

BAB II

LANDASAN TEORI

2.1. Skripsi

Menurut Kamus Besar Bahasa Indonesia (KBBI) edisi V, skripsi adalah karangan ilmiah yang wajib ditulis oleh mahasiswa sebagai bagian dari persyaratan akhir pendidikan akademisnya. Tiap-tiap mahasiswa diharuskan menentukan fokus penelitian yang akan dilakukan dengan mencari topik penelitian yang diminati. Mahasiswa dapat melakukan pencarian topik penelitian tersebut melalui pembacaan jurnal penelitian yang sudah ada, baik dalam skala nasional, maupun internasional, turut serta mengikuti penelitian yang dilakukan oleh dosen, membaca kumpulan tugas akhir yang pernah dibuat oleh mahasiswa sebelumnya, serta melakukan observasi terkait fenomena atau masalah yang terjadi, baik di lingkup perguruan tinggi, maupun di luar perguruan tinggi.

2.2. Plagiarisme dan Plagiat

Menurut Kamus Besar Bahasa Indonesia (KBBI) edisi V, plagiarisme adalah penjiplakan yang melanggar hak cipta dan plagiat merupakan tindakan pengambilan karangan milik orang lain lalu menjadikannya seolah-olah merupakan karangan sendiri. Berdasarkan Pasal 1 Ayat 1 Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010 tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi menjelaskan bahwa plagiat adalah perbuatan secara sengaja atau tidak sengaja dalam memperoleh atau mencoba memperoleh kredit atau nilai untuk suatu karya ilmiah, dengan mengutip sebagian atau seluruh karya dan/atau karya ilmiah pihak lain yang diakui sebagai karya ilmiahnya, tanpa menyatakan sumber secara tepat dan memadai (Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010 Tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi, 2010).

Tindakan plagiat meliputi lima hal berikut ini dan perlakuannya tidak terbatas (Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010

Tentang Pencegahan Dan Penanggulangan Plagiat di Perguruan Tinggi, 2010). Lima hal tersebut adalah sebagai berikut.

- a. Mengacu dan/atau mengutip istilah, kata-kata dan/atau kalimat, data dan/atau informasi dari suatu sumber tanpa menyebutkan sumber dalam catatan kutipan dan/atau tanpa menyatakan sumber secara memadai.
- b. Mengacu dan/atau mengutip secara acak istilah, kata-kata dan/atau kalimat, data dan/atau informasi dari suatu sumber tanpa menyebutkan sumber dalam catatan kutipan/dan atau tanpa menyatakan sumber secara memadai.
- c. Menggunakan sumber gagasan, pendapat, pandangan, atau teori tanpa menyatakan sumber secara memadai.
- d. Merumuskan dengan kata-kata dan/atau kalimat sendiri dari sumber kata-kata dan/atau kalimat, gagasan, pendapat, pandangan, atau teori tanpa menyatakan sumber secara memadai.
- e. Menyerahkan suatu karya ilmiah yang dihasilkan dan/atau telah dipublikasikan oleh pihak lain sebagai karya ilmiahnya tanpa menyatakan sumber secara memadai.

Mahasiswa yang terbukti melakukan plagiarisme akan dikenakan sanksi, mulai dari teguran, peringatan tertulis, penundaan pemberian sebagian hak mahasiswa, pembatalan nilai satu atau beberapa mata kuliah yang diperoleh mahasiswa, pemberhentian dengan hormat dari status sebagai mahasiswa, pemberhentian tidak dengan hormat dari status sebagai mahasiswa, dan pembatalan ijazah apabila mahasiswa telah lulus dari suatu program (Peraturan Menteri Pendidikan Nasional Republik Indonesia Nomor 17 Tahun 2010 Tentang Pencegahan dan Penanggulangan Plagiat di Perguruan Tinggi, 2010).

2.3. Algoritma *Winnowing*

Winnowing merupakan sebuah algoritma yang digunakan untuk melakukan proses *document fingerprinting* (Schleimer dkk., 2003). *Document fingerprinting* merupakan sebuah metode yang digunakan untuk melakukan deteksi keakuratan salinan antar dokumen atau antar teks saja. Prinsip kerja dari metode ini adalah dengan menggunakan *hasing*. *Hashing* merupakan sebuah fungsi yang bertujuan

untuk mengubah setiap *string* menjadi bilangan. Proses ini ditujukan untuk dapat mendeteksi kemiripan di dalam dokumen yang berjumlah banyak. Adapun langkah-langkah penerapan algoritma *winnowing* adalah sebagai berikut.

