

**SISTEM PAKAR MENDETEKSI KERUSAKAN MESIN
KASIR DI HYPERMART TANJUNG UNCANG
DENGAN METODE *CERTAINTY FACTOR*
BERBASIS *WEB***

SKRIPSI



**Oleh:
Chrimes Pandapotan Manurung
130210292**

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

**SISTEM PAKAR MENDETEKSI KERUSAKAN MESIN
KASIR DI HYPERMART TANJUNG UNCANG
DENGAN METODE *CERTAINTY FACTOR*
BERBASIS *WEB***

SKRIPSI

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



Oleh:

**Chrimes Pandapotan Manurung
130210292**

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

PERNYATAAN

Dengan ini saya 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 dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi.

Batam, 7 Februari 2017
Yang membuat pernyataan,

Chrimes Pandapotan Manurung
130210292

**SISTEM PAKAR MENDETEKSI KERUSAKAN MESIN
KASIR DI HYPERMART TANJUNG UNCANG
DENGAN METODE *CERTAINTY FACTOR*
BERBASIS *WEB***

Oleh
Chrimes Pandapotan Manurung
130210292

SKRIPSI
Untuk memenuhi salah satu syarat guna
memperoleh gelar Sarjana

Telah disetujui oleh Pembimbing pada tanggal
seperti tertera di bawah ini

Batam, 7 Februari 2017

Pastima Simanjuntak S.Kom, M.SI.
Pembimbing

ABSTRAK

Seiring berkembangnya minat belanja masyarakat saat ini, maka peran perusahaan *retail* harus lebih aktif dalam memperbaiki atau menangani masalah yang ada dalam perusahaan tersebut, salah satunya mesin kasir. Mesin kasir di Hypermart Tanjung Uncang dapat mengalami kerusakan yang disebabkan oleh berbagai hal dalam berbagai waktu, akan tetapi untuk mengetahui penyebab kerusakan dan apa penyebab kerusakan mesin kasir memerlukan seorang *staff* khusus (*IT*) untuk melakukan perbaikan terhadap mesin kasir tersebut, sedangkan *staff IT* terbatas dan tidak dapat mengatasi permasalahan di perusahaan yang memiliki lebih dari satu perusahaan di kota Batam secara bersamaan, sehingga diperlukan suatu sistem yang mempunyai kemampuan seperti seorang *staff IT* agar sistem dapat membantu mengetahui apa penyebab kerusakan dan kerusakan di satu mesin kasir tersebut, sistem ini berisi basis pengetahuan keahlian seorang *staff IT*. Penelitian ini dirancang menggunakan metode *certainty factor* dengan menggunakan bahasa pemograman *PHP* dan *database MySql*. Sistem pakar mendeteksi kerusakan mesin kasir berbasis *web* yang telah dikembangkan mempunyai keunggulan dalam kemudahan pemakaian. Dengan sistem yang berbasis *web* yang dimiliki, sistem pakar untuk mendeteksi kerusakan mesin kasir dapat di akses oleh Hypermart yang lain yang berada di kota Batam atau bahkan di seluruh Indonesia. Sehingga sistem ini dapat mengatasi persoalan keterbatasan *staff IT* untuk mendeteksi kerusakan mesin kasir.

Kata kunci: Sistem pakar, deteksi kerusakan, mesin kasir, *certainty factor*, *web*

ABSTRACT

As the current interest in public spending increases, the role of retail companies should be more active in improving or handling the problems that exist within the company, one of the cash register machines. The cash register at Hypermart Tanjung Uncang can suffer damage caused by various things at various times, but to know the cause of damage and what causes damage the cash register requires a special staff (IT) to make improvements to the cash register, while the IT staff is limited and can not overcome the problems in companies that have more than one company in the city of Batam simultaneously, so that required a system that has the ability as an IT staff for the system can help find out what causes damage and damage at one such checkout machine, this system contains the knowledge base expertise of an IT staff. This research is designed using certainty factor method by using PHP programming language and MySQL database. Expert systems detect web-based cashier damage that has been developed has advantages in ease of use. With a web-based system that is owned, expert system to detect the damage to the cash register can be accessed by other Hypermart located in the city of Batam or even throughout Indonesia. So this system can overcome the problem of IT staff limitations to detect the damage to the cash register machine

Keywords: *Expert system, damage detection, cash register, certainty factor, web*

KATA PENGANTAR

Segala puji dan 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, Bapak Andi Maslan, S.T, M.SI.
3. Ibu Pastima Simanjuntak, S.Kom., M.SI, selaku pembimbing Skripsi pada Program Studi Teknik Informatika Universitas Putera Batam.
4. Dosen dan Staff Universitas Putera Batam.
5. Bapak Priatama sebagai Depertemen Manager Supporing Hypermart Tanjung Uncang yang telah memberikan kepada saya dukungannya dan semangat.

6. Keluarga yang selalu memberikan doa dan motivasi yang baik
7. Rekan-rekan mahasiswa/i Universitas Putera Batam yang turut memberikan doa dan dukungannya.
8. Mitra kerja yang selalu memberikan masukan yang berguna untuk penelitian ini.
9. Serta pihak-pihak lain yang tidak dapat disebutkan satu per satu.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan selalu mencurahkan kasih karunianya, Amin.

Batam, Februari 2017

Chrismes Pandapotan Manurung

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN	xi
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	4
1.3 Batasan Masalah	4
1.4 Rumusan Masalah	5
1.5 Tujuan	5
1.6 Manfaat	6
1.6.1 Aspek Teoritis	6
1.6.2 Aspek Praktis	6
BAB II	7
KAJIAN PUSTAKA	7
2.1 Teori Dasar	7
2.1.1 Kecerdasan Buatan atau <i>Artificial Intelligence (AI)</i>	7
2.1.1.1 <i>Fuzzy Logic</i>	9
2.1.1.2 Jaringan Saraf Tiruan	13
2.1.1.3 Sistem Pakar	17
2.1.2 <i>Web</i>	26
2.1.3 <i>Database (Basis Data)</i>	27
2.1.4 Validasi Sistem	27
2.2 Variabel Penelitian	28
2.3 <i>Software</i> Pendukung	30
2.3.1 <i>Xampp</i>	32
2.3.2 <i>phpMyadmin</i>	33
2.3.3 <i>Hypertext Preprocessor (PHP)</i>	34
2.4 Penelitian Terdahulu	54
2.5 Kerangka Pemikiran	56
BAB III	58
METODE PENELITIAN	58
3.1 Desain Penelitian	58
3.2 Teknik Pengumpulan Data	61
3.3 Operasional Variabel	62
3.4 Metode Analisis Data	63
3.4.1 Desain Basis Pengetahuan	64

3.4.2 Struktur Kontrol (Mesin inferensi)	67
3.4.3 DesainUML (<i>Unified Modeling Language</i>)	69
3.4.4 <i>Prototype</i>	79
3.5 Lokasi dan Jadwal Penelitian	83
3.5.1 Lokasi	83
3.5.2 Jadwal Penelitian.....	83
BAB IV	1
HASIL PENELITIAN DAN PEMBAHASAN	1
4.1 Hasil Penelitian	1
4.2 Pembahasan Sistem	95
4.2.1 Pengujian Validasi Sistem	95
4.2.2 Pengujian dengan Pakar.....	101
5.1 Kesimpulan	103
5.2 Saran.....	103
LAMPIRAN I FORM WAWANCARA	107
LAMPIRAN II FOTO WAWANCARA.....	109
LAMPIRAN III SURAT PENELITIAN.....	110
LAMPIRAN IV SURAT BALASAN PENELITIAN.....	111

DAFTAR PUSTAKA
RIWAYAT HIDUP
SURAT KETERANGAN PENELITIAN
LAMPIRAN

DAFTAR TABEL

	Halaman
Tabel 2.1 Aturan kombinasi MCYIN	24
Tabel 2.2 Bobot Certainty Factor	24
Tabel 2.3 Simbol <i>Use Case Diagram</i>	48
Tabel 2.4 Simbol <i>Activity Diagram</i>	50
Tabel 2.5 Simbol <i>Sequence Diagram</i>	51
Tabel 2.6 Lanjutan	52
Tabel 2.7 Tabel <i>class Diagram</i>	53
Tabel 3.1 Tabel data kerusakan	65
Tabel 3.2 Tabel penyebab kerusakan dan solusi mesin kasir	65
Tabel 3.3 Tabel penyebab kerusakan mesin kasir	66
Tabel 3.4 Jadwal Penelitian.....	84
Tabel 4.1 Pengujian Menu Halaman Utama	95
Tabel 4.2 Pengujian Data Kerusakan.....	96
Tabel 4.3 Pengujian Tambah Data Kerusakan	96
Tabel 4.4 Pengujian <i>Edit</i> Data Kerusakan	96
Tabel 4.5 Pengujian Hapus Data Kerusakan	97
Tabel 4.6 Pengujian Data Penyebab Kerusakan.....	97
Tabel 4.7 Pengujian Tambah Data Penyebab Kerusakan	97
Tabel 4.8 Pengujian <i>Edit</i> Data Penyebab Kerusakan	98
Tabel 4.9 Pengujian Hapus Data Penyebab Kerusakan	98
Tabel 4.10 Pengujian Menu Basis Pengetahuan.....	98
Tabel 4.11 Pengujian Tambah Data Basis Pengetahuan.....	99
Tabel 4.12 Pengujian <i>Edit</i> Data Basis Pengetahuan	99
Tabel 4.13 Pengujian Hapus Data Basis Pengetahuan.....	99
Tabel 4.14 Pengujian Menu Ubah <i>Password</i>	100
Tabel 4.15 Pengujian Menu <i>Log Out</i>	100
Tabel 4.16 Pengujian Halaman <i>Login</i>	100
Tabel 4.17 Pengujian Deteksi Kerusakan	101
Tabel 4.18 Tabel Hasil Perbandingan Antara Sistem Dengan Pakar	102

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur Sistem Pakar	26
Gambar 2.2 Logo <i>Xampp</i>	32
Gambar 2.3 Logo <i>phpMyAdmin</i>	33
Gambar 2.4 Logo <i>PHP</i>	34
Gambar 2.5 Logo HTML (<i>Hyper text markup language</i>).....	37
Gambar 2.6 Logo CSS (<i>Cascading Style sheet</i>).....	40
Gambar 2.7 Logo <i>JavaScript</i>	42
Gambar 2.8 Logo <i>jQuery</i>	43
Gambar 2.9 Logo <i>MySQL</i>	44
Gambar 2.10 Logo <i>Sublime text</i>	45
Gambar 2.11 Logo <i>StarUML</i>	46
Gambar 2.12 Kerangka Pemikiran	56
Gambar 3.1 Desain Penelitian	59
Gambar 3.2 <i>Flowchart</i>	68
Gambar 3.3 Use Case Diagram	70
Gambar 3.4 <i>Diagram activity user login</i>	71
Gambar 3.5 <i>Diagram activity data kerusakan</i>	71
Gambar 3.6 <i>Diagram activity data penyebab kerusakan</i>	72
Gambar 3.7 <i>Diagram activity data basis pengetahuan</i>	72
Gambar 3.8 <i>Diagram activity lupa password</i>	73
Gambar 3.9 <i>Diagram activity logout</i>	73
Gambar 3.10 <i>Diagram activity deteksi kerusakan mesin kasir</i>	74
Gambar 3.11 <i>Sequence diagram login</i>	75
Gambar 3.12 <i>Sequence data kerusakan kasir</i>	75
Gambar 3.14 <i>Sequence diagram data basis pengetahuan</i>	76
Gambar 3.15 <i>Sequence diagram ubah password</i>	77
Gambar 3.16 <i>Sequence diagram logout</i>	77
Gambar 3.17 <i>Sequence diagram deteksi kerusakan</i>	78
Gambar 3.18 <i>class diagram</i>	78
Gambar 3.19 Tampilan halaman <i>web</i>	79
Gambar 3.20 Tampilan halaman utama	80
Gambar 3.21 Tampilan halaman kerusakan	80
Gambar 3.22 Tampilan halaman penyebab kerusakan	81

Gambar 3.23 Tampilan halaman basis pengetahuan	81
Gambar 3.24 Tampilan halaman ubah <i>password</i>	82
Gambar 3.25 Tampilan halaman deteksi kerusakan	82
Gambar 4.1 <i>Login</i>	86
Gambar 4.2 Menu Utama	87
Gambar 4.3 Data Kerusakan.....	88
Gambar 4.4 <i>Edit</i> Data Kerusakan	88
Gambar 4.5 Tambah Data Kerusakan	89
Gambar 4.6 Basis Pengetahuan	90
Gambar 4.7 Tambah Basis Pengetahuan.....	91
Gambar 4.8 Ubah Basis Pengetahuan	91
Gambar 4.9 Ubah <i>Password</i>	92
Gambar 4.10 Deteksi Kerusakan	93
Gambar 4.11 Hasil Deteksi	94

DAFTAR RUMUS

	Halaman
$CF(H,E) = MB(H,E) - MD(H,E)$ Rumus 2.1 Hasil Akhir <i>Certainty Factor</i>	22
$CF(H,e) = CF(E,e) * CF(H,E)$ Rumus 2.2 Dasar Rumus <i>Certainty Factor</i>	23
$CF(H,e) = CF(H,E)$ Rumus 2.3 Nilai Kepastian.....	23
$CF[H,E]=MB[H,E] - MD[H,E]$ Rumus 3.1 Hasil <i>Certainty Factor</i>	63
$CF[H,E]1= CF[H] * CF[E]$ Rumus 3.2 Dasar Rumus <i>Certainty Factor</i>	63
$CF_{combine}CF[H,E]1,2 = CF[H,E]1 + CF[H,E]2 * [1 - CF[H,E]1]$ Rumus 3.3 <i>Combine Certainty Factor</i>	64
$CF_{combine}CF[H,E]old,3 = CF[H,E]old + CF[H,E]3 * [1CF[H,E]old]$ Rumus 3.4 <i>Combine Certainty Factor Old</i>	64

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN I FORM WAWANCARA.....	107
LAMPIRAN II FOTO WAWANCARA	109
LAMPIRAN III SURAT PENELITIAN	110
LAMPIRAN IV SURAT BALASAN PENELITIAN	111

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kota Batam adalah kota yang terletak di Provinsi Kepulauan Riau, Indonesia. Wilayah kota Batam terletak di pulau Batam dan seluruh wilayah nya di kelilingi selat Singapura dan selat Malaka. Batam adalah kota yang terbesar di Kepulauan Riau dan kota terbesar keempat di wilayah Sumatera setelah Medan, Palembang dan Pekanbaru.

Selain dikelilingi selat Singapura dan Selat Malaka, kota Batam juga dijadikan sebagai tempat untuk melakukan investasi dengan cara mendirikan perusahaan asing yang bergerak di bidang otomotif, elektronik, pipa, gas dan salah satu nya yang cukup berkembang saat ini adalah perusahaan *retail*. Perusahaan *retail* yaitu perusahaan yang bergerak dibidang pemasaran produk meliputi semua aktifitas yang melibatkan penjualan secara langsung ke *consumen* akhir untuk penggunaan pribadi bukan organisasi

Pada saat ini perkembangan teknologi sudah sangat pesat, memudahkan untuk mendapatkan informasi dari mana saja, kapan saja dan siapa saja dengan mudah dan cepat. Teknologi membawa dampak positif pada berbagai bidang seperti bidang telekomunikasi, pendidikan, industri dan bidang *retail* salah satu nya

Sudah banyak saat ini teknologi yang digunakan dalam kehidupan sehari-hari. Mulai dalam memesan makanan, memesan kendaraan saat ingin bepergian dan lain sebagainya. Sehingga muncul ide para pakar untuk untuk menciptakan sistem yang berfungsi untuk membantu mereka dalam melakukan tugasnya. Selain dari membantu mereka dalam menyimpan ilmu yang mereka miliki, sistem ini juga berfungsi untuk mempercepat kinerja mereka kedepannya.

Secara umum sistem pakar (*Expert system*) adalah sistem yang mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah atau pekerjaan yang biasa dilakukan oleh pakar. Sedangkan menurut (Dewi, Soebroto, & Furqon, 2015) sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang dipikirkan oleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam. Sebagai contoh, IT adalah seorang pakar yang mampu mendiagnosis penyebab kerusakan suatu barang tertentu serta dapat memberikan penatalaksanaan terhadap kerusakan tersebut. Sistem pakar yang baik dirancang agar dapat menyelesaikan suatu permasalahan tertentu dengan meniru kerja para ahli. Dengan sistem pakar ini orang awam pun dapat menyelesaikan masalah yang cukup rumit yang sebenarnya hanya dapat diselesaikan dengan bantuan para ahli. Bagi para ahli, sistem pakar ini juga akan membantu aktifitas nya sebagai asisten yang sangat berpengalaman.

Dalam kehidupan kita sehari-hari kita tidak lepas dari kata belanja. Apakah itu belanja kebutuhan sehari-hari ataupun belanja pernak pernik yang akan kita pakai.

Mengingat kemajuan zaman sekarang orang akan lebih senang belanja di tempat yang menurut mereka nyaman dan kualitas yang baik dan sekaligus dapat mendapatkan hiburan di samping jadwal kerja mereka yang sangat sibuk.

Dengan adanya tempat berbelanja yang baik mendukung semua fasilitas mereka dalam melancarkan operasional tempat tersebut, salah satu nya harus menyediakan mesin kasir untuk mempercepat melakukan transaksi, sehingga proses pembayaran akan lebih cepat untuk memuaskan para *consumen* yang berbelanja.

