**PERANCANGAN SISTEM REKOMENDASI ARTIKEL ONLINE DENGAN *CONTENT BASED FILTERING***

**(Brilio.net)**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**MUHAMMAD IQBAL ROFIKURRAHMAN. NIM. 1941727019**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**JULI 2020**

HALAMAN PENGESAHAN

**PERANCANGAN SISTEM REKOMENDASI ARTIKEL ONLINE DENGAN *CONTENT BASED FILTERING***

**(Brilio.net)**

Disusun oleh:

MUHAMMAD IQBAL ROFIKURRAHMAN. NIM. 1941727019

Laporan Akhir ini telah diuji pada tanggal ... Juni 2020

Disetujui oleh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Penguji I | : | NIP. | ........................... |
|  |  |  |  |  |
| 2. | Penguji II | : | NIP. | ........................... |
|  |  |  |  |  |
| 3. | Pembimbing I | : | Rudy Ariyanto, S.T., M.Cs  NIP. 197903132008121002 | ........................... |
|  |  |  |  |  |
| 4. | Pembimbing II | : | Pak yoppy  NIP. | ........................... |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, S.T., M.Cs. | Imam Fahrur Rozi, S.T., M.T. |
| NIP. 19711110 199903 1 002 | |  | | --- | | NIP. 19840610 200812 1 004 | |

**PERNYATAAN**

Dengan ini saya menyatakan bahwa pada Skripsi ini tidak terdapat karya, baik seluruh maupun sebagian, yang sudah pernah diajukan untuk memperoleh gelar akademik di Perguruan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini serta disebutkan dalam daftar sitasi/pustaka.

|  |  |
| --- | --- |
|  | Malang, … Juni 2020  M IQBAL R. |

**ABSTRAK**

***ABSTRACT***

**KATA PENGANTAR**

Puji Syukur kami panjatkan kehadirat Allah SWT/Tuhan YME atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “PERANCANGAN SISTEM REKOMENDASI ARTIKEL ONLINE DENGAN *CONTENT BASED FILTERING*”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari bahwasannya dengan tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
2. Bapak Imam Fahrur Rozi, ST., MT., selaku ketua program studi Manajemen Informatika
3. \_\_\_\_, selaku dosen pembimbing II
4. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, … Juni 2020

M IQBAL R

**DAFTAR ISI**

Halaman

[**BAB I. PENDAHULUAN** 1](#_Toc40397660)

[**1.1** **Latar Belakang** 1](#_Toc40397661)

[**1.2** **Rumusan Masalah** 2](#_Toc40397662)

[**1.3** **Tujuan** 2](#_Toc40397663)

[**1.4** **Batasan Masalah** 3](#_Toc40397664)

[**1.5** **Sistematika Penulisan** 3](#_Toc40397665)

[BAB II. LANDASAN TEORI 5](#_Toc40397666)

[2.1 State-of-the-Art Penelitian Terdahulu 5](#_Toc40397667)

[2.2 *Text* *Mining* 8](#_Toc40397668)

[2.3 *Content Based Filtering* 8](#_Toc40397669)

[2.4 *Preprocessing* 8](#_Toc40397670)

[2.5 Tokenisasi 8](#_Toc40397671)

[2.6 *Stemming* 9](#_Toc40397672)

[2.7 Penghapusan *Stopword* 9](#_Toc40397673)

[**2.8**  **Pembobotan *Term Frequency - Inverse Document Frequency*** 9](#_Toc40397674)

[**2.9**  ***Dice Similarity*** 10](#_Toc40397675)

[BAB III. METODELOGI PENELITIAN 12](#_Toc40397676)

[3.1 Data 12](#_Toc40397677)

[3.2 Metode Pengambilan Data 13](#_Toc40397678)

[3.3 Metode Pengolahan Data 13](#_Toc40397679)

[3.4 Metode Pengembangan Perangkat Lunak 14](#_Toc40397680)

[3.5 Metode Pengujian Data 15](#_Toc40397681)

[BAB IV. ANALISIS DAN PERANCANGAN SISTEM 16](#_Toc40397682)

[4.1 Analisis Permasalahan 16](#_Toc40397683)

[4.2 Analisis Sistem 16](#_Toc40397684)

[4.2.1 Sistem Rekomendasi Terdahulu 17](#_Toc40397685)

[4.2.2 Sistem Rekomendasi Terbaru 18](#_Toc40397693)

[4.3 Analisis Data Masukan 18](#_Toc40397694)

[4.4 Analisis Algoritma 24](#_Toc40397695)

[4.4.1 Klasifikasi dengan algoritma *Support Vector Machine* 24](#_Toc40397696)

[4.5 Analisis Kebutuhan Non Fungsional 24](#_Toc40397697)

[4.6 Analisis Kebutuhan Fungsional 25](#_Toc40397698)

[4.6.1 *Database* 25](#_Toc40397699)

[4.6.2 *Use* *Case* 26](#_Toc40397700)

[4.6.3 *Activity Diagram* 29](#_Toc40397701)

[4.6.4 *Sequence Diagram* 31](#_Toc40397702)

[BAB V. IMPLEMENTASI DAN PENGUJIAN 36](#_Toc40397703)

[5.1 Implementasi *Database* 36](#_Toc40397704)

[5.2 Implementasi Proses Klasifikasi 36](#_Toc40397705)

[5.2.1 *Load Dataset* 36](#_Toc40397706)

[5.2.2 *Preprocessing* 36](#_Toc40397707)

[5.2.3 TF IDF 36](#_Toc40397708)

[5.2.4 *Training* 36](#_Toc40397709)

[5.2.5 *Testing* 36](#_Toc40397710)

[5.2.6 Pengujian 36](#_Toc40397711)

[5.3 Implementasi *Interface* 36](#_Toc40397712)

[5.4 Pengujian Fungsional 36](#_Toc40397713)

[5.5 Pengujian Akurasi 36](#_Toc40397714)

[BAB VI. HASIL DAN PEMBAHASAN 37](#_Toc40397715)

**DAFTAR GAMBAR**

[Gambar 2. 1 Hyperplane terbaik yang memisahkan kedua kelas -1 dan +1 13](#_Toc39521175)

[Gambar 2. 2 Hyperplane terbentuk di antara kelas -1 dan +1 14](#_Toc39521176)

[Gambar 2. 3 Pemetaan input space berdimensi dua dengan pemetaan ke dimensi tinggi. 15](#_Toc39521177)

[Gambar 3. 1 Diagram Alir Pengolahan Data .........................................................19](#_Toc39521410)

[Gambar 3. 2 Diagram Alur Waterfall 19](#_Toc39521411)

[Gambar 4. 1 Flowchart Sistem Klasifikasi Kalimat Mengandung Kalimat Perundungan Bentuk Tubuh ............................................................................................................................... 22](#_Toc39619424)

[Gambar 4. 2 Sub Proses Sistem 23](#_Toc39619425)

[Gambar 4. 3 Tahap Preprocessing 24](#_Toc39619426)

[Gambar 4. 4 Flowchart Tokenizing 25](#_Toc39619427)

[Gambar 4. 5 Flowchart Filtering 25](#_Toc39619428)

[Gambar 4. 6 Flowchart Stemming 26](#_Toc39619429)

[Gambar 4. 7 Use Case Diagram 32](#_Toc39619430)

[Gambar 4. 8 Activity Diagram 36](#_Toc39619431)

