BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi kini tengah berkembang pesat. Berbagai informasi dapat diperoleh dengan mudah menggunakan internet dan mesin pencari seperti *Google, Bing, Yahoo* dan lain sebagainya. Kita hanya perlu memasukkan kata kunci pada laman yang telah disediakan *browser* untuk mendapatkan berbagai informasi yang kita inginkan. Termasuk informasi lowongan kerja, berita harian, politik dan sebagainya. Mesin pencari tersebut dalam prosesnya mengadopsi sebuah disiplin ilmu yang dinamakan dengan *Information Retrieval* (IR) atau Temu Kembali Informasi.

IR menurut Salton[1] adalah suatu sistem yang dapat menemukan kembali (*retrieve*) informasi yang sesuai dengan kebutuhan *user* dari sekumpulan data secara otomatis. Salah satu metode klasik yang populer dan telah teruji dalam IR adalah *Vector Space Model* (VSM). VSM merepresentasikan kata kunci dengan masing-masing dokumen ke dalam sebuah vektor[2]. Kemudian nilai kemiripan dari kata kunci dan dokumen didapat dari besar sudut yang di bentuk oleh kedua vektor tersebut.

Pada penelitian klasifikasi berita online[3] yang dilakukan oleh Bening dkk (2018), pengujian untuk metode VSM berhasil mencapai tingkat akurasi yang tinggi, yaitu 91,25% pada empat kali pengujian.

Gambar 1.1 Pengujian VSM pada Klasifikasi Berita Online[3]

Pengujian tersebut dilakukan dengan membagi objek penelitian menjadi data latih dan data training, dengan rincian seperti pada Tabel 1.1:

Tabel 1.1 Pengujian VSM Klasifikasi Berita Online[3]

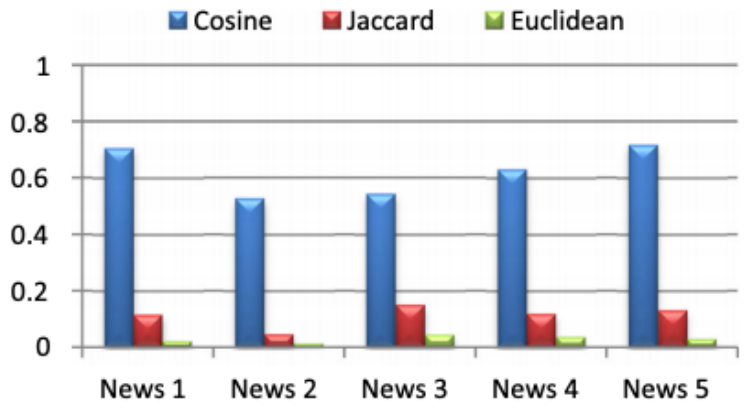
Pengujian	Data Latih	Data Uji	Akurasi
1	90%	10%	80%
2	80%	20%	90%
3	70%	30%	100%
4	60%	40%	95%
Rata-rata			91,25%

Penelitian lain yang menunjukkan keberhasilan VSM dalam menemukan kembali informasi terdapat pada jurnal[4], oleh Andri dkk (2018). Metode ini diimplementasikan pada sistem pencarian data mahasiswa FTI Perbanas Institut Jakarta dan memberikan hasil pencarian yang akurat.

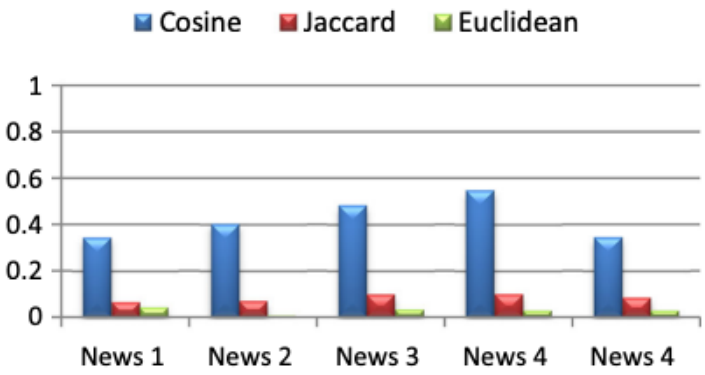
Untuk mendapatkan nilai kemiripan dari dua buah vektor dalam VSM, kita perlu menghitung bobot dari setiap vektor terlebih dahulu dengan menggunakan skema *Term Weighting* (pembobotan kata). Adapun metode-metode yang digunakan dalam pembobotan kata diantaranya *Simple* TF-IDF, *Incremental* TF-IDF, *Phrase Augmented* TF-IDF, *Latent Sematic Index* (LSI) dan *Document to Vector* (D2V). Perbandingan dari metode-metode tersebut terdapat dalam studi komparasi yang dilakukan oleh Omid dkk (2019)[5] untuk kasus *Text* Similarity terhadap data hak paten yang di kelola *United State Patent and Trade Mark Office*

(USPTO). Untuk konteks penelitian tersebut diketahui bahwa *Simple* TF-IDF merupakan metode yang lebih cocok karena objek yang digunakan tidak membutuhkan proses NLP yang rumit.