Mesin kasir juga tidak lepas dari kata rusak ataupun *blank* dan lain sebagai nya, adapun kerusakan kerusakan yang mesin kasir alami antara lain

Mesin kasir sering megalami *off line* terjadi karna beberapa hal yaitu karena *server down*, port pada POS rusak. Mesin kasir tidak mengeluarkan struk karena habisnya kertas dan sensor *head printer*. Mesin kasir tidak dapat *scanning* barang(*barcode*) karena *scanner* belum di *update*. Mesin kasir mengalami blue screen saat sedang dipakai diakibatkan karena memori yang didalam POS rusak.

Ini merupakan kerusakan-kerusakan yang di alami mesin kasir saat sedang beroperasi (melakukan transaksi). Sehingga hal ini yang menjadi pertimbangan untuk merancang aplikasi **“SISTEM PAKAR MENDETEKSI KERUSAKAN MESIN KASIR DI HYPERMART TANJUNG UNCANG DENGAN METODE CERTAINTY FACTOR BERBASIS WEB”**

1.2 Identifikasi Masalah

Dari penjelasan di atas, masalah yang teliti adalah sebagai berikut:

1. Mesin kasir *Off Line*
2. Mesin kasir tidak mengeluarkan struk
3. Mesin kasir tidak dapat scanning barang (*Barcode*)
4. Mesin kasir muncul *bluescreen* pada layar

1.3 Batasan Masalah

Untuk menghindari kesalah pahaman dan memperluas yang akan diteliti, maka batasan masalah yang diangkat adalah sebagai berikut

1. Pembuatan aplikasi ini menggunakan sistem pakar metode *Certainty Faktor* (Faktor Kepastian).
2. Bahasa pemograman yang digunakan adalah bahasa pemograman *PHP* dengan sistem berbasis *WEB*.
3. Database yang digunakan untuk membantu dan mendukung implementasi pada aplikasi tersebut menggunakan database *MYSQL*
4. Kerusakan yang diteliti yaitu, mesin kasir *off line*, mesin kasir tidak mengeluarkan struk, mesin kasir tidak dapat scanning barang(*barcode*) dan mesin kasir mengalami *bluescreen* di Hypermart Tanjung Uncang.

1.4 Rumusan Masalah

Penulis merumuskan masalah dalam melakukan penelitian untuk pembuatan aplikasi berbasis web untuk mendeteksi kerusakan mesin kasir adalah :

1. Bagaimana cara mendeteksi kerusakan mesin kasir dengan menggunakan metode *certainty factor*?
2. Bagaimana merancang sistem pakar mendeteksi kerusakan mesin kasir pada Hypermart Tanjung Uncang ?
3. Bagaimana implemetasi sistem pakar ini berbasis *Web*?

1.5 Tujuan

Tujuan dari penelitian ini selain sebagai pemenuhan tugas skripsi, penelitian ini juga bertujuan:

1. Berdasarkan indikasi-indikasi yang dihadapi maka kita dapat mengetahui kerusakan apa yang terjadi pada mesin kasir tersebut.
2. Untuk mengetahui rancangan sistem pakar untuk mendeteksi kerusakan mesin kasir pada Hypermart Tanjung Uncang.
3. Untuk mengetahui implementasi sistem pakar untuk mendeteksi kerusakan mesin kasir pada Hypermart Tanjung Uncang.

1.6 Manfaat

1.6.1 Aspek Teoritis

Hasil Penelitian ini diharapkan dapat bermanfaat untuk pengembangan ilmu *technology*, ilmu pendidikan dan program studi lain nya untuk memberikan referensi dalam pengkajian masalah-masalah ilmu teknologi yang berhubungan dengan semua bidang ilmu pada saat sekarang ini.

1. Mengembangkan ilmu pengetahuan tentang sistem pakar menggunakan metode *Certainty Factor* (factor kepastian).
2. Penelitian ini diharapkan dapat menjadi referensi dan masukan untuk penelitian selanjutnya.

1.6.2 Aspek Praktis

Secara praktis, manfaat yang dapat diambil dari penelitian ini bagi berbagai pihak antara lain :

1. Hasil penelitian ini diharapkan dapat menambah pengetahuan bagi karyawan khususnya karyawan Hypermart Tanjung Uncang dalam memperbaiki mesin kasir yang rusak.
2. Membantu *IT* yang bekerja di Hypermart Tanjung Uncang dalam melakukan *training* bagi para *IT* yang baru bekerja disana.

BAB II

KAJIAN PUSTAKA

2.1 Teori Dasar

Deskripsi teori paling tidak berisi tentang penjelasan terhadap variabel-variabel yang diteliti melalui pendefinisian, dan uraian yang lengkap dan mendalam dari berbagai referensi, sehingga ruang lingkup, kedudukan dan prediksi terhadap hubungan antara variabel yang akan diteliti menjadi lebih jelas dan terarah (Sugiyono, 2014:).

Pada bab ini akan dijelaskan tentang beberapa teori dasar antara lain kecerdasan buatan atau *Artificial Intelligence (AI)* dan beberapa subdisiplin ilmunya seperti logika *fuzzy (fuzzy logic)*, jaringan saraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*); *web*, basis data, dan *validitas* sistem.

2.1.1 Kecerdasan Buatan atau *Artificial Intelligence (AI)*

Menurut (Suyanto, 2014) kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia seperti penalaran, pembelajaran, pemecahan masalah, dan sebagainya.

Cerdas berarti memiliki pengetahuan, pengalaman, dan penalaran untuk membuat keputusan dan mengambil tindakan. Untuk membuat sebuah mesin menjadi cerdas (dapat bertindak seperti manusia) maka harus diberi bekal pengetahuan dan diberi kemampuan untuk menalar. Kecerdasan buatan memungkinkan komputer untuk berpikir atau menalar dan menirukan proses belajar manusia sehingga informasi baru dapat diserap sebagai pengetahuan, pengalaman, dan proses pembelajaran serta dapat digunakan sebagai acuan di masa-masa yang akan datang.

Menurut (Suyanto, 2014) tahun 1950 merupakan saat-saat awal dari *AI* yaitu saat awal sistem komputer dibangun dan ide-ide pembangunan mesin cerdas mulai terbentuk. Pada tahun 1950, Alan Turing menyimpan pertanyaan dalam pikirannya “apakah sebuah mesin mampu untuk berpikir”. Alan Turing melakukan percobaan yang cukup sederhana untuk menentukan apakah suatu mesin bisa dikatakan cerdas. Hasil percobaannya ini disebut dengan *Turing Test*. Dalam *Turing Test*, jika sebuah mesin mampu mengelabui seseorang yang menganggap mesin itu adalah manusia, maka mesin itu dianggap telah lulus dari tes kecerdasan (*intelligence test*).

Seseorang yang menjadi subjek percobaan diminta untuk menentukan terminal mana yang terkoneksi dengan komputer. Subjek boleh mengajukan pertanyaan, membuat pernyataan, menanyakan perasaan dan motivasi selama diperlukan. Jika subjek ternyata gagal menentukan terminal mana yang terkoneksi dengan komputer, maka komputer dinyatakan lulus tes dan dikatakan memiliki kesadaran

Kombinasi antara *AI* dengan bidang ilmu yang lainnya melahirkan sub disiplin ilmu dalam *AI*. Beberapa diantaranya adalah logika *fuzzy* (*fuzzy logic*), jaringan syaraf tiruan (*artificial neural network*), dan sistem pakar (*expert system*) (Sutojo, 2011: 12).

2.1.1.1 Fuzzy Logic

Fuzzy Logic, logika yang samar dan dapat diartikan pula sebagai suatu cara memetakan suatu ruang *input* dan ruang *output* yang dimiliki nilai selanjutnya. System logika *Fuzzy* mempunyai sifat yang mampu mengakomodasi ketidakpastian dalam proses akumulasi suatu data.

Menurut Suyanto *Fuzzy logic* didefinisikan sebagai suatu jenis *logic* yang bernilai ganda dan berhubungan dengan ketidakpastian dan ketidakbenaran parsial. Seperti pernah dibahas pada *propositional* dan *firs-order logic*, objek dasar dari suatu *logic* adalah *proposition* (proposisi) atau pernyataan yang menyatakan suatu fakta.

Menurut Kusumadewi dan Purnomo (2010: 1) *Logika fuzzy* merupakan salah satu komponen pembentuk *soft computing*. Logika *Fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zaedah pada tahun 1965. Dasar logika *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan element dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut.

Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang sesuai untuk diimplementasikan pada sistem, mulai dari sistem yang sederhana,

sistem kecil, *embedded system*, jaringan komputer, *multi-channel* atau *workstation* berbasis akuisisi data, dan sistem kontrol. Dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1, artinya suatu keadaan memungkinkan mempunyai dua nilai “Ya” dan “Tidak” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan di berbagai bidang seperti pada sistem diagnosis penyakit (dalam bidang kedokteran); pemodelan sistem pemasaran, sistem operasi (dalam bidang ekonomi); kendali kualitas air, prediksi adanya gempa bumi, klasifikasi dan pencocokan pola (Sutojo.T, Edy. M, dkk,2011: 211-212).

Ada beberapa keuntungan yang dapat diambil ketika menggunakan logika fuzzy untuk memecahkan suatu masalah yaitu :

1. Perancangannya tidak memerlukan persamaan matematik yang rumit
2. Mudah dimengerti
3. Memiliki toleransi terhadap data-data yang tidak tepat
4. Mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan
6. Dapat bekerja sama dengan teknik-teknik kendali secara konvensional
7. Logika *fuzzy* didasarkan pada bahasa alami

Sistem *inferensi fuzzy* adalah cara memetakan ruang input menuju ruang *output* menggunakan logika fuzzy. Empat elemen dasar sistem *inferensi fuzzy* antara lain (Sutojo.T, Edy. M, dkk,2011: 232):

1. Basis pengetahuan *fuzzy*, yaitu kumpulan aturan (*rule*) *fuzzy* dalam bentuk pernyataan *IF...THEN*.
2. *Fuzzifikasi*, yaitu proses untuk mengubah input sistem yang mempunyai nilai tegas menjadi variabel linguistik menggunakan fungsi keanggotaan yang disimpan dalam basis pengetahuan *fuzzy*.
3. Mesin *inferensi*, yaitu proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan yang telah ditetapkan pada basis pengetahuan *fuzzy*.
4. *Defuzzifikasi*, yaitu mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan *fuzzifikasi*.

Beberapa metode yang digunakan dalam sistem *inferensi fuzzy* adalah (Sutojo.T, Edy. M, dkk,2011: 233-237):

1. Metode *Tsukamoto*

Dalam inferensinya, metode *Tsukamoto* menggunakan tahapan sebagai berikut:

- a. *Fuzzifikasi*

- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin inferensi menggunakan fungsi implikasi *MIN* (*Minimum*)
- d. *Defuzzifikasi* menggunakan metode Rata-rata (*Average*)

2. Metode Mamdani

Metode ini sering digunakan karena strukturnya yang sederhana. Pada metode ini, untuk mendapatkan output diperlukan 4 tahapan sebagai berikut:

- a. *Fuzzifikasi*
- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Aplikasi fungsi implikasi menggunakan fungsi *MIN* (*Minimum*) dan komposisi antar-*rule* menggunakan fungsi *MAX* (*Maximum*) dengan menghasilkan himpunan *fuzzy* baru
- d. *Defuzzifikasi* menggunakan metode *Centroid* (Titik Tengah)

3. Metode Sugeno

Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985. Dalam metode ini, output sistem berupa konstanta atau persamaan linier. Dalam inferensinya, metode Sugeno menggunakan tahapan sebagai berikut:

- a. *Fuzzifikasi*

- b. Pembentukan basis pengetahuan *fuzzy* (*rule* dalam bentuk *IF...THEN*)
- c. Mesin *inferensi* menggunakan fungsi implikasi *MIN* (*Minimum*)
- d. *Defuzzifikasi* menggunakan metode Rata-rata (*Average*)

2.1.1.2 Jaringan Saraf Tiruan

Jaringan Syaraf Tiruan merupakan sistem komputasi dengan arsitektur dan operasi yang di ilhami dari pengetahuan tentang sel syaraf di dalam otak manusia. Prinsip kerja tersebut menjadikan Jaringan Syaraf Tiruan sangat sesuai untuk menyelesaikan berbagai masalah yang mempunyai tipe seperti otak manusia dalam menyelesaikan masalah (Laurence, 1994; Siang, 2015).

Jaringan saraf tiruan merupakan merupakan salah satu upaya manusia untuk memodelkan cara kerja atau fungsi sistem syaraf manusia dalam melaksanakan tugas tertentu. Pemodelan ini didasari oleh kemampuan otak manusia dalam mengorganisasikan sel-sel penyusunnya yang disebut neuron, sehingga mampu melaksanakan tugas-tugas tertentu, khususnya pengenalan pola dengan efektifitas yang sangat tinggi (Suyanto, 2014:169). Beberapa kelebihan yang dimiliki jaringan saraf tiruan antara lain:

1. Belajar adaptif, yaitu kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.
2. *Self-Organization*, yaitu kemampuan membuat organisasi sendiri atau representasi dari informasi yang diterimanya selama waktu belajar.
3. *Real Time Operation*, yaitu perhitungan jaringan saraf tiruan yang dapat dilakukan secara paralel sehingga perangkat keras yang dirancang dan diproduksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Selain mempunyai beberapa kelebihan, jaringan saraf tiruan juga mempunyai kelemahan-kelemahan:

1. Tidak efektif jika digunakan untuk melakukan operasi-operasi numerik dengan presisi tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmatika, operasi logika, dan simbolis.
3. Membutuhkan pelatihan untuk dapat beroperasi sehingga bila jumlah datanya besar, waktu yang digunakan untuk proses pelatihan sangat lama.

Keberhasilan penggunaan metode syaraf tiruan di tentukan oleh tiga factor, yaitu :

1. Pola-pola hubungan antar *neuron* yang disebut arsitektur jaringan.
2. Metode penentuan bobot penghubung yang disebut metode pembelajaran (*training/learning*).

3. Fungsi aktifasi yang digunakan.

Berdasarkan cara memodifikasi bobotnya, pelatihan jaringan saraf tiruan dibagi menjadi dua, yaitu (Sutojo, 2011: 301- 392):

1. Pelatihan dengan Supervisi (pembimbing)

Dalam pelatihan ini, jaringan dipandu oleh sejumlah pasangan data (masukan dan target) yang berfungsi sebagai pembimbing untuk melatih jaringan hingga diperoleh bobot yang terbaik. Algoritma yang termasuk dalam pelatihan dengan supervisi antara lain:

a. *Hebb-Rule*

Model ini diperkenalkan oleh D.O. Hebb yang menggunakan cara menghitung bobot dan bias secara iteratif dengan memanfaatkan model pembelajaran dengan supervisi sehingga bobot dan bias dapat dihitung secara otomatis tanpa harus melakukan cara coba-coba. Arsitektur jaringan ini terdiri dari beberapa unit *input* dihubungkan langsung dengan sebuah unit *output*, ditambah dengan sebuah bias.

b. *Perceptron*

Model ini ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969). Model jaringan ini merupakan model yang terbaik pada saat itu. Algoritma pelatihan *perceptron* digunakan baik untuk *input* biner maupun bipolar, dengan θ tertentu.

c. *Delta-Rule*

Selama pelatihan pola, *Delta-Rule* akan mengubah bobot dengan cara meminimalkan *error* antara *output* jaringan dengan target.

d. *Backpropagation*

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat *error* keluaran. Pelatihan jaringan ini terdiri dari 3 tahap, yaitu tahap perambatan maju (*forward propagation*), tahap perambatan balik, dan tahap perubahan bobot dan bias. Arsitektur jaringan ini terdiri dari *input layer*, *hidden layer*, dan *output layer*.

e. *Heteroassociative Memory*

Jaringan saraf *heteroassociative memory* adalah jaringan yang dapat menyimpan kumpulan pengelompokan pola dengan cara menentukan bobot-bobotnya sedemikian rupa. Algoritma pelatihan yang biasa digunakan adalah *Hebb-Rule*.

f. *Bidirectional Associative Memory (BAM)*

Bidirectional Associative Memory (BAM) adalah model jaringan saraf yang memiliki 2 lapisan, yaitu lapisan *input* dan lapisan *output* yang mempunyai hubungan timbal balik antara keduanya (bersifat *bidirectional*). Arsitektur jaringan ini terdiri dari 3 *neuron* pada lapisan *input* dan 2 *neuron* pada lapisan *output*. Model jaringan ini terbagi menjadi 2 jenis yaitu *BAM* Diskrit dan *BAM* Kontinu.

g. *Learning Vector Quantization (LVQ)*

Learning Vector Quantization (LVQ) adalah suatu model pelatihan pada lapisan kompetitif terawasi yang akan belajar secara otomatis untuk mengklasifikasikan vektor-vektor *input* ke dalam kelas-kelas tertentu.

2. Pelatihan tanpa Supervisi

Dalam pelatihan ini, tidak ada pembimbing yang digunakan untuk memandu proses pelatihan. Jaringan hanya diberi *input* tetapi tidak mendapatkan target yang diinginkan sehingga modifikasi bobot pada jaringan dilakukan menurut parameter tertentu. Model jaringan yang termasuk dalam pelatihan tanpa supervisi adalah jaringan kohonen yang diperkenalkan oleh Prof. Teuvo Kohonen pada tahun 1982. Pada jaringan kohonen, *neuron-neuron* pada suatu lapisan data akan menyusun dirinya sendiri berdasarkan *input* nilai tertentu dalam suatu *cluster*. *Cluster* yang dipilih sebagai pemenang adalah *cluster* yang mempunyai vektor bobot paling cocok dengan pola *input*, yaitu *cluster* yang memiliki jarak yang paling dekat.

