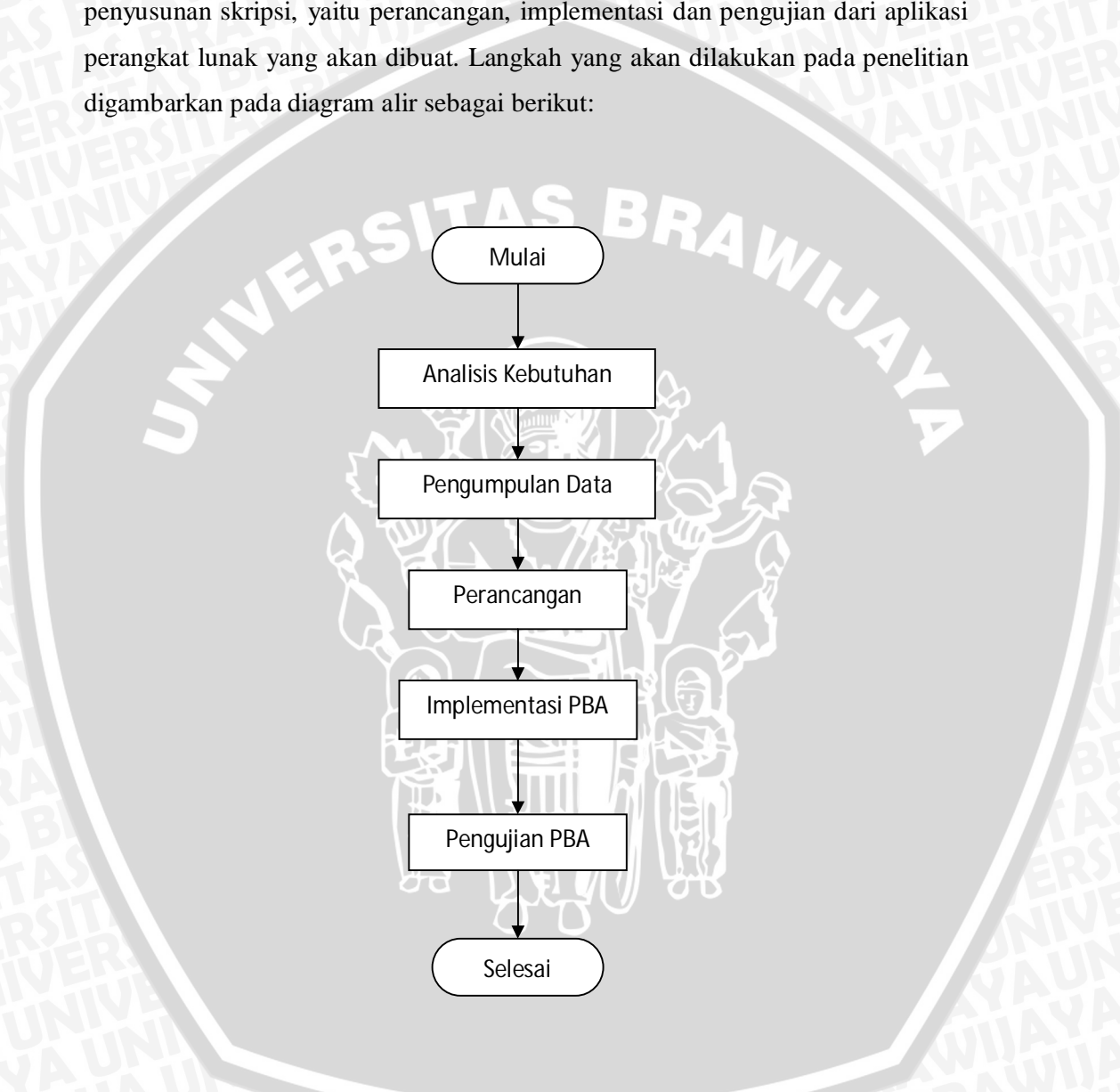


### BAB III

#### 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*.

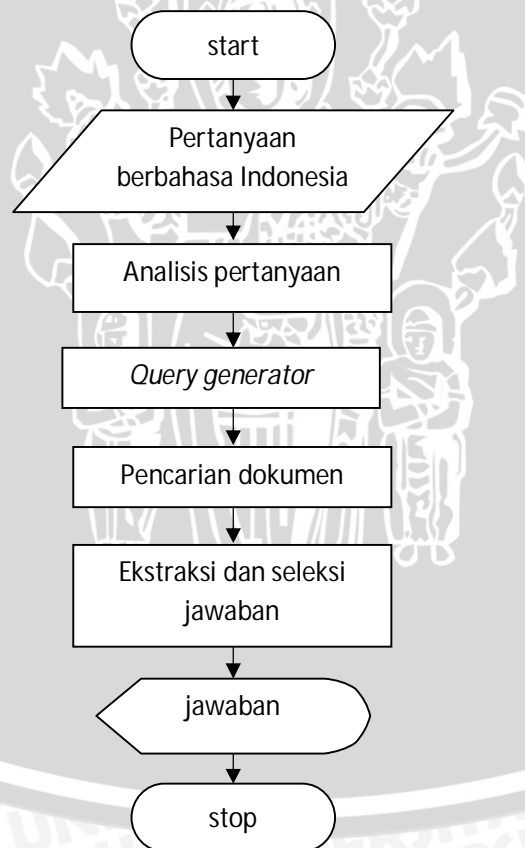
### 3.3 Perancangan Sistem

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.

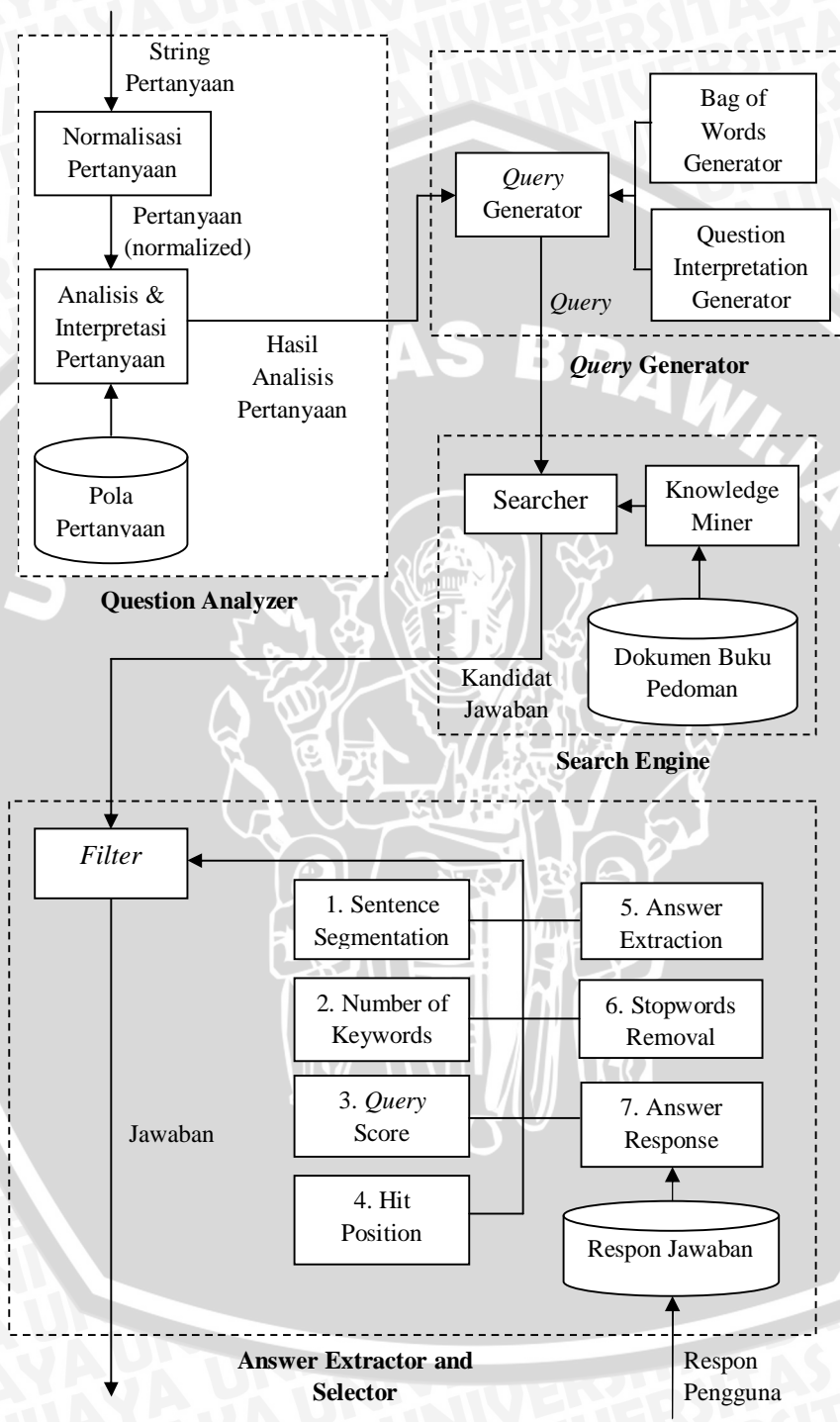
Untuk mempermudah alur jalan dari sistem, maka semua proses yang dijelaskan sebelumnya digambarkan dalam diagram alir berikut:



**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> )	+	<b>TP</b> ( <i>True Positive</i> )	<b>FP</b> ( <i>False Positive</i> )
	-	<b>FN</b> ( <i>False Negative</i> )	<b>TN</b> ( <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:

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