1. Teks *preprocessing*

Merupakan proses pembuangan karakter yang tidak relevan dalam memenuhi kebutuhan algoritma *winnowing*, yaitu *whitespace*. Pada tahap ini juga melakukan penghapusan tanda baca, spasi, simbol-simbol. Lalu pada tahapan ini pula akan mengubah seluruh huruf menjadi huruf kecil.

2. Pembentukan *n-gram*

Pembentukan *n-gram* dilakukan dengan cara membentuk sebuah rangkaian karakter sepanjang *n* dari hasil pembuangan karakter yang tidak relevan. Contoh mempunyai sebuah kalimat “pembentukan ngram” dengan besaran *n* adalah 5 maka kalimat akan menjadi “pembe ntuka nngra m”.

3. Perhitungan *hash* untuk setiap *n-gram*

Perhitungan *hash* merupakan perhitungan dari nilai-nilai setiap *n-gram*. Fungsi yang digunakan adalah *rolling hash*. *Rolling hash* merupakan sebuah cara untuk mentransformasikan sebuah *string* menjadi suatu nilai unik yang memiliki panjang tertentu (*fixed length*) untuk menjadi penanda pada *string* tersebut.

$$Hash = c_1 * b^{(n-1)} + c_2 * b^{(n-1)} + \dots + c_n * b^{(n-1)} * b$$

4. Pembentukan *window* dari nilai *hash*

Pada tahapan ini merupakan tahapan untuk membentuk sebuah *window* berukuran *X* berdasarkan nilai *hash* yang telah dibentuk. *Window* pertama berisi nilai *hash* pertama sampai dengan nilai *hash* ke-*X*. *Window* kedua dibentuk dari nilai *hash* kedua sampai dengan nilai *hash* ke-*X*+1 dan begitu seterusnya hingga terbentuk *window* dari keseluruhan *hash*.

5. Pemilihan *fingerprint* dari setiap *window*

Pada tahapan ini merupakan menentukan nilai *fingerprint* teks. Nilai *fingerprint* ditentukan dengan cara memilih nilai *hash* terkecil dari setiap *window*.

6. Persamaan *jaccard coefficient*

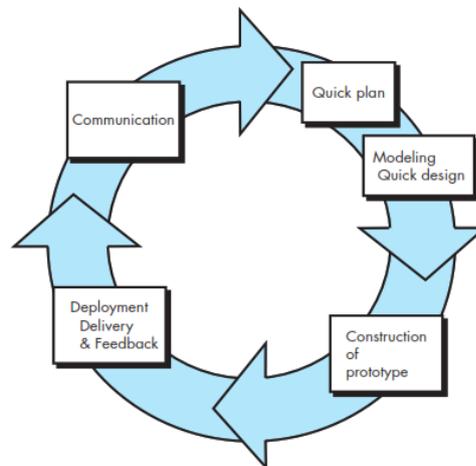
Nilai *fingerprint* yang telah terbentuk dengan algoritma *winning* akan digunakan untuk mengukur tingkat kemiripan dalam bentuk persentase menggunakan persamaan *jaccard coefficient*. Persamaan dari *jaccard coefficient* adalah sebagai berikut.

$$similarity = \frac{intersection}{(union - intersection)} \times 100\%$$

2.4. Metode *Prototype*

Pada penelitian ini penulis menggunakan metode pengembangan sistem dengan model *prototype*. *Prototype* adalah sebuah metode pengembangan perangkat lunak (*software*) yang banyak digunakan para *developer* (pengembang) perangkat lunak agar dapat saling melakukan interaksi dengan pengguna selama proses pembuatan sistem (Rusdiansyah, 2018).

Model *prototype* dapat dilihat pada gambar 2.2.



Gambar 2. 1 Metode *Prototype*

Berdasarkan gambar model *prototype* di atas, maka dapat diuraikan masing-masing pembahasan tahap di dalam model tersebut adalah sebagai berikut.

A. *Communication*

Komunikasi tim pengembang *software* melakukan pertemuan dengan para *stakeholder* untuk dapat menentukan kebutuhan *software* dan untuk menggambarkan area definisi untuk iterasi selanjutnya.

B. *Quick Plan*

Perencanaan Secara Cepat dalam pembuatan *prototype* dilakukan secara cepat. Setelah itu dilakukan pemodelan dengan bentuk rancangan cepat.

C. *Modelling Quick Design*

Model Rancangan Cepat menggunakan beberapa model yang berorientasi pada objek dengan menggunakan UML, yaitu *use case diagram* untuk mendefinisikan fungsi utama dari sistem, *class diagram* untuk menunjukkan kelas-kelas pada sistem, dan *activity diagram* untuk menggambarkan alur proses.

D. *Construction of Prototype*

Pembuatan Prototipe berdasarkan aspek-aspek perangkat lunak yang akan terlihat oleh *end user* (contohnya adalah rancangan antarmuka pengguna).

