

TEXT MINING UNTUK AKUISISI PENGETAHUAN SECARA OTOMATIS PADA SISTEM PAKAR

Ardi ¹

¹ Alumnus PPSTI MKOM Universitas Budi Luhur

Abstrak

Automatic knowledge acquisition method from documented source of expert system, now is still residing at under done development stage. This research developed a new method to do automatic knowledge acquisition from text of expert system by combines text mining and neural network technology. The design of research consists of three steps, that is: development of the method, development of the prototype and accuracy level evaluation of the produced knowledge base.

Automatic knowledge acquisition method from documented source, which developed at this research consist of six steps, that is: document parsing, topics retrieval, topics clustering, domain extraction, item extraction and solution extraction. Document parsing applied with parsing method of natural language processing to do POS tagging on text. Topics retrieval developed based on information retrieval technology to find studied topics/phrase of text. Topics clustering developed based on self organizing maps technology to group relevant and non relevant topics/phrase. Domain extraction, item extraction and solution extraction developed based on information extraction technology and knowledge graph theory. Domain extraction extracted information of a relevant topic/phrase on text. Item extraction extracted indication information of a domain. Solution extraction extracted solutions information of a domain. Produced knowledge base will represent with object oriented. Inference engine developed based on dempster-shafer theory and fabric fault advisory expert system journal as reference. Automatic knowledge acquisition system prototype and expert system prototype developed with object oriented software engineering approach. Evaluation result show that the produced knowledge base accuracy level is reach 0.6563 and accuracy level can be increase by adding number of knowledge source document.

Keywords : Automatic Knowledge Acquisition, Expert System, Object Oriented Knowledge Base Representation, Inference Engine, Text Mining, Neural Network.

PENDAHULUAN

Latar Belakang Masalah

Dengan akuisisi otomatis dari sumber terdokumentasi, kebutuhan terhadap pakar dapat dikurangi bahkan dapat dihilangkan. Basis pengetahuan sistem pakar dapat dibangun hanya dari beberapa text file yang berisi pengetahuan yang relevan dengan domain sistem pakar. Pada dasarnya bidang ini masih dalam tahap pengembangan metode baru yang belum matang. Penelitian ini mengembangkan metode tersebut dan mengusulkan sebuah metode akuisisi pengetahuan otomatis dari sumber terdokumentasi pada sistem pakar dengan menggunakan teknologi text mining yang

memanfaatkan neural network dan mengembangkan prototype sistem akuisisi pengetahuan otomatis dari sumber terdokumentasi.

Batasan Masalah

1. Akuisisi pengetahuan otomatis dibatasi pada sistem pakar jenis sistem interpretasi, sistem diagnostik, sistem *debugging* dan sistem perbaikan yang bersifat simbolik.
2. Akuisisi pengetahuan otomatis dibatasi hanya untuk menghasilkan basis pengetahuan sistem pakar dengan kompleksitas yang rendah dan ruang lingkup yang tidak terlalu luas.

3. *Text Mining* dilakukan pada *text file* yang telah ditentukan oleh *knowledge engineer* dan diasumsikan bahwa *text file* tersebut cukup relevan dengan *domain* pengetahuan sistem pakar yang akan dikembangkan.
4. *Text mining* yang dilakukan hanya dapat mengenali bahasa inggris, sehingga pengetahuan sistem pakar yang dihasilkan juga akan berbahasa inggris.
5. Mekanisme inferensi dibatasi hanya menggunakan metode inferensi *backward chaining*.

Tujuan

1. Mengembangkan metode akuisisi pengetahuan otomatis dari sumber terdokumentasi pada sistem pakar dengan mengkombinasikan teknologi *text mining* yang memanfaatkan *neural network*.
2. Merepresentasikan basis pengetahuan dan mengembangkan mesin inferensi pada sistem pakar untuk menggunakan basis pengetahuan yang ditelaah dihasilkan.
3. Mengembangkan *prototype* sistem akuisisi pengetahuan otomatis untuk menguji metode yang dikembangkan, dan *prototype* sistem pakar untuk menguji basis pengetahuan yang dihasilkan.
4. Mengukur tingkat akurasi basis pengetahuan sistem pakar yang dihasilkan dari proses akuisisi pengetahuan otomatis dari sumber terdokumentasi.

Manfaat

1. Dapat mengurangi biaya pengembangan sistem pakar dengan menghilangkan kebutuhan terhadap pakar dan mempercepat proses pengembangan sistem pakar
2. Sumbangsih bagi ilmu pengetahuan bidang artificial intelligence khususnya pada sistem pakar dengan memperkenalkan metode akuisisi pengetahuan otomatis dari sumber terdokumentasi untuk sistem pakar pada penelitian ini.

TINJAUAN PUSTAKA

Natural Language Processing
(NLP)_{[ANL01],[DNL01],[DNL02]}

POS *Tagging* adalah sebuah proses yang mengelompokkan kata ke sebuah penanda kelas dalam sebuah kalimat. Kata dikelompokkan menjadi: kata benda, kata kerja, kata sifat, kata keterangan dan kata depan. POS dapat dibagi menjadi 2 kategori yaitu *closed class* dan *open class*. Yang termasuk *open class* pada bahasa inggris, yaitu kata benda, kata kerja, kata sifat dan kata keterangan. *Closed class* pada bahasa inggris: *prepositions, particles, determiners, pronouns, conjunctions, auxiliary verbs, numerals*.

CFG adalah bentuk formal yang memungkinkan pemodelan fakta-fakta *constituent*. Sebuah CFG terdiri atas beberapa aturan dan produksi yang masing-masing mengekspresikan cara yang merupakan simbol bahasa dapat dikelompokkan dan diurutkan bersama, dan sebuah kamus kata dan simbol. CFG merupakan model sistem matematika yang sering digunakan untuk memodelkan struktur unsur dalam bahasa inggris dan bahasa alami yang lain. Frase kata benda pada bahasa inggris dapat mempunyai determinan, angka, pengukur, frase kata sifat yang mendahului kata benda, yang dapat diikuti dengan sebuah angkat dari *postmodifiers, gerundive VPs, invinitives VPs* dan *past participle* adalah kemungkinan yang paling umum. Subjek dalam bahasa inggris dapat diikuti dengan kata kerja utama pada orang dan angka. Kata kerja dapat di sub-kategorikan oleh tipe komplemen yang diharapkan. Sub-kategorisasi yang sederhana adalah *transitive* dan *intransitive*. Sub-kelas dari kata kerja adalah kata penghubung (*auxiliaries*) atau kata kerja bantu mempunyai batasan *syntactic* tertentu yang dapat dilihat sebagai salah satu jenis sub-kategorisasi. *Auxiliaries* memasukkan kata kerja *modal can, could, may, might, must, will, would, shall, should, have, be*.

Parsing dengan CFG mengarah pada proses yang menugaskan pemodelan pohon yang benar pada input *string*, dimana dapat menutupi semua kalimat dan mempunyai sebuah node 'S' pada bagian paling atas. Terdapat 2 batasan yang dapat digunakan untuk membantu pencarian yaitu batasan berasal dari data dan berasal dari *grammar*. Pada akhir proses *parsing*, pada pohon harus

selalu mempunyai 1 buah akar yang selalu dimulai dengan simbol S. Terdapat 2 jenis strategi pencarian parsing yaitu : *top-down parsing* dan *bottom-up parsing*. *Top-down parsing* dimulai simbol mulai S dan mencoba menguraikan input kalimat dengan mengkonstruksikan pohon uraian yang mendeskripsikan bagaimana *grammar* digunakan untuk menghasilkan kalimat. *Bottom-up parsing* dimulai dengan sebuah kata dari input dan mencoba untuk membangun pohon dari kata, dengan menerapkan aturan dari *grammar*.

Information Retrieval (IR)_{[ANL01],[EIR01],[EIR02],[EIR03],[EIE02]}

Information retrieval adalah salah satu metode *text mining* untuk menemukan informasi yang relevan dalam satu atau beberapa dokumen berupa teks yang isinya tidak terstruktur dan menggunakan bahasa alami manusia berdasarkan *query* atau *keyword* tertentu. *Vector space model* adalah salah satu model *retrieval information retrieval* yang sering digunakan. *Term-weighting* adalah sebuah metode yang diimplementasikan pada *vector space model* yang digunakan untuk mengisi *term weight* dalam dokumen dan *query vector* dimana setiap *term-weight* dihitung berdasarkan skema TF-IDF.

$$TF_i = f_{ij} / \max\{f_{ij}\}$$

$$IDF_i = \log_2(N / DF_i)$$

$$TF-IDF_i = TF_i * IDF_i$$

Dimana:

- TF = *Term Frequency*
- F_{ij} = Frekuensi kemunculan *term* i dalam dokumen j.
- max{f_{ij}} = Total frekuensi kemunculan *term* i pada keseluruhan dokumen
- IDF = *Inverse Document Frequency* dari *term* i.
- DF = Jumlah dokumen dimana *term* i muncul.
- N = Jumlah keseluruhan dokumen.

Similarity measure adalah sebuah fungsi yang menghitung derajat kemiripan antara 2 buah vektor.

$$\text{cosine}(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j, \mathbf{q} \rangle}{\|\mathbf{d}_j\| \times \|\mathbf{q}\|} = \frac{\sum_{i=1}^{|\mathcal{I}|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|\mathcal{I}|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|\mathcal{I}|} w_{iq}^2}}$$

W_{ij} = jumlah *term* i pada dokumen j.
 W_{iq} = TF-IDF *query* i.

Algoritma:

1. Hitung TF-IDF *term* terhadap dokumen (D_j) pada keseluruhan dokumen (D).
2. Konversi *query* menjadi TF-IDF (Q).
3. Untuk setiap D_j in D → hitung nilai S_j = cosine(D_j, Q)
4. Urutkan dari *score* yang terkecil.

