BAB 2
LANDASAN TEORI

2.1 Tata Bahasa

Pembangkitan pertanyaan yang dibangun harus mengikuti tata bahasa baku Bahasa Indonesia. Tata bahasa baku dapat digunakan sebagai acuan untuk membentuk kalimat pertanyaan dengan mengetahui jenis kata, bentuk kalimat, dan struktur kalimat.

2.1.1. Jenis Kata

Dilihat dari konsep makna yang dimiliki dan atau peran yang harus dilakukan, kata-kata dibedakan menjadi 16 (enam belas) jenis [1]. Tetapi jenis kata yang digunakan dalam penelitian ini hanya sembilan jenis, seperti yang telah diperlihatkan pada tabel 2.1 di bawah ini.

Tabel 2.1 Jenis Kata

NO	JENIS KATA	DEFINISI	CONTOH
1	Kata Benda	1. Kata dasar yang dapat diikuti frase **'yang ...'** 2. Kata berimbuhan yang berawalan pe-, berakhiran **-an**, **-nya**, berimbuhan gabung **pe-an**, **per-an**, **ke-an**	1. wanita 2. persekutuan 3. pertandingan

NO	JENIS KATA	DEFINISI	CONTOH
2	Kata Ganti	1. Kata benda yang menyatakan orang	1. **Kata Ganti Orang Pertama** : Saya, Aku, Kami, Kita 2. **Kata Ganti Orang Kedua** : Kamu, Engkau, Anda, Kalian 3. **Kata Ganti Orang Ketiga** : Dia, Nya, Beliau, Mereka
3	Kata Kerja	1. Kata dasar atau kata berimbuhan yang dapat diikuti frase **'dengan ...'** 2. Kata berimbuhan yang berawalan **me-**, **ber-**, **di-**, **ter-**, **per-**, dan berakhiran **–kan**, **-i**.	1. Pergi 2. Berjalan 3. Menulis 4. Terbaca 5. Dilihat 6. Persingkat 7. Damaikan 8. Datangi
4	Kata Sifat	1. Kata yang dapat diikuti dengan kata keterangan 'sekali' serta dapat dibentuk menjadi kata ulang berimbuhan gabung **se-nya** 2. Berada di belakang kata benda yang disifatinya 3. Pada tingkat klausa, dapat menduduki fungsi predikat	1. Rumah **besar** 2. Pemandangan **indah** 3. Anak itu **nakal**
5	Kata Bilangan	1. Kata yang menyatakan jumlah, nomor, urutan, atau himpunan	1. Satu, dua, tiga, ... 2. Pertama, kedua, ketiga, ... 3. 1, 2, 3, ... 4. I, II, III, ...
6	Kata Penghubung	1. Kata yang digunakan untuk menghubungkan kata dengan kata, klausa dengan klausa, atau kalimat dengan kalimat.	1. Dan, dengan, serta, ... 2. Karena, sebab 3. Ketika, sewaktu, sebelum, sesudah, ... 4. Sampai, hingga, ... 5. Seperti, laksana, ...

NO	JENIS KATA	DEFINISI	CONTOH
7	Kata Keterangan	1. Kata yang digunakan untuk memberi penjelasan pada kalimat atau bagian kalimat lain, yang sifatnya tidak menerangkan keadaan atau sifat.	1. Memang, pasti, tentu 2. Barangkali, mungkin, ... 3. Semoga, moga-moga, ... 4. Seringkali, sesekali, ... 5. Sudah, telah, sedang, ... 6. Ingin, mau, hendak, suka, ... 7. Boleh, wajib, harus, ...
8	Kata Tanya	1. Kata yang digunakan sebagai pembantu di dalam kalimat yang menyatakan pertanyaan.	1. Apa 2. Siapa 3. Mengapa 4. Bagaimana 5. Kapan 6. Mana
9	Kata Partikel	1. Morfem yang digunakan untuk menegaskan.	1. Kah, tah, lah, pun, per

2.1.2. Kata Tanya

Untuk menciptakan kalimat tanya, perlu pendalaman lebih lanjut untuk mengetahui aturan-aturan kata tanya yang akan dipakai pada pembangkitan pertanyaan. Kata tanya yang digunakan adalah **Apa**, **Siapa**, **Mana**, **Kapan**, **Mengapa**, dan **Bagaimana**.

2.1.2.1. Kata Tanya Apa

Kata tanya Apa digunakan dengan aturan :

1. Untuk menanyakan benda atau hal, baik tentang jenisnya maupun namanya, digunakan di depan kata benda atau kata penggantinya

   Contoh :

   a. Apa isi lemari ini?
   b. Apa nama benda ini?

2. Untuk menanyakan nama atau nama jenis barang digunakan di belakang kata benda atau frase benda yang ditanyakan. Dalam hal ini lazim diberi partikel KAH.

