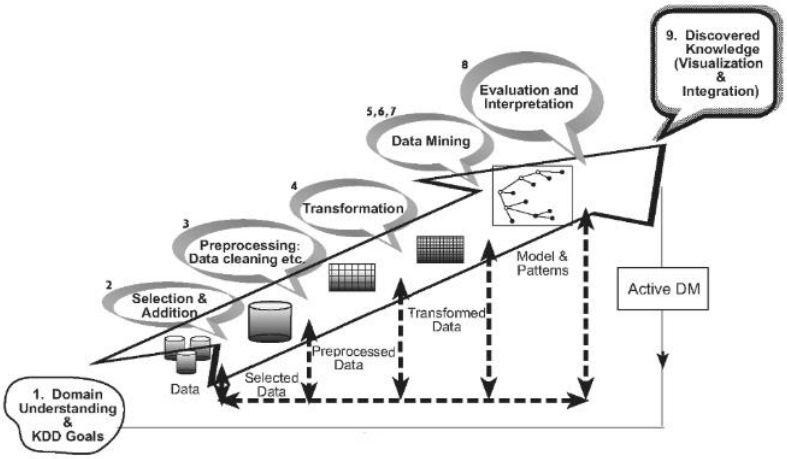
BAB 2
LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen adalah seperangkat metode dan teknik untuk mendeteksi dan mengekstraksi informasi subjektif seperti opini dan sikap dari sebuah bahasa. Analisis sentimen juga akan menentukan apakah seseorang memiliki sentimen positif atau negatif terhadap sesuatu [19]. Dalam pengumpulan data, analisis sentimen menggunakan berbagai algoritma yang berasal dari cabang dari *Artificial Intelligence*, seperti algoritma *Machine Learning* yang terdiri dari *Naïve Bayes, Support Vector Machine (SVM), Decision Trees*, dan regresi, serta algoritma *Deep Learning* yang terdiri dari *Recurrent Neural Network (RNN), Deep Neural Network (DNN),Deep Belief Network (DBN), dan Convolutional Neural Network (CNN*) [20].

2.2 Knowledge Discovery in Database

Knowledge Discovery in Databases (KDD) merupakan proses untuk menemukan pola yang menarik atau pengetahuan dari sejumlah sumber data seperti database, teks, gambar, Web, dll. Pola atau pengetahuan yang didapat haruslah valid, berpotensi berguna, dan dapat dimengerti [18].



Gambar 2.1 Proses Knowledge Discovery in Database

Berikut adalah tahapan KDD:

1) *Domain Understanding & KDD Goals*, Pada tahap ini, kita mengembangkan pemahaman tentang aplikasi yang digunakan. Ini adalah langkah persiapan awal untuk memahami apa yang harus dilakukan dengan mempertimbangkan berbagai faktor seperti transformasi, algoritma, dan representasi data. Orang yang bertanggung jawab atas proyek KDD harus memahami dan menetapkan tujuan pengguna akhir serta lingkungan tempat proses penemuan pengetahuan akan dilakukan, termasuk pengetahuan sebelumnya yang relevan. Selama proses KDD, mungkin diperlukan revisi dan penyesuaian pada langkah ini.
2) *Selection*, Setelah tujuan ditetapkan, data yang akan digunakan untuk penemuan pengetahuan harus ditentukan. Ini meliputi identifikasi data yang tersedia, memperoleh data tambahan yang diperlukan, dan mengintegrasikan semua data ke dalam satu kumpulan data, termasuk atribut yang akan dipertimbangkan. Proses ini penting karena Data Mining mengandalkan data yang tersedia untuk belajar dan menemukan pola. Jika beberapa atribut penting hilang, keseluruhan studi bisa gagal. Oleh karena itu, sebanyak mungkin atribut harus dipertimbangkan pada tahap ini. Namun, mengumpulkan, mengatur, dan mengoperasikan repositori data yang kompleks memerlukan biaya, sehingga perlu ada keseimbangan antara usaha dan hasil yang diharapkan. Proses ini dimulai dengan kumpulan data terbaik yang tersedia dan kemudian berkembang dengan mengamati efeknya dalam penemuan dan pemodelan pengetahuan.
3) *Preprocessing*, Pada tahap ini, keandalan data ditingkatkan. Ini termasuk pembersihan data, seperti menangani nilai yang hilang dan menghilangkan noise atau outlier.
4) *Transformation,* Tahap ini melibatkan persiapan dan pengembangan data yang lebih baik untuk penambangan data. Metode yang digunakan di sini meliputi pengurangan dimensi (seperti pemilihan dan ekstraksi fitur, serta pengambilan sampel rekaman) dan transformasi atribut (seperti diskritisasi atribut numerik dan transformasi fungsional). Langkah ini sering penting

untuk keberhasilan keseluruhan proyek KDD, tetapi biasanya sangat spesifik untuk proyek tertentu. Misalnya, dalam pemeriksaan medis, rasio antara atribut mungkin sering menjadi faktor yang paling penting. Dalam pemasaran, kita mungkin perlu mempertimbangkan efek di luar kendali kita serta aspek temporal (seperti mempelajari efek kumulatif iklan). Meskipun transformasi yang tepat mungkin tidak digunakan di awal, kita dapat memperoleh efek mengejutkan yang mengisyaratkan perlunya transformasi tertentu pada iterasi berikutnya. Proses KDD bersifat reflektif dan membantu memahami transformasi yang dibutuhkan

5) *Data Mining - Choosing the appropriate Data Mining task,* Memilih jenis penambangan data yang akan digunakan, seperti klasifikasi, regresi, atau klastering. Ini sangat tergantung pada tujuan KDD dan langkah-langkah sebelumnya. Ada dua tujuan utama dalam penambangan data: prediksi dan deskripsi. Prediksi sering disebut sebagai penambangan data terawasi, sedangkan deskripsi mencakup aspek penambangan data tak terawasi dan visualisasi. Sebagian besar teknik penambangan data didasarkan pada pembelajaran induktif, di mana model dibangun dengan menggeneralisasi dari sejumlah contoh pelatihan. Asumsi dasar pendekatan induktif adalah bahwa model yang dilatih dapat diterapkan pada kasus-kasus masa depan. Strategi ini juga mempertimbangkan pembelajaran meta untuk kumpulan data tertentu yang tersedia.
6) *Data Mining – Choosing Alghoritm,* Tahap ini melibatkan pemilihan metode spesifik untuk pencarian pola, termasuk beberapa penginduksi. Untuk setiap strategi pembelajaran meta, ada beberapa kemungkinan cara untuk mencapainya. Pembelajaran meta berfokus pada penjelasan mengapa suatu algoritma penambangan data berhasil atau tidak dalam masalah tertentu. Pendekatan ini mencoba memahami kondisi di mana algoritma penambangan data paling tepat. Setiap algoritma memiliki parameter dan taktik pembelajaran.
7) *Data Mining – Employing the Data Mining Alghoritm.* Pada langkah ini, algoritma mungkin perlu digunakan beberapa kali sampai hasil yang

memuaskan diperoleh, misalnya dengan menyetel parameter kontrol algoritma, seperti jumlah minimum instance dalam satu daun pohon keputusan.