Text pre-processing berfungsi untuk melakukan penyaringan terhadap kata-kata dalam teks sebelum pemrosesan *information retrieval*. Terdapat beberapa metode filterisasi pada *information retrieval*.

1. *Word extraction*. String yang spesifik dan tidak mempunyai arti dapat dikeluarkan.
2. *Stop words removal*. Kata-kata dengan frekuensi tertinggi dan bersifat umum akan dikeluarkan.
3. *Steaming*. Digunakan untuk mencari bentuk dasar sebuah kata, dengan menyatukan kata yang mempunyai *morphology*, demikian jumlah kata dapat direduksi.

Information retrieval dapat dioptimasi dengan menambahkan *thesaurus*. Untuk setiap *term* t dalam *query*, *query* diperlebar dengan sinonim dan kata yang berhubungan dengan t dari kamus. Secara umum *thesaurus* dapat meningkatkan *recall*. Secara signifikan dapat mengurangi *precision*, *particular* dengan *term* yang ambigu.

Akurasi *information retrieval* dapat dievaluasi berdasarkan penilaian dari manusia berdasarkan *output* yang dihasilkan. Hasil evaluasi sangat bersifat sangat subjektif. Akurasi dapat dihitung berdasarkan *precision* dan *recall*.

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}$$

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}$$

$$\text{-Measure} = (2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

Self Organizing Maps (SOM)_{[ANN01],[ANN02],[ANN03],[ANN04]}

Self organizing maps merupakan salah satu metode pembelajaran pada *neural network* tidak terawasi (*unsupervised learning*) yang tidak memerlukan target *output* pada proses pembelajaran.

Pada jaringan ini, suatu lapisan yang berisi *neurons* akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu kelompok yang dikenal dengan istilah *cluster*. Selama proses penyusunan diri, *cluster* yang memiliki vektor bobot paling cocok dengan pola input (memiliki jarak yang paling dekat) akan terpilih sebagai pemenang. *Neuron* yang menjadi pemenang akan memperbaiki bobotnya.

Algoritma:

1. Tentukan *learning rate*, tentukan *learning radius*, tentukan jumlah *neuron*, tentukan iterasi. Inisialisasi bobot *w* secara acak untuk setiap *neuron*.
2. Kerjakan selama kondisi < iterasi:
 - a. Untuk setiap vektor *input* *x* terhadap seluruh *neuron*, kerjakan:

$$D(j) = \sum (w_{ij} - x_i)^2$$
 - b. Tentukan *neuron* dengan hasil terkecil dari *D(j)*.
 - c. Untuk setiap unit *neuron* *j* perbaiki bobot:

$$w_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \alpha (x_i - w_{ij}(\text{lama}))$$
3. Perbaiki *learning rate*.

Information Extraction (IE)

_{[BNL01],[EIE01],[EIE02],[EIE03],[EIE04],[EIE05]}

Information extraction adalah salah satu metode *text mining* untuk melakukan ekstraksi data dari *text file* yang tidak terstruktur. *Information extraction* mengidentifikasi potongan-potongan spesifik dari informasi dari sebuah dokumen textual yang tidak terstruktur atau semi terstruktur. *Information extraction* melakukan transformasi dari informasi yang tidak terstruktur dalam dokumen ke dalam *database* yang terstruktur.

Information extraction terdiri atas tiga proses yaitu segmentasi, klasifikasi, asosiasi dan klusterisasi. *Information extraction* dapat diaplikasikan pada jenis teks yang berbeda. Terdapat beberapa macam model teknik

information extraction untuk melakukan proses tersebut, yaitu *lexicon*, *classify pre-segment candidat*, *sliding window*, *boundary models*, *finite state automata* dan *CFG*.

Slot pada *template* biasanya diisi dengan *substring* dari dokumen. Beberapa slot mungkin memiliki sekumpulan pengisi pra-spesifik konstan yang mungkin muncul dalam teks tersebut. Beberapa slot dimungkinkan diisi oleh beberapa *filler*.

Pola ekstraksi sederhana dapat dispesifikasikan dengan menggunakan pola *Regular Expression*. Pola *Pre-filler* dapat digunakan untuk mengidentifikasi pola kata/kalimat sebelum pola *filler*. Dapat juga digunakan pola *post-filler* untuk mengidentifikasi akhir dari *filler*. Jika slot mempunyai sekumpulan pengisi pra-spesifik, kategorisasi teks dapat digunakan untuk mengisi slot.

Evaluasi tingkat akurasi *information extraction* dilakukan secara terpisah, data pengujian yang tidak digunakan selama pengembangan sistem. Pengukuran dilakukan pada setiap dokumen menggunakan pengukuran yang diadaptasikan dari *information retrieval* sebagai berikut :

$$\text{Recall} = \frac{\text{Number of extracted slot/value pairs that are correct}}{\text{Total number of correct extractions in the solution template}}$$

$$\text{Precision} = \frac{\text{Number of extracted slot/value pairs that are correct}}{\text{Total number of slot/value pairs extracted by the system}}$$

$$F\text{-Measure} = \frac{2 * \text{precision} * \text{recall}}{(\text{precision} + \text{recall})}$$

Dempster-Shafer Theory_{[AES01],[ANN03]}

Salah satu model yang sering digunakan untuk ketidakpastian dalam sistem pakar adalah teori *dempster-shafer*. Teori ini menunjukkan suatu cara untuk memberikan bobot keyakinan sesuai dengan fakta yang dikumpulkan.

Secara umum teori *dempster-shafer* ditulis dalam bentuk interfal:

[*Belief*, *Plausability*].

Belief (Bel) adalah ukuran kekuatan *evidence* dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada *evidence*, dan jika bernilai 1 menunjukkan adanya kepastian.

Plausability (Pl) dinotasikan sebagai:

$$Pl(s) = 1 - Bel(-s)$$

Plausability juga bernilai 0 sampai 1. Jika kita yakin akan *_s*, maka dapat dikatakan bahwa $Bel(_s)=1$, dan $Pl(-s)=0$. Pada teori *dempster-shafer* kita mengenal adanya *frame of discernment* yang dinotasikan dengan θ . *Frame* ini merupakan semesta pembicaraan dari sekumpulan hipotesis.

Object Oriented Software Engineering (OOSE)

Belakangan ini muncul model proses *agile* yang merupakan proses yang adaptif, fokus pada hasil dan cenderung menggunakan iterasi yang singkat. *Extreme Programming* (XP) adalah salah satu contoh model proses *agile* yang merupakan metodologi ringan untuk proyek skala kecil sampai menengah. XP mendominasi *coding* sebagai kunci aktivitas dalam keseluruhan proyek.

Tahapan analisis di tuju untuk menganalisis, menspesifikasikan dan mendefinisikan sistem yang akan dibangun. Dalam proses *Object Oriented Analysis* (OOA) juga dilakukan pengumpulan persyaratan untuk menciptakan skenario *use case*.

Object Oriented Design (OOD) mentransformasikan model analisis OOA ke dalam model desain dan menemukan. *Design pattern* merupakan sebuah strategi desain kelas dan objek dalam OOD agar fleksibel, *reusable, generic, independent*.

Coding merupakan tahap yang menerjemahkan hasil OOD ke dalam bahasa pemrograman berorientasi objek. *Refactoring* adalah sebuah perubahan yang dilakukan pada struktur internal perangkat lunak tanpa mengubah *behavior*.

Knowledge graph theory and structural parsing [BNL01]

Knowledge graph adalah salah satu jenis sudut pandang baru yang digunakan untuk mendeskripsikan bahasa manusia dan lebih fokus pada semantik dari pada aspek sintaksis. *Knowledge graph* adalah sebuah metode representasi pengetahuan yang baru berdasarkan pada jaringan semantic.

Dengan interpretasi dari sebuah kalimat, sebagai pernyataan dari sebuah *graph*, sebagai panduan untuk mencoba mencari

indikator *chunks*. Berikut ini adalah *chunks indicators* yang digunakan pada *knowledge graph* :

1. Indicator 0: Sekumpulan koma (',') dan/atau tanda periode.
2. Indicator 1: *Frame word*, termasuk kata kerja bantu.
3. Indicator 2: Acuan kata-kata.
4. Indicator 3: "*Jumps*", berkenaan dengan *grammar*.
5. Indicator 4: Kata penghubung, termasuk kata depan.

Integrating an object oriented approach and rule-based reasoning in the design of a fabric fault advisory expert system. [CES01]

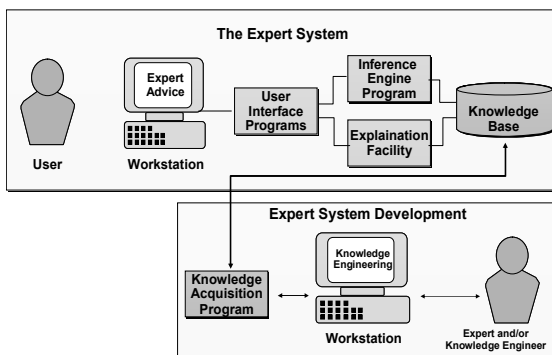
Paper ini menjelaskan tentang desain sebuah sistem pakar yang tidak hanya memberikan nasihat pada pengguna tekstil pada *fabric fault* tetapi juga pada pengguna non-tekstil, tetapi juga membantu untuk mendiagnosa *fabric fault*. Sistem pakar ini mengintegrasikan pendekatan berorientasi objek dan *rule-based reasoning*. Sistem ini berdasarkan pada premis yang merupakan desain dari komponen yang harus di pisahkan dari implementasi detail. Secara singkat, dijelaskan bahwa rules yang memanipulasi satu set object dipisahkan dari satu set objek yang sama. Pemisahan definisi objek dari rules yang memanipulasi mereka pada basis pengetahuan, dan pemisahan pengetahuan berbasis objek dari mesin inferensi berbasis object yang melakukan inferensi kondisi, artinya suatu perubahan yang dilakukan pada salah satu dari komponen ini tidak berarti perubahan juga harus dilakukan pada komponen lain, dan juga setiap komponen ini autonomous. Ketergantungan antar component dalam sebuah object pada pendekatan berorientasi objek, kemampuan sistem semacam ini, properties yang diwariskan, penggunaan kembali sebuah object adalah keuntungan yang sama dalam mengadopsi sebuah pendekatan berorientasi objek.

