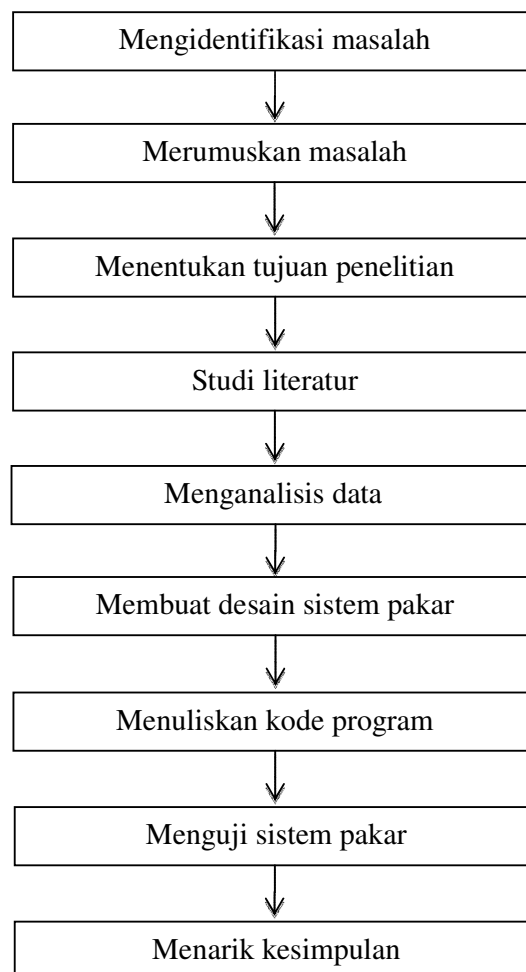


BAB III METODE PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan desain penelitian dengan beberapa tahap proses penelitian seperti yang terlihat pada gambar 3.1 di bawah ini.



Gambar 3.1 Desain Penelitian
(Sumber: Data Penelitian, 2016)

Dalam penelitian ini, penulis menggunakan jenis penelitian kualitatif. Menurut Sugiyono (2014: 9) penelitian kualitatif adalah metode penelitian yang berlandaskan pada filsafat *postpositivisme*, digunakan untuk meneliti pada kondisi objek yang alamiah. Berikut ini adalah penjelasan dari desain penelitian yang ditunjukkan pada gambar 3.1:

1. Mengidentifikasi masalah

Penelitian ini diawali dengan melakukan studi pendahuluan untuk mengidentifikasi permasalahan yang terjadi di lapangan agar peneliti mendapatkan apa yang sesungguhnya menjadi masalah untuk diselesaikan.

2. Merumuskan masalah

Pada tahap ini, peneliti merumuskan masalah secara lebih spesifik agar masalah tersebut dapat diselesaikan dengan baik melalui penelitian.

3. Menentukan tujuan penelitian

Peneliti menentukan tujuan penelitian yaitu mengimplementasikan sistem pakar berbasis *web* untuk mendeteksi kesalahan pada proses pengujian IC menggunakan metode *forward chaining*.

4. Studi literatur

Pada tahap ini dipelajari literatur yang berhubungan dengan topik penelitian diantaranya sistem pakar, *Integrated Circuit (IC)*, bahasa pemrograman *web*, *database* dan perangkat lunak pendukung. Sumber pengetahuan berasal dari buku, jurnal penelitian, internet dan sumber pustaka otentik lainnya.

5. Analisis

Pada tahap ini peneliti melakukan analisis mengenai kebutuhan apa saja yang diperlukan untuk membuat sistem pakar. Kegiatan yang termasuk pada tahap ini yaitu: melakukan pengumpulan data (wawancara, observasi dan dokumentasi), mengolah data yang diperoleh dengan menggunakan metode *forward chaining* sehingga menghasilkan basis pengetahuan (membuat tabel-tabel fakta, tabel keputusan, aturan sistem pakar, pohon keputusan dan alur kerja mesin inferensi).

6. Membuat desain sistem pakar

Pada tahap ini peneliti memodelkan sistem pakar ke dalam diagram UML menggunakan *software* StarUML. Pada penelitian ini terdapat 4 jenis UML yang digunakan yaitu *usecase diagram*, *activity diagram*, *sequence diagram* dan *class diagram*. Selain itu pada tahap ini juga peneliti membuat desain *database* dan desain antarmuka.