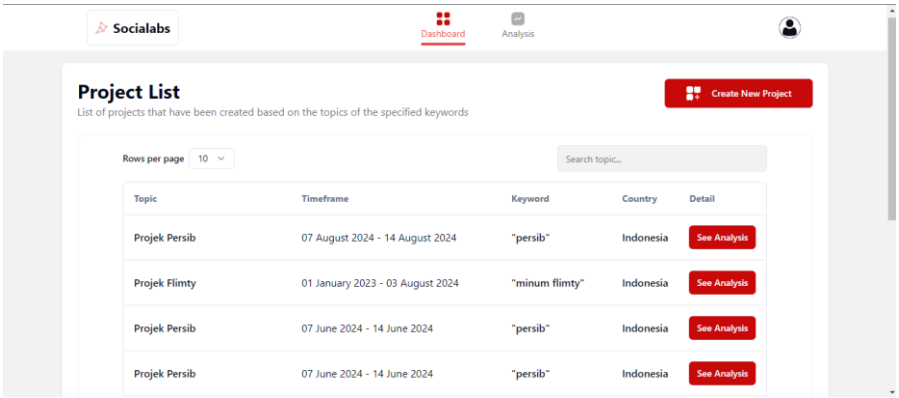
8) *Evaluation,* Pada tahap ini, kita mengevaluasi dan menafsirkan pola yang ditemukan (aturan, keandalan, dll.) sehubungan dengan tujuan yang ditetapkan pada langkah pertama. Di sini, kita mempertimbangkan langkah-langkah prapemrosesan sehubungan dengan efeknya pada hasil algoritma penambangan data (misalnya, menambahkan fitur di Langkah 4, dan mengulanginya dari sana). Langkah ini berfokus pada pemahaman dan kegunaan model yang diinduksi. Pengetahuan yang ditemukan juga didokumentasikan untuk penggunaan lebih lanjut.
9) *Visualization & Integration,* Pada tahap ini, pengetahuan menjadi aktif dalam arti bahwa kita dapat membuat perubahan pada sistem dan mengukur efeknya. Keberhasilan langkah ini menentukan efektivitas keseluruhan proses KDD. Ada banyak tantangan dalam langkah ini, seperti kehilangan kondisi laboratorium tempat beroperasi. Misalnya, pengetahuan ditemukan dari snapshot statis tertentu (biasanya sampel) dari data, tetapi sekarang data menjadi dinamis. Struktur data dapat berubah (atribut tertentu menjadi tidak tersedia), dan domain data dapat dimodifikasi (seperti atribut yang memiliki nilai yang tidak diasumsikan sebelumnya).

2.3 X (Twitter)

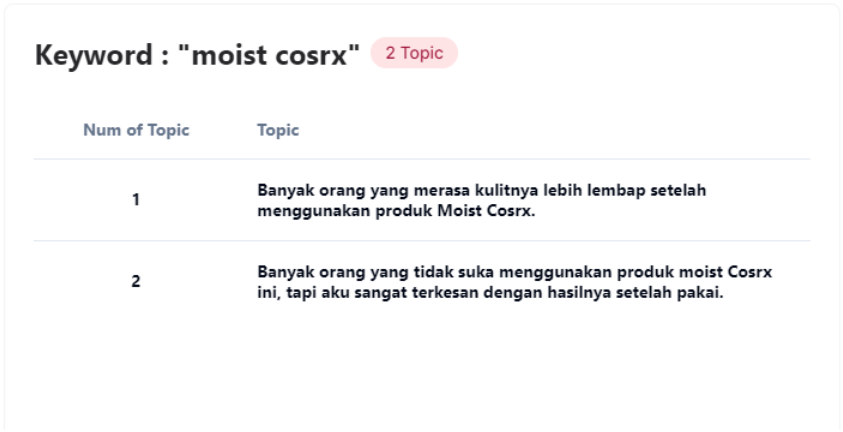
X (Twitter) adalah media sosial yang memungkinkan pengguna mengirim dan membaca postingan melalui microblogging. Microblog adalah sejenis alat komunikasi online yang memungkinkan pengguna memperbarui pemikiran dan tindakan orang tentang topik atau fenomena tertentu. Postingan ditampilkan sebagai teks dengan panjang maksimal 280 karakter di halaman profil pengguna. Meskipun postingan dapat dilihat oleh semua orang, pengirim dapat membatasi pesannya ke daftar teman mereka. Pengikut adalah istilah yang digunakan oleh pengguna lain untuk menunjukkan postingan mereka.

2.4 Aplikasi Analisis Sosial Media

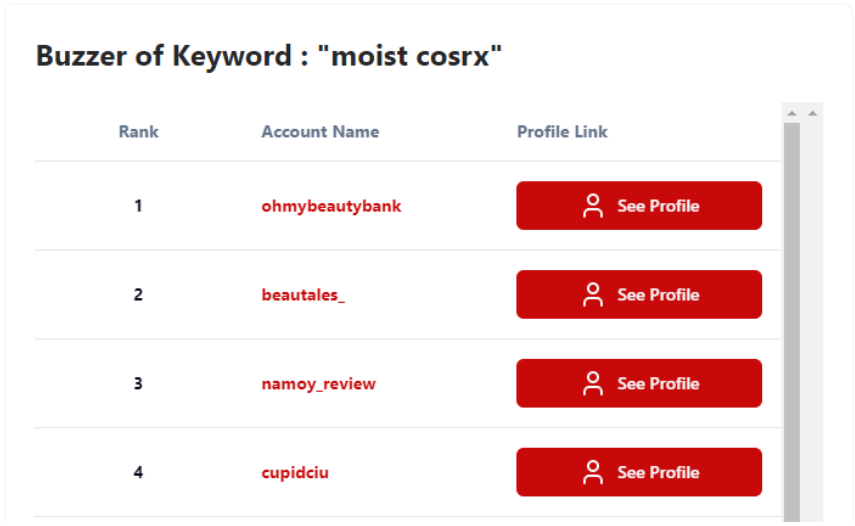
Aplikasi Analisis Sosial Media adalah alat yang membantu dalam menganalisis data dari media sosial seperti Twitter. Dengan dua fitur utamanya, aplikasi ini memungkinkan pengguna untuk mengidentifikasi topik-topik yang ada dalam dokumen atau data melalui fitur Topic Modelling. Ini dilakukan dengan menganalisis teks dari tweet atau dokumen dan menghasilkan kumpulan kalimat yang menggambarkan topik utama. Selain itu, terdapat fitur Analisis Jaringan Sosial memanfaatkan informasi dari model topik media sosial Twitter untuk menampilkan data tentang interaksi pengguna, seperti frekuensi retweet, balasan, atau kehadiran dalam percakapan tentang topik tertentu. Dengan menggunakan kedua fitur ini, pengguna dapat menggali informasi yang lebih dalam dari data media sosial untuk memahami topik yang sedang dianalisis. Berikut adalah tampilan dari aplikasi analisis sosial media saat ini:



Gambar 2.2 Halaman Dashboard Aplikasi Analisis Sosial Media



Gambar 2.3 Fitur Topic Modelling Pada Aplikasi Analisis Sosial Media



Gambar 2.4 Fitur Buzzer Pada Aplikasi Analisis Sosial Media

2.5 Topic Modelling

Topic modeling adalah metode untuk menganalisis kumpulan dokumen teks yang dikelompokkan menjadi beberapa topik. Ini termasuk dalam pendekatan clustering dalam penelitian pembelajaran mesin. Latent Dirichlet Allocation (LDA), yang diusulkan oleh Blei dan Jordan, adalah model probabilistik generatif yang digunakan untuk mencari struktur semantik dari kumpulan korpus berdasarkan analisis hierarchical bayesian. LDA menggabungkan kumpulan dokumen menjadi kumpulan topik campuran yang berisi kata-kata yang memiliki

probabilitas tertentu untuk muncul. Ini adalah teknik pemodelan topik yang paling umum digunakan [21].

