

**SISTEM PAKAR MENDIAGNOSA KERUSAKAN
MESIN MANUFAKTUR BERBASIS WEB
MENGUNAKAN METODE
*FORWARD CHAINING***

SKRIPSI



**Oleh:
Nellson Chandra
130210172**

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS PUTERA BATAM
2017**

**SISTEM PAKAR MENDIAGNOSA KERUSAKAN
MESIN MANUFAKTUR BERBASIS WEB
MENGUNAKAN METODE
*FORWARD CHAINING***

SKRIPSI

**Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana**



**Oleh:
Nelson Chandra
130210172**

**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,

Nellson Chandra
NPM: 130210172

**SISTEM PAKAR MENDIAGNOSA KERUSAKAN
MESIN MANUFAKTUR BERBASIS WEB
MENGUNAKAN METODE
*FORWARD CHAINING***

Oleh
Nellson Chandra
130210172

SKRIPSI
Untuk memenuhi salah satu syarat
guna memperoleh gelar Sarjana

Telah disetujui oleh Pembimbing pada tanggal
seperti tertera dibawah ini

Batam, 14 Februari 2017

Joni Eka Candra, S.T., M.T.

Pembimbing

ABSTRAK

Penelitian ini dilatarbelakangi oleh permasalahan seringnya terjadi penundaan dalam produksi apabila terjadi kerusakan pada mesin manufaktur. Hal ini dikarenakan para pengguna mesin tidak mengetahui kerusakan yang dialami oleh mesin yang rusak, mereka harus menunggu hadirnya seorang teknisi untuk memeriksa serta memperbaiki mesin tersebut. Tujuan penelitian ini adalah untuk mengoptimisasi pekerjaan di dalam perusahaan dengan merancang sebuah sistem pakar mesin manufaktur yang dapat menggantikan peran seorang teknisi di dalam perusahaan tersebut. Sistem ini menggunakan metode *forward chaining* yaitu metode inferensi yang menggunakan penalaran yang dimulai dari fakta dahulu untuk menguji kebenaran hipotesis. Sistem ini dibuat dengan menggunakan Bahasa pemrograman *C#* untuk bagian *server side*, *ASP.NET* untuk bagian *client-side*, serta *database SQLSERVER* untuk penyimpanan data. Hasil penelitian menunjukkan bahwa aplikasi sistem pakar tersebut dapat membantu para pengguna mengetahui kerusakan yang dialami mesin manufaktur. Kesimpulan yang bisa diambil adalah pengguna mesin manufaktur dapat berkonsultasi dengan sistem layaknya berkonsultasi langsung dengan seorang pakar untuk mendiagnosa kerusakan yang terjadi pada mesin manufaktur berdasarkan gejala-gejala yang dialami mesin.

Kata Kunci: Sistem Pakar, Mesin Manufaktur, *SQLSERVER*, *C#*, *ASP.NET*, *Forward Chaining*

ABSTRACT

This research is based on the problem that work in production is often halted due to manufacturing machine malfunction. This is caused by the machine users do not know what kind of malfunction is happening to the machine, they have to wait for the technician to check and repair the machine. This research hopefully is able to optimize work in the company by designing an expert system that can replace the technician. This system uses forward chaining method which interference method uses processing that starts with fact to test the hypothesis. This system is created using C# programming language for server-side, and ASP.NET for client-side, and SQLSERVER for database. Result of research shows that expert system application is able to diagnose what kind of breakdown the manufacturing machine is experiencing. The conclusion of the research is that using the application, users can consult about the symptoms of breakdown the machine is having and the system will show the result of diagnose like an expert.

Keyword: *Expert System, Manufacturing Machine, SQLSERVER, C#, ASP.NET, Forward 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. Joni Eka Candra, S.T, M.T. 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 PERNYATAAN	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Identifikasi Masalah	2
1.3. Pembatasan Masalah.....	3
1.4. Perumusan Masalah.....	3
1.5. Tujuan Penelitian.....	4
1.6. Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	5
2.1. Teori Dasar	5
2.1.1 Kecerdasan Buatan (<i>Artificial Intelligence</i>)	6
2.1.2 Tabel Keputusan	14
2.1.3 Pohon Keputusan.....	15
2.2. Variabel Penelitian.....	16
2.2.1 Mesin Manufaktur	17
2.3. <i>Software</i> Pendukung	19
2.3.1 UML (<i>Unified Modeling Language</i>)	19
2.3.2 WEB	28
2.3.3 Visual Studio 2010	30
2.3.4 Database SQL Server.....	32
2.4. Penelitian Terdahulu.....	33
2.5. Kerangka Pemikiran	34

2.5.1 <i>Expert / Pakar</i>	35
2.5.2 <i>User / Pengguna</i>	35
2.5.3 <i>Input Rule / Mesin</i>	35
2.5.4 <i>Input Masalah Mesin</i>	35
2.5.5 <i>Muncul Hasil Diagnosa</i>	36
BAB III METODE PENELITIAN	37
3.1. <i>Desain Penelitian</i>	37
3.2. <i>Operasional Variabel</i>	41
3.2.1. <i>Metode Forward Chaining</i>	41
3.3. <i>Teknik Pengumpulan Data</i>	44
3.3.1. <i>Metode Studi Pustaka</i>	45
3.3.2. <i>Metode Wawancara</i>	45
3.4. <i>Perancangan Sistem</i>	46
3.4.1. <i>UML (Unified Modeling Language)</i>	46
3.4.2. <i>Desain Basis Data</i>	66
BAB IV PEMBAHASAN	70
4.1. <i>Hasil Penelitian</i>	70
4.1.1. <i>Implementasi Antar Muka</i>	70
4.2. <i>Pembahasan</i>	79
4.2.1. <i>Rencana Pengujian</i>	79
4.2.2. <i>Hasil Pengujian</i>	80
BAB V KESIMPULAN DAN SARAN	86
5.1. <i>Kesimpulan</i>	86
5.2. <i>Saran</i>	86
DAFTAR PUSTAKA	87
RIWAYAT HIDUP	
SURAT KETERANGAN PENELITIAN	
LAMPIRAN	

DAFTAR TABEL

Tabel 2.1 Contoh Tabel Keputusan	15
Tabel 2.2 Simbol <i>Use Case Diagram</i>	21
Tabel 2.3 Simbol <i>Activity Diagram</i>	23
Tabel 2.4 Simbol <i>Sequence Diagram</i>	25
Tabel 2.5 Simbol <i>Class Diagram</i>	27
Tabel 3.1 Indikator Kerusakan Mesin Manufaktur (FC)	41
Tabel 3.2 Kode Indikator Kerusakan Mesin Manufaktur (FC)	42
Tabel 3.3 Kode Gejala Kerusakan Mesin Manufaktur	42
Tabel 3.4 Tabel Keputusan Kerusakan Mesin Manufaktur	42
Tabel 3.5 Desain Tabel <i>User</i>	66
Tabel 3.6 Desain Tabel Mesin	67
Tabel 3.7 Desain Tabel Kerusakan	67
Tabel 3.8 Desain Tabel Pertanyaan	68
Tabel 3.9 Desain Tabel Diagnosa	68
Tabel 4.1 Tabel Rencana Pengujian	79
Tabel 4.2 Tabel Pengujian Login	80
Tabel 4.3 Tabel Pengujian Pengolahan Data Mesin	81
Tabel 4.4 Tabel Pengujian Pengolahan Data Kerusakan	82
Tabel 4.5 Tabel Pengujian Pengolahan Data <i>Rule</i>	83
Tabel 4.6 Tabel Pengujian Pengolahan Data <i>User</i>	84
Tabel 4.7 Tabel Pengujian Pilih Mesin Diagnosa Kerusakan	84
Tabel 4.8 Tabel Pengujian Diagnosa Kerusakan	85
Tabel 4.9 Tabel Pengujian Lihat Riwayat Kerusakan	85

DAFTAR GAMBAR

Gambar 2.1. Gambar Struktur Sistem Pakar	8
Gambar 2.2 Contoh Pohon Keputusan	16
Gambar 2.3 Logo StarUML.....	28
Gambar 2.4 Tampilan WEB Google	30
Gambar 2.5 Logo Visual Studio 2008.....	30
Gambar 2.6 Logo SQL Server.....	32
Gambar 2.7 Kerangka Pemikiran Penelitian	34
Gambar 3.1 Desain Penelitian	38
Gambar 3.2 Pohon Keputusan.....	44
Gambar 3.3 <i>Use Case Diagram</i>	46
Gambar 3.4 <i>Class Diagram</i>	47
Gambar 3.5 <i>Activity Diagram</i> Login	48
Gambar: 3.6 <i>Activity Diagram</i> Tambah Mesin	49
Gambar: 3.7 <i>Activity Diagram</i> Ubah Mesin.....	50
Gambar 3.8 <i>Activiy Diagram</i> Tambah User.....	51
Gambar 3.9 <i>Activiy Diagram</i> Ubah User	52
Gambar 3.10 <i>Activiy Diagram</i> Tambah Rule	53
Gambar 3.11 <i>Activity Diagram</i> Ubah Rule	54
Gambar 3.12 <i>Activity Diagram</i> Tambah Kerusakan	55
Gambar 3.13 <i>Activity Diagram</i> Ubah Kerusakan.....	56
Gambar 3.14 <i>Activity Diagram</i> Diagnosa.....	57
Gambar 3.15 <i>Activity Diagram</i> Lihat Laporan Kerusakan.....	58
Gambar 3.16 <i>Sequence Diagram</i> Login	59
Gambar 3.17 <i>Sequence Diagram</i> Mesin.....	60
Gambar 3.18 <i>Sequence Diagram</i> Kerusakkan.....	61
Gambar 3.19 <i>Sequence Diagram</i> Rule	62
Gambar 3.20 <i>Sequence Diagram</i> User	63
Gambar 3.21 <i>Sequence Diagram</i> Diagnosa Kerusakan.....	64
Gambar 3.22 <i>Sequence Diagram</i> Lihat Laporan Kerusakkan.....	65

Gambar 4.1 <i>Form Login</i>	70
Gambar 4.2 <i>Form Menu Utama</i>	71
Gambar 4.3 <i>Form Mengelola Data Mesin</i>	72
Gambar 4.4 <i>Form Mengelola Data Kerusakan</i>	73
Gambar 4.5 <i>Form Mengelola Data Rule</i>	74
Gambar 4.6 <i>Form Mengelola Data User</i>	74
Gambar 4.7 <i>Form Pilih Mesin</i>	75
Gambar 4.8 <i>Form Diagnosa Mesin</i>	76
Gambar 4.9 <i>Form Hasil Diagnosa</i>	77
Gambar 4.10 <i>Form Lihat Riwayat Diagnosa</i>	78

BAB I

PENDAHULUAN

1.1. Latar Belakang

PT Multi Karya Bajatama adalah sebuah perusahaan telah berdiri sejak 19 Oktober 1998 dan bergerak di dalam bidang memproduksi serta mengolah bahan logam yang kemudian akan dijual kepada pembelinya. Untuk dapat mengolah bahan logam dengan cepat dan presisi tinggi, maka pastinya harus menggunakan mesin.

Meskipun untuk mengolah bahan logam tersebut menggunakan mesin, tentunya memerlukan sumber daya manusia untuk mengontrol mesin tersebut. Setelah 5 tahun, biasanya ketahanan mesin akan menyusut dan akan sering mengalami kerusakan. Apabila mesin pada PT Multi Karya Bajatama mengalami kerusakan, pengguna mesin tersebut harus menunggu kehadiran teknisi mesin untuk mengetahui kerusakan yang dialami mesin serta memperbaikinya, hal ini dikarenakan minimnya pengetahuan teknisi perusahaan tersebut tentang mesin yang dia gunakan. Hal ini tentunya merugikan perusahaan karena produksi mesin tersebut harus terhenti hingga selesai di perbaiki. Serta biaya untuk melakukan perbaikan mesin yang rusak sangat tinggi.

Untuk membantu PT Multi Karya Bajatama supaya dapat mengurangi waktu perbaikan mesin dan menghemat biaya perbaikan, penulis akan merancang sebuah sistem yang menggunakan metode sistem pakar sehingga para pengguna mesin tersebut dapat mengetahui kerusakan yang dialami oleh mesin dengan dibantu oleh sistem ini tanpa harus menunggu hadirnya seorang teknisi.

Berdasarkan paparan diatas, penulis ingin mengangkat penelitian dengan judul **“SISTEM PAKAR MENDIAGNOSA KERUSAKAN MESIN MANUFAKTUR BERBASIS WEB MENGGUNAKAN METODE FORWARD CHAINING”**

1.2. Identifikasi Masalah

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

- a. Kurangnya pengetahuan mengenai jenis kerusakan yang terjadi pada mesin manufaktur.
- b. Belum adanya aplikasi sistem pakar yang dapat mendeteksi kerusakan pada mesin manufaktur
- c. Tidak dapat melihat riwayat kerusakan mesin yang dapat membantu membuat keputusan mengganti mesin tersebut.

1.3. Pembatasan Masalah

Adapun batasan masalah pada tugas akhir ini adalah:

- a. Penulis hanya menganalisis kerusakan pada mesin yang terdapat di PT Multi Karya Bajatama, tidak pada PT lain.
- b. Sistem pakar dibuat dengan menggunakan metode *Forward Chaining*.
- c. Sistem pakar dirancang dengan menggunakan bahasa pemrograman *ASP.NET*, *C#* dengan *database SQL Server*.
- d. Data diambil dengan melakukan wawancara dengan teknisi dan mengambil data dari referensi buku, jurnal dan internet.

