

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

Disusun oleh:

TRIANTA ALMIRA RAMADHANI NIM. 1641720097

Skripsi ini telah diuji pada tanggal 30 Juli 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 : Deddy Kusbianto Purwoko Aji,
Ir., M.MKom
NIP. 19621128 198811 1 001
4. Penguji II : Budi Harijanto, ST., MMkom
NIP. 19620105 199003 1 002

Mengetahui,

Ketua Jurusan
Teknologi Informasi

Ketua Program Studi
Teknik Informatika

Rudy Ariyanto, S.T., M.Cs.
NIP. 19711110 199903 1 002

Imam Fahrur Rozi, S.T., M.T.
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 Juli 2020

Trianta Almira R.

ABSTRAK

Ramadhani, Trianta Almira. “Aplikasi *Chatbot* pada Sistem Informasi Penyewaan Scaffolding 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.

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

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

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

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.

Current technological developments make it easier for users to access existing applications. One of them is by implementing Website Based Information System. Based on the results of previous studies applying Chatbot with the N-gram, TF-IDF, and Cosine Similarity methods, it is proven that the Chatbot System in the application can facilitate the public in finding information

Therefore, CV. Scaffolding Samarinda requires providing fast information system services at all times. Based on the testing that has been done, it can be concluded that this system can do tokening (separating the compiler words from a document) and calculate the weight of TF-IDF and Cosine Similarity to find answers to the system. So when the user writes a question, the answer will appear according to customer expectations. The system can also make it easier for customers to receive information in accordance with what is expected at that time.

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

KATA PENGANTAR

Puji Syukur kami panjatkan kehadiran Allah SWT atas segala rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “APLIKASI *CHATBOT* PADA SISTEM INFORMASI PENYEWAAN SCAFFOLDING 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, 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, tante sekaligus pengganti ibu selama saya kuliah 4 tahun, yang mendorong saya untuk segera menyelesaikan dan tertib dalam melakukan semua kegiatan.

10. Joshua, Wella, Eris, Ayu, Yayas, Aang, Mbak Tya, Yoga, Agung, Ammar, Fika teman seperjuangan dari lulus SMA beserta keluarga.

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 Juli 2020

Penulis

DAFTAR ISI

| | Halaman |
|---|---------|
| HALAMAN SAMPUL | i |
| HALAMAN JUDUL | ii |
| HALAMAN PENGESAHAN | iii |
| PERNYATAAN | iv |
| ABSTRAK | v |
| <i>ABSTRACT</i> | vi |
| KATA PENGANTAR | vii |
| DAFTAR ISI | viii |
| DAFTAR GAMBAR | ix |
| DAFTAR TABEL | x |
| DAFTAR LAMPIRAN | xi |
| BAB I. PENDAHULUAN | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 1 |
| 1.3. Tujuan | 2 |
| 1.4. Batasan Masalah | 2 |
| 1.5. Sistematika Penulisan | 3 |
| BAB II. LANDASAN TEORI | 5 |
| 2.1. Penelitian Terdahulu | 5 |
| 2.2. <i>Chatbot</i> | 7 |
| 2.3. XAMPP | 7 |
| 2.4. MySQL | 8 |
| 2.5. PHP | 8 |
| 2.6. JavaScript | 9 |
| 2.7. TF-IDF | 9 |
| 2.8. <i>Cosine Similarity</i> | 10 |
| 2.9. <i>Recall</i> dan <i>Precision</i> | 10 |
| BAB III. METODOLOGI PENELITIAN..... | 12 |
| 3.1 Studi Literatur..... | 12 |

| | |
|---|-----------|
| 3.2 Identifikasi Masalah | 12 |
| 3.3 Metode Pengumpulan Data | 12 |
| 3.4 Metode Pengembangan Sistem..... | 12 |
| 3.4.1 Waterfall | 12 |
| 3.4.2 Bisnis Proses..... | 14 |
| 3.5 Metode Pengolahan Data..... | 16 |
| 3.6 Perhitungan Manual..... | 16 |
| BAB IV. ANALISIS DAN PERANCANGAN SISTEM..... | 21 |
| 4.1 Gambaran Umum Sistem..... | 21 |
| 4.2 Analisa Permasalahan..... | 22 |
| 4.3 Analisa Kebutuhan Non Fungsional..... | 22 |
| 4.4 Analisa Kebutuhan Fungsional..... | 23 |
| 4.5 Analisis Data..... | 23 |
| 4.6 Perancangan Antar Muka Pengguna..... | 44 |
| 4.7 Perancangan Basisdata | 40 |
| 4.8 Perancangan Sistem..... | 24 |
| BAB V. IMPLEMENTASI DAN PENGUJIAN | 47 |
| 5.1 Implementasi Basisdata | 47 |
| 5.2 Implementasi Anatarmuka Pengguna..... | 56 |
| 5.3 Implementasi Sistem..... | 47 |
| 5.4 Pengujian Fungsional | 59 |
| 5.5 Pengujian Akurasi..... | 65 |
| BAB VI. HASIL DAN PEMBAHASAN | 67 |
| BAB VII. KESIMPULAN DAN SARAN..... | 69 |
| 7.1 Kesimpulan..... | 69 |
| 7.2 Saran | 69 |
| DAFTAR PUSTAKA | 70 |
| LAMPIRAN - LAMPIRAN..... | 72 |

DAFTAR GAMBAR

| | Halaman |
|--|---------|
| Gambar 3.1 Waterfall Model | 13 |
| Gambar 3.2 Bisnis Proses Peminjaman | 13 |
| Gambar 3.3Bisnis Proses Pengembalian..... | 15 |
| Gambar 4.1Flowchart Alur Proses <i>Chatbot</i> | 21 |
| Gambar 4.2 Use Case | 24 |
| Gambar 4.3 Flowchart Pembobotan TF-IDF | 30 |
| Gambar 4.4 Flowchart Perhitungan Cosine Similarity | 30 |
| Gambar 4.5Context Diagram | 31 |
| Gambar 4.6DFD Level 1 | 32 |
| Gambar 4.7DFD Level 2 Proses 1 | 32 |
| Gambar 4.8DFD Level 2 Proses 2 | 33 |
| Gambar 4.9DFD Level 2 Proses 3 | 33 |
| Gambar 4.10DFD Level 2 Proses 4 | 34 |
| Gambar 4.11DFD Level 2 Proses 5 | 34 |
| Gambar 4.12DFD Level 2 Proses 6 | 35 |
| Gambar 4.13DFD Level 2 Proses 7 | 35 |
| Gambar 4.14DFD Level 2 Proses 8 | 36 |
| Gambar 4.15DFD Level 2 Proses 9 | 36 |
| Gambar 4.16Entitas Relationship Diagram..... | 40 |
| Gambar 4.17Mockup Halaman Awal Sistem..... | 44 |
| Gambar 4.18Mockup Halaman Chat..... | 44 |
| Gambar 4.19Mockup Halaman Login..... | 45 |
| Gambar 4.20Mockup Halaman Utama Operator | 45 |
| Gambar 4.21Mockup Halaman Data Jawaban..... | 45 |
| Gambar 4.22Mockup Halaman Perhitungan TF-IDF | 46 |
| Gambar 4.23Mockup Halaman Data Pertanyaan..... | 46 |
| Gambar 5.1Struktur Database | 47 |
| Gambar 5.2Tabel Operator..... | 47 |
| Gambar 5.3Tabel Jawaban | 48 |

| | |
|---|----|
| Gambar 5.4Tabel Index..... | 48 |
| Gambar 5.5Tabel Vektor..... | 48 |
| Gambar 5.6Tabel Pertanyaan | 48 |
| Gambar 5.7Tabel Stem..... | 48 |
| Gambar 5.8Tabel Barang | 49 |
| Gambar 5.9Tabel Pelanggan | 49 |
| Gambar 5.10Tabel Transaksi | 49 |
| Gambar 5.11Tabel Detail Transaksi | 50 |
| Gambar 5.12 Potongan Kode Menampilkan Data Master Jawaban | 50 |
| Gambar 5.13 Potongan Kode Preprocessing..... | 51 |
| Gambar 5.14 Potongan Kode Hitung Bobot TF-IDF..... | 52 |
| Gambar 5.15 Potongan Kode Hitung Panjang Vektor Dokumen | 53 |
| Gambar 5.16 Potongan Kode Hitung Similarity | 54 |
| Gambar 5.17 Potongan Kode Penentuan Respon <i>Chatbot</i> | 55 |
| Gambar 5.18Halaman Login..... | 56 |
| Gambar 5.19Halaman Dashboard | 57 |
| Gambar 5.20Halaman Master Jawaban..... | 57 |
| Gambar 5.21Halaman Perhitungan TF-IDF..... | 58 |
| Gambar 5.22Halaman Pertanyaan dari Pelanggan..... | 58 |
| Gambar 5.23Halaman Utama..... | 59 |
| Gambar 5.24Halaman <i>Chatbot</i> | 59 |
| Gambar 6.1 Contoh Perhitungan Bobot TF-IDF..... | 67 |

DAFTAR TABEL

| | Halaman |
|---|---------------|
| Tabel 2.1 Penelitian Terdahulu | 5 |
| Tabel 3.1 Contoh Respon <i>Chatbot</i> | 17 |
| Tabel 3.2 Perhitungan DF | 17 |
| Tabel 3.3 Perhitungan TF-IDF | 18 |
| Tabel 3.4 Perkalian Skalar | 18 |
| Tabel 3.5 Perkalian Vektor | 19 |
| Tabel 3.6 Hasil Perhitungan Similarity | 20 |
| Tabel 4.1 Spesifikasi Minimum Perangkat Lunak | 22 |
| Tabel 4.2 Spesifikasi Minimum Perangkat Keras | 23 |
| Tabel 4.3 Deskripsi Use Case | 25 |
| Tabel 4.4 Skenario Use Case Melakukan Login | 26 |
| Tabel 4.5 Skenario Use Case Mengelola Data Barang | 27 |
| Tabel 4.6 Skenario Use Case Mengelola Data Pelanggan | 27 |
| Tabel 4.7 Skenario Use Case Mengelola Data Transaksi | 28 |
| Tabel 4.8 Skenario Use Case Mengelola Data <i>Chatbot</i> | 28 |
| Tabel 4.9 Skenario Use Case Mengajukan Pertanyaan pada Sistem | 29 |
| Tabel 4.10 Skenario Use Case Mendapatkan Respon dari <i>Chatbot</i> | Error! |
| Bookmark not defined. | |
| Tabel 4.11 Spesifikasi proses login | 36 |
| Tabel 4.12 Spesifikasi proses kelola jawaban | 37 |
| Tabel 4.13 Spesifikasi proses kelola pertanyaan | 37 |
| Tabel 4.14 Spesifikasi kelola barang | 37 |
| Tabel 4.15 Spesifikasi kelola pelanggan | 38 |
| Tabel 4.16 Spesifikasi kelola transaksi | 38 |
| Tabel 4.17 Spesifikasi kelola pengaturan | 39 |
| Tabel 4.18 Spesifikasi kelola token | 39 |
| Tabel 4.19 Spesifikasi kelola chatting | 39 |
| Tabel 4.20 Tabel Admin | 40 |
| Tabel 4.21 Tabel Pertanyaan | 41 |
| Tabel 4.22 Tabel Pelanggan | 41 |

| | |
|---|----|
| Tabel 4.23Tabel Detail..... | 41 |
| Tabel 4.24Tabel Jawaban..... | 42 |
| Tabel 4.25Tabel Index | 42 |
| Tabel 4.26Tabel Barang | 42 |
| Tabel 4.27Tabel Stem | 43 |
| Tabel 4.28Tabel Transaksi | 43 |
| Tabel 4.29Tabel Vektor | 44 |
| Tabel 5.1 Pengujian Fungsional Halaman Login Admin..... | 60 |
| Tabel 5.2 Pengujian Fungsional Menu Operator | 60 |
| Tabel 5.3 Pengujian Fungsional Operator Data Barang..... | 61 |
| Tabel 5.4 Pengujian Fungsional Operator Data Pelanggan | 62 |
| Tabel 5.5 Pengujian Fungsional Operator Data Transaksi | 62 |
| Tabel 5.6 Pengujian Fungsional Operator Data Jawaban | 63 |
| Tabel 5.7 Pengujian Fungsional Operator Hasil Perhitungan..... | 63 |
| Tabel 5.8 Pengujian Fungsional Operator Menu History Jawaban..... | 64 |
| Tabel 5.9 Pengujian Fungsional User <i>Chatbot</i> | 64 |
| Tabel 5.10 Tabel Pengujian Akurasi..... | 65 |
| Tabel 6.1 Hasil Penghitungan Cosine Similarity | 68 |

DAFTAR LAMPIRAN

Lampiran 1 Source Code Tampilan *Chatbot*

Lampiran 2 Source Code Perhitungan *Chatbot*

BAB I. PENDAHULUAN

1.1 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 operator merespon pertanyaan dari pelanggan yang membutuhkan informasi mengenai penyewaan scaffolding. Dikarenakan operator memiliki jam kerja yang ditentukan oleh peraturan perusahaan. Sehingga pelanggan sulit berkomunikasi dengan Operator diluar jam kerja operator. Sedangkan pelanggan membutuhkan respon yang cepat untuk menangani kendala pada perusahaannya yang membutuhkan jasa dari CV. Scaffolding Samarinda.

