SISTEM PAKAR MENDIAGNOSA KERUSAKAN UPS (*UNINTERRUPTIBLE POWER SUPPLY)* BERBASIS WEB DENGAN METODE *FORWARD CHAINING* DAN *BACKWARD CHAINING*

SKRIPSI

Oleh:
Vorwi Kwannahagia
130210001

PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017

SISTEM PAKAR MENDIAGNOSA KERUSAKAN UPS (*UNINTERRUPTIBLE POWER SUPPLY)* BERBASIS WEB DENGAN METODE *FORWARD CHAINING* DAN *BACKWARD CHAINING*

SKRIPSI

Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana

Oleh:
Vorwi Kwannahagia
130210001

PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017

PERNYATAAN

Dengan ini menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, dan/atau magister), baik di Universitas Putera Batam maupun di perguruan tinggi lain.
2. Skripsi ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing.
3. Dalam skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta saksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam,14 Februari 2017

Yang membuat pernyataan,

Vorwi Kwannahagia

130210001

SISTEM PAKAR MENDIAGNOSA KERUSAKAN UPS (*UNINTERRUPTIBLE POWER SUPPLY)* BERBASIS WEB DENGAN METODE *FORWARD CHAINING* DAN *BACKWARD CHAINING*

Oleh:
Vorwi Kwannahagia
130210001

SKRIPSI
Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana

Telah disetujui oleh Pembimbing pada tanggal
seperti tertera dibawah ini

Batam,14 Februari 2017

Realize, S.Kom., M.SI.
Pembimbing

ABSTRAK

Uninterruptible power supply (UPS) merupakan komponen tambahan yang sangat penting pada suatu komputer, banyak kasus UPS rusak dikarenakan sesuatu hal yang tidak kita ketahui sehingga kita harus pergi ke teknisi/ahli untuk menemukan solusi dari kasus kerusakan yang kita dialami sehingga menghabiskan banyak waktu, biaya dan jarak. Oleh karena itu dibutuhkan sebuah sistem pakar yang mempunyai pengetahuan yang dimiliki seorang pakar untuk membantu menyelesaikan masalah itu secara cepat sehingga pengguna tidak perlu lagi pergi ke teknisi ataupun ahli. Sistem pakar ini menggunakan metode gabungan yaitu dengan menggunakan *forward chaining* yaitu penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis, dan metode *backward chaining* yaitu penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan. Sistem ini dibuat berbasis *web* dengan menggunakan pemrograman PHP dan basis data menggunakan MySQL Database. Sistem ini dilengkapi menu pakar untuk pengelolaan pengetahuan, sehingga pakar dapat menambahkan, mengubah, dan menghapus pengetahuan. Hasil Penelitian menunjukkan bahwa metode *forward chaining* dan *backward chaining* tepat untuk diagnosa kerusakan UPS, karena pada kenyataan, penalaran seorang pakar kerusakan UPS mengarah pada gejala-gejala yang dialami oleh kerusakan UPS kemudian diperoleh jenis kerusakan UPS. Kesimpulan yang bisa diambil adalah dengan sistem pakar ini dapat mewakili seorang pakar dalam mendiagnosa kerusakan UPS sehingga masyarakat lebih menghemat waktu, biaya dan jarak karena tanpa perlu data ke seorang teknisi ataupun ahli.

Kata kunci: Sistem Pakar, *Uninterruptible Power Supply* (UPS), *PHP*, *MySQL*, *Web, Forward Chaining, Backward Chaining.*

ABSTRACT

Uninterruptible power supply (UPS) is an additional component that is very important on a computer, many cases of UPS damaged due to any of the things that we do not know that we must go to the technicians/experts to find the solution of the case of damage that we experience so that it can be time and cost consuming. Therefore needed a system of experts who have knowledge that belongs to an expert to help resolve the problem quickly so the user does not need to go to the technicians/experts. This expert system using combined methods are forward chaining namely the logic started from the fact of the first to test the truth of the hypothesis and backward chaining method that is the logic of starting from the first hypothesis and to test the truth of the hypothesis must search for the facts that exist in the knowledge base. This system is created using PHP programming and data base using MySQL Database. This system is equipped with the menu of experts to knowledge management, so that experts can add, edit, and delete the knowledge. The results of the study showed that the forward chaining and backward chainging method right for diagnosis damage UPS, because in reality the logic of an expert damage UPS pointing at the symptoms experienced by the damage to the UPS then obtained the type of damage UPS. The conclusion to be drawn is with this expert system can represent an expert in diagnose damage UPS so that the community more save time and cost because without the need to datake to an expert.

Key Words: *Expert System, uninterruptible power supply (UPS), PHP, MySQL, Web, Forward Chaining, Backward chaining.*

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan karuniaNya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang merupakan salah satu persyaratan untuk menyelesaikan program studi strata satu (S1) pada Program Studi Teknik Informatika Universitas Putera Batam.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Karena itu, kritik dan saran akan senantiasa penulis terima dengan senang hati.

Dengan segala keterbatasan, penulis menyadari pula bahwa skripsi ini takkan terwujud tanpa bantuan, bimbingan, dan dorongan dari berbagai pihak. Untuk itu, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Rektor Universitas Putera Batam
2. Ketua Program Studi Teknik Informatika Universitas Putera Batam
3. Ibu Realize, S.Kom., M.SI. selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam
5. Kepada orang tua penulis, yang terus mendoakan keberhasilan penulis menyelesaikan skripsi ini.
6. Teman-teman seperjuangan yang juga selalu memberikan motivasi baik berupa *sharing* pendapat, motivasi dan hal-hal lainnya dalam rangka pembuatan skripsi ini.
7. Serta semua pihak yang tak dapat penulis sebutkan satu persatu yang telah membantu dalam penyusunan skripsi ini.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu mencurahkan hidayah serta taufik-Nya, Amin.

Batam, 14 Februari 2017

Penulis

DAFTAR ISI

Halaman

HALAMAN PERNYATAAN ..... **i**
HALAMAN PENGESAHAN ..... **ii**
ABSTRAK ..... **iii**
ABSTRACT ..... **iv**
KATA PENGANTAR ..... **v**
DAFTAR ISI ..... **vivii**
DAFTAR TABEL ..... **ixx**
DAFTAR GAMBAR ..... **x**
BAB I PENDAHULUAN
1.1 Latar Belakang 1
1.2 Identifikasi Masalah 3
1.3 Batasan Masalah 4
1.4 Rumusan Masalah 4
1.5 Tujuan Penelitian 5
1.6 Manfaat Penelitian 5
BAB II LANDASAN TEORI
2.1 Teori Dasar 6
2.1.1 Kecerdasan Buatan (Artificial Intelligence) 7
2.1.1.3 Sistem Pakar (*Expert System*) 7
2.1.1.3.2 Kelebihan Sistem Pakar 9
2.1.1.3.3 Struktur Sistem Pakar 11
2.1.1.3.4 Karakteristik Sistem Pakar 14
2.1.1.3.5 Metode Inferensi 155
2.1.1.3.5.1 Runut Maju (*Forward Chaining*) 155
2.1.1.3.5.2 Runut Balik (*Backward Chaining*) 166
2.1.2 Tabel Keputusan 166
2.1.3 Pohon Kuputusan 166
2.1.4 UML (*Unified Modeling Language*) 19
2.1.4.1 *Use Case Diagram* 19
2.1.4.2 *Activity* Diagram 21
2.1.4.3 *Sequence Diagram* 23
2.1.4.4 *Class Diagram* 25
2.2 Variabel 27
2.2.1 *Uninterruptible Power Supply* (UPS) 27
2.3 *Software* Pendukung 28
2.3.1 Bahasa Pemograman PHP 28
2.3.2 Aplikasi *Adobe Dreamweaver* 29

2.3.3 *MySQL Database* 30
2.3.4 *WampServer* 331
2.3.5 StarUML 331
2.4 Penelitian Terdahulu 32
2.5 Kerangka Berpikir..... 34
BAB III METODE PENELITIAN
3.1 Desain Penelitian 35
3.2 Operasional Variabel 37
3.2.1 Metode Forward Chaining 37
3.2.2 Metode Backward Chaining..... 441
3.3 Teknik Pengumpulan Data..... 46
3.4 Perancangan Sistem 47
3.4.1 UML (Unified Modelling Language) 47
3.4.1.1 *Use Case Diagram* 47
3.4.1.2 *Activity* Diagram..... 48
3.4.1.3 *Sequence Diagram* 50
3.4.1.4 *Class Diagram* 57
3.4.2 Desain Basis Data 58
3.4.3 Desain Antarmuka..... 60
3.5 Lokasi Dan Jadwal Penelitian..... 74
BAB IV HASIL PENELITIAN DAN PEMBAHASAN
4.1 Hasil Penelitian..... 75
4.1.1 Implementasi Antar Muka..... 75
4.2 Pembahasan 89
4.2.1 Pengujian Validasi 89
BAB V KESIMPULAN DAN SARAN
5.1 Kesimpulan 991
5.2 Saran 991
DAFTAR PUSTAKA..... 92
RIWAYAT HIDUP
SURAT KETERANGAN PENELITIAN
LAMPIRAN

DAFTAR TABEL

Halaman

Tabel 2.1 Simbol Use Case Diagram 20
Tabel 2.2 Simbol Activity Diagram 22
Tabel 2.3 Simbol Sequence Diagram 24
Tabel 2.4 Simbol Class Diagram 26
Tabel 3.1 Indikator Kerusakan UPS (UD) 37
Tabel 3.2 Kode Indikator Kerusakan UPS (UD) 38
Tabel 3.3 Kode Penyebab Kerusakan UPS (UD) 38
Tabel 3.4 Data Solusi Kerusakan UPS (UD) 39
Tabel 3.5 Tabel Keputusan Kerusakan UPS (UD) 39
Tabel 3.6 Indikator Penyebab Kerusakan UPS (PD) 41
Tabel 3.7 Kode Indikator Kerusakan UPS (PD) 42
Tabel 3.8 Kode Komponen Kerusakan UPS (PD) 42
Tabel 3.9 Data Solusi Kerusakan UPS (PD) 43
Tabel 3.10 Tabel Keputusan (PD) 44
Tabel 3.11 Tabel Basis Data *Admin* 58
Tabel 3.12 Tabel Basis Data Kerusakan 58
Tabel 3.13 Tabel Basis Data Pertanyaan 59
Tabel 3.14 Tabel Basis Data Solusi 59
Tabel 4.1 Hasil Pengujian Validasi 90

DAFTAR GAMBAR

Halaman

Gambar 2.1 Struktur Sistem Pakar 11
Gambar 2.2 Contoh Pohon Keputusan 18
Gambar 2.3 Logo php 29
Gambar 2.4 Tampilan Adobe Dreamweaver CS6 29
Gambar 2.5 Tampilan Database MySQL 30
Gambar 2.6 Logo WampServer 31
Gambar 2.7 Logo StarUML 32
Gambar 2.8 Kerangka Berpikir Peneliti 34
Gambar 3.1 Desain Penelitian 36
Gambar 3.2 Pohon Keputusan (UD) 40
Gambar 3.3 Pohon Keputusan (PD) 45
Gambar 3.4 Use Case Diagram 47
Gambar 3.5 Activity Diagram Admin 49
Gambar 3.6 Activity Diagram User 49
Gambar 3.7 *Sequence Diagram Login* Admin 50
Gambar 3.8 Sequence Diagram Input Data Kerusakan 51
Gambar 3.9 Sequence Diagram Edit Data Kerusakan 52
Gambar 3.10 Sequence Diagram Input Data Pertanyaan Kerusakan 53
Gambar 3.11 *Sequence Diagram Edit* Data Pertanyaan Kerusakan 54
Gambar 3.12 Sequence Diagram Input Data Solusi Kerusakan 55
Gambar 3.13 *Sequence Diagram Edit* Data Solusi Kerusakan 56
Gambar 3.14 Class Diagram 57
Gambar 3.15 Form Login Aplikasi 60
Gambar 3.16 Menu Utama 60
Gambar 3.16 Menu Utama Admin 61
Gambar 3.17 Menu Diagnosa Kerusakan 61
Gambar 3.18 Menu Diagnosa Kerusakan Pertanyaan 62
Gambar 3.19 Menu Diagnosa Kerusakan Solusi 62
Gambar 3.20 Menu Diagnosa Penyebab 63
Gambar 3.21 Menu Diagnosa Penyebab Pertanyaan 63
Gambar 3.22 Menu Diagnosa Penyebab Solusi 64
Gambar 3.23 Menu Data Jenis Kerusakan 64
Gambar 3.24 MenuTambah Data Jenis Kerusakan 65
Gambar 3.25 Menu *Edit* Data Jenis Kerusakan 65
Gambar 3.26 Menu Data Pertanyaan Kerusakan 66
Gambar 3.27 Menu Tambah Data Pertanyaan Kerusakan 66
Gambar 3.28 Menu *Edit* Data Pertanyaan Kerusakan 67
Gambar 3.29 Menu Data Solusi Kerusakan 67

Gambar 3.30 Menu Tambah Data Solusi Kerusakan 68
Gambar 3.31 Menu *Edit* Data Solusi Kerusakan..... 68
Gambar 3.32: Menu Data Jenis Penyebab..... 69
Gambar 3.33 Menu Tambah Data Jenis Penyebab..... 69
Gambar 3.34 Menu *Edit* Data Jenis Penyebab 70
Gambar 3.35 Menu Data Pertanyaan Penyebab 70
Gambar 3.27 Menu Tambah Data Pertanyaan Penyebab 71
Gambar 3.28 Menu *Edit* Data Pertanyaan Penyebab 71
Gambar 3.29 Menu Data Solusi Penyebab..... 72
Gambar 3.30 Menu Tambah Data Solusi Penyebab..... 72
Gambar 3.31 Menu *Edit* Data Solusi Penyebab 73
Gambar 3.31 Menu Ganti *Password* 73
Gambar 4.1 Tampilan Menu Utama..... 75
Gambar 4.2 Form Login..... 76
Gambar 4.3 Tampilan Menu Data Jenis Kerusakan..... 76
Gambar 4.4 Tampilan Tambah Data Jenis Kerusakan 77
Gambar 4.5 Tampilan Edit Data Jenis Kerusakan..... 77
Gambar 4.6 Tampilan Menu Data Pertanyaan Kerusakan 78
Gambar 4.7 Tampilan Tambah Data Pertanyaan Kerusakan 78
Gambar 4.8 Tampilan Edit Data Pertanyaan Kerusakan..... 79
Gambar 4.9 Tampilan Data Solusi Kerusakan 79
Gambar 4.10 Tampilan Tambah Data Solusi Kerusakan 80
Gambar 4.11 Tampilan Edit Data Solusi Kerusakan..... 80
Gambar 4.12 Tampilan Menu Data Jenis Penyebab..... 81
Gambar 4.13 Tampilan Tambah Data Jenis Penyebab..... 81
Gambar 4.14 Tampilan Edit Data Jenis Penyebab 82
Gambar 4.15 Tampilan Menu Data Pertanyaan Penyebab..... 82
Gambar 4.16 Tampilan Tambah Data Pertanyaan Penyebab 83
Gambar 4.17 Tampilan Edit Data Pertanyaan Penyebab 83
Gambar 4.18 Tampilan Data Solusi Penyebab..... 84
Gambar 4.19 Tampilan Tambah Data Solusi Penyebab..... 84
Gambar 4.20 Tampilan Edit Data Solusi Penyebab 85
Gambar 4.21 Tampilan Ganti Password..... 85
Gambar 4.22 Tampilan Diagnosa Kerusakan..... 86
Gambar 4.23 Tampilan Pertanyaan Diagnosa Kerusakan..... 86
Gambar 4.24 Tampilan Solusi Diagnosa Kerusakan..... 87
Gambar 4.25 Tampilan Diagnosa Penyebab 87
Gambar 4.26 Tampilan Pertanyaan Diagnosa Penyebab..... 88
Gambar 4.27 Tampilan Solusi Diagnosa Penyebab 88
Gambar 4.28 Tampilan Ubah Password..... 89

DAFTAR LAMPIRAN

LAMPIRAN I SOURCE CODE
LAMPIRAN II SURAT PERNYATAAN PENELITIAN
LAMPIRAN III TRANSKRIP WAWANCARA

BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan berkembangnya perangkat-perangkat elektronik yang ada, listrik merupakan salah satu hal utama yang bisa membuat suatu perangkat bekerja sesuai fungsinya. Namun terjadinya pemadaman listrik dengan tiba-tiba merupakan hal yang tidak bisa kita hindari saat kita sedang mengunakan perangkat elektronik. Pekerjaan kita bisa terhenti dan data juga bisa hilang bila tidak melakukan *back up* sebelumnya karena terjadinya pemadaman listrik secara tiba-tiba. Berkaitan dengan hal tersebut, diperlukan sebuah alat yang dapat menyimpan daya listrik yang berasal dari perusahaan listrik negara (PLN) untuk disalurkan ke peralatan elektronik apabila terjadi pemadaman dengan tiba-tiba.

