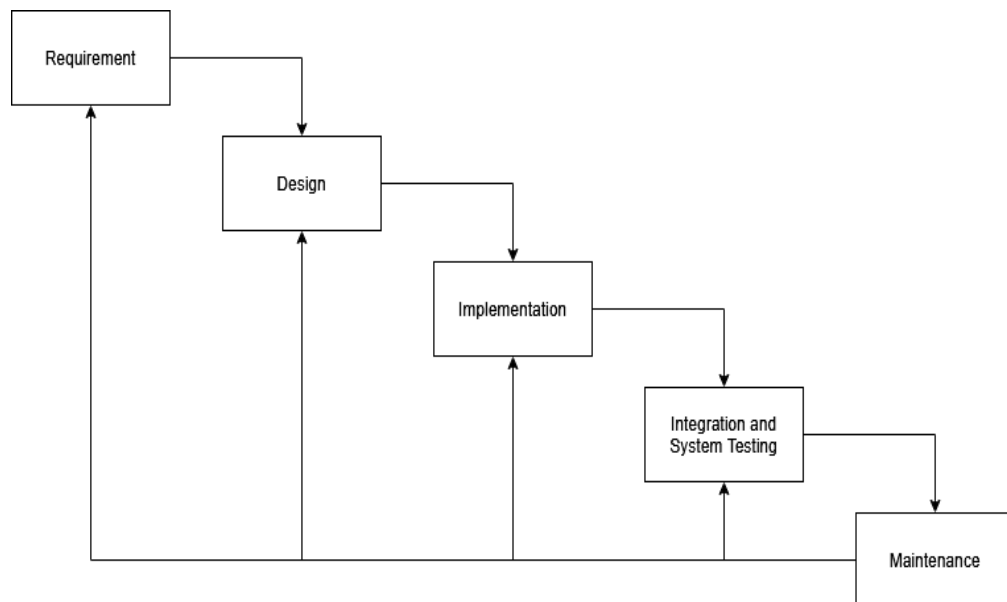


BAB III

METODE PENELITIAN

Pada bab ini akan dijelaskan mengenai metode penelitian yang digunakan oleh penulis. Metode penelitian adalah urutan langkah-langkah pengerjaan dari penelitian dan penjelasan singkat dari tahap penelitian. Alur penelitian yang dilakukan dijelaskan pada gambar berikut. Gambar 3.1 menunjukkan Alur penelitian yang dilakukan oleh penulis.



Gambar 3.1 Metode *Waterfall*

Metode *Waterfall* (metode air terjun) merupakan suatu model pengembangan secara sekuensial. Dalam proses pembuatannya metode *waterfall* dimulai dari analisis, desain, kode, pengujian dan pemeliharaan. Seperti metode pengembangan perangkat lunak lainnya. Metode *waterfall* juga memiliki kelebihan dan kekurangan. Model *Waterfall* adalah model menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, dan pengujian (Handrianto, 2020).

1. Kelebihan

Metode *waterfall* Memiliki *workflow* yang jelas, sehingga lebih memudahkan dalam tahap pengembangan karena dilakukan secara tahap pertahap mulai dari analisis kebutuhan sampai pada tahap pemeliharaan. Hal ini membuat dokumentasi yang jelas, dan juga dapat digunakan untuk pengembangan perangkat lunak dengan skala besar karena melibatkan sumber daya manusia yang banyak sesuai kemampuan masing masing.

2. Kekurangan

Metode ini memiliki proses yang lama, ini menjadi salah satu kekurangan dari metode ini. Metode ini juga memerlukan banyak riset dan juga penelitian pendukung untuk mengembangkan sistem menggunakan metode *waterfall*.

Pada penelitian ini, peneliti menggunakan metode *waterfall* sebagai metode penelitian. Metode ini dipilih karena beberapa pertimbangan berdasarkan kelebihan dan kekurangan yang telah dijelaskan sebelumnya.

Dalam pengembangannya metode *waterfall* memiliki beberapa tahapan, diantaranya yaitu :

1.1 Requirement (Analisis Kebutuhan)

Analisis kebutuhan perangkat lunak adalah fase pengumpulan kebutuhan yang ditingkatkan dan terfokus untuk memeriksa persyaratan perangkat lunak. Pada tahap ini, penulis melakukan analisis kebutuhan terkait kebutuhan data yang diperlukan. Hal ini bertujuan agar penulis lebih memahami perangkat lunak yang diharapkan oleh pengguna serta memenuhi kebutuhan dari pengguna agar mendapatkan nilai atau skor yang cukup baik pada saat dilakukan tahap pengujian diakhir pengembangan.

