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

**SKRIPSI**

**Oleh:**

**TRIANTA ALMIRA RAMADHANI NIM. 1641720097**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**JULI 2020**

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

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**TRIANTA ALMIRA RAMADHANI NIM. 1641720097**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**JULI 2020**

HALAMAN PENGESAHAN

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

Disusun oleh:

TRIANTA ALMIRA RAMADHANI NIM. 1641720097

Skripsi ini telah diuji pada tanggal 30 Juni 2020

Disetujui oleh:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Pembimbing I | : | Dimas Wahyu Wibowo, ST., MT.  NIP. 19841009 201504 1 001 | ........................... |
|  |  |  |  |  |
| 2. | Pembimbing II | : | Habibie Ed Dien, S.KOM., MT.  NIDN. 0012049209 | ........................... |
|  |  |  |  |  |
| 3. | Penguji I | : | Ir. Deddy Kusbianto P.A, M.MKom.  NIP. 19621128 198811 1 001 | ........................... |
|  |  |  |  |  |
| 4. | Penguji II | : | Budi Harijanto, S.T., M.MKom.  NIP. 19840610 200812 1 004 | ........................... |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, S.T., M.Cs. | Imam Fahrur Rozi, S.T., M.T. |
| NIP. 19711110 199903 1 002 | |  | | --- | | NIP. 19840610 200812 1 004 | |

# PERNYATAAN

Dengan ini saya menyatakan bahwa pada Skripsi ini tidak terdapat karya, baik seluruh maupun sebagian, yang sudah pernah diajukan untuk memperoleh gelar akademik di Perguruan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini serta disebutkan dalam daftar sitasi/pustaka.

|  |  |
| --- | --- |
|  | Malang, 30 Juni 2020  Trianta Almira R. |

# ABSTRAK

**Ramadhani, Trianta Almira**. “Aplikasi Chatbot pada Sistem Informasi Penyewaan Scaffolding dengan Menggunakan Metode TF-IDF”. **Pembimbing: (1) Dimas Wahyu Wibowo, ST., MT., (2)** **Habibie Ed Dien, S.KOM., MT.**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2020.**

CV. Scaffolding Samarinda merupakan usaha penyewaan scaffolding di Samarinda dan sekitarnya. Salah satu permasalahan yang dimiliki oleh CV. Tersebut adalah keterbatasan waktu jam kerja admin *Customer Service* merespon pertanyaan dari pelanggan yang membutuhkan informasi mengenai penyewaan scaffolding. Dikarenakan admin *Customer Service* memiliki keterbatasan jam kerja yang ditentukan oleh peraturan perusahaan. Sehingga pelanggan sulit berkomunikasi dengan *Customer Service* diluar jam kerja admin. Sedangkan pelanggan membutuhkan respon yang cepat untuk menangani kendala pada perusahaannya yang membutuhkan jasa dari CV. Scaffolding Samarinda.

Oleh sebab itu, CV. Scaffolding Samarinda membutuhkan pemberian layanan system 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 user menuliskan pertanyaan, akan muncul jawaban sesuai harapan *user*. Sistem juga dapat memudahkan *user* dalam menerima informasi sesuai dengan yang diharapkan saat itu juga.

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

# *ABSTRACT*

***Ramadhani, Trianta Almira****. “Chatbot Application on Scaffolding Rental Information System using TF-IDF Method”.* ***Counseling Lecturer: (1) Dimas Wahyu Wibowo, ST., MT., (2) Habibie Ed Dien, S.KOM., MT.***

***Thesis, Informatics Management Study Program, Department of Information Technology, State Polytechnic of Malang, 2020.***

*CV. Scaffolding Samarinda is a scaffolding rental business in Samarinda and surrounding areas. One that was approved by CV. Such is the limitation of the Customer Service admin's working hours to answer questions from customers who need information about scaffolding rental. Because the Customer Service admin has limited working hours determined by company regulations. Contacting customers is difficult to communicate with Customer Service outside admin working hours. While customers need a quick response to request a loan from their company that needs the services of a CV. Samarinda Scaffolding.*

*Therefore, CV. Scaffolding Samarinda requires a fast information service system at all times. Based on the tests that have been done, it can be concluded that this system can do Tokenizing and separate the TF-IDF and Cosine Similarity to find answers to the system. When asked the user to ask questions, answers will appear according to user expectations. The system can also provide the user in receiving information as expected at that time..*

***Keywords:*** *Chatbot, TF-IDF, Cosine Similarity, Scaffolding*

# KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “APLIKASI CHATBOT PADA SISTEM INFORMASI PENYEWAAN SCAFFOLDING DENGAN MENGGUNAKAN METODE TF-IDF”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari bahwasannya dengan tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Allah SWT atas segala rahmat dan hidayah-Nya memberikan ilmu yang bermanfaat dan berguna.
2. Orang Tua yang selalu mensupport, mendoakan, mendukung saya dalam berbagai hal untuk memberikan motivasi terselesaikannya skripsi tepat waktu.
3. Bapak Rudy Ariyanto, ST., M.Cs., selaku ketua jurusan Teknologi Informasi
4. Bapak Imam Fahrur Rozi, ST., MT., selaku ketua program studi Manajemen Informatika
5. Bapak Dimas Wahyu Wibowo, ST., MT., selaku dosen pembimbing I
6. Bapak Habibie Ed Dien, S.Kom., MT., selaku dosen pembimbing II
7. Bapak Ibu Dosen Penguji yang sudah menguji kemampuan saya.
8. H. Ahmad Soedharmo selaku kakek sekaligus ayah yang senantiasa mendoakan, mendorong serta mengingatkan selalu selama saya kuliah 4 tahun di POLINEMA. Sosok yang menjadikan saya bersemangat untuk menyelesaikan skripsi.
9. Hj. Erna Juwita selaku tante sekaligus pengganti ibu selama saya kuliah 4 tahun, yang mendorong saya untuk segera menyelesaikan dan tertib dalam melakukan semua kegiatan.
10. Joshua, Wella, Ayu, Yayas, Aang, Mbak Tya, Yoga, Agung, Ammar, Fika teman seperjuangan dari lulus SMA hingga saat ini.
11. Dan seluruh pihak yang telah membantu lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, 30 Juni 2020

Penulis

**DAFTAR ISI**

Halaman

[HALAMAN SAMPUL i](#_Toc42346694)

[HALAMAN JUDUL ii](#_Toc42346695)

[HALAMAN PENGESAHAN iii](#_Toc42346696)

[PERNYATAAN iv](#_Toc42346697)

[ABSTRAK v](#_Toc42346698)

[*ABSTRACT* vi](#_Toc42346699)

[KATA PENGANTAR iv](#_Toc42346700)

[DAFTAR ISI iv](#_Toc42346701)

[DAFTAR GAMBAR vi](#_Toc42346702)

[DAFTAR TABEL viii](#_Toc42346703)

[DAFTAR LAMPIRAN x](#_Toc42346704)

[BAB I. PENDAHULUAN 1](#_Toc42346705)

[1.1 Latar Belakang 1](#_Toc42346706)

[1.2 Rumusan Masalah 2](#_Toc42346707)

[1.3 Tujuan 2](#_Toc42346708)

[1.4 Batasan Masalah 2](#_Toc42346709)

[1.5 Sistematika Penulisan 3](#_Toc42346710)

[BAB II. LANDASAN TEORI 5](#_Toc42346711)

[2.1 Penelitian Terdahulu 5](#_Toc42346712)

[2.2 Chatbot 7](#_Toc42346713)

[2.3 XAMPP 7](#_Toc42346714)

[2.4 MySQL 8](#_Toc42346715)

[2.5 PHP 8](#_Toc42346716)

[2.6 JavaScript 9](#_Toc42346717)

[2.7 TF-IDF 9](#_Toc42346718)

[*2.8* *Cosine Similarity* 10](#_Toc42346719)

[2.9 *Recall* dan *Precision* 10](#_Toc42346720)

[BAB III. METODOLOGI PENELITIAN 12](#_Toc42346721)

[3.1 Studi Literatur 12](#_Toc42346722)

[3.2 Identifikasi Masalah 12](#_Toc42346723)

[3.3 Metode Pengumpulan Data 12](#_Toc42346724)

[3.4 Metode Pengembangan Sistem 12](#_Toc42346725)

[3.4.1 Waterfall 12](#_Toc42346726)

[3.4.2 Bisnis Proses 14](#_Toc42346727)

[3.5 Metode Pengolahan Data 16](#_Toc42346728)

[3.6 Perhitungan Manual 16](#_Toc42346729)

[BAB IV. ANALISIS DAN PERANCANGAN SISTEM 21](#_Toc42346730)

[4.1 Gambaran Umum Sistem 21](#_Toc42346731)

[4.2 Analisa Permasalahan 22](#_Toc42346732)

[4.3 Analisa Kebutuhan Non Fungsional 22](#_Toc42346733)

[4.4 Analisa Kebutuhan Fungsional 23](#_Toc42346734)

[4.5 Analisis Data 23](#_Toc42346735)

[4.6 Perancangan Antar Muka Pengguna 44](#_Toc42346736)

[4.7 Perancangan Basisdata 40](#_Toc42346737)

[4.8 Perancangan Sistem 24](#_Toc42346738)

[BAB V. IMPLEMENTASI DAN PENGUJIAN 47](#_Toc42346739)

[5.1 Implementasi Basisdata 47](#_Toc42346740)

[5.2 Implementasi Anatarmuka Pengguna 56](#_Toc42346741)

[5.3 Implementasi Sistem 47](#_Toc42346742)

[5.4 Pengujian Fungsional 60](#_Toc42346743)

[5.5 Pengujian Akurasi 65](#_Toc42346744)

[BAB VI. HASIL DAN PEMBAHASAN 67](#_Toc42346745)

[BAB VII. KESIMPULAN DAN SARAN 69](#_Toc42346746)

[7.1 Kesimpulan 69](#_Toc42346747)

[7.2 Saran 69](#_Toc42346748)

[DAFTAR PUSTAKA 70](#_Toc42346749)

[LAMPIRAN - LAMPIRAN 72](#_Toc42346750)

# DAFTAR GAMBAR

Halaman

[Gambar 3.1 Waterfall Model 13](#_Toc44178789)

[Gambar 3.2 Bisnis Proses Peminjaman 14](#_Toc44178790)

[Gambar 3.3 Bisnis Proses Pengembalian 15](#_Toc44178791)

[Gambar 4.1 Flowchart Alur Proses Chatbot 21](#_Toc44182624)

[Gambar 4.2 Use Case 24](#_Toc44182625)

[Gambar 4.3 Flowchart Pembobotan TF-IDF 29](#_Toc44182626)

[Gambar 4.4 Flowchart Perhitungan Cosine Similarity 30](#_Toc44182627)

[Gambar 4.5 Context Diagram 31](#_Toc44182628)

[Gambar 4.6 DFD Level 1 31](#_Toc44182629)

[Gambar 4.7 DFD Level 2 Proses 1 32](#_Toc44182630)

[Gambar 4.8 DFD Level 2 Proses 2 32](#_Toc44182631)

[Gambar 4.9 DFD Level 2 Proses 3 33](#_Toc44182632)

[Gambar 4.10 DFD Level 2 Proses 4 33](#_Toc44182633)

[Gambar 4.11 DFD Level 2 Proses 5 34](#_Toc44182634)

[Gambar 4.12 DFD Level 2 Proses 6 34](#_Toc44182635)

[Gambar 4.13 DFD Level 2 Proses 7 35](#_Toc44182636)

[Gambar 4.14 DFD Level 2 Proses 8 35](#_Toc44182637)

[Gambar 4.15 DFD Level 2 Proses 9 35](#_Toc44182638)

[Gambar 4.16 Entitas Relationship Diagram 40](#_Toc44182639)

[Gambar 4.17 Mockup Halaman Awal Sistem 44](#_Toc44182640)

[Gambar 4.18 Mockup Halaman Chat 44](#_Toc44182641)

[Gambar 4.19 Mockup Halaman Login 45](#_Toc44182642)

[Gambar 4.20 Mockup Halaman Utama Admin 45](#_Toc44182643)

[Gambar 4.21 Mockup Halaman Data Jawaban 45](#_Toc44182644)

[Gambar 4.22 Mockup Halaman Perhitungan TF-IDF 46](#_Toc44182645)

[Gambar 4.23 Mockup Halaman Data Pertanyaan 46](#_Toc44182646)

[Gambar 5.1 Struktur Database 47](#_Toc44182647)

[Gambar 5.2 Tabel Admin 47](#_Toc44182648)

[Gambar 5.3 Tabel Dokumen 48](#_Toc44182649)

[Gambar 5.4 Tabel Index 48](#_Toc44182650)

[Gambar 5.5 Tabel Vektor 48](#_Toc44182651)

[Gambar 5.6 Tabel Cache 48](#_Toc44182652)

[Gambar 5.7 Tabel Stem 49](#_Toc44182653)

[Gambar 5.8 Tabel Product 49](#_Toc44182654)

[Gambar 5.9 Tabel Customer 49](#_Toc44182655)

[Gambar 5.10 Tabel Transaksi 49](#_Toc44182656)

[Gambar 5.11 Tabel Detail Transaksi 50](#_Toc44182657)

[Gambar 5.12 Potongan Kode Menampilkan Data Master Jawaban 50](#_Toc44182658)

[Gambar 5.13 Potongan Kode Preprocessing 51](#_Toc44182659)

[Gambar 5.14 Potongan Kode Hitung Bobot TF-IDF 52](#_Toc44182660)

[Gambar 5.15 Potongan Kode Hitung Panjang Vektor Dokumen 53](#_Toc44182661)

[Gambar 5.16 Potongan Kode Hitung Similarity 55](#_Toc44182662)

[Gambar 5.17 Potongan Kode Penentuan Respon Chatbot 56](#_Toc44182663)

[Gambar 5.18 Halaman Login 56](#_Toc44182664)

[Gambar 5.19 Halaman Dashboard 57](#_Toc44182665)

[Gambar 5.20 Halaman Master Jawaban 57](#_Toc44182666)

[Gambar 5.21 Halaman Perhitungan TF-IDF 58](#_Toc44182667)

[Gambar 5.22 Halaman Pertanyaan dari Customer 58](#_Toc44182668)

[Gambar 5.23 Halaman Utama 59](#_Toc44182669)

[Gambar 5.24 Halaman Chatbot 59](#_Toc44182670)

[Gambar 6.1 Contoh Perhitungan Bobot TF-IDF 67](#_Toc44182671)

# DAFTAR TABEL

Halaman

[Tabel 2.1 Penelitian Terdahulu 5](#_Toc44178357)

[Tabel 3.1 Contoh Respon Chatbot 17](#_Toc44182479)

[Tabel 3.2 Perhitungan DF 17](#_Toc44182480)

[Tabel 3.3 Perhitungan TF-IDF 18](#_Toc44182481)

[Tabel 3.4 Perkalian Skalar 18](#_Toc44182482)

[Tabel 3.5 Perkalian Vektor 19](#_Toc44182483)

[Tabel 3.6 Hasil Perhitungan Similarity 20](#_Toc44182484)

[Tabel 4.1 Spesifikasi Minimum Perangkat Lunak 22](#_Toc44182521)

[Tabel 4.2 Spesifikasi Minimum Perangkat Keras 23](#_Toc44182522)

[Tabel 4.3 Deskripsi Use Case 25](#_Toc44182523)

[Tabel 4.4 Skenario Use Case Melakukan Login 25](#_Toc44182524)

[Tabel 4.5 Skenario Use Case Mengelola Data Produk 26](#_Toc44182525)

[Tabel 4.6 Skenario Use Case Mengelola Data Customer 26](#_Toc44182526)

[Tabel 4.7 Skenario Use Case Mengelola Data Transaksi 27](#_Toc44182527)

[Tabel 4.8 Skenario Use Case Mengelola Data Chatbot 27](#_Toc44182528)

[Tabel 4.9 Skenario Use Case Mengajukan Pertanyaan pada Sistem 28](#_Toc44182529)

[Tabel 4.10 Skenario Use Case Mendapatkan Respon dari Chatbot 28](#_Toc44182530)

[Tabel 4.11 Spesifikasi proses login 36](#_Toc44182531)

[Tabel 4.12 Spesifikasi proses kelola jawaban 36](#_Toc44182532)

[Tabel 4.13 Spesifikasi proses kelola pertanyaan 37](#_Toc44182533)

[Tabel 4.14 Spesifikasi kelola produk 37](#_Toc44182534)

[Tabel 4.15 Spesifikasi kelola customer 37](#_Toc44182535)

[Tabel 4.16 Spesifikasi kelola transaksi 38](#_Toc44182536)

[Tabel 4.17 Spesifikasi kelola pengaturan 38](#_Toc44182537)

[Tabel 4.18 Spesifikasi kelola token 38](#_Toc44182538)

[Tabel 4.19 Spesifikasi kelola chatting 39](#_Toc44182539)

[Tabel 4.20 Tabel Admin 40](#_Toc44182540)

[Tabel 4.21 Tabel Cache 41](#_Toc44182541)

[Tabel 4.22 Tabel Customer 41](#_Toc44182542)

[Tabel 4.23 Tabel Detail 41](#_Toc44182543)

[Tabel 4.24 Tabel Dokumen 42](#_Toc44182544)

[Tabel 4.25 Tabel Index 42](#_Toc44182545)

[Tabel 4.26 Tabel Produk 42](#_Toc44182546)

[Tabel 4.27 Tabel Stem 43](#_Toc44182547)

[Tabel 4.28 Tabel Transaksi 43](#_Toc44182548)

[Tabel 4.29 Tabel Vektor 44](#_Toc44182549)

[Tabel 5.1 Pengujian Fungsional Halaman Login Admin 60](#_Toc44182580)

[Tabel 5.2 Pengujian Fungsional Menu Admin 60](#_Toc44182581)

[Tabel 5.3 Pengujian Fungsional Admin Data Produk 61](#_Toc44182582)

[Tabel 5.4 Pengujian Fungsional Admin Data Pelanggan 62](#_Toc44182583)

[Tabel 5.5 Pengujian Fungional Admin Data Transaksi 62](#_Toc44182584)

[Tabel 5.6 Pengujian Fungsional Admin Data Master 63](#_Toc44182585)

[Tabel 5.7 Pengujian Fungsional Admin Hasil Perhitungan 64](#_Toc44182586)

[Tabel 5.8 Pengujian Fungsional Admin Menu History Jawaban 64](#_Toc44182587)

[Tabel 5.9 Pengujian Fungsional User Chatbot 64](#_Toc44182588)

[Tabel 5.10 Tabel Pengujian Akurasi 65](#_Toc44182589)

[Tabel 6.1 Hasil Penghitungan Cosine Similarity 68](#_Toc44182603)

# DAFTAR LAMPIRAN

Lampiran 1 Source Code Tampilan Chatbot

Lampiran 2 Source Code Perhitungan *TF-IDF*

**BAB I. PENDAHULUAN**

* 1. **Latar Belakang**

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.