7. Menuliskan kode program

Pada tahap ini sistem pakar yang sebelumnya telah didesain akan diterapkan pada bahasa pemrograman *web*. Sistem pakar ini dibuat dengan menggunakan bahasa pemrograman PHP dan *database* MySQL. *Software* yang digunakan pada tahap ini adalah Sublime Text, XAMPP dan Mozilla Firefox.

8. Menguji sistem pakar

Setelah sistem pakar selesai dibuat maka dilakukan pengujian dengan metode *black-box testing* untuk mengetahui apakah sistem pakar tersebut telah bekerja dengan benar dan sesuai dengan harapan. Sistem juga diuji dengan

membandingkan hasil deteksi pakar dengan hasil deteksi sistem untuk melihat apakah sistem telah berfungsi dengan baik.

9. Menarik kesimpulan

Tahap akhir pada penelitian ini adalah penarikan kesimpulan berupa ringkasan hasil penelitian yang akan menjawab secara ringkas rumusan masalah yang telah ditetapkan sebelumnya. Selain itu peneliti memberikan saran agar penelitian ini bisa lebih disempurnakan lagi.

3.2 Teknik Pengumpulan Data

Sugiyono (2014: 224) mengungkapkan bahwa teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data. Tanpa mengetahui teknik pengumpulan data, maka peneliti tidak akan mendapatkan data yang memenuhi standar data yang ditetapkan. Teknik pengumpulan data yang digunakan dalam penelitian ini adalah:

1. Wawancara

Untuk mendapatkan data-data yang diperlukan, peneliti melakukan wawancara langsung dengan Bapak Erijal Hadi yang bekerja sebagai *test engineer* di PT Unisem. Pedoman wawancara yang digunakan berupa garis-garis besar permasalahan yang berkaitan dengan proses tes korelasi pada mesin *tester*, penyebab dan solusi-solusi yang biasa dilakukan untuk mengatasinya.

2. Observasi

Peneliti terlibat langsung dengan teknisi dalam melakukan proses tes korelasi pada mesin *tester* dimulai dari menyiapkan *loadboard*, unit korelasi, memilih *test program* dan menjalankan prosedur tes korelasi. Dalam hal ini, penulis melakukan observasi berperan serta (Sugiyono, 2014: 145).

3. Dokumentasi

Peneliti melakukan studi literatur dengan mengumpulkan, membaca, dan memahami referensi teoritis yang berasal dari berbagai buku teori, jurnal penelitian dan *Standart Operating Procedure (SOP)* yang berlaku di PT Unisem.

3.3 Operasional Variabel

Variabel yang diolah dalam penelitian ini adalah kesalahan pada proses pengujian *Integrated Circuit (IC)*. Kesalahan pada proses pengujian IC dapat diidentifikasi sejak awal yaitu pada saat proses tes korelasi pada mesin *tester*. Terdapat 3 faktor penting pada proses tes korelasi yang dapat mengakibatkan unit korelasi yang seharusnya *good* menjadi *reject* saat di tes pada mesin *tester*. Ketiga faktor tersebut adalah *device name*, *fail category* dan *fail name*. *Fail category* dan *fail name* dapat diketahui dari tampilan pada layar monitor di mesin *tester*. Ketiga faktor tersebut menjadi indikator dari kesalahan pada proses tes korelasi. Dari ketiga indikator tersebut dapat diketahui penyebab kesalahan dan solusi untuk mengatasinya. Hubungan antara variabel, penyebab dan indikator disajikan pada tabel 3.1.

Tabel 3.1 Variabel, Penyebab dan Indikator Penelitian

Variabel	Penyebab	Indikator
Kesalahan pada proses pengujian IC	1. <i>Kerusakan pada set-up hardware (loadboard)</i> 2. <i>Test program</i>	1. <i>Device name</i> 2. <i>Fail category</i> 3. <i>Fail name</i>

Sumber: Data Penelitian (2016)

3.3.1 Analisis Data

Dalam melakukan analisis data, penulis melakukan proses akuisisi pengetahuan dengan mengumpulkan pengetahuan dan fakta-fakta melalui wawancara dengan *test engineer* PT Unisem, observasi dan dokumentasi. Dari proses akuisisi pengetahuan tersebut, penulis memperoleh data-data yang berhubungan dengan kesalahan pada proses pengujian IC dan solusi untuk mengatasinya. Pengetahuan dan fakta tersebut ditampilkan dalam tabel-tabel fakta yaitu tabel *device name* (tabel 3.2), tabel *fail category* (tabel 3.3), tabel *fail name* (tabel 3.4) serta tabel penyebab dan solusi (tabel 3.5).

