



APLIKASI SISTEM PAKAR DIAGNOSIS PADA SISTEM KOMPUTER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat dalam Menempuh Ujian Sidang
Sarjana Program Studi Sistem Informasi

Oleh :

ARINI MARLYANINGRUM

1108014



PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK UNIVERSITAS WIDYATAMA
BANDUNG
2013

LEMBAR PENGESAHAN

APLIKASI SISTEM PAKAR DIAGNOSIS PADA SISTEM KOMPUTER

Tugas Akhir

Program Studi Sistem Informasi

Fakultas Teknik

Universitas Widyatama

Oleh :

Arini Marlyaningrum

11.08.014

Telah disetujui dan disahkan di Bandung, Januari 2013

Pembimbing Kampus,

Sri Lestari,Ir.,M.T

NIP. 1968110111994032001

Ka.Prodi Sistem Informasi

Dekan Fakultas Teknik

M. Rozahi Istambul, S.Kom., M.T.

NID. 0414106701

Setiadi Yazid, Ir.,M.Sc.,Ph.D.

NID. 0315085402

SURAT PERNYATAAN

Saya yang bertandatangan dibawah ini :

Nama : Arini Marlyaningrum

NRP : 1108014

Tempat dan Tanggal Lahir : Bandung, 13 Maret 1990

Alamat : Jl. Cukang-kawung VI No.175/149a RT 08 RW 13
Bandung 40191

Menyatakan bahwa laporan Tugas Akhir ini adalah benar hasil karya saya sendiri.
Bila terbukti tidak demikian, saya bersedia menerima segala akibatnya.

Bandung, Januari 2013

Arini Marlyaningrum

ABSTRAK

Sistem pakar dikembangkan sejalan dengan adanya teknologi informasi. Pembangunan sistem pakar bertujuan sebagai sarana bantu untuk memberikan solusi di dalam kehidupan kita.. Sistem pakar ini dapat membantu teknisi dalam mendapatkan solusi dengan cepat dan dapat menghemat waktu. Sistem pakar menggunakan teknologi komputer untuk mengintegrasikan, memanipulasi dan menampilkan informasi atau karakteristik. Sistem pakar juga dapat membantu dalam pengambilan solusi yang lebih baik.

Dengan kemajuan teknologi yang sangat pesat saat ini, memunculkan suatu ide atau gagasan dari penulis untuk mencoba mengimplementasikan salah satu program aplikasi sistem pakar ke dalam aktivitas mutu pelayanan teknisi komputer. Penulis mencoba membangun sebuah aplikasi yang akan membantu untuk mempermudah dalam memberikan solusi kerusakan komputer pada hardware sehingga dapat mempermudah pengguna atau teknisi untuk mendapatkan solusi dengan cepat.

Sistem yang akan dibuat adalah “Perancangan Sistem Pakar Diagnosis Masalah Pada Komputer Menggunakan Visual Basic” Sistem ini akan menggunakan metode *prototype* serta *tool* untuk pemodelan menggunakan UML (*Unified Modeling Language*).

Sistem ini dibangun menggunakan aplikasi *Visual Basic 6.0* untuk mengolah database *Microsoft Acces*.

Kata Kunci : *Prototype, UML(Unifiield Modeling Language), Aplikasi Sistem Pakar, menggunakan Visual Basic 6.0*

ABSTRACT

An expert system was developed in line with the information technology. Development of an expert system is intended as a means to help provide solutions in our lives .. This expert system could help technicians in getting solutions quickly and can save time. Expert systems use computer technology to integrate, manipulate and display information or characteristics. Expert systems can also help in making a better solution.

With the rapid advances in technology today, bring an idea or ideas of the author to try to implement a program of expert system application into a computer technician service quality activities. The author tries to build an application that will help to facilitate in providing computer malfunction in the hardware solutions that can facilitate users or engineers to find a solution quickly.

The system to be made is "Diagnosis Expert System Design Issues in Computer Using Visual Basic" The system will use the method and prototype tool for modeling using UML (Unified Modeling Language).

The system is built using Visual Basic 6.0 applications to process Microsoft Access database.

Keywords: *Prototype, UML (Unified Modeling Language), Systems Applications, using Visual Basic 6.0*

KATA PENGANTAR

Dengan Puji dan Syukur kepada Tuhan yang Maha Esa yang telah melimpahkan berkat dan anugerah-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul **“APLIKASI SISTEM PAKAR DIAGNOSIS PADA SISTEM KOMPUTER”** dengan baik.

Laporan ini disusun sebagai syarat kelulusan dan syarat menempuh sidang Sarjana Program Studi Sistem Informasi Universitas Widyatama, Bandung.

Dalam penyusunan Laporan Tugas Akhir dan pembuatan aplikasi ini, penulis menyadari banyak sekali pihak yang memberikan bantuan, bimbingan dan petunjuk, baik langsung maupun tidak langsung. Oleh karena itu, penulis mengucapkan banyak terimakasih dan memberikan penghargaan yang tinggi kepada:

1. Allah SWT yang selalu melimpahkan rahmatnya kepada penulis dalam setiap langkah pembuatan aplikasi sampai penulisan Laporan Tugas Akhir ini.
2. Kedua orang tua, ketiga kakakku, dan keponakan-keponakanku yang selalu memberikan dukungan moril, materi ataupun non-materi. Serta bimbingan, kesabaran, ketabahan, dan kebijakan dalam mendidik.
3. Bapak Setiadi Yazid, Ir., M.Sc., Ph.D., selaku Dekan Fakultas Teknik.
4. Bapak M. Rozahi Istambul, S.Komp., M.T., selaku Ketua Jurusan Sistem Informasi yang memberikan ilmu kepada penulis, baik di dalam kelas, ataupun diluar kelas, dan motivasi kepada penulis.
5. Ibu Sri Lestari, Ir., M.T. selaku Pembimbing yang telah meluangkan waktu, tenaga, dan pikirannya dalam mengarahkan dan membimbing penulis dengan penuh kesabaran sampai terselesaikannya penyusunan Tugas Akhir ini.
6. Rekan-rekan Mahasiswa Sistem Informasi Angkatan 2008
7. Rekan-rekan HIMASI (Himpunan Mahasiswa Sistem Informasi) Angkatan 2008 – 2013.

8. Rekan-rekan Recok's and the Genk, Gocap Genk, dan Tian yang senantiasa mengajarkan pada penulis untuk terus bermimpi dan tidak pernah menyerah.
9. Semua pihak-pihak yang telah membantu dan pernah hadir di kehidupan penulis yang senantiasa memberikan inspirasi dan motivasi, yang tidak bisa disebutkan satu per satu namanya.

Semoga Tuhan memberikan balasan yang berlipat, atas segala kebaikan yang telah mereka berikan kepada penulis.

Akhir kata, Penulis mengharapkan kritik dan saran yang membangun untuk laporan tugas akhir ini, dan semoga laporan dan aplikasi ini dapat berguna bagi semua orang yang membaca laporan ini.

Bandung, Januari 2013

Arini Marlyaningrum
Penulis

DAFTAR ISI

LEMBAR JUDUL

LEMBAR PENGESAHAN	i
SURAT PERNYATAAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii

BAB I PENDAHULUAN

1.1 Latar Belakang	I-1
1.2 Identifikasi Masalah	I-1
1.3 Rumusan Masalah	I-2
1.4 Maksud dan Tujuan	I-2
1.5 Batasan Masalah	I-2
1.6 Metodologi Penelitian	I-3
1.7 Sistematika Penulisan	I-4

BAB II LANDASAN TEORI

2.1 Artificial Intelligence	II-1
2.2 Sistem Pakar	II-2
2.2.1 Ciri-ciri Sistem Pakar	II-2
2.2.2 Keuntungan Sistem Pakar	II-2
2.2.3 Kelemahan Sistem Pakar	II-3
2.2.4 Perbandingan Antara Seorang Pakar dan Sistem Pakar	II-4
2.2.5 Komponen Sistem Pakar	II-4
2.2.6 Pengembangan Mesin Inferensi	II-5

2.3	Metodologi Yang Digunaka	II-6
2.3.1	<i>Prototyping-based Methodologi</i>	II-6
2.3.2	<i>Object Oriented Programming (OOP)</i>	II-7
2.3.3	<i>Unified Modeling Language (UML)</i>	II-3
2.4	Tools untuk Pengembangan Perangkat Lunak	II-15
2.4.1	<i>Microsoft Access</i>	II-15
2.4.2	<i>Visual Basic</i>	II-17

BAB III ANALISIS SISTEM

3.1	Sistem Lama	III-1
3.2	Sistem Yang Akan Dibuat	III-1
3.3	Identifikasi Masalah	III-2
3.4	Kaidah	III-10
3.5	Pohon Keputusan	III-15
3.6	Analisis Perangkat Keras	III-22
3.7	Analisis Perangkat Lunak	III-22

BAB IV PERANCANGAN PERANGKAT LUNAK

4.1	Perancangan Sistem	IV-1
4.2	Perancangan Menu	IV-1
4.2.1	Penjelasan Perancangan Menu	IV-2
4.3	Model <i>Use Case</i>	IV-3
4.3.1	<i>Use Case Diagram</i>	IV-3
4.3.2	Definisi Aktor	IV-4
4.3.3	Definisi <i>Use Case</i>	IV-4
4.3.4	Skenario <i>Use Case</i>	IV-5
4.4	<i>Activity Diagram</i>	IV-5
4.4.1	Skenario <i>Use Case Login Admin</i>	IV-6
4.4.2	<i>Activity Diagram Login</i>	IV-7
4.4.3	Skenario <i>Use Case Admin Manage Menu Aturan</i>	IV-7

4.4.4	<i>Activity Diagram Use Case Login Admin Manage</i>	
	Menu Aturan.....	IV-8
4.4.5	<i>Skenario Use Case Admin Manage Menu Ciri Kerusakan</i>	IV-8
4.4.6	<i>Activity Diagram Admin Manage Menu Ciri Kerusakan</i>	IV-9
4.4.7	<i>Skenario Use Case Admin Manage Menu Macam Kerusakan</i>	IV-9
4.4.8	<i>Activity Diagram Admin Manage Menu Macam Kerusakan</i>	IV-10
4.4.9	<i>Skenario Use Case Admin Manage Menu Jenis Kerusakan</i>	IV-10
4.4.10	<i>Activity Diagram Admin Manage Menu Jenis Kerusakan</i>	IV-11
4.4.11	<i>Skenario Use Case Admin Manage Menu Keterangan</i>	IV-11
4.4.12	<i>Activity Diagram Admin Manage Menu Keterangan</i>	IV-12
4.4.13	<i>Skenario Use Case Admin Manage Menu Penjelasan</i>	IV-12
4.4.14	<i>Activity Diagram Admin Manage Menu Penjelasan</i>	IV-13
4.4.15	<i>Skenario Use Case User Menggunakan Sistem Pakar</i>	IV-13
4.4.16	<i>Activity Diagram User Menggunakan Sistem Pakar</i>	IV-14
4.5	<i>Realisasi Use Case Tahap Rancangan</i>	IV-14
4.5.1	<i>Class Diagram</i>	IV-14
4.5.2	<i>Kamus Data</i>	IV-15
4.5.3	<i>Sequence Diagram</i>	IV-18
4.5.3.1	<i>Sequence Diagram Login</i>	IV-18
4.5.3.2	<i>Sequence Diagram Admin Manage Menu Aturan</i>	IV-19
4.5.3.3	<i>Sequence Diagram Admin Manage Macam Kerusakan</i>	IV-20
4.5.3.4	<i>Sequence Diagram Admin Manage Ciri Kerusakan</i>	IV-20
4.5.3.5	<i>Sequence Diagram Admin Manage Jenis Kerusakan</i>	IV-21
4.5.3.6	<i>Sequence Diagram Admin Manage Menu Penjelasan</i>	IV-21
4.5.3.7	<i>Sequence Diagram User Menggunakan Sistem Pakar</i>	IV-22
4.5.4	<i>Collaboration Diagram Login</i>	IV-22
4.5.4.1	<i>Collaboration Diagram Login</i>	IV-23
4.5.4.2	<i>Collaboration Diagram Admin Manage Menu Aturan</i>	IV-23
4.5.4.3	<i>Collaboration Diagram Admin Manage Ciri Kerusakan</i>	IV-24
4.5.4.4	<i>Collaboration Diagram Admin Manage Macam Kerusakan</i>	IV-24

4.5.4.5	<i>Collaboration Diagram Admin Manage Jenis Kerusakan</i>	IV-25
4.5.4.6	<i>Collaboration Diagram Admin Manage Menu</i>	
	Penjelasan	IV-25
4.5.4.7	<i>Collaboration Diagram User Menggunakan Sistem Pakar</i> ...	IV-26
4.6	Lingkungan Operasional	IV-26
4.7	Perangkat Lunak	IV-26
4.8	Perangkat Keras	IV-27
4.9	Karakteristik Pengguna	IV-27
4.10	<i>Layout Antarmuka</i>	IV-27
4.10.1	Antarmuka Tampilan Utama Sistem	IV-28
4.10.2	Antarmuka Menu Admin <i>Login</i>	IV-28
4.10.3	Antarmuka Menu Admin Aturan	IV-28
4.10.4	Antarmuka Menu Admin Ciri Kerusakan	IV-30
4.10.5	Antarmuka Menu Admin Macam Kerusakan	IV-30
4.10.6	Antarmuka Menu Admin Jenis Kerusakan	IV-31
4.10.7	Antarmuka Menu Admin Penjelasan	IV-32
4.10.8	Antarmuka Menu Penelusuran	IV-32
4.10.9	Antarmuka Menu Penelusuran Berdasarkan Macam Kerusakan	IV-32
4.10.10	Antarmuka Menu Penelusuran Berdasarkan Jenis Kerusakan	IV-33
4.10.11	Antarmuka Menu Penelusuran Berdasarkan Ciri Kerusakan	IV-34
4.10.12	Antarmuka Menu Rekaman Data	IV-34
4.10.13	Antarmuka Menu Solusi	IV-35

BAB V IMPELMANTASI SISTEM

5.1	Implementasi	V-1
5.1.1	Lingkup dan Batasan Implementasi	V-1
5.2	Kebutuhan Sumber Daya	V-2
5.2.1	Kebutuhan Perangkat Keras (<i>Hardware</i>)	V-2
5.2.2	Kebutuhan Perangkat Lunak (<i>Software</i>)	V-2
5.3	Implementasi Antarmuka Pemakai	V-3

5.4	Hasil Pengujian	V-10
-----	-----------------------	------

BAB VI KESIMPULAN DAN SARAN

6.1	Kesimpulan	VI-1
-----	------------------	------

6.2	Saran	VI-1
-----	-------------	------

DAFTAR PUSTAKA	xiv
-----------------------------	-----

LAMPIRAN A

LAMPIRAN B



DAFTAR TABEL

1. Tabel 2.1 Perbandingan Seorang Pakar dan Sistem Pakar.....	II-4
2. Tabel 3.1 Spesifikasi PC	III-37
3. Tabel 3.2 Spesifikasi <i>Software</i>	III-38
4. Tabel 4.1 Definisi <i>Actor</i>	IV-4
5. Tabel 4.2. Definisi <i>UseCase</i>	IV-4
6. Tabel 4.5 Skenario <i>UseCase</i> Aplikasi Sistem Pakar Diagnosa Kerusakan Komputer	IV-5
7. Tabel 4.4 Skenario <i>UseCase Login</i> Admin	IV-6
8. Tabel 4.5 Skenario <i>UseCase</i> Admin Manage Menu Aturan	IV-7
9. Tabel 4.6 Skenario <i>UseCase</i> Admin Manage Ciri Kerusakan	IV-8
10. Tabel 4.7 Skenario <i>UseCase</i> Admin Manage Macam Kerusakan	IV-9
11. Tabel 4.8 Skenario <i>UseCase</i> Admin Manage Jenis Kerusakan	IV-10
12. Tabel 4.9 Skenario <i>UseCase</i> Admin Manage Menu Keterangan	IV-11
13. Tabel 4.10 Skenario <i>UseCase</i> Admin Manage Menu Penjelasan	IV-12
14. Tabel 4.11 Skenario <i>UseCase User</i> Menggunakan Sistem Pakar	IV-13
15. Tabel 4.12 Tabel <i>Login</i>	IV-16
16. Tabel 4.13 Tabel Ciri Kerusakan	IV-16
17. Tabel 4.14 Tabel Macam Kerusakan	IV-17
18. Tabel 4.15 Tabel Jenis Kerusakan	IV-17
19. Tabel 4.16 Tabel Relasi 1	IV-18
20. Tabel 4.17 Tabel Relasi 2	IV-18
21. Tabel 4.18 Karakteristik Pengguna Untuk Admin	IV-27
22. Tabel 5.1 Spesifikasi Minimum Perangkat Keras Yang Digunakan	V-2
23. Tabel 5.2 Pengujian Perangkat Lunak Pada Admin	V-7
24. Tabel 5.3 Pengujian Perangkat Lunak Pada User	V-8
25. Tabel 5.3 Pengujian Perbandingan Diagnosa Kerusakan Komputer Secara Manual dan Menggunakan Aplikasi	V-9

DAFTAR GAMBAR

1. Gambar 1.1 Model <i>Prototype</i>	I-3
2. Gambar 2.1 <i>Struktur</i> Sistem Pakar	II-6
3. Gambar 2.2 Model <i>Prototype</i>	II-7
4. Gambar 2.3 Contoh <i>Actor</i>	II-9
5. Gambar 2.4 Contoh <i>UseCase</i>	II-9
6. Gambar 2.5 Contoh <i>Class</i>	II-10
7. Gambar 2.6 <i>Start</i> dan <i>EndState</i>	II-11
8. Gambar 2.7 Transisi dari <i>Estate_1</i> Ke <i>Estate_2</i> Yang Diwakilin Tanda Panah	II-12
9. Gambar 2.8 <i>Actor</i> Pada <i>Sequence</i> Diagram	II-12
10. Gambar 2.9 Contoh <i>Object LifeLine</i>	II-13
11. Gambar 2.10 Contoh <i>Activation</i>	II-13
12. Gambar 2.11 Contoh <i>Massage</i>	II-13
13. Gambar 2.12 Contoh <i>Collaboration</i> Diagram	II-14
14. Gambar 2.13 Contoh <i>Activity</i> Diagram	II-15
15. Gambar 2.14 Komponen Utaman (<i>Object</i>)	II-15
16. Gambar 2.15 Komponen <i>Visual Basic</i> (VB)	II-18
17. Gambar 3.1 Pohon Keputusan Kerusakan Pada PowerSupply	III-16
18. Gambar 3.2 Pohon Keputusan Kerusakan Pada Motherboard	III-17
19. Gambar 3.3 Pohon Keputusan Kerusakan Pada Harddisk	III-18
20. Gambar 3.4 Pohon Keputusan Kerusakan Pada CD-ROOM	III-19
21. Gambar 4.1 Perancangan Menu	IV-2
22. Gambar 4.2 <i>UseCase</i> Diagram.....	IV-3
23. Gambar 4.3 <i>Activity</i> Diagram Aplikasi Sistem Pakar	IV-6
24. Gambar 4.4 <i>Activity</i> Diagram <i>Login</i> Admin	IV-7
25. Gambar 4.5 <i>Activity</i> Diagram <i>UseCase</i> Admin Manage Menu Aturan..	IV-8
26. Gambar 4.6 <i>Activity</i> Diagram Admin Manage Menu Ciri Kerusakan....	IV-9
27. Gambar 4.7 <i>Activity</i> Diagram Admin Manage Menu Macam	

Kerusakan.....	IV-10
28. Gambar 4.8 <i>Activity Diagram</i> Admin Manage Menu Jenis Kerusakan ..	IV-11
29. Gambar 4.9 <i>Activity Diagram</i> Admin Manage Menu Keterangan.....	IV-12
30. Gambar 4.10 <i>Activity Diagram</i> Admin Manage Menu Penjelasan	IV-13
31. Gambar 4.11 <i>Activity Diagram</i> User Menggunakan Sistem Pakar	IV-14
32. Gambar 4.12 <i>Class Diagram</i>	IV-15
33. Gambar 4.13 <i>Sequence Diagram</i> Login	IV-19
34. Gambar 4.14 <i>Sequence Diagram</i> Admin Manage Menu Aturan	IV-19
35. Gambar 4.15 <i>Sequence Diagram</i> Admin Manage Macam Kerusakan....	IV-20
36. Gambar 4.16 <i>Sequence Diagram</i> Admin Manage Ciri Kerusakan	IV-20
37. Gambar 4.17 <i>Sequence Diagram</i> Admin Manage Jenis Kerusakan.....	IV-21
38. Gambar 4.18 <i>Sequence Diagram</i> Admin Manage Menu Penjelasan	IV-21
39. Gambar 4.19 <i>Sequence Diagram</i> Admin User Manggunakan	
Sistem Pakar.....	IV-22
40. Gambar 4.20 <i>Collaboration Diagram</i> Login.....	IV-23
41. Gambar 4.21 <i>Collaboration Diagram</i> Admin Manage Menu Aturan....	IV-23
42. Gambar 4.22 <i>Collaboration Diagram</i> Admin Manage Ciri Kerusakan..	IV-24
43. Gambar 4.23 <i>Collaboration Diagram</i> Admin Manage	
Macam Kerusakan.....	IV-24
44. Gambar 4.24 <i>Collaboration Diagram</i> Admin Manage	
Jenis Kerusakan.....	IV-25
45. Gambar 4.25 <i>Collaboration Diagram</i> Admin Manage	
Menu Penjelasan	IV-25
46. Gambar 4.26 <i>Collaboration Diagram</i> User Menggunakan	
Sistem Pakar.....	IV-26
47. Gambar 4.27 Anatarmuka Tampilan Utama Sistem	IV-28
48. Gambar 4.28 Anatarmuka Menu Admin Login	IV-28
49. Gambar 4.29 Anatarmuka Menu Admin Aturan	IV-29
50. Gambar 4.30 Anatarmuka Menu Admin Ciri Kerusakan	IV-30
51. Gambar 4.31 Anatarmuka Menu Admin Macam Kerusakan	IV-30
52. Gambar 4.32 Anatarmuka Menu Admin Jenis Kerusakan	IV-31

53. Gambar 4.33 Anatarmuka Menu Admin Penjelasan	IV-31
54. Gambar 4.34 Anatarmuka Menu Penelusuran	IV-32
55. Gambar 4.35 Anatarmuka Menu Penelusuran Berdasarkan Macam Kerusakan.....	IV-32
56. Gambar 4.36 Anatarmuka Menu Penelusuran Berdasarkan Jenis Kerusakan.....	IV-33
57. Gambar 4.37 Anatarmuka Menu Penelusuran Berdasarkan Ciri Kerusakan	IV-34
58. Gambar 4.38 Anatarmuka Menu Rekaman Data	IV-34
59. Gambar 4.39 Anatarmuka Menu Solusi.....	IV-35
60. Gambar 5.1 Menu <i>Login</i>	V-3
61. Gambar 5.2 Menu Utama.....	V-3
62. Gambar 5.3 Menu Penelusuran	V-4
63. Gambar 5.4 Menu Penelusuran Macam Kerusakan.....	V-4
64. Gambar 5.5 Menu Penelusuran Jenis Kerusakan.....	V-5
65. Gambar 5.6 Menu Penelusuran Ciri Kerusakan.....	V-5
66. Gambar 5.7 Menu Rekaman Data Yang Akan Di Analisis	V-6
67. Gambar 5.8 Menu Solusi	V-6



CURRICULUM VITAE

Nama : ARINI MARLYANINGRUM
Jenis Kelamin : Perempuan
Tanggal Lahir : 13 Maret 1990
Tempat Lahir : Bandung
Agama : Islam
Alamat : Jl. Cukang-kawung VI No.175/149 a RT 08 RW 13
Bandung 40191
Telepon : 082129062665
Email : arini_marlya@yahoo.com
Tinggi / berat badan : 160 cm / 78 Kg
Status : Belum Menikah
Kewarganegaraan : Indonesia

PENDIDIKAN (FORMAL)

Sekolah Dasar : SDN. Ciujung VI Bandung, 1996 - 2002
SMP : SMP Negeri 27 Bandung, 2002- 2005
SMA : SMA Pasundan 2 Bandung, 2005 – 2008
Perguruan Tinggi : Universitas Widyatama Bandung,
Jurusan Sistem Informasi, 2008 – 2013

PENGALAMAN ORGANISASI

- Himpunan Mahasiswa Sistem Informasi, Universitas Widyatama, 2008 –
sampai sekarang
 - Anggota
 - Devisi Bendahara Mapi 2009

Demikian CV ini Saya buat dengan sebenarnya

Hormat Saya,

Arini Marlyaningrum



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Sistem pakar dikembangkan sejalan dengan adanya teknologi informasi. Pembangunan sistem pakar bertujuan sebagai sarana bantu untuk memberikan solusi di dalam kehidupan kita. Salah satu contohnya adalah kasus kerusakan komputer merupakan kasus yang memerlukan bantuan seorang pakar (teknisi) dalam menyelesaikan masalah dengan mengandalkan pengetahuan yang dimilikinya. Saat ini teknisi membutuhkan waktu yang cukup lama untuk mendiagnosa kerusakan komputer, bahkan teknisi sering sekali menunda perkerjanya hanya untuk menghasilkan solusi dari kerusakan komputer. Pada tugas akhir ini dibuat perangkat lunak untuk mengatasi kasus kerusakan komputer, dimana bisa membantu para teknisi untuk mendiagnosa kerusakan komputer dengan menghemat waktu. Di aplikasi ini dapat membantu teknisi senior untuk mendapatkan solusi dengan cepat dan teknisi junior apabila ada jenis dan ciri-ciri yang di temukan lagi bisa menambahkannya di basis aturan. Perangkat lunak yang dimaksud adalah sistem pakar untuk diagnosis kerusakan komputer yang di desain untuk memodelkan/mengemulasi kemampuan seorang pakar dalam memecahkan suatu masalah yang berbasiskan pada pengetahuan pakar itu sendiri. Bahasa pemrograman yang digunakan sebagai pembangun sistem adalah Visual Basic 6.0 dan Microsoft Access untuk pengolahan database-nya.

1.2 Identifikasi Masalah

Setelah mempelajari dan mengamati sistem yang akan dibangun maka ditemukan beberapa permasalahan dalam perencanaan pengembangan Sistem yang akan dibangun yaitu :

1. Saat ini apabila ada kerusakan komputer maka teknisi membutuhkan waktu yang lebih lama untuk mendiagnosanya.
2. Membantu teknisi dalam memberikan solusi kerusakan komputer.

1.3 Rumusan Masalah

1. Bagaimana cara membantu teknisi dalam memudahkan untuk mendiagnosa masalah yang berkaitan dengan kerusakan komputer?
2. Bagaimana membuat aplikasi yang dapat mempercepat teknisi dalam mendiagnosa kerusakan komputer?

1.4 Maksud dan Tujuan

Maksud dan Tujuan dari diadakannya penelitian, perancangan, dan pembuatan sistem pakar komputerisasi dalam menunjang penulisan skripsi ini adalah untuk :

1. Membangun sebuah sistem pakar yang dapat mempercepat teknisi dalam hal mendiagnosa kerusakan pada komputer.
2. Memberikan suatu solusi dengan merancang, dan mengimplementasikan sistem pakar komputerisasi yang telah dibuat dan akan digunakan untuk keperluan umum.

1.5 Batasan Masalah

Batasan masalah tugas akhir ini, yaitu:

1. Sistem pakar penanganan kerusakan komputer ini menggunakan bahasa pemrograman Visual Basic 6.0 dan Microsoft Access untuk pengolahan database-nya
2. Sistem pakar ini hanya terbatas mencakup pada permasalahan hardware.
3. Pembangunan sistem pakar menggunakan *tree* (pohon) dengan mesin inferensi *backward chaining* (penalaran mundur) dan *forward chaining* (penalaran maju).

1.6 Metodologi Penelitian

Tahapan yang dilakukan dalam penelitian adalah sebagai berikut:

1. Pengumpulan Data

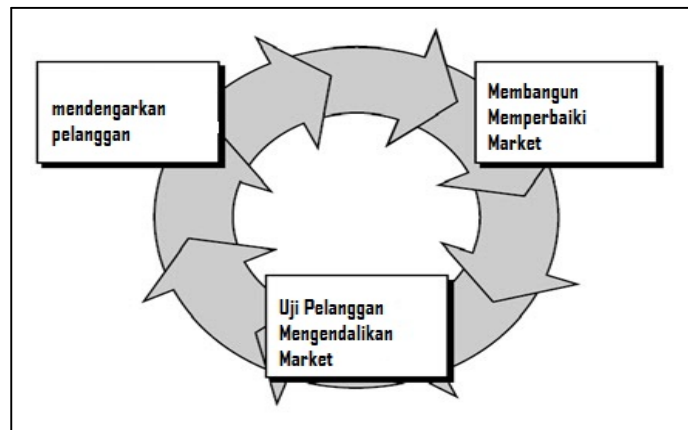
Wawancara, yaitu mengumpulkan data dan informasi secara langsung dengan narasumber yang terkait.