2.1.1.3 Sistem Pakar

Sistem pakar mulai dikembangkan pada pertengahan 1960, ditandai dengan lahirnya sistem pakar pertama bernama *General-purpose Problem Solver (GPS)* yang dikembangkan oleh Newel dan Simon. Kemudian bermunculan sistem pakar lain di berbagai bidang seperti *MYCIN* untuk diagnosis penyakit, *DENDRAL* untuk mengidentifikasi struktur molekul campuran yang tak dikenal, *XCON & XSEL* untuk membantu konfigurasi sistem komputer besar, *SOPHIE* untuk analisis sirkuit elektronik, *Prospector* digunakan di bidang geologi untuk membantu mencari dan menemukan deposit, *FOLIO* digunakan untuk membantu memberikan keputusan bagi seorang manajer dalam masalah stok dan investasi, *DELTA* dipakai untuk pemeliharaan lokomotif listrik diesel, dan sebagainya (Suyanto, 2014: 1).

Menurut (Kusumadewi Sri 2009: 109) sistem pakar merupakan salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia. Sistem pakar adalah suatu sistem yang dirancang untuk dapat menirukan keahlian seorang pakar dalam menjawab pertanyaan dan memecahkan suatu masalah. Dengan bantuan sistem pakar, seseorang yang bukan pakar dapat menjawab pertanyaan, menyelesaikan masalah serta mengambil keputusan yang biasanya dilakukan oleh seorang pakar.

Menurut (Sutojo, 2011:14-15) sistem pakar dapat digunakan oleh orang awam yang bukan pakar untuk meningkatkan kemampuan dalam memecahkan masalah. Sistem pakar juga dapat digunakan oleh pakar sebagai *assistant* yang berpengetahuan, serta digunakan untuk memperbanyak atau menyebarkan sumber pengetahuan yang semakin langka.

Suatu sistem dikatakan sebagai sistem pakar jika memiliki ciri-ciri sebagai berikut (Kusumadewi Sri, 2009: 122):

1. Terbatas pada domain keahlian tertentu
2. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti
3. Dapat menjelaskan alasan-alasan dengan cara yang dapat dipahami
4. Bekerja berdasarkan kaidah tertentu
5. Mudah dimodifikasi
6. Basis pengetahuan dan mekanisme inferensi diletakkan terpisah
7. Keluarannya (*output*) bersifat anjuran

Expert System - is a computer program that employs the knowledge of human experts to solve problems that usually would require human intelligence. This kind of programs represents the expertise knowledge, about a specific class of problems, as data or rules that can be called upon when needed. Expert systems can also provide some analysis of the problems and they can even recommend to users various actions in order to perform improvements and rectifications. Expert systems arrive at conclusions using reasoning capabilities. The focus of this paper requires human intelligence which can be substituted by an expert system and potentially improve productivity in Effective Pedagogy and Academic Quality Assurance, save time and money, preserve valuable knowledge, and improve learning and understanding.(Asabere & Enguah, 2012)

Sistem pakar pada penelitian ini menggunakan model representasi pengetahuan berbasis kaidah produksi. Menurut Firebaugh (1988) dalam (Kusumadewi Sri, 2009: 113-114) struktur sistem pakar yang berbasis kaidah produksi terdiri dari 4 komponen, yaitu:

1. Antarmuka pemakai

Antarmuka merupakan penghubung antara pemakai dengan sistem pakar. Komponen ini berfungsi sebagai alat komunikasi antara sistem dan pengguna (*user*) yang penting sekali bagi pengguna. Komponen ini harus didesain sedemikian rupa sehingga efektif dan mudah digunakan terutama bagi pengguna yang tidak ahli dalam bidang yang diterapkan pada sistem pakar.

2. Basis pengetahuan

Basis pengetahuan adalah komponen yang berisi sekumpulan kaidah yang berasal dari pengetahuan dalam domain tertentu dan secara umum disajikan dalam bentuk kaidah produksi (*IF...THEN...*). Pengetahuan pakar yang disajikan dalam format tertentu didapat dari sekumpulan pengetahuan pakar dan sumber-sumber pengetahuan lainnya seperti buku-buku, jurnal ilmiah, majalah, maupun dokumentasi tercetak lainnya. Basis pengetahuan diletakkan terpisah dari mesin inferensi agar pengembangan pengetahuan sistem pakar dapat dilakukan secara leluasa tanpa mengganggu mesin inferensi.

3. Struktur kontrol (Mesin Inferensi)

Struktur kontrol merupakan *interpreter* kaidah atau mesin inferensi yang menggunakan pengetahuan-pengetahuan yang tersimpan dalam basis pengetahuan untuk memecahkan atau menyelesaikan permasalahan yang ada. Inferensi merupakan proses untuk menghasilkan informasi berupa konklusi logis berdasarkan informasi yang tersedia atau fakta yang diketahui.

Dalam melakukan proses inferensi, sistem pakar memerlukan adanya proses pengujian kaidah-kaidah dalam urutan tertentu untuk mencari suatu kondisi yang sesuai dengan kondisi awal atau untuk memastikan kondisi yang sedang berjalan sudah dimasukkan ke dalam *database*. Proses pengujian itu disebut dengan peruntutan atau penalaran, yaitu proses pencocokan fakta atau kondisi tertentu yang tersimpan dalam basis pengetahuan maupun pada memori kerja dengan kondisi yang

dinyatakan dalam premis atau bagian kondisi pada suatu kaidah atau aturan (Kusumadewi Sri, 2009: 14).

Ada beberapa konsep penalaran yang dapat digunakan oleh mesin inferensi yaitu:

1. Penalaran maju (*forward chaining*)

Konsep ini dapat juga disebut sebagai pencarian yang dimotori data (*data driven search*). Runut maju melakukan proses perunutan (penalaran) dimulai dari premis-premis atau informasi masukan (*IF*) terlebih dahulu kemudian menuju konklusi atau *derived information (THEN)*. Konsep ini dapat dimodelkan sebagai berikut:

IF (informasi masukan) *THEN* (konklusi)

Informasi masukan dapat berupa suatu pengamatan sedangkan konklusi dapat berupa diagnosa sehingga dapat dikatakan jalannya penalaran runut maju dimulai dari pengamatan menuju diagnosa. Pada metode ini, sistem tidak melakukan praduga apapun terhadap konklusi, namun sistem akan menerima semua gejala yang diberikan pengguna lalu sistem akan memeriksa gejala-gejala tersebut dan selanjutnya mencocokkan dengan konklusi yang sesuai (Kusumadewi Sri, 2009: 19).

2. Penalaran mundur (*backward chaining*)

Secara umum konsep ini diaplikasikan ketika tujuan ditentukan sebagai kondisi atau keadaan awal. Konsep ini disebut juga *goal-driven search*. Arah penalaran atau perunutan dalam konsep ini berlawanan dengan *forward chaining*. Konsep ini dapat dimodelkan sebagai berikut:

Proses penalaran pada *backward chaining* dimulai dari tujuan kemudian merunut balik ke jalur yang mengarah ke tujuan tersebut, untuk membuktikan bahwa bagian kondisi pada kaidah atau aturan benar-benar terpenuhi. Proses *internal* selalu memeriksa konklusi (tujuan) terlebih dahulu sebagai praduga awal, kemudian memeriksa dan memastikan gejala-gejala (kondisi) telah terpenuhi dan selanjutnya mengeluarkan konklusi sebagai *output*. Jika sistem menemukan ada bagian kondisi yang tidak terpenuhi maka sistem akan memeriksa konklusi (tujuan) pada aturan atau kaidah berikutnya (Kusumadewi Sri, 2009: 21-22).

3. Faktor kepastian (*certainty factor*)

Faktor kepastian (*certainty factor*) menyatakan kepercayaan dalam sebuah kejadian (atau fakta atau hipotesis) berdasarkan bukti atau penilaian pakar (Turban, 2005). *Certainty factor* menggunakan suatu nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data. *Certainty factor* memperkenalkan konsep keyakinan dan ketidakpercayaan yang kemudian diformulasikan ke dalam rumusan dasar sebagai berikut:

$CF(H,E) = MB(H,E) - MD(H,E)$ **Rumus 2.1** Hasil Akhir *Certainty Factor*

$CF(H,E)$: *certainty factor*

$MB(H,E)$: ukuran kepercayaan (*measure of increased belief*) terhadap hipotesis H yang jika diberikan *evidence* E (antara 0 dan 1)

$MD(H,E)$: ukuran ketidakpercayaan (*measure of increased disbelief*) terhadap *evidence* H, jika diberikan *evidence* E (antara 0 dan 1).

Bentuk dasar rumus *certainty factor* sebuah aturan JIKA E MAKA H adalah seperti ditunjukkan oleh persamaan 2 berikut:

$$CF(H,e) = CF(E,e) * CF(H,E) \quad \text{Rumus 2.2 Dasar Rumus Certainty Factor}$$

Dimana:

$CF(E,e)$: *certainty factor evidence* E yang dipengaruhi oleh *evidence* e.

$CF(H,E)$: *certainty factor* hipotesis dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika $CF(E, e) = 1$.

$CF(H,e)$: *certainty factor* hipotesis yang dipengaruhi oleh *evidence* e.

Jika semua *evidence* pada *antecedent* diketahui dengan pasti maka persamaannya akan menjadi:

$$CF(H,e) = CF(H,E) \quad \text{Rumus 2.3 Nilai Kepastian}$$

Dalam aplikasinya, $CF(H,E)$ merupakan nilai kepastian yang diberikan oleh pakar terhadap suatu aturan, sedangkan $CF(E,e)$ merupakan nilai kepercayaan yang diberikan oleh pengguna terhadap gejala yang dialaminya.

Sebagai contoh, berikut ini adalah sebuah aturan dengan CF yang diberikan oleh

seorang pakar:

JIKA Mulai Batuk

DAN Badan Panas

DAN Pilek

DAN Pusing

DAN Mual

MAKA Demam, CF: 0,7

Metode MCYIN untuk menggabungkan evidence pada antecedent sebuah aturan yang ditunjukkan pada tabel berikut

Tabel 2.1 Aturan kombinasi MCYIN

Evidence E	Ketidakpastian anteseden
E1 AND E2	$\text{Min}[\text{CF}(\text{H},\text{E}_1),\text{CF}(\text{H},\text{E}_2)]$
E1 AND E2	$\text{Max}[\text{CF}(\text{H},\text{E}_1),\text{CF}(\text{H},\text{E}_2)]$
NOT E	$-\text{CF}(\text{H},\text{E})$

Dengan mewawancarai seorang pakar, nilai CF(rule) didapat dari interpretasi "term" dari pakar, yang diubah menjadi nilai CF tertentu sebagai berikut:

Tabel 2.2 Bobot Certainty Factor

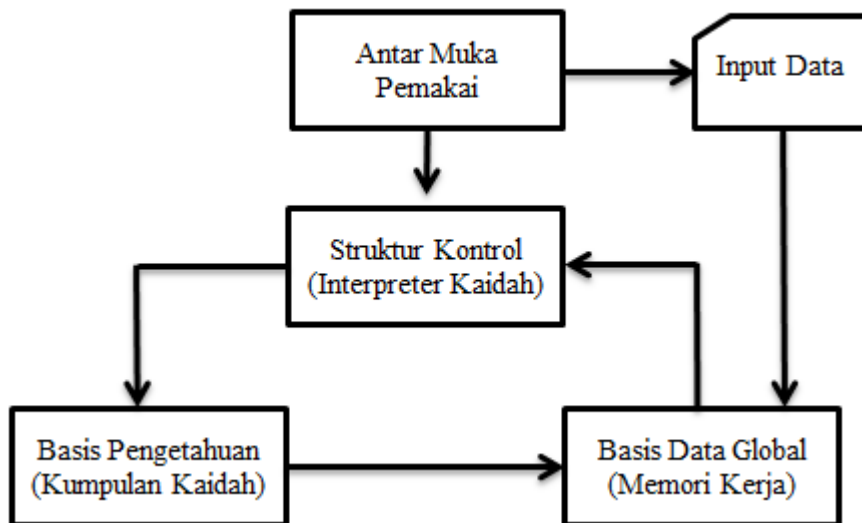
Uncertain Term	CF
Pasti tidak (<i>definitely not</i>)	-1.0
Hampir pasti tidak (<i>almost certainty not</i>)	-0.8
Mungkin tidak (<i>probably not</i>)	0.6
Barang kali tidak (<i>Maybe not</i>)	-0.4
Tidak Tahu (<i>Unknown</i>)	-0.2 s/d 0.2
Barang kali (<i>Maybe</i>)	0.4
Mungkin (<i>probably</i>)	0.6
Hampir pasti (<i>almost certainty</i>)	0.8
Pasti (<i>definitely</i>)	1

Kelebihan Certainty Factor:

- a. Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendeteksi suatu kerusakan sebagai salah satu contohnya.
- b. Perhitungan dengan menggunakan metode ini dalam sekali hitung hanya dapat mengelola dua data saja sehingga keakuratan data dapat terjaga.

Kekurangan Metode Certainty Factor :

- a. Ide umum dari pemodelan ketidakpastian manusia dengan menggunakan numerik metode certainty factors biasanya diperdebatkan. Sebagian orang akan membantah pendapat bahwa formula untuk metode *certainty factor* diatas memiliki sedikit kebenaran.
- b. Metode ini hanya dapat mengolah ketidakpastian/kepastian hanya dua data saja. Perlu dilakukan beberapa kali pengolahan data untuk data yang lebih.
- c. Nilai CF yang diberikan bersifat subyektif karena penilaian setiap pakar bisa saja berbeda-beda tergantung pengetahuan dan pengalaman pakar.



Gambar 2.1 Struktur Sistem Pakar
(Sumber: Data Penelitian, 2017)

2.1.2 Web

Menurut Putra Ifana Yoka (2015:9) web merupakan salah satu layanan yang didapat oleh pengguna komputer yang terhubung dengan internet. *Web* pada awalnya adalah ruang informasi dalam internet, dengan menggunakan teknologi *hypertext*. Pengguna dituntun untuk menemukan informasi dengan mengikuti *link* yang disediakan dalam dokumen web yang ditampilkan dalam *browser web*. Sekarang *web* menjadi standar *interface* pada layanan-layanan yang ada di internet seperti komunikasi melalui *e-mail*, *chatting*, transaksi bisnis, pencarian informasi, dan sebagainya.

2.1.3 Database (Basis Data)

Menurut A.S. dan Shalahuddin (2015: 43-44) sistem database adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data atau informasi yang sudah diolah dan membuat informasi tersedia saat dibutuhkan. *Database* adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Kebutuhan basis data meliputi memasukkan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan. Salah satu bentuk basis data yang dibutuhkan dalam sebuah sistem yaitu *Database Management System (DBMS)*. *DBMS* adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola (melakukan perubahan seperti penambahan, pengurangan dan penghapusan data yang berada dalam field suatu basis data) dan menampilkan data. Syarat *minimal* dari *DBMS* antara lain (A.S. dan Shalahuddin, 2013: 44-45):

Database atau biasa disebut basis data merupakan kumpulan data yang saling berhubungan. Data tersebut biasanya terdapat dalam tabel-tabel yang saling berhubungan satu sama lain, dengan menggunakan field/kolom pada tiap tabel yang ada

2.1.4 Validasi Sistem

Validasi mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa sistem atau perangkat lunak yang dibangun telah sesuai dengan yang

diharapkan. Beberapa pendekatan dalam melakukan pengujian untuk *validasi* sistem antara lain (Kusumadewi Sri, 2009: 275-276):

1. *Black-Box Testing* (pengujian kotak hitam)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Tujuannya untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari sistem atau perangkat lunak sesuai dengan *spesifikasi* yang dibutuhkan.

Pengujian dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan menggunakan sistem atau perangkat lunak apakah sesuai dengan *spesifikasi* yang dibutuhkan. Kasus uji yang dibuat untuk melakukan *black-box testing* harus dibuat dengan kasus benar dan kasus salah.

2. *White-Box Testing* (pengujian kotak putih)

Pendekatan ini dilakukan dengan menguji sistem atau perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan *spesifikasi* yang dibutuhkan. *White-box testing* dilakukan dengan memeriksa logika dari kode program. Pembuatan kasus uji dapat mengikuti standar pengujian dari standar pemrograman yang ada.

2.2 Variabel Penelitian

Variabel menurut Sugiono (2012: 38) adalah segala sesuatu yang berbentuk apa saja yang diterapkan oleh peneliti untuk dipelajari sehingga diperoleh informasi tentang hal tersebut.

Mesin kasir adalah salah satu alat yang paling di perlukan dalam sebuah perusahaan yang bergerak di bidang *retail* untuk digunakan untuk melakukan transaksi, dan tanpa mesin kasir maka kinerja penjualan tidak akan lancar, karena melakukan transaksi secara manual akan memerlukan waktu yang sangat lama, sehingga akan mengakibatkan antrian yang panjang yang membuat *costumer* tidak nyaman. Pada Penelitian ini penulis akan membahas tentang penyebab kerusakan mesin kasir.

Adapun penyebab terjadi nya kerusakan pada mesin kasir yang di akibatkan beberapa element yang saling berkaitan yaitu mulai dari *hardware*, *software*, jaringan dan *user* nya sendiri.

