

BAB 2

LANDASAN TEORI

2.1 Pembangkit Pertanyaan

Pembangkit pertanyaan adalah sebuah teknik untuk membangkitkan pertanyaan yang berasal dari sebuah kalimat atau teks yang diberikan kepada sistem. Proses yang dilakukan pada proses ini adalah membangkitkan pertanyaan dengan menggunakan kata tanya yang sesuai dengan kategori kalimat.

Secara umum, sistem pembangkit pertanyaan dapat membantu dalam bidang-bidang berikut[1].

1. *Intelligent tutoring systems*, pembangkit pertanyaan juga dapat mengajukan pertanyaan berdasarkan materi pembelajaran untuk memeriksa prestasi peserta didik atau membantu mereka fokus dalam penelitian. Pembangkit pertanyaan juga dapat membantu tutor untuk mempersiapkan pertanyaan potensial dari peserta didik.
2. *Close-domain Question Answering (QA) systems*, beberapa sistem QA domain tertutup menggunakan standar pasangan pertanyaan-jawaban untuk menyediakan layanan QA. Dengan menggunakan pendekatan sistem pembangkit pertanyaan tersebut memungkinkan untuk eksplorasi ke domain lain
3. *Natural language summarization/generating systems* : Pembangkit pertanyaan dapat membantu untuk menghasilkan pertanyaan, misalnya, *Frequently Asked Questions (FAQ)* dari sumber informasi yang diberikan dalam rangka memberikan daftar calon FAQ.

Garis besar konsep pembangkit pertanyaan terdiri dari tiga kategori yaitu : berbasis sintaks, berbasis semantik dan berbasis *template* [1]. Salah satu pendekatan yang dilakukan peneliti terdahulu adalah bagaimana pengolahan bahasa dengan menggunakan teknik secara otomatis untuk menghasilkan pertanyaan-pertanyaan. Implementasi pembangkit pertanyaan dalam teks bahasa Indonesia pun sudah dilakukan, *template* [4][5].

Adapun proses yang harus dilakukan untuk pembangkit pertanyaan adalah sebagai berikut[3].

1. Mengetahui sintaksis yang membentuk pertanyaan. Sintaksis untuk ‘*non-factoid*’ pada umumnya berbeda dengan sintaksis yang menyusun ‘*factoid-question*’. Dalam Bahasa Indonesia terdapat beberapa kata tanya, diantaranya apa, siapa, kapan, mana, mengapa, bagaimana dan sebagainya.
2. Mengetahui kata khusus dari kalimat yang mencirikan kategori pertanyaan yang akan dibangkitkan.

2.1.1 Pembangkit Pertanyaan Non-Factoid

Pada proses pembangkit pertanyaan *non-factoid* akan dilakukan proses pelabelan kandidat pertanyaan *non-factoid* [4]. Proses pelabelan pertanyaan *non-factoid* dilakukan dengan cara mendeteksi keberadaan kata kunci khusus *non-factoid*. Tabel 2.1 Berikut menunjukkan daftar kata kunci untuk pertanyaan *non-factoid*.

Tabel 2.1 Kata Kunci Non-Factoid [4]

Kategori Non-Factoid	Kata Kunci	
	Sebelum Target	Sesudah Target
Definisi	Disebut, dikenal, dinamakan, mendefinisikan	Adalah, yaitu, ialah, merupakan
Alasan	Oleh sebab itu, jadi, memungkinkan adanya, dengan demikian, maka, dikatakan, penyebab terjadinya, sehingga, walau demikian, namun demikian	Sebab, karena, bertujuan
Cara	Berfungsi untuk, berguna untuk	Dengan cara

Adapun *template* untuk membangkitkan pertanyaan *non-factoid* dapat dilihat pada Tabel 2.2 Berikut ini.

Tabel 2.2 *Template Non-Factoid* [4]

Kategori	Kata Kunci	Template
Definisi	Sebelum target	Apa yang dimaksud + (target) + ?
	Disebut, dikenal, dinamakan, mendefinisikan	
	Sesudah target	
	Adalah, yaitu, ialah, merupakan, diartikan	
Alasan	Sebelum target	Mengapa + (target) + ?
	Oleh sebab itu, jadi, memungkinkan, adanya, dengan, demikian, maka, dikatakan, penyebab terjadinya, sehingga, walau demikian, namun demikian	
	Sesudah target	
	Sebab, karena, bertujuan	
Metode	Sebelum target	Bagaimana cara + (target) + ?
	Berfungsi untuk, berguna untuk	
	Sesudah target	
	Dengan cara	

2.1.2 Pembangkit Pertanyaan *Factoid*

Pembangkit pertanyaan *factoid* dilakukan menggunakan *template* pertanyaan *factoid* dan hasil dari analisis sintaksis berupa fungsi gramatikal. *template* untuk pertanyaan *factoid* setiap kalimat dapat dilihat pada tabel 2.3 Berikut ini.

