

BAB II

LANDASAN TEORI

2.1 Tinjauan Terhadap Penelitian Yang Terkait

Berikut beberapa ulasan penelitian yang terkait dan menjadi inspirasi pada penelitian ini :

Rahimi Fitri, dan Arifin Noor Asyikin (2015) melakukan penelitian dengan judul : *“Aplikasi penilaian ujian essay otomatis menggunakan metode cosine similarity”* Ujian dengan sistem esai merupakan bentuk evaluasi dimana pilihan jawaban tidak disediakan, dan siswa harus menjawab dengan kalimat, sehingga dapat melatih siswa dalam menyampaikan sesuatu informasi secara verbal, selain itu ujian essay juga menuntut pemahaman yang lebih baik akan suatu ilmu dan dapat digunakan untuk mengukur tingkat pemahaman manusia akan suatu ilmu secara lebih mendalam. Untuk menyamakan frekuensi setiap kata yang terdapat didalam kalimat yang ada digunakan persamaan Tf atau Term Frequency, Term Frequency merupakan suatu faktor yang menentukan bobot kata yang didasarkan pada jumlah frekuensi kata dalam sebuah dokumen. Berdasarkan hasil penelitian bahwa sistem penilaian essay otomatis menggunakan metode cosine similarity telah berjalan dengan baik untuk ujian essay dalam bahasa Inggris Hasil uji coba menunjukkan kesesuaian nilai sistem dengan nilai yang diberikan oleh pengajar adalah rata-rata 89,48%.

Selanjutnya, Karfindo Karfindo, dan Firlan Mustafa (2017) melakukan penelitian dengan judul : *“Pengembangan aplikasi Computer Based Test (CBT) untuk Sekolah Menengah Atas (SMA)”*. Perkembangan teknologi informasi membuat banyak aspek mengalami perubahan, seperti penerapan Ujian Nasional

(UN) bagi siswa Sekolah Menengah Atas (SMA). Sebelumnya, UN diselenggarakan dengan menggunakan media kertas, saat ini beralih menggunakan media komputer. Hal ini memang membawa banyak dampak positif bagi perkembangan dunia pendidikan. Adanya peralihan media ujian ini membuat banyak muncul berbagai aplikasi Ujian Berbasis Komputer (UBK) atau Computer Based Test (CBT), tapi barulah sebatas siswa mengerjakan soal, setelah selesai muncullah hasil ujiannya. Dengan menerapkan ilmu statistik pada aplikasi UBK, pihak sekolah tidak hanya bisa mengetahui hasil akhir dari ujian siswa, tetapi juga dapat melihat berapa jumlah soal yang sulit untuk dikerjakan, pilihan opsi jawaban apa yang banyak dipilih siswa. Dengan adanya perhitungan secara statistik ini, dapat membantu pihak sekolah dalam menganalisa kemampuan para siswa, sehingga bisa mengambil keputusan untuk meningkatkan kemampuan siswa dengan lebih efektif.

Dan yang terakhir, Mohammad Agus Salim dan Yeni Anistyasari (2017) juga pernah melakukan penelitian dengan judul : *“Pengembangan aplikasi penilaian ujian essay berbasis online menggunakan algoritma nazief dan adriani dengan metode cosine similarity”* Tujuan dari penelitian ini adalah: (1) Membuat suatu sistem penilaian otomatis pada ujian essay berbasis *online* menggunakan algoritma Nazief dan Adriani dengan metode *cosine similarity*. (2) Mengetahui hasil koreksi hasil belajar siswa yang menggunakan metode konvensional dengan koreksi otomatis menggunakan algoritma Nazief dan Adriani dengan metode *cosine similarity*. Metode penelitian yang digunakan pada penelitian ini menggunakan *Research and Development* (R&D) hanya sampai tahap ujicoba produk. Peneliti melakukan ujicoba pada kelas X pada jurusan teknik komputer dan jaringan SMK

Negeri 3 Buduran sebanyak 1 kelas pada kompetensi dasar 3.1 mata pelajaran pemrograman dasar. Hasil penelitian menunjukkan bahwa : Aplikasi penilaian ujian essay berbasis online menggunakan algoritma Nazief dan Adrian dengan metode cosine similarity dibuat menggunakan bahasa pemrograman HTML, PHP, Javascript dengan bantuan JQuery 3.1.1, dan MySQL. Serta menggunakan Bootstrap sebagai CSS tampilan dari aplikasi. Setelah dilakukan validasi produk diperoleh hasil dari aspek pengoperasian aplikasi rata-rata indikator didapat nilai 94,61%, aspek desain aplikasi didapat nilai 93,33%, dan aspek isi didapat nilai 93,33%. Tidak ada perbedaan yang signifikan antara hasil koreksi aplikasi penilaian ujian essay berbasis *online* menggunakan algoritma Nazief dan Adriani dengan metode *cosine similarity* dan metode koreksi secara manual .