1. Mesin Kasir *Blue Screen*

Layar mesin kasir akan secara tiba-tiba berubah warna menjadi warna biru, dan biasanya kerusakan ini terjadi pada saat mesin kasir pertama kali di hidupkan. Jika mesin kasir mengalami hal ini ada beberapa hal yang menjadi penyebabnya yaitu, memori yang berada dalam mesin POS longgar yang diakibatkan pergerakan dari mesin itu sendiri yang mengakibatkan memori rusak. Pergerakan terjadi saat perubahan tempat dan lain-lain

2. Tidak mengeluarkan struk

Mesin kasir tidak mengeluarkan diakibatkan jika kertas yang digunakan habis, dan masalah ini juga bisa diakibatkan karena sensor *printer* rusak yang diakibatkan kertas yang menggoyang sensor

3. Tidak dapat *scanning barcode* barang

Mesin kasir tidak dapat melakukan *scanning barcode* barang diakibatkan karena *scanner* belum di *update* sehingga *barcode* yang di scan tidak terdaftar di *server* maupun di *scanner*

4. *Off Line*

Mesin kasir *offline* juga disebabkan karena *server down*, adanya *IP address* yang sama di kasir tersebut, *port* di mesin kasir rusak sehingga tidak dapat menerima jaringan melalui kabel.

2.3 Software Pendukung

Aplikasi yang penulis gunakan untuk mengimplementasikan penelitian tentang Sistem pakar deteksi penyebab kerusakan pada mesin kasir ini adalah berbasis *WEB*. Pemrograman web diambil dari 2 suku kata yaitu pemrograman dan *web*.

Pemrograman diartikan proses, cara, perbuatan program. Sedangkan *Web* adalah jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui *protokol transfer hypertext*. Orang banyak mengenal *web* dengan istilah *WWW (world wide web)*.

World Wide Web adalah layanan internet yang paling populer saat ini internet mulai dikenal dan digunakan secara luas setelah adanya layanan *WWW*. *WWW* adalah halaman-halaman *website* yang dapat saling terkoneksi satu dengan lainnya (*hyperlink*) yang membentuk samudra belantara informasi. *WWW* berjalan dengan *protokol HyperText Transfer Protokol (HTTP)*.

Halaman *Web* merupakan file teks murni (*plain text*) yang berisi *sintaks-sintaks HTML* yang dapat dibuka/ dilihat/ diterjemahkan dengan *Internet Browser*. *Sintaks HTML* mampu memuat konten text, gambar, *audio*, *video* dan animasi. Kini internet identik dengan *web*, karena kepopuleran *web* sebagai standar *interface* pada layanan-layanan yang ada di internet, dari awalnya sebagai penyedia informasi, ini digunakan juga untuk komunikasi dari email sampai dengan *chatting*, sampai dengan melakukan transaksi bisnis (*commerce*).

Banyak keuntungan yang diberikan oleh Aplikasi berbasis *Web* daripada aplikasi berbasis desktop, sehingga aplikasi berbasis *web* telah diadopsi oleh perusahaan sebagai bagian dari strategi teknologi informasinya, karena beberapa alasan :

1. Akses informasi mudah,
2. *Setup server* lebih mudah
3. Informasi mudah didistribusikan
4. Bebas *platform*, informasi dapat disajikan oleh *browser web* pada sistem operasi mana saja karena adanya standar dokumen berbagai tipe data dapat disajikan.

Dalam bahasa pemrograman berbasis *web* maka kita gunakan bahasa program berikut :

2.3.1 *Xampp*



Gambar 2.2 Logo *Xampp*
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Menurut Putra Yoka (2015: 25) *server web* adalah komputer yang digunakan untuk menyimpan dokumen-dokumen *web* yang akan melayani permintaan dokumen *web* dari kliennya. *XAMPP* adalah sebuah perangkat lunak *web server apache* yang didalamnya sudah tersedia *database server MySQL* dan dapat mendukung pemrograman *PHP*. *XAMPP* merupakan *software* yang mudah digunakan, gratis, dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cukup dengan menginstal *XAMPP* sudah tersedia *Apache Web Server*, *MySQL Database Server*, *PHPsupport* (*PHP 4* dan *PHP 5*) dan beberapa modul lainnya.

2.3.2 *phpMyAdmin*



Gambar 2.3 Logo *phpMyAdmin*
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

phpMyAdmin adalah perangkat lunak gratis yang ditulis dalam bahasa pemrograman *PHP* bertujuan untuk menangani administrasi *MySQL* melalui *web*. *phpMyAdmin* mendukung berbagai operasi pada *MySQL* dan *MariaDB*. Operasi-operasi yang sering digunakan seperti mengelola *database*, tabel, kolom, relasi, indeks, *users*, *permissions*, dan lain-lain, dapat dilakukan melalui antarmuka pengguna dengan tetap dapat mengeksekusi pernyataan *SQL* secara langsung (Abdulloh Rohi (2015: 4)).

phpMyAdmin merupakan aplikasi *web* yang bersifat *open source* (sumber terbuka) sejak pertama dibuat dan dikembangkan. Dengan dukungan dari banyak *developer* dan *translator*, aplikasi *phpMyAdmin* mengalami perkembangan yang cukup pesat dengan ketersediaan bahasa yaitu kurang dari 60 bahasa yang sudah didukung *phpMyAdmin*.

2.3.3 Hypertext Preprocessor (PHP)



Gambar 2.4 Logo *PHP*

(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

PHP sudah menjadi bahasa *scripting* umum yang banyak digunakan dikalangan *developer web*. Mempunyai banyak kelebihan menjadi alasan utama kenapa *PHP* lebih dipilih sebagai basis umum dalam membuat sebuah web

Menurut Welling dan Thomson(2009: 2) *PHP Hypertext Preprocessor (PHP)* adalah bahasa *scripting server-side* yang didesain khusus untuk sebuah *web*. Kode *PHP* yang akan dijalankan dapat ditanamkan atau disisipkan dalam sebuah halaman *HTML (Hyper Text Markup Language)*. Kode *PHP* akan diinterpretasikan pada *web server* dan menghasilkan *HTML* atau *output* lainnya yang akan dilihat oleh pengunjung. Beberapa pesaing utama *PHP* adalah *Perl, Microsoft ASP.NET, Ruby on Rails, JavaServer Pages (JSP)*, dan *ColdFusion*. Dibandingkan dengan produk-produk tersebut, *PHP* memiliki banyak kelebihan, yaitu Abdulloh Rohi (2015: 3):

1. Kinerja

Kinerja *PHP* sangat cepat. Cukup dengan menggunakan *server* tunggal yang tidak mahal sudah dapat melayani jutaan hits per hari.

2. Skalabilitas

PHP memiliki arsitektur “*shared-nothing*” yang berarti bahwa skala *horizontal* dengan sejumlah besar komoditas *server* dapat diterapkan secara efektif dengan biaya yang murah.

3. Integrasi *database*

PHP memiliki koneksi bawaan (asli) yang tersedia untuk beberapa sistem *database* seperti: *MySQL*, *PostgreSQL*, *Oracle*, *dbm*, *FilePro*, *DB2*, *Hyperwave*, *Informix*, *Interbase*, dan *Sybase database*.

4. *Built-in libraries*

PHP dirancang untuk digunakan pada *web* sehingga memiliki banyak fungsi *built-in* untuk dapat melakukan banyak tugas *web* yang berguna seperti menghasilkan gambar dengan cepat, koneksi ke layanan *web* dan layanan jaringan lainnya, *parse XML (eXtended Markup Language)*, mengirim *e-mail*, bekerja dengan *cookie*, dan menghasilkan dokumen *PDF*. Semua itu dilakukan cukup dengan beberapa baris kode.

5. Biaya yang murah

Menggunakan *PHP* tidak membutuhkan biaya alias gratis. *PHP* versi terbaru dapat di-*download* setiap saat dari situs resminya tanpa biaya.

6. Mudah dipelajari dan digunakan

Sintaks *PHP* didasarkan pada bahasa pemrograman lainnya, terutama bahasa *C* dan *Perl*. Jika pengguna sudah mengetahui bahasa *C* atau *Perl*, atau bahasa *C* lainnya

seperti *C++* atau *Java*, maka akan segera menjadi produktif dalam menggunakan *PHP*.

7. Dukungan *object-oriented* yang kuat

PHP memiliki rancangan fitur *object-oriented* yang baik seperti pewarisan (*inheritance*), atribut dan metode *private* dan *protected*, kelas dan metode *abstract*, antarmuka (*interfaces*), *constructors*, *destructors*, dan *iterators*.

8. Portabilitas

PHP tersedia untuk banyak sistem operasi yang berbeda. Kode *PHP* dapat ditulis pada sistem operasi *Unix* seperti *Linux*, *FreeBSD*, *Solaris* dan *IRIX*, *Operating System X (OS X)* maupun sistem operasi *Microsoft Windows* yang memiliki versi yang berbeda-beda. Kode yang ditulis biasanya akan bekerja tanpa dimodifikasi pada sistem lain yang menjalankan *PHP*.

9. Fleksibilitas dalam pembangunan

PHP memungkinkan penerapan tugas-tugas sederhana secara ringkas dan sama-sama mudah beradaptasi dalam penerapan aplikasi yang besar menggunakan kerangka kerja berdasarkan pola desain seperti *Model View Control (MVC)*.

10. Tersedianya kode sumber (*source code*)

PHP menyediakan akses ke kode sumber *PHP* (bersifat *open-source*), tidak seperti produk komersial lainnya yang bersifat *closed-source*. Pada *PHP*, dapat dilakukan modifikasi atau penambahan di dalamnya secara bebas tanpa harus menunggu produsen untuk mengeluarkan *patches* (program kecil untuk perbaikan sistem).

11. Tersedianya *support* (dukungan) dan dokumentasi

Zend Technologies, perusahaan di balik mesin yang menggerakkan *PHP*, mendanai pengembangan *PHP* dengan menawarkan *support* (dukungan) dan perangkat lunak terkait pada basis komersial. Dokumentasi dan komunitas *PHP* adalah sumber daya yang matang dan kaya akan informasi yang dapat dibagi.

2.3.4 *HTML (Hyper Text Markup Language)*



Gambar 2.5 Logo HTML (*Hyper text markup language*)
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Hyper text markup language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web internet* dan pemformatan *hyperteks* sederhana yang ditulis dalam berkas format *ASCII* agar dapat menghasilkan tampilan wujud yang terintegrasi, dengan kata lain berkas yang dibuat dalam perangkat lunak pengolah kata dan di simpan dalam format *ASCII* normal sehingga menjadi halaman *web* dengan perintah-perintah *HTML*. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan *SGML (Standar Generalized markup language)*, *HTML* adalah sebuah standar yang

digunakan secara luas dengan menampilkan halaman *web*. *HTML* saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (W3C)*. *HTML* dibuat oleh kolaborasi *Cailliau* dengan *Berners-Lee* Robert ketika mereka bekerja di *CERN* pada tahun 1989. *CERN* adalah lembaga penelitian fisika energi tinggi di Jenewa.

Sejarah *HTML*

Pada tahun 1980 seorang ahli fisika, Tim Berners Lee, dan juga seorang kontraktor di *CERN* (Organisasi Eropa untuk Riset Nuklir) mengusulkan dan menyusun *ENQUIRE*, sebuah sistem untuk ilmuwan *CERN* dalam membagi dokumen. Sembilan tahun kemudian, Berners-Lee mengusulkan adanya sistem markah berbasis internet. Berners-Lee menspesifikasikan *HTML* dan menulis jaringan beserta perangkat lunaknya di akhir 1990. Pada tahun yang sama, Berners-Lee dan Robert Cailliau, insinyur sistem data *CERN* berkolaborasi dalam sebuah permintaan untuk pendanaan, namun tidak diterima secara resmi oleh *CERN*. Di catatan pribadinya sejak 1990 dia mendaftarkan beberapa dari banyak daerah yang menggunakan "*hypertext*" dan pertama-tama menempatkan sebuah *ensiklopedia*. Penjelasan pertama yang dibagi untuk umum dari *HTML* adalah sebuah dokumen yang disebut Tanda *HTML*, pertama kali disebutkan di *internet* oleh Tim Berners-Lee pada akhir 1991. Tanda ini menggambarkan 18 elemen awal mula, versi sederhana dari *HTML*. Kecuali untuk *tag hyperlink*, yang sangat dipengaruhi oleh *SGMLguid*,

in-house Standard Generalized Markup Language (SGML) berbasis format dokumen di *CERN*. Sebelas elemen ini masih ada di *HTML 4*.

HTML adalah bahasa markah yang digunakan peramban untuk menafsirkan dan menulis teks, gambar dan bahan lainnya ke dalam halaman *web* secara *visual* maupun suara. Karakteristik dasar untuk setiap item dari markah *HTML* didefinisikan di dalam peramban, dan karakteristik ini dapat diubah atau ditingkatkan dengan menggunakan tambahan halaman *web designer CSS* (Naista David, 2016: 2). Banyak elemen teks ditemukan di laporan teknis *ISO* pada tahun 1988 TR 9537 Teknik untuk menggunakan *SGML*, yang pada gilirannya meliputi fitur bahasa format teks awal seperti yang digunakan oleh komandan *RUN OFF* dikembangkan pada awal 1960-an untuk sistem operasi: perintah-perintah format ini berasal dari perintah yang digunakan oleh pengetik untuk memformat dokumen *CTSS* secara manual. Namun, konsep *SGML* dari markah umum didasarkan pada unsur-unsur daripada hanya efek cetak, dengan pemisahan struktur dan markah juga; *HTML* telah semakin bergerak ke arah ini dengan *CSS*.

Dokumen *HTML* mirip dengan dokumen tulisan biasa, hanya dalam dokumen ini sebuah tulisan bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan *TAG* tertentu. Penjelajah *web* digunakan untuk menginterpretasikan susunan halaman ke gaya *built-in* penjelajah *web* dengan menggunakan jenis tulisan, tab, warna, garis, dan perataan *text* yang dikehendaki ke komputer yang menampilkan halaman *web*. Salah satu hal Penting tentang eksistensi *HTML* adalah tersedianya *Lingua franca* (bahasa Komunikasi) antar komputer dengan kemampuan

berbeda. Pengguna *Macintosh* tidak dapat melihat tampilan yang sama sebagaimana tampilan yang terlihat dalam *pc* berbasis *Windows*. Pengguna *Microsoft Windows* pun tidak akan dapat melihat tampilan yang sama sebagaimana tampilan yang terlihat pada pengguna yang menggunakan Produk-produk *Sun Microsystems*. namun demikian pengguna-pengguna tersebut dapat melihat semua halaman *web* yang telah diformat dan berisi *Grafika* dan *Pranala*.

2.3.5 CSS(*Cascading Style sheet*)



Gambar 2.6 Logo CSS (*Cascading Style sheet*)
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan

adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda (Naista David, 2016: 3).

Sejarah CSS

Nama CSS didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang kemudian membentuk hubungan ayah-anak (*parent-child*) pada setiap style. CSS sendiri merupakan sebuah teknologi *internet* yang direkomendasikan oleh *World Wide Web Consortium* atau W3C pada tahun 1996. Setelah CSS distandarisasikan, *internet explorer* dan *netscape* melepas browser terbaru yang telah sesuai atau hampir mendekati dengan standard CSS. Fakta menggunakan CSS diantaranya:

- a. telah didukung oleh kebanyakan *browser* versi terbaru, tetapi tidak didukung oleh *browser-browser* lama.
- b. lebih *fleksibel* dalam penempatan posisi *layout*, dalam *layouting* CSS, kita mengenal *Z-index* untuk menempatkan objek dalam posisi yang sama.
- c. menjaga *HTML* dalam penggunaan *tag* yang *minimal*, hal ini berpengaruh terhadap ukuran berkas dan kecepatan pengunduhan.
- d. dapat menampilkan konten utama terlebih dahulu, sementara gambar dapat di tampilkan sesudahnya.
- e. penerjemah CSS setiap *browser* berbeda, tata letak akan berubah jika di lihat di berbagai *browser*.

2.3.6 JavaScript



Gambar 2.7 Logo *JavaScript*
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Menurut (Abdullah Rohi, 2015: 3) *JavaScript* merupakan modifikasi dari bahasa *C++* dengan pola penulisan yang lebih sederhana. Beberapa hal penting dalam *JavaScript* adalah:

1. Menggunakan blok awal “{“ dan blok akhir “}”.
2. *Automatic conversion* dalam pengoperasian tipe data yang berbeda.
3. *Sensitive case*, yaitu membedakan antara huruf kecil dan huruf *capital* sehingga harus berhati-hati dalam menggunakan nama variabel , fungsi, dan lain-lain.
4. *Extension* umumnya menggunakan “*.js”.
5. Setiap *statement* dapat diakhiri tanda baca *semi colon* (;).
6. Jika tidak didukung oleh *browser* versi lama, *script* dapat disembunyikan diantara *tag* “<!--“ dan “-->”.
7. Jika program dalam satu baris terlalu panjang dapat disambung ke baris berikutnya menggunakan karakter *backslash* (\).

2.3.7 *jQuery*



Gambar 2.8 Logo *jQuery*
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Pertama kali *jQuery* dirilis oleh John Resig pada tahun 2006 (Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015). Pada perkembangannya *jQuery* tidak sekedar sebagai *library javascript*, namun memiliki keandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak *developer web* menggunakannya.

jQuery adalah sebuah *framework* berbasiskan *JavaScript*, yang berisi kumpulan kode atau fungsi *JavaScript* yang siap digunakan sehingga mempermudah dalam membuat kode *JavaScript*. Beberapa kemampuan yang dimiliki *jQuery* antara lain (Abdullah Rohi, 2015: 129):

1. memanipulasi elemen *HTML*
2. memanipulasi *CSS*
3. penanganan *event* pada *HTML*
4. efek-efek *JavaScript* dan animasi

2.3.8 MySQL dan SQL



Gambar 2.9 Logo MySQL

(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Data adalah bagian penting dari pemrograman *modern* sehingga keseluruhan bahasa program menyediakan fungsi untuk mengakses *database*. Standar utama untuk bahasa *database* adalah (*SQL*) *Structured Query Language* (Naista David, 2016: 4). *SQL* distandardisasi sebagai bahasa untuk menciptakan *database*, menyimpan informasi ke dalam *database*, dan mendapatkan kembali informasi darinya. Aplikasi khusus dan lingkungan pemrograman mengkhususkan diri untuk menginterpretasikan data *SQL*.

