

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dengan kemajuan teknologi yang sangat pesat ini sudah banyak aplikasi penunjang Al-Quran untuk memudahkan untuk mempelajarinya, yang bisa disebut atau di artikan dengan adanya Al-Quran Digital. Akan tetapi, ketika menggunakan aplikasi Al-Quran Digital yang sudah banyak beredar di dunia maya, pengguna aplikasi mengalami kesulitan pada saat mencari dari sebuah kata yang pengguna inginkan dengan salah satunya yaitu dengan kesalahan pengguna dalam penulisan dari suatu kata yang dicarinya dan aplikasi pada saat ini masih kurang atau belum menangani atau membenarkan kata yang salah.

Search engine adalah program komputer yang dirancang untuk mencari informasi yang tersedia didalam dunia maya. *web search engine* mengumpulkan informasi yang tersedia secara otomatis, dengan *search engine* tersebut kita dapat mencari berbagai informasi yang kita inginkan dari seluruh dunia, jadi dengan adanya *search engine* ini sangat memudahkan bagi kita untuk mencari data.

Dalam Penelitian ini menggunakan metode *cosine similarity*, metode *jaccard similarity*, dan *k-nearest neighbor (K-NN)* yaitu sebuah metode *information retrieval* yang menggunakan pendekatan probabilistik dalam menginferensi. Aplikasi yang paling banyak digunakan yaitu untuk klasifikasi teks. Disini menggunakan metode *cosine similarity*, metode *jaccard similarity*, dan *k-nearest neighbor (K-NN)* mengasumsikan bahwa probabilitas kemunculan sebuah kata independen terhadap posisinya di dalam teks. Tujuan dari asumsi

independensi posisi kata yaitu untuk mengurangi kompleksitas dalam perhitungan nilai-nilai probabilitas bersyarat pada teorema secara umum.

Text mining yaitu salah satu cara dalam mengatasi permasalahan pengklasifikasian. *Text mining* mengolah data berupa nama dan deskripsi yang di-input-kan. *Text mining* akan melakukan *tokenizing*, *filtering*, *stemming*, *tagging* dan *analyzing* terhadap kata-kata dalam nama dan deskripsi sehingga diperoleh *keyword*.

Sedangkan untuk proses pengukuran dari tingkat similaritas antar dokumen dilakukan dengan membandingkan suatu *keyword* dengan dokumen atau pencocokan antara *keyword* dokumen. Agar hasil pengukuran tingkat similaritas dari suatu dokumen dengan *keyword* mendapatkan hasil yang optimal atau akurat maka digunakan algoritma *text mining* dimana dalam prosesnya digunakan algoritma *cosine similarity*, *jaccard similarity*, dan *k-nearest neighbor (K-NN)* untuk membandingkan dari hasil bobot akhirnya.

Algoritma *cosine similarity* merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata didalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata didalam dokumen yang diberikan menunjukkan seberapa penting kata tersebut didalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi didalam dokumen dan frekuensi

keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (*database*).

Algoritma *jaccard similarity* merupakan salah satu metoda yang dipakai untuk menghitung *similarity* antara dua *objects (items)*. Seperti halnya *cosine distance* dan *matching coefficient*, secara umum perhitungan metode ini didasarkan pada *vector space similarity measure*. [24]

Algoritma *k-nearest neighbor (K-NN)* merupakan metode tambahan untuk membandingkan dari 2 metode yaitu *cosine similarity* dan *jaccard similarity*. *K-Nearest Neighbor (K-NN)* suatu metode yang menggunakan algoritma *supervised*. Perbedaan antara *supervised learning* dengan *unsupervised learning* adalah pada *supervised learning* bertujuan untuk menemukan pola baru dalam data dengan menghubungkan pola data yang sudah ada dengan data yang baru. Sedangkan pada *unsupervised learning*, data belum memiliki pola apapun, dan tujuan *unsupervised learning* untuk menemukan pola dalam sebuah data. [29]

Dari uraian di atas, maka untuk mendapatkan suatu data atau informasi dengan adanya aplikasi pencarian berdasarkan terjemah Al-Qur'an yang ditunjukkan untuk mempermudah pencarian. Dengan ini penulis mencoba meneliti dan menuangkan dalam bentuk Tugas Akhir dengan judul “**Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia**”

