

**SISTEM PAKAR PADA UNDANG – UNDANG REPUBLIK
INDONESIA NOMOR 19 TAHUN 2002 TENTANG HAK CIPTA
(PRO-ES)**

Diajukan Untuk Memenuhi Salah Satu Syarat
Dalam Menempuh Ujian Sidang Sarjana
di Jurusan Teknik Informatika

Oleh :
DIGY DEEF
0699059/43957006990127



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK UNIVERSITAS WIDYATAMA
BANDUNG
2005**

**SISTEM PAKAR PADA UNDANG-UNDANG REPUBLIK INDONESIA
NOMOR 19 TAHUN 2002 TENTANG HAK CIPTA
(PRO-ES)**

DIGY DEEF

43957006990127/0699059

Telah diperiksa dan disetujui untuk diajukan pada Sidang Tugas Akhir Program
Sarjana Fakultas Teknik Jurusan Teknik Informatika
Universitas Widyatama

Bandung, Maret 2005

Menyetujui

Pembimbing I,

Pembimbing II,

Dr. Ana Hadiana, Ir., M.Eng.

M. Ibnu Choldun, S.T., M.T.

Ketua Jurusan Teknik Informatika

Imam Rozali, S.T., M.T.

LEMBAR PENGESAHAN

Nama Mahasiswa : Digy Deef
NIM : 0699059
Judul Skripsi : Sistem Pakar Pada Undang-Undang Republik Indonesia
Nomor 19 Tahun 2002 Tentang Hak Cipta

Telah sidang pada tanggal **30 Maret 2005**

Yang bersangkutan dinyatakan lulus dengan nilai _____

PENILAI	TANDA TANGAN
Pembimbing : 1. Dr. Ana Hadiana, Ir., M.Eng. 2. M. Ibnu Choldun, S.T., M.T.	
Penguji : 1. Sunjana, S.Si., M.T. 2. Sukenda, S.Kom., M.T.	

Mengetahui
Ketua Jurusan Teknik Informatika,

Imam Rozali, S.T., M.T.

Bandung, 30 Maret 2005
Ketua Sidang,

Abdullah Fajar, S.Si.

ABSTRAKSI

Bidang Hukum merupakan bidang yang sangat jarang disentuh oleh teknologi informasi. Hal ini dikarenakan bidang ini berkaitan erat dengan masalah yang terjadi di masyarakat mulai masalah perdata sampai masalah pidana. Pengambilan segala keputusan berkaitan dengan perkara – perkara diatas ditentukan oleh seorang hakim yang belum memungkinkan digantikan oleh teknologi. Seorang pengacara mempunyai peranan penting dalam memberikan argumen-argumen yang membantu seorang hakim mengambil keputusan.

Sistem pakar ini mampu melakukan pencarian pasal, pencarian sanksi-sanksi terhadap pelanggaran hukum pada undang - undang Republik Indonesia nomor 19 tahun 2002 tentang Hak Cipta, dan user juga bisa melakukan konsultasi dengan sistem dimana user menjawab pertanyaan-pertanyaan yang diajukan oleh sistem.

PRO-ES dikembangkan dengan menggunakan inferensi *Backward-Chaining* dan teknik pencarian *Depth First Search (DFS)*. PRO-ES dibangun dengan menggunakan aplikasi Borland Delphi 7.0 menggunakan basis data Paradox 7.0 dalam lingkungan sistem operasi berbasis Windows XP.

ABSTRACT

The Law field is touch by information technology infrequently. Because of the law related with the civil problem and criminal problem that can happen in the society. All about decision making of that problem made in judge. The judge's decision cannot replace with the technology. A lawyers are playing the figure to give argument for judge's decision.

Expert system developed by writer related with determining sanctions and searching sections in the breaking law. *Undang-Undang Republik Indonesia Nomor 19 Tahun 2002* “ is law umbrella for copyright's someone.

PRO-ES developed with Backward-Chaining inference and Depth Searching First (DFS) method for searching. PRO-ES developed with Borland Delphi 7.0 application, and Paradox 7.0 as the database with Windows XP as the operating system.

Keyword: PRO-ES, Backward-Chaining, Depth First Search, Expert System.

KATA PENGANTAR

Puji syukur Alhamdulillah, Penulis panjatkan ke hadirat Allah SWT yang telah menuntun penulis dalam menyelesaikan penyusunan laporan tugas akhir ini. Salam serta Shalawat penulis haturkan pula kepada Nabi Muhammad SAW.

Tugas akhir ini penulis beri judul “ Sistem pakar pada Undang–Undang Republik Indonesia Nomor 19 tahun 2002 tentang Hak Cipta (PRO-ES)” . Tugas Akhir ini penulis susun sebagai salah satu syarat wajib dalam menempuh jenjang S-1 pada Fakultas Teknik Jurusan Teknik Informatika Universitas Widyatama.

Dalam penyusunan Laporan Tugas Akhir ini penulis banyak mendapat bantuan dari berbagai pihak, baik berupa bimbingan, petunjuk maupun saran-saran. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang tak terhingga kepada :

1. Kedua Orang Tua tercinta, yang telah mendidik dan memberikan dukungan secara moral maupun spiritual kepada penulis sehingga penulis mampu menyelesaikan laporan Tugas Akhir ini.
2. Bapak Dr. Ana Hadiana, Ir., M.Eng., selaku pembimbing I dalam penyusunan laporan Tugas Akhir ini.
3. Bapak M. Ibnu Choldun, S.T., M.T., selaku pembimbing II dalam penyusunan laporan Tugas Akhir ini.
4. Bapak Imam Rozali, S.T., M.T., dan Bapak Abdulah Fajar , S.Si, selaku Ketua Jurusan dan Sekretaris Jurusan Teknik Informatika Universitas Widyatama.
5. Teman-teman kampus Erik, Wayan, Arik, Dewa, Ono dan teman-teman lainnya yang tidak sempat saya sebutkan. Terima kasih atas bantuannya.
6. Buat Anne Christ dan keluarga besarnya, terima kasih atas bantuan moril dan materiilnya. Semoga Allah membalas semua kebaikan kalian.
7. Terima Kasih kepada Bapak Addy wibowo yang telah membantu dan mendukung secara moril sehingga Tugas Akhir ini dapat terselesaikan dengan tepat waktu.

Penulis berusaha untuk menyelesaikan Tugas Akhir ini sebaik mungkin, tetapi penulis sadar akan keterbatasan dan kekurangan yang penulis miliki. Untuk itu penulis mohon maaf atas keterbatasan dan kekurangan laporan ini dan mengharapkan adanya kritik dan saran untuk pengembangan diri penulis di masa mendatang. Semoga Allah SWT senantiasa melimpahkan rahmat dan hidayah-Nya kepada kita. Amin

Bandung, Maret 2005

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN

ABSTRAKSI

ABSTRACT

KATA PENGANTAR	i
DAFTAR ISI	iii
DAFTAR TABEL	vii
DAFTAR GAMBAR.....	x
DAFTAR SIMBOL	xii
DAFTAR LAMPIRAN.....	xiv

BAB I PENDAHULUAN

1.1 Latar Belakang Masalah	I-2
1.2 Perumusan Masalah	I-3
1.3 Tujuan	I-3
1.4 Ruang Lingkup	I-4
1.5 Metodologi Pengembangan Sistem	I-4
1.6 Sistematika Penulisan	I-5

BAB II LANDASAN TEORI

2.1 Konsep Dasar Sistem Pakar	II-1
2.1.1 Latar Belakang Sistem Pakar	II-2
2.1.2 Lingkungan Operasional Sistem Pakar	II-2
2.1.3 Definisi Sistem Pakar	II-2
2.1.4 Typologi Aplikasi Sistem Pakar	II-2
2.1.5 Contoh Klasik dari Pakar Klasik	II-3
2.1.6 Contoh Kontemporer dari Sistem Pakar	II-4
2.2 Arsitektur Sistem Pakar	II-4
2.2.1 <i>Knowledge Acquisition Interface</i> (antarmuka akuisisi pengetahuan)	II-5
2.2.2 <i>User Interface</i>	II-5

2.2.3	<i>Knowledge Base (Basis Pengetahuan)</i>	II-5
2.2.4	<i>Inference Engine</i>	II-6
2.3	Karakteristik Sistem Pakar	II-7
2.4	Sistem Pakar dan Bentuk Pengetahuan Lain	II-8
2.5	Representasi Pengetahuan	II-9
2.6	Teknik Inferensi	II-14
2.6.1	<i>Reasoning</i>	II-14
2.6.2	<i>Inference</i>	II-15
2.6.3	<i>Forward-Chaining</i>	II-15
2.6.4	<i>Backward-Chaining</i>	II-16
2.7	Metode Pengembangan Sistem Pakar	II-19
2.7.1	Metodologi Rekayasa Perangkat Lunak dengan Waterfall	II-19
2.7.2	Meotde Rekayasa perangkat Lunak dengan Prototipe (<i>Prototyping Model</i>)	II-21
2.7.3	Metode Rekayasa Perangkat Lunak dengan Model RAD (<i>Rapid Application Development</i>)	II-21
2.7.4	Metode Rekayasa Perangkat Lunak dengan Model Inkremental (<i>Incremental Model</i>)	II-22
2.7.5	Metode Rekayasa Perangkat Lunak dengan Model Spiral (<i>Spiral Model</i>)	II-23
2.7.6	Metode Rekayasa Perangkat Lunak dengan Model Perakitan Komponen (<i>Componen Assembly Model</i>)	II-24
2.7.7	Metode Rekayasa Perangkat Lunak dengan Model Pengembangan Konkuren (<i>Concurent Development Model</i>)	II-23
2.7.8	Metode Rekayasa Perangkat Lunak dengan Model Metode Formal (<i>Formal Methods Model</i>)	II-24
2.8	Perangkat Lunak (<i>Software</i>)	II-24
2.8.1	<i>Software Sistem</i>	II-24
2.8.2	<i>Software Aplikasi</i>	II-26

BAB III ANALISIS SISTEM

3.1 Identifikasi Masalah	III-2
3.2 Penyebab Masalah	III-3
3.3 Pendekatan Solusi (<i>the approach of solution</i>)	III-3
3.4 Hasil Analisis Sistem	III-4
3.5 Hasil Analisis Pengetahuan	III-4

BAB IV PERANCANGAN SISTEM

4.1 Perancangan Basis Pengetahuan	IV-1
4.1.1. Rules	IV-1
4.1.1.1. <i>Rule</i> untuk gol 1	IV-1
4.1.1.2. <i>Rule</i> untuk gol 2	IV-4
4.1.1.3. <i>Rule</i> untuk gol 3	IV-5
4.1.1.4. <i>Rule</i> untuk gol 4	IV-6
4.1.1.5. <i>Rule</i> untuk gol 5	IV-7
4.1.1.6. <i>Rule</i> untuk gol 6	IV-10
4.1.1.7. <i>Rule</i> untuk gol 7	IV-12
4.1.1.8. <i>Rule</i> untuk gol 8	IV-13
4.1.1.9. <i>Rule</i> untuk gol 9	IV-14
4.1.2. <i>Inference Networking</i>	IV-15
4.1.2.1. <i>Inference Networking</i> untuk gol 1.....	IV-15
4.1.2.2. <i>Inference Networking</i> untuk gol 2.....	IV-16
4.1.2.3. <i>Inference Networking</i> untuk gol 3.....	IV-17
4.1.2.4. <i>Inference Networking</i> untuk gol 4.....	IV-18
4.1.2.5. <i>Inference Networking</i> untuk gol 5.....	IV-19
4.1.2.6. <i>Inference Networking</i> untuk gol 6.....	IV-20
4.1.2.7. <i>Inference Networking</i> untuk gol 7.....	IV-21
4.1.2.8. <i>Inference Networking</i> untuk gol 8.....	IV-21
4.1.2.9. <i>Inference Networking</i> untuk gol 9.....	IV-22
4.2 <i>Data Flow Diagram</i> (Diagram Aliran Data).....	IV-22
4.2.1. <i>Data Context Diagram</i>	IV-22
4.2.2. <i>Data Flow Diagram Level 1</i>	IV-23

4.2.3. <i>Data Flow Diagram Level 2</i> proses 1	IV-24
4.2.4. <i>Data Flow Diagram Level 2</i> proses 2	IV-27
4.2.5. <i>Data Flow Diagram Level 2</i> proses 3	IV-29
4.2.6. <i>Data Flow Diagram Level 2</i> proses 4	IV-30
4.2.7. <i>Data Flow Diagram Level 2</i> proses 5	IV-32
4.2.8. <i>Data Flow Diagram Level 2</i> proses 6	IV-35
4.2.9. <i>Data Flow Diagram Level 2</i> proses 7	IV-38
4.3 Perancangan Data	IV-41
4.3.1. <i>Entity Relationship Diagram</i>	IV-41
4.3.2. Kamus Data (<i>Data Dictionary</i>)	IV-42
4.4 Perancangan Antar Muka	IV-44

BAB V IMPLEMENTASI SISTEM

5.1. Lingkungan Implementasi	V-1
5.1.1. Lingkungan Perangkat Keras (<i>hardware</i>).....	V-1
5.1.2. Lingkungan Perangkat Lunak (<i>software</i>).....	V-1
5.2. Implementasi Basis Data (<i>Database</i>).....	V-1
5.3. Implementasi Proses.....	V-4
5.4. Implementasi Antar Muka.....	V-11

BAB VI PENUTUP

6.1. Kesimpulan	VI-1
6.2. Saran.....	VI-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 2.1	Perbandingan Sistem Pakar dengan Buku.....	II-9
Tabel 2.2	Perbandingan Sistem Pakar dengan Seorang Pakar	II-9
Tabel 2.3	Tipe Pengetahuan	II-10
Tabel 4.1	<i>Rule 1</i>	IV-1
Tabel 4.2	<i>Rule 2</i>	IV-2
Tabel 4.3	<i>Rule 3</i>	IV-2
Tabel 4.4	<i>Rule 4</i>	IV-2
Tabel 4.5	<i>Rule 5</i>	IV-2
Tabel 4.6	<i>Rule 6</i>	IV-3
Tabel 4.7	<i>Rule 7</i>	IV-3
Tabel 4.8	<i>Rule 8</i>	IV-3
Tabel 4.9	<i>Rule 9</i>	IV-4
Tabel 4.10	<i>Rule 10</i>	IV-4
Tabel 4.11	<i>Rule 11</i>	IV-4
Tabel 4.12	<i>Rule 12</i>	IV-5
Tabel 4.13	<i>Rule 13</i>	IV-5
Tabel 4.14	<i>Rule 14</i>	IV-5
Tabel 4.15	<i>Rule 15</i>	IV-6
Tabel 4.16	<i>Rule 16</i>	IV-6
Tabel 4.17	<i>Rule 17</i>	IV-6
Tabel 4.18	<i>Rule 18</i>	IV-7
Tabel 4.19	<i>Rule 19</i>	IV-7
Tabel 4.20	<i>Rule 20</i>	IV-7
Tabel 4.21	<i>Rule 21</i>	IV-8
Tabel 4.22	<i>Rule 22</i>	IV-8
Tabel 4.23	<i>Rule 23</i>	IV-8
Tabel 4.24	<i>Rule 24</i>	IV-9
Tabel 4.25	<i>Rule 25</i>	IV-9
Tabel 4.26	<i>Rule 26</i>	IV-9
Tabel 4.27	<i>Rule 27</i>	IV-10

Tabel 4.28	<i>Rule 28</i>	IV-10
Tabel 4.29	<i>Rule 29</i>	IV-10
Tabel 4.30	<i>Rule 30</i>	IV-11
Tabel 4.31	<i>Rule 31</i>	IV-11
Tabel 4.32	<i>Rule 32</i>	IV-11
Tabel 4.33	<i>Rule 33</i>	IV-12
Tabel 4.34	<i>Rule 34</i>	IV-12
Tabel 4.35	<i>Rule 35</i>	IV-12
Tabel 4.36	<i>Rule 36</i>	IV-13
Tabel 4.37	<i>Rule 37</i>	IV-13
Tabel 4.38	<i>Rule 38</i>	IV-13
Tabel 4.39	<i>Rule 39</i>	IV-14
Tabel 4.40	<i>Rule 40</i>	IV-14
Tabel 4.41	Deskripsi Proses untuk Proses Penerimaan Jawaban.....	IV-25
Tabel 4.42	Deskripsi Proses untuk Proses Generate Pertanyaan.....	IV-26
Tabel 4.43	Deskripsi Proses untuk Proses Hasil Tanya Jawab	IV-26
Tabel 4.44	Deskripsi Proses untuk Proses Input Pasal.....	IV-27
Tabel 4.45	Deskripsi Proses untuk Proses Update Pasal.....	IV-28
Tabel 4.46	Deskripsi Proses untuk Proses Pencarian Pasal.....	IV-28
Tabel 4.47	Deskripsi Proses untuk Proses Hasil Pencarian.....	IV-28
Tabel 4.48	Deskripsi Proses untuk Proses Akumulasi Jawaban	IV-29
Tabel 4.49	Deskripsi Proses untuk Proses Penyimpanan Jawaban	IV-30
Tabel 4.50	Deskripsi Proses untuk Proses Pembacaan Jawaban.....	IV-31
Tabel 4.51	Deskripsi Proses untuk Proses Pencarian Gol.....	IV-31
Tabel 4.52	Deskripsi Proses untuk Proses Hasil Pencarian Gol.....	IV-31
Tabel 4.53	Deskripsi Proses untuk Proses Pemilihan Operasi.....	IV-33
Tabel 4.54	Deskripsi Proses untuk Proses Menambah <i>Knowledge Based</i>	IV-33
Tabel 4.55	Deskripsi Proses untuk Proses Mengubah <i>Knowledge Based</i>	IV-34
Tabel 4.56	Deskripsi Proses untuk Proses Menghapus <i>Knowledge Based</i>	IV-34

Tabel 4.57	Deskripsi Proses untuk Proses Hasil Operasi <i>Knowledge Based</i>	IV-34
Tabel 4.58	Deskripsi Proses untuk Proses Pembagian Informasi	IV-36
Tabel 4.59	Deskripsi Proses untuk Proses Penyajian Informasi Konsultasi.....	IV-36
Tabel 4.60	Deskripsi Proses untuk Proses Penyajian Informasi Pasal	IV-37
Tabel 4.61	Deskripsi Proses untuk Proses Penyajian Informasi <i>Knowledge Based</i>	IV-37
Tabel 4.62	Deskripsi Proses untuk Proses Penyajian Informasi <i>Password</i>	IV-37
Tabel 4.63	Deskripsi Proses untuk Proses <i>Update Password</i>	IV-39
Tabel 4.64	Deskripsi Proses untuk Proses <i>Input Password</i>	IV-39
Tabel 4.65	Deskripsi Proses untuk Proses Pembacaan <i>Password</i>	IV-40
Tabel 4.66	Deskripsi Proses untuk Proses <i>Data Invalid</i>	IV-40
Tabel 4.67	Deskripsi Proses untuk Proses <i>Data Valid</i>	IV-40
Tabel 4.68	Kamus Data	IV-42
Tabel 5.1	Tabel Pasal.....	V-2
Tabel 5.2	Tabel <i>Knowledge_Base</i>	V-2
Tabel 5.3	Tabel Rule	V-2
Tabel 5.4	Tabel Gol.....	V-2
Tabel 5.5	Tabel Pertanyaan	V-2

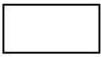
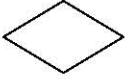


DAFTAR GAMBAR

Gambar 2.1	Arsitektur Sistem Pakar	II-4
Gambar 2.2	<i>Object-Attribut-Value</i>	II-11
Gambar 2.3	Contoh untuk Frame Burung	II-13
Gambar 2.4	<i>Semantic Network</i> dari Burung	II-13
Gambar 2.5	Proses <i>Inferensi</i> dalam Sistem Pakar	II-15
Gambar 2.6	Contoh <i>Inference Network</i>	II-17
Gambar 2.7	<i>Backword Chaining Inference</i> dalam <i>Rule Based System</i>	II-19
Gambar 2.8	Metode RPL dengan Metode Waterfall	II-20
Gambar 2.9	Metode RPL dengan Metode Prototype	II-21
Gambar 2.10	Metode RPL dengan Model RAD	II-22
Gambar 2.11	Metode RPL dengan Model Inkremental	II-22
Gambar 4.1	<i>Inference Networking</i> untuk Gol 1	IV-15
Gambar 4.2	<i>Inference Networking</i> untuk Gol 2	IV-16
Gambar 4.3	<i>Inference Networking</i> untuk Gol 3	IV-17
Gambar 4.4	<i>Inference Networking</i> untuk Gol 4	IV-18
Gambar 4.5	<i>Inference Networking</i> untuk Gol 5	IV-19
Gambar 4.6	<i>Inference Networking</i> untuk Gol 6	IV-20
Gambar 4.7	<i>Inference Networking</i> untuk Gol 7	IV-21
Gambar 4.8	<i>Inference Networking</i> untuk Gol 8	IV-21
Gambar 4.9	<i>Inference Networking</i> untuk Gol 9	IV-22
Gambar 4.10	<i>Data Context Diagram</i> untuk Sistem Pakar	IV-23
Gambar 4.11	<i>Data Flow Diagram Level 1</i> untuk Sistem Pakar	IV-24
Gambar 4.12	<i>Data Flow Diagram Level 2</i> untuk Proses 1	IV-25
Gambar 4.13	<i>Data Flow Diagram Level 2</i> untuk Proses 2	IV-27
Gambar 4.14	<i>Data Flow Diagram Level 2</i> untuk Proses 3	IV-29
Gambar 4.15	<i>Data Flow Diagram Level 2</i> untuk Proses 4	IV-30
Gambar 4.16	<i>Data Flow Diagram Level 2</i> untuk Proses 5	IV-32
Gambar 4.17	<i>Data Flow Diagram Level 2</i> untuk Proses 6	IV-35
Gambar 4.18	<i>Data Flow Diagram Level 2</i> untuk Proses 7	IV-38



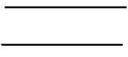


Gambar 4.19	<i>Entity Relationship Diagram</i>	IV-41
Gambar 4.20	<i>Form Utama Sistem Pakar</i>	IV-44
Gambar 4.21	<i>Form Sesi Konsultasi</i>	IV-44
Gambar 4.22	<i>Form Hasil Konsultasi</i>	IV-45
Gambar 4.23	<i>Form Login Expert</i>	IV-45
Gambar 4.24	<i>Form Update Rule</i>	IV-46
Gambar 4.25	<i>Form Pencarian Pasal</i>	IV-46
Gambar 5.1	<i>Form Utama</i>	V-11
Gambar 5.2	<i>Sesi Konsultasi</i>	V-12
Gambar 5.3	<i>Pencarian Pasal</i>	V-12
Gambar 5.4	<i>Update Pasal</i>	V-13
Gambar 5.5	<i>Form Data Rule</i>	V-13
Gambar 5.6	<i>Form Data Gol</i>	V-14
Gambar 5.7	<i>Form Data Pertanyaan</i>	V-14

DAFTAR SIMBOL

Tabel 1. Simbol-simbol pada *Entity Relationship Diagram*

No.	Simbol	Keterangan
1.		<i>Rectangle</i> , menggambarkan <i>entity set</i> dari suatu <i>database</i> .
2.		<i>Diamond</i> , menggambarkan <i>relationship</i> atau hubungan antar <i>database</i> .
3.		<i>Circle</i> , menggambarkan atribut atau <i>field</i> dari suatu <i>database</i> .
4.		<i>Line</i> , menggambarkan garis hubungan antara atribut dengan <i>entity set</i> dan <i>entity set</i> dengan <i>relationship</i> .

Tabel 2. Simbol-simbol pada Diagram Alur Data

No.	Simbol	Keterangan
1.		<i>Terminator/External entity</i> (kata benda), menggambarkan entitas luar yang terlibat dengan sistem, dapat berupa sumber asal datangnya data (<i>source</i>), atau yang menerima data (<i>sink</i>).
2.		Proses (kata kerja), menggambarkan kegiatan yang dilakukan oleh sistem, dapat berupa prosedur yang memanipulasi atau mengolah data.
3.		<i>Data store</i> (kata benda), merupakan tempat penyimpanan data untuk direferensi atau diolah lebih lanjut.
4.		<i>Data flow</i> (kata benda), menggambarkan aliran data yang menunjukkan transportasi data atau informasi.
5.		<i>Control flow</i> (kata benda), menggambarkan aliran kontrol data atau kejadian, dapat berupa nilai <i>boolean</i> atau nilai diskrit.

DAFTAR LAMPIRAN

Lampiran A : Antarmuka Aplikasi

Lampiran B : *Listing* Program

BAB I

PENDAHULUAN

Dewasa ini perkembangan perangkat keras (*hardware*) dan perangkat lunak (*software*) komputer yang semakin pesat, sangat mendukung pengembangan sistem secara umum dan sistem-sistem yang berbasis pengetahuan (*knowledge based*) yang merupakan subsistem dari sistem kecerdasan buatan (*artificial intelegent*). Domain-domain permasalahan yang tadinya hanya dapat diselesaikan oleh seorang pakar atau orang yang ahli dalam bidang tertentu, sekarang sudah mulai banyak yang memanfaatkan sistem pakar untuk membantu penyelesaian problema-problema yang kerap kali muncul pada domain tersebut.

Sistem pakar tidak dimaksudkan sebagai suatu sistem yang akan menggantikan tenaga manusia atau pakar sebagai solusi dalam memecahkan suatu masalah, melainkan dimanfaatkan untuk membantu seorang pakar apabila seorang pakar telah mengalami kelelahan berfikir. Selain hal tersebut sistem pakar juga dapat digunakan oleh orang yang ingin memperdalam atau memperluas wawasan dalam bidang tertentu.

Dalam pengembangan suatu sistem pakar harus diimbangi dengan peningkatan kualitas sumber daya manusia. Salah satu bidang ilmu yang menjadi implementasi penulis dalam menyusun laporan tugas akhir ini adalah bidang hukum.

Kurangnya penghargaan masyarakat terhadap karya orang lain, hal ini dapat dilihat dari banyaknya duplikasi karya tanpa seijin pencipta, sebagai contoh pembajakan karya sastra atau buku, pembajakan terhadap perangkat lunak yang banyak beredar dikalangan masyarakat kita. Atas dasar tersebut maka pemerintah Indonesia telah mengeluarkan Undang-undang no. 19 tahun 2002 tentang Hak Cipta yang mengatur tentang hak cipta atau hasil karya orang lain sebagai suatu hasil karya yang perlu dihargai. Selain hal tersebut undang-undang tersebut juga dimaksudkan sebagai perlindungan atas karya orang lain.

Dari pembahasan pada paragraf sebelumnya, maka penulis mengambil judul, “**Sistem Pakar Pada Undang-Undang Republik Indonesia Nomor 19 Tahun 2002 Tentang Hak Cipta (PRO-ES)**” sebagai judul tugas akhir yang akan dikembangkan oleh penulis agar undang-undang tersebut dapat lebih disosialisasikan kepada masyarakat.