Tabel 3.2 Data *Device Name*

Kode	<i>Device Name</i>
DEV1	Z88C0020PSG-CC
DEV2	Z16C3220VSG00TR-CC
DEV3	Z84C1506FEG-KA
DEV4	Z8523L10VEG-HB
DEV5	Z16C3010VEG-KC
DEV6	Z84C0006AEG-AN

Tabel 3.2 Lanjutan

Kode	<i>Device Name</i>
DEV7	Z86E83VEG-HA
DEV8	Z84C0006VEG-HB
DEV9	Z86E3016SSG-DG
DEV10	Z86E6316VSG-AN

Sumber: Data Penelitian (2016)

Tabel 3.3 Data *Fail Category*

Kode	<i>Fail Category</i>
FC2	<i>Bin-2</i>
FC3	<i>Bin-3</i>
FC4	<i>Bin-4</i>
FC5	<i>Bin-5</i>
FC6	<i>Bin-6</i>
FC7	<i>Bin-7</i>
FC8	<i>Bin-8</i>

Sumber: Data Penelitian (2016)

Tabel 3.4 Data *Fail Name*

Kode	<i>Fail Name</i>
FN2	<i>ROM fail</i>
FN3	<i>VBO fail</i>
FN4	<i>ICC fail</i>
FN5	<i>gross fail</i>
FN6	<i>functional fail</i>
FN7	<i>DC fail</i>
FN8	<i>continuity fail</i>

Sumber: Data Penelitian (2016)

Tabel 3.5 Data Penyebab dan Solusi

Kode	Penyebab	Solusi
PYB1	<i>Failure</i> ini disebabkan oleh <i>test program</i> yang perlu dimodifikasi	Gunakan <i>test program</i> yang telah dimodifikasi xxx_MHZ20_LOWV.EXE yang ada di folder 85"TEST"
PYB2	<i>Failure</i> ini disebabkan oleh <i>delay PARAM[1]</i> pada <i>test program</i> terlalu kecil nilainya	Tambahkan nilai <i>PARAM[1]=100</i> pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 1 lalu 100
PYB3	<i>Failure</i> ini disebabkan oleh <i>delay PARAM[3]</i> pada <i>test program</i> terlalu kecil nilainya	Tambahkan nilai <i>PARAM[3]=100</i> pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 3 lalu 100
PYB4	<i>Failure</i> ini disebabkan oleh <i>delay PARAM[6]</i> pada <i>test program</i> terlalu kecil nilainya	Tambahkan nilai <i>PARAM[6]=10</i> pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 6 lalu 10
PYB5	<i>Failure</i> ini disebabkan oleh kondisi <i>loadboard</i> kurang baik, seperti koneksi kabel yang buruk atau <i>loadboard</i> tidak dilengkapi dengan kapasitor	Pastikan sambungan kabel pada <i>loadboard</i> dalam keadaan baik, pastikan terdapat kapasitor sebagai <i>filter</i> tegangan antara positif (VCC) dan negatif (GND), jika masih tidak bisa <i>pass</i> , ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-DB-015B
PYB6	<i>Failure</i> ini disebabkan oleh <i>delay PARAM[3]</i> pada <i>test program</i> terlalu kecil nilainya	Tambahkan nilai <i>PARAM[3]=50</i> pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 3 lalu 50
PYB7	<i>Failure</i> ini disebabkan oleh kondisi <i>receptacle loadboard</i> kurang baik sehingga koneksi <i>contact socket</i> ke <i>loadboard</i> kurang baik	Ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-DB-052C
PYB8	<i>Failure</i> ini disebabkan oleh kondisi dari <i>contact finger</i> yang kurang baik seperti kotor, <i>misalign</i> atau patah.	Pastikan <i>contact finger</i> dalam kondisi yang baik, bersih, tidak ada yang patah dan tidak <i>misalign</i> . Jika perlu ganti <i>contact finger</i> dengan yang baru
PYB9	<i>Failure</i> ini disebabkan oleh kondisi <i>loadboard</i> kurang bagus, koneksi nya tidak stabil	Ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-AD-010B

Tabel 3.5 Lanjutan

Kode	Penyebab	Solusi
PYB10	<i>Failure</i> ini disebabkan oleh kondisi <i>loadboard</i> kurang bagus	Ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-DB-099A

