

BAB II

KAJIAN LITERATUR

2.1 Tinjauan Pustaka

Bagian ini berisi landasan teori terkait teori-teori yang digunakan untuk mendukung penyelesaian penelitian yang akan dilakukan dan pekerjaan yang sudah dilakukan oleh penelitian sebelumnya.

Pembahasan dimulai dengan pengenalan ujaran kebencian pada subbab 2.1.1. Pada bagian subbab 2.1.2 akan membahas mengenai ujaran kebencian di *Twitter*. Pada bagian 2.1.3 akan membahas *pre-processing* untuk membersihkan data sesuai dengan informasi yang dibutuhkan dalam dataset. Pada subbab 2.1.4 akan membahas *Recurrent Neural Network* (RNN), kemudian dilanjutkan pada subbab 2.1.5 tentang *Long Short-Term Memory* yang akan digunakan untuk mendeteksi ujaran kebencian berdasarkan dataset yang telah dibersihkan. Kemudian akan membahas metode penilaian kata yang digunakan TF-IDF pada subbab 2.1.6, dan pada subbab 2.1.7 akan membahas *Confusion Matrix* yang akan digunakan untuk melakukan evaluasi terhadap model.

2.1.1 Ujaran Kebencian

Ujaran kebencian adalah salah satu kejahatan yang dilakukan ketika seseorang menggunakan kata-kata atau tulisannya di depan umum untuk menghasut atau menyebarkan kebencian. Ujaran kebencian digambarkan sebagai komunikasi yang dirancang untuk menghina, menyinggung, atau menakut-nakuti seseorang karena beberapa sifat (ras, agama, orientasi seksual, asal kebangsaan, dan/atau kecacatan fisik) [10]. Fortuna, A. dan Nunes, S., [11] berpendapat bahwa ujaran kebencian terhadap suatu kelompok berdasarkan ciri-ciri, seperti penampilan fisik, agama, garis keturunan, asal kebangsaan atau etnis, orientasi seksual, dan identitas gender. Ujaran kebencian dapat mengambil banyak bentuk linguistik yang berbeda, bahkan dengan gaya bahasa yang halus maupun berkonotasi lucu.

Hate Speech berbeda dengan *Free Speech*. Pembedaan ini didasarkan pada apakah suatu ucapan dimaksudkan menyerang atau menyakiti secara langsung atau tidak langsung pada pendengar. Kebebasan berbicara menunjukkan bahwa pendapat yang membangun diungkapkan secara bijaksana dan bertanggung jawab. Keberhasilan ujaran kebencian tergantung pada apakah ujaran yang dilakukan secara berapi-api itu benar-benar dapat

membujuk audiens untuk menggunakan kekerasan atau melukai individu atau kelompok lain [12].

Disahkannya Surat Edaran Kapolri Publik Nomor 6 Tahun 2015 tentang ujaran kebencian menunjukkan bahwa ujaran kebencian dapat masuk dalam kategori delik yang diatur dalam Kitab Undang-Undang Hukum Pidana (KUHP) dan tindak pidana lain di luar KUHP, yang dapat berbentuk penghinaan, pencemaran nama, penghasutan, provokasi, dan menyebarkan berita bohong dengan tujuan terjadinya tindak diskriminasi, kekerasan, penghilangan nyawa dan konflik sosial [13].

Menurut Komnas HAM dalam Buku Saku Penanganan Ujaran Kebencian (*Hate Speech*), ujaran kebencian sangat berbahaya karena beberapa alasan berikut [14].

1. Merendahkan sesama umat manusia
Karena manusia adalah ciptaan Tuhan, maka tidak ada yang berhak merendahnya atau siapapun yang juga merupakan produk Tuhan.
2. Menimbulkan kerugian materil dan/atau immaterial
Korban dari ujaran kebencian dapat mengalami kerugian materil maupun kerugian immaterial akibat dampak dari ujaran kebencian yang ditujukan padanya.
3. Dapat membuat konflik
Konflik dapat terjadi melalui hasutan untuk membuat marah individu atau kelompok. Konflik ini dapat dimulai antara individu dan meningkat menjadi perselisihan antarkelompok.
4. Dapat berdampak pada pemusnahan kelompok
Stigmatisasi, pelabelan, pengucilan, diskriminasi, dan kekerasan dapat dihasilkan dari dorongan kebencian ini. Yang paling keji, itu dapat menginspirasi permusuhan luas yang mengakibatkan pembersihan etnis, pembakaran wilayah, atau bahkan genosida terhadap kelompok-kelompok yang menjadi sasaran ujaran kebencian.

Pada tahun 2018, Pemilihan Kepala Daerah (Pilkada) dilaksanakan serentak di 17 provinsi untuk pemilihan Gubernur dan Wakil Gubernur, 115 kabupaten untuk pemilihan Bupati dan Wakil Bupati, serta 39 kota untuk pemilihan Walikota dan Wakil Walikota [15]. Disusul kemudian pada tahun 2019 diselenggarakan pemilihan Presiden dan Wakil Presiden. Oleh karena itu, tahun 2018 dan 2019 sering disebut sebagai tahun politik. Penyebaran ujaran kebencian pun tak dapat dihindarkan, dimana sasaran dari ujaran kebencian adalah kedua pasangan Calon Presiden dan Wakil Presiden yang bertujuan untuk menjatuhkan nama baik [16].