Berdasarkan tinjauan terhadap penelitian-penelitian di atas, bahwa sistem penilaian ujian sekolah butuh pelaksanaan dengan cbt dan pengoreksian secara otomatis, dengan *cosine similarity* dapat digunakan untuk mengoreksi soal essay secara otomatis.

2.2 Sekilas Tentang Ujian Sekolah SMK TRIDHARMA BUDHIDAYA

Pada tahun 2017 Kemendikbud menginisiasi Ujian Sekolah Berstandar Nasional (USBN) pada satuan pendidikan SMP, SMA/SMK untuk beberapa mata pelajaran. Ketentuan tentang penyelenggaraan USBN tertuang dalam Permendikbud Nomor 3, Tahun 2017 tentang Penilaian Hasil Belajar oleh Pemerintah dan Penilaian Hasil Belajar oleh Satuan Pendidikan (Kemendikbud, 2017). Pada pasal 2 ayat (2) disebutkan bahwa penilaian hasil belajar oleh satuan pendidikan dilakukan melalui USBN. USBN adalah kegiatan pengukuran capaian kompetensi peserta didik yang dilakukan satuan pendidikan untuk mata pelajaran

tertentu dengan mengacu pada SKL untuk memperoleh pengakuan atas prestasi belajar. Sementara itu, USBN adalah kegiatan pengukuran dan penilaian kompetensi peserta didik terhadap standar kompetensi lulusan untuk mata pelajaran yang tidak diujikan dalam USBN. Pada tataran operasional, penyelenggaraan USBN diatur dalam Prosedur Operasional Standar (POS) USBN yang ditetapkan oleh Ditjen Pendidikan Dasar dan Menengah, Kemendikbud (Ditjen Dikdasmen, 2017a).

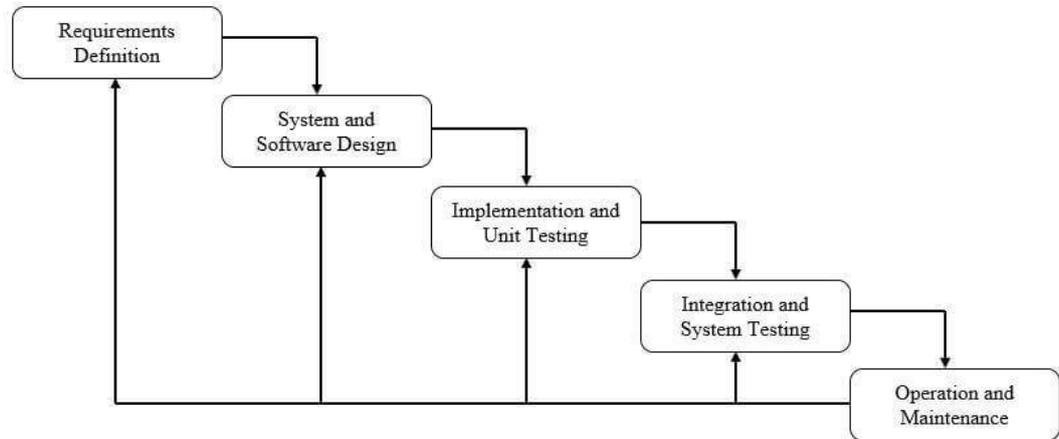
SMK Tridharma Budhidaya, diselenggarakan oleh Yayasan Daya Besar pimpinan Drs. Aggi Tjetju, SK., SM., HM dan didirikan pada 3 Agustus 1973 No.36 dan Diresmikan Pemakaian pada Tahun 1975. Hingga saat ini pada tahun pelajaran 2018/2019 telah meluluskan sebanyak 49 angkatan dengan jumlah peserta didik tiap tahun rata-rata meluluskan 80 peserta didik.

Ujian sekolah adalah kegiatan yang dilakukan oleh satuan pendidikan untuk mengukur pencapaian kompetensi peserta didik sebagai pengakuan prestasi belajar dari sekolah. Ujian membutuhkan kertas dalam pelaksanaannya serta melakukan audit penilaian secara manual yang biasanya dilakukan oleh guru, sehingga standar pengaplikasian ujian bisa berproses lama dalam hal pengoreksian dengan ini ujian sekolah di SMK Budhidaya ingin dilakukan secara komputersasi agar memudahkan dalam hal pengoreksian dan waktu.