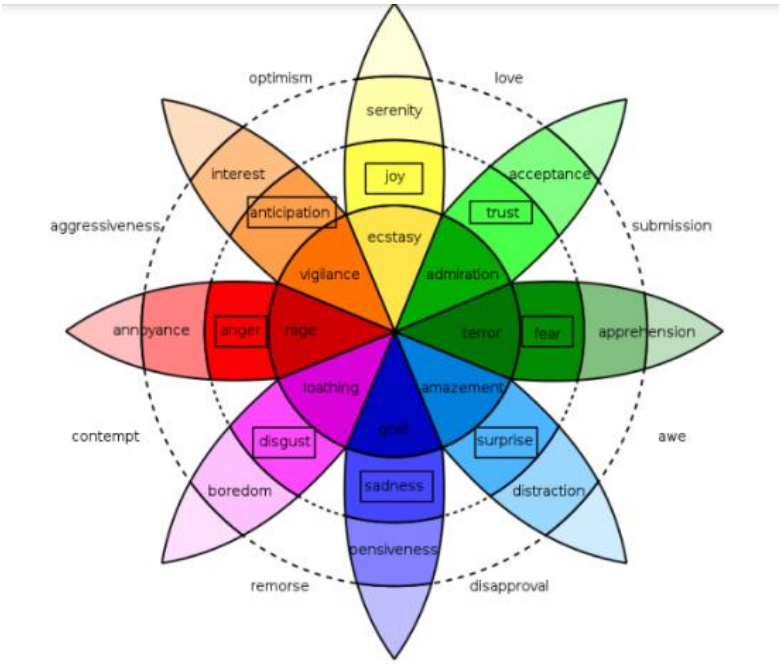
2.6 Text Mining

Text mining adalah proses yang membutuhkan pengetahuan yang mana pengguna berinteraksi dengan kumpulan dokumen menggunakan seperangkat alat analisis. Sebagaimana dalam data mining, text mining bertujuan untuk mengekstraksi informasi yang berguna dari sumber data dengan mengidentifikasi dan mengeksplorasi pola-pola menarik. Namun, dalam text mining, sumber data berupa koleksi dokumen, dan pola-pola menarik ditemukan bukan di antara catatan formal basis data tetapi dalam data teks yang tidak terstruktur dalam dokumen-dokumen tersebut.

Data mining dan text mining memiliki kesamaan. Tujuan mereka sama: mendapatkan pengetahuan dan informasi dari sekumpulan data yang besar. Text mining menggunakan data yang tidak terstruktur, sedangkan data mining menggunakan data yang sudah terstruktur [22]

2.7 Teori Emosi

Emosi adalah respons kompleks yang melibatkan pengalaman subjektif, respons fisiologis, dan ekspresi perilaku. Emosi memainkan peran penting dalam mempengaruhi perilaku manusia dan interaksi sosial. Menurut teori emosi dasar oleh Robert Plutchik, ada delapan emosi dasar yang dapat dikombinasikan untuk membentuk emosi yang lebih kompleks. Plutchik mengilustrasikan teori ini melalui model yang dikenal sebagai **Plutchik's Wheel of Emotions** [23]. Sebagaimana yang ditunjukkan pada **Gambar 2.5 Wheel of Emotion** berikut:



Gambar 2.5 Wheel of Emotion

Dalam analisis sentimen, emosi dapat dikategorikan menjadi dua kategori utama: **positif** dan **negatif**. Kategorisasi ini bertujuan untuk menyederhanakan analisis dan interpretasi data. Berdasarkan Plutchik's Wheel of Emotions, emosi-emosi dapat dikategorikan sebagai berikut pada **Tabel 2.1 Deskripsi Emosi Berdasarkan Sentimen**:

Tabel 2.1 Deskripsi Emosi Berdasarkan Sentimen

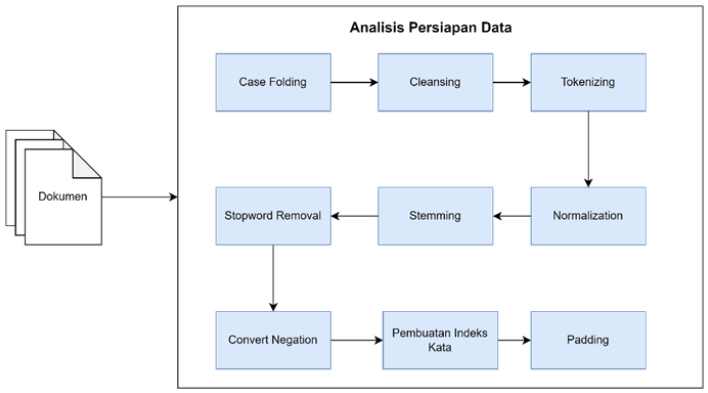
Sentimen	Emosi	Dekripsi
Positif	Love	Perasaan kasih sayang yang mendalam, keterikatan, dan perhatian. Dalam analisis sentimen, tweet yang mengekspresikan cinta sering mengandung kata-kata yang menyampaikan kehangatan dan koneksi positif.

	Joy	Perasaan gembira, bahagia, dan senang. Tweet yang mengekspresikan kegembiraan sering kali mengandung kata-kata yang menunjukkan kesenangan, kebahagiaan, dan kenikmatan.
	Anticipation	Harapan atau keinginan untuk hasil yang positif. Tweet dengan antisipasi sering mengandung kata-kata yang mengekspresikan kegembiraan atau ketidaksabaran untuk peristiwa yang akan datang.
	Trust	Keyakinan pada seseorang atau sesuatu. Tweet yang mengekspresikan kepercayaan sering kali memiliki kata-kata yang menunjukkan keandalan, keamanan, dan kepastian.
Negatif	Fear	Dalam konteks sosial media, emosi ketakutan seringkali muncul dalam tweet yang mengungkapkan kekhawatiran, kecemasan, atau ketidakamanan.
	Sadness	Perasaan sedih atau tidak bahagia. Tweet yang mengekspresikan kesedihan sering kali mengandung kata-kata yang menyampaikan rasa kehilangan, kekecewaan, atau kesedihan.

	Disgust	Penolakan atau kejijikan yang kuat. Tweet yang mengekspresikan jijik sering kali mengandung kata-kata yang menunjukkan penolakan, penghinaan, atau ketidaksukaan.
	Surprise	Tweet yang mengekspresikan kejutan negatif sering mengandung kata-kata yang menunjukkan peristiwa negatif yang tidak terduga.

2.8 Text Preprocessing

Text preprocessing adalah bagian penting dari *Text Mining*, yang merupakan proses pengolahan teks yang bertujuan untuk mengubah dokumen menjadi data terstruktur sesuai kebutuhan agar dapat diolah lebih lanjut dalam proses text mining [24]. Langkah pra-proses teks dalam klasifikasi bertujuan untuk meningkatkan akurasi klasifikasi data dengan melakukan pemrosesan terhadap kalimat dan kata-kata yang memiliki imbuhan. Tahap *text preprocessing* dapat dilihat pada **Gambar 2.6 Tahap Preprocessing** berikut



Gambar 2.6 Tahap Preprocessing

Pada umumnya, *text preprocessing* terdiri dari *case folding, tokenizing, stopwords,* dan *stemming.* Pada penelitian ini, tahapan *text-preprocessing* meliputi *case folding, cleansing, tokenizing, normalization, stemming, stopword removal,convert negation, pembuatan indeks kata, dan padding.* Keseluruhan tahapannya memiliki perannya masing-masing yang mana hal tersebut bertujuan agar dataset yang diolah memiliki dimensi lebih kecil daripada dataset sebelumnya.

2.8.1 Case Folding

Case folding merupakan proses penyeragaman karakter pada teks atau dokumen menjadi huruf kecil [25]. Berikut adalah contoh penerapan *case folding* yang dapat dilihat pada **Tabel 2.2 Contoh Penerapan *Case Folding***

Tabel 2.2 Contoh Penerapan *Case Folding*

Sebelum Case Folding	Setelah Case Folding
Baru saja menonton film horor yang seruww banget! Gak bisa tidur nich!	baru saja menonton film horor yang seruww banget! gak bisa tidur nich
Senang sekali bisa berkumpul dengan keluarga di akhir pekan ini.	senang sekali bisa berkumpul dengan keluarga di akhir pekan ini.
Sedang menikmati secangkir kopi di pagi yang cerah. Semangat untuk hari ini!	sedang menikmati secangkir kopi di pagi yang cerah. semangat untuk hari ini
Hari ini cuaca sangat panas sekali. Jangan lupa minum air putih ya, guys!	hari ini cuaca sangat panas sekali. jangan lupa minum air putih ya, guys
Merasa terharu mendengar lagu ini. Musisi Indonesia memang luar biasa!	merasa terharu mendengar lagu ini. musisi indonesia memang luar biasa!