2. Pengimplementasian

Implementasi hasil perancangan yaitu pengkodean dengan menggunakan bahasa pemrograman yang telah ditetapkan.

3. Pengembangan Sistem

Metode yang digunakan dalam penelitian ini adalah metode *Prototyping* yaitu, metode yang dilakukan dengan cara mendengarkan kebutuhan pelanggan, membangun perangkat lunak dan memperbaiki serta melakukan uji coba perangkat lunak oleh pelanggan sampai aplikasi yang dibuat menjadi sebuah prototype yang kemudian disempurnakan menjadi sebuah produk baru.



Gambar 1.1 Model Prototype menurut *Roger S. Pressman, Ph.D.*

1.7 Sistematika Penulisan

Penyusunan tugas akhir ini diuraikan dalam beberapa bab dan sub bab yang tersusun sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang masalah, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, rencana aktifitas, dan sistematika penulisan

BAB II LANDASAN TEORI

Bab ini berisi dasar teori yang digunakan dalam pembuatan sistem pakar untuk melandasi pemecahan masalah.

BAB III ANALISIS SISTEM

Bab ini membahas tentang perancangan sistem pakar. Uraian perancang system pakar ini meliputi akuisisi pengetahuan, representasi pengetahuan, perancangan basis data.

BAB IV PERANCANGAN PERANGKAT LUNAK

Bab ini membahas mengenai batasan dan kebutuhan sistem baru, perancangan data, perancangan arsitektural, serta perancangan antarmuka.

BAB V TESTING DAN IMPLEMENTASI

Bab ini menjelaskan tentang implementasi sistem dan pengujian sistem yang telah dibuat, sehingga dari proses tersebut akan didapatkan sebuah analisa hasil yang akan menjawab permasalahan yang ada dalam tugas akhir.

BAB IV PENUTUP

Bab ini berisikan kesimpulan dari pembuatan tugas akhir ini serta saran untuk perbaikan mendatang disebabkan keterbatasan dalam pengetahuan dalam tugas akhir ini.

BAB II

LANDASAN TEORI

2.1 Artificial Intelligence

Artificial Intelligence dapat diartikan menjadi kecerdasan buatan, yang mana pada prosesnya berarti membuat, atau mempersiapkan, mesin seperti komputer agar memiliki sebuah *intelligence* atau kecerdasan berdasarkan perilaku manusia. Artificial Intelligence pada dasarnya bertujuan untuk membuat komputer melaksanakan suatu perintah, yang dapat dilakukan oleh manusia. Salah satu bagian dari artificial intelligence adalah sistem pakar.[1]

Bab ini dapat dikelompokkan atas teori-teori mengenai artificial intelligence, sistem pakar, aplikasi berbasis *Visual Basic 6.0*. dan *Microsoft Access* untuk pengolahan database-nya.

Dalam hal ini, terdapat beberapa pengertian dari kecerdasan buatan, antara Lain :

1. Kecerdasan buatan merupakan ilmu yang mempelajari bagaimana membuat sebuah komputer dapat mengerjakan sesuatu yang masih lebih baik dikerjakan manusia (*Rich dan Knight, 1991, p 3*).
2. Kecerdasan buatan merupakan solusi berbasis komputer terhadap masalah yang ada, yang menggunakan aplikasi yang mirip dengan proses berpikir menurut manusia (*Rolston, 1988, p 15*).
3. Artificial Intelligence atau kecerdasan buatan adalah cabang ilmu komputer yang mempelajari bagaimana komputer melakukan hal-hal yang pada saat yang sama orang menemukannya lebih baik (*Turban, 1995, p 422*).
4. Artificial intelligence adalah subdivisi dari ilmu komputer untuk membuat perangkat keras dan piranti lunak komputer sebagai usaha untuk memperoleh hasil seperti yang dihasilkan oleh manusia (*Turban, 1992, p 4*).

Menurut (*Turban, 1995, p 452 - 456*), artificial intelligence memiliki banyak bidang terapan, yaitu:

1. *Expert System* (sistem pakar)
2. *Natural Language Processing* (pemrosesan bahasa alamiah)
3. *Computer Vision and Scene Recognition*
4. *Intelligence Computer Aided Instruction*
5. Speech (voice) Understanding
6. *Robotics and Sensory System*

Dapat diambil kesimpulan dari beberapa paragraf diatas bahwa, kecerdasan buatan (artificial intelligence) adalah suatu metode untuk membuat sebuah komputer dapat memiliki kecerdasan dan dapat berpikir layaknya manusia dalam mencari jalan keluar suatu permasalahan, dan membagi proses-proses pemikiran tersebut menjadi sebuah langkah dasar pemecahan masalah.

2.2 Sistem Pakar

Terdapat beberapa pengertian atas sistem pakar, antara lain:

Sistem pakar adalah suatu sistem komputer yang dirancang agar dapat melakukan penalaran layaknya seorang pakar atau ahli pada suatu bidang keahlian tertentu (*Turban, 1993, p 74*).

Sistem pakar adalah suatu sistem komputer yang mengambil sifat pengetahuan dari manusia untuk menyelesaikan suatu masalah yang spesifik dan berfungsi sebagai asisten dari ahli itu sendiri (*Turban, 1995, p 74*).

Sistem pakar adalah sistem yang membutuhkan dasar pengetahuan yang baik, yang dibangun seefisien mungkin. Sistem ini memerlukan satu atau lebih mekanisme penalaran untuk menerapkan pengetahuan terhadap masalah yang dihadapi. Setelah itu dibutuhkan suatu mekanisme penalaran untuk menerapkan pengetahuan pada permasalahan yang ada (*Rich dan Knight, 1991, p 547*).

2.2.1 Ciri-ciri Sistem Pakar

Ciri-ciri sistem pakar menurut *Azis (1994, p 4)*, adalah sebagai berikut:

1. Terbatas pada domain tertentu.
2. Dapat memberikan solusi untuk data-data yang tidak lengkap.
3. Dapat mengemukakan rangkaian-rangkaian alasan yang diberikan dengan cara yang mudah dipahami.
4. Berdasarkan pada kaidah atau rule tertentu.
5. Dirancang untuk dapat dikembangkan secara bertahap.
6. Pengetahuan dan mekanisme inferensi yang jelas terpisah.
7. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh pemakai dengan dialog.

2.2.2 Keuntungan Sistem Pakar

Secara garis besar , banyak manfaat yang dapat diambil dengan adanya sistem pakar (*Kerschberg: 1986, Schur: 1988*):

1. Menjadikan pengetahuan dan nasihat lebih mudah didapat

2. Meningkatkan output dan produktivitas
3. Menyimpan kemampuan dan keahlian pakar
4. Meningkatkan penyelesaian masalah
5. Meningkatkan reliabilitas
6. Memberikan respon (jawaban) yang cepat
7. Merupakan panduan yang *intelligence* (cerdas)
8. Dapat bekerja dengan informasi yang kurang lengkap dan mengandung ketidakpastian
9. *Intelligence database* (basis data cerdas), bahwa sistem pakar dapat digunakan untuk mengakses basis data dengan cara cerdas

2.2.3 Kelemahan Sistem Pakar

Disamping sistem pakar mempunyai keuntungan, sistem pakar juga memiliki kelemahan (*M.Arhami: 2005*):

1. Untuk mendapatkan pengetahuan tidaklah selalu mudah, karena kadangkala pakar dari masalah yang kita buat tidak ada, dan walaupun ada, kadang-kadang pendekatan yang dimiliki oleh pakar tersebut berbeda-beda
2. Untuk membuat suatu sistem pakar yang benar-benar berkualitas tinggi sangatlah sulit dan memerlukan biaya yang sangat besar untuk pengembangan dan pemeliharaannya
3. Boleh jadi sistem tak dapat membuat keputusan
4. Sistem pakar perlu diuji ulang secara teliti sebelum digunakan, sehingga dalam hal ini faktor manusia tetaplah dominan

2.2.4 Perbandingan Antara Seorang Pakar dan Sistem Pakar

Keunggulan sistem pakar dibandingkan dengan seorang pakar, yaitu:

Seorang pakar	Sistem pakar
1. Memiliki batas (umur)	1. Tidak memiliki batasan waktu
2. Ilmu pengetahuan sulit di transfer	2. Mudah ditransfer
3. Dipengaruhi oleh situasi dan emosi	3. Tidak terpengaruh oleh emosi
4. Biaya tinggi	4. Relatif

Tabel 2.1 Perbandingan Seorang Pakar dan Sistem Pakar

2.2.5 Komponen Sistem Pakar

Sistem pakar sebagai sebuah program yang difungsikan untuk menirukan pakar manusia harus bias melakukan hal-hal yang dapat dikerjakan oleh seorang pakar. Untuk membangun sebuah sistem yang seperti itu maka komponen-komponen yang harus dimiliki adalah sebagai berikut (*Giarranto dan Riley, 2005*):

Komponen utama pada sistem pakar meliputi:

1. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan merupakan inti dari suatu sistem pakar, yaitu berupa representasi pengetahuan dari pakar. Basis pengetahuan tersusun atas fakta dan kaidah. Fakta adalah informasi tentang objek, peristiwa, atau situasi. Kaidah adalah cara untuk membangkitkan suatu fakta yang sudah diketahui.

2. Mesin Inferensi (*Inference Engine*)

Mesin inferensi berperan sebagai otak dari system pakar. Mesin inferensi berfungsi untuk memandu proses penalaran terhadap suatu kondisi, berdasarkan pada basis pengetahuan yang tersedia. Di dalam mesin inferensi terjadi proses untuk memanipulasi dan mengarahkan kaidah model, dan fakta yang disimpan dalam basis pengetahuan dalam rangka mencapai solusi atau kesimpulan. Dalam prosesnya, mesin inferensi menggunakan strategi penalaran

dan strategi pengendalian. Strategi penalaran terdiri dari strategi penalaran pasti (*Exact Reasoning*) dan strategi penalaran yang tidak pasti (*Inexact Reasoning*). *Exact reasoning* akan dilakukan jika semua data yang dibutuhkan untuk menarik suatu kesimpulan tersedia, sedangkan *inexact reasoning* dilakukan pada keadaan sebaliknya. Strategi pengendalian berfungsi sebagai panduan arah dalam melakukan proses penalaran. Terdapat tiga teknik pengendalian yang sering digunakan, yaitu *forward chaining*, *backward chaining*, dan gabungan dari kedua teknik pengendalian tersebut.

3. Basis Data (*Data Base*)

Basis data terdiri atas semua fakta yang diperlukan, dimana fakta-fakta tersebut digunakan untuk memenuhi kondisi dari kaidah-kaidah dalam sistem. Basis data menyimpan semua fakta, baik fakta awal pada saat sistem mulai beroperasi, maupun fakta-fakta yang diperoleh pada saat proses penarikan kesimpulan sedang dilaksanakan. Basis data digunakan untuk menyimpan data hasil observasi dan data lain yang dibutuhkan selama pemrosesan.

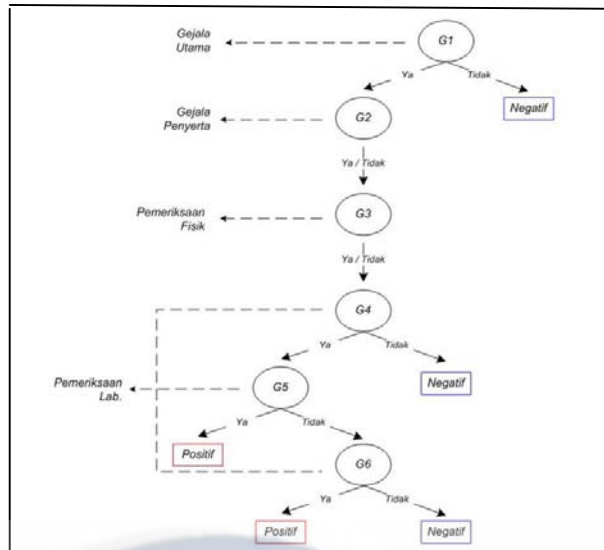
4. Antarmuka Pemakai (*User Interface*)

Fasilitas ini digunakan sebagai perantara komunikasi antara pemakai dengan komputer.

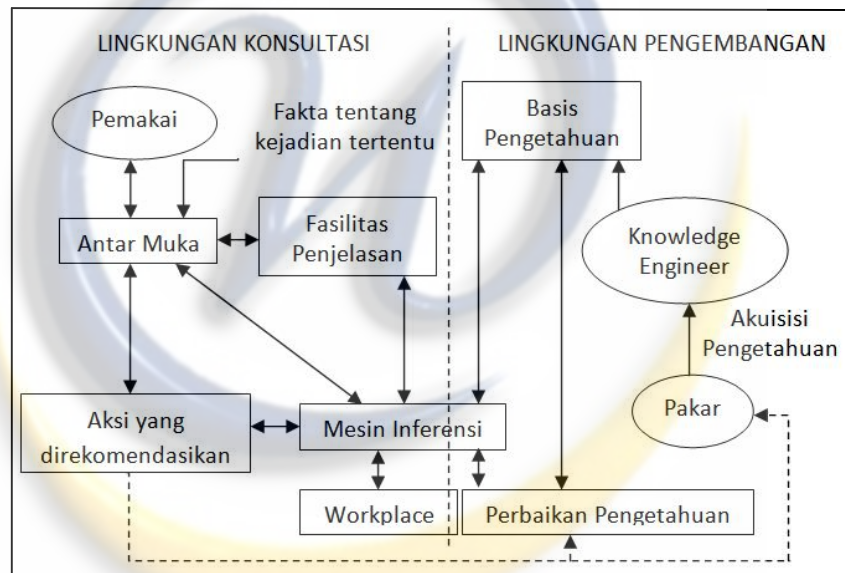
2.2.6 Pengembangan Mesin Inferensi

Forward Chaining

Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesa. *Forward chaining* merupakan grup dari multipel inferensi yang melakukan pencarian dari suatu masalah kepada solusinya. Jika, suatu klausa premis sesuai dengan inferensi (berniali TRUE), maka proses akan meng-assert konklusi. *Forward chaining* adalah *data driven* karena inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika, suatu aplikasi menghasilkan tree yang lebar dan tidak dalam, maka digunakan *forward cahinig*.



Gambar 2.1 Pohon Forward Chaining



Gambar 2.2 Struktur Sistem Pakar

Turban (2001) menyatakan bahwa konsep dasar dari suatu sistem pakar mengandung beberapa unsur/elemen, yaitu keahlian, ahli, pengalihan keahlian, inferensi aturan dan kemampuan menjelaskan.

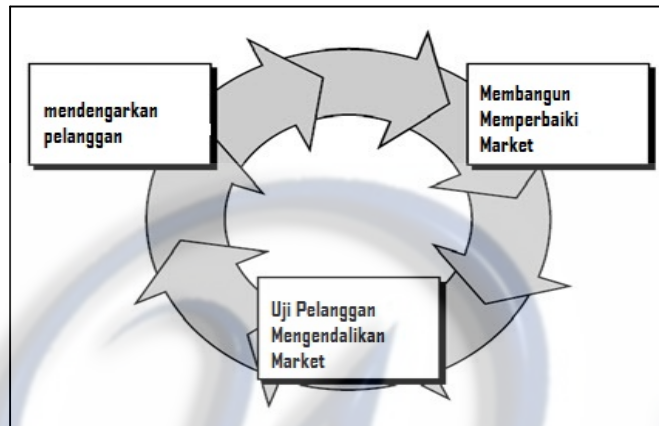
2.3 Metodologi Yang Digunakan

Adapun metodologi yang digunakan dalam pengembangan perancangan sistem pakar antara lain meliputi: *Prototyping-based Methodology*, *Object Oriented Programming* dan *Unified Modeling Language (UML)*.

2.3.1 Prototyping-based Methodology

Salah satu model rekayasa perangkat lunak adalah *Prototyping-based Methodology*. Pemodelan ini memungkinkan untuk pengembangan piranti lunak yang jauh lebih cepat dibanding metode *waterfall*.

Berikut adalah gambaran pengembangan system perangkat lunak atau *System Development Life Cycle (SDLC)* dengan menggunakan metode *prototyping*. [3]



Gambar 2.3 Model Prototype menurut Roger S. Pressman, Ph.D. [3]

2.3.2 Object Oriented Programming (OOP)

Object Oriented Programming (OOP) adalah suatu metode pemrograman yang berbasiskan pada objek, secara singkat pengertian dari OOP adalah koleksi objek yang saling berinteraksi dan saling memberikan informasi satu dengan yang lainnya. Suatu program disebut dengan pemrograman berbasis obyek (OOP) karena terdapat:

1. Encapsulation (pembungkusan)

1. Variabel dan method dalam suatu obyek dibungkus agar terlindungi.
2. Untuk mengakses, variabel dan method yang sudah dibungkus tadi perlu interface.
3. Setelah variabel dan method dibungkus, hak akses terhadapnya bisa ditentukan.
4. Konsep pembungkusan ini pada dasarnya merupakan perluasan dari tipe data struktur.

2. Inheritance (pewarisan)

1. Sebuah class bisa mewariskan atribut dan method-nya ke class yang lain.

2. Class yang mewarisi disebut *superclass*.
3. Class yang diberi warisan disebut *subclass*.
4. Sebuah *subclass* bisa mewariskan atau berlaku sebagai *superclass* bagi *class* yang lain disebut *multilevelinheritance*.

Keuntungan Penggunaan Pewarisan

- a. *Subclass* memiliki atribut dan method yang spesifik yang membedakannya dengan *superclass*, meskipun keduanya mirip (dalam hal kesamaan atribut dan method).
- b. Dengan demikian pada pembuatan *subclass*, programmer bisa menggunakan ulang *source code* dari *superclass* disebut dengan istilah *reuse*.
- c. *Class-class* yang didefinisikan dengan atribut dan method yang bersifat umum yang berlaku baik pada *superclass* maupun *subclass* disebut dengan *abstract class*.

3. Polymorphism (polimorfisme–perbedaan bentuk)

Polimorfisme artinya penyamaran suatu bentuk dapat memiliki lebih dari satu bentuk.

2.3.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). (Fowler Martin.2004) [2]

Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi object. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson dibawah bendera Rational Software Corp. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. UML di deskripsikan oleh beberapa diagram, diantaranya:

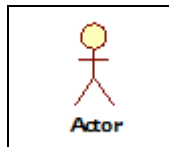
1.UseCase Diagram

UseCase Diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *usecase diagram*

lebih dititik beratkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *usecase diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Komponen-komponen yang terlibat dalam *usecase diagram*:

a. Aktor

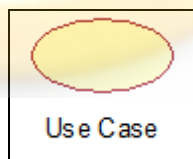


Gambar 2.4 Contoh Actor

Pada dasarnya aktor bukanlah bagian dari *usecase diagram*, namun untuk dapat terciptanya suatu *usecase diagram* diperlukan aktor, dimana aktor tersebut merepresentasi seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *usecase*, tetapi tidak memiliki kontrol atas *usecase*. Aktor digambarkan dengan *stickman*.

b. UseCase

Gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun.



Gambar 2.5 Contoh UseCase

Ada beberapa relasi yang terdapat pada *usecase diagram*:

1. *Association*, menghubungkan link antar element.
2. *Generalization*, disebut juga pewarisan (*inheritance*), sebuah element dapat merupakan spesialisasi dari elemen lainnya.
3. *Dependency*, sebuah element bergantung dalam beberapa cara ke element lainnya.

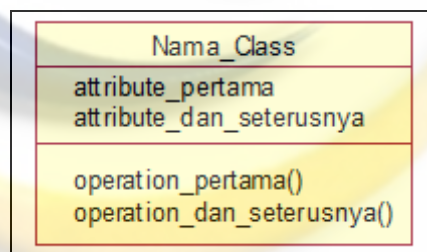
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi element lainnya.

Tipe relasi yang mungkin terjadi pada *usecase diagram*:

1. <<*include*>>, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *usecase* adalah bagian dari *usecase* lainnya.
2. <<*extends*>>, yaitu kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. <<*communicates*>>, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *usecase*.

c. *Class Diagram*

Class diagram adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *Class*, *Package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain.



Gambar 2.6 Contoh *Class*

Class memiliki tiga area pokok:

1. Nama (*Class Name*)
2. Atribut
3. Metode (*Operations*)

Pada UML, *class* di gambarkan dengan segi empat yang di bagi beberapa bagian. Bagian atas merupakan nama dari *class*. Bagian yang tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (metode/operasi).

Atribut dan metode dapat memiliki salah satu sifat berikut:

1. *Private*, tidak dapat di panggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan *class* lain yang mewarisinya.
3. *Public*, dapat dipanggil oleh *class* lain.

Hubungan antar *class*:

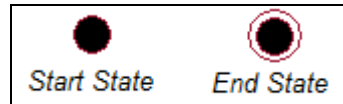
Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.

1. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
2. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru. Sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.
3. Hubungan dinamis, yaitu rangkaian pesan (*message*) *class* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

d. *Statechart Diagram*

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. *Statechart diagram* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

State adalah sebuah kondisi selama kehidupan sebuah objek atau ketika objek memenuhi beberapa kondisi, melakukan beberapa aksi atau menunggu sebuah *event*. *State* dari sebuah objek dapat di karakteristikkan oleh nilai dari satu atau lebih atribut-atribut dari *class*. *State* dari sebuah objek ditemukan dengan pengujian atau pemeriksaan pada atribut dan hubungan dari objek. Notasi UML untuk *state* adalah persegi panjang atau bujur sangkar dengan ujung yang dibulatkan.



Gambar 2.7 Start dan EndState

Masing-masing diagram harus mempunyai satu dan hanya satu *startstate* ketika objek mulai dibuat. Sebuah objek boleh mempunyai banyak *stopstate*.

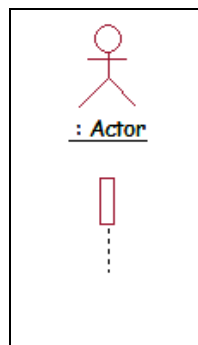


Gambar 2.8 Transisi dari State_1 ke State_2 yang diwakili tanda panah

Sebuah *statetransition* dapat mempunyai sebuah aksidan atau sebuah kondisi penjaga (*guardcondition*) yang terasosiasi dengannya, dan mungkin juga memunculkan sebuah *event*. Sebuah aksidan adalah kelakuan yang terjadi ketika *statetransition* terjadi. Sebuah *event* adalah pesan yang dikirim ke objek lain di sistem. Kondisi penjaga adalah ekspresi *boolean* (pilihanYa atau Tidak)dari nilaiatribut-atribut yang memungkinkan sebuah *statetransition* hanya jika kondisinya benar. Ke dua aksi dan penjaga adalah kelakuan dari objek dan secara tipikal menjadi operasi.

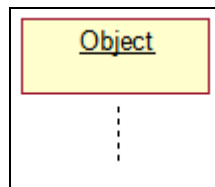
e. *Sequence Diagram*

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Di bawah merupakan simbol yang digunakan pada *sequence diagram*:



Gambar 2.9 Actor pada Sequence diagram

Actor adalah pesan dari seseorang atau sistem lain yang bertukar informasi dengan sistem yang lainnya, kemudian *lifeline* berhenti atau mulai pada titik yang tepat.



Gambar 2.10 Contoh *ObjectLifeline*

Objectlifeline menunjukkan keberadaan dari sebuah objek terhadap waktu. Yaitu objek dibuat atau dihilangkan selama suatu periode waktu diagram ditampilkan, kemudian *lifeline* berhenti atau mulai pada titik yang tepat.



Gambar 2.11 Contoh *Activation*

Activation menampilkan periode waktu selama sebuah objek atau aktor melakukan aksi. Dalam *object lifeline*, *activation* berada diatas *lifeline* dalam bentuk kotak persegi panjang, bagian atas dari kotak merupakan inisialisasi waktu dimulainya suatu kegiatan dan yang dibawah merupakan akhir dari waktu.



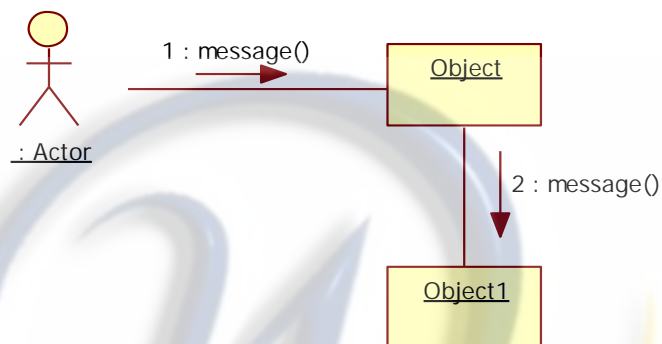
Gambar 2.12 Contoh *Message*

Message adalah komunikasi antar objek yang membawa informasi dan hasil pada sebuah aksi. *Message* menyampaikan dari *lifeline* sebuah objek kepada *lifeline* yang lain, kecuali pada kasus sebuah *message* dari objek kepada objek itu sendiri, atau dengan kata lain *message* dimulai dan berakhir pada *lifeline* yang sama.

f. Collaboration Diagram

Diagram ini menggambarkan interaksi objek yang diatur objek sekelilingnya dan hubungan antara setiap objek dengan objek yang lainnya. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagram*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

□

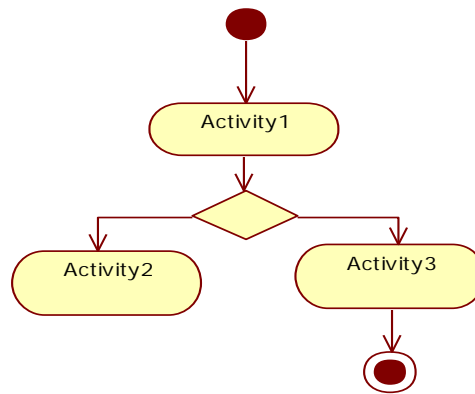


Gambar 2.13 Contoh Collaboration Diagram

a. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas keaktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *usecase*.

□



Gambar 2.14 Contoh Activity Diagram

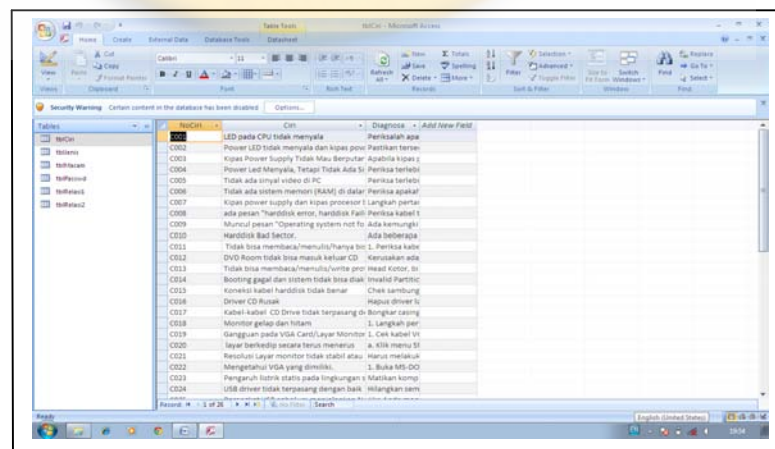
2.4 Tools untuk Pengembangan Perangkat Lunak

Adapun *software* yang digunakan untuk pembuatan Program Aplikasi Sistem Pakar yaitu dengan berbasis *Visual Basic 6.0*. dan *Microsoft Access* untuk pengolahan database-nya.

2.4.1 Microsoft Access

Microsoft Access adalah suatu program aplikasi basis data computer relasional yang digunakan untuk merancang, membuat dan mengolah berbagai jenis data dengan kapasitas yang besar.

Database adalah kumpulan tabel-tabel yang saling berelasi. Antar tabel yang satu dengan yang lain saling berelasi, sehingga sering disebut basis data relasional. Relasi antar tabel dihubungkan oleh suatu key, yaitu *primary key* dan *foreign key*.



Gambar 2.15 Komponen Utama (Object)

1. Table

Table adalah objek utama dalam database yang digunakan untuk menyimpan sekumpulan data sejenis dalam sebuah objek.

Table terdiri atas :

- a. Field Name : atribut dari sebuah table yang menempati bagian kolom.
- b. Record : Isi dari field atau atribut yang saling berhubungan yang menempati bagian baris.

2. Query (*SQL / Structured Query Language*)

Query adalah bahasa untuk melakukan manipulasi terhadap database. Digunakan untuk menampilkan, mengubah, dan menganalisa sekumpulan data. Query dibedakan menjadi 2, yaitu :

- a. **DDL (*Data Definition Language*)** digunakan untuk membuat atau mendefinisikan obyek-obyek database seperti membuat tabel, relasi antar tabel dan sebagainya.
- b. **DML (*Data Manipulation Language*)** digunakan untuk manipulasi database, seperti : menambah, mengubah atau menghapus data serta mengambil informasi yang diperlukan dari database.

