

## BAB II

### KAJIAN LITERATUR

#### 2.1. *State Of The Art*

Dalam pembangunan *chatbot* jasa rental alat camping ini sebelumnya terdapat beberapa penelitian dengan tema dan konsepnya yang serupa. Berikut merupakan beberapa penelitian terdahulu :

- a. Penelitian yang dilakukan pada tahun 2022 oleh Malvin, Constantine Dylan dkk, mengembangkan model sistem *chatbot* UMKM untuk layanan pelanggan dengan mengimplementasikan algoritma *Support Vector Machine* untuk mengklasifikasi teks dan memiliki hasil akurasi sebesar 87,81% dan presisi sebesar 90,65% [6].
- b. Penelitian yang dilakukan pada tahun 2021 oleh Rena Cahya Utama, dkk, membangun *chatbot* GRABS untuk meminimalisir waktu dalam menjawab pertanyaan dengan mengimplementasikan algoritma *Naive Bayes Classifier* dan menggunakan split ratio sebesar 0,8 dengan total 60 pertanyaan dalam pengujian. Hasil penelitian menunjukkan bahwa *chatbot* mencapai akurasi sebesar 93,33% dengan nilai kesalahan sebesar 6,66% [7].
- c. Penelitian yang dilakukan pada tahun 2019 oleh Uli Rizki, bertujuan untuk mengatasi multi respon maka dibuatlah pemeringkatan respon. Data yang digunakan adalah riwayat obrolan dan telah diberi label oleh peneliti. Dengan menggabungkan model *seq2seq* dan *Convolutional Neural Network* yang menghasilkan akurasi sebesar 86,7% [8].
- d. Penelitian yang dilakukan pada tahun 2021 oleh Feri Mutakim, dkk, mengembangkan model *chatbot* berbasis teks untuk mempermudah mahasiswa dan pihak kampus dalam melaksanakan kuliah daring dengan menerapkan algoritma *Artificial Neural Network*. Hasil pengujian menunjukkan bahwa model *chatbot* berbasis teks mencapai akurasi sebesar 97,27% dengan kesalahan 2,72% [9].
- e. Penelitian yang dilakukan pada tahun 2018 oleh Debys Suryani Hormansyah, dkk, Penelitian dengan tujuan membangun sebuah aplikasi *chatbot* berbasis web yang membantu pengunjung dalam mencari informasi terkait layanan

kesehatan di Kota Malang. Metode yang digunakan dalam penelitian ini adalah *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk mengolah dan mencari relevansi informasi. Hasil dari penelitian ini menunjukkan bahwa pengunjung dapat melakukan tanya jawab melalui *website* aplikasi yang telah dibuat. Sistem aplikasi juga diuji untuk ketetapan jawabannya dengan menggunakan pengujian *recall* dan *precision* [4].

- f. Penelitian yang dilakukan pada tahun 2020 oleh Daniel Adi dkk, membuat *intelligent bot* menggunakan bahasa indonesia. Algoritma CNN digunakan untuk mengenali intent dari pertanyaan yang sudah diajukan oleh *user*. Hasil terbaik dari beberapa pengujian adalah menghasilkan *precision* 93% [5].

Tabel 2.1 merupakan penjelasan singkat dari penelitian terdahulu.

No	Peneliti	Data	Metode	Tujuan
1	Malvin, Constantine Dylan, Abdul Haris Rangkuti [6]	Pertanyaan dan jawaban seputar UMKM	<i>Support Vector Machine</i>	<i>Chatbot</i> untuk mempermudah proses interaksi antara pembeli dan penjual
2	Rena Cahya Utama, Fauziah, Ratih Titi Komalasari [7]	<i>Frequently Asked Question</i> (FAQ) pada GRABDS	<i>Naive Bayes Classifier</i>	<i>Chatbot</i> otomasi layanan FAQ pada GRABDS
3	Uli Rizki [8]	10 ribu percakapan seputar pemesanan tiket	<i>Convolutional Neural Network</i>	Mengatasi multi respon dengan membuat pemeringkatan respon
4	Feri Mustakim, Fauziah, Nur Hayati [9]	<i>Frequently Asked Question</i> (FAQ) pada universitas Nasional	<i>Artificial Neural Network</i> (ANN)	<i>Chatbot</i> otomasi layanan FAQ pada universitas Nasional
5	Debys Suryani Hormansyah, Yoga Putra Utama [4]	<i>Frequently Asked Question</i> (FAQ) pada layanan publik kesehatan di malang	<i>Term Frequently Invers Document Frequently</i> (TF-IDF)	<i>Chatbot</i> otomasi layanan FAQ pada layanan publik kesehatan di malang

No	Peneliti	Data	Metode	Tujuan
6	Daniel Adi, Leo Willyanto Santoso [5]	336 Dataset pertanyaan seputar calon mahasiswa	<i>Convolutional Neural Network</i> (CNN)	Mebuat intellegent bot untuk memahami pesan (intent) yang disampaikan manusia

Tabel 2.1 State Of The Art

Berdasarkan penelitian terdahulu pada tabel 2.1 menunjukkan bahwa algoritma CNN memiliki performansi yang lebih baik, sehingga penelitian ini menggunakan algoritma CNN dan sebelumnya belum ada penelitian tentang *chatbot* UMKM jasa rental alat kamping.

