BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian ini dibuat berdasarkan beberapa penelitian lain yang telah dilakukan sebelumnya yang berkaitan dengan resep masakan, metode N-gram dan *cosine similarity*. Kajian pustaka ini dilakukan dengan tujuan untuk memperkuat landasan teori pemahaman mengenai permasalahan agar menghasilkan penyelesaian masalah dengan cara yang terbaik.

Pembahasan pertama akan dilakukan terhadap penelitian tentang sistem rekomendasi resep masakan menggunakan metode *item based collaborative filtering.* Berdasarkan hasil penelitian yang telah dilakukan oleh penulis tentang sistem rekomendasi pemilihan resep masakan ini, maka dapat diambil kesimpulan bahwa penelitian ini berhasil menyelesaikan sistem rekomendasi pemilihan resep masakan dengan menerapkan *metode item based collaborative filtering*. Sistem ini memiliki fitur menampilkan semua resep yang ada di dalam sistem, pemberian *rating* oleh pengguna, serta pencarian rekomendasi yang dapat membantu pengguna dalam memilih resep masakan berdasarkan bahan makanan yang ada dengan menampilkan rekomendasi resep masakan yang telah dihitung menggunakan metode *collaborative filtering*, dengan menggunakan kemiripan antar resep atau dengan menggunakan *item based collaborative filtering* (Kismarini, 2016).

Pada penelitian berikutnya membahas tentang Automatic Essay Scoring Sistem menggunakan N-gram dan *Cosine similarity* untuk Gamification Based Elearning Berdasarkan hasil eksperimen dan analisis yang dilakukan terhadap kinerja sistem, kita dapat menyimpulkan bahwa penerapan Metode kesamaan N-gram dan *Cosine* pada esai otomatis penilaian pemeriksaan memberikan hasil yang cukup bagus. *Unigram* dalam sistem menghasilkan nilai korelasi terbaik dengan 0,66 untuk pertanyaan itu jangan memperhitungkan urutan kata dalam jawaban. Sementara itu kombinasi n-gram menghasilkan korelasi terbaik nilai 0,67 (Utomo,2015).

Kemudian pada jurnal milik Ana Triana yang berjudul Pemanfaatan Metode *Vector Space Model Cosine similarity* pada Fitur Deteksi Hama dan Penyakit Tanaman Padi dapat disimpulkan bahwa metode *Vector Space Model* dapat digunakan untuk melakukan identifikasi *input* dengan hasil gejala yang sesuai dengan *input* pengguna sebagai *feedback* dan metode *Cosine similarity* dapat digunakan untuk melakukan identifikasi *output* berupa hama atau penyakit padi yang sesuai, sehingga keduanya dapat dimanfaatkan untuk pendeteksian hama dan penyakit pada tanaman padi (Triana et al., 2016).

2.2 Rekomendasi

Rekomendasi merupakan sebuah alat personalisasi yang menyediakan pengguna sebuah informasi daftar item-item yang sesuai dengan keinginan masing-masing pengguna. Sistem rekomendasi bertujuan untuk membantu pengguna dalam menentukan pilihan dengan cara memberikan rekomendasi. Rekomendasi yang diberikan diharapkan dapat membantu pengguna dalam proses pengambilan keputusan, seperti resep masakan apa yang akan dipilih ketika pengguna kebingungan dalam mengolah bahan makanan (Ricci, 2011).

Sistem rekomendasi umum (*generalized recommender system*) harus mengenal terlebih dahulu setiap pengguna yang ada. Setiap sistem rekomendasi harus membangun dan memelihara pengguna model atau pengguna profile yang berisi ketertarikan pengguna (Jannach, 2010). Sebagai contoh, sistem rekomendasi di *website cookpad* menyimpan setiap komentar pengguna, dan *review/rating* yang diberikan oleh pengguna terhadap suatu resep masakan.

2.3 Information Retrival

Information Retrival merupakan suatu sistem yang menemukan *(retrieve)* informasi yang sesuai dengan kebutuhan *pengguna* dari kumpulan informasi secara otomatis. Prinsip kerja sistem temu kembali informasi jika ada sebuah kumpulan dokumen dan seorang pengguna yang memformulasikan sebuah pertanyaan (*request* atau *query*). Jawaban dari pertanyaan tersebut adalah sekumpulan dokumen yang relevan dan membuang dokumen yang tidak relevan (Salton, 1989).

2.4 Text Mining

Text mining adalah sebuah penerapan konsep dari data *mining* yang digunakan untuk pencarian pola dalam teks yang berfungsi untuk mencari informasi yang diinginkan dengan hasil tertentu. Dalam tahap pemprosesan, *text mining* memerlukan beberapa tahap awal yang digunakan untuk mempersiapkan teks agar dapat diubah menjadi lebih terstruktur(Budi Susanto, 2013).

Text mining memiliki manfaat untuk mempermudah pencarian dan menghasilkan rekomendasi yang dapat membantu pengguna untuk menggunakan informasi dari sebuah sistem. Dalam proses *text minning* memiliki tahapan umum diantaranya adalah *case folding,* tokenisasi*.* Diantara tahapan tersebut merupakan tahap *preprocesing text*(Even, Yahir Zohar,2002).

