

**APLIKASI UNTUK MENDIAGNOSA PENYAKIT JANTUNG
PADA MANUSIA**



SKRIPSI

Oleh :

INDRIAS ROSIAWATI

J2A 605 061

**PROGRAM STUDI MATEMATIKA JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

UNIVERSITAS DIPONEGORO

SEMARANG

2010

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga Tugas Akhir ini dapat diselesaikan. Tugas Akhir yang berjudul “**APLIKASI UNTUK MENDIAGNOSA PENYAKIT JANTUNG PADA MANUSIA**” disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam di Universitas Diponegoro.

Banyak pihak yang telah membantu dalam penyelesaian Tugas akhir ini. Oleh karena itu, rasa hormat dan terima kasih penulis ingin sampaikan kepada :

1. Dra. Rum Hastuti, M.Si, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro Semarang.
2. Dr. Edy Sugiyanto SpPD selaku dokter yang membantu dalam penulisan tugas akhir ini.
3. Dr. Widowati, M.Si, selaku Ketua Jurusan Matematika F.MIPA Universitas Diponegoro Semarang.
4. Bambang Irawanto, S.Si, M.Si, selaku Ketua Program Studi Matematika F.MIPA Universitas Diponegoro Semarang.
5. Priyo Sidik Sasongko, S.Si, M.Kom selaku Dosen pembimbing I yang telah meluangkan waktu memberikan bimbingannya hingga selesainya Tugas Akhir.
6. Helmie Arif W, M.Cs, selaku Dosen Pembimbing II yang telah meluangkan waktu memberikan bimbingannya hingga selesainya Tugas Akhir.
7. Semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa dalam pembuatan Tugas Akhir ini masih banyak kekurangannya. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan Tugas Akhir ini. Semoga Tugas Akhir ini bisa membawa manfaat bagi penulis sendiri khususnya dan bagi para pembaca pada umumnya.

Semarang, Desember 2010

Penulis

ABSTRAK

Penderita penyakit jantung di Indonesia semakin meningkat setiap tahunnya. Tenaga medis yang profesional sangat terbatas sehingga masyarakat awam enggan memeriksakan kesehatan jantungnya. Dengan perkembangan teknologi komputer yang modern, masalah ini dapat dipecahkan dengan pembuatan sebuah aplikasi sistem. Oleh karena itu dibangun suatu aplikasi yang disebut aplikasi sistem untuk mendiagnosa penyakit jantung pada manusia. Aplikasi ini dibangun dengan menggunakan mesin inferensi *forward chaining* dan *backward chaining*, dan dikembangkan menggunakan bahasa pemrograman *Delphi 07* serta menggunakan basisdata *Microsoft Access 2003*. Sehingga dengan adanya aplikasi sistem ini masyarakat awam dapat mendiagnosa penyakit jantung yang diderita dan mendapatkan solusinya.

Kata Kunci : jenis penyakit jantung. Forward chaining, backward chaining,

ABSTRACT

Heart disease patients in Indonesia has increased every year. While medical professionals are very limited so with that limitations, ordinary people are reluctant to check the health of the heart. With the development of modern computer technology, problem can be solved by making an application system. Therefore it is necessary to build a system called application system diagnose heart disease in humans. This system applications built using forward chaining inference engine and backward chaining, and developed using Delphi 07 and uses Microsoft Access 2003 database. So by system application to diagnose the common people suffered heart disease and get the solution.

Key words : types of heart disease, forward chaining, backward chaining.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN I	ii
HALAMAN PENGESAHAN II	iii
KATA PENGANTAR	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Pembatasan Masalah	3
1.4 Tujuan dan Manfaat Penulisan.....	3
1.5 Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Kecerdasan Buatan	5
2.2 Sistem Pakar	5
2.2.1 Ciri – ciri Sistem Pakar	6
2.2.2 Keuntungan Sistem Pakar	6

2.2.3 Kelemahan Sistem Pakar	7
2.2.4 Alasan Pengembangan Sistem Pakar	7
2.2.5 Arsitektur Sistem Pakar	8
2.2.6 Representasi Pengetahuan	11
2.2.7 Mesin Inferensi	15
2.3 Specification Requirement System (SRS)	18
2.4 Entity Relationship Diagram (ERD)	19
2.5 Data Flow Diagram (DFD)	22
2.6 Jantung	23
2.7 Pengertian Delphi 07	25
BAB III ANALISA DAN PERANCANGAN SISTEM.....	29
3.1 Analisa Sistem	29
3.1.1 Akuisisi pengetahuan	29
3.1.2 Specification Requirement System (SRS)	33
3.1.3 Entity Relationship Diagram (ERD)	34
3.1.4 Data Flow Diagram (DFD)	36
3.1.4.1 Data Context Diagram	36
3.1.4.2 Data Flow Diagram Level 1	36
3.1.4.3 Data Flow Diagram Level 2 Proses 2	37
3.1.4.4 Data Flow Diagram Level 2 Proses 3	38
3.2. Perancangan Proses	39
3.2.1 Representasi Pengetahuan.....	39

3.2.1.1	Tabel Keputusan	40
3.2.1.2	Pohon Keputusan	42
3.2.1.3	Kaidah produksi.....	45
3.2.1.4	Mesin Inferensi	47
3.3	Perancangan Basis Data	49
3.3.1	Perancangan Antar Muka	51
3.3.1.1	Rancangan Menu Utama	51
3.3.1.2	Rancangan Konsultasi	52
3.3.1.3	Rancangan Menu <i>Login</i> Admin	54
3.3.1.4	Rancangan Menu Administrator	55
3.3.1.5	Rancangan Data Gejala	55
3.3.1.6	Rancangan Data Penyakit	56
3.3.1.7	Rancangan Data Aturan	57
3.3.1.8	Rancangan Data Dokter	57
3.3.1.9	Rancangan Hasil Diagnosa	58
3.3.1.9.1	Rancangan Hasil Diagnosa Forward Chaining	58
3.3.1.9.2	Rancangan Hasil Diagnosa Backward Chaining.....	59
BAB IV	IMPLEMENTASI DAN PENGUJIAN SISTEM	60
4.1	Implementasi Rancangan Fungsi	60
4.1.1	Implementasi Fungsi Forward Chaining.....	60
4.1.2	Implementasi Fungsi Backward Chaining	63
4.2	Implementasi Antar Muka	66
4.2.1	Implementasi Antar Muka Utama.....	66