1.1. Latar Belakang Masalah

Banyaknya permasalahan yang berkaitan dengan keilmuan seorang pakar memungkinkan dikembangkannya suatu sistem yang dapat memecahkan masalah. Pengembangan sistem tersebut dimaksudkan untuk membantu seorang pakar dalam menyelesaikan suatu masalah. Adapun latar belakang yang diangkat penulis berkaitan dengan sistem pakar yang dikembangkan adalah bidang hukum pada umumnya, dan Undang-Undang Republik Indonesia nomor 19 tahun 2002 tentang Hak Cipta pada khususnya. Latar belakang dari penyusunan tugas akhir ini adalah sebagai berikut :

- a. Indonesia adalah negara yang memiliki keanekaragaman etnik, suku bangsa dan budaya serta kekayaan dibidang seni dan sastra dengan pengembangan-pengembangannya yang memerlukan perlindungan hak cipta terhadap kekayaan intelektual yang lahir dari keanekaragaman tersebut.
- b. Indonesia telah menjadi anggota berbagai konvensi/perjanjian internasional di bidang hak kekayaan intelektual pada umumnya dan hak cipta pada khususnya yang memerlukan pengejawantahan lebih lanjut dalam sistem hukum nasional.
- c. Perkembangan dibidang perdagangan, industri, dan investasi telah sedemikian besar sehingga memerlukan peningkatan perlindungan bagi pencipta dan pemilik hak terkait dengan tetap memperhatikan kepentingan masyarakat luas.
- d. Keinginan penulis untuk mempelajari dan memperdalam pengetahuan dalam bidang sistem pakar yang merupakan bagian dari intelegensia buatan.

- e. Sosialisasi Undang-undang tentang Hak Cipta kepada masyarakat pada umumnya dan mahasiswa pada khususnya sebagai penghargaan terhadap karya intelektual seseorang.

1.2. Perumusan Masalah

Masalah – masalah yang dihadapi penulis dalam menyusun laporan tugas akhir yang berkaitan dengan permasalahan tentang undang–undang hak cipta tersebut adalah sebagai berikut :

- a. Banyaknya pasal-pasal yang ada dalam undang-undang tersebut, kadangkala menyulitkan para pelaku hukum dalam memilih pasal-pasal yang sesuai dengan kasus yang sedang ditangani.
- b. Penentuan kategori-kategori sanksi yang sesuai dengan pelanggaran yang dilakukan oleh seseorang.
- c. Masalah yang berkaitan dengan pembajakan dan penggandaan karya cipta orang lain adalah merupakan pelanggaran hukum.

1.3. Tujuan

Tujuan dari pembuatan Tugas Akhir Sistem Pakar Pada Undang-Undang Republik Indonesia Nomor 19 Tahun 2002 Tentang Hak Cipta (PRO-ES) ini adalah :

- a. Mensosialisasikan undang-undang Republik Indonesia no.19 tentang hak cipta baik kepada masyarakat luas maupun para mahasiwa.
- b. Memudahkan praktisi hukum untuk mencari pasal-pasal dalam undang-undang Republik Indonesia no.19 tentang hak cipta.
- c. Dapat menentukan sanksi-sanksi dalam undang-undang Republik Indonesia no.19 tentang hak cipta.

1.4. Ruang Lingkup

Penyusunan Laporan Tugas Akhir ini meliputi ruang lingkup permasalahan yang dibatasi dengan batasan sebagai berikut :

- a. Sistem ini dapat mencari pasal-pasal dalam undang-undang Republik Indonesia no.19 tentang hak cipta dengan menggunakan nama pasal, indeks pasal dan isi dari pasal sebagai *keyword*.
- b. Sistem ini dapat memutuskan sanksi-sanksi yang terkait dengan jawaban-jawaban *user* atas pertanyaan yang sistem ajukan.
- c. Sistem ini dikembangkan menggunakan bahasa prosedural.

Dari berbagai macam tipe permasalahan yang dapat diselesaikan oleh sistem pakar, maka Sistem pakar yang dikembangkan penulis merupakan sistem pakar tipe pemilihan (*selection*).

1.5. Metodologi Pengembangan Sistem

Dalam mengembangkan sistem pakar ini penulis menggunakan beberapa tahap pengembangan, yaitu sebagai berikut :

- a. Pernyataan masalah (*problem assessment*)
Dalam tahap ini tugas-tugas yang harus diselesaikan adalah menentukan motivasi dari suatu organisasi dalam mengembangkan sistem, mengidentifikasi permasalahan yang dihadapi
- b. Akuisisi pengetahuan (*knowledge acquisition*)
Pada tahap ini merupakan tahap yang paling kritis , karena dalam tahap ini perlu pengumpulan pengetahuan, analisis dan interpretasi dan perancangan metode-metode untuk pengetahuan yang merupakan opsional dari tahap ini.
- c. Perancangan (*design*)
Pada tahap ini ada beberapa tugas yang harus dilakukan antara lain, pemilihan teknik-teknik yang digunakan untuk merepresentasikan pengetahuan, pemilihan teknik kontrol, pemilihan perangkat lunak yang digunakan untuk mengembangkan sistem, pengembangan *prototype*, pengembangan *interface*, dan pengembangan produk itu sendiri.

d. Pengujian (*testing*)

Dalam tahap ini dilakukan pengujian terhadap sistem pakar yang dikembangkan. Pada sistem pakar ini penulis hanya melakukan pengujian secara implisit.

e. Dokumentasi (*documentation*)

Pada tahap ini dilakukan pendokumentasian atas proses-proses yang telah dilakukan antara lain proses akuisisi pengetahuan, proses perancangan dan proses pengujian sistem pakar tersebut.

f. Pemeliharaan (*maintenance*)

Dalam tahap ini dilakukan pendistribusian (*deploy*) perangkat lunak yang pada bidang tertentu yang berkaitan dengan sistem pakar yang sedang dikembangkan.

1.5. Sistematika Penulisan

Penyusunan laporan tugas akhir ini menggunakan kerangka pembahasan yang terbentuk dalam susunan bab, yang dapat dijelaskan sebagai berikut :

Bab satu pendahuluan, bab ini merupakan dasar penyusunan laporan tugas akhir yang didalamnya berisi tentang latar belakang masalah, perumusan masalah, ruang lingkup, metodologi pengembangan sistem, dan sistematika penulisan.

Bab dua landasan teori, bab ini berisi tentang teori-teori yang dipakai dalam penyusunan tugas akhir ini. Dalam bab ini pembahasan teori meliputi konsep-konsep dasar dalam sistem pakar.

Bab tiga analisis, bab ini berisi tentang pernyataan masalah (*problem assessment*) dan analisis kebutuhan sistem untuk menyelesaikan masalah yang dihadapi berdasarkan landasan teori yang dijelaskan pada bab II

Bab empat perancangan, bab ini berisi tentang perancangan sistem yang merupakan proses kelanjutan dari proses analisis. Bab ini meliputi akuisisi pengetahuan (*knowledge acquisition*).

Bab lima implementasi, bab ini berisi tentang implementasi hasil perancangan pada bab IV dan penyesuaian kebutuhan sistem agar sistem berjalan sesuai dengan rancangan yang dikembangkan penulis.

Bab enam penutup, bab ini berisi tentang kesimpulan dan saran dari hasil penyusunan laporan tugas akhir yang telah disusun.

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem Pakar

Sistem Pakar (*expert system*) merupakan sistem yang menggunakan pengetahuan manusia yang direpresentasikan dalam komputer yang kemudian dipergunakan untuk memecahkan masalah yang biasanya menggunakan kepakaran seseorang.

Kepakaran (*expertise*) merupakan suatu kebiasaan, pekerjaan spesifik yang menggunakan pengetahuan yang didapat dari pelatihan, membaca dan pengalaman. Pengetahuan yang dipergunakan sebagai kepakaran seseorang terdiri dari beberapa tipe, sebagai berikut ;

- a. Pengetahuan tentang kenyataan dalam area permasalahan tertentu.
- b. Pengetahuan tentang teori sebuah permasalahan tertentu.
- c. Pengetahuan tentang aturan dan prosedur untuk area permasalahan secara umum.
- d. Pengetahuan tentang aturan dalam situasi khusus
- e. Pengetahuan tentang strategi global untuk memecahkan semua permasalahan.
- f. *Meta-Knowledge* (pengetahuan tentang pengetahuan).

Pakar (*expert*) merupakan kepakaran manusia yang juga menyangkut akan kebiasaan yang melibatkan aktivitas sebagai berikut :

- a. Menyadari dan memformulasikan permasalahan
- b. Memecahkan masalah secara cepat dan tepat
- c. Menjelaskan solusi
- d. Belajar dari pengalaman
- e. Menstrukturasasi pengetahuan
- f. Melanggar aturan
- g. Mengontrol relevansi
- h. Pengalaman yang kurang menyenangkan

2.1.1 Latar Belakang Sistem Pakar

Sistem pakar mempunyai beberapa latar belakang yang mempengaruhinya. Latar belakang tersebut adalah sebagai berikut :

- a. Para pakar biasanya jumlahnya sedikit dan terkonsentrasi di kota-kota besar oleh karena itu biasanya jasa pakar mahal.
- b. Umur pakar terbatas sehingga jika pakar tersebut tidak tersedia lagi (meninggal) maka kepakarannyapun ikut hilang dan diperlukan waktu bertahun-tahun dengan biaya besar untuk menghasilkan pakar bidang tersebut.

2.1.2 Lingkungan Operasional Sistem Pakar

Lingkungan operasional dalam sistem pakar dibagi menjadi dua :

1. *ON-LINE*

Sistem pakar dapat langsung mengambil data dari sistem yang sedang didiagnosis. Sebagai contoh sistem pakar untuk mendeteksi kerusakan komponen pada mobil, pesawat atau komputer.

2. *OFF-LINE*

Sistem pakar dalam memperoleh informasi dilakukan melalui layar monitor dengan cara melakukan tanya jawab dengan pemakai atau *client*.

2.1.3 Definisi Sistem Pakar

Sistem pakar adalah program komputer yang menggunakan strategi intelegensi buatan seperti representasi simbolik, inferensi, dan pencarian syaraf (Buchanan, 1985) untuk menjalankan tugas yang hanya mungkin dilakukan oleh seorang pakar.

2.1.4 *Typology* Aplikasi Sistem Pakar

Menurut Hayes-Roth, Waterman, dan Lenat (1983:14) *typology* dari aplikasi sistem pakar adalah sebagai berikut :

- 1) *Interpretation* - menyimpulkan penjelasan dari situasi atau kondisi yang diperoleh dari sensor data.
- 2) *Prediction* – menyimpulkan konsekwensi dari situasi yang diberikan
- 3) *Diagnosis* – dugaan terhadap kesalahan (*malfuction*) dari segala sesuatu yang terlihat atau nampak.

- 4) *Design* – konfigurasi objek-objek dalam batasan tertentu
- 5) *Planning* – perancangan aksi-aksi.
- 6) *Monitoring* – membandingkan pengamatan atas segala sesuatu yang mudah terjadi kesalahan.
- 7) *Instruction* – diagnosis, membenaran kesalahan dan perbaikan perilaku murid (*student*)
- 8) *Control* – interpretasi, prediksi, perbaikan dan monitoring perilaku sistem.
- 9) *Prescription* – solusi yang direkomendasikan untuk sistem yang telah malakukan kesalahan (*malfunction*)
- 10) *Selection* – mengidentifikasi pilihan terbaik dari beberapa daftar yang memungkinkan.
- 11) *Simulation* – pemodelan interaktif antar komponen sistem

2.1.5 Contoh Klasik dari Pakar Klasik

Ada beberapa contoh sistem pakar klasik sebagai berikut :

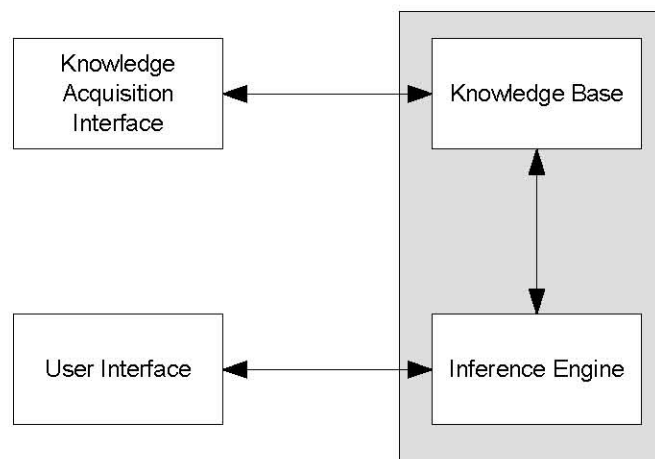
- a) MYCIN – diagnosa radang pembuluh darah.
- b) CADUCEUS (atau disebut INTERNIST) – diagnosis medis dalam obat-obatan tertentu.
- c) AI-RHEUM – Diagnosa penyakit rematik
- d) PROSPECTOR – Identifikasi persediaan air
- e) MACSYMA – Perhitungan Aljabar secara simbolik
- f) EMYCIN - "Empty" MYCIN (awalnya shell sistem pakar berbasis rule)
- g) R1 – Konfigurasi sistem komputer DEC VAX .
- h) HEARSAY-II – Pengenalan suara
- i) DIPMETER ADVISOR – mendiagnosa contoh tanah yang telah didapat dari hasil pengeboran.
- j) ELECTROPHORESIS INTERPRETER – Sistem Pakar pertama yang ada dalam sebuah *chip*

2.1.6 Contoh Kontemporer dari Sistem Pakar

- a) EX-SAMPLE – Menentukan ukuran sampel (perhitungan statistik).
- b) PEER REVIEW EMULATOR – Perancangan proposal riset (*semantic knowledge, forms*)
- c) STATISTICAL NAVIGATOR – Pengidentifikasian statistik-statistik yang sesuai (*decision making*)
- d) WHICHGRAPH – Pengidentifikasian gambar-gambar (graphs) yang sesuai.
- e) DESIGNER RESEARCH – Perancangan percobaan (design)
- f) HYPER-SOC – sistem pembelajaran pengenalan sosiologi (*simulations, modeling students*)
- g) MAKING IT – sistem pakar terkait (*embedded expert system*) untuk menciptakan aktor-aktor yang cerdas dan antar-muka bagi pendijitan video.

2.2. Arsitektur Sistem Pakar

Pada dasarnya sistem pakar terdiri dari empat komponen utama. Adapun komponen-komponen tersebut adalah sebagai berikut :



Gambar 2.1 Arsitektur Sistem Pakar

2.2.1 *Knowledge Acquisition Interface* (antar-muka akuisisi pengetahuan)

Knowledge Acquisition Interface mengontrol bagaimana seorang pakar dan rekayasa pengetahuan (*knowledge engineer*) berinteraksi dengan program untuk memasukkan pengetahuan ke dalam basis pengetahuan (*knowledge base*). Termasuk hal-hal yang dimaksudkan untuk membantu para pakar dalam mengekspresikan pengetahuannya dalam bentuk yang sesuai yang bisa diterima oleh komputer.

Proses ini mengekspresikan pengetahuan dalam basis pengetahuan yang biasa disebut dengan akuisisi pengetahuan (*knowledge acquisition*). Akuisisi pengetahuan akan menjadi sangat sulit, para penulis menyebut sebagai “*knowledge acquisition bottleneck* “. Karena aspek dari pengembangan sistem pakar ini membutuhkan waktu yang cukup banyak dan usaha yang keras.

2.2.2 *User Interface*

User interface adalah bagian dari program yang berinteraksi dengan pengguna. Prompt bagi pengguna digunakan untuk informasi yang diperlukan untuk memecahkan masalah, menampilkan kesimpulan dan menjelaskan alasannya.

Fasilitas dari user interface kerap kali memasukkan :

- Tidak menanyakan pertanyaan yang bersifat basa-basi
- Menjelaskan alasan atas pertanyaan yang diajukan
- Menyediakan dokumentasi dan referensi
- Mendefinisikan istilah-istilah yang bersifat teknik
- Laporan rekomendasi yang cukup detail
- Pembeneran rekomendasi

2.2.3 *Knowledge Base* (basis pengetahuan)

Knowledge Base terdiri dari detail pengetahuan tentang domain tertentu. Basis pengetahuan sangat beda dengan basis data yang mana basis pengetahuannya terdiri dari pengetahuan eksplisit (*explicit knowledge*) dan pengetahuan implisit (*implicit knowledge*). Banyak pengetahuan dalam basis pengetahuan tidak dalam kondisi *eksplisit*, tetapi diproses dengan mesin *inferensi* (*inference engine*) dari pernyataan *eksplisit* ke dalam basis pengetahuan. Hal ini

membuat basis pengetahuan lebih efisien dalam penyimpanan data bila dibandingkan dengan basis data.

Basis pengetahuan dapat terdiri dari beberapa tipe pengetahuan, dan proses mendapatkan pengetahuan untuk basis pengetahuan kerap kali berbeda antara yang satu dengan yang lainnya. Hal ini tergantung dari tipe pengetahuan. Ada beberapa jenis pengetahuan yang perlu dipertimbangkan dalam sistem pakar yang antara lain sebagai berikut :

1. *explicit knowledge*
2. *domain knowledge*
3. *heuristics*
4. *procedural knowledge*
5. *public knowledge*
6. *shallow knowledge*
7. *meta-knowledge*
8. *implicit knowledge*
9. *common sense atau world knowledge*
10. *algorithms*
11. *declarative atau semantic knowledge*
12. *provate knowledge*
13. *deep knowledge*

2.2.4 Inference Engine

Inference engine secara umum memproses *rule* untuk memberikan alasan dan menggambarkan kesimpulan-kesimpulan yang tidak dapat di lihat tetapi dapat di proses dari basis pengetahuan.

Mesin *inferensi* dapat dikatakan sama dengan *query* dalam sistem basis data. Tugas dari modul *inferensi* adalah mengeksplorasi basis pengetahuan untuk mencari solusi dari permasalahan tertentu.

Inference engine mampu memberikan alasan simbolik, bukan hanya alasan yang bersifat matematik. Bentuk *inference engine* diperkenankan untuk dikembangkan dalam berbagai variasi , tergantung dari berbagai faktor, termasuk bagaimana cara merepresentasikan pengetahuan.

2.3. Karakteristik Sistem Pakar

Ada beberapa karakteristik sistem pakar sebagai berikut :

- 1) Sistem pakar harus terspesialisasi layaknya seorang pakar
Hal ini berarti bahwa kemampuan sistem pakar diperoleh dari basis pengetahuan tertentu dalam domain yang spesifik dalam bidang yang spesifik juga, ini bukan berarti sistem pakar yang mampu menjadi pakar dalam segala hal.
- 2) *Domain knowledge* harus dipisah dari kode program agar dapat dengan mudah dimengerti.
Karena basis pengetahuan terdiri dari domain pengetahuan yang spesifik maka pengetahuan tersebut dapat dipisahkan dari mesin *inferensi* yang memungkinkan representasi domain pengetahuan tertentu. Dengan demikian seorang pakar hanya perlu mengekspresikan pengetahuan dengan pernyataan (*English Statement*) seperti rule yang merupakan cara merepresentasikan pengetahuan ke dalam kode program (*source code*)
- 3) Sangat dimungkinkan menggunakan shell sistem pakar untuk tujuan yang bersifat umum.
Sejak basis pengetahuan dapat dipisahkan dari program hal ini memungkinkan pengembangan program dengan tiga komponen dimana *non programmer* dapat menyelesaikan aplikasi sistem pakar. Program ini disebut dengan shell sistem pakar (*expert system shell*).
- 4) Sistem pakar mereduksi kebutuhan akan keahlian pemrograman (*Programming skill*).
Sistem pakar memisahkan kode program dengan *knowledge base*, ini memperbolehkan basis pengetahuan diekspresikan dalam sintaks yang mudah dimengerti oleh *non-programmer* atau orang yang mempunyai sedikit keahlian yang berhubungan dengan pemrograman. *Non-programmer* dapat mengerjakan aplikasi sistem pakar dengan *expert system shell* sebagai alat pengembang (*development tools*).
- 5) Mesin inferensi yang sama dapat digunakan pada basis pengetahuan yang berbeda.

Sebagai contoh, mesin inferensi yang dikembangkan MYCIN yang telah dikonversi menjadi EMYCIN (untuk “empty” MYCIN) membuat sistem pakar secara umum dapat diaplikasikan kepada sejumlah masalah yang berbeda.

- 6) Mesin inferensi yang berbeda dapat diterapkan pada basis pengetahuan yang sama.

Karena sistem pakar yang didasarkan pada *symbolic reasoning* sama baiknya dengan *numeric reasoning*, maka sistem pakar dapat dikembangkan untuk menyelesaikan tugas-tugas yang bersifat non-numeric. Jadi hal ini memungkinkan pengaplikasian program sistem pakar pada masalah-masalah dimana *formal algorithmic* tidak sesuai untuk menyelesaikan masalah tersebut. Contoh dari sistem pakar dapat diimplementasikan dalam diagnosis penyakit, konfigurasi komputer, prediksi perilaku sosial, dan beberapa masalah yang tidak dapat diselesaikan oleh *formal numeric solution*.

- 7) Sistem pakar tidak menjamin menjadi benar dan kadang-kadang gagal. Karakteristik ini tergantung dari representasi pengetahuan yang dilakukan oleh *knowledge engineering* maupun seorang pakar. Atau ada kemungkinan dari sumber yang tidak dapat dipercaya.

2.4. Sistem Pakar dan Bentuk Pengetahuan Lain

Cara lain untuk mengerti kemampuan dan keterbatasan sistem pakar dengan cara membandingkannya dengan bentuk pengetahuan yang lebih umum. Ada beberapa cara penting yang membedakan sistem pakar dengan seorang pakar, buku, dan bentuk pengetahuan lainnya.

- a) Sistem Pakar dengan Buku

Tabel 2.1 Perbandingan Sistem Pakar dengan Buku

PERSAMAAN
Keduanya memberikan kepada pengguna informasi yang cukup banyak
PERBEDAAN
<ul style="list-style-type: none"> - Sistem pakar dapat memberikan nasehat pada pengguna tertentu. - Dengan buku pembaca mungkin mengabaikan hal penting sehingga memungkinkan terjadi salah pengertian.

- b) Sistem Pakar dengan seorang pakar

Tabel 2.2 Perbandingan Sistem Pakar dengan Seorang Pakar

PERSAMAAN
<p>Keduanya memberikan pertanyaan kepada user. Keduanya merekomendasikan aksi berdasarkan jawaban user. Keduanya memberikan nasihat secara personalitas.</p>
PERBEDAAN
<ul style="list-style-type: none"> - Sistem pakar selalu bekerja tidak pernah bosan, tidak pernah lelah, tidak pernah malas berpikir, tidak pernah marah dan lain-lain. - Sistem Pakar dapat diduplikasi dan disebarluaskan. - Sistem Pakar Tidak pernah mengenal waktu istirahat. - Sistem Pakar tidak diam dan mati

2.5. Representasi Pengetahuan

Sebelum menjelaskan bagaimana cara merepresentasikan pengetahuan terlebih dahulu harus diketahui tipe-tipe pengetahuan karena ini akan mempengaruhi dalam penggunaan teknik representasi pengetahuan. Tipe dari pengetahuan menurut John Durkin dibedakan menjadi lima pengetahuan, yaitu :

Tabel 2.3 Tipe Pengetahuan

Nama		Terdiri dari	Keterangan
1	<i>Procedural Knowledge</i>	<i>Rule</i> <i>Strategy</i> <i>Agenda</i> <i>Procedure</i>	pengetahuan ini menguraikan bagaimana masalah dapat diselesaikan. Tipe ini mempunyai tujuan bagaimana melakukan sesuatu. <i>Rule, Strategy, agenda</i> dan prosedur adalah tipe <i>procedural knowledge</i> yang digunakan dalam sistem pakar ini.
2	<i>Declarative Knowledge</i>	<i>Concept</i> <i>Object</i> <i>Fact</i>	pengetahuan ini menjelaskan masalah apa yang diketahui. Ini termasuk pernyataan yang sederhana, dimana dijabarkan dalam bentuk salah (<i>false</i>) atau benar (<i>true</i>).
3	<i>Meta Knowledge</i>	Pengetahuan tentang tipe pengetahuan lain.	pengetahuan yang digunakan untuk pengetahuan lain. Tipe pengetahuan ini digunakan untuk mengambil pengetahuan lain untuk menyelesaikan suatu masalah.
4	<i>Heuristic Knowledge</i>	<i>Rules of Thumb</i> atau aturan ibu jari	pengetahuan ini sering disebut <i>shallow knowledge</i> ketepatan dan representasi pengetahuan dikompilasi oleh seorang pakar melalui pengalaman yang telah dialami oleh pakar tersebut dimasa lalu
5	<i>Structural Knowledge</i>	<i>Rule Set</i> Konsep <i>Relationship</i> Konsep <i>Object</i> <i>Relationship</i>	Pengetahuan ini menguraikan struktur pengetahuan. Tipe pengetahuan ini juga memodelkan masalah yang dihadapi oleh seorang pakar. Model dari konsep, subkonsep, dan objek adalah tip dari pengetahuan ini.

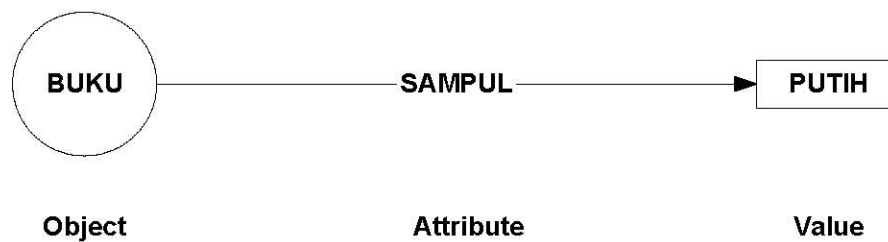
Representasi Pengetahuan adalah metode yang digunakan untuk menggabungkan pengetahuan dalam basis pengetahuan dari sistem pakar.

Teknik-teknik yang digunakan untuk representasi pengetahuan antara lain :

1. *Object-Attribute-Value triplets*

Didalam sistem pakar *fact* (fakta) digunakan untuk membantu menguraikan bagian-bagian dari *frame, semantic network* atau *rule*. Didalam AI (*Aritifical Intelegent*) dan sistem pakar , sebuah *fact* sering disebut sebagai

proposition. Proposition adalah sebuah pernyataan yang salah satunya *true* atau *false*. Sebuah *fact* mungkin juga digunakan untuk menyatakan nilai dari properti tertentu dari beberapa objek. Sebagai contoh, pernyataan “Buku bersampul putih” memberi pengertian bahwa nilai “putih” untuk sampul buku. *Fact* tipe ini dikenal dengan *object-attribute-value* (O-A-V) *triplet*. Berikut ini merupakan gambaran dari contoh diatas.



Gambar 2.2 Object-Attribute-Value

2. Rule

Fact (fakta) yang ditulis oleh user merupakan operasi yang sangat penting bagi sistem pakar. Operasi tersebut tersebut memungkinkan sistem untuk mengetahui kondisi yang sedang terjadi. Oleh Karena itu sistem harus mempunyai pengetahuan tambahan agar dapat bekerja secara *intelligent* dengan fakta yang ada untuk menyelesaikan suatu masalah. Sebuah struktur pengetahuan yang digunakan secara umum dalam sistem pakar dengan menyediakan pengetahuan tambahan tersebut, disebut juga dengan *rule*.