1.4. Perumusan Masalah

Adapun rumusan masalah dalam penelitian ini adalah:

- a. Bagaimana sebuah metode *Forward Chaining* dapat mengetahui jenis kerusakan yang terjadi pada mesin manufaktur?
- b. Bagaimana metode *Forward Chaining* dan bahasa pemrograman *C#* dapat mendeteksi kerusakan pada mesin manufaktur?
- c. Bagaimana sebuah sistem pakar dapat menyimpan serta menampilkan riwayat kerusakan yang dialami mesin tersebut?

1.5. Tujuan Penelitian

Adapun tujuan penelitian ini adalah:

- a. Merancang sebuah sistem pakar yang akan membantu pengguna mesin manufaktur untuk dapat mengetahui kerusakan yang dialami mesin tersebut.
- b. Menjelaskan ciri-ciri kerusakan pada mesin manufaktur dengan menggunakan *Forward Chaining*.
- c. Menerapkan metode *Forward Chaining* dan bahasa pemrograman ASP.NET dan C# untuk mendeteksi kerusakan pada mesin manufaktur.

1.6. Manfaat Penelitian

Berdasarkan uraian diatas, manfaat penelitian di bagi menjadi dua sub poin terpenting yang diberikan dari hasil perancangan sistem yaitu:

- a. Aspek Teoritis

Untuk mengetahui secara dalam tentang sistem pakar dengan metode *Forward Chaining* dan cara system pakar metode *Forward Chaining* mendiagnosa kerusakan mesin manufaktur.

- b. Aspek Praktis

Hasil penelitian ini bisa dijadikan sebagai media informasi untuk masyarakat serta menjadi alat diagnosa kerusakan mesin manufaktur pada PT Multi Karya Bajatama.

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. (Sudaryono, 2015:13)

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.

2.1.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan berasal dari bahasa Inggris “*Artificial Intelligence*” atau disingkat AI, yaitu *intelligence* adalah kata sifat yang berarti cerdas, sedangkan *artificial* artinya buatan. Kecerdasan buatan yang dimaksud disini merujuk pada mesin yang mampu berpikir, menimbang tindakan yang diambil dan mampu mengambil keputusan seperti yang dilakukan oleh manusia.

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 matematika, menulis puisi, mengemudikan mobil, dan melakukan diagnosis penyakit. AI relevan dengan berbagai macam *task* kecerdasan, AI merupakan sebuah ilmu yang universal (Budiharto & Suhartono, 2014:2). Adapun metode-metode *Artificial Intelligence* (AI) sebagai berikut:

2.1.1.1. Sistem Pakar atau *Expert System*

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 (Kusrini, 2008: 3).

Sistem pakar adalah sistem komputer yang ditujukan untuk meniru semua aspek kemampuan pengambilan keputusan seorang pakar. Sistem pakar memanfaatkan secara maksimal pengetahuan khusus selayaknya seorang pakar untuk memecahkan masalah(Rosnelly, 2012:2).

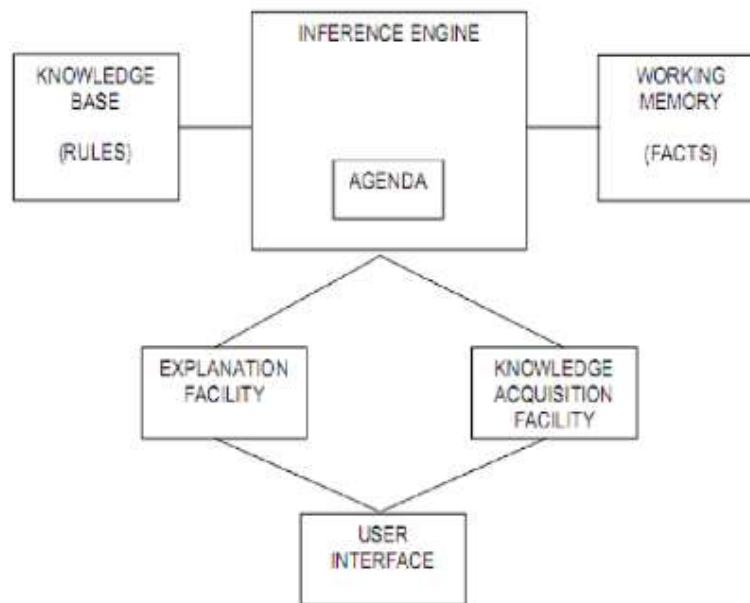
Pakar atau ahli didefinisikan sebagai seorang yang memiliki pengetahuan atau keahlian khusus yang tidak dimiliki oleh kebanyakan orang. Seorang pakar dapat memecahkan masalah yang tidak mampu dipecahkan kebanyakan orang. Dengan kata lain, dapat memecahkan suatu masalah dengan lebih efisien namun bukan berarti lebih murah. Pengetahuan yang dimuat ke dalam sistem pakar dapat berasal dari seorang pakar atau pun pengetahuan yang berasal dari buku, jurnal, majalah, dan dokumentasi yang dipublikasi lainnya, serta orang yang memiliki pengetahuan meskipun bukan ahli. Istilah sistem pakar sering disinonimkan dengan sistem berbasis pengetahuan.

a. **Kelebihan Sistem Pakar**

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti Meningkatkan ketersediaan, Mengurangi Biaya, Permanen, Keahlian multipel, Meningkatkan kehandalan, Penjelasan, Respon yang cepat, Stabil, Pembimbing pintar dan Basis data yang cerdas.

b. Struktur Sistem Pakar

Adapun struktur sistem pakar dapat dilihat pada gambar berikut (Rosnelly, 2012:13).



Gambar 2.1. Gambar Struktur Sistem Pakar
Sumber: Rosnelly (2012)

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

1. *Knowledge Base* (Base Pengetahuan)

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.

2. *Inference Engine* (Mesin Inferensi)

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*.

3. *Working Memory*

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.

4. *Explanation Facility*

Menyediakan kebenaran dari solusi yang dihasilkan kepada user (*reasoning chain*).

5. *Knowledge Acquisition Facility*

Meliputi proses pengumpulan, pemindahan dan perubahan dari kemampuan pemecahan masalah seorang pakar atau sumber pengetahuan terdokumentasi ke

program computer, yang bertujuan untuk memperbaiki atau mengembangkan basis pengetahuan.

6. *User Interface*

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.

c. ***Interference Engine* atau Mesin Interferensi**

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis (*logical conclusion*) atau implikasi berdasarkan informasi yang tersedia (Kusrini, 2008:8). Dalam sistem pakar proses inferensi dilakukan dalam suatu modul yang disebut *Inference Engine* (Mesin Inferensi).

Pada sistem pakar berbasis *rule*, domain pengetahuan direpresentasikan dalam sebuah kumpulan *rule* berbentuk IF-THEN, sedangkan data direpresentasikan dalam sebuah kumpulan fakta-fakta tentang kejadian saat ini. Mesin interferensi membandingkan masing-masing *rule* yang tersimpan dalam basis pengetahuan dengan fakta-fakta yang terdapat dalam *database*. Jika bagian IF (kondisi) dari *rule* cocok dengan fakta, maka *rule* dieksekusi dan bagian THEN (aksi) diletakkan dalam *database* sebagai fakta baru yang ditambahkan (Sutojo, 2011:171)

Ada dua metode inferensi yang penting dalam sistem pakar yaitu:

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).

Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (*controlling*) dan peramalan (*prognosis*) (Giarattano dan Riley, 1994).

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.

Tujuan inferensi adalah mengambil pilihan terbaik dari banyak kemungkinan. Metode inferensi runut balik ini cocok digunakan untuk memecahkan masalah diagnosi (Schnupp, 1989).

d. **Karakteristik Sistem Pakar.**

Sistem pakar umumnya dirancang untuk memenuhi beberapa karakteristik umum yaitu Kinerja sangat baik, Waktu respon yang baik, Dapat diandalkan, Dapat dipahami dan Fleksibel.

2.1.1.2. Jaringan Sistem Syaraf

Jaringan syaraf tiruan (JST) sistem pemrosesan informasi yang memiliki karakteristik serupa dengan jaringan syaraf biologis dan terdiri dari sejumlah besar elemen pemrosesan sederhana yang disebut neuron, unit, sel, atau node (Irwansyah & Faisal, 2015:42). Ada 3 hal yang diadopsi dari sistem syaraf otak manusia oleh JST adalah Neuron, Topologi, Toleransi Kesalahan (*Fault Tolerant*). Menurut Russel dan Norvig (2003), JST memiliki proses belajar yang berbeda dengan metode yang lainnya, sehingga *Neural Network* memiliki struktur yang unik dimana metode pelatihannya dibagi ke dalam dua kelompok, yakni sebagai berikut:

a. *Feed-Forward Neural Network* (FFNN)

FFNN adalah jaringan syaraf tiruan dimana hubungan antara sinyal informasinya bergerak hanya satu arah saja dalam mengasosiasikan *input* dengan *output* yang ekstensif digunakan dalam pengenalan pola (Irwansyah & Faisal, 2015:47).

b. *Feed-Back Neural Network* (FBNN)

FBNN merupakan jaringan syaraf tiruan yang memiliki sinyal informasi yang bergerak dari kedua arah pada *layer* jarignannya (Irwansyah & Faisal, 2015:49).

2.1.1.3. Logika *Fuzzy*

Logika *Fuzzy* merupakan salah satu cabang dari bidang *soft computing*. Logika *Fuzzy* merupakan suatu teori himpunan logika yang dikembangkan untuk mengatasi konsep nilai yang terdapat diantara kebenaran (*true*) dan kesalahan (*false*) (Irwansyah & Faisal, 2015:31).

Terdapat 3 metode dalam logika *fuzzy*, yaitu:

a. Metode Mamdani

Sering dikenal sebagai metode max-min. Untuk mendapatkan output diperlukan berbagai tahapan (Pusadan, 2014:14) :

1. Pembentukan himpunan *fuzzy*
2. Aplikasi fungsi implikasi (aturan)
3. Komposisi aturan

b. Metode Sugeno

Metode Sugeno adalah penalaran dengan metode output (konsekuen) sistem tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan linear (Rofiq, 2013:6).

c. Metode Tsukamoto

Pada metode ini, setiap aturan direpresentasikan menggunakan himpunan-himpunan *fuzzy*, dengan fungsi keanggotaan yang monoton. Untuk menentukan nilai *output crisp* atau hasil yang tegas, dicari dengan cara mengubah *input* menjadi sebuah bilangan pada domain himpunan *fuzzy* tersebut (Sholihin, Fuad, & Khamiliyah, 2013:501).

2.1.2 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 & Herfan, 2008:26).

Tabel 2.1 Contoh Tabel Keputusan

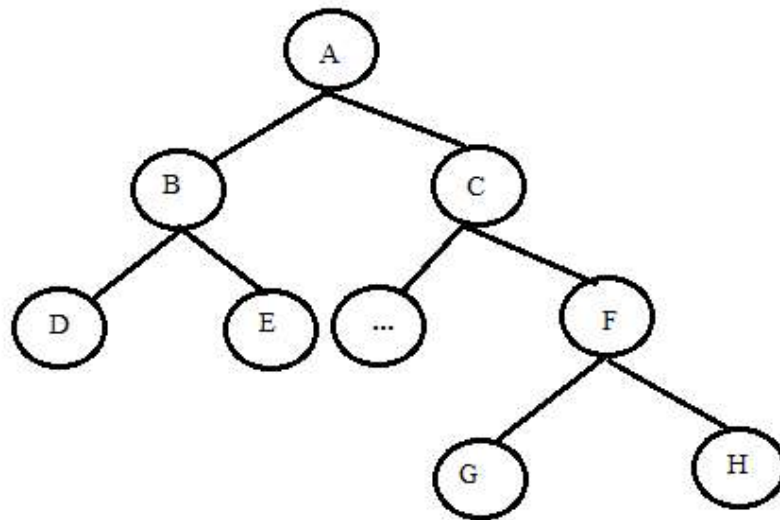
Hipotesa Indikator	Hipotesa1	Hipotesa2	Hipotesa3	Hipotesa4
Indikator A	Ya	Ya	Tidak	Ya
Indikator B	Ya	Tidak	Ya	Tidak
Indikator C	Ya	Tidak	Ya	Tidak
Indikator D	Tidak	Ya	Tidak	Ya

Sumber: Data Penelitian (2017)

2.1.3 Pohon Keputusan

Sebuah pohon keputusan adalah sebuah struktur yang dapat digunakan untuk membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan keputusan. Dengan masing-masing rangkaian pembagian, anggota himpunan hasil menjadi mirip satu dengan yang lain (Berry & Linoff, 2004 dalam buku (Kusrini & Luthfi, 2009:14)).

Sebuah pohon keputusan mungkin dibangun dengan saksama secara manual atau dapat tumbuh secara otomatis dengan menerapkan salah satu atau beberapa algoritma pohon keputusan untuk memodelkan himpunan data yang belum terklasifikasi.



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

2.2. Variabel Penelitian

Bendatu (2015: 42) 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 (Swarjana, 2015: 42).