4.2.2 Implementasi Antar Muka Konsultasi	67
4.2.2.1 Implementasi Menu Konsultasi Berdasarkan Gejala	67
4.2.2.2 Implementasi Menu Konsultasi Berdasarkan Penyakit	68
4.2.3 Implementasi Antar Muka Admin	69
4.2.3.1 Implementasi Data Gejala	70
4.2.3.2 Implementasi Data Penyakit	70
4.2.3.3 Implementasi Data Aturan	71
4.2.3.4 Implementasi Data Dokter	72
4.3 Pengujian Forward Chaining	72
4.4 Pengujian Backward Chaining	79
4.5 Analisis Hasil	84
4.5.1 Analisis Hasil Dari Forward Chaining	84
4.5.2 Analisis Hasil Dari Backward Chaining.....	84
4.6 Hasil Pengujian Sistem	85
BAB V PENUTUP.....	86
5.1 Kesimpulan	86
5.2 Saran	86
DAFTAR PUSTAKA	87

DAFTAR GAMBAR

Gambar 2.1	Arsitektur Sistem Pakar.....	8
Gambar 2.2	Jaringan Simantik Untuk Personal Komputer	12
Gambar 2.3	Jaringan Simantik Yang Diperluas.....	12
Gambar 2.4	Binary Tree.....	15
Gambar 2.5	Hubungan Sekering Dan Peralatannya.....	16
Gambar 2.6	Speed Bar, Main Menu Dan Parletter Bar.....	26
Gambar 2.7	Object Inspector	26
Gambar 2.8	Object Tree View	27
Gambar 2.9	Form	27
Gambar 2.10	Explorer Box Delphi	28
Gambar 3.1	Entity Relationship Diagram (ERD)	35
Gambar 3.2	Data Contex Diagram.....	36
Gambar 3.3	Data Flow Diagram Level 1	37
Gambar 3.4	Diagram Arus Data Level 2 Proses 2.....	38
Gambar 3.5	Diagram Arus Data Level 2 Proses 3	39
Gambar 3.6	Pohon Keputusan	44
Gambar 3.7	Flowchat Forward Chaining	48
Gambar 3.8	Flowchat Backward Chaining	49
Gambar 3.9	Rancangan Menu Utama	52
Gambar 3.10	Rancangan Konsultasi	53
Gambar 3.11	Rancangan Konsultasi Berdasarkan Gejala	53

Gambar 3.12 Rancangan Konsultasi Berdasarkan Penyakit	54
Gambar 3.13 Rancangan Login Admin	54
Gambar 3.14 Rancangan Administrator.....	55
Gambar 3.15 Rancangan Data Gejala	56
Gambar 3.16 Rancangan Data Penyakit	56
Gambar 3.17 Rancangan Data Aturan	57
Gambar 3.18 Rancangan Data Dokter	58
Gambar 3.19 Rancangan Hasil Diagnosa Forward Chaining	58
Gambar 3.20 Rancangan Hasil Diagnosa Backward Chaining.....	59
Gambar 4.1 Tampilan Menu Utama	67
Gambar 4.2 Tampilan Antar Muka Konsultasi.....	67
Gambar 4.3 Tampilan Menu Konsultasi Berdasarkan Gejala	68
Gambar 4.4 Tampilan Menu Konsultasi Berdasarkan Penyakit	68
Gambar 4.5 Tampilan Menu Admin	69
Gambar 4.6 Tampilan Menu Administrator	69
Gambar 4.7 Tampilan Data Gejala.....	70
Gambar 4.8 Tampilan Data Penyakit.....	71
Gambar 4.9 Tampilan Data Aturan	71
Gambar 4.16 Tampilan Data Dokter	72
Gambar 4.17 Tampilan Pengujian Forward Chaining (Gambar 4.11 - 4.26).....	73
Gambar 4.33 Tampilan Pengujian Backward Chaining (Gambar 4.27 – 4.36)	80

DAFTAR TABEL

Tabel 2.1 Simbol Entity Relationship Diagram	20
Tabel 2.2 Data Flow Diagram (DFD)	23
Tabel 3.1 Jantung Hipertensi	30
Tabel 3.2 Jantung Kongenital/bawaan	30
Tabel 3.3 Jantung Koroner.....	31
Tabel 3.4 Jantung Teroid	31
Tabel 3.5 Jantung Paru	31
Tabel 3.6 Jantung Kardiomiopati	32
Tabel 3.7 Jantung Kronis	32
Tabel 3.8 Jantung Akut	33
Tabel 3.9 Jantung Katup	33
Tabel 3.10 Jantung Perikarditis	33
Tabel 3.11 Tabel Keputusan	40
Tabel 3.12 Keterangan Gejala Penyakit Jantung	41
Tabel 3.13 Keterangan Penyakit Jantung	42
Tabel 3.14 Tabel Penyakit	50
Tabel 3.15 Tabel Gejala	50
Tabel 3.16 Tabel Aturan	50
Tabel 3.17 Tabel Admin	51
Tabel 3.18 Tabel Gejala Sementara	51
Tabel 4.1 Hasil Pengujian	85

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jantung merupakan organ yang sangat penting bagi manusia. Jantung adalah pusat kehidupan bagi manusia. Faktor kesehatan jantung juga dipengaruhi oleh pola makanan dan pola fikir manusia tersebut. Tanda – tanda penyakit jantung pada manusia antara lain sebagai berikut : nafas berat, rasa sakit pada rahang, rasa sakit pada punggung, berkeringat dingin, pingsan, gemeteran, rasa panas pada dada. Salah satu jenis penyakit jantung pada manusia adalah penyakit jantung Kongenital / bawaan. Penyakit ini mempunyai ciri – ciri gejala sesak nafas, sering mengalami infeksi paru, dan berat badan akan sedikit berkurang. Penyakit Jantung Kongenital / bawaan kebanyakan diderita oleh kaum wanita.

Kebanyakan masyarakat awam sangat kurang memperhatikan kesehatan, terutama kesehatan jantung. Penyakit jantung dapat menyerang siapa saja dan ada juga penyakit jantung bawaan dari lahir. Mereka enggan memeriksakan kesehatan jantungnya karena biaya yang harus dikeluarkan sangatlah mahal dan tenaga spesialis juga masih jarang dijumpai terutama di daerah pedesaan. Pada umumnya tenaga medis kebanyakan dijumpai di daerah perkotaan. Sedangkan masyarakat yang berada di daerah terpencil sangat sulit untuk memeriksakan ataupun berkonsultasi dengan tenaga spesialis sangatlah tidak terjangkau. Jadi selain biaya yang dikeluarkan sangat mahal, ketersediaan tenaga medis spesialis juga sangat terbatas sehingga

masyarakat yang ada di daerah terpencil sangatlah kesulitan untuk mengetahui penyakit jantung yang sedang dideritanya.

