BAB 2
LANDASAN TEORI

2.1 Plagiarisme

Plagiarisme merupakan sebuah praktik yang dapat merugikan akademik maupun personal dan dapat berdampak negatif terhadap perkembangan akademik maupun kesadaran diri manusia. Para akademisi berulang kali berupaya untuk mengatasi gangguan akademik tersebut. Salah satu upaya yang dilakukan oleh para akademisi maupun ahli sastra adalah memberikan sebuah pembelajaran dan definisi yang jelas pada teori plagiarisme, hal ini dilakukan agar mengurangi tindak plagiarisme yang terjadi di sekitar mereka [10].

Plagiarisme memiliki berbagai macam tipe, berikut adalah tipe-tipe plagiarisme [11].

1. Plagiarisme verbatim: Saat seseorang mengambil kata-kata orang lain secara tepat dan menyampaikannya atas nama mereka sendiri tanpa mengakui sumbernya secara terbuka, itu adalah bentuk umum dari plagiarisme verbatim. Sebagai contoh, menyalin dan menempelkan informasi dari sebuah artikel yang telah diterbitkan tanpa menyertakan referensi adalah contoh konkret dari tindakan plagiarisme yang sering terjadi.

2. Plagiarisme mosaik: Dalam bentuk plagiarisme ini, tidak ada penyalinan kata demi kata yang terjadi. Sebaliknya, hal ini melibatkan pencampuran ide dan pendapat orang lain dengan kata-kata yang dimodifikasi oleh penulis. Proses ini bisa diibaratkan sebagai menyalin dan menempel secara bertahap, dimana kata-kata asli dan kata-kata tambahan digabungkan untuk membentuk karya yang baru.

3. Plagiarisme parafrase: Dalam jenis plagiarisme ini, tidak ada penyalinan kata-kata secara langsung. Namun, melibatkan penggunaan ide dan pendapat orang lain dengan kata-kata yang disusun ulang oleh penulis dengan mengubah struktur bahasa.

4. Plagiarisme diri sendiri: Menggunakan kembali karya tulis atau karya ilmiahnya dalam bentuk yang sama dengan yang sebelumnya dan telah dipublikasikan.

5. Plagiarisme dunia maya: Ketika seseorang menyalin sebagian atau seluruh artikel atau makalah penelitian beserta ide-ide dari internet tanpa memberikan atribusi yang tepat.

6. Plagiarisme gambar: Tindakan plagiarisme melibatkan penggunaan gambar atau video tanpa mendapatkan izin yang diperlukan atau tanpa memberikan kutipan yang sesuai.

Plagiarisme juga mempunyai kriteria berdasarkan proporsi yang diplagiat, di antara lain [12].

1. Plagiarisme ringan: Plagiarisme yang bersifat ringan merujuk pada penjiplakan karya yang melibatkan jumlah kurang dari 30%. Jika jumlah penjiplakan melebihi batas tersebut, maka karya tersebut dapat dikategorikan sebagai plagiarisme. Selain itu, setiap negara juga memiliki batasan persentase kesamaan yang berlaku untuk setiap jurnalnya.

2. Plagiarisme sedang: Dalam plagiarisme sedang, terdapat kesamaan antara satu jurnal dengan jurnal lainnya sebesar 30% - 70%. Pada tingkat ini, tidak dapat lagi ditoleransi, dan persentase kesamaan tersebut dapat dikategorikan sebagai plagiat. Tindakan pengambilan karya yang melibatkan plagiarisme dalam tingkat ini dapat dilaporkan.

3. Plagiarisme berat: Plagiarisme berat terjadi ketika terdapat kesamaan yang sangat jelas dalam suatu karya dengan persentase di atas 70%. Jika tingkat kesamaan antara dua karya tulis melebihi 70%, maka karya tersebut dapat dengan jelas dikategorikan sebagai plagiat dan dapat dilaporkan kepada pihak yang berwenang.