Tabel 2.3 *Template Factoid*[4]

Kata tanya	<i>Template</i> Pertanyaan
Apa	Fungsi yang ditanyakan : Subjek
	NE Tag : <i>OTHER</i> Template : Apa yang + (predikat aktif) + (subjek) + (pelengkap)* + (keterangan)* +? Note : * = jika tersedia
	Fungsi yang ditanyakan : Objek
	NE Tag : <i>OTHER</i> Template : Apa yang + (predikat pasif) + (subjek) + (pelengkap)* + (keterangan)* +? Note : * = jika tersedia
Siapa	Fungsi yang ditanyakan : Subjek
	NE Tag : <i>PERSON, ORGANIZATION</i> Template : Siapa yang + (predikat aktif) + (objek) + (pelengkap)* + (keterangan)* +? Note : * = jika tersedia
	Fungsi yang ditanyakan : Objek
	NE Tag : <i>PERSON, ORGANIZATION</i> Template : Siapa yang + (predikat pasif) + (objek) + (pelengkap)* + (keterangan)* +? Note : * = jika tersedia
Mana	Fungsi yang ditanyakan : Keterangan tempat
	NE Tag : <i>LOCATION</i> Template : (di / ke / dari) + mana + (subjek) + (predikat) + (objek) + (pelengkap)* +?

Kata tanya	<i>Template</i> Pertanyaan
	Note : * = jika tersedia
Kapan	Fungsi yang ditanyakan : Keterangan waktu NE Tag : <i>TIME</i> Template : Kapan + (subjek) + (predikat) + (objek) + (pelengkap)* + (keterangan)* +? Note : * = jika tersedia

2.2 *Preprocessing*

Preprocessing pada pembangkit pertanyaan terdapat beberapa proses. Pada tahapan *preprocessing* pembangkit pertanyaan ini memiliki tahapan yaitu tokenisasi kalimat, tokenisasi kata, *casefolding*, pelabelan jenis kata (POS tag), pengenalan entitas bernama (NER), pendeteksian kalimat majemuk, penentuan fungsi gramatikal, penyesuaian klausa dan pendeteksian kalimat non-definisi.

2.2.1 *Tokenisasi Kalimat*

Tokenisasi kalimat adalah proses pemisahan teks pada dokumen menjadi kumpulan kalimat. Tokenisasi paragraf menjadi kalimat-kalimat tunggal dilakukan menggunakan pemisah (delimiter), dengan tanda titik, tanda seru dan tanda tanya.

2.2.2 *Tokenisasi Kata*

Tokenisasi kata adalah proses pemotongan teks berupa kalimat untuk mendapatkan tiap kata yang menyusunnya. Tokenisasi kalimat menjadi kata-kata tunggal dilakukan menggunakan pemisah (delimiter), dengan tanda spasi.

2.2.3 *Casefolding*

Casefolding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil atau huruf kapital. Tahapan ini dilakukan untuk menyeragamkan huruf agar tidak terdapat variasi huruf dalam teks masukkan. Pada penelitian ini semua huruf dalam teks dokumen diubah menjadi huruf kecil semua.

2.2.4 POS Tagging

Part Of Speech Tagging (POS-tagging) adalah suatu proses untuk memberikan label kelas kata secara otomatis pada suatu kata dalam kalimat[9]. Saat ini sudah banyak penelitian yang dilakukan terkait dengan POS Tagging untuk teks berbahasa Indonesia. Tabel 2.4 berikut adalah daftar *tagset* yang dapat diidentifikasi oleh POS *tagger* yang bersangkutan.