TF-IDF (*Term Frequency - Inverse Document Frequency*) merupakan metode algoritma yang menentukan frekuensi dari kemunculan sebuah term dalam dokumen yang bersangkutan. Metode ini untuk menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap kata di setiap dokumen. Semakin besar jumlah kemunculan suatu term (TF tinggi) dalam dokumen, semakin besar pula bobotnya atau akan memberikan nilai kesesuaian yang semakin besar. (informatikalogi.com).

Sehingga pada penelitian ini, penulis membuat Aplikasi *Chatbot* Pada Sistem Informasi Scaffolding 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.

1.2 Rumusan Masalah

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

Apakah penerapan *chatbot* dengan metode N-gram, TF-IDF, dan Cosine Similarity dapat meringankan beban kerja Operator CV. Scaffolding Samarinda dalam melayani memberikan informasi kepada Pelanggan setiap saat dalam keterbatasan waktu jam kerja perusahaan?

1.3 Tujuan

Tujuan dari pembuatan sistem *chatbot* ini adalah sebagai berikut :

Untuk meringankan beban kerja Operator CV. Scaffolding Samarinda dalam melayani pemberian informasi kepada Pelanggan setiap saat dalam keterbatasan waktu jam kerja perusahaan.

1.4 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 :

- a. Informasi mengenai barang yang tersedia
- b. Informasi mengenai harga sewa barang per unit
- c. Informasi mengenai jangka waktu peminjaman barang
- d. Informasi mengenai berat barang per unit

1.5 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini disusun untuk memberikan gambaran umum tentang penelitian yang dilakukan. Sistematika penulisan skripsi ini adalah sebagai 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 berguna dalam proses analisis permasalahan.

BAB III. METODOLOGI 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

anatomika 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 penulisan skripsi.

BAB II. LANDASAN TEORI

2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama dengan judul penelitian penulis. Namun, penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Tabel 2.1 Penelitian Terdahulu

| Nama Peneliti | Judul Penelitian | Hasil Penelitian |
|---|---|---|
| Suryani, Dhebys & Larasati Amalia, Eka, 2017. | Aplikasi <i>Chatbot</i> Objek Wisata Jawa Timur Berbasis AIML | Aplikasi <i>Chatbot</i> 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 <i>Chatbot</i> | Sistem <i>chatbot</i> 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 <i>Chatbot</i> Berbasis LINE untuk Layanan Publik Kesehatan Kota Malang | Question-Answering dalam bentuk <i>chatbot</i> 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 <i>Chatbot</i> Konsultasi Laktasi. | Diterapkannya metode Levenshtein Distance dan TF-IDF dan Cosine Similarity membuat aplikasi <i>chatbot</i> layak digunakan untuk customer service pusat laktasi |
| Melita, Ria & Dirjam, Taslimun, 2018. | Penerapan Metode Term Frequency – Inverse | Metode TF-IDF dan Cosine Similarity |

| | | |
|--|---|---|
| | Document Frequency (TF-IDF) dan Cosine Similarity pada Sistem Temu Kembali Informasi untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus : Syarah Umdatil Ahkam) | berhasil diterapkan dengan memberikan hasil berupa output dokumen, yaitu syarah hadits sesuai dengan query yang di input-kan |
| Kavitha B. R. dan Dr, Cethana R, Murthy, 2019 | <i>Chatbot</i> for healthcare system using Artificial Intelligence | Kombinasi antara TF-IDF dan Cosine Similarity memberikan hasil respon <i>chatbot</i> yang sesuai. |
| Natadian Astuti, Rani & Fatchan, Muhammad, 2019. | Perancangan Aplikasi <i>Chatbot</i> untuk Industri Komersial 4.0 | Adanya aplikasi <i>chatbot</i> membuat peran dari customer service menjadi lebih efektif karena dapat melayani pertanyaan dari pelanggan 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 healthcare system using Artificial Intelligence*” sebagai referensi utama penulis dalam melakukan penelitian. Hal ini dikarenakan adanya persamaan metode yang digunakan oleh penulis, yaitu TF-IDF dan *Cosine Similarity*, sebagai metode utama dalam pembuatan aplikasi *chatbot*.

2.2 Chatbot

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

Chatbot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunaanya 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 pelanggan dalam menanyakan pertanyaan yang diinginkan dan akan langsung direspon dengan cepat oleh sistem.

2.3 XAMPP

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

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

2.4 MySQL

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

MySQL adalah adalah multi user database yang menggunakan bahasa *StructuredQuery Language (SQL)*. MySQL dalam operasi *client server* melibatkan *serverdaemon* MySQL disisi server dan berbagai macam program serta *library* yang berjalan disisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TEX, mengaku mampu menyimpan data lebih dari 40 database, 10.000 tabel, dan sekitar 7.000.000 baris. Totalnya kurang lebih 100 *Gigabyte* data. MySQL digunakan untuk pembuatan database dan hasil pembobotan dari semua sistem yang dijalankan dalam sistem yang dibuat.

2.5 PHP

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang dapat digunakan untuk tujuan umum. *PHP* lebih populer digunakan untuk pengembangan aplikasi web. Dalam proses pembuatan halaman web, *PHP* tidak memerlukan kode yang panjang seperti pada *Perl* dan *Python* karena kode *PHP*

dapat disisipkan di dalam kode *HTML*. *PHP* dapat dijalankan dalam sebagian besar sistem operasi, termasuk *linux*, varian-varian *UNIX* (*HP-UX*, *OpenBSD*), *Windows*, dan *Mac OS X*. *PHP* juga mendukung sebagian besar server web yang ada saat ini, seperti : *Apache*, *IIS*, *nginx*, dan *lighttpd*. Bahasa yang digunakan dalam *CodeIgniter* menggunakan *PHP* dimana bahasa pemrograman *PHP* digunakan sebagai bahasa pemrograman umum.

2.6 JavaScript

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

2.7 TF-IDF

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

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

$$tf = tf_{ij} \quad (2.7.1)$$

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

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

- c. Jenis formula *TF* yang biasa digunakan untuk perhitungan adalah *TF* murni (*raw TF*). Dengan demikian rumus umum untuk *Term*

Weighting TFIDF adalah penggabungan dari formula perhitungan raw TF dengan formula IDF dengan cara mengalikan nilai TF dengan nilai IDF:

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

2.8 Cosine Similarity

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

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

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

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

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

n = Jumlah term yang unik dalam data set

2.9 Recall dan Precision

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

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

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

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

Sedangkan nilai *relatif Recall* dihitung dengan rumus :

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

Kedua ukuran diatas biasanya diberi nilai dalam bentuk persentase, 1 sampai 100%. Sebuah sistem informasi akan dianggap baik jikatingkat *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

3.1 Studi Literatur

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

- *Chatbot*
- TF-IDF
- Cosine Similarity
- Scaffolding

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

3.2 Identifikasi Masalah

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

3.3 Metode Pengumpulan Data

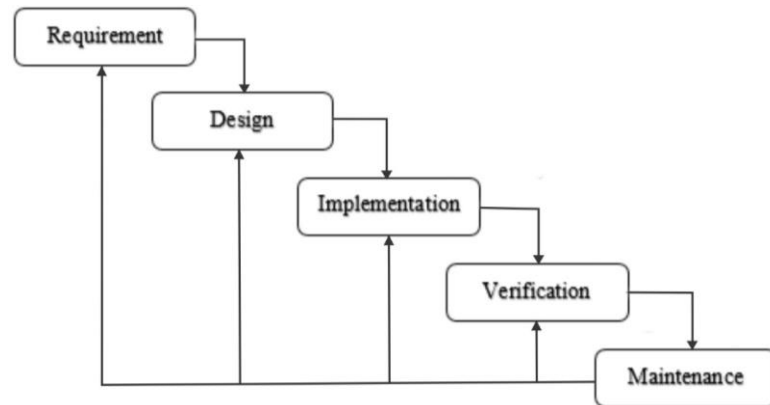
Metode yang digunakan dalam pengambilan data adalah melalui wawancara. Wawancara dilakukan menggunakan media *Whatsapp* dengan operator penyewaan scaffolding. Wawancara dilakukan pada bulan November 2019 hingga bulan maret. Hasil wawancara yang dilakukan adalah didapatkan data barang yang disewakan,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

3.4 Metode Pengembangan Sistem

3.4.1 Waterfall

Metode yang digunakan dalam perancangan “Aplikasi *Chatbot* pada Sistem Informasi Penyewaan Scaffolding Menggunakan Metode TF-IDF” adalah System Development Life Cycle (SDLC) dengan waterfall model.

Waterfall model merupakan bentuk umum yang digunakan dalam perancangan sebuah sistem karena dalam setiap tahapan yang dilakukan harus diselesaikan sebelum menuju tahap selanjutnya sehingga tahapan dilakukan secara berurutan dan mendapatkan hasil yang maksimal. Tahapan-tahapan yang dilakukan yaitu :



Gambar 3.1 Waterfall Model

a. Requirement

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

b. Design

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

c. Implementation

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

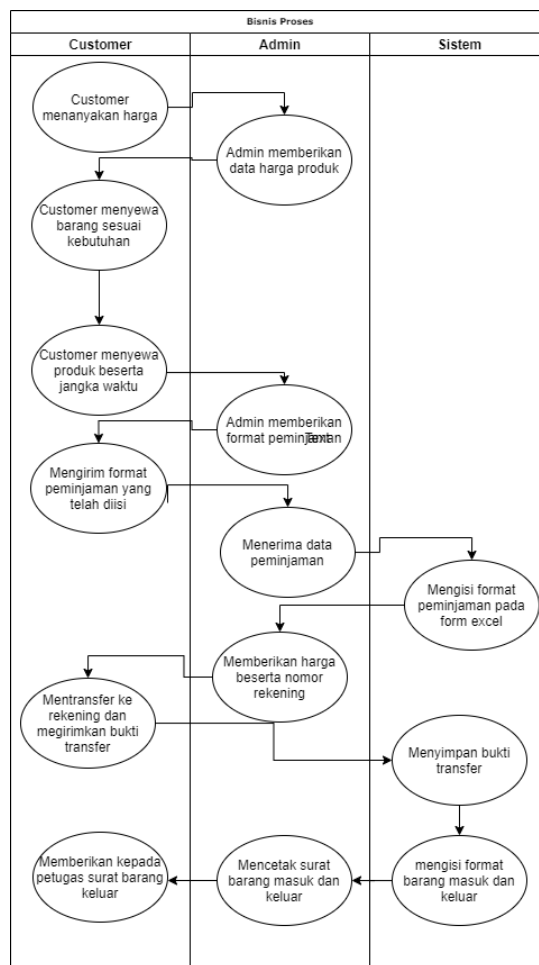
d. Verification

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

e. Maintenance

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

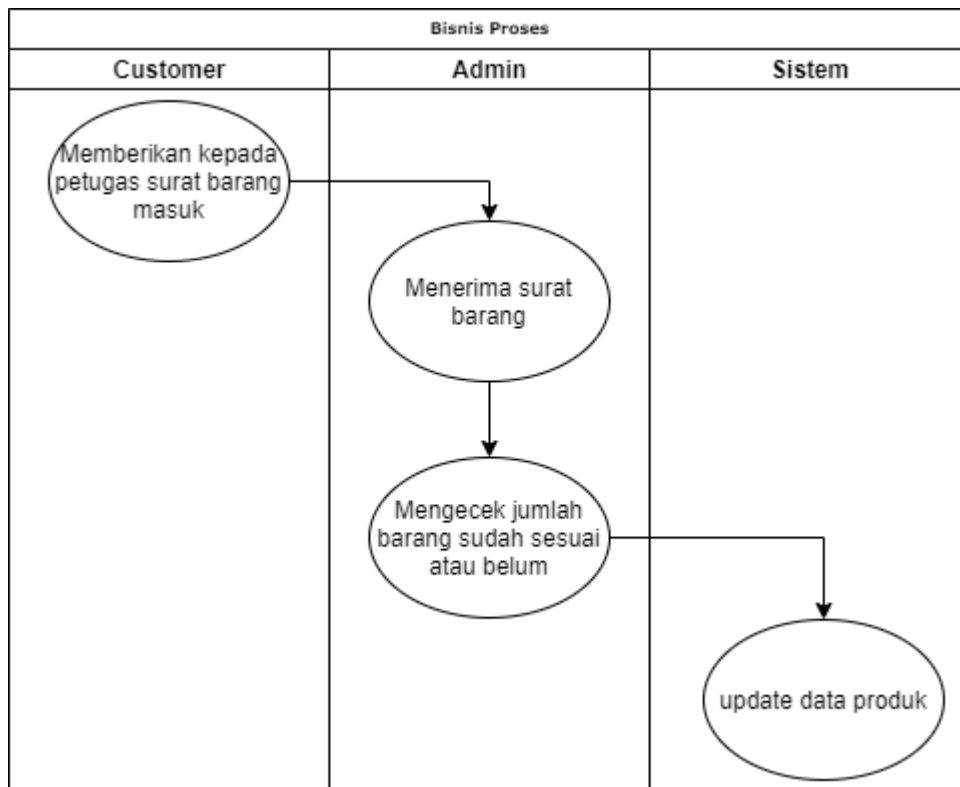
3.4.2 Bisnis Proses



Gambar 3.2Bisnis Proses Peminjaman

a. Deskripsi Bisnis Proses

Pelanggan menghubungi operator melalui *WhatsApp* untuk menanyakan harga, operator memberikan data harga. Pelanggan akan menyewa barang dengan jangka waktu dan mencantumkan nama barang, operator memberikan format peminjaman, pelanggan mengisi format peminjaman dan memberikan foto ktp, operator mencatat dan memberikan harga yang telah ditentukan. Pelanggan membayar melalui atm/membayar langsung, menunjukkan bukti pembayaran, menyimpan bukti pembayaran. Operator mengirimkan barang dan memberikan surat keluar dan surat masuk. Surat keluar diberikan kepada petugas.