CV. Scaffolding Samarinda merupakan usaha penyewaan scaffolding di Samarinda dan sekitarnya. Salah satu permasalahan yang dimiliki oleh CV. Tersebut adalah keterbatasan waktu jam kerja admin Customer Service merespon pertanyaan dari pelanggan yang membutuhkan informasi mengenai penyewaan scaffolding. Dikarenakan admin *Customer Service* memiliki jam kerja yang ditentukan oleh peraturan perusahaan. Sehingga pelanggan sulit berkomunikasi dengan Customer Service diluar jam kerja admin. Sedangkan pelanggan membutuhkan respon yang cepat untuk menangani kendala pada perusahaannya yang membutuhkan jasa dari CV. Scaffolding Samarinda.

TF-IDF (*Term Frequency - Inverse Document Frequemcy*) 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).

Sehingga 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. Aplikasi chatbot ini diharapkan dapat menjawab pertanyaan dari calon pelanggan dengan informasi/penjelasan yang mudah dipahami.

* 1. **Rumusan Masalah**

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil adalah sebagai berikut:

1. Bagaimana mengatasi *Customer* yang mengharapkan informasi yang cepat setiap saat dalam waktu 24 jam?
   1. **Tujuan**

Tujuan dari pembuatan sistem chatbot ini adalah sebagai berikut :

1. Untuk memudahkan pelanggan saat ingin bertanya mengenai penyewaan scaffolding dengan jawaban sesuai jam bertanya dan pertanyaan beruntut akan menghasilkan jawaban yang beruntut
   1. **Batasan Masalah**

Agar skripsi penulis yang berjudul Aplikasi Chatboot pada Sistem Informasi Penyewaan Scaffolding dengan Menggunakan Metode TF-IDF dapat berjalan sesuai dengan rencana dan tujuan awal, maka penulis memberikan batasan-batasan masalah yaitu :

1. Informasi mengenai barang yang tersedia
2. Informasi mengenai harga sewa barang per unit
3. Informasi mengenai jangka waktu peminjaman barang
4. Informasi mengenai berat barang per unit
   1. **Sistematika Penulisan**

Sistematika penulisan laporan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan skrisi ini adalah sebalagi berikut :

BAB I. PENDAHULUAN

Menguraikan tentang latar belakang permasalahan, mencoba merumuskan inti permasalahan yang dihadapi, menentukan tujuan dan kegunaan penelitian serta sistematika penulisan

BAB II. LANDASAN TEORI

Membahas berbagai konsep dasar dan teori-teori yang berkaitan dengan topik penelitian yang dilakukan dan hal-hal yang bergina dalam proses analisis permasalahan.

BAB III. METODELOGI PENELITIAN

Membandingkan sistem pelayanan yang digunakan selama ini melalui Whatsapp dengan Chatbot, meningkatkan sistem pelayanan untuk mendapatkan informasi sesuai dengan jam pertanyaan yang , pengambilan data penelitian, lokasi penelitian yang berkaitan dengan Chatbot.

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan analisis terhadap seluruh spesifikasi sistem yang mencakup analisis prosedur yang sedang berjalan, pengkodean, kebutuhan non fungsional dan analisis basis data. Selain analisis sistem, bab ini terdapat juga perancangan anatarmuka untuk aplikasi yang akan dibangun sesuai dengan hasil analisis yang telah dibuat.

BAB V. IMPLEMENTASI DAN PENGUJIAN

Merupakan tahapan yang dilakukan dalam penelitian secara garis besar sejak dari tahap persiapan sampai penarikan kesimpulan, metode, dan kaidah yang diterapkan dalam penelitian.

BAB VI. HASIL DAN PEMBAHASAN

Mendeskripsikan hasil pembahasan dari penelitian yang telah dilakukan dan telah diuji sesuai dengan teori yang diharapkan.

BAB VII. KESIMPULAN DAN SARAN

Berisi kesimpulan dan saran yang sudah diperoleh dari hasil penulisanskripsi.

**BAB II. LANDASAN TEORI**

* 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

|  |  |  |
| --- | --- | --- |
| Nama Peneliti | Judul Penelitian | Hasil Penelitian |
| Suryani, Dhebys & Larasati Amalia, Eka, 2017. | Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML | Aplikasi Chatbot dapat memberikan informasi kepada wisatawan yang ingin berwisata di wilayah Jawa Timur |
| Satria Paliwabet, I Nyoman & Gede Darma Putra, I Ketut, 2017. | Pencarian Informasi Wisata Daerah Bali Menggunakan Teknologi Chatbot | Sistem chatbot dengan menggunakan metode Fulltext Search Boolean Mode dari MySQL dapat diterapkan dengan baik |
| Suryani, Dhebys & Aulia, Indinabilah, 2018 | Penerapan Metode TF-IDF dan N-Gram pada Pengembangan Aplikasi Chatbot Berbasis LINE untuk Layanan Publik Kesehatan Kota Malang | Question-Answering dalam bentuk chatbot menggunakan N-Gram, TF-IDF dan Cosine Similarity dapat berkomunikasi dan menyampaikan informasi. |
| Astiningrum, Mungki & Shoburu Rohmah, Maya, 2018. | Implementasi NLP dengan Konversi Kata pada Sistem Chatbot Konsultasi Laktasi. | Diterapkannya metode Levenshtein Distance dan TF-IDF dan Cosine Similarity membuat aplikasi chatbot layak digunakan untuk customer service pusat laktasi |
| Melita, Ria & Dirjam, Taslimun, 2018. | 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 : Syarah Umdatil Ahkam) | Metode TF-IDF dan Cosine Similarity berhasil diterapkan dengan memberian hasil berupa output dokumen, yaitu syarah hadits sesuai dengan query yang di input-kan |
| Kavitha B. R. dan Dr, Cethana R, Murthy, 2019 | Chatbot for healtcare system using Artificial Intelligence | Kombinasi antara TF-IDF dan Cosine Similarity memberikan hasil respon chatbot yang sesuai. |
| Natadian Astuti, Rani & Fatchan, Muhammad, 2019. | Perancangan Aplikasi Chatbot untuk Industri Komersial 4.0 | Adanya aplikasi chatbot membuat peran dari customer service menjadi lebih efektif karena dapat melayani pertanyaan dari customer selama 24 jam. |
| Tirtana, A., Zulkarnain, A., Dwi Listio, Y., 2019. | Pembuatan Sistem Pencarian Pekerjaan Menggunakan TF-IDF | Penerapan metode TF-IDF memberikan hasil pencarian yang lebih relevan daripada pencarian tanpa pembobotan. |
| Riyani, Ade & Burhanuddin Aulia, 2019 | Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen | Algoritma Cosine Similarity dan pembobotan TF-IDF telah berhasil mendeteksi kemiripan pada suatu dokumen |

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 healtcare 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*.

* 1. **Chatbot**

Program chatbot pertama ditulis oleh Joseph Weizembaum, 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 Artificiall Intelligence.

Chatbot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunanya 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.

* 1. **XAMPP**

*XAMPP* adalah pengembangan *PHP* di lingkungan paling populer. *XAMPP* merupakan distribusi *Apache* yang benar-benar gratis dan mudahdipasang 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 distribusiinstalasi yang mudah bagi para pengembang untuk memasuki dunia *Apache*. Agar nyaman bagi pengembang, *XAMPP* dikonfigurasikan dengan semua fitur dihidupkan. Dalam hal penggunaann 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.

* 1. **MySQL**

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

MySQL adalah 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* yangberjalan 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.

* 1. **PHP**

*PHP* (*Hypertext Preprocessor*) adalah bahasa pemrograman yang dapatdigunakan 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 yangada saat ini, seperti : *Apache*, *IIS*, *nginx*, dan *lighttpd*. Bahasa yang digunakan dalam CodeIgnitere menggunakan PHP dimana bahasa pemrograman PHP digunakan sebagai bahasa pemrograman umum.

* 1. **JavaScript**

*JavaScript* adalah bahasa *script* berdasar pada objek yangmemperbolehkan 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 itumempunyai 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.

* 1. **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 kemunculansuatu term di dokumen

(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.

*IDFj* = log( *D* / dfj ) (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:

 (2.7.3)

* 1. ***Cosine Similarity***

*Cosine similarity* merupakan rumus yang digunakan Untuk menghitungkesamaan atau *similarity* dengan menentukan sudut antara vektor dokumen dengan vektor *query* dalam dimensi *V* pada bidang *Euclidean*. Hasil dari *cosine* *similarity* memiliki nilai antara 0 sampai dengan 1. Nilai 0 merupakan nilaiyangdidapat apabila dokumen tidak berhubungan dengan *query*, sedangan 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 dimasukan ke dalam rumus *Cosine Similarity*.

(2.7.4)

Ket: = Tingkat kesamaan dokumen dengan query tertentu

= Term ke-i dalam vektor untuk dokumen ke-j

= Term ke-i dalam vektor untuk query ke-k

= Jumlah term yang unik dalam data set

* 1. ***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* *retrival*) berkaitan dengan kemampuan menemukan kembali informasi yang sudahtersimpan (Pendit 2008).

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

*Precision*= Jumlah dokumen relevan yang ditemukan   x100%

                   Jumlah semua dokumen yang ditemukan

Sedangkan nilai *relatif* *Recall* dihitung dengan rumus :

*Recall*     = Jumlah dokumen relevan yang ditemukan  x100%

             Jumlah semua dokumen relevan di dalam koleksi

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.

**BAB III. METODOLOGI PENELITIAN**

* 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.

* 1. **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.

* 1. **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.

* 1. **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

* 1. **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.

* 1. **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.*

* 1. **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 sesuaai dengan desain sistem yang dibuat.

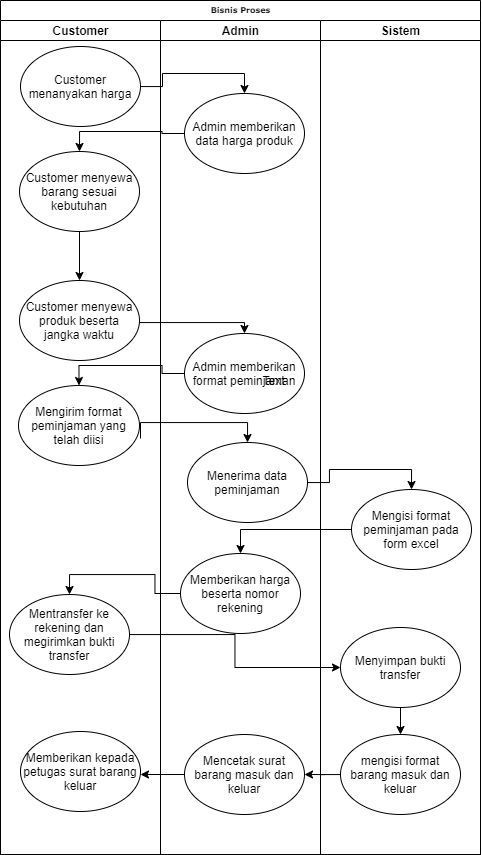
* 1. **Verification**

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.

* 1. **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.

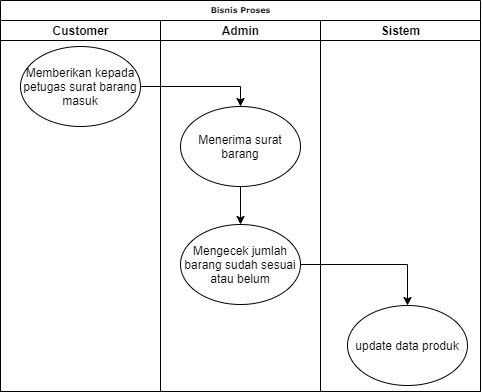
**3.4.2 Bisnis Proses**



Gambar 3.2 Bisnis Proses Peminjaman

* + - * 1. **Deskripsi Bisnis Proses**

Customer menguhubungi 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 yg 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.



Gambar 3.3 Bisnis Proses Pengembalian

* + - * 1. **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.

* 1. **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.

1. *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.

1. *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.

1. *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.

* 1. **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 sewa main 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 |
| Minggu | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 6 |
| Hari | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 6 |
| Jumat | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 6 |
| Libur | 1 | 0 | 1 | 1 | 1 | 0 | 4 | 1,5 |

Tabel 3.3 Perhitungan TF-IDF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dokumen** | **Q** | **d1** | **d2** | **d3** | **d4** | **d5** |
| Harga | 0,1761 | 0 | 0,1761 | 0,1761 | 0,1761 | 0 |
| Sewa | 0 | 0 | 0 | 0 | 0 | 0 |
| Scaffolding | 0,7782 | 0 | 0 | 0 | 0 | 0 |
| Buka | 0 | 0,7782 | 0 | 0 | 0 | 0 |
| Pukul | 0 | 0,7782 | 0 | 0 | 0 | 0 |
| Pagi | 0 | 0,7782 | 0 | 0 | 0 | 0 |
| Main frame | 0 | 0 | 0,3010 | 0,3010 | 0,3010 | 0 |
| Untuk | 0 | 0 | 0,7782 | 0 | 0 | 0 |
| Bulan | 0 | 0 | 0 | 0,3010 | 0,3010 | 0,3010 |
| Minggu | 0 | 0 | 0 | 0 | 0 | 0,7782 |
| Hari | 0 | 0 | 0 | 0 | 0 | 0,7782 |
| Jumat | 0 | 0 | 0 | 0 | 0 | 0,7782 |
| Libur | 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 3.4 Perkalian Skalar

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dokumen** | **d1** | **d2** | **d3** | **d4** | **d5** |
| Harga | 0 | 0,031008 | 0,031008 | 0,031008 | 0 |
| Sewa | 0,006270 | 0,006270 | 0,006270 | 0,006270 | 0 |
| Scaffolding | 0 | 0 | 0 | 0 | 0 |
| Buka | 0 | 0 | 0 | 0 | 0 |
| Pukul | 0 | 0 | 0 | 0 | 0 |
| Pagi | 0 | 0 | 0 | 0 | 0 |
| Main frame | 0 | 0 | 0 | 0 | 0 |
| Untuk | 0 | 0 | 0 | 0 | 0 |
| Bulan | 0 | 0 | 0 | 0 | 0 |
| Minggu | 0 | 0 | 0 | 0 | 0 |
| Hari | 0 | 0 | 0 | 0 | 0 |
| Jumat | 0 | 0 | 0 | 0 | 0 |
| Libur | 0,006270 | 0,037278 | 0,037278 | 0,037278 | 0 |
| **Jumlah** | 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 3.5 Perkalian Vektor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dokumen** | **Q** | **d1** | **d2** | **d3** | **d4** | **d5** |
| Harga | 0,031008 | 0 | 0,031008 | 0,031008 | 0,031008 | 0 |
| Sewa | 0,00627 | 0,00627 | 0,00627 | 0,00627 | 0,00627 | 0 |
| Scaffolding | 0,605519 | 0 | 0 | 0 | 0 | 0 |
| Buka | 0 | 0,605519 | 0 | 0 | 0 | 0 |
| Pukul | 0 | 0,605519 | 0 | 0 | 0 | 0 |
| Pagi | 0 | 0,605519 | 0 | 0 | 0 | 0 |
| Main frame | 0 | 0 | 0,090619 | 0,090619 | 0,090619 | 0 |
| Untuk | 0 | 0 | 0,605519 | 0 | 0 | 0 |
| Bulan | 0 | 0 | 0 | 0,090619 | 0,090619 | 0,090619 |
| Minggu | 0 | 0 | 0 | 0 | 0 | 0,605519 |
| Hari | 0 | 0 | 0 | 0 | 0 | 0,605519 |
| Jumat | 0 | 0 | 0 | 0 | 0 | 0,605519 |
| Libur | 0,642797 | 1,822828 | 0,733416 | 0,218516 | 0,218516 | 1,907177 |
| **Jumlah** | 0,031008 | 0 | 0,031008 | 0,031008 | 0,031008 | 0 |
| **Jumlah Akar** | 0,006270 | 0,006270 | 0,006270 | 0,006270 | 0,006270 | 0 |

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

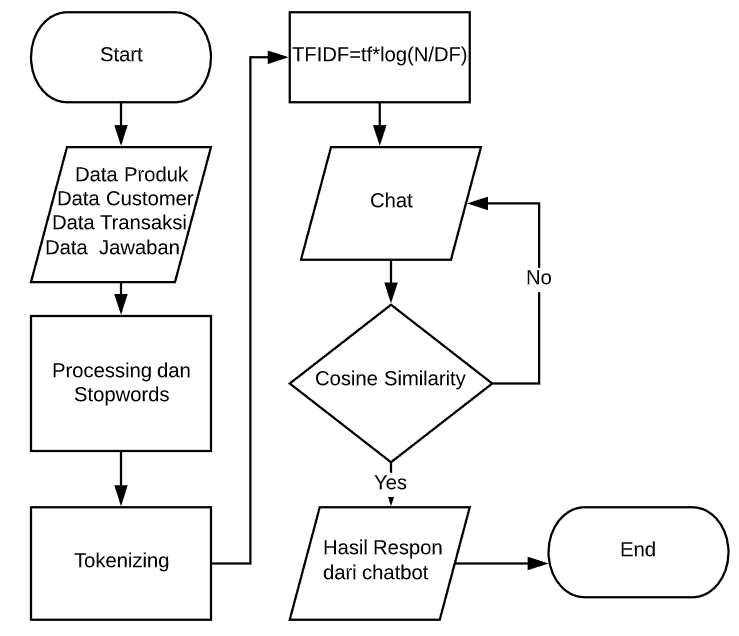
|  |  |  |  |
| --- | --- | --- | --- |
| **Kode** | **Dokumen** | **Nilai Similarity** | **Persentase** |
| **d1** | Penyewaan buka pada pukul 8 pagi | 0,005792 | 0,58% |
| **d2** | Harga sewa main frame 1,9 Rp55.000 untuk sebulan | 0,032241 | 3,22% |
| **d3** | Harga sewa main frame 1,9 Rp45.000 untuk 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.

**BAB IV. ANALISIS DAN PERANCANGAN SISTEM**

* 1. **Gambaran Umum Sistem**

Sistem *chatbot* berbasis *web* berdasarkan data barang dari CV Penyewaan digunakan untuk memberikan respon terhadap pertanyaan *customer* yang diajukan kepada sistem *chatbot*. Sistem ini menggunakan algoritma TF-IDF dan *Cosine Similarity* untuk mencari jawaban terbaik yang akan diberikan oleh sistem berdasarkan *input* dari *customer*. Algoritma TF-IDF digunakan untuk memberikan pembobotan kata pada setiap data jawaban yang ada di sistem. Kemudian dilanjutkan dengan metode *Cosine Similarity* untuk menghitung similaritas antara *input* dari *customer* dengan data jawaban yang ada pada sistem. Sistem *chatbot* ini menggunakan bahasa pemrograman PHP dan *MySQL* sebagai *database*-nya. Dengan adanya sistem ini, diharapkan *customer* tidak akan merasa kesulitan dalam mencari informasi seputar layanan dan produk yang ditawarkan oleh CV Penyewaan. Adapun alur aplikasi *chatbot* ini dijelaskan pada gambar 4.1



Gambar 4.1 Flowchart Alur Proses Chatbot

* 1. **Analisa Permasalahan**

Permasalahan yang terjadi adalah *customer service* memiliki waktu kerja dan seringkali tidak bisa melayani pertanyaan dari *customer* selama 24 jam nonsto. Hal ini menyebabkan *customer* harus menunggu lama untuk mendapatkan jawaban dari informasi yang mereka tanyakan kepada *customer service* sesuai dengan jam bertanya.

