

**SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN  
MOBIL DIESEL**

**SKRIPSI**



**Disusun Oleh**

**IRJA ATJO SUTRISNO**

**05. 12. 637**

**PROGRAM STUDI TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2012**

WORLD MANUFACTURING OUTPUT REPORT  
ECONOMIC REPORT

1970

WORLD MANUFACTURING OUTPUT REPORT  
CONTINUED FROM PAGE  
100 (1st 1/2)

THE WORLD MANUFACTURING OUTPUT REPORT  
CONTAINS THE FOLLOWING INFORMATION:  
WORLD MANUFACTURING OUTPUT REPORT  
WORLD MANUFACTURING OUTPUT REPORT  
1970

**LEMBAR PERSETUJUAN**

**SISTEM PAKAR UNTUK ANALISA KERUSAKAN  
MESIN MOBIL DIESEL**

**SKRIPSI**

*Disusun Dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana  
Teknik Komputer Dan Informatika Strata Satu (S-1)*

*Disusun oleh :*  
**IRJA ATJO SUTRISNO**  
**NIM : 05.12.637**

**Mengetahui,**  
**Ketua Program Studi Teknik Elektro S-1**



**Ir. Yusuf Ismail Nakhoda, MT**  
**NIP.Y. 1018800189**

**Diperiksa dan Disetujui,**



**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Dr. Eng. Arynanto S, ST, MT**  
**NIP. Y. 1030800417**

**Ir. Taufik Hidayat, MT**  
**NIP.Y. 1018700151**

**PROGRAM STUDI TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2012**

## SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan dibawah ini:

Nama : Irja Atjo Sutrisno

Nim : 05.12.637

Program Studi : TEKNIK ELEKTRO S-1

Konsentrasi : TEKNIK KOMPUTER & INFORMATIKA

Dengan ini saya menyatakan bahwa skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam skripsi ini tidak memuat karya orang lain kecuali ditentukan sumbernya sesuai dengan ketentuan yang berlaku. Demikian surat pernyataan ini saya buat, dan apabila dikemudian hari ada pelanggaran atas surat pernyataan ini, Saya bersedia menerima sanksinya.

Malang 8 Agustus 2012

Yang Membuat Pernyataan



IRJA ATJO SUTRISNO

NIM : 05.12.637

## ABSTRAK

### SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL

Irja Atjo Sutrisno NIM 05.12.637

Email [irja\\_acho@yahoo.co.id](mailto:irja_acho@yahoo.co.id)

Dosen Pembimbing I, Dr. Eng Aryuanto S, ST, MT Dosen Pembimbing II, Ir  
Taufik Hidayat, MT

*Kebijakan pemerintah untuk mengatur dan menaikan harga bahan bakar minyak jenis premium, dengan menyamakan harga bensin dan solar tentunya mempengaruhi minat masyarakat akan mobil diesel. Keuntungan mobil bermesin diesel adalah penggunaannya irit bahan bakar dan mesinnya yang tangguh. Diesel adalah salah satu jenis bahan bakar minyak. Di Indonesia, diesel lebih dikenal dengan nama solar.*

*Mesin mobil diesel dianggap susah untuk diperbaiki atau diutak-atik, sehingga harus dilakukan oleh teknisinya. Kerusakan pada mesin mobil diesel sering terjadi karena kurangnya perawatan pemilik mobil terhadap kendaraannya sendiri.*

*Dari hasil analisa penyebab kerusakan Mesin Mobil Diesel, didalam aplikasi ini adalah (kerusakan mesin) bukanlah kerusakan yang bisa didiagnosa secara pasti, mengingat berbagai pertimbangan yang dipakai untuk melakukan proses diagnose*

*Kata kunci : Analisa kerusakan Mobil Diesel, sistem pakar.*

## **KATA PENGANTAR**

Puji syukur kehadirat TUHAN YANG MAHA ESA, yang telah memberikan Rahmatnya, sehingga penulis dapat menyelesaikan laporan skripsi ini dengan baik dan lancar. Laporan skripsi ini merupakan salah satu persyaratan akademik dalam menyelesaikan program Strata 1 Jurusan Teknik Elektro, Konsentrasi Komputer & Informatika, Institut Teknologi Nasional Malang. Adapun judul laporan Skripsi ini adalah:

### **SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL**

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak-pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Soeparno Djiwo,MT selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir.Sidik Noetjahjono,MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir.Yusuf Ismail Nahkoda,MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
4. Bapak Dr.Eng.Aryuanto Soetedjo,ST,MT selaku Sekretaris Jurusan Teknik Elektro S-1 ITN Malang.
5. Bapak Dr.Eng.Aryuanto Soetedjo,ST,MT selaku dosen pembimbing I.
6. Bapak Ir. Taufik Hidayat, MT selaku dosen pembimbing II.
7. Ayahanda Daeng Haruna Arif & ibunda tercinta Hj.Sutarmi, kaka perempuan Wati adik perempuan Riska & adik laki-laki Ardian & Ragil, dan semua keluarga besar di Merauke Papua, Malang, Makasar, yang telah memberikan dukungan baik materi, moril maupun spiritual dan untuk selalu berdoa dan berusaha beserta nasehat yang telah diberikan sampai saat ini.
8. Bapak Ibrahim Ashari,ST.MT dan seluruh dosen, pegawai ITN Kampus 2 Malang.
9. Teman-teman angkatan 2005 T.INFOKOM ITN Malang.

10. Teman-teman Kos Merah Jln.Perusahaan Gang III, teman-teman Ikatan Pelajar Mahasiswa Merauke di malang (IPMAMER) yang selalu menemani dan memberi semangat dalam menyelesaikan studi di ITN Malang.

Penulis berharap agar buku laporan Skripsi ini dapat memberikan banyak manfaat bagi semua pihak yang membutuhkan, khususnya bagi rekan-rekan mahasiswa. Penulis menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan, oleh karena itu mohon maaf apabila dalam buku ini terdapat hal-hal yang kurang berkenan dihati para pembaca.

Penulis juga mengharap koreksi, kritik serta saran-saran yang bermanfaat demi kesempurnaan buku Laporan Skripsi ini.

Malang, Juli 2012

Penulis

# DAFTAR ISI

	Halaman
<b>LEMBAR PERSETUJUAN</b> .....	i
<b>SURAT ORISINALITAS</b> .....	ii
<b>ABSTRAKSI</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian .....	2
1.4. Manfaat Penelitian .....	3
1.5. Batasan Masalah .....	3
1.6. Sistematika Penelitian.....	4
<b>BAB II TEORI DASAR</b> .....	6
2.1. Sistem Pakar.....	6
2.2. Struktur Sistem Pakar .....	6
2.2.1. Subsistem Akuisisi Pengetahuan .....	7
2.2.2. Basis Pengetahuan (Knowledge Base) .....	8
2.2.3. Mesin Inferensi .....	9
2.2.4. Antarmuka Pengguna (User Interface) .....	12
2.2.5. Tempat Kerja (Blackboard) .....	12
2.2.6. Subsistem Penjelasan (Justifier) .....	13

2.2.7. Sistem Perbaikan Pengetahuan (Knowlegde-base-editor)...	13
2.2.8. Fakta Tentang Kejadian Tertentu .....	14
2.3 Fungsi Rekursif.....	14
2.4 Lazarus .....	15
2.5 MySQL .....	18
2.6 Perangkat Pemodelan Sistem dalam Pembuatan Suatu Program Aplikasi	
.....	20
2.6.1 Tujuan Pemanfaatan <i>Database</i> .....	20
2.6.1. DFD (Data Flow Diagram).....	20
2.6.2. Entity Relationship .....	21
2.6.2.1. Kardinalitas atau Derajat Relasi .....	22
2.6.3. Pengertian Data Base .....	24
2.6.4. Bagan Alir (Flowchart).....	25
2.6.4.1. Simbol-simbol dalam Flowchart .....	24
2.6.5. Teknik Normalisasi.....	28
2.6.5.1. Atribut Kunci.....	28
2.7 Mesin-mesin Mobil Diesel.....	29
2.7.1. Prinsip Kerja Mesin Diesel .....	30
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>32</b>
3.1 Alat dan Bahan Penelitian.....	32
3.1.1. Alat Penelitian.....	32
3.1.2. Bahan Penelitian.....	32
3.2 Rancangan Penelitian.....	33
3.3 Metode Penelitian .....	33
3.3.1 Metode Pengumpulan Data .....	33

3.3.2 Metode Pengembangan Perangkat Lunak.....	34
3.4 Analisis Sistem.....	43
3.4.1. Perancangan Basis Data .....	43
3.4.2. Normalisasi .....	45
3.4.2.1. Teknik Normalisasi.....	45
3.4.2.2. Entity Relation Diagram (ERD) .....	46
3.4.3. Struktur Tabel.....	47
3.4.4. DFD (Data Flow Diagram) .....	48
3.4.4.1. Context Diagram.....	48
3.4.4.2. DFD Level 0 .....	49
3.4.4.3. DFD Level 1 Proses Inputan Data Kerusakan.....	50
3.4.4.4. DFD Level 1 Proses Analisa Sebab Kerusakan.....	51
3.5. Flowchart Sistem .....	52
3.6. Perancangan Antar Muka.....	52
3.6.1. Rancangan Menu Utama .....	53
3.6.2. Rancangan Menu Admin.....	53
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>55</b>
4.1. Implementasi.....	55
4.2. Penjelasan Program.....	55
4.2.1 Menu Halaman Depan .....	55
4.2.2 Halaman Menu Login Admin .....	56
4.2.3 Menu Halaman Data Admin .....	56
4.2.4 Menu Halaman Data Kerusakan .....	57
4.2.5 Menu Halaman Data Penyebab Kerusakan .....	57
4.2.6 Menu Halaman Proses Analisa .....	58

4.2.7 Menu Halaman Proses Pilihan Penyebab Kerusakan .....	58
4.2.8 Menu Halaman Hasil Deteksi Kerusakan.....	59
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>60</b>
5.1 Kesimpulan .....	60
5.2 Saran .....	61
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2.1 Struktur Sistem Pakar.....	7
Gambar 2.2 Proses Forward Chaining.....	10
Gambar 2.3 Proses Backward Chaining.....	10
Gambar 2.4 Diagram Alir Teknik Penelusuran Depth First Search.....	11
Gambar 2.5 Diagram Alir Teknik Penelusuran Breadth First Search.....	11
Gambar 2.6 Best-First Search.....	12
Gambar 2.7 Tampilan Awal dari IDE Lazarus.....	16
Gambar 2.8 Menu Utama dan Daftar Komponen Lazarus.....	16
Gambar 2.9 Form Utama.....	17
Gambar 2.10 Object Inspector.....	17
Gambar 2.11 Source Editor.....	18
Gambar 2.12 Simbol Entitas.....	22
Gambar 2.13 Simbol Tabel.....	22
Gambar 2.14 Simbol Penghubung.....	22
Gambar 2.15 Relasi Satu Ke Satu.....	23
Gambar 2.16 Relasi Satu Ke Satu.....	23
Gambar 2.17 Relasi Banyak Ke Satu.....	24
Gambar 2.18 Relasi Banyak Ke Banyak.....	24
Gambar 2.19 Simbol-simbol Flowchart.....	27
Gambar 3.1 Rancangan Penelitian.....	33
Gambar 3.2 Relasi Antar Tabel.....	47
Gambar 3.3 Context Diagram.....	48
Gambar 3.4 Data Folw Diagram.....	49
Gambar 3.5 DFD Level 1 Proses 2.....	50
Gamabr 3.6 DFD Level 1 Proses Analisa Kerusakan.....	51
Gambar 3.7 <i>Flowchart</i> Sistem.....	52
Gambar 3.8 Rancangan Menu Utama.....	53
Gambar 3.9 Rancangan Menu Admin.....	53
Gambar 4.1 Menu Halaman Depan.....	55
Gambar 4.2 Menu Login Admin.....	56
Gambar 4.3 Menu Halaman Data Admin.....	56
Gambar 4.4 Menu Halaman Data Kerusakan.....	57

<b>Gambar 4.5 Menu Halaman Data Penyebab Kerusakan.....</b>	<b>57</b>
<b>Gambar 4.6 Menu Halaman Proses Analisa Kerusakan .....</b>	<b>58</b>
<b>Gambar 4.7 Menu Halaman Proses Analisa Pilihan Penyebab Kerusakan .</b>	<b>58</b>
<b>Gambar 4.8 Menu Halaman Hasil Deteksi Kerusakan dan Solusi Perbaikan.</b>	<b>59</b>

## **DAFTAR TABEL**

<b>Tabel 2.1 Perbandingan Rekursif Dengan Interatif .....</b>	<b>14</b>
<b>Tabel 3.1 Redudansi Data .....</b>	<b>44</b>
<b>Tabel 3.2 Tabel Bentuk Tak Normal .....</b>	<b>45</b>
<b>Tabel 3.3 Tabel Bentuk Normal Pertama.....</b>	<b>45</b>
<b>Tabel 3.4 Tabel Bentuk Normal Kedua .....</b>	<b>46</b>
<b>Tabel 3.5 Tabel Kerusakan .....</b>	<b>47</b>
<b>Tabel 3.6 Sebab.....</b>	<b>48</b>
<b>Tabel 3.9 Tabel Sebab Kerusakan.....</b>	<b>48</b>



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Kebijakan pemerintah untuk mengatur dan menaikkan harga bahan bakar minyak jenis premium, dengan menyamakan harga bensin dan solar tentunya mempengaruhi minat masyarakat akan mobil diesel. Keuntungan mobil bermesin diesel adalah penggunaannya irit bahan bakar dan mesinnya yang tangguh. Diesel adalah salah satu jenis bahan bakar minyak. Di Indonesia, diesel lebih dikenal dengan nama solar.

Mesin mobil diesel dianggap susah untuk diperbaiki atau diutak-atik, sehingga harus dilakukan oleh teknisinya. Kerusakan pada mesin mobil diesel sering terjadi karena kurangnya perawatan pemilik mobil terhadap kendaraannya sendiri.

Perkembangan kemajuan teknologi yang sangat pesat, berpengaruh pula pada perkembangan perangkat Lunak saat ini, sehingga perangkat Lunak semakin memasyarakat. Perkembangan ini sangatlah membantu dalam menyajikan informasi yang cepat dan efisien. Seiring perkembangan teknologi, dikembangkan pula teknologi yang mampu mengadopsi proses dan cara berpikir manusia yaitu teknologi *Artificial Intelligence* atau Kecerdasan Buatan. Sistem Pakar adalah salah satu bagian dari Kecerdasan Buatan yang mengandung pengetahuan dan pengalaman yang dimasukkan oleh satu atau banyak pakar ke dalam suatu area pengetahuan tertentu, sehingga setiap orang dapat menggunakannya untuk memecahkan berbagai masalah yang bersifat spesifik.

Sistem pakar pada dasarnya dibuat dan digunakan untuk membantu memecahkan suatu masalah, tanpa tergantung dari seorang pakar. Kebutuhan akan keahlian manusia tentunya sangat variatif sesuai dengan bentuk masalah, misalnya pada usaha jasa bengkel yang menangani kerusakan mesin mobil diesel. Dalam menangani kerusakan mesin mobil diesel sangat bergantung pada keahlian teknisinya, namun keterbatasan

teknisi akan menghambat penanganan kerusakan. Untuk mengantisipasi hal ini maka perlu adanya pembuatan sistem pakar untuk membantu menganalisa sebab kerusakan mesin mobil diesel. Sistem ini dapat mempermudah teknisi dan pemilik mobil dalam menangani kerusakan mesin mobil diesel secara cepat. Berdasarkan hal tersebut di atas maka dibuatlah “SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL” sebagai upaya untuk mengembangkan cara menemukan kerusakan mesin Mobil Diesel melalui sistem komputer.

## **1.2 Rumusan Masalah**

1. Bagaimana sistem pakar yang akan dibuat dapat membantu menyelesaikan kerusakan pada Mobil Diesel.
2. Bagaimana seorang mekanik dengan menggunakan sistem pakar yang akan dibuat bisa menyelesaikan pekerjaannya dalam waktu yang cukup efisien dan efektif.

## **1.3 Tujuan Penelitian**

Adapun yang menjadi tujuan penulisan dalam penyusunan Skripsi adalah sebagai berikut : Sistem Pakar dibangun untuk menemukan penyebab kerusakan mesin mobil diesel agar pengguna dapat mengetahui penyebab kerusakan mesin mobil dengan memanfaatkan Aplikasi Lazarus pada *personal computer*.

## **1.4 Manfaat Penelitian**

Sedangkan manfaat penyusunan Skripsi ini adalah sebagai berikut :

### **1. Bagi Penulis**

Menambah motivasi dan inspirasi untuk mengembangkan software-software yang lebih baik, efektif dan efisien.

2. Untuk memberikan kemudahan bagi orang awam maupun pakar sehingga mendapatkan penanganan lebih cepat pada kerusakan Mobil Diesel.
3. Menjadikan pengetahuan yang mudah didapatkan untuk mencari solusi kerusakan pada mobil.
4. Penghematan waktu dalam menyelesaikan kerusakan-kerusakan pada Mobil Diesel
5. Sebagai pedoman bagi orang-orang yang membutuhkan dalam memperbaiki kerusakan pada Mobil Diesel
6. Bagi Akademis  
Sebagai masukan positif dalam proses belajar mengajar dan menunjang peningkatan pengetahuan mahasiswa angkatan selanjutnya dengan tujuan memantau perkembangan mutu akademik, serta menambah perbendaharaan literatur bagi perpustakaan ITN Malang.

## **1.5 Batasan Masalah**

Dalam penyusunan Skripsi agar menjadi sistematis dan mudah dimengerti, maka akan diterapkan beberapa batasan masalah. Selain itu maksud dari pembatasan masalah adalah karena keterbatasan waktu dalam melakukan penelitian dan pengumpulan data secara terperinci. Batasan masalah juga akan memudahkan penyusunan laporan yang sistematis agar mudah dipahami oleh pembaca.

Batasan-batasan masalah antara lain :

1. Spesifikasi dan data utama penunjang untuk mendiagnosis kerusakan menggunakan model mesin 4JA1.
2. Pembuatan aplikasi sistem pakar ini menggunakan metode forward chaining dari data – data penunjang.
3. Sistem pakar untuk analisa kerusakan mesin Mobil Diesel tidak menangani langsung masalah yang dihadapi, tetapi hanya

membantu memberitahukan penyebab kerusakan kepada pengguna dan teknisi.

4. Hanya membahas penyebab kerusakan Mobil diesel
5. Penyimpanan data-data menggunakan database MySQL
6. Bahasa pemrograman menggunakan LAZARUS Free Pascal, aplikasi ditampilkan dalam bentuk Personal komputer

## **1.6 Sistematika Penulisan**

Agar lebih jelas dan mudah untuk dimengerti, maka penulis akan memberikan rangkuman dari pembuatan tugas akhir ini untuk memberikan gambaran-gambaran pokok sebagai berikut :

### **BAB I. PENDAHULUAN**

Bab ini menjelaskan secara umum mengenai latar belakang masalah, permasalahan dengan batasan-batasan masalah yang digunakan, tujuan, metode serta pembahasan tugas akhir ini.

### **BAB II. DASAR TEORI**

Bab ini berisi dasar-dasar teoritis kecerdasan buatan dan sistem pakar untuk melandasi pemecahan masalah serta teori-teori yang digunakan dalam pembuatan tugas akhir ini.

### **BAB III. METODOLOGI PENELITIAN: SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL.**

Bab ini membahas tentang perancangan umum maupun uraian lebih lanjut mengenai perancangan sistem dalam pembuatan perangkat lunak. Uraian perancangan sistem ini meliputi perancangan data mengenai data input dan output sistem, perancangan proses mengenai bagaimana sistem akan bekerja dengan proses-proses tertentu, maupun

perancangan antar muka dalam desain dan implementasi yang akan digunakan dalam pembuatan tugas akhir ini.

#### **BAB IV. HASIL ANALISA SISTEM PAKAR**

Bab ini menjelaskan tentang pengujian sistem secara umum maupun terperinci. Pengujian sistem secara umum akan membahas mengenai lingkungan uji coba untuk menggunakan sistem ini. Selanjutnya secara lebih terperinci dijelaskan dalam pengujian baik user umum maupun admin, beserta langkah-langkah dalam uji coba tersebut, kemudian dianalisa kembali apakah telah sesuai dengan tujuan pembuatan pada bab I.

#### **BAB V. KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran.



## **BAB II**

### **DASAR TEORI**

#### **2.1 Sistem Pakar**

Sistem pakar adalah sebuah perangkat lunak komputer yang memiliki basis pengetahuan untuk domain tertentu dan menggunakan penalaran inferensi menyerupai seorang pakar dalam memecahkan masalah. Sedangkan definisi lain dari sistem pakar adalah sistem berbasis komputer yang menggunakan pengetahuan, fakta, dan teknik penalaran dalam memecahkan masalah yang biasanya hanya dapat dipecahkan oleh seorang pakar dalam suatu bidang tertentu.

Sistem pakar menggabungkan pengetahuan dan penelusuran data untuk memecahkan masalah yang secara normal memerlukan keahlian manusia. Tujuan dari Sistem Pakar sebenarnya bukan untuk menggantikan peran manusia, tetapi untuk mensubtitusikan pengetahuan manusia kedalam bentuk sistem, sehingga dapat digunakan oleh orang banyak (Nugroz, 2007).

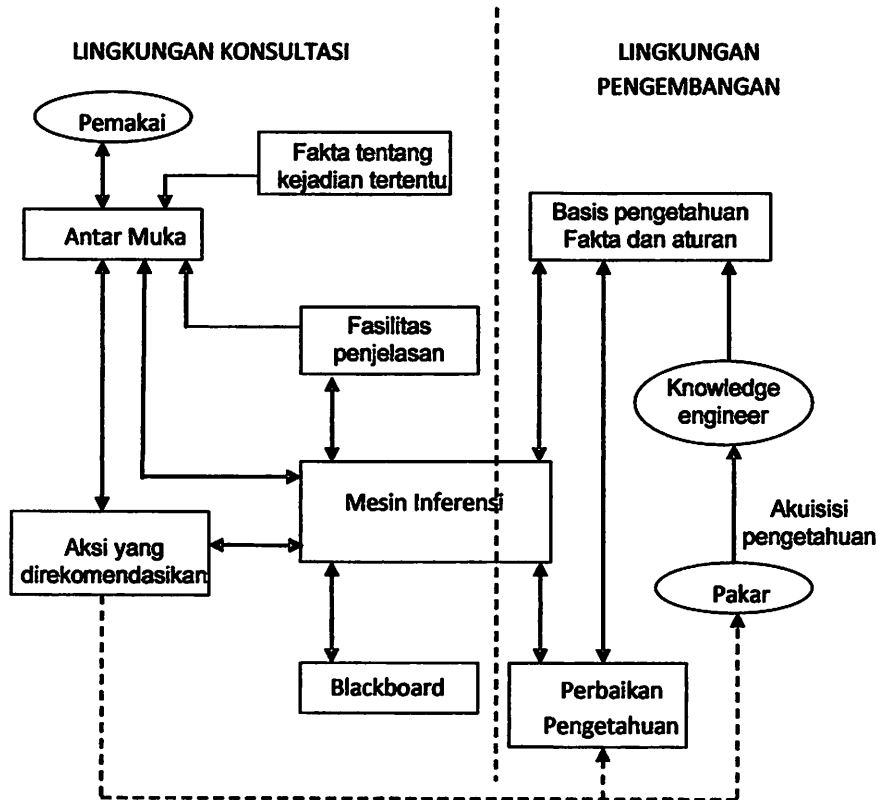
#### **2.2 Struktur Sistem Pakar**

Menurut Turban (2005), sistem pakar dapat ditampilkan dengan dua lingkungan, yaitu : lingkungan pengembangan dan lingkungan konsultasi (runtime). Lingkungan pengembangan digunakan oleh ES builder untuk membangun komponen dan memasukkan pengetahuan ke dalam basis pengetahuan. Lingkungan konsultasi digunakan oleh user nonpakar untuk memperoleh pengetahuan dan nasihat pakar. Lingkungan ini dapat dipisahkan setelah sistem lengkap.

