BAB 2
LANDASAN TEORI

Analisis sentimen adalah sebuah metode NLP (*Natural Language Processing*) untuk melihat bagaimana perasaan individu tentang hal-hal tertentu. Kemajuan analisis sentimen telah dibuat lebih mudah dengan banyaknya data teks *online*, terutama dalam hal berspekulasi tentang sikap, opini, dan kepercayaan orang. Analisis sentimen telah digunakan secara luas untuk meramalkan sentimen dan tren publik dalam berbagai skenario. Terdapat banyak algoritma yang dapat digunakan untuk melakukan analisis sentimen, salah satunya adalah algoritma *deep learning* LSTM. LSTM sudah terbukti merupakan salah satu metode *deep learning* terbaik untuk melakukan analisis sentimen [9] dan [10].

2.1 Deep learning

Deep learning adalah cabang dari *machine learning* [12]. Ini adalah algoritma yang mencoba menggunakan abstraksi data tingkat tinggi menggunakan beberapa lapisan pemrosesan yang terdiri dari struktur kompleks atau beberapa transformasi nonlinier. Dalam *deep learning* algoritma didasarkan pada karakterisasi data *training*. Konsep *deep learning* sedikit berbeda dengan *shallow learning*, contohnya *Support Vector Machine* dan *Logistic Regression*. Model *shallow learning* ini hanya memiliki satu lapisan atau tidak memiliki node lapisan tersembunyi. *Deep learning* didasarkan pada beberapa node lapisan tersembunyi [13]. Inti dari *deep learning* adalah jaringan node yang berlapis-lapis. *Deep learning* menggunakan *input* dari lapisan sebelumnya sebagai *output* dari lapisan berikutnya untuk mempelajari fitur data yang sangat abstrak.

2.2 Analisis Sentimen

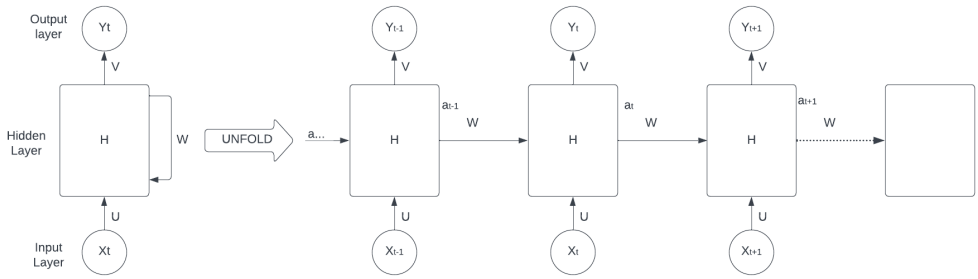
Analisis sentimen adalah teknik *Natural Language Processing* (NLP) yang digunakan untuk menentukan apakah sentimen yang mendasari adalah positif atau negatif. Informasi subjektif dari teks dapat diekstraksi menggunakan analisis sentimen dengan mengenali konteks dan posisinya. Data dari berbagai sumber, seperti komentar media sosial, berita, ulasan konsumen, dan banyak lagi, dapat digunakan untuk analisis sentimen.

Analisis sentimen menggunakan algoritma yang berbeda untuk menganalisis kata, frasa, dan konteks yang tersedia dalam teks dan prosedur yang berbeda untuk menentukan keseluruhan sentimen yang dikomunikasikan. Terdapat berbagai cara untuk melakukan analisis sentimen, mulai dari metode *rule-based* yang menggunakan daftar istilah positif dan negatif sebagai data berlabel untuk melatih algoritma pembelajaran mesin hingga membangun sebuah klasifikasi [14]. Memahami sentimen sosial, maksud yang mendasari, dan respons terhadap berbagai karakteristik manusia dapat dilakukan dengan bantuan analisis sentimen, yang membantu dalam pengambilan keputusan.