Sistem Pakar

Sistem pakar merupakan suatu sistem informasi yang menangkap dan menggunakan pengetahuan serta metode pengambilan keputusan yang digunakan oleh seorang atau

beberapa orang ahli dalam bidang keahlian tertentu. Sistem pakar didasarkan pada sistem pengetahuan, sehingga memungkinkan komputer dapat berfikir dan mengambil keputusan atau kesimpulan dari sekumpulan kaidah.

Berdasarkan jenis dan tipenya, sistem pakar memiliki banyak variasi antara lain: sistem pakar vs sistem berbasis pengetahuan, sistem pakar berbasis aturan, sistem berbasis *frame*, sistem *hybrid*, sistem berbasis model, sistem *off-the-shelf* dan sistem pakar *real-time*. Berdasarkan fungsinya, sistem pakar dapat diklasifikasikan dalam beberapa cara, salah satu cara adalah berdasarkan area persoalan umum yang ditanganinya.



Gambar 1. Struktur sistem pakar. ([ASE01], 24, dimodifikasi).

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecahan masalah dari beberapa sumber pengetahuan terdokumentasi ke program komputer untuk konstruksi atau perluasan basis pengetahuan. Metode akuisisi dapat dilakukan secara Otomatis, yaitu pengetahuan dihasilkan dari fakta yang ada. Dalam metode ini, peran pakar dan *knowledge engineer* hampir tidak diperlukan. Metode yang dapat digunakan untuk akuisisi pengetahuan otomatis adalah induksi dan akuisisi dari sumber terdokumentasi. Terdapat dua alasan utama penggunaan akuisisi pengetahuan otomatis, yaitu: *knowledge engineer* yang baik sangat mahal dan sulit didapat, dan pakar *domain* biasanya sibuk dan kadang tidak kooperatif. Akibatnya metode elisitasi manual bahkan semiotomatis lambat dan mahal. Pengetahuan sering dapat diperoleh dari sumber lain selain atau pengganti pakar

manusia. Pengetahuan terdokumentasi memiliki potensi besar untuk otomatisasi. Pendekatan ini digunakan dalam sistem berbasis pengetahuan di mana penanganannya sejumlah besar informasi kompleks lebih diutamakan daripada keahlian tingkat dunia. Pengetahuan yang diperoleh dari sumber terdokumentasi sering disebut sebagai *text mining*. Pengetahuan dari sumber terdokumentasi dapat dianalisis secara manual atau dengan menggunakan teknologi *Artificial Intelligence* (gabungan *natural language processing*, agen cerdas dan sistem pakar), sehingga dengan demikian sistem cerdas dapat digunakan untuk membangun sistem pakar lain. Data yang disimpan dalam sistem komputer lain dapat diambil secara elektronik untuk membuat atau memperbaharui basis pengetahuan pada sistem pakar, tanpa campur tangan *knowledge engineer* atau pakar manusia. Bidang ini pada dasarnya berada pada tahap pengembangan metode baru yang menginterpretasikan arti untuk menentukan aturan dan bentuk lain pengetahuan, misalnya *frame* untuk pertimbangan berbasis kasus. Sejumlah metode baru dikembangkan dan diimplementasikan.

Basis pengetahuan berisi pengetahuan penting untuk pengertian, formulasi, dan pemecahan masalah. Basis pengetahuan memasukkan dua elemen dasar, yaitu: *Facts*, seperti situasi masalah dan teori dari area masalah. Heuristic khusus atau *rules* yang menghubungkan penggunaan pengetahuan untuk pemecahan masalah spesifik dalam sebuah *domain* khusus. Basis pengetahuan harus direpresentasikan dalam format yang dapat dipahami oleh manusia dan dieksekusi pada komputer. Terdapat banyak metode untuk representasi pengetahuan. Yang paling populer adalah aturan produksi, *frame*, pohon keputusan, objek dan logika juga berguna dalam beberapa kasus.

Frame. Merupakan struktur data yang menyertakan semua pengetahuan tentang objek tertentu. *Frame* pada dasarnya adalah aplikasi pemrograman berorientasi objek untuk *artificial intelligence* dan sistem pakar. *Frame* mencakup dua elemen dasar yaitu *slot* dan *facet*. *Slot* adalah set atribut yang

mendeskripsikan objek yang direpresentasikan oleh *frame*.

Representasi pengetahuan berorientasi objek. Merupakan pendekatan lanjutan yang mengadopsi banyak konsep *frame* yang telah lama menjadi norma dalam analisis dan desain. Pada dasarnya *frame* adalah objek dan sesuai dengan paradigma pendekatan berorientasi objek terhadap pengembangan sistem pakar. Objek mengenkapsulasi properti dan tindakan, seperti halnya *frame* menyimpan pengetahuannya. Selanjutnya, *frame* diatur dalam kelas, yang diatur ke dalam hierarki. Tiap *frame* mewarisi propertinya dari *frame* induk, seperti halnya objek mewarisi properti. Aturan dapat memadukan kelakuan *frame* atau dilekatkan ke dalam *frame*. Paradigma berorientasi objek mencocokkan struktur sistem pakar *hybrid* untuk bekerja dengan *frame* dan aturan. Objek memilih set aturan untuk diterapkan pada situasi tertentu.

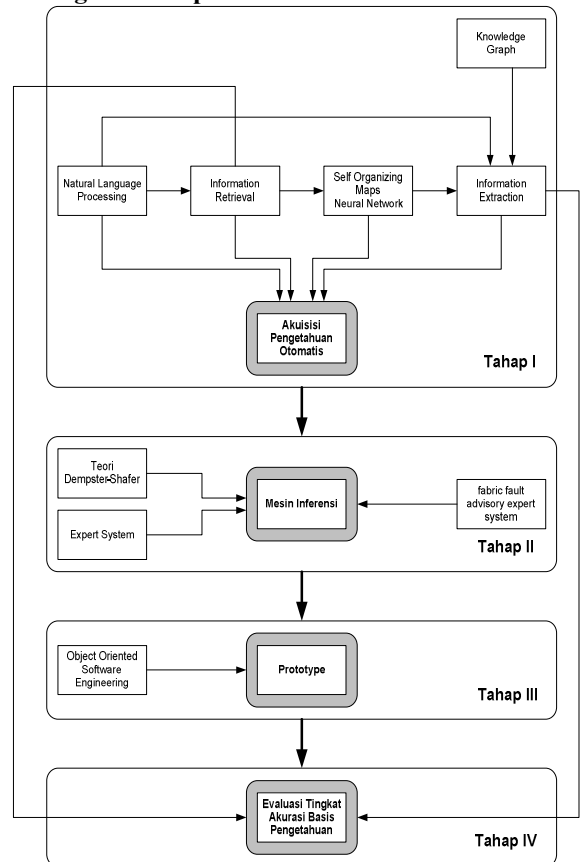
Mesin inferensi menyediakan metodologi untuk *reasoning* tentang informasi dalam basis pengetahuan dan untuk formulasi kesimpulan. Komponen ini memberikan arah tentang bagaimana menggunakan pengetahuan sistem dengan membangun agenda yang mengorganisir dan mengontrol langkah-langkah yang diambil untuk pemecahan masalah. Dua pendekatan populer untuk menarik kesimpulan adalah metode *forward chaining* dan *backward chaining*. *Forward chaining* mencocokkan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF), setelah semua kondisi terpenuhi, aturan dipilih untuk mendapatkan kesimpulan. Pada *backward chaining* pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN). Pendekatan ini dimulai dari kesimpulan dan hipotesis bahwa kesimpulan adalah benar.

Sistem penjelasan adalah Kemampuan melacak pertanggungjawaban untuk konklusi ke sumber adalah krusial dalam transfer keahlian dan dalam pemecahan masalah. Subsistem penjelasan dapat melacak tanggungjawab dan penjelasan perilaku sistem pakar.

Sistem perbaikan pengetahuan adalah evaluasi pengetahuan sistem pakar diperlukan untuk pembelajaran komputer sehingga

program dapat menganalisis alasan keberhasilan atau kegagalannya sehingga mengarah pada peningkatan kualitas basis pengetahuan. Komponen tersebut tidak tersedia dalam sistem pakar komersil saat ini, tetapi sedang dikembangkan dalam sistem pakar eksperimental pada beberapa universitas dan lembaga riset.

Kerangka Konsep



Gambar 2. Kerangka konsep.

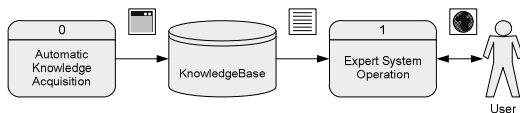
METODE PENELITIAN

Penelitian ini akan mengusulkan sebuah metode baru untuk melakukan akuisisi pengetahuan otomatis dari sumber terdokumentasi pada sistem pakar. Metode tersebut akan di implementasikan lewat pengembangan *prototype* dan hasil *output* dari *prototype* akan dievaluasi melalui rumus pengukuran tingkat akurasi tertentu.



Gambar 3. Desain penelitian.