2.3 Sekilas Tentang Metode *Waterfall*

Menurut (Ian Sommerville, 2011) dalam buku "*Software Engineering, 9th Edition*". Metodologi *waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah

(seperti air terjun) melewati fase-fase requirements, system and desain, implementasi (konstruksi), pengujian dan maintenance.



Gambar 2.1 Metodologi *Waterfall* (Ian Sommerville, 2011)

Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut :

1. *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2. *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4. *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke customer

5. *Operation dan maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata.

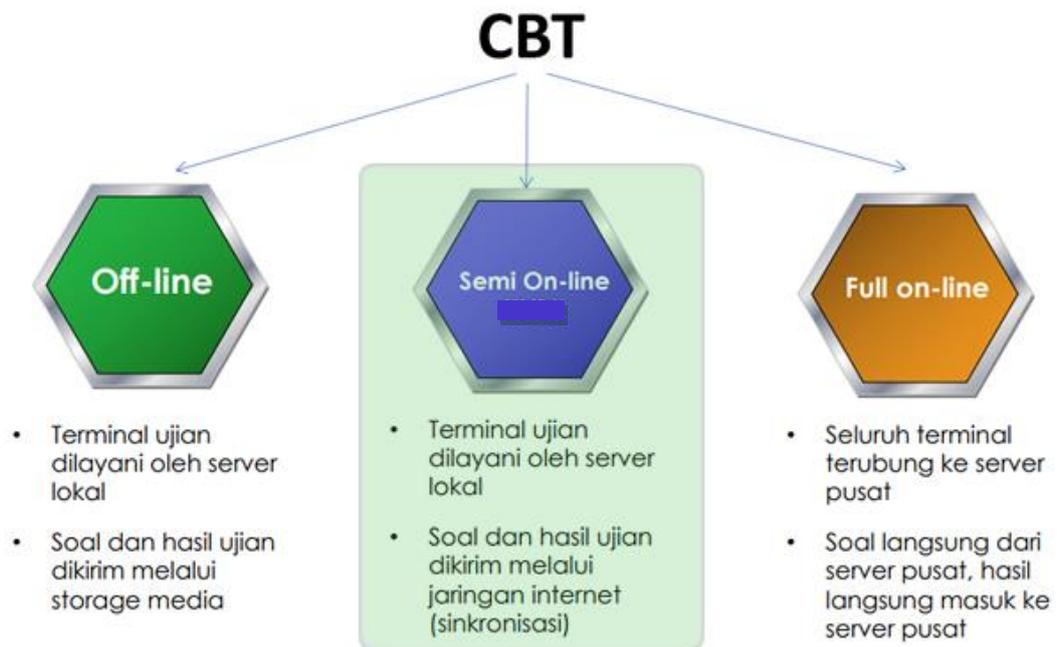
2.4 **Computer Based Test (CBT)**

Menurut (Putra, 2013), *Computer Based Test (CBT)* telah banyak digunakan secara luas karena teknologi informasi sudah menjadi sangat populer. *Test* tersebut mudah diadministrasikan oleh komputer atau perangkat elektronik yang setara, dan siswa segera dapat mengakses hasil *test* mereka. Banyak peneliti menyatakan bahwa sistem *Computer Based Test (CBT)* merupakan alat evaluasi diri yang berharga untuk belajar mandiri.

Karakteristik dari tes ini sama dengan tes konvensional yaitu menggunakan satu perangkat tes untuk beberapa peserta dengan panjang tes yang sama. Perbedaannya terletak pada teknik penyampaian butir soal yang tidak lagi menggunakan kertas, baik untuk naskah soal maupun lembar jawaban. Sistem koreksi langsung dilakukan oleh komputer.

- CBT offline Menggunakan Jaringan Lokal,
- CBT semi-online Menggunakan Jaringan internet dan lokal, dan
- CBT online Menggunakan Full jaringan Internet.

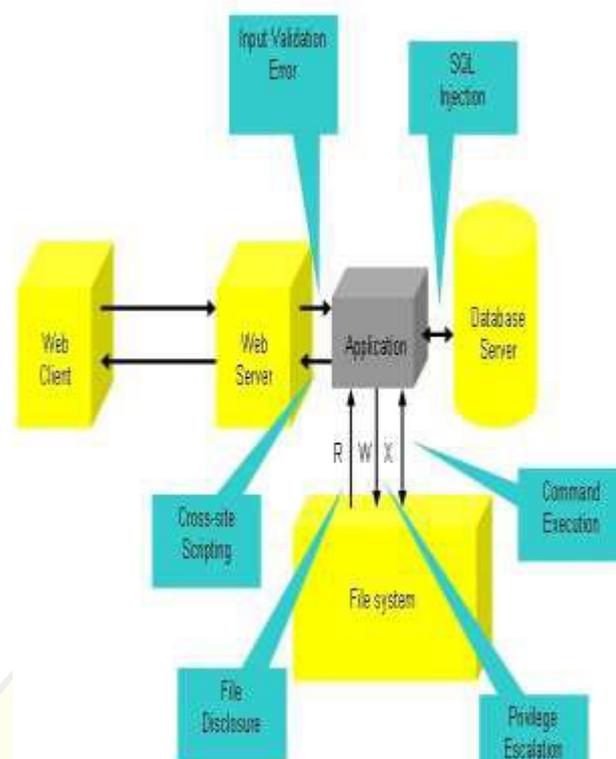
Dari uraian di atas, arsip dapat dibedakan melalui bagan sebagai berikut :