2.8.2 Cleansing

Cleansing adalah proses menghilangkan karakter-karakter yang tidak diinginkan atau tidak relevan dari teks, seperti tanda baca,angka,atau simbol khusus [25]. Adapun simbol yang dapat dihapus untuk data tweet yaitu *username,url,* dan *hashtag.* Berikut merupakan contoh penerapan dari *cleansing* yang dapat dilihat pada **Tabel 2.3 Contoh Penerapan *Cleansing***

Tabel 2.3 Contoh Penerapan *Cleansing*

Sebelum Cleansing	Setelah Cleansing

Baru nonton film seruww nich! @User123 #FilmSeru	Baru saja menonton film seruww! Gak bisa tidur nich!
Keluarga aku kumpul di akhir pekan ini. #QualityTime	Senang berkumpul keluarga akhir pekan ini.
Nikmatin secangkir kopi di pagi yang cerah. #SemangatPagi	Menikmati secangkir kopi pagi cerah. Semangat hari ini!
Hari ini cuaca panas banget! Jangan lupa minum air putih, ya! #CuacaPanas	Hari ini cuaca sangat panas. Jangan lupa minum air putih, guys!
Terharu denger lagu ini. #MusikIndonesia	Terharu denger lagu ini.

2.8.3 Tokenizing

Tokenizing adalah proses memecah teks menjadi unit-unit yang lebih kecil yang disebut token [25]. Token bisa berupa kata-kata, frasa, atau simbol-simbol terentu. Pada penelitian ini *tokenizing* dilakukan untuk memisahkan kata yang dipisahkan oleh spasi atau simbol untuk data tweet nya. Berikut adalah contoh penerapan dari *tokenizing* yang bisa dilihat pada **Tabel 2.4 Contoh Penerapan *Tokenizing***

Tabel 2.4 Contoh Penerapan *Tokenizing*

Sebelum Tokenizing	Setelah Tokenizing
Baru nonton film seruww nich	["Baru", "nonton", "film", "seruww", "nich"]
Keluarga aku kumpul di akhir pekan ini	["Keluarga", "aku", "kumpul", "di", "akhir", "pekan", "ini"]
Nikmatin secangkir kopi di pagi yang cerah	["Nikmatin", "secangkir", "kopi", "di", "pagi", "yang", "cerah"]
Hari ini cuaca panas banget! Jangan lupa minum air putih, ya	["Hari", "ini", "cuaca", "panas", "banget", "Jangan", "lupa", "minum", "air", "putih", "ya"]
Terharu denger lagu ini.	["Terharu", "denger", "lagu", "ini"]

2.8.4 Normalization

Normalisasi digunakan untuk membuat istilah yang memiliki arti serupa namun ditulis dengan cara yang berbeda menjadi seragam. Ini bisa disebabkan oleh kesalahan penulisan, singkatan kata, atau bahasa informal [25]. Berikut adalah

contoh penerapan dari *normalization* yang bisa dilihat pada **Tabel 2.5 Contoh Penerapan *Normalization*.**

Tabel 2.5 Contoh Penerapan *Normalization*

Sebelum Normalization	Setelah Normalization
Baru nonton film seruww nich	["Baru", "nonton", "film", "seru", "nih"]
Keluarga aku kumpul di akhir pekan ini	["Keluarga", "aku", "kumpul", "di", "akhir", "pekan", "ini"]
Nikmatin secangkir kopi di pagi yang cerah	["Nikmatin", "secangkir", "kopi", "di", "pagi", "yang", "cerah"]
Hari ini cuaca panas banget! Jangan lupa minum air putih, ya	["Hari", "ini", "cuaca", "panas", "banget", "Jangan", "lupa", "minum", "air", "putih", "ya"]
Terharu denger lagu ini.	["Terharu", "denger", "lagu", "ini"]

2.8.5 Stemming

Stemming adalah proses dalam pemrosesan bahasa alami yang digunakan untuk menghilangkan infleksi dan afiks dari kata, sehingga hanya meninggalkan akar kata atau "stem" [25]. Berikut adalah contoh penerapan dari *stemming* yang bisa dilihat pada **Tabel 2.6 Contoh Penerapan *Stemming*.**

Tabel 2.6 Contoh Penerapan *Stemming*

Sebelum Normalization	Setelah Stemming
Baru nonton film seruww nich	["Baru", "nonton", "film", "seru", "nih"]
Keluarga aku kumpul di akhir pekan ini	["Keluarga", "aku", "kumpul", "di", "akhir", "pekan", "ini"]
Nikmatin secangkir kopi di pagi yang cerah	["Nikmat", "secangkir", "kopi", "di", "pagi", "yang", "cerah"]
Hari ini cuaca panas banget! Jangan lupa minum air putih, ya	["Hari", "ini", "cuaca", "panas", "banget", "Jangan", "lupa", "minum", "air", "putih", "ya"]
Terharu denger lagu ini.	["Haru", "denger", "lagu", "ini"]

2.8.6 Stopword Removal

Stopwords adalah kata-kata umum yang sering muncul dalam teks dan tidak memberikan banyak informasi penting dalam analisis teks, seperti "dan", "atau",

"di", "ke", dan sebagainya [25]. Penghapusan stopwords adalah proses mengidentifikasi dan menghapus kata-kata tersebut dari teks untuk memproses dan menganalisis teks lebih efisien dan relevan Berikut adalah contoh penerapan *stopwords removal* yang dapat dilihat pada **Tabel 2.7 Contoh Penerapan *Stopword Removal***

Tabel 2.7 Contoh Penerapan *Stopword Removal*

Sebelum Stopwords Removal	Setelah Stopwords Removal
["Baru", "nonton", "film", "seru", "nih"]	["Baru", "nonton", "film", "seru", "nih"]
["Keluarga", "aku", "kumpul", "di", "akhir", "pekan", "ini"]	["Keluarga", "aku", "kumpul", "akhir", "pekan"]
["Nikmat", "secangkir", "kopi", "di", "pagi", "yang", "cerah"]	["Nikmat", "secangkir", "kopi", "pagi", "cerah"]
["Hari", "ini", "cuaca", "panas", "banget", "Jangan", "lupa", "minum", "air", "putih", "ya"]	["Hari", "cuaca", "panas", "banget", "Jangan", "lupa", "minum", "air", "putih"]
["Terharu", "denger", "lagu", "ini"	["Haru", "denger", "lagu"]

2.8.7 Convert Negation

Mengubah kata negatif dan menggabungkannya dengan kata setelahnya untuk menjadi satu kesatuan kata disebut konversi negatif. Proses ini, misalnya, mengubah kata "tidak ada" menjadi "xada" atau “tidak_ada”, membutuhkan daftar kata negatif agar dapat menemukan kata negatif dalam data komentar. Penanda negasi umum dalam bahasa Indonesia adalah tidak, bukan, jangan, dan belum [26].

2.8.8 Pembuatan Indeks Kata

Indeks kata adalah proses mengubah setiap kata dalam korpus teks menjadi angka unik. Hal ini biasanya dilakukan dengan menggunakan tokenizer yang membuat kamus (vocabulary) dari seluruh korpus, di mana setiap kata diberi indeks yang berbeda [27].