Ujaran kebencian pada saat itu berpusat pada suku, agama, ras, dan hubungan antargolongan. Contoh ujaran kebencian dapat dilihat pada Tabel 2.1 berikut ini.

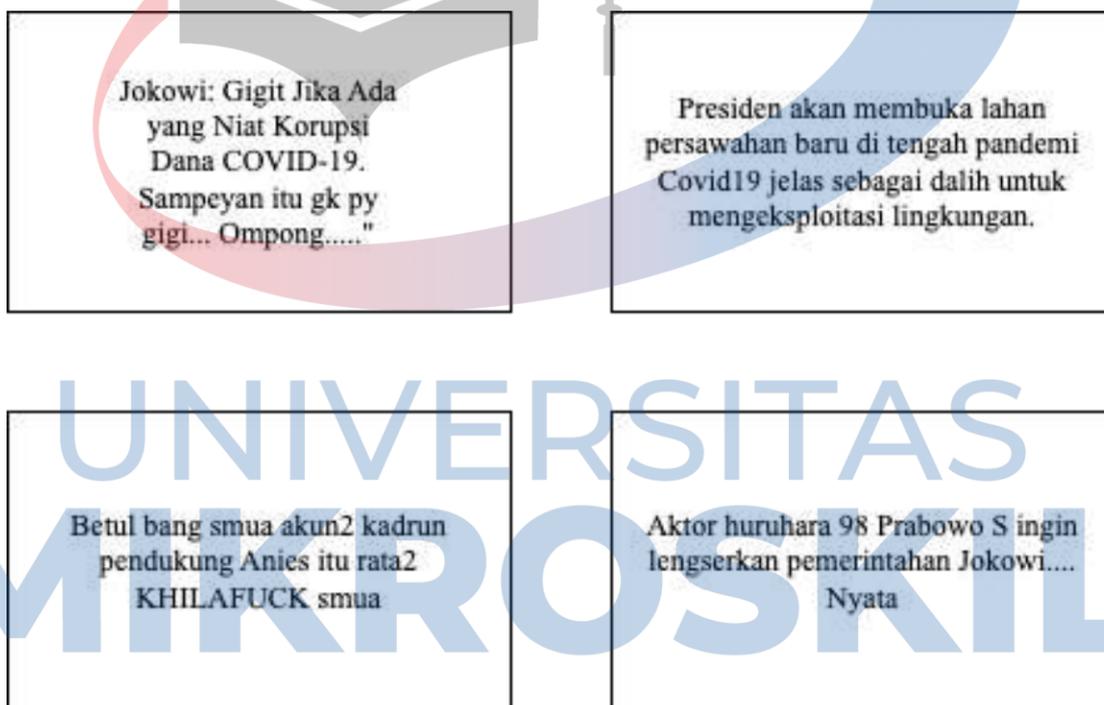
Tabel 2.1 Contoh Ujaran Kebencian pada Domain Politik [17]

Ujaran	Keterangan
kaum cebong kafir lihat dongok dungu haha	Ujaran yang menyinggung individu atau kelompok tertentu.
Aktor huruhara 98 Prabowo S ingin lengserkan pemerintahan Jokowi.... Nyata	Ujaran yang mengandung kebencian yang ditujukan terhadap seseorang atau individu.
partai komunis indonesia muncul jelang pilih kepala daerah pilih presiden goreng gosong	Ujaran yang mengandung kebencian yang ditujukan terhadap kelompok tertentu (kelompok agama, ras, politik, dll).
Betul bang smua akun2 kadrung pendukung Anies itu rata2 KHILAFUCK smua	Ujaran yang mengandung kebencian yang ditujukan terhadap agama, kelompok agama, atau kepercayaan tertentu.
Kata Si Homo ini Pendukung Jokowi kini hanya tersisa Buzzer saja?	Ujaran yang mengandung kebencian berdasarkan gender. Biasanya berisi bahasa yang merendahkan jenis kelamin atau orientasi seksual tertentu.
ARTINYA APA ? Pribumi sadari & lihat siapa mereka ? KECUALI KALIAN sama-sama CINA KOMUNIS / MISIONARIS abaikan tweet ini usir cina Indonesia	Ujaran yang mengandung kebencian yang ditujukan kepada individu atau kelompok dengan cara yang menghasut atau provokatif.

2.1.2 Ujaran Kebencian di *Twitter*

Dengan maraknya penyebaran internet dan pertumbuhan jaringan sosial *online*, masalah ujaran kebencian menjadi lebih serius, karena interaksi antara orang-orang menjadi tidak langsung, dan ucapan orang-orang cenderung lebih agresif ketika mereka merasa lebih aman secara fisik. Banyak kelompok pembenci melihatnya sebagai sarana komunikasi perekrutan yang belum pernah terjadi sebelumnya. Salah satu jaringan sosial *online* yang populer saat ini adalah *Twitter*, media sosial dan layanan web mikro blogging, yang menjadi favorit bagi masyarakat untuk berinteraksi antar sesama.