Gambar 2.2 Bagan klasifikasi CBT

2.5 Arsitektur CBT

Menurut Leo Willyanto Santoso (2008:95) Dalam implementasi aplikasi ujian online ini, pertama dilakukan dengan pengujian pada web client, web server dan aplikasi web ke dalam database server Mysql. Konsep gambaran Arsitektur aplikasi CBT ke database server secara keseluruhan dapat dilihat pada Gambar 2.1.



Gambar 2.3 Mekanisme CBT Leo Willyanto Santoso (2008:95)

2.6 Konsep Implementasi

Karena implementasi merupakan kegiatan yang penting dari keseluruhan proses perencanaan kebijakan. Menurut Mulyadi (2015:12), implementasi mengacu pada tindakan untuk mencapai tujuan-tujuan yang telah ditetapkan dalam suatu keputusan. Tindakan ini berusaha untuk mengubah keputusan-keputusan tersebut menjadi pola-pola operasional serta berusaha mencapai perubahan-perubahan besar atau kecil sebagaimana yang telah diputuskan sebelumnya. Implementasi pada hakikatnya juga merupakan upaya pemahaman apa yang seharusnya terjadi setelah program dilaksanakan. Dalam tataran praktis, implementasi adalah proses pelaksanaan keputusan dasar.

Proses tersebut terdiri atas beberapa tahapan yakni:

1. Tahapan pengesahan peraturan perundangan.

2. Pelaksanaan keputusan oleh instansi pelaksana.
3. Ketersediaan kelompok sasaran untuk menjalankan keputusan.
4. Dampak nyata keputusan baik yang dikehendaki maupun tidak.
5. Dampak keputusan sebagaimana yang diharapkan instansi pelaksana.
6. Upaya perbaikan atas kebijakan atau peraturan perundangan.

Proses persiapan implementasi setidaknya menyangkut beberapa hal penting yakni:

1. Penyiapan sumber daya, unit dan metode.
2. Penerjemahan kebijakan menjadi rencana dan arahan yang dapat diterima dan dijalankan.
3. Penyediaan layanan, pembayaran dan hal lain secara rutin.

2.6.1 Algoritma Cosine Similarity

Metode Cosine Similarity adalah metode untuk menghitung kesamaan atau kemiripan dari dua buah dokumen. Untuk menyamakan frekuensi setiap kata yang terdapat didalam kalimat yang ada digunakan persamaan Tf atau *Term Frequency*, *Term Frequency* merupakan suatu faktor yang menentukan bobot kata yang didasarkan pada jumlah frekuensi kata dalam sebuah dokumen. Nilai jumlah kemunculan suatu kata (*tf*) diperhitungkan dalam pemberian bobot terhadap suatu kata. Semakin besar jumlah kemunculan suatu kata (*tf* tinggi) dalam dokumen, semakin besar pula bobotnya dalam dokumen atau akan memberikan nilai kesesuaian yang semakin besar (Yates, et al., 1999).

Metode ini bertujuan adalah menyamakan kedua kalimat pada suatu dokumen yang nantinya akan dibandingkan, kemudian hasil perbandingan akan digunakan sebagai nilai similarity.

Ada beberapa metode yang dapat digunakan untuk mengatasi permasalahan tersebut seperti : *Project Essay Grade* (PEG), *Intelligent Essay Assessor*, *e-Rater*, *Bayesian Essay Test Scoring System* (Betsy) (Valenti, et al., 2003). Metode lainnya seperti *Latent Semantic Analysis* (LSA) yang pada awalnya digunakan untuk melakukan *indexing* pada sistem temu balik informasi (Deerwester, et al., 1990; Aditya, et al., 2009), kemudian metode ini dikembangkan lagi dalam sebuah system yang bernama SIMPLE atau system penilaian esai otomatis untuk menilai ujian dalam bahasa Indonesia dimana setiap kalimat yang terdapat dalam setiap jawaban diekstraksi dan direpresentasikan kedalam bentuk matematik. Nilai dari jawaban esei diperhitungkan, dengan mencocokkan ada atau tidak adanya kata yang dianggap penting, pada kelompok kata dalam matriks yang telah dipersiapkan oleh *human rater* (penilai manusia). (Putri Ratna, et al., 2007). Menghitung *tf* dan pembobotan kata Tahapan selanjutnya adalah menghitung frekuensi dari setiap kata yang terbentuk dengan rumus sebagai berikut .