   Contoh :

a. Binatang *apa* yang berbulu tebal itu?
b. Buku *apakah* yang kau pegang itu?

3. Untuk menyatakan benda digunakan pada awal kalimat dalam kalimat pasif yang memiliki kata penghubung YANG. Dalam hal ini lazim diberi partikel KAH. Contoh :
   a. *Apa* yang diminta anak itu?
   b. *Apa* yang digunakan untuk menulis?
4. Untuk meminta pengakuan *ya* atau *tidak/bukan* digunakan :
   a. Di muka kalimat verbal, dalam hal ini lazim diberi partikel KAH. Contoh : *Apakah* kamu mencuri?
   b. Di muka kalimat nominal. Di sini pun lazim diberi partikel KAH. Contoh : *Apakah* orang itu gurumu?

2.1.2.2. Kata Tanya Siapa

Kata tanya Siapa digunakan dengan aturan :

1. Untuk menanyakan nama yang digunakan di depan kata *nama* yang diikuti dengan kata benda yang menyatakan orang atau kata gantinya. Contoh :
   a. *Siapa* nama anak itu?
   b. *Siapa* nama guru itu?
2. Untuk menanyakan identitas orang digunakan :
   a. Di depan kata benda yang menyatakan orang (bisanya berawalan pe-). Contoh : *Siapa* pengarang lagu ini?
   b. Di depan frase dengan kata penghubung YANG. Contoh : *Siapa* yang memecahkan piring?

2.1.2.3. Kata Tanya Mana

Kata tanya Mana dengan fungsi menanyakan ‘tempat keberadaan’ digunakan di depan kata benda. Contoh :

1. *Mana* buku telepon itu?
2. *Mana* temanmu?
3. *Mana* rumahmu?
4. *Mana* buku itu kau dapat?

Untuk menanyakan tempat keberadaan dengan lebih pasti, di depan kata Mana perlu ditempatkan kata depan Di, Ke, atau Dari.

1. *Di mana* mereka belajar?
2. *Ke mana* perginya maling itu?
3. *Dari mana* uang itu kau dapatkan?

2.1.2.4. Kata Tanya Kapan

Kata tanya Kapan dengan fungsi untuk menanyakan waktu digunakan di depan kalimat berpredikat kata kerja. Contoh :

1. *Kapan* paman akan datang?
2. *Kapan* acara wisudamu akan dimulai?
3. *Kapan* peristiwa itu terjadi?

2.1.2.5. Kata Tanya Mengapa

Kata tanya Mengapa dengan fungsi untuk menyatakan sebab atau alasan digunakan di muka kalimat berpredikat kata kerja atau kata sifat. Contoh :

1. *Mengapa* kamu tidak datang kemarin?
2. *Mengapa* anak itu menangis?

2.1.2.6. Kata Tanya Bagaimana

Kata tanya Bagaimana digunakan dengan aturan :

1. Untuk menanyakan keadaan digunakan di depan kata benda.
   Contoh :
   a. *Bagaimana* cuaca di sana?
   b. *Bagaimana* harganya?

2. Untuk menanyakan cara atau proses digunakan di depan kata kerja. Secara bebas boleh disertai kata **Cara** di antara kata tanya **Bagaimana** dengan kata kerjanya itu.
   a. *Bagaimana* cara merebus ketupat?
   b. *Bagaimana* cara mendidik anak seperti dia?

3. Untuk menanyakan apa yang harus dilakukan oleh atau terhadap sesuatu digunakan di depan kata benda. Dalam hal ini diantara kata tanya **Bagaimana** itu dengan kata bendanya perlu disisipkan kata depan **Dengan**.
    a. Kalau kita dapat rumah dinas *bagaimana* dengan rumah ini?
    b. Kalau daerah ini akan dijadikan waduk, *bagaimana* dengan penduduk di sini?

2.1.3. Penggunaan Imbuhan

Imbuhan dalam suatu kata dapat merubah fungsi, jenis, dan makna pada kata tersebut. Dengan mengetahui perubahan kata yang disebabkan oleh imbuhan, akan membantu mencari pembentukan kalimat tanya yang lebih akurat dan relevan dengan teks. Imbuhan pada Bahasa Indonesia terdiri dari **Akhiran**, **Awalan**, dan **Imbuhan Gabung**. Akhiran, awalan, dan imbuhan gabung yang dipakai pada penelitian ini dijelaskan pada tabel 2.2, 2.3, dan 2.4.