Metode yang dilakukan dalam analisis kebutuhan data pada penelitian ini adalah sebagai berikut :

3.1.1 Sumber Data

Berdasarkan Sumber data, data yang diambil yaitu data primer dan data sekunder.

1. Data Primer

Data Primer Pada penelitian ini, data primer yang digunakan yaitu pertanyaan-pertanyaan yang sering di ajukan kepada PTIK (Pusat Teknologi Informasi dan Komunikasi) dan juga permasalahan yang sering dialami oleh mahasiswa atau masyarakat kampus.

2. Data Sekunder

Data Sekunder pada penelitian ini, didapatkan dari study literatur dari beberapa buku, jurnal serta penelitian penelitian sebelumnya.

3.1.2 Metode Pengumpulan Data

1. Wawancara

Metode pengumpulan data yang pertama adalah wawancara. Wawancara dilakukan untuk mengetahui pertanyaan-pertanyaan yang sering diajukan masyarakat kampus kepada pihak PTIK (Pusat Teknologi Informasi dan Komunikasi) yang ada di Universitas Panca Marga Probolinggo. Pertanyaan-pertanyaan tersebut dikumpulkan lalu direkapitulasi guna mengurutkan berdasarkan pertanyaan yang diajukan. Hal ini penting dilakukan untuk Menyusun atau menentukan dokumen yang dianggap relevan yang akan dimasukan kedalam database.

2. Kuesioner

Metode pengumpulan data yang kedua adalah kuesioner. Kuesioner dibutuhkan sebagai alat pengumpulan data sampel pertanyaan dari mahasiswa maupun masyarakat umum. Dalam penyebaran kuesioner, penulis memanfaatkan google form sebagai media dari kuesioner. Penulis memilih google form karena google form memiliki kemudahan dalam penggunaannya daripada kuesioner tradisional (menggunakan kertas). Penulis hanya perlu menyebar link google form kepada mahasiswa dan masyarakat umum untuk mengisi kuesioner yang sudah disiapkan.

3. Dokumentasi

Metode pengumpulan data yang ketiga adalah dokumentasi. Pada metode pengumpulan data ini dengan teknik pengumpulan data dengan menggunakan buku pendoman akademik yang telah disahkan oleh rektorat. Ini bertujuan untuk mendapatkan data yang lebih akurat sebagai dokumen yang akan dimasukkan kedalam database.

1.2 Design (Desain Sistem)

Desain merupakan suatu tahapan yang berfokus pada desain perangkat lunak seperti pemodelan UML, Struktur data, Arsitektur perangkat lunak, dan *User Interface* (antar muka). Tahap desain dilakukan dengan menerjemahkan kebutuhan perangkat lunak berdasarkan dari hasil analisis kebutuhan ke dalam bentuk desain, sehingga dapat diimplementasikan menjadi program pada tahap implementasi

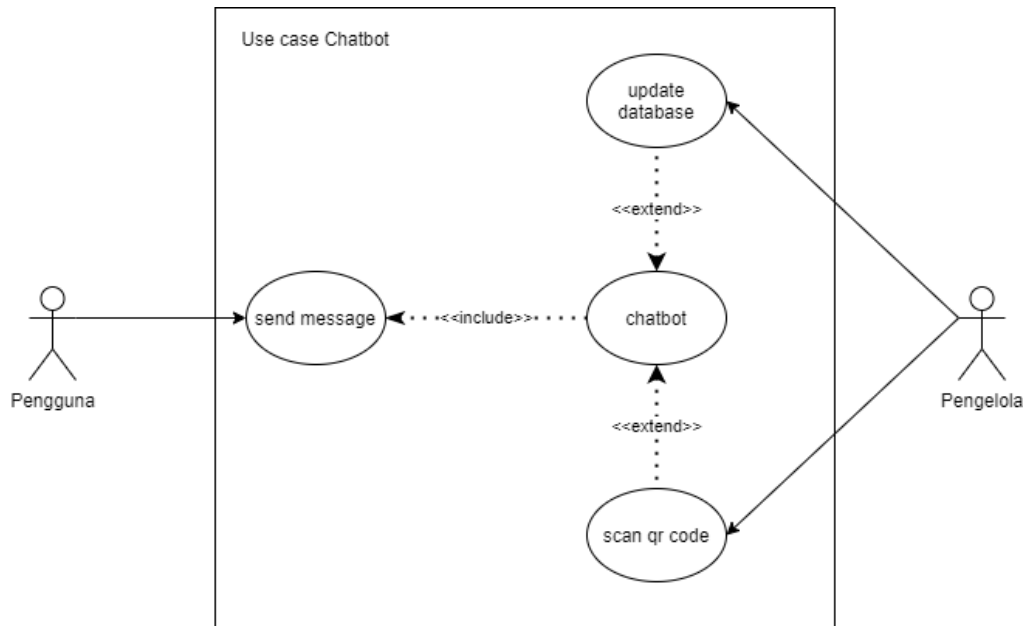
Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari terlebih dahulu dan menyiapkan desain dari sistem. Desain sistem membantu dalam