Orang-orang biasanya menggunakan *Twitter* untuk mendapatkan pembaruan dari koneksi pribadi dan profesional mereka dan untuk melihat berita dunia terbaru. Dengan postingan tanpa batas, orang dapat memposting ulang postingan apa pun di *Twitter*. Akibatnya, lebih mudah untuk mengirim komentar negatif dan ujaran kebencian di *Twitter*. Berikut pada Gambar 2.1 merupakan contoh ujaran kebencian di media sosial *Twitter*.



Gambar 2.1 Contoh Ujaran Kebencian di *Twitter* [17,18]

Hal ini juga terjadi pada domain politik, di mana tim yang bertugas memastikan tokoh yang mereka usung agar terpilih, akan menghalalkan segala cara untuk mengkampanyekan calonnya, termasuk menyerang lawan politik dengan ujaran-ujaran kebencian. Praktik seperti ini tentu mencoreng demokrasi, karena menyerang dengan menggunakan ujaran kebencian mengandung unsur jahat, melanggar norma, dan

memberikan pendidikan politik yang buruk bagi masyarakat serta dapat merugikan pihak yang dijatuhkan [19].

2.1.3 Pre-processing

Informasi yang dikumpulkan di *Twitter* tidak terstruktur yang mencakup *noise*, nilai nol, dll. Data tersebut harus disempurnakan terlebih dahulu sehingga dapat diproses dan digunakan untuk klasifikasi. Untuk klasifikasi, pengelompokan, dan aplikasi deteksi anomali untuk menggunakan teks pendek secara efisien, pra-pemrosesan sangat penting. Pra-pemrosesan memiliki dampak yang signifikan terhadap kinerja sistem secara keseluruhan.

Mempersiapkan *tweet* untuk tugas-tugas seperti pengenalan peristiwa, deteksi informasi penipuan, dan analisis sentimen adalah bagian dari proses prapemrosesan. Orang sering mengikuti seperangkat norma linguistik tidak tertulis mereka sendiri di media sosial. Oleh karena itu, setiap pengguna *Twitter* memiliki gaya penulisannya masing-masing, lengkap dengan akronim, tanda baca yang khas, dan kata-kata yang salah eja. Dalam *tweet*, *emoji* dan *emoticon* digunakan untuk mengekspresikan kompleksitas, emosi, dan ide. Selain itu, hampir semua *tweet* yang diposting pengguna berisi URL yang mengarah ke informasi tambahan, sebutan pengguna (@username), dan simbol hashtag (#), yang menghubungkan *tweet* ke topik tertentu dan juga dapat menyampaikan suasana hati [20].

Tahap pra-pemrosesan data seperti yang dapat dilihat pada Gambar 2.2 terdiri dari 5 langkah, yaitu sebagai berikut [21].

1. Data Cleaning

Pembersihan data merupakan proses menambahkan data yang hilang dan mengoreksi, memperbaiki, atau menghapus data yang keliru atau tidak relevan menurut gugusan data.

2. Case Folding

Dalam menuliskan *tweet*, terkadang orang menggunakan huruf kapital yang tidak sesuai kaidah Bahasa Indonesia. Jika sebuah kata ditulis dengan huruf kapital yang tidak beraturan, kata tersebut akan menghasilkan vektor yang berbeda ketika divektorkan. Sehingga proses *case folding*, yaitu mengubah semua karakter menjadi huruf kecil, diperlukan dalam hal ini.

3. Tokenizing

Dalam bahasa Indonesia, ada banyak ekspresi dengan dua kata atau lebih yang semuanya mengacu pada item yang sama, seperti “buah tangan”, “adu domba”, atau

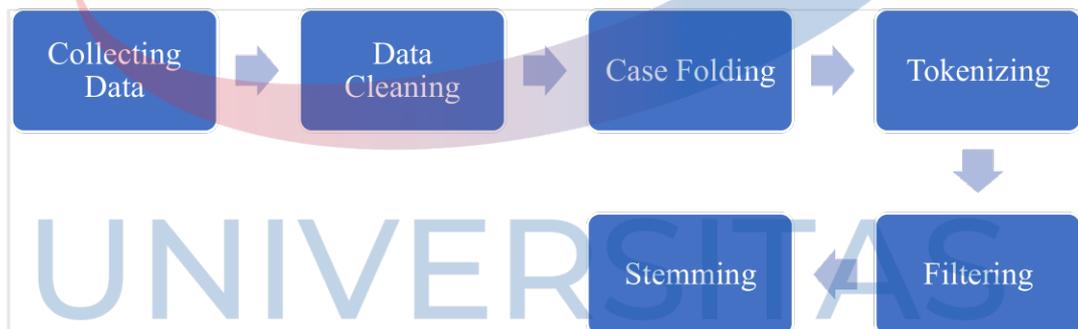
“gelap mata”. Ungkapan "buah tangan" membutuhkan model untuk dapat membedakan antara kata "buah" dan "tangan". *Tokenizing* adalah proses memisahkan kata dengan menggunakan spasi sebagai pemisahannya. Dengan menggunakan *tokenizing* dapat membedakan mana antara pemisah kata atau bukan.