Tabel 2.2 Akhiran

NO	IMBUHAN	FUNGSI	CONTOH
1	**–kan**	Membentuk Kata Kerja Transitif	Tenangkan dulu anak-anak ini.
2	**–i**	Membentuk Kata Kerja Transitif	Lengkapi dulu syaratnya
3	**–an**	Membentuk Kata Benda	Tulisan adikku
4	**–nya**	Kata Ganti orang ketiga tunggal (objek/pemilik)	1. Bukunya sudah hilang. 2. Saya minta tolong kepadanya

NO	IMBUHAN	FUNGSI	CONTOH
		Sebagai Akhiran : 1. Membentuk kata benda 2. Memberi penekanan / penegasan 3. Membentuk kata keterangan	1. Tenggelamnya kapal tampomas. 2. Saya ingin mandi, airnya tidak ada. 3. Agaknya dia akan datang terlambat

Tabel 2.3 Awalan

NO	IMBUHAN	FUNGSI	CONTOH
1	**ber-**	(Ber,Be,Bel) Membentuk kata Kerja Intransitif	1. Berlibur 2. beragam 3. belajar
2	**per-**	(Per,Pe,Pel) Membentuk Kata Kerja Perintah	1. Percepat 2. Peringan 3. Pelajar
3	**me-**	(me, mem, men,meny, meng, menge) Membentuk kata kerja aktif transitif dan intranssitif.	Dia menyanyi didepanku.
4	**di-**	Membentuk kata kerja pasif	Dia ditangkap polisi
5	**ter-**	1. Membentuk kata kerja pasif yang menyatakan keadaan. 2. Membentuk kata benda yang menyakan orang	1. Kelas pelajarana ini adalah yang terbaik menurut saya. 2. Pihak keluarga memukuli tersangka pembunuhan.

NO	IMBUHAN	FUNGSI	CONTOH
6	**ke-**	1. Membentuk kata bilangan yang menyatakan : tingkat atau kedudukan dalam suatu deretan, kumpulan atau himpunan. 2. Membentuk kata kerja pasif dg arti “tidak disengaja” 3. Membentuk kata benda dg arti “orang atau sesuatu yang di..”	1. Saya duduk dikursi ketiga. 2. Kedua orang itu teman saya. 3. Ayah ketipu 1 juta.
7	**se-**	Membentuk kata keterangan	Warga sekampung berdemo
8	**pe-**	(pe, pem, pen, peny, peng, penge) Membentuk kata benda	Siapa penulis buku ini ?

Tabel 2.4 Imbuhan Gabung.

NO	IMBUHAN	FUNGSI	CONTOH
1	**ber-kan**	Membentuk kata kerja intransitif yang dilengkapi dengan sebuah pelengkap sedangkan makna yang didapat sebagai hasil pengimbuhan itu adalah menyatakan menjadikan yang disebut pelengkapnya sebagai yang disebut kata dasarnya	1. Hanya bersenjatakan bambu runcing 2. Bersenjatakan Artinya menjadikan bambu runcing sebagai senjata

NO	IMBUHAN	FUNGSI	CONTOH
2	**ber-an**	Membentuk kata kerja intransitif	Mereka berlarian kesana kemari
3	p**er-kan**	Membentuk kata kerja	Jangan perdulikan diri ini.
4	p**er-i**	Membentuk Kata Kerja	Mereka akan perenglengkapi masalah pertanian.
5	**me-kan**	Membentuk kata kerja aktif transitif	Pemerintah melebarkan jalan.
6	**me-i**	Membentuk kata kerja transitif aktif	Bulan menerangi bumi Menerangi artinya membuat jadi terang.
7	**memper-**	Membentuk kata kerja aktif transitif	Jalan tol itu memperlancar arus lalu lintas.
8	**memper-kan**	Membentuk kata kerja aktif transitif	Mereka masih saja memberdebatkan masalah itu.
9	**memper-i**	Membentuk kata kerja aktif transitif	Saya akan memperbaiki dulu rumah itu.
10	**di-kan**	Membentuk kata kerja pasif.	Berita itu disiarkan oleh RRI.
11	**di-i**	Membentuk kata kerja pasif	Pembangunannya diawasi oleh camat.
12	**diper-**	Membentuk kata kerja pasif.	Jalan ini diperlebar oleh pemerintah.
13	**diper-kan**	Membentuk kata kerja pasif	Mudah-mudahan engkau dipertemukan kembali.

NO	IMBUHAN	FUNGSI	CONTOH
14	**diper-i**	Membentuk kata kerja pasif	Kami dilengkapi pemerintah dengan senjata.
15	**ter-kan**	Membentuk kata kerja pasif Keadaan	Akhirnya pekerjaan itu terselesaikan juga.
16	**ter-i**	Membentuk kata kerja pasif intransitif	Permintaan kami akhirnya terpenuhi juga.
17	**ke-an**	Membentuk kata benda abstrak dan sejumlah kata kerja	1. Kedatangan beliau disambut meriah 2. Ibu pergi ke kelurahan
18	**se-nya**	1. Membentuk kata penghubung 2. Membentuk kata keterangan	1. Sebenarnya saya tidak akan hadir diacara itu. 2. Setibanya dirumah saya akan makan karedok.
19	**pe-an**	(pe-an, pem-an, pen-an,peny-an,peng-an,penge-an) Membentuk kata benda	Ayah bekerja dipelelangan ikan.
20	**per-an**	(Per-an,pe-an,pel-an) Membentuk kata benda	Usaha perhotelan di indramayu cukup baik.

