersebut maka dapat disimpulkan bahwa penggunaan *query expansion* dapat meningkatkan hasil klasifikasi menjadi lebih baik (Tanjung, 2017).

Penelitian terkait metode BM25 telah dilakukan sebelumnya oleh (Pardede, 2015). Peneliti melakukan penelitian terhadap metode BM25 yang diimplementasikan pada pembobotan dokumen yang diteliti. Objek yang digunakan adalah dokumen berbahasa Indonesia yang terdapat pada komputer dengan format \*.doc, \*.txt, \*.docx, dan \*.pdf. Sedangkan metode yang digunakan adalah metode BM25 dan PLSA, dengan hasil penelitian yang didapat berupa nilai rata-rata *F-measure* pada metode BM25 adalah 61.649, sedangkan pada metode PLSA adalah 56.8877. Kesimpulan yang dapat diambil adalah metode BM25 lebih efektif dalam melakukan pemeringkatan dokumen dibandingkan dengan metode PLSA.

Perbandingan penelitian ini dengan penelitian terdahulu yang telah dijelaskan sebelumnya adalah studi kasus yang diangkat. Dalam penelitian ini studi kasus yang diangkat mengenai sistem temu kembali informasi dokumen jurnal pada JPTIIK Universitas Brawijaya. Selain itu, dalam penelitian ini sistem menggunakan metode BM25 sebagai metode pemeringkatan hasil dokumen yang relevan untuk penerapan *query expansion* terhadap dokumen jurnal pada JPTIIK Universitas Brawijaya, yang bertujuan untuk meningkatkan hasil relevansi pencarian dokumen dengan cara menambahkan kata dalam *query*. Untuk memahami semua paparan kajian pustaka yang telah dijelaskan sebelumnya dengan lebih jelas maka dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kajian pustaka

No.	Peneliti	Judul	Objek	Metode	Hasil
1	(Pardede, 2015)	Implementasi dan Perbandingan Metode Okapi BM25 dan PLSA pada Aplikasi <i>Information Retrieval</i> .	Dokumen berbahasa Indonesia yang terdapat pada komputer yang memiliki format *.doc, *.txt, *.docx, dan *.pdf.	Metode Okapi B25 dan PLSA pada <i>Information Retrieval</i> .	Hasil dari penelitian perbandingan metode Okapi BM25 dengan metode PLSA tersebut adalah metode BM25 lebih efektif dalam melakukan peerangkingan dokumen, nilai rata-rata <i>F-Measure</i> pada metode BM25 adalah 61.649, sedangkan pada metode PLSA adalah 56.8877.
2	(Irmawati, 2017)	Sistem Temu Kembali Informasi pada Dokumen Dengan metode <i>Vector Space Model</i> .	5 link dokumen <i>online</i> yang didapatkan dari proses pencarian pada <a href="http://www.google.com">www.google.com</a> .	Metode <i>Vector Space Model</i> pada sistem temu kembali informasi.	Hasil dari penelitian pada sistem temu kembali informasi menggunakan metode VSM ini adalah rerata nilai <i>recall</i> hampir 100%, dan rerata nilai <i>precision</i> sekitar 73.6%.
3	(Pertwi, 2017)	Sistem Temu Kembali Informasi Dalam Dokumen (Pencarian 10 Kata Kunci di Ejournal BSI).	Makalah-makalah penelitian yang ada pada <a href="http://ejournal.bsi.ac.id/ejurnal">http://ejournal.bsi.ac.id/ejurnal</a> .	Metode <i>Latent Semantic Indexing</i> (LSI).	Hasil dari penelitian metode LSI pada sistem temu kembali informasi tersebut adalah nilai rerata presisi 1.53% dan nilai rerata <i>recall</i> 10.29%.

No.	Peneliti	Judul	Objek	Metode	Hasil
4	(Tanjung, 2017)	Klasifikasi <i>Tweet</i> Pada Twitter Dengan Menggunakan Metode <i>Fuzzy K-Nearest Neighbour (Fuzzy K-NN)</i> dan <i>Query Expansion</i> Berbasis Apriori.	<i>Tweet</i> pada Twitter <i>kompas</i> dan <i>detik</i> .	<i>Query Expansion</i> Berbasis Apriori dengan <i>Fuzzy K-Nearest Neighbour (Fuzzy K-NN)</i>	Nilai akurasi sebesar 76% jika menggunakan <i>preprocessing</i> , dan akurasi sebesar 82% jika tanpa menggunakan <i>preprocessing</i> , selain itu juga didapat nilai akurasi sebesar 76% jika menggunakan <i>query expansion</i> , dan nilai akurasi sebesar 58% jika tanpa menggunakan <i>query expansion</i> .
5	(Haryanto, 2018)	Analisis Sentimen <i>Review Barang</i> Berbahasa Indonesia Dengan Metode <i>Support Vector Machine</i> dan <i>Query Expansion</i> .	400 data dari situs <i>tokopedia.com</i> , dengan data komentar positif, dan data komentar negatif masing-masing berjumlah 200.	<i>Query Expansion</i> dengan metode <i>Support Vector Machine (SVM)</i>	Hasil dari penelitian untuk menganalisis sentimen tersebut adalah nilai rata-rata akurasi 96.25% dengan menggunakan metode SVM dan <i>Query Expansion</i> , sedangkan jika tanpa <i>query expansion</i> nilai rata-rata akurasi 94.75%.

## 2.2 Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer

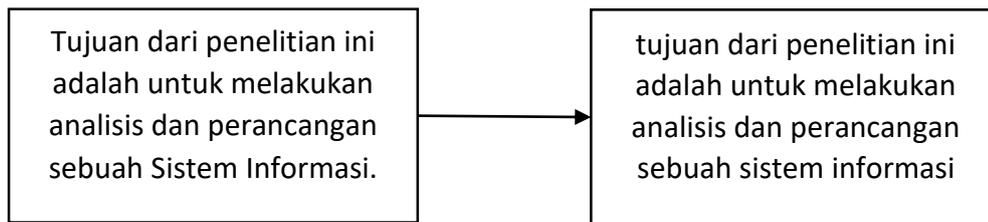
Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya merupakan kumpulan tulisan mahasiswa/i Fakultas Ilmu komputer Universitas Brawijaya yang berupa karya ilmiah dari penelitian yang telah dilakukan, yang berasal dari berbagai macam program studi yang terdapat pada Fakultas Ilmu Komputer, diantaranya Magister Ilmu Komputer, Teknik Informatika, Sistem Komputer, Sistem Informasi, Teknologi Informasi dan Pendidikan Teknologi Informasi. J-PTIHK Universitas Brawijaya memiliki 3 vol arsip jurnal, vol 1 diterbitkan pada tahun 2017, vol 2 diterbitkan pada tahun 2018, dan vol 3 diterbitkan pada tahun 2019. J-PTIHK dapat diakses melalui *web browser* yang beralamatkan *j-ptiik.ub.ac.id*. Jurnal yang terdapat pada J-PTIHK bertujuan untuk memberikan kontribusi terhadap sumber daya penelitian di bidang ilmu komputer dan teknologi informasi.

## 2.3 Sistem Temu Kembali Informasi

Sistem Temu Kembali Informasi adalah sistem yang bertujuan untuk menemukan dokumen yang biasanya berupa teks dari sekumpulan data yang tidak terstruktur, guna memenuhi kebutuhan informasi yang diperlukan pada koleksi sekumpulan data yang besar yang biasanya disimpan di komputer (Manning, Raghavan, & Schutze, 2009). Sistem Temu Kembali Informasi mengembalikan dokumen sesuai dengan *query* yang telah dimasukkan pada sistem oleh pengguna dalam bentuk peringkat, peringkat paling tinggi merupakan dokumen yang paling relevan terhadap *query* masukkan pengguna. Fokus pada sistem temu kembali informasi adalah mengembalikan dokumen yang relevan sebanyak-banyaknya sekaligus mengembalikan dokumen yang tidak relevan sesedikit mungkin. Efektifitas dan efisiensi sistem temu kembali informasi dipengaruhi oleh kualitas *index* yang dihasilkan pada tahapan *preprocessing*, yang meliputi tahapan *case folding*, *tokenization*, *filtering*, dan *stemming* (Sanjaya, 2017).

### 2.3.1 Case Folding

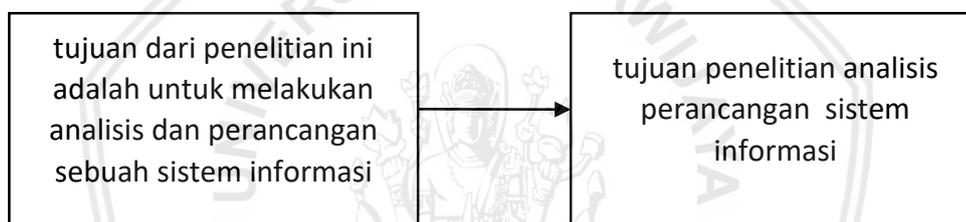
*Case folding* adalah tahap untuk mengganti huruf besar dari setiap kata menjadi huruf kecil (*lower case*). Sehingga setiap huruf dari setiap kata dalam kalimat menjadi konsisten (Sanjaya, 2017). Pada Gambar 2.1 berikut menunjukkan contoh dari tahapan *case folding*.



Gambar 2.1 Case Folding

### 2.3.2 Filtering

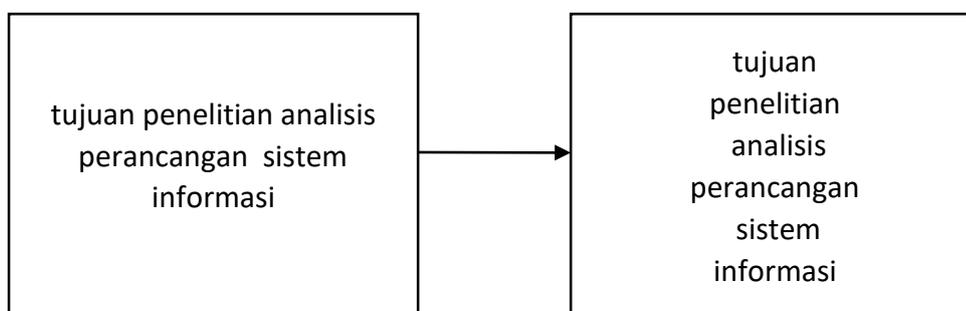
*Filtering* yaitu tahap untuk menghilangkan semua *stop words* yaitu kata yang terlalu sering muncul dan tidak memiliki makna, seperti contohnya yang, adalah, hanya, namun, ini dan lain-lain (Sanjaya, 2017). *Stopword list* yang digunakan dalam penelitian ini adalah *stopword list* Sastrawi. Pada Gambar 2.3 berikut menunjukkan contoh dari tahapan *filtering*.



Gambar 2.2 Filtering

### 2.3.3 Tokenization

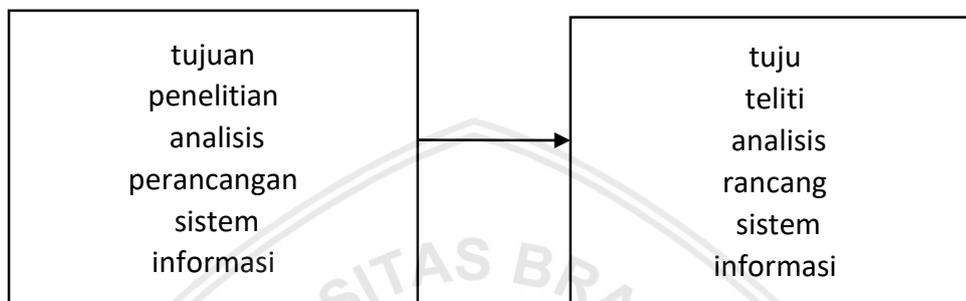
*Tokenization* adalah tahapan pada *preprocessing* untuk memecah kalimat menjadi satuan kata per kata, yang berdasarkan spasi sebagai pemisah. Pada Gambar 2.2 berikut menunjukkan contoh dari tahapan *tokenization*.



Gambar 2.3 Tokenization

### 2.3.4 Stemming

*Stemming* digunakan untuk menghasilkan kata dasar dengan cara menghilangkan semua imbuhan, sesuai dengan struktur morfologi bahasa Indonesia (Sanjaya, 2017). Pada penelitian ini menggunakan *stemming* Sastrawi yang menggunakan algoritma berbasis Nazief dan Adriani. Pada Gambar 2.4 berikut menunjukkan contoh dari tahapan *stemming*.



Gambar 2.4 Stemming

## 2.4 Query Expansion

*Query expansion* adalah suatu teknik menyusun ulang *query* dengan menambahkan kata pada *query* sistem temu kembali informasi yang tergantung pada *query* awal pengguna, guna meningkatkan jumlah dokumen relevan yang dikembalikan (Campos, 2013). Pada Sistem Temu Kembali Informasi akan mengembalikan seluruh dokumen sesuai dengan *query* yang telah dimasukkan pada sistem oleh pengguna, namun tidak semua dokumen yang dikembalikan relevan dan sesuai dengan kebutuhan informasi pengguna. Faktanya, *query* yang dimasukkan pengguna ke dalam sistem juga berupa kata yang tidak jelas atau tidak spesifik, dan pengguna sering mengalami kesulitan untuk mendeskripsikan informasi yang sebenarnya dibutuhkan ke dalam bentuk kata.

Untuk mengurangi hasil pengembalian dokumen yang tidak relevan atau tidak sesuai dengan kebutuhan informasi pengguna, dan untuk melakukan perbaikan *query* agar sesuai dengan informasi yang sebenarnya dibutuhkan, maka dapat diterapkan teknik *query expansion*. Terdapat 3 teknik pada *query expansion* (Selberg, 1997) :

1. *Manual Query Expansion* (MQE)

Teknik *query expansion* dimana pengguna secara manual memperbaiki *query* sendiri.

## 2. Automatic Query Expansion (AQE)

Teknik *query expansion* dimana proses memperbaiki *query* dilakukan secara otomatis tanpa kontrol dari pengguna. Seperti contohnya, sistem yang menambahkan istilah *thesaurus* pada *query*.

## 3. Interactive Query Expansion (IQE)

Teknik *query expansion* dimana pengguna dapat berinteraksi dengan sistem saat melakukan proses *query expansion*.