3. Form

Form digunakan untuk mengontrol proses masukan data (*input*), menampilkan data (*output*), memeriksa dan memperbaharui data.

4. Report

Form digunakan untuk menampilkan data yang sudah dirangkum dan mencetak data secara efektif.

Tipe Data

Field - field dalam sebuah tabel harus ditentukan tipe datanya. Ada beberapa tipe data dalam Access, yaitu :

1. Text

Text digunakan untuk field alfanumeric (misal : nama, alamat, kode pos, telp), sekitar 255 karakter tiap fieldnya

2. Memo

Memo dapat menampung 64000 karakter untuk tiap fieldnya, tapi tidak bisa diurutkan/diindeks.

3. Number

Number digunakan untuk menyimpan data numeric yang akan digunakan untuk proses perhitungan matematis.

4. Date/Time

5. Currency

6. Auto Number

7. Yes/No

8. OLE Object

OLE Object digunakan untuk eksternal objek, seperti bitmap atau file suara.

9. Hyperlink

10. Lookup Wizard

Jika menggunakan tipe data ini untuk sebuah field, maka bisa memilih sebuah nilai dari tabel lain atau dari sebuah daftar nilai yang ditampilkan dalam combo box.[5]

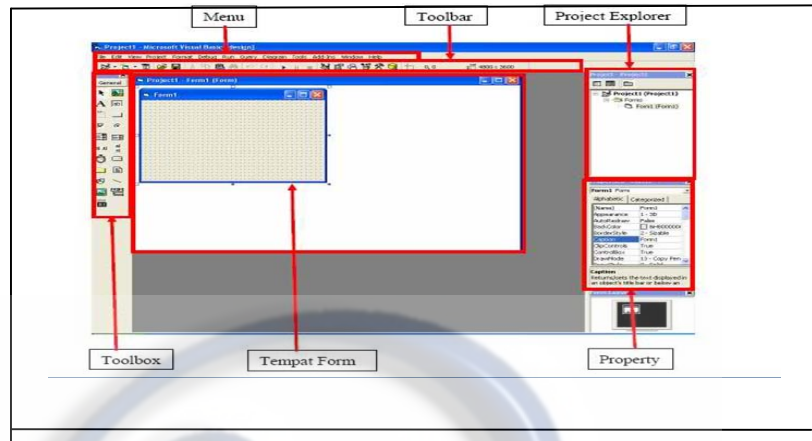
2.4.2 Microsoft Visual Basic

Basic adalah salah satu *development tools* untuk membangun aplikasi dalam lingkungan *Windows*. Dalam pengembangan aplikasi, *Visual Basic* menggunakan pendekatan *Visual* untuk merancang *user interface* dalam bentuk *form*, sedangkan untuk kodingnya menggunakan dialek bahasa *Basic* yang cenderung mudah dipelajari. *Visual Basic* telah menjadi *tools* yang terkenal bagi para pemula maupun para *developer*. *Visual Basic* adalah bahasa pemrograman berbasis *Microsoft Windows* yang merupakan *Object Oriented Programming* (OOP), yaitu pemrograman berorientasi objek, *Visual Basic* menyediakan objek-objek yang sangat kuat, berguna dan mudah.

Dalam lingkungan *Windows*, *User-interface* sangat memegang peranan penting, karena dalam pemakaian aplikasi yang kita buat, pemakai senantiasa berinteraksi dengan *User-interface* tanpa menyadari bahwa dibelakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan.

Pada pemrograman *Visual*, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur properti dari objek-objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program

untuk menangani kejadian-kejadian (*event*). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *Bottom Up*. [4]



Gambar 2.16 Komponen VB

Ada beberapa hal yang harus dipahami dalam mempelajari Visual Basic :

1. Objek

Sering disebut *entity* adalah sesuatu yang bisa dibedakan dengan lainnya. Pada dasarnya seluruh benda di dunia bisa dikatakan sebagai objek, contoh : mobil, komputer, radio, dan lain-lain.

Dalam *Visual Basic* objek-objek yang dimaksud disebut kontrol. Jenis-jenis kontrol antara lain : *Label*, *Text Box*, *Combo Box*, *List Box*, dan masih banyak lagi.

2. Properti

Sering disebut atribut, adalah ciri-ciri yang menggambarkan suatu objek. Misalnya disebut objek mobil jika mempunyai ban, spion, rem, dan lain-lain.

3. *Event*

Suatu kejadian yang menimpa objek. Bagaimana jika mobil didorong, ditabrak, dicat dan sebagainya.

4. Metode

Kemampuan yang dimiliki oleh suatu objek. Contohnya jika mobil berbelok, mundur, maju.

BAB III

ANALISIS SISTEM

3.1 Sistem Lama

Pada saat ini, beberapa teknisi mengalami kesulitan dalam hal mendiagnosa kerusakan komputer, dikarenakan dalam mengatasi permasalahan pada komputer teknisi masih membutuhkan waktu yang tidak cukup lama untuk mendiagnosa kerusakan komputer karena, pengetahuannya yang masih kurang. Teknisi sering sekali mengalami kesulitan dalam mendapatkan satu jenis atau ciri kerusakan komputer, sehingga sering sekali teknisi menunda pekerjaannya dan perbaikanpun akan terhambat. Teknisi membutuhkan pengetahuan untuk mengatasi masalah yang ada pada komputer, baik dalam buku-buku pengetahuan atau pedoman masalah pada komputer maupun dari para teknisi yang sudah berpengalaman.

3.2 Sistem Yang Akan Dibuat

1. Sistem sudah komputerisasi.
2. Memudahkan para teknisi komputer khususnya para pemula dalam mencari solusi tentang permasalahan yang terjadi pada komputer.
3. Dapat memberikan respon (jawaban) yang cepat terhadap permasalahan komputer.
4. Sistem akan memberikan sebuah basis pengetahuan dimana meliputi macam kerusakan, jenis kerusakan, dan ciri kerusakan.
5. Setelah mengetahui kerusakan komputer maka sistem akan memberikan solusi dalam pemecahan masalah sistem dan dilakukan secara bertahap.
6. Pada sistem juga akan memberikan informasi tentang macam kerusakan dan gejala-gejala yang terjadi dan berikut dengan solusi atau diagnosa kerusakan.

3.3 Identifikasi Masalah

Langkah pertama dalam mengembangkan aplikasi adalah mengidentifikasi masalah yang akan dikaji. Dalam hal ini adalah dengan mengidentifikasi permasalahan yang akan dibuat terlebih dahulu, adapun masalah-masalah yang akan diambil dalam aplikasi untuk mendeteksi kerusakan pada komputer serta cara mengatasinya.

Jenis kerusakan dan gejala adalah sebagai berikut:

1. Power Supply

a. PC sering restart

Masalah: Motherboard rusak

Solusi: Ganti motherboard dengan yang baru.

Masalah: Power supply rusak

Solusi: Lepas kabel yang terhubung dengan motherboard.

Masalah: Memori Kotor

Solusi: Cabut memori bersihkan memori dengan koas lalu pasang dengan slot yang lain

Masalah: Memori tidak terdeteksi

Solusi: Cek harddisk pada run program pada sistem operasi.

b. PC mati

Masalah: Tidak ada aliran listrik

Solusi: Cek kabel power supply atau catuan listrik apakah sudah terhubung dengan listrik utama.

Masalah: Tegangan listrik tidak stabil

Solusi: Gunakan stabilizier atau UPS, atau pindahkan ke stop kontak yang lain tanpa menggunakan terminal.

Masalah: PC sering mati tiba-tiba

Solusi: Cek kipas power supply apakah berputar dengan yang baik jika tidak maka ganti power supply dengan yang baru.

Masalah: Suhu PC panas

Solusi: Cek tegangan pada power supply apakah sudah sesuai dengan kebutuhan motherboard.

2. MotherBoard

a. Motherboard mengalami masalah.

Masalah: Semua perangkat tidak terdeteksi sama sekali(sama sekali tidak ada system operasi).

Solusi: Ganti motherboard dengan yang baru.

Masalah: Kipas motherboard tidak berjalan.

Solusi: Cek kabel soket kipas pada motherboard, cabut lalu masukan kembali dan nyalakan PC.

b. Komponen yang terhubung dengan motherboard bermasalah.

Masalah: Bunyi beep 3 kali selang 3 detik dan bunyi lagi.

Solusi: Cek memori mungkin tidak pas pada motherboard atau tidak terpasang lalu pasang dengan baik.

Masalah: Bunyi beep panjang.

Solusi: Matikan PC lalu diamkan beberapa menit agar tidak merambah pada kerusakan yang lain.

3. Harddisk

a. Cek pada setup BIOS

Masalah: Harddisk tidak terdeteksi pada saat proses booting

Solusi: Cek pada setup BIOS, apakah masih dapat mendeteksi hardisk. Pada beberapa motherboard, setup BIOS dapat ditampilkan dengan menekan tombol **DEL** pada saat booting. Setelah BIOS muncul, pilih opsi **Auto Detect Disk Drive** jika ada. Jika tidak ada, cek konfigurasi utama pasti ada prosedur untuk mengenali harddisk.

b. Cek sambungan kabel harddisk

Masalah: Koneksi kabel harddisk tidak benar

Solusi: Cek sambungan kabel harddisk yang ada di dalam casing jika ada kabel harddisk yang longgar/tidak menancap dengan benar, betulkan sampai benar-benar menancap.

c. Chek setting jumper Harddisk

Masalah: Setting Jumper CD Drive salah

Solusi: Perhatikan setting jumper yang harus dipasang. Lihat pada buku manual.

d. Partisi ulang harddisk

Masalah: Pada saat proses booting muncul pesan kesalahan “*Invalid Partition Table*”. Setelah itu booting gagal dan sistem tidak bisa diaktifkan

Solusi: *Invalid Partition Table*, berarti bahwa hardisk telah kehilangan partisinya. Yang harus dilakukan adalah dengan mempartisi ulang harddisk. Untuk mempartisi ulang harddisk lakukan langkah-langkah sebagai berikut:

1. Lakukan booting dengan disket start up. Pastikan bahwa di dalam disket tersebut berisi file sistem yang dilengkapi dengan file-file perintah, seperti: format.com dan fdisk.exe
2. Setelah disket start up dimasukan, tekan tombol power dan tunggu sampai muncul prompt A atau “A:\>”
3. Setelah prompt A muncul, jalankan program fdisk.exe dengan cara mengetikkan **fdisk** lalu enter
4. akan muncul tampilan sebagai berikut:

Fdisk Option

Current Fix disk drive

Choose one of the following

1. Create DOS Partition or Logical DOS Drive
 2. Set Active Partition
 3. Delete Partition
 4. Display Partition
 5. Change Current Fixed Disk Partition
5. Jika harddisk belum dipartisi sebelumnya maka pilihan Create DOS Partition or Logical DOS Drive. Kemudian dari sub menu yang muncul, pilih Create Primary DOS Partition. Akan muncul pertanyaan Anda akan menggunakan seluruh dari kapasitas harddisk untuk Primary atau ingin

membaginya. Jika Anda ingin membagi harddisk menjadi dua drive (Drive C dan Drive D), maka jawab pertanyaan tersebut dengan N

6. Masukan ukuran *Primary DOS partition* dalam ukuran MB atau prosentase. Nantinya partisi yang ini akan menjadi Drive C dan sisanya akan menjadi Drive D
7. Buatlah partisi tersebut aktif agar bisa digunakan untuk booting komputer. Untuk mengaktifkannya, pilih angka “2” pada sub menu utama (**Set active partition**)
8. Langkah selanjutnyabuatlah partisi kedua. Pilih angka “1” pada menu utama dan pilih angka “2” pada sub menunya. Maka akan muncul pertanyaan yang menanyakan ukuran partisi. Masukan ukuran yang diinginkan dan tekan enter hingga muncul pesan “extended DOS partition was created”.
9. Proses harddisk partisi selesai dan restart kembali komputer.

e. Harddisk kehilangan system

Masalah: Pada saat booting muncul pesan kesalahan “*Error Loading Operating System*” dan “*Missing Operating System*”

Solusi: *Missing Operating System*, berarti harddisk kehilangan system untuk boot. Lakukan langkah memformat harddisk Anda dan mengisinya dengan system sebagai berikut:

1. Lakukan booting menggunakan disket sistem, tinggi sampai A promt-nya muncul. Selanjutnya ketikkan perintah pemformatan drive sebagai berikut:

Format C:/S (enter)

2. Setelah diketikan perintah tersebut muncul pesan sebagai berikut:

Warning: All Data on Non removable Disk

Drive C Will be Lost

Proceed with Format ? (Y/N)

3. Tekan “Y” jika Anda ingin melakukan pemformatan. Akan muncul proses format dan tunggu sampai 100 % complete. Setelah proses format selesai, maka akan muncul pesan:

Format Complete

System Transferred

Volume label (11 character, Enter for none) ?

4. Ketikan nama label yang diinginkan maksimal 11 karakter. Kemudian tekan enter dan proses format selesai.
5. Jika Anda mempartisi harddisk menjadi 2. untuk memformat Drive D, cukup dengan perintah: **Format D: (enter)**.

4. CD/DVD ROM

a. Cek pada setup BIOS

Masalah: CD/DVD ROM tidak terdeteksi pada saat proses booting

Solusi: Cek pada setup BIOS, apakah masih dapat mendeteksi CD/DVD ROM. Pada beberapa motherboard, setup BIOS dapat ditampilkan dengan menekan tombol **DEL** pada saat booting. Setelah BIOS muncul, pilih opsi **Auto Detect Disk Drive** jika ada. Jika tidak ada, cek konfigurasi utama pasti ada prosedur untuk mengenali CD/DVD ROM.

b. Update ulang Driver CD/DVD

Masalah: Driver CD/DVD Rusak

Solusi: Hapus driver lama untuk CD/DVD Drive dan menggantinya dengan driver yang baru dengan menggunakan **Device Manager**.

Langkah-langkah:

1. Klik **Start** pada taskbar Windows kemudian pilih **Setting** dan **Control Panel**
2. Double klik icon **System**
3. Pilih tabulasi **Device Manager**
4. Dari daftar device yang muncul pilih **CDROM**
5. Pilih tombol **Remove** untuk menghapus driver yang aktif.
6. Kemudian lakukan booting ulang komputer agar Windows mendeteksi kembali secara otomatis CD/DVD ROM yang ada.

c. Cek sambungan CD/DVD ROM dengan benar

Masalah: Kabel-kabel yang terhubung ke CD/DVD Drive tidak terpasang dengan benar

Solusi: Bongkar casing dan cek kabel-kabel yang terpasang pada CD/DVD Drive. Ada tiga buah kabel yang menancap pada CD/DVD Drive yaitu kabel

power, kabel data dan kabel audio. Cek juga tancapan kabel-kabel tersebut pada motherboard. Jangan sampai ada yang longgar dan tidak tertancap dengan benar.

- d. Cek setting jumper CD/DVD Drive

Masalah: Setting Jumper CD/DVD Drive salah

Solusi: Perhatikan setting jumper yang harus dipasang. Lihat pada buku manual.

5. Kerusakan VGA

- a. Monitor Blank.

Masalah: Pada nyalakan monitor, layar monitor gelap dan hitam

Solusi: 1. Langkah pertama, cek kabel VGA dan pastikan kabel VGA tersebut sudah masuk ke port VGA yang ada di casing komputer.

2. Cek indikator pada monitor, apakah powernya berjalan dengan normal atau tidak.

3. Coba booting ulang.

- b. Gangguan pada VGA Card/Layar Monitor.

Masalah: 1. Komputer menjadi macet atau hang ketika digunakan untuk bermain game 3D.

2. Tidak dapat digunakan untuk menjalankan permainan tertentu.

3. Windows tidak bisa digunakan dalam mode normal.

4. Ada titik-titik kecil di layar monitor.

Solusi: 1. Cek kabel VGA yang terhubung ke monitor dan ke casing komputer, apakah sudah menancap dengan benar atau belum, kalau belum masukkan kabel tersebut sesuai dengan kaki-kaki yang tersedia.

2. Cek konektor atau pin yang terdapat pada kabel VGA. Kalau ada yang patah atau putus, segera ganti yang baru.

3. Install kembali atau perbaharui driver dari VGA Card yang dipasang. Untuk mendapatkan driver VGA tersebut, bisa dengan men-download dari situs yang menyediakan driver dari VGA itu.

4. Cobalah untuk mendownload versi terbaru versi terakhir software Direct-X dari situs Microsoft.
5. Kunjungi situs-situs yang menangani berbagai permasalahan tentang permainan atau games.
6. Buka casing komputer, cabut VGA card dari slotnya (PC/AGP) dari motherboard, dan pasang kembali. Apabila pada waktu komputer dihidupkan, layar masih hitam dan belum muncul gambarnya, ganti VGA card dengan yang baru, karena dapat dipastikan VGA card ada mengalami kerusakan.

c. VGA atau animasi tidak berjalan dengan normal

Masalah: Pesan kesalahan pada layar monitor. Apakah selalu menemui pesan kesalahan ketika memainkan suatu permainan atau games?

Solusi: a. Klik menu Start >> pilih Help and Support. Di bawah perintah Pick a Help Topic, pilih “Fixing a problem”.

b. Pada kotak sebelah kiri, pilih “Games, sound, and video problems”.

c. Pada kotak sebelah kanan, select “Games and Multimedia Troubleshooter”.

d. Klik pilihan yang menggambarkan permasalahan yang dihadapi kemudian klik Next. Ulangi langkah-langkah tersebut sampai permasalahan bisa terpecahkan. Gunakan tombol Back untuk mengulangi langkah terakhir.

6. Masalah Pada USB

a. USB tidak terdeteksi

Masalah: Kebanyakan Masalah Umum Device / perangkat yang Anda gunakan termasuk dalam perangkat yang rentan terhadap perubahan dalam masalah listrik dan arus listrik statis. Salah satu perangkat yang sering mengalami ini adalah USB stick.

Solusi: Matikan komputer, cabut perangkat USB Anda mengikuti semua kabel listrik masuk ke CPU dan Monitor. Tunggu 20 sampai 30 menit kemudian hidupkan seperti biasa. Untuk sebagian besar kasus, metode ini

terbukti berhasil. Kadang-kadang port USB bisa juga longgar karena kegagalan deteksi hardware USB. Jika masih tidak terdeteksi, gunakan Device Manager untuk memindai untuk menemukan perangkat keras atau perangkat keras baru informasi perubahan. Untuk melakukannya, klik Start> Run> Ketik "devmgmt.msc" (tanpa tanda kutip) [OK].

- b. Kesalahan Memasang driver Ketika perangkat USB dipasang ke komputer.

Masalah: USB driver belum terpasang dengan baik dan itu mengakibatkan kerusakan fatal pada driver USB lainnya.

Solusi: Hilangkan semua perangkat yang tersembunyi di Device Manager dengan cara sebagai berikut:

1. Klik Start> Run Ketik> CMD [OK],
2. Ketik "set DEVMGR_SHOW_DETAILS = 1?" (Tanpa tanda kutip) dan diakhiri dengan ENTER.
3. Ketik "set DEVMGR_SHOW_NONPRESENT_DEVICES = 1?" (Tanpa tanda kutip) dan akhiri dengan ENTER.
4. Ketik "devmgmt.msc mulai" (tanpa tanda kutip) dan akhiri dengan ENTER.
5. Klik menu View> Show Hidden Devices.
6. Pada panel kiri, klik tombol "" untuk melihat isi perangkat imaging, Unknown Perangkat dan USB Devices.
7. Apakah Anda menemukan Devices USB dan Unknown Devices (termasuk berwarna abu-abu)? Jika demikian, Klik kanan> Uninstall.

3.4 Kaidah

Kaidah produksi biasanya dituliskan dalam bentuk jika-maka (IF-THEN). Kaidah ini dapat dikatakan sebagai hubungan implikasi dua bagian yaitu bagian premise (jika) dan bagian konklusi (maka). Apabila bagian premise dipenuhi maka bagian konklusi juga akan bernilai benar. Sebuah kaidah- terdiri dari klausa-klausa. Sebuah klausa mirip sebuah kalimat subyek, kata kerja dan obyek yang menyatakan suatu fakta. Ada sebuah klausa premise dan klausa konklusi pada suatu kaidah. Suatu kaidah juga dapat terdiri atas beberapa premis dan lebih dari satu konklusi. Antara premise dan konklusi dapat berhubungan dengan “OR” atau “AND”. Berikut kaidah-kaidah produksi dalam menganalisis kerusakan terhadap komponen komputer yang gagal dalam menjalankan fungsinya:

RULE Masalah pada power supply

IF Ada kerusakan dengan piranti komputer Is Ya
AND Ada masalah pada power supply Is Ya
AND Masalah pada motherboard rusak Is Ya
AND Power supply rusak Is Ya
AND Memori kotor Is Tidak
AND Apakah memori tidak terdeteksi Is Ya
THEN Masalah pada power supply

RULE Power supply pada aliran listrik

IF Ada kerusakan pada piranti komputer Is Ya
AND Ada masalah pada power supply Is Ya
AND Masalah pada aliran listrik Is Ya
AND Tegangan listrik tidak stabil Is Tidak
THEN Power supply pada aliran listrik

RULE PC sering mati

IF Ada kerusakan pada piranti komputer Is Ya

AND Ada masalah pada power supply Is Ya

AND PC mati total Is Tidak

AND Power supply sudah sesuai dengan kebutuhan motherboard Is Ya

THEN PC sering mati.

RULE Motherboard

IF Ada kerusakan dengan piranti komputer Is Ya

AND Semua perangkat tidak terdeteksi (sama sekali tidak ada system operasi) Is Ya

AND Kipas motherboard tidak jalan Is Ya

THEN Cek kabel soket kipas pada motherboard, cabut lalu masukan kembali dan nyalakan PC

RULE Komponen yang terhubung dengan motherboard

IF Ada kerusakan dengan piranti komputer Is Ya

AND Bunyi beep 3 kali selang 3 detik dan bunyi lagi Is Ya

AND Bunyi beep panjang Is Ya

THEN Komponen yang terhubung dengan motherboard

RULE Cek pada setup BIOS

IF Ada kerusakan dengan piranti komputer Is Ya

AND Ada masalah pada harddisk Is Ya

AND Harddisk terdeteksi pada saat proses booting Is Tidak

THEN Cek pada setup BIOS

RULE Cek sambungan kabel harddisk

IF Ada kerusakan dengan piranti komputer Is Ya

AND Ada masalah pada harddisk Is Ya

AND Harddisk terdeteksi pada saat proses booting Is Ya

AND Kabel harddisk sudah terkoneksi dengan benar Is Tidak

THEN Chek sambungan kabel harddisk

RULE Chek setting jumper Harddisk dengan benar

IF Ada kerusakan dengan piranti komputer Is Ya

AND Ada masalah pada harddisk Is Ya

AND Harddisk terdeteksi pada saat proses booting Is Ya

AND Kabel harddisk sudah terkoneksi dengan benar Is Ya

AND Setting jumpur harddisk sudah benar Is Tidak

THEN Chek setting jumper Harddisk dengan benar

RULE Partisi ulang harddisk

IF Ada kerusakan dengan piranti komputer Is Ya

AND Ada masalah pada harddisk Is Ya

AND Harddisk terdeteksi pada saat proses booting Is Ya

AND Kabel harddisk sudah terkoneksi dengan benar Is Ya

AND Pada saat booting muncul pesan kesalahan "*Invalid partition table*",
booting gagal dan sistem tidak bisa diaktifkan Is Ya

THEN Partisi ulang harddisk

RULE Harddisk kehilangan sistem

IF Ada kerusakan dengan piranti komputer Is Ya

AND Ada masalah pada harddisk Is Ya

AND Harddisk terdeteksi pada saat proses booting Is Ya

AND Kabel harddisk sudah terkoneksi dengan benar Is Ya

AND Pada saat booting muncul pesan kesalahan "*Invalid partition table*",
booting gagal dan sistem tidak bisa diaktifkan Is Tidak

AND Pada saat booting muncul pesan kesalahan "*Error Loading Operating System*" dan "*Missing Operating System*" Is Ya

THEN Harddisk kehilangan system

RULE Chek pada setup BIOS

IF Ada kerusakan dengan piranti komputer Is Ya
AND Ada masalah pada CD/DVD ROM Is Ya
AND CD/DVD ROM terdeteksi pada saat proses booting Is Tidak
THEN Chek pada setup BIOS

RULE Update ulang driver CD/DVD

IF Ada kerusakan dengan piranti komputer Is Ya
AND Ada masalah pada CD/DVD ROM Is Ya
AND Driver CD/DVD bagus Is Tidak
THEN Update ulang driver CD/DVD

RULE Chek sambungan kabel CD/DVD ROM

IF Ada kerusakan dengan piranti komputer Is Ya
AND Ada masalah pada CD/DVD ROM Is Ya
AND Driver CD/DVD bagus Is Ya
AND Koneksi kabel ke CD/DVD Drive sudah benar Is Tidak
THEN Chek sambungan kabel CD/DVD ROM

RULE Chek setting jumper CD/DVD dengan benar

IF Ada kerusakan dengan piranti komputer Is Ya
AND Ada masalah pada CD/DVD ROM Is Ya
AND Driver CD/DVD bagus Is Ya
AND Koneksi kabel ke CD/DVD Drive sudah benar Is Ya
AND Setting jumper CD/DVD Drive sudah benar Is Tidak
THEN Chek setting jumper CD/DVD dengan benar

RULE Masalah pada VGA

IF Ada kerusakan dengan piranti komputer Is Ya
AND Ada masalah pada VGA Is Ya
AND Monitor blank Is Ya

AND Nyalakan monitor Is Ya

AND Monitor berkedip-kedip Is Tidak

THEN Cek indicator pada monitor, apakah powernya berjalan dengan normal atau tidak

RULE Gangguan pada VGA Card/Layar monitor

IF Ada kerusakan dengan piranti komputer Is Ya

AND Komputer menjadi macet/hang ketika digunakan untuk bermain games 3D Is Ya

AND Ada titik-titik dilayar monitor Is Ya

AND Cek konektor/pin pada layar VGA Is Ya

THEN Gangguan pada VGA card/layar monitor.