2.1.4. Klausa

Klausa merupakan satuan sintaksis yang berada di atas satuan frase dan di bawah satuan kalimat, berupa runtutan kata-kata berkonstruksi predikatif. Artinya, di dalam satuan atau konstruksi itu terdapat sebuah predikat, bila dalam satuan tidak terdapat predikat, maka satuan itu bukan sebuah klausa. Selain fungsi predikat yang harus ada di klausa, fungsi subjek dapat dikatakan wajib ada, sedangkan fungsi lainnya opsional [5].

Klausa yang memiliki fungsi-fungsi sintaksis yang lengkap, yaitu terdapat subjek dan predikat, maka klausa tersebut disebut klausa bebas atau klausa utuh. Sedangkan klausa yang fungsinya tidak lengkap disebut klausa terikat. Contoh pada kalimat “Saya akan datang kalau diundang.”. Pada kalimat tersebut terdapat dua buah klausa, yaitu :

1. Saya akan datang
2. kalau diundang

Klausa nomor satu memiliki fungsi-fungsi sintaksis yang lengkap dan dapat berdiri sendiri, sehingga klausa tersebut adalah klausa bebas. Sedangkan klausa nomor dua tidak memiliki fungsi sintaksis yang lengkap, yaitu tidak memiliki fungsi subjek. Sehingga klausa nomor dua terikat dengan klausa nomor satu.

Klausa bebas di dalam sebuah kalimat majemuk subordinatif disebut klausa atasan, sedangkan klausa terikat disebut klausa bawahan [5].

2.1.5. Pembentukan Kalimat

Kalimat adalah satuan bahasa yang berisi suatu pikiran atau amanat yang lengkap. Lengkap berarti dalam kalimat tersebut terdapat unsur subjek, predikat, objek, dan keterangan [1]. Menurut strukturnya, sebuah kalimat sederhana dalam bahasa Indonesia memiliki pola sebagai berikut

Tabel 2.5 Kalimat Sederhana Berdasarkan Struktur

NO	STRUKTUR	CONTOH
1	Subjek + Predikat	1. Ibuku memasak. 2. Kakakku seorang pelajar. 3. Temanku itu baik sekali.
2	Subjek + Predikat + Objek	1. Ayah bermain bola. 2. Ibu menyapu lantai. 3. Aku makan nasi goreng.
3	Subjek + Predikat + Objek + Keterangan	1. Ayah mengantar adik ke sekolah. 2. Aku makan nasi goreng di pagi hari 3. Adik bermain bola di taman.

NO	STRUKTUR	CONTOH
4	Subjek + Predikat + Objek + Objek	1. Ayah membelikan saya buku. 2. Nuri membacakan adiknya buku cerita. 3. Ibu membukakan saya pintu.

2.2 *Pre-Processing*

Pre-processing adalah proses yang dilakukan sebelum memasuki proses utama dalam sistem. Pada sistem pembangkitan pertanyaan ini, hasil akhir dari *pre-processing* adalah ditemukannya fungsi-fungsi sintaksis dalam suatu kalimat dan kata dasar beserta imbuhannya dari setiap kata yang ada pada kalimat tersebut. Tahapan *pre-processing* pada sistem ini adalah *case folding*, tokenisasi, *stemming*, dan deteksi jenis kata.

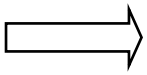
2.2.1. *Case Folding*

Case folding adalah proses merubah semua huruf dalam teks menjadi huruf kapital atau non-kapital[16]. Fungsi *case folding* dalam sistem ini adalah mencegah terjadinya kata yang tidak terdeteksi dalam kamus yang disimpan dalam basis data karena kata-kata yang ada di basis data semua memiliki huruf non-kapital.

2.2.2. Tokenisasi

Proses pemahaman bahasa/kalimat diawali dengan tokenisasi (pemotongan) suatu kalimat menjadi unsur-unsurnya yaitu kata [2]. Berdasarkan hal tersebut, tokenisasi dilakukan dengan tujuan untuk mengolah kalimat dimulai dari unsur kalimatnya, yaitu kata. Dengan diketahui kata dari kalimat, maka dapat diketahui jenis kata yang akan membantu dalam pembentukan struktur kalimat Bahasa Indonesia (SPOK).

Budi adalah seorang pelajar



No	Hasil Tokenisasi
1	Budi
2	adalah
3	seorang
4	pelajar

Gambar 2.1 Tokenisasi

Gambar 2.1 memperlihatkan tokenisasi pada suatu teks "Budi adalah seorang pelajar". Hasil tokenisasi pada teks tersebut adalah mendapat empat token, yaitu Budi, adalah, seorang, dan pelajar.

