BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka dalam penelitian ini merupakan referensi penulisan dalam membangun aplikasi pencarian jasa pemeliharaan bangunan. Referensi penelitian ditunjukkan pada Tabel 2.1.

Tabel 2.1 Tinjauan Pustaka

No.	Peneliti	Tahun	Objek	Metode	Hasil
1.	Araque, Oscar	2017	*Tweet* berbahasa Spanyol	RNN dan LSTM	Dengan mengkombinasi-kan kedua tipe fitur berbeda: *word embedding* dan *sentiment lexicon values* ini, performa analisis sentimenpun meningkat.
2.	Hassan, Abdalraouf	2017	Data IMDB	RNN, LSTM, *Bag of words* dan N-*grams*	Model sederhana RNN-LSTM dengan word2vec menghasilkan tingkat akurasi sebesar 95.1%
3.	Irfangi	2019	*Tweet* mengenai transporta-si *online* di Indonesia	*Naïve Bayes Classifier*	Hasil uji akurasi pengujian 109 data, dihasilkan akurasi sebesar 84%.

No.	Peneliti	Tahun	Objek	Metode	Hasil
4.	Oni Harnantyo	2019	*Tweet* mengenai tempat wisata yang ada di Yogyakarta	RNN dan LSTM	Sistem mampu melakukan klasifikasi dengan tingkat akurasi 95.98% melalui *library* dan 70% melalui *form* klasifikasi. Belum mampu melakukan praproses untuk kata-kata singkatan dan slang.
5.	Septian Narsa Putra	2019	*Tweet* mengenai Divisi Humas Polri	*Naïve Bayes Classifier*	Hasil akurasi pengujian klasifikasi dengan metode *Naïve Bayes Classifier* adalah 86%. *Tweet* terbagi menjadi tiga topik: kegiatan polisi, layanan masyarakat dan komentar masyarakat.
6.	Usulan	2019	Komentar pada konten IG STMIK AKAKOM Yogyakarta	RNN dan LSTM	Sentimen positif, negatif dan netral pada konten Instagram STMIK AKAKOM Yogyakarta

Araque (2017) juga melakukan penelitian dengan mengimplementasikan metode *Long Short Term Memory* (LSTM) untuk melakukan analisis sentimen pada *tweet* berbahasa Spanyol. Peneliti mengimplementasikan dua tipe fitur yang berbeda, *word embedding* dan *sentiment lexicon values*. Hasil yang

didapat mengindikasikan bahwa kombinasi dua fitur ini menambah performa analisis sentimen.

Hassan (2017) melakukan penelitian dengan mengimplementasikan metode *Long Short Term Memory* (LSTM) untuk melakukan analisis sentimen pada situs IMDB. Peneliti menggunakan metode *Long Short Term Memory* (LSTM) berbasis word2vec. Penelitian tersebut menghasilkan hasil yang lebih akurat untuk melakukan analisis sentimen.

Irfangi (2019) melakukan penelitian dengan mengimplementasikan metode *Naïve Bayes Classifier* untuk melakukan analisis sentimen terhadap trasportasi online di Indonesia pada media Twitter. Hasil uji akurasi pengujian 109 data, dihasilkan nilai akurasi sebesar 84%.

Oni (2019) melakukan penelitian dengan mengimplementasikan metode *Reccurent Neural Network* (RNN) dengan *Long Short Term Memory* (LSTM) untuk melakukan analisis sentimen terhadap tempat wisata di Yogyakarta pada media Twitter. Sistem mampu melakukan klasifikasi dengan tingkat akurasi 95.98% melalui *library* dan 70% melalui *form* klasifikasi. Namun, sistem belum mampu melakukan praproses untuk kata-kata singkatan dan slang.

Septian (2019) melakukan penelitian dengan mengimplementasikan metode *Naïve Bayes Classifier* untuk melakukan analisis sentimen pada media Twitter milik Divisi Humas Polri dimana *tweet* yang akan dianalisis diklasifiasikan menjadi tiga topik: kegiatan polisi, layanan masyarakat dan komentar masyarakat. Hasil akurasi pengujian klasifikasi pada sistem ini adalah 86%.

2.2 Dasar Teori

2.2.1 Instagram

Diluncurkan pada Oktober 2010, Instagram adalah layanan jejaring sosial dimana pengguna dapat berbagi foto dan video ke pengikutnya (*follower*) dari beragam perangkat (Agustina, 2016).

Menurut data dari Statista pada bulan Juli 2019, tercatat bahwa Indonesia menempati posisi keempat di dunia dengan jumlah pengguna aktif Instagram sekitar 59 juta orang (statista.com, 2019).

Beberapa fitur yang ada pada Instagram:

1. Pengikut, sistem sosial di dalam Instagram adalah dengan menjadi mengikuti akun pengguna lainnya, atau memiliki pengikut Instagram.
2. Mengunggah Foto, foto yang ingin diunggah dapat diperoleh melalui kamera iDevice ataupun foto-foto yang ada di album foto di iDevice tersebut.
3. Kamera, penggunaan kamera melalui Instagram juga dapat langsung menggunakan efek-efek yang ada, untuk mengatur pewarnaan dari foto yang dikehendaki oleh sang pengguna. Ada juga efek kamera *tilt-shift* yang fungsinya adalah untuk memfokuskan sebuah foto pada satu titik tertentu.
4. Efek Foto, Instagram memiliki 16 efek foto yang dapat digunakan oleh para pengguna pada saat mereka hendak menyunting fotonya: X-Pro II, Lomo-fi, Earlybird, Sutro, Toaster, Brannan, Inkwell, Walden, Hefe, Nashville, 1977, Lord Kelvin, Valencia, Amaro, Rise dan Hudson.

5. Judul Foto, para pengguna dapat memasukkan judul untuk menamai foto tersebut sesuai dengan apa yang ada dipikiran para pengguna sebelum mengunggah foto tersebut.
6. Label Foto, adalah sebuah kode yang memudahkan para pengguna untuk mencari foto tersebut dengan menggunakan "kata kunci". Bila para pengguna memberikan label pada sebuah foto, maka foto tersebut dapat lebih mudah untuk ditemukan

2.2.2 Analisis Sentimen

Analisis sentimen mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik, dan text mining yang bertujuan menganalisa pendapat, sentimen, evaluasi, sikap, penilaian dan emosi seseorang apakah pembicara atau penulis berkenaan dengan suatu topik, produk, layanan, organisasi, individu, ataupun kegiatan tertentu (Liu, 2012).

Dengan pertumbuhan media sosial yang begitu pesat, para individu dan organisasi menggunakan konten dari media sosial untuk pengambilan keputusan. Pada saat ini, jika seseorang ingin membeli sebuah produk, dia tidak terbatas hanya bertanya pada salah satu teman dan keluarga untuk meminta pendapat karena ada banyak ulasan pengguna dan diskusi pada forum web. Untuk organisasi, mereka tidak harus melakukan survei untuk mengumpulkan opini publik karena hal tersebut sudah banyak tersedia di internet.

Analisis sentimen terdiri dari tiga level analisis yaitu:

1. Level Dokumen

Level dokumen menganalisis satu dokumen penuh dan mengklasifikasikan 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 negatif. Sentimen netral berarti kalimat tersebut bukan opini (Liu, 2012).

3. Level Entitas dan Aspek

Level aspek tidak melakukan analisis pada konstruksi bahasa (dokumen, paragraf, kalimat, 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).

2.2.3 *Preprocessing*

Tahap *preprocessing* atau praproses data merupakan proses untuk mempersiapkan data mentah sebelum dilakukan proses lain. Pada umumnya, praproses data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem. Praproses sangat penting dalam melakukan analisis sentimen, terutama untuk

media sosial yang sebagian besar berisi kata - kata atau kalimat yang tidak formal dan tidak terstruktur serta memiliki *noise* yang besar (Clark, 2003).

Tahap *preprocessing* memiliki beberapa tahap proses sebagai berikut:

- *Case Folding*, bertujuan membuat semua teks menjadi huruf kecil.
- *Remove Punctuation*, bertujuan menghapus semua karakter *non alphabet* misalnya simbol, spasi, dan lain-lain.
- *Remove Hashtag*. *Hashtag* merupakan suatu penunjuk sebuah kata yang dibicarakan oleh sesama pengguna Instagram yang memiliki simbol "#". Biasanya akan digunakan sebagai judul topik pembicaraan dan juga berfungsi sebagai pengelompokan terhadap percakapan yang berhubungan dengan kata yang diberi simbol *hashtag*. Proses ini juga dapat dikategorikan antara penting dan tidak penting, dapat dilakukan ataupun tidak dilakukan proses *Remove Hashtag*.
- *Removal* URL. URL yang terdapat pada data komentar Instagram membuat data tidak efektif dan tidak memiliki arti. Untuk itu perlu adanya penghapusan URL tersebut. Kemunculan alamat web atau URL ini disebabkan karena banyaknya pengguna mempromosikan sebuah produk pada situs mereka sehingga pengguna lain langsung bisa masuk pada halaman web yang dimaksud.
- *Remove Mention*. Pada Instagram, simbol "@" digunakan untuk menunjuk atau mengajak teman berkomunikasi langsung. Pada suatu

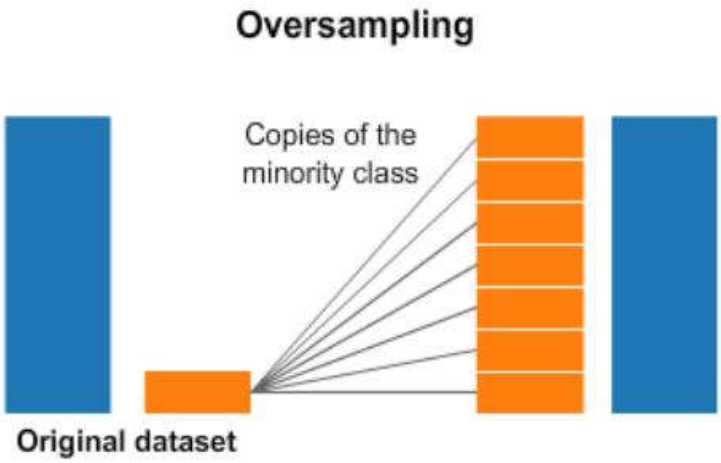
analisis sentimen, nama pengguna tidak diperhatikan sehingga perlu dihapus.

- *Convert Word*. Saat ini penggunaan bahasa gaul mengakibatkan penggunaan Bahasa Indonesia tidak baku. Sehingga *convert word* diperlukan untuk mengkonversi kata yang tidak baku menjadi kata baku yang sesuai dengan KBBI.
- *Convert Emoticon* dan *Emoji*. Seringnya ekspresi diungkapkan dengan sebuah gambar (*Emoji*) atau simbol karakter *keyboard* (*Emoticon*) menyebabkan perlu adanya pengkonversian ke dalam bentuk *string* yang dapat diartikan maknanya.
- *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 sehingga perlu dilakukan penghapusan. Untuk melakukan proses penghapusan kata ini diperlukan sebuah data atau daftar kata yang diinginkan untuk dihapus.
- *Stemming*, digunakan untuk mendapatkan kata dasar dari suatu kata. Hal ini dilakukan untuk menormalisasi kata.

2.2.4 *Random Oversampling* dan Prinsip Pareto

Random Oversampling adalah salah satu metode *random resampling* yang digunakan untuk menyeimbangkan jumlah data pada *dataset* pelatihan dengan cara menduplikat contoh (data) pada kelas minoritas/kelas dengan

jumlah data paling sedikit dalam *dataset* pelatihan (Brownlee, 2020). Lihat Gambar 2.1.



Gambar 2.1 Ilustrasi *Random Oversampling*

Prinsip Pareto (juga dikenal sebagai aturan 80/20, *the law of the vital few*, atau *the principle of factor sparsity*) menyatakan bahwa untuk banyaknya kejadian, sekitar 80% efek berasal dari 20% penyebabnya. Pada tahun 1906, Pareto melakukan pengamatan bahwa 20 persen populasi memiliki 80 persen properti di Italia. Hal ini menuntunnya untuk menemukan pola yang sama di berbagai bidang kehidupannya. Konsep Pareto ini kemudian digeneralisasi oleh seorang konsultan manajemen Joseph M. Juran dengan nama Prinsip Pareto (Medina, 2019). Prinsip Pareto dapat diterapkan dalam menentukan ukuran pembagian *dataset* menjadi data pelatihan dan data pengujian.

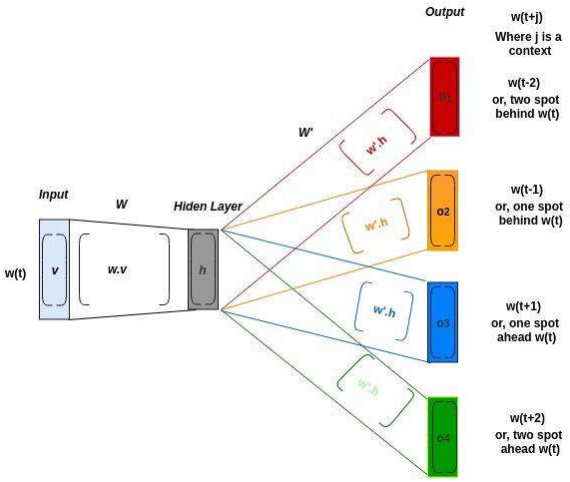
2.2.5 *Word Embedding*

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 (Karani, 2018).

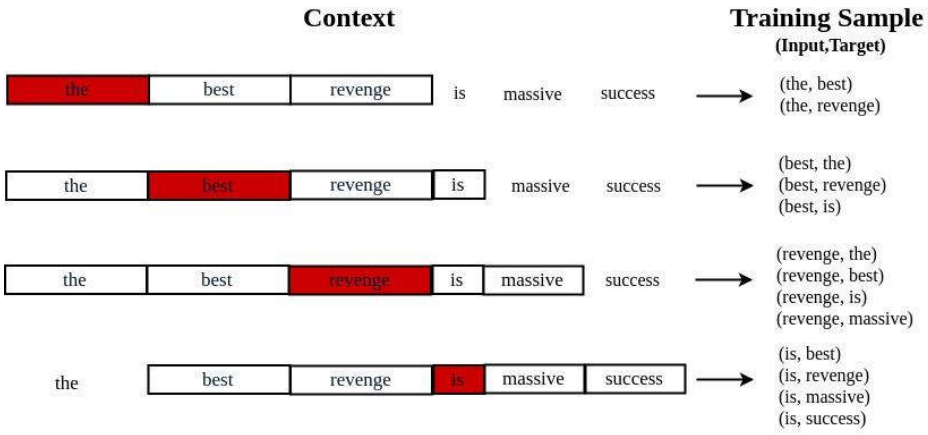
Word embedding adalah representasi teks dimana teks yang memiliki arti yang sama memiliki representasi yang hampir sama. Dalam *word embedding*, kata – kata direpresentasikan ke dalam sebuah vektor. Setiap kata mewakili satu vektor kemudian vektor – vektor ini akan dilatih pada jaringan saraf tiruan. Representasi kata dilatih berdasarkan penggunaan kata – kata. Hal ini menjadikan kata – kata yang memiliki makna yang sama akan memiliki representasi yang sama.

Salah satu model *word embedding* yang digunakan adalah *word2vec*: *Skip-gram* model. Model ini diperkenalkan oleh Tomas Mikolov pada tahun 2013 (Mikolov dkk, 2013). Ilustrasi *feeding forward Skip-Gram* dengan *windows* (jarak antara kata-kata konteks dengan posisi kata yang menjadi *input*-an) = 2 dapat dilihat pada Gambar 2.2 (Romadhan, 2018).



Gambar 2.2 Arsitektur Model *Skip-gram*

Secara arsitektur *Skip-gram* menggunakan *current word* [sebagai *input*, w(t)] untuk memprediksi konteks [sebagai target, w(t+j)] disekitarnya, di mana *Skip-gram* akan mempelajari distribusi probabilitas dari kata-kata di dalam konteks dengan *windows* yang telah di tentukan. Misal konteks yang digunakan saat ini adalah "the best revenge is massive success" dengan nilai *windows* = 2. Lihat Gambar 2.3.



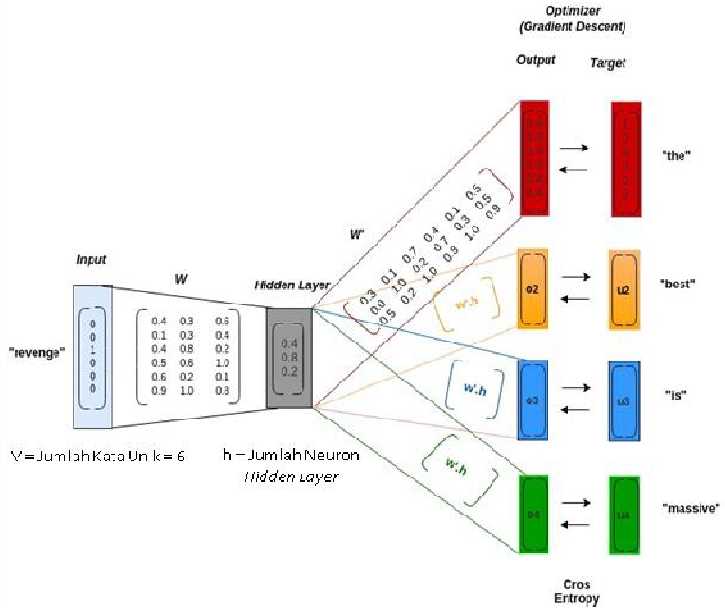
Gambar 2.3 Ilustrasi *Input-Target* Pada *Skip-gram*

Untuk merepresentasikan konteks ke dalam arsitektur *Skip-gram*, maka kita harus merubah setiap kata menjadi *one-hot encoded vectors*. Lihat Gambar 2.4.

```
the     = [1,0,0,0,0,0]
best    = [0,1,0,0,0,0]
revenge = [0,0,1,0,0,0]
is      = [0,0,0,1,0,0]
massive = [0,0,0,0,1,0]
success = [0,0,0,0,0,1]
```

Gambar 2.4 *One-Hot Encoded Vectors*

Misal *current word* = **revenge**, maka ilustrasi *Skip-gram* akan menjadi seperti pada Gambar 2.5.



Gambar 2.5 *Feed Forward Neural Network Skip-gram*

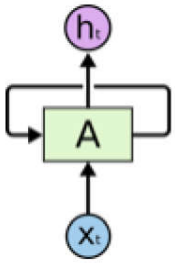
Inisialisasi bobot pada W dan W' adalah *random*. Bobot W dan W' merupakan matrik dengan ukuran W = V x h dan W' = h x V. Pada proses *feedforward*, vektor *input* akan di *dot product* dengan bobot W dan menghasilkan nilai pada *hidden layer*. Kemudian *hidden layer* di *dot product* dengan bobot W' dan menghasilkan vektor *output*. Setelah mendapatkan nilai *output* pada tahap *feedforward*, maka akan dihitung nilai *error*-nya dengan menggunakan metode *cross entropy*.

Selanjutnya adalah tahap *backpropagation* dengan memanfaatkan teknik *gradient descent* yaitu dengan melakukan *update* bobot W dan W'. Proses ini akan diulang kembali ke tahap *feedforward* hingga tercapai nilai *error* minimum.

Setelah didapatkan nilai *error* minimum pada *cross entropy*, maka vektor yang merepresentasikan kata tersebut diambil dari bobot W dengan cara mengalikan *dot product* antara *one-hot encoded vector* masing-masing kata dengan bobot W, sedangkan bobot pada W' akan diabaikan.

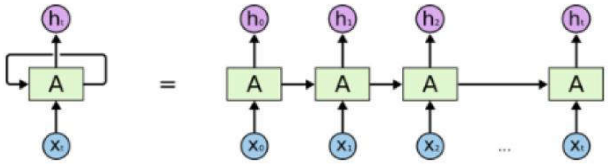
2.2.6 *Recurrent Neural Network* (RNN)

Recurrent Neural Network atau RNN merupakan bagian dari jaringan saraf tiruan untuk pemrosesan data sekuensial. Berbeda dengan *Convolutional Neural Network* (CNN) yang merupakan metode untuk pemrosesan data seperti gambar, RNN sering digunakan untuk memproses nilai yang bersifat sekuensial x^1, ..., ...,..., x^n (Olah, 2015).



Gambar 2.6 Bentuk Konseptual Sederhana RNN

Pada Gambar 2.6, x_t merupakan masukan dan h_t merupakan keluaran. Pengulangan ini membuat informasi dapat dilewatkan dari satu langkah pada jaringan ke selanjutnya. Pengulangan ini membuat RNN dapat dijabarkan sebagai banyak salinan dari jaringan yang sama. Lihat Gambar 2.7.

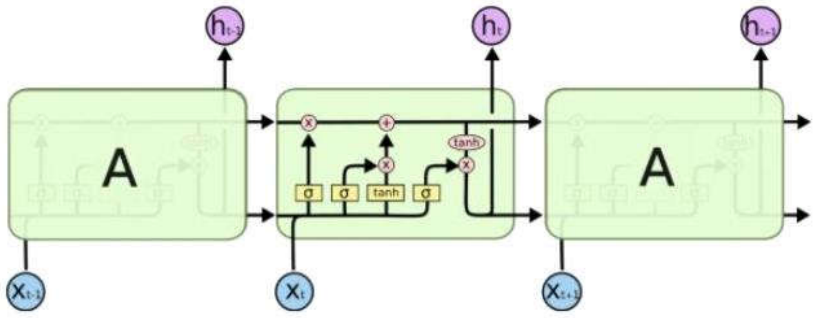


Gambar 2.7 Struktur RNN yang Dijabarkan

2.2.7 *Long Short Term Memory*

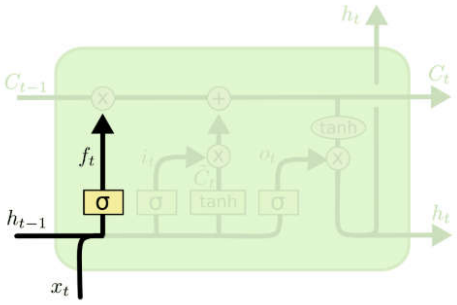
Long Short Term Memory atau biasa disingkat LSTM adalah bentuk spesial dari RNN yang dapat melakukan pembelajaran pada dependensi jangka panjang (*long-term dependencies*). Model ini diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 (Olah, 2015).

Semua *recurrent neural network* memiliki bentuk rangkaian modul jaringan saraf yang berulang. LSTM juga memiliki struktur yang sama namun memiliki tambahan fitur berupa gerbang pada sel. Jaringan saraf ini terdiri dari 3 gerbang: (1) gerbang *forget* untuk menentukan apa yang relevan untuk disimpan dari setiap proses sebelumnya, (2) gerbang *input* untuk menentukan informasi apa yang relevan untuk ditambahkan dari proses saat ini, dan (3) gerbang *output* untuk menentukan *hidden state* yang akan digunakan pada proses selanjutnya; serta 2 fungsi aktivasi: (1) *sigmoid* untuk menentukan data mana yang perlu disimpan dan data mana yang perlu dilupakan, dan (2) *tanh* untuk mendistribusikan gradien agar tidak terjadi *vanishing*/*exploding* gradien. Lihat Gambar 2.8.



Gambar 2.8 Detail Struktur LSTM

LSTM akan menentukan informasi apa yang akan dibuang dari sel. Keputusan ini dibuat oleh gerbang *forget* (f_t). *Layer* ini akan mengambil nilai *hidden state* lama (h_{t-1}) dan masukkan sekarang (x_t) sebagai masukkan fungsi *sigmoid* sehingga akan menghasilkan keluaran antara 0 dan 1. Fungsi aktivasi ini berfungsi untuk menentukan apakah nilai pada *state* sel lama (C_{t-1}) perlu dilupakan atau tetap disimpan. Keluaran 0 merepresentasikan bahwa nilai C_{t-1} akan dilupakan sedangkan keluaran 1 merepresentasikan bahwa nilai C_{t-1} tetap disimpan. Lihat Gambar 2.9.

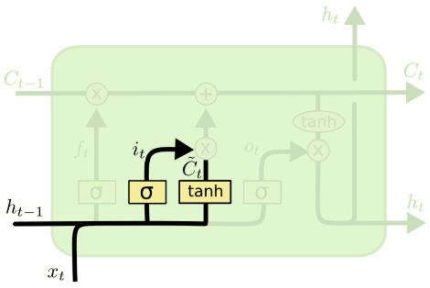


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Gambar 2.9 Struktur Gerbang *Forget*

Langkah selanjutnya adalah menentukan nilai yang akan disimpan pada sel (C_t). Pertama, sebuah fungsi *sigmoid* pada gerbang *input* (i_t), dengan masukkan h_{t-1} dan x_t, menentukan nilai mana yang akan diperbarui dengan mengubah nilai menjadi antara 0 (berarti tidak penting) dan 1 (berarti penting). Selanjutnya sebuah fungsi *tanh*, dengan masukkan yang sama yaitu h_{t-1} dan x_t, membuat vektor dari nilai kandidat baru ($\tilde{C}_t$) yang dapat ditambah ke C_{t-1}. Setelah nilai keluaran *sigmoid* dan *tanh* telah didapat, kalikan nilai keluaran *tanh* dengan nilai keluaran *sigmoid*. Keluaran dari *sigmoid* ini akan menentukan

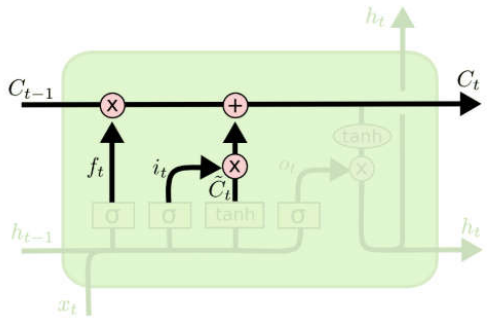
informasi mana yang penting untuk disimpan dari keluaran *tanh*. Lihat Gambar 2.10.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Gambar 2.10 Struktur Gerbang *Input*

Selanjutnya, C_{t-1} akan diperbarui ke *state* sel baru (C_t). Pertama, f_t akan dikalikan dengan C_{t-1} dengan mengabaikan informasi yang sudah dilupakan sebelumnya. Lalu, keluaran dari i_t ditambahkan dengan $\tilde{C}_t$ yang akan memperbarui *state* sel dengan nilai baru (C_t) yang menurut jaringan saraf relevan. Lihat Gambar 2.11.