2.8.9 Padding

Padding adalah proses menambahkan nilai-nilai (biasanya nol) ke kalimat sehingga semua kalimat memiliki panjang yang sama. Ini penting untuk batch

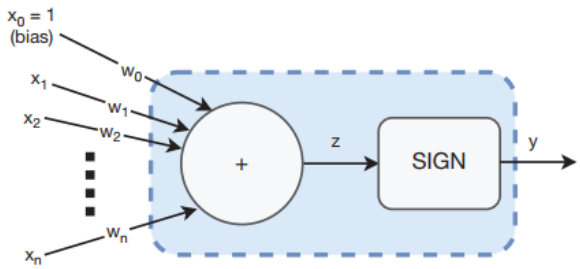
processing dalam pembelajaran mesin karena model biasanya membutuhkan input yang berukuran sama [28].

2.9 Artificial Neural Network

Artificial neural network adalah suatu sistem pengolahan informasi yang meniru prinsip kerja neuron dalam otak manusia, dengan struktur yang mirip dengan jaringan syaraf biologis. Seperti halnya otak manusia, jaringan ini terdiri dari neuron-neuron yang saling terhubung dan mengubah informasi yang diterima dengan mengirimkan sinyal keluar ke neuron lainnya. Jaringan saraf tiruan dibuat sebagai abstraksi dari model matematika yang menggambarkan pemahaman manusia. Ada dua model arsitektur jaringan saraf tiruan, yaitu lapis tunggal dan multilayer. Pada jaringan multilayer, terdapat beberapa lapisan, termasuk lapisan input, lapisan tersembunyi, dan lapisan output [29].

Unit terkecil dari neural network disebut sebagai *perceptron* yang mana *perceptron* tersebut terdiri dari tiga komponen utama yaitu nilai input, weight, dan output. Untuk menghasilkan sebuah output, setiap input akan dikalikan dengan weight dari masing-masing input tersebut, kemudian hasil dari perkalian tersebut akan dijumlahkan seperti persamaan berikut.

$$\boldsymbol{z} = (\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{w_i} \times \boldsymbol{x_i}) \qquad 2\text{-}1$$



Gambar 2.7 Model Perceptron

Keterangan:

z : merupakan total dari penjumlahan antara weight dan input
$\boldsymbol{n}$: merupakan jumlah lapisan input dari jaringan syaraf
$\boldsymbol{w_i}$: merupakan bobot (weight) yang diberikan pada masing-masing input neuron
$\boldsymbol{x_i}$: merupakan nilai input dari masing-masing neuron atau unit
$\sum$: merupakan simbol sigma yang menandakan operasi penjumlahan dari *i* = 1 hingga *n*, artinya bobot dan input akan dijumlahkan untuk setiap neuron atau unit dalam jaringan

Untuk menghasilkan sebuah output, diperlukan fungsi aktivasi yang bertujuan unuk mengubah fungsi linear pada output perceptron nya. Output dari neuron yang kini dapat dinyatakan dalam bentuk

$$y = f(z)$$

Keterangan:

y : output dari neuron
f : fungsi aktivasi
z : total dari penjumlahan antara weight dan input

2.9.1 Fungsi Aktivasi

Dalam konteks jaringan saraf tiruan, fungsi aktivasi memiliki peran penting sebagai sinyal untuk menentukan output yang akan diberikan kepada neuron-neuron lainnya. Keberadaan fungsi aktivasi ini sangat krusial dalam operasi jaringan saraf tiruan, di mana penggunaannya disesuaikan dengan kebutuhan dan tujuan spesifik, serta akan memengaruhi penentuan bobot-bobot yang digunakan. Berikut adalah beberapa jenis fungsi aktivasi yang sering digunakan. [22]

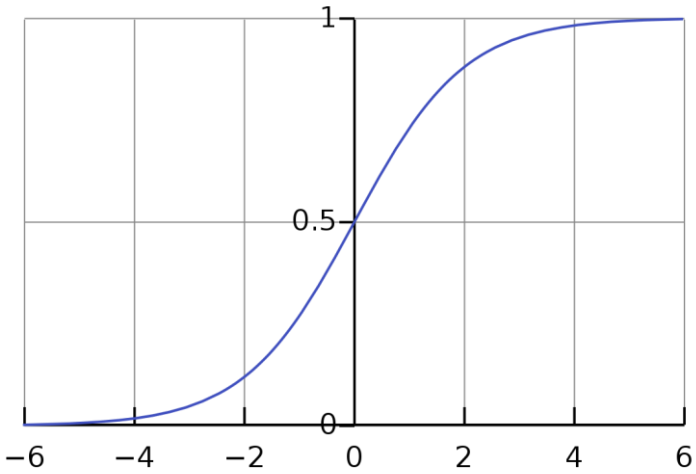
1) Fungsi Sigmoid Biner (*Logistic*)

   Fungsi sigmoid biner merupakan pilihan umum dan efektif dalam penerapan pada jaringan saraf tiruan, terutama saat algoritma pembelajarannya menggunakan metode backpropagation. Fungsi sigmoid biner memiliki rentang nilai antara 0 hingga 1, sehingga cocok digunakan pada jaringan yang menghasilkan keluaran dengan rentang nilai tersebut. Berikut adalah rumus secara sistematis untuk fungsi sigmoid biner.

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad \text{2-2}$$

$$f'(x) = \sigma f(x)[1 = f(x)] \quad \text{2-3}$$

Keterangan :
$f(x)$: fungsi aktivasi
x : jumlah sinyal-sinyal input yang terboboti
σ : laju pembelajaran (*learning rate*)

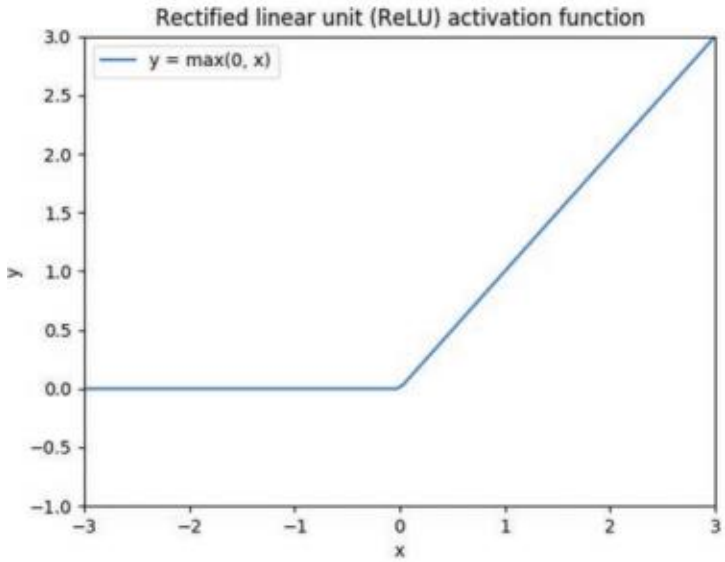


Gambar 2.8 Grafik Fungsi Aktivasi *Logistic Sigmoid*

2) Fungsi ReLu

Fungsi ReLu (rectified liner unit) merupakan fungsi non-linier dimana pengaktifan neuron tidak dilakukan secara bersamaan, dan hanya ketika output dari transformasi linier bernilai nol [30]. Dalam penggunaannya, fungsi ReLu dituliskan menggunakan persamaan

$$f(x) = \max(0, x) \quad \text{2-4}$$



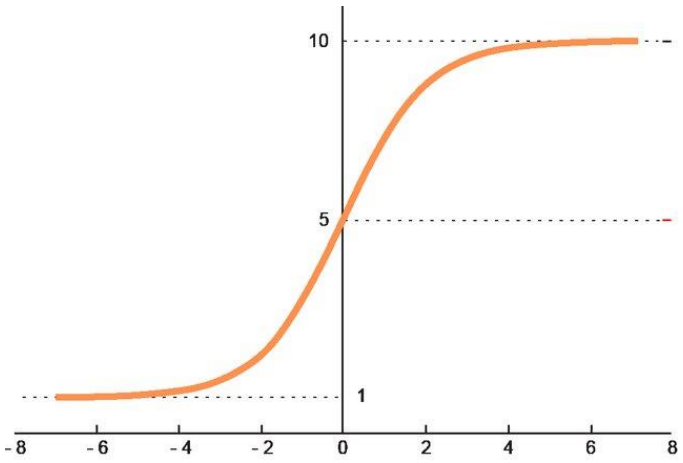
Gambar 2.9 Grafik Fungsi Aktivasi ReLu

Keterangan:
x : Nilai data input
$f(x)$: Hasil output ReLu berupa nilai dalam bentuk 0 dan 1