## 2.2. Landasan Teori

### 2.2.1 UMKM

Menurut UU No 9 Tahun 2008 Usaha Mikro, Kecil, dan Menengah (UMKM) merujuk pada usaha produktif yang dimiliki oleh individu atau entitas usaha, yang telah memenuhi syarat sebagai usaha mikro sesuai dengan kriteria yang telah ditetapkan. Dalam konteks kemajuan suatu negara, salah satu aspek yang diperhatikan adalah perkembangan kesejahteraan masyarakat [10]. Usaha kecil adalah jenis usaha produktif yang berdiri sendiri atau merupakan badan usaha yang bukan merupakan anak perusahaan atau anak cabang dari usaha menengah maupun besa. Hal ini sesuai dengan kriteria yang telah ditetapkan dalm UU tersebut [11].

Sedangkan usaha menengah adalah usaha produktif yang dilakukan oleh perorangan atau badan usaha yang bukan merupakan anak perusahaan atau cabang dari usaha kecil atau usaha besar. Jumlah hasil tahunan usaha tersebut memenuhi kriteria yang ditetapkan sebagaimana dimaksud dari UU tersebut [11].

### 2.2.2 Chatbot

Secara sederhana, *chatbot* yaitu program komputer yang dirancang khusus untuk melakukan komunikasi dalam waktu nyata dengan pengguna melalui penggunaan bahasa alami. Awal mula *chatbot* muncul pada tahun 1960, dengan tujuan untuk membuktikan fungsi *chatbot* yang mana dapat memanipulasi pengguna dengan berkomunikasi secara spontan. *Chatbot* dikembangkan sebagai

simulasi percakapan yang sesungguhnya dimana untuk bisa berinteraksi dengan menggunakan bahasa yang digunakan pengguna [12].

*Chatbot* dapat berkomunikasi mengenai poin tertentu yang sudah diatur sebelumnya yang mana *chatbot* harus bisa mengenali dan merespon sebuah percakapan yang sudah diajukan oleh pengguna. Kemampuan *chatbot* dalam memahami dan memberikan respon ditentukan tergantung ruang lingkup dari pemahaman *chatbot* tersebut [13].

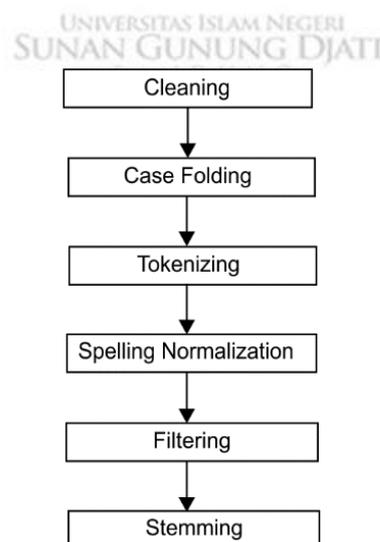
### 2.2.3 Text Mining

*Text mining* adalah penambangan dengan memanfaatkan komputer untuk mendapatkan hal baru yang belum diketahui sebelumnya ataupun untuk mengambil informasi dari sebuah data tekstual yang di ekstrak secara otomatis. *Text mining* merupakan teknik yang sering digunakan pada sistem *clustering*, *classification*, *information retrieval* dan *information extraction* [14].

Pada umumnya *text mining* memiliki dua tahapan yaitu *text preprocessing* dan *feature selection*, berikut penjelasannya :

#### a. Text Preprocessing

*Text preprocessing* merupakan tahapan utama untuk mempersiapkan teks menjadi data yang siap digunakan untuk proses selanjutnya. Proses yang dilakukan *text preprocessing* dapat dilihat pada gambar 2.1



Gambar 2.1 *Text Preprocessing* [15]

*Text preprocessing* terdiri dari beberapa proses yaitu *cleaning*, *case folding*, *tokenizing*, *spelling normalization*, *filtering* dan *stemming* [15].

