

**BAB II**  
**TINJAUAN PUSTAKA DAN DASAR TEORI**

**2.1 Tinjauan Pustaka**

Tinjauan pustaka yang diacu pada penelitian ini adalah sebagai berikut

Tabel 2. 1 Tinjauan Pustaka

No.	Penulis	Objek	Metode	Hasil
1.	Oni Harnantyo (2019)	<i>Tweet</i> berbahasa Indonesia yang membicarakan tentang beberapa tempat wisata di Yogyakarta	<i>Recurrent Neural Network</i> Dengan <i>Long Short Term Memory</i>	Tingkat akurasi yang didapat 99,31%
2.	Araque (2017)	<i>Tweet</i> berbahasa Spanyol	<i>RNN</i> dan <i>LSTM</i>	Peningkatan performa analisis sentimen menggunakan kombinasi dua tipe fitur <i>word embedding</i> dan <i>lexicon values</i>

3.	Ruales (2014)	Klasifikasi sentimen pada ulasan film	<i>RNN dan LSTM</i>	RNN dengan LSTM mempunyai hasil tingkat error yang lebih kecil sebesar 0.134 dibanding RNN tanpa LSTM
4.	Hari Purnomo (2019)	<i>Tweet</i> berbahasa Indonesia yang berhubungan dengan Pemerintah Provinsi Jawa Tengah	<i>Naïve Bayes Classifier</i>	Tingkat akurasi yang didapatkan sebesar 95%
5.	Dede Aryadani (2020)	Respon atau tanggapan masyarakat terhadap konten sosial media Instagram STMIK Akakom Yogyakarta	<i>Recurrent Neural Network</i> dengan <i>Long Short Term Memory</i>	Tingkat pengujian sebesar 65% dan tingkat akurasi penerapan sebesar 79,46%
6.	Rizky Maulana (2016)	<i>Tweet</i> pengguna <i>twitter</i>	<i>Support Vector Machine</i>	Dengan 1400 <i>tweet</i> pada dataset dan 200 data uji didapatkan akurasi

				sebesar 79%
7.	Rosit Sanusi (Usulan)	<i>Tweet</i> yang berhubungan dengan program kartu prakerja	<i>Recurrent Neural Network dengan Long Short Term Memory</i>	Klasifikasi sentimen positif, netral, negatif pada tweet berbahasa Indonesia tentang kartu prakerja.

Oni Harnantyo (2020) melakukan penelitian terhadap sentimen tweet berbahasa Indonesia yang membicarakan tentang beberapa tempat wisata di Yogyakarta. Analisis sentimen yang dilakukan menggunakan metode *Recurrent Neural Network (RNN)* dengan *Long Short Term Memory (LSTM)*. Hasil dari penelitian ini adalah sistem klasifikasi dengan tingkat akurasi pelatihan sebesar 99,31 %.

Araque (2017) juga melakukan penelitian dengan melakukan analisis sentimen terhadap *tweet* berbahasa Spanyol menggunakan metode *Long Short Term Memory (LSTM)*. Dengan mengkombinasikan 2 tipe fitur berbeda, *word embedding* dan *lexicon values* hasil yang didapat peneliti menunjukkan bahwa kombinasi 2 fitur tadi akan menambah performa analisis sentimen.

Ruales (2014) juga melakukan penelitian untuk membandingkan metode *Long Short Term Memory (LSTM)* dengan beberapa metode untuk mengklasifikasikan sentimen terhadap ulasan film. Dari penelitian tersebut diketahui bahwa tingkat

error *Reccurent Neural Network* (RNN) dengan *Long Short Term Memory* (LSTM) lebih kecil dibanding *Reccurent Neural Network* (RNN) tanpa *Long Short Term Memory* (LSTM) yakni sebesar 0.134.