2.2.3. *Stemming*

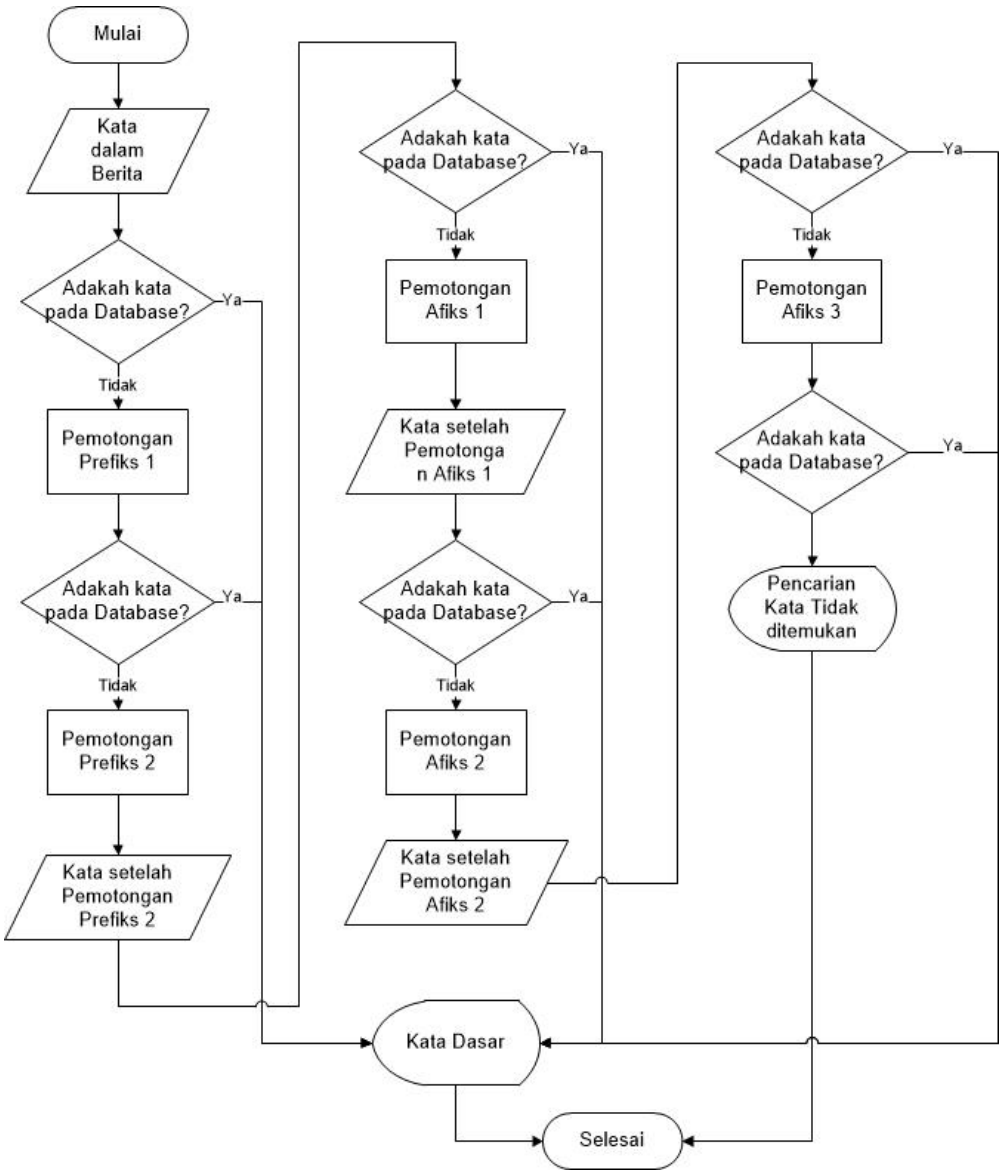
Stemming merupakan suatu proses yang terdapat dalam sistem *Information Retrieval* (IR) yang mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata akarnya dengan menggunakan aturan-aturan tertentu[3]. Stemming memiliki tujuan untuk meningkatkan performa IR dengan mentransformasikan kata-kata dalam suatu teks menjadi kata dasarnya. Algoritma Stemming lebih banyak digunakan untuk teks Bahasa Inggris yang memiliki struktur imbuhan lebih mudah untuk diolah, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan teks Bahasa Indonesia memiliki struktur imbuhan yang lebih rumit dan kompleks, yaitu perlu melakukan proses menghilangkan sufiks, prefiks, dan konfiks.

Algoritma stemming untuk teks berbahasa Indonesia, diantaranya: Algoritma Porter, Algoritma Nazief dan Adriani. Berdasarkan hasil penelitian [3], kesimpulan dari perbandingan antara Algoritma Porter dengan Algoritma Nazief dan Adriani, adalah :

1. Algoritma Porter membutuhkan waktu yang lebih singkat dibandingkan Algoritma Nazief & Adriani.
2. Algoritma Porter memiliki persentase keakuratan lebih kecil dibandingkan Algoritma Nazief & Adriani.