Sebuah *rule* adalah bentuk dari *procedural knowledge* yang diasosiasikan untuk memberikan informasi bagi beberapa aksi. Struktur *rule* secara logik menghubungkan satu atau beberapa alasan (*premise*) yang terdapat dalam pernyataan JIKA (*IF*) dengan satu atau lebih kesimpulan (*conclusion*) yang terdapat dalam pernyataan MAKA (*THEN*).

Sebagai contoh,

JIKA buku berwarna merah
MAKA saya menyukai buku

Secara umum *rule* dapat mempunyai beberapa alasan yang digabungkan dengan pernyataan AND (*conjunction*), atau pernyataan OR (*disjunction*) atau dapat juga menggabungkan keduanya.

Rule dapat direpresentasikan kedalam berbagai bentuk pengetahuan. Ada beberapa tipe *rule* yang diuraikan oleh John Durkin sebagai berikut

a) *Relationship*

Contoh : *Jika saya mengantuk*
 Maka saya akan tidur

b) *Recommendation*

Contoh : *Jika saya tidak bisa tidur*
 Maka saya minum obat tidur

c) *Directive*

Contoh : *Jika saya tidak bisa tidur*
 Dan saya tidak sakit
 Maka konsultasi ke psikolog

d) *Strategy*

Contoh : *Jika saya tidak bisa tidur*
 Maka cobalah membaca buku

e) *Heuristic*

Contoh : *Jika tidak bisa tidur*
 Dan badan terasa capai
 Maka menderita penyakit insomnia

3. *Frame*

Frame merupakan sebuah struktur untuk merepresentasikan pengetahuan dari berbagai konsep atau objek. Sebuah frame juga dapat digambarkan dalam berbagai bentuk seperti yang terlihat pada gambar berikut :

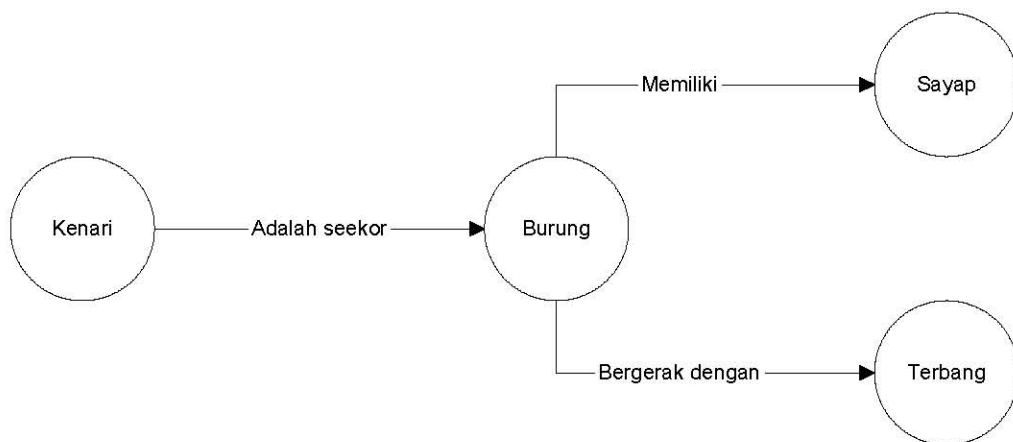


Gambar 2.3 Contoh untuk Frame Burung

4. *Semantic Network*

Semantic network merupakan salah satu teknik untuk menggambarkan hubungan antar object yang terbentuk melalui node-node dan links.

Contoh representasi pengetahuan melalui semantic network dapat dilihat pada gambar berikut.



Gambar 2.4 Semantic Network dari burung

Ada beberapa keunggulan yang dimiliki oleh *semantic network* :

- Kemudahan dalam penambahan *node* dan *link*, serta mudah dimengerti.
- Lebih detail dalam menggambarkan suatu *object*.
- Hampir sama dengan penyimpanan informasi.
- Dapat membuat alasan dan membuat definisi pernyataan antara *link* yang tidak terhubung.

5. *Logic*

Bentuk pengetahuan yang paling tua dalam merepresentasikan pengetahuan dalam komputer adalah *logic*. Dalam *logic* sendiri ada dua teknik yang sering digunakan dalam representasi pengetahuan yaitu *Propositional Logic* dan *Predicate Calculus*.

a) *Propositional Logic*

Propositional logic merepresentasikan dan memberi alasan dengan dalil (*proposition*), dimana salah satu pernyataannya adalah *true* atau *false*.

b) *Predicate Calculus*

Predicate Calculus membagi pernyataan menjadi beberapa bagian, pemberian nama, karakteristik objek, keterangan lain tentang objek

2.6. Teknik Inferensi

Pada pembahasan ini perlu dipertimbangkan bagaimana sistem memberi alasan (*reason*) berhubungan dengan pengetahuan yang diperlukan untuk menyelesaikan masalah. *Reasoning* di dalam sistem pakar dilakukan dengan menggunakan teknik inferensi (*inference techniques*) dan strategi kendali (*control strategies*). Teknik inferensi memandu sistem agar dapat mengkombinasikan pengetahuan yang ada dalam basis pengetahuan dengan fakta permasalahan yang ada di memori kerja (*working memory*).

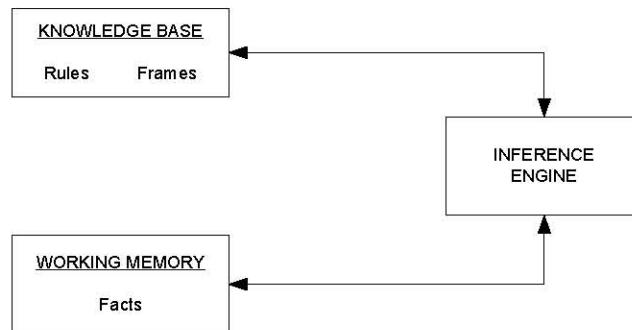
2.6.1 *Reasoning*

Reasoning adalah proses kerja untuk menggambarkan suatu kesimpulan dengan strategi-strategi untuk menyelesaikan masalah, fakta-fakta dan pengetahuan. Ada beberapa macam *reasoning*, yang antara lain :

- a) *Deductive Reasoning*
- b) *Inductive Reasoning*
- c) *Abductive Reasoning*
- d) *Analogical Reasoning*
- e) *Common-Sense Reasoning*
- f) *Non-Monotonic Reasoning*

2.6.2 Inference

Inference digunakan dalam sistem pakar untuk memperoleh informasi terbaru dari informasi yang sudah ada. Sistem pakar dalam menjalankan *inference* (pemrosesan) menggunakan modul yang disebut dengan *inference engine*. Pengolahan Inferensi dalam sistem pakar dapat dilihat pada gambar berikut ini.



Gambar 2.5 Proses Inferensi dalam Sistem Pakar

2.6.3 Forward-Chaining

Forward-Chaining adalah strategi inferensi yang dimulai dengan sekumpulan fakta, fakta baru yang diperoleh dengan menggunakan *rule* dimana alasan yang digunakan sesuai dengan fakta yang ada, dan melanjutkan proses ini sampai gol diraih atau sampai tidak ada *rule* selanjutnya yang mempunyai alasan yang sesuai dengan fakta yang ada maupun fakta yang diketahui.

Operasi dari sebuah *forward-chaining* system dimulai dengan inisialisasi tentang masalah yang dinyatakan dalam *working memory*. Hal ini dapat dibangun dengan sejumlah cara, seperti informasi yang diperoleh dari basis data, sensor atau menanyakan kepada *user*.

Sementara metodologi yang digunakan untuk merancang sistem *forward-chaining* memiliki beberapa tugas utama yaitu,

- 1) Mendefinisikan masalah (*define the problem*)
- 2) Mendefinisikan data input (*define input data*)
- 3) Mendefinisikan struktur data-driven (*define data-driven struktur*)
- 4) Menulis program utama (*write initial code*)
- 5) Menguji sistem (*testing the system*)
- 6) Merancang antar-muka (*design the interface*)

7) Mengembangkan sistem (*expand the system*)

8) Evaluasi sistem (*evaluate system*)

2.6.4 *Backward-Chaining*

Backward-Chaining adalah strategi inferensi yang diperoleh untuk membuktikan suatu hipotesis dengan dukungan informasi. *Backward-Chaining* dimulai dengan sebuah gol yang harus dibuktikan. Langkah dalam membuktikan suatu gol adalah sebagai berikut :

Pertama, periksa *working memory* untuk melihat jika gol telah ditambahkan sebelumnya. Langkah ini diperlukan karena mungkin basis pengetahuan lain sudah siap membuktikan suatu gol.

Kedua, jika sebuah gol tidak dibuktikan sebelumnya, maka sistem akan mencari rule satu per satu dalam bagian THEN. Tipe rule ini disebut dengan gol rule. Selanjutnya sistem akan memeriksa jika alasan gol tercantum dalam *working memory*. Alasan yang tidak tercantum akan menjadi gol baru yang disebut dengan subgol.

Ada tujuh tugas utama yang harus diterapkan ketika mengembangkan sistem *backward-chaining*, yaitu :

1) Mendefinisikan masalah (*define the problem*)

Selama pendefinisian masalah , jangan meminta seorang pakar untuk menanyakan detail permasalahan. Carilah pengertian dari suatu masalah dan solusinya secara umum. Untuk detail masalah dan solusi dapat diperoleh dari seorang pakar, laporan-laporan, dokumen-dokumen dan buku-buku yang merupakan sumber informasi bagi semua proyek sistem pakar.

2) Mendefinisikan gol/tujuan (*define the goals*)

Dalam menentukan gol suatu masalah harus difokuskan pada perancangan gol secara sempit, kemudian perluas gol untuk memudahkan pengembangan sistem.

3) Merancang aturan gol (*design the goal rules*)

Perancangan gol *rule* memiliki cara yang sama dengan tipe *rule* lainnya. Bentuk umum dari gol *rule* dapat dilihat seperti,

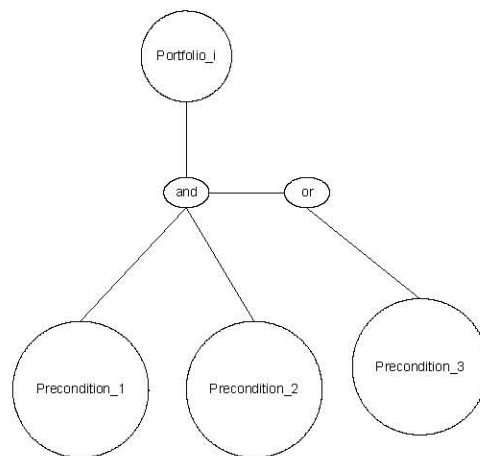
IF *Precondition_1*
 AND *Precondition_2*
 - -
 - -
 THEN *Portfolio_i*

Atau dapat juga diuraikan dengan teknik akuisisi pengetahuan dengan menggunakan tabel keputusan (*decision tables*). Decision table berisi urutan faktor-faktor keputusan dengan kolom label (*label column*) dan menggambarkan *precondition* yang diperlukan untuk memperoleh suatu kesimpulan yang direpresentasikan pada kolom lain. Nilai suatu faktor keputusan (*decision tables*) ditempatkan pada baris (*row*) yang mempunyai detail keputusan. Berikut contoh *decision tables*.

Tabel 2.4 Contoh Decision Tables

<i>Decision Factors</i>		<i>Result</i>
<u>Uang</u>	<u>Usia</u>	<u>Hadiah</u>
Banyak	Dewasa	Mobil
Banyak	Anak-anak	Komputer
Sedikit	Dewasa	Rumah
Sedikit	Anak-Anak	Kalkulator

Selain itu ada juga bentuk lain dari representasi gol rule yaitu *inference network*. *Inference network* menunjukkan hubungan setiap bagian informasi secara logik. Berikut ini contoh *inference network*.



Gambar 2.6 Contoh Inference Network

4) Mengembangkan sistem (*expand the system*)

Pengembangan sistem dilakukan sekali dalam waktu tertentu. Ini memungkinkan seorang pakar berkonsentrasi pada satu masalah dan menghindari sejumlah permasalahan yang muncul .

5) Memperhalus sistem (*refine the sistem*)

Makna dari memperhalus sistem disini adalah mempermudah penulisan program dengan menggunakan pendekatan tertentu. Pendekatan tersebut antara lain pendekatan menggunakan variabel bagi angka.

Sebagai contoh : *Inisialisasi Usia=40*

Rule 1

IF umur_client < usia

THEN client masih muda

6) Merancang antarmuka (*design the interface*)

Dalam pecancangan antarmuka sistem harus memiliki tampilan (*display*) yang terdiri dari :

1. Tampilan Perkenalan (*introductory display*)
2. Tampilan Pertanyaan (*questions display*)
3. Layar direksi (*screen directions*)

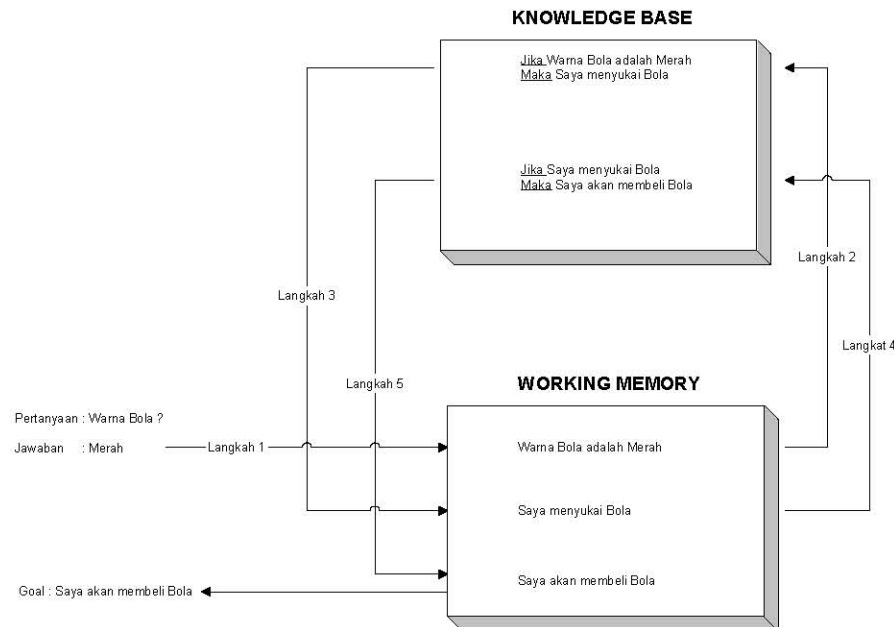
Dalam *screen directions* harus ada kontrol yang memandu sistem untuk : melanjutkan sesi, meminta pertanyaan dengan banyak jawaban, dan mengakhiri sesi.

4. Tampilan Kesimpulan (*conclusion display*)

7) Evaluasi sistem (*evaluate the system*)

Evaluasi sistem dilakukan untuk melengkapi sebuah *prototype*. Semua rule di-*coding*-kan kedalam sistem dan antarmuka dirancang untuk merekomendasikan sesi sebelumnya.

Untuk memperjelas gambaran tentang *backward chaining* dapat digambarkan dengan ilustrasi sebagai berikut :



Gambar 2.7 Backward Chaining Inference dalam Rule-Based System

2.7. Metode Pengembangan Sistem Pakar

2.7.1 Metode Rekayasa Perangkat Lunak dengan Model *Waterfall*

Pengembangan sebuah sistem informasi berbasis komputer dapat menggunakan beberapa metode sebagai acuan. Setiap metode akan dibagi menjadi tahapan-tahapan yang akan memudahkan dalam pembangunan sistem informasi. Metode yang sering juga disebut metode "*waterfall*" atau "*classic life cycle*" ini menggunakan pendekatan yang sistematis dan sekuensial dalam membangun perangkat lunak yang dimulai pada level sistem dan pengembangan melalui tahapan analisis, perancangan, pengkodean, pengujian, dan pemeliharaan.

a. Analisis

Dari rumusan sistem yang diperoleh dari tahap pertama, selanjutnya dilakukan analisis yang berkaitan dengan proses dan data yang diperlukan oleh sistem serta keterkaitannya. Tujuan dilakukan tahapan ini adalah untuk memahami sistem yang ada pada saat ini agar dapat mendefinisikan permasalahan sistem sehingga selanjutnya dapat menentukan kebutuhan sistem secara garis besar

sebagai persiapan ke tahap perancangan. Analisis di sini dilakukan dengan pemodelan menggunakan metode *Data Flow Oriented* dengan tool *Data Flow Diagram* (DFD).

b. Perancangan

Pada tahap perancangan ini diberikan gambaran umum yang jelas kepada pengguna dan rancang bangun yang lengkap tentang sistem yang akan dikembangkan kepada pihak-pihak yang terlibat dalam pengembangan sistem. Tahap perancangan ini dilakukan sebagai persiapan untuk tahap implementasi.

c. Implementasi/*Coding*/Pemrograman

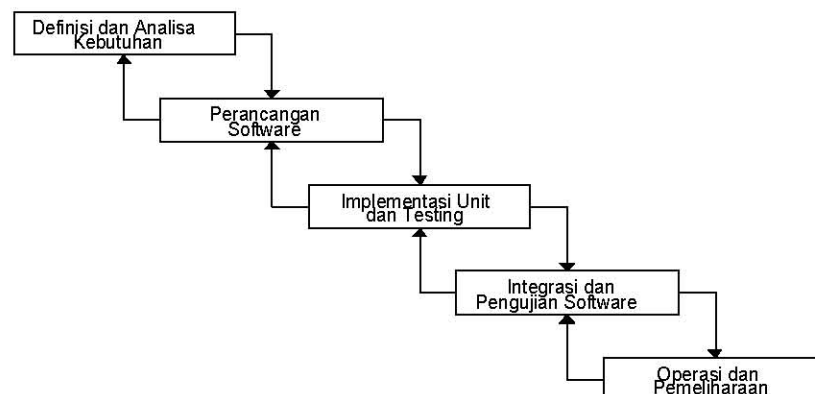
Setelah tahap perancangan sistem, selanjutnya dilakukan konversi rancangan sistem ke dalam kode-kode bahasa pemrograman yang diinginkan. Pada tahap ini dilakukan pembuatan komponen-komponen sistem yang meliputi modul program, antarmuka dan basis data.

d. Pengujian

Tahap pengujian ini dilakukan untuk mendapatkan serta memastikan bahwa perangkat lunak yang dihasilkan adalah valid dan sesuai dengan kebutuhan yang telah dideskripsikan.

d. Pemeliharaan

Pada tahap pemeliharaan ini perangkat lunak sudah diserahkan kepada pengguna. Pada tahap ini dilakukan evaluasi terhadap sistem yang baru untuk mengetahui apakah sistem telah memenuhi tujuan yang ingin dicapai. Dari hasil evaluasi ini dimungkinkan untuk melakukan perubahan-perubahan yang diperlukan terhadap sistem agar sistem senantiasa dapat digunakan dengan baik

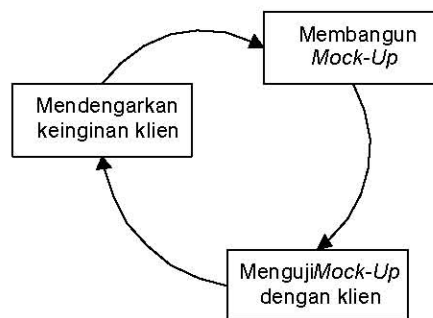


Gambar 2.8 Metode RPL dengan Metode *Waterfall*

2.7.2 Metode Rekayasa Perangkat Lunak dengan Model Prototipe (*Prototyping Model*)

Model ini bertujuan untuk membuat *prototype* dari perangkat lunak yang akan dikembangkan. Kelemahan model ini adalah :

1. *User* cenderung menganggap bahwa *prototype* tersebut adalah produk yang hampir jadi dan hanya memerlukan sedikit perbaikan.
2. Pengembang seringkali melakukan kerja sama untuk membuat *prototype* cepat berfungsi, sehingga implementasi dilakukan dengan kurang mempertimbangkan aspek jangka panjang sistem yang dikembangkan.

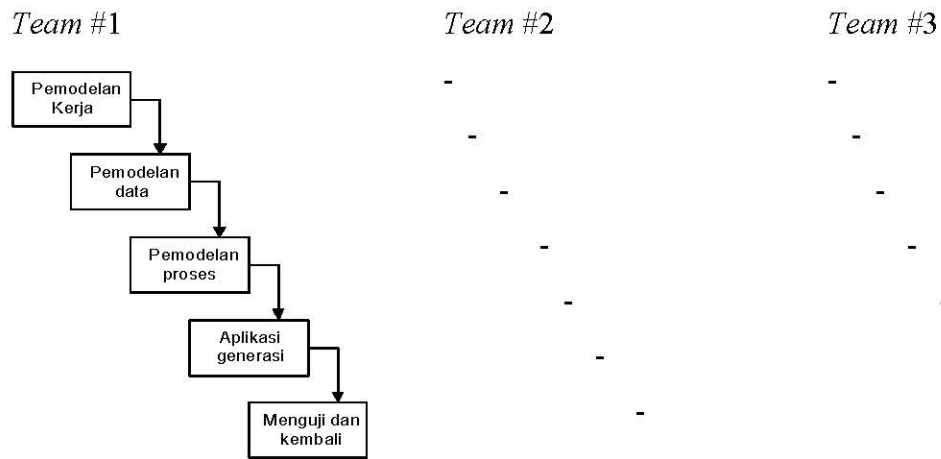


Gambar 2.9 Metode RPL dengan Model Prototipe

2.7.3 Metode Rekayasa Perangkat Lunak dengan Model RAD (*Rapid Application Development*)

Merupakan model pengembangan perangkat lunak secara sekuensial dengan siklus pengembangan yang sangat singkat. Kelemahan model ini adalah :

1. Untuk proyek dengan skala besar, diperlukan jumlah SDM yang memadai.
2. Pengembang dan *user* harus memiliki komitmen waktu yang tinggi untuk dapat melaksanakan model ini dengan cepat.
3. Aplikasi tertentu tidak cocok dikembangkan dengan model ini, seperti aplikasi-aplikasi untuk sistem yang tidak modular, sistem yang memerlukan kinerja tinggi, dan sistem dengan teknologi baru yang memiliki resiko bisnis lebih besar.

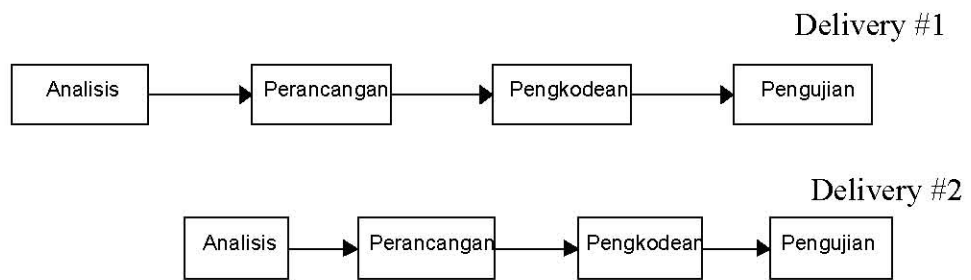


Gambar 2.10 Metode RPL dengan Model RAD

2.7.4 Metode Rekayasa Perangkat Lunak dengan Model Inkremental (Incremental Model)

Model ini menggabungkan elemen dari model linier sekuensial dengan filosofi pengulangan pada model *prototype*. Model ini terdiri dari beberapa linier sekuensial yang menghasilkan inkremen perangkat lunak.

Model inkremental cocok diterapkan pada proyek yang memiliki keterbatasan SDM. Selain itu model ini juga baik untuk meminimalkan resiko teknis yang mungkin terjadi.



Gambar 2.11 Metode RPL dengan Model Inkremental

2.7.5 Metode Rekayasa Perangkat Lunak dengan Model Spiral (*Spiral Model*)

Model ini menggabungkan aspek pengulangan pada model prototipe dan aspek sistematis dari model linier sekuensial. Pada model ini, perangkat lunak dikembangkan melalui serangkaian rilis yang meningkat.

Model spiral dapat digunakan sepanjang hidup *software*, artinya untuk perubahan yang mungkin terjadi setelah *software* diimplementasikan dapat dilakukan dengan menggunakan model ini. Model ini sangat realistis untuk diterapkan dalam pengembangan software skala besar. Disamping itu, model ini memiliki mekanisme untuk meminimalkan resiko. Namun pengurangan resiko ini tentu saja sangat bergantung pada keahlian personil yang melakukan penilaian resiko tersebut.

2.7.6 Metode Rekayasa Perangkat Lunak dengan Model Perakitan Komponen (*Component Assembly Model*)

Model ini dikembangkan dalam rangka teknologi berorientasi objek, dimana kelas yang ada merupakan komponen yang dapat digunakan untuk berbagai aplikasi yang berbeda. Model perakitan komponen ini mirip dengan model spiral, dimana perbedaannya hanya pada tahap konstruksi perangkat lunak.

2.7.7 Metode Rekayasa Perangkat Lunak dengan Model Pengembangan Konkuren (*Concurrent Development Model*)

Model pengembangan konkuren memfokuskan pada keadaan dari suatu proyek dimana beberapa aktivitas pada tahap yang berbeda dapat terjadi secara konkuren. Model ini direpresentasikan sebagai serangkaian aktivitas teknis, tugas dan keadaan dari aktifitas tersebut. Maka dengan demikian semua aktivitas ada secara bersamaan/konkuren, tetapi dalam keadaan yang berbeda. Model ini mendefinisikan serangkaian kejadian (*event*) yang dapat merubah suatu aktivitas dari suatu keadaan ke keadaan yang lain.

2.7.8 Metode Rekayasa Perangkat Lunak dengan Model Metode Formal (*Formal Methods Model*)

Model metode formal menekankan kegiatan pengembangan *software* (spesifikasi, pengembangan, dan verifikasi) menggunakan notasi matematis. Penggunaan model ini menjanjikan suatu produk *software* yang bebas dari cacat. Namun demikian, masih terdapat hambatan dalam pembangunan model ini dalam lingkungan bisnis, antara lain karena :

1. Biaya pengembangan model formal sangat lama dan mahal.
2. Diperlukan pelatihan bagi pengembang *software* untuk dapat menggunakan model ini.
3. Sulit digunakan sebagai mekanisme komunikasi dengan user yang awam.

2.8. Perangkat Lunak (*Software*)

Perangkat lunak (*software*) dapat dikategorikan ke dalam 3 bagian, yaitu [Jog H.M.'88] :

1. Perangkat Lunak Sistem Operasi (*Operating System*), yaitu program yang ditulis untuk mengendalikan dan mengkoordinasi kegiatan dari sistem komputer.
2. Perangkat Lunak Bahasa (*Language Software*), yaitu program yang digunakan untuk menterjemahkan instruksi-instruksi yang ditulis dalam bahasa pemrograman ke dalam bahasa mesin supaya dapat dimengerti oleh komputer.
3. Perangkat Lunak Aplikasi (*Application Software*), yaitu program yang ditulis dan diterjemahkan oleh *language software* untuk menyelesaikan suatu aplikasi tertentu.

2.8.1 *Software* Sistem

Software sistem merupakan *background* program yang memungkinkan *software* aplikasi dapat berfungsi pada peralatan *hardware* sistem komputer [Charles S.P.,'96].

