BAB 3

ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisis masalah yang menjadi fokus dalam pembangunan sistem *question answering* dan juga alternatif pendekatan yang digunakan sebagai solusi dari masalah-masalah yang ada. Pembahasan akan dimulai dengan menganalisis masalah yang mungkin ditemui dalam setiap komponen dan metode atau pendekatan yang mungkin dilakukan pada komponen-komponen tersebut.

3.1 Analisis Masalah

Tingkat kebutuhan masyarakat Indonesia akan informasi yang mudah didapatkan semakin meningkat. Termasuk didalamnya adalah informasi tentang dunia farmasi, khususnya informasi mengenai obat-obatan yang beredar di Indonesia. Ada banyak buku referensi yang bisa dijadikan acuan untuk mendapatkan informasi yang lengkap tentang obat-obatan, salah satunya adalah buku ISO (Informasi Spesialite Obat Indonesia). Namun di era sekarang yang sudah serba digital dengan tuntutan informasi yang lebih mudah diakses, buku bukan lagi menjadi pilihan pertama.

Internet telah menjadi pilihan utama dalam mencari informasi, termasuk informasi tentang kefarmasian. Namun dengan melimpahnya informasi yang ada di internet, maka perlu waktu lagi untuk mencari informasi yang tepat diperlukan. Oleh karenanya, muncul suatu sistem tanya jawab atau Question Answering System (QAS) yang bisa menjawab kebutuhan informasi penggunanya dengan cepat dan mudah serta lebih menarik.

3.1.1 Analisis Masalah Pada Question Answering System

Pada tugas akhir ini, sistem *question answering* yang dibangun adalah sistem *question answering* dalam bahasa indonesia dengan domain pengetahuan kefarmasian khususnya obat-obatan. Dari hasil pencarian singkat tentang

pengetahuan obat-obatan di Indonesia dengan bahasa Indonesia di Internet ternyata masih sedikit sekali tersedianya situs atau aplikasi *mobile* yang khusus memuat lengkap seputar obat-obatan, adapun pengetahuan kefarmasian obat-obatan yang lengkap tersedia pada buku, salah satunya terdapat pada buku ISO (*informasi Spesialite Obat*). Isi dari buku ISO ini adalah informasi dari setiap merk obat lengkap dengan isi zat aktif, penggolongan obat, kegunaan obat dan dosis yang tentu berguna untuk pengguna obat dan oleh tenaga kefarmasian.

Mengacu pada penelitian yang dilakukan oleh mizuno dkk.[15], Dalam penelitian tersebut didapatkan bahwa di dunia nyata, pertanyaan yang terlihat seperti *factoid* ternyata mendapatkan jawaban yang memiliki penjelasan seperti pada pertanyaan *non-factoid*. Oleh karena itu dalam penelitian tersebut dikembangkan sistem QA yang dapat menjawab pertanyaan seperti pada dunia nyata (*real world question*) dan yang dinamakan *universal question answering*[15]. Hal ini membuat penulis ingin mengimplementasikan sistem *Question Answering* yang dapat menjawab pertanyaan seperti pada dunia nyata (*real world question*) namun spesifik hanya pada domain kefarmasian obat-obatan.

Berdasarkan studi literatur yang telah diuraikan pada BAB 2, dapat disimpulkan bahwa terdapat tiga pendekatan dalam mencari jawaban atas pertanyaan pada sistem QA, yaitu:

1. Jawaban diekstrak atau diambil dari dokumen yang diperkirakan relevan dengan query atau pertanyaan yang diajukan.

2. Jawaban ditelusuri dari basis data jawaban yang saling berhubungan. Seperti halnya yang digunakan pada metode analisis semantik.

3. Jawaban diambil dari kasus yang paling mirip dengan pertanyaan yang diajukan.

Cara mendapatkan jawaban juga berkaitan dengan jenis domain yang dipilih dalam pembangunan sistem QA. Berdasarkan domain masalah, QA dapat

dikelompokan menjadi *closed domain* dan *open domain.* Selain kedua jenis ini, ada juga QA yang dapat menangani *closed domain* dan *open domain* sekaligus atau disebut *closed-open domain.*

Pendekatan pencarian jawaban yang cocok digunakan pada open domain QA adalah mengekstrak jawaban dari kumpulan dokumen yang diperkirakan relevan. Dengan menggunakan pendekatan ini, pencarian jawaban akan lebih luas dan hampir tidak terbatas, sesuai dengan sifat *open domain.*

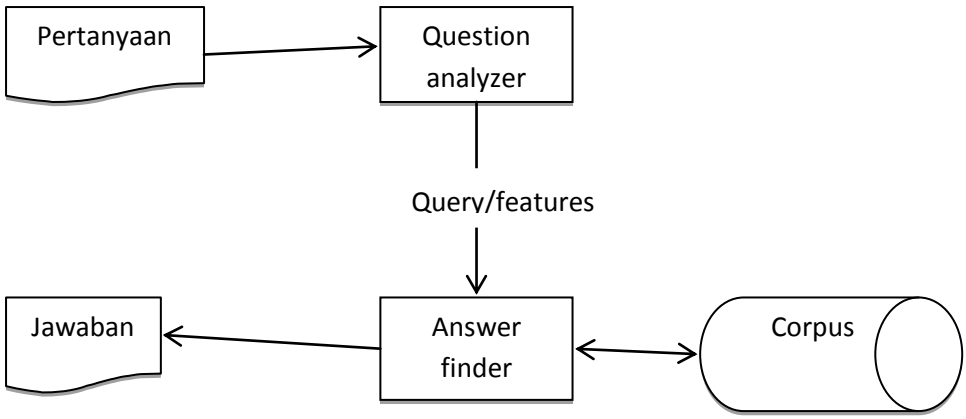
Untuk *closed domain* dibutuhkan teknik pencarian jawaban yang mampu menggunakan kapan suatu istilah digunakan secara umum dan kapan digunakan dalam bidang tertentu. Pendekatan yang dapat digunakan adalah dengan menelusuri basis data jawaban menggunakan metode analisis semantik. Dengan metode ini, sistem dapat membedakan penggunaan kata dalam kalimat yang berbeda sesuai dengan representasi semantiknya.

Pada *closed domain QA*, salah satu pendekatan pencarian jawaban yang sesuai adalah dengan mencari jawaban pada kumpulan kasus yang sudah pernah ditangani. Hal ini karena pertanyaan pada *closed domain* lebih spesifik dan permasalahannya kebanyakan membutuhkan jawaban yang lebih detil.

Berdasarkan keterkaitan antara pendekatan / teknik pencarian jawaban dan domain yang ditangani, pendekatan yang cocok untuk digunakan pada tugas akhir ini adalah dengan mencari jawaban pada basis data jawaban yang telah dibuat. Pertanyaan pada domain pengetahuan kefarmasian khusus obat-obatan membutuhkan jawaban yang jelas dan benar. Oleh karena itu pencarian jawaban lebih cocok menggunakan kumpulan data yang berisi pola pertanyaan dan jenis jawaban yang sudah dibuat berdasarkan buku ISO. Untuk itu, dibutuhkan *corpus* yang berisi dokumen-dokumen pola pertanyaan beserta jawabannya.

Dalam pembangunan sistem QA ini, terdapat dua komponen utama yang harus diimplementasikan untuk menunjang analisis diatas, yaitu komponen *question analyzer* dan komponen *answer finder*. Komponen *question analyzer* berfungsi untuk mengekstrak EAT (*Expected Answer Type*) dan kata-kata kunci dari

pertanyaan. Komponen *answer finder* berfungsi untuk menemukan dan mengambil jawaban dari EAT yang didapat dari pertanyaan. Arsitektur QA yang dibangun dapat dilihat pada gambar 3.1.



Gambar 3.1 Ilustrasi Arsitektur QA yang dibangun

3.1.1.1 Analisis Pada Komponen *Question Analyzer*

Komponen *question analyzer* berfungsi untuk mengekstrak informasi-informasi yang diperlukan untuk mencari jawaban dari pertanyaan yang diajukan. Beberapa informasi yang lazim menjadi keluaran dari komponen ini antara lain adalah tipe jawaban yang diharapkan / *Expected Answer Type* (EAT) / question type, kata-kata kunci, dan fokus pertanyaan. Lalu permasalahan yang ada pada komponen ini biasanya dibagi menjadi dua bagian, yaitu penentuan tipe pertanyaan dan penentuan topik / fokus pertanyaan yang dampaknya pada penentuan kata kunci pertanyaan.

3.1.1.1.1 Penentuan Tipe Pertanyaan / Question Type

Tipe pertanyaan yang akan ditangani adalah tipe yang mendekati tipe pertanyaan *real world question*. Dari pengamatan didapatkan bahwa pertanyaan-pertanyaan yang diajukan biasanya membutuhkan jawaban yang benar dan mengandung penjelasan secara singkat ataupun yang lengkap, baik itu berupa

pertanyaan factoid ataupun pertanyaan non-factoid. Misalnya untuk pertanyaan “bagaimana dosis dari zevit-c?” atau “apakah kegunaan dari flamar?”.

Berdasarkan hasil studi literatur, khususnya pada subbab 2.3.1, terdapat beberapa metode yang dapat digunakan untuk menentukan tipe pertanyaan ini, antara lain metode *pattern matching,* metode *machine learning,* dan analisis semantik. Masing-masing metode akan dijelaskan analisis kelebihan dan kekurangannya sebagai berikut:

1. Metode pattern matching

   Kelebihan pada metode pattern matching / pencocokan pola diantaranya adalah sebagai berikut:

   a. Mudah untuk diimplementasikan karena hanya perlu menyediakan rule dengan kata kunci dan atau kata tanya tertentu.

   b. Proses yang dilakukan cukup mudah dan cepat

   c. Sederhana

   Sedangkan kekurangan dari metode ini adalah:

   a. Perlu dilakukan pengamatan yang cukup untuk memperoleh pola yang akurat dan merepresentasikan klasifikasi kalimat

   b. Pembuatan pola dilakukan secara manual

2. Metode parsing

   Beberapa kelebihan dari metode ini adalah sebagai berikut:

   a. Menghasilkan susunan dan posisi / peran masing-masing kata

   b. Lebih akurat dalam menilai maksud dan topik suatu kalimat

   Kekurangan dari metode ini adalah:

   a. Memerlukan parser khusus berbahasa indonesia

b. Akurasi parser bahasa indonesia yang tersedia saat ini masih dibawah 50% [25]

3. Metode machine learning / pembelajaran mesin

   Kelebihan dari metode machine learning diantaranya adalah:

   a. Dari hasil penelitian yang sudah ada, akurasi yang dihasilkan untuk pengekstrakan EAT rata-rata diatas 90% [10].

   b. Algoritma yang tersedia cukup banyak dan tinggal diimplementasikan

   Kekurangannya diantaranya adalah:

   a. Diperlukan banyak data untuk melakukan pembelajaran

4. Metode analisis semantik

   Kelebihan metode analisis semantik adalah:

   a. Menghasilkan posisi kata pada suatu kalimat

   b. Topik atau maksud suatu kalimat bisa didapatkan dengan lebih akurat walaupun kata-katanya berbeda susunan

   Kekurangan:

   a. Memerlukan analyzer khusus bahasa indonesia. Akurasi *resource* yang tersedia saat ini masih rendah, yaitu 22% [12].

Tabel 3.1 Daftar aturan pola EAT

EAT	Kata Tanya	Kata kunci
Farmakoterapi	Bagaimana, bagaimanakah, apa, apakah	Farmakologi, terapi
Nama obat	Apa, apakah	Nama, nama obat, merk, merk dagang, nama dagang
Penggolongan obat	Bagaimana, bagaimanakah, apa, apakah	Penggolongan, penggolongan obat, golongan, golongan obat
Zat isi	Apa, apakah	Zat isi, zat berkhasiat, kandungan, kandungan utama
Perusahaan pembuat	Apa, apakah, siapa, siapakah	Nama, nama perusahaan, industri, pembuat, industri pembuat, pabrik, pabrik pembuat
Indikasi	Apa, apakah, bagaimana, bagaimanakah, jelaskan, jelaskanlah	Kegunaan, manfaat, indikasi, indikasi obat, akibat, guna
Efek Samping	Apa, apakah, bagaimana, bagaimanakah	Efek samping
Kontra Indikasi	Siapa, apa, apakah, adakah, bagaimana, bagaimanakah	Kontra indikasi, kontra, tidak boleh, dilarang untuk, dilarang, tidak diperbolehkan
Dosis	Bagaimana, bagaimanakah	Dosis, penggunaan, cara pakai, cara pemakaian, dosis pemakaian

Kemasan	bagaimana, bagaimanakah, apa, apakah, apa saja	bentuk, bentuk kemasan, kemasan, sediaan
Harga	Berapa, berapakah	Harga, harga obat

Dari alternatif-alternatif metode tersebut, metode yang dipilih adalah metode pencocokan pola. Metode in dipilih karena cukup mudah untuk diimplementasikan, tidak membutuhkan kakas khusus, dan penyediaan pola untuk pertanyaan domain kefarmasian obat-obatan dapat dilakukan dengan lebih cepat. Aturan penentuan EAT dapat dilihat pada tabel 3.1.

