DETEKSI PLAGIARISME PADA ARTIKEL BERITA BERBAHASA INDONESIA MENGGUNAKAN BM25

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Dina Dahniawati
NIM: 155150200111081

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

PENGESAHAN

DETEKSI PLAGIARISME PADA ARTIKEL BERITA BERBAHASA INDONESIA MENGGUNAKAN BM25

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Dina Dahniawati
NIM: 155150200111081

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

Dosen Pembimbing 1	Dosen Pembimbing 2
Indriati, S.T, M.Kom	Ir. Sutrisno, M.T
NIP: 19831013 201504 2 002	NIP: 19570325 198701 1 001

Mengetahui
Ketua Jurusan **Teknik Informatika**

Tri Astoro 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 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, 21 Maret 2019

Dina Dahniawati

NIM: 155150200111081

PRAKATA

Puji syukur penulis panjatkan kepada Allah SWT atas berkat rahmat, hidayah, dan karunia-Nya kepada penulis, sehingga penulis dapat menyelesaikan skripsi dengan judul "Deteksi Plagiarisme Pada Artikel Berita Berbahasa Indonesia Menggunakan BM25".

Dalam kesempatan ini, penulis ingin mengucapkan terima kasih sebesar-besarnya kepada pihak-pihak yang telah membantu dalam menyelesaikan skripsi ini, yaitu:

1. Keluarga penulis khususnya Ayah, Ibu, adik, dan juga nenek yang selalu memberikan semangat, motivasi, perhatian, saran, serta doa yang selalu dipanjatkan.
2. Ibu Indriati, S.T, M.Kom, selaku dosen pembimbing 1 yang telah sabar dan ikhlas dalam membimbing, memberikan ilmu, saran, serta perhatian kepada penulis dalam menyelesaikan skripsi.
3. Bapak Ir. Sutrisno, M.T, selaku dosen pembimbing 2 yang juga telah sabar dan ikhlas dalam membimbing, memberikan ilmu, dan saran kepada penulis dalam menyelesaikan skripsi.
4. Bapak dan Ibu dosen yang telah memberikan ilmu kepada penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya.
5. Bapak Drs. Joko Wahono selaku pakar yang telah memvalidasi data yang digunakan dalam skripsi.
6. Sahabat-sahabat penulis yang senantiasa menyempatkan waktunya untuk mendengarkan keluh kesah penulis, memberikan saran, semangat, dan motivasi selama penulis menyelesaikan skripsi, yaitu Dama Yuliana, Isti Marlisa F, Desy Wulandari, Yuni Panca W, Sri Maryani, Septiana Dyah SP, Rena Ayudana R, Evi Febri R, Astrid Ganadya NI, Padma Jati H, dan Riska Nur A.
7. Teman-teman Kos Putri Sakinah yang telah membantu dalam berbagai situasi dan selalu memberikan perhatian kepada penulis.
8. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi.

Penulis menyadari bahwa laporan ini masih banyak kekurangan, baik dalam penulisan laporan maupun isi laporan, sehingga penulis mengharapkan adanya kritik dan saran yang dapat membangun dari semua pihak. Semoga skripsi ini dapat bermanfaat dan membawa dampak yang nyata. Terima kasih.

Malang, 21 Maret 2019

Penulis

dina.dahniawt@gmail.com

ABSTRAK

Dina Dahniawati, Deteksi Plagiarisme Pada Artikel Berita Berbahasa Indonesia Menggunakan BM25

Pembimbing: Indriati, S.T, M.Kom dan Ir. Sutrisno, M.T

Salah satu kasus yang sempat mencoreng dunia jurnalistik yaitu adanya plagiarisme yang pernah dilakukan oleh seorang wartawan terkait dengan artikel berita yang ditulisnya. Pada awalnya tindakan plagiarisme tidak diberikan pengamatan secara ketat, sehingga penggunaan kembali terhadap keseluruhan artikel berita dapat dilakukan secara bebas. Namun seiring dengan berkembangnya waktu, agensi berita tidak lagi mampu mengabaikan kasus plagiarisme, sehingga deteksi plagiarisme menjadi hal yang sangat penting untuk diterapkan. Dalam penelitian ini metode yang digunakan untuk mendeteksi plagiarisme adalah BM25. Proses perhitungan deteksi plagiarisme menggunakan BM25 diawali dengan *text preprocessing,* pencarian nilai *term frequency, inverse document frequency,* pembobotan menggunakan BM25, kemudian perhitungan persentase plagiarismenya. Pengujian dilakukan dengan mengubah nilai *threshold* sebesar 75%, 50%, dan 25%. Kemudian hasil perhitungan plagiarisme menggunakan BM25 akan dibandingkan dengan hasil dari *cosine similarity*. Hasil rata-rata dari BM25 lebih mendekati *threshold* dengan selisih sebesar 6,12%, 9,77%, dan 10,01%. Dimana hasil tersebut membuktikan bahwa BM25 bekerja lebih baik daripada *cosine similarity* yang mempunyai selisih sebesar 14,25%, 26,43%, dan 32,36% dari *threshold*. Nilai rata-rata *precision* dari metode BM25 yang diperoleh untuk masing-masing *threshold* yaitu sebesar 0,87, 0,80, dan 0,63.

Kata kunci: deteksi plagiarisme, jurnalistik, artikel berita, BM25, *cosine similarity*

ABSTRACT

Dina Dahniawati, Plagiarism Detection in Indonesian News Articles Using BM25

Supervisors: Indriati, S.T, M.Kom and Ir. Sutrisno, M.T

One of the cases that had tarnished the world of journalism was the plagiarism that had been carried out by a journalist related to the news articles he wrote. Plagiarism was not given strict observation, so that the reuse of all news articles could be carried out freely in the past. But as time goes by, news agencies are no longer able to ignore the case of plagiarism, so detection of plagiarism is very important to implement. The method used to detect plagiarism in this study is BM25. The process of calculating plagiarism using BM25 begins with text preprocessing, searching for term frequency, inverse document frequency, weighting using BM25, then calculating the percentage of plagiarism. Testing is done by changing the threshold value by 75%, 50%, and 25%. Then the results of plagiarism using BM25 will be compared with the results of cosine similarity. The average results from BM25 are closer to the threshold with a difference of 6.12%, 9.77%, and 10.01%. These results prove that BM25 works better than cosine similarity which has a difference of 14.25%, 26.43% and 32.36% of the threshold. The average value of precision from BM25 for each threshold is 0.87, 0.80, and 0.63.

Keywords: *plagiarism detection, journalism, news articles, BM25, cosine similarity*

DAFTAR ISI

DAFTAR TABEL

DAFTAR KODE PROGRAM

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1 PENDAHULUAN

Bab ini berisi latar belakang permasalahan yang diambil, rumusan masalah yang akan dipecahkan, tujuan yang akan dicapai, manfaat penelitian yang dihasilkan, batasan masalah yang digunakan, dan sistematika pembahasan laporan.

1.1 Latar Belakang

Kecenderungan masyarakat dalam mengakses berbagai informasi melalui internet mengakibatkan eksistensi informasi pada media cetak seperti koran dan majalah meredup. Seperti sekarang ini telah banyak *website* berita *online* yang menyediakan berbagai macam kategori berita dan dipublikasikan secara cepat seperti kompas.com, cnnindonesia.com, liputan6.com, dan *website* berita lainnya. Tentunya hal tersebut sangat membantu masyarakat dalam mengakses berita secara mudah, cepat, dan praktis, sehingga hal ini dapat menjadi salah satu manfaat dari perkembangan teknologi saat ini. Namun tidak selamanya perkembangan teknologi membawa sisi positif untuk masyarakat. Contoh yang paling nyata adalah seringnya penyalahgunaan dalam memakai fungsi *copy-paste* atau salin-tempel yang ada di setiap komputer. Perbuatan tersebut merupakan salah satu penyebab maraknya kasus plagiarisme. Plagiarisme merupakan tindakan berupa penerbitan, penyalahgunaan, pernyataan, perampasan ide atau pemikiran orang lain, dan perampasan tulisan atau hasil karya orang lain sebagai milik diri sendiri. Plagiarisme dapat dikatakan pula sebagai bentuk penghinaan terhadap seseorang atau institusi yang menjadi korban dan dapat merusak nilai-nilai yang dipegang (Kock dan Davison, 2003).

Kasus plagiarisme sudah sering terjadi di berbagai bidang, mulai dari pendidikan hingga jurnalistik. Plagiarisme dalam dunia jurnalistik terutama pada penulisan artikel berita terdengar aneh karena orang berpikir bahwa artikel berita tidak bisa diplagiat atau dijiplak dan wartawan pasti terikat dengan adanya kode etik jurnalistik. Tetapi apabila dilihat lebih teliti, ada saja wartawan ataupun penulis berita di sebuah *website* yang menulis berita sama persis dengan berita dari *website* lain tanpa mencantumkan sumbernya. Terlepas dari ada tidaknya kerjasama antar agensi berita dalam mengelola *website*, berita yang dipublikasikan haruslah sesuai fakta, terpercaya, dan jelas sumber beritanya. Salah satu kasus plagiarisme dalam dunia jurnalistik adalah seorang wartawan dari sebuah koran ternama di Amerika yang melakukan penjiplakan atas artikel berita yang ditulis oleh wartawan lain dan menjadikannya seolah-olah artikel tersebut merupakan hasil karyanya.

Dalam dunia jurnalistik, kasus plagiarisme tidak diberikan pengamatan secara ketat sehingga penggunaan kembali terhadap keseluruhan artikel berita dapat dilakukan secara bebas dan berlebihan. Kasus tersebut terus berulang dan merupakan sebuah praktik umum yang dianggap biasa saja di masa lalu (Kienreich et al., 2006). Namun seiring dengan perkembangan teknologi, agensi berita tidak lagi mampu mengabaikan kasus plagiarisme tersebut. Maka dari itu deteksi

plagiarisme menjadi suatu hal yang pokok dan sangat penting untuk sebuah agensi berita. Seperti sekarang ini telah banyak aplikasi yang dapat digunakan untuk mendeteksi plagiarisme dalam berbagai konten, seperti aplikasi *online* Turnitin, Unplag, WriteCheck, Copyscape, dan aplikasi deteksi plagiarisme lainnya.

Terdapat beberapa penelitian mengenai deteksi plagiarisme yang telah dilakukan, salah satunya menggunakan metode *fingerprinting* dengan algoritme himpunan kata. Nilai potensi yang didapatkan dari ukuran kemiripan dokumen uji dengan dokumen yang ada di korpus yaitu sebesar 0,4 untuk paragraf yang ditulis ulang, nilai potensi sebesar 0,6 untuk paragraf yang ada pengubahan kata atau tanda baca, dan nilai potensi sebesar 0,8 untuk paragraf yang sama persis (Ismail dan Yunarso, 2014). Selain itu terdapat pula penelitian yang membandingkan metode BM25 dengan beberapa metode yaitu *cosine, identity, PlagiRank,* serta sistem deteksi plagiarisme JPlag dan MOSS. Hasil yang didapatkan dari penelitian tersebut menyatakan bahwa metode BM25 adalah metode yang paling efektif dalam mendeteksi plagiarisme dan hasil perhitungannya sebanding dengan sistem deteksi plagiarisme JPlag (Burrows, Tahaghoghi dan Zobel, 2007).

Berdasarkan permasalahan dan penelitian terkait yang telah dijelaskan, maka pada penelitian ini akan dibuat deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan metode BM25. Deteksi plagiarisme ini nantinya akan menampilkan nilai persentase plagiarisme antar artikel berita. Perhitungan plagiarisme artikel-artikel tersebut akan dimulai dengan mengolah teks atau *text preprocessing,* menghitung nilai kemiripan artikel menggunakan metode BM25, kemudian menghitung persentase plagiarisme antar artikel. Untuk mengetahui kinerja metode BM25 dalam mendeteksi plagiarisme akan dilakukan dengan membandingkan antara hasil perhitungan metode BM25 dan *cosine similarity* serta pemotongan artikel berdasarkan nilai *threshold.* Metode *cosine similarity* dipilih karena sudah banyak digunakan dalam menghitung kemiripan antar dokumen, sedangkan nilai *threshold* digunakan untuk membatasi panjang artikel yang akan diuji dan menjadi patokan dalam menganalisis hasil perhitungan plagiarisme yang didapatkan dari metode BM25 dan *cosine similarity*.

Dengan demikian deteksi plagiarisme ini diharapkan dapat membantu dalam menghitung nilai persentase plagiarisme dalam penulisan artikel berita. Dengan adanya penelitian ini maka masyarakat ataupun wartawan pada khususnya diharapkan dapat lebih sadar dalam menghargai karya orang lain dan benar-benar menaati peraturan dalam kode etik jurnalistik.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan metode BM25 adalah sebagai berikut:

1. Bagaimana melakukan pengujian terhadap deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan metode BM25?
2. Bagaimana hasil pengujian yang diperoleh dari penggunaan metode BM25 dalam deteksi plagiarisme pada artikel berita berbahasa Indonesia?

3. Bagaimana perbandingan antara metode BM25 dengan metode *cosine similarity* dalam deteksi plagiarisme pada artikel berita berbahasa Indonesia?

1.3 Tujuan

Tujuan dari penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 adalah sebagai berikut:

1. Melakukan pengujian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan metode BM25 dengan mengubah nilai *threshold* artikel berita.
2. Mengetahui hasil pengujian yang diperoleh dari deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan metode BM25 berdasarkan perubahan nilai *threshold*.
3. Membandingkan hasil plagiarisme menggunakan metode BM25 dengan metode *cosine similarity* dalam deteksi plagiarisme pada artikel berita berbahasa Indonesia.

1.4 Manfaat

Manfaat dari penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 adalah sebagai berikut:

1. Dapat membedakan metode mana yang lebih baik antara BM25 dan *cosine similarity* dalam deteksi plagiarisme pada artikel berita berbahasa Indonesia.
2. Dapat mempermudah dalam menghitung nilai persentase plagiarisme pada artikel berita berbahasa Indonesia.

1.5 Batasan Masalah

Batasan masalah dari penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 adalah sebagai berikut:

1. Data yang digunakan adalah artikel berita berbahasa Indonesia dari berbagai *website* penyedia berita.
2. Artikel berita yang digunakan merupakan artikel yang terbit antara bulan Agustus 2018 hingga bulan Maret 2019.
3. Kata kutipan dari narasumber yang ada di dalam artikel tidak digunakan.

1.6 Sistematika Pembahasan

Sistematika pembahasan dari penulisan laporan pada penelitian ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang permasalahan yang diambil, rumusan masalah yang akan dipecahkan, tujuan yang akan dicapai, manfaat penelitian yang dihasilkan, batasan masalah yang digunakan, serta sistematika pembahasan.

BAB II LANDASAN KEPUSTAKAAN

Bab ini berisi teori-teori yang sudah ada atau dari penelitian terdahulu yang digunakan sebagai dasar dan pedoman dalam penelitian ini.

BAB III METODOLOGI

Bab ini berisi penjelasan tentang tipe penelitian, strategi penelitian, objek dan lokasi penelitian, pengumpulan dan teknik analisis data, serta peralatan pendukung yang digunakan.

BAB IV PERANCANGAN

Bab ini berisi perancangan algoritme, pengujian, serta contoh perhitungan manualisasi dari metode yang digunakan.

BAB V IMPLEMENTASI

Bab ini berisi kode program dan pembahasannya yang telah dibuat berdasarkan perancangan.

BAB VI PENGUJIAN DAN ANALISIS

Bab ini berisi hasil dan analisis dari pengujian yang telah dilakukan.

BAB VII PENUTUP

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran yang digunakan untuk memperbaiki penelitian kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini menjelaskan dasar teori yang telah ada atau dari penelitian terdahulu yang digunakan sebagai pedoman dalam melakukan penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25.

2.1 Kajian Pustaka

Pedoman atau dasar dalam penelitian ini diambil dari beberapa penelitian terdahulu yang mempunyai ruang lingkup sama atau hampir mendekati dengan penelitian yang dilakukan. Penelitian pertama yang menjadi dasar dari penelitian ini dilakukan oleh Burrows, Tahaghoghi dan Zobel (2007) dengan judul "*Efficient Plagiarism Detection for Large Code Repositories*". Penelitian tersebut membandingkan beberapa metode yang digunakan untuk deteksi plagiarisme dengan sistem deteksi plagiarisme yang sudah ada. Nilai *precision* dan *recall* yang didapatkan dari metode BM25 yaitu sebesar 60% dan 30%. Dalam penelitian tersebut juga disebutkan bahwa metode BM25 adalah metode yang paling efektif dengan hasil berupa enam dokumen teratas merupakan dokumen yang benar-benar terdapat plagiarisme, dengan kata lain keenam dokumen tersebut adalah pasangan dokumen asli dan jiplakan. Dan hasil pengujian antara metode BM25 dengan sistem deteksi plagiarisme JPlag mendapatkan nilai yang sebanding.

Penelitian kedua dilakukan oleh Kienreich et al., (2006) dengan judul "*Plagiarism Detection in Large Sets of Press Agency News Articles*". Dalam penelitian tersebut dijelaskan bahwa seiring dengan berkembangnya waktu dan teknologi, maka kasus plagiarisme dalam artikel berita tidak dapat didiamkan begitu saja. Penanganan plagiarisme dilakukan dengan mendeteksi setiap artikel yang terindikasi plagiarisme menggunakan sebuah himpunan awal berisi artikel yang dianggap mirip, dimana pengelompokan artikel berita berdasarkan sketsa dari artikel berita tersebut. Kemudian himpunan artikel tersebut dianalisis menggunakan *fuzzy sentence analysis* berdasarkan *double Levensthein metric* untuk mengetahui plagiarismenya.

Penelitian selanjutnya dilakukan oleh Ismail dan Yunarso (2014) dengan judul "Aplikasi Berbasis Web Pendeteksi Plagiarisme Menggunakan Algoritme Himpunan Kata". Penelitian tersebut menggunakan metode *fingerprinting* dengan algoritme himpunan kata dan mendapatkan nilai potensi sebesar 0,4 untuk paragraf yang ditulis ulang, 0,6 untuk paragraf yang ada pengubahan tanda baca atau kata, dan sebesar 0,8 untuk paragraf yang sama persis. Nilai potensi tersebut didapatkan dari nilai ukuran kemiripan dokumen uji dengan dokumen yang sudah ada di korpus. Dalam penelitiannya, himpunan kata yang digunakan berupa *paragraph chunking*, dimana tiap paragraf dijadikan satu kesatuan utuh atau satu himpunan. Tiap *chunking* tersebut akan dibandingkan dengan *chunking* lainnya yang sudah ada didalam *database.* Penggunaan *paragraph chunking* lebih disebabkan oleh asumsi peneliti bahwa plagiarisme biasanya dilakukan dalam satuan paragraf serta penggunaannya yang lebih efektif.

Penelitian selanjutnya dilakukan oleh organisasi iThenticate pada tahun 2013 dengan judul "*Research Ethics: Decoding Plagiarism and Attribution in Research*". Penelitian tersebut dilakukan dengan mengadakan survei secara *online* terhadap para peneliti mengenai pengetahuan, pemahaman, dan pengalaman mereka tentang berbagai bentuk plagiarisme. Dari penelitian tersebut didapatkan lima tipe plagiarisme yang dianggap serius, yaitu *complete* sebesar 88%, *verbatim* sebesar 84%, *unethical collaboration* dan *misleading attribution* yang masing-masing sebesar 82%, dan *replication* sebesar 77%. Dari berbagai macam bentuk plagiarisme yang ada, terdapat bentuk plagiarisme yang sulit untuk dideteksi yaitu *unethical collaboration*, *misleading attribution,* dan *replication.* Sedangkan bentuk yang mudah dan cocok digunakan dengan aplikasi deteksi plagiarisme yaitu *verbatim* dan *paraphrasing*.

2.2 Jurnalistik

Dalam Kamus Besar Bahasa Indonesia (KBBI), jurnalistik merupakan suatu hal yang berhubungan dengan kewartawanan atau persuratkabaran. Berdasarkan dari bentuk dan pengelolaannya, jurnalistik dibagi menjadi tiga (Juwito, 2008), yaitu:

1. Jurnalistik media cetak yang meliputi surat kabar harian, surat kabar mingguan, tabloid harian, tabloid mingguan, dan majalah.
2. Jurnalistik media elektronik *auditif* meliputi siaran radio.
3. Jurnalistik media *audiovisual* yang meliputi siaran televisi dan media *online*.

Bidang jurnalistik sangat berperan penting dalam memenuhi hak masyarakat untuk memperoleh berita yang benar. Maka dari itu di dalam bidang jurnalistik terdapat pedoman yang harus ditaati yaitu kode etik jurnalistik. Dalam penafsiran Pasal 2 Kode Etik Jurnalistik Poin G disebutkan bahwa wartawan tidak boleh melakukan plagiarisme. Plagiarisme yang dimaksud yaitu termasuk menyatakan hasil liputan wartawan lain sebagai hasil karyanya sendiri.

2.3 Berita

Dalam Kamus Besar Bahasa Indonesia (KBBI), berita adalah suatu cerita atau keterangan mengenai kejadian atau peristiwa yang sedang hangat dibicarakan. Berita dapat disampaikan melalui surat kabar atau media cetak seperti koran dan majalah, selain itu berita juga dapat disiarkan melalui televisi, radio, atau alat komunikasi lainnya, bahkan seiring dengan berkembangnya teknologi maka berita sudah dapat diakses secara *online.* Tetapi dalam penulisan berita tidak boleh dilakukan secara sembarangan, ada hal-hal yang harus diperhatikan oleh penulis (Juwito, 2008), yaitu:

1. Unsur penulisan berita yang dikenal dengan 5W + 1H*,* yaitu merujuk pada apa, siapa, mengapa, kapan, dimana, dan bagaimana.
2. Syarat-syarat yang harus dipenuhi dalam penulisan berita yaitu berdasarkan kenyataan atau fakta, merupakan berita terbaru yang masih hangat dibicarakan

atau ketepatan waktu dalam penulisan, menyangkut kepentingan orang banyak, dan dapat menarik pembaca.
3. Struktur dan teknik penulisan berita.

2.4 Plagiarisme

Tindakan plagiarisme sangat sering terjadi dan memakan banyak korban. Tetapi para pelaku seperti tidak sadar atas apa yang telah dilakukannya adalah perbuatan yang salah besar dan dapat dikatakan sebagai pelanggaran hak cipta dan diatur di dalam Undang-Undang. Berikut akan dijelaskan tentang pengertian plagiarisme lebih lanjut.

2.4.1 Pengertian Plagiarisme

Plagiarisme menurut Kamus Besar Bahasa Indonesia (KBBI) adalah penjiplakan yang melanggar hak cipta. Dalam Undang-Undang Nomor 19 Tahun 2002, hak cipta adalah hak eksklusif bagi seseorang yang menghasilkan suatu karya berdasarkan pikiran, imajinasi, keterampilan, dan kecekatan yang dituangkan dalam bentuk yang khas. Disebutkan pula pada Bagian Keempat Pasal 12, bahwa ciptaan yang dilindungi adalah ciptaan dalam bidang ilmu pengetahuan, seni, dan sastra, yang salah satunya mencakup karya tulis yang diterbitkan dan semua hasil karya tulis orang lain. Tak hanya itu, plagiarisme merupakan suatu tindak kejahatan yang dilakukan tanpa adanya kekerasan fisik secara nyata namun dapat berakibat sangat fatal. Bahkan plagiarisme yang dilakukan secara penuh dan berulang-ulang dapat mengakibatkan matinya kreativitas seseorang. Tindakan plagiarisme juga merupakan salah satu perbuatan yang tidak terpuji dan dapat merugikan orang lain maupun diri sendiri.

Berdasarkan penelitian yang dilakukan oleh organisasi iThenticate pada tahun 2013, terdapat sepuluh tipe plagiarisme yaitu:

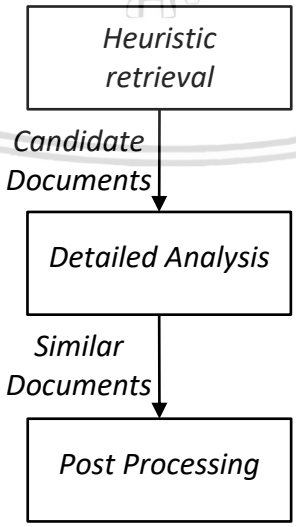
1. *Duplication* yaitu menggunakan kembali jurnal atau laporan dari penelitian terdahulu tanpa adanya atribusi.
2. *Replication* yaitu pengumpulan jurnal dengan berbagai publikasi.
3. *Paraphrasing* yaitu mengambil kata dan menggunakannya bersama teks yang asli.
4. *Verbatim* yaitu menyalin kata tanpa memberikan tanda adanya sitasi.
5. *Misleading attribution* yaitu berupa penghapusan nama pencipta.
6. *Invalid source* yaitu sumber referensi yang digunakan tidak ada atau tidak bisa dicari.
7. *Secondary source* yaitu hanya menulis sumber sekunder.
8. *Unethical collaboration* yaitu menggunakan tulisan satu sama lain dalam suatu kelompok.
9. *Repetitive research* yaitu menggunakan kembali data dan metodologi yang sama dari penelitian sebelumnya tanpa atribusi yang tepat.
10. *Complete* yaitu mengambil semua karya seseorang dan mengklaimnya sebagai milik sendiri.

Sanksi yang diberikan kepada pelaku plagiarisme atau yang biasa disebut dengan plagiator, khususnya di tingkat pendidikan tinggi universitas diatur dalam Peraturan Menteri Pendidikan RI Nomor 17 Tahun 2010 Pasal 12, yaitu dapat berupa sanksi yang paling ringan yakni teguran, peringatan tertulis, penundaan pemberian sebagian hak, pembatalan nilai mata kuliah, dan bahkan bisa sampai dengan pemberhentian dengan hormat maupun tidak hormat dari status mahasiswa, serta pembatalan ijazah. Terdapat beberapa faktor yang menyebabkan masalah plagiarisme semakin banyak (Wijaya, 2017), yaitu:

1. Tingginya rasa malas untuk mencari dan membaca referensi baik berupa buku maupun jurnal yang berhubungan dengan penelitian.
2. Masih kurangnya pengetahuan terhadap pemahaman dalam melakukan kutipan seperti mengutip dari sumber sekunder tanpa mencari sumber primernya.
3. Kurangnya melatih pikiran dan logika untuk melakukan analisis dari penelitian yang sudah dilakukan.
4. Kurangnya pemahaman, pengetahuan, atau wawasan mengenai objek penelitian.
5. Adanya keterbatasan waktu dalam menyelesaikan penelitian.