Gambar 3.3Bisnis Proses Pengembalian

b. Deskripsi Bisnis Proses

Pelanggan mengembalikan barang dengan memberikan surat masuk, operator menerima surat barang masuk, mengecek kelengkapan barang apakah sesuai dengan jumlah peminjaman. Kemudian operator mengupdate data barang dengan menambahkan barang yang telah dikembalikan oleh pelanggan.

3.5 Metode Pengolahan Data

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

1. *Case Folding*

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

2. *Tokenizing*

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

3. *Filtering*

Filtering merupakan tahapan untuk menghapus setiap kata-kata yang tidak penting atau kata-kata yang tidak memiliki arti. Proses *filtering* biasa disebut *stop-words removal*. Contoh kata-kata yang dikategorikan sebagai *stop-words* yaitu: dan, dari, di, dan sebagainya.

4. *Stemming*

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

3.6 Perhitungan Manual

TF-IDF adalah proses pembobotan kalimat dengan menjumlahkan nilai setiap *token* per kalimat untuk masalah pencarian informasi. Ide pokok dalam TF-IDF adalah menghitung *weight* setiap kalimat untuk digunakan pada tahap selanjutnya. Setelah menghitung bobot kalimat, kalimat tersebut

diurutkan dari bobot yang tertinggi. Sebagai contoh data respon *chatbot* yang akan digunakan dalam perhitungan TF-IDF ini termuat pada tabel 3.1

Tabel 3.1 Contoh Respon *Chatbot*

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

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

Tabel 3.2 Perhitungan DF

| Dokumen | q | d1 | d2 | d3 | d4 | d5 | df | log(d/df) |
|-------------|---|----|----|----|----|----|----|-----------|
| Harga | 1 | 0 | 1 | 1 | 1 | 0 | 4 | 1,5 |
| Sewa | 1 | 1 | 1 | 1 | 1 | 0 | 5 | 1 |
| Scaffolding | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 6 |
| Buka | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 6 |
| Pukul | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 6 |
| Pagi | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 6 |
| Main frame | 0 | 0 | 1 | 1 | 1 | 0 | 3 | 2 |
| Untuk | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 6 |
| bulan | 0 | 0 | 0 | 1 | 1 | 1 | 3 | 2 |
| 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 3.5Perkalian 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

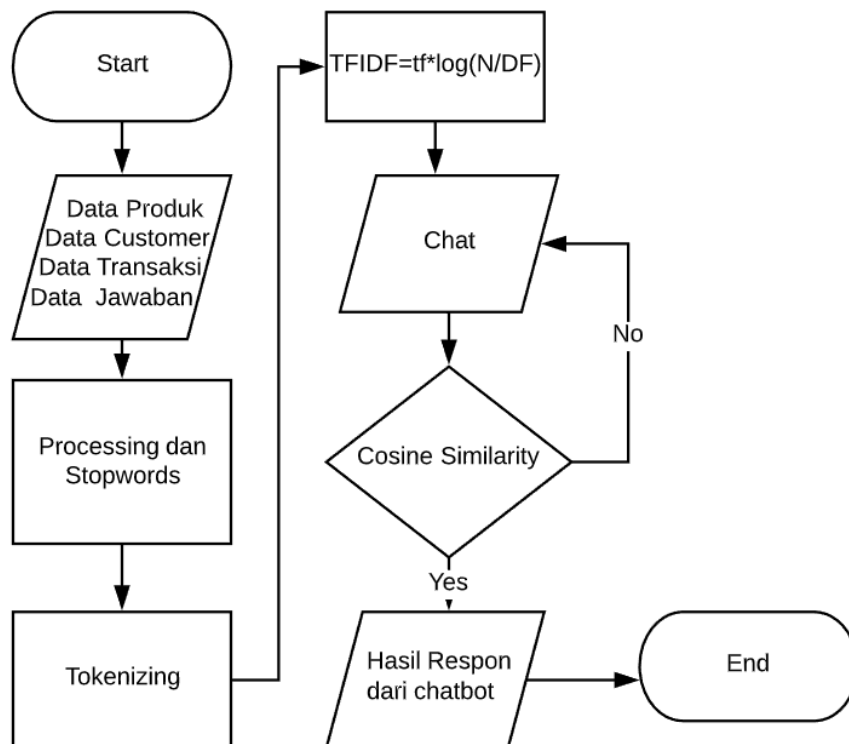
| Kode | Dokumen | Nilai Similarity | Persentase |
|------|--|------------------|------------|
| d1 | Penyewaan buka pada pukul 8 pagi | 0,005792 | 0,58% |
| d2 | Harga sewa main frame 1,9 Rp55.000 untuk sebulan | 0,032241 | 3,22% |
| d3 | Harga sewa main frame 1,9 Rp45.000 untuk 2 minggu | 0,093118 | 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

4.1 Gambaran Umum Sistem

Sistem *chatbot* berbasis *web* berdasarkan data barang dari CV Penyewaan digunakan untuk memberikan respon terhadap pertanyaan pelanggan 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 pelanggan. 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 pelanggan dengan data jawaban yang ada pada sistem. Sistem *chatbot* ini menggunakan bahasa pemrograman PHP dan *MySQL* sebagai *database*-nya. Dengan adanya sistem ini, diharapkan pelanggan tidak akan merasa kesulitan dalam mencari informasi seputar layanan dan barang yang ditawarkan oleh CV Penyewaan. Adapun alur aplikasi *chatbot* ini dijelaskan pada gambar 4.1



Gambar 4.1 Flowchart Alur Proses *Chatbot*

4.2 Analisa Permasalahan

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

4.3 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:

4.3.1 Perangkat Lunak (*Software*)

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

Tabel 4.1 Spesifikasi Minimum Perangkat Lunak

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

4.3.2 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

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

4.4 Analisa Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berisi proses yang dilakukan oleh sistem. Dalam hal ini pelanggan 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 pelanggan.
3. Sistem dapat melakukan perhitungan TF-IDF.
4. Sistem dapat melakukan perhitungan *Cosine Similarity*.
5. Sistem dapat memberikan respon jawaban berdasarkan apa yang telah di-*input*-kan pelanggan.

4.5 Analisis Data

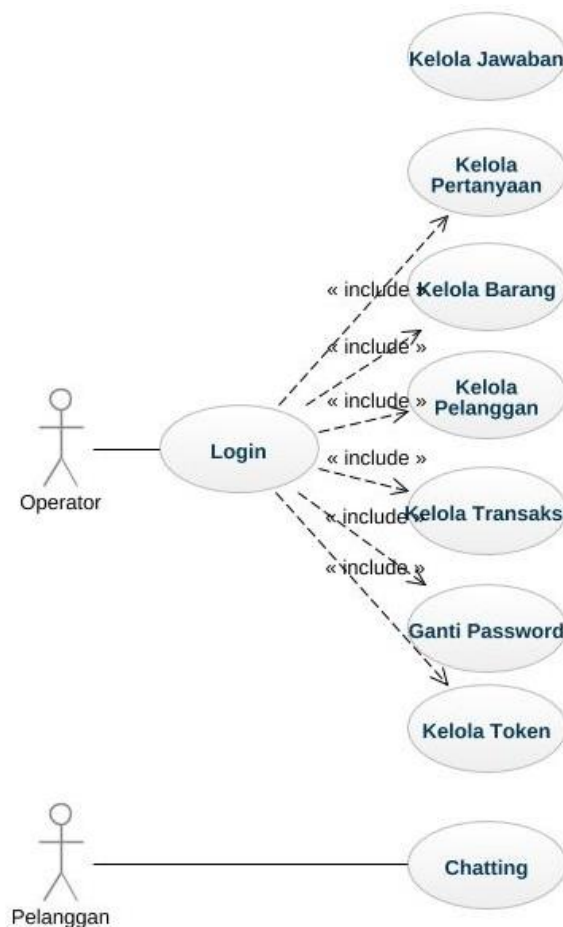
Analisis data adalah proses pengolahan dan penghitungan dari seluruh data yang sudah diperoleh dari penelitian yang telah dilakukan secara sistematis. Analisis data yang digunakan pada penelitian ini untuk menguji variabel yang ada adalah penghitungan bobot tiap kata dan menghitung kemiripan kata untuk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

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.

4.6 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. | <i>Use Case</i> | Deskripsi |
|-----|-----------------|---|
| 1 | Melakukan login | Melakukan login untuk dapat mengakses sistem |
| 2 | Data Jawaban | Menambah, mengubah dan menghapus data Jawaban |
| 3 | Data Pertanyaan | Menghapus data Pertanyaan |
| 4 | Data Barang | Menambah, mengubah dan menghapus data barang |
| 5 | Data Pelanggan | Menambah, mengubah dan menghapus data pelanggan |
| 6 | Data Transaksi | Menambah, mengubah dan menghapus data transaksi |
| 7 | Ganti Password | Mengganti Password untuk operator |
| 8 | Kelola Token | Menghitung TF-IDF dan Cosine Similarity |
| 9 | Chatting | Menuliskan pertanyaan dan mendapatkan jawaban dari sistem |

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 <i>login</i> |
| Aktor | Operator |
| Tujuan | Operator bisa masuk ke dalam sistem |
| Kondisi Awal | - |
| Skenario | <ul style="list-style-type: none"> a. Operator mengisi username dan password b. Jika sesuai, tampil halaman <i>dashboard</i> sistem c. Jika tidak sesuai, tampil notifikasi username dan password tidak valid |
| Invariant 1 | - |

2. Mengelola Data Jawaban

Tabel 4.5 Skenario *Use Case* Mengelola Data Jawaban

| | |
|----------------------|---|
| Nama Use Case | Mengelola data jawaban |
| Aktor | Operator |
| Tujuan | Operator dapat menambah, mengubah dan menghapus data jawaban |
| Kondisi Awal | Operator sudah login |
| Skenario | <ul style="list-style-type: none"> a. Operator melihat data jawaban b. Operator menambah data jawaban c. Operator menyimpan data jawaban |
| Invariant 1 | Operator mengubah data jawaban |
| Invariant 2 | Operator menghapus data jawaban |

3. Mengelola Data Pertanyaan

Tabel 4.6 Skenario *Use Case* Mengelola Data Pertanyaan

| | |
|----------------------|---|
| Nama Use Case | Mengelola data pertanyaan |
| Aktor | Operator, Pelanggan |
| Tujuan | Operator dapat menghapus data pertanyaan. |
| Kondisi Awal | Operator sudah login |
| Skenario | <ul style="list-style-type: none"> a. Operator melihat data Pertanyaan |

| | |
|--------------------|------------------------------------|
| Invariant 1 | Operator menghapus data pertanyaan |
|--------------------|------------------------------------|

4. Mengelola Data Barang

Tabel 4.7 Skenario *Use Case* Mengelola Data Barang

| | |
|----------------------|--|
| Nama Use Case | Mengelola data barang |
| Aktor | Operator |
| Tujuan | Operator dapat menambah, mengubah dan menghapus data barang |
| Kondisi Awal | Operator sudah login |
| Skenario | <ul style="list-style-type: none"> b. Operator melihat data barang c. Operator menambah data barang d. Operator menyimpan data barang |
| Invariant 1 | Operator mengubah data barang |
| Invariant 2 | Operator menghapus data barang |

5. Mengelola Data Pelanggan

Tabel 4.8 Skenario *Use Case* Mengelola Data Pelanggan

| | |
|----------------------|---|
| Nama Use Case | Mengelola data pelanggan |
| Aktor | Operator |
| Tujuan | Operator dapat menambah, mengubah dan menghapus data pelanggan |
| Kondisi Awal | Operator sudah login |
| Skenario | <ul style="list-style-type: none"> a. Operator melihat data pelanggan b. Operator menambah data pelanggan c. Operator menyimpan data pelanggan |
| Invariant 1 | Operator mengubah data pelanggan |
| Invariant 2 | Operator menghapus data pelanggan |

6. Mengelola Data Transaksi

Tabel 4.9 Skenario *Use Case* Mengelola Data Transaksi

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

7. Mengelola Token

Tabel 4.10 Skenario *Use Case* Mengelola Hitung TF-IDF

| | |
|----------------------|---|
| Nama Use Case | Mengelola data <i>chatbot</i> |
| Aktor | Sistem |
| Tujuan | Sistem dapat menampilkan perhitungan data <i>chatbot</i> |
| Kondisi Awal | Operator sudah login |
| Skenario | a. Operator melihat data TF-IDF b. Operator menyimpan data TF-IDF c. Operator melakukan perhitungan TF-IDF pada data d. Operator melihat hasil perhitungan data TF-IDF |
| Invariant 1 | Menampilkan hasil vektor kalimat jawaban |
| Invariant 2 | - |

