

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Berdasarkan penelitian yang dilakukan oleh Fataruba (2018) mengenai Penerapan Metode *Cosine Similarity* Untuk Pengecekan Kemiripan Jawaban Ujian Siswa menyimpulkan bahwa metode *cosine similarity* dapat digunakan untuk pengecekan kemiripan jawaban ujian siswa.

Berdasarkan penelitian yang dilakukan oleh Hidayat dan Afuan (2018) mengenai Penilaian Ujian Otomatis Untuk Soal Bertipe *Essay* Pada PJJ APTIKOM Menggunakan *Cosine Similarity* menyimpulkan bahwa metode *cosine similarity* dapat digunakan untuk penilaian ujian secara otomatis untuk soal bertipe *essay*.

Berdasarkan penelitian yang dilakukan oleh Nurdiana, dkk (2016) mengenai Perbandingan Metode *Cosine Similarity* Dengan Metode *Jaccard Similarity* Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia menyimpulkan bahwa metode *cosine similarity* dapat digunakan untuk pencarian terjemahan Al-Qur'an.

Berdasarkan penelitian yang dilakukan oleh Pratama, dkk (2019) mengenai Deteksi Plagiarisme Pada Artikel Jurnal Menggunakan Metode *Cosine Similarity* menyimpulkan bahwa metode *cosine similarity* dapat digunakan untuk mendeteksi plagiarisme pada artikel jurnal.

Berdasarkan penelitian yang dilakukan oleh Rinarta (2017) mengenai Pemodelan Penilaian *Essay* Otomatis Secara *Realtime* Menggunakan Kombinasi *Text Stemming* Dan *Cosine Similarity* menyimpulkan bahwa metode *cosine similarity* dapat dikombinasikan dengan metode *text stemming* dan memiliki hasil penilaian *essay* yang baik.

Berdasarkan penelitian yang dilakukan oleh Wahyu, dkk (2017) mengenai Penerapan Algoritma *Cosine Similarity* Dan Pembobotan TF-IDF Pada Sistem Klasifikasi Dokumen Skripsi menyimpulkan bahwa metode *cosine similarity* dapat digunakan untuk pengecekan dokumen skripsi.

II.2. Landasan Teori

Berikut adalah beberapa landasan teori yang terdapat dari beberapa jurnal yang berkaitan dengan penelitian ini :

II.2.1. Ujian *Essay*

Ujian *essay* merupakan sebuah alat evaluasi dalam pendidikan untuk mengetahui seberapa mendalam tingkat pemahaman siswa dalam menyerap materi pada saat proses pembelajaran. Ujian *essay* pun dapat mengembangkan pemikiran kritis siswa terhadap persoalan yang akan dijawabnya melalui sebuah kalimat panjang. Kelemahan dalam implementasi ujian *essay* adalah sulitnya dalam penentuan skor, karena tidak sembarang orang bisa melakukan penilaian terhadap jawaban *essay*, umumnya, jawaban yang *essay* lebih bervariasi sehingga sulit dalam mencocokkan

dengan kunci jawaban. Bisa saja jawaban dari ujian *essay* tersebut benar, tetapi karena variasi jawaban, kata yang dituliskan bisa saja memiliki kesamaan arti dan makna, tetapi karena kunci jawaban berbeda penulisan kemungkinan jawaban *essay* tersebut tidak cocok sangat besar. Berbeda dengan ujian pilihan yang bisa dikoreksi oleh siapapun karena terdapat kunci jawaban yang sama persis. Disamping itu jumlah tenaga pengajar saat ini lebih sedikit dibandingkan dengan siswanya, sehingga untuk proses penilaian *essay* membutuhkan waktu dan tenaga yang besar. (Ekojono, dkk, 2020 : 73).