Rumus perhitungan yang digunakan untuk perbaikan *query* dengan cara menentukan beberapa pilihan kata untuk menambah *query* dapat dilihat pada persamaan 2.1 (Russell & Norvig, 2010).

$$OW_i = \frac{r_i}{R} \times w_i \quad (2.1)$$

Keterangan:

$r_i$  : Jumlah dokumen relevan yang terdapat kata  $i$

$R$  : Jumlah dokumen yang relevan

$w_i$  : Nilai bobot kata  $i$  yang dihitung dengan rumus IDF

## 2.5 Metode BM25

Metode BM25 (*Best Matching*) adalah metode pada sistem temu kembali informasi yang digunakan untuk pemeringkatan hasil relevansi dokumen terhadap *query* yang ingin dicari. BM25 merupakan metode yang efektif dan memiliki akurasi yang tinggi dalam mengurutkan hasil dokumen yang relevan (Manning, Raghavan, & Schutze, 2009). Persamaan 2.2, dan 2.3 merupakan persamaan metode BM25 (Russell & Norvig, 2010):

### 1. Persamaan rumus BM25

$$score(D, Q) = \sum_{i=1}^N IDF(q_i) \times \frac{f(q_i, D)(k_1+1)}{f(q_i, D) + k_1(1-b + b \frac{|D|}{avgdl})} \quad (2.2)$$

Keterangan:

$f(q_i, D)$  : Banyaknya  $q_i$  pada dokumen  $D$

$q_i$  : Kata yang dicari

$|D|$  : Panjang dokumen  $D$  yang dihitung per kata

$avgdl$  : Rata – rata panjang dokumen di dalam koleksi

$k_1$  : 1.2

$b$  : 0.75

## 2. Persamaan rumus IDF

$$IDF(q_i) = \log \left( \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \right) \quad (2.3)$$

Keterangan:

 $N$  : Total banyaknya dokumen di dalam koleksi $n(q_i)$  : Jumlah dokumen yang terdapat  $q_i$ **2.6 Evaluasi Sistem**

Evaluasi sistem pada penelitian ini menggunakan evaluasi untuk temu kembali berperingkat yaitu nilai  $Precision@K$ , yang bertujuan untuk menghitung nilai persentase pada dokumen teratas sebanyak K dokumen. Dokumen yang berada di bawah peringkat K maka akan diabaikan. Rumus untuk perhitungan nilai  $Precision@K$  dapat dilihat pada persamaan 2.4.

Rumus dari  $Precision@K$ 

$$Precision@K = \frac{r}{K} \quad (2.4)$$

Keterangan:

 $r$  : Jumlah dokumen relevan pada K dokumen teratas $K$  : Nilai *threshold* peringkat

## BAB 3 METODOLOGI

Bab 3 menjelaskan tentang serangkaian langkah dalam menyelesaikan permasalahan dengan objek dokumen J-PTIHK Universitas Brawijaya dengan menggunakan untuk *query expansion* dengan metode BM25 pada sistem temu kembali informasi pencarian jurnal berbahasa Indonesia. Bab 3 terdiri dari sub bab Tipe Penelitian, Strategi Penelitian, Lokasi Penelitian, Teknik Pengumpulan Data, Perancangan Algoritma, Teknik Penerapan Algoritma, Peralatan Pendukung, dan Teknik Analisis Data.

### 3.1 Tipe Penelitian

Tipe penelitian yang diterapkan pada penelitian ini adalah penelitian non implementatif analitik. Menurut buku panduan skripsi Fakultas Ilmu Komputer Universitas Brawijaya tahun 2017, tipe penelitian non implementatif adalah penelitian yang berfokus pada pengamatan terhadap fenomena atau situasi yang kemudian menghasilkan hasil pengamatan sebagai produk utamanya. Pada penelitian ini dilakukan pendekatan secara analitik, yaitu sebuah kegiatan penelitian tipe non implementatif yang bertujuan untuk menjelaskan hubungan antar elemen pada fenomena atau situasi yang sedang diamati, dengan produk utamanya adalah hasil analisis.

### 3.2 Strategi Penelitian

Strategi penelitian yang diterapkan adalah strategi penelitian deskriptif kualitatif, karena pada penelitian ini bertujuan untuk mengembangkan metode yang ada yaitu *query expansion* dengan menggunakan metode BM25, melalui data yang telah diperoleh yaitu dokumen jurnal pada JPTIHK Universitas Brawijaya. Penelitian kualitatif yang dilakukan merupakan penelitian kualitatif dengan strategi non interaktif (penelitian analisis) yang merupakan penelitian berdasarkan analisis dari data yang telah dikumpulkan

### 3.3 Lokasi Penelitian

Lokasi penelitian untuk penerapan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia menggunakan metode BM25 adalah Laboratorium Kecerdasan Buatan Fakultas Ilmu Komputer Universitas Brawijaya.

### 3.4 Teknik Pengumpulan Data

Teknik pengumpulan data pada penelitian ini menggunakan jenis data primer, yang diperoleh peneliti secara langsung dengan mengambil jurnal

pada arsip jurnal vol 2, dan vol 3, sebanyak 350 dokumen yang bersumber dari J-PTIHK Universitas Brawijaya, yang beralamatkan *j-ptiik.ub.ac.id*.

### 3.5 Perancangan Algoritma

Dalam proses penerapan metode BM25 pada *query expansion* terdapat beberapa tahapan, antara lain:

1. Data dikumpulkan menjadi satu dan disimpan pada database.
2. Data dilakukan proses *preprocessing* yang meliputi *case folding*, *tokenization*, *Filtering*, dan *Stemming*.
  - a. *Case folding* adalah tahap untuk merubah huruf besar dari setiap kata menjadi huruf kecil dan menghilangkan semua karakter selain huruf.
  - b. *Tokenization* adalah tahap untuk memecah kalimat menjadi satuan kata per kata berdasarkan spasi sebagai pemisah.
  - c. *Filtering* adalah tahap untuk menghilangkan semua *stop words* yaitu kata yang terlalu sering muncul dan tidak memiliki makna.
  - d. *Stemming* adalah tahap untuk menghilangkan semua imbuhan, dan menghasilkan kata dasar.
3. Dilakukan pembobotan setiap kata dengan menggunakan rumus TF IDF.
4. Menerapkan metode BM25 untuk pemeringkatan terhadap setiap dokumen sesuai dengan *query* masukkan dari pengguna.
5. Perhitungan untuk penentuan *query* baru berdasarkan dokumen hasil pemeringkatan.
6. Mengembalikan beberapa dokumen dengan peringkat tertinggi dan beberapa pilihan kata untuk proses perbaikan *query*.

### 3.6 Teknik Penerapan Algoritma

Teknik penereapan algoritma dilakukan dengan mengacu pada pengumpulan data dan perancangan algoritma yang telah dibuat. Teknik penerapan algoritma adalah sebagai berikut.

1. Implementasi database dengan menggunakan phpMyAdmin dan XAMPP pada server.
2. Implementasi penerapan *query expansion* dengan metode BM25 dilakukan dengan menggunakan bahasa Python.

### 3.7 Peralatan Pendukung

Kebutuhan peralatan pendukung yang digunakan untuk menunjang penelitian ini adalah sebagai berikut.

1. Kebutuhan perangkat keras, meliputi:
  - a. Laptop dengan *processor* intel Core i5.
  - b. Laptop dengan *memory* 4GB RAM.
2. Kebutuhan perangkat lunak, meliputi:
  - a. Sistem operasi *Windows* 10.
  - b. Spyder (Python 3.6).
  - c. XAMPP Control Panel v3.2.2.
  - d. phpMyAdmin.
3. Kebutuhan data, yang berupa jurnal diambil dari *website* Jurnal J-PTIIK Universitas Brawijaya, yang beralamatkan *j-ptiik.ub.ac.id*.

### 3.8 Teknik Analisis Data

Pengujian sistem dilakukan untuk menunjukkan apakah fungsi sistem dapat berjalan sesuai dengan yang dirancang saat beroperasi, selain itu juga bertujuan untuk menguji apakah keluaran yang dihasilkan sesuai dengan yang diinginkan pengguna. Pengujian sistem dilakukan dengan melihat hasil nilai *Precision@K*.

## BAB 4 PERANCANGAN

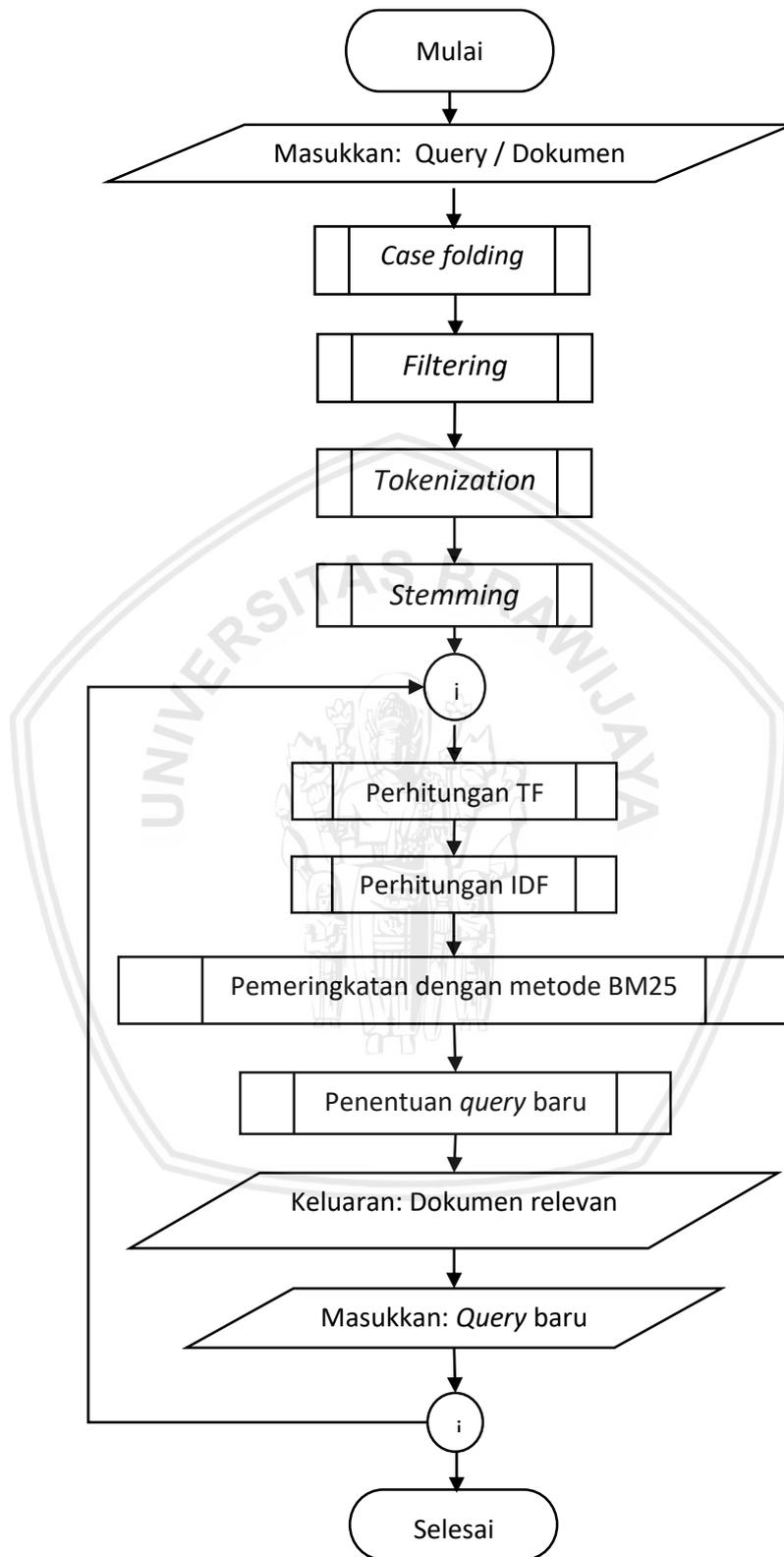
Bab 4 menjelaskan tentang perancangan rancangan sistem, antara lain rancangan algoritma, rancangan pengujian, perhitungan manualisasi, serta rancangan tampilan antarmuka dalam penerapan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia menggunakan metode BM25.

### 4.1 Deskripsi Umum Sistem

Pada penelitian ini sistem yang dikembangkan adalah sebuah perangkat lunak yang bertujuan untuk memudahkan pengguna dalam melakukan pencarian terhadap dokumen JPTIIK Universitas Brawijaya, berdasarkan abstrak berbahasa Indonesia, dengan menerapkan *query expansion* menggunakan metode BM25. Data yang akan digunakan sebagai data latih pada sistem ini sebanyak 350 dokumen jurnal. Masukkan dari pengguna berupa *query* yang akan menjadi data uji pada sistem. Data uji dan data latih akan dilakukan proses *preprocessing*, dan selanjutnya akan dilakukan proses pembobotan setiap kata pada *query* menggunakan rumus TF IDF, lalu dilakukan tahapan pemeringkatan dokumen menggunakan metode BM25 untuk mengembalikan dokumen yang relevan. Lalu dari dokumen yang relevan akan dilakukan perhitungan untuk menentukan *query* baru. *Query* baru yang dihasilkan ini diharapkan dapat meningkatkan nilai relevansi pada sistem temu kembali dokumen jurnal berbahasa Indonesia pada penelitian ini.

### 4.2 Diagram Alir Sistem

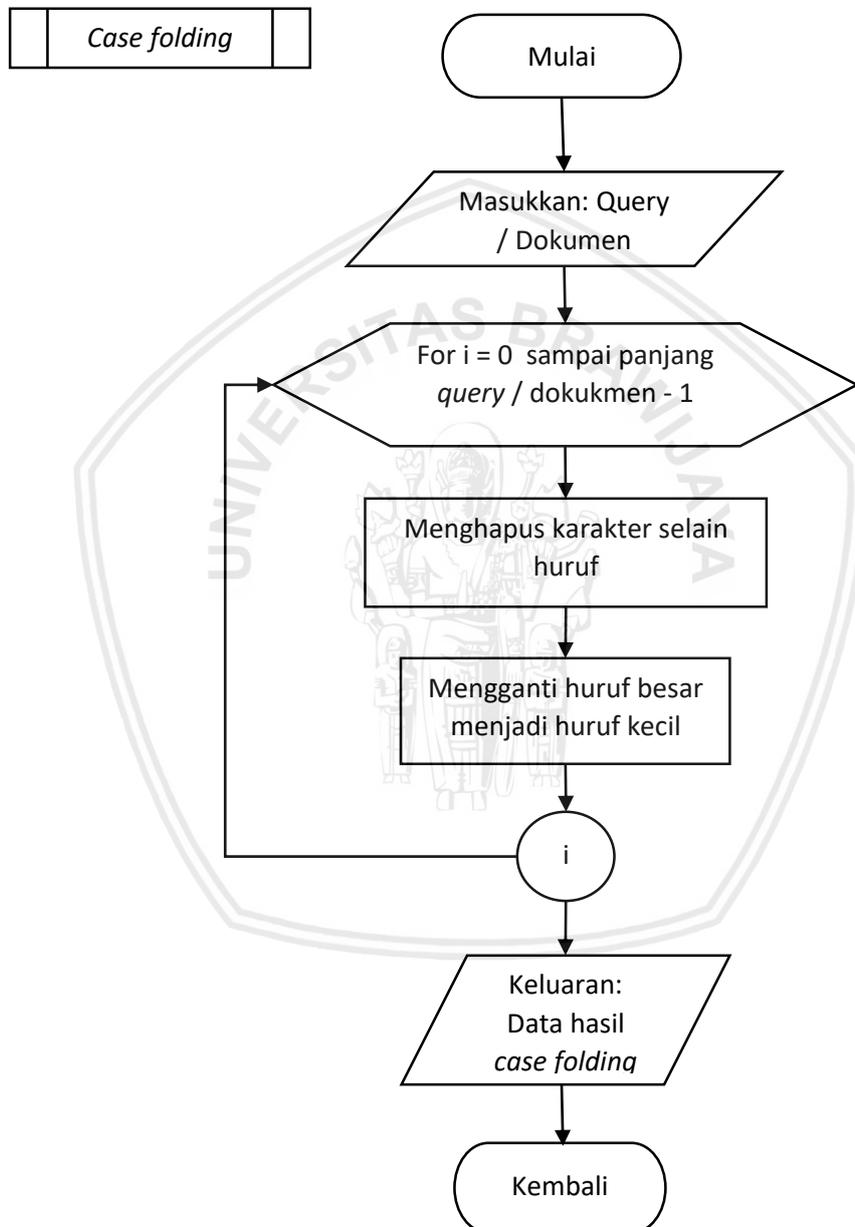
Diagram alir sistem digunakan untuk mempermudah dalam memahami alur kerja sistem dengan menggunakan simbol-simbol grafis. Diagram alir sistem pada penelitian ini dapat dilihat pada Gambar 4.1. Pertama dilakukan *preprocessing* yaitu *case folding*, *filtering*, *tokenization*, dan *stemming* pada masukkan yang berupa *query* atau dokumen, selanjutnya dilakukan pembobotan setiap kata pada *query* dengan rumus TF IDF, lalu dokumen akan di ranking menggunakan metode BM25 berdasarkan *query* yang dimasukkan, dari hasil pemeringkatan dokumen tersebut akan dilakukan proses penentuan *query* baru terhadap seluruh dokumen yang relevan, setelah *query* baru berhasil ditentukan maka *query* tersebut akan kembali lagi ke proses perhitungan TF IDF, hingga menghasilkan dokumen yang relevan sesuai dengan *query* baru.