Uninterruptible Power Supply (UPS) merupakan sistem penyedia daya listrik ketika suatu peralatan elektronik kehilangan energi dari sumber utamanya. UPS merupakan sumber tenaga alternatif sementara yang menggantikan supply tenaga listrik utama, dalam hal ini sumber listrik dari PLN. UPS sendiri merupakan sebuah sistem yang berdiri sendiri tanpa sistem supply tenaga listrik dari PLN (Bawotong et al. 2015, p.1).

UPS bekerja diantara perangkat elektronik dan colokan listrik. Dari colokan listrik, arus *alternating current* (AC) diubah menjadi *direct current* (DC) dan dialirkan ke baterai yang berada pada UPS. Arus DC dari baterai akan diubah

menjadi arus AC oleh inverter. Listrik yang disimpan pada baterai akan dipakai ketika sumber energi utama terputus. Penggunaan UPS banyak ditemukan di perusahaan, perkantoran dan tempat lainnya tetapi pengetahuan atas kerusakan UPS masih terhitung sedikit. Keterbatasan teknisi UPS juga masih menjadi faktor bagi pengguna dan belum terdapat sebuah sistem yang mampu membantu pengguna dan teknisi baru untuk mengetahui kerusakan UPS.

Kecerdasan buatan atau *Artificial Intelligence* (AI) merupakan bagian dari ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Sistem pkakar merupakan salah satu bagian dari kecerdasan buatan. Sistem pakar adalah sistem yang menggunakan pengetahuan manusia yang dimasukkan ke dalam komputer untuk memecahkan masalah-masalah yang biasanya diselesaikan oleh pakar. Permasalahan yang ditangani oleh seorang pakar bukan hanya permasalahan yang mengandalkan algoritma namun terkadang juga permasalahan yang sulit dipahami. Oleh karena itu sistem pakar dibangun bukan berdasarkan algoritma tertentu tetapi berdasarkan basis pengetahuan dan aturan. Sesuai dengan kemampuan dari sistem pakar yang merupakan salah satu cabang dari ilmu kecerdasan buatan yaitu mampu meniru penalaran dari seorang pakar dari bidang ilmu tertentu. Berdasarkan alur pelacakan solusi, sistem pakar memiliki metode *forward chaining* yaitu penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis, dan metode *backward chaining* yaitu penalaran dimulai dari hipotesis terlebih dahulu, dan untuk menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan.

Penelitian yang akan dilakukan adalah terhadap kerusakan UPS. Seperti yang dimaksudkan tadi, pengetahuan terhadap kerusakan UPS masih terhitung sedikit. Maka berdasarkan permasalahan diatas, dibutuhkan suatu aplikasi sistem pakar untuk mendeteksi terdahulu permasalahan yang ada didalam UPS tersebut agar pengguna dapat melakukan reparasi secara tersendiri ataupun dapat membantu para teknisi memperbaiki alat tersebut. Hal inilah yang mendorong penulis untuk melakukan penelitian dengan judul "**SISTEM PAKAR MENDIAGNOSA KERUSAKAN UPS (*UNINTERRUPTIBLE POWER SUPPLY)* DENGAN METODE *FORWARD CHAINING* DAN *BACKWARD CHAINING***".

1.2 Identifikasi Masalah

Berdasarkan uraian-uraian diatas, maka peneliti menguraikan permasalahan yang ada sebagai berikut:

1. Kurangnya pengetahuan mengenai kerusakan yang terjadi pada UPS.
2. Belum adanya suatu sistem yang membantu pengguna dan teknisi baru untuk mengetahui kerusakan pada UPS.
3. Sedikitnya teknisi atau ahli dalam bidang ini.

1.3 Batasan Masalah

Agar pembahasan terarah dan sesuai dengan yang diharapkan, mengingat banyaknya ruang lingkup permasalahan yang ada dan keterbatasan informasi yang dimiliki oleh peneliti, maka peneliti membuat batasan masalah. Adapun pembatasan masalah dan ruang lingkup dalam penelitian ini adalah sebagai berikut:

1. Sistem ini hanya untuk mendiagnosa kerusakan UPS dengan merk ICA.
2. Metode pendeteksi masalah kerusakan UPS yang digunakan adalah metode *Forward Chaining* dan *Backward Chaining*.
3. Aplikasi ini sebagai diagnosa awal kerusakan UPS sebelum melakukan servis pada UPS.
4. Sistem pakar dirancang berbasis *web* dengan menggunakan bahasa pemograman PHP dan MySQL *database.*
5. Data diambil dengan melakukan wawancara dengan teknisi ataupun ahli dan dari referensi buku dan jurnal.

1.4 Rumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah :

Bagaimana implementasi sistem pakar mendiagnosa kerusakan UPS dalam memberikan solusi yang berguna untuk penggunanya?

1.5 Tujuan Penelitian

Adapun tujuan dalam penelitian ini adalah :

Untuk menghasilkan suatu sistem pakar yang dapat digunakan dalam mendiagnosa kerusakan pada UPS sehingga pengguna dapat mengetahui tindakan yang harus dilakukan serta dapat melakukan pecegahan secara dini.

1.6 Manfaat Penelitian

Berdasarkan uraian di atas, manfaat penelitian di bagi menjadi dua bagian terpenting yang diberikan dari hasil perancangan sistem yaitu:

1. Aspek Teoritis

   Untuk mengetahui secara dalam tentang sistem pakar dengan metode *Forward Chaining* dan *Backward Chaining* dan cara sistem pakar dengan metode *Forward Chaining* dan *Backward Chaining* mendiagnosa kerusakan UPS.

2. Aspek Praktis

   Hasil penelitian bisa dijadikan media informasi untuk masyarakat dan menambah wawasan tentang kerusakan UPS serta solusi memperbaikinya.

BAB II
LANDASAN TEORI

2.1 Teori Dasar

Landasan teori perlu ditegakkan agar penelitian mempunyai dasar yang kokoh, dan bukan sekedar perbuatan coba-coba (*trial and error*). Adanya landasan teori merupakan ciri bahwa penelitian merupakan cara ilmiah untuk mendapatkan data. Kerlinger mengatakan bahwa teori adalah seperangkat konstruk (konsep), definisi, dan proposisi yang berfungsi untuk melihat fenomena secara sistematis melalui spesifikasi hubungan antarvariabel sehingga dapat berguna untuk menjelaskan dan meramalkan fenomena. Dengan kata lain, teori adalah generalisasi atau kumpulan generalisasi yang dapat digunakan untuk menjelaskan berbagai fenomena secara sistematis.

Teori dasar ini bertujuan untuk mengkaji tentang teori-teori yang berhubungan dengan penelitian yang sedang dikerjakan. Teori adalah serangkaian bagian atau variabel, definisi, dan dalil yang saling berhubungan yang menghadirkan sebuah pandangan sistematis mengenai fenomena dengan menentukan hubungan antar variabel, dengan menentukan hubungan antar variabel, dengan maksud menjelaskan fenomena alamiah (Sudaryono 2015, p.13).

2.1.1 Kecerdasan Buatan (Artificial Intelligence)

Kecerdasan Buatan atau *Artificial Intelligence* (AI) merupakan bidang ilmu komputer yang mempunyai peran penting di era kini dan masa akan datang. Bidang ini telah berkembang sangat pesat di 20 tahun terakhir seiring dengan pertumbuhan kebutuhan akan perangkat cerdas pada industri dan rumah tangga. AI mencakup bidang yang cukup besar. Mulai dari yang paling umum hingga yang khusus. Dari *learning* atau *perception* hingga pada permainan catur, pembuktian teori matekmatika, menulis puisi, mengemudikan mobil, dan melakukan diagnosis penyakit. AI relevan dengan berbagai amcam *task* kecerdasan (Budihartono & Suhartono 2014, p.2), AI merupakan sebuah ilmu yang universal.

2.1.2 Sistem Pakar (*Expert System*)

Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek (*emualtes*) kemampuan pengambilan keputusan (*decision making*) seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah.

Pakar atau ahli (*expert*) didefinisikan sebagai seseorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah,dan dokumentasi yang dipublikasikan lainnya, serta orang yang memiliki pengetahuan

meskipun bukan ahli. Istilah sistem berbasis pakar (*expert system*), sering disinonimkan dengan sistem berbasis pengetahuan (*knowledge-based system*) atau sistem pakar berbasis pengetahuan (*knowledge-based expert system*) (Rosnelly 2012, p.2).

Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud di sini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit. Contoh yang lain, montir adalah seorang yang punya keahlian dan pengalanan dalam menyelesaikan kerusakan mesin motor/mobil; psikolog adalah orang yanf ahli dalam memahami kerpribadian seseorang, dan lain-lain.

Sistem Pakar, yang mencoba memecahkan masalah yang biasanya hanya bisa dipecahkan oleh seorang pakar, dipandang berhasil ketika mampu mengambil keputusan seperti yang dilakukan oleh pakar aslinya baik dari sisi proses pengambilan keputusannya maupun hasil keputusan yang diperoleh (Kusrini 2008, p.3).

2.1.2.1 Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya (Rosnelly 2012, p.5), seperti:

a) Meningkatkan ketersediaan (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).

b) Mengurangi biaya (*reduced cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang user menjadi berkurang.

c) Mengurangi bahaya (*reduced danger*). Sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.

d) Permanen (*permanence*). Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.

e) Keahlian multipel (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.

f) Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku jika sistem dibuat oleh salah seorang pakar, sehingga akan selalu sama dengan pendapat

pakar tersebut kecuali sang pakar melakukan kesalahan yang mungkin terjadi pada saat tertekan atau stres.

g) Penjelasan (*explanation*). Sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan.

h) Seorang pakar mungkin saja terlalu lelah, tidak bersedia atau tidak mampu melakukannya setiap waktu. Hal ini akan meningkatkan tingkat kepercayaan bahwa kesimpulan yang dihasilkan adalah benar.

i) Respon yang cepat (*fast response*). Respon yang cepat atau *real time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.

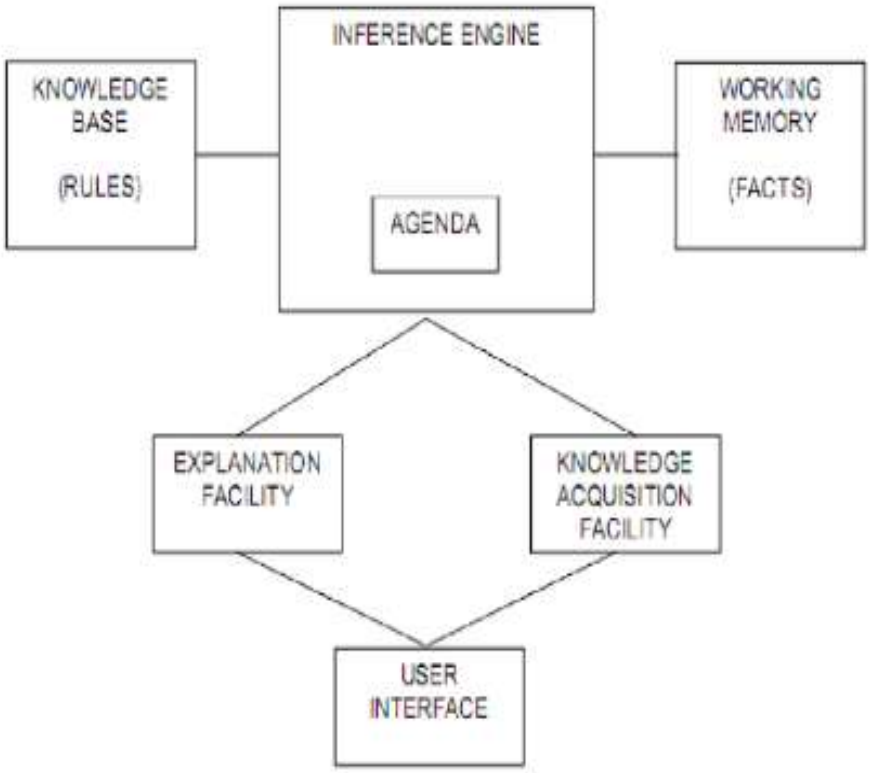
j) Stabil, tidak emosional, dan memberikan respon yang lengkap setiap saat (*steady, unemotional, and complete response at all times*). Karakteristik ini diperlukan pada situasi *real time* dan keadaan darurat (*emergency*) ketika seorang pakar mungkin tidak berada pada kondisi puncak disebabkan oleh stres atau kelelahan.

k) Pembimbing pintar (*intelligent tutor*). Sistem pakar dapat berperan sebagai *intelligent tutor* dengan memberikan kesempatan pada user untuk menjalankan contoh program dan menjelaskan proses *reasoning* yang dilakukan.

l) Basis data cerdas (*intelligent database*). Sistem pakar dapat digunakan untuk mengakses basis data secara cerdas.

2.1.2.2 Struktur Sistem Pakar

Menurut Rosenelly (2012:13) Adapun struktur sistem pakar dapat dilihat pada gambar berikut:



Gambar 2.1 Struktur Sistem Pakar
Sumber: Rosenelly (2012)

Komponen yang terdapat dalam struktur sistem pakar ini adalah *knowledge base* (*rules*), *inference engine*, *working memory*, *explanation facility*, *knowledge acquisition facility*, *user interface* (Rosnelly 2012, p.13).

a) Basis Pengetahuan (*Knowledge Base*)

Rosenelly (2012:13) Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi dan dua elemen dasar, yaitu fakta dan aturan. Fakta merupakan informasi tentang objek dalam area permasalahan tertentu, sedangkan aturan merupakan informasi tentang cara bagaimana memperoleh fakta baru dari fakta yang telah diketahui. Pada struktur sistem pakar diatas, *knowledge base* disini untuk menyimpan pengetahuan dari pakar berupa rule/aturan.

b) Mesin Inferensi (*Inference Engine*)

Rosenelly (2012:13) Mesin inferensi merupakan otak dari sebuah sistem pakar dan dikenal juga dengan sebutan *control structure* atau *rule interpreter*. Komponen ini mengandung mekanisme pola pikir dan penalaran yang digunakan oleh pakar dalam menyelesaikan suatu masalah. Mesin inferensi disini adalah *processor* pada sistem pakar yang mencocokkan bagian kondisi dari rule yang tersimpan didalam *knowledge base* dengan fakta yang tersimpan di working memory.

c) *Working Memory*

Rosenelly (2012:13) Berguna untuk menyimpan fakta yang dihasilkan oleh *inference engine* dengan penambahan parameter berupa derajat kepercayaan atau dapat juga dikatakan sebagai *global database* dari fakta yang digunakan oleh *rule-rule* yang ada.

d) *Explanation Facility*

Rosenelly (2012:13) Menyediakan kebenaran dari solusi yang dihasilkan kepada user (*reasoning chain*).

e) *Knowledge Acquisition Facility*

Rosenelly (2012:13) Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke program komputer, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

f) *User Interface*

Rosenelly (2012:13) Mekanisme untuk memberi kesepakatan kepada user dan sistem pakar untuk berkomunikasi. Antar muka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem.

Selain itu antarmuka menerima informasi dari sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pemakai.

2.1.2.3 Karakteristik Sistem Pakar

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum (Rosnelly 2012, p.13),berikut ini:

a) Kinerja sangat baik (*high perfotmance*). Sistem harus mampu memberikan respon berupa saran (*advice*) dengan tingkat kualitas yang sama dengan seorang pakar atau melebihinya.

b) Waktu respon yang baik (*adequte respon time*). Sistem juga harus mampu bekerja dalam waktu yang sama baiknya (*reasonable*) atau lebih cepat dibandingkan dengan seorang pakar dalam menghasilkan keputusan. Hal ini sangat penting terutama pada sistem waktu nyata (*real time*)

c) Dapat diandalkan (*good reliability*). Sistem harus dapat diandalkan dan tidak mudah rusak/*crash*.

d) Dapat dipahami (*understandable*). Sistem harus mampu menjelaskan langkah-langkah penalaran yang dilakukannya seperti seorang pakar.

e) Fleksibel (*flexibility*). Sistem harus menyediakan mekanisme untuk menambah, mengubah, dan menghapus pengetahuan.

2.1.2.3 Metode Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia. Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut *Inference Engine* (Mesin Inferensi).

Ketika representasi pengetahuan (RP) pada bagian *knowledge base* telah lengkap, atau paling tidak telah berada pada level yang cukup akurat, maka RP tersebut telah siap digunakan. *Inference engine* merupakan modul yang berisi program tentang bagaimana mengendalikan proses *reasoning* (Kusrini 2008, p.8).

Ada dua metode inferensi yang penting dalam sistem pakar yaitu: runut maju (*forward chaining*) dan runut balik (*backward chaining*).

2.1.2.3.1 Runut Maju (*Forward Chaining*)

Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson,1998).