Sumber: Data Penelitian (2016)

3.3.2 Desain Basis Pengetahuan

Dari tabel-tabel fakta di atas, selanjutnya dibentuk tabel-tabel solusi yang merupakan hubungan antara data-data *device name*, *fail category*, *fail name* dan penyebab yang telah diberi kode sebelumnya. Hubungan antara data-data tersebut disusun berdasarkan sumber pengetahuan dan fakta yang telah diperoleh dari pakar yaitu *test engineer* PT Unisem. Tabel-tabel solusi ini disusun berdasarkan kesalahan yang terjadi pada masing-masing *device name*. Tabel-tabel solusi ini berguna untuk menyusun aturan (*rule*) yang akan diterapkan dalam sistem pakar. Susunan data solusi dapat dilihat pada tabel-tabel dibawah ini.

Tabel 3.6 Solusi Untuk *Device* Z88C0020PSG-CC

	Kode	Keterangan
<i>Device name</i>	DEV1	Z88C0020PSG-CC
<i>Fail category</i>	FC6	<i>Bin-6</i>
<i>Fail name</i>	FN6	<i>functional fail</i>
Penyebab	PYB1	<i>Failure</i> ini disebabkan oleh <i>test program</i> yang perlu dimodifikasi
Solusi		Gunakan <i>test program</i> yang telah dimodifikasi yaitu xxx_MHZ20_LOWV.EXE yang ada di folder 85"TEST"

Sumber: Data Penelitian (2016)

Tabel 3.7 Solusi Untuk *Device* Z16C3220VSG00TR-CC

	Kode	Keterangan
<i>Device name</i>	DEV2	Z16C3220VSG00TR-CC
<i>Fail category</i>	FC4	<i>Bin-4</i>
<i>Fail name</i>	FN4	<i>ICC fail</i>
Penyebab	PYB2	<i>Failure</i> ini disebabkan oleh <i>delay</i> PARAM[1] pada <i>test program</i> terlalu kecil nilainya
Solusi		Tambahkan nilai PARAM[1]=100 pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 1 lalu 100

Sumber: Data Penelitian (2016)

Tabel 3.8 Solusi Untuk *Device* Z84C1506FEG-KA

	Kode	Keterangan
<i>Device name</i>	DEV3	Z84C1506FEG-KA
<i>Fail category</i>	FC4	<i>Bin-4</i>
<i>Fail name</i>	FN4	<i>ICC fail</i>
Penyebab	PYB3	<i>Failure</i> ini disebabkan oleh <i>delay</i> PARAM[3] pada <i>test program</i> terlalu kecil nilainya
Solusi		Tambahkan nilai PARAM[3]=100 pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 3 lalu 100

Sumber: Data Penelitian (2016)

Tabel 3.9 Solusi Untuk *Device* Z8523L10VEG-HB

	Kode	Keterangan
<i>Device name</i>	DEV4	Z8523L10VEG-HB
<i>Fail category</i>	FC6	<i>Bin-6</i>
<i>Fail name</i>	FN6	<i>functional fail</i>
Penyebab	PYB4	<i>Failure</i> ini disebabkan oleh <i>delay</i> PARAM[6] pada <i>test program</i> terlalu kecil nilainya
Solusi		Tambahkan nilai PARAM[6]=10 pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 6 lalu 10

Sumber: Data Penelitian (2016)

Tabel 3.10 Solusi Untuk *Device* Z16C3010VEG-KC

	Kode	Keterangan
<i>Device name</i>	DEV5	Z16C3010VEG-KC
<i>Fail category</i>	FC5	<i>Bin-5</i>
<i>Fail name</i>	FN5	<i>Gross fail</i>
Penyebab	PYB5	<i>Failure</i> ini disebabkan oleh kondisi <i>loadboard</i> kurang baik, seperti koneksi kabel yang buruk atau <i>loadboard</i> tidak dilengkapi dengan kapasitor
Solusi		Pastikan sambungan kabel pada <i>loadboard</i> dalam keadaan baik, pastikan terdapat kapasitor sebagai <i>filter</i> tegangan antara positif (VCC) dan negatif (GND), jika masih tidak bisa <i>pass</i> , ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-DB-015B

Sumber: Data Penelitian (2016)