Dengan perkembangan teknologi yang sangat modern, maka dikembangkan pula suatu teknologi yang mampu mengadopsi cara berfikir manusia yaitu teknologi *Artificial Intelligence* atau kecerdasan buatan. Sistem pakar adalah salah satu bagian dari kecerdasan buatan yang menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang memerlukan keahlian manusia. Tujuan dari sistem pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mempresentasikan pengetahuan manusia dalam bentuk sistem, sehingga dapat digunakan oleh orang banyak. Sistem pakar akan memberikan solusi yang memuaskan layaknya seorang pakar. Sistem pakar dibuat pada wilayah pengetahuan tertentu untuk suatu kepakaran tertentu yang mendekati kemampuan manusia pada salah satu bidang yang spesifik.

Berdasarkan hal di atas, tentu dengan mengandalkan kemajuan di bidang teknologi dan informasi, kiranya pengembangan sebuah "Program aplikasi untuk mendignosa penyakit jantung" menjadi sangat penting guna memberikan sosialisasi kepada masyarakat menyangkut dunia kesehatan, memberikan bekal pengetahuan dan pembelajaran, serta memberikan pengetahuan akan pentingnya kesehatan bagi masyarakat awam. Pemilihan masalah menyangkut jenis penyakit – penyakit jantung sebagai sampel penelitian pada program aplikasi bidang medis ini adalah kenyataan bahwa penyakit – penyakit jantung merupakan jenis penyakit yang cukup berbahaya.

1.2. Perumusan Masalah

Permasalahan yang dapat dirumuskan adalah bagaimana cara membuat aplikasi yang dapat membantu dalam menentukan diagnosa terhadap penyakit jantung pada manusia.

1.3. Batasan Masalah

Agar pembahasan tidak meluas maka perlu pembatas masalah – masalah sebagai berikut :

1. Aplikasi yang dibangun untuk mengetahui jenis penyakit jantung berdasarkan gejala kemudian merekomendasikan saran dan pencegahannya.
2. Tidak akan dibahas mengenai jantung secara mendalam hanya jenis penyakit jantung yang dikenal oleh masyarakat awam.
3. Bahasa pemrograman yang digunakan menggunakan Delphi.

1.4. Tujuan dan Manfaat

1.4.1 Tujuan

Untuk membuat aplikasi sebuah sistem pakar dalam mendiagnosa penyakit jantung pada manusia.

1.4.2. Manfaat

Memberikan pengetahuan tentang penyakit jantung dan gejala – gejalanya disertai tindakan yang harus diambil untuk pencegahannya sebagai langkah awal dalam mengantisipasi penyakit jantung.

1.5 Sistematika Penulisan

Untuk memberikan gambaran pokok permasalahan sebagai berikut:

BAB I PENDAHULUAN

Dalam bab ini dijelaskan tentang latar belakang, perumusan masalah, pembatasan masalah, tujuan dan manfaat, sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan tinjauan pustaka yang berisi tentang teori – teori kecerdasan buatan, sistem pakar, *entity relationship diagram*, *specification requirement system*, *data flow diagram*, jenis penyakit jantung, Delphi 7.0.

BAB III ANALISA DAN PERANCANGAN SISTEM

Pada bab ini membahas tentang analisis dan perancangan sistem yang berisi tentang analisis sistem, komponen sistem pakar, rancangan basis data, *entity relationship diagram*, *specification requirement system*, *data flow diagram* dan perancangan antarmuka.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas tentang implementasi antar muka dan pengujian .

BAB V PENUTUP

Bab ini penulisan membuat kesimpulan dan saran yang berisi kesimpulan dan saran mengenai aplikasi sistem pendignosa penyakit jantung pada manusia.

BAB II

LANDASAN TEORI

2.1 Kecerdasan Buatan

Kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia. Aktifitas manusia yang ditirukan seperti penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya. Sesuai dengan definisi tersebut, maka teknologi kecerdasan buatan dipelajari dalam bidang – bidang seperti : Robotika (*Robotics*), Penglihatan komputer (*Computer Vision*), Pengolahan bahasa alami (*Natural Language Processing*), Pengenalan pola (*Pattern Recognition*), Sistem Syaraf Buatan (*Artificial Neural System*), Pengenalan Suara (*Speech Recognition*), dan Sistem Pakar (*Expert System*). Kecerdasan Buatan menyelesaikan permasalahan dengan mendayagunakan komputer untuk memecahkan masalah yang kompleks dengan cara mengikuti proses penalaran manusia. Salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia adalah Sistem pakar. (Hartati , 2008)

2.2. Sistem Pakar

Sistem pakar (*Expert System*) merupakan salah satu cabang dari kecerdasan buatan. Definisi dari sistem pakar yaitu sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti biasa dilakukan oleh para ahli. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli. Bagi para

ahli pun sistem pakar ini juga akan membantu aktivitasnya sebagai asisten yang berpengalaman. (Kusumadewi, 2003)

2.2.1 Ciri – Ciri Sistem Pakar

Adapun ciri – ciri sistem pakar (Kusrini, 2006):

1. Terbatas pada bidang keahlian yang spesifik.
2. Dapat memberikan penalaran untuk data – data yang tidak lengkap / tidak pasti.
3. Dapat mengemukakan rangkaian alasan – alasan yang diberikan dengan cara yang dapat dipahami.
4. Berdasarkan rule atau kaidah tertentu.
5. Dirancang untuk dikembangkan secara bertahap.
6. Keluaran bersifat anjuran atau nasehat.
7. Keluaran tergantung dari dialog dengan user.

2.2.2 Keuntungan Sistem Pakar

Secara garis besar banyak manfaat yang dapat diambil dengan adanya sistem pakar, antara lain (Kusrini, 2006) :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Dapat melakukan proses berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Mampu mengambil dan melestarikan keahlian para pakar.

2.2.3 Kelemahan Sistem Pakar

Di samping memiliki beberapa keuntungan, sistem pakar juga memiliki beberapa kelemahan, antara lain :

