

BAB II

TINJAUAN PUSTAKA

1.1 Penelitian Terdahulu

Dalam penelitian ini, penulis melakukan *Study Literature* dengan menggali dari berbagai penelitian sebelumnya yang berkaitan dengan penelitian dari penulis sebagai bahan rujukan serta perbandingan baik mengenai kelebihan maupun kekurangan yang sudah ada.

1. Ananda Dwi R , Firdha Imamah , Yusuf Mei Andre S , Ardiansyah dalam penelitiannya tahun 2017 yang berjudul “*APLIKASI CHATBOT (MILKI BOT) YANG TERINTEGRASI DENGAN WEB CMS UNTUK CUSTOMER SERVICE PADA UKM MINSU*”.

Penelitian ini membahas peranan customer service sebagai sarana pelayanan pesanan pelanggan, salah satunya pada UKM MINSU yang memiliki bisnis berjualan susu. Aplikasi yang dibangun menggunakan sistem pakar dengan metode *Forward Chaining* sebagai pelanarannya, dan metode wawancara sebagai metode pengumpulan data. Hasil akhir dari penelitian ini merupakan penerapan Aplikasi *ChatBot* yang terintegrasi dengan *Web CMS* untuk *Customer Service* pada UKM MINSU.

2. Agus Suparno. Dalam penelitiannya tahun 2020 yang berjudul “*CHAT BOT SEBAGAI IMPLEMENTASI PEMANFAATAN TEKNOLOGI ARTIFICIAL INTELLIGENCE DENGAN CHANNEL TELEGRAM*”.

Penelitian ini membahas pemanfaatan Teknologi *Artificial Intelligence* dengan Channel Telegram, peneliti memanfaatkan Microsoft Azure sebagai bahan utama *ChatBot*, dimana dalam Microsoft Azure terdapat

layananan QNA Maker sehingga memudahkan dalam proses pembuatan *ChatBot*. Peneliti memilih Telegram sebagai media ChatBot dengan beberapa alasan, yakni salah satunya selain mudah dan gratis telegram merupakan aplikasi dengan tingkat keamanan yang cukup baik. Peneliti juga membuat *Knowledge Base* sebagai rangkaian pengetahuan dasar yang dijadikan bahan pembelajaran dari ChatBotnya. Hasil akhir dari penelitian ini adalah *chatBot* yang terintegrasi dengan *chanel* telegram yang dikembangkan menggunakan QNA Maker Milik Microsoft Azure sebagai bahan utama dari *chatBot*.

3. Guntoro. Loneli Costaner. Lisnawita. Dalam penelitiannya pada tahun 2020 yang berjudul “APLIKASI *CHATBOT* UNTUK LAYANAN INFORMASI DAN AKADEMIK KAMPUS BERBASIS *ARTIFICIAL INTELLIGENCE MARKUP LANGUAGE (AIML)*”

Penelitian ini membahas mengenai pengimplementasian Aplikasi ChatBot untuk layanan informasi dan akademik kampus. Adapun metode yang digunakan untuk pembelajaran chatBot menggunakan *Artificial Markup Language (AIML)*. Sedangkan *Knowledge base* atau pengetahuan dari aplikasi yang dibuat yaitu seputar Kampus, yang terdiri dari alamat kampus, syarat pendaftaran, Langkah pendaftaran, program studi, jalur kuliah, dan lain sebagainya. Tahapan dalam penelitian, peneliti mengembangkan aplikasi ChatBot ini dengan pengumpulan kebutuhan, desain, membuat prototype, evaluasi dan perbaikan. Berdasarkan hasil uji dari metode *White Box* dan *Black Box* , ChatBot mampu berjalan baik dengan skor 100%, sedangkan pengujian menggunakan UAT skor yang

didapat sebesar 95 %. Hal tersebut menunjukkan bahwa *chatbot* mampu menjawab pertanyaan-pertanyaan yang diajukan. Hasil akhir dari penelitian ini adalah Aplikasi *Chatbot* yang dikembangkan dengan AIML (*Artificial Markup Language*) untuk layanan informasi dan akademik kampus.

4. Tjut Awaliyah Zuraiyah, Dian Kartika Utami, Degi Herlambang dalam penelitiannya pada tahun 2019 yang berjudul “IMPLEMENTASI CHATBOT PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN RECURRENT NEURAL NETWORK” Penelitian ini membahas mengenai pembuatan aplikasi *Chatbot* otomatis yang dapat berkomunikasi dengan pengguna mengenai informasi pendaftaran mahasiswa baru di Universitas Pakuan menggunakan metode *Recurrent Neural Network (RNN)* untuk pengolahan *teks processing*. Aplikasi *chatbot* diimplementasikan menggunakan bahasa pemrograman *Python* dan API Telegram. Langkah-langkah implementasi *chatbot* yang dilakukan oleh peneliti meliputi pra-pemrosesan teks, konversi data ke format JSON, pelatihan data, pembungkusan kata, dan koneksi penuh. Pada aplikasi *chatbot* penerimaan mahasiswa baru Universitas Pakuan ini memiliki halaman chat dengan memanfaatkan platform Telegram. Untuk mengembangkan aplikasi ini peneliti menggunakan bahasa pemrograman *Python* dan metode *Recurrent Neural Network (RNN)* sebagai pengolah kata. Kemudian peneliti melakukan pemeriksaan data menggunakan *tensorflow* atau alat pembelajaran mendalam untuk mencapai tingkat akurasi yang tinggi dalam menjawab pertanyaan yang diketik atau dikirimkan oleh pengguna.