Software sistem dapat dibedakan menjadi 3 bagian, yaitu [Charles S.P.,'96] :

- a. *Operating System*, adalah kumpulan utama dari program yang mengatur aktivitas sistem komputer. Contoh : *Windows '98, Macintosh System Software, MS-DOS, UNIX, dan Penpoint.*

Windows '98

Windows '98 dilengkapi fasilitas-fasilitas yang dapat mempermudah kerja user, yaitu :

- *Built-in Networking*, fasilitas jaringan terintegrasi di sistem operasi.
- Pengamanan level pemakai *pass-through*.
- *Registry*, semua informasi mengenai sistem disatukan sehingga mempermudah pengelolaan sistem.

Pada *Windows '98* antarmuka dengan sistem dilakukan sebagai berikut :

- Kebanyakan dilakukan melalui fasilitas berbasis grafik : *menu, icon.*
- Tersedia *Task bar* yang mempermudah perpindahan aplikasi yang saat itu dijalankan sistem operasi.
- Terdapat jendela untuk mengetikkan baris perintah secara langsung.

Keluarga *Windows* menyediakan kumpulan *system Call (API)* yang dipanggil program aplikasi seperti memanggil program atau subrutin. Keluarga *Windows* dikembangkan dengan C, *system call* berbentuk fungsi bahasa C. Meskipun demikian kita dapat memanggil dalam bahasa apapun, kompilator atau interpreter menerjemahkan menjadi panggilan ke sistem operasi *Windows*.

Kumpulan *system call* di keluarga *Windows* disebut WIN32 API. Bentuk fungsi, struktur, pesan, makro dan antarmuka konsisten dan seragam untuk semua keluarga *Microsoft* berbasis 32 bit.

Penciptaan dan pengelolaan antarmuka pemakai, juga fitur-fitur *Windows*, yaitu *Carets, Clipboard, Cursors, Hooks, Icons, dan Menus.*

Kendali umum pada *Windows* terdapat di *common control library, COMCTL32.DLL. Common control DLL* berisi antarmuka pemrograman sehingga aplikasi dapat menciptakan dan memanipulasi *controls* juga menerima masukan *controls*.

- b. *Language Translator*, adalah suatu program sistem yang berfungsi untuk mengkonversikan program aplikasi ke dalam bahasa mesin. Contoh : *Language Translator* pada COBOL dan BASIC.
- c. *Utility Programs*, adalah suatu program yang menjelaskan atau memperluas kegunaan dari sistem operasi suatu sistem komputer. Contoh : *Backup utilities*, *Spooling utilities*, dan *Data compression utilities*.

2.8.2 *Software Aplikasi*

Software aplikasi merupakan suatu perangkat yang memungkinkan pemakai (*user*) memahami sistem komputer [Charles S.P.,'96]. Berikut beberapa contoh dari *software* aplikasi :

1. *Delphi 7.0*

Pemrograman *Delphi* adalah salah satu paket pemrograman visual yang handal dalam membangun aplikasi-aplikasi berbasis *windows*. Dimana dalam pembuatan aplikasi dimulai dengan memperkirakan kebutuhan, merancang tampilan program terlebih dahulu dan selanjutnya diikuti dengan pembuatan kode untuk program tersebut. Komponen paket program *Delphi* adalah :

a. *Window* utama

Pada *window* utama ini semua kegiatan pembuatan program dilakukan. Menu-menu untuk perancangan program terdapat pada *window* ini, yang digunakan selama *design time*.

b. *Component Palette*

Window ini berada pada *window* utama yang berisi beberapa peralatan/komponen untuk perancangan aplikasi selama *design time*. Kontrol-kontrol yang terdapat pada *window* ini merupakan gabungan dari kontrol standar yang digunakan oleh aplikasi *windows*. Kontrol-kontrol tersebut dapat ditambah maupun dikurangi sesuai dengan kebutuhan pemakai dengan pilihan *New Component*, *Install Component* atau *Import ActiveX Control*. Kontrol tambahan ini biasanya disediakan perusahaan partai ketiga dimana kontrol-kontrol tersebut dibuat untuk meningkatkan unjuk kerja program misalnya dalam membuat sebuah aplikasi *Database* yang cukup rumit. Tapi tidak menutup kemungkinan seorang *user* berkeinginan untuk membuat suatu *Object* sendiri.

c. *Object Inspector*

Bagian ini digunakan untuk mengatur sifat objek (*properties*) dan *events*-nya (hal yang akan dilakukan) dari suatu *Component* baik itu berupa *Form*, *Button*, *ListBox*, *ComboBox* atau *Component* lain selama *design time*.

Object inspector terdiri dari beberapa bagian :

- Bagian untuk memilih objek
- *Properties* adalah bagian untuk mengatur sifat-sifat objek
- *Events* adalah bagian yang akan dilakukan dari objek tersebut

Setiap objek yang dibuat secara otomatis akan diberi nama dan memiliki nilai properti *default*.

d. *Project Manager*

Window ini berisi informasi tentang sebuah *project* yang sedang dibangun, sangat berguna untuk manajemen *project*. Dalam membangun sebuah *project* yang besar biasanya melibatkan banyak modul yang berisikan bagian-bagian program yang telah dibagi-bagi agar mempermudah manajemen program dan pengecekan kesalahan. Maka daftar modul yang terlibat dalam pembangunan *project* akan ditampilkan dalam *window* ini.

e. *Unit & Code Explorer*

Unit adalah *window* yang digunakan untuk melakukan *coding*. Dalamnya terdapat *variable*, *function*, *procedure*, *include file (uses)*, *type data* yang digunakan untuk mendukung program. dalam satu modul (*unit*) akan ditampilkan disebelahnya pada *Code Explorer*.

f. *Window Form*

Window ini digunakan untuk membuat tampilan program yang artinya tempat untuk menempatkan komponen-komponen *Delphi*.

Kode-kode program semua diletakkan pada sebuah *unit*, baik itu *event* atau *methods*. *Methods* adalah kejadian yang dapat dilakukan oleh object tersebut sedangkan *event* adalah kejadian yang akan dilakukan oleh object.

Setiap aplikasi *Windows* selalu digerakkan oleh pesan (*message*). Pesan ini dikirimkan oleh *window* ke aplikasi dan aplikasi memberikan respon karena pesan yang diterimanya. Cara ini merupakan teknik yang dilakukan oleh *windows* untuk

implementasi aplikasi-aplikasi yang berada dalam lingkungannya, terutama untuk manajemen sistem supaya beberapa program dapat dijalankan pada saat yang bersamaan (*multitasking*). Pesan-pesan tersebut ada jika terdapat kejadian-kejadian (*methods* atau *event*) apakah itu berasal dari aplikasi itu sendiri atau dari sistem.

Pada *Delphi* sebuah *project* terdiri atas beberapa *file*. Jika program semakin kompleks maka program akan terdiri dari beberapa *form*. Antara *project*, *form*, dan *unit* diletakkan pada *file-file* terpisah.

File-file dalam *Delphi* antara lain :

- file *.DPR (*Delphi Project*) adalah *file* utama dari proyek *Delphi*
- file *.RES (*resource*) adalah *file* yang menyimpan informasi icon
- file *.DFR (*Delphi Form*) adalah *file* yang menyimpan informasi *form*
- file *.PAS (*Delphi Source File*) adalah *file* yang menyimpan *coding* atau *file unit*

Untuk aplikasi yang paling sederhana *Delphi* akan membuat 4 *file* yaitu 1 *file* untuk proyek (*.DPR), 1 *file source* (*.RES), 1 *file* untuk *form* (*.DFR) dan 1 *file* untuk *source* program atau *file unit* (*.PAS). Setiap penambahan *file form* akan mengakibatkan penambahan pada *file unit*.

Delphi selain digunakan untuk merancang juga merupakan sebuah *compiler*. Dengan *compiler* ini, program dapat dibuat menjadi sebuah *file* yang dapat berdiri sendiri tanpa membutuhkan program *Delphi* untuk menjalankannya.

Program-program yang dibuat dengan menggunakan *Delphi* merupakan sebuah aplikasi yang kompleks. Walaupun program yang dirancang berukuran kecil, jika menggunakan Database maka ada beberapa *file* yang harus disertakan. Untuk itu *Delphi* mempunyai fasilitas *InstallShield* yang berfungsi mengemas *file-file delphi* agar program (*exe-nya*) dapat di jalankan komputer lain dengan cara meng-*install-nya*.

Objek User interface yang digunakan oleh *Delphi* sama dengan yang dimiliki oleh *windows*. Bagi pemrogram *objek user interface* disebut sebagai *Component*. Komponen-komponen ini digunakan sebagai media input bagi pemakai yang tujuannya untuk media berinteraksi antara pemakai dengan program yang dijalankan. Komponen dapat berupa suatu *tools* yang digunakan untuk

masukan (*input*) ataupun keluaran (*output*), atau hanya sebagai pesan (*message*) yang ditampilkan ke layar monitor. Jenis *Component* dalam *Delphi* pada saat perancangannya dibagi menjadi 2 yaitu :

- visual adalah komponen yang dapat dilihat pada saat *runtime*
- non visual adalah komponen yang hanya dapat dilihat pada saat desain.

Setiap komponen memiliki properti seperti warna, tulisan, posisi dan lain-lain. Hal ini dikarenakan *Component* pada *Delphi* mempunyai karakteristik :

- *Inheritance*
- *Encapsulation*
- *Polymorphism*

Form merupakan dasar dari setiap aplikasi yang dikembangkan dengan menggunakan *Delphi*. Dari *form* dapat dirancang beberapa *file window*, antara lain : *window* utama, *window* dialog, *window Multiple Document Interface (MDI)*.

Delphi adalah bahasa pemrograman yang bekerja dalam lingkungan *Windows*. *Delphi* juga merupakan sarana (*tool*) untuk menghasilkan program-program aplikasi berbasis *Windows (Object Oriented Programming)*.

2. *Paradox*

Pada aplikasi yang akan dikembangkan penulis menggunakan Database *Paradox* versi 7 yang merupakan built-in *Delphi* 7.0. Konsep yang terdapat pada *Paradox* 7 sama dengan konsep database pada umumnya. Ada beberapa tipe data yang disediakan oleh *paradox* 7.0 antara lain :

- | | | |
|-----------------|------------------------|----------------------------|
| - <i>Alpha</i> | - <i>Number</i> | - <i>\$(money)</i> |
| - <i>Short</i> | - <i>Long Interger</i> | - <i>#(BCD)</i> |
| - <i>Time</i> | - <i>Date</i> | - <i>@(Timestamp)</i> |
| - <i>Memo</i> | - <i>Graphics</i> | - <i>Formatted Memo</i> |
| - <i>OLE</i> | - <i>Logical</i> | - <i>± (autoincrement)</i> |
| - <i>Binery</i> | - <i>Bytes</i> | |

Untuk memulai membuat tabel terlebih dulu kita harus membuka menu *Tools* dengan sub menu *Database Desktop*. Kita juga dapat memakai sistem database sesuai dengan yang kita inginkan. Sebelum kita membuat database dengan *paradox* 7.0 terlebih dulu kita harus membuka menu *Databases* dengan sub menu *Explorer* sebagai *Connectivity System* maupun *Database Connectivity*.

BAB III

ANALISIS SISTEM

Analisis sistem merupakan proses awal yang harus dilaksanakan untuk menentukan permasalahan yang sedang dihadapi. Tahap ini adalah sangat penting, karena proses analisis yang kurang akurat akan menyebabkan hasil dari suatu pengembangan perangkat lunak akan tidak sesuai dengan yang diharapkan. Jadi proses ini harus benar-benar sesuai dengan keinginan pihak pengguna agar hasil pengembangan perangkat lunak memuaskan pengguna.

Dalam dunia hukum sering kali terjadi kasus-kasus yang melibatkan beberapa tersangka yang kemungkinan tidak sengaja melakukan tindakan yang melanggar hukum. Tersangka secara tidak sadar melakukan tindakan yang melanggar hukum karena tersangka tidak tahu betul apa yang menyebabkan dirinya dikenai tindakan hukum. Ini bisa disebabkan karena tersangka tidak mengetahui bahwa hal yang telah dilakukan ada aturan yang mengikatnya.

Masalah yang akan dianalisis ini merupakan salah satu masalah hukum yang berkaitan dengan Undang-Undang Republik Indonesia Nomor 19 Tahun 2002 Tentang Hak Cipta. Undang-undang ini membatasi seseorang dalam melakukan tindakan yang berkaitan dengan hak cipta dengan aturan yang telah disyahkan oleh pemerintah melalui undang-undang tersebut. Undang-undang hak cipta adalah penghargaan bagi setiap pencipta karya seni termasuk dengan karya intelektual dibuat oleh seseorang.

Pada dasarnya permasalahan yang kadang dihadapi oleh para penegak hukum (pakar hukum) dan pelaku hukum sedikit banyak memakan waktu dalam usaha menyelesaikan suatu kasus atau masalah tertentu. Permasalahan-permasalahan yang dihadapi akan sangat kompleks apabila para penegak hukum atau pelaku hukum tidak memiliki kemampuan yang cukup atau mempunyai pengalaman yang memadai dalam menyelesaikan suatu masalah hukum. Di bawah ini ada beberapa *point* penting tentang masalah tersebut.

3.1 Identifikasi Masalah

Masalah-masalah yang dapat diidentifikasi sebagai dasar bagi pengembangan sistem pakar ini terdiri dari dua bagian permasalahan, yaitu permasalahan bagi penegak hukum dan pelaku hukum.

- 1) Permasalahan bagi penegak hukum
 - a) Banyaknya penegak hukum atau pakar hukum yang kadang kala mengalami kesulitan dalam menentukan pasal-pasal untuk menjerat pelaku kejahatan.
 - b) Pendetang baru dalam bidang hak cipta menuntut para penegak hukum untuk menguasai pasal-pasal dalam undang-undang tersebut.
 - c) Seringkali terjadi perbedaan interpretasi terhadap pasal-pasal oleh para penegak hukum.
 - d) Memerlukan ruang untuk menyimpan buku undang-undang tersebut ditempat yang aman.
 - e) Para penegak hukum membutuhkan waktu yang cukup lama untuk mencari pasal-pasal tertentu untuk disesuaikan dengan kasus yang baru terjadi.
 - f) Dalam sesi konsultasi para penegak hukum dimungkinkan mengalami kelelahan berfikir dalam memberikan keterangan atas sanksi kepada pelaku hukum.
- 2) Permasalahan bagi pelaku hukum
 - a) Banyak pelaku hukum tidak tahu apakah tindakan mereka benar atau melanggar hukum.
 - b) Banyak pelaku hukum yang tidak tahu persis isi dari pasal-pasal yang berhubungan dengan kasus yang dihadapi pelaku hukum tersebut.
 - c) Banyak pelaku hukum tidak tahu sanksi apa yang akan dikenakan oleh aparat hukum jika pelaku hukum tersebut telah melanggar pasal-pasal tentang hak cipta.

3.2 Penyebab Masalah

Dari pembahasan sebelumnya telah dijelaskan permasalahan yang sering terjadi dalam bidang hukum baik bagi para penegak hukum maupun para pelaku hukum. Dibawah ini ada beberapa penyebab terjadinya permasalahan, yaitu :

- 1) Kemampuan seorang pakar hukum dalam hal ini penegak hukum untuk dapat mengingat semua pasal-pasal yang ada dalam undang-undang hak cipta tersebut.
- 2) Keterbatasan otak manusia yang memungkinkan kelelahan pada otak manusia. Dalam kasus ini penegak hukum kadang-kadang merasa lelah setelah menangani sesi konsultasi dengan pasien hukum.
- 3) Karena penegak hukum adalah seorang manusia yang memiliki keterbatasan berpikir ada kemungkinan salah persepsi dalam memberikan saran atau nasehat kepada pasien hukum.
- 4) Pendekatan yang dilakukan para penegak hukum berbeda antara satu dengan yang lainnya dalam usaha membantu pelaku hukum (*client*) yang sedang menghadapi kasus.

3.3 Pendekatan Solusi (*the approach of solution*)

Berdasarkan permasalahan dan penyebab masalah diatas, dapat dicarikan beberapa pendekatan untuk menyelesaikan permasalahan di atas. Solusi untuk menyelesaikan permasalahan tersebut diuraikan dalam pendekatan-pendekatan berikut ini.

Pendekatan pertama dalam penyebab masalah *point* pertama adalah dengan menyediakan suatu perangkat lunak yang dapat menyimpan data atau biasa disebut dengan basisdata (*database*). Basisdata ini digunakan untuk menyimpan pasal-pasal yang cukup banyak jumlahnya.

Pendekatan kedua dalam penyebab masalah *point* kedua adalah pengembangan sistem pakar yang dapat memproses atau mengambil kesimpulan yang tepat bagi penegak hukum dan pelaku hukum.

Pendekatan ketiga dalam penyebab masalah *point* ketiga adalah pengembangan sistem pakar yang dapat memberikan persepsi terhadap pasal-pasal secara tepat tanpa terjadi *perception ambiguous*.

Pendekatan keempat dalam penyebab masalah *point* keempat adalah perlu adanya pengembangan perangkat lunak dengan teknik inferensi tertentu dalam serta dapat memberikan saran, nasehat, alasan dan rekomendasi bagi setiap pengguna.

3.4 Hasil Analisis Sistem

Hasil analisis yang diperoleh dari penjabaran atas permasalahan dalam identifikasi masalah yang dikombinasikan dengan pendekatan solusi, maka diperoleh kesimpulan-kesimpulan sebagai berikut.

- 1) Perlu adanya pengembangan sistem pakar untuk membantu para penegak hukum dan pelaku hukum
- 2) Sistem pakar yang dikembangkan harus dapat menyimpan pasal-pasal ke dalam suatu basisdata.
- 3) Dalam mengambil suatu kesimpulan, sistem pakar harus dapat mengajukan beberapa pertanyaan yang kemudian pengguna sistem memberi jawaban. Dari hasil jawaban tersebut dapat diidentifikasi solusi yang tepat bagi pengguna.
- 4) Perlu adanya pelayanan sistem pakar yaitu pelayanan sesi konsultasi dan sesi pencarian pasal.

3.5 Hasil Analisis Pengetahuan (*Knowledge Analysis Result*)

Dari hasil analisis pengetahuan yang akan digunakan dalam pengembangan basis pengetahuan ini terdapat beberapa *goal* yang harus terpenuhi dalam sistem pakar. *Goal* tersebut merupakan sanksi-sanksi bagi para pengguna jika melakukan perbuatan hukum yang melanggar undang-undang hak cipta, yaitu

- 1) Pasal 72 ayat 1

Barangsiapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp 5.000.000.000,00 (lima miliar rupiah).

- 2) Pasal 72 ayat 2
Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu Ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
- 3) Pasal 72 ayat 3
Barangsiapa dengan sengaja dan tanpa hak memperbanyak penggunaan untuk kepentingan komersial suatu Program Komputer dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
- 4) Pasal 72 ayat 4
Barangsiapa dengan sengaja melanggar Pasal 17 dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
- 5) Pasal 72 ayat 5
Barangsiapa dengan sengaja melanggar Pasal 19, Pasal 20, atau Pasal 49 ayat (3) dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).
- 6) Pasal 72 ayat 6
Barangsiapa dengan sengaja dan tanpa hak melanggar Pasal 24 atau Pasal 55 dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).
- 7) Pasal 72 ayat 7
Barangsiapa dengan sengaja dan tanpa hak melanggar Pasal 25 dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).
- 8) Pasal 72 ayat 8
Barangsiapa dengan sengaja dan tanpa hak melanggar Pasal 27 dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).

9) Pasal 72 ayat 9

Barangsiapa dengan sengaja melanggar Pasal 28 dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 1.500.000.000,00 (satu miliar lima ratus juta rupiah).

BAB IV

PERANCANGAN SISTEM

4.1 Perancangan Basis Pengetahuan (*Knowlegde Base Designing*)

Perancangan basis pengetahuan merupakan langkah awal dalam perancangan sistem yang berkaitan dalam pengembangan sistem pakar ini. Dalam proses perancangan basis pengetahuan ini dibagi menjadi 9 gol atau hasil akhir yang menjadi landasan dalam menentukan *rule-rule*. Gol-gol tersebut telah dibahas pada bab analisis sistem.

4.1.1 Rules

Perancangan *rule* dalam gol 1 ini memanfaatkan variabel yang biasa digunakan dalam *backward-chaining*. Variabel tersebut akan memudahkan representasi pengetahuan dengan menggunakan *rule*. Variabel tersebut adalah :

4.1.1.1 Rule untuk gol 1

Gol 1 : Barangsiapa dengan sengaja dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam Pasal 2 ayat (1) atau Pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling singkat 1 (satu) bulan dan/atau denda paling sedikit Rp 1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp 5.000.000.000,00 (lima miliar rupiah).

Tabel 4.1 Rule 1

Kode Rule	1	
Precondition	Portfolio	Variabel
IF	Penerima hak cipta = “true”	A1
THEN	Memiliki hak eksklusif	A2
ELSE	Tidak berhak memperbanyak ciptaan orang lain	A3

Tabel 4.2 Rule 2

Kode Rule	2	
Precondition	Portfolio	Variabel
IF	Memiliki hak eksklusif = "true"	A2
THEN	Seorang produser rekaman	A4
ELSE	Tidak berhak memperbanyak rekaman suara	A5

Tabel 4.3 Rule 3

Kode Rule	3	
Precondition	Portfolio	Variabel
IF	Tidak berhak memperbanyak ciptaan orang lain = "true"	A3
THEN	Melanggar pasal 2 ayat 1	Pasal 2 ayat 1
ELSE	Memiliki hak eksklusif	A2

Tabel 4.4 Rule 4

Kode Rule	4	
Precondition	Portfolio	Variabel
IF	Seorang produser rekaman = "true"	A4
THEN	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran
ELSE	Tidak berhak memberi ijin kepada orang lain untuk membuat rekaman suara	A8

Tabel 4.5 Rule 5

Kode Rule	5	
Precondition	Portfolio	Variabel
IF	Tidak berhak memperbanyak rekaman suara = "true"	A4
THEN	Melanggar pasal 49 ayat 1	Pasal 49 ayat 1
ELSE	Tidak berhak menyewakan rekaman suara	A6

Tabel 4.6 Rule 6

Kode Rule	6	
Precondition	Portfolio	Variabel
IF	Tidak berhak menyewakan rekaman suara = "true"	A6
THEN	Melanggar pasal 49 ayat 1	Pasal 49 ayat 1
ELSE	Tidak berhak membuat dan memperbanyak gambar pertunjukan tanpa ijin	A7

Tabel 4.7 Rule 7

Kode Rule	7	
Precondition	Portfolio	Variabel
IF	Tidak berhak membuat dan memperbanyak gambar pertunjukan tanpa ijin = "true"	A7
THEN	Melanggar pasal 49 ayat 1	Pasal 49 ayat 1
ELSE	Seorang produser rekaman	A4

Tabel 4.8 Rule 8

Kode Rule	8	
Precondition	Portfolio	Variabel
IF	Tidak berhak memberi ijin kepada orang lain untuk membuat rekaman suara = "true"	A8
THEN	Melanggar pasal 49 ayat 2	Pasal 49 ayat 2
ELSE	Tidak berhak menyewakan rekaman suara kepada orang lain	A9

Tabel 4.9 Rule 9

Kode Rule	9	
Precondition	Portfolio	Variabel
IF	Tidak berhak menyewakan rekaman suara kepada orang lain = “true”	A9
THEN	Melanggar pasal 49 ayat 2	Pasal 49 ayat 2
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.2 Rule untuk gol 2

Gol 2 : Barangsiapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu Ciptaan atau barang hasil pelanggaran Hak Cipta atau Hak Terkait sebagaimana dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

Tabel 4.10 Rule 10

Kode Rule	10	
Precondition	Portfolio	Variabel
IF	Menyiarkan barang hasil pelanggaran hak cipta = “true”	B1
THEN	Melanggar pasal 72 ayat 2	Pasal 72 ayat 2
ELSE	Memamerkan suatu ciptaan kepada umum	B2

Tabel 4.11 Rule 11

Kode Rule	11	
Precondition	Portfolio	Variabel
IF	Memamerkan suatu ciptaan kepada umum = “true”	B2
THEN	Melanggar pasal 72 ayat 2	Pasal 72 ayat 2
ELSE	Mengedarkan suatu ciptaan kepada umum	B3

Tabel 4.12 Rule 12

Kode Rule	12	
Precondition	Portfolio	Variabel
IF	Mengedarkan suatu ciptaan kepada umum = "true"	B3
THEN	Melanggar pasal 72 ayat 2	Pasal 72 ayat 2
ELSE	Menjual suatu ciptaan kepada umum	B4

Tabel 4.13 Rule 13

Kode Rule	13	
Precondition	Portfolio	Variabel
IF	Menjual suatu ciptaan kepada umum = "true"	B4
THEN	Melanggar pasal 72 ayat 2	Pasal 72 ayat 2
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.3 Rule untuk gol 3

Gol 3 : Barangsiapa dengan sengaja dan tanpa hak memperbanyak penggunaan untuk kepentingan komersial suatu Program Komputer dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).

Tabel 4.14 Rule 14

Kode Rule	14	
Precondition	Portfolio	Variabel
IF	Memiliki hak cipta untuk memperbanyak suatu ciptaan = "true"	C1
THEN	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran
ELSE	Memperbanyak program komputer untuk kepentingan komersial	C2

Tabel 4.15 Rule 15

Kode Rule	15	
Precondition	Portfolio	Variabel
IF	Memperbanyak program komputer untuk kepentingan komersial = “true”	C2
THEN	Melanggar pasal 72 ayat 3	Pasal 72 ayat 3
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.4 Rule untuk gol 4

Gol 4 : Barangsiapa dengan sengaja melanggar Pasal 17 dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).

Tabel 4.16 Rule 16

Kode Rule	16	
Precondition	Portfolio	Variabel
IF	Mengedarkan suatu ciptaan yang merendahkan nilai-nilai agama = “true”	D1
THEN	Melanggar pasal 17	Pasal 17
ELSE	Mengedarkan suatu ciptaan yang menimbulkan kesukuan atau ras	D2

Tabel 4.17 Rule 17

Kode Rule	17	
Precondition	Portfolio	Variabel
IF	Mengedarkan suatu ciptaan yang menimbulkan kesukuan atau ras = “true”	D2
THEN	Melanggar pasal 17	Pasal 17
ELSE	Mengedarkan suatu ciptaan yang menimbulkan gangguan atau bahaya keamanan negara	D3

Tabel 4.18 Rule 18

Kode Rule	18	
Precondition	Portfolio	Variabel
IF	Mengedarkan suatu ciptaan yang menimbulkan gangguan atau bahaya keamanan negara = "true"	D3
THEN	Melanggar pasal 17	Pasal 17
ELSE	Mengedarkan suatu ciptaan yang bertentangan dengan norma umum dan mengganggu ketertiban umum	D4

Tabel 4.19 Rule 19

Kode Rule	19	
Precondition	Portfolio	Variabel
IF	Mengedarkan suatu ciptaan yang bertentangan dengan norma umum dan mengganggu ketertiban umum = "true"	D4
THEN	Melanggar pasal 17	Pasal 17
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.5 Rule untuk gol 5

Gol 5 : Barangsiapa dengan sengaja melanggar Pasal 19, Pasal 20, atau Pasal 49 ayat (3) dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).

