

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Sebelumnya

Berikut beberapa penelitian terdahulu yang terkait dengan metode yang digunakan peneliti untuk membuat sistem ujian *essay* dengan metode *Cosine Similarity & TF-ID*.

2.1.1. Mohammad Agus Salim (2017). “Pengembangan Aplikasi Penilaian Ujian Essay Berbasis Online Menggunakan Algoritma Nazief Dan Adriani Dengan Metode Cosine Similarity” Universitas Negeri Surabaya.

Penelitian ini dilakukan oleh Agus Salim pada tahun 2017. penelitian ini mengangkat masalah yaitu itu ujian yang ada pada SMK Negeri 3 Buduran Sidoarjo yang masih menggunakan metode koreksi manual untuk ujian *essay* atau ujian tulis. oleh karena itu peneliti dalam hari ini adalah Agus Salim melakukan penelitian berupa pembuatan sistem ujian menggunakan metode *Cosine Similarity* yang diharapkan dapat memecahkan masalah pada SMK Negeri 3 Buduran Sidoarjo. sistem dibangun menggunakan media online atau menggunakan basis web peternakan sistem ini akan diakses secara *online* oleh banyak siswa pada SMK Negeri 3 Buduran Sidoarjo. Tuliskan dibangun menggunakan bahasa pemrograman *PHP* untuk melakukan pemrosesan data. Hal ini dikarenakan sistem dibangun menggunakan tak sesuai dan bahasa Bella yang dapat digunakan untuk melakukan pemrosesan data. Selain itu sistem menggunakan database *MySQL* untuk menampung semua data yang digunakan pada saat sistem berjalan. sistem melakukan perhitungan menggunakan media *Cosine Similarity* untuk mencocokkan antara jawaban dari siswa dan kunci jawaban yang telah ditentukan oleh guru sebelumnya. Dari Aspek pengoperasian Aplikasi penilaian ujian *essay* berbasis *online* secara keseluruhan fungsi dapat berjalan dengan sangat baik dan mudah. dalam pengoperasiannya rata-rata indikator mendapatkan nilai 94,61%.

2.1.2. Loura Yasni, Imam Much Ibnu Subroto, Sam Farisa Chaerul Haviana (2018). “Sistem Pendukung Keputusan Pemilihan Lokasi TPA Sampah Menggunakan Metode Min_Max Inference Fuzzy” Universitas Islam Sultan Agung, Semarang.

Penelitian ini membahas tentang penentuan dosen pembimbing pada tugas akhir di kampus Tempat penelitian dilakukan. tugas akhir merupakan salah satu syarat kelulusan bagi mahasiswa yang melakukan pendidikan di perguruan tinggi. Rumah Sakit merupakan pengujian seberapa kompetensi mahasiswa menempuh pendidikan selama sekian tahun di kampus merdeka. dalam melakukan pembuatan tugas akhir mahasiswa akan didampingi oleh dosen pembimbing yang akan mengarahkan mereka dalam pengerjaan tugas akhir mereka. dosen pembimbing harus dipilih berdasarkan kesesuaian dari tugas akhir yang diangkat mahasiswa baik dari tema topik dan masalah yang diangkat harus sesuai dengan karakteristik karakteristik dan kemampuan si dosen pembimbing. Oleh karena itu kalau masalah ini akan diangkat dan dibuat sebuah sistem dosen pembimbing menggunakan metode cosine similarity. metode ini digunakan untuk melakukan klasifikasi dengan cara melakukan stretching atau pemecahan pada proposal yang diajukan mahasiswa. dengan metode similarity akan diperoleh Tema atau masalah atau topik yang diangkat mahasiswa dalam melakukan tugas akhir mereka. dari sini akan diperoleh dosen pembimbing yang cocok untuk di berikan ke mahasiswa tersebut.