Tabel 2.4 Daftar Tagset POS Tagger [9]

NO	POS	KETERANGAN	CONTOH
1	OP	Open Paranthesis	(([
2	CP	Close Paranthesis)]
3	GM	Slash	/
4	;	Semicolon	;
5	:	Colon	:
6	“	Quotation	“ ”
7	.	Sentence Terminator	. ! ?
8	,	Comma	,
9	-	Dash	-
10	...	Elipsis	...
11	JJ	Adjective	Kaya, Manis
12	RB	Adverb	Sementara, Nanti
13	NN	Common Noun	Mobil
14	NNP	Proper Noun	Bekasi, Indonesia
15	NNG	Genitive Noun	Bukunya
16	VBI	Intransitive Verb	Pergi
17	VBT	Transitive Verb	Membeli
18	IN	Preposition	Di, Ke, Dari
19	MD	Modal	Bisa
20	CC	Coor-Conjunction	Dan, Atau, Tetapi
21	SC	Subor-Conjunction	Jika, Ketika
22	DT	Determinan	Para, Ini, Itu
23	UH	Interjection	Wah, Aduh, Oi
24	CDO	Ordinal Numerals	Pertama, Kedua
25	CDC	Colective Numerals	Bertiga
26	CDP	Primary Numerals	Satu, Dua

NO	POS	KETERANGAN	CONTOH
27	CDI	Irregular Numerals	Beberapa
28	PRP	Personal Pronouns	Saya, Kamu
29	WP	WH – Pronouns	Apa, Siapa
30	PRN	Number Pronouns	Kedua-duanya
31	PRL	Locative Pronouns	Sini, Situ, Sana
32	NEG	Negation	Bukan, Tidak
33	SYM	Symbols	@#\$\$%^&
34	RP	Particles	Pun, Kah
35	FW	Foreign Words	Foreign, Word

2.2.5 Pengenalan Entitas Bernama (NER)

Named Entity Recognition (NER) atau yang disebut pengenalan entitas bernama adalah sebuah proses yang digunakan untuk mengenali dan mengklasifikasi istilah dan frasa sebagai entitas nama dari sebuah teks bebas [12]. NER adalah komponen utama dari ekstraksi informasi yang digunakan untuk mengenali entitas nama (nama orang, lokasi, organisasi), ekspresi waktu (tanggal, waktu, durasi) dan ekspresi bilangan (uang, persen, numerik, kardinal) dari sebuah dokumen, tetapi dapat juga diperluas untuk identifikasi gen, protein dan lainnya sesuai kebutuhan. Penelitian ini menggunakan aplikasi Indonesia NER, yang menggunakan metode HMM [10][11]. Tabel 2.5 Berikut adalah daftar entitas yang dapat diidentifikasi oleh NER yang bersangkutan.

Tabel 2.5 Daftar NER

NER	Contoh
PERSON	Joko Widodo
LOCATION	Bandung, Indonesia
ORGANIZATION	Universitas Gadjah Mada
TIME	15.00
QUANTITY	Satu, dua
OTHER	Lain-lain

2.2.6 Deteksi Kalimat Majemuk

Kalimat yang diproses akan dikelompokkan menjadi tiga jenis, yaitu kalimat majemuk setara, kalimat majemuk bertingkat dan kalimat lainnya. Proses pengelompokkan kalimat majemuk dengan mengidentifikasi kata hubung yang berada pada kalimat. Kata hubung ini adalah kata kunci untuk memisahkan dua klausa yang terdapat dalam sebuah kalimat majemuk, karena kalimat majemuk terdiri dari dua klausa dimana klausa pertama sebagai klausa induk dan klausa kedua sebagai klausa anak [4]. Adapun daftar kata penghubung dan ciri-ciri untuk memisahkan kalimat majemuk setara dan majemuk bertingkat pada tabel 2.6 berikut.

Tabel 2.6 Daftar Kata Penghubung[4]

No	Kalimat Majemuk	Penghubung Klausa	Ciri-ciri Penghubung
1	Majemuk setara	<ol style="list-style-type: none"> 1. Tapi 2. Namun 3. Serta 4. Lalu 5. Kemudian 6. Bahkan 7. Sedangkan 8. Sementara 9. Padahal 10. Bukan.. <p>Melainkan</p>	Lokasi penghubung tetap, yaitu di awal klausa koordinatif.
2	Majemuk bertingkat	<ol style="list-style-type: none"> 1. Karena / sebab 2. Ketika 3. Sejak 4. Jikalau 5. Asalkan 6. Meskipun / walupun 7. Bila / apabila 8. Maka 9. Sehingga 	Subordinator selalu mengikuti unsur intern klausa utama sebagai informasi lanjutan. Karena fungsinya itu, subordinator selalu terletak di awal klausa subordinatif / bawahan.