E. *Deployment Delivery & Feedback*

Penyerahan dan Pemberian Umpan Balik pengembangan perangkat lunak diserahkan kepada para *stakeholder* untuk dapat dievaluasi dan memberikan umpan balik yang akan digunakan untuk memperbaiki spesifikasi kebutuhan kedepannya.

2.5. Unified Modelling Language (UML)

Menurut (Booch G, 2005) dalam (Teguh Prihandoyo, 2018) *Unified Modelling Language* (UML) adalah metode pemodelan visual yang biasa digunakan dalam sebuah perancangan dan pembuatan sebuah perangkat lunak (*software*) yang berorientasi pada objek. UML juga merupakan sebuah standar penulisan *blue print* dimana di dalamnya terdapat bisnis proses, dan penulisan kelas dalam bahasa yang spesifik. Adapun diagram UML yang digunakan pada penelitian ini adalah sebagai berikut.

2.5.1. *Use Case Diagram*

Use Case Diagram merupakan sebuah gambaran dari fungsionalitas dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dengan sistem. Pada *use case diagram* terdapat *actor* yang merupakan gambaran dari entitas manusia

atau sebuah sistem yang melakukan pekerjaan di sistem tersebut. Adapun simbol-simbol yang digunakan pada *use case diagram* dapat dilihat pada tabel 2.1.

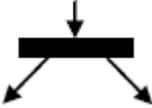
Tabel 2. 1 *Use Case Diagram*

No.	Gambar	Nama	Keterangan
1.		Actor	Merupakan himpunan peran pengguna ketika berinteraksi.
2.		Include	Merupakan hubungan perubahan yang terjadi pada suatu elemen mandiri (independent) dan akan mempengaruhi elemen yang tidak mandiri (non-independent).
3.		Assositaion	Merupakan penghubung antara objek satu dengan objek lainnya.
4.		Sistem	Merupakan paket yang menampilkan sistem secara terbatas.
5.		Use Case	Merupakan deskripsi dari urutan aksi yang ditampilkan pada sistem yang menghasilkan suatu hasil terukur bagi aktor.

2.5.2. *Activity Diagram*

Activity diagram merupakan gambaran dari aliran aktivitas-aktivitas didalam sistem yang berjalan. Adapun simbol-simbol yang digunakan pada *activity diagram* dapat dilihat pada tabel 2.2.

Tabel 2. 2 *Activity Diagram*

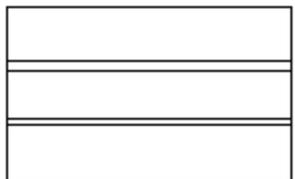
Simbol	Deskripsi
Status Awal 	Merupakan status awal dari aktivitas <i>diagram</i> .
Aktivitas 	Merupakan aktivitas sistem yang biasanya diawali dengan kata kerja.
Percabangan 	Merupakan percabangan jika ada aktivitas lebih dari satu.
Penggabungan 	Merupakan penggabungan yang dilakukan jika ada lebih dari satu aktivitas lalu akan digabungkan menjadi satu.
Status Akhir 	Merupakan status akhir dari sistem dan merupakan akhir dari suatu sistem aktivitas.
<i>Database</i> 	Merupakan simbol yang menunjukkan penyimpanan pada database sistem.
<i>Fork</i> 	Merupakan simbol yang menunjukkan aktivitas dilakukan secara paralel.

<p><i>Join</i></p> 	<p>Merupakan simbol yang menunjukkan aktivitas gabungan (aktivitas yang digabungkan).</p>
--	---

2.5.3. Class Diagram

Class diagram merupakan gambaran struktur dan deskripsi dari *class*, *package*, serta objek yang saling berhubungan. Contohnya adalah asosiasi, pewarisan, dan lain sebagainya. Adapun simbol-simbol yang digunakan pada class diagram dapat dilihat pada tabel 2.3.

Tabel 2. 3 *Class Diagram*

Simbol	Keterangan
<p><i>Class</i></p> 	<p>Merupakan simbol untuk menambahkan kelas baru pada diagram.</p>
<p><i>Assosiation</i></p> 	<p>Merupakan simbol yang menggambarkan relasi dari asosiasi.</p>
<p><i>Generalization</i></p> 	<p>Merupakan simbol yang menggambarkan relasi generalisasi.</p>
<p><i>Realize</i></p> 	<p>Merupakan simbol yang menggambarkan relasi realisasi.</p>
<p><i>Association Class</i></p> 	<p>Merupakan simbol yang berguna untuk menghubungkan kelas asosiasi pada suatu relasi asosiasi.</p>

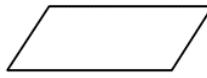
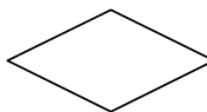
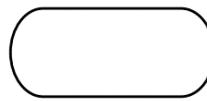
<i>Return Message</i> 	Merupakan simbol yang menggambarkan pengembalian dari pemanggilan prosedur.
<i>Aggregation</i> 	Merupakan simbol yang menggambarkan relasi agregasi.