2.1.3. Dewa Ayu Rai Ariantini, Arie S. M. Lumenta, Agustinus Jacobus (2016) “Pengukuran Kemiripan Dokumen Teks Bahasa Indonesia Menggunakan Metode Cosine Similarity”, Universitas Sam Ratulangi Manado, Indonesia.

perkembangan zaman yang terjadi saat ini menyebabkan banyaknya media-media yang memuat berbagai macam karya tulis yang dapat dibaca atau dijadikan referensi. hal ini menjadikan Semua orang dapat membuat karya tulis mereka sendiri dengan cara menjadikan karya tulis yang telah dibuat sebelumnya menjadi referensi. namun sayangnya banyak sekali lahir

baru-baru ini memiliki tingkat plagiasi yang tinggi. Hal ini dikarenakan kurangnya inovasi dalam menciptakan sebuah karya tulis sehingga para pencipta karya tulis makan cara cepat dengan cara menulis kembali hasil tulisan pada karya tulis terdahulu. Untuk mengatasi hal ini dilakukan penelitian untuk menciptakan sebuah sistem yang dapat mengkorupsi seberapa tingkat kemiripan dua karya tulis atau lebih. Sistem ini dibuat menggunakan metode cosine similarity yakni salah satu metode text mining yang akan memecah dokumen berdasarkan tiap kata dalam melakukan pembubutan untuk dicari seberapa tingkat kemiripan dari dua dokumen tersebut

Proses perhitungan kemiripan adalah proses untuk mencari kemiripan antar dokumen. Selanjutnya proses preproses teks adalah untuk menghilangkan tanda baca dan merubah semua huruf menjadi huruf kecil pada dokumen tugas mahasiswa. Kemudian proses sistem menghitung nilai *TF* atau *Term frekuensi*, adalah untuk menghitung kemunculan kata kemudian menyimpannya kedalam basis data tabel indeks. Selanjutnya adalah proses perhitungan kemiripan adalah proses untuk mencari kemiripan antar dokumen teks. kemudian sistem menghitung nilai kemiripan dokumen A dan dokumen B maka *user* bisa melihat seberapa sbanyak dokumen yang mirip. Hasil penelitian menunjukkan bahwa persentase tingkat ketepatan klasifikasi sistem adalah sebesar 81%

**2.1.4. Rizki Tri Wahyuni, Dhidik Prastiyanto, dan Eko Supraptono (2017).
 “Penerapan Algoritma Cosine Similarity dan Pembobotan TF-ID pada Sistem Klasifikasi Dokumen Skripsi”, Universitas Negeri Semarang.**

Penelitian oleh Rizki Tri Wahyuni, Dhidik Prastiyanto, dan Eko Supraptono (2017) membahas tentang Banyaknya arsip dokumen skripsi yang terkumpul dalam bentuk soft file yang tidak terklasifikasi dengan baik mengakibatkan proses pencarian kembali menjadi sulit. Untuk mengakses informasi yang dibutuhkan menjadi kurang cepat dan tepat apabila keseluruhan dokumen disimpan dalam satu folder *database*. Maka dari itu diperlukan suatu sistem yang dapat mengklasifikasikan dokumen secara

otomatis ke dalam folder berbeda pada *database* agar lebih mudah dalam mengelola dokumen yang ada. Metode *TF-ID* merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*Term*) terhadap dokumen. Metode *Cosine Similarity* merupakan metode untuk menghitung kesamaan antara dua buah objek yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah model *waterfall*, sedangkan metode penelitian yang digunakan adalah metode *Research and Development (R&D)*. Data latih yang digunakan dalam penelitian ini berjumlah 50 dokumen skripsi dengan beberapa kategori yang berbeda. Hasil penelitian menunjukkan bahwa persentase tingkat ketepatan klasifikasi sistem adalah sebesar 98%

2.1.5. Maskur, Faiz Rizky Andriansyah (2016). “Implementasi Web Semantik Untuk Aplikasi Pencarian Tugas Akhir Menggunakan Ontologi Dan Cosine Similarity”, Universitas Muhammadiyah Malang.

