BAB III
METODOLOGI PENELITIAN

III.1. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem ini digunakan untuk menerapkan *chatbot,* yang akan diimplementasikan atau dipasang pada *website* DIM ITERA. Sistem *chatbot* ini diharapkan dapat berfungsi dengan baik sesuai dengan kebutuhan.

a. Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan adalah *notebook* atau *Personal Computer* (PC) yang terhubung dengan jaringan internet dengan spesifikasi minimal sebagai berikut :

1. *Processor Core 2 Duo*
2. *RAM* 2 GB

b. Kebutuhan Perangkat Lunak

Pada penelitian ini dibutuhkan beberapa perangkat lunak untuk memfasilitasi pembangunan sistem, antara lain :

1. *Browser* untuk menjalankan aplikasi berbasis *website.*
2. *Apache* sebagai *toolsweb service.*
3. *Windows 7* atau yang versi yang lebih baru, sebagai sistem operasi yang digunakan untuk mengakses aplikasi.

III.2. Tahapan Penelitian

Tahapan – tahapan yang dilakukan dalam pelaksanaan penelitian ini dilakukan agar hasil yang didapatkan tidak menyimpang dari tujuan yang telah ditentukan di awal. Dalam membangun aplikasi *chatbot*, penulis menggunakan metodologi *prototype* seperti yang telah dipaparkan pada subbab I.4. Adapun tahapan – tahapan yang dimaksud adalah sebagai berikut :

a. Komunikasi

Komunikasi adalah awal dari tahapan penelitian ini. Tahapan ini meliputi pengumpulan data dan analisa *prototype*. Pada pengumpulan

data, penulis mencari dan mengumpulkan informasi mengenai *chatbot* yang ingin dibangun, dengan kemampuan cepat dan tepat dalam menjawab setiap pertanyaan yang diajukan. Proses tersebut dimulai dengan mencari kebutuhan yang terkait dalam pembangunan *chatbot*, serta mencari permasalahan yang akan dihadapi *chatbot* dalam proses pengisian DIM ITERA. Dalam mencari informasi tersebut, penulis melakukan wawancara secara langsung ke pihak PMB ITERA atau pihak terkait lainnya. Penulis akan berfokus pada pertanyaan seputar kendala yang dialami ketika merespon setiap pertanyaan yang diajukan pada website DIM ITERA.

Pada tahap ini, penulis juga akan mendapatkan pengetahuan mengenai pertanyaan yang sering ditanyakan oleh mahasiswa beserta jawaban yang diberikan oleh pihak PMB ITERA. Hasil dari proses pengumpulan data tersebut, menjadi acuan bagi penulis dalam menganalisa *prototype* yang akan dibangun.

b. Perencanaan Secara Cepat

Pada tahap ini, penulis melakukan perencanaan secara cepat berdasarkan data pada tahap komunikasi. Perencanaan ini dimaksudkan untuk dapat menghasilkan rancangan sistem aplikasi yang akan dibangun. Hasil dari tahap komunikasi sebelumnya menjadi acuan dalam pembuatan dokumen *AIML* pada tahap perencanaan ini.

c. Pemodelan Perancangan Secara Cepat

Berdasarkan hasil pada tahap sebelumnya, pada tahap ini penulis membuat pemodelan pada aplikasi yang akan dibangun. Hasil pemodelan yang didapatkan berupa *flowchartchatbot* dan *Entity Relationship Diagram (ERD) chatbot.*

d. Pembentukan *Prototype*

Tahap pembentukan *prototype* merupakan tahap penerjemahan hasil analisis ke dalam bentuk *source code*. Adapun bahasa pemrograman yang digunakan yaitu *AIML* dan *PHP*. *AIML* digunakan untuk membuat struktur pengetahuan dari *chatbot,* sedangkan bahasa *PHP* digunakan pada saat membangun *chatbot* ke dalam *website* DIM ITERA.