Gambar 4.1 Diagram Alir Sistem

### 4.2.1 Case Folding

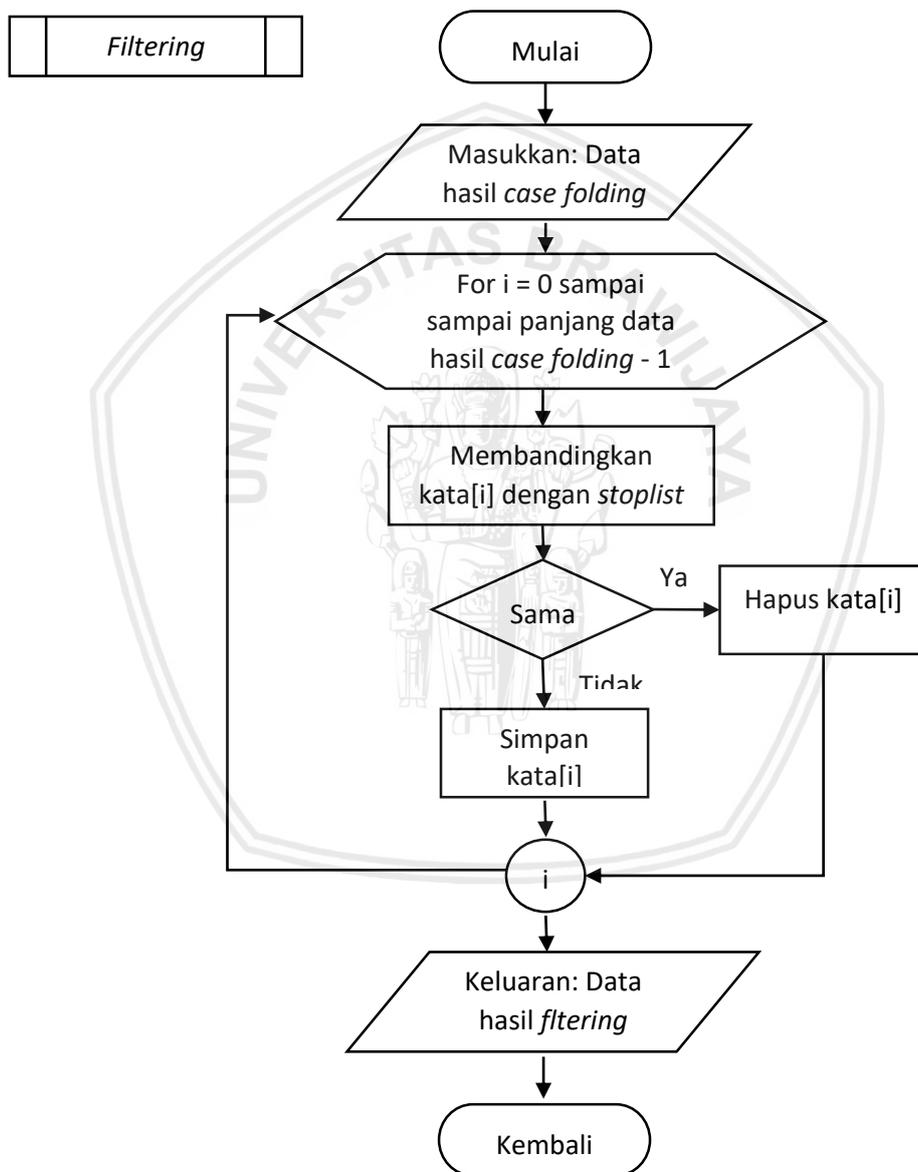
Proses *case folding* adalah tahapan untuk mengganti semua huruf besar menjadi huruf kecil (*lower case*) pada setiap kata. Sehingga setiap huruf dari setiap kata dalam kalimat menjadi konsisten (Sanjaya, 2017). Diagram alir pada proses *case folding* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Case Folding

### 4.2.2 Filtering

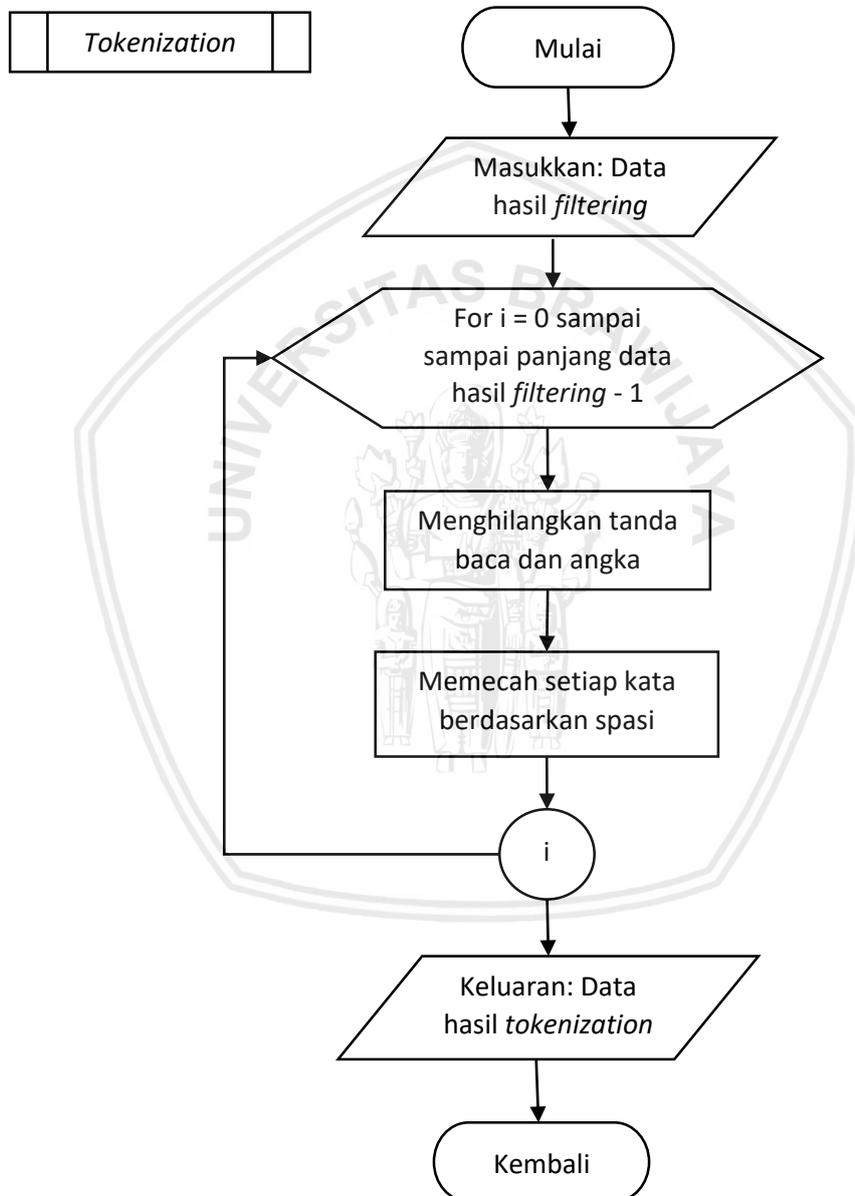
Proses *filtering* adalah tahap untuk menghilangkan semua *stop words* yaitu kata yang terlalu sering muncul dan tidak memiliki makna, seperti contohnya yang, adalah, hanya, namun, ini dan lain-lain (Sanjaya, 2017). *Stopword list* yang digunakan pada penelitian ini adalah *stopword list* Sastrawi. Data yang diproses pada tahapan *filtering* adalah data hasil *case folding*. Diagram alir pada proses *filtering* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Filtering

### 4.2.3 Tokenization

*Tokenization* adalah proses memecah setiap kalimat menjadi satuan kata berdasarkan spasi sebagai pemisah dan menghilangkan semua tanda baca, angka, dan karakter lainnya selain huruf. Data yang akan diproses pada tahapan ini adalah data hasil *filtering*. Diagram alir pada proses *tokenization* dapat dilihat pada Gambar 4.4.

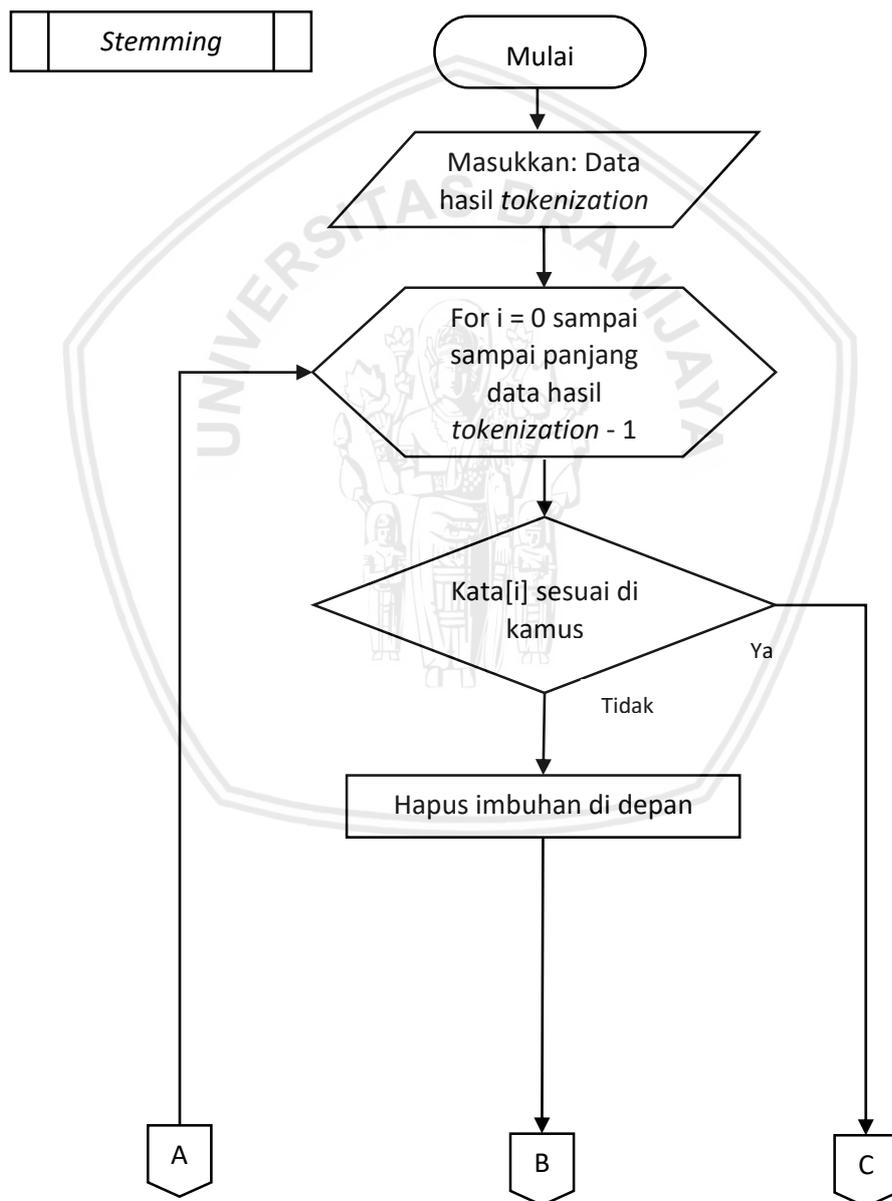


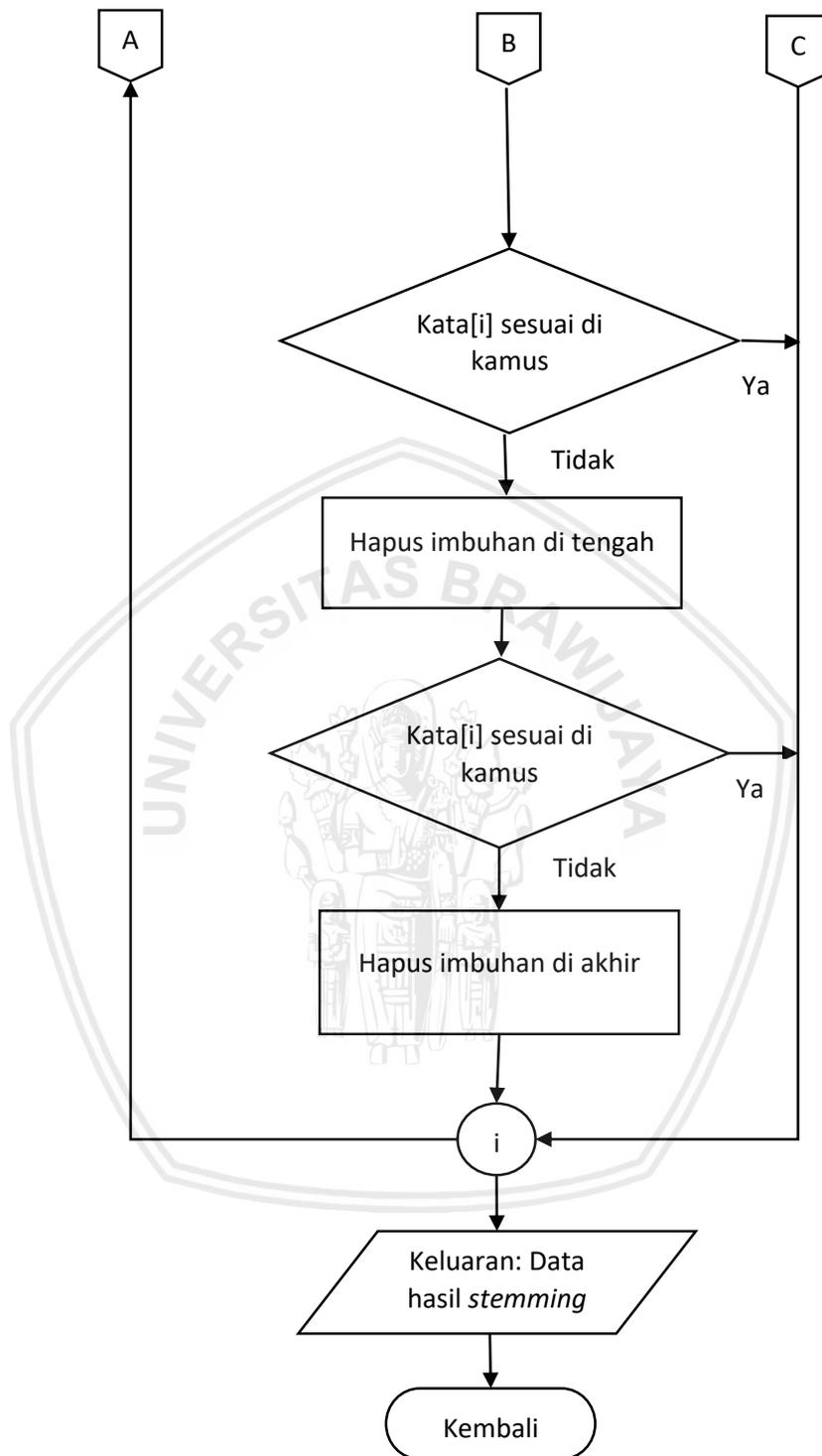
Gambar 4.4 *Tokenization*

#### 4.2.4 Stemming

Proses *stemming* yaitu tahap untuk menghasilkan kata dasar dengan cara menghilangkan semua imbuhan, yang sesuai dengan struktur morfologi bahasa Indonesia (Sanjaya, 2017).