$$W_{ij} = \frac{Tf_{ij}}{\sum(Tf_{ij})} \times \left(\log \left(\frac{N}{n_i} \right) + 1 \right)$$

Penjelasan dari formula tersebut adalah sebagai berikut :

Dimana :

- W_{ij} : Bobot kata/term t_j terhadap dokumen di
- tf_{ij} : Jumlah kemunculan kata/term t_j dalam di
- N : jumlah semua dokumen yang ada dalam database
- n_i : Jumlah dokumen yang mengandung kata/term
- t_j : (Minimal ada satu kata yaitu term t_j)

Perhitungan kemiripan menggunakan *cosine similarity* Tahapan ini dilakukan perhitungan kemiripan dari jawaban esai siswa dengan jawaban kunci yang dimiliki Guru, dengan menggunakan persamaan berikut ini.

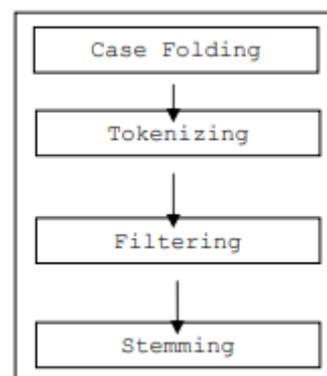
$$Similarity(q,d) = \frac{\sum_{k=1}^t W_{qk} \times W_{dk}}{\sqrt{\sum_{k=1}^t (W_{qk})^2} \sqrt{\sum_{k=1}^t (W_{dk})^2}}$$

Jadi pada tahapan ini hanya membandingkan dua buah dokumen. Pada tahapan ini menghasilkan output berupa nilai kemiripan antara kedua dokumen tersebut yang selanjutnya akan dikonversi menjadi nilai siswa.

2.6.2 Penambangan Teks (Text Mining)

Text Mining memiliki definisi menambang data yang berupa teks dimana sumber data biasanya didapatkan dari dokumen, dan tujuannya adalah mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisa keterhubungan antara dokumen (Ratna Maria, 2013). Text mining bisa dianggap subjek riset yang tergolong baru. Text mining dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian dan menganalisa unstructured text dalam jumlah besar. Dalam member solusi, text mining mengadopsi dan mengembangkan banyak teknik dari bidang lain, seperti data mining, information retrieval, statistic dan matematik, machine learning, linguistic, natural language processing, dan visualization. Kegiatan riset untuk text mining antara lain ekstraksi dan penyimpanan teks, preprocessing akan konten teks, pengumpulan data statistic dan indexing dan analisa konten. Permasalahan yang dihadapi pada text mining sama dengan permasalahan yang terdapat pada data mining, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang

terus berubah, dan data noise. Perbedaan diantara keduanya adalah pada data yang digunakan, pada data mining, data yang digunakan adalah structured data, sedangkan pada text mining, data yang digunakan pada 12 umumnya adalah unstructured data, atau minimal semistructured. Hal ini menyebabkan adanya tantangan tambahan pada text mining yaitu struktur teks yang kompleks dan tidak lengkap, arti yang tidak jelas dan tidak standard, serta bahasa yang berbeda ditambah translasi yang tidak akurat. Tahapan yang dilakukan secara umum yaitu Ekstraksi dokumen. Teks yang dilakukan proses text mining, pada umumnya memiliki beberapa karakteristik diantaranya adalah memiliki dimensi yang tinggi, terhadap noise pada data, dan terdapat struktur teks yang tidak baik. Cara yang digunakan dalam mempelajari struktur data teks adalah dengan terlebih dahulu menentukan fitur-fitur yang mewakili setiap kata untuk setiap fitur yang ada pada dokumen, sebelum menentukan fitur-fitur yang mewakili, diperlukan tahap pre-processing yang dilakukan secara umum dalam text mining pada dokumen, yaitu case folding, tokenizing, filtering, dan stemming (Raymond J.Mooney, 2016), seperti terlihat pada Gambar 2.4.



Gambar 2.4 : Proses Penambangan Teks

2.6.3 Case Folding dan Tokenizing

Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil, hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimeter.