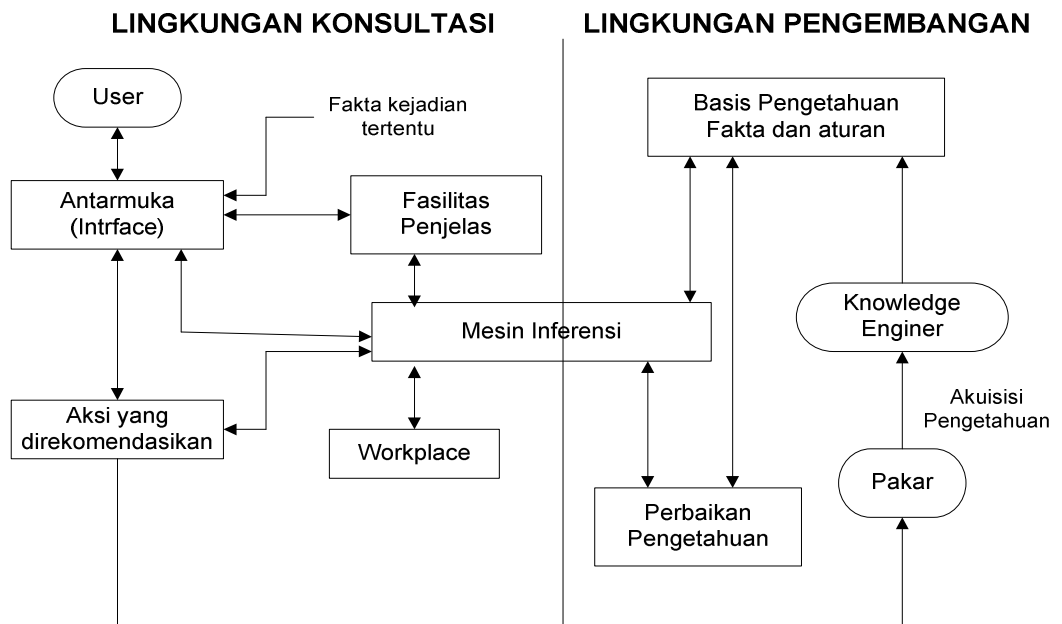
1. Biaya yang sangat mahal membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Hanya dapat menangani pengetahuan yang konsisten.

2.2.4 Alasan Pengembangan Sistem Pakar

Sistem pakar sendiri dikembangkan lebih lanjut dengan alasan (Kusrini, 2006):

1. Dapat menyediakan kepakaran setiap waktu dan di berbagai lokasi.
2. Secara otomatis mengerjakan tugas – tugas rutin yang membutuhkan seorang pakar.
3. Pengetahuan secara pakar mahal nilainya.
4. Kepakaran dibutuhkan juga pada situasi yang kurang mendukung.

2.2.5 Arsitektur Sistem Pakar



Gambar 2.1 Arsitektur Sistem Pakar

Sistem pakar disusun oleh dua bagian utama, yaitu Lingkungan Pengembangan dan Lingkungan Konsultasi. Lingkungan Pengembangan digunakan untuk memasukkan pengembangan pakar ke dalam lingkungan sistem pakar, Lingkungan konsultasi digunakan oleh nonpakar untuk memperoleh pengetahuan dan nasihat pakar. Kebanyakan sistem pakar saat ini tidak berisi komponen perbaikan – pengetahuan. (Turban,2005)

Komponen – komponen yang terdapat dalam sistem pakar adalah :

- 1). Pakar merupakan seseorang yang ahli di bidang tertentu.
- 2). Akuisisi Pengetahuan merupakan penerimaan atau perolehan pengetahuan yang dapat diperoleh dari seorang pakar, buku teks, laporan penelitian dengan dukungan dari seorang *knowledge engineer*.

- 3). *Knowledge Engineer* yaitu seorang spesialis sistem yang menerjemahkan pengetahuan yang dimiliki seorang pakar menjadi pengetahuan yang akan tersimpan dalam basis pengetahuan pada sebuah sistem pakar.
- 4). Basis Pengetahuan, terdiri dari dua jenis, yaitu fakta (situasi dan teori) dan *rule* atau aturan.
- 5). Perbaikan Pengetahuan , yakni mereka dapat menganalisis pengetahuannya sendiri dan kegunaannya, belajar darinya, dan meningkatkannya untuk konsultasi mendatang.
- 6.) Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, dapat dikatakan sebagai mesin pemikir (*thinking machine*). Pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan. Konsep yang biasanya digunakan untuk mesin inferensi adalah runut balik (*backward chaining*) dan menggunakan runut maju (*forward chaining*).
- 7). *Workplace*
Merupakan area dari sekumpulan memori kerja (*working memory*). *Workplace* digunakan untuk merekam hasil – hasil dan kesimpulan yang dicapai. Ada 3 tipe keputusan yang dapat direkam, yaitu :
 - a) Rencana : Bagaimana menghadapi masalah.
 - b) Agenda : Aksi – aksi yang potensial yang sedang menunggu untuk dieksekusi.
 - c) Solusi : Calon aksi yang dibangkitkan.

8). Fasilitas Penjelasan

Proses menentukan keputusan yang dilakukan oleh mesin inferensi selama sesi konsultasi mencerminkan proses penalaran seorang pakar. Karena pemakai kadangkala bukanlah ahli dalam bidang tersebut, maka dibuatlah fasilitas penjelasan. Fasilitas penjelasan inilah yang dapat memberikan informasi kepada pemakai mengenai jalannya penalaran sehingga dihasilkan suatu keputusan. Bentuk penjelasannya dapat berupa keterangan yang diberikan setelah suatu pertanyaan diajukan, yaitu penjelasan atas pertanyaan mengapa, atau penjelasan atas pertanyaan bagaimana sistem mencapai konklusi.

9). Antarmuka (*Interface*)

Sistem pakar menggantikan seorang pakar dalam suatu situasi tertentu, sistem harus menyediakan pendukung yang diperlukan oleh pemakai yang tidak memahami masalah teknis. Sistem pakar juga menyediakan komunikasi antara sistem dan pemakainya, yang disebut sebagai antarmuka. Antarmuka yang efektif dan ramah pengguna (*user-friendly*) penting sekali terutama bagi pemakai yang tidak ahli dalam bidang yang diterapkan pada sistem pakar. (Hartati ,2008)

10). Aksi yang direkomendasi, merupakan saran atau solusi untuk permasalahan yang sedang dihadapi *user*.

11). *User* , yang dimaksud dengan *user* adalah :

a) *Learner* (pelajar) untuk mempelajari bagaimana Sistem pakar menyelesaikan permasalahan.

- b) *Client* (yaitu bukan pakar) yang menginginkan *advice* (nasehat). Bertindak seperti seorang konsultan atau penasihat.
- c) Pakar, di sini sistem pakar bertindak sebagai kologen atau asisten.
- d) Pembangun sistem pakar yang ingin meningkatkan basis pengetahuan.