Tiga komponen utama yang tampak secara virtual di setiap sistem pakar adalah basis pengetahuan, mesin inferensi, dan antarmuka pengguna. Sistem pakar yang berinteraksi dengan pengguna dapat pula berisi komponen tambahan sebagai berikut :

- Subsistem akuisisi pengetahuan.
- *Blackboard* (tempat kerja).

- Subsistem penjelasan (*justifier*).
- Sistem perbaikan pengetahuan.



Gambar 2.1 Struktur Sistem Pakar (sumber: Turban (2005))

Kebanyakan sistem pakar saat ini tidak berisi komponen perbaikan pengetahuan.

Deskripsi singkat tiap komponen dapat dilihat pada bagian berikut :

### 2.2.1 Subsistem Akuisisi Pengetahuan

Akuisisi pengetahuan adalah akumulasi, transfer, dan transformasi keahlian pemecahan masalah para pakar atau sumber pengetahuan terdokumentasi ke program komputer, untuk membangun atau memperluas basis pengetahuan. Sumber pengetahuan potensial antara lain pakar manusia, buku teks, dokumen multimedia, database, laporan riset khusus, dan informasi yang terdapat dalam web.

Mendapatkan pengetahuan dari pakar adalah tugas kompleks yang sering menimbulkan kemacetan dalam konstruksi sistem pakar. Dalam membangun sistem yang besar, seseorang memerlukan knowledge

*engineer* atau *pakar elisitasi pengetahuan* untuk berinteraksi dengan satu atau lebih pakar manusia. Biasanya *knowledge engineer* membantu pakar menyusun area persoalan dengan menginterpretasikan dan mengintegrasikan jawaban manusia, menyusun analogi, mengajukan contoh perbandingan dan menjelaskan kesulitan konseptual.

### **2.2.2 Basis Pengetahuan (Knowledge Base)**

Basis pengetahuan merupakan representasi pengetahuan dari seorang pakar yang diperlukan untuk memahami, memformulasikan dan memecahkan masalah. Terdiri dari dua elemen dasar, yaitu :

- 1) Fakta yang berupa informasi tentang situasi permasalahan, teori dari area permasalahan atau informasi tentang objek.
- 2) Spesial heuristik yang merupakan informasi tentang cara bagaimana membangkitkan fakta baru dari fakta yang sudah diketahui. Dalam sistem pakar berbasis rule, bagian ini berupa rules.

Knowledge base adalah jantung sebuah sistem pakar. Bagian ini adalah totalitas keahlian pakar yang telah disarikan dan diformat ke dalam external memory komputer. Sampai saat ini terdapat berbagai cara representasi pengetahuan yang telah dikenal, misalnya :

#### **a. Rule-Based Knowledge**

Pengetahuan direpresentasikan dalam suatu bentuk fakta (facts) dan aturan (rules). Bentuk representasi ini terdiri atas premise dan kesimpulan. Pada penalaran berbasis aturan, pengetahuan dipersentasikan dengan menggunakan aturan berbentuk : IF-THEN. Bentuk ini digunakan apabila kita memiliki sejumlah pengetahuan pakar pada suatu permasalahan tertentu, dan pakar dapat menyelesaikan masalah tersebut secara berurutan. Disamping itu, bentuk ini juga digunakan apabila dibutuhkan penjelasan tentang langkah-langkah pencapaian solusi.

#### **b. Case-Base Reasoning**

Pada penalaran berbasis kasus (cases), basis pengetahuan berisi solusi-solusi yang telah dicapai sebelumnya, kemudian diturunkan

suatu solusi untuk keadaan sekarang (fakta yang ada). Bentuk ini digunakan apabila user menginginkan untuk mengetahui lebih banyak lagi pada kasus-kasus yang hampir sama (mirip). Selain itu, bentuk ini juga digunakan apabila telah dimiliki sejumlah situasi atau kasus tertentu dalam basis pengetahuan atau dapat diartikan pengetahuan direpresentasikan dalam bentuk kesimpulan kasus.

**c. Frame-Based Knowledge.**

Pengetahuan direpresentasikan dalam suatu bentuk hirarki atau jaringan frame.

**d. Object-Based Knowledge.**

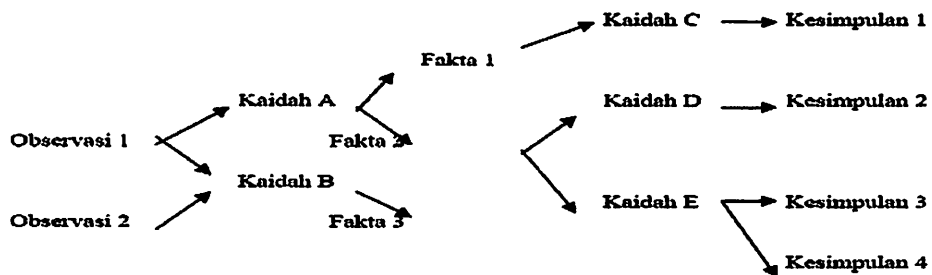
Pengetahuan direpresentasikan sebagai jaringan dari objek-objek. Objek adalah elemen data yang terdiri dari data dan metode (proses).

### **2.2.3 Mesin Inferensi (*Inference Engine*)**

Menurut Turban, Mesin Inferensi adalah program komputer yang memberikan metodologi untuk penalaran tentang informasi yang ada dalam basis pengetahuan dan dalam workplace, dan untuk menformulasikan kesimpulan. (Arhami, 2005) Terdapat dua pendekatan untuk mengontrol inferensi dalam sistem pakar berbasis aturan, yaitu (Arhami, 2005):

**1. Pelacakan ke depan (forward chaining)**

Pelacakan kedepan adalah pendekatan yang dimotori data (data-driven). Dalam pendekatan ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Pelacakan ke depan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN.

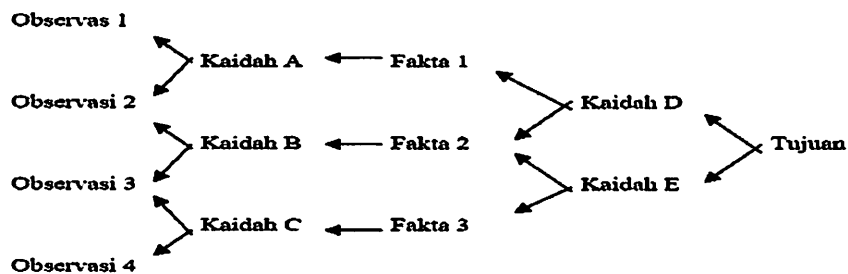


Gambar 2.2. Proses Forward Chaining

(Sumber: Muhammad Arhami, 2005)

## 2. Pelacakan ke belakang (backward chaining)

Pelacakan ke belakang adalah pendekatan yang di motori tujuan terlebih dahulu (goal-driven). Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya.

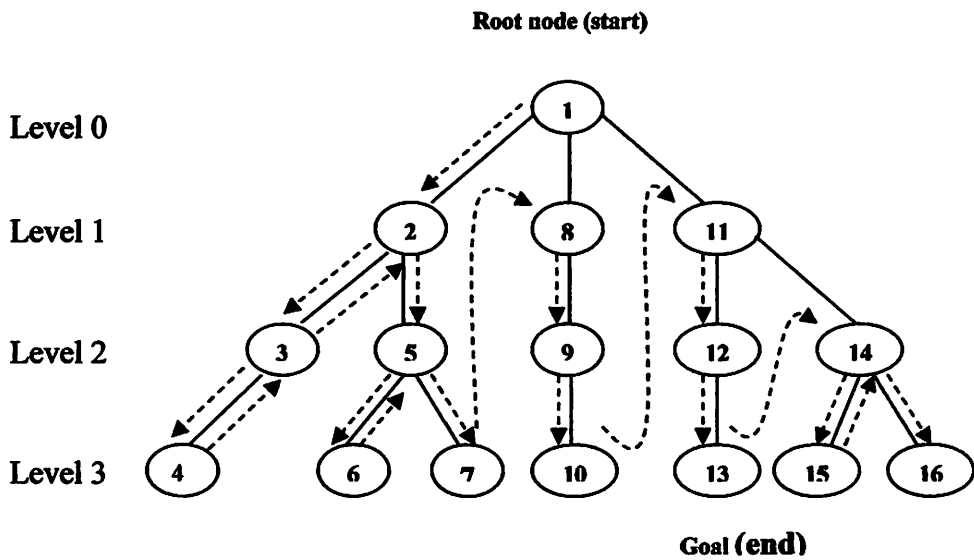


Gambar 2.3 Proses Backward Chaining

(Sumber: Muhammad Arhami, 2005)

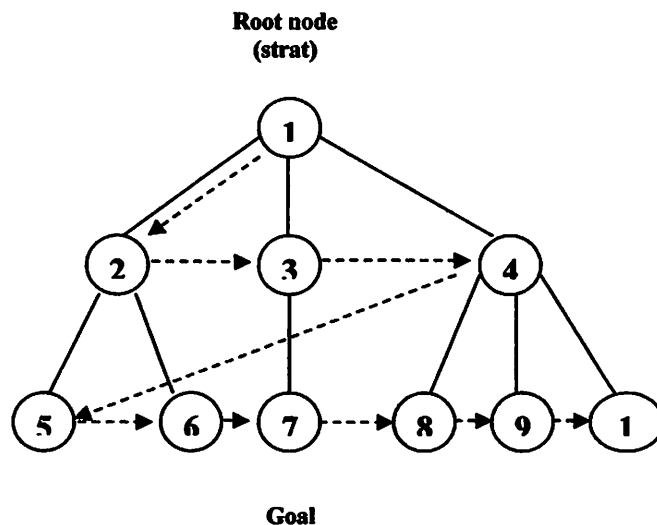
Kedua metode inferensi tersebut dipengaruhi oleh tiga macam penelusuran, yaitu Depth-first search, Breadth-first search dan Best-first search (Arhami, 2005).

1. Depth-first search, melakukan penelusuran kaidah secara mendalam dari simpul akar bergerak menurun ke tingkat dalam yang berurutan.



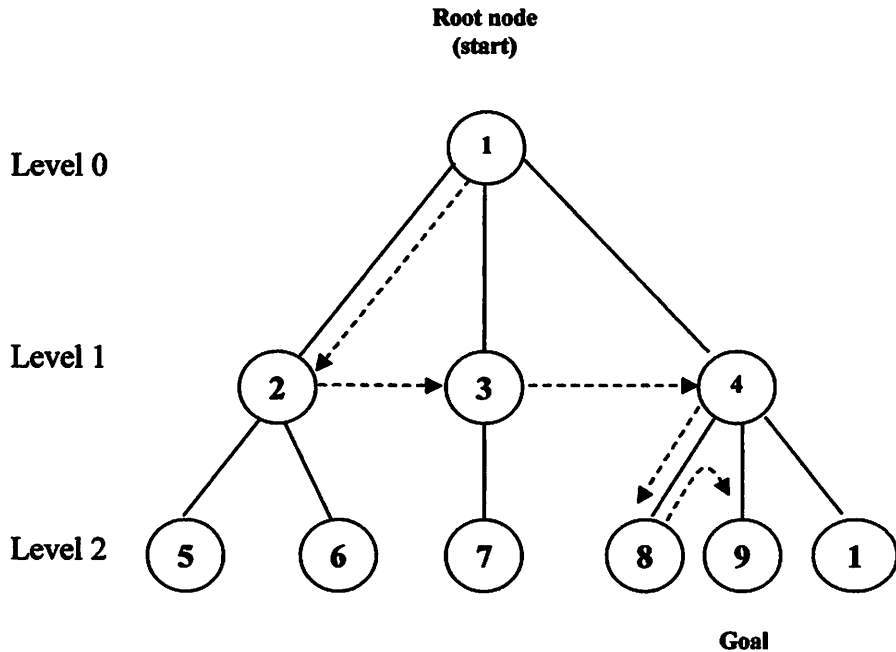
Gambar 2.4. Diagram Alir Teknik Penelusuran Depth First Search  
(Sumber: Muhammad Arhami, 2005)

2. Breadth-first search, bergerak dari simpul akar, simpul yang ada pada setiap tingkat diuji sebelum pindah ke tingkat selanjutnya.



Gambar 2.5 Diagram Alir Teknik Penelusuran Breadth-first search  
(Sumber: Muhammad Arhami, 2005)

3. Best-first search, bekerja berdasarkan kombinasi kedua metode sebelumnya.



Gambar 2.6. Best-First Search  
(Sumber: Muhammad Arhami, 2005)

#### 2.2.4 Antarmuka Pengguna (User Interface).

Merupakan bagian dari sistem pakar yang berfungsi sebagai pengendali input-output. User interface melayani user selama proses konsultasi mulai dari tanya-jawab untuk mendapatkan fakta-fakta yang dibutuhkan oleh inference engine sampai menampilkan output yang merupakan kesimpulan atau rekomendasi yang dihasilkan oleh inference engine.

#### 2.2.5 Tempat Kerja (Blackboard)

Blackboard adalah area kerja memori yang disimpan sebagai database untuk deskripsi persoalan terbaru yang ditetapkan oleh data input, digunakan juga untuk perekaman hipotesis dan keputusan sementara. Tiga

tipe keputusan dapat direkam dalam blackboard, antara lain : rencana (bagaimana mengatasi persoalan), agenda (tindakan potensial sebelum eksekusi), dan solusi (hipotesis kandidat dan arah tindakan alternatif yang telah dihasilkan sistem sampai dengan saat ini).

#### **2.2.6 Subsistem Penjelasan (Justifier)**

Bagian yang harus siap memberikan penjelasan saat user perlu mengetahui apakah alasan diberikannya sebuah solusi. Bagian ini secara konkrit membedakan sebuah sistem pakar dengan sistem aplikasi yang biasa, karena pada pemrograman konvensional tidaklah biasa sebuah sistem menyediakan informasi tambahan mengapa atau dari mana sebuah solusi diperoleh. Bagian ini mempunyai kemampuan untuk menelusuri konklusi dan menerangkan tingkah laku sistem pakar dengan menjawab pertanyaan-pertanyaan seperti :

- Mengapa pertanyaan tersebut diajukan oleh sistem pakar?
- Bagaimana atau darimana konklusi tersebut diperoleh?
- Mengapa alternatif tersebut ditolak?

Pada sistem pakar berbasis aturan, biasanya penjelasan ini dilakukan dengan cara memperlihatkan aturan-aturan yang digunakan. Fasilitas ini penting untuk menambah rasa percaya user pada hasil output program sistem pakar yang digunakannya.

#### **2.2.7 Sistem Perbaikan Pengetahuan (*Knowledge-base editor*)**

Bagian yang digunakan untuk menambah, menghapus atau memperbaiki basis pengetahuan. Bagian ini tidaklah mutlak, karena mayoritas sistem pakar berbasis pengetahuan dalam format text-file, sehingga bagian ini dapat digantikan dengan berbagai word processor yang tersedia. Namun demikian bila sistem pakar dituntut untuk memiliki kemampuan machine learning dari pengalaman konsultasinya, bagian ini menjadi sangat vital.

#### **2.2.8 Fakta Tentang Kejadian Tertentu.**

Bagian ini hanya diperlukan saat data yang telah dimiliki pemakai (file database atau spreadsheet) diperlukan sebagai referensi untuk menarik kesimpulan.

### 2.3 Fungsi Rekursif

Rekursif adalah salah satu metode dalam dunia matematika dimana definisi sebuah fungsi mengandung fungsi itu sendiri. Dalam dunia pemrograman, rekursi diimplementasikan dalam sebuah fungsi yang memanggil dirinya sendiri. Contoh fungsi rekursif misalnya adalah fungsi pangkat, faktorial, dan barisan fibonacci. Ide dasar dalam memecahkan suatu masalah dengan rekursif adalah sebagai berikut :

- Tentukan kasus penyetop atau kasus dasar di mana pemanggilan rekursif tidak lagi diperlukan (karena solusinya sudah diperoleh).
- Terapkan suatu langkah untuk menggiring kasus kompleks ke kasus penyetopnya dengan metode yang mencerminkan fungsinya.

Berikut adalah perbandingan keuntungan dan kelebihan fungsi rekursif dibandingkan iteratif (Widiarta, 2007) :

Tabel 2.1 Perbandingan rekursif dengan iteratif (sumber: Widiarta, 2007)

Rekursif	Iteratif
Kode program biasanya lebih ringkas dan mudah dipahami	Kode program lebih panjang, untuk beberapa kasus solusi iteratif lebih sulit diterapkan
Membutuhkan alokasi memori yang besar	Relatif lebih kecil alokasi memorinya
Tidak cocok ketika kinerja tinggi diperlukan, karena terjadi <i>overhead</i> pemanggilan fungsi dalam jumlah yang relatif besar	Cocok diterapkan ketika kinerja aplikasi harus diterapkan (hanya ada satu kali pemanggilan fungsi)

## 2.4 Lazarus

Lazarus adalah pemrograman yang bersifat gratis (free) dan open source yang berfungsi sebagai alat untuk kompilator Free Pascal. IDE Lazarus (screenshot) adalah lingkungan pemrograman yang kaya fitur untuk membuat aplikasi stand alone grafis dan konsol serta stabil. Lazarus dapat dipakai pada Linux, Mac OS X dan Win32 dan menyediakan sumber editor yang disesuaikan dan bentuk (form) visual diciptakan sesuai dengan paket manajer, debugger dan memiliki integrasi GUI yang lengkap dengan compiler Free Pascal. (Wikipedia. 2012)

Lazarus mulai versi 0.9.26.2 sudah sangat stabil dan bisa dibandingkan dengan Delphi 7. Kelebihan Lazarus jika dibandingkan dengan Delphi adalah sebagai berikut:

1. Open Source dan Free.
2. Multiplatform, mendukung Windows, Linux, Mac OS dan Pocket PC.
3. Bisa menghasilkan code 64-bit.
4. Dikembangkan oleh komunitas open source sehingga berkembang dengan sangat pesat.

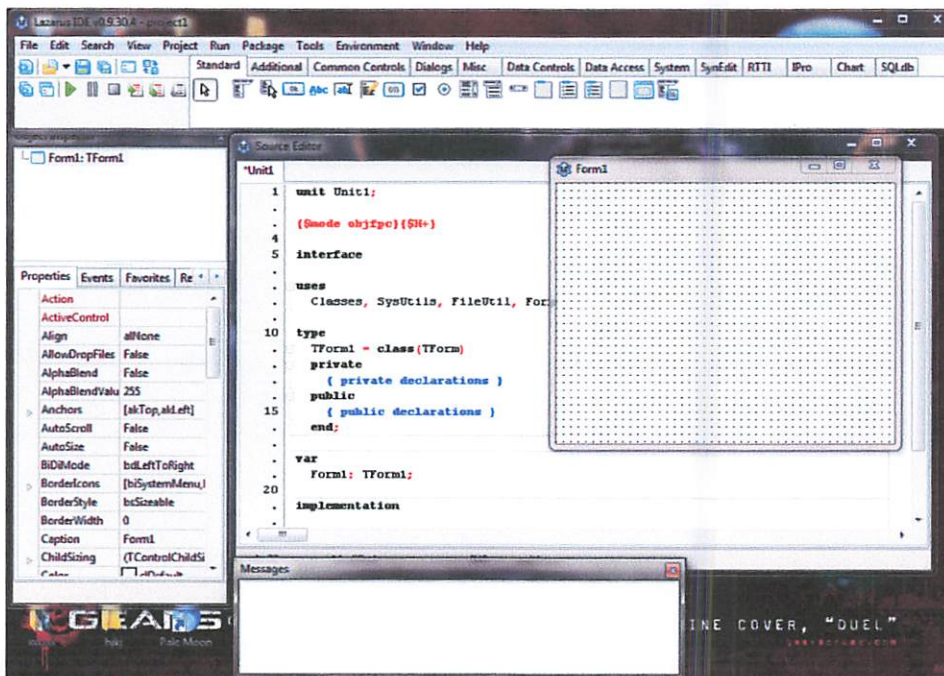
Kekurangan Lazarus sendiri antara lain:

1. Lazarus sampai saat ini masih dalam pengembangan, jadi belum ada versi stabilnya.
2. Dukungan ke database masih tidak selengkap Borland.
3. MDI, untuk linux masih belum mendukung MDI Form
4. Reporting, lazarus belum mempunyai internal database reporting tools (Delphi -> QuickReport.
5. Besarnya ukuran executable, sebelum mendistribusikan file executable kita harus memotong ukuran file dengan program khusus.

## Tutorial Lazarus.

Bahasa pemrograman yang dijadikan landasan dalam Lazarus adalah Pascal. Karena itu saat pengembangan aplikasi, apa yang disediakan oleh Lazarus terasa sama sebagaimana yang terdapat di Delphi, IDE Pascal Visual yang terkenal di lingkungan Windows.

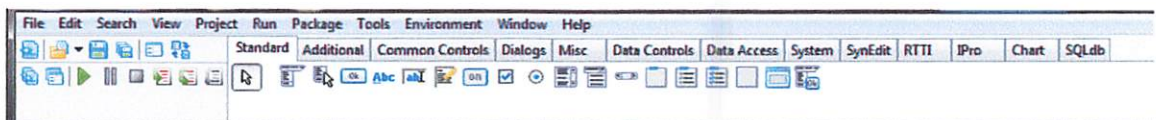
Berikut tampilan awal dari Lazarus.



Gambar 2.7 Tampilan awal dari IDE Lazarus.