Tabel 3.11 Solusi Untuk *Device* Z84C0006AEG-AN

	Kode	Keterangan
<i>Device name</i>	DEV6	Z84C0006AEG-AN
<i>Fail category</i>	FC7	<i>Bin-7</i>
<i>Fail name</i>	FN7	<i>DC fail</i>
Penyebab	PYB6	<i>Failure</i> ini disebabkan oleh <i>delay</i> PARAM[3] pada <i>test program</i> terlalu kecil nilainya
Solusi		Tambahkan nilai PARAM[3]=50 pada <i>test program</i> . Caranya tekan CTRL+A lalu D lalu P lalu 3 lalu 50

Sumber: Data Penelitian (2016)

Tabel 3.12 Solusi Untuk *Device* Z86E83VEG-HA

	Kode	Keterangan
<i>Device name</i>	DEV7	Z86E83VEG-HA
<i>Fail category</i>	FC3	<i>Bin-8</i>
<i>Fail name</i>	FN3	<i>Continuity fail</i>
Penyebab	PYB7	<i>Failure</i> ini disebabkan oleh kondisi <i>receptacle loadboard</i> kurang baik sehingga koneksi <i>contact socket</i> ke <i>loadboard</i> kurang baik
Solusi		Ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-DB-052C

Sumber: Data Penelitian (2016)

Tabel 3.13 Solusi Untuk *Device* Z84C0006VEG-HB

	Kode	Keterangan
<i>Device name</i>	DEV8	Z84C0006VEG-HB
<i>Fail category</i>	FC3	<i>Bin-3</i>
<i>Fail name</i>	FN3	<i>VBO fail</i>
Penyebab	PYB8	<i>Failure</i> ini disebabkan oleh kondisi dari <i>contact finger</i> yang kurang baik seperti kotor, <i>misalign</i> atau patah
Solusi		Pastikan <i>contact finger</i> dalam kondisi yang baik, bersih, tidak ada yang patah dan tidak <i>misalign</i> . Jika perlu ganti <i>contact finger</i> dengan yang baru

Sumber: Data Penelitian (2016)

Tabel 3.14 Solusi Untuk *Device* Z86E3016SSG-DG

	Kode	Keterangan
<i>Device name</i>	DEV9	Z86E3016SSG-DG
<i>Fail category</i>	FC6	<i>Bin-6</i>
<i>Fail name</i>	FN6	<i>Functional fail</i>
Penyebab	PYB9	<i>Failure</i> ini disebabkan oleh kondisi <i>loadboard</i> kurang bagus, koneksi nya tidak stabil
Solusi		Ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-AD-010B

Sumber: Data Penelitian (2016)

Tabel 3.15 Solusi Untuk *Device* Z86E6316VSG-AN

	Kode	Keterangan
<i>Device name</i>	DEV10	Z86E6316VSG-AN
<i>Fail category</i>	FC4	<i>Bin-2</i>
<i>Fail name</i>	FN4	<i>ROM fail</i>
Penyebab	PYB10	<i>Failure</i> ini disebabkan oleh kondisi <i>loadboard</i> kurang bagus
Solusi		Ganti <i>loadboard</i> dengan <i>loadboard</i> yang memiliki nomor ID ZL-DB-099A

Sumber: Data Penelitian (2016)

Berdasarkan tabel-tabel solusi yang telah disusun, maka dengan menggunakan representasi pengetahuan model kaidah produksi (bentuk IF...THEN...), dapat disusun aturan-aturan yang akan diterapkan dalam sistem pakar ini yaitu:

1. Aturan 1: IF *DEV1* AND *FC6* AND *FN6* THEN *PYB1*. Artinya jika *device name* Z88C0020PSG-CC dan *fail category* Bin-6 dan *fail name functional fail* maka *failure* ini disebabkan oleh *test program* yang perlu dimodifikasi. Solusinya adalah gunakan *test program* yang telah dimodifikasi yaitu xxx_MHZ20_LOWV.EXE yang ada di *folder* 85"TEST".
2. Aturan 2: IF *DEV2* AND *FC4* AND *FN4* THEN *PYB2*. Artinya jika *device name* Z16C3220VSG00TR-CC dan *fail category* Bin-4 dan *fail name ICC fail* maka *failure* ini disebabkan oleh *delay* PARAM[1] pada *test program* terlalu kecil nilainya. Solusinya adalah tambahkan nilai PARAM[1]=100 pada *test program*. Caranya tekan CTRL+A lalu D lalu P lalu 1 lalu 100.
3. Aturan 3: IF *DEV3* AND *FC4* AND *FN4* THEN *PYB3*. Artinya jika *device name* Z84C1506FEG-KA dan *fail category* Bin-4 dan *fail name ICC fail* maka *failure* ini disebabkan oleh *delay* PARAM[3] pada *test program* terlalu kecil nilainya. Solusinya adalah tambahkan nilai PARAM[3]=100 pada *test program*. Caranya tekan CTRL+A lalu D lalu P lalu 3 lalu 100.
4. Aturan 4: IF *DEV4* AND *FC6* AND *FN6* THEN *PYB4*. Artinya jika *device name* Z8523L10VEG-HB dan *fail category* Bin-6 dan *fail name functional fail* maka *failure* ini disebabkan oleh *delay* PARAM[6] pada *test program*