Hari Purnomo (2019) melakukan analisa terhadap tweet berbahasa Indonesia yang berhubungan dengan Pemerintah Jawa Tengah. Dari hasil penelitian klasifikasi topik dan sentimen didapat tingkat akurasi sebesar 95 % dengan menggunakan metode *Naïve Bayes Classifier*.

Dede Aryadani (2020) melakukan analisis terhadap respon atau tanggapan masyarakat terhadap konten sosial media Instagram STMIK Akakom Yogyakarta. Dengan menggunakan metode *Recurrent Neural Network* dengan *Long Short Term Memory* penelitian ini menghasilkan tingkat akurasi pengujian sebesar 65% dan tingkat akuasi penerapan sebesar 79,46%.

Rizky Maulana (2016) dalam sebuah penelitian mengenai analisis sentimen terhadap pengguna *twitter* dengan menggunakan metode *Support Vector Machine* berbasis *cloud computing*. Di hasilkan dari 1400 *tweet* dataset dan 200 data uji didapat tingkat akurasi sebesar 79,5%.

## **2.2 Dasar Teori**

### **2.2.1 Analisis Sentimen**

Analisis sentimen atau biasa disebut *opinion mining*, adalah bidang studi yang menganalisa pendapat manusia, sentimen, evaluasi , penilaian, attitude, dan emosi terhadap entitas tertentu misalnya produk, layanan, organisasi, individu, isu, event, dan topik (Liu, 2012).

Analisis sentimen terdiri dari tiga level analisis yaitu :

#### 1. Level Dokumen

Level dokumen menganalisis satu dokumen penuh dan mengklasifikasi dokumen tersebut ke dalam sentimen positif atau negatif. level analisis ini berasumsi bahwa keseluruhan dokumen hanya berisi opini tentang satu entitas saja. Level analisis ini tidak cocok diterapkan pada dokumen yang membandingkan lebih dari satu entitas (Liu, 2012).

#### 2. Level Kalimat

Level kalimat menganalisis satu kalimat dan menentukan tiap kalimat bernilai positif, netral, atau negatives. Sentimen netral berarti kalimat tersebut bukan opini (Liu, 2012).

#### 3. Level Entitas dan Aspek

Level aspek tidak melakukan analisis pada konstruksi bahasa (dokumen, paragraf, klausa, atau frasa) melainkan langsung pada opini itu sendiri. Hal ini didasari bahwa opini terdiri dari sentimen (positif atau negatif) dan target dari opini tersebut. Tujuan level analisis ini adalah untuk menemukan sentimen entitas pada tiap aspek yang dibahas (Liu, 2012).

### **2.2.2 Twitter**

Twitter merupakan salah satu sosial media dengan jumlah pengguna paling banyak diseluruh dunia (Kemp, 2020). Twitter termasuk kedalam microblogging

yaitu sistem yang mampu untuk mengirim dan menerima post pendek atau biasa disebut tweet yang mempunyai panjang maksimal 140 karakter.

Cara kerja dari twitter sendiri adalah pengguna mengikuti pengguna lain, jika kita mengikuti seseorang maka kita dapat melihat tweet orang tersebut dilaman twitter kita atau *timeline*. Pengguna bisa memilih untuk mengikuti orang atau organisasi dengan berbagai latar belakang. Selain itu pengguna bisa menuliskan tweet kita sendiri atau melakukan posting Kembali *tweet* orang lain (*retweet*).

### **2.2.3 Twitter API**

*API* merupakan cara program komputer "berbicara" satu sama lain agar mereka dapat meminta dan menyajikan informasi. Ini dilakukan dengan mengizinkan aplikasi perangkat lunak memanggil apa yang disebut sebagai *endpoint*: alamat yang terkait dengan informasi jenis tertentu yang kami sediakan (*endpoint* umumnya unik seperti nomor telepon). Twitter mengizinkan akses ke bagian dari layanan kami melalui *API* untuk memungkinkan orang-orang membangun perangkat lunak yang terintegrasi dengan Twitter seperti solusi yang membantu sebuah perusahaan menjawab umpan balik pelanggan di Twitter.

Data Twitter berbeda dari data yang dibagi oleh kebanyakan platform sosial lain karena data tersebut mencerminkan informasi yang dipilih pengguna untuk dibagikan ke publik. Platform API kami menyediakan akses luas ke data Twitter publik yang telah dipilih pengguna untuk dibagikan ke dunia. Kami juga mendukung API yang memungkinkan pengguna mengelola informasi Twitter

mereka yang non-publik (mis. *Direct Message*) dan memberikan informasi ini ke pengembang yang telah diizinkan pengguna untuk melakukannya.

#### **2.2.4 Program Kartu Prakerja**

Menurut Peraturan Presiden (PERPRES) Republik Indonesia Nomor 36 Tahun 2020 Tentang Pengembangan Kompetensi Kerja Melalui Program Kartu Prakerja pasal 1 ayat 1 yaitu “*Program Kartu Prakerja adalah program pengembangan kompetensi kerja yang ditujukan untuk pencari kerja, pekerja/buruh yang terkena pemutusan hubungan kerja, dan/atau pekerja/ buruh yang membutuhkan peningkatan kompetensi*”. PERPRES ini ditetapkan pada tanggal 26 Februari 2020 dan mulai tanggal 28 Februari 2020.

Program ini mempunyai tujuan antara lain mengembangkan kompetensi angkatan kerja dan meningkatkan produktivitas dan daya saing angkatan kerja.

#### **2.2.5 Preprocessing**

Tahap *preprocessing* atau ekstraksi data adalah tahap untuk mengolah data mentah sebelum digunakan diproses produksi. Pada tahap ini biasanya data yang tidak diperlukan akan dihilangkan atau dieleminasi sehingga data akan menjadi bentuk yang lebih mudah dipahami untuk diproses di sistem (Siti, 2016).

Tahap *preprocessing* menurut penilitan yang dilakukan oleh Siti (2016) dibahas sebagian tahapannya antara lain sebagai berikut :

1. *Case Folding* : proses merubah semua huruf yang ada di teks dokumen menjadi huruf kecil

2. *Remove Punctuation* : proses menghapus semua karakter bukan alfabet misalnya simbol, spasi, dan lain – lain.
3. *Remove Username* : bertujuan menghapus nama pengguna, yang biasanya ditandai dengan simbol “@” .
4. *Remove Hashtag* : bertujuan menghapus simbol *hashtag* atau tanda pagar “#”. Simbol ini di twitter merupakan suatu penunjuk sebuah kata yang dibicarakan oleh sesama pengguna twitter. Simbol ini juga digunakan sebagai judul topik pembicaraan dan juga pengelompokan terhadap percakapan yang berhubungan dengan kata yang diberi *hashtag*.
5. *Clean Number* : berfungsi untuk menghapus angka yang selalu ada di depan dan dibelakang kata. Meskipun dalam penulisan komentar selalu menyertakan sebuah angka di setiap awal atau akhir kalimat untuk menunjukkan bahwa kalimat tersebut diulang-ulang maka dalam bahasa Indonesia yang baik itu merupakan hal yang salah. Begitu juga pada sebuah penelitian, apabila menemukan sebuah kata yang menggunakan tambahan angka maka perlu dihapus.
6. *Clean One Character* : Berfungsi menghapus jika terdapat hanya satu huruf saja, karena tidak mengandung arti. Seringnya muncul sebuah huruf pada komentar twitter membuat sebuah hasil data ekstraksi yang banyak dan tidak baik. Satu huruf yang dimaksud adalah sebagai contoh y, g, k dan lain sebagainya. Walaupun maksud dari penulis komentar bahwa y adalah ya, g adalah tidak, k adalah kok. Maka untuk proses ekstraksi data itu merupakan

sebuah kata yang tidak mudah dideklarasikan karena tidak memiliki arti yang jelas.

7. *Removal URL* : Seringnya muncul sebuah url dari data twitter membuat data tidak efektif dan tidak memiliki arti. Untuk itu perlu adanya penghapusan url tersebut. Kemunculan alamat web atau url ini disebabkan karena banyaknya user mempromosikan sebuah produk pada situs mereka supaya user yang lain langsung bisa masuk pada halaman web yang dimaksud.
8. *Remove Retweets* : Pada twitter untuk menunjuk atau mengajak teman berkomunikasi langsung adalah dengan menambahkan simbol “@” sebelum user name yang dituju. Pada suatu penelitian tidak memperhatikan sebuah nama user dan banyaknya user yang komentar.
9. *Convert Number* : Seringnya pemakaian bahasa gaul pada twitter melibatkan angka menjadi variasi dalam menulis seperti “s4y4n9” dan lainnya. Dalam Bahasa Indonesia yang baik kata “s4y4n9” tidak memiliki makna, padahal maksud dari kata tersebut adalah sayang. Untuk itu perlu adanya proses convert number untuk mengkonversi angka menjadi huruf.
10. *Remove Stopword* : *Stopword* diproses pada sebuah kalimat jika mengandung kata-kata yang sering keluar dan di anggap tidak penting seperti waktu, penghubung, dan lain sebagainya (Vijayarani). Untuk itu perlu dilakukan penghapusan. Untuk melakukan proses penghapusan kata ini diperlukan sebuah data atau daftar kata yang diinginkan untuk dihapus.

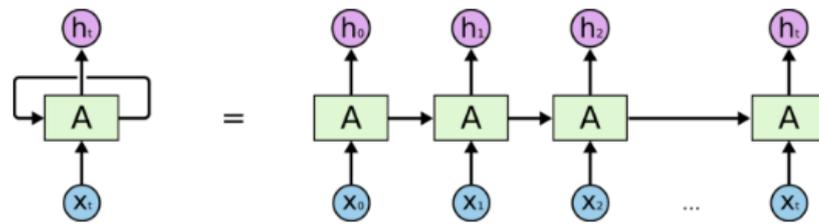
### 2.2.6 Word Embedding

Teks yang akan diproses pada RNN harus dikonversi dalam bentuk embedding. Embedding adalah sebuah vektor atau representasi numerik. Hal ini sangat diperlukan sehingga operasi aritmatika dapat dilakukan menggunakan vektor tersebut (Oni, 2019).

Pada sekitar tahun 2000, mulai dikembangkan teknik word embedding (Bengio et al., 2003; Mikolov et al., 2013; Pennington, Socher and Manning, 2014; Bojanowski et al., 2017). *Word embedding* memetakan setiap kata dalam dokumen ke dalam *dense vector*, di mana sebuah vektor merepresentasikan proyeksi kata di dalam ruang vektor. Posisi kata tersebut dipelajari dari teks atau berdasarkan kata-kata di sekitarnya. *Word embedding* ini dapat menangkap makna semantik dan sintaktik kata. (Nurdin dkk, 2020).

### 2.2.7 RNN

*Recurrent Neural Network* atau *RNN* adalah keluarga dari neural network yang digunakan untuk memproses data sekuensial (Rumelhart et al , 1986a). Berbeda dengan Convolutional Neural Network (CNN) yang digunakan untuk pemrosesan data seperti gambar, RNN sering digunakan untuk memproses nilai yang bersifat sekuensial  $x^1, \dots, \dots, x^n$  (Olah, 2015)



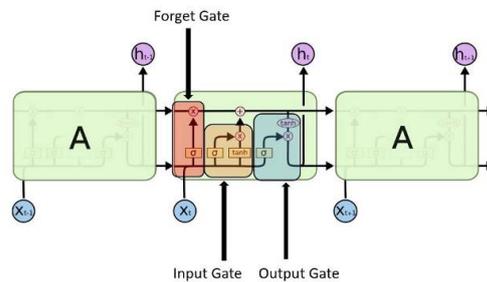
An unrolled recurrent neural network.

Gambar 2. 1 Struktur RNN

Gambar 2.1  $x_t$  merupakan masukan dan  $h_t$  merupakan keluaran, sederhananya bentuk konseptual dari RNN (gambar sebelah kiri) terlihat seperti 1 jaringan dengan pengulangan. Pengulangan ini membuat informasi dapat dilewatkan dari satu langkah pada jaringan ke selanjutnya. Jadi RNN dapat dijabarkan sebagai banyak salinan dari jaringan yang sama (gambar sebelah kanan).

### 2.2.8 Long Short Term Memory

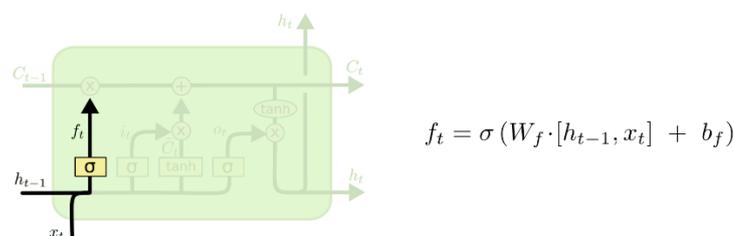
*LSTM* atau *Long Short Term Memory* merupakan bentuk spesial dari RNN yang dapat melakukan pembelajaran pada dependensi jangka panjang (*long-term dependencies*). Model ini diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 (Olah, 2015).



Gambar 2. 2 Struktur LSTM

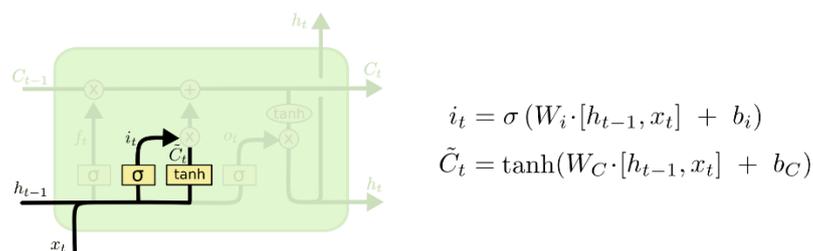
Gambar 2.2 seluruh *recurrent neural network* mempunyai bentuk rangkaian jaringan syaraf yang berulang. Begitu pula dengan LSTM namun disini LSTM mempunyai tambahan fitur pada gerbang sel yaitu *forget gate*, *input gate*, dan *output gate*.

Langkah pertama dalam LSTM adalah menentukan informasi yang akan dibuang atau dilupakan dalam sel, hal ini merupakan kewenangan dai *Forget Gate* yang mempunyai fungsi untuk membuat keputusan informasi apa yang akan dibuang dari sel. Disini layer akan memperhatikan  $h_{t-1}$  dan  $x_t$  yang nantinya akan menghasilkan keluaran 0 atau 1. Jika keluaran 0 menunjukkan bahwa informasi akan dilupakan sedangkan keluaran 1 menunjukkan bahwa informasi tidak akan dilupakan (Gambar 2.3).



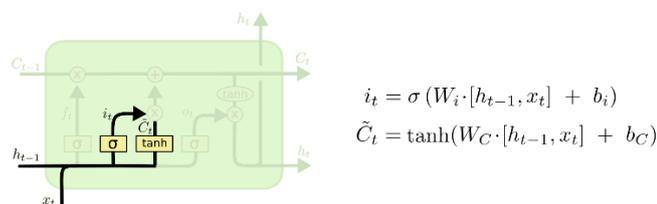
Gambar 2. 3 Struktur Forget Gate

Langkah selanjutnya adalah menentukan informasi baru yang akan disimpan dalam sel. Langkah ini terdiri dari bagian, pertama *layer sigmoid* disebut “*input gate layer*” yang akan menentukan nilai mana yang akan diperbaharui. Selanjutnya sebuah *layer tanh* yang akan membuat vektor sebagai nilai kandidat yang baru ( $C_t$ ), yang dapat ditambahkan kedalam *state*. Berikutnya kedua *layer* ini akan dikombinasikan untuk memperbaharui *state*.



Gambar 2. 2 Struktur Input Gate

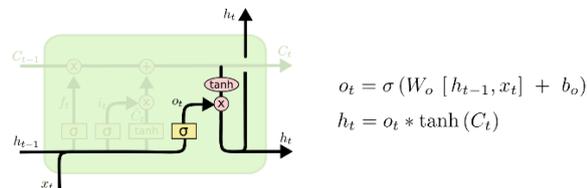
Berikutnya, state lama akan diperbaharui ( $C_{t-1}$ ) menjadi state baru  $C_t$ . Dimana state lama akan dikalikan dengan  $f_t$  dengan mengabaikan informasi yang telah dilupakan sebelumnya. Lalu, keluaran dari  $i_t$  ditambahkan dengan  $\tilde{C}_t$ , sehingga akan memperbaharui state dengan nilai baru ( $C_t$ ) menurut jaringan saraf relevan (gambar 2.3)



Gambar 2. 3 Menghitung State Baru ( $C_t$ )

Langkah terakhir adalah menentukan keluarannya. Pertama *layer sigmoid* akan dijalankan untuk bagian dari sel yang akan dikeluarkan. Kemudian sel

tersebut akan dilewatkan melalui *layer tanh* (untuk mendorong nilai antara -1 dan 1) dan mengalikannya dengan output dari *sigmoid gate* (gambar 2.4).



Gambar 2. 4 Struktur Output Gate

### 2.2.9 Evaluasi Model Klasifikasi

Konsep dasar dalam evaluasi sebuah model klasifikasi adalah tingkat gagasan kesalahan. Jika aplikasi dari model klasifikasi dalam kasus yang dipilih mengarah pada prediksi kelas yang berbeda dari contoh kelas yang sebenarnya maka terjadi kesalahan dalam klasifikasi. Jika ada kesalahan yang sama pentingnya, maka jumlah kesalahan dalam himpunan yang diamati dapat menjadi indikator kerja pengklasifikasi.

Pendekatan ini didasarkan pada akurasi sebagai ukuran untuk mengevaluasi kualitas klasifikasi model. Pengukuran ini dapat didefinisikan sebagai rasio jumlah contoh yang diklasifikasikan dengan benar sesuai dengan jumlah total contoh yang diklasifikasikan. (Jasmina, dkk)

$$Akurasi = \frac{Jumlah\ Prediksi\ Benar}{Jumlah\ Total\ Prediksi}$$

Gambar 2. 5 Rumus Akurasi

Kelemahan utama dari akurasi sebagai ukuran untuk evaluasi adalah sebagai berikut: (1) Mengabaikan perbedaan antara jenis kesalahan dan (2) bergantung pada distribusi kelas dalam dataset.

### **2.2.10 Keras**

Keras adalah sebuah library jaringan syaraf tiruan tingkat tinggi yang ditulis dengan python yang berjalan diatas Tensorflow, CNTK, dan Theano (A. Santoso, G. Ariyanto, 2018).

Keuntungan menggunakan keras antara lain :

1. Menawarkan prototyping pengembangan yang mudah dan cepat.
2. Mempunyai arsitektur yang ringkas dan mudah dibaca.
3. Dapat digunakan untuk model dengan tingkat performa yang rendah.

### **2.2.11 Tensorflow.js**

TensorFlow.js adalah Library untuk melatih dan *deploying* model *machine learning* di *JavaScript*. Model dari tensorflow js dapat berjalan di web browser dan node js (D. Smilkov dkk, 2019). Library ini merupakan bagian dari ekosistem tensorflow dimana disediakan API tingkat tinggi yang compatible dengan python, sehingga memungkinkan untuk menggunakan model antar environment baik itu python ataupun javascript melalui konversi.