3.1.1.1.2 Pemilihan Kata Kunci Dan Penentuan Topik Pertanyaan

Kata kunci adalah kata-kata yang diambil/diekstrak dari pertanyaan yang menggambarkan inti pertanyaan tersebut. Kata-kata kunci ini yang digunakan untuk melakukan pencarian dan pengambilan jawaban yang pertanyaannya memiliki kata kunci yang mirip atau sama. Kata kunci ini berupa kata-kata yang memang terdapat di pertanyaan. Kata-kata kunci yang diperoleh akan menggambarkan topik dari pertanyaan tersebut.

3.1.1.1.3 Analisis Solusi Untuk Komponen *Question Analyzer*

Secara keseluruhan solusi yang akan diterapkan pada komponen *question analyzer* dapat dilihat pada gambar 3.2. Gambar ini menunjukan alur proses yang terjadi pada *question analyzer*.

Pertanyaan yang masuk ke komponen *question analyzer* akan dicocokkan dengan metode pencocokan pola untuk mendapatkan EAT. Adapun proses yang terjadi didalam *question analyzer* ini adalah sebagai berikut:

1. Masukan pertanyaan dari pengguna akan diekstrak menggunakan teknik *parser* untuk mendapatkan penggalan kata-kata penyusun pertanyaan.

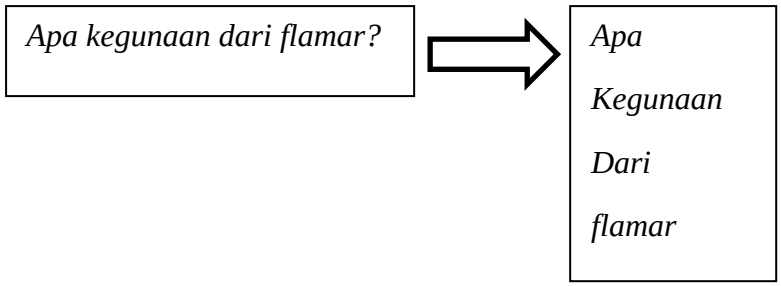
Berikut langkah-langkah umum proses *parsing* dari masukan kalimat pertanyaan “apa kegunaan dari flamar?” :

a. Memotong setiap kata didalam teks dan mengubahnya menjadi huruf kecil.

b. Hanya huruf ‘a’ sampai ‘z’ yang diterima, sedangkan karakter lain selain huruf tersebut seperti ‘?’ ‘.’ Akan dihilangkan atau diabaikan.

c. Hasilnya adalah kata-kata yang merupakan penyusun kalimat yang dimasukkan.

Gambaran teknik *parsing* dan hasilnya dapat dilihat pada gambar 3.2 berikut ini.



Gambar 3.2 Gambaran tahap *parsing*

2. Kemudian kata-kata yang penting (*word list*) atau penyusun dari *matching pattern* akan diambil sehingga kata-kata selain yang tersedia di *word list* dihilangkan atau diabaikan. Tahapan prosesnya adalah sebagai berikut:

   a. Kata-kata yang didapat dari proses *parsing* akan dihitung jumlah katanya, jika lebih dari 0 maka akan dilakukan proses membandingkan dengan kata-kata penting yang ada di *database*.

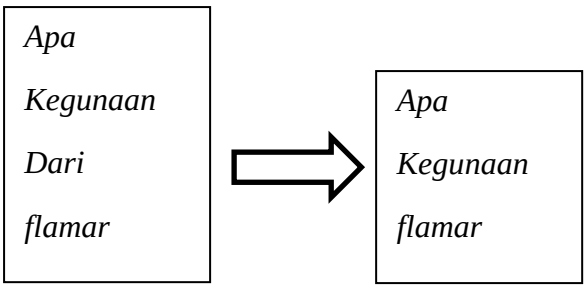
   b. Dilakukan pengecekan apakah kata-kata hasil *parsing* sama dengan kata-kata penting yang ada di *database* atau tidak.

c. Jika sama, maka kata-kata tersebut akan diambil sekaligus kata-kata yang dianggap tidak penting akan diabaikan.

Contoh gambaran yang terjadi pada tahap ini adalah sebagai berikut:

Wordlist : apa, kegunaan, flamar (kata yang akan diambil atau tetap ditampilkan)

Stoplist : dari (merupakan kata yang dianggap tidak penting, akan diabaikan)



Gambar 3.3 tahap *filtering*

3. Dari proses-proses yang telah dilakukan sebelumnya, maka dari masukan kalimat pertanyaan yaitu "apa kegunaan dari flamar?" menghasilkan kalimat akhir "apa kegunaan flamar". Kalimat akhir tersebut kemudian akan dicocokan dengan pola yang telah dibentuk sebelumnya untuk mendapatkan EAT yang sesuai (pola bisa dilihat pada tabel 3.1 diatas)

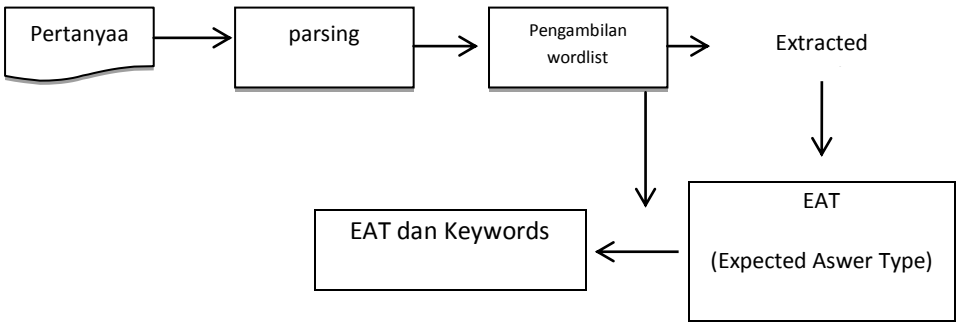
4. Berdasarkan pola yang telah dibuat, maka "apa kegunaan flamar" cocok dengan pola EAT "apa+kegunaan". EAT dari pertanyaan ini adalah indikasi.

5. Selanjutnya setelah EAT yang sesuai beserta nama item (nama obat) sebagai kunci pencarian yang didapatkan dari kalimat pertanyaan, maka tersebut menjadi fitur-fitur yang akan digunakan pada komponen selanjutnya untuk mencari jawaban yang sesuai di *database* yaitu:

EAT = indikasi,

Nama item = flamar.

Berikut adalah gambaran secara keseluruhan dari *question analyzer*.



Gambar 3.4 Proses pada *question analyzer*

Tabel 3.2 contoh pertanyaan yang telah diproses oleh *question analyzer*

Pertanyaan	**EAT**	**Kata kunci**	**Kata item (nama obat)**	**Fitur pencarian**
Apa kegunaan dari flamar?	indikasi	kegunaan	flamar	EAT : indikasi Kata tanya : apa Keywords : kegunaan Nama item : flamar

3.1.1.2 Analisis pada komponen answer finder

Komponen *answer finder* ini berfungsi untuk mengambil jawaban dengan pencocokan fitur-fitur di *corpus/database* dengan fitur-fitur pertanyaan yang telah diproses oleh komponen sebelumnya. Dari kumpulan data tersebut, diambil salah satu data yang cocok yang kemudian dijadikan jawaban bagi pertanyaan yang diajukan. Permasalahan pada komponen ini dapat dibagi menjadi dua bagian yaitu penentuan *corpus* yang berupa kumpulan pasangan EAT+kata item dengan jawaban.

3.1.1.2.1 Penentuan *Corpus*

Corpus adalah kumpulan contoh kasus (case) yang dijadikan sumber pencarian jawaban oleh sistem QA. *Corpus* yang digunakan dalam sistem QA ini adalah kumpulan dokumen pasangan EAT, kata kunci dan jawabannya dalam bahasa Indonesia. Terdapat dua alternatif utama yang dapat digunakan dalam sistem ini dari segi cara pengaksesannya, yaitu:

1. *Corpus offline*

   Corpus offline dapat berupa kumpulan dokumen dalam bentuk file-file teks maupun tersimpan dalam basis data. Khusus dalam tugas akhir ini *corpus* hanya mengacu pada satu sumber, yaitu dengan mengonversi buku ISO (informasi Spesialite Obat) menjadi file-file teks digital.

2. *Corpus online*

   Corpus online dapat berupa kumpulan artikel pada situs web yang diakses secara online. Masalah yang timbul dari metode ini adalah bagaimana cara mengakses bagian artikel dari halaman web tersebut.

Tabel 3.3. Contoh format dokumen dalam *corpus*

EAT	**Kata item**	**Jawaban**
Indikasi	flamar	Kegunaan dari flamar adalah untuk reumatik non artikuler, inflamasi dan bentuk degeneratif reumatik, artritis reumatoid, osteoartritis, spondilitis, ankilosa, spondilartrosis

Dari kedua alternatif diatas, yang diterapkan adalah dengan menggunakan *corpus offline* dari buku ISO, contoh format dokumen kasus dalam corpus dapat dilihat pada Tabel 3.3.

3.1.1.2.2 Pencarian Dan Pengembalian Jawaban Dari *Corpus*

Seperti yang telah diuraikan diawal bab 3 dan berdasarkan pilihan yang diambil untuk membangun *corpus*, pendekatan pencarian jawaban yang cocok untuk sistem QA ini adalah menelusuri *corpus*/*database* yang telah ada atau yang telah dibuat sebelumnya untuk kemudian dilakukan pencocokan dengan kata kunci dan EAT yang dihasilkan dari pertanyaan yang diberikan oleh pengguna sehingga akan mendapat jawaban yang sesuai.

3.1.1.2.3 Analisis Solusi Untuk Komponen *Answer Finder*

Analisis solusi yang digunakan pada komponen *answer finder* adalah dengan melakukan pencocokan antara kata kunci dan EAT yang didapat dari pertanyaan dengan yang telah dibuat sebelumnya di *corpus/database* untuk kemudian diambil jawaban yang telah tersedia dalam *database/corpus* tersebut. Adapun tahapan jelasnya seperti berikut ini.

- Input pertanyaan: “apa kegunaan dari flamar?”
- Output jawaban: “kegunaan dari flamar adalah untuk reumatik non artikuler, inflamasi, dan bentuk degeneratif reumatik, osteoartritis, spondilitis, ankilosa, spondilartrosis” (jawaban didapat dari *corpus/database*)

Sebelum diproses oleh *question analyzer:*:

1. Apa kegunaan dari flamar?
2. Array ([0]=>apa [1]=>kegunaan [2]=>dari [3]=>flamar)

Sesudah diproses oleh *question analyzer*:

1. [Array ([0]=>apa [1]=>kegunaan)] = [EAT = indikasi](didapat dari *pattern matching*)
2. Flamar = kata item.

Setelah melewati tahap-tahap yang ada, didapat hasil yang berupa EAT dan kata kunci untuk kemudian dicocokan dengan EAT dan kata kunci yang telah ada dalam *corpus/database*. Didalam *database/corpus* terdapat tabel yang berisi *field* EAT, kata kunci, dan jawaban yang bisa dilihat pada tabel 3.3 diatas sehingga akan menghasilkan jawaban berikut.

Kegunaan dari flamar adalah untuk reumatik non artikuler, inflamasi dan bentuk degeneratif reumatik, artritis reumatoid, osteoartritis, spondilitis, ankilosa, spondilartrosis

3.2 Penelitian Dan Kajian Terkait.

Disini akan dibahas tentang beberapa penelitian tentang sistem *Question Answering* yang menggunakan bahasa Indonesia dan yang bisa menangani pertanyaan tipe *factoid* dan *non-factoid*.

3.2.1 Pendekatan *Machine Learning* Untuk *Question Answering* Dalam Bahasa Indonesia

Pada penelitian yang dilakukan oleh Purwarianti dkk[10] dilakukan pendekatan pembelajaran mesin untuk membangun sistem *question answering* dalam bahasa Indonesia. Pada umumnya, sistem *question answering* (QA) disusun oleh tiga komponen, yaitu *question analyzer*, *passage retriever* dan *answer finder*. *Question analyzer* memiliki fungsi untuk menentukan kata kunci, menghasilkan EAT (*Expected Answer Type*), dan menghasilkan informasi lainnya yang bisa menggambarkan fokus dari pertanyaan yang diajukan. *Passage retriever* bertugas untuk mencari bagian dari dokumen yang diperkirakan mengandung jawaban dari pertanyaan. Lalu dari keluaran *question analyzer* dan *passage retriever* tersebut, *answer finder* menentukan kandidat jawaban yang cocok dengan EAT yang dihasilkan. Pendekatan QA dengan pembelajaran mesin telah menjadi alternatif untuk menggantikan aturan yang dibuat manual dalam membangun QA. Dalam penelitian ini algoritma pembelajaran mesin yang digunakan adalah SVM.

1. Question analyzer

Question analyzer terbagi menjadi dua komponen, yaitu question shallow parser dan question classifier. Question shallow parser bertujuan untuk mengekstrak keyword, main word, dan phrase label dari kalimat pertanyaan. Fungsi penting dari question shallow parser ini adalah POS tagger.