3. Kamus sangat berpengaruh dalam menggunakan Algoritma Nazief & Adriani. Semakin lengkap kamus yang digunakan, maka semakin akurat hasil yang didapat.

Algoritma Nazief dan Adriani menggunakan basis data yang menyimpan kamus kata dasar untuk pengecekan kata setiap pemotongan imbuhan dilakukan. Berikut adalah alur dari algoritma Nazief dan Adriani.



Gambar 2.2 Algoritma Nazief dan Adriani

Pada gambar 2.2 digambarkan alur dari algoritma Nazief dan Adriani. Terlihat jelas pada gambar tersebut bahwa setiap ada pemotongan yang dilakukan baik itu awalan (prefiks) atau akhiran (afiks) akan selalu ada pemeriksaan kata dalam basis data.

2.2.4. *POS Tagging*

Part of Speech Tagging (POS-Tagging) adalah suatu proses yang memberikan label kelas kata secara otomatis pada suatu kata dalam kalimat[6]. Hasil dari POS ini sangat berpengaruh terhadap keluaran dari proses *Parsing*[7].

Sudah ada banyak penelitian yang dilakukan terkait dengan POS Tagging untuk teks berbahasa Indonesia. Metode-metode yang dipakai dalam penelitian-penelitian tersebut pun memiliki akurasi yang cukup tinggi, diantaranya Brill *tagger* yang memiliki akurasi 88%[8], *Conditional Random Field (CRF)* dan *Maximum Entropy Method* memiliki akurasi 97,57% [9] dan *Hidden Markov Model* yang memiliki akurasi 96,5% [8].

Pada penelitian ini tidak menggunakan POS Tagging secara otomatis karena menggunakan data yang didapatkan dari Kateglo yang telah memiliki jenis kata pada setiap kata dasarnya. Sedangkan untuk jenis kata yang memiliki imbuhan, menggunakan aturan perubahan jenis kata berdasarkan imbuhan.

2.3 Proses Utama

Proses utama adalah proses yang dilakukan untuk mengolah data awal yang telah didapatkan pada *pre-processing* untuk mendapatkan hasil akhir berupa pertanyaan dan jawaban yang relevan dengan teks masukan. Proses utama yang dilakukan adalah analisis sintaksis dan pembangkitan pertanyaan.

2.3.1. Analisis Sintaksis

Sintaksis adalah cabang linguistik yang bidang kajiannya meliputi satuan lingual berwujud kata, frasa, klausa, kalimat, hingga wacana. Satuan tersebut disebut satuan sintaksis. Istilah sintaksis berasal dari bahasa Yunani, yaitu *syntaxis* yang berarti susunan atau tersusun secara bersama. Sintaksis berusaha menjelaskan hubungan fungsional antara unsur-unsur dalam satuan sintaksis yang

tersusun bersama dalam wujud frasa, klausa, kalimat, dan wacana. Hubungan fungsional berarti hubungan saling ketergantungan antara unsur yang satu dengan unsur yang lain [17].

Analisis sintaksis dilakukan dengan menganalisa kalimat beserta klausa yang ada pada kalimat tersebut. Analisa ini akan menghasilkan fungsi-fungsi sintaksis yang dapat membantu untuk membentuk pertanyaan pada proses utama dalam sistem. Fungsi sintaksis meliputi subjek, predikat, objek, pelengkap, dan keterangan [17].