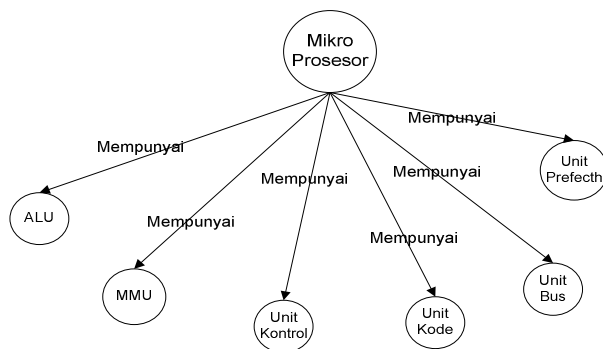
2.2.6 Representasi Pengetahuan

Sistem pakar merupakan sistem yang berbasis pengetahuan, mengerjakan tugas yang biasanya dilakukan oleh seorang pakar. Representasi pengetahuan dimaksudkan untuk mengorganisasikan pengetahuan dalam bentuk dan format tertentu untuk bisa dimengerti oleh komputer. Untuk membuat sistem pakar yang efektif harus dipilih representasi pengetahuan yang tepat. Pemilihan representasi pengetahuan yang tepat akan membuat sistem pakar dapat mengakses basis pengetahuan tersebut untuk keperluan pembuatan keputusan. Beberapa model representasi pengetahuan yang penting yaitu :

1. Jaringan Semantik

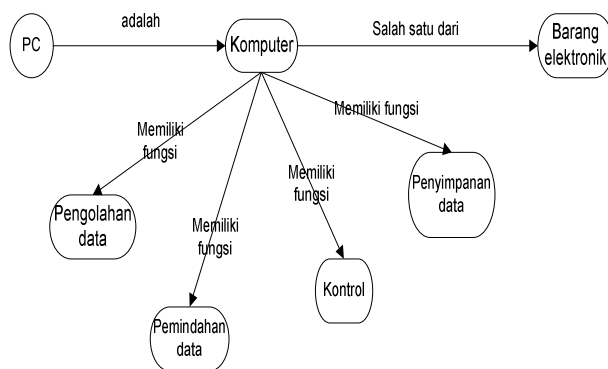
Teknik representasi pengetahuan yang digunakan untuk informasi proposional, sedangkan yang dimaksud dengan informasi proposional adalah pernyataan yang mempunyai nilai benar atau salah. Representasi jaringan semantik merupakan penggambaran grafis dari pengetahuan yang memperlihatkan hubungan hirarkis dari objek – objek. Komponen dasar untuk mempresentasikan pengetahuan dalam bentuk jaringan semantik adalah simpul (*node*) dan penghubung (*link*). Obyek direpresentasikan oleh simpul. Hubungan antara obyek – obyek dinyatakan oleh penghubung yang diberikan

label untuk menyatakan hubungan yang dipresentasikan. Contoh jaringan semantik terlihat pada gambar 2.2 dan gambar 2.3(Hartati, 2008)



Gambar 2.2 Contoh jaringan semantik

Terdapat deskripsi : PC (*Personal Computer*) merupakan salah satu barang elektronik. Semua PC memiliki fungsi pengolahan data, pemindahan data, kontrol, dan penyimpanan data ; tetapi tidak semua barang elektronik memiliki fungsi – fingsi tersebut. Penyajian dalam bentuk jaringan semantiknya seperti terlihat pada gambar 2.3



Gambar 2.3 Contoh jaringan semantik yang diperluas.

2). Bingkai (Frame)

Bingkai berupa kumpulan slot-slot yang berisi atribut untuk mendeskripsikan pengetahuan. Pengetahuan yang termuat dalam slot dapat berupa kejadian, lokasi, situasi ataupun elemen-elemen lainnya. Bingkai digunakan untuk representasi pengetahuan deklaratif.

3). Logika Predikat

Logika predikat berdasarkan pada kebenaran dan kaidah inferensi untuk merepresentasikan simbol – simbol dan hubungannya satu dengan yang lain. Logika predikat selain digunakan untuk menentukan kebenaran (*truthfulness*) atau kesalahan (*falsity*) sebuah pernyataan, juga dapat digunakan untuk merepresentasikan pernyataan tentang obyek tertentu.

4). Kaidah Produksi

Kaidah menyediakan cara formal untuk merepresentasikan rekomendasi, arahan, atau strategi. Kaidah produksi dituliskan dalam bentuk jika – maka (*if – then*). Kaidah *if – then* menghubungkan antesenden (*antecedent*) dengan konsekuensi yang diakibatkannya. Premis mengacu pada fakta yang harus benar sebelum konklusi tertentu dapat diperoleh. Antesenden mengacu situasi yang terjadi sebelum konsekuensi dapat diamati. Data mengacu pada informasi yang harus tersedia sehingga sebuah hasil dapat diperoleh. Berbagai struktur kaidah *if – then* yang menghubungkan obyek atau atribut sebagai berikut :

IF [kondisi] THEN [aksi]

Contoh :

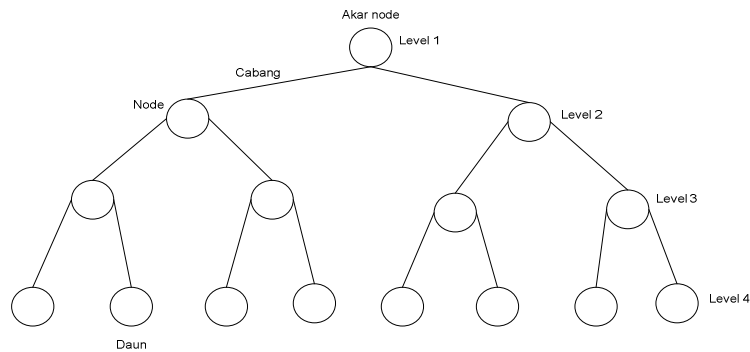
IF [demam AND kelelahan AND sesak nafas]

THEN [ambil aspirin AND istirahat]

5). Pohon keputusan

Tree (pohon keputusan) adalah suatu hirarki struktur yang terdiri dari node (simbol) yang menyimpan informasi atau pengetahuan dan cabang yang menghubungkan node. Cabang disebut juga link atau edge dan node disebut juga vertek. Gambar 2.4 menunjukkan binary tree yang mempunyai 0,1 atau 2 cabang per node. Dengan berorientasi pada tree (pohon), akan node adalah node yang tertinggi dalam hirarki dan daun adalah paling bawah. Tree dapat dianggap sebagai suatu tipe khusus dari jaringan simantik yang setiap nodenya, kecuali akar, pasti mempunyai satu node orang tua dan mempunyai nol atau lebih node anak.

Untuk tipe biasa dari binary tree, maksimum mempunyai dua anak untuk setiap node, dan sisi kanan dan kiri dari node anak dibedakan.



Gambar 2.4 Binary Tree

Jika node mempunyai lebih dari satu orang tua disebut dengan jaringan. Gambar 2.4 menunjukkan hanya ada satu urutan dari edge atau path dari akar untuk tipe node. Oleh karena itu dalam hal ini tidak mungkin untuk memindahkan secara berlawanan dengan arah panah.

2.2.7 Mesin Inferensi

Mesin Inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan memformulasikan kesimpulan.