2.4.2 Sistem Deteksi Plagiarisme

Semakin ramainya kasus plagiarisme, maka semakin banyak pula sistem aplikasi yang digunakan untuk mendeteksi plagiarisme seperti Turnitin, Unplag, Writecheck, dan Copyscape. Sistem deteksi plagiarisme tersebut bekerja dengan membandingkan sekumpulan teks antar beberapa dokumen untuk menemukan kemiripannya. Tahap utama dalam sistem deteksi plagiarisme yaitu *heuristic retrieval, detailed analysis,* dan *post processing* (Sanjalawe dan Anbar, 2017). Tahap utama sistem deteksi plagiarisme dapat dilihat pada Gambar 2.1.



Gambar 2.1. Tahap Utama Sistem Deteksi Plagiarisme

Sumber: (Sanjalawe dan Anbar, 2017)

Tahap pertama yang dilakukan dalam deteksi plagiarisme adalah identifikasi kandidat atau dokumen di dalam korpus yang terindikasi ada plagiarisme dengan menggunakan pendekatan heuristik. Dari hasil dokumen yang telah didapatkan kemudian dihitung kemiripan antar semua dokumen-dokumennya menggunakan metode yang dipilih oleh peneliti. Dan tahap terakhir adalah sistem harus dapat menampilkan atau menerapkan hasil berdasarkan proses yang telah dilakukan. Dalam *post processing* akan ditampilkan persentase seberapa tinggi tingkat plagiarisme dari suatu artikel berita terhadap artikel lainnya.

Nilai persentase didapatkan dari perbandingan antara bobot dokumen yang terindikasi plagiarisme dengan bobot dokumen yang sama tetapi dengan *threshold* 100% atau dokumen asli yang tidak ada perubahan. Nilai persentase plagiarisme dapat dihitung menggunakan rumus pada Persamaan 2.1.

$$Plagiarisme_{d_i} = \frac{bobot_{d_i}}{bobot100\%_{d_i}} \times 100\% \quad (2.1)$$

Keterangan:
$Plagiarisme_{d_i}$: Nilai plagiarisme pada dokumen *d* ke-*i*
$bobot_{d_i}$: Bobot dokumen *d* ke-*i* sebagai *query*
$bobot100\%_{d_i}$: Bobot asli dokumen *d* ke-*i* di dalam korpus

2.5 *Text Mining*

Text mining adalah sebuah proses untuk mengekstrak informasi-informasi penting dari sumber data atau koleksi dokumen dengan melakukan identifikasi dan penelusuran pola pada teks yang tidak terstruktur (Feldman dan Sanger, 2007). *Text mining* telah banyak digunakan dalam berbagai kasus yang berkaitan dengan teks seperti pada *information retrieval, clustering,* dan *information extraction*. Dalam *text mining* terdapat beberapa tahap untuk mengolah teks (Dang dan Ahmad, 2014), yaitu:

1. Pengumpulan informasi dari data yang tidak terstruktur.
2. Mengubah informasi dari data yang tidak terstruktur menjadi data yang terstruktur.
3. Melakukan pengenalan pola terhadap data terstruktur.
4. Menganalisis pola dari data terstruktur.
5. Mengekstrak informasi dari data terstruktur.
6. Menyimpan informasi yang telah didapatkan ke dalam *database.*

2.6 *Text Preprocessing*

Dalam konteks *text mining, text preprocessing* digunakan untuk mengolah teks sebelum dilakukan pemrosesan yang lebih lanjut atau perhitungan pembobotan kata. *Text preprocessing* terdiri dari beberapa proses, yaitu tokenisasi, *case folding, cleaning*, *stopword removal,* dan *stemming.* Proses dalam *text preprocessing* sama sekali tidak terikat dengan urutan karena tidak mempengaruhi hasil *text preprocessing* nantinya. Urutan proses tersebut juga dapat disesuaikan dengan aplikasi apa yang akan dibuat. Tetapi ketika *text*

preprocessing tidak diikutsertakan dalam pengolahan teks, maka data dan analisis yang dihasilkan akan sangat tidak konsisten dan tidak baik (Kalra dan Aggarwal, 2017).

2.6.1 Tokenisasi

Tokenisasi merupakan proses dalam *text preprocessing* yang bertujuan untuk mencari kata atau token. Tokenisasi dapat diartikan pula sebagai proses memisahkan tiap-tiap kata yang kemudian disebut sebagai token (Manning, Raghavan dan Schutze, 2009). Token yang dihasilkan dari proses tokenisasi akan digunakan sebagai dasar dalam perhitungan selanjutnya.

2.6.2 *Case Folding*

Case folding merupakan proses dalam *text preprocessing* yang bertujuan untuk mengubah huruf besar menjadi huruf kecil. Proses ini dilakukan untuk menyeragamkan semua kata di dalam teks dan mempermudah pencarian (Manning, Raghavan dan Schutze, 2009).

2.6.3 *Cleaning*

Cleaning merupakan proses dalam *text preprocessing* yang bertujuan untuk menghapus tanda baca, angka, tag html, *link,* dan yang lainnya. Proses ini dapat diartikan sebagai proses penyaringan kata tanpa perlu memperhatikan tanda baca yang ada. Sehingga hasil dari *cleaning* ini berupa kumpulan kata tanpa adanya tanda baca, angka, tag html, ataupun *link* yang memisahkan.

2.6.4 *Stopword Removal*

Stopword removal merupakan proses dalam *text preprocessing* yang bertujuan untuk menghapus kata-kata yang tidak mempunyai makna. Penghapusan ini berdasarkan kamus yang berisi daftar kata yang tidak penting seperti kata hubung dan kata ganti. Terdapat pula *stop list* atau daftar kata berhenti yang ditentukan berdasarkan frekuensi kata tersebut, yaitu kata yang tidak termasuk dalam konteks tetapi sering muncul. Kemudian kata di dalam daftar *stop list* tersebut yang akan dihapus dalam proses *stopword removal.* Penggunaan *stop list* dapat memperkecil jumlah informasi yang akan disimpan (Manning, Raghavan dan Schutze, 2009). Kata-kata yang masuk dalam daftar tersebut dianggap tidak penting dan tidak mempunyai makna, sehingga ketika dihapus tidak akan memberikan efek yang yang besar terhadap teks yang akan diolah.

2.6.5 *Stemming*

Stemming merupakan proses dalam *text preprocessing* yang bertujuan untuk mencari kata berimbuhan dan mengembalikan kata tersebut ke bentuk dasarnya. Selain itu, *stemming* dapat mengurangi bentuk infleksi dan biasanya mengacu pada heuristik yang kasar, yaitu dengan memotong ujung suatu kata (Manning, Raghavan dan Schutze, 2009). Dalam *stemming* teks Bahasa Indonesia terdapat *library* Sastrawi yang dapat digunakan untuk membantu menemukan kata-kata

yang berimbuhan untuk dihapus imbuhannya dan kemudian mengembalikan kata tersebut ke kata dasarnya.

2.7 Metode BM25

Ketika sebuah kata atau *term* muncul di dalam dokumen dengan jumlah yang banyak, maka *term* tersebut bisa jadi merupakan representasi dari dokumen tersebut. Hal itulah yang kemudian menjadikan adanya hubungan antara dokumen dengan relevansi *query* (Robertson dan Zaragoza, 2009). Salah satu metode yang dapat digunakan untuk mengetahui adanya hubungan antara dokumen dengan *query* yaitu metode BM25. BM25 digunakan untuk menghitung bobot dari setiap kata di dalam dokumen berdasarkan *query* dan menjumlahkan semua nilai bobot kata untuk memperoleh total bobot dokumen. Dalam metode BM25 terdapat 3 faktor utama yang mempengaruhi nilai bobot, yaitu *term frequency, inverse document frequency,* dan panjang dokumen (Russel dan Norvig, 2010). Rumus metode BM25 dapat dilihat pada Persamaan 2.2.

$$BM25_{(dj,q1:N)} = \sum_{i \in 1}^{N} IDF_{(qi)} \cdot \frac{TF_{(qi,dj)} \cdot (k+1)}{TF_{(qi,dj)} + k \cdot \left(1 - b + b \cdot \frac{|d_j|}{L}\right)} \quad (2.2)$$

Keterangan:
N : Jumlah dokumen di dalam korpus
IDF_{qi} : *Inverse document frequency* dari kata q_i
$TF_{qi,dj}$: *Term frequency* dari kata q_i di dokumen d_j
k, b : Parameter untuk evaluasi
$|d_j|$: Panjang dokumen d_j
L : Rata-rata panjang dokumen di dalam korpus

Parameter *k* dan *b* merupakan parameter yang dapat diubah. Dari beberapa percobaan, nilai parameter *b* yang cukup baik yaitu berada pada 0,5 < *b* < 0,8, sedangkan nilai parameter *k* berada pada 1,2 < *k* < 2. Tetapi nilai dari parameter *b* dan *k* tidak selamanya berpedoman pada nilai terbaik yang telah didapatkan tersebut karena nilai optimal juga dapat tergantung pada faktor lain seperti jenis *query* atau dokumen (Robertson dan Zaragoza, 2009).

2.7.1 *Term Frequency*

Term Frequency (TF) adalah proses pertama yang dilakukan untuk menghitung bobot suatu kata atau *term* yang ditetapkan berdasarkan jumlah kemunculan kata *t* di dalam dokumen *d* (Manning, Raghavan dan Schutze, 2009). Konsep utama dari TF adalah menambah nilai bobot kata sebanyak 1 ketika kata tersebut berulang atau muncul setelahnya.

2.7.2 *Inverse Document Frequency*

Inverse Document Frequency (IDF) adalah nilai kebalikan dari jumlah *Document Frequency* (DF). Nilai DF didapatkan dari jumlah dokumen di dalam korpus yang mengandung suatu kata yang sama. Ketika suatu kata mempunyai nilai DF yang

besar maka nilai IDF yang didapatkan akan semakin kecil. Dengan kata lain, kata yang sering muncul di setiap dokumen akan mempunyai nilai IDF yang lebih kecil daripada kata yang jarang muncul di dokumen (Manning, Raghavan dan Schutze, 2009). Persamaan IDF untuk metode BM25 berbeda dengan persamaan IDF pada TF-IDF *weighting.* Persamaan yang digunakan untuk menghitung IDF dalam metode BM25 terdapat pada Persamaan 2.3.

$$IDF_{qi} = log \frac{N - DF_{qi} + 0{,}5}{DF_{qi} + 0{,}5} \quad (2.3)$$

Keterangan:
IDF_{qi} : *Inverse document frequency* dari kata q_i
N : Jumlah dokumen di dalam korpus
DF_{qi} : *Document frequency* dari kata q_i

2.8 Metode *Cosine Similarity*

Cosine similarity atau *cossim* adalah salah satu metode yang digunakan untuk melakukan perhitungan kemiripan antar dokumen dengan cara pembobotan *term frequency-inverse document frequency* atau TF-IDF serta pemodelan ruang vektor. Semakin besar hasil dari perhitungan *cosine similarity* maka semakin mirip pula dokumen yang diujikan (Herwijayanti, Ratnawati dan Muflikhah, 2018).

Langkah pertama yaitu menghitung *term frequency* dari suatu *term* di masing-masing dokumen. Kemudian dari nilai tersebut akan dihitung bobot *term frequency*-nya. Persamaan bobot TF yang digunakan dalam *cosine similarity* dapat dilihat pada Persamaan 2.4. Ketika nilai kemunculan dari suatu *term* di suatu dokumen lebih dari 0 maka dihitung menggunakan persamaan *log,* sedangkan untuk nilai kemunculan lainnya akan langsung diberi nilai 0.

$$TF_{t,d} = \begin{cases} 1 + log\ tf_{t,d} & , \quad untuk\ \ tf_{t,d} > 0 \\ 0 & , \quad \text{lainnya} \end{cases} \quad (2.4)$$

Keterangan:
$TF_{t,d}$: Bobot *term frequency* dari kata *t* di dalam dokumen *d*
$tf_{t,d}$: *Term frequency* dari kata *t* di dalam dokumen *d*

Setelah mendapatkan nilai *term frequency* dan bobot TF, maka diperlukan nilai IDF yang dapat dihitung dengan Persamaan 2.5.

$$IDF_t = log \frac{N}{DF_t} \quad (2.5)$$

Keterangan:
IDF_t : *Inverse document frequency* dari kata *t*
N : Jumlah dokumen di dalam korpus
DF_t : *Document frequency* dari kata *t*

Setelah mendapatkan nilai TF-IDF, langkah selanjutnya yaitu menghitung bobot TF-IDF dengan Persamaan 2.6

$$W_{td,tf} = TF_{t,d} \cdot IDF_t \quad (2.6)$$

Keterangan:
$W_{td,tf}$: Bobot kata *t* di dokumen *d* dengan frekuensi *f*
$TF_{t,d}$: Bobot *term frequency* dari kata *t* di dalam dokumen *d*
IDF_t : *Inverse document frequency* dari kata *t*

Kemudian normalisasi *term* pada masing-masing dokumen dengan Persamaan 2.7.

$$W_{t,d} = \frac{W_{t,d}}{\sqrt{\sum_{t=1}^{n} W_{t,d}^2}} \quad (2.7)$$

Keterangan:
$W_{t,d}$: Bobot kata *t* di dokumen *d*
n : Banyaknya *term* di dalam suatu dokumen

Lagkah terakhir yaitu menghitung nilai kemiripan antar dokumen dengan Persamaan 2.8.

$$cossim(d_1, d_2) = \sum W_{d1} \cdot W_{d2} \quad (2.8)$$

Keterangan:
$cossim(d_1, d_2)$: Bobot *cosine similarity* antara dokumen d_1 dan d_2
W_{d1} : Bobot dokumen d_1
W_{d2} : Bobot dokumen d_2

2.9 *Precision*

Precision merupakan suatu perhitungan yang digunakan untuk evaluasi sistem dan telah banyak digunakan dalam sistem yang berhubungan dengan *Information Retrieval*. Nilai *precision* dalam deteksi plagiarisme diperoleh dari jumlah kalimat yang terindikasi plagiasi berdasarkan hasil sistem yang sama dengan kalimat yang telah dianalisis oleh pakar dibagi dengan jumlah kalimat dari hasil sistem.

BAB 3 METODOLOGI

Bab ini menjelaskan tentang tipe penelitian yang digunakan, strategi penelitian, objek dan lokasi penelitian, pengumpulan data, analisis data, serta peralatan pendukung yang digunakan dalam penelitian ini.

3.1 Tipe Penelitian

Deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 menggunakan tipe penelitian non-implementatif dengan pendekatan analitik. Tipe penelitian non-implementatif adalah suatu tipe penelitian yang mengkaji atau mempelajari dan menganalisis hubungan antar suatu kejadian. Dari analisis tersebut kemudian didapatkan sebuah hasil penyelidikan atau bisa disebut dengan hasil analisis ilmiah dari penelitian yang telah dilakukan tanpa adanya produk nyata atau perancangannya. Sedangkan pendekatan analitik digunakan untuk menjelaskan derajat hubungan antar elemen yang ada dalam objek penelitian dengan kejadian atau fenomena yang sedang diteliti.

3.2 Strategi Penelitian

Strategi penelitian merupakan suatu cara yang dilakukan peneliti dalam melakukan penelitiannya. Dalam penelitian ini menggunakan studi kasus kumulatif dengan mengumpulkan informasi yang ada dari berbagai sumber dan kemudian menganalisisnya untuk mendapatkan suatu bahasan atau kesimpulan yang kemudian akan digunakan sebagai topik kajian dalam penelitian ini.

3.3 Objek dan Lokasi Penelitian

Objek penelitian dalam deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 adalah agensi berita yang menyediakan layanan berita secara *online* seperti kompas.com, cnnindonesia.com, liputan6.com, dan agensi berita lainnya atau *website* penyedia berita yang tidak termasuk dalam agensi berita resmi. Dari berbagai artikel berita, beberapa artikel akan dipilih melalui analisis yang dilakukan di Laboratorium Komputasi Cerdas Fakultas Ilmu Komputer Universitas Brawijaya untuk kemudian digunakan sebagai kumpulan artikel atau dokumen dalam penelitian ini.

3.4 Pengumpulan Data

Data yang digunakan untuk penelitian ini didapat dari artikel-artikel berita berbahasa Indonesia dengan berbagai macam topik dan sumber. Artikel berita tersebut didapatkan langsung dari berbagai *website* penyedia konten berita seperti kompas.com, cnnindonesia.com, liputan6.com, dan *website* berita lainnya. Artikel berita yang sudah didapatkan kemudian disimpan dalam bentuk teks dengan format *.txt* dan disimpan dalam satu folder yang kemudian disebut sebagai korpus.

Dalam deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25, kutipan dari narasumber akan dihapus karena kutipan tersebut tidak perlu diubah dan berperan penting untuk membuktikan bahwa berita tersebut benar adanya atau tidak dimanipulasi dan dapat dipercaya. Untuk pemotongan artikel berdasarkan *threshold* yang akan digunakan sebagai *query* dilakukan berdasarkan panjang artikel yang telah disetujui oleh pakar.

3.5 Teknik Analisis Data

Artikel berita yang sudah didapatkan kemudian dipilih berdasarkan adanya kemiripan judul dan isi berita. Dari artikel tersebut akan dilakukan penghapusan kutipan dari narasumber. Kemudian artikel tersebut diolah untuk mengetahui hasil persentase plagiarisme. Pengujian yang akan dilakukan pada penelitian ini yaitu dengan membandingkan hasil perhitungan menggunakan metode BM25 dengan *cosine similarity.* Hasil perbandingan tersebut akan digunakan untuk mengetahui metode mana yang lebih efektif untuk digunakan dalam deteksi plagiarisme.

3.6 Peralatan Pendukung

Peralatan pendukung meliputi kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*). Peralatan pendukung tersebut digunakan untuk mengetahui spesifikasi kebutuhan perangkat apa saja yang harus disiapkan dalam penelitian. Peralatan pendukung yang digunakan dalam penelitian ini adalah sebagai berikut:

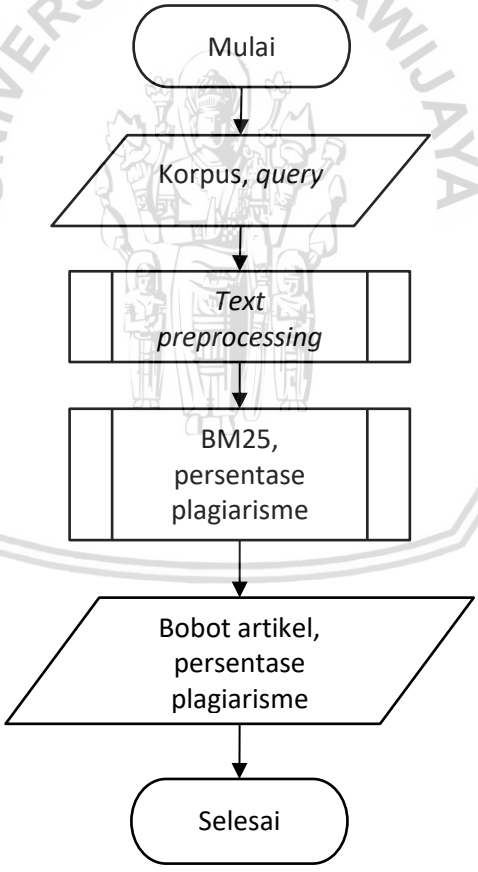
1. Kebutuhan perangkat keras:
   - Intel Core i5-6200U CPU 2.4 GHz
   - RAM 4 GB
2. Kebutuhan perangkat lunak:
   - Sistem operasi Windows 10
   - Bahasa pemrograman Python 2.7
   - *Library* Sastrawi

BAB 4 PERANCANGAN

Bab ini berisi perancangan algoritme, perancangan pengujian yang akan dilakukan, serta contoh perhitungan manualisasi dari metode yang digunakan.

4.1 Deskripsi Umum

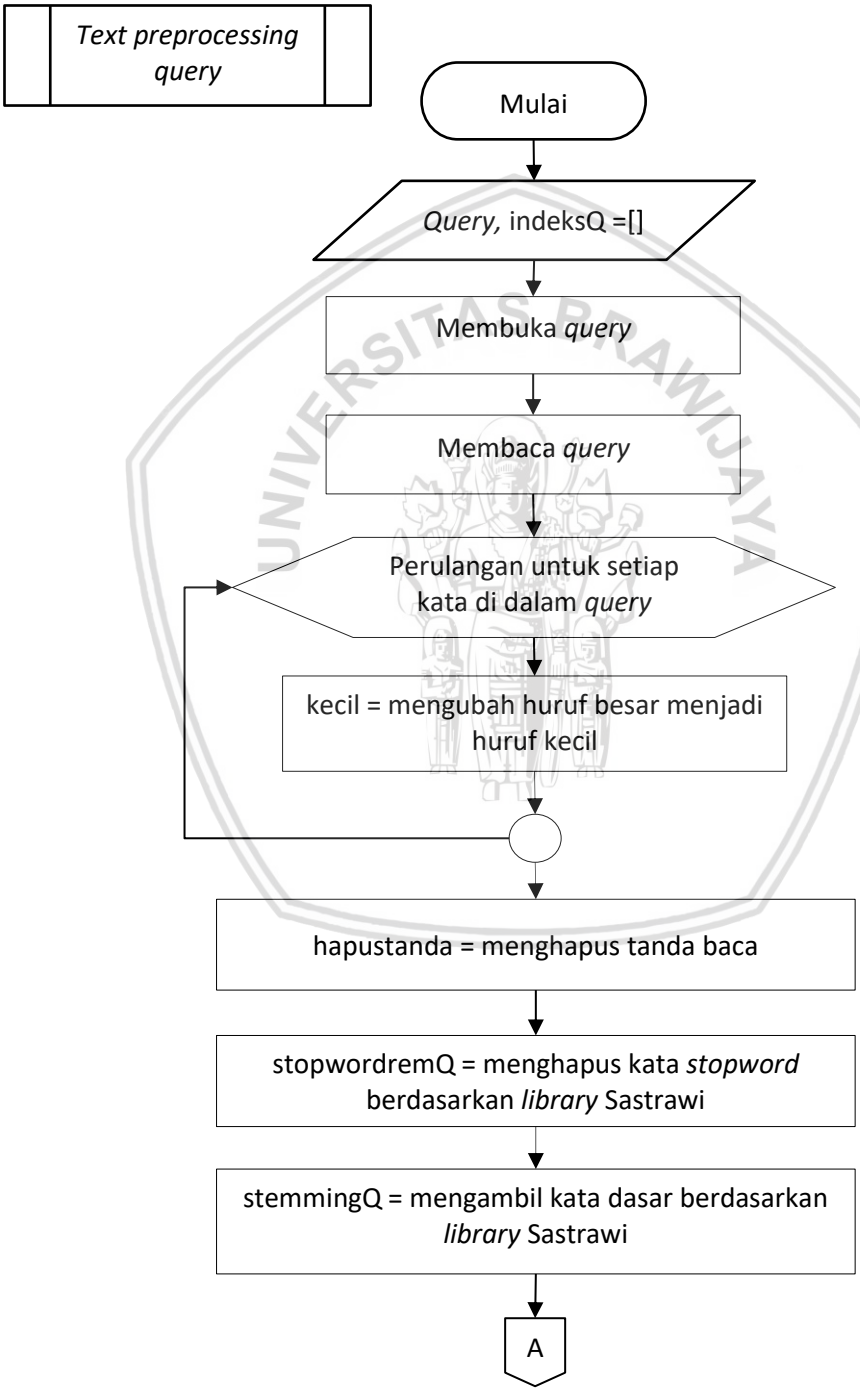
Deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 ini akan digunakan untuk menghitung nilai persentase plagiarisme suatu artikel berita. *Query* yang akan digunakan adalah sebuah artikel berita yang kemudian akan melalui proses *text preprocessing* untuk mengolah teks sebelum dilakukan perhitungan menggunakan metode BM25. Proses tersebut akan membutuhkan korpus berisi kumpulan beberapa artikel berita sebagai pembandingnya. Dari hasil perhitungan akan diketahui seberapa besar nilai bobot artikel berita yang kemudian digunakan sebagai dasar perhitungan persentase plagiarisme. Alur proses algoritme deteksi plagiarisme secara umum dapat dilihat pada Gambar 4.1.