4. Filtering

Setelah dilakukan *tokenizing*, langkah berikutnya adalah *filtering* yang digunakan untuk mendapatkan kata kunci dari hasil token. Kata-kata umum yang sering muncul dan tidak memiliki arti disebut *stopwords*. Misalnya, penggunaan konjungsi seperti dan, yang, serta, setelah, dan lain-lain. Menghapus *stopword* ini dapat mengurangi ukuran indeks dan waktu pemrosesan. Selain itu, juga dapat mengurangi tingkat *noise*.

5. Stemming

Ini adalah metode untuk menentukan akar kata. Tanpa perlu kehilangan makna, akar istilah atau kata dasar dapat digunakan untuk menyaring hasil indeks. Selanjutnya juga untuk mengelompokkan kata-kata lain yang memiliki kata dasar yang sama dan makna yang sama tetapi memiliki bentuk yang berbeda karena ditempatkan dengan imbuhan yang berbeda.



Gambar 2.2 Pra-pemrosesan Data [21]

Hasil dari langkah pra-pemrosesan ditunjukkan pada Tabel 2.2 di bawah ini.

Tabel 2.2 Pra-pemrosesan Data Tweet

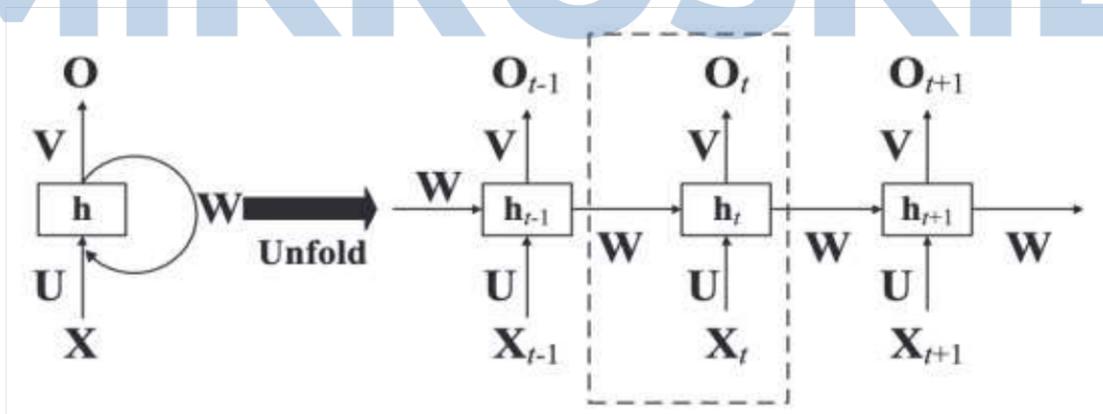
Pra-pemrosesan	Hasil Tweet
Tweet Asli	ARTINYA APA ? Pribumi sadari & lihat siapa mereka ? KECUALI KALIAN sama-sama CINA KOMUNIS / MISIONARIS abaikan tweet ini usir cina Indonesia
Data Cleaning	ARTINYA APA Pribumi sadari dan lihat siapa mereka KECUALI KALIAN sama-sama CINA KOMUNIS atau MISIONARIS abaikan tweet ini usir cina Indonesia

Tabel 2.2 Pra-pemrosesan Data Tweet (Sambungan)

Pra-pemrosesan	Hasil Tweet
Case Folding	artinya apa pribumi sadari dan lihat siapa mereka kecuali kalian sama-sama cina komunis atau misionaris abaikan tweet ini usir cina indonesia
Tokenizing	['artinya', 'apa', 'pribumi', 'sadari', 'dan', 'lihat', 'siapa', 'mereka', 'kecuali', 'kalian', 'sama-sama', 'cina', 'komunis', 'atau', 'misionaris', 'abaikan', 'tweet', 'ini', 'usir', 'cina', 'Indonesia']
Filtering	['artinya', 'pribumi', 'sadari', 'lihat', 'siapa', 'mereka', 'kecuali', 'kalian', 'sama-sama', 'cina', 'komunis', 'misionaris', 'abaikan', 'tweet', 'ini', 'usir', 'cina', 'Indonesia']
Stemming	arti pribumi sadar lihat siapa mereka kecuali kalian sama-sama cina komunis misionaris abai tweet ini usir cina indonesia

2.1.4 Recurrent Neural Network

Recurrent Neural Networks (RNN) adalah salah satu bentuk arsitektur *Artificial Neural Networks* (ANN) yang dirancang khusus untuk memproses data yang bersambung/berurutan (*sequential data*). *Recurrent Neural Networks* (RNN) memiliki kemampuan untuk menyimpan memori sehingga arsitektur ini dapat mengenali pola data dan menggunakannya untuk membuat prediksi yang akurat [22]. Maka dari itu, *Recurrent Neural Networks* (RNN) sangat tepat untuk digunakan terhadap pengenalan suara, klasifikasi, prediksi, ataupun pemrosesan bahasa alami. Gambar 2.3 menunjukkan struktur tradisional dari arsitektur *Recurrent Neural Networks* (RNN).