Menurut (Giarattano dan Riley, 1994) metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*) (Kusrini 2008, p.8).

2.1.2.3.2 Runut Balik (*Backward Chaining*)

Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut (Giarattano dan Riley, 1994). Runut balik disebut juga sebagai *goal-driven reasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur.

Menurut (Schnupp, 1989) tujuan inferensi adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah diagnosis (Kusrini 2008, p.8).

2.1.3 Tabel Keputusan

Dalam proses pengambilan keputusan, semua informasi yang diperlukan disusun dalam bentuk ringkasan hasil yang disebut sebagai tabel hasil atau tabel keputusan. Tabel ini merupakan suatu matriks yang terdiri dari baris yang menunjukkan berbagai *alternative* pilihan/keputusan dan kolom yang menunjukkan nilai harapan untuk setiap alternative pilihan/keputusan pada berbagai keadaan atau situasi yang mungkin terjadi (Herjanto 2009, p.26)

2.1.4 Pohon Kuputusan

Pohon keputusan adalah pohon yang ada dalam analisa pemecahan masalah, pemetaan mengenail alternatif-alternatif pemecahan masalah yang dapat

diambil dari masalah tersebut. Berikut ini pengertian pohon keputusan yang diambil dari situs Wikipedia:

Decision tree learning,used in data mining and machine learning, uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. More descriptive names for such tree models are classification tree or regression trees. In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications.

Pohon tersebut juga memperlihatkan faktor-faktor kemungkinan atau probablitas yang akan mempengaruhi alternatif-alternatif keputusan tersebut, disertai dengan estimasi hasil akhir yang akan didapat bila kita mengambil alternatif keputusan tersebut. Pohon keputusan misalnya saja bisa digunakan manajemen dan bisnis, yaitu dalam hal perencanaan bisnis berikutnya. Perencanaan bisnis yang baik sebenarnya memuat tahap-tahap yang dapat memaksimalkan peluang keberhasilan. Salah satu tahap yang penting adalah tahap pengambilan keputusan.(Juju 2010, p.56)

Pohon keputusan memiliki manfaat sebagai berikut:

a) Berguna dalam mengeksplorasi data, sehingga data yang tersembunyi bisa diolah dan dikembangkan lagi.
b) Untuk mem-*break down* proses pengambilan keputusan yang kompleks menjadi lebih simpel sehingga pengambil keputusan akan lebih menginterpretasikan solusi dari permasalahan.
c) Bisa dijadikan sebagai *tool* pengambilan keputusan terakhir.

d) Mengubah keputusan yang kompleks menjadi lebih *simple*, spesifik dan mudah.

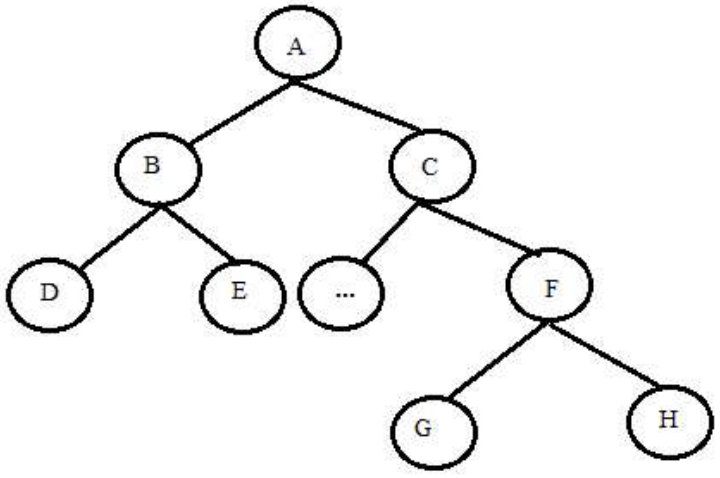
Selain memiliki kelebihan, pohon keputusan juga memiliki kekurangan, diantaranya:

a) Kesulitan dalam mendesain pohon keputusan yang optimal.

b) Hasil kualitas keputusan yang didapatkan dari metode pohon keputusan sangat tergantung pada bagaimana pohon tersebut didesain.

c) Terjadi *overlap* terutama ketika kelas-kelas dan kriteria yang digunakan jumlahnya banyak. Hal tersebut juga dapat menyebabkan meningkatkan waktu pengambilan keputusan dan jumlah memori yang diperlukan. Tetapi ketika kriterianya lebih simple tentu saja pengambilan keputusan menjadi lebih cepat.

d) Pengakumulasian jumlah error dari setiap tingkat dalam sebuah pohon keputusan yang besar.



Gambar 2.2 Contoh Pohon Keputusan
Sumber: Data Penelitian (2017)

2.1.5 UML (*Unified Modeling Language*)

Salah satu pemodelan yang saat ini paling banyak digunakan adalah UML. UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Salahuddin 2011, p.113).

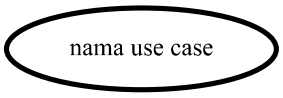
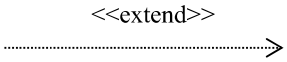
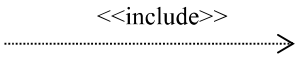
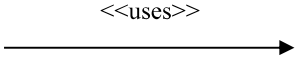
2.1.5.1 *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakukan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Use case atau diagram *use case* merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut *actor* dan *use case* (Rosa & Salahuddin 2011, p.155).

Berikut adalah simbol-simbol yang ada pada diagram *use case:*

Tabel 2.1*Simbol Use Case Diagram*

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan meggunakan kata kerja di awal frase nama *use case*
Aktor / *actor* nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang; tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Asosiasi / *association* 	Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* yang memiliki interaksi dengan aktor
Ekstensi / *extend* 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan
Generalisasi / *generalization* 	Hubungan generalisai dan spesialisasi (umum-khusu) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
Menggunakan / *include* / *uses*  	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsina atau sebagai syarat dijalankan *use case* ini.

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.1.5.2 *Activity* Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.

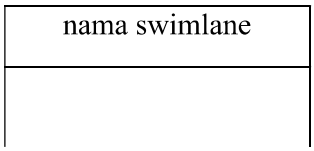
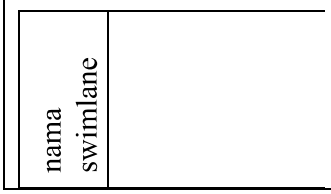
Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa & Salahuddin, 2011, p. 161). Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
b. Urutan atau pengelompokkan tampilan dari sistem/user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

   Rancangan menu yang ditampilkan pada perangkat lunak

   Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision*	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa, A. & Shalahuddin, M. (2011)

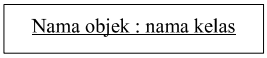
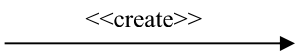
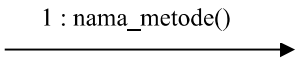
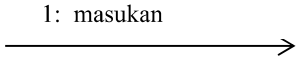
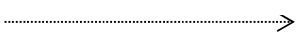
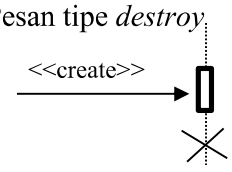
2.1.5.3 *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antara objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa & Salahuddin 2011, p.163).

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor / *actor* nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang; tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garis hidup / *lifeline* 	Menyatakan kehidupan suatu objek
Objek  Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan
Pesan tipe *create*  <<create>>	Objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe *call*  1 : nama_metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe *send*  1: masukan	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe *return*  1: keluaran	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe *destroy*  <<create>>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.1.5.4 *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

a. Atribut merupakan variabel-variabel yang dimiliki suatu kelas.

b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau programmer dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas (Rosa & Salahuddin 2011, p.165). Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

   Kelas yang memiliki fungsi awal di eksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*View*)

   Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

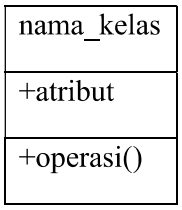
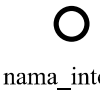
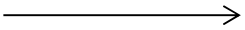
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.4 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka / *interface* 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi / *association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah / *directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan / *dependency* 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua bagian (*whole part*)

Sumber: Rosa, A. & Shalahuddin, M. (2011)

2.2 Variabel

Variabel penelitian juga menjadi suatu bagian penting dalam penelitian. Kemampuan peneliti untuk memahami variable penelitian sangat tergantung pada penguasaan konsep tentang penelitian terutama variable penelitian. Selain itu, pengalaman peneliti dalam melaksanakan penelitian maupun menyusun proposal peneltian juga dapat menambah pemahaman tentang kemampuan mengidentifikasi variable penelitian. Variabel adalah sebuah konsep yang dioperasionalkan. Lebih tepatnya, operasional properti dari sebuah objek agar dapat dioperasionalkan, diaplikasikan, dan menjadi property dari objek. (I Ketut Swarjana 2015, p.42)

2.2.1 *Uninterruptible Power Supply* (UPS)

Uninterruptible power supply (UPS) adalah perangkat yang berfungsi untuk memberikan daya listrik cadangan. Bila listrik mati mendadak, peralatan yang terhubung ke UPS tidak ikut mati. UPS biasanya dilengkapi dengan baterai kering dengan daya tahan 2-3 tahun. Daya tahan UPS ketika listrik mati antara 10-30 menit, tergantung bebannya (Komputer 2008, p.44).

Didalam UPS terdapat charger untuk mengisi baterai. Jadi, bila baterai kosong, ia akan diisi kembali. UPS terbagi menjadi dua macam, yaitu:

a) **Sistem *Onlline/Dual Conversion***

Listrik yang bentuknya arus bolak-balik diubah menjadi arus searah, kemudian dari arus searah ini diubah lagi ke arus bolak-balik. Dengan dua kali konversi ini, gangguan listrik akan tersaring sehingga pasokannya lebih bersih. Waktu perpindahan dari sistem listrik ke sistem baterai berlangsung seketika (nol detik).

b) Sistem *Off-line/Line Interactive*

Ketika ada arus listrik, arus ini akan diteruskan ke beban setelah memalui saringan dan *stabilizer*. Ketika listrik mati, sakelar langsung pindah dan inverter bekerja membarikan listrik ke beban (ada waktu pindah 3-4 milidetik = 0.004 detik).

2.3 *Software* Pendukung

Program pendukung yang digunakan untuk menjalakan sistem aplikasi tersebut adalah:

2.3.1 Bahasa Pemograman PHP

PHP merupakan bahasa pemrograman berbasis *web* yang dibuat secara khusus untuk membangun aplikasi berbasis *web*. Selain tersedia secara gratis, PHP juga mudah dipelajari oleh siapapun. Akan lebih mudah lagi, jika kita suda

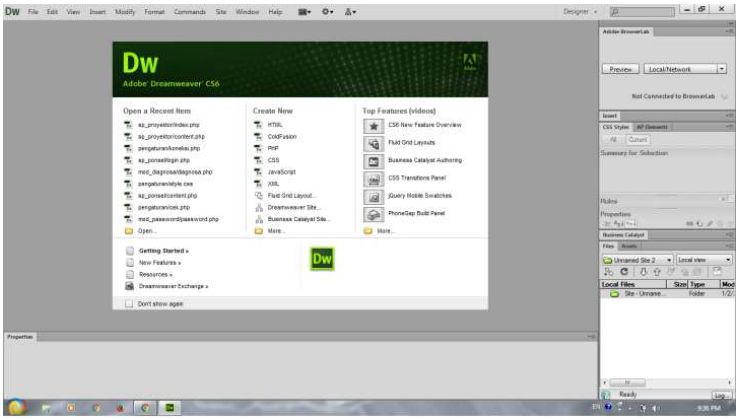
pernah mempelajari Bahasa C atau C++. Banyak perintah PHP yang diturunkan atau mengadopsi perintah-perintah di Bahasa C (Solichin 2016, p.23).

Gambar 2.3 Logo *php*
Sumber: https://wikipedia.org/wiki/PHP#/media/File:PHP-logo.svg

2.3.2 Aplikasi *Adobe Dreamweaver*

Adobe Dreamweaver merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah website, baik secara grafis maupun dengan menuliskan kode sumber secara langsung (Wahana Komputer 2010, p.2).

Adobe Dreamweaver memudahkan pengembang *website* untuk mengelola halaman-halaman *website* dan asset-asetnya, baik gambar(image), animasi flash, video, suara dan lain sebagainya.

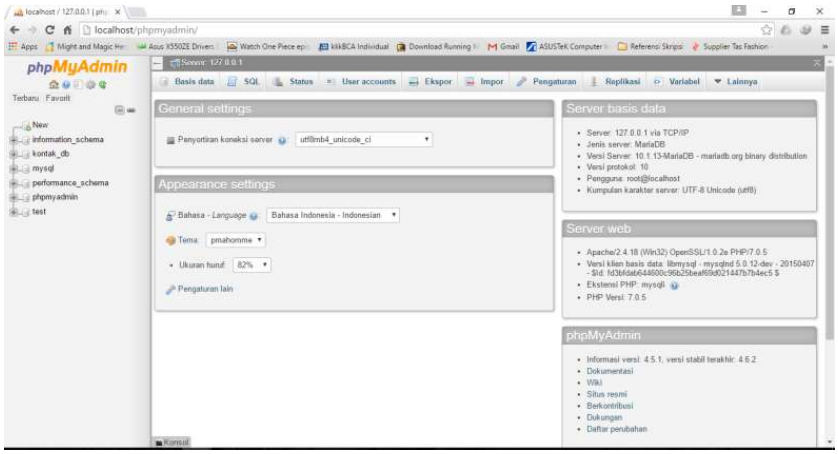


Gambar 2.4 Tampilan Adobe Dreamweaver CS6
Sumber: Data Peneliti (2017)

2.3.3 *MySQL Database*

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*database management system*), database ini *multithread, multi-user* (Miftakhul Huda 2010, p.181).

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Stuctured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.



Gambar 2.5 Tampilan Database MySQL
Sumber: Data Peneliti (2017)

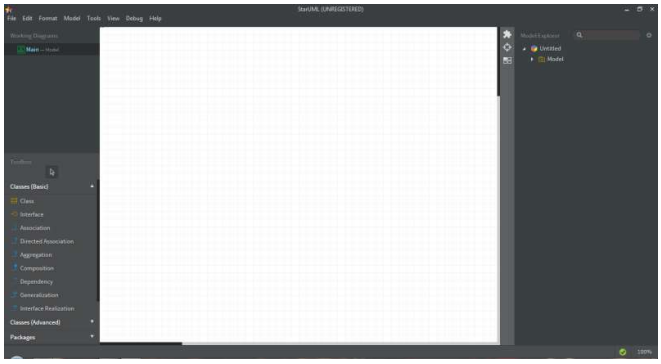
2.3.4 *WampServer*

WampServer adalah singkatan dari Windows, Apache, MySQL dan PHP. Wampserver adalah aplikasi yang menggabungkan antara Apache, Mysql, dan PHP. Sehingga, Wampserver akan mencukupi semua persyaratan yang diminta oleh Joomla. Dengan demikian, tidak perlu lagi meng-install masing –masing aplikasi apache, mysql, dan php (Meissa 2009).

Gambar 2.6 Logo *WampServer*
Sumber: http://www.wampserver.com/en/

2.3.5 StarUML

StarUML adalah *platform* pemmodelan perangkat lunak yang mendukung UML (*Unified Modeling Language*). StarUML yang berbasiskan pada UML versi 1.4, menyediakan sebelas jenis *Diagram* yang berbeda, dan mendukung notasi UML 2.0 StarUML juga secara aktif mendukung pedekatan MDA (*Model Driven Architercture*) dengan mendukung konsep UML. StarUML unggul dalam hal kustomisasi lingkungan kerja pengguna, dan memiliki ekstensibilitas tinggi pada fungsionalitasnya. (Triandini & Suardika 2012, p.1)



Gambar 2.7 *Logo StarUML*
Sumber: Data Peneliti (2017)

2.4 Penelitian Terdahulu

Penelitian Harison dan Alexyusanderia (Jurnal Penelitian Momentum Vol. 16 No.2 Agustus 2014) dengan judul "Sistem Pakar Perawatan Dan Perbaikan Ringan Mobil Bensin Menggunakan *Video Tutorial* Berbasis *Web*" yang bertujuan untuk kemudahan dalam memberikan petunjuk dalam menentukan bagian komponen mobil yang mengalami kerusakan karna setiap kendala kerusakan yang di diagnosis sistem ini dengan memberikan video komponen mobil tersebut dan memberikan pengarahan atau petunjuk dalam memperbaiki kerusakan ringan kendaraan

Penelitian Purwanto dan Bramuditya Adi Putra (Jurnal Penelitian ELKOM Vol. 9 No.1 April 2016) dengan judul "Sistem Pakar Diagnosa Kerusakan Pada Televisi Led Dengan Mmenggunakan Metode *Forward Chaining*" yang bertujuan untuk membantu meringankan beban seorang teknisi dalam menentukan gejala kerusakan pada televisi LED dan solusi untuk memperbaikinya dan teknisi bisa belajar mengenai mendiagnosa kerusakan tanpa bergantung pada pakar.

