

**APLIKASI CHATBOT INFORMASI DEPARTEMEN TEKNIK INFORMATIKA
UNIVERSITAS HASANUDDIN**



TUGAS AKHIR

*Disusun dalam rangkai memenuhi salah satu persyaratan
Untuk menyelesaikan program Strata-1 Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Makassar*

Disusun Oleh:

BEATRIX WANDA

D421 15 017

**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS HASANUDDIN
MAKASSAR**

2022

LEMBAR PENGESAHAN SKRIPSI
APLIKASI CHATBOT INFORMASI DEPARTEMEN TEKNIK
INFORMATIKA UNIVERSITAS HASANUDDIN

Disusun dan diajukan oleh

BEATRIX WANDA

D42115017

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka
Penyelesaian Studi Program Sarjana Program Studi Teknik Informatika Fakultas
Teknik Universitas Hasanuddin pada tanggal 24 Agustus 2022 dan dinyatakan
telah memenuhi syarat kelulusan.

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

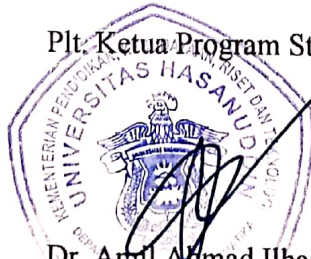


Dr. Ir. Ingrid Nurtanio, MT
Nip. 196108131988112001



Anugrayani Bustamin, S.T., M.T.
Nip. 199012012018074001

Plt. Ketua Program Studi,



Dr. Amil Ahmad Ilham, S.T., M.IT.
Nip. 19731010-199802 1 002

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : BEATRIX WANDA
Nim : D421 15 017
Program Studi : S1 Teknik Informatika

Menyatakan dengan sebenar-benarnya bahwa skripsi yang berjudul :

APLIKASI CHATBOT INFORMASI DEPARTEMEN TEKNIK INNFORMATIKA UNIVERSITAS HASANUDDIN

Adalah karya ilmiah saya sendiri dan sepanjang pengetahuan saya di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan/ditulis/diterbitkan sebelumnya, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila dikemudian hari ternyata didalam naskah skripsi ini terdapat unsur-unsur djiplakan, saya bersedia menerima sanksi atas perbuatan tersebut dan diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2000, pasal 25 ayat 2 dan pasal 70).

Gowa, 23 Agustus 2022

Yang membuat Pernyataan


BEATRIX WANDA

ABSTRACT

Chatbot is an application designed to communicate with machines. This communication helps users in finding information. The information provided varies, such as about academics. This chatbot was built specifically for information covering the Department of Informatics Engineering, Hasanuddin University. This chatbot was built using the TF-IDF weighting method. The Term Frequency Inverse Document Frequency (TF-IDF) method is a weighting method in the form of integration between term frequency and inverse document frequency. Term frequency is the number of occurrences of a term in a document. *Inverse Document Frequency*, namely reducing the dominance of terms that often appear in various documents, by taking into account the inverse of the frequency of documents containing a word. The TF-IDF method is used in this study to select which questions in the database best match the questions entered by the user by looking for the highest weight. From the results of the study, the accuracy of the answer system was tested with black box testing and the percentage answer accuracy was 93%. Testing from the *user acceptance* and the results were 94.4%.

Keywords: Information, Chatbot, *black box* testing, *user acceptance*, TF – IDF.

ABSTRAK

Chatbot adalah suatu aplikasi yang dirancang untuk berkomunikasi dengan mesin. Komunikasi ini membantu *user* dalam mencari sebuah informasi. Informasi yang diberikan bermacam-macam, seperti tentang akademik. Chatbot yang dibangun ini dikhususkan untuk sebuah informasi yang melingkupi departemen Teknik Informatika Universitas Hasanuddin. Chatbot ini dibangun dengan menggunakan metode pembobotan TF-IDF. Metode *Term Frequency Inverse Document Frequency* (TF-IDF) merupakan metode pembobotan dalam bentuk integrasi antar *term frequency* dengan *inverse document frequency*. *Term frequency* yaitu jumlah kemunculan sebuah term pada sebuah dokumen. *Inverse Document Frequency* yaitu pengurangan dominasi term yang sering muncul di berbagai dokumen, dengan memperhitungkan kebalikan frekuensi dokumen yang mengandung suatu kata. Metode TF-IDF digunakan pada penelitian ini untuk memilih pertanyaan mana di database yang paling cocok dengan pertanyaan yang diinput oleh user dengan mencari bobot tertinggi. Dari hasil penelitian dilakukan pengujian akurasi jawaban sistem dengan teks inputan user dan didapatkan persentase keakuratan jawaban sebesar 93%. Dilakukan juga pengujian dari sisi *user acceptance* dan diperoleh hasil sebesar 94.4%.

Kata Kunci: Informasi, Chatbot, *black box testing*, *user acceptance*, TF – IDF.

KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Mahakuasa atas segala limpahan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan tugas akhir yang berjudul **“Aplikasi Chatbot Informasi Departemen Teknik Informatika Universitas Hasanuddin”**. Tugas Akhir ini dibuat untuk memenuhi salah satu syarat memperoleh gelar Sarjana Strata-1 Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Di dalam pengerjaan tugas akhir ini, saya tentunya menyadari segala kekurangan dan keterbatasan. Oleh karena itu, dengan segala kerendahan hati saya memohon maaf atas segala kesalahan dan kekurangan pada penyusunan tugas akhir ini. saya juga sangat menyadari tanpa bantuan dan partisipasi dari berbagai pihak tugas akhir ini tidak dapat terselesaikan dengan baik. Untuk itu, saya menyampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Tuhan Yang Maha Kuasa, atas segala rahmat serta karunia-Nya yang tak terhingga kepada saya dalam mengerjakan tugas akhir ini;
2. Kedua orang tua dan seluruh keluarga, yang selalu memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendampingi dan membantu penulis;
3. Ibu Dr.Ir. Ingrid Nurtanio, M.T., selaku pembimbing I dan Ibu Anugrahyani Bustamin, S.T., M.T., selaku pembimbing II yang selalu menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan saya dalam penyusunan tugas akhir;