Analisis sentimen menyatukan berbagai bidang penelitian seperti *Natural Language Processing* (NLP), *data mining*, dan *text mining*, dan dengan cepat menjadi sangat penting bagi organisasi karena organisasi tersebut berusaha untuk mengintegrasikan metode kecerdasan komputasi ke dalam operasi yang telah dibuat, dan berusaha untuk menjelaskan lebih lanjut, dan meningkatkan, produk dan layanan mereka [15]. Dalam analisis sentimen, atau penggalian opini, tujuannya adalah untuk menemukan opini orang yang diungkapkan dalam bahasa tertulis (teks). Sentimen secara istilah berarti ”apa yang dirasakan seseorang tentang sesuatu”, ”pengalaman pribadi, perasaan seseorang”, ”sikap terhadap sesuatu” atau ”pendapat”.

2.3 Recurrent Neural Network

Recurrent Neural Network (RNN) [16] merupakan adalah jenis *Neural Network* dimana *output* dari langkah sebelumnya dimasukkan sebagai *input* ke langkah yang akan diambil. Pada *neural network* tradisional, semua *input* dan *output* tidak bergantung satu sama lain. Namun, dalam kasus-kasus ketika diperlukan untuk memprediksi kata berikutnya dari sebuah kalimat, kata-kata sebelumnya diperlukan dan karenanya ada kebutuhan untuk mengingat kata-kata sebelumnya. Maka muncullah RNN, yang memecahkan masalah ini dengan bantuan *Hidden Layer* (Lapisan Tersembunyi). Fitur utama dan paling penting dari RNN adalah *Hidden Layer*, yang mengingat beberapa informasi tentang suatu urutan. Keadaan ini juga disebut sebagai *Memory State* karena mengingat *input* sebelumnya ke jaringan. RNN menggunakan parameter yang sama untuk setiap *input* karena melakukan tugas yang sama pada semua *input* atau *Hidden Layer* untuk menghasilkan *output*. Hal ini mengurangi kompleksitas parameter, tidak seperti *neural network* lainnya.