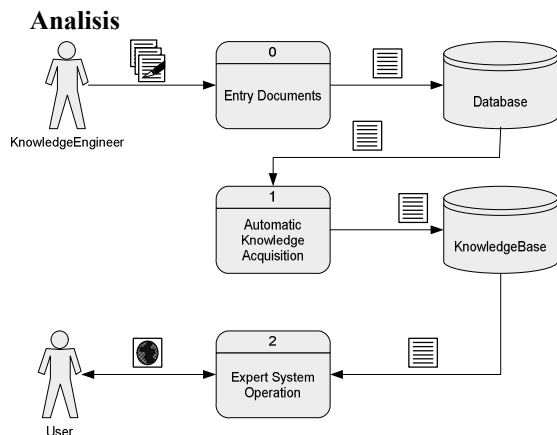
Secara keseluruhan sistem pakar akan menggunakan basis pengetahuan hasil dari proses akuisisi pengetahuan otomatis. Penelitian ini menekankan pada penggunaan teknologi *Text Mining* yang memanfaatkan teknologi *Neural Network* untuk melakukan proses akuisisi pengetahuan otomatis pada sistem pakar. Berikut adalah ringkasan singkat penelitian ini.



Gambar 4. Ringkasan penelitian.

Proses penelitian dibagi menjadi 3 tahap, yaitu tahap pengembangan model/metode, tahap pengembangan *prototype* sistem akuisisi pengetahuan otomatis dan *prototype* sistem pakar dan tahap ketiga adalah evaluasi tingkat akurasi pengetahuan.

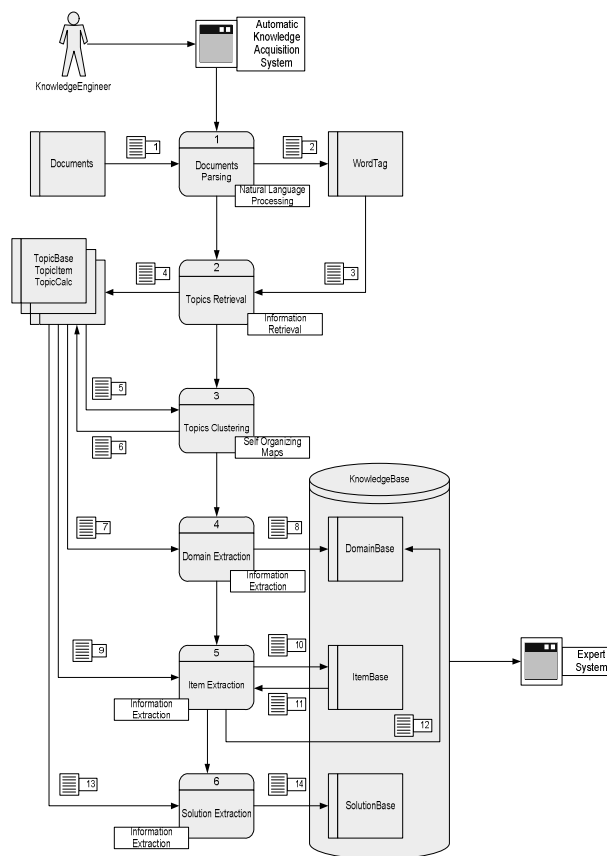
ANALISIS DAN INTERPRETASI



Gambar 5. Gambaran sistem secara umum.

Secara umum terdapat tiga proses besar yang terjadi dalam model yang dikembangkan pada penelitian ini. Proses tersebut adalah proses pemasukan dokumen, proses akuisisi pengetahuan otomatis dan proses operasi sistem pakar. Proses pemasukan dokumen dilakukan oleh *knowledge engineer* ke dalam *database* sistem. Kemudian berdasarkan dokumen dalam *database* tersebut dilakukan proses akuisisi pengetahuan otomatis. Hasil akuisisi kemudian akan disimpan kembali ke *database* sebagai basis pengetahuan yang akan digunakan oleh sistem pakar untuk berinteraksi dengan pengguna.

Pengembangan Text Mining Tahapan proses akuisisi pengetahuan otomatis



Gambar 6. Diagram konteks proses akuisisi pengetahuan otomatis.

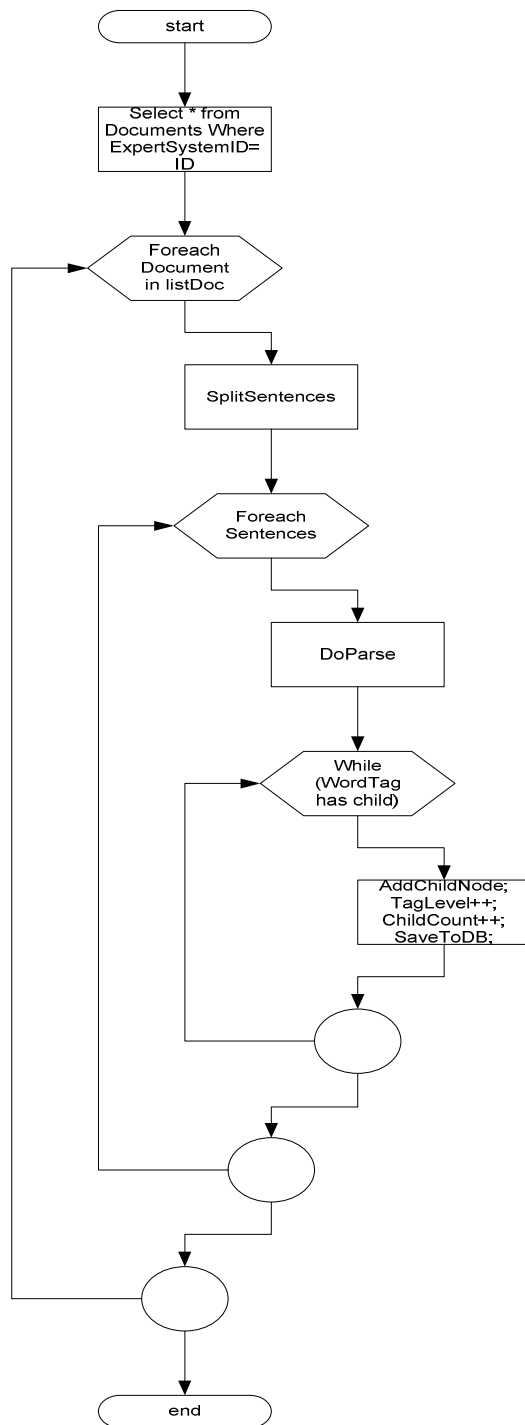
Seperti yang terlihat pada gambar 6, bahwa *file* yang termasuk pada kategori basis pengetahuan sistem pakar ada tiga yaitu *DomainBase file*, *ItemBase file* dan *SolutionBase file*. Untuk mengisi ketiga *file* basis pengetahuan tersebut diperlukan masing-masing satu proses yang juga termasuk dalam proses akuisisi pengetahuan otomatis.

Pada proses akuisisi pengetahuan otomatis, langkah awal akan dilakukan *knowledge engineer* sebagai aktor tunggal dalam proses ini. *Knowledge engineer* hanya melakukan klik tombol proses yang tersedia pada sistem dan sistem akan melakukan keenam proses tersebut. Tahapan proses tersebut, yaitu: *documents parsing*, *topics retrieval*, *topics clustering*, *domain extraction*, *item extraction*, *solution extraction*.

Documents parsing

Merupakan proses yang melakukan *POS tagging* pada setiap kalimat dalam keseluruhan dokumen sumber pengetahuan. *Parsing* dilakukan dari *file* dokumen yang sebelumnya telah di-input oleh *knowledge engineer*. Hasil proses ini akan disimpan pada *WordTag file* berupa kata-kata dan frase dalam dokumen tersebut yang telah diurai berdasarkan hasil *POS tagging*.

Tahapan ini menggunakan metode *parsing with CFG* pada teori *natural language processing* untuk melakukan proses *parsing* kalimat pada semua dokumen. Proses *parsing* ini berguna untuk mencari setiap kata benda pada kalimat, karena setiap *domain* pada sebuah permasalahan pasti merupakan kata benda. Proses pencarian dilakukan dengan memberi *tag* pada setiap frase, dan frase yang merupakan kata benda mempunyai *tag* 'NP' atau 'NN'.

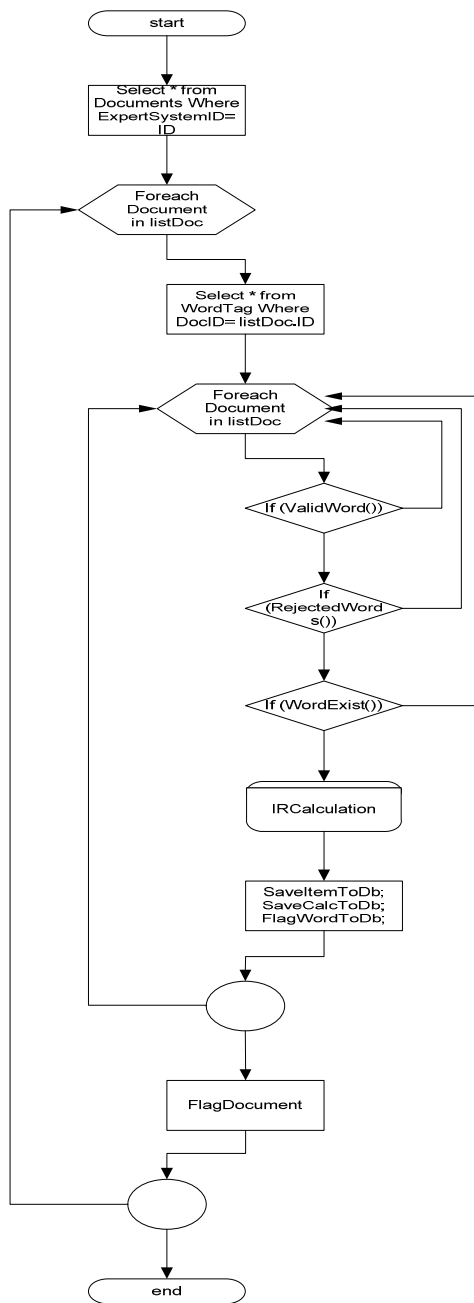


Gambar 7. Flowchart proses document parsing.