2.3.2. Pembangkitan Pertanyaan (*Question Generation*)

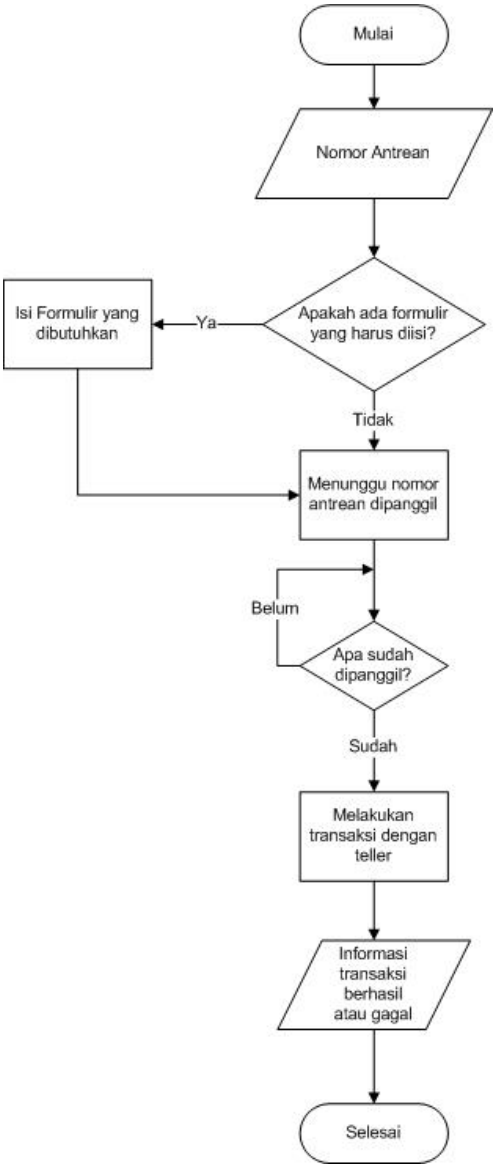
Pembangkitan pertanyaan atau *Question Generation* adalah sebuah teknik untuk membangkitkan pertanyaan yang berasal dari sebuah kalimat atau teks dalam bentuk bahasa alami. Garis besar konseptual *question generation* terdiri dari tiga kategori yaitu berbasis sintaks, berbasis semantik, dan berbasis template. Sistem *question generation* dalam kategori sintaksis sering menggunakan unsur semantik dan sebaliknya. Sedangkan sistem berbasis template mengunakan beberapa tingkat sintaksis dan/atau informasi semantik. [10]

Banyak penelitian yang mengusulkan algoritma dan pendekatan baru yang aktif menggunakan kalimat dalam dokumen sebagai sumber pertanyaan dan sekaligus memberikan sebuah jawaban. Salah satu pendekatan yang dilakukan peneliti terdahulu adalah bagaimana pengolahan bahasa dengan menggunakan teknik yang secara otomatis untuk menghasilkan pertanyaan-pertanyaan. Di antaranya adalah *syntactic transformations* untuk kasus input berupa *informational text* dan dengan menggunakan metode statistik [11][12]. Implementasi dalam teks bahasa Indonesia pun sudah dilakukan, diantaranya menggunakan prosesor semantik, secara khusus diterapkan dalam bidang kesehatan, dan berbasis template [13][14][15].

2.4 *Flowchart*

Flowchart atau Bagan Alir Program merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program [23]. *Flowchart* digunakan

untuk menggambarkan proses logika dan algoritma dari sistem pembangkitan pertanyaan. Simbol-simbol yang digunakan telah dijelaskan pada daftar simbol, berikut contoh *flowchart* pada kehidupan sehari-sehari, yaitu saat ingin melakukan transaksi di *teller* suatu bank.



Gambar 2.3 Contoh *Flowchart*

Pada gambar 2.3, nomor antrean menjadi data awal yang didapatkan oleh nasabah saat ingin melakukan transaksi dengan *teller* di bank. Setelah

mendapatkan nomor antrean, nasabah diharuskan mengisi suatu formulir bila diperlukan, contohnya formulir penyetoran atau formulir penarikan. Jika tidak perlu mengisi formulir, maka nasabah langsung menunggu nomor antreannya dipanggil oleh *teller*. Jika sudah dipanggil, maka nasabah mulai melakukan transaksi dengan *teller* sampai selesai. Hasil transaksi tersebut akan diberikan oleh *teller* kepada nasabah baik itu berhasil ataupun tidak.

2.5 *Unified Modelling Language* (UML)

Unified Modeling Language (UML) adalah adalah keluarga notasi grafis yang membantu dalam menggambarkan dan merancang sistem perangkat lunak terutama yang dibangun dengam menggunakan orientasi objek. Definisi ini adalah definisi yang telah disederhanakan. Pada kenyataannya, pendapat orang-orang berbeda tentang UML berbeda satu sama lain. Hal ini disebabkan oleh sejarah dan perspektif yang berbeda tentang membuat proses rekayasa perangkat lunak yang efektif [4].

Diagram-diagram UML yang digunakan pada sistem ini adalah *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

2.5.1. *Use Case Diagram*

Use case adalah teknik untuk menggambarkan kebutuhan fungsional dari sebuah sistem. *Use case* menggambarkan interaksi antara pengguna dengan sistem, menyediakan sebuah cerita bagaimana sebuah sistem digunakan [4]. Dalam bahasan *use case*, para pengguna disebut sebagai aktor. Aktor merupakan sebuah peran yang dimainkan seorang pengguna dalam kaitannya dengan sistem. Aktor tidak harus manusia, jika sebuah sistem melakukan sebuah layanan untuk sistem komputer lain, sistem lain tersebut merupakan aktor.

2.5.2. *Activity Diagram*

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, activity diagram memainkan

peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity* diagram mendukung behavior paralel [4]. Node pada sebuah activity diagram disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah activity yang tersusun dari action.

