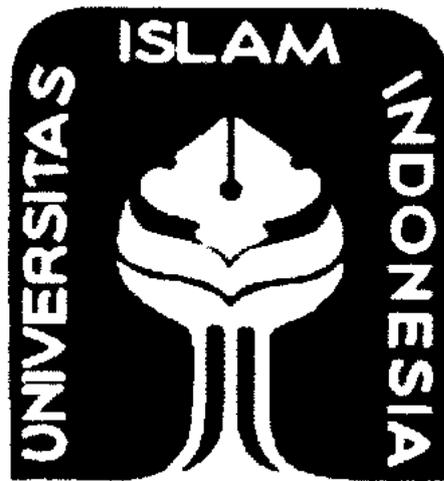


**APLIKASI REPRESENTASI
BASIS PENGETAHUAN BERBASIS ATURAN
TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Oleh :

Nama : Hendra Yuniarto Tri Sarjana

NIM : 06 523 038

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2010

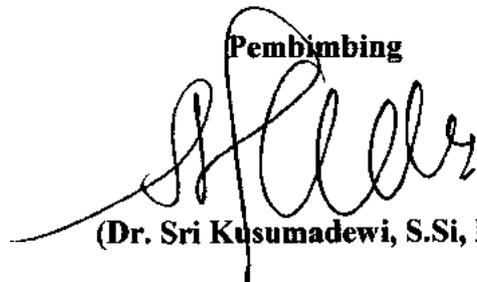
LEMBAR PENGESAHAN PEMBIMBING

**APLIKASI REPRESENTASI
BASIS PENGETAHUAN BERBASIS ATURAN**



Yogyakarta, 29 Desember 2010

Pembimbing


(Dr. Sri Kusumadewi, S.Si, MT) XVI/55

LEMBAR PENGESAHAN PENGUJI

APLIKASI REPRESENTSI BASIS PENGETAHUAN BERBASIS ATURAN

TUGAS AKHIR

oleh:

Nama : Hendra Yuniyanto Tri Sarjana

No Mahasiswa : 06 523 038

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 29 Desember 2010

Tim Penguji

DR., Sri Kusumadewi, S.Si., MT.

Ketua

Zainudin Zukhri S.T., M.I.T.

Anggota I

Syarif Hidayat, S.Kom., M.I.T.

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika



(Yudi Prayudi, S.Si., M.Kom.)

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini

Nama : Hendra Yunianto Tri Sarjana

NIM : 06 523 038

Tugas Akhir dengan judul :

APLIKASI REPRESENTASI BASIS PENGETAHUAN BERBASIS ATURAN

Dengan ini saya menyatakan dengan sesungguhnya bahwa dalam tugas akhir ini tidak terdapat keseluruhan tulisan atau karya yang saya ambil dengan menyalin, meniru dalam bentuk rangkaian kalimat atau simbol atau algoritma atau program yang menunjukkan gagasan atau pendapat atau pemikiran orang lain, yang saya aku seolah-olah sebagai tulisan atau karya saya sendiri.

Apabila saya melakukan hal tersebut diatas, baik sengaja atau tidak, dengan ini saya menyatakan menarik tugas akhir yang saya ajukan sebagai hasil karya saya sendiri. Bila dikemudian hari terbukti bahwa saya melakukan tindakan diatas, gelar dan ijazah yang telah diberikan oleh Universitas Islam Indonesia batal saya terima.

Yogyakarta, 29 Desember 2010

Yang Membuat Pernyataan

(Hendra Yunianto Tri Sarjana)

HALAMAN PERSEMBAHAN

Rasa syukur kehadiran Allah SWT atas karuniaNya dan KeridhoanNya.

**Segala puji bagi Nabi Muhammad SAW,
Yang tiada pernah menyerah dalam menyelamatkan dan membimbing
umatnya ke jalan yang benar dan menuju keselamatan.**

**Alm. Ayah dan Ibu yang saya sayangi,, tiada kata yang bisa terucapkan untuk
berterima kasih atas semuanya**

**Kakakku, terima kasih telah memberi semangat hingga dapat terselesainya
Tugas Akhirku.**

Harapanku dan cita-cita yang selalu menyertai perjalananku.

Teman-temanku semua yang tidak bisa aku sebutkan satu-satu.

Teman-teman yang nantinya akan mengembangkan Tugas Akhir saya.

MOTTO

“ Sesungguhnya sesudah kesulitan itu ada kemudahan ; Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain ”.

(Q.S. Alam Nasyrah ayat 6 dan 7)

“ Dunia hanya berjalan tiga hari, yaitu : Kemarin, yang kita tidak berpengharapan apa-apa lagi darinya. Hari ini, yang harus kita peroleh kebaikan dan kesuksesannya. Dan besok hari, yang tidak kita ketahui apakah kita termasuk yang masih hidup atau yang tergolong sudah meninggal “.

(Al Hasan Al Bashri)

certainly do not need reasons

Dualisme Cartesian

الجامعة الإسلامية
الاستاذة الاءاءاءاء

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Alhamdulillah rabbil 'alamin. Segala puji bagi Allah SWT yang telah memberikan kesempatan bagi penulis untuk menyelesaikan Laporan Tugas Akhir ini. Sesungguhnya hanya atas izin dan kehendak-Nya penulis dapat menyelesaikan tugas akhir ini.

Tugas akhir ini merupakan syarat wajib di jurusan Teknik Informatika Universitas Islam Indonesia untuk memperoleh gelar sarjana. Untuk itu pada kesempatan baik ini, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Ir. Gumbolo HS., M.Sc., selaku dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi S.Si., M.Kom., selaku Ketua Jurusan Teknik Informatika.
3. Dr. Sri Kusumadewi, S.Si., MT., selaku dosen pembimbing tunggal, terima kasih atas masukan dan bimbingan yang telah diberikan.
4. Alm.Bapak, Ibu di rumah dan kakakku serta saudara-saudaraku, atas do'a, kasih sayang dan semangat yang tiada hentinya, serta kesabaran yang luar biasa.
5. Keluarga Laboratorium Komputasi dan Sistem Cerdas (KSC), atas dukungan dan kerja samanya. Bowo (pak bow / wong kentir), Arif (ndut), liza (ndut 2), kisti (kizzy), henry, robby tahun perjuangan yang mantap. Ardhy, ifa, mendem (yudha), alfian, andhika, yogi. Serta teman-teman Laboratorium Informatika Terpadu
6. Rekan-rekan Teknik Informatika UII angkatan 2006, atas bantuan, dan motivasi sejak masuk perkuliahan hingga selesainya Tugas Akhir ini.
7. Saudara Arif (Ndut / Surip) dan Yudha karena telah membagi ilmu tentang sebuah sistem. Teman seperjuangan TA Hery Ning Subangkit.

8. Rekan-rekan Kos CV Usaha Mulya, terutama Eyang putri yang telah memberikan kamarnya untuk ditempati selama di jogja
9. Keluarga KKN BT-26 Jati Imogiri Sriharjo Bantul, Pak dukuh, bu dukuh, dek nita, Abi (pak ketua), Virdie (bu kordes), Dewi (inem), Dynar (Njenong) , Yaya (dokter G atau dokter gila), Guruh (ustad ucup massur), Priyo (adek kecil atau anak lanang), mas narto (Dosen Pembimbing Lapangan), teman-teman KKN Sriharjo Imogiri Bantul. Yang telah memberikan nuansa keluarga yang baru.
10. Teman-teman (Mbambungan IPA 2), SMA Negeri 1 Madiun 2005.
11. Seluruh pihak yang telah setia mendampingi, memberi semangat, motivasi, dan bantuan doa kepada penulis.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna. Untuk itu, penulis mengharapkan kritik dan saran yang bersifat membangun agar dapat berguna di kemudian hari. Penulis berharap semoga Tugas Akhir ini bermanfaat bagi semua orang, dan diri penulis sendiri. Amin.

Wassalamu 'alaikum Wr. Wb.

Yogyakarta, 29 Desember 2010

الإمامة الإسلامية
Penulis

ABSTRAKSI

Metode *Rule-Based Reasoning* adalah salah satu metode dalam sistem pendukung keputusan yang didalamnya memiliki basis pengetahuan. Dalam metode ini penyelesaian masalah berdasarkan pendekatan kecerdasan buatan menggunakan teknik penyelesaian masalah berdasarkan pada aturan-aturan yang terdapat pada basis pengetahuan.

Penelitian yang dilakukan yaitu membangun suatu aplikasi representasi basis pengetahuan yang menggunakan metode *Rule-Based Reasoning*. Implementasi sistem dibangun dengan menggunakan bahasa pemrograman php dan menggunakan MySQL sebagai basis data untuk menyimpan pengetahuan sebelumnya.

Hasil akhir dari penelitian Tugas Akhir ini adalah suatu aplikasi dengan metode berbasis aturan. Aplikasi ini dapat membantu menyederhanakan pernyataan-pernyataan menjadi aturan-aturan yang lebih sederhana yang berguna dalam pengambilan keputusan tertentu untuk berbagai macam masalah yang telah memiliki basis pengetahuan sebelumnya.

Kata Kunci : *Representation, Rule-Based Reasoning, CNF (Conjunctive normal form)*.

DAFTAR ISI

HALAMAN JUDUL	ii
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PERSEMBAHAN.....	vi
MOTTO.....	vii
KATA PENGANTAR.....	viii
ABSTRAKSI.....	x
TAKARIR.....	xi
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian.....	3
1.6.1 Metode pengumpulan data.....	3
1.6.2 Metode pengembangan sistem.....	3
1.7 Sistematika Penelitian.....	4
BAB II LANDASAN TEORI.....	6
2.1 Basis Pengetahuan (<i>Knowledge Base</i>).....	6
2.2 Motor Inferensi (mesin inferensi).....	6
2.3 Logika.....	7
2.3.1 Logika Proposisi.....	8
2.3.2 Hukum-hukum Logika.....	11

BAB III ANALISIS SISTEM	12
3.1 Analisis Masalah.....	12
3.2 Model Basis Pengetahuan.....	13
3.3 Mekanisme Inferensi	14
BAB IV PENGEMBANGAN SISTEM.....	15
4.1 Analisis Kebutuhan Perangkat Lunak	15
4.1.1 Analisis Kebutuhan Input.....	15
4.1.2 Analisis Kebutuhan Output.....	15
4.1.3 Analisis Kebutuhan Proses.....	15
4.1.4 Analisis Kebutuhan Antar muka Perangkat Lunak.....	16
4.1.5 Analisis Kebutuhan Daftar Use Case	16
4.2 Perancangan Perangkat Lunak.....	16
4.2.1 Metode Perancangan	16
4.2.2 Perancangan Use Case Diagram	17
4.2.3 Perancangan Activity Diagram	17
4.2.4 Perancangan Tabel Basis Data	24
4.2.5 Relasi Tabel.....	27
4.2.6 Rancangan Antar muka Perangkat Lunak	29
4.3 Implementasi Perangkat Lunak	35
BAB V PENGUJIAN.....	40
5.1 Pengujian pada Proses Konfigurasi Basis Pengetahuan	40
5.2 Proses Inferensi.....	47
BAB VI PENUTUP	53
6.1 Simpulan.....	53
6.2 Saran	53
DAFTAR PUSTAKA	55