RULE VGA atau animasi tidak berjalan dengan normal

IF Ada kerusakan dengan piranti komputer Is Ya

AND Ada masalah pada VGA animasi Is Ya

AND Menemui kesalahan ketika memainkan suatu permainan atau games Is Ya

THEN VGA atau animasi tidak berjalan normal

RULE USB tidak terdeteksi

IF Ada kerusakan dengan piranti komputer Is Ya

AND USB terdeteksi Is Tidak

AND Ada masalah dengan arus listrik Is Ya

AND Menggunakan USB stik Is Ya

THEN USB tidak terdeteksi

RULE Kesalahan memasang perangkat USB ke komputer

IF Ada kerusakan dengan piranti komputer Is Ya

AND USB sudah terpasang dengan baik Is Tidak

AND Perangkat USB terpasang pada komputer Is Ya

AND Driver USB akan mengakibatkan kerusakan Is Ya

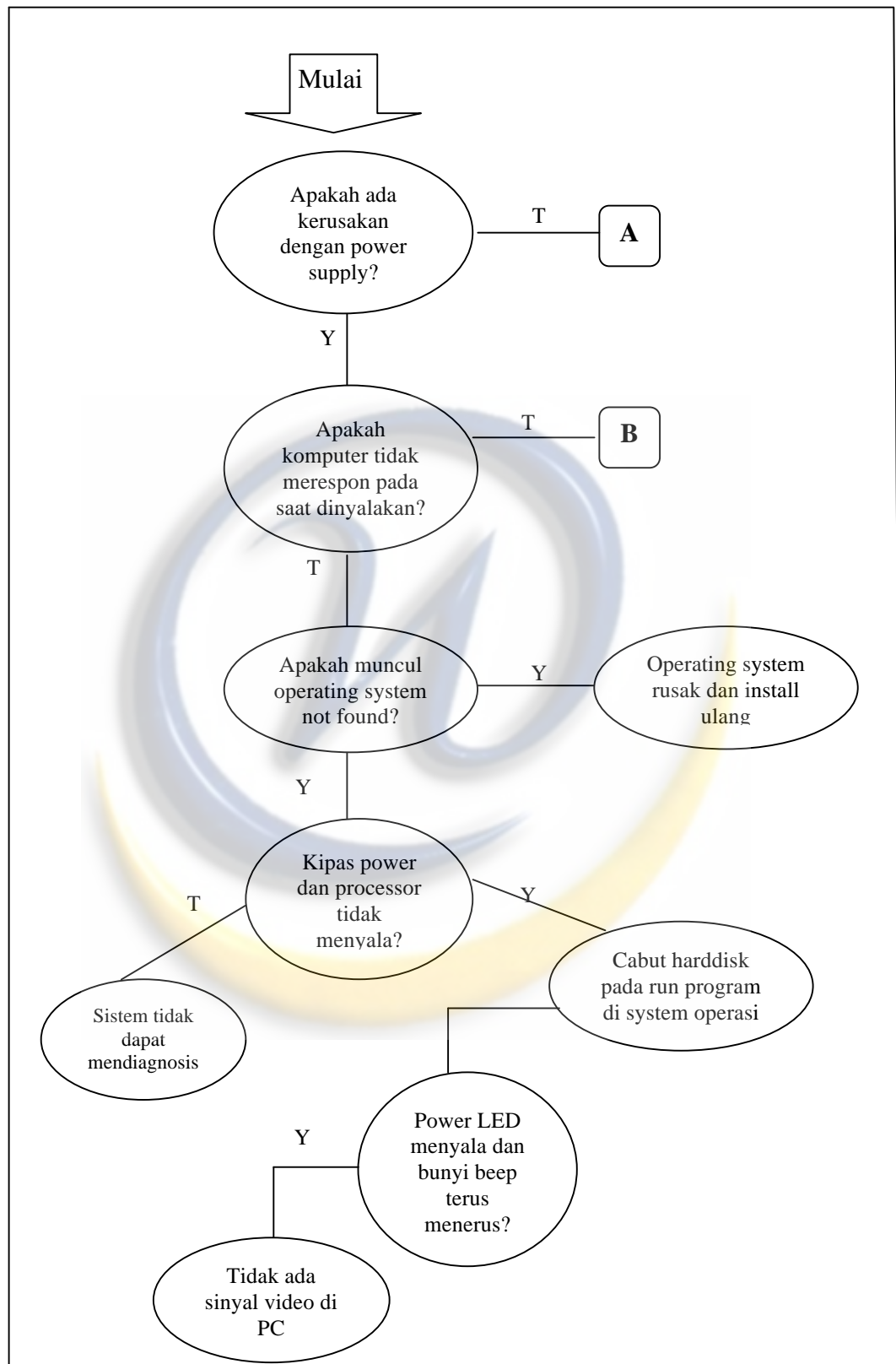
AND Semua perangkat yang tersembunyi di device manager Is Ya

THEN Kesalahan memasang perangkat USB ke komputer

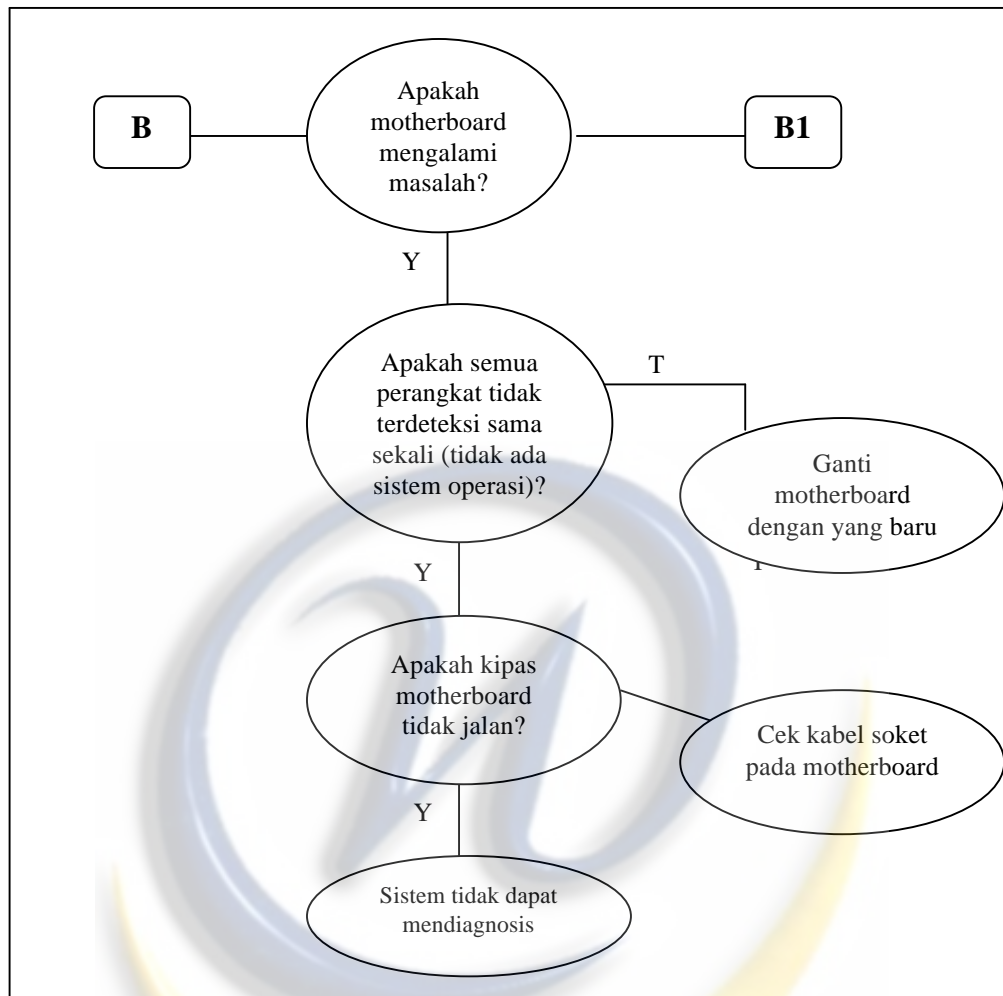
3.5 Pohon Keputusan

Pohon keputusan merupakan salah satu bentuk semantik, yaitu metoda untuk mempresentasikan pengetahuan yang berupa gambaran dari suatu pengetahuan yang memperlihatkan hubungan dari objek-objek. Objek tersebut dipresentasikan dalam bentuk node dan hubungan antar objek dinyatakan dengan garis penghubung.

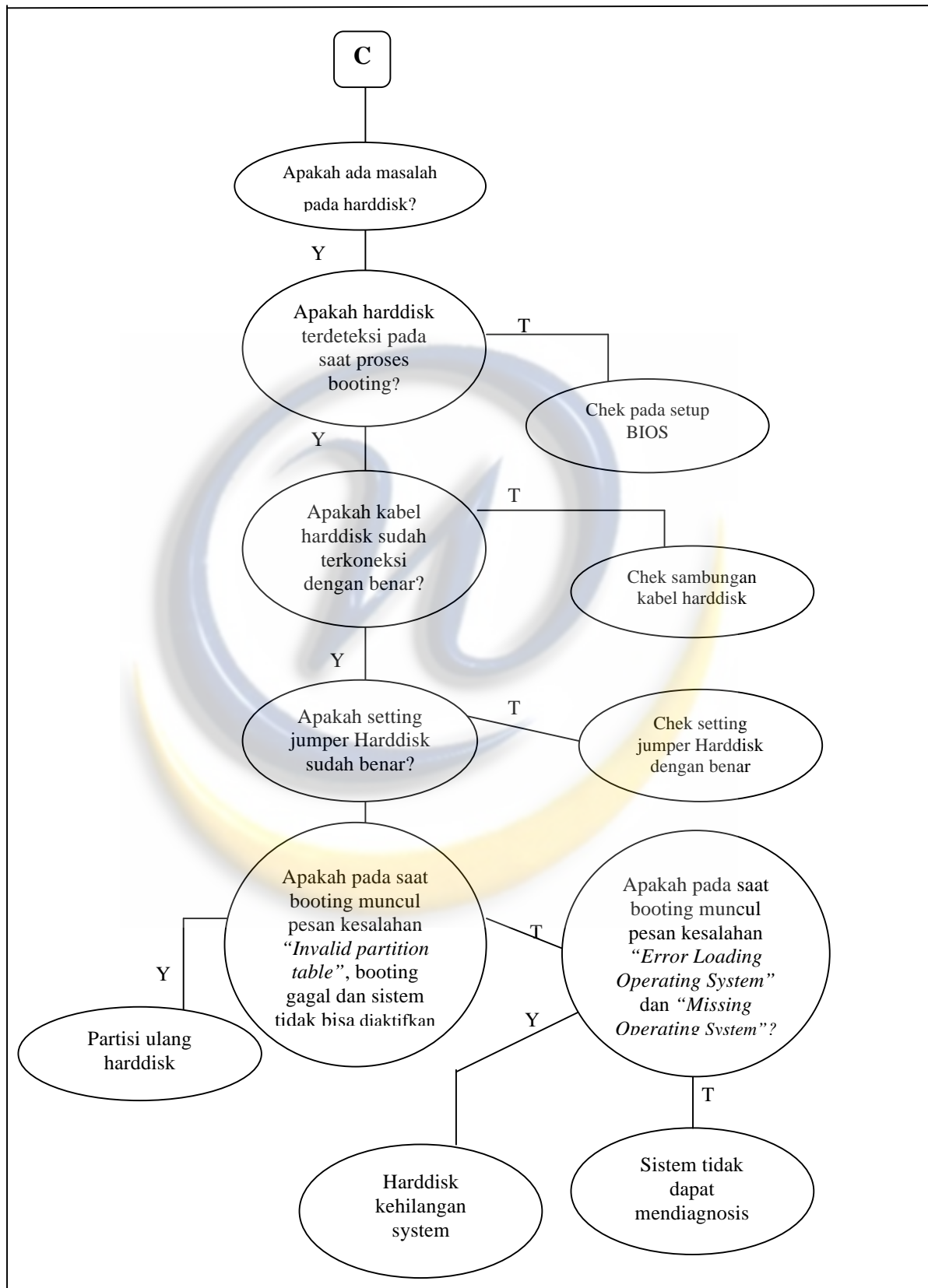




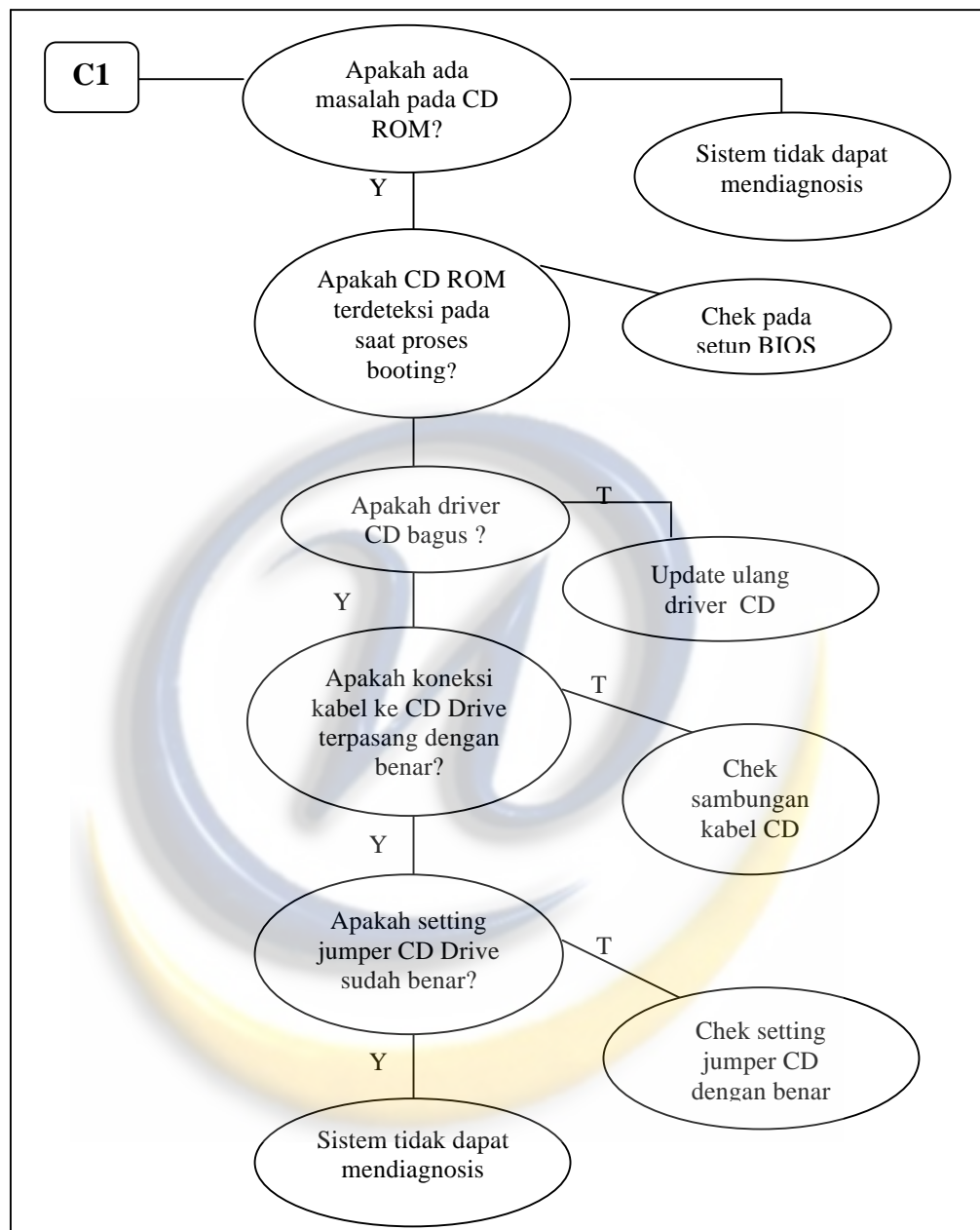
Gambar 3.1 Pohon keputusan kerusakan pada Power Supply



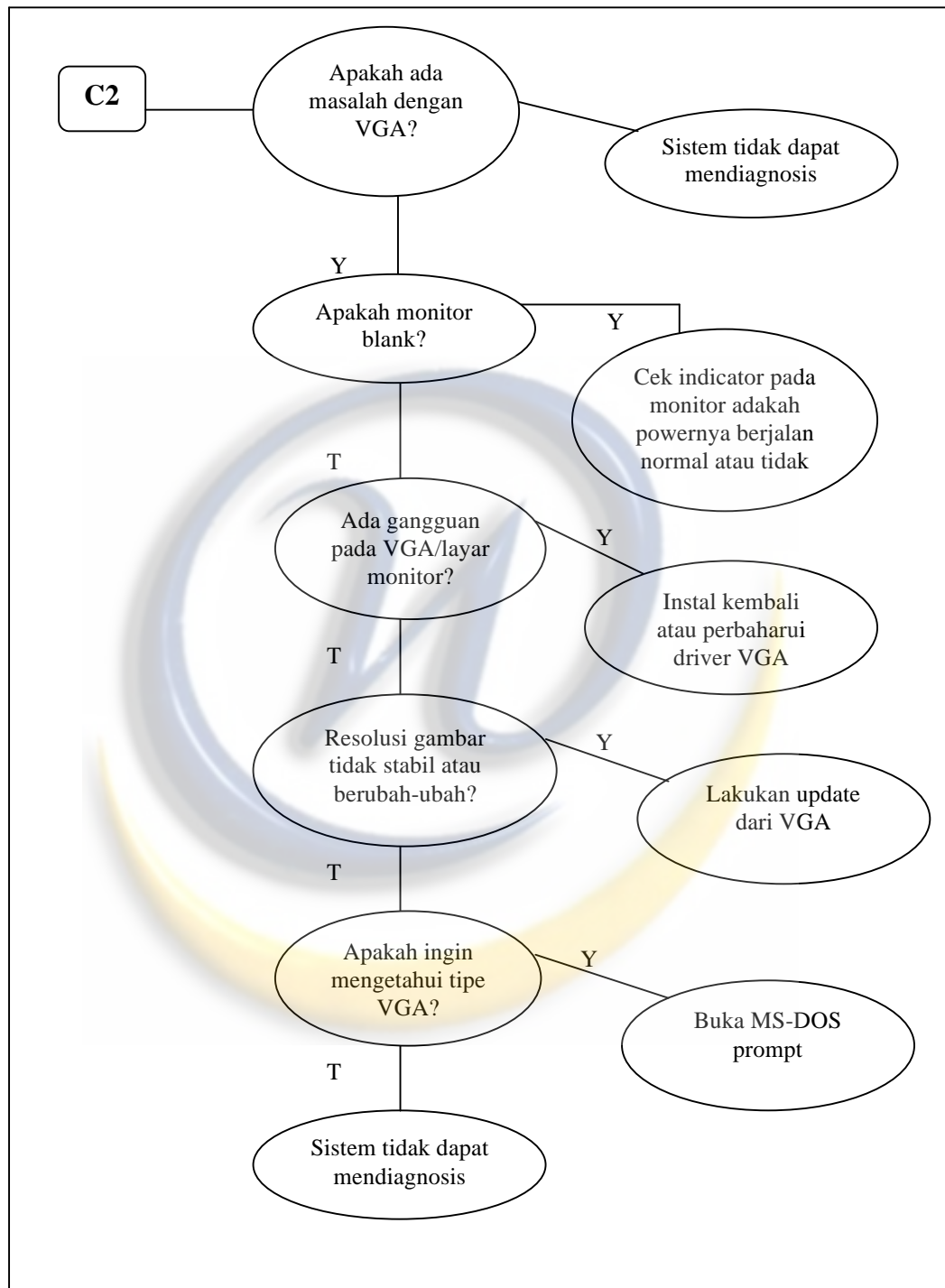
Gambar 3.2 Pohon Keputusan Kerusakan Pada Motherboard



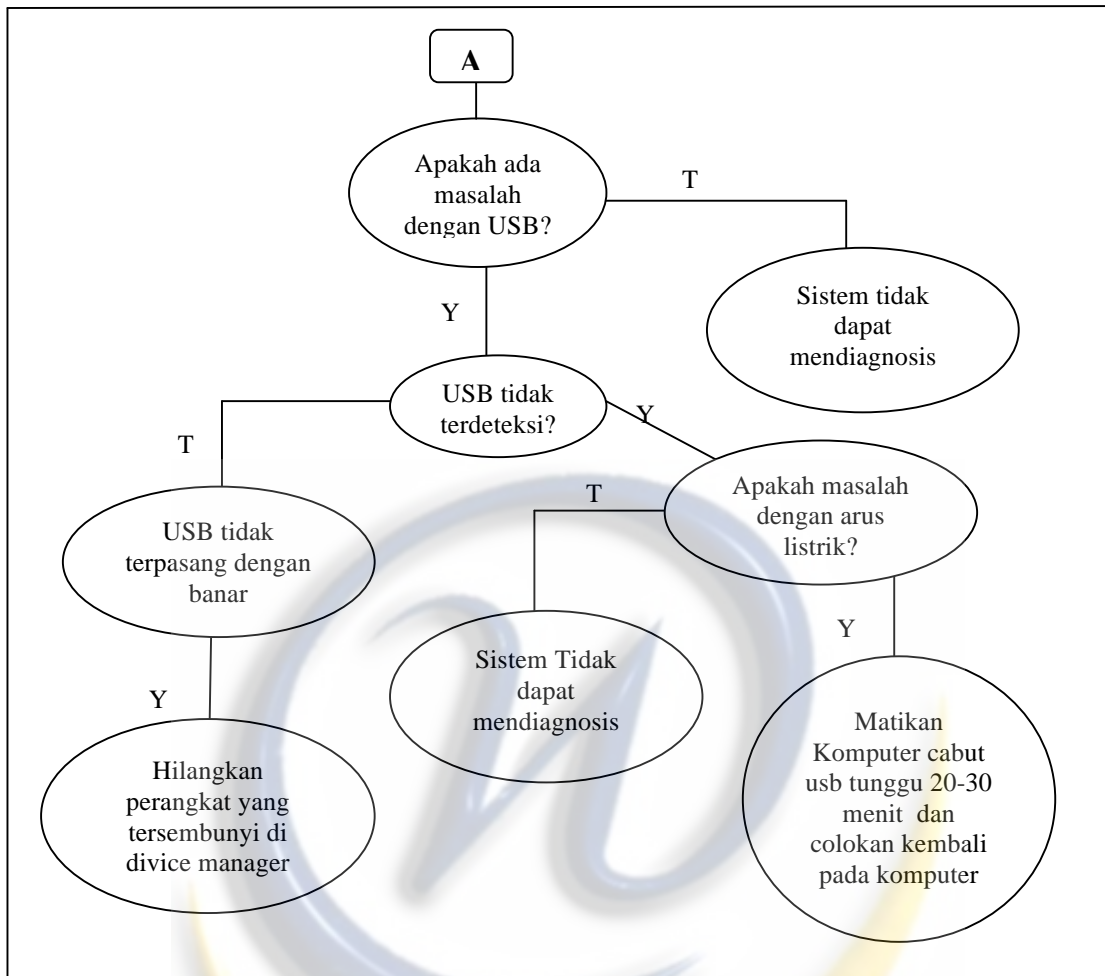
Gambar 3.3 Pohon keputusan kerusakan pada Harddisk



Gambar 3.4 Pohon keputusan kerusakan pada CD-Room



Gambar 3.5 Pohon Keputusan Kerusakan Pada VGA



Gambar 3.6 Pohon Keputusan Kerusakan Pada USB

3.6 Analisis Perangkat Keras

Perangkat keras minimum server yang direkomendasikan untuk menjalankan aplikasi sistem pakar ini adalah sebagai berikut :

Tabel 3.1
Spesifikasi PC

No	Uraian	Spesifikasi
1	Processor	Dengan kecepatan 1GHz
2	Memory (RAM)	700 MB
3	Hard Disk drive	80 GB
4	Monitor	14 “
5	Keyboard	Standar
6	Mouse	Standar

3.7 Analisis Perangkat Lunak

Dalam membangun aplikasi sistem pakar ini, perangkat lunak sangat berperan dalam kinerja aplikasi. Dan pada pembuatan aplikasi sistem pakar ini, ada syarat minimum bagi perangkat lunak yang digunakan agar kinerja aplikasi bisa lebih optimal. Kebutuhan perangkat lunak tersebut, adalah:

Tabel 3.2
Spesifikasi Software

No	Uraian	Spesifikasi
1	Sistem operasi <i>windows XP</i>	700 MB
2	<i>Visual Basic 6.0</i>	500 MB
3	<i>Microsoft Access 2007</i>	700 MB

BAB IV

PERANCANGAN SISTEM

4.1 Perancangan Sistem

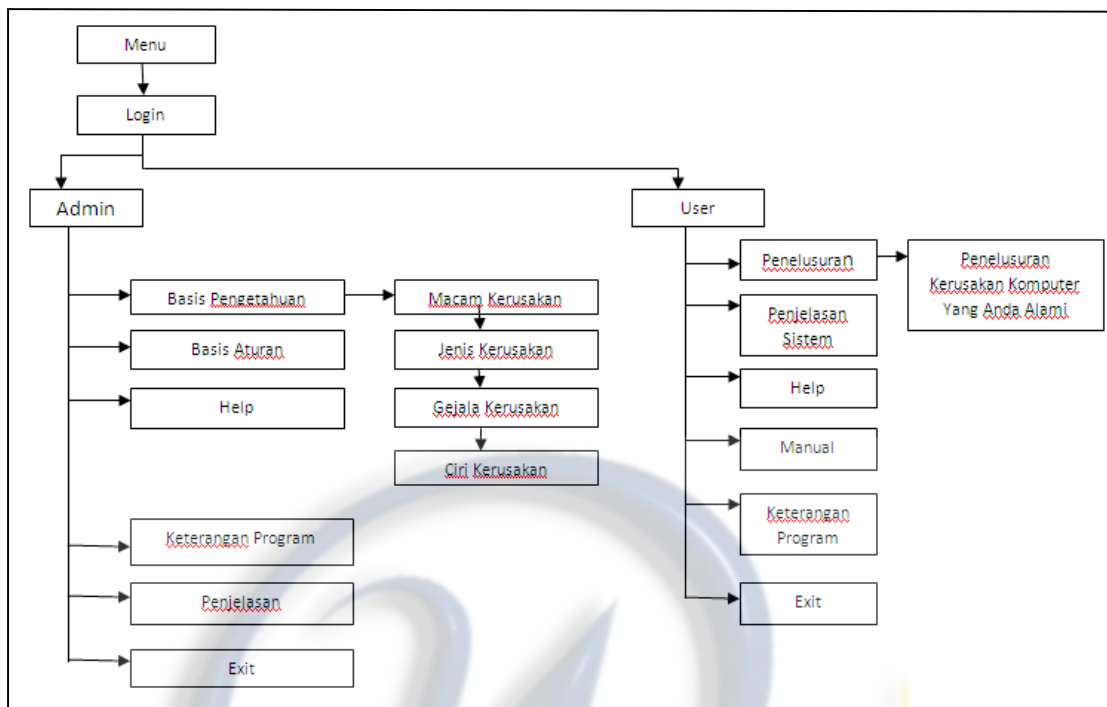
Perancangan aplikasi ini dibangun bertujuan untuk memudahkan pengguna dalam mendapatkan informasi mengenai kerusakan komputer yang dialaminya, serta mempermudah pengguna dalam mengetahui solusi dari masalah pengguna tersebut.

Perancangan Aplikasi Sistem pakar diagnosa kerusakan komputer yang dibangun ini bersifat *object oriented* (berorientasi objek) dengan menggunakan *Unified Modeling Language (UML)* sebagai bahasa pemodelan. Pembangunan Aplikasi Sistem pakar diagnosa kerusakan komputer ini dilakukan dengan menggunakan *tools* utama sebagai berikut :

1. VB6 sebagai *framework* aplikasi untuk membaca bahasa pemrograman *Visual Basic*.
2. *Microsoft Access* sebagai *Database Server*.
3. *IBM Rational Rose* sebagai *tools* untuk mengolah *UML*

4.2 Perancangan Menu

Berikut ini merupakan menu Aplikasi Sistem pakar diagnosa kerusakan komputer digambarkan sebagai berikut:



Gambar 4. 1 Perancangan Menu

4.2.1 Penjelasan Perancangan Menu

- Menu Login Admin : Admin Masuk pada menu sistem
- Macam Kerusakan : Admin menambahkan macam kerusakan komputer pada aplikasi Sistem Pakar
- Jenis Kerusakan : Admin menambahkan jenis kerusakan komputer pada aplikasi Sistem Pakar
- Ciri Kerusakan : Admin menambahkan ciri kerusakan komputer pada aplikasi Sistem Pakar
- Basis Aturan : Admin menambahkan pilihan Basis Pengetahuan pada aplikasi Sistem Pakar
- Keterangan Program : Keterangan Program pada aplikasi Sistem Pakar
- Penjelasan : Penjelasan untuk pemeliharaan komputer secara rutin. Informasi diberikan untuk user yang menggunakan aplikasi Sistem Pakar

h. Penelusuran : User menggunakan aplikasi Sistem Pakar untuk mengetahui solusi untuk kerusakan komputer.

1. Penelusuran berdasarkan Macam Kerusakan

User memilih macam kerusakan computer yang sedang dialami

2. Penelusuran berdasarkan Jenis Kerusakan

User memilih jenis kerusakan computer yang sedang dialami

3. Penelusuran berdasarkan Ciri kerusakan

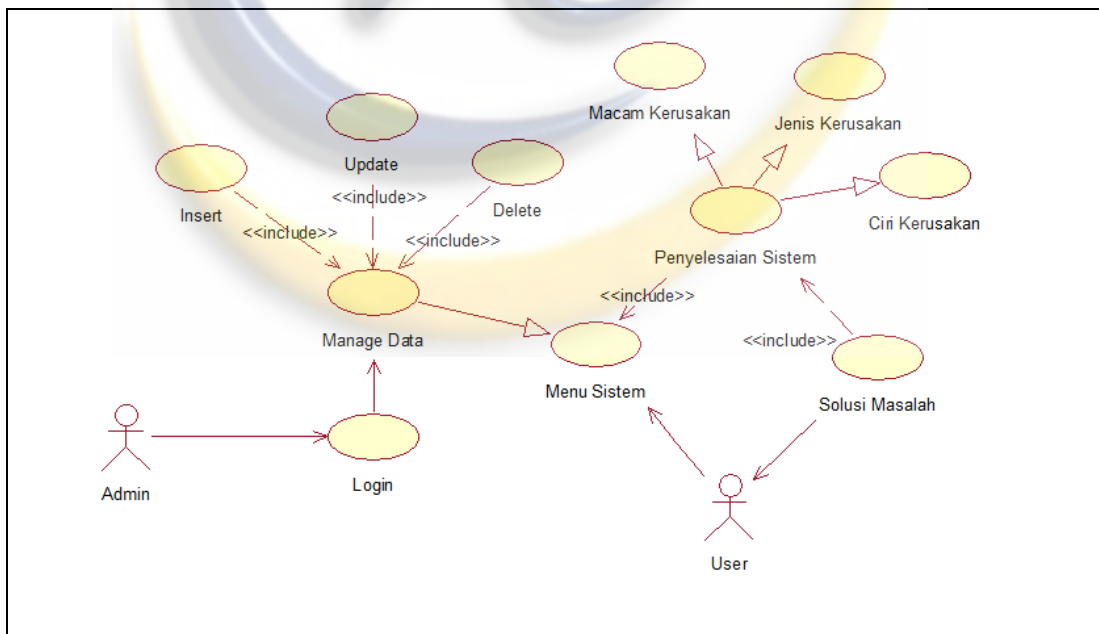
User memilih ciri kerusakan yang sedang dialami.