4. Bapak Dr. Amil Ahmad Ilham, S.T., M.IT., selaku Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas segala bimbingan dan dukungan selama masa perkuliahan saya;
5. Bapak Dr. Indrabayu, ST.,M.T.,M.Bus.Sys. dan Bapak Dr. Eng. Zulkifli Tahir, ST., M.Sc., yang telah menyempatkan waktunya memberikan saran kepada saya;
6. Bapak dan Ibu Dosen Departemen Teknik Elektro dan Teknik Informatika Universitas Hasanuddin atas bimbingan dan nasehat selama perkuliahan;
7. Teman-teman Teknik Informatika 2015 – HYPERV15OR FT-UH atas dukungan, semangat serta pengalaman yang diberikan selama ini;
8. Segenap Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah membantu saya;
9. Seluruh pihak yang tak sempat saya sebutkan satu persatu yang telah banyak meluangkan waktu, tenaga, dan pikiran selama penyusunan tugas akhir ini.

Harapan saya, semoga Tugas Akhir ini dapat memberikan sumbangsih dan manfaat besar bagi masyarakat.

Makassar, Juni 2022

Penulis

DAFTAR ISI

ABSTRAK	ii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian	3
1.4. Manfaat Penelitian	3
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1 Chatbot	6
2.1.1 Sejarah Singkat Chatbot	8
2.1.2 Perkembangan Chatbot	9
2.2 Preprocessing Data.....	10
2.3 Term Frequency Inverse Document Frequency (TF – IDF)	15
2.4 Aplikasi Berbasis Web	18
2.4.1 PHP.....	18
2.4.2 Laravel.....	21
2.4.3 MySQL.....	24
2.5 Black box Testing	24
2.6 User Acceptance Testing.....	26
BAB III METODOLOGI PENELITIAN.....	29
3.1 Tahapan Penelitian	29
3.2 Instrumen Penelitian.....	30
3.3 Perancangan Desain Interface	31

3.3.1 Halaman Admin	31
3.3.2 Aplikasi Chatbot.....	32
3.4 Perancangan Implementasi Sistem.....	33
BAB IV HASIL DAN PEMBAHASAN	39
4.1 Hasil Pembuatan Aplikasi	39
4.2 Pengujian sistem dan Pembahasan.....	40
BAB V PENUTUP.....	46
5.1 Kesimpulan	46
5.2 Saran.....	46
DAFTAR PUSTAKA	47
LAMPIRAN	49

DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian.....	29
Gambar 3.2 Desain Interface Halaman Admin	32
Gambar 3.3 Desain interface chatbot	32
Gambar 3.4 Gambaran Umum Cara Kerja Sistem	33
Gambar 3.5 Flowchart preprocessing data input.....	34
Gambar 3.6 Flowchart Algoritma TF - IDF	36
Gambar 4.1 Halaman Admin.....	39
Gambar 4.2 Tampilan chatbot	40

DAFTAR TABEL

Tabel 3.1 Tabel Contoh Pembobotan	38
Tabel 4.1 Hasil Pengujian User Acceptance	42
Tabel 4.2 Tabel bobot jawaban	43
Tabel 4.3 hasil pembobotan user acceptance test oleh mahasiswa	43

BAB I

PENDAHULUAN

1.1. Latar Belakang

Informasi adalah bagian penting dalam kehidupan sehari-hari, dimana informasi tersebut diharapkan bisa didapatkan dengan cepat dan akurat. Aktivitas kerja sekarang semakin membutuhkan media komunikasi dan penyedia informasi yang cepat. Hal ini didukung dengan hadirnya teknologi internet yang semakin berkembang pesat dengan koneksi yang semakin cepat dan murah. Keadaan seperti inilah yang melahirkan fasilitas komunikasi secara real-time. Alat bantu untuk memperoleh informasi merupakan salah satu kendala yang terjadi hingga saat ini. Pengguna tentunya menginginkan sebuah alat bantu penyedia informasi yang lebih interaktif layaknya model diskusi, yang dapat menggunakan bahasa sehari-hari.

Mahasiswa, Dosen dan Staf Pegawai Departemen Teknik Informatika Universitas Hassanuddin saat ini belum memiliki sistem digital untuk memperoleh informasi tentang departemen secara cepat dan spesifik. Informasi tentang Departemen Teknik Informatika Unhas saat ini hanya dapat diperoleh melalui website *<https://eng.unhas.ac.id/informatics/>*. Pengunjung memperoleh informasi dengan cara mengakses website tersebut dan mencari informasi. Tetapi data yang ada pada website tersebut tidak spesifik.

Chatbot adalah sebuah program komputer yang dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada pengguna (manusia) melalui bentuk teks, suara, dan atau visual. Percakapan yang terjadi antara komputer dengan manusia merupakan bentuk respon dari program

yang telah dideklarasikan pada database program pada komputer. Kemampuan komputer dalam menyimpan banyaknya data tanpa melupakan satu pun informasi yang disimpannya digabungkan dengan kepraktisan bertanya pada sumber informasi langsung dibandingkan dengan mencari informasi sendiri serta kemampuan learning yang dimilikinya menyebabkan chatbot adalah customer service yang handal. Aplikasi-aplikasi chatbot saat ini mulai banyak bermunculan. Hal ini dikarenakan perkembangan dunia kecerdasan buatan yang semakin maju. Aplikasi chatbot tersebut masih mengalami kerancuan dalam melakukan proses penalaran dengan mengambil kesimpulan dari beberapa data. Sehingga seringkali mengakibatkan aplikasi-aplikasi chatbot tersebut tidak mengerti dan memahami topik yang sedang dibicarakan walaupun didalam basis pengetahuannya terdapat data-data yang mendukung.