* 1. **Analisa Kebutuhan Non Fungsional**

Terdapat beberapa analisa kebutuhan non fungsional meliputi kebutuhan perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut adalah rincian kebutuhan sistem yang akan dibuat:

* + 1. Perangkat Lunak (*Software*)

Spesifikasi umum perangkat lunak yang digunakan dalam pembuatan sistem tercantum dalam tabel 4.1.

Tabel 4.1 Spesifikasi Minimum Perangkat Lunak

|  |  |
| --- | --- |
| ***Software*** | **Keterangan** |
| Windows 10 | Sistem operasi yang digunakan untuk menjalankan program *chatbot*. |
| Visual Studio Code | Aplikasi *text editor* yang digunakan untuk menulis kode program. |
| Apache HTTP Server | Sebagai *web server* dari aplikasi *chatbot* yang akan dibuat. |
| MySQL | Sebagai *Database Management System* (DBMS) untuk menyimpan data. |
| PHP | Sebagai bahasa pemrograman utama yang digunakan untuk membuat aplikasi *chatbot*. |

* + 1. Perangkat Keras (*Hardware*)

Spesifikasi umum perangkat keras yang digunakan dalam pembuatan sistem dapat dilihat pada tabel 4.2.

Tabel 4.2 Spesifikasi Minimum Perangkat Keras

|  |  |
| --- | --- |
| ***Hardware*** | **Keterangan** |
| *Processor* | AMD Dual-Core A4-9120 APU 2.2 GHz |
| RAM | 4 GB |
| *Hard disk* | 500 GB |
| *Monitor* | Disesuaikan |
| Perangkat *Input* | *Keyboard* dan *Mouse* |

* 1. **Analisa Kebutuhan Fungsional**

Kebutuhan fungsional adal ah kebutuhan yang berisi proses yang dilakukan oleh sistem. Dalam hal ini *customer* dapat menggunakan layanan yang ada pada sistem. Layanan yang dimaksud adalah sebagai berikut:

1. Sistem dapat *load dataset* dari *database.*
2. Sistem dapat menerima *input* pertanyaan dari *customer*.
3. Sistem dapat melakukan perhitungan TF-IDF.
4. Sistem dapat melakukan perhitungan *Cosine Similarity*.
5. Sistem dapat memberikan respon jawaban berdasarakan apa yang telah di-*input*-kan *customer*.
   1. **Analisis Data**

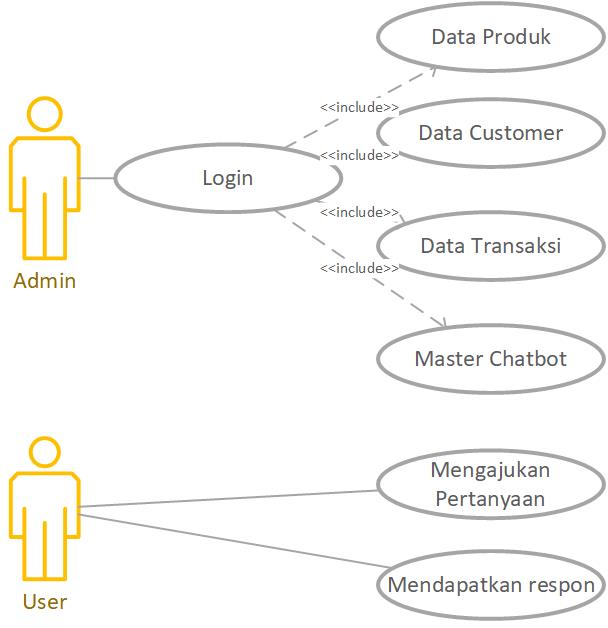
Analisis data adalah proses pengolahan dan penghitungan dari seluruh data yang sudah diperoleh dari penelitian yang telah dilakukan secara sistematis. Analisis data yang digunakan pada penelitian ini untuk menguji variabel yang ada adalah penghitungan bobot tiap kata dan menghitung kemiripan kata untuk mendapatkan jawaban yang diharapkan sesuai dengan pertanyaan yang diajukan oleh customer.

Namun sebelum data diuji perlu dahulu dilakukan uji perhitungan manual guna melihat hasil apakah akan sesuai dengan jawaban yang nantinya diharapkan. Jika hasil yang diperlihatkan kemiripan tinggi maka uji yang dilakukan sesuai dengan harapan jawaban yang diinginkan.

* 1. **Perancangan Sistem**

4.5.1 *Use Case*

*Use Case* merupakan sebuah teknik yang digunakan dalam pengembangan sebuah *software* atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan. *Use Case* menjelaskan interaksi yang terjadi antara aktor, yaitu inisiator dari sistem itu sendiri, dengan sistem yang ada. Sebuah *Use Case* direpresentasikan dengan langkah yang sederhana. *Use Case* pada sistem ini dapat dilihat pada gambar 4.2.



Gambar 4.2 Use Case

4.5.1.1 Deskripsi *Use Case*

Deskripsi *Use Case* bertujuan untuk memberikan gambaran umum tentang fungsional suatu proses yang di dalamnya melibatkan sebuah sistem.

Tabel 4.3 Deskripsi Use Case

|  |  |  |
| --- | --- | --- |
| **No.** | ***Use Case*** | **Deskripsi** |
| 1 | Melakukan login | Melakukan login untuk dapat mengakses sistem |
| 2 | Mengelola data produk | Menambah, mengubah dan menghapus data produk |
| 3 | Mengelola data *customer* | Menambah, mengubah dan menghapus data *customer* |
| 4 | Mengelola data transaksi | Menambah, mengubah dan menghapus data transaksi |
| 5 | Mengelola data *chatbot* | Menambah, mengubah, menghapus dan melakukan perhitungan data *chatbot* |
| 6 | Mengajukan pertanyaan | Melakukan *input* pertanyaan pada sistem mengenai informasi yang diharapkan *customer* |
| 7 | Mendapatkan respon | Mendapatkan respon sesuai dengan hasil yang di-*input* oleh *customer* |

4.5.1.2 Skenario *Use Case*

Skenario *Use Case* berisi uraian nama, aktor yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi tentang *use case,* pra-kondisi yang harus dipenuhi dan pasca-kondisi yang diharapkan setelah berjalannya fungsional *use case*.

1. Melakukan *login*

Tabel 4.4 Skenario Use Case Melakukan Login

|  |  |
| --- | --- |
| **Nama Use Case** | Melakukan *login* |
| **Aktor** | Admin |
| **Tujuan** | Admin bisa masuk ke dalam sistem |
| **Kondisi Awal** | - |
| **Skenario** | 1. Admin mengisi username dan password 2. Jika sesuai, tampil halaman *dashboard* sistem 3. Jika tidak sesuai, tampil notifikasi username dan password tidak valid |
| **Invariant 1** | - |

1. Mengelola data produk

Tabel 4.5 Skenario Use Case Mengelola Data Produk

|  |  |
| --- | --- |
| **Nama Use Case** | Mengelola data produk |
| **Aktor** | Admin |
| **Tujuan** | Admin dapat menambah, mengubah dan menghapus data produk |
| **Kondisi Awal** | Admin sudah login |
| **Skenario** | 1. Admin melihat data produk 2. Admin menambah data produk 3. Admin menyimpan data produk |
| **Invariant 1** | Admin mengubah data produk |
| **Invariant 2** | Admin menghapus data produk |

1. Mengelola data *customer*

Tabel 4.6 Skenario Use Case Mengelola Data Customer

|  |  |
| --- | --- |
| **Nama Use Case** | Mengelola data *customer* |
| **Aktor** | Admin |
| **Tujuan** | Admin dapat menambah, mengubah dan menghapus data *customer* |
| **Kondisi Awal** | Admin sudah login |
| **Skenario** | 1. Admin melihat data *customer* 2. Admin menambah data *customer* 3. Admin menyimpan data *customer* |
| **Invariant 1** | Admin mengubah data *customer* |
| **Invariant 2** | Admin menghapus data *customer* |

1. Mengelola data transaksi

Tabel 4.7 Skenario Use Case Mengelola Data Transaksi

|  |  |
| --- | --- |
| **Nama Use Case** | Mengelola data transaksi |
| **Aktor** | Admin |
| **Tujuan** | Admin dapat melihat, menambah dan menghapus data transaksi penyewaan |
| **Kondisi Awal** | Admin sudah login |
| **Skenario** | 1. Admin melihat data transaksi 2. Admin mengisi form transaksi 3. Admin menyimpan data transaksi |
| **Invariant 1** | Admin menambah barang pada data transaksi penyewaan |
| **Invariant 2** | Admin merubah data transaksi penyewaan |
| **Invariant 3** | Admin menghapus data transaksi penyewaan |

1. Mengelola data *chatbot*

Tabel 4.8 Skenario Use Case Mengelola Data Chatbot

|  |  |
| --- | --- |
| **Nama Use Case** | Mengelola data *chatbot* |
| **Aktor** | Admin |
| **Tujuan** | Admin dapat menambah, mengubah, menghapus dan melakukan perhitungan data *chatbot* |
| **Kondisi Awal** | Admin sudah login |
| **Skenario** | 1. Admin melihat data *chatbot* 2. Admin menambah data *chatbot* 3. Admin menyimpan data *chatbot* 4. Admin melakukan perhitungan TF-IDF pada data *chatbot* 5. Admin melihat hasil perhitungan data *chatbot* |
| **Invariant 1** | Admin mengubah data  *chatbot* |
| **Invariant 2** | Admin menghapus data *chatbot* |

1. Mengajukan pertanyaan pada sistem

Tabel 4.9 Skenario Use Case Mengajukan Pertanyaan pada Sistem

|  |  |
| --- | --- |
| **Nama Use Case** | Mengajukan pertanyaan pada sistem |
| **Aktor** | *User* |
| **Tujuan** | *User* dapat melakukan *input* pertanyaan pada sistem |
| **Kondisi Awal** | - |
| **Skenario** | 1. *User* mengisikan pertanyaan pada form *input chatbot* 2. *User* menekan tombol *submit* |
| **Invariant 1** | - |

1. Mendapatkan respon dari *chatbot*

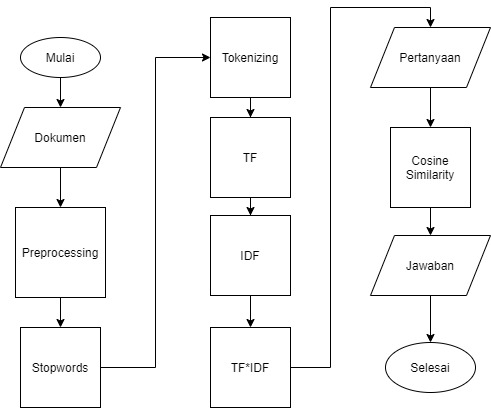
Tabel 4.10 Skenario Use Case Mendapatkan Respon dari Chatbot

|  |  |
| --- | --- |
| **Nama Use Case** | Mendapatkan respon dari *chatbot* |
| **Aktor** | *User* |
| **Tujuan** | *User* dapat menerima respon atas pertanyaan yang telah dimasukan sebelumnya |
| **Kondisi Awal** | *User* telah melakukan *input* pertanyaan |
| **Skenario** | 1. Sistem menerima *input* dari user 2. Sistem melakukan perhitungan *similaritas* terhadap *input* dari *user* 3. Sistem melakukan pencarian respon yang memiki similaritas tertinggi 4. User akan menerima respon dari sistem terkait pertanyaan yang diajukan |
| **Invariant 1** | Sistem akan memunculkan respon ketidaktahuan apabila pertanyaan dari *user* tidak cocok pada jawaban apapun |

4.5.2 *Flowchart*

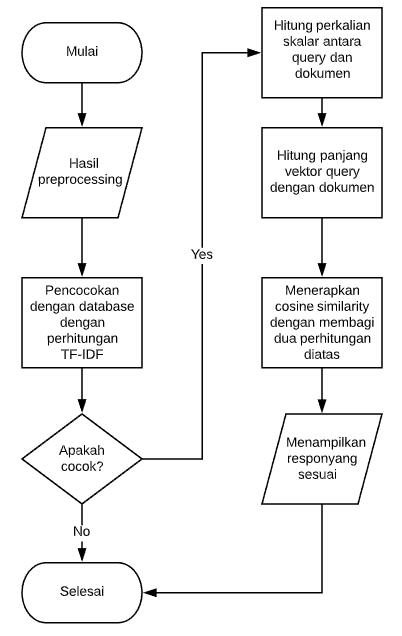
*Flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Dalam sistem ini dijabarkan *flowchart* proses pembobotan TF-IDF dan perhitungan *Cosine Similarity*.

4.5.2.1 *Flowchart* Pembobotan TF-IDF



Gambar 4.3 Flowchart Pembobotan TF-IDF

4.5.2.2 *Flowchart* Perhitungan *Cosine Similarity*



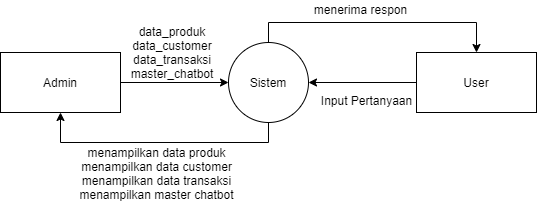
Gambar 4.4 Flowchart Perhitungan Cosine Similarity

4.5.3 *Data Flow Diagram* (DFD)

*Data Flow Diagram* (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data pada suatu sistem atau menjelaskan proses kerja suatu sistem, yang penggunaanya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas.

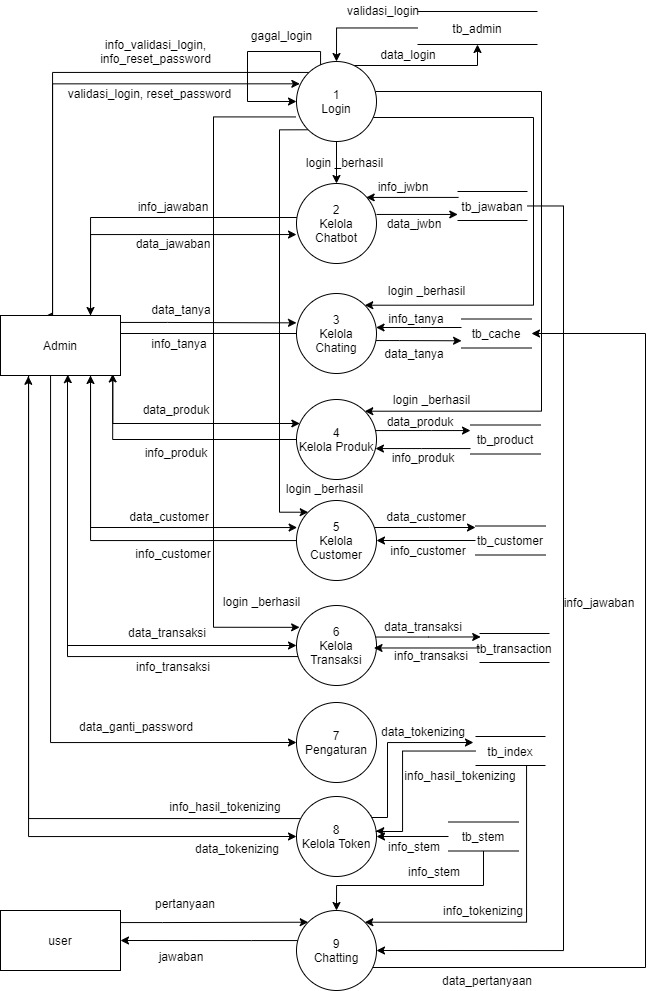
4.5.3.1 *Context Diagram*

Context Diagram merupakan tingkatan tertinggi dalam aliran data diagram dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data utama menuju ke sistem. *Context Diagram* pada sistem ini ditunjukkan dalam gambar 4.5