[Gambar 4. 9 Sequence Diagram Load Dataset 37](#_Toc39619432)

[Gambar 4. 10 Sequence Diagram Preprocessing 37](#_Toc39619433)

[Gambar 4. 11 Sequence Diagram TF IDF 37](#_Toc39619434)

[Gambar 4. 12 Sequence Diagram Training Dataset 38](#_Toc39619435)

[Gambar 4. 13 Sequence Diagram Testing Dataset 38](#_Toc39619436)

[Gambar 4. 14 Sequence Diagram Pengujian 39](#_Toc39619437)

**DAFTAR TABEL**

[Tabel 2. 1 State-of-the-Art Penelitian Terdahulu 6](#_Toc39521646)

[Tabel 2. 2 Kernel yang Umum Digunakan 16](#_Toc39521647)

[Tabel 3. 1 Data Tweet ............................................................................................17](#_Toc39521681)

[Tabel 4. 1 Contoh Proses Processing ……………………………………………………….……26](#_Toc39694977)

[Tabel 4. 2 Perbedaan Tweet Sebelum dan Sesudah Preprocessing 27](#_Toc39694978)

[Tabel 4. 3 Contoh tweet untuk perhitungan TFIDF 27](#_Toc39694979)

[Tabel 4. 4 Daftar Kata 28](#_Toc39694980)

[Tabel 4. 5 Term Frequency 28](#_Toc39694981)

[Tabel 4. 6 Document Frequency 28](#_Toc39694982)

[Tabel 4. 7 Invers Document Frequency 29](#_Toc39694983)

[Tabel 4. 8 Perhitungan TF-IDF 29](#_Toc39694984)

[Tabel 4. 9 Kebutuhan Perangkat Lunak 30](#_Toc39694985)

[Tabel 4. 10 Kebutuhan Perangkat Keras 30](#_Toc39694986)

[Tabel 4. 11 Atribut Tabel Dataset 31](#_Toc39694987)

[Tabel 4. 12 Atribut Tabel Terms 31](#_Toc39694988)

[Tabel 4. 13 Atribut Tabel Term\_Frequency 31](#_Toc39694989)

[Tabel 4. 14 Deskripsi Use Case 32](#_Toc39694990)

[Tabel 4. 15 Skenario Use Case Melihat Dataset 33](#_Toc39694991)

[Tabel 4. 16 Skenario Use Case Melakukan Preprocessing 33](#_Toc39694992)

[Tabel 4. 17 Skenario Use Case Melakukan TF-IDF 33](#_Toc39694993)

[Tabel 4. 18 Skenario Use Case Melakukan Training Dataset 34](#_Toc39694994)

[Tabel 4. 19 Skenario Use Case Testing Dataset 34](#_Toc39694995)

[Tabel 4. 20 Skenario Use Case Melakukan Pengujian 34](#_Toc39694996)

**DAFTAR LAMPIRAN**

**BAB I. PENDAHULUAN**

* 1. **Latar Belakang**

Artikel merupakan karangan tertulis yang panjangnya tidak dapat ditentukan, dimana tujuannya untuk menyampaikan gagasan maupun fakta dengan maksud meyakinkan, mendidik, ataupun menghibur [1]. Di era modern, artikel sering ditulis di berbagai media online seperti website. Artikel online dapat dibagi menjadi berbagai kategori sesuai pembahasannya. Dalam sebuah kategori pastinya akan memiliki jumlah artikel yang sangat banyak. Sehingga pengkategorian artikel, tidak dapat digunakan sebagai acuan sarana untuk mempermudah pengguna dalam memilih artikel yang tepat. Sistem rekomendasi hadir sebagai sarana agar pengguna dapat lebih mudah memilih artikel yang relevan dengan artikel lain sesuai yang ingin pengguna akses, tanpa perlu harus melakukan pencarian di kolom pencarian website.

Sistem rekomendasi merupakan sistem yang sering diterapkan oleh *website* penyedia artikel [2], karena sistem rekomendasi dapat ini memberikan dampak yang baik terhadap statistik *website*. Dalam sistem ini tingkat relevansi terhadap artikel yang direkomendasikan harus diperhatikan, agar para pengguna dapat memberikan respon yang baik pada *website* dan artikel yang sedang diakses. Pada sistem rekomendasi terdapat berbagai teknik dalam penerapannya, yaitu terdiri dari *content based filtering, collaborative filtering,* dan *hybrid filtering*.

Brilio.net merupakan sebuah perusahaan media penyedia artikel yang sudah cukup dikenal di Indonesia. *website* ini menyediakan artikel tentang gaya hidup, berita *viral*, teknologi dan artikel-artikel jenaka yang sering diperbincangkan oleh banyak netizen di Indonesia, dan memiliki jumlah pengunjung yang banyak dengan rata-rata sepuluh ribu pengunjung setiap harinya, brilio.net belum menerapkan sistem rekomendasi (*content based filtering*) yang berdasarkan sesuai dengan kemiripan artikel yang sedang diakses oleh pengguna, hal ini dapat mengurangi ketertarikan pengguna untuk membaca artikel lainnya, karena artikel yang direkomendasikan hanya berdasarkan artikel dengan jumlah akses terbanyak saja, sehingga tidak ada peningkatan dalam statistik *website* dibagian lama pengguna mengakses.

Pada penelitian sebelumnya telah dibuat sistem rekomendasi artikel yang serupa dengan menggunakan metode yang berbeda yaitu metode *cosine similarity* dengan hasil menunjukkan bahwa artikel yang direkomendasikan adalah relevan [3], kelemahannya sistem ini beroperasi kurang cepat, dikarenakan alur sistem yang akan melakukan operasi pengecekan kemiripan artikel setiap  pengguna mengakses, sehingga semakin banyak pengguna dan artikel yang diakses, semakin lama juga proses perhitungannya.

Brilio.net membutuhkan sebuah sistem rekomendasi artikel online dengan proses perhitungan yang lebih cepat dan dapat menampilkan artikel yang relevan sesuai dengan artikel yang diakses oleh pengguna. Sebuah artikel perbandingan menyebutkan bahwa metode *dice similarity* tidak membutuhkan komputasi yang terlalu berat [4]. Oleh sebab itu dibuatlah perancangan sistem rekomendasi artikel online dengan *content based filtering*.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil, yaitu :

1. Bagaimana membuat sebuah sistem rekomendasi artikel yang relevan berdasarkan artikel yang sedang diakses oleh pengguna ?
2. Bagaimana mempercepat proses pemberian rekomendasi artikel kepada pengguna, tanpa terpengaruh dari jumlah artikel dan pengguna yang mengakses ?
   1. **Tujuan**

Berdasarkan rumusan masalah yang telah dijelaskan, maka tujuan dari penelitian ini yaitu :

1. Membuat prototipe sistem dalam memberikan rekomendasi artikel yang relevan sesuai dengan artikel yang sedang diakses oleh pengguna.
2. Mempercepat proses perhitungan dalam memberikan rekomendasi artikel tanpa terpengaruh dengan banyaknya artikel yang sedang diakses.
   1. **Batasan Masalah**

Agar skripsi penulis yang berjudul “Perancangan Sistem Rekomendasi Artikel Online dengan *Content Based Filtering*” dapat berjalan sesuai dengan rencana dan tujuan awal, maka penulis memberikan batasan-batasan masalah yaitu:

1. Sistem hanya memberikan rekomendasi artikel dari website Brilio.net saja.
2. Sistem ini memberikan rekomendasi sesuai kemiripan artikel yang sedang diakses, tanpa memperhatikan perilaku pengguna sebelumnya.
3. Artikel yang direkomendasikan adalah artikel yang telah diambil dan disimpan ke penyimpanan lokal sistem sebelumnya.
4. Sistem ini memberikan rekomendasi tidak bergantung kepada kategori artikel.
   1. **Sistematika Penulisan**

Uraian dalam laporan Skripsi penulis menyusun dengan Sistematika penulisan sebagai berikut:

**BAB I : PENDAHULUAN**

Pendahuluan berisikan tentang latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat.