Terms Frequency-Inverse Document frequency (TF-IDF) merupakan suatu cara untuk memberikan bobot hubungan suatu kata (token) terhadap dokumen. Bobot hubungan antara kata dalam sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi dan rendah apabila frekuensi kata sebut rendah. Metode TF-IDF digunakan untuk mempresentasikan nilai numerik dokumen yang kemudian dapat dihitung kedekatan antar dokumen sehingga dengan menggunakan metode ini diharapkan sistem dapat memberikan *output* yang sesuai berdasarkan *input* yang diberikan dengan memperhitungkan kemiripan antar keduanya. Metode TF-IDF ini penulis gunakan karena merupakan metode pembobotan kata yang terkenal efisien, mudah dan memiliki hasil yang akurat. (Robertson dan Stephen dalam Abdul Aziz Maarif, 2015).

Berdasarkan uraian diatas, maka dalam penelitian ini akan dibuat sebuah aplikasi *chatbot* yang menyediakan berbagai informasi yang dibutuhkan mahasiswa seputaran Departemen Teknik Informatika Universitas Hasanuddin.

1.2. Rumusan Masalah

Rumusan masalah pada tugas akhir ini adalah :

1. Bagaimana mengembangkan aplikasi chatbot untuk menyediakan informasi yang dibutuhkan mahasiswa dalam ruang lingkup Departemen Teknik Informatika?
2. Bagaimana performa dari aplikasi chatbot yang dikembangkan dari sisi *user acceptance* ?

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Mengembangkan aplikasi chatbot untuk menyediakan informasi yang dibutuhkan mahasiswa dalam ruang lingkup Departemen Teknik Informatika.
2. Mengetahui performa dari aplikasi chatbot yang dikembangkan dari sisi *user acceptance*.

1.4. Manfaat Penelitian

1. Penelitian ini dapat di gunakan untuk menambah wawasan mengenai pembuatan *chatbot*. Hasil penelitian ini dapat menjadi referensi dalam peningkatan kemampuan akademik.
2. Penelitian ini dapat menjadi rekomendasi bagi penelitian selanjutnya yang berkaitan dengan pembuatan *chatbot*.

3. Penelitian ini dapat digunakan sebagai media untuk mendapatkan atau mencari informasi tentang Departemen Teknik Informatika Unhas.

1.5. Batasan Masalah

Dalam pelaksanaan penelitian ini, untuk menghindari luasnya pembahasan, maka masalah dibatasi pada:

1. Percakapan hanya dapat dilakukan menggunakan bahasa Indonesia.
2. Topik pembicaraan hanya dibatasi mengenai informasi tentang profil dosen, kalender akademik, jadwal kuliah dan profil departemen teknik informatika.

1.6. Sistematika Penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka diuraikan beberapa tahapan dari saya secara sistematis, yaitu:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika saya yang berkaitan dengan penelitian yang dilakukan.

BAB II LANDASAN TEORI

Bab ini menjelaskan mengenai dasar teori yang berkaitan dengan tugas akhir ini.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang tahapan yang akan dilakukan dalam penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang hasil yang didapatkan berdasarkan perancangan skenario yang telah dibuat pada penelitian ini.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian dan saran yang diperlukan untuk pengembangan pada penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Chatbot

Chatbot atau juga bisa disebut dengan *chatterbot* merupakan sebuah mesin atau program komputer yang dapat melakukan percakapan dan obrolan dengan user atau pengguna, baik melalui teks maupun dengan suara. Singkatnya chatbot merupakan sebuah mesin yang sudah dilatih dengan pola tertentu untuk menanggapi atau merespon setiap chat dari user dengan cara mencocokkan dengan pola yang sudah ditentukan atau diprogram sebelumnya. Chatbot adalah sebuah program komputer yang bertujuan untuk mensimulasikan sebuah kecerdasan buatan untuk dapat melakukan sebuah percakapan dengan manusia (Shawar & Atwell, 2007). Chatbot adalah sebuah implementasi dari bidang ilmu pengolahan bahasa alami, pembelajaran mesin, rekayasa perangkat lunak dan kecerdasan buatan. Sebuah chatbot akan dianggap menggunakan sebuah kecerdasan buatan, ketika masukan bahasa alami terdapat beberapa kata yang susunannya tidak sama pada sebuah kalimat, namun program mampu mempelajari bahwa masukan tersebut memiliki arti yang sama dengan masukan sebelumnya, sehingga program mampu merespon dengan balasan yang sesuai. Chatbot dapat dikembangkan sesuai dengan kebutuhan yang ada, namun setiap chatbot yang dikembangkan memiliki tipe tersendiri dalam mengeksekusi masukan pengguna. Chatbot terdiri atas berbagai tipe sebagai berikut:

1. *Button-Based* Chatbot

Button-based chatbot adalah tipe chatbot yang masukan pengguna berdasarkan tombol perintah yang telah disediakan oleh chatbot tersebut. Tombol perintah yang disediakan oleh sebuah *button-based* chatbot adalah sebuah hirarki pohon keputusan. Pada *button-based* chatbot ini *knowledge base* yang disiapkan oleh pengembang tidak terlalu banyak, karena masukan pengguna hanya berdasarkan tombol yang disediakan oleh sistem. Kelemahan dari tipe chatbot ini adalah pengguna tidak dapat memberi masukan atau pertanyaan dengan leluasa, karena setiap tombol telah diatur dengan sebuah *query* untuk sebuah masukan.

