BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Penelitian yang sudah pernah membuat sistem ini berhasil menciptakan pembangkitan pertanyaan *non-factoid* secara otomatis dengan menggunakan tiga jenis kategori, yaitu definisi, alasan, dan metode. Penggunaan kategori tersebut hanya dapat membangkitkan tiga jenis kata tanya, yaitu *apa yang dimaksud, mengapa,* dan *bagaimana*. Sedangkan kata tanya umum seperti siapa, apa, mana, dan kapan tidak dapat dibangkitkan karena kata tanya tersebut menciptakan kalimat tanya *factoid*.

Untuk membangkitkan kata tanya siapa, apa, mana dan kapan yang belum teratasi pada penelitian sebelumnya, dibutuhkan fungsi sintaksis sebagai target yang ditanyakan. Fungsi-fungsi sintaksis yang dibutuhkan adalah subjek, predikat, objek, dan keterangan. Sebagai contoh, kata tanya siapa membutuhkan target tanya berupa fungsi sintaksis subjek atau objek. Oleh karena itu dibutuhkan analisis sintaksis yang berfungsi untuk mencari fungsi-fungsi sintaksis pada suatu kalimat yang kemudian akan digunakan sebagai acuan *template* pertanyaan.

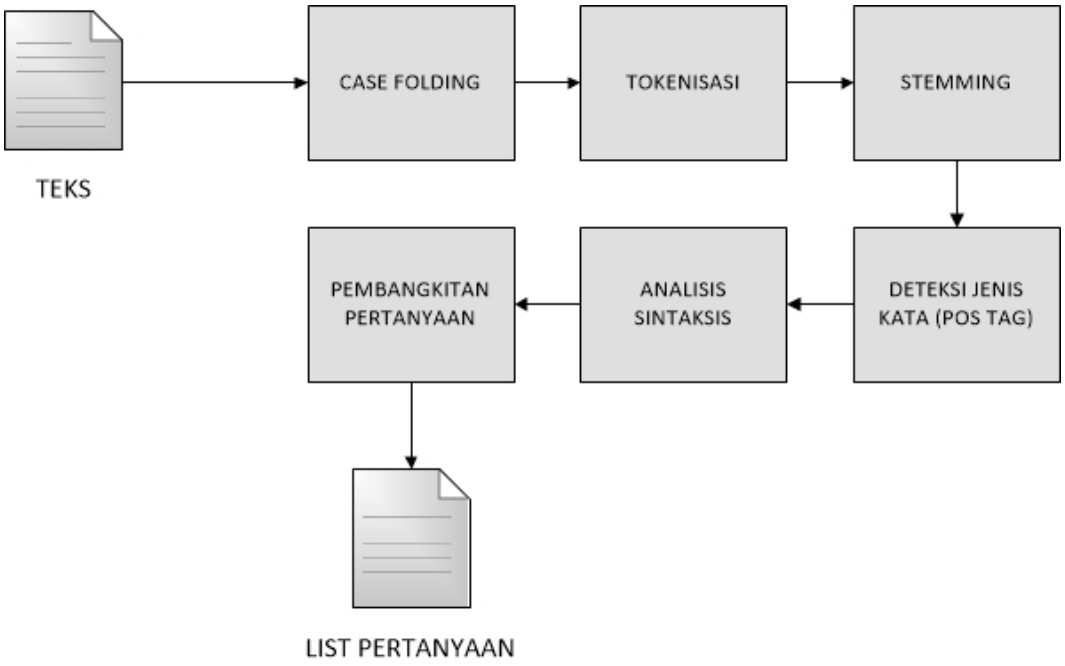
Dari hasil pertanyaan dan jawaban yang dibangkitkan oleh sistem, diperlukan perhitungan akurasi relevansi pertanyaan dan jawaban dengan teks masukan untuk menilai tingkat akurasi dari sistem pembangkit pertanyaan otomatis.

3.2 Analisis Solusi

Berdasarkan masalah yang telah dijelaskan, penelitian ini akan menggunakan metode yang mirip tapi ditambah dengan suatu teknik analisis, yaitu analisis sintaksis, agar memungkinkan sistem pembangkitan pertanyaan dapat membangkitkan kalimat pertanyaan *factoid* dan bentuk kata tanya siapa, apa, kapan, mana, mengapa, dan bagaimana.

Alur umum sistem ini adalah pertama-tama *user* mengunggah file teks soal cerita pada sistem. Teks tersebut akan diproses oleh sistem menjadi kumpulan

token (kata atau angka) menggunakan tokenisasi. Kumpulan token tersebut yang akan menjadi data awal pembangkitan pertanyaan yang selanjutnya akan dideteksi kata dasar, imbuhan, jenis kata, dan analisis sintaksisnya. Sehingga pada akhirnya akan menciptakan kumpulan pertanyaan yang sesuai dengan soal cerita yang diinput oleh *user*.



Gambar 3.1 Tahapan Utama Sistem

3.2.1. Analisis Masukan

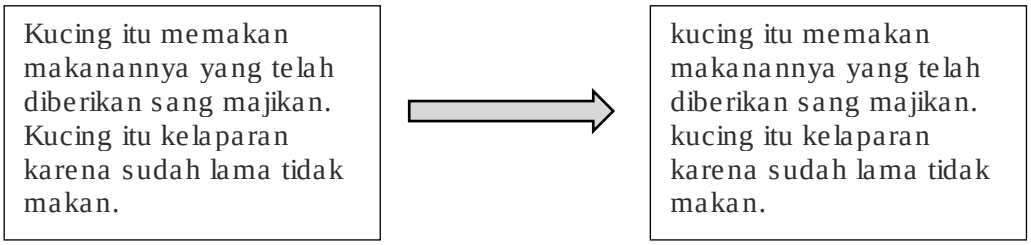
Input atau masukan dalam sistem ini berupa file yang memiliki *extention* .txt dan memiliki ukuran maksimal 1 Mb. Masukan yang diharapkan adalah teks yang memiliki struktur kalimat baku karena sistem ini tidak dapat membaca kalimat-kalimat tidak baku dan mengandung simbol-simbol. Sistem ini hanya dapat membaca huruf dan angka saja.

3.2.2. *Pre-Processing*

Tahap awal yang dilakukan dalam sistem setelah mendapatkan teks soal cerita adalah *pre-processing*. *Pre-processing* merupakan tahapan yang sangat penting dalam sistem ini karena menentukan akurasi hasil pertanyaan yang dibangkitkan nanti. *Pre-processing* dalam sistem ini terdiri dari empat tahap, yaitu *case folding*, tokenisasi, *stemming*, dan deteksi jenis kata.

3.2.2.1. *Case Folding*

Case folding dilakukan untuk merubah huruf kapital menjadi huruf non-kapital. *Case folding* berfungsi agar mengurangi kesalahan sistem dalam membaca suatu kata saat melakukan pengecekan ke basis data kata dasar, karena kata dasar dalam basis data semuanya menggunakan huruf non-kapital Berikut perubahan teks input sebelum dan sesudah dilakukan *case folding*.



Gambar 3.2 *Case Folding*

3.2.2.2. Tokenisasi Kalimat

Teks yang dimasukkan ke dalam sistem berupa kumpulan kalimat, sehingga diperlukan tokenisasi kalimat untuk memisahkan kumpulan kalimat tersebut. Tokenisasi ini dilakukan karena pertanyaan yang akan diciptakan berasal dari per kalimat. Berikut penggambaran tokenisasi kalimat.

kucing itu memakan makanannya yang telah diberikan sang majikan. kucing itu kelaparan karena sudah lama tidak makan.



No	Hasil Tokenisasi Kalimat
1	kucing itu memakan makanannya yang telah diberikan sang majikan.
2	kucing itu kelaparan karena sudah lama tidak makan.

Gambar 3.3 Tokenisasi Kalimat

Pada penggambaran di atas, teks masukan yang memiliki dua kalimat dipisahkan menjadi dua *string* atau data yang berbeda. Pemisahan kalimat ini berdasarkan tanda baca seperti titik, tanda tanya, dan tanda seru.

3.2.2.3. Tokenisasi Kata

Setelah kalimat dipisahkan dalam tokenisasi kalimat, tokenisasi kata dilakukan untuk mendapatkan token dalam bentuk per kata. Berikut penggambaran tokenisasi kata.

kucing itu memakan makanannya yang telah diberikan sang majikan.



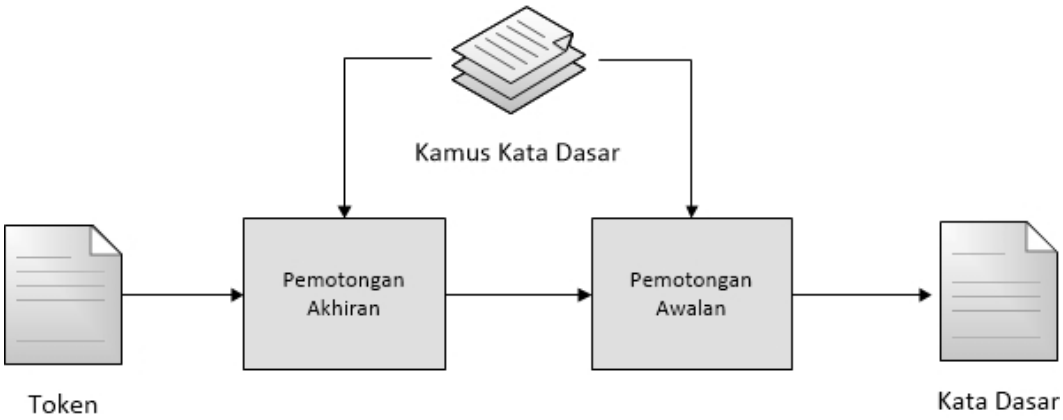
No	Hasil Tokenisasi Kata
1	kucing
2	itu
3	memakan
4	makanannya
5	yang
6	telah
7	diberikan
8	sang
9	majikan

Gambar 3.4 Tokenisasi Kata

Hasil dari tokenisasi kata adalah mendapatkan token-token berupa kata atau angka yang berasal dari satu kalimat. Cara mendapatkan token-token tersebut adalah dengan memisahkan per kata berdasarkan spasi kosong.

3.2.2.4. *Stemming*

Stemming dilakukan untuk mencari kata dasar dari token yang telah didapatkan dari tahap sebelumnya. Tujuan digunakan *stemming* dalam sistem ini untuk menentukan jenis kata dan menangani perubahan predikat aktif menjadi predikat pasif.



Gambar 3.5 Proses *Stemming*

Proses *stemming* dibantu dengan kamus kata dasar yang didapatkan dari Kateglo, sehingga setelah dilakukan proses pemotongan akhiran dan awalan akan dilakukan pengecekan terlebih dahulu ke dalam kamus kata dasar yang sudah ada sebelum ditentukan telah ditemukan kata dasarnya. Proses *stemming* tersebut melakukan pendekatan algoritma Nazief dan Adriani berdasarkan pertimbangan akurasi yang telah dijelaskan pada sub-bab 2.2.3. Berikut contoh dari input dan output pada tahapan *stemming*.

Tabel 3.1 Contoh *Stemming*

Token	**Kata Dasar**	**Imbuhan**	
		Awalan	**Akhiran**
kucing	kucing	-	-
itu	itu	-	-
memakan	makan	me	-
makanannya	makan	-	an - nya

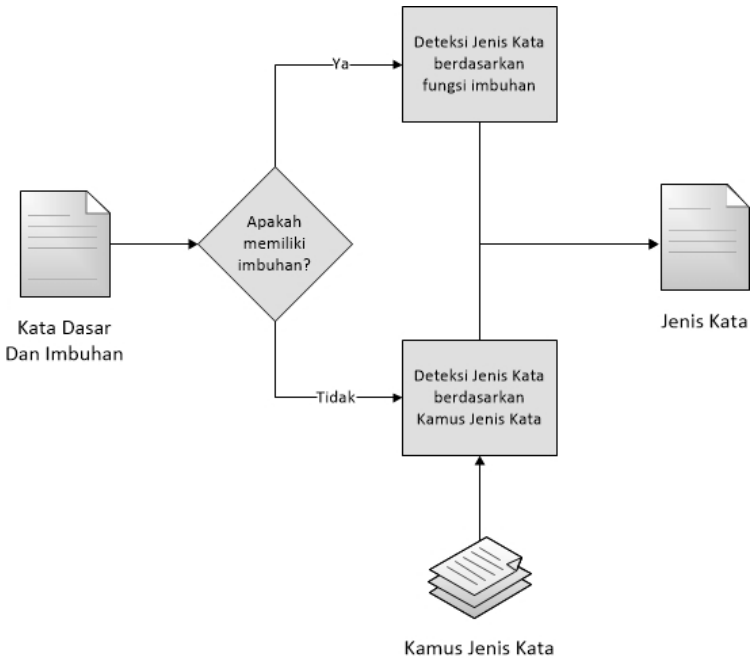
Token	Kata Dasar	Imbuhan	
		Awalan	Akhiran
yang	yang	-	-
telah	telah	-	-
diberikan	beri	di	kan
sang	sang	-	-
majikan	majikan	-	-

Output dari tahapan ini adalah token dengan kata dasar dan imbuhannya bila memang ditemukan.

3.2.2.5. Deteksi Jenis Kata (POS Tag)

Jenis kata dalam sistem pembangkitan pertanyaan ini sangat diperlukan karena digunakan untuk membantu dalam analisis sintaksis. Dengan bantuan jenis kata, fungsi-fungsi sintaksis seperti subjek, predikat, objek, dan keterangan dapat ditemukan. Dengan ditemukannya fungsi-fungsi sintaksis tersebut, berdasarkan penjelasan pada subbab 2.1.4, pembentukan kalimat dapat dilakukan.

Pendeteksian jenis kata dalam sistem ini memanfaatkan kamus kata dasar kateglo yang telah memiliki jenis kata untuk setiap kata dasar di dalam kamusnya. Jika token tersebut ditemukan imbuhan, maka digunakan fungsi imbuhan dalam tata bahasa Indonesia berdasarkan tabel 2.2, 2.3, dan 2.4. Berikut penggambaran proses dalam pendeteksian jenis kata.



