BAB 2

TINJAUAN PUSTAKA

2.1 Auto Essay Scoring

Auto Essay Scoring (AES) adalah salah satu fitur yang menggunakan *Natural Language Processing* (NLP) yang berfungsi untuk memeriksa dan memberikan skor pada jawaban siswa yang berbentuk essay secara otomatis. AES sangat berguna untuk melakukan pemeriksaan dan pemberian skor dalam skala besar karena AES dapat melakukan pemeriksaan dengan cepat dibandingkan secara manual (Rajagede, 2021).

Sedangkan untuk *Natural Language Processing* (NLP) merupakan salah satu kecerdasan buatan yang dikembangkan untuk memahami bahasa natural dan memproses bahasa natural tersebut sehingga komputer dapat memberikan respon (Chandra et al., 2020).

Terdapat lima langkah dalam melakukan proses NLP yaitu :

a. *Lexical Analysis*, pada langkah ini NLP mengidentifikasi dan menganalisis teks atau kalimat menjadi kata per kata,
b. *Syntax Analysis*, menganalisis kata-kata dalam kalimat berdasarkan tata bahasa atau grammar,
c. *Semantic Analysis*, pada langkah tersebut dilakukan penentuan makna dari sebuah kata,
d. *Discourse Integration*, menganalisis kalimat sebelumnya yang akan mempengaruhi kalimat selanjutnya,
e. *Pragmatics Analysis*, merupakan proses penggalian makna atau informasi yang lebih dalam dari teks (Sari, 2018).

Bahasa natural adalah bahasa yang digunakan oleh manusia untuk berkomunikasi. Bahasa tersebut diterima oleh komputer dan diproses serta dipahami maksud dari pengguna terlebih dahulu sehingga dapat dipahami dengan baik oleh komputer. Penelitian ini menggunakan beberapa penelitian sebelumnya untuk dijadikan referensi dan terangkum pada tabel 2.1

Tabel 2.1 Penelitian Terkait

Penulis dan Tahun	Objek	Metode	Hasil
(Kavitha & Kumar 2018)	Membandingkan metode *Multi Layer Perceptron* (MLP) dan *Support Vector Regression* (SVR)	MLP dan SVR dengan kernel *Linear, Radial Basis Function* (RBF) dan *polynomial*	Hasilnya penggunaan SVR dengan kernel *polynomial* metode yang paling ideal dengan skor RMSE sebesar 0.01679
(Sang & Joon , 2019)	Research paper classification	TF-IDF dan Latent Dirichlet allocation (LDA)	Hasilnya F-score value menggunakkan Metode 3 (kombinasi TF-IDF dan LDA) lebih tinggi dibandingkan dengan metode lain sebesar 0.96
(Hendayanti et al., 2019).	Memprediksi Jumlah Kunjungan Wisatawan Domestik ke Bali	SVR	Hasil dari penelitian ini diperoleh bahwa nilai MAPE yang diperoleh sebesar 11,34% sedangkan menggunakan data testing nilai MAPE yang diperoleh 7,30%.
(Cahyono et al., 2019).	Memprediksi Indeks Harga Konsumen	SVR	Hasil uji coba diperoleh nilai terbaik yaitu MAPE=0.1716, dengan menggunakan Kernel Gaussian RBF
(Bahri & Wajhillah, 2020)	AES	TF-IDF	Penelitian ini mendapatkan rata – rata peningkatan sebesar 11,81%