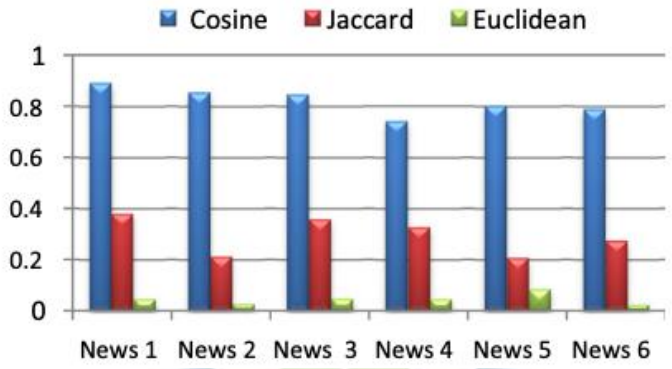
Selanjutnya pada algoritma TF-IDF[3], nilai bobot untuk setiap vektor didapat dengan menggabungkan frekuensi kemunculan sebuah kata di dalam sebuah dokumen dengan frekuensi dari *inverse* dokumen (IDF) yang mengandung kata tersebut. Setelah itu kita dapat melakukan perangkingan dokumen-dokumen mana yang memiliki tingkat kemiripan terbesar dengan kata kunci menggunakan perhitungan similaritas teks dengan pendekatan *Cosine Similarity*, *Jaccard Similarity* atau *Euclidean Distance*. Perbandingan untuk ketiga pendekatan ini dilakukan oleh Ritika dan Satwinder (2020)[6] pada identifikasi item berita teratas di situs berita dan mengukur kesamaan antara dua item berita yang sama dalam dua bahasa yang berbeda berdasarkan event yang sama. Dari hasil penelitian ini, di dapat bahwa *Cosine Similarity* menempati nilai akurasi tertinggi dibandingkan pendekatan lainnya, hal tersebut ditunjukkan oleh Gambar 1.2 sampai Gambar 1.4.



Gambar 1.2 Comparison of similarity coefficients for different news[6]



Gambar 1.3 Comparison of similarity coefficients for completely dissimilar news



Gambar 1.4 Comparison of similarity coefficients for articles of same

Django adalah sebuah web *framework* dengan bahasa pemrograman Python yang kaya akan *library*[7]. *Library* tersebut dapat digunakan untuk mempermudah implementasi *Vector Space Model*. Seperti *stopword removal*, *sastrawi*, *numpy*, dan *punkt* yang digunakan untuk *preprocessing* (proses untuk menghilangkan kata tidak bermakna serta tanda baca) sebelum masuk ke perhitungan TF-IDF. Selain itu, Django juga sudah mendukung Teknik ORM (*Object Relational Mapping*) yang dapat memudahkan kita dalam mengolah *query database*.

Permasalahan yang ditemukan dalam beberapa penelitian sebelumnya, menjelaskan bahwa metode VSM cenderung memakan waktu yang kurang cepat untuk proses pencarian dengan data yang banyak. Pada jurnal[8] rata-rata pencarian memakan waktu 2,24 detik dengan persentase keberhasilan sebesar 81% pada 75

data hadis. Juga dalam jurnal[9] yang memakan waktu 6,042 detik bahkan setelah dilakukan penambahan algoritma *hamming distance* untuk mempercepat pencariannya.

Hal ini disebabkan oleh proses VSM dalam memberikan bobot untuk setiap kata pada semua dokumen, meskipun dokumen tersebut tidak mengandung kata dari *keyword* yang dicari. Sehingga semakin banyak dokumen yang digunakan untuk pencarian, maka akan semakin lama pula proses pembobotannya. Sedangkan dari penelitian-penelitian yang telah dilakukan, data yang dihasilkan dari pencarian dengan metode VSM ini dipastikan mengandung minimal satu kata dari *keyword* yang di ingin cari. Oleh karena itu, peneliti mencoba untuk melakukan filterisasi dokumen terlebih dahulu sebelum proses pembobotan.