Tabel 4.20 Rule 20

Kode Rule	20	
Precondition	Portfolio	Variabel
IF	Memperbanyak potret tanpa ijin yang punya = "true"	E1
THEN	Melanggar pasal 19 ayat 1	Pasal 19 ayat 1
ELSE	Memperbanyak potret tanpa ijin ahli waris	E2

Tabel 4.21 Rule 21

Kode Rule	21	
Precondition	Portfolio	Variabel
IF	Memperbanyak potret tanpa ijin ahli waris = "true"	E2
THEN	Melanggar pasal 19 ayat 1	Pasal 19 ayat 1
ELSE	Menyebarkan potret/photo tersebut tanpa sepengetahuan yang ada di photo	E3

Tabel 4.22 Rule 22

Kode Rule	22	
Precondition	Portfolio	Variabel
IF	Menyebarkan potret/photo tersebut tanpa sepengetahuan yang ada di photo = "true"	E3
THEN	Melanggar pasal 19 ayat 2	Pasal 19 ayat 2
ELSE	Memotret tanpa persetujuan orang yang dipotret	E4

Tabel 4.23 Rule 23

Kode Rule	23	
Precondition	Portfolio	Variabel
IF	Memotret tanpa persetujuan orang yang dipotret	E4
THEN	Melanggar pasal 19 ayat 2	Pasal 19 ayat 2
ELSE	Memiliki atau merupakan lembaga penyiaran	E5

Tabel 4.24 Rule 24

Kode Rule	24	
Precondition	Portfolio	Variabel
IF	Memiliki atau merupakan lembaga penyiaran = "true"	E5
THEN	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran
ELSE	Memberi ijin kepada orang lain untuk membuat ulang karya siaran	E6

Tabel 4.25 Rule 25

Kode Rule	25	
Precondition	Portfolio	Variabel
IF	Memberi ijin kepada orang lain untuk membuat ulang karya siaran	E6
THEN	Melanggar pasal 49 ayat 3	Pasal 49 ayat 3
ELSE	Memperbanyak atau menyiarkan karya siaran melalui sistem elektromagnetik	E7

Tabel 4.26 Rule 26

Kode Rule	26	
Precondition	Portfolio	Variabel
IF	Memperbanyak atau menyiarkan karya siaran melalui sistem elektromagnetik = "true"	E7
THEN	Melanggar pasal 49 ayat 3	Pasal 49 ayat 3
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.6 Rule untuk gol 6

Gol 6 : Barangsiapa dengan sengaja dan tanpa hak melanggar Pasal 24 atau Pasal 55 dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).

Tabel 4.27 Rule 27

Kode Rule	27	
Precondition	Portfolio	Variabel
IF	Pemegang hak cipta = “true”	F1
THEN	Tidak mencantumkan nama pencipta atau ahli waris	F3
ELSE	Pencipta karya seni	F2

Tabel 4.28 Rule 28

Kode Rule	28	
Precondition	Portfolio	Variabel
IF	Tidak mencantumkan nama pencipta atau ahli waris = “true”	F3
THEN	Melanggar pasal 24 ayat 1	Pasal 24 ayat 1
ELSE	Mengubah ciptaan yang sudah diserahkan kepada orang lain	F4

Tabel 4.29 Rule 29

Kode Rule	29	
Precondition	Portfolio	Variabel
IF	Mengubah ciptaan yang sudah diserahkan kepada orang lain = “true”	F4
THEN	Melanggar pasal 24 ayat 2	Pasal 24 ayat 2
ELSE	Mengubah judul ciptaan atau nama pencipta atau menyamarkan nama pencipta	F5

Tabel 4.30 Rule 30

Kode Rule	30	
Precondition	Portfolio	Variabel
IF	Mengubah judul ciptaan atau nama pencipta atau menyamakan nama pencipta = "true"	F5
THEN	Melanggar pasal 24 ayat 3	Pasal 24 ayat 3
ELSE	Pencipta karya seni	F2

Tabel 4.31 Rule 31

Kode Rule	31	
Precondition	Portfolio	Variabel
IF	Pencipta karya seni = "true"	F2
THEN	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran
ELSE	Mengubah suatu ciptaan dan tidak sesuai dengan kepatutan yang ada dalam masyarakat	F6

Tabel 4.32 Rule 32

Kode Rule	32	
Precondition	Portfolio	Variabel
IF	Mengubah suatu ciptaan dan tidak sesuai dengan kepatutan yang ada dalam masyarakat = "true"	F6
THEN	Melanggar pasal 24 ayat 4	Pasal 24 ayat 4
ELSE	Meniadakan nama pencipta atau mengubah nama judul ciptaan	F7

Tabel 4.33 Rule 33

Kode Rule	33	
Precondition	Portfolio	Variabel
IF	Pemegang hak cipta dalam bidang informasi elektronik = "true"	G1
THEN	Melanggar pasal 55	Pasal 55
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.7 Rule untuk gol 7

Gol 7 : Barangsiapa dengan sengaja dan tanpa hak melanggar Pasal 25 dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).

Tabel 4.34 Rule 34

Kode Rule	34	
Precondition	Portfolio	Variabel
IF	Meniadakan nama pencipta atau mengubah nama judul ciptaan = "true"	F7
THEN	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran
ELSE	Menghilangkan nama pencipta elektronik untuk kepentingan komersil	G2

Tabel 4.35 Rule 35

Kode Rule	35	
Precondition	Portfolio	Variabel
IF	Menghilangkan nama pencipta elektronik untuk kepentingan komersil = "true"	G2
THEN	Melanggar pasal 25	Pasal 25
ELSE	Mengubah nama pencipta informasi elektronik untuk kepentingan komersil	G3

Tabel 4.36 Rule 36

Kode Rule	36	
Precondition	Portfolio	Variabel
IF	Mengubah nama pencipta informasi elektronik untuk kepentingan komersil = "true"	G3
THEN	Melanggar pasal 25	Pasal 25
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.8 Rule untuk gol 8

Gol 8 : Barangsiapa dengan sengaja dan tanpa hak melanggar Pasal 27 dipidana dengan pidana penjara paling lama 2 (dua) tahun dan/atau denda paling banyak Rp 150.000.000,00 (seratus lima puluh juta rupiah).

Tabel 4.37 Rule 37

Kode Rule	37	
Precondition	Portfolio	Variabel
IF	Memiliki hak yang berkaitan dengan ciptaan = "true"	H1
THEN	Merusak atau meniadakan ijin pencipta	H2
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

Tabel 4.38 Rule 38

Kode Rule	38	
Precondition	Portfolio	Variabel
IF	Merusak atau meniadakan ijin pencipta = "true"	H2
THEN	Melanggar pasal 27	Pasal 27
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.1.9 Rule untuk gol 9

Gol 9 : Barangsiapa dengan sengaja melanggar Pasal 28 dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp 1.500.000.000,00 (satu miliar lima ratus juta rupiah).

Tabel 4.39 Rule 39

Kode Rule	39	
Precondition	Portfolio	Variabel
IF	Menciptakan sautu produk yang berteknologi tinggi khususnya serat optik = "true"	I1
THEN	Telah memenuhi peraturan perijinan dan persyaratan produksi yang ditetapkan oleh pihak yang berwenang	I2
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

Tabel 4.40 Rule 40

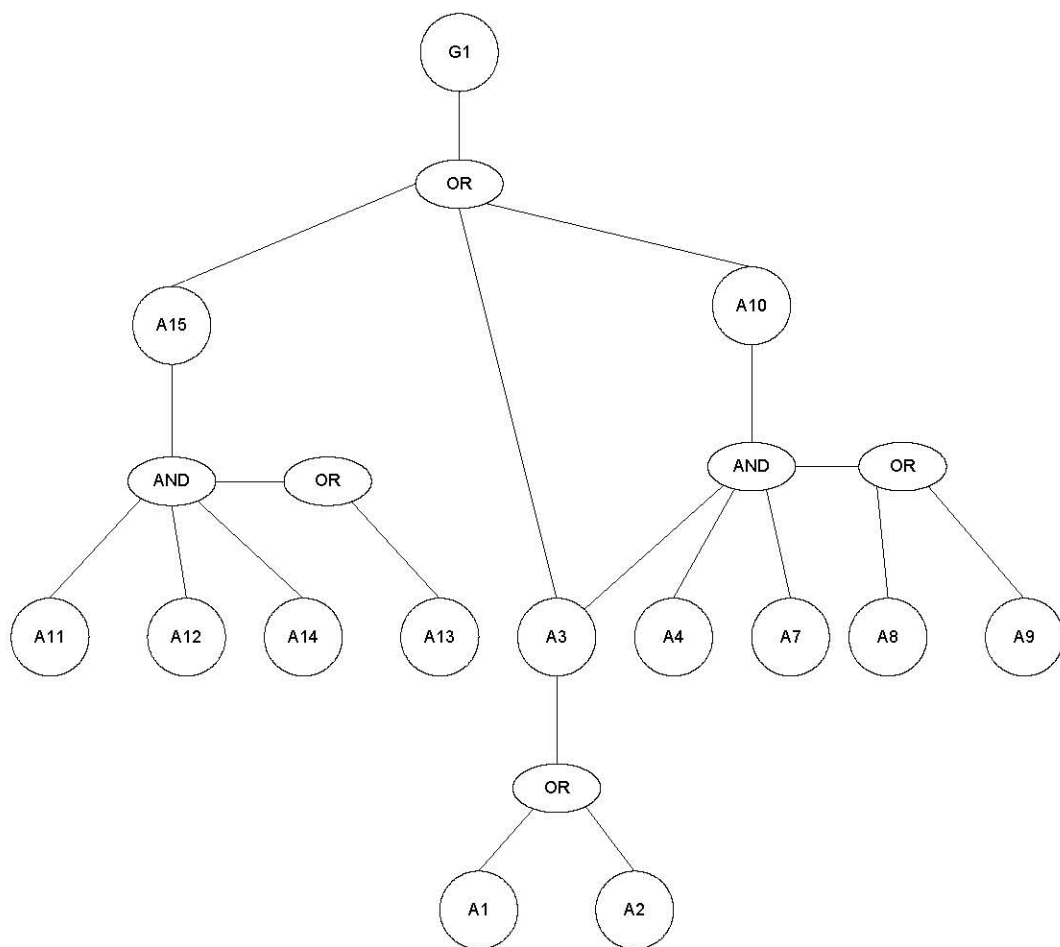
Kode Rule	40	
Precondition	Portfolio	Variabel
IF	Telah memenuhi peraturan perijinan dan persyaratan produksi yang ditetapkan oleh pihak yang berwenang = "true"	I2
THEN	Melanggar pasal 28	Pasal 28
ELSE	Tidak melakukan pelanggaran	Tidak melakukan pelanggaran

4.1.2 Inference Networking

Inference Networking merupakan notasi yang digunakan untuk merancang inferensi serta memperjelas *relationship* antar informasi. Selain hal tersebut *inference engine* juga digunakan untuk merepresentasikan gol. Untuk masing-masing *inference networking* adalah sebagai berikut :

4.1.2.1 Inference Networking untuk Gol 1

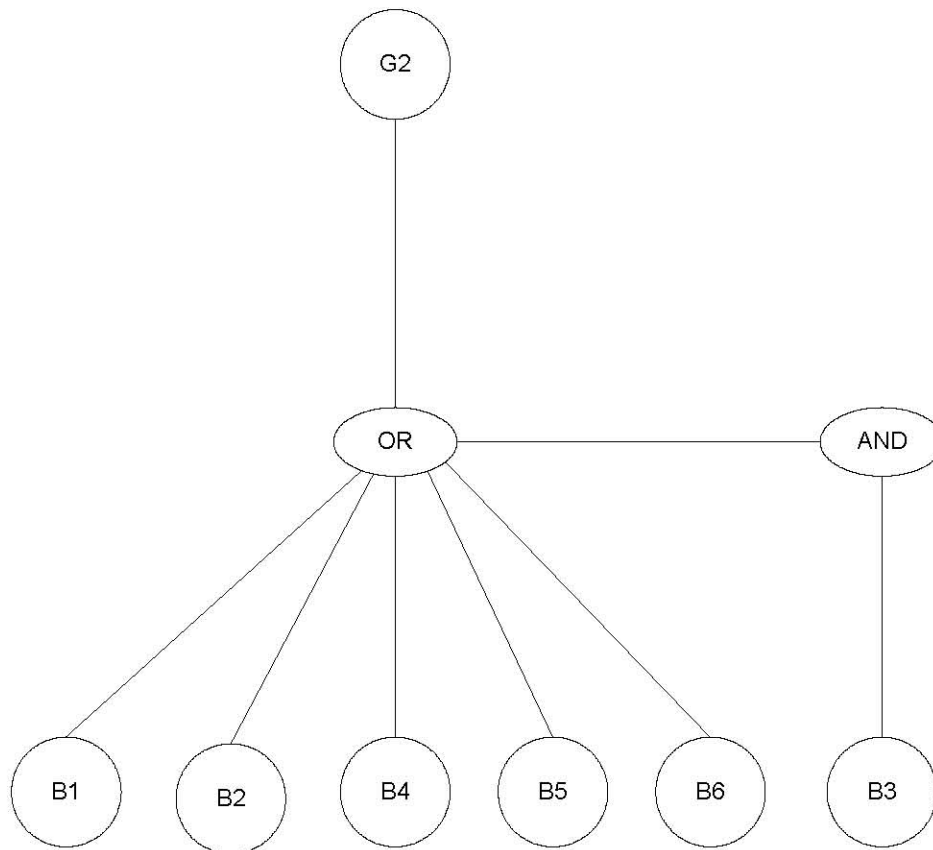
Representasi gol untuk gol 1 terdiri dari empat belas *node* yang terdiri dari satu gol (*portfolio*) dan tiga belas *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.1 Inference Networking untuk Gol 1

4.1.2.2 *Inference Networking* untuk Gol 2

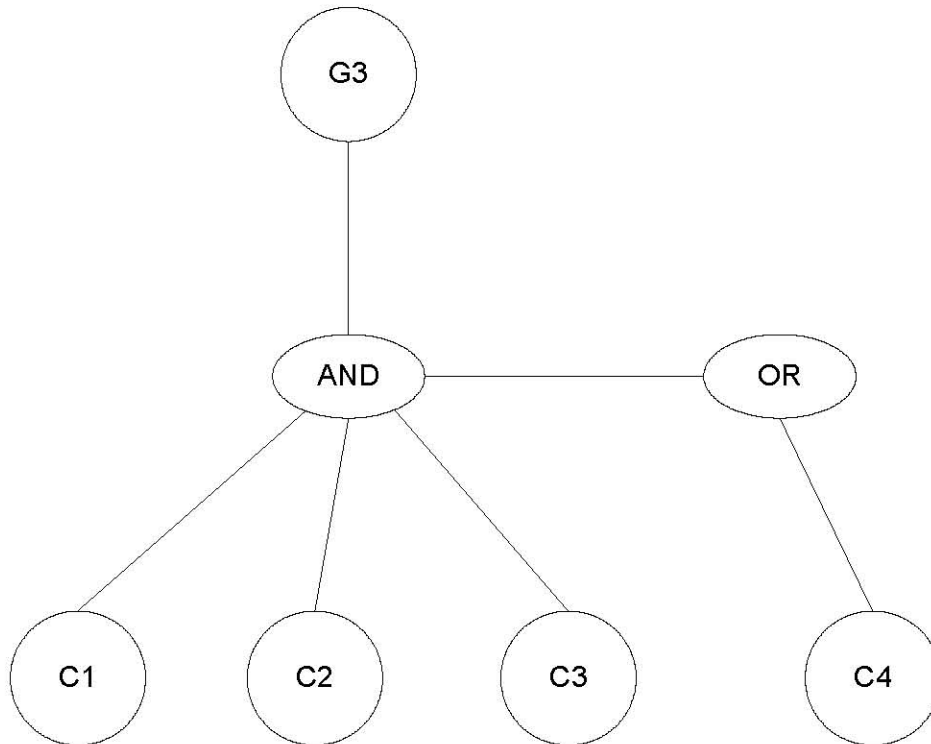
Representasi gol untuk gol 2 terdiri dari tujuh *node* yang terdiri dari satu gol (*portfolio*) dan enam *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.2 *Inference Networking* untuk Gol 2

4.1.2.3 *Inference Networking* untuk Gol 3

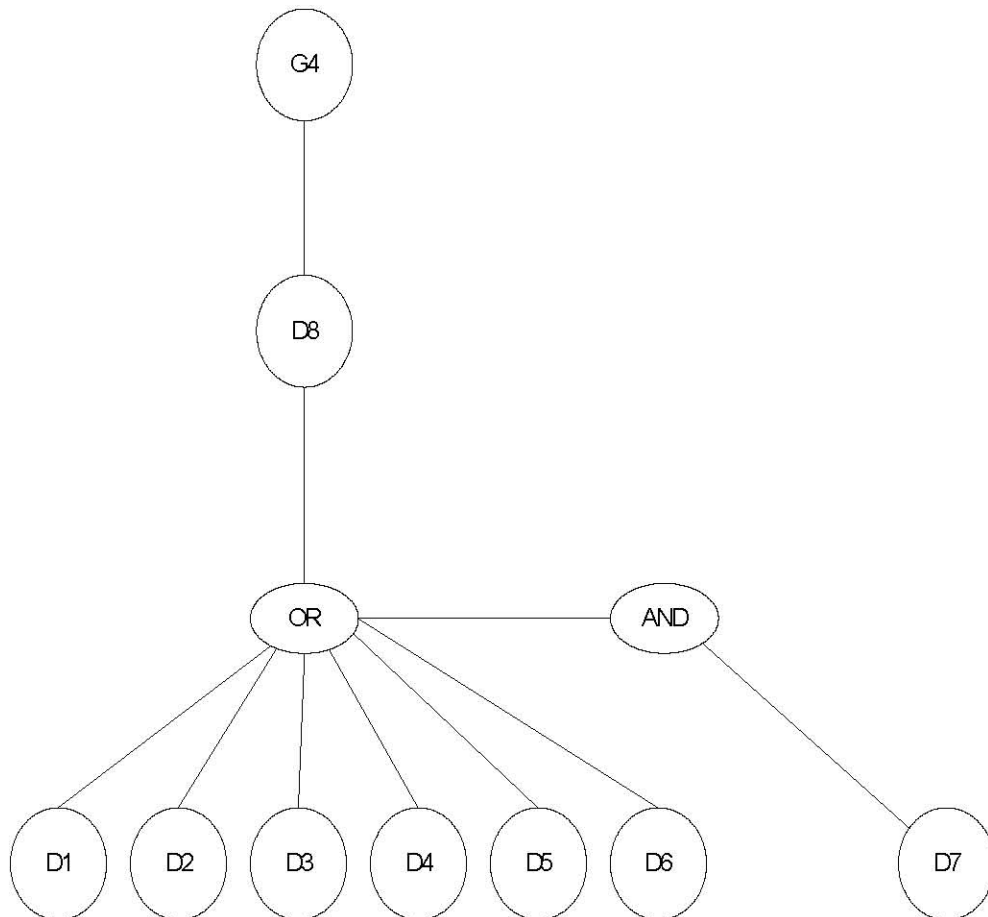
Representasi gol untuk gol 3 terdiri dari lima *node* yang terdiri dari satu gol (*portfolio*) dan empat *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.3 *Inference Networking* untuk Gol 3

4.1.2.4 *Inference Networking* untuk Gol 4

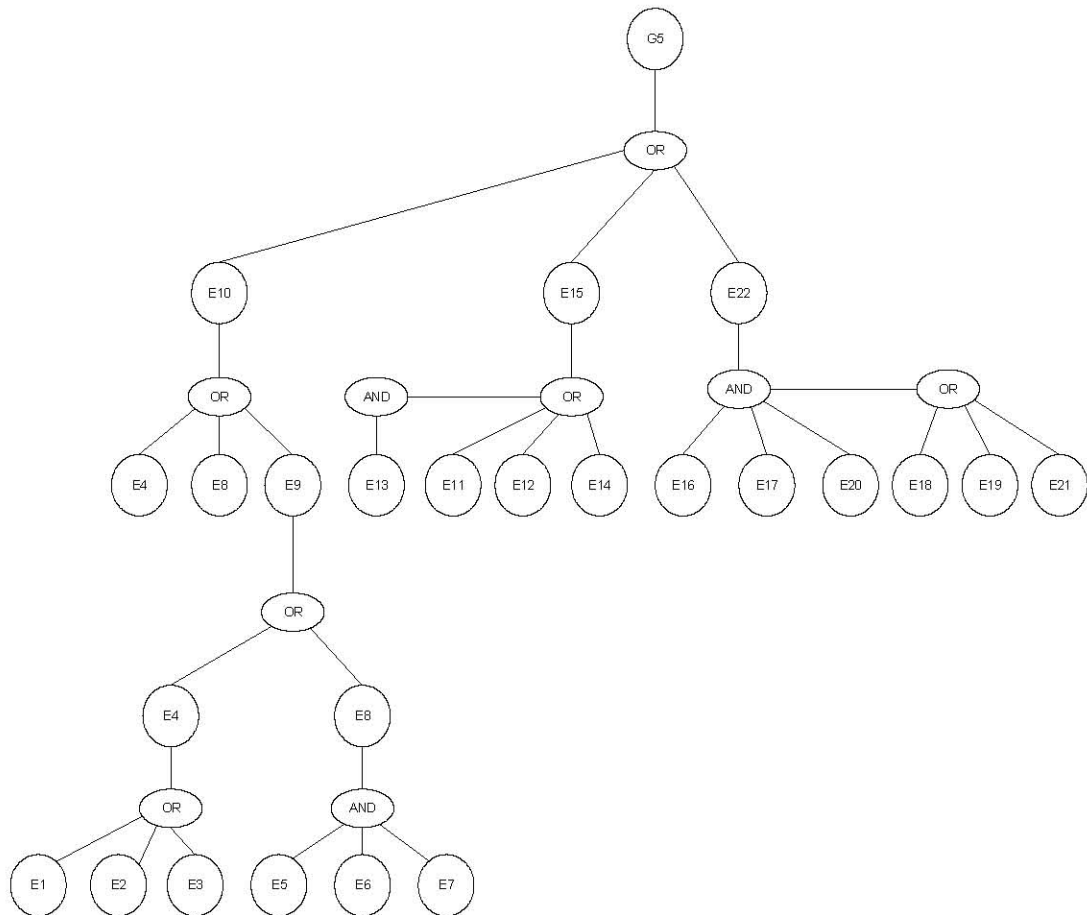
Represenasi gol untuk gol 4 terdiri dari sembilan *node* yang terdiri dari satu gol (*portfolio*) dan delapan *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.4 *Inference Networking* untuk Gol 4

4.1.2.5 *Inference Networking* untuk Gol 5

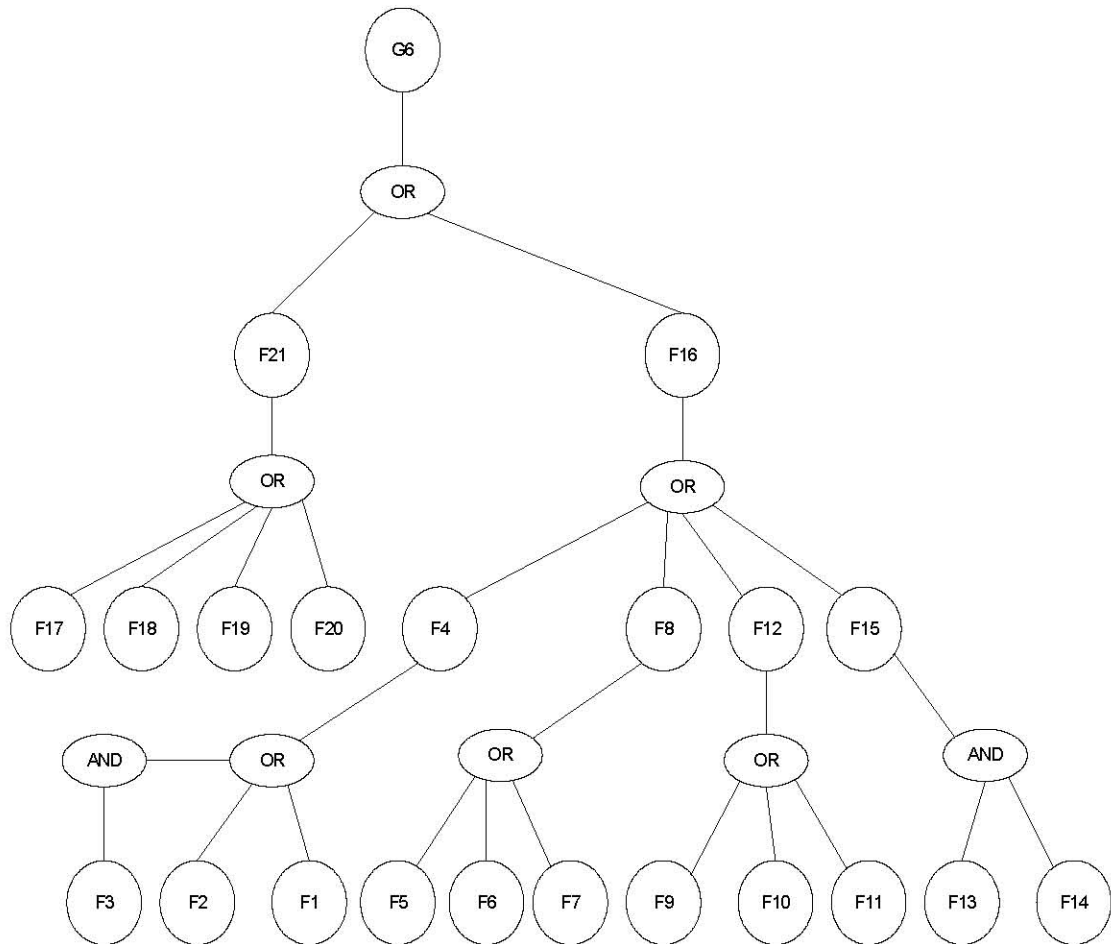
Representasi gol untuk gol 5 terdiri dari dua lima *node* yang terdiri dari satu gol (*portfolio*) dan dua puluh empat *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.5 *Inference Networking* untuk Gol 5

4.1.2.6 *Inference Networking* untuk Gol 6

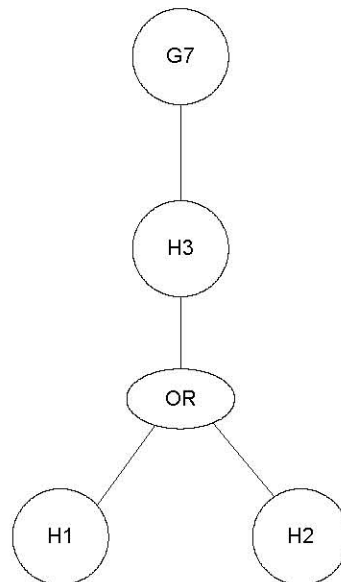
Representasi gol untuk gol 6 terdiri dari dua puluh dua *node* yang terdiri dari satu gol (*portfolio*) dan dua puluh satu *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.6 *Inference Networking* untuk Gol 6

4.1.2.7 *Inference Networking* untuk Gol 7

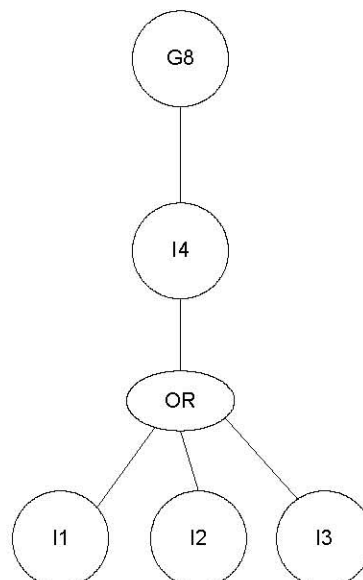
Representasi gol untuk gol 7 terdiri dari empat *node* yang terdiri dari satu gol (*portfolio*) dan tiga *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.7 *Inference Networking* untuk Gol 7

4.1.2.8 *Inference Networking* untuk Gol 8

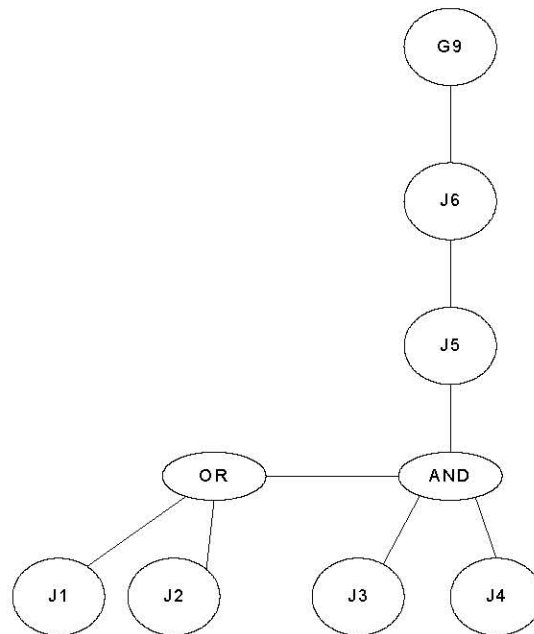
Representasi gol untuk gol 8 terdiri dari lima *node* yang terdiri dari satu gol (*portfolio*) dan tujuh *precondition* yang dijelaskan dalam gambar dibawah.