menentukan perangkat keras dan sistem prasyarat yang akan digunakan, dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3.2.1 Use Case Diagram

Use case Diagram dari Sistem Chabot dapat Digambarkan sebagai

berikut :



Gambar 3.1 *Use Case Chatbot*

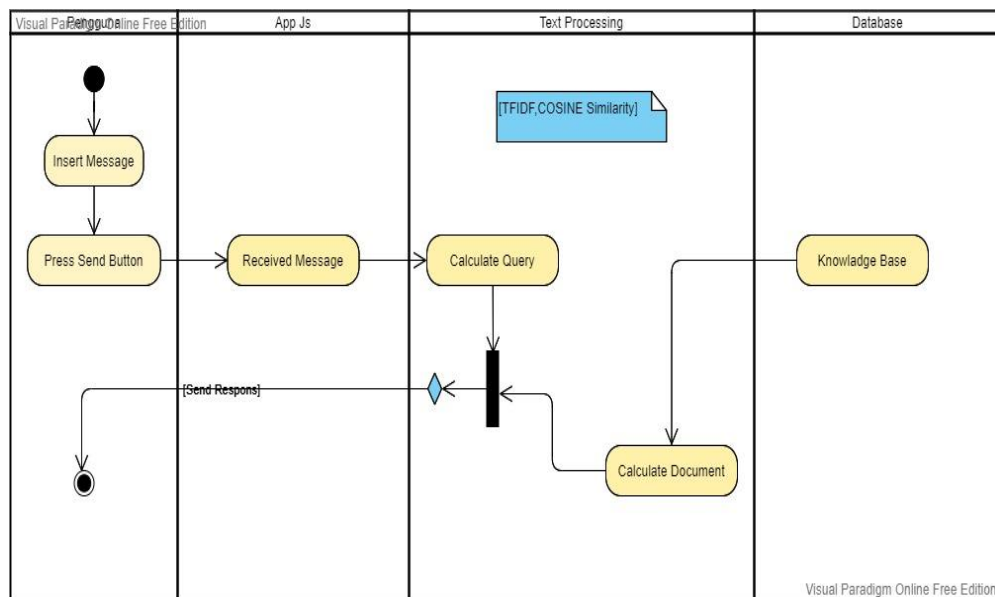
Pada gambar 3.1 digambarkan bahwa ada 2 (dua) *actor* yang terdapat dalam sistem chatbot, dimana pengguna hanya dapat mengirimkan pesan kepada chatbot, pesan tersebut hanya berupa pesan text yang dikirim melalui aplikasi pesn instan yaitu whatsapp.

Adapun pengelola yaitu dapat mengupdate pengetahuan chatbot dengan mengedit file *base.csv* untuk menambah dokumen atau data didalam database, penambahan data ini berguna untuk meningkat daya pengetahuan chatbot seputar akademik universitas panca marga probolinggo agar mampu melayani informasi dan komunikasi menjadi lebih baik antara chatbot dengan pengguna.

Pengelola juga dapat melakukan *scan qr code* agar dapat menggunakan chatbot, cara kerja dari *scan qr code* ini tidak jauh berbeda dengan *scan qr code* milik whatsapp yaitu whatsapp web. Dimana *library* whatsapp api yang penulis gunakan adalah whatsapp web.js, mengutip dari lama resmi whatsapp-web.js (wwebjs.dev, 2022) dikatakan bahwa *library* ini terhubung ke versi resmi *WhatsApp Web*. Maka dari itu, tingkat kemungkinan untuk diblokir oleh pihak whatsapp sangat kecil, meskipun pengembang dari *whatsapp api* (*whatsappweb.js*) tidak menjamin 100% tidak akan diblokir oleh pemilik *whatsapp*.