Gambar 3.6 Proses Deteksi Jenis Kata

Berikut contoh kata atau token yang akan dideteksi jenis katanya pada proses deteksi jenis kata yang digambarkan pada Gambar 3.4 dan bagaimana hasil keluaran dari proses tersebut.

Tabel 3.2 Hasil Deteksi Jenis Kata

Kata (Token)	Kata Dasar	Jenis Kata Dasar	Imbuhan		Jenis Kata (Token)
			Awalan	Akhiran	
kucing	kucing	Kata Benda	-	-	Kata Benda
itu	itu	Kata Tunjuk	-	-	Kata Tunjuk
memakan	makan	Kata Kerja	me	-	Kata Kerja
makanannya	makan	Kata Kerja	-	an - nya	Kata Benda
yang	yang	Konjungsi	-	-	Konjungsi
telah	telah	Keterangan	-	-	Keterangan
diberikan	beri	Kata Kerja	di	kan	Kata Kerja
sang	sang	Kata Partikel	-	-	Kata Partikel
majikan	majikan	Kata Benda	-	-	Kata Benda

Pada Tabel 3.2 data token, kata dasar, dan imbuhan yang sudah didapatkan dalam tahapan tokenisasi dan *stemming* sebelumnya menjadi input untuk deteksi jenis kata berdasarkan kata dasar dan imbuhan. Kolom jenis kata (token) merupakan jenis kata yang akan digunakan pada tahap selanjutnya, yaitu analisis sintaksis.

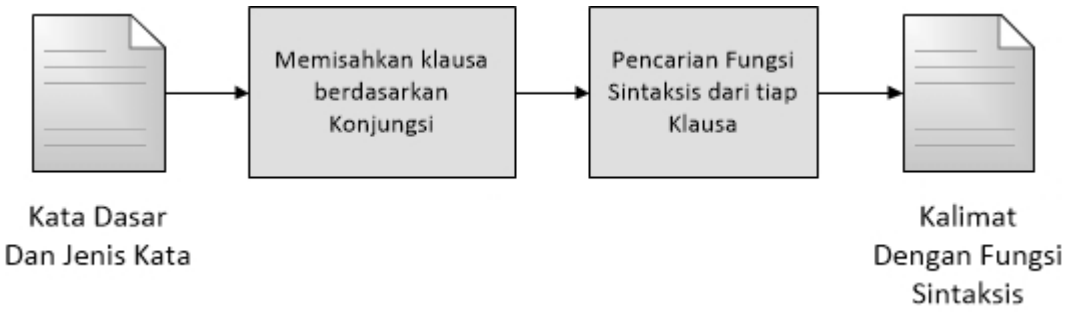
3.2.3. Proses Utama

Proses utama adalah proses lanjutan setelah *pre-processing* selesai dilakukan. Hasil dari *pre-processing* akan diolah untuk mencari fungsi-fungsi sintaksis seperti subjek, predikat, objek, dan keterangan. Kalimat tanya akan dibangkitkan menggunakan *template* berdasarkan fungsi-fungsi sintaksis yang telah ditemukan. Proses utama dalam penelitian ini dibagi menjadi dua proses yaitu analisis sintaksis untuk mencari fungsi sintaksis dan pembangkitan pertanyaan untuk membangkitkan pertanyaan berdasarkan *template* sintaksis.

3.2.3.1. Analisis Sintaksis

Kalimat terdiri dari satu atau lebih klausa yang disambung dengan konjungsi. Klausa memiliki fungsi-fungsi sintaksis yang salah satunya adalah fungsi sintaksis yang sangat penting, yaitu predikat. Predikat penting karena klausa sendiri adalah satuan sintaksis yang bersifat predikatif.

Berdasarkan hal tersebut, sebelum menemukan fungsi-fungsi sintaksis dalam suatu kalimat, perlu dicari konjungsi dari kalimat tersebut yang menyambung tiap klausa. Setelah itu, mencari predikat sebagai fungsi utama yang harus ada di klausa. Berikut proses penentuan fungsi-fungsi sintaksis dalam suatu kalimat.



Gambar 3.7 Proses Analisis Sintaksis

Pencarian fungsi sintaksis setelah terbagi menjadi beberapa klausa dimulai dengan mencari predikat. Karena seperti yang telah dijelaskan sebelumnya, klausa adalah satuan sintaksis yang predikatif, sehingga kehadiran predikat sangat penting dalam klausa itu sendiri.

Token-token yang sudah memiliki informasi kata dasar, imbuhan, dan jenis kata yang telah didapatkan dari tahap-tahap sebelumnya akan dicari jenis kata konjungsi. Jika jenis kata konjungsi ada pada kumpulan token dalam satu kalimat tersebut, maka token-token akan dipisah berdasarkan kata konjungsi yang ditemukan sehingga terbagi menjadi klausa. Setelah klausa ditemukan, maka pencarian fungsi-fungsi sintaksis dapat dilakukan.

Pencarian fungsi-fungsi sintaksis dari suatu klausa menggunakan aturan-aturan berdasarkan jenis kata yang telah didapatkan pada tahap sebelumnya. Berikut adalah tabel aturan pencarian fungsi-fungsi sintaksis dalam klausa.

Tabel 3.3 Aturan Pencarian Fungsi Sintaksis

No	Fungsi	Aturan
1	Predikat	Cari kata kerja pada kalimat tersebut dengan ketentuan : 1. Bila kata kerja hanya ada satu, maka jadikan predikat. K. Kerja Contoh : Ibu pergi ke pasar. P 2. Bila kata kerja ada dua, maka ambil kata kerja yang kedua sebagai predikat dan kata kerja pertama sebagai subjek. K. Kerja K. Kerja Contoh : Berenang menyehatkan tubuh. S P Setelah menemukan predikat, maka dilakukan pemeriksaan apakah predikat tersebut predikat aktif atau pasif. Berikut ketentuannya : 1. Bila predikat tersebut memiliki awalan di-, ter-, dan te-, maka predikat tersebut adalah pasif. Contoh : Tikus dimakan kucing. P. Pasif 2. Bila predikat tersebut tidak memiliki awalan atau memiliki awalan selain ketentuan no 1, maka predikat tersebut adalah aktif. Contoh : Kucing memakan tikus. P. Aktif

No	Fungsi	Aturan
2	Subjek	Lihat kata sebelum predikat dengan ketentuan : 1. Bila kata tersebut adalah kata benda atau kata kerja, jadikan subjek. K. Benda K. Kerja Contoh : Kucing memakan makanannya. S P 2. Bila kata tersebut adalah pronomina (kata ganti atau kata tunjuk), maka : a. Jika tidak ada kata benda sebelum kata ini, jadikan kata ini sebagai subjek. Pro Contoh : Dia melempar batu. S P. Aktif b. Jika ada kata benda sebelum kata ini, jadikan kata benda dan tersebut menjadi subjek. KB Pro Contoh : Kucing itu memakan makanannya. S P. Aktif
3	Objek	Lihat kata setelah predikat dengan ketentuan jika kata tersebut adalah kata benda, kata tunjuk, kata bilangan atau kata ganti, maka jadikan kata tersebut Objek. KB Contoh : Kucing itu memakan makanannya. P. Aktif O
4	Keterangan	Bila saat mendeteksi tiap kata menemukan kata keterangan tempat, cara, waktu, atau akibat, beri tanda bahwa kata tersebut adalah kata keterangan. K.Ket Contoh : Kucing itu memakan makanannya di teras. Ket. Tempat

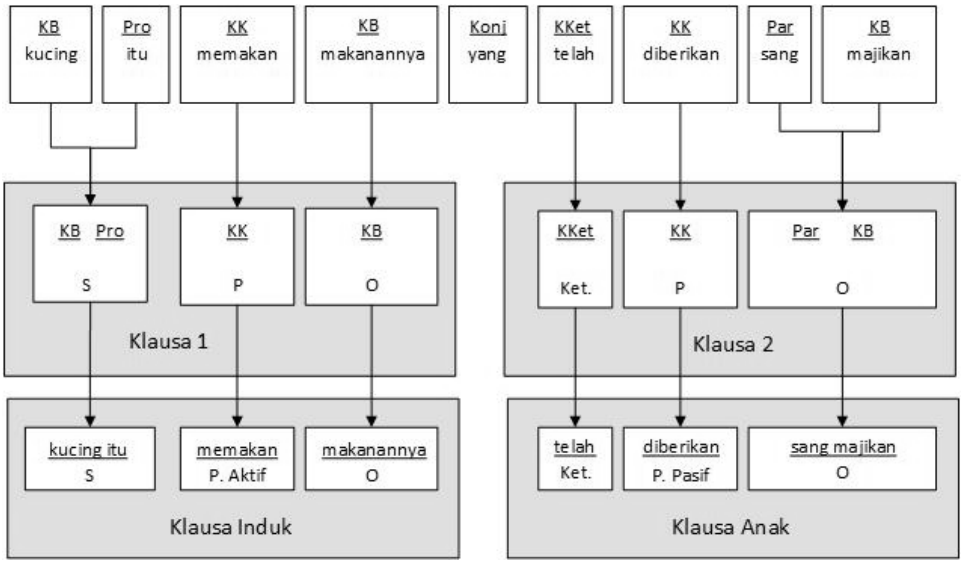
Setelah semua token ditemukan fungsi sintaksisnya dengan menggunakan aturan pada tabel 3.3, maka dilakukan pemeriksaan terhadap klausa apakah klausa

tersebut klausa bebas atau klausa terikat dengan ketentuan yang dijelaskan pada subbab 2.1.4 mengenai klausa bebas dan klausa terikat. Aturan penentuan klausa bebas atau klausa terikat dijelaskan pada tabel 3.4 di bawah ini.

Tabel 3.4 Aturan Penentuan Jenis Klausa

Jenis Klausa	Aturan
Klausa Bebas / Klausa Induk	1. Jika konjungsi pada kalimat merupakan konjungsi koordinatif, klausa sudah dipastikan klausa bebas. 2. Jika fungsi sintaksisnya lengkap, klausa tersebut adalah klausa bebas.
Klausa Terikat / Klausa Anak	Klausa anak adalah klausa yang berada setelah konjungsi subordinatif. Berikut adalah beberapa kondisi setelah menemukan klausa anak : 1. Jika tidak memiliki subjek (dilesapkan), subjek klausa anak sama dengan subjek klausa induk. 2. Jika fungsi sintaksis lengkap, lihat posisi klausa. Bila posisi klausa setelah dan bersebelahan dengan konjungsi, maka klausa itu adalah klausa anak.

Berikut adalah penggambaran pemisahan klausa dan pencarian fungsi sintaksis berdasarkan aturan di atas.



Gambar 3.8 Contoh Analisis Sintaksis

Pada contoh di atas, token-token selain token yang memiliki jenis kata konjungsi disatukan menjadi klausa. Token-token yang posisinya berada sebelum kata konjungsi disebut klausa 1, sedangkan token-token yang posisinya berada setelah konjungsi disebut klausa 2. Setelah token-token tersebut terbagi menjadi klausa, maka pencarian fungsi sintaksis dilakukan. Pencarian fungsi sintaksis menggunakan aturan pada tabel 3.3, dimana predikat dicari terlebih dahulu. Setelah mendapatkan predikat, dilakukan pencarian subjek dan objek dengan mencari kata benda yang posisinya berada di sebelum dan setelah predikat. Pada klausa 1 terlihat bahwa Pronomina (Pro) bersatu dengan KB (Kata Benda) menjadi objek karena sesuai aturan pada tabel 3.3. Jika ditemukan kata Pronomina dan kata benda sebelumnya, kedua kata tersebut menjadi subjek.

Setelah fungsi sintaksis ditemukan, maka penentuan jenis klausa dilakukan dengan menggunakan aturan pada tabel 3.4. Pada gambar 3.8 terlihat klausa 1 memiliki fungsi sintaksis yang lengkap, yaitu minimal memiliki subjek dan predikat. Oleh karena itu, klausa 1 ditentukan sebagai klausa induk. Sedangkan klausa 2 tidak memiliki fungsi sintaksis yang lengkap. Subjek pada klausa 2 tidak ditemukan, sehingga klausa 2 adalah klausa anak. Dengan melihat predikat klausa

induk yang merupakan predikat pasif dapat disimpulkan bahwa subjek dari klausa anak adalah objek dari klausa induk.

3.2.3.2. Pembangkitan Pertanyaan

Setelah tahapan analisis sintaksis selesai dilakukan, maka data yang terkumpul adalah berupa fungsi-fungsi sintaksis dari setiap kata atau kumpulan kata dalam kalimat. Pembangkitan pertanyaan akan membentuk kalimat pertanyaan yang umum dengan memanfaatkan fungsi-fungsi sintaksis tersebut sebagai jawabannya. Kata tanya yang akan dibentuk adalah siapa, apa, mana, kapan, mengapa, dan bagaimana.

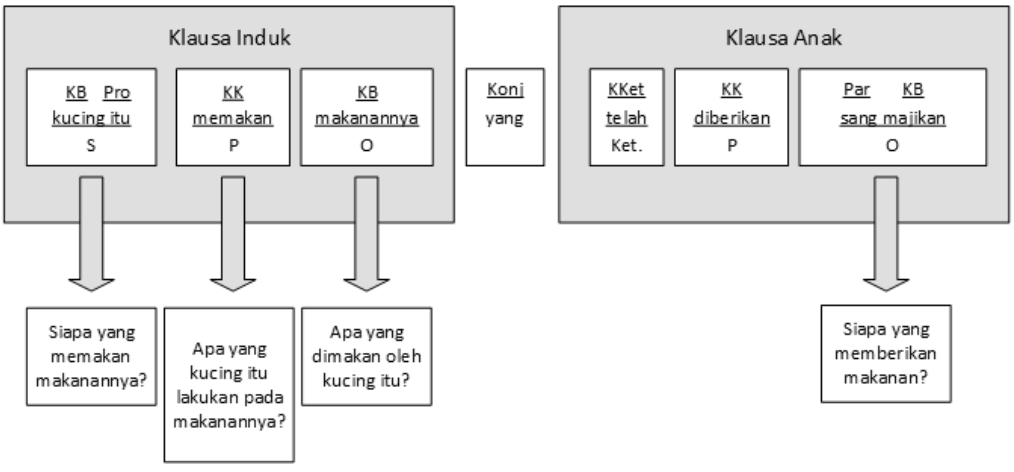
Pembangkitan kalimat pertanyaan dibantu dengan menggunakan *template* berdasarkan fungsi sintaksis yang didapatkan dari hasil analisis sintaksis. Pada *template* ini, fungsi sintaksis yang akan ditanyakan akan dihilangkan dari kalimat dan diganti dengan kata tanya. Berikut adalah tabel *template* pertanyaan yang digunakan untuk membuat pertanyaan.