1.2 Rumusan Masalah

Berdasarkan dalam latar belakang maka didapat rumusan permasalahan yang akan diselesaikan dalam penelitian yaitu :

1. Bagaimana membangun aplikasi pencarian terjemah Al-Qur'an yang mampu menampilkan hasil pencarian terjemah Al-Qur'an dengan menggunakan metode *Cosine Similarity*, metode *Jaccard Similarity* dan metode *K-Nearest Neighbor (K-NN)* ?

1.3 Tujuan Penelitian

Adapun tujuan penelitian yaitu mengimplementasikan metode *Cosine Similarity*, metode *Jaccard* dan metode *K-Nearest Neighbor (K-NN) Similarity* pada aplikasi pencarian terjemah Al-Qur'an yang mampu mencari isi dari terjemah Al-Qur'an Bahasa Indonesia.

1.4 Batasan Masalah

Batasan masalah ini bertujuan untuk memudahkan dan menghindari adanya kegiatan di luar sasaran, sehingga dalam pembuatan laporan ini perlu ditentukan suatu batasan masalah. Batasan masalah tersebut sebagai berikut :

1. Aplikasi yang di rancang hanya bisa melakukan pencarian dari terjemahan Al-Qur'an versi Muhammad Quraish Shihab.
2. Algoritma yang digunakan pada pengklasifikasian yaitu metode *cosine similarity*, *jaccard similarity* dan *k-nearest neighbor (K-NN)*.
3. Menggunakan metode *waterfall* atau model air terjun sebagai tahapan pengembangan perangkat lunak.
4. Implementasi yang di bangun untuk aplikasi tersebut untuk *website dinamis* dengan menggunakan *PHP (Personal Home Page)* dengan media untuk penyimpanannya digunakan *MySql* dan untuk tampilan

website menggunakan *HTML*, *CSS* dan *JQuery* untuk mendapatkan tampilan yang lebih *dinamis*.

5. Rancangan yang diidentifikasi menggunakan *UML (Unified Modeling Language)* pada metode perancangan ini meliputi *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*.
6. Pencarian pada pengklasifikasian berdasarkan pencarian perkata pada terjemah Al-Qur'an dengan jumlah 6236 ayat terjemah.

1.5 Manfaat Penelitian

Manfaat dari penelitian yaitu terbentuknya suatu aplikasi pencarian terjemah Al-Qur'an dengan mengimplementasikan *text mining* untuk memudahkan dalam proses pencarian dan mendapatkan informasi terjemah dari Al-Qur'an sesuai yang dikehendaki berdasarkan tingkat kemiripan dokumen terhadap kata kunci sehingga dapat dijadikan referensi untuk penggunaan metode *cosine similarity*, *jaccard similarity* dan *k-nearest neighbor (K-NN)*.

1.6 Metodologi Penelitian

Pembuatan skripsi ini terbagi menjadi beberapa tahap pengerjaan yang tertera sebagai berikut :

1.6.1 Pengumpulan Data Yang Diperlukan.

Beberapa metode yang akan digunakan dalam pengumpulan data :

1. Studi *Litelatur*

Pada penggunaan metode ini penulis akan melakukan pencarian dari beberapa macam *literature* dan *document* untuk menunjang pengerjaan tugas akhir yang khususnya berkaitan dengan aplikasi *search engine* terjemah Al-Qur'an menggunakan teknik *text mining*.