3.2.2 Class Diagram

Class Diagram dari sistem *ChatBot* dapat digambarkan sebagai berikut :



Gambar 3.2 Class Diagram Sistem ChatBot.

Pada gambar 3.2 digambarkan pengguna mengirimkan sebuah pesan melalui aplikasi whatsapp, lalu whatsapp api menerima pesan tersebut yang *request* dari whatsapp server.

Dari pesan tersebut akan dilakukan proses *text preprocessing* untuk menormalisasikan pesan menjadi sebuah token yang nanti akan dihitung bobot kata dari sebuah dokumen. Dimana pada tahap *text preprocessing*, *query* akan melalui beberapa tahapan, diantaranya adalah *filter question* untuk memfilter dokumen yang akan diproses berdasarkan kata tanya pada *query*, *case folding* untuk mengecilkan semua huruf, *tokenizing* untuk memecah kalimat menjadi sebuah kata perkata (token), *stopword removal* untuk menghapus kata bantu, dan *stemming* yang berguna untuk mengembalikan sebuah kata menjadi kata dasar dengan menghilangkan imbuhan.

Pada tahap *processing*, bobot *query* (pesan pengguna) dan bobot dokumen dalam database dihitung setiap bobotnya menggunakan *TF-IDF* (*Term Frequency – Inverse Document Frequency*), *VSM* (*Vector Space Model*) dan menghitung kemiripan antara *query* dengan dokumen yang ada dalam sistem menggunakan *Cosine Similarity*.

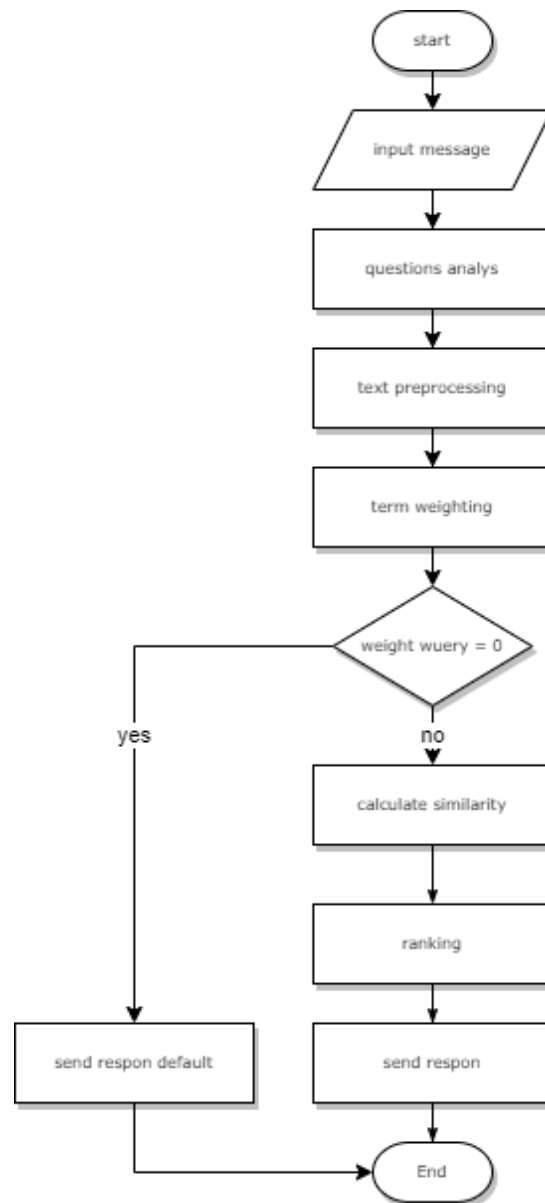
Dokumen yang akan diproses adalah dokumen yang memiliki *document type* (kata tanya) yang terkandung dalam sebuah *query*. Hal ini dijelaskan pada sub bab 3.2.4 arsitektur sistem pada uraian berikutnya, atau dapat dilihat pada gambar 3.5 mengenai gambaran sistem dari *chatbot* yang akan penulis kembangkan.