2. *Keyword Recognition-Based* Chatbot.

Keyword recognition-based chatbot adalah tipe sebuah chatbot yang merespon masukan pengguna dengan cara mengenali beberapa kata kunci yang ada pada masukan pengguna. Masukan yang diterima oleh chatbot jenis ini berupa kalimat atau susunan kata. Cara *keyword recognition-based* chatbot menentukan sebuah respon atau balasan yang tepat kepada sebuah masukan yang telah diberikan adalah mengecek setiap kata sampai menemukan kata kunci yang cocok dengan *knowledge base* pada sistem. Tipe chatbot ini mempunyai kelemahan yaitu ketika merespon sebuah masukan yang mempunyai kata kunci yang sama.

3. *Contextual* Chatbot

Contextual chatbot adalah sebuah tipe chatbot yang memanfaatkan *machine learning* dan *artificial intelligent* untuk mengolah percakapan

sistem dengan pengguna. *Contextual chatbot* dalam menerima sebuah masukan sama dengan *keyword recognition-based* chatbot yaitu menerima sebuah masukan berupa susunan kata atau kalimat. Perbedaan *contextual chatbot* dengan tipe chatbot yang lain yaitu setiap masukan pengguna akan diingat dan dipelajari lagi sehingga ketika ada sebuah masukan dengan susunan kata berbeda namun memiliki arti yang sama, sistem mampu memberi respon dengan tepat. *Contextual chatbot* memerlukan pengembangan *knowledge base* yang harus dikembangkan secara berkala, agar dapat mempelajari setiap masukan pengguna

2.1.1 Sejarah Singkat Chatbot

Sejarah klasik dari chatterbot awal adalah ELIZA (1966) dan PARRY (1972). Program yang baru-baru saja dikembangkan yaitu A.L.I.C.E, Jabberwacky dan D.U.D.E. Pada masanya, ELIZA dan PARRY digunakan untuk menstimulasi percakapan tertulis, namun banyak chatterbot kini mendukung fitur fungsional seperti permainan dan kemampuan pencarian website. Tahun 1984, sebuah buku berjudul *The Policeman's Beard is Half Constructed* dipublikasikan. Buku ini diduga ditulis oleh sebuah chatterbot Racter –walaupun program ini dirilis untuk tidak mampu melakukannya.

Salah satu penelitian penting di bidang kecerdasan buatan (AI) adalah pemrosesan bahasa alami (*Natural Processing Language*). Biasanya bidang AI lemah memberdayakan perangkat lunak (atau software) khusus atau bahasa pemrograman yang dibuat secara

spesifik dengan fungsi yang lebih sempit. Contohnya A.L.I.C.E menggunakan bahasa pemrograman yang disebut sebagai AIML yang fungsinya spesifik yaitu sebagai agen percakapan, selanjutnya banyak diadopsi oleh pengembang Alicebots lain. Meskipun demikian, A.L.I.C.E masih murni berdasarkan teknik pencocokan pola tanpa kemampuan penalaran –teknik yang sama yang digunakan ELIZA pada tahun 1966. Berbeda dengan AI kuat, yang membutuhkan cita rasa dan kemampuan penalaran logis.

Jabberwacky mempelajari respon baru dan berbasis pada konteks interaksi pengguna waktu nyata (atau real-time), bukan dengan digerakan basis data statis. Beberapa chatterbot terbaru juga mengkombinasikan pembelajaran waktu nyata dengan algoritma yang mengoptimalkan kemampuan berkomunikasi berbasis percakapannya, salah satu contoh yang sangat populernya yaitu Kyle, pemenang Penghargaan Leodis AI 2009. Meskipun, saat ini belum ada tujuan umum percakapan kecerdasan buatan, dan beberapa pengembang perangkat lunak berfokus pada aspek praktis, pengambilan informasi (atau information retrieval)

2.1.2 Perkembangan Chatbot

Saat ini kebanyakan chatbot digunakan dalam bidang bisnis, sebagai salah satu bentuk pelayanan customer yang mampu beroperasi 24 jam, untuk menjawab berbagai pertanyaan customer. Dengan adanya chatbot pelayanan customer diharapkan akan lebih maksimal.

Selain dalam bidang bisnis, pemanfaatan chatbot saat ini juga banyak digunakan dalam bidang lain, seperti chatbot pada bidang pendidikan sebagai pembantu mata pelajaran tertentu, chatbot pada bidang pariwisata yang dapat membagikan informasi wisata di daerah tertentu dan chatbot sebagai hiburan dan percakapan seperti aplikasi Sim simi.

Sistem percakapan otomatis kini telah berkembang, dan perusahaan perusahaan sudah menggunakan sistem-sistem tersebut untuk membantu *call center* memberikan panduan kontak. Chatbot pun sudah di implementasikan melalui jejaring sosial, seperti *twitter* dan *Windows Live Messenger*. Portal online populer seperti *eBay* dan *PayPal* juga menggunakan agen virtual multi bahasa untuk memudahkan penggunaannya. Misalnya, *PayPal* menggunakan *chatterbot* Louise untuk menangani *query* dalam bahasa Inggris dan chatbot Lea untuk *query* dalam bahasa Perancis. Chatbot tersebut dikembangkan oleh VirtuOz. Selain itu Chatbot juga di implementasikan untuk bidang komersial, pendidikan, *entertainment* dan sektor pelayanan public (Kerly dkk, 2007).

2.2 Preprocessing Data

Data preprocessing adalah proses yang mengubah data mentah ke dalam bentuk yang lebih mudah dipahami. Proses ini penting dilakukan karena data mentah sering kali tidak memiliki format yang teratur. Preprocessing data penting untuk dilakukan karena dapat memberikan fungsi atau manfaat pada data mining.