Question classifier berfungsi mengklasifikasikan pertanyaan menjadi enam buah EAT dengan menggunakan algoritma pembelajaran mesin, yaitu date, location, name, organization, person, dan quantity. Pada question classifier terdapat beberapa atribut yaitu simple rule based class candidate, wordnet distance, dan bi-gram frequency. Pada simple rule based class candidate, beberapa kata tanya sederhana didefinisikan EAT-nya. Pada wordnet distance, kata-kata dalam bahasa Indonesia diterjemahkan kedalam bahasa Inggris dan dicari "kedalamannya" (jarak) dengan kata (synset) yang berhubungan. Atribut yang ketiga adalah bi-gram frequency, yaitu menambahkan preceeding word dari kata pada pertanyaan. Daftar dari kata-kata ini diambil dari korpus.

2. Passage Retriever

Modul passage retriever terdiri dari dua komponen, yaitu document retriever dan passage retriever. Document retriever mengambil dokumen yang relevan berdasarkan kumpulan kata kunci dan sinonimnya lalu passage retriever mengambil bagian dari dokumen yang paling mungkin mengandung jawaban dari pertanyaan.

3. Answer Finder

Pada bagian answer finder ini, SVM digunakan untuk mengklasifikasikan kata-kata didalam korpus, yaitu "B" jika kata merupakan awal dari jawaban, "i" jika merupakan bagian dari jawaban, dan "O" jika bukan merupakan bagian dari jawaban.

Eksperimen dan pengujian dilakukan ketiga komponen sistem yaitu question analyser, passage retriever, dan answer finder. Pada pengujian komponen question

analyser, bagian question classifier didapatkan bahwa algoritma SVM memiliki akurasi tertinggi dibanding algoritma pembelajaran mesin yang lain. Atribut-atribut tambahan lainnya, yaitu bi-gram frequency dan lain-lain juga memberikan tambahan pada akurasi sistem. Lalu pada komponen passage retriever, penggunaan IDF pada dokumen ternyata menunjukan hasil yang lebih baik dibandingkan metode lainnya, misalnya TFxIDF. Lalu hasil pada answer finder menunjukkan bahwa fitur-fitur yang digunakan menunjukan hasil yang cukup memuaskan.

Jadi dapat disimpulkan bahwa penggunaan algoritma pembelajaran mesin dan metode-metode yang dijelaskan pada penelitian ini menjanjikan hasil yang cukup memuaskan untuk pembangunan sistem QA berbahasa Indonesia.

3.2.2. Question Answering System Dan Penerapannya Pada Alkitab

Pada penelitian yang dilakukan oleh Gunawan dkk [27] dibangun sebuah sistem *question answering closed domain* yang menggunakan Alkitab berbahasa Inggris versi *World English Bible* (WEB) sebagai sumber informasi untuk menjawab pertanyaan. Hal ini menjadikan sistem *question answering* terbatas hanya untuk memberikan jawaban dari pertanyaan-pertanyaan dengan ruang lingkup informasi yang tersimpan dalam Alkitab itu sendiri. Selain itu, pertanyaan juga dibatasi pada tiga buah tipe jawaban yaitu *person, location, date*. Arsitektur *question answering system* yang dibangun terdiri dari enam tahapan proses, yaitu penganalisis pertanyaan, *preprocessing* koleksi dokumen, pemilihan kandidat jawaban, analisis kandidat dokumen, ekstraksi jawaban, dan pemberian respon.

1. Question Analysis

Pertanyaan yang diinputkan oleh user harus memenuhi beberapa persyaratan tertentu seperti batasan penggunaan kosa kata dan bahasa yang baku. Tahap *question analysis* ini akan menghasilkan dua buah representasi pertanyaan sebagai output. Representasi pertama yaitu berupa query yang akan diteruskan pada tahap *candidate document selection*, dan representasi yang kedua merupakan representasi semantik dari pertanyaan.

Adapun langkah-langkah yang dapat dilakukan untuk menghasilkan query ialah: ekstraksi keyword dari pertanyaan, mengkombinasikan keyword dengan menggunakan operator tertentu, dan mengembangkan query, misalnya dengan menemukan sinonim dan/atau variasi morfologi dari setiap keyword. Sebagai contoh, dari pertanyaan: "*Who won the 1998 Nobel Peace Prize?"* dapat dihasilkan daftar keyword query: {won|win|gain|gained|get|got|acquire|acquired| obtain|obtained, 1998, Nobel, Peace, Prize}.

Representasi semantik dari pertanyaan mengarah pada jenis informasi atau tipe jawaban yang diinginkan oleh pertanyaan. Misalnya, pertanyaan "*When*.." menginginkan jawaban berupa *Date* (tanggal) atau *Time* (waktu), "*Where*.." menanyakan *Location* (Lokasi), dan "*Who*.." menginginkan jawaban yang berupa *Person* (orang) atau *Organization* (organisasi). Adapun teknik pendekatan yang bisa dipakai dalam proses analisis pertanyaan adalah representasi semantik, pencocokan pola, pembelajaran mesin dan metode *parsing*.

2. Document Collection Preprocessing

Proses kedua yang menyusun arsitektur umum QA system adalah preprocessing koleksi dokumen. Proses ini bertujuan untuk mempermudah akses pada koleksi dokumen dalam upaya menemukan jawaban yang tepat atas pertanyaan user dalam waktu yang sesingkat mungkin. Sebagian besar QA system yang berpartisipasi pada TREC *question answering track* melakukan *preprocessing* dengan mengandalkan *document indexing engine*. Akan tetapi, tahap *preprocessing* tidak hanya terbatas pada *indexing* saja. Ada tiga buah alternatif lain yang dapat digunakan dalam proses *preprocessing* yaitu *part-of-speech tagging* (POStagging), *Named Entity recognition*, dan pengubahan teks dokumen kedalam bentuk logika atau bentuk lain agar mudah diakses seperti *T-expression*.

3. Candidate Document Selection

Dengan berdasarkan pada query yang dihasilkan dari proses analisis pertanyaan, tahap candidate document selection akan mengidentifkasi dokumen-

dokumen yang mungkin mengandung jawaban atas sebuah pertanyaan. Proses ini biasanya diselesaikan dengan bantuan satu atau lebih search engine, baik yang sudah ada, maupun yang dikembangkan sendiri sesuai dengan kebutuhan sistem.

4. Candidate Document Analysis

Dokumen-dokumen yang dikembalikan oleh search engine akan dianalisis lebih lanjut dengan tujuan untuk memperkecil ukuran dokumen yang nantinya akan diakses untuk ekstraksi jawaban. Proses analisis kandidat dokumen ini sebenarnya tidak diperlukan lagi apabila sistem telah melaksanakan preprocessing secara lengkap terhadap semua dokumen, atau analisis ini memang tidak didesain sebagai bagian dari sebuah QA system. Umumnya, QA system melakukan analisis kandidat dokumen dengan menerapkan teknik yang serupa dengan alternatif teknik yang dapat digunakan pada tahap preprocessing koleksi dokumen. Sebagai tambahan, dapat juga dilakukan pemisahan kalimat (sentence splitting), paragraf, atau bagian teks tertentu terhadap setiap dokumen.

5. Answer Extraction

Tahap answer extraction bertugas untuk mencocokkan kandidat dokumen dengan representasi semantik dari pertanyaan, dan kemudian menghasilkan daftar jawaban yang telah diurutkan berdasarkan probabilitas kebenarannya. Proses pertama yang dilakukan pada tahap ini adalah menemukan sebuah unit teks (mungkin berupa kalimat apabila pada proses sebelumnya dilakukan sentence splitting) dari kandidat dokumen yang mengandung string dengan tipe semantik sesuai dengan tipe jawaban yang diharapkan. Selanjutnya, sejumlah batasan akan diterapkan pada unit teks yang ditemukan, seperti kesesuaian antara term (keyword) query dengan term penyusun unit teks. Penerapan batasan ini akan menentukan kelayakan dari setiap unit teks untuk dipertimbangkan sebagai kandidat jawaban.

6. Response Generation

Setelah sistem berhasil mengekstrak jawaban, maka proses selanjutnya ialah menentukan respon yang akan dihasilkan, yaitu bagaimana jawaban tersebut akan disajikan. Bentuk penyajian jawaban dari QA system yang berbeda tentunya juga berbeda. Ada sistem yang menyajikan daftar dokumen (contoh: AskJeeves), daftar paragraf (contoh: IONAUT), daftar kalimat (contoh: AnswerBus), atau daftar frase jawaban (contoh: Mulder) sebagai respon.

Penelitian dan aplikasi QA system yang dibuat masih dapat dikembangkan lagi dengan menggali lebih dalam mengenai tugas-tugas pemahaman bacaan, karena dirasakan bahwa QA system yang dikembangkan dengan menggunakan Alkitab sebagai sumber informasi akan menjadi lebih sempurna apabila sistem terlebih dahulu melakukan pemahaman terhadap kisah-kisah yang terdapat dalam pasal dan ayat-ayat Alkitab. Untuk memperbesar kemungkinan ditemukannya jawaban, pada tahap preprocessing dari WEBqa dapat dilakukan pengenalan anaphora. Selain itu, perumusan yang digunakan untuk ekstraksi jawaban pada WEBqa dapat diubah atau diganti untuk lebih memperhitungkan kedekatan antar keyword penyusun query pada kandidat kalimat jawaban.

3.2.3. Eksperimen non-factoid question answering pada NTCIR-6: menuju answer type detection untuk real world questions

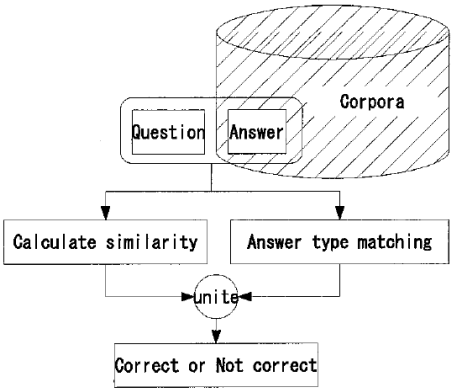
Dalam penelitian yang dilakukan oleh Mizuno dkk [15] dibahas tentang upaya pembangunan sistem question answering (QA) yang dapat menangani *real world questions* atau yang mereka sebut sebagai *universal question answering*. *Real world questions* disini adalah seluruh jenis pertanyaan yang tampak sebagai pertanyaan factoid memiliki jawaban yang tidak pendek (eksak) namun memiliki penjelasan didalamnya.

Penelitian ini difokuskan pada pengembangan sistem yang memiliki fokus kepada deteksi tipe jawaban (answer type detection). Sistem ini dikembangkan dengan menggunakan pendekatan pembelajaran mesin untuk mengecek apakah tipe yang diharapkan dari pertanyaan konsisten dengan tipe dari jawaban.

Pendekatan ini juga dikembangkan untuk mengembangkan sistem yang dapat menjawab semua jenis pertanyaan baik *factoid* maupun *non-factoid* atau yang disebut *Universal Question Answering* (UQA) tadi.

Pada penelitian ini digunakan sebuah parameter khusus yang disebut answer type. Hal ini dimaksudkan agar tipe jawaban yang diinginkan tidak hanya diambil dari sisi pertanyaan namun juga dari sisi jawaban. Jadi *answer type* diambil dari sisi pertanyaan dan juga sisi (kandidat) jawaban. Jika kedua *answer type* pertanyaan dan *answer type* jawaban sama, maka pertanyaan dan jawaban ini cocok atau itulah jawaban dari pertanyaan tersebut. Selain menetapkan parameter *answer type*, dalam penelitian ini juga dilakukan analisis terhadap kategori-kategori yang sering ditanyakan pada *Real World Questions* seperti *sport, hobby, lifestyle*, dsb. Dari masing-masing kategori tersebut didapatkan sebaran *answer type* yang sering ditanyakan.

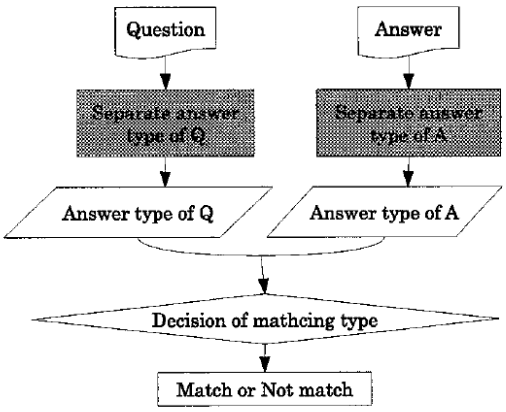
Untuk menilai suatu jawaban benar atau tidak diberikan dua parameter, yaitu topik jawaban sama dengan topik pertanyaan dan juga *answer type* pertanyaan sama dengan *answer type* jawaban. Struktur umum dari sistem ini dapat dilihat pada Gambar 3.3.



Gambar 3.5 Pendekatan sistem UQA [15]

Untuk merealisasikan pendekatan ini dikembangkan dua metode yaitu **metode 1** dan **metode 2.** Pada **metode 1**, pencocokan *answer type* pertanyaan dan jawaban dilakukan pada sebuah proses dimana penentuan *answer type* dari jawaban dan pertanyaan tersebut dilakukan secara terpisah. Sebagai contoh, misalnya *answer type* dari pertanyaan adalah *Reason* dan *answer type* dari jawaban adalah *Reason*, maka jawaban ini cocok dengan pertanyaannya. Pada setiap *answer type* digunakan *binary classifier* untuk mendeteksi apakah ini adalah tipe yang diharapkan dari pertanyaan atau kandidat jawaban. Untuk *classifier* digunakan *Support Vector Machine* (SVM) dan untuk *feature vector* digunakan word uni-grams. Ilustrasi dari proses ini dapat dilihat pada gambar 3.3.