2.2.7 Identifikasi Frasa

2.2.7.1 Frasa

Frasa adalah satuan gramatikal yang dibentuk dari dua atau beberapa kata yang bersama-sama mendukung satu fungsi gramatikal [19]. Sebagai suatu konstruksi, frasa disusun oleh beberapa unsur pembentuk yang saling berhubungan. Frasa dapat dikelompokkan menjadi tujuh jenis frasa, yaitu; frasa nomina, frasa verba, frasa adjektiva, frasa numeralia, frasa pronomina, frasa adverbial dan frasa preposisional [20].

1. Frasa Verba

Frasa verba (FVB) atau frasa kerja adalah frasa yang berintikan kata kerja (VB), frasa verba merupakan hasil perluasan. Perluasan itu terjadi dari suatu inti yang berupa kata kerja yang bergabung dengan kata lain[20].

2. Frasa Nomina

Frasa Nomina (FN) atau frasa benda adalah frasa yang berintikan kata nomina (NN). Pewatas yang terletak sebelum nomina inti disebut pewatas depan. Pada frasa nomina, pewatas depan biasanya berupa kata numeralia dan adverbial. Sedangkan pewatas yang berada setelah nomina inti disebut pewatas belakang, biasanya berupa nomina, adjektiva, verba, adverbial, numeralia, dan determinan (ini, itu) [20].

3. Frasa Adjektiva

Frasa adjektiva (FJJ) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih yang dapat menggantikan kategori adjektiva. Adjektiva (JJ) berfungsi sebagai inti [20].

4. Frasa Numeralia

Frasa numeralian (FCD) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih, yang dapat menggantikan kategori numeralia. Numeralia (CD) sebagai inti. Umumnya, frasa ini dibentuk dengan

menambahkan kata penggolong, adverbia, atau kata gugus setelah numeralia [20].

5. Frasa Pronomina

Frasa pronomina (FPR) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih yang dapat menggantikan kategori pronomina. Pronomina (PR) berfungsi sebagai inti [20].

6. Frasa Adverbia

Frasa adverbia (FRB) adalah satuan sintaksis yang terbentuk dari dua kata atau lebih dengan adverbia (RB) yang berfungsi sebagai inti [20].

7. Frasa Preposisional

Frasa preposisional (FIN) merupakan frasa eksosntris, tidak terdiri atas inti dan pewatas, tetapi terdiri dari perangkai dan sumbu. Preposisi (IN) berfungsi sebagai perangkai, sedangkan jenis kata yang berfungsi sebagai sumbu adalah nomina, adjektiva, atau adverbia [20].

2.2.7.2 Proses Identifikasi Frasa

Proses identifikasi frasa mengacu pada pola frasa yang telah dijelaskan dan dilakukan pada penelitian yang sebelumnya [4]. Pada proses identifikasi frasa terhadap klausa ini dilakukan dengan cara mencocokkan kombinasi kata di dalam klausa dengan pola frasa yang telah didefinisikan. Pendeteksian pola frasa yang dilakukan berurutan dari kata pertama sampai kata terakhir di dalam klausa. Aturan identifikasi frasa dapat dilihat pada tabel 2.7 berikut ini.

Tabel 2.7 Aturan Identifikasi Frasa[4]

No	Aturan
1	<p>Pointer berada pada kata ke-n, kemudian sistem menyimpan kata ke-n tersebut ke variabel x dan menyimpan POS tag dari kata ke-n ke variabel y, Selanjutnya n ditambah satu.</p> <p>Jika kata ke-n masih ada maka pointer bergerak ke-n, jika tidak ada, salin variabel x lalu tambahkan ke variabel z["frasa"] sedangkan variabel y disalin dan ditambahkan ke variabel z["kategori"].</p>

No	Aturan
2	Pointer berada pada kata ke-n, kemudian sistem akan mengkombinasikan POS tag kata ke-n dengan POS tag yang ada pada variabel y, kombinasi tersebut disimpan ke variabel k. Selanjutnya sistem akan mencari pola frasa yang sesuai dengan kombinasi pada variabel k. Jika tidak ditemukan pola frasa yang sama, sistem akan melanjutkan proses ke poin (3). Namun, jika ditemukan pola frasa yang sama, ganti nilai variabel y dengan nilai variabel k lalu tambahkan kata ke-n ke variabel x. Nilai n ditambah satu dan sistem akan mengulang proses poin (2). Variabel k selalu dikosongkan di setiap perulangan.
3	Susunan kata pada variabel x ditentukan sebagai frasa kemudian variabel x disalin dan ditambahkan ke variabel z['frasa'] sedangkan kategori frasa yang sesuai dengan pola frasa pada variabel y disimpan ke variabel z['kategori']. Variabel x dan y dikosongkan lalu sistem kembali melakukan proses pada poin (1).