Gambar 2.3 Struktur tradisional *Recurrent Neural Networks* (RNN) [23]

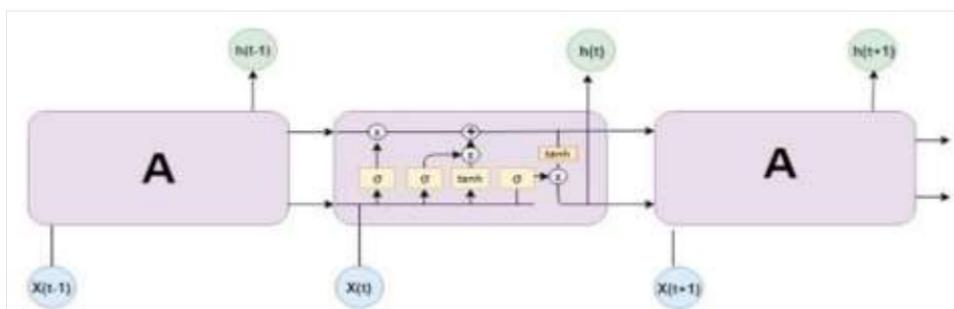
X_t adalah unit input dari langkah t , yang mewakili vektor kata dari kata ke- t dalam teks, h_t adalah keadaan tersembunyi dari langkah t , O_t mewakili output dari step t , output dari langkah ini adalah pengklasifikasi softmax, output dipilih sesuai dengan kebutuhan model. U , V , dan W adalah parameter bobot jaringan yang semuanya perlu dipelajari dalam model. Seperti terlihat pada Gambar 2.3, kotak garis putus-putus merupakan perhitungan satuan ke- t , dapat dilihat pada persamaan (1) sebagai berikut:

$$\begin{cases} h_t = f(W h_{t-1} + U x_t + b_h) \\ o_t = \text{softmax}(V h_t + b_o) \end{cases} \dots\dots\dots(1)$$

Di mana istilah bias diwakili oleh variabel b_h dan b_o . Kata masukan saat ini dan keadaan tersembunyi dari langkah sebelumnya menentukan setiap keadaan tersembunyi dari arsitektur *Recurrent Neural Networks* (RNN), seperti yang ditunjukkan oleh Persamaan. 1. Suatu tugas tidak dapat dikeluarkan jika setiap langkah sintetik tidak memerlukan penambahan pengklasifikasi. Kelemahan arsitektur tradisional *Recurrent Neural Networks* (RNN) adalah bahwa ketika panjang teks meningkat, jumlah dan kedalaman lapisan jaringan terus meningkat. Jaringan mengalami cukup banyak kehilangan selama proses sintesis informasi, yang menyebabkan pembelajaran berkonsentrasi pada langkah terakhir memori. Oleh karena itu, efisiensi pembelajaran teks panjang kurang baik.

2.1.5 Long Short-Term Memory

Recurrent Neural Networks (RNN) diketahui memiliki masalah ketergantungan jangka panjang, dengan kata lain, semakin panjang urutannya, semakin sulit untuk diingat. Untuk mengatasi masalah ini, maka dikembangkan model *Long Short-Term Memory* [24]. *Long Short-Term Memory* (LSTM) adalah jenis spesifik arsitektur *Recurrent Neural Networks* (RNN). *Long Short-Term Memory* (LSTM) dirancang untuk memodelkan urutan temporal dan ketergantungan jarak jauh. Model ini juga mampu menangkap data dari masa lalu dan menggunakannya untuk prediksi masa depan. Arsitektur *Long Short-Term Memory* (LSTM) dapat dilihat pada Gambar 2.4 di bawah ini.



Gambar 2.4. Arsitektur *Long Short-Term Memory* [25]

Long Short-Term Memory (LSTM) menggunakan kombinasi khusus unit tersembunyi, produk elemen, dan kuantitas antar unit untuk menerapkan gerbang kontrol "sel memori". Sel-sel ini dirancang untuk menyimpan informasi tidak berubah untuk waktu yang lama. Fitur utama *Long Short-Term Memory* (LSTM) adalah kemampuannya untuk mempelajari ketergantungan jangka panjang, yang tidak mungkin dilakukan dengan *Recurrent Neural Networks* (RNN) sederhana. Untuk memprediksi langkah berikutnya, bobot pada jaringan harus diperbarui, yang mengharuskan menjaga informasi langkah asli. Jaringan *Long Short-Term Memory* (LSTM) terdiri dari lapisan input, lapisan output, dan sejumlah lapisan tersembunyi rekursif di antara mereka. Kelas tersembunyi rekursif terdiri dari beberapa modul memori. Setiap modul berisi satu atau lebih sel memori yang menghubungkan sendiri dan tiga gerbang yaitu sebagai berikut [26].

1. *Forget Memory*, menampilkan angka antara 0 dan 1, di mana nilai 1 mengindikasikan "sepenuhnya simpan ini", sedangkan, 0 menunjukkan "benar-benar mengabaikan."
2. *Memory Gate*, memilih data baru mana yang akan disimpan di dalam sel. Pertama, lapisan sigmoid "lapisan pintu input" memilih nilai mana yang akan diubah. Selanjutnya, lapisan tanh membuat vektor nilai kandidat baru yang dapat ditambahkan ke keadaan.
3. *Output Gate*, memutuskan apa yang akan menjadi keluaran setiap sel. Nilai output akan didasarkan pada status sel bersama dengan data tambahan yang difilter dan paling baru.