Pada penelitian ini menggunakan *stemming* Sastrawi yang menggunakan algoritma berbasis Nazief dan Adriani. Data yang diproses pada tahapan *stemming* adalah data hasil *tokenization*. Diagram alir pada proses *stemming* dapat dilihat pada Gambar 4.5.

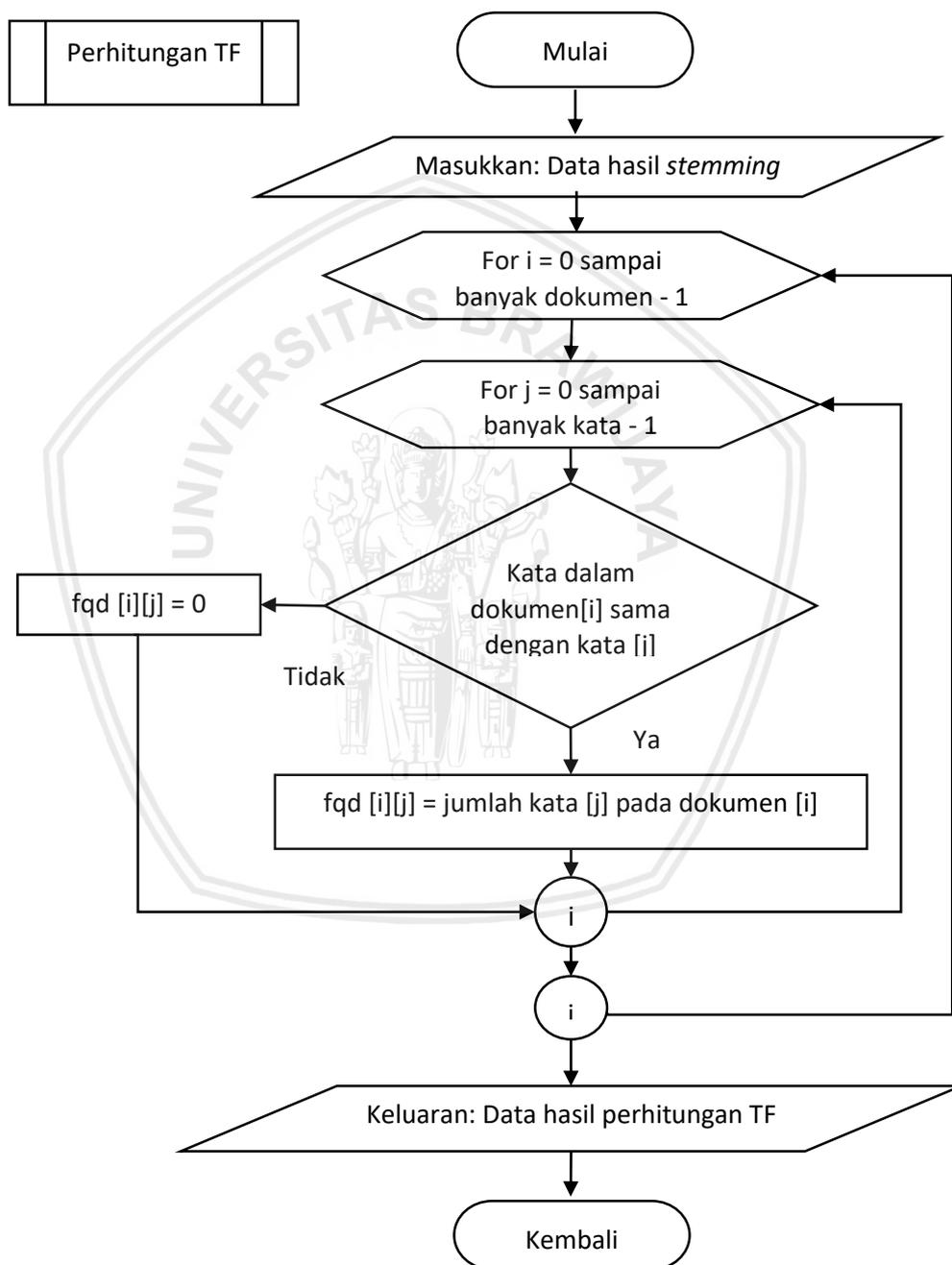




Gambar 4.5 Stemming

### 4.2.5 Perhitungan TF

Perhitungan TF dilakukan untuk mengetahui jumlah setiap kata dalam *query* yang berada pada setiap dokumen data latih. Data yang diproses pada tahapan perhitungan ini adalah data hasil *stemming*. Diagram alir pada proses perhitungan TF dapat dilihat pada Gambar 4.6.

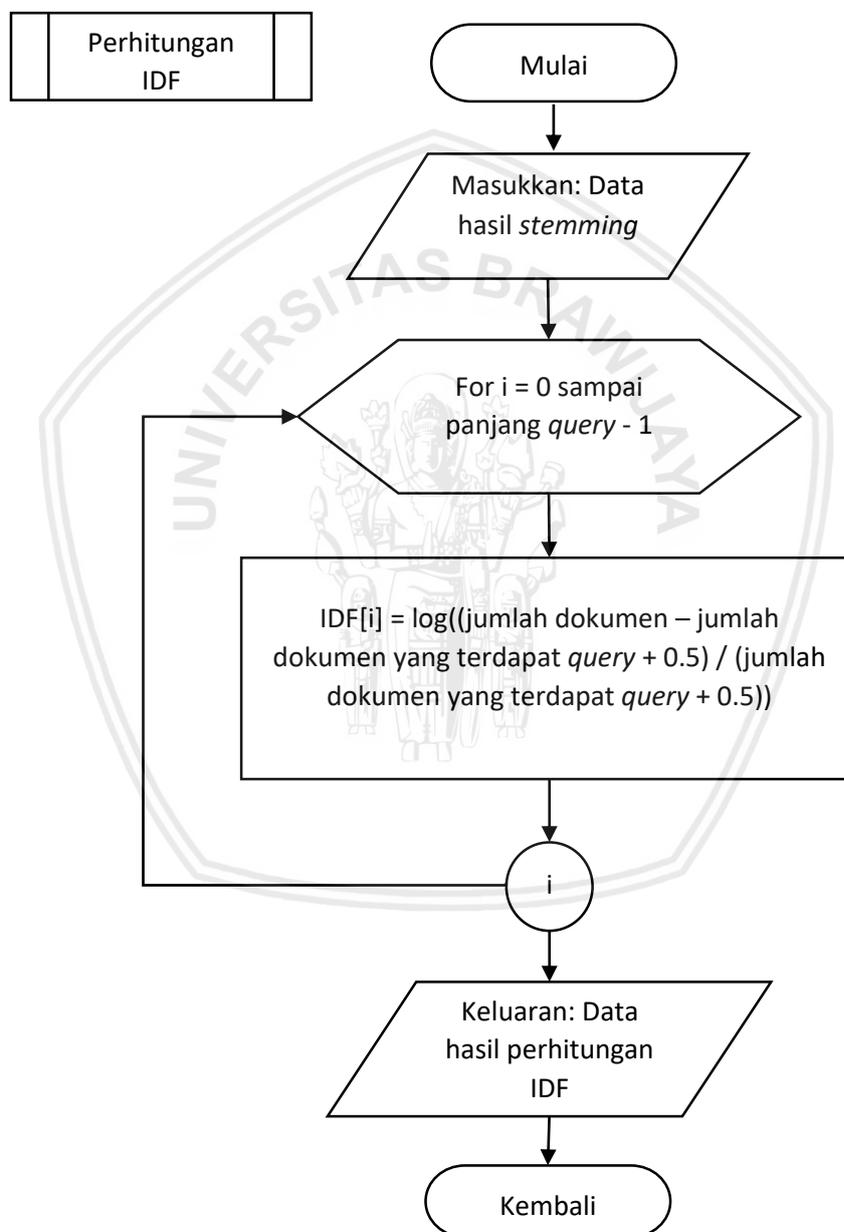


Gambar 4.6 Perhitungan TF



### 4.2.6 Perhitungan IDF

Perhitungan nilai IDF dilakukan pada setiap kata dalam *query* berdasarkan jumlah dokumen yang terdapat kata tersebut. Data yang diproses pada tahapan perhitungan ini adalah data hasil *stemming*. Rumus pada perhitungan IDF dapat dilihat pada persamaan 2.3. Diagram alir pada proses perhitungan IDF dapat dilihat pada Gambar 4.7.

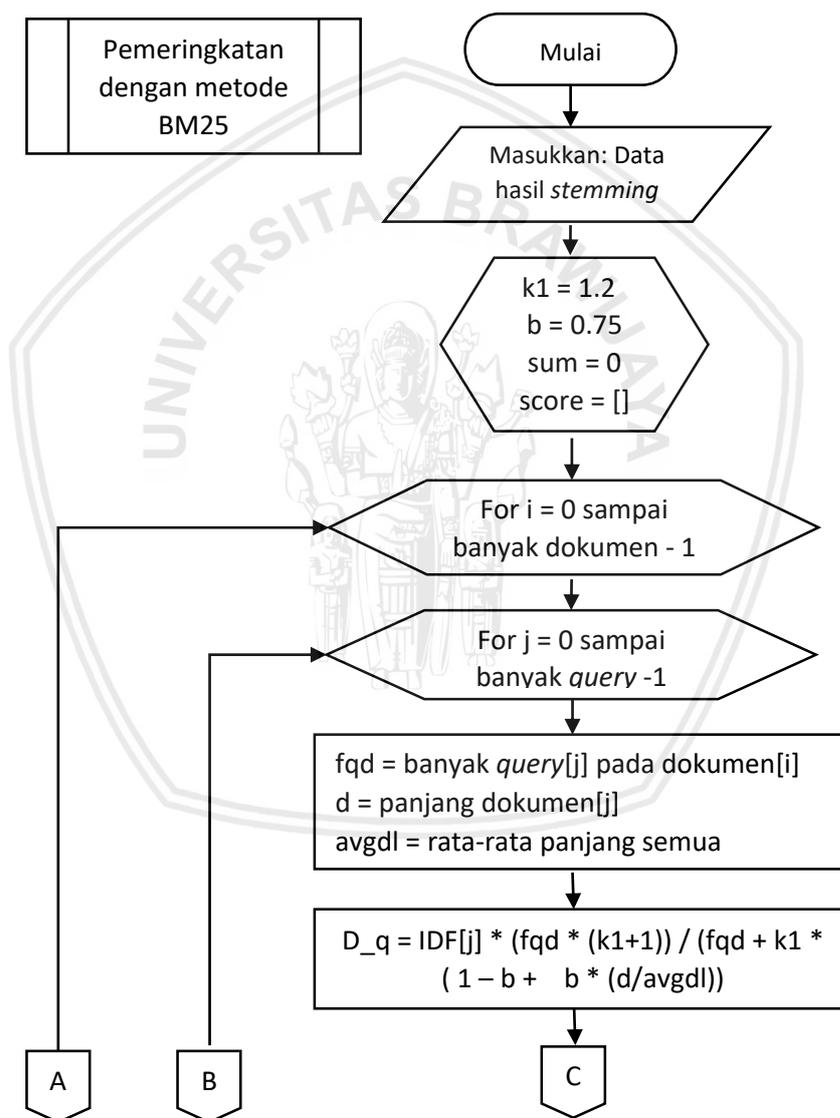


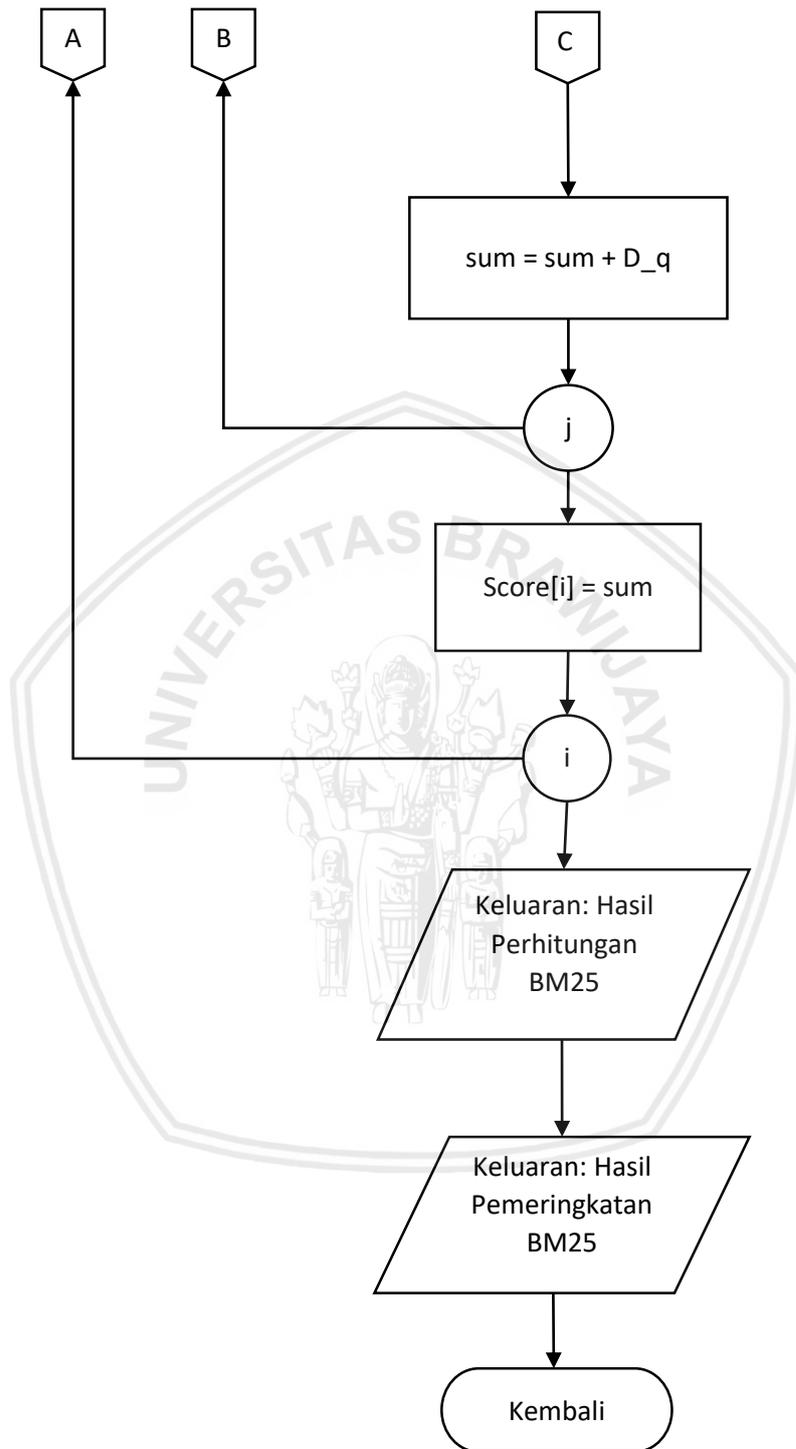
Gambar 4.7 Perhitungan IDF



### 4.2.7 Pemeringkatan dengan Metode BM25

Pada penelitian ini proses pemeringkatan dokumen data latih dilakukan dengan menggunakan metode BM25. Metode BM25 (*Best Matching*) adalah metode dalam sistem temu kembali informasi yang digunakan untuk pemeringkatan hasil relevansi dokumen terhadap *query* yang ingin dicari. Rumus perhitungan untuk metode BM25 dapat dilihat pada persamaan 2.2. Gambar 4.9 menunjukkan diagram alir pada proses pemeringkatan dengan metode BM25.