II.2.2. Metode *Cosine Similarity*

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada *vector space similarity measure*. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah *vector* dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran. (Nurdiana, dkk, 2016 : 60).

$$\frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i x B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \dots\dots\dots(1)$$

Keterangan :

A = vektor

B = vektor

A_i = bobot term i dalam blok A_i

B_i = bobot term i dalam blok B_i

i = jumlah term dalam kalimat

n = jumlah vector

Dimana A merupakan bobot setiap ciri pada vector A , dan B merupakan bobot setiap ciri pada B . jika dikaitkan dengan *information retrieval* maka A adalah bobot setiap istilah pada dokumen A , dan B merupakan bobot setiap istilah pada dokumen B . Pada penelitian ini digunakan *cosine similarity* karena citra merupakan salah satu data yang memiliki dimensi tinggi. Pada citra dapat dikatakan bahwa setiap *pixel* merupakan dimensi yang berbeda dan nilai warna pada setiap *pixel* tersebut merupakan nilai dari setiap dimensi. Pengukuran kemiripan dapat dilakukan dengan membandingkan dokumen 1 dengan dokumen 2 kemudian sistem akan menghitung nilai kemiripan. adalah nilai yang diperoleh dari term A dan term B kemudian kedua nilai tersebut dijumlahkan, kemudian nilai semua nilai term dokumen A semua nilainya dipangkatkan dua, begitu juga dengan *term* B_{i2} semua nilai yang diperoleh dipangkatkan dua kemudian semua nilai yang diperoleh dijumlahkan. (Fataruba, 2018 : 89).

II.2.3. Web

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses dimana saja dengan biaya yang relative murah. *Web* merupakan bentuk implementasi dari bahasa pemrograman *web* (*web programming*). Sejarah perkembangan bahasa pemrograman *web* diawali dengan munculnya *Hypertext*

Markup Language (HTML). Yang kemudian dikembangkan dengan munculnya *Cascading Style Sheet* (CSS) yang bertujuan untuk memperindah tampilan *website*. (Heriansyah, dkk, 2019 : 3).

World Wide Web (WWW) atau yang biasa disingkat dengan *web*, ini merupakan salah satu bentuk layanan yang dapat diakses melalui internet. *Web* adalah fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, bunyi, animasi, dan data multimedia lainnya, yang mana data tersebut saling berhubungan satu sama lainnya. (Heriansyah, dkk, 2019 : 3). *Website* (situs *web*) adalah merupakan alamat (URL) yang berfungsi sebagai tempat penyimpanan data dan informasi dengan berdasarkan topik tertentu. (Dewantara dan Nugrahanti, 2018 : 317).

Website merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *webpage* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hypertext*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* dapat diakses dan dibaca melalui *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi *browser* lainnya. (Damayanti, dkk, 2019 : 164).

II.2.4. *Hyper Text Markup Language* (HTML)

Hyper Text Markup Language (HTML) merupakan *file* teks atau *file ASCII* yang berisi instruksi/*script* kepada *web browser* untuk menampilkan suatu tampilan grafis dari sebuah halaman *web*. Pada file HTML ini terdapat “tag” atau kode-kode

yang dapat dimengerti nantinya oleh *web browser*. (Heriansyah, dkk, 2019 : 3).

II.2.5. Hypertext Preprocessor (PHP)

Hypertext Preprocessor (PHP) merupakan bagian dari bahasa pemrograman *web* atau *script-script* yang membuat dokumen HTML *secara on the fly* yang dieksekusi di *server web*. PHP merupakan sebuah bahasa *scripting* komputer, pada awalnya didesain untuk menghasilkan halaman *web* yang dinamis, secara umum digunakan sebagai *script server-side*, tetapi dapat digunakan dari *interface commandline* atau secara berdiri sendiri sebagai aplikasi grafis. (Heriansyah, dkk, 2019 : 3).

PHP merupakan singkatan dari *Hypertext Preprocessor* yaitu merupakan bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan script yang terintegrasi dengan HTML dan berada pada *server* (*server-side HTML embedded scripting*). PHP digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh pengguna, sehingga informasi atau konten yang didapatkan selalu *up to date*. Semua *script* PHP dieksekusi pada *server* di mana *script* tersebut dijalankan. (Romansyah, dkk, 2019 : 4).