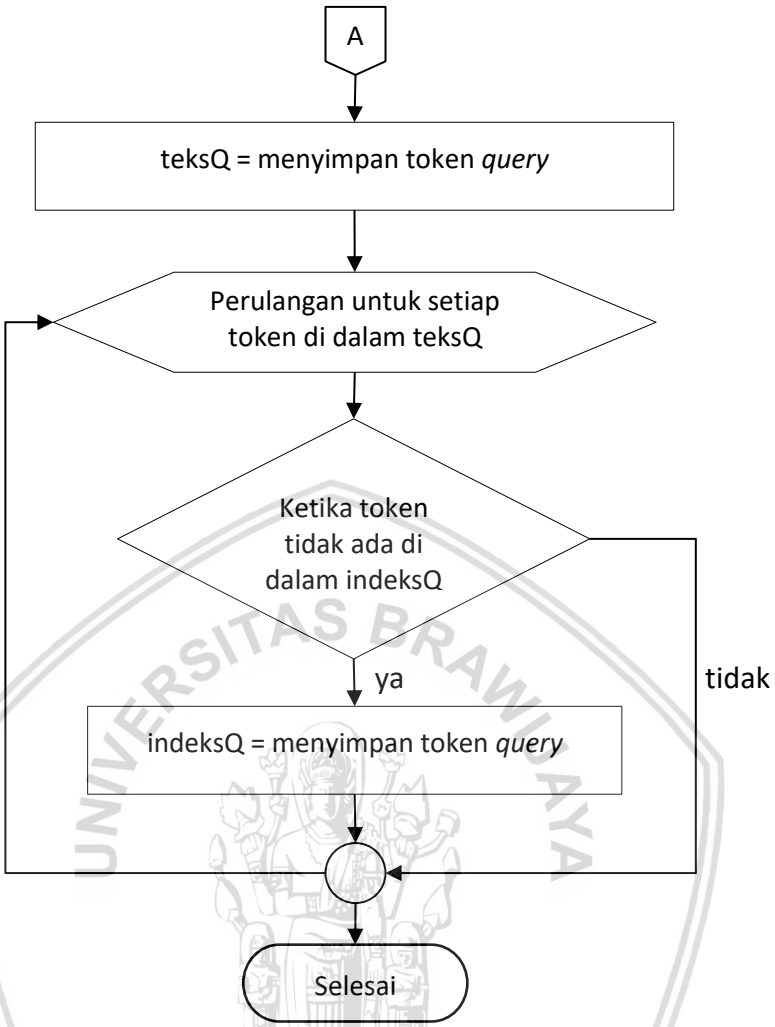
Gambar 4.1 Alur Proses Algoritme Deteksi Plagiarisme Pada Artikel Berita Berbahasa Indonesia Menggunakan BM25

4.1.1 *Text Preprocessing Query*

Text preprocessing terhadap *query* merupakan proses yang dilakukan untuk menghasilkan kata atau token dari *query* yang digunakan sebagai pedoman dalam mencari kemiripan dengan dokumen di dalam korpus. Hasil dari *text preprocessing query* akan diolah untuk proses pembobotan dengan metode BM25. Proses *text preprocessing query* dapat dilihat pada Gambar 4.2 dan Gambar 4.3.



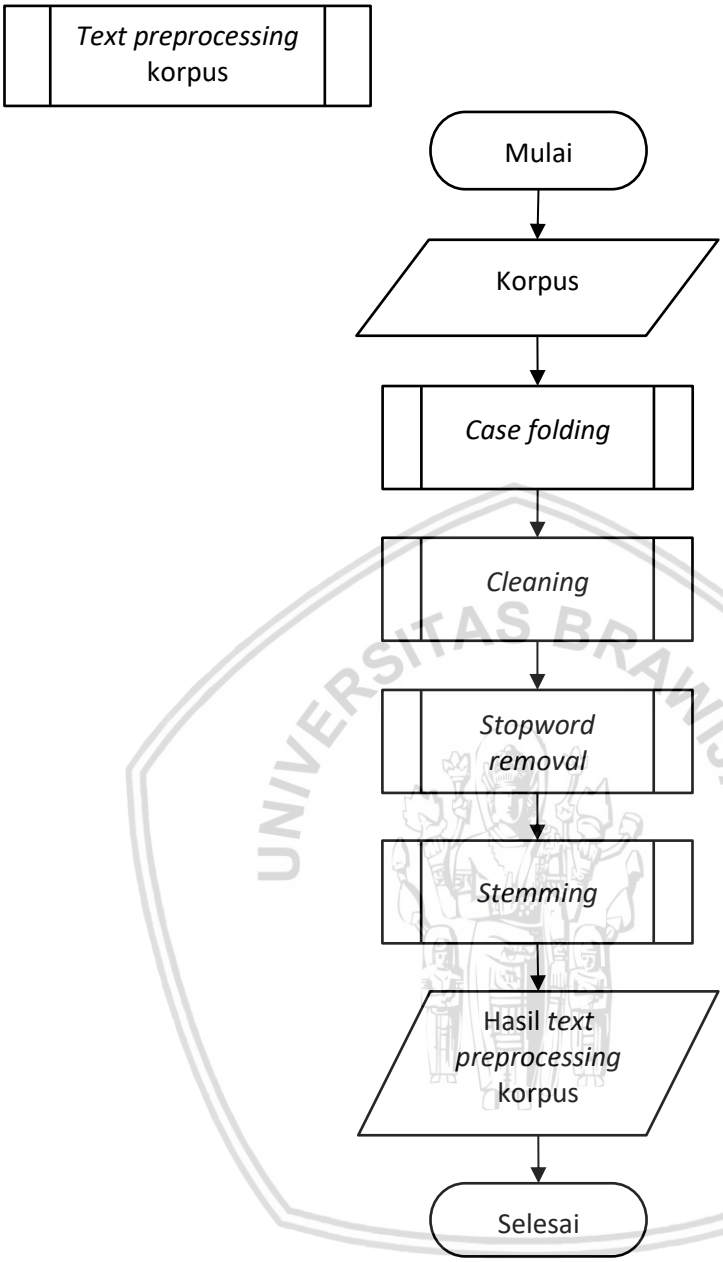
Gambar 4.2 Diagram Alir *Text Preprocessing Query*



Gambar 4.3 Diagram Alir *Text Preprocessing Query* (Lanjutan)

4.1.2 *Text Preprocessing* Korpus

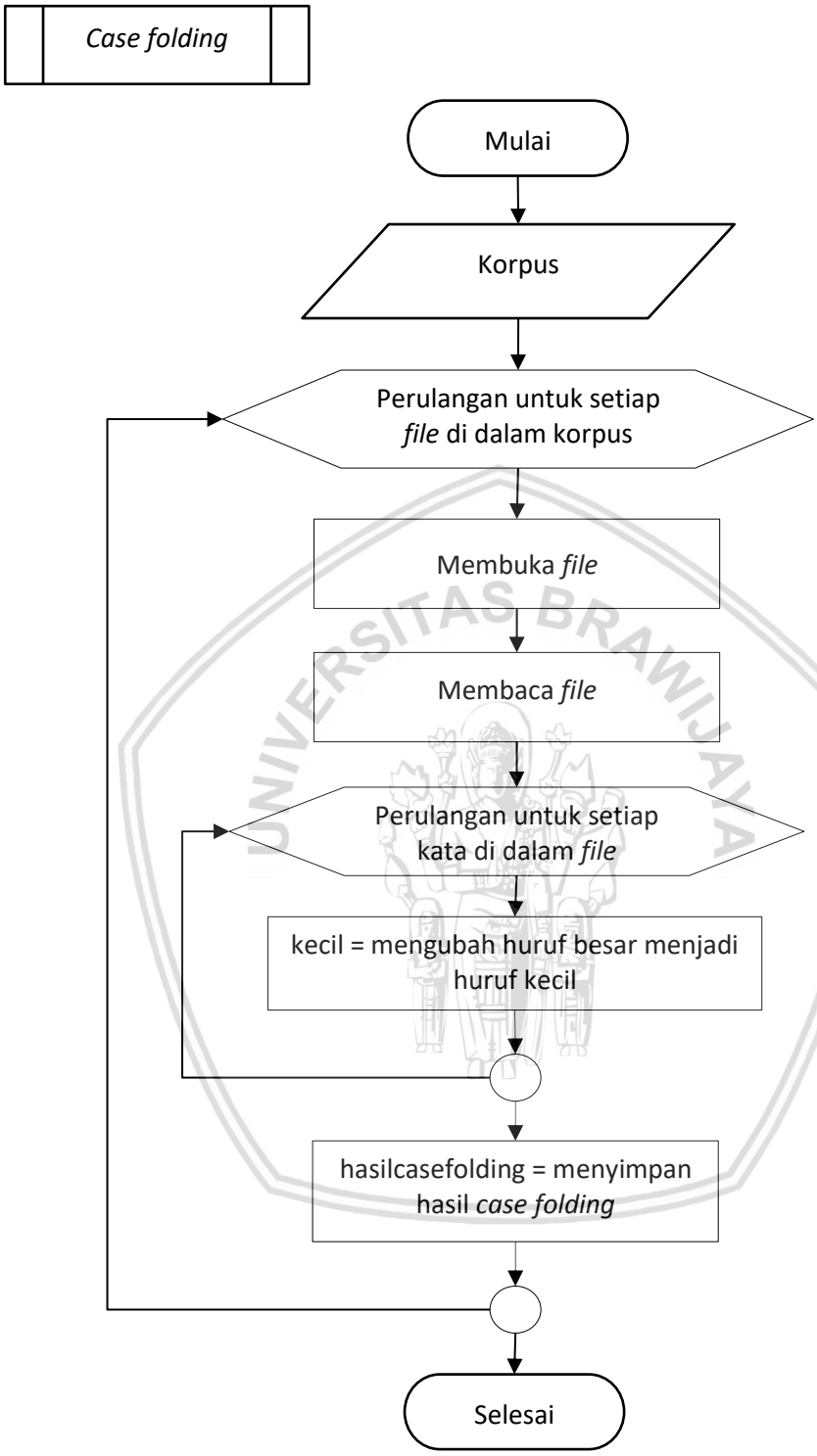
Text preprocessing terhadap korpus merupakan proses yang dilakukan untuk menghasilkan kata atau token dari kumpulan dokumen yang ada di dalam korpus untuk kemudian digunakan sebagai pembanding kemiripan dengan *query*. Beberapa tahap di dalam *text preprocessing* korpus yaitu *case folding, cleaning*, *stopword removal,* dan *stemming*. Setelah melalui proses *text preprocessing* maka akan didapatkan keluaran berupa daftar kata hasil *text preprocessing* yang akan diolah untuk proses pembobotan dengan metode BM25. Proses *text preprocessing* untuk korpus dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir *Text Preprocessing* Korpus

4.1.2.1 *Case Folding*

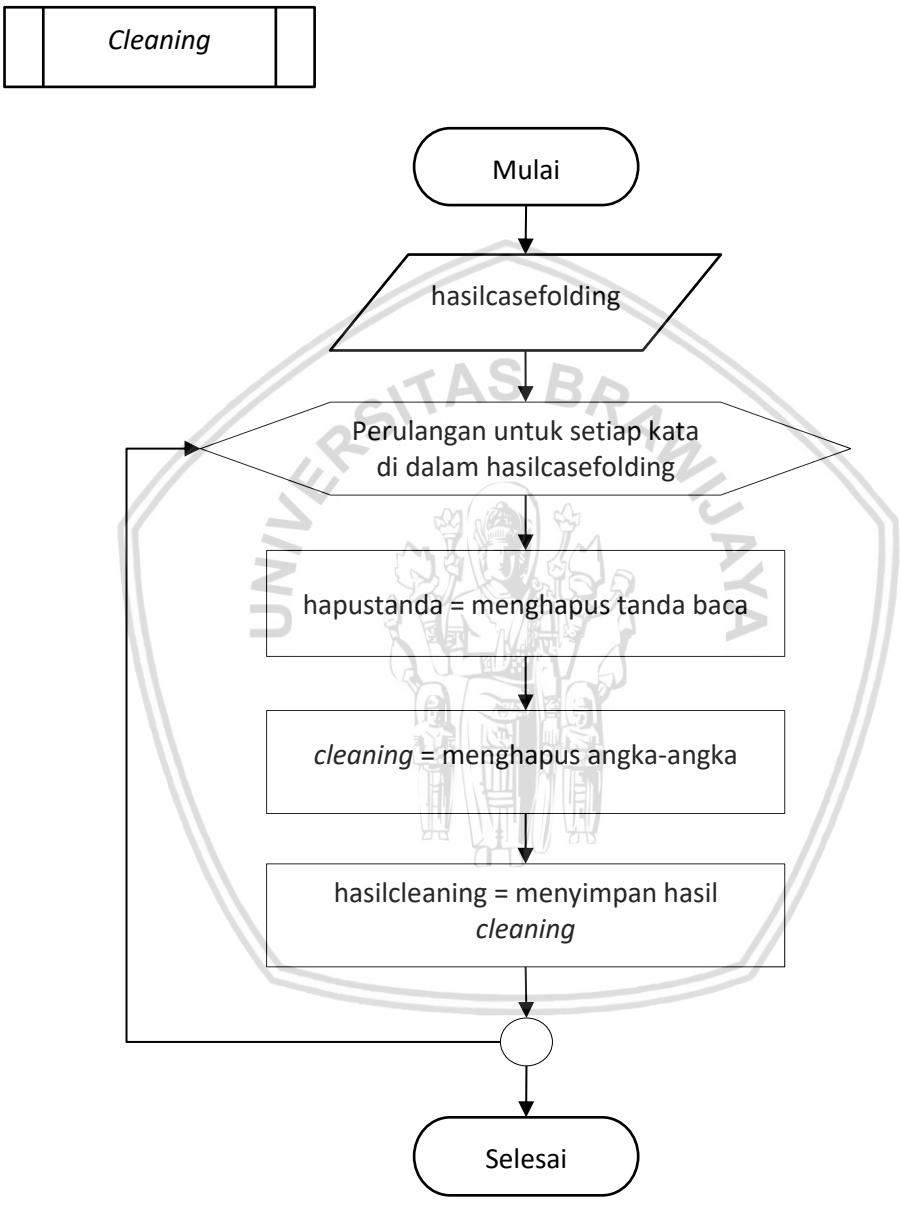
Masukan yang digunakan dalam proses *case folding* berupa artikel berita asli yang dikumpulkan menjadi sebuah korpus. Masing-masing artikel akan dibaca perbaris dan setiap hurufnya akan diubah menjadi huruf kecil. Kemudian hasil dari *case folding* akan disimpan ke dalam *list hasilcasefolding*. Alur dari proses *case folding* dapat dilihat pada Gambar 4.5.



Gambar 4.5 Diagram Alir *Case Folding*

4.1.2.2 *Cleaning*

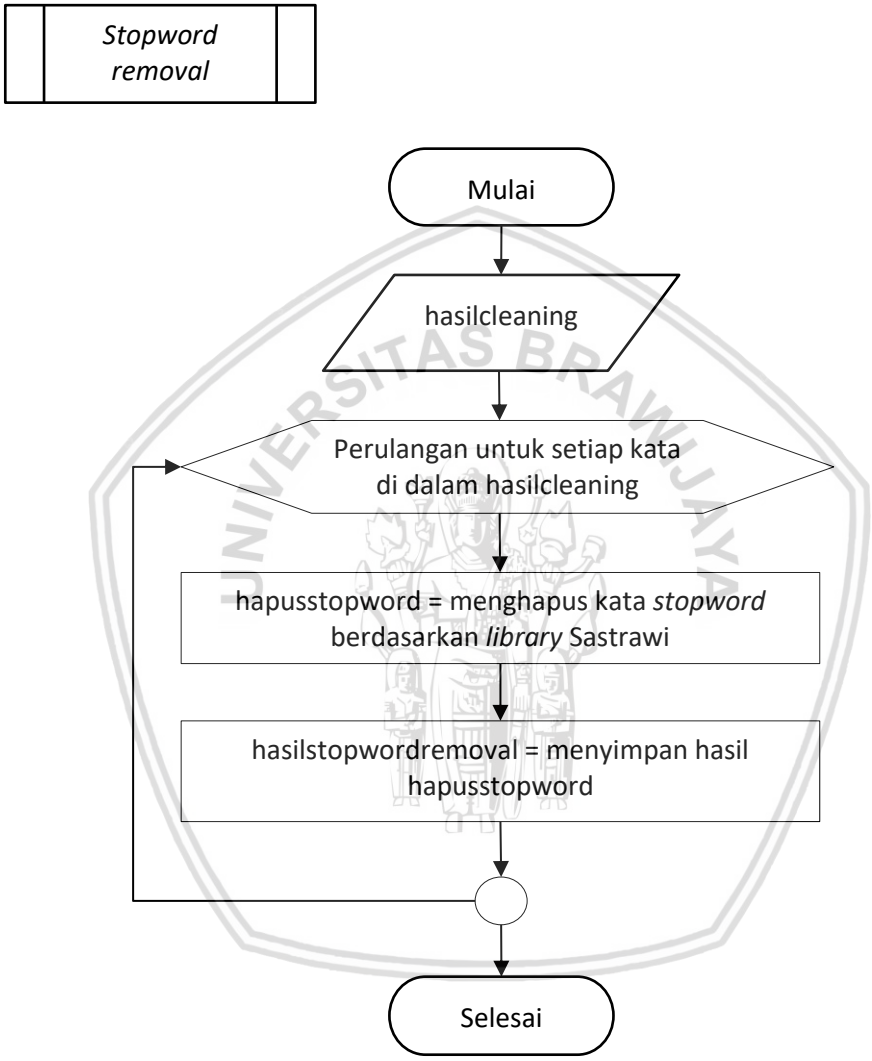
Masukan yang digunakan dalam proses *cleaning* adalah hasil dari proses *case folding,* kemudian cek tanda baca dan angka yang ada lalu menghapusnya. Hasil dari *cleaning* akan disimpan ke dalam *list hasilcleaning*. Alur dari proses *cleaning* dapat dilihat pada Gambar 4.6.



Gambar 4.6 Diagram Alir *Cleaning*

4.1.2.3 *Stopword Removal*

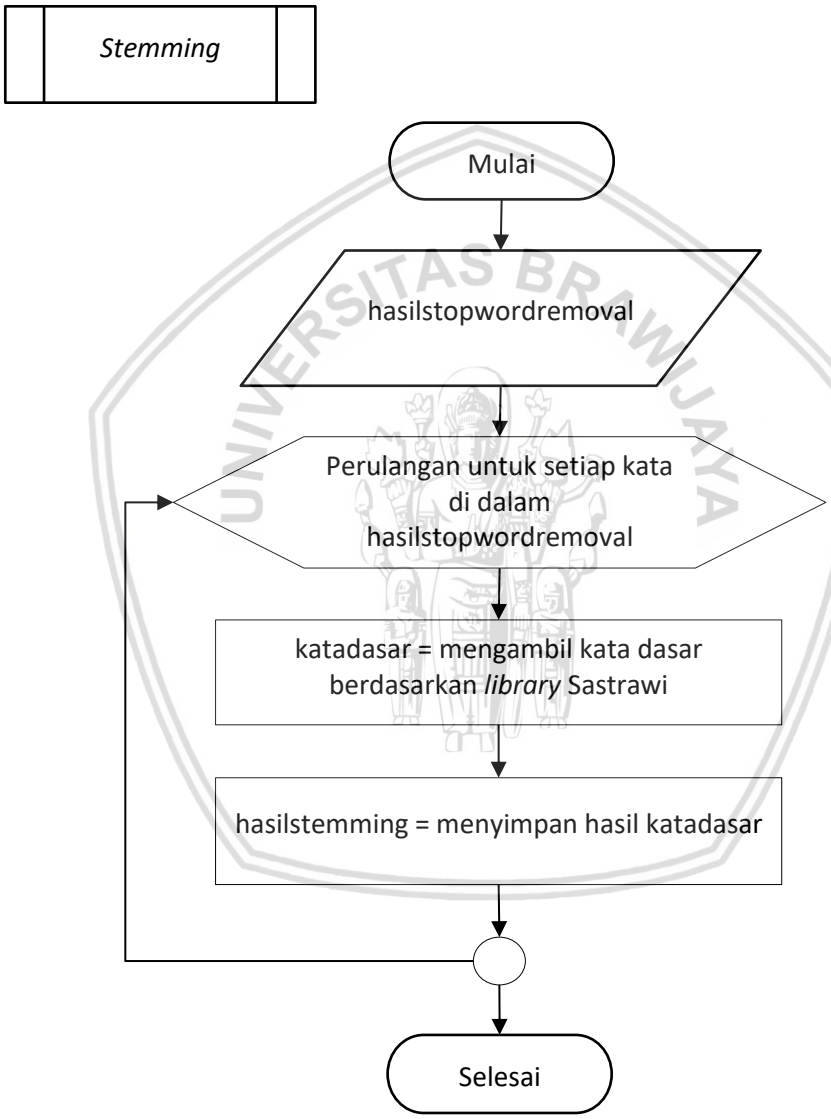
Masukan yang digunakan dalam proses *stopword removal* adalah hasil dari proses *cleaning.* Proses *stopword removal* akan menggunakan *library* Sastrawi untuk mendapatkan daftar kata *stopword* dan kemudian menghapusnya. Hasil dari *stopword removal* akan disimpan ke dalam *list hasilstopwordremoval*. Alur dari proses *stopword removal* dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir *Stopword Removal*

4.1.2.4 *Stemming*

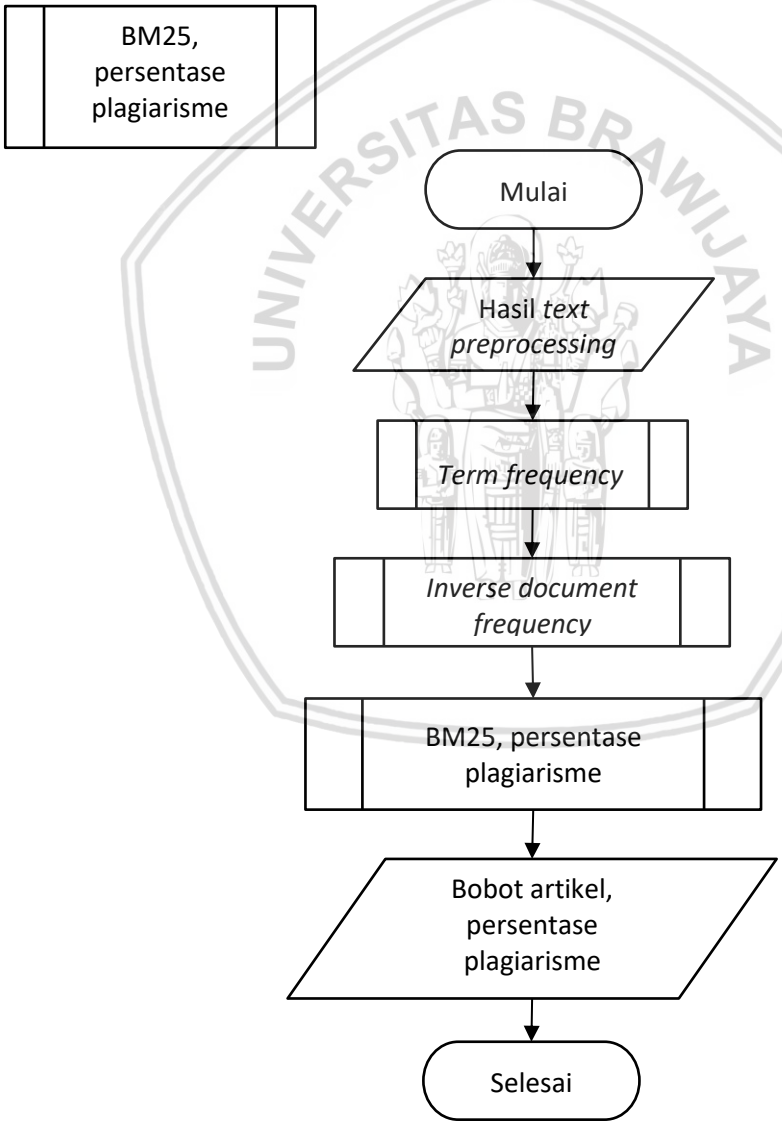
Masukan yang digunakan dalam proses *stemming* adalah hasil dari proses *stopword removal.* Proses *stemming* akan menggunakan *library* Sastrawi untuk mendapatkan daftar kata dasar beserta imbuhannya. Pencarian dilakukan dengan mencari kata yang terindikasi memiliki imbuhan. Setelah menemukan kata berimbuhan maka imbuhan dalam kata tersebut dihapus dan akan dikembalikan menjadi bentuk kata dasar. Hasil dari *stemming* akan disimpan ke dalam *list hasilstemming*. Alur dari proses *stemming* dapat dilihat pada Gambar 4.8.



Gambar 4.8 Diagram Alir *Stemming*

4.1.3 BM25 dan Persentase Plagiarisme

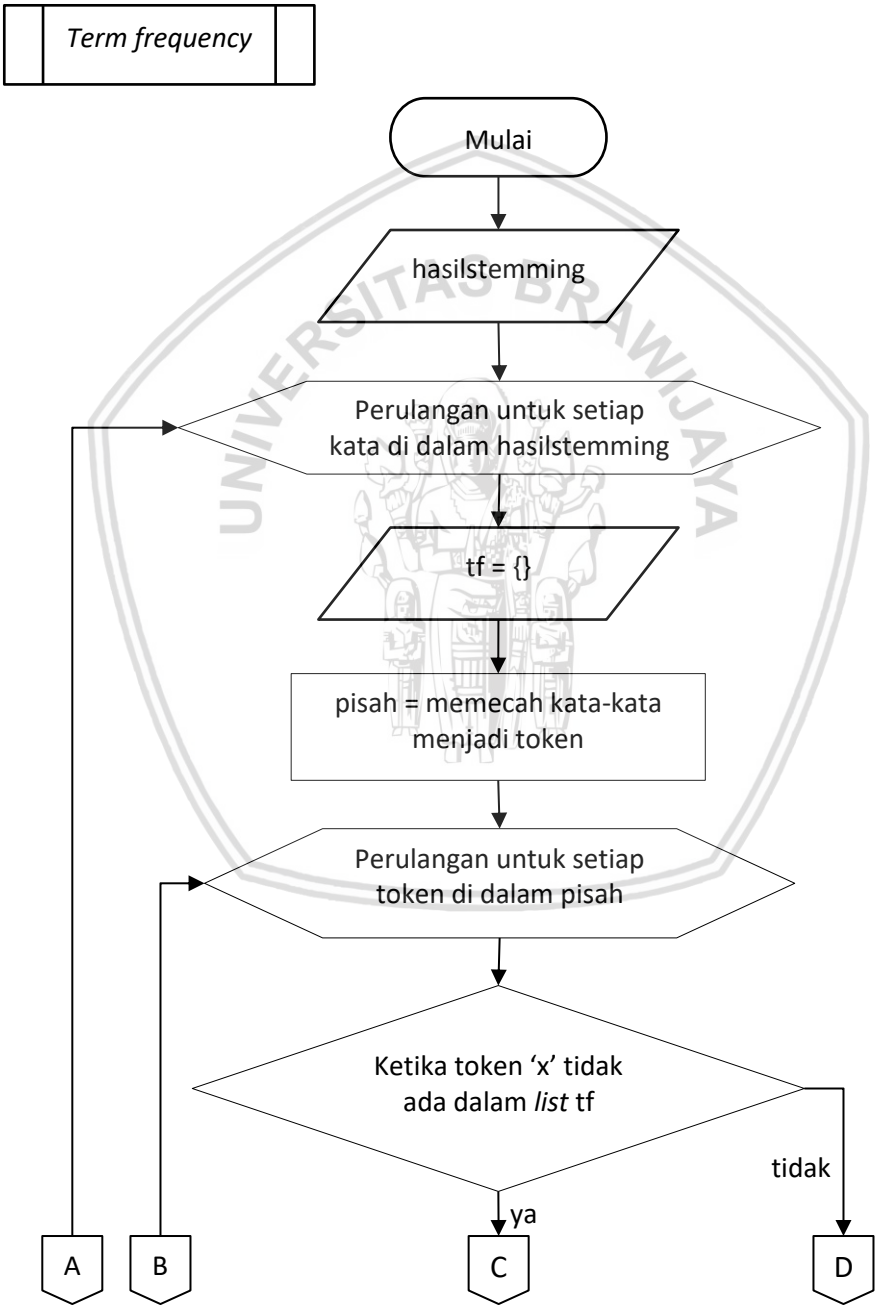
Dalam menghitung bobot dokumen menggunakan metode BM25, hal pertama yang harus dilakukan adalah menghitung frekuensi kata atau *Term Frequency* (TF), kemudian *Inverse Document Frequency* (IDF) dan terakhir yaitu menghitung bobot per kata dengan metode BM25 sesuai dengan Persamaan 2.2. Masukan yang digunakan berasal dari hasil proses *text preprocessing* yang terakhir yaitu *stemming.* Setelah semua proses dalam tahap pembobotan kata selesai, maka akan didapatkan hasil berupa nilai bobot kemiripan antar artikel *query* dengan artikel yang ada di dalam korpus. Dari nilai kemiripan tersebut akan dihitung lagi untuk menemukan nilai persentase plagiarismenya. Proses dalam perhitungan bobot kemiripan dokumen menggunakan metode BM25 dan perhitungan plagiarismenya dapat dilihat pada Gambar 4.9.