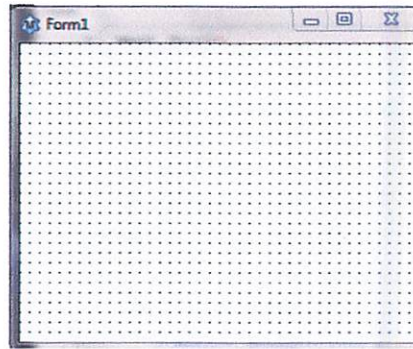
Bagian-bagian yang terdapat dalam Lazarus yang sering digunakan dalam pengembangan perangkat lunak:

### 1) Menu Utama dan daftar komponen.



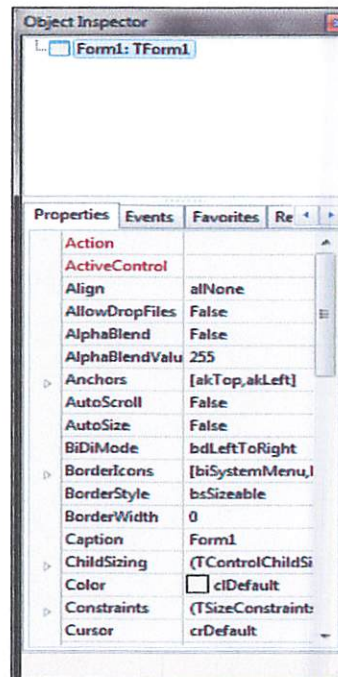
Gambar 2.8 Menu utama dan daftar komponen Lazarus.

- 2) Form Utama. Jendela ini adalah bagian utama dari sebuah aplikasi visual. Form merupakan komponen paling penting, di atas form ini diletakkan komponen-komponen lain.



Gambar 2.9 Form Utama.

- 3) Object Inspector. Jendela ini digunakan untuk mengatur perilaku dari suatu control atau komponen yang diletakan di atas suatu form atau jendela aplikasi yang sedang dikembangkan.



Gambar 2.10 Object Inspector.

- 4) Source Editor. Jendela ini digunakan untuk menuliskan baris-baris kode program untuk mengatur apa yang harus dilakukan oleh aplikasi. Kode program ditulis mengikuti aturan dalam bahasa Pascal.
- Pada bagian paling bawah terdapat jendela Messages. Pada jendela ini akan muncul pesan-pesan tertentu saat eksekusi program terutama jika terjadi kesalahan atau error.



Berikut beberapa tipe data yang dikenal dalam database MySQL:

1) Tipe Data Numerik

Tipe data numerik dibedakan menjadi dua macam kelompok, tipe data integer dan tipe data floating point. Tipe data integer untuk data bilangan bulat, tipe data floating point untuk data bilangan desimal.

2) Tipe Data String

Pada tipe data ini dapat berisi nilai string (alphanumeric/karakter) dan numerik. Nilai numerik disini tidak dapat untuk operasi perhitungan sebelum dilakukan konversi.

3) Tipe Data Tanggal

Tipe data ini menyimpan informasi waktu, baik tanggal maupun jam.

4) Tipe Data Char () dan Varchar ()

Tipe Data Char () dan Varchar () pada dasarnya adalah sama, yang membedakan dari keduanya adalah jumlah memori untuk menyimpan data. Memori yang dibutuhkan untuk menyimpan data bertipe Char () bersifat statis atau besarnya bergantung pada jumlah karakter yang ditetapkan pada saat field tersebut dideklarasikan. Sementara pada tipe data Varchar (), besarnya memori penyimpanan bergantung pada berapa karakter yang di pakai ditambah 1 byte yang berisi data jumlah karakter yang dipakai.

**Kelebihan-kelebihan MySQL**

1. Dapat bekerja di beberapa platform yang berbeda, seperti LINUX, Windows, MacOS dll.
2. Dapat dikoneksikan pada bahasa C, C++, Java, Perl, PHP dan Python.
3. Memiliki lebih banyak tipe data seperti : signed/unsigned integer yang memiliki panjang data sebesar 1,2,3,4 dan 8 byte, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET dan tipe ENUM.
4. Mendukung penuh terhadap kalimat SQL GROUP BY dan ORDER BY. Mendukung terhadap fungsi penuh ( COUNT ( ),COUNT (DISTINCT), AVG ( ), STD ( ), SUM ( ), MAX ( ) AND MIN ( ) ).

5. Mendukung terhadap LEFT OUTER JOIN dengan ANSI SQL dan sintak ODBC.
6. Menggunakan GNU automake, autoconf, dan LIBTOOL untuk portabilitas.
7. Kita dapat menggabungkan beberapa tabel dari database yang berbeda dalam *query* yang sama.
8. Ditulis dengan menggunakan bahasa C dan C++. Diuji oleh compiler yang sangat jauh berbeda.
9. Privilege (hak) dan password sangat fleksibel dan aman serta mengujikan 'Host-Based' Verifikasi.

Kekurangan MySQL:

1. Untuk koneksi ke bahasa pemrograman visual seperti VB (*Visual Basic*), Delphi, dan Foxpro, MySQL kurang support karena koneksi ini menyebabkan field yang dibaca harus sesuai dengan koneksi dari program visual tersebut, dan ini yang menyebabkan MySQL jarang dipakai dalam program Visual.
2. Data yang ditangani belum begitu besar, kurang lebih 100 Gigabyte data.

## **2.6 Perangkat Pemodelan Sistem dalam Pembuatan Suatu Program Aplikasi**


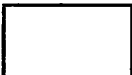
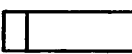
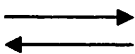
Dalam perancangan suatu sistem informasi memerlukan suatu pemodelan sistem untuk menggambarkan dan mengkomunikasikan secara sederhana rancangan sistem yang telah dibuat, agar sistem mudah dipahami dan dikoreksi. Dalam tahap-tahap ini dilakukan pemecahan masalah secara logika dengan menggunakan alat bantu, yaitu DFD, ERD dan Normalisasi.

### **2.6.1 DFD ( *Data Flow Diagram* )**

Data Flow Diagram adalah gambaran sistem secara logika. Gambaran ini tidak tergantung perangkat keras, perangkat lunak, struktur data atau organisasi file. Keuntungan dari data flow yaitu, memudahkan pemakai atau user yang kurang menguasai bidang komputer untuk

mengerti sistem yang akan dikerjakan atau dikembangkan . Ada 4 simbol data flow diagram, yaitu :

Tabel 2.2 Tabel simbol data flow diagram (sumber: Jogiyanto 2005)

Simbol-simbol	Keterangan
	Proses Menunjukkan tranformasi dari masukan
	Kesatuan luar ( <i>Entity</i> )
	Penyimpanan data Untuk menyimpan data atau tempat menemukan data
	Aliran data Menunjukkan arah dari dalam sistem

Proses pada data flow diagram dapat merupakan sekumpulan program, satu modul atau sub progam, dapat juga merupakan file elemen dari satu database atau satu bagian record, penyimpanan data dapat juga beupa disket berupa *magnetic drum* dan *magnetic tape*.

### 2.6.2 *Entity Relationship (ERD)*

Menurut Edi Winarko (2006:13) Entity Relationship Diagram (ER-Diagram) adalah sebuah diagram yang menggambarkan hubungan atau relasi antar entitas (Entity), setiap entity terdiri atas satu atau lebih atribut yang merepresentasikan seluruh kondisi atau fakta dari dunia nyata yang ditinjau. Dengan ER-Diagram untuk mentransformasikan keadaan dari dunia nyata ke dalam bentuk basis data. Dalam pembahasan

tentang ER-Diagram, terdapat beberapa komponen yang terkait dan perlu dibahas:

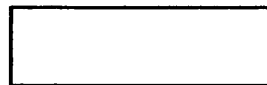
1. Entitas



Gambar 2.12. Simbol Entitas (Sumber : winarko, 2006)

Dilambangkan dengan lingkaran elipse dengan keterangan nama field didalamnya. Entitas memiliki fungsi sebagai simbol untuk identitas nama field yang ada dalam tabel.

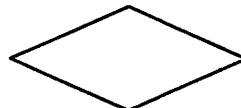
2. Tabel



Gambar 2.13. Simbol Tabel (Sumber : winarko, 2006)

Dilambangkan dengan persegi panjang dengan keterangan nama label di dalamnya. Simbol ini akan berhubungan langsung dengan entitas dan penghubung.

3. Penghubung.



Gambar 2.14. Simbol Penghubung (Sumber : winarko, 2006)

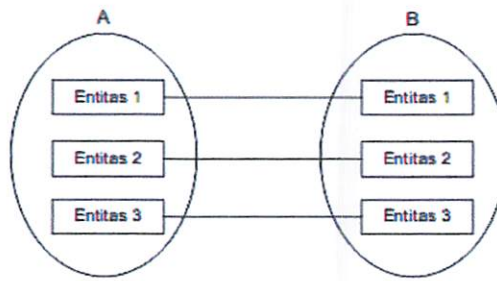
Dilambangkan dengan belah ketupat yang akan berhubungan dengan entitas yang menghubungkan antar tabel.

### 2.6.2.1 Kardinalitas atau Derajat Relasi

Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dapat berupa:

a. Satu ke satu (One to One)

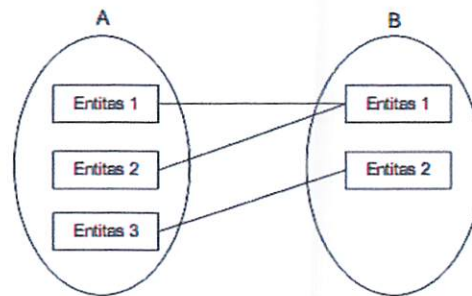
Setiap setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B, dan begitu sebaliknya setiap entitas pada himpunan entitas A berhubungan dengan paling banyak dengan entitas A.



Gambar 2.15. Relasi satu ke satu (Sumber: Fathansyah , 1999)

b. Satu ke banyak (One to Many)

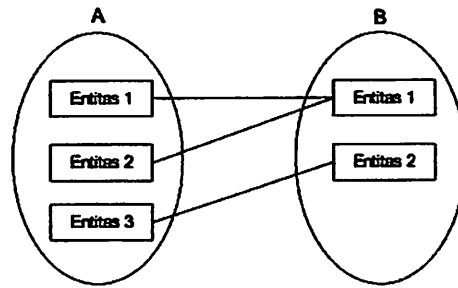
Setiap setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi tidak sebaliknya , dimana setiap entitas pada himpunan entitas B behubungan dengan paling banyak dengan satu entitas pada himpunan entitas A.



Gambar 2.16. Relasi satu ke satu (Sumber: Fathansyah , 1999)

c. Banyak ke Satu (Many to One)

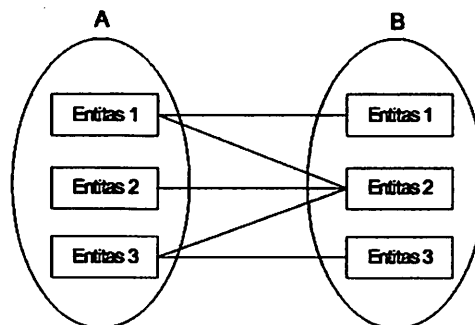
Setiap entitas pada himpunan entitas A behubungan paling banyak dengan satu entitas pada himpunan entitas B, tetapi tidak sebaliknya , dimana setiap entitas pada himpunan entitas A berhubungan degan paling banyak satu entitas pada himpunan entitas B.



Gambar 2. 17. Relasi banyak satu (Sumber: Fathansyah , 1999)

d. Banyak ke banyak (Many to Many)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan demikian juga sebaliknya, dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Gambar 2. 18. Relasi banyak ke banyak  
(Sumber: Fathansyah , 1999)

### 2.6.3 Pengertian Data Base

Sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi atau tugas khusus) yang saling berhubungan dan saling bersama-sama bertujuan untuk memenuhi suatu proses atau pekerjaan tertentu (Fathansyah, 1999). Sedangkan basis dapat didefinisikan dalam sejumlah sudut pandang (Fathansyah, 1999), yaitu:

1. Himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file atau tabel atau arsip yang berhubungan yang disimpan dalam media penyimpanan elektronik.

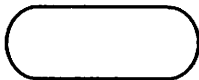
#### **2.6.4 Bagan Alir (*Flowchart*)**

Bagan alir (*Flowchart*) dapat didefinisikan sebagai sebuah bagan (*chart*) yang menunjukkan aliran di dalam program atau prosedur sistem secara logika (Jogianto, 2005).

##### **2.6.4.1 Simbol-simbol dalam *Flowchart***

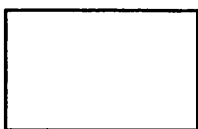
Simbol-simbol *flowchart* ada beberapa macam, yaitu :

**Simbol titik terminal**



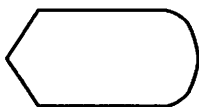
digunakan untuk menunjukkan awal dan akhir dari suatu proses

**Simbol proses**



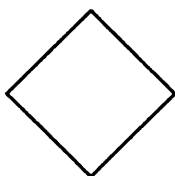
digunakan untuk menunjukkan proses dari operasi program

**Simbol display**



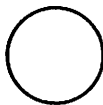
digunakan untuk menunjukkan output yang ditampilkan di monitor

**Simbol keputusan**



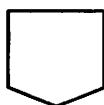
digunakan untuk menunjukkan suatu penyeleksian kondisi di dalam program

**Simbol On-Page  
Connector**



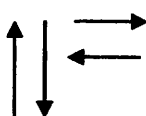
digunakan untuk menghubungkan suatu langkah dengan langkah lain dari flowchart dalam satu halaman

**Simbol Off-Page  
Connector**



digunakan untuk menghubungkan suatu langkah dengan langkah lain dari flowchart dalam halaman yang berbeda

**Simbol garis alir**



digunakan untuk menghubungkan setiap langkah dalam flowchart dan menunjukkan kemana arah aliran diagram

**Simbol pita magnetik**



digunakan untuk menunjukkan input / output menggunakan pita magnetik

**Simbol drum magnetik**



digunakan untuk menunjukkan input / output menggunakan drum magnetik

**Simbol pita kertas  
berlubang**



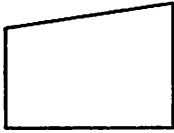
digunakan untuk menunjukkan input / output menggunakan pita kertas berlubang

**Simbol kartu prolog**



digunakan untuk menunjukkan input / output menggunakan kartu plong

**Simbol keyboard**



digunakan untuk menunjukkan input yang menggunakan on line keyboard

**Simbol dokumen**



digunakan untuk menunjukkan dokumen input dan output baik untuk proses manual, mekanik, atau komputer

**Simbol input /  
output**



digunakan untuk mewakili data input / output

**Simbol proses  
terdefinisi**



digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain

**Simbol persiapan**



digunakan untuk memberi nilai awal suatu besaran

**Gambar 2.19 Simbol-simbol *Flowchart* (Sumber Jogianto, 2005)**

### 2.6.5 Teknik Normalisasi

Proses **Normalisasi**, merupakan proses pengelompokan data elemen menjadi *table-table* yang menunjukkan *entity* dan relasinya.

#### a. Bentuk Tak Normal

Merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya (Harianto Kristanto. Ir, Konsep dan Perencanaan Database, Andi Offset, Yogyakarta, 2004).

#### b. Bentuk Normal Pertama

Bentuk normal pertama memiliki ciri yaitu setiap data dibentuk dalam flat file (file datar/rata), data bentuk, dalam satu record demi satu record. Tidak ada set atribut yang berulang-ulang atribut bernilai ganda (Harianto Krintanto. Ir, Konsep dan Perancangan database, Andi Offset, Yogyakarta, 2004).

#### c. Bentuk Normal Kedua

Mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama. Sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci field. Kunci field haruslah unik dan mewakili atribut lain yang menjadi anggotanya (Harianto Kristanto. Ir, Konsep dan Perancangan Database, Andi Offset, Yogyakarta, 2004)

#### 2.6.5.1 Atribut Kunci

Setiap file selalu terdapat kunci dari file berupa satu field atau satu set field yang dapat mewakili record. Misalnya no kerusakan merupakan kunci dari tabel kerusakan. Kunci terdiri dari bermacam-macam diantaranya adalah sebagai berikut:

##### a. Kunci field (primary key)

Kunci primer adalah satu atribut atau minimal satu set atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian

spesifik, tapi juga dapat mewakili setiap kejadian dari satu entity baru.

b. Kunci kandidat (candidate key)

Kunci kandidat adalah satu atribut atau minimal satu set atribut yang mengidentifikasi secara unik satu kejadian spesifik dari entity.

c. Kunci alternatif (alternate key)

Kunci alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci primer. Sering kunci alternatif dipakai sebagai kunci pengurutan dalam laporan.

d. Kunci tamu (foreign key)

Kunci tamu adalah satu set atribut (set atribut) yang melengkapi satu hubungan (relationship) yang menunjukkan ke induknya. Kunci tamu ditempatkan pada entity anak dan sama dengan kunci primer induk yang tidak direlasikan. Hubungan antara entity induk dengan entity anak adalah hubungan satu lawan banyak (one to many relationship).

## 2.7 Mesin Mobil Diesel

Mesin diesel adalah sejenis mesin pembakaran dalam; lebih spesifik lagi, sebuah mesin pemicu kompresi, dimana bahan bakar dinyalakan oleh suhu tinggi gas yang dikompresi, dan bukan oleh alat berenergi lain (seperti busi). Mesin ini ditemukan pada tahun 1892 oleh Rudolf Diesel, yang menerima paten pada 23 Februari 1893. Diesel menginginkan sebuah mesin untuk dapat digunakan dengan berbagai macam bahan bakar termasuk debu batu bara. Dia mempertunjukkannya pada Exposition Universelle (Pameran Dunia) tahun 1900 dengan menggunakan minyak kacang (lihat biodiesel). Kemudian diperbaiki dan disempurnakan oleh Charles F. Kettering. (<http://id.wikipedia.org/>)

Mesin mobil diesel terdiri dari *Swirl Chamber Type Engine* dan *Direct Injection Type Engine*. Kesemuanya mempunyai komponen Glow

Plug yaitu perangkat yang digunakan untuk memanaskan kompresi udara di ruang pembakaran pada saat mobil dinyalakan (starter). *Swirl Chamber* adalah suatu kondisi bahan bakar yang masuk secara cepat terurai oleh putaran udara yang kuat sehingga menimbulkan pengapian singkat. *Combustion Chamber* adalah suatu ruang yang injection nozzle secara langsung menyemprotkan bahan bakar.

Mesin diesel lebih memiliki efisiensi bahan bakar lebih baik bila dibandingkan mesin bensin karena kenyataannya mesin diesel mempunyai rasio kompresi yang lebih tinggi. Keuntungan lain adalah bahwa, mesin diesel dapat lebih mudah turbocharged dari mesin bensin karena fakta bahwa jika rasio kompresi dan tekanan dalam silinder tinggi selama stroke inlet, campuran mulai menyala dengan cepat, sedangkan piston dalam perjalanan ke atas. Mesin diesel tidak memiliki bahan bakar dalam silinder dan dengan demikian memungkinkan turbocharger untuk menyedot udara lebih banyak tanpa terjadi masalah. (<http://forum.kompas.com/otomotif-umum>) 27-03-12.

### 2.7.1 Prinsip Kerja Mesin Diesel.

Mesin/motor diesel (diesel engine) merupakan salah satu bentuk motor pembakaran dalam (internal combustion engine) di samping motor bensin dan turbin gas. Motor diesel disebut dengan motor penyalaan kompresi (compression ignition engine) karena penyalaan bahan bakarnya diakibatkan oleh suhu kompresi udara dalam ruang bakar. Dilain pihak motor bensin disebut motor penyalaan busi (spark ignition engine) karena penyalaan bahan bakar diakibatkan oleh percikan bunga api listrik dari busi.

Cara pembakaran dan pengatomisasian (atomizing) bahan bakar pada motor diesel tidak sama dengan motor bensin. Pada motor bensin campuran bahan bakar dan udara melalui karburator dimasukkan ke dalam silinder dan dibakar oleh nyala listrik dari busi. Pada motor diesel yang diisap oleh torak dan dimasukkan ke dalam ruang bakar hanya udara, yang

selanjutnya udara tersebut dikompresikan sampai mencapai suhu dan tekanan yang tinggi. Beberapa saat sebelum torak mencapai titik mati atas (TMA) bahan bakar solar diinjeksikan ke dalam ruang bakar. Dengan suhu dan tekanan udara dalam silinder yang cukup tinggi maka partikel-partikel bahan bakar akan menyala dengan sendirinya sehingga membentuk proses pembakaran. Agar bahan bakar solar dapat terbakar sendiri, maka diperlukan rasio kompresi 15-22 dan suhu udara kompresi kira-kira 600°C.

Meskipun untuk motor diesel tidak diperlukan system pengapian seperti halnya pada motor bensin, namun dalam motor diesel diperlukan sistem injeksi bahan bakar yang berupapompa injeksi (injection pump) dan pengabut (injector) serta perlengkapan bantu lain. Bahan bakar yang disemprotkan harus mempunyai sifat dapat terbakar sendiri (self ignition).