****

Gambar 4.5 Context Diagram

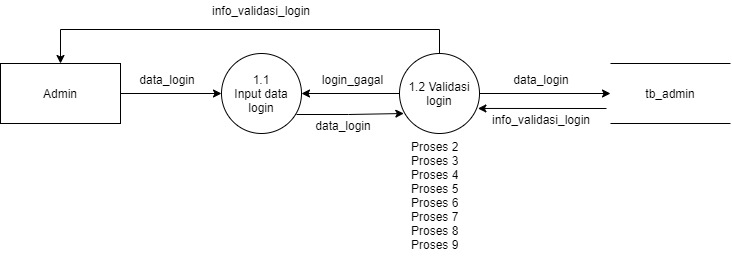
4.5.3.2 DFD Level 1



Gambar 4.6 DFD Level 1

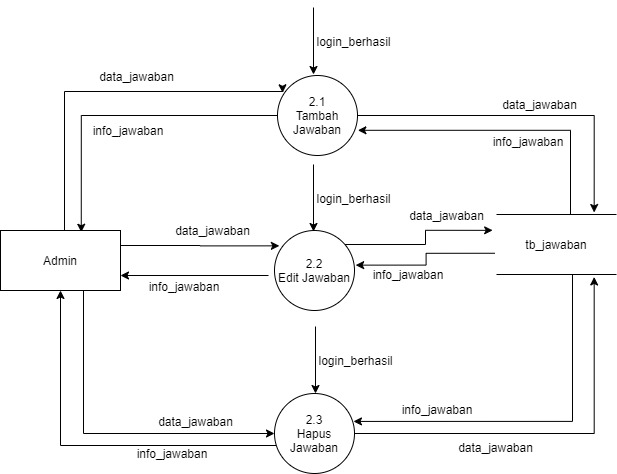
4.5.3.3 DFD Level 2

1. DFD Level 2 Proses 1



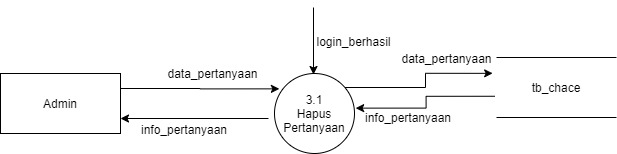
Gambar 4.7 DFD Level 2 Proses 1

* + - 1. DFD Level 2 Proses 2



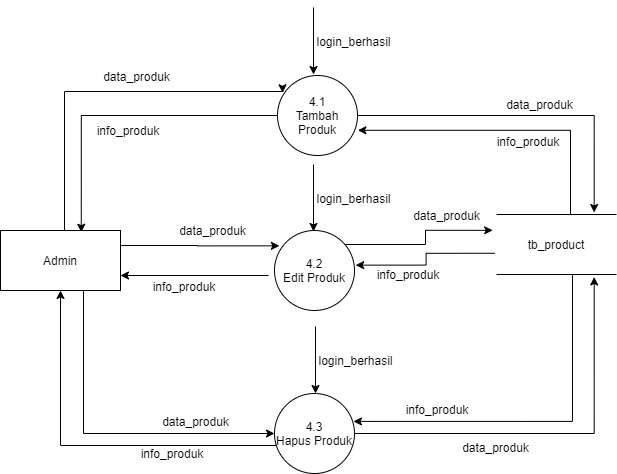
Gambar 4.8 DFD Level 2 Proses 2

* + - 1. DFD Level 2 Proses 3



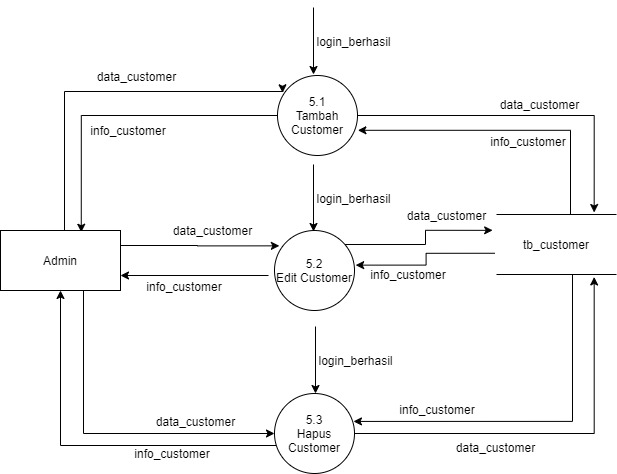
Gambar 4.9 DFD Level 2 Proses 3

* + - 1. DFD Level 2 Proses 4



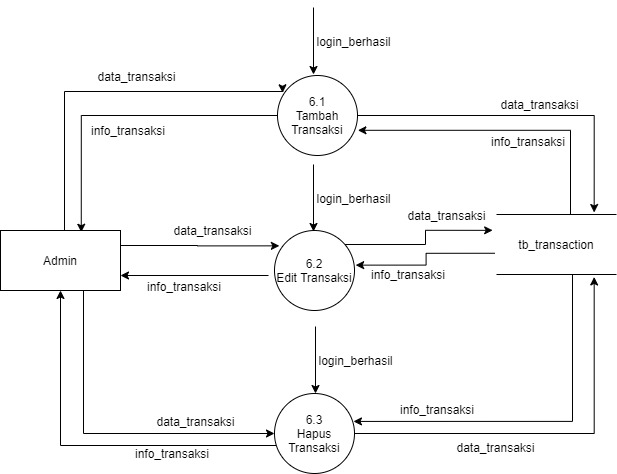
Gambar 4.10 DFD Level 2 Proses 4

* + - 1. DFD Level 2 Proses 5



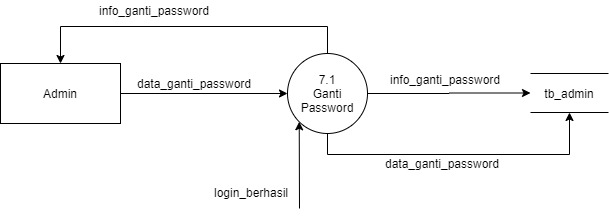
Gambar 4.11 DFD Level 2 Proses 5

* + - 1. DFD Level 2 Proses 6



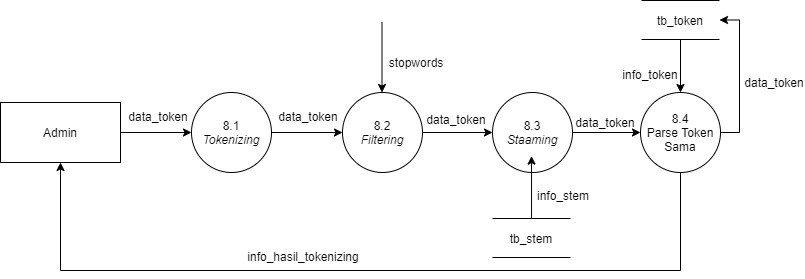
Gambar 4.12 DFD Level 2 Proses 6

* + - 1. DFD Level 2 Proses 7



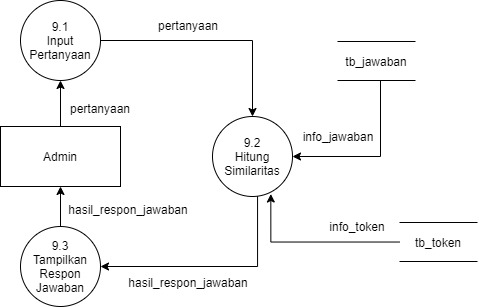
Gambar 4.13 DFD Level 2 Proses 7

* + - 1. DFD Level 2 Proses 8



Gambar 4.14 DFD Level 2 Proses 8

* + - 1. DFD Level 2 Proses 9



Gambar 4.15 DFD Level 2 Proses 9

4.5.4 Spesifikasi Proses

Tabel 4.11 Spesifikasi proses login

|  |  |
| --- | --- |
| No. Proses | 1 |
| Nama | Login |
| Deskripsi | Proses yang digunakan ketika akan mengakses halaman admin *chatbot*. |
| Data Masuk | data\_login |
| Data Keluar | info\_validasi\_login, login\_berhasil, login\_gagal |
| Logika Proses | 1. Jika pengguna sudah login sistem akan menampilkan layar *dashboard*. 2. Jika belum, lanjut ke no. 3. 3. Sistem menampilkan halaman login. 4. Pengguna mengisi data login (*username* dan *password*). 5. Setelah pengguna tekan tombol login, sistem akan memeriksa data login ke *database* tabel admin. 6. Jika data ditemukan dan *username*/*password* benar, maka sistem akan menampilkan layar utama dari halaman admin yang berisi menu-menu untuk mengakses proses-proses 2,3,4,5,6,7,8. 7. Jika data tidak ditemukan atau *username*/*password* salah, maka sistem akan menampilkan pesan gagal login dan sistem akan menampilkan halaman login kembali. |

Tabel 4.12 Spesifikasi proses kelola jawaban

|  |  |
| --- | --- |
| No. Proses | 2 |
| Nama | Kelola Jawaban |
| Deskripsi | Proses yang digunakan untuk mengelola jawaban dari pertanyaan yang mungkin akan ditanyakan oleh user. |
| Data Masuk | data\_jawaban, login\_berhasil |
| Data Keluar | info\_jawaban |
| Logika Proses | 1. Dalam halaman kelola jawaban terdapat proses tambah jawaban, edit jawaban, dan hapus jawaban. 2. Tambah jawaban ditekan akan menampilkan form tambah jawaban. 3. Edit jawaban ditekan akan menampilkan form edit jawaban untuk jawaban yang dimaksud. 4. Hapus jawaban ditekan akan menampilkan konfirmasi penghapusan data jawaban. |

Tabel 4.13 Spesifikasi proses kelola pertanyaan

|  |  |
| --- | --- |
| No. Proses | 3 |
| Nama | Kelola Pertanyaan |
| Deskripsi | Proses yang digunakan untuk mengelola pertanyaan dari yang diajukan oleh user. |
| Data Masuk | data\_pertanyaan, login\_berhasil |
| Data Keluar | info\_pertanyaan |
| Logika Proses | 1. Dalam halaman kelola pertanyaan terdapat proses hapus pertanyaan. 2. Hapus jawaban ditekan akan menampilkan konfirmasi penghapusan data jawaban. |

Tabel 4.14 Spesifikasi kelola produk

|  |  |
| --- | --- |
| No. Proses | 4 |
| Nama | Kelola Produk |
| Deskripsi | Proses yang digunakan untuk mengelola produk yang tersedia. |
| Data Masuk | data\_produk, login\_berhasil |
| Data Keluar | info\_produk |
| Logika Proses | 1. Dalam halaman kelola produk terdapat proses tambah produk, edit produk, dan hapus produk. 2. Tambah produk ditekan akan menampilkan form tambah produk. 3. Edit produk ditekan akan menampilkan form edit produk untuk produk yang dimaksud. 4. Hapus produk ditekan akan menampilkan konfirmasi penghapusan data layanan. |

Tabel 4.15 Spesifikasi kelola customer

|  |  |
| --- | --- |
| No. Proses | 5 |
| Nama | Kelola Customer |
| Deskripsi | Proses yang digunakan untuk mengelola customer dari penyewaan. |
| Data Masuk | data\_customer, login\_berhasil |
| Data Keluar | info\_customer |
| Logika Proses | 1. Dalam halaman kelola customer terdapat proses tambah customer, edit customer, dan hapus customer. 2. Tambah customer ditekan akan menampilkan form tambah customer. 3. Edit customer ditekan akan menampilkan form edit customer untuk customer yang dimaksud. 4. Hapus customer ditekan akan menampilkan konfirmasi penghapusan data customer. |

Tabel 4.16 Spesifikasi kelola transaksi

|  |  |
| --- | --- |
| No. Proses | 6 |
| Nama | Kelola Transaksi |
| Deskripsi | Proses yang digunakan untuk mengelola transaksi dari penyewaan. |
| Data Masuk | data\_transaksi, login\_berhasil |
| Data Keluar | info\_transaksi |
| Logika Proses | 1. Dalam halaman kelola transaksi terdapat proses tambah transaksi, edit transaksi, dan hapus transaksi. 2. Tambah transaksi ditekan akan menampilkan form tambah transaksi. 3. Edit transaksi ditekan akan menampilkan form edit transaksi untuk transaksi yang dimaksud. 4. Hapus transaksi ditekan akan menampilkan konfirmasi penghapusan data transaksi. |

Tabel 4.17 Spesifikasi kelola pengaturan

|  |  |
| --- | --- |
| No. Proses | 7 |
| Nama | Kelola Pengaturan |
| Deskripsi | Proses yang digunakan untuk mengelola pengaturan dari penyewaan. |
| Data Masuk | data\_ganti\_password, login\_berhasil |
| Data Keluar | info\_ganti\_password |
| Logika Proses | 1. Dalam halaman kelola pengaturan terdapat proses ganti password. |

Tabel 4.18 Spesifikasi kelola token

|  |  |
| --- | --- |
| No. Proses | 8 |
| Nama | Kelola Token |
| Deskripsi | Proses yang digunakan untuk mengelola token dari penyewaan. |
| Data Masuk | data\_token |
| Data Keluar | info\_hasil\_tokenizing |
| Logika Proses | 1. Dalam halaman kelola token terdapat proses tokenizing, filtering, steaming, parse token sama. 2. Proses *tokenizing* untuk memotong *string input* berdasarkan kata yang tersusun. 3. Proses *filtering* untuk menghapus setiap kata-kata yang tidak penting atau kata-kata yang tidak memiliki arti. 4. Proses steaming untuk mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akar/dasar (*root word*). |

Tabel 4.19 Spesifikasi kelola chatting

|  |  |
| --- | --- |
| No. Proses | 9 |
| Nama | Kelola Chatting |
| Deskripsi | Proses yang digunakan untuk mengelola chatting dari penyewaan. |
| Data Masuk | Pertanyaan |
| Data Keluar | Jawaban |
| Logika Proses | 1. Dalam halaman kelola chatting terdapat proses input pertanyaan, hitung similaritas, dan menampilkan hasil jawaban. 2. Saat customer masuk ke halam chatbot, customer mengetikkan pertanyaan. 3. Sistem menghitung kemiripan kata pertanyaan dengan jawaban yang diinginkan. 4. Sistem menampilkan jawaban dengan kemiripan tinggi. |

* 1. **Perancangan Basisdata**



Gambar 4.16 Entitas Relationship Diagram

Tabel Admin

Nama Tabel : tb\_admin

Deskripsi : Berisi data admin

Tabel 4.20 Tabel Admin

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | Admin\_id | Int(11) | AutoIncrement, PK |
| 2. | Admin\_name | Varchar(128) |  |
| 3. | Admin\_username | Varchar(15) |  |
| 4. | Admin\_password | Varchar(15) |  |

Tabel Cache

Nama Tabel : tb\_cache

Deskripsi : Berisi data cache

Tabel 4.21 Tabel Cache

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | Id | Int(11) | AutoIncrement, PK |
| 2. | Query | Varchar(255) |  |
| 3. | Doc\_id | Int(11) | FK |
| 4. | Nilai | Float |  |

Tabel Customer

Nama Tabel : tb\_customer

Deskripsi : Berisi data customer

Tabel 4.22 Tabel Customer

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | customer\_id | Int(5) | AutoIncrement, PK |
| 2. | customer\_nik | Varchar(20) |  |
| 3. | customer\_name | Varchar(255) |  |
| 4. | customer\_address | Text |  |
| 5. | Customer\_phone | Varchar(20) |  |
| 6. | Customer\_company | Varchar(150) |  |
| 7. | Customer\_project | Varchar(150) |  |

Tabel Detail

Nama Tabel : tb\_detail

Deskripsi : Berisi data detail transaksi

Tabel 4.23 Tabel Detail

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | detail\_id | Int(10) | AutoIncrement, PK |
| 2. | Transaction\_id | int(11) | FK |
| 3. | Product\_id | Varchar(5) | FK |
| 4. | quantity | Int(5) |  |

Tabel Dokumen

Nama Tabel : tb\_dokumen

Deskripsi : Berisi data dokumen

Tabel 4.24 Tabel Dokumen

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | id | Int(5) | AutoIncrement, PK |
| 2. | dokumen | Varchar(255) |  |

Tabel Index

Nama Tabel : tb\_index

Deskripsi : Berisi data index

Tabel 4.25 Tabel Index

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | id | Int(11) | AutoIncrement, PK |
| 2. | Term | Varchar(255) |  |
| 3. | Id\_doc | Int(11) | FK |
| 4. | jumlah | int(11) |  |
| 5. | bobot | Float |  |

Tabel Produk

Nama Tabel : tb\_product

Deskripsi : Berisi data produk

Tabel 4.26 Tabel Produk

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | product\_id | Int(11) | AutoIncrement, PK |
| 2. | product\_name | Varchar(25) |  |
| 3. | product\_owp | int(5) |  |
| 4. | product\_twp | int(5) |  |
| 5. | Product\_omp | Int(5) |  |
| 6. | Product\_ep | Int(5) |  |
| 7. | Product\_stock | Int(3) |  |
| 8. | Product\_weight | Int(5) |  |

Tabel Stem

Nama Tabel : tb\_stem

Deskripsi : Berisi data stem

Tabel 4.27 Tabel Stem

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | id | Int(11) | AutoIncrement, PK |
| 2. | term | Varchar(255) |  |
| 3. | stem | varchar(255) |  |

Tabel Transaksi

Nama Tabel : tb\_transaction

Deskripsi : Berisi data transaksi

Tabel 4.28 Tabel Transaksi

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | id | Int(11) | AutoIncrement, PK |
| 2. | Id\_cust | Int(5) | FK |
| 3. | fdate | Date |  |
| 4. | ldate | Date |  |
| 5. | loanstatus | Varchar(20) |  |
| 6. | paidstatus | Varchar(20) |  |

Tabel Vektor

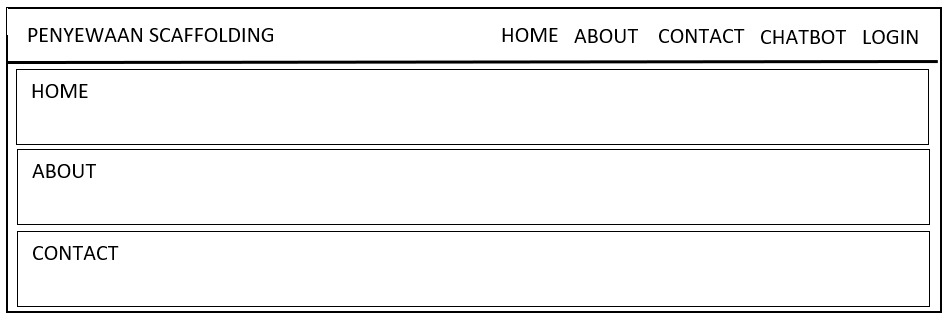
Nama Tabel : tb\_vektor

Deskripsi : Berisi data vektor

Tabel 4.29 Tabel Vektor

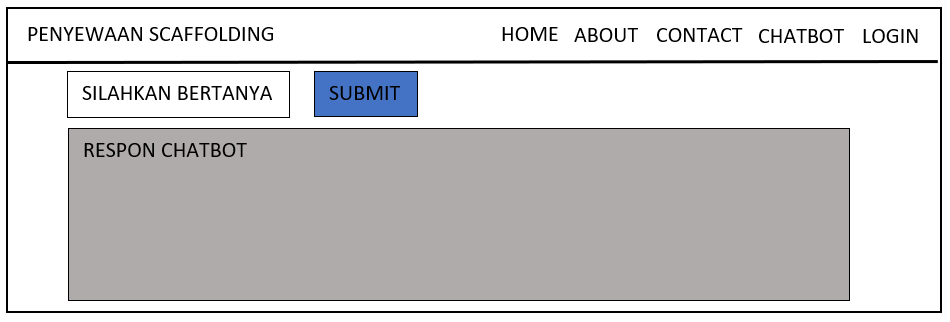
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Nama Field | Tipe Data | Keterangan |
| 1. | Doc\_id | Int(11) | AutoIncrement, PK |
| 2. | panjang | Float |  |