Gambar 4.8 *Inference Networking* untuk Gol 8

4.1.2.9 *Inference Networking untuk Gol 9*

Represenasi gol untuk gol 9 terdiri dari tujuh *node* yang terdiri dari satu gol (*portfolio*) dan enam *precondition* yang dijelaskan dalam gambar dibawah.



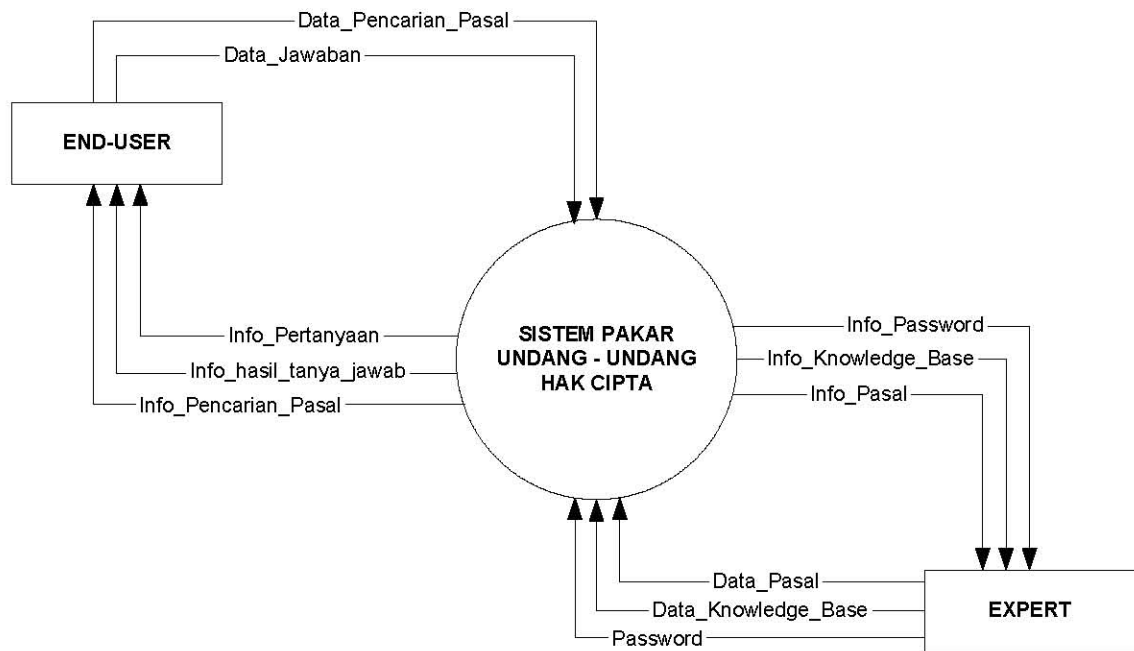
Gambar 4.9 *Inference Networking untuk Gol 9*

4.2 *Data Flow Diagram (Diagram Aliran Data)*

Diagram aliran data memungkinkan pengembangann model domain informasi dan domain fungsional pada saat yang sama. Diagram Aliran Data level 0 atau biasa disebut Diagram Konteks untuk sistem pakar ini adalah sebagai berikut :

4.2.1 *Data Context Diagram*

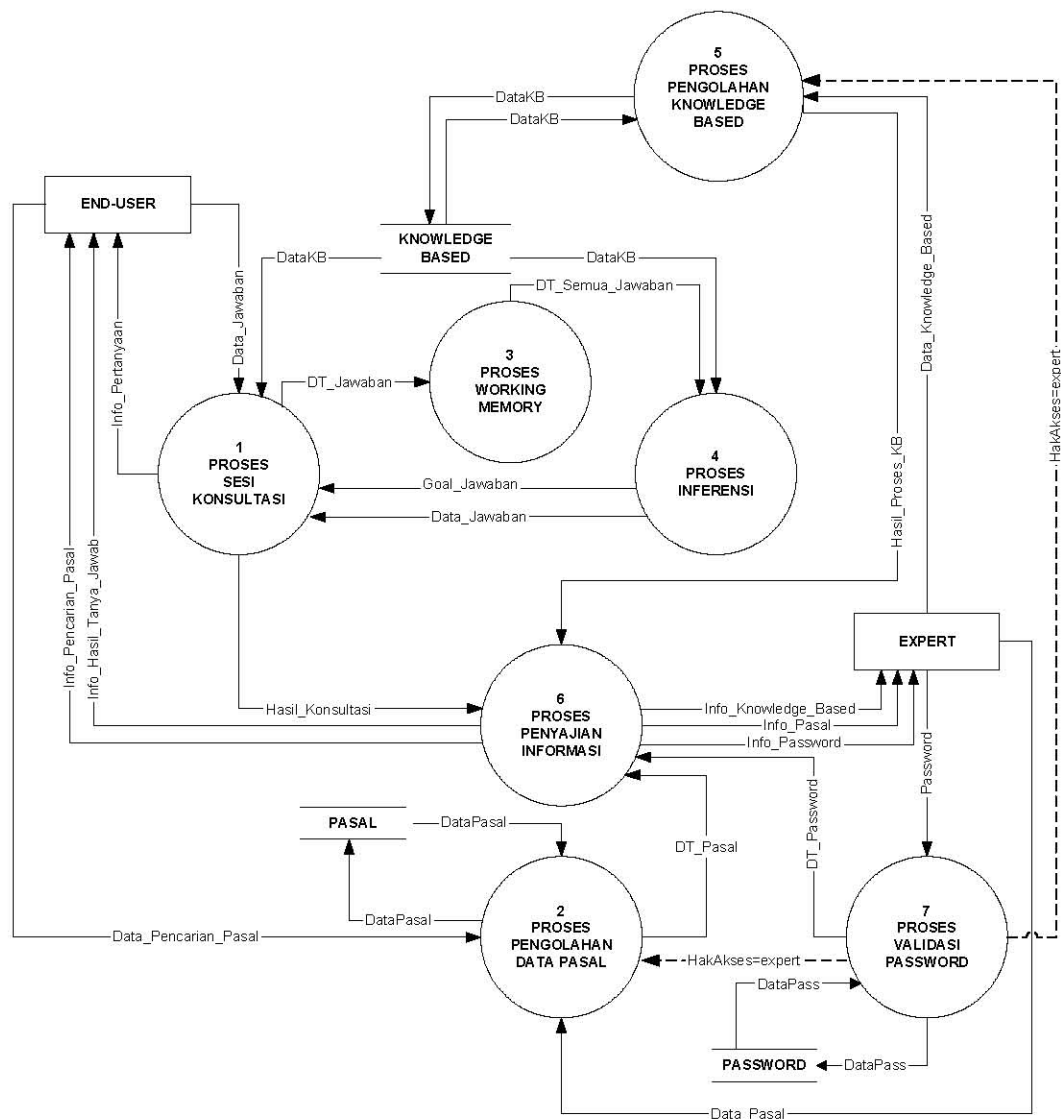
Digram konteks dibawah ini mengandung dua terminator yang masing-masing memiliki fungsi yang berbeda. Terminator END-USER merupakan terminator yang dapat melakukan pencarian pasal-pasal dan memasukan data jawaban sebagai jawaban atas pertanyaan yang diajukan oleh sistem. Sedangkan , terminator EXPERT layaknya seorang admin yang berhak memanipulasi data yang berisi basis pengetahuan dan pasal. Sebelumnya EXPERT harus melakukan validasi password untuk mengakses sistem.



Gambar 4.10 Data Context Diagram untuk Sistem Pakar

4.2.2 Data Flow Diagram Level 1

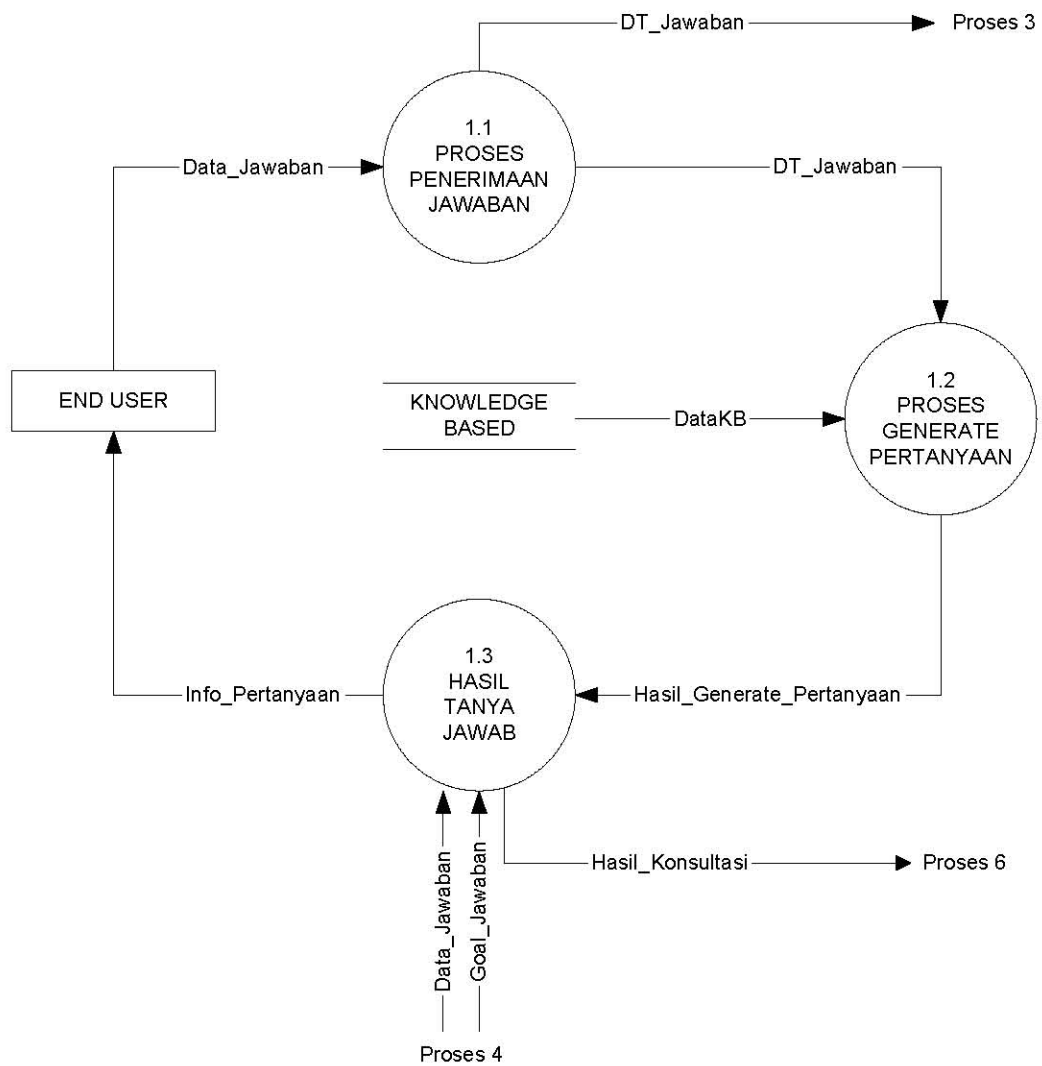
Data Flow Diagram level 1 dalam sistem pakar ini terdiri dari tujuh proses utama yang merupakan proses utama sistem yaitu proses sesi konsultasi, proses pengolahan data pasal, proses working memory, proses inferensi, proses pengolahan knowledge base dan proses validasi password serta tiga data stored yang masing-masing adalah KNOWLEDGE BASE, PASAL dan PASSWORD.



Gambar 4.11 Data Flow Diagram Level 1 untuk Sistem Pakar

4.2.3 Data Flow Diagram Level 2 Proses 1

Data Flow Diagram Level 2 untuk Proses 1 (Sesi Konsultasi) memiliki 3 proses yaitu proses menerima jawaban, proses generate pertanyaan dan proses hasil tanya jawab. Untuk memperjelas input dan output dalam DFD level 2 ini akan diuraikan dalam spesifikasi proses sesudah gambar berikut :



Gambar 4.12 Data Flow Diagram Level 2 untuk Proses 1

Spesifikasi Proses untuk Proses 1.1 yaitu proses menerima jawaban yang dimasukan oleh END USER berupa Data_Jawaban. Pada dasarnya proses ini menampung data jawaban secara temporary yaitu penyimpanan data jawaban kedalam buffer.

Tabel 4.41 Deskripsi Proses untuk Proses Penerimaan Jawaban

Nomor	1.1
Nama	Proses Penerimaan Jawaban
Input	Data_Jawaban
Output	DT_Jawaban

Keterangan Proses	Pada awalnya END USER memasukan Data_Jawaban sebagai akibat dari pertanyaan yang diajukan oleh sistem. Data yang telah dimasukan oleh END USER diterima oleh proses secara temporary sampai semua data jawaban terkumpul. Output dari proses ini adalah DT_Jawaban yang terbagi atas dua tujuan yaitu menuju proses generate pertanyaan dan menuju proses working memory.
--------------------------	---

Tabel 4.42 Deskripsi Proses untuk Proses Generate Pertanyaan

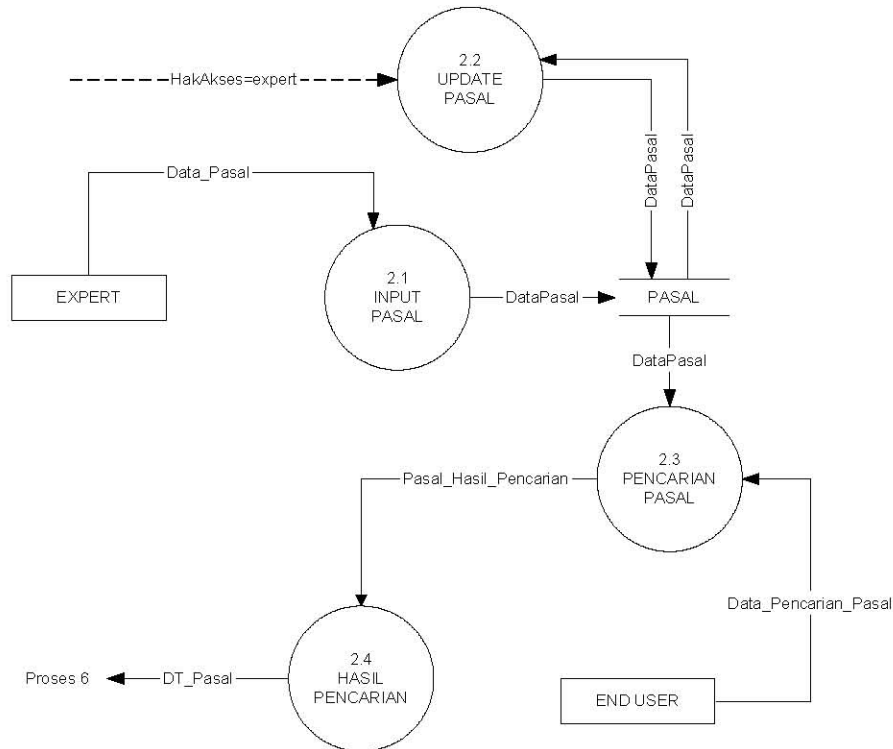
Nomor	1.2
Nama	Proses Generate Pertanyaan
Input	DT_Jawaban, DataKB
Output	Hasil_Generate_Pertanyaan
Keterangan Proses	Proses generate pertanyaan merupakan proses menyeleksi data pertanyaan yang sesuai dengan <i>rule</i> jawaban. Dalam proses ini terjadi pembacaan data atas tabel KNOWLEDGE BASED. Hasil_Generate_Pertanyaan adalah hasil dari proses ini sebagai pertanyaan dan jawaban yang telah dilakukan oleh sistem dan END USER.

Tabel 4.43 Deskripsi Proses untuk Proses Hasil Tanya Jawab

Nomor	1.3
Nama	Proses Hasil Tanya Jawab
Input	Hasil_Generate_Pertanyaan, Data_Jawaban, Gol_Jawaban.
Output	Hasil_Konsultasi, Info_Pertanyaan.
Keterangan Proses	Dalam proses ini terjadi akumulasi masukan dari proses 4, dengan Hasil_Generate_Pertanyaan yang dikeluarkan oleh proses sebelumnya. Hasil akhir dari proses ini adalah info_pertanyaan yang diberikan oleh proses kepada END USER secara langsung.

4.2.4 Data Flow Diagram Level 2 Proses 2

Data Flow Diagram Level 2 untuk Proses 2 (Pengolahan Data Pasal) terdiri dari tiga proses yaitu proses input pasal, proses pencarian pasal, proses hasil pencarian. Dalam pembahasan level ini juga terdapat satu hak akses bagi EXPERT serta dua terminator yakni EXPERT dan END USER.



Gambar 4.13 Data Flow Diagram Level 2 untuk Proses 2

DFD level 2 ini dapat diuraikan dalam tabel spesifikasi proses yang masing-masing adalah sebagai berikut :

Tabel 4.44 Deskripsi Proses untuk Proses Input Pasal

Nomor	2.1
Nama	Proses Input Pasal
Input	Data_Pasal
Output	DataPasal
Keterangan Proses	Dalam proses ini terjadi pemasukan data_pasal kedalam proses input pasal oleh EXPERT. Selanjutnya terjadi penulisan DataPasal kedalam tabel PASAL

Tabel 4.45 Deskripsi Proses untuk Proses Update Pasal

Nomor	2.2
Nama	Proses Update Pasal
Input	HakAkses=expert
Output	DataPasal
Keterangan Proses	Proses ini merupakan proses memanipulasi data pasal yang hanya dapat diakses oleh expert melalui hak akses. Penyimpanan data yang berupa tabel PASAL dapat dilakukan penulisan dan pembacaan data oleh expert.

Tabel 4.46 Deskripsi Proses untuk Proses Pencarian Pasal

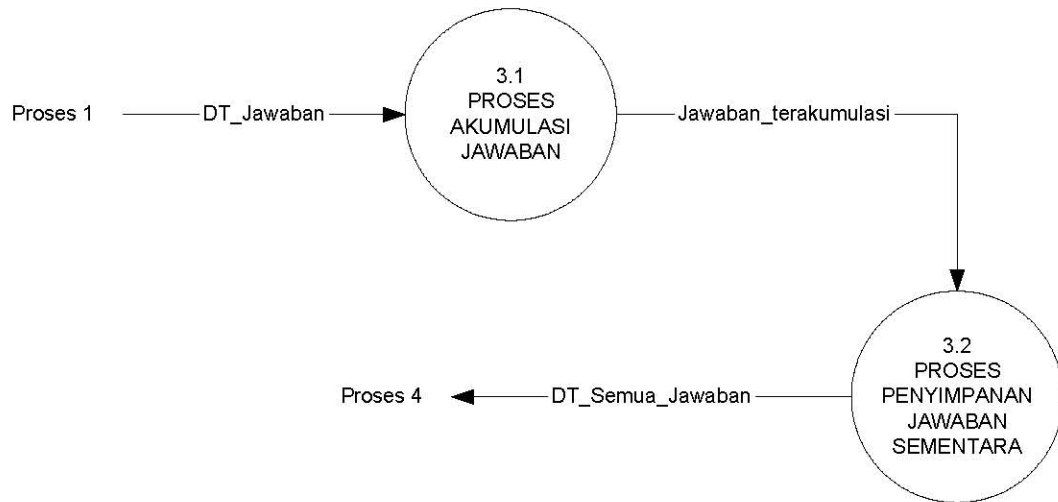
Nomor	2.3
Nama	Proses Pencarian Pasal
Input	DataPasal, Data_Pencarian_Pasal
Output	Pasal_Hasil_Pencarian
Keterangan Proses	Dalam proses ini terjadi penyesuaian antara Data_Pencarian_Pasal dan DataPasal. Data_Pencarian_Pasal merupakan kata kunci yang digunakan untuk mencari informasi. Sedangkan DataPasal adalah data yang berada dalam tabel PASAL.

Tabel 4.47 Deskripsi Proses untuk Proses Hasil Pencarian

Nomor	2.4
Nama	Proses Hasil Pencarian
Input	Pasal_Hasil_Pencarian
Output	DT_Pasal
Keterangan Proses	Proses ini merupakan proses mempersiapkan data yang akan disajikan. Data yang akan dipersiapkan adalah DT_Pasal yang notabenen merupakan output dari proses ini.

4.2.5 Data Flow Diagram Level 2 Proses 3

Data Flow Diagram Level 2 untuk Proses 3 (working memory) hanya terdiri dari dua proses yaitu proses akumulasi atau pengumpulan jawaban dan proses penyimpanan jawaban sementara kedalam memory sementara.



Gambar 4.14 Data Flow Diagram Level 2 untuk Proses 3

Penjelasan masing-masing proses dapat dilihat dalam tabel deskripsi proses berikut ini:

Tabel 4.48 Deskripsi Proses untuk Proses Akumulasi Jawaban

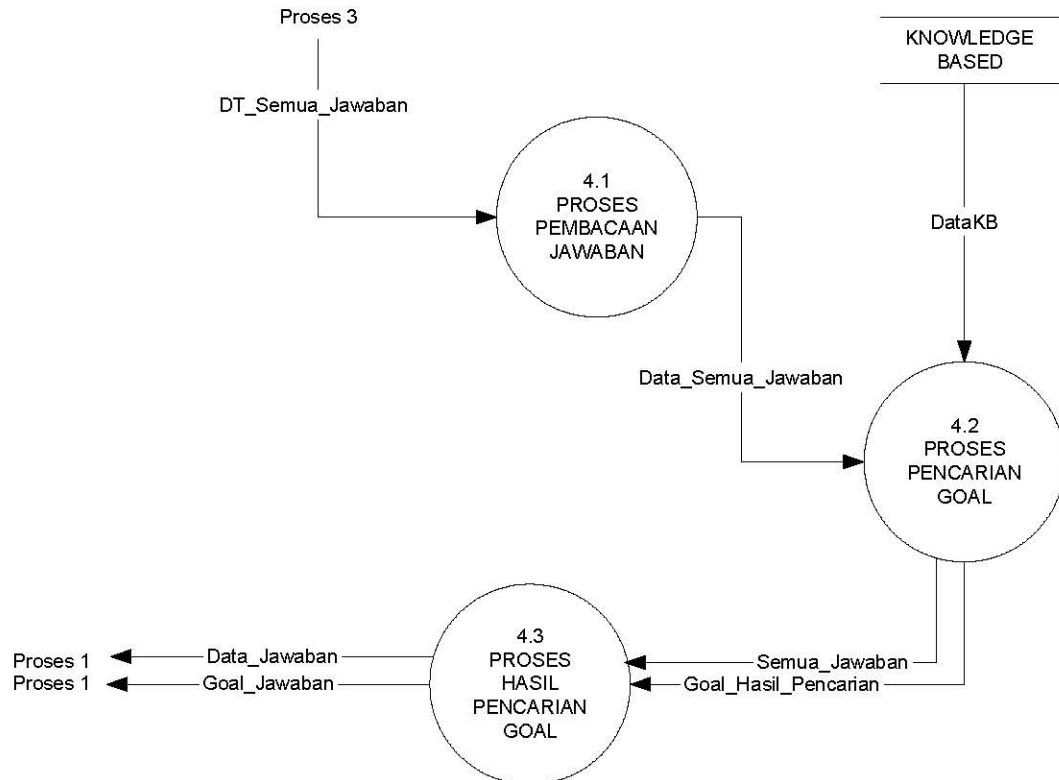
Nomor	3.1
Nama	Proses Akumulasi Jawaban
Input	DT_Jawaban
Output	Jawaban_terakumulasi
Keterangan Proses	Proses ini berfungsi untuk mengumpulkan jawaban yang diperoleh dari proses 1. Sebagai keluaran dari proses ini adalah Jawaban_Terakumulasi yang dapat dikatakan sebagai sekelompok jawaban.

Tabel 4.49 Deskripsi Proses untuk Proses Penyimpanan Jawaban Sementara

Nomor	3.2
Nama	Proses Penyimpanan Jawaban Sementara
Input	Jawaban_Terakumulasi
Output	DT_Semua_Jawaban
Keterangan Proses	Jawaban_Terakumulasi dalam proses sebelumnya kemudian dikumpulkan dalam memory sementara (buffer). DT_Semua_Jawaban adalah keluaran dari proses penyimpanan ini yang selanjutnya untuk diproses untuk menghasilkan informasi bagi END USER melalui sesi konsultasi.

4.2.6 Data Flow Diagram Level 2 Proses 4

Data Flow Diagram Level 2 Proses 4 (Proses Inferensi) terdiri dari tiga proses yakni proses pembacaan jawaban, proses pencairan gol, dan proses hasil pencarian gol serta sebuah data stored yang berupa tabel KNOWLEDGE BASED.

**Gambar 4.15 Data Flow Diagram Level 2 Proses 4**

Penjelasan untuk masing-masing proses terurai dalam sebuah tabel deskripsi proses berikut ini .

Tabel 4.50 Deskripsi Proses untuk Proses Pembacaan Jawaban

Nomor	4.1
Nama	Proses Pembacaan Jawaban
Input	DT_Semua Jawaban
Output	Data_Semua_Jawaban
Keterangan Proses	Proses ini merupakan proses membaca data yang diperoleh dari proses sebelumnya yang berupa DT_Semua_Jawaban. Keluaran dari proses ini adalah Data_Semua_Jawaban.

Tabel 4.51 Deskripsi Proses untuk Proses Pencarian Gol

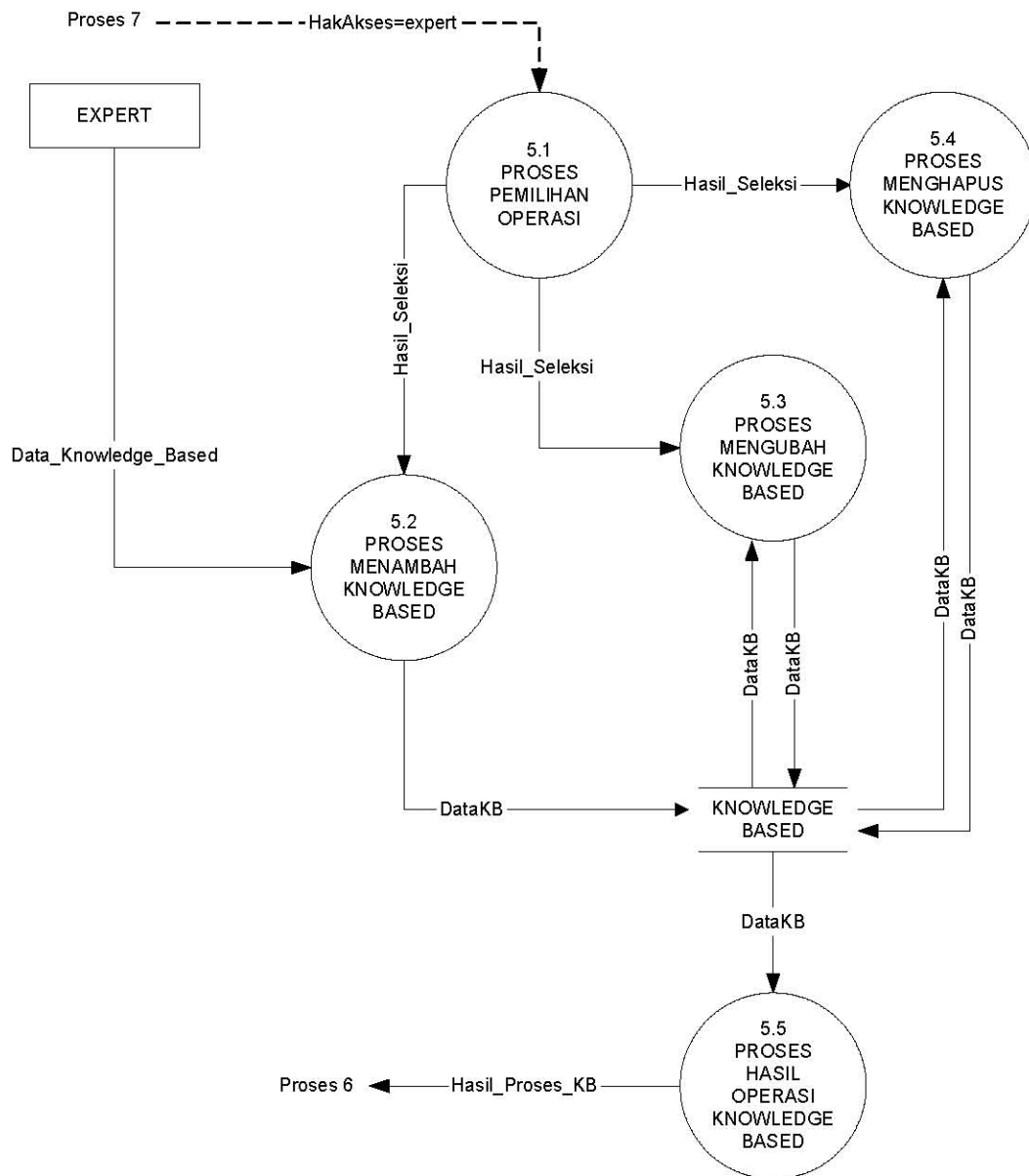
Nomor	4.2
Nama	Proses Pencarian Gol
Input	Data_Semua_Jawaban, DataKB
Output	Semua_Jawaban, Gol_Hasil_Pencarian
Keterangan Proses	Dalam proses ini terjadi pencarian gol yang dipadukan dengan jawaban yang ada dalam proses sebelumnya dengan cara membaca tabel KNOWLEDGE BASED yang berupa DataKB.

Tabel 4.52 Deskripsi Proses untuk Proses Hasil Pencarian Gol

Nomor	4.3
Nama	Proses Hasil Pencarian Gol
Input	Semua_Jawaban, Gol_Hasil_Pencarian
Output	Data_Jawaban, Gol_Jawan
Keterangan Proses	Keluaran dari proses ini adalah Data_Jawaban yang sebelumnya telah dimasukan oleh USER dan Gol_Jawaban adalah jawaban yang diperoleh oleh sistem dari tabel KNOWLEDGE BASED.

4.2.7 Data Flow Diagram Level 2 Proses 5

Data Flow Diagram Level 2 Proses 5 (Pengolahan Knowledge Base) memiliki lima proses yakni proses pemilihan operasi, proses menambah Knowledge Based, proses mengubah knowledge based, proses menghapus knowledge based.



Gambar 4.16 Data Flow Diagram Level 2 Proses 5

Kelima proses diatas dapat diuraikan dan dijelaskan dengan tabel deskripsi proses sebagai berikut:

Tabel 4.53 Deskripsi Proses untuk Proses Pemilihan Operasi

Nomor	5.1
Nama	Proses Pemilihan Operasi
Input	HakAkses=expert
Output	Hasil_Seleksi
Keterangan Proses	Hasil_Seleksi dalam proses ini merupakan hasil proses pemilihan operasi yang menjadi kontrol bagi pengoperasian fungsi. Hasil_seleksi menjadi input bagi ketiga proses selanjutnya.

Tabel 4.54 Deskripsi Proses untuk Proses Menambah Knowledge Based

Nomor	5.2
Nama	Proses Pemilihan Operasi
Input	Data_Knowledge_Based, Hasil_Seleksi
Output	dataKB
Keterangan Proses	Dalam proses ini terjadi penulisan dataKB kedalam tabel KNOWLEDGE BASED yang selanjutnya disimpan sebagai data. Proses ini juga terjadi interaksi langsung antara proses dengan terminator EXPERT yang memasukan Data_Knowledge_Based.

Tabel 4.55 Deskripsi Proses untuk Proses Mengubah Knowledge Based

Nomor	5.3
Nama	Proses Mengubah Knowledge Based
Input	Hasil_Seleksi, DataKB
Output	dataKB
Keterangan Proses	Hasil_Seleksi dalam proses ini adalah kontrol pengoperasian proses dan DataKB adalah penyimpanan dataKB kedalam tabel KNOWLEDGE BASE dengan terlebih dahulu terjadi pembacaan data dari tabel KNOWLEDGE BASED untuk melihat data yang akan diubah.

Tabel 4.56 Deskripsi Proses untuk Proses Menghapus Knowledge Based

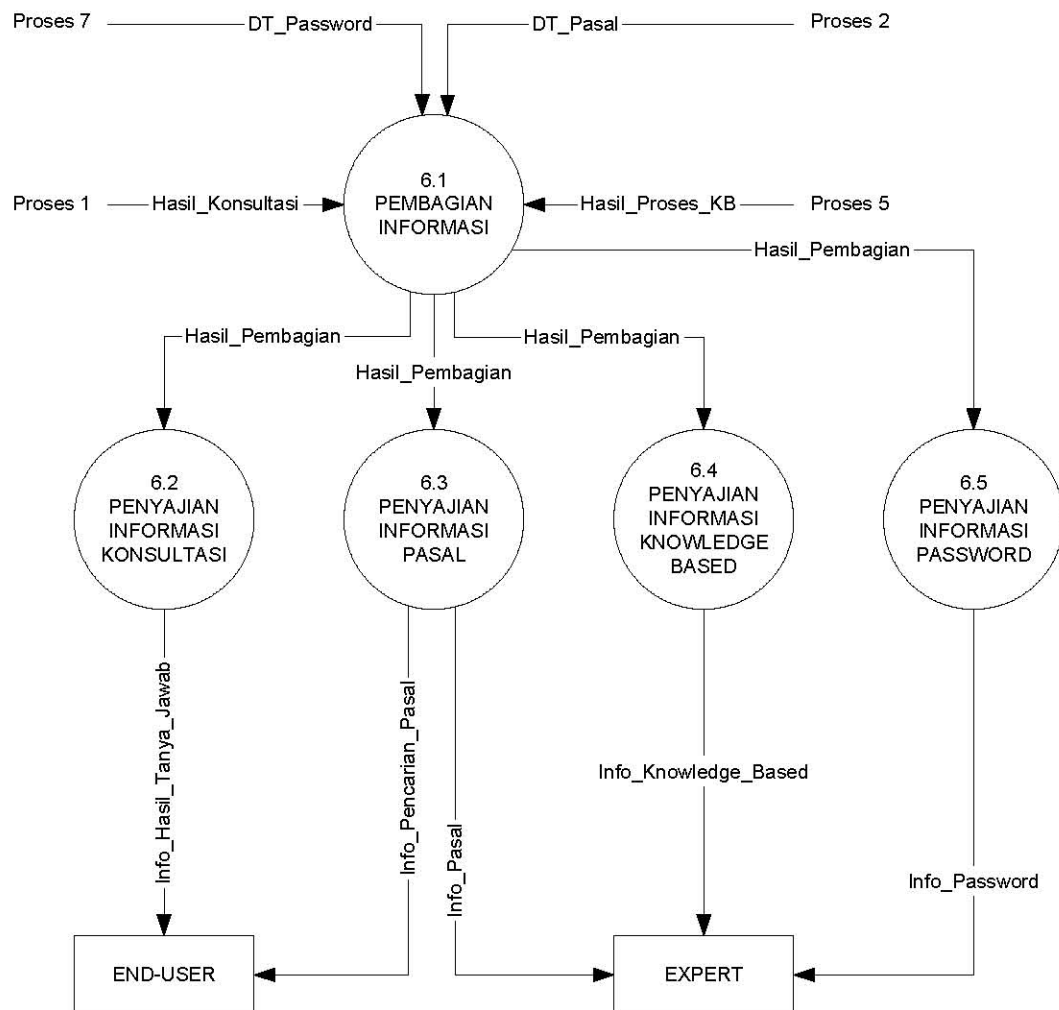
Nomor	5.4
Nama	Proses Menghapus Knowledge Based
Input	DataKB, Hasil_Seleksi
Output	DataKB
Keterangan Proses	Sama halnya dengan proses update knowledge based hanya fungsi dari proses adalah menghapus data yang diinginkan.

Tabel 4.57 Deskripsi Proses untuk Proses Hasil Operasi Knowledge Based

Nomor	5.5
Nama	Proses Hasil Operasi Knowledge Based
Input	DataKB
Output	Hasil_Proses_KB
Keterangan Proses	Fungsi dari proses ini adalah mempersiapkan informasi yang masuk (DataKB) untuk dilanjutkan dalam proses 6 melalui data yang keluar yang berupa Hasil_Proses_KB.

4.2.8 Data Flow Diagram Level 2 Proses 6

Data Flow Diagram Level 2 untuk Proses 6 (Penyajian Informasi) memiliki lima proses yakni, proses pembagian informasi, proses penyajian informasi konsultasi, penyajian informasi pasal, penyajian informasi Knowledge Based, proses penyajian informasi password.



Gambar 4.17 Data Flow Diagram Level 2 Proses 6

Penjelasan dari gambar pada halaman sebelumnya dapat dijabarkan dalam tabel deskripsi proses sebagai berikut :

Tabel 4.58 Deskripsi Proses untuk Proses Pembagian Informasi

Nomor	6.1
Nama	Proses Pembagian Informasi
Input	DT_Password, DT_Pasal, Hasil_Konsultasi, Hasil_Proces_KB
Output	Hasil_Pembagian
Keterangan Proses	Proses pembagian informasi merupakan proses menyajikan informasi kedalam sub-sub proses. Ini tergantung dari masing-masing pengguna sistem. Hasil_Pembagian yang merupakan keluaran sistem dilanjutkan ke empat proses yang lebih spesifik.

Tabel 4.59 Deskripsi Proses untuk Proses Penyajian Informasi Konsultasi

Nomor	6.2
Nama	Proses Penyajian Informasi Konsultasi
Input	Hasil_Pembagian
Output	Info_Hasil_Tanya_Jawab
Keterangan Proses	Input Hasil_Pembagian ini diproses untuk menghasilkan suatu keluaran yang berupa Info_Hasil_Tanya_Jawab. Proses ini merupakan proses yang dibutuhkan untuk menyajikan informasi pada sesi konsultasi.

Tabel 4.60 Deskripsi Proses untuk Proses Penyajian Informasi Pasal

Nomor	6.3
Nama	Proses Penyajian Informasi Pasal
Input	Hasil_Pembagian
Output	Info_Pencarian_Pasal, Info_Pasal
Keterangan Proses	Dalam proses ini input hasil_pembagian diteruskan kedalam proses untuk menghasilkan informasi yang disesuaikan dengan data pencarian pasal yang berupa Info_Pencarian_Pasal dan informasi pasal. Dalam proses ini juga terjadi interaksi langsung antara proses dengan Info_Hasil_Tanya_Jawab.

Tabel 4.61 Deskripsi Proses untuk Penyajian Informasi Knowledge Based

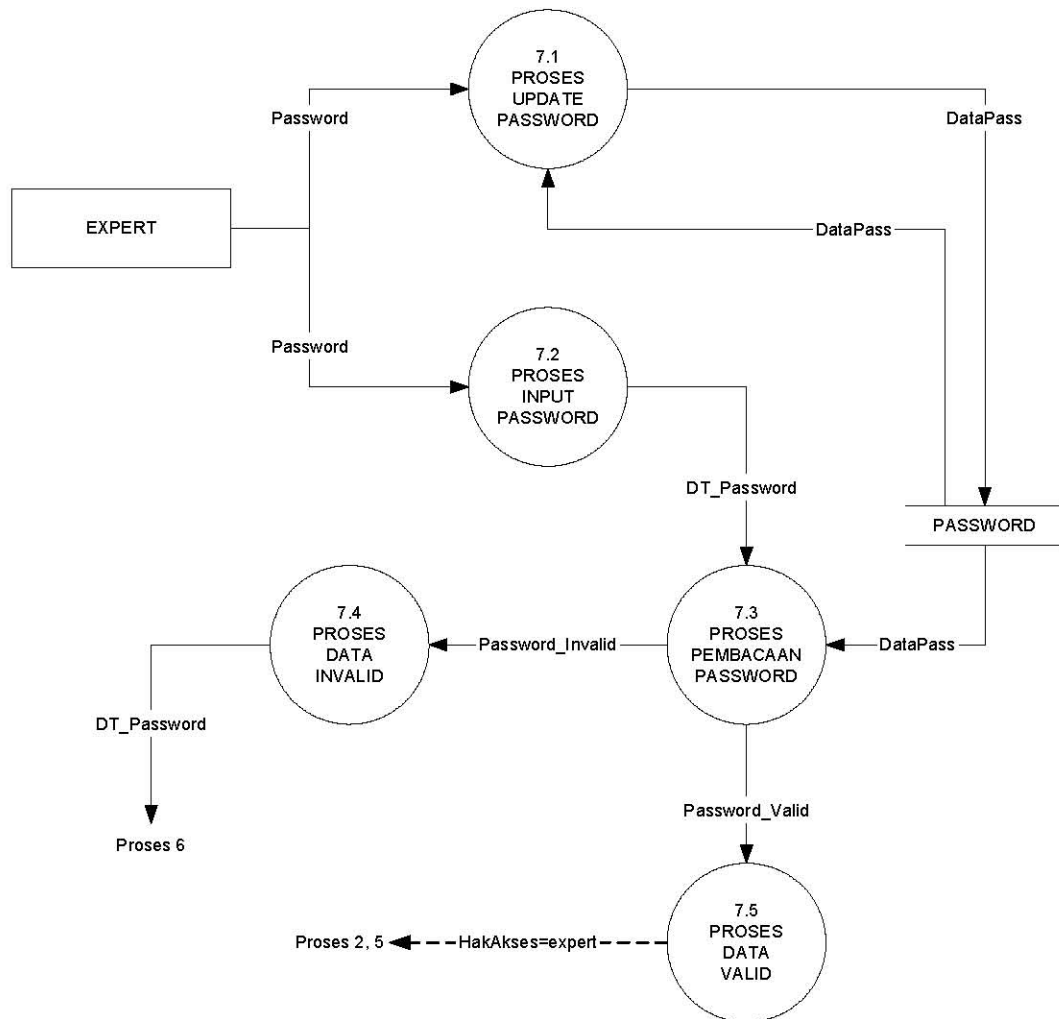
Nomor	6.4
Nama	Proses Penyajian Informasi Pasal
Input	Hasil_Pembagian
Output	Info_Knowledge_Based
Keterangan Proses	Proses ini merupakan proses menyajikan informasi Knowledge based yang menjadi hak akses bagi expert. Informasi ini berupa basis pengetahuan yang memungkinkan melihat informasi dengan lebih spesifik.

Tabel 4.62 Deskripsi Proses untuk Proses Penyajian Informasi Password

Nomor	6.5
Nama	Proses Penyajian Informasi Password
Input	Hasil_Pembagian
Output	Info_Password
Keterangan Proses	Proses ini merupakan proses menyajikan informasi Password expert sebagai keluaran dari proses yang dilakukan pada proses sebelumnya yaitu proses 7 yang didalamnya melibatkan proses validasi password.

4.2.9 Data Flow Diagram Level 2 Proses 7

Data Flow Diagram Level 2 Proses 7 (Validasi Password) terdiri dari lima proses yang masing-masing adalah proses update password, proses input password, proses pembacaan password, proses data invalid, proses data valid.



Gambar 4.18 Data Flow Diagram Level 2 Proses 7

Penjelasan dari masing-masing proses dijelaskan dalam tabel deskripsi proses berikut ini.

Tabel 4.63 Deskripsi Proses untuk Proses Update Password

Nomor	7.1
Nama	Proses Update Password
Input	Password
Output	DataPass
Keterangan Proses	Dalam proses ini terjadi perubahan data password oleh EXPERT. Proses ini membaca data dari tabel PASSWORD dan menulis kembali data password kedalam tabel sebagai hasil dari proses update ini.

Tabel 4.64 Deskripsi Proses untuk Proses Input Password

Nomor	7.2
Nama	Proses Input Password
Input	Password
Output	DT_Password
Keterangan Proses	Proses Input password ini bukan berarti proses memasukna data kedalam tabel password. Tapi lebih tepat proses menangkap data yang dilakukan oleh buffer memory, yang kemudian dilanjutkan keproses pembacaan password.

Tabel 4.65 Deskripsi Proses untuk Proses Pembacaan Password

Nomor	7.3
Nama	Proses Pembacaan Password
Input	DT_Password
Output	Password_Invalid, Password_Valid
Keterangan Proses	Uraian proses ini adalah terjadi pembacaan data password yang berada dalam tabel Password. Kemudian data tersebut dicocokkan dengan data yang ada di buffer memory. Keluaran proses ini adalah Password_Invalid jika data tidak cocok, sementara Password_Valid jika data cocok dengan data yang ada dalam tabel Password.

Tabel 4.66 Deskripsi Proses untuk Proses Data Invalid

Nomor	7.4
Nama	Proses Data Invalid
Input	Password_Invalid
Output	DT_Password
Keterangan Proses	Proses ini memproses password yang tidak cocok dengan masukan user untuk kemudian dilanjutkan keproses selanjutnya untuk menciptakan pesan bagi pengguna

Tabel 4.67 Deskripsi Proses untuk Proses Data Valid

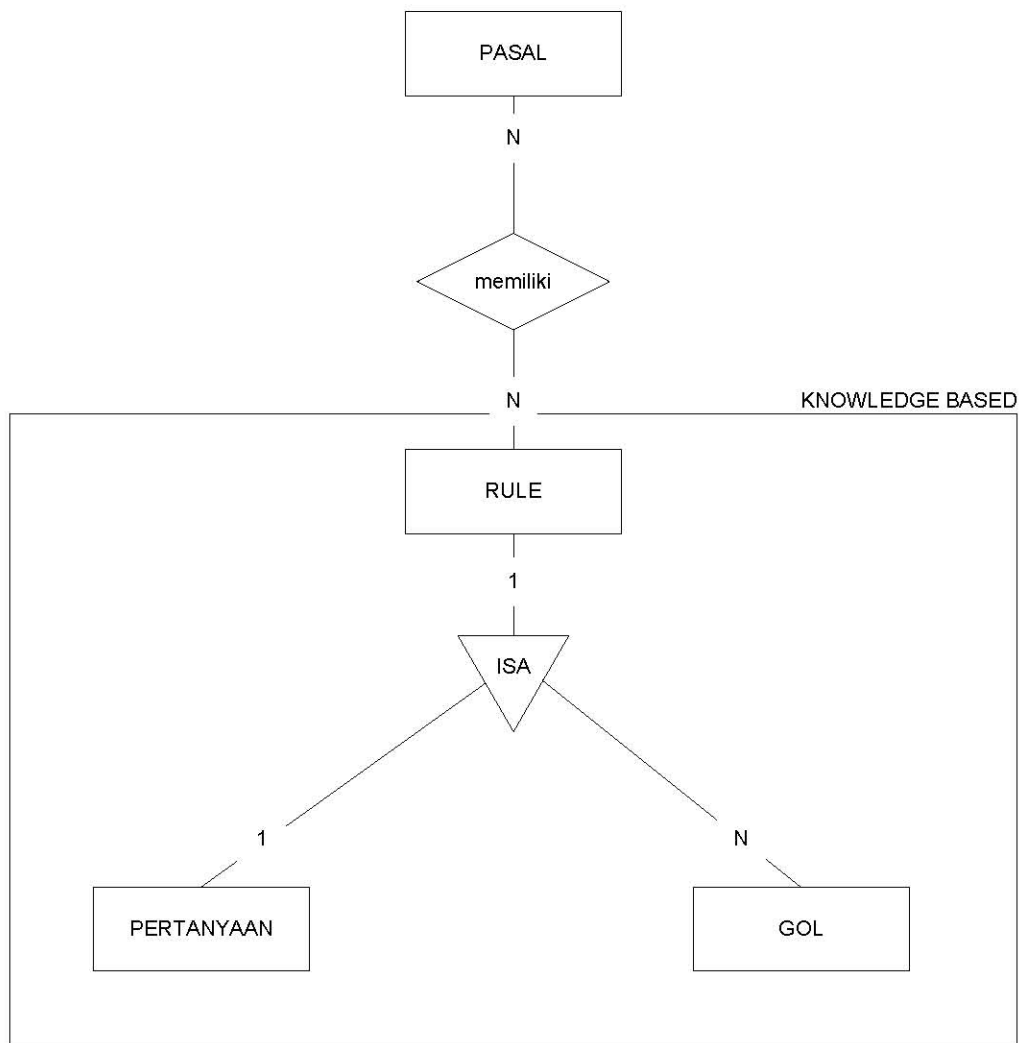
Nomor	7.5
Nama	Proses Data valid
Input	Password_valid
Output	Hak Akses=expert
Keterangan Proses	Proses ini dipicu oleh password yang cocok dengan data yang berada dalam tabel password (Password_Valid). Untuk kemudian diperoleh hakakses bagi expert untuk melakukan operasi-operasi yang diinginkan sesuai dengan proses yang ada.

4.3 Perancangan Data

Perancangan data dimanfaatkan untuk memperoleh struktur file yang ada dalam data dan relationship antar tabel. Hasil akhir dari proses ini adalah *entity relationship diagram* dengan kamus data (*data dictionary*).

4.3.1 *Entity Relationship Diagram*

Berikut ini digambarkan relasi antar tabel yang dipergunakan dalam sistem pakar dengan menggunakan *Entity Relationship Diagram*.



Gambar 4.19 *Entity Relationship Diagram*

4.3.2 Kamus Data (*data dictionary*)

Kamus data yang digunakan dalam perancangan aliran data diperoleh istilah-istilah yang akan diuraikan dengan tabel sebagai berikut.

Tabel 4.68 Kamus Data

No	Data	Kamus Data
1	Data_Jawaban	/*data jawaban yang dimasukan end user sebagai jawaban atas pertanyaan sistem*/
2	DT_Jawaban	/*data jawaban yang mengalir antar proses*/
3	DataKB	/*data knowledge based*/
4	Knowledge Based	[KD_Rule + Nama_Rule + Variabel + Pertanyaan + Gol]
5	Info_Pertanyaan	/*informasi pertanyaan*/
6	Hasil_Generate_Pertanyaan	/*hasil generate pertanyaan*/
7	Hasil_Konsultasi	/*hasil keseluruhan sesi konsultasi*/
8	Data_Jawaban	/*data yang dimasukan oleh end user sebagai jawaban*/
9	Gol_Jawaban	/*gol yang didapat setelah sesi konsultasi berakhir*/
10	HakAkses=expert	/*hak istimewa bagi expert dalam mengakses sistem*/
11	DataPasal	/*data pasal yang mengalir dari dan ke tabel pasal*/
12	Pasal	[KD_Pasal + Nama_Pasal + Isi_Pasal + KD_Variabel]
13	Pasal_Hasil_Pencarian	/*pasal yang diperoleh dari hasil pencarian*/
14	DT_Pasal	/*bentuk data yang berasal dari tabel pasal*/
15	Data_Pencarian_Pasal	/*data hasil pencarian pasal*/

No	Data	Kamus Data
16	Jawaban_Terakumulasi	/*data jawaban yang terkumpul*/
17	Semua_Jawaban	/*semua jawaban yang dimasukan end user atas interaksi dengan sistem*/
18	Gol_Hasil_Pencarian	/*gol hasil pencarian*/
19	Data_Knowledge_Based	/*data yang berada dalam knowledge based*/
20	Hasil_Seleksi	/*hasil penyeleksian operasi*/
21	Hasil_Proses_KB	/*hasil pemrosesan knowledge based*/
22	DT_Password	/*Data Password yang ditransfer antar proses*/
23	Password	[IDPass + Nama_Expert + Password]
24	Hasil_Pembagian	/*kontrol bagi pembagian informasi*/
25	Info_Hasil_Tanya_Jawab	/*informasi yang ditampilkan setelah proses tanya jawab berakhir*/
26	Info_Pencarian_Pasal	/*informasi hasil pencarian pasal*/
27	Info_Pasal	/*informasi pasal*/
28	Info_Knowledge_Based	/*informasi knowledge based*/
29	DataPass	/*bentuk data yang dikeluarkan oleh tabel password*/
30	Data_Invalid	/*data password yang tidak sesuai dengan data password yang ada dalam tabel password*/
31	Data_Valid	/*data password yang sama dengan data yang ada dalam tabel pasword*/

4.4 Perancangan Antar Muka

Perancangan antarmuka menjelaskan rutinitas program yang akan dijalankan oleh sebuah sistem komputerisasi untuk menjelaskan interaksi antara pemakai dengan program yang dibuat.

The screenshot shows the main menu of the 'SISTEM PAKAR UNDANG-UNDANG HAK CIPTA'. At the top, the title is centered. Below the title, there are five rounded rectangular buttons arranged horizontally: 'KONSULTASI', 'SESI', 'PASAL', 'KNOWLEDGE BASED', and 'KELUAR'.