DAFTAR GAMBAR

Gambar 2. 1 Proses Logika	7
Gambar 4.1 Gambar <i>Use Case Diagram</i>	17
Gambar 4.2 Gambar <i>Activity Diagram</i> Keseluruhan Sistem	18
Gambar 4.3 Gambar <i>Activity Diagram</i> Manajemen Kasus.....	19
Gambar 4.4 Gambar <i>Activity Diagram</i> Manajemen variabel.....	20
Gambar 4.5 Gambar <i>Activity Diagram</i> Manajemen Nilai Variabel.....	21
Gambar 4.6 Gambar <i>Activity Diagram</i> Manajemen Basis Pengetahuan	22
Gambar 4.7 Gambar <i>Activity Diagram</i> Daftar Aturan	23
Gambar 4.8 Gambar <i>Activity Diagram</i> Pengujian.....	24
Gambar 4.9 Relasi Tabel Basis Data.....	28
Gambar 4.10 Antarmuka Halaman Utama	29
Gambar 4.11 Antar muka Halaman Kasus.....	30
Gambar 4.12 Antar muka Halaman Variabel.....	31
Gambar 4.13 Antar muka Halaman Nilai Variabel.....	32
Gambar 4.14 Antar muka Halaman Basis Pengetahuan.....	33
Gambar 4.15 Antarmuka Halaman Daftar Aturan	34
Gambar 4.16 Antar muka Halaman Pengujian.....	35
Gambar 4.17 Halaman Utama	36
Gambar 4.18 Halaman Kasus.....	36
Gambar 4.19 Halaman Variabel.....	37
Gambar 4.20 Halaman Nilai Variabel.....	38
Gambar 4.21 Halaman Daftar Aturan	39

BAB I

PENDAHULUAN

1.1 Latar Belakang

Hampir setiap waktu semua orang berhadapan dengan suatu masalah dalam hal pengambilan keputusan. Di zaman sekarang konsep Sistem Pendukung Keputusan telah berkembang dengan pesat. Banyak sekali metode yang telah digunakan untuk membantu manusia dalam mengambil sebuah keputusan yang tepat. Jenis kasus yang terjadi saat ini semakin beranekaragam dan kompleks serta cukup sulit, sehingga dalam mengambil sebuah keputusan yang tepat orang sering kali mengalami kesulitan jika tidak menggunakan metode pengambilan keputusan yang sesuai. Salah satu metode yang cukup mudah dan sering digunakan adalah sistem berbasis aturan.

Sistem berbasis aturan adalah salah satu teknik penyelesaian masalah berdasarkan pada pengetahuan yang berbentuk aturan-aturan. Aturan-aturan yang disimpan dalam basis pengetahuan akan menjadi pedoman dalam mengambil sebuah keputusan dari suatu masalah yang ada. Dalam menerangkan masalah digunakan aturan (*rule*). Aturan tersebut terdiri dari dua bagian yaitu *IF* dan *THEN*, dimana *IF* merupakan kondisi yang mungkin benar atau tidak benar, sedangkan *THEN* adalah kesimpulan atau tindakan yang dilakukan jika kondisi benar.

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah di dalam domain tertentu. Apabila pengetahuan direpresentasikan dengan aturan (*Rule Based Reasoning*), terdapat 2 metode penalaran yang dapat digunakan yaitu penalaran maju (*Forward Chaining*) dan penalaran mundur (*Backward Chaining*) [KUS03].

Beberapa kelebihan yang dimiliki oleh metode berbasis aturan antara lain modifikasi dan perawatan relatif mudah, setiap aturan bersifat *independent* dari aturan yang lainnya. Selain itu terdapat juga kelemahan dari metode ini seperti

pengetahuan yang kompleks membutuhkan aturan yang sangat banyak. Sistem dengan banyak aturan mempunyai keterbatasan dalam proses pencarian.

Salah satu kendala dalam sistem berbasis aturan seperti di atas adalah untuk pengetahuan yang kompleks akan membutuhkan lebih banyak aturan dalam menyelesaikan suatu permasalahan. Untuk mengatasi kendala tersebut dibutuhkan manajemen sistem yang dapat mengelola basis pengetahuan yang sifatnya kompleks serta dapat mentransformasikan aturan yang diberikan oleh *user* ke dalam bentuk klausa yang dapat dipahami oleh computer, sehingga para pengambil keputusan dapat mengambil keputusan menjadi lebih mudah dan lebih baik lagi.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah didapatkan rumusan masalah yaitu bagaimana membangun sebuah aplikasi *representasi* dari basis pengetahuan yang berbasis pada aturan yang bersifat dinamis, sehingga *user* dapat membangun sistem dengan aturan-aturan yang kompleks.

1.3 Batasan Masalah

Dalam melaksanakan suatu penelitian diperlukan adanya batasan masalah agar tidak menyimpang dari tujuan awalnya sehingga tujuan awalnya dapat tercapai. Batasan masalah yang digunakan dalam penelitian ini sebagai berikut :

- a. Operator yang digunakan untuk penalarannya hanya And, Or and Not.
- b. Penalaran yang digunakan dalam sistem ini adalah penalaran berbasis aturan (*Rule Base Reasoning*), dengan menggunakan metode penalaran maju (*Forward Chaining*).
- c. Pernyataan yang dimasukan user hanya 1 frase saja.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah membangun sebuah aplikasi *representasi* dari basis pengetahuan untuk sistem pendukung keputusan dengan metode berbasis aturan.

1.5 Manfaat Penelitian

Dengan penelitian yang di lakukan ini diharapkan dapat memberikan manfaat sebagai berikut :

- a. Membantu user dalam mengambil keputusan yang lebih baik lagi.
- b. Memberikan kemudahan bagi user dalam hal mengolah basis pengetahuan berdasarkan aturan-aturan yang telah di tentukan sebelumnya.

1.6 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian sebagai berikut :

1.6.1 Metode pengumpulan data

Metode pengumpulan data merupakan metode untuk mengumpulkan data-data yang diperlukan dalam penelitian dan penyelesaian tugas akhir. Metode tersebut antara lain studi pustaka yaitu pengumpulan data dengan pembelajaran atau studi, analisis dan dokumentasi literatur, atau sumber catatan lain.

1.6.2 Metode pengembangan sistem

Metode pengembangan sistem disusun berdasarkan hasil dari data yang telah diperoleh, yang meliputi :

1. Analisis masalah

Tahap analisis masalah yang dilakukan dalam penelitian ini yaitu pertama melakukan identifikasi masalah yang diangkat dalam penelitian ini yaitu tentang metode berbasis aturan. Kedua analisa penyebab masalah, masalah yang diangkat dalam penelitian ini timbul dikarenakan untuk basis pengetahuan yang kompleks akan membutuhkan lebih banyak aturan-aturan dalam menyelesaikan masalah maka dibutuhkan sistem yang dapat mengelola basis pengetahuan yang kompleks. Ketiga model basis pengetahuan, model basis pengetahuan yang dipakai untuk menyelesaikan masalah yang ada yaitu penalaran berbasis pada aturan (*Rule Based Reasoning*).

2. Analisis kebutuhan sistem

Tahap analisa kebutuhan sistem yang dilakukan dalam penelitian ini yaitu melakukan analisa terhadap kebutuhan *input*, proses, kebutuhan *output* dan kebutuhan antarmuka.

3. Perancangan Sistem

Tahap perancangan sistem yang dilakukan dalam penelitian ini yaitu mendefinisikan kebutuhan yang menggambarkan arus data dalam sistem yang penggunaannya sangat membantu untuk memahami sistem secara logika, tersruktur dan jelas berupa UML (*Unified Modeling Language*).

4. Implementasi

Tahap implementasi yang dilakukan dalam penelitian ini yaitu penerapan dari konsep yang telah dibuat sebelumnya ke dalam bentuk program dengan menggunakan bahasa pemrograman PHP.

5. Pengujian Sistem

Pada tahap pengujian sistem ini pengujian dilakukan bersamaan dengan perbaikan sistem, sehingga ketika sistem diuji dan menemukan kesalahan(bug) maka perbaikan segera dilakukan. Hal ini berulang sampai sistem ini tidak ditemukan kesalahan lagi dan sesuai dengan yang diharapkan.

1.7 Sistematika Penelitian

Dalam penyusunan laporan tugas akhir ini, dibagi menjadi beberapa bab yaitu sebagai berikut:

Bab I Pendahuluan pada bab ini terdapat enam sub bab yang mengemukakan tentang garis besar masalah dalam penelitian ini mengenai penalaran basis aturan pada basis pengetahuan, yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II

LANDASAN TEORI

2.1 Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah didalam domain tertentu. Ada dua bentuk pendekatan basis pengetahuan yang sangat umum digunakan, yaitu [KUS03]:

a. Penalaran berbasis aturan (*Rule-Based Reasoning*)

Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk IF-THEN. Bentuk ini digunakan untuk menyelesaikan permasalahan secara berurutan. Disamping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) dalam pencapaian solusi.

b. Penalaran berbasis kasus (*Case-Based Reasoning*)

Pada penalaran berbasis kasus, basis pengetahuan akan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian akan dihasilkan suatu solusi untuk keadaan yang terjadi sekarang (fakta yang ada). Bentuk ini digunakan apabila user menginginkan untuk tahu lebih banyak lagi pada kasus-kasus yang hamper sama(mirip). Selain itu bentuk ini juga digunakan apabila sistem telah memiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan.

2.2 Motor Inferensi (mesin inferensi)

Mesin inferensi adalah bagian yang mengandung mekanisme fungsi berfikir dan pola-pola penalaran sistem yang digunakan oleh seorang pakar. Mekanisme ini akan menganalisis suatu masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan terbaik. Mesin inferensi akan memulai pelacakanya dengan mencocokkan kaidah-kaidah dalam pengetahuan dengan fakta-fakta yang ada dalam basis pengetahuan.

Ada dua cara yang dapat dikerjakan dalam melakukan inferensi, yaitu [KUS03]:

a. Forward Chaining(pelacakan maju)

Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.

b. Backward Chaining

Pencocokan fakta aturan atau pernyataan dimulai dari sebelah kanan (THEN dulu). Dengan kata lain, penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenarannya hipotesis tersebut dicari harus dicari fakta-fakta yang ada dalam basis pengetahuan.

2.3 Logika

Logika adalah bentuk representasi pengetahuan yang paling tua. Pada dasarnya proses logika adalah proses membentuk kesimpulan atau menarik inferensi berdasarkan fakta yang telah ada (Gambar 2.1). Input dari proses logika berupa premis atau fakta-fakta yang diakui kebenarannya sehingga dengan melakukan penalaran pada proses logika dapat dibentuk suatu inferensi atau kesimpulan yang benar pula.



Gambar 2.1 Proses Logika

Ada 2 penalaran yang dapat dilakukan untuk mendapatkan konklusi atau kesimpulan :

a. Penalaran Deduktif

Penalaran dimulai dari prinsip umum untuk mendapatkan kesimpulan yang lebih khusus.

b. Penalaran Induktif

Penalaran dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan yang umum. Pada penalaran induktif ini munculnya pernyataan baru dapat mengakibatkan hilangnya kesimpulan yang sudah diperoleh sebelumnya.

2.3.1 Logika Proposisi

Proposisi adalah suatu pernyataan yang dapat bernilai benar (B) atau salah (S). simbol-simbol seperti P dan Q menunjukkan proposisi. Dua atau lebih proposisi dapat digabungkan menggunakan operator logika, operator-operator tersebut sebagai berikut :

a. Operator NOT (\neg)

Operator NOT digunakan untuk memberikan nilai negasi (lawan) dari pernyataan yang telah ada. Tabel 2.1 menunjukkan tabel kebenaran untuk operator NOT.