Pada proses pembelajaran mesin, penulis menerapkan metode *CBR*. Seperti yang telah dijelaskan pada Bab I dan Bab II. *AIML* memiliki struktur pengetahuan yang statis, artinya struktur ini tidak memungkinkan aplikasi dapat 'belajar sendiri'. Maka, penulis menerapkan proses *CBR* untuk membuat aplikasi dapat belajar. Untuk dapat menjawab pertanyaan baru diluar *pattern* yang telah dibuat pada dokumen *AIML* diterapkan proses *retrieve*. Proses ini mencari similaritas pertanyaan baru dengan pertanyaan yang ada pada dokumen *AIML* dengan menerapkan konsep *regex match*. Untuk memperbaiki jawaban atau proses *revise* pada *CBR* digunakan metode *human-based*.

Pada proses pengujian aplikasi *chatbot*, penulis menerapkan *K-fold cross validation* sebagai metode untuk mengestimasi performansi dari model pelatihan yang telah dibangun [13]. Metode ini akan membagi sekumpulan data pertanyaan dan jawaban menjadi data *training* dan data *testing* sejumlah *k* bagian data. Keluaran metode ini adalah estimasi nilai *threshold* yang tepat untuk proses *retrieve* pada *CBR*.

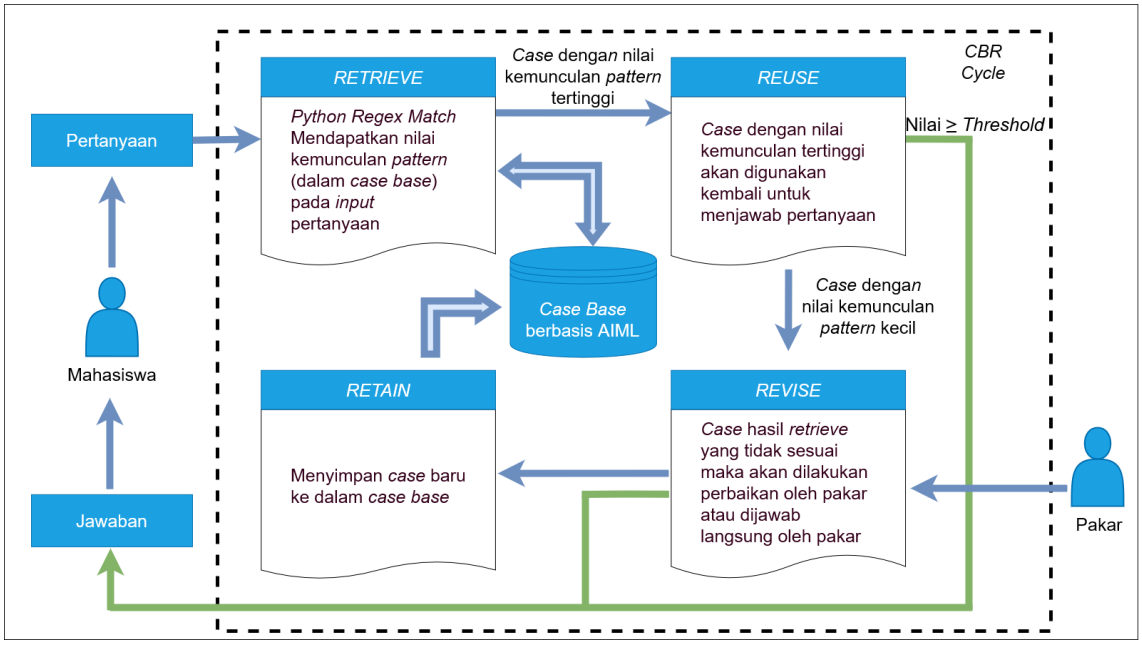
e. Penyerahan *Prototype* Ke Pengguna dan Umpan Balik