Tabel 3.5 *Template* Pertanyaan

Kata Tanya	***Template* Pertanyaan**
Siapa	Fungsi yang ditanyakan : **Subjek** Template: **Siapa yang +** (Predikat Aktif) + (Objek) + (Keterangan) + ?
	Fungsi yang ditanyakan : **Objek** Template: **Siapa yang +** (Predikat Pasif) + (Subjek) + (Keterangan) + ?
Apa	Fungsi yang ditanyakan : **Subjek** Template: **Apa yang +** (Predikat Aktif) + (Objek) + (Keterangan) + ?
	Fungsi yang ditanyakan : **Predikat Aktif** Template: **Apa yang +** (Subjek) + **lakukan pada +** (Objek) + (Keterangan) + ?

Kata Tanya	*Template* Pertanyaan
	Fungsi yang ditanyakan : **Predikat Pasif** Template: **Apa yang +** (Objek) + **lakukan pada +** (Subjek) + (Keterangan) + ?
	Fungsi yang ditanyakan : **Objek** Template: **Apa yang +** (Predikat Pasif) + **oleh** + (Subjek) + (Keterangan) + ?
Mana	Fungsi yang ditanyakan : **Ket. Tempat** Template: (Di/Ke/Dari) **mana +** (Subjek) + (Predikat) + (Objek) + ?
Kapan	Fungsi yang ditanyakan : **Ket. Waktu** Template: **Kapan +** (Subjek) + (Predikat) + (Objek) + ?
Mengapa	Fungsi yang ditanyakan : **Ket. Sebab** Template: **Mengapa +** (Subjek) + (Predikat) + (Objek) + ?
Bagaimana	Fungsi yang ditanyakan : **Ket. Cara** Template: **Bagaimana +** (Subjek) + (Predikat) + (Objek) + ?

Contoh pembangkitan pertanyaan menggunakan *template* pada tabel 3.4 menggunakan contoh teks masukan digambarkan pada gambar di bawah ini.



Gambar 3.9 Contoh Pembangkitan Pertanyaan.

Pada gambar 3.9, terlihat bahwa Klausa 1 memiliki tiga pertanyaan dan Klausa 2 memiliki satu pertanyaan dengan menggunakan aturan *template* pada tabel 3.4. Untuk lebih jelasnya, dapat dilihat pada tabel di bawah ini.

Tabel 3.6 Contoh Pembangkitan Pertanyaan

Kata Tanya	Contoh Pertanyaan
Siapa	Fungsi yang ditanyakan : **Subjek** Klausa : kucing itu memakan makanannya. Kalimat Tanya : **Siapa yang** memakan makanannya? Jawaban : kucing itu
	Fungsi yang ditanyakan : **Objek** Contoh Klausa : (makanan kucing itu) telah diberikan sang majikan. Kalimat Tanya : **Siapa yang** memberikan makanan kucing itu? Jawaban : sang majikan.
Apa	Fungsi yang ditanyakan : **Predikat Aktif** Contoh Klausa : kucing itu memakan makanannya. Kalimat Tanya : **Apa yang** kucing itu **lakukan pada** makanannya? Jawaban : Kucing itu memakan makanannya.
	Fungsi yang ditanyakan : **Objek** Contoh Klausa : Kucing itu memakan makanannya. Kalimat Tanya : **Apa yang** dimakan **oleh** kucing itu ? Jawaban : makanannya

Hasil dari pembangkitan pertanyaan untuk kalimat “kucing itu memakan makanannya yang telah diberikan sang majikan” memiliki empat kalimat pertanyaan yang terdiri dari dua kata tanya, yaitu siapa dan apa.

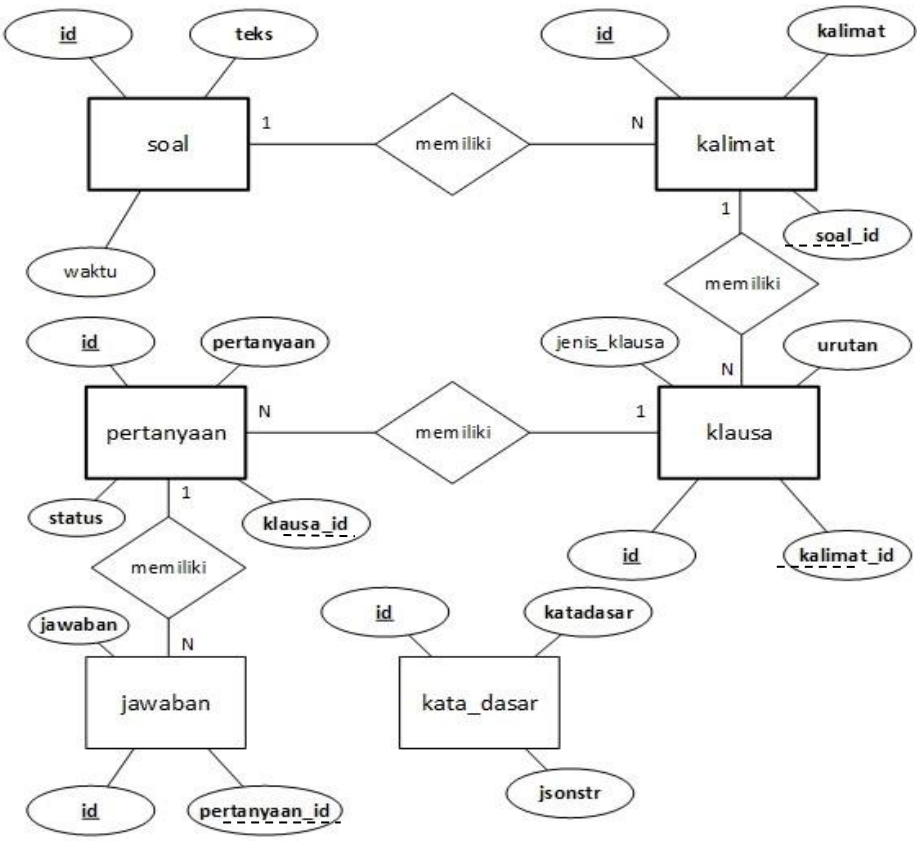
3.4 Analisis Basis Data

Analisis basis data menggambarkan data yang dipakai pada sistem pembangkitan pertanyaan. Dalam sistem ini, data yang disimpan adalah hasil dari proses pembangkitan pertanyaan yang akan digunakan untuk mengukur akurasi

pertanyaan yang dibuat. Berikut adalah penggambaran basis data yang digambarkan menggunakan *Entity Relationship Diagram* [26].

3.4.1 *Entity Relationship Diagram*

Model ERD pada sistem pembangkitan pertanyaan yang akan dibangun adalah sebagai berikut.



Gambar 3.10 *Entity Relationship Diagram*

3.4.2 Kamus Data ERD

Berikut adalah tabel kamus data ERD yang telah digambarkan sebelumnya.

Tabel 3.7 Kamus Data

Entitas	Atribut
soal	id, teks, waktu
kalimat	id, soal_id, kalimat
klausa	id, kalimat_id, jenis_klausa, urutan
pertanyaan	id, klausa_id, pertanyaan, status
jawaban	id, pertanyaan_id, jawaban
kata_dasar	id, katadasar, jsonstr

Entitas kata_dasar adalah entitas yang memiliki data kamus kata dasar yang didapatkan dari kateglo. Atribut jsonstr pada entitas ini berisi data-data yang didapatkan dari kateglo seperti arti kata, jenis kata, sinonim, dan sebagainya. Sedangkan untuk tiga entitas lainnya digunakan untuk menyimpan hasil pertanyaan yang berhasil diciptakan oleh sistem. Entitas soal menyimpan teks yang dimasukkan ke dalam sistem, entitas kalimat menyimpan kalimat-kalimat yang terdapat pada teks, entitas klausa menyimpan jenis-jenis klausa (terikat atau bebas) yang terdapat pada tiap kalimat, dan entitas pertanyaan serta jawaban menyimpan pertanyaan dan jawaban yang berhasil diciptakan.

Pada entitas t_pertanyaan terdapat atribut status yang berisi status pertanyaan setelah diuji kebenaran pertanyaan yang dibangkitkan. Status pertanyaan ada dua, yaitu **RELEVAN** dan **TIDAK RELEVAN**. Status RELEVAN berarti pertanyaan yang diciptakan oleh sistem relevan dengan teks yang diberikan, sedangkan status TIDAK RELEVAN berarti pertanyaan yang diciptakan tidak relevan dengan teks yang diberikan. Status ini akan digunakan untuk melihat akurasi dari sistem pembangkitan pertanyaan ini.

3.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan pada sistem yang mencakup kebutuhan perangkat lunak, perangkat keras, dan pengguna.

3.5.1 Analisis Perangkat Lunak

Analisis perangkat lunak dilakukan untuk mengetahui spesifikasi perangkat lunak yang dibutuhkan untuk mendukung sistem dapat digunakan. Spesifikasi minimum perangkat lunak yang direkomendasikan adalah sebagai berikut.

1. Sistem operasi *Windows* 7.
2. *Apache Tomcat* 7.
3. Java 7.
4. MySQL.
5. *Google Chrome* dan *Mozilla Firefox* sebagai *web browser*.

3.5.2 Analisis Perangkat Keras

Analisis perangkat keras dilakukan untuk mengetahui spesifikasi perangkat keras yang dibutuhkan agar sistem dapat berjalan dengan baik. Spesifikasi minimum perangkat keras yang direkomendasikan adalah sebagai berikut.

1. Prosesor dengan kecepatan 1 GHz.
2. RAM 1 GB.
3. Harddisk 80 GB.
4. Monitor 15”.
5. Mouse dan Keyboard.

3.5.3 Analisis Pengguna

Pengguna atau *user* yang menggunakan sistem ini adalah User. Secara spesifik, *user* yang dapat mengunakan sistem ini adalah *user* Bahasa Indonesia,

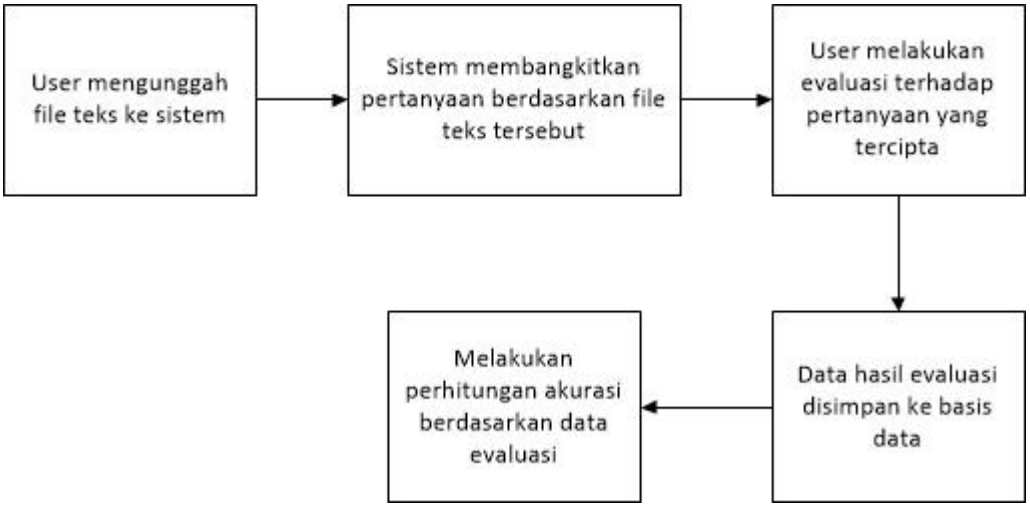
karena teks masukan pada sistem ini adalah teks soal cerita bahasa Indonesia. Selain karena teks masukannya adalah teks soal cerita bahasa Indonesia, *user* bahasa Indonesia dibutuhkan untuk mengevaluasi hasil pertanyaan yang diciptakan untuk menghitung akurasi sistem pembangkit pertanyaan ini. Berikut adalah rincian analisis pengguna pada sistem ini.

Tabel 3.8 Analisis Pengguna

Pengguna	**Tanggung Jawab**	**Hak Akses**	**Tingkat Keterampilan**	**Pengalaman**
User	Mengunggah teks soal cerita bahasa Indonesia dan mengevaluasi pertanyaan yang diciptakan oleh sistem.	Pembangkitan pertanyaan dan evaluasi pertanyaan.	Mampu memahami petunjuk-petunjuk dalam sistem.	Memahami tata bahasa Indonesia baku dan memahami cara mengoperasikan komputer serta menggunakan aplikasi web.

3.6 Analisis Pengujian

Pengujian akurasi akan dilakukan dengan menghitung jumlah pertanyaan yang RELEVAN oleh *user* saat melakukan evaluasi pertanyaan. RELEVAN berarti pertanyaan tersebut relevan dengan teks yang dimasukkan. Data-data ini disimpan di dalam tabel pertanyaan. Berikut adalah skema pengujian terhadap sistem.



Gambar 3.11 Skema Pengujian Sistem

Perhitungan akurasi pertanyaan yang berhasil diciptakan oleh sistem diambil berdasarkan data hasil evaluasi pertanyaan dengan menjumlahkan pertanyaan yang memiliki status RELEVAN dan membagikannya dengan jumlah total pertanyaan untuk satu teks atau soal. Berikut adalah rumus perhitungannya.

$$Akurasi = \frac{Jumlah\ Pertanyaan\ RELEVAN}{Total\ Pertanyaan} * 100\%$$

Berikut adalah contoh penggunaan rumus di atas dengan data yang dimasukkan ke dalam sistem.