Tabel 2.1 Tabel Kebenaran Operator NOT

P	NOT (P)
B	S
S	B

Keterangan :

P : proposisi atau pernyataan

B : pernyataan yang bernilai benar

S : pernyataan yang bernilai salah

b. Operator AND (\wedge)

Disebut juga Konjungsi operator and digunakan untuk mengkombinasikan 2 buah proposisi. Hasil yang diperoleh akan bernilai benar jika kedua proposisi bernilai benar, dan akan bernilai salah jika salah satu dari atau kedua proposisi bernilai salah. Tabel 2.2 menunjukkan tabel kebenaran untuk operator AND.

d. Implikasi (\rightarrow)

Implikasi: jika P maka Q akan menghasilkan nilai salah jika P benar dan Q salah, selain itu akan selalu bernilai benar. Tabel 2.4 menunjukkan tabel kebenaran untuk operator implikasi

Tabel 2.4 Tabel Kebenaran Operator Implikasi

P	Q	$P \rightarrow Q$
B	B	B
B	S	S
S	B	B
S	S	B

Keterangan :

P : proposisi atau pernyataan

Q : proposisi atau pernyataan

B : pernyataan yang bernilai benar

S : pernyataan yang bernilai salah

e. Ekuivalen (\Leftrightarrow)

Ekuivalen akan menghasilkan nilai benar jika P dan Q keduanya benar atau keduanya salah. Tabel 2.5 menunjukkan tabel kebenaran untuk operator Ekuivalensi

Tabel 2.5 Tabel Kebenaran Operator Ekuivalensi

P	Q	$P \Leftrightarrow Q$
B	B	B
B	S	S
S	B	S
S	S	B

BAB III

ANALISIS SISTEM

3.1 Analisis Masalah

Salah satu kendala dalam sistem berbasis aturan adalah untuk pengetahuan yang kompleks akan membutuhkan lebih banyak aturan dalam menyelesaikan suatu permasalahan.

Contoh :

1. If A and B and C or D and E and F or G and H then K.
2. If A and B or D and E and F or G then J.
3. If A and C or G then L.
4. If B and C then K.
5. If D or E then G.
6. If L and C and B or H and I or A then C.
7. If T and P and A or B and L and not E or C and F and D or G and I and L and J or H then R.
8. If S or G and H and V or A and C and D then B
9. If B and D or C and E and G or F and J and L or O and P and R and M or N then A.
10. If G and not K and L and A and H and I or J and B and C or D and E and F and M or N then X.
11. If (not A And B) then X.
12. If (A And B) Or (D And E And F) then X.
13. If A And (B Or C) then D.
14. If If A And B And (C Or D) And E And (F Or G) And H then Y.
15. If A And (C Or D) And E Or H then Y.

Dari beberapa contoh pernyataan di atas dapat dilihat bahwa semakin beragam variabel dan operator maka basis pengetahuan semakin beragam juga hal ini menjadikan suatu sistem menjadi semakin kompleks. Hal tersebut menyebabkan manajemen dari suatu sistem yang berbasis pada aturan akan menjadi lebih.

Dari pernyataan-pernyataan diatas dapat disederhanakan lagi menjadi aturan yang lebih kecil tetapi aturan ini tidak mengubah makna sesungguhnya dari pernyataan tersebut sehingga dalam manajemen sistem akan menjadi lebih mudah.

3.2 Model Basis Pengetahuan

Basis pengetahuan berisi pengetahuan-pengetahuan dalam penyelesaian masalah didalam domain tertentu. Salah satu pendekatan yang digunakan dalam basis pengetahuan dengan menggunakan Penalaran berbasis aturan (*Rule-Based Reasoning*). Pada penalaran berbasis aturan, pengetahuan direpresentasikan dengan menggunakan aturan berbentuk IF-THEN. Bentuk ini digunakan untuk menyelesaikan permasalahan secara berurutan. Disamping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang jejak (langkah-langkah) dalam pencapaian solusi.

Contoh :

1. If A and B and C or D and E and F or G and H then K

Dari pernyataan tersebut dapat di ubah menjadi aturan-aturan

- If A and B then K
- If D and E and F then K
- If G and H then K

2. If A and B or D and E and F or G or H and I then L

Dari pernyataan tersebut dapat di ubah menjadi aturan-aturan

- If A and B and D and E and F then L

BAB IV

PENGEMBANGAN SISTEM

4.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak merupakan analisis terhadap semua kebutuhan yang diperlukan dalam lingkup perangkat lunak yang akan dikembangkan. Analisis kebutuhan yang dilakukan terhadap perangkat lunak akan menghasilkan spesifikasi kebutuhan perangkat lunak tersebut.

4.1.1 Analisis Kebutuhan Input

Analisis kebutuhan input perangkat lunak ini sebagai berikut :

1. Data kasus, untuk menentukan nama dari kasus.
2. Data variabel, untuk menentukan variabel-variabel yang berkaitan dengan kasus.
3. Data nilai variabel, untuk menentukan nilai-nilai dari tiap variabel.
4. Data basis pengetahuan, berisi aturan-aturan yang terbentuk dari pernyataan-pernyataan yang dimasukkan oleh pakar.

4.1.2 Analisis Kebutuhan Output

Analisis kebutuhan output perangkat lunak ini sebagai berikut :

1. Informasi kasus yang terdapat dalam basis pengetahuan.
2. Informasi variabel yang berhubungan dengan kasus yang di pilih atau kasus yang akan diuji.
3. Informasi nilai variabel yang menjelaskan nilai dari tiap variabel.
4. Informasi aturan-aturan yang telah terbentuk.

4.1.3 Analisis Kebutuhan Proses

Kebutuhan proses aplikasi ini sebagai berikut :

1. Proses penerapan algoritma CNF dalam sebuah pernyataan sehingga aturan-aturan yang diinginkan terbentuk.

2. Proses pengujian terhadap data input dari user dengan data-data yang terdapat dalam basis pengetahuan

4.1.4 Analisis Kebutuhan Antar muka Perangkat Lunak

Analisis kebutuhan antar muka perangkat lunak merupakan antar muka untuk merepresentasikan hubungan antara perangkat lunak representasi basis pengetahuan berbasis aturan dengan lingkungannya. Kebutuhan Antar muka yang diperlukan sebagai berikut :

1. Antar muka Halaman Utama
2. Antar muka Data Kasus
3. Antar muka Data Variabel
4. Antar muka Data Nilai Variabel
5. Antar muka Informasi Daftar aturan
6. Antar muka Informasi Basis Pengetahuan
7. Antar muka Informasi Hasil pengujian

4.1.5 Analisis Kebutuhan Daftar Use Case

Analisis kebutuhan daftar *Use Case* dalam perangkat lunak ini sebagai berikut :

1. *Use Case* Manajemen Kasus
2. *Use Case* Manajemen Variabel
3. *Use Case* Manajemen Nilai Variabel
4. *Use Case* Basis Pengetahuan
5. *Use Case* Lihat Daftar Aturan
6. *Use Case* Pengujian

4.2 Perancangan Perangkat Lunak

4.2.1 Metode Perancangan

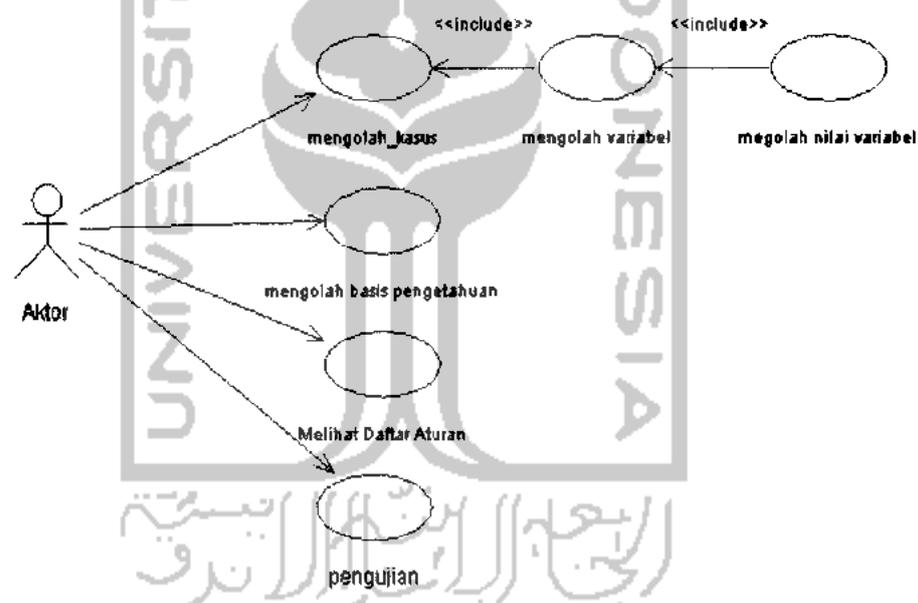
Langkah awal dalam membangun Aplikasi Representasi Basis Pengetahuan Berbasis Aturan ini adalah dengan merancang diagram UML terlebih dahulu. UML adalah sebuah standar untuk visualisasi merancang dan mendokumentasikan

sistem perangkat lunak. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi perangkat lunak.

4.2.2 Perancangan Use Case Diagram

Perancangan *Use Case Diagram* berfungsi untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem dilihat dari perspektif pengguna diluar sistem. Sebuah *Use Case* diagram mempresentasikan interaksi yang terjadi antara aktor dengan proses atau sistem yang dibuat.

Untuk menggambarkan aplikasi yang akan dibangun ini, *Use Case* diagram tersebut dapat dilihat pada gambar 4.1