**BAB II : LANDASAN TEORI**

Landasan teori berisikan tentang tinjauan pustaka dari aplikasi yang penulis buat.

**BAB III : METODELOGI PENELITIAN**

Berisi mengenai tahapan yang dilakukan untuk menyelesaikan masalah pada tugas akhir yang bersumber dari proses dalam perencanaa tugas akhir. Metode penelitian berisi urauian tentang metode pengmbilan data, metode pengembangan sistem, fase-fase pengembangan sistem.

**BAB IV : ANALISIS DAN PERANCANGAN**

Analisa dan Perancangan berisikan tentang analisa sistem aplikasi dan perancangannya.

**BAB V : IMPLEMENTASI DAN PENGUJIAN**

Implementasi dan pengujian berisikan penerapan/implementasi dan pengujian dari aplikasi yang telah penulis buat.

**BAB VI : HASIL DAN PEMBAHASAN**

Hasil dan Pembahasan berisikan tentang pembahasan serta analisa dari hasil proses tersebut.

**BAB VII : KESIMPULAN dan SARAN**

Kesimpulan berisikan tentang kesimpulan dari penelitian dan saran.

# BAB II. LANDASAN TEORI

## 2.1 State-of-the-Art Penelitian Terdahulu

Penelitian yang dilakukan oleh (Bening, Herwijayanti, 2017) berupa sebuah sistem klasifikasi berita online dengan menggunakan pembobotan TF.IDF dan cosine similarity, sistem tersebut melakukan klasifikasi berita dari sebuah situs yaitu kompas.com pengelompokan berita dianggap penting untuk kenyamanan user dalam mencari berita sesuai kategori. Hasil dari pengelompokan ini akurat 91.25 % akan lebih akurat apabila data yang digunakan lebih banyak.

Penelitian yang dilakukan oleh (Kusumaning Putri D, 2016) berupa sebuah sistem rekomendasi berbasis konten dengan inferenzy fuzzy mamdani, dengan sistem ini menunjukkan bahwa metode inferenzy fuzzy juga dapat memberikan sebuah rekomendasi artikel kepada user dengan hasil rekomendasi ditampilkan dalam bentuk ranking berdasarkan berita yang tingkat relevansinya lebih tinggi ke yang lebih rendah. Sistem telah berhasil mendapatkan hasil rekomendasi berita dengan nilai rata-rata precision 0,970989, nilai rata-rata recall 0,871204 dan nilai rata-rata f-measure 0,907818.

Penelitian yang dilakukan oleh (Fikri, dan Ahmad Dzul, 2018) berupa perbandingan metode dice similarity dengan cosine similarity menggunakan query expansion pada pencarian ayatul ahkam dalam terjemah Al-quran berbahasa Indonesia. Undergraduate thesis, Universitas Islam Negeri Maulana Malik Ibrahim. Tujuan penelitian ini adalah untuk mengetahui perbandingan akurasi dan relevansi dokumen yang dihasilkan di antara kedua metode tersebut dengan atau tanpa menggunakan query expansion. Nilai yang menjadi ukuran adalah recall, precision dan f-measure. Dengan hasil dice similarity lebih baik.

Penelitian yang dilakukan oleh (Mas'udiah, P.E., Atmaja, M.D. & Mustafa, L.D. 2017) berupa sebuah sistem Information Retrieval Tugas Akhir dan Perhitungan Kemiripan Dokumen Mengacu Pada Abstrak Menggunakan Vector Space Model. Sistem ini merupakan sebuah sistem temu kembali informasi judul tugas akhir dan perhitungan kemiripan dokumen menggunakan vector space model. Sistem secara otomatis akan melakukan indexing secara offline dan temu kembali (retrieval) secara real time. Proses retrieval dimulai dengan mengambil query dari pengguna, menerapkan stop word removal sehingga dihasilkan keyword yang compaq tetapi dapat mewakili query tersebut, kemudian sistem menghitung kemiripan antara keyword dengan daftar dokumen yang diwakili oleh term-term di dalam index. hasil pengujian terlihat ketika kata kunci “android” dimasukkan maka akan tampil empat dokumen yang diurutkan sesuai tingkat kemiripannya, yaitu docId 3 dengan tingkat kemiripan 0.9512, docId 4 dengan tingkat kemiripan 0.5020, docId 2 dengan tingkat kemiripan 0.2671, docId 8 dengan tingkat kemiripan 0.1522.

Penelitian yang dilakukan oleh (Prasetyo, V.R., 2019) berupa sebuah sistem Penentuan Pembimbing Tugas Akhir Mahasiswa Jurusan Teknik Informatika Universitas Surabaya Dengan Metode Dice Coefficient, pada sistem ini berasal dari sebuah masalah dalam proses pemilihan pembimbing dalam mahasiswa untuk menyelesaikan tugas akhir yang takut salah dalam memilih pembimbing, hasil dari penelitian dengan sistem ini dengan metode K-Fold Cross Validation dengan nilai k = 100 untuk 1000 data TA yang ada, didapatkan nilai akurasi sistem sebesar 36,25%. Apabila diambil 100 data TA yang memiliki dosen pembimbing yang ideal, sistem memberikan nilai akurasi yang lebih baik yaitu sebesar 45,5%.

Penelitian yang dilakukan oleh (Yasni, L., Subroto, I.M.I. & Haviana, S.F.C. 2018) berupa sebuah sistem Implementasi Cosine Similarity Matching Dalam Penentuan Dosen Pembimbing Tugas Akhir. Pada penelitian ini dihitung tingkat kesamaan antara judul, topik, dan abstrak tugas akhir mahasiswa dibandingkan dengan data dosen pembimbing berupa keahlian dosen pembimbing, tugas akhir yang pernah dibimbing oleh dosen. Kemudian metode Cosine similarity akan menghitung tingkat kesamaan kedua query tersebut. Nilai kemiripan yang tertinggi akan dimunculkan sebagai dosen pembimbing yang direkomendasikan. Hasil dari penelitian ini sistem rekomendasi dosen pembimbing tugas akhir dengan menerapkan metode Cosine Similarity membantu mahasiswa mendapatkan dosen pembimbing sesuai dengan tugas akhir yang diajukan.