4.3 Model Use Case

Model *use case* menjelaskan mengenai aktor-aktor yang terlibat dengan perangkat lunak yang dibangun beserta proses-proses yang ada didalamnya.

4.3.1 Use Case Diagram

Diagram use case dari Aplikasi Sistem pakar diagnosa kerusakan komputer adalah sebagai berikut:



Gambar 4. 2 Use case Diagram

4.3.2 Definisi Aktor

Definisi aktor merupakan penjelasan dari apa yang dilakukan oleh aktor-aktor yang terlibat dalam perangkat lunak yang dibangun. Adapun deskripsi dari aktor-aktor yang terlibat dalam Aplikasi Sistem pakar diagnosa kerusakan komputer sebagai berikut:

Tabel 4. 1 Definisi Aktor

No	Aktor	Deskripsi
1	<i>Admin</i>	Melakukan tugas <i>login</i> dan memanipulasi (<i>Insert, Update, dan Delete</i>) semua isi content Aplikasi Sistem pakar diagnosa kerusakan komputer.
2	<i>User</i>	Mencari pemecahan masalah melalui sistem Aplikasi Sistem pakar diagnosa kerusakan computer

4.3.3 Definisi Use Case

Use case adalah urutan transaksi/proses yang dilakukan oleh sistem, di mana menghasilkan sesuatu yang dapat dilihat/diamati oleh *actor* tertentu. Deskripsi dari *use case* yang ada dalam Aplikasi Sistem pakar diagnosa kerusakan komputer adalah sebagai berikut :

Tabel 4. 2 Definisi Use Case

No	<i>Use case</i>	Deskripsi
1	Login	Validasi untuk <i>user</i> sistem sebelum masuk ke dalam sistem
2	Menu Sistem	Menu Dari Aplikasi Sistem pakar diagnosa kerusakan komputer untuk pengguna.

4.3.4 Skenario Use Case

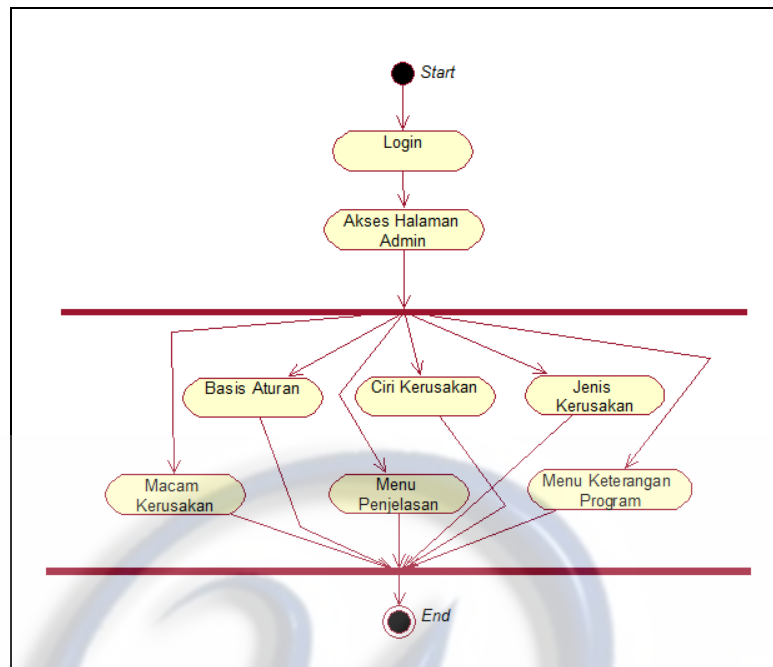
Skenario (*flow of event*) *use case* dari Aplikasi Sistem pakar diagnosa kerusakan komputer adalah sebagai berikut :

Tabel 4. 3 Skenario Use Case Aplikasi Sistem pakar diagnosa kerusakan komputer

No	Use Case	Deskripsi
1.	Login admin.	Melakukan tugas <i>login</i> dan memanipulasi (<i>Insert, Update, dan Delete</i>) semua isi content Aplikasi Sistem pakar diagnosa kerusakan komputer.

4.4 Activity Diagram

Activity diagram adalah salah satu cara untuk memodelkan event-event yang terjadi dalam suatu *use case*. Berikut ini activity diagram dari Aplikasi Sistem pakar diagnosa kerusakan komputer untuk User :



Gambar 4. 3 Activity Diagram Aplikasi Sistem Pakar

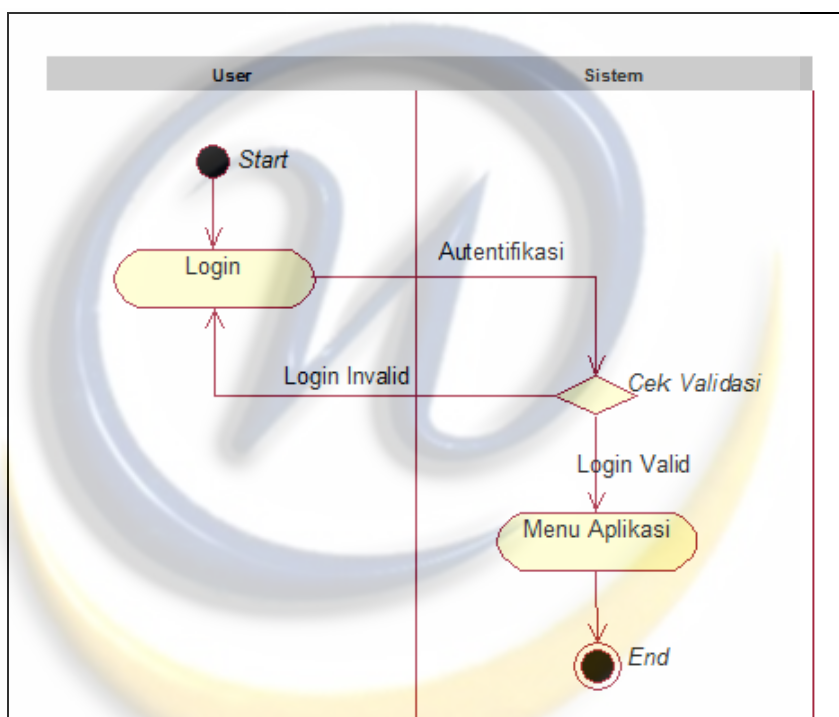
4.4.1 Skenario Use Case Login Admin

Tabel 4. 4 Skenario Use Case Login Admin

Name	<i>Login</i>
Level	Level Admin
Actors	<i>Admin</i>
Goal	Untuk menjaga keamanan data (security), serta untuk memasuki halaman administrator
Precondition	<i>Admin</i> belum valid dan belum berhasil masuk ke menu aplikasi sesuai level login
Postcondition	<i>Admin</i> telah <i>valid</i> untuk masuk menu aplikasi sesuai level login

Steps	<p><i>Admin</i> memasukkan <i>username</i> dan <i>password</i></p> <p><i>Admin</i> menekan tombol login</p> <p><i>Admin</i> telah berhasil login memasuki ke menu aplikasi sesuai level login.</p>
-------	--

4.4.2 Activity Diagram Login Admin



Gambar 4. 4 Activity Diagram Login Admin

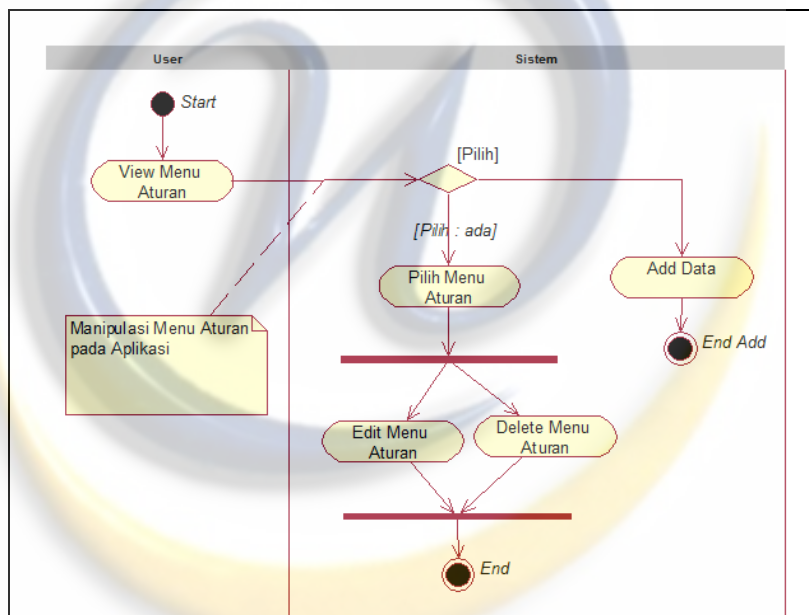
4.4.3 Skenario Use Case Admin Manage Menu Aturan

Tabel 4. 5 Skenario Use Case Admin Manage Menu Aturan

Name	Pengelolaan Menu Aturan pada aplikasi
Level	Level Admin
Actors	<i>Admin</i>
Goal	Untuk mengelola Menu Aturan pada Aplikasi

Precondition	<i>Admin</i> belum mengakses menu admin dan tidak bisa merubah menu Aturan pada Aplikasi
Postcondition	<i>Admin</i> dapat mengakses halaman Admin dan memanipulasi isi menu Aturan
Steps	<i>Admin</i> mengakses halaman Admin <i>Admin</i> memilih Menu Aturan <i>Admin</i> memanipulasi Menu Aturan

4.4.4 Activity Diagram Use Case Admin Manage Menu Aturan



Gambar 4. 5 Activity Diagram Use Case Admin Manage Menu Aturan

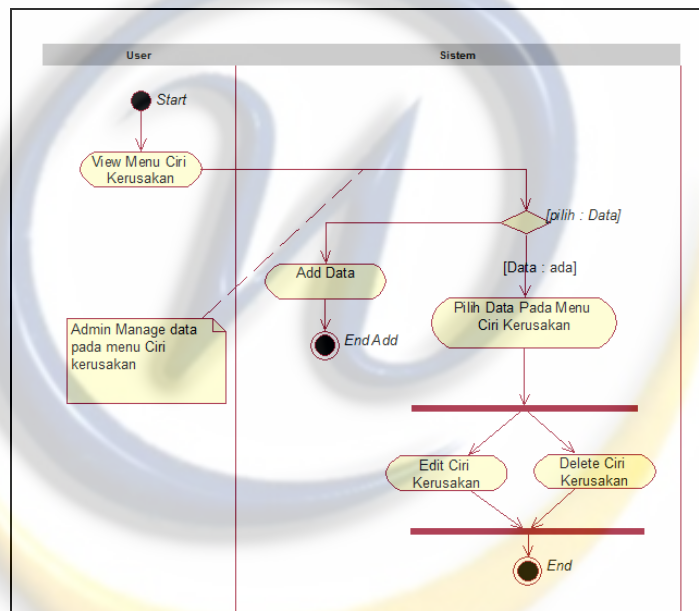
4.4.5 Skenario Use Case Admin Manage Menu Ciri Kerusakan

Tabel 4. 6 Skenario Use Case Admin Manage Ciri Kerusakan

Name	Admin mengelola menu ciri kerusakan pada aplikasi
Level	Level Admin
Actors	<i>Admin</i>

Goal	Untuk mengelola menu ciri kerusakan pada aplikasi
Precondition	<i>Admin</i> belum mengakses menu aplikasi dan tidak bisa merubah menu ciri kerusakan
Postcondition	<i>Admin</i> dapat menu aplikasi dan memanipulasi menu ciri kerusakan pada aplikasi
Steps	<i>Admin</i> mengakses menu aplikasi

4.4.6 Activity Diagram Admin Manage Menu Ciri Kerusakan



Gambar 4. 6 Activity Diagram Admin Manage Menu Ciri Kerusakan

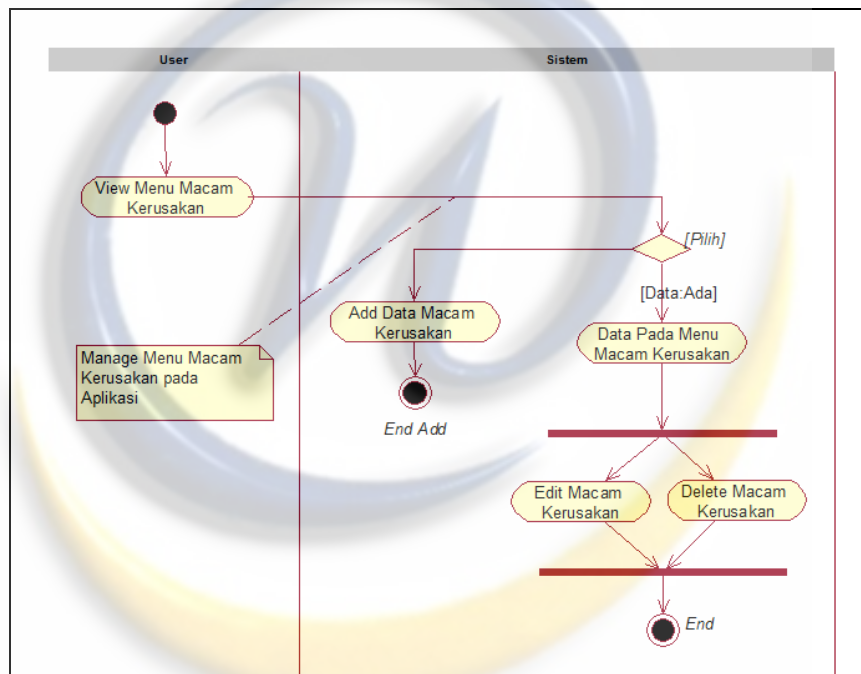
4.4.7 Skenario Use Case Admin Manage Menu Macam Kerusakan

Tabel 4. 7 Skenario Use Case Admin Manage Macam Kerusakan

Name	Admin mengelola menu macam kerusakan pada aplikasi
Level	Level Admin
Actors	<i>Admin</i>

Goal	Untuk mengelola menu macam kerusakan pada aplikasi
Precondition	<i>Admin</i> belum mengakses menu aplikasi dan tidak bisa merubah menu macam kerusakan
Postcondition	<i>Admin</i> dapat menu aplikasi dan memanipulasi menu macam kerusakan pada aplikasi
Steps	<i>Admin</i> mengakses menu aplikasi

4.4.8 Activity Diagram Admin Manage Menu Macam Kerusakan



Gambar 4. 7 Activity Diagram Admin Manage Menu Macam Kerusakan

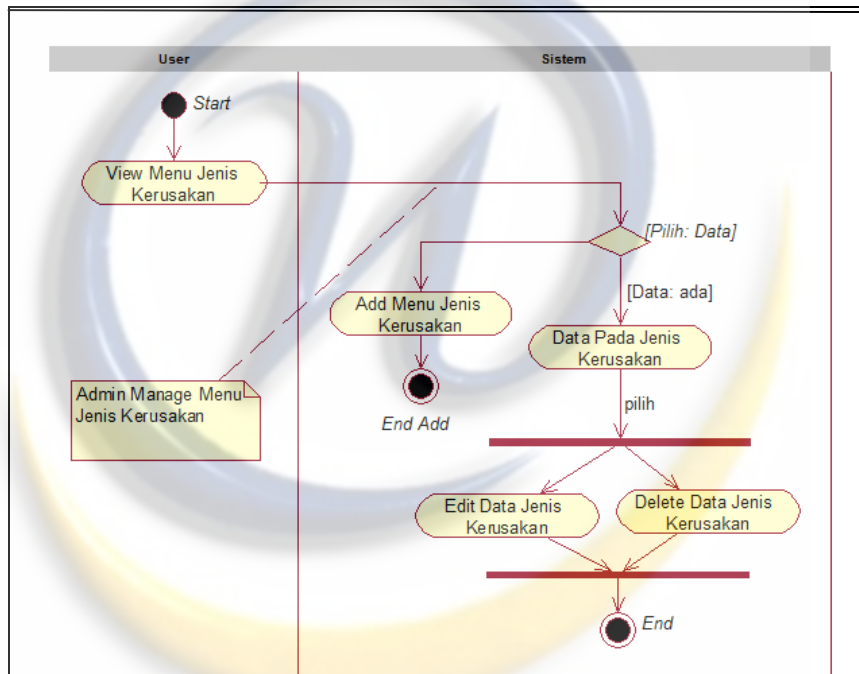
4.4.9 Skenario Use Case Admin Manage Menu Jenis Kerusakan

Tabel 4. 8 Skenario Use Case Admin Manage Menu Jenis Kerusakan

Name	Admin mengelola menu Jenis kerusakan pada aplikasi
Level	Level Admin
Actors	<i>Admin</i>

Goal	Untuk mengelola menu Jenis kerusakan pada aplikasi
Precondition	<i>Admin</i> belum mengakses menu aplikasi dan tidak bisa merubah menu Jenis kerusakan
Postcondition	<i>Admin</i> dapat menu aplikasi dan memanipulasi menu Jenis kerusakan pada aplikasi
Steps	<i>Admin</i> mengakses menu aplikasi

4.4.10 Activity Diagram Admin Manage Menu Jenis Kerusakan



Gambar 4. 8 Activity Diagram Admin Manage Menu Jenis Kerusakan

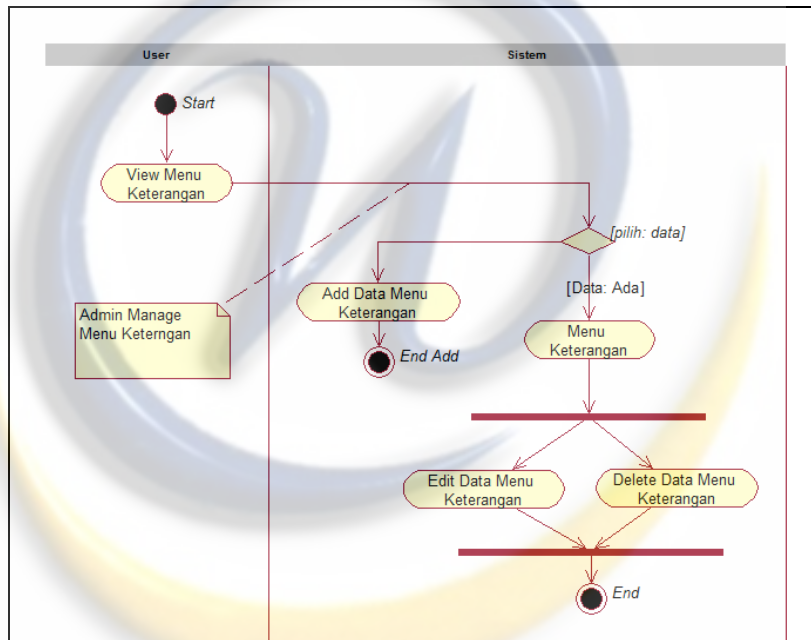
4.4.11 Skenario Use Case Admin Manage Menu Keterangan

Tabel 4. 9 Skenario Use Case Admin Manage Menu Keterangan

Name	Admin mengelola menu Keterangan pada aplikasi
Level	Level Admin
Actors	<i>Admin</i>

Goal	Untuk mengelola menu Keterangan pada aplikasi
Precondition	<i>Admin</i> belum mengakses menu aplikasi dan tidak bisa merubah menu Keterangan
Postcondition	<i>Admin</i> dapat menu aplikasi dan memanipulasi menu Keterangan pada aplikasi
Steps	<i>Admin</i> mengakses menu aplikasi

4.4.12 Activity Diagram Admin Manage Menu Keterangan



Gambar 4. 9 Activity Diagram Admin Manage Menu Keterangan

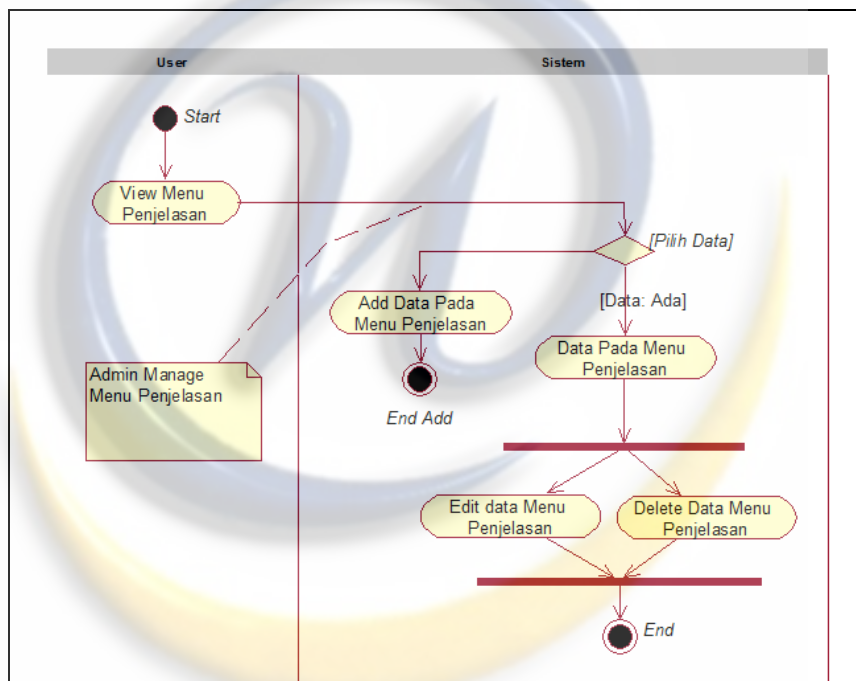
4.4.13 Skenario Use Case Admin Manage Menu Penjelasan

Tabel 4. 10 Skenario Use Case Admin Manage Menu Penjelasan

Name	Admin mengelola menu Penejelasan pada aplikasi
Level	Level Admin
Actors	<i>Admin</i>

Goal	Untuk mengelola menu Penjelasan pada aplikasi
Precondition	<i>Admin</i> belum mengakses menu aplikasi dan tidak bisa merubah menu Penjelasan
Postcondition	<i>Admin</i> dapat menu aplikasi dan memanipulasi menu Penjelasan pada aplikasi
Steps	<i>Admin</i> mengakses menu aplikasi

4.4.14 Activity Diagram Admin Manage Menu Penjelasan



Gambar 4. 10 Activity Diagram Admin Manage Menu Penjelasan

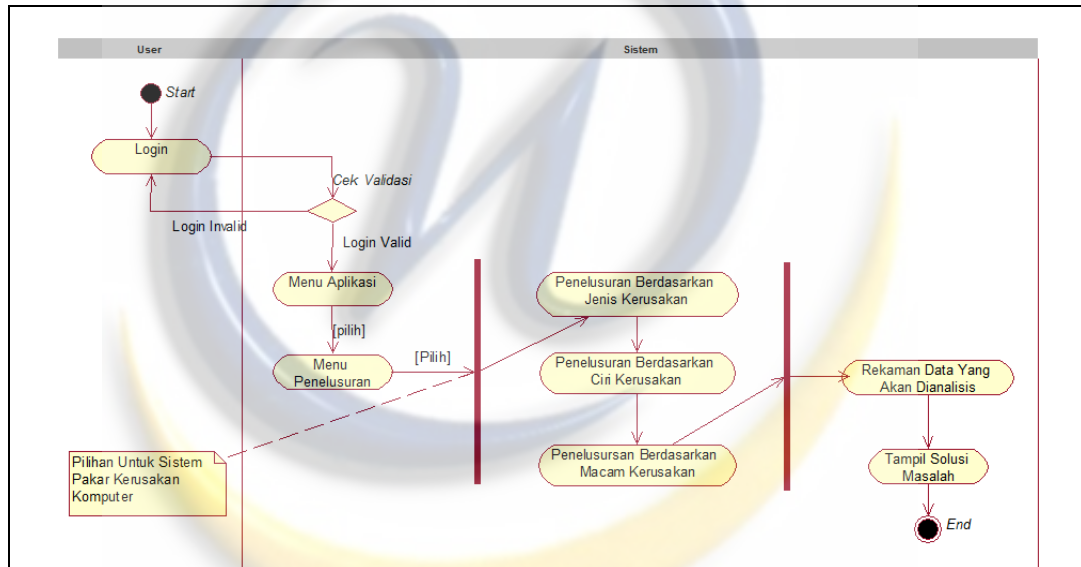
4.4.15 Skenario Use Case User Menggunakan Sistem Pakar

Tabel 4. 11 Skenario Use Case User Menggunakan Sistem Pakar

Name	User Menggunakan Aplikasi
Level	Level User
Actors	<i>User</i>

Goal	Untuk Mencari Solusi dari permasalahan yang dialami User
Precondition	User belum mengakses Aplikasi dan tidak bisa merubah mencari solusi permasalahan
Postcondition	User dapat mendapatkan solusi dari permasalahan melalui system
Steps	User mengakses Menu Aplikasi

4.4.16 Activity Diagram User Menggunakan Sistem Pakar



Gambar 4. 11 Activity Diagram User Menggunakan Sistem Pakar

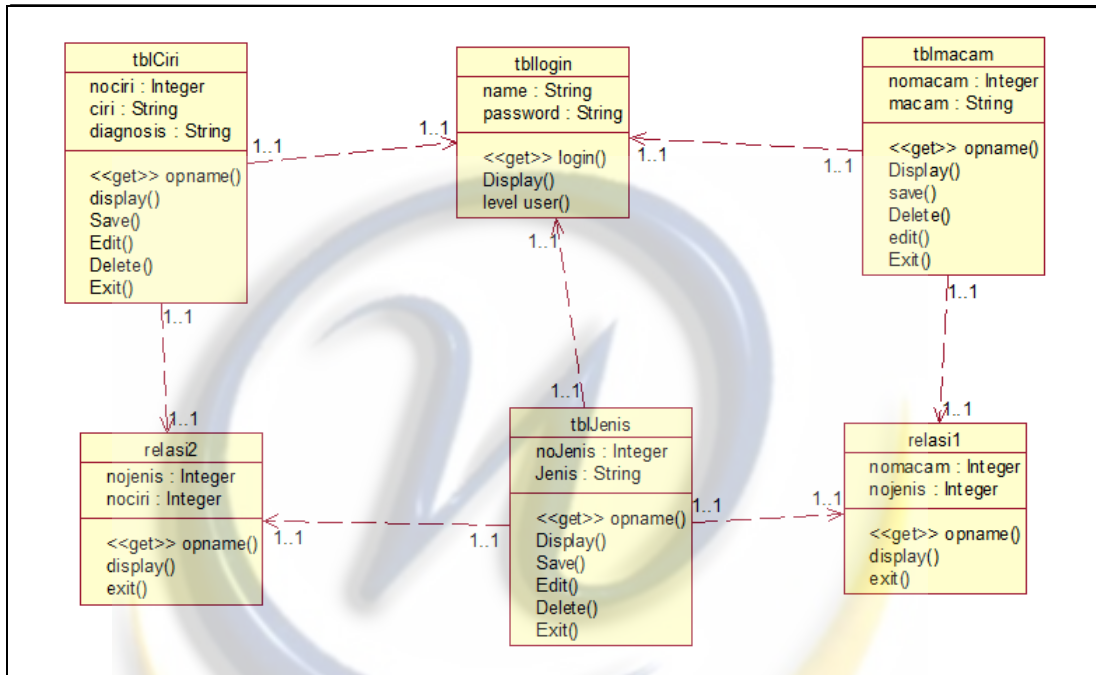
4.5 Realisasi Use Case Tahap Perancangan

Realisasi *use case* tahap perancangan digambarkan dalam *class Diagram*, *sequence Diagram*, dan *collabaration Diagram* yang menggambarkan interaksi setiap objek dari kelas perancangan yang terlibat di dalam *use case* tersebut.

4.5.1 Class Diagram

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas yang ada dalam sistem perangkat lunak yang akan dikembangkan. *Class*

Diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Berikut ini digambarkan *class* diagram dari Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer:



Gambar 4. 12 Class Diagram

4.5.2 Kamus Data

Berikut ini akan dijelaskan tabel – tabel yang digunakan dalam perancangan proses *Unified Modelling Language* (UML).