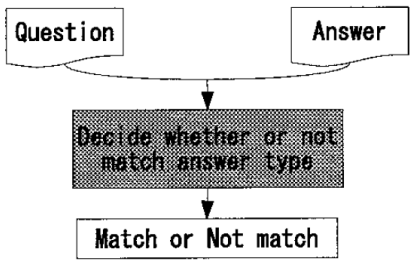
Untuk **metode 2**, pencocokan langsung dilakukan hanya satu kali yaitu langsung kepada pertanyaan dan kandidat jawaban, sehingga dalam metode ini hanya memerlukan satu *binary classifier* saja. Ilustrasi dapat dilihat pada gambar 3.4.



Gambar 3.6 Konfigurasi metode 1 UQA [15]

Eksperimen bagi sistem yang dibangun ini dilakukan pada dua jenis kumpulan dokumen, yaitu menggunakan *collection test* dari NTCIR *Question Answering Challenge 4* (QAC4) dan juga pada *website.* Untuk menilai performansi sistem digunakan *Mean Reciprocal Rank* (MRR). Berdasarkan hasil eksperimen

didapatkan bahwa **metode 2** lebih baik dibandingkan metode 1 dan secara umum sistem ini bekerja cukup memuaskan.

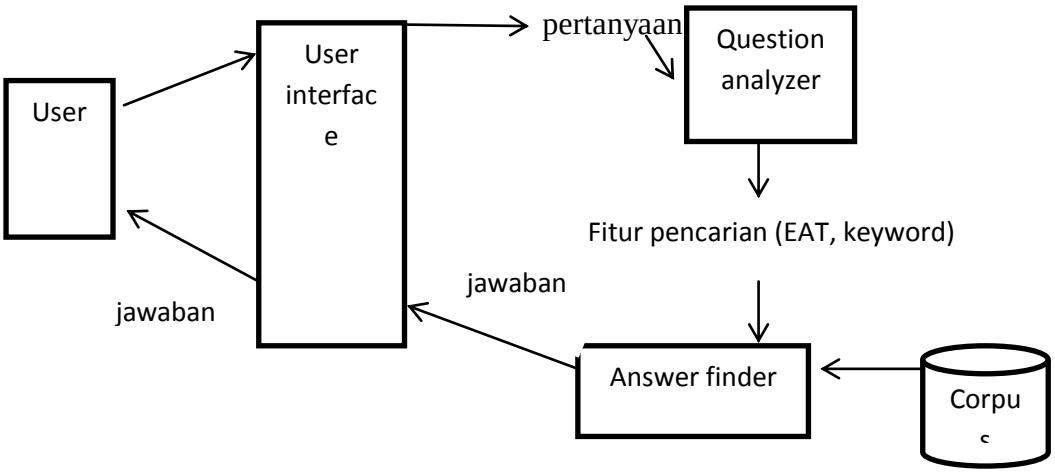


Gambar 3.7 Konfigurasi metode 2 UQA [15]

3.3. Analisis Sistem

3.3.1. Deskripsi Umum Sistem

Perangkat lunak yang dibangun adalah aplikasi *mobile android* yang dapat memberikan jawaban atas pertanyaan dari pengguna dalam bahasa Indonesia. Domain yang ditangani oleh sistem ini adalah domain kefarmasian obat-obatan, jadi sistem ini hanya menerima pertanyaan dan menyediakan jawaban dalam bidang kefarmasian obat-obatan. Sistem menerima pertanyaan lalu mencari jawaban dalam *database/corpus* yang berupa pasangan dokumen kata kunci, EAT dan jawaban. Sistem kemudian akan menampilkan jawaban yang diambil dari hasil pencocokan antara EAT dan kata kunci yang didapat dari input pertanyaan dengan EAT dan kata kunci yang sudah tersedia di *corpus*. Arsitektur perangkat lunak dapat dilihat pada gambar 3.6 berikut ini.



Gambar 3.8 Arsitektur perangkat lunak

3.3.2. Deskripsi Masukan

Masukan pada sistem QA ini adalah pertanyaan dalam bahasa Indonesia. Tipe pertanyaan yang digunakan adalah *real world question* jadi tidak membedakan antara pertanyaan factoid dan non-factoid. Pertanyaan harus meliputi domain kefarmasian khusus mengenai obat-obatan dan menggunakan kata-kata bahasa Indonesia yang baku. Contoh pertanyaan untuk setiap EAT pertanyaan dapat dilihat pada tabel 3.4

Tabel 3.4 contoh pertanyaan masukan untuk masing-masing EAT

EAT	Kata Tanya	Kata kunci
Farmakoterapi	Bagaimana, bagaimanakah, apa, apakah	Farmakologi, terapi
Nama obat	Apa, apakah	Nama, nama obat, merk, merk dagang, nama dagang
Penggolongan obat	Bagaimana, bagaimanakah, apa, apakah	Penggolongan, penggolongan obat, golongan, golongan obat
Zat isi	Apa, apakah	Zat isi, zat berkhasiat, kandungan, kandungan utama

Perusahaan pembuat	Apa, apakah, siapa, siapakah	Nama, nama perusahaan, industri, pembuat, industri pembuat, pabrik, pabrik pembuat
Indikasi	Apa, apakah, bagaimana, bagaimanakah, jelaskan, jelaskanlah	Kegunaan, manfaat, indikasi, indikasi obat, akibat, guna
Efek Samping	Apa, apakah, bagaimana, bagaimanakah	Efek samping
Kontra Indikasi	Siapa, apa, apakah, adakah, bagaimana, bagaimanakah	Kontra indikasi, kontra, tidak boleh, dilarang untuk, dilarang, tidak diperbolehkan
Dosis	Bagaimana, bagaimanakah	Dosis, penggunaan, cara pakai, cara pemakaian, dosis pemakaian
Kemasan	bagaimana, bagaimanakah, apa, apakah, apa saja	bentuk, bentuk kemasan, kemasan, sediaan
Harga	Berapa, berapakah	Harga, harga obat

3.3.3. Deskripsi proses

Proses yang dilakukan terbagi menjadi dua proses utama, sesuai dengan jumlah komponen penyusun sistem QA, yaitu:

1. Proses analisis pertanyaan

   Pada proses ini pertanyaan yang dimasukkan oleh pengguna akan dianalisis untuk mengambil tipe pertanyaan / *Expected Answer Type* (EAT) dan kata-kata kunci untuk dijadikan fitur pencarian. Proses ini dilakukan dengan pencocokan pola kata-kata pada pertanyaan dan pemanfaatan POS *tagger*.

2. Proses pencarian dan pengambilan solusi/jawaban

Pada proses ini file-file yang ada pada *corpus* diproses kedalam komponen *answer finder*. File-file tersebut dimuat ke dalam memori sebagai *case*, lalu dari kumpulan *case* tersebut akan dicari yang paling memiliki kecocokan fitur dengan fitur dari pertanyaan pengguna. *Case* yang cocok akan diambil bagian solusinya (jawaban) sebagai jawaban dari pertanyaan pengguna.

3.3.4. Deskripsi keluaran

Keluaran dari perangkat lunak sistem QA ini adalah jawaban dari pertanyaan yang dimasukkan oleh pengguna. Karena tipe pertanyaan yang digunakan adalah *real world question* maka setiap jawaban akan memiliki penjelasan yang mencukupi. Jawaban-jawaban dari pertanyaan didapatkan dari buku ISO (Informasi Spesialite Obat). Perangkat lunak akan langsung menampilkan satu jawaban dalam bentuk paragraf atau kalimat singkat.

3.3.5. Analisis Sumber Data Farmasi Obat-Obatan

Sumber data yang dipakai dalam penelitian ini adalah sebuah buku yang memuat informasi tentang obat-obatan yang beredar dan dipasarkan di Indonesia. Namanya adalah ISO Indonesia yang adalah kependekan dari Informasi Spesialite Obat Indonesia.

ISO Indonesia dibuat oleh Ikatan Sarjana Farmasi Indonesia (ISFI) yang bertujuan untuk memberikan informasi yang lengkap tentang obat-obatan yang beredar di Indonesia kepada masyarakat Indonesia pada umumnya juga kepada para tenaga kefarmasian yang membutuhkan informasi ini dalam melakukan pekerjaannya. Secara umum, buku ini memuat informasi kefarmasian obat-obatan diantaranya yaitu mengenai produk-produk obat yang beredar di Indonesia, singkatan latin pada resep dokter, indeks pabrik farmasi serta harga obat.

Adapun informasi tiap produk obat yang tertera pada buku ini antara lain adalah nama obat, zat berkhasiat yang dikandung, indikasi, kontra indikasi, efek samping obat, dosis pemakaian, nama pabrik pembuat, penggolongan obat, serta harga obat.

Seiring dengan perkembangan teknologi dengan kecepatan alir informasi dan kemudahannya, buku sebagai sumber informasi bukan menjadi pilihan pertama lagi. Oleh karena itu, diperlukan suatu aplikasi layaknya buku ISO yang bisa memberikan informasi tentang obat-obatan yang lebih mudah diakses dan bisa digunakan kapanpun, dimanapun.

3.3.6. Analisis Data Untuk *Database/Corpus*

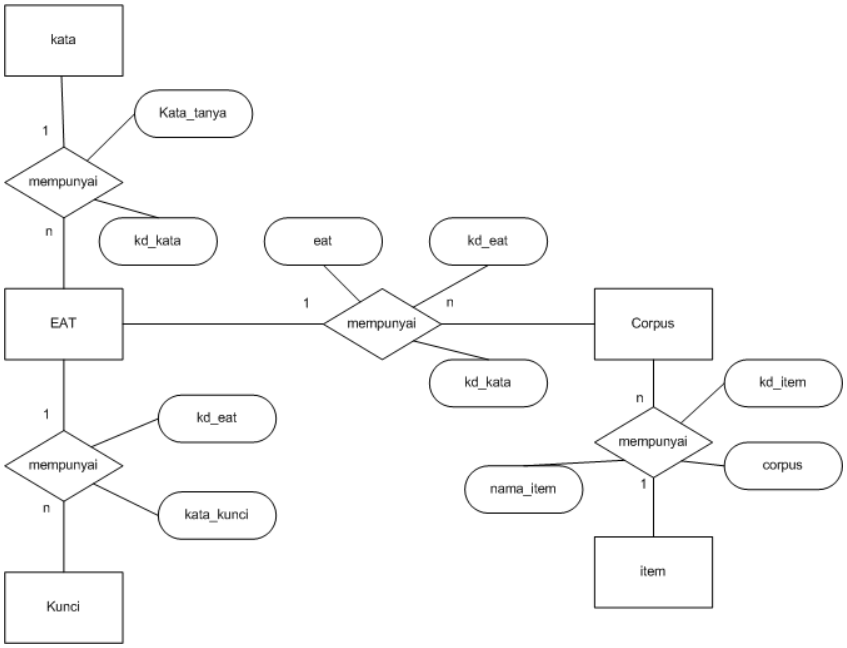
Sebagaimana telah dijelaskan sebelumnya, aplikasi yang akan dibangun ini hanya mampu memberikan jawaban atau informasi terkait obat-obatan sesuai dengan yang ada pada buku ISO Indonesia. Adapun detail informasi yang dapat diberikan oleh aplikasi ini mengenai obat-obatan adalah sebagai berikut:

Informasi mengenai produk obat.

a. Farmakoterapi
b. Nama obat
c. Penggolongan obat
d. Zat isi
e. Perusahaan pembuat
f. Indikasi
g. Efek Samping
h. Kontra Indikasi
i. Dosis
j. Kemasan
k. Harga

3.3.6.1. Analisis basis data

Analisis basis data adalah kegiatan menganalisis data yang akan diolah dan disimpan dalam database. Dalam analisis ini direpresentasikan dari mana data berasal dan atribut dari data tersebut.



Gambar 3.9 Entity relationship diagram database

3.4. Analisis Kebutuhan Non Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non fungsional ini akan dijelaskan mengenai analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.4.1. Analisis Kebutuhan Perangkat Lunak

Aplikasi *Question Answering System* domain kefarmasian obat-obatan ini memiliki kebutuhan perangkat lunak sebagai berikut:

1. Sistem operasi *mobile* Android.
2. Minimal versi Android 4.1 *Jelly bean*.

Kebutuhan perangkat lunak yang diperlukan cukup mudah untuk dipenuhi oleh banyak pengguna karena aplikasi yang hendak dirancang termasuk ringan sehingga bisa berjalan dengan baik pada *mobile* Android 4.1 keatas.