Model *Long Short-Term Memory* (LSTM) memiliki alur kerja sebagai berikut [27].

1. Tentukan informasi apa yang harus dihapus atau dilupakan dari kotak status. Identifikasi ini berasal dari lapisan sigmoid yang dikenal sebagai lapisan *gate forget*. Hal ini dilambangkan dengan h_{t-1} dan x_t . Outputnya adalah nilai 1 atau 0 di setiap status sel.

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f) \dots\dots\dots(2)$$

2. Tentukan informasi baru untuk disimpan dalam status sel. Langkah ini memiliki dua bagian:

- a. Kelas sigmoid disebut kelas *input gate* yang mendefinisikan nilai yang diperbarui
- b. Kelas tanh menghasilkan vektor nilai kandidat baru (\hat{C}_t) yang dapat ditambahkan ke *state*. Kemudian menggabungkan dua langkah untuk membuat *state* baru.

$$\begin{aligned} i_t &= \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \\ \hat{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \end{aligned} \dots\dots\dots(3)$$

3. Perbarui sel *state* lama (C_{t-1}) ke sel *state* baru (C_t) dengan melakukan hal berikut:
 - a. Kalikan *state* lama dengan f_t , hapus item yang ingin Anda hapus sebelumnya.

- b. Kemudian tambahkan operasi $i_t * C_t$. Ini adalah nilai kandidat baru yang diskalakan berdasarkan seberapa banyak ingin memperbarui setiap nilai *state*.

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \dots\dots\dots(4)$$

4. Outputnya didasarkan pada *state* sel, tetapi akan menjadi versi yang difilter. Untuk menentukan output, jalankan kelas sigmoid yang menentukan bagian sel mana yang harus menjadi output. Kemudian mengatur sel *state* melalui tanh (untuk mendorong nilai dari -1 ke 1) dan mengalikannya dengan output dari gerbang sigmoid untuk hanya menampilkan output yang diinginkan.

$$\begin{aligned} o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \dots\dots\dots(5)$$

2.1.6 TF-IDF

Metode penilaian kata yang digunakan dalam penelitian ini adalah *Term Frequency - Inverse Document Frequency* (TF-IDF). TF-IDF adalah ukuran statistik yang mencerminkan pentingnya sebuah kata dalam dokumen dalam suatu kumpulan atau korpus. Metrik ini mempertimbangkan frekuensi kata dalam dokumen target, serta frekuensi dalam dokumen lain dalam korpus. Semakin tinggi frekuensi kata dalam dokumen target dan semakin rendah frekuensi dalam dokumen lain, semakin besar pentingnya kata itu [28].

Cara kerja penilaian kata pada metode TF-IDF adalah dengan menggunakan 2 parameter *scoring*, yaitu skoring lokal $tf_{i,j}$ yang merupakan skor yang diperoleh dari frekuensi kemunculan kata *i* pada dokumen *j* dan skoring global menggunakan idf_i yang merupakan skor yang diperoleh dengan mempertimbangkan jumlah kemunculan kata *i* (DF_i) untuk semua *N* dokumen. Kemudian, skor skoring lokal dikalikan dengan skor skoring global [2]. Cara menghitung skor menggunakan persamaan sebagai berikut [23].

$$f(t_i) = \frac{tf_{i,j} * \log \frac{N}{n+1}}{\sqrt{\sum t_i \in d_j [tf_{i,j} * \log \frac{N}{n+1}]^2}} \dots\dots\dots(6)$$

2.1.7 Confusion Matrix

Confusion Matrix adalah metode untuk menguji atau mengevaluasi suatu metode untuk mengklasifikasikan hasil yang diberikan oleh sistem dengan hasil yang seharusnya. Dalam metode ini terdapat 4 (empat) istilah yang digunakan, yaitu [29].

1. TP adalah *True Positive*, merupakan jumlah data positif yang diprediksi dengan benar oleh sistem.

2. TN adalah *True Negative*, merupakan jumlah data negatif yang diprediksi dengan benar oleh sistem.
3. FN adalah *False Negative*, merupakan jumlah data negatif namun diprediksi salah oleh sistem.
4. FP adalah *False Positive*, merupakan jumlah data positif namun diprediksi salah oleh sistem.

Tabel dari *Confusion Matrix* dapat dilihat pada table 2.3 di bawah ini.

Tabel 2.3 *Confusion Matrix* [29]

		<i>Observed</i>	
		<i>True</i>	<i>False</i>
<i>Predicted Class</i>	<i>True</i>	<i>True Positive</i> (TP)	<i>False Positive</i> (FP)
	<i>False</i>	<i>False Negative</i> (FN)	<i>True Negative</i> (TN)

Metode ini menggunakan beberapa rumus untuk mengukur daya banding hasil klasifikasi, antara lain [29].