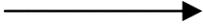
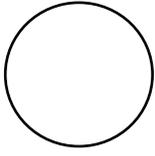
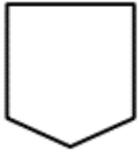
2.6. Flowchart

Flowchart merupakan cara penulisan dari algoritma dengan menggunakan sebuah notasi grafis. *Flowchart* berisi gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari sebuah program serta menggambarkan hubungan proses (Ridho Barakhbah dkk., 2013).

Simbol-simbol *flowchart* yang biasa dipakai adalah simbol standar yang dikeluarkan oleh *American National Standards Institute (ANSI)* dan *International Organization for Standardization (ISO)*. Simbol-simbol *flowchart* tersebut dapat dilihat pada tabel 2.4.

Tabel 2. 4 *Flowchart*

Simbol	Nama	Fungsi
	<i>Input / Output Data</i>	Pembacaan data (<i>read</i>) / penulisan (<i>write</i>).
	<i>Decision</i>	Kondisi yang menghasilkan 2 nilai, yaitu benar atau salah.
	<i>Terminator</i>	Simbol awal (<i>start</i>) / simbol akhir (<i>end</i>).

	<i>Flow Line</i>	Simbol aliran / penghubung/
	<i>Proses</i>	Perhitungan / pengolahan.
	<i>Preparation</i>	Pemberian nilai awal.
	<i>Predefined Process</i>	Menjalankan sub program / fungsi / prosedur.
	<i>On Page Connector</i>	Penghubung <i>flowchart</i> pada suatu halaman.
	<i>Off Page Connector</i>	Penghubung <i>flowchart</i> pada halaman yang berbeda.

2.7. Penelitian Terdahulu

Telah banyak peneliti sebelumnya yang telah melakukan penelitian mengenai pendeteksian plagiarisme judul skripsi. Berikut peneliti sajikan beberapa referensi dari penelitian terdahulu yang telah dilakukan sebelumnya dapat dilihat pada tabel 2.5.

Tabel 2. 5 Penelitian Terdahulu

No	Judul, Penulis, Tahun	Algoritma	Teks Preprocessing	Variabel
1.	Pengembangan Sistem Informasi Pengelolaan Judul Skripsi dan Tugas Akhir dengan Fitur Deteksi Kemiripan Menggunakan Algoritma <i>Winnowing</i> (Putra Bayu Pratama dkk., 2021).	<i>Winnowing</i>	<i>Case Folding, Text Cleaning (white space, punctuation, and special character).</i>	Judul Skripsi
2.	Penerapan Algoritma <i>Knuth-Morris-Pratt</i> dalam Mendeteksi Tingkat Kemiripan Judul Skripsi Berbasis Web (Suci Islamiyati & Fikri, 2022).	<i>Knuth-Morris-Pratt</i>	-	Judul Skripsi

3.	Analisis <i>String Matching</i> Pada Judul Skripsi Dengan Algoritma <i>Knuth-Morris Pratt</i> (KMP) (Astuti, 2017).	<i>Knuth-Morris-Pratt</i>	-	Judul Skripsi
4.	<i>Software Development</i> Dengan <i>Extreme Programming</i> (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android (Ahmad dkk., 2020).	<i>Boyer-Moore</i>	-	Judul Skripsi
5.	Deteksi Plagiarisme Tingkat Kemiripan Judul Skripsi Pada Fakultas Teknologi Informasi Menggunakan Algoritma <i>Winnowing</i> (Alamsyah & Rasyidan, 2019).	<i>Winnowing</i>	<i>Case Folding, Text Cleaning (white space, punctuation, and special character).</i>	Judul Skripsi
6.	Deteksi Tingkat Kemiripan Judul	<i>Oliver</i>	<i>Case Folding</i>	Judul Skripsi

	Menggunakan Algoritma <i>Oliver</i> Pada Sistem Informasi Pengajuan Skripsi (Fitrianingsih dkk., 2022).			
7.	Deteksi Kemiripan Judul Skripsi Menggunakan Algoritma <i>Levenshtein</i> <i>Distance</i> Pada Kampus STMIK MIC Cikarang (Harira Irawan dkk., 2021).	<i>Levenshtein</i> <i>Distance</i>	-	Judul Skripsi