II.2.6. Cascading Style Sheet (CSS)

Cascading Style Sheet(CSS) adalah suatu bahasa *stylesheet* yang mengatur tampilan suatu dokumen. Pada umumnya CSS digunakan untuk memformat halaman

webyang ditulis dengan HTML dan XHTML. Dengan CSS, tampilan *website* akan lebih cantik dan konsisten. (Heriansyah, dkk, 2019 : 3).

II.2.7. Basis Data

Basis data adalah suatu susunan atau kumpulan data operasional lengkap dari suatu organisasi/perusahaan yang diorganisir atau dikelola dan disimpan secara terintegrasi dengan menggunakan metode tertentu menggunakan komputer sehingga mampu menyediakan informasi optimal yang diperlukan pemakainya menurut Sistem Basis Data. (Dewantara dan Nugrahanti, 2018 : 317).

II.2.8. My Structure Query Language (MySQL)

My Structure Query Language (MySQL) adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS darisekian banyak DBMS, seperti Oracle, MS SQL, Postagre SQL, dan lain-lain. MySQL merupakan DBMS yang multithread, multi-user yang bersifat gratis di bawah lisensi GNU *General Public License* (GPL). Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan Swedia, yaitu MySQL AB. MySQL AB memegang hak cipta kode sumbernya. Dengan kombinasi antara pemrograman PHP dan MySQL dapat digunakan untuk mengembangkan sebuah aplikasi *web*. MySQL merupakan salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai

sumber dan pengolahan datanya. (Romansyah, dkk, 2019 : 4).

II.2.9. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak system operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP *Server*, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. (Dewantara dan Nugrahanti, 2018 : 317).

XAMPP merupakan sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MYSQL di komputer lokal". XAMPP berperan sebagai *server web* pada komputer lokal. XAMPP juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi *website* tanpa harus *online* atau terakses dengan internet. (Romansyah, dkk, 2019 : 4).

II.2.10. PHPMyAdmin

PHPMYAdmin adalah tools yang dapat digunakan dengan mudah untuk memanajemen *database* MySQL secara visual dan *Server* MySQL, sehingga kita tidak perlu lagi harus menulis *query* SQL setiap akan melakukan perintah operasi *database*. Tools ini cukup populer, anda dapat mendapatkan fasilitas ini ketika menginstal paket *triad* phpMyAdmin, karena termasuk dalam XAMPP yang sudah di *install*. (Damayanti, dkk, 2019 : 164).

II.2.11. Unified Modeling Language (UML)



Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak digunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2015 : 95).


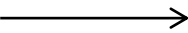
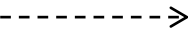

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan</p>



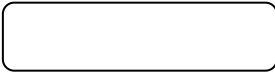
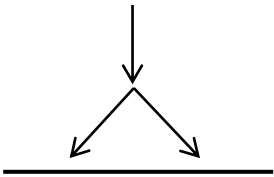
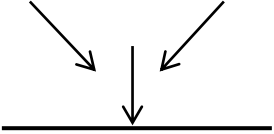
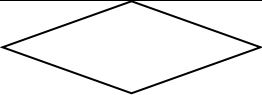
	<i>use case</i> , tetapi tidak memiliki <i>control</i> terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Urva dan Siregar, 2015 : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini :

Tabel II.2. Simbol *Activity Diagram*

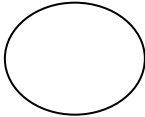
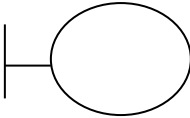
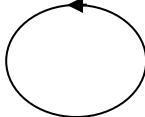

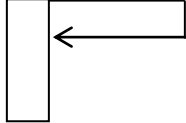
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true, false</i> .
New Swimline	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.


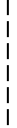
(Sumber : Urva dan Siregar, 2015 : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<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.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Urva dan Siregar, 2015 : 95)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih

1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar, 2015 : 95)