3.4.2. Analisis Kebutuhan Perangkat Keras

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Spesifikasi minimun perangkat keras yang sebaiknya tersedia agar dapat menjalankan aplikasi Question Answering system ini dengan baik adalah sebagai berikut:

a. Perangkat mobile dengan sistem operasi Android
b. Kecepatan prosesor minimal 600MHz
c. Kapasitas memory external minimal 256MB

3.4.3. Analisis Pengguna (*user*)

Analisis pengguna untuk aplikasi Question Answering System domain kefarmasian obat-obatan adalah sebagai berikut:

Tabel 3.5 Analisis Pengguna

Karakteristik Fisik Pengguna	
Umur	13 tahun keatas
Gender	Laki-laki dan perempuan
Handedness	Bertangan kiri ataupun bertangan kanan
Disabilities	Tidak buta dan tidak punya cacat fisik
Pengetahuan Dan Pengalaman Pengguna	

Computer literacy	Memiliki pengalaman dalam mengoperasikan perangkat *mobile*
System experience	Memiliki pengetahuan berinteraksi dengan *input mobile phone Android*
Application experience	Pernah dan atau tidak pernah menggunakan aplikasi yang sejenis
Education	Minimal SMP
Skills	Sudah bisa membaca
Task experience	Mengetahui cara menggunakan *mobile phone Android*
Native language or Culture	Bahasa Indonesia
Karakteristik Psikologis Pengguna	
Motivation	Ingin mengetahui lebih detail tentang obat-obatan
Patience	-
Stress level	-
Attitude	Positive

3.5. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan ditempatkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan. Pada bagian ini akan dibahas tentang spesifikasi kebutuhan perangkat lunak, use case diagram, activity diagram, class diagram, dan sequence diagram.

3.5.1. SKPL (Spesifikasi Kebutuhan Perangkat Lunak)

a. Kebutuhan Fungsional

Tabel 3.6 Kebutuhan Fungsional

SKPL-ID	Kebutuhan
SKPL-F-001	Aplikasi harus bisa menampilkan halaman utama aplikasi seperti input pertanyaan, tombol eksekusi aplikasi, dan output jawaban.
SKPL-F-002	Aplikasi harus bisa menerima pertanyaan *real world question* dari pengguna seputar obat-obatan (*input*)
SKPL-F-003	Sistem aplikasi harus bisa menganalisa pertanyaan yang diberikan oleh pengguna dan mendapatkan EAT, serta kata-kata kunci yang diperlukan (query, stemmed query, dll).
SKPL-F-004	Sistem dari aplikasi harus dapat mencari kasus yang paling mendekati pertanyaan dari pengguna dan mengambil jawabannya.
SKPL-F-005	Sistem aplikasi harus dapat mengelola dan menyimpan *database* / *corpus* tempat pasangan pertanyaan-jawaban berada agar dapat diakses oleh modul *answer finder*.
SKPL-F-006	Sistem aplikasi harus bisa memberikan / mengembalikan (*output*) jawaban yang paling sesuai dengan pertanyaan yang diajukan oleh pengguna.

b. Kebutuhan Non-Fungsional

Tabel 3.7 Kebutuhan Non-Fungsional

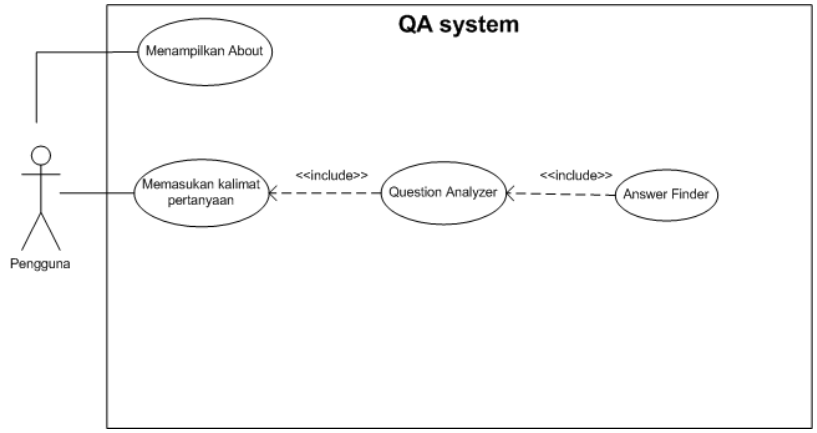
SKPL-ID	Kebutuhan
SKPL-NF-001	Aplikasi ini harus memiliki nilai kenyamanan dipakai yang tinggi bagi penggunanya. Aplikasi akan dibangun dengan antarmuka yang mudah dimengerti, dan tidak membingungkan pengguna.
SKPL-NF-002	Aplikasi ini harus bisa cepat memberikan jawaban terhadap pengguna
SKPL-NF-003	Aplikasi ini harus bisa memberikan jawaban yang singkat, sesuai kebutuhan dan memiliki tingkat akurasi pengembalian jawaban tepat yang cukup tinggi.
SKPL-NF-004	Bahasa yang digunakan adalah bahasa Indonesia yang baku, baik dan benar.

3.5.2. Use Case Diagram

Pada sub bab berikut ini dijelaskan bagaimana *Use Case* yang terjadi dalam pembangunan aplikasi ini yang dimodelkan dalam sekumpulan *Use Case* dan *Actor* dan bagaimana hubungan–hubungannya yang akan dipetakan dalam diagram *Use Case*, dilengkapi dengan skenario untuk menjelaskan dari gambaran *Use Case* yang ada dengan menggambarkan skenario per *Use Case*.

3.5.2.1. Use Case Diagram

Use case diagram merupakan salah satu model untuk menggambarkan fungsionalitas atau perilaku dari sistem.



Gambar 3.10 Use Case Diagram

3.5.2.2. Use Case Scenario

Use Case Scenario merupakan suatu penggambaran langkah-langkah aksi aktor terhadap sistem.

a. *Use case scenario* untuk menampilkan halaman *About*

Tabel 3.8 use case scenario halaman about

Identifikasi	
Nomor	UC-01
Nama	Menampilkan halaman About
Tujuan	Untuk menampilkan halaman About
Deskripsi	Proses untuk menampilkan halaman About
Aktor	Pengguna (*user*)
Skenario Utama	
Aksi Reaktor	**Reaksi Sistem**
1. Memilih menu About	
	2. Load halaman About
	3. Menampilkan halaman About
4. Menerima tampilan halaman About	
Kondisi	Aplikasi QA System menampilkan halaman About

akhir	

b. *Use case scenario* memasukan kalimat pertanyaan

Tabel 3.9 *Use Case Scenario* memasukan kalimat pertanyaan

Identifikasi	
Nomor	UC-02
Nama	Memasukan kalimat pertanyaan
Tujuan	Sebagai tempat input pertanyaan dari pengguna
Deskripsi	Sistem menerima input kalimat pertanyaan dari pengguna untuk kemudian akan diproses lebih lanjut oleh sistem QA.
Aktor	Pengguna (*user*)

Skenario utama	
Aksi Reaktor	**Reaksi Sistem**
1. Mengetik pertanyaan di *textbox* yang telah disediakan	
2. Menekan tombol *enter*	
	3. Memasukan kalimat pertanyaan kedalam sistem QA.
Alternatif skenario	
	1a. Jika *user* tidak mengetik pertanyaan dan menekan tombol *enter*, maka akan ada *pop-up* peringatan agar pengguna untuk mengetikan kalimat pertanyaannya.
Kondisi akhir	Input kalimat pertanyaan dari pengguna diproses lebih lanjut oleh modul *question analyzer*

c. *Use Case scenario* Question Analyzer

Tabel 3.10 Use Case Scenario question analyzer

Identifikasi	
Nomor	UC-03
Nama	Question Analyzer
Tujuan	Modul yang berfungsi untuk memproses pertanyaan dari pengguna dan menghasilkan EAT (Expected Answer Type) dan kata-kata kunci.
Deskripsi	Proses untuk mendapatkan EAT dan kata-kata kunci yang terdapat pada kalimat pertanyaan yang dimasukkan oleh pengguna aplikasi menggunakan metode *pattern matching*
Aktor	Sistem
Skenario Utama	
Aksi Reaktor	**Reaksi Sistem**
	1. Menerima dan memproses kalimat pertanyaan dari pengguna
	2. Mengambil EAT dan kata-kata kunci yang terdapat pada kalimat pertanyaan
Alternatif Skenario	
	2a. Jika masukan kalimat pertanyaan tidak ditemukan kecocokan pola dengan yang telah dibuat di sistem, maka modul ini tidak akan menghasilkan/memberikan EAT yang diperlukan oleh modul *answer finder* untuk memberikan jawaban.
Kondisi akhir	Menghasilkan EAT dan kata-kata kunci

d. Use case scenario Answer Finder

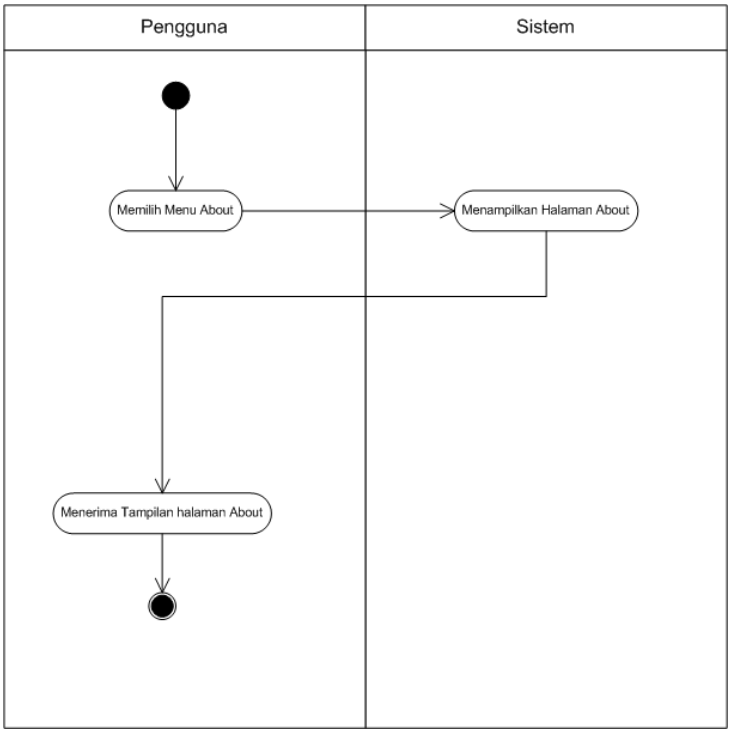
Tabel 3.11 use case scenario answer finder

Identifikasi	
Nomor	UC-04
Nama	Answer Finder
Tujuan	Modul yang bertugas untuk mencari jawaban yang sesuai dengan pertanyaan dari pengguna (*user*)
Deskripsi	Proses untuk mengambil jawaban yang sesuai dari *database/corpus* dengan pencocokan EAT dan kata kunci hasil dari question analyzer dengan yang sudah dibuat di *database/corpus*
Aktor	sistem
Skenario utama	
Aksi Reaktor	**Reaksi Sistem**
	1. Menerima EAT dan kata-kata kunci dari modul *question analyzer*
	2. Mencocokan EAT dan kata kunci dari modul *question analyzer* dengan yang sudah terdapat di *database*
	3. Jika ada pola yang cocok, maka jawaban yang terdapat pada database akan diambil dan dikembalikan kepada pengguna sebagai jawaban.
Alternatif Skenario	
	3a. Jika pada masukan kalimat pertanyaan terdapat nama obat (nama item) yang belum ada di dalam *database* sistem, maka sistem akan mengembalikan *pop-up message* bahwa data tidak ditemukan
Kondisi akhir	Menghasilkan kandidat jawaban dan siap untuk dikembalikan kepada pengguna sebagai jawaban yang dianggap paling sesuai

3.5.2.3. Activity Diagram

Activity diagram menunjukkan aktivitas oleh manusia atau objek dalam cara yang mirip dengan *Flow Chart*.

