

**IMPLEMENTASI *PATTERN BASED APPROACH* PADA
QUESTION ANSWERING SYSTEM
(STUDI KASUS BUKU PEDOMAN PENDIDIKAN
PROGRAM TEKNOLOGI INFORMASI DAN ILMU
KOMPUTER)**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer



Disusun oleh :

RESNU DWIHANU V

NIM. 0810680058

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER**

MALANG

2013



LEMBAR PERSETUJUAN

**IMPLEMENTASI *PATTERN BASED APPROACH* PADA
QUESTION ANSWERING SYSTEM
(STUDI KASUS BUKU PEDOMAN PENDIDIKAN
PROGRAM TEKNOLOGI INFORMASI DAN ILMU
KOMPUTER)**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer



Disusun oleh :

RESNU DWIHANU V
NIM. 0810680058

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Sutrisno, MT.

NIP. 19570325 198701 1 001

Eriq M. Adams J., ST., M.Kom.

NIK. 850410 06 1 1 0027



LEMBAR PENGESAHAN

**IMPLEMENTASI *PATTERN BASED APPROACH* PADA
QUESTION ANSWERING SYSTEM
(STUDI KASUS BUKU PEDOMAN PENDIDIKAN
PROGRAM TEKNOLOGI INFORMASI DAN ILMU
KOMPUTER)**

SKRIPSI

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh :

RESNU DWIHANU V
NIM. 0810680058

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 22 Januari 2013

Penguji I

Ir. Heru Nurwasito, M.Kom.
NIP. 19650402 199002 1 001

Penguji II

Yusi Tyroni M. S.Kom., MS
NIP. 19800228 200604 1 001

Penguji III

Indriati, ST.,M.Kom
NIK. 831013 06 1 2 0035

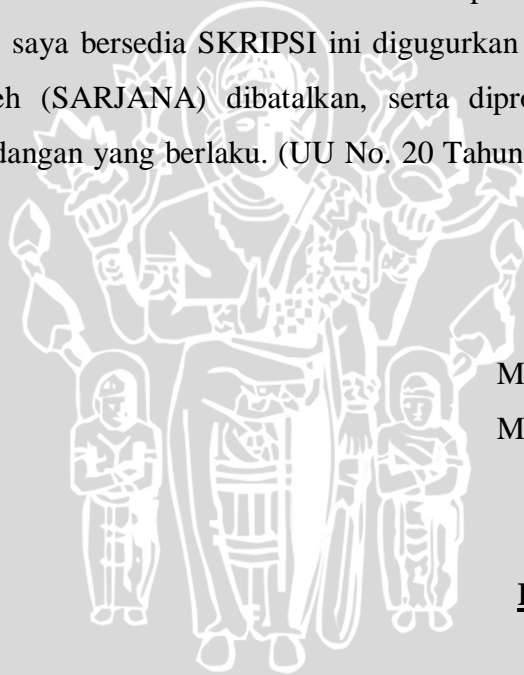
Mengetahui,
Ketua Program Studi Teknik Informatika

Drs. Marji., M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah SKRIPSI ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku. (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, Januari 2013
Mahasiswa,

Resnu Dwihanu V
0810680058



KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, yang telah melimpahkan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi dengan judul "IMPLEMENTASI *PATTERN BASED APPROACH* PADA *QUESTION ANSWERING SYSTEM* (STUDI KASUS BUKU PEDOMAN PENDIDIKAN PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER)".

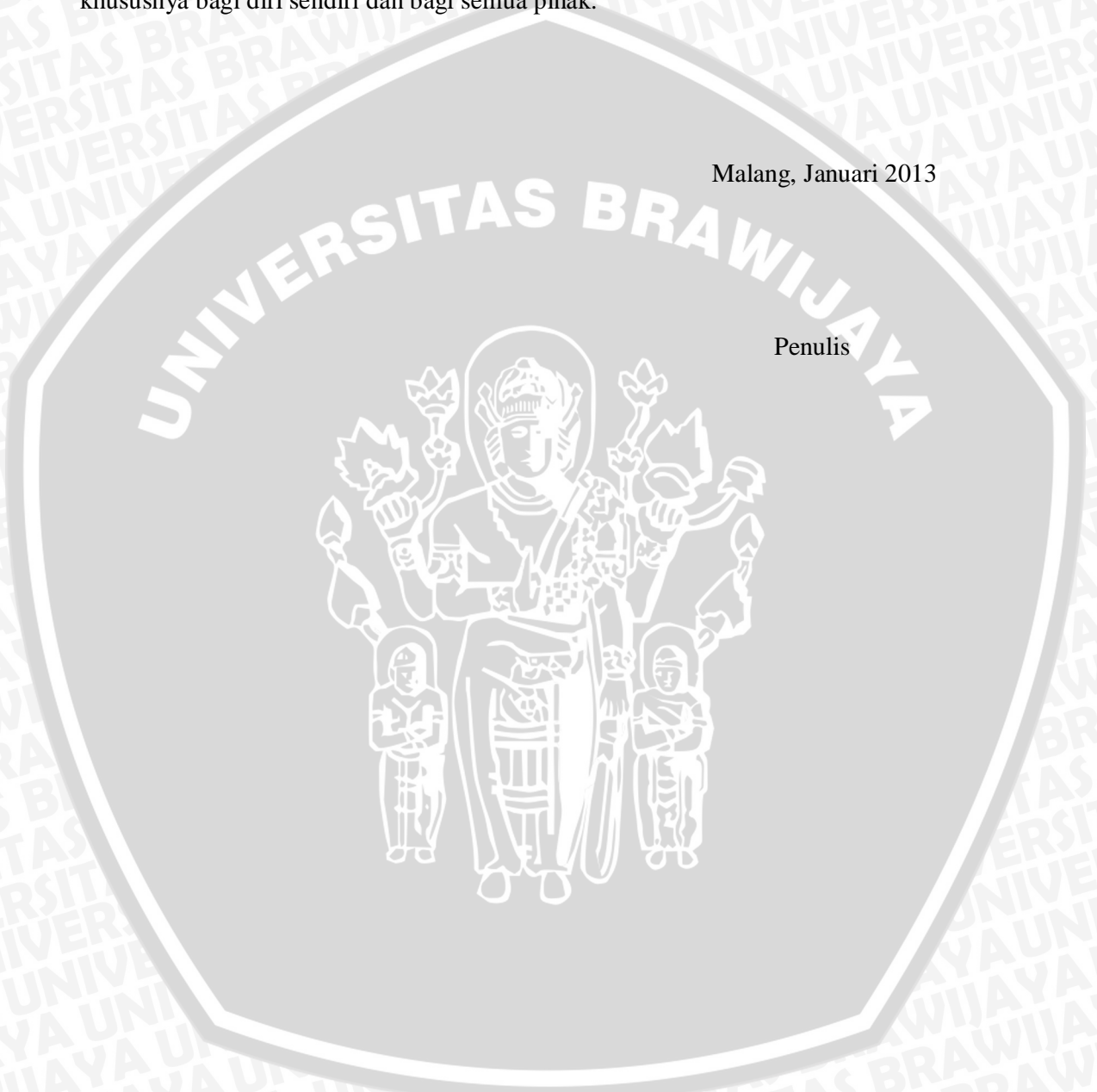
Penyusunan skripsi ini dapat terlaksana dengan baik karena adanya bantuan secara langsung maupun tidak langsung dari pihak tertentu, diantaranya :

1. Bapak Agus Eko dan Ibu Tri Hendarwati, sebagai orang tua saya yang telah memberikan dukungan moral dan material.
2. Bapak Ir. Sutrisno, MT. selaku dosen pembimbing I yang telah memberikan ilmu dan saran untuk skripsi ini.
3. Bapak Eriq M. Adams J., ST., M.Kom. selaku dosen pembimbing II yang juga memberikan ilmu dan saran untuk skripsi ini.
4. Bapak ibu dosen penguji yang telah memberikan kritik dan saran untuk skripsi ini.
5. Segenap bapak dan ibu dosen program studi Teknik Informatika beserta seluruh staf administrasi yang telah membantu selama perkuliahan.
6. Retni Cindra Gusti, kakak saya yang telah memberikan doa dan semangat.
7. Johan Iswara, Ayok Luthfi, Arif Fahmi, Rakhma Indah, Novita Rudiarsih, Puri Cemani, Gita Ayu, Davi Nugraha, dan teman-teman Teknik Informatika 2008, serta kakak dan adik tingkat yang telah membantu memberi kritik dan saran pada skripsi ini.
8. Semua pihak yang telah membantu dan memberi semangat dalam proses penulisan skripsi ini.

Saya sadar bahwa skripsi ini masih terdapat banyak kekurangan, oleh karena itu kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Saya berharap skripsi ini dapat bermanfaat khususnya bagi diri sendiri dan bagi semua pihak.

Malang, Januari 2013

Penulis



ABSTRAK

Resnu Dwihanu. 2013. Implementasi *Pattern Based Approach* Pada *Question Answering System* (Studi Kasus Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer). Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Ir. Sutrisno, MT. dan Eriq M. Adams J., ST., M.Kom.

Setiap mahasiswa pasti membutuhkan acuan atau pedoman dalam melakukan kegiatan akademik di dunia perkuliahan. Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer dimaksudkan untuk dapat menjadi acuan bagi setiap mahasiswa di PTIHK, akan tetapi pemanfaatannya masih memiliki beberapa keterbatasan. Untuk dapat meningkatkan pemanfaatan Buku Pedoman Pendidikan tersebut, dalam skripsi ini diimplementasikan sebuah *Question Answering System* dengan memanfaatkan metode *Pattern Based Approach*. *Input* pertanyaan pada *Question Answering System* diinterpretasikan dengan menggunakan pola pertanyaan. Hasil interpretasi pertanyaan tersebut digunakan untuk mencari dokumen relevan yang akan menjadi kandidat jawaban. Kandidat jawaban melalui proses *filtering* dan *scoring* untuk membuang kandidat yang tidak sesuai. Pola jawaban digunakan untuk mengekstrak potongan kalimat yang menjadi jawaban pada kandidat jawaban. Tahap perancangan terdiri dari perancangan *Question Answering System*, dan Perancangan Algoritma *Pattern Based Approach*. Perancangan *Question Answering System* terdiri dari *System Plan*, *Arsitektur Sistem*, dan Perancangan *Database*. Perancangan algoritma terdiri dari pengumpulan dokumen, pembentukan pola, dan algoritma yang diimplementasikan pada tiap komponen arsitektur sistem. Pengujian algoritma *Pattern Based Approach* dilakukan dengan menguji tingkat akurasi jawaban yang dihasilkan *Question Answering System*. Hasil pengujian akurasi adalah presisi sebesar 80,85% dan recall sebesar 100%.

Kata kunci : sistem temu kembali informasi, sistem tanya jawab, pendekatan berbasis pola.

ABSTRACT

Resnu Dwihanu. 2013. *Pattern Based Approach Implementation to Question Answering System (Case Study Educational Handbooks of Information Technology and Computer Science Program)*. Information Technology and Computer Science Program, Brawijaya University, Malang. Advisor: Ir. Sutrisno, MT. and Eriq M. Adams J., ST., M.Kom.

Every undergraduate student needs references or guidelines for their academic activities. Educational Handbook of Information Technology and Computer Science Program is meant to be a reference for every student in PTIIK, but its utility still has some limitations. Question Answering System can be used to maximize the Educational Handbook utilization. Pattern Based Approach method was implemented into the Question Answering. Question accepted by Question Answering System interpreted using Question Patterns. Interpreted question was used to search the relevant documents and became answer candidate. Answer candidate was filtered and scored, and then answer was extracted using Answer Patterns. The Design phase consists of Question Answering System design and Algorithm design. The Question Answering System design consists of System Plan, System Architecture, and Database Design. The algorithm design consists of the documents collection, the patterns formations, and the algorithm that was implemented on each component of the system architecture. The result of the test are 80.85% precision and 100% recall.

Key words : *information retrieval system, question answering system, pattern based approach.*

DAFTAR ISI

	Halaman
JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN ORISINALITAS SKRIPSI	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR PERSAMAAN	xiv
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
BAB II DASAR TEORI	
2.1 <i>Information Retrieval System</i>	5
2.2 <i>Natural Language Processing</i>	6
2.3 <i>Question Answering System</i>	7
2.3.1 <i>Arsitektur Question Answering System</i>	8
2.3.2 <i>Question Analysis</i>	8
2.3.3 <i>Query Generation and Search</i>	9
2.3.4 <i>Answer Extraction and Selection</i>	9
2.4 <i>Pattern Based Approach</i>	9
BAB III METODE PENELITIAN	
3.1 Analisis Kebutuhan	12

3.2	Pengumpulan Data	12
3.3	Perancangan Sistem.....	12
3.4	Implementasi	15
3.4.1	<i>Question Analyzer</i>	15
3.4.2	<i>Query Generator</i>	15
3.4.3	<i>Search Engine</i>	15
3.4.4	<i>Answer Extractor</i>	16
3.4.5	<i>Pattern Learning</i>	18
3.5	Pengujian	20

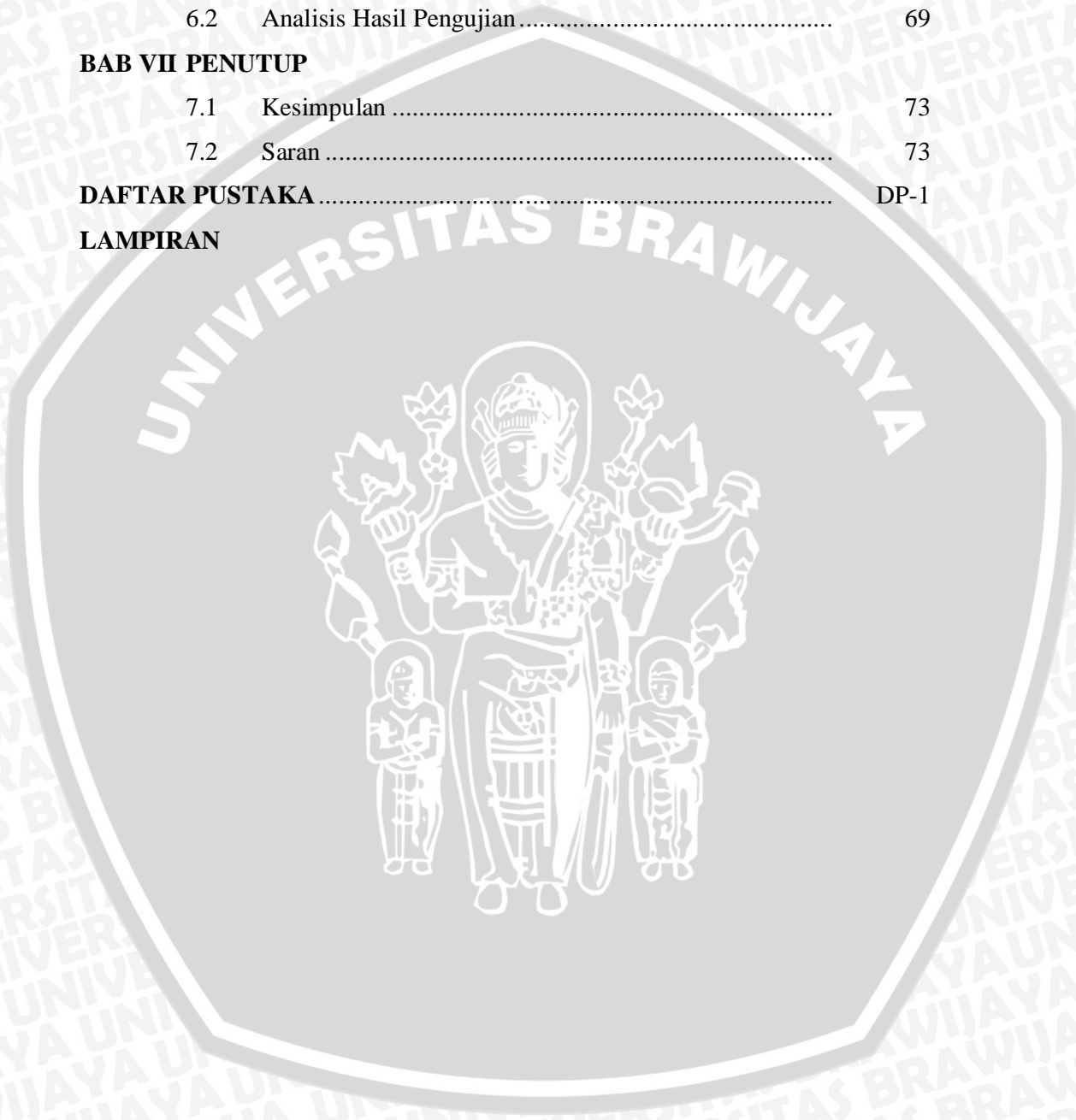
BAB IV PERANCANGAN

4.1	Perancangan <i>Question Answering System</i>	23
4.1.1	<i>System Plan Question Answering System</i>	23
4.1.2	Arsitektur <i>Question Answering System</i>	24
4.1.3	Perancangan <i>Database</i>	29
4.2	Perancangan Algoritma	31
4.2.1	Pengumpulan Data	32
4.2.2	Algoritma pada <i>Question Analyzer</i>	37
4.2.3	Algoritma pada <i>Search Engine</i>	40
4.2.4	Algoritma pada <i>Answer Extractor and Selector</i>	41
4.2.5	Contoh Implementasi Kasus	46

BAB V IMPLEMENTASI

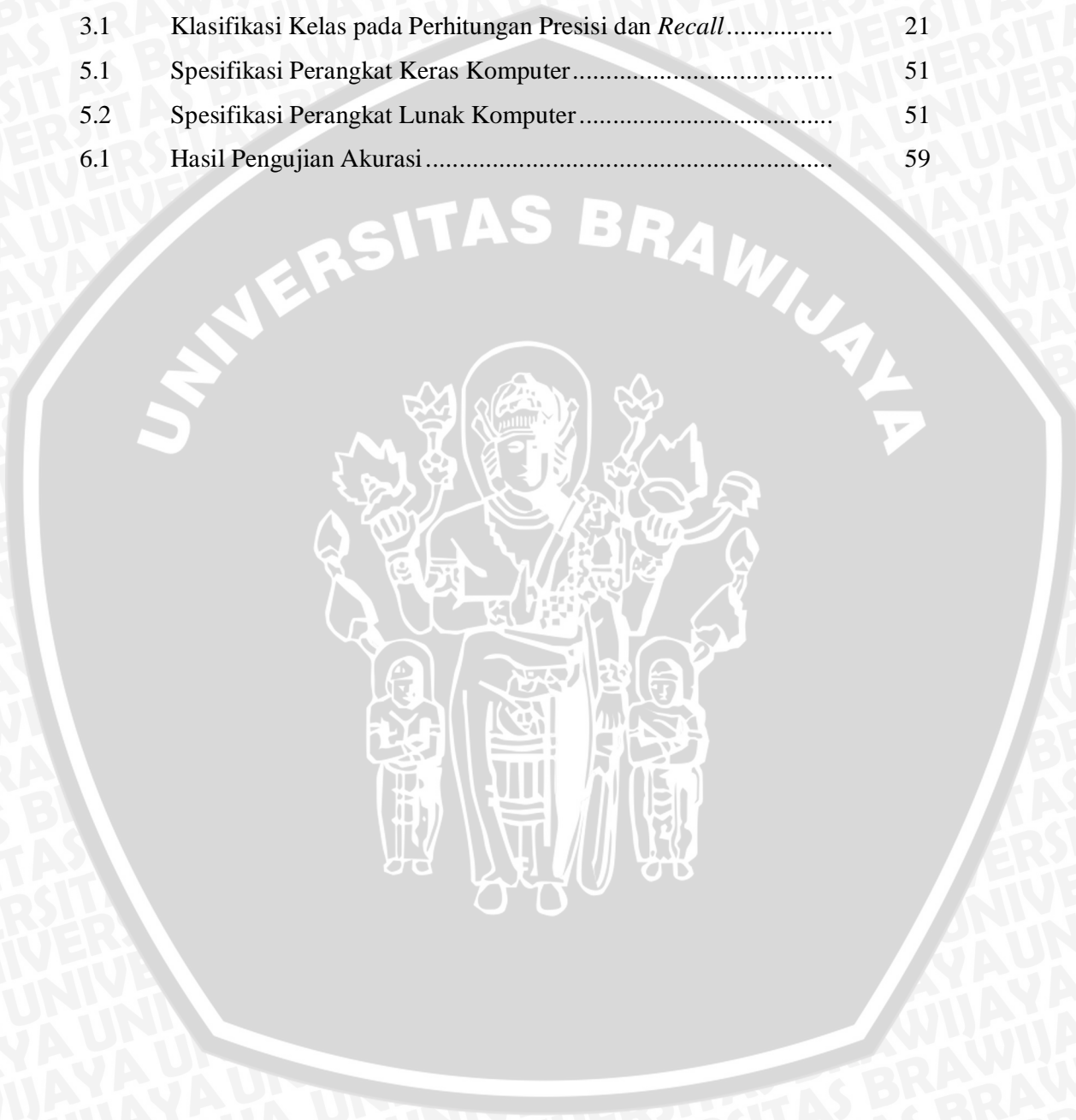
5.1	Spesifikasi Sistem.....	51
5.1.1	Spesifikasi Perangkat Keras	51
5.1.2	Spesifikasi Perangkat Lunak	51
5.2	Batasan-Batasan Implementasi	52
5.4	Implementasi Antarmuka.....	52
5.4.1	Tampilan Halaman Utama	52
5.4.2	Tampilan Halaman <i>Login</i>	54
5.4.3	Tampilan Halaman <i>Administrator</i>	55
5.4.4	Tampilan Halaman Pengelolaan Dokumen.....	56
5.4.5	Tampilan Halaman Pengelolaan Pola Pertanyaan.....	56
5.4.6	Tampilan Halaman Pengelolaan Pola Jawaban.....	57

5.4.7	Tampilan Halaman Pengelolaan Respon Jawaban.....	57
BAB VI PENGUJIAN DAN ANALISIS		
6.1	Pengujian Akurasi	58
6.2	Analisis Hasil Pengujian.....	69
BAB VII PENUTUP		
7.1	Kesimpulan	73
7.2	Saran	73
DAFTAR PUSTAKA		DP-1
LAMPIRAN		



DAFTAR TABEL

Tabel		Halaman
3.1	Klasifikasi Kelas pada Perhitungan Presisi dan <i>Recall</i>	21
5.1	Spesifikasi Perangkat Keras Komputer.....	51
5.2	Spesifikasi Perangkat Lunak Komputer.....	51
6.1	Hasil Pengujian Akurasi.....	59



DAFTAR GAMBAR

Gambar		Halaman
2.1	Bagian – bagian <i>Information Retrieval (IR) System</i>	5
2.2	Arsitektur <i>QA System</i> pada <i>Framework OpenEphyra</i>	8
3.1	Diagram Alir metode penelitian.....	11
3.2	Diagram Alir Sistem.....	13
3.3	Arsitektur Sistem.....	15
4.1	Diagram Alir Perancangan	22
4.2	<i>System Plan Question Answering System</i>	23
4.3	Arsitektur Sistem.....	25
4.4	<i>ERD Question Answering System</i>	30
4.5	<i>Physical Diagram database</i>	31
4.6	Diagram Alir <i>Preprocessing</i> Dokumen.....	33
4.7	<i>Pseudocode Question Analyzer</i>	40
4.8	<i>Pseudocode Search Engine</i>	41
4.9	Diagram Alir Proses <i>Answer Extractor and Selector</i>	42
4.10	<i>Pseudocode Answer Extractor and Selector</i>	46
5.1	Tampilan Halaman Utama	53
5.2	Tampilan Pemrosesan Pertanyaan pada Halaman Utama	53
5.3	Tampilan Jawaban pada Halaman Utama	54
5.4	Tampilan Halaman <i>Login</i>	55
5.5	Tampilan Halaman <i>Administrator</i>	55
5.6	Tampilan Halaman Pengelolaan Dokumen	56
5.7	Tampilan Halaman Pengelolaan Pola Pertanyaan	56
5.8	Tampilan Halaman Pengelolaan Pola Jawaban	57
5.9	Tampilan Halaman Pengelolaan Respon Jawaban.....	57
6.1	Cuplikan isi tabel hasil_jawaban.....	70
6.2	Cuplikan tabel temp_jawaban2.....	70
6.3	Cuplikan tabel respon_jawaban	71

DAFTAR PERSAMAAN

Persamaan		Halaman
3-1	<i>Number of Keywords</i> [SCH-05]	16
3-2	<i>Confidence dan Support</i> [TOB-10]	20
3-3	Presisi.....	21
3-4	<i>Recall</i>	21
4-1	<i>Confidence dan Support</i> [TOB-10]	37
4-2	<i>Number of Keywords</i> [SCH-05]	43
4-3	Skor <i>Number of Keywords</i>	43
4-4	Skor <i>Hit Position</i>	44



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Setiap mahasiswa pasti membutuhkan acuan atau pedoman dalam melakukan kegiatan akademik di dunia perkuliahan. Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer memang dapat menjadi acuan bagi setiap mahasiswa Program Teknologi Informasi Dan Ilmu Komputer (PTI IK) Universitas Brawijaya. Tetapi pemanfaatan Buku Pedoman Pendidikan masih banyak memiliki kekurangan. Salah satu contohnya adalah masih banyak mahasiswa yang kurang memahami dan mengetahui isi dari Buku Pedoman Pendidikan dikarenakan malas membaca isi buku pedoman. Contoh lainnya adalah untuk mencari suatu informasi yang terdapat pada Buku Pedoman Pendidikan dapat memakan waktu yang lama. Orang dari pihak luar pun juga akan mengalami kesulitan jika ingin mengetahui informasi mengenai PTI IK karena tidak memiliki sumber informasinya, yaitu Buku Pedoman Pendidikan. Selain itu, Buku Pedoman Pendidikan selalu diperbaharui setiap periode tertentu. Pembaharuan ini dapat menimbulkan kendala bagi mahasiswa, terutama mahasiswa lama dalam mengakses informasi karena buku pedoman yang dimilikinya sudah tidak sesuai dengan isi buku pedoman yang terbaru.