Tahap tokenizing atau parsing adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya, sedangkan tahap filtering adalah tahap mengambil kata-kata penting dari hasil term. Bisa menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist/stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-word. Contoh stopwords adalah "yang", "dan", "di", "dari", dan seterusnya.

2.6.4 Stemming

Tahap stemming adalah tahap mencari root kata dari tiap kata hasil filtering. Pada tahap ini dilakukan proses pengambilan berbagai bentukan kata kedalam suatu representasi yang sama. Tahap ini kebanyakan dipakai untuk teks berbahasa Inggris dan lebih sulit diterapkan pada teks berbahasa Indonesia. Hal ini dikarenakan bahasa Indonesia tidak memiliki rumus bentuk baku yang permanen.

2.6.5 Tahap Penilaian

Menurut (Sugiyono, 2016:144). Dalam Jurnal Informasi Teknologi dan Pembelajaran Volume 2 No 2 yang berjudul "*Pengembangan aplikasi test online dengan menggunakan framework Codeigniter di smk darul ma'wa plumpang pada mata pelajaran jaringan Dasar kelas x*". Hal pertama yang dilakukan untuk implementasi algoritma Cosine Similarity ini yaitu melakukan proses teknik purposive sampling adalah "teknik pengambilan sampel sumber data dengan

pertimbangan tertentu.” Untuk mengetahui daya pembeda dari instrumen soal yang digunakan untuk melakukan tes selanjutnya, untuk mengetahui tingkat kesukaran setiap soal Maka format data dibuat menjadi seperti berikut ini :

Untuk aturan Penskoran Consine Similarity adalah sebagai berikut :

Tabel 2.1 Aturan Penskoran Consine Similarity

Consine Similarity (X)	Skor Perolehan
$X \geq 0,75$	1
$0,5 \leq X < 0,75$	0,5
$X < 0,5$	0,1
$X = 0$	0

Untuk aturan Penilaian Validator adalah sebagai berikut :

Tabel 2.2 Skala Penilaian validator

Kategori	Persentase (%)	Bobot
Tidak baik	0 - 25	1
Kurang baik	26 – 50	2
Baik	50 - 75	3
Sangat Baik	75 - 100	4

2.7 Software Yang Digunakan

Software yang digunakan dalam pembuatan tugas akhir ini adalah aplikasi berbasis web.

2.7.1 Aplikasi Web

Menurut Sidik dalam Arizona (2017:107) mengatakan bahwa, ” *Situs Web* (Website) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep hiperlink yang memudahkan surfer (sebutan bagi pemakai komputer yang melakukan penyelusuran informasi di Internet) untuk mendapatkan informasi dengan cukup mengklik suatu link berupa teks atau gambar maka informasi dari teks atau gambar akan ditampilkan secara lebih terperinci (detail)”.

2.7.1.1 Tools untuk Membangunnya

Proses *design* akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan *detail* (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan *programmer* untuk melakukan aktivitas pembuatan sistemnya.

2.7.1.1.1 HTML

Menurut (Didik Setiawan , 2017) dalam buku “*Buku Sakti Pemrograman Web*” . *HTML (Hyper Text Markup Language)* adalah sebuah bahasa formatting yang digunakan untuk membuat sebuah halaman website. Di dalam dunia pemrograman berbasis website(Web Programming), *HTML* menjadi pondasi dasar pada halaman website. sebuah file *HTML* di di simpan dengan ekstensi *.html* (dot html). dan dapat di eksekusi atau diakses menggunakan web browser(Google Chrome, Mozilla Firefox, Opera, Safari dan lain-lain). Hingga kini telah mengembangkan HTML 5, sebuah versi terbaru dari HTML yang mendukung tidak

hanya gambar dan teks, namun juga menu interaktif, audio, video dan lain sebagainya.

2.7.1.1.2 CSS

Menurut (Didik Setiawan , 2017) dalam buku "*Buku Sakti Pemrograman Web*". *CSS (Cascading Style Sheet)* merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. *CSS* digunakan oleh *web programmer* dan juga web designer untuk menentukan warna, tata letak *font*, dan semua aspek lain dari presentasi dokumen di situs mereka. Saat ini, hampir tidak ada situs web yang dibangun tanpa kode *CSS*.

2.7.1.1.3 PHP (PHP Hypertext Preprocessor)

Menurut (Didik Setiawan , 2017) dalam buku "*Buku Sakti Pemrograman Web*". *PHP (PHP Hypertext Preprocessor)* adalah bahasa pemrograman tingkat tinggi yang dipasang pada dokumen HTML. *PHP* termasuk bahasa program yang hanya bisa berjalan di sisi *server*, atau sering disebut *Side Server Language*. Jadi, program yang dibuat dengan kode *PHP* tidak bisa berjalan kecuali dia dijalankan pada server web. salah satu web server yang biasa kita pakai dan gratis adalah *Apache*, karena berbasis web, maka aplikasi yang dibuat dengan *PHP* adalah berbasis web, artinya kita bisa melihat hasilnya dengan menjalankannya dari web browser seperti; *Internet Explorer(IE)*, *Mozilla Firefox*, *Chrome* atau browser populer lainnya.