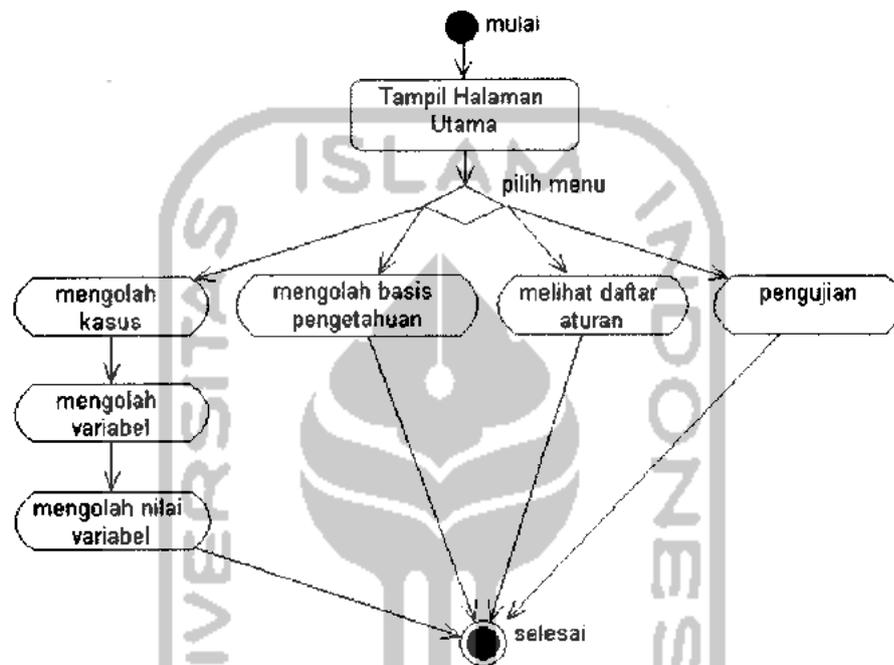


Gambar 4.1 Gambar *Use Case Diagram*

4.2.3 Perancangan Activity Diagram

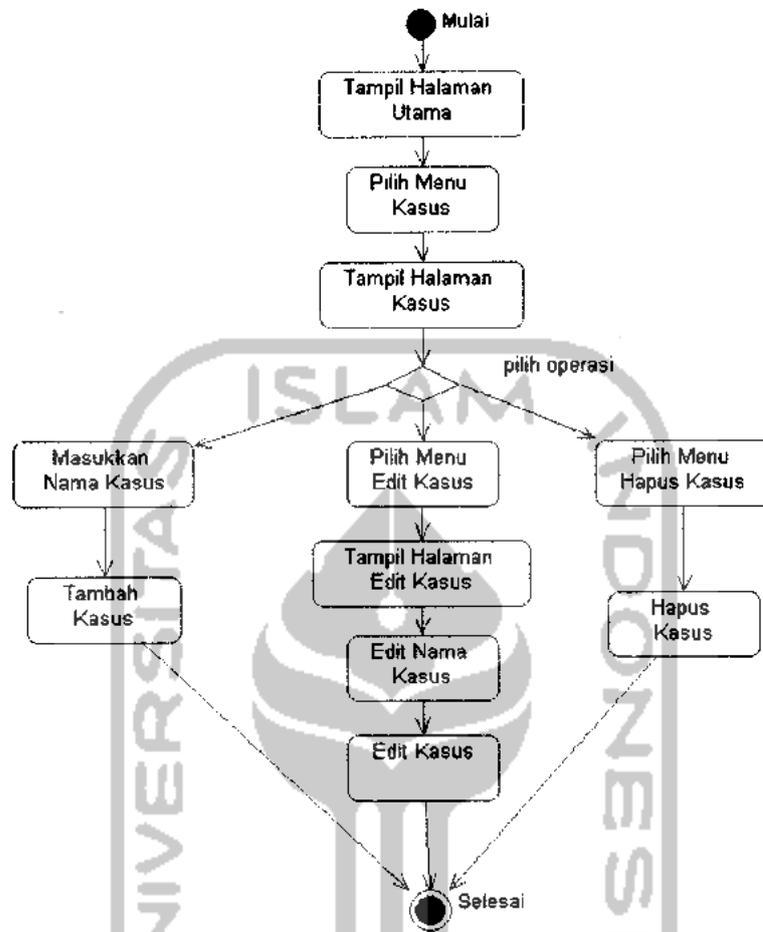
Untuk menggambarkan aktivitas yang terjadi didalam sistem digunakan *activity diagram*. Didalam *activity diagram* akan digambarkan berbagai aliran aktifitas dalam sistem yang akan dibangun, bagaimana aliran aktifitas dalam sistem, bagaimana aliran aktifitas berawal, *decision* yang mungkin terjadi dan bagaimana aktifitas itu berakhir.

Activity diagram umumnya tidak menggambarkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan proses yang terjadi. Gambaran *activity* diagram dalam keseluruhan sistem ini dapat dilihat pada gambar 4.2.



Gambar 4.2 Gambar *Activity Diagram* Keseluruhan Sistem

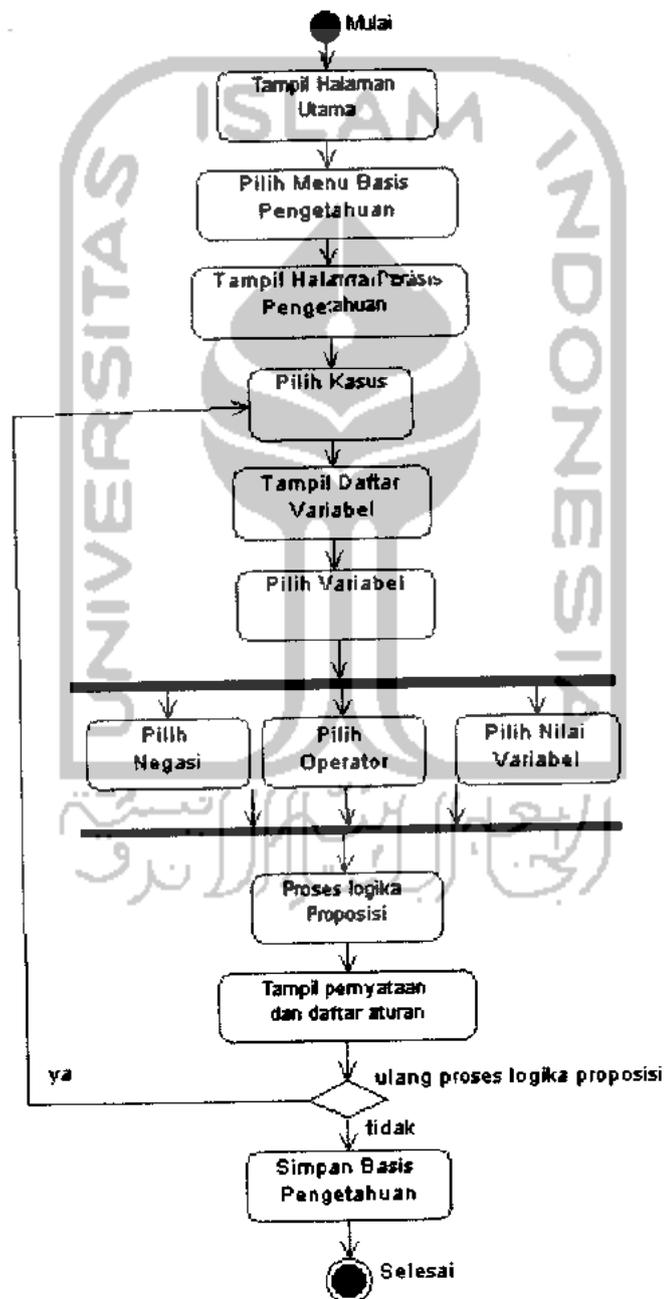
Gambar 4.3 merupakan gambar *activity* diagram untuk Manajemen kasus. Proses pembuatan kasus diawali dari halaman utama yang menuju halaman kasus. Pada proses pembuatan, *user* memasukan kasus yang dibutuhkan. Kasus-kasus tersebut akan ditampilkan ke dalam tabel kasus.



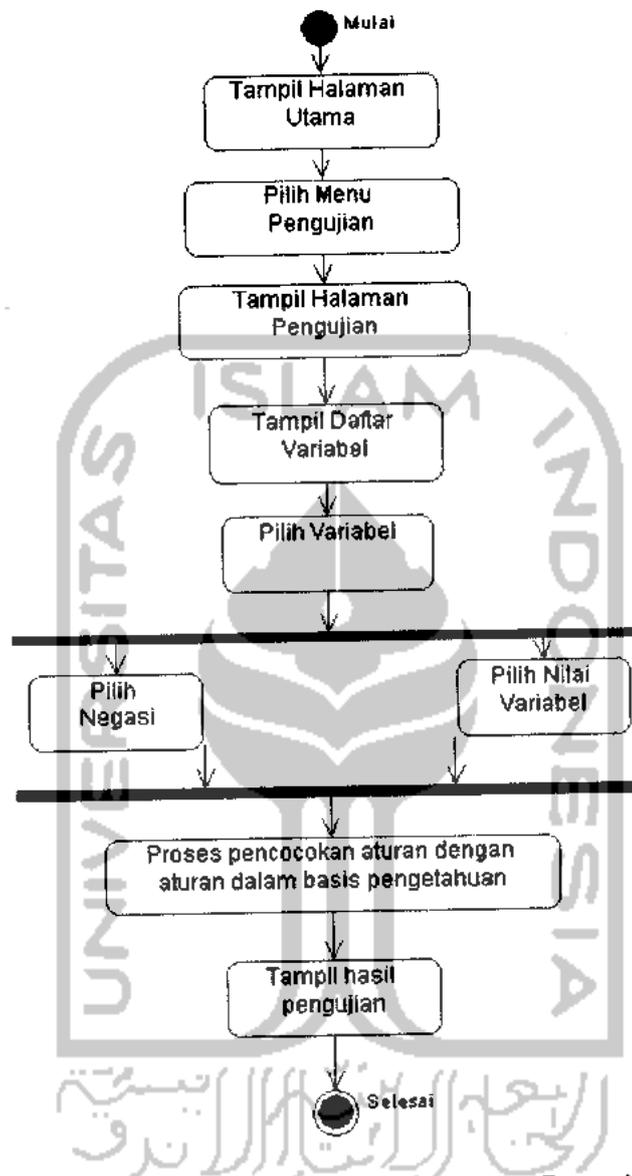
Gambar 4.3 Gambar *Activity Diagram* Manajemen Kasus

Gambar 4.4 merupakan gambar *activity diagram* untuk Manajemen variabel. Proses pembuatan variabel diawali dari halaman utama kemudian pilih kasus yang ingin ditambahkan variabelnya. Pada proses pembuatan, *user* memasukan variabel yang dibutuhkan. Variabel tersebut akan ditampilkan ke dalam tabel variabel.

muncul daftar variabel yang telah dimasukkan pada proses sebelumnya. Untuk membentuk pernyataan yang nantinya akan diproses menjadi aturan user memilih variabel-variabel. Untuk melengkapi proses tersebut user menentukan negasi, nilai variabel dan operator yang berlaku disetiap variabel. Setelah pernyataan lengkap maka akan diproses oleh sistem menjadi aturan-aturan yang lebih sederhana.



Gambar 4.6 Gambar Activity Diagram Manajemen Basis Pengetahuan



Gambar 4.8 Gambar Activity Diagram Pengujian

4.2.4 Perancangan Tabel Basis Data

Basis data memerankan bagian yang penting dalam sistem karena berfungsi sebagai penyimpanan basis pengetahuan. Dalam perencanaan sistem, terdapat 7 tabel basis data yang masing-masing perinciannya dijelaskan sebagai berikut :