8. Pengaturan Ganti Password

Tabel 4.11 Skenario *Use Case* Pengaturan Ganti Password

| | |
|----------------------|--|
| Nama Use Case | Pengaturan Ganti Password |
| Aktor | Operator |
| Tujuan | Operator dapat mengganti password |
| Kondisi Awal | Operator sudah login |
| Skenario | a. Operator mengganti password |
| Invariant 1 | Menampilkan pemberitahuan Password berhasil diganti. |
| Invariant 2 | - |

9. Mengajukan pertanyaan pada sistem

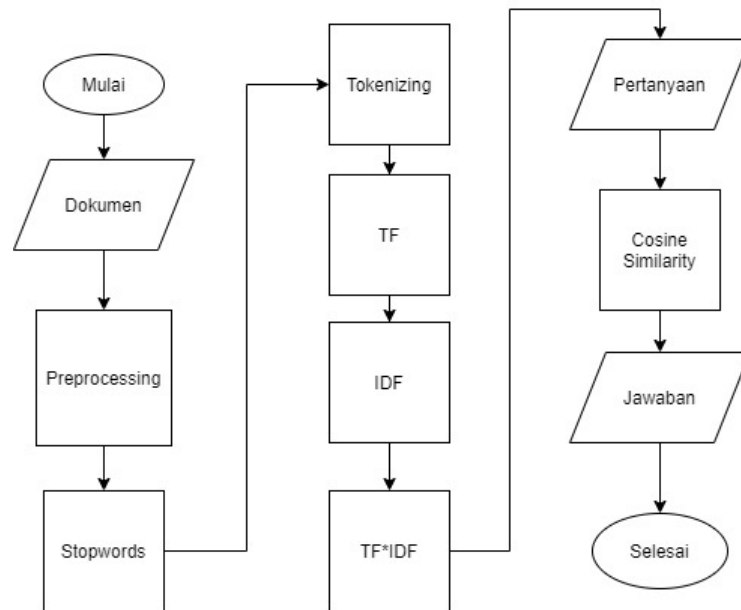
Tabel 4.12 Skenario *Use Case* Mengajukan Pertanyaan pada Sistem

| | |
|----------------------|--|
| Nama Use Case | Mengajukan pertanyaan pada sistem |
| Aktor | <i>User</i> |
| Tujuan | <i>User</i> dapat melakukan <i>input</i> pertanyaan pada sistem |
| Kondisi Awal | - |
| Skenario | a. <i>User</i> mengisikan pertanyaan pada form <i>input chatbot</i> b. <i>User</i> menekan tombol <i>submit</i> |
| Invariant 1 | - |

4.5.2 Flowchart

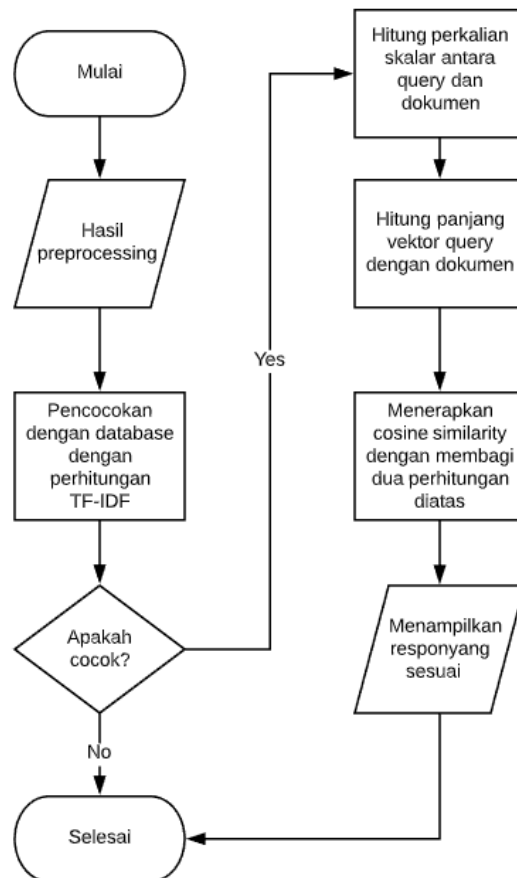
Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Dalam sistem ini dijabarkan *flowchart* proses pembobotan TF-IDF dan perhitungan *Cosine Similarity*.

4.5.2.1 Flowchart Pembobotan TF-IDF



Gambar 4.3 Flowchart Pembobotan TF-IDF

4.5.2.2 Flowchart Perhitungan Cosine Similarity



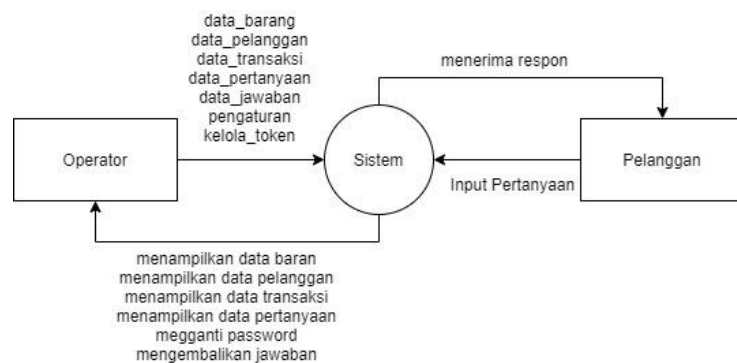
Gambar 4.4 Flowchart Perhitungan Cosine Similarity

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 penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas.