Setelah *prototype* berhasil dibangun, selanjutnya *prototype* tersebut diserahkan kepada tim PMB ITERA untuk dilakukan pengujian. Dalam melakukan pengujian, penulis menggunakan metode *blackbox.* Pengujian *blackbox* merupakan pengujian dimana kasus uji didesain berdasarkan spesifikasi dan berfokus pada *output* yang dihasilkan sebagai respon dari *input* yang dipilih dan kondisi – kondisi eksekusi

[7]. Dengan pengujian ini, penguji tidak memerlukan pengetahuan mengenai bahasa pemrograman, sehingga penguji hanya memeriksa fungsionalitas dari aplikasi. Selain itu, dengan metode *blackbox*, akan membuat penguji dan penulis menyamakan pendapat atau persepsi mengenai *chatbot* yang sedang dibangun. Setelah mendapatkan hasil pengujian tersebut, penulis lalu melakukan evaluasi sistem *chatbot* dengan melakukan perhitungan *precision, recall* dan *accuracy*. Untuk mengukur presisi, akurasi dan *recall*, maka akan dibuat *confusion matrix* berdasarkan hasil pengujian *blackbox*. Setelah pengujian selesai, maka tim PMB ITERA memberikan umpan balik berupa kekurangan aplikasi. Berdasarkan umpan balik, penulis akan melakukan perbaikan terhadap *chatbot*. Jika masih terdapat kekurangan atau hal – hal yang ingin ditambahkan, maka akan kembali ke tahap pembentukan *prototype*.

III.3. Arsitektur Umum

Gambar 3.1. menggambarkan rancangan umum pada aplikasi *chatbot.* Mahasiswa memberikan pertanyaan melalui aplikasi. Pertanyaan tersebut lalu masuk proses *retrieve*. Proses ini menghasilkan nilai kemiripan antara pertanyaan (*input*) dengan pertanyaan dalam *case base*. Apabila nilai kemiripan lebih dari *threshold* yang telah ditentukan, maka jawaban dari pertanyaan (*case base*) akan dikirimkan ke mahasiswa sebagai jawaban, proses ini dinamakan *reuse*. Apabila nilai kemiripan kurang dari *threshold*, maka pertanyaan (*input*) akan disimpan terlebih dahulu, lalu kemudian dijawab langsung oleh pakar, proses ini dinamakan *revise*. Nilai *threshold* bergantung dari banyaknya kata dalam *pattern* yang dibuat. Semakin banyak kata dalam *pattern*, maka semakin kecil *threshold* yang diberikan. Pertanyaan baru dari *revise* sebelumnya akan disimpan ke dalam *case base* guna meningkatkan pengetahuan *chatbot* dalam menjawab setiap pertanyaan yang diberikan. Proses penyimpanan kasus baru dinamakan *retain*.