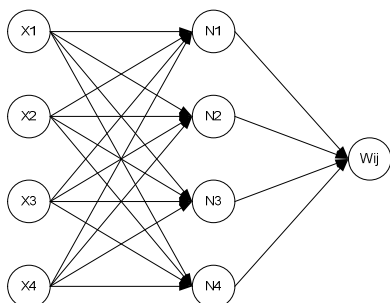
Topics retrieval

Merupakan proses yang menemukan topik-topik apa saja yang dibahas pada keseluruhan dokumen. Topik tersebut dapat berupa kata, frase kata atau kalimat singkat. Topik-topik tersebut akan dicari pada *WordTag file* dan hasil dari proses ini akan disimpan pada *TopicBase file* dan setiap kata/frase pada *WordTag file* akan disimpan pada *TopicItem file* sebagai relasi *child*. Selama proses pencarian, setiap kata, frase atau kalimat tersebut akan diberikan nilai, hasil perhitungan tersebut akan disimpan pada *TopicCalculation file*. Beberapa nilai akan dinormalisasi dari *TopicCalculation file* dan disimpan pada *TopicBase file*.

Tahapan ini menggunakan metode *information retrieval* pada teknologi *Text Mining*. Tahapan ini mencari topik-topik yang dibahas pada setiap dokumen yang dimasukkan sebagai sumber pengetahuan sistem pakar. Topik yang dibahas akan berbentuk frase kata benda, dan ada kecenderungan frekuensi kemunculan kata/frase tersebut akan cukup tinggi, tetapi kata/frase dengan frekuensi kemunculan yang cukup tinggi pada semua dokumen bukan merupakan topik yang dibahas. Untuk itu maka digunakan model *retrieval term weighting (TF, IDF, TF-IDF)* untuk memberikan nilai tingkat relevansi dan derajat kemiripan (*cosine similarity*) pada sebuah frase.



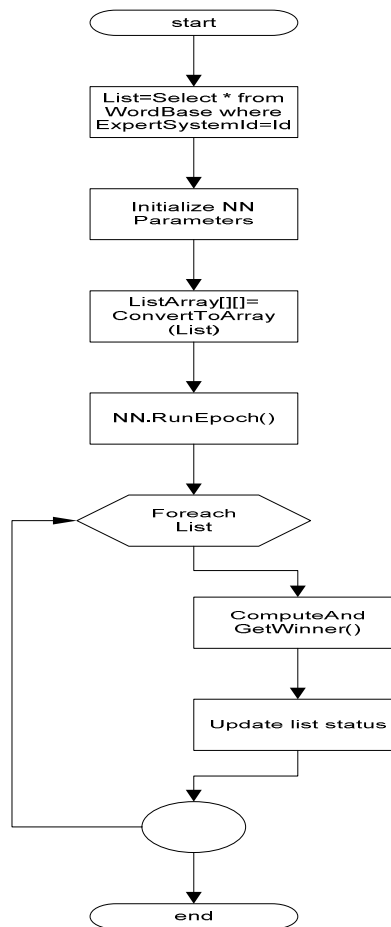
Gambar 8 Flowchart proses topics retrieval.
Topics clustering



Gambar 9. Arsitektur *self organizing maps neural networks*.

Merupakan proses yang melakukan *clustering* terhadap kandidat-kandidat topik pada *TopicBase file*, dimana setiap kandidat topik akan dikelompokkan berdasarkan nilai yang diberikan pada proses sebelumnya. Tujuan pengelompokan adalah untuk memisahkan antara kandidat yang cukup berpotensi dan yang kurang potensial untuk menjadi *domain* sistem pakar. Hasil proses ini akan melakukan *update* dengan memberi status tertentu pada *record* kandidat topik tersebut.

Tahapan ini menggunakan salah satu algoritma pembelajaran *neural network* yaitu *self organizing maps*. *Self organizing maps* digunakan untuk melakukan *clustering* pada setiap frase dimana tidak dibutuhkan data pelatihan. *Input* yang dimasukkan adalah *TF average*, *IDF*, *TF-IDF average*, *cosine similarity average*, dengan empat buah *neuron*. Tujuan melakukan *clustering* adalah untuk mengelompokkan topik yang relevan dan tidak relevan berdasarkan nilai hasil perhitungan tahapan sebelumnya.

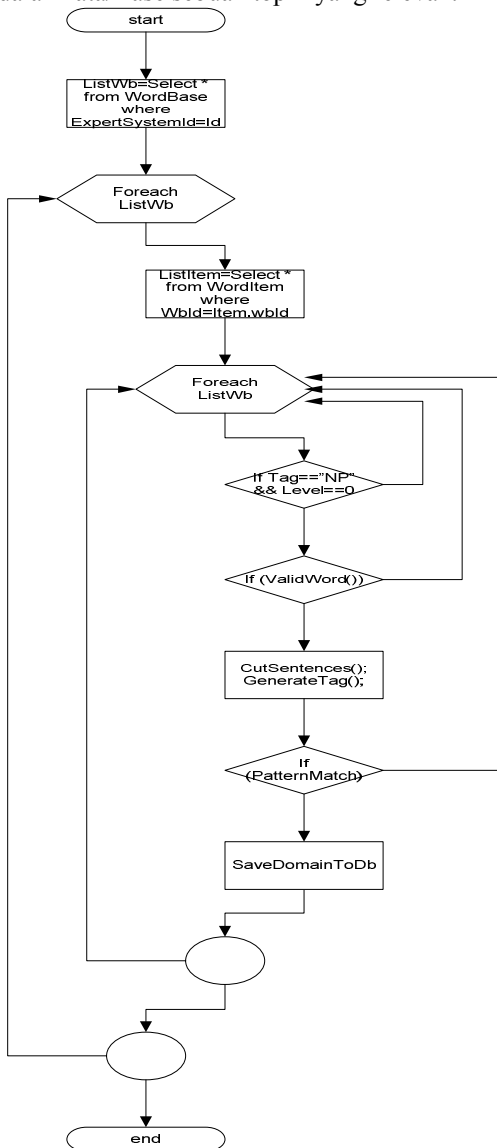


Gambar 10. *Flowchart* proses *topics clustering*.

Domain extraction

Merupakan proses yang melakukan pembentukan ulang kalimat kandidat *domain* yang telah dipilih untuk menjadi *domain* berdasarkan hasil yang disimpan pada *TopicBase file* pada proses sebelumnya. Hasil pembentukan kalimat harus mempunyai pola *tagging* tertentu yang mengandung makna dan pengetahuan tentang *domain* tersebut. Kandidat topic yang memenuhi syarat pola kalimat akan disimpan pada *DomainBase file*. Kalimat pada *DomainBase file* merupakan kalimat yang akan diinformasikan kepada pengguna sistem pakar tentang masalah yang dihadapinya berdasarkan indikasi-indikasi yang disampaikan pengguna pada sistem pakar lewat interaksi tanya-jawab antara pengguna dan sistem pakar.

Tahapan ini menggunakan metode *information extraction* untuk melakukan ekstraksi kalimat. Tahapan ini melakukan ekstraksi informasi pada setiap kalimat yang mengandung kata/frase/topik yang relevan dalam teks. Ekstraksi dilakukan dengan mendeteksi pola *tagging* kalimat topik tersebut. Pola *tagging* kalimat yang digunakan adalah NP_VBZ_!TO!NP!WHPN!VP! dan harus mengandung tag RB atau kata *cannot*, *not* atau kata lain yang ditentukan, dimana NP adalah kata/frase sebuah topik yang relevan.

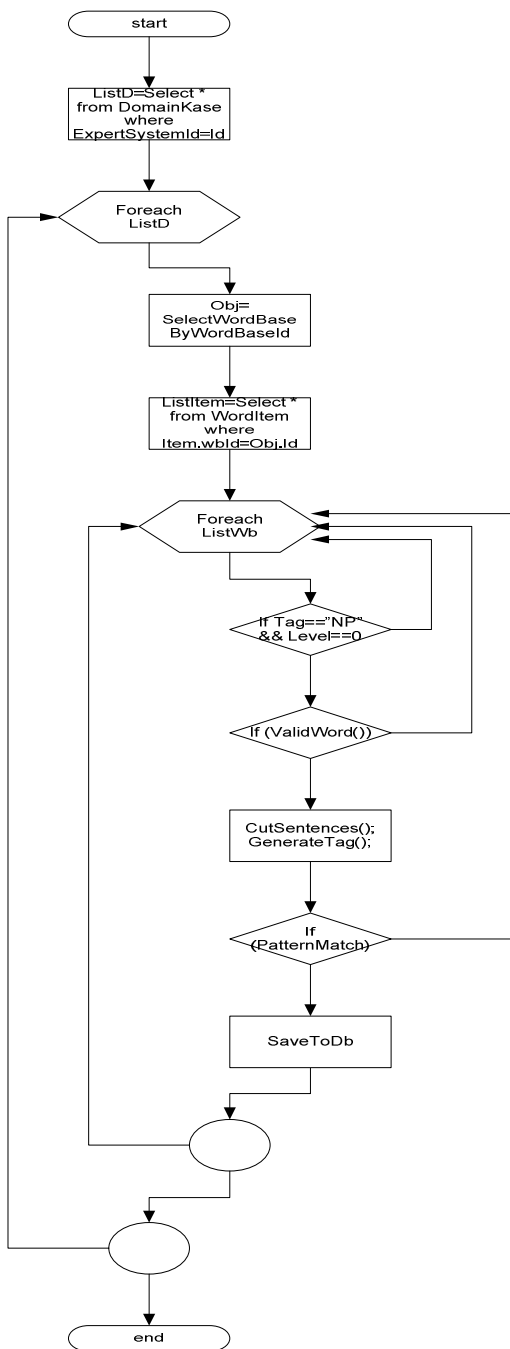


Gambar 11. Flowchart domain extraction.