Gambar 4.20 Form Utama Sistem Pakar

The screenshot shows the 'SESI KONSULTASI' (Consultation Session) form. At the top, the title is centered. Below the title, there is a large rectangular text input field labeled 'Pertanyaan :'. Below this field, there is a section labeled 'Jawaban' (Answer) containing two radio button options: 'Tidak' (No) and 'Benar' (Yes). To the right of these options, there are two rounded rectangular buttons: 'Lanjut' (Next) and 'Sesi Selanjutnya' (Next Session).

Gambar 4.21 Form Sesi Konsultasi

SISTEM PAKAR UNDANG-UNDANG HAK CIPTA	
HASIL KONSULTASI	
Working Memory	Hasil Konsultasi

Gambar 4.22 Form Hasil Konsultasi

MENU UTAMA EXPERT	
Username	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="EXPERT"/> <input type="button" value="NON-EXPERT"/>	

Gambar 4.23 Form Login Expert

**MENU UTAMA
EXPERT**

KD_Rule	Nama_Rule	TRUE	FALSE	Goal

Gambar 4.24 Form Update Rule

**MENU UTAMA
EXPERT**

KD_Pasal	Nama_Pasal	Isi_Pasal

Gambar 4.25 Form Pencarian Pasal

BAB V

IMPLEMENTASI SISTEM

5.1 Lingkungan Implementasi

Sistem Pakar yang dikembangkan ini diimplementasikan dalam lingkungan–lingkungan yang terkait dengan kondisi pengguna saat ini. Adapun pembagian lingkungan implementasi meliputi lingkungan perangkat keras (*hardware*) dan perangkat lunak (*software*).

5.1.1 Lingkungan Perangkat Keras (*hardware*)

Spesifikasi perangkat keras yang diperlukan pada saat implementasi sistem pakar ini adalah sebagai berikut :

Minimal requirement

- a. Processor : intel pentium III 533 MHZ
- b. Memori (RAM) : 128 MB
- c. Harddisk : 5 GB
- d. Monitor : 14 “

5.1.2 Lingkungan Perangkat Lunak (*software*)

Spesifikasi perangkat lunak (*software*) yang digunakan untuk membangun sistem ini adalah sebagai berikut :

- a. Sistem operasi : Windows XP
- b. Program aplikasi : Borland Delphi 7.0
- c. Basis data : Paradox 7.0

5.2 Implementasi Basis Data (*Database*)

Dalam sistem pakar ini implementasi basis data dimaksudkan untuk menyimpan basis pengetahuan. Hal dilakukan karena pemanfaatan bahasa prosedural sebagai *developer tools*. Adapun tabel-tabel yang ada dalam sistem pakar ini terbagi atas empat tabel, yaitu :

1. Tabel Pasal, berisi data tentang pasal-pasal yang ada dalam undang-undang hak cipta.

Tabel 5.1 Tabel Pasal

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_Pasal	Numeric		Kode Pasal
2	Nama_Pasal	Alpha	50	Nama Pasal
d	Indeks_Pasal	Alpha	50	Indeks_Pasal
4	Isi_Pasal	Memo	150	Isi Pasal

2. Tabel Knowledge_Based, berisi informasi yang akan diolah oleh mesin inferensi.

Tabel 5.2 Tabel Knowledge_Based

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_Rule	Alpha	6	Kode Rule
2	Kode_Pertanyaan	Alpha	4	Kode_Pertanyaan
5	Kode_Gol	Numeric	4	Kode Pertanyaan

3. Tabel Rule, berisi informasi aturan kondisi benar salah yang akan diolah oleh mesin inferensi.

Tabel 5.3 Tabel Rule

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_Rule	Alpha	6	Kode_Rule
2	Kode_Pertanyaan	Alpha	4	Kode_Pertanyaan
3	Kondisi_Benar	Alpha	4	Kode_Benar
4	Kondisi_Salah	Alpha	4	Kode_Salah

4. Tabel Gol, berisi data tentang sanksi-sanksi yang ada dalam undang-undang hak cipta ini.

Tabel 5.4 Tabel Gol

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_Gol	Numeric		Kode Gol
2	Isi Gol	Alpha	50	Isi Gol

5. Tabel Pertanyaan, berisi data tentang pertanyaan yang digenerate dari jawaban tertentu.

Tabel 5.5 Tabel Pertanyaan

No	Nama Field	Tipe Data	Ukuran	Keterangan
1	Kode_Pertanyaan	Numeric		Kode Knowledge Based
2	Isi_Pertanyaan	Alpha	50	Kode_Gol
3	Kode_Gol	Memo	250	Kode Pertanyaan

5.3 Implementasi Proses

Dalam perancangan sistem ada beberapa proses yang mendukung sistem pakar ini, sebagai implementasi dari proses tersebut dapat dilihat pada algoritma berikut ini:

1. Proses Sesi Konsultasi

Procedure Sesi Konsultasi

{I.S. data pertanyaan ditampilkan untuk dapat dikorespondensikan dengan jawaban yang akan di inputkan oleh pengguna sistem (*end user*)}

{F.S menerima data jawaban untuk selanjutnya disimpan dalam buffer memory. Data tersebut akan disesuaikan dengan data pertanyaan yang ada dalam tabel knowledge based untuk menghasilkan suatu kesimpulan dari proses sesi konsultasi ini}

Kamus Lokal

Pertanyaan,jawaban,temp : string

I: integer

Working_memory : string

Knowledge_Based : tabel

Algoritma

Open (Knowledge_Based)

Output (informasi)

Repeat

 Output (pertanyaan)

 Input (jawaban)

 For $i \leftarrow 1$ to Knowledge_Based.EOF do

 Temp \leftarrow pertanyaan

 endfor

Output(temp)

Until Gol terpenuhi

Working memory \leftarrow (temp + jawaban + gol)

Close (Knowledge_Based)

2. Proses Pengolahan Data Pasal

Procedure PengolahanData Pasal

{I.S. menampilkan menu program dengan dua fungsi pilihan yaitu update pasal dan input pasal tergantung dari hak akses pengguna(*end user*)}

{F.S menampilkan data hasil pencairan yang dilakukan oleh pengguna sistem serta menampilkan data hasil input data pasal}

Kamus Lokal

I : integer,Pasal:tabel

Algoritma

Input (hak akses)

If (hak akses = "Expert") then

 Output (kontrol)

 Input(operasi)

 Depend operasi on

 Operasi ="Update Pasal" : Open database

 Output (pasal)

 Update (pasal)

 Close database

 Operasi="Input Pasal" : Open database

 Input (pasal)

 Output (pasal)

 Close database

 End depend

Else if (hak akses="End User") then

 Input (pasal)

 Repeat

 Keyword \leftarrow pasal

 For (i \leftarrow 1 to pasal.EOF) do

 Output (pasal.ayat)

 Endfor

 Until ayat ketemu

End if

3. Proses Working Memory

Procedure Working Memory

{I.S. membaca data jawaban dari proses sesi konsultasi. Hasil pembacaan terhadap data jawaban disimpan dalam buffer memory}

{F.S terakumulasinya data jawaban dan kemudian disimpan dalam tabel sementara. Keluaran dari program ini adalah Data semua jawaban yang telah diinput oleh user}

Kamus Lokal

DT_Jawaban, Jawaban_Terakumulasi :string

Algoritma

Input (DT_Jawaban)

If (DT_Jawaban <> "") then

Repeat

 Jawaban_Terakumulasi ← Jawaban_Terakumulasi + DT_Jawaban

 Until Jawaban_Terakumulasi terpenuhi

 Proses Inferensi

Else

Output ("Masukan Data Jawaban")

Endif

4. Proses Inferensi

Procedure Inference

{I.S. membaca data semua jawaban yang diperoleh dari proses working memory}
{F.S menghasilkan gol jawaban dan data jawaban yang telah diinput oleh pengguna}

Kamus Lokal

DT_Semua_Jawaban, Data_Jawaban, Goal_Jawaban:string

Algoritma

Open database

Input (DT_Semua_Jawaban)

Repeat

Read(DT_Semua_Jawaban)

Hasil ← select Gol from Knowledge_Based where Jawaban=DT_Semua_Jawaban

Data_Jawaban ← DT_Semua_Jawaban

Goal_Jawaban ← Hasil

Until goal terpenuhi

Output (Data_Jawaban, Goal_Jawaban)

Close database

5. Proses Pengolahan Knowledge Based

Procedure Pengolahan Knowledge Based

{I.S. ditampilkan tiga kontrol yang memungkinkan seorang expert memilih operasi yang dilakukan}

{F.S keluaran dari proses ini menghasilkan informasi hasil operasi yang dipilih}

Kamus LokalAlgoritma

Open (Knowledge_Based)

Read (operasi)

Depend operasi on

Operasi="Tambah" : Input DataKB

Output (Hasil_Proses_KB)

Operasi="Update" : Update DataKB

Output (Hasil_Proses_KB)

Operasi="Hapus" : Delete DataKB

Output (Hasil_Proses_KB)

enddepend

Close(Knowledge_Based)

6. Proses Penyajian Informasi

Procedure Penyajian Informasi

{I.S. membaca hak akses dari masing-masing pengguna}

{F.S keluaran dari proses ini menghasilkan informasi yang ditentukan oleh hak akses pengguna}

Kamus Lokal

Info_Hasil_Tanya_Jawab, Info_Hasil_Pencarian, Info_Pasal,
Info_Knowledge_Based, Info_Password :String

Algoritma

Read (hak akses)

If (hak akses="End User") then

Output (Info_Hasil_Tanya_Jawab)

Output (Info_Pencarian_Pasal)

Else

If (hak akses="Expert") then

Output (Info_Pasal)

Output (Info_Knowledge_Based)

Output (Info_Password)

endif

7. Proses Validasi Password

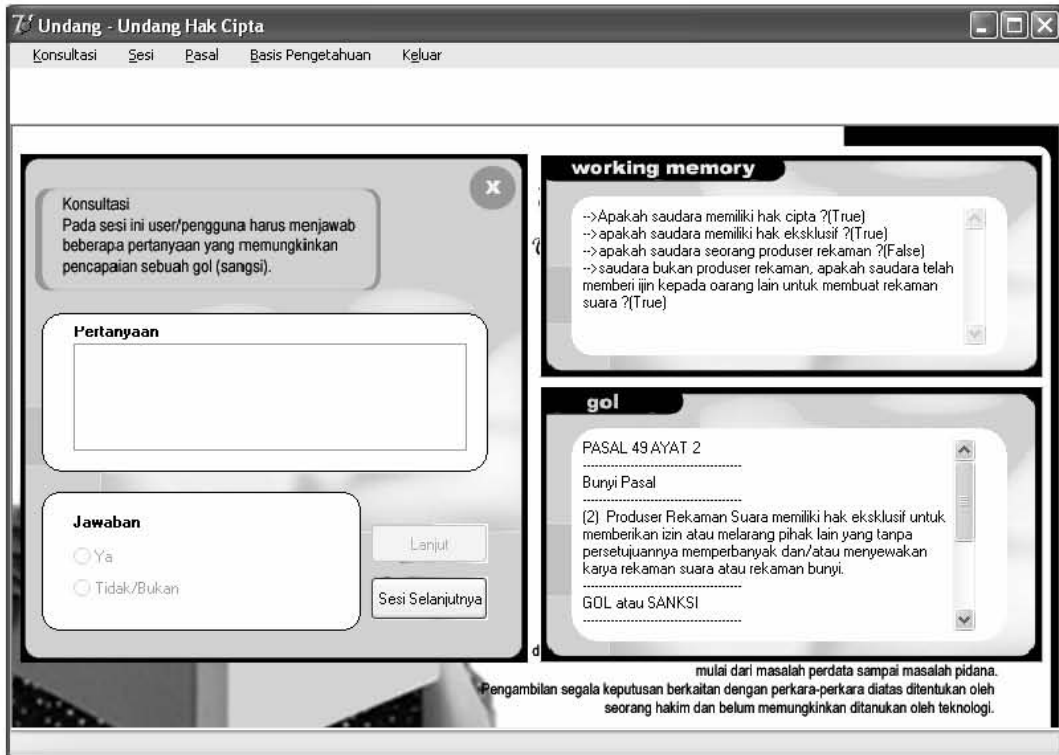
<p><u>Procedure</u> Penyajian Informasi</p> <p>{I.S. membaca data password}</p> <p>{F.S informasi hasil input ataupun hasil update password yang berupa DT_Password}</p>
<p><u>Kamus Lokal</u></p> <p>Password :string</p>
<p><u>Algoritma</u></p> <p>Open password</p> <p>Input (password)</p> <p>If (password=password.pass) then</p> <p> Update (password)</p> <p> Write (password)</p> <p>Else</p> <p> Pesan (“ Password yang anda masukan salah”)</p> <p>End if</p> <p>Close password</p>

5.4 Implementasi Antarmuka

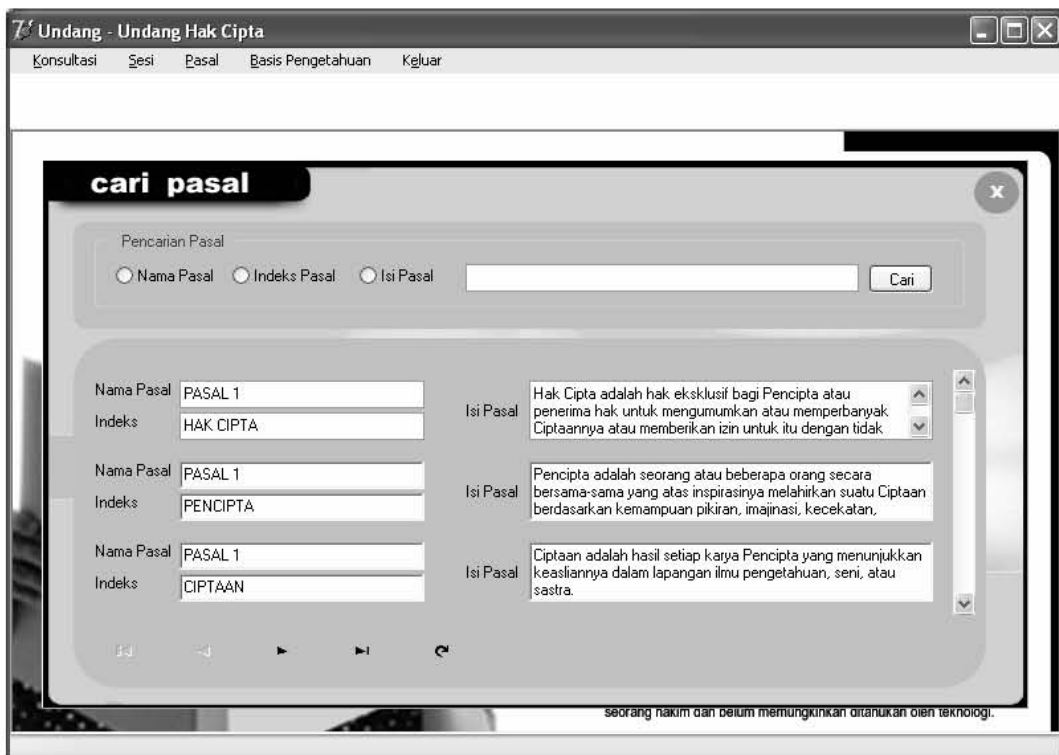
Pada tahap perancangan telah dijelaskan rancangan antarmuka yang digunakan dalam sistem ini. Dari hasil perancangan tersebut dibawah ini merupakan implementasi dari rancangan antarmuka.



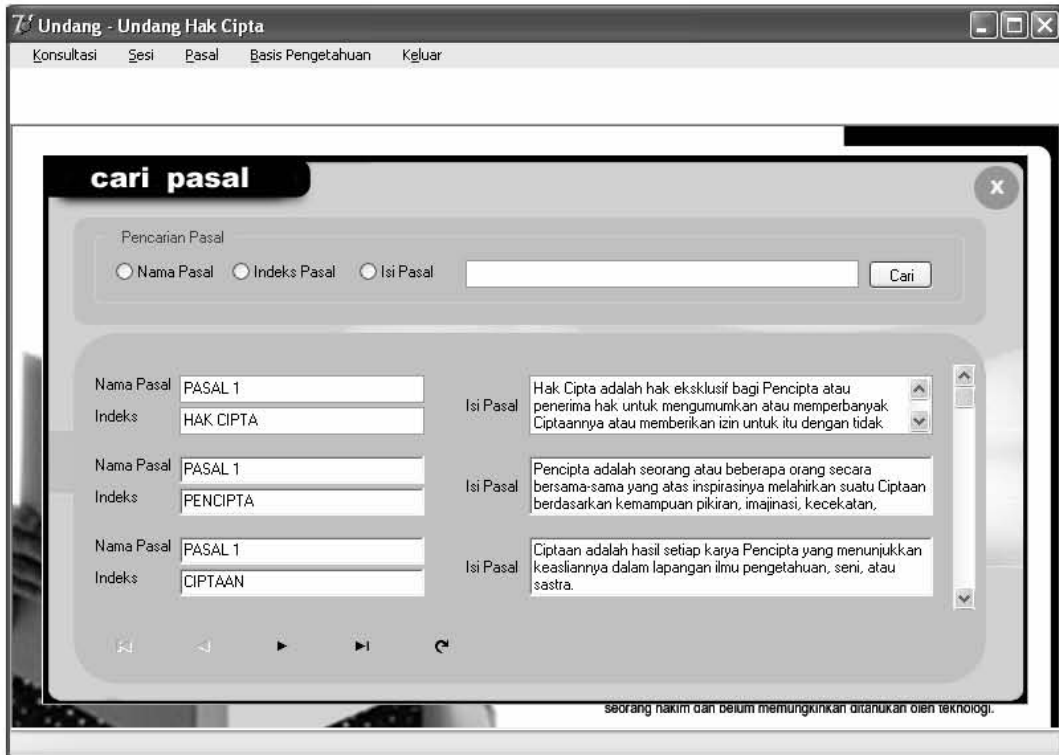
Gambar 5.1 Form Utama



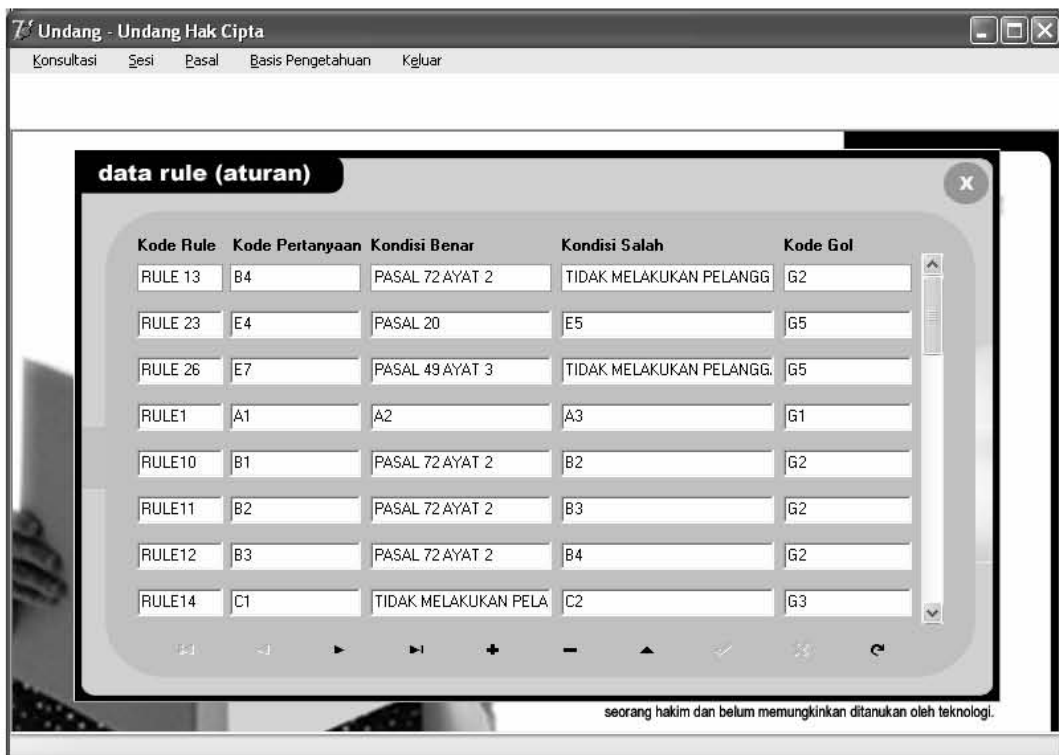
Gambar 5.2 Sesi Konsultasi



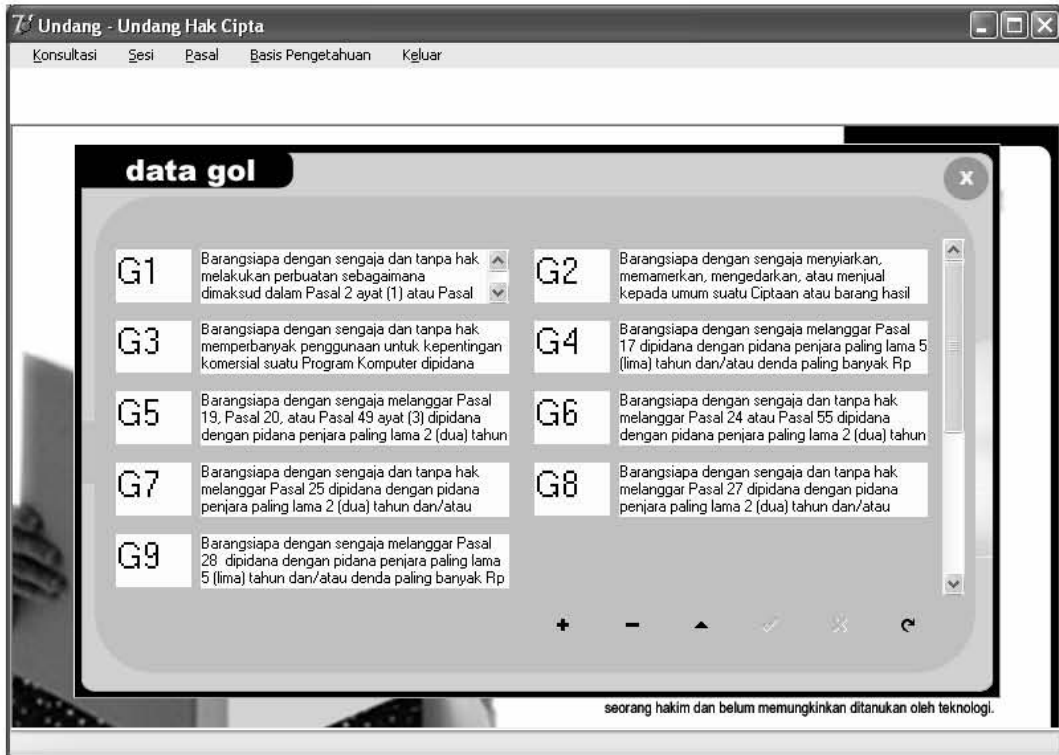
Gambar 5.3 Pencarian Pasal



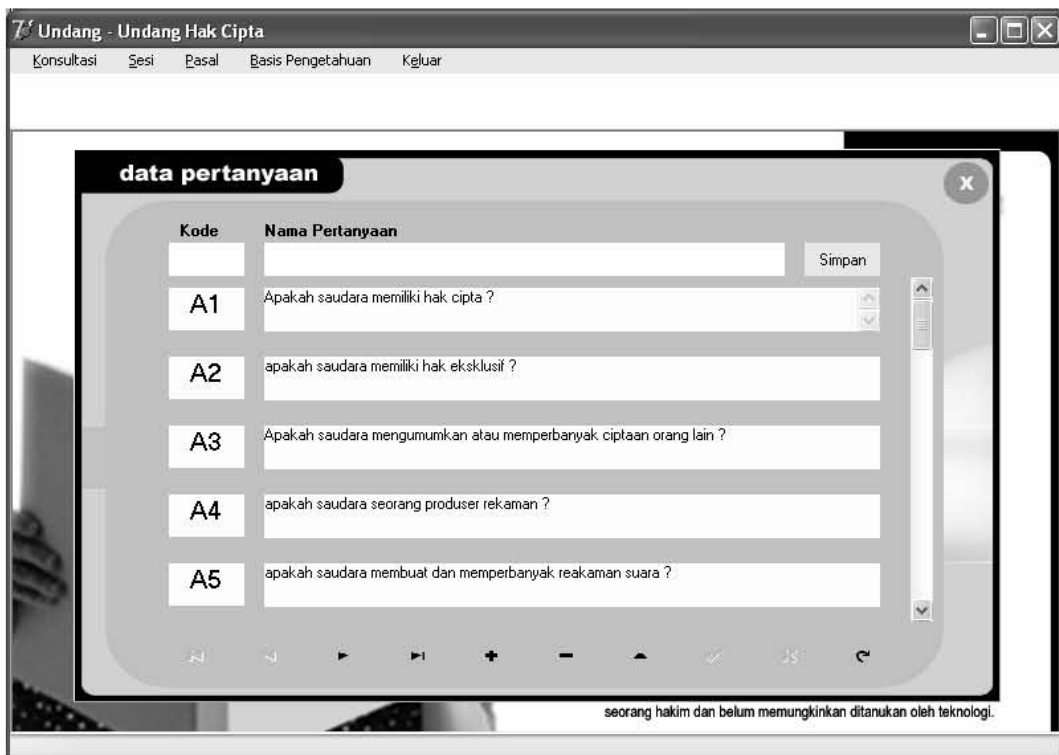
Gambar 5.4 Update Pasal



Gambar 5.5 Form Data Rule



Gambar 5.6 Form Data Gol



Gambar 5.7 Form Data Pertanyaan

BAB VI

PENUTUP

Sistem Pakar pada Undang-Undang Republik Indonesia nomor 19 tentang Hak Cipta (PRO-ES) ini diharapkan dapat menjadi bahan atau salah satu referensi bagi pengembangan sistem pakar lainnya atau bagi mahasiswa dalam menyusun tugas akhir yang berkaitan dengan sistem pakar. Ada beberapa kesimpulan dan saran yang dapat disampaikan sebagai hasil dari evaluasi pengembangan sistem dalam laporan tugas akhir ini.

6.1 Kesimpulan

Berdasarkan permasalahan yang telah dibahas dan diselesaikan melalui laporan ini, maka terdapat beberapa kesimpulan yaitu :

- a. Sistem pakar ini memudahkan pencarian pakar dan menentukan sanksi-sanksi yang terdapat dalam undang-undang tersebut.
- b. Sistem pencarian buku yang selama ini dilakukan oleh para pelaku hukum atau pengacara masih dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama. Dengan sistem pakar yang dikembangkan ini akan lebih efektif dan efisien.
- c. Sistem pakar ini dikembangkan dengan menggunakan bahasa pemrograman prosedural (*Delphi 7.0*) karena penulis merasa tertarik dan ingin belajar lebih banyak tentang bahasa pemrograman ini.

6.2 Saran

Setelah mengembangkan sistem pakar ini, ada beberapa saran yang harus diterapkan guna pengembangan sistem lebih lanjut.

- a. Sistem pakar ini dapat dijadikan bahan masukan bagi pengembangan sistem pakar yang lebih kompleks.
- b. Dalam pengembangan sistem pakar khususnya perancangan basis pengetahuan, hendaknya ditangani oleh seorang yang ahli dalam merumuskan pengetahuan yaitu *knowledge engineering*.