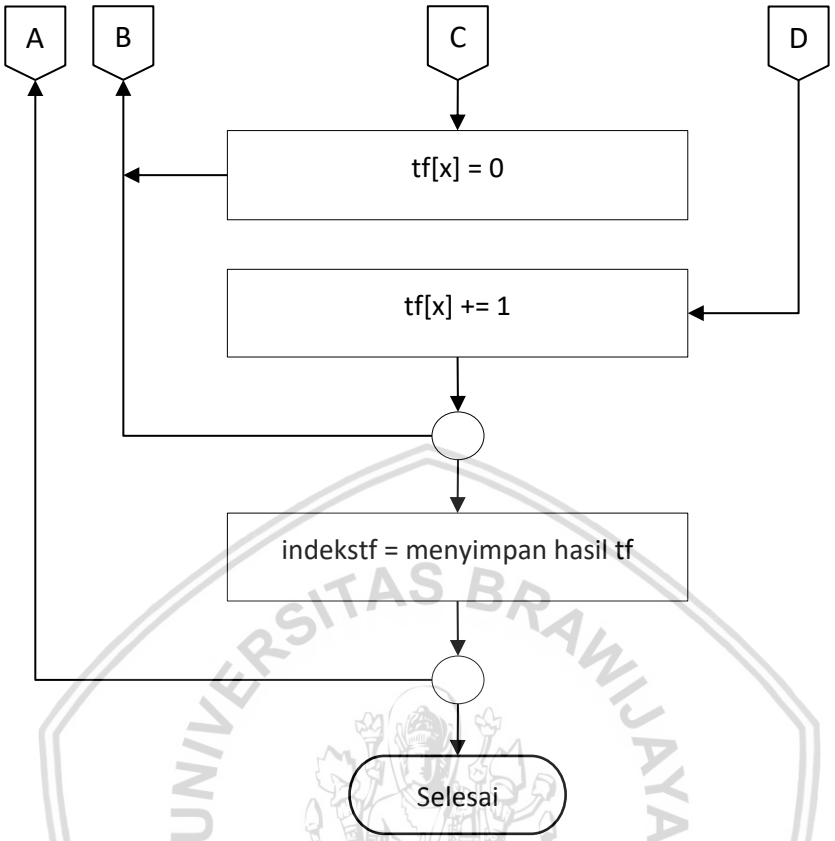
Gambar 4.9 Diagram Alir Perhitungan BM25 dan Persentase Plagiarisme

4.1.3.1 *Term Frequency*

Masukan yang digunakan dalam menghitung *Term Frequency* (TF) adalah hasil dari proses *text preprocessing* yang terakhir yaitu *stemming*. Setiap kata akan dicek kemunculannya di tiap-tiap artikel di dalam korpus. Hasil TF akan disimpan sebagai *value* bersama dengan kata atau *term*-nya yang berfungsi sebagai *key* di dalam *dict* yang bernama *tf.* Kemudian hasil dari TF akan disimpan ke dalam *list indekstf*. Alur dari proses *term frequency* dapat dilihat pada Gambar 4.10 dan Gambar 4.11.



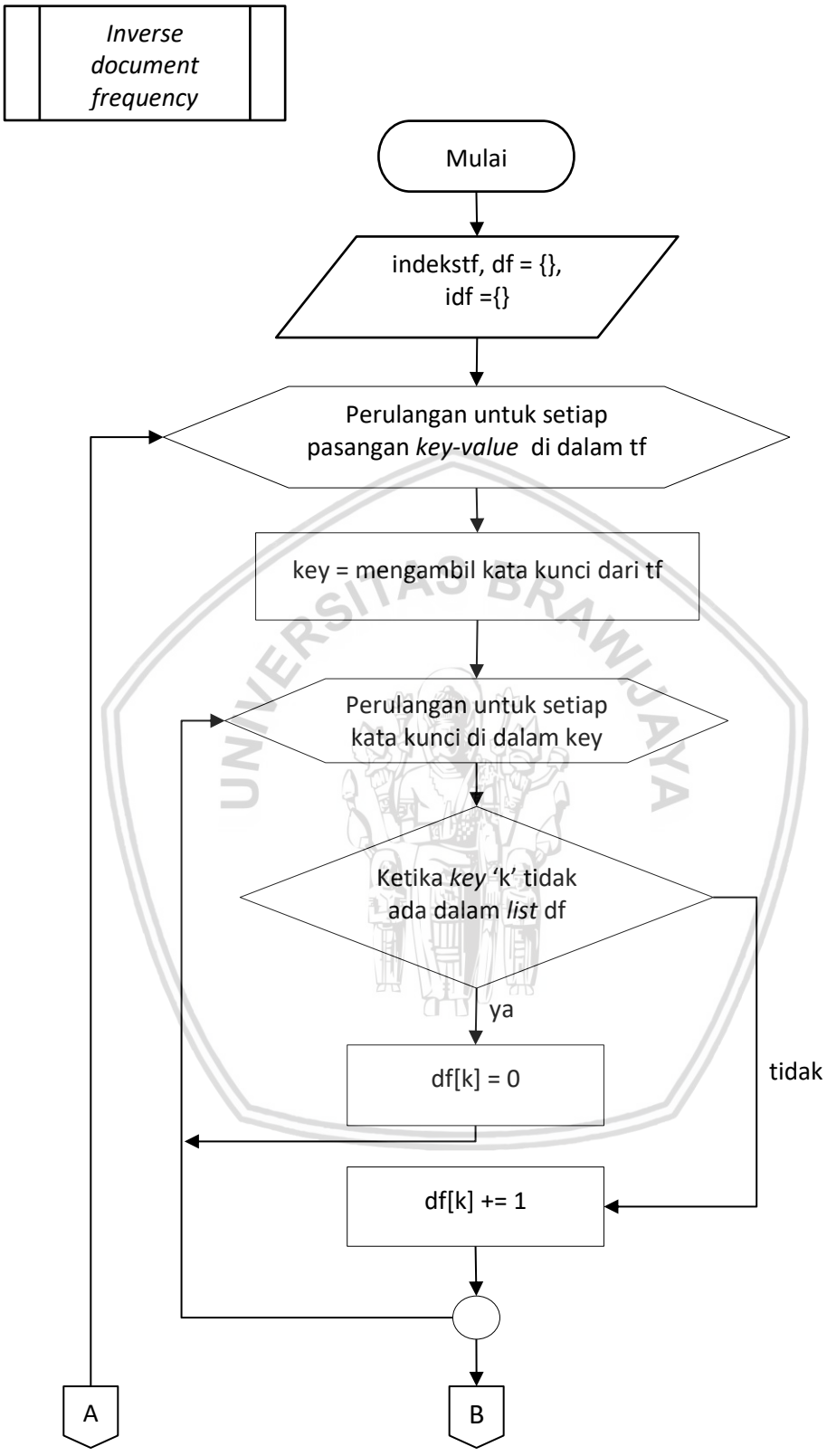
Gambar 4.10 Diagram Alir *Term Frequency*



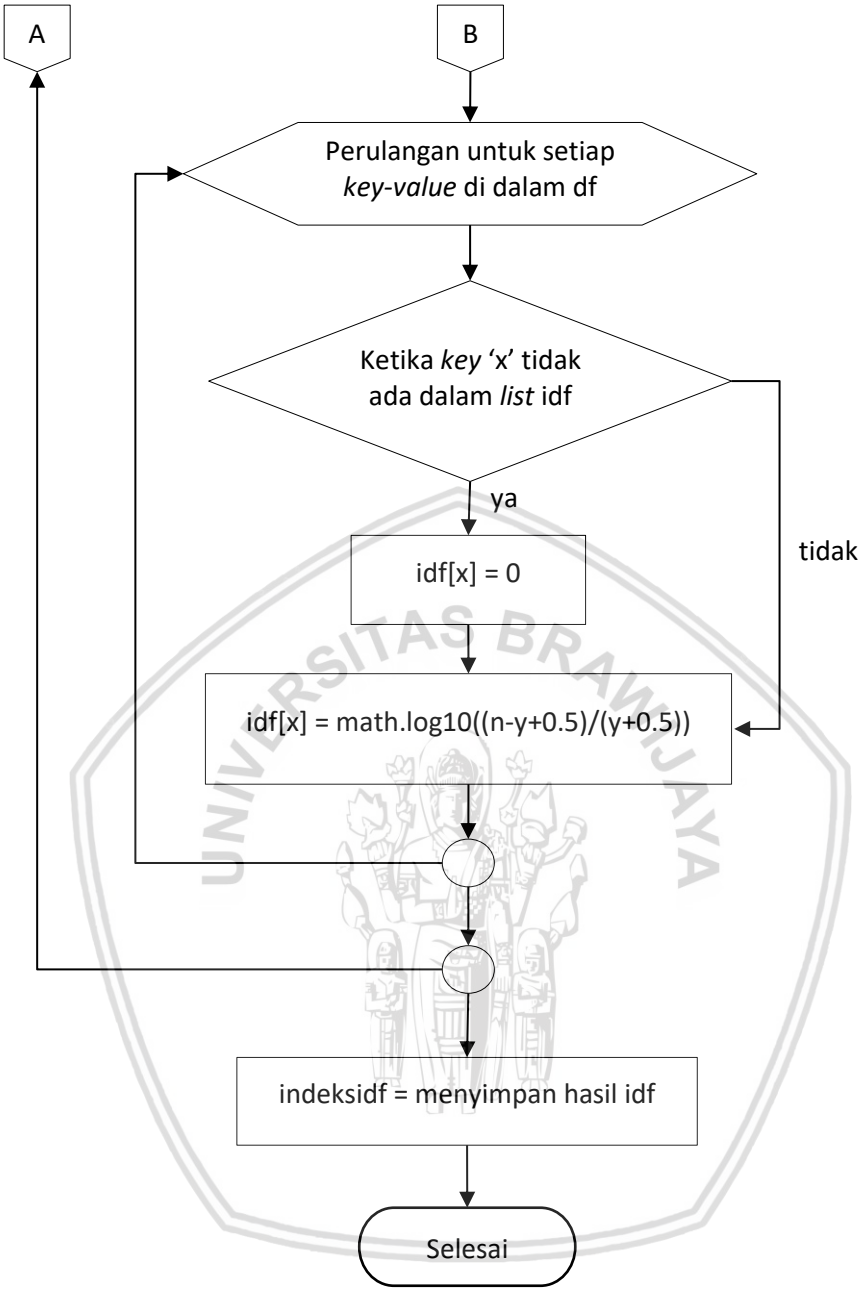
Gambar 4.11 Diagram Alir *Term Frequency* (Lanjutan)

4.1.3.2 *Inverse Document Frequency*

Perhitungan *Inverse Document Frequency* (IDF) membutuhkan nilai *Document Frequency* (DF) yang merupakan nilai dari jumlah dokumen di dalam korpus yang mengandung suatu *term* yang sama. Masukan yang digunakan adalah jumlah dokumen di dalam korpus dan hasil dari *term frequency* yang berupa *list indekstf*. Nilai di dalam *indekstf* disimpan dalam tipe data *dict,* sehingga untuk mendapatkan *term* harus menggunakan nilai *key*. Kemudian hasil DF akan digunakan untuk perhitungan IDF dan akan disimpan sebagai *value* bersama dengan kata atau *term*-nya yang berfungsi sebagai *key* di dalam *dict* bernama *idf*, kemudian hasil tersebut akan disimpan lagi ke dalam *list indeksidf*. Alur dari proses *inverse document frequency* dapat dilihat pada Gambar 4.12 dan Gambar 4.13.



Gambar 4.12 Diagram Alir *Inverse Document Frequency*

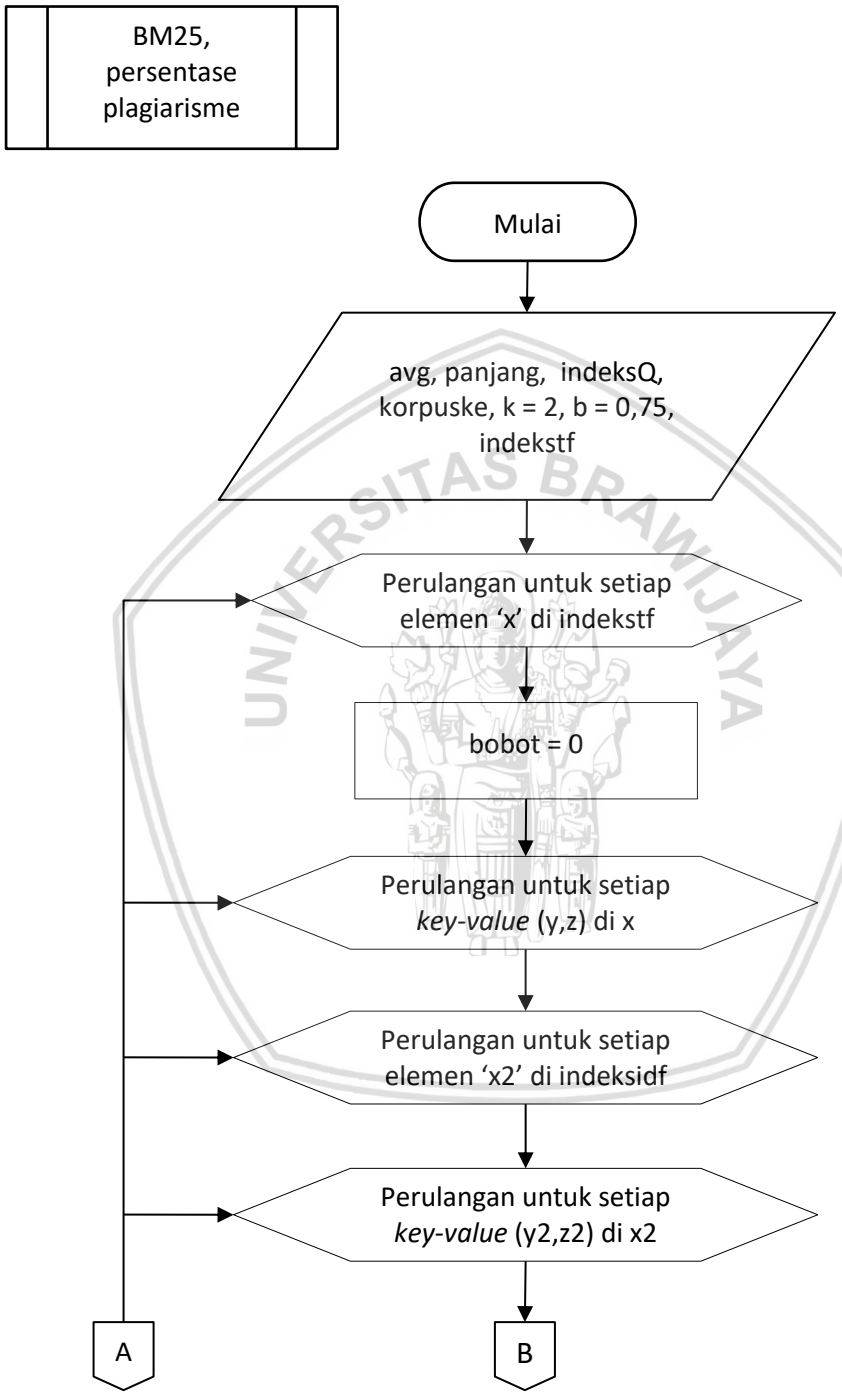


Gambar 4.13 Diagram Alir *Inverse Document Frequency* (Lanjutan)

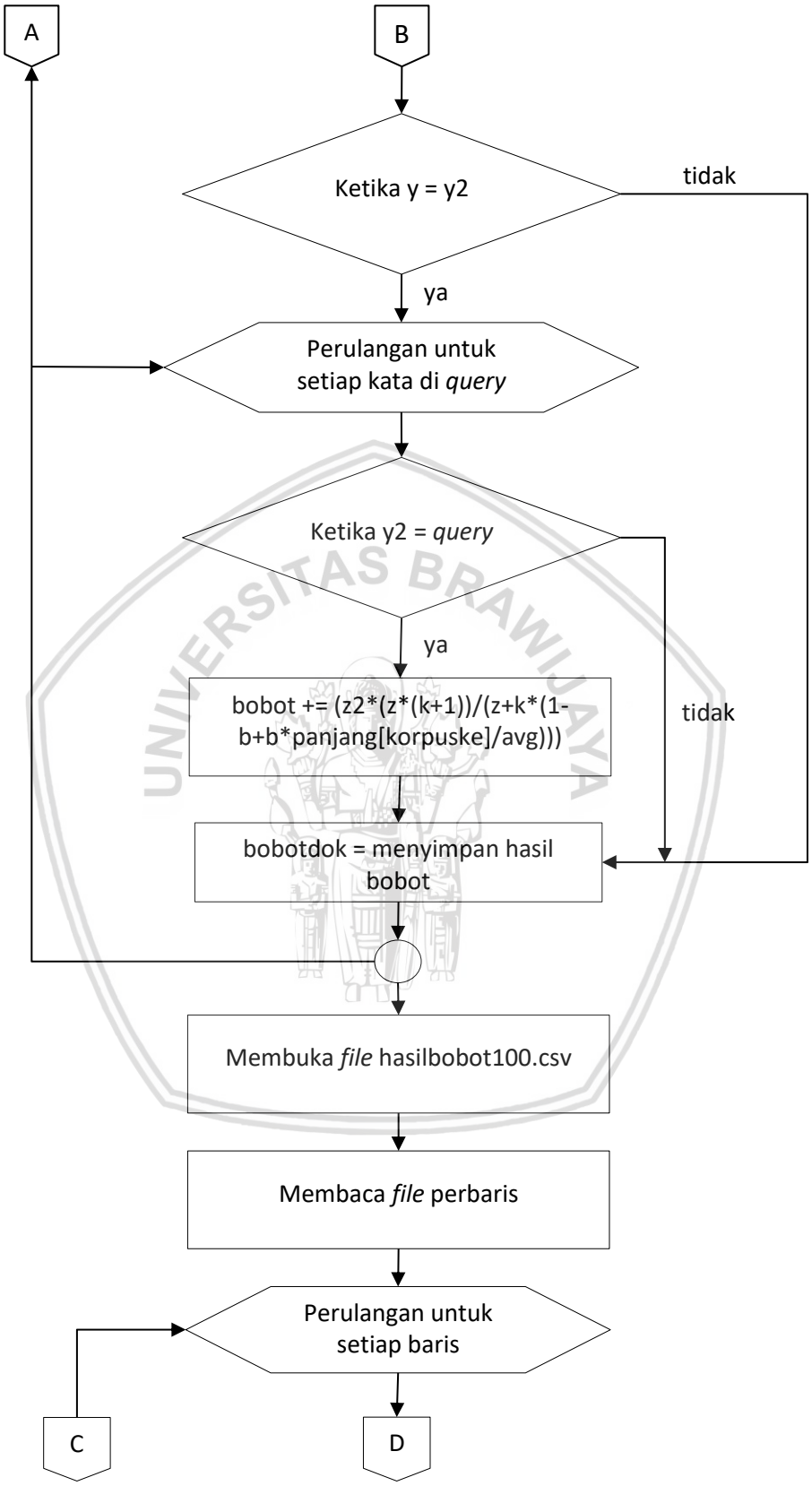
4.1.3.3 BM25 dan Persentase Plagiarisme

Setelah mendapatkan nilai *term frequency* dan *inverse document frequency,* langkah selanjutnya adalah menghitung nilai bobot kemiripan setiap dokumen menggunakan rumus BM25 pada Persamaan 2.2. Masukan yang digunakan adalah semua parameter pada BM25 dan keluaran yang akan dihasilkan yaitu nilai bobot kemiripan dari masing-masing artikel serta persentase plagiarismenya. Dalam menghitung persentase plagiarisme dibutuhkan nilai bobot artikel ketika 100%, maka dari itu setiap nilai bobot artikelnya disimpan dalam *file* berformat *.csv* untuk memudahkan perhitungan selanjutnya.

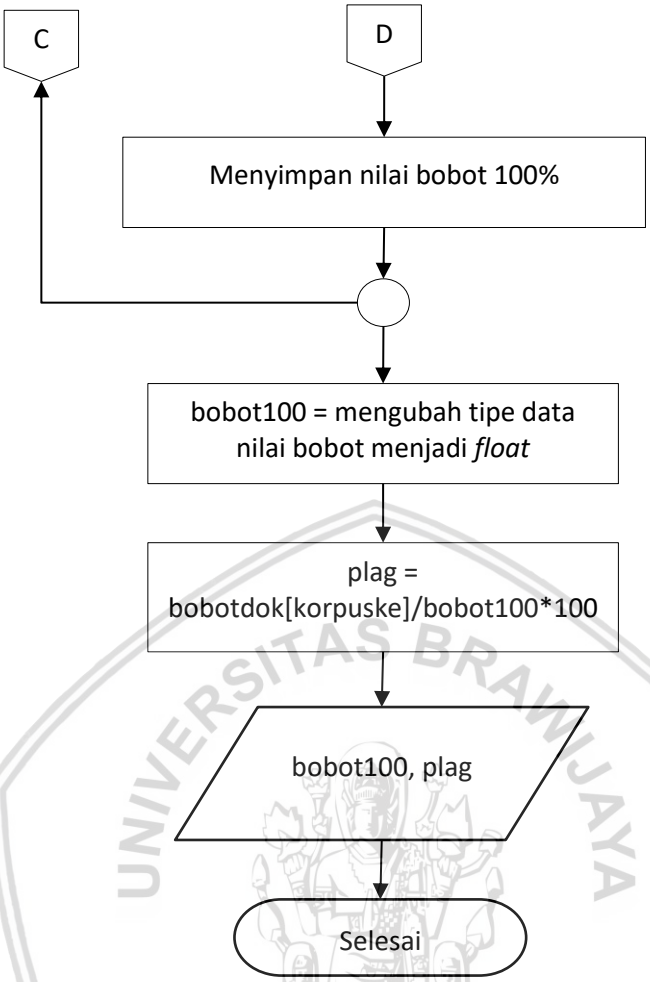
Proses perhitungan nilai bobot kemiripan artikel dengan metode BM25 sekaligus perhitungan persentase plagiarismenya dapat dilihat pada Gambar 4.14, Gambar 4.15 dan Gambar 4.16.



Gambar 4.14 Diagram Alir BM25 dan Persentase Plagiarisme



Gambar 4.15 Diagram Alir Perhitungan BM25 dan Persentase Plagiarisme (Lanjutan)



Gambar 4.16 Diagram Alir Perhitungan BM25 dan Persentase Plagiarisme (Lanjutan)

4.2 Manualisasi

Manualisasi adalah sebuah proses perhitungan yang dilakukan secara manual dalam mengolah dokumen pada proses *text preprocessing,* perhitungan bobot kemiripan dokumen menggunakan metode BM25 dan perhitungan persentase plagiarismenya. Proses ini bertujuan sebagai contoh sekaligus gambaran dari proses perhitungan pada program deteksi plagiarisme yang akan dibuat. Dalam manualisasi ini digunakan tiga dokumen atau artikel yang dikumpulkan ke dalam korpus. Korpus yang digunakan dapat dilihat dapat dilihat pada Tabel 4.1.

Tabel 4.1 Korpus

Dok	Kalimat
D1	Bonus bagi para atlet Indonesia peraih medali di Asian Games 2018 tidak sekadar uang maupun rumah, tapi berupa jaminan menjadi pegawai negeri sipil (PNS). Tak cuma itu, para atlet berprestasi juga memiliki kesempatan berkarier masuk TNI dan Polri tanpa tes.

Dok	Kalimat
D2	Pasangan bakal calon presiden dan bakal calon wakil presiden, Prabowo Subianto dan Sandiaga Uno akan sowan ke sejumlah organisasi massa di Indonesia. Pimpinan Pusat Muhammadiyah menjadi ormas pertama yang dikunjungi.
D3	Usai mengumumkan cawapresnya, Jokowi banjir dukungan di Banten. Terutama, terpilihnya KH.Ma'ruf Amien sebagai pendampingnya.

4.2.1 *Text Preprocessing*

Text preprocessing digunakan untuk mengolah teks sebelum dilakukan pemrosesan yang lebih lanjut atau perhitungan pembobotan kata. Beberapa proses dalam *text preprocessing* yaitu *case folding, cleaning*, *stopword removal, stemming,* dan tokenisasi.