Proses ini utamanya dilakukan untuk memastikan kualitas data baik sebelum digunakan saat analisis data. Dalam proses ini Anda dapat memastikan enam hal, yakni akurasi data, kelengkapan, konsistensi, ketepatan waktu, tepercaya, dan dapat diinterpretasi dengan baik. Jika sebuah data telah diproses berdasarkan enam acuan tersebut, proses analisis data akan lebih mudah dilakukan karena data dari berbagai sumber telah dimuat dalam sebuah set data dengan format yang sama.

Ada beberapa step yang perlu dilakukan ketika akan melakukan data preprocessing. Berikut ini beberapa tahapannya:

1. Data Cleaning

Tahap pertama yang perlu dilakukan ketika akan *preprocessing* data adalah *data cleaning* atau membersihkan data. Artinya, data mentah yang telah Anda peroleh perlu diseleksi kembali. Kemudian, hapus atau hilangkan data-data yang tidak lengkap, tidak relevan, dan tidak akurat. Dengan melakukan tahap ini, Anda akan menghindari kesalahpahaman ketika menganalisis data tersebut.

Ada dua hal yang harus Anda perhatikan ketika melakukan *data cleaning*, yakni pastikan data-data yang dikumpulkan tidak lagi mengandung data dengan missing values. Lalu, Anda juga harus memastikan bahwa data-data tersebut seluruhnya diperlukan saat proses analisis data. Dengan demikian, data yang Anda kumpulkan telah disesuaikan dan tidak mubazir.

2. Data Integration

Karena *data preprocessing* akan menggabungkan beberapa data dalam suatu dataset, maka Anda harus mengecek data-data yang datang dari berbagai

sumber tersebut supaya memiliki format yang sama. Proses ini menjadi salah satu step penting dalam proses ini. Beberapa permasalahan bisa muncul ketika melakukan data integration. Misalnya, Anda ingin menggabungkan data dari beberapa sumber. Anda harus mengetahui bahwa data pada sumber pertama dimiliki oleh si A, dan data pada sumber kedua juga terkait dengan si A. Kelihatannya seperti hal mudah, padahal dua sumber tersebut memiliki format yang berbeda. Itulah yang membuat data integration sedikit lebih rumit.

3. Transformasi Data

Proses berikutnya yang harus dilakukan adalah transformasi data. Seperti yang telah dijelaskan di atas, data akan diambil dari berbagai sumber yang kemungkinan memiliki perbedaan format. Anda harus menyamakan seluruh data yang terkumpul supaya dapat mempermudah proses analisis data.

Misalnya, Anda akan mengambil data karyawan pada sumber pertama yang menggunakan format DD/MM/YYYY. Kemudian, pada sumber berikutnya, data karyawan menggunakan format MM/DD/YYYY. Ketika akan mengumpulkan data, keduanya tentu perlu diubah dan diseragamkan dalam satu format yang sama.

4. Mengurangi Data

Tahap terakhir yang perlu dilakukan adalah mengurangi jumlah data (data reduction). Maksudnya adalah Anda harus mengurangi sampel data yang diambil, tetapi dengan catatan, tidak akan mengubah hasil analisis data. Ada tiga teknik yang bisa diterapkan saat melakukan pengurangan data, yakni

dimensionality reduction (pengurangan dimensi), numerosity reduction (pengurangan jumlah), dan data compression (kompresi data). Ketiga teknik tersebut bisa disesuaikan dengan kebutuhan; apakah data yang diolah besar, sedang, atau perlu dikompresi dan berisiko merugikan.

Text preprocessing adalah suatu proses untuk menyeleksi data *text* agar menjadi lebih terstruktur lagi dengan melalui serangkaian tahapan yang meliputi tahapan *case folding*, *tokenizing*, *filtering* dan *stemming*. Tapi, sesungguhnya tidak ada aturan pasti tentang setiap tahapan dalam *text preprocessing*. Semua itu tergantung dengan jenis serta kondisi data yang kita miliki. *Text preprocessing* merupakan salah satu implementasi dari *text mining*. Text mining sendiri adalah suatu kegiatan menambang data, dimana data yang biasanya diambil berupa *text* yang bersumber dari dokumen-dokumen yang memiliki *goals* untuk mencari kata kunci yang mewakili dari sekumpulan dokumen tersebut sehingga nantinya dapat dilakukan analisa hubungan antara dokumen-dokumen tersebut.

Berikut ini merupakan tahapan - tahapan dalam *text preprocessing* sebagai salah satu teknik pengolahan data *text*.

1. Case Folding

Tahapan pertama yang biasanya dilakukan adalah tahapan *case folding*. Tahapan ini hampir selalu disertakan ketika melakukan *text preprocessing* karena data yang kita miliki tidak selalu terstruktur dan konsisten dalam penggunaan huruf kapital. Jadi, peran dari *case folding*

adalah untuk menyamaratakan penggunaan huruf kapital. Misalnya data teks yang kita dapat berupa tulisan "DaTA SCIENCE" maka dengan case folding artinya kita mengubah semua huruf menjadi huruf kecil (*lowercase*) semua. Sementara itu, karakter lain yang bukan termasuk huruf dan angka, seperti tanda baca dan spasi dianggap sebagai delimiter. Delimiter ini bisa juga dihapus atau diabaikan dengan menggunakan perintah yang ada di Python.