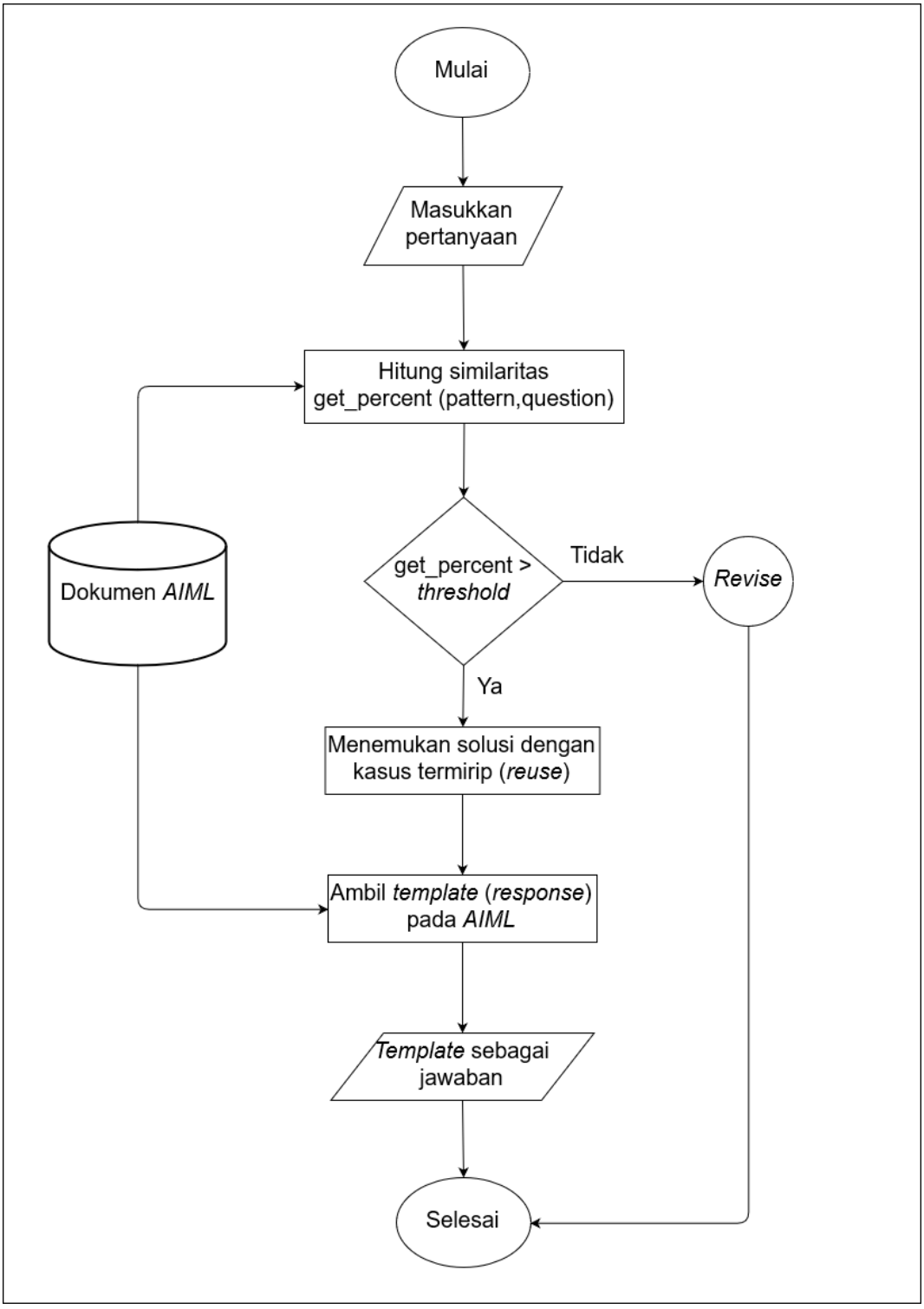


Gambar 3.1. Rancangan Umum

Setiap proses dalam *CBR Cycle* yang ditunjukkan pada Gambar 3.1. memiliki proses tersendiri. Adapun penjelasannya sebagai berikut :