2.2 Rabin-Karp

Algoritma Rabin-Karp digunakan secara umum untuk memecahkan masalah pencocokan *string* dengan menggunakan fungsi *hash*. Algoritma ini bekerja dengan mengoperasikan fungsi *hash* dan membandingkan *substring* dalam teks (n) dengan *string* yang sedang dicari (m). Jika kedua fungsi *hash* tersebut sama, maka proses perbandingan dilakukan lagi. Namun, jika fungsi *hash* tersebut berbeda, maka *substring* akan digeser ke kanan sejauh (n-m) kali. [13].

Rabin-Karp mewakili setiap karakter sebagai angka desimal. Pada sebuah kata ”ABC”, karakter ”A” dianggap nilai 0, karakter ”B” adalah nilai 1, dan ”C”

adalah nilai 2. *Hash value* sendiri dihitung pada pola dan *substring* pada teks. *Hash value* dapat dihitung menggunakan rumus berikut:

$$Hash\ value = (d^{m-1}.T[0]) + (d^{m-2}.T[1] + ... + (d^{0}.T[m]).modq \quad (2.1)$$

Keterangan :

d = basis 0,1,2,3,..., dst

T[i] = *array* teks

m = panjang pola

q = nilai *modulo*

Sumber : [14]

Ketika algoritma Rabin-Karp mencocokkan nilai *hash* ke *string* berikutnya dalam teks (digeser satu karakter ke kanan), ia menghitung dengan nilai *hash* yang ada alih-alih menghitung ulang nilai *hash* satu per satu dari awal.

$$t_{s+1} = (d(t_s - T[s+1]h) + T[s+m+1].modq \quad (2.2)$$

Keterangan :

t_s = *hash value* dengan panjang m dari *substring* T[s+1..s+m], untuk s=0,1,...,n-m

t_{s+1} = *hash value* selanjutnya yang dihitung dari t_s

d = radix desimal (bilangan basis 10)

h $= d^{m-1}$

n = panjang teks

m = panjang pola

q = nilai *modulo*

Sumber : [14]

Nilai *hash* akan dibandingkan, jika hasilnya sama maka pola ditemukan dan semua kesamaan akan dikonversi menjadi bilangan persen sebagai sebuah persentase kecocokan. Jumlah *dataset* yang akan dibuat ialah sebanyak dua ratus lima puluh *dataset* jurnal atau pun karya tulis lainnya, *dataset* ini nantinya dibandingkan dengan gambar ataupun *file* yang pengguna masukkan menggunakan metode *pattern matching* dan *string matching* dengan Rabin-Karp. Pembuatan dua ratus lima puluh *dataset* tersebut didasarkan oleh penelitian-penelitian sebelumnya

yang memakai jurnal atau karya lain. Pembuatan *dataset* dilakukan dengan cara *website scraping*. Pada penelitian yang dilakukan oleh Rubbo P, Helmann C, Bilynkievycz dos Santos C, berisikan pengambilan data *paper* dari *data set* jurnal sebanyak 117 sebagai basis data [15]. Jumlah jurnal tersebut akan menjadi landasan banyaknya *dataset* yang akan dibuat pada penelitian ini.

2.3 Optical Character Recognition (OCR)

OCR sendiri adalah sebuah teknologi yang digunakan untuk mengubah bentuk gambar menjadi teks, teknologi ini menggunakan sebuah kode untuk mengenali teks secara penuh [16]. Sistem OCR telah mengalami perkembangan sejak lama. Pada tahun 1914, Emanuel Goldberg memulai pengembangan sistem OCR untuk digunakan dalam telegram dan alat baca bagi orang tunanetra. Sejak saat itu, sistem OCR terus ditingkatkan hingga saat ini, sehingga dapat mencapai tingkat akurasi yang lebih baik, bahkan dalam situasi di mana karakter sulit dikenali [17]. Penggunaan OCR sendiri akan menggunakan bahasa pemrograman Python dan memakai *library* API Google Cloud Vision dalam penerapannya.