Variabel yang akan digunakan oleh peneliti dalam penelitiannya ada sebagai berikut:

2.2.1 Mesin Manufaktur

Mesin Manufaktur adalah tulang punggung dari setiap negara industri. Staff teknikal dan manufaktur dalam industri harus tau berbagai macam proses manufaktur, bahan yang di proses, alat yang diperlukan untuk manufaktur komponen berbeda atau produk dengan proses rencana keamanan yang optimal untuk menghindari kecelakaan. (Singh, 2006:1)

Indikator kerusakan yang terdapat pada mesin manufaktur adalah sebagai berikut:

1. Pompa air pendingin

Merupakan sebuah tipe pompa yang digunakan untuk mensirkulasi pendingin, biasanya dalam bentuk air. (Singh, 2006:148)

2. Minyak pelumas

Merupakan sebuah zat yang digunakan untuk mengurangi gesekan antara dua permukaan yang saling bersentuhan. (Singh, 2006:290)

3. Sparepart

Merupakan sebuah bagian yang dapat ditukarkan yang disimpan di sebuah gudang dan digunakan untuk memperbaiki atau menggantikan unit yang gagal. (Singh, 2006:467)

2.3. *Software Pendukung*

Software pemrograman biasanya dipakai untuk memudahkan para pembuat program (*programmer*) untuk menulis program yang kemudian dibentuk menjadi sebuah objek yang bisa diakses oleh *system software* dalam bentuk aplikasi. *Software* pendukung yang digunakan untuk merancang sistem aplikasi ini adalah:

2.3.1 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 analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa & Shalahuddin, 2011, p. 113)

Berikut adalah beberapa jenis diagram UML:

2.3.1.1. *Use Case Diagram*

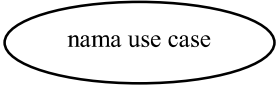
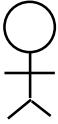



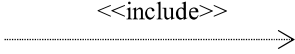
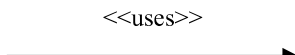
Use case diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem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

Use case atau diagram *use case* merupakan pemodelan 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 (2) hal utama pada use case yaitu pendefinisian apa yang disebut *actor* dan *use case* (Rosa & Shalahuddin, 2015:155)

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

Tabel 2.1 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use Case</i></p>  <p>nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
<p>Aktor / <i>actor</i></p>  <p>nama aktor</p>	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 aktor
<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan aktor
<p>Ekstensi / <i>extend</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
<p>Generalisasi / <i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
<p>Menggunakan / <i>include</i> / <i>uses</i></p>  	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

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

2.3.1.2. 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.


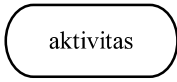



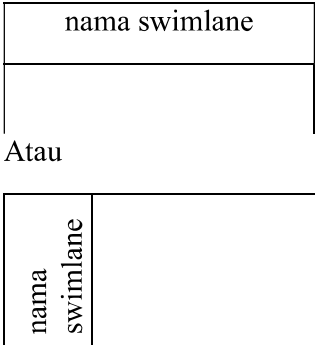
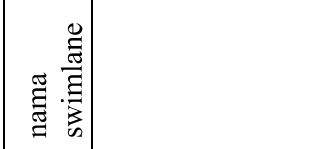
Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut: (Rosa & Shalahuddin, 2015:161)

- 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 / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	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)

2.3.1.3. *Sequence Diagram*



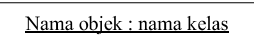

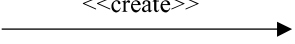
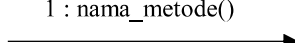
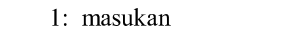
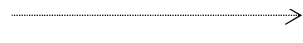
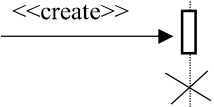
Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima

antara objek. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

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.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Aktor / <i>actor</i></p> 	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
<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan
<p>Pesan tipe <i>create</i></p> 	Objek yang lain, arah panah mengarah pada objek yang dibuat
<p>Pesan tipe <i>call</i></p> 	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
<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
<p>Pesan tipe <i>return</i></p> 	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
<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

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

2.3.1.4. 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. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut (Rosa & Shalahuddin, 2015: 165):

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

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

Untuk membuat diagram-diagram diatas, penulis menggunakan aplikasi yang bernama StarUML. StarUML adalah *platform* pemodelan perangkat lunak yang mendukung UML (*Unified Modeling Language*). StarUML yang berbasis pada UML versi 1.4, menyediakan sebelas jenis *Diagram* yang berbeda, dan mendukung notasi UML 2.0 StarUML juga secara aktif mendukung pendekatan 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: 1).



Gambar 2.3 Logo StarUML
Sumber: Data Penelitian (2017)

2.3.2 WEB

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses dimana saja dengan biaya yang relatif murah. *Web* merupakan bentuk implementasi dari bahasa pemrograman *web* (*web programming*). Sejarah perkembangan bahasa pemrograman web diawali dengan munculnya HTML (*Hypertext Markup Language*), yang kemudian dikembangkan dengan munculnya CSS (*Cascading Style Sheet*) yang bertujuan untuk memperindah tampilan *website*. Pada kenyataannya, bahasa tersebut masih memiliki kelemahan, yaitu *script* program dapat dilihat secara utuh, sehingga anda dapat mengetahui kerangka situs tersebut dan memberi peluang *hacker* untuk mengubahnya dengan mudah.

Bahasa pemrograman saat ini sudah sangat berkembang dengan berbagai kemudahan dalam penyajian dan *interface* yang lebih *user friendly*. Penyajian yang baik dari suatu bahasa pemrograman akan menghasilkan sebuah *web* dinamis sehingga pengunjung akan lebih mudah mendapatkan informasi yang dibutuhkan.

Beberapa bahasa pemrograman *web* berbasis *server* (*server side*) mulai muncul dan dikembangkan oleh beberapa perusahaan perangkat lunak, seperti: ASP (*Active Server Pages*) oleh *Microsoft*, JSP (*Java Server Pages*) oleh *Sun Microsystems*, dan CGI (*Common Gateway Interface*).

Bahasa pemrograman tersebut memiliki beberapa kendala, misalnya: *Active Server Pages* keluaran *Microsoft* hanya dapat dijalankan melalui sistem operasi buatan *Microsoft*, dan *Common Gateway Interface* yang hanya berjalan pada sistem operasi berbasis UNIX. Untuk mengatasi keterbatasan tersebut, muncullah bahasa pemrograman web yang dapat dijalankan pada sistem operasi *Microsoft*, sekaligus sistem operasi berbasis UNIX. Yaitu: bahasa pemrograman PHP (*PHP Hypertext Preprocessor*). (Andi, 2009: 2)



Gambar 2.4 Tampilan WEB Google
Sumber: Data Penelitian (2017)

2.3.3 Visual Studio 2010

Visual Studio adalah *Integrated Development Environment* (IDE) dari untuk membangun aplikasi console dan *Graphical User Interface* (GUI) dengan menggunakan bahasa yang didukung pada *.NET Framework*. Aplikasi GUI yang dapat dibangun diantaranya adalah *Windows Form*, *Website*, *Web Application*, *Windows Mobile*.



Gambar 2.5 Logo Visual Studio 2008
Sumber: Data Penelitian (2017)

Dengan menggunakan Visual Studio 2010, penulis menuliskan *coding* aplikasi menggunakan bahasa *programming* sebagai berikut.

a. Bahasa Pemrograman C#

C# merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka *.NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic dan lain-lain dengan beberapa penyederhanaan.

Bahasa Pemrograman C# (baca: *C-Sharp*) dirancang oleh *Microsoft Corp* sebagai bahasa pemrograman yang sangat berdaya-guna, aman (*secure*), serta mudah digunakan. Bahasa pemrograman C# juga dapat digunakan untuk mengembangkan aplikasi sarana bergerak (*mobile application*), aplikasi berbasis *Web* (*Web-based applications*), serta aplikasi berskala besar (*enterprise*).

b. Bahasa Pemrograman ASP.NET

Active Server Pages .NET (sering disingkat ASP.NET) adalah kumpulan teknologi dalam *Framework .NET* untuk membangun aplikasi web dinamik dan *XML Web Service* (Layanan Web XML). Halaman ASP.NET dijalankan di *server* kemudian akan dibuat halaman markup (penanda) seperti HTML (*Hypertext Markup Language*), WML (*Wireless Markup Language*), atau XML (*Extensible Markup Language*) yang dikirim ke *browser desktop* atau *mobile*.

ASP.NET merupakan teknologi *Microsoft* yang dikhususkan untuk pengembangan aplikasi berbasis web dinamis berbasis platform (Kurniawan 2010:

1).*NET Framework*. ASP.NET didesain untuk memberikan kemudahan pada pengembang web untuk membuat aplikasi berbasis web dengan cepat, mudah, dan efisien karena meminimalkan penulisan kode program dengan bantuan komponen-komponen yang tersedia, sehingga dapat meningkatkan produktivitas.

2.3.4 Database SQL Server

Microsoft *SQL Server* adalah sebuah sistem manajemen basis data relasional (RDBMS) produk Microsoft. Bahasa kueri utamanya adalah *Transact – SQL* yang merupakan implementasi dari SQL standar ANSI/ISO yang digunakan oleh *Microsoft* dan *Sybase*. Umumnya *SQL server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar.



Gambar 2.6 Logo SQL Server
Sumber: Data Penelitian (2017)

2.4. Penelitian Terdahulu

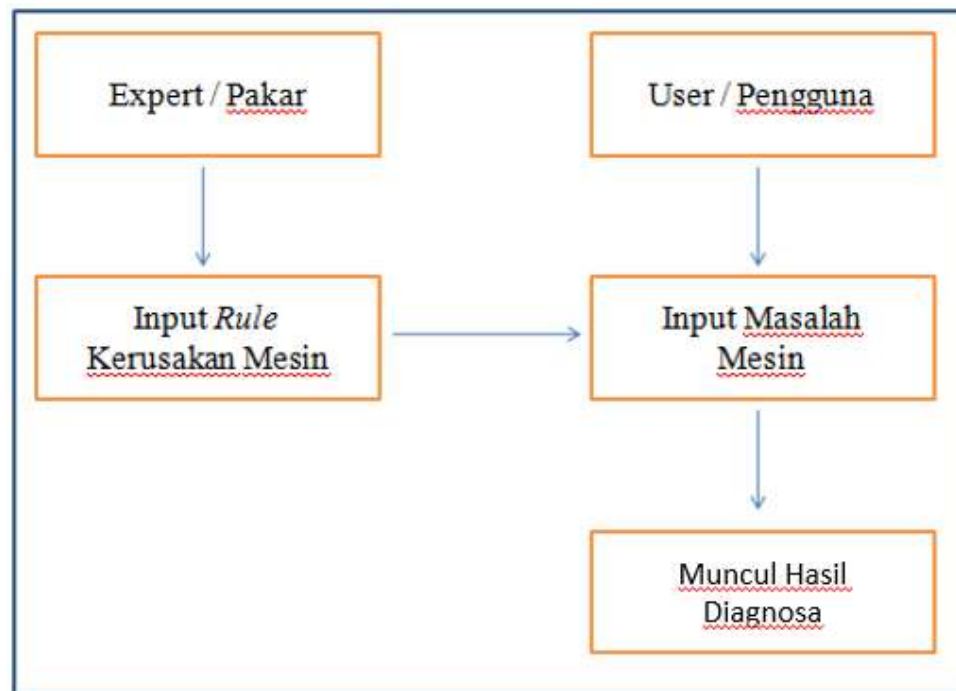
Penulis	Judul	Metode	Hasil
Nency Extise Putri (2016)	Sistem Pakar Kerusakan <i>Hardware</i> Komputer Dengan Metode <i>Forward Chaining</i>	<i>Forward Chaining</i>	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Purwanto dan Putra (2016)	Sistem Pakar Diagnosa Kerusakan Pada Televisi LED Dengan Menggunakan Metode <i>Forward Chaining</i>	<i>Forward Chaining</i>	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Harison dan Alexyusandera (2014)	Sistem Pakar Perawatan dan Perbaikan Ringan Mobil Bensin Menggunakan Video Tutorial Berbasis Web	<i>Forward Chaining</i>	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Rosmawati Tamin (2015)	Sistem Pakar untuk Diagnosa Kerusakan Pada Printer Menggunakan Metode <i>Forward Chaining</i>	<i>Forward Chaining</i>	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.
Guntur dan Nita Merlina (2016)	Sistem Pakar Diagnosa Kerusakan Pada Mesin Pendingin Ruangan dengan Metode <i>Forward Chaining</i>	<i>Forward Chaining</i>	Kesimpulan dari Hasil Penelitian adalah metode tersebut layak digunakan.

2.5. Kerangka Pemikiran

Identifikasi masalah dalam penelitian ini adalah banyaknya pengguna mesin manufaktur ini yang tidak dapat memperbaiki sendiri dengan alasan kekurangan ilmu pengetahuan dan belum adanya sebuah sistem yang mampu membantu pengguna untuk memperbaiki kerusakan pada mesin manufaktur tersebut, lambatnya kehadiran teknisi dapat mengakibatkan perusahaan mengalami kerugian yang besar karena mesin manufaktur tidak dapat bekerja selagi rusak.

