

BAB II

LANDASAN TEORI

2.1 COVID-19

COVID-19 ialah penyakit menular yang berasal dari virus *novel-coronavirus* yang menginfeksi sistem pernafasan. Penyebaran virus ini bermula di Wuhan, China pada Desember 2019 [27], [28]. Penularan virus ini dapat melalui *droplet* yakni tetesan kecil, seperti partikel uap air yang dikeluarkan dari mulut selama batuk, bersin, atau berbicara yang dapat menularkan infeksi saat ditularkan melalui udara ke orang lain oleh penderita COVID-19. Oleh karena itu, kontak secara tidak langsung dengan menyentuh permukaan benda yang terkena *droplet* dan melakukan kontak sentuhan dengan mata, hidung atau mulut dapat menjadi media penularan COVID-19 [29]. Lapisan yang berada di dalam mata, hidung atau mulut yakni lapisan mukosa merupakan lapisan yang paling sesuai bagi virus untuk mulai menginfeksi tubuh [30]. Oleh karena itu, penggunaan masker wajah dan atau *face shield*, sangat disarankan untuk mencegah COVID-19 menginfeksi tubuh manusia. Sebab, orang yang menggunakan masker wajah mempunyai risiko yang lebih rendah untuk menderita penyakit COVID-19 [29], [31]. Di samping itu, menjaga kebersihan pribadi seperti sering mencuci tangan juga penting untuk dilakukan sebagai upaya pencegahan karena tangan memiliki frekuensi interaksi yang cukup tinggi terhadap benda seperti saat menyentuh. Terlebih, COVID-19 adalah jenis virus yang berukuran mikroskopis sehingga tidak dapat dilihat secara langsung oleh mata, maka prediksi terhadap keberadaan virus juga sulit untuk dilakukan [32].

Pada dasarnya, penderita COVID-19 memiliki sejumlah gejala yang sering kali sulit untuk dibedakan dengan penyakit influenza. Perbedaan dari COVID-19 dan influenza adalah pada frekuensi munculnya gejala, seperti gejala batuk yang terkadang terjadi pada penderita influenza, sedangkan gejala batuk selalu terjadi pada penderita COVID-19 [33]. Penyakit ini umumnya sangat rentan diderita oleh orang berusia lanjut dan orang dengan kondisi medis tertentu seperti mengidap

penyakit diabetes, ginjal, obesitas, dan lain-lain. Guna menekan penyebaran COVID-19, maka pemerintah membuat *rapid test* dan *PCR test* untuk mendeteksi sedini mungkin penyakit COVID-19. *Rapid test* merupakan tes yang dilakukan dengan cepat untuk mendeteksi kondisi antibodi dalam darah, sedangkan *PCR test* merupakan tes yang mendeteksi secara langsung eksistensi virus COVID-19 dalam darah sehingga menghasilkan tingkat akurasi yang lebih tinggi [34]. Di samping itu, tindakan pencegahan terhadap COVID-19 juga wajib untuk dilakukan seperti dengan mencuci tangan sesuai dengan enam langkah, menggunakan masker, konsumsi makanan bergizi, menghindari kontak fisik (*social distancing*), dan mematuhi protokol kebersihan yang telah ditetapkan oleh pemerintah saat berada di lingkungan eksternal (berada di luar rumah) [35]. Selain itu, juga dapat digunakan vaksin untuk mencegah penyebaran virus COVID-19 ini seperti dengan penggunaan vaksin Pfizer/BioNTech, Gamaleya, Moderna, AstraZeneca, Sinopharm, ataupun Sinovac [36].