Tabel 2.8 berikut ini adalah rangkuman pola frasa, pola frasa ini berfungsi untuk menjadi acuan kombinasi pola frasa yang disesuaikan terhadap aturan pada Tabel 2.7.

Tabel 2.8 Rangkuman Kombinasi Pola Frasa [4]

NO	Kombinasi	Frasa
1	NN + NN	FN
2	NN + NN + NN	FN
3	NN + CD	FN
4	NN + DT	FN
5	NN + PR	FN
6	IN + JJ	FN
7	NN + VB + DT	FN
8	NN + JJ	FN
9	NN + DT	FN
10	NN + JJ + JJ + PR + DT	FN
11	NN + PR + CC + JJ + DT	FN
12	VB + RB	FVB
13	VB + DT	FVB
14	VB + JJ	FVB
15	JJ + JJ	FJJ
16	JJ + RB	FJJ
17	JJ + NN	FJJ

18	JJ + VB	FJJ
19	CD + CD	FCD
20	CD + RB	FCD
21	CD + NN	FCD
22	CD + NN + DT	FCD
23	PR + CD	FPR
24	PR + RB	FPR
25	PR + DT	FPR
26	RB + NN	FN
27	RB + VB	FVB
28	RB + JJ	FJJ
29	RB + CD	FCD
30	RB + DT	FRB
31	RB + PR	FPR
32	RB + PR + CD + RB	FPR
33	RB + RB + VB	FVB
34	RB + RB + DT	FRB
35	MD + RB + VB	FVB
36	MD + JJ	FJJ
37	MD + JJ + VB	FVB
38	MD + VB	FVB
39	IN + NN	FIN
40	IN + NN + NN	FIN
41	IN + NN + JJ	FIN
42	IN + NN + VB	FIN
43	IN + NN + VB + DT	FIN
44	IN + PR	FIN
45	NEG + VB	FVB
46	NEG + VB + VB	FVB
47	VB + VB	FVB
48	VB + VB + VB	FVB
49	NN + CC	FN
50	NN + CC + NN	FN
51	RB + NN + NN	FN
52	CD + NN + NN	FCD
53	SC + JJ	FJJ

54	IN + NN + NN + NN	FIN
55	SC + NN	FN

2.2.8 Penentuan Fungsi Gramatikal

Proses penentuan fungsi gramatikal dilakukan berdasarkan aturan tata bahasa pembentukan pola kalimat yang telah dilakukan pada penelitian sebelumnya [4]. Proses penentuan fungsi gramatikal hampir sama dengan proses identifikasi frasa, yaitu melakukan pengecekan setiap kata maupun frasa yang terdapat pada klausa secara berurutan dari awal. Penentuan fungsi gramatikal ini hanya dilakukan pada klausa induk untuk kalimat majemuk bertingkat, dan untuk kalimat majemuk setara dilakukan pada kedua klausa yang telah dipisahkan. Tabel 2.9 berikut ini menunjukkan aturan-aturan untuk menentukan fungsi gramatikal.

Tabel 2.9 Aturan Fungsi Gramatikal [4]

No	Aturan
1	Jika kata pertama pada klausa adalah konjungsi, jadikan kata tersebut sebagai klausa keterangan sampai tanda baca koma (,).
2	Jika belum ada subjek, dan jenis kata atau jenis frasa sekarang termasuk jenis pembentuk subjek, jadikan kata atau frasa tersebut sebagai subjek.
3	Jika kata atau frasa sebelumnya adalah subjek, dan jenis kata atau jenis frasa sekarang termasuk jenis pembentuk predikat, jadikan kata atau frasa tersebut sebagai predikat.
4	Jika kata atau frasa sebelumnya adalah predikat berjenis verba transitif (VBT) dan jenis kata atau jenis frasa sekarang termasuk jenis pembentuk objek, jadikan kata atau frasa tersebut sebagai objek.
5	Jika kata atau frasa sebelumnya adalah predikat berjenis verba intransitif (VBI), jadikan kata atau frasa sekarang sebagai pelengkap.
6	Jika fungsi subjek, predikat, objek sudah ada. Dan jika kata atau frasa sekarang adalah konjungsi, jadikan kata atau frasa tersebut sebagai klausa keterangan sampai akhir klausa.
7	Jika fungsi subjek, predikat, objek sudah ada. Dan Jika kata atau frasa sekarang mengandung kata depan, jadikan kata atau frasa tersebut sebagai keterangan sampai akhir klausa.
8	Jika fungsi subjek, predikat, objek sudah ada. Dan Jika kata atau frasa sekarang TIDAK mengandung kata depan, jadikan kata atau frasa tersebut sebagai pelengkap sampai akhir klausa.