1. Akurasi, yaitu perbandingan hasil prediksi benar dari keseluruhan data yang dihitung dengan persamaan:

$$\frac{TP+TN}{TP+TN+FP+FN} \times 100\% \dots\dots\dots(7)$$

2. Presisi, yaitu menggambarkan jumlah data positif yang diprediksi secara benar dan dibagi dengan total data yang positif, dapat dihitung dengan persamaan:

$$\frac{TP}{TP+FP} \times 100\% \dots\dots\dots(8)$$

3. Recall, yaitu menunjukkan beberapa persen kategori positif yang diprediksi dengan benar oleh sistem, dapat dihitung dengan persamaan:

$$\frac{TP}{TP+FN} \times 100\% \dots\dots\dots(9)$$

4. Error, yaitu menunjukkan tingkat kesalahan prediksi yang dihasilkan oleh sistem, dapat dihitung dengan persamaan:

$$\frac{FP}{TP} \times 100\% \dots\dots\dots(10)$$

2.2 Penelitian Sebelumnya

Penelitian ini mengacu pada beberapa penelitian sebelumnya yang membahas deteksi ujaran kebencian dengan berbagai macam metode sebagai sumber referensi. Penelitian sebelumnya dapat dilihat pada Tabel 2.3 di bawah ini.

Tabel 2.4 Penelitian Sebelumnya Terkait Pendeteksian Ujaran Kebencian

No	Nama Peneliti, Judul dan tahun	Classifiers / Teknik Deteksi	Dataset yang digunakan	Hasil / Fitur Utama
1.	Buntoro, G. A., dkk., 2020, The Implementation Of The Machine Learning Algorithm For The Sentiment Analysis Of Indonesia's 2019 Presidential Election.	<i>Naïve Bayes Classifier</i> (NBC) dan <i>Support Vector Machine</i> (SVM)	Melakukan <i>crawling</i> pada Twitter. Data yang diambil hanya berupa tweet dalam bahasa Indonesia yang terdiri dari 5.000 tweet yang mengandung kata kunci Jokowi dan 5.000 tweet yang mengandung kata kunci Prabowo	<ol style="list-style-type: none"> 1. Nilai akurasi tertinggi 79.02% untuk algoritma <i>Support Vector Machine</i> (SVM) dengan tokenisasi abjad. 2. Model yang dibangun masih membuat sejumlah kesalahan ketika proses klasifikasi dataset dengan distribusi sentimen tidak seimbang.
2.	Talita, A. S. dan Wiguna, A., 2019, Implementasi Algoritma Long Short-Term Memory (LSTM) Untuk Mendeteksi Ujaran Kebencian (Hate Speech) Pada Kasus Pilpres 2019.	<i>Long Short-Term Memory</i> (LSTM)	Data <i>corpus</i> Universitas Leipzig dan data kalimat ujaran kebencian (Hate Speech/HS) dan non ujaran kebencian (Non Hate Speech/NHS) yang berasal dari Facebook	<ol style="list-style-type: none"> 1. <i>Long Short-Term Memory</i> sudah cukup baik dalam mendeteksi kalimat ujaran kebencian dengan nilai parameter <i>recall</i> mencapai 0.7021. 2. Nilai parameter akurasi dan <i>precision</i> cenderung rendah.

Tabel 2.4 Penelitian Sebelumnya Terkait Pendeteksian Ujaran Kebencian (Sambungan)