4.2.1.1 *Case Folding*

Proses *case folding* merupakan proses pertama dalam *text preprocessing*. Berdasarkan proses *case folding* pada Gambar 4.3, maka diperoleh hasil *case folding* dari korpus yang dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil *Case Folding*

Dok	Kalimat
D1	bonus bagi para atlet indonesia peraih medali di asian games 2018 tidak sekadar uang maupun rumah, tapi berupa jaminan menjadi pegawai negeri sipil (pns). tak cuma itu, para atlet berprestasi juga memiliki kesempatan berkarier masuk tni dan polri tanpa tes.
D2	pasangan bakal calon presiden dan bakal calon wakil presiden, prabowo subianto dan sandiaga uno akan sowan ke sejumlah organisasi massa di indonesia. pimpinan pusat muhammadiyah menjadi ormas pertama yang dikunjungi.
D3	usai mengumumkan cawapresnya, jokowi banjir dukungan di banten. terutama, terpilihnya kh.ma'ruf amien sebagai pendampingnya.

4.2.1.2 *Cleaning*

Proses *cleaning* merupakan proses kedua dalam *text preprocessing*. Berdasarkan proses *cleaning* pada Gambar 4.4, maka diperoleh hasil *cleaning* dari korpus pada Tabel 4.3.

Tabel 4.3 Hasil *Cleaning*

Dok	Kalimat
D1	bonus bagi para atlet indonesia peraih medali di asian games tidak sekadar uang maupun rumah tapi berupa jaminan menjadi pegawai negeri sipil pns tak cuma itu para atlet berprestasi juga memiliki kesempatan berkarier masuk tni dan polri tanpa tes
D2	pasangan bakal calon presiden dan bakal calon wakil presiden prabowo subianto dan sandiaga uno akan sowan ke sejumlah organisasi massa di

Dok	Kalimat
	indonesia pimpinan pusat muhammadiyah menjadi ormas pertama yang dikunjungi
D3	usai mengumumkan cawapresnya jokowi banjir dukungan di banten terutama terpilihnya khmaruf amien sebagai pendampingnya

4.2.1.3 *Stopword Removal*

Proses *stopword removal* merupakan proses ketiga dalam *text preprocessing*. Berdasarkan proses *stopword removal* pada Gambar 4.5, maka diperoleh hasil *stopword removal* dari korpus pada Tabel 4.4.

Tabel 4.4 Hasil *Stopword Removal*

Dok	Kalimat
D1	bonus atlet indonesia peraih medali asian games uang rumah jaminan pegawai negeri sipil pns atlet berprestasi memiliki kesempatan berkarier tni polri tes
D2	pasangan calon presiden calon wakil presiden prabowo subianto sandiaga uno sowan organisasi massa indonesia pimpinan pusat muhammadiyah ormas dikunjungi
D3	mengumumkan cawapresnya jokowi banjir dukungan banten terpilihnya khmaruf amien pendampingnya

4.2.1.4 *Stemming*

Proses *stemming* merupakan proses keempat dalam *preprocessing text*. Berdasarkan proses *stemming* pada Gambar 4.6, maka diperoleh hasil *stemming* dari korpus pada Tabel 4.5.

Tabel 4.5 Hasil *Stemming*

Dok	Kalimat
D1	bonus atlet indonesia raih medali asian games uang rumah jamin pegawai negeri sipil pns atlet prestasi milik sempat karier tni polri tes
D2	pasang calon presiden calon wakil presiden prabowo subianto sandiaga uno sowan organisasi massa indonesia pimpin pusat muhammadiyah ormas kunjung
D3	umum cawapresnya jokowi banjir dukung banten pilih khmaruf amien damping

4.2.1.5 Tokenisasi

Proses tokenisasi merupakan proses terakhir dalam *text preprocessing*. Berdasarkan pada perancangan yang telah dilakukan, proses tokenisasi dilakukan sebelum perhitungan *term frequency.* Hasil tokenisasi dari korpus dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil Tokenisasi

D1	D2	D3
asian	calon	amien

D1	D2	D3
atlet	indonesia	banjir
bonus	kunjung	banten
games	massa	cawapresnya
indonesia	muhammadiyah	damping
jamin	organisasi	dukung
karier	ormas	jokowi
medali	pasang	khmaruf
milik	pimpin	pilih
negeri	prabowo	umum
pegawai	presiden	
pns	pusat	
polri	sandiaga	
prestasi	sowan	
raih	subianto	
rumah	uno	
sempat	wakil	
sipil		
tes		
tni		
uang		

4.2.2 Metode BM25

Setelah melakukan *text preprocessing* dan mendapatkan hasil berupa *term* atau token, maka langkah selanjutnya adalah menghitung bobot atau kemiripan tiap dokumen di dalam korpus terhadap *query*. Metode BM25 meliputi perhitungan *term frequency, inverse document frequency,* dan pembobotan kata menggunakan BM25.

4.2.2.1 *Term Frequency*

Perhitungan *term frequency* merupakan proses pertama dalam perhitungan bobot menggunakan BM25. Berdasarkan proses *term frequency* pada Gambar 4.8, maka diperoleh hasil *term frequency* dari korpus pada Tabel 4.7.

Tabel 4.7 Hasil *Term Frequency*

Term	D1	D2	D3
asian	1	0	0
atlet	2	0	0
bonus	1	0	0
games	1	0	0
indonesia	1	1	0
jamin	1	0	0
karier	1	0	0
medali	1	0	0

Term	D1	D2	D3
milik	1	0	0
negeri	1	0	0
pegawai	1	0	0
pns	1	0	0
polri	1	0	0
prestasi	1	0	0
raih	1	0	0
rumah	1	0	0
sempat	1	0	0
sipil	1	0	0
tes	1	0	0
tni	1	0	0
uang	1	0	0
calon	0	2	0
kunjung	0	1	0
massa	0	1	0
muhammadiyah	0	1	0
organisasi	0	1	0
ormas	0	1	0
pasang	0	1	0
pimpin	0	1	0
prabowo	0	1	0
presiden	0	2	0
pusat	0	1	0
sandiaga	0	1	0
sowan	0	1	0
subianto	0	1	0
uno	0	1	0
wakil	0	1	0
amien	0	0	1
banjir	0	0	1
banten	0	0	1
cawapresnya	0	0	1
damping	0	0	1
dukung	0	0	1
jokowi	0	0	1
khmaruf	0	0	1
pilih	0	0	1
umum	0	0	1

4.2.2.2 *Inverse Document Frequency*

Sebelum menghitung *inverse document frequency*, diperlukan nilai *document frequency* terlebih dahulu. Hasil *document frequency* dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil *Document Frequency*

Term	DF
asian	1
atlet	1
bonus	1
games	1
indonesia	2
jamin	1
karier	1
medali	1
milik	1
negeri	1
pegawai	1
pns	1
polri	1
prestasi	1
raih	1
rumah	1
sempat	1
sipil	1
tes	1
tni	1
uang	1
calon	1
kunjung	1
massa	1
muhammadiyah	1
organisasi	1
ormas	1
pasang	1
pimpin	1
prabowo	1
presiden	1
pusat	1
sandiaga	1
sowan	1
subianto	1
uno	1
wakil	1

Term	DF
amien	1
banjir	1
banten	1
cawapresnya	1
damping	1
dukung	1
jokowi	1
khmaruf	1
pilih	1
umum	1

Setelah mendapatkan hasil *document frequency,* langkah selanjutnya yaitu menghitung *inverse document frequency* menggunakan rumus IDF pada Persamaan 2.3.

- *Term* = asian

$$IDF_{asian} = log \frac{N - DF_{asian} + 0{,}5}{DF_{asian} + 0{,}5}$$

$$IDF_{asian} = log \frac{3 - 1 + 0{,}5}{1 + 0{,}5}$$

$$IDF_{asian} = log \frac{2{,}5}{1{,}5}$$

$$IDF_{asian} = 0{,}22184875$$

Hasil perhitungan *inverse document frequency* selengkapnya dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil *Inverse Document Frequency*

Term	IDF
asian	0,22184875
atlet	0,22184875
bonus	0,22184875
games	0,22184875
indonesia	-0,22184875
jamin	0,22184875
karier	0,22184875
medali	0,22184875
milik	0,22184875
negeri	0,22184875
pegawai	0,22184875
pns	0,22184875
polri	0,22184875
prestasi	0,22184875

Term	IDF
raih	0,22184875
rumah	0,22184875
sempat	0,22184875
sipil	0,22184875
tes	0,22184875
tni	0,22184875
uang	0,22184875
calon	0,22184875
kunjung	0,22184875
massa	0,22184875
muhammadiyah	0,22184875
organisasi	0,22184875
ormas	0,22184875
pasang	0,22184875
pimpin	0,22184875
prabowo	0,22184875
presiden	0,22184875
pusat	0,22184875
sandiaga	0,22184875
sowan	0,22184875
subianto	0,22184875
uno	0,22184875
wakil	0,22184875
amien	0,22184875
banjir	0,22184875
banten	0,22184875
cawapresnya	0,22184875
damping	0,22184875
dukung	0,22184875
jokowi	0,22184875
khmaruf	0,22184875
pilih	0,22184875
umum	0,22184875

4.2.2.3 Perhitungan BM25

Setelah mendapatkan nilai *term frequency* dan *inverse document frequency*, maka langkah selanjutnya yaitu menghitung bobot kemiripan dokumen menggunakan BM25 sesuai dengan Persamaan 2.2. Dalam penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 akan dilakukan beberapa kali perhitungan bobot, yaitu ketika artikel *query* 100% sama dengan artikel asli di dalam korpus, kemudian dipotong menjadi 75%, 50%, dan 25% dari panjang artikel yang sebenarnya. Terdapat beberapa parameter yang

dibutuhkan dalam proses perhitungan bobot menggunakan BM25, yaitu parameter *k, b*, panjang dokumen, dan rata-rata panjang dokumen di dalam korpus. Nilai parameter yang digunakan yaitu k = 2, dan b = 0,75, sedangkan untuk panjang dokumen dan rata-rata panjang dokumen dapat dilihat pada Tabel 4.10.

Tabel 4.10 Panjang dan Panjang Rata-Rata Dokumen

Dok	Panjang Dokumen
D1	40
D2	30
D3	14
Total	84
Rata-Rata	28

Pada manualisasi ini akan dijelaskan proses perhitungan BM25 terhadap *threshold* 25% atau *query* dengan panjang 25% dari artikel asli di dalam korpus. *Query* dengan *threshold* 25% dapat dilihat pada Tabel 4.11.

Tabel 4.11 *Query*

Kalimat
Bonus bagi para atlet Indonesia peraih medali di Asian Games.

Query akan diproses terlebih dahulu seperti proses *text preprocessing* pada korpus. Setelah mendapatkan *term* atau token, maka proses selanjutnya yaitu perhitungan *term frequency* dan *inverse document frequency.* Hasil dari perhitungan *term frequency* dan *inverse document frequency* dapat dilihat pada Tabel 4.12.

Tabel 4.12 Hasil *Term Frequency* dan *Inverse Document Frequency Query*

Term	Term Frequency			Document Frequency	Inverse Document Frequency
	D1	D2	D3		
asian	1	0	0	1	0,22184875
atlet	2	0	0	1	0,22184875
bonus	1	0	0	1	0,22184875
games	1	0	0	1	0,22184875
indonesia	1	1	0	2	-0,22184875
medali	1	0	0	1	0,22184875
raih	1	0	0	1	0,22184875

Proses selanjutnya yaitu menghitung nilai kemiripan dokumen sesuai dengan rumus BM25 pada Persamaan 2.2. Dalam perhitungan ini dokumen yang akan digunakan yaitu dokumen D1.

$$BM25_{d1} = IDF_{(asian)} \times \frac{TF_{(asian)} \times (k+1)}{TF_{(asian)} + k \times \left(1 - b + b \times \frac{|d_{d1}|}{L}\right)} + IDF_{(atlet)} \times \frac{TF_{(atlet)} \times (k+1)}{TF_{(atlet)} + k \times \left(1 - b + b \times \frac{|d_{d1}|}{L}\right)} + \cdots + IDF_{(raih)} \times \frac{TF_{(raih)} \times (k+1)}{TF_{(raih)} + k \times \left(1 - b + b \times \frac{|d_{d1}|}{L}\right)}$$

$$BM25_{d1} = 0{,}22184875 \times \frac{1 \times (2+1)}{1 + 2 \times \left(1 - 0{,}75 + 0{,}75 \times \frac{|40|}{28}\right)} + 0{,}22184875 \times \frac{2 \times (2+1)}{2 + 2 \times \left(1 - 0{,}75 + 0{,}75 \times \frac{|40|}{28}\right)} + \cdots + 0{,}22184875 \times \frac{1 \times (2+1)}{1 + 2 \times \left(1 - 0{,}75 + 0{,}75 \times \frac{|40|}{28}\right)}$$

$$BM25_{d1} = 1{,}017492727$$

4.2.2.4 Perhitungan Persentase Plagiarisme

Setelah mendapatkan bobot kemiripan dokumen menggunakan BM25, maka proses selanjutnya yaitu menghitung persentase plagiarisme dokumen menggunakan Persamaan 2.1.

$$Plagiarisme_{d1} = \frac{BM25_{d1}}{Bobot\ 100\%\ BM25_{d1}} \times 100\%$$

$$Plagiarisme_{d1} = \frac{1{,}017492727}{3{,}575278311} \times 100\%$$

$$Plagiarisme_{d1} = 28{,}4591195\%$$

Dari hasil perhitungan tersebut maka didapatkan bahwa antara dokumen D1 dengan *query* terindikasi plagiarisme sebesar 28,4591195%.

4.2.3 Metode *Cosine Similarity*

Korpus dan *query* yang digunakan dalam perhitungan *cosine similarity* (*cossim*) dapat dilihat pada Tabel 4.1 dan Tabel 4.11. Tahap pertama yaitu *text preprocessing* dan mendapatkan hasil berupa *term* atau token*,* maka langkah selanjutnya adalah menghitung bobot atau kemiripan tiap dokumen di dalam korpus terhadap *query*. Metode *cossim* meliputi perhitungan *term frequency, inverse document frequency,* normalisasi dan pembobotan kata menggunakan *cosine similarity*.

4.2.3.1 *Term Frequency*

Perhitungan *term frequency* merupakan proses pertama dalam perhitungan bobot menggunakan *cosine similarity.* Nilai *term frequency* dapat dilihat pada Tabel 4.7. Kemudian bobot *term frequency* dihitung berdasarkan Persamaan 2.4.

- *Term* = asian

  $TF_{asian,d1} = 1 + log\, tf_{asian,d1}$

  $TF_{asian,d1} = 1 + log\, 1$

  $TF_{asian,d1} = 0$

  Bobot *term frequency* selengkapnya dapat dilihat pada Tabel 4.13.

Tabel 4.13 Hasil Bobot *Term Frequency*

Term	D1	D2	D3
asian	1	0	0
atlet	1,30102999	0	0
bonus	1	0	0
games	1	0	0
indonesia	1	1	0
jamin	1	0	0
karier	1	0	0
medali	1	0	0
milik	1	0	0
negeri	1	0	0
pegawai	1	0	0
pns	1	0	0
polri	1	0	0
prestasi	1	0	0
raih	1	0	0
rumah	1	0	0
sempat	1	0	0
sipil	1	0	0
tes	1	0	0
tni	1	0	0
uang	1	0	0
calon	0	1,30102999	0
kunjung	0	1	0
massa	0	1	0
muhammadiyah	0	1	0
organisasi	0	1	0
ormas	0	1	0
pasang	0	1	0
pimpin	0	1	0

Term	D1	D2	D3
prabowo	0	1	0
presiden	0	1,30102999	0
pusat	0	1	0
sandiaga	0	1	0
sowan	0	1	0
subianto	0	1	0
uno	0	1	0
wakil	0	1	0
amien	0	0	1
banjir	0	0	1
banten	0	0	1
cawapresnya	0	0	1
damping	0	0	1
dukung	0	0	1
jokowi	0	0	1
khmaruf	0	0	1
pilih	0	0	1
umum	0	0	1

4.2.3.2 *Inverse Document Frequency*

Sebelum menghitung *inverse document frequency*, diperlukan nilai *document frequency* terlebih dahulu. Hasil *document frequency* dapat dilihat pada Tabel 4.8. Kemudian IDF dapat dihitung menggunakan Persamaan 2.5.

- *Term* = asian

$$IDF_{asian} = log \frac{N}{DF_{asian}}$$

$$IDF_{asian} = log \frac{3}{1}$$

$$IDF_{asian} = 0{,}477121$$

Hasil IDF selengkapnya dapat dilihat pada Tabel 4.14.

Tabel 4.14 Hasil *Inverse Document Frequency*

Term	IDF
asian	0,477121255
atlet	0,477121255
bonus	0,477121255
games	0,477121255
indonesia	0,176091259
jamin	0,477121255
karier	0,477121255
medali	0,477121255

Term	IDF
milik	0,477121255
negeri	0,477121255
pegawai	0,477121255
pns	0,477121255
polri	0,477121255
prestasi	0,477121255
raih	0,477121255
rumah	0,477121255
sempat	0,477121255
sipil	0,477121255
tes	0,477121255
tni	0,477121255
uang	0,477121255
calon	0,477121255
kunjung	0,477121255
massa	0,477121255
muhammadiyah	0,477121255
organisasi	0,477121255
ormas	0,477121255
pasang	0,477121255
pimpin	0,477121255
prabowo	0,477121255
presiden	0,477121255
pusat	0,477121255
sandiaga	0,477121255
sowan	0,477121255
subianto	0,477121255
uno	0,477121255
wakil	0,477121255
amien	0,477121255
banjir	0,477121255
banten	0,477121255
cawapresnya	0,477121255
damping	0,477121255
dukung	0,477121255
jokowi	0,477121255
khmaruf	0,477121255
pilih	0,477121255
umum	0,477121255

4.2.3.3 Bobot *Term Frequency – Inverse Document Frequency*

Perhitungan bobot *term frequency – inverse document frequency* atau bobot TF-IDF dapat dilakukan berdasarkan Persamaan 2.6.

- *Term* = asian

  $W_{d1,asian} = TF_{asian} \cdot IDF_{asian}$

  $W_{d1,asian} = 1 \cdot 0{,}477121255$

  $W_{d1,asian} = 0{,}477121255$

  Hasil bobot TF-IDF selengkapnya dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil Bobot TF-IDF

Term	D1	D2	D3
asian	0,477121255	0	0
atlet	0,620749064	0	0
bonus	0,477121255	0	0
games	0,477121255	0	0
indonesia	0,176091259	0,176091259	0
jamin	0,477121255	0	0
karier	0,477121255	0	0
medali	0,477121255	0	0
milik	0,477121255	0	0
negeri	0,477121255	0	0
pegawai	0,477121255	0	0
pns	0,477121255	0	0
polri	0,477121255	0	0
prestasi	0,477121255	0	0
raih	0,477121255	0	0
rumah	0,477121255	0	0
sempat	0,477121255	0	0
sipil	0,477121255	0	0
tes	0,477121255	0	0
tni	0,477121255	0	0
uang	0,477121255	0	0
calon	0	0,620749064	0
kunjung	0	0,477121255	0
massa	0	0,477121255	0
muhammadiyah	0	0,477121255	0
organisasi	0	0,477121255	0
ormas	0	0,477121255	0
pasang	0	0,477121255	0
pimpin	0	0,477121255	0
prabowo	0	0,477121255	0

Term	D1	D2	D3
presiden	0	0,620749064	0
pusat	0	0,477121255	0
sandiaga	0	0,477121255	0
sowan	0	0,477121255	0
subianto	0	0,477121255	0
uno	0	0,477121255	0
wakil	0	0,477121255	0
amien	0	0	0,477121255
banjir	0	0	0,477121255
banten	0	0	0,477121255
cawapresnya	0	0	0,477121255
damping	0	0	0,477121255
dukung	0	0	0,477121255
jokowi	0	0	0,477121255
khmaruf	0	0	0,477121255
pilih	0	0	0,477121255
umum	0	0	0,477121255

4.2.3.4 Normalisasi

Tahap selanjutnya yaitu menormalisasi nilai bobot dari *term* yang telah didapatkan berdasarkan Persamaan 2.7.

- *Term* = asian

$$W_{asian,d1} = \frac{W_{asian,d1}}{\sqrt{\sum_{t=1}^{n} W_{asian,d1}^2}}$$

$$W_{asian,d1} = \frac{0{,}477121255}{\sqrt{0{,}477121255^2 + 0{,}620749064^2 + \ldots + 0^2}}$$

$$W_{asian,d1} = 0{,}21911238$$

Hasil normalisasi selengkapnya dapat dilihat pada Tabel 4.16.

Tabel 4.16 Hasil Normalisasi

Term	D1	D2	D3
asian	0,21911238	0	0
atlet	0,28507177	0	0
bonus	0,21911238	0	0
games	0,21911238	0	0
indonesia	0,08086786	0,08817034	0
jamin	0,21911238	0	0
karier	0,21911238	0	0
medali	0,21911238	0	0

Term	D1	D2	D3
milik	0,21911238	0	0
negeri	0,21911238	0	0
pegawai	0,21911238	0	0
pns	0,21911238	0	0
polri	0,21911238	0	0
prestasi	0,21911238	0	0
raih	0,21911238	0	0
rumah	0,21911238	0	0
sempat	0,21911238	0	0
sipil	0,21911238	0	0
tes	0,21911238	0	0
tni	0,21911238	0	0
uang	0,21911238	0	0
calon	0	0,310814155	0
kunjung	0	0,238898531	0
massa	0	0,238898531	0
muhammadiyah	0	0,238898531	0
organisasi	0	0,238898531	0
ormas	0	0,238898531	0
pasang	0	0,238898531	0
pimpin	0	0,238898531	0
prabowo	0	0,238898531	0
presiden	0	0,310814155	0
pusat	0	0,238898531	0
sandiaga	0	0,238898531	0
sowan	0	0,238898531	0
subianto	0	0,238898531	0
uno	0	0,238898531	0
wakil	0	0,238898531	0
amien	0	0	0,316227766
banjir	0	0	0,316227766
banten	0	0	0,316227766
cawapresnya	0	0	0,316227766
damping	0	0	0,316227766
dukung	0	0	0,316227766
jokowi	0	0	0,316227766
khmaruf	0	0	0,316227766
pilih	0	0	0,316227766
umum	0	0	0,316227766

4.2.3.5 *Cosine Similarity* dan Persentase Plagiarisme

Tahap terakhir yaitu menghitung bobot kemiripan antar dokumen dengan *cosine similarity* berdasarkan Persamaan 2.8. *Query* dapat dilihat pada Tabel 4.11.

- *Cosine similarity* antara dokumen D1 dengan *query*

$$cossim(d_1, q) = \sum W_{d1} \cdot W_q$$

$$cossim(d_1, q) = 0{,}21911238 \cdot 0{,}403691674 + \ldots + 0{,}21911238 \cdot 0{,}403691674$$

$$cossim(d_1, q) = 0{,}569398872$$

Oleh karena nilai maksimal pada *cosine similarity* adalah 1, maka perhitungan persentase plagiarisme untuk *cossim* yaitu bobot yang telah didapatkan langsung dikalikan dengan 100%. Sehingga plagiarisme antara *query* dengan dokumen D1 yaitu sebesar 56,9398872%.

4.3 Perancangan Pengujian

Pengujian dalam penelitian ini yaitu dengan membandingkan hasil perhitungan plagiarisme menggunakan metode BM25 dengan *cosine similarity. Cosine similarity* digunakan sebagai pembanding karena metode tersebut juga merupakan salah satu metode dalam *Information Retrieval* yang digunakan untuk mencari nilai kemiripan antar dokumen. Nilai *threshold* dari tiap artikel *query* akan diubah terlebih dahulu untuk membatasi panjangnya. Nilai *threshold* yang akan digunakan yaitu sebesar 75%, 50%, dan 25%. Nilai *threshold* sebesar 75% menjelaskan bahwa artikel yang akan diuji merupakan artikel dengan judul yang sama dengan salah satu artikel di dalam korpus tetapi memiliki panjang yang lebih sedikit yaitu ¾ dari panjang artikel asli, begitu pula dengan nilai *threshold* lainnya. Hasil dari pengujian masing-masing *threshold* akan ditampilkan seperti Tabel 4.17.

Tabel 4.17 Pengujian dengan *Threshold* dan *Cosine Similarity*

Dokumen	Persentase Plagiarisme	
	BM25	*Cosine Similarity*

Selain itu dilakukan pula pengujian nilai parameter *b* dan *k* sesuai dengan batas nilai yang optimal dan nilai yang jauh dari batas optimal. Hasil pengujian parameter *b* dan *k* akan ditampilkan seperti Tabel 4.18.

Tabel 4.18 Hasil Pengujian Parameter BM25

Parameter		Threshold		
b	*k*	75%	50%	25%

BAB 5 IMPLEMENTASI

Bab ini menjelaskan tentang implementasi program berdasarkan perancangan yang telah dibuat.

5.1 Implementasi Program

Implementasi program merupakan suatu tahapan yang bertujuan untuk menerapkan perancangan yang telah dibuat agar menghasilkan suatu sistem yang sesuai. Dalam penelitian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 ini, perancangan yang telah dibuat akan diimplementasikan menggunakan bahasa pemrograman Python 2.7.

5.1.1 Implementasi *Text Preprocessing Query*

Pengolahan teks pada *query* dilakukan untuk menghasilkan kata atau token yang digunakan sebagai pedoman dalam mencari kemiripan dengan dokumen di dalam korpus. Implementasi *text preprocessing query* dapat dilihat pada Kode Program 5.1.