			dengan dice similarity coefficient
(Alida & Mustikasari, 2020)	Prediksi nilai tukar rupiah berdasarkan US Dollar	SVR dengan kernel *linear*, *polynomial*, dan *radial basis function*	Hasilnya kernel RBF mendapatkan akurasi paling baik dengan nilai akurasi R2 = 95.94% dan RMSE = 1.25%.
(Yudhawan, 2020)	Peramalan Harga Saham Perusahaan Pertambangan di Indonesia	SVR	Hasilnya data saham PTBA dengan kernel RBF mendapatkan nilai akurasi terbaik 97,9%.
(Thamrin et al., 2021)	AES	SVM dengan TF-IDF dan LSA	Penelitian ini menunjukkan bahwa hasil klasifikasi *Support Machine Vector* (SVM) dengan TF-IDF menghasilkan RMSE terbaik sebesar 2.730
(Verdikha et al., 2021)	AES	*Support Vector Regression* (SVR), *Logistic Regression* (LR), dan *MLP Regression* (MLP-R)	Hasilnya penggunaan SVR mendapatkan nilai RMSE yang paling baik dengan skor 2.166

2.2 TF-IDF

TF-IDF telah banyak digunakan dalam bidang pencarian informasi dan *text mining*, menurut Ahmad et al., (2018) *Term Frequency* (TF) pada TF-IDF adalah frekuensi dari kemunculan sebuah term dalam dokumen yang bersangkutan, sedangkan *Inverse Document Frequency* (IDF) merupakan sebuah perhitungan dari bagaimana *term* didistribusikan secara luas pada koleksi dokumen yang bersangkutan, jadi *Term Frequency-Inverse Document Frequency* (TF-IDF) merupakan metode penggabungan dua konsep untuk perhitungan bobot, yaitu TF dan IDF, dibawah merupakan perhitungan umum IDF dan TF-IDF:

$$idf_j = log\left(\frac{N+1}{df_j+1}\right) + 1 \quad (2.1)$$

- N : jumlah semua dokumen dalam koleksi
- df_j : jumlah dokumen yang mengandung term

$$\text{TF-IDF} = \text{tf } x \text{ idf} \quad (2.2)$$

Tabel 2.2 Contoh data

d1	anak pria
d2	anak wanita
d3	anak pria wanita

Untuk memahami tentang TF-IDF disediakan tabel 2.2 yang menunjukkan contoh data yang akan digunakan untuk melakukan perhitungan TF-IDF, terlihat bahwa terdapat 3 dokumen yang memiliki label “d1”, “d2”, “d3” dengan fitur yaitu “anak”, “pria”, “wanita”.

Tabel 2.3 TF pada dokumen

TF	Anak	pria	wanita
d1	1	1	0
d2	1	0	1
d3	1	1	1

Tabel 2.3 menunjukan hasil dari TF yang terdapat pada ketiga dokumen, terlihat bahwa frekuensi kata “anak” muncul pada ketiga dokumen tersebut, dengan frekuensi kemunculan sebanyak 1 kali pada setiap dokumen, dan frekuensi kata “pria” hanya muncul pada dokumen 1 dan dokumen 3 dengan frekuensi kemunculan sebanyak 1 kali pada masing-masing dokumen, sedangkan frekuensi kata “wanita” muncul pada dokumen 2 dan dokumen 3 dengan frekuensi kemunculan sebanyak 1 kali pada masing-masing dokumen.

Tabel 2.4 Hasil IDF

IDF	Anak	pria	wanita
d1	1	1.287682	0
d2	1	0	1.287682
d3	1	1.287682	1.287682

Tabel 2.4 merupakan hasil IDF pada ketiga dokumen, yang menunjukkan bahwa kata “anak” pada setiap dokumen mendapatkan nilai 1, ini merupakan hasil dari perhitungan menggunakkan rumus pada IDF, begitu juga pada kata “pria” dan “wanita”. Misalnya untuk perhitungan IDF term “anak” pada dokumen 1 adalah sebagai berikut :

$$log\left(\frac{3+1}{3+1}\right)+1 = \log\left(\frac{4}{4}\right) = 0+1 \\ = 1$$