* 1. **Perancangan Antar Muka Pengguna**
  2. Halaman awal



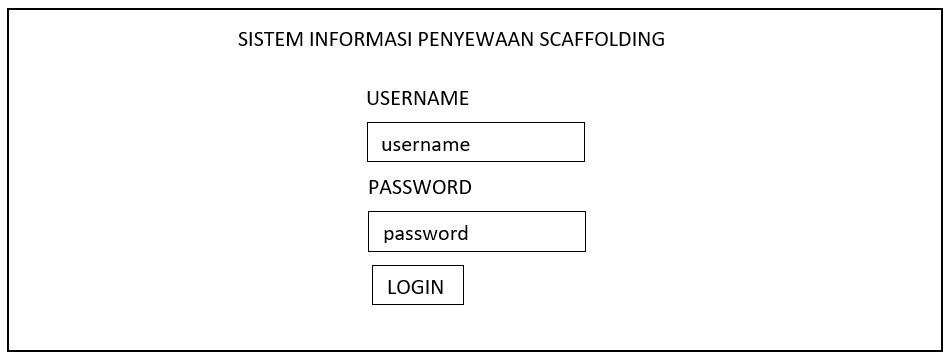
Gambar 4.17 Mockup Halaman Awal Sistem

* 1. Halaman Chat



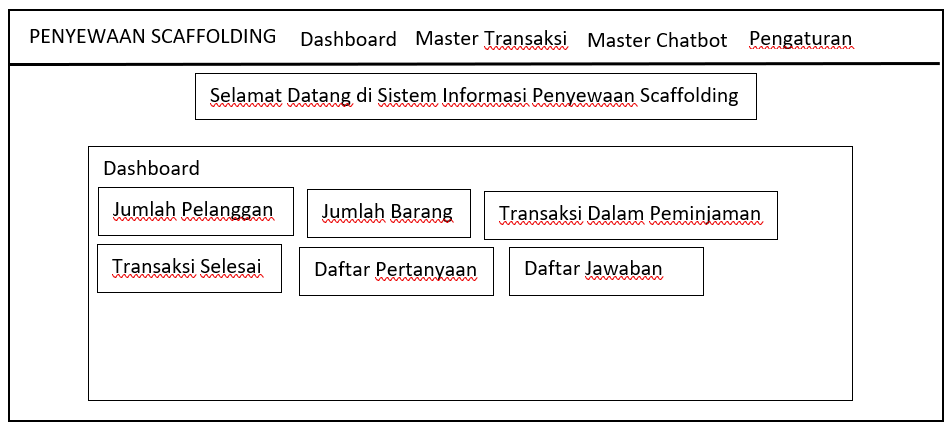
Gambar 4.18 Mockup Halaman Chat

* 1. Halaman Login



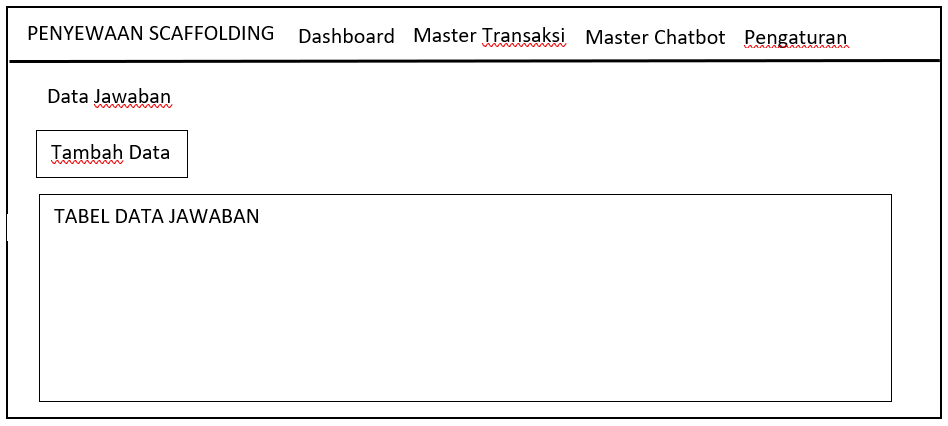
Gambar 4.19 Mockup Halaman Login

* 1. Halaman Admin



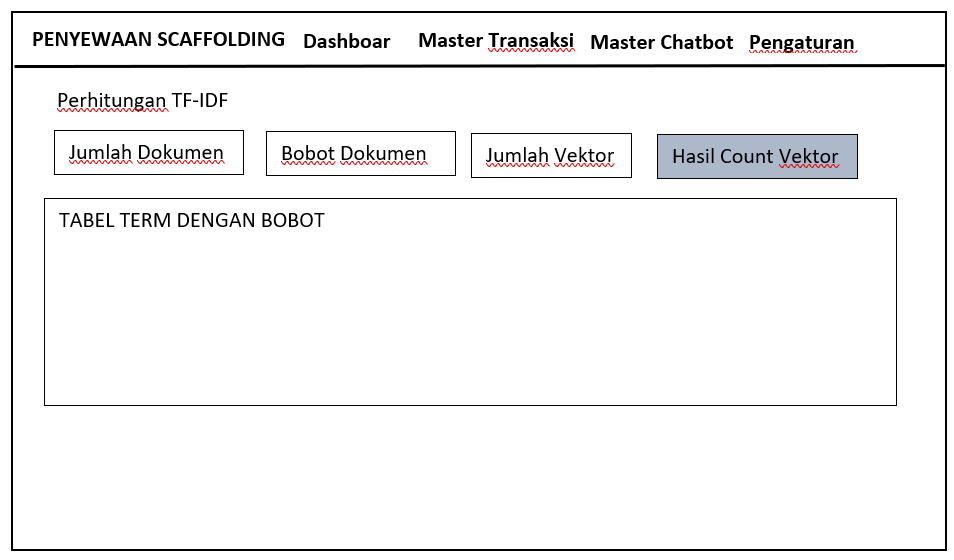
Gambar 4.20 Mockup Halaman Utama Admin

* 1. Halaman Master Chatbot



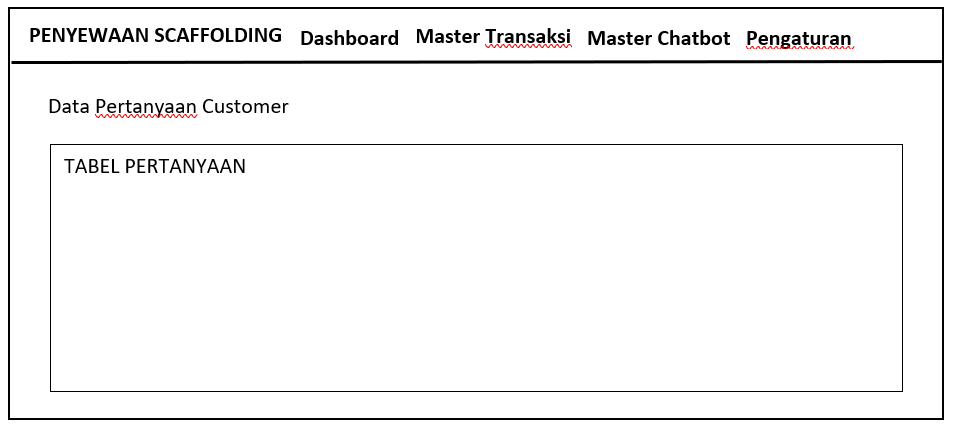
Gambar 4.21 Mockup Halaman Data Jawaban

* 1. Halaman Penghitungan TF-IDF



Gambar 4.22 Mockup Halaman Perhitungan TF-IDF

* 1. Halaman Pertanyaan

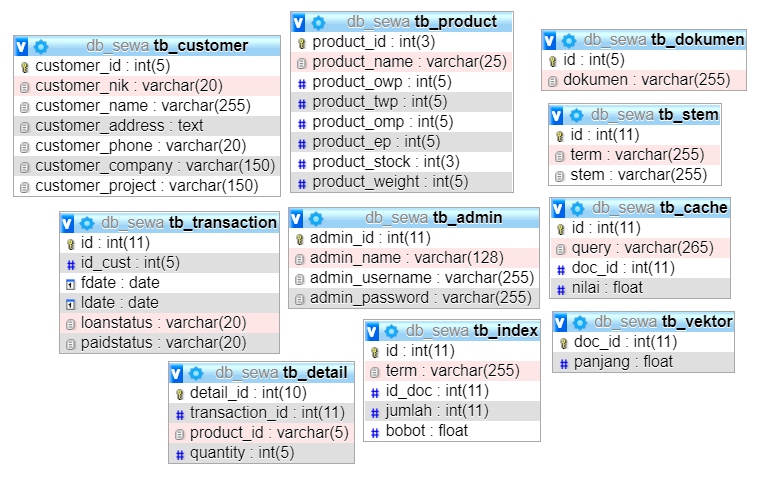


Gambar 4.23 Mockup Halaman Data Pertanyaan

**BAB V. IMPLEMENTASI DAN PENGUJIAN**

* 1. **Implementasi Basisdata**

Implementasi Basisdata untuk sistem yang dibangun diberi nama db\_sewa. Basisdata yang dibuat memiliki beberapa tabel yang sebelumnya telah dirancangkan. Berikut merupakan gambar dari struktur database.

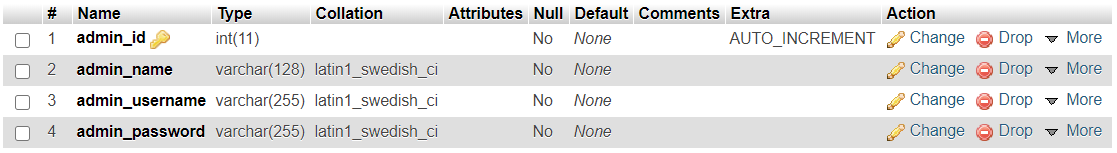


Gambar 5.1 Struktur Database

Dibawah ini akan menjelaskan pada masing-masing tabel

### 5.1.1 Tabel *Admin*

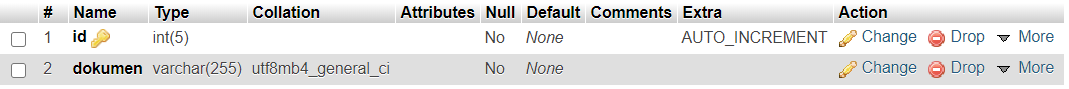
Tabel *tb\_admin* yang sebelumnya telah dirancang pada tabel 4.20 diimplementasikan pada gambar 5.2.



Gambar 5.2 Tabel Admin

### Tabel Dokumen

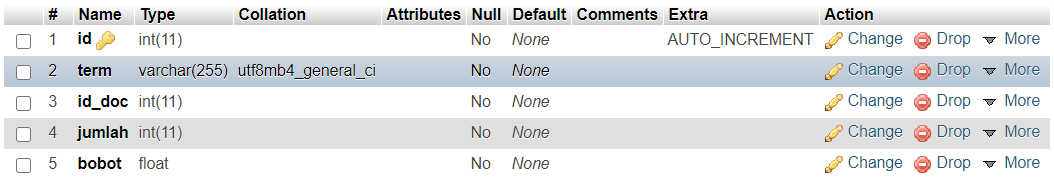
Tabel *tb\_dokumen* yang sebelumnya telah dirancang pada tabel 4.24 diimplementasikan pada gambar 5.3.



Gambar 5.3 Tabel Dokumen

### Tabel *Index*

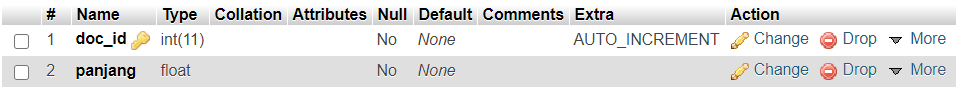
Tabel *tb\_index* yang sebelumnya telah dirancang pada tabel 4.25 diimplementasikan pada gambar 5.4.



Gambar 5.4 Tabel Index

### Tabel Vektor

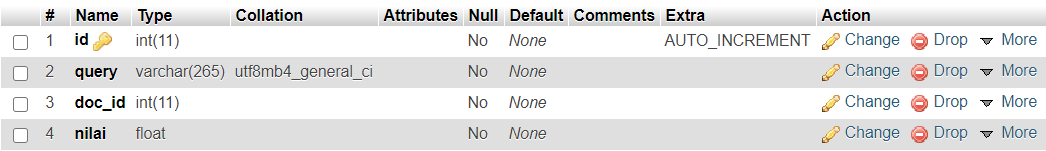
Tabel *tb\_vektor* yang sebelumnya telah dirancang pada tabel 4.29 diimplementasikan pada gambar 5.5.



Gambar 5.5 Tabel Vektor

### Tabel *Cache*

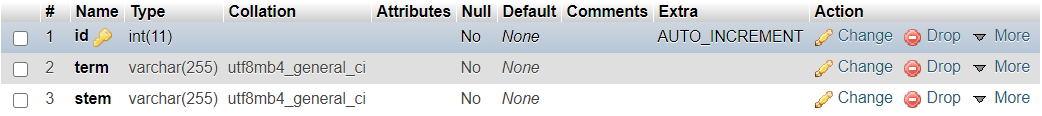
Tabel *tb\_cache* yang sebelumnya telah dirancang pada tabel 4.21 diimplementasikan pada gambar 5.6



Gambar 5.6 Tabel Cache

### Tabel *Stem*

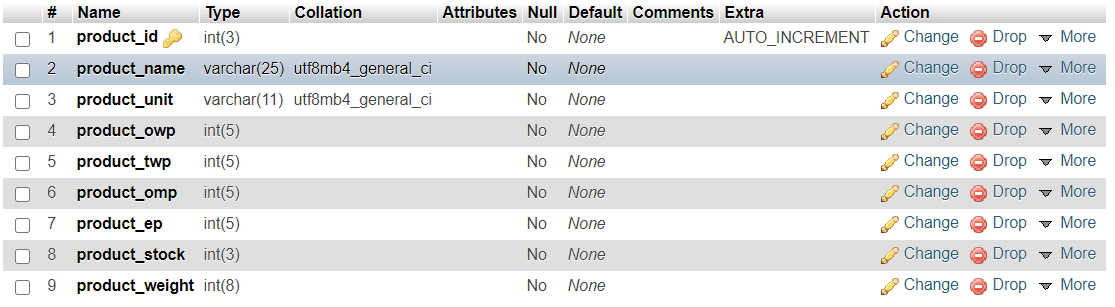
Tabel *tb\_stem* yang sebelumnya telah dirancang pada tabel 4.27 diimplementasikan pada gambar 5.7.



Gambar 5.7 Tabel Stem

### Tabel *Product*

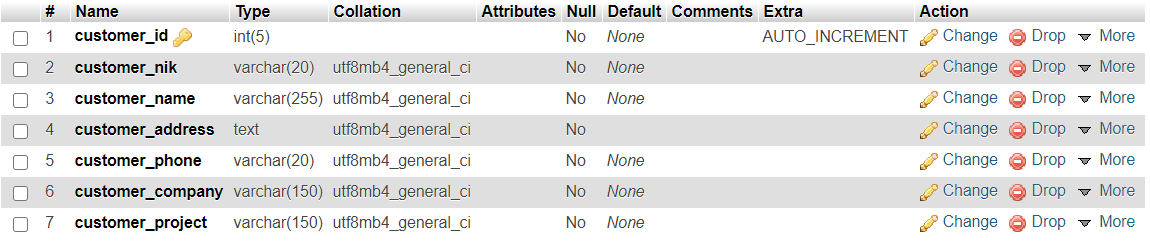
Tabel *tb\_product* yang sebelumnya telah dirancang pada tabel 4.26 diimplementasikan pada gambar 5.8.



Gambar 5.8 Tabel Product

### Tabel *Customer*

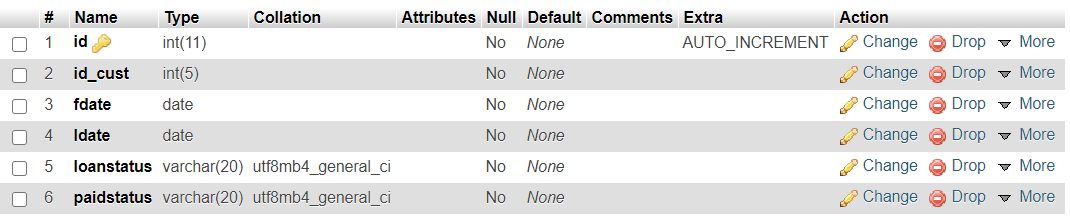
Tabel *tb\_customer* yang sebelumnya telah dirancang pada tabel 4.22 diimplementasikan pada gambar 5.9.



Gambar 5.9 Tabel Customer

### Tabel Transaksi

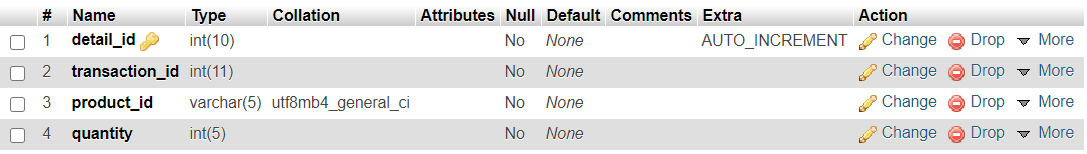
Tabel *tb\_transaksi* yang sebelumnya telah dirancang pada tabel 4.28 diimplementasikan pada gambar 5.10.



Gambar 5.10 Tabel Transaksi

### Tabel Detail Transaksi

Tabel *tb\_detail\_transaksi* yang sebelumnya telah dirancang pada tabel 4.23 diimplementasikan pada gambar 5.10.



Gambar 5.11 Tabel Detail Transaksi

* 1. **Implementasi Sistem**

Implementasi proses sistem yang berdasarkan pada perancangan terdiri dari beberapa potongan kode program sebagai berikut.

### Menampilkan Data *Master* Jawaban

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk menampilkan data *master* jawaban *chatbot*. Potongan kode program dapat dilihat pada gambar 5.12.

|  |
| --- |
| <?php  include '../koneksi.php';  $data = mysqli\_query($koneksi, "SELECT \* FROM tb\_dokumen");  $no = 1;  while($d = mysqli\_fetch\_array($data)) {  ?> |

Gambar 5.12 Potongan Kode Menampilkan Data Master Jawaban

### *Preprocessing*

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk melakukan proses *preprocessing*. Potongan kode progam dapat dilihat pada gambar 5.13.

|  |
| --- |
| function preproses($teks)  {      include '../koneksi.php';      //Bersihkan tanda baca, ganti dengan space      $teks = str\_replace("'", " ", $teks);      $teks = str\_replace("-", " ", $teks);      $teks = str\_replace(")", " ", $teks);      $teks = str\_replace("(", " ", $teks);      $teks = str\_replace("\"", " ", $teks);      $teks = str\_replace("/", " ", $teks);      $teks = str\_replace("=", " ", $teks);      $teks = str\_replace(".", " ", $teks);      $teks = str\_replace(",", " ", $teks);      $teks = str\_replace(":", " ", $teks);      $teks = str\_replace(";", " ", $teks);      $teks = str\_replace("!", " ", $teks);      $teks = str\_replace("?", " ", $teks);      //Ubah ke huruf kecil      $teks = strtolower(trim($teks));      //Stopword remove      $stoplist = array("yang", "juga", "dari", "dia", "kami", "kamu", "ini", "itu", "atau", "dan", "tersebut", "pada", "dengan", "adalah", "yaitu");      foreach ($stoplist as $i => $value) {          $teks = str\_replace($stoplist[$i], "", $teks);      }      //Terapkan stemming (ubah ke kata dasar)      $restem = mysqli\_query($koneksi, "SELECT \* FROM tb\_stem ORDER BY id");      while ($rowstem = mysqli\_fetch\_array($restem)) {          $teks = str\_replace($rowstem['term'], $rowstem['stem'], $teks);      }      //Return teks      $teks = strtolower(trim($teks));      return $teks;  } |

Gambar 5.13 Potongan Kode Preprocessing

### Hitung Bobot TF-IDF

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk melakukan proses hitung bobot TF-IDF. Potongan kode program dapat dilihat pada gambar 5.14.

|  |
| --- |
| function hitungBobot()  {      include '../koneksi.php';      //Hitung total doc id      $resn = mysqli\_query($koneksi, "SELECT DISTINCT id\_doc FROM tb\_index");      $n = mysqli\_num\_rows($resn);      //Ambil setiap record dalam tb\_index      //Hitung bobot untuk setiap term      $resBobot = mysqli\_query($koneksi, "SELECT \* FROM tb\_index ORDER BY id");      $num\_rows = mysqli\_num\_rows($resBobot);      print("Terdapat " . $num\_rows . " term yang diberikan bobot. <br>");      while ($rowBobot = mysqli\_fetch\_array($resBobot)) {          //$w = tf \* log(n/N)          $term = $rowBobot['term'];          $tf = $rowBobot['jumlah'];          $id = $rowBobot['id'];          //Jumlah dokumen yang mengandung term tersebut (N)          $resNTerm = mysqli\_query($koneksi, "SELECT COUNT(\*) AS N FROM tb\_index WHERE term = '$term'");          $rowNTerm = mysqli\_fetch\_array($resNTerm);          $NTerm = $rowNTerm['N'];          $w = $tf \* log($n / $NTerm);          //Update bobot          $resUpdateBobot = mysqli\_query($koneksi, "UPDATE tb\_index SET bobot = $w WHERE id = $id");      }  } |