Penelitian Rosmawati Tamin (Jurnal Penelitian Vol.1 No.1 Tahun 2015) dengan judul "Sistem Pakar Uuntuk Diagnosa Kerusakan Pada Printer Menggunakan Meode *Forward Chaining*" yang dibangun untuk memudahkan para pengguna printer jenis canon untuk mengetahui penyebab, akibat dan gejala-gejala yang ditimbulkan dari kerusakan printer, memudahkan para pengguna printer jenis canon untuk mencari solusi kerusakan printer, memudahkan para pengguna printer untuk mendapatkan informasi mengenai cara merawat printer dengan baik melalui penyajian informasi berita yang terdapat dalam website aplikasi mendeteksi kerusakan pada printer dan memudahkan para teknisi untuk memperbaiki printer.

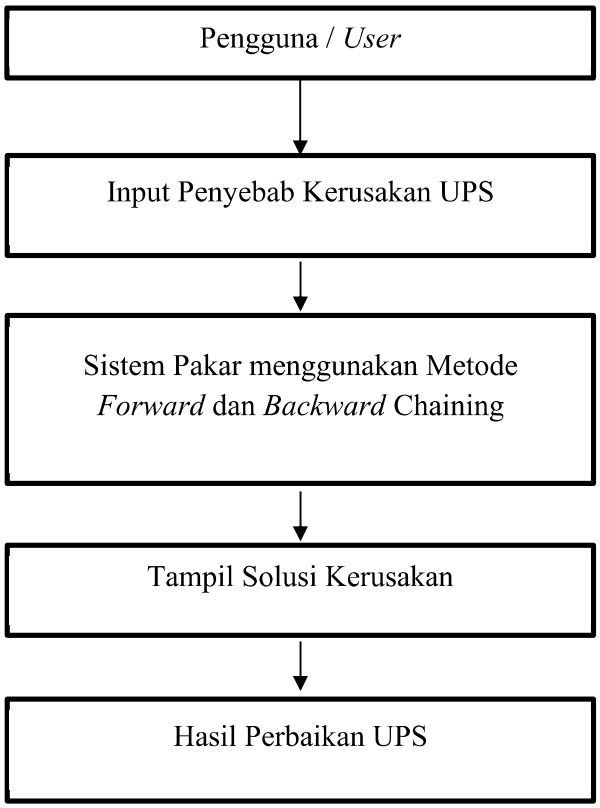
Penelitian Akim Manaor Hara Pardede dan Budi Serasi Ginting (Jurnal Penelitian KAPUTAMA Vol.7 No.1 Juli 2013) dengan judul "Perancangan Sistem Pakar Deteksi Kerusakan Prnter Canon Berbasis *Web*" yang bertujuan mempermudah bagi *non* pakar untuk dapat langsung berkonsultasi mengenai diagnosa kerusakan yang terjadi pada mesin printer canon kapan saja dan dimana saja dan perancangan ini diterapkan dalam bentuk *website* yang dapat diakses melalui internet maka akan lebih mempermudah para pengguna untuk mengaksesnya dimanapun mereka berada selama masih didukung dengan sarana yang mendukung untuk mengaksesnya.

Penelitan Siska Iriani (Jurnal *Indonesian Journal on Networking and Security* Vol.4 No.1 Januari 2016) dengan judul "Penerapan Metode *Backward Chaining* Pada Sistem Pakar Mendiagnosa Penyakit Tulang Manusia" yang bertujuan membantu pengguna untuk mendiagnosa jenis penyakit, serta

penjelasan dari penyakit-penyakit tersebut dan cara pencegahan, pengobatan serta penyebab timbulnya berdasarka gejala penyakit yang dialami.

2.5 Kerangka Berpikir

Penjelasan sementara terhadap suatu gejala yang menjadi objek permasalahan peneliti. Kerangka berpikir ini disusun dengan berdasarkan pada tinjauan pustaka dan hasil penelitian yang relevan atau terkait.



Gambar 2.8 Kerangka Berpikir Peneliti
Sumber: Data Peneliti (2017)

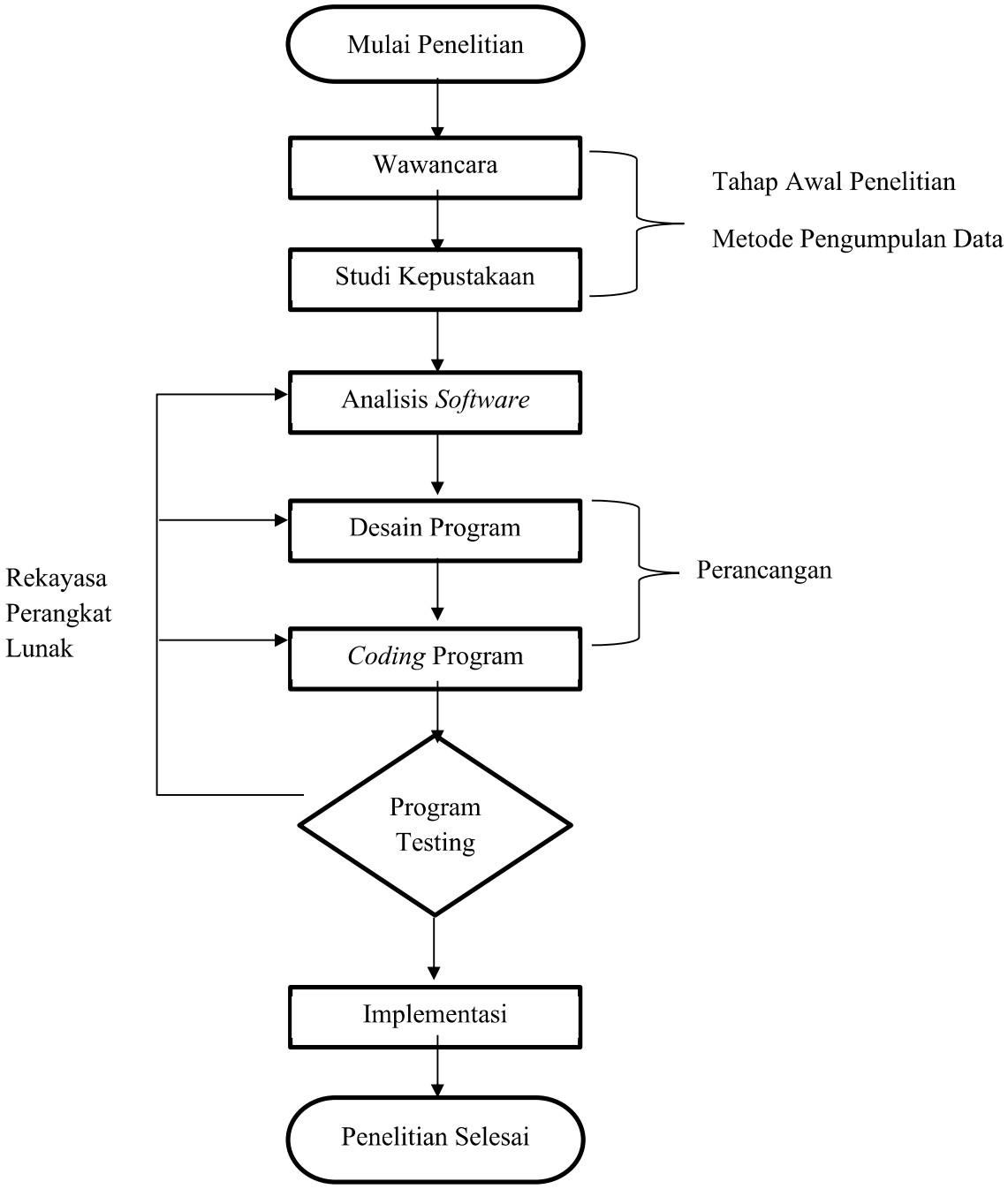
BAB III
METODE PENELITIAN

3.1 Desain Penelitian

Desain penelitian ini bertujuan untuk melaksanakan penelitian sehingga diperoleh suatu logika dalam pengujian untuk memperoleh hasil penelitian dan dalam membuat kesimpulan sesuai dengan fokus penelitian.

Desain penelitian adalah semua proses yang diperlukan dalam perencanaan dan pelaksanaan penelitian. Desain penelitian adalah kerangka atau cetak biru dalam melaksanakan suatu proyek riset. Desain penelitian adalah penjelasan mengenai berbagai komponen yang akan digunakan peneliti dan kegiatan yang akan dilakukan selama proses penelitian (Martono et al. 2010, p.131).

Untuk menyelesaikan masalah didalam pengembangan perangkat lunak, penulis memilih menggunakan metode pengembangan sistem model sekuensial linier. Model ini sering juga disebut dengan siklus kehidupan klasik atau model air terjun.



Gambar 3.1 Desain Penelitian
Sumber: Data Peneliti (2017)

3.2 Operasional Variabel

3.2.1 Metode *Forward Chaining*

Runut maju berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson, 1998).

Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*) (Giarattano dan Riley, 1994). Berikut adalah operasional variabel menggunakan metode inferensi *Forward Chaining*. (Kusrini 2008, p.8)

Tabel 3.1 Indikator Kerusakan UPS (FC)

Variabel	Indikator
Kerusakan Komponen UPS	1. Baterai bermasalah 2. Trafo bermasalah 3. *Charger* bermasalah

Sumber: Data penelitian (2017)

Tabel 3.2 Kode Indikator Kerusakan UPS (FC)

Kode	Indikator	Penyebab
R01	Baterai	1. Mati total 2. Listrik mati, UPS juga mati 3. Listrik mati, UPS nyala tetapi tidak tahan lama
R02	Trafo	1. UPS tidak mensuplai listrik 2. UPS tidak hidup saat listrik mati dengan baterai baru
R03	*Charger*	1. Baterai tidak bisa diisi 2. UPS tidak tahan lama dengan baterai baru 3. Baterai cepat rusak dengan baterai baru

Sumber: Data Penelitian (2017)

Tabel 3.3 Kode Penyebab Kerusakan UPS (FC)

Kode Penyebab	Nama Penyebab
T01	Mati Total
T02	Listrik mati, UPS juga mati
T03	Listrik mati, UPS nyala tetapi tidak tahan lama
T04	UPS tidak mensuplai listrik
T05	UPS tidak hidup saat listrik mati dengan baterai baru
T06	Baterai tidak bisa diisi
T07	UPS tidak tahan lama dengan baterai baru
T08	Baterai cepat rusak dengan baterai baru

Sumber: Data Penelitian (2017)

Tabel 3.4 Data Solusi Kerusakan UPS (UD)

Indikator	Penyebab	Solusi
Baterai bermasalah	1. 1 Mati total 2. Listrik mati, UPS juga mati 3. Listrik mati, UPS nyala tetapi tidak tahan lama	1. Baterai rusak, ganti baterai baru.
Trafo bermasalah	1. UPS tidak mensuplai listrik 2. UPS tidak hidup saat listrik mati dengan baterai baru	1. Trafo rusak, ganti trafo baru 2. Trafo drop, melakukan service gulung ulang trafo
Charger bermasalah	1. Baterai tidak bisa diisi 2. UPS tidak tahan lama dengan baterai baru 3. Baterai cepat rusak dengan baterai baru	1. Melalukan *service.*

Sumber: Data Penelitian (2017)

Tabel 3.5 Tabel Keputusan Kerusakan UPS (FC)

Kode Kerusakan	Kode Penyebab							
	T01	T02	T03	T04	T05	T06	T07	T08
R01	√	√	√					
R02				√	√			
R03						√	√	√

Sumber: Data Penelitian (2017)

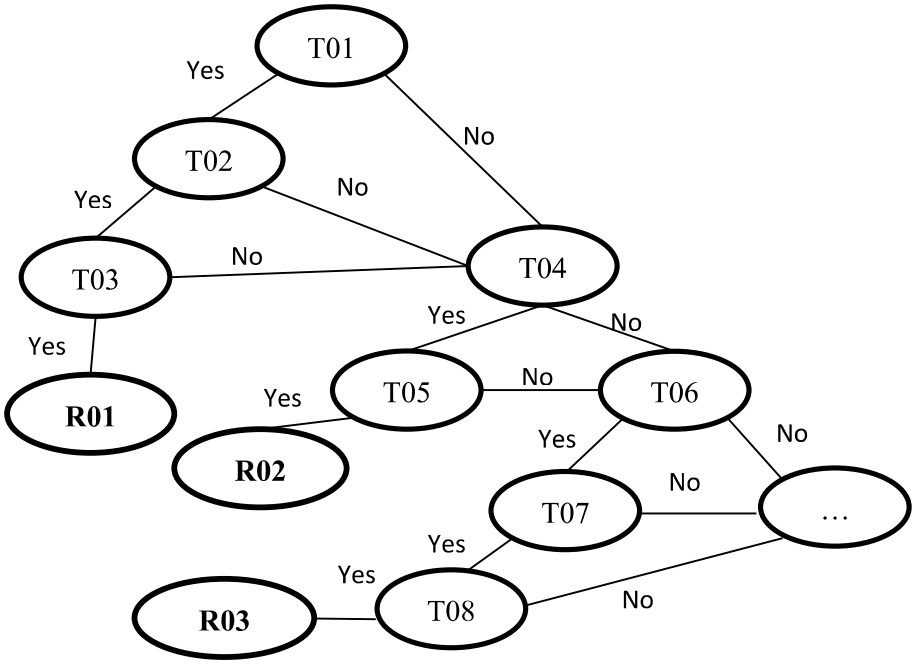
Kaidah berdasarkan dari tabel diatas:

1. IF T01 AND T02 AND T03 THEN R01
2. IF T04 AND T05 THEN R02
3. IF T06 AND T07 AND T08 THEN R03

Penjelasan dari kaidah diatas:

1. Jika Mati total; listrik mati, UPS juga mati; listrik mati, UPS nyala tetapi tidak tahan lama maka Baterai bermasalah.
2. Jika UPS tidak mensuplai listrik, UPS tidak hidup saat listrik mati dengan baterai baru maka Trafo bermasalah.
3. Jika Baterai tidak bisa diisi, UPS tidak tahan lama dengan baterai baru, baterai cepat rusak dengan baterai baru maka *Charger* bermasalah.

Berdasarkan tabel keputusan diatas maka pohon keputusannya adalah sebagai berikut:



Gambar 3.2 Pohon Keputusan (UD)
Sumber: Data Olahan Peneliti (2017)

3.2.2 Metode *Backward Chaining*

Runut balik merupakan metode penalaran kebalikan dari runut maju. Dalam runut balik penalaran dimulai dengan tujuan kemudian merunut balik ke jalur yang akan mengarahkan ke tujuan tersebut (Giarattano dan Riley, 1994). Runut balik disebut juga sebagai *goal-driven reasoning*, merupakan cara yang efisien untuk memecahkan masalah yang dimodelkan sebagai masalah pemilihan terstruktur.