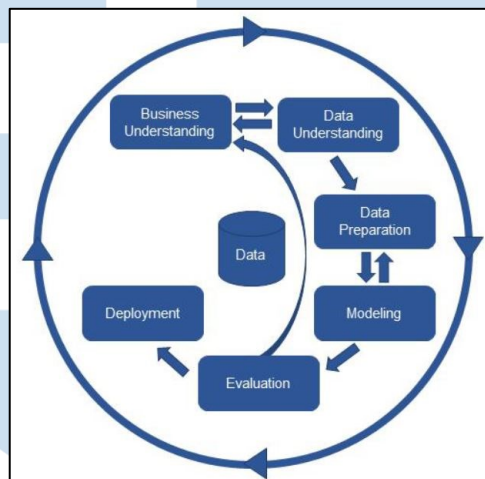
2.2 Hoaks

Hoaks adalah informasi salah atau informasi palsu yang dianggap benar sehingga menyebabkan terjadinya kesalahan dalam persepsi manusia [37], [38]. Berdasarkan definisi tersebut, maka terdapat dua kata kunci utama terkait dengan hoaks yaitu keaslian (*authenticity*) dimana hoaks dibuat atas informasi yang belum terverifikasi kebenarannya; dan niat (*intent*) dimana hoaks dibuat dengan niat untuk menyesatkan persepsi publik [11]. Penyebaran hoaks umumnya didasarkan atas sejumlah tujuan seperti untuk mempengaruhi atau memanipulasi opini publik [39]. Hoaks dapat dengan mudah tersebar dengan adanya penggunaan media sosial yang semakin masif saat ini [37].

2.3 CRISP-DM Framework

Cross Industry Standard Process – Data Mining (CRISP-DM) adalah model kerangka kerja mengenai proses independen dari penggunaan teknologi *data mining* di sektor industri dimana kerangka kerja ini memuat siklus hidup (*life cycle*) dari proyek *data mining* [40]. *CRISP-DM* pada dasarnya mengelaborasi sejumlah

langkah dari proposal *Knowledge Discovery in Databases (KDD)* menjadi enam langkah utama, antara lain pemahaman terhadap bisnis (*business understanding*), pemahaman terhadap data (*data understanding*), persiapan data (*data preparation*), pembangunan model (*modelling*), evaluasi hasil permodelan (*evaluation*), dan pengembangan model (*deployment*) [41]. Berikut ini adalah siklus *data mining* berdasarkan kerangka kerja *CRISP-DM*.



Gambar 2.1 Kerangka Kerja *CRISP-DM* [42]

Adapun penjelasan dari masing-masing tahapan dalam kerangka kerja *CRISP-DM* pada gambar 2.1 sebagai berikut [43].

- a. *Business understanding*
Business understanding merupakan tahapan pemahaman terhadap situasi bisnis baik mengenai sumber daya yang akan digunakan, jenis *data mining process* yang akan dilakukan, sampai dengan kriteria keberhasilan *data mining*.
- b. *Data understanding*
Data understanding merupakan tahapan pengumpulan, eksplorasi, dan deskripsi data yang dilakukan untuk memeriksa kualitas data. Umumnya, *data understanding* akan berkaitan dengan analisa statistik deskriptif.
- c. *Data preparation*

Data preparation ialah tahapan pemilihan data yang dilakukan untuk meningkatkan kualitas data yang akan digunakan dalam proses *modeling*. Umumnya, kualitas data yang buruk dapat ditangani dengan membersihkan data.

d. *Modeling*

Modeling adalah tahapan yang terdiri atas pemilihan teknik pemodelan, pembuatan *test-case*, dan *model* untuk menjawab *business understanding*.

e. *Evaluation*

Evaluation adalah proses pemeriksaan hasil pemodelan dengan mengacu pada *business understanding* yang telah ditetapkan.

f. *Deployment*

Deployment adalah tahapan dalam *data mining* yang berfokus pada implementasi dari model yang telah dibangun. *Deployment* umumnya memuat sejumlah sub-proses seperti proses *development*, *monitoring*, dan *maintenance*.