Kode Program 5.1 Implementasi *Text Preprocessing Query*

Algoritme *text preprocessing query*

```
1  def queryindexing():
2      querynya = open(query,'r')
3      baca_query = querynya.readlines()
4      for q in baca_query:
5          textQ = q.lower()
6      hapustanda = textQ.translate(string.maketrans(" ","
7      "),string.punctuation)
8      cleaningQ = re.sub(r'\d+',' ',hapustanda)
9      factory = StopWordRemoverFactory()
10     stopword = factory.create_stop_word_remover()
11     stopwordremQ = stopword.remove(cleaningQ)
12     factory2 = StemmerFactory()
13     stemmer = factory2.create_stemmer()
14     stemmingQ = stemmer.stem(stopwordremQ)
15     teksQ = stemmingQ.split()
16     for wordQ in teksQ:
17         if wordQ not in indeksQ:
18             indeksQ.append(wordQ)
19     return indeksQ
```

Penjelasan:

1. Baris 2 – 3 adalah proses membaca perbaris *query.*
2. Baris 4 – 5 adalah proses perulangan untuk setiap kata diubah menjadi huruf kecil semua (*case folding*).
3. Baris 6 – 8 adalah proses menghapus tanda baca dan angka (*cleaning*).
4. Baris 9 – 11 adalah proses menghapus *stopword* (*stopword removal*).
5. Baris 12 – 14 adalah proses mengembalikan kata dasar (*stemming*).
6. Baris 15 adalah proses tokenisasi.
7. Baris 16 – 18 adalah proses perulangan untuk menyimpan hasil *text preprocessing query.*
8. Baris 19 adalah proses mengembalikan nilai *indeksQ*.

5.1.2 Implementasi *Text Preprocessing* Korpus

Tahap pertama dalam implementasi deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 adalah *text preprocessing*. Dalam proses *text preprocessing* akan dibagi menjadi beberapa proses yaitu *case folding, cleaning, stopword removal,* dan *stemming*.

5.1.2.1 Implementasi *Case Folding*

Case folding merupakan proses untuk mengubah huruf besar menjadi huruf kecil. Implementasi *case folding* dapat dilihat pada Kode Program 5.2.

Kode Program 5.2 Implementasi *Case Folding*

```
Algoritme case folding
 1  hasilcasefolding = []
 2  def casefolding():
 3      files = glob.glob(korpus)
 4      for x in files:
 5          with open(x) as f:
 6              baca_korpus = f.readlines()
 7          for y in baca_korpus:
 8               kecil = y.lower()
 9           hasilcasefolding.append(kecil)
10      return hasilcasefolding
```

Penjelasan:

1. Baris 1 adalah inisialisasi variabel *hasilcasefolding* yang bertipe data *list.*
2. Baris 3 adalah inisialisasi variabel *files* yang berisi kumpulan dokumen atau artikel di dalam korpus. Fungsi dari penggunaan *glob* yaitu untuk menemukan semua artikel di dalam korpus.
3. Baris 4 adalah proses perulangan untuk setiap artikel di dalam korpus.
4. Baris 5 – 6 adalah proses membaca perbaris dari tiap artikel di dalam korpus.

5. Baris 7 – 8 adalah proses perulangan untuk setiap kata diubah menjadi huruf kecil semua.
6. Baris 9 adalah proses menambahkan hasil *case folding* ke dalam *list hasilcasefolding.*
7. Baris 10 adalah proses mengembalikan nilai *hasilcasefolding*.

5.1.2.2 Implementasi *Cleaning*

Cleaning merupakan proses untuk menghilangkan tanda baca dan angka di dalam teks. Implementasi *cleaning* dapat dilihat pada Kode Program 5.3.

Kode Program 5.3 Implementasi *Cleaning*

```
Algoritme cleaning
1  hasilcleaning = []
2  def cleaning():
3      files = casefolding()
4      for x in files:
5          hapustanda = x.translate(string.maketrans(" ","
6           "),string.punctuation)
7          cleaning = re.sub(r'\d+',' ',hapustanda)
8          hasilcleaning.append(cleaning)
9      return hasilcleaning
```

Penjelasan:

1. Baris 1 adalah inisialisasi variabel *hasilcleaning* yang bertipe data *list.*
2. Baris 3 adalah proses pemanggilan *casefolding* yang bertujuan untuk mengambil nilai dari *hasilcasefolding*.
3. Baris 4 – 7 adalah proses perulangan untuk mencari tanda baca dan angka kemudian menghapusnya.
4. Baris 8 adalah proses menambahkan hasil *cleaning* ke dalam *list hasilcleaning.*
5. Baris 9 adalah proses mengembalikan nilai *hasilcleaning*.

5.1.2.3 Implementasi *Stopword Removal*

Stopword removal merupakan proses untuk menghapus kata-kata yang tidak penting. Daftar kata di dalam *stopword* mencakup kata hubung dan kata yang tidak bermakna. Implementasi *stopword removal* dapat dilihat pada Kode Program 5.4.

Kode Program 5.4 Implementasi *Stopword Removal*

```
Algoritme stopword removal
1  hasilstopwordremoval = []
2  def stopwordremoval():
```

Algoritme *stopword removal*	
3	`    files = cleaning()`
4	`    for x in files:`
5	`        hapusstopword = stopword.remove(x)`
6	`        hasilstopwordremoval.append(hapusstopword)`
7	`    return hasilstopwordremoval`

Penjelasan:

1. Baris 1 adalah inisialisasi variabel *hasilstopwordremoval* yang bertipe data *list.*
2. Baris 3 adalah proses pemanggilan *cleaning* yang bertujuan untuk mengambil nilai dari *hasilcleaning*.
3. Baris 4 – 5 adalah proses perulangan untuk menghapus kata *stopword* menggunakan *library* Sastrawi.
4. Baris 6 adalah proses menambahkan hasil *stopword removal* ke dalam *list hasilstopwordremoval.*
5. Baris 7 adalah proses mengembalikan nilai *hasilstopwordremoval*.

5.1.2.4 Implementasi *Stemming*

Stemming merupakan proses untuk menghapus imbuhan kata dan mengembalikan bentuk asli atau kata dasar dari setiap kata tersebut. Implementasi *stemming* dapat dilihat pada Kode Program 5.5.

Kode Program 5.5 Implementasi *Stemming*

Algoritme *stemming*	
1	`hasilstemming = []`
2	`def stemming():`
3	`    files = stopwordremoval()`
4	`    for x in files:`
5	`        katadasar = stemmer.stem(x)`
6	`        hasilstemming.append(katadasar)`
7	`    return hasilstemming`

Penjelasan:

1. Baris 1 adalah inisialisasi variabel *hasilstemming* yang bertipe data *list.*
2. Baris 3 adalah proses pemanggilan *stopwordremoval* yang bertujuan untuk mengambil nilai dari *hasilstopwordremoval*.
3. Baris 4 – 5 adalah proses perulangan untuk mengembalikan kata berimbuhan menjadi kata dasar menggunakan *library* Sastrawi.
4. Baris 6 adalah proses menambahkan hasil *stemming* ke dalam *list hasilstemming.*
5. Baris 7 adalah proses mengembalikan nilai *hasilstemming*.

5.1.3 Implementasi Metode BM25 dan Persentase Plagiarisme

Setelah melalui proses *text preprocessing,* maka proses selanjutnya yaitu menghitung nilai kemiripan antar dokumen atau artikel menggunakan metode BM25. Proses dalam BM25 akan dibagi menjadi beberapa proses, yaitu menghitung *term frequency*, *inverse document frequency*, dan BM25. Setelah mendapatkan nilai kemiripan menggunakan BM25, langkah selanjutnya yaitu menghitung persentase plagiarisme artikel.

5.1.3.1 Implementasi *Term Frequency*

Perhitungan *term frequency* merupakan proses untuk menghitung kemunculan atau frekuensi tiap kata pada masing-masing artikel yang ada di dalam korpus. Implementasi *term frequency* dapat dilihat pada Kode Program 5.6.

Kode Program 5.6 Implementasi *Term Frequency*

Algoritme *term frequency*

```
1   indekstf = []
2   def termfreq():
3       files = stemming()
4       for x in files:
5           tf = {}
6           pisah = x.split()
7           for x in pisah:
8               if x not in tf:
9                   tf[x] = 0
10              tf[x] += 1
11          indekstf.append(tf)
12      return indekstf
```

Penjelasan:

1. Baris 1 adalah inisialisasi variabel *indekstf* yang bertipe data *list.*
2. Baris 3 adalah proses pemanggilan *stemming* yang bertujuan untuk mengambil nilai dari *hasilstemming*.
3. Baris 4 – 6 adalah proses perulangan untuk memisah kata dari *hasilstemming* menjadi *term*. Variabel *tf* bertipe data *dict* digunakan untuk menyimpan *term* dan hasil TF dalam bentuk *key* dan *value.*
4. Baris 7 – 10 adalah proses perulangan untuk menghitung kemunculan *term x* di setiap artikel. Ketika *x* tidak ada di dalam *tf*, maka nilai *tf* dari *x* tersebut bernilai 0. Ketika *x* ada di dalam *tf*, maka nilai *tf* dari *x* akan bertambah 1.
5. Baris 11 adalah proses menambahkan hasil *tf* ke dalam *list indekstf.*
6. Baris 12 adalah proses mengembalikan nilai *indekstf*.

5.1.3.2 Implementasi *Inverse Document Frequency*

Sebelum menghitung *inverse document frequency* diperlukan nilai *document frequency* yang merupakan proses untuk menghitung kemunculan atau frekuensi tiap kata di tiap-tiap artikel di dalam korpus. Implementasi *inverse document frequency* dapat dilihat pada Kode Program 5.7.

Kode Program 5.7 Implementasi *Inverse Document Frequency*

Algoritme *inverse document frequency*

```
1   n = jumlahdokumen()
2   indeksidf = []
3   def idocfreq():
4       tf = termfreq()
5       df = {}
6       idf = {}
7       for x in tf:
8           key = x.keys()
9           for k in key:
10              if k not in df:
11                  df[k] = 0
12              df[k] += 1
13          for x, y in iteritems(df):
14              if x not in idf:
15                  idf[x] = 0
16              idf[x] = math.log10((n-y+0.5)/(y+0.5))
17      indeksidf.append(idf)
18      return indeksidf
```

Penjelasan:

1. Baris 1 adalah proses pemanggilan *jumlahdokumen.*
2. Baris 2 adalah inisialisasi variabel *indeksidf* yang bertipe data *list.*
3. Baris 4 adalah proses pemanggilan *termfreq* yang bertujuan untuk mengambil nilai dari *indekstf*.
4. Baris 5 – 6 adalah inisialisasi variabel *df* dan *idf* bertipe data *dict,* masing-masing untuk menyimpan hasil DF dan IDF dalam bentuk *key* dan *value*.
5. Baris 7 – 8 adalah proses perulangan untuk mencari *term.*
6. Baris 9 – 12 adalah proses perulangan untuk menghitung jumlah artikel yang mengandung *term k* (*document frequency*). Ketika *k* tidak ada di dalam *df*, maka nilai *df* dari *k* tersebut bernilai 0. Ketika *k* ada di dalam *df*, maka nilai *df* dari *k* akan bertambah 1.

7. Baris 13 – 16 adalah proses perulangan untuk menghitung *inverse document frequency* berdasarkan nilai DF di dalam *dict df*, *x* berfungsi sebagai *key* dan *y* berfungsi sebagai *value.* Ketika *x* tidak ada di dalam *idf*, maka nilai *idf* dari *x* tersebut bernilai 0. Ketika *x* ada di dalam *idf*, maka nilai *idf* dari *x* dihitung berdasarkan rumus pada Persamaan 2.3.
8. Baris 17 adalah proses menambahkan hasil *idf* ke dalam *list indeksidf.*
9. Baris 18 adalah proses mengembalikan nilai *indeksidf*.

5.1.3.3 Implementasi BM25 dan Persentase Plagiarisme

Metode BM25 digunakan untuk menghitung kemiripan antar artikel. Implementasi BM25 dapat dilihat pada Kode Program 5.8.

Kode Program 5.8 Implementasi BM25 dan Perhitungan Persentase Plagiarisme

Algoritme BM25 dan perhitungan persentase plagiarisme

```
1  avg = reratapanjangdok()
2  queryin = queryindexing()
3  bobotdok = []
4  bobotdok100 = []
5  def bm25():
6      for x in indekstf:
7          bobot = 0
8          for y,z in iteritems(x):
9              for x2 in indeksidf:
10                 for y2,z2 in iteritems(x2):
11                     if y == y2:
12                         for q in queryin:
13                             if y2 == q:
14                                 bobot +=
15                                 (z2*(z*(k+1))/(z+k*(1-b+b*
16                                 panjang[korpuske]/avg)))
17         bobotdok.append(bobot)
18     with open('hasilbobot100.csv') as f:
19         rows = csv.reader(f)
20         for row in rows:
21             bobotdok100.append(row[korpuske])
22         bobot100 = float(bobotdok100[korpuske])
23         plag = bobotdok[korpuske]/bobot100*100
24         print plag,'%'
```

Penjelasan:

1. Baris 1 – 2 adalah proses pemanggilan *reratapanjangdok* dan *queryindexing.*
2. Baris 3 – 4 adalah inisialisasi variabel *bobotdok* dan *bobotdok100* yang bertipe data *list.*
3. Baris 6 – 8 adalah proses perulangan untuk setiap *x* di dalam *indekstf* untuk mendapatkan nilai *key* dan *value,* serta inisialisasi awal nilai dari variabel *bobot*.
4. Baris 9 – 10 adalah proses perulangan untuk setiap *x2* di dalam *indeksidf* untuk mendapatkan nilai *key* dan *value*.
5. Baris 11 – 12 adalah proses seleksi kondisi, ketika nilai *key y* di dalam *x* dan nilai *key y2* di dalam *x2* sama, maka dilakukan perulangan untuk setiap *q* di dalam *queryin*.
6. Baris 13 – 17 adalah proses seleksi kondisi, ketika nilai *key y2* di *x2* sama dengan *q* di *queryin* maka dilakukan perhitungan BM25. Ketika kondisi tidak terpenuhi, program akan langsung menambahkan nilai bobot ke dalam *list bobotdok*.
7. Baris 18 – 19 adalah proses membaca *hasilbobot100.csv*, yaitu file yang berisi nilai bobot artikel ketika *threshold* 100%.
8. Baris 20 – 22 adalah perulangan untuk tiap *row* di dalam *rows* file csv dan menambahkan nilai bobot *threshold* 100% ke dalam *list bobotdok100*, kemudian mengubahnya menjadi tipe data *float.*
9. Baris 23 – 24 adalah proses perhitungan persentase plagiarisme dan menampilkannya sebagai *output*.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini akan menjelaskan tentang hasil pengujian yang telah dilakukan serta pembahasan dan analisis mengenai hasil yang didapatkan.

6.1 Pengujian dengan *Threshold* dan *Cosine Similarity*

Pengujian yang dilakukan yaitu menggunakan dokumen berita sebanyak 30 artikel dari berbagai macam topik dan dikumpulkan sebagai korpus. *Query* yang digunakan dalam pengujian ini adalah masing-masing dokumen di dalam korpus itu sendiri.

6.1.1 Hasil Pengujian dengan *Threshold* 75% dan *Cosine Similarity*

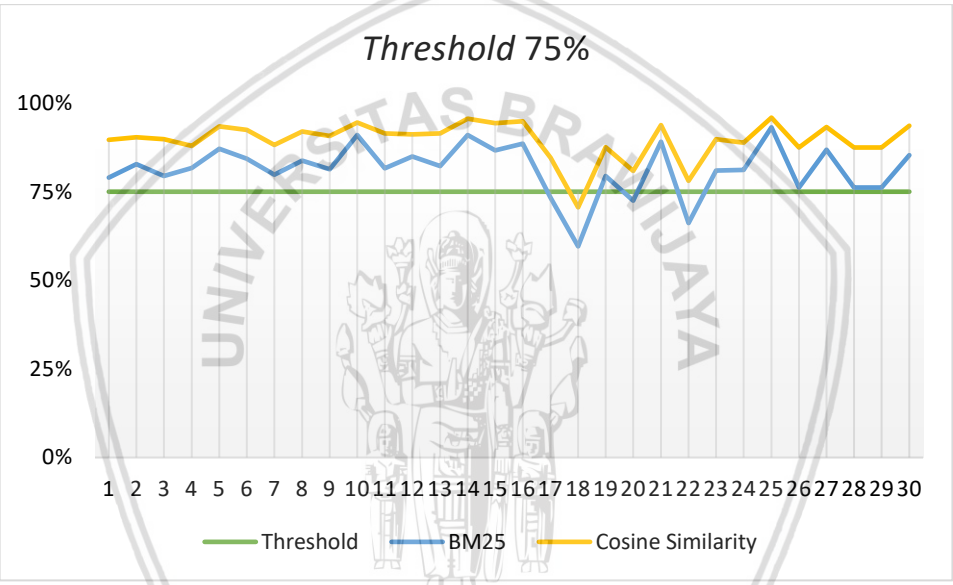
Threshold 75% menunjukkan bahwa artikel *query* yang semula mempunyai *threshold* 100% akan dipotong sebesar 25%, sehingga panjang artikel *query* yang digunakan untuk pengujian ini menjadi ¾ dari artikel yang asli. Hasil pengujian dengan *threshold* 75% dan *cosine similarity* dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Pengujian dengan *Threshold* 75% dan *Cosine Similarity*

Dokumen	Persentase Plagiarisme	
	BM25	*Cosine Similarity*
D1	79%	90%
D2	83%	90%
D3	79%	90%
D4	82%	88%
D5	87%	93%
D6	84%	92%
D7	80%	88%
D8	84%	92%
D9	81%	91%
D10	91%	95%
D11	82%	91%
D12	85%	91%
D13	82%	91%
D14	91%	96%
D15	87%	94%
D16	89%	95%
D17	73%	85%
D18	60%	71%
D19	79%	88%
D20	72%	81%
D21	89%	94%
D22	66%	78%
D23	81%	90%

	Persentase Plagiarisme	
Dokumen	**BM25**	***Cosine Similarity***
D24	81%	89%
D25	93%	96%
D26	76%	88%
D27	87%	93%
D28	76%	88%
D29	76%	88%
D30	85%	94%

Hasil yang ditunjukkan pada Tabel 6.1 dapat disajikan dalam bentuk grafik seperti pada Gambar 6.1.



Gambar 6.1 Hasil Pengujian dengan *Threshold* 75% dan *Cosine Similarity*

Dari Gambar 6.1 terlihat bahwa hasil dari perhitungan plagiarisme menggunakan metode BM25 lebih baik daripada metode *cosine similarity.* Di dalam artikel yang telah dipotong sebanyak 75% tersebut, perbandingan kata yang telah dihasilkan dari proses *text preprocessing* dengan panjang kata yang sebenarnya tidak sama. Ketika suatu artikel mengandung sedikit *stopword*, maka kata di dalam artikel tersebut tidak banyak berubah dan perbedaan panjang artikel tidak terlalu jauh. Tetapi ketika artikel yang telah dipotong ternyata mengandung banyak *stopword*, maka panjang artikel tersebut berkurang karena banyaknya kata yang harus dihapus. Seperti yang dapat dilihat pada Tabel 6.1, dimana hasil dari dokumen D18 berada dibawah *threshold.*

Dapat pula dilihat hasil dari dokumen D22 pada Tabel 6.1. Nilai yang dihasilkan untuk dokumen D22 menggunakan BM25 lebih kecil atau berada di bawah *threshold*, sedangkan nilai dari perhitungan *cosine similarity* lebih besar atau berada di atas *threshold*. Hal ini dipengaruhi oleh nilai minimal dari kedua metode tersebut. Nilai minimal dari *cosine similarity* adalah 0, sedangkan nilai minimal dari

perhitungan BM25 tidak terbatas. Sehingga ketika menjumlahkan total bobot kata dalam dokumen, perhitungan menggunakan BM25 mungkin saja menemukan hasil yang negatif dan mengakibatkan nilai BM25 lebih kecil daripada *cosine similarity.*

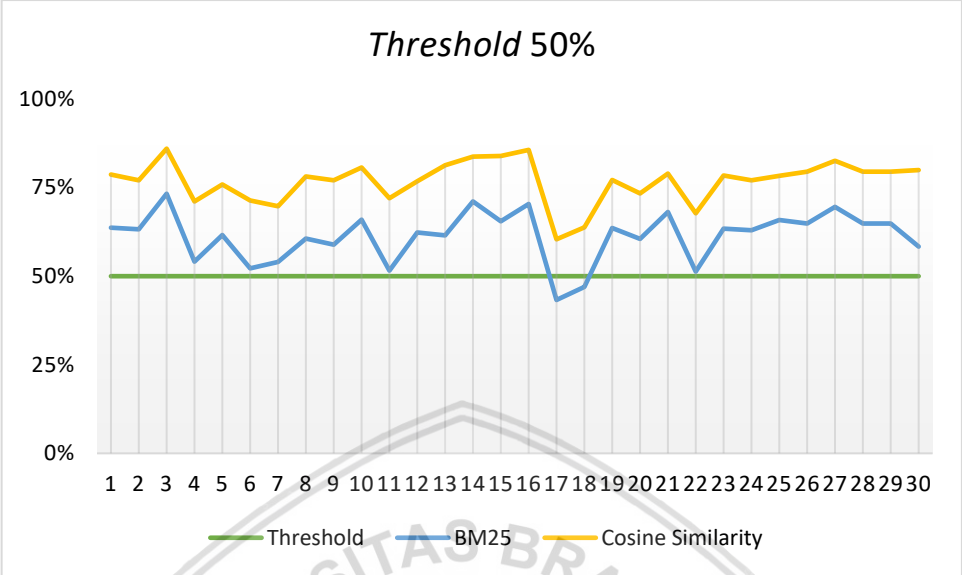
6.1.2 Hasil Pengujian dengan *Threshold* 50% dan *Cosine Similarity*

Threshold 50% menunjukkan bahwa artikel *query* memiliki panjang setengah dari artikel yang asli. Hasil pengujian dengan *threshold* 50% dan *cosine similarity* dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian dengan *Threshold* 50% dan *Cosine Similarity*

	Persentase Plagiarisme	
Dokumen	**BM25**	***Cosine Similarity***
D1	64%	79%
D2	63%	77%
D3	73%	86%
D4	54%	71%
D5	62%	76%
D6	52%	71%
D7	54%	70%
D8	61%	78%
D9	59%	77%
D10	66%	81%
D11	52%	72%
D12	62%	77%
D13	61%	81%
D14	71%	84%
D15	66%	84%
D16	70%	86%
D17	43%	60%
D18	47%	64%
D19	64%	77%
D20	61%	73%
D21	68%	79%
D22	51%	68%
D23	63%	78%
D24	63%	77%
D25	66%	78%
D26	65%	79%
D27	70%	83%
D28	65%	79%
D29	65%	79%
D30	58%	80%

Hasil yang ditunjukkan pada Tabel 6.2 dapat disajikan dalam bentuk grafik seperti pada Gambar 6.2.



Gambar 6.2 Hasil Pengujian dengan *Threshold* 50% dan *Cosine Similarity*

Dari Gambar 6.2 juga terlihat bahwa hasil dari perhitungan plagiarisme menggunakan metode BM25 lebih baik daripada menggunakan metode *cosine similarity.* Nilai yang dihasilkan dari perhitungan BM25 lebih mendekati *threshold.* Tetapi pada Tabel 6.2 nilai yang dihasilkan dari dokumen D17 menggunakan BM25 lebih kecil atau berada di bawah *threshold*, sedangkan nilai dari perhitungan *cosine similarity* lebih besar atau berada di atas *threshold*. Hal ini dipengaruhi oleh nilai minimal dari kedua metode tersebut dan *term* hasil *text preprocessing* yang ternyata tidak sesuai dengan besar *threshold*. Nilai minimal dari perhitungan BM25 tidak terbatas. Sehingga ketika menjumlahkan total bobot dokumen, perhitungan menggunakan BM25 mungkin saja ditemukan hasil yang negatif dan mengakibatkan nilai BM25 menjadi lebih kecil daripada *cosine similarity.*

6.1.3 Hasil Pengujian dengan *Threshold* 25% dan *Cosine Similarity*

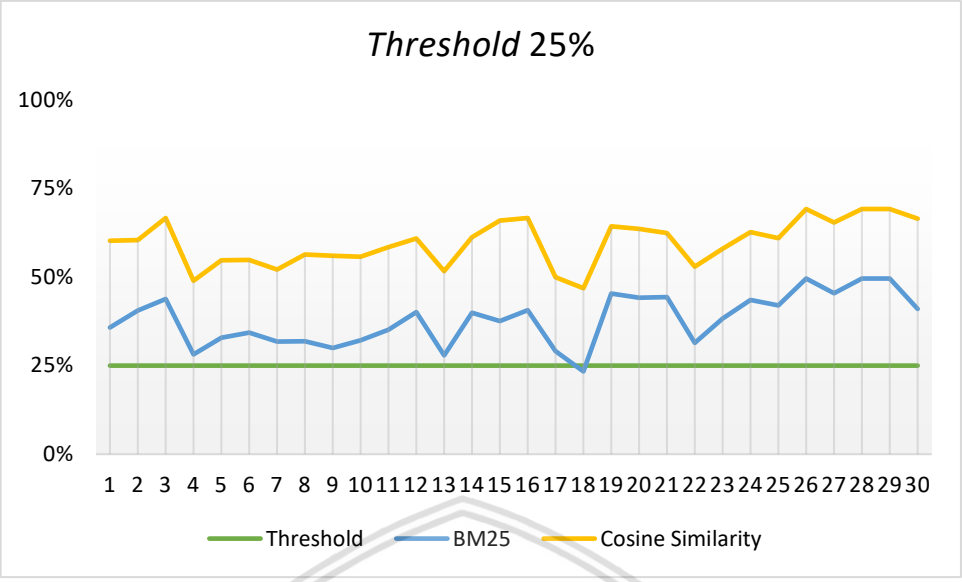
Threshold 25% menunjukkan bahwa artikel *query* memiliki panjang sebesar ¼ dari artikel yang asli. Hasil pengujian dengan *threshold* 25% dan *cosine similarity* dapat dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian dengan *Threshold* 25% dan *Cosine Similarity*

	Persentase Plagiarisme	
Dokumen	**BM25**	***Cosine Similarity***
D1	36%	60%
D2	41%	60%
D3	44%	67%
D4	28%	49%
D5	33%	55%

Dokumen	Persentase Plagiarisme	
	BM25	*Cosine Similarity*
D6	34%	55%
D7	32%	52%
D8	32%	56%
D9	30%	56%
D10	32%	56%
D11	35%	58%
D12	40%	61%
D13	28%	52%
D14	40%	61%
D15	38%	66%
D16	41%	67%
D17	29%	50%
D18	23%	47%
D19	45%	64%
D20	44%	64%
D21	44%	62%
D22	31%	53%
D23	38%	58%
D24	44%	63%
D25	42%	61%
D26	50%	69%
D27	45%	65%
D28	50%	69%
D29	50%	69%
D30	41%	66%

Hasil yang ditunjukkan pada Tabel 6.3 dapat disajikan dalam bentuk grafik seperti pada Gambar 6.3.