3) Fungsi Softmax

Softmax adalah sebuah fungsi yang mengambil vektor dari bilangan real sebanyak K sebagai input, kemudian mengonversinya menjadi distribusi probabilitas yang terdiri dari K probabilitas. Sebelum menerapkan softmax, beberapa elemen vektor mungkin negatif atau lebih besar dari satu, dan totalnya mungkin tidak sama dengan 1. Namun, setelah menerapkan softmax, setiap elemen vektor akan berada dalam rentang antara 0 dan 1, dan totalnya akan sama dengan 1. Dengan demikian, elemen vektor yang lebih besar akan memiliki probabilitas yang lebih besar. Fungsi ini dapat dirumuskan dengan persamaan sebagai berikut:

$$p(x) = \frac{e^x}{\sum_{k=1}^{k} e^x} \qquad 2\text{-}5$$



Gambar 2.10 Grafik fungsi aktivasi softmax

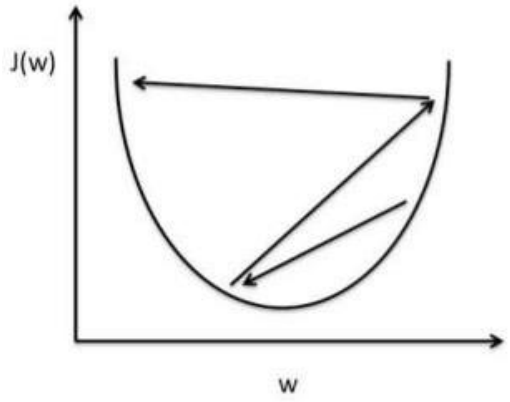
Keterangan:
e^x : Nilai eksponensial positif dari nilai data input
$\sum_{k=1}^{k} e^x$: Jumlah semua nilai eksponensial dalam data dengan jumlah sama dengan 1
$p(x)$: Probabilitas dari nilai x

2.9.2 Laju Pembelajaran

Laju pembelajaran (learning rate) adalah parameter krusial yang memengaruhi efisiensi suatu jaringan dalam mencapai target optimalnya dalam waktu yang diinginkan. Dalam konteks pengoptimalan, penting untuk memastikan bahwa perubahan bobot dan kesalahan yang dihasilkan tetap minimal. Seringkali, proses pelatihan jaringan membutuhkan banyak iterasi yang memakan waktu lama. Oleh karena itu, penggunaan parameter seperti learning rate (α) diperlukan untuk mempercepat proses iterasi atau perulangan. Nilai α umumnya berada dalam rentang antara 0 hingga 1 ($0 \leq \alpha \geq 1$) [22].

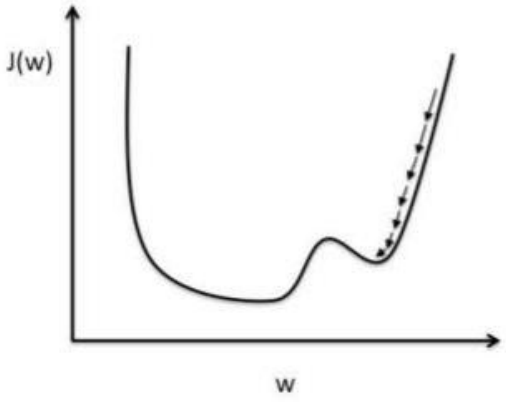
Penting untuk dipertimbangkan bahwa laju pembelajaran (learning rate) merupakan faktor kunci dalam mengatur perubahan bobot pada setiap langkah dalam proses pelatihan jaringan. Jika laju pembelajaran diatur ke nilai yang terlalu besar, penurunan gradien juga dapat melampaui batas solusi dan gagal untuk konvergen. Lebih jauh lagi, bahkan dengan ukuran langkah yang kecil, algoritma algoritma tidak dijamin untuk menemukan minimum global karena bisa terjebak

dalam minimum lokal. Namun, dalam praktiknya, algoritma ini telah terbukti bekerja dengan baik untuk jaringan saraf [22]. Berikut adalah ilustrasi untuk pemilihan α yang besar.



Gambar 2.11 Perubahan Bobot Untuk *Learning Rate* Besar

Jika pemilihan α terlalu kecil, maka akan memakan waktu cukup lama untuk konvergen, berikut adalah ilustrasinya



Gambar 2.12 Perubahan Bobot Untuk *Learning Rate* Kecil

2.9.3 Inisialisasi Bobot

Inisialisasi bobot adalah langkah penting dalam melatih jaringan saraf tiruan, karena bobot awal yang dipilih dapat memengaruhi konvergensi dan stabilitas

pelatihan model. Distribusi Glorot Uniform, yang dikembangkan oleh Xavier Glorot dan Yoshua Bengio, adalah salah satu metode inisialisasi bobot yang dirancang untuk mengatasi masalah vanishing gradient dan exploding gradient. Metode ini mengusulkan inisialisasi bobot dengan mempertimbangkan jumlah unit input dan output pada setiap lapisan jaringan, untuk menjaga konsistensi varians aktivasi dan gradien selama pelatihan. Strategi ini secara signifikan meningkatkan kinerja pelatihan pada jaringan saraf tiruan yang dalam maupun jaringan saraf tiruan berulang, sehingga lebih stabil dan efektif dalam mencapai konvergensi [31]. Berikut adalah persamaan dari inisisalisasi bobot Glorot Uniform:

$$w \sim U\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right) \tag{2-6}$$

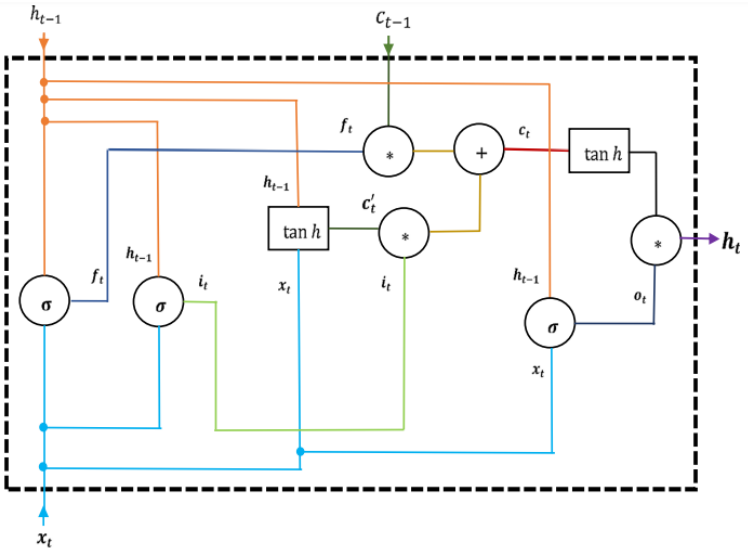
Dengan U (α, β) adalah distribusi uniform pada interval (α, β), m dan n adalah jumlah unit layer yang terhubung oleh bobot w_{ij}. Bobot w_{ij} menghubungkan neuron ke-i pada lapisan pertama (dengan jumlah m unit) ke neuron ke-j pada lapisan kedua (dengan jumlah n unit).