Menurut (Schnupp, 1989) Tujuan inferensi adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah diagnosis.(Kusrini 2008, p.8)

Tabel 3.6 Indikator Penyebab Kerusakan UPS (BC)

Variabel	Indikator
Penyebab Kerusakan UPS	1. Mati Total 2. Listrik mati, UPS juga mati 3. Listrik mati, UPS nyala tetapi tidak tahan lama 4. UPS tidak mensuplai listrik 5. UPS tidak hidup saat listrik mati dengan baterai baru 6. Baterai tidak bisa diisi 7. UPS tidak tahan lama dengan baterai baru 8. Baterai cepat rusak dengan baterai baru

Sumber: Data Penelitian (2017)

Tabel 3.7 Kode Indikator Kerusakan UPS (BC)

Kode	Indikator	Komponen Kerusakan
T01	Mati Total	Baterai bermasalah
T02	Listrik mati, UPS juga mati	
T03	Listrik mati, UPS nyala tetapi tidak tahan lama	
T04	UPS tidak mensuplai listrik	Trafo bermasalah
T05	UPS tidak hidup saat listrik mati dengan baterai baru	
T06	Baterai tidak bisa diisi	*Charger* bermasalah
T07	UPS tidak tahan lama dengan baterai baru	
T08	Baterai cepat rusak dengan baterai baru	

Sumber: Data Penelitian (2017)

Tabel 3.8 Kode Komponen Kerusakan UPS (BC)

Kode	Nama Kerusakan
R01	Baterai bermasalah
R02	Trafo bermasalah
R03	*Charger* bermasalah

Sumber: Data Penelitian (2017)

Tabel 3.9 Data Solusi Kerusakan UPS (BC)

Indikator Gejala	**Kerusakan**	**Solusi**
Mati Total	Baterai bermasalah	1. Baterai rusak, ganti baterai baru.
Listrik mati, UPS juga mati		
Listrik mati, UPS nyala tetapi tidak tahan lama		
UPS tidak mensuplai listrik	Trafo bermasalah	1. Trafo rusak, ganti trafo baru 2. Trafo drop, melakukan service gulung ulang trafo
UPS tidak hidup saat listrik mati dengan baterai baru		
Baterai tidak bisa diisi	*Charger* bermasalah	1. Melalukan service.
UPS tidak tahan lama dengan baterai baru		
Baterai cepat rusak dengan baterai baru		

Sumber: Data Penelitian (2017)

Tabel 3.10 Tabel Keputusan (BC)

Kode Penyebab	Kode Kerusakan		
	R01	**R02**	**R03**
T01	√		
T02	√		
T03	√		
T04		√	
T05		√	
T06			√
T07			√
T08			√

Sumber: Data Penelitian (2017)

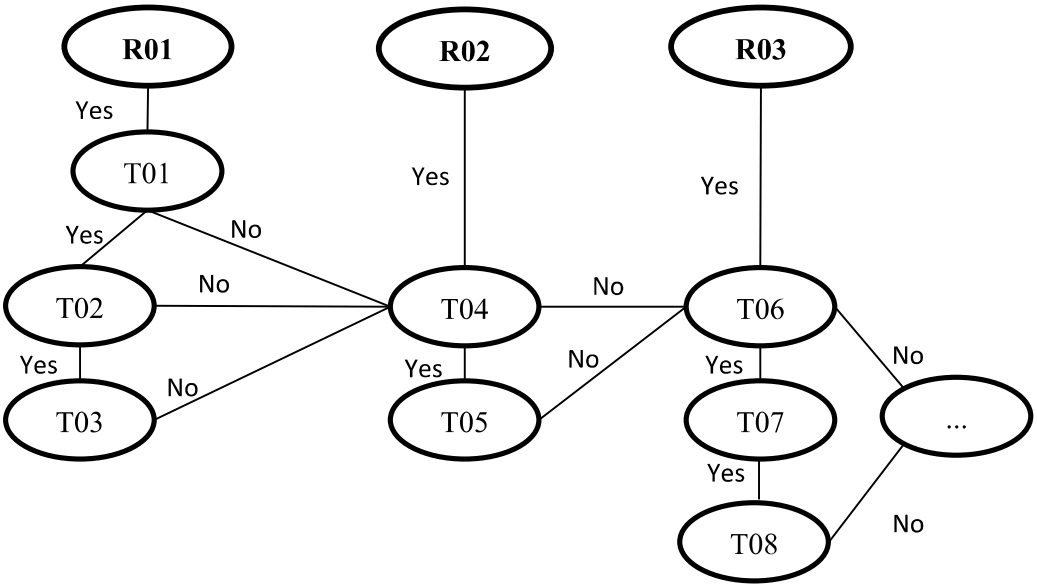
Kaidah berdasarkan dari tabel diatas:

1. IF R01 THEN T01 AND T02 AND T03
2. IF R02THEN T04 AND T05
3. IF R03 THEN T06 AND T07 AND T08

Penjelasan dari kaidah diatas:

1. Jika baterai bermasalah maka mati total; listrik mati, UPS juga mati; listrik mati, UPS nyala tetapi tidak tahan lama.
2. Jika trafo bermasalah maka UPS tidak mensuplai listrik, UPS tidak hidup saat listrik mati dengan baterai baru.
3. Jika *charger* bermasalah baterai tidak bisa diisi, UPS tidak tahan lama dengan baterai baru, baterai cepat rusak dengan baterai baru.

Berdasarkan tabel keputusan diatas maka pohon keputusannya adalah sebagai berikut:



Gambar 3.3 Pohon Keputusan (PD)
Sumber: Data Olahan Peneliti (2017)

3.3 Teknik Pengumpulan Data

Data yang dibutuhkan dalam analisis ini adalah gejala dan solusi. Dari data yang diperoleh dapat dilakukan analisis terhadap kebutuhan sistem, yang selanjutnya dijadikan sebagai acuan untuk menerjemahkan ke dalam bahasa pemograman.

a. Teknik Wawancara

Merupakan metode yang dilakukan dengan cara melakukan diskusi serta tanya jawab dengan sumber yang dianggap memiliki pengetahuan lebih mengenai permasalahan yang dijadikan objek penelitian. Alat bantu yang peneliti gunakan dalam wawancara ini yaitu alat bantu berupa alat tulis untuk mencatat hasil wawancara.

b. Teknik Studi Pustaka

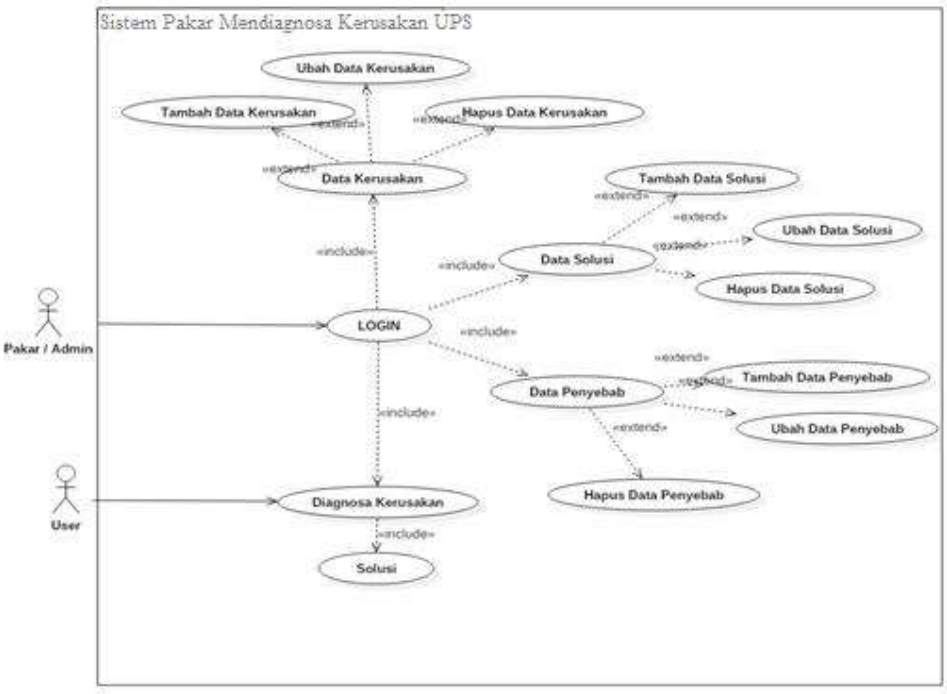
Merupakan metode yang dilakukan dengan cara mencari bahan yang mendukung dalam pendelisian masalah melalui buku-buku yang erat kaitannya dengan objek permasalahanya.

3.4 Perancangan Sistem

3.4.1 UML (Unified Modelling Language)

UML (*Unified Modeling Language*) adalah salah standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan *requirement*, membuat analis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

3.4.1.1 *Use Case Diagram*



Gambar 3.4 Use Case Diagram
Sumber: Data Olahan Peneliti (2017)

a) **Admin**, orang yang memiliki akses ke seluruh halaman sistem termasuk menambah/mengubah/menghapus data.

b) **User**, orang yang hanya bisa melakukan diagnosa dalam sistem .

c) **Login**, proses untuk mengakses dengan memasukkan identitas dan kata sandi.

d) **Data Kerusakan**, data yang diinput admin untuk membuat *rule.*

e) **Data Penyebab**, data yang diinput oleh admin untuk membuat *rule.*

f) **Data Solusi**, data yang diinput oleh pakar/admin untuk membuat *rule.*

g) **Tambah Data**, hanya dapat dilakukan oleh admin untuk menambah data.

h) **Edit Data**, hanya dapat dilakukan oleh admin untuk mengubah data.

i) **Hapus Data**, hanya dapat dilakukan oleh admin untuk menghapus data.

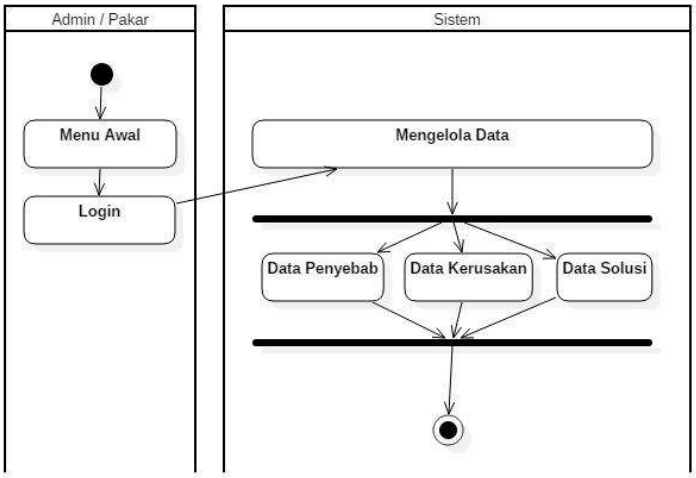
j) **Diagnosa Kerusakan**, mengidentifikasi mengenai suatu kerusakan.

k) **Solusi**, hasil atau output dari diagnosa kerusakan.

3.4.1.2 *Activity* Diagram

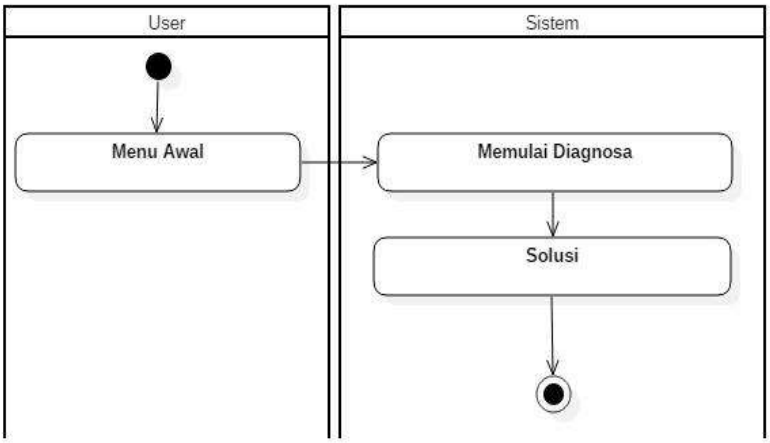
Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.

Proses dimana admin masuk ke menu awal, melakukan *login*, mengelola data termasuk menambah/mengubah/menghapus data.



Gambar 3.5 *Activity Diagram* Admin
Sumber: Data Olahan Peneliti (2017)

Proses dimana *user* atau pengguna melakukan diagnosa kerusakan termasuk menjawab pertanyaan dan mendapatkan solusi.

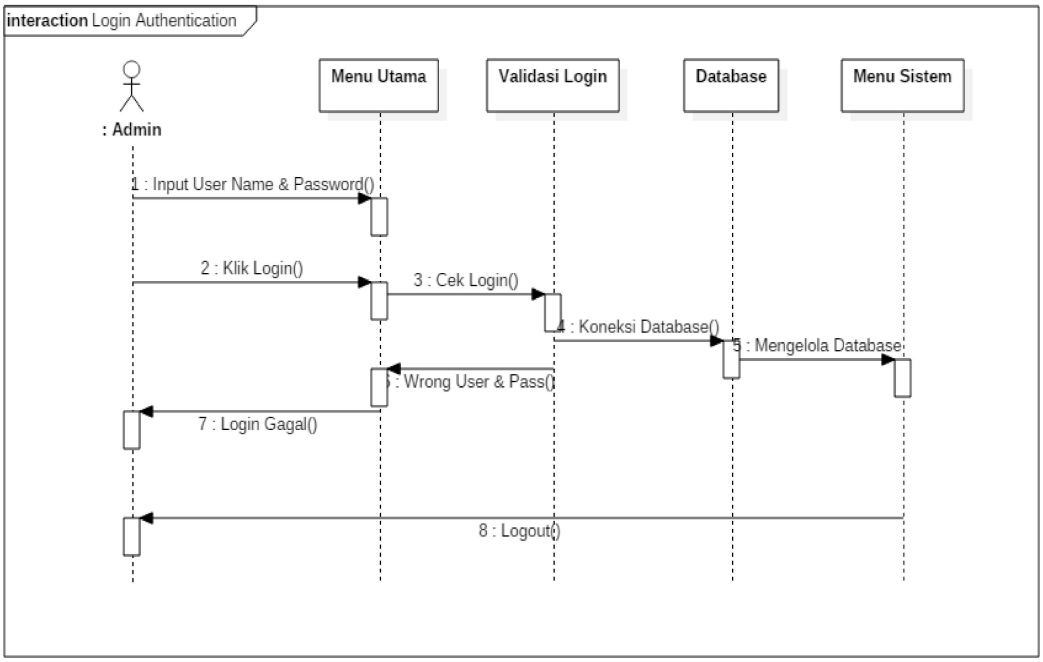


Gambar 3.6 *Activity Diagram* User
Sumber: Data Olahan Peneliti (2017)

3.4.1.3 *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antara objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

a) Sequence Diagram Login Admin

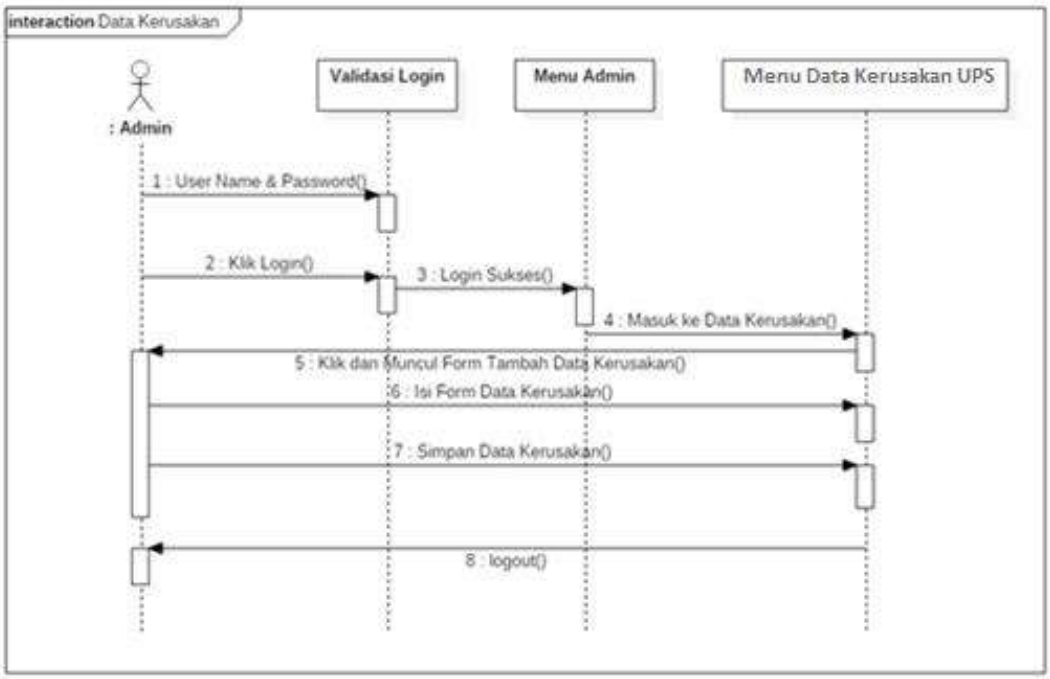


Gambar 3.7 *Sequence Diagram Login* Admin
Sumber: Data Olahan Peneliti (2017)

Admin menginput *ID* dan *Password*, dilanjutkan dengan klik *Login* kemudian validasi *login*, jika valid admin akan masuk ke *database* dan mengelola

data jika invalid atau salah *ID/Password* maka *Login* gagal, setelah mengelola data *admin logout*.