1. Tabel Login

Nama Tabel : tbllogin

Prymary Key : name

Keterangan : Tabel ini berisikan data – data Login

Tabel 4. 12 Tabel Login

Field	Type	Size	Keterangan
username	varchar	7	Username
Password	varchar	10	Password

2. Tabel Ciri Kerusakan

Nama Tabel : tblciri

Primary Key : nociri

Keterangan : Tabel ini berisikan ciri kerusakan

Tabel 4. 13 Tabel Ciri Kerusakan

Field	Type	Size	Keterangan
Nociri	int	3	Nomer cirri kerusakan untuk pengurutan data berdasarkan ciri kerusakan
Cirri	varchar	15	Cirri dari kerusakan
Diagnosis	varchar	50	Diagnosis berdasarkan cirri kerusakan

3. Tabel Macam Kerusakan

Nama Tabel : tblMacam

Primary Key : no_macam

Keterangan : tabel ini berisi data mengenai macam kerusakan

Tabel 4. 14 Tabel Macam Kerusakan

Field	Type	Size	Keterangan
No_macam	int	3	Nomer identitas untuk pengurutan macam kerusakan
Macam	varchar	35	Macam dari kerusakan

4. Tabel Jenis Kerusakan

Nama Tabel : tblJenis

Primary Key : no_jenis

Keterangan : Tabel untuk data jenis kerusakan

Tabel 4. 15 Tabel Jenis Kerusakan

Field	Type	Size	Keterangan
No_jenis	Int	3	Nomer identitas untuk pengurutan jenis kerusakan
Jenis	varchar	35	jenis dari kerusakan

5. Tabel Relasi 1

Nama tabel : tblrelasi1

Primary key : no_macam

Keterangan : tabel relasi dari tblmacam dan tbl jenis

Tabel 4. 16 Tabel Relasi 1

Field	Type	Size	Keterangan
No_macam	Int	3	Nomer identitas macam kerusakan
No_jenis	Int	3	Nomer identitas jenis kerusakan

6. Tabel Relasi 2

Nama tabel : tblrelasi2

Primary key : no_jenis

Keterangan : tabel relasi dari tblciri dan tbl jenis

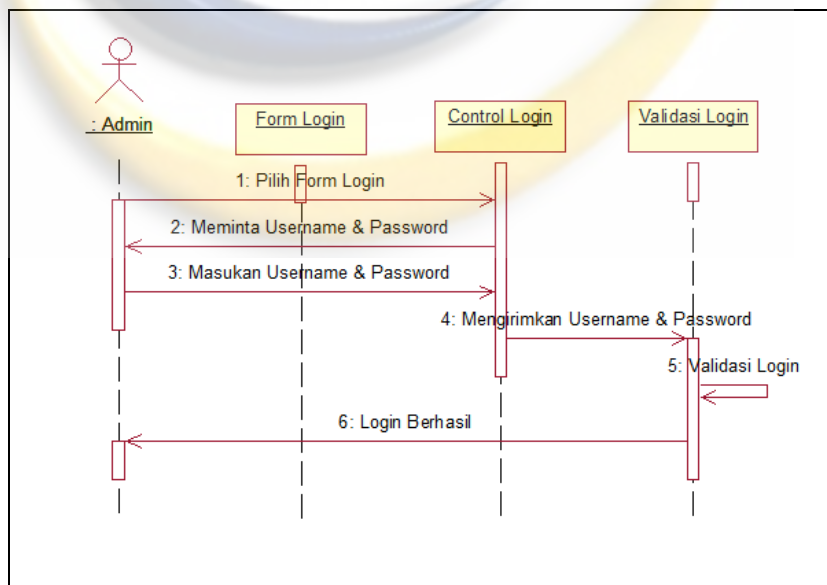
Tabel 4. 17 Tabel Relasi 2

Field	Type	Size	Keterangan
No_jenis	Int	3	Nomer identitas jenis kerusakan
No_ciri	Int	3	Nomer identitas cirikerusakan

4.5.3 Sequence Diagram

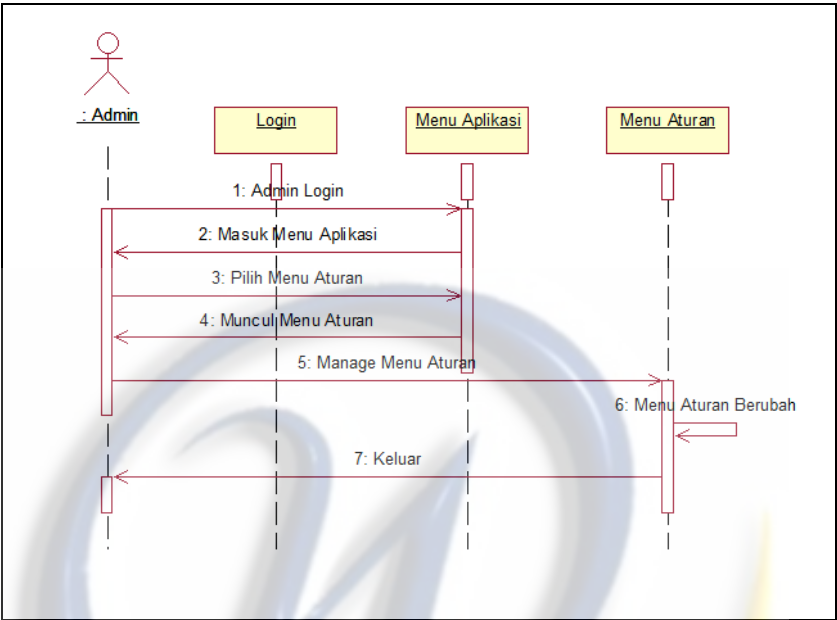
Sequence Diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. berikut *Sequence Diagram Use Case* Login admin Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer:

4.5.3.1 Sequence Diagram Login



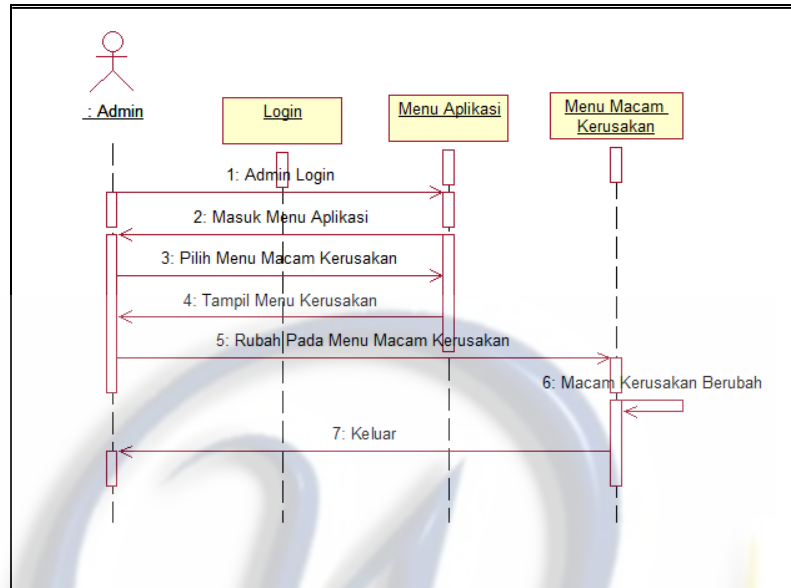
Gambar 4. 13 Sequence Diagram Login

4.5.3.2 Sequence Diagram Admin Manage Menu Aturan



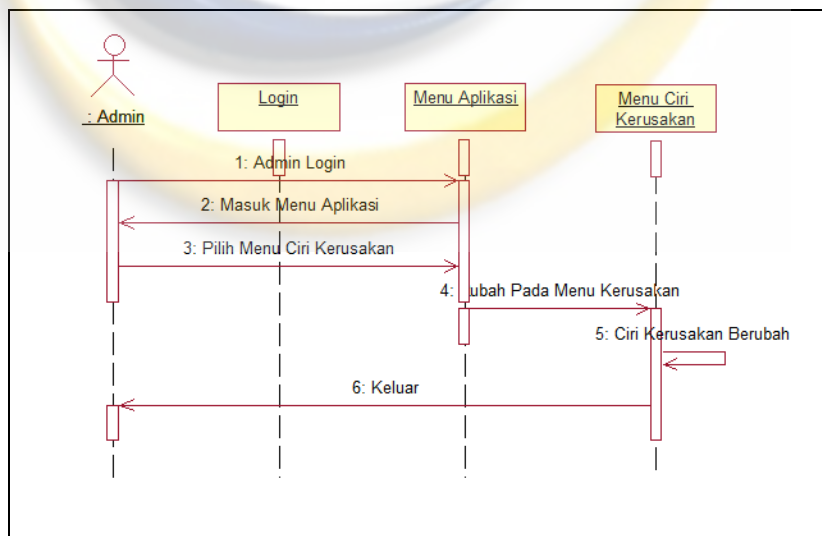
Gambar 4. 14 Sequence Diagram Admin Manage Menu Aturan

4.5.3.3 Sequence Diagram Admin Manage Macam Kerusakan



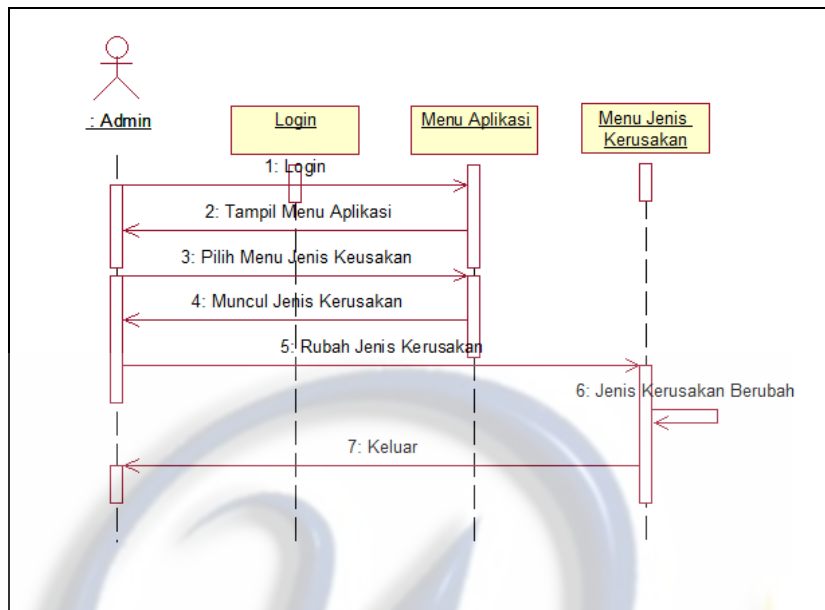
Gambar 4. 15 Sequence Diagram Admin Manage Macam Kerusakan

4.5.3.4 Sequence Diagram Admin Manage Ciri Kerusakan



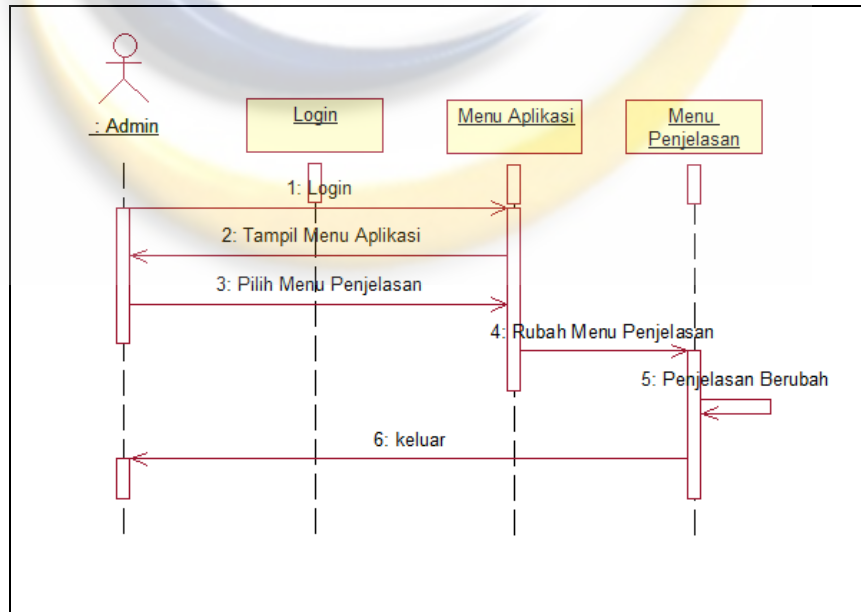
Gambar 4. 16 Sequence Diagram Admin Manage Ciri Kerusakan

4.5.3.5 Sequence Diagram Admin Manage Jenis Kerusakan



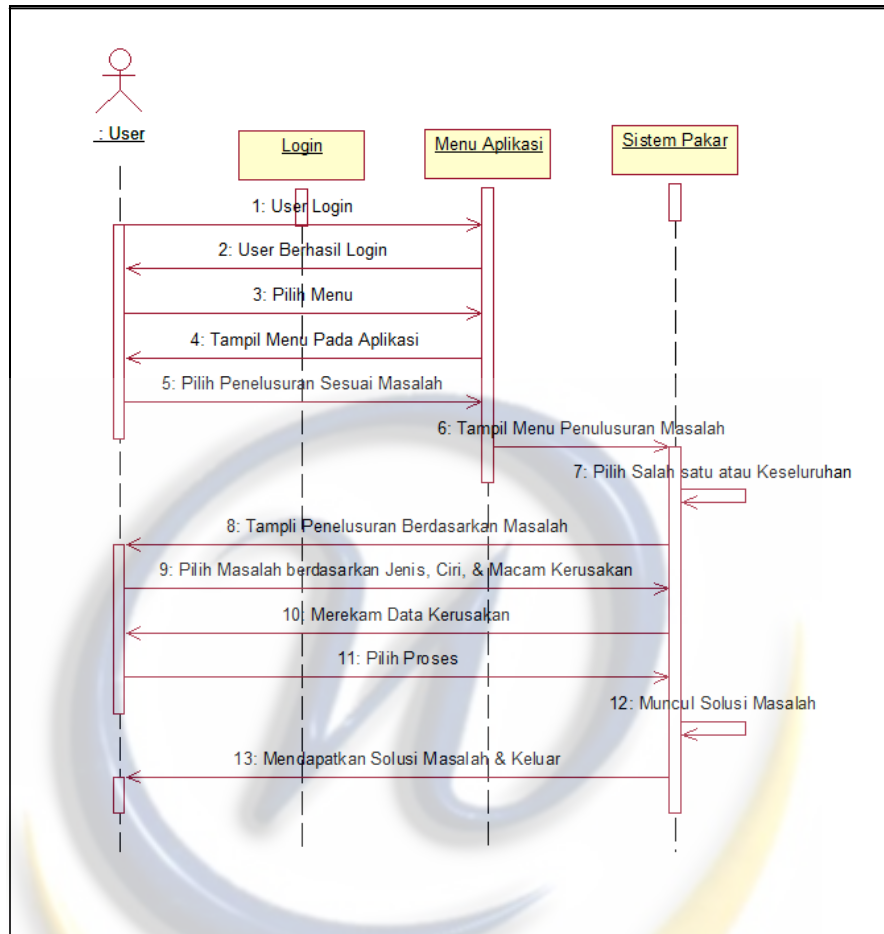
Gambar 4. 17 Sequence Diagram Admin Manage Jenis Kerusakan

4.5.3.6 Sequence Diagram Admin Manage Menu Penjelasan



Gambar 4. 18 Sequence Diagram Admin Manage Menu Penjelasan

4.5.3.7 Sequence Diagram User Menggunakan Sistem Pakar

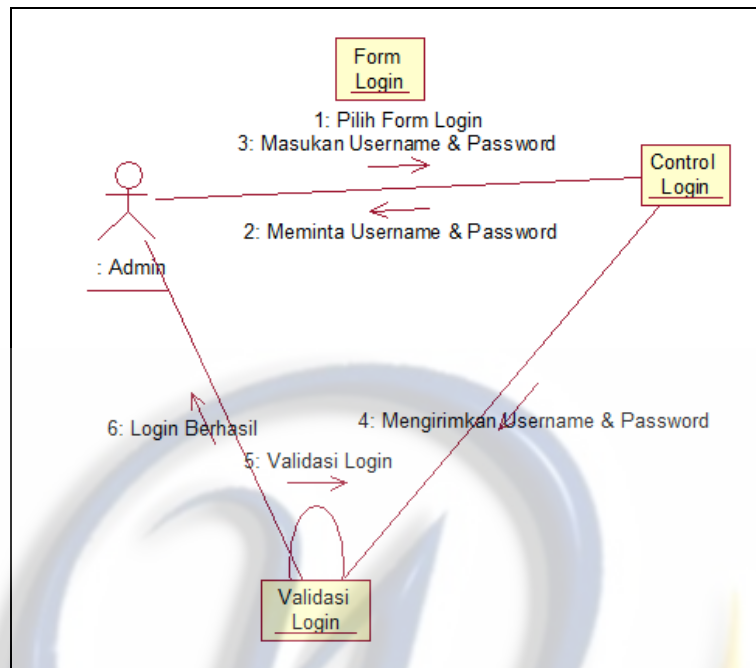


Gambar 4. 19 Sequence Diagram User Menggunakan Sistem Pakar

4.5.4 Collaboration Diagram Login

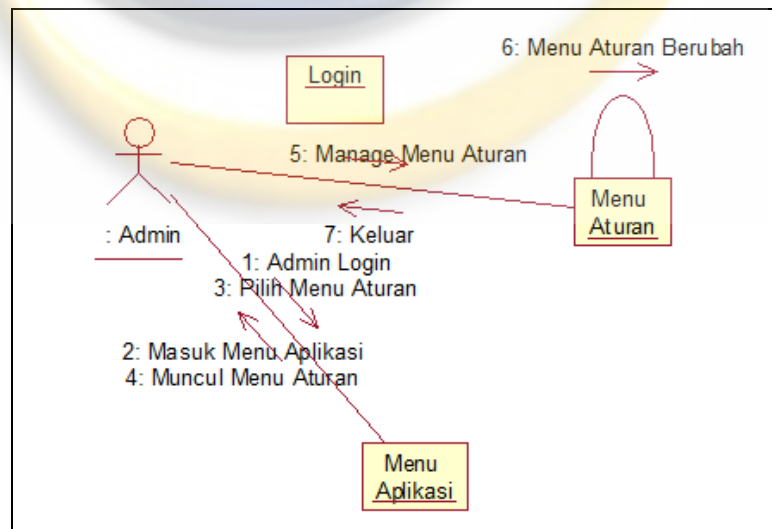
Collaboration Diagram digunakan untuk memodelkan interaksi antar objek didalam sistem, berbeda dengan *sequence* Diagram, yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* Diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan objek. Berikut ini *collaboration* Diagram dari Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer:

4.5.4.1 Collaboration Diagram Login



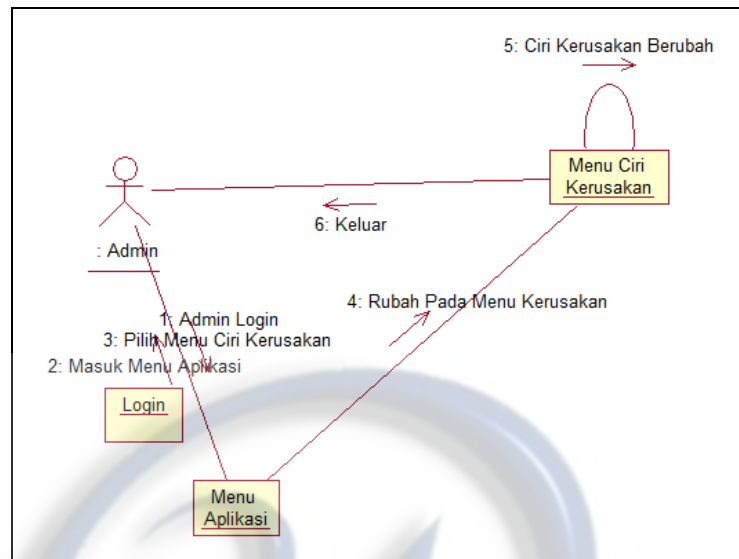
Gambar 4. 20 Collaboration Diagram Login

4.5.4.2 Collaboration Diagram Admin Manage Menu Aturan



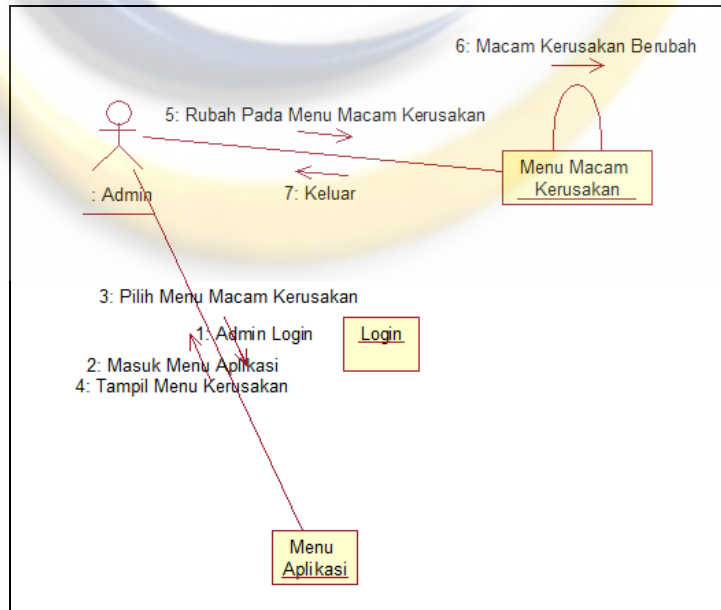
Gambar 4. 21 Collaboration Diagram Admin Manage Menu Aturan

4.5.4.3 Collaboration Diagram Admin Manage Ciri Kerusakan



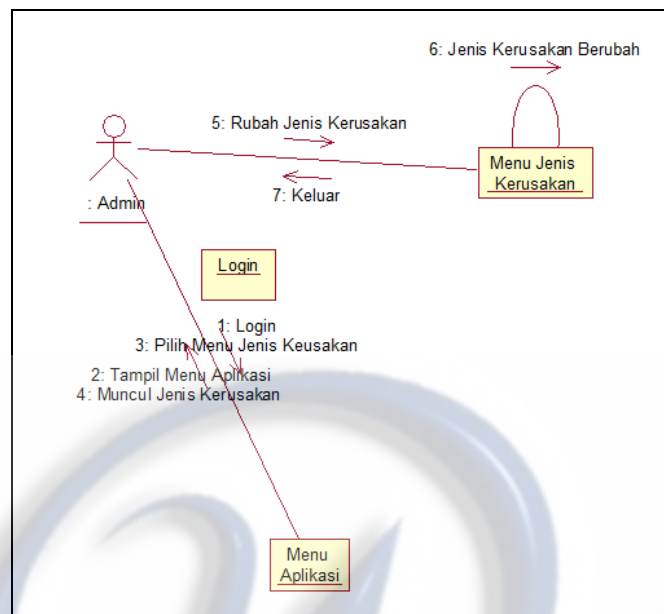
Gambar 4. 22 Collaboration Diagram Admin Manage Ciri Kerusakan

4.5.4.4 Collaboration Diagram Admin Manage Macam Kerusakan



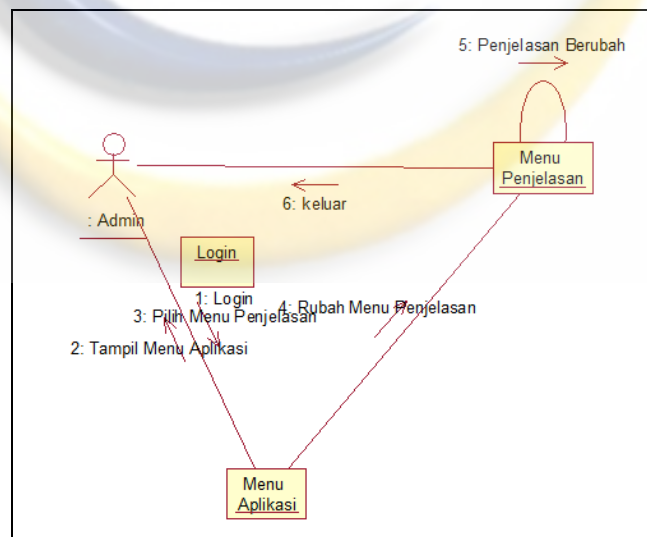
Gambar 4. 23 Collaboration Diagram Admin Manage Macam Kerusakan

4.5.4.5 Collaboration Diagram Admin Manage Jenis Kerusakan



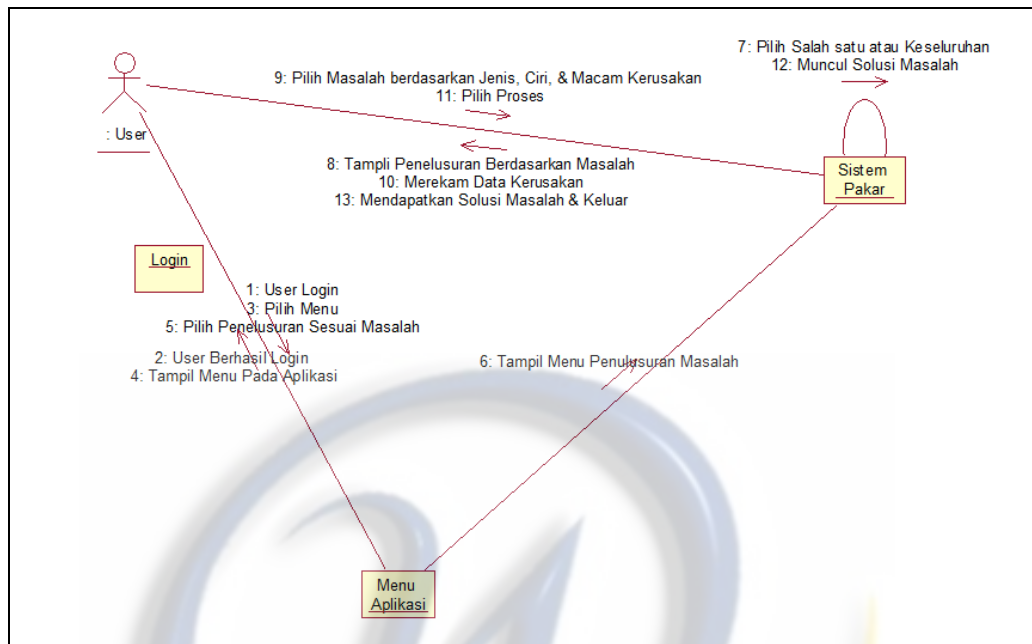
Gambar 4. 24 Collaboration Diagram Admin Manage Jenis Kerusakan

4.5.4.6 Collaboration Diagram Admin Manage Menu Penjelasan



Gambar 4. 25 Collaboration Diagram Admin Manage Menu Penjelasan

4.5.4.7 Collaboration Diagram User Menggunakan Sistem Pakar



Gambar 4. 26 Collaboration Diagram User Menggunakan Sistem Pakar

4.6 Lingkungan Operasional

Lingkungan operasional merupakan kebutuhan-kebutuhan perangkat lunak pada saat diimplementasikan, baik itu perangkat lunak, perangkat keras, maupun karakteristik dari pengguna perangkat lunak tersebut.

4.7 Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini adalah sebagai berikut:

1. Sistem operasi *Windows XP Profesional, Windows Vista, Windows 7*
2. Software VB6 Tools Pembangun
3. MS Access 2000 sebagai tempat penyimpanan database.
4. Rational Rose untuk membuat atau merancang UML

4.8 Perangkat Keras

Perangkat keras yang digunakan pada Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer ini adalah sebagai berikut:

- a. *Intel Pentium 1.6 GHz*
- b. *Memory 512 MB*
- c. *Hard disk 40 GB*

4.9 Karakteristik Pengguna

Kualifikasi dan hak akses yang harus dimiliki pengguna adalah sebagai berikut:

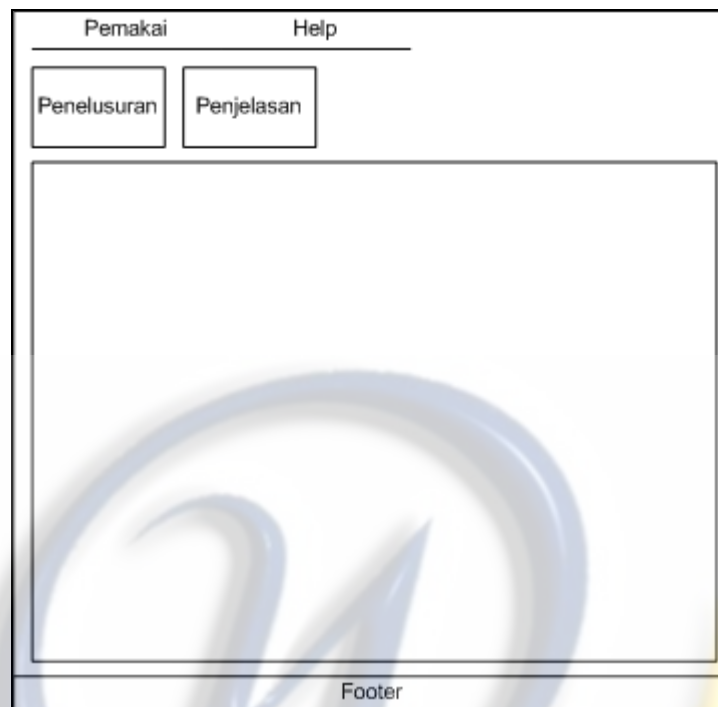
Tabel I-1
Karakteristik Pengguna Untuk Admin

Uraian	Keterangan
Kualifikasi	1. Mempunyai kemampuan dasar di bidang komputer 2. Dapat mengoperasikan sistem operasi <i>Windows</i>

4.10 Layout Antarmuka

Layout antarmuka merupakan rancangan antarmuka yang akan digunakan sebagai perantara *user* dengan perangkat lunak yang dikembangkan. *Layout* antarmuka dari Aplikasi Sistem Pakar Diagnosis Kerusakan Komputer:

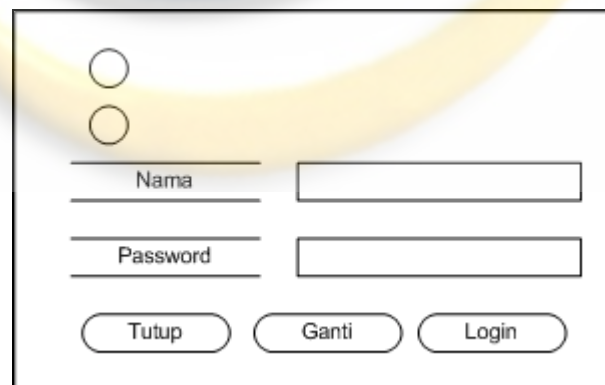
4.10.1 Antarmuka Tampilan Utama Sistem



The screenshot shows a web application interface. At the top, there is a header bar with two links: 'Pemakai' and 'Help'. Below the header, there are two buttons: 'Penelusuran' and 'Penjelasan'. The main content area is a large, empty rectangular box. At the bottom of the page, there is a 'Footer' label.