Penelitian oleh Maskur, Faiz Rizky Andriansyah (2016) membahas tentang Sistem pencarian pada sistem umumnya masih menggunakan sistem pencarian konvensional, yaitu pencarian yang mencocokkan kata kunci dengan data yang ada. Pencarian konvensional memiliki kelemahan yaitu tidak bisa mencari data yang relevan dengan kata kunci. Untuk mengatasi masalah ini dibangun sistem yang dapat melakukan pencarian yang dapat mencari hasil yang relevan dengan kata kunci. Sistem ini berbasis web yang digabungkan dengan ontologi. Ontologi merupakan hirarki yang menjelaskan antara hubungan suatu kata. Ontologi berfungsi untuk mencari hubungan antara kata yang dimasukkan oleh pengguna. Pencarian data dalam ontologi menggunakan *Query Sparql* yang menyerupai *Query database* pada umumnya namun tidak terlalu kompleks seperti *Query database*. Untuk menunjang hasil yang diperoleh digunakan *Cosine Similarity*. Metode *Cosine Similarity* digunakan untuk memberikan bobot pada setiap dokumen yang didapatkan, sehingga dokumen yang didapat semakin relevan dengan kata kunci. Pengujian menggunakan metode *precision* untuk menghitung nilai

relevansi hasil yang didapat dari ontologi, sedangkan pengujian dengan kappa *statistics* digunakan untuk menghitung nilai dari hasil *Cosine Similarity* dengan cara membandingkan hasil yang didapat dari sistem dan hasil menurut pengamatan pakar. Dari hasil pengujian didapatkan nilai *Cosine Similarity* nya lebih besar sama dengan 0,33%.

2.2. Landasan Teori

Bagian ini bertujuan untuk menjelaskan mengenai referensi penelitaian terdahulu yang menguraikan landasan teori-teori yang mendukung judul, dan mendasari pembahasan secara detail. Pada bab ini juga dituliskan tentang metode, teknik dan *tools* (komponen) yang digunakan untuk pembuatan aplikasi atau tujuan penelitian.

2.2.1. Definisi Sistem

Sistem adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu. Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara dimana yang berperan sebagai penggeraknya yaitu rakyat yang berada di negara tersebut.

Ada banyak pendapat tentang pengertian dan definisi sistem yang dijelaskan oleh beberapa ahli. Berikut pengertian dan definisi sistem menurut beberapa ahli:

- Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan

kesatuan yang nyata, seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi (Jogianto, 2005:2).

- Sistem adalah kumpulan-kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dengan lainnya (Indrajit, 2001:2).
- Sistem adalah himpunan dari bagian-bagian yang saling berhubungan, yang secara bersama mencapai tujuan-tujuan yang sama (Sidharta, 1995:9).
- Sistem adalah seperangkat elemen yang membentuk kumpulan atau prosedur-prosedur atau bagan-bagan pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data dan/atau barang pada waktu rujukan tertentu untuk menghasilkan informasi dan/atau energi dan/atau barang (Murdick, R. G, 1991:27).
- Sistem adalah kumpulan dari elemen-elemen yang beroperasi bersama-sama untuk menyelesaikan suatu sasaran (Davis, G. B, 1991:45).

2.2.2. Definisi Ujian *Essay*

Tes bentuk *essay* adalah sejenis tes kemajuan belajar yang memerlukan jawaban yang bersifat pembahasan atau uraian kata-kata (Arikunto, 2003: 162). Secara umum terdapat dua macam bentuk ujian untuk mengevaluasi sejauh mana pelajar dapat mengambil ilmu yang telah diberikan, yaitu secara lisan dan tertulis. Ujian secara lisan memiliki kelemahan yaitu sang penguji dan pelajar yang diuji harus berada pada ruang dan waktu yang sama, hal ini bertentangan dengan konsep dasar *e-learning* yang tidak mengenal batasan ruang dan waktu, ujian jenis ini juga kurang efisien bila ditinjau dari segi waktu dan sumber daya, apalagi bila yang akan diuji berjumlah banyak. Berbeda dengan ujian secara lisan, ujian secara tertulis jauh lebih menghemat waktu dan sumber daya, ujian secara tertulis juga dapat dilakukan dari jarak yang jauh, sehingga dapat dikatakan bahwa pada kondisi umum ujian secara tertulis lebih baik daripada ujian secara lisan.