2.10 Deep Learning

Deep learning adalah cabang kecerdasan buatan yang memiliki kemampuan untuk belajar dari data dalam jumlah besar, menghasilkan pencapaian luar biasa di berbagai bidang, seperti pengenalan gambar, pemrosesan bahasa alami, dan pengemudian otonom. Saat ini, *deep learning* menjadi bidang penelitian yang sangat aktif, dengan menunjukkan kapasitasnya untuk melampaui model *machine learning* tradisional di berbagai domain. Perbedaan utama antara *machine learning* dan *deep learning* terletak pada pendekatan mereka terhadap ekstraksi fitur. Model *machine learning* tradisional mengandalkan ekstraksi fitur manual yang memerlukan keahlian pakar, sementara model *deep learning* mampu melakukan ekstraksi fitur secara otomatis melalui lapisan-lapisan dalamnya. Hal Ini memungkinkan model *deep learning* untuk belajar lebih efisien dan menghasilkan hasil yang lebih akurat dari data yang diberikan [32].

2.11 Long Short Term Memory (LSTM)

LSTM adalah salah satu varian RNN yang paling populer, yang memiliki kemampuan untuk menangani masalah *vanisihing gradient* pada RNN standar dan dapat menangkap ketergantungan jangka panjang [33].Berikut adalah gambaran arsitektur LSTM pada **Gambar 2.13 Arsitektur LSTM**:



Gambar 2.13 Arsitektur LSTM

Secara umum, jaringan LSTM terdiri dari blok memori yang disebut sel. Setiap sel memiliki dua state: cell state dan hidden state. Sel-sel dalam jaringan LSTM digunakan untuk membuat keputusan penting dengan menyimpan atau mengabaikan informasi tentang komponen-komponen penting. Komponen-komponen ini disebut gate, yang diatur sebagai berikut: forget gate, input gate, dan output gate [34]. Berdasarkan **Gambar 2.13 Arsitektur LSTM** Model LSTM mengoperasikan 3 tahap:

1. Tahap pertama, jaringan bekerja dengan forget gate untuk memeriksa jenis informasi apa yang perlu diabaikan atau disimpan untuk status sel. Perhitungan dimulai dengan mempertimbangkan input pada langkah waktu saat ini $x_t$dan nilai sebelumnya dari hidden state h_{t-1} menggunakan fungsi sigmoid S. Rumus untuk perhitungan di forget gate adalah sebagai berikut.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad \text{2-7}$$

Yang mana:

f_t = output dari forget gate pada waktu t

σ = fungsi aktivasi sigmoid

W_f = matriks bobot untuk hidden state

U_f = matriks bobot untuk input

h_{t-1} = hidden state pada waktu $t-1$

x_f = input pada waktu t

b_f = bias untuk forget gate

2. Tahap kedua, perhitungan jaringan berlanjut dengan mengubah cell state lama C_{t-1} menjadi keadaan sel baru C_t. Proses ini memilih informasi baru mana yang harus dimasukkan ke dalam memori jangka panjang (cell state). Untuk mendapatkan nilai cell state yang baru, proses kalkulasi harus memperhitungkan nilai referensi dari forget gate f_t, input gate i_t dan nilai cell update gate C'_t. Rumus untuk langkah ini ditunjukkan sebagai berikut:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad \text{2-8}$$

$$C'_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad \text{2-9}$$

$$C_t = (C_{t-1} \times f_t) + (i_t \times C'_f) \quad \text{2-10}$$

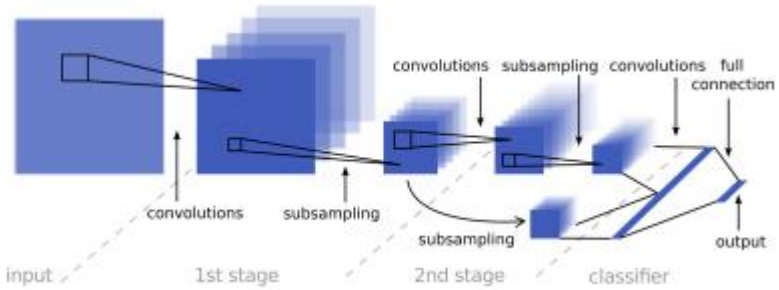
3. Tahap terakhir, setelah pembaruan cell state selesai, langkah terakhir adalah menentukan nilai hidden state h_t .Tujuan dari proses ini bertindak agar hidden state dijadikan sebagai memori jaringan, yang berisi informasi tentang data sebelumnya dan digunakan untuk prediksi. Untuk menentukan nilai hidden state, perhitungan harus memiliki nilai referensi dari cell state yang baru dan output gate. Adapun rumusnya sebagai berikut:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad 2\text{-}11$$

$$h_t = o_t \times \tanh(C_t) \quad 2\text{-}12$$

2.12 Convolutional Neural Network (CNN)

Convolutional Neural Network adalah salah satu algoritma *deep learning* yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang didesain untuk melakukan pengolahan data menjadi dua dimensi seperti gambar, suara atau teks. *Convolutional Neural Network* digunakan untuk mengklasifikasikan data yang telah dilabeli dengan menggunakan metode *supervised learning*, cara kerjanya melibatkan data latih dan target variabel yang sudah ditentukan dengan tujuan untuk mengklasifikasikan data ke dalam kategori yang sesuai [35]. Adapun untuk arsitektur CNN secara umum bisa dilihat pada **Gambar 2.14 Arsitektur CNN**:



Gambar 2.14 Arsitektur CNN [36]

Layer pada CNN terdiri dari *convolutional layer*, *pooling layer*, dan *fully connected layer*. Adapun penjelasan secara detailnya adalah sebagai berikut

1. Convolutional layer

   Convolutional layer adalah inti dari arsitektur CNN yang mana layer ini memiliki sejumlah filter yang digunakan untuk mempelajari sebuah input baik itu gambar,teks, maupun audio. Melalui layer ini, fitur akan diekstraksi dan kemudian diproses ke layer berikutnya untuk

mengekstraksi fitur yang lebih kompleks. Adapun persamaan umum dari proses ini adalah sebagai berikut:

$$Q_j = f(\sum_{i=1}^{N} I_{i,i} * K_{i,j} + B_j) \quad (2\text{-}13)$$

Yang mana:

Q_j : Output matriks dari proses konvolusi

f : Fungsi aktivasi

I : Matriks masukan

K : Matriks filter konvolusi

B_j : Nilai bias pada filter

i, i : Posisi dari matriks masukan

N : Panjang kernel

i, j : Posisi dari matriks filter konvolusi

2. Filter

   Filter dalam CNN 1D adalah sekumpulan bobot yang diterapkan pada input untuk mendeteksi fitur spesifik dalam data. Setiap filter memindai input dan melakukan operasi konvolusi untuk menghasilkan output yang disebut feature map atau activation map. Dalam konteks pemrosesan teks atau sinyal, filter dapat digunakan untuk mengenali pola seperti kata-kata, frasa, atau fitur temporal dalam sinyal.

3. Kernel

   Kernel adalah matriks bobot kecil yang digunakan dalam operasi konvolusi. Pada CNN 1D, kernel memiliki dimensi satu dimensi dan panjang tertentu, seperti (3,), (5,), atau (7,). Kernel ini bergerak melintasi input data satu langkah pada satu waktu (stride), melakukan perkalian elemen dan penjumlahan dengan bagian dari input yang dilaluinya. Hasilnya adalah nilai tunggal yang mewakili fitur yang dideteksi oleh kernel tersebut pada posisi tertentu.

4. Stride

Stride dalam konteks Convolutional Neural Networks (CNN) 1D adalah jumlah langkah pergeseran kernel saat bergerak melintasi input data selama operasi konvolusi. Stride menentukan seberapa banyak kernel bergeser setiap kali melakukan operasi konvolusi. Secara default, stride biasanya diset ke 1, yang berarti kernel bergeser satu elemen input setiap kali.