2.4 Metode Pengumpulan Data

Pada penelitian, data dapat dikumpulkan dengan menggunakan dua metode berikut.

a. Pengumpulan Data secara Primer

Pengumpulan data secara primer ialah proses pengumpulan data yang bersumber dari data asli oleh peneliti yang dikumpulkan untuk tujuan penelitian atau proyek tertentu [44]. Pengumpulan data secara primer dapat diperoleh dari aktivitas survey, eksperimen, observasi dan wawancara [45].

b. Pengumpulan Data secara Sekunder

Pengumpulan data sekunder ialah proses pengumpulan data dimana data yang diperoleh merupakan *output* dari hasil pengolahan data primer yang disampaikan dalam bentuk yang tidak sama dengan aslinya [46]. Pengumpulan data secara sekunder dapat diperoleh dari buku, artikel, jurnal, *scientific papers*, dan dokumen lainnya yang berkaitan dengan topik penelitian [45].

2.5 Data Pre-Processing

Tahapan *data pre-processing* memiliki peranan yang esensial dalam *proses text-mining* [47]. Adapun fokus utama dari adanya *data pre-processing* adalah melakukan *data cleansing* agar proses pengolahan data menjadi lebih optimal [48]. Pada umumnya, terdapat sejumlah proses yang menjadi bagian dari tahapan tersebut, antara lain sebagai berikut [49].

a. *Punctuation Removal*

Punctuation Removal adalah proses penghapusan seluruh *special characters* (\$,&, %, dan lain-lain), *emoticon*, serta *hyperlink* [49].

b. *Tokenizing*

Tokenization adalah proses pemecahan teks atau string input menjadi satuan kata penyusunnya yang disebut sebagai token [50].

c. *Case Folding*

Case folding adalah proses transformasi atau konversi dari karakter huruf kapital (*uppercase*) menjadi karakter huruf kecil (*lowercase*) [51].

d. *Stop word Removal*

Stop word removal adalah proses penghapusan token atau kata yang sering muncul dalam teks seperti kata hubung (dan, atau, ke, kemudian, dan lain-lain) [52].

Selain proses di atas, juga terhadap beberapa proses tambahan yang digunakan untuk melakukan tahapan *data pre-processing* khususnya untuk kasus dengan penggunaan algoritma *Convolutional Neural Network* dan *Bidirectional Long Short-Term Memory*. Proses tersebut adalah *padding* dan *truncation* yang merupakan proses untuk menstandarisasi panjang input *sequence* yang digunakan dalam *natural language processing* [53].

a. *Padding*

Terdapat dua jenis *padding* yaitu *pre-padding* dan *post-padding*. *Pre-padding* adalah *padding* yang dilakukan dengan mengisi seluruh *sequence* awal dengan nilai nol sehingga panjang *sequence* sesuai dengan panjang

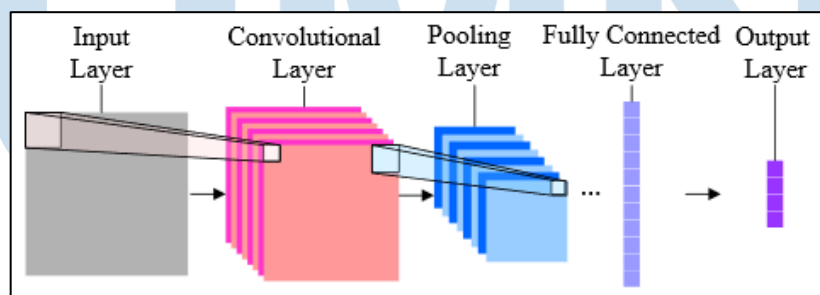
sequence terpanjang, sedangkan *post-padding* adalah *padding* yang dilakukan dengan mengisi seluruh *sequence* akhir dengan nilai nol sehingga panjang *sequence* sesuai dengan panjang *sequence* terpanjang [54].

b. *Truncation*

Terdapat dua jenis *truncation* yaitu *pre-sequence truncation* dan *post-sequence truncation*. *Pre-sequence truncation* merupakan *truncation* yang dilakukan dengan memotong seluruh bagian awal dari *sequence* sehingga panjang *sequence* sesuai dengan panjang dari *sequence* terkecil, sedangkan *post-sequence* merupakan *truncation* yang dilakukan dengan memotong seluruh bagian akhir dari *sequence* sehingga panjang *sequence* sama dengan panjang dari *sequence* terkecil [54].