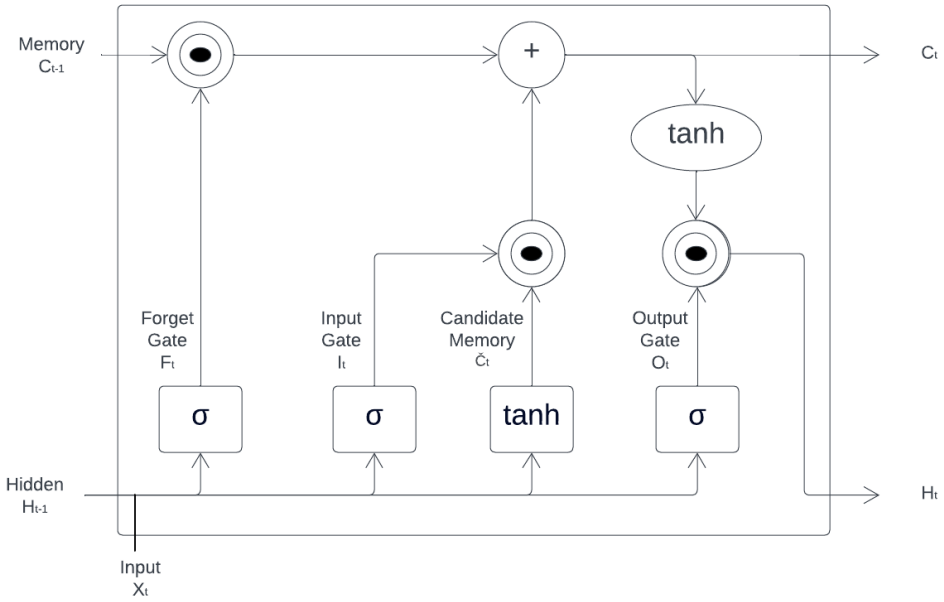
Gambar 2.1. Arsitektur RNN

Sumber: Geeks For Geeks

Gambar 2.1 merupakan arsitektur dari RNN. RNN menerima *input* X dan menghasilkan *output* Y dengan memindai data secara berurutan dari kiri ke kanan, dengan setiap langkah memperbarui keadaan tersembunyi dan menghasilkan *output* [16]. Semua ini memiliki parameter yang sama di semua langkah waktu. Hal ini berarti bahwa setiap langkah memiliki set parameter yang sama, yang diwakili oleh U, V, W digunakan secara konsisten di seluruh jaringan. U mewakili parameter bobot yang mengatur koneksi dari *input* layer X ke hidden layer H, W mewakili bobot yang terkait dengan koneksi diantara hidden layer, dan V untuk koneksi dari hidden layer H ke *output* layer Y. Pembagian parameter ini memungkinkan RNN untuk secara efektif menangkap ketergantungan temporal dan memproses data sekuensial secara lebih efisien dengan mempertahankan informasi dari *input* sebelumnya dalam kondisi tersembunyinya saat ini.

2.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) merupakan jenis *Recurrent Neural Network* (RNN) yang kuat dan cocok untuk menangani data berurutan (*sequential data*) dengan ketergantungan jangka panjang (*long term depedencies*). LSTM adalah jenis RNN yang dirancang untuk mengatasi masalah gradien yang hilang, yang merupakan masalah (keterbatasan) umum pada RNN. LSTM memiliki arsitektur khusus bernama *memory cell* yang memungkinkannya mempelajari *long-term dependencies* dalam urutan data sehingga cocok untuk tugas-tugas seperti penerjemahan mesin, pengenalan suara, dan pembuatan teks. Hal tersebut merupakan alasan mengapa penelitian ini menggunakan algoritma LSTM, bukan RNN.



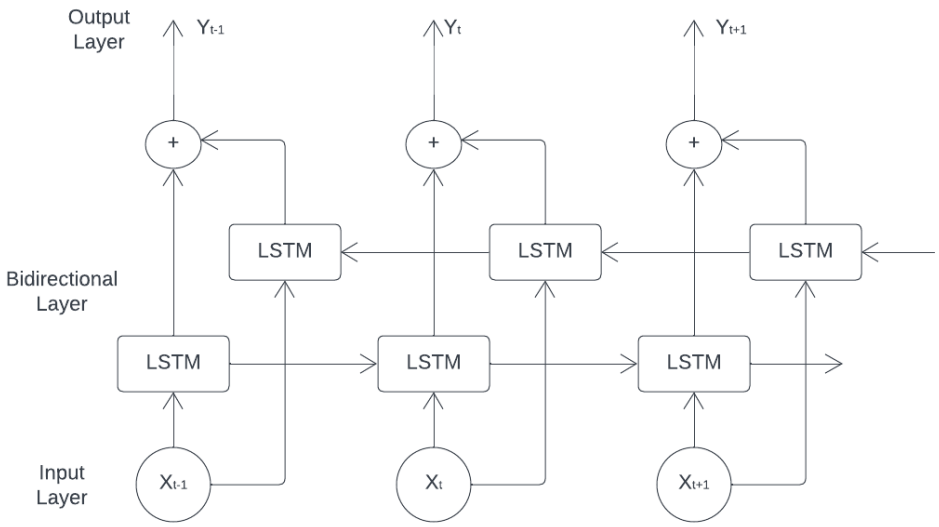
Gambar 2.2. Arsitektur Unidirectional LSTM