Tabel 2.1 Penelitian Terdahulu

Nama peneliti, Tahun dan Judul	Media	Metode	Hasil
Ananda Dwi R , Firdha Imamah , Yusuf Mei Andre S , Ardiansyah, 2017, “Aplikasi chatBot (<i>milki Bot</i>) yang terintegrasi dengan web cms untuk customer service pada ukm minsu”	<i>Web CMS</i>	<i>Forward Chaining</i>	Aplikasi <i>ChatBot</i> yang terintegrasi dengan Web CMS untuk Customer Service pada UKM MINSU
Agus Suparno, 2020, “Chat Bot sebagai implementasi Pemanfaatan Teknologi Artificial Intelligence dengan Channel Telegram”	<i>Telegram</i>	<i>QNA Maker (Microsoft Azure)</i>	<i>chatBot</i> yang terintegrasi dengan chanel telegram yang dikembangkan menggunakan <i>QNA Maker</i> Milik Microsoft Azure
Guntoro , Loneli Costaner , Lisnawita, 2020, “Aplikasi ChatBot untuk Layanan Informasi dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (AIML)”	<i>Web Siakad</i>	<i>AIML (Artificial Intelligence Markup Language)</i>	Aplikasi ChatBot yang dikembangkan dengan AIML (<i>Artificial Markup Language</i>) yang dapat memberikan informasi mengenai informasi pendaftaran mahasiswa di Universitas Lancang Kuning
Zuraiyah Tjut Awaliyah, Dian Kartika Utami, Degi Herlambang, 2019, “Implementasi chatBot pada pendaftaran mahasiswa baru menggunakan recurrent neural network”	Telegram	<i>Recurrent Neural Network (RNN)</i>	Aplikasi ChatBot sebagai media informasi pendaftaran siswa baru di Universitas Pakuan menggunakan <i>Recurrent Neural Network (RNN)</i>

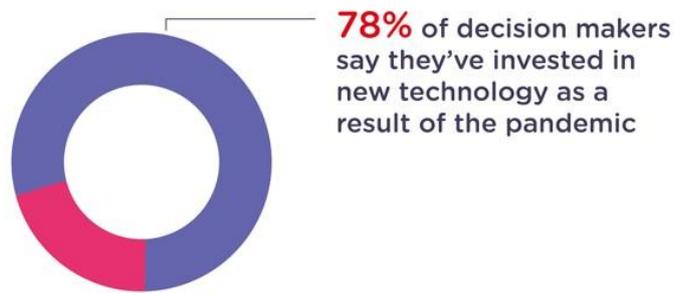
Nama peneliti, Tahun dan Judul	Media	Metode	Hasil
Ferry Agus Pratama, 2022, "Implementasi chatBot sebagai virtual assistent di universitas panca marga menggunakan metode tf-idf"	<i>Whatsapp</i>	TF-IDF VSM (<i>Term Frequency dan Invers Document Frequency Vector Space Model</i>) dan <i>Cosine Similarity</i>	Aplikasi <i>ChatBot</i> sebagai <i>Virtual Assistant</i> di Universitas Panca Marga Probolinggo dengan memanfaatkan TF-IDF VSM pembototan kata dan <i>cosine similarity</i> sebagai penghitung kemiripan dokumen

1.2 Landasan Teori

1.2.1 *ChatBot*

Chatbot adalah layanan obrolan berbasis *AI* (kecerdasan buatan), atau robot virtual yang dapat mensimulasikan percakapan manusia. *ChatBot* secara harfiah berasal dari dua kata, Chat dan Bot. Dalam dunia komputer, chatting dapat diartikan sebagai kegiatan komunikasi tertulis. Meskipun roBot adalah sebuah program, ia memiliki banyak data, dan ketika diberikan input, data tersebut akan menghasilkan output sebagai jawaban. Pengunjung situs. *Chatbot* mengoptimalkan komunikasi seolah-olah mereka tidak tahu, dan tetap menganggap bahwa pihak lain adalah karyawan yang tepat untuk tugas dan fungsi utama mereka Herwin dan Khusaeri Anddesa (2019:54).

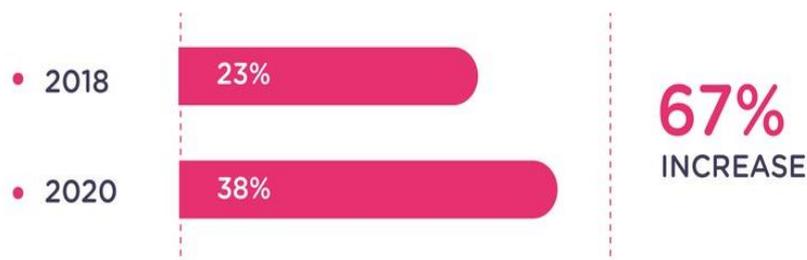
Pandemi COVID-19 yang dimulai pada tahun 2020 telah menyebabkan perubahan pada kebiasaan hidup masyarakat di seluruh dunia, termasuk Indonesia, khususnya di bidang industri, sektor, dan jasa bisnis. Hal ini mengakibatkan semua pekerjaan dilakukan secara online dengan istilah "*work from home*" (WFH).