Gambar 6.3 Hasil Pengujian dengan *Threshold* 25% dan *Cosine Similarity*

Dari Gambar 6.3 juga terlihat bahwa hasil dari perhitungan plagiarisme menggunakan metode BM25 lebih baik daripada menggunakan metode *cosine similarity.* Nilai yang dihasilkan dari perhitungan BM25 lebih mendekati *threshold.* Tetapi diantara ketiga pengujian *threshold,* hasil perhitungan menggunakan BM25 dan *cosine similarity* dari *threshold* 25% ini mempunyai perbadingan yang paling besar. Hal ini dikarenakan panjang dokumen atau artikel dan kata hasil *text preprocessing* yang semakin kecil. Hal ini jelas berpengaruh pada BM25 yang menggunakan panjang dokumen sebagai salah satu parameternya. Tetapi berbeda untuk *cosine similarity.* Metode *cosine similarity* tidak tergantung pada panjang dokumen, sehingga ketika semakin banyaknya kata yang sama atau ditemukan, maka semakin tinggi pula nilainya.

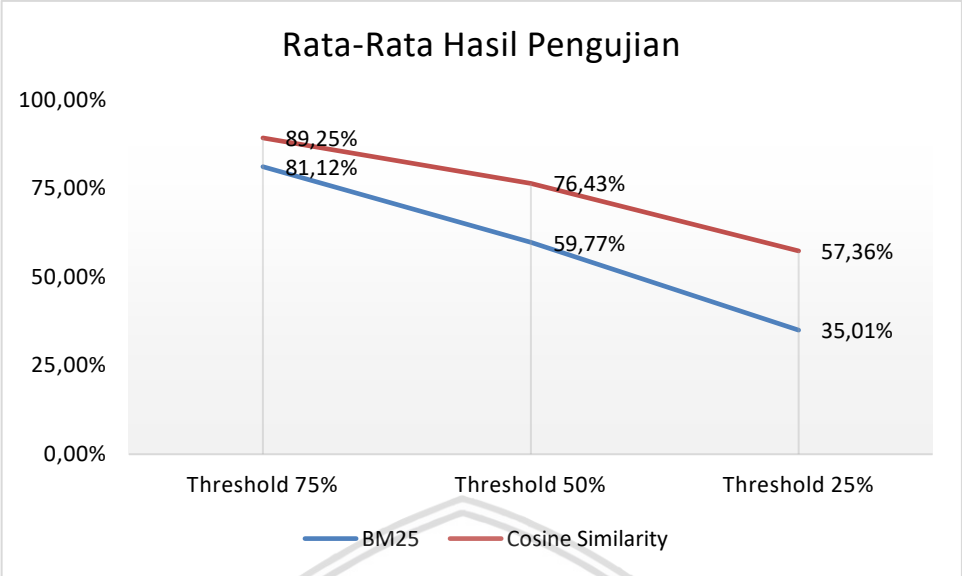
6.1.4 Rata-Rata Hasil Pengujian

Setelah mendapatkan hasil pengujian dari masing-masing *threshold*, maka langkah selanjutnya yaitu menghitung rata-rata hasil pengujiannya. Perhitungan rata-rata ini bertujuan untuk menyimpulkan hasil pengujian secara umum. Tabel rata-rata hasil pengujian dapat dilihat pada Tabel 6.4.

Tabel 6.4 Rata-Rata Hasil Pengujian

Threshold	**Rata-Rata**		**Selisih dengan *Threshold***	
	BM25	***Cosine Similarity***	**BM25**	***Cosine Similarity***
75%	81,12%	89,25%	6,12%	14,25%
50%	59,77%	76,43%	9,77%	26,43%
25%	35,01%	57,36%	10,01%	32,36%

Hasil rata-rata yang ditunjukkan pada Tabel 6.4 dapat disajikan dalam bentuk grafik seperti pada Gambar 6.4.



Gambar 6.4 Rata-Rata Hasil Pengujian

Dari Gambar 6.4 dapat disimpulkan bahwa hasil perhitungan menggunakan BM25 secara rata-rata berada di bawah *cosine similarity.* Hal ini menunjukkan bahwa BM25 lebih baik dengan hasil yang bisa lebih mendekati nilai *threshold.* Sedangkan nilai yang dihasilkan dari perhitungan *cosine similarity* lebih tinggi atau lebih jauh daripada nilai BM25 ke *threshold.* Perbedaan mencolok yang dihasilkan dari metode BM25 dan *cosine similarity* dengan *threshold* disebabkan oleh pengambilan kata atau pemotongan kata yang sesuai dengan nilai *threshold* belum tentu diproses semua. Misalnya di dalam artikel yang telah dipotong sebanyak 75%, perbandingan kata yang telah dihasilkan dari proses *text preprocessing* dengan panjang kata yang sebenarnya tidak sama. Ketika suatu artikel mengandung sedikit *stopword*, maka kata di dalam artikel tersebut tidak banyak berubah dan perbedaan panjang artikel tidak terlalu jauh. Tetapi ketika artikel yang telah dipotong ternyata mengandung banyak *stopword*, maka panjang artikel tersebut berkurang karena banyaknya kata yang harus dihapus. Selain itu, panjang artikel dan rata-rata panjang artikel di dalam korpus berpengaruh pada perhitungan BM25, sedangkan pada *cosine similarity* tidak. Ketika panjang artikel sangat kecil tetapi apabila ditemukan banyak kesamaan antara *query* dengan artikel, maka nilai *cosine similarity* juga tinggi.

6.2 Pengujian Parameter BM25

Berdasarkan Persamaan 2.2, parameter dalam BM25 yang dapat diubah atau disesuaikan dengan sistem yang akan dibuat yaitu paramater *b* dan *k. Query* yang digunakan pada pengujian ini yaitu dokumen D4, karena berdasarkan pengujian dengan *threshold* dokumen tersebut menghasilkan nilai plagiarisme yang mendekati *threshold.* Hasil dari pengujian dengan merubah nilai parameter *b* dan *k* dapat dilihat pada Tabel 6.5.

Tabel 6.5 Hasil Pengujian Parameter BM25

Parameter		Threshold		
b	***k***	**75%**	**50%**	**25%**
0,8	2,5	82%	55%	29%
0,7	1,75	82%	54%	28%
0,6	1,9	82%	54%	28%
0,75	1,5	81%	54%	28%
0,65	1,5	81%	54%	28%
0,7	1,4	81%	54%	27%
0,6	1,3	81%	54%	27%

Berdasarkan Tabel 6.5, ketika nilai parameter *b* dan *k* berada dalam batas nilai yang optimal, maka hasil persentase plagiarismenya tidak jauh berbeda. Sedangkan ketika nilainya diubah menjauhi batas seperti pada baris pertama, maka nilainya semakin besar dan menjauhi *threshold.* Maka dari hasil pengujian parameter ini dapat disimpulkan bahwa nilai optimal untuk parameter *b* dan *k* tidak dapat ditentukan secara tepat. Nilai dari parameter *b* dan *k* dapat ditentukan dengan melihat jenis dokumen untuk menghasilkan perhitungan BM25 yang optimal dan menyesuaikan kebutuhan sistem yang akan dibuat.

6.3 Hasil *Precision*

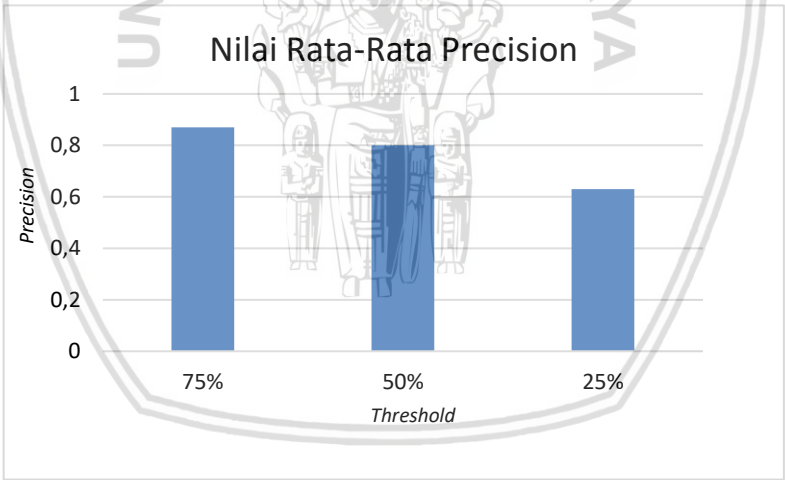
Dari pengujian yang telah dilakukan dengan masing-masing nilai *threshold,* maka didapatkan hasil evaluasi berupa nilai *precision* yang dapat dilihat pada Tabel 6.6.

Tabel 6.6 Hasil *Precision*

	Threshold		
Dokumen	**75%**	**50%**	**25%**
D1	1,00	1,00	1,00
D2	0,89	1,00	0,67
D3	0,83	1,00	1,00
D4	0,75	0,67	0,50
D5	0,90	0,71	0,75
D6	0,83	0,75	0,50
D7	0,92	0,56	0,60
D8	0,89	1,00	1,00
D9	1,00	0,83	1,00
D10	1,00	1,00	1,00
D11	0,67	0,75	1,00
D12	0,94	0,73	0,67
D13	1,00	1,00	1,00
D14	0,83	0,50	0,50
D15	1,00	1,00	0,50
D16	1,00	1,00	0,50

	Threshold		
Dokumen	**75%**	**50%**	**25%**
D17	0,91	1,00	1,00
D18	0,86	1,00	1,00
D19	0,82	0,71	0,50
D20	0,82	0,71	0,50
D21	0,89	0,83	0,67
D22	0,75	0,80	0,67
D23	1,00	0,60	0,67
D24	0,89	0,83	0,67
D25	0,92	0,75	0,50
D26	0,75	0,67	0,00
D27	0,67	0,50	0,00
D28	0,75	0,67	0,00
D29	0,75	0,67	0,00
D30	1,00	0,67	0,50
Rata-rata	**0,87**	**0,80**	**0,63**

Dari Tabel 6.6, rata-rata nilai *precision* dapat digambarkan dengan grafik seperti pada Gambar 6.5.



Gambar 6.5 Nilai Rata-Rata *Precision*

Nilai rata-rata *precision* paling tinggi didapatkan ketika nilai *threshold* sebesar 75%. Hal ini dikarenakan bahwa semakin panjang artikel *query*, maka semakin banyak pula kalimat-kalimat di dalamnya. Oleh karena itu, persamaan kalimat yang dianggap plagiasi oleh sistem dan hasil analisis dari pakar mempunyai banyak kemiripan, sehingga nilai *precision*-nya tinggi. Berbeda dengan *threshold* 25% yang menghasilkan nilai *precision* paling rendah. Hal ini disebabkan oleh semakin kecilnya panjang artikel maka semakin sedikit pula kalimat yang ada dan hasil analisis dari pakar berbeda dengan apa yang dihasilkan oleh sistem. Seperti pada dokumen D26 yang menghasilkan *precision* 0,00 saat *threshold* 25%.

BAB 7 PENUTUP

Bab ini akan menjelaskan tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang diharapkan dapat memperbaiki penelitian selanjutnya.

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang didapat adalah sebagai berikut:

1. Pengujian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 dapat dilakukan dengan mengubah panjang artikel yang dibatasi oleh *threshold* sebesar 75%, 50%, dan 25%. Pengujian juga dilakukan dengan membandingkan hasil plagiarisme menggunakan metode BM25 dengan *cosine similarity* dan perubahan nilai parameter *b* dan *k* pada BM25.
2. Hasil yang diperoleh dari pengujian deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 hampir mendekati nilai *threshold.* Masing-masing *threshold* mendapatkan nilai rata-rata sebesar 81,12%, 59,77%, dan 35,01% dengan nilai rata-rata *precision* sebesar 0,87, 0,80, dan 0,63.
3. Hasil perbandingan deteksi plagiarisme pada artikel berita berbahasa Indonesia menggunakan BM25 dengan *cosine similarity* menunjukkan bahwa BM25 lebih baik karena nilai yang dihasilkan lebih mendekati *threshold* dengan selisih sebesar 6,12%, 9,77%, dan 10,01%. Sedangkan hasil perhitungan menggunakan metode *cosine similarity* mendapatkan selisih sebesar 14,25%, 26,43%, dan 32,36%.

7.2 Saran

Dalam penelitian deteksi plagiarisme ini menggunakan pengujian yang hanya dilakukan dengan membandingkan BM25 dengan *cosine similarity.* Sehingga untuk penelitian selanjutnya diharapkan dapat menggunakan metode pembanding lainnya atau bahkan dengan sistem deteksi plagiarisme yang sudah ada untuk menemukan hasil yang lebih akurat.

DAFTAR PUSTAKA

Burrows, S., Tahaghoghi, S.M.M. dan Zobel, J., 2007. Efficient Plagiarism Detection for Large Code Repositories. *Softw. Pract. Exper.*, [daring] hal.151–175. Tersedia pada: <www.interscience.wiley.com>.

Dang, S. dan Ahmad, P.H., 2014. Text Mining : Techniques and its Application. *IJETI International Journal of Engineering & Technology Innovations*, [daring] 1(4), hal.22–25. Tersedia pada: <www.IJETI.com%0Awww.ijeti.com>.

Feldman, R. dan Sanger, J., 2007. *The Text Mining Handbook*. [daring] New York: Cambridge University Press. Tersedia pada: <www.cambridge.org>.

Herwijayanti, B., Ratnawati, D.E. dan Muflikhah, L., 2018. Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(1), hal.306–312.

Ismail dan Yunarso, E.W., 2014. Aplikasi Berbasis Web Pendeteksi Plagiarisme Menggunakan Algoritma Himpunan Kata. *Infotel*, 6(2), hal.95–100.

iThenticate, 2013. Survey Summary | Research Ethics: Decoding Plagiarism and Attribution in Research. [daring] hal.5. Tersedia pada: <http://www.ithenticate.com/Portals/92785/resources/decoding-plagiarism-and-attribution>.

Juwito, 2008. *Menulis Berita Dan Feature's*. iv ed. Surabaya: Unesa University Press.

Kalra, V. dan Aggarwal, R., 2017. Importance of Text Data Preprocessing & Implementation in RapidMiner. In: *Proceedings of the First International Conference on Information Technology and Knowledge Management*. New Delhi, hal.71–75.

Kienreich, W., Granitzer, M., Sabol, V. dan Klieber, W., 2006. Plagiarism Detection in Large Sets of Press Agency News Articles. hal.181–188.

Kock, N. dan Davison, R., 2003. Dealing With Plagiarism In The Information Systems Research Community: A Look At Factors That Drive Plagiarism And Ways To Address Them. *IS Research*, 27(4), hal.511–532.

Manning, C.D., Raghavan, P. dan Schutze, H., 2009. *An Introduction To Informational Retreival*. Online ed. [daring] *Information Retrieval*, England: Cambridge University Press. Tersedia pada: <http://www.informationretrieval.org/>.

Peraturan Dewan Pers Nomor : 6 / Peraturan-DP / V / 2008 Tentang Pengesahan Surat Keputusan Dewan Pers Nomor : 03 / SK-DP / III / 2006 Tentang Kode Etik Jurnalistik Sebagai Peraturan Dewan Pers. Jakarta: Dewan Pers Nasional.

Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010. Jakarta: Menteri Pendidikan Nasional.

Robertson, S. dan Zaragoza, H., 2009. *The Probabilistic Relevance Framework: BM25 and Beyond*. *Foundations and Trends® in Information Retrieval*, NOW The Essence of Knowledge.

Russel, S. dan Norvig, P., 2010. *Artificial Intelligence A Modern Approach*. Third ed. [daring] *Prentice Hall Series In Artificial Intelligence*, New Jersey: Pearson Education. Tersedia pada: <www.pearsonhighered.com>.

Sanjalawe, Y. k. dan Anbar, M., 2017. The Plagiarism Detection Systems for Higher Education - A Case Study in Saudi Universities. *International Journal of Software Engineering & Applications*, 8(2), hal.33–49.

Undang-Undang Republik Indonesia Nomor 19 Tahun 2002 Tentang Hak Cipta.[daring] Tersedia pada: <www.hukumonline.com>.

Wijaya, H., 2017. *Plagiarisme dalam Penelitian*. [daring] Tersedia pada: <https://www.researchgate.net/publication/312032045>.

LAMPIRAN A SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Drs. Joko Wahono

NIP : 19640315 199103 1 009

Jabatan : Guru Bahasa Indonesia SMK N 1 Ngawi

Dengan ini saya menyatakan bersedia sebagai pakar demi menunjang data penelitian skripsi mahasiswa yang berjudul "Deteksi Plagiarisme Pada Artikel Berita Berbahasa Indonesia Menggunakan BM25" yang dilakukan oleh:

Nama : Dina Dahniawati

NIM : 155150200111081

Demikian pernyataan ini saya buat untuk dapat digunakan sebagaimana mestinya.

Ngawi, 15 Maret 2019

Drs. Joko Wahono

NIP. 19640315 199103 1 009

LAMPIRAN B ARTIKEL BERITA

Dok	Isi Berita
D1	Selain PNS, Peraih Medali Asian Games Bisa Jadi TNI atau Polri. Beragam bonus telah disiapkan pemerintah bagi para atlet berprestasi yang meraih medali pada Asian Games 2018. Bukan hanya bonus uang, para atlet juga dijanjikan diangkat sebagai pegawai negeri sipil (PNS). Syafruddin, selaku Chef de Mission (CdM) atau Kepala Kontingen Indonesia di Asian Games 2018 mengatakan, dirinya telah menandatangani usulan presiden terkait penerimaan PNS bagi para peraih medali Asian Games ke - 18. Hingga saat ini, kata Syafruddin, pihaknya telah melakukan pendataan kepada sejumlah atlet berpestasi yang ingin menjadi PNS. Syafruddin mengatakan, selain menjadi PNS, para atlet berpestasi tersebut juga mendapat akses utama menjadi anggota TNI dan Polri. Syafruddin yang juga menjabat Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (MenPAN - RB) menjelaskan, jika syarat menjadi TNI dan Polri hanya dua, yakni kesehatan dan fisik. Dirinya pun yakin para atlet peraih medali Asian Games 2018 tersebut telah memenuhi kualifikasi kesehatan dan fisik.
D2	Tak cuma Jadi PNS, Atlet Peraih Medali Bisa Masuk TNI-Polri Tanpa Tes. Keistimewaan didapatkan atlet - atlet Indonesia peraih medali di Asian Games 2018. Selain mendapat jaminan diangkat menjadi Pegawai Negeri Sipil (PNS), mereka juga dapat menjadi anggota Tentara Nasional Indonesia dan Polisi Republik Indonesia tanpa tes apa pun. Hal itu diungkapkan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, Syafruddin Prawiranegara. Ini sudah mendapat persetujuan dari pimpinan TNI maupun Polri. Sehingga, atlet - atlet peraih medali yang ingin masuk kedua instansi tersebut dikatakannya sudah diberikan karpet merah. Alasannya, lanjut dia, persyaratan untuk menjadi TNI maupun Polri hanya membutuhkan dua aspek, yakni kesehatan serta fisik. Dan, atlet peraih medali tentu dikatakannya telah memiliki kedua persyaratan tersebut secara sempurna. Selain itu, dia merincikan, adapun untuk para atlet yang diberikan kewenangan untuk masuk sebagai PNS, TNI, maupun Polri adalah para atlet yang meraih emas, perak, maupun perunggu di ajang Asian Games. Sementara itu, untuk atlet SEA Games hanya peraih medali emas dan perak. Sedangkan untuk atlet yang berlaga di Pekan Olahraga Nasional hanya peraih medali emas.
D3	Atlet berprestasi dapat karpet merah menjadi anggota TNI / Polri. Prestasi yang ditorehkan para atlet Indonesia dalam gelaran Asian Games 2018 menjadi kebanggaan bagi bangsa Indonesia. Sebagai bentuk apresiasi, pemerintah memberikan bonus berupa uang dan jaminan menjadi Pegawai Negeri Sipil (PNS) bagi para atlet yang

Dok	Isi Berita
	mampu meraih medali, baik medali emas, perak, atau juga perunggu. Chief de Mission (CdM) Asian Games 2018, Syafruddin, mengatakan telah menandatangani usulan presiden terkait penerimaan PNS bagi para atlet peraih medali yang berminat. Ia mengatakan, pihaknya tengah melakukan pendataan kepada sejumlah atlet berprestasi yang ingin menjadi PNS. Tak hanya itu, Syafruddin mengatakan para atlet yang berprestasi juga mendapat jaminan bisa masuk menjadi anggota TNI dan Polri. Ia mengatakan hal tersebut telah disetujui oleh Panglima TNI dan juga Kapolri. Lebih lanjut, Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (Menpan RB) itu mengatakan, bagi para atlet yang berminat menjadi anggota TNI dan Polri, hanya ada dua prasyarat yang harus dipenuhi. Yaitu dari sisi kesehatan dan fisik.
D4	Atlet Peraih Medali Dijamin Masuk PNS Maupun TNI - Polri. Bonus bagi para atlet peraih medali di Asian Games 2018 tidak sekadar uang maupun rumah, tapi berupa jaminan menjadi pegawai negeri sipil (PNS). Tak cuma itu, para atlet berprestasi juga memiliki kesempatan berkarier masuk TNI dan Polri tanpa tes. Bagi atlet peraih medali yang ingin berkarier di militer, Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (PANRB) Syafruddin, mengatakan, atlet tersebut langsung diterima tanpa tes. Menurut Syafruddin, staf Kemenpan sudah mendata siapa saja atlet yang ingin menjadi aparatur sipil negara (ASN).
D5	Ini Keistimewaan Peraih Medali: Bisa Masuk TNI - Polri Tanpa Tes. Pemerintah benar - benar memanjakan atlet yang meraih medali di Asian Games 2018. Tidak hanya bonus uang, PNS, atau rumah. Melainkan, setiap atlet yang meraih medali bisa jadi anggota Kepolisian Republik Indonesia dan Tentara Nasional Indonesia (TNI). Demikian diungkapkan Chef de Mission (CdM) Syafruddin terkait apresiasi dari pemerintah. Sosok yang menjabat sebagai Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi itu menyebut Polri - TNI memberi karpet merah bagi peraih medali di Asian Games 2018. Mantan Wakil Kepala Kepolisian Republik Indonesia (Wakapolri) ini mengungkap alasan tersebut. Menurutnya, untuk masuk Polri dan TNI hanya butuh dua aspek. Itu meliputi kesehatan serta fisik. Tentu, setiap atlet yang meraih medali sudah memenuhi persyaratan itu secara sempurna. Menteri kelahiran Makassar ini juga menjelaskan secara rinci karpet merah pada setiap event olahraga. Untuk Asian Games 2018, setiap peraih medali baik emas, perak, dan perunggu bisa masuk PNS, TNI, atau Polri. Sementara, SEA Games hanya untuk peraih emas dan perak. Pun demikian dengan Pekan Olahraga Nasional (PON) yang khusus bagi peraih emas.
D6	Tak Perlu Pintar, Seluruh Atlet Peraih Medali di Asian Games 2018 Bisa Masuk TNI POLRI Tanpa Tes. MENTERI Pemberdayaan Aparatur Negara

Dok	Isi Berita
	dan Reformasi Birokrasi (Menpan - RB) Syafruddin memberikan apresiasi kepada para atlet yang meraih medali di Asian Games 2018 kemarin dengan mempersilakan masuk TNI ataupun Polri tanpa tes. Ia mengatakan telah mendapat mandat langsung dari Panglima TNI dan juga Kapolri perihal apresiasi ini. Bahkan tak hanya dipermudah menjadi anggota, atlet juga mempunyai kesempatan untuk masuk sebagai Pegawai Negeri Sipil (PNS) dibidang apapun. Apabila ada atlet yang mau melanjutkan sekolahnya diperbolehkan untuk mendaftar PNS kapanpun tanpa adanya batas waktu. Diketahui penawaran ini diberikan untuk atlet yang meraih medali emas, medali perak dan medali perunggu di tingkat Asian Games. Sementara di tingkat Sea Games hanya untuk atlet peraih medali emas dan medali perak, sedangkan di tingkat Pekan Olahraga Nasional (PON) hanya untuk peraih medali emas.
D7	Sabet Emas di Nomor Tarung, Atlet Pencak Silat Peluk Jokowi - Prabowo. Presiden Joko Widodo melanjutkan safari menyaksikan atlet - atlet Indonesia di berbagai pertandingan Asian Games 2018. Kali ini, dia menyambangi pertandingan hari terakhir dari cabang pencak silat di Taman Mini Indonesia Indah, Jakarta, Rabu (29/8). Dalam kesempatan itu, Jokowi datang pada pertandingan tarung kedua hari itu, mempertemukan Hanifan Yudani Kusumah dan Thai Linh Nguyen (Vietnam) di kelas C (55 - 60 kg). Dia datang saat kedua pesilat tengah melakoni ronde kedua sekitar pukul 16.45. Sementara itu, di sana pun terdapat Calon Presiden Prabowo Subianto menjabat sebagai ketua Ikatan Pencak Silat Seluruh Indonesia (IPSI). Kehadiran kedua tokoh negara tersebut membuat penonton bersorak. Jokowi, sapaan akrab presiden, langsung menyalami Prabowo. Kemudian, semua penonton menyaksikan momen tersebut, meneriakkan yel - yel Indonesia sampai menggema di dalam arena. Mereka terlihat tegang. Mereka tampak fokus menyaksikan pertandingan. Mereka jarang bertegur sapa. Padahal , Jokowi - Prabodo duduk bersebelahan. Hanifan sendiri menang tipis atas Thai dengan skor 3 - 2. Usai berselebrasi atas kemenangan tersebut, Hanifan berlari ke arah Jokowi dan Prabowo. Dia memeluk bakal calon Presiden RI sekaligus. Tangan kanan merangkul pundak Prabowo, sementara tangan kiri merangkul pundak Jokowi. Atas kemenangan Hanifan, pencak silat Indonesia kembali memenangi emas Asian Games 2018 di nomor tarung.
D8	Jokowi - Ma'ruf Amin Banjir Dukungan di Banten. Usai mengumumkan cawapresnya, Jokowi banjir dukungan di Banten. Terutama, terpilihnya KH.Ma'ruf Amien sebagai pendampingnya. Beberapa deklarasi dukungan yang telah berjalan seperti di Kabupaten Lebak, lalu di Kota Cilegon ada Koordinator Besar Relawan Jokowi Banten, di Kota Serang ada dua kali deklarasi, yakni dari jawara dan ulama, lalu dari Sekretariat