Tabel 3.9 Contoh Pengujian

Teks Masukan	**Kucing itu memakan makanannya yang telah diberikan sang majikan. Sang majikan puas melihat kucingnya makan dengan lahap.**		
No	**Pertanyaan**	**Jawaban**	**Status**
1	Siapa yang memakan makanannya?	Kucing itu	RELEVAN
2	Apa yang dimakan oleh kucing itu?	Makanannya	RELEVAN
3	Apa yang diberikan sang majikan?	Telah	TIDAK RELEVAN

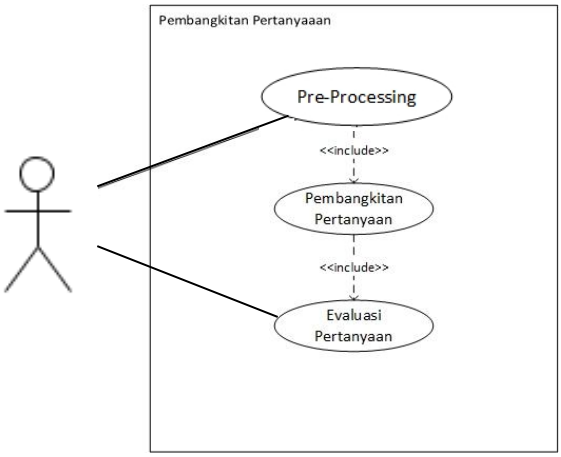
Pada tabel 3.9 ditemukan tiga pertanyaan yang dapat dibangkitkan oleh sistem berdasarkan teks masukannya. Dengan menggunakan rumus akurasi, maka hasil akurasi dari contoh pada tabel 3.9 adalah 2/3 * 100%, sehingga hasilnya adalah 66.67%.

3.7 Perancangan Sistem

Perancangan sistem ini bertujuan untuk menggambarkan bagaimana sistem akan dibangun. Tahapan pemodelan untuk pembuatan sistem pada penelitian ini menggunakan Unified Modeling Language atau yang biasa disingkat menjadi UML. Diagram-diagram yang digunakan dalam merancang sistem adalah *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram* [4].

3.7.1 *Use Case Diagram*

Aktor dalam sistem ini ada satu, yaitu *user* yang melakukan input teks pada sistem yang selanjutnya proses pembangkitan pertanyaan akan dilakukan. *User* pun memberikan penilaian pada pertanyaan yang berhasil diciptakan oleh sistem pada proses evaluasi pertanyaan untuk mengukur relevansi hasil pertanyaan yang dibangkitkan terhadap teks masukan. Berikut adalah *use case diagram* pada sistem ini.



Gambar 3.12 *Use Case Diagram*

Penjelasan tiap *use case* akan dijelaskan dalam bentuk *use case scenario*. Penjelasan dalam skenario meliputi tujuan *use case*, aktor yang melakukan *use case* tersebut, dan skenario yang berjalan dalam *use case*. Berikut adalah *use case scenario* untuk *use case pre-processing*, pembangkitan pertanyaan dan evaluasi pertanyaan

Tabel 3.10 *Use Case Scenario Pre-processing*.

Identifikasi	
Nama	*Pre-processing*.
Tujuan	Mendapatkan kata dasar, imbuhan, dan jenis kata.
Deskripsi	Proses awal dalam sistem untuk mendapatkan data yang dibutuhkan pada proses utama. Proses ini mencakup mengunggah *file*, *case folding*, tokenisasi, *stemming*, dan deteksi jenis kata.
Aktor	*User*.
***Use Case* yang berkaitan**	Proses Utama
Skenario Utama	
Kondisi Awal	Halaman untuk mengunggah file ditampilkan.
Aksi Aktor	**Reaksi Sistem**
1. Klik tombol Cari dan pilih file yang ingin diunggah.	2. Sistem membaca dan memeriksa *file* yang diunggah.
	3. Bila *file* yang diunggah memiliki tipe ekstensi dan ukuran yang *valid*, maka proses selanjutnya dapat dilakukan. 4. Melakukan proses *case folding*. 5. Melakukan proses tokenisasi kalimat. 6. Melakukan proses tokenisasi kata. 7. Melakukan proses *stemming*. 8. Melakukan proses deteksi jenis kata.
Skenario Alternatif	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan pesan bahwa *file* yang diunggah terlalu besar atau salah tipe ekstensi.
2. Menggunggah kembali *file* yang memiliki ukuran dan tipe ekstensi yang sesuai.	3. Sistem membaca dan memeriksa *file* yang diunggah
	4. Bila *file* yang diunggah *valid*, maka proses selanjutnya dapat dilakukan.

	5. Melakukan proses *case folding*. 6. Melakukan proses tokenisasi kalimat. 7. Melakukan proses tokenisasi kata. 8. Melakukan proses *stemming*.
Aksi Aktor	**Reaksi Sistem**
	9. Melakukan proses deteksi jenis kata.
Kondisi Akhir	Kata dasar, imbuhan, dan jenis kata ditemukan.

Tabel 3.11 *Use Case Scenario* Pembangkitan Pertanyaan.

Identifikasi	
Nama	Pembangkitan Pertanyaan
Tujuan	Membangkitkan pertanyaan menggunakan *template* berdasarkan fungsi sintaksis.
Deskripsi	Proses utama terdiri dari dua proses, yaitu analisis sintaksis untuk mencari fungsi sintaksis menggunakan data yang didapatkan dari *pre-processing*, dan pembangkitan pertanyaan untuk membangkitkan pertanyaan menggunakan *template* berdasarkan fungsi sintaksis.
Aktor	*User*
***Use Case* yang berkaitan**	*Pre-processing*, Evaluasi Pertanyaan
Skenario Utama	
Kondisi Awal	Data kata dasar, imbuhan, dan jenis kata yang didapatkan dari *pre-processing*.
Aksi Aktor	**Reaksi Sistem**
	1. Melakukan pencarian fungsi sintaksis menggunakan aturan analisis sintaksis.
	2. Menggunakan *template* berdasarkan fungsi sintaksis yang ditemukan untuk membangkitkan pertanyaan.
	3. Sistem menampilkan hasil semua proses dan daftar pertanyaan beserta jawabannya yang berhasil dibangkitkan.
Kondisi Akhir	User dapat melihat hasil dari tiap proses dan daftar pertanyaan yang berhasil dibangkitkan oleh sistem beserta dengan jawabannya.

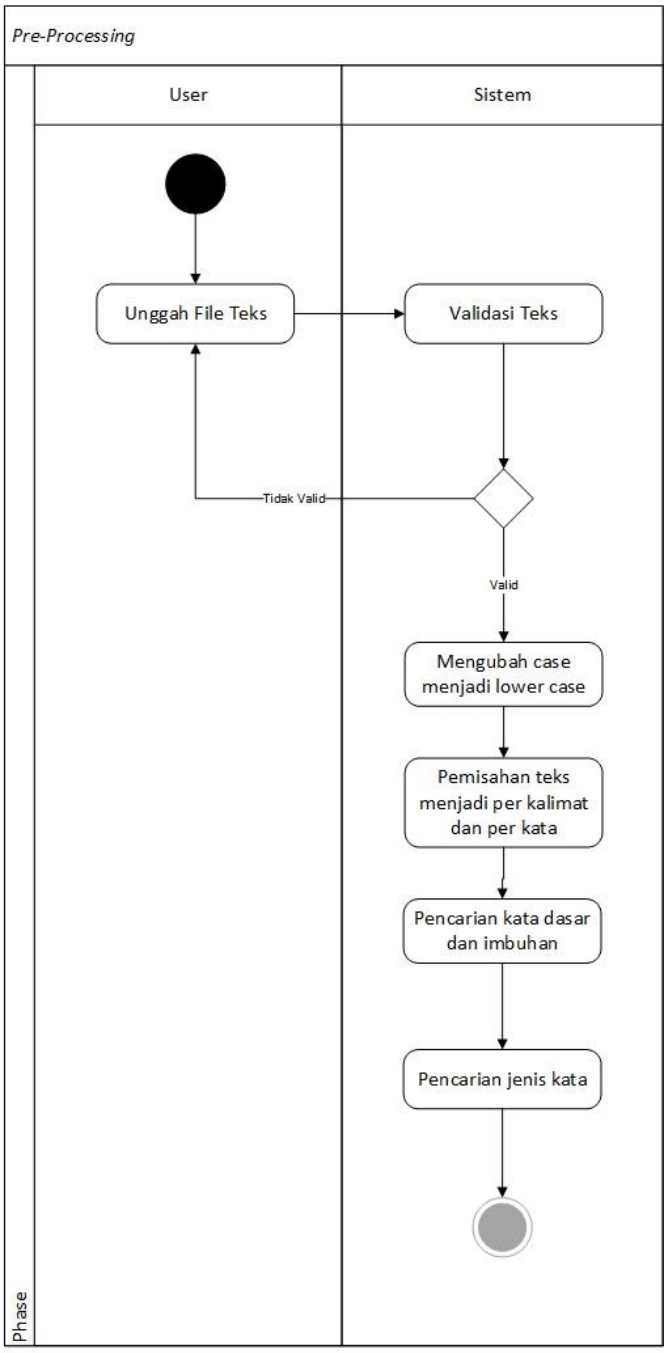
Tabel 3.12 *Use Case Scenario* Evaluasi Pertanyaan.

Identifikasi	
Nama	Evaluasi Pertanyaan
Tujuan	Mendapatkan data untuk menghitung akurasi pertanyaan yang dibangkitkan.

Deskripsi	Proses evaluasi pertanyaan dilakukan oleh User dengan memilih pertanyaan-pertanyaan yang relevan dengan teks masukan.
Aktor	User
***Use Case* yang berkaitan**	Proses Utama
Skenario Utama	
Kondisi Awal	Tampilan daftar pertanyaan yang dibangkitkan oleh sistem beserta jawabannya.
Aksi Aktor	**Reaksi Sistem**
1. Memilih pertanyaan yang relevan dengan teks masukan.	2. Menyimpan data pertanyaan dan jawaban beserta informasi bahwa pertanyaan tersebut relevan atau tidak.
Kondisi Akhir	User dapat melihat persentase akurasi pertanyaan yang dibangkitkan dari teks masukan.

3.7.2 *Activity Diagram*

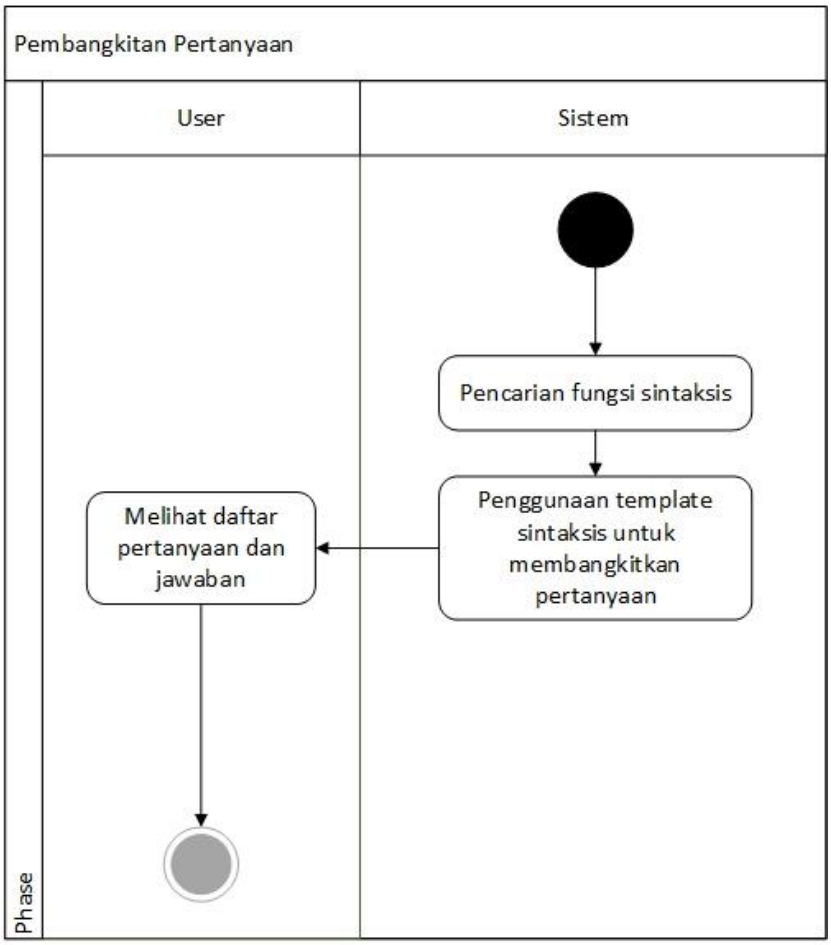
Activity diagram dibentuk untuk menggambarkan alur aktifitas yang terjadi di antara aktor dan sistem yang ada di use case diagram. Dikarenakan terdapat tiga *use case* pada *use case diagram*, maka *activity diagram* yang tercipta pun ada tiga. Berikut adalah *activity diagram* yang pertama, yaitu *pre-processing*.