terlalu kecil nilainya. Solusinya adalah tambahkan nilai PARAM[6]=10 pada *test program*. Caranya tekan CTRL+A lalu D lalu P lalu 6 lalu 10.

5. Aturan 5: IF *DEV5* AND *FC5* AND *FN5* THEN *PYB5*. Artinya jika *device name* Z16C3010VEG-KC dan *fail category* Bin-5 dan *fail name* Gross fail maka *failure* ini disebabkan oleh kondisi *loadboard* kurang baik, seperti koneksi kabel yang buruk atau *loadboard* tidak dilengkapi dengan kapasitor. Solusinya adalah pastikan sambungan kabel pada *loadboard* dalam keadaan baik, pastikan terdapat kapasitor sebagai *filter* tegangan antara positif (VCC) dan negatif (GND), jika masih tidak bisa *pass*, ganti *loadboard* dengan *loadboard* yang memiliki nomor ID ZL-DB-015B.
6. Aturan 6: IF *DEV6* AND *FC7* AND *FN7* THEN *PYB6*. Artinya jika *device name* Z84C0006AEG-AN dan *fail category* Bin-7 dan *fail name* DC fail maka *failure* ini disebabkan oleh *delay* PARAM[3] pada *test program* terlalu kecil nilainya. Solusinya adalah tambahkan nilai PARAM[3]=50 pada *test program*. Caranya tekan CTRL+A lalu D lalu P lalu 3 lalu 50.
7. Aturan 7: IF *DEV7* AND *FC3* AND *FN3* THEN *PYB7*. Artinya jika *device name* Z86E83VEG-HA dan *fail category* Bin-3 dan *fail name* VBO fail maka *failure* ini disebabkan oleh kondisi *receptacle loadboard* kurang baik sehingga koneksi *contact socket* ke *loadboard* kurang baik. Solusinya adalah ganti *loadboard* dengan *loadboard* yang memiliki nomor ID ZL-DB-052C.
8. Aturan 8: IF *DEV8* AND *FC3* AND *FN3* THEN *PYB8*. Artinya jika *device name* Z84C0006VEG-HB dan *fail category* Bin-3 dan *fail name* VBO fail maka *failure* ini disebabkan oleh kondisi dari *contact finger* yang kurang baik

seperti kotor, *misalign* atau patah. Solusinya adalah pastikan *contact finger* dalam kondisi yang baik, bersih, tidak ada yang patah dan tidak *misalign*. Jika perlu ganti *contact finger* dengan yang baru.

9. Aturan 9: IF *DEV9* AND *FC6* AND *FN6* THEN *PYB9*. Artinya jika *device name* Z86E3016SSG-DG dan *fail category* Bin-6 dan *fail name* *functional fail* maka *failure* ini disebabkan oleh kondisi *loadboard* kurang bagus, koneksinya tidak stabil. Solusinya adalah ganti *loadboard* dengan *loadboard* yang memiliki nomor ID ZL-AD-010B.
10. Aturan 10: IF *DEV10* AND *FC4* AND *FN4* THEN *PYB10*. Artinya jika *device name* Z86E6316VSG-AN dan *fail category* Bin-4 dan *fail name* *ICC fail* maka *failure* ini disebabkan oleh kondisi *loadboard* kurang bagus. Solusinya adalah ganti *loadboard* dengan *loadboard* yang memiliki nomor ID ZL-DB-099A.

Selanjutnya berdasarkan 10 aturan sistem pakar yang sudah terbentuk di atas, dibuat tabel keputusan dan pohon keputusan. Tabel keputusan dan pohon keputusan ini berguna untuk menentukan alur penelusuran pada mesin inferensi sehingga menghasilkan kesimpulan. Adapun tabel keputusan tersebut disajikan pada tabel 3.16.