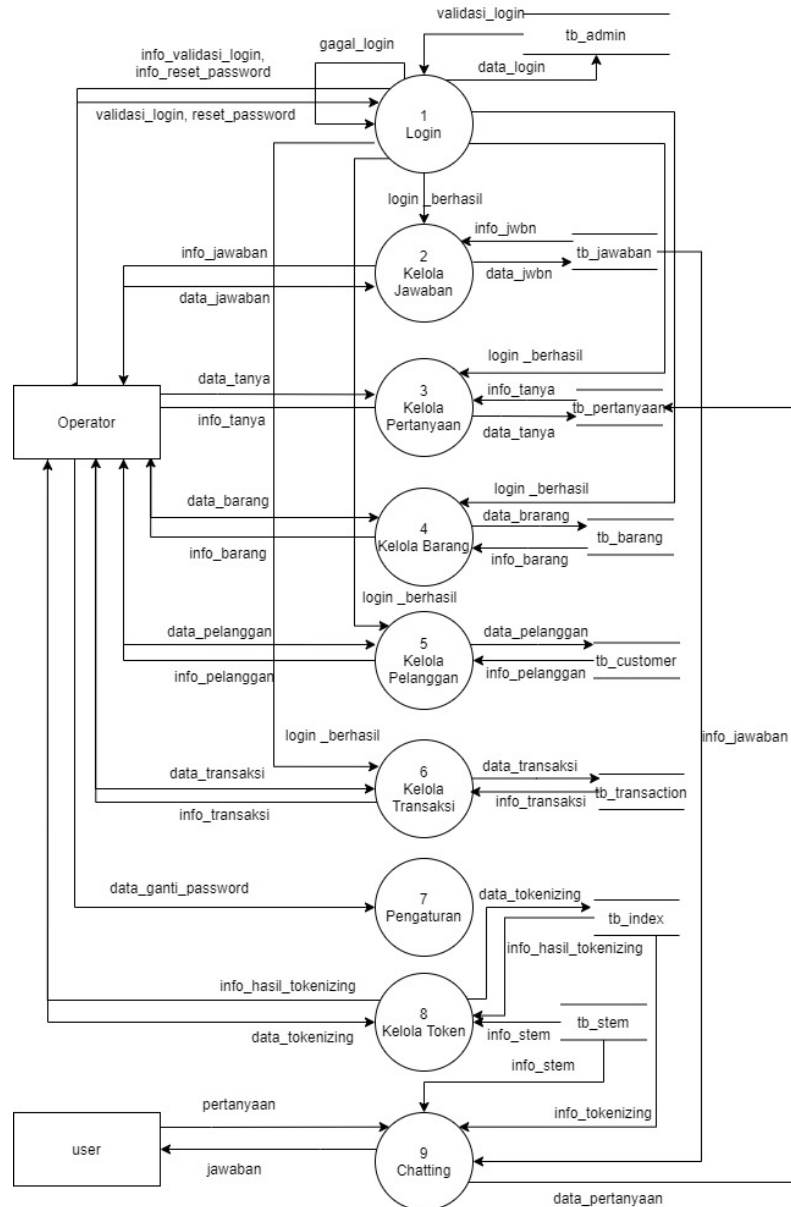
4.5.3.1 Context Diagram

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

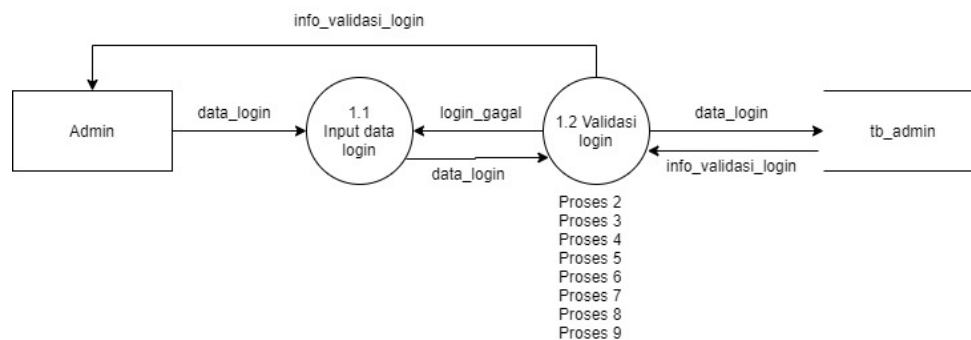
4.5.3.2 DFD Level 1



Gambar 4.6 DFD Level 1

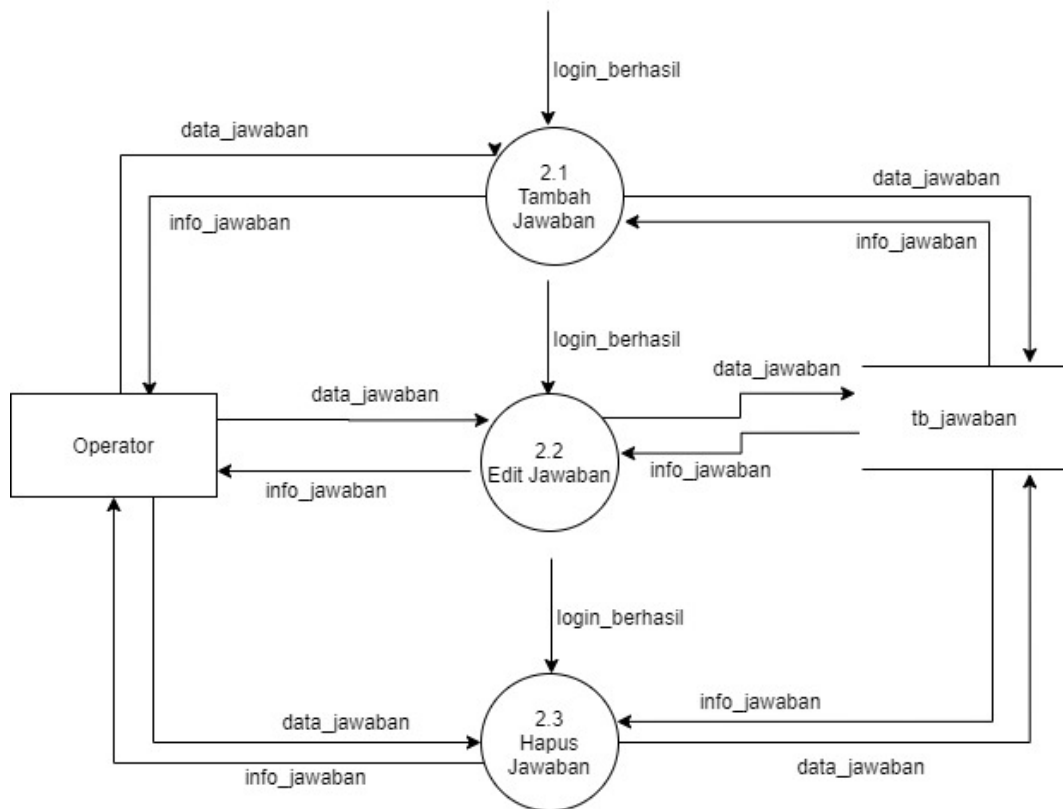
4.5.3.3 DFD Level 2

1. DFD Level 2 Proses 1



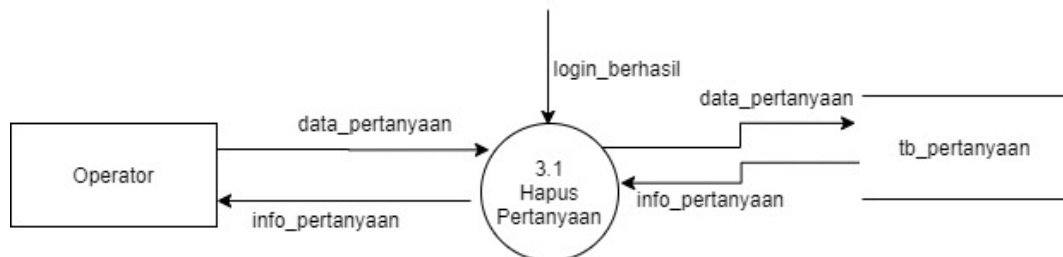
Gambar 4.7 DFD Level 2 Proses 1

2. DFD Level 2 Proses 2



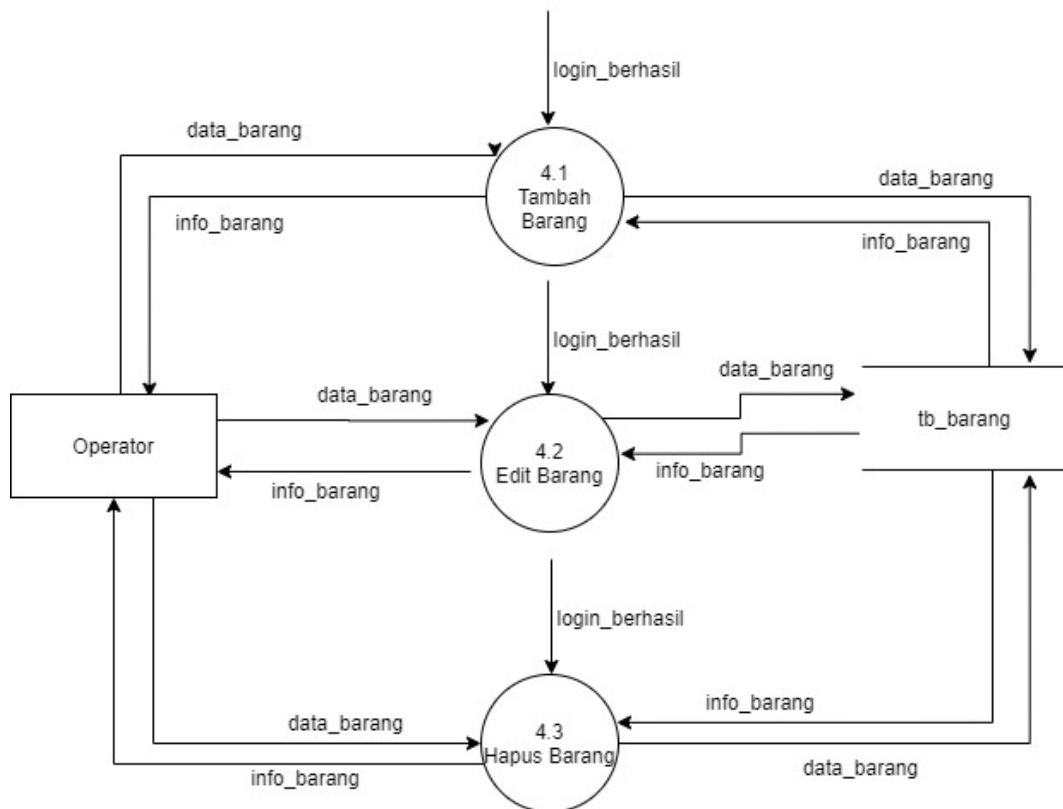
Gambar 4.8 DFD Level 2 Proses 2

3. DFD Level 2 Proses 3



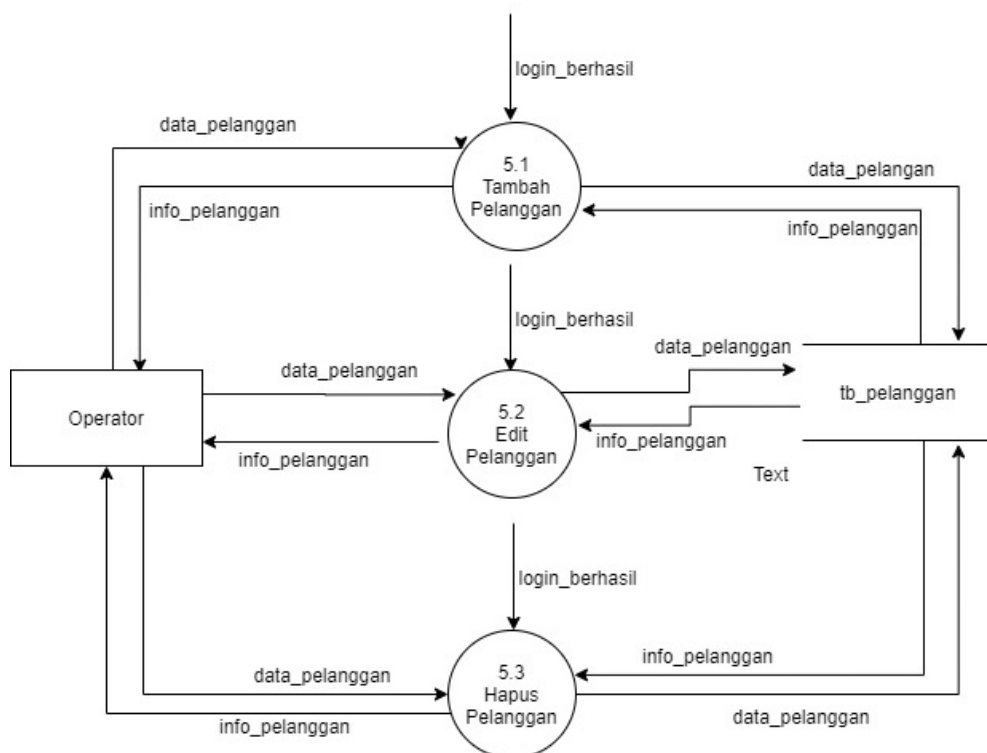
Gambar 4.9 DFD Level 2 Proses 3

4. DFD Level 2 Proses 4



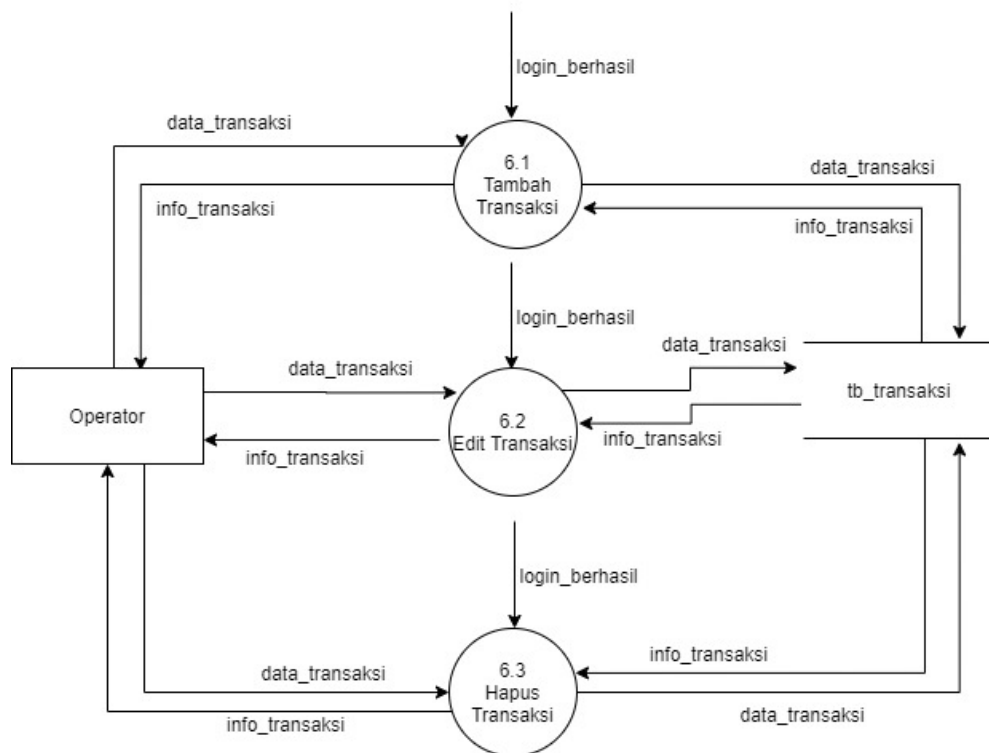
Gambar 4.10 DFD Level 2 Proses 4

5. DFD Level 2 Proses 5



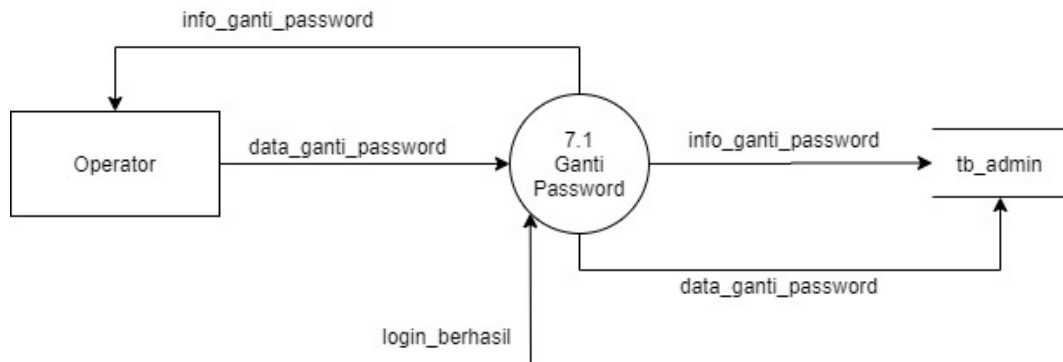
Gambar 4.11 DFD Level 2 Proses 5

6. DFD Level 2 Proses 6



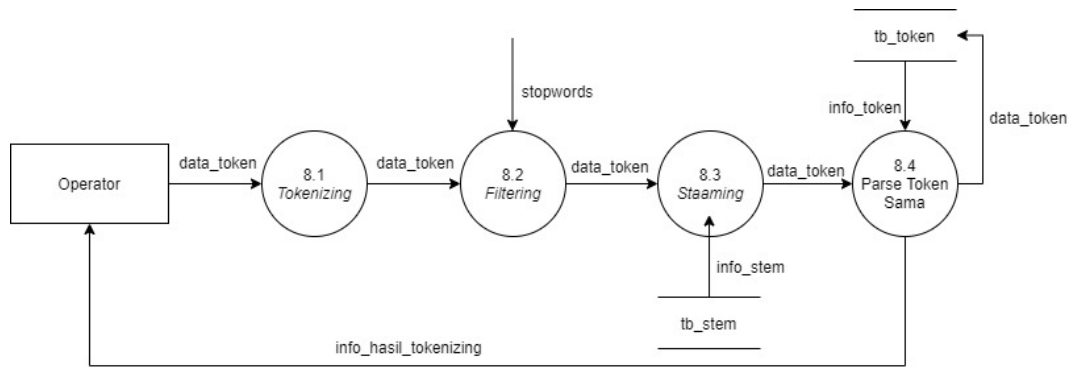
Gambar 4.12 DFD Level 2 Proses 6

7. DFD Level 2 Proses 7



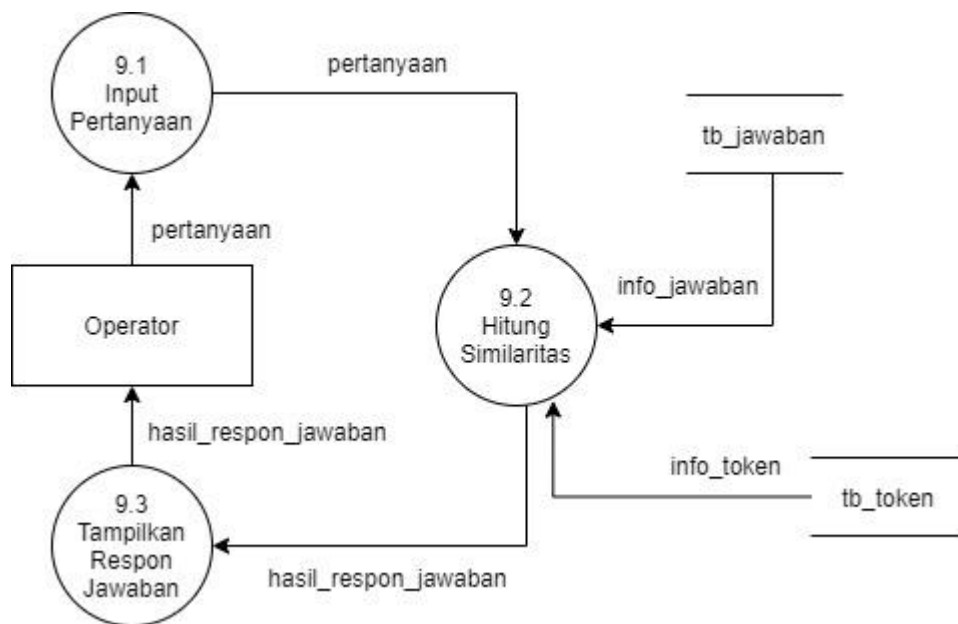
Gambar 4.13 DFD Level 2 Proses 7

8. DFD Level 2 Proses 8



Gambar 4.14DFD Level 2 Proses 8

9. DFD Level 2 Proses 9



Gambar 4.15DFD Level 2 Proses 9

4.5.4 Spesifikasi Proses

Tabel 4.13 Spesifikasi proses login

| | |
|---------------|--|
| No. Proses | 1 |
| Nama | Login |
| Deskripsi | Proses yang digunakan ketika akan mengakses halaman operator <i>chatbot</i> . |
| Data Masuk | data_login |
| Data Keluar | info_validasi_login, login_berhasil, login_gagal |
| Logika Proses | <ol style="list-style-type: none"> 1. Jika pengguna sudah login sistem akan menampilkan layar <i>dashboard</i>. 2. Jika belum, lanjut ke no. 3. 3. Sistem menampilkan halaman login. 4. Pengguna mengisi data login (<i>username</i> dan <i>password</i>). |

| | |
|--|---|
| | <ol style="list-style-type: none"> 5. Setelah pengguna tekan tombol login, sistem akan memeriksa data login ke <i>database</i> tabel admin. 6. Jika data ditemukan dan <i>username/password</i> benar, maka sistem akan menampilkan layar utama dari halaman operator yang berisi menu-menu untuk mengakses proses-proses 2,3,4,5,6,7,8. 7. Jika data tidak ditemukan atau <i>username/password</i> salah, maka sistem akan menampilkan pesan gagal login dan sistem akan menampilkan halaman login kembali. |
|--|---|