Gambar 3.13 ***Activity Diagram Pre-processing***

Pada *activity diagram* di atas digambarkan bahwa *user* melakukan inisialisasi dengan menggungah *file* teks kepada sistem, sistem yang menerima teks masukan akan melakukan validasi terhadap teks tersebut. Jika file yang diberikan tidak sesuai terlalu besar atau bukan file ber-ekstensi .txt, maka user

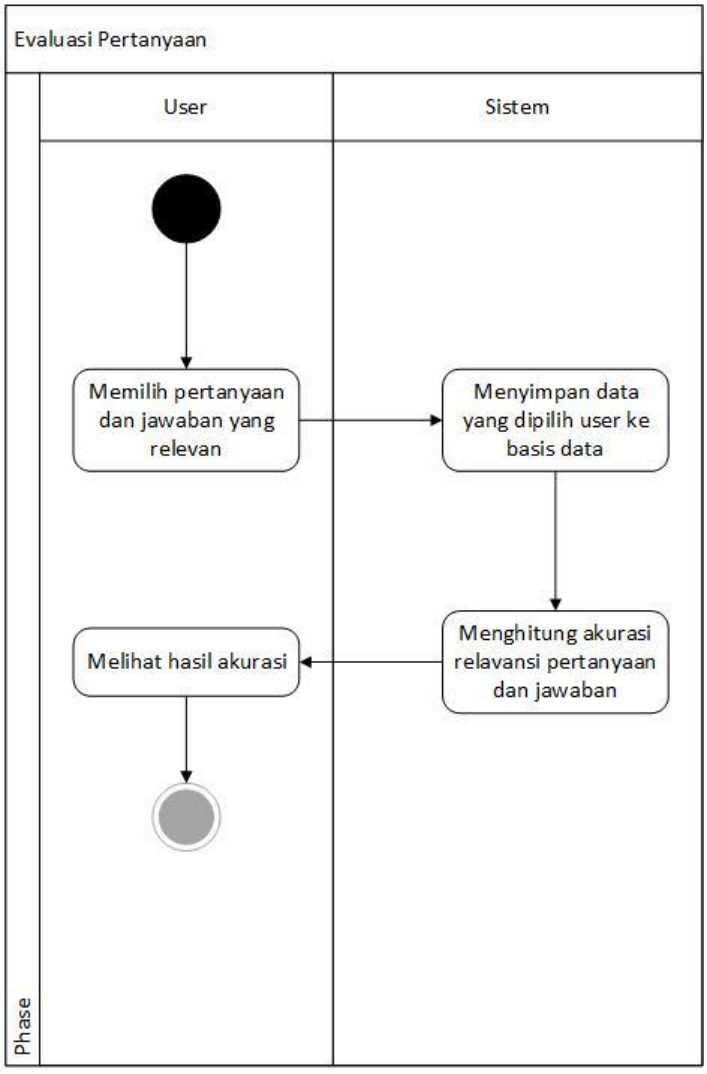
harus menggungah teks lain. Jika file yang diberikan lolos validasi, maka proses selanjutnya dilakukan, yaitu merubah *case*, pemisahan teks, pencarian kata dasar, dan pendeteksian jenis kata. Selanjutnya adalah *activity diagram* untuk proses utama.



Gambar 3.14 *Activity Diagram* Pembangkitan Pertanyaan

Pada proses utama, data yang didapatkan dari *pre-processing* akan diolah untuk dicari fungsi-fungsi sintaksisnya. Dengan mengetahui fungsi-fungsi sintaksis dari tiap kalimat, penggunaan *template* sintaksis dapat dilakukan untuk membangkitkan pertanyaan dan jawaban berdasarkan teks masukan. Setelah *template* sintaksis digunakan, daftar pertanyaan dan jawaban yang berhasil dibangkitkan akan ditampilkan kepada *user*.

Langkah selanjutnya adalah saat *user* menerima hasil dari sistem berupa kalimat-kalimat pertanyaan beserta jawabannya, *user* dapat melakukan evaluasi ketepatan atau akurasi dari pertanyaan yang berhasil dibuat oleh sistem. Setelah *user* selesai melakukan evaluasi, maka data teks soal, pertanyaan-pertanyaan, dan jawaban-jawabannya disimpan dalam database. Berikut adalah *activity diagram* untuk evaluasi pertanyaan.



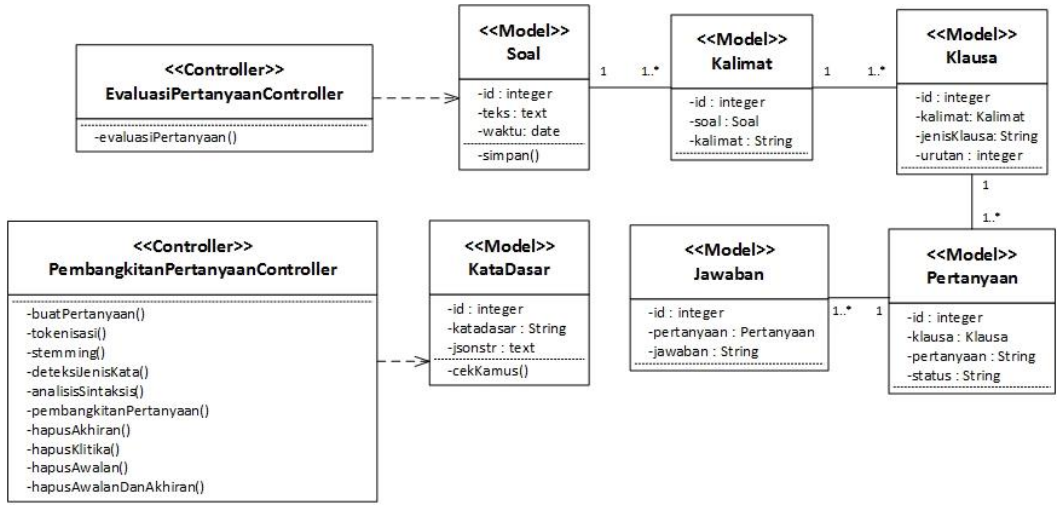
Gambar 3.15 *Activity Diagram* Evaluasi Pertanyaan.

Dengan menyimpan data evaluasi yang dilakukan oleh *user*, maka persentase akurasi pertanyaan yang dibangkitkan sistem dapat dilihat oleh *user* itu sendiri atau sebagai hasil pengujian oleh peneliti dan pengembang sistem.

3.7.3 *Class Diagram*

Sistem menggunakan Framework berasaskan MVC (*Model*, *View*, *Controller*) sehingga *class-class* yang tercipta akan terbagi dua yaitu *Model* dan *Controller* dikarenakan *View* tidak berbentuk *class* melainkan berbentuk halaman web seperti HTML.

Model bersifat sebagai penampung properti-properti yang sesuai dengan tabel yang ada di database. Sedangkan *Controller* berisi operasi-operasi atau fungsi-fungsi yang digunakan untuk mengolah input yang dimasukkan dalam sistem. Berikut adalah *class diagram* beserta relasi-relasi antar *class*.

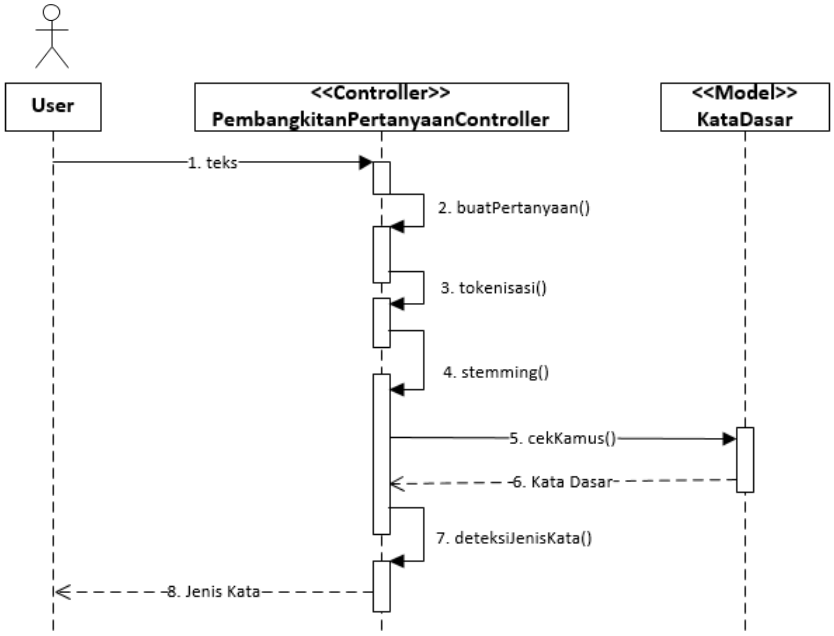


Gambar 3.16 *Class Diagram*

3.7.4 *Sequence Diagram*

Dalam aplikasi ini terdapat tiga proses, yaitu *pre-processing*, pembangkitan pertanyaan, dan evaluasi pertanyaan. Oleh karena itu terdapat tiga

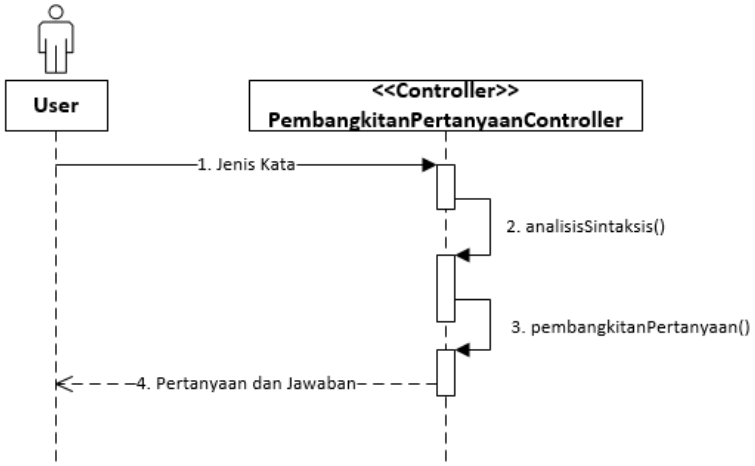
sequence diagram yang menggambarkan interaksi antar *class* yang terjadi pada tiga proses tersebut. Berikut adalah *sequence diagram* untuk *pre-processing*.



Gambar 3.17 *Sequence Diagram Pre-Processing*

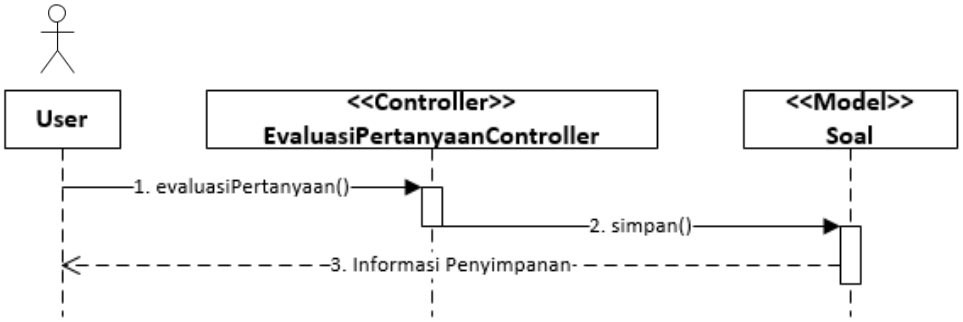
Operasi buatPertanyaan() adalah operasi awal yang dipanggil saat *user* melakukan input teks soal cerita pada sistem. Jika ada kesalahan dalam memasukkan teks yang berupa file, maka ada pesan balikan yang memberitahukan *user* bahwa file yang diunggah tidak sesuai format, misalnya salah ekstensi atau ukuran yang terlalu besar. Untuk pengecekan pada kamus saat melakukan *stemming*, PembangkitanPertanyaanController memanggil operasi cekKamus() yang ada pada Model KataDasar. Pesan balikan dari Model KataDasar kepada PembangkitanPertanyaanController adalah pesan apakah kata dasar yang dicari pada kamus ditemukan atau tidak. Pendeteksian jenis kata dasar menggunakan operasi deteksiJenisKata().

Proses selanjutnya adalah proses pembangkitan pertanyaan. Proses ini adalah proses utama dimana data yang didapatkan dari *pre-processing* diolah untuk mendapatkan fungsi sintaksis dan membangkitkan pertanyaan menggunakan *template* berdasarkan fungsi sintaksis yang ditemukan. Berikut adalah *sequence diagram* untuk proses pembangkitan pertanyaan.



Gambar 3.18 *Sequence Diagram* Pembangkitan Pertanyaan

Untuk proses yang ketiga yaitu proses evaluasi pertanyaan, penggambaran interaksi antar *class* pada proses tersebut digambarkan pada *sequence diagram* di bawah ini.



Gambar 3.19 *Sequence Diagram* Evaluasi Pertanyaan

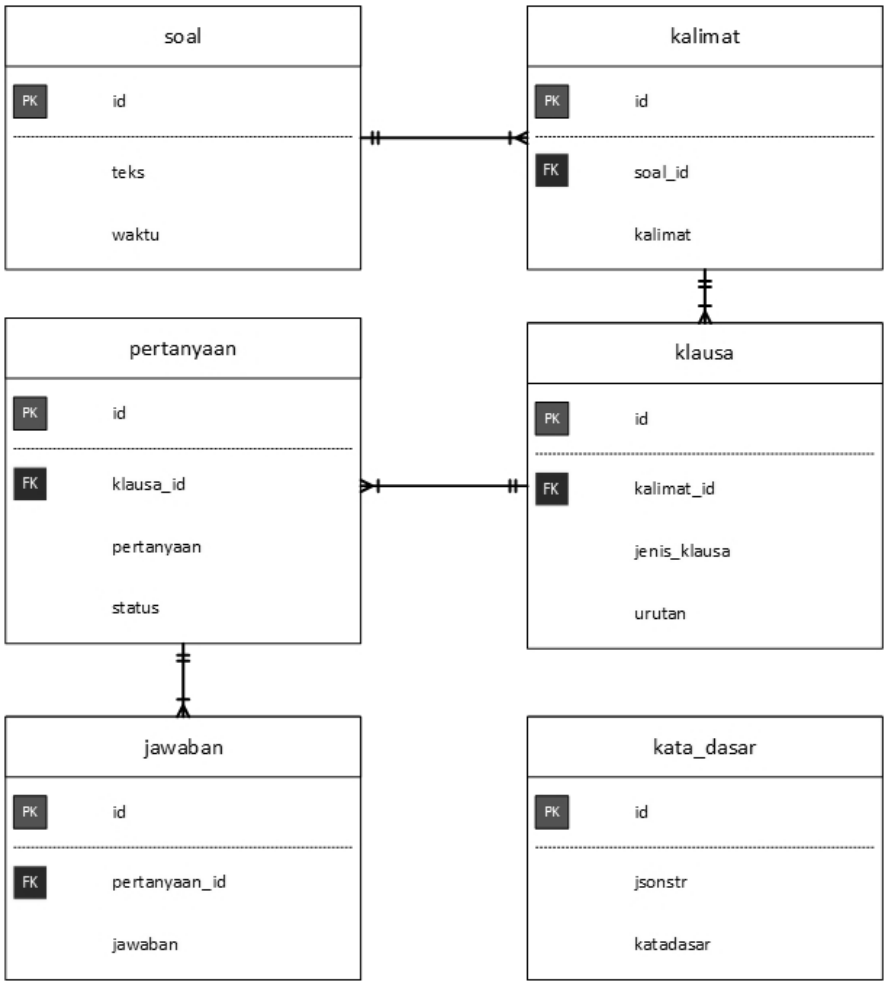
Pada EvaluasiPertanyaanController terdapat satu operasi yaitu evaluasiPertanyaan(). Operasi tersebut akan membaca *feedback* dari *user* tentang pertanyaan yang telah diciptakan oleh sistem, apakah sesuai dan relevan atau tidak. Setelah evaluasi pertanyaan pada operasi tersebut selesai, maka akan dipanggil operasi simpan() pada Model Soal. Operasi ini akan menyimpan semua kalimat pertanyaan beserta jawaban yang berhasil diciptakan sistem dan menyimpan status tiap pertanyaan apakah pertanyaan tersebut diterima atau tidak.