Mesin inferensi mengarahkan pencarian melalui basis pengetahuan, proses yang dapat melibatkan aplikasi aturan inferensi disebut pencocokan pola. Program kontrol memutuskan aturan mana yang diinvestigasi, alternative mana yang dieliminasi, dan atribut mana yang sesuai. Program kontrol yang paling populer untuk sistem berbasis aturan, yaitu *backward chaining* dan *forward chaining*.

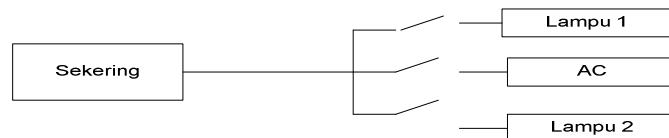
1. Forward Chaining

Forward chaining merupakan peruntutan yang dimulai dengan menampilkan kumpulan data atau fakta yang menyakinkan menuju konklusi akhir. Jadi dimulai dari premis – premis atau informasi masukan (if) dahulu kemudian menuju konklusi atau *derived information* (then) atau dapat dimodelkan sebagai berikut:

IF (informasi masukan)

THEN (konklusi)

Informasi masukan dapat berupa data, bukti, temuan atau pengamatan. Sedangkan konklusi dapat berupa tujuan, hipotesa, penjelasan atau diagnosa. Sehingga jalannya *forward chaining* maju dapat dimulai dari data menuju tujuan dan bukti menuju hipotesa, dari temuan menuju penjelasan, atau dari pengamatan menuju diagnosa.



Gambar 2.5 Hubungan antara sekering dan peralatannya

Dari gambar 2.5 dapat dibuat salah satu kaidah sebagai berikut:

IF Lampu 1 dinyalakan

AND Lampu 1 tidak menyala

AND Lampu 1 dihubungkan dengan sekering

AND Sekering masih utuh

THEN Lampu 1 rusak

Secara sederhana *forward chaining* diterangkan sebagai berikut, untuk kaidah di atas, agar sistem pakar mencapai konklusi, harus display terlebih dahulu fakta lampu 1 mati, lampu 1 dihubungkan dengan sekering, dan sekering masih utuh, baru sistem bisa mengeluarkan konklusi bahwa lampu 1. rusak. (Hartati,2008)

2. *Backward Chaining*

Backward chaining merupakan proses peruntukan yang arahnya kebalikan dari runut maju. Proses penalaran runut balik di mulai dengan tujuan / goal kemudian merunut balik ke jalur yang akan mengarahkan ke goal tersebut, mencari bukti – bukti bahwa itu diaplikasikan ketika tujuan atau hipotesis yang dipilih itu sebagai titik awal penyelesaian masalah. Disebut juga *goal – driven search*. Runut balik dimodelkan sebagai berikut:

Tujuan,

IF (kondisi),

Contoh runut maju di atas dinyatakan dalam runut balik sebagai berikut :

Lampu 1 rusak

IF Lampu 1 dinyalakan

AND Lampu 1 tidak menyala

AND Lampu 1 dihubungkan dengan sekering

AND Sekering masih utuh

Secara sederhana runut balik diterangkan sebagai berikut:

Untuk kaidah di atas sistem menduga terlebih dahulu bahwa lampu 1 rusak. Kebenaran praduga ini dibuktikan dengan menanyakan apakah ada fakta lampu 1 mati, apakah ada fakta lampu 1 dihubungkan dengan sekering dan sekering masih utuh, bila keduanya dipenuhi maka praduga sistem benar, dan sistem mengeluarkan kesimpulan bahwa lampu 1 rusak.

Dari sisi pemakai (user), mekanisme inferensi baik dengan metode runut balik maupun runut maju tidak kelihatan bedanya, hanya bagi pengembang sistem pakar, kedua metode runut balik, proses internal selalu mengecek konklusi terlebih dahulu, sebagai praduga awal, baru kemudian mengecek gejala – gejala dipenuhi user atau tidak, bila keseluruhan gejala terpenuhi, maka praduga sistem benar dan dikeluarkan sebagai output, bila ada gejala yang tidak terpenuhi berarti praduga sistem salah, selanjutnya sistem akan mengecek konklusi berikutnya.

Sebaliknya pada metode runut maju, sistem tidak melakukan praduga apapun, namun sistem akan menerima semua gejala yang diberikan user, kemudian sistem akan mengecek gejala – gejala tersebut memenuhi konklusi yang mana. Jadi lampu 1 rusak (goal) akan tercapai jika kondisi – kondisi yaitu : lampu 1 dinyalakan, lampu 1 tidak nyala, lampu 1 telah dihubungkan dengan sekering, dan sekering masih utuh, semuanya terbukti terpenuhi (Hartati, 2008)

2.3 Specification Requirement System (SRS)

Specification requirement system terdiri dari beberapa macam yaitu *user requirement* (persyaratan user), *system requirement* (persyaratan sistem), dan *software design specification* (spesifikasi perancangan sistem). Persyaratan user



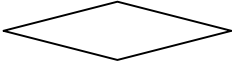

merupakan pernyataan, dalam bahasa natural ditambah diagram, mengenai apa yang kita harapkan disediakan oleh sistem dan batasan operasinya. Persyaratan sistem menentukan layanan dan batasan sistem secara rinci. Sedangkan spesifikasi perancangan sistem merupakan gambaran abstrak dari perancangan perangkat lunak yang menjadi dasar bagi perancangan dan implementasi yang lebih rinci.

Persyaratan perangkat lunak sering diklasifikasikan sebagai persyaratan fungsional dan non fungsional. Namun di sini hanya menggunakan persyaratan fungsional. *Functional requirement* berisi tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem bereaksi terhadap input tertentu, dan bagaimana sistem harus berlaku pada situasi-situasi tertentu.

2.4 Entity Relationship Diagram (ERD)

Model data *Entity-Relationship* (ER) terdiri dari sekumpulan objek-objek yang disebut dengan entitas dan hubungan yang terjadi diantara objek-objek tersebut. Entitas merupakan suatu objek dasar atau individu yang mewakili sesuatu yang nyata eksistensinya dan dapat dibedakan dari objek-objek yang lain. Suatu entitas mempunyai sekumpulan sifat dan nilai dari beberapa sifat tersebut adalah unik yang dapat mengidentifikasi entitas tersebut. Sekumpulan entitas yang mempunyai tipe yang sama (sejenis) dan berada dalam lingkup yang sama membentuk suatu himpunan entitas. Suatu entitas memiliki atribut. Atribut merupakan sifat-sifat atau properti yang dimiliki oleh entitas. Atribut inilah yang membedakan antara entitas satu dengan entitas yang lain. Sedangkan relasi menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari sejumlah himpunan entitas yang berbeda (Widodo, P.A, dkk, 2004).