Seorang *programmer* akan mulai dengan menciptakan suatu struktur data di dalam *SQL* dan kemudian menulis suatu program dalam bahasa (*PHP*) untuk mengakses data tersebut. Program *PHP* kemudian bias memformulasikan permintaan atau memperbarui data tersebut, yang dilewatkan ke *interpreter SQL*.

Banyak *relational Database Management System (RDBMS)* yang tersedia, tetapi *MySQL* khususnya sangat cocok untuk berkerja sama dengan *PHP*. Bagaimanapun, konsep dasar *SQL* tetap sama apa pun jenis *database* yang anda

pergunakan. Secara umum, modifikasi dalam *Microsoft Access*, *Microsoft SQL Server*, dan *Oracle*, seperti halnya pada sejumlah paket *RDBMS* lainnya.

Ada sejumlah paket *RDBMS* yang tersedia. Program tersebut bervariasi dalam kemampuan, fleksibilitas, dan harga. Namun pada dasarnya, semua bekerja dengan cara yang sama. Dalam buku ini, akan digunakan *databaseMySQL*. Program ini sangat cocok berpasangan dengan *PHP* dengan beberapa pertimbangan. *MySQL* menggunakan suatu *format* standar *SQL* bahasa data yang terkenal.

2.3.9 Sublime Text



Gambar 2.10 Logo *Sublime text*

(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Sublime Text Editor adalah *editor* teks untuk berbagai bahasa pemrograman termasuk pemrograman *PHP*. *Sublime Text Editor* merupakan *editor text* lintas-*platform* dengan *Python application programming interface (API)*. *Sublime Text Editor* juga mendukung banyak bahasa pemrograman dan bahasa *markup*, dan fungsinya dapat ditambah dengan *plugin*, dan *Sublime Text Editor* tanpa lisensi perangkat lunak. *Sublime Text Editor* pertama kali dirilis pada tanggal 18 januari

2008, dan sekarang versi *Sublime Text Editor* sudah mencapai versi 3 yang dirilis pada tanggal 29 Januari 2013. *Sublime Text* mendukung *operation system* seperti *Linux*, *Mac OS X*, dan *windows*. Sangat Banyak fitur yang tersedia pada *Sublime Text Editor* diantaranya *minimap*, membuka *script* secara *side by side*, *bracket highlight* sehingga tidak bingung mencari pasangannya, kode *snippets*, *drag and drop direktori* ke *sidebar* terasa mirip dengan *Text Mate* untuk *Mac OS* (Abdullah. Rohi, 2015: 131)

2.3.10 StarUML



Gambar 2.11 Logo *StarUML*
(sumber: <http://staruml.io/>)

StarUML merupakan salah satu *CASE (Computer-Aided Software Engineering)* *tools* atau perangkat pembantu berbasis komputer untuk rekayasa perangkat lunak yang mendukung alur hidup perangkat lunak (*life cycle support*). *StarUML* termasuk ke dalam kelompok *upper CASE tools* yang mendukung perencanaan strategis dan pembangunan perangkat lunak (A.S. dan Shalahuddin, 2013: 122-123).

Terdapat 13 macam diagram dalam *UML 2.3* yang dibagi menjadi 3 kategori yaitu (A.S. dan Shalahuddin, 2013: 140-141):

1. *Structure diagrams*

Kategori ini terdiri dari kumpulan *diagram* yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Diagram *UML*

yang termasuk dalam kategori ini antara lain *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram*, dan *deployment diagram*.

2. *Behaviour diagrams*

Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Diagram *UML* yang termasuk dalam kategori ini antara lain *use case diagram*, *activity diagram*, dan *state machine diagram*.

3. *Interaction diagrams*

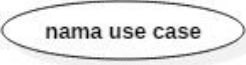


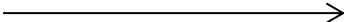
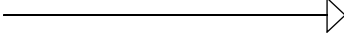

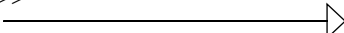
Kategori ini terdiri dari kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. Diagram *UML* yang termasuk dalam kategori ini antara lain *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction overview diagram*.

Menurut A.S. dan Shalahudin (2013: 18) *use case* dan *sequence diagram* merupakan bagian dari desain sistem. Dalam penelitian ini, diagram yang akan digunakan untuk desain sistem yaitu:

1. *Use case diagram*

Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu sistem atau lebih aktor dengan sistem informasi yang akan dibuat

Tabel 2.3 Simbol *Use Case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<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>
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i></p>
<p>asosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor</p>
<p>Ekstensi/<i>extend</i></p> <p><<extend>></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu. Arah panah mengarah pada <i>use case</i> yang ditambahkan.</p>
<p>generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum – khusus) antara 2 buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p>Menggunakan/<i>include/uses</i></p> <p><<include>></p>  <p><<uses>></p> 	<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 menjalankannya atau sebagai syarat dijalankannya <i>use case</i> ini. Arah panah mengarah pada <i>use case</i> yang ditambahkan</p>

(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

Use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Ada 2 hal utama yang terdapat pada *use case* yaitu aktor dan *use case*. Tabel diatas adalah simbol-simbol yang digunakan dalam *use case diagram* (A.S. dan Shalahuddin, 2013: 155).

2. Activity Diagram






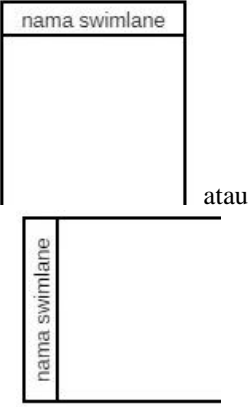
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

Diagram ini bersifat dinamis. *Diagram* ini adalah tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. *Diagram* ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar *subsistem*) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum. Struktur diagram ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat *diagram* ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan

Dibawah ini akan menjelaskan simbol-simbol *Activity Diagram* yaitu:

Tabel 2.4 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
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi





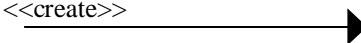
(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

3. Sequence Diagram

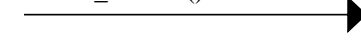
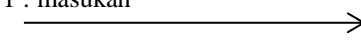
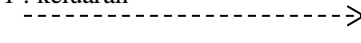
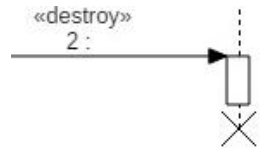
Diagram ini bersifat dinamis. *Diagram* sequence merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

Dibawah ini akan menjelaskan simbol-simbol *Sequence* Diagram yaitu:

Tabel 2.5 Simbol *Sequence* Diagram

Simbol	Deskripsi
Aktor/ <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Garis hidup/ <i>lifeline</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif
Pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat

Tabel 2.6 Lanjutan

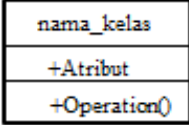
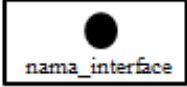
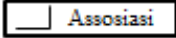
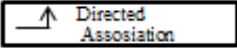
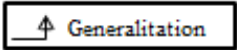
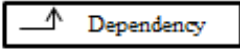
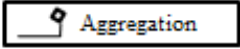
Simbol	Deskripsi
<p>pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
<p>Pesan tipe <i>send</i></p> <p>1 : masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya. Arah panah mengarah pada objek yang dituju
<p>pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu. Arah panah mengarah pada objek penerima
<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek lain. Arah panah mengarah pada objek yang diakhiri

(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

4. *Class Diagram*

Class diagram digunakan untuk melakukan visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe *diagram* yang paling banyak digunakan. *Class diagram* juga dapat memperlihatkan hubungan antar kelas di dalam model desain (*logical view*) dari suatu sistem. Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi serta relasi

Tabel 2.7 Tabel *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 berarah
Asosiasi Berarah (Direction Association) 	Relasi antar kelas dengan kelas yang satu digunakan kelas lain
Generalisasi (Generalitation) 	Relasi antar kelas dengan makna generalisasi antar makna
Ketergantungan(Dependency) 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi(Aggregation) 	Relasi antar kelas dengan makna semua bagian

(Sumber: Priyanto Hidayatulloh & Jauhari Khairul Kawistara, 2015)

2.4 Penelitian Terdahulu

Untuk mendukung teori yang berkaitan dengan penelitian, peneliti mencantumkan beberapa penelitian terdahulu di bidang sistem pakar.

Mujilahwati (2014) Diagnosa Penyakit tanaman hias Menggunakan Metode Certainty Factor Berbasis Web, ISSN No. 2085-0859 Vol : 6 No : 2 . Membuat suatu aplikasi yang berfungsi sebagai sistem alternatif pendukung keputusan dalam mendiagnosa tanaman hias. Dengan ada nya sistem yang di buat ini, diharapkan mampu membantu dan merubah sistem lama yang kurang tepat. Adapun manfaat sistem pakar penyakit tanaman hias ini sebagai berikut : 1. Memberikan pandangan serta langkah dalam sebuah sistem untuk mengetahui sejenis penyakit. 2. Memperkecil ada nya kesalahan dalam pengambilan keputusan tentang sejenis penyakit,

Yosi octavina, abdul fadlil (2014) Sistem pakar untuk mendiagnosa penyakit pada saluran pernafasan dan paru menggunakan *Certainty Factor* dengan issn : 2338-5197 Vol : 2 No : 2. Diperoleh fakta bahwa hasil pengembangan dan pembahasan maka dapat disimpulkan: 1. Dari sebuah perangkat lunak (*software*) baru tentang sistem pakar yang mampu sebagai pendukung unuk mengambil keputusan dengan memberikan solusi untuk membantu diagnosa pada saluran pernafasan dan paru-paru dengan jumlah penyakit 18 penyakit yang sudah di verifikasi ahli pakar (dokter spesialis penafasan paru) dan dapat sebagai media belajar bagi mahasiswa kedokteran dan digunakan di klinik umum khususnya di daerah. 2. Perangkat lunak yang

dihasilkan mampu mengidentifikasi (dokter spesifikasi pernafasan paru). Hasil dari sistem ini telah di uji dengan pihak dokter (spesialis pernafasan paru) yaitu dr. H. Yusrizal Djam'an shaleh sp.P di di Rumah Sakit PKU Muhammadiyah Yogyakarta.

Roni Pambuni, Sumarno (2015) Aplikasi Sistem Pakar Diagnosa Penyakit Kanker Menggunakan Metode *Cetainty Factor* dengan ISSN : 2339-028X. Diperoleh Fakta Bahwa 1. Semakin banyak gejala yang mendekati penyakit sehingga memperbesar kemungkinan seseorang terdeteksi terkena salah satu penyakit kanker 2. Besarnya nilai Densitas total total ditentukan oleh banyaknya kecocokan antara id gejala dan id penyakit, serta besarnya nilai *densitas* (bobot) tiap aturan pada kaidah dignosa. 3. Nilai *densitas* (bobot) mendekati satu, maka kepastiannya mendekati benar.

Ahmad Sahal, Sri Hartati (2013) Sistem Pakar Sebagai Alat Bantu Asuhan Keperawatan Menangani Penyakit Demam Berdarah Dengan Algoritma Rete Dan *Certainty Factor* dengan ISSN : 1907-2430 Vol : VIII No : 24 diperoleh fakta sistem pakar dengan inferensi *Forward Chaining*, Algoritma Rete, Dan *Certainty Factor* dapat di pergunakan untuk mendiagnosa tingkat resiko penyakit Deman Berdarah (DBD), dengan masukan berupa gejala gejala yang dimiliki pasien. Dari beberapa kasus yang diujicobakan diperoleh hasil diganosa yang sama antara perhitungan sistem dan perhitungan manual.

Jadiaman Parhusip (2012) Sistem Pakar Diagnosa Penyakit Jantung Menggunakan Metode *Certainty Factor* Berbasis Web dengan ISSN 1979-2328 diperoleh fakta bahwa sistem pakar diagnose penyakit jantung ini dibangun dengan

menggunakan metode pengembangan perangkat lunak *waterfall*. Dengan adanya fasilitas akuisisi pengetahuan (*knowledge acquisition facility*) yang menyediakan fasilitas dialog antara dengan sistem yang mana fasilitas akuisisi ini digunakan untuk memasukkan fakta-fakta dan kaidah-kaidah pengetahuan mengenai penyakit jantung sehingga sistem pakar ini dapat mendiagnosa seseorang yang menderita penyakit jantung apabila seorang dokter (pakar) tidak ada dengan mengakses secara online. Jadi sistem pakar ini dapat digunakan dan diakses masyarakat secara luas dan kapan saja.

2.5 Kerangka Pemikiran

Dari analisis pembahasan masalah di atas dapat kita lihat dalam kerangka berpikir berikut



Gambar 2.12 Kerangka Pemikiran

(Sumber: Data Penelitian, 2017)

Data-data yang dibutuhkan berkaitan dengan kerusakan mesin kasir dianalisis terlebih dahulu agar lebih sederhana atau mudah dilakukan proses pengolahan datanya. Data-datanya tersebut kemudian diolah menggunakan sistem pakar menggunakan metode *certainty factor*. Sistem pakar yang menggunakan metode

certainty factor ini menggunakan bahasa pemrograman *PHP* dan *database MySQL* yang dapat digunakan untuk mendeteksi kerusakan mesin kasir dan menghasilkan *output* (hasil deteksi).

BAB III

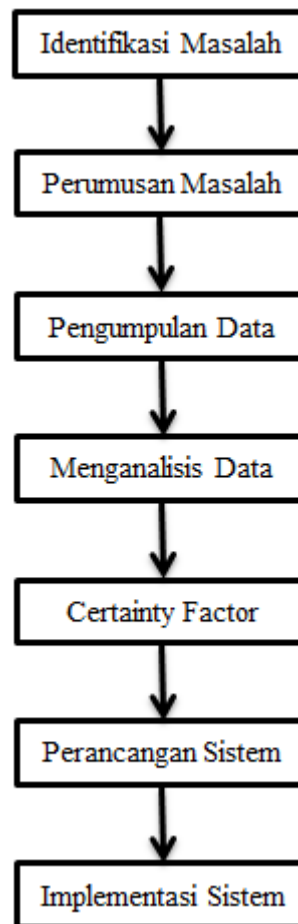
METODE PENELITIAN

Metode penelitian pada dasarnya merupakan cara ilmiah untuk menemukan data dengan tujuan dan kegunaan tertentu. Data yang telah diperoleh dari penelitian dapat digunakan untuk memahami, memecahkan dan menganalisis suatu masalah (Sugiyono, 2014: 2).

3.1 Desain Penelitian

Menurut Hasibuan (2007: 61) desain penelitian merupakan pedoman dalam melakukan proses penelitian diantaranya dalam menentukan *instrument* pengambilan data, penentuan sample, pengumpulan data serta analisis data.

Tahapan penelitian yang dilakukan pada analisis untuk menentukan apa penyebab terjadinya kerusakan mesin kasir pada Hypermart Tanjung Uncang Batam dengan menggunakan *Certainty Factor* ditunjuk pada gambar 3.1 dibawah ini.



Gambar 3.1 Desain Penelitian
(Sumber: Data Penelitian, 2017)

Berikut ini adalah penjelasan dari desain penelitian yang ada pada gambar di atas:

1. Identifikasi Masalah

Penelitian diawali dengan melakukan studi pendahuluan untuk mengidentifikasi permasalahan yang berkaitan dengan topik penelitian agar peneliti mendapatkan apa yang sesungguhnya menjadi masalah untuk dipecahkan. Pada tahap

mengidentifikasi masalah adalah untuk menunjukkan bahwa kurangnya wawasan karyawan untuk mendeteksi kerusakan mesin kasir.

2. Perumusan Masalah

Setelah mengidentifikasi masalah maka selanjutnya yang dilakukan adalah merumuskan masalah dimana setelah menentukan masalah atau variabel yang akan di teliti maka perlu adanya menganalisis variabel tersebut apakah layak untuk dilakukannya penelitian pada masalah tersebut.

3. Pengumpulan Data

Pengumpulan data dilakukan guna untuk mendapatkan rincian tentang variabel yang di ambil untuk di teliti supaya memperlengkap informasi yang di kumpulkan, dan dalam penelitian ini teknik pengumpulan data menggunakan wawancara sebagai alat untuk mendapatkan data yang diperlukan.

4. Menganalisis Data

Menganalisis data adalah dimana setelah melakukan pengumpulan data akan didatakannya data-data yang diperlukan, maka pada tahap menganalisis data adalah menelaah data-data yang penting atau penyusunan secara berurut mengenai data yang telah ditepkan agar supaya lebih mudah dipahami dan di ambil untuk di terapkan dalam sistem pakar.

5. *Certainty Factor*

Certainty Factor adalah metode untuk membuktikan apakah suatu fakta itu pasti ataukah tidak pasti yang berbentuk metric yang biasanya digunakan dalam sistem pakar

6. Perancangan Sistem

Perancangan system merupakan bagian dari merancang *web* dari sistem pakar yang menggunakan metode *certainty factor* agar dapat mendeteksi kerusakan mesin kasir sesuai dengan rule yang sudah ada.

7. Implementasi Sistem

Implementasi sistem merupakan tahap akhir dari kerangka kerja penelitian yaitu dimana sistem yang sudah di buat dan di rancang dapat di uji cara kerjanya.