Ada tiga jenis ujian yang dapat dilakukan secara tertulis, yaitu pilihan ganda, isian singkat dan *essay*. Dari ketiga jenis ujian yang dapat dilakukan secara tertulis tersebut, ujian yang paling mudah untuk diperiksa secara otomatis adalah pilihan ganda. Pada ujian jenis pilihan ganda, pelajar cukup

memilih jawaban yang dianggapnya paling benar dari beberapa pilihan yang disediakan. Bentuk pilihan ganda bisa berupa “pilihan ganda satu jawaban” atau “pilihan ganda beberapa jawaban”. Kelemahan jenis ujian ini adalah kurang dapatnya penguji untuk mengetahui sejauh mana pengetahuan pelajar yang diuji, dan dimungkinkannya untuk menebak jawaban dalam ujian jenis ini. Pemeriksaan jawabannya sangatlah mudah, pemeriksa cukup mencocokkan pilihan yang diisi dengan kunci jawaban. Tingkat akurasi dari pemeriksaan secara otomatis bila dibandingkan dengan pemeriksaan manual mencapai 100%, karena bentuk ujiian seperti ini sangat mudah diolah oleh komputer, komputer cukup membandingkan jawaban pelajar dengan kunci jawaban yang ada di *database*-nya. Bentuk ujian pilihan ganda telah banyak diaplikasikan dalam pengujian menggunakan komputer.

Jenis ujian tertulis berikutnya adalah dengan isian singkat. Pada ujian jenis ini, pelajar cukup mengisi jawaban atas pertanyaan yang diajukan dalam satu atau beberapa kata. Ujian jenis ini sedikit lebih baik dari pilihan ganda karena pelajar yang diuji tidak dapat menebak jawaban karena tidak disediakan pilihan. Masih kurang dapatnya dievaluasi secara mendalam sejauh mana pengetahuan pelajar yang diuji merupakan kelemahan jenis ujian ini. Pemeriksaan jawaban seperti ini masih cukup mudah jika digunakan komputer, karena komputer cukup mencocokkan jawaban yang ditulis pelajar yang diuji (berupa *string*) dengan jawaban yang ada dalam *database*. Pada ujian seperti ini perlu diperhatikan masalah persamaan arti kata (*sinonim*), agar penilaian yang dilakukan oleh komputer dapat tepat.

Jenis ujian tertulis yang terakhir adalah dengan cara soal *essay*. Soal *essay* biasanya meminta pelajar untuk menuliskan jawaban berupa hasil analisis, konsep yang didapatnya setelah mengikuti pelajaran, atau uraian solusi berkaitan dengan masalah yang dikemukakan pada soal. Bentuk soal *essay* merupakan bentuk pengujian yang secara akurat mampu menggambarkan pemahaman pelajar secara mendalam mengenai topik yang diujikan. Selain itu ujian *essay* juga merangsang kemampuan mengemukakan pendapat dalam bentuk tulisan. Ujian dalam bentuk *essay* memiliki kekurangan dari segi

sumber daya yang dibutuhkan yaitu manusia dan waktu. Ketika sebuah ujian *essay* diikuti oleh banyak peserta, waktu yang dibutuhkan untuk memeriksa jawaban akan bertambah. Akibatnya pemeriksaan ujian dilakukan dengan terburu-buru sehingga memungkinkan terjadinya ketidaktelitian dalam memeriksa. Apabila pemeriksaan ujian dilakukan oleh banyak manusia, ketidaksamaan pertimbangan yang digunakan dalam melakukan penilaian antara peserta ujian satu dengan yang lain sering kali terjadi. Hal ini akan berdampak ketidakadilan nilai yang didapat oleh peserta ujian.

Untuk mengatasi masalah-masalah dalam penilaian ujian *essay*, sistem penilaian *essay* secara otomatis menggunakan komputer dapat menjadi sebuah solusi yang baik. Bantuan komputer akan sangat meringankan beban pemeriksa ujian apabila ujian tersebut diikuti oleh orang banyak, hal ini juga berdampak baik bagi peserta ujian, karena dapat menghindari faktor subjektifitas dalam memeriksa ujian sehingga nilai yang didapat oleh peserta ujian dapat dikatakan adil. Berikut ini adalah beberapa keuntungan menggunakan komputer sebagai alat untuk memeriksa jawaban *essay*:

1. Komputer dapat memberikan waktu dan sumber daya untuk menguji materi yang berhubungan dengan ujian dalam jumlah yang hampir tidak terbatas sebelum digunakan untuk melakukan proses penilaian terhadap ujian itu sendiri.
2. Komputer dapat memeriksa dan menganalisa ujian *essay* secara lebih mendetil dibandingkan dengan manusia.
3. Komputer dapat membandingkan setiap *essay* dalam sebuah set virtual secara terus menerus dalam ukuran berapa pun satu sama lain, satu hal yang tidak mungkin dilakukan oleh manusia.
4. Komputer dapat berlaku konsisten dalam melakukan proses penilaian, dari *essay* ke *essay*, dari waktu ke waktu. Ia tidak akan lelah, bosan, terganggu, tidak teliti, atau melenceng dari standar yang telah ditentukan.
5. Komputer dapat bersifat objektif sepenuhnya, dan penilaian yang dilakukannya tidak terpengaruh oleh hubungannya dengan siswa yang dinilai.

6. Komputer dapat melakukan analisis yang kompleks dan rumit yang manusia tidak dapat lakukan tanpa bantuan alat atau orang lain.

2.2.3. Metode *Cosine Similarity*

Cosine Similarity merupakan metode yang digunakan untuk menghitung tingkat kesamaan (*similarity*) antar dua buah objek. Untuk tujuan klustering dokumen, fungsi yang baik adalah fungsi *Cosine Similarity*. *Cosine Similarity* berfungsi untuk membandingkan kemiripan antar dokumen, dalam hal ini yang dibandingkan adalah *query* dengan dokumen latih. Dalam menghitung *Cosine Similarity*, pertama yang dilakukan yaitu melakukan perkalian skalar antara *query* dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang *query* yang telah dikuadratkan, setelah itu di hitung akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut di bagi dengan hasil perkalian panjang dokumen dan *query*. Rumus perhitungan *Cosine Similarity* adalah sebagai berikut :

$$\text{cosSim}(d_j, q_k) = \frac{\sum_{i=1}^n (t_{d_{ij}} \times t_{q_{ik}})}{\sqrt{\sum_{i=1}^n t_{d_{ij}}^2 \times \sum_{i=1}^n t_{q_{ik}}^2}} \dots\dots\dots(2.1)$$

Keterangan :

$\text{cosSim}(d_j, q_k)$: tingkat kesamaan dokumen dengan *query* tertentu

$t_{d_{ij}}$: *Term* ke-i dalam vector untuk dokumen ke-j

$t_{q_{ik}}$: *Term* ke-i dalam vector untuk dokumen ke-k

n : jumlah *Term* yang unik dalam data set

Berikut adalah contoh perhitungan manual dengan menggunakan metode *Cosine Similarity* :

1. Ditentukan terlebih dahulu masing-masing *query*, yaitu *query* dari jawaban (D), *query* dari *key* jawaban (Q) dan gabungan keduanya (*Queries*)
2. Ketiga *query* tersebut dihilangkan stoplist atau simbol-simbol yang tidak mempengaruhi penilaian, seperti tanda titik, tanda koma, tanda seru, dan sebagainya

3. Ketiga *query* tersebut dihilangkan *stopwords* atau kata-kata umum yang lazim digunakan dalam suatu *query*, seperti "dan", "jika", "di", "namun", "tetapi", dan sebagainya

Tabel 2.1 Sampel *Query*

Dokumen	Term Yang Mewakili Dokumen
Q	data keterangan mentah belum diolah belum bisa dijadikan acuan mengambil suatu keputusan
D1	data data mentah sekumpulan
D2	belum belum bisa data diolah fakta informasi mentah menyajikan
D3	acuan belum belum bisa data dijadikan diolah informasi keputusan mentah pengambilan

4. Dihitung nilai *Term frequency query* jawaban dan *query key* jawaban terhadap *queries*. Jadi perhitungan *Term* di *query* jawaban dan *query key* jawaban merujuk pada *Term* yang terdapat dalam *queries*
5. Dihitung nilai *document frequency* (n) atau banyaknya file (N) yang memiliki suatu *Term* untuk tiap *Term* dalam *queries*
6. Dihitung nilai *inverse document frequency* dengan rumus : $\log(N/n) + 1$