Gambar 5.14 Potongan Kode Hitung Bobot TF-IDF

### Hitung Panjang Vektor Dokumen

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk melakukan proses hitung vektor dokumen. Potongan kode program dapat dilihat pada gambar 5.15.

|  |
| --- |
| function panjangVektor()  {      include '../koneksi.php';      //Hapus isi tabel vektor      mysqli\_query($koneksi, "TRUNCATE TABLE tb\_vektor");      //Ambil setiap doc id dari table index      //Hitung panjang vektor untuk setiap doc id      //Simpan ke table vektor      $resDocId = mysqli\_query($koneksi, "SELECT DISTINCT id\_doc FROM tb\_index");      $num\_rows = mysqli\_num\_rows($resDocId);      print('Terdapat ' . $num\_rows . " dokumen yang dihitung panjang vektornya. <br>");      while ($rowDocId = mysqli\_fetch\_array($resDocId)) {          $docId = $rowDocId['id\_doc'];          $resVektor = mysqli\_query($koneksi, "SELECT bobot FROM tb\_index WHERE id\_doc = $docId");          //Jumlahkan semua bobot kuadrat          $panjangVektor = 0;          while ($rowVektor = mysqli\_fetch\_array($resVektor)) {              $panjangVektor = $panjangVektor + $rowVektor['bobot'] \* $rowVektor['bobot'];          }          //Hitung akarnya          $panjangVektor = sqrt($panjangVektor);          //Masukkan ke dalam tabel vektor          $resInsertVektor = mysqli\_query($koneksi, "INSERT INTO tb\_vektor (doc\_id, panjang) VALUES ($docId, $panjangVektor)");      }  } |

Gambar 5.15 Potongan Kode Hitung Panjang Vektor Dokumen

### Hitung *Similarity*

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk melakukan proses hitung *similarity*. Potongan kode program dapat dilihat pada gambar 5.16.

|  |
| --- |
| function hitungSimilarity($query)  {      include '../koneksi.php';      //Ambil jumlah total dokumen yang telah diindex      $resn = mysqli\_query($koneksi, "SELECT COUNT(\*) AS n FROM tb\_vektor");      $rown = mysqli\_fetch\_array($resn);      $n = $rown['n'];      //Terapkan preprocessing pada query      $aquery = explode(" ", $query);      //Hitung panjang vektor query      $panjangQuery = 0;      $aBobotQuery = array();      for ($i = 0; $i < count($aquery); $i++) {          //hitung bobot untuk term ke-i pada query, log(n/N);          //hitung jumlah dokumen yang mengandung term tersebut          $resNTerm = mysqli\_query($koneksi, "SELECT COUNT(\*) AS n FROM tb\_index WHERE term = '$aquery[$i]'");          $rowNTerm = mysqli\_fetch\_array($resNTerm);          $NTerm = $rowNTerm['n'];          //idf = 0          $idf = 0;          if ($NTerm > 0) {              $idf = log($n / $NTerm);              //Simpan di array              $aBobotQuery[] = $idf;              $panjangQuery = $panjangQuery + $idf \* $idf;          }      }      $panjangQuery = sqrt($panjangQuery);      $jumlah\_mirip = 0;      //Ambil setiap term dari doc id, bandingkan dengan query      $resDocId = mysqli\_query($koneksi, "SELECT \* FROM tb\_vektor ORDER BY doc\_id");      while ($rowDocId = mysqli\_fetch\_array($resDocId)) {          $dotproduct = 0;          $docId = $rowDocId['doc\_id'];          $panjangDocId = $rowDocId['panjang'];          $resTerm = mysqli\_query($koneksi, "SELECT \* FROM tb\_index WHERE id\_doc = $docId");          while ($rowTerm = mysqli\_fetch\_array($resTerm)) {              for ($i = 0; $i < count($aquery); $i++) {                  //Jika term sama                  if ($rowTerm['term'] == $aquery[$i]) {                      $dotproduct = $dotproduct + $rowTerm['bobot'] \* $aBobotQuery[$i];                  }              }          }          if ($dotproduct > 0) {              $similarity = $dotproduct / ($panjangQuery \* $panjangDocId);              //Simpan kemiripan ke tabel cache              $resInsertCache = mysqli\_query($koneksi, "INSERT INTO tb\_cache (query, doc\_id, nilai) VALUES ('$query', $docId, $similarity)");              $jumlah\_mirip++;          }      }      if ($jumlah\_mirip == 0) {          $resInsertCache = mysqli\_query($koneksi, "INSERT INTO tb\_cache (query, doc\_id, nilai) VALUES ('$query', 0, 0)");      }  } |

Gambar 5.16 Potongan Kode Hitung Similarity

### Penentuan Respon *Chatbot*

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk melakukan proses penentuan respon *chatbot* berdasarakan *input* masukandari user. Potongan kode program dapat dilihat pada gambar 5.17.

|  |
| --- |
| function ambilCache($keyword)  {      include '../koneksi.php';      $resCache = mysqli\_query($koneksi, "SELECT \* FROM tb\_cache WHERE query = '$keyword' ORDER BY nilai DESC LIMIT 1");      $num\_rows = mysqli\_num\_rows($resCache);      if ($num\_rows > 0) {          //Tampilkan respon          while ($rowCache = mysqli\_fetch\_array($resCache)) {              $doc\_id = $rowCache['doc\_id'];              $nilai\_similarity = $rowCache['nilai'];              if ($doc\_id != 0) {                  $responChat = mysqli\_query($koneksi, "SELECT \* FROM tb\_dokumen WHERE id = $doc\_id");                  $rowChat = mysqli\_fetch\_array($responChat);                  $answer = $rowChat['dokumen'];                  print($answer);              } else {                  print('Jawaban tidak ditemukan...');              }          }      } else {          hitungSimilarity($keyword);          $resCache = mysqli\_query($koneksi, "SELECT \* FROM tb\_cache WHERE query = '$keyword' ORDER BY nilai DESC LIMIT 1");          $num\_rows = mysqli\_num\_rows($resCache);          while ($rowCache = mysqli\_fetch\_array($resCache)) {              $doc\_id = $rowCache['doc\_id'];              $nilai\_similarity = $rowCache['nilai'];              if ($doc\_id != 0) {                  $responChat = mysqli\_query($koneksi, "SELECT \* FROM tb\_dokumen WHERE id = $doc\_id");                  $rowChat = mysqli\_fetch\_array($responChat);                  $answer = $rowChat['dokumen'];                  print($answer);              } else {                  print('Jawaban tidak ditemukan...');              }          }      }  } |

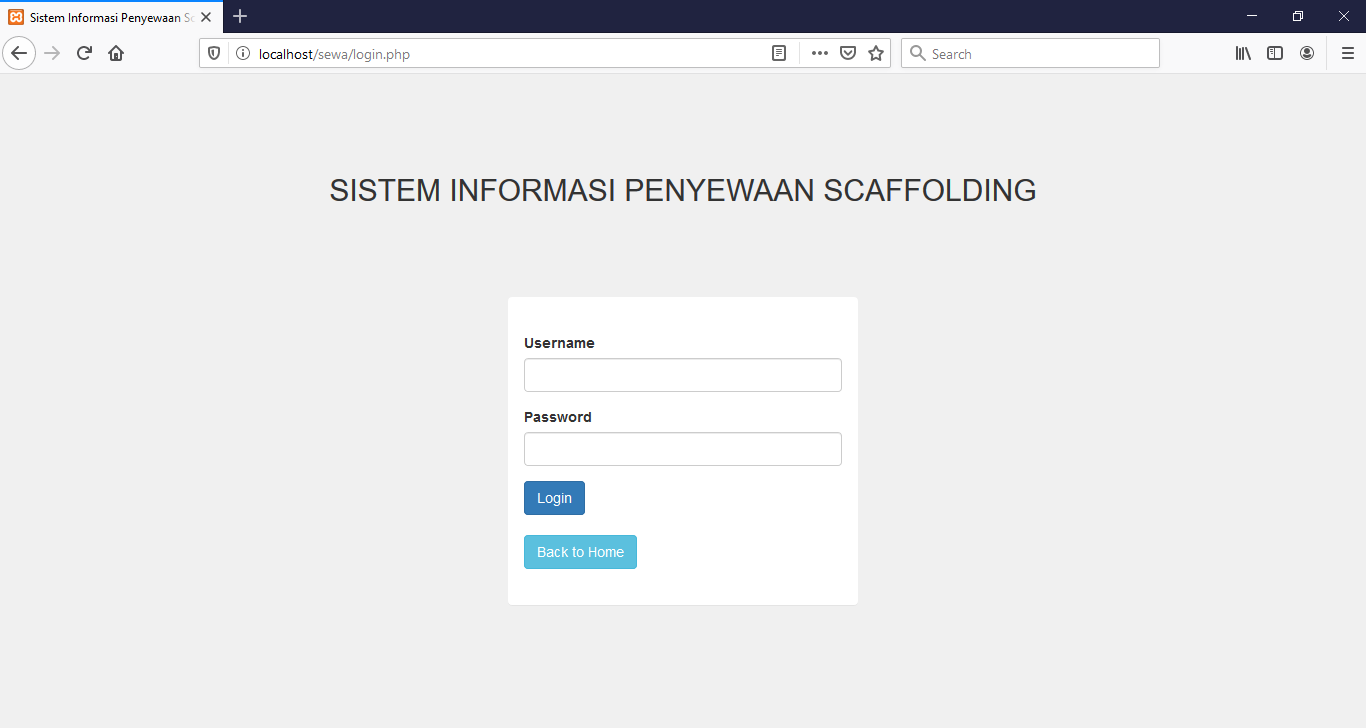
Gambar 5.17 Potongan Kode Penentuan Respon Chatbot

* 1. **Implementasi Anatarmuka Pengguna**

Implementasi antar muka yang dirancangkan sebelumnya. Berikut adalah implementasi masing-masing antar muka.

### Antarmuka Pengguna Login Admin

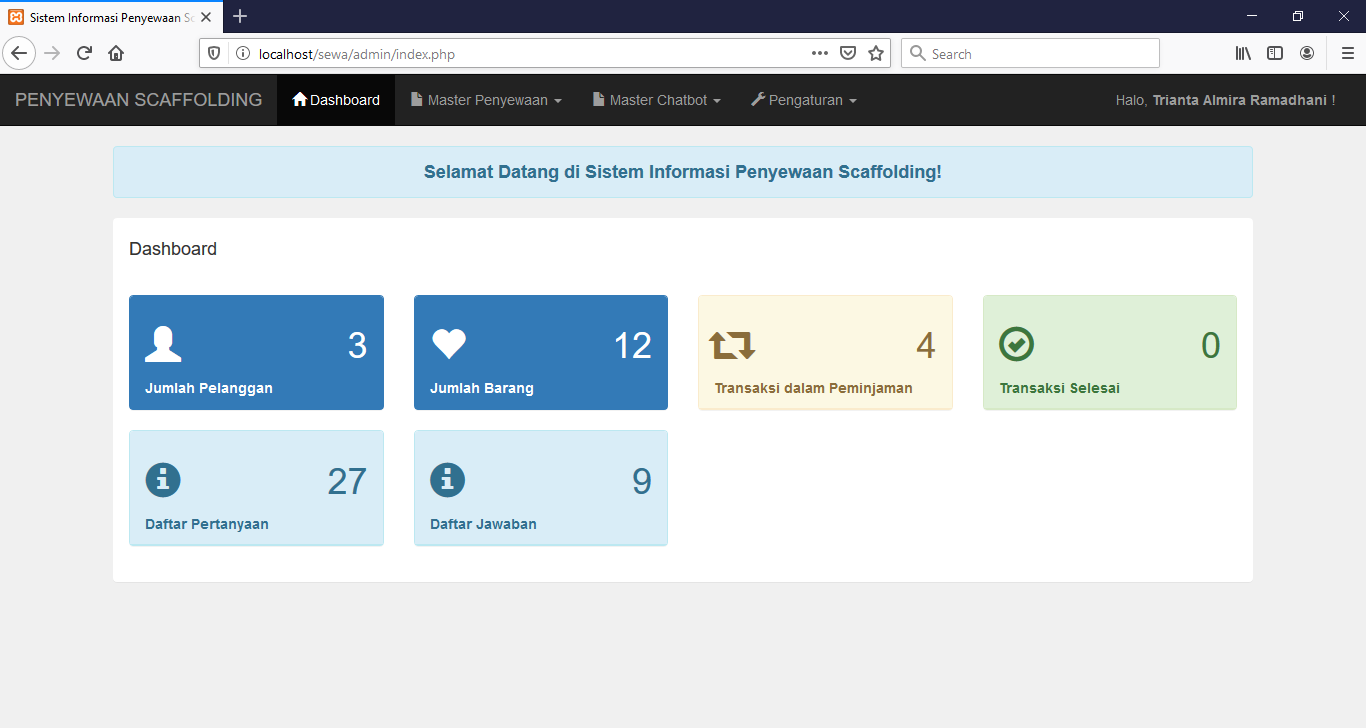
Implementasi dari antarmuka pengguna *login admin* yang sebelumnya telah dirancangkan pada gambar 4.19 dapat dilihat pada gambar 5.18.



Gambar 5.18 Halaman Login

### Antarmuka Pengguna Halaman *Dashboard*

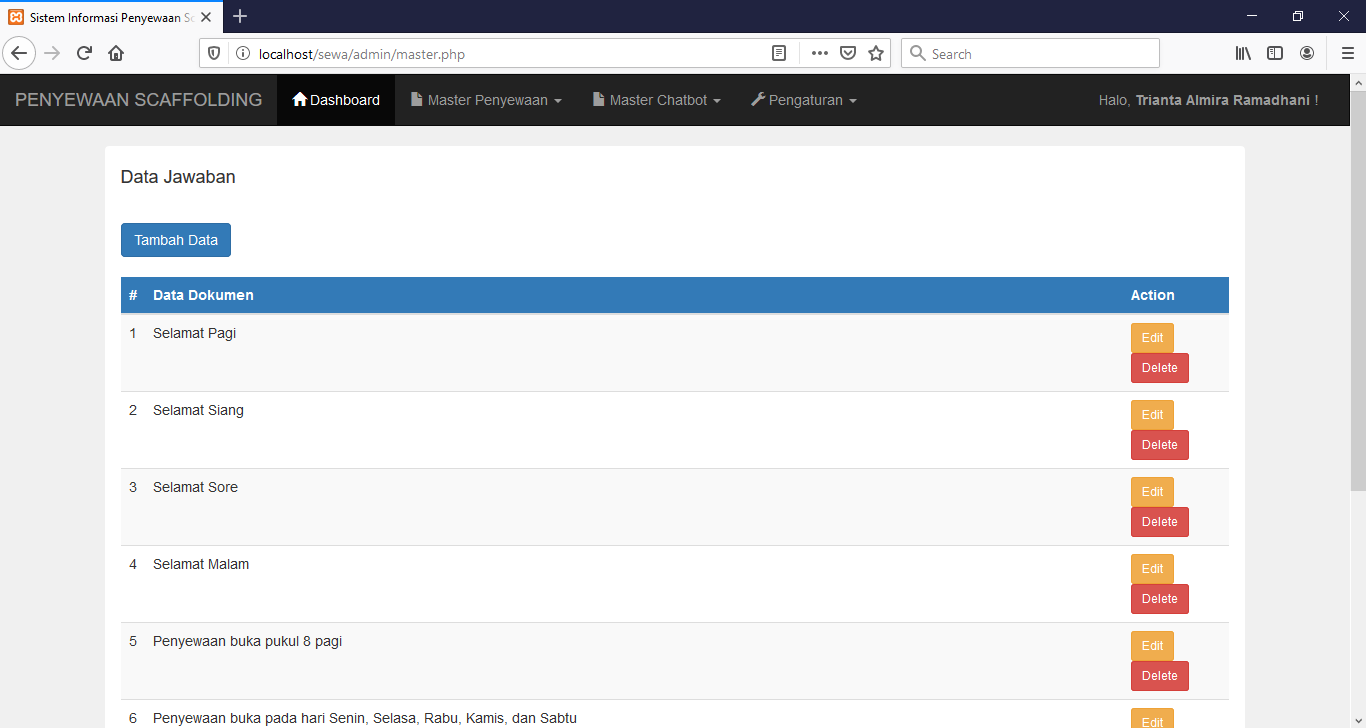
Implementasi dari antarmuka pengguna halaman *dashboard* yang sebelumnya telah dirancangkan pada gambar 4.20 dapat dilihat pada gambar 5.19.