Dok	Isi Berita
	Bersama (Sekber) Relawan Jokowi Banten. Menurut pengampu Pondok Pesantren (Ponpes) Al-Fataniyah itu, dipilihnya Kyai Ma'ruf Amien menjadi pendamping Jokowi merupakan kegembiraan tersendiri bagi warga Banten. Karena, adanya putra asli Bumi Seribu Kyai Sejuta Santri yang juga cicit dari Syaikh Nawawi al-Bantani, yang akan menjadi pemimpin di Indonesia. Koordinator ponpes salafi di Banten itu menegaskan, kalau masih ada yang menggunjing kehadiran Jokowi bersama Kyai Ma'ruf, berarti sedang mengancam keutuhan NKRI. Lembaga survei Charta Politica pernah merilis hasil surveinya pada 06 Juni 2018. Menurut Charta Politica, saat itu Prabowo unggul dengan 28,5 persen dan Jokowi hanya meraup suara 26,9 persen di Bumi Seribu Kyai Sejuta Santri ini. Survei itu dilakukan pada 23 - 29 Mei 2018 melalui wawancara kepada 800 responden di Banten itu, juga dilakukan di Jawa Barat, Jawa Tengah dan Jawa Timur. Survei menggunakan metode multistage random sampling dengan margin of error kurang lebih 3,46 persen untuk Banten, dengan tingkat kepercayaan 95 persen.
D9	Gelombang Dukungan untuk Jokowi - Ma'ruf Amin dari Banten. Dukungan untuk Jokowi - Ma'ruf Amin, terus mengalir dari Banten. Deklarasi dukungan antara lain berasal dari Kabupaten Lebak, Kota Cilegon (dari Koordinator Besar Relawan Jokowi Banten), dan dari Kota Serang (jawara dan ulama, serta Sekretariat Bersama Relawan Jokowi Banten). Menurut pengampu Pondok Pesantren (Ponpes) Al-Fataniyah itu, dipilihnya Ma'ruf Amien menjadi pendamping Jokowi merupakan kegembiraan tersendiri bagi warga Banten. Sebab, Ma'ruf Amin merupakan putra asli Bumi Seribu Kiai Sejuta Santri itu yang juga cicit dari Syaikh Nawawi al-Bantani. Koordinator ponpes salafi di Banten itu menilai, jika masih ada yang menggunjing kehadiran Jokowi bersama Ma'ruf, berarti sedang mengancam keutuhan NKRI. Sementara itu, pada 6 Juni 2018, Charta Politica merilis hasil surveinya. Dalam surveinya, Jokowi meraih suara 26,9 persen di Banten. Sedangkan Prabowo, meraih suara 28,5 persen. Survei yang dilakukan pada 23 - 29 Mei 2018 melalui wawancara kepada 800 responden di Banten itu, juga dilakukan di Jawa Barat, Jawa Tengah dan Jawa Timur. Survei menggunakan metode multistage random sampling dengan margin of error kurang lebih 3,46 persen untuk Banten, dengan tingkat kepercayaan 95 persen.
D10	Prabowo - Sandiaga Sudah Ajukan Bertemu PBNU, tetapi Belum Dijawab. Pasangan bakal calon presiden dan bakal calon wakil presiden, Prabowo Subianto - Sandiaga Uno akan sowan ke sejumlah organisasi massa di Indonesia. Pimpinan Pusat Muhammadiyah menjadi ormas pertama yang dikunjungi. Prabowo - Sandiaga berkunjung ke Kantor PP Muhammadiyah di Menteng, Jakarta Pusat, Senin (13/8/2018) malam ini. Fadli Zon berharap, setelah Muhammadiyah, Prabowo - Sandiaga

Dok	Isi Berita
	bisa berkunjung ke kantor Pengurus Besar Nahdlatul Ulama. Ia memastikan Prabowo - Sandi tetap akan mengunjungi PBNU meski tersiar isu bahwa pimpinan ormas tersebut mendukung Joko Widodo - Ma'ruf Amin. Permintaan untuk berkunjung ke ormas Islam terbesar di Indonesia itu pun sudah diajukan. Adapun dalam kunjungan ke ormas ini, menurut Fadli, pasangan Prabowo - Sandi akan meminta masukan dan wejangan.
D11	6 Pesan PP Muhammadiyah untuk Prabowo - Sandiaga. Bakal Calon Presiden dan Wakil Presiden Prabowo Subianto - Sandiaga Uno bertandang ke Kantor PP Muhammadiyah Menteng, Jakarta Pusat. Ketua Umum PP Muhammadiyah Haedar Nashir pun menyampaikan enam pesan usai dikunjungi oleh Prabowo - Sandiaga. Pesan utama dan yang utama adalah agar kedua calon pasangan presiden dan wakil presiden ini bisa untuk tetap berpegang teguh pada Pancasila sebagai landasan negara. Kedua, Muhammadiyah berharap agar kelak pasangan ini bisa mewujudkan kedaulatan ekonomi, politik, dan budaya negara, dalam bingkai yang lebih berani menolak hegemoni asing dan memutus mata rantai impor. Pesan keempat dan kelima, Muhammadiyah ingin agar bagaimana Indonesia ke depan bisa memiliki SDM unggul. Lewat sisi pendidikan dan kesehatan, dan juga reformasi birokrasi agar bisa dirasakan semua kalangan.
D12	Berbalut bendera merah putih, Jokowi dan Prabowo dipeluk atlet silat Asian Games. Venue Pencak Silat Asian Games di Taman Mini Indonesia Indah (TMII) menjadi saksi sejarah. Gegap gempita suporter Indonesia menyaksikan pemandangan yang langka terjadi. Presiden Joko Widodo dan Prabowo Subianto, sama - sama memeluk atlet pencak silat Indonesia Hanifan Yudani Kusumah. Momentum itu terjadi setelah Yudani memastikan meraih emas di cabang olah raga pencak silat untuk kategori Single Men Class 55kg - 60kg. Setelah ditetapkan sebagai pemenang dalam pertandingan melawan atlet Vietnam, Yudani berlari dengan mengibarkan bendera merah putih. Dia meluapkan kegembiraan dan rasa haru serta kebanggaan. Setelah berlari, Yudani naik ke kursi VVIP. Di sana dia disambut bangga oleh Presiden Joko Widodo, Wakil Presiden Jusuf Kalla, Ketua Umum Pengurus Besar Ikatan Pencak Silat Indonesia (IPSI) Prabowo Subianto, Presiden ke- 5 RI Megawati Soekarnoputri, Menko PMK Puan Maharani, dan chef de mission atau ketua kontingen Indonesia Syafruddin. Di sini pemandangan langka terjadi. Pantauan Liputan6.com, Yudani mendapat ucapan selamat dari Presiden Jokowi. Setelah itu, Yudani memeluk erat Prabowo Subianto. Menyaksikan itu, Jokowi memberi tepuk tangan. Tiba - tiba, Yudani memeluk Presiden Jokowi dan Prabowo Subianto. Pelukan ketiganya diselimuti bendera merah putih yang dibawa Yudani. Ekspresi Jokowi dan Prabowo

Dok	Isi Berita
	menyiratkan keduanya tak menyangka bakal mendapat pelukan hangat dari Yudani. Seisi venus pencak silat mendadak heboh. Apalagi Presiden Jokowi dan Prabowo Subianto bakal bertarung dalam Pemilihan Presiden. Wapres JK, Menko Puan Maharani dan Presiden ke- 5 RI Megawati menyaksikan momen itu. Usai mendapat pelukan itu, Prabowo dan Jokowi saling bertatap dan melempar tawa serta senyum. Setelah itu, Prabowo terlihat berbincang dengan Megawati.
D13	Ini Catatan BMKG Soal Jumlah Gempa Susulan di Lombok. Gempa yang menguncang Lombok hingga kini masih terus dirasakan. Meski tidak sekuat beberapa waktu lalu yang berkekuatan 7,0 SR, namun dari catatan BMKG hingga pagi tadi telah ada 593 gempa susulan. Kepala Bagian Humas BMKG, Hary Tirto Djatmiko, mengatakan update gempa bumi Lombok dari tanggal 5 Agustus hingga tanggal 13 Agustus 2018 pukul 10.00 WITA tercatat sebanyak 593 gempa susulan. Diketahui, gempa berkekuatan 7,0 SR ini terjadi pada hari Minggu 5 Agustus 2018. Dari catatan Badan Nasional Penanganan Bencana (BNPB) total korban tewas akibat gempa mencapai 392 jiwa yang tersebar dibeberapa wilayah, di antaranya Kabupaten Lombok Utara sebanyak 339 orang, Lombok Barat 30 orang, Kota Mataram 9 orang, Lombok Timur 10 orang, Lombok Tengah 2 orang dan Kota Lombok 2 orang.
D14	BMKG Catat Ada 593 Gempa Susulan di Lombok hingga Hari Ini. Gempa susulan terus melanda Lombok, Nusa Tenggara Barat (NTB), sejak gempa 7,0 skala Richter (SR) pada Minggu, 5 Agustus 2018. Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) mencatat hingga pagi ini, sudah terjadi 593 gempa susulan. Berdasarkan data yang diperoleh dari BMKG, dari 593 gempa susulan itu, 24 di antaranya dapat dirasakan masyarkat. Terbaru, gempa 4,3 SR mengguncang Lombok. Gempa itu terjadi pada pukul 08.42 WIB. Getaran gempa juga dirasakan di Mataram. Sekadar diketahui, berdasarkan data yang diperoleh dari BNPB per Minggu (12/8/2018), akibat gempa 7,0 SR yang melanda Lombok, NTB. Selain itu, 1.353 lainnya luka - luka.
D15	Terjadi 593 Gempa Susulan Di Lombok Hingga Hari Ini. Pasca gempa berkekuatan 7,0 SR yang melanda Lombok pekan lalu hingga kini sudah ada ratusan gempa susulan yang terjadi. Berdasarkan data BMKG tercatat ada 593 gempa susulan. Gempa berkekuatan 7,0 SR ini terjadi pada hari Minggu (5/8) pekan lalu. Korban tewas akibat gempa terus bertambah. Badan Nasional Penanganan Bencana (BNPB) mencatat total korban tewas akibat gempa Lombok hingga saat ini sudah 392 orang. Korban tewas itu tersebar di Kabupaten Lombok Utara sebanyak 339 orang, Lombok Barat 30 orang, Kota Mataram 9 orang, Lombok Timur 10 orang, Lombok Tengah 2 orang dan Kota Lombok 2 orang.

Dok	Isi Berita
D16	Sampai Pagi Ini Tercatat 593 Gempa Susulan Guncang Lombok. Sampai pagi ini tercatat ada 593 gempa susulan di Lombok, Nusa Tenggara Barat (NTB) pasca gempa berkekuatan 7.0 SR yang terjadi beberapa waktu lalu. Gempa berkekuatan 7,0 SR ini terjadi pada hari Minggu (05/08/2018) pekan lalu. Korban tewas akibat gempa terus bertambah. Badan Nasional Penanganan Bencana (BNPB) mencatat total korban tewas akibat gempa Lombok hingga saat ini sudah 392 orang. Korban tewas itu tersebar di Kabupaten Lombok Utara sebanyak 339 orang, Lombok Barat 30 orang, Kota Mataram 9 orang, Lombok Timur 10 orang, Lombok Tengah 2 orang dan Kota Lombok 2 orang.
D17	Tercatat 593 Gempa Susulan Hantam Lombok, Duka Bencana Nasional. Indonesia kembali berduka. Pasalnya, baru - baru ini Lombok Utara, Nusa Tenggara Barat (NTB) dan sekitarnya kembali diguncang gempa bahkan sampai berkekuatan 7,0 SR (5/8/2018). Guncangan tersebut terjadi tepat pukul 18.46 WITA dan berpotensi tsunami. Pasca gempa berkekuatan 7,0 SR, hingga tanggal 13 Agustus 2018 tercatat masih terjadi gempa susulan sebanyak 593 gempa susulan. Badan Nasional Penanganan Bencana (BNPB) mencatat korban tewas akibat gempa tersebut hingga saat ini sudah 392 orang. Korban tewas tersebut tersebar di kabupaten Lombok Utara sebanyak 339 orang, Lombok Barat 30 orang, kota Mataram 9 orang, Lombok Timur 10 orang, Kota Lombok 2 orang dan Lombok Tengah 2 orang. Sementara korban luka saat ini mencapai 1.353 orang, dengan rincian 783 orang di antaranya luka berat dan 570 lainnya luka ringan. Akibat gempak yang mengguncang Lombok ini, sebanyak 387.067 warga mengungsi di beberapa tempat pengungsian. Selain itu, dilansir dari detik.com gempa tersebut juga menimbulkan kerusakan fisik di antaranya 67.875 unit rumah rusak, 606 sekolah rusak, 6 jembatan rusak, 3 rumah sakit rusak, 10 puskesmas rusak, 15 masjid rusak, 50 unit mushola rusak, 20 unit perkantoran rusak. Bantuan kepada korban gempa terus diupayakan oleh TNI dan juga tim PDB (Penanganan Darurat Bencana), sehingga bantuan dapat tersalurkan secara merata. Hingga saat ini, penyaluran bantuan hingga daerah sulit masih dilakukan baik melalui jalur darat maupun udara. Presiden Joko Widodo (Jokowi) sudah meninjau langsung kondisi korban dan penanganan bencana di Lombok, Nusa Tenggara Barat. Bahkan, Jokowi juga berkunjung ke rumah Zohri, peraih medali emas olimpiade atletik lari.
D18	Hingga Pagi Ini, Tercatat Ada 593 Gempa Susulan di Lombok. Gempa susulan masih terjadi pasca gempa Lombok berkekuatan 7,0 SR. Hingga pagi ini, tercatat ada 593 gempa susulan. Gempa berkekuatan 7,0 SR ini terjadi pada hari Minggu (5/8) pekan lalu. Korban tewas akibat gempa terus bertambah. Badan Nasional Penanganan Bencana (BNPB) mencatat total korban tewas akibat gempa Lombok hingga saat ini

Dok	Isi Berita
	sudah 392 orang. Korban tewas itu tersebar di Kabupaten Lombok Utara sebanyak 339 orang, Lombok Barat 30 orang, Kota Mataram 9 orang, Lombok Timur 10 orang, Lombok Tengah 2 orang dan Kota Lombok 2 orang. Terkait gempa di Lombok, Kapolri Jenderal Tito Karnavian dijadwalkan akan ke NTB hari ini. Selain Tito, Presiden Joko Widodo (Jokowi) dijadwalkan terbang ke NTB siang ini.
D19	UEFA Hukum Sergio Ramos Dua Laga Karena Sengaja Cari Kartu Kuning. Badan sepak bola Eropa UEFA resmi menghukum kapten Real Madrid, Sergio Ramos larangan bermain dalam dua laga karena ulahnya yang sengaja mencari kartu kuning dalam laga melawan Ajax Amsterdam beberapa waktu lalu. Dalam laga leg pertama babak 16 besar Liga Champions tersebut, Ramos mendapatkan kartu kuning pada menit ke-89, tepat setelah melanggar bomber Ajax, Kasper Dolberg. Kartu kuning ini membuat Ramos absen dalam laga leg kedua karena aturan akumulasi kartu. Tentu saja, tindakannya tersebut bisa membuat keuntungan bagi Real Madrid dan juga Ramos sendiri. Sebab dengan begitu, ia tak perlu khawatir akan absen di leg pertama babak perempat final, karena semua kartu kuning akan diputihkan sejak babak delapan besar. Hukuman UEFA. Badan Kontrol, Etika, dan Disiplin UEFA resmi mengumumkan bahwa mereka telah memberikan hukuman terharap Ramos berupa larangan bermain dalam dua laga selanjutnya. Artinya, Ramos kini tak hanya harus absen di laga leg kedua versus Ajax, tapi juga tak bisa tampil di partai leg pertama perempat final jika Madrid lolos. Kontroversi Ramos. Selepas laga di markas Ajax tersebut, Ramos mengatakan bahwa dirinya akan berbohong jika pelanggaran tersebut tidak dilakukan dengan niatan. UEFA pun langsung melakukan investigasi terhadap kasus ini. Sergio Ramos lalu membuat pembelaan melalui media sosial saat mendengar tudingan soal itu. Ia menyatakan pembelaan diri dengan berkata bahwa dirinya merasa tersakiti setelah mendapatkan tuduhan seperti demikian.
D20	UEFA Selidiki Dugaan Sergio Ramos Sengaja Incar Kartu Kuning. Badan Sepak Bola Eropa (UEFA) sedang menyelidiki dugaan bek Real Madrid, Sergio Ramos, sengaja mengincar kartu kuning. Ramos mendapat kartu kuning saat laga melawan Ajax Amsterdam pada leg pertama babak 16 besar Liga Champions di Johan Cruyff Arena, Rabu (13/2/2019) atau Kamis (14/2/2019) dini hari WIB. Pada laga yang dimenangkan Madrid dengan skor 2 - 1 itu, Ramos mendapat kartu kuning pada menit 89 karena melanggar pemain Ajax, Kasper Dolberg. Adanya kartu kuning membuat Ramos harus absen dalam laga leg kedua di Santiago Bernabeu. Namun ia dipastikan bisa berlaga pada babak perempat final tanpa catatan kartu kuning, tentunya jika Madrid lolos. Usai laga tersebut, Ramos mengaku dirinya sengaja mengincar kartu kuning.

Dok	Isi Berita
	Padahal seorang pemain yang dengan sengaja mengincar kartu kuning dapat dilarang berlaga dalam dua pertandingan. Namun melalui Twitter, Ramos kemudian menolak disebut sengaja mengincar kartu kuning. Pada 2010, Ramos juga pernah terbukti sengaja ingin mendapatkan kartu kuning saat timnya menang 4 - 0 atas Ajax dalam fase grup. Ketika itu, Ramos diinstruksikan pelatihnya saat itu, Jose Mourinho, untuk menerima kartu kuning. Pemain lainnya yang juga diinstruksikan serupa adalah Xabi Alonso. Mourinho kemudian didenda dan dilarang mendampingi tim selama dua pertandingan. Sementara Alonso dan Ramos hanya dikenai denda.
D21	Sengaja Dapat Kartu Kuning, Sergio Ramos Dihukum Lebih Berat. Sergio Ramos mendapat hukuman tambahan karena terbukti sengaja mendapat kartu kuning dalam laga Real Madrid lawan Ajax Amsterdam pada leg pertama babak 16 besar Liga Champions. Ramos yang telah mengoleksi satu kartu kuning memutuskan untuk mendapatkan kartu kuning di laga lawan Ajax setelah ia merasa posisi timnya aman. Ramos lalu melanggar Kasper Dolberg usai timnya unggul 2 - 1 pada menit ke-89. Dengan tambahan satu kartu kuning, maka Ramos bakal absen pada leg kedua di Santiago Bernabeu. Meski harus absen, hal itu juga berarti Ramos kembali bersih dan siap tampil di babak perempat final. Keputusan Ramos mendapat kartu kuning tentu didasarkan pada keyakinan bahwa Real Madrid bisa mengatasi perlawanan Ajax di leg kedua dengan modal kemenangan 2 - 1 pada leg pertama. Namun keputusan Ramos untuk mendapat kartu kuning akhirnya berbuah hukuman lebih berat. UEFA memutuskan sanksi tambahan sehingga Ramos harus absen dalam dua laga Liga Champions. Hukuman itu berarti Ramos juga tak bisa tampil dalam leg pertama perempat final bila Real Madrid lolos ke delapan besar. Ramos sendiri mengatakan dalam sepak bola selalu ada keputusan sulit yang harus diambil. Namun ia menyatakan tidak dengan sengaja melanggar Dolberg.
D22	Sergio Ramos Diduga Sengaja Dapat Kartu Kuning. Kapten Real Madrid Sergio Ramos diduga sengaja mendapat kartu kuning dalam laga timnya ke kandang Ajax, Kamis (14/2). Ramos mendapat kartu kuning dari wasit Damir Skomina pada menit ke- 89 setelah mengasari striker Ajax, Kasper Dolberg. Kartu kuning tersebut membuat Ramos sudah mengoleksi tiga kartu kuning sejak fase grup Liga Champions musim ini. Artinya, Ramos akan absen saat Madrid gantian menjadi tuan rumah buat Ajax, dalam leg kedua 16 Besar di Santiago Bernabeu, Madrid, 6 Maret nanti. Nah, bek tersubur di Real itu disebut - sebut sengaja melanggar Dolberg agar mendapat kartu kuning, dan absen pada laga berikutnya. Kalau dilihat dari tayangan ulang, tekel Ramos terjadi saat Dolberg sudah tak lagi menguasai bola. Pundit BBC Sport, Martin Keown salah satu yang menuding adanya kesengajaan di balik kartu

Dok	Isi Berita
	kuning Ramos itu. Setelah laga, Ramos menyanggah indikasi itu. Seolah kurang puas, Ramos pun kembali menegaskannya dalam kicuan akun Twitternya @SergioRamos.
D23	Ramos Didakwa Sengaja Dapat Kartu Kuning. BEK Real Madrid Sergio Ramos, Selasa (26/2), didakwa oleh UEFA karena dianggap sengaja mendapatkan kartu kuning di laga leg pertama 16 besar Liga Champions melawan Ajax. Pemain Spanyol itu diganjar kartu kuning karena melanggar Kasper Dolberg saat Real Madrid menang 2 - 1 dalam laga leg pertama di Amsterdam pada 13 Februari, dua menit sebelum Marco Asensio mencetak gol kemenangan El Real. Tiga kartu kuning berarti Ramos terkena skorsing satu laga dan Ramos memilih absen di laga leg kedua melawan Ajax ketimbang absen di putaran berikutnya. UEFA mengatakan kasus Ramos akan diputuskan pada Kamis (28/2). UEFA memberi preseden pada musim lalu ketika bek Real Madrid Dani Carvajal diganjar skorsing dua laga karena sengaja mendapatkan kartu kuning di babak penyisihan grup. Sehari setelah laga, Ramos membantah dirinya sengaja mendapatkan kartu kuning, Namun, tidak lama selepas laga, pemain berusia 32 tahun itu tampaknya mengaku sengaja melakukan pelanggaran. Real Madrid akan melanjutkan upaya mereka menjadi juara Liga Champions untuk keempat kalinya secara beruntun dengan laga leg kedua 16 besar melawan Ajax di Santiago Bernabeu pada 5 Maret.
D24	UEFA Kembali Hukum Sergio Ramos. UEFA resmi menghukum kapten Real Madrid, Sergio Ramos larangan bermain dalam dua laga karena ulahnya yang sengaja mencari kartu kuning dalam laga melawan Ajax Amsterdam beberapa waktu lalu. Dalam laga leg pertama babak 16 besar Liga Champions tersebut, Ramos mendapatkan kartu kuning pada menit ke- 89, tepat setelah melanggar bomber Ajax, Kasper Dolberg. Kartu kuning ini membuat Ramos absen dalam laga leg kedua karena aturan akumulasi kartu. Tindakannya tersebut bisa membuat keuntungan bagi Real Madrid dan juga Ramos sendiri. Sebab dengan begitu, ia tak perlu khawatir akan absen di leg pertama babak perempat final, karena semua kartu kuning akan diputihkan sejak babak delapan besar. Badan Kontrol, Etika, dan Disiplin UEFA resmi mengumumkan bahwa mereka telah memberikan hukuman terharap Ramos berupa larangan bermain dalam dua laga selanjutnya. Artinya, Ramos kini tak hanya harus absen di laga leg kedua versus Ajax, tapi juga tak bisa tampil di partai leg pertama perempat final jika Madrid lolos. Selepas laga di markas Ajax tersebut, Ramos mengatakan bahwa dirinya akan berbohong jika pelanggaran tersebut tidak dilakukan dengan niatan. UEFA pun langsung melakukan investigasi terhadap kasus ini. Sergio Ramos lalu membuat pembelaan melalui media sosial saat mendengar tudingan soal itu. Ia menyatakan pembelaan diri

Dok	Isi Berita
	dengan berkata bahwa dirinya merasa tersakiti setelah mendapatkan tuduhan seperti demikian.
D25	UEFA Selidiki Kartu Kuning Sergio Ramos. Di laga Ajax Amsterdam vs Real Madrid, Sergio Ramos dituding sengaja mendapat kartu kuning. UEFA mulai menyelidikinya. Saat Madrid mengalahkan Ajax 2 - 1 dalam leg pertama babak 16 besar Liga Champions di Johann Cruijff ArenA, Kamis (14/2/2019) dini hari WIB, Ramos menelan kartu kuning di menit ke- 89 karena melanggar Kasper Dolberg. Kartu itu diklaim sengaja diincar Ramos. Kenapa Diincar? Akibat kartu itu, Ramos harus absen di leg kedua babak 16 besar melawan Ajax. Dengan begitu, kapten Los Blancos tersebut akan bersih dari akumulasi kartu di perempatfinal, itu pun jika Madrid lolos. Dan tudingan Ramos sengaja mengincar kartu kuning itu diketahui dari wawancaranya usai pertandingan. Ia mengaku kadang harus membuat keputusan seperti itu. Komentar Ramos itu kini diselidiki UEFA. Seperti dilansir Associated Press, UEFA disebut sudah memulai proses penyelidikan. Jika terbukti sengaja melakukannya, hukuman Ramos akan ditambah satu pertandingan. Artinya, selain absen di leg kedua melawan Ajax, ia juga harus absen di leg pertama babak perempatfinal. Ini bukan kali pertama Ramos sengaja mendapat kartu kuning di Liga Champions. Kejadian serupa pernah terjadi di 2010 dan 2013.