2. *Tokenizing*

Kita ambil contoh adalah data *tweet* atau kumpulan *dataset* pesan *spam* pasti terdiri dari kalimat. Untuk memudahkan proses analisis data kita harus memecah kalimat-kalimat tersebut menjadi kata atau disebut dengan token. Dengan tokenizing kita dapat membedakan mana antara pemisah kata atau bukan. Jika menggunakan bahasa pemrograman python biasanya *tokenizing* juga mencakup proses *removing number*, *removing punctuation* seperti simbol dan tanda baca yang tidak penting, serta *removing whitespace*. Selain itu tokenizing juga akan merujuk pada NLTK, tetapi yang sangat disayangkan adalah NLTK belum *support* bahasa Indonesia. Tapi, jangan khawatir karena kita masih bisa menggunakan modul sastrawi

3. *Filtering*

Lanjutan dari tahapan *tokenizing* adalah tahapan *filtering* yang digunakan untuk mengambil kata-kata yang penting dari hasil token tadi. Kata umum yang biasanya muncul dan tidak memiliki makna disebut

dengan *stopword*. Misalnya penggunaan kata penghubung seperti dan, yang,serta, setelah, dan lainnya. Penghilangan *stopword* ini dapat mengurangi ukuran index dan waktu pemrosesan. Selain itu, juga dapat mengurangi level *noise*. Namun terkadang *stopping* tidak selalu meningkatkan nilai *retrieval*. Pembangunan daftar *stopword* (disebut *stoplist*) yang kurang hati-hati dapat memperburuk kinerja sistem Information Retrieval (IR). Belum ada suatu kesimpulan pasti bahwa penggunaan *stopping* akan selalu meningkatkan nilai *retrieval*, karena pada beberapa penelitian, hasil yang didapatkan cenderung bervariasi.

4. *Stemming*

Tahap *stemming* adalah tahapan yang juga diperlukan untuk memperkecil jumlah indeks yang berbeda dari satu data sehingga sebuah kata yang memiliki *suffix* maupun *prefix* akan kembali ke bentuk dasarnya. Selain itu juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda karena mendapatkan imbuhan yang berbeda pula. Di library NLTK juga sudah tersedia modul untuk proses *stemming* antara lain, *porter*, *lancaster*, *wordnet*, dan *snowball*. Tapi, kembali lagi modul-modul tersebut belum *support* untuk *text* berbahasa Indonesia.

2.3 *Term Frequency Inverse Document Frequency (TF – IDF)*

Term Frequency Inverse Document Frequency (TF-IDF) merupakan metode pembobotan dalam bentuk integrasi antara *term frequency* dengan *inverse document frequency*. Algoritma *Term Frequency Inverse-Document Frequency*

merupakan suatu algoritma yang menggabungkan antara *Term frequency* dengan *Inverse Document Frequency*. *Term frequency* yaitu jumlah kemunculan sebuah term pada sebuah dokumen. *Inverse Document Frequency* yaitu pengurangan dominasi *term* yang sering muncul di berbagai dokumen, dengan memperhitungkan kebalikan frekuensi dokumen yang mengandung suatu kata. Metode TF-IDF digunakan pada penelitian ini untuk memilih pertanyaan mana di database yang paling cocok dengan pertanyaan yang diinput oleh user dengan mencari bobot tertinggi.

Metode TF-IDF merupakan suatu cara untuk memberikan bobot hubungan suatu kata (*term*) terhadap dokumen. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukkan seberapa

penting kata itu di dalam dokumen tersebut. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata

tersebut yang rendah pada kumpulan dokumen.

Rumus algoritma TF – IDF adalah sebagai berikut :

$$tf = 0.5 + 0.5 \times \frac{tf}{\max(tf)} \quad (1)$$

$$idf_t = \log\left(\frac{D}{df_t}\right)$$

$$W_{dt} = tf_{d,t} \times IDF_{d,t}$$

Keterangan :

tf	= banyaknya kata yang dicari pada sebuah dokumen
$\max(tf)$	= jumlah kemunculan terbanyak term pada dokumen yang sama.
Nilai D	= total dokumen
df_t	= jumlah dokumen yang mengandung term t.
IDF	= Inversed Document Frequency ($\log_2 (D/df)$)
d	= dokumen ke-d
t	= kata ke-t dari kata kunci
W_{dt}	= bobot dokumen ke-d terhadap kata ke-t

Berikut adalah contoh perhitungan TF-IDF. Misalkan diberikan beberapa kalimat sebagai berikut:

d1 = saya berangkat ke kampus untuk belajar

d2 = ibu pergi berbelanja ke pasar

d3 = ayah ke kantor untuk bekerja

Corpus atau kumpulan kata yang dibentuk dari beberapa kalimat tersebut adalah

“ayah”, “bekerja”, “belajar”, “berangkat”, “berbelanja”,

“ibu”, “kampus”, “kantor”, “ke”, “pasar”, “pergi”, “saya”,
dan “untuk”.

Misalkan kita ingin menghitung bobot pada kata “berangkat” pada kalimat pertama, karena kata “berangkat” muncul satu kali pada dokumen pertama maka perhitungan bobot kata “berangkat” menjadi :

$$\begin{aligned}W_{berangkat,d1} &= tf_{d,t} \times IDF_{d,t} && (2) \\&= 1 \times \log\left(\frac{3}{1}\right) \\&= 1 \times 0.477 \\&= 0.477\end{aligned}$$

2.4 Aplikasi Berbasis Web

Aplikasi berbasis web dapat diakses kapan pun dan dari mana pun selama ada internet. Dapat diakses hanya dengan menggunakan web browser (umumnya sudah tersedia di PC, PDA, dan handphone terbaru), tidak perlu menginstall aplikasi client khusus. Alasan menggunakan aplikasi berbasis web karena Macromedia Dreamwaver MX memasukkan server side development dan memberikan suatu keistimewaan serta kemudahan didalamnya. Ditambah dengan yang lain yaitu membangun bantuan terhadap PHP dan bantuan yang baik untuk HTML dan CSS (Gareth, Downes Powel, 2002 : 3).