Kerangka berpikir ini disusun dengan berdasarkan pada tinjauan pustaka dan hasil penelitian yang relevan atau terkait.

Adapun kerangka pemikiran sebagai berikut



Gambar 2.7 Kerangka Pemikiran Penelitian
Sumber: Data Olahan Peneliti (2016)

Adapun uraian dari setiap point diatas adalah sebagai berikut:

2.5.1 *Expert / Pakar*

Merupakan seorang pakar yang memiliki akses seluruh halaman yang terdapat di dalam sistem. Termasuk halaman untuk mengisi *rule*, mau pun penambahan *user* / pengguna.

2.5.2 *User / Pengguna*

Merupakan pengguna sistem, biasanya hanya melakukan login kedalam sistem untuk melaporkan kerusakan yang ada di sistem. Tidak dapat mengisi *rule* ataupun menambah *user* lain.

2.5.3 *Input Rule / Mesin*

Penambahan *Rule* atau Mesin kedalam sistem, sehingga para User dapat melaporkan mesin yang terjadi kerusakan.

2.5.4 *Input Masalah Mesin*

Apabila pengguna mesin mengalami masalah saat menggunakan mesin, maka dapat ke langkah ini untuk memberitahukan kepada mesin bahwa terjadi kerusakan pada mesin yang mereka gunakan, serta gejala-gejala yang terjadi.

2.5.5 Muncul Hasil Diagnosa

Dari gejala-gejala yang telah di input oleh pengguna, maka sistem akan menampilkan hasil diagnosa kerusakan mesin nya.

BAB III

METODE PENELITIAN

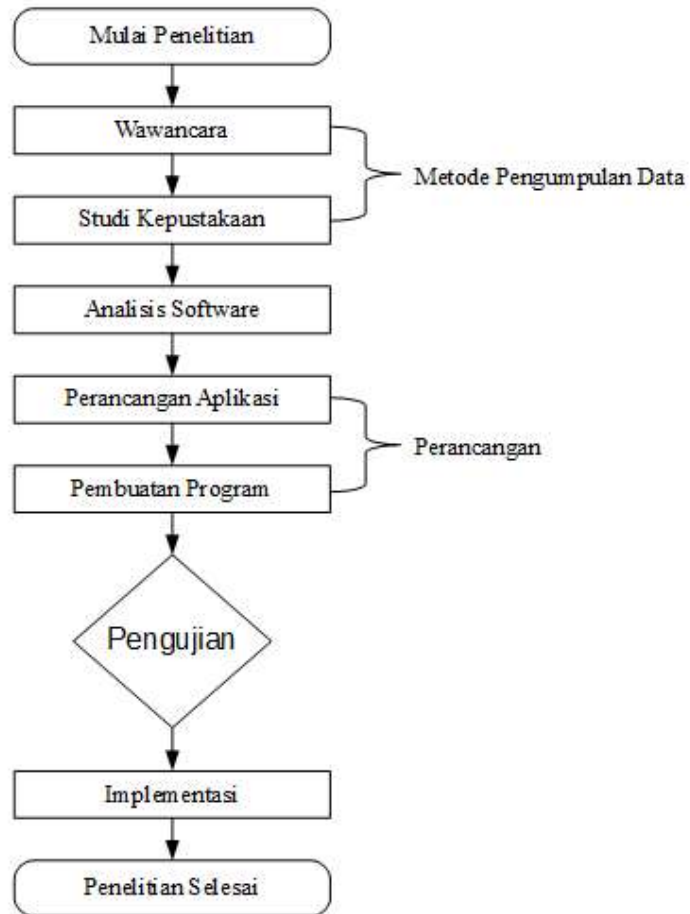
3.1. Desain Penelitian

Desain penelitian ini bertujuan untuk melaksanakan penelitian sehingga diperoleh suatu logika dalam pengujian untuk memperoleh hasil penelitian serta kesimpulan yang 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, 2013:131).

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

Berikut adalah alur desain penelitian:



Gambar 3.1 Desain Penelitian
Sumber: Data penelitian (2017)

Desain penelitian pada Aplikasi Sistem Pakar Mendiagnosa Kerusakan Mesin Manufaktur Berbasis Web dapat dilihat pada gambar 3.1, adapun penjelasan yang terdapat pada desain penelitian diatas sebagai berikut:

a. Mulai Penelitian

Langkah pertama untuk memulai penelitian adalah menentukan topik serta menentukan masalah yang menjadi penelitian. Penulis melakukan penelitian tentang “Diagnosa Kerusakan Pada Mesin Manufaktur”.

b. Wawancara

Setelah menentukan topik dan masalah yang menjadi penelitian, maka perlu dikumpulkan data-data yang diperlukan oleh penelitian, yaitu melalui wawancara dengan pakar dalam bidang mesin manufaktur.

c. Studi Kepustakaan

Selain dengan wawancara, penulis juga mencari data melalui buku-buku yang ada di internet.

d. Analisis *Software*

Setelah mendapatkan data-data yang diperlukan dalam penelitian, maka harus ditentukan *software* apa saja yang dibutuhkan dalam penelitian yang akan dilakukan tersebut. Disini penulis memerlukan *software SQLServer* untuk *database*, *Microsoft Visual Studio* untuk bagian *coding*, serta *Browser Google Chrome* untuk menjalankan aplikasi.

e. Perancangan Aplikasi

Setelah menentukan *software* yang dibutuhkan, maka langkah selanjutnya adalah merancang aplikasi yang ingin dibuat.

f. Pembuatan Program

Setelah aplikasi yang diinginkan dirancang, maka yang diperlukan cuma membuat *coding* dari aplikasi tersebut sehingga bisa mendapatkan hasil yang

diinginkan. Adapun bahasa pemrograman yang digunakan penuli dalam pembuatan aplikasi ini adalah *ASP.NET*, *C#*, *Javascript*, *Jquery*, *Bootstrap*, *CSS*, serta *software* pendukung lainnya.

g. Pengujian

Langkah berikutnya setelah program selesai dibuat adalah untuk melakukan uji coba dengan menggunakan mesin sebenarnya, untuk menentukan apakah aplikasi hasil penelitian benar bekerja sesuai tujuan. Pada Tugas Akhir/Skripsi ini pendekatan yang dilakukan untuk melakukan pengujian adalah pendekatan *Black-Box Testing* (pengujian kotak hitam), yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian ini dimaksud untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

h. Implementasi

Apabila setelah dilakukan pengujian terbukti aplikasi hasil penelitian bekerja dengan benar sesuai tujuan, maka langkah selanjutnya adalah mengimplementasikan aplikasi tersebut dilapangan.

i. Selesai

Setelah selesai, yang harus dilakukan adalah merawat aplikasi tersebut dengan benar sehingga tidak salah melakukan diagnosa

3.2. Operasional Variabel

Operasional Variabel adalah segala sesuatu yang berbentuk apa saja yang ditetapkan peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut, kemudian ditarik kesimpulannya (Sugiyono, 2010: 58).

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). Untuk memudahkan pemahaman mengenai metode ini, berikut adalah operasional variabel menggunakan metode inferensi *Forward Chaining*.

Tabel 3.1 Indikator Kerusakan Mesin Manufaktur (FC)

Variabel	Indikator
Mesin Manufaktur	1. Masalah pada pompa air pendingin
	2. Masalah pada minyak pelumas
	3. Masalah pada sparepart internal mesin

Sumber: Data penelitian (2017)

Tabel 3.2 Kode Indikator Kerusakan Mesin Manufaktur (FC)

Kode	Indikator	Gejala
FK01	Masalah pada pompa air pendingin	Temperatur mesin tinggi, Pompa air pendingin tidak berputar, Pompa air pendingin bocor
FK02	Masalah pada minyak pelumas	Pipa pengeluaran pelumas salah, tekanan minyak pelumas terlalu tinggi, jumlah minyak pelumas mesin dibawah garis minimal, minyak pelumas bocor
FK03	Masalah pada sparepart mesin	Mesin bergetar tak beraturan, mesin mengeluarkan suara abnormal.

Sumber: Data penelitian (2017)

Tabel 3.3 Kode Gejala Kerusakan Mesin Manufaktur

Kode Gejala	Nama Gejala
FP01	Temperatur mesin tinggi
FP02	Pompa air pendingin tidak berputar
FP03	Pompa air pendingin bocor
FP04	Pipa pengeluaran pelumas salah
FP05	Tekanan minyak pelumas terlalu tinggi
FP06	Jumlah minyak pelumas mesin dibawah garis minimal
FP07	Minyak pelumas bocor
FP08	Mesin bergetar tak beraturan
FP09	Mesin mengeluarkan suara abnormal.

Sumber: Data penelitian (2017)

Tabel 3.4 Tabel Keputusan Kerusakan Mesin Manufaktur

Kode Kerusakan	Kode Gejala								
	FP01	FP02	FP03	FP04	FP05	FP06	FP07	FP08	FP09
FK01	√	√	√						
FK02				√	√	√	√		
FK03								√	√

Sumber: Data Penelitian (2017)

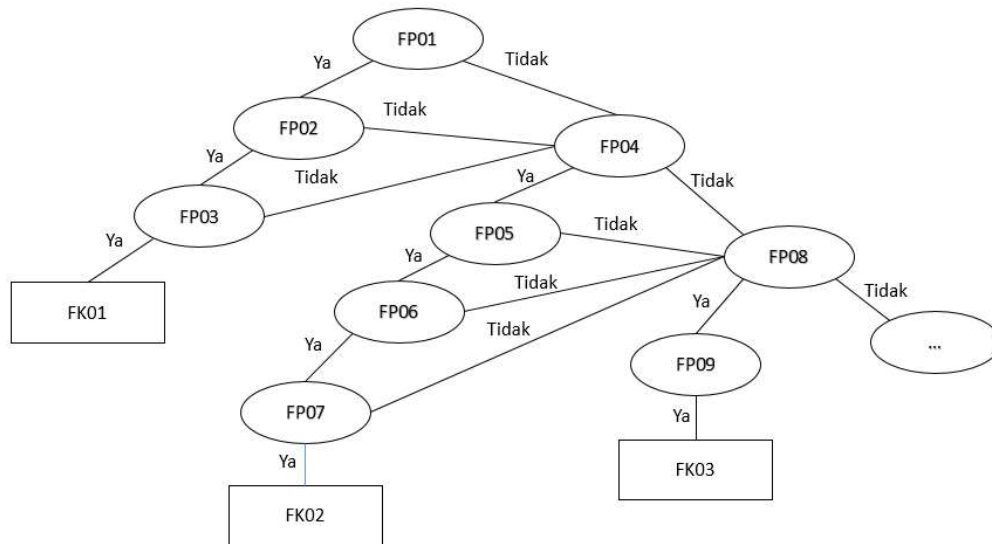
Kaidah berdasarkan dari table diatas:

1. IF FK01 THEN FP01 AND FP02 AND FP03
2. IF FK02 THEN FP04 AND FP05 AND FP06 AND FP07
3. IF FK03 THEN FP08 AND FP09

Penjelasan dari kaidah diatas adalah:

1. Jika temperatur mesin manufakur tinggi dan pompa air pendingin bocor dan pompa air pendingin tidak berputar, maka Masalah pada pompa air pendingin
2. Jika pipa pengeluaran pelumas salah dan tekanan minyak pelumas terlalu tinggi dan jumlah minyak pelumas mesin dibawah garis min dan minyak pelumas bocor, maka masalah pada minyak pelumas.
3. Jika mesin mengeluarkan suara abnormal dan bergetar tidak beraturan, maka masalah pada sparepart mesin.

Berdasarkan tabel keputusan diatas maka pohon keputusannya adalah sebagai berikut



Gambar 3.2 Pohon Keputusan
Sumber: Data Penelitian (2017)

3.3. Teknik Pengumpulan Data

Pengumpulan data dapat dilakukan dalam berbagai setting, berbagai sumber, dan berbagai cara. Bila dilihat dari setting-nya, data dapat dikumpulkan pada setting alamiah (natural settings), pada laboratorium dengan metode eksperimen, dirumah dengan berbagai responden, pada suatu seminar, diskusi di jalan dan lain-lain. (Sugiyono, 2012: 137)

Wawancara digunakan sebagai teknik pengumpulan data apabila peneliti ingin melakukan studi pendahuluan untuk menemukan permasalahan yang harus diteliti, dan juga apabila peneliti ingin mengetahui hal-hal dari responden yang lebih mendalam dan jumlah respondennya sedikit/kecil. (Sugiyono, 2012:137)

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 pemrograman.

Cara pengumpulan data dalam penelitian ini yaitu:

3.3.1. Metode Studi Pustaka

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

3.3.2. Metode 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.

Wawancara yang dilakukan peneliti yaitu menggunakan metode wawancara semi terstruktur, maksudnya adalah topic dan tema sudah ditentukan sebelumnya yaitu berdasarkan indikator yang ingin diteliti namun rumusan pertanyaan tidak baku, disesuaikan dengan situasi dan keadaan yang dihadapi.