2.4 Google Cloud Vision

API Google Cloud Vision adalah suatu kerangka kerja machine learning yang dirancang oleh perusahaan Google untuk memproses data visual, seperti OCR (Optical Character Recognition), deteksi wajah, pencarian gambar, dan lain sebagainya. Seluruh dataset yang digunakan pada API ini telah dilakukan pra-pelatihan atau pemrosesan sebelumnya. [18]. Google Cloud Vision sendiri digunakan karena memiliki *plugin* yang dapat mengecek tulisan pada foto atau dokumen. Hasil teks yang didapatkan dari API tersebut, yang nantinya akan dipakai dalam penelitian.

2.5 React Native

React Native adalah sebuah kerangka kerja yang dikembangkan oleh para *developer* Facebook, dan digunakan sebagai kerangka kerja *Mobile-Cross Platform* yakni sebuah kerangka kerja yang dapat berjalan diberbagai perangkat. Perangkat yang tersedia untuk kerangka kerja tersebut cukup banyak, mulai dari Android, IOS, Web, Windows, bahkan macOS. React Native sendiri dapat memakai bahasa pemrograman JavaScript versi ES6 atau juga TypeScript untuk pengembangan

aplikasinya [19]. React Native ini akan dipakai sebagai kerangka kerja Android pada penelitian ini, dikarenakan memiliki *library* yang mendukung dalam proses penelitian.

2.6 Python

Python telah menjadi bahasa pemrograman pilihan untuk area penelitian *machine learning* dalam beberapa tahun terakhir, bukan hanya karena mudah digunakan tetapi juga adanya dukungan komunitas yang baik. Python adalah bahasa pemrograman tingkat tinggi, dengan sistem komputasi paralel, sintaks relatif sederhana, dan kode yang *readable* [20]. Pada Python sendiri sangat banyak *library* yang mendukung berjalannya *machine learning*, kecerdasan buatan dan juga *image processing* [21]. Proses pengecekan memakai bahasa pemrograman Python, hal ini dikarenakan Python memiliki *library* pendukung untuk membuat OCR.

2.7 Confusion Matrix

Confusion matrix adalah metode yang umum digunakan untuk menilai kinerja dari algoritma klasifikasi. *Confusion matrix* adalah cara standar untuk menunjukkan jumlah *True Positive* (TP), *False positive* (FP), *True Negatif* (TN), dan *False Negatif* (FN) agar lebih visual [22]. Tabel 4 merupakan tabel *confusion matrix*.

Tabel 2.1. Tabel *Confusion Matrix*

	Classifier says YES	Classifier says NO
In reality YES	True positives	False Negatives
In reality NO	False Positives	True negatives

Sumber: [23]

Confusion matrix memungkinkan untuk menentukan nilai dari *accuracy, precision, recall,* dan F1. Berikut adalah persamaan dari keempat nilai tersebut.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{2.3}$$

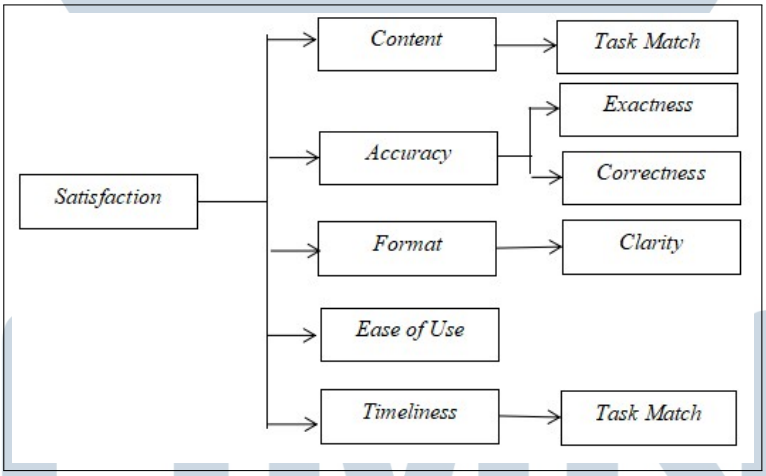
$$Precision = \frac{TP}{TP+FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall} \tag{2.6}$$

2.8 End User Computing Satisfaction (EUCS)

EUCS adalah sebuah model yang digunakan untuk mengukur tingkat kepuasan pengguna terhadap suatu sistem. EUCS adalah metode yang efektif terhadap seseorang yang secara langsung berinteraksi dengan aplikasi tersebut, [24]. Dimensi dalam metode EUCS telah dirumuskan oleh Doll dan Torkzadeh yaitu *content, accuracy, format, ease of use, timeliness*. Gambar 2.1 adalah gambaran dimensi metode pada EUCS.