Linear Search atau juga biasa dikenal dengan *Sequential Search* merupakan algoritma pencarian yang terbilang cukup sederhana dan mudah. Caranya dengan membandingkan setiap elemen secara beruntun, mulai dari elemen pertama sampai elemen yang dicari ditemukan atau seluruh elemen sudah diperiksa[10]. Berbeda dengan *Binary Search* yang hanya dapat bekerja jika data sudah diurutkan artinya terdapat proses *sorting* terlebih dahulu sebelum pencarian dilakukan.

Dengan menggunakan *Linear Search* untuk proses filterisasi, maka dapat dipastikan dokumen yang akan dilakukan pembobotan mengandung minimal satu kata dari *keyword* yang dicari. Selain itu *Linear Search* juga tidak membutuhkan pengurutan terlebih dahulu sehingga dapat menghemat waktu pada proses pencarian. Begitupun dengan ORM, yang memudahkan kita untuk membuat *query*

pencarian pada proses filterisasi yang akan digunakan sebagai pembanding filterisasi dengan *Linear Search*.

Adapun objek penelitian yang digunakan adalah data hadis yang berasal dari Ensiklopedi Hadis Kitab 9 Imam pada aplikasi LIDWA Pusaka. Aplikasi ini memuat kurang lebih 62.000 data hadis yang dapat dijadikan sebagai media pembelajaran atau objek penelitian[11].

Hadis merupakan sumber hukum kedua bagi umat islam setelah Al-Quran. Penggunaan data hadis sebagai objek penelitian ini ditujukan untuk memudahkan pengguna yang ingin melakukan pencarian hadis pada kitab 9 imam dengan menerapkan konsep *Vector Space Model* (VSM). Seperti yang telah dijelaskan diatas bahwa kelebihan VSM adalah mampu mengembalikan informasi yang paling relevan dengan kata kunci yang dicari, dan juga *keyword* yang digunakan tidak hanya berupa kata namun juga dapat berupa frasa ataupun kalimat. Sementara itu kebanyakan pencarian data hadis saat ini hanya dapat berupa kata atau pencarian dengan konsep *string matching*, sehingga dapat menyulitkan pengguna mencari hadis yang dimaksud.

Berdasarkan latar belakang di atas, penelitian ini berjudul: **"ANALISIS *VECTOR SPACE MODEL* (VSM) TF-IDF DENGAN *LINEAR SEARCH* DAN ORM DJANGO PADA PENCARIAN DATA HADIS".**

1.2 Perumusan Masalah

Berangkat dari permasalahan di atas, maka di dapatlah rumusan masalah sebagai berikut :

1. Bagaimana menerapkan *Vector Space Model* (VSM) TF-IDF dengan *Linear Search* dan ORM Django pada pencarian data hadis?
2. Bagaimana kinerja *Vector Space Model* (VSM) TF-IDF dengan *Linear Search* dan ORM Django pada pencarian data hadis?

1.3 Tujuan Tugas Akhir

Adapun tujuan tugas akhir pada proposal ini diantaranya:

1. Menerapkan *Vector Space Model* (VSM) TF-IDF dengan *Linear Search* dan ORM Django pada pencarian data hadis.
2. Mengetahui kinerja *Vector Space Model* (VSM) TF-IDF dengan *Linear Search* dan ORM Django pada pencarian data hadis.

1.4 Batasan Masalah

Mengingat luasnya pembahasan dan perkembangan yang dapat ditemukan dalam permasalahan di atas, maka perlu adanya batasan-batasan masalah mengenai apa yang akan dibuat dan diselesaikan dalam penelitian ini. Batasan-batasan tersebut yaitu:

1. Data yang digunakan pada aplikasi ini merupakan data hadis dari Ensiklopedi Hadis Kitab 9 Imam (LIDWA Pusaka).
2. Kata kunci untuk pencarian data menggunakan bahasa Indonesia.
3. Fitur yang akan dikembangkan hanya pencarian dan detail hadis serta menampilkan *history* pencarian yang akan digunakan untuk analisis perbandingan pencarian dari metode yang dipakai.
4. Parameter yang digunakan untuk pencarian adalah isi hadis dalam bahasa Indonesia.

5. Parameter pengujian berupa waktu pencarian, tingkat relevansi (*recall*) serta kinerja algoritma (MAP) antara VSM dengan filter terhadap VSM *only*.