2.4.1 PHP

PHP menurut Anhar (2010 : 3) adalah bahasa pemrograman web server-side yang bersifat open source, PHP juga merupakan script yang terintegrasi dengan HTML dan berada pada server (

server side HTML embedded script). PHP juga merupakan script yang digunakan untuk membuat halaman website yang sangat dinamis, dinamis berarti halaman tampilan yang akan ditampilkan dibuat saat halaman itu diminta oleh client. PHP pertama kali dibuat oleh Rasmus Lerdorf seorang pemrogram C yang handal dari Greenland Denmark di tahun 1995, PHP diberi nama FI (Form Interpreted) yang digunakan untuk mengelola form dari web. Pada perkembangannya, kode-kode yang digunakan dirilis untuk umum sehingga mulai banyak dikembangkan oleh programmer diseluruh dunia. Tahun 1997 PHP dirilis dengan versi 2.0, pada versi ini sudah terintegrasi dengan bahasa pemrograman C dan sudah dilengkapi dengan modul sehingga kualitas kerja PHP lebih meningkat secara signifikan. Ditahun yang sama sebuah perusahaan program bernama Zend merilis ulang PHP versi ini dengan lebih baik, bersih dan cepat. Seiring berkembangnya jaman ditahun 1994 PHP versi 4.0 mulai dirilis dan versi ini paling banyak digunakan pada awal abad 21 karena PHP versi ini sudah mampu membangun web kompleks dengan stabilitas kecepatan yang tinggi. Ditahun 2004 perusahaan program Zend merilis PHP lagi dengan versi terbarunya 5.0 yang inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek kedalam PHP untuk menjawab perkembangan bahasa pemrograman kearah paradigma berorientasi objek.

Kelebihan PHP adalah sebagai berikut :

1. PHP memiliki tingkat akses yang lebih cepat.
2. PHP memiliki tingkat lifecycle yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
3. PHP memiliki tingkat keamanan tinggi.
4. PHP mampu berjalan di beberapa server yang ada, misalnya Apache, Microsoft IIS, PWS, AOLServer, phttpd dan Xitami.
5. PHP mampu berjalan di linux sebagai platform sistem operasi utama bagi PHP, namun juga dapat berjalan di FreeBSD, Unix, Solaris, Windows dan yang lain.
6. PHP juga mendukung akses ke beberapa database yang sudah ada, baik yang bersifat *free* ataupun komersial. Database itu antara lain MySQL, PostgreSQL, mSQL, Informix, dan MicrosoftSQL server.
7. PHP bersifat *free* atau gratis.

Bahasa program PHP sering digunakan karena PHP adalah bahasa open source yang memiliki kesederhanaan dan memiliki beberapa fitur built-in yang berfungsi untuk menangani kebutuhan standart dalam pembuatan aplikasi web. PHP juga merupakan bahasa script yang paling mudah dipahami karena memiliki beberapa referensi. PHP juga dapat digunakan untuk berbagai sistem operasi antara lain : Unix, Macintosh serta windows. PHP dapat

dijalankan secara runtime melalui console serta dapat menjalankan perintah-perintah system. Open source disini memiliki arti code-code PHP terbuka untuk umum dan tidak berbayar atas pembelian dari license. Web server yang mendukung PHP dapat ditemukan dimana-mana, mulai dari Apache, IIS, Lighttpd hingga Xitami dengan konfigurasi yang relatif mudah. Selain itu PHP juga dilengkapi dengan berbagai macam pendukung lain seperti support langsung keberbagai macam databasea yang populer seperti Oracle, MySQL dan lain-lain.

2.4.2 Laravel

Menurut Aminudin (2015:1) Laravel adalah sebuah Framework PHP dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti framework-framework yang lain, Laravel dibangun dengan konsep MVC (Model-Controller-View), kemudian Laravel dilengkapi juga command line tool yang bernama “Artisan” yang bisa digunakan untuk packaging bundle dan instalasi bundle melalui command prompt. Berikut ini beberapa fitur yang dimiliki oleh framework Laravel menurut Aminudin (2015:5) :

1) Bundles

Bundles yaitu sebuah fitur dengan system pengemasan modular dan berbagai bundle telah tersedia untuk digunakan dalam aplikasi Anda.

2) *Eloquent ORM*

Eloquent ORM merupakan penerapan PHP lanjutan dari pola “active record” menyediakan metode internal untuk mengatasi kendala hubungan antara objek database. Pembangun *query Laravel Fluent* didukung *Eloquent*.

3) *Application Logic*

Application Logic merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* maupun sebagai bagian dari deklarasi *Route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh framework Sinatra.

4) *Reverse Routing*

Reverse Routing mendefinisikan hubungan antara link dan route, sehingga jika suatu saat ada perubahan pada route secara otomatis akan tersambung dengan link yang relevan. Ketika link yang dibuat dengan menggunakan nama-nama dari route yang ada, secara otomatis laravel akan membuat URI yang sesuai.

5) *Restful Controllers*

Restful Controllers memberikan sebuah option (pilihan) untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.

6) *Class Auto Loading*

Class Auto Loading menyediakan otomatis loading untuk class-class PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah loading yang tidak perlu.

7) *View Composers*

View Composers adalah kode unit logical yang dapat dijalankan ketika sebuah view di load.

8) *IoC Container*

IoC Container memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.

9) *Migrations*

Migrations menyediakan versi sistem control untuk skema database, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak database.

10) *Unit Testing*

Unit Testing mempunyai peran penting dalam framework Laravel, dimana unit testing ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. Unit testing dapat dijalankan melalui fitur “artisan command-line”.