Sumber: Geeks For Geeks

Pada Gambar 2.2 menunjukkan arsitektur dari LSTM yang terdiri dari *input gate*, *forget gate* dan *output gate*. *Forget gate* merupakan tempat terjadinya proses pemilihan informasi yang ada pada *cell state* dengan menggunakan persamaan 1. Informasi akan dibuang dari *cell state* jika *forget gate* bernilai 0, sebaliknya informasi akan disimpan *cell state* jika *forget gate* bernilai 1. Lalu pada *input gate*, informasi akan melewati 2 lapisan yaitu *sigmoid* dan *tanh*. *Cell state* dihasilkan dari nilai *output* kedua lapisan yang telah digabungkan. Kemudian dilakukan proses perbaruan nilai *cell state*. Selanjutnya akan melewati *output gate*, yang mana didalamnya nilai *output cell state* akan ditentukan. Lapisan *sigmoid* digunakan untuk memilih *output* berdasarkan *cell state* yang ada, dan selanjutnya akan diteruskan ke lapisan *tanh*.

Terdapat dua jenis LSTM, yaitu Uni-LSTM (*Unidirectional* LSTM) dan Bi-LSTM (*Bidirectional* LSTM) [7]. Pada Gambar 2.2 menunjukkan arsitektur dari algoritma *Unidirectional* LSTM, sedangkan pada Gambar 2.3 menunjukkan arsitektur dari algoritma *Bidirectional* LSTM [17]. Dapat dilihat bahwa perbedaan utama dari Uni-LSTM (*Unidirectional* LSTM) dan Bi-LSTM (*Bidirectional* LSTM) adalah Uni-LSTM hanya memproses urutan *input* dalam satu arah, dari awal hingga akhir, sedangkan Bi-LSTM memproses urutan *input* dalam dua arah, dari awal hingga akhir dan dari akhir hingga awal. Uni-LSTM hanya dapat memanfaatkan

informasi dari kondisi masa lalu, sedangkan Bi-LSTM dapat memanfaatkan informasi dari kondisi masa lalu dan masa depan untuk menangkap ketergantungan jangka panjang dan konteks dalam urutan *input* sehingga cocok untuk tugas-tugas yang melibatkan *Natural Language Processing* dan analisis data [18]. Hal tersebut merupakan beberapa alasan mengapa penelitian ini menggunakan Bi-LSTM.



Gambar 2.3. Arsitektur Bidirectional LSTM

Sumber: Geeks For Geeks

2.5 Natural Language Processing

Natural Language Processing (NLP) adalah bagian dari kecerdasan buatan (*Artificial Intelligence*) dan linguistik, yang dikhususkan untuk membuat komputer memahami pernyataan atau kata-kata yang ditulis dalam bahasa manusia. NLP dibuat untuk memudahkan pekerjaan pengguna dan untuk memuaskan keinginan untuk berkomunikasi dengan komputer dalam bahasa alami [19]. Dengan kata lain, tujuan dari *Natural Language Processing* adalah untuk mengakomodasi satu atau lebih spesialisasi algoritma atau sistem. Metrik penilaian NLP pada sistem algoritmik memungkinkan integrasi pemahaman bahasa (*language understanding*) dan pembuatan bahasa (*language generation*). Terdapat banyak hal yang dapat diterapkan dengan NLP, seperti mendeteksi *spam*, menjadi *virtual agents* atau *chatbots*, sampai analisis sentimen [20].

Dalam bahasa pemrograman Python, terdapat banyak *tools* dan *libraries* yang disediakan untuk penggunaan NLP pada bidang tertentu. Semua hal tersebut dapat ditemukan di *Natural Language Toolkit* (NLTK), yang merupakan koleksi *open source* dari *libraries*, program, dan edukasi pembelajaran untuk membuat sistem NLP. Terdapat banyak hal yang dapat dilakukan menggunakan NLTK, seperti penguraian kalimat, segmentasi kata, *stemming* dan *lemmatization* (metode pemangkasan kata hingga ke akarnya), dan *tokenization* (untuk memecah frasa, kalimat, paragraf, dan bagian menjadi token yang membantu komputer memahami teks dengan lebih baik) [20]. NLTK juga mencakup pustaka untuk mengimplementasikan kemampuan seperti penalaran semantik, kemampuan untuk mencapai kesimpulan logis berdasarkan fakta yang diekstrak dari teks.