g. Tabel Daftar

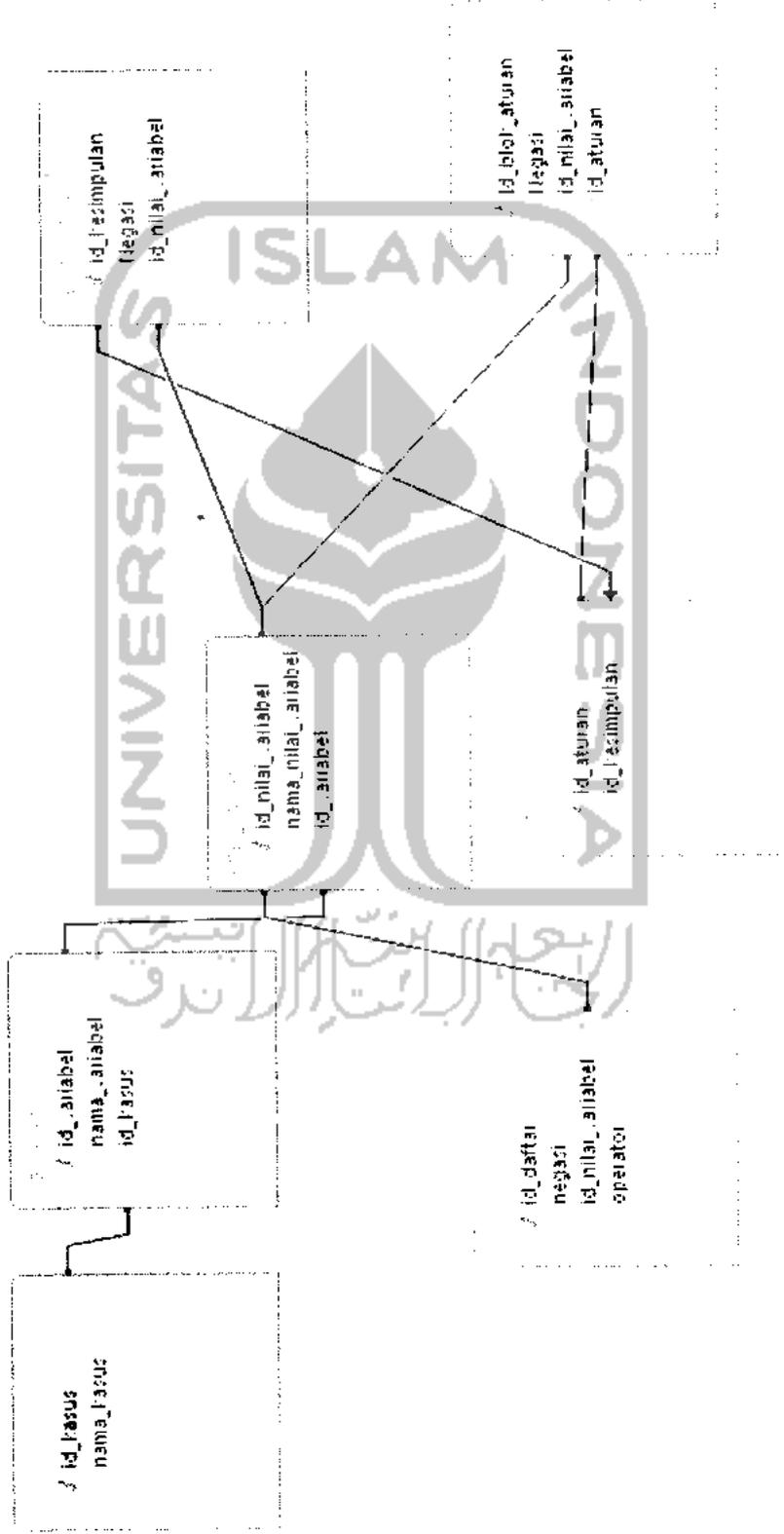
Tabel daftar merupakan tabel yang berisi semua aturan beserta kesimpulan sebelum aturan-aturan tersebut diproses. Struktur tabel daftar ditunjukkan pada tabel 4.7.

Tabel 4.7 Tabel Daftar

No.	Nama Kolom	Tipe Data	Keterangan
1	id_daftar	Int(6)	Primary Key
2	Negasi	Enum('0','1')	
3	id_nilai_variabel	Int(6)	Foreign Key
4	Operator	text	

4.2.5 Relasi Tabel

Dalam pembuatan sistem, relasi antar tabel bertujuan untuk menjelaskan hubungan antar tabel disertai dengan bentuk hubungannya. Relasi antar tabel yang digunakan untuk pembuatan sistem dapat dilihat pada Gambar 4.9.



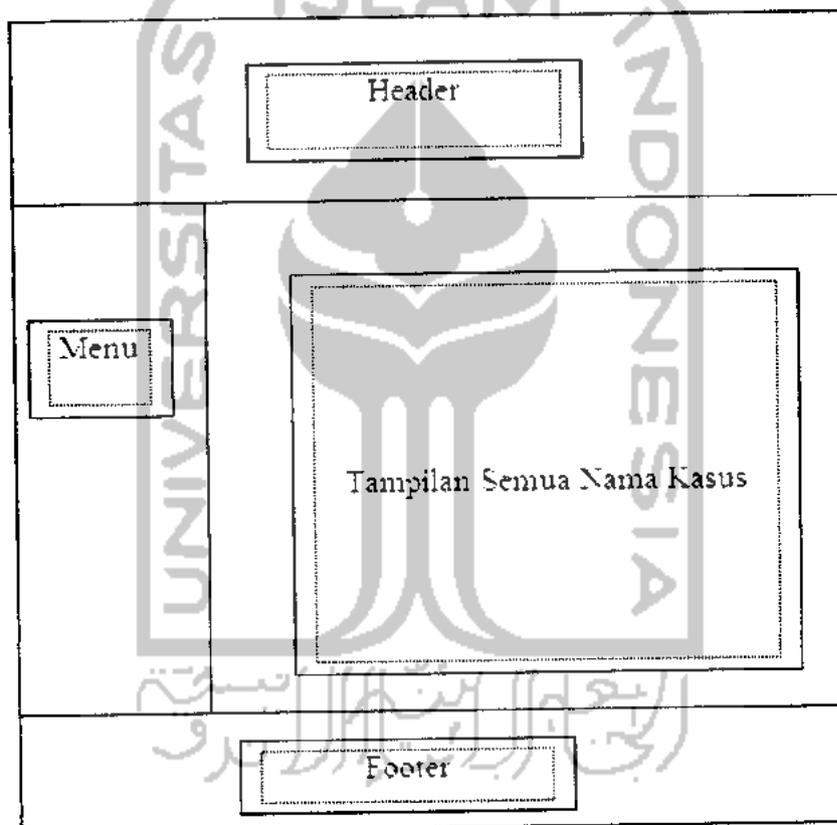
Gambar 4.9 Relasi Tabel Basis Data

4.2.6 Rancangan Antarmuka Perangkat Lunak

Perancangan antarmuka merupakan gambaran umum tampilan (*interface*) perangkat lunak yang akan dibangun, sehingga mempermudah dalam membangun perangkat lunak.

a. Rancangan Antarmuka halaman utama

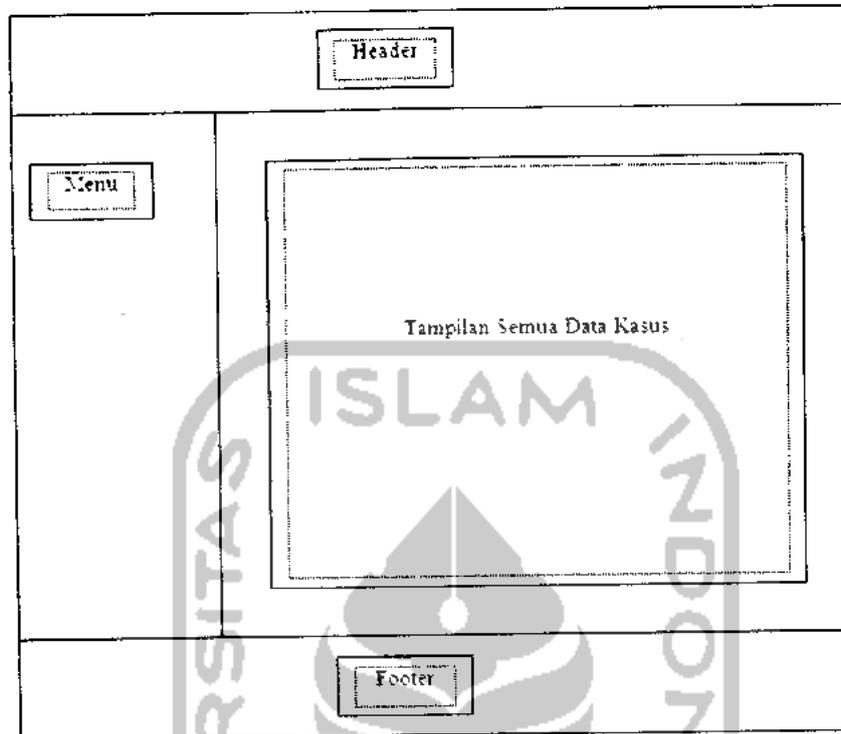
Rancangan antarmuka ini merupakan halaman utama ketika aplikasi ini diakses. (Gambar 4.10).



Gambar 4.10 Antarmuka Halaman Utama

b. Rancangan Antarmuka Halaman Kasus

Rancangan antarmuka Halaman kasus ini adalah halaman kasus Aplikasi Representasi Basis Pengetahuan Berbasis Aturan. Halaman ini merupakan proses untuk melakukan input data kasus serta data kasus apa saja yang telah ada dalam basis pengetahuan. (Gambar 4.11).



Gambar 4.11 Antar muka Halaman Kasus

c. Rancangan Antarmuka Halaman Variabel

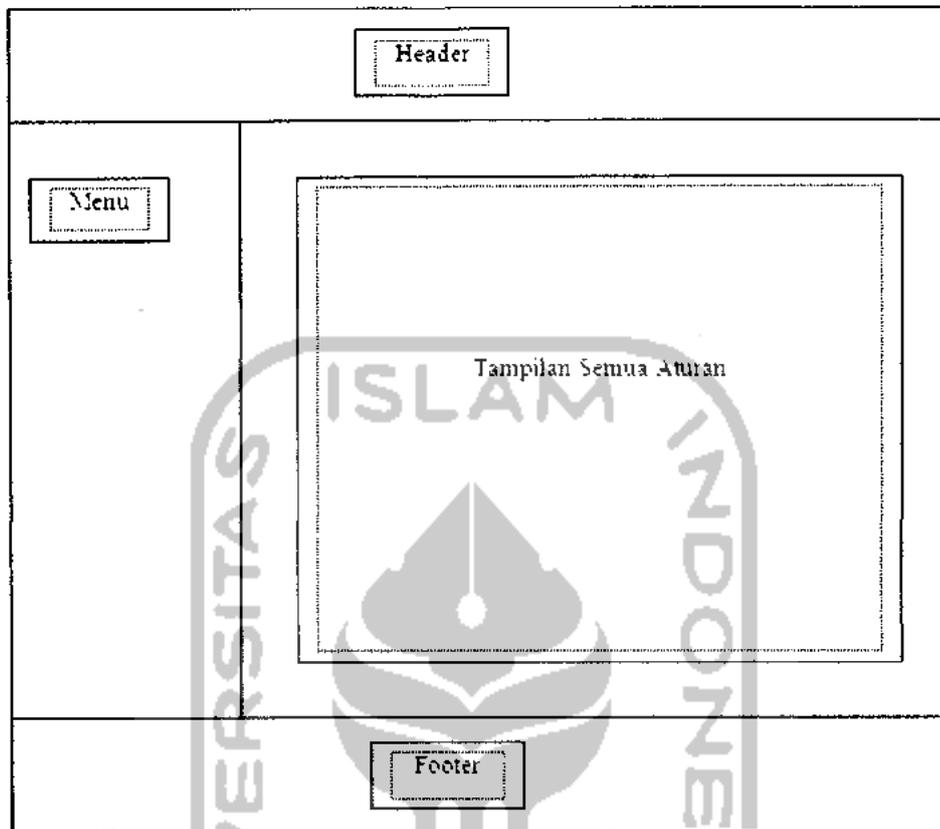
Rancangan antarmuka menu variabel ini adalah halaman variabel Aplikasi Representasi Basis Pengetahuan Berbasis Aturan. Halaman ini merupakan proses untuk melakukan input data variabel serta data variabel apa saja yang telah ada dalam basis pengetahuan. (Gambar 4.12).

The image shows a web application interface for managing variable values. It features a header section at the top, a menu on the left side, and a footer at the bottom. The main content area includes two input fields labeled 'Nama Variabel' and 'Nama Nilai Variabel', a dropdown menu, and two buttons labeled 'Tambah' and 'Cancel'. A large central box contains the text 'Tampilan Semua Data Variabel' and a logo of Universitas Islam Negeri.