Tabel 2.2 Perhitungan *TF* dan *IDF*

No.	Term	TF				df	IDF
		Q	D1	D2	D3		$\log(n/df)+1$
1.	Acuan	1	0	0	1	2	1.30103
2.	Belum	2	0	2	2	3	1.12493874
3.	Bisa	1	0	1	1	3	1.12493874
4.	Data	1	2	1	1	4	1
5.	Dijadikan	1	0	0	1	2	1.30103
6.	Diolah	1	0	1	1	3	1.12493874
7.	Fakta	0	0	1	0	1	1.60205999
8.	Informasi	0	0	1	1	2	1.30103
9.	Keputusan	1	0	0	1	2	1.30103
10.	Keterangan	1	0	0	0	1	1.60205999
11.	Mengambil	1	0	0	0	1	1.60205999
12.	Mentah	1	1	1	1	4	1
13.	Menyajikan	0	0	1	0	1	1.60205999
14.	Pengambilan	0	0	0	1	1	1.60205999
15.	Sekumpulan	0	1	0	0	1	1.60205999
16.	Suatu	1	0	0	0	1	1.60205999

7. Dikalikan nilai *Term frequency* dengan nilai *inverse document frequency* tiap *Term* dalam Q maupun D.

Tabel 2.3 Perhitungan $TF/IDF = TF * IDF$

No.	Term	Q	D1	D2	D3
1.	Acuan	1.30103	0	0	1.30103
2.	Belum	2.24987748	0	2.24987748	2.24987748
3.	Bisa	1.12493874	0	1.12493874	1.12493874
4.	Data	1	2	1	1
5.	Dijadikan	1.30103	0	0	1.30103
6.	Diolah	1.12493874	0	1.12493874	1.12493874
7.	Fakta	0	0	1.60205999	0
8.	Informasi	0	0	1.30103	1.30103
9.	Keputusan	1.30103	0	0	1.30103
10.	Keterangan	1.60205999	0	0	0
11.	Mengambil	1.60205999	0	0	0
12.	Mentah	1	1	1	1
13.	Menyajikan	0	0	1.60205999	0
14.	Pengambilan	0	0	0	1.60205999
15.	Sekumpulan	0	1.60205999	0	0
16.	Suatu	1.60205999	0	0	0

8. Dihitung hasil perkalian skalar masing-masing *query* jawaban terhadap *query* *key* jawaban. Hasil perkalian dari setiap jawaban dengan *query* dijumlahkan (sesuai pembilang pada rumus di atas)

Tabel 2.4 Hasil Perkalian Skalar Tiap D terhadap Q

No.	Term	D1	D2	D3
1.	Acuan	0	0	1.6926790609
2.	Belum	0	5.0619486750112	5.0619486750112
3.	Bias	0	1.2654871687528	1.2654871687528
4.	Data	2	1	1
5.	Dijadikan	0	0	1.6926790609
6.	Diolah	0	1.2654871687528	1.2654871687528
7.	Fakta	0	0	0
8.	Informasi	0	0	0
9.	Keputusan	0	0	1.6926790609
10.	Keterangan	0	0	0
11.	Mengambil	0	0	0
12.	Mentah	1	1	1
13.	menyajikan	0	0	0
14.	pengambilan	0	0	0
15.	sekumpulan	0	0	0
16.	suatu	0	0	0
SUM (Ji . Q) :		3	9.5929230125167	14.670960195217

9. Dihitung hasil perkalian vektor tiap *query key* jawaban dan *query* jawaban

Tabel 2.5 (TF/IDF(Q,D))²

No.	Term	Q	D1	D2	D3
1.	acuan	1.692679	0	0	1.692679
2.	belum	5.061949	0	5.061949	5.061949
3.	bisa	1.265487	0	1.265487	1.265487
4.	data	1	4	1	1
5.	dijadikan	1.692679	0	0	1.692679
6.	diolah	1.265487	0	1.265487	1.265487
7.	fakta	0	0	2.566596	0
8.	informasi	0	0	1.692679	1.692679
9.	keputusan	1.692679	0	0	1.692679
10.	keterangan	2.566596	0	0	0
11.	mengambil	2.566596	0	0	0
12.	mentah	1	1	1	1
13.	Menyajikan	0	0	2.566596	0
SUM		22.37074	7.566596	16.418794	18.930235
(TF/IDF(Q,D)) ² :		9			
SQRT(SUM(TF/IDF(Q,D)) ²)		4.729773	2.750745	4.052011	4.350889