Gambar 5.19 Halaman Dashboard

### *Interface* Halaman Kelola Data *Master*

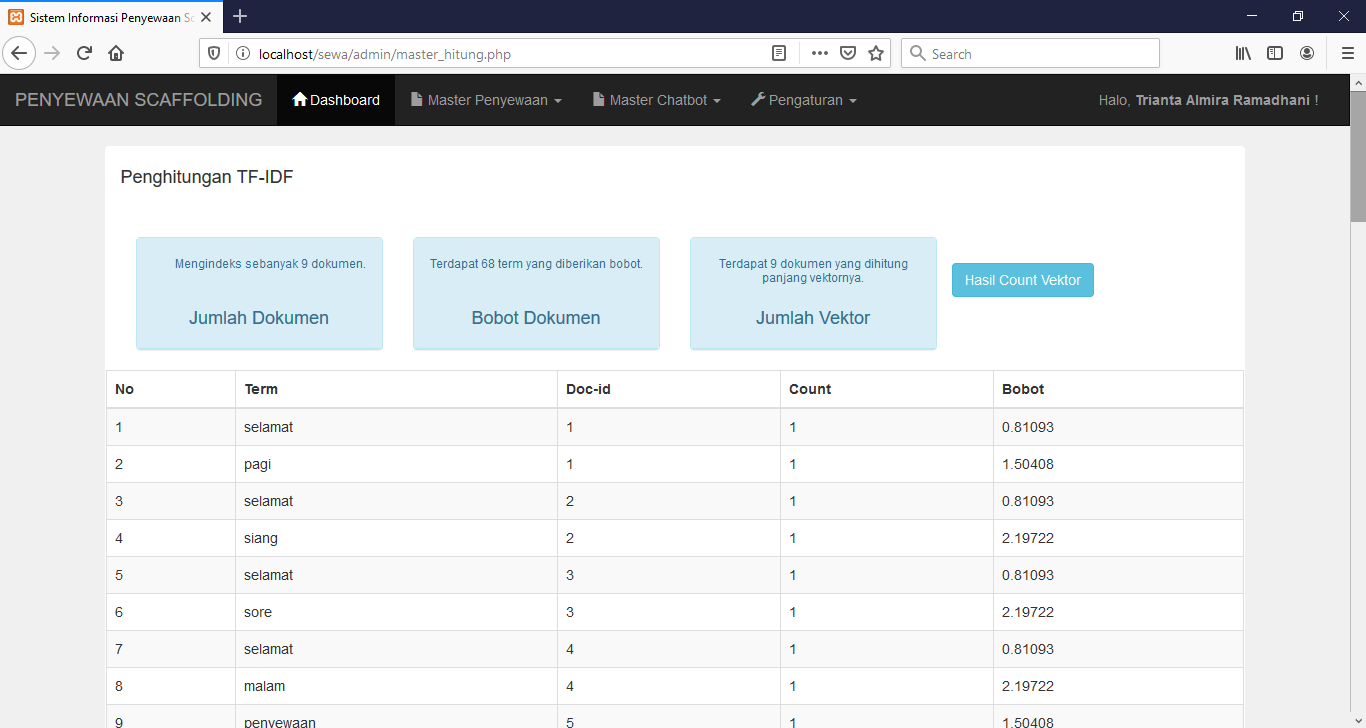
Implementasi dari antarmuka penggunahalaman kelola data *master* yang sebelumnya telah dirancangkan pada gambar 4.21 dapat dilihat pada gambar 5.20



Gambar 5.20 Halaman Master Jawaban

### *Interface* Halaman Hitung TF-IDF

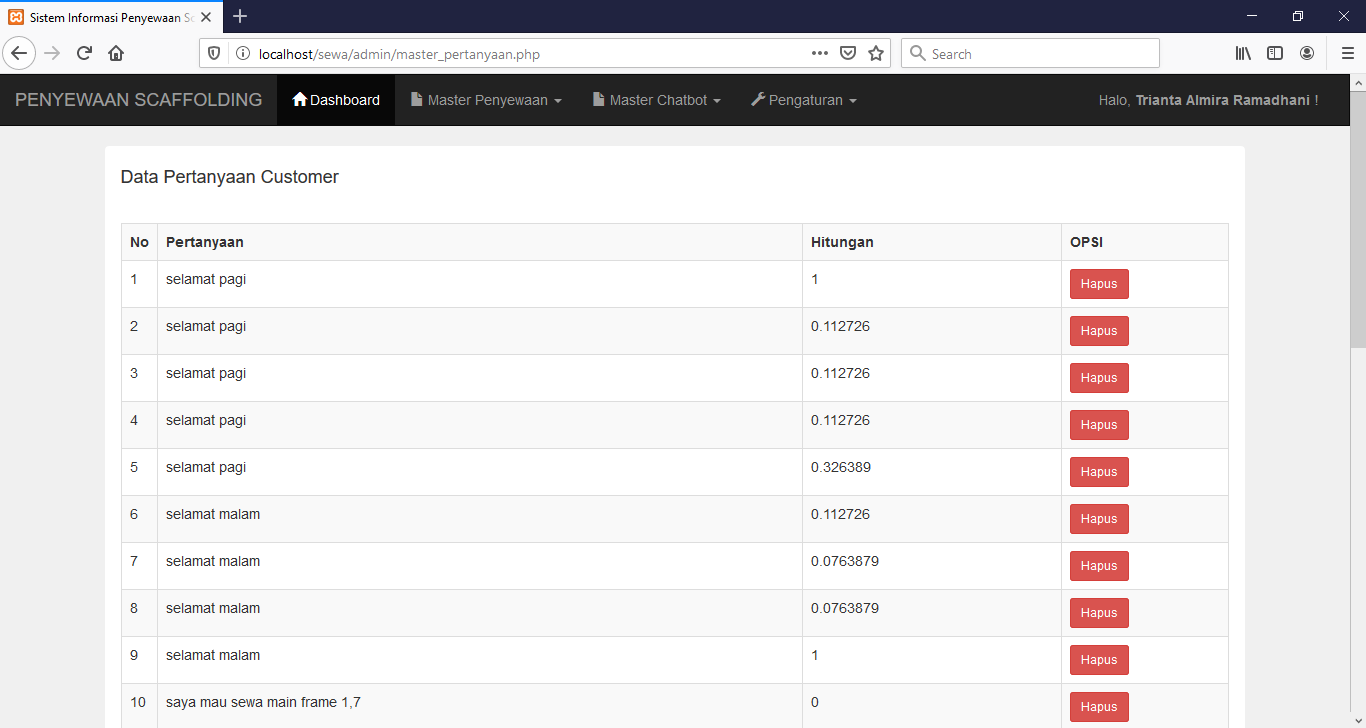
Implementasi dari antarmuka penggunahalaman hitung TF-IDF yang sebelumnya telah dirancangkan pada gambar 4.22 dapat dilihat pada gambar 5.21



Gambar 5.21 Halaman Perhitungan TF-IDF

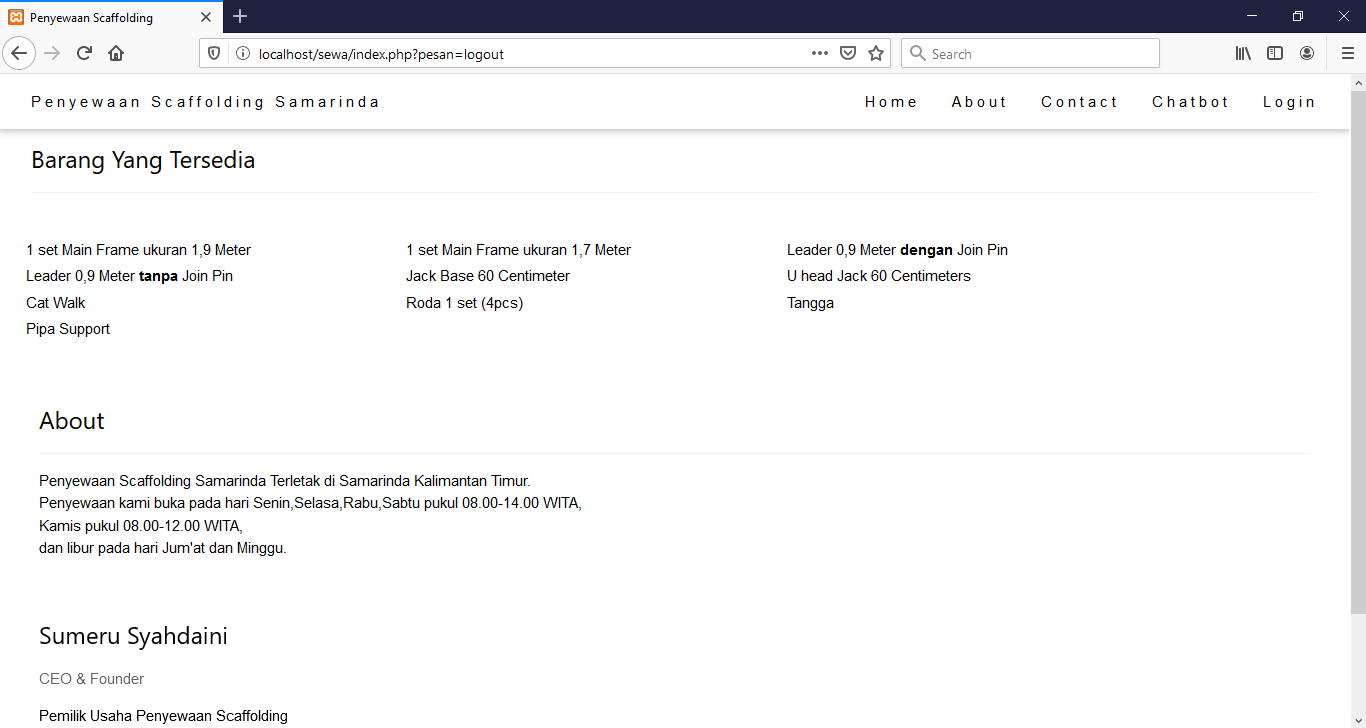
### *Interface* Halaman Data Pertanyaan

Implementasi dari antarmuka penggunahalaman hitung TF-IDF yang sebelumnya telah dirancangkan pada gambar 4.23 dapat dilihat pada gambar 5.22



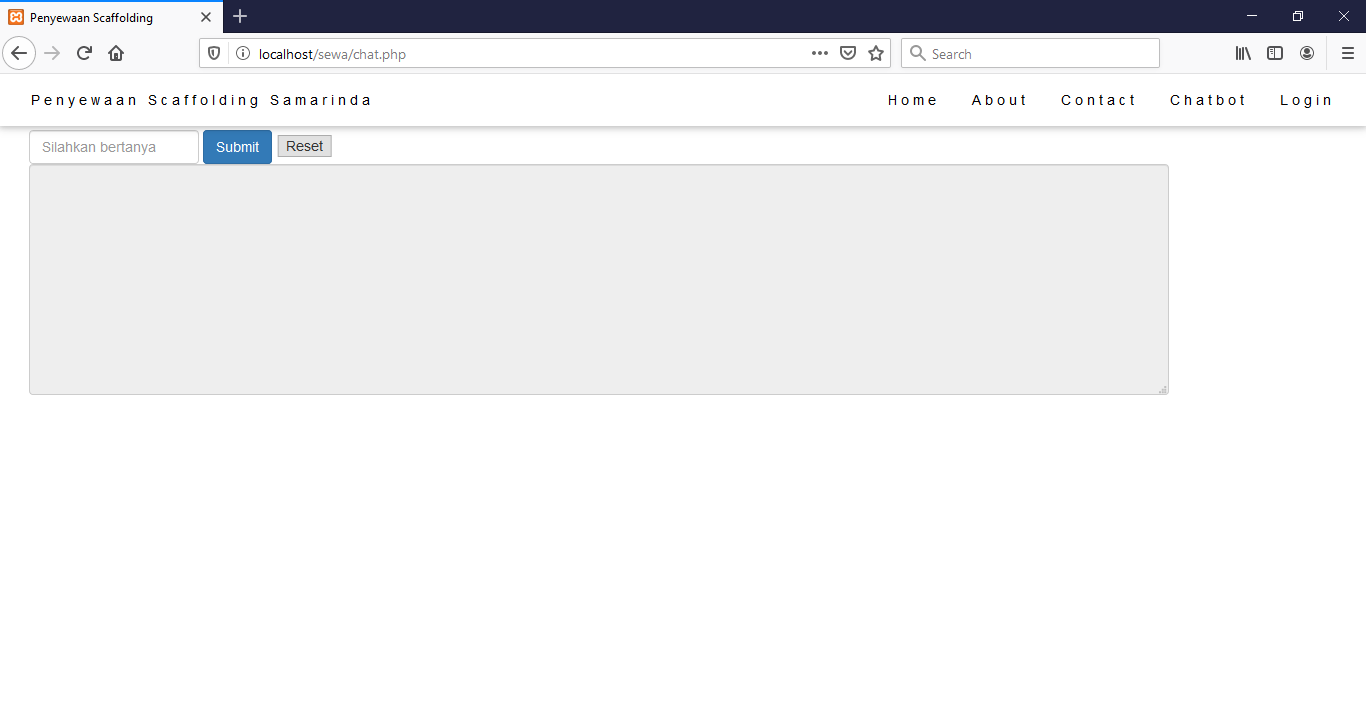
Gambar 5.22 Halaman Pertanyaan dari Customer

Halaman utama dimana customer dapat melihat barang apa saja yang disewakan, mengetahui mengenai penyewaan scaffolding, siapa pemiliknya dan dapat memberikan komentar mengenai proses penyewaan yang telah dilakukan. Halaman utama ditunjukan pada Gambar 5.23 dari implementasi gambar 4.17.



Gambar 5.23 Halaman Utama

Halaman berikutnya adalah halaman bagi customer untuk melakukan chat dengan bot untuk menanyakan jam buka, harga sewa barang, hari kerja. Ditujukan pada Gambar 5.24 dari implementasi gambar 4.18.



Gambar 5.24 Halaman Chatbot

* 1. **Pengujian Fungsional**

Pengujian fungsional dilakukan dengan cara menguji setiap fitur aplikasi dan melihat kecocokan hasil yang terjadi dengan hasil yang diinginkan.

### Pengujian Sistem

Pengujian sistem ini dilakukan dengan cara menjalankan aplikasi secara detail pada setiap menu yang bertujuan mengetahui fitur mana yang sudah berfungsi baik maupun yang tidak berfungsi. Hal ini bertujuan agar aplikasi dapat berjalan sebagaimana fungsinya.

Tabel 5.1 Pengujian Fungsional Halaman Login Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Form Login Admin | Mengalihkan admin ke *dashboard* admin sesuai *username* dan *password*. | Admin yang berhasil memasukkan *username* dan *password* beralih ke halaman *dashboard*. | Berhasil |

Tabel 5.2 Pengujian Fungsional Menu Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu *Dashboard* | Menampilkan halaman *dashboard* | Halaman *dashboard* berhasil tampil. | Berhasil |
| 2. | *Dropdown Master Penyewaan Barang* | Menampilkan data barang | Data barang berhasil tampil | Berhasil |
| 3. | *Dropdown Master Penyewaan Pelanggan* | Menampilkan data pelanggan | Data pelanggan berhasil tampil | Berhasil |
| 4. | *Dropdown Master Penyewaan Transaksi* | Menampilkan data Transaksi | Data Transaksi berhasil tampil | Berhasil |
| 5. | *Dropdown Master Chatbot Jawaban* | Menampilkan data Jawaban | Data Jawaban berhasil tampil | Berhasil |
| 6. | *Dropdown Master Chatbot Hitung TF-IDF* | Menampilkan data hitung TF-IDF | Data Hitung TF-IDF Berhasil tampil | Berhasil |
| 7. | *Dropdown Master Chatbot History Pertanyaan* | Menampilkan data pertanyaan | Data Pertanyaan berhasil tampil | Berhasil |
| 8. | *Dropdown Pengaturan Ganti Password* | Menampilkan halaman ganti password | Halaman ganti password berhasil tampil | Berhasil |
| 9. | *Dropdown Pengaturan Logout* | Menampilkan halaman awal sistem | Halaman awal berhasil tampil | Berhasil |

Tabel 5.3 Pengujian Fungsional Admin Data Produk

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu Data Produk | Menampilkan data produk. | Data produk berhasil tampil. | Berhasil |
| 2. | Tombol Tambah Data Produk | Menampilkan *form* tambah dan menyimpan data produk. | Form tambah data tampil dan data produk berhasil tersimpan. | Berhasil |
| 3. | Tombol *Edit* Data Produk | Menampilkan *form edit* dan mengubah data produk. | Form *edit* data tampil dan data produk berhasil dirubah. | Berhasil |
| 4. | Tombol *Delete* Data Produk | Menghapus data produk. | Data produk berhasil terhapus. | Berhasil |

Tabel 5.4 Pengujian Fungsional Admin Data Pelanggan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu Data Pelanggan | Menampilkan data pelanggan. | Data pelanggan berhasil tampil. | Berhasil |
| 2. | Tombol Tambah Data Pelanggan | Menampilkan *form* tambah dan menyimpan data pelanggan. | Form tambah data tampil dan data pelanggan berhasil tersimpan. | Berhasil |
| 3. | Tombol *Edit* Data Pelanggan | Menampilkan *form edit* dan mengubah data pelanggan. | Form *edit* data tampil dan data pelanggan berhasil dirubah. | Berhasil |
| 4. | Tombol *Delete* Data Pelanggan | Menghapus data pelanggan. | Data pelanggan berhasil terhapus. | Berhasil |

Tabel 5.5 Pengujian Fungional Admin Data Transaksi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu Data Transaksi | Menampilkan data transaksi. | Data transaksi berhasil tampil. | Berhasil |
| 2. | Tombol Tambah Data Transaksi | Menampilkan *form* tambah dan menyimpan data transaksi. | Form tambah data tampil dan data transaksi berhasil tersimpan. | Berhasil |
| 3. | Tombol Detail Data Transaksi | Menampilkan halaman detail data transaksi. | Detail data transaksi berhasil tampil. | Berhasil |
| 4. | Tombol *Edit* Data Transaksi | Menampilkan *form edit* dan mengubah data transaksi. | Form *edit* data tampil dan data transaksi berhasil dirubah. | Berhasil |
| 5. | Tombol *Delete* Data Transaksi | Menghapus data transaksi. | Data transaksi berhasil terhapus. | Berhasil |

Tabel 5.6 Pengujian Fungsional Admin Data Master

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu Data Jawaban | Menampilkan datajawaban. | Data jawaban berhasil tampil. | Berhasil |
| 2. | Tombol Tambah Data Jawaban | Menampilkan *form* tambah dan menyimpan datajawaban. | Form tambah data tampil dan data jawaban berhasil tersimpan. | Berhasil |
| 4. | Tombol *Edit* DataJawaban | Menampilkan *form edit* dan mengubah data jawaban. | Form *edit* data tampil dan datajawaban berhasil dirubah. | Berhasil |
| 5. | Tombol *Delete* Data Jawaban | Menghapus datajawaban. | Data jawaban berhasil terhapus. | Berhasil |

Tabel 5.7 Pengujian Fungsional Admin Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu Data Perhitungan TF-IDF | Menampilkan data hasil perhitungan TF-IDF | Data perhitungan berhasil tampil. | Berhasil |
| 2. | Tombol Hitung Vektor | Menampilkan *modal* hasil perhitungan vektor jawaban | *Modal* perhitungan vektor jawaban berhasil tampil. | Berhasil |

Tabel 5.8 Pengujian Fungsional Admin Menu History Jawaban

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | Menu Data History Pertanyaan | Menampilkan data *similarity* antara *query user* dengan dokumen sistem. | Data *similarity* berhasil tampil. | Berhasil |
| 2. | Tombol *Delete History* Pertanyaan | Menghapus data *history* pertanyaan. | Data history pertanyaan berhasil terhapus. | Berhasil |

Tabel 5.9 Pengujian Fungsional User Chatbot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Menu** | **Hasil yang Diharapkan** | **Hasil yang Diperoleh** | **Status** |
| 1. | *Form* Pertanyaan *User* | Sistem menampilkan respon informasi berdasarkan *keyword* dari *user*. | Respon informasi berhasil tampil sesuai dengan *keyword* dari *user*. | Berhasil |

### Pengujian Metode

Pengujian metode yang digunakan pada sistem ini dimaksudkan untuk menguji ketepatan perhitungan dari metode yang digunakan, dengan menggunakan metode TF-IDF dan *Cosine Similarity* untuk menentukan respon dari *chatbot* sesuai dengan *keyword* yang dimasukkan oleh user. Pengujian ketepatan penghitungan metode menggunak teknik *Recall and Precision*. Penghitungan dilakukan dengan melakukan percobaan sebanyak 25 kali menggunakan *query* terhadap 25 dokumen yang ada di dalam *database*. *Query* yang digunakan adalah *query* yang memiliki masing-masing 1 untuk setiap *query*. Artinya, 1 *query* hanya relevan dengan 1 dokumen dimana dokumen tersebut merupakan dokumen jawaban yang relevan berdasarkan query jawaban yang dimasukkan.