Tabel 2. 1 State-of-the-Art Penelitian Terdahulu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Judul | Penulis  Univ/ Tahun | Permasalahan | Metode | Hasil |
| 1 | Klasifikasi Berita Online dengan menggunakan Pembobotan TF-IDF dan Cosine Similarity | Bening Herwijayanti  Brawijaya/ 2017 | Pengelompokan berita yang belum sesuai dan masih banyak human error saat proses pengelompokan berita | Cosine Similarity | akurat 91.25 % akan lebih akurat apabila data yang digunakan lebih banyak. |
| 2 | Implementasi Fuzzy Mamdani untuk Keperluan Sistem Rekomendasi Berita Berbasis Konten | Diyah Utami Kusumaning Putri  UGM/ 2016 | Penelitian ini menunjukkan bahwa sistem inferensi fuzzy mamdani dapat digunakan untuk memberikan rekomendasi personal kepada user berdasarkan berita yang sedang dibaca user saat ini. | Inferesi Fuzzy | hasil rekomendasi berita dengan nilai rata-rata precision 0,970989, nilai rata-rata recall 0,871204 dan nilai rata-rata f-measure 0,907818. |

## 2.2 *Text* *Mining*

*Text* *mining* merupakan proses analisis dalam data yang berupa teks dimana sumber data didapatkan dari dokumen (Ronen Feldman, 2007). Tujuan text mining adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisis keterhubungan antar dokumen. Text mining sering digunakan untuk beberapa proses seperti pencocokan dokumen.

## 2.3 *Content Based Filtering*

Sistem rekomendasi berbasis konten (*Content-based Recommendation System*) menggunakan ketersediaan konten (sering juga disebut dengan fitur, atribut atau karakteristik) sebuah item sebagai basis dalam pemberian rekomendasi (Ricci, 2011). Sistem ini cocok digunakan untuk membuat rekomendasi yang tidak membutuhkan histori atau aktifitas pengguna sebelumnya.

Perbedaan metode *Content Based Filtering* dengan collaborative filtering adalah persyaratan yang harus dipenuhi yaitu rekaman hasil pengguna selama mengakses sehingga bisa dikelompokkan tema atau judul artikel apa saja yang pengguna sukai. Oleh sebab itu metode collaborative filtering ini tidak dapat diterapkan pada studi kasus ini.

## 2.4 *Preprocessing*

*Preprocessing* merupakan suatu proses yang dilakukan untuk mendapatkan data yang bersih sehingga proses pencarian data menjadi lebih akurat. Perbandingan dokumen atau data tanpa dilakukan *preprocessing* maka tidak dapat dinilai sebagai hasil yang relevan. preprocessing sendiri terdiri dari beberapa tahapan seperti Penghapusan *stopword*, *stemming*, dan tokenisasi*.*

*Preprocessing* dilakukan sebelum proses pembobotan kata, sehingga kata yang akan diolah menjadi sebuah kata baku dan memiliki arti sendiri, tanpa adanya imbuhan atau kata bantu.

## 2.5 Tokenisasi

Tokenisasi adalah proses pemecahan dokumen teks menjadi token-token kata. Tiga metode yang sering digunakan untuk melakukan tokenisasi, yaitu: Unigram (satu kata), Bigram (dua kata), dan Trigram (tiga kata). tokenisasi dilakukan biasanya di awal *preprocessing*.

Contoh terdapat sebuah kalimat “kuliah kerja di polinema” dilakukan tokenisasi dua kata sehingga menjadi :

* kuliah kerja
* di polinema

Pada penelitian ini proses tokenisasi dilakukan dalam bentuk satu kata sehingga dapat meningkatkan tingkat relevansi terhadap hasil perbandingan dokumen.

## 2.6 *Stemming*

*Stemming* memiliki 2 tahapan utama yaitu pengecekan kata dasar dan penghapusan afiks, prefiks, dan sufiks pada suatu kata. dengan proses *stemming* kata dasar dapat diperoleh seperti contoh “meramal” menjadi “ramal”, “penyaringan” menjadi “saring”.

*Stemming* dapat dilakukan dengan menyiapkan perpustakaan kata terlebih dahulu, semakin banyak jumlah kata semakin baik. Hal yang harus diperhatikan adalah bahasa yang akan digunakan dalam teknik *stemming* ini.

## 2.7 Penghapusan *Stopword*

*Stopword* merupakan kata-kata yang terdapat pada suatu kalimat, di mana kata tersebut memiliki pengaruh yang kecil pada proses information retrieval [9]*.* Penghapusan *stopword* ini adalah proses awal dalam *preprocessing.* Kata-kata yang termasuk ke dalam *stopword* adalah kata ganti (saya, kamu,dia) dan kata hubung (di, ke, dari, tetapi).

Contoh terdapat sebuah kalimat “saya kuliah di polinema” maka hasilnya setelah dilakukan penghapusan stopword akan menjadi “kuliah” dan “polinema”. Sama hal nya dengan *stemming*, penghapusan *stopword* juga memerlukan kumpulan kata yang sudah tersimpan sebelumnya.

**2.8 Pembobotan *Term Frequency - Inverse Document Frequency***

*Term Frequency - Inverse Document Frequency* atau TF - IDF adalah suatu metode algoritma yang berguna untuk menghitung bobot setiap kata yang diperoleh dari hasil *preprocessing* sebelumnya. Metode ini akan menghitung nilai Term Frequency (TF) dan Inverse Document Frequency (IDF) pada setiap token (kata) di setiap dokumen dalam korpus. Secara sederhana, metode TF-IDF digunakan untuk mengetahui seberapa sering suatu kata muncul di dalam dokumen.

Hasil perhitungan TF-IDF dapat memberikan sebuah nilai bobot setiap dokumen sesuai dengan kata kunci yang dicari, setelah dilakukan *sorting* dengan bentuk vektor untuk mengetahui dokumen yang paling mirip.

Rumus umum untuk *Term* *Weighting* TF-IDF adalah penggabungan dari formula perhitungan raw TF dengan formula IDF dengan cara mengalikan nilai TF dengan nilai IDF :

Wdt = tfdt \* Idft

Idft = (log(N/df))

(1).

Dimana:

Wdt = bobot dokumen ke­-d terhadap kata ke­-t.

tfdt = banyaknya kata yang dicari pada sebuah dokumen.

N = total dokumen.

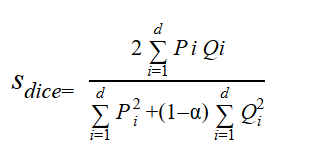
df = banyak dokumen yang mengandung kata yang dicari.

Setelah dilakukan pembobotan maka dokumen akan dihitung tingkat kemiripannya dengan metode *dice similarity.*

**2.9  *Dice Similarity***

*Dice Similarity* adalah metode untuk melihat tingkat kedekatan atau kesamaan (similarity) term dengan cara pembobotan term*.* Dokumen dipandang sebagai sebuah vektor yang memiliki *magnitude* (jarak) dan direction (arah). Kelebihan dari metode ini adalah proses komputasi yang tidak terlalu berat namun akurat dan dapat menentukan prioritas antara precision dan recall berdasarkan kebutuhan melalui nilai pada persamaan (2).

Untuk perhitungan metode *dice similarity* ini dapat dilakukan dengan rumus sebagai berikut :



(2).

Dimana P dan Q adalah dokumen yang berbeda. Pi adalah term i yang ada di dokumen P sedangkan Qi adalah term i yang ada di dokumen Q, dan α adalah nilai konstanta antara 0 dan 1. Apabila nilai α lebih besar dari 0,5 sehingga nilai precision akan lebih tinggi. Sedangkan jika nilai α lebih kecil dari 0,5 maka nilai recall akan lebih tinggi.

Sebagai contoh apabila **P** dan **Q** adalah himpunan set string (bukan vektor), jumlah unigram dari “Photography” and “Photographic” masing-masing adalah **|P| = 9** dan **|Q| = 10** dimana **|P n Q| = 8**, sehingga dapat diperoleh **dice(P,Q) = 0.842** atau persentase similarity dari kedua kata tersebut adalah 84.2%.