3.2 Teknik Pengumpulan Data

Teknik pengumpulan data merupakan cara-cara yang digunakan peneliti dalam mendapatkan data-data yang berkaitan dengan pokok bahasan dalam rangka untuk mendukung penelitian yang sedang dilakukan. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah:

1. Wawancara

Penelitian ini mendapatkan data-data kerusakan dengan cara wawancara langsung pada *IT Staff* di Hypermart Tanjung Uncang. Dalam metode wawancara, alat bantu yang digunakan peneliti berupa alat perekam untuk merekam pembicaraan selama proses wawancara dilakukan. Pedoman wawancara yang dilakukan berupa garis-garis besar permasalahan yang akan ditanyakan yaitu hal-hal yang berkaitan dengan mesin kasir dan penyebab kerusakannya.

2. Studi Literatur

Peneliti melakukan studi literatur dengan mengumpulkan, membaca, dan memahami referensi teoritis yang berasal dari buku-buku teori, buku elektronik (*e-book*), jurnal-jurnal penelitian, dan sumber pustaka otentik lainnya yang berkaitan dengan penelitian. Studi literatur bertujuan untuk menemukan variabel yang akan diteliti, membedakan hal-hal yang sudah dilakukan dan menentukan hal yang perlu dilakukan, melakukan sintesa dan memperoleh perspektif baru, dan menentukan makna dan hubungan antar variabel (Sarwono, 2006: 47).

3.3 Operasional Variabel

Menurut Sugiono (2014 :38), variable penelitian adalah suatu atribut atau sifat atau nilai dari orang, objek atau kegiatan yang mempunyai variasi tertentu untuk ditetapkan oleh penelti untuk dipelajari kemudian ditarik kesimpulannya. Operasional variabel yang dibahas dalam penelitian ini adalah

Variabel *Input*

1. *Blue screen*
2. Tidak dapat mengeluarkan struk
3. Tidak dapat *scanning barcode* barang
4. *Off line*

Varibael *Output*

1. Keputusan apa yang menjadi kerusakan mesin kasir tersebut

3.4 Metode Analisis Data

Analisis hasil akan dilakukan dengan menggunakan metode *Certainty Factor* (CF). *Certainty Factor* adalah untuk mengakomodasi ketidakpastian seorang pakar yang diusulkan oleh Shortlife dan Buchanan pada tahun 1975. Seorang pakar (misalnya dokter) sering menganalisis informasi yang ada dengan ungkapan ketidakpastian, untuk mengakomodasi hal ini kita menggunakan *certainty factor* guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi. Konsep ini kemudian diformulasikan dalam rumus dasar sebagai berikut

$$CF[H,E]=MB[H,E] - MD[H,E] \quad \textbf{Rumus 3.1 Hasil Certainty Factor}$$

Keterangan : f

CF=*Certainty Factor* dalam hipotesa H akan dipengaruhi Oleh Fakta E.

MB[H,E]= *Measure of belief* (ukuran kepercayaan) terhadap hipotesa H, jika diberikan *evidence* E (antara 0 dan 1).

MD[H,E]= *Measure of disbelief* (ukuran kepercayaan) terhadap *evidence* H, jika diberikan *evidence* E (antara 0 dan 1).

H=*Hipotesa*.

E=*Evidence* (peristiwa atau waktu).

$$CF[H,E]1= CF[H] * CF[E] \quad \textbf{Rumus 3.2 Dasar Rumus Certainty Factor}$$

Keterangan:

CF[E]=*Certainty factor evidence* E yang di pengaruhi oleh *evidence* E.

$CF[H]$ = *Certainty factor* hipotesa dengan asumsi *evidence* diketahui dengan pasti, yaitu ketika $CF[E,r] = 1$.

$CF[H,E]$ = *Certainty factor* hipotesa yang dipengaruhi oleh *evidence* E diketahui dengan pasti.

Certainty factor untuk kaidah dengan kesimpulan yang serupa (*similarly concluded*)

rules):

$CF_{combine}CF[H,E]_{1,2} = CF[H,E]_1 + CF[H,E]_2 * [1 - CF[H,E]_1]$ **Rumus 3.3**

Combine Certainty Factor

$CF_{combine}CF[H,E]_{old,3} = CF[H,E]_{old} + CF[H,E]_3 * [1 - CF[H,E]_{old}]$ **Rumus 3.4**

Combine Certainty Factor Old

Pengecekan mesin kasir akan dilakukan pada aplikasi sistem pakar tersebut, kemudian aplikasi akan mengambil keputusan apa yang menjadi kerusakannya.

3.4.1 Desain Basis Pengetahuan

Sebelum melakukan desain basis pengetahuan, peneliti telah melakukan proses akuisi pengetahuan dengan mengumpulkan pengetahuan dan fakta dari sumber-sumber yang tersedia. Sumber pengetahuan dan fakta yang didapat berupa data-data yang berhubungan dengan kerusakan mesin kasir, penyebab kerusakannya dan juga solusi mengatasinya. Pengetahuan dan fakta tersebut ditampilkan dengan tabel dibawah ini.

Tabel 3.1 Tabel data kerusakan

Kode	Nama Kerusakan
K01	<i>Blue Screen</i>
K02	Tidak Dapat Mengeluarkan Struk
K03	Tidak Dapat Scan <i>Barcode</i> Barang
K04	Kasir Off Line

(Sumber: Data Penelitian, 2017)

Tabel 3.2 Tabel penyebab kerusakan dan solusi mesin kasir

Kerusakan	Penyebab Kerusakan	Solusi
<i>Blue Screen</i>	1.Memori Kasir Rusak	1.Dapat melakukan <i>ghost</i> system kasir dari kasir yang memiliki memory yang rusak 2.Melakukan <i>Safe Mode</i>
Tidak Dapat Mengeluarkan struk	1. Kabel USB pada dari pos ke printer rusak 2. Port pada POS rusak atau longgar 3.Sensor printer mati(menyatakan bahwa kertas dalam printer habis)	1.Melakukan pergantian kabel 2.Mengencangkan Port pada POS yang longgar 3.Memperbaiki posisi sensor printer dan membersihkan head printer 4. Melakukan pergantian kertas printer
Tidak Dapat Scan <i>Barcode</i> Barang	1. Kabel USB pada dari pos ke printer rusak 2. <i>Port</i> pada POS rusak atau longgar 3. <i>Scanner</i> belum di <i>update</i>	1.Melakukan pergantian kabel 2.Mengencangkan <i>Port</i> pada POS yang longgar 3.Melakukan <i>update</i> pada <i>scanner</i>

Tabel 3.2 Lanjutan

Kasir <i>Off Line</i>	1. Kabel LAN rusak 2. Server Down	1. Masuk ke dalam bios dan mengubah port LAN dari mode <i>disable</i> ke <i>enable</i> 2. Melakukan pengurutan kabel untuk melihat keberadaan kabel 3. Menjalankan aplikasi tomcat yang merupakan motor untuk menghubungkan antar server dengan kasir
-----------------------	--	---

(Sumber: Data Penelitian, 2017)

Tabel 3.3 Tabel penyebab kerusakan mesin kasir

Kode	Penyebab Kerusakan
P01	Sensor <i>Printer</i> Mati
P02	<i>Memory</i> Rusak
P03	<i>Port</i> Pada POS Rusak
P04	<i>Scanner</i> Tidak <i>Update</i>
P05	Server <i>Down</i>
P06	Ada IP <i>Adress</i> Yang Sama
P07	Kabel <i>LAN</i> Rusak
P08	Kabel USB dari POS ke printer Rusak

(Sumber: Data Penelitian, 2017)

3.4.2 Struktur Kontrol (Mesin inferensi)

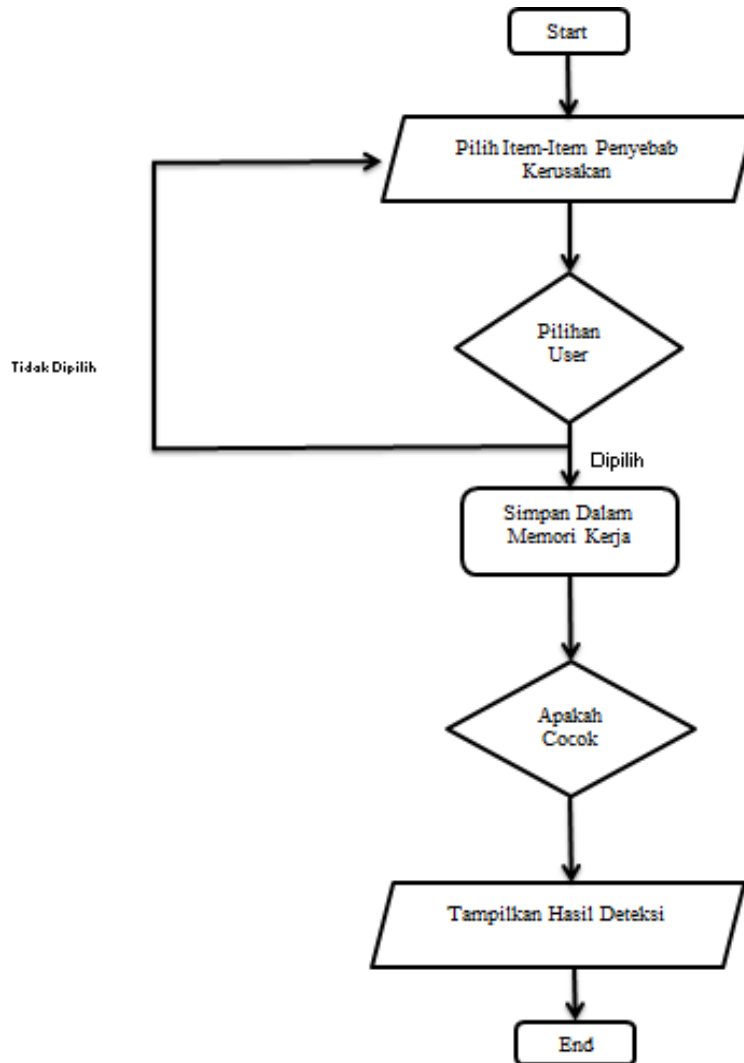
Mesin inferensi merupakan otak dari sistem pakar, berupa perangkat lunak yang melakukan tugas inferensi penalaran sistem pakar, biasa dikatakan sebagai sistem pemikir (*Thinking Machine*), pada prinsipnya mesin inferensi inilah yang akan mencari solusi dari suatu permasalahan. Konsep yang biasanya digunakan untuk mesin inferensi adalah runut balik(*top-down*), yaitu proses penalaran yang berawal dari tujuan yang kita inginkan, menelusuri fakta-fakta yang mendukung untuk mencapai tujuan. Selain itu dapat juga menggunakan runut maju(*bottom-up*), yaitu proses penalaran yang bermula dari kondisi yang diketahui menuju tujuan yang di inginkan. (Hartati & Iswanti, 2008)

Mesin inferensi dalam sistem pakar ini menggunakan metode penelusuran *certainty factor*. Langkah-langkah yang digunakan dalam proses penelusurannya adalah sebagai berikut:

1. Memilih item item penyebab kerusakan mesin kasir
2. Jika pengguna sudah memilih penyebab kerusakan akan melakukan langkah 3, dan jika tidak maka sistem akan melakukan langkah 4
3. Menyimpan penyebab kerusakan dalam memori kerja untuk mencari hasil deteksi dari penyebab kerusakan yang dipilih
4. Memeriksa apakah pengguna melakukan pemilihan item-item penyebab kerusakan

Menampilkan hasil deteksi

Berikut ini adalah gambar *flowchart* mesin inferensi yang digunakan dalam sistem pakar ini.



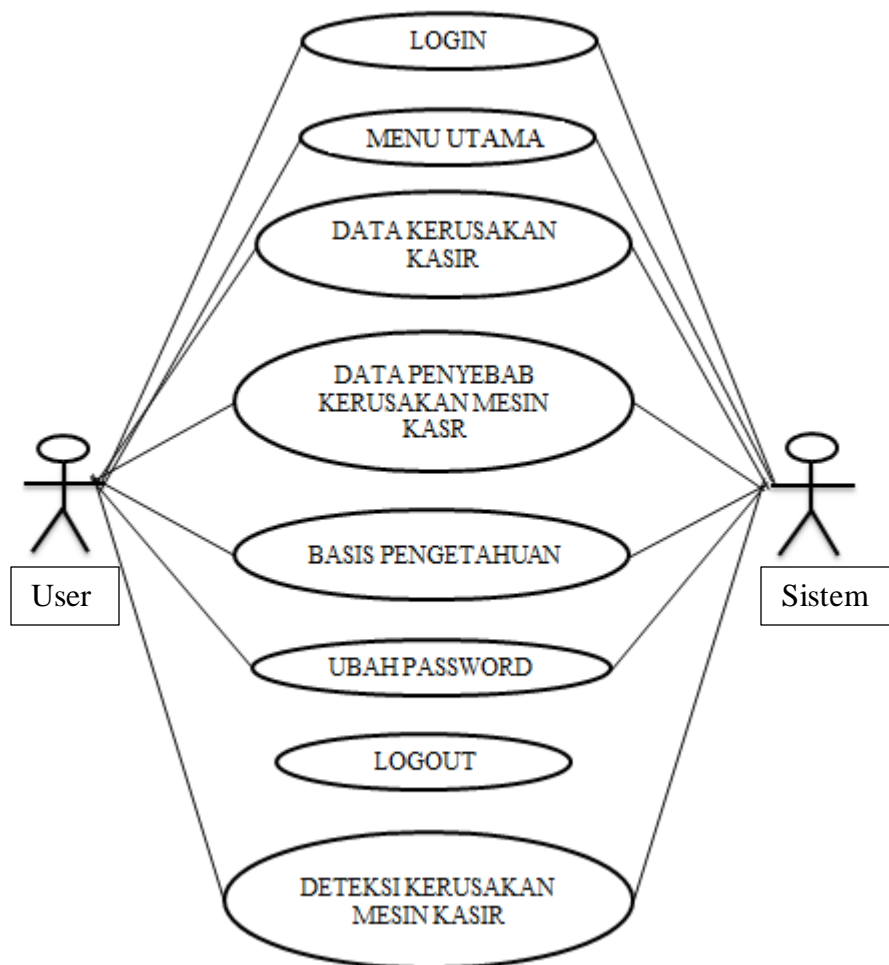
Gambar 3.2 *Flowchart*
(Sumber: Data Penelitian, 2017)

3.4.3 DesainUML (*Unified Modeling Language*)

Desain sistem pada penelitian ini menggunakan bahasa pemodelan *Unified Modelling Language (UML)* yang digambarkan dengan bantuan aplikasi *StarUML* versi 2.5.1. Diagram *UML* yang akan digunakan dalam penelitian ini antara lain:

1. *Use Case Diagram*

Aktor yang digunakan dalam sistem pakar ini terdiri dari 2 orang yaitu sistem dan *user*. Dalam sistem pakar ini yang berperan sebagai sistem adalah peneliti sendiri sedangkan *user* adalah *staff* baru yang ingin menangani permasalahan yang berkaitan dengan penyebab kerusakan mesin kasir di Hypermart Tanjung Uncang. *Use case* yang terdapat dalam sistem antara lain *login* , menu utama, data kerusakan kasir, data penyebab kerusakan kasir, basis pengetahuan, ubah *password*, deteksi kerusakan mesin kasir, *logout*, didalam data kerusakan kasir, data penyebab kerusakan dan basis pengetahuan *user* dapat menghapus dan merubahnya. *Use casediagram* yang dirancang untuk sistem pakar dalam penelitian ini dapat dilihat pada gambar dibawah ini.