Dalam memenuhi kebutuhan mahasiswa akan informasi pada Buku Pedoman Pendidikan, salah satu implementasi dari sistem temu kembali informasi (*Information Retrieval System*) dapat digunakan, yaitu berupa *Question Answering System*. Secara umum, *Question Answering System* adalah sistem yang menerima masukan berupa pertanyaan dari pengguna dan mengembalikan keluaran berupa jawaban. Dibandingkan dengan *search engine* yang hasilnya tidak selalu sesuai dengan kebutuhan pengguna, *Question Answering System* dapat memberikan informasi yang dibutuhkan oleh pengguna yang bertanya [BAS-09]. *Question Answering System* menghasilkan informasi yang lebih spesifik yaitu langsung berupa jawaban. Sedangkan hasil dari *search engine* masih berupa kumpulan dokumen dan belum bisa menyediakan informasi yang dibutuhkan

pengguna secara langsung karena pengguna masih harus mencari informasi yang diinginkan pada kumpulan dokumen yang dihasilkan *search engine*.

Pattern Based Approach (PBA) adalah salah satu metode yang dapat diterapkan dalam membangun sebuah *Question Answering System*. Metode pendekatan ini memanfaatkan pola pertanyaan (*Question Pattern*) dan pola jawaban (*Answer Pattern*) dalam proses untuk menemukan jawaban dari pertanyaan. Metode ini memiliki kesamaan dengan metode yang diterapkan pada *framework Open Ephyra*. Metode ini dipilih karena metode *Pattern Based Approach* ini sangat menjanjikan dan cocok untuk diimplementasikan dalam *Question Answering* berbahasa Indonesia dengan akurasi interpretasi sebesar 90% dan akurasi jawaban sebesar 35% [TOB-10].

Oleh karena itu dalam skripsi ini akan diimplementasikan metode *Pattern Based Approach* pada sebuah *Question Answering System* yang dapat digunakan untuk mendapatkan informasi dan menjawab pertanyaan yang terdapat pada Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer.

1.2 Rumusan Masalah

Berdasar latar belakang masalah diatas, maka perumusan masalah dalam skripsi ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan metode *Pattern Based Approach* pada *Question Answering System*?
2. Bagaimana menguji algoritma *Pattern Based Approach* pada *Question Answering System*?

1.3 Batasan Masalah

Pembahasan dalam skripsi ini dibatasi sebagai berikut:

1. Sumber informasi yang digunakan dokumen adalah Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer Tahun Akademik 2012/2013.
2. Masukan yang diterima sistem adalah pertanyaan dalam bahasa Indonesia.
3. Proses *stemming* dilakukan dengan menggunakan metode Arifin.
4. Sistem tidak memperhatikan kesalahan ejaan atau penulisan kata.

5. Aplikasi dibangun menggunakan bahasa pemrograman *php*.
6. *Database Management System* yang digunakan adalah *MySQL*.

1.4 Tujuan

Tujuan dari penelitian ini adalah mengimplementasikan metode *Pattern Based Approach* pada *Question Answering System* dan diterapkan pada Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer, serta menguji algoritma *Pattern Based Approach* yang diimplementasikan pada *Question Answering System*.

1.5 Manfaat

Manfaat dari penelitian ini adalah dapat memudahkan pengguna terutama bagi mahasiswa PTIIK dalam mendapatkan informasi yang terdapat pada Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer.

1.6 Sistematika Penulisan

Sistematika penulisan dari penelitian ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, manfaat, sistematika penulisan, dan jadwal pengerjaan.

BAB II Dasar teori

Menguraikan teori dasar dan teori penunjang yang berkaitan dengan *Information Retrieval System*, *Natural Language Processing*, *QA System*, dan *Stemming Arifin*.

BAB III Metode Penelitian

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur, analisa kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV Perancangan

Membahas analisis kebutuhan dan perancangan algoritma *Pattern Based Approach* yang diimplementasikan pada *Question Answering System*.

BAB V Implementasi

Membahas tentang implementasi dari sistem.

BAB VI Pengujian dan Analisis

Memuat proses dan hasil pengujian terhadap sistem yang telah direalisasikan.

BAB VII Penutup

Memuat kesimpulan serta saran yang diperoleh dari pembuatan dan pengujian sistem untuk pengembangan lebih lanjut.



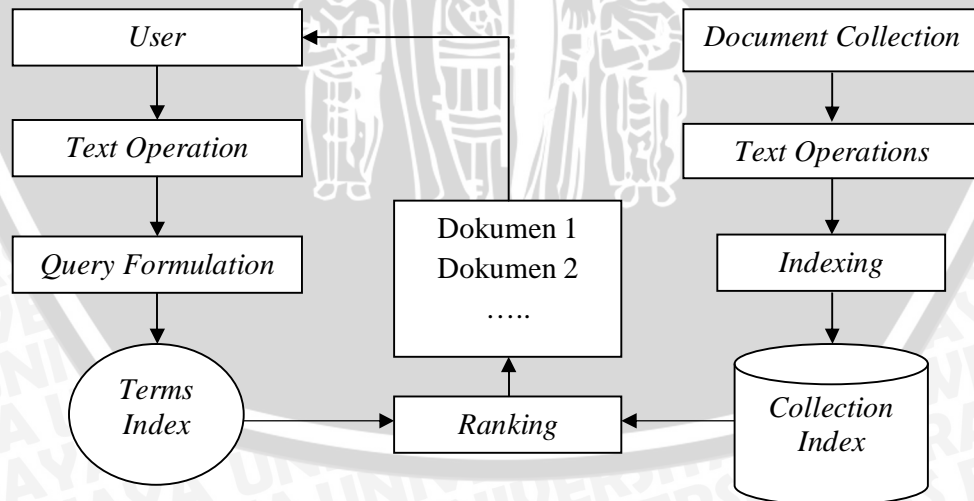
BAB II DASAR TEORI

Bab ini membahas dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai Implementasi *Pattern Based Approach* pada *Question Answering System* beberapa dasar teori yang dimaksud adalah *Information Retrieval System*, *Natural Language Processing*, *Question Answering System*, *Pattern Learning Approach*, dan *Stemming Arifin*.

2.1 Information Retrieval System

Information Retrieval (IR) System adalah sistem yang digunakan untuk menemukan kembali (*retrieve*) informasi-informasi yang relevan terhadap kebutuhan pengguna dari suatu kumpulan informasi secara otomatis [BUN-08].

Sistem *IR* terutama berhubungan dengan pencarian informasi yang isinya tidak memiliki struktur. Demikian pula ekspresi kebutuhan pengguna yang disebut *query*, juga tidak memiliki struktur. Hal ini yang membedakan sistem *IR* dengan sistem *database*. Dokumen adalah contoh informasi yang tidak terstruktur. Isi dari suatu dokumen sangat tergantung pada pembuat dokumen tersebut.



Gambar 2.1 Bagian – bagian *Information Retrieval (IR) System*

Sumber: [BUN-08]

Dari gambar 2.1, terlihat bahwa terdapat dua proses operasi dalam sistem *IR*. Proses pertama dimulai dari koleksi dokumen dan proses kedua dimulai dari *query* pengguna. Proses pertama yaitu pemrosesan terhadap koleksi dokumen menjadi *database* indeks tidak ada ketergantungan dengan proses kedua. Sedangkan proses kedua tergantung dari keberadaan *database* indeks yang dihasilkan pada proses pertama.

Bagian-bagian dari sistem *IR* menurut gambar 1 meliputi :

- 1) **Text Operations** (operasi terhadap teks) yang meliputi pemilihan kata-kata dalam *query* maupun dokumen (*term selection*) dalam pentransformasian dokumen atau *query* menjadi *term index* (indeks dari kata-kata).
- 2) **Query formulation** (formulasi terhadap *query*) yaitu memberi bobot pada indeks kata-kata *query*.
- 3) **Ranking** (perangkingan), mencari dokumen-dokumen yang relevan terhadap *query* dan mengurutkan dokumen tersebut berdasarkan kesesuaiannya dengan *query*.
- 4) **Indexing** (pengindeksan), membangun *database* indeks dari koleksi dokumen. Dilakukan terlebih dahulu sebelum pencarian dokumen dilakukan.

Sistem *IR* menerima *query* dari pengguna, kemudian melakukan perangkingan terhadap dokumen pada koleksi berdasarkan kesesuaiannya dengan *query*. Hasil perangkingan yang diberikan kepada pengguna merupakan dokumen yang menurut sistem relevan dengan *query*. Namun relevansi dokumen terhadap suatu *query* merupakan penilaian pengguna yang subjektif dan dipengaruhi banyak factor seperti topik, pewaktuan, sumber informasi maupun tujuan pengguna.

2.2 Natural Language Processing

Natural Language Processing (NLP) atau pengolahan bahasa alami merupakan salah satu bidang ilmu *Artificial Intelligence* (Kecerdasan Buatan) yang mempelajari komunikasi antara manusia dengan komputer melalui bahasa alami. *NLP* mencoba membuat komputer mampu memahami suatu perintah yang dituliskan dalam bentuk bahasa sehari-hari dan diharapkan komputer juga

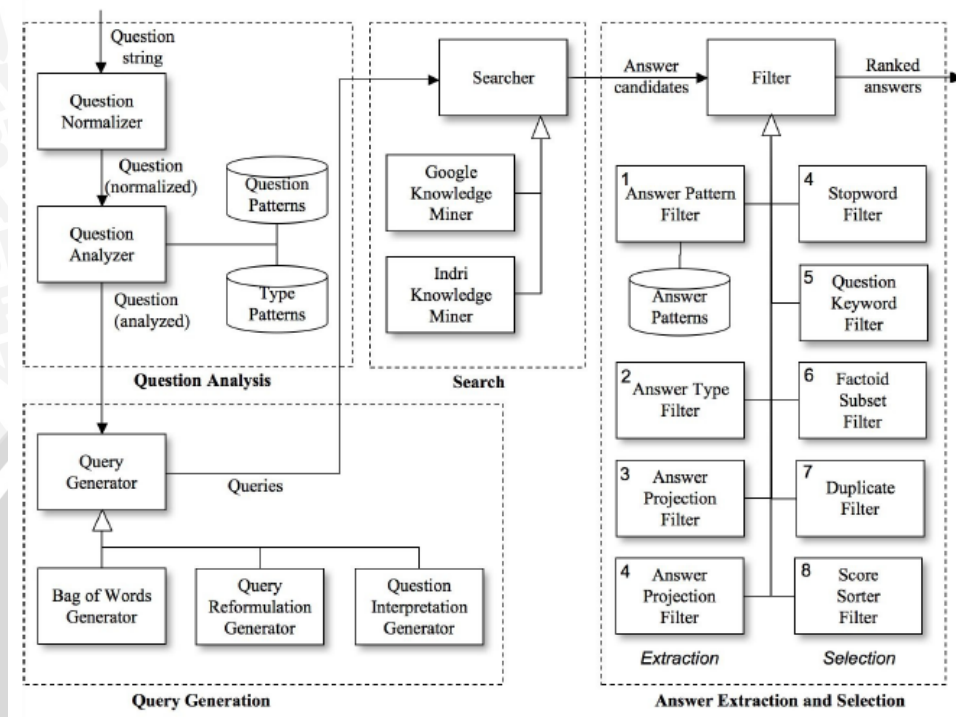
merespon dalam bahasa yang mirip dengan bahasa *natural*. Setelah komputer bisa memahami perintah dalam bahasa natural, maka diharapkan sistem komputer juga dapat memberikan respon dalam bahasa natural pula. [FAT-09]

Natural Language processor tidak memperdulikan bagaimana suatu kalimat diinputkan ke komputer. Tugasnya adalah mengekstrak informasi dari kalimat. *NLP* tidak bisa digunakan sendirian, kecuali dalam riset. Tetapi *NLP* dapat menyediakan *front end* untuk program komputer yang lain-terutama *database manager* dan *generalized problem solver*. Juga untuk menjalankan system operasi, banyak riset yang mengarah ke *NLP driven Operating System*. Dan yang paling utama adalah pada pengembangan robotika, dimana dituntut interaksi yang efektif antara mesin dan manusia. [FAT-09]

2.3 Question Answering System

Question Answering System merupakan sebuah sistem yang secara otomatis menjawab suatu pertanyaan yang diajukan dalam bahasa alami (*natural language*). Untuk memperoleh jawaban dari pertanyaan, *Question Answering System* dapat menggunakan basisdata ataupun koleksi dokumen dalam bahasa alami. Hasil kembalian dari *Question Answering System* adalah berupa kutipan teks singkat atau bahkan frase sebagai jawabannya.

Question Answering System merupakan kombinasi antara *Information Retrieval (IR)* dengan *Natural Language Processing (NLP)*. *Question Answering System* memiliki tujuan menampilkan jawaban berdasarkan *query* dalam bentuk pertanyaan yang diajukan oleh pengguna [ARM-11]. Perbedaan yang mendasar antara *Question Answering System* dengan *IR* terletak pada masukan (*query*) dan keluaran yang dihasilkan. Pada *IR query* yang dimasukkan berupa kata atau kalimat pertanyaan dan keluaran yang dihasilkan adalah dokumen yang dianggap relevan oleh sistem. Sedangkan pada *Question Answering System*, *query* berupa kalimat tanya dan keluarannya berupa jawaban (entitas) yang dianggap sesuai oleh sistem sehingga memungkinkan sistem tidak mengembalikan jawaban apapun.



Gambar 2.2 Arsitektur *Question Answering System* pada *OpenEphyra*

Sumber: [SCH-07]

2.3.1 Arsitektur *Question Answering System*

Question Answering System yang dikembangkan dengan tujuan, sumber informasi, dan teknik yang berbeda dapat memiliki arsitektur yang berbeda pula. Gambar 2.2 menunjukkan arsitektur dari *Question Answering System* pada *OpenEphyra*. *OpenEphyra* adalah *framework open source* yang digunakan untuk membangun *Question Answering System*. *Question Answering System* pada *OpenEphyra* memiliki empat bagian utama, yaitu *Question Analysis*, *Query Generation*, *Search*, dan *Answer Extraction and Selection*. [SCH-07]

2.3.2 *Question Analysis*

Pertanyaan yang diinputkan oleh pengguna seringkali tidak sesuai dengan aturan dan persyaratan yang ditentukan. Hal ini dapat mengakibatkan sistem sulit mengenali maksud dari pertanyaan yang diajukan. Oleh karena itu diperlukan

proses yang mengubah pertanyaan menjadi format yang dapat dikenali oleh *Question Answering System*.

Question Analysis memproses input pertanyaan dari pengguna menjadi pertanyaan yang memenuhi syarat-syarat yang telah ditentukan. Bagian ini mengandung dua buah proses, yaitu *Question Normalizer* dan *Question Analyzer*.

2.3.3 Query Generation and Search

Representasi pertanyaan perlu diubah menjadi *string query* yang untuk selanjutnya *query* tersebut digunakan untuk mencari dokumen yang relevan. Pada *OpenEphyra* digunakan tiga buah metode *query* generator, yaitu *Bag of Words*, *Query Reformulation*, dan *Question Interpretation Generator*. Sedangkan pada *search engine* digunakan *Google Knowledge Miner* dan *Indri Knowledge Miner*.

2.3.4 Answer Extraction and Selection

Dokumen-dokumen yang dikembalikan oleh *search engine* dianalisis lebih lanjut, yaitu dengan membuang dokumen yang tidak memenuhi kriteria yang ditentukan. Hal ini dilakukan dengan memproses dokumen-dokumen ke dalam *filter* yang telah ditentukan. *Filter* tersebut juga memberikan skor atau nilai untuk dapat melakukan *ranking* pada dokumen sesuai dengan tingkat relevansinya. Hasil yang paling tinggi skornya adalah kandidat yang paling mungkin untuk menjadi jawaban dari pertanyaan yang diajukan.

2.4 Pattern Based Approach

OpenEphyra adalah salah satu contoh *Question Answering System* yang menggunakan *Pattern Based Approach* untuk penggolongan pertanyaan. *OpenEphyra* dapat mempelajari pasangan pertanyaan-jawaban dan menggunakan *IR System* standar untuk mengambil potongan teks yang sesuai untuk ekstraksi pola. Dari hasil pengujian metode *Pattern Based Approach* pada *Question Answering* berbahasa Indonesia didapatkan hasil akurasi interpretasi sebesar 90% dan akurasi jawaban sebesar 35 % [TOB-10].

Dua tahap utama dalam *Pattern Based Approach* yaitu

1. Pertama adalah dengan mempelajari pola pertanyaan dari *template* pertanyaan sesuai dengan tipe pertanyaannya. Tujuannya adalah untuk menginterpretasikan pertanyaan dan mengubahnya ke dalam bentuk *query*. Template pertanyaan ini dikembangkan secara *manual* yang independen untuk setiap *natural language*.
2. Langkah kedua adalah dengan mempelajari pola jawaban dari pasangan pertanyaan-jawaban. Tujuannya adalah untuk mengekstrak kandidat jawaban dari dokumen yang relevan dan mengurutkannya.

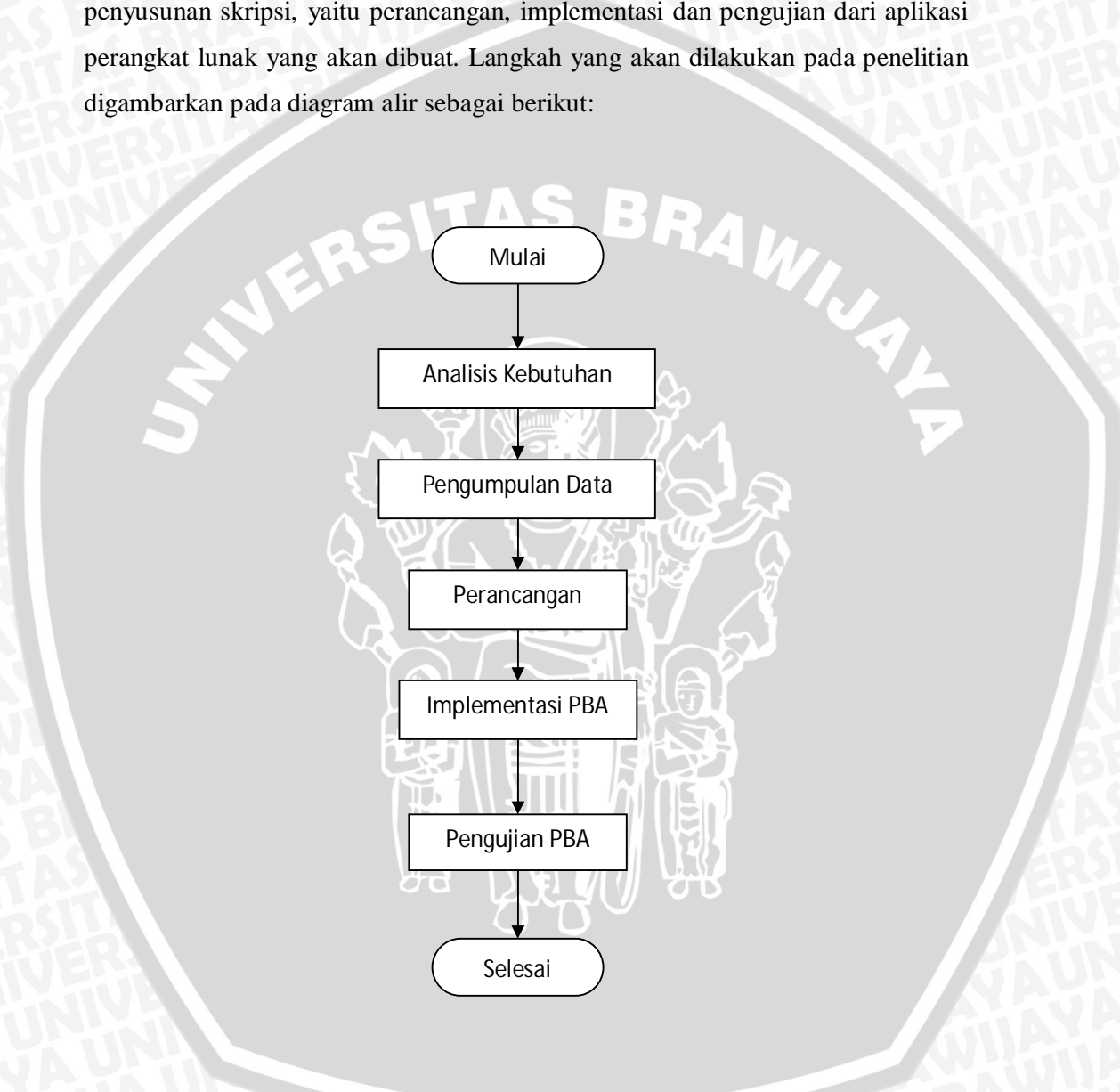
2.5 Stemming Arifin

Stemmin Arifin adalah metode *stemming* yang dikembangkan oleh Arifin dan Setiono. Metode ini lebih sederhana jika dibandingkan dengan metode Nazief dan Adriani, tetapi tetap menggunakan pendekatan yang sama dengan menggunakan kamus, dan secara progresif menghilangkan imbuhan. Pendekatan ini berusaha menghilangkan awalan dan dilanjutkan dengan akhiran, dan berhenti bila kata hasil *stemming* ditemukan pada kamus atau jumlah imbuhan yang dihilangkan mencapai maksimum 2 awalan dan 3 akhiran. Jika kata tidak dapat ditemukan setelah semua awalan dan akhiran dihilangkan, imbuhan akan dikembalikan pada kata dalam semua kombinasi yang memungkinkan sehingga kemungkinan kesalahan pada *stemming* dapat diminimalisir. Pada pengujian *stemming* menggunakan metode Arifin pada kata-kata dalam bahasa Indonesia didapatkan tingkat kebenaran sebesar 87,7% [ASI-05].

BAB III

METODOLOGI PENELITIAN

Pada bab ini menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan skripsi, yaitu perancangan, implementasi dan pengujian dari aplikasi perangkat lunak yang akan dibuat. Langkah yang akan dilakukan pada penelitian digambarkan pada diagram alir sebagai berikut:



Gambar 3.1 Diagram alir metode penelitian

Sumber: Metode Penelitian

3.1 Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Kebutuhan sistem yang akan dibuat meliputi perangkat keras dan perangkat lunak.

Aplikasi yang akan dibuat adalah aplikasi untuk menjawab pertanyaan (*Question Answering*) berbahasa Indonesia menggunakan pendekatan berbasis pola (*Pattern Based*). Pembuatan aplikasi ini menggunakan bahasa pemrograman *php* yang terhubung pada *database*. *Database* ini digunakan untuk menyimpan dokumen-dokumen yang akan menjadi jawaban serta pola-pola yang digunakan untuk pencarian jawaban yang tepat. Dari gambaran tersebut maka pada pembuatan aplikasi dibutuhkan komputer dan perangkat lunak berupa *Apache* dan *database MySQL* sebagai media untuk membuat aplikasi.

3.2 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data-data yang dibutuhkan dalam *Question Answering System* yaitu berupa kumpulan dokumen yang mengandung jawaban dari pertanyaan yang diajukan. Dokumen tersebut berupa kumpulan teks yang didapatkan dari Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer Tahun Akademik 2012/2013 dan disimpan di dalam *database*. Teks tersebut dipisah-pisah untuk tiap topik bahasan, jadi sebuah *record* data pada tabel *database* hanya memuat satu topik bahasan. Untuk isi dokumen yang tidak berupa teks, misalkan dalam bentuk tabel, ditransformasi dahulu menjadi bentuk paragraf sebelum dimasukkan ke dalam *database*.

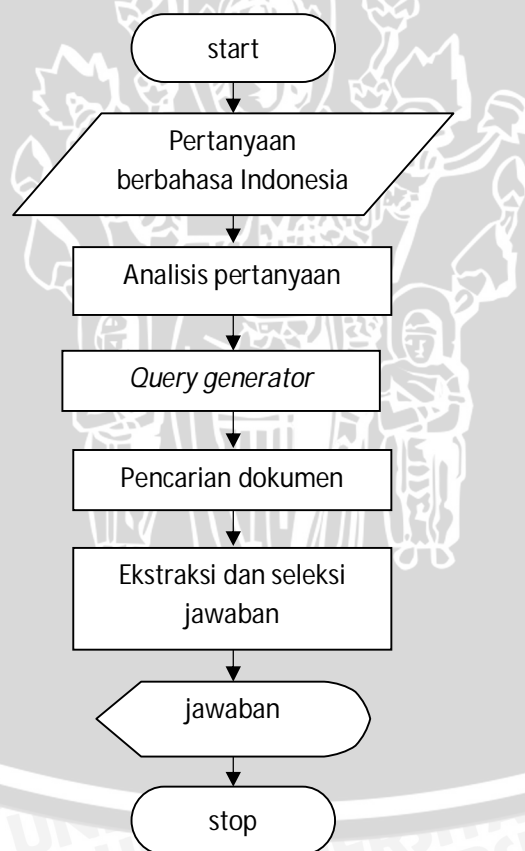
3.3 Perancangan Sistem

Perancangan sistem dilakukan setelah semua kebutuhan sistem didapatkan melalui tahap analisis kebutuhan. Perancangan meliputi Arsitektur Sistem dan Perancangan Algoritma.

Hasil dari analisis kebutuhan sistem dapat membantu merancang bagaimana aplikasi berjalan, adapun rancangan yang didapat sebagai berikut:

1. Pengguna memasukkan pertanyaan berbahasa Indonesia melalui *user interface* yang berbasis *web*.
2. Sistem menganalisis pertanyaan yang dimasukkan pengguna untuk mengekstrak informasi penting yang terdapat pada pertanyaan.
3. Sistem mengubah informasi hasil analisis menjadi *query*.
4. Sistem menggunakan *query* untuk mencari dokumen-dokumen yang akan menjadi kandidat jawaban dengan menggunakan *search engine*.
5. Sistem melakukan *filter* dan pembobotan pada kandidat jawaban sehingga dapat diambil jawaban yang dinilai paling tepat.
6. Sistem menampilkan *output* berupa jawaban.

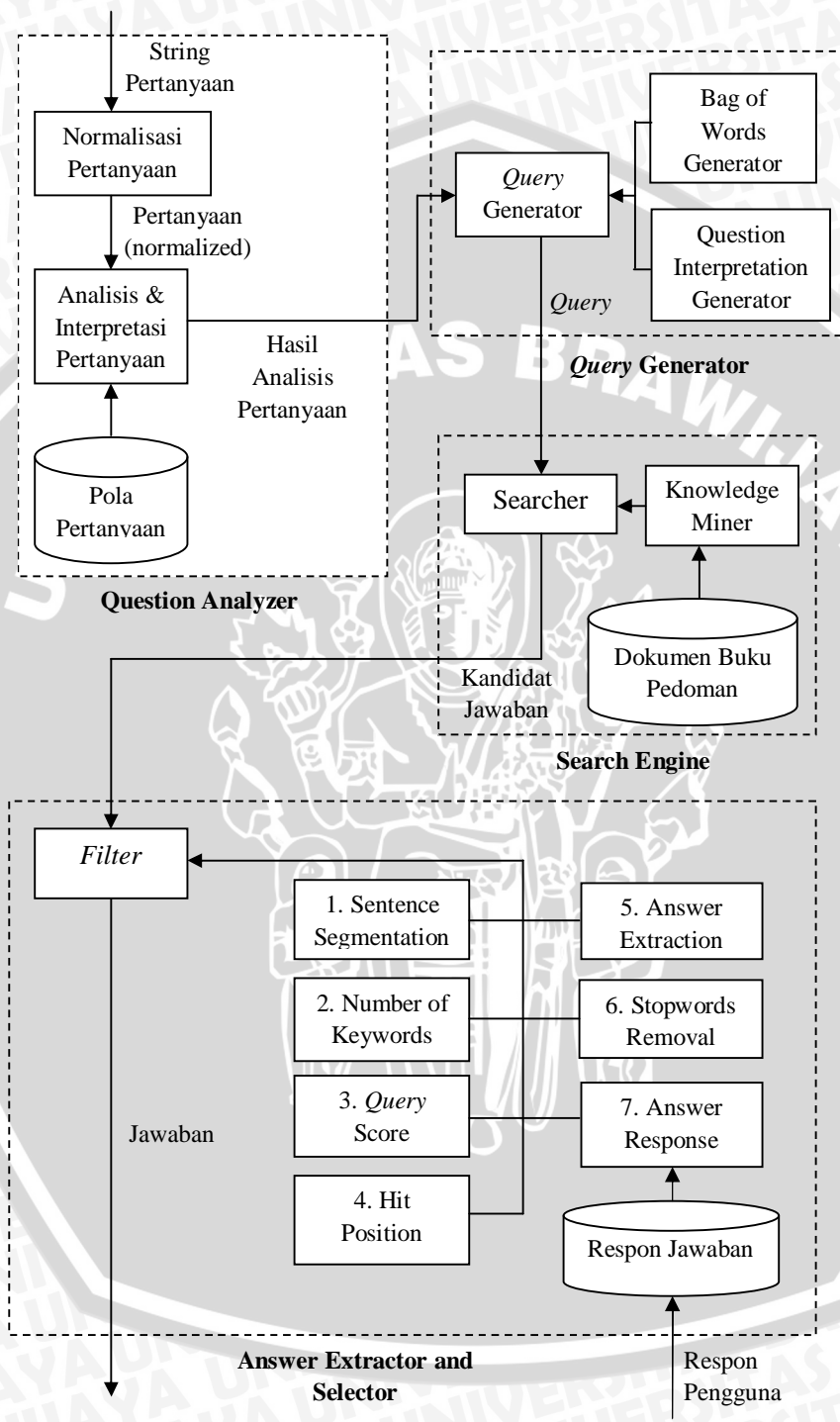
Untuk mempermudah alur jalan dari sistem, maka semua proses yang dijelaskan sebelumnya digambarkan dalam diagram alir berikut:



Gambar 3.2 Diagram Alir Sistem

Sumber: Metode Penelitian

Gambaran Arsitektur *Question Answering System* yang dikembangkan dapat dilihat pada Gambar 3.3



Gambar 3.3 Arsitektur *Question Answering System*

Sumber: Perancangan

3.4 Implementasi

Implementasi aplikasi dilakukan dengan mengacu kepada perancangan aplikasi. Dari gambar arsitektur sistem pada Gambar 3.3 dapat dilihat *Question Answering System* dengan pendekatan berbasis pola (*Pattern Based Approach*) terdiri dari empat komponen utama, yaitu: *Question Analyzer*, *Query Generator*, *Search engine*, dan *Answer Extractor*.

3.4.1 *Question Analyzer*

Question Analyzer menerima *input* berupa *string* pertanyaan. Selanjutnya dilakukan normalisasi pada *string* pertanyaan dimasukkan, seperti dengan menghilangkan tanda baca, dan mengganti huruf capital dengan huruf kecil.

Pertanyaan yang sudah dinormalisasi kemudian dianalisis dengan menginterpretasikannya. Pertanyaan diinterpretasikan dengan memanfaatkan pola pertanyaan (*Question Patterns*) yang sudah dipersiapkan. Tujuan tahap analisis ini adalah untuk memperoleh properti, target dan konteks dari pertanyaan. Pada dasarnya setiap pertanyaan menanyakan tentang sebuah properti dari target pada suatu konteks yang spesifik

Contohnya pada pertanyaan “*Kapan Indonesia merdeka?*” dapat ditafsirkan sebagai berikut:

- **Properti:** *TIME* (waktu)
- **Target:** merdeka
- **Konteks:** Indonesia

3.4.2 *Query Generator*

Hasil dari *Question Analyzer* digunakan oleh *Query Generator* untuk memperoleh *query* yang selanjutnya akan digunakan pada *search engine*. Ada dua metode yang digunakan untuk mendapatkan *query*, yaitu *Bag of Words* dan *Question Interpretation Generator*.

3.4.3 *Search engine*

Search engine mencari dokumen-dokumen yang relevan dengan jawaban yang diinginkan dari pertanyaan dengan menggunakan *query* yang dihasilkan

query generator. Dokumen-dokumen yang diperoleh akan menjadi kandidat jawaban dan masih perlu melalui beberapa proses untuk menjadi jawaban yang diinginkan.

3.4.4 *Answer Extractor*

Kandidat jawaban harus melewati beberapa *filter* sebelum diperoleh jawaban yang diinginkan dari pertanyaan. *Filter* yang digunakan antara lain:

1. *Sentence Segmentation*

Dokumen hasil *search engine* dipecah menjadi kalimat-kalimat dan kalimat tersebut menjadi kandidat jawaban yang baru. Tiap kandidat jawaban diberi nilai/ skor awal nol.

2. *Number of Keywords*

Kandidat jawaban dibuang jika tidak memenuhi kondisi sebagai berikut [SCH-05]

$$M \geq \lceil \sqrt{K-1} \rceil + 1 \quad (3-1)$$

Dimana K adalah jumlah kata kunci dalam pertanyaan dan M adalah jumlah kata kunci yang juga terdapat dalam *string* kandidat jawaban. Jika memenuhi kondisi ini, sebuah kandidat jawab akan mendapat skor sejumlah nilai M.

3. *Query Score*

Skor diberikan kepada *query* oleh yang dihasilkan *query generator*. Semakin tinggi skornya, semakin spesifik pula *query*, yang berarti dokumen yang didapatkan dari *query* tersebut lebih relevan. Oleh karena itu skor diberikan kepada dokumen sesuai dengan skor dari *query*

4. *Hit Position*

Search engine biasanya mengembalikan hasil pencarian yangurut berdasarkan tingkat ketepatannya. Urutan ini yang menjadi skor untuk kandidat dimana kandidat teratas mendapat skor tertinggi dan skor akan menurun untuk kandidat seterusnya.

5. Answer Extraction

Filter ini memanfaatkan atribut yang diperoleh pada *question analyzer*, yaitu properti, target, dan konteks. Sebuah kandidat jawaban adalah potongan teks yang mengandung objek target dan konteks dari pertanyaan. Objek target dan konteks tersebut kemudian diganti dengan *tag* <T> dan <C>.

Contohnya pada pertanyaan “Kapan Indonesia merdeka?” kita telah mendapatkan target, properti, dan konteks. Potongan Teks:

“Belanda akhirnya mengakui bahwa kemerdekaan Indonesia adalah tanggal 17 Agustus 1945”

Akan diubah menjadi:

“Belanda akhirnya mengakui bahwa <T> <C> adalah tanggal 17 Agustus 1945”

Pola jawaban (*answer pattern*) digunakan untuk menentukan jawaban yang terdapat pada potongan teks yang sudah diubah tersebut. Untuk setiap properti yang ditanyakan, terdapat file terpisah yang berisikan serangkaian pola jawaban. Pola-pola jawaban ini adalah serangkaian kata-kata yang mengandung *tag* target <T> dan *tag* properti <P>.

Pada contoh pertanyaan, properti yang dicari adalah *TIME* atau tanggal. Pada pola jawaban untuk properti *TIME* dicari pola yang sesuai dengan potongan teks. Misalkan ditemukan pola jawaban yang sesuai “<T> <C> adalah tanggal <P>”. Maka akan diambil objek yang merepresentasikan *tag* <P> yaitu “17 Agustus 1945”.

Jika *string* yang diekstrak belum ada pada kandidat jawaban, maka *string* tersebut akan menjadi kandidat jawaban yang baru dan diberi skor awal sesuai dengan nilai *confidence* dari pola (*pattern*). Sedangkan jika sudah ada, maka skor kandidat tersebut ditambahkan dengan nilai *confidence* dari *string* yang baru saja diekstrak.

6. *Stopwords*

Filter ini diterapkan pada semua kandidat jawaban dan membuang kandidat jika *string* jawaban mengandung kata umum (*common words*) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Misalnya: “dia”, “mereka”, “disini”, “itu”.

7. *Answer Response*

Setiap pengguna yang mengajukan pertanyaan ke sistem akan diminta memberikan respon tentang jawaban yang dihasilkan oleh *Question Answering System*, apakah sudah sesuai dengan yang diinginkan pengguna atau tidak. Pengguna akan diberikan pilihan: “Sesuai”, “Ragu-ragu/tidak tahu”, dan “Tidak sesuai” mengenai hasil jawaban. Respon dari pengguna ini akan disimpan oleh sistem dan dapat digunakan oleh sistem dalam menentukan jawaban dari pertanyaan yang serupa jika sewaktu-waktu pertanyaan tersebut diajukan lagi.

Respon dari pengguna ini akan berpengaruh pada pemberian skor atau nilai pada kandidat jawaban. Jika ternyata responnya adalah “sesuai”, maka akan diberikan skor +1, jika “Ragu-ragu/tidak tahu” diberikan skor +0, sedangkan jika “Tidak sesuai” maka diberikan skor -1. Skor ini akan terus terakumulasi untuk setiap respon pengguna terhadap jawaban dari pertanyaan yang sama, jadi jika ada dua pengguna memberikan respon “Tidak sesuai” untuk suatu jawaban yang sama maka skornya terakumulasi menjadi -2.

Jawaban akhir yang ditampilkan kepada pengguna adalah kandidat jawaban yang memiliki skor tertinggi setelah melalui proses *filtering* dan *scoring* tersebut.

3.4.5 *Pattern Learning*

Proses interpretasi pertanyaan dan ekstraksi jawaban membutuhkan suatu pola teks yang harus dipersiapkan lebih dahulu pada sistem. Pola Pertanyaan (*Question Patterns*) digunakan untuk menginterpretasikan pertanyaan seperti untuk menentukan properti dan untuk mendapatkan target dan konteks dari

pertanyaan. Pola Jawaban (*Answer Patterns*) digunakan untuk mengekstrak jawaban dari sebuah teks kalimat. Bagian ini akan menjelaskan bagaimana kedua pola tersebut dibentuk.

1. *Question Patterns*

Sistem ini menggunakan 8 tipe properti pertanyaan dan tiap propertinya memiliki 1 sampai 20 pola pertanyaan. Tipe properti ini diambil berdasarkan hasil dari CLEF 2008 [FOR-08] tiap properti dan pola pertanyaan yang mewakilinya ditentukan secara *manual*. Tipe-tipe properti tersebut yaitu *PEOPLE* (Orang), *TIME* (Waktu), *LOCATION* (Tempat), *ORGANIZATION* (Organisasi), *MEASURE* (Ukuran), *COUNT* (Angka), *OBJECT* (Obyek), dan *OTHER* (Lain).

Pada pengembangan setiap pola pertanyaan, langkah-langkah yang dilakukan [TOB-10] adalah:

- a. Menentukan kata tanya utama untuk setiap tipe. Seperti untuk tipe *LOCATION* (Tempat) ditentukan kata tanya utamanya adalah "dimana".
- b. Menentukan dimana letak kata kunci dari pertanyaan, yaitu target $\langle T \rangle$ dan konteks $\langle C \rangle$ dari pertanyaan.
- c. Kata atau frase lain yang dapat memberikan arti khusus bagi sebuah tipe pertanyaan.

2. *Answer Patterns*

Pola jawaban (*Answer Patterns*) dibentuk secara otomatis dan terbagi menjadi dua tahap. Tahap pertama adalah dengan mengekstrak pola jawaban (*Pattern Extraction*) dari potongan teks. Tahap selanjutnya adalah dengan memberikan nilai pada pola jawaban (*Pattern Assessment*) dan membuang pola yang tidak relevan dari hasil penilaian tersebut.

a. *Pattern Extraction*

Proses pembelajaran pola jawaban dilakukan dengan menggunakan kumpulan pasangan pertanyaan-jawaban sebagai *training set*. Setiap *training set* diinterpretasikan menggunakan *Question Patterns* yang

sebelumnya sudah didapatkan untuk mendapatkan properti, target dan konteks dari pertanyaan.

b. Pattern Assessment and Filtering

Tujuan pada tahap ini adalah untuk membuang pola jawaban yang dinilai terlalu spesifik atau yang kurang relevan. Seperti pada langkah sebelumnya, hasil interpretasi training set digunakan untuk mendapatkan *query* untuk *search engine*, dan potongan teks hasil pencarian diubah melalui mengganti kata atau frase dengan *tag* properti, target, dan konteks yang sesuai.

Hasil dari proses ini kemudian diaplikasikan pada pola jawaban hasil dari *Pattern Extraction* sebelumnya sesuai dengan tipe properti masing-masing. Untuk setiap pola pertanyaan a , akan dicatat seberapa sering pola ini dapat digunakan untuk mengekstrak jawaban yang benar ($\#correct_a$) dan seberapa sering didapatkan jawaban yang salah ($\#incorrect_a$). Kemudian untuk setiap tipe properti p , jumlah potongan teks juga dicatat ($\#snippets_p$). Nilai ini digunakan untuk menghitung pengukuran berikut [TOB-10]

$$Confidence_a = \frac{\#correct_a}{\#correct_a + \#incorrect_a}$$

$$Support_a = \frac{\#correct_a}{\#snippets_p}$$

(3-2)

Hasil pengukuran tersebut kemudian dibandingkan dengan nilai *threshold*. Pola dengan nilai *confidence* yang rendah dinilai kurang dapat diandalkan sehingga dibuang. Sedangkan pola dengan nilai *support* rendah berarti pola tersebut terlalu spesifik dan juga akan dibuang.

3.5 Pengujian

Pengujian merupakan tahap dilakukan pengujian berdasarkan implementasi yang telah dibuat melalui pengujian perangkat lunak dengan pengujian akurasi tingkat kebenaran jawaban pada *Question Answering System*.

Pengujian akurasi dilakukan dengan menggunakan 50 sampel pertanyaan dan setiap tipe properti minimal diwakilkan dengan satu pertanyaan. Sampel pertanyaan yang digunakan sebanyak 50 agar dapat mencakup pertanyaan-pertanyaan yang sering ditanyakan mengenai isi buku pedoman. Jawaban yang dikeluarkan sistem dibandingkan dengan jawaban sebenarnya yang sesuai dengan isi Buku Pedoman Pendidikan. Dari hasil pengujian tersebut dihitung nilai presisi dan *recall* dari pengujian.

Tabel 3.1 Klasifikasi Kelas pada Perhitungan Presisi dan *Recall*

		<i>Actual Class</i> (<i>Expectation</i>)	
		+	-
<i>Predicted Class</i> (<i>Observation</i>)	+	<i>TP</i> (<i>True Positive</i>)	<i>FP</i> (<i>False Positive</i>)
	-	<i>FN</i> (<i>False Negative</i>)	<i>TN</i> (<i>True Negative</i>)

Sumber: Metode Penelitian

Presisi (P) didapatkan dengan perhitungan sebagai berikut:

$$P = \frac{TP}{TP+FP} \tag{3-3}$$

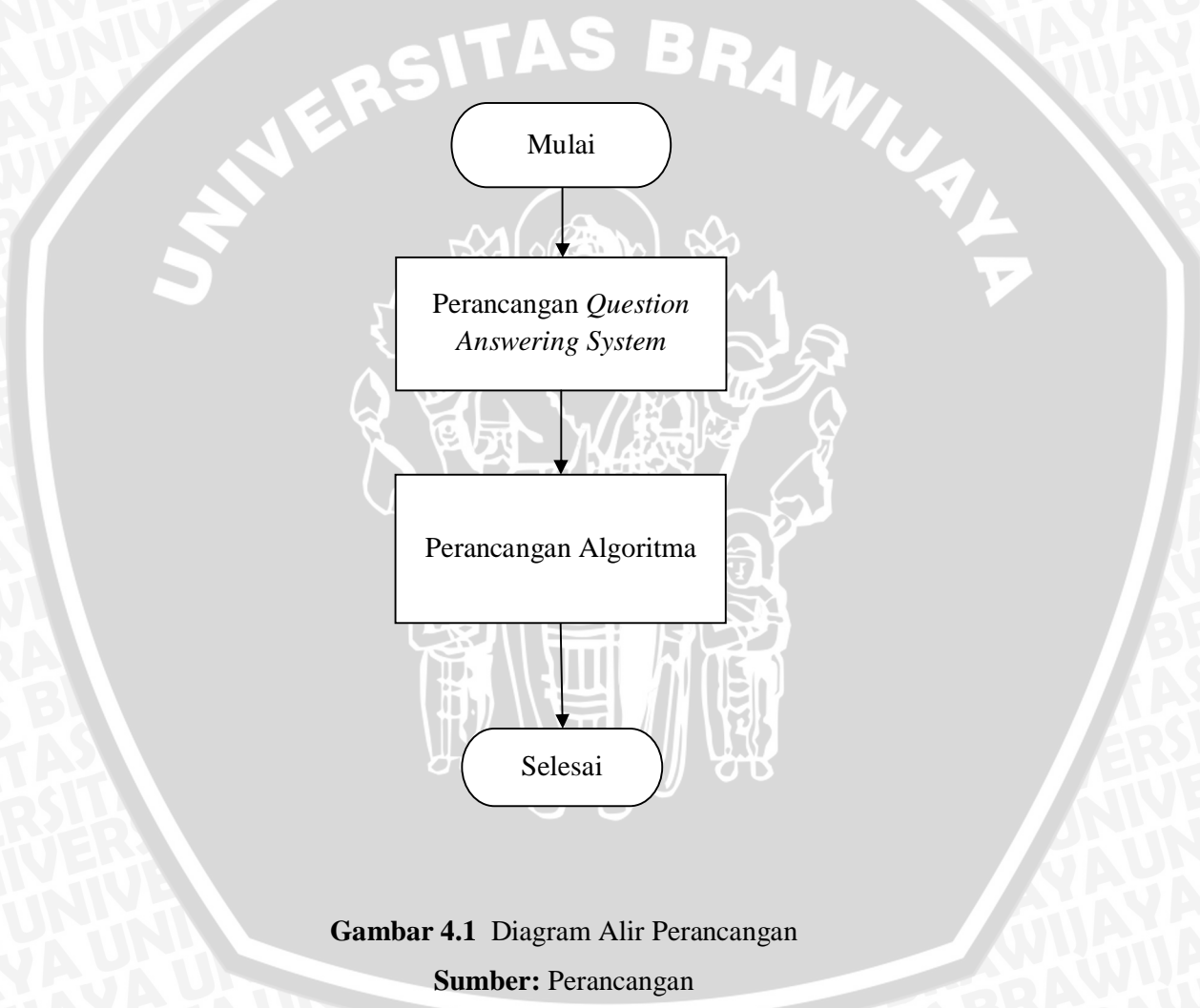
Recall (R) didapatkan dengan perhitungan sebagai berikut:

$$R = \frac{TP}{TP+FN} \tag{3-4}$$

BAB IV

PERANCANGAN

Bab ini menjelaskan perancangan *Question Answering System* dengan menggunakan metode *Pattern Based Approach* sebagai produk akhir skripsi. Perancangan *Question Answering System* ini terdiri dari dua bagian yaitu Perancangan *Question Answering System* dan Perancangan Algoritma seperti yang dapat dilihat pada gambar diagram alir pada Gambar 4.1.



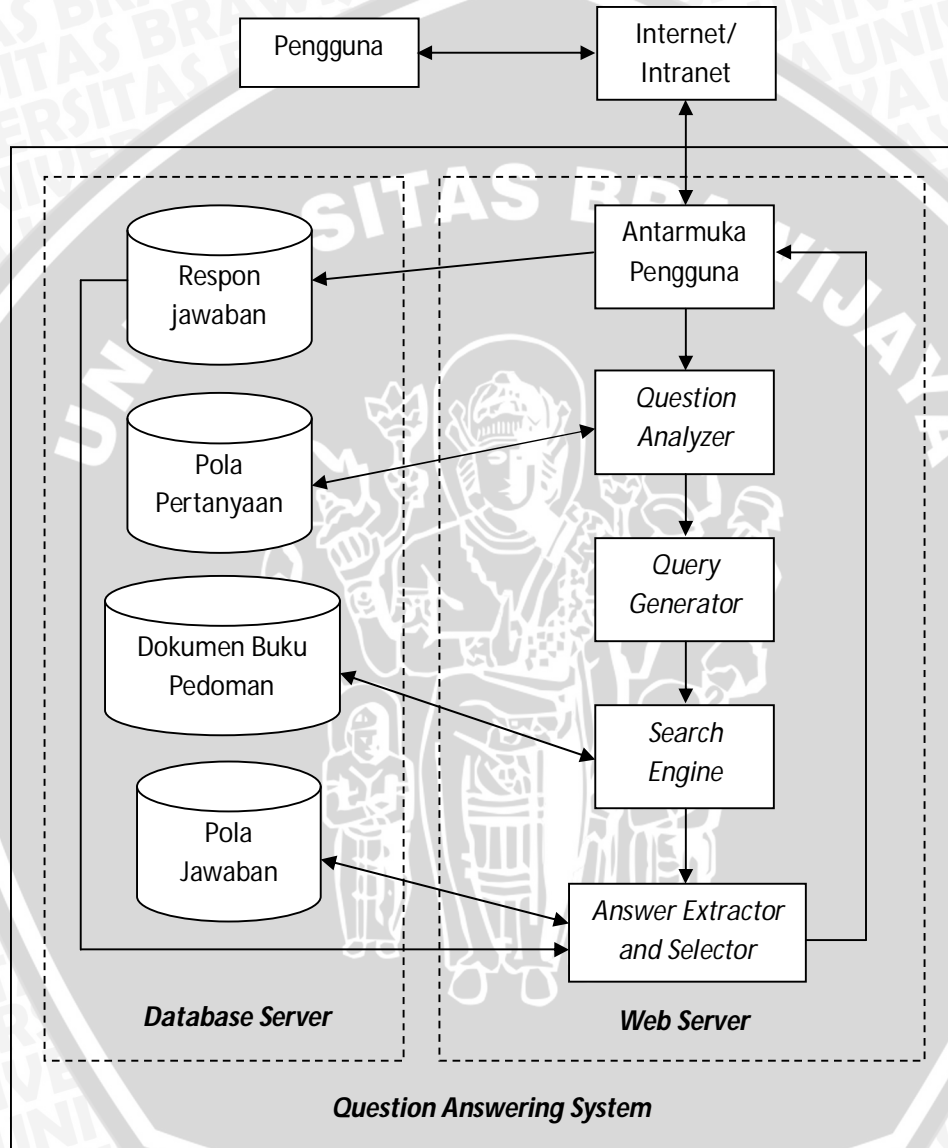
Gambar 4.1 Diagram Alir Perancangan

Sumber: Perancangan

4.1 Perancangan *Question Answering System*

Perancangan *Question Answering System* terbagi menjadi 3 bagian, yaitu *System Plan*, *Arsitektur Sistem* dan Perancangan *database*.

4.1.1 *System Plan Question Answering System*



Gambar 4.2 *System Plan Question Answering System*

Sumber: Perancangan

Pada Bagian ini dijelaskan *system plan* dari *Question Answering System* yang dirancang. Gambaran *system plan Question Answering System* yang dikembangkan dapat dilihat pada gambar 4.2

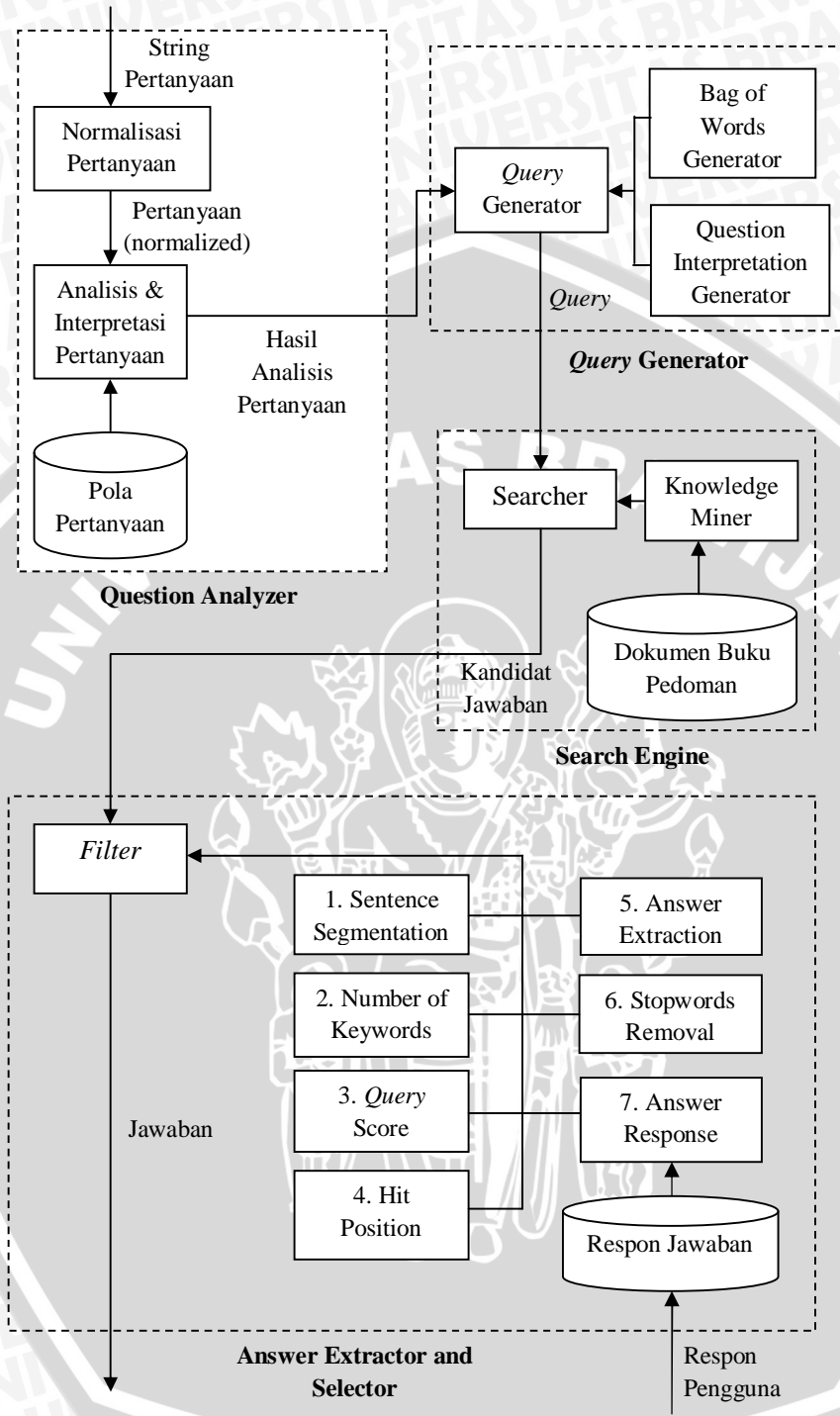
Question Answering System yang dirancang merupakan sistem yang berbasis *web* dan terhubung pada suatu *database*. Pertama pengguna memasukkan pertanyaan pada antarmuka pengguna yang berbasis *web*. Kemudian pertanyaan dilanjutkan pada *Question Analyzer*. *Question Analyzer* mengakses *database* untuk mendapatkan pola pertanyaan yang tersimpan. Dengan pola pertanyaan yang didapat dilakukan interpretasi pada pertanyaan. Hasil interpretasi pada *Question Analyzer* dilanjutkan ke *Query Generator* untuk dihasilkan *query* yang digunakan pada pencarian dokumen. Pencarian dokumen pada *Search Engine* dilakukan dengan mencari dokumen yang terdapat pada dokumen buku pedoman yang terdapat pada *database* untuk didapatkan dokumen yang relevan. Selanjutnya dokumen yang relevan diproses pada *Answer Extractor and Selector*. Pada *Answer Extractor and Selector* ini diakses pola jawaban pada *database* untuk dapat mengekstrak jawaban dari dokumen. Jawaban yang dihasilkan kemudian ditampilkan pada pengguna melalui antarmuka pengguna.

4.1.2 Arsitektur *Question Answering System*

Pada Bagian ini dijelaskan arsitektur dari sistem yang dirancang. Gambaran Arsitektur *Question Answering System* yang dikembangkan dapat dilihat pada Gambar 4.3. Arsitektur *Question Answering System* terdiri dari 4 komponen utama, yaitu *Question Analyzer*, *Query Generator*, *Search Engine* dan *Answer Extractor and Selector*.

1. *Question Analyzer*

Question Analyzer menerima *input* dari pengguna berupa *string* pertanyaan. *Input* berupa *string* pertanyaan harus diolah terlebih dahulu di dalam *Question Analyzer*. Pengolahan tersebut terbagi dalam dua tahap, pertama dengan melakukan normalisasi pada *string* pertanyaan, dan kemudian dilanjutkan dengan analisis terhadap pertanyaan dengan menggunakan pola pertanyaan.



Gambar 4.3 Arsitektur Sistem

Sumber: Perancangan

A. Normalisasi Pertanyaan

Tahap-tahap normalisasi pertanyaan adalah sebagai berikut:

- 1.) Menghilangkan tanda baca (koma, titik, tanda tanya, dsb).

Proses menghilangkan semua tanda baca serta karakter atau simbol-simbol lainnya dan hanya akan menyisakan huruf dan angka saja pada *string* pertanyaan.

- 2.) Mengganti huruf kapital dengan huruf kecil.

Proses ini mengganti semua huruf kapital yang terdapat pada *string* pertanyaan dengan huruf kecilnya.

- 3.) Mengganti beberapa kata dengan padanan katanya.

Proses ini adalah dengan mengubah kata atau bagian dari pertanyaan dengan padanan katanya, baik itu sinonim, singkatan, maupun bentuk kata bakunya.

Tahap normalisasi ini dilakukan agar setiap kata atau frase pada pertanyaan yang dimasukkan dapat dikenali dan diproses oleh sistem.

B. Analisis & Interpretasi Pertanyaan

Pada bagian ini pertanyaan yang telah melalui proses normalisasi dianalisis dan diinterpretasikan dengan pola pertanyaan. Hasil interpretasi pertanyaan ini adalah berupa properti, target dan konteks dari pertanyaan.

Daftar pola pertanyaan yang telah disiapkan pada sistem dapat dilihat pada lampiran 2. Beberapa contoh bentuk pola pertanyaan itu sendiri adalah sebagai berikut:

- (dimana) letak <T> <C>
- (dimana) <C> <T>
- (dimana) <T> <C> berada
- (dimanakah) letak <T> <C>
- (dimanakah) <T> <C> berada

Pola pertanyaan diatas adalah contoh pola pertanyaan yang mewakili tipe properti lokasi atau tempat. Kata didalam tanda kurung mewakili kata tanya untuk pertanyaan. *Tag* <T> mewakili target sedangkan <C> mewakili konteks. Target dan konteks serta tipe properti yang menyertai inilah yang menjadi hasil dari interpretasi pertanyaan.

Dari proses tersebut masih memungkinkan sebuah pertanyaan memiliki lebih dari satu interpretasi, karena \bisa saja pertanyaan cocok dengan beberapa pola pertanyaan. Untuk itu dari beberapa pola tersebut dicarilah pola pertanyaan yang paling spesifik, yaitu dengan menghitung berapa kata pada pola yang cocok dengan kata pada pertanyaan. Hasil interpretasi dari pola pertanyaan yang paling spesifik inilah yang dipakai pada proses selanjutnya.

2. *Query Generator*

Untuk dapat menghasilkan jawaban dari pertanyaan, sistem perlu mencari dokumen-dokumen yang relevan dengan pertanyaan yang diajukan. *Query Generator* berfungsi untuk menghasilkan *query* yang diperlukan dalam pencarian dokumen pada bagian *Search Engine*. Dalam menghasilkan *query* ini digunakan dua macam metode, yaitu *Bag of Words* dan *Question Interpretation Generator*.

A. *Bag of Words*

Metode ini menggunakan pertanyaan yang sudah dinormalisasi pada proses sebelumnya. Pada pertanyaan tersebut kemudian dihilangkan kata tanya. Selanjutnya dihilangkan kata-kata *stopword* yang terdapat pada pertanyaan. Kata-kata yang tersisa inilah yang akan menjadi kata kunci (*keyword*) dalam pencarian dokumen.

B. *Question Interpretation Generator*

Question Interpretation Generator menggunakan hasil interpretasi yang diperoleh dari *Question Analyzer*, yaitu target dan konteks dari pertanyaan. Target dan konteks dari pertanyaan inilah yang menjadi *keyword* dalam pencarian dokumen.

Kelebihan *Bag of Words* dibandingkan *Question Interpretation* adalah akan mendapatkan hasil pencarian yang lebih banyak, akan tetapi kekurangannya adalah lebih besar kemungkinan hasil pencarian tersebut tidak sesuai dengan yang seharusnya dicari. Sedangkan *Question Interpretation* memiliki keunggulan pada presisi atau ketepatan hasil pencarian yang diperoleh jika dibandingkan dengan *Bag of Words*, meskipun data yang diperoleh mungkin tidak sebanyak pada *Bag of Words*.

3. *Search Engine*

Search Engine mencari dokumen-dokumen yang relevan dengan jawaban yang diinginkan dari pertanyaan dengan menggunakan *query* yang dihasilkan *Query Generator*. Dokumen-dokumen yang diperoleh akan menjadi kandidat jawaban dan dokumen-dokumen ini masih perlu melalui beberapa proses sebelum didapatkan jawaban yang diinginkan.

Hasil dari ketiga proses ini akan tersimpan pada tabel pada *database* dan disertai frekuensi kemunculan setiap katanya. Frekuensi inilah yang akan menentukan dokumen mana yang lebih relevan dan lebih tinggi tingkat ketepatannya.

Dalam proses pencarian dokumen, sistem akan mendaftarkan dokumen mana saja yang memiliki kata yang terdapat pada kata kunci atau *keyword* dari hasil *query generator*. Tidak semua *keyword* harus ada pada dokumen-dokumen ini, bisa saja suatu dokumen hanya memiliki satu *keyword* saja. Sedangkan pada dokumen yang memiliki beberapa *keyword*, frekuensi pada setiap *keyword* yang ditemukan pada dokumen akan dijumlahkan dan akan menjadi skor pada dokumen tersebut untuk menentukan tingkat ketepatan dokumen.

4. *Answer Extractor and Selector*

Dokumen-dokumen hasil pencarian pada proses sebelumnya akan digunakan dalam penentuan kandidat jawaban pada proses ini. Kandidat jawaban harus melewati beberapa *filter* dan *scoring* untuk dapat menentukan jawaban yang paling tepat. *Filter* yang digunakan antara lain:

a. *Sentence Segmentation*

Dokumen hasil *Search Engine* dipecah menjadi bentuk kalimat-kalimat. Proses ini hampir mirip dengan proses *parsing*, tetapi bedanya dokumen-dokumen yang ada dipisahkan menjadi bentuk kalimat.

b. *Number of Keywords*

Proses ini membuang kalimat kandidat jawaban yang jumlah *keyword* dalam kalimat kandidat jawaban tidak memenuhi *threshold* atau suatu nilai

minimum tertentu. Hal ini dilakukan untuk membuang kalimat-kalimat yang kurang relevan dengan pencarian jawaban yang diinginkan.

c. Query Score

Proses ini memberikan skor kepada kalimat sesuai dengan *query* yang digunakan untuk mendapatkan dokumen. *Query* yang dimaksud adalah *Bag of Words* dan *Question Interpretation Generator* yang sebelumnya dijelaskan pada bagian *Search Engine*.

d. Hit Position

Proses ini memberikan skor kepada kandidat jawaban sesuai dengan total frekuensi *keyword* yang ditemukan pada dokumen darimana kalimat itu berasal.

e. Answer Extraction

Proses ini mengekstrak kata-kata atau bagian dari kalimat yang memenuhi kondisi sebagai jawaban. Untuk dapat mengekstrak jawaban digunakan pola-pola jawaban yang sebelumnya sudah disiapkan. Pola jawaban yang tersedia tercantum dalam lampiran 3.

f. Stopwords Removal

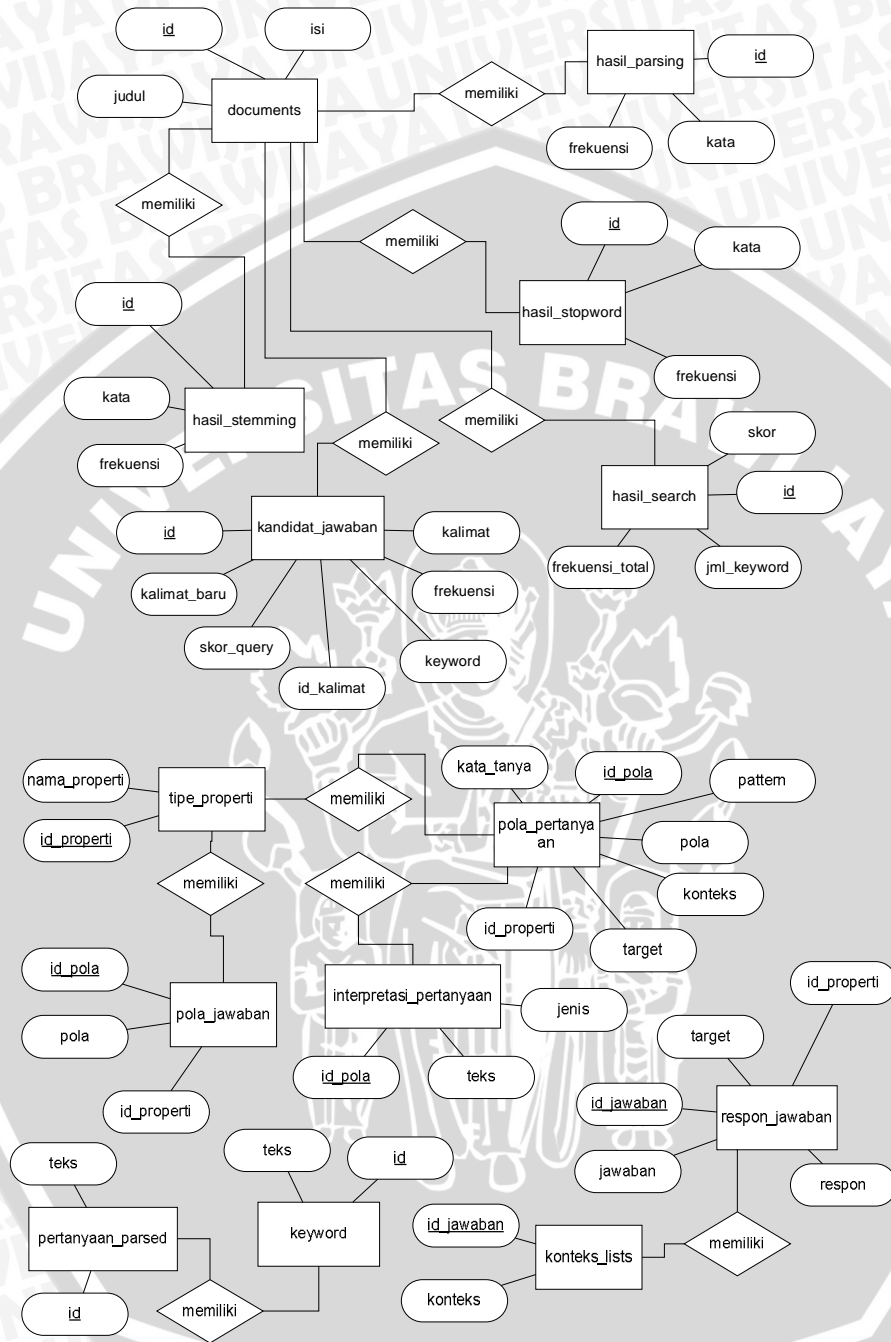
Proses ini membuang kata-kata yang termasuk dalam *stopword* yang terdapat pada kandidat jawaban. Daftar kata *stopword* tercantum dalam lampiran 1.

g. Answer Response

Proses ini memberikan atau bahkan mengurangi skor dari sebuah kandidat jawaban sesuai dengan respon dari pengguna yang sebelumnya telah menggunakan sistem ini.

4.1.3 Perancangan Database

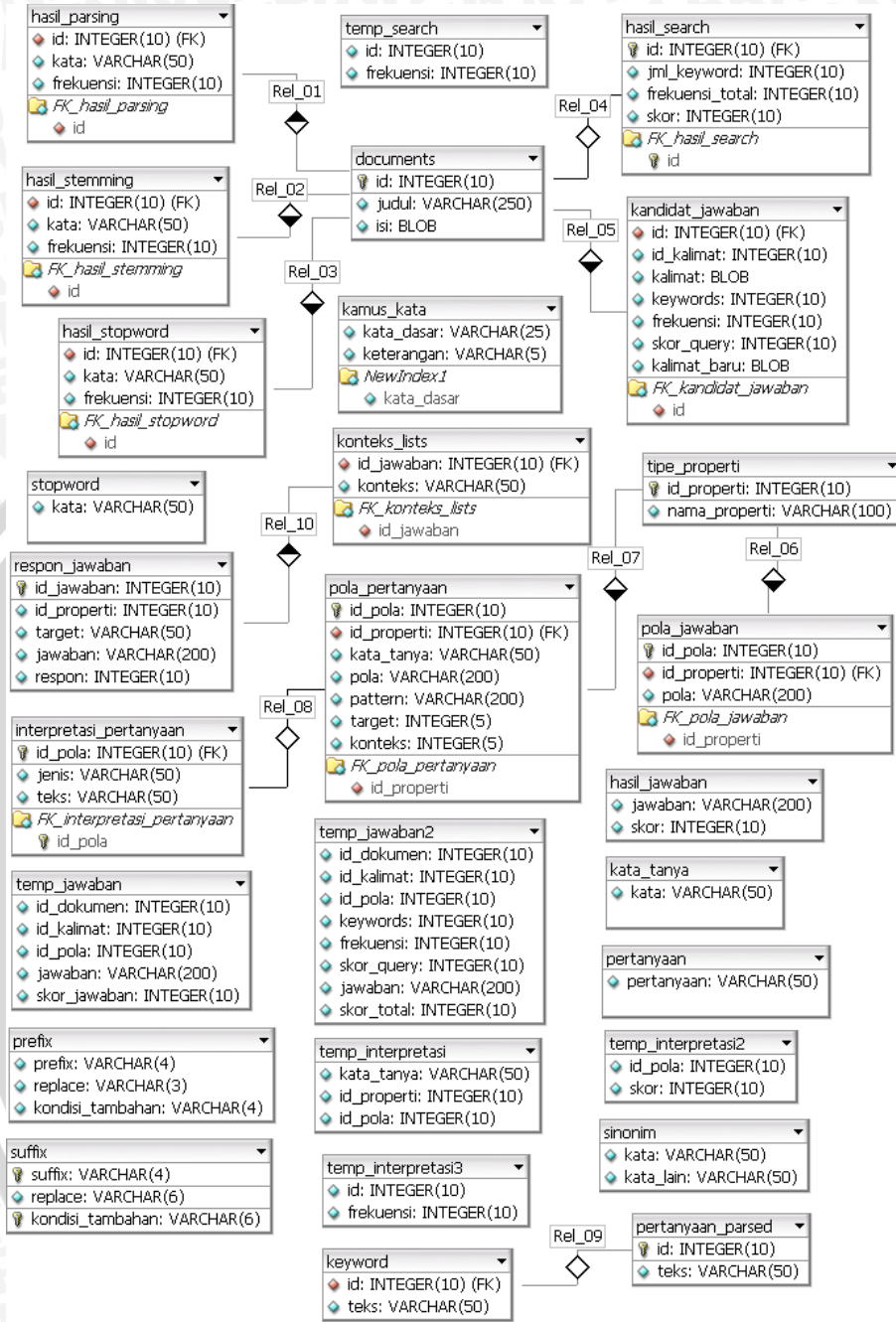
Question Answering System ini menggunakan *database* dalam menyimpan dan memproses data-data yang diperlukan dalam menghasilkan jawaban. Adapun perancangan tabel *Entity Relationship Diagram* basis data sistem ini diperlihatkan pada Gambar 4.4.



Gambar 4.4 ERD Question Answering System

Sumber: Perancangan

Implementasi dari ERD pada Gambar 4.4 dapat dilihat dalam bentuk physical diagram pada Gambar 4.5.



Gambar 4.5 Physical Diagram database

Sumber: Perancangan

4.2 Perancangan Algoritma

Pada bagian ini dijelaskan mengenai perancangan algoritma *Pattern Based Approach* yang digunakan dalam *Question Answering System*. Perancangan

algoritma ini meliputi bagaimana pengumpulan data-datanya, serta perancangan algoritma pada tiap modul-modulnya.

4.2.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data-data yang dibutuhkan oleh sistem, yang meliputi dokumen, pola pertanyaan, dan pola jawaban. Bagian ini menjelaskan bagaimana data didapatkan dan disimpan ke dalam *database*.

1. Dokumen

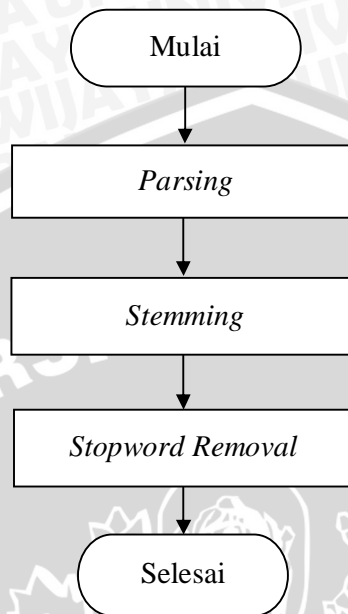
Dokumen yang digunakan dalam sistem ini berasal dari Buku Pedoman Pendidikan. Buku Pedoman yang digunakan adalah berupa *softcopy* Buku Pedoman dengan *format pdf*. Dokumen-dokumen ini dimasukkan ke dalam *database* baik secara *manual* maupun secara otomatis dengan menggunakan beberapa *tools*.

Penggunaan *tools* dipilih supaya proses memasukkan data ke dalam *database* dapat dilakukan dengan lebih cepat. Data yang dimasukkan dengan menggunakan *tools* antara lain data mengenai mata kuliah yang memuat kode mata kuliah, beban studi dari mata kuliah, dan pustaka yang digunakan dalam mata kuliah tersebut. *Tools* yang digunakan berupa aplikasi berbentuk *php* dimana pengguna mengisikan *form* yang terdapat padanya dan *tools* secara otomatis memasukkan isian tersebut ke dalam dokumen sesuai dengan bentuk dan *format* yang diinginkan.

Proses memasukkan data secara manual juga dilakukan karena beberapa dokumen tidak memungkinkan untuk dimasukkan secara otomatis ataupun dengan menggunakan *tools*. Dokumen-dokumen tersebut meliputi dokumen yang berupa gambar, tabel, maupun diagram alir, dan dokumen-dokumen lain yang bukan berupa teks. Dalam memasukkan dokumen-dokumen tersebut, pertama dokumen diubah ke dalam bentuk teks secara manual dan kemudian dimasukkan ke dalam *database*.

Ketika semua dokumen telah dimasukkan ke dalam *database*, proses selanjutnya adalah dengan melakukan *preprocessing* pada dokumen tersebut. Tahap *preprocessing* ini meliputi *parsing*, *stemming*, dan *stopword removal* yang

dilihat pada diagram alir pada Gambar 4.6. Tahap *preprocessing* ini dilakukan untuk mempermudah dalam proses pencarian dokumen.



Gambar 4.6 Diagram Alir *Preprocessing* Dokumen

Sumber: Perancangan

- a. *Parsing* adalah proses memisahkan dokumen menjadi bentuk kata. Dalam parsing ini yang digunakan untuk memisahkan antara satu kata dengan kata yang lain adalah spasi atau *whitespace* yang terletak diantara kata-kata. Kata-kata tersebut juga akan dihitung frekuensinya di dalam dokumen. Kata dan frekuensi hasil dari proses ini akan tersimpan di dalam tabel yang memuat hasil parsing.
- b. *Stemming* adalah proses menghilangkan imbuhan. Proses *stemming* ini diberlakukan pada kata-kata hasil *parsing* pada proses sebelumnya. Imbuhan yang terdapat pada kata terdiri dari *prefix* (awalan) dan *suffix* (akhiran). Langkah-langkah dalam proses *stemming* ini:
 - i. Membandingkan kata yang akan dilakukan *stemming* dengan kata yang terdapat pada tabel yang berisikan kamus kata dasar dalam bahasa Indonesia. Jika kata tersebut ada di dalam tabel kata dasar maka tidak perlu dilakukan *stemming*, sedangkan jika tidak ada maka proses *stemming* dilanjutkan ke langkah berikutnya.

- ii. Membandingkan awalan dan akhiran dengan daftar awalan dan akhiran yang terdapat pada tabel yang memuat *prefix* dan tabel yang memuat *suffix*.
 - iii. Awalan dan akhiran yang sesuai dengan isi tabel akan dihilangkan atau digantikan dengan huruf atau karakter tertentu sehingga kata berimbuhan menjadi kata dasar.
- c. *Stopword removal* adalah proses yang membuang kata-kata yang termasuk dalam *stopword*. Proses ini dilakukan dengan membandingkan kata dengan tabel yang berisi daftar kata yang termasuk ke dalam *stopword*. Jika kata tersebut ada dalam tabel *stopword*, maka kata akan dibuang. Daftar kata *stopword* terdapat pada lampiran 1.

2. Question Patterns

Pola Pertanyaan (*Question Patterns*) diperoleh dari kumpulan sampel-sampel pertanyaan yang lazim digunakan. Dengan menggunakan sampel pertanyaan tersebut, pola pertanyaan dibentuk secara manual berdasarkan delapan tipe properti yang sudah ditentukan. Tipe-tipe properti tersebut antara lain:

- *PEOPLE* (Orang)
- *TIME* (Waktu)
- *LOCATION* (Tempat)
- *ORGANIZATION* (Organisasi)
- *MEASURE* (Ukuran)
- *COUNT* (Angka)
- *OBJECT* (Obyek)
- *OTHER* (Lain)

Tahap-tahap yang dilalui dalam pengembangan setiap pola pertanyaan antara lain:

- a. Menentukan kata tanya untuk setiap pola pertanyaan. Untuk dapat menentukan kata tanya dapat dilakukan dengan membandingkan dengan tabel yang memuat seluruh kata tanya pada *database*. Contoh:
Sampel pertanyaan “Berapakah kode mata kuliah Data Mining?”

Dengan membandingkan sampel pertanyaan dengan tabel yang memuat daftar kata tanya diperoleh kata tanya dari pertanyaan ini adalah “berapakah”. Kemudian ditentukan bahwa pertanyaan ini mewakili properti *COUNT* atau angka.

- b. Menentukan dimana letak kata atau posisi kunci dari pertanyaan, yaitu target $\langle T \rangle$ dan konteks $\langle C \rangle$ dari pertanyaan. Dari sampel pertanyaan kita ubah kata yang mewakili target dengan $\langle T \rangle$ dan kata yang mewakili konteks dengan $\langle C \rangle$. Contoh: Dari sampel pertanyaan sebelumnya yang ingin ditanyakan adalah “kode” dari “Data Mining”. Oleh karena itu ditentukan “kode” adalah sebagai target sedangkan “Data Mining” sebagai konteksnya. Didapatkan: “Berapakah $\langle T \rangle$ mata kuliah $\langle C \rangle$?”
- c. Menentukan kata atau frase lain yang dapat memberikan arti khusus bagi sebuah tipe pertanyaan. Contoh: Pada sampel pertanyaan sebelumnya, frase yang memiliki arti khusus bagi pertanyaan adalah “mata kuliah”. Dari hasil pemrosesan sampel pertanyaan tersebut maka diperoleh pola pertanyaan dengan bentuk:

(berapakah) $\langle T \rangle$ mata kuliah $\langle C \rangle$

Pola-pola pertanyaan yang didapatkan dari proses ini kemudian disimpan ke dalam sebuah tabel pada *database* yang memuat kumpulan pola-pola pertanyaan dari semua tipe properti.

3. Answer Patterns

Pola pertanyaan (*Answer Patterns*) dapat dibentuk secara otomatis yaitu melalui dua tahap, yaitu *Pattern Extraction* dan *Pattern Assessment*.

a. Pattern Extraction

Pada Bagian ini disiapkan kumpulan pertanyaan yang disertai dengan jawaban sebagai *data training*. Pertanyaan tersebut kemudian di-input-kan ke sistem dan pertanyaan akan mengalami pemrosesan seperti yang dijelaskan sebelumnya, mulai dari *question analyzer*, *query generator*, *search engine*, sampai pada *answer extractor and selector*.

Hasil interpretasi dari *question analyzer* digunakan untuk membentuk *query* untuk *search engine*, yaitu dengan menggunakan target dan konteks dari

pertanyaan disertai dengan jawaban dari pertanyaan pada training set. Pada contoh pertanyaan “*Dimanakah letak Universitas Brawijaya?*” dengan jawaban “kota Malang” akan didapat hasil interpretasi sebagai berikut:

- Properti: *LOCATION*
- Target: universitas
- Konteks: brawijaya

Dari hasil interpretasi pertanyaan tersebut kemudian dibentuklah sebuah *query* untuk pencarian dokumen yang terdiri dari target, konteks, dan jawaban dari pertanyaan. *Query* yang didapat adalah sebagai berikut:

“universitas” “brawijaya” “kota Malang”

Query digunakan pada *search engine* untuk memperoleh potongan-potongan teks yang sesuai. Kata atau frase yang mengandung target, konteks, dan jawaban (properti) digantikan dengan tag <T>, <C>, dan <P>. Selanjutnya Pola jawaban diekstrak.

Contoh kasusnya jika dari hasil pemrosesan didapatkan potongan teks sebagai berikut “*Universitas Brawijaya terletak di Kota Malang.*” maka pola jawaban yang diekstrak dari potongan teks tersebut:

<T> <C> terletak di <P>

Pola yang diekstrak ini kemudian disimpan dalam sebuah tabel *database* dan termasuk dalam tipe properti *LOCATION*.

b. *Pattern Assessment and Filtering*

Bagian ini berfungsi untuk menilai pola yang didapatkan dari *Pattern Extraction* layak untuk digunakan atau tidak. Pada bagian ini juga digunakan kumpulan pertanyaan yang disertai dengan jawabannya sebagai *data training*. Pertanyaan juga di-*input*-kan ke dalam sistem dan akan melalui tahap-tahap pemrosesan pertanyaan. Pola jawaban yang terbentuk pada proses sebelumnya diuji pada tahap ini.

Pengujian dilakukan dengan mencatat seberapa sering sebuah pola jawaban dipakai dalam menjawab pertanyaan. Kemudian dilakukan penghitungan untuk setiap pola pertanyaan sebagai berikut:

Untuk setiap pola pertanyaan a , akan dicatat seberapa sering pola ini dapat digunakan untuk mengekstrak jawaban yang benar ($\#correct_a$) dan seberapa sering didapatkan jawaban yang salah ($\#incorrect_a$). Kemudian untuk setiap tipe properti p , jumlah potongan teks juga dicatat ($\# snippets_p$). Nilai ini digunakan untuk menghitung pengukuran berikut [TOB-10]:

$$Confidence_a = \frac{\#correct_a}{\#correct_a + \#incorrect_a}$$

$$Support_a = \frac{\#correct_a}{\# snippets_p}$$

(4-1)

Dimana:

- $\#correct_a$: banyaknya jawaban yang benar yang diekstrak dari pola a
- $\#incorrect_a$: banyaknya jawaban yang salah yang diekstrak dari pola a
- $\# snippets_p$: jumlah potongan teks untuk tipe properti p

Kemudian ditentukan nilai threshold untuk nilai minimal dari *confidence* dan *support* untuk menentukan suatu pola pertanyaan layak digunakan atau tidak. Pola dengan nilai *confidence* yang lebih rendah dibandingkan nilai *threshold* dibuang. Begitu pula pola dengan nilai *support* yang lebih rendah dari *threshold* juga akan dibuang. Sedangkan pola yang memenuhi nilai *threshold* akan tetap disimpan di dalam *database*. Pola yang masih tersimpan inilah yang digunakan sebagai pola jawaban pada *Question Answering System*.

4.2.2 Algoritma pada *Question Analyzer*

Bagian ini menjelaskan tentang algoritma yang akan diterapkan pada *Question Analyzer*. Pertama, dilakukan normalisasi pada *string* pertanyaan yaitu dengan menghilangkan semua tanda baca dan karakter lain sehingga hanya tersisa huruf dan angka saja. Kemudian proses selanjutnya dilanjutkan dengan mengubah huruf kapital menjadi huruf kecil. Kedua proses ini dapat dilakukan dengan operasi yang terdapat pada *php*. Untuk menghilangkan tanda baca atau karakter

lain digunakan perintah *preg_replace* dan *regular expression* yang dapat menghilangkan tanda baca dan karakter tertentu, sedangkan untuk mengubah ke huruf kecil digunakan perintah *strtolower*.

Proses selanjutnya adalah dengan mengubah kata atau bagian dari pertanyaan dengan padanan katanya, baik itu sinonim, singkatan, maupun kata bakunya. Hal ini dilakukan untuk mengupayakan agar setiap kata pada pertanyaan dapat dikenali oleh sistem. Untuk itu diperlukan sebuah tabel khusus yang memuat seluruh padanan kata, sinonim, singkatan, atau bentuk baku kata. Setiap kata pada pertanyaan akan dibandingkan dengan isi tabel tersebut, dan jika ditemukan maka kata pada pertanyaan tersebut akan langsung digantikan dengan kata yang terdapat pada tabel. Contohnya, jika pada pertanyaan terdapat kata “*ptiik*” maka sistem akan mengubahnya menjadi “*program teknologi informasi dan ilmu komputer*”.

Proses selanjutnya adalah *Question Analyzer*, yaitu dengan menganalisis pertanyaan dengan membandingkan dengan pola pertanyaan untuk mendapatkan target dan konteks dari pertanyaan. Langkah-langkah yang ditempuh dalam proses ini antara lain:

1. Pertama dicari dahulu kata tanya yang terdapat pada pertanyaan yaitu dengan membandingkannya dengan tabel yang berisi kumpulan kata tanya.
2. Jika kata tanya sudah didapatkan langkah selanjutnya adalah dengan mencari pola pertanyaan yang memiliki kata tanya yang sama dengan pertanyaan.
3. Dari pola pertanyaan yang didapatkan tersebut, kemudian dicari pola yang paling mirip dengan pertanyaan dengan membandingkan tiap *string* katanya.
4. Jika sudah didapatkan pola yang paling mirip kemudian kita dapatkan tipe propertinya dan kita ambil dan simpan kata-kata yang mewakili target dan konteks. Dalam hal ini konteks bisa saja terdiri dari lebih dari satu kata, sedangkan target hanya boleh terdapat satu saja.

Contohnya kasusnya jika terdapat sebuah pertanyaan “*Dimanakah letak UB?*”

Pertama-tama dilakukan normalisasi pada pertanyaan tersebut:

- b.) Menghilangkan tanda baca, hasilnya: “*Dimanakah letak UB*”
- c.) Mengganti huruf kapital dengan huruf kecil, hasilnya: “*dimanakah letak ub*”
- d.) Mengganti dengan padanan katanya: “*dimanakah letak universitas brawijaya*”

Dari pertanyaan hasil normalisasi dicari kata yang menjadi kata tanya, dan didapatkan kata “dimanakah” sebagai kata tanya. Kemudian pertanyaan dibandingkan dengan pola pertanyaan yang memiliki kata tanya “dimanakah” untuk dicari pola yang paling mirip dengan pertanyaan. Dari proses ini ditemukan pola pertanyaan yang mirip adalah:

- (dimanakah) letak <T> <C>
- (dimanakah) <C> <T>

Kedua pola itu memiliki kata tanya yang sama dengan pertanyaan dan juga memiliki pola yang mirip pula dengan pertanyaan. Untuk itu dicarilah pola yang paling spesifik dengan mencari jumlah kata yang sama. Pada pola pertama memiliki dua kata yang sama, yaitu kata “dimanakah” dan kata “letak”. Sedangkan pada pola kedua hanya satu kata yang sama, yaitu kata “dimanakah” saja. Sehingga ditentukan Pola pertama yang digunakan untuk interpretasi.

Selanjutnya adalah dengan mengambil kata yang mewakili <T> dan <C> sebagai target dan konteks dari pertanyaan. Sedangkan pola yang digunakan berasal dari properti lokasi. Sehingga didapatkan

- Properti: *LOCATION* (tempat)
- Target: universitas
- Konteks: brawijaya

Hasil interpretasi suatu pertanyaan bisa saja menghasilkan lebih dari satu konteks. Sedangkan target hanya boleh terdapat satu saja dalam sebuah pertanyaan. Pada contoh kasus di atas sebenarnya konteksnya adalah semua kata yang mengikuti atau terletak setelah kata yang menjadi target, tetapi berhubung kata target hanya diikuti oleh satu kata maka hanya didapatkan satu konteks saja.

Berikut *Pseudocode* yang pada *Question Analyzer* dapat dilihat pada Gambar 4.7.

1	input pertanyaan
2	remove tanda_baca from pertanyaan
3	pertanyaan to lowercase
4	replace words in pertanyaan with synonyms
5	
6	get kata_tanya_pertanyaan from pertanyaan
7	for each pola_pertanyaan in pola_pertanyaan table do
8	if kata_tanya_pola=kata_tanya_pertanyaan
9	if pola_pertanyaan like pertanyaan
10	get properti
11	get target
12	get konteks

Gambar 4.7 *Pseudocode Question Analyzer*

Sumber : Perancangan

4.2.3 Algoritma pada *Search Engine*

Pencarian dokumen pada sistem ini menggunakan dua metode, yaitu *Bag of Words* dan *Question Interpretation Generator*. Untuk metode *Bag of Words*, langkah-langkahnya adalah sebagai berikut:

1. Pertama dilakukan *parsing* pada *string* pertanyaan sehingga menjadi beberapa kata.
2. Kemudian dilakukan *stemming* pada kata hasil parsing sehingga kata tersebut menjadi bentuk kata dasarnya.
3. Setelah *stemming* kemudian dari kata-kata tersebut dibuang kata yang termasuk dalam kata tanya dan *stopword*, yaitu dengan membandingkannya dengan tabel yang berisi kata tanya dan tabel yang berisi *stopword*.
4. Kata-kata yang tersisa inilah yang kemudian digunakan sebagai *keyword* atau kata kunci dalam pencarian dokumen.

Untuk metode *Question Interpretation Generator* digunakan target dan konteks dari hasil interpretasi pertanyaan sebagai kata kunci untuk pencarian dokumen.

Pencarian dokumen dilakukan dengan membandingkan kata kunci (*keyword*) yang sudah diperoleh dengan kata-kata dari dokumen yang sudah melalui proses *parsing*, *stemming*, dan *stopword* yang dijelaskan pada bagian pengumpulan data sebelumnya. Untuk setiap kata kunci yang ditemukan dalam dokumen dihitung juga frekuensi kemunculan katanya. Frekuensi dari kata kunci

tersebut kemudian dijumlahkan untuk setiap dokumennya untuk dapat menentukan dokumen mana yang lebih relevan. Hasil pencarian inilah yang akan digunakan sebagai kandidat jawaban pada proses selanjutnya.

Untuk setiap dokumen yang didapatkan dari proses *Bag of Words* diberikan skor sebanyak 1, sedangkan untuk hasil *Question Interpretation Generator* diberikan skor sebanyak 2. Hal ini dikarenakan metode *Question Interpretation Generator* lebih akurat dan spesifik dalam hasil pencariannya. Pemberian skor ini nantinya akan dipergunakan dalam proses *Answer Extractor and Selector* yang memang membutuhkan penilaian atau *scoring* pada setiap filternya.

Berikut *Pseudocode* yang pada *Search Engine* dapat dilihat pada Gambar 4.8.

1	parsing pertanyaan into pertanyaan_parsed
2	stemming pertanyaan_parsed
3	remove kata_tanya from pertanyaan_parsed
4	remove stopwords from pertanyaan_parsed
5	
6	insert pertanyaan_parsed into keyword table of database
7	
8	search document that have words similar to keywords
9	
10	search document that have words similar to target and konteks
11	
12	get search result

Gambar 4.8 *Pseudocode Search Engine*

Sumber : Perancangan

4.2.4 Algoritma pada *Answer Extractor and Selector*

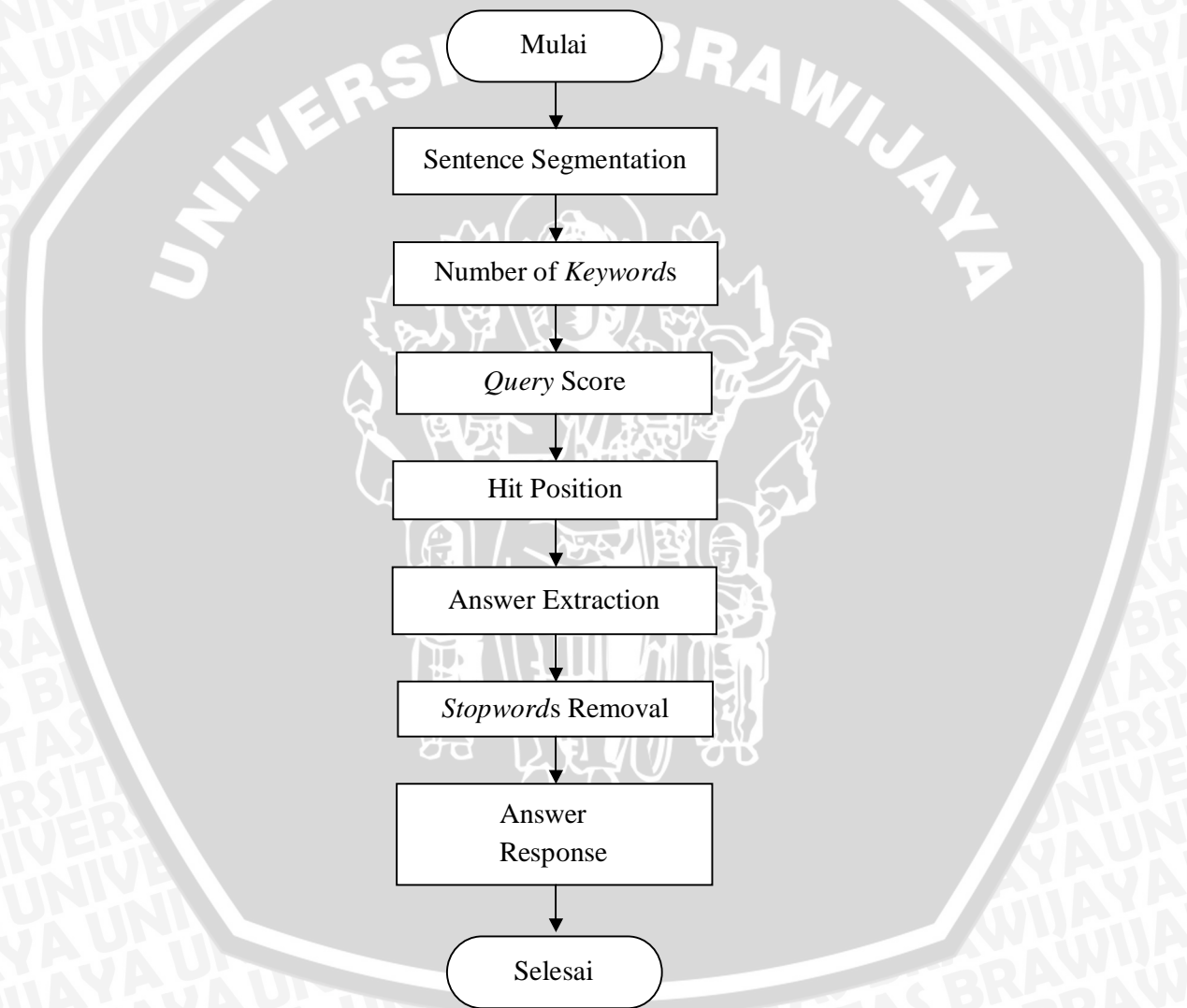
Pada bagian ini, kandidat jawaban melalui beberapa filter dan pemberian skor untuk dapat menentukan jawaban yang paling tepat. *Filter-filter* yang digunakan antara lain *Sentence Segmentation*, *Number of Keywords*, *Query Score*, *Hit Position*, *Answer Extraction*, *Stopwords Removal*, dan *Answer Response* seperti yang terlihat pada Gambar 4.9.

a. *Sentence Segmentation*

Kandidat jawaban yang masih berupa dokumen dipisahkan menjadi bentuk kalimat-kalimat. Pada proses memisahkan kalimat ini, yang digunakan sebagai penanda atau batas antara satu kalimat dengan kalimat yang lain

adalah tanda titik, tanda tanya, dan tanda seru. Proses yang dilakukan ini hampir sama dengan proses parsing pada *preprocessing* dokumen, hanya saja hasil dari *Sentence Segmentation* ini adalah berupa kalimat.

Selanjutnya tiap kalimat yang dihasilkan akan dimasukkan ke dalam sebuah tabel *database* yang memuat semua kalimat serta skor yang dimiliki kalimat tersebut. Pada proses ini, kalimat-kalimat tersebut kita sebut sebagai kalimat kandidat jawaban yang baru dan diberi skor awal 0.



Gambar 4.9 Diagram Alir Proses *Answer Extractor and Selector*

Sumber: Perancangan

b. Number of Keywords

Pada bagian ini dihitung jumlah *keyword* yang ada pada setiap kalimat akan dihitung. Yang dimaksud dengan *keyword* disini adalah kata kunci yang digunakan untuk pencarian dokumen pada proses sebelumnya. Kemudian jumlah *keyword* yang ditemukan di dalam kalimat dimasukkan ke dalam persamaan sebagai berikut [SCH-05]:

$$M \geq \lceil \sqrt{K - 1} \rceil + 1 \quad (4-2)$$

Dimana:

- K = jumlah kata kunci atau *keyword* pada pertanyaan
- M = jumlah kata kunci atau *keyword* yang juga terdapat pada kalimat yang menjadi kandidat jawaban.

Kalimat-kalimat kandidat jawaban akan dibuang jika jumlah *keyword* yang terdapat pada kandidat jawaban tidak memenuhi kondisi di atas. Jika memenuhi kondisi ini, sebuah kalimat kandidat jawaban akan mendapat skor berdasarkan nilai M . Agar perbedaan skor tidak terlampau jauh maka perlu dilakukan normalisasi pada pemberian skor yaitu dengan skala skor antara 0 sampai 10. Perhitungan skor adalah sebagai berikut:

$$\text{skor} = \frac{M}{K} \times 10 \quad (4-3)$$

c. Query Score

Pemberian skor ini berdasarkan dari jenis *query* yang digunakan untuk mendapatkan dokumen. Pada bagian ini dokumen yang diperoleh dari metode *Bag of Words* akan mendapatkan skor 1, sedangkan dokumen yang diperoleh dari metode *Question Interpretation* skor yang didapatkan adalah 2. Hal ini dikarenakan *Question Interpretation* memiliki presisi yang lebih tinggi dan lebih spesifik dibandingkan dengan metode *Bag of Words*.

d. Hit Position

Pada bagian ini skor yang diberikan pada kandidat jawaban adalah sesuai dengan total frekuensi kata kunci atau *keyword* yang muncul pada dokumen tempat kalimat kandidat jawaban berasal. Karena frekuensi total

pada setiap dokumen bisa saja sangat berbeda jauh, dimana suatu dokumen bisa saja memiliki frekuensi total sampai ratusan sedangkan dokumen lain bisa saja hanya memiliki 1 frekuensi total, maka perlu dilakukan normalisasi pada skor yang diberikan agar perbedaan skor antar dokumen tidak terlampau jauh. Skor yang diberikan pada bagian ini diberikan dengan skala 0 sampai 10. Untuk mendapatkan skor tersebut dilakukan perhitungan sebagai berikut:

$$\text{skor} = \frac{Fa - F_{\min}}{F_{\max} - F_{\min}} \times 10 \quad (4-4)$$

Dimana:

- Fa = frekuensi kata kunci pada dokumen a
- F_{\max} = frekuensi kata kunci tertinggi dari seluruh dokumen
- F_{\min} = frekuensi kata kunci tertinggi dari seluruh dokumen

e. *Answer Extraction*

Pada bagian ini, kata-kata yang mewakili target dan konteks hasil interpretasi pertanyaan pada proses *Question Analyzer* akan digantikan dengan tag $\langle T \rangle$ untuk target, dan $\langle C \rangle$ untuk konteks. Proses ini dapat dilakukan dengan perintah *str_replace* yang terdapat pada *php*.

Selanjutnya kalimat kandidat jawaban ini dibandingkan dengan pola jawaban. Berikut beberapa pola jawaban yang mewakili tipe properti lokasi:

- $\langle T \rangle \langle C \rangle$ terletak di $\langle P \rangle$
- $\langle T \rangle \langle C \rangle$ ada di $\langle P \rangle$
- $\langle T \rangle \langle C \rangle$ berada di $\langle P \rangle$
- $\langle T \rangle \langle C \rangle$ adalah $\langle P \rangle$
- $\langle T \rangle \langle C \rangle$ bernama $\langle P \rangle$

Pada pola pertanyaan tersebut tag $\langle P \rangle$ mewakili potongan kalimat yang nantinya akan kita ekstrak dan akan menjadi jawaban dari pertanyaan.

Untuk dapat melakukan proses diatas, pertama kita cari dulu semua pola jawaban yang memiliki tipe properti yang sesuai dengan hasil interpretasi pertanyaan. Dari semua pola jawaban yang didapatkan, kemudian dicari pola yang paling mirip dengan kalimat kandidat jawaban. Untuk dapat menentukan

kemiripan dapat dilakukan dengan membandingkan posisi <T>, posisi <C>, serta kalimat yang menyertai *tag* <T> dan <C> tersebut.

Setelah didapatkan pola yang paling mirip, kemudian diekstrak potongan kalimat yang mewakili <P> pada kalimat kandidat jawaban. Potongan kalimat ini dapat berupa satu kata atau beberapa kata. Potongan kalimat inilah yang kemudian menjadi kandidat jawaban yang baru. Kandidat jawaban yang baru ini mewarisi skor dari kalimat kandidat jawaban sebelumnya.

Jika dari hasil pengestrakan beberapa kalimat ditemukan potongan kalimat yang sama maka potongan kalimat tersebut akan menjadi satu kandidat jawaban saja. Sedangkan skornya merupakan penjumlahan dari skor tiap kandidat.

f. Stopwords Removal

Filter ini diterapkan pada semua kandidat jawaban dan membuang kandidat jika *string* jawaban mengandung kata *stopword* di dalamnya. Proses yang dilakukan pada bagian ini mirip dengan *stopword* removal pada *preprocessing*, yaitu dengan membuang kata pada kandidat jawaban yang terdapat pada tabel *stopword*.

g. Answer Response

Bagian ini yang membedakan *Question Answering System* yang dirancang dengan *OpenEphyra*. Untuk setiap jawaban yang dihasilkan, sistem meminta pengguna memberikan respon mengenai hasil jawaban yang dikeluarkan. Respon dari pengguna ini akan menentukan skor yang diberikan pada bagian ini, yaitu sebagai berikut:

- Respon “jawaban sesuai” akan memberikan skor +1
- Respon “ragu-ragu/tidak tahu” akan memberikan skor +0
- Respon “jawaban tidak sesuai” akan memberikan skor -1

Skor-skor pada bagian ini akan terus terakumulasi dan tersimpan di dalam tabel *respon_jawaban* untuk setiap jawaban yang sama dan berasal dari properti, target dan konteks yang sama pula.

Jawaban akhir yang ditampilkan kepada pengguna adalah kandidat jawaban yang memiliki skor tertinggi setelah melalui semua proses *filtering* dan *scoring* tersebut. Jika terdapat kandidat jawaban yang memiliki skor yang sama, maka yang ditampilkan adalah kandidat jawaban yang potongan kalimatnya lebih pendek.

Berikut *Pseudocode* yang pada *Search Engine* dapat dilihat pada Gambar 4.10.

```

1 parse search result into sentences
2 insert parsed search result into kandidat_jawaban
3
4 for each kandidat_jawaban do
5     get number of keyword(M) in kandidat_jawaban
6     if  $M < \sqrt{K-1}+1$  do
7         remove kandidat_jawaban
8     else
9         give score to kandidat_jawaban =  $M/K*10$ 
10        give query score to kandidat_jawaban
11        give score to kandidat_jawaban =  $(F_a - F_{min}) / (F_{max} - F_{min}) * 10$ 
12
13
14        replace word in kandidat_jawaban where word=target
15        with <T>
16        replace word in kandidat_jawaban where word=konteks
17        with <C>
18
19        for each answer pattern where
20        properti=answer_pattern.properti do
21            for each kandidat_jawaban where
22            kandidat_jawaban like answer_pattern do
23                extract jawaban
24                remove stopwords from jawaban
25                insert jawaban into hasil_jawaban table
26
27 print hasil_jawaban with most score

```

Gambar 4.10 Pseudocode Answer Extractor and Selector

Sumber : Perancangan

4.2.5 Contoh Implementasi Kasus

Berikut contoh dari perhitungan algoritma *Pattern Based Approach* yang diimplementasikan pada *Question Answering System*. Pada contoh kasus ini pengguna memasukkan pertanyaan “Apakah prasyarat mata kuliah Data Mining?”. Berikut Proses yang berjalan pada sistem:

1. Question Analyzer

- a. Normalisasi pertanyaan

- Menghilangkan tanda baca: “Apakah prasyarat mata kuliah Data Mining”
- Mengganti huruf kapital dengan huruf kecil: “apakah prasyarat mata kuliah data mining”
- Mengganti dengan padanan katanya: “apakah prasyarat mata kuliah data mining”

b. Analisis/Interpretasi Pertanyaan

- Dicari kata tanya pada pertanyaan dan didapatkan kata tanya “apakah”.
- Dilakukan pencarian pada pola pertanyaan dengan kata tanya “apakah” dan memiliki pola yang mirip dengan pertanyaan. Didapatkan pola pertanyaan “apakah <T> mata kuliah <C>” dengan properti *OBJECT*.
- Diekstrak kata-kata yang mewakili target <T> dan konteks <C> dan didapatkan:
 - Target : “prasyarat”
 - Konteks : “data” dan “mining”

2. *Query Generator & Search Engine*

a. *Bag of Words*

- Parsing pertanyaan yang telah dinormalisasi, didapatkan kata-kata sebagai berikut: “apakah”, “prasyarat”, “mata”, “kuliah”, “data”, “mining”.
- Stemming pada kata-kata: “apakah”, “prasyarat”, “mata”, “kuliah”, “data”, “mining”, “syarat”.
- Membuang kata tanya dan stopword: “prasyarat”, “mata”, “kuliah”, “data”, “mining”, “syarat”.

Sehingga kata kunci untuk metode *bag of words* adalah “prasyarat”, “mata”, “kuliah”, “data”, “mining”, dan ”syarat”.

b. *Question Interpretation Generator*

Kata kunci untuk metode ini adalah target dan konteks hasil interpretasi sehingga didapatkan kata kunci sebagai berikut: “prasyarat”, “data”, “mining”.

3. Answer Extractor and Selector

a. Sentence Segmentation

Dokumen hasil pencarian dipisahkan mejadi kalimat-kalimat. Didapatkan sebuah dokumen: “Kode Mata Kuliah Data Mining adalah IFK15032. <kal>Beban Studi Mata Kuliah Data Mining adalah 3 SKS. <kal> Prasyarat Mata Kuliah Data Mining adalah Basis Data.”

Tag <kal> yang terdapat pada dokumen digunakan sebagai pemisah antara satu kalimat dengan kalimat yang lain. Tag <kal> ini ditambahkan secara manual pada dokumen ketika proses pengumpulan dokumen. Setiap string yang dipisahkan dengan <kal> akan menjadi satu kalimat sehingga dokumen tersebut akan terpecah menjadi 3 kalimat, yaitu

- Kalimat 1: “Kode Mata Kuliah Data Mining adalah IFK15032.”
- Kalimat 2: ”Beban Studi Mata Kuliah Data Mining adalah 3 SKS.”
- Kalimat 3: ”Prasyarat Mata Kuliah Data Mining adalah Basis Data.”

Setiap kalimat itu menjadi kandidat jawaban dan memiliki skor awal 0.

b. Number of Keywords

Jumlah *keyword* (M) pada setiap kandidat jawaban dihitung sehingga didapatkan:

- Pada Kalimat 1, $M = 4$
- Pada Kalimat 2, $M = 4$
- Pada Kalimat 3, $M = 5$

Jumlah *keyword* pencarian (K) adalah 6, sehingga ketiga kalimat memenuhi kondisi pada persamaan (4-2). Maka perhitungan skor untuk setiap kalimat kandidat jawaban didapatkan:

- Skor Kalimat 1 = $\frac{4}{6} \times 10 = 6.67 \approx 7$
- Skor Kalimat 2 = $\frac{4}{6} \times 10 = 6.67 \approx 7$
- Skor Kalimat 3 = $\frac{5}{6} \times 10 = 8.33 \approx 8$

c. *Query Score*

Dikarenakan kalimat kandidat jawaban berasal dari dokumen yang sama maka kalimat kandidat jawaban memiliki skor yang sama, yaitu 2 karena dokumen dapat diperoleh melalui metode *Question Interpretation Generator*.

d. *Hit Position*

Skor diberikan sesuai dengan frekuensi kemunculan *keyword* pada dokumen. Didapatkan frekuensi tertinggi dari hasil pencarian adalah 146, sedangkan frekuensi terkecil adalah 1. Sedangkan Frekuensi total pada dokumen yang digunakan pada contoh kasus adalah 60. Perhitungan skornya:

$$\text{skor} = \frac{60 - 1}{146 - 1} \times 10 = 4,06 \approx 4$$

e. *Answer Extraction*

Setiap kata yang mewakili target dan konteks pada kalimat kandidat jawaban diubah dengan <T> dan <C> sehingga:

- Kalimat 1: "Kode Mata Kuliah <C> adalah IFK15032."
- Kalimat 2: "Beban Studi Mata Kuliah <C> adalah 3 SKS."
- Kalimat 3: "<T> Mata Kuliah <C> adalah Basis Data."

Setiap kalimat kandidat jawaban dibandingkan dengan pola jawaban dengan tipe properti *OBJECT* dan diekstrak kata atau potongan kalimat yang akan menjadi jawaban. Pada kalimat 1 dan 2 tidak ditemukan pola jawaban yang cocok karena kalimat 1 dan 2 tidak memiliki kata yang mewakili target dari pertanyaan. Sedangkan pada kalimat 3 ditemukan pola jawaban yang cocok, yaitu "<T> <C> adalah <P>". Kemudian diekstrak potongan kata yang mewakili <P>, yaitu "basis data". Potongan kata inilah yang menjadi kandidat jawaban yang baru dan mewarisi skor dari kalimat 3. Sedangkan bila ada kalimat lain yang menghasilkan kandidat jawaban "basis data" juga maka skornya akan diakumulasikan.

Jika dijumlahkan skor yang dimiliki kalimat 3 adalah 14, yaitu 8 dari *Number of Keywords*, 2 dari *Query Score*, dan 4 dari *Hit Position*.

f. *Stopwords Removal*

Karena tidak ada kata *stopwords* pada kandidat jawaban maka tidak ada kata yang dihilangkan.

g. *Answer Response*

Answer Response memberikan skor berdasarkan respon pengguna sebelumnya yang mengajukan pertanyaan yang sama dengan pertanyaan yang sedang diproses. Karena belum ada respon dari pengguna mengenai pertanyaan yang sedang diproses ini maka tidak ada skor yang diberikan.

Sedangkan bila pertanyaan yang sama pernah diajukan dan pengguna sebelumnya memberikan respon “tidak sesuai”, maka skor akan berkurang -1 untuk setiap respon “tidak sesuai”. Misalkan bila ada 3 kali respon “tidak sesuai” maka skor akhir dari kandidat jawaban “basis data” adalah 11.

4. Pemilihan Jawaban

Jawaban yang ditampilkan pada pengguna adalah kandidat jawaban yang memiliki skor akhir tertinggi. Bila terdapat beberapa kandidat jawaban dengan skor akhir sama, maka akan ditampilkan kandidat jawaban yang lebih pendek, yaitu dengan jumlah karakter/huruf yang lebih sedikit.

BAB V

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi algoritma *Pattern Based Approach* pada *Question Answering System* berdasarkan hasil yang telah diperoleh dari perancangan. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi tiap modul pada file program, dan implementasi algoritma.

5.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sistem yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras

Pengembangan *Question Answering System* menggunakan sebuah komputer dengan spesifikasi perangkat keras yang dijelaskan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Keras Komputer

Nama Komponen	Spesifikasi
Prosesor	Intel (R) Core(TM) Duo CPU T2450 @2.00 GHz (2CPUs)
Memori(RAM)	1526MB
Hardisk	Hitachi HTS54321219A300
Kartu Grafis	Mobile Intel(R) 945GM/GU Express Chipset Family

Sumber: Implementasi

5.1.2 Spesifikasi Perangkat Lunak

Pengembangan *Question Answering System* menggunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 5.2.

Tabel 5.2 Spesifikasi Perangkat Lunak Komputer

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows XP Professional (5.1, Build 2600)
Bahasa Pemrograman	PHP

Tools pemrograman	Macromedia Dreamweaver 8
Web Server	Apache 2.2.17
Web Browser	Google Chrome
DBMS	MySQL

Sumber: Implementasi

5.2 Batasan-Batasan Implementasi

Beberapa batasan dalam mengimplementasikan sistem adalah sebagai berikut:

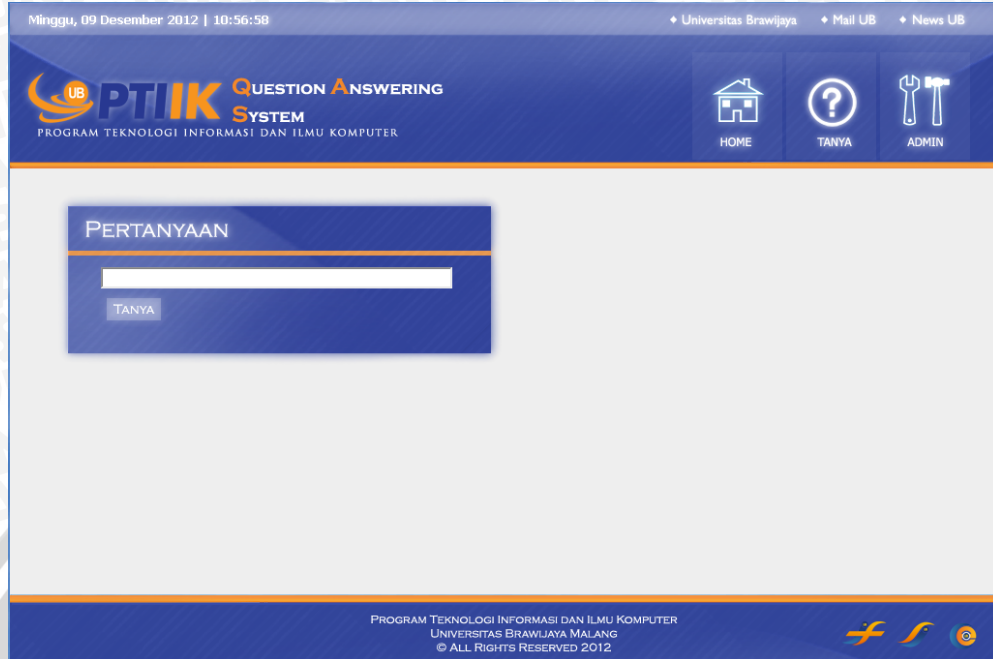
- *Input* yang diterima oleh sistem berupa pertanyaan berbahasa Indonesia.
- *Output* yang diterima pengguna berupa jawaban terbaik yang diperoleh dari pemrosesan jawaban oleh sistem
- Pertanyaan harus memiliki kata tanya agar dapat diproses sistem.
- Bahasa pemrograman yang digunakan adalah PHP.
- *DBMS* yang digunakan adalah MySQL.
- Metode yang digunakan adalah *Pattern Based Approach*.
- Sumber informasi yang digunakan sebagai dokumen berasal dari Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer Tahun Akademik 2012/2013.
- Sistem ini bersifat dinamis, yaitu menyediakan fasilitas *insert* dan *update*.

5.3 Implementasi Antarmuka

Pada implementasi antarmuka ini ditampilkan beberapa tampilan dari aplikasi *Question Answering System*.

5.3.1 Tampilan Halaman Utama

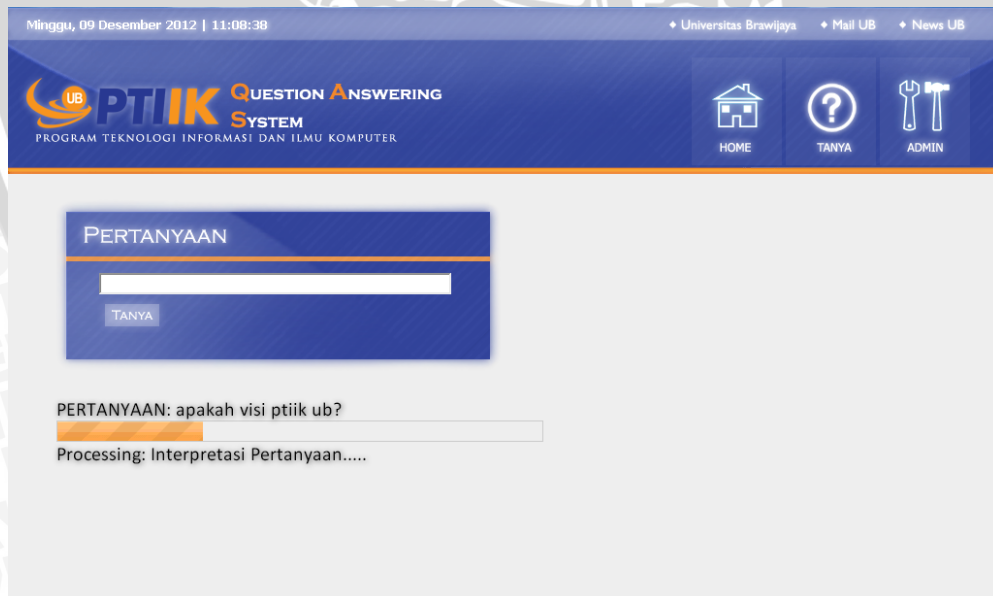
Pada halaman utama pengguna dapat langsung memasukkan pertanyaan pada *text box* yang tersedia. Tampilan halaman utama ditunjukkan pada Gambar 5.1.



Gambar 5.1 Tampilan Halaman Utama

Sumber : Implementasi

Tampilan Halaman Utama ketika sedang memroses pertanyaan dapat dilihat pada Gambar 5.2.



Gambar 5.2 Tampilan Pemrosesan Pertanyaan pada Halaman Utama

Sumber : Implementasi

Tampilan jawaban yang dihasilkan dari hasil pemrosesan pertanyaan dapat dilihat pada Gambar 5.3. Setelah menampilkan jawaban, pengguna diminta untuk menginputkan respon mengenai jawaban yang dihasilkan sistem.

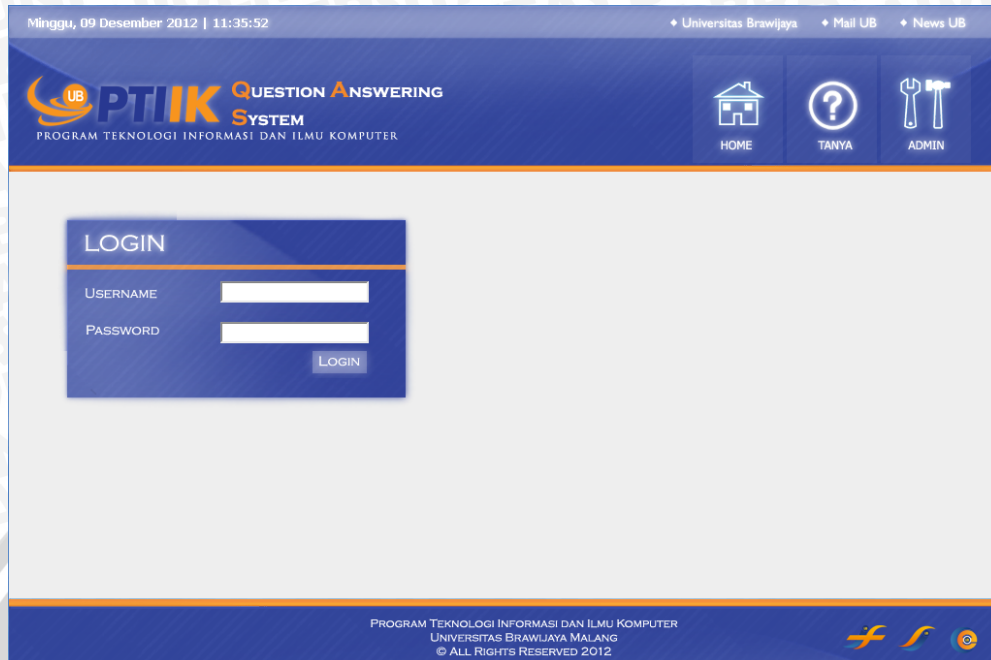


Gambar 5.3 Tampilan Jawaban pada Halaman Utama

Sumber : Implementasi

5.3.2 Tampilan Halaman *Login*

Untuk dapat melakukan pengolahan data pada *database administrator* harus melalui halaman *Login* terlebih dahulu dan memasukkan *username* dan *password*. Halaman *Login* ditunjukkan pada Gambar 5.4

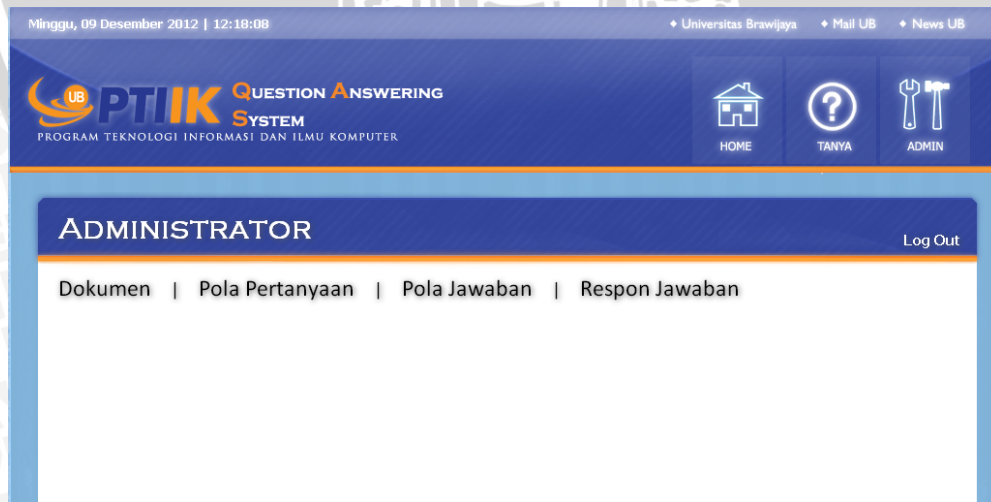


Gambar 5.4 Tampilan Halaman Login

Sumber : Implementasi

5.3.3 Tampilan Halaman *Administrator*

Setelah login, halaman beralih ke halaman *administrator*. Pada halaman ini *Administrator* dapat melihat, menambah, menghapus, ataupun mengubah data yang terdapat pada *database*. Data-data yang dapat diolah pada halaman ini antara lain dokumen, pola pertanyaan, pola jawaban, dan respon jawaban. Tampilan halaman *administrator* ditunjukkan pada Gambar 5.5



Gambar 5.5 Tampilan Halaman *Administrator*

Sumber : Implementasi

5.3.4 Tampilan Halaman Pengelolaan Dokumen

Pada halaman ini dokumen yang terdapat pada *database* dapat dilihat, serta dapat ditambahkan, diubah, maupun dihapus. Tampilan halaman ini dapat dilihat pada Gambar 5.6.

ADMINISTRATOR				Log Out
Dokumen Pola Pertanyaan Pola Jawaban Respon Jawaban				
Dokumen				
id	judul	isi		
1	Visi PTIIK	Visi Program Teknologi Informasi Dan Ilm...	Edit	Del
2	Misi PTIIK	Misi Program Teknologi Informasi Dan Ilm...	Edit	Del
3	Tujuan PTIIK	Tujuan Program Teknologi Informasi Dan I...	Edit	Del
4	Sejarah PTIIK	Program Teknologi Informasi dan Ilmu...	Edit	Del

Gambar 5.6 Tampilan Halaman Pengelolaan Dokumen

Sumber : Implementasi

5.3.5 Tampilan Halaman Pengelolaan Pola Pertanyaan

Pada halaman ini dapat pola pertanyaan yang terdapat pada *database* dapat dilihat, serta dapat ditambahkan, diubah, maupun dihapus. Tampilan halaman ini dapat dilihat pada Gambar 5.7

ADMINISTRATOR							Log Out
Dokumen Pola Pertanyaan Pola Jawaban Respon Jawaban							
Pola Pertanyaan							
id_pola	id_properti	kata_tanya	pola	target	konteks		
1	3	dimana	(dimana) letak <T> <C>	3	4	Edit	Del
2	3	dimana	(dimana) <C> <T>	4	3	Edit	Del
3	3	dimana	(dimana) <T> <C> berada	1	2	Edit	Del
4	3	dimanakah	(dimanakah) letak <T> <C>	3	4	Edit	Del
5	3	dimanakah	(dimanakah) <C> <T>	4	3	Edit	Del
6	3	dimanakah	(dimanakah) <T> <C> berada	1	2	Edit	Del
7	3	dimana	<T> <C> ada (dimana)	1	2	Edit	Del
8	3	dimana	(dimana) saja <T> <C> berada	1	2	Edit	Del
9	3	apa	(apa) nama tempat <T> <C>	3	4	Edit	Del
10	3	apakah	(apakah) nama tempat <T> <C>	3	4	Edit	Del

Gambar 5.7 Tampilan Halaman Pengelolaan Pola Pertanyaan

Sumber : Implementasi

5.3.6 Tampilan Halaman Pengelolaan Pola Jawaban

Pada halaman ini dapat pola jawaban yang terdapat pada *database* dapat dilihat, serta dapat ditambahkan, diubah, maupun dihapus. Tampilan halaman ini dapat dilihat pada Gambar 5.8

id_pola	id_properti	pola	Edit	Del
1	3	<T> <C> terletak di <P>	Edit	Del
2	3	<T> <C> ada di <P>	Edit	Del
3	3	<T> <C> berada di <P>	Edit	Del
4	1	<T> <C> adalah <P>	Edit	Del
5	1	<T> <C> bernama <P>	Edit	Del
6	8	<T> <C> berbunyi <P>	Edit	Del
7	3	<T> <C> di <P>	Edit	Del
8	7	<T> <C> adalah <P>	Edit	Del
9	7	<T> <C> antara lain <P>	Edit	Del
10	7	<T> <C> sebagai berikut <P>	Edit	Del

Gambar 5.8 Tampilan Halaman Pengelolaan Pola Jawaban

Sumber : Implementasi

5.3.7 Tampilan Halaman Pengelolaan Respon Jawaban

Pada halaman ini dapat respon jawaban yang terdapat pada *database* dapat dilihat, serta dapat ditambahkan, diubah, maupun dihapus. Tampilan halaman ini dapat dilihat pada Gambar 5.9

id_jawaban	id_properti	target	jawaban	respon	Edit	Del
1	7	tujuan	<ul style="list-style-type: none"> mahasiswa akan dapat menjelaskan prinsip-prinsip dan terminologi dari teknologi informasi mahasiswa akan dapat menunjukkan penggunaan teknologi informasi untuk memfasilitasi berbagai proses kerja sehari-hari mahasiswa mampu memahami sub bidang ilmu teknologi informasi dan computer 	-5	Edit	Del
2	7	visi	menjadi lembaga unggul dalam pengembangan ilmu pengetahuan di bidang teknologi informasi dan komputer di tingkat nasional dan internasional melalui integrasi tri darma perguruan tinggi.	3	Edit	Del
			1) menghasilkan lulusan yang memiliki kompetensi di bidang tik, berjiwa enterpreneur dan dapat dipercaya sehingga mampu bekerjasama dan memberikan kontribusi di tingkat nasional dan internasional. 2) mengembangkan ilmu pengetahuan dan			

Gambar 5.9 Tampilan Halaman Pengelolaan Respon Jawaban

Sumber : Implementasi

BAB VI

PENGUJIAN DAN ANALISIS

Pada bab ini dilakukan proses pengujian terhadap algoritma *Pattern Based Approach* yang telah diimplementasikan pada *Question Answering System*. Pengujian dilakukan dengan menguji tingkat akurasi dari algoritma *Pattern Based Approach*.

6.1 Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui ketepatan algoritma *Pattern Based Approach* dalam menghasilkan jawaban yang benar. Data yang diuji berjumlah 50 sampel pertanyaan. Setiap tipe properti minimal ditanyakan sebanyak 1 kali dalam sampel pertanyaan. Sampel pertanyaan yang diambil sebanyak 50 supaya pengujian dapat mencakup sebanyak mungkin pertanyaan yang sering ditanyakan mengenai isi buku pedoman.

Prosedur pengujiannya adalah beberapa mahasiswa memasukkan pertanyaan ke *Question Answering System*, kemudian sistem akan memproses pertanyaan dan menampilkan jawaban terbaik yang dihasilkan. Hasil jawaban yang dikeluarkan sistem kemudian dibandingkan dengan jawaban yang sebenarnya sesuai dengan isi Buku Pedoman Pendidikan.

Dari hasil perbandingan antara hasil sistem dan hasil sebenarnya akan didapatkan beberapa kemungkinan

- *True Positive* (TP): Pertanyaan yang diajukan berhubungan dengan isi buku pedoman dan sistem menghasilkan jawaban yang benar.
- *False Positive* (FP): Pertanyaan yang diajukan berhubungan dengan isi buku pedoman dan sistem menghasilkan jawaban yang salah atau tidak menghasilkan jawaban.
- *True Negative* (TN): Pertanyaan yang diajukan tidak berhubungan dengan isi buku pedoman dan sistem tidak menghasilkan jawaban.
- *False Negative* (FN): Pertanyaan yang diajukan tidak berhubungan dengan isi buku pedoman tetapi sistem menghasilkan jawaban.

Tabel 6.1 Tabel Hasil Pengujian Akurasi

No	Pertanyaan	Jawaban Sistem	Jawaban Benar	Ket	Tipe
1	bagaimana bunyi visi ptiik?	menjadi lembaga unggul dalam pengembangan ilmu pengetahuan di bidang teknologi informasi dan komputer di tingkat nasional dan internasional melalui integrasi tri darma perguruan tinggi.	menjadi lembaga unggul dalam pengembangan ilmu pengetahuan di bidang teknologi informasi dan komputer di tingkat nasional dan internasional melalui integrasi tri darma perguruan tinggi.	TP	8
2	berapa jumlah sks kecerdasan buatan?	3 sks	3 sks	TP	6
3	apakah skripsi itu?	suatu karya tulis ilmiah yang didasarkan atas penelitian/ perencanaan/ perancangan/ sigi/ studi literatur/ studi perbandingan/ studi kasus/ studi kelayakan dalam bidang rekayasa yang sesuai dengan jurusan/ program studinya.	suatu karya tulis ilmiah yang didasarkan atas penelitian/ perencanaan/ perancangan/ sigi/ studi literatur/ studi perbandingan/ studi kasus/ studi kelayakan dalam bidang rekayasa yang sesuai dengan jurusan/ program studinya.	TP	7
4	Dimanakah alamat PTIIK?	Jl. Veteran No.8 Malang	Jl. Veteran No.8 Malang	TP	3
5	Apa program studi di bawah PTIIK?	program studi informatika, program studi sistem komputer dan program studi sistem informasi.	program studi informatika, program studi sistem komputer dan program studi sistem informasi.	TP	4
6	berapa lama batas waktu penyelesaian skripsi?	enam bulan terhitung mulai tanggal diterbitkannya surat tugas.	enam bulan terhitung mulai tanggal diterbitkannya surat tugas.	TP	5
7	apa yang dimaksud dengan satuan kredit semester?	144 sks.	adalah takaran penghargaan terhadap pengalaman belajar yang diperoleh selama satu semester melalui kegiatan terjadwal per minggu sebanyak 1 jam perkuliahan atau 2	FP	7

			jam praktikum, atau 4 jam kerja lapangan yang masing-masing diiringi sekitar 1–2 jam kegiatan terstruktur dan sekitar 1–2 jam kegiatan mandiri.		
8	apa pustaka sistem administrasi server?	<p>1. lee, digital circuit and logic design, new delhi, phi, 1981.</p> <p>2. mano, digital logic and computer design, new delhi, phi, 1986.</p> <p>3. mismail, budiono. dasar-dasar logika digital. malang: lembaga penerbitan universitas brawijaya, 1982.</p>	<p>1. lee, digital circuit and logic design, new delhi, phi, 1981.</p> <p>2. mano, digital logic and computer design, new delhi, phi, 1986.</p> <p>3. mismail, budiono. dasar-dasar logika digital. malang: lembaga penerbitan universitas brawijaya, 1982.</p>	TP	7
9	bagaimana syarat menempuh ujian akhir sarjana?	<p>a. terdaftar sebagai mahasiswa pada tahun akademik yang bersangkutan.</p> <p>b. telah lulus pkk-maba</p> <p>c. telah mengumpulkan satuan kegiatan mahasiswa (skm) sebanyak 500 poin.</p> <p>d. lulus semua mata kuliah, tugas dan praktikum yang telah ditetapkan dengan ipk > 2,00</p> <p>e. telah menyelesaikan skripsi yang ditandatangani oleh dosen (para dosen) pembimbing sebagai tanda persetujuannya untuk menempuh ujian akhir sarjana.</p> <p>f. lulus seminar hasil skripsi bila jurusan/program studi yang bersangkutan menyelenggarakannya.</p> <p>g. memenuhi syarat-syarat akademik lain yang ditetapkan oleh masing-masing</p>	<p>a. terdaftar sebagai mahasiswa pada tahun akademik yang bersangkutan.</p> <p>b. telah lulus pkk-maba</p> <p>c. telah mengumpulkan satuan kegiatan mahasiswa (skm) sebanyak 500 poin.</p> <p>d. lulus semua mata kuliah, tugas dan praktikum yang telah ditetapkan dengan ipk > 2,00</p> <p>e. telah menyelesaikan skripsi yang ditandatangani oleh dosen (para dosen) pembimbing sebagai tanda persetujuannya untuk menempuh ujian akhir sarjana.</p> <p>f. lulus seminar hasil skripsi bila jurusan/program studi yang bersangkutan menyelenggarakannya.</p> <p>g. memenuhi syarat-syarat akademik lain yang ditetapkan oleh masing-masing</p>	TP	8

		jurusan/program studi.	jurusan/program studi.		
10	apakah prasyarat data mining?	sistem basis data	sistem basis data	TP	7
11	berapa jangka waktu revisi skripsi?	paling lama 1 (satu) bulan terhitung dari tanggal kelulusan ujian akhir sarjana, dan: a. apabila revisi melebihi 1 bulan, maka diadakan ujian akhir sarjana. b. apabila revisi melebihi masa studi, maka dikeluarkan dari program.	paling lama 1 (satu) bulan terhitung dari tanggal kelulusan ujian akhir sarjana, dan: a. apabila revisi melebihi 1 bulan, maka diadakan ujian akhir sarjana. b. apabila revisi melebihi masa studi, maka dikeluarkan dari program.	TP	5
12	kapan evaluasi keberhasilan studi mahasiswa dilakukan?	a. akhir tiap semester. b. akhir tahun pertama (dua semester). c. akhir tahun kedua (empat semester). d. akhir tahun ketiga (enam semester). e. akhir tahun keempat (delapan semester). f. akhir program studi sarjana (setelah mencapai 144 sks). g. akhir batas waktu studi (empat belas semester).	a. akhir tiap semester. b. akhir tahun pertama (dua semester). c. akhir tahun kedua (empat semester). d. akhir tahun ketiga (enam semester). e. akhir tahun keempat (delapan semester). f. akhir program studi sarjana (setelah mencapai 144 sks). g. akhir batas waktu studi (empat belas semester).	TP	2
13	siapakah penasehat akademik?	dosen yang ditugaskan untuk membimbing mahasiswa sesuai dengan program studi tempatnya bertugas.	dosen yang ditugaskan untuk membimbing mahasiswa sesuai dengan program studi tempatnya bertugas.	TP	1
14	berapa jumlah sks jaringan komputer?	4 sks	4 sks	TP	6
15	apakah mata kuliah falsafah ilmu pengetahuan masih ada?	-	-	TN	7

16	berapa jumlah sks mata kuliah grafika komputer?	3 sks	3 sks	TP	6
17	apa kode mata kuliah grafika komputer?	ifk15203	ifk15203	TP	7
18	bagaimana bunyi pasal 21?	pasal 21 visi, misi, tujuan, kurikulum, silabus dan peraturan khusus jurusan/program studi ditetapkan oleh masing-masing program studi dan disampaikan pada bagian lain pedoman pendidikan ini	pasal 21 visi, misi, tujuan, kurikulum, silabus dan peraturan khusus jurusan/program studi ditetapkan oleh masing-masing program studi dan disampaikan pada bagian lain pedoman pendidikan ini	TP	8
19	siapakah ketua program ptiik?	Ir. Sutrisno, MT	Ir. Sutrisno, MT	TP	1
20	berapa range nilai A?	antara $80 < n \leq 100$	antara $80 < n \leq 100$	TP	5
21	apa yang dimaksud dengan kuliah kerja nyata?	3 sks	kegiatan akademik terstruktur yang dilakukan di perusahaan, proyek dan/atau instansi yang disetujui Ketua Jurusan/Program Studi.	FP	7
22	bagaimana syarat alih program?	a. lulusan d3 perguruan tinggi dengan program studi yang bersesuaian dan berakreditasi ban pt sekurang-kurangnya b. b. mempunyai ipk $\geq 3,0$. c. program pendidikan d3 ditempuh selama tidak lebih dari empat tahun. d. calon yang ditugaskan dari suatu instansi sebagai mahasiswa tugas belajar harus telah mempunyai masa kerja di bidang keahliannya sekurangkurangnya dua tahun	a. lulusan d3 perguruan tinggi dengan program studi yang bersesuaian dan berakreditasi ban pt sekurang-kurangnya b. b. mempunyai ipk $\geq 3,0$. c. program pendidikan d3 ditempuh selama tidak lebih dari empat tahun. d. calon yang ditugaskan dari suatu instansi sebagai mahasiswa tugas belajar harus telah mempunyai masa kerja di bidang keahliannya sekurangkurangnya dua tahun	TP	8

		dan ipk $\geq 2,50$.	dan ipk $\geq 2,50$.		
23	mata kuliah dasar pemrograman komputer berganti nama menjadi apa?	-	dasar pemrograman	FP	7
24	siapakah dosen penguji?	i. ketua dan sekretaris serendah-rendahnya mempunyai jabatan lektor kepala, atau lektor dengan tambahan gelar master, atau asisten ahli dengan tambahan gelar doktor. ii. penguji serendah-rendahnya mempunyai jabatan lektor atau asisten ahli dengan tambahan gelar master/doktor.	i. ketua dan sekretaris serendah-rendahnya mempunyai jabatan lektor kepala, atau lektor dengan tambahan gelar master, atau asisten ahli dengan tambahan gelar doktor. ii. penguji serendah-rendahnya mempunyai jabatan lektor atau asisten ahli dengan tambahan gelar master/doktor.	TP	1
25	berapa range nilai b+?	antara $75 < n \leq 80$	antara $75 < n \leq 80$	TP	5
26	apa semester pendek?	program perkuliahan yang dilaksanakan pada saat liburan semester genap disesuaikan dengan kalender akademik program.	program perkuliahan yang dilaksanakan pada saat liburan semester genap disesuaikan dengan kalender akademik program.	TP	7
27	penasehat akademik mempunyai tugas apa?	a. memberi penjelasan dan petunjuk tentang rencana studi yang ditempuh mahasiswa bimbingannya. b. memberi bimbingan dan nasehat dalam memilih beberapa matakuliah yang sesuai dengan rencana studi mahasiswa bimbingannya. c. memberi bimbingan dan nasehat dalam masalah akademik dan non-akademik sehubungan dengan rencana studi mahasiswa bimbingannya.	a. memberi penjelasan dan petunjuk tentang rencana studi yang ditempuh mahasiswa bimbingannya. b. memberi bimbingan dan nasehat dalam memilih beberapa matakuliah yang sesuai dengan rencana studi mahasiswa bimbingannya. c. memberi bimbingan dan nasehat dalam masalah akademik dan non-akademik sehubungan dengan rencana studi mahasiswa bimbingannya.	TP	7
28	bagaimana profil lulusan studi teknik informatika	prf-1 : it support prf-2 : perekayasa perangkat lunak (software	prf-1 : it support prf-2 : perekayasa perangkat lunak (software	TP	8

	dan ilmu komputer?	engineer/developer) prf-3 : pengembang sistem cerdas (intelligence system developer) prf-4 : perancang dan pengembang game (game designer and developer) prf-5 : perekayasa jaringan komputer (computer network engineer)	engineer/developer) prf-3 : pengembang sistem cerdas (intelligence system developer) prf-4 : perancang dan pengembang game (game designer and developer) prf-5 : perekayasa jaringan komputer (computer network engineer)		
29	Apa saja syarat-syarat menempuh ujian akhir semester?	a. terdaftar sebagai mahasiswa pada tahun akademik yang bersangkutan. b. telah lulus pkk-maba c. telah mengumpulkan satuan kegiatan mahasiswa (skm) sebanyak 500 poin. d. lulus semua mata kuliah, tugas dan praktikum yang telah ditetapkan dengan ipk > 2,00 e. telah menyelesaikan skripsi yang ditandatangani oleh dosen (para dosen) pembimbing sebagai tanda persetujuannya untuk menempuh ujian akhir sarjana. f. lulus seminar hasil skripsi bila jurusan/program studi yang bersangkutan menyelenggarakannya. g. memenuhi syarat-syarat akademik lain yang ditetapkan oleh masing-masing jurusan/program studi.	a. terdaftar sebagai mahasiswa pada tahun akademik yang bersangkutan. b. telah lulus pkk-maba c. telah mengumpulkan satuan kegiatan mahasiswa (skm) sebanyak 500 poin. d. lulus semua mata kuliah, tugas dan praktikum yang telah ditetapkan dengan ipk > 2,00 e. telah menyelesaikan skripsi yang ditandatangani oleh dosen (para dosen) pembimbing sebagai tanda persetujuannya untuk menempuh ujian akhir sarjana. f. lulus seminar hasil skripsi bila jurusan/program studi yang bersangkutan menyelenggarakannya. g. memenuhi syarat-syarat akademik lain yang ditetapkan oleh masing-masing jurusan/program studi.	TP	7
30	evaluasi keberhasilan studi mahasiswa ditentukan berdasarkan apa?	Indeks prestasi kumulatif dan perolehan kreditnya.	Indeks prestasi kumulatif dan perolehan kreditnya.	TP	7
31	apakah tujuan	memperlihatkan kemampuan mahasiswa	memperlihatkan kemampuan mahasiswa	TP	7

	penyusunan skripsi?	dalam hal pemecahan persoalan berdasarkan ilmu dan teknologi.	dalam hal pemecahan persoalan berdasarkan ilmu dan teknologi.		
32	bagaimana pelaksanaan administrasi akademik?	a. melaksanakan daftar ulang di awal semester. b. penentuan rencana studi semester mahasiswa. c. perubahan rencana studi mahasiswa. d. presensi kuliah, tugas, seminar, praktikum dan praktik kerja. e. kuis, ujian tengah semester dan ujian akhir semester f. pengisian dan pengesahan kartu hasil studi. g. pengesahan hasil studi mahasiswa dalam laporan hasil studi.	a. melaksanakan daftar ulang di awal semester. b. penentuan rencana studi semester mahasiswa. c. perubahan rencana studi mahasiswa. d. presensi kuliah, tugas, seminar, praktikum dan praktik kerja. e. kuis, ujian tengah semester dan ujian akhir semester f. pengisian dan pengesahan kartu hasil studi. g. pengesahan hasil studi mahasiswa dalam laporan hasil studi.	TP	8
33	berapakah maksimal sks yang dapat diambil jika IP 3,2?	-	24 sks	FP	6
34	siapakah Wakil Ketua I PTIIK?	ir. heru nurwarsito, m.kom	ir. heru nurwarsito, m.kom	TP	1
35	apakah prasyarat data mining?	sistem basis data	sistem basis data	TP	7
36	berapakah bobot nilai c+?	2,5	2,5	TP	6
37	siapakah wakil ketua bidang kemahasiswaan?	edy santoso, s.si., m.kom	edy santoso, s.si., m.kom	TP	1
38	Apakah alih program?	perpindahan dari lulusan program D3 ke program pendidikan S1.	perpindahan dari lulusan program D3 ke program pendidikan S1.	TP	7
39	apa pustaka sistem administrasi server?	1. lee, digital circuit and logic design, new delhi, phi, 1981. 2. mano, digital logic and computer design, new delhi, phi, 1986.	1. lee, digital circuit and logic design, new delhi, phi, 1981. 2. mano, digital logic and computer design, new delhi, phi, 1986.	TP	7