# BAB III. METODELOGI PENELITIAN

## 3.1 Data

Data yang akan diolah merupakan data yang berasal dari artikel yang telah dibuat sebelumnya pada situs brilio.net (data uji), pengolahan data berfokus kepada isi artikel seperti judul, gambar, isi konten, tanggal terbit dan alamat URL. Dalam proses perhitungan algoritma, semua data kecuali alamat URL, gambar, dan tanggal terbit akan digabung menjadi satu dokumen untuk dibandingkan dengan dokumen yang telah dibuat sebelumnya.

Akan ada proses pengambilan data secara terjadwal pada sistem untuk memperoleh data baru pada situs brilio.net, data baru yang diperoleh akan di catat jumlahnya dalam sebuah *history.*

Tabel 3. 1 Contoh data hasil crawl

|  |  |  |  |
| --- | --- | --- | --- |
| Thumbnail | Title | Date | Url |
| https://cdn-brilio-net.akamaized.net/news/2020/05/13/184420/gara-gara-corona-mira-lesmana-ungkap-produser-film-rugi-miliaran-200513t.jpg | Mira Lesmana sebut produser film sudah rugi miliaran karena corona | 2020-05-13 | https://brilio.net/selebritis/gara-gara-corona-mira-lesmana-ungkap-produser-film-rugi-miliaran-200513t.html |
| https://cdn-brilio-net.akamaized.net/news/2020/05/13/184399/10-momen-kebersamaan-ranbir-kapoor-alia-bhatt-makin-mesra-200513v.jpg | 10 Momen kebersamaan Ranbir Kapoor & Alia Bhatt, makin mesra | 2020-05-13 | https://brilio.net/selebritis/10-momen-kebersamaan-ranbir-kapoor-alia-bhatt-makin-mesra-200513v.html |

Pada tabel 3.1 terdapat beberapa kolom yang tidak ditampilkan, sebagai contoh adalah kolom *content*, pada kolom content semua *text* akan diambil.

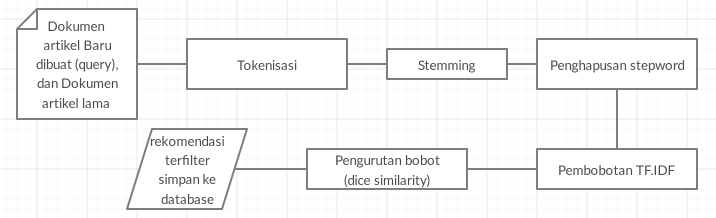
## 3.2 Metode Pengambilan Data

Metode yang digunakan dalam proses pengambilan data adalah metode *crawling,*  dimana metode ini juga dikenal dengan metode *scrapping* yang intinya melakukan ekstraksi sebuah halaman pada situs tertentu untuk memperoleh data yang dibutuhkan, khususnya pada penelitian kali ini adalah situs brilio.net.

Metode ini memiliki tingkat kesulitan tersendiri tergantung situs yang akan dilakukan ekstraksi. Semua data yang diperoleh adalah data legal, dan semua data yang tampil saat membuka situs secara manual adalah data yang dapat dilakukan proses *crawling.*

## 3.3 Metode Pengolahan Data

Metode pengolahan data dimulai saat pertama kali data diperoleh dari proses *crawling*. Data yang akan diperoleh untuk diolah adalah judul artikel dan isi konten, yang telah diformat menjadi satu dokumen tanpa adanya tag html menjadi teks keseluruhan. Sebelumnya akan disediakan beberapa data sebagai data training untuk mengenalkan sistem.



Gambar 3. 1 Alur pengolahan data

Disaat ada artikel baru yang terbaca dari sistem *crawling* maka data tersebut akan dilakukan pemrosesan seperti proses *preprocessing* dan dilanjutkan dengan melakukan perhitungan pembobotan dengan TF.IDF, setelah menentukan bobot dari TF.IDF selanjutnya bobot tersebut akan dihitung nilai kedekatannya dengan metode *dice similarity* dimana metode ini adalah metode yang sering digunakan untuk mencari nilai kedekatan suatu dokumen.

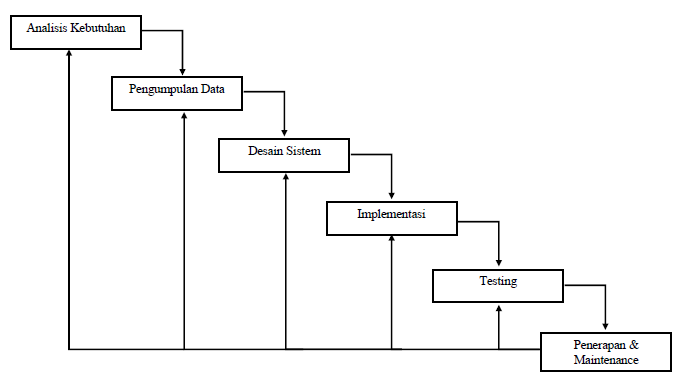
Keunggulan dari *dice similarity* dibandingkan metode lain yaitu dapat menentukan prioritas *precision* dan *recall* sesuai kebutuhan. Maka *dice similarity* menjadi metode yang dirasa tepat untuk menjalankan sebuah sistem rekomendasi artikel berita yang cepat.

Setelah data rekomendasi ditentukan dan difilter sesuai kebutuhan, selanjutnya data akan disimpan kedalam database dengan *id* sesuai dengan *id* artikel yang baru, dimana artikel baru ini digunakan sebagai *query* pembanding dengan artikel lainnya.

Untuk mengatasi keterlambatan pembaharuan data rekomendasi, maka dibuatlah sebuah alur sistem dimana ketika artikel lama di akses kembali oleh pengguna, perhitungan rekomendasi akan dilakukan pembaharuan berdasarkan data artikel terbaru berdasarkan tanggal pembuatan artikel.

## 3.4 Metode Pengembangan Perangkat Lunak

Metode dalam penelitian ini menggunakan metode waterfall. Dalam metode ini terdapat 6 tahap, langkah-langkahnya adalah sebagai berikut:



Gambar 3. 2 Diagram Alur Waterfall

1. *Requirement Analysis*

Pada tahap ini akan dijelaskan tentang semua kebutuhan sistem untuk menjalankan sistem rekomendasi artikel ini, baik kebutuhan fungsional dan kebutuhan non fungsional**.**

1. Pengumpulan Data

Proses pengumpulan data dilakukan setiap hari dan dilakukan oleh sistem, sehingga data yang diolah adalah data dinamis yang sifatnya tidak tetap.

1. *Design*

Desain sistem akan terbagi menjadi beberapa bagian, seperti bagian admin atau sistem kontrol dan bagian rekomendasi atau desain untuk pengguna pada situs brilio.net

1. *Implementation*

Dalam tahap ini dilakukan pembuatan aplikasi. Pembuatan perangkat lunak dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan*.*

1. *Testing*

Pada tahap testing dilakukan uji coba pada aplikasi yang telah dibuat untuk mengetahui apakah perangkat lunak yang dibuat telah sesuai dengan desainnya dan apakah masih terdapat kesalahan

1. *Maintenance*

Tahap ini dilakukan jika perangkat lunak sudah jadi danmengalami kerusakan. Maka akandilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

## 3.5 Metode Pengujian Data

Pengujian yang dilakukan untuk menguji sistem dilakukan dengan 2 langkah, yaitu pengujian fungsionalitas dan pengujian performa sistem. Pengujian bertujuan untuk memastikan bahwa sistem yang dibangun telah berjalan sesuai dengan yang diharapkan.