Setelah melalui proses *processing*, sistem akan melakukan ranking berdasarkan bobot tertinggi dari sebuah kemiripan antara *query* dan dokumen. Dokumen dengan bobot tertinggi yang akan dikirimkan kepada pengguna sebagai hasil respon dari sistem terhadap *query* yang pengguna kirimkan. Hal ini berupa respon dokumen yang paling relevan dengan *query*.

3.2.3 Flowchart Diagram (Diagram Alir)

Flowchart dari sistem *ChatBot* dapat digambarkan sebagai berikut :

1. *Flowchart* Dialog ChatBot



Gambar 3.3 *Flowchart* Dialog *ChatBot*.

Pada gambar 3.3 menggambarkan alur atau proses dari dialog *chatbot*. Langkah awal pada *flowchart* yaitu pengguna memasukan atau mengetikan pertanyaan melalui aplikasi whatsapp, dimana pertanyaan ini akan menjadi

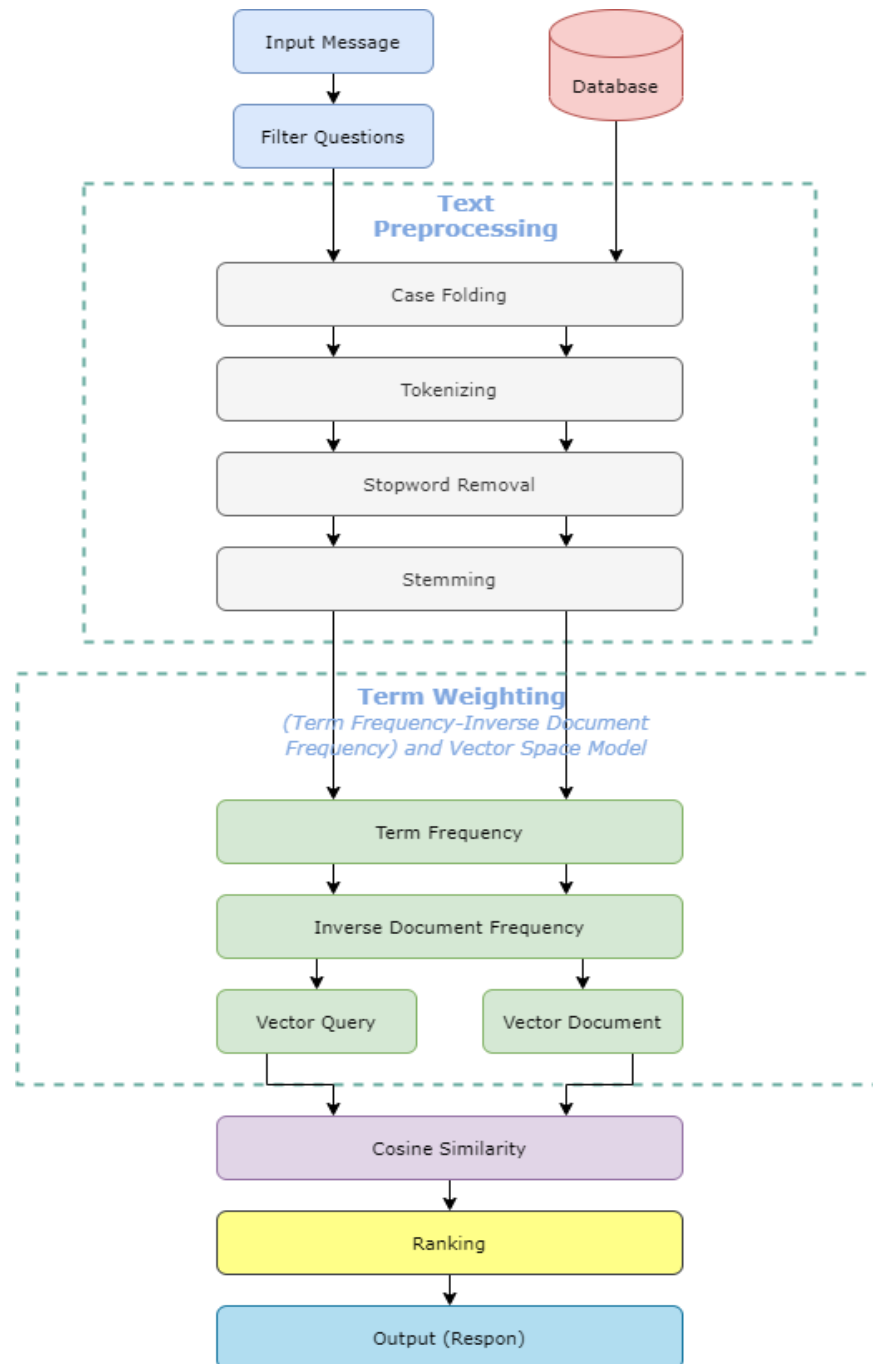
variable input kedalam sistem yang akan dibawa pada tahap selanjutnya. Lalu pertanyaan atau masukan tersebut akan melewati tahap *questions analys*, tahap ini berfungsi untuk memfilter pertanyaan dari pengguna. Hal ini berhubungan dengan tahap selanjutnya mengenai dokumen dengan *type questions* mana yang akan dihitung, kemudian tahap selanjutnya ialah preprocessing.