Setelah melakukan percobaan terhadap 25 dokumen dengan *query* tersebut, didapatkan hasil benar yang ditemukan berjumlah 21 dokumen yang ditemukan relevan (sesuai dengan *query*), dan 4 dokumen yang ditemukan tidak relevan (tidak sesuai dengan *query*). Oleh karena itu, maka hasil pengujian *Recall and Precision* adalah sebagai berikut:

* 1. **Pengujian Akurasi**

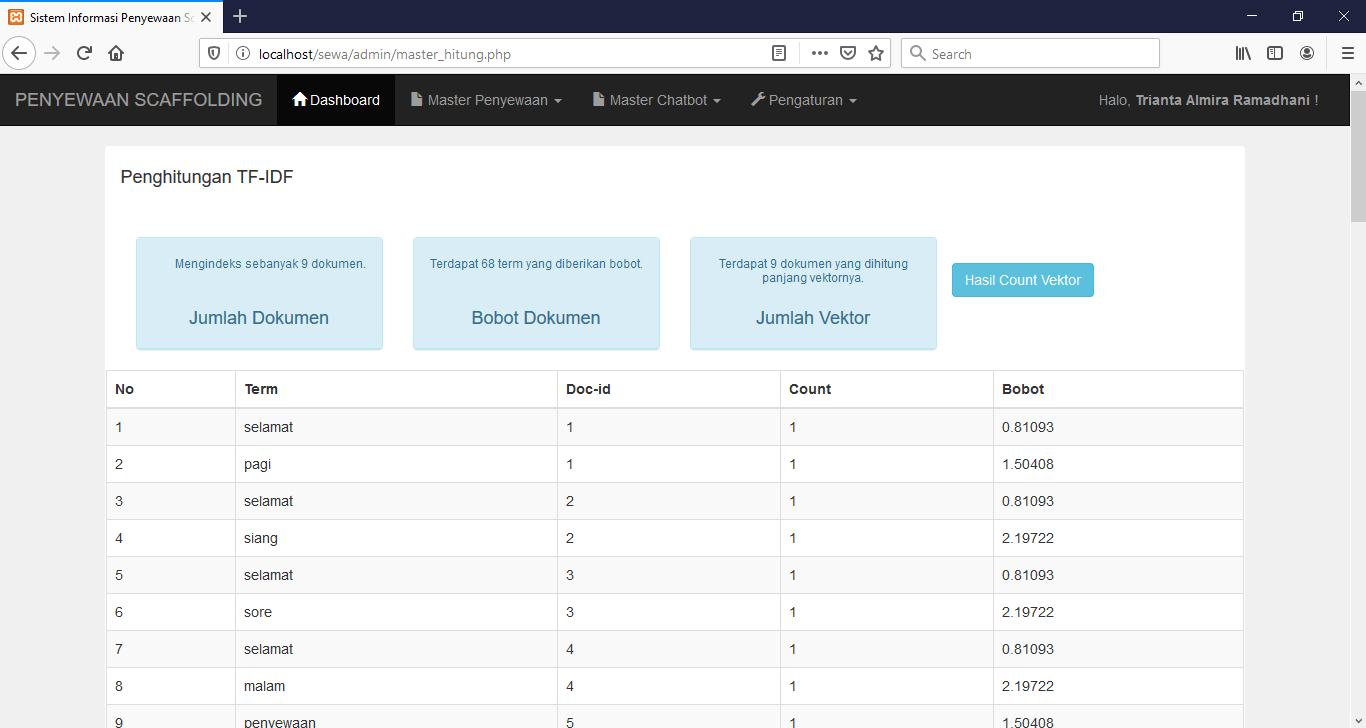
Tabel 5.10 Tabel Pengujian Akurasi

|  |  |  |  |
| --- | --- | --- | --- |
| No | Pertanyaan | Jawaban Sistem | Akurasi |
| 1. | Pinjam main frame | harga sewa main frame 1,7 selama 1 minggu 30.000 rupiah, selama 2 minggu 40.000 rupiah, selama 1 bulan 50.000 rupiah | Akurat |
| 2. | Selamat pagi | Selamat pagi | Akurat |
| 3 | Selamat siang | Selamat siang | Akurat |
| 4 | Selamat sore | Selamat sore | Akurat |
| 5 | Selamat malam | Selamat malam | Akurat |
| 6 | Mau menyewa main frame | harga sewa main frame 1,7 selama 1 minggu 30.000 rupiah, selama 2 minggu 40.000 rupiah, selama 1 bulan 50.000 rupiah | Akurat |
| 7 | Buka jam berapa? | Penyewaan buka pukul 8 pagi | Akurat |
| 8 | Buka hari apa? | Penyewaan buka pada hari Senin, Selasa, Rabu, Kamis, dan Sabtu | Akurat |
| 9 | Libur hari apa? | hari jumat dan minggu libu | Akurat |
| 10. | Mau beli scaffolding | Jawaban tidak ditemukan | Akurat |

**BAB VI. HASIL DAN PEMBAHASAN**

* 1. **Hasil**

Data yang digunakan dalam penelitian ini adalah data harga layanan penyewaan 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:



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 dilihat 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.

## 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*.

**BAB VII. KESIMPULAN DAN SARAN**

* 1. **Kesimpulan**

Berdasarkan pembahasan yang telah dilakukan pada bab I hingga bab VI, maka dapat disimpulkan bahwa Sistem ini dapat Melakukan Tokenezing (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 user. Sistem juga dapat memudahkan user dalam menerima informasi sesuai dengan yang diharapkan.

* 1. **Saran**

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

1. Untuk kedepannya disarankan chatbot dapat menambahkan respon inputan dengan kata-kata yang salah dalam pengetikan.
2. Untuk kedepannya disarankan chatbot menambahkan informasi-informasi penyewaan menjadi lebih detail.
3. Untuk kedepannya disarankan chatbot dapat dikembangkan lebih lanjut.

**DAFTAR PUSTAKA**

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>.

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.

Suryani, Dhebys & Larasati, Eka. (2017). Aplikasi Chatbot Objek Wisata Jawa Timur Berbasis AIML. *SMARTICS Journal: 3,* 47-54.

Suryani, Dhebys & Putera, Yoga. (2018). Aplikasi Chatbot Berbasis Web Pada Sistem Informasi Layanan Publik Kesehatan di Malang dengan Menggunakan Metode TF-IDF. *Jurnal* *Informatika Polinema, 4(3):* 224-228.

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.

Dwi Listio, Y., Zulkarnain, A., Tirtana, A. (2019). Pembuatan Sistem Pencarian Pekerjaan Menggunakan TF-IDF. *Jurnal Ilmiah Teknologi Informasi Asia, 13(2):* 91-100.

Yamaguchi, Hiroshi & Mozgoyov, Maxim. (2018). A Chatbot Based on AIML Rules Extracted From Twitter Dialogues. *Communication Paper Poznan, 17:* 37-42.

Astiningrum, Mungki & Shoburu Rohmah, Maya. (2018). Implementasi NLP dengan Konversi Kata pada Sistem Chatbot Konsultasi Laktasi. *Jurnal Informatika Polinema, 5(1):* 46-52.

Natadian Astuti, Rani & Fatchan, Muhammad. (2019). Perancangan Aplikasi Teknologi Chatbot untuk Industri Komersial 4.0. *e-Prosiding SNasTekS 2019, 4:* 339-348.

Melita, Ria & Dirjam, Taslimun. (2018). Penerapan Metode Term Frequency Inverse Document Frequency dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Syarah Umdatil Ahkam). *Jurnal* *Teknik Informatika, 11(2):* 149-164.

Satria Paliwahet, I Nyoman & Gede Darma Putra, I Ketut. (2017). Pencarian Informasi Wisata Daerah Bali Menggunakan Teknologi Chatbot. *Lontar Komputer, 8(3):* 144-153.

Riyani, Ade & Burhanuddin, Auliya. (2019). Penerapan Cosine Similarity dan Pembobotan TF-IDF untuk Mendeteksi Kemiripan Dokumen. *Jurnal Linguistik Komputasional, 2(1): 23-27*

# LAMPIRAN – LAMPIRAN

Lampiran 1 Source Code Tampilan *Chatbot*

|  |
| --- |
| <!DOCTYPE html>  <html>  <title>Penyewaan Scaffolding</title>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="assets/css/chat.css">  <link rel="stylesheet" type="text/css" href="assets/css/bootstrap.css">  <script type="text/javascript" src="assets/js/jquery.js"></script>  <script type="text/javascript" src="assets/js/bootstrap.js"></script>  <script src="assets/js/chat.js"></script>  <body>      <!-- Navbar (sit on top) -->      <div class="w3-top">          <div class="w3-bar w3-white w3-wide w3-padding w3-card">              <a href="index.php/#home" class="w3-bar-item w3-button">Penyewaan Scaffolding Samarinda</a>              <!-- Float links to the right. Hide them on small screens -->              <div class="w3-right w3-hide-small">                  <a href="index.php/#home" class="w3-bar-item w3-button">Home</a>                  <a href="index.php/#about" class="w3-bar-item w3-button">About</a>                  <a href="index.php/#contact" class="w3-bar-item w3-button">Contact</a>                  <a href="chat.php" class="w3-bar-item w3-button">Chatbot</a>                  <a href="login.php" class="w3-bar-item w3-button">Login</a>              </div>          </div>      </div>      <?php include 'fungsi.php'; ?>      <div class="col-md-9">          <h3>Percobaan Chatbot</h3>          <div class="container mt-3">              <div class="card">                  <div class="card-body">                      <form action="" method="post" class="form-inline">                          <input type="text" name="keyword" id="keyword" class="form-control mr-2" placeholder="Silahkan bertanya" required>                          <button type="submit" class="btn btn-primary" name="submit">Submit</button>                          <input type="button" value="Reset" onclick="javascript:eraseText();">                      </form>                      <textarea name="respon" id="respon" cols="30" rows="10" class="form-control mt-3" readonly>                      <?php                      if (isset($\_POST['submit'])) {                          $keyword = $\_POST['keyword'];                          ambilCache($keyword);                      }                      ?>                  </textarea>                  </div>              </div>          </div>          <h5><b>Kalimat yang ditanyakan harus baku<br>          Tidak disingkat</b></h5>          <br><a href="https://api.whatsapp.com/send?phone=6281253831945">Contact 1</a>          <br><a href="https://api.whatsapp.com/send?phone=6281250366999">Contact 2</a>          <br><a href="https://api.whatsapp.com/send?phone=6281285201500">Contact 3</a>      </div>      <script>          function eraseText() {              document.getElementById("respon").value = "";          }      </script>  </body>  </html> |

Lampiran 2 Source Code Perhitungan *TF-IDF*

|  |
| --- |
| <?php  function preproses($teks)  {      session\_start();      include 'koneksi.php';      $teks = str\_replace("'", " ", $teks);      $teks = str\_replace("-", " ", $teks);      $teks = str\_replace(")", " ", $teks);      $teks = str\_replace("(", " ", $teks);      $teks = str\_replace("\"", " ", $teks);      $teks = str\_replace("/", " ", $teks);      $teks = str\_replace("=", " ", $teks);      $teks = str\_replace(".", " ", $teks);      $teks = str\_replace(",", " ", $teks);      $teks = str\_replace(":", " ", $teks);      $teks = str\_replace(";", " ", $teks);      $teks = str\_replace("!", " ", $teks);      $teks = str\_replace("?", " ", $teks);      $teks = strtolower(trim($teks));      $stoplist = array("yang", "juga", "dari", "dia", "kami", "kamu", "ini", "itu", "atau", "dan", "tersebut", "pada", "dengan", "adalah", "yaitu");      foreach ($stoplist as $i => $value) {          $teks = str\_replace($stoplist[$i], "", $teks);      }      $restem = mysqli\_query($koneksi, "SELECT \* FROM tb\_stem ORDER BY id");      while ($rowstem = mysqli\_fetch\_array($restem)) {          $teks = str\_replace($rowstem['term'], $rowstem['stem'], $teks);      }      $teks = strtolower(trim($teks));      return $teks;  }  function buatIndex()  {      include '../koneksi.php';      mysqli\_query($koneksi, "TRUNCATE TABLE tb\_index");      $resDokumen = mysqli\_query($koneksi, "SELECT \* FROM tb\_jawaban ORDER BY id");      $num\_rows = mysqli\_num\_rows($resDokumen);      print("Mengindeks sebanyak " . $num\_rows . " dokumen. <br>");      while ($row = mysqli\_fetch\_array($resDokumen)) {          $docId = $row['id'];          $dokumen = $row['dokumen'];          $dokumen = preproses($dokumen);          $aDokumen = explode(" ", trim($dokumen));          foreach ($aDokumen as $j => $value) {              if ($aDokumen[$j] != "") {                  $rescount = mysqli\_query($koneksi, "SELECT jumlah FROM tb\_index WHERE term = '$aDokumen[$j]' AND id\_doc = $docId");                  $num\_rows = mysqli\_num\_rows($rescount);                  if ($num\_rows > 0) {                      $rowcount = mysqli\_fetch\_array($rescount);                      $count = $rowcount['jumlah'];                      $count++;                      mysqli\_query($koneksi, "UPDATE tb\_index SET jumlah = $count WHERE term = '$aDokumen[$j]' AND id\_doc = $docId");                  } else {                      mysqli\_query($koneksi, "INSERT INTO tb\_index (term, id\_doc, jumlah) VALUES ('$aDokumen[$j]', $docId, 1)");                  }              }          }      }  }  function hitungBobot()  {      include '../koneksi.php';      $resn = mysqli\_query($koneksi, "SELECT DISTINCT id\_doc FROM tb\_index");      $n = mysqli\_num\_rows($resn);      $resBobot = mysqli\_query($koneksi, "SELECT \* FROM tb\_index ORDER BY id");      $num\_rows = mysqli\_num\_rows($resBobot);      print("Terdapat " . $num\_rows . " term yang diberikan bobot. <br>");      while ($rowBobot = mysqli\_fetch\_array($resBobot)) {          //$w = tf \* log(n/N)          $term = $rowBobot['term'];          $tf = $rowBobot['jumlah'];          $id = $rowBobot['id'];          $resNTerm = mysqli\_query($koneksi, "SELECT COUNT(\*) AS N FROM tb\_index WHERE term = '$term'");          $rowNTerm = mysqli\_fetch\_array($resNTerm);          $NTerm = $rowNTerm['N'];          $w = $tf \* log($n / $NTerm);          $resUpdateBobot = mysqli\_query($koneksi, "UPDATE tb\_index SET bobot = $w WHERE id = $id");      }  }  function panjangVektor()  {      include '../koneksi.php';      mysqli\_query($koneksi, "TRUNCATE TABLE tb\_vektor");      $resDocId = mysqli\_query($koneksi, "SELECT DISTINCT id\_doc FROM tb\_index");      $num\_rows = mysqli\_num\_rows($resDocId);      print('Terdapat ' . $num\_rows . " dokumen yang dihitung panjang vektornya. <br>");      while ($rowDocId = mysqli\_fetch\_array($resDocId)) {          $docId = $rowDocId['id\_doc'];          $resVektor = mysqli\_query($koneksi, "SELECT bobot FROM tb\_index WHERE id\_doc = $docId");          $panjangVektor = 0;          while ($rowVektor = mysqli\_fetch\_array($resVektor)) {              $panjangVektor = $panjangVektor + $rowVektor['bobot'] \* $rowVektor['bobot'];          }          $panjangVektor = sqrt($panjangVektor);          $resInsertVektor = mysqli\_query($koneksi, "INSERT INTO tb\_vektor (doc\_id, panjang) VALUES ($docId, $panjangVektor)");      }  }  function hitungSimilarity($query)  {      include 'koneksi.php';      $resn = mysqli\_query($koneksi, "SELECT COUNT(\*) AS n FROM tb\_vektor");      $rown = mysqli\_fetch\_array($resn);      $n = $rown['n'];      $aquery = explode(" ", $query);      $panjangQuery = 0;      $aBobotQuery = array();      for ($i = 0; $i < count($aquery); $i++) {          $resNTerm = mysqli\_query($koneksi, "SELECT COUNT(\*) AS n FROM tb\_index WHERE term = '$aquery[$i]'");          $rowNTerm = mysqli\_fetch\_array($resNTerm);          $NTerm = $rowNTerm['n'];          $idf = 0;          if ($NTerm > 0) {              $idf = log($n / $NTerm);              $aBobotQuery[] = $idf;              $panjangQuery = $panjangQuery + $idf \* $idf;          }      }      $panjangQuery = sqrt($panjangQuery);      $jumlah\_mirip = 0;      $resDocId = mysqli\_query($koneksi, "SELECT \* FROM tb\_vektor ORDER BY doc\_id");      while ($rowDocId = mysqli\_fetch\_array($resDocId)) {          $dotproduct = 0;          $docId = $rowDocId['doc\_id'];          $panjangDocId = $rowDocId['panjang'];          $resTerm = mysqli\_query($koneksi, "SELECT \* FROM tb\_index WHERE id\_doc = $docId");          while ($rowTerm = mysqli\_fetch\_array($resTerm)) {              for ($i = 0; $i < count($aquery); $i++) {                  if ($rowTerm['term'] == $aquery[$i]) {                      $dotproduct = $dotproduct + $rowTerm['bobot'] \* $aBobotQuery[$i];                  }              }          }          if ($dotproduct > 0) {              $similarity = $dotproduct / ($panjangQuery \* $panjangDocId);              $resInsertCache = mysqli\_query($koneksi, "INSERT INTO tb\_pertanyaan (query, doc\_id, nilai) VALUES ('$query', $docId, $similarity)");              $jumlah\_mirip++;          }      }      if ($jumlah\_mirip == 0) {          $resInsertCache = mysqli\_query($koneksi, "INSERT INTO tb\_pertanyaan (query, doc\_id, nilai) VALUES ('$query', 0, 0)");      }  }  function ambilCache($keyword)  {      include 'koneksi.php';      $resCache = mysqli\_query($koneksi, "SELECT \* FROM tb\_pertanyaan WHERE query = '$keyword' ORDER BY nilai DESC LIMIT 1");      $num\_rows = mysqli\_num\_rows($resCache);      if ($num\_rows > 0) {          while ($rowCache = mysqli\_fetch\_array($resCache)) {              $doc\_id = $rowCache['doc\_id'];              $nilai\_similarity = $rowCache['nilai'];              if ($doc\_id != 0) {                  $responChat = mysqli\_query($koneksi, "SELECT \* FROM tb\_jawaban WHERE id = $doc\_id");                  $rowChat = mysqli\_fetch\_array($responChat);                  $answer = $rowChat['dokumen'];                  print($answer);              } else {                  print('Jawaban tidak ditemukan...');              }          }      } else {          hitungSimilarity($keyword);          $resCache = mysqli\_query($koneksi, "SELECT \* FROM tb\_pertanyaan WHERE query = '$keyword' ORDER BY nilai DESC LIMIT 1");          $num\_rows = mysqli\_num\_rows($resCache);          while ($rowCache = mysqli\_fetch\_array($resCache)) {              $doc\_id = $rowCache['doc\_id'];              $nilai\_similarity = $rowCache['nilai'];              if ($doc\_id != 0) {                  $responChat = mysqli\_query($koneksi, "SELECT \* FROM tb\_jawaban WHERE id = $doc\_id");                  $rowChat = mysqli\_fetch\_array($responChat);                  $answer = $rowChat['dokumen'];                  print($answer);              } else {                  print('Silahkan menghubungi kontak dibawah.');              }          }      }  } |