2.5.3. *Class Diagram*

Class diagram menggambarkan jenis-jenis objek yang terdapat pada sistem dan berbagai macam relasi statis yang ada diantara objek-objek tersebut. *Class diagram* juga menunjukkan properti-properti dan operasi-operasi pada suatu *class* dan batasan atau *constraint* yang berlaku pada cara objek-objek tersebut saling berhubungan [4].

Multiplicity dari suatu properti adalah indikasi seberapa banyak objek yang dapat mengisi properti tersebut. Secara umum, *multiplicity* didefinisikan dengan batas bawah dan batas atas. Batas bawah berupa angka positif dan nol, sedangkan batas atas berupa angka positif dan * (tidak terbatas) [4]. Berikut adalah tabel *multiplicity* yang terdapat pada *class diagram*.

Tabel 2.6 *Multiplicity*

Multiplicity	**Arti**
0..0	Nol
0..1	Nol atau Satu
0..*	Nol atau Banyak
1..1	Tepat satu
1. .*	Satu atau banyak

2.5.4. *Sequence Diagram*

Interaction diagrams menggambarkan bagaimana kumpulan objek dapat berkolaborasi dalam tindakan tertentu. UML mendefinisikan beberapa bentuk *interaction diagram*, salah satunya yang paling umum adalah *sequence diagram*. Secara khusus, *sequence diagram* menangkap perilaku dari suatu skenario.

Diagram ini menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek tersebut dalam suatu *use case* [4].

2.6 Bahasa Pemrograman

Bahasa pemrograman adalah sekumpulan aturan yang disusun sedemikian rupa sehingga pengguna komputer dapat membuat program komputer berdasarkan aturan tersebut [18]. Ada banyak macam bahasa pemrograman, diantaranya *Java*, PHP, *Phyton*, C, C++, dan lain lain. Dalam sistem pembangkitan pertanyaan ini, bahasa pemrograman yang dipakai adalah *Java*, *Groovy*, dan *MySQL*.

2.6.1. *Java*

Java merupakan bahasa pemrograman yang disusun oleh James Gosling yang dibantu oleh rekan-rekannya di suatu perusahaan perangkat lunak yang bernama *Sun Microsystems*, pada tahun 1991. Bahasa pemrograman ini mula-mula diinisialisasi dengan nama *Oak*, namun pada tahun 1995 diganti namanya menjadi *Java*. *Java* adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri ataupun pada lingkungan jaringan [21].

2.6.2. *Groovy*

Groovy adalah bahasa dinamis baru yang dapat diinterpretasi atau dikompilasi dan didesain secara spesifik untuk *Java platform*. *Groovy* telah terpengaruh oleh bahasa-bahasa seperti *Ruby*, *Phyton*, *Perl*, dan *Smalltalk*, dan tentu saja *Java* [22].

Tidak seperti bahasa lain yang porting ke JVM (*Java Virtual Machine*), *Groovy* didesain dengan JVM di dalamnya. Sebagai contoh, *Groovy* bergantung pada *Java* API daripada memberi suplai kepada API-nya sendiri, sehingga *developer* tidak perlu memutuskan antara IO *package* dari *Java* dan IO *method* dari *library* bahasa pemrograman lainnya [22].

Grails adalah sebuah *framework open source* yang menggunakan bahasa pemgrogramman *Groovy* (berdasarkan Java Platform). Hal ini dimaksudkan untuk

menjadi framework dengan *high-productivity* dengan mengikuti "*coding and invention*" paradigma, menyediakan lingkungan pengembagan yang berdiri sendiri. *Grails* didesain untuk mudah di pelajari, mudah mengembangkan aplikasi dan mudah di rawat. *Grails* mencoba menawarkan keseimbangan antara konsistensi dan fitur-fitur yang powerful.

Grails adalah salah satu dari sekian banyak framework MVC (*model-view-controller*) yang di buat pada platform *Java*. Karena di bangun pada platform *Java*, maka *Grails* dapat memanfaatkan pustaka-pustaka *Java* yang telah ada sebelumnya. Dan dalam membuat *Grails*, pihak pengembang memanfaatkan *Hibernate* dan *Springs* sebagai pondasi dasarnya namun dengan antarmuka yang lebih sederhana dan konsisten.

2.6.3. SQL

SQL atau *Structured Query Language* adalah bahasa komputer yang biasanya digunakan untuk mengelola dan berinteraksi dengan data dalam basis data yang berelasi. SQL merupakan bahasa yang paling banyak diimplentasikan dalam basis data, sehingga menjadi standar bahasa untuk mengelola basis data [19]. Bahasa SQL terbagi menjadi beberapa bagian, yaitu *schema statements* yang digunakan untuk mendefinisikan struktur data yang tersimpan dalam basis data, *data statements* yang digunakan untuk memanipulasi struktur data yang telah didefinisikan sebelumnya oleh *schema statements*, dan terakhir adalah *transaction statements* yang digunakan untuk memulai, mengakhiri, atau *roll back* transaksi [20].