Item extraction

Merupakan proses yang melakukan pencarian terhadap indikasi-indikasi dari *domain* permasalahan pada *DomainBase file*. Pada proses ini dilakukan pembentukan ulang kalimat berdasarkan *domain* tertentu, dimana hasil pembentukan ulang kalimat tersebut mempunyai pengetahuan indikasi pada *domain* tersebut. Pola kalimat yang memenuhi syarat akan disimpan pada *ItemBase file*. Jika sebuah *domain* ternyata tidak mempunyai kalimat atau memenuhi pola atau tidak mempunyai pengetahuan indikasi, maka *domain* tersebut dianggap bukan merupakan *domain* permasalahan dan akan dihapus dari *DomainBase file*.

Tahapan ini sama dengan *domain extraction* tetapi dengan pola *tagging* yang berbeda. Tahapan ini melakukan ekstraksi informasi indikasi sebuah *domain* yang mencerminkan ciri *domain* tersebut pada setiap kalimat yang mengandung kata/frase/topik sebuah *domain*. Pola *tagging* kalimat yang digunakan adalah NP_VBZ.

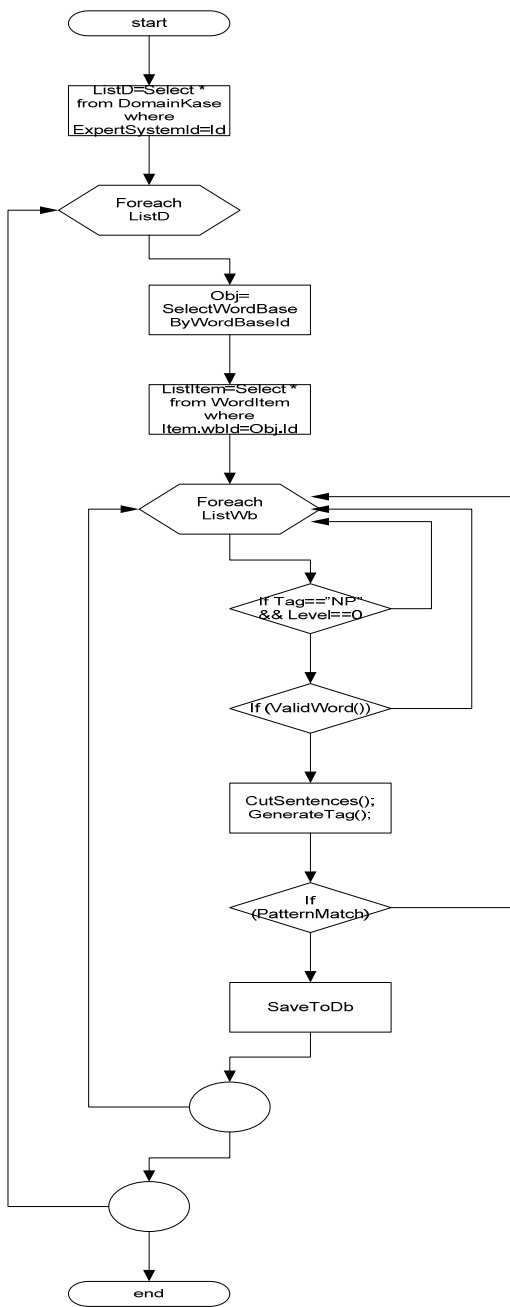


Solution extraction

Merupakan proses yang melakukan pencarian terhadap solusi dari domain permasalahan pada *DomainBase file*. Pada proses ini juga dilakukan pembentukan ulang kalimat berdasarkan domain tertentu, dimana hasil pembentukan ulang kalimat tersebut mempunyai pengetahuan tentang solusi terhadap permasalahan tersebut. Pola kalimat yang mempunyai pengetahuan tentang solusi pada domain tersebut akan disimpan pada *SolutionBase file*. Domain yang tidak mempunyai solusi tetapi mempunyai indikasi, tetap dianggap sebagai domain permasalahan. Hal ini akan diinformasikan pada user bahwa permasalahan yang dihadapi belum mempunyai solusi yang dapat diberikan, karena solusi tersebut belum terdapat pada basis pengetahuan sistem pakar.

Tahapan ini juga sama dengan *domain extraction* tetapi dengan pola *tagging* yang berbeda. Tahapan ini melakukan ekstraksi informasi solusi sebuah *domain* pada setiap kalimat yang mengandung kata/frase/topik sebuah *domain* dan terdapat POS *tagging* kalimat VB.

Gambar 12. Flowchart item extraction.



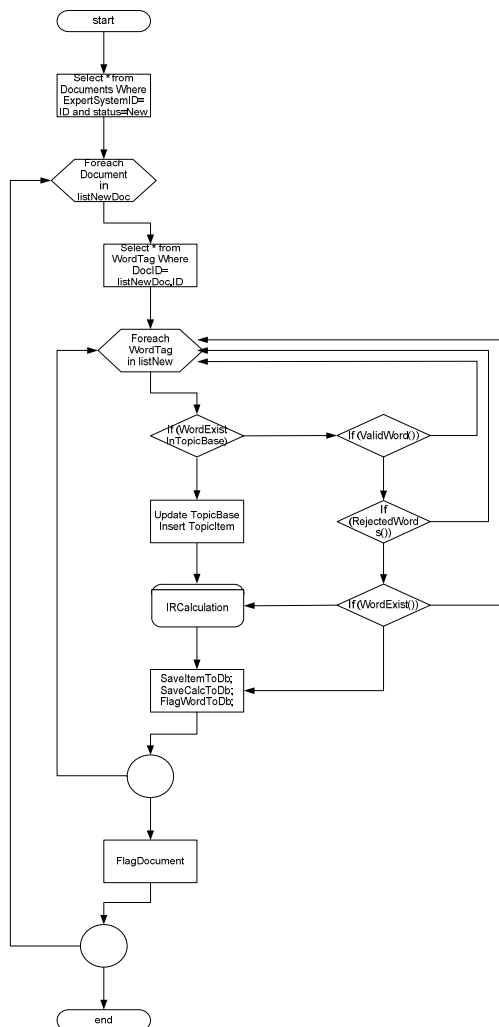
Gambar 13. Flowchart solution extraction.

Knowledge Base Update

Update knowledge dapat berguna untuk menambah isi dari basis pengetahuan. Kesalahan informasi pengetahuan yang diterima pada saat proses akuisisi sebelumnya dapat diperbaiki pada proses *update* ini. Dengan demikian kualitas basis pengetahuan akan bisa ditingkatkan.

Update knowledge dilakukan dengan cara menambahkan dokumen-dokumen sumber pengetahuan pada sistem. Kemudian proses akuisisi dilakukan dari tahap awal sampai tahap akhir, sesuai yang telah dijelaskan sebelumnya. Tetapi tahap pertama, kedua, keempat, kelima, keenam hanya dilakukan pada dokumen-dokumen yang baru dimasukkan tersebut. Hanya proses ketiga yang dilakukan terhadap seluruh data pada dokumen. Proses ketiga ini berfungsi untuk melakukan pengelompokan ulang, untuk mencari calon topik baru atau menghapus topik yang ternyata tidak relevan.

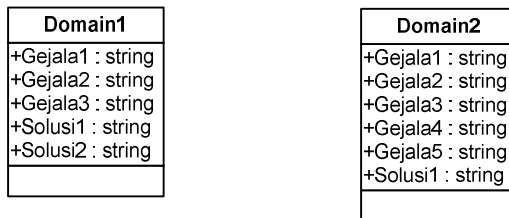
Secara umum, proses *update* basis pengetahuan sebenarnya sama dengan semua tahapan. Tetapi pada tahap kedua, terdapat sedikit perbedaan. Pada proses *update* ini, ditambahkan satu proses untuk memeriksa kata/frase pada *TopicBase file*, jika ternyata topik tersebut sudah muncul, maka seluruh nilai-nilai variabel perhitungan akan ditambahkan, kemudian perubahan tersebut disimpan kembali ke *file*. Setelah perubahan nilai disimpan ke *file*, proses selanjutnya sama dengan proses akuisisi pada tahapan kedua hingga tahap keenam.



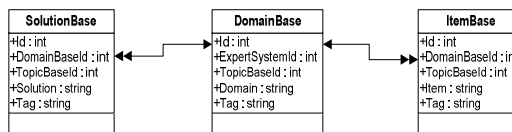
Gambar 14. Flowchart proses knowledge base update.

Representasi Pengetahuan dan Pengembangan Mesin Inferensi

Hasil akuisisi pengetahuan otomatis menghasilkan representasi pengetahuan berorientasi objek. Mekanisme inferensi dilakukan berdasarkan gejala yang terjadi pada sebuah domain. Proses pelacakan dapat terjadi antar objek, pelacakan gejala terhadap sebuah objek diprioritaskan terhadap objek yang memiliki probabilitas yang paling besar. Pemberian bobot probabilitas sebuah domain dilakukan berdasarkan jawaban pengguna terhadap pertanyaan gejala yang diajukan.