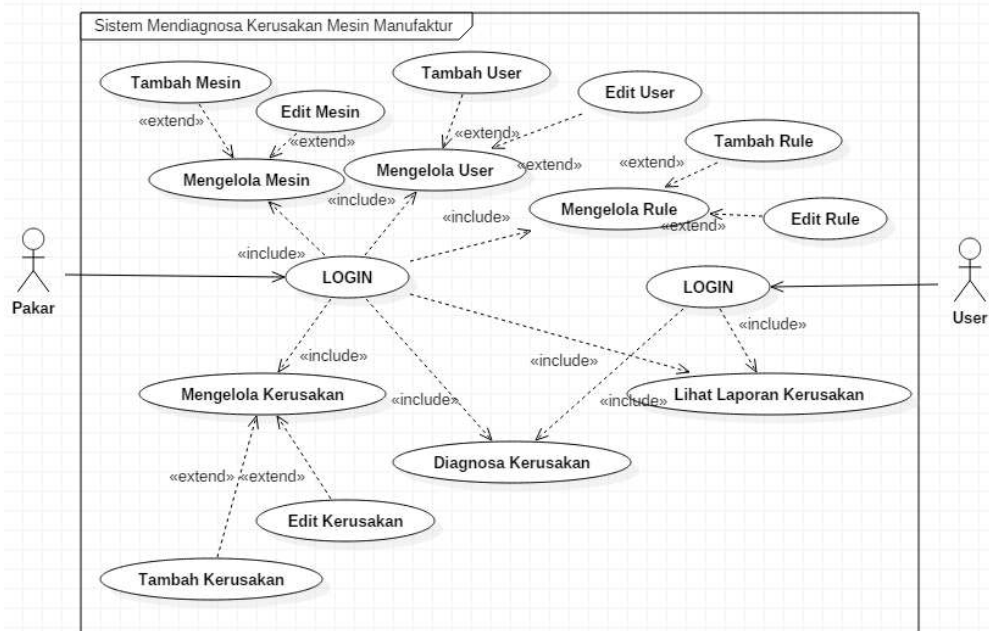
3.4. Perancangan Sistem

3.4.1. UML (*Unified Modeling Language*)

Pada tahap perancangan UML menggunakan beberapa diagram antara lain:

1. *Use Case Diagram*

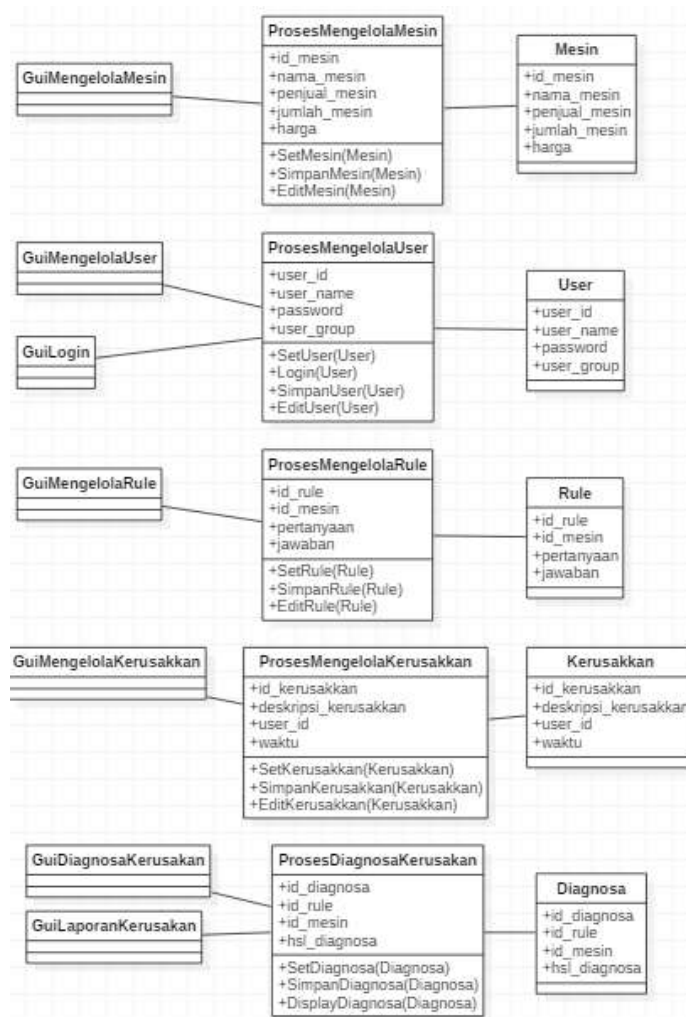
Use Case Diagram digunakan untuk memahami sistem dan mengevaluasi bahwa yang dilakukan sistem adalah untuk membantu memecahkan masalah kerusakan mesin manufaktur yang dialami pengguna. *Use case diagram* sistem pakar kerusakan mesin manufaktur dapat dilihat digambr 3.2.



Gambar 3.3 *Use Case Diagram*
Sumber: Data Penelitian (2017)

2. Class Diagram

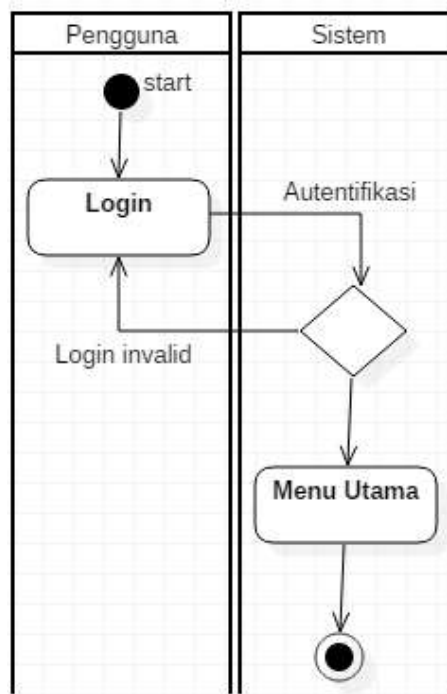
Class Diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana class tersebut saling berkolaborasi untuk mencapai suatu tujuan. *Class Diagram* sistem Pakar diagnosa kerusakan mesin manufaktur seperti pada gambar 3.3



Gambar 3.4 *Class Diagram*
Sumber: Data Penelitian (2017)

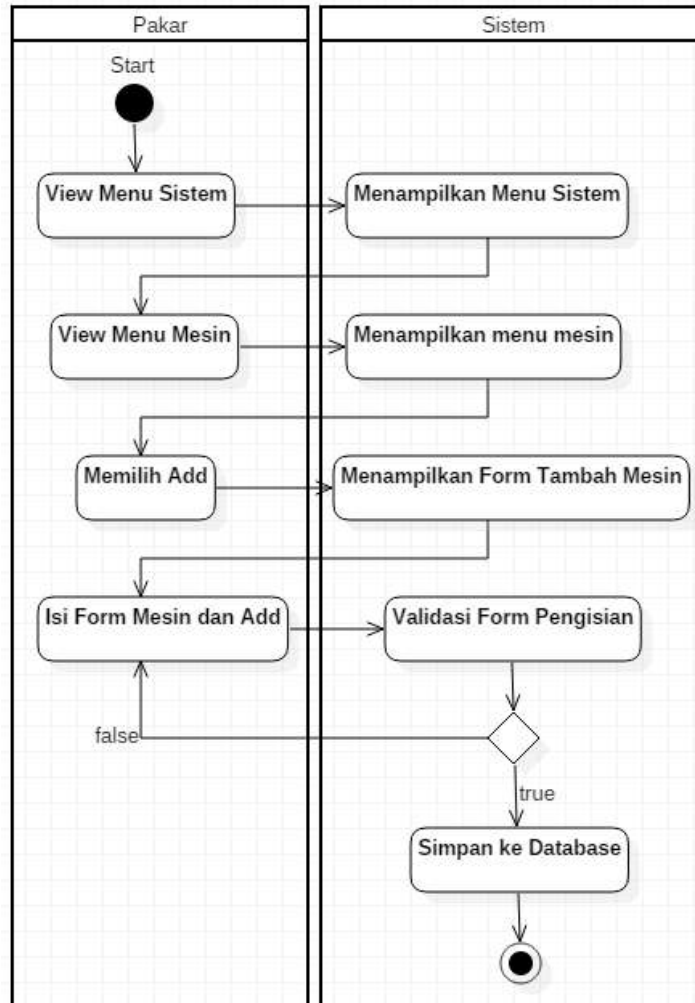
3. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan aliran kejadian dalam *use case* sistem dengan tujuan untuk memudahkan menkomunikasikan langkah-langkah ke aliran kejadian. *Activity Diagram* pada sistem pakar kerusakan mesin manufaktur dapat dilihat dibawah ini.

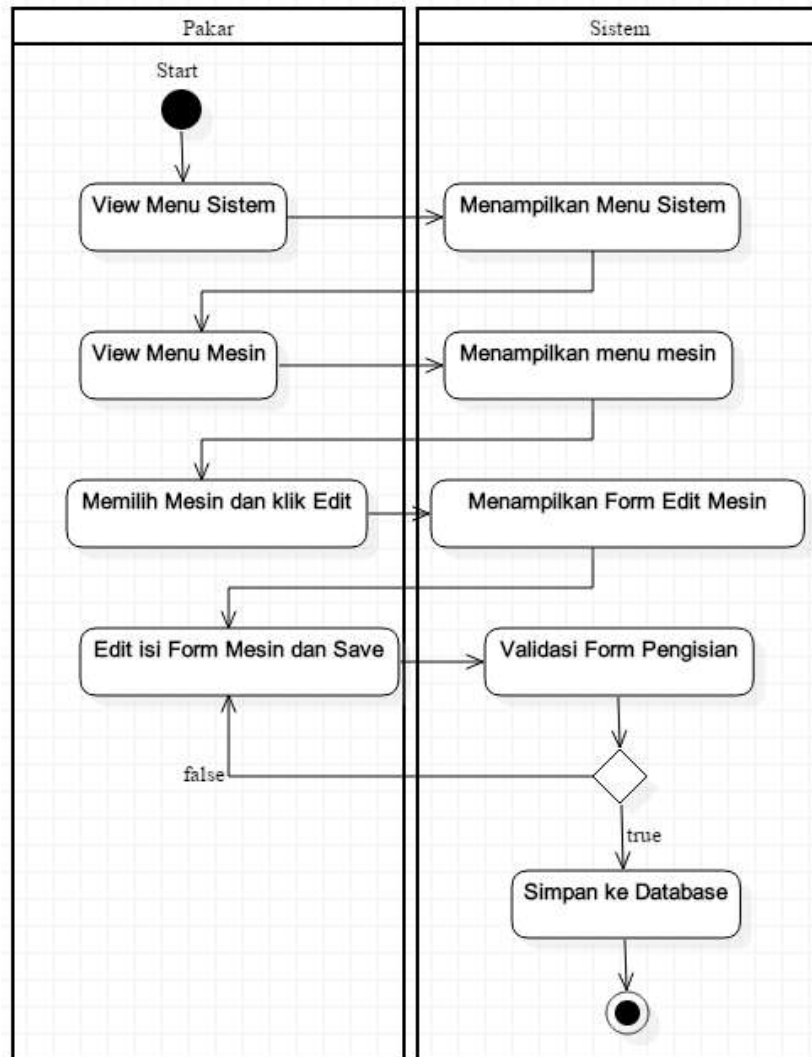


Gambar 3.5 *Activity Diagram* Login

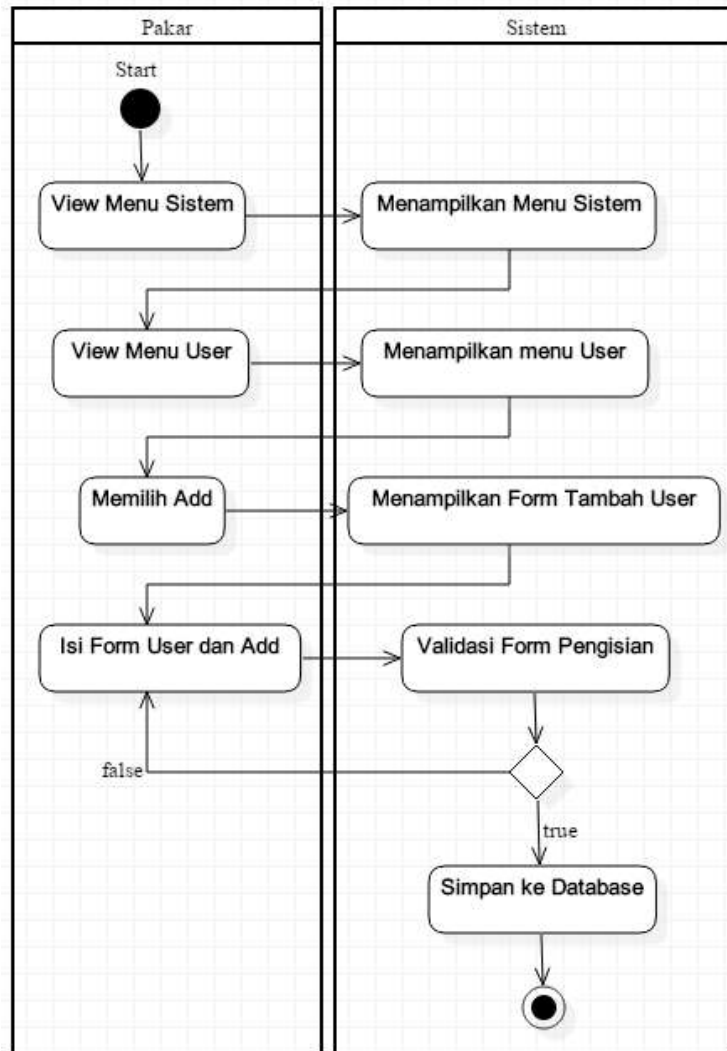
Sumber: Data Penelitian (2017)