2.6 Algoritma *Convolutional Neural Network*

Convolutional Neural Network (CNN) ialah versi regularisasi multi-layer *perceptron* yang diklasifikasikan sebagai *deep feed-forward artificial neural network* yang mengombinasikan beberapa lapisan pemrosesan dengan menggunakan beberapa elemen paralel serupa dengan visi manusia [55]. Pemrosesan *Convolutional Neural Network* umumnya ditemukan dalam bidang visi komputer, pemrosesan bahasa, dan pengenalan suara [56]–[59]. Adapun struktur umum dari *Convolutional Neural Network* sebagaimana yang direpresentasikan pada gambar 2.2.



Gambar 2.2 Struktur dari *Convolutional Neural Network* [60]

Input layer menerima gambar yang telah ternormalisasi dengan ukuran yang identik. Sebuah unit dalam *local receptive field* kemudian akan diproses oleh

convolution kernel untuk membentuk *feature map*. Satu piksel dalam *feature map* dapat dikalkulasikan dengan menggunakan rumus 2.1 sebagai berikut.

$$C_k = f(x \times W + b)$$

Rumus 2.1 Rumus Perhitungan *Feature Map*

dimana C_k adalah nilai dari k -th piksel dalam *feature map*, x adalah vektor pixel-value dari unit dalam *local receptive field*, W adalah koefisien dari vektor, b adalah bias, dan f adalah fungsi aktivasi yang dapat berupa *sigmoid*, *tanh*, *ReLU*, dan lain-lain. Untuk mencapai *invariance* dari *feature* berkaitan dengan *geometrical shift* dan *distortion*, *convolutional layer* disertai dengan *polling layer* untuk melakukan *subsample* terhadap *feature maps* [56]. Untuk k -th unit dalam *polling layer*, dapat dikalkulasikan dengan menggunakan rumus 2.2.

$$P_k = f(\beta \times \text{down}(C) + \alpha)$$

Rumus 2.2 Rumus Perhitungan *Polling Layer*

dimana P_k adalah nilai dari k -th unit dalam *feature map* yang terdapat dalam *polling layer*, C adalah nilai dari vektor dalam *feature map* yang terdapat dalam *convolutional layer*, β adalah nilai dari koefisien, α merupakan bias, dan $\text{down}(C)$ adalah fungsi *subsampling*. Adapun rumus yang digunakan untuk mengkalkulasikan $\text{down}(C)$ adalah rumus 2.3 sebagai berikut.

$$\text{down}(C) = \max \{C_{s,l} \mid |s| \leq \frac{m}{2}, |l| \leq \frac{m}{2}, s, l \in \mathbb{Z}^+\}$$

Rumus 2.3 Rumus Perhitungan *Down (C)*

dimana $C_{s,l}$ adalah nilai dari piksel dalam unit C dalam *feature map* dan m adalah ukuran dari *subsampling*. *Convolutional* dan *pooling layers* mengekstrak *elemental features*. Setiap unit dalam *pooling layer* terakhir akan dikoneksikan sebagai input dalam *fully-connected layer* yang berperan sebagai *hidden layer* dalam normal neural network. *Fully-connected layer* akan disertai dengan *output layer* dengan menggunakan *softmax connection* [60]. Probabilitas dari *softmax regression classifying* input dari vektor F yang berasal dari *layer* sebelumnya ke dalam grup c dapat dikalkulasikan dengan rumus 2.4 sebagai berikut.