2.4.1 Prepocessing Text

Pada *text mining,* struktur data yang baik dapat mempermudah proses komputerisasi secara otomatis. Maka dari itu, diperlukan beberapa tahapan untuk pengubahan dari informasi yang strukturnya sembarang menjadi lebih terstruktur sesuai dengan kebutuhan. Tahapan awal dari *text mining* adalah *text preprocessing*

yang bertujuan untuk mempersiapkan teks menjadi data yang terstruktur(Even, Yahir Zohar,2002).

2.4.2 Case Folding

Case folding merupakan kesamaan case dalam sebuah dokumen untuk mempermudah pencarian. Tidak semua dokumen teks konsisten dalam penggunaan huruf capital. Oleh karena itu peran case folding adalah mengkonversi teks dalam dokumen menjadi suatu bentuk standar (mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf "a" sampai dengan "z" yang diterima)(Even, Yahir Zohar,2002).

2.4.3 Tokenisasi

Tokenisasi merupakan proses pemotongan dokumen menjadi bagian-bagian yaitu token. Tokenizing berfungsi untuk membuang beberapa karakter tertentu yang dianggap sebagai tanda baca. Metode ini adalah serangkaian metode dalam proses *preprocessing*(Even, Yahir Zohar,2002).

2.5 Pembobotan Kata (*Term*)

Metode *TF-IDF* (*Term Frequency Inverse Document Frequency)* merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen (Robertson, 2005). Metode ini menggabungkan dua konsep untuk perhitungan bobot yaitu, frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Terdapat beberapa cara atau metode dalam melakukan pembobotan kata pada metode *TF-IDF*, yaitu melalui skema pembobotan *query* dan dokumen.

2.5.1 *Term Frequency* (*TF*) dan Pembobotan *TF* (W*TF*)

Term Frequency (*TF*) ialah frekuensi kemunculan term (kata) dalam suatu dokumen, sedangkan W*TF* ialah suatu proses untuk melakukan pembobotan untuk tiap term (kata). Adapun untuk menentukan nilai *TF* dan *WTF* ditunjukkan pada persamaan 2.1 (Nathania dkk, 2017).

$$W_{tf_{t,d}} = \begin{cases} 1 + log_{10} tf_{t,d}, & dimana\ tf_{t,d} > 0 \\ 0, & dimana\ \ tf_{t,d} = 0 \end{cases} \qquad 2.1$$

Keterangan:

- $W_{tf_{t,d}}$: Hasil dari pembobotan $tf_{t,d}$
- $tf_{t,d}$: Frekuensi kemunculan *t* pada dokumen *d*

2.5.2 Document Frequency (DF_t) dan Inverse Document Frequency (IDF_t)

Document Frequency (DFt) ialah jumlah dokumen yang memiliki term (kata) *t*, dan Inverse Document Frequency ialah jumlah dari dokumen yang memiliki term (kata) *t* yang dicari dalam kumpulan dokumen yang ada, *IDF* dapat dihitung menggunakan persamaan 2.2 (Nathania, 2017)

$$idf_t = log_{10} N/df_t \quad 2.2$$

Keterangan:

- idf_t : Hasil dari invers df_t
- df_t : Jumlah dokumen yang memiliki *t*
- N : Banyak dokumen yang ada

2.5.3 Pembobotan *TF-IDF* ($W_{t,d}$)

Pembobotan *TF-IDF* ($W_{t,d}$) ialah proses perkalian dari pembobotan *TF* dan IDF_t, untuk menghitung nilai $W_{t,d}$ dapat menggunakan persamaan 2.3 (Nathania, 2017)

$$W_{t,d} = W_{tf_{t,d}} * idf_t \quad 2.3$$

2.5.4 Normalisasi

Normalisasi dilakukan untuk mempermudah dalam menghitung nilai *cosine similarity*, adapun persamaan yang dapat digunakan untuk melakukan normalisasi dapat dilihat pada persamaa 2.4 (Nathania, 2017).

$$w_{t,d} = \frac{w_{t,d}}{\sqrt{\sum_{t=1}^{n} {w_{t,d}}^2}} \quad 2.4$$

2.6 N-Gram

N-gram adalah suatu proses pemotongan terhadap N-karakter yang diambil dari *string.* Untuk menentukan *N-gram* yang utuh dilakukan dengan menambahkan garis bawah (*blank*) pada awal dan akhir kata. Hal tersebut dilakukan untuk membantu menentukan kondisi awal kata dan akhir kata. Maka pada kata "TEKS" untuk mejadi *N-gram* dapat diproses sebagai berikut:

Unigram : T,E,K,S

Bigram : _T, TE, EK,KS, dan S

Trigram : _TE,TEK,EKS, KS_ dan S_ _

Sebuah kata dengan panjang *k* ditambahkan dengan garis bawah, akan memiliki *k*+1 *bigram*, *k*+1 *trigram*, *k*+1 quadgram.