Gambar 4. 27 Antarmuka Tampilan Utama Sistem

4.10.2 Antarmuka Menu Admin Login



The screenshot shows an admin login form. It features two radio buttons at the top. Below them are two input fields: 'Nama' and 'Password'. At the bottom, there are three buttons: 'Tutup', 'Ganti', and 'Login'.

Gambar 4. 28 Antarmuka Menu Admin Login

4.10.3 Antarmuka Menu Admin Aturan

The screenshot displays a web-based administrative interface for managing damage rules. It features a form with the following components:

- Macam Kerusakan**: A text input field at the top left.
- Jenis Kerusakan**: A larger text input field below the first one.
- Ciri Kerusakan**: A text input field located in the bottom left section.
- Diagnosa Kerusakan**: A text input field located in the bottom right section.
- Tambah**: A button at the bottom center for adding new rules.
- Tutup**: A button at the bottom right for closing the form.

The interface is overlaid with a large, semi-transparent watermark consisting of a blue '@' symbol and a yellow crescent moon.

Gambar 4. 29 Antarmuka Menu Admin Aturan

4.10.4 Antamuka Menu Admin Ciri Kerusakan

Ciri Kerusakan Komputer

No / Ciri Kerusakan

Diagnosa

Isi Diagnosa

Navigasi

List View

Tambah Edit Simpan Hapus Keluar

Gambar 4. 30 Antamuka Menu Admin Ciri Kerusakan

4.10.5 Antarmuka Menu Admin Macam Kerusakan

Macam Kerusakan Mesin

Tambah

List View

Edit Hapus Tutup

Gambar 4. 31 Antarmuka Menu Admin Macam Kerusakan

4.10.6 Antarmuka Menu Admin Jenis Kerusakan

Jenis Kerusakan Komputer

Jenis Kerusakan

Gejala Kerusakan

Isi Diagnosa

Navigasi

List View

Tambah Edit Simpan Hapus Keluar

Gambar 4. 32 Antarmuka Menu Admin Jenis Kerusakan

4.10.7 Antarmuka Menu Admin Penjelasan

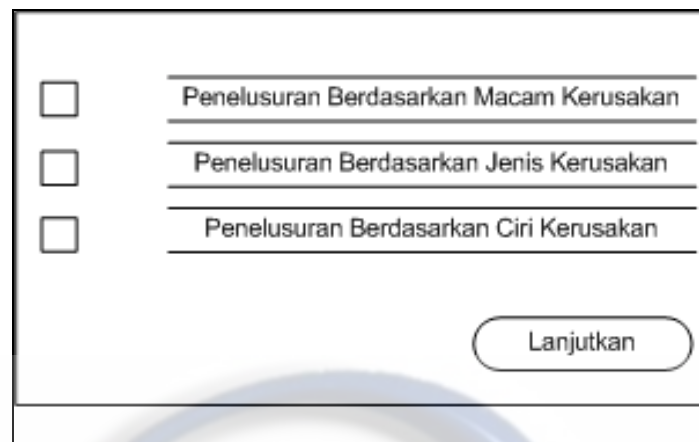
Penjelasan Sistem

Isi Penjelasan

Simpan Tutup

Gambar 4. 33 Antarmuka Menu Admin Penjelasan

4.10.8 Antarmuka Menu Penelusuran



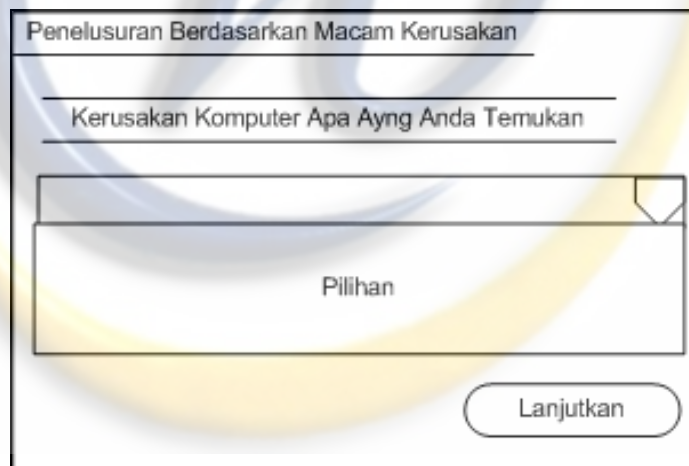
The screenshot shows a search menu interface with three radio button options and a 'Lanjutkan' button. The options are:

- ☐ Penelusuran Berdasarkan Macam Kerusakan
- ☐ Penelusuran Berdasarkan Jenis Kerusakan
- ☐ Penelusuran Berdasarkan Ciri Kerusakan

The 'Lanjutkan' button is located at the bottom right of the menu.

Gambar 4. 34 Antarmuka Menu Penelusuran

4.10.9 Antarmuka Menu Penelusuran Berdasarkan Macam Kerusakan



The screenshot shows the search menu interface for 'Macam Kerusakan'. It includes a title bar, a text input field, and a 'Lanjutkan' button. The text input field contains the text 'Kerusakan Komputer Apa Ayng Anda Temukan'.

Penelusuran Berdasarkan Macam Kerusakan

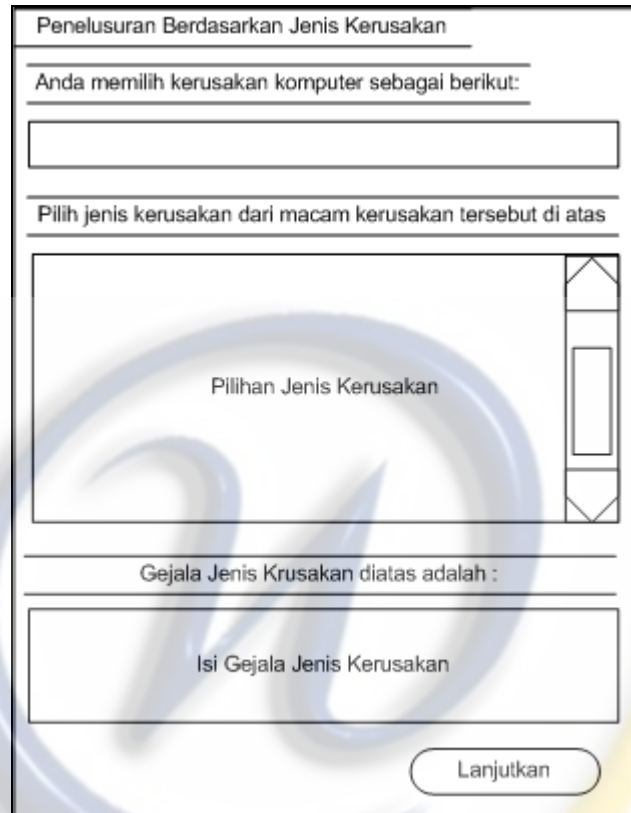
Kerusakan Komputer Apa Ayng Anda Temukan

Pilihan

Lanjutkan

Gambar 4. 35 Antarmuka Menu Penelusuran Berdasarkan Macam Kersusakan

4.10.10 Antarmuka Menu Penelusuran Berdasarkan Jenis Kerusakan



Penelusuran Berdasarkan Jenis Kerusakan

Anda memilih kerusakan komputer sebagai berikut:

Pilih jenis kerusakan dari macam kerusakan tersebut di atas

Pilihan Jenis Kerusakan

Gejala Jenis Kerusakan diatas adalah :

Isi Gejala Jenis Kerusakan

Lanjutkan

Gambar 4. 36 Antarmuka Menu Penelusuran Berdasarkan Jenis Kerusakan

4.10.11 Antarmuka Menu Penelusuran Berdasarkan Ciri Kerusakan

Penelusuran Berdasarkan Ciri Kerusakan

Anda memilih kerusakan komputer sebagai berikut:

Pilih Ciri kerusakan dari Jenis kerusakan tersebut di atas

Pilihan Ciri Kerusakan

Lanjutkan

Gambar 4. 37 Antarmuka Menu Penelusuran Berdasarkan Ciri Kerusakan

4.10.12 Antarmuka Menu Rekaman Data

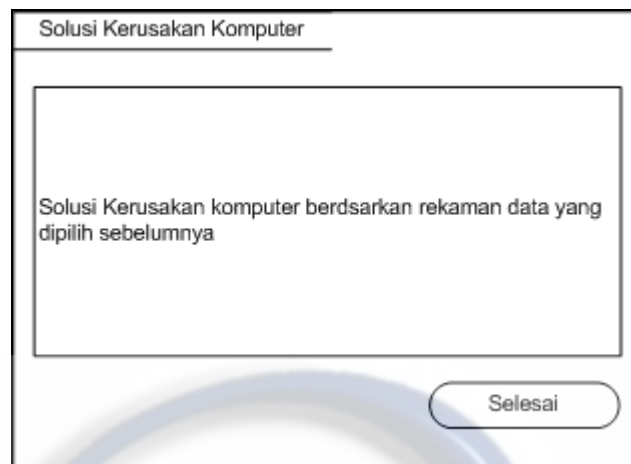
Rekaman data yang akan di analisis

Rekaman Data

Proses

Gambar 4. 38 Antarmuka Menu Rekaman Data

4.10.13 Antarmuka Menu Solusi



Solusi Kerusakan Komputer

Solusi Kerusakan komputer berdsarkan rekaman data yang dipilih sebelumnya

Selesai

Gambar 4. 39 Antarmuka Menu Solusi

BAB V

IMPLEMENTASI SISTEM

5.1 Implementasi

Pada bab ini akan diuraikan cara dan langkah-langkah untuk mengimplementasikan rancangan perangkat lunak, kebutuhan perangkat lunak maupun perangkat keras yang digunakan, pengujian sistem dan klasifikasi infrastruktur .

Berikut ini aktifitas yang dilakukan dalam mengimplementasikan Aplikasi Sistem pakar diagnosa kerusakan komputer :

a. Persiapan Instalasi Perkakas (*Tools*) Yang Akan Digunakan

Tahap awal yang dilakukan dalam rangka instalasi perkakas apa saja yang dibutuhkan untuk implementasi Aplikasi Sistem pakar diagnosa kerusakan komputer.

b. Pengetesan Program Pada Perangkat Lunak

Tahap ini dilakukan untuk menghindari kesalahan antara lain :

- a. Kesalahan penulisan *source code* program
- b. Kesalahan *database*
- c. Kesalahan saat program yang dijalankan
- d. Kesalahan logika

c. Evaluasi Perangkat Lunak

Tahap ini diperlukan untuk memastikan apakah perangkat lunak dapat berjalan dengan baik, serta mengidentifikasi *error* sehingga perbaikan dapat segera dilakukan hingga perangkat lunak benar-benar siap guna.

5.1.1 Lingkup dan Batasan Implementasi

Lingkup dan batasan implementasi Aplikasi Sistem pakar diagnosa kerusakan komputer, yaitu :

- a. Penerapan Aplikasi Sistem pakar diagnosa kerusakan komputer hanya berjalan didalam sistem operasi *windows*.
- b. Bahasa pemrograman yang digunakan yaitu *VB6*.
- c. *Database yang digunakan adalah MS Access*

5.2 Kebutuhan Sumberdaya

Kebutuhan sumber daya manusia yang dibutuhkan untuk pengujian hanya lah satu pengguna saja. Sedangkan kebutuhan *hardware* dan *software* nya kemungkinan sama pada saat implementasi aplikasi ini. Ataupun sebagai berikut :

5.2.1 Kebutuhan Perangkat Keras (*Hardware*)

Konfigurasi minimal perangkat keras untuk mendukung sistem yang dirancang sebagai berikut:

Tabel 5. 1 Spesifikasi Minimum Perangkat Keras yang Digunakan

No	Uraian	Spesifikasi
1	<i>Processor</i>	<i>Processor Intel pentium 1.6 Ghz</i>
2	<i>Memory (RAM)</i>	<i>512MB</i>
3	<i>Hard Disk drive</i>	<i>40 GB</i>
4	<i>Graphic Card (VGA)</i>	<i>Onboard</i>
5	<i>Monitor</i>	<i>14 “</i>
6	<i>Keyboard</i>	<i>Standar</i>
7	<i>Mouse</i>	<i>Standar</i>

5.2.2 Kebutuhan Perangkat Lunak (*Software*)

perangkat lunak yang dipakai adalah sebagai berikut :

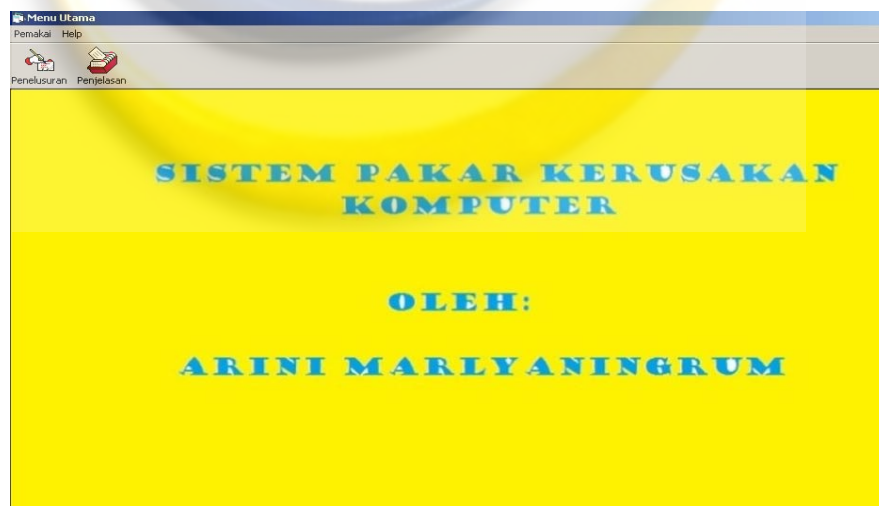
1. Sistem Operasi *Windows XP, Windows 7 Ultimate*
2. *MS Access*
3. *VB6*

5.3 Implementasi Antarmuka Pemakai

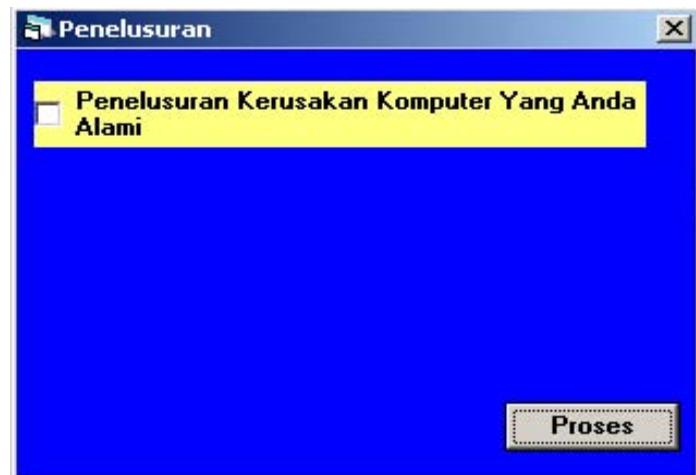
Implementasi rancangan antarmuka dengan menggunakan bahasa pemrograman VB6, tahapan antarmuka pemakai merupakan bagian yang menyediakan sarana untuk pemakai agar bisa berkomunikasi dengan sistem dalam bentuk program aplikasi. Antarmuka pemakai akan mengajukan beberapa pertanyaan untuk informasi awal dalam pencarian suatu solusi yang akan dilakukan.



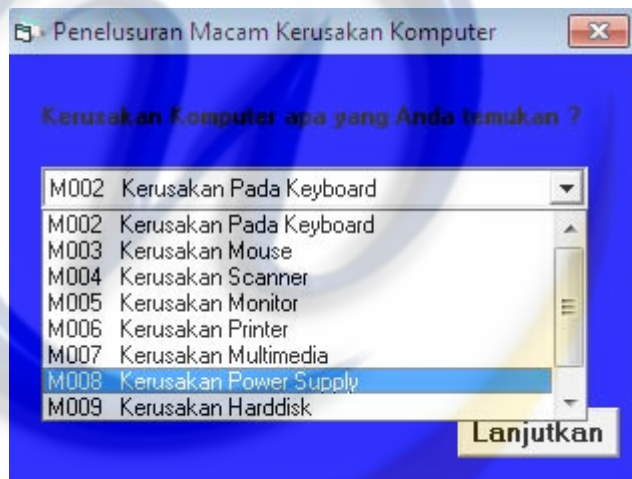
Gambar 5. 1 Menu *Login*



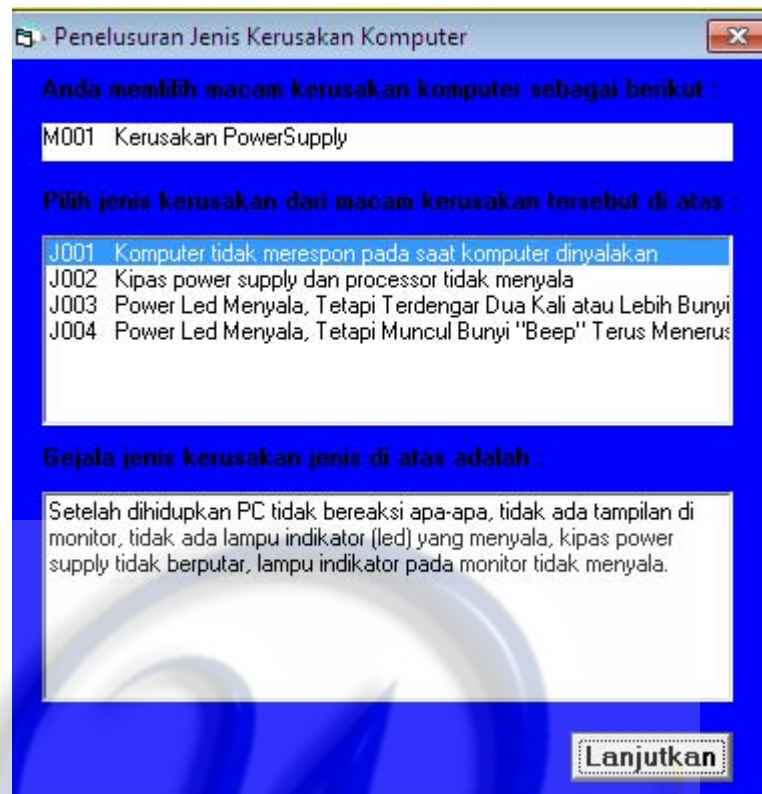
Gambar 5. 2 Menu Utama



Gambar 5.3 Menu penelusuran



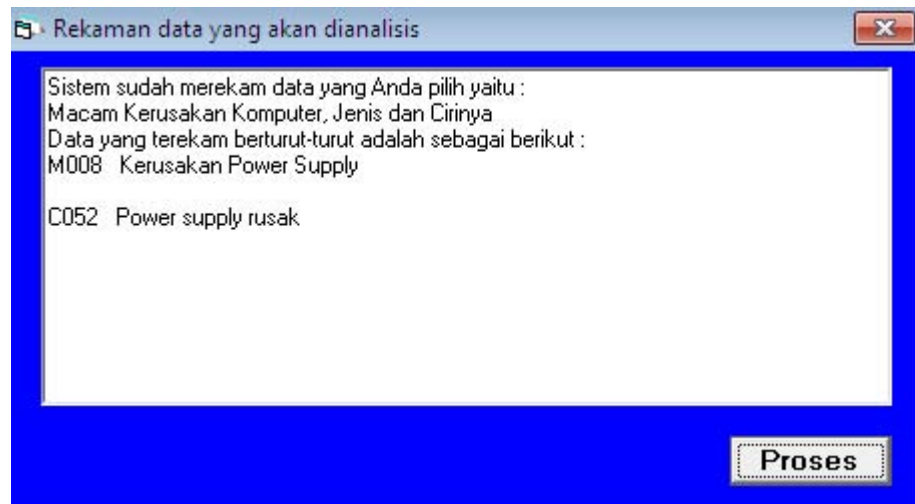
Gambar 5.4 Menu penelusuran macam kerusakan



Gambar 5.5 Menu penelusuran jenis kerusakan



Gambar 5.6 Menu penelusuran ciri kerusakan



Gambar 5.7 Menu rekaman data yang akan dianalisis



Gambar 5.8 Menu Solusi

5.4 Hasil Pengujian

Hasil pengujian dilakukan dengan menggunakan metode *Black Box Testing*, merupakan metode *testing* pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Dengan menggunakan metode pengujian *black box*, perekayasa sistem dapat menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi tidak benar atau hilang
2. Kesalahan antar muka
3. Kesalahan pada struktur data (pengaksesan basis data)
4. Kesalahan inisialisasi dan akhir program
5. Kesalahan kinerja.

Tabel 5. 2 Pengujian Perangkat Lunak Pada Admin

No.	Fungsi yang diuji	Cara Pengujian	Hasil Yang diharapkan	Hasil Pengujian
1	Login	Melakukan <i>login</i> untuk membuka <i>form</i> utama	Menampilkan <i>form</i> Utama.	OK
2	Form Utama	Membuka <i>form</i> Utama.	Menampilkan daftar menu dari fasilitas yang disediakan aplikasi berdasarkan hak akses.	OK
3	Tambah isi macam kerusakan	Pada <i>form</i> utama pilih menu home, lalu klik <i>icon</i> tambah.	Menambahkan isi data macam kerusakan	OK
4	Edit isi macam kerusakan	Pada <i>form</i> utama pilih menu home, lalu klik edit.	Merubah isi data macam kerusakan	OK
5	Hapus isi macam kerusakan	Pada <i>form</i> utama pilih data pada menu home, lalu klik hapus.	Menghapus data isi macam kerusakan	OK

6	Tambah ciri kerusakan	Pada <i>form</i> utama pilih menu <i>About</i> , lalu klik <i>icon</i> tambah.	Menambahkan data ciri kerusakan	OK
7	Edit ciri kerusakan	Pada <i>form</i> utama pilih menu <i>About</i> , lalu klik <i>icon</i> edit.	Merubah data isi ciri kerusakan	OK
8	Hapus ciri kerusakan	Pada <i>form</i> utama pilih data pada menu <i>About</i> , lalu klik <i>icon</i> hapus.	Menghapus data isi ciri kerusakan	OK
9	Tambah jenis kerusakan	Pada <i>form</i> utama pilih menu <i>list</i> , lalu klik <i>icon</i> tambah	Menambahkan jenis kerusakan	OK
10	Edit jenis kerusakan	Pada <i>form</i> utama pilih menu <i>list</i> , lalu klik <i>icon</i> edit	Merubah data jenis kerusakan	OK
11	Hapus jenis kerusakan	Pada <i>form</i> utama pilih data pada menu <i>list sale</i> , lalu klik <i>icon</i> hapus.	Menghapus data jenis kerusakan	OK

Tabel 5. 3 Pengujian Perangkat Lunak Pada User

No.	Fungsi yang diuji	Cara Pengujian	Hasil Yang diharapkan	Hasil Pengujian
1	<i>Form</i> Utama	Membuka <i>form</i> Utama.	Menampilkan daftar menu dari fasilitas yang disediakan aplikasi berdasarkan hak akses.	OK
2	Penelusuran	Pada <i>form</i> utama klik proses untuk memilih macam, jenis, gejala, ciri dari kerusakan komputer	Melanjutkan ke <i>form</i> macam kerusakan	OK

3	Macam Kerusakan	Pada <i>form</i> utama pilih macam kerusakan yang dialami, lalu klik lanjut.	Melanjutkan ke <i>form</i> jenis kerusakan komputer.	OK
4	Jenis Kerusakan	Pada <i>form</i> utama pilih jenis kerusakan computer yang dialami, lalu klik lanjut.	Melanjutkan ke <i>form</i> ciri kerusakan komputer.	OK
5	Ciri Kerusakan	Pada <i>form</i> utama pilih ciri kerusakan computer yang dialami , lalu klik lanjut.	Melanjutkan ke <i>form</i> rekaman data analisis.	OK
6	Rekaman Data Analisis	Pada <i>form</i> utama menampilkan hasil analisi yang dipilih dari macam, jenis, ciri kerusakan computer, lalu klik proses.	Melanjutkan ke <i>form</i> solusi.	OK
7	Solusi	Pada <i>form</i> utama memproses hasil macam, jenis, ciri dari kerusakan komputer.	Menghasilkan solusi dari masalah kerusakan komputer.	OK

Tabel 5.4 Pengujian Perbandingan diagnosa kerusakan komputer secara manual dan menggunakan aplikasi

No.	Fungsi yang diuji	Waktu untuk mendiagonosai kerusakan secara manual		Waktu untuk mendiagnosea kerusakan menggunakan aplikasi	Hasil Pengujian
1	Power Supply PC sering restart	2 jam 10 menit	1 jam 30 menit	45 menit	OK
	Rata-rata	2 jam			

2	Power Supply PC mati	50 menit	1 jam	38 menit	OK
	Rata-rata	1 jam			
3	Komponen yang terhubung pdMotherboard bermasalah	1 jam 15 menit	55 menit	35 menit	OK
	Rata-rata	1 jam			
4	Harddisk kehilangan sistem	30 menit	38 menit	20 menit	OK
	Rata-rata	35 menit			
5	Monitor tidak menampilkan gambar	23 menit	32 menit	15 menit	OK
	Rata-rata	30 menit			
6	Kabel-kabel CD ROOM tidak terpasang dengan benar	32 menit	45 menit	25 menit	OK
	Rata-rata	45 menit			

BAB VI PENUTUP

Berdasarkan hasil pengembangan sistem yang telah penulis lakukan, penulis mencoba membuat suatu kesimpulan dan mengajukan saran-saran yang berhubungan dengan pembahasan yang telah dikemukakan di bab-bab sebelumnya.

6.1 Kesimpulan

Dari berbagai penjelasan yang telah diuraikan dalam laporan ini, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Aplikasi yang dibuat untuk memudahkan teknisi dalam kerusakan komputer yang dalam cara penyajiannya terdiri dari macam, jenis, cir-ciri kerusakan komputer.
2. Waktu yang dibutuhkan untuk mendiagnosa kerusakan pada komputer apabila menggunakan aplikasi adalah 51% dari waktu yang dibutuhkan teknisi apabila mendiagnosa secara manual.

6.2 Saran

Adapun saran yang dapat penulis berikan untuk pengembangan dari sistem yang telah dibuat antara lain :

1. Aplikasi ini dapat dikembangkan sampai kepada report untuk setiap solusi yang diberikan.
2. Jenis dan ciri komputer yang terdapat pada program aplikasi ini jumlahnya dapat ditambahkan lebih banyak lagi.