Tabel 2.5 Hasil TF-IDF

TF-IDF	Anak	pria	wanita
d1	1	1.287682	0
d2	1	0	1.287682
d3	1	1.287682	1.287682

Tabel 2.5 adalah hasil perhitungan TF-IDF, terlihat bahwa kata “anak” pada setiap dokumen mendapatkan hasil 1, dan kata “pria” dan “wanita” mendapatkan hasil 1,287… ini didapatkan dengan perhitungan menggunakkan

rumus TF-IDF, yaitu TF x IDF, untuk contoh perhitungan TF-IDF “anak” pada d1 adalah :

TF anak d1 = 1

IDF anak d1 = 1

TF x IDF = 1 x 1 = 1

2.3 Normalisasi TFIDF

Hasil TFIDF sebelumnya akan dilakukan proses normalisasi, Menurut Rusnedy et al., (2021) Normalisasi data adalah salah satu cara yang dilakukan untuk merubah data dengan cara mengubah jenjang nilai data ke rentang 0-1. Normalisasi data perlu dilakukan sebelum proses Data Mining, agar tidak ada parameter yang lebih mendominasi. Pada penelitian ini menggunakan L2-Norm dengan rumus :

$$v_{norm} = \frac{\vec{v}}{||v||_p} = \frac{\vec{v}}{\sqrt{{v_1}^2+{v_2}^2+\cdots+{v_n}^2}} \quad (2.3)$$

Keterangan:

$\vec{v}$ = Nilai vektor yang akan dinormalisasi

$\underset{\|v\|p}{\longrightarrow} = \vec{v}$ pada dokumen dengan nilai p = 2

Contoh perhitungan normalisasi akan diperlihatkan dibawah menggunakkan hasil TFIDF pada dokumen 1

$$v_{norm} = \frac{\vec{v}}{||v||_2} = \frac{[1, 1{,}28768, 0]}{\sqrt{1^2 + 1{,}28768^2 + 0^2}}$$

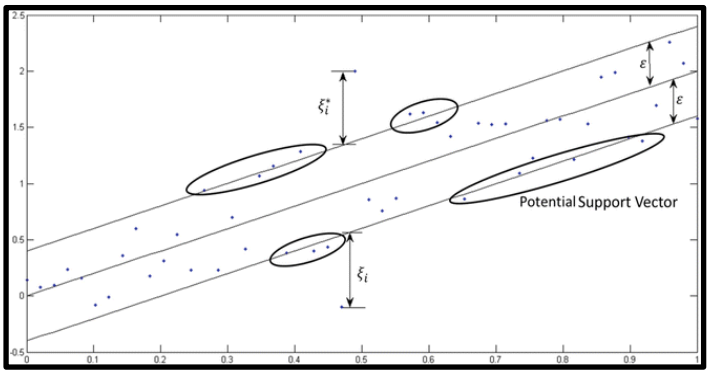
$$= \sqrt{1 + 1{,}65812} = \sqrt{2{,}65812} = 1.63037$$

$$= \frac{[1, 1{,}28768, 0]}{1.63037}$$

$$= [\,0.613355, 0.78980, 0]$$

2.4 Support Vector Regression

Support Vector Regression (SVR) adalah salah satu metode pada *machine learning* yang mengeluarkan output berupa bilangan riil dan kontinu, SVR ditemukan oleh Vladimir N. Vapnik pada tahun 1999 dan merupakan pengembangan dari *Support Vector Machine* (SVM) untuk mengatasi masalah regresi, yang bertujuan untuk mengecilkan persentase error atau selisih yang terdapat pada nilai prediksi dengan nilai sebenarnya, cara kerja SVR yaitu dengan mengecilkan persentase error dan memaksimalkan margin. Margin adalah jarak antara *hyperplane* dengan data terdekat. Data yang paling dekat dengan margin disebut *support vector* (Hendayanti et al., 2019). Sebagai ilustrasi SVR, dapat dilihat pada gambar 2.1 :



Gambar 2.1 Ilustrasi SVR (Yudhawan, 2020)