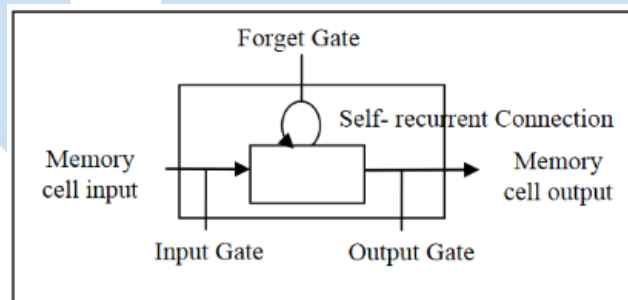
$$p(y^{(F)} = c|F; \theta) = \frac{e^{\theta_j^T F}}{\sum_{n=1}^N e^{\theta_n^T F}} \quad 1 \leq j \leq N$$

Rumus 2.4 Rumus Perhitungan Probabilitas dari *Softmax Regression*

dimana $y^{(F)}$ adalah kelompok identitas dari input F , θ adalah vektor bobot antara *output layer* dan *previous layer*, dan N adalah jumlah dari grup.

2.7 Algoritma *Bidirectional Long Short-Term Memory*

Long Short-Term Memory adalah hasil modifikasi terhadap algoritma *Recurrent Neural Network* dengan adanya ketergantungan pembelajaran jangka panjang (*long-term learning*) [26]. Gambar 2.3 adalah struktur umum dari algoritma *Long Short-Term Memory*.



Gambar 2.3 Struktur dari *Long Short-Term Memory* [26]

Long Short-Term Memory pada dasarnya terdiri atas *Input Gates (IG)*, *Output Gates (OG)*, dan *Forget Gates (FG)* dimana masing-masing *gates* mengontrol alur informasi dari *Candidate Hidden State (CHS)*, *Current State (CS)*, dan *Hidden Sequence (HS)* sebagaimana direpresentasikan dalam Rumus 2.5, Rumus 2.6, Rumus 2.7, Rumus 2.8, Rumus 2.9, dan Rumus 2.10. Umumnya, *Long Short-Term Memory* memproses informasi secara satu arah baik dari kiri ke kanan maupun dari kanan ke kiri. Hal tersebut dapat mengakibatkan timbulnya masalah pada informasi yang dihasilkan di masa depan. Guna mengatasi permasalahan tersebut, maka digunakan *Bidirectional Long Short Term Memory* yang mempelajari *original input sequence* dari awal hingga akhir dan dari akhir ke awal *sequence* dalam waktu bersamaan sebagaimana dalam Rumus 2.11.