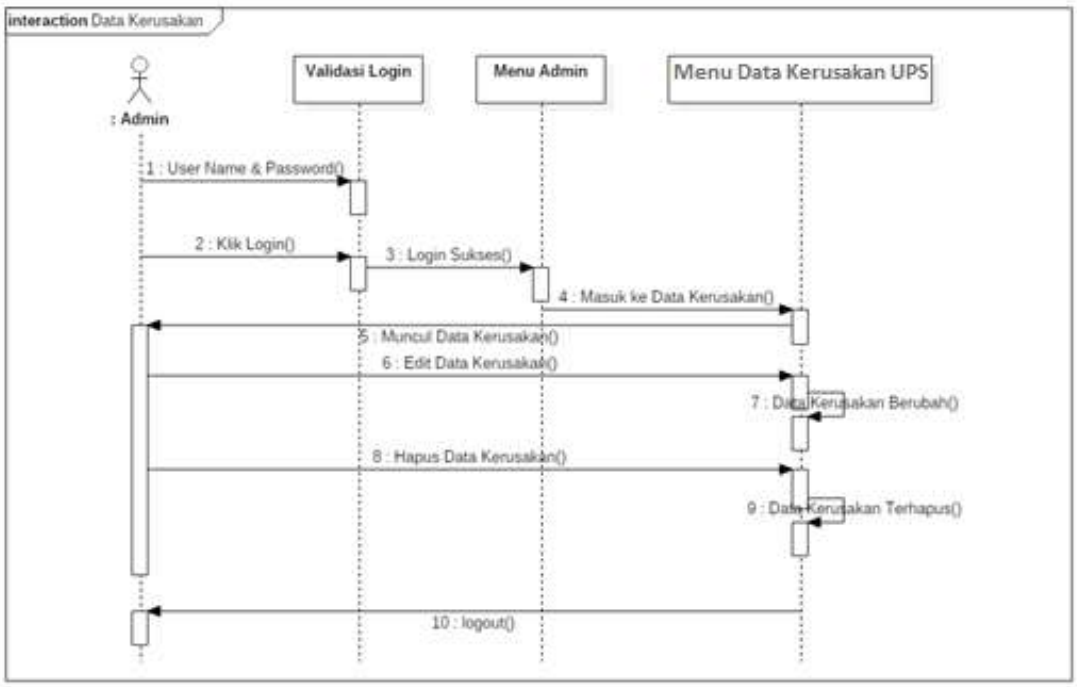
b) Sequence Diagram Input Data Kerusakan



Gambar 3.8 *Sequence Diagram Input* Data Kerusakan
Sumber: Data Olahan Peneliti (2017)

Admin melakukan *login*, klik *login* jika *login* sukses *admin* akan masuk ke data kerusakan, disini *admin* muncul form tambah data kerusakan, selanjutnya *admin* akan mengisi form data kerusakan dan menyimpannya.

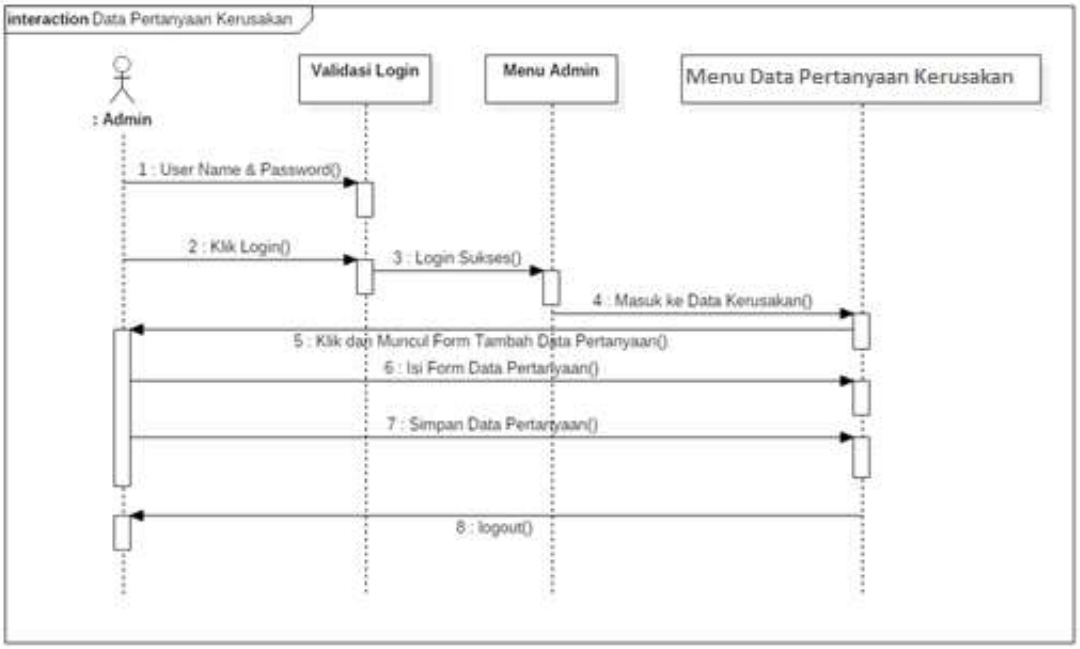
c) Sequence Diagram Edit Data Kerusakan



Gambar 3.9 *Sequence Diagram Edit* Data Kerusakan
Sumber: Data Olahan Peneliti (2017)

Admin melakukan *login*, klik *login* jika *login* sukses *admin* akan masuk ke data kerusakan, disini *admin* muncul form *edit* dan hapus data kerusakan, jika *admin* mengubah, data kerusakan akan mengubah, jika *admin* menghapus, maka data kerusakan akan terhapus.

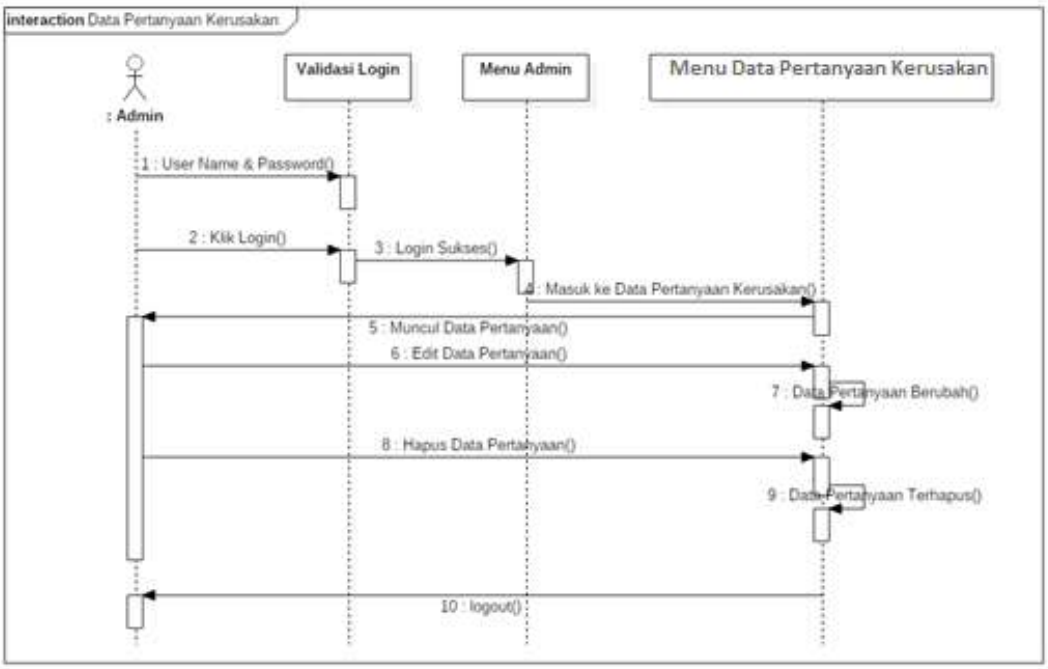
d) Sequence Diagram Input Data Pertanyaan



Gambar 3.10 *Sequence Diagram Input* Data Pertanyaan Kerusakan
Sumber: Data Olahan Peneliti (2017)

Admin melakukan *login*, klik *login* jika *login* sukses *admin* akan masuk ke data pertanyaan, disini *admin* muncul form tambah data pertanyaan, selanjutnya admin akan mengisi form data pertanyaan dan menyimpannya.

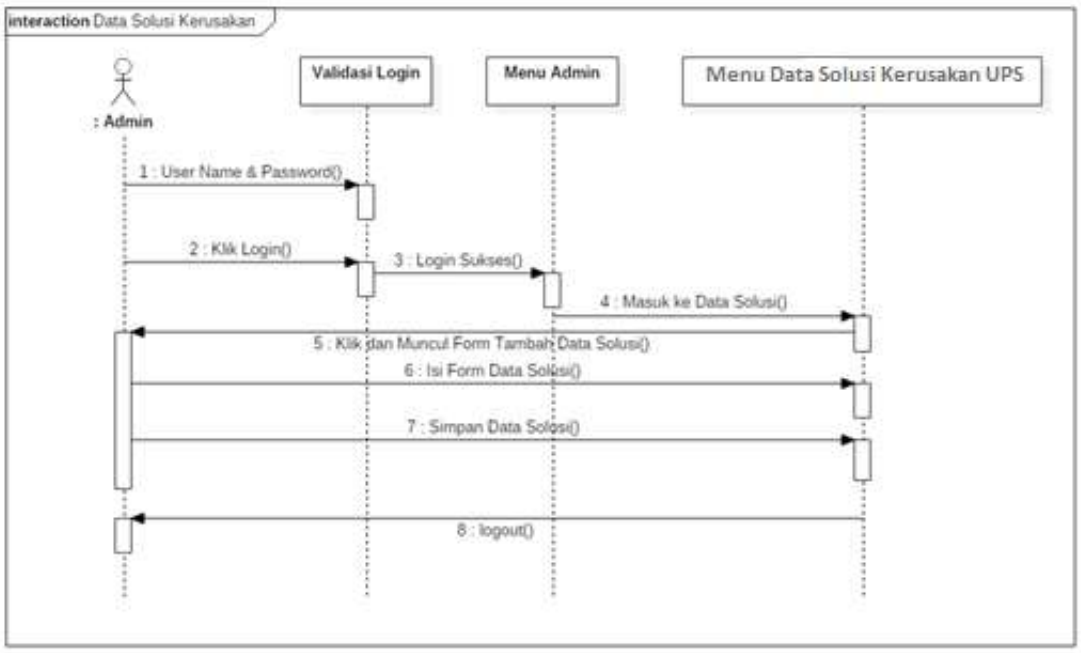
e) Sequence Diagram Edit Data Pertanyaan



Gambar 3.11 *Sequence Diagram Edit* Data Pertanyaan Kerusakan
Sumber: Data Olahan Peneliti (2017)

Admin melakukan *login*, klik *login* jika *login* sukses admin akan masuk ke data pertanyaan, disini *admin* muncul form *edit* dan hapus data pertanyaan, jika *admin* mengubah, data pertanyaan akan mengubah, jika admin menghapus, maka data pertanyaan akan terhapus.

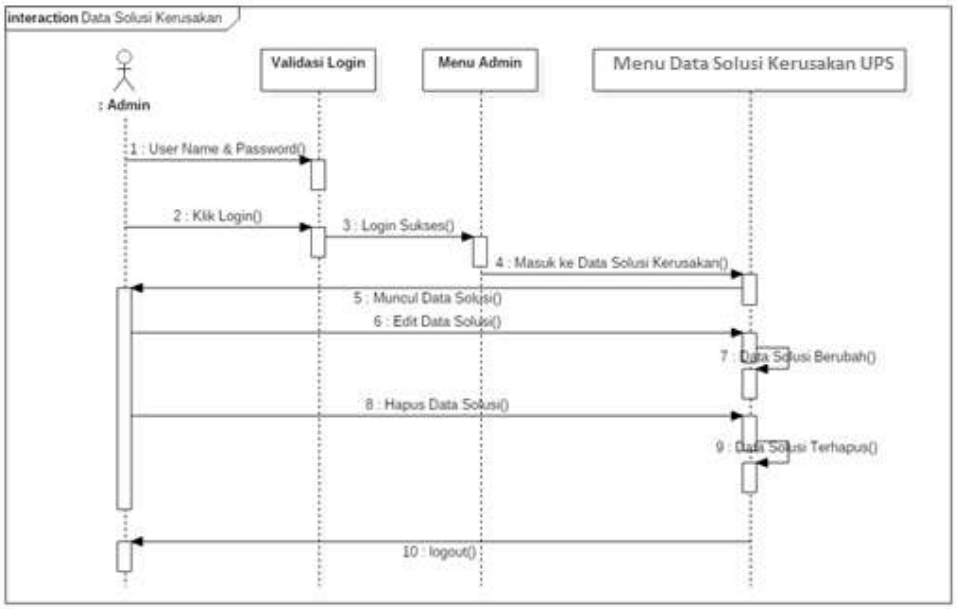
f) Sequence Diagram Input Data Solusi



Gambar 3.12 Sequence *Diagram Input* Data Solusi Kerusakan
Sumber: Data Olahan Peneliti (2017)

Admin melakukan *login*, klik *login* jika *login* sukses *admin* akan masuk ke data solusi, disini *admin* muncul form tambah data solusi, selanjutnya *admin* akan mengisi form data solusi dan menyimpannya.

g) Sequence Diagram Edit Data Solusi

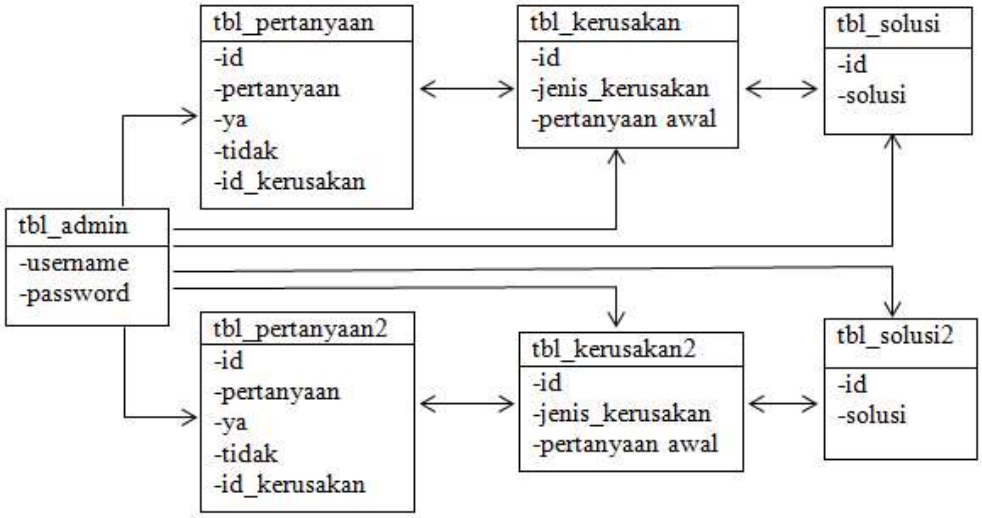


Gambar 3.13 *Sequence Diagram Edit* Data Solusi Kerusakan
Sumber: Data Olahan Peneliti (2017)

Admin melakukan *login*, klik *login* jika *login* sukses *admin* akan masuk ke data solusi, disini *admin* muncul form *edit* dan hapus data solusi, jika admin mengubah, data solusi akan mengubah, jika *admin* menghapus, maka data solusi akan terhapus.

3.4.1.4 *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.



Gambar 3.14 *Class Diagram*
Sumber: Data Olahan Peneliti (2017)

3.4.2 Desain Basis Data

Untuk penyimpanan data dalam sistem pakar mendiagnosa kerusakan mesin manufaktur ini, penulis menggunakan *SQL Server*. Untuk rincian tabel yang terdapat didalam sistem mendiagnosa kerusakan mesin manufaktur ini, dapat melihat tabel-tabel yang dibawah ini.

1. Tabel Basis Data *Admin*

Tabel 3.11 Tabel Basis Data *Admin*

Field	Tipe	Panjang
Username	Varchar	20
Password	Varchar	50

Sumber: Data Penelitian (2017)

2. Tabel Basis Data Kerusakan

Tabel 3.12 Tabel Basis Data Kerusakan

Field	Tipe	Panjang
Id	Varchar	5
Jenis_kerusakan	Text	
Pertanyaan_awal	Varchar	5

Sumber: Data Penelitian (2017)

3. Tabel Basis Data Pertanyaan

Tabel 3.13 Tabel Basis Data Pertanyaan

Field	Tipe	Panjang
Id	Varchar	5
Pertanyaan	Text	
Ya	Varchar	5
Tidak	Varchar	5
Id_kerusakan	Varchar	5

Sumber: Data Penelitian (2017)

4. Tabel Basis Data Solusi

Tabel 3.14 Tabel Basis Data Solusi

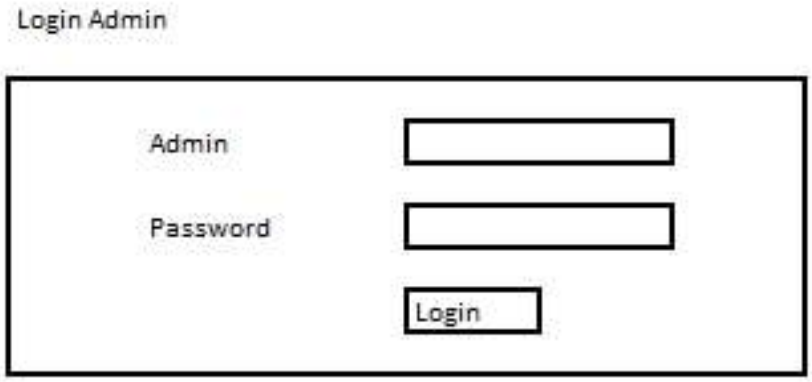
Field	Tipe	Panjang
Id	Varchar	5
Solusi	Text	

Sumber: Data Penelitian (2017)

3.4.3 Desain Antarmuka

1. Menu *Form Login*

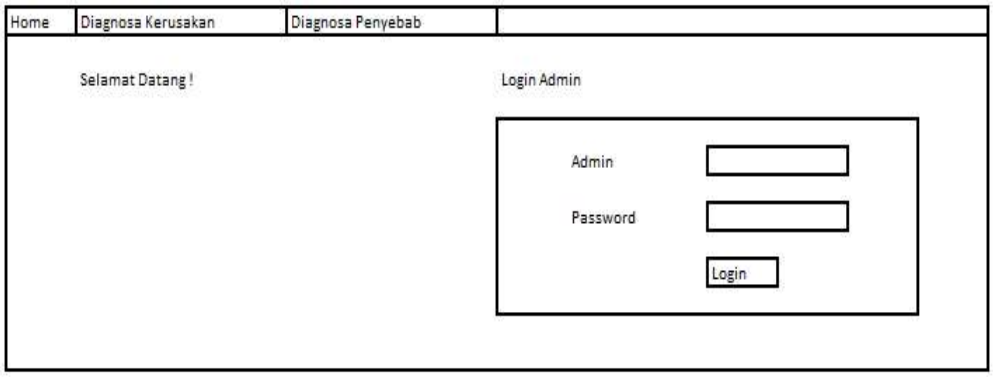
   Form ini menampilkan menu untuk masuk ke dalam aplikasi sistem pakar.