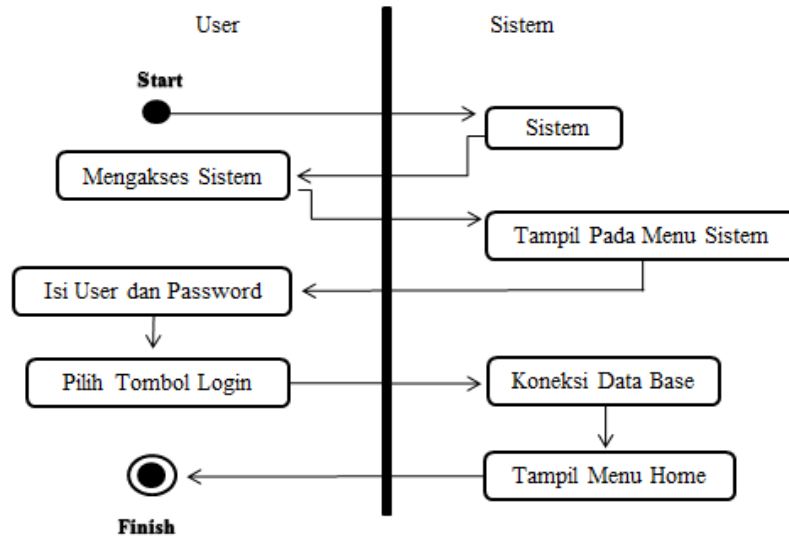


Gambar 3.3 Use Case Diagram
(Sumber: Data Penelitian, 2017)

2. *Activity diagram*

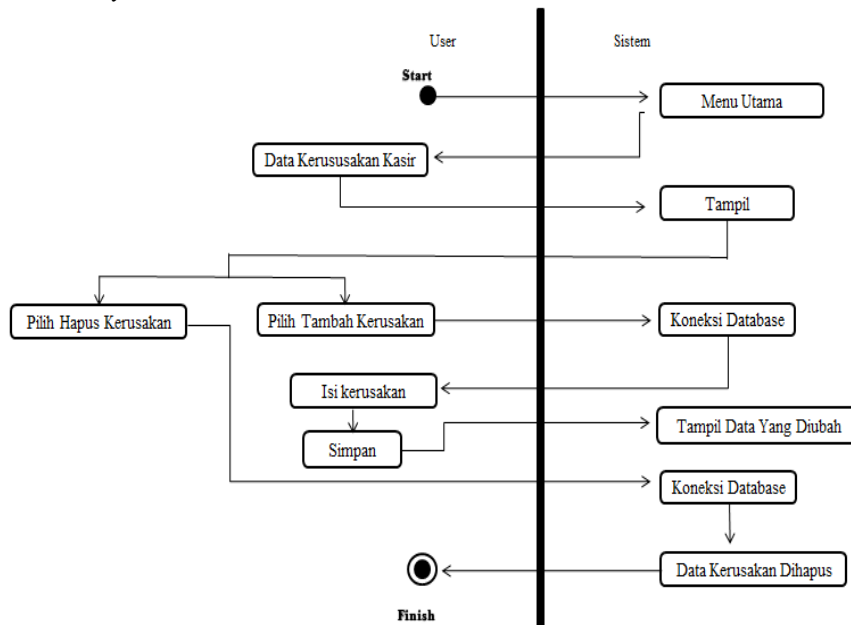
Activity diagram menggambarkan aktifitas yang dapat dilakukan oleh sistem atau menu yang ada pada perangkat lunak, bukan apa yang dilakukan oleh aktor (A.S. dan Shalahuddin, 2013: 161). *Activity diagram* yang dirancang untuk sistem pakar dalam penelitian ini akan ditunjukkan melalui gambar-gambar dibawah ini.

a. Activity login



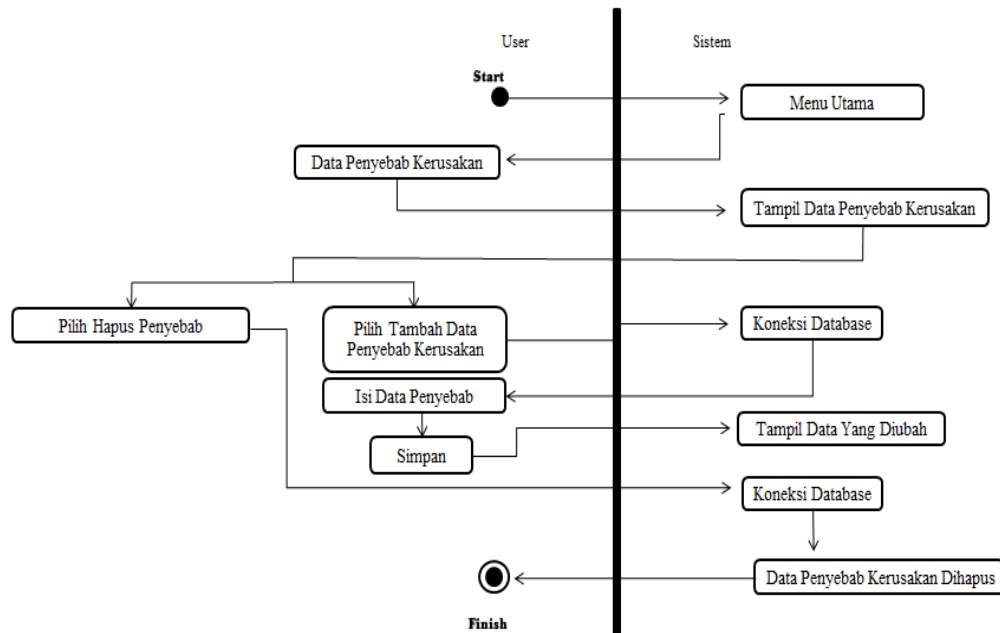
Gambar 3.4 Diagram activity user login
(Sumber: Data Penelitian, 2017)

b. Activity data kerusakan kasir



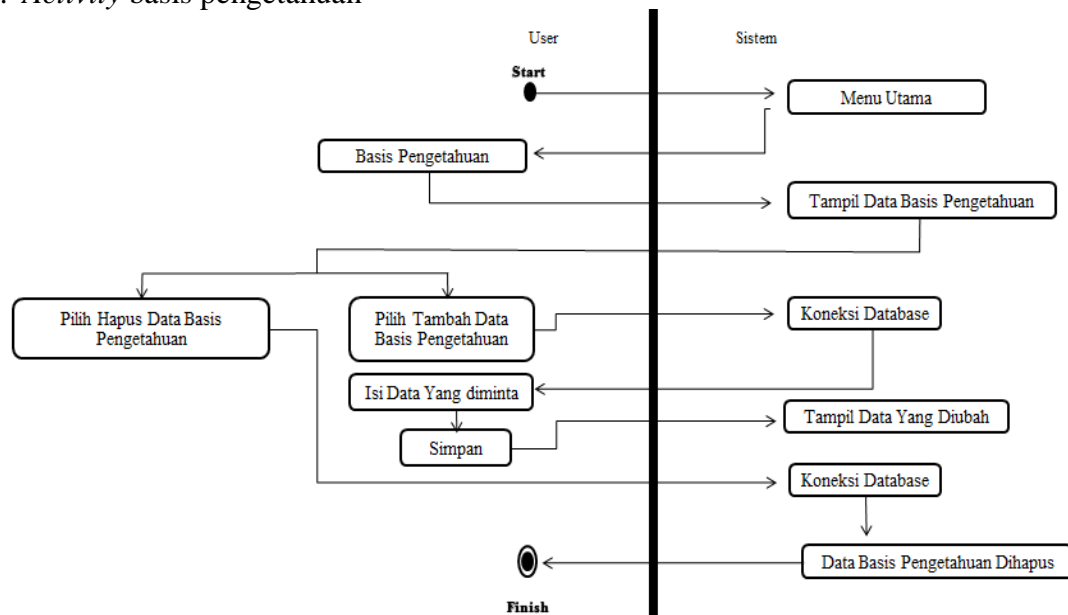
Gambar 3.5 Diagram activity data kerusakan
(Sumber: Data Penelitian, 2017)

c. *Activity* data penyebab kerusakan kasir

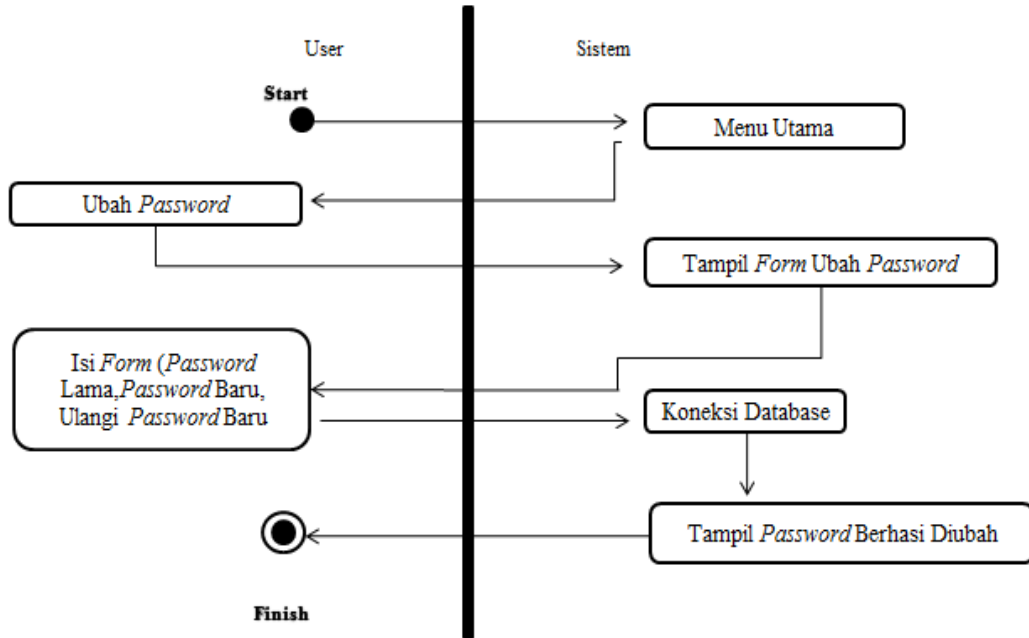


Gambar 3.6 *Diagram activity* data penyebab kerusakan
(Sumber: Data Penelitian, 2017)

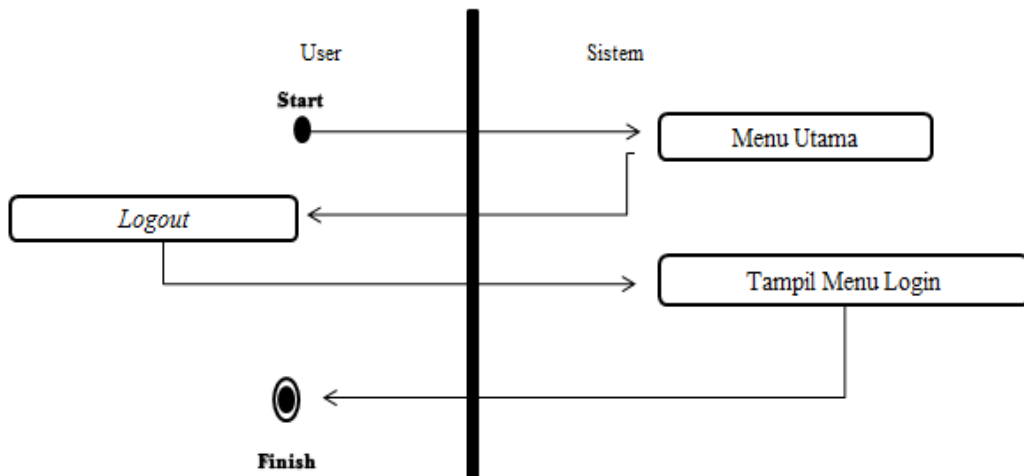
d. *Activity* basis pengetahuan



Gambar 3.7 *Diagram activity* data basis pengetahuan
(Sumber: Data Penelitian, 2017)

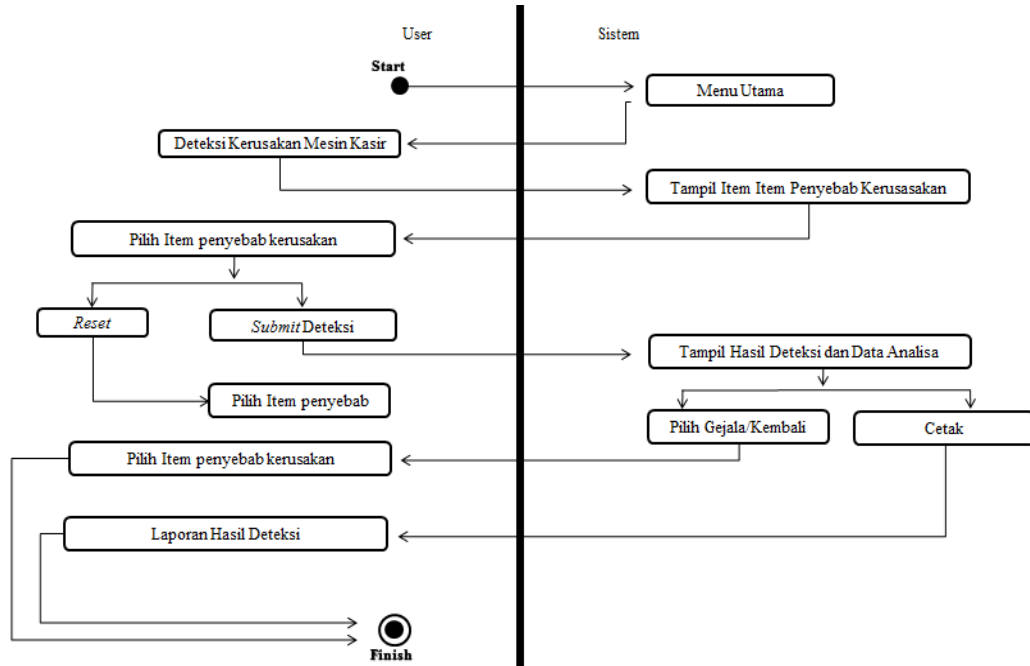
e. *Activity ubah password*

Gambar 3.8 Diagram *activity* lupa password
(Sumber: Data Penelitian, 2017)

f. *Activity logout*

Gambar 3.9 Diagram *activity* logout
(Sumber: Data Penelitian, 2017)

g. *Activity* deteksi kerusakan mesin kasir

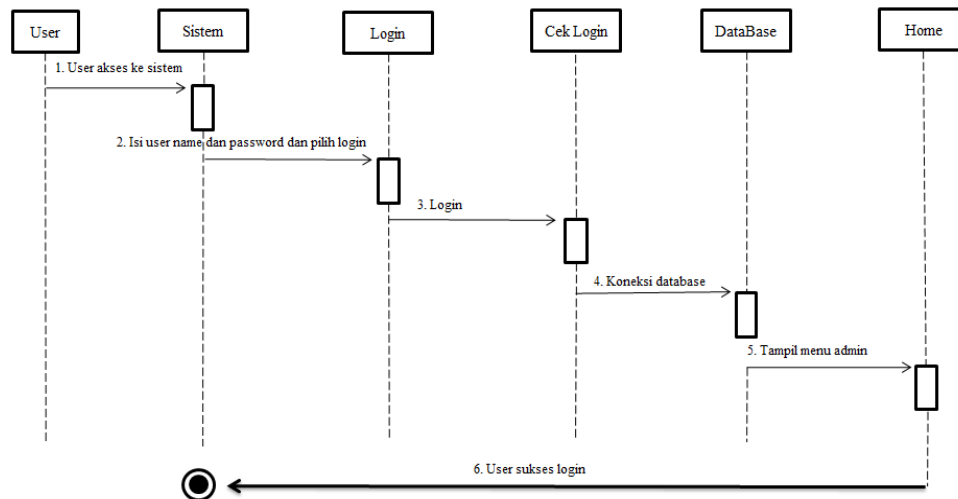


Gambar 3.10 *Diagram activity* deteksi kerusakan mesin kasir
(Sumber: Data Penelitian, 2017)

3. *Sequence diagram*

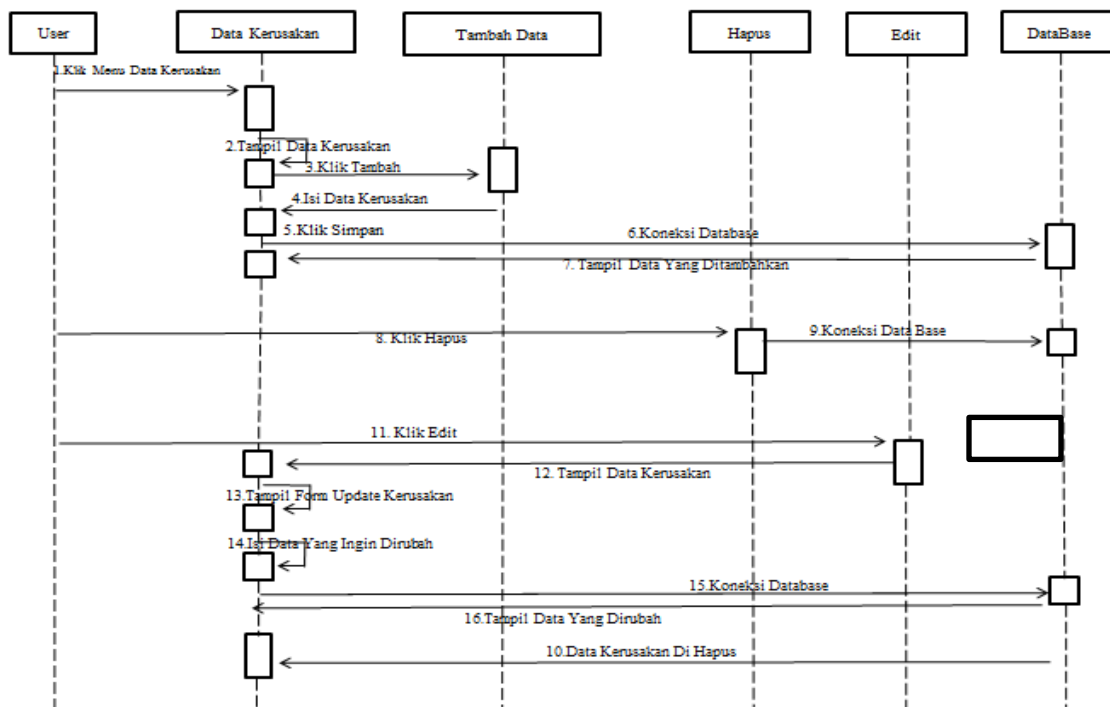
Sequence diagram merupakan diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek (A.S. dan Shalahuddin, 2013: 165). Berikut ini adalah gambar-gambar *sequence diagram* yang digunakan dalam sistem pakar pada penelitian ini.