1. Pengujian Fungsionalitas

Uji coba fungsionalitas dikatakan berhasil apabila fungsi yang ada pada sistem sesuai dengan yang diharapkan. Uji fungsionalitas yang diterapkan pada penelitian ini ialah dengan mencoba menjalankan setiap fitur yang ada dan yang disediakan pada sistem ini

1. Pengujian Akurasi Sistem

Dalam pengujian ini, dilakukan untuk mengetahui keakuratan aplikasi dalam menampilkan data hasil pencarian oleh pengguna.

# BAB IV. ANALISIS DAN PERANCANGAN SISTEM

## 4.1 Analisis Permasalahan

Sebuah situs besar penyedia artikel akan terus menerbitkan artikel setiap hari nya, jumlah artikel yang terbit tidak dapat diperkirakan. Hasil rata-rata jumlah artikel harian pada situs brilio.net bisa mencapai 20 - 30 artikel perharinya. Tema atau kategori artikel yang di unggah tidak selalu sama. Penerapan sistem rekomendasi artikel merupakah suatu hal yang sangat penting agar para pengguna bisa terus mengakses dalam waktu yang lama.

Permasalahan yang terjadi adalah situs brilio.net saat ini menerapkan rekomendasi artikel manual berdasarkan editor atau penulis artikel, dimana artikel yang direkomendasikan belum tentu berkaitan satu sama lain. Dalam hal ini pembuatan sistem rekomendasi artikel dibutuhkan, sehingga sistem dapat memberikan rekomendasi artikel secara otomatis berdasarkan tingkat kemiripan artikel yang telah dibuat.

Dengan adanya sistem rekomendasi otomatis daftar rekomendasi sistem yang telah dibuat sebelumnya akan terus diperbaharui mengikuti hasil perhitungan tingkat kemiripan artikel yang baru. Pada sistem ini jumlah rekomendasi dibatasi menjadi 5 artikel terbaik, dan perubahan data rekomendasi akan dilakukan pada artikel yang telah diunggah 30 hari terakhir.

## 4.2 Analisis Sistem

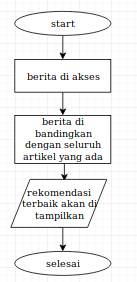
Sistem rekomendasi artikel yang telah dibuat merupakan sebuah sistem yang terpisah dari situs asli yaitu brilio.net (sistem utama). sehingga dibutuhkan sebuah komunikasi antar sistem agar pengolahan dan pemindahan data dapat berjalan dengan baik, dalam praktik nya sistem rekomendasi terpisah ini memiliki beberapa keuntungan seperti berikut:

1. Perubahan atau perbaikan sistem dapat lebih mudah
2. Saat sistem error sistem utama tidak memperoleh dampak yang siginfikan
3. Proses sistem tidak membrikan beban pada sistem utama

Gambaran sistem secara teknis dapat digolongkan menjadi *server* dan *client* sistem. Dimana situs utama akan menjadi bagian *client* yang akan melakukan request kepada *server*, begitu juga sebaliknya *server* akan memberikan respon terharap request yang ada.

### Sistem Rekomendasi Terdahulu

Pada penelitan sebelumnya telah dibuat beberapa macam sistem rekomendasi artikel yang memiliki tujuan yang sama yaitu untuk memberikan rekomendasi artikel sesuai dengan artikel yang sedang diakses, berikut adalah alur sistem penelitian terdahulu



Gambar 4. 1 alur sistem terdahulu

Pada sistem rekomendasi sebelumnya, proses perhitungan dilakukan ketika berita saat itu diakses oleh pengguna, ketika seorang pengguna mengakses satu artikel maka prosesnya akan cepat, namun hal ini berbeda ketika ribuan pengguna mengakses ribuan artikel yang berbeda. maka dibutuhkan sebuah sistem yang baru dengan alur sistem yang berbeda sehingga dapat mempercepat proses pemberian rekomendasi tanpa terpengaruh oleh banyaknya pengguna dan artikel.

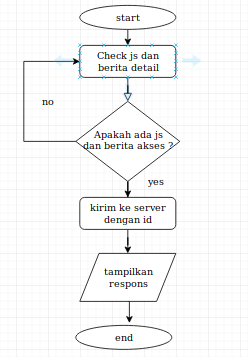
Kelebihan dari sistem yang terdahulu semua rekomendasi yang diberikan adalah berasal dari semua artikel yang terbaru, namun hal ini dapat membuat proses sistem menjadi sangat lama terutama dalam proses perhitungan.



### Sistem Rekomendasi Terbaru

Sistem rekomendasi terbaru menerapkan sebuah alur yang melakukan sebuah perhitungan atau kalkulasi kemiripan terlebih dahulu pada saat pembuatan artikel, sehingga saat artikel yang dibuat dan diakses oleh pengguna sistem tidak memerlukan lagi proses perhitungan kemiripan ulang. Hal ini dapat meringankan dan mempercepat kinerja sistem.

Berikut adalah alur sistem yang terletak pada bagian situs utama atau *client* sistem:

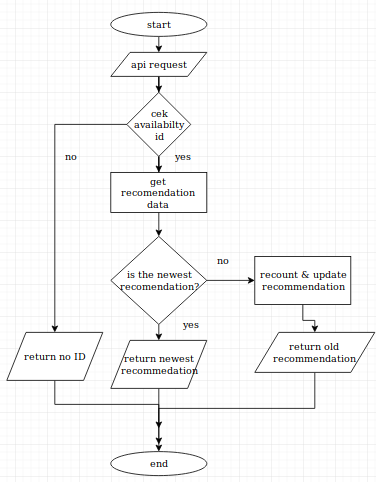


Gambar 4. 2 Alur sistem *client*

Pada gambar 4.2 merupakan alur sistem disaat pengguna mulai mengakses berita pada situs brilio.net. Sistem akan melakukan pengecekan apakah ada file javascript yang telah disediakan untuk melakukan pengiriman data kepada server, jika ada, maka id artikel akan dikirimkan dan menunggu server akan memberikan sebuah jawaban berupa data rekomendasi artikel yang telah dilakukan perhitungan sebelumnya. Sehingga akan ditampilkan ke dalam sebuah tag atau elemen yang sudah disediakan di halaman brilio.net.

Dalam pemasangan kode javascript ini, pihak brilio dapat menentukan sesuai parameter jenis rekomendasi apa yang akan ditampilkan, bisa berupa gambar atau hanya list saja. namun hal ini tidak berpengaruh terhadap proses perhitungan rekomendasi, hanya tipe respon dari server saja yang berbeda.