Tabel 2.1 Simbol – simbol ERD

Simbol	Keterangan
	Himpunan Entitas , digunakan untuk menggambarkan objek
	Atribut , digunakan untuk menjelaskan karakter dari entity.
	Relasi , menggambarkan himpunan relationship.
	Garis , digunakan untuk menghubungkan entitas dengan entitas, entitas dengan relasi maupun entitas dengan atribut.

ERD terdiri atas elemen – elemen yaitu Entitas, Atribut, Relasi dan Kunci

(*Key*). Berikut ini merupakan penjelasan dari elemen – elemen tersebut :

a). Entitas

Entitas adalah suatu objek dasar atau individu yang mewakili sesuatu yang nyata dan dapat dibedakan dengan yang lainnya. Himpunan entitas adalah sekumpulan entitas dengan berbagai atribut yang sama.

b). Atribut

Atribut merupakan sifat – sifat atau karakteristik yang melekat dalam sebuah entitas.

c). Relasi

Relasi adalah hubungan di antara sejumlah entitas yang berasal dari sejumlah himpunan entitas yang berbeda. Relation atau hubungan dapat memiliki atribut. Sebuah kelas hubungan dapat melibatkan banyak kelas entitas.

d). Kunci (Key)

Kunci merupakan suatu atribut yang unik yang dapat digunakan untuk membedakan suatu entitas dengan lain dalam suatu himpunan entitas. Terdapat 3 macam kunci, yaitu :

- i). Superkey, adalah himpunan yang terdiri dari satu atau lebih yang dapat membedakan setiap baris data dengan unik pada sebuah table.
- ii). Candidate key, adalah himpunan atribut minimal yang dapat membedakan setiap baris data dengan unik dalam sebuah table.
- iii). Primary key, merupakan kunci yang paling unik, lebih ringkas, dan digunakan sebagai acuan.

e). Kardinalitas Relasi

Kardinalitas Relasi adalah jumlah maksimum entitas yang mana entitas tersebut dapat berelasi dengan entitas pada himpunan entitas yang lain.

i). Satu ke satu (1-1)

Suatu entitas di dalam himpunan A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B dan entitas di dalam himpunan entitas B dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas A.

ii). Satu ke banyak (1-N)

Suatu entitas di dalam himpunan entitas A dihubungkan dengan lebih dari satu entitas di dalam himpunan entitas B, dan entitas di dalam

himpunan entitas B hanya dapat dihubungkan dengan paling banyak satu entitas dalam himpunan entitas A.

iii). Banyak ke satu (N-1)

Suatu entitas di dalam himpunan entitas A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B dan entitas di dalam himpunan entitas B dapat dihubungkan dengan lebih dari satu entitas dalam himpunan entitas A.

iv). Banyak ke banyak (N-N)

suatu entitas di dalam himpunan entitas A dapat dihubungkan dengan lebih dari satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B dapat dihubungkan dengan lebih dari satu entitas dalam himpunan entitas A.

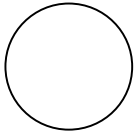
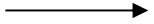
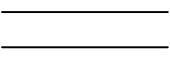

2.5 Data Flow Diagram (DFD)

DFD adalah suatu model logika atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, yang mana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut.

DFD menggambarkan proses penyimpanan data dan proses yang mentransformasikan data. DFD menunjukkan hubungan antara data pada sistem dan proses pada sistem (Kristanto, 2008).

Tabel 2.2 Data Flow Diagram (DFD)

Simbol	Keterangan
--------	------------

	<p>Proces Proses sering juga disebut dengan <i>buble</i>. Berfungsi menggambarkan transformasi aliran data masuk menjadi aliran data keluar.</p>
	<p>Data Flow Berfungsi menggambarkan aliran data atau paket informasi dari satu bagian sistem ke bagian lain. Arah panah menggambarkan aliran data.</p>
	<p>Data Store Berfungsi menggambarkan model dari kumpulan paket data yang tersimpan.</p>
	<p>Eksternal Entity Sering juga disebut terminator, berfungsi menggambarkan kesatuan luar yang berhubungan dengan sistem.</p>

2.6 Jantung

Jantung adalah organ berupa otot, berbentuk kerucut, berongga dan dengan basisnya di atas dan puncaknya di bawah. *Apex – nya* (puncak) miring ke sebelah kiri. Berat jantung kira – kira 300 gram. Jantung berada di dalam torak, antara kedua paru – paru di belakang sternum, dan lebih menghadap ke kiri dari pada ke kanan. Ukuran jantung kira – kira sebesar kepalan tangan. Jantung terbagi oleh sebuah septum, (sekat) menjadi dua belahan, yaitu kiri dan kanan. Sesudah lahir tidak ada hubungan satu dengan yang lain antara kedua belahan ini. Setiap belahan kemudian dibagi lagi dalam dua ruang, yang atas disebut atrium dan yang bawah disebut ventrikel. Maka di kiri terdapat 1 atrium dan 1 ventrikel, dan di kanan juga ada 1 atrium 1 ventrikel.

Kejadian – kejadian yang terjadi dalam jantung selama peredaran darah disebut *siklus jantung*. Gerakan jantung berasal dari nodus sinus – atrial, kemudian kedua atrium berkontraksi. Debaran jantung atau lebih tepat debaran *apex*, adalah pukulan

ventrikel kiri kepada dinding anterior yang terjadi selama kontraksi ventrikel.
(Evelyn , 2002)

Penyakit jantung terbagi menjadi 10 bagian, diantaranya adalah sebagai berikut:

1. Gagal Jantung Kronik

Kondisi patofisiologi (kelainan fungsi jantung), dimana terdapat kegagalan jantung memompa darah yang sesuai dengan kebutuhan jaringan.

2. Gagal Jantung Akut

Serangan cepat (*rapid onset*) dari gejala – gejala atau tanda – tanda (*symptom and signs*) akibat fungsi jantung yang abnormal.

3. Jantung Katup

Penyakit jantung yang disebabkan karena kelainan katup jantung

4. Jantung Perikarditis

Inflamansi pericardius, kantung membrane yang membungkus jantung / peradangan perikard (selaput jantung).

5. Jantung Koroner

Penyakit jantung akibat gangguan / kelainan pada pembuluh darah koroner

6. Jantung Hipertensi

Penyakit jantung yang disebabkan karena hipertensi.

7. Jantung Kardiomiopati

Suatu kelompok penyakit yang langsung mengenai otot jantung atau miokard itu sendiri.

8. Penyakit Jantung Kongenital/ bawaan

Merupakan kelainan struktur atau fungsi dari sistem kardiovaskuler yang ditemukan pada saat lahir walaupun dapat ditemukan di kemudian hari.