3.8 Perancangan Basis Data

Perancangan basis data dilakukan untuk memberikan gambaran terhadap basis data yang akan dibuat. Perancangan basis data menggunakan skema relasi dan struktur tabel. Skema relasi digunakan untuk menggambarkan relasi atau hubungan data dari tiap tabel di basis data, sedangkan struktur tabel menjelaskan atribut-atribut atau kolom-kolom dalam tabel.

3.8.1 Skema Relasi

Seperti yang telah dijelaskan sebelumnya, skema relasi menggambarkan relasi data tiap tabel. Dalam skema ini terlihat nama tabel, nama kolom-kolomnya, serta *primary key* dan *foreign key*.



Gambar 3.20 Skema Relasi

3.8.2 Struktur Tabel

Penjelasan tabel-tabel pada basis data yang digambarkan pada skema relasi di atas akan dijelaskan pada struktur tabel berikut.

Tabel 3.13 Struktur Tabel soal

Nama Kolom	Tipe Data	Ukuran	Kunci	Keterangan
id	integer	11	*Primary*	NOT NULL, AUTO_INCREMENT
teks	text			NOT NULL
waktu	Datetime			NOT NULL

Tabel 3.14 Struktur Tabel kalimat

Nama Kolom	Tipe Data	Ukuran	Kunci	Keterangan
id	integer	11	*Primary Key*	NOT NULL, AUTO_INCREMENT
soal_id	integer	11	*Foreign Key*	NOT NULL, REFERENCE soal
kalimat	varchar	255		NOT NULL

Tabel 3.15 Struktur Tabel klausa

Nama Kolom	Tipe Data	Ukuran	Kunci	Keterangan
id	integer	11	*Primary Key*	NOT NULL, AUTO_INCREMENT
kalimat_id	integer	11	*Foreign Key*	NOT NULL, REFERENCE kalimat
jenis_klausa	varchar	255		
urutan	Integer	11		NOT NULL

Tabel 3.16 Struktur Tabel pertanyaan

Nama Kolom	Tipe Data	Ukuran	Kunci	Keterangan
id	integer	11	*Primary Key*	NOT NULL, AUTO_INCREMENT
klausa_id	integer	11	*Foreign Key*	NOT NULL, REFERENCE klausa
pertanyaan	varchar	255		NOT NULL
status	varchar	8		NOT NULL

Tabel 3.17 Struktur Tabel jawaban

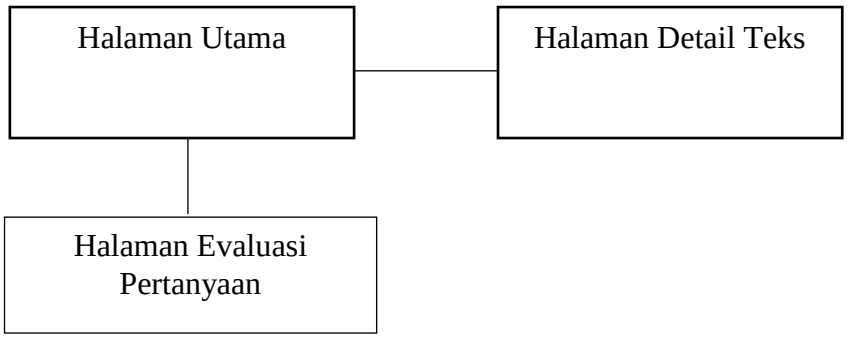
Nama Kolom	Tipe Data	Ukuran	Kunci	Keterangan
id	integer	11	*Primary Key*	NOT NULL, AUTO_INCREMENT
pertanyaan_id	Integer	11	*Foreign Key*	NOT NULL, REFERENCE pertanyaan
jawaban	varchar	255		NOT NULL

Tabel 3.18 Struktur Tabel kata_dasar

Nama Kolom	Tipe Data	Ukuran	Kunci	Keterangan
id	integer	11	*Primary Key*	NOT NULL, AUTO_INCREMENT
kata_dasar	varchar	255		NOT NULL
jsonstring	text			NOT NULL

3.9 Perancangan Struktur Menu

Struktur menu menggambarkan perancangan menu yang akan digunakan oleh *user*. Struktur menu yang akan dibuat pada sistem ini adalah sebagai berikut.



Gambar 3.21 Struktur Menu

Terdapat tiga halaman yaitu halaman utama, halaman evaluasi pertanyaan, dan halaman detail teks. Halaman utama adalah halaman yang akan ditampilkan pertama kali saat memasuki sistem. Halaman ini menunjukkan informasi akurasi sistem, total pertanyaan yang telah diciptakan, total pertanyaan yang relevan, dan *list* teks masukan yang telah dimasukkan sebelumnya. Selain itu, proses mengunggah teks sebagai teks masukan ke dalam sistem dilakukan pada halaman utama. Halaman evaluasi pertanyaan menampilkan hasil setiap proses yang dijalankan oleh sistem dalam membangkitkan pertanyaan. Disebut halaman evaluasi karena pada halaman ini pertanyaan yang relevan dengan teks masukan akan dipilih. Halaman detail teks adalah halaman yang menunjukkan kembali hasil pembangkitan pertanyaan pada teks-teks yang sudah tersimpan dalam basis data.

Penjelasan lebih jelas tentang tampilan antar muka kedua halaman tersebut akan digambarkan pada Perancangan Antar Muka.

3.10 Perancangan Antar Muka

Tampilan antar muka atau *interface* adalah tampilan dari suatu halaman entah itu halaman web ataupun halaman dalam suatu program aplikasi yang digunakan oleh *user* sebagai media untuk berkomunikasi dengan sistem. Seperti yang telah diperlihatkan dalam struktur menu, sistem ini memiliki dua halaman menu. Oleh karena itu, tampilan antar muka dalam aplikasi ini pun ada dua. Berikut adalah perancangan antar muka kedua halaman tersebut.

T01

PEMBANGKITAN PERTANYAAN

Cari

Unggah File

Total Teks	Total Pertanyaan	Total Pertanyaan Relevan	Akurasi Keseluruhan
6	50	25	50%

List Teks yang Sudah Diunggah				
Waktu Unggah	**Teks**	**Jumlah Pertanyaan**	**Jumlah Pertanyaan Relevan**	**Akurasi**
20-02-2016	Kucing itu makan dengan cepat karena kelaparan	5	5	100%

- Tombol “Cari” jika diklik muncul jendela baru untuk memilih *file* yang akan diunggah

- Tombol “Unggah File” akan mengunggah *file* yang dipilih ke sistem dan melakukan proses pembangkitan pertanyaan. Selanjutnya akan diarahkan ke T02

- Data “Teks” bila diklik akan menuju ke T03

- Muncul M01 bila penyimpanan evaluasi pertanyaan di T02 berhasil.

- Muncul M02 bila file yang diunggah tidak sesuai format.

Ukuran Layar : 1024 x 768
Font : Arial berwarna hitam
Warna Background : Putih

Gambar 3.22 Perancangan Antar Muka Halaman Utama

T02

EVALUASI PERTANYAAN

Case Folding	
Input	Output
Tokenisasi Kalimat	
Input	Output
Tokenisasi Kata	
Input	Output

Stemming			
Token	Kata Dasar	Awalan	Akhiran

Deteksi Jenis Kata					
Token	Kata Dasar	Jenis K. Dasar	Awalan	Akhiran	Jenis Token

Analisis Sintaksis		
Token	Jenis Token	Fungsi Sintaksis

Pembangkitan Pertanyaan		
Pertanyaan	Jawaban	Relevan?
Siapa yang memberi makan kucing?	Sang majikan	☐
Apa yang kucing itu lakukan pada makanannya?	Kucing itu memakan makanannya	☐

Simpan

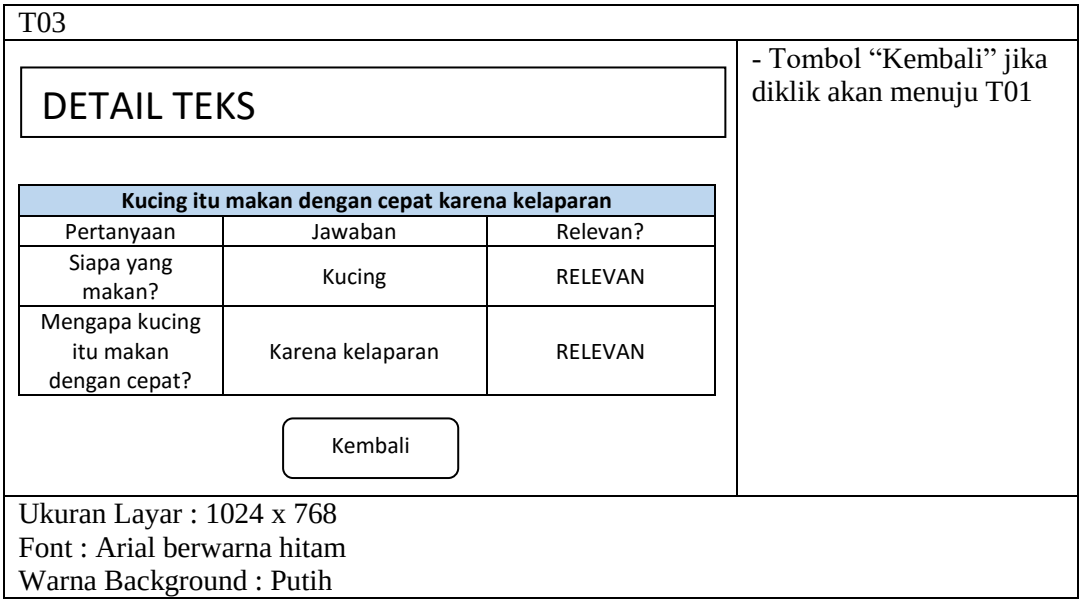
- Tombol “Simpan” jika diklik akan menyimpan evaluasi pertanyaan ke basis data.

- Checkbox pada kolom “Relevan?” untuk menentukan bahwa pertanyaan dan jawaban tersebut relevan atau tidak dengan teks.

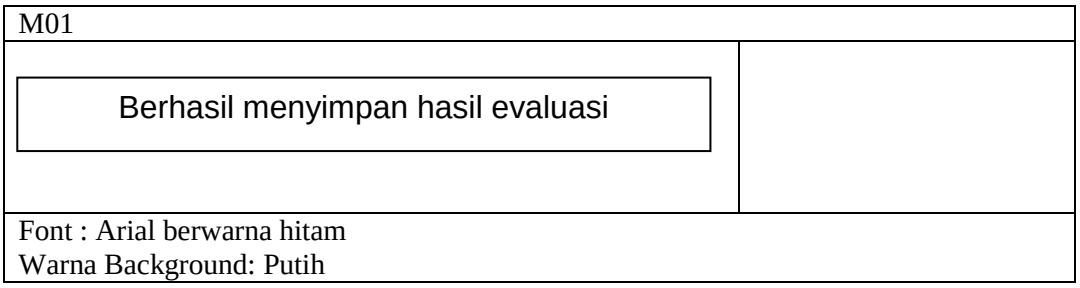
- Muncul M02 bila penyimpanan evaluasi pertanyaan gagal.