Garis yang berada di tengah pada gambar 2.1 adalah sebuah *hyperplane* yang diapit oleh dua garis batas. Pada gambar tersebut juga terlihat ada ε sebagai jarak antara *hyperplane* dengan 2 garis batas. Pada gambar tersebut ada beberapa titik-titik (*data points*) yang dilingkari, artinya titik-titik ini merupakan data poin yang bisa menjadi calon pembatas, sehingga semua data poin bisa masuk ke dalam satu kluster, dengan mempertahankan nilai ε tetap kecil (Yudhawan, 2020). Misalnya terdapat data training sebagai berikut :

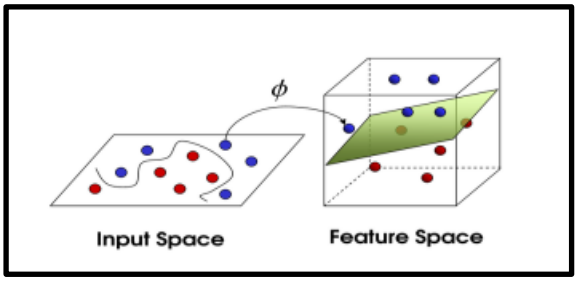
$$\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \subset \mathcal{X} \times \mathbb{R} \qquad (2.4)$$

Pada data training tersebut $\mathcal{X}$ merupakan *input vector* dari $\mathbb{R}$, y merupakan output dan l adalah banyaknya data. Penulis menggunakan metode SVR karena metode ini dapat mencari fungsi $f(x)$ atau *hyperplane* yang

mempunyai deviasi maksimal sebesar $\sum$ untuk mendapatkan nilai target yi dari semua data training, nilai kesalahan tidak diterima apabila nilainya melebihi $\sum$ (Cahyono et al., 2019)

2.5 Fungsi Kernel Pada SVR

Fungsi kernel dapat digunakan untuk membantu metode *Support Vector Regression* menyelesaikan permasalahan *non-linear*. Dengan menggunakan fungsi kernel, data yang awalnya tidak dapat dipisahkan secara *linier* akan dibawa ke dimensi yang lebih tinggi atau *feature space* sehingga pada dimensi yang baru, *hyperplane* dapat dibangun untuk memisahkan data *non-linear*. Gambaran penggunaan fungsi kernel pada data *non-linear* dapat dilihat pada gambar 2.2 :



Gambar 2.2 Fungsi kernel (Ningrum, 2018)

Dengan fungsi kernel, suatu data x di *input space* dibawa ke *feature space* dengan dimensi yang lebih tinggi melalui $\varphi: x \rightarrow \varphi(x)$ (Alida & Mustikasari, 2020). Dalam penelitian ini penulis akan menggunakkan kernel *polynomial*, kerena menurut Widayani (2021) *polynomial* merupakan fungsi kernel yang sangat cocok digunakan untuk permasalahan yang semua training dataset-nya dinormalisasi. Dibawah ini merupakan rumus penggunaan kernel *polynomial*

$$\kappa(x_i, x_j) = (x_i^T x_j + 1)^d \quad (2.5$$

2.6 Root Mean Square Error

Root Mean Square Error (RMSE) merupakan metode alternatif untuk mengukur tingkat kesalahan suatu model. Nilai yang dihasilkan RMSE merupakan nilai rata-rata kuadrat dari jumlah kesalahan pada model prediksi.

Menurut Sanjaya (2020) RMSE merupakan teknik yang banyak digunakan dalam berbagai penelitian untuk mengukur tingkat kesalahan dari suatu model. Untuk persamaan matematik *Root Mean Square Error* (RMSE) adalah sebagai berikut :

$$\sqrt{\sum_{i=1}^{n} \left(\frac{\widehat{y_i} - y_i}{n}\right)^2} \tag{2.6}$$

Keterangan:

$\widehat{y_i}$: Nilai hasil peramalan

y_i : Nilai aktual / Nilai sebenarnya

n : Jumlah data

i : Urutan data pada dokumen