$$IG_t = \sigma(W_{IG}X_t + R_{IG}h_{t-1} + b_{IG})$$

Rumus 2.5 Persamaan *Input Gates*

$$OG_t = \sigma(W_{OG}X_t + R_{OG}h_{t-1} + b_{OG})$$

Rumus 2.6 Persamaan *Output Gates*

$$FG_t = \sigma(W_{FG}X_t + R_{FG}h_{t-1} + b_{FG})$$

Rumus 2.7 Persamaan *Forget Gates*

$$CHS_t = \tanh(W_{CHS}X_t + R_{CHS}h_{t-1} + b_{CHS})$$

Rumus 2.8 Persamaan *Candidate Hidden States*

$$CS_t = FG_t \otimes CS_{t-1} + IGT_t \otimes HS_t$$

Rumus 2.9 Persamaan *Current State*

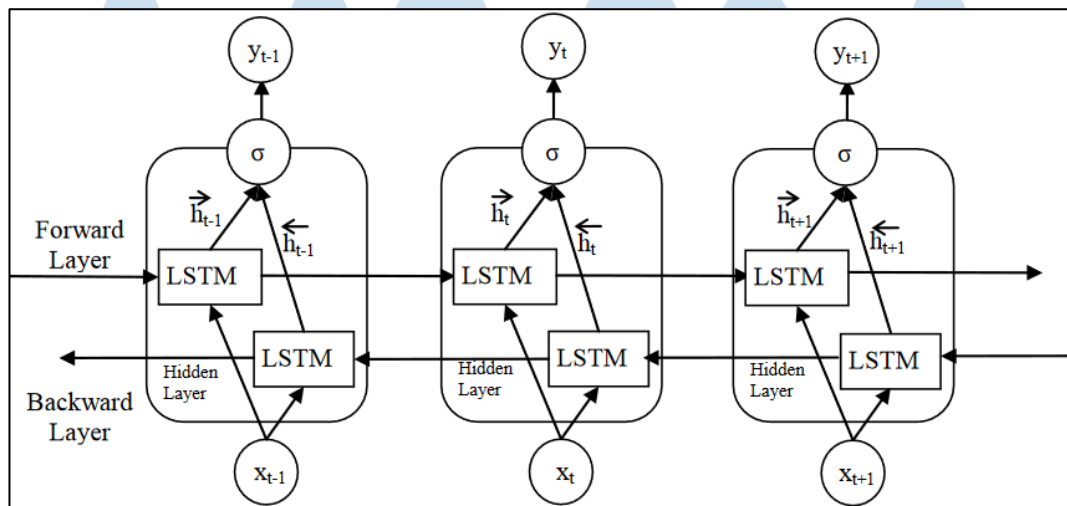
$$HS_t = OG_t \otimes \tanh(CS_t)$$

Rumus 2.10 Persamaan *Hidden Sequence*

$$Y_t = V(\vec{HS}_t; \overleftarrow{HS}_t)$$

Rumus 2.11 Persamaan *Bidirectional Long Short-Term Memory*

Dimana W_{IG} , W_{OG} , W_{FG} , dan W_{CHS} merujuk pada bobot dari matriks *input* x_t , R_{IG} , R_{OG} , R_{FG} , dan R_{CHS} yang dirujuk oleh bobot dari matriks *input* sebelumnya h_{t-1} , b_{IG} , b_{OG} , b_{FG} , dan b_{CHS} yang direpresentasikan sebagai nilai bias. Y_t merepresentasikan *output* dari *forward Long Short-Term Memory* dan *backward Long Short-Term Memory units* [26]. Adapun struktur umum dari *Bidirectional Long Short-Term Memory* adalah dalam Gambar 2.4.



Gambar 2.4 Struktur *Bidirectional Long Short-Term Memory* [26]

NUSANTARA

2.8 Performance Metrics

Performance metrics digunakan untuk mengevaluasi kinerja dari algoritma yang digunakan [61]. Umumnya akan digunakan *confusion matrix* yakni representasi tabular dari performa model klasifikasi pada data *testing*. *Confusion matrix* terdiri atas empat parameter sebagaimana yang terlihat dalam Gambar 2.5.

		Actual Class (Observation)	
		Y	N
Predicted class (expectation)	Y	TP correct result	FP unexpected result
	N	FN missing result	TN correct absence of result

TP, true positive; FP, false positive; FN, false negative; TN, true negative.

Gambar 2.5 *Confusion Matrix* [62]

2.8.1 Accuracy

Accuracy adalah metrik yang merepresentasikan persentase dari observasi yang diprediksi secara benar baik itu *true positive* ataupun *false negative* dibandingkan dengan seluruh data [61]. Untuk mengkalkulasikan metrik *accuracy* dapat digunakan rumus 2.12 sebagai berikut.

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

Rumus 2.12 Rumus Perhitungan Nilai *Accuracy* [61]

Pada kebanyakan kasus, tingginya nilai akurasi akan merepresentasikan bahwa model memiliki kinerja yang baik.

2.8.2 Recall

Recall atau *sensitivity* merepresentasikan rasio prediksi *true positive* dibandingkan dengan jumlah keseluruhan data *positive* [61]. Untuk menghitung nilai dari *recall*, digunakan rumus 2.13 sebagai berikut.

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

Rumus 2.13 Rumus Perhitungan Nilai *Recall* [61]

2.8.3 Precision

Precision adalah metrik yang merepresentasikan rasio dari *true positives* dibandingkan dengan seluruh observasi yang bernilai *positive* [61]. Kalkulasi nilai *precision* dapat dilakukan dengan menggunakan rumus 2.14.