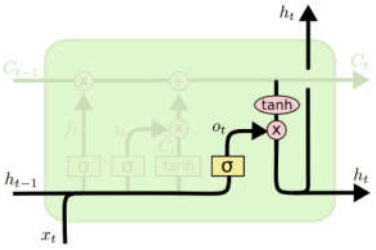


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Gambar 2.11 Menghitung *State* Sel Baru (C_t)

Langkah terakhir adalah menentukan *hidden state* selanjutnya (h_t) sebagai keluaran dari gerbang *output* (o_t). *Hidden state* mengandung informasi dari masukkan sebelumnya. *Hidden state* juga digunakan untuk melakukan prediksi. Pertama, *sigmoid* pada o_t akan menentukan bagian dari sel yang akan

dikeluarkan. *Sigmoid* akan diberikan masukkan berupa h_{t-1} dan x_t. Kemudian, C_t akan dimasukkan ke dalam fungsi *tanh*. Lalu kalikan keluaran dari *tanh* dengan keluaran dari *sigmoid* untuk menentukan informasi apa yang perlu dibawa oleh h_t. Keluaran dari perkalian dua keluaran fungsi tersebut adalah h_t. C_t dan h_t kemudian dibawa ke *node* LSTM berikutnya. Lihat Gambar 2.12.



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Gambar 2.12 Struktur Gerbang *Output*

2.2.8 Scikit-learn

Scikit-learn adalah modul Python yang mengintegrasikan berbagai algoritma pembelajaran *machine learning* untuk masalah berskala menengah yang diawasi dan tidak terawasi. Paket ini berfokus pada "membawa *machine learning* ke non-spesialis" menggunakan *general-purpose high-level language*. Kelebihan dari modul ini ada pada kemudahan penggunaan, kinerja, dokumentasi, dan konsistensi API. Modul ini memiliki ketergantungan minimal dan didistribusikan di bawah lisensi BSD yang disederhanakan, mendorong penggunaannya dalam pengaturan akademik dan komersial (Pedregosa, 2011).

2.2.9 TextBlob

TextBlob merupakan *library* Python yang digunakan untuk memproses data tekstual. *Library* ini menyediakan API sederhana untuk menyelam ke dalam tugas *Natural Language Processing* (NLP). Contohnya seperti

pemberian tag kata, ekstraksi kata benda, analisis sentimen, klasifikasi, terjemahan dan lain sebagainya (Li, 2019).

2.2.10 Keras

Keras merupakan *library deep learning* yang ditulis dalam bahasa pemrograman *Python* dan berjalan diatas Tensorflow, CNTK, atau Theano. *Library* ini berfokus pada peningkatkan kemudahan pada percobaan (Tanner, 2019).

Beberapa keuntungan menggunakan Keras yaitu:

- Memungkinkan pembuatan prototype dengan cepat dan mudah.
- Mendukung jaringan konvolusional dan jaringan rekuren dan kombinasi keduanya.
- Dapat berjalan pada CPU dan GPU.

2.2.11 *Framework* Flask

Flask merupakan *Framework Python* berbasis pada Werkzeug, Jinja2, dan terinspirasi dari *Framework* Ruby Sinatra. *Framework* ini dikembangkan oleh Pocoo milik Armin Ronacher. Meskipun usia Flask lebih muda dibandingkan dengan *Framework Python* lainnya, Flask telah mendapat popularitas di antara pengembang web berbasis *Python* (Irsyad, 2018).

Flask didesain untuk dapat dengan mudah digunakan dan dikembangkan. Ide dibalik pengembangan Flask adalah membangun dasar yang kuat untuk aplikasi dengan kompleksitas yang beragam. Pengguna dapat dengan mudah menambahkan ekstensi yang dibutuhkan. Flask sangat bagus untuk semua jenis

projek terutama untuk *prototyping*. Flask bergantung pada dua *library*, Jinja2 sebagai *template engine* dan Werkzeug WSGI.