10. Dihitung nilai *Cosine Similarity* (nilai vektor beda antara D terhadap Q) dengan rumus:

$$\text{sim}(d,q) = \frac{\sum_{k=1}^i (\text{weight}_{ik} \text{weight}_{qk})}{\sqrt{\sum_{k=1}^i (\text{weight}_{ik}^2 \text{weight}_{qk}^2)}} \dots (2.2)$$

Tabel 2.6 Hasil Cosine

No.	Teks	Nilai <i>Similitary</i>	Nilai (Nilai <i>Similitary</i> * bobot)
1.	D1 : data data mentah sekumpulan	0.23058480	23.06 %
2.	D2 : belum belum bisa data diolah fakta informasi mentah menyajikan	0.50054143	50.05 %
3.	D3 : acuan belum belum bisa data dijadikan diolah informasi keputusan mentah pengambilan	0.71291907	71.29 %

2.2.4. Metode TF-ID

Dalam skema arsitektur temu balik informasi, terdapat sebuah proses yang berkaitan dengan pembobotan kata atau *Term* baik secara lokal maupun global. Pembobotan lokal hanya berpedoman pada frekuensi munculnya *Term* dalam

suatu dokumen dan tidak melihat frekuensi kemunculan *Term* tersebut di dalam dokumen lainnya. Pendekatan dalam pembobotan lokal yang paling banyak diterapkan adalah *Term frequency (TF)* meskipun terdapat skema lain seperti pembobotan *biner*, *augmented normalized TF*, *logaritmik TF* dan *logaritmik alternatif*.

Pembobotan global digunakan untuk memberikan tekanan terhadap *Term* yang mengakibatkan perbedaan dan berdasarkan pada penyebaran dari *Term* tertentu di seluruh dokumen. Banyak skema didasarkan pada pertimbangan bahwa semakin jarang suatu *Term* muncul di dalam total koleksi maka *Term* tersebut menjadi semakin berbeda. Pemanfaatan pembobotan ini dapat menghilangkan kebutuhan *stop word removal* karena *stop word* mempunyai bobot global yang sangat kecil. Namun pada prakteknya lebih baik menghilangkan *stop word* di dalam fase *pre-processing* sehingga semakin sedikit *Term* yang harus ditangani. Pendekatan terhadap pembobotan global mencakup *inverse document frequency (IDF)*, *squared IDF*, *probabilistic IDF*, *GF-IDF*, *entropy*. Pendekatan *IDF* merupakan pembobotan yang paling banyak digunakan saat ini. Beberapa aplikasi tidak melibatkan bobot global, hanya memperhatikan *TF*, yaitu ketika *TF* sangat kecil atau saat diperlukan penekanan terhadap frekuensi *Term* di dalam suatu dokumen (Karyono, 2012). Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut yang rendah pada kumpulan dokumen (*database*) (Intan, 2006).

Rumus umum untuk *TF-ID* (Al-Talib, 2013) : $TF-ID(ti,dj) = TF(ti,dj) \log N/n_i$(2.3)

Keterangan :

- $TF-ID(ti,dj)$: pembobotan kata atau *Term* i pada dokumen j
- $TF(ti,dj)$: banyak kata atau *Term* i pada dokumen j
- N : total dokumen dalam dataset
- n_i : total dokumen yang memunculkan *Term* i