Tahap awal dari *preprocessing* disini adalah *case folding*, *case folding* adalah tahap mengubah semua masukan menjadi huruf kecil. Hanya huruf ‘a’ sampai ‘z’ yang diterima, Karakter selain huruf dihilangkan dan dianggap delimiter.

Lalu hasil dari *case folding* akan dipecah menjadi kata perkata dengan mengidentifikasi spasi sebagai pemicu utama pemisah kata, proses ini dinamakan dengan *tokenizing* atau tokenisasi. Hasil dari tahap *tokenizing* akan diproses kedalam *stemming*, dimana pada tahap ini setiap kata yang sudah dipecah akan dikembalikan menjadi kata dasar dengan menghilangkan kata imbuhan. dari kata berimbuhan (*affixed word*) dengan cara menghilangkan semua imbuhan (*affix*) yang terdiri dari awalan (*prefix*), sisipan (*infix*), akhiran (*suffix*) dan kombinasi awalan dan akhiran (*confix*).

Setelah melewati tahap *preprocessing*, masukan akan difilter apakah ada *pattern* yang tersimpan dalam database sesuai dengan masukan atau tidak. Jika nilai *true*, atau “iya” maka sistem akan mengirimkan respon atau jawaban sesuai dengan *pattern* yang ada dalam pertanyaan. Apabila bernilai *false*, atau “tidak” maka sistem akan mengirimkan respon default bahwa pertanyaan yang diajukan tidak dimengerti oleh *chatbot*.

3.2.4 Arsitektur Sistem



Gambar 3.5 Arsitektur Sistem *ChatBot*

Pada gambar 3.5 merupakan arsitektur sistem *chatbot*, pada tahap ini dilakukan proses preprocessing, pembobotan kata, menghitung kemiripan dokumen dan juga ranking berdasarkan nilai bobot tertinggi.

Pada saat pengguna mengirimkan pesan (*query*) sistem akan melakukan tahap *preprocessing*, Adapun dari tahapan yang telah digambarkan pada gambar 3.5 adalah sebagai berikut:

1. *filter questions*

Pada tahap ini *query* (pesan pengguna) dilakukan tahap *filtering* pertanyaan, dimana *query* akan dilihat apakah mengandung kata tanya atau tidak. Kata tanya sebagai *filtering* disini penulis simpan dalam sebuah array, berikut daftar kata tanya sebagai filter dari *query* :

```
pertanyaan = ["apa", "siapa", "kenapa", "kapan", "dimana", "bagaimana",  
"berapa"]
```

tahap ini dilakukan untuk mempersempit perhitungan sebuah dokumen, sehingga dokumen yang akan dihitung hanya dokumen dengan tingkat relevan yang tinggi berdasarkan *document type* dari *filter questions*.

2. *Case Folding*

Pada tahap ini, *query* dikonversi menjadi huruf kecil dengan fungsi `lower()` yang ada dalam python. Tahap *case folding* tidak hanya memproses *query* dari pengguna, melainkan juga memproses dokumen dalam sistem.

3. *Tokenizing*

Pada tahap ini, *query* dan dokumen dipecah menjadi kata perkata dan disimpan kedalam sebuah array.

4. *Stopword Removal*

Pada tahap ini, *query* dan dokumen dilakukan *stopword removal* untuk menghilangkan kata yang termasuk kedalam *stopword* seperti kata penghubung.