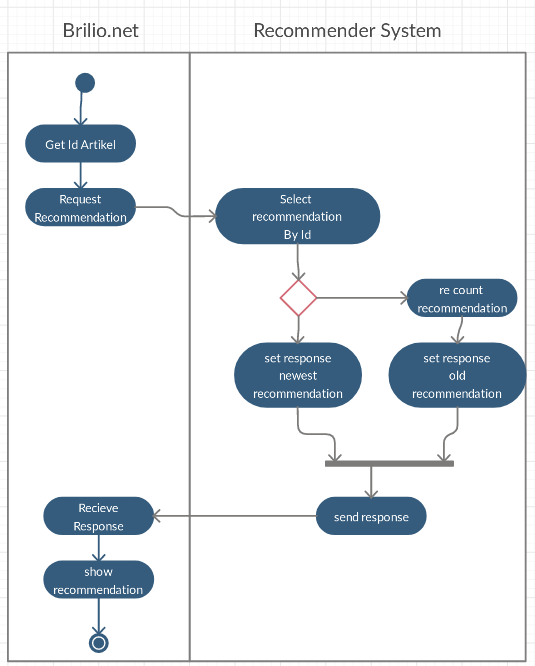
Berikut adalah alur sistem yang terletak pada bagian *server*:



Gambar 4. 3 Alur sistem *client*

Pada gambar 4.3 dijelaskan alur tentang proses sistem memberikan rekomendasi kepada situs brilio, sistem akan berjalan otomatis ketika ada request dari pihak brilio berupa ID artikel yang sedang pengguna akses. Dijelaskan bahwa sistem melakukan pengecekan ketersediaan ID yang diminta, setelah sukses maka sistem akan memperoleh data rekomendasi yang telah dihitung sebelumnya, dan dilakukan perbandingan apakah data ini merupakan data rekomendasi lama atau tidak. sehingga jika data termasuk data lama maka sistem akan melakukan perhitungan ulang dengan artikel yang lebih terbaru.

Berikut adalah *activity diagram* dari dua entitas sebagai client dan server:



Gambar 4. 4 activity diagram client server

Entitas brilio.net selaku client yang akan merequest setiap ada pengguna yang mengakses artikel, lalu *recommender system* selaku server yang akan menyediakan list rekomendasi yang sudah dihitung sebelumnya saat pertama kali artikel dimasukkan ke dalam sistem.

## 4.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis untuk mengetahui spesifikasi kebutuhan sistem yang dibangun. Spesifikasi kebutuhan,yaitu analisis perangkat lunak dan perangkat keras.

1. Analisis Kebutuhan Perangkat Lunak

Tabel 4. 1 Kebutuhan perangkat lunak

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat** | **Keterangan** |
| 1. | *Web Browser* | *Google Chrome / Firefox* |
| 2. | *Web Server* | *Apache / Xampp* |
| 3. | *DBMS* | *PhpMyAdmin* |
| 4. | *Text Editor* | *Visual Studio Code* |
| 5. | *Framework* | *Python & laravel* |

1. Analisis Kebutuhan Perangkat Keras

Tabel 4. 2 Kebutuhan perangkat keras

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat** | **Keterangan** |
| 1. | *Processor* | *Intel Core i7* |
| 2. | *Memory* | *8GB* |
| 3. | *Hardisk* | *100GB* |

## 4.4 Analisis Kebutuhan Fungsional

Analisa kebutuhan fungsional berisikan proses-proses yang dilakukan oleh sistem. Dalam hal ini user sebagai pengguna dapat menggunakan layanan-layanan pada sistem, antara lain :

Sistem dapat mengumpulkan data artikel terbaru dan menyimpannya kedalam database

Sistem dapat melakukan *preprocessing* data yang disimpan didalam database

Sistem dapat melakukan proses pembobotan

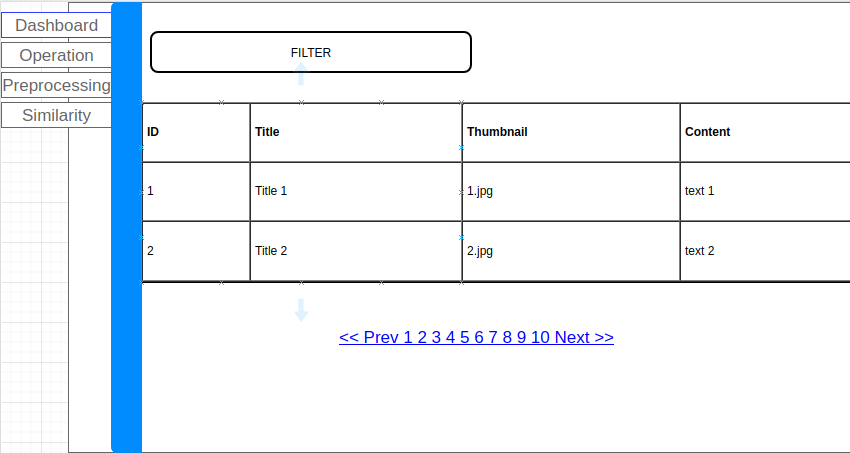
Sistem dapat melakukan perhituangan *similarity*

Sistem dapat melakukan pengurutan artikel dengan kemiripan tertinggi dan menampilkan nya saat artikel diakses

## 4.5 Perancangan Antarmuka

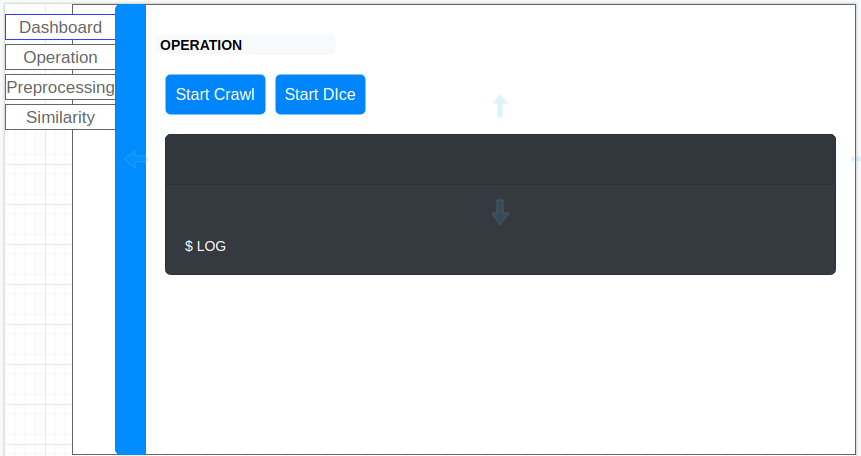
Perancangan antarmuka sistem rekomendasi terdiri dari beberapa menu yang memiliki tujuan tersendiri.

Bagian pertama adalah halaman *dashboard*



Gambar 4. 5 Halaman *dashboard*

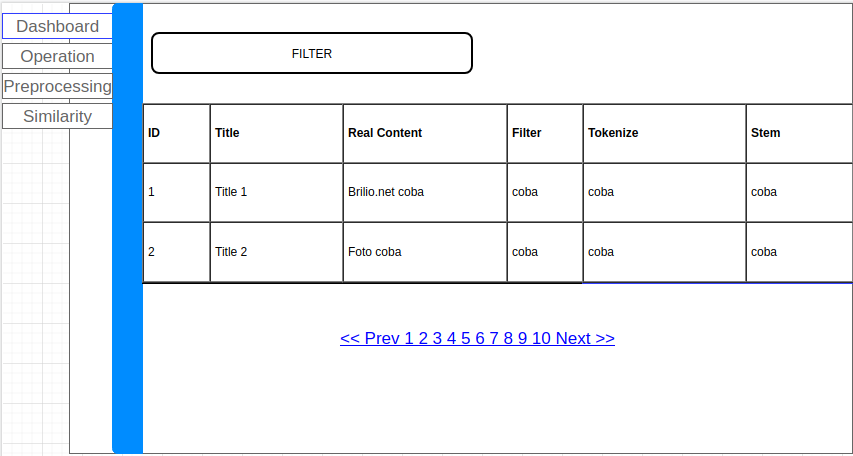
Pada halaman *dashboard* akan disediakan hasil *crawling* artikel dari situs brilio.net namun hanya di tampilkan kolom yang dibutuhkan saja. Akan ada fungsi untuk filter pada halaman ini.