1. Alur *Retrieve* ke *Reuse*

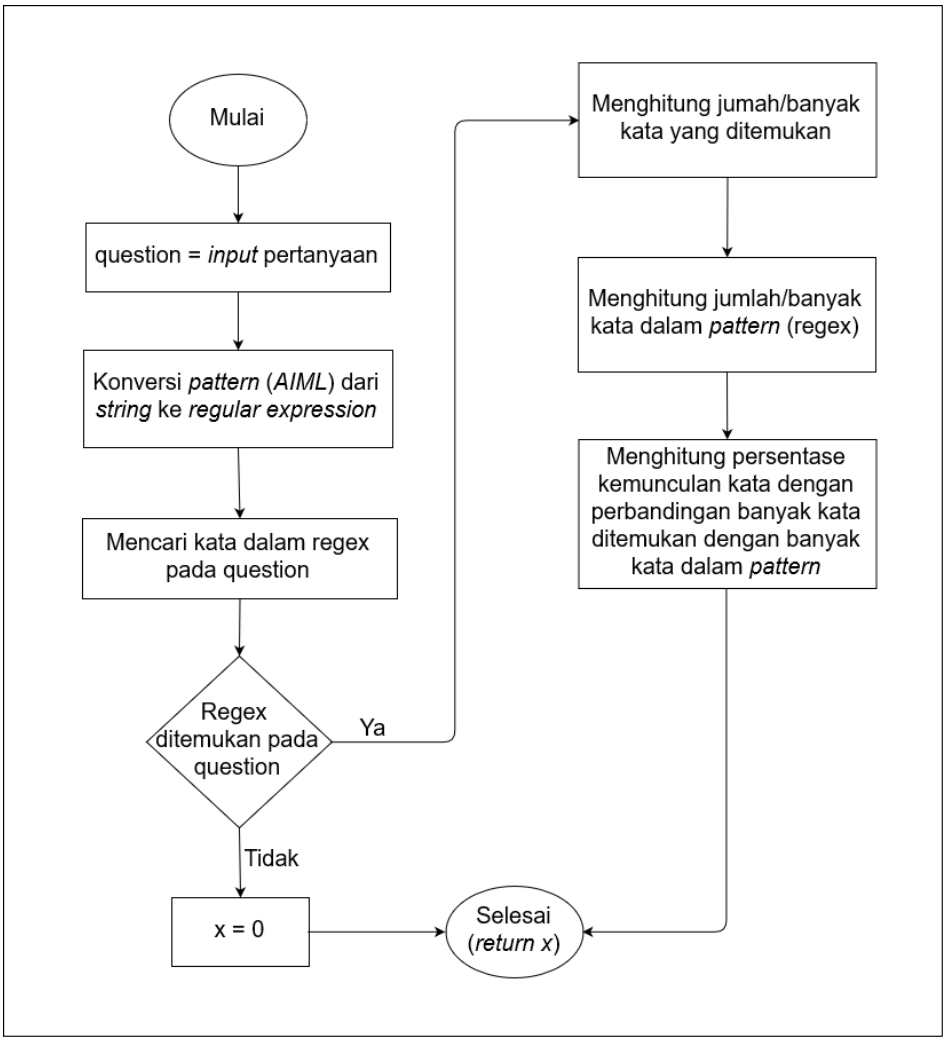
   Proses dimulai dari masukan berupa pertanyaan dari pengguna. Pertanyaan tersebut akan dicari kemiripannya dengan pertanyaan (kasus) yang sudah tersimpan dalam *case base* (*AIML*). Fungsi get_percent akan menghasilkan nilai kemiripan pertanyaan (masukan) dengan setiap pertanyaan yang tersimpan dalam *case base*. Apabila nilai kemiripan tersebut melebihi *threshold*, maka solusi dari kasus termirip (nilai get_percent tertinggi) akan dipakai kembali (reuse). Jika tidak ditemukan nilai melebihi *threshold* maka dilanjutkan ke proses *revise*. Alur ini digambarkan pada Gambar 3.2.