Berikut beberapa kata yang termasuk kedalam *stopword* :

Tabel 3.1 Daftar *Stopword*

Daftar <i>Stopword</i>
'ada', 'adalah', 'agak', 'agar', 'akan', 'amat', 'anda', 'antara', 'anu', 'apakah', 'apalagi', 'atau', 'bagaimanapun', 'bagi', 'bahwa', 'begitu', 'belum', 'bisa', 'boleh', 'dahulu', 'dalam', 'dan', 'dapat', 'dari', 'daripada', 'demi', 'demikian', 'dengan', 'di', 'dia', 'dimana', 'dll', 'dsb', 'dst', 'dua', 'dulunya', 'guna', 'hal', 'hanya', 'harus', 'ia', 'ingin', 'ini', 'itu', 'itulah', 'jika', 'juga', 'kah', 'kami', 'karena', 'ke', 'kecuali', 'kemana', 'kembali', 'kepada', 'ketika', 'kita', 'lagi', 'lain', 'maka', 'mari', 'masih', 'melainkan', 'menurut', 'mereka', 'namun', 'nanti', 'nggak', 'oh', 'ok', 'oleh', 'pada', 'para', 'pasti', 'pula', 'pun', 'saat', 'saja', 'sambil', 'sampai', 'saya', 'sebab', 'sebagai', 'sebelum', 'sebetulnya', 'secara', 'sedangkan', 'seharusnya', 'sehingga', 'sekitar', 'selagi', 'selain', 'sementara', 'seolah', 'seperti', 'seraya', 'serta', 'sesuatu', 'sesudah', 'setelah', 'seterusnya', 'setiap', 'setidaknya', 'sudah', 'supaya', 'tanpa', 'tapi', 'telah', 'tentang', 'tentu', 'terhadap', 'tetapi', 'tidak', 'toh', 'tolong', 'untuk', 'walau', 'ya', 'yaitu', 'yakni', 'yang'

5. Tahap Kelima, *Stemming*

Pada tahap ini, query dan dokumen yang sudah melalui tahap *tokenizing* dilakukan proses *stemming*, yaitu mengubah semua *term* atau kata menjadi kata dasar. Berikut ini adalah beberapa contoh dari hasil *stemming* dari sebuah dokumen :

Tabel 3.2 Hasil Tahap *Stemming*

Sebelum <i>Stemming</i>		Sesudah <i>Stemming</i>	
'Yayasan',	'Menteri',	'yayasan',	'menteri',
'PancaMarga',	'Kehakiman',	'pancamarga',	'hakim',
'Probolinggo',	'Republik',	'probolinggo',	'republik',
'didirikan',	'Indonesia',	'diri',	'indonesia',
'pada',	'24',	'hari',	'24',
'hari',	'Juni',	'senin',	'juni',
'senin',	'1985',	'tanggal',	'1985',
'tanggal',	'No.',	'14',	'm',
'14',	'M.38.HT.03.01',	'november',	'38',
'November',	'tahun',	'1938',	'ht',
'1938',	'dan',	'akte',	'03',
'dengan',	'SK',	'notaris',	'01',
'Akte',	'Dalam', 'Negeri',	'no',	'tahun',
'Notaris', 'No',	'15',	'42',	'negeri',
'42',	'1987',	'sk',	'15',

Sebelum <i>Stemming</i>		Sesudah <i>Stemming</i>	
'Sk',	'Nomor', '81/DJA/1987'		'1987', 'nomor', '81', 'dja'

Setelah *query* dan dokumen melalui tahap *preprocessing*. *term* atau kata akan melalui tahap selanjutnya yaitu tahap pembobotan kata, dimana setiap kata *query* dan dokumen dihitung bobot katanya. Pada tahap ini tidak semua dokumen akan dihitung bobot katanya, namun hanya dokumen yang mengandung *document type* dari *query* (pesan pengguna) yang akan dihitung bobot katanya.

Pada proses TF-IDF akan dihitung bobot setiap *term* dan VSM (*Vector Space Model*) model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query*. pada model ini dilakukan vektorisasi *query* dan dokumen dengan menghitung bobot setiap kata dan vector document dan jarak *query*.

Pembobotan kata menggunakan TF-IDF VSM pada *query* dan dokumen akan menghasilkan bobot nilai bobot dari sebuah *term*. Dari nilai tersebut akan dihitung *similarity* (kemiripan) menggunakan *Cosine similarity*. Dimana bobot kata dan dokumen akan dijumlah lalu dibagi dengan hasil *summary* dari hasil perhitungan jarak *query* dan dokumen.