Kesuksesan pencocokan berdasarkan *N-gram* dilakukan pada tiap kata yang dikomposisikan menjadi bagian-bagian kecil, dan kesalahan yang ada di dalam

pencocokan hanya berpengaruh pada sebagian kecil bagian tersebut. Maka dari itu kita menghitung *N-gram* yang sama pada dua kata, kemudian kita akan mendapatkan kesamaan dua kata tersebut yang tidak terpengaruh oleh kesalahan tekstual (Permadi, 2008).

2.7 Cosine similarity

Metode *cosine similarity* merupakan perhitungan metode digunakan untuk kesamaan atau kedekatan antar dua dokumen. Nilai kesamaan vector *query* dan vector dokumen maka hasil *query* akan dipandang relevan dengan dokumen. Normalisasi sangat diperlukan untuk dokumen yang panjang cenderung memiliki nilai lebih besar dan memiliki frekuensi kemunculan kata yang besar pula. $CosSim\ (d_j, q) = \frac{d_j \cdot q}{|d_j|\,|q|} = \frac{\sum_{i=1}^{t}(w_{ij} \times w_{iq})}{\sqrt{\sum_{i=1}^{t}(w_{ij})^2} \times \sqrt{\sum_{i=1}^{t}(w_{iq})^2}}$ 2.5

Keterangan :

d = dokumen

q = kata kunci

w_{ij} = bobot kata ke *i* pada dokumen *j*

w_{iq} = bobot kata ke *i* pada dokumen *q*

Secara umum,fungsi *similarity* adalah fungsi yang menerima dua buah objek dan mengembalikan kemiripan*(similarity)* antara kedua objek tersebut berupa bilangan riil. Umumnya, nilai yang dihasilkan oleh fungsi *similarity* berkisar pada interval [0...1]. Namun ada juga beberapa fungsi *similarity* yang menghasilkan nilai yang berada di luar interval tersebut. Untuk memetakan hasil fungsi tersebut pada interval [0...1] dapat dilakukan normalisasi [1]. *Cosine similarity* akan digunakan dalam ruang positif,dimana hasilnya dibatasi dengan(0,1) (Triana, 2016). Perhitungan *cosine similarity* ditunjukkan pada persamaan sebagai berikut:

$$CosSim\ (d_j, q) = d_j \bullet \mathrm{q} = \sum_{i=1}^{t}(w_{ij} \times w_{iq}) \qquad 2.6$$

Keterangan:

d = dokumen

q = kata kunci

w_{ij} = bobot kata ke *i* pada dokumen *j*

w_{iq} = bobot kata ke *i* pada dokumen *q*

2.8 Evaluasi

Evaluasi merupakan proses membandingkan kriteria dan standar untuk melihat keberhasilan yang ditetapkan dengan hasil implementasi. Informasi yang didapatkan yaitu antara standar yang ditetapkan dengan hasil yang telah dicapai. Evaluasi yang digunakan untuk mengukur kinerja sistem pada permasalahan rekomendasi yaitu menghitung nilai *recall, precison.* Hasil dari perhitungan tersebut selanjutnya akan dibandingkan (Purwanti, 2015). Misalnya yang relevan 7 dari top 10 hasil pencarian relevan, maka nilai presisinya sebesar 0,7 atau 70%.

2.8.1 Threshold

Untuk memperoleh hasil pencarian dokumen yang maksimal diperlukan sebuah nilai ambang batas (*threshold*) agar sistem dapat memilah mana dokumen yang mirip dan mana yang tidak. Dokumen dengan nilai ≥ 50% *threshold* dapat dinyatakan mirip, sedangkan dokumen dengan nilai < 50% *threshold* dinyatakan tidak mirip. Untuk mendapatkan nilai batas diperlukan suatu data training untuk melakukan uji coba (Anshori, 2010).

2.8.2 *Precision Recall* Relevansi

Penerapan prinsip relevansiyangdigunakanpadaperkembangan*systeminformation retrieval* adalah penggunaan *recall*, *precision* dan *f-measure*. *Presicion* dapat mengevaluasi kemampuan dari sistem temu kembali informasi untuk menemukan data *top-ranked* yang paling relevan dimana didefinisikan sebagai suatu persentase data yang dikembalikan benar-benar relevan dengan *query* pengguna. *Precision* merupakan proporsi dari suatu set yang diperoleh yang relevan (Purwanti, 2015).

$$\boldsymbol{Precision} = \frac{Jumlah\ dokumen\ relevan\ terambil}{Jumlah\ seluruh\ dokumen\ terambil} \qquad 2.8$$

Recall dapat mengevaluasi kemampuan dari sistem temu kembali informasi untuk menemukan seluruh item yang relevan dengan koleksi data dimana didefinisikan sebagai suatu persentase data yang relevan terhadap *query* dari pengguna dan yang diterima*. Recall* merupakan proporsi dari semua dokumen yang relevan dikoleksi termasuk dokumen yang diperoleh (Purwanti, 2015).

$$\boldsymbol{Recall} = \frac{Jumlah\ dokumen\ relevan\ terambil}{Jumlah\ seluruh\ dokumen\ relevan} \qquad 2.9$$