Gambar 2.1. Model EUCS

Sumber: [25]

Penjelasan pada tiap dimensi pada metode EUCS:

a) Dimensi *Content*

Mengevaluasi tingkat kepuasan pengguna berdasarkan konten yang disediakan oleh aplikasi. Isi aplikasi mencakup fungsi dan modul yang tersedia bagi pengguna, serta informasi yang dapat diperoleh dari aplikasi tersebut.

b) Dimensi *Accuracy*

Mengukur kepuasan pengguna berdasarkan akurasi data saat sistem

menerima *input* dan mengubahnya menjadi informasi.

c) Dimensi Format
Mengukur kepuasan pengguna berdasarkan tampilan dan estetika antarmuka sistem, format laporan, atau informasi yang dihasilkan oleh aplikasi.

d) Dimensi *Ease of use*
Mengukur kepuasan pengguna berdasarkan tingkat kegunaan atau kemudahan pengguna saat menggunakan sistem, termasuk proses penginputan data, pemrosesan data, dan penggunaan aplikasi yang mudah.

e) Dimensi *Timeliness*
Mengukur kepuasan pengguna berdasarkan kecepatan aplikasi dalam menampilkan atau menyediakan data dan informasi yang dibutuhkan oleh pengguna sesuai dengan waktu yang diharapkan. [25].

2.9 Skala Likert

Skala Likert digunakan untuk menghasilkan nilai atau skor yang mencerminkan sifat individu seperti perilaku, pengetahuan, dan sikap. Skala Likert melibatkan beberapa pertanyaan dengan pilihan respon berupa lima titik untuk menilai atau memberikan skor terhadap perilaku individu. Pilihan respon tersebut meliputi Sangat Puas, Puas, Netral, Tidak Puas, dan Sangat Tidak Puas [26]. Skala Likert ini akan digunakan untuk kuesioner tingkat kepuasan pengguna atau konsumen. Contoh Penulisan skor dari Skala Likert dapat dilihat pada Tabel 2.2.

Tabel 2.2. Tabel Kepuasan Skala Likert

Skala	Skor
Sangat Puas	5
Puas	4
Netral	3
Tidak Puas	2
Sangat Tidak Puas	1

Sumber: [26]

Proses perhitungan, dilakukan mulai dari menghitung total skor dari data yang dikumpulkan. Setelah itu melakukan perhitungan skor tertinggi (Y) dan skor terendah (X). Lalu melakukan perhitungan persentase indeks atau hasil. Didapatkan

nilai interval yang sesuai dengan kategori skala, pada penelitian ini skala yang disebutkan ada lima kategori, maka interval akan dibagi menjadi lima bagian. Adapun rumus dalam perhitungan dalam Skala Likert yaitu:

$$\text{Total skor} = T \times Pn \tag{2.7}$$

Keterangan :

T = Total jumlah Responden yang memilih

Pn = Pilihan angka skor Likert

$$Y = \text{Skor tertinggi Skala Likert} \times \text{Jumlah responden} \tag{2.8}$$

$$X = \text{Skor terendah Skala Likert} \times \text{Jumlah responden} \tag{2.9}$$

Keterangan :

Y = Skor tertinggi

X = Skor terendah

$$\text{Indeks}(\%) = \frac{\text{Total skor}}{Y} \times 100 \tag{2.10}$$

Keterangan :

Indeks (%) = Persentase hasil

Y = Skor tertinggi

$$I = \frac{100}{\text{Jumlah skor dalam Skala Likert}} \tag{2.11}$$

Keterangan :

I = Interval

Sumber : [27]