1. *Cleaning* adalah proses pembersihan kata yang tidak diperlukan guna mengurangi proses *noise* pada klasifikasi. Kata yang dihilangkan yaitu simbol, *emoticon*, url, HTML.
2. *Case Foding* merupakan tahapan untuk mengubah semua huruf menjadi huruf kecil dan hanya menggunakan huruf a hingga z.
3. *Tokenizing* merupakan proses pemisahan kalimat ke bentuk yang lebih kecil berupa kata.
4. *Spelling Normalization* untuk memperbaiki dan mengganti kata-kata yang salah eja ataupun kata singkatan kedalam bentuk tertentu.
5. *Filtering* Filtering merupakan pengambilan kata-kata penting dari hasil *tokenizing*. Proses *filtering* dapat dilakukan dengan menghapus kata yang kurang dan tidak memiliki arti seperti kata hubung (*stopword*) atau menyimpan kata penting (*wordlist*).
6. *Stemming* untuk mengidentifikasi penulisan terhadap kata berimbuhan dan mengembalikan ke bentuk dasar dengan dilakukannya penggantian sesuai isi dari kamus besar bahasa indonesia (KBBI).

b. *Feature Selection*

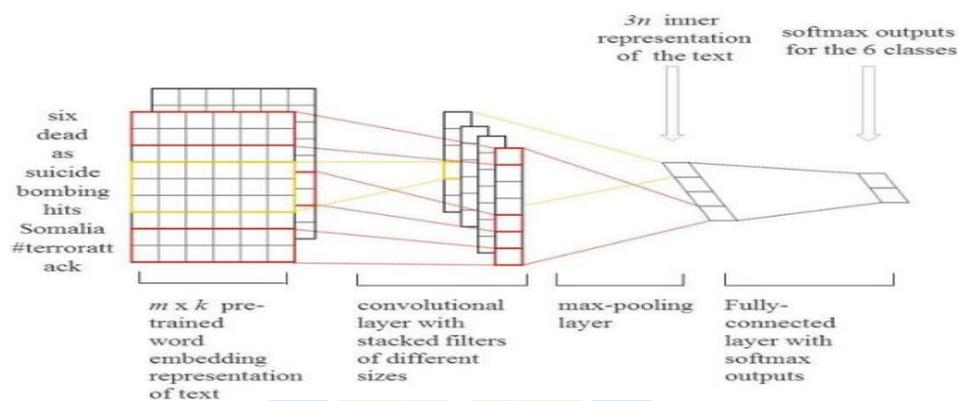
*Feature selection* merupakan suatu proses menghapus *features* yang berlebihan dan tidak relevan dari dataset yang sebenarnya . Dengan adanya *feature selection* membuat dataset menjadi sederhana sehingga efektif ketika dilakukan proses klasifikasi [16].

#### 2.2.4 Klasifikasi

Klasifikasi adalah proses kategorisasi terstruktur dari berbagai objek, gagasan, buku, atau benda lain ke dalam kelompok tertentu berdasarkan label yang sama. **Tujuan dari klasifikasi** adalah untuk mengelompokkan dan mengorganisasi suatu hal dengan sistem tertentu sehingga lebih mudah untuk dikenali, dipelajari, dan dipahami oleh pengguna [17].

### 2.2.5 Convolutional Neural Network (CNN)

*Convolutional Neural Network* (CNN) yaitu algoritma *deep learning* yang memiliki kemampuan untuk memahami berbagai jenis data, termasuk teks, suara, dan gambar. Pada awalnya algoritma CNN dikhususkan untuk menangani kasus citra, namun seiring berjalannya waktu algoritma CNN telah berkembang dan kini dapat digunakan untuk mengklasifikasi teks dengan memanfaatkan *FastText* pada tahap *Word Embedding*.



Gambar 2.2 Arsitektur CNN [18]

Gambar 2.2 menunjukkan arsitektur algoritma CNN yang digunakan untuk mengklasifikasi teks. Dalam algoritma ini, kalimat teks dipetakan ke dalam vektor *embedding* yang menghasilkan representasi vektor kata dari setiap token.

Proses ekstraksi dalam algoritma CNN terdiri dari beberapa layer diantaranya adalah *embedding layer*, *convolutional layer*, *pooling layer*, *dropout* dan *fully connected layer*.

a. *Embedding Layer*

*Word embedding* berfungsi untuk memetakan kata-kata dalam dokumen menjadi vektor padat (*dense vector*) yang merepresentasikan kata-kata tersebut ke dalam ruang vektor..

b. *Convolutional Layer*

*Convolutional Layer* adalah blok utama CNN tempat sebagian besar perhitungan dilakukan. *Convolutional Layer* ini menggunakan operasi konvolusi untuk mengekstrak fitur-fitur dari vektor *embedding* [19].

c. *Pooling Layer*

*Pooling layer* bertujuan untuk mereduksi dimensi dari fitur-fitur yang dihasilkan oleh *layer* sebelumnya, sehingga mengurangi kompleksitas dan mempercepat proses [20].

d. *Dropout*

*Dropout* digunakan untuk mencegah overfitting dengan secara acak mengabaikan beberapa neuron selama pelatihan model.

e. *Fully Connected Layer*

Lapisan *fully connected* ini menghubungkan hasil dari lapisan sebelumnya dan mengaplikasikan fungsi aktivasi *softmax* untuk mendapatkan probabilitas klasifikasi pada output layer [19].