1.5 Metodologi Pengerjaan Tugas Akhir

1.5.1 Metode Penelitian

Metode yang digunakan dalam penelitian kali ini adalah sebagai berikut :

1. Studi Pustaka (*Library Search*)

Studi Pustaka dilakukan dengan cara mempelajari teori-teori dan buku-buku yang berhubungan dengan objek kajian sebagai dasar dalam penelitian ini, dengan tujuan memperoleh dasar teoritis gambaran dari apa yang dilakukan. Teori yang dipelajari yaitu: Pembobotan dengan *Term Frequency - Inverse Document Frequency* (TF-IDF), *Vector Space Model*, *Cosine Similarity*, Django, *Linear Search, Object Relation Mapping* dan sebagainya.

2. Melakukan kajian secara *online* di Internet

Browsing pada halaman-halaman situs di Internet yang membahas tentang algoritma-algoritma yang akan digunakan dalam pembuatan program, seperti contoh algoritma untuk melakukan pembobotan, pencarian kalimat, dan sebagainya. *Browsing* juga dilakukan untuk mengumpulkan *ebook* ataupun artikel yang akan dibutuhkan dalam proses peringkasan. *Browsing* itu sendiri adalah kegiatan menelusuri informasi di internet melalui sebuah media yang dinamakan *browser*.

3. Analisa Data

Penelitian dilakukan menggunakan data hadis dari Ensiklopedi Hadis Kitab 9 Imam. Setelah dilakukan pengumpulan data, tahap selanjutnya dilakukan studi

pustaka dan analisa atas data yang sudah diperoleh untuk membuat perancangan dan implementasi aplikasi pencarian hadis.

4. Implementasi dan pengujian

Implementasi dilakukan dengan membangun sebuah aplikasi berbasis *website,* menggunakan bahasa pemrograman Python. Sedangkan untuk pengujian dilakukan melalui metode *Black Box* dan *Recall.*

1.5.2 Metode Pengembangan Sistem

Metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah *Rapid Application Development* (RAD). Metode RAD merupakan gabungan dari konsep metode iteratif yang juga berperan sebagai pelengkap atas metode klasik *waterfall*[12]. Sesuai namanya (*rapid*) RAD berfokus pada pengembangan aplikasi yang cepat, membutuhkan waktu sekitar 30-90 hari dimulai dari proses pendefinisian kebutuhan aplikasi hingga tahap *cutover*[13].

Tahap pertama dari metode RAD pada penelitian ini adalah pendefinisian *requirements* aplikasi, analisa masalah dan pengumpulan data hadis. Berdasarkan kebutuhan-kebutuhan tersebut, tahap selanjutnya adalah *user design* (perancangan sistem) yang memuat proses iterasi pembuatan *prototype* aplikasi hadis meliputi *database*, arsitektur sistem, dan antarmuka serta perbaikan *prototype* yang mengacu pada *feedback* klien. Proses ini akan terus berulang sampai klien menyetujui *prototype*. Selanjutnya tahap konstruksi sistem mulai dibangun dengan mengimplementasikan algoritma VSM, TF-IDF, *Linear Search* dan ORM Django. Pengujian dan pemeliharaan dilakukan pada tahap akhir *cutover* dengan menggunakan metode *Black Box Testing* agar dapat diketahui apakah sistem telah

berjalan dengan baik dan sesuai kebutuhan atau tidak. Tahap ini juga merupakan tahap akhir dari *life cycle* metode RAD.

1.6 Sistematika Penulisan

Sistematika penulisan menjelaskan gambaran umum serta struktur laporan tugas akhir terhadap penelitian yang akan dilakukan, terdiri dari lima bab dengan masing-masing subnya. Antara lain:

BAB I : PENDAHULUAN

Bab ini berisikan latar belakang penelitian, perumusan masalah yang diambil, tujuan penelitian, batasan masalah, metode serta sistematika pemulisan.

BAB II : STUDI PUSTAKA

Studi pustaka terdiri atas tinjauan pustaka dan landasan teori yang digunakan.

BAB III : ANALISIS DAN PERANCANGAN

Bab ini menjelaskan hasil analisa penelitian serta perancangan sistem yang akan dikembangkan sebagai solusi dari latar belakang yang diambil peneliti.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab IV merupakan uraian atas implementasi sistem yang telah dirancang serta hasil pengujian untuk sistem tersebut.

BAB V : PENUTUP

Bab terakhir berisi kesimpulan dan saran untuk penelitian selanjutnya.