9. Penyakit Jantung Paru

Pembesaran jantung kanan yang disebabkan oleh penyakit paru kronis dan tidak berhubungan dengan kelainan jantung kiri.

10. Penyakit Jantung Teroid

Penyakit Jantung teroid disebabkan oleh kelebihan atau kekurangan hormon teroid.

2.7. PENGERTIAN DELPHI

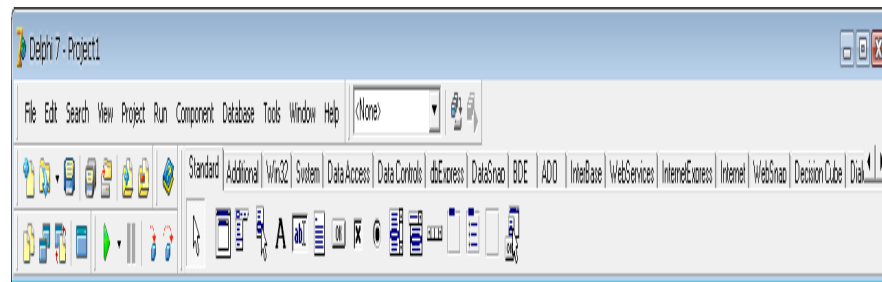
Delphi adalah salah satu dari pemrograman secara visual, bahasa yang digunakan lebih mengarah ke bahasa pascal. Banyak orang mengatakan bahwa pascal adalah nenek moyangnya Delphi, ini karena Delphi adalah pengembangan dari turbo pascal yang populer. Seperti bahasa lainnya, Delphi mengalami pengembangan yang sangat pesat, Delphi memiliki support yang tinggi terhadap database – database yang sudah terkenal, dan dilengkapi dengan obyek – obyek yang memudahkan dalam pembuatan program, baik program database maupun program lain (Andi, 2003).

Tampilan dan menu pada Delphi :

1. Main Menu, Speed Bar, dan Palette Bar

Di dalam main menu, terdapat fasilitas – fasilitas yang disediakan oleh Delphi untuk mengontrol serta menangani masalah yang berhubungan dengan pekerjaan seperti menu untuk menyimpan, menu untuk meng- compile, dan masih banyak lagi.

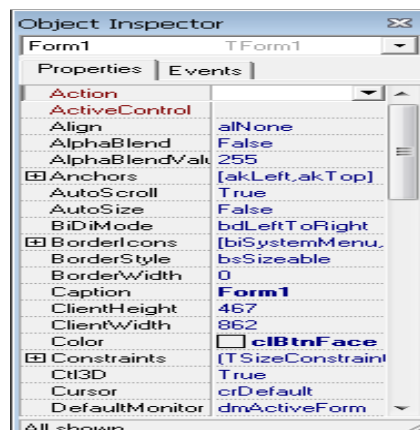
Kegunaan speed bar seperti main menu, tapi pada speed bar berupa icon sehingga cukup mengeklik saja untuk menjalankan fasilitas yang dikehendaki. Palette bar adalah tempat obyek – obyek yang digunakan untuk menyusun sebuah program.



Gambar 2.6. Speed Bar, Main Menu, dan Palette Bar

2. Objek Inspector

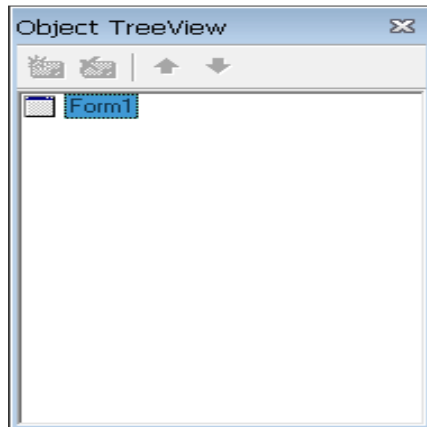
Objek Inspector terdapat dua tab, yaitu: properties berfungsi mengubah sifat – sifat yang dibawa oleh objek tersebut. Events digunakan sebagai daftar procedure yang dilakukan oleh objek tersebut ketika suatu peristiwa terjadi.



Gambar 2.7. Object Inspector

3. Object Tree View

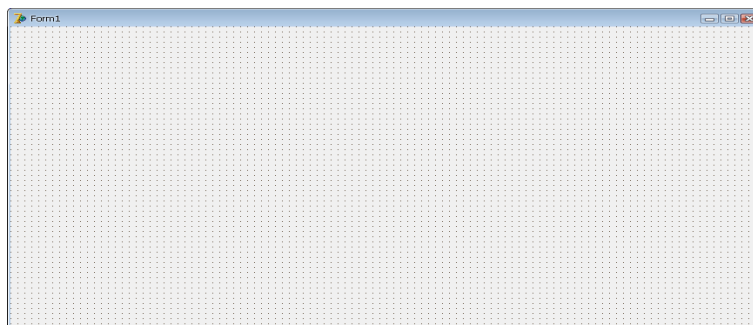
Kotak ini sebagai gambar, object apa saja yang telah terpasang pada form serta dapat menunjukan object yang sulit ditunjuk.



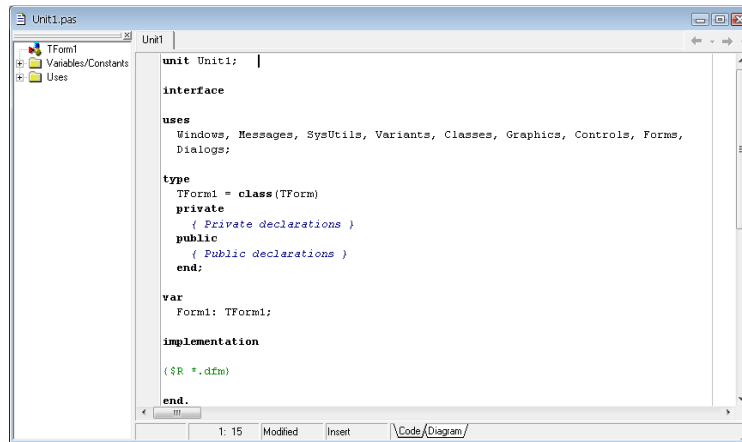
Gambar 2.8. Object Tree View

4. Form, Code Editor, Explorer Box

Form adalah tempat untuk memasang object – object. Code editor adalah tempat untuk menuliskan source code atau bahasa program. Explorer box yaitu kotak yang menggambarkan macam – macam class yang dimuat dalam unit, procedure atau function yang telah dibuat, mana variabel yang ada di unit, serta unit – unit yang digunakan oleh unit itu sendiri.



Gambar 2.9. Form



Gambar 2.10.Explorer Box