2. *Observasi Lapangan*

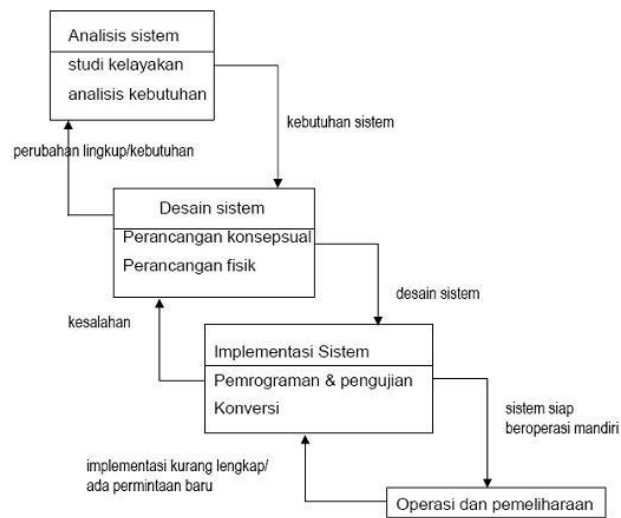
Melakukan penelitian dari berbagai macam *website* di internet yang menyediakan informasi yang *relevan* dengan permasalahan dalam pembuatan sistem ini.

1.6.2 Perancangan dan Desain Sistem

Rancangan kerja *search engine* sesuai data yang ada dan mengimplementasikan model yang diinginkan oleh pengguna, serta perancangan database dengan di dukung pembuatan *Use Case*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram* untuk mempermudah dalam proses pembuatan aplikasi selanjutnya.

1.6.3 Metode Perangkat Lunak

Pada pembuatan perangkat lunak ini dikembangkan dengan menggunakan metode pengembangan *waterfall* (Model Air Terjun), yaitu :



Gambar 1.1 Mekanisme Pengembangan Sistem dengan *Waterfall* (Model Air Terjun) [31]

Metodologi pengembangan sistem merupakan metodologi yang digunakan untuk mengembangkan suatu sistem informasi, yaitu suatu proses standar yang diikuti untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan memelihara sistem informasi. Daur hidup dari pengembangan sistem ini disebut dengan daur hidup pengembangan sistem (*SDLC = System Development Life Cycle*). Uraian lebih jelasnya mengenai pengembangan perangkat lunak diatas adalah sebagai berikut.

1. Analisis Sistem

Analisis sistem dilakukan karena adanya permintaan sistem yang baru. Tujuan utama analisis sistem adalah menentukan hal-hal detail terkait dengan apa yang akan dikerjakan oleh sistem. Tahap Analisis Sistem ini meliputi studi kelayakan dan analisis kebutuhan.

a. Studi Kelayakan

Studi kelayakan meliputi penentuan masalah, pembentukan sasaran sistem, pengidentifikasian pemakai sistem, dan pembentukan lingkup sistem

b. Analisis Kebutuhan

Analisis kebutuhan dimaksudkan untuk menghasilkan spesifikasi kebutuhan, yaitu spesifikasi rinci tentang hal-hal yang akan dilakukan oleh sistem ketika diimplementasikan. Untuk melakukan analisis kebutuhan, analisis sistem biasanya melakukan langkah-langkah seperti wawancara, riset terhadap sistem sekarang, *observasi* lapangan, kuis dan pengamatan terhadap sistem serupa.

2. Desain Sistem

Desain sistem dilakukan setelah proses analisis sistem dikerjakan.

Perancangan sistem ini dibagi menjadi 2 kelompok :

a. Perancangan konseptual terdiri dari evaluasi alternatif rancangan adalah menentukan alternatif-alternatif rancangan yang bisa dipakai, penyiapan spesifikasi rancangan adalah meliputi keluaran, penyimpanan data, masukan, prosedur pemrosesan, operasi dan penyiapan laporan

b. Perancangan fisik terdiri dari rancangan, rancangan masukan, rancangan antar muka, rancangan platform, rancangan basisdata, rancangan modul, rancangan kontrol, dokumentasi, rencana pengujian, rencana konversi

3. Implementasi Sistem

Hasil perancangan sistem diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan dengan desain yang telah dibuat.

4. Operasi dan Pemeliharaan

Pemeliharaan sistem tetap diperlukan karena mungkin sistem masih menyisakan masalah-masalah yang tidak terdeteksi selama masa pengujian sistem dan pemeliharaan juga bisa dipacu karena kinerja sistem yang menjadi menurun sehingga adanya perubahan-perubahan dalam penulisan program.