Tabel 3.16 Tabel Keputusan Sistem Pakar

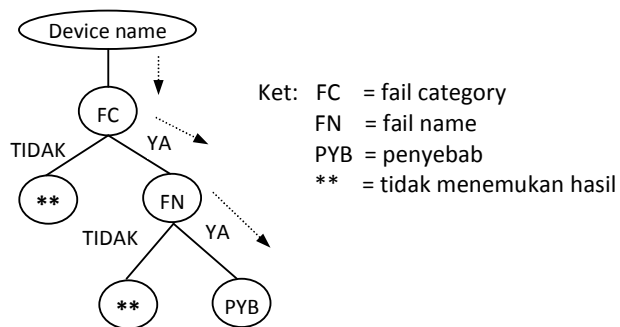
Penyebab Indikator	PYB1	PYB2	PYB3	PYB4	PYB5	PYB6	PYB7	PYB8	PYB9	PYB10
DEV1	√									
DEV2		√								
DEV3			√							
DEV4				√						
DEV5					√					
DEV6						√				

Tabel 3.16 Lanjutan

Penyebab Indikator	PYB1	PYB2	PYB3	PYB4	PYB5	PYB6	PYB7	PYB8	PYB9	PYB10
DEV7							√			
DEV8								√		
DEV9									√	
DEV10										√
FC2										
FC3							√	√		
FC4		√	√							√
FC5					√					
FC6	√			√					√	
FC7						√				
FC8										
FN2										
FN3							√	√		
FN4		√	√							√
FN5					√					
FN6	√			√					√	
FN7						√				
FN8										

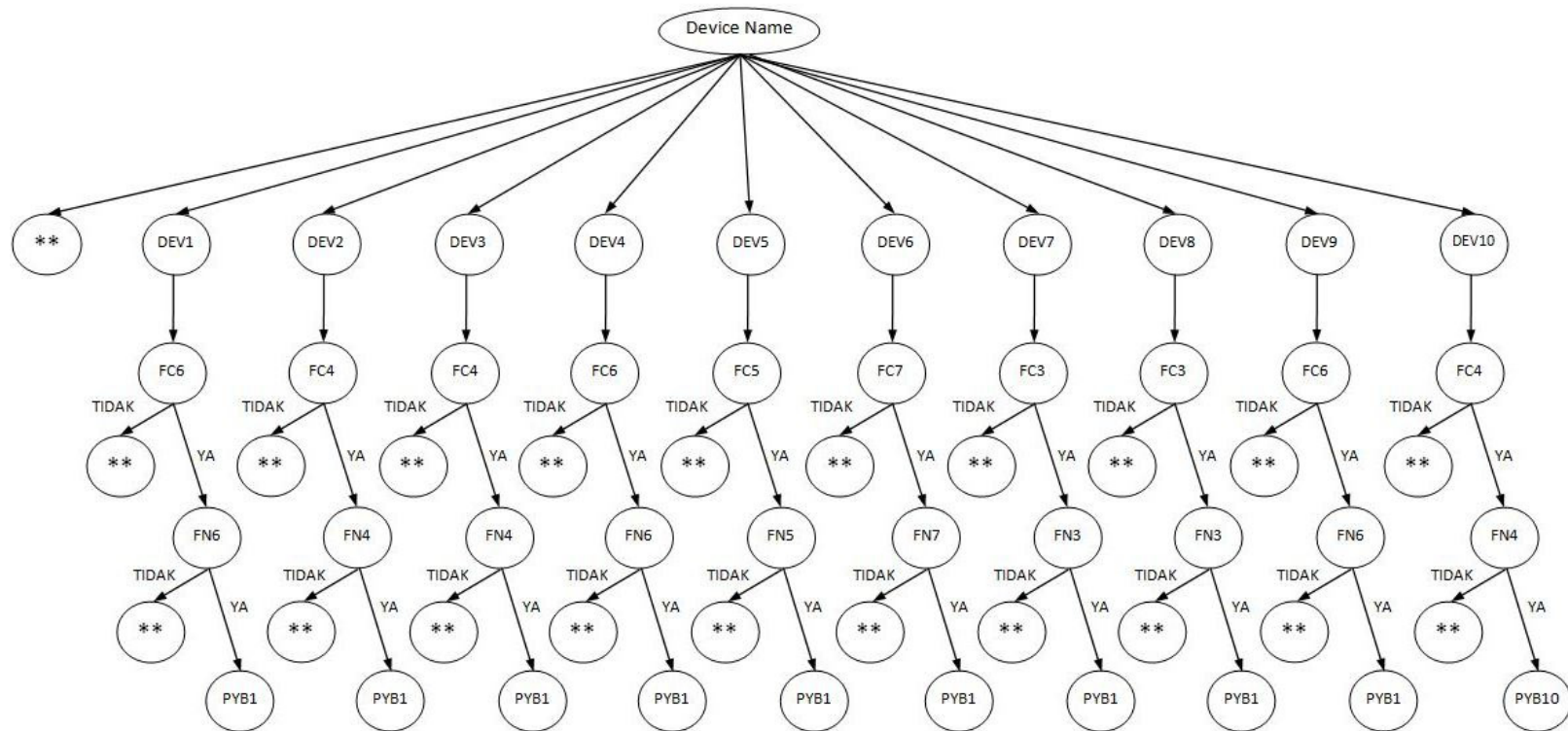
Sumber: Data Penelitian (2016)