Gambar 4.13 Antar muka Halaman Nilai Variabel

e. Rancangan Antarmuka Halaman Basis Pengetahuan

Rancangan antarmuka Halaman basis pengetahuan ini adalah halaman basis pengetahuan Aplikasi Representasi Basis Pengetahuan Berbasis Aturan. Halaman ini merupakan proses untuk melakukan input data basis pengetahuan yang berupa sebuah pernyataan dan selanjutnya dari pernyataan tersebut akan di rubah menjadi aturan-aturan.(Gambar 4.14).



Gambar 4.15 Antarmuka Halaman Daftar Aturan

g. Rancangan Antarmuka Halaman Pengujian

Rancangan antarmuka halaman pengujian ini adalah halaman pengujian Aplikasi Representasi Basis Pengetahuan Berbasis Aturan. Halaman ini merupakan proses untuk melakukan pengujian terhadap aturan-aturan yang terdapat pada basis pengetahuan. (Gambar 4.16).

The screenshot shows a testing page interface with the following elements:

- Header:** A box labeled "Header" at the top center.
- Left Sidebar:** A box labeled "Menu" on the left side.
- Form Fields:**
 - A box labeled "Nama Kesur" followed by an input field containing the letter "M".
 - Below it, two buttons: "Pilih" and "Cancel".
 - A box labeled "Tampilkan Semua Data Variabel" followed by a "Simpan" button.
 - A larger box labeled "Tampilkan Pernyataan yang akan diuji:" containing a large empty text area.
 - Below this box, two buttons: "Ok" and "Cancel".
 - A box labeled "Tampilkan Keseluruhan" at the bottom of the main content area.
- Footer:** A box labeled "Footer" at the bottom center.

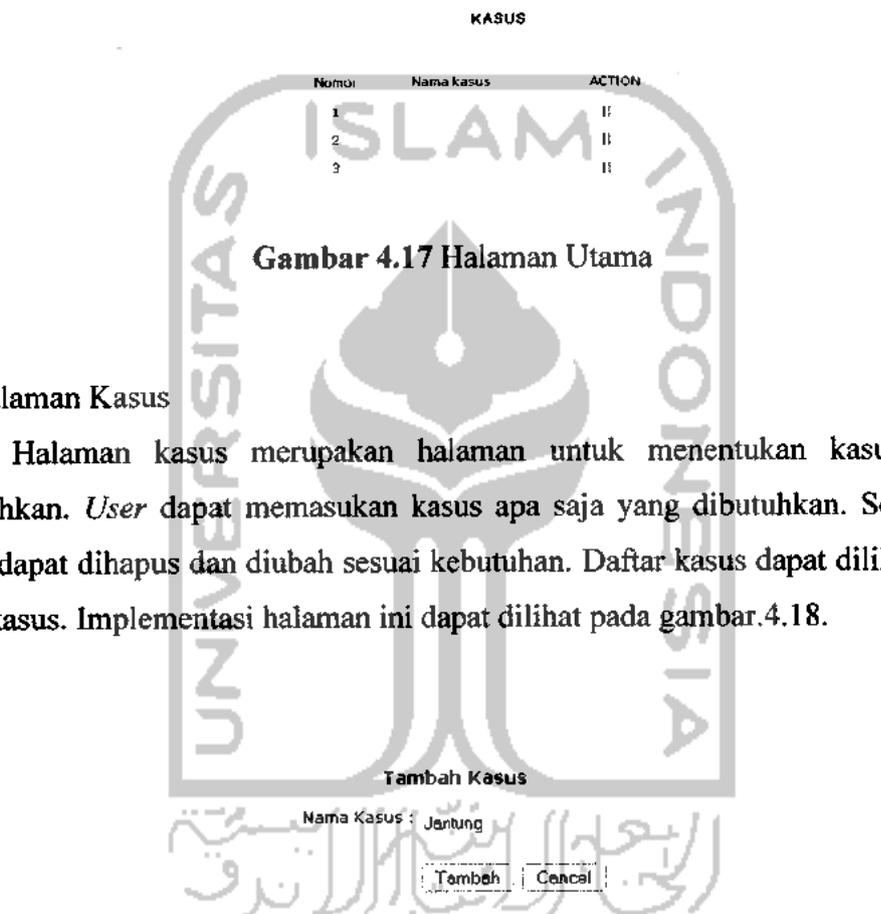
Gambar 4.16 Antar muka Halaman Pengujian

4.3 Implementasi Perangkat Lunak

Implementasi antar muka dibuat sesederhana mungkin sehingga dapat mempermudah *user* dalam menggunakan sistem. Halaman ini memuat menu utama yang ada dalam sistem.

a. Halaman Utama

Halaman ini akan muncul pertama kali saat sistem di akses. Pada halaman ini terdapat beberapa menu utama, yaitu menu kasus, menu basis pengetahuan, menu daftar aturan dan menu pengujian. Implementasi dari halaman Utama ditujukan pada Gambar 4.17.



Gambar 4.17 Halaman Utama

b. Halaman Kasus

Halaman kasus merupakan halaman untuk menentukan kasus yang dibutuhkan. *User* dapat memasukan kasus apa saja yang dibutuhkan. Selain itu kasus dapat dihapus dan diubah sesuai kebutuhan. Daftar kasus dapat dilihat pada tabel kasus. Implementasi halaman ini dapat dilihat pada gambar.4.18.

Tampil Data Kasus

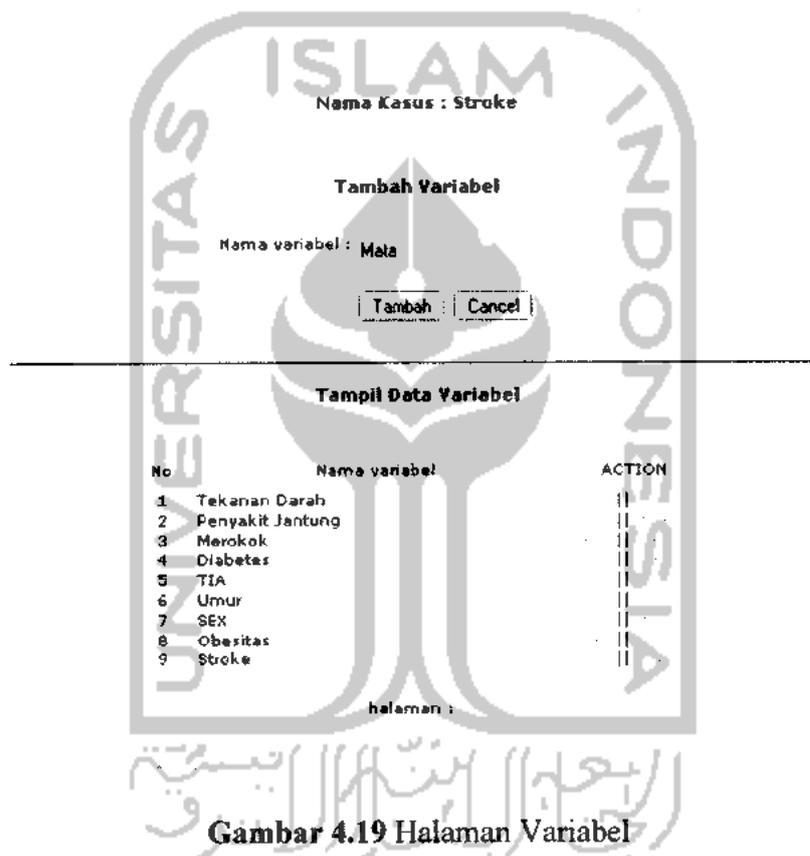
Nomor	Nama Kasus	ACTION
1		II
2		II
3		II

halaman :

Gambar 4.18 Halaman Kasus

c. Halaman Variabel

Halaman variabel merupakan halaman untuk menentukan variabel. *User* dapat memasukan variabel apa saja yang dibutuhkan. Selain itu variabel dapat dihapus dan diubah sesuai kebutuhan. Daftar variabel dapat dilihat pada tabel variabel. Implementasi halaman ini dapat dilihat pada gambar 4.19.



Gambar 4.19 Halaman Variabel

d. Halaman Nilai Variabel

Halaman nilai variabel merupakan halaman untuk menentukan nilai variabel yang dibutuhkan. *User* memasukan nilai variabel dari setiap variabel yang telah dipilih terlebih dahulu. Selain dapat ditambah nilai variabel dapat dihapus dan diubah sesuai kebutuhan. Daftar nilai variabel dapat dilihat pada tabel nilai variabel. Implementasi halaman ini dapat dilihat pada gambar 4.20.

Nama Kasus : Stroke

Tambah Nilai variabel

Nama variabel : Tekanan Darah
 Nama Nilai variabel : Sedang

Tambah Cancel

Tampil Data variabel

ID nilai	Name Sub variabel	Nilai	ACTION
1	Tekanan Darah	Tinggi	
2	Tekanan Darah	normal	
3	Penyakit Jantung	Ya	
4	Penyakit Jantung	Tidak	
5	Merokok	Ya	
6	Merokok	tidak	
7	Diabetes	Ya	
8	Diabetes	Tidak	
9	TIA	Ya	
10	Umur	Tua	

halaman :

Gambar 4.20 Halaman Nilai Variabel

e. **Halaman Daftar Aturan**

Halaman nilai variabel merupakan halaman untuk melihat semua aturan yang telah terbentuk dari pernyataan yang telah dimasukan oleh *user*. Implementasi halaman ini dapat dilihat pada gambar 4.21.

BAB V

PENGUJIAN

Dalam bab ini akan dijelaskan tentang pengujian Aplikasi Representasi Basis Pengetahuan Berbasis Aturan yang telah dibangun. Dengan pengujian ini diharapkan tingkat kesalahan baik dalam proses pengolahan data maupun dari sistem itu sendiri menjadi sangat kecil.

5.1 Pengujian pada Proses Konfigurasi Basis Pengetahuan

Halaman basis pengetahuan merupakan halaman untuk memproses pernyataan-pernyataan yang dimasukkan oleh *user* menjadi aturan-aturan yang lebih sederhana. *User* memilih terlebih dahulu kasus yang ingin dibuat pernyataannya, setelah *user* memilih kasus akan tampil daftar variabel-variabel yang berhubungan dengan kasus tersebut. Untuk membentuk variabel-variabel menjadi pernyataan *user* memilih variabel-variabelnya, setelah variabel dipilih, *user* menentukan tanda kurung buka, tanda kurung tutup, negasi, nilai variabel dan operator yang dipilih pada tiap variabel. Setelah pernyataan terbentuk dengan benar sistem akan memproses pernyataan tersebut menjadi aturan-aturan yang lebih sederhana. Aturan-aturan ini dapat dilihat pada 3 tabel yaitu tabel aturan, tabel blok aturan dan tabel kesimpulan Implementasi proses basis pengetahuan dapat dilihat pada gambar 5.1.