### 2.2.6 Cosine Similarity

*Cosine Similarity* adalah algoritma di bidang *text mining* yang berfungsi untuk mengklasifikasi sebuah teks atau dokumen [21]. *Cosine Similarity* memiliki konsep normalisasi panjang vektor dengan membandingkan dua dokumen yaitu dokumen A dan dokumen B. Untuk menghitung *cosine similarity* dapat dilihat dari persamaan 2.1.

$$\text{similarity} = \cos(\emptyset) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2.1)$$

Keterangan :

A = Vektor A, yang berbanding kemiripannya

B = Vektor B, yang berbanding kemiripannya

$\|A\|$  = Panjang vektor A

$\|B\|$  = Panjang vektor B

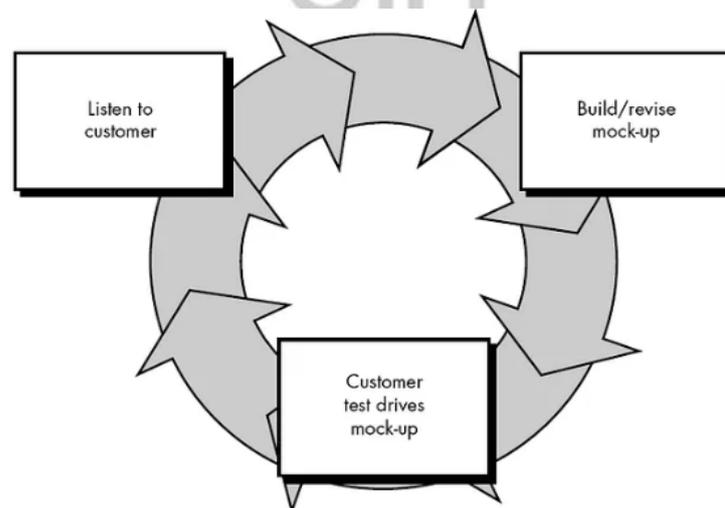
### 2.2.7 FastText

*FastText* adalah sebuah metode untuk mendapatkan representasi vektor dari kata-kata (*word embedding*). *FastText* merupakan pengembangan lebih lanjut dari *Word2Vec*. Kata-kata masukan akan direpresentasikan ke dalam vektor dan ditempatkan sedemikian rupa sehingga kata-kata yang mempunyai arti serupa akan muncul berdekatan, sedangkan kata-kata sebaliknya akan muncul jauh dari vektor.

Selain itu, *FastText* juga menyediakan model pretrained untuk 157 bahasa yang telah dilatih berdasarkan *common crawl* dan 11 *neighbour*. Perbedaan utama antara *FastText* dan *Word2Vec* adalah *FastText* dapat memproses kata-kata masukan yang tidak ada dalam kosakata atau kata-kata di luar kosakata [22]. Seperti *Word2Vec*, *FastText* memiliki dua arsitektur, yaitu *Continuous Bag of Words* (CBOW) dan *Skip-Gram*. CBOW memprediksi kata saat ini (sebagai target) dari konteks (sebagai masukan) di sekitarnya. Sedangkan *Skip-Gram* menggunakan kata saat ini (sebagai masukan) untuk memprediksi konteksnya (sebagai target).

### 1.2.8 Prototype

Metode *Prototype* adalah salah satu metode dalam pengembangan perangkat lunak yang melibatkan interaksi antara pengembang dan client. Biasanya *client* hanya menyediakan beberapa kebutuhan perangkat lunak umum tanpa memberikan inputan terperinci, proses yang harus dilakukan atau output yang sesuai dengan kebutuhan. Pada saat membangun sebuah sistem, pengembang belum tentu yakin dengan efisiensi dan algoritma, tingkat penyesuaian sistem operasi dan rancangan *interface client* yang akan dibuat. Situasi seperti itu biasanya sering terjadi, sehingga metode *prototype* ini sangat cocok untuk proses pengembangan perangkat lunak [23].



Gambar 2.3 Metode *Prototype* [23]

Metode *prototype* dimulai dengan mengumpulkan beberapa kebutuhan. Pengembang bertemu *client* untuk menentukan tujuan serta menentukan kebutuhan

perangkat lunak yang akan dibuat. Kemudian desain diselesaikan dengan cepat dan berfokus untuk merepresentasikan kebutuhan perangkat lunak yang diperlihatkan kepada *client*, seperti input dan output. Selain itu, *prototype* dievaluasi untuk melengkapi perangkat lunak yang akan dikembangkan.