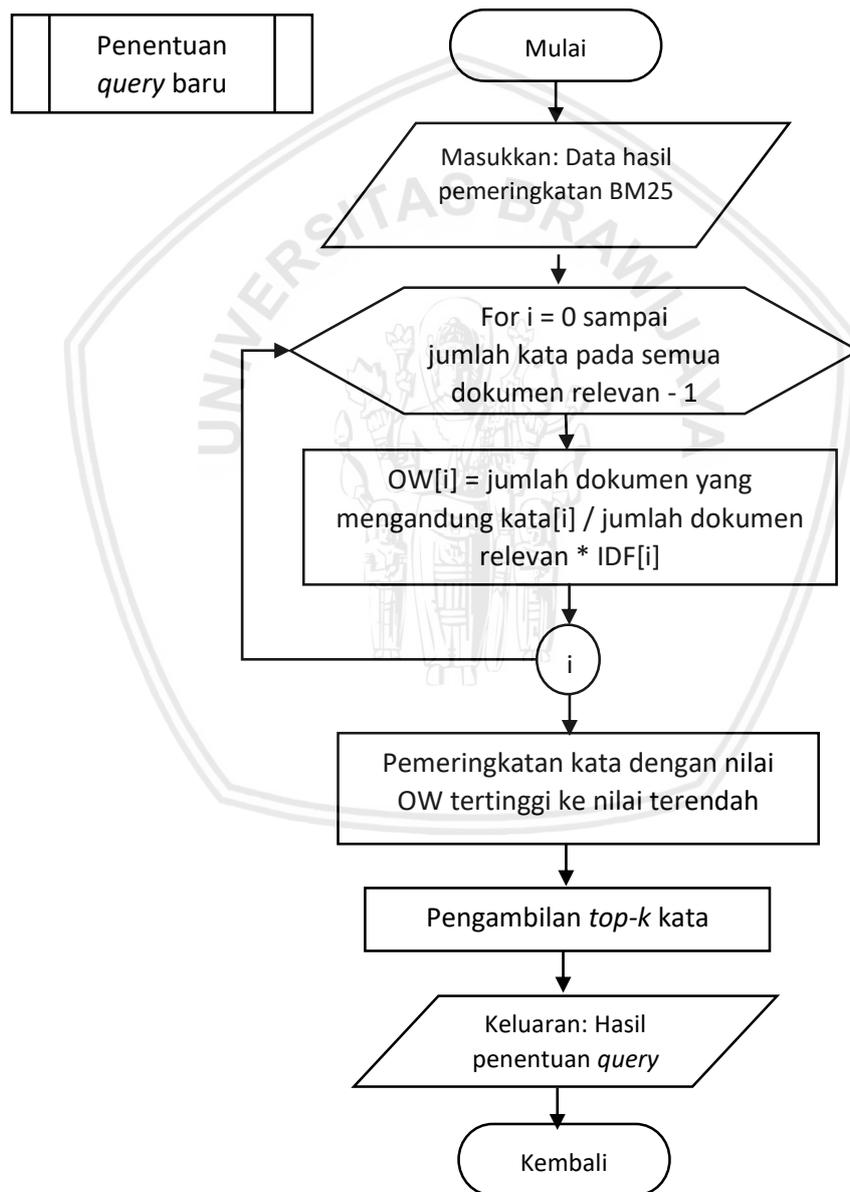




Gambar 4.8 Pemeringkatan dengan Metode BM25

#### 4.2.8 Penentuan Query Baru

Penentuan *query* baru dilakukan berdasarkan dokumen-dokumen yang relevan. Setiap kata pada dokumen yang relevan akan dilakukan perhitungan, dan dilakukan pemeringkatan kata, lalu akan diambil *top-k* kata untuk pemilihan kata yang akan digunakan dalam proses perbaikan *query*. Rumus perhitungan yang digunakan untuk penentuan *query* baru dapat dilihat pada persamaan 2.1. Gambar 4.10 menunjukkan diagram alir pada proses penentuan *query* baru.



Gambar 4.9 Penentuan *query* baru

### 4.3 Manualisasi

Pada tahap manualisasi ini menjelaskan proses perhitungan yang dilakukan terhadap data uji maupun data latih yang terdapat pada penelitian, sebagai gambaran umum dalam proses pengerjaan perhitungan pada penelitian ini.

#### 4.3.1 Dataset

Pada penelitian ini menggunakan dataset yang berasal dari jurnal pada *website* JPTIHK Universitas Brawijaya, yang beralamatkan *j-ptiik.ub.ac.id*. Data latih yang digunakan pada manualisasi ini adalah 3 kalimat dalam abstrak jurnal dan sebanyak 5 dokumen jurnal, sedangkan data uji yang digunakan sejumlah 1 query. Dataset yang digunakan dapat dilihat pada Tabel 4.1.

**Tabel 4.1 Dataset**

No.	Dokumen	Keterangan
1	Dalam penelitian ini menggunakan 400 data komentar yang terbagi menjadi dua yaitu positif dan negatif. Adapun metode yang digunakan adalah metode Support Vector Machine kernel Polynomial berderajat dua dengan Query Expansion. Query Expansion digunakan untuk memperluas kata pada data uji yang memiliki sinonim yang tidak terdapat pada data latih.	Dokumen 1
2	Sehingga dalam penelitian ini dibuat suatu sistem yang mampu melakukan klasifikasi kualitas ikan secara otomatis yang diharapkan dapat mengurangi kesalahan dalam proses penyortiran ikan. Ikan tersebut nantinya akan diambil datanya berupa citra digital yang akan diolah untuk didapatkan beberapa fitur khusus yang akan dijadikan sebagai parameter dalam melakukan klasifikasi menggunakan metode Naive Bayes. Ikan akan diletakkan pada kotak styrofoam yang telah dilengkapi dengan kamera webcam dan lampu sebagai pencahayaan.	Dokumen 2
3	Metode yang diterapkan untuk menyelesaikan persoalan ini adalah Analytical Hierarchy Process (AHP) dan Algoritme Genetika. Individu pada metode Algoritme Genetika berjumlah 6 gen. Pada individu awal yang terbentuk dilakukan proses reproduksi yaitu crossover dan mutasi. Proses terakhir yang dilakukan dalam penelitian ini adalah penghitungan	Dokumen 3

No.	Dokumen	Keterangan
	nilai fitness.	
4	Klasifikasi jenis kanker dilakukan dengan menggunakan metode Fuzzy k-Nearest Neighbor (FkNN). Data yang digunakan merupakan 752 data sekuens protein dengan panjang sekuens yaitu 393. Kelas klasifikasi meliputi kelas bukan kanker, kelas kanker payudara, kelas kanker usus dan kelas kanker paru.	Dokumen 4
5	Hasil dari analisis kebutuhan yang dilakukan, didapatkan 19 kebutuhan fungsional dan 1 kebutuhan nonfungsional. Pengujian yang dilakukan dalam penelitian ini adalah pengujian whitebox dan blackbox untuk kebutuhan fungsional. Sedangkan kebutuhan nonfungsional dilakukan pengujian compatibility.	Dokumen 5
6	Klasifikasi Fitur	Query

#### 4.3.2 Preprocessing

Pada tahapan *preprocessing* ini terdapat proses *case folding*, *filtering*, *tokenization*, dan *stemming*.

Proses *case folding* adalah proses yang mengganti semua huruf besar menjadi huruf kecil (*lower case*) pada setiap kata. Hasil dari proses *case folding* dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Manualisasi Case Folding**

No.	Dokumen	Keterangan
1	dalam penelitian ini menggunakan data komentar yang terbagi menjadi dua yaitu positif dan negatif adapun metode yang digunakan adalah metode support vector machine kernel polynomial berderajat dua dengan query expansion query expansion digunakan untuk memperluas kata pada data uji yang memiliki sinonim yang tidak terdapat pada data latih	Dokumen 1
2	sehingga dalam penelitian ini dibuat suatu sistem yang mampu melakukan klasifikasi kualitas ikan secara otomatis yang diharapkan dapat mengurangi kesalahan dalam proses penyortiran ikan ikan tersebut nantinya akan diambil datanya berupa citra digital yang akan diolah untuk didapatkan beberapa	Dokumen 2

No.	Dokumen	Keterangan
	fitur khusus yang akan dijadikan sebagai parameter dalam melakukan klasifikasi menggunakan metode naive bayes ikan akan diletakkan pada kotak styrofoam yang telah dilengkapi dengan kamera webcam dan lampu sebagai pencahayaan	
3	metode yang diterapkan untuk menyelesaikan persoalan ini adalah analytical hierarchy process ahp dan algoritme genetika individu pada metode algoritme genetika berjumlah gen pada individu awal yang terbentuk dilakukan proses reproduksi yaitu crossover dan mutasi proses terakhir yang dilakukan dalam penelitian ini adalah penghitungan nilai fitness	Dokumen 3
4	klasifikasi jenis kanker dilakukan dengan menggunakan metode fuzzy knearest neighbor fkn data yang digunakan merupakan data sekuens protein dengan panjang sekuens yaitu kelas klasifikasi meliputi kelas bukan kanker kelas kanker payudara kelas kanker usus dan kelas kanker paru	Dokumen 4
5	hasil dari analisis kebutuhan yang dilakukan didapatkan kebutuhan fungsional dan kebutuhan nonfungsional pengujian yang dilakukan dalam penelitian ini adalah pengujian whitebox dan blackbox untuk kebutuhan fungsional sedangkan kebutuhan nonfungsional dilakukan pengujian compatibility	Dokumen 5
6	klasifikasi fitur	Query

Setelah *case folding* proses selanjutnya adalah *tokenization* yang akan memecah setiap kalimat menjadi kata berdasarkan spasi sebagai pemisah kata, dan menghilangkan semua tanda baca, angka, dan karakter lainnya selain huruf. Hasil dari proses *tokenization* dapat dilihat pada Tabel 4.3.

**Tabel 4.3 Manualisasi *Tokenization***

No.	Dokumen	Keterangan
1	dalam // penelitian // ini // menggunakan // data // komentar // yang // terbagi // menjadi // dua // yaitu // positif // dan // negatif // adapun // metode // yang // digunakan // adalah // metode // support // vector // machine // kernel // polynomial // berderajat // dua // dengan // query // expansion // query // expansion // digunakan // untuk // memperluas // kata // pada // data // uji // yang //	Dokumen 1



No.	Dokumen	Keterangan
	memiliki // sinonim // yang // tidak // terdapat // pada // data // latih	
2	sehingga // dalam // penelitian // ini // dibuat // suatu // sistem // yang // mampu // melakukan // klasifikasi // kualitas // ikan // secara // otomatis // yang // diharapkan // dapat // mengurangi // kesalahan // dalam // proses // penyortiran // ikan // ikan // tersebut // nantinya // akan // diambil // datanya // berupa // citra // digital // yang // akan // diolah // untuk // didapatkan // beberapa // fitur // khusus // yang // akan // dijadikan // sebagai // parameter // dalam // melakukan // klasifikasi // menggunakan // metode // naive // bayes // ikan // akan // diletakkan // pada // kotak // styrofoam // yang // telah // dilengkapi // dengan // kamera // webcam // dan // lampu // sebagai // pencahayaan	Dokumen 2
3	metode // yang // diterapkan // untuk // menyelesaikan // persoalan // ini // adalah // analytical // hierarchy // process // ahp // dan // algoritme // genetika // individu // pada // metode // algoritme // genetika // berjumlah // gen // pada // individu // awal // yang // terbentuk // dilakukan // proses // reproduksi // yaitu // crossover // dan // mutasi // proses // terakhir // yang // dilakukan // dalam // penelitian // ini // adalah // penghitungan // nilai // fitness	Dokumen 3
4	klasifikasi // jenis // kanker // dilakukan // dengan // menggunakan // metode // fuzzy // knearest // neighbor // fknn // data // yang // digunakan // merupakan // data // sekuens // protein // dengan // panjang // sekuens // yaitu // kelas // klasifikasi // meliputi // kelas // bukan // kanker // kelas // kanker // payudara // kelas // kanker // usus // dan // kelas // kanker // paru	Dokumen 4
5	hasil // dari // analisis // kebutuhan // yang // dilakukan // didapatkan // kebutuhan // fungsional // dan // kebutuhan // nonfungsional // pengujian // yang // dilakukan // dalam // penelitian // ini // adalah // pengujian // whitebox // dan // blackbox // untuk // kebutuhan // fungsional // sedangkan // kebutuhan // nonfungsional // dilakukan // pengujian // compatibility	Dokumen 5
6	klasifikasi // fitur	Query

Selanjutnya adalah proses *filtering* yang akan menghapus *stopword* sesuai dengan *stopword list*. *Stopword list* yang digunakan adalah *stopword list* Sastrawi. Hasil dari proses *filtering* dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Manualisasi *Filtering***

No.	Dokumen	Keterangan
1	penelitian data komentar terbagi positif negatif metode metode support vector machine kernel polynomial berderajat query expansion query expansion memperluas data uji memiliki sinonim data latih	Dokumen 1
2	penelitian sistem klasifikasi kualitas ikan otomatis diharapkan mengurangi kesalahan proses penyortiran ikan diambil datanya citra digital diolah didapatkan fitur dijadikan parameter klasifikasi metode naive bayes ikan diletakkan kotak styrofoam dilengkapi kamera webcam lampu pencahayaan	Dokumen 2
3	metode diterapkan menyelesaikan analytical hierarchy process ahp algoritme genetika individu metode algoritme genetika gen individu terbentuk proses reproduksi crossover mutasi proses penelitian penghitungan nilai fitness	Dokumen 3
4	klasifikasi jenis kanker metode fuzzy knearest neighbor fknn data data sekuens protein sekuens kelas klasifikasi meliputi kelas kanker kelas kanker payudara kelas kanker usus kelas kanker paru	Dokumen 4
5	hasil analisis kebutuhan didapatkan kebutuhan fungsional kebutuhan nonfungsional pengujian penelitian pengujian whitebox blackbox kebutuhan fungsional kebutuhan nonfungsional pengujian compatibility	Dokumen 5
6	klasifikasi fitur	<i>Query</i>

Langkah terakhir dari *preprocessing* adalah proses *stemming*, pada penelitian ini menggunakan *library stemming* JSastrawi. Hasil dari proses *stemming* dapat dilihat pada Tabel 4.5.