2.2.9 Penyesuaian Klausa

Proses penyesuaian klausa hanya dilakukan untuk melakukan penyesuaian klausa kedua dengan pertama pada kalimat majemuk setara. Kalimat majemuk setara dapat dipecah menjadi dua buah klausa, sehingga klausa-klausa tersebut dipisahkan sehingga dapat berdiri sendiri-sendiri. Kemudian, klausa-klausa ini akan diproses pada proses pembangkit pertanyaan satu-persatu secara bergantian. Tabel 2.10 berikut adalah aturan untuk penyesuaian klausa.

Tabel 2.10 Aturan Penyesuaian Klausa[4]

No	Aturan
1	Jika subjek klausa kedua adalah kata ganti (PR), maka kata ganti tersebut mengacu pada subjek di klausa pertama.
2	Jika klausa kedua tidak memiliki subjek, maka subjek klausa kedua sama dengan subjek klausa pertama.
3	Jika objek pada klausa kedua adalah kata milik (NNG), maka objek tersebut sama dengan objek pada klausa pertama.

2.2.10 Deteksi Kalimat Non-Definisi

Proses deteksi kalimat non-definisi dilakukan untuk memeriksa suatu kalimat agar pada proses pembangkit pertanyaan tidak menghasilkan pertanyaan definisi. Hal ini disebabkan karena pada proses penentuan fungsi gramatikal kalimat yang mengandung kata 'adalah' dan 'ialah' akan selalu dibangkitkan sebagai pertanyaan definisi, padahal tidak semua kalimat yang mengandung kata 'adalah' dan 'ialah' sebagai kata definisi[5]. Tabel 2.11 berikut ini menunjukkan aturan-aturan deteksi kalimat non-definisi.

Tabel 2.11 Aturan Deteksi Kalimat Non-Definisi [5]

NO	Aturan
1	Jika terdapat penggabungan kata antara nomina dengan pendamping nomina-nya (beberapa, semua, seluruh, sejumlah, banyak) sebelum kata 'adalah', 'ialah' dan membentuk makna kuantitas atau jumlah, maka kalimat tersebut merupakan kalimat non-definisi.
2	Jika terdapat penggabungan kata antara nomina dengan pendamping nomina-nya (ini, itu)

NO	Aturan
	sebelum kata 'adalah', 'ialah' dan membentuk makna penunjuk nomina tersebut, maka kalimat tersebut merupakan kalimat non-definisi.
3	Jika terdapat penggabungan kata antara nomina dengan pendamping nomina-nya (di, pada, dari, ke, kepada) sebelum kata 'adalah', 'ialah' dan membentuk makna lokasi, maka kalimat tersebut merupakan kalimat non-definisi.
4	Jika terdapat NER dengan kategori PERSON sebelum kata 'adalah', 'ialah' maka kalimat tersebut merupakan kalimat non-definisi.
5	Jika terdapat penggabungan kata antara nomina dengan pendamping nomina-nya (yang) sebelum kata 'adalah', 'ialah' dan merupakan perluasan makna nomina tersebut, maka kalimat tersebut merupakan kalimat non-definisi.
6	Jika pada kata pertama merupakan keterangan yang dilengkapi dengan tanda baca koma (,) dan sebelum kata penghubung 'adalah', 'ialah', maka kalimat tersebut merupakan kalimat non-definisi.
7	Jika terdapat penggabungan kata antara verba dengan pendamping verba-nya (belum, sedang, akan, boleh, dapat, harus, wajib, mesti, jangan) sebelum kata 'adalah', 'ialah' dan membentuk makna berbagai modalitas, maka kalimat tersebut merupakan kalimat non-definisi.