<http://www.agussuwasono.com/artikel/mechanical/426-teori-dasar-mesin-diesel.html>



## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Alat dan Bahan Penelitian.**

##### **3.1.1 Alat Penelitian**

Penelitian ini dilaksanakan dengan menggunakan beberapa perangkat keras dan perangkat lunak, diantaranya yaitu:

1. Perangkat keras
  - a. Prosesor Intel Pentium Dual-Core 2.0 GHz
  - b. RAM 1 GB
  - c. VGA 256 MB 32 Bit
  - d. Harddisk 250 GB
2. Perangkat lunak
  - a. MySql 1.6.4
  - b. LAZARUS / Free Pascal

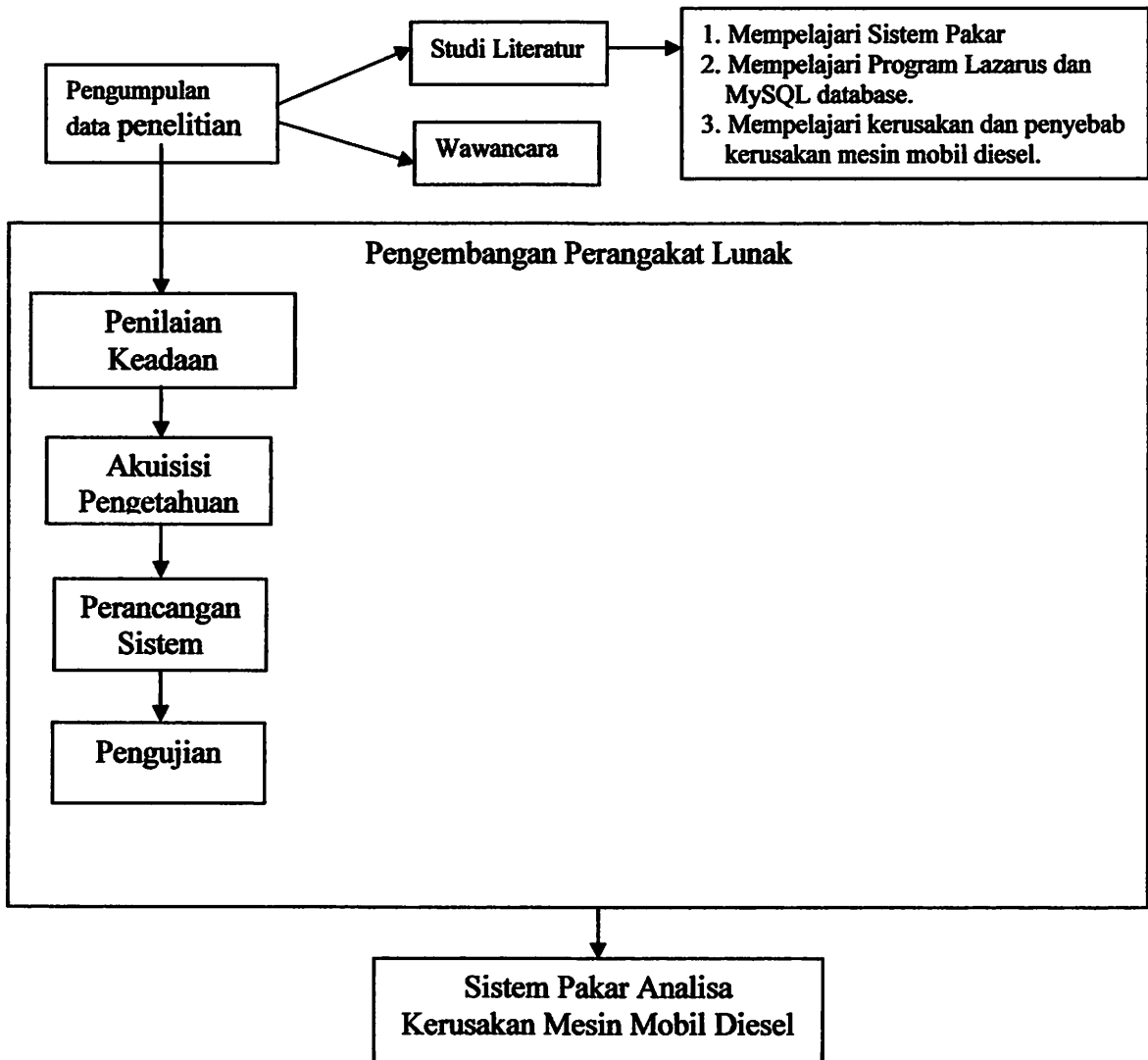
##### **3. Sistem Operasi**

Sistem operasi yang digunakan adalah Microsoft Windows 7 Ultimate 32-bit.

##### **3.1.2 Bahan Penelitian**

Data yang menjadi input sistem adalah kerusakan dan sebab dari kerusakan mesin mobil diesel. Pengguna terlebih dahulu melakukan login kemudian diberikan pilihan berupa data-data kerusakan yang dialami. Sedangkan output sistem adalah penyebab dari kerusakan yang terjadi pada mesin mobil diesel yang dialami.

### 3.2 Rancangan Penelitian.



Gambar 3.1 Rancangan Penelitian

### 3.3 Metode Penelitian.

Metode yang digunakan dalam penelitian ini meliputi metode pengumpulan data dan metode pengembangan perangkat lunak.

#### 3.3.1 Metode Pengumpulan Data.

##### a. Studi Literatur

Yaitu dengan melakukan studi mengenai sistem pakar, tools yang akan digunakan, dan mesin mobil diesel dan

kerusakannya melalui literatur–literatur seperti buku, jurnal, dan sumber ilmiah lain seperti buku, google, artikel, dan dokumen teks yang berhubungan.

**b. Wawancara.**

Yaitu dengan melakukan wawancara kepada pakar atau teknisi mesin mobil diesel, untuk memperoleh data yang diperlukan untuk penelitian dan pembangunan perangkat lunak.

### **3.3.2 Metode Pengembangan Perangkat Lunak**

Perancangan sistem dimulai dengan mengidentifikasi permasalahan hingga pengujian. Secara garis besar ada 4 tahap yaitu identifikasi, konseptualisasi, formalisasi atau rancangan dan pengujian. Dikarenakan tiap-tiap tahap saling berhubungan dan saling menunjang, maka tahap-tahap tadi harus dikerjakan secara berurutan satu sama lain. Sistem sederhana yang akan dirancang ini merupakan bagian kecil dari sistem analisa secara keseluruhan. Sedangkan permasalahan spesifik yang akan diangkat adalah mengenai kerusakan mesin mobil diesel mulai dari bagian-bagian mesin, kerusakan yang sering dialami dan penyebab gangguan pada mesin.

**a. Tahap Identifikasi**

Pada tahap ini pengidentifikasian permasalahan yang akan dibuat sistem pakar berkaitan dan terbatas pada kerusakan mesin. Knowledge engineer harus menentukan batasan-batasan permasalahan yang bersifat spesifik dan bersifat umum tentang bagian-bagian mesin dan troubleshooting yang sering timbul pada bagian-bagian tersebut.

**b. Tahap Konseptualisasi**

Dalam tahap konseptualisasi ini ditentukan unsur-unsur apa saja yang terkait dari bagian gejala-gejala serta penyebab yang biasa di timbulkan.

**Berikut ini adalah data-data kerusakan dan penyebab gangguan:**

No	Data Kerusakan	Kemungkinan penyebab kerusakan/gangguan	Cara Mengatasi
1	Sistem injeksi bahan bakar tidak berfungsi dengan baik.	• Bahan bakar Bocor dari pipa tekanan tinggi.	Keraskan Mur Pengikat pipa tekanan tinggi / ganti dengan pipa tekanan yang baru
		• Nozzle rusak	Ganti dengan nozzle yang baru
		• Ada udara pada saluran bahan bakar	Keluarkan udara dari saluran bahan bakar dengan cara memompakan priming pump
		• Saat penginjeksian bahan bakar terlambat	Setel pada penginjeksian bahan bakar
		• Control rack tidak berfungsi	Perbaiki mekanisme control rack
		• Automatic timer tidak tepat	Setel automatic timer
2	Gangguan pada saluran sistem bahan bakar.	• Saringan pada pompa pemindah tersumbat	Bongkar pompa pemindah dan bersihkan saringannya
		• Saringan bahan	Ganti elemen saringan bahan bakar

		bakar tersumbat	
		• Tangki bahan bakar kotor	Bersihkan tangki bahan bakar
		• Kemampuan pompa pemindah yang sangat rendah	Perbaiki / ganti
		• Injeksi bahan bakar tidak tepat	Setel saat penginjeksian dengan cara menggeserkan pompa injeksi
		• Jumlah bahan bakar yang diinjeksikan ke silinder tidak sama	Kalibrasi pompa injeksi pada test bench
		• Governor kurang baik	Setel pompa injeksi pada test bench
		• Ada angin pada saluran bahan bakar	Bleeding
		• Bahan bakar bocor pada pipa tekanan tinggi	Keraskan mur pengikat pipa tekanan tinggi / ganti pipa tekanan tinggi
3	Selama Mesin Hidup Banyak Mengeluarkan Asap	• Saat penginjeksian bahan bakar terlalu cepat	Tepatkan saat penginjeksian sesuai spesifikasi
		• Bahan bakar bercampur air	Ganti bahan bakar
		• Cincin torak dan dinding silinder bocor sehingga minyak pelumas	Ganti cincin torak dan perbaiki dinding silindernya

		naik ke ruang bahan bakar	
		<ul style="list-style-type: none"> <li>Oli silinder katup bocor, sehingga minyak pelumas masuk ke ruang bakar</li> </ul>	Ganti
		<ul style="list-style-type: none"> <li>Bahan bakar yang diinjeksikan terlalu banyak</li> </ul>	Kalibrasi pompa injeksi pada test bench
		<ul style="list-style-type: none"> <li>Jumlah bahan bakar yang diinjeksikan ke dalam silinder tidak sama</li> </ul>	Kalibrasi pompa injeksi pada test bench
		<ul style="list-style-type: none"> <li>Udara yang masuk kedalam silinder terlalu sedikit karena saringan udaranya tersumbat</li> </ul>	Bersihkan saringan / ganti
		<ul style="list-style-type: none"> <li>Oli mesin terpompa ke atas</li> </ul>	Periksa permukaan oli dan sesuaikan jumlahnya
4	Selama Mesin di Starter Mengeluarkan banyak Asap	<ul style="list-style-type: none"> <li>Timer injektor timing terlalu cepat / lambat</li> </ul>	Bongkar automatic timer dan setel
		<ul style="list-style-type: none"> <li>Pemasangan pompa injeksi terhadap saat penginjeksian tidak tepat</li> </ul>	Periksa pemasangan pompa injeksi dan tempatkan saat penginjeksian (sesuai spesifikasi)

		<ul style="list-style-type: none"> <li>• Atomisasi bahan bakar kurang baik</li> </ul>	Perbaiki dan ganti nozzle
		<ul style="list-style-type: none"> <li>• Nozzle rusak</li> </ul>	Ganti Nozzle
		<ul style="list-style-type: none"> <li>• Kotoran (karbon) berkumpul pada nozzle needle</li> </ul>	Bersihkan dan ganti jika rusak
		<ul style="list-style-type: none"> <li>• Sekrup control pinion clamp lepas / longgar</li> </ul>	Setel posisi control pinion dan keraskan sekrupnya
		<ul style="list-style-type: none"> <li>• Delivery value rusak</li> </ul>	Ganti
		<ul style="list-style-type: none"> <li>• Delivery value spring putus</li> </ul>	Ganti pegas
		<ul style="list-style-type: none"> <li>• Setelan full load capsule tidak baik</li> </ul>	Setel full load capsule
		<ul style="list-style-type: none"> <li>• Pneumatic governor link / stopper aus</li> </ul>	Perbaiki / ganti mekanisme governor pneumatic
5	Mesin Asapnya Banyak, tetapi Mesin Tidak Mau Menyala	<ul style="list-style-type: none"> <li>• Saat penginjeksian bahan bakar terlambat</li> </ul>	Setel saat penginjeksian
		<ul style="list-style-type: none"> <li>• Timer lock nut longgar atau lepas</li> </ul>	Kerusakan mur pengunci timer
		<ul style="list-style-type: none"> <li>• Atomisasi bahan bakar tidak baik</li> </ul>	Perbaiki dan setel nozzle / ganti
		<ul style="list-style-type: none"> <li>• Kotoran (karbon) berkumpul pada nozzle needle</li> </ul>	Bersihkan nozzle / ganti jika rusak
		<ul style="list-style-type: none"> <li>• Angka cetane terlalu rendah</li> </ul>	Ganti jenis bahan bakar
		<ul style="list-style-type: none"> <li>• Bahan bakar tercampur air</li> </ul>	Ganti bahan bakar

6	Mesin Susah dihidupkan	<ul style="list-style-type: none"> <li>Bahan bakar tidak sampai ke pompa injeksi / bocor</li> </ul>	Periksa saluran bahan bakar dan perbaiki jika bocor
		<ul style="list-style-type: none"> <li>Saluran bahan bakar tersumbat oleh udara</li> </ul>	Bleeding
		<ul style="list-style-type: none"> <li>Saringan bahan bakar dan saluran bahan bakar tersumbat</li> </ul>	Bersihkan / ganti saringan bahan bakar dan saluran bahan bakar
		<ul style="list-style-type: none"> <li>Kemampuan pompa pemindah menurun / tidak sesuai spesifikasi</li> </ul>	Perbaiki / ganti pompa pemindah
		<ul style="list-style-type: none"> <li>Pipa tekanan tinggi lepas, longgar / pecah</li> </ul>	Kerusakan mur / ganti
		<ul style="list-style-type: none"> <li>Control rack tidak mencapai posisi tempat bahan bakar yang diinjeksikan</li> </ul>	Perbaiki control rack dan perbaiki pula mekanismenya
		<ul style="list-style-type: none"> <li>Delivery valve rusak</li> </ul>	Ganti
		<ul style="list-style-type: none"> <li>Jumlah bahan bakar yang diinjeksikan terlalu sedikit, karena pompa pemindah rusak</li> </ul>	Perbaiki / Ganti pompa pemindah
		<ul style="list-style-type: none"> <li>Tekanan penginjeksian bahan bakar terlalu rendah</li> </ul>	Setel / kalibrasi nozzle sesuai spesifikasi
		<ul style="list-style-type: none"> <li>Saat penginjeksian bahan bakar terlalu rendah</li> </ul>	Setel saat penginjeksian

		<ul style="list-style-type: none"> <li>• Governor / full load capsule setelahnya kurang baik</li> </ul>	Setel pompa injeksi pada test bench
7	Putaran Mesin Kasar, Khususnya Pada Putaran Idle	<ul style="list-style-type: none"> <li>• Jam bahan bakar yang diinjeksikan ke dalam silinder tidak sama</li> </ul>	Kalibrasi pompa injeksi pada test bench
		<ul style="list-style-type: none"> <li>• Control rack tidak berfungsi</li> </ul>	Bongkar control rack dan setel mekanismenya
		<ul style="list-style-type: none"> <li>• Idling spring capsule setelahnya kurang baik</li> </ul>	Setel
		<ul style="list-style-type: none"> <li>• Diafragma governor tidak berfungsi</li> </ul>	Periksa dan ganti jika rusak
		<ul style="list-style-type: none"> <li>• Atomisasi Nozzle kurang baik / tidak tepat</li> </ul>	Stel nozzle dan ganti jika rusak
8	Engine Knocking / Detonasi	<ul style="list-style-type: none"> <li>• Atomisasi nozzle kurang baik / tekanan penginjeksian tidak tepat</li> </ul>	Overhaul nozzle, bersihkan komponen-komponen dan ganti jika rusak, kemudian stel tekanan nozzle sesuai spesifikasi
		<ul style="list-style-type: none"> <li>• Bahan bakar tidak tepat (angka cetane rendah)</li> </ul>	Ganti dengan bahan bakar yang sesuai
		<ul style="list-style-type: none"> <li>• Saat injeksi bahan bakar terlalu cepat</li> </ul>	Setel pemasangan pompa terhadap mesin
		<ul style="list-style-type: none"> <li>• Jumlah bahan bakar yang diinjeksikan ke dalam silinder tidak</li> </ul>	Kalibrasi pompa injeksi pada test bench

		rata	
9	Mesin Panas	<ul style="list-style-type: none"> <li>• Air pendingin kurang</li> </ul>	Cek kebocoran pada system pendingin / tambah air pendingin
		<ul style="list-style-type: none"> <li>• Oli bocor dari fan coupling.</li> </ul>	Periksa dan perbaiki apa bila ada yang rusak ganti dengan yang baru
		<ul style="list-style-type: none"> <li>• Tali kipas kendur</li> </ul>	Setel ulang tali kipas / ganti apa bila sudah aus
		<ul style="list-style-type: none"> <li>• Tutup radiator rusak atau radiator core tersumbat</li> </ul>	Ganti tutup radiator / perbaiki radiator core yang tersubat
		<ul style="list-style-type: none"> <li>• Pompa air rusak</li> </ul>	Ganti dengan yang baru
		<ul style="list-style-type: none"> <li>• Sealing cap rusak</li> </ul>	Ganti dengan yang baru
		<ul style="list-style-type: none"> <li>• Sistem pendingin tersumbat oleh kotoran</li> </ul>	Perbaiki dan bersikan hingga tidak tersumbat lagi
		<ul style="list-style-type: none"> <li>• Penyetelan timing pengabutan bahan bakar tidak tepat.</li> </ul>	Setel / kalibrasi nozzle sesuai spesifikasi
10	Tekanan oli tidak naik	<ul style="list-style-type: none"> <li>• Kekentalan oli mesin tidak baik</li> </ul>	Ganti dengan yang baru sesuai spesifikasi
		<ul style="list-style-type: none"> <li>• Oli mesin kurang</li> </ul>	Tambah Oli mesin yang sama pada awal digunakan
		<ul style="list-style-type: none"> <li>• Meter tekanan oli rusak</li> </ul>	Ganti dengan yang baru
		<ul style="list-style-type: none"> <li>• Relief valve aus atau spring bypass valve lemah</li> </ul>	Ganti dengan yang baru

	<ul style="list-style-type: none"> <li>• Saringan pompa oli tersumbat</li> </ul>	Bersihkan saringan pompa Oli hingga tidak tersumbat lagi
	<ul style="list-style-type: none"> <li>• Bagian yang berhubungan dengan pompa oli aus</li> </ul>	Periksa dan perbaiki bagian yang berhubungan dengan pompa oli / ganti pompa Oli
	<ul style="list-style-type: none"> <li>• Bushing rocker arm aus</li> </ul>	Ganti dengan yang baru
	<ul style="list-style-type: none"> <li>• Camshaft dan bearing camshaft aus</li> </ul>	Ganti dengan yang baru
	<ul style="list-style-type: none"> <li>• Poros engkol dan bantalan aus</li> </ul>	Ganti dengan yang baru

Data-data di atas adalah data tentang berbagai kerusakan mesin pada mobil diesel beserta penyebab gangguan yang ditimbulkan. Data tersebut merupakan data awal sebagai inputan dari sistem sebelum di proses menjadi data output.

#### **c. Tahap Rancangan**

Di dalam tahap rancangan ini semua permasalahan yang saling berelasi atau berhubungan akan diformulasikan sesuai dengan software/ bahasa pemrograman yang akan digunakan untuk memaparkan hubungan relasional tersebut sesuai dengan bentuk format yang digunakan oleh sistem analisa. Dalam tahap ini sering disebut juga basis pengetahuan.

#### **d. Tahap Pengujian**

Sistem yang telah selesai akan diuji. Apabila sistem yang telah dibentuk ini masih kurang optimal, maka akan dilakukan perbaikan-perbaikan sampai semua permasalahan dapat teratasi,

lengkap dan akurat. Sistem yang telah diperbaiki akan diuji kembali sampai sistem itu benar-benar lengkap dan akurat serta layak digunakan.

### **3.4 Analisis Sistem.**

#### **3.4.1 Perancangan Basis Data**

Hampir semua operasi dalam komputer berhubungan dengan pengolahan data dan sebagian besar program yang berhubungan dengan operasi data pasti menggunakan database sebagai tempat penyimpanan dan pengolahan data.

Ada beberapa hal yang objektif yang perlu dipertimbangkan dalam membangun suatu program yang mengolah data dalam jumlah besar, diantaranya adalah :

1. mampu mengeliminasi redudansi data, artinya data tidak perlu ditulis berulang-ulang tetapi hanya informasi singkat yang disimpan dengan benar ditempat yang benar.
2. mampu mencari lokasi atau tempat dimana suatu data disimpan, artinya data dapat dinavigasi dengan baik karena hal ini akan banyak membantu proses pelacakan data.
3. adanya kemudahan dalam mengimplementasikan database sehingga semua kesulitan dalam implementasi program dapat ditekan seminimal mungkin sehingga proses membangun suatu program pengolahan data menjadi lebih mudah.
2. Dalam membangun database terdapat sekumpulan tabel didalamnya yang menyimpan berbagai informasi tentang suatu topik pembahasan. Tabel-tabel tersebut sangatlah penting dalam melakukan proses pengolahan data.

Dalam merancang suatu database perlu diperhatikan cara untuk melakukan normalisasi data. Tujuannya untuk mengeliminasi redudansi data. Contoh redudansi data dapat digambarkan dalam database berikut :

Tabel 3.1 reduidansi data

Macam Kerusakan	Penyebab gangguan
Tekanan oli tidak naik	Meter tekanan oli rusak
Tekanan oli tidak naik	Saringan oli tersumbat
Tekanan oli tidak naik	Relief valve aus
Motor starter tidak berputar	Terminal kabel batere kendor
Motor starter tidak berputar	Batere lemah
Motor starter tidak berputar	Tali kipas kendor
Motor starter tidak berputar	Sekring utama konslet
-----	-----
-----	-----

Ada beberapa aturan umum yang menjadi pedoman dalam mengorganisasikan data menjadi tabel-tabel yang membentuk database. Pedoman berikut bukanlah pedoman yang mutlak harus diikuti karena dalam banyak hal rancangan suatu database lebih memperhatikan kasus dan bagaimana penyelesaiannya.

1. definisikan setiap topik atau bahasan untuk setiap tabel dan pastikan bahwa semua data dalam tabel terhubung dengan topik yang dimaksud.
2. jika ada informasi yang ditulis berulang-ulang dalam suatu tabel, pecahkan tabel tersebut menjadi beberapa tabel kemudian atur hubungan antar tabel sehingga tetap berelasi.
3. jangan menyimpan informasi dalam tabel jika informasi tersebut akan dioperasikan atau dikalkulasi dengan informasi pada tabel lain.
4. gunakan teknik normalisasi data untuk meningkatkan akurasi data yang diinputkan.

### 3.4.2 Normalisasi.

Proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan *entity* dan relasinya.