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

Rumus 2.14 Rumus Perhitungan Nilai *Precision* [61]

2.8.4 F1-Score

F1-score merepresentasikan gabungan antara nilai *precision* dan *recall*. *F1-score* mengkalkulasikan nilai rata-rata (*mean*) antara kedua nilai tersebut [61]. Perhitungan terhadap nilai *F1-score* dilakukan dengan rumus 2.15.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Rumus 2.15 Rumus Perhitungan Nilai *F1-Score* [61]

2.9 Penelitian Terdahulu

Adapun penelitian sebelumnya terkait dengan deteksi hoaks yang dijadikan acuan pada penelitian ini yaitu pada Tabel 2.1.

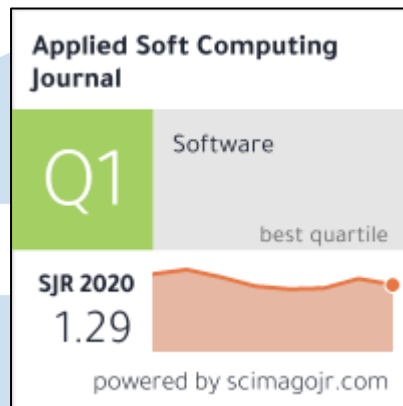
Tabel 2.1 Perbandingan Penelitian Sebelumnya

Nama Artikel (Vol, Tahun)	Nama Penulis	Masalah	Metode	Hasil
“Cross-SEAN: A Cross-Stitch Semi-Supervised Neural Attention Model For COVID-19 Fake News Detection” (Vol. 107, 2021) [63]	(William Scott Paka, Rachit Bansal, Abhay Kaushik, Shubhashis Sengupta)	Adanya pandemi COVID-19, disertai dengan adanya <i>fake news</i> di media sosial menyebabkan kebingungan dalam masyarakat, dan bahkan mampu membahayakan hidup seseorang.	Dengan menggunakan <i>cross-stitch based semi-supervised end-to-end neural attention model</i> terhadap <i>dataset CTF</i>	Penggunaan <i>Cross SEAN</i> mampu mencapai <i>F1-score</i> sebesar 0.95, mengungguli metode baseline terbaik yakni model <i>MixText</i> sebesar 9%.
“Detection of Fake News using Deep Learning CNN-RNN based Methods” (2021) [25]	(I. Kadek Sastrawan, I.P.A Bayupati, Dewa Made Sri Arsa)	Penyebaran berita palsu dapat merugikan dunia politik dan sosial. Terlebih, ketidakmampuan masyarakat	Mengimplementasikan model <i>CNN</i> dengan <i>fastText word embedding model</i> , model <i>ResNet</i> dengan	<i>Bidirectional LSTM</i> dengan <i>GloVe word embedding model</i> memiliki performa terbaik dengan nilai

Nama Artikel (Vol, Tahun)	Nama Penulis	Masalah	Metode	Hasil
		dalam membedakan berita asli dan palsu juga menjadi salah satu problematika yang juga berperan dalam penyebaran berita palsu.	<i>GloVe word embedding model</i> , dan <i>Bidirectional LSTM</i> dengan <i>GloVe word embedding model</i> terhadap <i>ISOT Fake News Dataset</i> , <i>Fake News Dataset</i> , <i>Fake or Real News Dataset</i> , dan <i>Fake News Detection Dataset</i>	<i>accuracy</i> sebesar 99.95%, nilai <i>precision</i> sebesar 99.95%, nilai <i>recall</i> sebesar 99.95%, dan nilai <i>f1-score</i> sebesar 99.95%.
“Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches” (Vol. 2021, 2021) [64]	(Waqas Haider Bangyal, Rukhma Qasim, Najeeb ur Rehman, Zeehshan Ahmad, Hafsa Dar, Laiqa Rukhsar, Zahra Aman, Jamil Ahmad)	Sulitnya proses <i>capturing</i> , <i>gathering</i> , dan <i>analyzing</i> opini masyarakat khususnya pada media sosial yang memiliki data berskala besar.	Menerapkan algoritma <i>Logistic Regression</i> , <i>Random Forest</i> , <i>Decision Tree</i> , <i>SVM</i> , <i>KNN</i> , <i>Adaboost</i> , <i>MLP/BPA</i> , <i>Naive Bayes</i> , <i>CNN</i> , <i>LSTM</i> , <i>RNN</i> , dan <i>GRU</i> terhadap <i>COVID Fake News Dataset</i>	Model CNN dan BiLSTM memiliki tingkat akurasi terbaik yakni sebesar 97% dibandingkan dengan <i>classifiers</i> lainnya.