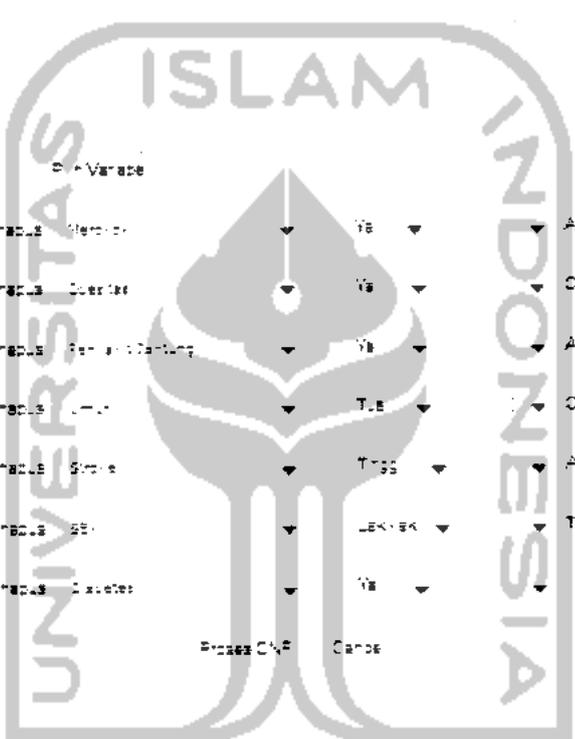
Pilih Kasus

Kasus : Stroke ▼

Pilih Kasus Cancel

Daftar Variabel :

Tekanan Darah
TIA



Daftar Variabel

▼	Tegak Merokok	▼	Ya	▼	AND	▼
▼	Tegak Obesitas	▼	Ya	▼	OR	▼
▼	Tegak Penyakit Jantung	▼	Ya	▼	AND	▼
▼	Tegak Umur	▼	Tua	▼	OR	▼
▼	Tegak Stroke	▼	Tinggi	▼	AND	▼
▼	Tegak SEI	▼	Laki-laki	▼	THEN	▼
▼	Tegak Diabetes	▼	Ya	▼		▼

Proses OR Cancel

Pilih Kasus

Kasus : Stroke ▼

Pilih Kasus Cancel

Pernyataan :

IF Merokok Ya AND (Obesitas Ya OR Penyakit Jantung Ya AND Umur Tua) OR Stroke Tinggi AND SEI Laki-laki THEN Diabetes Ya

Aturan yang Terbentuk :

IF Merokok Ya AND Obesitas Ya THEN Diabetes Ya

IF Stroke Tinggi AND SEI Laki-laki THEN Diabetes Ya

IF Merokok Ya AND Penyakit Jantung Ya AND Umur Tua THEN Diabetes Ya

Gambar 5.1 Gambar Proses Penyerdehanaan Aturan

- o IF Merokok Ya AND Penyakit Jantung Ya AND Umur Tua Ya THEN Diabetes Ya
- o IF Stroke Tinggi AND SEX Laki-Laki THEN Diabetes Ya

Sehingga dari pernyataan :

- IF Merokok Ya AND (Obesitas Ya OR Penyakit Jantung Ya AND Umur Tua) OR Stroke Tinggi AND SEX Laki-Laki THEN Diabetes Ya

Dapat disederhanakan menjadi aturan :

- o IF Merokok Ya AND Obesitas Ya THEN Diabetes Ya
- o IF Stroke Tinggi AND SEX Laki-Laki THEN Diabetes Ya
- o IF Merokok Ya AND Penyakit Jantung Ya AND Umur Tua Ya THEN Diabetes Ya

Pseudocode dari proses penyederhanaan proses pernyataan menjadi aturan sebagai berikut :

```

stringOr ← "OR";
stringAnd ← "AND";
stringAnswer ;
debugCount ← 0;

Fungsi orOperation(varA, varB) {
    if(isAbleToParse (varA) = true){
        parsingString(varA);
    }
    Else{
        stringAnswer[] ← varA;
    }

    if(isAbleToParse (varB) = true){
        parsingString(varB);
    }
    Else{
        stringAnswer[] ← varB;
    }
}

```

```

Fungsi andOperation(varA, varB) {
  stringAns ← null;
  stringAnswer[] ← varA
  stringAnd ← varB;
}

Fungsi solve(string) {
  if (ada kurung buka, pada string){
    newVar ← parsingBracket(string);
  } else {
    newVar[] ← string;
  }

  for (i = 0 to jumlah newVar){
    if (ada kurung buka, pada newVar[i]) {
      solve(newVar[i]);
    } else {
      parsingString(newVar[i]);
    }
  }
  End for
}

Fungsi parsingBracket(string) {
  newVar;
  arrStrSplit;
  count ← 0;
  stringTemp ← null;
  skip ← false;

  for (i = 0 to jumlah string){
    char = pisahkan string dari huruf ke 1;
    if (char = kosong dan skip = false) {
      arrStrSplit[] ← stringTemp;
      stringTemp ← kosong;
    } else
      if char = kurung tutup dan skip = false) {
        arrStrSplit[] ← stringTemp;
        stringTemp ← kosong;
        skip ← true;
      } else {
        stringTemp ← char;
      }
  }

  arrStrSplit[] ← stringTemp;
  bracket ← checkBracket(arrStrSplit[0],
    arrStrSplit[1]);
}

```

```

if (bracket = 0) {
    nonBracket ← 1;
} else {
    nonBracket ← 0;
}

if (cek jika dalam arrStrSplit[bracket] masih
terdapat Or) {
    var ← arrStrSplit{bracket};
    if (cek jika jumlah nonBracket < bracket)
    {
        newVar[] ← arrStrSplit[nonBracket] .
            " " . var[0] . " " .
            arrStrSplit[2];
        newVar[] ← $arrStrSplit[nonBracket] .
            " " . var[1] . " " .
            arrStrSplit[2];
    } else {
        newVar[] ← var[0] . " " .
            arrStrSplit[nonBracket] . " " .
            arrStrSplit[2];
        newVar[] ← var[1] . " " .
            arrStrSplit[nonBracket] . " " .
            arrStrSplit[2];
    }
} else {
    if (cek jika jumlah nonBracket < bracket)
    {
        newVar[] ← arrStrSplit[nonBracket] .
            " " . arrStrSplit[bracket] . " " .
            arrStrSplit[2];
    } else {
        newVar[] ← arrStrSplit[bracket] . "
            " . arrStrSplit[nonBracket] . " "
            . arrStrSplit[2];
    }
}
return newVar;
End For
}

Fungsi checkBracket(varA, varB) {
    bracket ← -1;
    isEmpty ← false;
    varBSplit ← varB;

    if (cek jika varBSplit[0] = Or atau varBSplit[0] =
        And ) {
        bracket ← 0;
    } else {
        bracket ← 1;
    }
    return bracket;
}

```

```

Fungsi parsingString(string) {
  if (cek jika dalam string masih terdapat Or) {
    var ← stringOr;
    orOperation(var[0], var[1]);
  } else
    if (cek jika dalam string masih terdapat And) {
      var ← string;
      andOperation(var[0], var[1]);
    } else {
      stringAnswer[] ← string;
    }
}

Fungsi isAbleToParse(string) {
  able ← false;
  if (cek jika dalam string masih terdapat Or) {
    able ← true;
  }
  if (cek jika dalam string masih terdapat And) {
    able ← false;
  }
  return able;
}

Fungsi removeSpace(string) {
  space ← false;
  stringSebelumnya ← kosong;
  for (i = 0 to jumlah string){
    char ← string;

    if (cek jika char = null dan string Sebelumnya =
        null){
    } else {
      Temp ← char;
    }
    stringSebelumnya ← char;
  }
  return temp;
End For
}

Fungsi showAnswer() {
  stringAnswer ← stringAnswer;

  for (i = stringAnswer to no = val){
    temp[] ← removeSpace(val);
  }
}

```

```

stringAnswer ← temp;
jawab ← stringAnswer;
stringAnswer ← null;
return jawab;
}

```

5.2 Proses Inferensi

Pada proses inferensi aturan, *user* akan menguji aturan-aturan yang terdapat dalam basis pengetahuan yang diperoleh pada proses sebelumnya. Pada proses inferensi ini *user* memasukkan pernyataan seperti pada proses sebelumnya tetapi tanpa menyertakan kesimpulan dari pernyataan tersebut, kemudian pernyataan tersebut akan dibentuk aturan-aturan yang lebih sederhana melalui bentuk *CNF*. Dari aturan tersebut akan dicocokkan dengan aturan-aturan yang terdapat pada basis pengetahuan. Hasil akhir dari proses inferensi ini adalah kesimpulan dari tiap aturan. Implementasi proses inferensi dapat dilihat pada gambar 5.2 dan 5.3

Pilih Kasus

Kasus : Stroke

Pilih Kasus Cancel

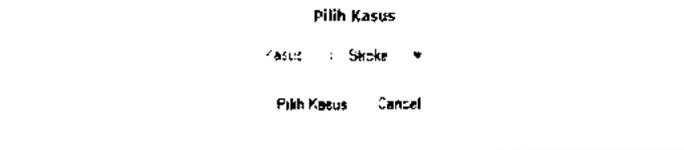
Daftar Variabel :

TIA
Umur
SEX
Obesitas
Stroke
Diabetes

Pilih Variabel

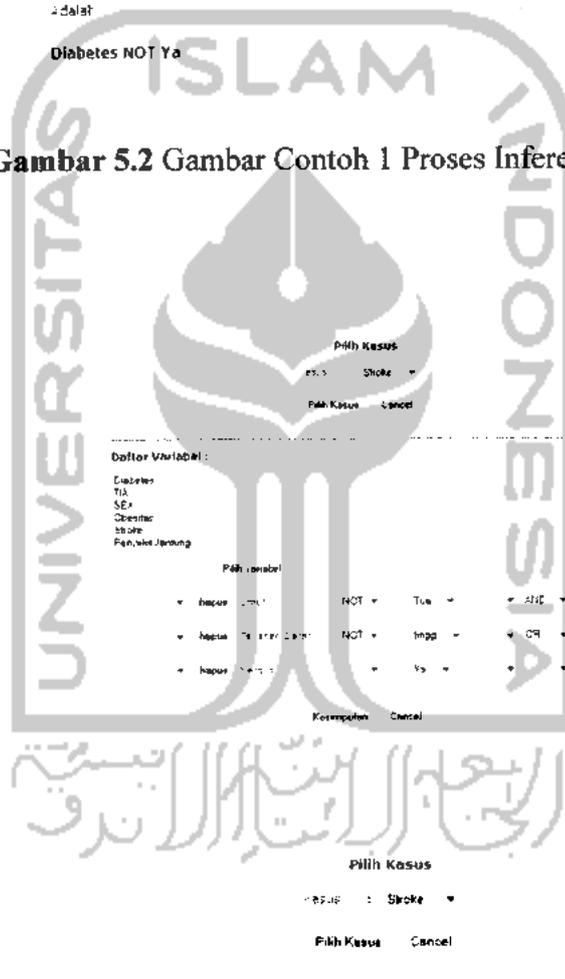
▼ hapus	Tekanan Darah	▼ Tidak	▼ AND
▼ hapus	Perilaku Berjalan	▼ Ya	▼ AND
▼ hapus	Merokok	▼ Ya	▼ AND

Kesimpulan Cancel



Resimpulan Dan Alasan :
 IF Tekanan Darah tinggi AND Penyakit Jantung YA AND Hiperlipidemia
 adalah
 Diabetes NOT Ya