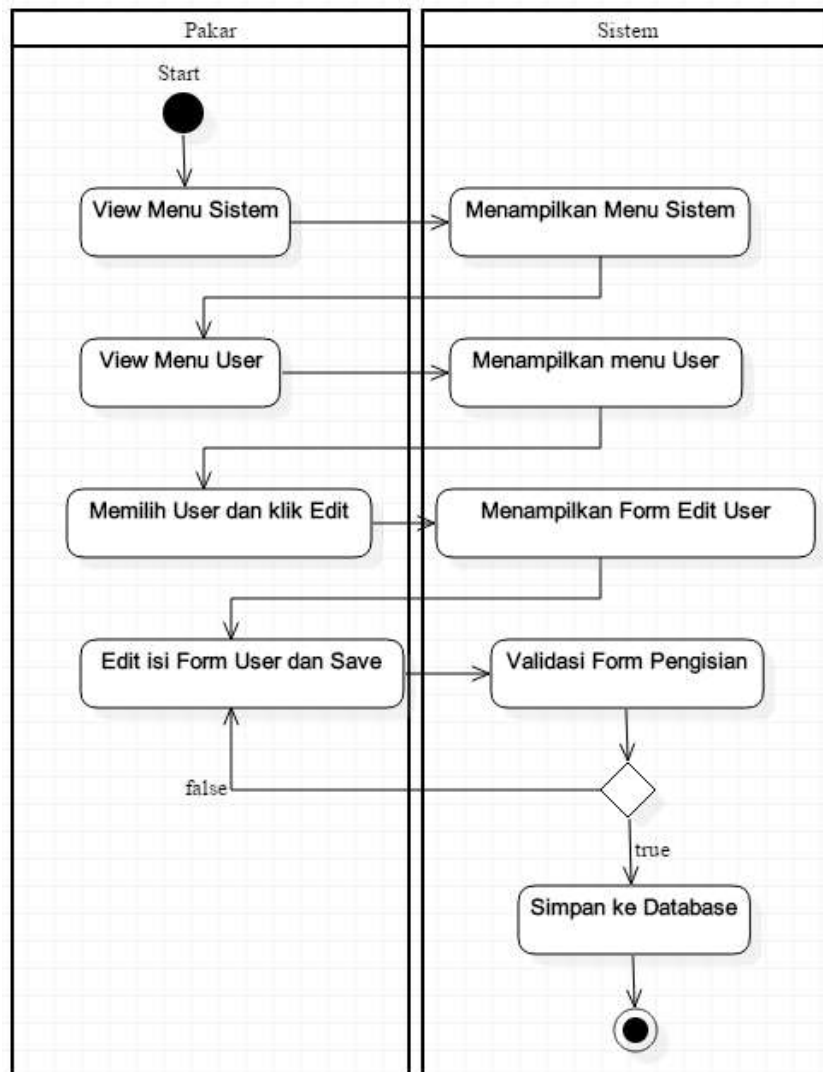
Gambar: 3.6 Activity Diagram Tambah Mesin
 Sumber: Data Penelitian (2017)



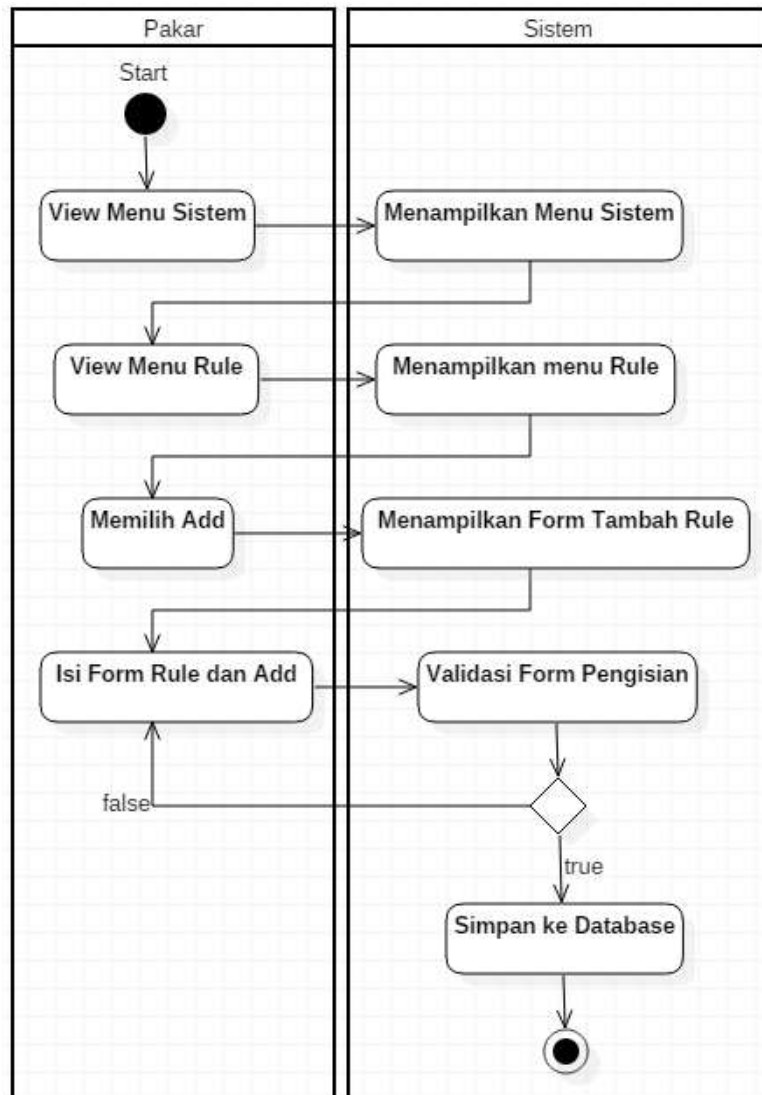
Gambar: 3.7 Activity Diagram Ubah Mesin
 Sumber: Data Penelitian (2017)



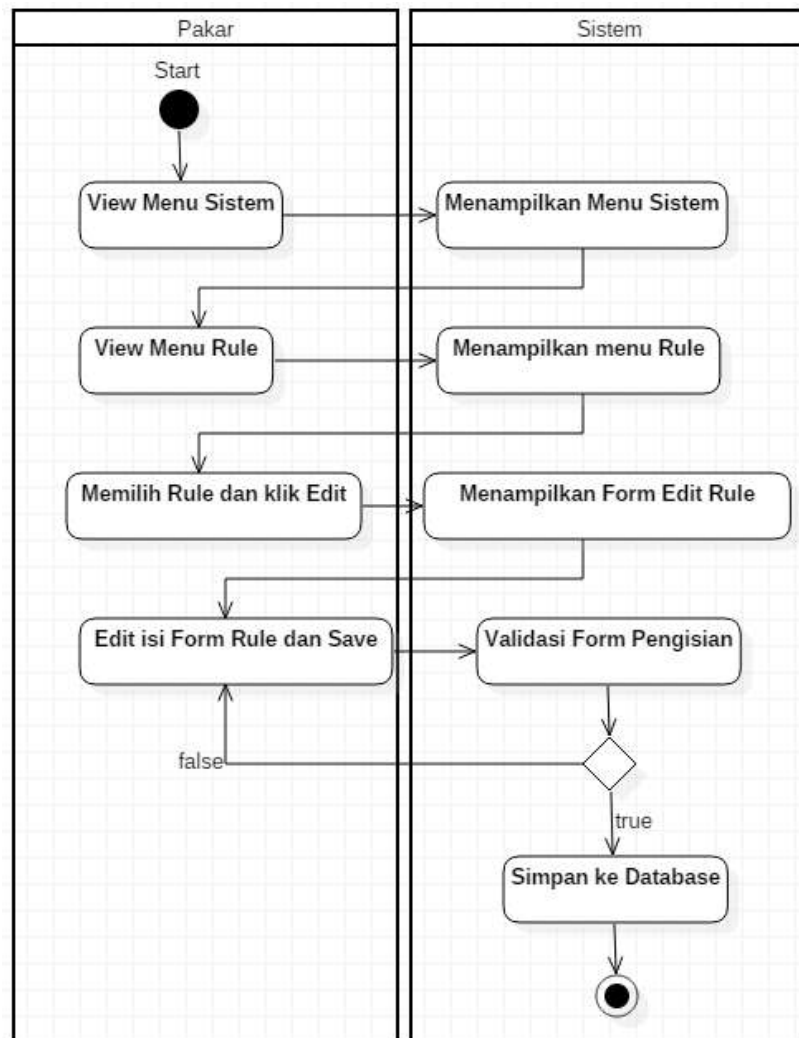
Gambar 3.8 *Activiy Diagram Tambah User*
Sumber: Data Penelitian (2017)



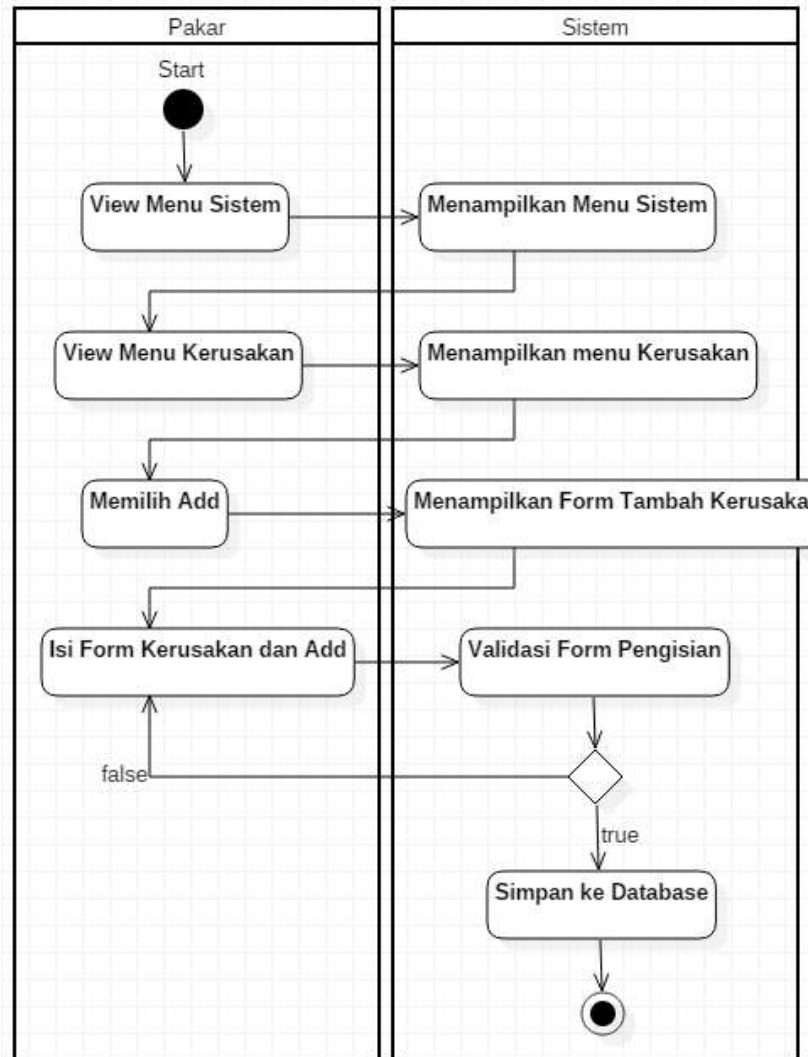
Gambar 3.9 *Activiy Diagram Ubah User*
Sumber: Data Penelitian (2017)



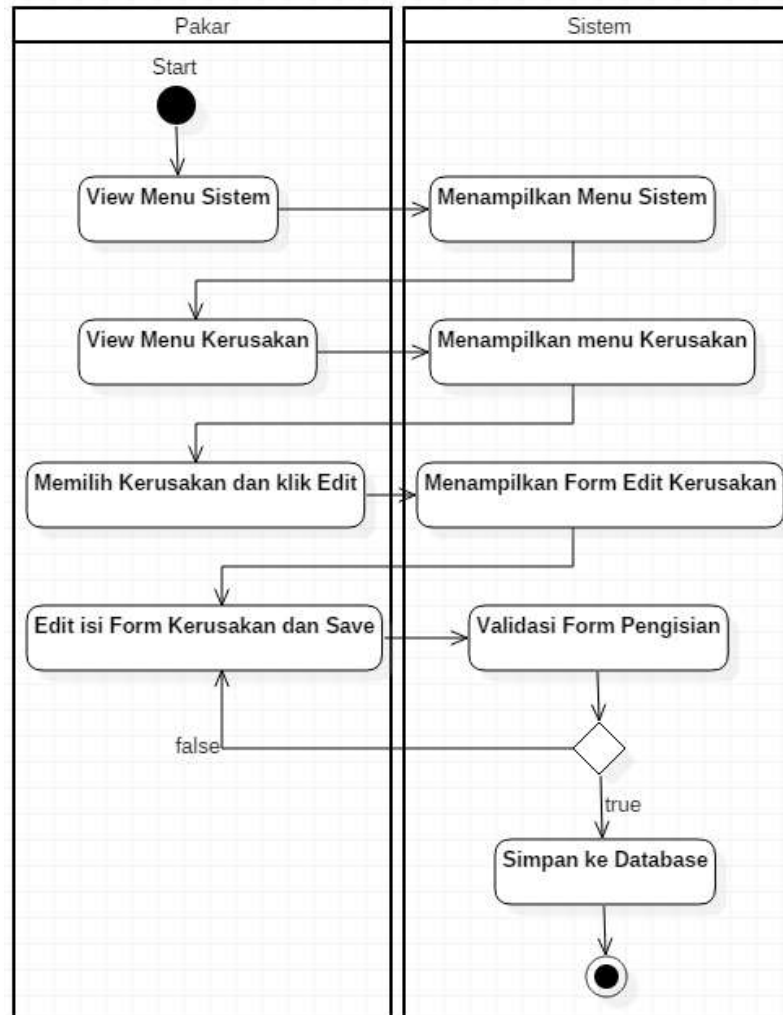
Gambar 3.10 *Activiy Diagram Tambah Rule*
Sumber: Data Penelitian (2017)