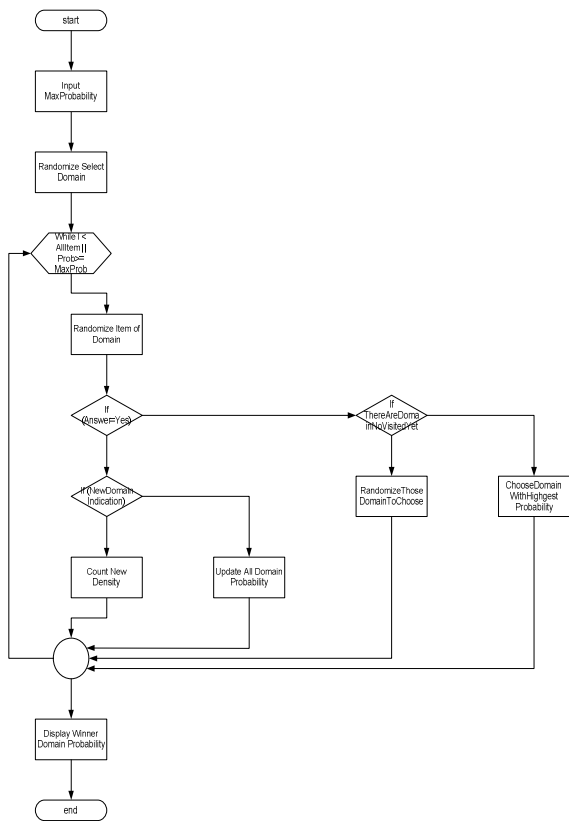


Gambar 15. Representasi pengetahuan berorientasi objek.



Gambar 16. Struktur basis pengetahuan.

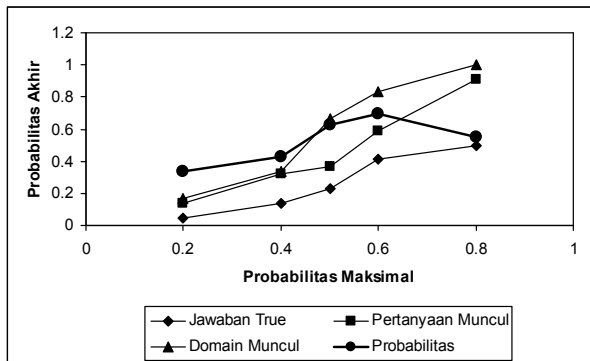
Mekanisme inferensi ini menggunakan penalaran statistik probabilitas dari teori Dempster-Shafer dengan tujuan untuk mengatasi ketidakpastian pada penalaran non-monotis dan untuk mengurangi jalur pelacakan setiap gejala pada semua objek. Selain itu digunakan juga tinjauan studi fabric fault advisory expert system sebagai referensi tambahan untuk mengembangkan mesin inferensi. Mekanisme inferensi dilakukan secara berulang hingga nilai probabilitas akhir sebuah domain telah mencapai nilai probabilitas maksimal yang ditentukan pengguna.



Gambar 17. Flowchart mekanisme inferensi.

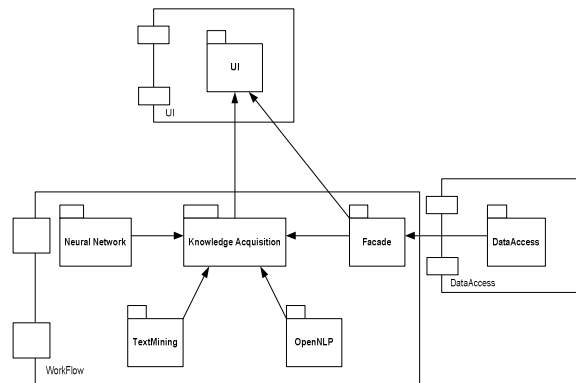
Sesuai dengan teori *dempster-shafer* bahwa nilai probabilitas akan bergerak antara 1 dan 0, nilai bobot probabilitas sebuah domain juga bergerak antara 1 dan 0. Semakin mendekati nilai 1 artinya bobot probabilitas tersebut semakin besar demikian sebaliknya. Nilai bobot probabilitas sebuah domain akan bertambah jika *item* dijawab *true* oleh pengguna. Dalam proses inferensi jika telah terdapat sebuah domain yang bobot probabilitasnya telah mencapai nilai bobot maksimal yang ditentukan maka proses inferensi akan berhenti dan domain tersebut yang diputuskan sebagai masalah yang dihadapi pengguna dengan nilai probabilitas terakhir hasil *input* pengguna.

Berdasarkan hasil percobaan ditemukan bahwa nilai probabilitas maksimal yang harmonis berada antara 0.6 dan 0.8. Berikut adalah grafik hasil percobaan tersebut.

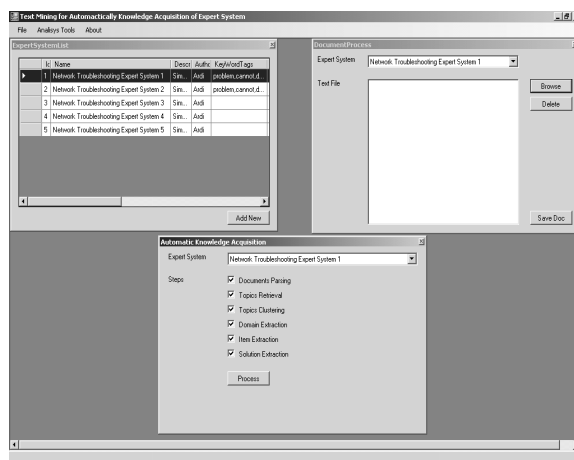


Gambar 18. Grafik nilai probabilitas hasil percobaan inferensi.

Pengembangan Prototype Pengembangan prototype sistem akuisisi pengetahuan otomatis.

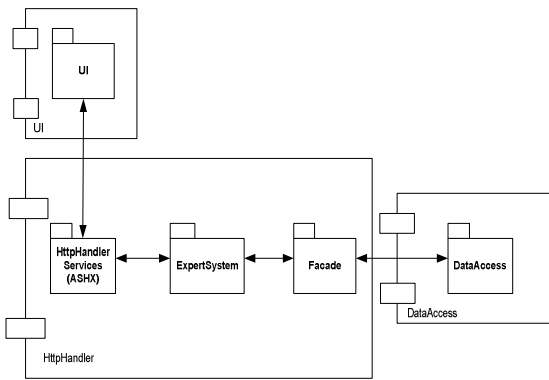


Gambar 19. Package diagram sistem akuisisi pengetahuan otomatis.

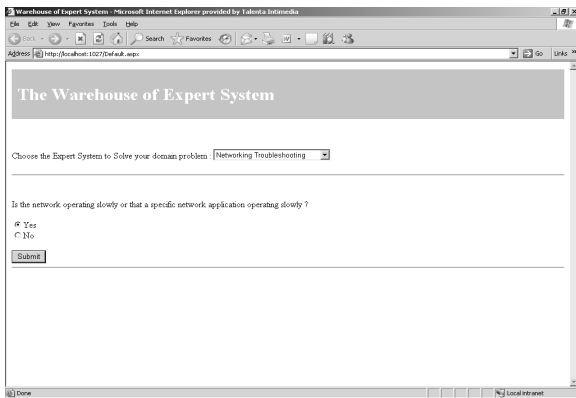


Gambar 20. User Interface sistem akuisisi pengetahuan otomatis.

Pengembangan prototype sistem pakar



Gambar 21. Package diagram sistem pakar.



Gambar 22. User interface sistem pakar.

Evaluasi Tingkat Akurasi Basis Pengetahuan

Evaluasi dilakukan untuk mengukur seberapa tinggi tingkat akurasi basis pengetahuan yang dihasilkan melalui proses akuisisi pengetahuan otomatis. Evaluasi dilakukan dengan mengadaptasikan model perhitungan *F-Measure* pada *information retrieval* dan *information extraction*, kemudian nilai-nilai tersebut dinormalisasi. Tingkat akurasi dievaluasi berdasarkan hasil penilaian manusia secara subjektif.

$$AcEs = \frac{FIR + FIE}{2}$$

$$Ac = \frac{\sum_i AcEs}{N}$$

AcEs = Tingkat akurasi 1 buah sistem pakar jenis tertentu yang dihasilkan.

Ac = Tingkat akurasi sistem pakar secara keseluruhan.

N = Total keseluruhan sistem pakar yang dihasilkan.

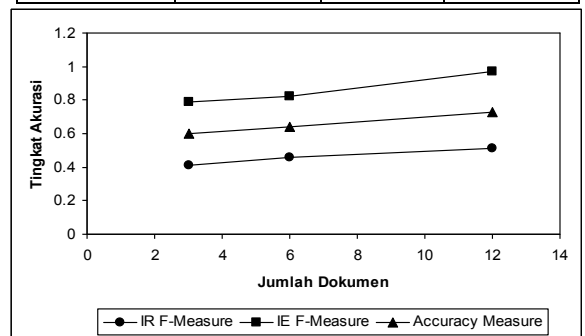
FIE = *F-Measure information retrieval* pada sistem pakar.

FIR = *F-Measure information extraction* pada sistem pakar.

Berikut adalah hasil percobaan terhadap 3 basis pengetahuan yang dihasilkan:

Tabel 1. Hasil evaluasi.

Expert System	FIR	FIE	AcEs
(1). MODEM	0.414	0.79	0.602
(2). Network	0.46	0.823	0.641
(3). Car	0.5136	0.97	0.726
<u>Rata-rata</u>	<u>0.46253</u>	<u>0.861</u>	<u>0.6563</u>



Gambar Grafik evaluasi.

Dari hasil pengujian diatas bahwa nilai akhir (Ac) yang diterima adalah 0.6563, tingkat akurasi basis pengetahuan meningkat jika jumlah dokumen sumber pengetahuan semakin banyak dimasukkan. Pada percobaan ini terlihat bahwa jumlah dokumen sangat mempengaruhi tingkat akurasi basis pengetahuan sistem pakar. Nilai akurasi akhir

yang rendah disebabkan karena jumlah dokumen yang dimasukkan pada setiap percobaan bervariasi dari basis pengetahuan dengan jumlah sedikit sampai ke jumlah dokumen yang banyak.

Semakin banyak dokumen yang dimasukkan juga menyebabkan semakin lamanya proses akuisisi pengetahuan dilakukan. Rata-rata waktu yang dibutuhkan 1 dokumen untuk keseluruhan proses adalah sekitar 30 menit jam sampai 1 jam, tergantung dari usuran dokumen. Proses yang paling lama dilakukan adalah pada proses *parsing* dan *topics retrieval*, kedua proses tersebut memakan waktu kurang lebih 60% dari keseluruhan proses.