Tabel 2.7 Perhitungan *TF* dan *IDF*

No.	Kata Dalam Kalimat	<i>TF</i>				<i>df</i>	<i>IDF</i>
		Kunci	Jwb 1	Jwb 2	Jwb 2		$\text{Log}(n/df)+1$
1.	acuan	1	0	0	1	2	1.30103
2.	belum	2	0	2	2	3	1.12493874
3.	bisa	1	0	1	1	3	1.12493874
4.	data	1	2	1	1	4	1
5.	dijadikan	1	0	0	1	2	1.30103
6.	diolah	1	0	1	1	3	1.12493874
7.	fakta	0	0	1	0	1	1.60205999
8.	informasi	0	0	1	1	2	1.30103
9.	keputusan	1	0	0	1	2	1.30103
10.	keterangan	1	0	0	0	1	1.60205999
11.	mengambil	1	0	0	0	1	1.60205999
12.	mentah	1	1	1	1	4	1
13.	menyajikan	0	0	1	0	1	1.60205999
14.	pengambilan	0	0	0	1	1	1.60205999
15.	sekumpulan	0	1	0	0	1	1.60205999
16.	suatu	1	0	0	0	1	1.60205999

Tabel 2.7 merupakan tabel yang berisi contoh penerapan metode *TF-ID* dalam pembuatan sistem ini. Metode *TF-ID* digunakan untuk melakukan pembobotan pada setiap kata yang akan dilakukan pengujian. Langkah ini dilakukan untuk mengambil nilai bobot yang nantinya akan digunakan untuk melakukan perhitungan pada langkah selanjutnya dengan metode *Cosine Similarity*.

2.2.5. Bahasa Pemrograman *PHP*

Menurut Arief pada tahun 2011 Bahasa pemrograman *PHP* adalah Bahasa *server-side –scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis. Karena *PHP* merupakan *server-side-scripting* maka sintaks dan perintah-perintah *PHP* akan diesksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan *format HTML*.

Dalam pembuatan sistem ini pemrograman *PHP* digunakan untuk melakukan pemrosesan data yang diolah pada sistem ini. Bahasa *PHP* digunakan karena Bahasa *PHP* merupakan Bahasa yang memang sudah diterapkan untuk

melakukan pemrosesan data yang diterapkan untuk aplikasi atau sistem yang berjalan di *platform* web.

Bahasa *PHP* digunakan karena Bahasa *PHP* merupakan salah satu Bahasa pemrograman yang mudah untuk dipahami dan dikembangkan. Selain itu Bahasa *PHP* juga merupakan Bahasa yang memiliki referensi yang cukup banyak yang dapat dicari di dunia maya atau internet. Ketika terjadi *error* atau kesalahan, Bahasa *PHP* relatif lebih mudah untuk dipecahkan daripada Bahasa pemrograman lain seperti *android* atau *java*. Selain itu Bahasa *PHP* juga ringan pada saat dilakukan pengembangan. Oleh karena itu tidak perlu butuh spesifikasi laptop yang tinggi untuk dapat melakukan pembuatan program dengan Bahasa pemrograman *PHP*.

2.2.6. Bootstrap

Bootstrap merupakan *Framework* ataupun *Tools* untuk membuat aplikasi web ataupun situs web *responsive* secara cepat, mudah dan gratis. *Bootstrap* terdiri dari *CSS* dan *HTML* untuk menghasilkan *Grid*, *Layout*, *Typography*, *Tabel*, *Form*, *Navigation*, dan lain-lain. Di dalam *Bootstrap* juga sudah terdapat *jQuery plugins* untuk menghasilkan komponen UI yang cantik seperti *Transitions*, *Modal*, *Dropdown*, *Scrollspy*, *Tooltip*, *Tab*, *Popover*, *Alert*, *Button*, *Carousel* dan lain-lain.

Dengan bantuan *Bootstrap*, kita bisa membuat *responsive* website dengan cepat dan mudah dan dapat berjalan sempurna pada browser-browser populer seperti *Chrome*, *Firefox*, *Opera* dan *Internet Explorer*.

Bootstrap digunakan dalam pembuatan sistem ini dikarenakan *bootstrap* merupakan *framework* atau *library* yang dapat berjalan di platform aplikasi berbasis web. Seperti yang sudah dijelaskan sebelumnya bahwa *framework PHP* dapat membantu membuat sistem yang dibuat menjadi ramah terhadap resolusi layar tempat sistem itu diakses. Hal ini dapat membuat hasil tampilan sistem yang dibuat menjadi lebih menarik dan mudah ketika digunakan.