Gambar 2.1 Statistika Investasi dalam teknologi baru
 Sumber : landAsdfg.io

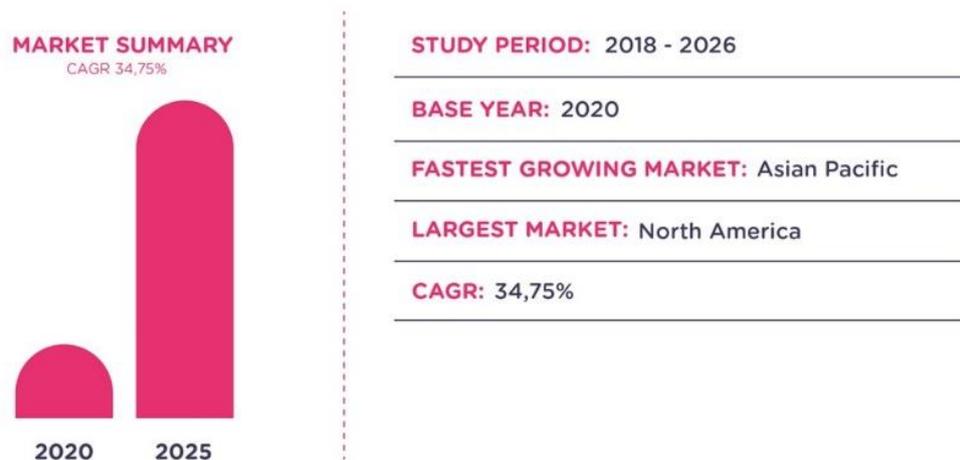
Berdasarkan Statistika pada gambar 2.1, teknologi terbukti menjadi satu satunya sektor dengan ruang peningkatan terbesar untuk saat ini, oleh karena perusahaan dari semua lini di paksa beradaptasi dengan *new normal*, 78% pengambil keputusan kini berinvestasi dalam teknologi baru sebagai akibat langsung dari pandemi COVID-19. Guna meningkatkan mutu pelayanan secara online, banyak perusahaan yang mengadopsi teknologi yang lebih canggih terutama pada bagian *customer service*. Layanan mandiri ini biasa disebut dengan *chatbot*.

DECISIONS MAKERS WHO SAY THEIR ORGANIZATION USE CHATBOTS



Gambar 2.2 Statistika penggunaan *Chatbot*
 Sumber : landAsdfg.io

Data mengatakan bahwa penggunaan *chatbot* dari tahun ke tahun semakin meningkat, selama beberapa tahun terakhir, banyak perusahaan telah memanfaatkan teknologi ini. Statistika penggunaan *chatbot* menunjukkan 67% antara 2018 dan 2020.



Gambar 2.3 Statistika Pasar *Chatbot*
Sumber : *landAsdfg.io*

chatbot yang berbasis AI (*Artificial Intelligence*) semakin meningkat dari sebelum pandemi hingga saat ini, bahkan sebelum pandemi *chatbot* sudah banyak diadopsi dan dikembangkan. Meningkatkannya penggunaan *Chatbot* disebabkan oleh tingkat efisiensi dan efektifitas yang dihasilkan *chatbot* terhadap penanganan kepada pengguna. Hal ini dikarenakan *chatbot* sangat membantu pekerjaan manusia, bahkan *chatbot* sendiri mampu menangani percakapan dengan manusia (berkomunikasi) tanpa ada campur tangan operator (orang yang mengoperasikan *chatbot*). Pada tahun 2020, *compound annual growth rate (CAGR)* mencatat 34.75% perkiraan dari tahun 2021-2026, pasar *chatbot* mengalami lonjakan yang sangat pesat dan hal ini diproyeksikan hingga tahun 2026.



Gambar 2.4 Statistika perkembangan *Chatbot* berdasarkan wilayah
Sumber : *landAsdfg.io*

Statistika diatas adalah penggunaan *Chatbot* berdasarkan Wilayah dari 2019 – 2024, Asia-Pasifik menjadi pengguna *Chatbot* paling tinggi. Banyak sekali *start-up* yang berinvestasi dalam teknologi chatBot dalam proses pemasaran, penjualan dan pelayanan.

Menurut Guzman dan Inez (dalam Ahmad Iswandi, 2018:7), *Chatbot* terdiri dari 3 (tiga) kombinasi, dimana ketiga kombinasi ini membentuk sebuah sistem yang kompleks, diantaranya :

1. *User Interface* (Antarmuka pengguna)

Antarmuka pengguna merupakan jembatan yang menghubungkan *chatbot* dengan pengguna (*user*) untuk saling berinteraksi melalui aplikasi. Antarmuka pengguna dirancang untuk membuat pengguna merasa nyaman dan lebih mudah menggunakan *chatbot*. *User Interface* bertujuan agar aplikasi yang diciptakan memiliki sifat user friendly atau mudah untuk digunakan bahkan oleh orang awam terhadap teknologi.

2. *Artificial Intelligence* (Kecerdasan buatan)

AI atau kecerdasan buatan mengacu pada simulasi kecerdasan manusia dalam mesin yang dirancang berpikir seperti manusia agar menghasilkan perilaku yang lebih natural. Hal ini membuat *chatbot* lebih pintar dalam menangkap setiap interaksi pengguna.

3. Integrasi

Integrasi adalah penggabungan komponen-komponen yang lebih kecil menjadi satu. Integrasi dengan sistem lain dapat meningkatkan fitur yang kaya di *chatbot*.

Pada perkembangan dunia bisnis atau teknologi saat ini, *Chatbot* menjadi salah satu solusi alternatif yang digunakan untuk memudahkan dalam melayani dengan pelanggan.

1.2.2 *Kecerdasan Buatan (Artificial Intelligence)*

Secara sederhana AI atau biasa disebut dengan kecerdasan buatan merupakan pemvisualisasian kepandaian manusia kedalam suatu sistem, dengan hal ini sistem dapat menirukan kecerdasan manusia sehingga sistem dapat berperilaku hampir sama dengan manusia. Dari perspektif pemrograman, AI juga meliputi studi tentang pemrograman simbolik, pemecahan masalah, dan proses pencarian (Desiani, Anita dan Muhammad Arhami, 2006).

AI (*Artificial Intelligence*) adalah bidang studi yang berusaha menerangkan dan meniru perilaku cerdas dalam bentuk proses komputasi. Dari perspektif Kecerdasan (*Intelligence*), AI adalah bagaimana membuat mesin yang "cerdas" dan dapat melakukan hal-hal yang sebelumnya hanya dapat dilakukan oleh manusia (Desiani, Anita dan Muhammad Arhami, 2006).

Artificial Intelligence atau kecerdasan buatan menjadi penggerak revolusi industri 4.0 yang menjanjikan banyak kemudahan bagi sektor pemerintah maupun *Internet of Things* (IoT) dan *big data* contohnya dimana AI dapat diimplementasikan. Teknologi yang telah banyak diadopsi di era industri 4.0 ini mampu menghubungkan setiap perangkat, seseorang dapat mengotomatisasi semua perangkat tanpa harus berada di lokasi.

1.2.3 Asisten Virtual (*Virtual Assistant*)

Menurut Kamus Besar Bahasa Indonesia (KBBI) kata virtual memiliki arti “nyata”. Sedangkan Asisten memiliki arti “orang yang bertugas membantu orang lain dalam melaksanakan tugas profesional”. Namun secara definisi asisten virtual atau *Virtual Assistant* adalah orang yang memberikan layanan administratif kepada klien dari jarak jauh (*remote*).

Remote sendiri memiliki arti pengendali jarak jauh. *Remote working* atau kerja *remote* dapat dikatakan bekerja secara jauh tidak dilakukan didalam kantor.

Hakikatnya, asisten virtual sama dengan asisten pada umumnya. Hanya saja dilakukan secara virtual atau online. Asisten virtual tidak membutuhkan ruangan khusus layaknya asisten pada umumnya. Sehingga hal ini dapat meminimalisir biaya tambahan bagi perusahaan.

Secara garis besar asisten virtual dapat dikatakan sebagai *customer service* yang bekerja secara online untuk melayani klien dari jarak jauh.

1.2.4 Text Preprocessing

Text Preprocessing adalah teknik awal dalam data mining untuk mengubah data mentah atau *raw data* yang dikumpulkan dari berbagai sumber menjadi

informasi yang lebih bersih dan bisa digunakan untuk pemrosesan data lebih lanjut. Pada tahap *preprocessing* dilakukan penyiapan dokumen mentah atau *raw data* menjadi dokumen yang siap diproses untuk Langkah selanjutnya (Nugraha, 2020). Pada tahap ini proses untuk mengolah inputan dari pengguna dilakukan antara lain :

1. *Case Folding*

Case Folding adalah Teknik mengubah semua huruf menjadi huruf kecil.

Case Folding hanya mengatasi huruf ‘a’ sampai dengan ‘z’, karakter Selain huruf akan dihilngkan atau tidak dianggap.

Berikut contoh dari *case folding* :

Tabel 2.2 Hasil *Case Folding*

Sebelum Case Folding	Sesudah Case Folding
Berapa biaya Pendaftaran di UNIVERSITAS PANCA MARGA ?	berapa biaya pendaftaran di universitas panca marga ?

2. *Tokenizing*

Tokenizing adalah Proses memisahkan kalimat menjadi susunan kata.

Umumnya setiap kata dipisahkan oleh spasi, sehingga proses tokenisasi bergantung pada spasi untuk memisahkan kata.

Berikut contoh dari *tokenizing* :

Tabel 2.3 Hasil *Tokenizing*

Sebelum Tokenizing	Sesudah Tokenizing
berapa biaya pendaftaran di universitas panca marga ?	berapa biaya pendaftaran di universitas panca marga

3. Stopword Removal

Stopword Removal adalah Teknik atau proses menghapus kata yang tidak perlu atau kata yang termasuk kedalam *stopword*, *stopword* adalah kata-kata yang biasanya dan sering muncul dan akan diabaikan dalam pemrosesan.

Berikut beberapa contoh *Stopword* :

Tabel 2.4 kata *Stopword*

Daftar beberapa <i>Stopword</i>
'ada', 'adalah', 'agak', 'agar', 'akan', 'amat', 'anda', 'antara', 'anu', 'apakah', 'apalagi', 'atau', 'bagaimanapun', 'bagi', 'bahwa', 'begitu', 'belum', 'bisa', 'boleh', 'dahulu', 'dalam', 'dan', 'dapat', 'dari', 'daripada', 'demi', 'demikian', 'dengan', 'di', 'dia', 'dimana', 'dll',

4. Stemming

Stemming adalah Teknik atau proses mengubah setiap kata menjadi kata dasarnya. Pada proses ini, setiap kata yang sudah melalui tahap *tokenizing* akan dihilangkan kata imbuhan, seperti “-an”, “-nya”, “pe-”, “me-” dan lain sebagainya. *Stemming* bertujuan untuk memudahkan proses pada tahap selanjutnya, karena kata yang akan diolah hanya kata penting atau kata yang tidak memiliki kata imbuhan.

Berikut contoh dari *Stemming* :

Tabel 2.5 Hasil *Stemming*

Sebelum <i>Stemming</i>	Sesudah <i>Stemming</i>
berapa	berapa
biaya	biaya
pendaftaran	daftar
di	universitas
universitas	panca
panca	marga
marga	

1.2.5 Use Case Diagram

Use case diagram adalah satu dari berbagai jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case Diagram* atau sering disebut sebagai diagram perilaku, digunakan untuk menggambarkan secara ringkas keterhubungan antar aktor dan sistem. *Use Case Diagram* juga digunakan untuk memperjelas bagaimana suatu sistem bekerja, didalam *use case diagram* itu sendiri memiliki hubungan dengan entitas yang berada diluar sistem yang sering disebut dengan aktor.

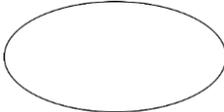
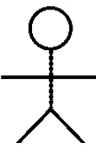
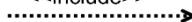
use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat, *use case* diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan yang berhak menggunakan fungsi-fungsi tersebut.

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Diagram ini penting untuk mengorganisasikan dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna. Secara sederhana, *use case diagram* dapat dikatakan sebagai pemodelan interaksi antara aktor dengan sistem yang digambarkan melalui simbol-simbol diagram. Hal ini bertujuan untuk memperjelas antara hubungan pengguna (aktor) dengan sistem yang akan dibangun.

Sebuah *use case* diagram menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem (Kurniawan, Tri A, 2018). Syarat Penamaan *use case* sesederhana dan semudah mungkin untuk dipahami.

Berikut simbol-simbol yang ada pada *use case* Diagram diantaranya adalah :

Tabel 2.6 Simbol Use Case Diagram

Simbol	Penjelasan
Use Case 	Use case menggambarkan fungsi, kebutuhan dari perspektif pengguna. Biasanya dinyatakan dalam kata kerja awal di awal nama <i>use case</i> .
Aktor 	Aktor merupakan pengguna, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat.
Asosiasi 	Merelasikan hubungan antara aktor dengan use case.
Generalisasi 	Hubungan antara generalisasi dan spesialisasi (Umumnya spesifik) Antara dua kasus penggunaan di mana satu fungsi lebih umum daripada yang lain.
Include <<include>> 	Perilaku yang harus dipenuhi agar peristiwa itu terjadi. Dalam kondisi ini, satu use case adalah bagian dari use case lainnya.
Extend <<extend>> 	Perilaku yang hanya berjalan di bawah kondisi tertentu

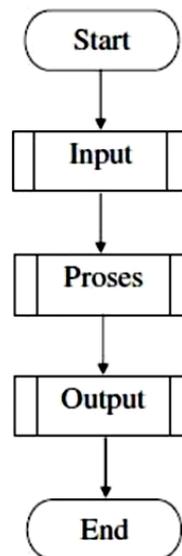
1.2.6 Flowchart Diagram

Flowchart adalah penggambaran secara grafik dari tahapan penyelesaian suatu masalah yang terdiri atas sekumpulan simbol, dimana masing-masing simbol merepresentasikan suatu kegiatan tertentu (dosenpendidikan, 2022). Bagan alir atau *Flowchart* merupakan kumpulan dari notasi diagram simbolik yang menunjukkan aliran data dan urutan operasi dalam sistem.

Sebenarnya tidak ada formula atau standar yang mutlak (menentukan) dalam hal desain diagram alur. Hal ini didasarkan pada diagram alir dan merupakan diagram hasil berpikir ketika menganalisis suatu masalah komputer.

Secara Umum *Flowchart* berfungsi untuk menggambarkan secara jelas suatu alur pekerjaan dan kerangka acuan dalam pengambilan keputusan. Hal ini dinilai dapat meringkas suatu alur menjadi lebih mudah dipahami dan dianalisa berdasarkan urutan serta langkah dari suatu proses.

Flowchart adalah bagan dengan symbol-simbol tertentu yang menjelaskan alur dari suatu sistem. *Flowchart* biasanya terdiri dari 5 bagian, yaitu start, input, proses, output, end.



Gambar 2.5 Tahap umum *Flowchart*

1. *Start* : digunakan sebagai simbol awal permulaan dari suatu sistem, biasanya disimbolkan dengan terminal.
2. *Input* : digunakan sebagai tempat menampung dari berbagai variabel masukan yang ada dalam sistem.
3. *Process* : digunakan sebagai tempat pemrosesan dari inputan.

4. *Output* : digunakan sebagai keluaran dari semua masukan yang sudah diproses oleh sistem.
5. *End* : digunakan simbol selesai atau akhir dari suatu sistem.

Flowchart dapat digunakan untuk menyajikan kegiatan manual, kegiatan pemrosesan ataupun keduanya menggunakan simbol, notasi atau simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok. Berikut adalah simbol-simbol dari flowchart :

1. *Flow Direction Symbols* (Simbol Penghubung/alur)

Flow Direction Symbols atau Simbol Penghubung digunakan untuk menghubungkan antara simbol yang satu dengan yang lainnya.

Tabel 2.7 *Flow Direction Symbols Flowchart*

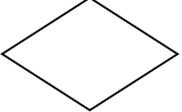
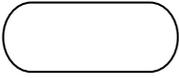
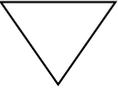
Simbol	Penjelasan
<p style="text-align: center;"><i>Arus / Flow</i></p> 	Menyatakan jalannya suatu proses
<p style="text-align: center;"><i>Communication Link</i></p> 	Menyatakan bahwa adanya transisi suatu data atau informasi dari suatu ke lokasi lainnya.
<p style="text-align: center;"><i>Connector</i></p> 	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman yang sama
<p style="text-align: center;"><i>Offline Connector</i></p> 	Menyatakan sambungan dari suatu proses ke proses lainnya dalam halaman yang berbeda

2. *Processing Symbols* (Simbol Proses)

Simbol yang menunjukkan proses operasi dalam suatu sistem.

Tabel 2.8 *Processing Symbols Flowchart*

Simbol	Penjelasan
<p style="text-align: center;">Proses</p> 	Menyatakan suatu pemrosesan yang dilakukan oleh komputer atau system

Simbol	Penjelasan
Simbol Manual 	Menyatakan suatu pemrosesan yang tidak dilakukan oleh komputer (manual)
<i>Decision / Logika</i> 	Menyatakan suatu kondisi dengan dua kemungkinan, <i>true / false</i>
<i>Predefined Process</i> 	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
<i>Terminal</i> 	Menyatakan permulaan atau akhir suatu program atau sistem
<i>Offline Storage</i> 	Menunjukkan bahwa data akan disimpan ke suatu media tertentu

3. *Input / Output Symbols* (Simbol Input – output)

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media input dan output.

Tabel 2.9 Input / Output Symbols Flowchart

Simbol	Penjelasan
Input / Output 	Menyatakan proses <i>input</i> dan <i>output</i>
Disk Storage 	Menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>
Document 	Mencetak dokumen

1.2.7 *TF-IDF (Term Frequency – Inverse Document Frequency)*

TF-IDF Term-Frequency Inverse Document Frequency adalah perhitungan atau pembobotan kata melalui teknik *tokenizing* (tokenisasi), *stopwords*, dan *steming*, dan frekuensi munculnya kata dalam dokumen yang diberikan menunjukkan pentingnya kata itu di dalam sebuah dokumen (Rofiqi, 2019). *TF-IDF* melakukan pembobotan kata (*term*) pada setiap dokumen dengan memperhatikan frekuensi atau seringnya kata tersebut muncul dalam sebuah dokumen. Hasil akhir dari metode pembobotan kata ini adalah bobot kata pada setiap dokumen, dimana kata yang paling sering muncul akan memiliki bobot paling tinggi dalam sebuah dokumen.

1. (*TF*) *Term Frequency*

TF mewakili berapa kali sebuah kata muncul dalam dokumen.

Pentingnya *term*_i dalam sebuah dokumen *j* dapat dinyatakan sebagai:

$$tf = tf_{ij} \quad (1)$$

Keterangan :

tf : *Term frequency*

*tf*_{ij} : Banyaknya muncul *term* (kata)_i pada dokumen *j*

Term Frequency (tf) menghitung banyaknya kemunculan *term* dalam dokumen . Semakin sering suatu istilah muncul, semakin besar nilai *tf*-nya.

Berikut beberapa cara untuk mendefinisikan *term frequency*.

skema	bobot tf
biner	0, 1
pencacahan mentah	$f_{t,d}$
frekuensi istilah	$\frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$
penormalan log	$\log(1 + f_{t,d})$
penormalan ganda 0,5	$0,5 + 0,5 \times \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
penormalan ganda K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Gambar 2.6 *Term frequency*

Contoh :

Dalam contoh ini, penulis asumsikan semua dokumen sudah melewati tahapan *preprocessing*.

d1 = "sks sitem kredit semester"

d2 = "maksimal sks 1 semester 24 sks"

d3 = "1 sks 45 menit"

disini penulis menggunakan cara ke 4 dalam tabel dalam menentukan *term frequency*.

Tabel 2.10 Term Frequency

term				tf = log (1 + tf_{ij})		
	d1	d2	d3	d1	d2	d3
sks	1	2	1	0.301	0.362	0.301
sistem	1	0	0	0.301	0	0
kredit	1	0	0	0.301	0	0
semester	1	1	0	0.301	0.301	0
maksimal	0	1	0	0	0.301	0
1	0	1	1	0	0.301	0
24	0	1	0	0	0.301	0
45	0	0	1	0	0	0.301
menit	0	0	1	0	0	0.301

2. *IDF (Inverse Document Frequency)*

IDF (Inverse Document Frequency) merupakan perhitungan dari jumlah seluruh dokumen (D) dibagi dengan Frekuensi Dokumen (D_{fi}) dari *term* (kata) i . Jika suatu istilah sering muncul, itu tidak dianggap sebagai istilah dokumen. Sebaliknya, jika istilah tersebut jarang muncul dalam korpus, maka istilah tersebut dikatakan terkait dengan dokumen.

IDF (Inverse Document Frequency) dapat dinyatakan sebagai berikut:

$$idf_i = \log\left(\frac{N}{D_{fi}}\right) \quad (2)$$

Keterangan :

idf_i = *inverse document frequency*

N = Jumlah dokumen

D_{fi} = Jumlah kemunculan *term* i terhadap Dokumen

Nilai D_{fi} dapat dicari dengan menghitung seberapa sering sebuah kata (*term*) muncul dalam sebuah dokumen, apabila sebuah kata muncul pada semua dokumen, maka nilai D_{fi} terhadap kata tersebut adalah sebanyak dokumen yang ada.

Semakin jarang (tidak sering muncul) suatu istilah di antara dokumen, semakin besar nilai *idf*-nya. Nilai *idf* adalah hasil dari logaritma dari kebalikan dari jumlah dokumen yang memiliki istilah kata (*term*) yang dibagi jumlah seluruh dokumen (N), seperti yang digambarkan pada rumus diatas.

Berikut beberapa cara menentukan idf :

skema	bobot tf
biner	0, 1
pencacahan mentah	$f_{t,d}$
frekuensi istilah	$\frac{f_{t,d}}{\sum_{t \in d} f_{t,d}}$
penormalan log	$\log(1 + f_{t,d})$
penormalan ganda 0,5	$0,5 + 0,5 \times \frac{f_{t,d}}{\max_{\{t \in d\}} f_{t,d}}$
penormalan ganda K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t \in d\}} f_{t,d}}$

Gambar 2.8 Inverse document frequency

Contoh :

d1 = "sks sitem kredit semester"

d2 = "maksimal sks 1 semester 24 sks"

d3 = "1 sks 45 menit"

dari tabel 2.9 dapat ditemukan tf sebagai berikut, maka perhitungan dari idf menggunakan rumus diatas adalah :

Tabel 2.11 perhitungan inverse document frequency

term	tf			idf	
	d1	d2	d3	D_{fi}	$\log(N / D_{fi})$
sks	0.301	0.362	0.301	3	0
sistem	0.301	0	0	1	0.477
kredit	0.301	0	0	1	0.477
semester	0.301	0.301	0	2	0.176
maksimal	0	0.301	0	1	0.477
1	0	0.301	0	1	0.176
24	0	0.301	0	1	0.477
45	0	0	0.301	1	0.477
menit	0	0	0.301	1	0.477

3. Term Weighting TF-IDF

Term Weighting TF-IDF adalah penggabungan dari formula perhitungan *tf* dengan formula *idf* dengan cara mengalikan nilai *tf* dengan nilai *idf* pada masing masing dokumen.