Tabel 4.14 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 | <ol style="list-style-type: none"> 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 4.15 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 | <ol style="list-style-type: none"> 1. Dalam halaman kelola pertanyaan terdapat proses hapus pertanyaan. 2. Hapus jawaban ditekan akan menampilkan konfirmasi penghapusan data jawaban. |

Tabel 4.16 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 | <ol style="list-style-type: none"> 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 4.17 Spesifikasi kelola pelanggan

| | |
|---------------|--|
| No. Proses | 5 |
| Nama | Kelola Pelanggan |
| Deskripsi | Proses yang digunakan untuk mengelola pelanggan dari penyewaan. |
| Data Masuk | data_pelanggan, login_berhasil |
| Data Keluar | Info_pelanggan |
| Logika Proses | <ol style="list-style-type: none"> 1. Dalam halaman kelola pelanggan terdapat proses tambah pelanggan, edit pelanggan, dan hapus pelanggan. 2. Tambah pelanggan ditekan akan menampilkan form tambah pelanggan. 3. Edit pelanggan ditekan akan menampilkan form edit pelanggan untuk pelanggan yang dimaksud. 4. Hapus pelanggan ditekan akan menampilkan konfirmasi penghapusan data pelanggan. |

Tabel 4.18 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 | <ol style="list-style-type: none"> 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 4.19 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 4.20 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 | <ol style="list-style-type: none"> 1. Dalam halaman kelola token terdapat proses tokenizing, filtering, steaming, parse token sama. 2. Proses <i>tokenizing</i> untuk memotong <i>string input</i> berdasarkan kata yang tersusun. 3. Proses <i>filtering</i> 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 (<i>root word</i>). |

Tabel 4.21 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 | <ol style="list-style-type: none"> 1. Dalam halaman kelola chatting terdapat proses input pertanyaan, hitung similaritas, dan menampilkan hasil jawaban. 2. Saat pelanggan masuk ke dalam <i>chatbot</i>, pelanggan mengetikkan pertanyaan. 3. Sistem menghitung kemiripan kata pertanyaan dengan jawaban yang diinginkan. 4. Sistem menampilkan jawaban dengan kemiripan tinggi. |

4.7 Perancangan Basisdata



Gambar 4.16 *Entitas Relationship Diagram*

Tabel Admin

Nama Tabel : tb_admin

Deskripsi : Berisi data operator

Tabel 4.22Tabel 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 4.23Tabel 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 Pelanggan

Nama Tabel : tb_pelanggan

Deskripsi : Berisi data pelanggan

Tabel 4.24Tabel Pelanggan

| 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 4.25Tabel 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 4.26Tabel 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 4.27Tabel 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 4.28Tabel 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 4.29Tabel 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 4.30Tabel 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 4.31 Tabel Vektor

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

4.8 Perancangan Antar Muka Pengguna

a. Halaman awal

| | |
|-----------------------|----------------------------------|
| PENYEWAAN SCAFFOLDING | HOME ABOUT CONTACT CHATBOT LOGIN |
| HOME | |
| ABOUT | |
| CONTACT | |

Gambar 4.17 Mockup Halaman Awal Sistem

b. Halaman Chat

| | |
|--|----------------------------------|
| PENYEWAAN SCAFFOLDING | HOME ABOUT CONTACT CHATBOT LOGIN |
| <div>SILAHKAN BERTANYA <input type="button" value="SUBMIT"/></div> <div>RESPON CHATBOT</div> | |

Gambar 4.18 Mockup Halaman Chat

c. Halaman Login

SISTEM INFORMASI PENYEWAAN SCAFFOLDING

USERNAME

PASSWORD

LOGIN

Gambar 4.19 Mockup Halaman Login

d. Halaman Operator

PENYEWAAN SCAFFOLDING Dashboard Master Transaksi Master Chatbot Pengaturan

Selamat Datang di Sistem Informasi Penewaan Scaffolding

Dashboard

Jumlah Pelanggan

Jumlah Barang

Transaksi Dalam Peminjaman

Transaksi Selesai

Daftar Pertanyaan

Daftar Jawaban

Gambar 4.20 Mockup Halaman Utama Operator

e. Halaman Master *Chatbot*

PENYEWAAN SCAFFOLDING Dashboard Master Transaksi Master Chatbot Pengaturan

Data Jawaban

Tambah Data

TABEL DATA JAWABAN

Gambar 4.21 Mockup Halaman Data Jawaban

f. Halaman Penghitungan TF-IDF

PENYEWAAN SCAFFOLDING [Dashboard](#) [Master Transaksi](#) [Master Chatbot](#) [Pengaturan](#)

Perhitungan TF-IDF

Jumlah Dokumen

Bobot Dokumen

Jumlah Vektor

Hasil Count Vektor

TABEL TERM DENGAN BOBOT

Gambar 4.22Mockup Halaman Perhitungan TF-IDF

g. Halaman Pertanyaan

PENYEWAAN SCAFFOLDING [Dashboard](#) [Master Transaksi](#) [Master Chatbot](#) [Pengaturan](#)

Data [Pertanyaan Customer](#)

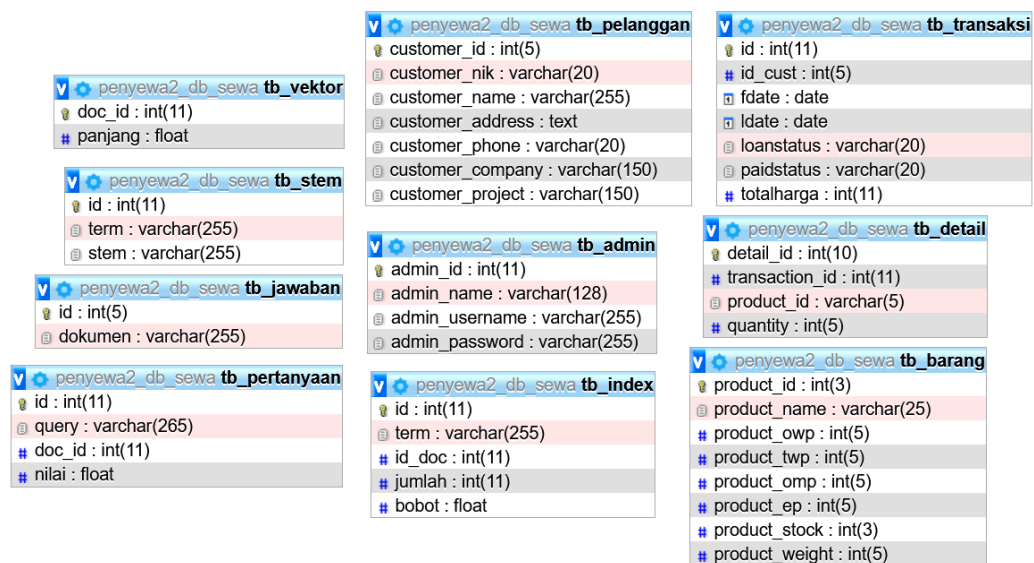
TABEL PERTANYAAN

Gambar 4.23Mockup Halaman Data Pertanyaan

BAB V. IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Basisdata

Implementasi Basisdata untuk sistem yang dibangun diberi nama db_sewa. Basisdata yang dibuat memiliki beberapa tabel yang sebelumnya telah dirancang. 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.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------------|--------------|-------------------|------------|------|---------|----------|----------------|--|
| 1 | admin_id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | admin_name | varchar(128) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 3 | admin_username | varchar(255) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 4 | admin_password | varchar(255) | latin1_swedish_ci | | No | None | | | Change Drop More |

Gambar 5.2 Tabel Admin

5.1.2 Tabel Jawaban

Tabel *tb_jawaban* yang sebelumnya telah dirancang pada tabel 4.24 diimplementasikan pada gambar 5.3.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|----------------|--------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | id | int(5) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | dokumen | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Gambar 5.3Tabel Dokumen

5.1.3 Tabel *Index*

Tabel *tb_index* yang sebelumnya telah dirancang pada tabel 4.25 diimplementasikan pada gambar 5.4.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|---------------|--------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | term | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | id_doc | int(11) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 4 | jumlah | int(11) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 5 | bobot | float | | | No | None | | | Change Drop More |

Gambar 5.4Tabel Index

5.1.4 Tabel Vektor

Tabel *tb_vektor* yang sebelumnya telah dirancang pada tabel 4.29 diimplementasikan pada gambar 5.5.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|----------------|---------|-----------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | doc_id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | panjang | float | | | No | None | | | Change Drop More |

Gambar 5.5Tabel Vektor

5.1.5 Tabel Pertanyaan

Tabel *tb_pertanyaan* yang sebelumnya telah dirancang pada tabel 4.21 diimplementasikan pada gambar 5.6

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|---------------|--------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | query | varchar(265) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | doc_id | int(11) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 4 | nilai | float | | | No | None | | | Change Drop More |

Gambar 5.6Tabel Pertanyaan

5.1.6 Tabel *Stem*

Tabel *tb_stem* yang sebelumnya telah dirancang pada tabel 4.27 diimplementasikan pada gambar 5.7.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|-------------|--------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | term | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | stem | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Gambar 5.7Tabel Stem

5.1.7 Tabel Barang

Tabel *tb_barang* yang sebelumnya telah dirancang pada tabel 4.26 diimplementasikan pada gambar 5.8.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|-----------------------|-------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | product_id | int(3) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | product_name | varchar(25) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | product_unit | varchar(11) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 4 | product_owp | int(5) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 5 | product_twp | int(5) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 6 | product_omp | int(5) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 7 | product_ep | int(5) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 8 | product_stock | int(3) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 9 | product_weight | int(8) | | | No | None | | | Change Drop More |

Gambar 5.8Tabel Barang

5.1.8 Tabel Pelanggan

Tabel *tb_pelanggan* yang sebelumnya telah dirancang pada tabel 4.22 diimplementasikan pada gambar 5.9.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|-------------------------|--------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | customer_id | int(5) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | customer_nik | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | customer_name | varchar(255) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 4 | customer_address | text | utf8mb4_general_ci | | No | | | | Change Drop More |
| <input type="checkbox"/> 5 | customer_phone | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 6 | customer_company | varchar(150) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 7 | customer_project | varchar(150) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Gambar 5.9 Tabel Pelanggan

5.1.9 Tabel Transaksi

Tabel *tb_transaksi* yang sebelumnya telah dirancang pada tabel 4.28 diimplementasikan pada gambar 5.10.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|-------------------|-------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | id_cust | int(5) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | fdate | date | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 4 | ldate | date | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 5 | loanstatus | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 6 | paidstatus | varchar(20) | utf8mb4_general_ci | | No | None | | | Change Drop More |

Gambar 5.10 Tabel Transaksi

5.1.10 Tabel Detail Transaksi

Tabel *tb_detail_transaksi* yang sebelumnya telah dirancang pada tabel 4.23 diimplementasikan pada gambar 5.10.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|----------------------------|----------------|------------|--------------------|------------|------|---------|----------|----------------|--------------------|
| <input type="checkbox"/> 1 | detail_id | int(10) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| <input type="checkbox"/> 2 | transaction_id | int(11) | | | No | None | | | Change Drop More |
| <input type="checkbox"/> 3 | product_id | varchar(5) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| <input type="checkbox"/> 4 | quantity | int(5) | | | No | None | | | Change Drop More |

Gambar 5.11 Tabel Detail Transaksi

5.2 Implementasi Sistem

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

5.3.1 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_jawaban");
    $no = 1;
    while($d = mysqli_fetch_array($data)) {
    ?>
```

Gambar 5.12 Potongan Kode Menampilkan Data *Master* Jawaban

5.3.2 *Preprocessing*

Pada potongan kode berikut dijelaskan bagaimana alur kode program yang digunakan untuk melakukan proses *preprocessing*. Potongan kode program 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*

5.3.3 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 ORD

```

```

ER 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

5.3.4 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

5.3.5 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_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)");
    }
}

```

Gambar 5.16 Potongan Kode Hitung *Similarity*

5.3.6 Penentuan Respon *Chatbot*

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

```
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 ($num_rows > 0) {
        //Tampilkan respon
        while ($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_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('Jawaban tidak ditemukan...');
            }
        }
    }
}
```