1.6.4 Pembuatan Aplikasi

Tahap ini merupakan tahap pembuatan dan pengembangan aplikasi dengan desain *system* yang diterapkan pada tahap sebelumnya. aplikasi *search engine* terjemah Al-Qur'an menggunakan teknik *text mining* dengan algoritma *cosine similarity*, algoritma *jaccard similarity* dan *k-nearest neighbor (K-NN)* dibangun menggunakan *PHP* dan *MySQL*.

1.6.5 Uji Coba System

Proses uji coba diperlukan untuk memastikan bahwa *system* yang telah dibuat sudah benar, sesuai dengan karakteristik yang ditetapkan dan tidak ada kesalahan - kesalahan dalam alur programnya.

1.6.6 Penyusunan Buku Skripsi

Tahap terakhir merupakan dokumentasi pelaksanaan skripsi. Diharapkan buku tugas akhir ini bermanfaat bagi pembaca yang ingin mengembangkan sistem ini lebih lanjut maupun pada lain kasus.

1.7 *State of The Art*

Berikut ini adalah penelitian yang dilakukan dan memiliki korelasi yang searah dengan penelitian yang dibahas, antara lain :

Nurhayati Sri., (2011). Berdasarkan penelitian yang dilakukan oleh nurhayati sri menjelaskan bahwa aplikasi dengan algoritma *NBC (naive bayes classifier)* dengan hasil mempermudah dan mempercepat proses pencarian dari kesenian tradisional jawa barat berdasarkan data masukan *keyword* dari *user*, dan untuk hasil yang *user* inginkan harus mempunyai jaringan yang tersedia lebih mendukung dan untuk tahap *filtering* tidak terlalu efektif jika menggunakan algoritma *NBC (Naive Bayes classifier)* yang tercantum dengan cara *stoplist*.

Anwar Sofiyan Agus, dkk., (2012). Pada peneliti yang dibahas oleh anwar sofiyan agus, dkk. Membahas dengan pencarian ayat hasil *stemming* yang akan digunakan sebagai kata kunci (*keyword*) pencarian pada *database indeks* Al-Quran. Perpaduan tersebut dimaksudkan untuk meningkatkan hasil pencarian ayat, dan selanjutnya dapat dikategorikan sebagai teknik *inexact string matching*. Hasil uji coba membuktikan bahwa teknik *inexact string matching* dapat diimplementasikan untuk mendukung pencarian ayat al Quran dengan nilai *F-measure* tertinggi pada data uji coba adalah 100 % dan nilai *F-measure* terendah adalah 66.66 %. Uji coba juga membuktikan bahwa teknik *inexact string*

matching lebih banyak memberikan solusi atau kemungkinan dari ayat yang dimaksud dari pada teknik *exact string matching*.

Dewantara Fajar M, dkk., (2013). Berdasarkan penelitian yang dilakukan oleh dewantara fajar m, dkk. Menyimpulkan bahwa algoritma *levenshtein distance* merupakan matrik yang digunakan untuk mengukur perbedaan jarak antara dua *string*, *Levenshtein Distance* antara dua *string* ditentukan berdasarkan jumlah minimum perubahan atau pengeditan yang diperlukan untuk melakukan *transformasi* dari satu bentuk *string* ke bentuk *string* yang lain. Setelah diketahui jarak minimumnya, kemudian ditentukan nilai persentasenya menggunakan rumus *similarity*, dan ditentukannya tingkat kemiripan sebesar 65% untuk menampilkan *string suggestion* kata yang cocok pada mesin pencari Al-Quran Digital khususnya pada surat Yasin.

Holle FH Khadijah, dkk., (2012). Pada penelitian yang dilakukan oleh Holle FH Khadijah, dkk. Meyimpulkan bahwa *search engine* ini dapat mencocokkan dokumen dengan ayat yang terdapat dalam *database* hingga didapatkan ayat yang sesuai dengan *input* teks berbahasa indonesia. Metode *cosine similarity* dapat digunakan dan cocok untuk *search engine* ayat Al-Quran dengan tingkat *precision* 9,56%, *recall* 56,33%, dan *accuracy* 99,76%. Dari 30 dokumen yang diuji, 23 dokumen terdapat ayat sesuai yang ditemukan dan pada 7 dokumen, ayat yang dimaksud tidak ditemukan. Sehingga dari 30 dokumen, 76,67% dapat dikatakan berhasil dan 23,33% masih kurang. Hasil pencarian tidak maksimal untuk dokumen yang umum atau menggunakan kata dengan makna kiasan.