2.7.1.1.4 JavaScript

Menurut (Didik Setiawan , 2017) dalam buku "*Buku Sakti Pemrograman Web*". *JavaScript* merupakan salah satu bahasa script website yang paling banyak

digunakan untuk menambah manipulasi script *HTML* dan *CSS* pada sisi client/browser. *Javascript* disisipkan pada halaman web menggunakan tag `<script>`. *Javascript* mampu memberikan fungsionalitas lebih pada website, seperti validasi *form*, berkomunikasi dengan *server* serta membuat website lebih interaktif dan animatif.

2.7.1.1.5 JQuery

Menurut (Didik Setiawan, 2017) dalam buku "*Buku Sakti Pemrograman Web*". *JQuery* adalah *library Javascript* yang dibangun untuk mempercepat memperingkas serta menyederhanakan manipulasi dokumen *HTML*, penanganan *event*, animasi, dan interaksi *Ajax* untuk mempercepat pengembangan web.

2.7.1.1.6 Bootstrap

Menurut (Gregorius, 2016) dalam buku "*Pemrograman Bootstrap untuk Pemula*". *Bootstrap* adalah *framework front-end* yang intuitif dan *powerful* untuk pengembangan aplikasi web yang lebih cepat dan mudah. *Bootstrap* menggunakan *HTML*, *CSS*, dan *Javascript*.

Bootstrap dikembangkan pada tahun 2011 oleh tim desainer dan pengembang di *twitter*, Mark Otto dan Jacob Thornton. Tujuan utamanya adalah untuk menjaga konsistensi dan mempermudah perawatan pada kode yang mereka buat.

2.7.1.1.7 Basis Data

Menurut (Jubilee Enterprise, 2017) dalam buku "*Otodidak MySQL untuk Pemula*". *Basis data (atau database)* adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap database mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada didalamnya.

2.7.1.1.7.1 MySQL

Menurut (Jubilee Enterprise,2017) dalam buku "*Otodidak MySQL untuk Pemula*". *MySQL* adalah software atau program Database Server. sedangkan *SQL(Structured Query Language)* adalah bahasa pemrogramannya, dia itu bahasa permintaan (query) dalam database server, termasuk dalam *MySQL* itu sendiri. *SQL* juga dipakai dalam software database server lain, seperti SQL Server, Oracle, PostgreSQL.

2.8 Pemodelan Sistem Dengan UML (Unified Modelling Language)

Menurut (Adi Nugroho, 2017) dalam buku "*Pemrograman Berorientasi Objek Menggunakan C#*". *Unified Modelling Language(UML)* adalah bahasa pemodelan untuk sistem-sistem atau perangkat lunak yang memiliki paradigma 'berorientasi objek'. UML itu sendiri digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sehingga mudah dipahami.

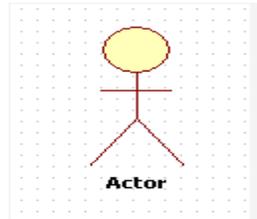
Hingga saat ini, UML terus berkembang dengan cepat. UML versi 2.0 mencakup 13 macam diagram dan perangkat yang berfungsi untuk menggambarkan sistem informasi berorientasi objek dengan sangat lengkap dan rinci.

2.8.1 Use Case Diagram

Menurut (Adi Nugroho, 2017) dalam buku "*Pemrograman Berorientasi Objek Menggunakan C#*". *Use Case Diagram* adalah diagram yang digunakan untuk mendeskripsikan bagaimana entitas eksternal akan menggunakan sistem atau perangkat lunak. Entitas eksternal itu bisa saja berupa manusia atau sistem yang lain dan sering dinamakan sebagai *actor*. Deskripsi *use case diagram* ini lebih menekankan pada sistem dari sudut pandang penggunaanya dan interaksi yang terjadi diantara pengguna dengan sistem.