DAFTAR PUSTAKA

- [1] Arhami, Muhammad, Konsep Dasar Sistem Pakar, Andi, Edisi Pertama, Yogyakarta, 2005.
- [2] Fowler, Martin dan Scott, Kendall. *UML Distilled: A brief guide to the standard object modeling language*. Reading : Addison Wesley, 2000.
- [3] Pressman Roger S., *Rekayasa Perangkat Lunak Pendekatan Praktisi* (Buku Satu). Mc Graw Hill Book co. Andi Offset. Yogyakarta.2002.
- [4] <http://www.visualbasicindonesia.com/definisi-visual-basic/Yogyakarta>. 2001.
- [5] <http://id.scribd.com/doc/102259326/Pengertian-Microsoft-Access>

LAMPIRAN

Kode (Coding) Program

Code program yang dibuat dalam aplikasi ini meliputi MDIUtama, FrmAturan, FrmCiriKerusakan, FrmJenisKerusakan, FrmKeterangan, FrmMacamKerusakan, FrmPassword, FrmPenelusuran, FromPenjelasan, berikut kode programnya :

Source Code Program Untuk MDI Utama
<pre>Option Explicit Private Sub MDIForm_Load() Me.imgIkon.ListImages.Remove "Pengetahuan" Me.imgIkon.ListImages.Remove "Penelusuran" Me.imgIkon.ListImages.Remove "Penjelasan" Me.imgIkon.ListImages.Clear Me.imgIkon.ListImages.Add , "Pengetahuan", LoadPicture(App.Path & "\\FOLDER05.ico") Me.imgIkon.ListImages.Add , "Penelusuran", LoadPicture(App.Path & "\\CRDFLE13.ico") Me.imgIkon.ListImages.Add , "Penjelasan", LoadPicture(App.Path & "\\CRDFLE04.ico") Me.Picture = LoadPicture(App.Path & "\\arini.JPG") App.HelpFile = App.Path & "\\Help.hlp" End Sub Private Sub MDIForm_Unload(Cancel As Integer) 'tutup FrmCiriKerusakan Unload FrmCiriKerusakan 'tutup FrmJenisKerusakan Unload FrmJenisKerusakan 'tutup FrmKeterangan 'tutup FrmMacamKerusakan Unload FrmKeterangan Unload FrmMacamKerusakan 'akhiri program Call mdlTutup.Tutup End End Sub Private Sub mnuAturan_Click() Load FrmAturan FrmAturan.SetFocus End Sub Private Sub mnuCiri_Click() 'Tampilkan FrmCiriKerusakan Load FrmCiriKerusakan FrmCiriKerusakan.SetFocus End Sub Private Sub mnuJenis_Click() 'Tampilkan FrmJenisKerusakan Load FrmJenisKerusakan</pre>

```
FrmJenisKerusakan.SetFocus
End Sub

Private Sub mnuKeluar_Click()
'tutup MDIUtama
Unload MDIUtama
End Sub

Private Sub mnuKeterangan_Click()
'Tampilkan FrmKeterangan
Load FrmKeterangan
FrmKeterangan.SetFocus
End Sub

Private Sub mnuMacam_Click()
'Tampilkan FrmMacamKerusakan
Load FrmMacamKerusakan
FrmMacamKerusakan.SetFocus
End Sub

Private Sub mnuManual_Click()
Me.cdHelp.HelpFile = App.Path & "\\Help.hlp"
Me.cdHelp.HelpCommand = cdHelpContext
Me.cdHelp.ShowHelp
End Sub

Private Sub mnuPenelusuran_Click()
'mendeklarasikan variabel i dengan tipe Integer
Dim i As Integer

'Tampilkan FrmPenelusuran
Load FrmPenelusuran
'tanda pilihan pada semua opsi FrmPenelusuran
For i = 0 To 2

Next i
FrmPenelusuran.SetFocus
End Sub

Private Sub mnuPenjelasan_Click()
Load FrmPenjelasan
FrmPenjelasan.cmdSimpan.Visible = True
FrmPenjelasan.SetFocus
End Sub

Private Sub mnuPenjelasan2_Click()
Load FrmPenjelasan
FrmPenjelasan.cmdSimpan.Visible = False
FrmPenjelasan.SetFocus
End Sub

Private Sub mnuSelesai_Click()
Call mnuKeluar_Click
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
Select Case Button.Key
Case "Penelusuran"
```

```
Call mnuPenelusuran_Click
Case "Penjelasan"
    If Me.mnuPakar.Visible = True Then
        Call mnuPenjelasan_Click
    Else
        Call mnuPenjelasan2_Click
    End If
End Select
End Sub

Private Sub Toolbar1_ButtonMenuClick(ByVal ButtonMenu As MSComctlLib.ButtonMenu)
Select Case ButtonMenu.Key
    Case "Macam"
        Call mnuMacam_Click
    Case "Jenis"
        Call mnuJenis_Click
    Case "Ciri"
        Call mnuCiri_Click
End Select
End Sub

*Frm Aturan

Option Explicit

Private Sub cmbMacam_Click()
Call Me.TampilkanJenis
End Sub

Private Sub cmdTambah_Click()
Load FrmTambahAturan
FrmTambahAturan.SetFocus
End Sub

Private Sub cmdTutup_Click()
Unload Me
End Sub

Private Sub Form_Activate()
Dim i As Integer

Me.cmbMacam.Clear
If tblMacam.RecordCount <> 0 Then
    tblMacam.MoveFirst
    For i = 1 To tblMacam.RecordCount
        Me.cmbMacam.AddItem tblMacam!nomacam & " " & tblMacam!macam
        tblMacam.MoveNext
    Next i
    Me.cmbMacam.ListIndex = 0
    Call Me.TampilkanJenis
End If
End Sub

Sub TampilkanJenis()
Dim Ambil As String
Dim Cari As String
Dim i As Integer
```

```
Me.lstJenis.Clear
'Me.txtGejala.Text = ""
Ambil = Mid(Me.cmbMacam.Text, 1, 4)
If tblRelasi1.RecordCount <> 0 Then
    tblRelasi1.MoveFirst
    For i = 1 To tblRelasi1.RecordCount
        If tblRelasi1!nomacam = Ambil Then
            Cari = tblRelasi1!nojenis
            tblJenis.Index = "idJenis"
            tblJenis.Seek "=", Cari
            Me.lstJenis.AddItem tblJenis!nojenis & " " & tblJenis!jenis
        End If
        tblRelasi1.MoveNext
        Call Me.TampilkanCiri
        If Me.lstJenis.ListCount <> 0 Then
            Me.lstJenis.ListIndex = 0
            Call lstJenis_Click
        End If
    Next i
End If
End Sub

Sub TampilkanCiri()
Dim Ambil As String
Dim Cari As String
Dim i As Integer

Me.txtDiagnosa.Text = ""
Me.lstCiri.Clear
Ambil = Mid(Me.lstJenis.Text, 1, 4)
If tblRelasi2.RecordCount <> 0 Then
    tblRelasi2.MoveFirst
    For i = 1 To tblRelasi2.RecordCount
        If tblRelasi2!nojenis = Ambil Then
            Cari = tblRelasi2!nociri
            tblCiri.Index = "idCiri"
            tblCiri.Seek "=", Cari
            Me.lstCiri.AddItem tblCiri!nociri & " " & tblCiri!ciri
        End If
        tblRelasi2.MoveNext
        If Me.lstCiri.ListCount <> 0 Then
            Me.lstCiri.ListIndex = 0
            Call lstCiri_Click
        End If
    Next i
End If
End Sub

Private Sub lstCiri_Click()
Dim Cari As String
Dim i As Integer

Cari = Mid(Me.lstCiri.Text, 1, 4)
tblCiri.Index = "idCiri"
tblCiri.Seek "=", Cari
Me.txtDiagnosa.Text = tblCiri!diagnosa
End Sub
```

```
Private Sub lstJenis_Click()  
Dim Cari As String  
Dim i As Integer  
  
Cari = Mid(Me.lstJenis.Text, 1, 4)  
tblJenis.Index = "idJenis"  
tblJenis.Seek "=", Cari  
Me.txtGejala.Text = tblJenis!gejala  
Call Me.TampilkanCiri  
End Sub
```

***FrmCiriKerusakan**

Option Explicit

```
Private Sub cmdAtas_Click()  
tblCiri.MoveFirst  
Call Me.TampilData  
End Sub
```

```
Private Sub cmdBawah_Click()  
tblCiri.MoveLast  
Call Me.TampilData  
End Sub
```

```
Private Sub cmdEdit_Click()  
Me.txtCiri.Locked = False  
Me.txtDiagnosa.Locked = False  
tblCiri.Edit  
End Sub
```

```
Private Sub cmdHapus_Click()  
If MsgBox("Apakah yakin akan menghapus data " & Me.txtCiri.Text & " ?", vbYesNo + vbQuestion,  
"Konfirmasi") = vbYes Then  
    tblCiri.Delete  
    Call cmdSebelum_Click  
    Call Me.Tampil_Grid  
End If  
End Sub
```

```
Private Sub cmdSebelum_Click()  
tblCiri.MovePrevious  
If tblCiri.BOF Then  
    tblCiri.MoveFirst  
End If  
Call Me.TampilData  
End Sub
```

```
Private Sub cmdSesudah_Click()  
tblCiri.MoveNext  
If tblCiri.EOF Then  
    tblCiri.MoveLast  
End If  
Call Me.TampilData  
End Sub
```

```
Private Sub cmdSimpan_Click()
```

```
tblCiri!nociri = Me.txtNomor.Text
tblCiri!ciri = Me.txtCiri.Text
tblCiri!diagnosa = Me.txtDiagnosa.Text
tblCiri.Update
Me.txtCiri.Locked = True
Me.txtDiagnosa.Locked = True
Call Me.Tampil_Grid
End Sub
```

```
Private Sub cmdTambah_Click()
Dim Ambil As String
```

```
If tblCiri.RecordCount <> 0 Then
    tblCiri.MoveLast
    Ambil = tblCiri!nociri
    Ambil = Right(Ambil, 3)
    Ambil = Val(Ambil) + 1001
    Ambil = Str(Ambil)
    Ambil = Right(Ambil, 3)
    Ambil = "C" & Ambil
    Me.txtNomor.Text = Ambil
    Me.txtCiri.Text = ""
    Me.txtDiagnosa.Text = ""
    Me.txtCiri.Locked = False
    Me.txtDiagnosa.Locked = False
    Me.txtCiri.SetFocus
End If
tblCiri.AddNew
End Sub
```

```
Private Sub cmdTutup_Click()
Unload Me
End Sub
```

```
Private Sub Form_Activate()
Call Me.Tampil_Grid
End Sub
```

```
Private Sub Form_Load()
Call cmdAtas_Click
End Sub
```

```
Sub TampilData()
Me.txtNomor.Text = tblCiri!nociri
Me.txtCiri.Text = tblCiri!ciri
Me.txtDiagnosa.Text = tblCiri!diagnosa
End Sub
```

```
Sub Tampil_Grid()
Dim i As Integer
```

```
Me.grdCiri.Refresh
Me.grdCiri.Refresh
```

```
Me.grdCiri.ColWidth(0) = 800
Me.grdCiri.ColWidth(1) = 4000
Me.grdCiri.ColWidth(2) = 0
```

```
Me.grdCiri.Row = 0
For i = 0 To Me.grdCiri.Cols - 1
    Me.grdCiri.Col = i
    Me.grdCiri.CellFontBold = True
    Me.grdCiri.CellAlignment = flexAlignCenterCenter
Next i
End Sub
```

***FrmJenisKerusakan**

Option Explicit

```
Private Sub cmdAtas_Click()
tblJenis.MoveFirst
Call Me.TampilData
End Sub
```

```
Private Sub cmdBawah_Click()
tblJenis.MoveLast
Call Me.TampilData
End Sub
```

```
Private Sub cmdEdit_Click()
Me.txtJenis.Locked = False
tblJenis.Edit
End Sub
```

```
Private Sub cmdHapus_Click()
If MsgBox("Apakah yakin akan menghapus data " & Me.txtJenis.Text & " ?", vbYesNo + vbQuestion,
"Konfirmasi") = vbYes Then
    tblJenis.Delete
    Call cmdSebelum_Click
    Call Me.Tampil_Grid
End If
End Sub
```

```
Private Sub cmdSebelum_Click()
tblJenis.MovePrevious
If tblJenis.BOF Then
    tblJenis.MoveFirst
End If
Call Me.TampilData
End Sub
```

```
Private Sub cmdSesudah_Click()
tblJenis.MoveNext
If tblJenis.EOF Then
    tblJenis.MoveLast
End If
Call Me.TampilData
End Sub
```

```
Private Sub cmdSimpan_Click()
tblJenis!nojenis = Me.txtNomor.Text
tblJenis!jenis = Me.txtJenis.Text
tblJenis!gejala = Me.txtGejala.Text
tblJenis.Update
Me.txtJenis.Locked = True
```



```
Call Me.Tampil_Grid
End Sub

Private Sub cmdTambah_Click()
Dim Ambil As String

If tblJenis.RecordCount <> 0 Then
    tblJenis.MoveLast
    Ambil = tblJenis!nojenis
    Ambil = Right(Ambil, 3)
    Ambil = Val(Ambil) + 1001
    Ambil = Str(Ambil)
    Ambil = Right(Ambil, 3)
    Ambil = "J" & Ambil
    Me.txtNomor.Text = Ambil
    Me.txtJenis.Text = ""
    'Me.txtGejala.Text = ""
    Me.txtJenis.Locked = False
    'Me.txtGejala.Locked = False
    Me.txtJenis.SetFocus
End If
tblJenis.AddNew
End Sub

Private Sub cmdTutup_Click()
Unload Me
End Sub

Private Sub Form_Activate()
Call Me.Tampil_Grid
End Sub

Sub Tampil_Grid()
Dim i As Integer

Me.grdJenis.Refresh
Me.grdJenis.Refresh

Me.grdJenis.ColWidth(0) = 800
Me.grdJenis.ColWidth(1) = 4000
'Me.grdJenis.ColWidth(2) = 0

Me.grdJenis.Row = 0
For i = 0 To Me.grdJenis.Cols - 1
    Me.grdJenis.Col = i
    Me.grdJenis.CellFontBold = True
    Me.grdJenis.CellAlignment = flexAlignCenterCenter
Next i
End Sub

Sub TampilData()
Me.txtNomor.Text = tblJenis!nojenis
Me.txtJenis.Text = tblJenis!jenis
End Sub

Private Sub Form_Load()
Call cmdAtas_Click
End Sub
```

***FrmKeterangan**

Option Explicit

```
Private Sub cmdTutup_Click()  
'Menutup form dan keluar dari program  
Unload FrmKeterangan  
'menghapus FrmKeterangan dari memori utama  
Set FrmKeterangan = Nothing  
End Sub
```

```
Private Sub Form_Load()  
Rem posisi kiri tampilan form pada layar monitor  
FrmKeterangan.Left = 0  
Rem posisi atas tampilan form pada layar monitor  
FrmKeterangan.Top = 0  
Rem mengatur lebar form menjadi 5000  
FrmKeterangan.Width = 7000  
Rem mengatur tinggi form menjadi 3600  
FrmKeterangan.Height = 5000  
End Sub
```

```
Private Sub Timer1_Timer()  
Me.lblSelamat.Caption = Mid(Me.lblSelamat.Caption, 2, Len(Me.lblSelamat.Caption)) +  
Mid(Me.lblSelamat.Caption, 1, 1)
```

```
If Me.lblPakar.ForeColor = vbWindowText Then  
    Me.lblPakar.ForeColor = vbHighlightText  
Else  
    Me.lblPakar.ForeColor = vbWindowText  
End If  
End Sub
```

***FrmMacamKerusakan**

Option Explicit

```
Private Sub cmdEdit_Click()  
Dim EditData As String  
Dim i As Integer  
Dim Kodenya As String  
Dim Datanya As String
```

```
Kodenya = Mid(Me.lstMacam.Text, 1, 4)  
Datanya = Mid(Me.lstMacam.Text, 8, Len(Me.lstMacam.Text))
```

```
If Datanya <> "" Then  
    EditData = InputBox("Masukkan data yang baru dari data :", "Konfirmasi", Datanya)  
    If EditData <> "" Then  
        'menentukan indeks  
        tblMacam.Index = "idMacam"  
        'melakukan pencarian data  
        tblMacam.Seek "=", Kodenya  
        'menghapus data dari tabel  
        tblMacam.Edit  
        tblMacam!macam = EditData  
        tblMacam.Update
```

```
Call Form_Load
End If
End If
End Sub

Private Sub cmdHapus_Click()
Dim Hapus As String
Dim i As Integer

'mengambil kode macam kerusakan
Hapus = Mid(Me.lstMacam.Text, 1, 4)
If Hapus <> "" Then
    If MsgBox("Apakah yakin akan menghapus data " & Hapus & " ?", vbYesNo + vbQuestion,
"Konfirmasi") = vbYes Then
        'menentukan indeks
        tblMacam.Index = "idMacam"
        'melakukan pencarian data
        tblMacam.Seek "=", Hapus
        'menghapus data dari tabel
        tblMacam.Delete
        Call Form_Load
    End If
End If
End Sub

Private Sub cmdTambah_Click()
'mendeklarasikan variabel Jawab dengan tipe Integer
Dim Jawab As Integer
Dim Ambil As String

If Me.txtMacam.Text <> "" Then
    'mengambil data terakhir pada lstMacam dan disimpan dalam Ambil
    Ambil = Me.lstMacam.List(Me.lstMacam.ListCount - 1)
    'mengambil empat digit karakter terdepan dari variabel Ambil
    Ambil = Mid(Ambil, 1, 4)
    'mengambil 3 digit karakter terakhir dari variabel ambil
    Ambil = Right(Ambil, 3)
    'melakukan penjumlahan dengan angka 1001 untuk mendapatkan nomor baru
    Ambil = Val(Ambil) + 1001
    'mengkonversikan hasil penjumlahan menjadi bentuk string
    Ambil = Str(Ambil)
    'mengambil 3 digit terakhir dari variabel Ambil
    Ambil = Right(Ambil, 3)
    'menambahkan bariabel Ambil dengan kode M
    Ambil = "M" & Ambil

    'menyimpan data ke dalam tblMacam
    tblMacam.MoveLast
    tblMacam.AddNew
    tblMacam!nomacam = Ambil
    tblMacam!macam = Me.txtMacam.Text
    tblMacam.Update

    Me.lstMacam.AddItem Me.txtMacam.Text
    Me.txtMacam.Text = ""
    Me.txtMacam.SetFocus
Else
    Jawab = MsgBox("Anda belum menginputkan macam kerusakan !" & vbNewLine & "Silahkan
```

```
menginputkan macam kerusakan terlebih dahulu !", vbOKOnly + vbCritical, "Konfirmasi")
    If Jawab = vbOK Then
        Me.txtMacam.SetFocus
    End If
End If
End Sub

Private Sub cmdTutup_Click()
Unload Me
End Sub

Private Sub Form_Load()
Dim i As Integer

Me.lstMacam.Clear
tblMacam.MoveFirst
If tblMacam.RecordCount <> 0 Then
    For i = 1 To tblMacam.RecordCount
        Me.lstMacam.AddItem tblMacam!nomacam & " " & tblMacam!macam
        tblMacam.MoveNext
    Next i
End If
End Sub

*FrmPassword

Option Explicit

Private Sub cmdGanti_Click()
If Me.cmdGanti.Caption = "Ganti" Then
    If (Me.txtNama.Text = tblPasswd!nama) And (Me.txtPasswd.Text = tblPasswd!passwd) Then
        MsgBox "Silahkan memasukkan nama dan password baru !", vbOKOnly + vbInformation,
        "Konfirmasi"
        Me.txtNama.Text = ""
        Me.txtPasswd.Text = ""
        Me.txtNama.SetFocus
        Me.cmdGanti.Caption = "Simpan"
    Else
        MsgBox "Masukkan terlebih dahulu nama dan password Anda yang lama dengan benar",
        vbOKOnly + vbInformation, "Konfirmasi"
        Exit Sub
    End If
Else
    If (Me.txtNama.Text <> "") And (Me.txtPasswd.Text <> "") Then
        tblPasswd.MoveFirst
        tblPasswd.Edit
        tblPasswd!nama = Me.txtNama.Text
        tblPasswd!passwd = Me.txtPasswd.Text
        tblPasswd.Update
        MsgBox "Nama dan password Anda yang baru siap digunakan !", vbOKOnly + vbInformation,
        "Konfirmasi"
        Me.txtNama.Text = ""
        Me.txtPasswd.Text = ""
        Me.txtNama.SetFocus
        Me.cmdGanti.Caption = "Ganti"
    Else
        MsgBox "Masukkan terlebih dahulu nama dan password Anda dengan benar", vbOKOnly +
        vbInformation, "Konfirmasi"
```

```
End If
End If
End Sub

Private Sub cmdLogin_Click()
If Me.optPilihan(0).Value = True Then
    MDIUtama.mnuPakar.Visible = False
    MDIUtama.Toolbar1.Buttons(1).Visible = False
    MDIUtama.StatusBar1.Panels(3).Text = "Pemakai"
Else
    If (Me.txtNama.Text = tblPasswd!nama) And (Me.txtPasswd.Text = tblPasswd!passwd) Then
        MDIUtama.mnuPemakai.Visible = False
        MDIUtama.Toolbar1.Buttons(2).Visible = False
        MDIUtama.StatusBar1.Panels(3).Text = "Pakar"
    Else
        MsgBox "Password Anda Salah !", vbOKOnly + vbCritical, "Konfirmasi"
        Me.txtNama.Text = ""
        Me.txtPasswd.Text = ""
        Me.txtNama.SetFocus
        Exit Sub
    End If
End If
MDIUtama.Show
MDIUtama.SetFocus
End Sub

Private Sub Form_Load()
Call mdlBuka.Buka
End Sub

Private Sub FrmTutup_Click()
End
End Sub

Private Sub optPilihan_Click(Index As Integer)
If Index = 1 Then
    Me.txtNama.SetFocus
End If
End Sub

Private Sub txtNama_Change()
Me.txtNama.SelStart = Len(Me.txtNama.Text)
End Sub

Private Sub txtNama_GotFocus()
Me.optPilihan(1).Value = True
End Sub

Private Sub txtNama_KeyPress(KeyAscii As Integer)
Dim strValid As String
'mendefinisikan validitas inputan
strValid = "abcdefghijklmnopqrstuvwxyz"
strValid = strValid & "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

'Jika tombol Enter ditekan maka pindahkan fokus ke txtPasswd
If KeyAscii = vbKeyReturn Then
    Me.txtPasswd.SetFocus
Else
```

```
If InStr(strValid, Chr(KeyAscii)) = 0 Then
    KeyAscii = 0
End If
End If
End Sub

Private Sub txtPasswd_Change()
If Len(Me.txtPasswd.Text) = 8 Then
    Me.cmdLogin.SetFocus
End If
End Sub

Private Sub txtPasswd_KeyPress(KeyAscii As Integer)
Dim strValid As String
'mendefinisikan validitas inputan
strValid = "0123456789"

If InStr(strValid, Chr(KeyAscii)) = 0 Then
    KeyAscii = 0
End If
End Sub

*FrmPenelusuran

Option Explicit

Private Sub cmdProses_Click()
'Tampilkan FrmPenelusuranMacam
Load FrmPenelusuranMacam
FrmPenelusuranMacam.SetFocus
End Sub

*FrmPenjelasan

Option Explicit

Private Sub cmdProses_Click()
'Tampilkan FrmPenelusuranMacam
Load FrmPenelusuranMacam
FrmPenelusuranMacam.SetFocus
End Sub

*FrmRekamanData

Option Explicit

Private Sub cmdProses_Click()
Dim Ambil As String
Ambil = Mid(FrmPenelusuranCiri.lstCiri.Text, 1, 4)
tblCiri.Index = "idCiri"
tblCiri.Seek "=", Ambil
Load FrmAnimasi
End Sub

Private Sub Form_Activate()
Dim Keterangan As String
Keterangan = "Sistem sudah merekam data yang Anda pilih yaitu :"
Keterangan = Keterangan & vbCrLf
```

```
Keterangan = Keterangan & "Macam Kerusakan Komputer, Jenis dan Cirinya"
Keterangan = Keterangan & vbNewLine
Keterangan = Keterangan & "Data yang terekam berturut-turut adalah sebagai berikut : "
Keterangan = Keterangan & vbNewLine
Keterangan = Keterangan & FrmPenelusuranMacam.cmbPenelusuranMacam.Text
Keterangan = Keterangan & vbNewLine
Keterangan = Keterangan & FrmPenelusuranJenis.lstJenis.Text
Keterangan = Keterangan & vbNewLine
Keterangan = Keterangan & FrmPenelusuranCiri.lstCiri.Text
Me.txtRekaman.Text = Keterangan
End Sub
```

***FrmSolusi**

Option Explicit

```
Private Sub cmdSelesai_Click()
Unload FrmRekamanData
Unload FrmPenelusuranMacam
Unload FrmPenelusuranJenis
Unload FrmPenelusuranCiri
Unload FrmPenelusuran
Unload Me
End Sub
```

***FromTambahAturan**

Option Explicit

```
Sub TampilJenis()
Dim Ambil As String
Dim i As Integer
Dim j As Integer

For j = 0 To Me.lstJenis.ListCount - 1
    Me.lstJenis.Selected(j) = False
Next j

Ambil = Mid(Me.cmbMacam.Text, 1, 4)
tblRelasi1.MoveFirst
For i = 1 To tblRelasi1.RecordCount
    If tblRelasi1!nomacam = Ambil Then
        For j = 0 To Me.lstJenis.ListCount - 1
            If Mid(Me.lstJenis.List(j), 1, 4) = tblRelasi1!nojenis Then
                Me.lstJenis.Selected(j) = True
            End If
        Next j
    End If
    tblRelasi1.MoveNext
Next i
End Sub

Private Sub cmbMacam_Click()
Call Me.TampilJenis
End Sub

Private Sub cmdSimpanJenis_Click()
Dim Ambil As String
```



```
Dim i As Integer
'menghapus aturan yang lama
Ambil = Mid(Me.lstJenis.Text, 1, 4)
If tblRelasi2.RecordCount <> 0 Then
    tblRelasi2.MoveFirst
    For i = 1 To tblRelasi2.RecordCount
        If tblRelasi2!nojenis = Ambil Then
            tblRelasi2.Delete
        End If
        tblRelasi2.MoveNext
    Next i
End If
'menambahkan aturan yang baru
For i = 0 To Me.lstCiri.ListCount - 1
    If Me.lstCiri.Selected(i) = True Then
        tblRelasi2.AddNew
        tblRelasi2!nojenis = Ambil
        tblRelasi2!nociri = Mid(Me.lstCiri.List(i), 1, 4)
        tblRelasi2.Update
    End If
Next i
End Sub

Private Sub cmdSimpanMacam_Click()
Dim Ambil As String
Dim i As Integer
'menghapus aturan yang lama
Ambil = Mid(Me.cmbMacam.Text, 1, 4)
If tblRelasi1.RecordCount <> 0 Then
    tblRelasi1.MoveFirst
    For i = 1 To tblRelasi1.RecordCount
        If tblRelasi1!nomacam = Ambil Then
            tblRelasi1.Delete
        End If
        tblRelasi1.MoveNext
    Next i
End If
'menambahkan aturan yang baru
For i = 0 To Me.lstJenis.ListCount - 1
    If Me.lstJenis.Selected(i) = True Then
        tblRelasi1.AddNew
        tblRelasi1!nomacam = Ambil
        tblRelasi1!nojenis = Mid(Me.lstJenis.List(i), 1, 4)
        tblRelasi1.Update
    End If
Next i
End Sub

Private Sub cmdTutup_Click()
Unload Me
End Sub

Private Sub Form_Activate()
Dim i As Integer

Me.cmbMacam.Clear
If tblMacam.RecordCount <> 0 Then
    tblMacam.MoveFirst
```

```
For i = 1 To tblMacam.RecordCount
    Me.cmbMacam.AddItem tblMacam!nomacam & " " & tblMacam!macam
    tblMacam.MoveNext
Next i
Me.cmbMacam.ListIndex = 0
End If

Me.lstJenis.Clear
If tblJenis.RecordCount <> 0 Then
    tblJenis.MoveFirst
    For i = 1 To tblJenis.RecordCount
        Me.lstJenis.AddItem tblJenis!nojenis & " " & tblJenis!jenis
        tblJenis.MoveNext
    Next i
    Me.lstJenis.ListIndex = 0
End If

Me.lstCiri.Clear
If tblCiri.RecordCount <> 0 Then
    tblCiri.MoveFirst
    For i = 1 To tblCiri.RecordCount
        Me.lstCiri.AddItem tblCiri!nociri & " " & tblCiri!ciri
        tblCiri.MoveNext
    Next i
    Me.lstCiri.ListIndex = 0
End If

Call Me.TampilJenis
Call Me.TampilCiri
End Sub

Sub TampilCiri()
    Dim Ambil As String
    Dim i As Integer
    Dim j As Integer

    For j = 0 To Me.lstCiri.ListCount - 1
        Me.lstCiri.Selected(j) = False
    Next j

    Ambil = Mid(Me.lstJenis.Text, 1, 4)
    tblRelasi2.MoveFirst
    For i = 1 To tblRelasi2.RecordCount
        If tblRelasi2!nojenis = Ambil Then
            For j = 0 To Me.lstCiri.ListCount - 1
                If Mid(Me.lstCiri.List(j), 1, 4) = tblRelasi2!nociri Then
                    Me.lstCiri.Selected(j) = True
                End If
            Next j
            'tblRelasi2.MoveNext
        End If
        tblRelasi2.MoveNext
    Next i
End Sub

Private Sub lstJenis_Click()
    Call Me.TampilCiri
End Sub
```