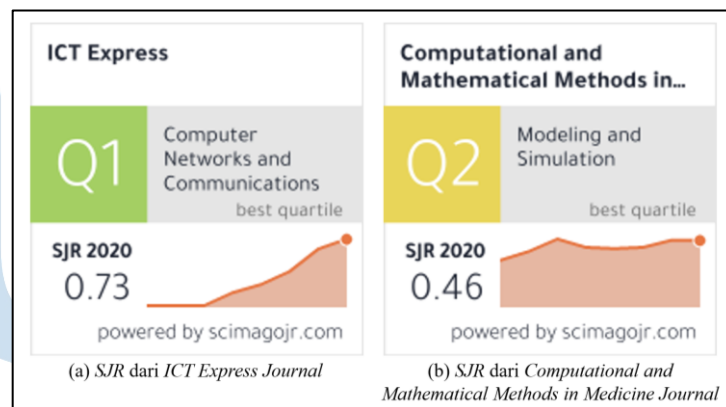
Penelitian ini menggunakan *dataset* yang sama dengan yang digunakan dalam artikel “Cross-SEAN: A Cross-Stitch Semi-Supervised Neural Attention Model for COVID-19 Fake News Detection” yang merupakan bagian dari jurnal “Applied Soft Computing” yang memiliki *SCImago Journal Rank (SJR) Q1* sebagaimana tertera dalam Gambar 2.6 [63].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2.6 SJR dari *Applied Soft Computing Journal*

Akan tetapi, penelitian ini mengadopsi algoritma yang berbeda. Penelitian ini mengadopsi algoritma dengan tingkat akurasi tertinggi terhadap objek serupa yakni hoaks atau *fake news* berdasarkan artikel “Detection of Fake News using Deep Learning CNN-RNN based Methods” dari jurnal “ICT Express” yang memiliki *SCImago Journal Rank (SJR) Q1* dan artikel “Fake News Detection using Bi-directional LSTM-Recurrent Neural Network” dari jurnal “Computational and Mathematical Methods in Medicine” dengan *SCImago Journal Rank (SJR) Q2* sebagaimana dalam Gambar 2.7 [65], [66].



Gambar 2.7 SJR dari Dua Jurnal Terdahulu

Adapun perbedaan penelitian ini dengan penelitian dalam jurnal “ICT Express” adalah penelitian ini menggunakan objek yang berbeda yakni hoaks terkait COVID-19, sedangkan penelitian dalam jurnal “ICT Express” menggunakan hoaks atau *fake news* yang tidak terkait dengan COVID-19, seperti hoaks dalam bidang politik dan

sosial. Sementara itu, perbedaan penelitian ini dengan penelitian dalam jurnal “Computational and Mathematical Methods in Medicine” adalah penelitian ini menggunakan data hoaks COVID-19 pada media sosial Twitter berupa *tweet* sebagai objek penelitian, sedangkan penelitian dalam jurnal “Computational and Mathematical Methods in Medicine” menggunakan data hoaks atau *fake news* dari media sosial Facebook, Instagram, dan media sosial lainnya berupa *headlines* sebagai objek penelitian.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA