

APLIKASI CHATBOT PADA SISTEM INFORMASI PENYEWAAN SCAFFOLDING MENGGUNAKAN METODE TF-IDF

Trianta Almira Ramadhani¹, Dimas Wahyu Wibowo, ST., MT., Habibie Ed Dien, S.Kom., MT.

Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang

¹almiraramadhani0301@gmail.com, ² [alamat](#) e-mail penulis 2, ³ [alamat](#) e-mail penulis 3

Abstrak

Perkembangan teknologi saat ini semakin memudahkan pengguna dalam mengakses aplikasi yang ada. Salah satunya dengan menerapkan Sistem Informasi Berbasis Website. Berdasarkan hasil penelitian sebelumnya yang menerapkan Chatbot dengan metode N-gram, TF-IDF, dan Cosine Similarity, terbukti Sistem Chatbot pada aplikasi tersebut dapat memudahkan masyarakat dalam mencari informasi

Oleh sebab itu, CV. Scaffolding Samarinda membutuhkan pemberian layanan sistem informasi yang cepat di setiap saat. Berdasarkan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa Sistem ini dapat melakukan *Tokenizing* (memisah kata penyusun dari suatu dokumen) dan menghitung bobot *TF-IDF* dan *Cosine Similarity* untuk mencari jawaban pada Sistem. Sehingga saat pelanggan menuliskan pertanyaan, akan muncul jawaban sesuai harapan *user*. Sistem juga dapat memudahkan pelanggan dalam menerima informasi sesuai dengan yang diharapkan saat itu juga.

Kata Kunci: Chatbot, TF-IDF, Cosine Similarity, Scaffolding

1. Pendahuluan

Perkembangan teknologi saat ini semakin memudahkan pengguna dalam mengakses aplikasi yang ada. Penggunaan aplikasi menggunakan desktop sudah dikalahkan dengan luasnya penggunaan internet yang bisa digunakan di berbagai platform dan bisa diakses oleh pengguna dari kalangan manapun, dan menjadikan website dapat memberikan kemudahan bagi masyarakat dalam mengaksesnya. Salah satunya dengan menerapkan Sistem Informasi Berbasis Website.

Berdasarkan hasil penelitian sebelumnya terkait penelitian yang dilakukan oleh (Kavitha, Cethana, 2019) yang menerapkan *chatbot* untuk membantu masyarakat jika ingin mengetahui penyakit yang mungkin di derita dengan menanyakannya melalui aplikasi android dan akan diproses oleh sistem pada web dan akan menampilkan jawaban sesuai harapan dengan metode N-gram, TF-IDF, dan Cosine Similarity. Sistem *Chatbot* pada aplikasi tersebut menggantikan peran Dokter untuk membantu dokter untuk mengurangi biaya perawatan dan menghemat waktu. Sehingga pengguna aplikasi tidak perlu ke dokter atau ke spesialis untuk menanyakannya. Penelitian lain yang saya gunakan milik (Dhebys, Eka, 2017) yang menggunakan *chatbot* untuk memudahkan masyarakat dalam mencari objek wisata di daerah Jawa Timur. *Chatbot* pada aplikasi tersebut memudahkan objek wisata memberikan informasi kepada pengunjung tanpa harus menyebarkan pamflet, brosur, dan poster.

TF-IDF (*Term Frequency - Inverse Document Frequency*) merupakan metode algoritma yang menentukan frekuensi dari kemunculan sebuah term dalam dokumen yang bersangkutan. Metode ini untuk menghitung nilai *Term*

Frequency (TF) dan *Inverse Document Frequency* (IDF) pada setiap kata di setiap dokumen. Semakin besar jumlah kemunculan suatu term (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. (informatikalogi.com).

Bagaimana mengatasi *Customer* yang sangat mengharapkan bisa mendapatkan informasi yang cepat setiap saat dalam waktu 24 jam tersebut, maka pada penelitian ini, penulis membuat Aplikasi Chatbot pada Sistem Informasi Scaffolding dengan Menggunakan Metode TF-IDF yang diharapkan dapat memudahkan serta dapat mengatasi permasalahan yang telah dipaparkan di atas. Dengan aplikasi chatbot ini diharapkan dapat menjawab pertanyaan dari calon pelanggan dengan informasi/penjelasan yang mudah dipahami, dan terlaksana dalam memberikan informasi pada saat itu juga.

Tujuan untuk memudahkan pelanggan saat ingin bertanya mengenai penyewaan scaffolding dengan jawaban sesuai jam bertanya dan pertanyaan beruntun akan menghasilkan jawaban yang beruntun, yaitu tentang:

- Informasi mengenai barang yang tersedia
- Informasi mengenai harga sewa barang per unit
- Informasi mengenai jangka waktu peminjaman barang
- Informasi mengenai berat barang per unit

2. Tinjauan Pustaka

2.1. Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji

penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama dengan judul penelitian penulis. Namun, penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Tabel 2.1
Penelitian Terdahulu

<i>Nama Peneliti</i>	<i>Judul Penelitian</i>	<i>Hasil Penelitian</i>
<i>Suryani, Dhebys & Larasati Amalia, Eka, 2017.</i>	<i>Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML</i>	<i>Aplikasi Chatbot dapat memberikan informasi kepada wisatawan yang ingin berwisata di wilayah Jawa Timur</i>
<i>Satria Paliwabet, I Nyoman & Gede Darma Putra, I Ketut, 2017.</i>	<i>Pencarian Informasi Wisata Daerah Bali Menggunakan Teknologi Chatbot</i>	<i>Sistem chatbot dengan menggunakan metode Fulltext Search Boolean Mode dari MySQL dapat diterapkan dengan baik</i>
<i>Suryani, Dhebys & Aulia, Indinabilah, 2018</i>	<i>Penerapan Metode TF-IDF dan N-Gram pada Pengembangan Aplikasi Chatbot Berbasis LINE untuk Layanan Publik Kesehatan Kota Malang</i>	<i>Question-Answering dalam bentuk chatbot menggunakan N-Gram, TF-IDF dan Cosine Similarity dapat berkomunikasi dan menyampaikan informasi.</i>
<i>Astiningrum, Mungki & Shoburu Rohmah, Maya, 2018.</i>	<i>Implementasi NLP dengan Konversi Kata pada Sistem Chatbot Konsultasi Laktasi.</i>	<i>Diterapkannya metode Levenshtein Distance dan TF-IDF dan Cosine Similarity membuat aplikasi chatbot layak digunakan untuk customer service pusat laktasi</i>
<i>Melita, Ria & Dirjam, Taslimun, 2018.</i>	<i>Penerapan Metode Term Frequency – Inverse Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus :</i>	<i>Metode TF-IDF dan Cosine Similarity berhasil diterapkan dengan memberikan hasil berupa output dokumen, yaitu syarah hadits sesuai dengan query yang di</i>

<i>Syarah Umdatil Ahkam)</i>	<i>input-kan</i>
<i>Kavitha B. R. dan Dr, Cethana R, Murthy, 2019</i>	<i>Chatbot for healthcare system using Artificial Intelligence</i> <i>Kombinasi antara TF-IDF dan Cosine Similarity memberikan hasil respon chatbot yang sesuai.</i>
<i>Natadian Astuti, Rani & Fatchan, Muhammad, 2019.</i>	<i>Perancangan Aplikasi Chatbot untuk Industri Komersial 4.0</i> <i>Adanya aplikasi chatbot membuat peran dari customer service menjadi lebih efektif karena dapat melayani pertanyaan dari customer selama 24 jam.</i>
<i>Tirtana, A., Zulkarnain, A., Dwi Listio, Y., 2019.</i>	<i>Pembuatan Sistem Pencarian Pekerjaan Menggunakan TF-IDF</i> <i>Penerapan metode TF-IDF memberikan hasil pencarian yang lebih relevan daripada pencarian tanpa pembobotan.</i>
<i>Riyani, Ade & Burhanuddin Aulia, 2019</i>	<i>Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen</i> <i>Algoritma Cosine Similarity dan pembobotan TF-IDF telah berhasil mendeteksi kemiripan pada suatu dokumen</i>

Dari beberapa judul penelitian yang penulis telah angkat, penulis memilih penelitian yang dilakukan oleh Kavitha B. R. dan Dr. Chetana R. Murthy yang berjudul “Chatbot for healthcare system using Artificial Intelligence” sebagai referensi utama penulis dalam melakukan penelitian. Hal ini dikarenakan adanya persamaan metode yang digunakan oleh penulis, yaitu TF-IDF dan Cosine Similarity, sebagai metode utama dalam pembuatan aplikasi chatbot.

2.2. Chatbot

Program chatbot pertama ditulis oleh Joseph Weizenbaum, profesor MIT pada tahun 1966. pada waktu itu tentu saja chatbot dibuat masih amat sangat sederhana. Meskipun perkembangan kecerdasan buatan saat ini sangat pesat dan canggih, namun chatbot tetap mempertahankan kedudukannya dalam dunia Artificial Intelligence.

Chatbot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan pengguna dalam bahasa alami. Karena chatbot hanya sebuah program, dan bukan robot (chatbot tidak memiliki tubuh dan tidak memiliki mulut sehingga

tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antar manusia sebagai pengguna dengan chatbot dilakukan dengan cara mengetik apa yang akan dibicarakan dan chatbot akan memberikan respon. Orang membuat dan mengembangkan program chatbot disebut bot *master*.

Chatbot merupakan layanan masyarakat dalam bentuk layanan obrolan virtual dengan kecerdasan buatan (*Artificial Intelligence*) yang menirukan percakapan manusia melalui pesan suara, obrolan teks maupun pesan suara dan obrolan teks. Fitur chatbot telah digunakan di berbagai industri untuk penyampaian informasi atau melakukan tugas, seperti memberitahu cuaca terkini (*Weather Bot*), membantu memilih dan memesan bahan makanan (*Grocery Bot*), membantu melakukan reservasi penerbangan, membantu memberikan solusi atas suatu (*Life Advice Bot*) dan bot sebagai teman untuk bercakap-cakap seperti SimSimi. Aplikasi Chatbot ini yang akan saya buat untuk memudahkan customer dalam menanyakan pertanyaan yang diinginkan dan akan langsung direspon dengan cepat oleh sistem.

2.3. XAMPP

XAMPP adalah pengembangan *PHP* di lingkungan paling populer. *XAMPP* merupakan distribusi *Apache* yang benar-benar gratis dan mudah dipasang yang berisi *MariaDB*, *PHP*, dan *Perl*. Paket *open source* *XAMPP* telah diatur agar sangat mudah untuk diinstal dan digunakan.

Banyak orang tahu dari pengalaman mereka sendiri bahwa tidak mudah untuk menginstal *server web Apache* dan semakin sulit jika ingin menambahkan *MariaDB*, *PHP*, dan *Perl*. Tujuan *XAMPP* adalah untuk membangun distribusi instalasi yang mudah bagi para pengembang untuk memasuki dunia *Apache*. Agar nyaman bagi pengembang, *XAMPP* dikonfigurasi dengan semua fitur dihidupkan. Dalam hal penggunaan komersial, silahkan lihat lisensi produk dari sudut pandang *XAMPP* penggunaan komersial juga gratis. Saat ini ada distribusi untuk *windows*, *Linux*, dan *OS X*. *XAMPP* digunakan untuk membuat database dan menjalankan *PHP* yang telah dibuat serta tidak memerlukan biaya untuk menginstalnya.

2.4. MySQL

Menurut Kustiyahningsih (2011:145), “MySQL adalah sebuah basis data yang mengandung satu atau jumlah tabel. Tabel terdiri atas sejumlah baris dari setiap baris mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau sejumlah tabel”.

MySQL adalah multi user database yang menggunakan bahasa *Structured Query Language (SQL)*. MySQL dalam operasi *client server* melibatkan *server daemon* MySQL disisi server dan berbagai macam program serta *library* yang berjalan disisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TEX, mengaku

mampu menyimpan data lebih dari 40 database, 10.000 tabel, dan sekitar 7.000.000 baris. Totalnya kurang lebih 100 *Gigabyte* data. MySQL digunakan untuk pembuatan database dan hasil pembobotan dari semua sistem yang dijalankan dalam sistem yang dibuat.

2.5. PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang dapat digunakan untuk tujuan umum. *PHP* lebih populer digunakan untuk pengembangan aplikasi web. Dalam proses pembuatan halaman web, *PHP* tidak memerlukan kode yang panjang seperti pada *Perl* dan *Python* karena kode *PHP* dapat disisipkan di dalam kode *HTML*. *PHP* dapat dijalankan dalam sebagian besar sistem operasi, termasuk *linux*, varian-varian *UNIX (HP-UX, OpenBSD)*, *Windows*, dan *Mac OS X*. *PHP* juga mendukung sebagian besar server web yang ada saat ini, seperti: *Apache*, *IIS*, *nginx*, dan *lighttpd*. Bahasa yang digunakan dalam *CodeIgniter* menggunakan *PHP* dimana bahasa pemrograman *PHP* digunakan sebagai bahasa pemrograman umum.

2.6. JavaScript

JavaScript adalah bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen *HTML*. Dimana objek tersebut dapat berupa suatu *window*, *frame*, *URL*, dokumen, *form*, *button*, atau *item* yang lain. Yang semuanya itu mempunyai properti yang saling berhubungan dengannya, dan masing-masing memiliki nama, lokasi, warna nilai, dan atribut lain. *JavaScript* digunakan untuk pembuatan Desain Website dan Desain Chatbot yang dibuat.

2.7. TF-IDF

Metode *Term Frequency Inverse Document Frequency (TF-IDF)* adalah cara pemberian bobot hubungan suatu kata (term) terhadap dokumen.

TF murni (*raw TF*), nilai *TF* diberikan berdasarkan jumlah kemunculan suatu term di dokumen

$$tf \quad (2.7.1)$$

Inverse Document Frequency (IDF) merupakan sebuah perhitungan dari bagaimana term di distribusikan secara luas pada koleksi IDF dokumen yang bersangkutan. IDF menunjukkan hubungan ketersediaan sebuah term dalam seluruh dokumen. Semakin sedikit jumlah dokumen yang mengandung term mengandung term yang dimaksud, maka nilai IDF semakin besar.

$$IDF_j = \log(D / df_j) \quad (2.7.2)$$

Jenis formula *TF* yang biasa digunakan untuk perhitungan adalah *TF* murni (*raw TF*). Dengan demikian rumus umum untuk *Term Weighting TFIDF* adalah penggabungan dari formula perhitungan *raw TF* dengan formula *IDF* dengan cara mengalikan nilai *TF* dengan nilai *IDF*:

$$w_{ij} = tf_{ij} \times \log\left(\frac{D}{df_j}\right) \quad (2.7.3)$$

2.8. Cosine Similarity

Cosine similarity merupakan rumus yang digunakan Untuk menghitung kesamaan atau *similarity* dengan menentukan sudut antara vektor dokumen dengan vektor *query* dalam dimensi V pada bidang *Euclidean*. Hasil dari *cosinesimilarity* memiliki nilai antara 0 sampai dengan 1. Nilai 0 merupakan nilai yang didapat apabila dokumen tidak berhubungan dengan *query*, sedangkan nilai 1 berarti dokumen memiliki keterhubungan tinggi dengan *query* (Lahitani, Permanasari dan Setiawan, 2016). Cosine Similarity digunakan untuk mengambil jawaban yang ditanya oleh customer. Setelah TF-IDF selesai dihitung, bobot tertinggi akan dimasukkan ke dalam rumus *Cosine Similarity*.

$$\cos(d_j, q_k) = \frac{\sum_{i=1}^n (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^n td_{ij}^2 \times \sum_{i=1}^n tq_{ik}^2}} \quad (2.7.4)$$

Ket: $\cos(d_j, q_k)$ = Tingkat kesamaan dokumen dengan query tertentu

td_{ij} = Term ke- i dalam vektor untuk dokumen ke- j

tq_{ik} = Term ke- i dalam vektor untuk query ke- k

n = Jumlah term yang unik dalam data set

2.9. Recall dan Precision

Menurut Kurniawan (2010) *Recall* adalah perbandingan jumlah dokumen relevan yang terambil sesuai dengan *query* yang diberikan dengan total kumpulan dokumen yang relevan dengan *query*. *Precision* adalah perbandingan jumlah dokumen yang relevan terhadap *query* dengan jumlah dokumen yang terambil dari hasil pencarian. *Precision* dapat diartikan sebagai ketepatan atau kecocokan (antara permintaan informasi dengan jawaban terhadap permintaan itu). Sedangkan istilah *recall* dibidang sistem temu kembali informasi (*information retrieval*) berkaitan dengan kemampuan menemukan kembali informasi yang sudah tersimpan (Pendit 2008).

Rumus penilaian *precision* yang digunakan dalam penelitian ini adalah :

$$Precision = \frac{\text{Jumlah dokumen relevan yang ditemukan}}{\text{Jumlah semua dokumen yang ditemukan}} \times 100\%$$

Sedangkan nilai *relatif Recall* dihitung dengan rumus :
Recall

$$= \frac{\text{Jumlah dokumen relevan yang ditemukan}}{\text{Jumlah semua dokumen relevan di dalam koleksi}} \times 100\%$$

Kedua ukuran diatas biasanya diberi nilai dalam bentuk persentase, 1 sampai 100%. Sebuah sistem informasi akan dianggap baik jika tingkat *recall* maupun *precision*-nya tinggi. Jika seseorang mencari dokumen tentang “Perpustakaan” dan sistem tersebut memiliki 100 buku tentang perpustakaan maka kinerja yang paling baik adalah jika sistem tersebut berhasil menemukan 100 dokumen tentang perpustakaan.

Jika sistem tersebut memberikan 100 temuan, dan ditemukan tersebut ada 50 dokumen tentang perpustakaan, maka nilai *recall*-nya adalah 0,5 (atau 50%) dan nilai *precision*-nya juga 0,5. Kalau sistem tersebut memberikan 1 dokumen saja dan dokumen tersebut adalah “perpustakaan” maka *recall*-nya bernilai 0,01 dan *precision*-nya 1. Nilai *precision*-nya yang tinggi sebenarnya terjadi karena sistem memberikan 1 jawaban kepada pencari informasi. Kalau sistem memberikan 100 dokumen dan hanya 1 yang relevan maka nilai *recall*-nya tetap 0,01 tetapi *precision*-nya merosot ke 0,01 (Pendit 2008). Metode ini digunakan untuk menghitung kecocokan jawaban dari sistem kepada user.

3. Metode Penelitian

3.1. Studi Literatur

Studi literatur adalah pencarian referensi dari berbagai sumber yang memiliki hubungan dengan studi kasus yang ditemukan dan mencari solusi dari permasalahan yang ada. Referensi berisi tentang :

- Chatbot
- TF-IDF
- Cosine Similarity
- Scaffolding

Referensi yang dibutuhkan bisa di dapat dari jurnal, artikel laporan penelitian, dan situs-situs internet. Output dari studi literatur adalah terkumpulnya referensi yang relevan dengan studi kasus yang ada.

3.2. Identifikasi Masalah

Masalah yang dihadapi adalah sulitnya customer menghubungi admin karena keterbatasan waktu sehingga dengan adanya aplikasi diharapkan customer lebih cepat mendapat respon jawaban sesuai dengan jam bertanya.

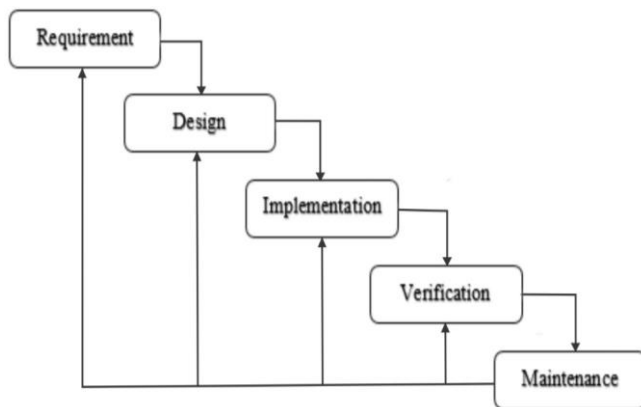
3.3. Metode Pengumpulan Data

Metode yang digunakan dalam pengambilan data adalah melalui wawancara. Wawancara dilakukan menggunakan media *Whatsapp* dengan admin penyewaan scaffolding. Wawancara dilakukan pada bulan November 2019 hingga bulan maret. Hasil wawancara yang dilakukan adalah didapatkan data barang yang disewakan, harga barang, dan screenshot *chat* admin dengan *customer*, dan informasi-informasi lain terkait dengan operasional penyewaan.

3.4. Metode Pengembangan Sistem

3.4.1. Waterfall

Metode yang digunakan dalam perancangan “Aplikasi Chatbot pada Sistem Informasi Penyewaan Scaffolding dengan Menggunakan Metode TF-IDF” adalah System Development Life Cycle (SDLC) dengan waterfall model. Waterfall model merupakan bentuk umum yang digunakan dalam perancangan sebuah sistem karena dalam setiap tahapan yang dilakukan harus diselesaikan sebelum menuju tahap selanjutnya sehingga tahapan dilakukan secara berurutan dan mendapatkan hasil yang maksimal. Tahapan-tahapan yang dilakukan yaitu :



Gambar 3.1 Waterfall Model

a. Requirement

Pada tahap ini merupakan tahap pengumpulan data untuk seluruh kebutuhan perangkat lunak yang dibutuhkan. Pengumpulan data dilakukan dengan cara observasi, wawancara, atau studi literatur.

b. Design

Pada tahap ini dilakukan penerjemahan kebutuhan sistem dalam sebuah perancangan sebelum dilakukan implementasi ke dalam bentuk *coding*. Tahap ini berfokus pada struktur data, arsitektur data, arsitektur perangkat lunak, dan representasi *interface*.

c. Implementation

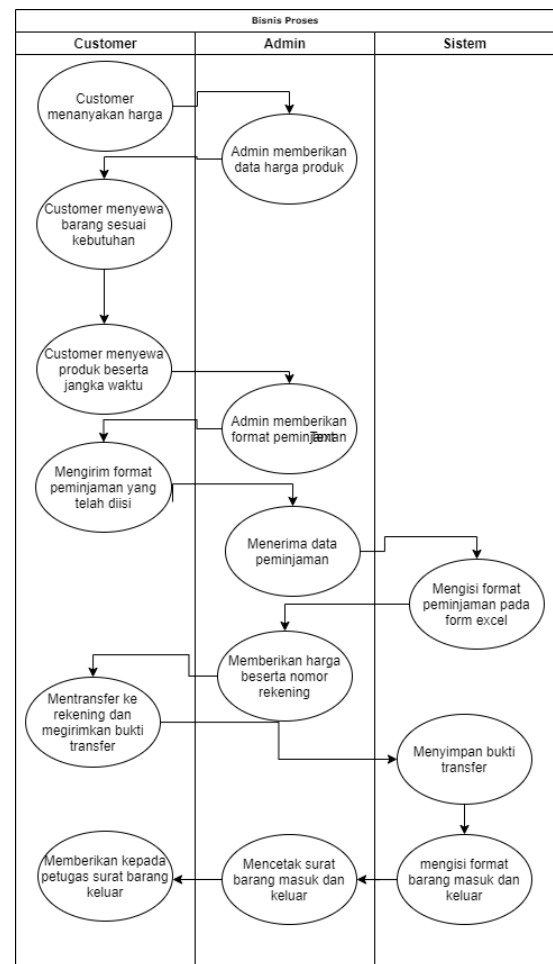
Implementasi merupakan tahap pemrograman. Pada tahap ini program dibuat sesuai dengan kebutuhan dan fungsi yang dibutuhkan dan diinginkan. Namun selain itu juga program di analisis apakah telah sesuai dengan desain sistem yang dibuat.

d. Verification

Tahap pengujian program merupakan tahap yang dilakukan setelah implementasi pembuatan, dimana tahap ini bertujuan untuk mengetahui apakah sistem yang dibuat telah sesuai dengan sistem yang diharapkan dan apakah ada kesalahan yang terjadi dari implementasi program.

e. Maintenance

Tahap maintenance merupakan tahap yang dilakukan setelah sistem yang dibuat sudah jadi. Sehingga pada tahap ini merupakan tahapan pemeliharaan dan memperbaiki kesalahan apabila ditemukan suatu masalah yang tidak ditemukan pada tahap sebelumnya.

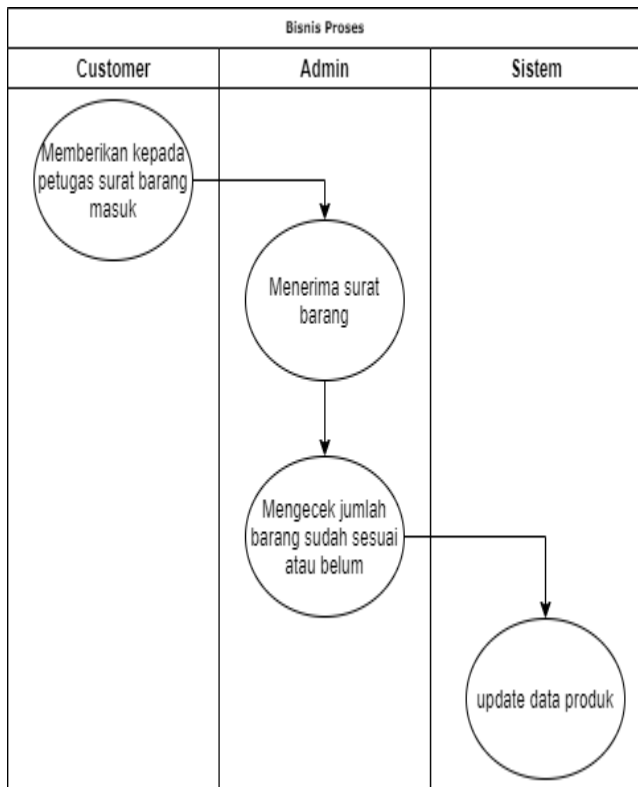


Gambar3.2 Bisnis Proses Peminjaman

f. Deskripsi Bisnis Proses

Customer menghubungi admin melalui whatsapp untuk menanyakan harga, admin memberikan data harga. Customer akan menyewa barang dengan jangka waktu dan mencantumkan nama barang, admin memberikan format peminjaman, customer mengisi format peminjaman dan memberikan foto ktp, admin mencatat dan memberikan harga yang telah ditentukan. Customer membayar melalui atm/membayar langsung, menunjukkan bukti pembayaran, menyimpan bukti pembayaran. Admin mengirimkan barang dan memberikan surat keluar dan surat masuk. Surat keluar diberikan kepada petugas.

3.4.2 Bisnis Proses



Gambar 3.3 Bisnis Proses Pengembalian

g. Deskripsi Bisnis Proses

Customer mengembalikan barang dengan memberikan surat masuk, admin menerima surat barang masuk, mengecek kelengkapan barang apakah sesuai dengan jumlah peminjaman. Kemudian admin mengupdate data produk dengan menambahkan barang yang telah dikembalikan oleh customer.

3.5. Metode Pengolahan Data

Pada penelitian ini, dilakukan pengolahan data layanan penyewaan, data barang beserta harga yang diperoleh dari wawancara dengan admin penyewaan. Data-data tersebut akan diolah menjadi data set jawaban *chatbot* yang akan dirancang. Kemudian data set yang telah diolah akan masuk ke dalam tahapan *preprocessing*. Tahapan *preprocessing* adalah sebagai berikut:

1. Case Folding

Case Folding merupakan tahapan mengubah semua huruf menjadi huruf kecil. Dengan menerapkan proses ini akan lebih memudahkan pemrosesan kata karena dapat meminimalisir logika yang digunakan dalam proses selanjutnya.

2. Tokenizing

Tokenizing merupakan tahapan pemotongan *string input* berdasarkan tiap kata yang menyusunnya. Hasil dari proses ini adalah sekumpulan kata-kata tanpa tanda baca, angka maupun karakter.

3. Filtering

Filtering merupakan tahapan untuk menghapus setiap kata-kata yang tidak penting atau kata-kata yang tidak memiliki arti. Proses *filtering* biasa disebut *stop-words*

removal. Contoh kata-kata yang dikategorikan sebagai *stop-words* yaitu: dan, dari, di, dan sebagainya.

4. Stemming

Stemming merupakan tahapan pada proses *Information Retrieval (IR)* untuk mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akar/dasar (*root word*). Proses *stemming* ini biasa digunakan dalam teks berbahasa Indonesia yang memiliki struktur imbuhan yang tetap dan mudah untuk diolah.

3.6. Perhitungan Manual

TF-IDF adalah proses pembobotan kalimat dengan menjumlahkan nilai setiap *token* per kalimat untuk masalah pencarian informasi. Ide pokok dalam TF-IDF adalah menghitung *weight* setiap kalimat untuk digunakan pada tahap selanjutnya. Setelah menghitung bobot kalimat, kalimat tersebut diurutkan dari bobot yang tertinggi. Sebagai contoh data respon *chatbot* yang akan digunakan dalam perhitungan TF-IDF ini termuat pada tabel 3.1

TABEL 3.1 CONTOH RESPON CHATBOT

Kode	Jawaban/Respon
Q	Harga sewa scaffolding
d1	Penyewaan buka pada pukul 8 pagi
d2	Harga sewamain frame 1,9 Rp55.000 untuk sebulan
d3	Harga sewa main frame 1,9 Rp45.000 untuk 2 minggu
d4	Harga sewa main frame 1,9 Rp35.000 untuk 1 minggu
d5	Hari Jumat dan Minggu Libur.

Proses TF-IDF yang berdasarkan data pada tabel 3.1 sebagai contoh perhitungan TF menggunakan proses *preprocessing*. Dalam melakukan perhitungan digunakan beberapa dokumen untuk mempermudah perhitungan TF-IDF. Hasil proses TF-IDF termuat pada tabel 3.2 dan tabel 3.3.

TABEL 3.2 PERHITUNGAN DF

Dokumen	q	d1	d2	d3	d4	d5	df	log(d/df)
Harga	1	0	1	1	1	0	4	1,5
Sewa	1	1	1	1	1	0	5	1
Scaffolding	1	0	0	0	0	0	1	6
Buka	0	1	0	0	0	0	1	6
Pukul	0	1	0	0	0	0	1	6
Pagi	0	1	0	0	0	0	1	6
Main frame	0	0	1	1	1	0	3	2
Untuk	0	0	1	0	0	0	1	6
bulan	0	0	0	1	1	1	3	2

<i>Minggu</i>	0	0	0	0	0	1	1	6
<i>Hari</i>	0	0	0	0	0	1	1	6
<i>Jumat</i>	0	0	0	0	0	1	1	6
<i>Libur</i>	1	0	1	1	1	0	4	1,5

<i>Jumlah</i>	0	0,031008	0,031008	0,031008	0
---------------	---	----------	----------	----------	---

Langkah selanjutnya yaitu melakukan perkalian vektor tiap query key jawaban dengan query jawaban. Hasil perkalian vektor ini dapat dilihat dalam tabel 3.7.

TABEL 3.3 PERHITUNGAN TF-IDF

<i>Dokumen</i>	<i>Q</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>d5</i>
<i>Harga</i>	0,1761	0	0,1761	0,1761	0,1761	0
<i>Sewa</i>	0	0	0	0	0	0
<i>Scaffolding</i>	0,7782	0	0	0	0	0
<i>Buka</i>	0	0,7782	0	0	0	0
<i>Pukul</i>	0	0,7782	0	0	0	0
<i>Pagi</i>	0	0,7782	0	0	0	0
<i>Main frame</i>	0	0	0,3010	0,3010	0,3010	0
<i>Untuk</i>	0	0	0,7782	0	0	0
<i>Bulan</i>	0	0	0	0,3010	0,3010	0,3010
<i>Minggu</i>	0	0	0	0	0	0,7782
<i>Hari</i>	0	0	0	0	0	0,7782
<i>Jumat</i>	0	0	0	0	0	0,7782
<i>Libur</i>	0,1761	0	0,1761	0,1761	0,1761	0

Kemudian dihitung hasil perkalian skalar masing-masing query jawaban terhadap query key jawaban. Hasil perkalian dari setiap jawaban dengan query dijumlahkan. Proses perkalian skalar ini dapat dilihat dalam tabel 3.6.

Tabel 3.4Perkalian Skalar

<i>Dokumen</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>	<i>d4</i>	<i>d5</i>
<i>Harga</i>	0	0,031008	0,031008	0,031008	0
<i>Sewa</i>	0,006270	0,006270	0,006270	0,006270	0
<i>Scaffolding</i>	0	0	0	0	0
<i>Buka</i>	0	0	0	0	0
<i>Pukul</i>	0	0	0	0	0
<i>Pagi</i>	0	0	0	0	0
<i>Main frame</i>	0	0	0	0	0
<i>Untuk</i>	0	0	0	0	0
<i>Bulan</i>	0	0	0	0	0
<i>Minggu</i>	0	0	0	0	0
<i>Hari</i>	0	0	0	0	0
<i>Jumat</i>	0	0	0	0	0
<i>Libur</i>	0,006270	0,037278	0,037278	0,037278	0

TABEL 3.5 PERKALIAN VEKTOR

<i>Dokumen</i>	<i>Q</i>	<i>d1</i>	<i>d2</i>	<i>d3</i>
<i>Harga</i>	0,031008	0	0,031008	0,031008
<i>Sewa</i>	0,00627	0,00627	0,00627	0,00627
<i>Scaffolding</i>	0,605519	0	0	0
<i>Buka</i>	0	0,605519	0	0
<i>Pukul</i>	0	0,605519	0	0
<i>Pagi</i>	0	0,605519	0	0
<i>Main frame</i>	0	0	0,090619	0,090619
<i>Untuk</i>	0	0	0,605519	0
<i>Bulan</i>	0	0	0	0,090619
<i>Minggu</i>	0	0	0	0
<i>Hari</i>	0	0	0	0
<i>Jumat</i>	0	0	0	0
<i>Libur</i>	0,642797	1,822828	0,733416	0,218516
<i>Jumlah</i>	0,031008	0	0,031008	0,031008
<i>Jumlah Akar</i>	0,006270	0,006270	0,006270	0,006270

Langkah terakhir adalah menghitung nilai *Cosine Similarity* dengan rumus seperti yang tercantum dalam rumus 2.10.1. Hasil perhitungan *Cosine Similarity* dapat dilihat dalam tabel 3.6.

Tabel 3.6 Hasil Perhitungan Similarity

<i>Kode</i>	<i>Dokumen</i>	<i>Nilai Similarity</i>	<i>Persentase</i>
<i>d1</i>	<i>Penyewaan buka</i>	0,005792	0,58%

	pada pukul 8 pagi		
d2	Harga sewa main frame 1,9 Rp55.000 untuk sebulan	0,032241	3,22%
d3	Harga sewa main frame 1,9 Rp45.000 untuk 2 minggu	0,093118	9,31%
d4	Harga sewa main frame 1,9 Rp35.000 untuk 1 minggu	0,170595	17,06%
d5	Hari Jumat dan Minggu Libur.	0	0%

Berdasarkan hasil perhitungan *Cosine Similarity* pada tabel di atas, maka bisa disimpulkan jika respon pada dokumen ke-2 merupakan respon yang paling mendekati dari pertanyaan yang diajukan oleh user.

4. Hasil

Data yang digunakan dalam penelitian ini adalah data harga layanan percetakan yang sudah diubah ke dalam bentuk dokumen jawaban *chatbot*. Sebelum diimplementasikan ke dalam algoritma *cosine similarity*, data dokumen jawaban tersebut akan masuk ke dalam tahap *preprocessing*. Setelah dokumen tersebut melalui tahap *preprocessing* maka akan masuk ke dalam tahap pembobotan TF-IDF. Penghitungan bobot TF-IDF dilakukan untuk memberi bobot tiap *term* yang terdapat pada kalimat yang dicari oleh *user*. Penghitungan bobot TF-IDF menggunakan rumus (3) dan menghasilkan pembobotan sebagai berikut:

No	Term	Doc-id	Count	Bobot
1	selamat	1	1	0.81083
2	pagi	1	1	1.50408
3	selamat	2	1	0.81083
4	siang	2	1	2.19722
5	selamat	3	1	0.81083
6	siang	3	1	2.19722
7	selamat	4	1	0.81083
8	malam	4	1	2.19722
9	penyewaan	5	1	1.50408

GAMBAR 6.1 CONTOH PERHITUNGAN BOBOT TF-IDF

PADA GAMBAR 6.1 MERUPAKAN CONTOH HASIL DARI PENGHITUNGAN TF-IDF YANG TELAH DILAKUKAN OLEH SISTEM. SETELAH DILAKUKAN PEMBOBOTAN KATA, MAKA AKAN DILAKUKAN PENGHITUNGAN MENGGUNAKAN COSINE SIMILARITY UNTUK MEMBANDINGKAN TINGKAT KEMIRIPAN ANTAR DOKUMEN DENGAN QUERY YANG DIMASUKKAN OLEH USER. PENGHITUNGAN COSINE SIMILARITY MENGGUNAKAN RUMUS SESUAI DENGAN RUMUS (4). SEBAGAI CONTOH USER MEMASUKKAN INPUT “HARGA BANNER KOREA” KE DALAM CHATBOT. HASIL PENGHITUNGAN COSINE SIMILARITY MENGHASILKAN 3 DOKUMEN JAWABAN YANG MEMILIKI TINGKAT

KEMIRIPAN TERTINGGI DIANTARA DOKUMEN LAINNYA. TINGKAT KEMIRIPAN 3 DOKUMEN TERSEBUT DAPAT DILIHAH PADA TABEL 6.1.

TABEL 6.1 HASIL PENGHITUNGAN COSINE SIMILARITY

No.	QUERY USER	DOKUMEN	NILAI SIMILARITY
1	HARGA SEWA SCAFFOLDING	HARGA SEWA 1 SET MAIN FRAME 1,7 SELAMA 1 MINGGU 30.000 RUPIAH, SELAMA 2 MINGGU 40.000 RUPIAH, SELAMA 1 BULAN 50.000 RUPIAH	0.192847
2		PENYEWAAN BUKA PUKUL 8 PAGI	0.365148
3		Penyewaan buka pada hari Senin, Selasa, Rabu, Kamis, dan Sabtu	0.374634

BERDASARKAN HASIL PADA TABEL 6.1 MAKA DIDAPATI PADA DOKUMEN YANG KE 3 MERUPAKAN DOKUMEN DENGAN TINGKAT SIMILARITAS TERTINGGI TERHADAP QUERY YANG DIINPUTKAN OLEH USER. DOKUMEN TERSEBUT KEMUDIAN DIKIRIM SEBAGAI RESPON CHATBOT ATAS KALIMAT YANG TELAH DIINPUTKAN OLEH USER.

5. Pembahasan

Berdasarkan dari hasil pengujian data-data penelitian yang telah dilakukan oleh peneliti, berikut paparan pembahasan hasil penelitian:

Penggunaan metode TF-IDF dan *cosine similarity* dapat diterapkan ke dalam sistem *chatbot* untuk merespon permintaan informasi dari *user*.

Perhitungan TF-IDF dapat digunakan untuk membobotkan setiap *term* yang dicari oleh *user* dalam koleksi dokumen jawaban.

Metode *cosine similarity* dapat digunakan untuk melakukan perhitungan tingkat similaritas antar dokumen terhadap *query* yang dicari oleh *user*.

Penggunaan metode *cosine similarity* kurang cocok diterapkan pada *chatbot* penjualan online karena tidak dapat merespon permintaan pesanan dari *user*.

6. Kesimpulan dan Saran

6.1. Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab I hingga bab VI, maka dapat disimpulkan bahwa Sistem ini dapat Melakukan Tokenizing (Memisah kata penyusun dari suatu dokumen) dan menghitung bobot Tf-Idf setiap kata tersebut pada tiap array. Kemudian menggunakan Metode TF-IDF dan Cosine Similarity untuk mencari jawaban pada sistem. Sehingga saat user menuliskan

pertanyaan, akan muncul jawaban sesuai harapan pelanggan. Sistem juga dapat memudahkan user dalam menerima informasi sesuai dengan yang diharapkan. Sehingga tugas Operator menjadi ringan karena sudah tidak lagi harus selalu membuka *WhatsApp* untuk melayani para pelanggan yang membutuhkan informasi tentang penyewaan scaffolding.

6.2. Saran

Untuk pengembangan sistem informasi Aplikasi Chatbot pada Sistem Informasi Penyewaan Scaffolding Menggunakan metode TF-IDF lebih lanjut agar semakin memberikan manfaat untuk user ada beberapa hal yang bisa dijadikan bahan kajian lebih lanjut, yaitu

Untuk peneliti selanjutnya disarankan *chatbot* dapat dikembangkan lebih lanjut.

DaftarPustaka:

- [1] Andreev, Andrey (2018, July 11) CodeIgniter [Online]. Available : <https://codeigniter.com/>.Oswald and Kay, (2002) Apache Friends [Online]. Available:<https://www.apachefriends.org/index.html>.
- [2] Astiningrum, Mungki & Shoburu Rohmah, Maya. (2018). Implementasi NLP dengan Konversi Kata pada Sistem Chatbot Konsultasi Laktasi. *Jurnal Informatika Polinema*, 5(1): 46-52.
- [3] Dwi Listio, Y., Zulkarnain, A., Tirtana, A. (2019). Pembuatan Sistem Pencarian Pekerjaan Menggunakan TF-IDF. *Jurnal Ilmiah Teknologi Informasi Asia*, 13(2): 91-100.
- [4] Melita, Ria & Dirjam, Taslimun. (2018). Penerapan Metode Term Frequency Inverse DocumentFrequency dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam). *JurnalTeknik Informatika*, 11(2): 149-164.
- [5] Natadian Astuti, Rani & Fatchan, Muhammad. (2019). Perancangan Aplikasi Teknologi Chatbot untuk Industri Komersial 4.0. *e-Prosiding SNasTekS 2019*, 4: 339-348.
- [6] Riyani, Ade & Burhanuddin, Auliya. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. *Jurnal Linguistik Komputasional*, 2(1): 23-27
- [7] R. Kavitha B. & Murthy Chethana R. "Chatbot for healthcare system using Artificial Intelligence," International Journal of Advance Research, Ideas and Innovations in Technology, vol.5 pp.1304-1307, June, 2019.
- [8] Satria Paliwahet, I Nyoman & Gede Darma Putra, I Ketut. (2017). Pencarian Informasi Wisata Daerah Bali Menggunakan Teknologi Chatbot. *Lontar Komputer*, 8(3): 144-153
- [9] Suryani, Dhebys & Larasati, Eka. (2017). Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML. *SMARTICS Journal*: 3, 47-54.
- [10] Suryani, Dhebys & Putera, Yoga. (2018). Aplikasi Chatbot Berbasis Web Pada Sistem Informasi Layanan Publik Kesehatan di Malang dengan Menggunakan Metode TF-IDF. *JurnalInformatika Polinema*, 4(3): 224-228.
- [11] Suryani, Dhebys & Aulia, Indinabilah. (2018). Penerapan Metode TF-IDF dan N-Gram pada Pengembangan Aplikasi Chatbot Berbasis LINE untuk Layanan Publik Kesehatan di Kota Malang. *Jurnal Informatika Polinema*. 5(1): 7-11.
- [12] Yamaguchi, Hiroshi & Mozgoyov, Maxim. (2018). A Chatbot Based on AIML Rules Extracted From Twitter Dialogues. *Communication Paper Poznan*, 17: 37-42.