Ukuran Layar : 1024 x 768
Font : Arial berwarna hitam
Warna Background : Putih

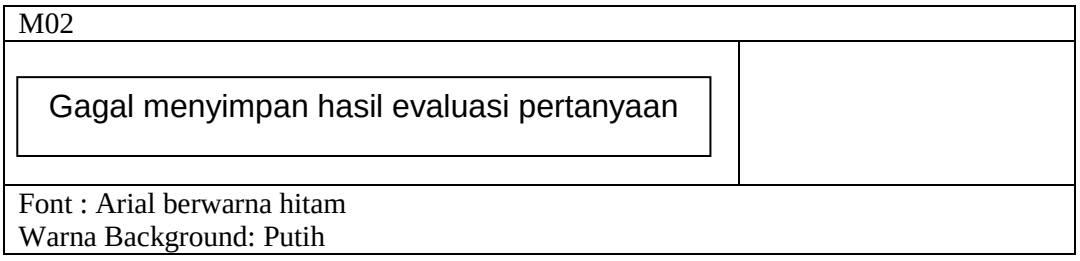
Gambar 3.23 Perancangan Antar Muka Halaman Evaluasi Pertanyaan



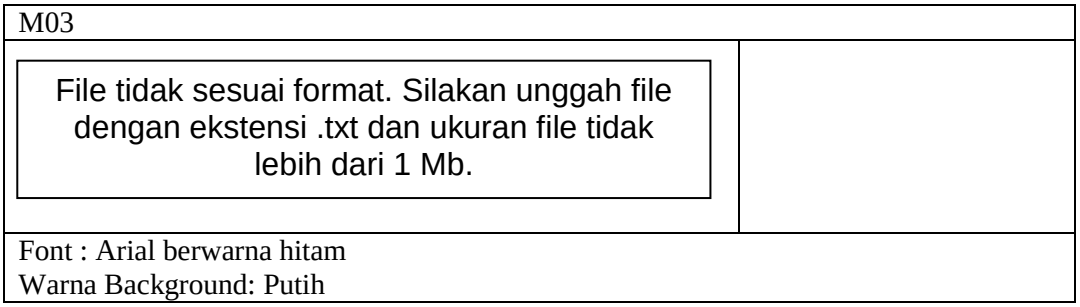
Gambar 3.24 Perancangan Antar Muka Halaman Detail Teks



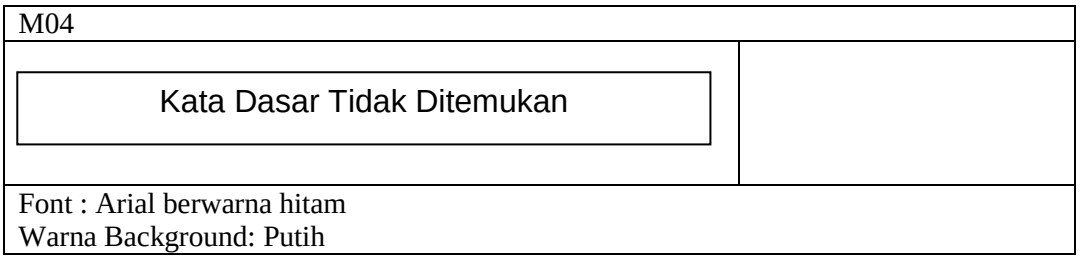
Gambar 3.25 Perancangan Antar Muka Pesan Berhasil Simpan



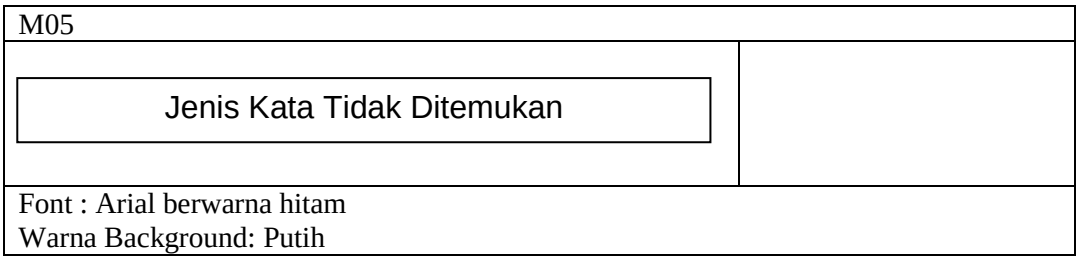
Gambar 3.26 Perancangan Antar Muka Pesan Gagal Simpan



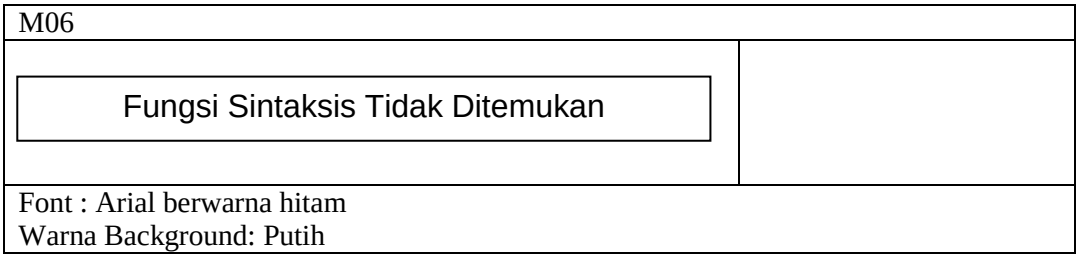
Gambar 3.27 Perancangan Antar Muka Pesan Gagal Unggah File



Gambar 3.28 Perancangan Antar Muka Pesan Kata Dasar Tidak Ditemukan



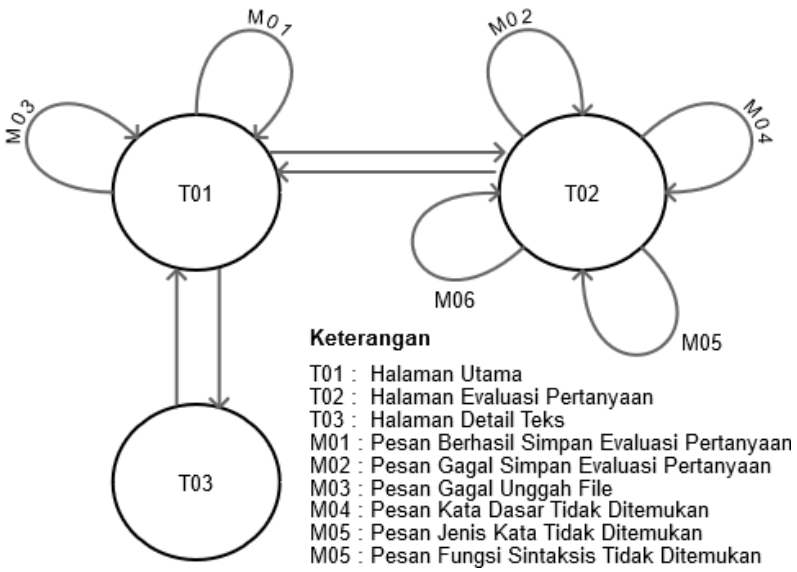
Gambar 3.29 Perancangan Antar Muka Pesan Jenis Kata Tidak Ditemukan



Gambar 3.30 Perancangan Antar Muka Fungsi Sintaksis Tidak Ditemukan

3.11 Jaringan Semantik

Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu halaman ke halaman lainnya. Jaringan semantik dari sistem ini adalah sebagai berikut.



Gambar 3.31 Jaringan Semantik

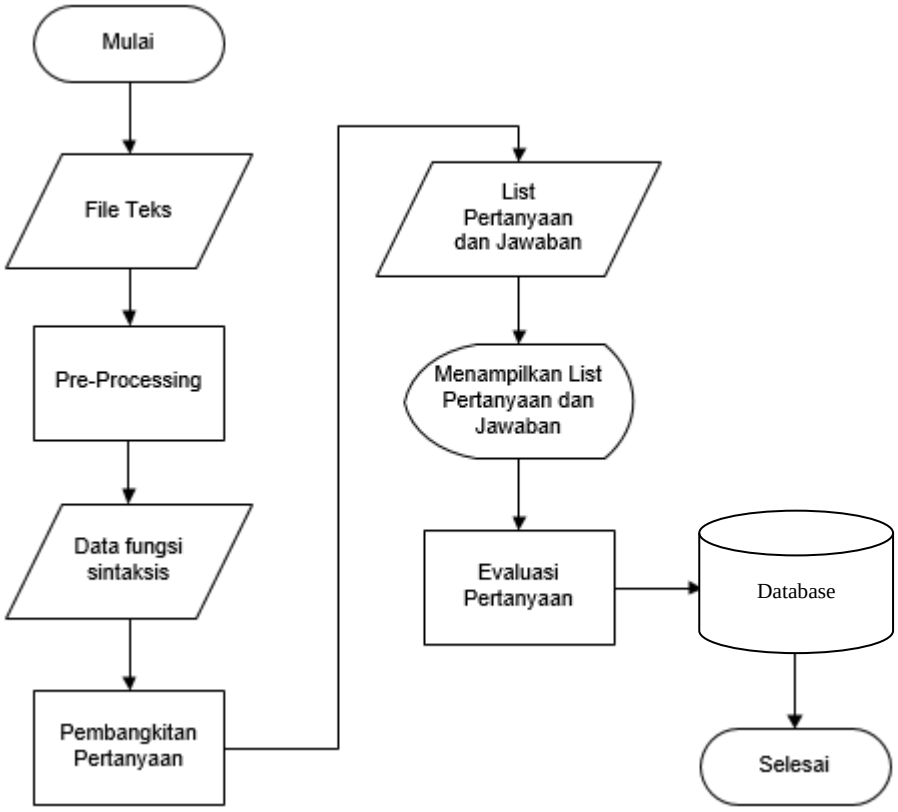
3.12 Perancangan Prosedural

Dalam perancangan prosedural, analisa solusi yang telah dilakukan akan digambarkan dan dijelaskan dari segi algoritmanya. Penggambaran algoritma akan menggunakan diagram *flowchart*. Berikut adalah *flowchart* untuk setiap proses yang ada pada sistem ini.

3.12.1 *Flowchart* Sistem Secara Umum

Flowchart ini menggambarkan proses sistem secara umum. Proses dimulai saat *user* mengunggah file teks sebagai masukan, lalu sistem melakukan proses *pre-processing* dan proses utama yaitu pembangkitan pertanyaan. Setelah pertanyaan dan jawaban berhasil diciptakan oleh sistem dan ditampilkan kepada *user*, maka *user* akan melakukan evaluasi hasil pertanyaan yang telah dibuat oleh sistem. Evaluasi ini dilakukan untuk menentukan akurasi pembangkitan

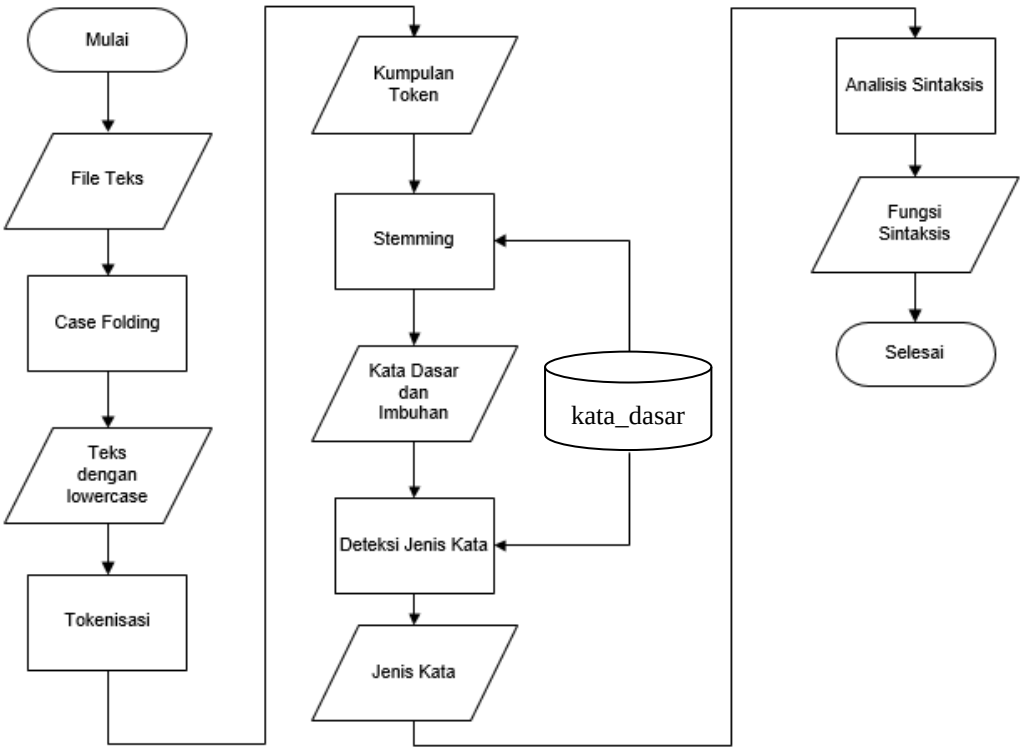
pertanyaan pada penelitian ini. Data-data tersebut akan disimpan ke dalam *database*. Berikut adalah *flowchart* sistem yang menggambarkan proses secara umum.



Gambar 3.32 *Flowchart* Sistem Secara Umum

3.12.2 *Flowchart Pre-Processing*

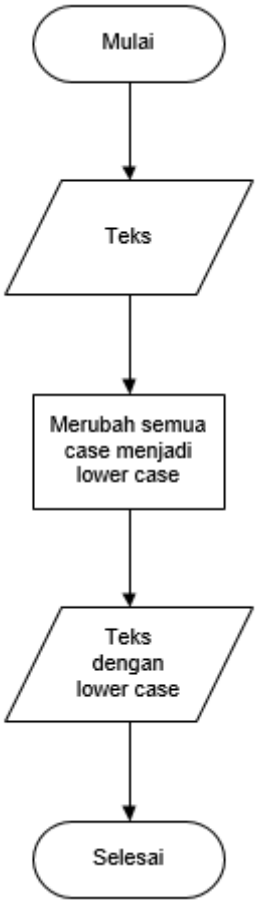
Proses *pre-processing* adalah proses paling awal setelah *user* mengunggah file ke sistem. Proses ini dilakukan untuk mendapatkan data yang dibutuhkan pada proses utama yaitu proses pembangkitan pertanyaan. Seperti yang telah dijelaskan bahwa pada sistem ini terdapat lima tahap proses *pre-processing*, yaitu *case folding*, tokenisasi, *stemming*, deteksi jenis kata, dan analisis sintaksis.



Gambar 3.33 ***Flowchart Pre-Processing***

3.12.2.1 *Flowchart Case Folding*

Case folding adalah proses merubah semua huruf pada teks masukan menjadi *lower case*. Ini dilakukan untuk mencegah kesalahan dalam mencari data pada kamus di basis data. Berikut adalah diagram alur atau *flowchart* pada proses *case folding*.



Gambar 3.34 *Flowchart Case Folding*

3.12.2.2 *Flowchart* Tokenisasi

Tokenisasi adalah proses pencarian token yang dapat berupa kata atau kalimat. Token-token tersebut didapatkan dengan cara memisahkan teks berdasarkan *separator* yang diinginkan. Tokenisasi kalimat menggunakan *separator* berupa titik, tanda tanya, dan tanda seru. Sedangkan tokenisasi kata menggunakan *separator* berupa spasi kosong (*white space*). Berikut adalah diagram alur atau *flowchart* dari tokenisasi kalimat dan tokenisasi kata.