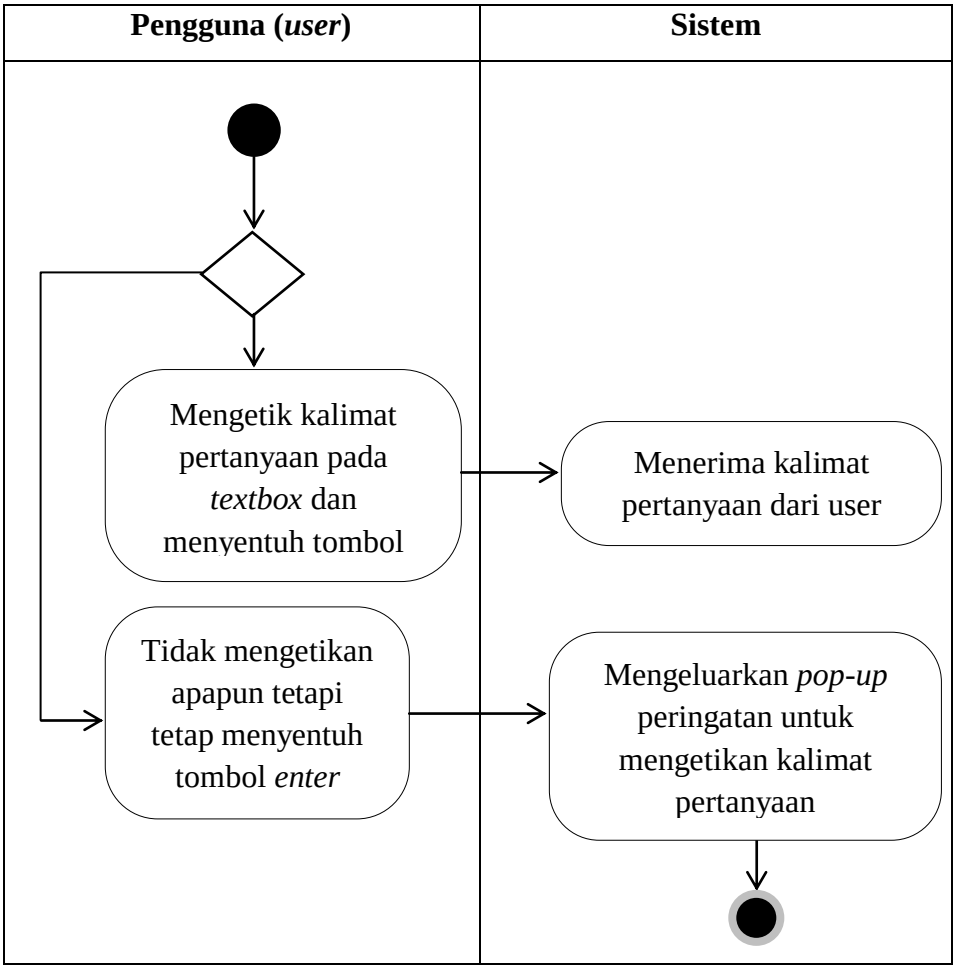
1. Activity diagram untuk menampilkan halaman About



Gambar 3.11 Activity diagram halaman about

2. Activity diagram memasukan kalimat pertanyaan.

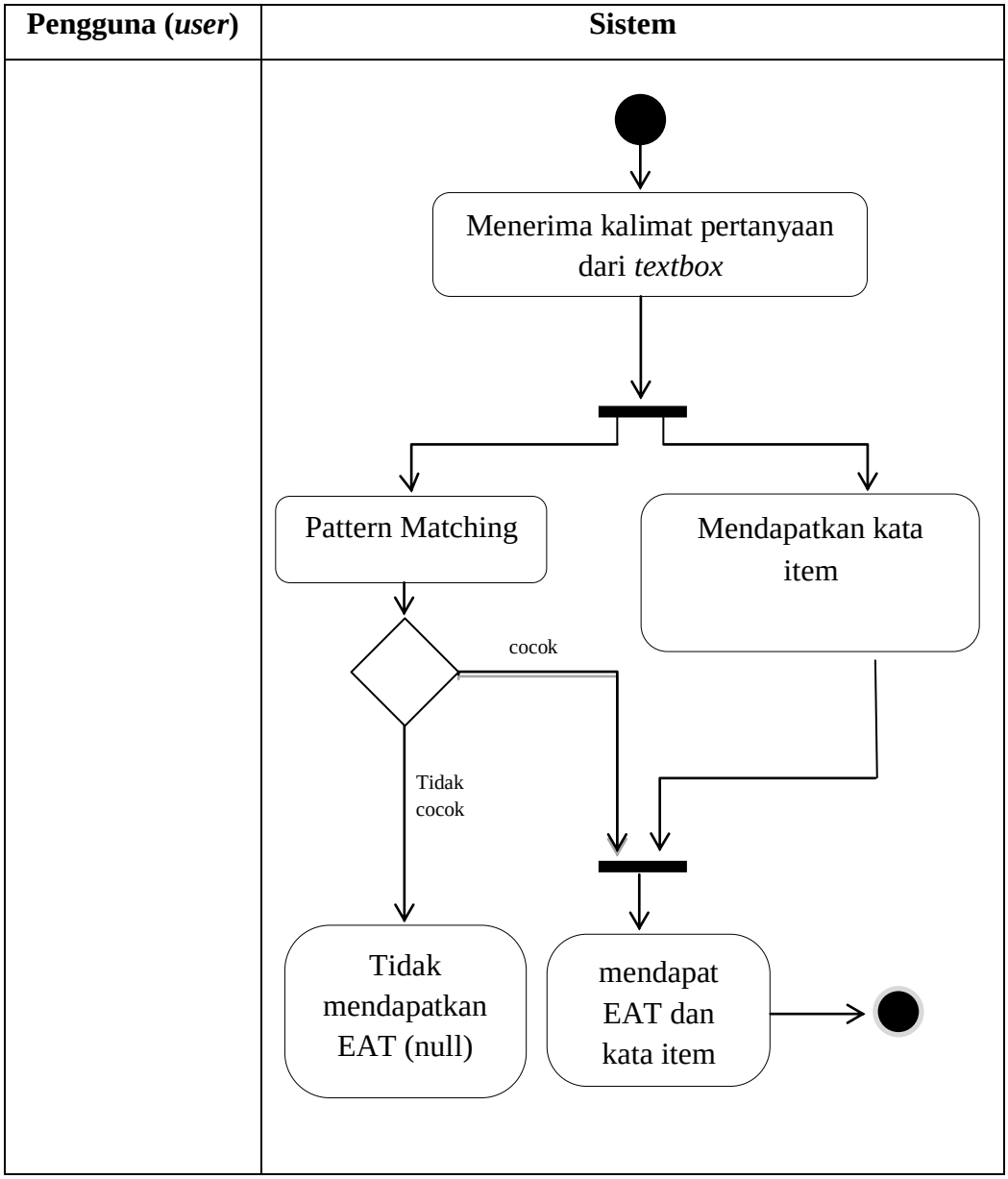
Berikut ini adalah diagram yang menunjukan aksi pada aktivitas menampilkan *textbox* untuk memasukan pertanyaan yang dapat dilihat pada gambar 3.12 di halaman selanjutnya.



Gambar 3.12 Activity Diagram menampilkan *textbox* untuk memasukan pertanyaan

3. Activity Diagram *Question Analyzer*

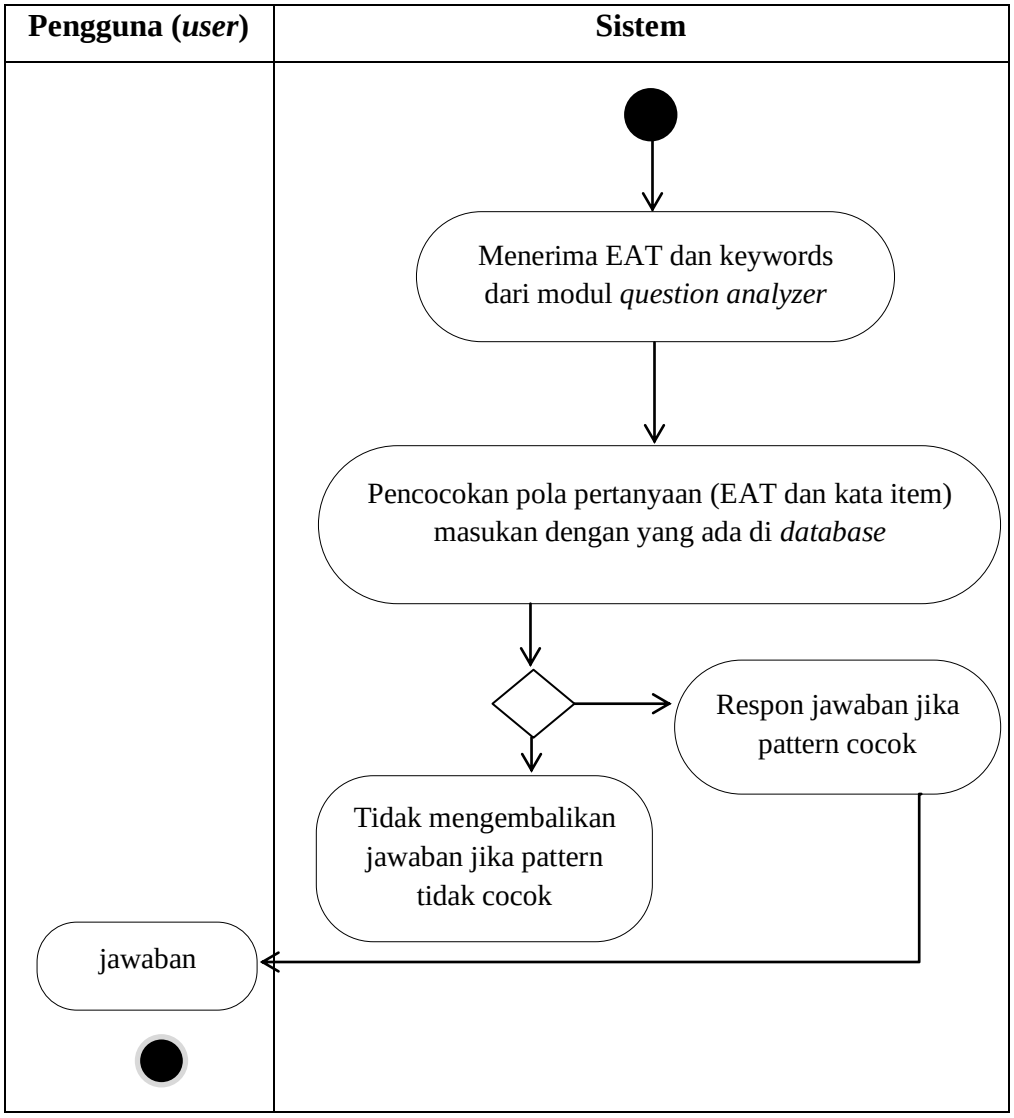
Menampilkan *Activity Diagram* yang menunjukan pemrosesan sistem menerima kalimat pertanyaan untuk mendapatkan EAT dan kata-kata kunci dapat dilihat pada gambar 3.13 di halaman selanjutnya



Gambar 3.13 Activity Diagram *Question Analyzer*

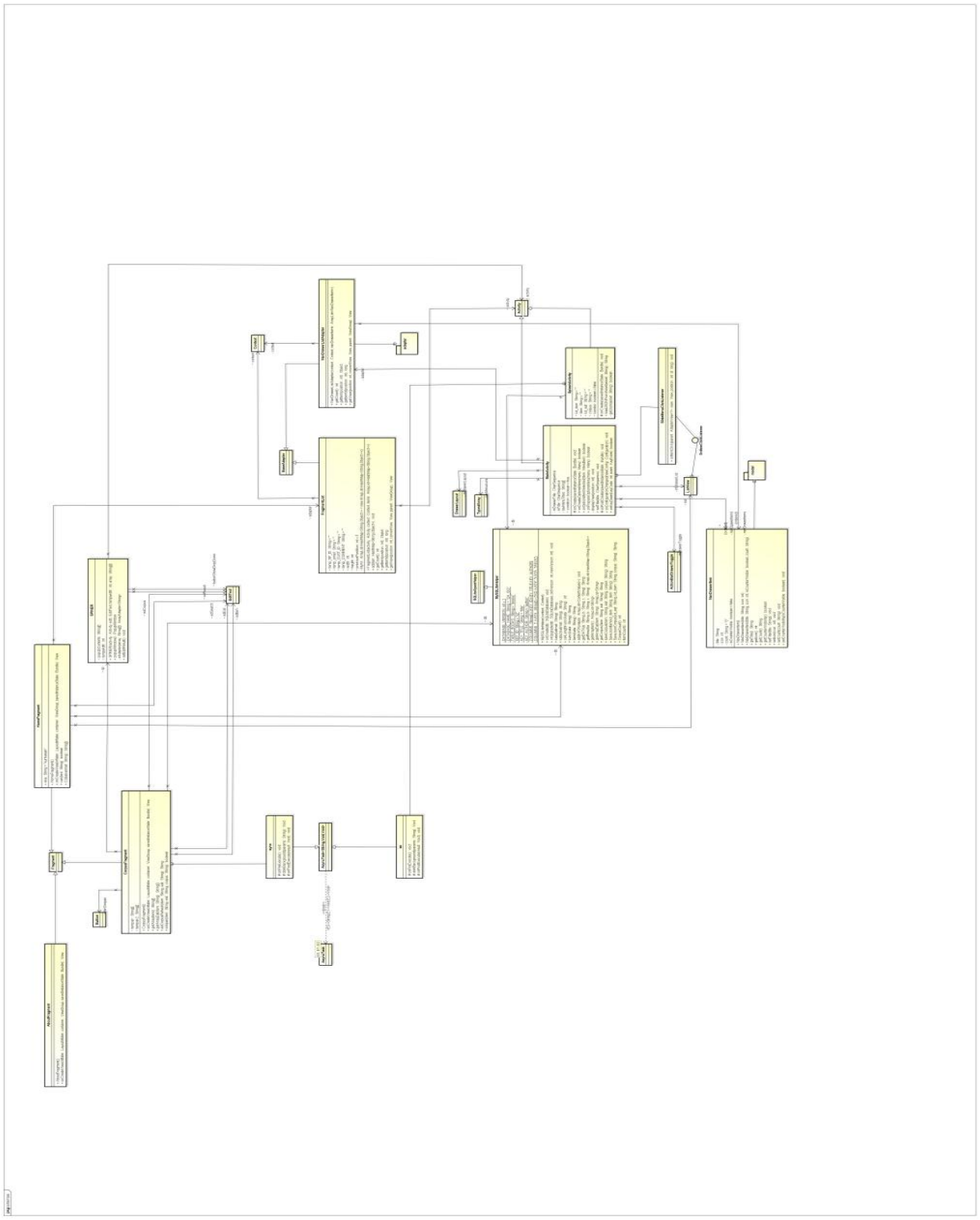
4. Activity Diagram Answer Finder

Menampilkan *activity diagram* yang menunjukan aktifitas sistem dalam mencari jawaban di modul *answer finder* yang dapat dilihat pada gambar 3.14 berikut ini.