Hasil dari perhitungan *similarity* akan didapatkan hasil akhir yaitu bobot dari setiap dokumen, dari nilai tersebut akan dilakukan ranking berdasarkan nilai terbesar dari bobot dokumen. Dimana semua nilai bobot dokumen disimpan dalam sebuah array, dan array dengan indeks pertama karena dianggap

sebagai dokumen yang memiliki kemiripan lebih tinggi untuk dikirimkan kepada pengguna sebagai respon dari sistem.

1.3 *Implementation* (Pengkodean)

Pada tahap ketiga, sistem pertama kali dikembangkan kedalam program kecil. Program kecil atau bisa disebut dari unit kecil ini, akan berhubungan dengan tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk melihat fungsionalitas secara bertahap, hal ini berguna untuk tahap *maintenance* selanjutnya serta mempermudah proses pengembangan sistem.

Penulis disini memetakan tahap *Implementation* atau pengkodean dengan mencatat 3 (tiga) hal, yaitu diantaranya :

1. Analisis kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak mencakup segala aplikasi yang akan digunakan dalam pengkodean.

- a. Windows 10 Pro sebagai sistem operasi yang digunakan
- b. Visual Studio code sebagai *code* editor untuk menulis kode program
- c. Node.js sebagai *runtime environment* menjalankan kode *JavaScript* pada sisi server
- d. Python compiler sebagai compiler untuk menjalankan kode python
- e. Browser (Google Chrome) sebagai aplikasi untuk mengakses *user interface scan barcode whatsapp*.
- f. *Terminal / Command Prompt* sebagai untuk menjalankan perintah *running* program
- g. *Library* pendukung (*Whatsapp web.js, pysastrawi, express, qr-code, child-process*).

2. Analisis kebutuhan Perangkat Keras
 - a. Processor Core i5 @3.2Ghz
 - b. RAM 8 GB
 - c. Monitor
 - d. Mouse
 - e. Keyboard
3. Analisis Kebutuhan Perangkat Tambahan
 - a. Wifi adapter (usb) / Wifi sebagai koneksi ke internet
 - b. Aplikasi Whatsapp dan Nomor Whatsapp untuk uji coba chatbot.

1.4 *Integration and system testing* (Penerapan dan Pengujian)

Pada tahap ini, seluruh unit yang dikembangkan diintegrasikan kedalam sistem lalu diuji dari masing masing unit secara keseluruhan. Guna untuk melihat setiap kegagalan maupun kesalahan yang terjadi.

Pada penelitian ini penulis menggunakan 3 metode pengujian, yaitu diantaranya :

1. BlackBox Testing

Pengujian *black-box* adalah tahap dimana kelayakan dari program yang dibuat diuji. Pengujian ini penting untuk menghindari kesalahan saat menjalankan program yang dibuat.

Menurut Syafnidawaty (2020). *blackbox testing* atau yang lebih sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program.

2. Akurasi

Akurasi merupakan nilai yang diperoleh dari hasil pengukuran mendekati nilai sebenarnya. Semakin dekat nilai hasil dari pengukuran, maka sistem dapat dikatakan semakin akurat. Hal ini dilakukan untuk mengetahui sejauh mana tingkat akurasi *chabot* dalam memberikan jawaban atau *response* untuk pengguna.

3. *User Acceptance Test*

User Acceptance Test merupakan pengujian dengan melibatkan pengguna tanpa mengetahui kode atau sistem secara langsung. Hal ini untuk mengukur tingkat kinerja dari aplikasi serta menguji tingkat kepuasan pengguna terhadap aplikasi yang telah dikembangkan.

1.5 Maintenance (Pemeliharaan)

Tahap akhir dari metode *waterfall* adalah pemeliharaan perangkat lunak. Pemeliharaan termasuk dari memperbaiki kesalahan yang tidak ditemukan pada tahap sebelumnya. Setelah *chatBot* di upload ke Heroku dan dilakukan uji coba dalam kemampuannya. *ChatBot* akan dikembangkan lagi untuk memperbaiki fungsi-fungsi yang tidak berjalan dengan benar, atau memperbaiki bug yang tidak terdeteksi pada tahap pengujian.