2.3 Pemodelan Sistem

Pemodelan sistem merupakan proses menggambarkan konsep sistem yang mampu menjelaskan sistem secara utuh. Adapun pemodelan sistem pada penelitian tugas akhir ini adalah sebagai berikut.

2.3.1 Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah atau urutan prosedur dari suatu program. *Flowchart* sangat membantu analisis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* dibagi menjadi 5 jenis, yaitu [13]:

- a. *Flowchart* sistem, *flowchart* sistem digunakan untuk menerangkan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam

sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem.

- b. *Flowchart* dokumen, *flowchart* dokumen kegunaan utamanya untuk menelusuri alur form dan laporan sistem dari satu bagian ke bagian lain baik bagaimana alur form dan laporan diproses, dicatat dan disimpan.
- c. *Flowchart* skematik, *flowchart* skematik digunakan sebagai alat komunikasi antara analisis sistem dengan seseorang yang tidak familiar dengan simbol-simbol *flowchart* secara umum. Oleh karena itu *flowchart* skematik menggunakan gambar-gambar komputer, peripheral, form-form atau peralatan lain agar orang tersebut dapat mengerti tanpa ada salah pengertian.
- d. *Flowchart* program, *flowchart* program digunakan untuk menjelaskan keterangan yang lebih rinci tentang bagaimana setiap langkah program untuk menggambarkan urutan instruksi dari program komputer.
- e. *Flowchart* proses, *flowchart* proses digunakan untuk penggambaran rekayasa industrial yang memecah dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem.

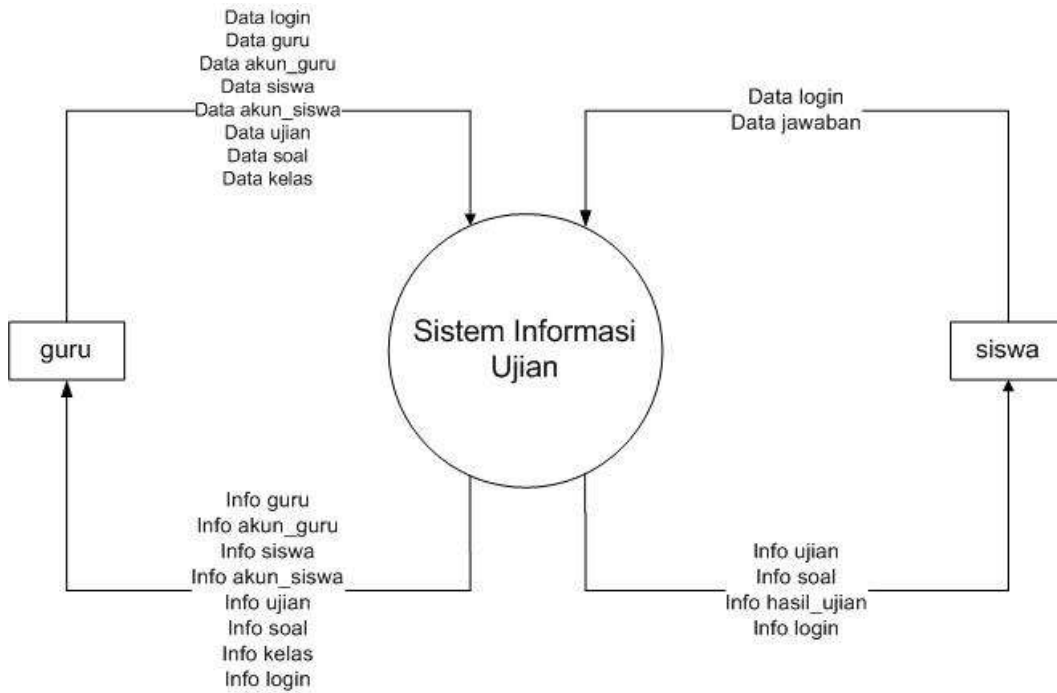
Tujuan dari penggunaan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana dan jelas. Berikut adalah contoh gambar *flowchart* untuk melaksanakan ujian tulis.