Term Weighting dapat dinyatakan sebagai berikut:

$$w_{ij} = tf_{ij} \times \log\left(\frac{N}{D_{fi}}\right) \quad (3)$$

atau

$$w_{ij} = tf \times idf \quad (4)$$

Keterangan :

W_{ij} = Bobot *term* terhadap dokumen

tf = *term frequency*

idf = *Inverse document frequency*

berikut beberapa cara dalam menentukan *term wheighting* atau pembobotan kata :

skema	bobot istilah dokumen (<i>d</i>)	bobot istilah kueri (<i>q</i>)
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0,5 + 0,5 \times \frac{f_{t,q}}{\max_t f_{t,q}}\right) \times \log \frac{N}{n_t}$
2	$\log(1 + f_{t,d})$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \times \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \times \log \frac{N}{n_t}$

Gambar 2.9 *Term wheighting*

Contoh :

d1 = "sks sitem kredit semester"

d2 = "maksimal sks 1 semester 24 sks"

d3 = "1 sks 45 menit"

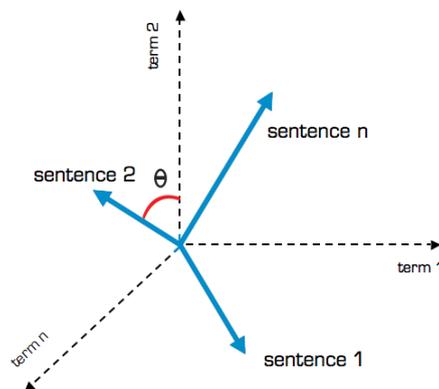
dari tabel 2.9 dapat ditemukan tf dan idf sebagai berikut, maka perhitungan dari *term weighing* menggunakan rumus diatas adalah

Tabel 2.12 perhitungan *term weighing*

term	tf			idf	d1	w _{ij}	
	d1	d2	d3			d2	d3
sks	0.301	0.362	0.301	0	0	0	0
sistem	0.301	0	0	0.477	1.143	0	0
kredit	0.301	0	0	0.477	1.143	0	0
semester	0.301	0.301	0	0.176	1.053	1.053	0
maksimal	0	0.301	0	0.477	0	1.053	0
1	0	0.301	0	0.176	0	1.053	0
45	0	0	0.301	0.477	0	0	0.143
menit	0	0	0.301	0.477	0	0	0.143

1.2.8 VSM (*Vector Space Model*)

Vector Space Model adalah sebuah pendekatan natural yang berbasis pada *vector* dari setiap kata dalam suatu dimensi spasial, Sebuah dokumen dilihat sebagai *vector* yang memiliki jarak dan arah. Relevansi sebuah *query* ke sebuah dokumen didasarkan pada similaritas diantara *vector* dokumen dan *vector query*.



Gambar 2.10 *vector space model*

Gambar diatas menggambarkan pemodelan dokumen teks di ruang dimensi dimana *sentence* adalah kalimat dokumen sedangkan term adalah term atau kata. Metode *Vector Space Model* menghitung nilai sudut *cosinus* dari dua vektor yaitu vektor dari *term* pada *query* dan vektor dari *term* pada dokumen (Priandono, 2020).

Jumlah N kata yang berbeda sebagai kosakata atau istilah *indeks* kata [*indeks*]. Kata-kata ini membentuk ruang vektor dengan dimensi N. Setiap i-word dalam dokumen atau query diberi bobot w_i . Baik dokumen maupun kueri direpresentasikan sebagai vektor tak berdimensi.

	T_1	T_2	T_3	T_4	...	T_N
D_1	W_{11}	W_{21}	W_{31}	W_{41}	...	W_{N1}
D_2	W_{12}	W_{22}	W_{32}	W_{42}	...	W_{N2}
D_3	W_{13}	W_{23}	W_{33}	W_{43}	...	W_{N3}

Gambar. 2.11 Vektor dimensi

Pada VSM, sebuah kata direpresentasikan dengan sebuah dimensi dari ruang vector. Hal ini dicapai dengan menetapkan bobot non-biner untuk istilah dalam kueri dan dokumen. Bobot istilah ini digunakan untuk menghitung tingkat kesamaan antara setiap dokumen yang tersimpan dalam sistem dan query yang diterima dari permintaan pengguna, Lalu Dokumen yang terambil diranking dalam urutan yang memiliki kemiripan.

Model vektor memperhitungkan pertimbangan dokumen yang terkait dengan kueri pengguna. Hasilnya, kumpulan dokumen yang diambil jauh lebih akurat. Dokumen d_j dan query q direpresentasikan sebagai vektor t-dimensi.

1.2.9 Cosine Similarity

Cosine Similarity adalah metode yang digunakan untuk menghitung similarity (tingkat kesamaan) antar dua buah objek. Secara umum perhitungan metode ini didasarkan pada vector space similarity measure. Metode *cosine similarity* digunakan untuk mendeteksi dokumen-dokumen yang mirip, cosine similarity akan menghitung tingkat kesamaan antar dua buah atau lebih objek yang dinyatakan dalam vektor. Jumlahnya ada dua vektor dengan menggunakan kata kunci (*cosine*) (Dedy, 2020).

Berikut rumus perhitungan cosine similarity :

$$\cos = \frac{A \cdot B}{|A||B|} \quad (5)$$

keterangan :

A = Vektor A, yang akan dibandingkan kemiripannya

B = Vektor B, yang akan dibandingkan kemiripannya

|A| = Panjang vektor A

|B| = Panjang vektor B

Ini menggunakan model ruang vektor dan pembobotan TF-IDF untuk mewakili nilai numerik dalam dokumen, memungkinkan Anda untuk menghitung kedekatan antar dokumen. Kesamaan antar dokumen dihitung menggunakan fungsi ukuran kesamaan. Semakin tinggi hasil fungsi kesamaan, semakin tinggi kesamaan antara dua objek yang dievaluasi, dan sebaliknya, semakin kecil hasil fungsi kesamaan, semakin jauh jarak kedua objek yang dievaluasi. Pengukuran ini memungkinkan Anda untuk menentukan peringkat dokumen menurut kesamaan kueri (relevansi).

1.2.10 Akurasi Sistem Temu balik Informasi

Sistem Temu Balik Informasi atau biasa disebut dengan Information Retrieval adalah penemuan bahan seperti dokumen yang bersifat terstruktur untuk menemukan kembali informasi-informasi yang relevan terhadap dari suatu kumpulan informasi.

Inti dari system temu balik informasi adalah mencari dokumen dokumen yang relevan sesuai dengan *query* (permintaan) dari pengguna. Oleh karena itu, perlu dihitung *similarity* dari setiap dokumen terhadap *query*. Dari hasil perhitungan tersebut akan menghasilkan bobot dari setiap dokumen, dimana bobot ini akan dilakukan ranking untuk mengurutkan relevansi dokumen terhadap sebuah *query*.

Dalam menentukan relevansi, penulis membutuhkan pengujian untuk mengetahui seberapa besar akurasi (keberhasilan) dari sistem yang dibangun. Akurasi merupakan kedekatan nilai yang diperoleh dari hasil pengukuran dengan nilai yang sebenarnya. Semakin dekat nilai hasil dari pengukuran, maka sistem dapat dikatakan semakin akurat. Nilai akurasi adalah nilai yang menggambarkan seberapa akurat hasil klasifikasi dari system (Fauzal, 2020).

Perhitungan akurasi dapat menggunakan rumus sebagai berikut :

$$\text{Akurasi} = \frac{\text{Jumlah jawaban benar}}{\text{Jumlah total pertanyaan}} \times 100\% \quad (6)$$

Studi kasus :

Ada 100 pertanyaan yang diajukan dan hanya 80 pertanyaan yang dapat dijawab dengan benar, 20 jawaban lainnya tidak bias terjawab oleh sistem.

Maka perhitungan dari tingkat akurasi sebagai berikut :

$$\text{Akurasi} = \frac{80}{100} \times 100\%$$

$$\text{Akurasi} = 80\%$$

1.2.11 User Acceptance Test

User Acceptance test adalah pengujian yang dilakukan untuk menyesuaikan perangkat lunak yang dibangun sudah sesuai dengan kebutuhan yang ada, dimana proses tersebut akan menentukan perangkat lunak yang telah dibangun dapat diterima atau tidak (Nugraha, 2020).

Proses UAT atau (*user acceptance test*) berbeda dengan pengujian sistem (memastikan software tidak crash dan sesuai dengan dokumen permintaan pengguna), melainkan memastikan bahwa solusi dalam sistem tersebut akan bekerja untuk pengguna (yaitu, tes bahwa pengguna menerima solusi di dalam sistem) .

Tujuan dari UAT adalah untuk memberikan penjelasan tentang faktor-faktor penentu penerimaan computer yang bersifat umum, yang mampu menjelaskan perilaku pengguna diberbagai teknologi komputasi.

1.2.12 Whatsapp

Whatsapp adalah aplikasi gratis yang menyediakan layanan bertukar pesan dan panggilan yang sederhana, aman, dan reliabel serta tersedia pada berbagai telepon di seluruh dunia. Whatsapp menjadi salah satu aplikasi pesan instan

paling populer diseluruh dunia, bahkan di universitas panca marga probolinggo mayoritas civitas akademik , staff , dosen dan juga mahasiswa menggunakan media whatsapp sebagai layanan untuk bertukar pesan.

Ada beberapa alasan yang menjadi landasan penulis untuk memilih aplikasi whatsapp sebagai media dari chatbot, diantaranya adalah :

1. *User friendly*

Whatsapp memiliki tampilan yang user friendly atau mudah digunakan, hal ini dapat dilihat dari tampilan whatsapp yang terlihat sederhana, rapi serta pemilihan warna yang tidak terlalu mencolok.

2. Banyak pengguna

Whatsapp memiliki banyak pengguna diseluruh dunia, mengutip dari situs resmi whatsapp, “Lebih dari 2 miliar orang di lebih dari 180 negara menggunakan WhatsApp”. Salah satunya adalah negara Indonesia, whatsapp menjadi salah satu pesan instan dengan banyak pengguna, bahkan di universitas panca marga probolinggo, grup kelas, grup Angkatan, bahkan grup civitas akademik menggunakan media whatsapp. Hal ini yang menjadi landasan penulis untuk memilih whatsapp sebagai media dari chatbot.

3. *Free Access*

Selain mudah digunakan dan banyak pengguna, whatsapp adalah aplikasi pesan instan gratis. Untuk dapat menggunakan whatsapp, pengguna tidak perlu membayar apapun kepada pemilik whatsapp.

Untuk dapat menggunakan layanan whatsapp, pengguna hanya memerlukan nomor telepon serta sambungan internet, kedua hal tersebut menjadi syarat utama untuk dapat menggunakan aplikasi whatsapp.