Gambar 3.15 Form Login Aplikasi
Sumber: Data Penelitian (2017)

2. Menu Utama

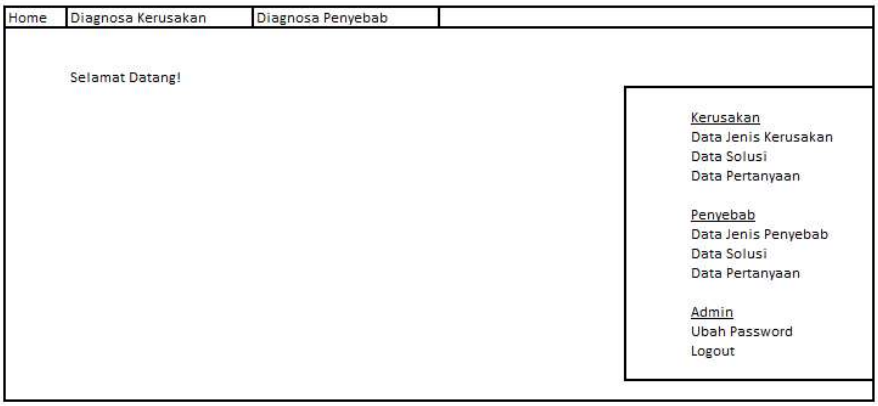
   Menu ini menampilkan menu utama dari sistem *user*.



Gambar 3.16 Menu Utama
Sumber: Data Penelitian (2017)

3. Menu Utama Admin

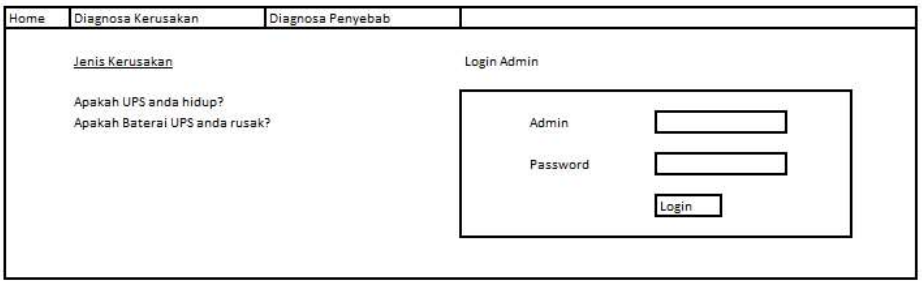
Menu ini menampilkan menu utama setelah login admin.



Gambar 3.17 Menu Utama Admin
Sumber: Data Penelitian (2017)

4. Menu Diagnosa Kerusakan

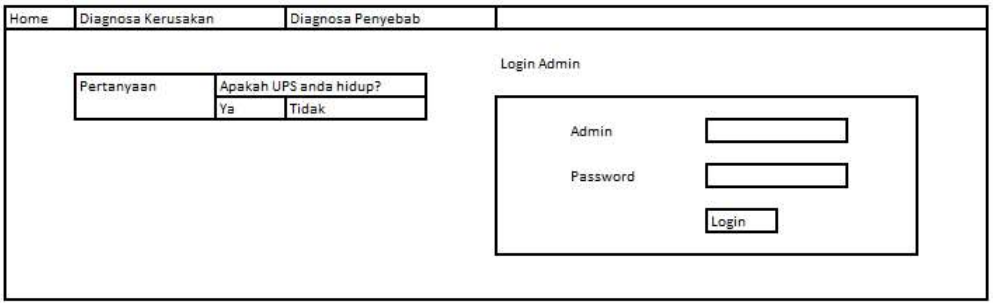
Pada menu ini menampilkan list jenis kerusakan yang telah diinput oleh pakar yang dimana *user* dapat memilih jenis kerusakan untuk di diagnosa.



Gambar 3.18 Menu Diagnosa Kerusakan
Sumber: Data Penelitian (2017)

5. Menu Diagnosa Kerusakan Pilih Pertanyaan

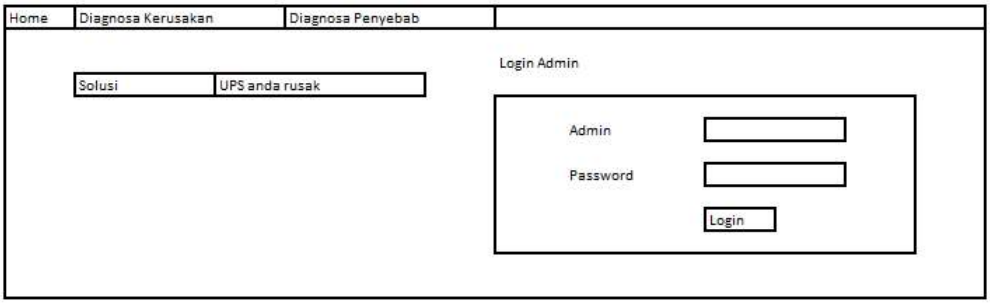
Pada menu ini menampilkan pertanyaan diagnosa dimana *user* dapat menjawab sesuai yang dialami.



Gambar 3.19 Menu Diagnosa Kerusakan Pertanyaan
Sumber: Data Penelitian (2017)

6. Menu Diagnosa Kerusakan Solusi

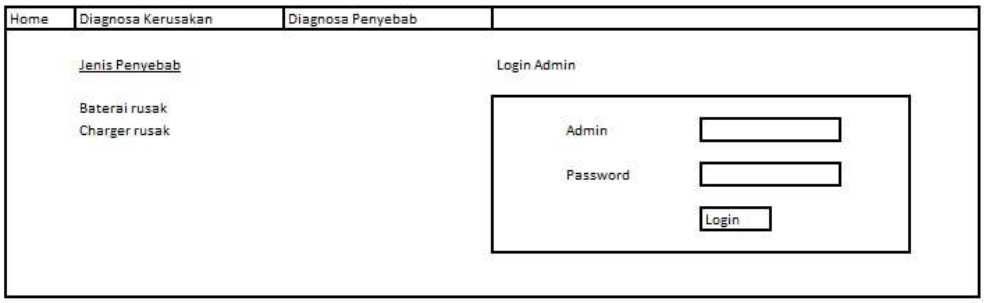
Pada menu ini menampilkan solusi hasil diagnosa.



Gambar 3.20 Menu Diagnosa Kerusakan Solusi
Sumber: Data Penelitian (2017)

7. Menu Diagnosa Penyebab

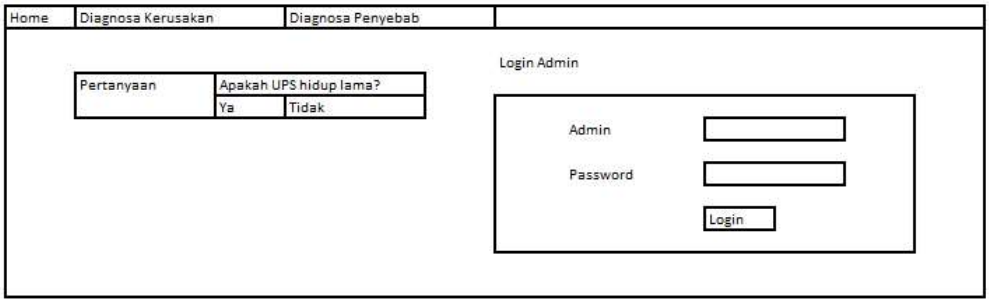
Pada menu ini menampilkan list jenis kerusakan yang telah diinput oleh pakar yang dimana *user* dapat memilih jenis kerusakan untuk di diagnosa.



Gambar 3.21 Menu Diagnosa Penyebab
Sumber: Data Penelitian (2017)

8. Menu Diagnosa Penyebab Pilih Pertanyaan

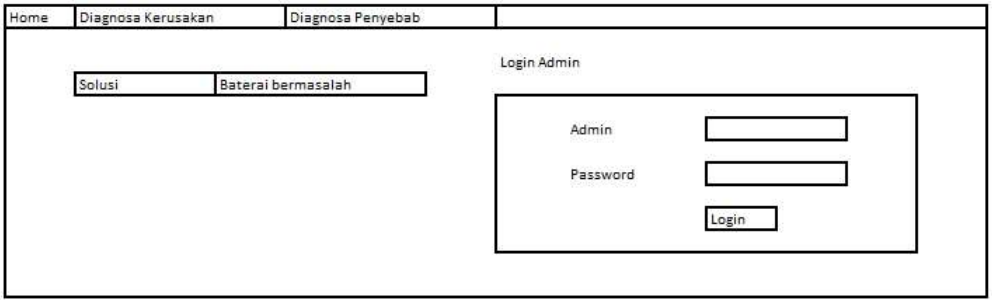
Pada menu ini menampilkan pertanyaan diagnosa dimana *user* dapat menjawab sesuai yang dialami.



Gambar 3.22 Menu Diagnosa Penyebab Pertanyaan
Sumber: Data Penelitian (2017)

9. Menu Diagnosa Penyebab Solusi

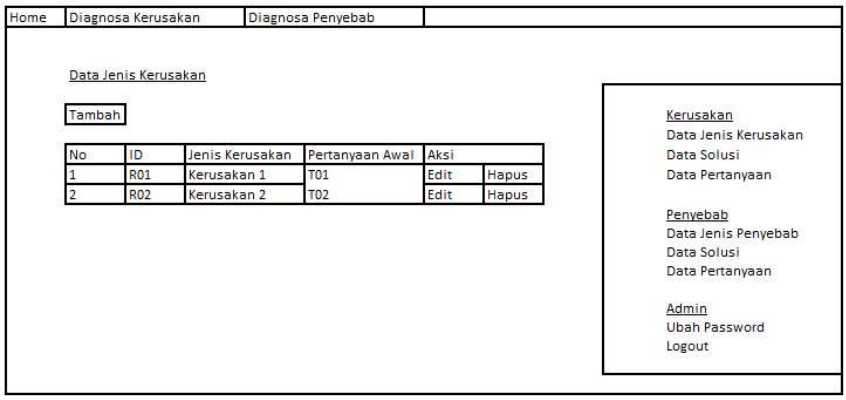
Pada menu ini menampilkan solusi hasil diagnosa.



Gambar 3.23 Menu Diagnosa Penyebab Solusi
Sumber: Data Penelitian (2017)

10. Menu Data Jenis Kerusakan

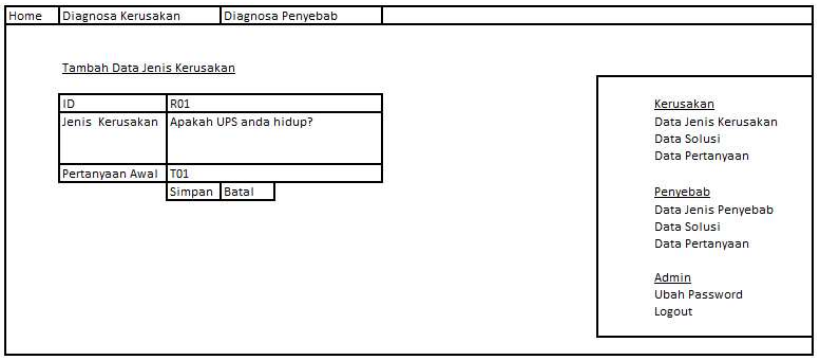
Pada menu ini menampilkan menu data kerusakan yang dimana akan digunakan hanya oleh pihak *admin* untuk menambah, mengubah ataupun menghapus data jenis kerusakan.



Gambar 3.24 Menu Data Jenis Kerusakan
Sumber: Data Penelitian (2017)

11. Menu Tambah Data Jenis Kerusakan

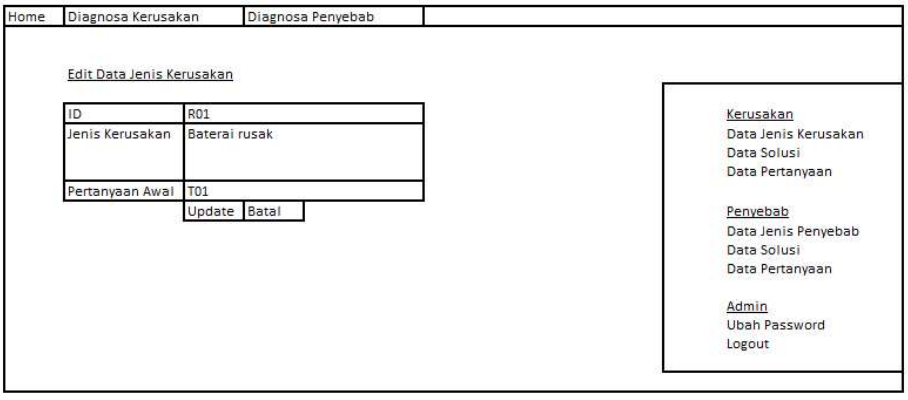
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk menambah data jenis kerusakan.



Gambar 3.25 MenuTambah Data Jenis Kerusakan
Sumber: Data Penelitian (2017)

12. Menu Data Edit Jenis Kerusakan

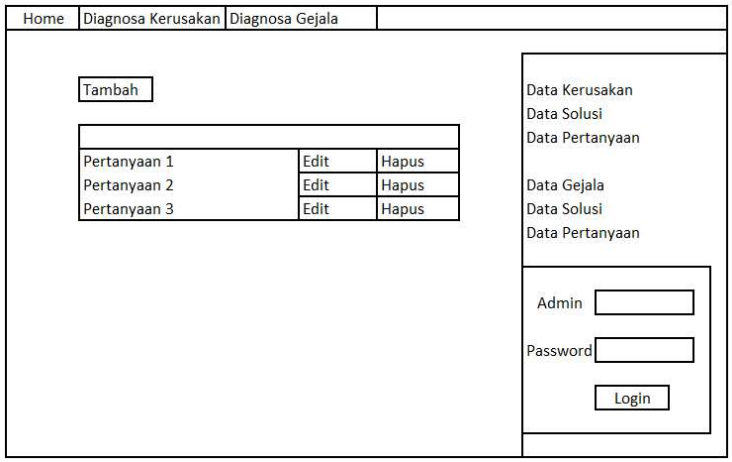
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk mengedit data jenis kerusakan.



Gambar 3.26 Menu *Edit* Data Jenis Kerusakan
Sumber: Data Penelitian (2017)

13. Menu Data Pertanyaan Kerusakan

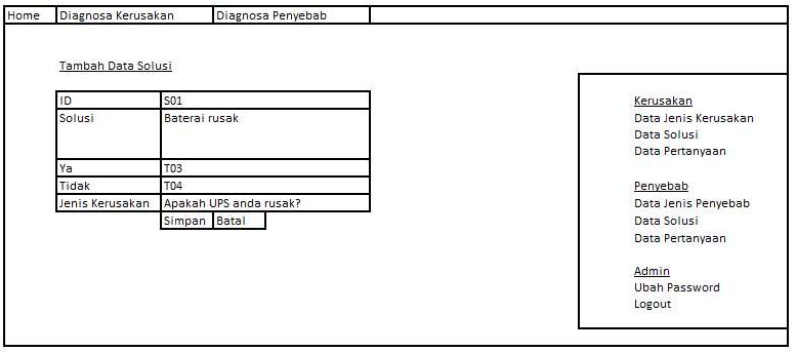
Pada menu ini menampilkan menu data kerusakan yang dimana akan digunakan hanya oleh pihak *admin* untuk menambah, mengubah ataupun menghapus data pertanyaan kerusakan.