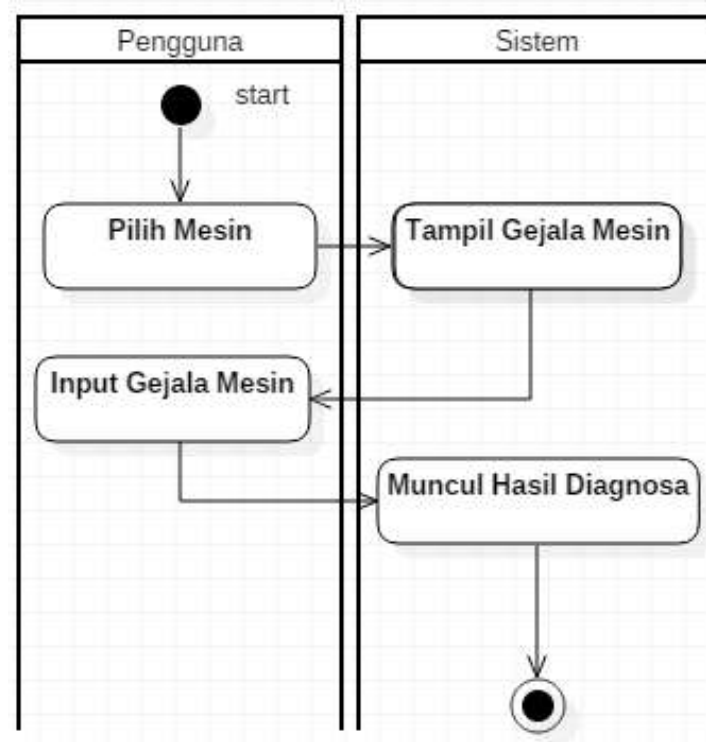
Gambar 3.11 Activity Diagram Ubah Rule
Sumber: Data Penelitian (2017)



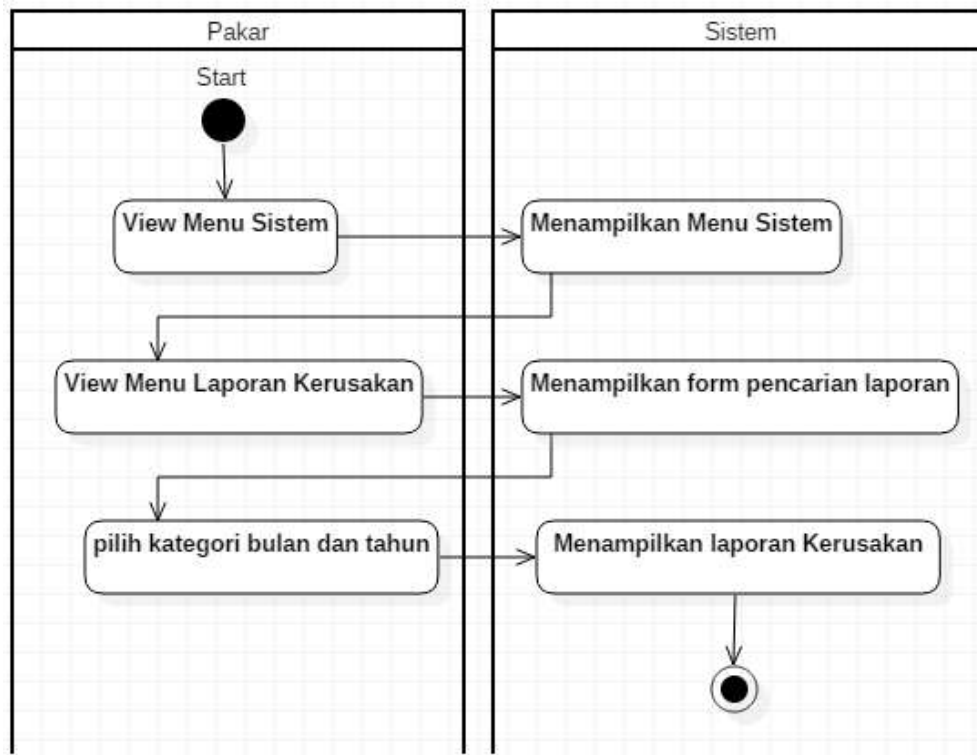
Gambar 3.12 Activity Diagram Tambah Kerusakan
Sumber: Data Penelitian (2017)



Gambar 3.13 Activity Diagram Ubah Kerusakan
Sumber: Data Penelitian (2017)



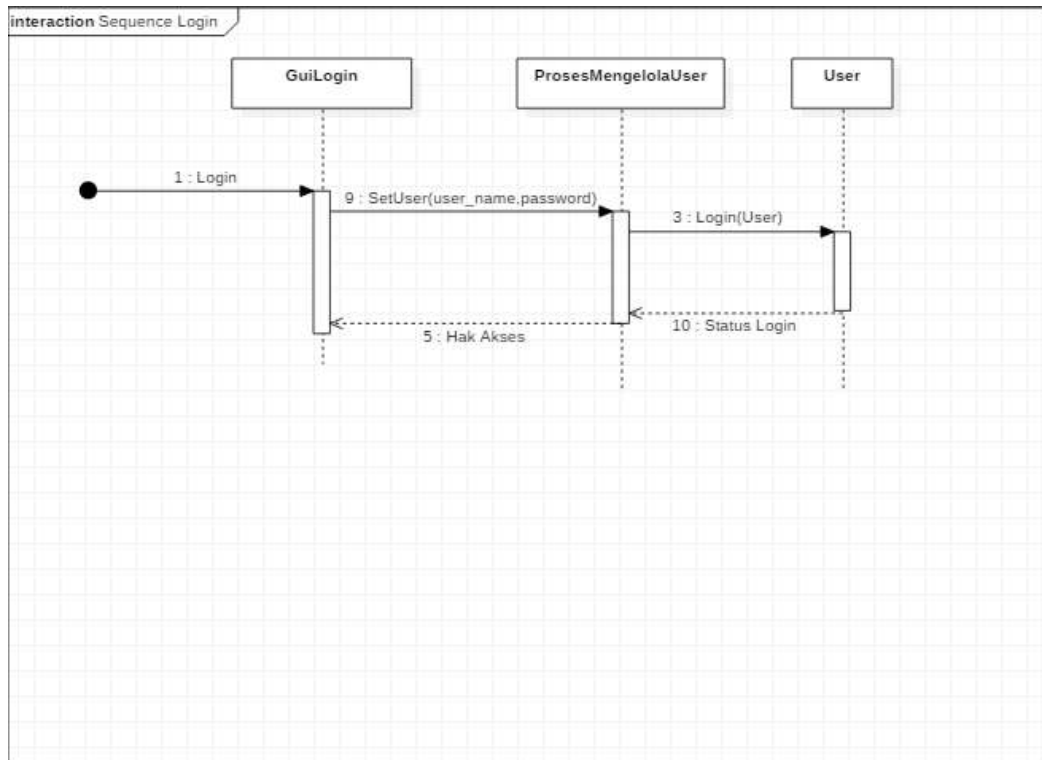
Gambar 3.14 *Activity Diagram* Diagnosa
Sumber: Data Penelitian (2017)



Gambar 3.15 *Activity Diagram* Lihat Laporan Kerusakan
Sumber: Data Penelitian (2017)

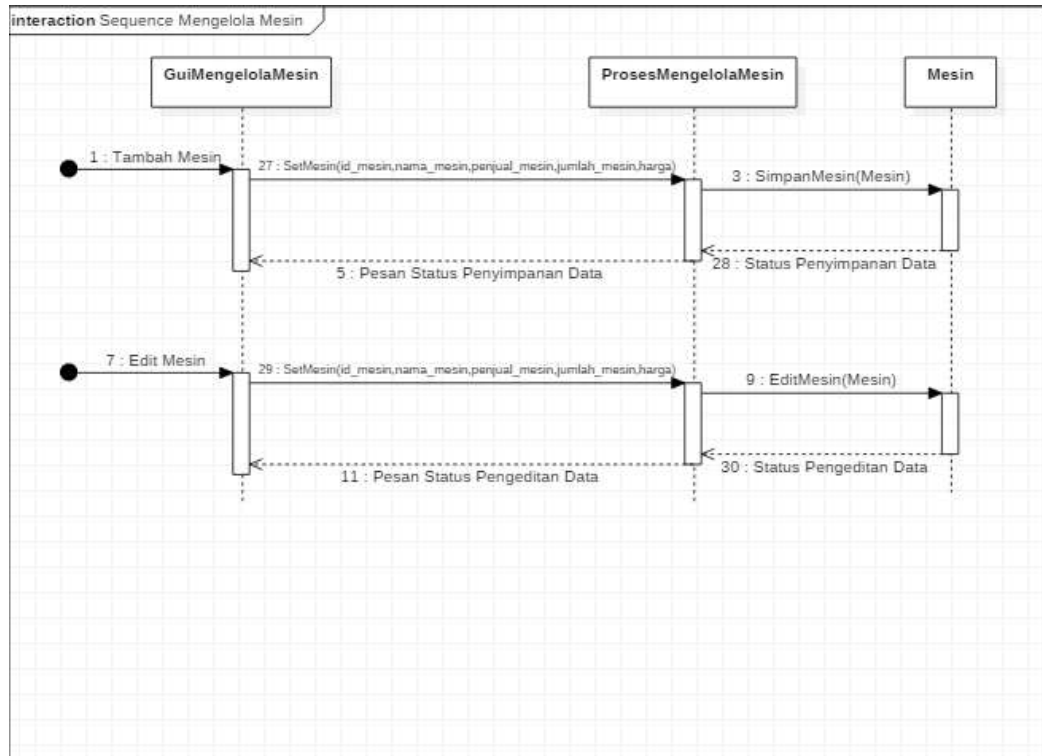
4. Sequence Diagram

a. Sequence Diagram Login



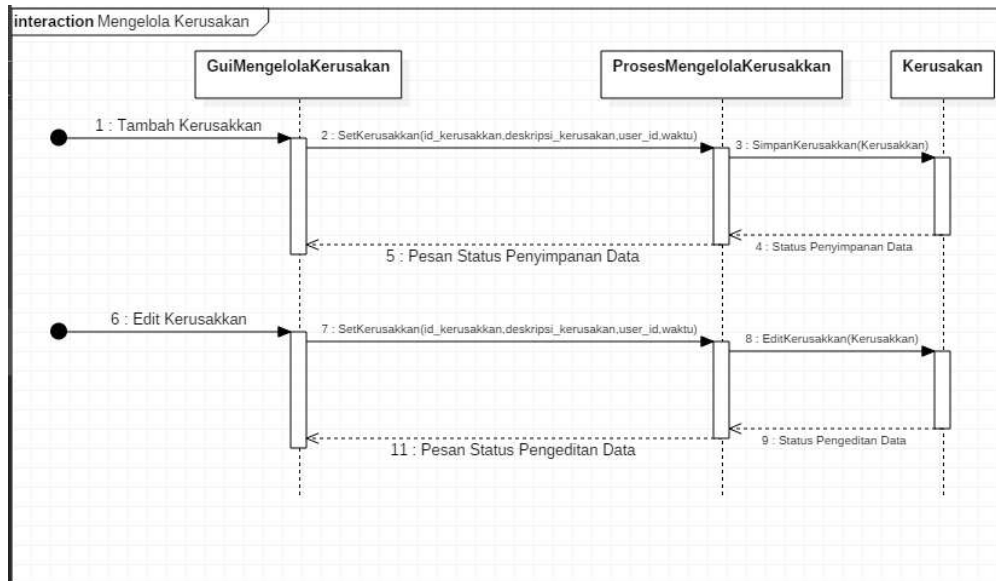
Gambar 3.16 *Sequence Diagram* Login
Sumber: Data Penelitian (2017)

b. *Sequence Diagram Mesin*



Gambar 3.17 *Sequence Diagram Mesin*
Sumber: Data Penelitian (2017)