Nurani Atri, dkk., (2007). Pada penelitian yang dilakukan oleh nurani atri, dkk. Menjelaskan proses klasifikasi akan dilakukan menggunakan metode *naive bayesian classifier (NBC)*. Dalam penelitian ini, proses klasifikasi referensi buku menggunakan metode *naive bayesian* memiliki nilai *presisi* 63%. Dalam melaksanakan tugasnya untuk mengklasifikasikan daftar isi buku referensi sistem dipengaruhi oleh berbagai faktor seperti pola data dan jumlah data *training*. *metode bayesian* memerlukan pengetahuan awal untuk dapat mengambil suatu keputusan. Tingkat keberhasilan metode ini sangat tergantung pada pengetahuan awal yang diberikan. Dan Untuk buku-buku yang dijadikan referensi bersama dapat mengklasifikasikan dengan baik berdasarkan nilai probabilitas tertingginya.

Hasil dari para peneliti yang telah di uraikan sebelumnya, maka di dapat kesimpulan yang terperinci dari setiap peneliti tersebut di Tabel 1.1 :

Table 1.1 Kesimpulan dari Peneliti

| No | Peneliti | Metode | Data | Teknologi | Layanan | Kinerja |
|----|---------------------------------|-------------------------------------|---------------------------|----------------|---------------|--|
| 1 | Nurhayati Sri. (2011) | <i>NBC (Naïve Bayes Classifier)</i> | Kesenian Tradisional | <i>Website</i> | <i>Search</i> | Pencarian data dan proses klasifikasi data kesenian secara otomatis |
| 2 | Anwar Sofiyan Agus, dkk. (2012) | <i>Inexact String Matching</i> | Ayat Al-Quran | <i>Desktop</i> | <i>Search</i> | Nilai <i>F-measure</i> tertinggi 100 % dan nilai <i>F-measure</i> terendah 66.66 % |
| 3 | Dewantara Fajar M, dkk. (2013) | <i>Levenshtein Distance</i> | Terjemah Al-Quran | <i>Website</i> | <i>Search</i> | Tingkat kemiripan 65% pada surat Yasin. |
| 4 | Holle FH Khadijah, dkk. (2012) | <i>Cosine Similarity</i> | Ayat Al-Quran | <i>Desktop</i> | <i>Search</i> | <i>Precision</i> 9,56%, <i>recall</i> 56,33%, dan <i>accuracy</i> 99,76%. |
| 5 | Nurani Atri, dkk. (2007) | <i>NBC (Naïve Bayes Classifier)</i> | Buku referensi matakuliah | <i>Desktop</i> | <i>Search</i> | <i>Precision</i> 63% |

1.8 Sistematika Penulisan

Sistematika pembuatan perangkat lunak berbasis *website* terdiri dari beberapa bab yang masing-masing bab telah dirancang dengan suatu tujuan tertentu. Berikut penjelasan tentang masing-masing bab :

BAB I : PENDAHULUAN

Dalam BAB I dijelaskan tentang latar belakang masalah, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II: LANDASAN TEORI

BAB II berisi tentang kajian *literature* atau studi pustaka yang berkaitan dengan permasalahan yang diamati. Teori-teori ini selanjutnya mendasari kegiatan dari analisa sistem yang akan dilakukan.

BAB III: PERANCANGAN

Pada BAB III berisi tentang menguraikan analisa terhadap sistem yang akan dibangun yaitu spesifikasi kebutuhan, arus sistem pengolahan data dan spesifikasi perangkat lunak yang dibuat.

BAB IV: IMPLEMENTASI DAN PENGUJIAN

Pada BAB IV berisi tentang perangkat implementasi program, implementasi dan pengujian program dari sistem penerjemah yang dibangun.

BAB V: PENUTUP

BAB V merupakan bab akhir yang mengemukakan hasil analisa atau kesimpulan dan saran mengenai hasil dari penelitian.