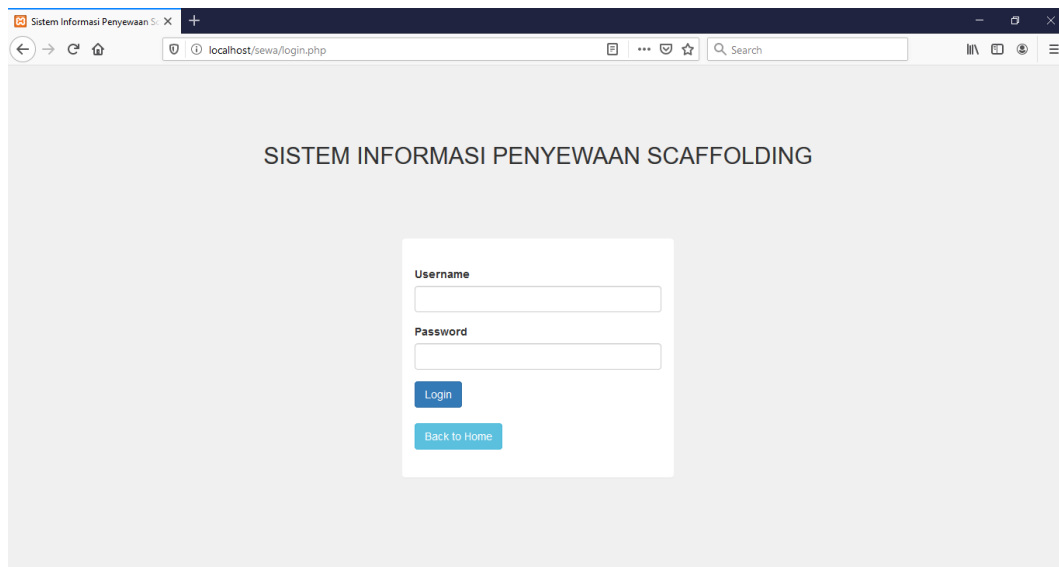
Gambar 5.17 Potongan Kode Penentuan Respon *Chatbot*

5.3 Implementasi Anatarmuka Pengguna

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

5.3.1 Antarmuka Pengguna Login Operator

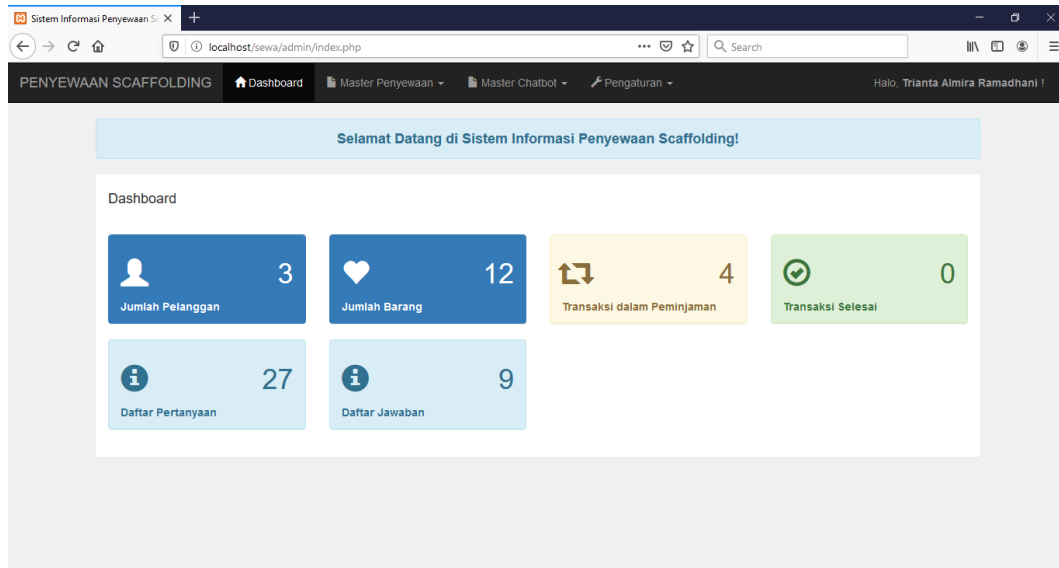
Implementasi dari antarmuka pengguna *login* operator yang sebelumnya telah dirancang pada gambar 4.19 dapat dilihat pada gambar 5.18.



Gambar 5.18Halaman Login

5.3.2 Antarmuka Pengguna Halaman *Dashboard*

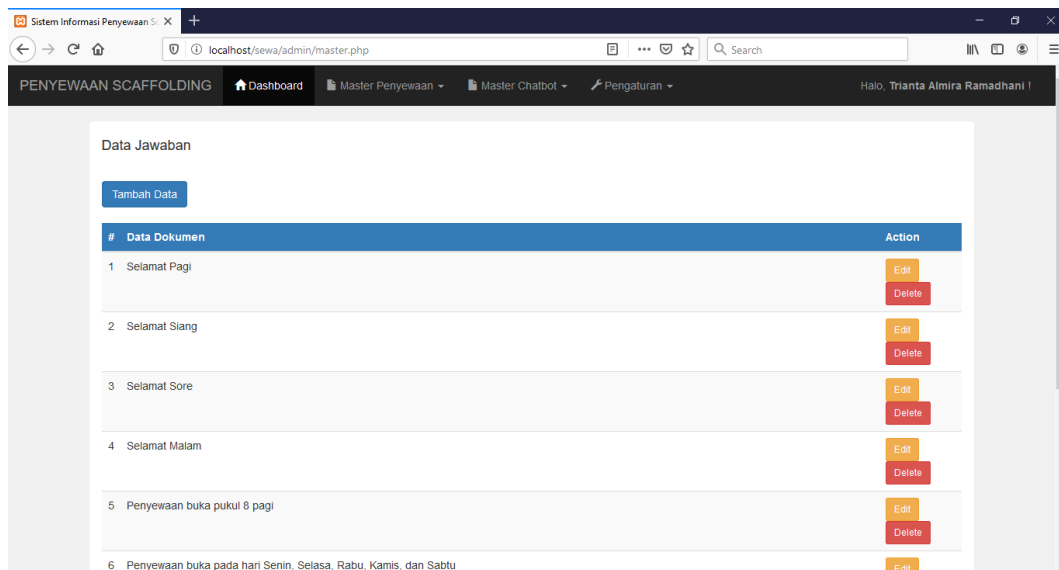
Implementasi dari antarmuka pengguna halaman *dashboard* yang sebelumnya telah dirancang pada gambar 4.20 dapat dilihat pada gambar 5.19.



Gambar 5.19 Halaman Dashboard

5.3.3 Interface Halaman Kelola Data Master

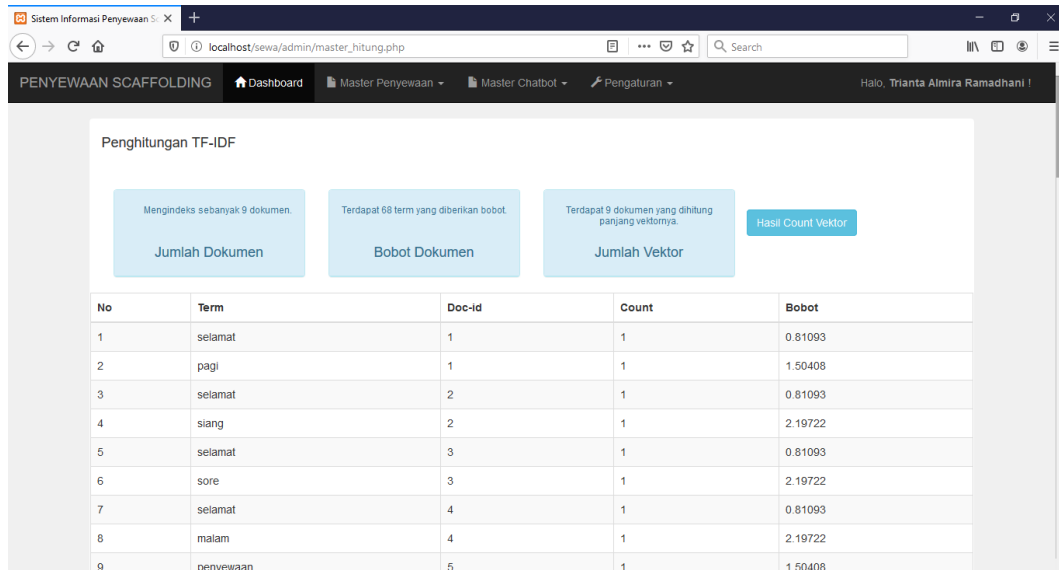
Implementasi dari antarmuka penggunahalaman kelola data *master* yang sebelumnya telah dirancang pada gambar 4.21 dapat dilihat pada gambar 5.20



Gambar 5.20 Halaman Master Jawaban

5.3.4 Interface Halaman Hitung TF-IDF

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



Penghitungan TF-IDF

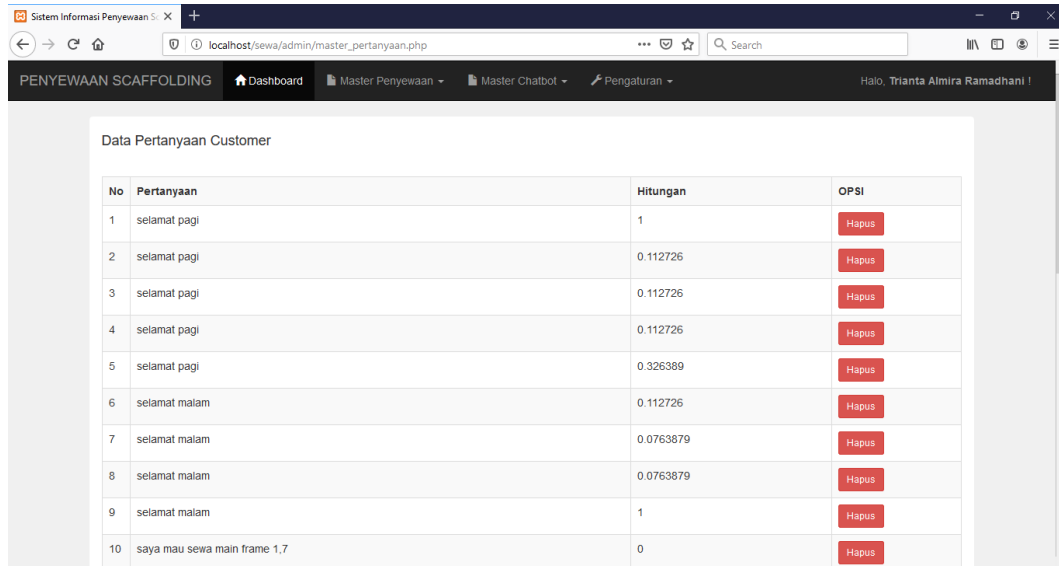
Mengeindeks sebanyak 9 dokumen. Terdapat 68 term yang diberikan bobot. Terdapat 9 dokumen yang dihitung panjang vektornya. Hasil Count Vektor

| No | Term | Doc-id | Count | Bobot |
|----|-----------|--------|-------|---------|
| 1 | selamat | 1 | 1 | 0.81093 |
| 2 | pagi | 1 | 1 | 1.50408 |
| 3 | selamat | 2 | 1 | 0.81093 |
| 4 | siang | 2 | 1 | 2.19722 |
| 5 | selamat | 3 | 1 | 0.81093 |
| 6 | sore | 3 | 1 | 2.19722 |
| 7 | selamat | 4 | 1 | 0.81093 |
| 8 | malam | 4 | 1 | 2.19722 |
| 9 | penyewaan | 5 | 1 | 1.50408 |

Gambar 5.21 Halaman Perhitungan TF-IDF

5.3.5 Interface Halaman Data Pertanyaan

Implementasi dari antarmuka pengguna halaman hitung TF-IDF yang sebelumnya telah dirancang pada gambar 4.23 dapat dilihat pada gambar 5.22



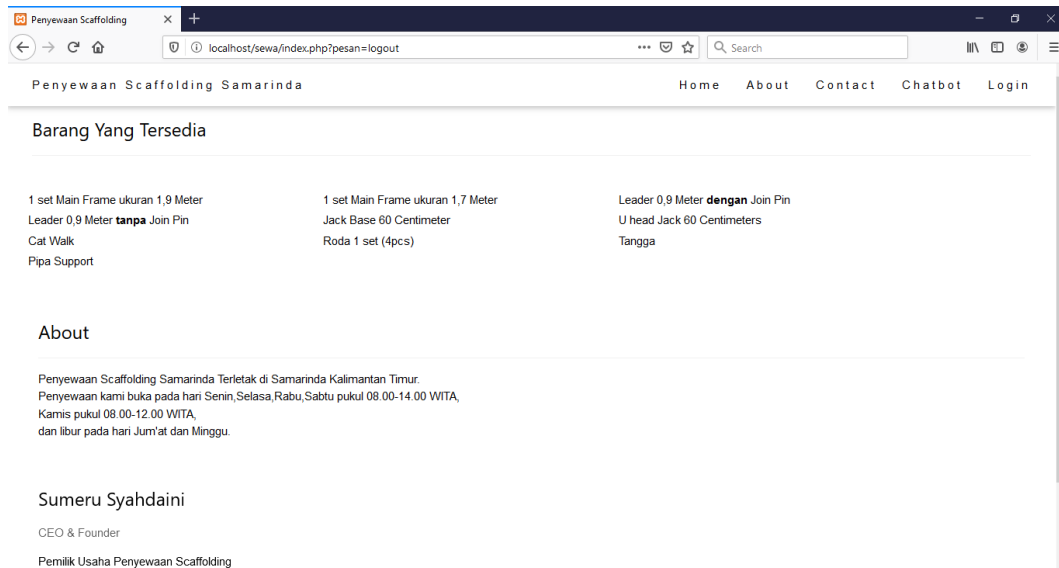
Data Pertanyaan Customer

| No | Pertanyaan | Hitungan | OPSI |
|----|------------------------------|-----------|-------|
| 1 | selamat pagi | 1 | Hapus |
| 2 | selamat pagi | 0.112726 | Hapus |
| 3 | selamat pagi | 0.112726 | Hapus |
| 4 | selamat pagi | 0.112726 | Hapus |
| 5 | selamat pagi | 0.326389 | Hapus |
| 6 | selamat malam | 0.112726 | Hapus |
| 7 | selamat malam | 0.0763879 | Hapus |
| 8 | selamat malam | 0.0763879 | Hapus |
| 9 | selamat malam | 1 | Hapus |
| 10 | saya mau sewa main frame 1,7 | 0 | Hapus |