11) *Automatic Pagination*

Automatic Pagination menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.4.3 MySQL

MySQL adalah sebuah perangkat lunak Pembuat database yang bersifat terbuka atau open source dan berjalan disemua platform baik Linux maupun Sistem operasi Windows, MySQL merupakan program pengakses database yang bersifat network sehingga dapat digunakan untuk aplikasi Multi User / Pengguna Banyak (Abdul , 2006). MySQL (*My Struktur Query Language*) merupakan sebuah *database server* yang awalnya berjalan pada sistem Unix dan Linux. Seiring dengan berjalannya waktu dan banyak user yang minat menggunakan database ini, MySQL mulai merilis versi yang dapat diinstal di hampir semua platform termasuk sistem operasi windows. *License* dari MySQL adalah *freeware*, yang artinya kita dapat mendownload dan menggunakannya tanpa harus membayar. Umumnya orang menyebut database seperti MySQL, PostgreSQL atau yang lainnya dengan sebutan database server saja. Beberapa juga menyebutnya dengan database engine, mesin database, SMD, DBMS, *Back End* atau program database saja.

2.5 *Black box Testing*

Menurut Pressman (2007), *black box testing* adalah melakukan pengujian terhadap apa yang dilakukan oleh sistem, khususnya perilaku dan

juga masalah bisnis. *Black box testing* bertujuan untuk mengidentifikasi bug-bug yang ada pada hasil, kinerja dan juga perilaku sistem. Pengujian ini biasanya dilakukan oleh pihak penguji ketika *integration test*, *system test*, dan *acceptance test*, tetapi juga berguna untuk tahap yang lebih awal untuk membantu membangun unit *test case* dan *component test case* yang lebih baik.

Menurut Perry (2006:69), *functional testing* juga dapat disebut sebagai *black box testing* karena tidak ada pengetahuan dari logika internal sistem yang digunakan untuk membuat *test case*. Biasanya dalam pengujian fungsional, teknik validasi lebih digunakan untuk melakukan pengujian. Tim penguji melakukan validasi terhadap *function key* yang ada dan mengobservasi hasilnya. Kelebihan dari *functional testing*: - Melakukan simulasi terhadap kegunaan sistem yang sebenarnya. - Tidak membuat asumsi terhadap struktur sistem. II-14 Kekurangan dari *functional testing*: - Dapat berpotensi menghilangkan logika yang salah dalam software. - Memungkinkan terjadinya pengujian yang redundan. Kedua metode ini berguna untuk validasi pada seluruh sistem. Seperti contohnya, sebuah *functional test case* dapat diambil dari dokumentasi tentang deskripsi bagaimana melakukan sebuah fungsi seperti menerima inputan bar code. Sebuah *structural test case* dapat diambil dari dokumentasi teknikal manual. Untuk pengujian sistem yang efektif diperlukan kedua metode tersebut.

Menurut Lewis (2009:39), pada *black box testing* atau pengujian fungsional, kondisi pengujian dikembangkan berdasarkan fungsionalitas dari

program atau sistem yang akan diuji, oleh karena itu, penguji membutuhkan informasi mengenai data input dan output yang diamati, tetapi tidak mengetahui bagaimana program atau sistem tersebut bekerja atau tidak perlu mengetahui bagaimana struktur internal dari program tersebut melakukan eksekusi. Penguji berfokus pada pengujian fungsionalitas dari program terhadap spesifikasi. Dari definisi di atas, black box testing adalah pengujian terhadap apa yang dilakukan oleh sistem, khususnya perilaku dan juga masalah bisnis di mana penguji membutuhkan informasi mengenai data input dan output yang diamati tetapi tidak perlu mengetahui struktur internal dari sistem tersebut.

2.6 *User Acceptance Testing*

Menurut Perry (2006:70), *User Acceptance Testing* merupakan pengujian yang dilakukan oleh end-user dimana *user* tersebut adalah staf / karyawan perusahaan yang langsung berinteraksi dengan sistem dan dilakukan verifikasi apakah fungsi yang ada telah berjalan sesuai dengan kebutuhan/fungsinya.

Menurut Lewis (2009:134), setelah dilakukan *system testing*, *acceptance testing* menyatakan bahwa sistem software memenuhi persyaratan. *Acceptance testing* merupakan pengujian yang dilakukan oleh pengguna yang menggunakan teknik pengujian *black box* untuk menguji sistem terhadap spesifikasinya. Pengguna akhir bertanggung jawab untuk memastikan semua fungsionalitas yang relevan telah diuji. *Acceptance testing*

biasanya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan-persyaratan tertentu. Pada pengembangan *software* dan *hardware* komersial, *acceptance test* biasanya disebut juga "alpha tests" (yang dilakukan oleh pengguna *in-house*) dan "beta tests" (yang dilakukan oleh pengguna yang sedang menggunakan atau akan menggunakan sistem tersebut). Alpha dan beta test biasanya juga menunjukkan bahwa produk sudah siap untuk dijual atau dipasarkan. *Acceptance testing* mencakup data, environment dan skenario yang sama atau hampir sama pada saat live yang biasanya berfokus pada skenario penggunaan produk tertentu.

Dari definisi di atas, *user acceptance testing* adalah pengujian yang dilakukan oleh pengguna dari sistem tersebut untuk memastikan fungsi-fungsi yang ada pada sistem tersebut telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna (L. Binus, 2017). Berikut adalah rumus menghitung skor pengujian user :

acceptance testing (UAT), yaitu (Muqsith, 2016):

$$Skor = \frac{jumlah\ jawaban}{(jumlah\ pertanyaan \times jumlah\ responden)} \times 100\% \quad (3)$$

Keterangan:

- Jumlah jawaban = Jumlah jawaban responden ya / tidak
- Jumlah pertanyaan = Jumlah pertanyaan yang diajukan kepada responden

- Jumlah responden = Jumlah responden yang mengisi kuisisioner