Gambar 3. 2. Diagram Alur Proses *Retrieve*

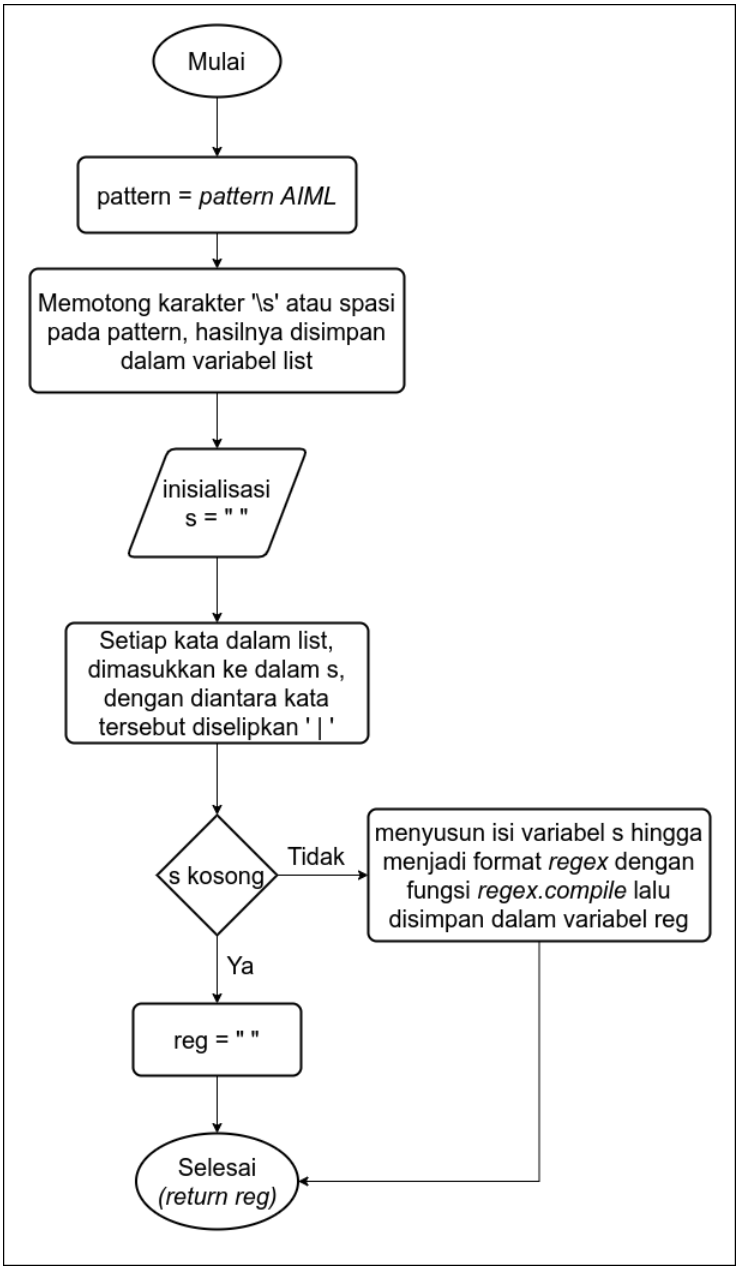
2. Menghitung Nilai Similaritas

Fungsi get_percent menghasilkan bilangan desimal dengan 2 angka di belakang koma. Fungsi ini menggunakan konsep *regular expression* (*regex*). Masukan pertanyaan dari pengguna menjadi parameter 1 dalam fungsi ini. Parameter 2 berupa *pattern* dari dokumen *AIML* (*case base*) yang dikonversikan ke dalam format *regex* dengan menggunakan fungsi make_regex. Kemudian dengan fungsi re.findall, maka akan dicari kata – kata dalam *pattern* pada pertanyaan. Fungsi ini menghasilkan deretan kata yang ditemukan pada pertanyaan. Lalu dengan membandingkan banyak kata yang ditemukan dengan jumlah kata dalam *pattern*, maka akan menghasilkan persentase kemiripan *pattern* dengan pertanyaan. Alur ini digambarkan pada Gambar 3.3.