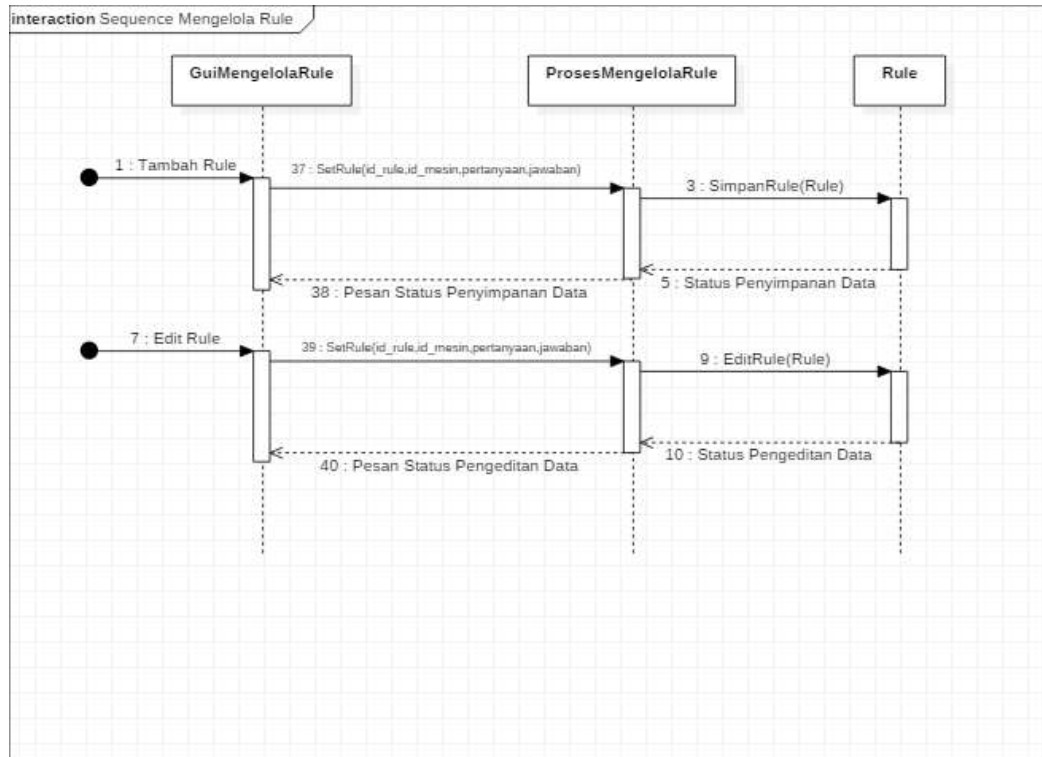
c. *Sequence Diagram Kerusakan*



Gambar 3.18 *Sequence Diagram Kerusakan*

Sumber: Data Penelitian (2017)

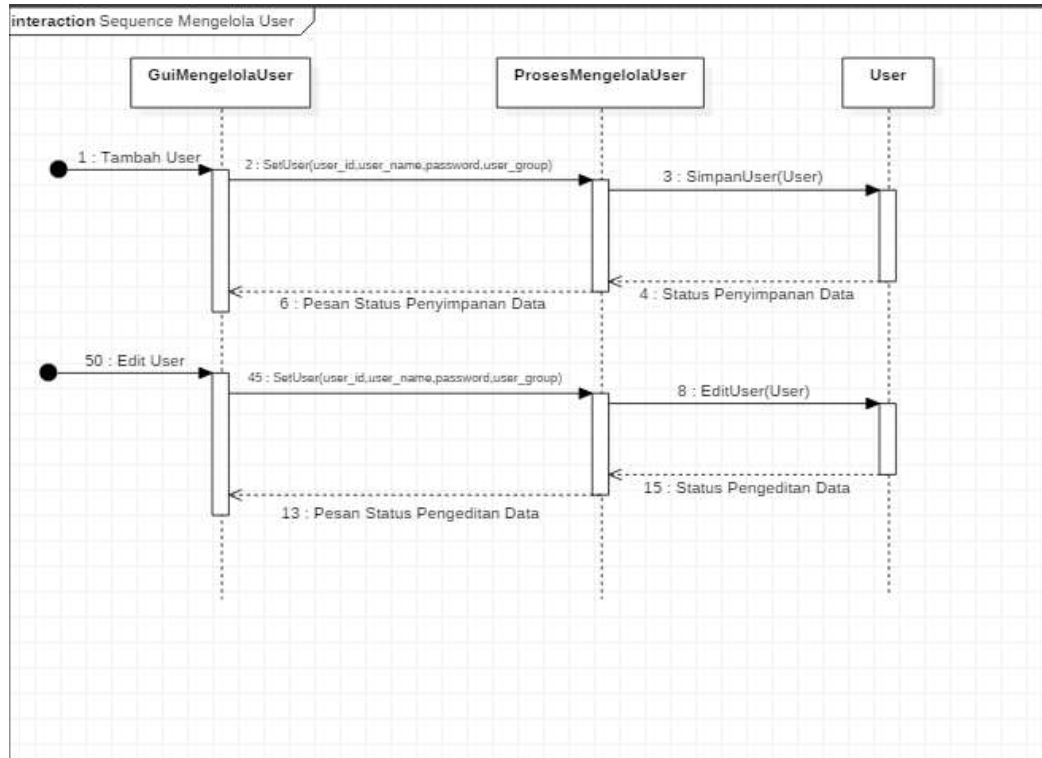
d. *Sequence Diagram Rule*



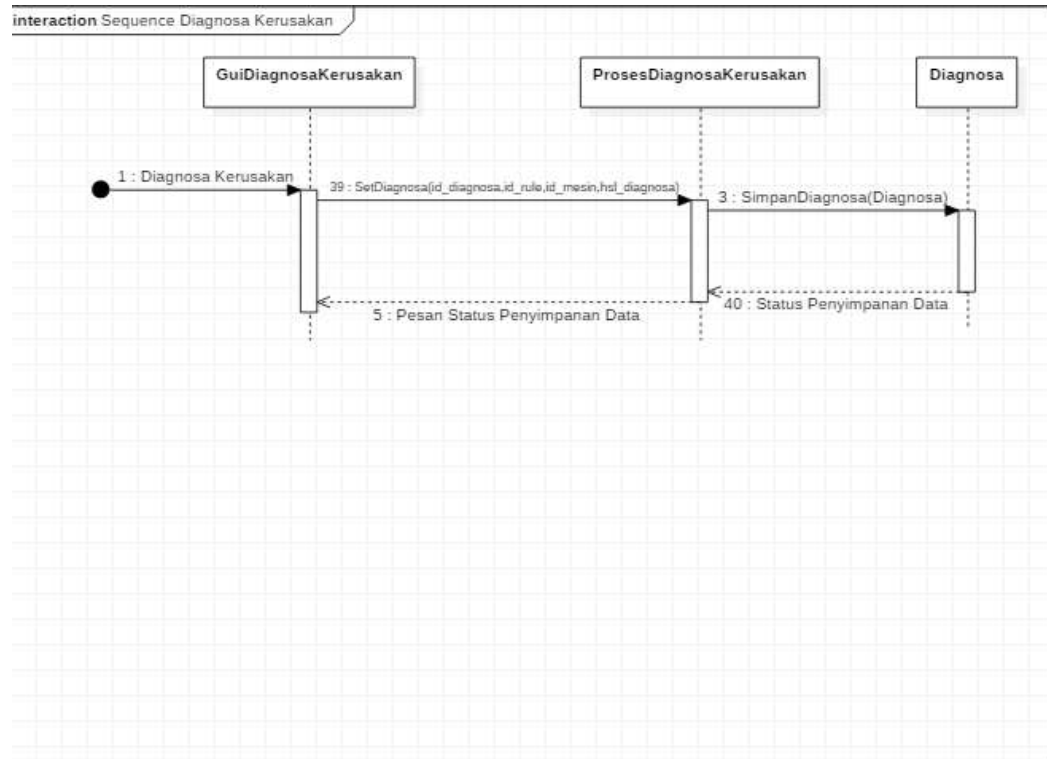
Gambar 3.19 *Sequence Diagram Rule*

Sumber: Data Penelitian (2017)

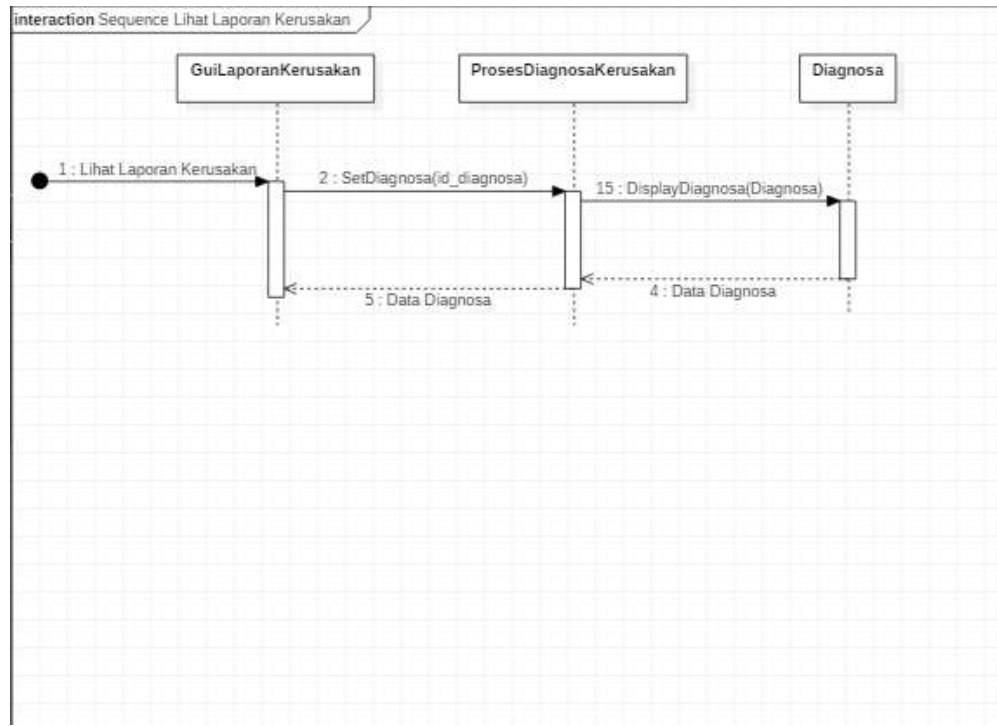
e. *Sequence Diagram User*



Gambar 3.20 *Sequence Diagram User*
Sumber: Data Penelitian (2017)

f. *Sequence Diagram* Diagnosa Kerusakan

Gambar 3.21 *Sequence Diagram* Diagnosa Kerusakan
Sumber: Data Penelitian (2017)

g. *Sequence Diagram* Lihat Laporan Kerusakkan

Gambar 3.22 *Sequence Diagram* Lihat Laporan Kerusakkan
Sumber: Data Penelitian (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. Desain Tabel *User*

Tabel 3.5 Desain Tabel *User*

DESAIN TABEL USER	
Nama Kolom	Tipe Data
UserID	varchar(20)
Username	varchar(50)
Password	nvarchar(20)
GroupID	Varchar(20)
UpdatedBy	Nvarchar(20)
UpdatedDate	datetime()

Sumber: Data Penelitian (2017)

2. Desain Tabel Mesin

Tabel 3.6 Desain Tabel Mesin

DESAIN TABEL MESIN	
Nama Kolom	Tipe Data
MacID	nvarchar(20)
MacName	nvarchar(50)
VendorID	nvarchar(20)
PurchaseAmount	Int
Price	Bigint
UpdatedBy	nvarchar(20)
UpdatedOn	datetime()

Sumber: Data Penelitian (2017)

3. Desain Tabel Kerusakan

Tabel 3.7 Desain Tabel Kerusakan

DESAIN TABEL KERUSAKAN	
Nama Kolom	Tipe Data
ProblemID	Varchar(100)
ProblemDesc	Varchar(100)
UpdatedBy	Varchar(20)
UpdatedDate	Datetime()

Sumber: Data Penelitian (2017)

4. Desain Tabel Pertanyaan

Tabel 3.8 Desain Tabel Pertanyaan

DESAIN TABEL PERTANYAAN	
Nama Kolom	Tipe Data
MacID	Varchar(20)
ProblemID	Varchar(100)
KodePertanyaan	Int
Pertanyaan	Varchar(100)
UpdatedBy	Varchar(20)
UpdatedDate	Datetime

Sumber: Data Penelitian (2017)

5. Desain Tabel Diagnosa

Tabel 3.9 Desain Tabel Diagnosa

DESAIN TABEL DIAGNOSA	
Nama Kolom	Tipe Data
DiagnosaID	Varchar(20)
MacID	Varchar(100)
KodePertanyaan	Int
Pertanyaan	Varchar(100)
Pilihan	Varchar(5)
Result	Varchar(20)
CheckedBy	Varchar(20)
CheckedDate	datetime()

Sumber: Data Penelitian (2017)

3.5. Waktu dan Tempat Penelitian

Penelitian akan dilakukan pada PT. Multi Karya Bajatama dengan jadwal seperti pada tabel dibawah ini.

Jadwal Kegiatan	Oktober				November				Desember				Januari				Februari			
	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
Identifikasi Masalah				■	■	■														
Pengumpulan Data				■	■	■														
Analisa Data				■	■	■														
Perancangan Sistem							■	■	■	■										
Pengkodean									■	■	■	■	■	■	■					
Uji Coba																■	■	■		
Penulisan Skripsi				■	■	■	■	■	■	■	■	■	■	■	■	■	■			