#### 3.4.2.1 Teknik Normalisasi

##### a. Bentuk tak normal

Merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

Tabel 3.2 Tabel Bentuk Tak Normal

Id_Kerusakan	Nm_Kerusakan	Id_Sebab	Nm_Sebab
A1	Starter tidak berputar	G1	Terminal batere kendor
A2	Starter tidak berputar	G2	Tali kipas rusak
A3	Mesin berputar tetapi tidak hidup		

##### b. Bentuk Normal Pertama

Bentuk normal pertama memiliki ciri yaitu setiap data dibentuk dalam file-file (file datar/rata), data dibentuk dalam satu record demi satu record. Tidak ada set atribut yang berulang-ulang

Tabel 3.3 Tabel Bentuk Normal Pertama

Id_Kerusakan	Nm_Kerusakan
A1	Starter tidak berputar
A2	Mesin berputar tetapi tidak hidup

<b>Id_Sebab</b>	<b>Nm_Sebab</b>
G1	Terminal batere kendor
G2	Tali kipas rusak

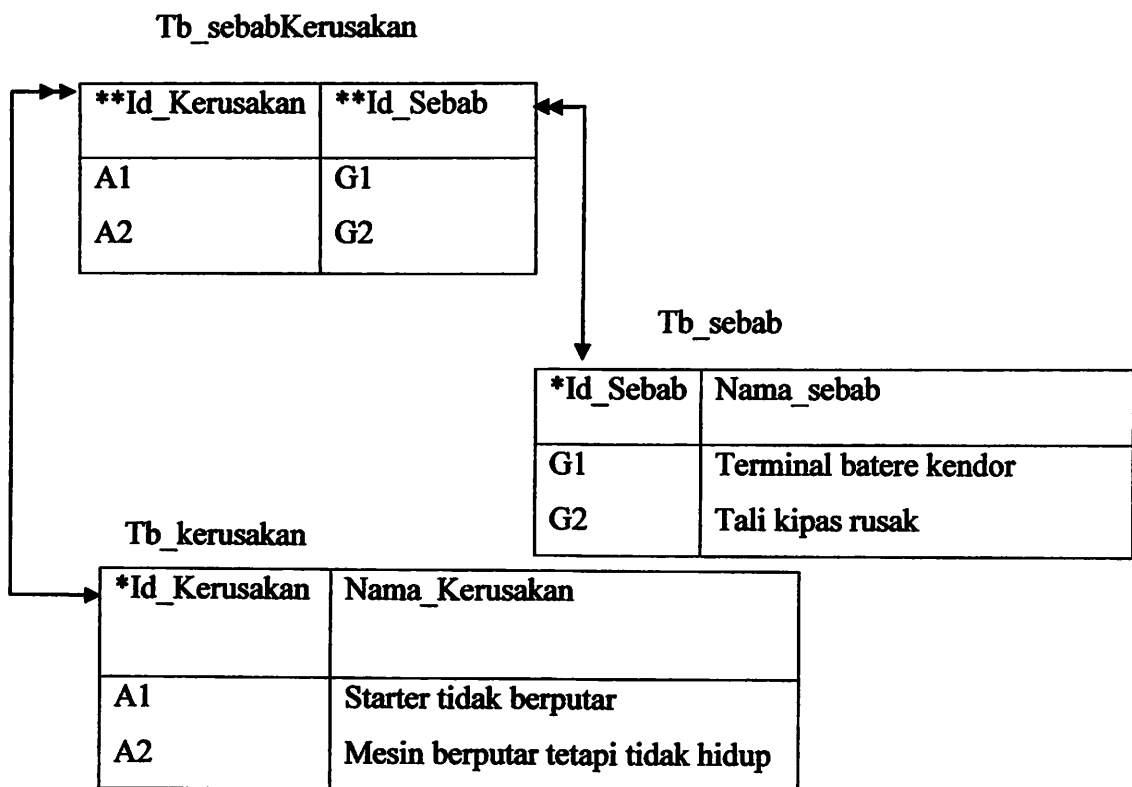
c. Bentuk Normal Kedua

Mempunyai syarat yaitu bentuk data telah memiliki kriteria bentuk normal kesatu. Atribut haruslah bergantung secara fungsi pada kunci utama. Sehingga untuk membentuk normal kedua haruslah sudah ditentukan kunci-kunci field. Kunci field haruslah unik dan dapat mewakili atribut lain yang menjadi anggotanya.

Tabel 3.4 Tabel bentuk normal kedua.

<b>Tb_kerusakan</b>	<b>Tb_sebab</b>	<b>Tb_sebabkerusakan</b>
*Id_kerusakan Nama_kerusakan	*Id_sebab Nama_sebab	**Id_kerusakan **Id_sebab

### 3.4.2.2 Entity Relation Diagram (ERD)



**Gambar 3.2 Relasi Antar Tabel**

**Keterangan :**

**\*** : Kunci Primer (Primary Key)

**\*\*** : Kunci Tamu (Foreign Key)

**↔** : Relasi one many.

### 3.4.3 Struktur Tabel

Berikut adalah penjelasan dari struktur basis data dari file yang terdapat pada relasi antar tabel

#### 1. Kerusakan

**Tabel 3.5 Tabel kerusakan**

No	Nama Field	Data Type	Width	Keterangan
1	Id_K	Varchar	10	Id kerusakan
2	Nama Kerusakan	Text		Nama kerusakan mesin

## 2. Sebab

Tabel 3.6 Sebab

No	Nama Field	Data Type	Width	Keterangan
1	Id_S	Varchar	10	Id Sebab
2	Nama Sebab	Text		Nama sebab kerusakan mesin

## 3. Sebab Kerusakan

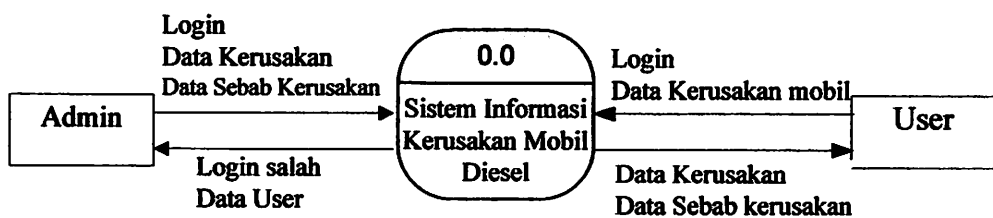
Tabel 3.9 Tabel Sebab Kerusakan

No	Nama Field	Data Type	Width	Keterangan
1	Id_S	Varchar	10	Id Sebab
2	Id_K	Varchar	10	Id Kerusakan

### 3.4.4 DFD (Data Flow Diagram)

#### 3.4.4.1 Context Diagram

Hubungan sistem dengan kesatuan luarnya dapat dilihat pada diagram konteks. Diagram konteks dari Sistem pakar untuk analisa kerusakan mesin mobil diesel dapat dilihat dari gambar berikut:

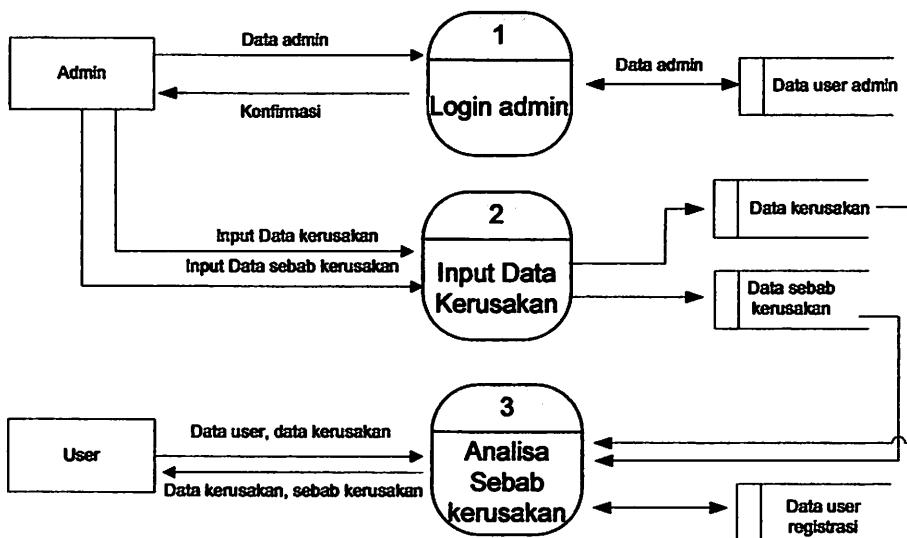


Gambar 3.3 Context Diagram

Diagram konteks diatas menerangkan bahwa arus data secara umum melibatkan dua buah *entitas*, yaitu; Admin dan User.

- a. Admin merupakan seseorang yang mengelola sistem yang bertugas memasukkan data-data kerusakan mobil, yang meliputi data kerusakan, data sebab kerusakan.
- b. User merupakan pengguna atau pemakai dari aplikasi sistem pakar untuk analisa kerusakan mesin mobil diesel.

### 3.4.4.2 DFD level 0



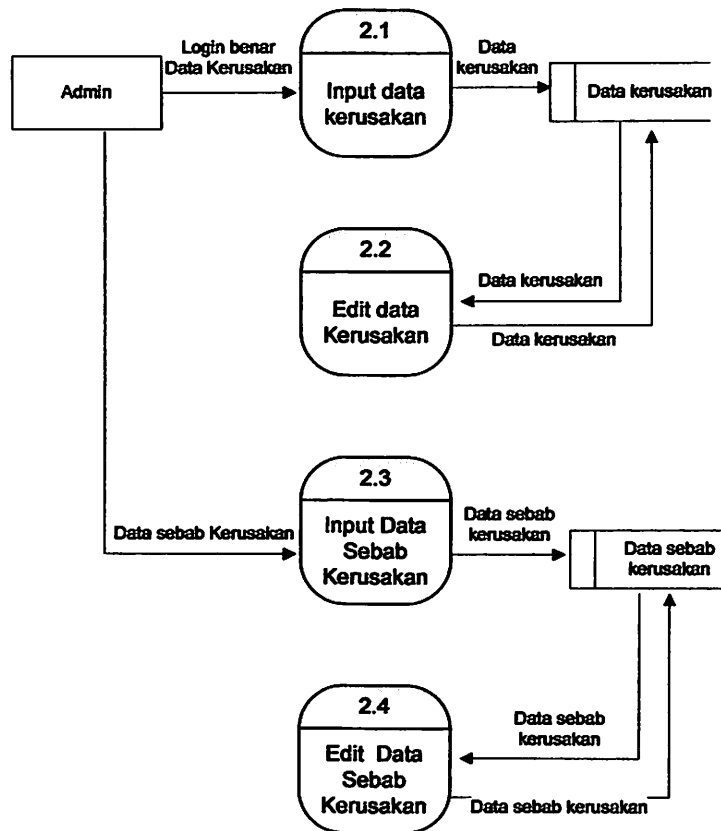
Gambar 3.4 *Data Flow Diagram Level 0*

Di dalam DFD level 0 terdapat 3 proses yaitu proses Login admin, proses input data kerusakan dan proses identifikasi sebab kerusakan.

DFD level 0 diatas mempunyai penjelasan sebagai berikut :

- a. Seorang admin harus login terlebih dahulu untuk memasuki sistem. Jika login benar maka bisa menginputkan data kerusakan, data sebab kerusakan.
- b. User bisa melihat semua informasi yang ada pada sistem pakar.

### 3.4.4.3 DFD level 1 proses input data kerusakan.



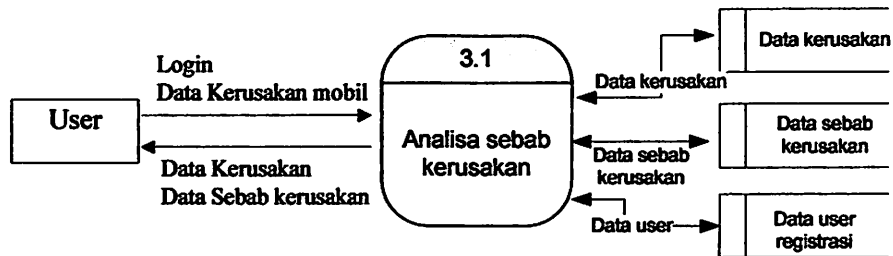
Gambar 3.5 DFD Level 1 Proses 2

Di dalam proses kerusakan terdiri dari 2 proses yaitu input data kerusakan dan edit data kerusakan. Proses ini dimulai dari admin yang mempunyai data kerusakan telah melakukan login dengan benar kemudian menginputkan data-data tentang kerusakan tersebut. Setelah diinputkan, data tersebut kemudian disimpan di database kerusakan. Setelah itu data tadi disimpan di database kerusakan. Setelah itu data tadi akan diproses dan kemudian akan disimpan lagi di database kerusakan.

Di dalam proses pendataan sebab kerusakan terdiri dari 2 proses yaitu data sebab kerusakan dan edit data sebab kerusakan. Proses ini dimulai dari admin yang mempunyai data sebab kerusakan telah melakukan login dengan benar kemudian menginputkan data-data tentang sebab kerusakan tersebut. Setelah

diinputkan, data tersebut kemudian disimpan di database sebab kerusakan. Setelah itu data tadi akan diproses dan kemudian akan disimpan lagi di database sebab kerusakan.

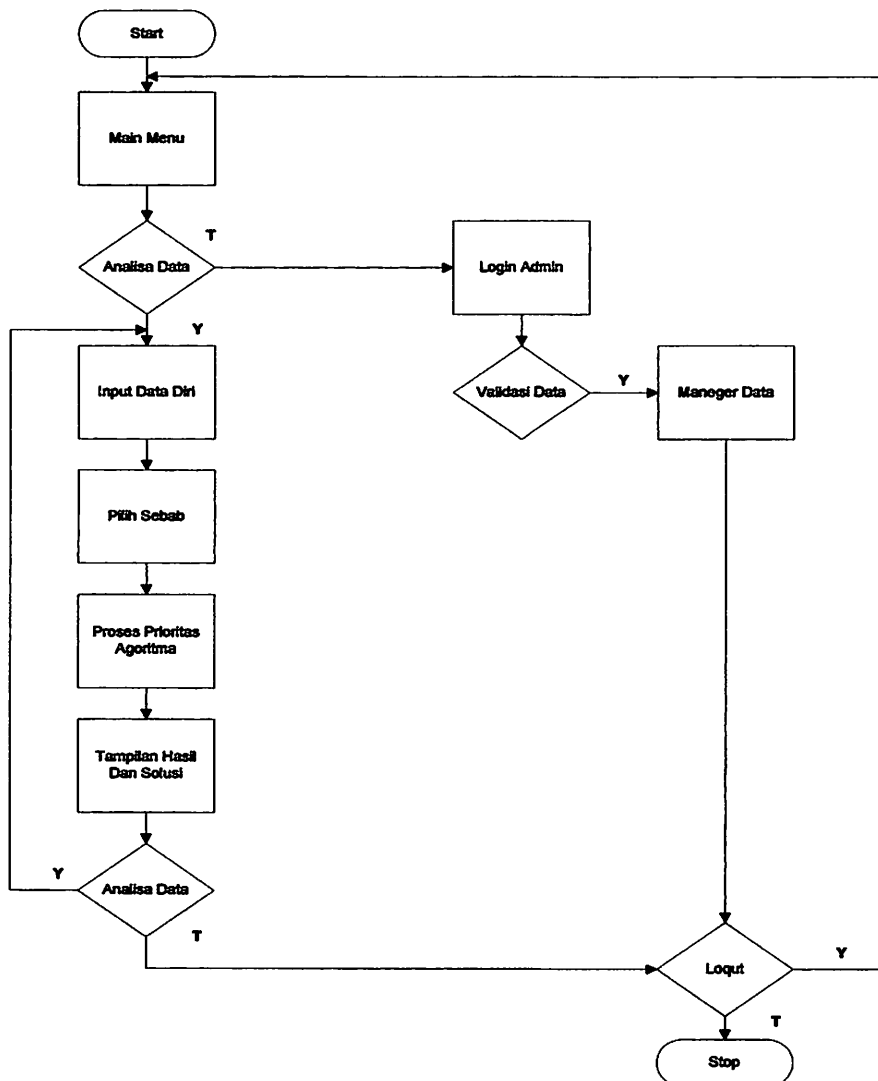
#### 3.4.4.4 DFD level 1 proses Analisa sebab kerusakan



Gambar 3.6 DFD level 1 proses analisa kerusakan.

Dalam proses analisa sebab kerusakan, user atau pengguna melakukan login, setelah login sukses user memberikan data kerusakan ke sistem. Sistem akan memproses data kerusakan dengan mencari sebab kerusakan. Kemudian data kerusakan dan sebab kerusakan akan ditampilkan pada hasil output kepada user.

### 3.5 Flowchart Sistem



Gambar 3.7 Flowchart Sistem

### 3.6 Perancangan Antar Muka.

Antar muka (interface) merupakan bagian dari sistem pakar yang digunakan sebagai alat komunikasi antara sistem dan user . Perancangan antar muka dalam sistem pakar untuk analisa kerusakan mesin mobil diesel dibedakan atas dua bagian yaitu:

1. Perancangan antar muka untuk pakar (admin).

Antar muka untuk admin dirancang agar admin dapat melakukan proses pengelolaan sistem.

## 2. Perancangan antar muka untuk pengguna (user).

Antar muka untuk user dirancang agar user dapat mencari informasi kerusakan mesin mobil diesel dan mencari penyebabnya.

### 3.6.1 Rancangan Menu Utama

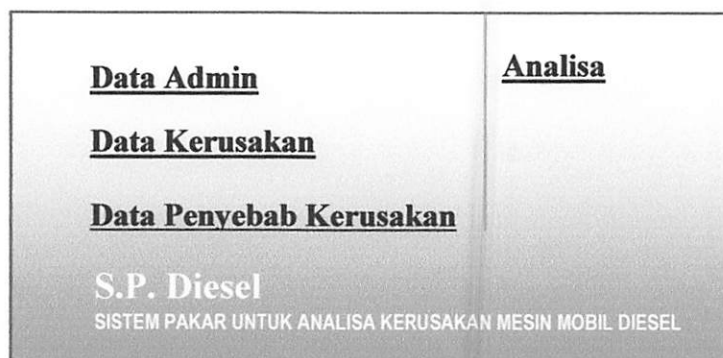


Gambar 3.8 Rancangan Menu Utama

Login Admin di klik maka akan masuk ke halaman bagi admin untuk mengisi Id Admin dan password.

Analisa merupakan halaman bagi pengguna untuk memasukan data diri dan memilih kerusakan mesin mobil yang terjadi sehingga akan diproses berupa data sebab dari kerusakan mesin mobil diesel.

### 3.6.2 Rancangan Menu Admin



Gambar 3.9 Rancangan Menu Admin

Halaman Admin hanya bisa diakses oleh admin. Data Admin merupakan halaman daftar admin dan data admin yang tersimpan. Data kerusakan merupakan halaman yang digunakan admin untuk menginput, mengedit atau menghapus data kerusakan. Data Penyebab kerusakan merupakan halaman yang digunakan admin untuk menginput, mengedit, atau menghapus data penyebab kerusakan. Analisa merupakan halaman bagi admin untuk menganalisa sebab dari kerusakan mesin mobil diesel.



## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Implementasi.

Penerapan aplikasi Sistem Pakar untuk analisa kerusakan mesin mobil Diesel ini dibangun menggunakan metode penalaran maju (*forward chaining*). Cara kerja metode ini adalah dimulai dari fakta kerusakan mesin yang ada, dilanjutkan pada sebab-sebab dari kerusakan. Dengan adanya program ini diharapkan mampu memberikan informasi yang dibutuhkan user atau pengguna dalam menemukan penyebab dari kerusakan pada mesin mobil Diesel.

### 4.2 Penjelasan Program.

Pada penjelasan program ini dijelaskan kegunaan program yang dibuat beserta tampilan desainnya. Berikut ini tampilan-tampilan form yang ada dalam program yang dibuat :

#### 4.2.1 Menu Halaman Depan

Dalam menu halaman depan ditampilkan halaman menu yang dapat diakses oleh pengguna dan admin. Berikut tampilannya:

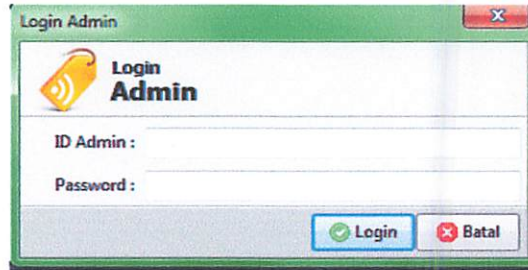


Gambar 4.1 Menu Halaman Depan.

#### 4.2.2 Halaman Menu Login Admin

Halaman login merupakan halaman bagi admin untuk mengisi ID pengguna dan password, jika ingin mengakses halaman admin.

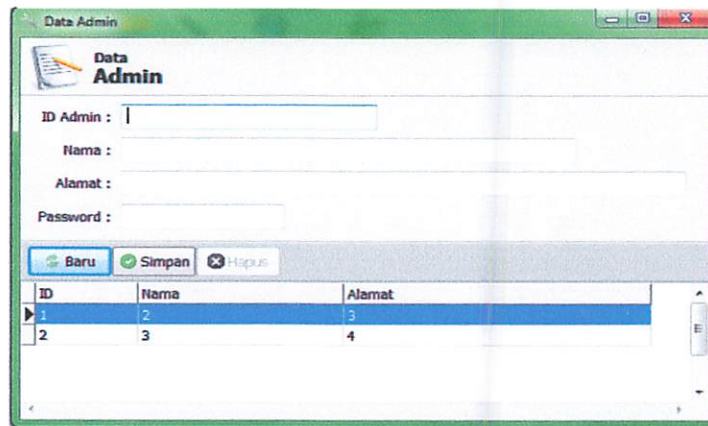
Adapun tampilannya adalah :



Gambar 4.2 Menu Login Admin

#### 4.2.3 Menu Halaman Data Admin.

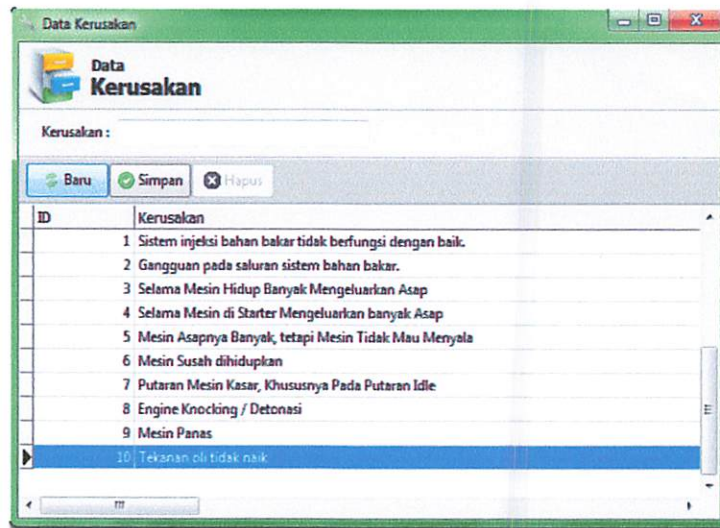
Pada halaman ini admin dapat memasukan data admin dan data admin akan ditampilkan. Adapun tampilannya sebagai berikut:



Gambar 4.3 Menu Halaman Data Admin.

#### 4.2.4 Menu Halaman Data Kerusakan.

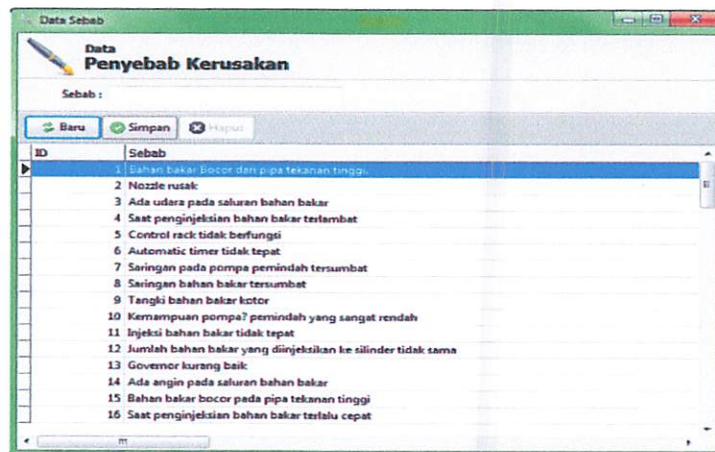
Halaman data kerusakan merupakan halaman bagi admin untuk memasukkan data-data kerusakan dari mesin mobil diesel dan tampilan data-data kerusakan. Adapun tampilannya sebagai berikut:



Gambar 4.4 Menu Halaman Data Kerusakan.