5. Padding

Padding adalah teknik menambahkan nilai ekstra di awal dan/atau akhir urutan input untuk mengontrol ukuran output setelah operasi konvolusi. Padding penting untuk memastikan bahwa informasi dari ujung-ujung input tetap dipertahankan dan untuk mengontrol ukuran output feature map. Ada dua jenis padding utama:

A. Valid Padding: Tidak menambahkan padding ke input. Outputnya lebih kecil dari input karena operasi konvolusi mengurangi panjang input.

B. Same Padding: Menambahkan padding sehingga output memiliki panjang yang sama dengan input. Ini memastikan bahwa informasi dari seluruh input diperhitungkan. Untuk konvolusi 1D dengan padding 'same', padding yang ditambahkan di kedua sisi input dihitung sebagai berikut:

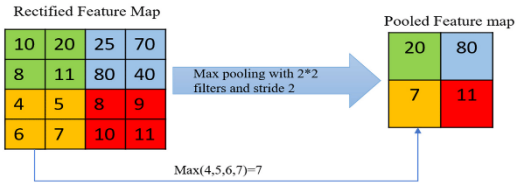
$$P = [\frac{k-1}{2}] \qquad 2\text{-}14$$

Jika ukuran kernel adalah ganjil, panjang padding di kedua sisi akan membuat panjang keluaran tetap sama dengan panjang masukan.

6. Pooling layer

Pooling layer digunakan untuk mengurangi dimensi dari *feature map*. Salah satu jenis pooling yang umum digunakan adalah Max Pooling, di mana nilai maksimum dari setiap jendela (window) pada *feature map* dipilih. Proses ini membantu mereduksi ukuran feature map sambil mempertahankan fitur-fitur penting dari input. Akibatnya, output

yang dihasilkan adalah matriks *feature map* dengan dimensi yang lebih kecil, berisi nilai maksimum yang dipilih dari setiap jendela.:



Gambar 2.15 Max Pooling [37]

7. Dropout Layer

   Pada layer dropout, sebagian dari neuron pada layer sebelumnya diatur menjadi nol secara acak selama proses pelatihan [38]. Proses ini bertujuan untuk mencegah model dari *overfitting* dengan cara memaksa model untuk tidak bergantung pada neuron tertentu dan membuatnya lebih generalisasi terhadap data baru. Dropout membantu model untuk belajar tidak terlalu spesifik terhadap data latihannya.

8. Flatten Layer

   Layer flatten berfungsi untuk meratakan (flatten) output dari layer konvolusi atau pooling menjadi vektor satu dimensi sebelum memasukkannya ke dalam layer fully connected. Hal Ini diperlukan karena layer fully connected membutuhkan input dalam bentuk vektor satu dimensi, sedangkan layer konvolusi atau pooling menghasilkan output dalam bentuk tensor multi-dimensi. Flatten mengubah tensor menjadi vektor tanpa mengubah data yang ada.

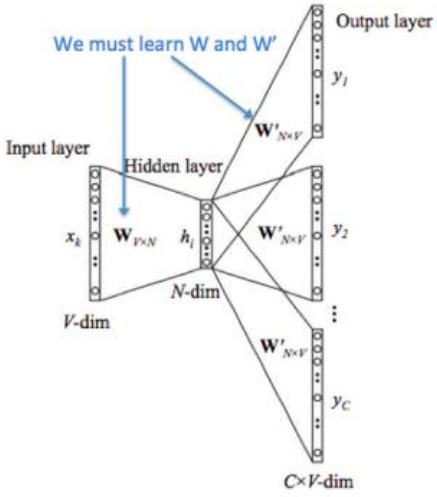
9. Fully connected layer

   Pada layer ini, *feature map* akan menjadi input untuk *fully connected layer. Layer* ini terdiri dari terdiri dari beberapa *layer* yang mana untuk setiap *layer* terdiri dari sekumpulan neuron yang akan melakukan perkalian matriks antara input matriks dengan bobotnya seperti pada jaringan *artificial neural network* umumnya. Setelah melalui beberapa *layer* tahapan terakhir adalah *output layer* yang mana *layer* tersebut bertugas sebagai klasifikasi sentimen dengan

menerapkan fungsi aktivasi sigmoid yang akan menghitung probabilitas dari setiap kelas.

2.13 Word Embedding

Word embedding adalah model pembelajaran yang menghasilkan representasi kata yang terdistribusi kontinu dalam ruang dimensi rendah. Secara umum. model pembelajaran yang digunakan adalah jaringan saraf tiruan (JST). Salah satu, metode *word embedding* yang terkenal adalah Word2Vec [39]. Word2Vec menggunakan jaringan saraf tiruan (JST) untuk menghasilkan representasi vektor kata dengan tingkat kemiripan semantik yang tinggi. Dua arsitektur yang digunakan adalah *continuous bag-of-words* (CBOW) dan *Skip-gram*. Penelitian ini menggunakan model *Skip-gram*, yang bertujuan memprediksi kata-kata di sekitar kata yang diberikan (context word). Dengan menggunakan ukuran window, model dapat menentukan kata-kata target dengan memperhitungkan jumlah kata yang diamati sebelum dan sesudah context word. Berikut adalah visualisasi neural network dari algoritma *skip-gram* pada **Gambar 2.16 Skip Gram**.



Gambar 2.16 Skip Gram

2.14 Confusion Matrix

Confusion Matrix adalah alat evaluasi kinerja model klasifikasi yang memberikan gambaran yang jelas tentang bagaimana model melakukan prediksi dibandingkan dengan label sebenarnya [40]. Matriks ini mengklasifikasikan hasil prediksi dalam empat kategori utama:

1 **True Positive (TP)**: Jumlah kasus positif yang diklasifikasikan dengan benar sebagai positif.
2 **True Negative (TN)**: Jumlah kasus negatif yang diklasifikasikan dengan benar sebagai negatif.
3 **False Positive (FP)**: Jumlah kasus negatif yang diklasifikasikan secara keliru sebagai positif (juga dikenal sebagai Type I Error).
4 **False Negative (FN)**: Jumlah kasus positif yang diklasifikasikan secara keliru sebagai negatif (juga dikenal sebagai Type II Error).

Confusion matrix adalah alat yang berguna dalam memahami sejauh mana model berfungsi dengan baik pada setiap kelas. Dari *confusion matrix,* dapat menghitung berbagai metrik kinerja model, termasuk akurasi dengan persamaan seperti berikut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \qquad 2\text{-}15$$

2.15 Pengujian Black Box

Pengujian *Black Box* merupakan metode yang banyak digunakan dalam pengujian perangkat lunak tanpa melihat kode sumber atau struktur internal dari sistem.Ada delapan metode utama dalam pengujian Black Box, *yaitu Equivalence Partitioning, Boundary Value Analysis, Cause Effect Graph, Random Data Selection, Feature Test, All-Pair Testing, Fuzzing, dan Orthogonal Array Testing*. Salah satu keunggulan dari pengujian *Black Box* adalah penguji tidak perlu memahami bahasa pemrograman atau cara kerja internal program, sehingga pengujinya tidak harus berasal dari latar belakang teknis. Namun, terdapat beberapa

kekurangan, seperti tidak semua perangkat lunak dapat diuji dengan metode ini. Salah satu metode spesifik dalam pengujian *Black Box* adalah *Equivalence Partitioning,* yang membagi data masukan ke dalam beberapa partisi ekuivalen, di mana setiap partisi merepresentasikan kelompok data yang menghasilkan output serupa. Metode ini bertujuan untuk menguji semua jenis masukan yang valid maupun tidak valid dengan mengelompokkan data berdasarkan fungsinya [41].