Gambar 2.1 Contoh *Flowchart*

2.3.2 Diagram Konteks

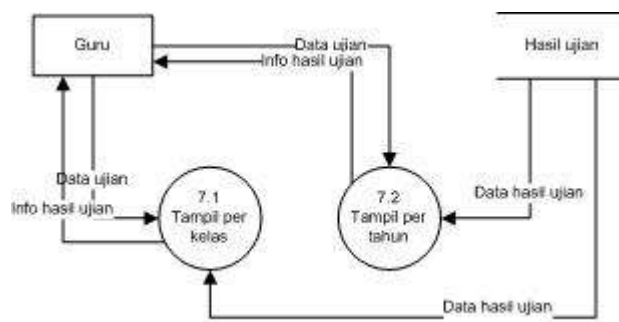
Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup sebuah sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem dan *output* dari sistem [17]. Gambar 2.2 adalah contoh diagram konteks sistem informasi ujian.



Gambar 2.2 Contoh Diagram Konteks

2.3.3 Data Flow Diagram (DFD)

Data flow diagram digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. DFD merupakan alat pada metodologi pengembangan sistem yang terstruktur dan dapat mengembangkan arus data di dalam sistem dengan berurutan dan jelas[16]. Gambar 2.3 adalah contoh dari DFD pada proses tampil hasil ujian.



Gambar 2.3 Contoh Data Flow Diagram

2.4 Bahasa Pemrograman

Bahasa pemrograman adalah instruksi standar untuk memerintah komputer dimana didalamnya terdapat kumpulan sintaks yang sistematis untuk dipakai mendefinisikan program komputer. Ada banyak macam bahasa pemrograman seperti Visual Basic, C, C++, C#, Java, PHP, Python dan lain-lain.

2.4.1 PHP

PHP (*Personal Home Page*) adalah bahasa *server side* yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis. Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya akan dikirimkan ke *browser* dengan format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman *web* lebih terjamin. PHP dirancang untuk membuat halaman *web* yang dinamis, yaitu halaman *web* yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman *web*. PHP dapat berjalan pada berbagai *web server* seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami dan lain-lain. PHP juga mampu berjalan di banyak sistem operasi yang beredar saat ini, seperti : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac Os, Solaris [14].

Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data atau *Database Management System* (DBMS), sehingga dapat menciptakan halaman *web* yang dinamis.

2.4.2 SQL

SQL (*Structure Query Language*) adalah bahasa komputer yang biasanya digunakan untuk mengelola dan berinteraksi dengan data dalam basis data yang berelasi. Dengan SQL, kita dapat mengakses database, menjalankan query untuk mengambil data dari database, menambahkan data ke database, menghapus data di dalam database, dan mengubah data dalam database. SQL sering juga disebut dengan istilah *query*, dan bahasa SQL secara praktiknya digunakan sebagai

bahasa standar untuk manajemen *database* relasional. Hingga saat ini hampir seluruh server *database* atau *software database* mengenal dan mengerti bahasa SQL [16].

2.5 Perangkat Lunak Pendukung

Pada penelitian ini, terdapat beberapa perangkat lunak yang digunakan untuk mendukung dalam pembuatan program penelitian ini sebagai berikut.

2.5.1 XAMPP

XAMPP adalah perangkat lunak bebas (*free software*), yang mendukung untuk banyak sistem operasi, yang merupakan kompilasi dari beberapa program. XAMPP merupakan satu paket software web server yang terdiri dari Apache, Mysql, PHP dan phpMyadmin. Penggunaan XAMPP sangat mudah, karena tidak memerlukan konfigurasi Apache, PHP dan MySQL secara manual, XAMPP melakukan instalasi dan konfiigurasi secara otomatis [18].

2.5.2 Web Browser

Web browser adalah sebuah aplikasi perangkat lunak yang memungkinkan pengguna untuk menayangkan dan berinteraksi dengan tulisan, gambar, video, music dan berbagai informasi lainnya yang terdapat pada halaman *web* di sebuah situs di *World Wide Web* atau di jaringan LAN lokal. Penggunaan *web browser* adalah untuk membawa sumber informasi kepada pengguna, proses ini dimulai ketika pengguna memasukkan *Uniform Resource Locator (URL)*. *Web browser* memungkinkan pengguna secara cepat dan mudah mengakses informasi yang diberikan oleh banyak situs *web* dengan cara menjelajahi link tersebut. *Web browser* memformat informasi HTML untuk di tayangkan, oleh karena itu penampakan halaman *web* akan agak berbeda dari satu *browser* ke *browser* yang lain. Ada beberapa *web browser* yang dapat digunakan di komputer, yaitu, Mozilla Firefox, Google Chrome, Safari, Konqueror, Opera, Flock, Epiphany, K-Meleon dan AOL [15].