Gambar 3.27 Menu Data Pertanyaan Kerusakan
Sumber: Data Penelitian (2017)

14. Menu Tambah Data Pertanyaan Kerusakan

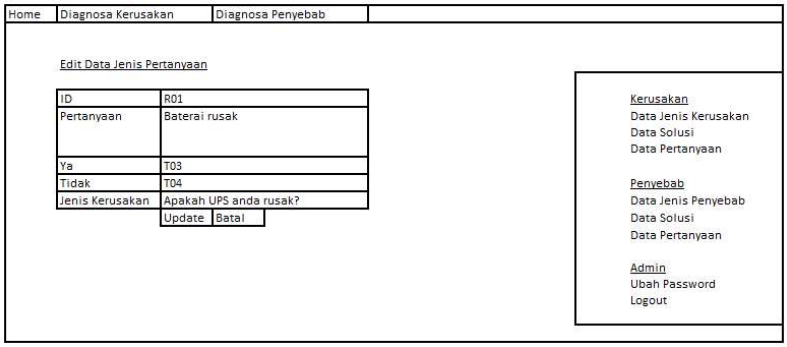
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk menambah data pertanyaan kerusakan.



Gambar 3.28 Menu Tambah Data Pertanyaan Kerusakan
Sumber: Data Penelitian (2017)

15. Menu Edit Data Pertanyaan Kerusakan

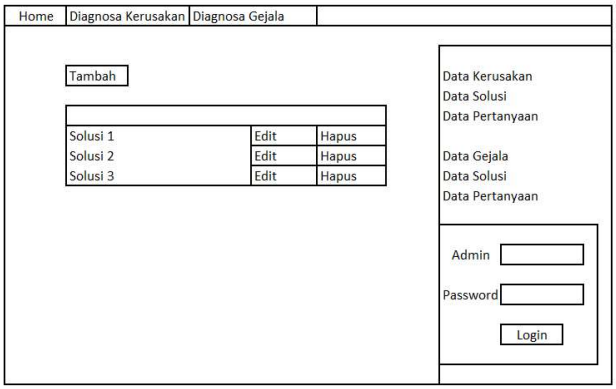
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk mengedit data pernyataan kerusakan.



Gambar 3.29 Menu *Edit* Data Pertanyaan Kerusakan
Sumber: Data Penelitian (2017)

16. Menu Data Solusi Kerusakan

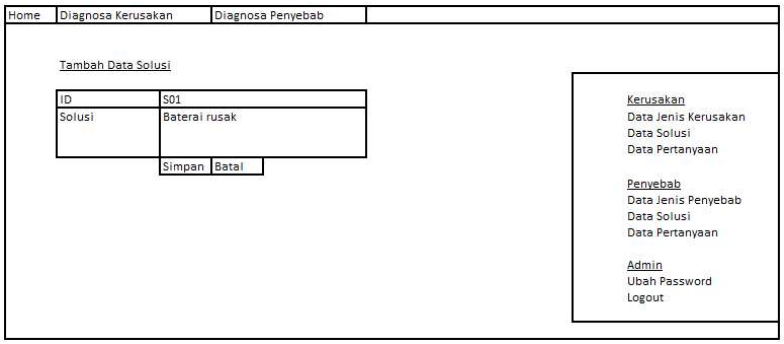
Pada menu ini menampilkan menu data kerusakan yang dimana akan digunakan hanya oleh pihak *admin* untuk menambah, mengubah ataupun menghapus data solusi kerusakan.



Gambar 3.30 Menu Data Solusi Kerusakan
Sumber: Data Penelitian (2017)

17. Menu Tambah Data Solusi Kerusakan

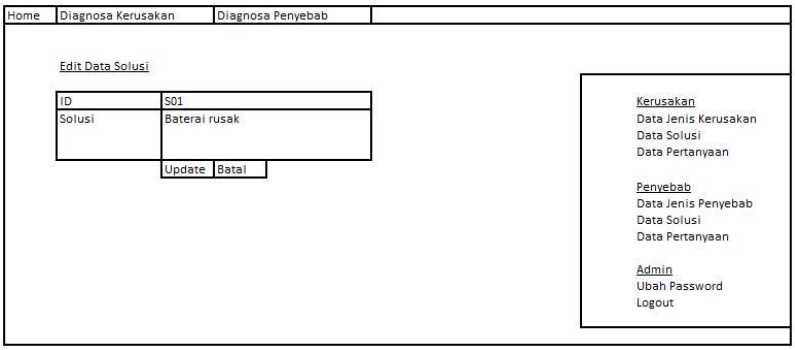
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk menambah data solusi kerusakan.



Gambar 3.31 Menu Tambah Data Solusi Kerusakan
Sumber: Data Penelitian (2017)

18. Menu Edit Data Solusi Kerusakan

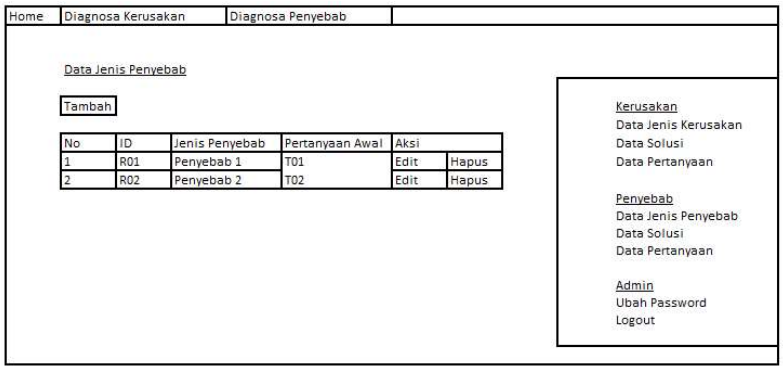
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk mengedit data solusi kerusakan.



Gambar 3.32 Menu *Edit* Data Solusi Kerusakan
Sumber: Data Penelitian (2017)

19. Menu Data Jenis Penyebab

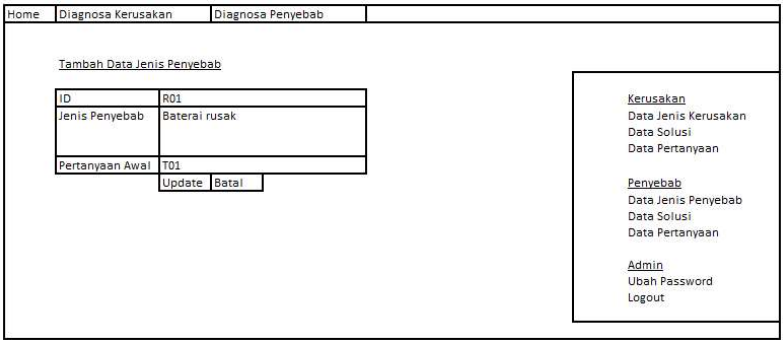
Pada menu ini menampilkan menu data kerusakan yang dimana akan digunakan hanya oleh pihak *admin* untuk menambah, mengubah ataupun menghapus data jenis penyebab sistem pakar



Gambar 3.33: Menu Data Jenis Penyebab
Sumber: Data Penelitian (2017)

20. Menu Tambah Data Penyebab

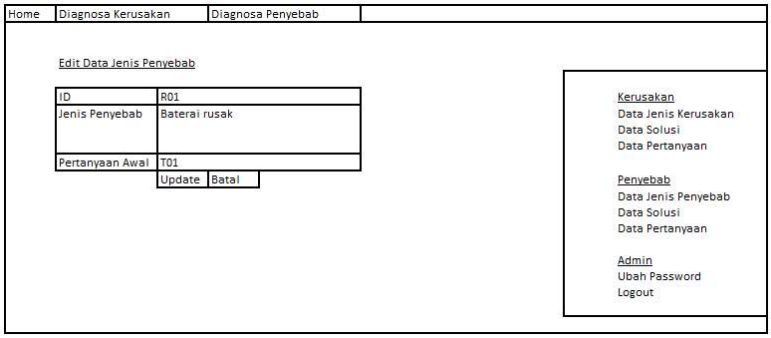
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk menambah data jenis penyebab.



Gambar 3.34 Menu Tambah Data Jenis Penyebab
Sumber: Data Penelitian (2017)

21. Menu Edit Data Penyebab

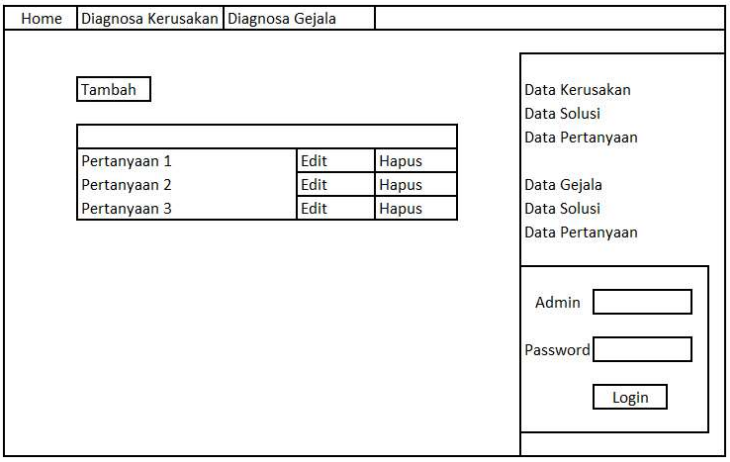
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk mengedit data jenis penyebab.



Gambar 3.35 Menu *Edit* Data Jenis Penyebab
Sumber: Data Penelitian (2017)

22. Menu Data Pertanyaan Penyebab

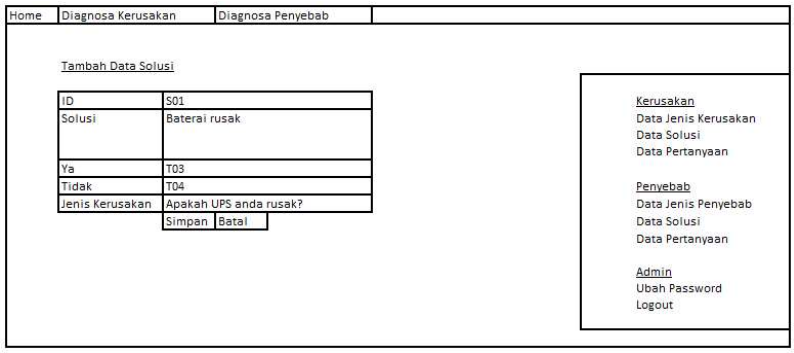
Pada menu ini menampilkan menu data kerusakan yang dimana akan digunakan hanya oleh pihak *admin* untuk menambah, mengubah ataupun menghapus data pertanyaan penyebab.



Gambar 3.36 Menu Data Pertanyaan Penyebab
Sumber: Data Penelitian (2017)

23. Menu Tambah Data Pertanyaan Penyebab

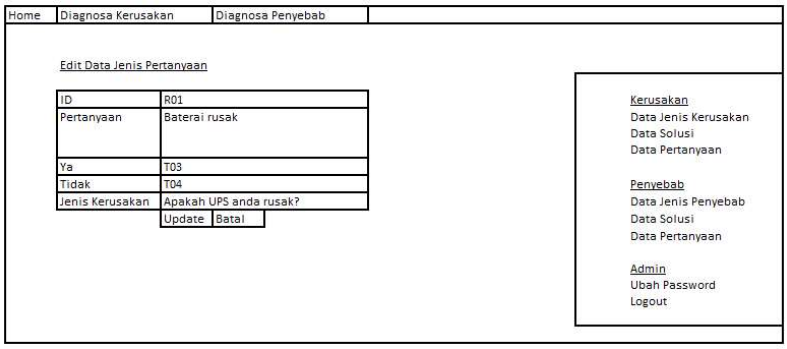
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk menambah data pertanyaan penyebab.



Gambar 3.37 Menu Tambah Data Pertanyaan Penyebab
Sumber: Data Penelitian (2017)

24. Menu Edit Data Pertanyaan Penyebab

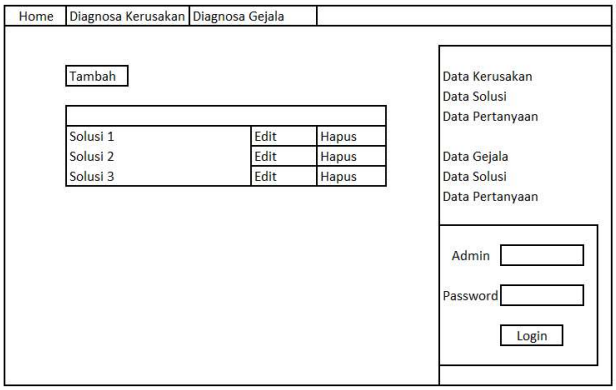
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk mengedit data pernyataan penyebab..



Gambar 3.38 Menu *Edit* Data Pertanyaan Penyebab
Sumber: Data Penelitian (2017)

25. Menu Data Solusi Penyebab

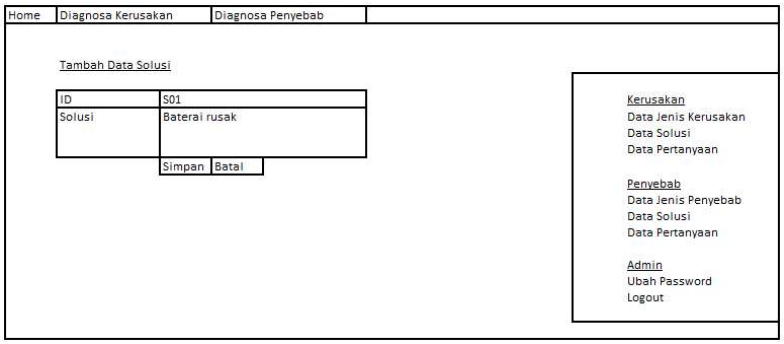
Pada menu ini menampilkan menu data kerusakan yang dimana akan digunakan hanya oleh pihak *admin* untuk menambah, mengubah ataupun menghapus data solusi penyebab.



Gambar 3.39 Menu Data Solusi Penyebab
Sumber: Data Penelitian (2017)

26. Menu Tambah Data Solusi Penyebab

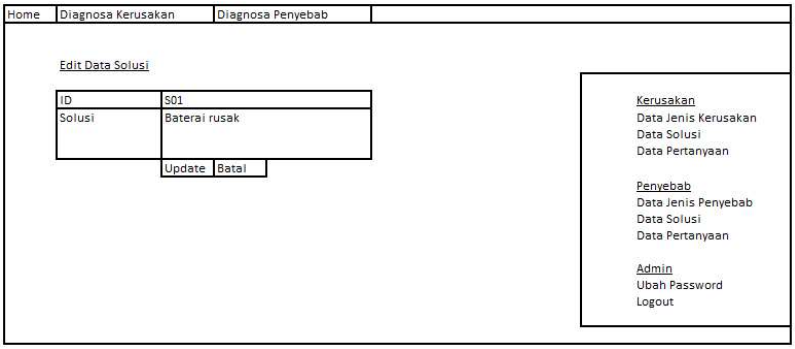
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk menambah data solusi penyebab.



Gambar 3.40 Menu Tambah Data Solusi Penyebab
Sumber: Data Penelitian (2017)

27. Menu Edit Data Solusi Penyebab

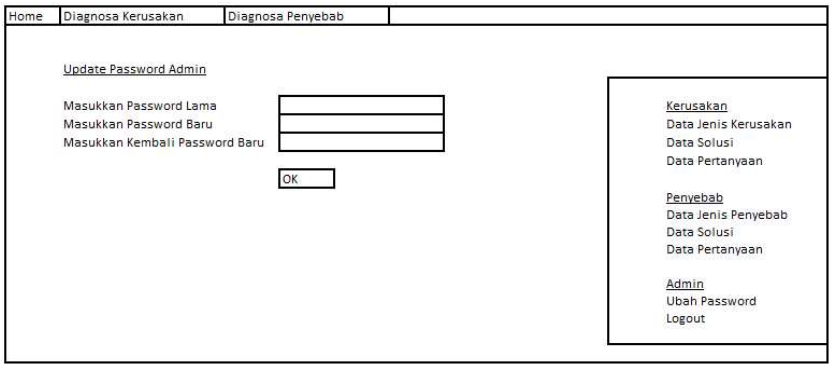
Pada menu ini menampilkan kolom-kolom pengisian yang dapat diisi oleh *admin* untuk mengedit data solusi penyebab.



Gambar 3.41 Menu *Edit* Data Solusi Penyebab
Sumber: Data Penelitian (2017)

28. Menu Ganti *Password*

Menu ini digunakan untuk mengganti *password admin.*



Gambar 3.42 Menu Ganti *Password*
Sumber: Data Penelitian (2017)

3.5 Lokasi Dan Jadwal Penelitian

Penelitian di laksanakan di, penelitian akan dilaksanakan sesudah seminar proposal.

Jadwal / Kegiatan	September				Oktober				November				Desember				Januari				Februari			
	2016				2016				2016				2016				2017				2017			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																								
Pengumpulan Data																								
Analisa Data																								
Perancangan Sistem																								
Pengkodean																								
Uji Coba																								
Penulisan Skripsi																								