Gambar 5.22 Halaman Pertanyaan dari Pelanggan

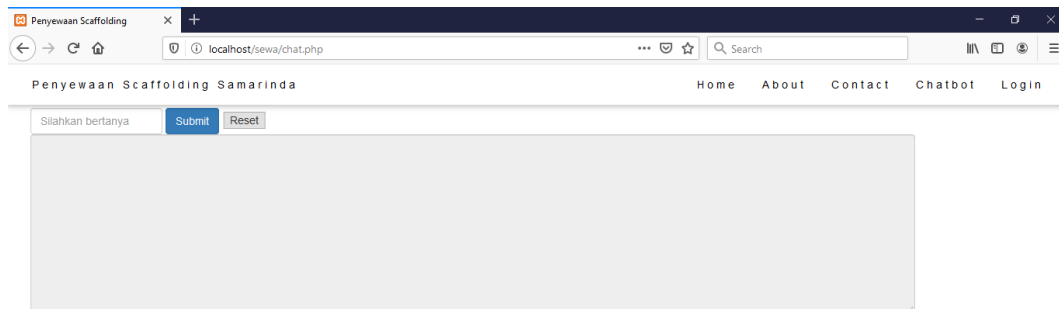
Halaman utama dimana pelanggan dapat melihat barang apa saja yang disewakan, mengetahui mengenai penyewaan scaffolding, siapa pemiliknya dan dapat memberikan komentar mengenai proses penyewaan

yang telah dilakukan. Halaman utama ditunjukkan pada Gambar 5.23 dari implementasi gambar 4.17.



Gambar 5.23 Halaman Utama

Halaman berikutnya adalah halaman bagi pelanggan 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

5.4 Pengujian Fungsional

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

5.3.6 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 Operator

| No. | Menu | Hasil yang Diharapkan | Hasil yang Diperoleh | Status |
|-----|---------------------|--|---|----------|
| 1. | Form Login Operator | Mengalihkan operator ke <i>dashboard operator</i> sesuai <i>username</i> dan <i>password</i> . | Operator yang berhasil memasukkan <i>username</i> dan <i>password</i> beralih ke halaman <i>dashboard</i> . | Berhasil |

Tabel 5.2 Pengujian Fungsional Menu Admin

| No. | Menu | Hasil yang Diharapkan | Hasil yang Diperoleh | Status |
|-----|--|--------------------------------------|---|----------|
| 1. | Menu <i>Dashboard</i> | Menampilkan halaman <i>dashboard</i> | Halaman <i>dashboard</i> berhasil tampil. | Berhasil |
| 2. | <i>Dropdown Master Penyewaan Barang</i> | Menampilkan data barang | Data barang berhasil tampil | Berhasil |
| 3. | <i>Dropdown Master Penyewaan Pelanggan</i> | Menampilkan data pelanggan | Data pelanggan berhasil tampil | Berhasil |

| | | | | |
|----|---|------------------------------------|--|----------|
| 4. | <i>Dropdown Master Penyewaan Transaksi</i> | Menampilkan data Transaksi | Data Transaksi berhasil tampil | Berhasil |
| 5. | <i>Dropdown Master Chatbot Jawaban</i> | Menampilkan data Jawaban | Data Jawaban berhasil tampil | Berhasil |
| 6. | <i>Dropdown Master Chatbot Hitung TF-IDF</i> | Menampilkan data hitung TF-IDF | Data Hitung TF-IDF Berhasil tampil | Berhasil |
| 7. | <i>Dropdown Master Chatbot History Pertanyaan</i> | Menampilkan data pertanyaan | Data Pertanyaan berhasil tampil | Berhasil |
| 8. | <i>Dropdown Pengaturan Ganti Password</i> | Menampilkan halaman ganti password | Halaman ganti password berhasil tampil | Berhasil |
| 9. | <i>Dropdown Pengaturan Logout</i> | 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 <i>form</i> tambah dan menyimpan data produk. | Form tambah data tampil dan data produk berhasil tersimpan. | Berhasil |
| 3. | Tombol <i>Edit</i> Data Produk | Menampilkan <i>form edit</i> dan mengubah | Form <i>edit</i> data tampil dan data | Berhasil |

| | | | | |
|----|-------------------------------------|------------------------|--------------------------------|----------|
| | | data produk. | produk berhasil dirubah. | |
| 4. | Tombol <i>Delete</i> 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 <i>form</i> tambah dan menyimpan data pelanggan. | Form tambah data tampil dan data pelanggan berhasil tersimpan. | Berhasil |
| 3. | Tombol <i>Edit</i> Data Pelanggan | Menampilkan <i>form edit</i> dan mengubah data pelanggan. | Form <i>edit</i> data tampil dan data pelanggan berhasil dirubah. | Berhasil |
| 4. | Tombol <i>Delete</i> Data Pelanggan | Menghapus data pelanggan. | Data pelanggan berhasil terhapus. | Berhasil |

Tabel 5.5 Pengujian Fungsional 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 <i>form</i> 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 <i>Edit</i> Data Transaksi | Menampilkan <i>form edit</i> dan mengubah data transaksi. | Form <i>edit</i> data tampil dan data transaksi berhasil dirubah. | Berhasil |
| 5. | Tombol <i>Delete</i> 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 <i>form</i> tambah dan menyimpan datajawaban. | Form tambah data tampil dan data jawaban berhasil tersimpan. | Berhasil |
| 4. | Tombol <i>Edit</i> DataJawaban | Menampilkan <i>form edit</i> dan mengubah data jawaban. | Form <i>edit</i> data tampil dan datajawaban berhasil dirubah. | Berhasil |
| 5. | Tombol <i>Delete</i> 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 <i>modal</i> hasil perhitungan vektor jawaban | <i>Modal</i> 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 <i>similarity</i> antara <i>query user</i> dengan dokumen sistem. | Data <i>similarity</i> berhasil tampil. | Berhasil |
| 2. | Tombol <i>Delete History</i> Pertanyaan | Menghapus data <i>history</i> pertanyaan. | Data history pertanyaan berhasil terhapus. | Berhasil |

Tabel 5.9 Pengujian Fungsional *User Chatbot*

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

5.3.7 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 menggunakan 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:

5.5 Pengujian Akurasi

Tabel 5.10 Tabel Pengujian Akurasi

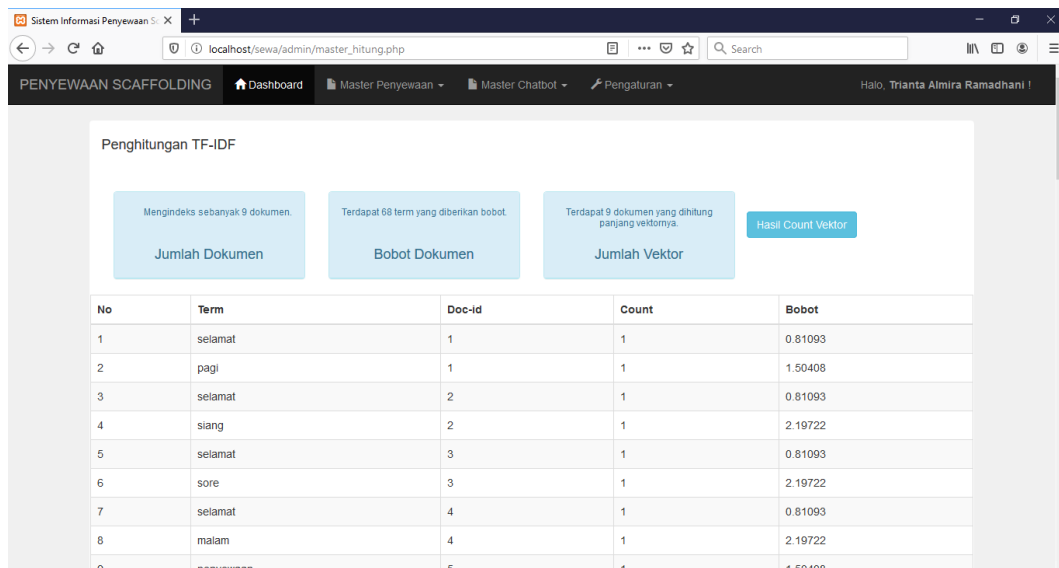
| No | Pertanyaan | Jawaban Sistem | Akurasi |
|----|------------------------|--|---------|
| 1. | Pinjam main frame | harga sewa main frame 1,7 selama 1 minggu 30.000 rupiah, selama 2 minggu 40.000 rupiah, selama 1 bulan 50.000 rupiah | Akurat |
| 2. | Selamat pagi | Selamat pagi | Akurat |
| 3 | Selamat 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

6.1 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:



| No | Term | Doc-id | Count | Bobot |
|----|-----------|--------|-------|---------|
| 1 | selamat | 1 | 1 | 0.81093 |
| 2 | pagi | 1 | 1 | 1.50408 |
| 3 | selamat | 2 | 1 | 0.81093 |
| 4 | siang | 2 | 1 | 2.19722 |
| 5 | selamat | 3 | 1 | 0.81093 |
| 6 | sore | 3 | 1 | 2.19722 |
| 7 | selamat | 4 | 1 | 0.81093 |
| 8 | malam | 4 | 1 | 2.19722 |
| 9 | penyewaan | 5 | 1 | 1.50408 |

Gambar 6.1 Contoh Perhitungan Bobot TF-IDF

Pada gambar 6.1 merupakan contoh hasil dari penghitungan TF-IDF yang telah dilakukan oleh sistem. Setelah dilakukan pembobotan kata, maka akan dilakukan penghitungan menggunakan *cosine similarity* untuk membandingkan tingkat kemiripan antar dokumen dengan *query* yang dimasukkan oleh *user*. Penghitungan *cosine similarity* menggunakan rumus sesuai dengan rumus (4). Sebagai contoh *user* memasukkan input “*harga sewa scaffolding*” 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*.

6.2 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

7.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan pada bab I hingga bab VI, maka dapat disimpulkan bahwa Sistem ini dapat Melakukan Tokenizing (Memisah kata penyusun dari suatu dokumen) dan menghitung bobot Tf-Idf setiap kata tersebut pada tiap array. Kemudian menggunakan Metode TF-IDF dan Cosine Similarity untuk mencari jawaban pada sistem. Sehingga saat user menuliskan pertanyaan, akan muncul jawaban sesuai harapan pelanggan. Sistem juga dapat memudahkan user dalam menerima informasi sesuai dengan yang diharapkan. Sehingga tugas Operator menjadi ringan karena sudah tidak lagi harus selalu membuka *WhatsApp* untuk melayani para pelanggan yang membutuhkan informasi tentang penyewaan *scaffolding*.

7.2 Saran

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

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

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 DocumentFrequency 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*

```
<?php
    include '../koneksi.php';
    $data = mysqli_query($koneksi, "SELECT * FROM tb_jawaban");
    $no = 1;
    while($d = mysqli_fetch_array($data)) {
?>
```

Lampiran 1 Source Code Perhitungan *Chatbot*

```
<?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 ju
mlah = $count WHERE term = '$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)) {
            $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_v
ektor");
$rown = mysqli_fetch_array($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 F
ROM 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);
        $aBobotQuery[] = $idf;
        $panjangQuery = $panjangQuery + $idf * $idf;
    }
}
$panjangQuery = sqrt($panjangQuery);
$jumlah_mirip = 0;
$resDocId = mysqli_query($koneksi, "SELECT * FROM tb_vektor OR
DER 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 * $panjangD
ocId);
        $resInsertCache = mysqli_query($koneksi, "INSERT INTO
tb_pertanyaan (query, doc_id, nilai) VALUES ('$query', $docId, $si
milarity)");
        $jumlah_mirip++;
    }
}

```

```

    }
    if ($jumlah_mirip == 0) {
        $resInsertCache = mysqli_query($koneksi, "INSERT INTO tb_p
ertanyaan (query, doc_id, nilai) VALUES ('$query', 0, 0)");
    }
}
function ambilCache($keyword){
    include 'koneksi.php';
    $resCache = mysqli_query($koneksi, "SELECT * FROM tb_pertanya
n 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 * FRO
M 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_perta
nyaan 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 * FRO
M tb_jawaban WHERE id = $doc_id");
                $rowChat = mysqli_fetch_array($responChat);

                $answer = $rowChat['dokumen'];
                print($answer);
            } else {
                print('Silahkan menghubungi kontak dibawah.');
```