a. *Sequence Diagram Log in*



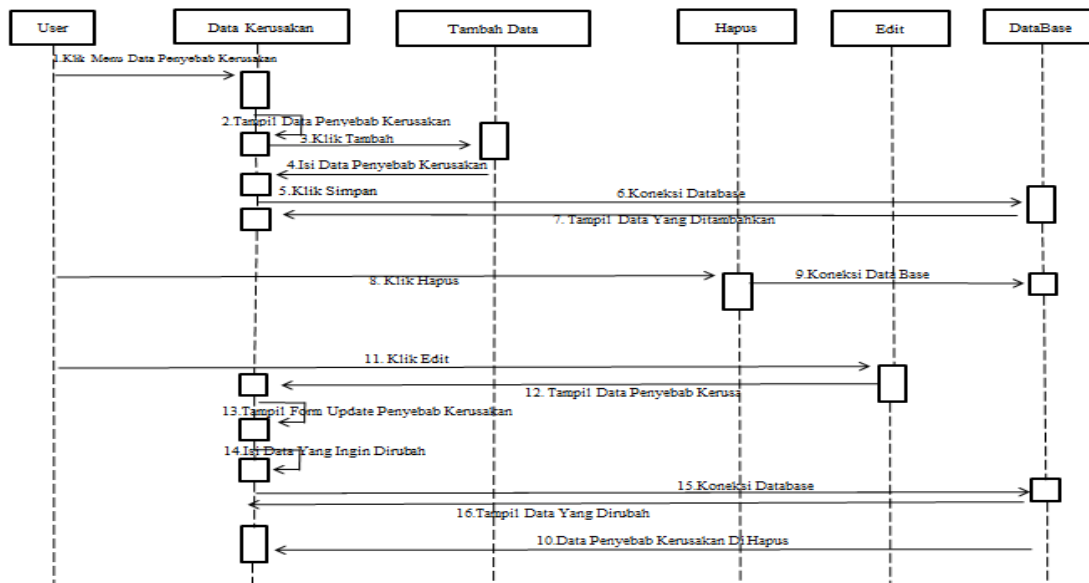
Gambar 3.11 *Sequence diagram login*
(Sumber: Data Penelitian, 2017)

b. *Sequence diagram data kerusakan kasir*



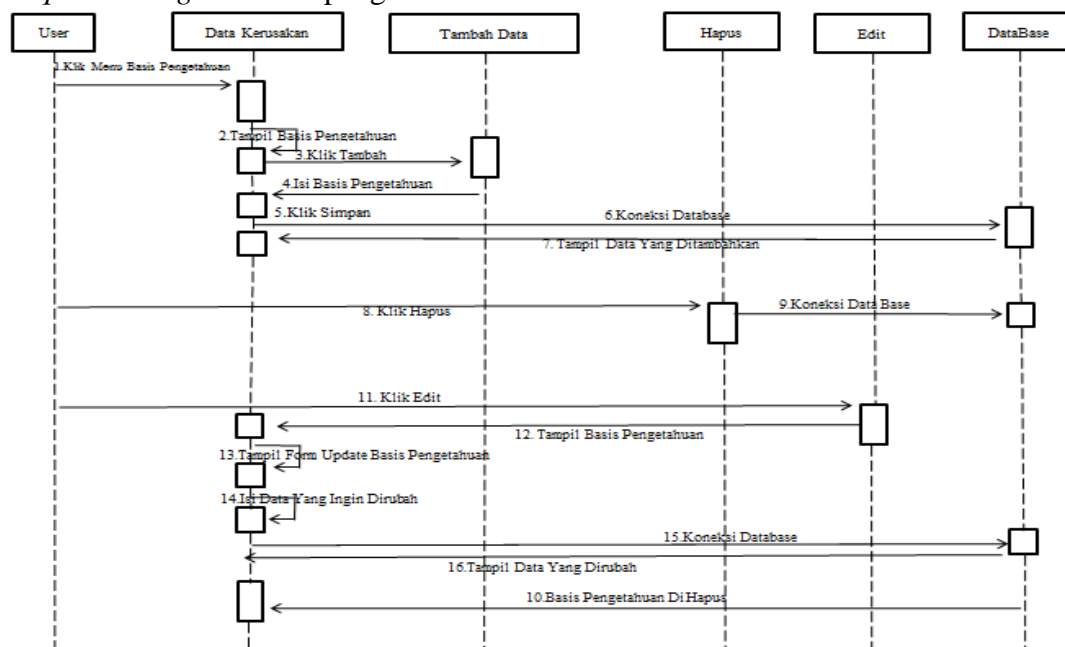
Gambar 3.12 *Sequence data kerusakan kasir*
(Sumber: Data Penelitian, 2017)

c. *Sequence diagram* data penyebab kerusakan kasir



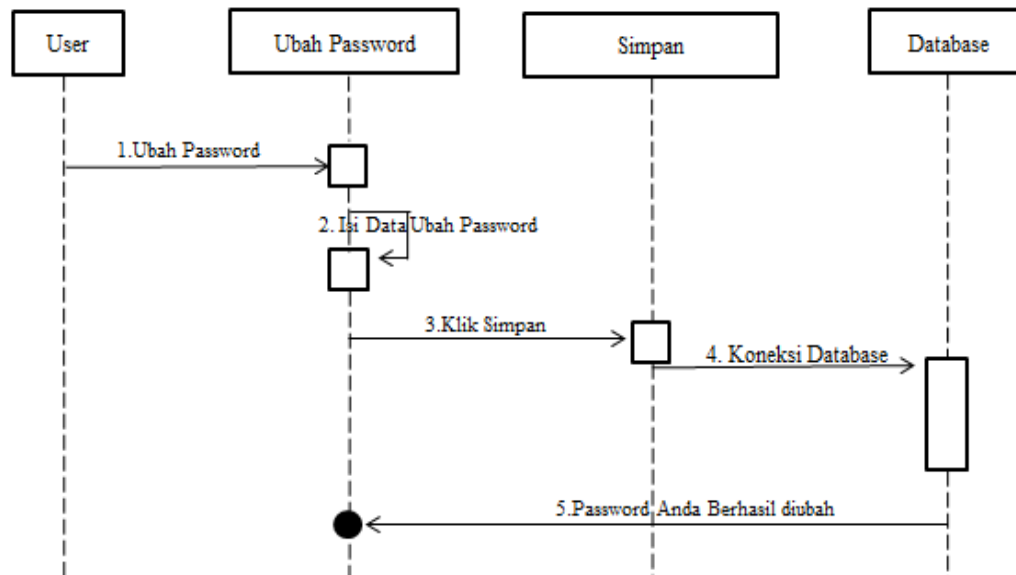
Gambar 3.13 Sequence data penyebab kerusakan kasir
(Sumber : Data Penelitian, 2017)

d. *Sequence diagram* basis pengetahuan



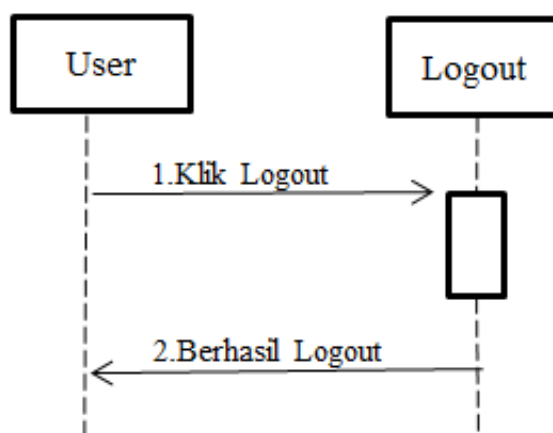
Gambar 3.14 Sequence diagram data basis pengetahuan
(Sumber : Data Penelitian, 2017)

e. *Sequence diagram ubah password*



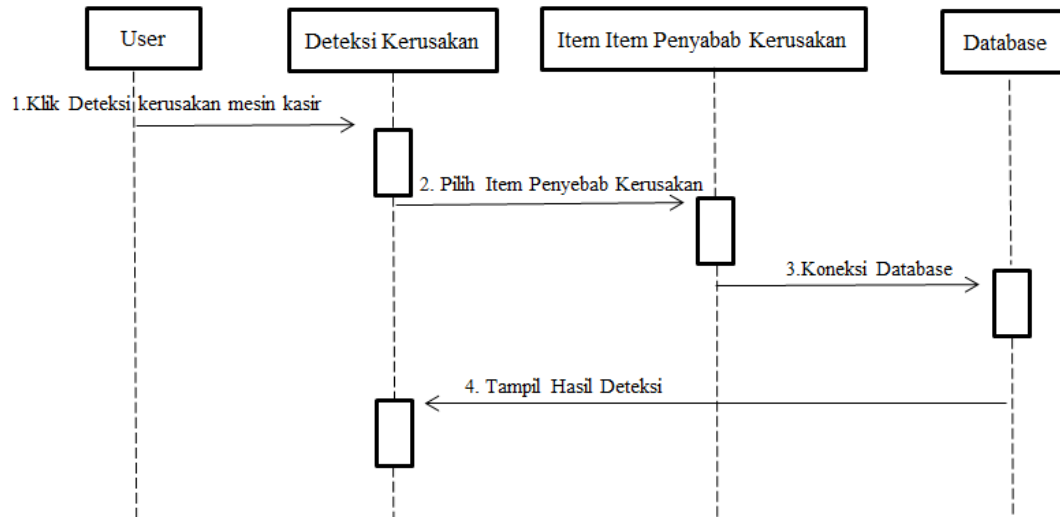
Gambar 3.15 *Sequence diagram ubah password*
(Sumber: Data Penelitian, 2017)

f. *Sequence diagram logout*



Gambar 3.16 *Sequence diagram logout*
(Sumber: Data Penelitian, 2017)

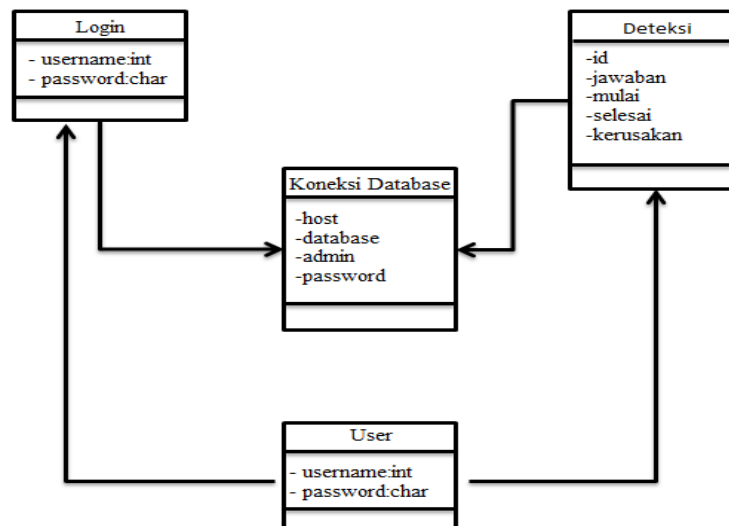
g. *Sequence diagram* deteksi kerusakan



Gambar 3.17 *Sequence diagram* deteksi kerusakan
(Sumber: Data Penelitian, 2017)

4. *class diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat atribut dan membangun sistem.



Gambar 3.18 *class diagram*
(Sumber: Data Penelitian, 2017)

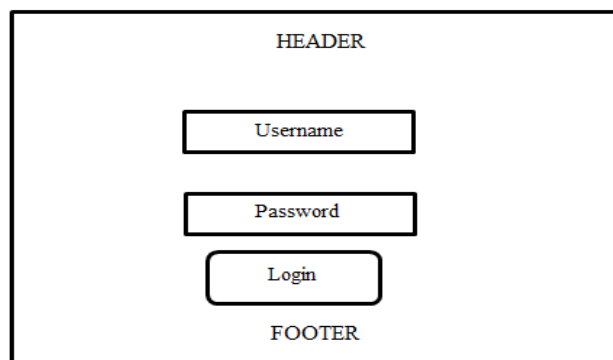
Penjelasan dari Gambar 3.17 *Class diagram*, aplikasi sistem pakar sebagai berikut:

1. *Login* adalah kelas proses yang diambil dari pendefinisian *use case login* untuk pintu masuk *user* ke sistem aplikasi pakar.
2. Koneksi *database* adalah kelas utilitas untuk koneksi ke *database*.
3. *User* adalah kelas proses yang diambil dari pendefinisian *use case* untuk aktor yang menggunakan aplikasi sistem pakar
4. Deteksi adalah kelas proses yang diambil dari pendefinisian *use case* untuk mengelola deteksi yang didalamnya menangani proses kerusakan

3.4.4 Prototype

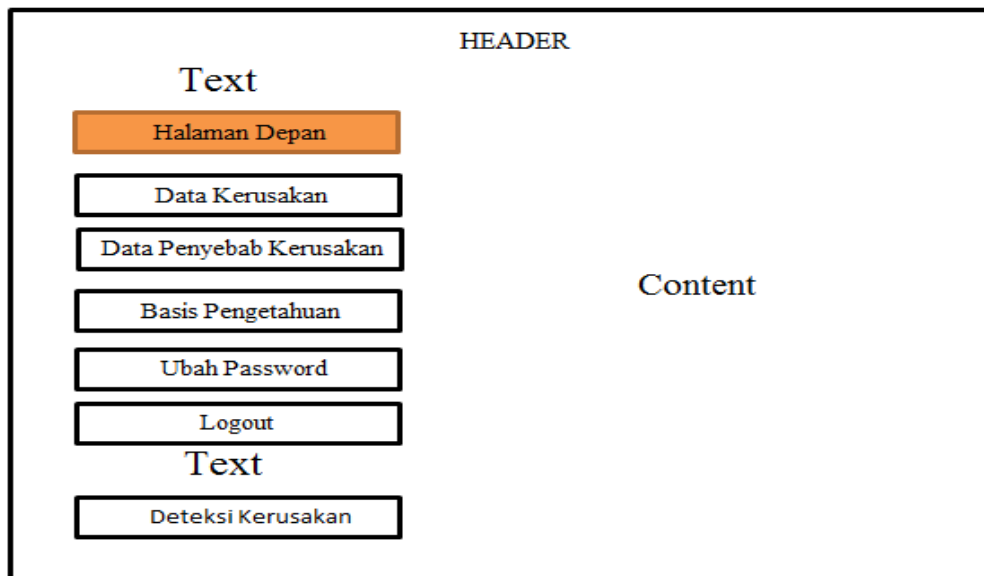
Berikut ini adalah desain tampilan sistem pakar untuk mendeteksi kerusakan mesin kasir.

1. Tampilan halaman *web*



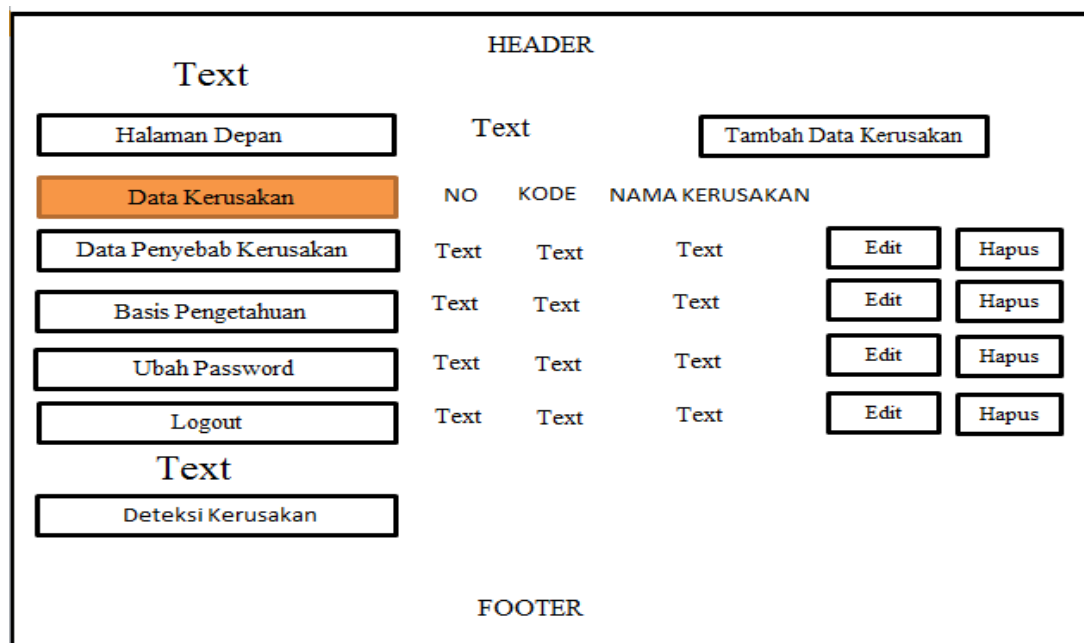
Gambar 3.19 Tampilan halaman *web*
(Sumber: Data Penelitian, 2017)

2. Tampilan halaman utama



Gambar 3.20 Tampilan halaman utama
(Sumber: Data Penelitian, 2017)

3. Tampilan halaman data kerusakan



Gambar 3.21 Tampilan halaman kerusakan
(Sumber: Data Penelitian, 2017)

3.5 Lokasi dan Jadwal Penelitian

3.5.1 Lokasi

Penelitian ini dilakukan bertempat di Hypermart Tanjung Uncang yang beralamat di Kompleks Tunas Regency, Jln Brigjen Katamso, Sungai Binti, Sagulung, Kota Batam. Alasan peneliti memilih instansi ini sebagai lokasi penelitian adalah:

1. Efisiensi biaya dan waktu
2. Mudah untuk mendapatkan data
3. Semua aspek mendukung

3.5.2 Jadwal Penelitian

Setiap rancangan penelitian perlu dilengkapi dengan jadwal penelitian yang dilaksanakan yang berisi jadwal kegiatan apa saja yang akan dilakukan selama penelitian (Sugiyono, 2014: 286). Berikut ini adalah tabel jadwal kegiatan yang dilakukan selama penelitian berlangsung.

Tabel 3.4 Jadwal Penelitian

No	Kegiatan	Tahun 2016/2017																
		Okt '16				Nov '16				Des '16				Jan '17				Feb '17
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1
1	Pengajuan Judul	■	■															
2	Penyusunan Bab I		■	■	■													
3	Penyusunan Bab II				■	■	■	■										
4	Penyusunan Bab III						■	■	■	■								
5	Penyusunan Bab IV										■	■	■	■	■	■		
6	Penyusunan Bab V, Daftar Pustaka, Lampiran																■	■

(Sumber: Data Penelitian, 2017)