Gambar 4. 6 Halaman *operation*

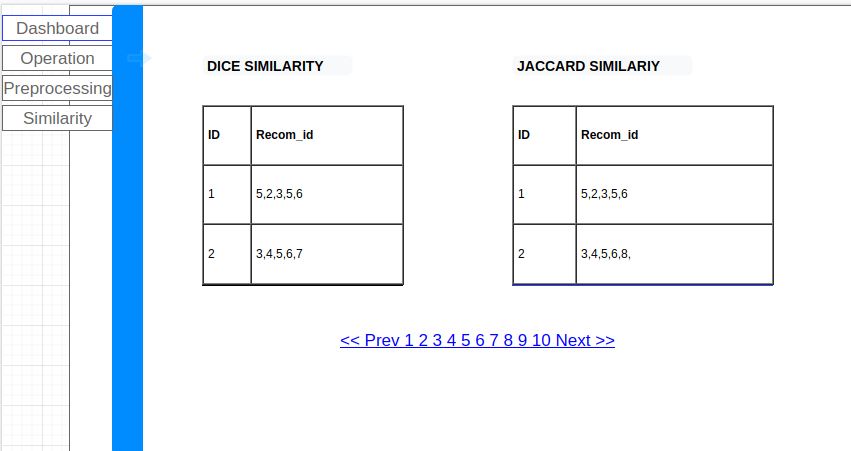
Pada halaman *operation* akan ada dua aksi untuk memulai proses *crawling* dan perhitungan *similarity*. hasil operasi akan ditampilkan pada bagian log. Log yang ditampilkan merupakan hasil *output* dari pemrograman *python* yang telah disiapkan dan di panggil sebelumnya.

Pada bagian selanjutnya adalah halaman *preprocessing*



Gambar 4. 7 Halaman *preprocessing*

Pada halaman *preprocessing* akan ada tabel untuk menampilkan hasil pengolahan kata dari artikel yang telah diperoleh sebebelumnya. Artikel tersebut akan diolah agar proses perhitungan dapat lebih maksimal. Pengolahan berupa *filtering* yang berguna untuk menhilangkan karakter yang tidak dibutuhkan dan mengubah semua kata menjadi huruf kecil, pengolahan selanjutnya adalah tokenisasi yang berguna untuk menghapus beberapa kata yang tidak terpakai, dan yang terakhir adalah proses pengolahan *stemming* yang berguna untuk mengubah kata kerja menjadi kata baku.



Gambar 4. 8 Halaman *similarity*

Pada halaman *similarity* akan menampilkan hasil rangking artikel dari proses perhitungan.

## 4.6 Perancangan Basisdata

Perancangan basisdata pada sistem rekomendasi ini terdiri dari beberapa tabel dan menggunakan Mysql sebagai *database management system*.

Tabel 4. 3 Tabelartikel

|  |  |  |
| --- | --- | --- |
| **No** | **Atribut** | **Tipe Data** |
| 1 | Id | Int(11) PK |
| 2 | Thumbnail | Text |
| 3 | Title | Varchar(500) |
| 4 | Real\_content | Text |
| 5 | Date | Date |
| 6 | Url | Varchar(225) |
| 7 | Filter | Text |
| 8 | Tokenize | Text |
| 9 | Term | Text |
| 10 | Content | Text |

Pada tabel artikel memiliki sepuluh kolom yang berguna menyimpan data artikel yang diperoleh hasil dari *crawling* dan hasil *preprocessing* sebelumnya.

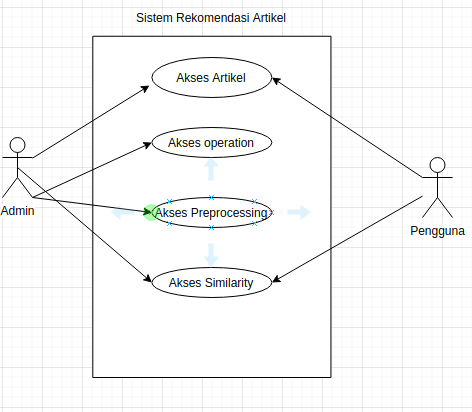
Tabel 4. 4 Tabel similairty

|  |  |  |
| --- | --- | --- |
| **No** | **Atribut** | **Tipe Data** |
| 1 | Article\_id | Int(11) FK id |
| 2 | Recommendation\_id | Varchar(50) |
| 3 | Similarity | Varchar(100) |
| 4 | Percent | Varchar(100) |

Pada tabel selanjutnya akan ada dua tabel yang mirip yaitu tabel *similarity* dan tabel *similarity\_other*, dua tabel ini berguna untuk menyimpan hasil perhitungan dan perangkingan artikel yang telah diproses sebelumnya. Perbedaan dari dua tabel ini adalah metode perhitungan yang digunakan, yaitu menggunakan metode dice dan jaccard *similarity.*

## 4.7 *Use Case*

*Use case* merupakan diagram yang terdiri dari aktor dan proses yang dapat dilakukan oleh aktor tersebut. *Use case* dibuat berdasarkan layanan-layanan yang dimiliki oleh sistem. Berikut adalah *Use case* pada sistem rekomendasi ini.



Gambar 4. 9 Use Case Sistem

Deskripsi *use case*

Tabel 4. 5 Deskripsi use case

|  |  |  |
| --- | --- | --- |
| **No** | **Proses** | **Deskripsi** |
| 1 | Akses Artikel | Dua aktor dapat mengakses proses ini, namun perebedaanya adalah aktor pengguna hanya mengakses artikel melalui portal web utama atau halaman homepage. |
| 2 | Akses Operation | Proses ini berisi tentang bagaimana artikel akan di crawling dan diranking sesuai hasil perhitungan |
| 3 | Akses *preprocessing* | Proses ini berisi tentang hasil *preprocessing* yang telah dioperasikan sebelumnya. |
| 4 | Akses *Similarity* | Proses ini juga dapat diakses dua aktor namun perbedaanya aktor pengguna hanya mengakses hasil berupa artikel lengkap rekomendasi saja. |

Aktor penguna tidak dapat mengakses proses keseluruhan pada halaman admin.

# BAB V. IMPLEMENTASI DAN PENGUJIAN

## 5.1 Implementasi *Database*

Implementasi *database* dengan nama perundungan yang memiliki tiga tabel, yaitu tabel dataset, term dan term\_frequency sesuai dengan analisis dan perancangan sebagai berikut**.**

Gambar 5. 1 Tabel Dataset

## 5.2 Implementasi Proses Klasifikasi

Implementasi proses klasifikasi berdasarkan pada analisis dan perancangan terdiri dari beberapa proses sebagai berikut.

## 5.2.1 *Load Dataset*

### 5.2.2 *Preprocessing*

### 5.2.3 TF IDF

### 5.2.4 *Training*

### 5.2.5 *Testing*

### 5.2.6 Pengujian

## 5.3 Implementasi *Interface*

## 5.4 Pengujian Fungsional

## 5.5 Pengujian Akurasi

# BAB VI. HASIL DAN PEMBAHASAN