#### 4.2.5 Menu Halaman Data Penyebab Kerusakan.

Halaman data penyebab kerusakan merupakan halaman bagi admin untuk memasukkan data-data sebab dari kerusakan mesin mobil diesel dan tampilan data-data sebab kerusakan. Adapun tampilannya sebagai berikut:



Gambar 4.5 Menu Halaman Data Penyebab Kerusakan.

#### 4.2.6 Menu Halaman Proses Analisa

Halaman proses Analisa merupakan halaman inti dari sistem pakar analisa kerusakan mesin mobil diesel dimana dalam halaman ini pengguna

melakukan proses analisa menginputkan data diri terlebih dahulu kemudian memilih kerusakan. Adapun tampilan halamannya sebagai berikut:

Gambar 4.6 Menu Halaman Proses Analisa kerusakan.

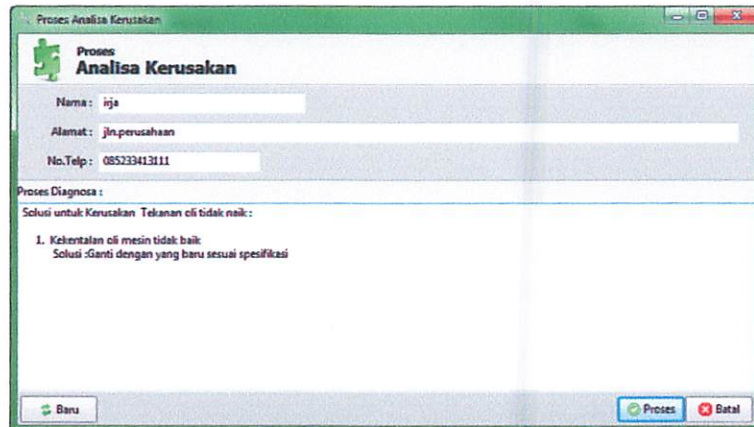
#### 4.2.7 Menu Halaman Proses pilihan penyebab Kerusakan

Halaman Proses pilihan Kerusakan ini adalah kelanjutan dari Form pilih jenis, dimana pada form ini program akan menanyakan kepada pemakai tentang gejala yang terjadi pada mobil untuk di deteksi hingga nanti program akan memberikan hasil deteksinya

Gambar 4.7 Menu Halaman Proses analisa pilihan penyebab Kerusakan

#### 4.2.8 Menu Halaman Hasil Deteksi Kerusakan

Pada form Hasil Deteksi ini terdapat hasil akhir dari deteksi kerusakan pada mobil, dimana pada form ini terdapat penyelesaian dari penelusuran yang dilakukan. Pada form ini anda akan diberikan cara menanggulangi masalah yang mungkin anda hadapi.



The screenshot shows a software window titled "Proses Analisa Kerusakan". The window has a green header bar with a logo on the left and the title "Proses Analisa Kerusakan" in the center. Below the header, there are three input fields: "Nama : ija", "Alamat : jln.perusahaan", and "No.Telp : 085233413111". Below these fields, there is a section titled "Proses Diagnosa :". Under this section, there is a heading "Solusi untuk Kerusakan Tekanan oli tidak naik:" followed by a list item "1. Kekentalan oli mesin tidak baik" with a sub-solution "Solusi :Ganti dengan yang baru sesuai spesifikasi". At the bottom of the window, there are three buttons: "Baru" (with a refresh icon), "Proses" (with a green play icon), and "Batal" (with a red stop icon).

Gambar 4.8 Menu Halaman Hasil Deteksi Kerusakan dan solusi perbaikan



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Setelah dilakukan pengujian dan analisa program, maka dapat diperoleh simpulan sebagai berikut :

1. Penggunaan metode Forward Chining merupakan langkah yang tepat dalam pembuatan Aplikasi sistem pakar untuk Analisa Kerusakan Mesin Mobil Diesel. Karena motode forward chaining adalah strategi untuk memprediksi atau mencari solusi dari suatu masalah yang dimulai dengan sekumpulan fakta yang diketahui, kemudian menerapkan fakta baru berdasarkan aturan yang permissnya cocok dengan fakta yang diketahui.
2. Dengan menggunakan program sistem pakar untuk analisa kerusakan pada Mesin Mobil Diesel yang akan di buat ini sekiranya dapat membantu menyelesaikan kerusakan pada Mesin Mobil Diesel
3. Dengan menggunakan program sistem pakar untuk analisa kerusakan pada Mesin Mobil Diesel yang akan dibuat bisa menyelesaikan pekerjaan seorang mekanik dalam waktu yang lebih singkat.
4. Dari hasil analisa penyebab kerusakan Mesin Mobil Diesel, didalam aplikasi ini adalah (kerusakan mesin) bukanlah kerusakan yang bisa didiagnosa secara pasti, mengingat berbagai pertimbangan yang dipakai untuk melakukan proses diagnose.

## **5.2 Saran**

Adapun berbagai saran untuk melengkapi kesimpulan yang diambil adalah sebagai berikut :

1. Pengetahuan tentang kerusakan Mobil Diesel semakin diperkaya dengan penambahan data-data baik itu kerusakan mesin maupun kelistrikan sehingga informasi yang dimiliki akan semakin luas dan banyak
2. Tampilan program perlu dibuat lebih menarik lagi dengan menambahkan contoh gambar kerusakan dari setiap komponen yang ada.



## DAFTAR PUSTAKA

- Arhami, Muhammad, *Konsep Dasar Sistem Pakar*, Yogyakarta : Andi Offset, 2005.
- Kusrini, *Sistem Pakar Teori dan Aplikasi*, Yogyakarta : Andi Offset, 2006.
- MADCOM, *Aplikasi Manajemen Database Pendidikan Berbasis Web dengan PHP dan MySQL*, Yogyakarta : Andi. 2007.
- Jogiyanto. 1990. 1999. 2005. *Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur, Teori dan Praktek Aplikasi Bisnis*. Penerbit Andi. Yogyakarta.
- Madcoms. 2008. *PHP dan MSOL untuk Pemula*. Penerbit Andi. Yogyakarta
- Syarif, Iwan dan Badriyah, Tessa, *Pembuatan Alat Bantu Ajar Sistem Pakar dengan Teknik inferensi Backward Chaining*, Surabaya : Kawan Pustaka, 2002.
- Permana, Danu, *Merawat dan Memperbaiki Mobil Diesel*, Puspa Swara. Jakarta. 1997.
- Boentarto, *Teknik servis mobil*, Effhar Offset. Semarang. 2005.
- ..., Informasi Mengenai Sistem Pakar  
[www.iel.ipb.ac.id/agrinetmedia/modul/aplikasi/spdt/isi/infoSP.htm](http://www.iel.ipb.ac.id/agrinetmedia/modul/aplikasi/spdt/isi/infoSP.htm).
- ..., Sistem Pakar.  
[www.iel.ipb.ac.id/agrinetmedia/modul/aplikasi/servercabai/pakar.html](http://www.iel.ipb.ac.id/agrinetmedia/modul/aplikasi/servercabai/pakar.html).
- ..., Informasi Mesin Diesel.  
<http://www.agussuwasono.com/artikel/mechanical/426-teori-dasar-mesin-diesel.html>
- ..., Free Pascal  
[http://www.schoolfreeware.com/Free\\_P...Tutorials.html](http://www.schoolfreeware.com/Free_P...Tutorials.html)



# LAMPIRAN



PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
**INSTITUT TEKNOLOGI NASIONAL MALANG**

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**


NAMA : Irja Atjo Sutrisno  
NIM : 05.12.637  
PROGRAM STUDI : Teknik Elektro S-1  
KONSENTRASI : Teknik Komputer & Informatika  
MASA BIMBINGAN : Semester Genap Tahun Akademik 2011 - 2012  
JUDUL : **SISTEM PAKAR UNTUK ANALISA  
KERUSAKAN MESIN MOBIL DIESEL**

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :


Hari : Rabu  
Tanggal : 8 Agustus 2012  
Dengan Nilai : 73 (B+) *or*

**PANITIA UJIAN SKRIPSI**

**Ketua Majelis Penguji,**

  
**Ir. Yusuf Ismail Nakhoda, MT**  
NIP.Y.1018800189

**Sekretaris Majelis Penguji,**

  
**Dr. Eng. Aryuanto S, ST, MT**  
NIP.Y.1030800417

**ANGGOTA PENGUJI**

**Dosen Penguji I**



**Bambang Prio Hartono, ST, MT**  
NIP.Y.1028400082

**Dosen Penguji II**



**Bima Aulia, ST**



INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
Jl. Karanglo km 2, Malang

## FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Irja Atjo Sutrisno  
NIM : 05.12.637  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer dan Informatika  
Masa Bimbingan : Semester Genap Tahun Akademik 2011 - 2012  
Judul Skripsi : **SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL**

Penguji / Tanggal	Uraian	Paraf
Penguji I 8 Agustus 2012	1. Abstrak	
Penguji II 8 Agustus 2012	1. Pengujian user di tambah minimal 10 2. Buku referensi tentang kerusakan Mobil Diesel 3. Penulisan 1,5 spasi 4. Perbaiki aplikasi ini 5. Daftar pustaka sesuai format skripsi	

Disetujui :

Dosen Penguji I

**Bambang Prio Hartono, ST, MT.**

NIP.Y.1028400082

Dosen Penguji II

**Bima Aulia, ST.**

Mengetahui :

Dosen Pembimbing I

**Dr. Eng. Aryuanto S, ST, MT.**

NIP.Y.1030800417

Dosen Pembimbing II

**Ir. Taufik Hidayat, MT.**

NIP. Y.1018700151





INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Sirata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA  
NIM  
Perbaikan melalui

: Liza Atjo S.  
: 0512637

Akmal

Malang,

( \_\_\_\_\_ )



## PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : IRJA ATJO SUKUNO.....  
 NIM : 08.12.637.....  
 Semester : .....  
 Fakultas : Teknologi Industri  
 Jurusan : Teknik Elektro S-1  
 Konsentrasi : **TEKNIK ELEKTRONIKA**  
**TEKNIK ENERGI LISTRIK**  
**TEKNIK KOMPUTER DAN INFORMATIKA**  
**TEKNIK KOMPUTER**  
**TEKNIK TELEKOMUNIKASI**  
 Alamat : .....

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat **SKRIPSI Tingkat Sarjana**. Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan-persyaratan pengambilan **SKRIPSI** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (.....)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (.....)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (.....)
4. Telah menempuh mata kuliah  $\geq 134$  sks dengan IPK  $\geq 2$  dan tidak ada nilai E (.....)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas  
 Recording T

Malang

201

A.  
 C. \*



JURUSAN TEKNIK ELEKTRO S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG

## DAFTAR PRESTASI AKADEMIK PRAKTIKUM KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

Nama Mahasiswa	:	IRJA ATJO SUTRISNO
NIM	:	05.12.637
Tempat, Tanggal Lahir	:	Merak, 10-10-1983
Jenjang	:	Strata 1 (S1)
Fakultas	:	Teknologi Industri
Jurusan / Program Studi	:	Teknik Elektro
Konsentrasi	:	Teknik Komputer dan Informatika

Praktikum Laboratorium	Kode	Nama Praktikum	SKS	Nilai
I	EL-2215 25	Fisika	1	B+
		Rangkaian Listrik		A
		Rangkaian Logika dan Digital		B+
		Dasar Komputer dan Pemrograman		B
II	EL-4216	Dasar Elektronika	1	B
		Dasar Sistem Telekomunikasi		B
		Mikrokontroler		B
III				



## LEMBAR PENGAJUAN JUDUL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik / Teknik Elektronika / Teknik Komputer &  
Informatika / Teknik Komputer / Teknik Telekomunikasi\*)

1.	Nama Mahasiswa: IRJA ATJO SUTRISNO		Nim:
2.	Waktu Pengajuan	Tanggal:	Bulan: Tahun:
3.	Spesifikasi Judul (berilah tanda silang)**)		
	a. Sistem Tenaga Elektrik	c. Elektronika & Komponen	
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi	
	d. Sistem Kendali Industri	h. lainnya .....	
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*)  <i>Dr. Aryananto, ST, MT</i>		Ketua Jurusan  <i>[Signature]</i> <u>Ir. Yusuf Ismail Nakhoda, MT</u> NIP. Y. 1018800189
5.	Judul yang diajukan mahasiswa:	SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL	
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu	.....	
7.	Catatan: .....		Disetujui 30/11/2011 Dosen <i>[Signature]</i>
	Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu		

**Perhatian:**

1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: \*) Coret yang tidak perlu  
\*\*) dilingkari a, b, c, .....atau g sesuai bidang keahlian



INSTITUT TEKNOLOGI NASIONAL  
Jl.Raya Karanglo,Km 2  
MALANG

Lampiran : 1 (satu) berkas  
**Pembimbing Skripsi**

Kepada : Yth. Bapak **Dr. Aryuanto, ST. MT**  
Dosen Institut Teknologi Nasional  
MALANG

Yang bertanda tangan di bawah ini :

Nama : **Irja Atjo Sutrisno**  
Nim : **05.12.637**  
Jurusan : **Teknik Elektro S-1**  
Konsentrasi : **Teknik ~~Elektronika~~/Komputer & Informatika**

Dengan ini mengajukan permohonan, Kiranya Bapak bersedia menjadi dosen pembimbing Utama / Pendamping \*), untuk penyusunan Skripsi dengan judul (Proposal terlampir) :

**SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL  
DIESEL**

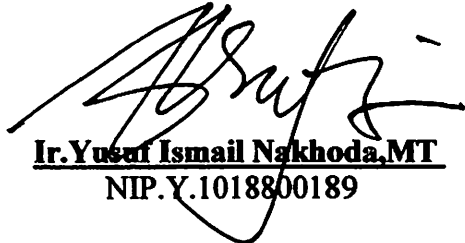
Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak kami ucapkan terima kasih.

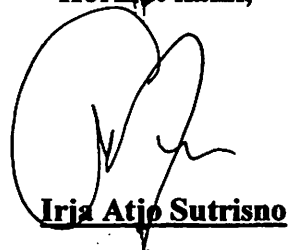
Malang, 3 Desember 2011

**Mengetahui**

**Ketua Jurusan Teknik Elektro S-1**

  
**Ir. Yusuf Ismail Nakhoda, MT**  
NIP. Y.1018800189

**Hormat kami,**

  
**Irja Atjo Sutrisno**

Form S-3a

\*)coret yang tidak perlu



INSTITUT TEKNOLOGI NASIONAL  
JL.Raya Karanglo,Km 2  
MALANG

## PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai dengan Permohonan Mahasiswa :

Nama : **Irja Atjo Sutrisno**  
Nim : **05.12.637**  
Semester : **XIV (Empat Belas)**  
Jurusan : **Teknik Elektro S-1**  
Konsentrasi : **Teknik Komputer & Informatika**

Dengan ini menyatakan **bersedia / tidak bersedia** \*) menjadi Dosen Pembimbing Utama / Pendamping \*\*), untuk penyusunan Skripsi Mahasiswa tersebut dengan judul :

### ***SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL,***

Demikian surat pernyataan ini kami buat agar dapat digunakan seperlunya.

Malang, 3 Desember 2011

Hormat kami,

**Dr. Aryanto, ST. MT**  
NIP.P.1030800417

Catatan :

1. Setelah disetujui agar formulir ini diserahkan mahasiswa/I yang bersangkutan kepada jurusan untuk diproses lebih lanjut.
2. \*) Coret yang tidak perlu



INSTITUT TEKNOLOGI NASIONAL  
Jl.Raya Karanglo,Km 2  
MALANG

Lampiran : 1 (satu) berkas  
**Pembimbing Skripsi**

Kepada : Yth. Bapak Ir. **Taufik Hidayat, MT.**  
Dosen Institut Teknologi Nasional  
MALANG

Yang bertanda tangan di bawah ini :

Nama : **Irja Atjo Sutrisno**  
Nim : **05.12.637**  
Jurusan : **Teknik Elektro S-1**  
Konsentrasi : **~~Teknik Elektronika~~/ Komputer & Informatika**

Dengan ini mengajukan permohonan, Kiranya Bapak bersedia menjadi dosen pembimbing Utama / Pendamping \*), untuk penyusunan Skripsi dengan judul (Proposal terlampir) :

**SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL  
DIESEL**


Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami dan atas kesediaan Bapak kami ucapkan terimakasih.

Malang, 3 Desember 2011

**Mengetahui**

**Ketua Jurusan Teknik Elektro S-1**

  
**Ir. Yusuf Ismail Nakhoda, MT**  
NIR.Y.1018800189

**Hormat kami,**

  
**Irja Atjo Sutrisno**

Form S-3a

\*)coret yang tidak perlu



INSTITUT TEKNOLOGI NASIONAL  
JL.Raya Karanglo,Km 2  
MALANG

## PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai dengan Permohonan Mahasiswa :

Nama : Irja Atjo Sutrisno  
Nim : 05.12.637  
Semester : XIV (Empat Belas)  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer & Informatika

Dengan ini menyatakan **bersedia / tidak bersedia** \*) menjadi Dosen Pembimbing Utama / Pendamping \*) , untuk penyusunan Skripsi Mahasiswa tersebut dengan judul :

### ***SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL***

Demikian surat pernyataan ini kami buat agar dapat digunakan seperlunya.

Malang, 3 Desember 2011

**Hormat kami,**

**Ir. Taufik Hidayat, MT**  
NIP.Y.1018700151

**Catatan** :

1. Setelah disetujui agar formulir ini diserahkan mahasiswa/I yang bersangkutan kepada jurusan untuk diproses lebih lanjut.
2. \*) Coret yang tidak perlu



## BERITA ACARA SEMINAR PROPOSAL SKRIPSI JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/ Teknik Komputer & Informatika\*)

1.	Nama Mahasiswa: <u>IPJA ATJO SUPRIANO</u>			Nim: <u>05-12-632</u>
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	<u>17-12-2011</u>	<u>09.00</u>	Ruang:
Spesifikasi Judul (berilah tanda silang)**)				
3.	a. Sistem Tenaga Elektrik	e. Elektronika & Komponen		
	b. Energi & Konversi Energi	f. Elektronika Digital & Komputer		
	c. Tegangan Tinggi & Pengukuran	g. Elektronika Komunikasi		
	d. Sistem Kendali Industri	h. lainnya <u>komputer dan informatika</u>		
4.	Judul Proposal yang diseminarkan Mahasiswa	<u>SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN Mobil DIESEL</u>		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	..... ..... .....		
6.	Catatan: <u>Metode ??</u>	..... ..... .....		
	Catatan:	..... ..... ..... .....		



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

P.T. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 30 Januari 2012

Nomor : ITN- 009/EL-FTI/2012  
Lampiran : -  
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **DR. ENG. ARYUANTO S, ST, MT**  
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing  
Jurusan Teknik Elektro S-1  
di  
Malang

Dengan hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi  
Untuk Mahasiswa :

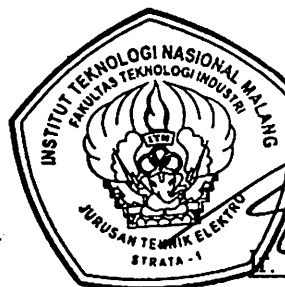
Nama : IRJA ATJO. S  
Nim : 0512637  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer & Informatika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya  
kepada Saudara/i selama masa waktu (enam ) 6 bulan, terhitung mulai  
tanggal :

17 Desember 2012 s/d 17 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,  
Jurusan Teknik Elektro S-1,

Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan  
Teknik Elektro S-1

*[Signature]*  
Yusuf Ismail/Nakhoda, MT  
Nip. Y 1018800189

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG  
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

T. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 30 Januari 2012

Nomor : ITN- 010/EL-FTI/2012  
Lampiran : -  
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr/I. **IR. H. TAUFIK HIDAYAT, MT**  
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing  
Jurusan Teknik Elektro S-1  
di  
Malang

Dengan hormat  
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi  
Untuk Mahasiswa :

Nama : IRJA ATJO. S  
Nim : 0512637  
Fakultas : Teknologi Industri  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik **Komputer & Informatika**

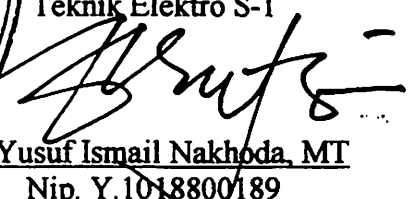
Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya  
kepada Saudara/i selama masa waktu (enam ) 6 bulan, terhitung mulai  
tanggal :

17 Desember 2012 s/d 17 Juni 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,  
Jurusan Teknik Elektro S-1,  
Demikian atas perhatian serta bantuannya kami sampaikan terima kasih



Ketua Jurusan  
Teknik Elektro S-1

  
Ir. Yusuf Ismail Nakhoda, MT  
Nip. Y.1018800189

Tembusan Kepada Yth :

1. Mahasiswa Yang Berangkutan
2. Arsip

Form. S 4a



**PROGRAM STUDI TEKNIK ELEKTRO S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

**FORMULIR BIMBINGAN SKRIPSI**

nim : 05.12.637  
nama : Irja Atjo Sutrisno  
Masa Bimbingan : Semester Genap 2011-2012  
Judul : SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL

Tanggal	Uraian	Paraf Pembimbing
16 Mei 2012	Revisi Bab I dan Bab II	
17 Mei 2012	Acc Bab I	
28 Mei 2012	Revisi Bab II dan Bab III	
11 Juni 2012	Acc Bab II dan Bab III	
19 Juni 2012	Revisi Bab IV dan Bab V	
23 Juni 2012	Acc Bab IV dan Bab V	

Malang,  
Dosen Pembimbing I

**Dr. Eng. Ardyanto S, ST, MT**  
NIP. Y.1030800417



**PROGRAM STUDI TEKNIK ELEKTRO S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

**FORMULIR BIMBINGAN SKRIPSI**

nim : 05.12.637  
nama : Irja Atjo Sutrisno  
Masa Bimbingan : Semester Genap 2011-2012  
Judul : SISTEM PAKAR UNTUK ANALISA KERUSAKAN MESIN MOBIL DIESEL

Tanggal	Uraian	Paraf Pembimbing
22 Mei 2012	Bab I, II Cantumkan Sumber Data (catatan kaki) Pedoman Dari Buku	
18 Juni 2012	Demo Program	
20 Juni 2012	Revisi Makalah Seminar Hasil	
25 Juni 2012	Revisi Uji Keakuratan Kelayakan	
29 Juni 2012	Acc Makalah Seminar Hasil	
30 Juli 2012	Revisi Program	