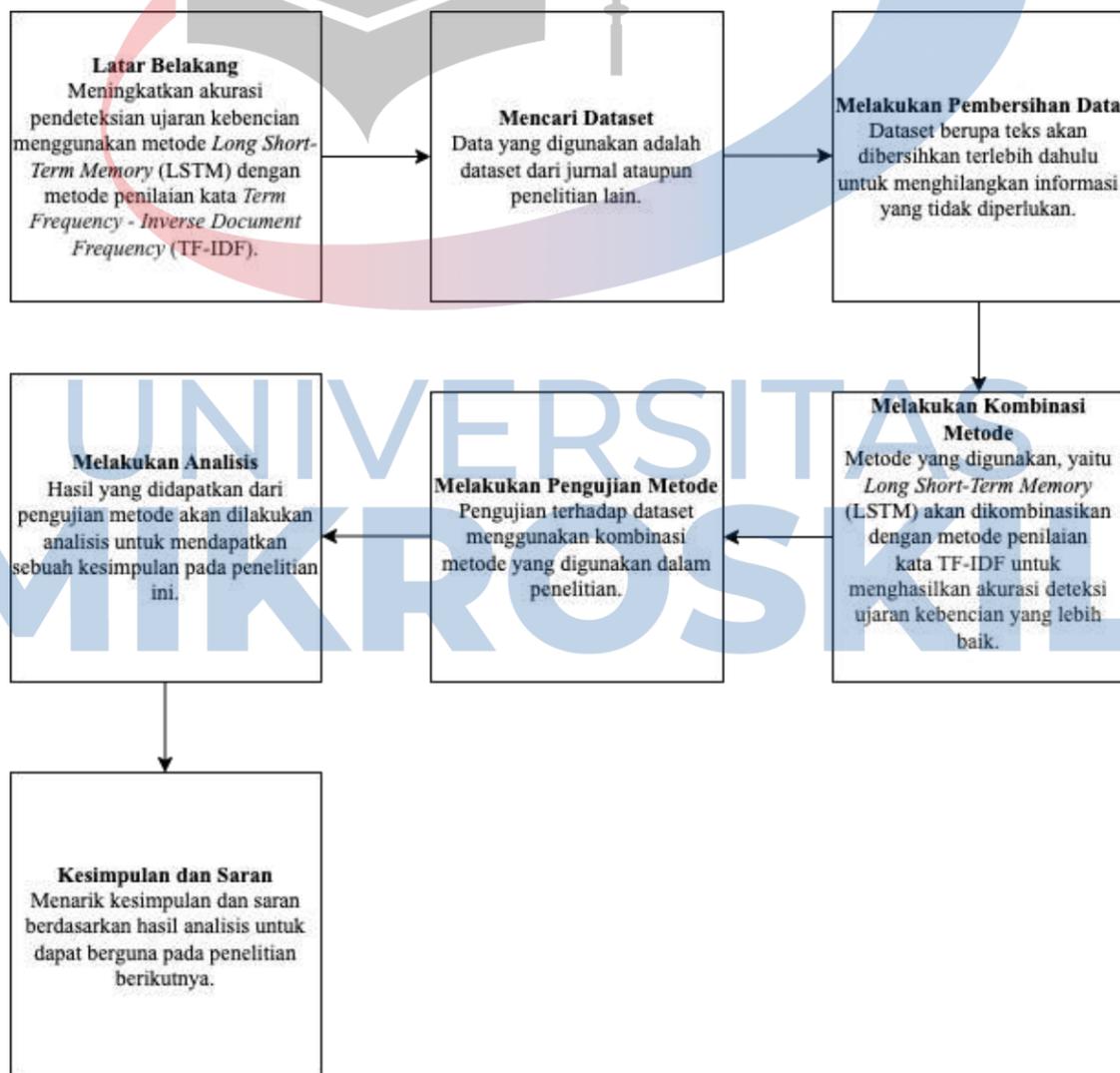
No	Nama Peneliti, Judul dan tahun	Classifiers / Teknik Deteksi	Dataset yang digunakan	Hasil / Fitur Utama
3.	Alfina, I., Fanany, M. I., dan Mulia, R., 2017, Hate Speech Detection in the Indonesian Language: A Dataset and Preliminary Study	NB, SVM, BLR, dan RFDT	1.100 tweets yang diperoleh menggunakan <i>Twitter Streaming API</i> .	<ol style="list-style-type: none"> 1. <i>Random Forest Decision Tree (RFDT)</i> menghasilkan akurasi 93.5%. 2. Model <i>BOW (Bag of Words)</i> yang dipakai tidak memadai untuk mendeteksi ujaran kebencian
4.	Tabassum, A. dan Patil, R. R., 2020, A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing	Natural Language Processing	Berasal dari peneliti lain.	Teknik Ekstraksi Fitur yang paling banyak digunakan adalah model <i>Bag-of-words</i> dan <i>TF-IDF</i> . <i>TF-IDF</i> sejauh ini merupakan pilihan terbaik untuk menyortir fitur yang menonjol dan mengurangi fitur yang tidak relevan.
5.	Wang, J. H., dkk, 2018, An LSTM Approach to Short Text Sentiment Classification with Word Embeddings	Naïve Bayes (NB), Extreme Learning Machine (ELM), and Long Short-Term Memory (LSTM)	<p>Terdapat 3 set data:</p> <ol style="list-style-type: none"> 1. 50.000 ulasan di dataset Ulasan Film IMDB 2. Dari 200 film teratas, diambil 12.000 komentar 3. Dari media sosial Taiwan, mengambil 3.500 posting dan 34.488 komentar. 	Metode pembelajaran mendalam seperti LSTM menunjukkan kinerja klasifikasi sentimen yang lebih baik ketika ada lebih banyak jumlah pelatihan data.

2.3 Kerangka Konsep

Penelitian ini dimulai dari ide untuk meningkatkan akurasi deteksi ujaran kebencian dengan melakukan analisis ujaran kebencian berdasarkan dataset ujaran kebencian dan ujaran non-kebencian menggunakan metode *Long Short-Term Memory* (LSTM) dengan menggunakan metode penilaian kata adalah *Term Frequency - Inverse Document Frequency* (TF-IDF).

Dataset tersebut nantinya akan dilakukan pengujian terhadap akurasi deteksi berdasarkan kombinasi metode pada penelitian ini. Pengujian dilakukan dalam beberapa skenario, yaitu pengujian terhadap deteksi ujaran kebencian tanpa metode penilaian kata dan kombinasi metode penilaian kata dalam penelitian ini, pengujian terhadap deteksi ujaran kebencian dengan beragam jenis ujaran yang tersedia dalam dataset.

Untuk evaluasi terhadap model pada penelitian ini dilakukan menggunakan *Confusion Matrix*. Kerangka Konsep Penelitian dapat dilihat pada Gambar 2.5 di bawah ini.



Gambar 2.5. Kerangka Konsep Penelitian