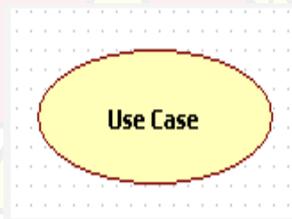
Komponen-komponen pembentuk diagram use case sebagai berikut :

1. Aktor : Merupakan pelaku eksternal yang berinteraksi dengan sistem untuk pertukaran informasi. Yang disimbolkan sabagai berikut :



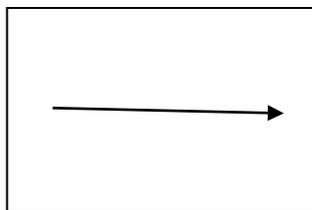
Gambar 2.5 Simbol Aktor (Adi Nugroho, 2017)

2. *Use Case* : Merupakan Urutan langkah-langkah yang secara tindakan saling terkait (skenario), baik terotomatisasi maupun secara manual, untuk tujuan melengkapi satu tugas bisnis tunggal.



Gambar 2.6 Simbol *Use Case* (Adi Nugroho, 2017)

3. Relasi : Merupakan garis atau simbol yang menghubungkan antara *element use case* dan aktor.



Gambar 2.7 Simbol Relasi (Adi Nugroho, 2017)

Relasi memiliki tipe relasi yang mungkin terjadi pada *diagram Use Case* :

1. <<include>> yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.
2. <<extends>> yaitu kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan alarm.

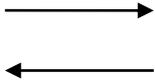
2.8.2 Activity Diagram

Menurut (Adi Nugroho, 2017) dalam buku "*Pemrograman BerorientasiObjek Menggunakan C#*". *Activity Diagram* adalah teknik yang menggambarkan aliran aktivitas-aktivitas yang harus terjadi selama berjalannya suatu operasi atau proses. Kita dapat dapat mengonstruksi *activity diagram* untuk melihat aliran kerja.

Activity diagram memberi tahu tentang apa yang terjadi, tetapi diagram ini tidak menyampaikan *class* mana yang bertanggung jawab untuk setiap action. Dalam pemodelan proses bisnis, hal ini tidak menyampaikan bagian mana dari sebuah organisasi yang melakukan *action* apa. Hal ini tidak menjadi masalah, seringkali, lebih masuk akal untuk berkonsentrasi pada apa yang telah dilakukan daripada siapa melakukan bagian apa dari *behavior* tersebut.

Tabel 2.3 Komponen - komponen pembentuk *Activity Diagram* (Adi Nugroho, 2017)

NO	SIMBOL	KETERANGAN
1		Menggambarkan awal sebuah proses
2		Menggambarkan sebuah kegiatan atau tugas yang perlu dilakukan

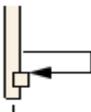
3		Menggambarkan sasaran yang mengawali kegiatan
4		Sebuah bar sinkronisasi. Simbol ini memperbolehkan anda untuk menggambarkan kegiatan yang dapat muncul secara paralel
5	[]	Menggambarkan sebuah sasaran yang merupakan sebuah hasil dari kegiatan keputusan
6		Menggambarkan sebuah kegiatan keputusan / tindakan yang harus diambil pada kondisi tertentu
7		Menggambarkan akhir dari sebuah proses

2.8.3 Sequence Diagram

Menurut (Adi Nugroho, 2017) dalam buku "*Pemrograman Berorientasi Objek Menggunakan C#*". *Sequence Diagram* menggambarkan objek-objek dari suatu kelas saling berinteraksi selama sistem atau perangkat lunak berjalan. Pada *Sequence Diagram*, objek-objek saling berinteraksi dengan saling mengirim pesan-pesan satu dengan yang lainnya.

Tabel 2.4 Komponen-komponen yang digunakan dalam *Sequence Diagram*.
(Adi Nugroho, 2017)

GAMBAR	KETERANGAN
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas

	<p>berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilanform entry dan form cetak.</p>
	<p><i>Control class</i>, Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri</p>
	<p><i>Activation</i>, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi</p>

	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i></p>
---	--

2.8.4 Deployment Diagram

Menurut (Ade Handini, 2016) dalam Jurnal Khatulistiwa Informatika Vo.1 IV No. 2 yang berjudul "*Pemodelan UML Sistem Informasi Monitoring Penjualan Dan Stok Barang*". Deployment Diagram digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem.

Deployment diagram adalah salah satu jenis alat atau bahasa (UML) yang digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis Object Oriented yang akan dibangun. Tujuan atau fungsi dari deployment diagram yaitu untuk menggambarkan/memvisualisasikan secara umum proses yang terjadi pada suatu sistem/software. Dalam deployment diagram terdapat simbol-simbol serta relasi yang sering digunakan yaitu :

1. Node digunakan untuk menggambarkan infrastruktur apa saja yang terdapat pada sistem. Biasanya node digambarkan sebagai server, pc, dan lain-lain.
2. Komponen digunakan untuk menggambarkan elemen-elemen apa saja yang terdapat pada suatu node.