Gambar 5.2 Gambar Contoh 1 Proses Inferensi



Gambar 5.3 Gambar Contoh 2 Proses Inferensi

```

for (i = 0 to count($_POST[id_variabel])) {
    stringVar2 ← $_POST[not][i] $_POST[nilai_variabel][i];
}

lihat_id_aturan ← mysql_query("select DISTINCT(id_aturan)
                               from blok_aturan");

while (id_aturan = mysql_fetch_array(lihat_id_aturan)) {
    lihat_blok_aturan ← mysql_query("
        select negasi, id_nilai_variabel from blok_aturan
        where id_aturan = 'id_aturan[0]'
    ");

    while (blok_aturan =
        mysql_fetch_assoc(lihat_blok_aturan)) {
        temp_aturan[] = blok_aturan;
    }

    data_blok_aturan[] = array("id_aturan" => id_aturan[0],
                               "isi" => temp_aturan);
    temp_aturan = null;
}

foreach (jawaban2 as no => val) {
    if (is_array(val)) {
        foreach (val as n => v) {
            if (count(v) == 2) {
                variable = v[1];
                not = 0;
            } else {
                variable = v[0];
                not = 1;
            }

            temp_aturan_uji[] =
                array("negasi" => not,
                    "id_nilai_variabel" =>
                    variable);
        }

    } else {
        if (count(v) == 2) {
            variable = v[1];
            not = 0;
        } else {
            variable = v[0];
            not = 1;
        }

        temp_aturan_uji[] =
            array("negasi" => not, "id_nilai_variabel"
                => variable);
    }
}

```

```

data_blok_aturan_uji[] =
array("isi" => temp_aturan_uji);
temp_aturan_uji = null;
}

// pengecekan dari data base

foreach (data_blok_aturan_uji as no => blok_aturan_uji) {
    foreach (data_blok_aturan as id => blok_aturan) {
        if (blok_aturan_uji[isi] == blok_aturan[isi]) {
            array_kes[] ← blok_aturan["id_aturan"];
        } else {
            array_kes[] ← "-1";
        }
    }
}

// tampil jawaban

foreach (array_kes as no => val) {
    id_aturan ← val;

    if (id_aturan ← "-1") {
        string_kesimpulan ← " kesimpulan tidak ada ";
    } else {
        lihat_kesimpulan = mysql_query("
        select * from kesimpulan, aturan
        where
            kesimpulan.id_kesimpulan =
            aturan.id_kesimpulan and
            aturan.id_aturan = 'id_aturan'
        ");
        kesimpulan =
            mysql_fetch_array(lihat_kesimpulan);
        negasi_kesimpulan ← "0";
        negasi_kesimpulan ← kesimpulan[negasi];
        id_variabel_kesimpulan ←
            kesimpulan[id_nilai_variabel];
        string_kesimpulan ← "";

        if (negasi_kesimpulan == "0") {
            string_kesimpulan ← "NOT";
        } else {
            string_kesimpulan ← "";
        }
    }
}

```

```

lihat_id_nama_variabel = mysql_query("
select * from nilai_variabel, variabel,
      kasus
where
      nilai_variabel.id_variabel =
      variabel.id_variabel and
      variabel.id_kasus = kasus.id_kasus
      and
      kasus.id_kasus = $ksks and
      nilai_variabel.id_nilai_variabel =
      '$id_variabel_kesimpulan'
");

variabel =
mysql_fetch_array($lihat_id_nama_variabel);
string_kesimpulan = variabel[nama_variabel]
variabel[nama_nilai_variabel];
}

```

Tabel 5.1 Tabel Hasil Pengujian

No	Pernyataan	Aturan
1	If A And B And C Then D	<ul style="list-style-type: none"> • If A And B And C Then D
2	If A And B Or C Then D	<ul style="list-style-type: none"> • If A And B Then D • If C Then D
3	If A Or B then C	<ul style="list-style-type: none"> • If A Then C • If B Then C
4	If A and B and C or D and E and F or G and H then K	<ul style="list-style-type: none"> • If A and B and C then K • If D and E and F then K • If G and H then K
5	IF A and (B or C) then D	<ul style="list-style-type: none"> • If A and B then D • If B and C then D
6	IF (A and B) or (C and D and E) then F	<ul style="list-style-type: none"> • If A and B then F • If C and D and E then F

7	If A and B and (C or D) and E and (F or G) and H then I	<ul style="list-style-type: none"> • If A and B and C and E and F and H then I • If A and B and C and E and G and H then I • If A and B and D and E and F and H then I • If A and B and D and E and G and H then I
8	If A and (C or D) and E or H then B	<ul style="list-style-type: none"> • If A and C and E then B • If A and D and E then B • If H then B
9	If A Or (B And C Or D) And E Or (F And G Or H) And I Then J	<ul style="list-style-type: none"> • If A Then J • If B And C And E Then J • If F And G And I Then J • If H And I Then J • If D And E Then J
10	If A And B And (C Or D And E) Or G And F Or H And I Then J	<ul style="list-style-type: none"> • If A And B And C Then J • If G And F Then J • If H And I Then J • If A And B And D And E Then J

BAB VI

PENUTUP

6.1 Simpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan maka penulis dapat mengambil beberapa kesimpulan, sebagai berikut :

1. Aplikasi Representasi Basis Pengetahuan dengan menggunakan metode berbasis aturan ini dapat membantu *user* dalam menyelesaikan beberapa masalah dalam pencarian solusi berdasarkan aturan-aturan yang telah ada di dalam basis pengetahuan.
2. Dengan mengubah pernyataan menjadi beberapa aturan, maka proses inferensi tidak melalui proses resolusi dengan menggunakan *conjunctive normal form* (CNF).

6.2 Saran

Berdasarkan hasil penelitian yang telah penulis kerjakan, terdapat kelemahan-kelemahan pada sistem yang penulis telah bangun, meliputi :

1. Dalam proses pembentukan pernyataan sistem hanya mampu menangani satu tingkat atau satu buah kurung dalam tiap pernyataan.
2. Komponen-komponen dalam pembentukan pernyataan seperti tanda buka kurung, operator not, operator and, operator or dan tanda tutup kurung disusun lebih dinamis lagi agar lebih memudahkan *user* dalam proses pembentukan pernyataan.

Kelemahan-kelemahan di atas memerlukan saran agar Aplikasi Representasi Basis Pengetahuan Berbasis Aturan menjadi lebih baik, saran tersebut sebagai berikut:

1. Sebaiknya dalam proses mengubah pernyataan menjadi aturan-aturan ditambahkan hukum De Morgan.

2. Sebaiknya penggunaan operator menjadi lebih bervariasi, tidak hanya operator and, or dan not saja, sehingga sistem ini menjadi lebih kompleks.
3. Sebaiknya aplikasi ini dapat menangani proses pernyataan yang bertingkat.



DAFTAR PUSTAKA

- [KUS03] Kusumadewi, Sri. *Artificial Intelligence (teknik dan aplikasinya)*. Graha Ilmu, Yogyakarta. 2003.
- [ARI10] Aribowo Arnold, Frits Harris Kristian, Berlinton Sitorus Budi
<http://journal.uii.ac.id/index.php/Snati/article/view/548/472>, diakses tanggal 4 agustus 2010
- [RIS10] Riskadewi, Hendrik Antonius
<http://home.unpar.ac.id/~integral/Volume%2010/Integral%2010%20No.%203/Forward%20Chaining.pdf>, diakses tanggal 6 agustus 2010

UNIVERSITAS ISLAM INDONESIA
الرَّجَاءُ الْإِسْلَامِيَّةُ الْإِسْلَامِيَّةُ

DAFTAR PUSTAKA

- [KUS03] Kusumadewi, Sri. *Artificial Intelligence (teknik dan aplikasinya)*. Graha Ilmu, Yogyakarta. 2003.
- [ARI10] Aribowo Arnold, Frits Harris Kristian, Berlinton Sitorus Budi
<http://journal.uii.ac.id/index.php/Snati/article/view/548/472>, diakses tanggal 4 agustus 2010
- [RIS10] Riskadewi, Hendrik Antonius
<http://home.unpar.ac.id/~integral/Volume%2010/Integral%2010%20No.%203/Forward%20Chaining.pdf>, diakses tanggal 6 agustus 2010

UNIVERSITAS ISLAM INDONESIA
الرَّحْمَةُ الرَّحِيمِ
الرَّحْمَةُ الرَّحِيمِ

a. Tabel Kasus

Tabel Kasus merupakan tabel yang mendefinisikan kasus-kasus apa saja yang terdapat dalam sistem. Struktur tabel kasus ditunjukkan pada tabel 4.1.

Tabel 4.1 Tabel Kasus

No.	Nama Kolom	Tipe Data	Keterangan
1	<u>id_kasus</u>	Int(6)	Primary Key
2	nama_kasus	Text	

b. Tabel Variabel

Tabel Variabel merupakan tabel yang mendefinisikan variabel-variabel apa saja yang dibutuhkan dalam setiap kasus. Struktur tabel variabel ditunjukkan pada tabel 4.2.

Tabel 4.2 Tabel Variabel

No.	Nama Kolom	Tipe Data	Keterangan
1	<u>id_variabel</u>	Int(6)	Primary Key
2	nama_variabel	Text	
3	id_kasus	Int(6)	Foreign Key

c. Tabel Nilai Variabel

Tabel Nilai Variabel merupakan tabel yang mendefinisikan nilai variabel dari setiap variabel. Struktur tabel nilai variabel ditunjukkan pada tabel 4.3.

Tabel 4.3 Tabel Nilai Variabel

No.	Nama Kolom	Tipe Data	Keterangan
1	<u>id_nilai_variabel</u>	Int(6)	Primary Key
2	nama_nilai_variabel	Text	
3	id_variabel	Int(6)	Foreign Key
4	id_kasus	Int(6)	Foreign Key

d. Tabel Kesimpulan

Tabel kesimpulan merupakan tabel yang mendefinisikan kesimpulan dari tiap aturan yang ada. Struktur tabel kesimpulan ditunjukkan pada tabel 4.4

Tabel 4.4 Tabel Kesimpulan

No.	Nama Kolom	Tipe Data	Keterangan
1	id_kesimpulan	Int(6)	Primary Key
2	Negasi	Enum('0','1')	
3	id_nilai_variabel	Int(6)	Foreign Key

e. Tabel Blok Aturan

Tabel Blok Aturan merupakan tabel yang mendefinisikan aturan-aturan yang ada. Struktur tabel aturan ditunjukkan pada tabel 4.6.

Tabel 4.5 Tabel Blok Aturan

No.	Nama Kolom	Tipe Data	Keterangan
1	id_blok_aturan	Int(6)	Primary Key
2	Negasi	Enum('0','1')	
3	id_nilai_variabel	Int(6)	Foreign Key
4	id_aturan	Int(6)	Foreign Key

f. Tabel Aturan

Tabel aturan merupakan tabel yang mendefinisikan aturan beserta kesimpulan yang telah dibentuk sebelumnya. Struktur tabel aturan ditunjukkan pada tabel 4.6.

Tabel 4.6 Tabel Aturan

No.	Nama Kolom	Tipe Data	Keterangan
1	id_aturan	Int(6)	Primary Key
2	id_aturan	Enum('0','1')	

g. Tabel Daftar

Tabel daftar merupakan tabel yang berisi semua aturan beserta kesimpulan sebelum aturan-aturan tersebut diproses. Struktur tabel daftar ditunjukkan pada tabel 4.7.

Tabel 4.7 Tabel Daftar

No.	Nama Kolom	Tipe Data	Keterangan
1	id_daftar	Int(6)	Primary Key
2	Negasi	Enum('0','1')	
3	id_nilai_variabel	Int(6)	Foreign Key
4	Operator	text	

4.2.5 Relasi Tabel

Dalam pembuatan sistem, relasi antar tabel bertujuan untuk menjelaskan hubungan antar tabel disertai dengan bentuk hubungannya. Relasi antar tabel yang digunakan untuk pembuatan sistem dapat dilihat pada Gambar 4.9.