**Tabel 4.5 Manualisasi Stemming**

No.	Dokumen	Keterangan
1	teliti data komentar bagi positif negatif metode metode support vector machine kernel polynomial derajat query expansion query expansion luas data uji milik sinonim data latih	Dokumen 1
2	teliti sistem klasifikasi kualitas ikan otomatis harap kurang salah proses sortir ikan ikan ambil data citra digital olah dapat fitur jadi parameter klasifikasi metode naive bayes ikan letak kotak styrofoam lengkap kamera webcam lampu cahaya	Dokumen 2
3	metode terap selesai analytical hierarchy process ahp algoritme genetika individu metode algoritme genetika gen individu bentuk proses reproduksi crossover mutasi proses teliti hitung nilai fitness	Dokumen 3
4	klasifikasi jenis kanker metode fuzzy knearest neighbor fknn data data sekuens protein sekuens kelas klasifikasi liput kelas kanker kelas kanker payudara kelas kanker usus kelas kanker paru	Dokumen 4
5	hasil analisis butuh dapat butuh fungsional butuh nonfungsional uji teliti uji whitebox blackbox butuh fungsional butuh nonfungsional uji compatibility	Dokumen 5
6	klasifikasi fitur	Query

### 4.3.3 Pembobotan TF IDF

Tahap selanjutnya adalah menghitung pembobotan nilai TF IDF. Dari hasil *preprocessing* dapat dihitung frekuensi kemunculan kata di dalam *query*. Tahap selanjutnya adalah menghitung nilai IDF menggunakan rumus pada persamaan 2.3. Hasil dari perhitungan TF IDF dapat dilihat pada Tabel 4.6.

Contoh perhitungan untuk kata klasifikasi, dengan nilai  $n(q_i)$  adalah 2:

$$\begin{aligned}
 IDF(q_i) &= \log\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}\right) \\
 &= \log\left(\frac{5 - 2 + 0.5}{2 + 0.5}\right) = 0.14613
 \end{aligned}$$

**Tabel 4.6 Hasil Perhitungan TF IDF**

Term	f(q,D1)	f(q,D2)	f(q,D3)	f(q,D4)	f(q,D5)	$n(q_i)$	IDF
klasifikasi	0	2	0	2	0	2	0.14613
fitur	0	1	0	0	0	1	0.47712



### 4.3.4 Perhitungan Metode BM25

Setelah diketahui nilai TF IDF, maka tahap selanjutnya adalah melakukan perhitungan dengan metode BM25 untuk melakukan pemeringkatan dokumen. Perhitungan dengan metode BM25 menggunakan rumus pada persamaan 2.2. Hasil dari perhitungan menggunakan metode BM25 dapat dilihat pada Tabel 4.7.

Contoh perhitungan dengan metode BM25 untuk dokumen 1, dengan nilai panjang dokumen ( $|D|$ ) adalah 25, nilai  $k_1$  adalah 1.2, nilai  $b$  adalah 0.75, dan nilai  $avgdl$  adalah 26.2:

$$\begin{aligned}
 score(D, Q) &= \sum_{i=1}^N IDF(q_i) \times \frac{f(q_i, D)(k_1+1)}{f(q_i, D) + k_1(1-b + b \frac{|D|}{avgdl})} \\
 &= 0.14613 \times \frac{0 * (1.2 + 1)}{0 + 1.2 (1 - 0.75 + 0.75 \frac{25}{26.2})} + 0.47712 \times \\
 &\quad \frac{0 * (1.2 + 1)}{3 + 1.2 (1 - 0.75 + 0.75 \frac{25}{26.2})} \\
 &= 0
 \end{aligned}$$

Tabel 4.7 Hasil Perhitungan BM25

Dokumen 1	0
Dokumen 2	0.60307
Dokumen 3	0
Dokumen 4	0.19921
Dokumen 5	0

Setelah didapatkan hasil perhitungan dengan menggunakan metode BM25 maka selanjutnya dokumen akan diurutkan berdasarkan hasil perhitungannya, semakin besar nilai perhitungan BM25 maka dokumen semakin relevan. Hasil pengurutan dokumen dapat dilihat pada tabel 4.8.

Tabel 4.8 Hasil Pengurutan Dokumen

Peringkat	Dokumen	Nilai hasil BM25
1	2	0.60307
2	4	0.19921
3	1	0
4	3	0
5	5	0

### 4.3.5 Penentuan *Query* baru

Setelah diketahui nilai hasil perhitungan BM25, dan dokumen telah diurutkan, maka tahap selanjutnya adalah penentuan *query* baru. Perhitungan untuk menentukan *query* baru menggunakan rumus pada persamaan 2.1. Hasil dari perhitungan penentuan *query* baru dapat dilihat pada Tabel 4.9.

Dari hasil pemeringkatan dokumen pada Tabel 4.7, dilakukan permisalan jika dokumen 2 dan dokumen 4 adalah dokumen yang relevan, maka semua kata pada dokumen 2 dan dokumen 4 akan dihitung menggunakan rumus persamaan 2.1 untuk menentukan kata pada *query* baru. Contoh perhitungan penentuan *query* baru untuk kata teliti, dengan nilai  $r_i$  adalah 1, nilai R 2 dan nilai IDF 0.14613:

$$\begin{aligned} OW_i &= \frac{r_i}{R} \times w_i \\ &= \frac{1}{2} \times -0.47712 = -0.23856 \end{aligned}$$

**Tabel 4.9 Hasil Penentuan *Query* Baru**

Kata	$r_i$	$w_i$	$OW_i$
teliti	1	-0.47712	-0.23856
metode	2	-0.47712	-0.47712
data	2	-0.14613	-0.14612
sistem	1	0.47712	0.23856
klasifikasi	2	0.14613	0.14613
kualitas	1	0.47712	0.23856
ikan	1	0.47712	0.23856
otomatis	1	0.47712	0.23856
harap	1	0.47712	0.23856
kurang	1	0.47712	0.23856
salah	1	0.47712	0.23856
proses	1	0.14613	0.07306
sortir	1	0.47712	0.23856
ambil	1	0.47712	0.23856
citra	1	0.47712	0.23856
digital	1	0.47712	0.23856
olah	1	0.47712	0.23856
dapat	1	0.47712	0.23856
fitur	1	0.47712	0.23856
jadi	1	0.47712	0.23856
parameter	1	0.47712	0.23856
naive	1	0.47712	0.23856

Kata	$r_i$	$w_i$	$OW_i$
bayes	1	0.47712	0.23856
letak	1	0.47712	0.23856
kotak	1	0.47712	0.23856
styfoam	1	0.47712	0.23856
lengkap	1	0.47712	0.23856
kamera	1	0.47712	0.23856
webcam	1	0.47712	0.23856
lampu	1	0.47712	0.23856
cahaya	1	0.47712	0.23856
jenis	1	0.47712	0.23856
kanker	1	0.47712	0.23856
fuzzy	1	0.47712	0.23856
knearest	1	0.47712	0.23856
neighbor	1	0.47712	0.23856
fknn	1	0.47712	0.23856
sekuens	1	0.47712	0.23856
protein	1	0.47712	0.23856
kelas	1	0.47712	0.23856
liput	1	0.47712	0.23856
payudara	1	0.47712	0.23856
usus	1	0.47712	0.23856
paru	1	0.47712	0.23856

Setelah didapatkan hasil perhitungan penentuan kata untuk *query* baru maka selanjutnya kata tersebut akan diurutkan berdasarkan hasil perhitungannya, setelah itu akan diambil *top-k* kata sebagai rekomendasi untuk pengguna dalam memperbaiki *query* untuk proses pencarian selanjutnya. Dalam manualisasi ini diambil semua kata dengan hasil perhitungan 0.23856 sebagai rekomendasi kata untuk pengguna. Hasil rekomendasi kata untuk *query* baru dapat dilihat pada tabel 4.10.

**Tabel 4.10 Hasil Rekomendasi Kata *Query* baru**

Kata	$OW_i$
webcam	0.23856
usus	0.23856
styrofoam	0.23856
sortir	0.23856
sistem	0.23856
sekuens	0.23856
salah	0.23856
protein	0.23856
payudara	0.23856
paru	0.23856

Kata	$OW_i$
parameter	0.23856
otomatis	0.23856
olah	0.23856
neighbor	0.23856
naive	0.23856
liput	0.23856
letak	0.23856
lengkap	0.23856
lampu	0.23856
kurang	0.23856
kualitas	0.23856
kotak	0.23856
knearest	0.23856
kelas	0.23856
kanker	0.23856
kamera	0.23856
jenis	0.23856
jadi	0.23856
ikan	0.23856
harap	0.23856
fuzzy	0.23856
fknn	0.23856
fitur	0.23856
digital	0.23856
citra	0.23856
cahaya	0.23856
bayes	0.23856
ambil	0.23856
klasifikasi	0.14612
proses	0.07306
dapat	0.07306
data	-0.14163
teliti	-0.23856
metode	-0.47712

#### 4.4 Perancangan Pengujian

Pengujian pada penelitian ini dilakukan dengan menggunakan 350 data latih, dan 10 *query*. Metode perhitungan yang digunakan untuk melakukan pengujian adalah metode *precision@k*. Rumus perhitungan yang digunakan untuk nilai *precision@k* menggunakan persamaan 2.4. Pengujian dilakukan untuk menguji hasil sebelum penerapan *query expansion* dan hasil setelah penerapan *query expansion*, untuk pencarian dokumen jurnal berbahasa Indonesia, pengujian dilakukan setiap penambahan jumlah kata yang

digunakan untuk memperbaiki *query*. Perancangan pengujian yang dilakukan dapat dilihat pada Tabel 4.11.

**Tabel 4.11 Perancangan Pengujian Penerapan *Query Expansion***

Nilai K	<i>Precision@K</i> Sebelum QE	<i>Precision@K</i> Setelah QE	Kenaikan

#### 4.5 Perancangan Tampilan Antarmuka

Antarmuka berfungsi sebagai perantara antara pengguna dengan sistem yang telah dibangun, agar dapat saling berinteraksi. Pada penelitian *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia menggunakan metode BM25 ini, menerapkan tampilan antarmuka berupa *Command Line Interface* (CLI). Pada tampilan antarmuka CLI ini pengguna akan berinteraksi dengan sistem melalui *text terminal*, dimana pengguna dapat berinteraksi dengan cara mengetik baris perintah tertentu.

## BAB 5 IMPLEMENTASI

Bab 5 menjelaskan tentang bagaimana penerapan metode *query expansion* pada sistem temu kembali informasi jurnal berbahasa Indonesia menggunakan metode BM25, berdasar pada perancangan yang telah dibuat. Penerapan yang telah dilakukan yaitu penerapan proses *case folding*, *filtering*, *tokenization*, *stemming*, perhitungan nilai TF perhitungan nilai IDF, pemeringkatan dengan metode BM25, dan penentuan *query* baru.

### 5.1 Lingkungan Implementasi

Lingkungan implementasi pada penelitian ini dibagi menjadi dua lingkungan yaitu perangkat keras dan perangkat lunak, yang berdasar pada perancangan yang sudah dijelaskan sebelumnya.

#### 5.1.1 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Sistem Operasi Windows 10 (64 bit)
2. Microsoft Office 2013
3. Spyder IDE
4. Bahasa pemrograman Python
5. XAMPP Control Panel v3.2.2

#### 5.1.2 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Processor Intel Core i5
2. Harddisk 500 GB HDD
3. RAM 4.00 GB

### 5.2 Implementasi Sistem

Pada sub bab ini menjelaskan tentang bagaimana tahapan penerapan metode *query expansion* pada sistem temu kembali informasi jurnal berbahasa Indonesia menggunakan metode BM25. Tahapan yang dilakukan antara lain *case folding*, *filtering*, *tokenization*, *stemming*, perhitungan nilai TF, perhitungan nilai IDF, pemeringkatan dengan metode BM25, dan penentuan *query* baru.

### 5.2.1 Case Folding

*Case Folding* adalah tahapan untuk mengganti semua huruf besar menjadi huruf kecil (*lower case*) pada setiap kata. Kode program *case folding* dapat dilihat pada *Source Code 5.1*.

Algoritma 1: Case Folding	
1	class case_folding:
2	def __init__(self,data):
3	self.data = data
4	for i in range (0,len(data)):
5	dat_clear = re.sub('[^a-zA-Z]', ' ',str(data[i]))
6	data_casefold.append(dat_clear.lower())
7	self.data casefold = data casefold

#### Source Code 5.1 Case Folding

Pada baris 1-7 dilakukan pengimplementasian dari proses *case folding*, dimana setiap kata pada seluruh dokumen diganti menjadi huruf kecil (*lower case*), pada baris ke 5 merupakan kode program untuk proses menghilangkan semua karakter selain huruf.

### 5.2.2 Filtering

*Filtering* adalah tahapan untuk menghilangkan semua *stop words* yaitu kata yang terlalu sering muncul dan tidak memiliki makna. *Stopword list* yang digunakan dalam penelitian ini adalah *stopword list* Sastrawi. Kode program *filtering* dapat dilihat pada *Source Code 5.2*.

Algoritma 2: Filtering	
1	class filtering:
2	def __init__(self,data):
3	self.data = data
4	for i in range(0,len(data)):
5	data_filter.append(stopwords.remove (data[i]))
6	self.data filter = data filter

#### Source Code 5.2 Filtering

Pada baris 1-6 dilakukan pengimplementasian dari proses *filtering*, dimana setiap kata didalam seluruh dokumen akan di hapus apabila kata tersebut sesuai dengan kata yang terdapat pada *stopwords list* Sastrawi. Data yang diproses pada tahapan perhitungan ini adalah data hasil *case folding*.

### 5.2.3 Tokenization

*Tokenization* adalah tahapan untuk memecah setiap kalimat menjadi satuan kata berdasarkan spasi sebagai pemisah. Kode program *tokenization* dapat dilihat pada *Source Code 5.3*.

Algoritma 3: Tokenization	
1	class tokenization:
2	def __init__(self,data):
3	self.data = data

Algoritma 3: Tokenization	
4	for i in range (0,len(data)):
5	data_tokens.append(word_tokenize(data[i]))
6	self.data_tokens = data_tokens

#### Source Code 5.3 Kode Program *Tokenization*

Pada baris 1-6 dilakukan pengimplementasian dari proses *tokenization*, dimana setiap kalimat didalam seluruh dokumen akan dipecah menjadi satuan kata berdasarkan spasi sebagai pemisahannya. Data yang diproses pada tahapan perhitungan ini adalah data hasil *filtering*.

#### 5.2.4 Stemming

*Stemming* adalah tahapan untuk menghasilkan kata dasar dengan cara menghilangkan semua imbuhan, sesuai dengan struktur morfologi bahasa Indonesia. *Stemming* yang digunakan pada penelitian ini adalah *stemming* Sastrawi yang menggunakan algoritma berbasis Nazief dan Adriani. Kode program *stemming* dapat dilihat pada *Source Code* 5.4.