Malang,

Dosen Pembimbing II

**Ir. Taufik Hidayat, MT**  
NIP. Y.1018700151

```
unit UnitAdmin;
```

```
{ $mode objfpc } { $H+ }
```

```
interface
```

```
uses
```

```
Classes, SysUtils, mysql50conn, sqldb, odbconn, db,  
FileUtil, Forms,
```

```
Controls, Graphics, Dialogs, ExtCtrls, StdCtrls, Buttons,  
DBGrids, DbCtrls;
```

```
type
```

```
{ TFormAdmin }
```

```
TFormAdmin = class(TForm)
```

```
    btnBaru: TBitBtn;
```

```
    btnSimpan: TBitBtn;
```

```
    btnHapus: TBitBtn;
```

```
    DataSource1: TDataSource;
```

```
    DBEdit1: TDBEdit;
```

```
    DBGrid1: TDBGrid;
```

```
    Image1: TImage;
```

```
    Label1: TLabel;
```

```
    Label2: TLabel;
```

```
    A: TLabeledEdit;
```

```
B: TLabelEdit;

C: TLabelEdit;

D: TLabelEdit;

dbCon: TODBConnection;

Panel1: TPanel;

pnlMain: TPanel;

Panel3: TPanel;

SQLQuery1: TSQLQuery;

SQLQuery1alamat: TStringField;

SQLQuery1id_admin: TStringField;

SQLQuery1nama: TStringField;

Conn: TSQLQuery;

SQLTransaction1: TSQLTransaction;

procedure btnBaruClick(Sender: TObject);

procedure btnHapusClick(Sender: TObject);

procedure btnSimpanClick(Sender: TObject);

procedure DataSource1DataChange(Sender: TObject; Field:
TField);

procedure DBGrid1CellClick(Column: TColumn);

procedure DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;

    Shift: TShiftState; X, Y: Integer);

procedure FormCreate(Sender: TObject);

procedure MySQL50Connection1AfterConnect(Sender: TObject);
```

```

    procedure pnlMainClick(Sender: TObject);

private

    procedure ExecSQL(var Ado:TSQLQuery;StrSQL:string);

    procedure OpenSQL(var Ado:TSQLQuery;StrSQL:string);

public

    { public declarations }

end;

var

    FormAdmin: TFormAdmin;

implementation

{$R *.lfm}

{ TFormAdmin }

procedure TFormAdmin.ExecSQL(var Ado:TSQLQuery;StrSQL:string);
begin
    Ado.Close;

    Ado.SQL.Clear;

    Ado.SQL.Append(StrSQL);

    Ado.ExecSQL;

end;

procedure TFormAdmin.OpenSQL(var Ado:TSQLQuery;StrSQL:string);
begin
    Ado.Close;

```

```
Ado.SQL.Clear;
```

```
Ado.SQL.add(StrSQL);
```

```
Ado.Prepared;
```

```
Ado.Open;
```

```
end;
```

```
procedure TFormAdmin.MySQL50Connection1AfterConnect (Sender:  
TObject);
```

```
begin
```

```
end;
```

```
procedure TFormAdmin.pnlMainClick(Sender: TObject);
```

```
begin
```

```
end;
```

```
procedure TFormAdmin.Datasource1DataChange(Sender: TObject;  
Field: TField);
```

```
begin
```

```
end;
```

```
procedure TFormAdmin.DBGrid1CellClick(Column: TColumn);
```

```
begin
```

```
end;
```

```
procedure TFormAdmin.DBGrid1MouseUp(Sender: TObject; Button:  
TMouseButton;
```

```
Shift: TShiftState; X, Y: Integer);
```

```
begin
```

```
btnBaruClick(self);
```

```
btnSimpan.Caption := 'Edit';
```

```
btnHapus.Enabled := True;
```

```
PnlMain.Enabled := false;
```

```
end;
```

```
procedure TFormAdmin.btnBaruClick(Sender: TObject);
```

```
begin
```

```
    A.Clear;
```

```
    B.Clear;
```

```
    c.Clear;
```

```
    d.Clear;
```

```
    A.Enabled:=true;
```

```
    btnSimpan.Caption:='Simpan';
```

```
    btnHapus.Enabled := false;
```

```
    pnlMain.Enabled := True;
```

```
end;
```

```
procedure TFormAdmin.btnHapusClick(Sender: TObject);
```

```
var
```

```
    StrSQL : string;
```

```
begin
```

```
    if (DBEdit1.Text='') then exit;
```

```
    if (MessageDlg('Yakin Untuk Menghapus Data  
'?', mtConfirmation, [mbYes, mbNo], 0) = mrYes) then
```

```
        begin
```

```
        StrSQL := 'DELETE FROM admin      WHERE  id_admin      =  
''+DBEdit1.Text +' " '  ;
```

```
        ExecSQL(Conn, StrSQL);
```

```
    end;
```

```
    btnBaruClick(sender);
```

```
    SQLQuery1.Refresh ();
```

```
    dbGrid1.Refresh;
```

```
end;
```

```
procedure TFormAdmin.btnSimpanClick(Sender: TObject);
```

```
var
```

```
    StrSql : string;
```

```
    strCode : string;
```

```
    i : integer;
```

```
begin
```

```
    if (btnSimpan.Caption <>'Edit') then
```

```
    begin
```

```
        if (trim(A.Text)='' ) or (trim(B.Text)='' ) or  
(trim(C.Text)='' )or (trim(D.Text)='' )
```

```
        then
```

```
        begin
```

```
            MessageDlg('Inputan Tidak Boleh  
Kosong', mtError, [mbOK], 0);
```

```
            exit;
```

```

end;

end;

if (btnSimpan.Caption ='Simpan') then

begin

    if (MessageDlg('Yakin Untuk Menyimpan Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then

begin

    OpenSQL( conn,'SELECT * FROM admin WHERE id_admin
="' +A.Text+'");

    if ( conn.RecordCount = 0) then

        ExecSQL( conn,'INSERT INTO admin (id_admin,
nama, alamat,password ) VALUES
("'" +trim(A.Text)+'"', "'" +trim(B.Text)+'"', "'" +trim(C.Text)+'"'
,'"'+trim(D.Text)+'"')

    else

        MessageDlg('Data Sudah Ada Dalam
Database..',mtError,[mbOK],0);

        btnBaruClick(sender);

        SQLQuery1.Refresh();

        DBGrid1.Refresh;

end;

end else if (btnSimpan.Caption ='Edit') then

begin

    strSQL :='SELECT * FROM admin WHERE id_admin=
"' +DBEdit1.Text+'";

```

```

OpenSQL(conn, strSQL);

PnlMain.Enabled := true;

while (not( Conn.Eof)) do

begin

    A.Text :=trim(Conn.fieldByName('id_admin').AsString);

    B.Text :=trim(Conn.fieldByName('nama').AsString);

    C.Text :=trim(Conn.fieldByName('alamat').AsString);

    D.Text :=trim(Conn.fieldByName('password').AsString);

    Conn.Next;

end;

A.Enabled := false;

btnSimpan.Caption := 'Update';

Conn.Close;

end else if (btnSimpan.Caption = 'Update') then

begin

    if (MessageDlg('Yakin Untuk Mengupdate Data
?', mtConfirmation, [mbYes, mbNo], 0) = mrYes) then

        begin

            StrSQL := 'UPDATE admin SET nama
="' + trim(B.Text) + '", alamat = "' + trim(C.Text) + '" , password
="' + trim(D.Text) + '"          WHERE id_admin= "' + DBEdit1.Text + '"
';

            ExecSQL( Conn, StrSQL);

```

```
end;
```

```
  btnBaruClick(sender);
```

```
  SQLQuery1.Refresh();
```

```
  DBGrid1.Refresh;
```

```
end;
```

```
end;
```

```
procedure TFormAdmin.FormCreate(Sender: TObject);
```

```
begin
```

```
  dbCon.Connected:= true;
```

```
  SQLQuery1.Active:=True;
```

```
  btnBaruClick(Sender);
```

```
end;
```

```
end.
```

```
unit UnitAnalisa;
```

```
{ $mode objfpc } { $H+ }
```

```
interface
```

```
uses
```

```
  Classes, SysUtils, db, sqldb, odbconn, FileUtil, Forms,  
  Controls, Graphics,
```

```
Dialogs, ExtCtrls, StdCtrls, DbCtrls, Buttons, DBGrids,  
CheckLst;
```

```
type
```

```
{ TFormAnalisa }
```

```
TFormAnalisa = class(TForm)
```

```
  A: TLabelledEdit;
```

```
  B: TLabelledEdit;
```

```
  btnBaru: TBitBtn;
```

```
  btnHapus: TBitBtn;
```

```
  btnSimpan: TBitBtn;
```

```
  C: TLabelledEdit;
```

```
  CheckListBox1: TCheckListBox;
```

```
  Conn: TSQLQuery;
```

```
  Datasource1: TDataSource;
```

```
  dbCon: TODBConnection;
```

```
  DBEdit1: TDBEdit;
```

```
  Image1: TImage;
```

```
  Label1: TLabel;
```

```
  Label2: TLabel;
```

```
  Label3: TLabel;
```

```
  Memo1: TMemo;
```

```
Panel1: TPanel;

Panel2: TPanel;

Panel3: TPanel;

pnlMain: TPanel;

RadioGroup1: TRadioGroup;

ScrollBar1: TScrollBar;

SQLQuery1: TSQLQuery;

SQLQuery1alamat: TStringField;

SQLQuery1alamat1: TStringField;

SQLQuery1lid_admin: TStringField;

SQLQuery1lid_admin1: TStringField;

SQLQuery1nama: TStringField;

SQLQuery1nama1: TStringField;

SQLTransaction1: TSQLTransaction;

procedure btnBaruClick(Sender: TObject);

procedure btnHapusClick(Sender: TObject);

procedure btnSimpanClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure pnlMainClick(Sender: TObject);

private

    { private declarations }

public

    procedure ExecSQL(var Ado:TSQLQuery;StrSQL:string);
```

```
procedure OpenSQL(var Ado:TSQLQuery;StrSQL:string);
```

```
end;
```

```
var
```

```
FormAnalisa: TFormAnalisa;
```

```
S : array of TRadioGroup;
```

```
L : array of TLabel;
```

```
arrayGejala : TStringList;
```

```
strPeny : TStringList;
```

```
strCountPeny : TStringList;
```

```
strGejala : TStringList;
```

```
arraySolution : TStringList;
```

```
mode :integer;
```

```
implementation
```

```
{ $R *.lfm }
```

```
{ TFormAnalisa }
```

```
procedure TFormAnalisa.ExecSQL(var  
Ado:TSQLQuery;StrSQL:string);
```

```
begin
```

```
Ado.Close;
```

```
Ado.SQL.Clear;
```

```
Ado.SQL.Append(StrSQL);
```

```
Ado.ExecSQL;
```

```
end;
```

```
procedure TFormAnalisa.OpenSQL(var  
Ado:TSQLQuery;StrSQL:string);
```

```
begin
```

```
Ado.Close;
```

```
Ado.SQL.Clear;
```

```
Ado.SQL.add(StrSQL);
```

```
Ado.Prepared;
```

```
Ado.Open;
```

```
end;
```

```
procedure TFormAnalisa.btnHapusClick(Sender: TObject);
```

```
begin
```

```
close;
```

```
end;
```

```
procedure TFormAnalisa.btnBaruClick(Sender: TObject);
```

```
var strSQL : string; j:integer;
```

```
begin
```

```
strSQL := 'SELECT * FROM kerusakan ' ;
```

```
OpenSQL(conn,strSQL);
```

```
CheckListBox1.Items.Clear;
```

```
RadioGroup1.Items.Clear;
```

```
while (not( Conn.Eof)) do
begin
CheckListBox1.Items.Add(trim(Conn.fieldByName('id_k').AsString
)+' |'+trim(Conn.fieldByName('nama_kerusakan').AsString));

RadioGroup1.Items.Add(trim(Conn.fieldByName('id_k').AsString)+
' |'+trim(Conn.fieldByName('nama_kerusakan').AsString));

    Conn.Next;

end;

RadioGroup1.ItemIndex:= 0;

Mem1.Lines.Clear;

Mem1.Hide;

RadioGroup1.Show;

ScrollBar1.Hide;

for j:=0 to length(S)-1 do
begin

    FreeAndNil( S[j]);

    FreeAndNil( L[j]);

end;

a.Clear;

b.Clear;

C.Clear;

mode := 0;
```

```
end;

procedure TFormAnalisa.btnSimpanClick(Sender: TObject);

var

    i,coun,maxs,ids,lasPos: integer;

    strSql: String;

    strCode : string;

begin

    Memol.Lines.Clear;

    if (a.Text='') or (b.Text='') or (c.Text='') then

        begin

            Application.MessageBox('Inputan Tidak Boleh Kosong..', 'Info', 0);

            exit;

        end;

    if (mode=0) then

        begin

            RadioGroup1.Hide;

            ScrollBox1.Show;

            strPeny := TStringList.Create;

            strGejala := TStringList.Create;

            strCountPeny := TStringList.Create;
```

```

strCode :=
trim(copy(RadioGroup1.Items[RadioGroup1.ItemIndex],0,pos(
|',RadioGroup1.Items[RadioGroup1.ItemIndex])));

strSQL :='SELECT ' +
        'sebab_kerusakan.id, ' +
        ' sebab_kerusakan.id_s as id_s, ' +
        ' sebab_kerusakan.id_k, ' +
        ' sebab_kerusakan.solusi, ' +
        ' sebab.id_s, ' +
        ' sebab.nama_sebab ' +
        ' FROM ' +
        ' sebab_kerusakan ' +
        ' INNER JOIN sebab ON sebab.id_s =
sebab_kerusakan.id_s ' +
        ' WHERE
sebab_kerusakan.id_k="' +strCode+'"' ;

OpenSQL(conn,strSQL);

arrayGejala := TStringList.Create;
arraySolution := TStringList.Create;

while (not( Conn.Eof)) do

begin

arrayGejala.Add(Conn.fieldByName('id_s').AsString);

strGejala.Add(Conn.fieldByName('nama_sebab').AsString);

```

```

arraySolution.Add(Conn.fieldByName('solusi').AsString);

        Conn.Next;

    end;

    SetLength( S, (arrayGejala.Count));

    SetLength( L, (arrayGejala.Count));

    lasPos := 0;

    for i:=0 to arrayGejala.Count-1 do

    begin

                L[i] := TLabel.Create(self);

                with L[i] do

                begin

                        Parent := ScrollBox1;

                        Left := 18;

                        Top := lasPos+5;

                        Width := 780;

                        Height := 10;

                                AutoSize := true;

                                        Caption :=
format('%d. %s', [i+1, trim(strGejala[i])]);

                end;

                S[i] := TRadioGroup.Create(self);

                with S[i] do

```

```

begin
    Left := 32;
    Width := 720;
    Height :=70;
    Anchors := [akLeft, akTop,
akRight];

    BorderStyle:= bsNone
;

    Parent := ScrollBox1;
    Top :=(L[i].top + L[i].height) +5;
    lasPos := Top+Height ;
    Columns := 1 ;
    Items.Add('Ya.' );
    Items.Add('Tidak.' );
    ItemIndex:= 1;
    TabOrder := 0 ;

end;

end;

Panel2.Top:=lasPos +20;
ScrollBox1.Repaint;

mode:=1;

end else if mode=1 then

begin
    coun:= 0;

```

```
Memol.Show;
```

```
Memol.Lines.Add('Solusi untuk Kerusakan '
+trim(copy(RadioGroup1.Items[RadioGroup1.ItemIndex],pos('
|',RadioGroup1.Items[RadioGroup1.ItemIndex])+2,100)) +' : '
);
```

```
Memol.Lines.Add('');
```

```
for i:=0 to length(S)-1 do
```

```
begin
```

```
if (S[i].ItemIndex=0) then
```

```
begin
```

```
inc(coun);
```

```
Memol.Lines.Add(' '+ IntToStr(coun) + ('. '+
strGejala[i]));
```

```
Memol.Lines.Add(' Solusi :'+
arraySolution[i]);
```

```
end;
```

```
end;
```

```
ScrollBar1.hide;
```

```
end;
```

```
if coun=0 then
```

```
begin
```

```
Memol.Lines.Add(' Tidak Ada solusi untuk kerusakan
Anda..');
```

```
end;
```

```

ExecSQL( conn,'INSERT INTO user ( nama, alamat,telp )
VALUES
("'" +trim(A.Text)+'"', "'" +trim(B.Text)+'"', "'" +trim(C.Text)+'"' )');

exit;

for i:=0 to CheckListBox1.Items.Count-1 do
begin
    if (CheckListBox1.Checked[i]) then
    begin
        strGejala.Add(
trim(copy(CheckListBox1.Items[i],0,pos('
|',CheckListBox1.Items[i]))));

        Mem1.Lines.Add(' '+CheckListBox1.Items[i]);
    end;
end;

for i:=0 to strGejala.Count-1 do
begin
    strSQL :='SELECT * FROM sebab_kerusakan WHERE
id_k="'+strGejala[i]+'"' ;

    OpenSQL(conn,strSQL);

    CheckListBox1.Items.Clear;

    while (not( Conn.Eof)) do
    begin
        if
strPeny.IndexOf(Conn.fieldByName('id_s').AsString)=-1 then

            begin

```

```

strPeny.Add(Conn.fieldByName('id_s').AsString);

        strCountPeny.Add('1');

    end else

        begin

            coun:= StrToInt(strCountPeny[i]);

            inc(coun);

            strCountPeny[i] := IntToStr( coun);

        end;

    Conn.Next;

end;

end;

Memol.Show;

maxs:=0;

ids :=-1;

Memol.Lines.Add('');

for i:= 0 to strCountPeny.Count -1 do

begin

    coun:= StrToInt(strCountPeny[i]);

    if (coun>maxs) then

begin

        maxs:= StrToInt(strCountPeny[i]);

        ids:= i;


```

```
end;
```

```
end;
```

```
Mem1.Lines.Add('Asumsi Kerusakan :');
```

```
if ids<>-1 then
```

```
begin
```

```
strSQL := 'SELECT * FROM sebab WHERE id_s=
```

```
''+strPeny[ids]+' " ' ;
```

```
OpenSQL(conn, strSQL);
```

```
PnlMain.Enabled := true;
```

```
while (not( Conn.Eof)) do
```

```
begin
```

```
Mem1.Lines.Add(' '+
```

```
Conn.fieldByName('nama_sebab').AsString);
```

```
Conn.Next;
```

```
end;
```

```
end else
```

```
Mem1.Lines.Add(' Tidak Ada Kerusakan yang dikenali..');
```

```
end;
```

```
procedure TFormAnalisa.FormCreate(Sender: TObject);
```

```
begin
```

```
btnBaruClick(self);
```

```
end;
```

```
procedure TFormAnalisa.pnlMainClick(Sender: TObject);
```

```
begin
```

end;

end.

unit UnitMain;

{\$mode objfpc}{\$H+}

interface

uses

Classes, SysUtils, odbcconn, sqlldb, FileUtil, Forms,  
Controls, Graphics,

Dialogs, ExtCtrls, StdCtrls, ComCtrls, UNITAdmin  
,UnitKerusakan,

UnitSebab,UnitSebabKerusakan,UnitLogin,unitAnalisa

;

type

{ TFormMain }

TFormMain = class(TForm)

Image1: TImage;

ImageList1: TImageList;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

```
ToolBar1: TToolBar;  
  
ToolButton1: TToolButton;  
  
ToolButton10: TToolButton;  
  
ToolButton2: TToolButton;  
  
ToolButton4: TToolButton;  
  
ToolButton5: TToolButton;  
  
ToolButton6: TToolButton;  
  
procedure FormCreate(Sender: TObject);  
  
procedure Image1Click(Sender: TObject);  
  
procedure ToolButton10Click(Sender: TObject);  
  
procedure ToolButton1Click(Sender: TObject);  
  
procedure ToolButton2Click(Sender: TObject);  
  
procedure ToolButton3Click(Sender: TObject);  
  
procedure ToolButton4Click(Sender: TObject);  
  
procedure ToolButton5Click(Sender: TObject);  
  
procedure ToolButton6Click(Sender: TObject);
```

```
private
```

```
{ private declarations }
```

```
public
```

```
{ public declarations }
```

```
end;
```

```
var
```

```
FormMain: TFormMain;
```

```
implementation
```

```
{ $R *.lfm }
```

```
{ TFormMain }
```

```
procedure TFormMain.Image1Click(Sender: TObject);
```

```
begin
```

```
end;
```

```
procedure TFormMain.ToolButton10Click(Sender: TObject);
```

```
begin
```

```
    FormLogin.ShowModal;
```

```
end;
```

```
procedure TFormMain.FormCreate(Sender: TObject);
```

```
begin
```

```
end;
```

```
procedure TFormMain.ToolButton1Click(Sender: TObject);
```

```
begin
```

```
    formkerusakan.show;
```

```
end;
```

```
procedure TFormMain.ToolButton2Click(Sender: TObject);
```

```
begin
    FormSebabKerusakan.show;
end;

procedure TFormMain.ToolButton3Click(Sender: TObject);
begin
end;

procedure TFormMain.ToolButton4Click(Sender: TObject);
begin
    FormSebab.Show;
end;

procedure TFormMain.ToolButton5Click(Sender: TObject);
begin
    FormAdmin.Show;
end;

procedure TFormMain.ToolButton6Click(Sender: TObject);
begin
    FormAnalisa.Show;
end;

end.

unit UnitLogin;

{$mode objfpc}{$H+}
```

interface

uses

Classes, SysUtils, db, sqldb, odbconn, FileUtil, Forms,  
Controls, Graphics,

Dialogs, ExtCtrls, StdCtrls, DbCtrls, Buttons, DBGrids ;

type

{ TFormLogin }

TFormLogin = class(TForm)

A: TLabelledEdit;

btnHapus: TBitBtn;

btnSimpan: TBitBtn;

Conn: TSQLQuery;

B: TLabelledEdit;

Datasource1: TDataSource;

dbCon: TODBConnection;

DBEdit1: TDBEdit;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Panel1: TPanel;

```
Panel3: TPanel;

pnlMain: TPanel;

SQLQuery1: TSQLQuery;

SQLQuerylalamat: TStringField;

SQLQuerylalamat1: TStringField;

SQLQuerylid_admin: TStringField;

SQLQuerylid_admin1: TStringField;

SQLQuerylnama: TStringField;

SQLQuerylnama1: TStringField;

SQLTransaction1: TSQLTransaction;

procedure btnHapusClick(Sender: TObject);

procedure btnSimpanClick(Sender: TObject);

procedure FormCreate(Sender: TObject);
```

```
private
```

```
procedure ExecSQL(var Ado:TSQLQuery;StrSQL:string);

procedure OpenSQL(var Ado:TSQLQuery;StrSQL:string);
```

```
public
```

```
{ public declarations }
```

```
end;
```

```
var
```

```
FormLogin: TFormLogin;
```

implementation

uses

UnitMain;