		3. mismail, budiono. dasar-dasar logika digital. malang: lembaga penerbitan universitas brawijaya, 1982.	3. mismail, budiono. dasar-dasar logika digital. malang: lembaga penerbitan universitas brawijaya, 1982.		
40	Berapa lama rentang waktu paling lambat pengajuan alih program yang dilakukan oleh mahasiswa?	-	paling lambat satu bulan sebelum kuliah Tahun Akademik Baru dimulai.	FP	5
41	apa saja kecurangan dalam administrasi akademik?	a. memalsu dokumen akademik dan tanda tangan. b. memalsu surat puas, menyuap, serta memalsu data dalam proses praktikum, pengerjaan tugas. c. mengubah isi krs dan khs secara tidak sah.	a. memalsu dokumen akademik dan tanda tangan. b. memalsu surat puas, menyuap, serta memalsu data dalam proses praktikum, pengerjaan tugas. c. mengubah isi krs dan khs secara tidak sah.	TP	7
42	apa saja tugas dari dosen pembimbing skripsi?	a. membimbing mahasiswa dalam penyusunan skripsi. b. memberi nilai skripsi mahasiswa bimbingannya. c. mendampingi mahasiswa pada waktu ujian akhir.	a. membimbing mahasiswa dalam penyusunan skripsi. b. memberi nilai skripsi mahasiswa bimbingannya. c. mendampingi mahasiswa pada waktu ujian akhir.	TP	7
43	Bagaimana cara mahasiswa agar dapat mengajukan cuti akademik?	a. mengajukan permohonan pindah secara tertulis dengan alasan yang kuat kepada rektor dengan tembusan kepada ketua program serta dilampiri persyaratan yang diperlukan. b. telah mengikuti pendidikan secara terus-menerus sekurang-kurangnya dua semester dan setinggi-tingginya tiga semester serta telah mengumpulkan sekurang-kurangnya 40 dan 60 sks untuk berturut-turut 2 dan 3 semester dengan ipk $\geq 3,00$.	Permohonan cuti akademik diajukan kepada rektor, dengan disertai alasan-alasan yang kuat, diketahui oleh ketua program, orang tua/wali/instansi mahasiswa yang bersangkutan. Pengajuan ini paling lambat satu bulan sejak penutupan registrasi akademik.	FP	7

		<p>c. tidak berstatus putus studi karena tidak memenuhi ketentuan akademik.</p> <p>d. tidak pernah mendapat hukuman akademik atau hukuman lain dari perguruan tinggi asal atau instansi pemerintah lain.</p> <p>e. mendapat persetujuan dari perguruan tinggi asal.</p> <p>f. lulus ujian penjurusan.</p>			
44	apa sanksi bagi mahasiswa yang melakukan kecurangan?	-	<p>a. Skorsing, bila dipidana kurang dari satu tahun.</p> <p>b. Diberhentikan sebagai mahasiswa Universitas Brawijaya, jika dipidana lebih dari satu tahun.</p>	FP	7
45	apa saja kecurangan akademik?	<p>a. menyontek, mengambil pekerjaan peserta lain, kerjasama selama kuis/ujian berlangsung.</p> <p>b. menjiplak laporan kkn-p, praktikum, pengabdian masyarakat, pengerjaan tugas, dan skripsi.</p>	<p>a. menyontek, mengambil pekerjaan peserta lain, kerjasama selama kuis/ujian berlangsung.</p> <p>b. menjiplak laporan kkn-p, praktikum, pengabdian masyarakat, pengerjaan tugas, dan skripsi.</p>	TP	7
46	Apa saja syarat-syarat seorang mahasiswa dinyatakan selesai mengikuti kuliah Program Studi?	nilai yang terbaik. yang dimaksud dengan memperbaiki nilai adalah dengan memprogram dan menempuh kembali mata kuliah yang bersangkutan.	<p>i. IPK $\geq 2,00$.</p> <p>ii. Telah lulus seluruh mata kuliah wajib.</p> <p>iii. Tidak ada nilai E.</p> <p>iv. Total sks mata kuliah yang mempunyai nilai D dan D+, tidak melebihi 10% dari sks mata kuliah yang harus ditempuh.</p> <p>v. Telah menyelesaikan skripsi.</p> <p>vi. Telah menyelesaikan tugas akademik lainnya.</p> <p>vii. Telah lulus ujian akhir sarjana.</p> <p>viii. Telah memenuhi syarat-syarat administrasi.</p> <p>ix. Memiliki nilai TOEIC dengan skor 500 dari institusi yang diakui oleh Program.</p> <p>x. Memiliki sertifikat program aplikasi</p>	FP	7

			komputer sekurang-kurangnya 2 (dua) jenis aplikasi komputer dari institusi yang diakui oleh Program. xi. Memiliki publikasi Ilmiah, minimal satu publikasi.		
47	apakah KKN-P itu wajib?	-	Wajib	FP	7
48	kapan ptiik dibentuk?	Tahun 2011	Tahun 2011	TP	2
49	Bagaimana prosedur daftar ulang?	-	-	TN	8
50	Bagaimana prosedur terminal?	-	-	TN	8

Sumber: Pengujian

Kolom tipe pada Tabel 6.1 mewakili *id* dari setiap tipe properti. *Id* untuk tiap propertinya antara lain:

- *id*=1 untuk properti *PEOPLE* (Orang),
- *id*=2 untuk properti *TIME* (Waktu)
- *id*=3 untuk properti *LOCATION* (Tempat)
- *id*=4 untuk properti *ORGANIZATION* (Organisasi)
- *id*=5 untuk properti *MEASURE* (Ukuran)
- *id*=6 untuk properti *COUNT* (Angka)
- *id*=7 untuk properti *OBJECT* (Obyek)
- *id*=8 untuk properti *OTHER* (Lain)

Berdasarkan pengujian pada Tabel 6.1 didapatkan:

- *TP* sebanyak 38
- *FP* sebanyak 9
- *TN* sebanyak 3
- *FN* sebanyak 0

Dari hasil pengujian tersebut kemudian dihitung nilai presisi dan *recall*, yaitu sebagai berikut:

1. Perhitungan Presisi

$$\text{Presisi} = \frac{38}{38+9} \times 100\% = 80,85\%$$

2. Perhitungan *Recall*

$$\text{Recall} = \frac{38}{38+0} \times 100\% = 100\%$$

6.2 Analisis Hasil Pengujian

Dari pengujian yang dilakukan pada implementasi algoritma *Pattern Based Approach* pada *Question Answering System* dihasilkan tingkat presisinya adalah 80.85% dengan *recall* sebesar 100%. Dengan pengujian pada 50 sampel pertanyaan didapatkan 38 *True Positive*, 9 *False Positive*, 3 *True Negative*, dan tidak ada *False Negative*.

Hasil *False Positive* didapatkan pada sampel pertanyaan nomor 7, 21, 23, 33, 40, 43, 44, 46, dan 47. Berikut analisis yang dilakukan pada beberapa sampel pertanyaan untuk dapat mengetahui penyebab kesalahan:

1. Dari hasil analisis pada sampel nomor 7 dengan pertanyaan “*apa yang dimaksud dengan satuan kredit semester?*” ditemukan penyebab kesalahan terdapat pada proses *scoring* kandidat jawaban. Untuk dapat mengetahuinya dapat dilakukan dengan melihat isi tabel *hasil_jawaban* yang memuat semua kandidat jawaban beserta skornya. Berikut cuplikan isi dari tabel *hasil_jawaban* pada Gambar 6.1.

	jawaban	skor
<input type="checkbox"/>	suatu angka yang menunjuk... 969B	19
<input type="checkbox"/>	setara dengan beban tugas... 127B	16
<input type="checkbox"/>	144 sks. 9B	29
<input checked="" type="checkbox"/>	takaran penghargaan terha... 303B	16
<input type="checkbox"/>	beban tugas di laborator... 96B	19

→ jawaban yang ditampilkan
 → jawaban benar

Gambar 6.1 Cuplikan isi tabel hasil_jawaban

Sumber: Pengujian

Pada Gambar 6.1 dapat dilihat bahwa jawaban yang ditampilkan adalah "144 sks." Dengan skor 29. Sedangkan jawaban yang sebenarnya adalah "takaran penghargaan terhadap pengalaman belajar yang diperoleh selama satu semester melalui kegiatan terjadwal per minggu sebanyak 1 jam perkuliahan atau 2 jam pratikum, atau 4 jam kerja lapangan yang masing-masing diiringi sekitar 1–2 jam kegiatan terstruktur dan sekitar 1–2 jam kegiatan mandiri." dengan skor 19 seperti yang ditunjukkan pada gambar.

Jawaban yang ditampilkan memiliki skor lebih tinggi dari jawaban yang sebenarnya karena skor tersebut merupakan akumulasi dari beberapa kandidat jawaban yang sama. Hasil ini dapat diketahui dengan melihat isi dari tabel temp_jawaban2. Cuplikan tabel temp_jawaban2 ditunjukkan pada Gambar 6.2.

jawaban	skor_total
suatu angka yang menunj... 969B	19
kurikulum yang disusun ... 3K	19
setara dengan beban tug... 127B	16
interaksi antara mahasi... 1K	16
144 sks. 9B	10
144 sks. 9B	10
144 sks. 9B	10
takaran penghargaan ter... 303B	16

Gambar 6.2 Cuplikan tabel temp_jawaban2

Sumber: Pengujian

Total skor dari ketiga kandidat jawaban dengan jawaban "144 sks." Tersebut adalah 30. Skor 29 pada tabel hasil_jawaban didapatkan karena skor total dari ketiga kandidat dikurangi dengan skor yang terdapat pada respon

jawaban yang sebesar -1 sehingga didapatkan angka 29. Skor pada respon jawaban dapat dilihat di tabel respon_jawaban yang terlihat pada gambar 6.3.

jawaban		respon
: <img ...	42B	-1
 prfl :...	597B	1
 prf-1 ...	334B	2
tuhan yang maha...	156B	-3
nilai yang meny...	161B	10
144 sks.	9B	-1
memperlihatkan ...	97B	3
: a. me...	1K	-4

Gambar 6.3 Cuplikan tabel respon_jawaban

Sumber: Pengujian

Sedangkan jawaban yang sebenarnya hanya mendapatkan skor 19 dikarenakan jawaban tersebut tidak memiliki kandidat lain yang memiliki jawaban yang sama sehingga skor tidak terakumulasi. Untuk meminimalisir agar kesalahan ini tidak terjadi lagi dapat dilakukan dengan proses learning jawaban dengan memanfaatkan skor dari respon jawaban.

2. Pada sampel nomor 21 dengan pertanyaan “*apa yang dimaksud dengan kuliah kerja nyata?*” ditemukan penyebab kesalahan yang sama dengan kesalahan pada sampel nomor 7, yaitu pada proses scoring kandidat jawaban. Penyebab kesalahan pada *scoring* ini juga sama dengan sampel 7, yaitu jawaban salah yang ditampilkan sistem memiliki lebih dari satu kandidat, sedangkan jawaban sebenarnya hanya memiliki satu kandidat.
3. Pada sampel 23 dengan pertanyaan “*mata kuliah dasar pemrograman komputer berganti nama menjadi apa?*” ditemukan penyebab kesalahan adalah pertanyaan tidak dapat diinterpretasikan oleh sistem. Hal ini disebabkan karena tidak ada pola pertanyaan yang mirip dengan string pertanyaan. Untuk meminimalisir agar kesalahan ini tidak terjadi lagi dapat dilakukan dengan menambahkan lagi pola pertanyaan agar setiap pertanyaan yang dimasukkan dapat diinterpretasikan dengan baik.

Hasil *True Negative* terdapat sampel nomor 15, 49, dan 50. Berikut analisis pada sampel pertanyaan 15 dengan pertanyaan “*apakah mata kuliah*

falsafah ilmu pengetahuan masih ada?'. Hasil ini didapatkan karena tidak ditemukannya dokumen yang relevan dengan pertanyaan tersebut. Pada dokumen buku pedoman tidak terdapat dokumen yang mengandung *"mata kuliah falsafah ilmu pengetahuan"* sehingga pertanyaan dianggap tidak berhubungan dengan isi buku pedoman dan dikelompokkan dalam kelas *negative*. Sistem juga tidak menghasilkan jawaban karena memang tidak ada dokumen yang dapat diproses untuk mendapatkan jawaban sehingga sampel ini digolongkan dalam *True Negative*. Hal yang serupa juga terjadi pada sampel nomor 49 dan 50 dimana isi dari pertanyaan tidak tercantum pada buku pedoman pendidikan.



BAB VII

PENUTUP

7.1 Kesimpulan

Berdasarkan implementasi dan hasil pengujian dari implementasi algoritma *Pattern Based Approach* pada *Question Answering System*, maka didapatkan kesimpulan sebagai berikut:

1. Algoritma *Pattern Based Approach* dapat diimplementasikan dengan baik pada *Question Answering System* berbahasa Indonesia dengan studi kasus Buku Pedoman Pendidikan Program Teknologi Informasi Dan Ilmu Komputer Tahun Akademik 2012/2013 dan aplikasi *Question Answering System* yang dihasilkan dari implementasi algoritma dapat digunakan sebagai sumber informasi mengenai Program Teknologi Informasi Dan Ilmu Komputer Universitas Brawijaya.
2. Hasil pengujian akurasi algoritma *Pattern Based Approach* pada *Question Answering System* dengan menggunakan 50 sampel pertanyaan mendapatkan nilai presisi sebesar 80,85% dan recall sebesar 100%.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan algoritma ini antara lain:

1. Untuk mempercepat proses eksekusi pada sistem dapat dilakukan dengan penggunaan *query database* yang lebih efisien dan juga dengan menggunakan spesifikasi perangkat keras yang lebih baik.
2. Masih perlu dikembangkannya suatu perangkat atau *tools* untuk dapat memasukkan dokumen secara otomatis ke dalam *database* sehingga proses pengumpulan dokumen tidak perlu dilakukan secara *manual* lagi.

DAFTAR PUSTAKA

- [BAS-09] Baskoro, Fajar., Rubikartin, Wisda. 2009. *Pengembangan Question/Answering Portal Dengan Answer Quality Predictor Pada Aplikasi E-Learning*. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi 2009
- [BUN-08] Bunyamin, Hendra., Negara, Chathalea Puspa. 2008. *Aplikasi Information Retrieval (IR) CATA Dengan Metode Generalized Vector Space Model*. Bandung: Universitas Kristen Maranatha.
- [FAT-09] Fatta, Hanif Al. 2009. *Natural Language Processing Dengan Teknik State Machine Parser*. Yogyakarta: STMIK AMIKOM
- [ARM-11] Armelia, Suci. 2011. *Pemilihan Passages Dalam Question Answering System Untuk Dokumen Berbahasa Indonesia*. Bogor: Institut Pertanian Bogor.
- [TOB-10] Toba, Hapnes. & Adriani, Mirna. 2010. *Pattern Based Approach in Indonesian Question- Answering System*. Depok: Universitas Indonesia.
- [SCH-07] Schlaefter, Nico. 2007. *Deploying Semantic Resources for Open Domain Question Answering*. Diploma Thesis. Language Technologies Institute School of Computer Science Carnegie Mellon University.
- [SCH-05] Schlaefter, Nico. 2005. Pattern Learning and Knowledge Annotation for Question Answering. Carnegie Mellon University, USA
Universität Karlsruhe (TH), Germany
- [FOR-08] Forner, Pamela, et. al. 2008. *OVERVIEW OF THE CLEF 2008 MULTILINGUAL QUESTION ANSWERING TRACK*.
- [ASI-05] Asian, Jelita, et. al. 2005. *Stemming Indonesian*. Australia: School of Computer Science and Information Technology RMIT University.

[ASI-07] Asian, Jelita. 2007. Effective Techniques for Indonesian Text Retrieval. Australia: School of Computer Science and Information Technology RMIT University.



Lampiran 1: Kata-Kata Stopword Vega [ASI-07]

adalah	itupun	samping
agar	jadi	saya
akan	jangan-jangan	seakan
aku	janjikan	seakan-akan
anda	kah	sebab
andaikata	kalau	sebabnya
antara	kalau-kalau	sebaliknya
apa	kalaupun	sebelum
apakah	kami	sebiji
apalagi	kamu	sebongkah
asal	kapan	sebuah
atas	kapankah	sebungkus
atau	karena	sebutir
bagaimana	kau	sedangkan
bagaimanakah	ke	seekor
bagi	kecuali	sehelai
bahkan	kemudian	sehingga
bahwa	kenapa	sejak
begitu	kepada	selagi
begitulah	ketika	selain
berkat	kita	selanjutnya
biji	lagi	selembar
bolehkan	lah	semenjak
bongkah	lalu	sementara
buah	lembar	seolah
buat	maka	seolah-olah
bungkus	malah	seorang
butir	malahan	seperti
dalam	melainkan	sepiring
dan	mengapa	seraya
dapatkah	mengapakah	serta
dari	mengenai	seseorang
daripada	menurut	sesudah
demi	mereka	setelah
demikian	meskipun	seterusnya
dengan	mula	siapa
di	mula-mula	siapakah
dia	namun	supaya
dimana	oleh	tanpa
dimanakah	orang	tempat
ekor	padahal	tentang
guna	pertama-tama	terhadap
hanya	piring	tetapi
helai	pula	untuk
hingga	pun	yaitu
ialah	sambil	yakni
itu	sampai	yang
itulah	sampai-sampai	

Lampiran 2: Pola Pertanyaan

Properti	Pola Pertanyaan	location	(manakah) <T> <C>
people	(siapa) <T> <C>	location	(mana letak) <T> <C>
people	(siapakah) <T> <C>	location	(manakah) letak <T> <C>
people	(siapa) nama <T> <C>	location	(dimana) alamat <T> <C>
people	(siapakah) nama <T> <C>	location	(dimanakah) alamat <T> <C>
people	<T> <C> (siapa)	organization	(apa) nama organisasi <T> <C>
people	<T> <C> (siapakah)	organization	(apa) nama <T> <C>
people	(siapa) yang menjadi <T> <C>	organization	organisasi (apa) <T> <C>
people	(siapakah) yang menjadi <T> <C>	organization	organisasi (apa) yang <T> <C>
people	(siapa) yang <T> <C>	measure	(berapa) panjang <T> <C>
people	(siapakah) yang <T> <C>	measure	(berapakah) panjang <T> <C>
time	(kapan) <T> <C>	measure	(berapa) lama <T> <C>
time	(kapankah) <T> <C>	measure	(berapa) lamakah <T> <C>
time	tanggal (berapa) <C> <T>	measure	(berapakah) lama <T> <C>
time	tanggal (berapakah) <C> <T>	measure	(berapa) semester <T> <C>
time	<T> <C> (kapan)	measure	(berapa) semesterkah <T> <C>
time	<T> <C> tanggal (berapa)	measure	sudah (berapa) lama <T> <C>
time	<T> <C> tanggal (berapakah)	measure	sudah (berapa) lamakah <T> <C>
time	pada semester (berapa) <T> <C>	count	(berapa) banyak <T> <C>
time	pada semeseter (berapakah) <T> <C>	count	(berapakah) banyak <T> <C>
time	(bilamana) <T> <C>	count	(berapa) banyakkah <T> <C>
location	(dimana) letak <T> <C>	count	(berapa) jumlah <T> <C>
location	(dimana) <C> <T>	count	(berapakah) jumlah <T> <C>
location	(dimana) <T> <C> berada	count	(berapa) jumlahkah <T> <C>
location	(dimanakah) letak <T> <C>	count	(berapa) <T> <C>
location	(dimanakah) <C> <T>	count	(berapakah) <T> <C>
location	(dimanakah) <T> <C> berada	count	(berapa) jumlah sks <T> <C>
location	<T> <C> ada (dimana)	count	(berapakah) jumlah sks <T> <C>
location	(dimana) saja <T> <C> berada	count	(berapa) beban studi <T> <C>
location	(apa) nama tempat <T> <C>	count	(berapakah) beban studi <T> <C>
location	(apakah) nama tempat <T> <C>	count	(berapa) jumlah sks mata kuliah <T> <C>
location	(apa) nama lokasi <T> <C>		
location	(apakah) nama lokasi <T> <C>		
location	(mana) <T> <C>		

count	(berapakah) jumlah sks mata kuliah <T> <C>	object	(apa) sajakah <T> <C>
count	(berapa) beban studi mata kuliah <T> <C>	object	(apa) <T> mata kuliah <C>
count	(berapakah) beban studi mata kuliah <T> <C>	object	(apakah) <T> mata kuliah <C>
object	(apa) <T> <C>	other	(bagaimana) <T> <C>
object	(apakah) <T> <C>	other	(bagaimanakah) <T> <C>
object	<T> <C> (apa)	other	(bagaimana) cara <T> <C>
object	<T> <C> (apakah)	other	(bagaimanakah) cara <T> <C>
object	(apa) <T> <C> itu	other	(bagaimana) bunyi <T> <C>
object	(apakah) <T> <C> itu	other	(bagaimanakah) bunyi <T> <C>
object	(apa) yang dimaksud dengan <T> <C>	other	(bagaimana) cara untuk <T> <C>
object	(apakah) yang dimaksud dengan <T> <C>	other	(bagaimanakah) cara untuk <T> <C>
object	(apa) saja <T> <C>		



Lampiran 3: Pola Jawaban

Properti	Pola Jawaban
people	<T> <C> adalah <P>
people	<T> <C> bernama <P>
people	<T> <C> ialah <P>
people	<T> <C> yaitu <P>
people	<T> <C> sebagai berikut <P>
time	<T> <C> adalah <P>
time	<T> <C> pada <P>
time	<T> <C> pada tanggal <P>
time	<T> <C> pada hari <P>
time	<T> <C> pada jam <P>
time	<T> <C> adalah pada <P>
time	<T> <C> pada waktu <P>
location	<T> <C> terletak di <P>
location	<T> <C> ada di <P>
location	<T> <C> berada di <P>
location	<T> <C> di <P>
location	<T> <C> adalah <P>
organization	<T> <C> adalah <P>
organization	<T> <C> antara lain <P>
measure	<T> <C> adalah <P>
measure	<T> <C> sekitar <P>
count	<T> <C> adalah <P>
count	jumlah sks <T> <C> adalah <P>
count	beban studi <T> <C> adalah <P>
count	<T> <C> sebanyak <P>
count	<T> <C> sejumlah <P>
object	<T> <C> adalah <P>
object	<T> <C> antara lain <P>
object	<T> <C> sebagai berikut <P>
object	<T> <C> yaitu <P>
object	<T> <C> ialah <P>
other	<T> <C> berbunyi <P>
other	<T> <C> adalah <P>
other	<T> <C> antara lain <P>