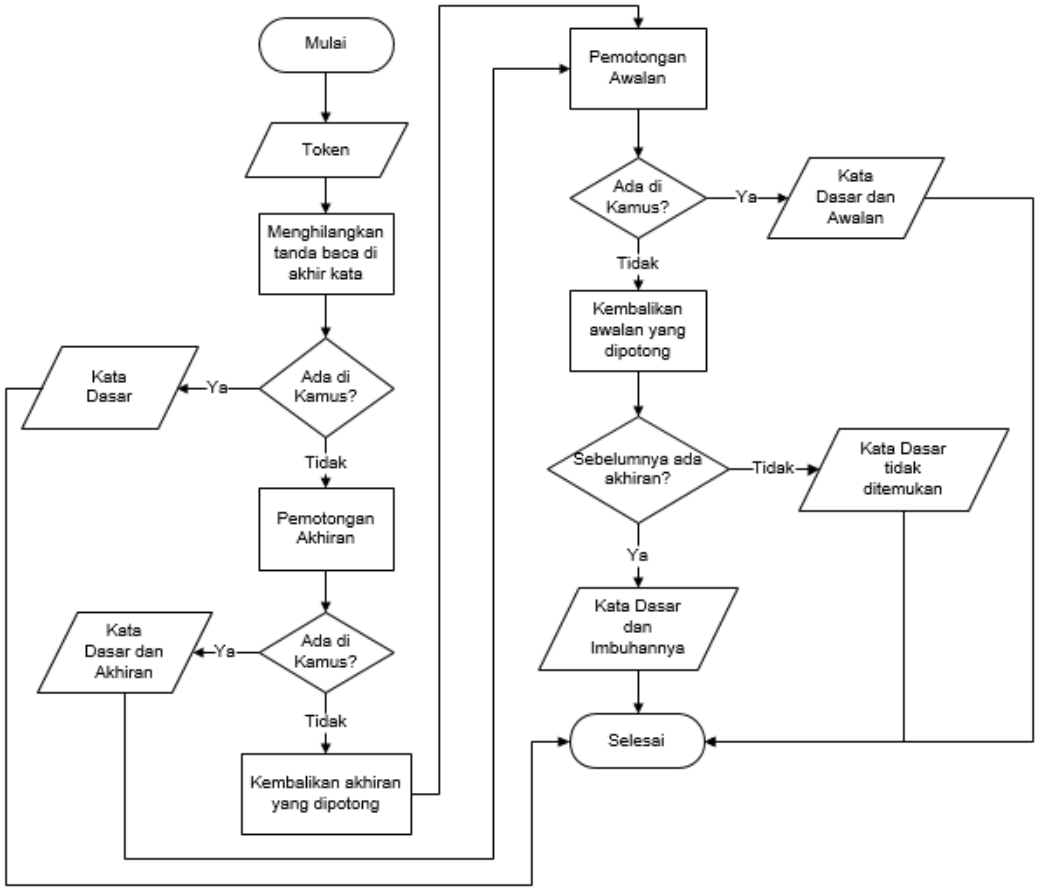
Gambar 3.35 *Flowchart* Tokenisasi Kalimat



Gambar 3.36 *Flowchart* Tokenisasi Kata

3.12.2.3 *Flowchart Stemming*

Proses stemming dilakukan untuk mencari kata dasar dan imbuhan dari suatu token atau kata. *Stemming* dilakukan dengan menggunakan pendekatan algoritma Nazief dan Adriani dimana setiap pemotongan imbuhan, hasil potongan tersebut akan diperiksa di kamus kata dasar yang telah tersedia di basis data. Berikut adalah flowchart dari proses *stemming*.

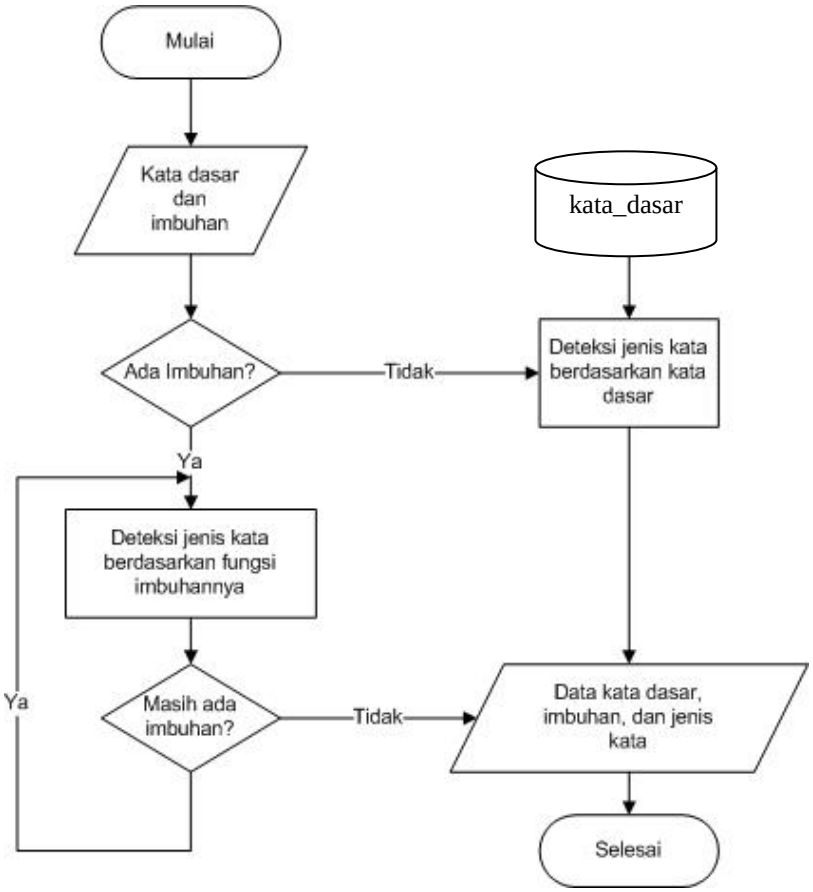


Gambar 3.37 *Flowchart* Proses *Stemming*

Pada *flowchart* di atas terlihat setiap ada proses yang memotong atau menghilangkan sesuatu pasti ada pengecekan ke dalam kamus kata dasar. Jika memang ditemukan ada pada kamus kata dasar sebelum pemotongan imbuhan, maka kata tersebut adalah kata dasar yang tidak memiliki imbuhan. Jika tidak, maka pemotongan imbuhan dilakukan dimulai dengan pemotongan akhiran. Bila ditemukan, maka data yang didapat adalah kata dasar dan akhiran. Pada pemotongan awalan, bila ditemukan kata dasar di kamus kata dasar maka ada data yang didapat adalah kata dasar dan awalan atau imbuhan gabungan. Jika tidak ditemukan kata dasar pada, maka disimpulkan bahwa kata dasar tidak ditemukan pada kamus data yang dimiliki.

3.12.2.4 *Flowchart* Deteksi Jenis Kata

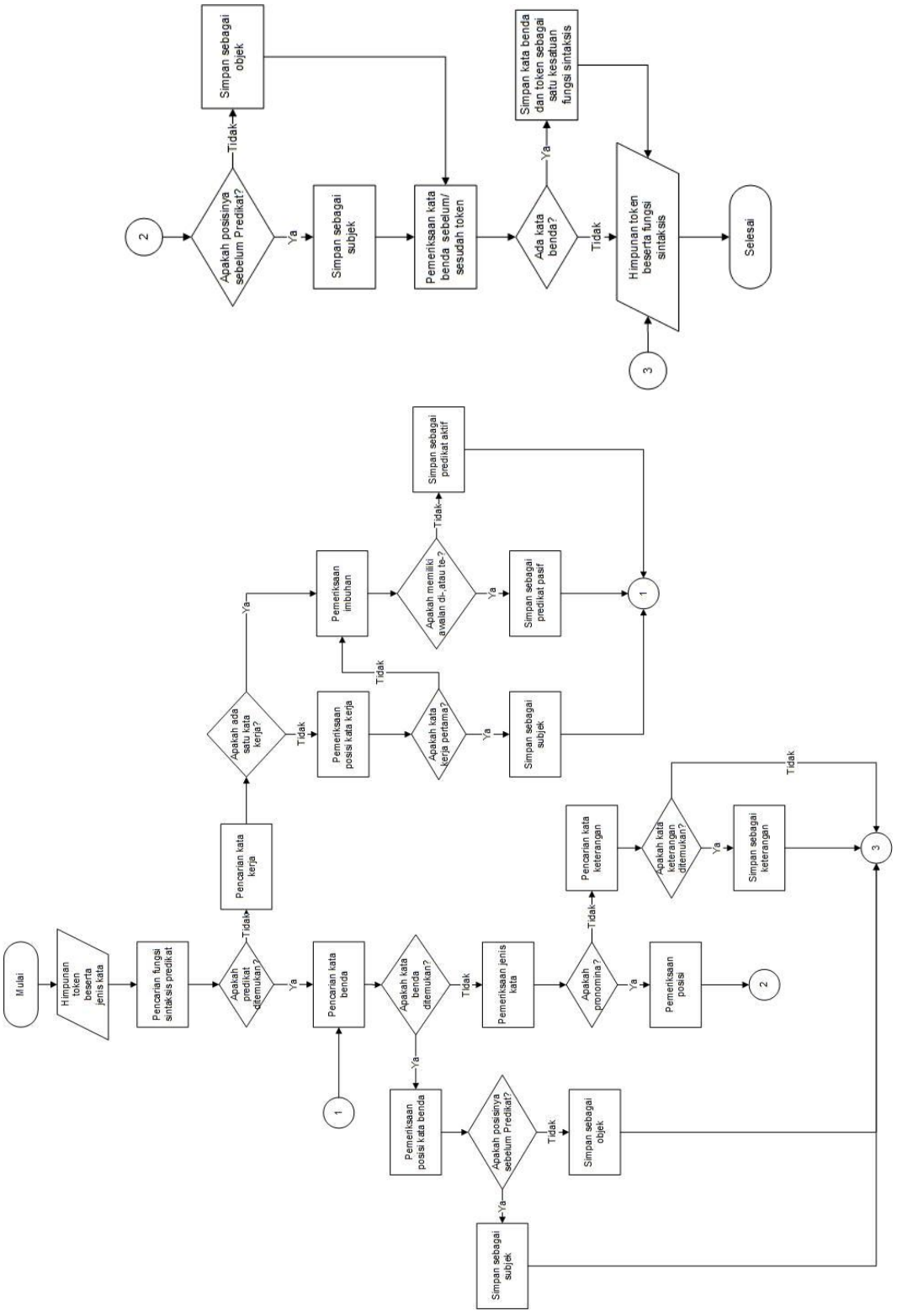
Deteksi jenis kata dalam sistem ini memanfaatkan data yang didapat dari kateglo yang menjadi kamus kata dasar pada basis data sistem ini. Di dalam kamus tersebut terdapat jenis kata untuk tiap kata dasarnya. Tetapi untuk kata yang memiliki imbuhan harus disesuaikan dengan fungsi imbuhannya untuk mendapatkan jenis kata. Berikut adalah flowchart deteksi jenis kata.



Gambar 3.38 *Flowchart* Deteksi Jenis Kata

3.12.2.5 *Flowchart* Analisis Sintaksis

Analisis sintaksis adalah proses dimana mencari fungsi-fungsi sintaksis dari tiap klausa yang ada dalam suatu kalimat. Berikut adalah *flowchart* dari proses analisis sintaksis.

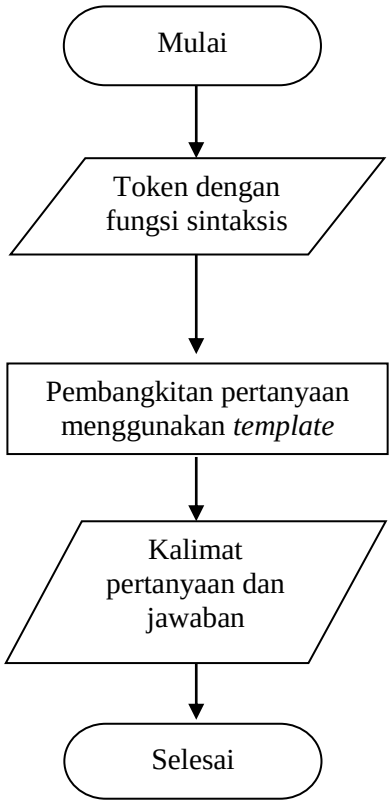


Gambar 3.39 *Flowchart* Analisis Sintaksis

Masukan pada proses ini adalah token yang sudah memiliki jenis kata. Sebelum melakukan pencarian fungsi sintaksis, pemisahan klausa pada kalimat dilakukan dengan mencari kata konjungsi atau kata hubung. Setelah melakukan pemisahan klausa, pencarian fungsi sintaksis tiap klausa dilakukan dengan menggunakan aturan pada tabel 3.3. Dengan menggunakan aturan tersebut, maka token-token yang menjadi masukan akan memiliki fungsi sintaksisnya.

3.12.3 *Flowchart* Pembangkitan Pertanyaan

Fungsi-fungsi sintaksis yang telah ditemukan menjadi informasi penting untuk pembangkitan kalimat pertanyaan. Dengan menggunakan *template* yang telah ditentukan sebelumnya, fungsi-fungsi sintaksis tersebut dapat menjadi kalimat pertanyaan ataupun jawabannya. Berikut adalah *flowchart* dari proses pembangkitan pertanyaan.



Gambar 3.40 Flowhcart Pembangkitan Pertanyaan

Token-token dan fungsi sintaksis yang telah didapatkan dari proses sebelumnya menjadi masukan pada *flowchart* pembangkitan pertanyaan. Proses yang dilakukan pada pembangkitan pertanyaan adalah dengan menggunakan *template* yang ada pada tabel 3.4. Dengan *template* tersebut, maka kalimat pertanyaan dan jawaban akan dapat dibangkitkan.