{ \$R \*.lfm }

{ TFormLogin }

procedure TFormLogin.ExecSQL(var Ado:TSQLQuery;StrSQL:string);

begin

Ado.Close;

Ado.SQL.Clear;

Ado.SQL.Append(StrSQL);

Ado.ExecSQL;

end;

procedure TFormLogin.OpenSQL(var Ado:TSQLQuery;StrSQL:string);

begin

Ado.Close;

Ado.SQL.Clear;

Ado.SQL.add(StrSQL);

Ado.Prepared;

Ado.Open;

end;

```

procedure TFormLogin.btnSimpanClick(Sender: TObject);
var
    StrSql : string;
begin
    if (trim(A.Text)='' ) or (trim(B.Text)='' )
    then
        begin
            MessageDlg('Inputan Tidak Boleh
Kosong',mtError, [mbOK], 0);

            exit;

        end;

        strSQL :='SELECT * FROM admin WHERE id_admin=
'+A.Text+' and password= ''+B.Text+' ' ;

        OpenSQL(conn, strSQL);

        if (Conn.RecordCount>0) then
            begin
                FormMain.ToolButton1.Visible:=True;

                FormMain.ToolButton2.Visible:=True;

                FormMain.ToolButton4.Visible:=True;

                FormMain.ToolButton5.Visible:=True;

                FormMain.ToolButton6.Visible:=True;

                FormMain.ToolButton10.Visible:=false;

                close;
            end;
        end;
end;

```

```
end else
```

```
begin
```

```
    FormMain.ToolButton1.Visible:=false;
```

```
    FormMain.ToolButton2.Visible:=false;
```

```
    FormMain.ToolButton4.Visible:=false;
```

```
    FormMain.ToolButton5.Visible:=false;
```

```
    FormMain.ToolButton6.Visible:=true;
```

```
    FormMain.ToolButton10.Visible:=true;
```

```
    MessageDlg('Login Admin Gagal..',mtError,[mbOK],0);
```

```
end;
```

```
end;
```

```
procedure TFormLogin.btnHapusClick(Sender: TObject);
```

```
begin
```

```
    close();
```

```
end;
```

```
procedure TFormLogin.FormCreate(Sender: TObject);
```

```
begin
```

```
    dbCon.Connected:= true;
```

```
    SQLQuery1.Active:=True;
```

```
end;
```

```
end.
```

```
unit UnitSebabKerusakan;
```

```
{ $mode objfpc } { $H+ }
```

```
interface
```

```
uses
```

```
Classes, SysUtils, db, sqldb, odbconn, FileUtil, Forms,  
Controls, Graphics,
```

```
Dialogs, ExtCtrls, StdCtrls, DbCtrls, Buttons, DBGrids;
```

```
type
```

```
{ TFormSebabKerusakan }
```

```
TFormSebabKerusakan = class(TForm)
```

```
  A1: TLabelEdit;
```

```
  B: TDBLookupComboBox;
```

```
  btnBaru: TBitBtn;
```

```
  btnHapus: TBitBtn;
```

```
  btnSimpan: TBitBtn;
```

```
  Conn: TSQLQuery;
```

```
  Datasource1: TDataSource;
```

```
  Datasource2: TDataSource;
```

```
  Datasource3: TDataSource;
```

```
  dbCon: TODBConnection;
```

```
DBEdit1: TDBEdit;
DBGrid1: TDBGrid;
A: TDBLookupComboBox;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Image1: TImage;
Label1: TLabel;
Label2: TLabel;
Label4: TLabel;
Label5: TLabel;
Panel1: TPanel;
Panel3: TPanel;
pnlMain: TPanel;
SQLQuery1: TSQLQuery;
SQLQuerylid: TAutoIncField;
SQLQuerylid_k1: TAutoIncField;
SQLQuerylid_k2: TAutoIncField;
SQLQuerylnama_kerusakan: TStringField;
SQLQuerylnama_kerusakan1: TStringField;
SQLQuerylnama_kerusakan2: TStringField;
SQLQuerylnama_sebab: TStringField;
SQLQuerylsolusi: TMemoField;
```

```

SQLQuery2: TSQLQuery;

SQLQuery3: TSQLQuery;

SQLTransaction1: TSQLTransaction;

procedure AChange(Sender: TObject);

procedure BChange(Sender: TObject);

procedure btnBaruClick(Sender: TObject);

procedure btnHapusClick(Sender: TObject);

procedure btnSimpanClick(Sender: TObject);

procedure DataSource1DataChange(Sender: TObject; Field:
TField);

procedure DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;

    Shift: TShiftState; X, Y: Integer);

procedure FormCreate(Sender: TObject);

procedure Panel1Click(Sender: TObject);

procedure pnlMainClick(Sender: TObject);

private

    procedure ExecSQL(var Ado:TSQLQuery;StrSQL:string);

    procedure OpenSQL(var Ado:TSQLQuery;StrSQL:string);

public

    { public declarations }

end;

var

```

```
FormSebabKerusakan: TFormSebabKerusakan;
```

```
implementation
```

```
{ $R *.lfm }
```

```
{ TFormSebabKerusakan }
```

```
procedure TFormSebabKerusakan.ExecSQL (var  
Ado:TSQLQuery; StrSQL:string);
```

```
begin
```

```
    Ado.Close;
```

```
    Ado.SQL.Clear;
```

```
    Ado.SQL.Append(StrSQL);
```

```
    Ado.ExecSQL;
```

```
end;
```

```
procedure TFormSebabKerusakan.OpenSQL (var  
Ado:TSQLQuery; StrSQL:string);
```

```
begin
```

```
    Ado.Close;
```

```
    Ado.SQL.Clear;
```

```
    Ado.SQL.add(StrSQL);
```

```
    Ado.Prepared;
```

```
    Ado.Open;
```

```
end;

procedure TFormSebabKerusakan.AChange(Sender: TObject);

begin

    Edit1.Text := a.KeyValue;

end;

procedure TFormSebabKerusakan.BChange(Sender: TObject);

begin

    Edit2.Text:= B.KeyValue;

end;

procedure TFormSebabKerusakan.btnBaruClick(Sender: TObject);

begin

    A.Enabled := true;

    B.Enabled := true;

    A1.Clear;

    PnlMain.Enabled := true;

    btnSimpan.Caption:='Simpan';

    btnHapus.Enabled := false;

end;

procedure TFormSebabKerusakan.btnHapusClick(Sender: TObject);

var

    StrSQL : string;

begin
```

```

if (DBEdit1.Text='') then exit;

if (MessageDlg('Yakin Untuk Menghapus Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then

begin
    StrSQL :='DELETE FROM sebab_kerusakan    WHERE id
=    '"+DBEdit1.Text +' "    '    ;

    ExecSQL(Conn,StrSQL);

end;

btnBaruClick(sender);

SQLQuery1.Refresh ();

dbGrid1.Refresh;

end;

procedure TFormSebabKerusakan.btnSimpanClick(Sender: TObject);

var

    StrSql : string;

    strCode : string;

    i : integer;

begin

if (btnSimpan.Caption <>'Edit') then

begin

if (A.KeyValue=null ) or (B.KeyValue=null )

then

```

```

begin
    MessageDlg('Inputan Tidak Boleh
Kosong',mtError,[mbOK],0);

    exit;

end;

end;

if (btnSimpan.Caption ='Simpan') then
begin
    if (MessageDlg('Yakin Untuk Menyimpan Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then
begin
    OpenSQL( conn,'SELECT * FROM sebab_kerusakan WHERE
id_k =' +Edit1.text+' and id_s = '+Edit2.Text+' ');

    if ( conn.RecordCount = 0) then

        ExecSQL( conn,'INSERT INTO sebab_kerusakan
(id_k, id_s,solusi ) VALUES ( '+trim(Edit1.Text)+' ,
'+trim(Edit2.Text)+' , '+trim(a1.Text)+' )')

    else

        MessageDlg('Data Sudah Ada Dalam
Database..',mtError,[mbOK],0);

        SQLQuery1.Refresh();

        DBGrid1.Refresh;

        btnBaruClick(sender);

end;
end;

```

```

end else if (btnSimpan.Caption ='Edit') then
begin
    strSQL :='SELECT * FROM sebab_kerusakan WHERE id=
'+DBEdit1.Text+' " ' ;
    OpenSQL(conn,strSQL);
    PnlMain.Enabled := true;
    while (not( Conn.Eof)) do
begin
    A.KeyValue :=trim(Conn.fieldByName('id_k').AsString);
    B.KeyValue :=trim(Conn.fieldByName('id_s').AsString);
    Edit1.Text := trim(Conn.fieldByName('id_k').AsString);
    Edit2.Text := trim(Conn.fieldByName('id_s').AsString);
    A1.Text := trim(Conn.fieldByName('solusi').AsString);
    Conn.Next;
end;
A.Enabled := false;
btnSimpan.Caption := 'Update';
Conn.Close;
end else if (btnSimpan.Caption ='Update') then
begin
    if (MessageDlg('Yakin Untuk Mengupdate Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then
begin

```

```
StrSQL := 'UPDATE sebab_kerusakan SET id_k
="' + trim(Edit1.Text) + '", id_s = "' + trim(Edit2.Text) + '" ,solusi=
"' + trim(A1.Text) + '" WHERE id= "' + DBEdit1.Text + '" ';
```

```
ExecSQL( Conn, StrSQL);
```

```
end;
```

```
btnBaruClick(sender);
```

```
SQLQuery1.Refresh();
```

```
DBGrid1.Refresh;
```

```
end;
```

```
end;
```

```
procedure TFormSebabKerusakan.DataSource1DataChange(Sender:
TObject;
```

```
Field: TField);
```

```
begin
```

```
end;
```

```
procedure TFormSebabKerusakan.DBGrid1MouseUp(Sender: TObject;
```

```
Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
```

```
begin
```

```
btnBaruClick(self);
```

```
btnSimpan.Caption := 'Edit';
```

```
btnHapus.Enabled := True;
```

```
PnlMain.Enabled := false;
```

```
end;
```

```
procedure TFormSebabKerusakan.FormCreate(Sender: TObject);
```

```
begin
    dbCon.Connected:= true;
    SQLQuery1.Active:=True;
    SQLQuery2.Active:=True;
    SQLQuery3.Active:=True;
    btnBaruClick(Sender);
end;

procedure TFormSebabKerusakan.Panel1Click(Sender: TObject);
begin
end;

procedure TFormSebabKerusakan.pnlMainClick(Sender: TObject);
begin
end;

end.
```

```
unit UnitSebab;
```

```
{$mode objfpc}{$H+}
```

```
interface
```

```
uses
```

Classes, SysUtils, db, sqldb, odbconn, FileUtil, Forms,  
Controls, Graphics,  
Dialogs, ExtCtrls, StdCtrls, DbCtrls, Buttons, DBGrids;

type

{ TFormSebab }

TFormSebab = class(TForm)

A: TLabelledEdit;

btnBaru: TBitBtn;

btnHapus: TBitBtn;

btnSimpan: TBitBtn;

Conn: TSQLQuery;

Datasource1: TDataSource;

dbCon: TODBCConnection;

DBEdit1: TDBEdit;

DBGrid1: TDBGrid;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Panel1: TPanel;

Panel3: TPanel;

pnlMain: TPanel;

```
SQLQuery1: TSQLQuery;
```

```
SQLQuerylid_k1: TAutoIncField;
```

```
SQLQuerylid_s: TAutoIncField;
```

```
SQLQuerylnama_kerusakan1: TStringField;
```

```
SQLQuerylnama_sebab: TStringField;
```

```
SQLTransaction1: TSQLTransaction;
```

```
procedure btnBaruClick(Sender: TObject);
```

```
procedure btnHapusClick(Sender: TObject);
```

```
procedure btnSimpanClick(Sender: TObject);
```

```
procedure DBGrid1MouseUp(Sender: TObject; Button:
```

```
TMouseButton;
```

```
    Shift: TShiftState; X, Y: Integer);
```

```
procedure FormCreate(Sender: TObject);
```

```
private
```

```
    procedure ExecSQL(var Ado:TSQLQuery;StrSQL:string);
```

```
    procedure OpenSQL(var Ado:TSQLQuery;StrSQL:string);
```

```
public
```

```
    { public declarations }
```

```
end;
```

```
var
```

```
    FormSebab: TFormSebab;
```

implementation

{ \$R \*.lfm }

{ TFormSebab }

procedure TFormSebab.ExecSQL(var Ado:TSQLQuery;StrSQL:string);

begin

    Ado.Close;

    Ado.SQL.Clear;

    Ado.SQL.Append(StrSQL);

    Ado.ExecSQL;

end;

procedure TFormSebab.OpenSQL(var Ado:TSQLQuery;StrSQL:string);

begin

    Ado.Close;

    Ado.SQL.Clear;

    Ado.SQL.add(StrSQL);

    Ado.Prepared;

    Ado.Open;

end;

procedure TFormSebab.FormCreate(Sender: TObject);

begin

    dbCon.Connected:= true;

```

SQLQuery1.Active:=True;

btnBaruClick(Sender);

end;

procedure TFormSebab.btnBaruClick(Sender: TObject);
begin
    A.Clear;

    A.Enabled:= true;

    btnSimpan.Caption:='Simpan';

    btnHapus.Enabled := false;

    pnlMain.Enabled := True;

end;

procedure TFormSebab.btnHapusClick(Sender: TObject);
var
    StrSQL : string;
begin
    if (DBEdit1.Text='') then exit;

    if (MessageDlg('Yakin Untuk Menghapus Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then
        begin
            StrSQL :='DELETE FROM sebab WHERE id_S =
'+DBEdit1.Text +' ' ;

            ExecSQL(Conn,StrSQL);

        end;

    btnBaruClick(sender);

```

```
SQLQuery1.Refresh ();
```

```
dbGrid1.Refresh;
```

```
end;
```

```
procedure TFormSebab.btnSimpanClick(Sender: TObject);
```

```
var
```

```
  StrSql : string;
```

```
  strCode : string;
```

```
  i : integer;
```

```
begin
```

```
  if (btnSimpan.Caption <>'Edit') then
```

```
  begin
```

```
    if (trim(A.Text)='' )
```

```
    then
```

```
    begin
```

```
      MessageDlg('Inputan Tidak Boleh  
Kosong', mtError, [mbOK], 0);
```

```
      exit;
```

```
    end;
```

```
  end;
```

```
  if (btnSimpan.Caption ='Simpan') then
```

```
  begin
```

```

    if (MessageDlg('Yakin Untuk Menyimpan Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then

    begin

        OpenSQL( conn,'SELECT * FROM sebab WHERE
nama_sebab =''+A.Text+''');

        if ( conn.RecordCount = 0) then

            ExecSQL( conn,'INSERT INTO sebab (nama_sebab
) VALUES (''+trim(A.Text)+'" )')

        else

            MessageDlg('Data Sudah Ada Dalam
Database..',mtError,[mbOK],0);

            btnBaruClick(sender);

            SQLQuery1.Refresh();

            DBGrid1.Refresh;

        end;

    end else if (btnSimpan.Caption ='Edit') then

    begin

        strSQL :='SELECT * FROM sebab WHERE id_s=
''+DBEdit1.Text+'"' ;

        OpenSQL(conn,strSQL);

        PnlMain.Enabled := true;

        while (not( Conn.Eof)) do

        begin

            A.Text :=trim(Conn.fieldByName('nama_sebab').AsString);

```

```
Conn.Next;
```

```
end;
```

```
btnSimpan.Caption := 'Update';
```

```
Conn.Close;
```

```
end else if (btnSimpan.Caption = 'Update') then
```

```
begin
```

```
    if (MessageDlg('Yakin Untuk Mengupdate Data  
?', mtConfirmation, [mbYes, mbNo], 0) = mrYes) then
```

```
        begin
```

```
            strSQL := 'UPDATE sebab SET nama_sebab  
="' + trim(A.Text) + '" WHERE id_s= "' + DBEdit1.Text + '"';
```

```
            ExecSQL( Conn, strSQL);
```

```
        end;
```

```
        btnBaruClick(sender);
```

```
        SQLQuery1.Refresh();
```

```
        DBGrid1.Refresh;
```

```
    end;
```

```
end;
```

```
procedure TFormSebab.DBGrid1MouseUp(Sender: TObject; Button:  
TMouseButton;
```

```
    Shift: TShiftState; X, Y: Integer);
```

```
begin
```

```
btnBaruClick(self);  
  
btnSimpan.Caption := 'Edit';  
  
btnHapus.Enabled := True;  
  
PnlMain.Enabled := false;
```

```
end;
```

```
end.
```

```
unit unitkerusakan;
```

```
{$mode objfpc}{$H+}
```

```
interface
```

```
uses
```

```
Classes, SysUtils, db, sqldb, odbconn, FileUtil, Forms,  
Controls, Graphics,
```

```
Dialogs, ExtCtrls, StdCtrls, DbCtrls, Buttons, DBGrids;
```

```
type
```

```
{ TFormKerusakan }
```

```
TFormKerusakan = class(TForm)
```

```
    A: TLabelEdit;
```

```
    btnBaru: TBitBtn;
```

```
btnHapus: TBitBtn;

btnSimpan: TBitBtn;

Conn: TSQLQuery;

Datasourcel: TDataSource;

dbCon: TODBConnection;

DBEdit1: TDBEdit;

DBGrid1: TDBGrid;

Image1: TImage;

Label1: TLabel;

Label2: TLabel;

Panel1: TPanel;

Panel3: TPanel;

pnlMain: TPanel;

SQLQuery1: TSQLQuery;

SQLQuerylalamat1: TStringField;

SQLQuerylid_admin1: TStringField;

SQLQuerylid_k: TAutoIncField;

SQLQuerylnama1: TStringField;

SQLQuerylnama_kerusakan: TStringField;

SQLTransaction1: TSQLTransaction;

procedure btnBaruClick(Sender: TObject);

procedure btnHapusClick(Sender: TObject);

procedure btnSimpanClick(Sender: TObject);
```

```
procedure DBGrid1MouseUp(Sender: TObject; Button:
TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure FormCreate(Sender: TObject);
private
    procedure ExecSQL(var Ado:TSQLQuery;StrSQL:string);
    procedure OpenSQL(var Ado:TSQLQuery;StrSQL:string);
public
    { public declarations }
end;
```

```
var
```

```
FormKerusakan: TFormKerusakan;
```

```
implementation
```

```
{ $R *.lfm }
```

```
{ TFormKerusakan }
```

```
procedure TFormKerusakan.ExecSQL(var
Ado:TSQLQuery;StrSQL:string);
```

```
begin
```

```
    Ado.Close;
```

```
Ado.SQL.Clear;
```

```
Ado.SQL.Append(StrSQL);
```

```
Ado.ExecSQL;
```

```
end;
```

```
procedure TFormKerusakan.OpenSQL(var  
Ado:TSQLQuery;StrSQL:string);
```

```
begin
```

```
Ado.Close;
```

```
Ado.SQL.Clear;
```

```
Ado.SQL.add(StrSQL);
```

```
Ado.Prepared;
```

```
Ado.Open;
```

```
end;
```

```
procedure TFormKerusakan.FormCreate(Sender: TObject);
```

```
begin
```

```
dbCon.Connected:= true;
```

```
SQLQuery1.Active:=True;
```

```
btnBaruClick(Sender);
```

```
end;
```

```
procedure TFormKerusakan.btnBaruClick(Sender: TObject);
```

```
begin
```

```
A.Clear;
```

```
A.Enabled:= true;
```

```
btnSimpan.Caption:='Simpan';
```

```
btnHapus.Enabled := false;
```

```
pnlMain.Enabled := True;
```

```
end;
```

```
procedure TFormKerusakan.btnHapusClick(Sender: TObject);
```

```
var
```

```
StrSQL : string;
```

```
begin
```

```
if (DBEdit1.Text='') then exit;
```

```
if (MessageDlg('Yakin Untuk Menghapus Data  
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then
```

```
begin
```

```
StrSQL :='DELETE FROM kerusakan WHERE id_k =  
'+DBEdit1.Text +' ' ;
```

```
ExecSQL(Conn,StrSQL);
```

```
end;
```

```
btnBaruClick(sender);
```

```
SQLQuery1.Refresh ();
```

```
dbGrid1.Refresh;
```

```
end;
```

```
procedure TFormKerusakan.btnSimpanClick(Sender: TObject);
```

```
var
```

```
StrSql : string;
```

```
strCode : string;
```

```
i : integer;
```

```
begin
```

```
if (btnSimpan.Caption <>'Edit') then
```

```
begin
```

```
if (trim(A.Text)='') )
```

```
then
```

```
begin
```

```
MessageDlg('Inputan Tidak Boleh  
Kosong',mtError,[mbOK],0);
```

```
exit;
```

```
end;
```

```
end;
```

```
if (btnSimpan.Caption ='Simpan') then
```

```
begin
```

```
if (MessageDlg('Yakin Untuk Menyimpan Data  
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then
```

```
begin
```

```
OpenSQL( conn,'SELECT * FROM kerusakan WHERE  
nama_kerusakan ='"+A.Text+"'");
```

```
if ( conn.RecordCount = 0) then
```

```
ExecSQL( conn,'INSERT INTO kerusakan  
(nama_kerusakan ) VALUES ("'+trim(A.Text)+'" )')
```

```
else
```

```

                MessageDlg('Data Sudah Ada Dalam
Database..',mtError,[mbOK],0);

                btnBaruClick(sender);

                SQLQuery1.Refresh();

                DBGrid1.Refresh;

        end;

end else if (btnSimpan.Caption ='Edit') then

begin

        strSQL :='SELECT * FROM kerusakan WHERE id_k=
''+DBEdit1.Text+'"' ;

        OpenSQL(conn,strSQL);

        PnlMain.Enabled := true;

        while (not( Conn.Eof)) do

        begin

                A.Text

:=trim(Conn.fieldByName('nama_kerusakan').AsString);

                Conn.Next;

        end;

        btnSimpan.Caption := 'Update';

        Conn.Close;

end else if (btnSimpan.Caption ='Update') then

begin

        if (MessageDlg('Yakin Untuk Mengupdate Data
?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then

```

```

begin
    StrSQL := 'UPDATE kerusakan SET nama_kerusakan
="' + trim(A.Text) + '" WHERE id_k= "' + DBEdit1.Text + '" ' ;
    ExecSQL( Conn, StrSQL);
end;

btnBaruClick(sender);

SQLQuery1.Refresh();

DBGrid1.Refresh;

end;

end;

procedure TFormKerusakan.DBGrid1MouseUp(Sender: TObject;
Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
    btnBaruClick(self);

    btnSimpan.Caption := 'Edit';

    btnHapus.Enabled := True;

    PnlMain.Enabled := false;

end;

end.

```