Algoritma 4: Stemming	
1	class stemming:
2	def __init__(self,data):
3	self.data = data
4	for i in range(0,len(data)):
5	dat_str = str(data[i])
6	data_stem.append(stemmer.stem(dat_str))
7	self.data_stem = data_stem

#### Source Code 5.4 *Stemming*

Pada baris 1-7 dilakukan pengimplementasian dari proses *stemming*, dimana setiap kata didalam seluruh dokumen akan dihapus imbuhanannya sehingga menghasilkan kata dasar, dengan menggunakan *stemming* Sastrawi yang berbasis algoritma Nazief dan Adriani. Data yang diproses pada tahapan perhitungan ini adalah data hasil *tokenization*.

#### 5.2.5 Perhitungan Nilai TF

Perhitungan nilai TF adalah tahapan untuk mengetahui jumlah setiap kata dari hasil *stemming* yang berada pada setiap dokumen data latih. Kode program perhitungan nilai TF dapat dilihat pada *Source Code* 5.5.

Algoritma 5: Perhitungan Nilai TF	
1	class calc_fqd:
2	def __init__(self,data):
3	self.data = data
4	fqd = []
5	y = str(data)
6	kd = y.split()
7	kata = Counter(kd)
8	keys = np.array(list(kata.keys()))
9	values = np.array(list(kata.values()))
10	for i in range(0,len(term_)):
11	fqd.append(0)

Algoritma 5: Perhitungan Nilai TF	
12	for j in range(0,len(keys)):
13	if (term_[i][0]==keys[j]):
14	fqd[i]+=values[j]
15	self.fqd = fqd

### Source Code 5.5 Perhitungan Nilai TF

Pada baris 1-15 dilakukan pengimplementasian dari proses perhitungan nilai TF, dimana setiap kata dalam setiap dokumen akan dihitung jumlahnya. Data yang diproses pada tahapan perhitungan ini adalah data hasil *stemming*.

### 5.2.6 Perhitungan Nilai IDF

Perhitungan nilai IDF adalah tahapan untuk mengetahui nilai IDF pada setiap kata dalam *query* berdasarkan jumlah dokumen yang terdapat kata tersebut, menggunakan rumus persamaan 2.3. Kode program perhitungan nilai IDF dapat dilihat pada *Source Code 5.6*.

Algoritma 6: Perhitungan Nilai IDF	
1	class calc_idf:
2	def __init__(self,query):
3	self.query = query
4	for i in range(0,len(ind)):
5	idf.append(math.log10(((sheet.ncols-1)-
6	df[ind[i]]+0.5)/(df[ind[i]]+0.5)))

### Source Code 5.6 Perhitungan Nilai IDF

Pada baris 1-6 dilakukan pengimplementasian dari proses perhitungan nilai IDF, dimana setiap kata dalam *query* akan dihitung nilai IDF nya berdasarkan jumlah dokumen yang terdapat kata tersebut. Data yang diproses pada tahapan perhitungan ini adalah data hasil *stemming*.

### 5.2.7 Pemeringkatan Metode BM25

Pemeringkatan metode BM25 adalah tahap untuk mengurutkan dokumen berdasarkan hasil perhitungan dengan menggunakan metode BM25 yang telah didapatkan menggunakan rumus pada persamaan 2.2, semakin besar nilai perhitungan BM25 maka dokumen semakin relevan dan peringkat semakin atas. Kode program pemeringkatan metode BM25 dapat dilihat pada *Source Code 5.7*.

Algoritma 7: Pemeringkatan Metode BM25	
1	class calc_score:
2	def __init__(self,query):
3	self.query = query
4	for i in range(0,sheet.ncols-1):
5	temp_score=0
6	for j in range(0,len(ind)):
7	temp_score += (idf[j]) *
8	(((fqd[ind[j]][i])*(k1+1)) / (fqd[ind[j]][i]+(k1*(1-
9	b+(b*(D[i]/avgdl[0]))))))
10	score.append(temp_score)
11	class relevance doc:

Algoritma 7: Pemeringkatan Metode BM25	
12	def __init__(self, query):
13	self.query = query
14	for i in range(0, len(score)):
15	temp_sr.append([])
16	for j in range(0, 1):
17	temp_sr[i].append(score[i])
18	temp_sr[i].append(i)
19	score_relv = sorted(temp_sr, reverse=True)
20	for i in range(0, len(score_relv)):
21	if(score_relv[i][0] != 0):
22	doc_relv.append(score_relv[i])
23	ket.append("r")
24	else:
25	ket.append("t")
	score_relv.clear()

### Source Code 5.7 Pemeringkatan Metode BM25

Pada baris 1-26 dilakukan pengimplementasian dari proses pemeringkatan dengan metode BM25. Pada baris 1-9 dilakukan proses perhitungan menggunakan metode BM25 pada setiap dokumen berdasarkan *query* dengan rumus persamaan 2.2. Selanjutnya pada baris 10-25 dilakukan proses pemeringkatan dokumen sesuai dengan nilai perhitungan metode BM25 yang telah didapatkan, semakin besar nilai perhitungan metode BM25 maka peringkat akan semakin tinggi.

### 5.2.8 Penentuan Query Baru

Penentuan *query* baru adalah tahap untuk menentukan kata baru yang berasal dari dokumen-dokumen relevan, sebagai perbaikan dari *query*. Penentuan *query* baru dilakukan setelah diketahui nilai hasil perhitungan BM25, dan dokumen telah diurutkan. Perhitungan untuk menentukan *query* baru menggunakan rumus pada persamaan 2.1. Kode program penentuan *query* baru dapat dilihat pada *Source Code 5.8*.

Algoritma 8: Penentuan Query Baru	
1	class recom_newquery:
2	def __init__(self, doc_relv):
3	self.doc_relv = doc_relv
4	temp_rec_query = []
5	for i in range(0, len(doc_relv)):
6	for j in range(0, sheet.nrows):
7	if(int(sheet.cell_value(j, doc_relv[i][1]+1)) !=
8	0):
9	temp_rec_query.append(sheet.cell_value(j, 0))
10	kata = Counter(temp_rec_query)
11	rec_query = np.array(list(kata.keys()))
12	for i in range(0, len(rec_query)):
13	temp_ri = 0
14	for j in range(0, len(doc_relv)):
15	for k in range(0, sheet.nrows):
16	if(rec_query[i] == sheet.cell_value(k, 0)):
17	if(int(sheet.cell_value(k, doc_relv[j][1]+1)) != 0):
18	temp_ri += 1
19	ri.append(temp_ri)
20	removing()

#### Algoritma 8: Penentuan *Query* Baru

```

21         self.rec_query = rec_query
22     class det_newquery:
23         def __init__(self, rec_query,R,n):
24             self.rec_query = rec_query
25             self.R = R
26             self.n = n
27             new_query = []
28             for i in range(0,len(rec_query)):
29                 temp_owi.append([])
30                 for j in range(0,1):
31                     temp_owi[i].append((ri[i]/R)*idf[i])
32                     temp_owi[i].append(rec_query[i])
33             owi = sorted(temp_owi, reverse=True)
34             for i in range(0,len(run4.data_stem)):
35                 new_query.append(run4.data_stem[i])
36
37             for i in range(0,n+1):
38                 new_query.append(owi[i][1])
39
40             new_query = Counter(new_query)
41             new_query = np.array(list(new_query.keys()))
42             removing()
43             self.new_query = new_query

```

#### Source Code 5.8 Penentuan *Query* Baru

Pada baris 1-43 dilakukan pengimplementasian dari proses penentuan *query* baru. Pada baris 1-21 dilakukan proses untuk menentukan kata yang akan digunakan sebagai rekomendasi untuk perbaikan *query*, yang diambil dari seluruh dokumen relevan. Selanjutnya pada baris 22-43 dilakukan proses perhitungan setiap kata rekomendasi dengan menggunakan rumus persamaan 2.1. Setelah perhitungan selesai maka kata akan di peringkat secara menurun, semakin kecil nilai maka peringkat akan semakin bawah.

### 5.3 Implementasi Tampilan Antarmuka

Implementasi tampilan antarmuka seperti yang telah di rancang sebelumnya yaitu menerapkan tampilan antarmuka berupa *Command Line Interface* (CLI). Pada Gambar 5.1 tersebut dilakukan pengujian terhadap sistem yang telah dibangun dengan *query* ikan, dengan penambahan kata sebanyak 3 kata untuk *query expansion*.

```

In [8]: runfile('D:/SKRIPSI/CODE/skripsi.py', wdir='D:/SKRIPSI/CODE')

Query : ikan
Dokumen Relevan : 5 dokumen
[[3.6774654497072667, 279], [3.3776397411174615, 246], [3.0273267671895923, 321], [2.3873868443664112, 233],
[1.8908088541347448, 325]]
Nilai Precision@K ( 10 ) : 0.5

===Setelah Query Expansion===
Query : ['ikan' 'kolam' 'budidaya' 'rentan']
Dokumen Relevan : 12 dokumen
[[9.23855776383099, 321], [7.794069898576387, 279], [4.426094595352096, 233], [3.5303879358803494, 40],
[3.3776397411174615, 246], [1.8908088541347448, 325], [1.855679089208994, 124], [1.773370090231601, 151],
[1.7333058605699578, 101], [1.658596191636474, 82], [1.5473249884227678, 200], [1.506887611680904, 173]]
Nilai Precision@K ( 10 ) : 1.0

```

Gambar 5.1 Tampilan Antarmuka

## BAB 6 HASIL DAN PEMBAHASAN

Bab 6 menjelaskan tentang pembahasan dan hasil dari pengujian sistem berdasarkan implementasi yang telah dilakukan terhadap hasil *query expansion* pada sistem temu kembali dokumen jurnal bahasa Indonesia menggunakan metode BM25.

### 6.1 Pengujian *Precision@K*

Pengujian untuk sistem temu kembali dokumen jurnal bahasa Indonesia ini menggunakan metode *Precision@K*. Perhitungan dari metode *Precision@K* dapat dilihat pada persamaan 2.4. Pada pengujian ini akan membandingkan hasil nilai rata-rata *Precision@K* pada sistem sebelum menggunakan *query expansion* dan setelah menggunakan *query expansion* pada 350 data latih, dan setiap percobaannya akan dilakukan dengan nilai *K* yang berbeda. Pengujian pada penelitian ini menggunakan 4 skenario pengujian. Pada setiap skenario pengujian dilakukan 10 kali dengan nilai *K* 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100, dimana pada setiap nilai *K* dilakukan pengujian dengan 10 *query* yaitu “Memprediksi”, “Penjadwalan”, “Twitter”, “Citra”, “IOT”, “Transportasi”, “Optimasi”, “Android”, “Nirkabel”, dan “LVQ”. Hasil detail dari setiap *query* dapat dilihat pada Lampiran A.2

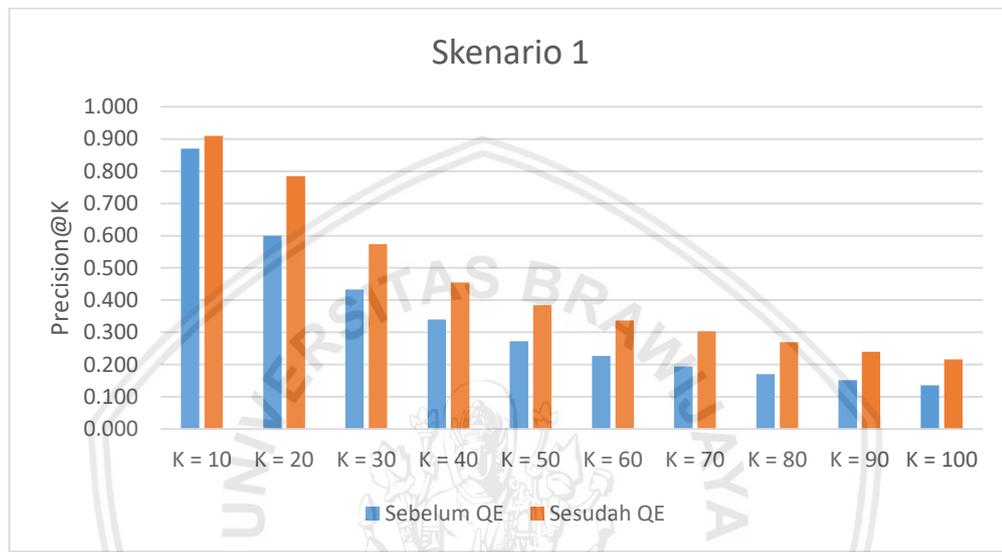
#### 6.1.1 Skenario 1

Pengujian pada skenario 1 ini, dilakukan dengan penambahan 1 kata peringkat teratas sesuai dengan hasil yang telah didapatkan melalui proses pemerinkatan kata untuk *query* baru dengan menggunakan rumus persamaan 2.1. Hasil pengujian pada skenario 1 dapat dilihat pada Tabel 6.1.

**Tabel 6.1 Hasil Pengujian Skenario 1**

Nilai K	<i>Precision@K</i> Sebelum QE	<i>Precision@K</i> Setelah QE	Kenaikan
10	0.870	0.910	0.004
20	0.600	0.785	0.185
30	0.433	0.573	0.140
40	0.340	0.455	0.115
50	0.272	0.384	0.112
60	0.227	0.337	0.110
70	0.194	0.303	0.109
80	0.170	0.270	0.100
90	0.151	0.240	0.089
100	0.136	0.216	0.080

Seperti yang dapat dilihat pada Tabel 6.1, hasil pengujian kenaikan nilai *Precision@K* pada 10 *query* tersebut menghasilkan nilai yang tertinggi pada saat  $K = 20$  dengan nilai kenaikan sebesar 0.185. Hal tersebut menunjukkan bahwa setelah dilakukannya *query expansion* pada saat penambahan 1 kata untuk setiap *query*, persebaran dokumen yang relevan berada pada peringkat 20 dokumen teratas.



**Gambar 6.1 Grafik Skenario 1**

Berdasarkan grafik yang dapat dilihat pada Gambar 6.1, nilai *Precision@K* sebelum *query expansion* tertinggi pada saat  $K = 10$ , dan nilai *Precision@K* tertinggi setelah *query expansion* dengan penambahan 1 kata pada saat  $K = 10$ .