KESIMPULAN DAN SARAN

Kesimpulan

Penelitian telah mengembangkan sebuah metode baru untuk akuisisi pengetahuan otomatis pada sistem pakar dari sumber terdokumentasi berupa *text file* yang terdiri dari enam tahapan dan satu proses *update* basis pengetahuan. *Output* setiap tahapan akan digunakan pada tahapan berikutnya sebagai *input*. Pengembangan metode ini, mengkombinasi teknologi *natural language processing*, *information retrieval*, *self organizing maps* dan *information extraction* dan tinjauan studi *knowledge graph* sebagai tambahan referensi pada teknologi *information extraction*. Keenam tahapan proses tersebut, yaitu: *Document parsing*, *Topics retrieval*, *Topics clustering*, *Domain extraction*, *Item extraction*, *Solution extraction*, ditambah satu proses *update* pengetahuan.

Hasil akuisisi pengetahuan otomatis menghasilkan representasi pengetahuan berorientasi objek. Mekanisme inferensi ini menggunakan penalaran statistik probabilitas dari teori *dempster-shafer* dengan tujuan untuk mengatasi ketidakpastian pada penalaran non-monotonis dan untuk mengurangi jalur pelacakan setiap gejala pada semua objek. Mekanisme inferensi dilakukan secara berulang hingga nilai probabilitas akhir sebuah *domain* telah mencapai nilai probabilitas maksimal yang ditentukan pengguna. Berdasarkan hasil percobaan

ditemukan bahwa nilai probabilitas maksimal yang harmonis berada antara 0.6 dan 0.8.

Prototype dikembangkan dengan pendekatan rekayasa perangkat lunak berorientasi objek menggunakan UML sebagai *tools* untuk memodelkan dan mendokumentasikan sistem. *Software process* yang digunakan untuk pengembangan *prototype* adalah proses *iterative* menggunakan model *agile* metode *Extreme Programming (XP)*. Tahapan *coding* mendominasi keseluruhan pengembangan *prototype*.

Evaluasi dilakukan untuk seberapa tinggi tingkat akurasi basis pengetahuan yang dihasilkan melalui proses akuisisi pengetahuan otomatis. Dari hasil pengujian diketahui bahwa nilai akhir (*Ac*) yang diterima adalah 0.6563, performance sistem meningkat jika semakin banyak dokumen. Pada percobaan ini terlihat bahwa jumlah dokumen sangat mempengaruhi tingkat akurasi *knowledge base* sistem pakar. Nilai akurasi akhir yang rendah disebabkan karena jumlah dokumen yang dimasukkan pada setiap percobaan bervariasi dari *knowledge base* dengan jumlah sedikit sampai ke jumlah dokumen yang banyak.

Saran

- Untuk penelitian lebih lanjut, pada proses *domain extraction*, *item extraction* dan *solution extraction* dapat dikembangkan menggunakan *machine learning*, sehingga sistem dapat belajar sendiri menentukan pola kalimat yang akan diekstrak.
- Pada penelitian ini, metode akuisisi pengetahuan otomatis hanya bisa melakukan pembacaan dari *text file*, untuk penelitian lebih lanjut, dapat dikembangkan agar akuisisi dapat dilakukan dari file HTML. Hal tersebut dapat dilakukan dengan menggunakan pengenalan pola HTML, *regular expression* dan *machine learning information extraction*.
- Pada penelitian ini, sumber pengetahuan yang dimasukkan harus dicari terlebih dahulu secara manual. Untuk penelitian lebih lanjut dapat dikembangkan agar sistem dapat mencari sendiri dokumen tersebut melalui *API Search Engine* yang

- tersedia. Dengan demikian *knowledge engineer* tinggal memasukkan *keyword* tentang basis pengetahuan sistem pakar yang akan dibangun. Hal ini dapat dilakukan dengan menggunakan metode *information retrieval* pada penelitian ini.
- d. Untuk penelitian lebih lanjut, dapat dikembangkan *tools* untuk proses *parsing* dalam bahasa Indonesia, sehingga dapat dibangun basis pengetahuan sistem pakar dalam bahasa Indonesia.
- e. Pada penelitian ini proses *clustering* dilakukan dengan *neural network self organizing maps* yang tipe pembelajarannya adalah *unsupervised learning*. Untuk penelitian ke depan dapat dilakukan percobaan dengan menggunakan pembelajaran *supervised learning*. Untuk pembelajaran *supervised learning* dapat dicoba menggunakan algoritma pembelajaran *backpropagation* atau tipe jaringan ANFIS (*Adaptive Neuro-Fuzzy Inferences System*). Hal ini untuk membandingkan tingkat akurasi basis pengetahuan yang dihasilkan, dan mengetahui algoritma pembelajaran mana yang lebih baik.
- f. Untuk penelitian lebih lanjut, proses akuisisi yang sangat lambat harus dipercepat. Salah satu hal yang dapat dilakukan untuk meningkatkan *performance* sistem adalah melakukan *refactoring* terhadap kode program pada *prototype* untuk mempercepat proses.

DAFTAR PUSTAKA

- [AES01] Marimin, **Teori dan Aplikasi Sistem Pakar dalam Teknologi Manajerial**. IPB Press, Bogor, 2007.
- [AES02] Turban, Efraim, et. Al, **Decision Support System and Intelligent System - 7th ed.** Pearson Education, New Jersey 2005.
- [ANL01] Jurafsky, Daniel & Martin, James H, **Speech and Language Processing**, Prentice Hall, New Jersey, 1999.
- [ANN01] Dewi, Sri Kusuma, **Artificial Intelligence, Teknik dan Aplikasinya**. Graha Ilmu, Yogyakarta, 2003.
- [ANN02] Fausset, Laurence, **Fundamental of Neural Networks (Architecture, Algorithms, and Applications)**. Prentice-Hall, New Jersey 1994.
- [ANN03] Kristanto, Andri, **Jaringan Syaraf Tiruan, Konsep Dasar, Algoritma dan Aplikasi**. Gaya Media, Yogyakarta 2004.
- [ANN04] Zurada, M Jacet, **Introduction to Artificial Neural Systems**, West Publishing Company, St. Paul, 1992.
- [ASE01] Fowler, Martin, **UML Distilled - 3rd ed.** Pearson Education, Boston, 2003.
- [ASE02] Freeman, Eric, et. Al. **Head First Design Patterns**. O Reilly, Sebastopol, 2004.
- [ASE03] Pressman, Roger S, **Software Engineering: A Practitioner's Approach**. McGraw-Hill, New York 2005.
- [ASE04] Sommerville, Ian, **Software Engineering - 7th ed.** Addison-Wesley, Massachusetts 2006.
- [BNL01] Zhang, Lei, 2002, **Knowledge Graph Theory and Structural Parsing**, PhD Thesis, University of Twente, Netherlands,
<http://www.tup.utwente.nl/catalogue/book/index.jsp?isbn=9036518350> (diakses 10 Maret 2008)
- [CES01] Nomusa et. Al, September 2007. **Integrating an Object Oriented Approach and Rule-based Reasoning in the Design of a Fabric Fault Advisory Expert System**, Fibres and Textile in Eastern Europe,
<http://proquest.umi.com/pqdweb?did=978659391&sid=3&Fmt=2&clientId=73831&RQT=309&VName=PQD> (diakses 10 Maret 2008)
- [DNL01] Northedge, Richard, **Statistical parsing of English Sentences**,
<http://www.codeproject.com/KB/recipes/englishparsing.aspx>, (diakses 2 Maret 2008)
- [DNL02] Northedge, Richard, **Maximum Entropy Modeling Using SharpEntropy**,
<http://www.codeproject.com/KB/cs/sharpenropy.aspx>, (diakses 2 Maret 2008)
- [ENN01] Kirillov, Andrew, **Neural Networks on C#**, 2008,
<http://www.codeproject.com/>

- [KB/recipes/aforge_neuro.aspx](#) (diakses 7 July 2008)
- [EIR01] Mayfield, James, **Information Retrieval**, 2008, http://www.clsp.jhu.edu/ws2000/presentations/preliminary/jim_mayfield/homewood_ir_tutorial.ppt (diakses 13 Maret 2008)
- [EIR02] Radev, Dragomir R., **Information Retrieval**, 2008, <http://tangra.si.umich.edu/~radev/650/notes/1.ppt> (diakses 3 Juli 2008)
- [EIR03] Mooney, J Raymond, **Mining Complex Types of Data: Information Retrieval**, 2008, <http://www.cs.utexas.edu/users/mooney/ir-course> (diakses 10 Maret 2008).
- [EIE01] Cowie, Jim & Wilks, Yorick, **Information Extraction**, 2008, <http://www.dcs.shef.ac.uk/~yorick/papers/infoext.pdf> (diakses 14 Maret 2008)
- [EIE02] Even, Yair & Zohar, **Introduction to Text Mining**, 2008, <http://algdocs.ncsa.uiuc.edu/PR-20021116-2.ppt> (diakses 13 Maret 2008)
- [EIE03] Grishman, Ralph, **Information Extraction: Techniques and Challenges**, 2008, http://web.njit.edu/~q123/teaching/cis634FTF/notes/Information_extraction_paper.pdf (diakses 14 Maret 2008)
- [EIE04] McCallum, Andrew & Cohen, William, **Information Extraction from the World Wide Web**, 2008, <http://www.cs.cmu.edu/~wcohen/nips-ie-tutorial.ppt> (diakses 13 Maret 2008)
- [EIE05] Mihalcea, Rada & Somai, Andras, **Introduction to Information Extraction**, 2008, <http://lit.csci.unt.edu/~classes/CSC5200/Lectures/IntroInformationExtraction.ppt> (diakses 13 Maret 2008)