Gambar 3.14 Activity Diagram Answer Finder

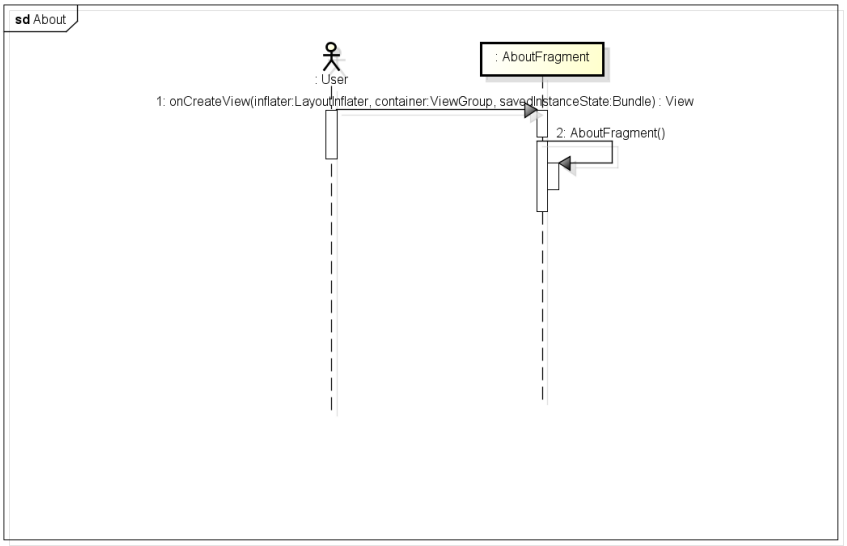
3.5.3. Class Diagram



Gambar 3.15 Class Diagram ISO farmasi

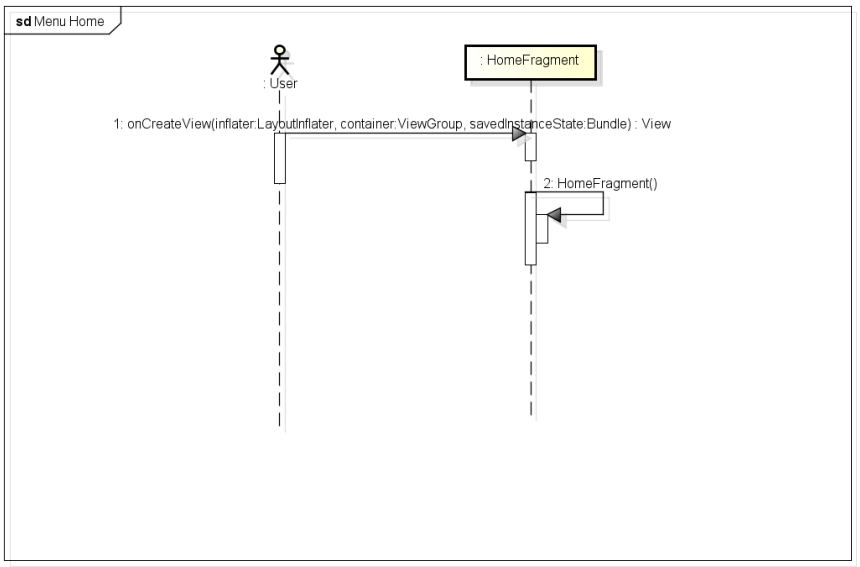
3.5.4. Sequence Diagram

3.5.4.1. Sequence Diagram About



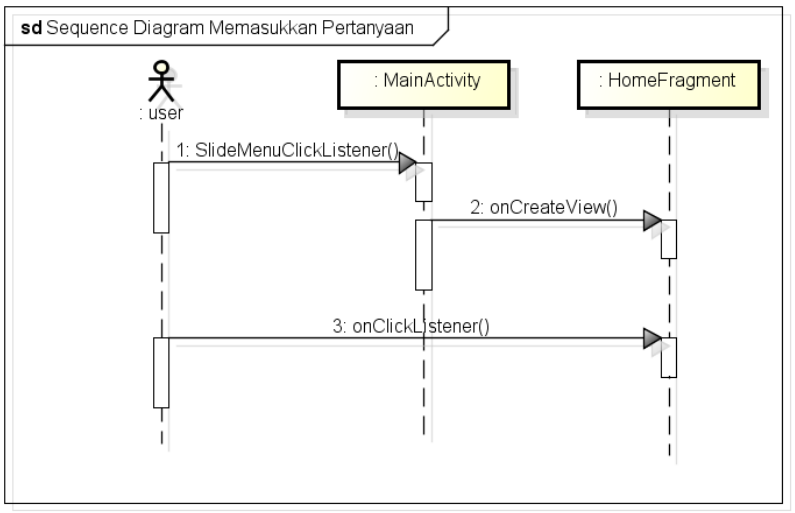
Gambar 3.16 *Sequence Diagram* Menu About

3.5.4.2. Sequence Diagram Home



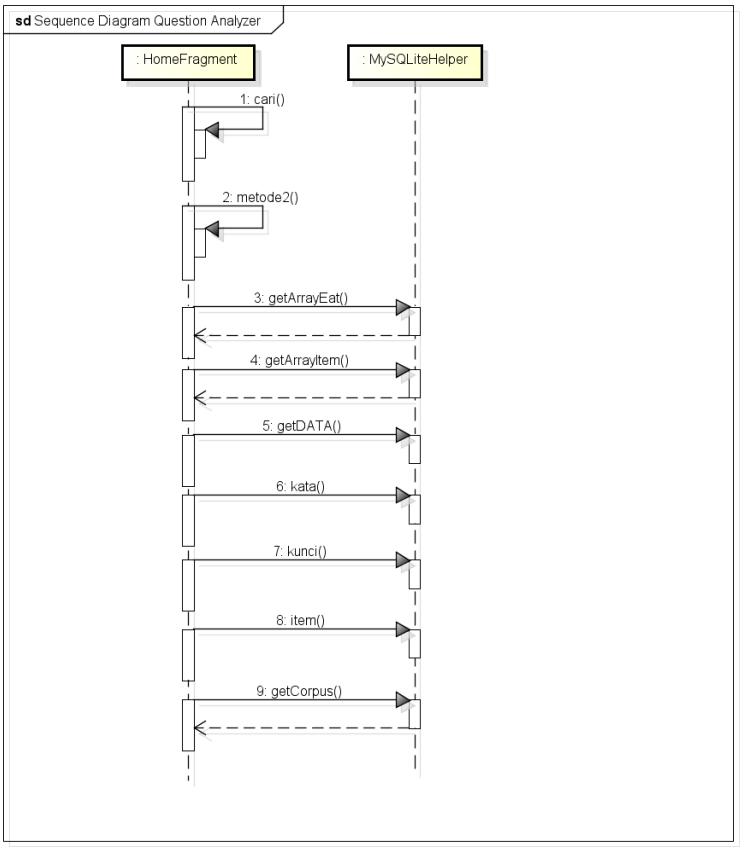
Gambar 3.17 *Sequence Diagram* Menu Home

3.5.4.3. Sequence diagram memasukan pertanyaan



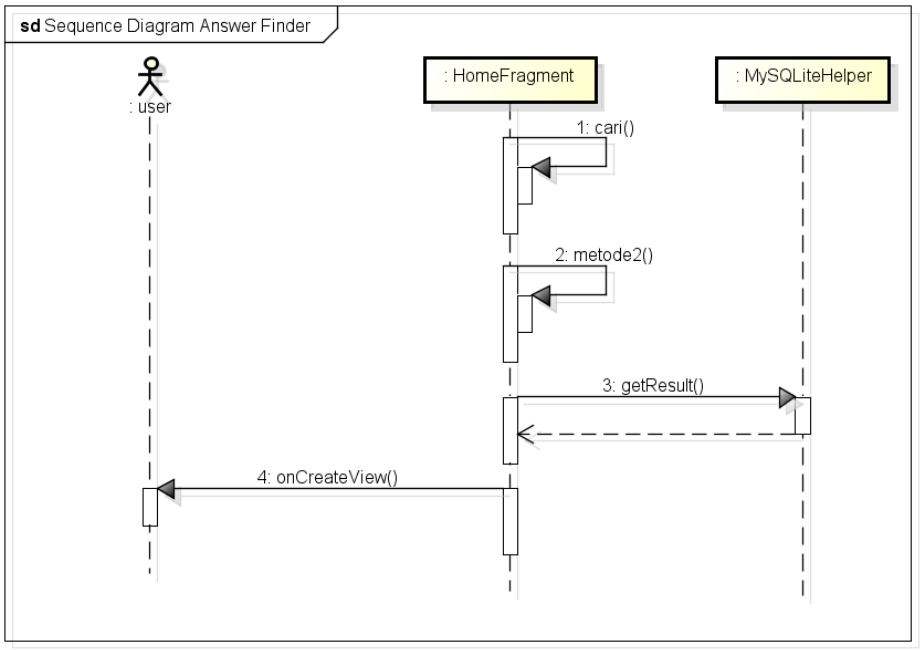
Gambar 3.18 *Sequence Diagram* Memasukan Pertanyaan

3.5.4.4. Sequence Diagram Question Analyzer



Gambar 3.19 *Sequence Diagram Question Analyzer*

3.5.4.5. Sequence Diagram Answer Finder



Gambar 3.20 *Sequence Diagram Answer Finder*

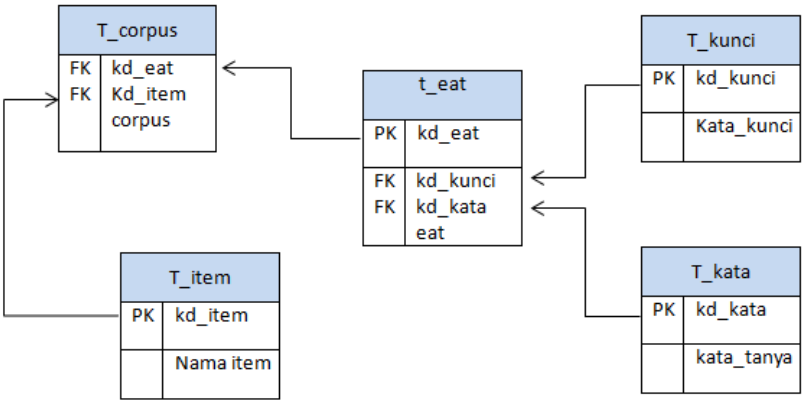
3.6. Perancangan Sistem

3.6.1 Perancangan Database

Perancangan *database* yaitu perancangan yang bisa menciptakan atau merancang kumpulan data yang tersimpan dalam aplikasi dalam rangka membantu proses yang terjadi dalam aplikasi sehingaa mendapatkan hasil yang diinginkan.

3.6.1.1 Diagram Skema

Pada tahap ini akan digambarkan diagram struktur database yang digunakan. Struktur relasi database ini dapat dilihat pada gambar 3.21 berikut ini.



Gambar 3.21 Skema relasi

3.6.1.2 Struktur Tabel

Tabel adalah sekumpulan data informasi spesifik tentang subjek tertentu yang disusun dalam bentuk kolom dan baris. Tabel merupakan komponen utama dan pertama dari sebuah database. Struktur tabel untuk setiap tabel yang terdapat dalam database sistem ini adalah:

1. Tabel item

   Tabel item digunakan untuk menyimpan data nama obat-obatan yang didapat dari buku ISO. Strukur tabel item bisa dilihat pada tabel 3.12 berikut ini.

Tabel 3.12 Tabel item

Nama field	Tipe	Size	Keterangan
kd_item	integer	4	Primary key
Nama_item	varchar	20	

2. Tabel kata

   Tabel kata digunakan sebagai tempat menyimpan kata tanya sebagai salah satu pembentuk pola yang akan dicocokan dengan pola kalimat pertanyaan dari pengguna. Berikut struktur tabel kata yang bisa dilihat pada tabel 3.13 dibawah ini.

Tabel 3.13 tabel kata

Nama field	Tipe	Size	Keterangan
kd_kata	integer	4	Primary key
Kata_tanya	varchar	20	

3. Tabel kunci

Tabel kunci merupakan tabel yang menyimpan data kata-kata kunci sebagai salah satu pembentuk pola dalam metode *matching pattern*, juga sebagai penentu EAT (*Expected Answer Type*). Struktur tabel kunci dapat dilihat pada tabel 3.14 di halaman berikutnya.

Tabel 3.14 Tabel kunci

Nama field	Tipe	Size	Keterangan
kd_kunci	integer	4	Primary key
Kata_kunci	varchar	20	

4. Tabel EAT

Tabel EAT juga merupakan salah satu elemen pembentuk pola yang sudah dibentuk untuk menjalankan metode *matching pattern.* EAT didapat dari gabungan kata kunci dengan kata tanya. Strukur tabel EAT dapat dilihat pada tabel 3.15 berikut ini.

Tabel 3.15 tabel EAT

Nama field	Tipe	Size	Keterangan
kd_eat	integer	4	Primary key
kd_kunci	Integer	4	Foreign key
kd_kata	integer	4	Foreign key
eat	varchar	20	

5. Tabel Corpus

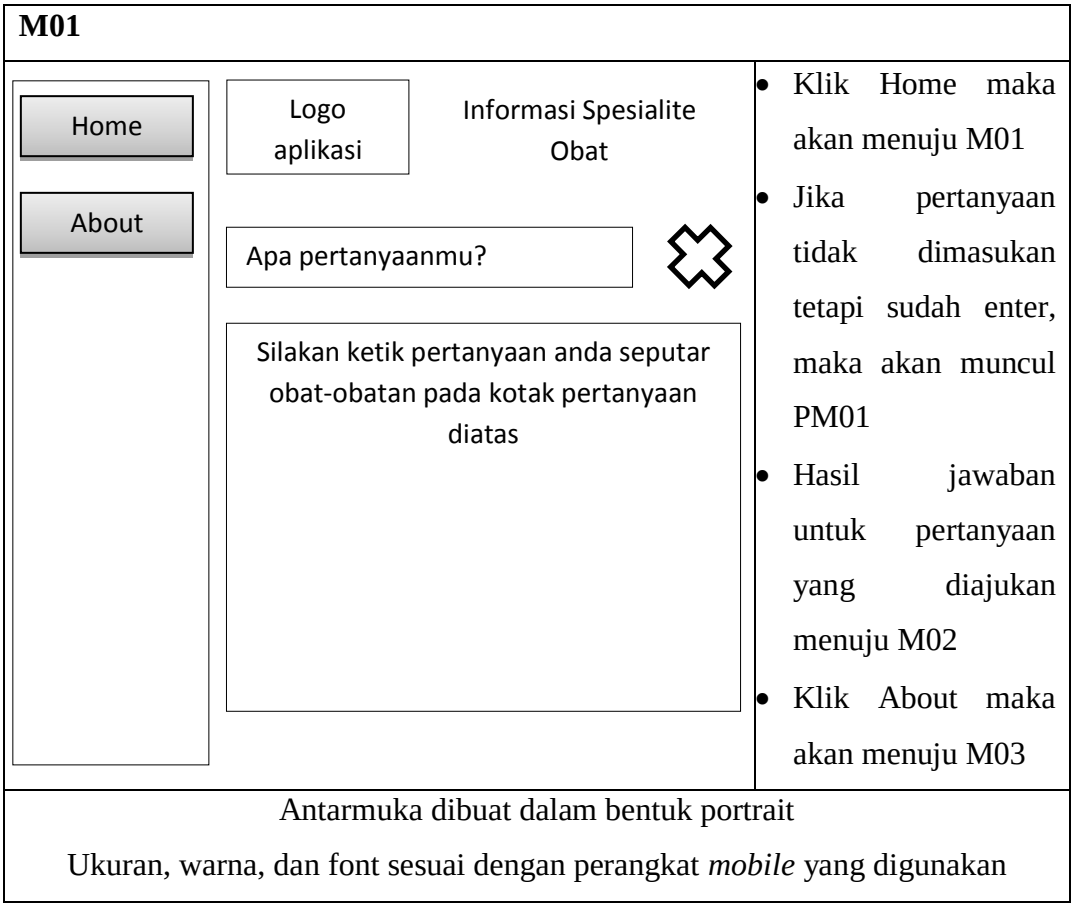
Tabel corpus merupakan tabel tempat data jawaban disimpan. Strukur tabel corpus bisa dilihat pada tabel 3.16 berikut ini.

Tabel 3.16 tabel corpus

Nama field	Tipe	Size	Keterangan
kd_eat	integer	4	Foreign key
kd_item	varchar	20	Foreign key
corpus	varchar	200	

3.6.2 Perancangan Antarmuka

3.6.2.1 Perancangan Antarmuka Halaman Utama



Gambar 3.22 Perancangan Antarmuka Halaman Utama

3.6.2.2 Perancangan Antarmuka Hasil Jawaban

M02



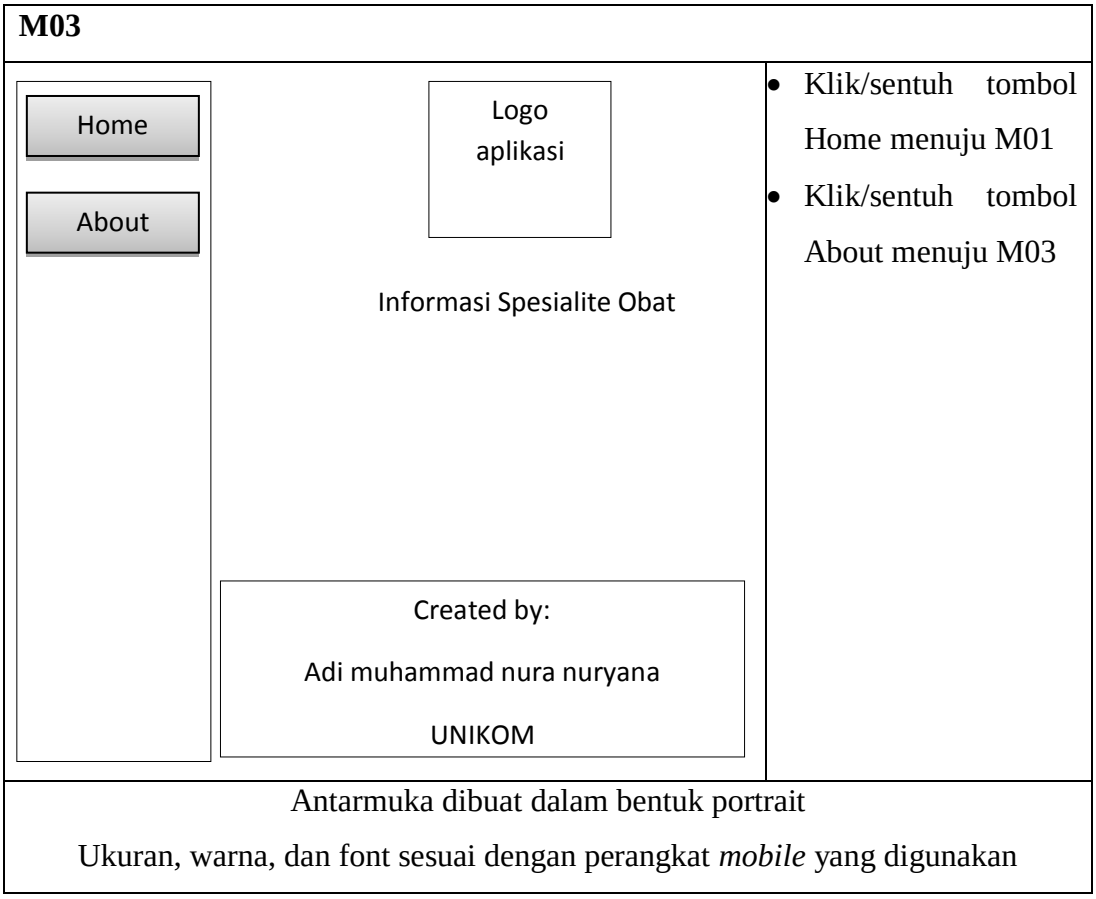
- Klik/sentuh tombol Copy untuk mengcopy jawaban dan akan muncul PM02
- Klik atau sentuh tanda silang untuk menghapus pertanyaan dan hasil jawaban sebelumnya untuk menanyakan pertanyaan berikutnya.
- Klik/sentuh tombol About akan menuju M03.

Antarmuka dibuat dalam bentuk portrait

Ukuran, warna, dan font sesuai dengan perangkat *mobile* yang digunakan

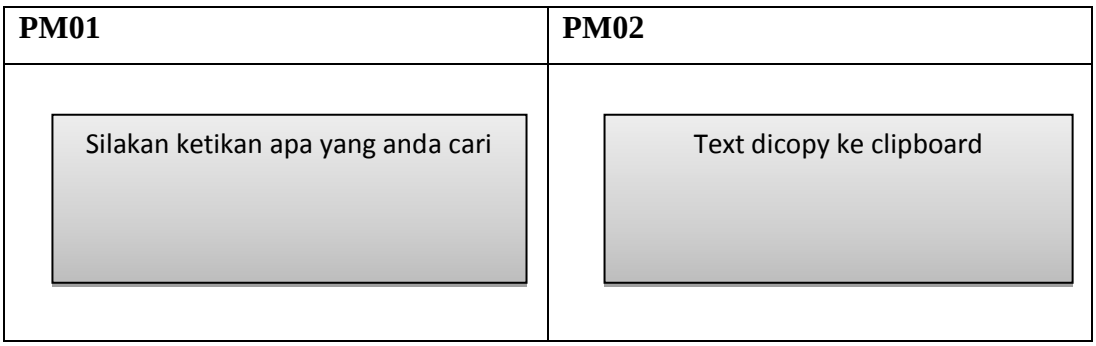
Gambar 3.23 Perancangan Antarmuka Hasil Jawaban

3.6.2.3 Perancangan Antarmuka halaman About



Gambar 3.24 Perancangan Antarmuka Halaman About

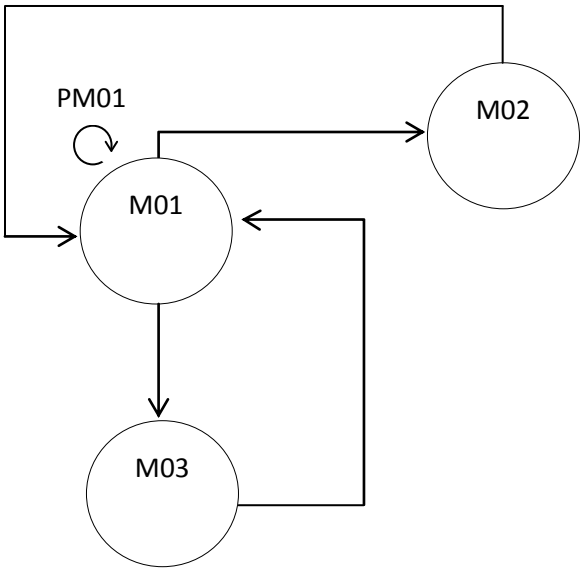
3.6.2.4 Perancangan Pesan aplikasi



Gambar 3.25 Perancangan Pesan Aplikasi

3.6.3 Jaringan Semantik

Setelah melakukan perancangan antarmuka maka dilakukan perancangan terhadap hubungan antar *form*. Untuk menggambarkan hubungannya digambarkan dalam sebuah jaringan semantik.

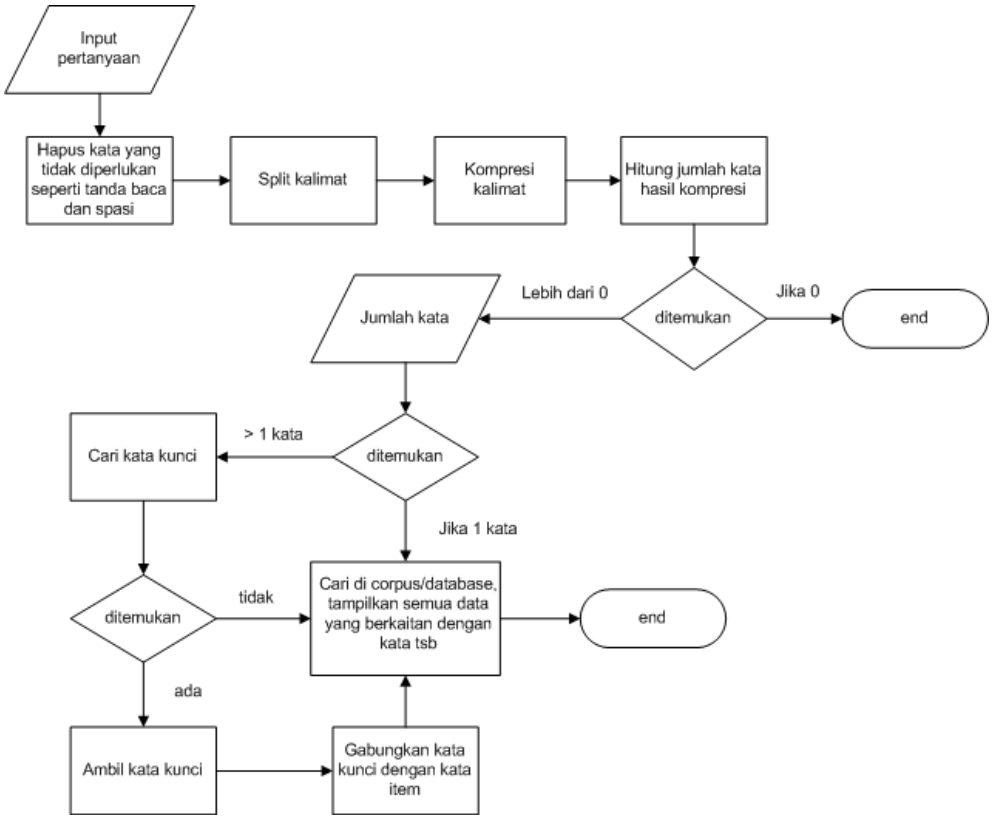


Gambar 3.26 Jaringan Semantik Aplikasi

3.6.4 Perancangan Method

Perancangan method mentransformasi elemen-elemen objek dari arsitektur program ke dalam suatu deskripsi method dari komponen-komponen perangkat lunak. Adapun perancangan method yang ada pada aplikasi ini adalah sebagai berikut:

Perancangan method *question analyzer*



Gambar 3.27 Method *question analyzer*