Gambar 3.3. Diagram Alur Fungsi get_percent

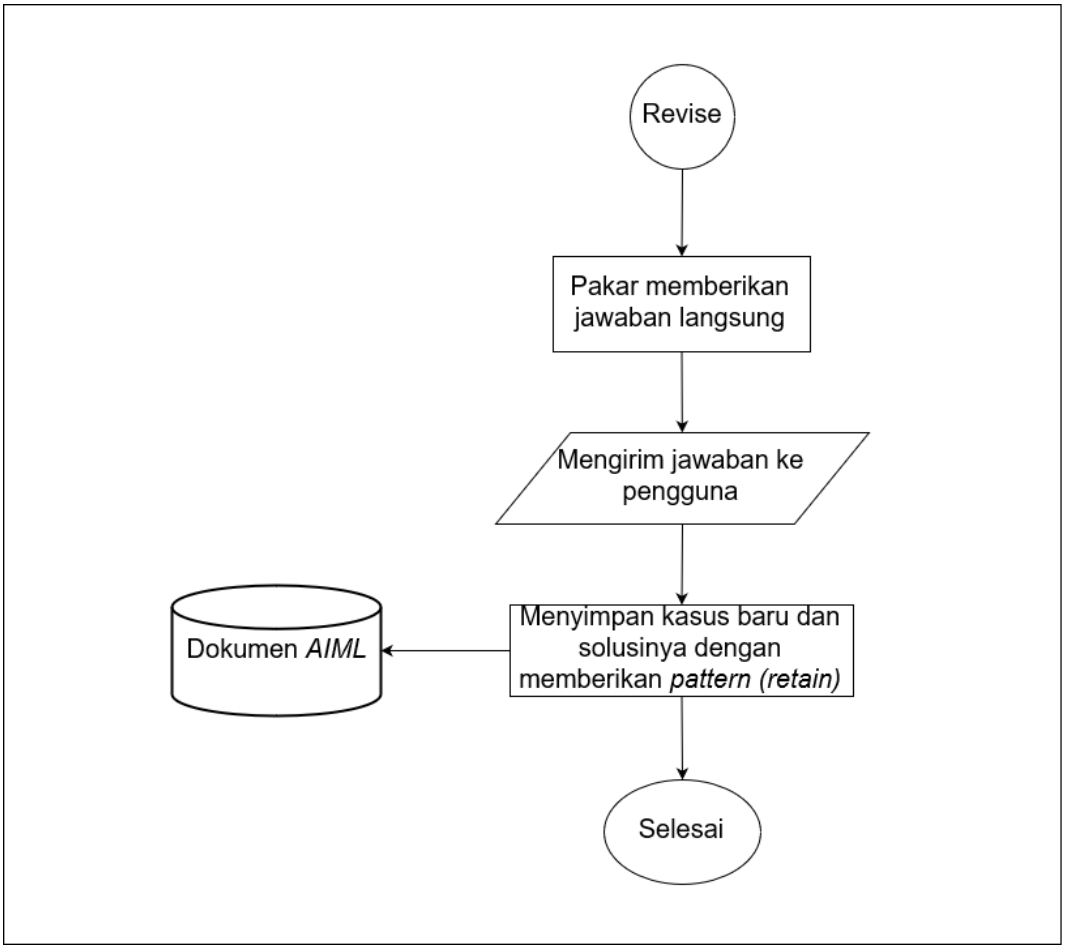
Pada Gambar 3.3. digambarkan bahwa terdapat proses untuk mengkonversi *pattern AIML* ke dalam format *regular expression* yang disebut make_regex. Hal ini bertujuan untuk memudahkan proses pencarian kata yang terdapat dalam *pattern,* pada masukan pertanyaan. Fungsi ini menerima masukan *pattern AIML* berupa *string*, yang kemudian kalimat tersebut dipisahkan setiap katanya lalu diselipkan '|'. Setelah itu, disusun kembali dengan fungsi re.compile untuk dijadikan dalam format *regex.* Alur ini digambarkan dalam Gambar 3.4.



Gambar 3. 4. Diagram Alur Fungsi make_regex

3. Alur *Revise* ke *Retain*

   Pada fungsi get_percent, apabila mengembalikan nilai yang kurang dari *threshold*, maka pertanyaan tersebut akan dijawab langsung oleh tim pakar. Pertanyaan (masukan) dan jawaban (dari pakar) menjadi kasus baru yang kemudian akan disimpan dalam *case base*. Alur ini digambarkan pada Gambar 3.5.



Gambar 3. 5. Diagram Alur Revise ke Retain