### 6.1.2 Skenario 2

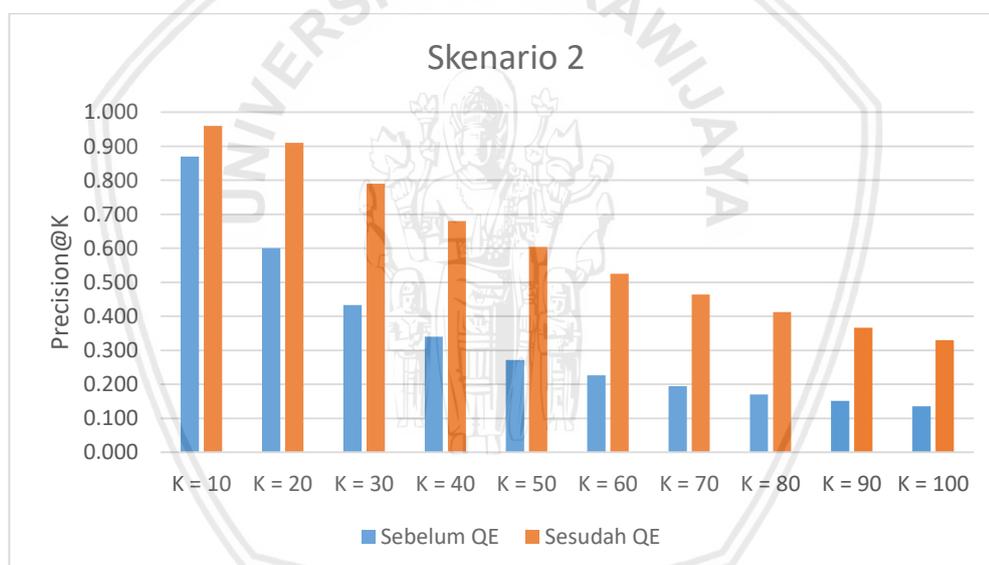
Pengujian pada skenario 2 ini, dilakukan dengan penambahan 2 kata peringkat teratas sesuai dengan hasil yang telah didapatkan melalui proses pemeringkatan kata untuk *query* baru dengan menggunakan rumus persamaan 2.1. Hasil pengujian pada skenario 2 dapat dilihat pada Tabel 6.2.

**Tabel 6.2 Hasil Pengujian Skenario 2**

Nilai K	<i>Precision@K</i> Sebelum QE	<i>Precision@K</i> Setelah QE	Kenaikan
10	0.870	0.960	0.090
20	0.600	0.910	0.310
30	0.433	0.790	0.357
40	0.340	0.680	0.340
50	0.272	0.604	0.332

Nilai K	<i>Precision@K</i> Sebelum QE	<i>Precision@K</i> Setelah QE	Kenaikan
60	0.227	0.525	0.298
70	0.194	0.464	0.270
80	0.170	0.413	0.243
90	0.151	0.367	0.216
100	0.136	0.330	0.194

Seperti yang dapat dilihat pada Tabel 6.2, hasil pengujian kenaikan nilai *Precision@K* pada 10 *query* tersebut menghasilkan nilai yang tertinggi pada saat  $K=30$  dengan nilai kenaikan sebesar 0.357. Hal tersebut menunjukkan bahwa setelah dilakukannya *query expansion* pada saat penambahan 2 kata untuk setiap *query* persebaran dokumen yang relevan berada pada peringkat 30 dokumen teratas.



Gambar 6.2 Grafik Skenario 2

Berdasarkan grafik yang dapat dilihat pada Gambar 6.2, nilai *Precision@K* sebelum *query expansion* pada saat  $K=10$ , dan nilai *Precision@K* tertinggi setelah *query expansion* dengan penambahan 2 kata tertinggi pada saat  $K=10$ .

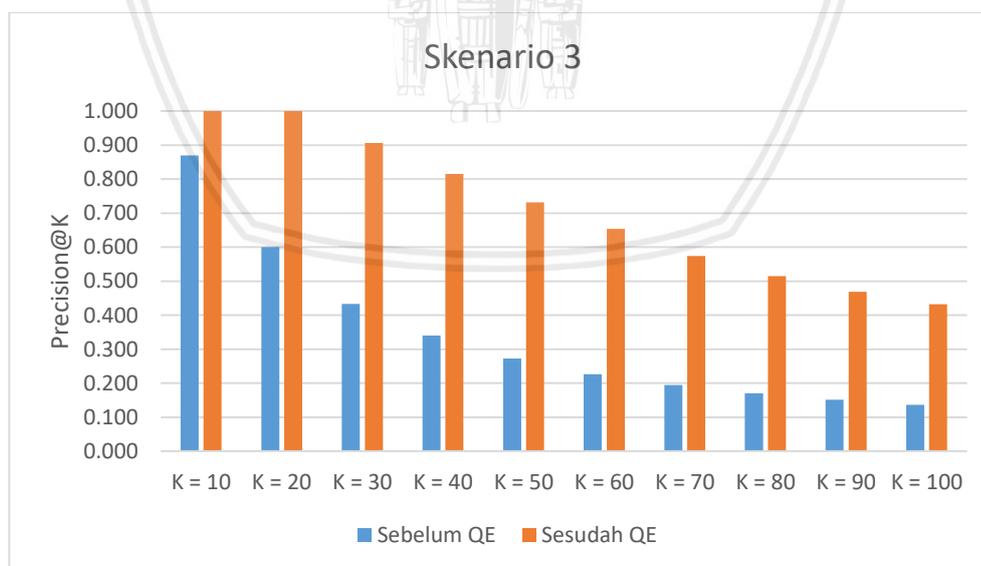
### 6.1.3 Skenario 3

Pengujian pada skenario 3 ini, dilakukan dengan penambahan 3 kata peringkat teratas sesuai dengan hasil yang telah didapatkan melalui proses pemeringkatan kata untuk *query* baru dengan menggunakan rumus persamaan 2.1. Hasil pengujian pada skenario 3 dapat dilihat pada Tabel 6.3.

**Tabel 6.3 Hasil Pengujian Skenario 3**

Nilai K	<i>Precision@K</i> Sebelum QE	<i>Precision@K</i> Setelah QE	Kenaikan
10	0.870	1.000	0.130
20	0.600	1.000	0.400
30	0.433	0.907	0.473
40	0.340	0.815	0.475
50	0.272	0.732	0.460
60	0.227	0.653	0.427
70	0.194	0.574	0.380
80	0.170	0.515	0.345
90	0.151	0.469	0.318
100	0.136	0.432	0.296

Seperti yang dapat dilihat pada Tabel 6.2, hasil pengujian kenaikan nilai *Precision@K* pada 10 *query* tersebut menghasilkan nilai yang tertinggi pada saat  $K = 40$  dengan nilai kenaikan sebesar 0.475. Hal tersebut menunjukkan bahwa setelah dilakukannya *query expansion* pada saat penambahan 3 kata untuk setiap *query* persebaran dokumen yang relevan berada pada peringkat 40 dokumen teratas.



**Gambar 6.3 Grafik Skenario 3**



Berdasarkan grafik yang dapat dilihat pada Gambar 6.3, nilai *Precision@K* sebelum *query expansion* tertinggi pada saat  $K= 10$ , dan nilai *Precision@K* tertinggi setelah *query expansion* dengan penambahan 3 kata pada saat  $K= 10$  dan  $K= 20$ .

#### 6.1.4 Skenario 4

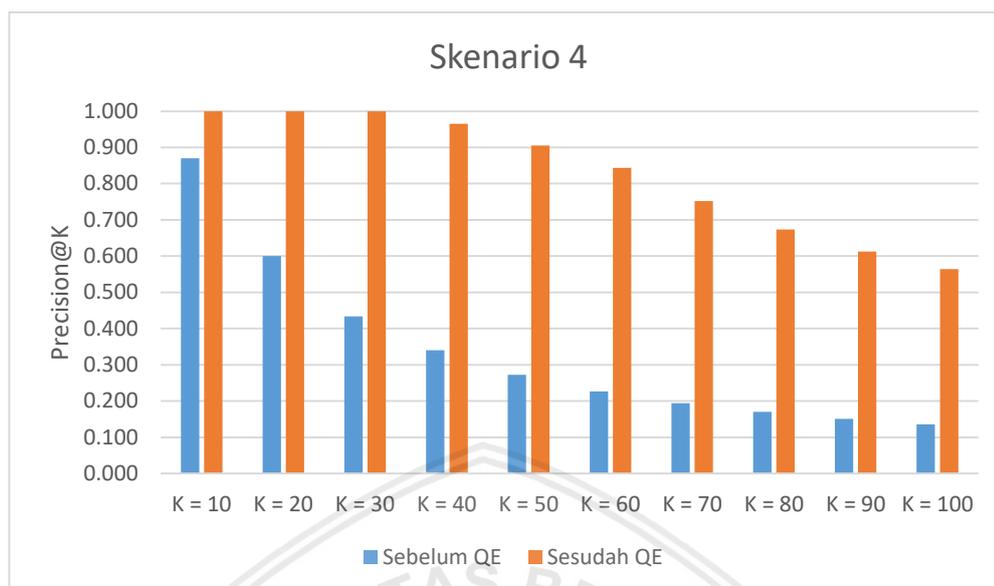
Pengujian pada skenario 4 ini, dilakukan dengan penambahan 4 kata peringkat teratas sesuai dengan hasil yang telah didapatkan melalui proses pemeringkatan kata untuk *query* baru dengan menggunakan rumus persamaan 2.1. Hasil pengujian pada skenario 4 dapat dilihat pada Tabel 6.4.

**Tabel 6.4 Hasil Pengujian Skenario 4**

Nilai K	<i>Precision@K</i> Sebelum QE	<i>Precision@K</i> Setelah QE	Kenaikan
10	0.870	1.000	0.130
20	0.600	1.000	0.400
30	0.433	1.000	0.567
40	0.340	0.966	0.626
50	0.272	0.905	0.633
60	0.227	0.844	0.617
70	0.194	0.752	0.557
80	0.170	0.673	0.503
90	0.151	0.613	0.461
100	0.136	0.564	0.428

Seperti yang dapat dilihat pada Tabel 6.4, hasil pengujian kenaikan nilai *Precision@K* pada 10 *query* tersebut menghasilkan nilai yang tertinggi pada saat  $K= 50$  dengan nilai kenaikan sebesar 0.633. Hal tersebut menunjukkan bahwa setelah dilakukannya *query expansion* pada saat penambahan 4 kata untuk setiap *query* persebaran dokumen yang relevan berada pada peringkat 50 dokumen teratas.

Berdasarkan grafik yang dapat dilihat pada Gambar 6.4, nilai *Precision@K* sebelum *query expansion* tertinggi pada saat  $K= 10$ , dan nilai *Precision@K* tertinggi setelah *query expansion* dengan penambahan 4 kata pada saat  $K= 10$ ,  $K= 20$  dan  $K= 30$ .



Gambar 6.4 Grafik Skenario 4

## 6.2 Pembahasan Hasil Pengujian

Dari hasil pengujian yang telah dilakukan, dapat diketahui bahwa *query expansion* dapat mempengaruhi nilai *Precision@K*. Dari skenario 1 hingga skenario 4, diperoleh nilai *Precision@K* yang cenderung naik jika menggunakan *query expansion*. Jumlah penambahan kata berpengaruh terhadap naiknya nilai *Precision@K* dan juga persebaran dokumen yang relevan. Semakin banyak kata yang ditambahkan dalam *query* awal, maka nilai *Precision@K* yang dihasilkan juga semakin tinggi, dan persebaran dokumen yang relevan juga semakin bertambah. Penambahan kata tersebut berfungsi untuk menaikkan peringkat dari dokumen yang relevan sesuai dengan kebutuhan informasi pengguna.

Perubahan nilai *K* juga mempengaruhi nilai *Precision@K* yang dihasilkan. Semakin kecil nilai *K* maka nilai *Precision@K* akan semakin tinggi namun semakin banyak dokumen relevan yang diabaikan, dan semakin besar nilai *K* maka nilai *Precision@K* akan semakin kecil, karena banyak dokumen yang tidak relevan juga ikut terhitung. Sehingga dapat diambil kesimpulan bahwa nilai *Precision@K* yang dihasilkan, berbanding terbalik dengan nilai *K* yang digunakan.

## BAB 7 PENUTUP

Bab 7 menjelaskan tentang penutup laporan yang berupa kesimpulan terhadap hasil sistem dari penelitian yang telah dilakukan beserta dengan saran untuk pengembangan sistem yang lebih lanjut.

### 7.1 Kesimpulan

Berdasarkan dari serangkaian tahapan yang telah dilakukan pada penelitian ini, maka kesimpulan yang didapatkan dari penelitian yang berjudul “*Query Expansion* Pada Sistem Temu Kembali Informasi Dokumen Jurnal Berbahasa Indonesia Menggunakan Metode BM25” adalah sebagai berikut.

1. Metode BM25 dapat digunakan untuk menerapkan *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia. Setiap dokumen yang tersedia dilakukan proses *preprocessing*, selanjutnya akan dilakukan perhitungan untuk nilai TF IDF pada setiap kata di dalam *query*, lalu dilakukan pemeringkatan dokumen dengan metode BM25, dan selanjutnya dilakukan proses untuk penentuan *query* baru yang berdasarkan pada seluruh dokumen relevan.
2. Hasil pengujian *query expansion* pada sistem temu kembali informasi dokumen jurnal berbahasa Indonesia menggunakan metode BM25 yaitu menghasilkan kenaikan rata-rata nilai *Precision@K* sebesar 0.309. Penambahan jumlah kata pada *query* awal menggunakan *query expansion* juga berpengaruh terhadap nilai *Precision@K* yang dihasilkan.

### 7.2 Saran

Saran dari hasil sistem pada penelitian ini untuk pengembangan sistem yang lebih lanjut yaitu mencari metode lain yang dapat digunakan untuk penerapan *query expansion*, sehingga kata-kata yang ditambahkan semakin sesuai dengan kebutuhan informasi pengguna dan akan mengembalikan dokumen yang lebih spesifik dan relevan.

## DAFTAR REFERENSI

- Campos, L.M.D., Fernandez, J.M. dan Huete, J.F., 2013. *Query Expansion in Information Retrieval Systems using a Bayesian Network - Based Thesaurus*. Universidad de Granada: Spain.
- Haryanto, D.J., Muflikah L. dan Fauzi M.A., 2018. *Analisis Sentimen Reviues Barang Berbahasa Indonesia dengan Metode Support Vector Machine dan Query Expansion*. Universitas Brawijaya: Malang.
- Irmawati, 2017. *Sistem Temu Kembali Informasi pada Dokumen dengan Metode Vector Space Model*. Universitas Nasional: Jakarta.
- Manning, C.D., Raghavan, P. dan Schutze, H., 2009. *An Introduction to Information Retrieval*. Cambridge University Press: Cambridge, England.
- Nasution, M.K.M., 2017. *Abstrak – Suatu Karya Ilmiah*. Universitas Sumatera Utara: Medan
- Pardede, J., Husada, M.G. dan Riansyah, R., 2015. *Implementasi Perbandingan Metode Okapi BM25 dan PLSA pada Aplikasi Information Retrieval*. Itenas Bandung: Bandung.
- Pertiwi, M.W., dan Taufiqurrochman, 2017. *Sistem Temu Kembali Informasi dalam Dokumen (Pencarian 10 Kata Kunci di Ejournal BSI)*. Nusa Mandiri: Jakarta Pusat.
- Russell, S.J., Norvig, P., 2010. *Artificial Intelligence: A Modern Approach, 3<sup>rd</sup> ed.* New Jersey.
- Sanjaya, F., 2017. *Pemanfaatan Sistem Temu Kembali Informasi dalam Pencarian Dokumen Menggunakan Metode Vector Space Model*. Sekolah Tinggi Informatika & Komputer Indonesia (STIKI): Malang.
- Selberg, E.W., 1997. *Information Retrieval Advances using Relevanxe Feedback*. University of Washington: Seattle.
- Tanjung, 2017. *Klasifikasi Tweets pada Twitter dengan Menggunakan Metode Fuzzy K-Nearest Neighbour (Fuzzy K-NN) dan Query Expansion Berbasis Apriori*. Universitas Brawijaya: Malang.