Mesin inferensi pada sistem pakar ini menggunakan metode *forward chaining* dengan teknik penelusuran *depth-first search*. Dalam penerapannya, untuk *device name* yang dimasukkan oleh *user*, sistem pakar akan memberikan pertanyaan tentang *fail category* dan *fail name* seperti pada gambar 3.2.

**Gambar 3.2** Penelusuran Teknik *Depth First Search*

(Sumber: Data Penelitian, 2016)

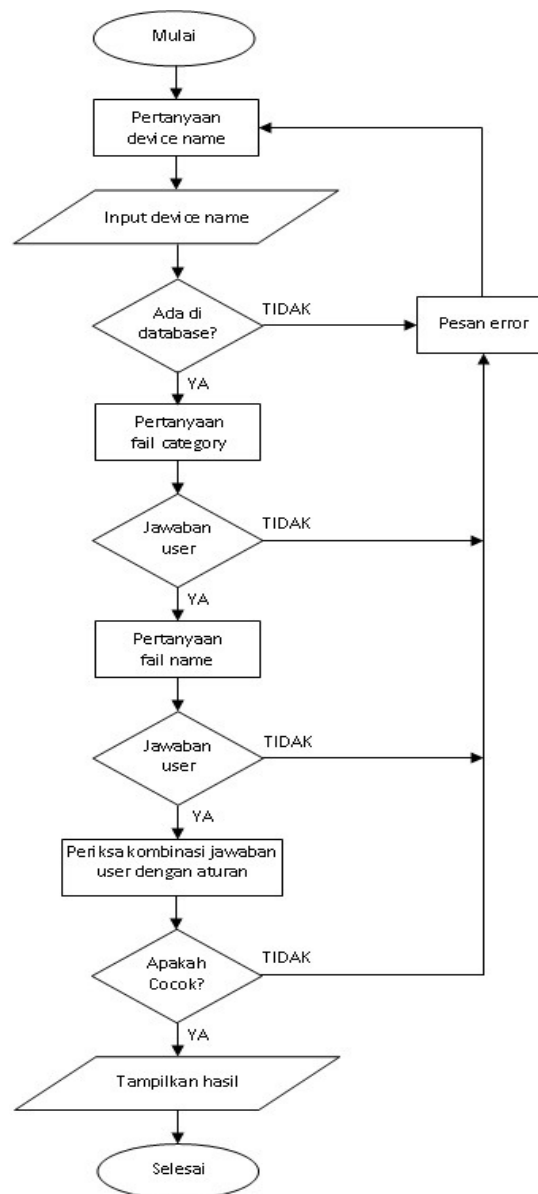
Pohon keputusan selengkapnya dapat dilihat pada gambar 3.3 berikut ini.



Gambar 3.3 Pohon Keputusan Sistem Pakar
(Sumber: Data Penelitian, 2016)

3.3.3 Desain Mesin Inferensi

Mesin inferensi pada sistem pakar ini menggunakan metode *forward chaining*. Adapun alur kerja mesin inferensi pada sistem pakar ini ditunjukkan pada gambar 3.4 berikut.



Gambar 3.4 Alur Kerja Mesin Inferensi Metode *Forward Chaining*
(Sumber: Data Penelitian, 2016)

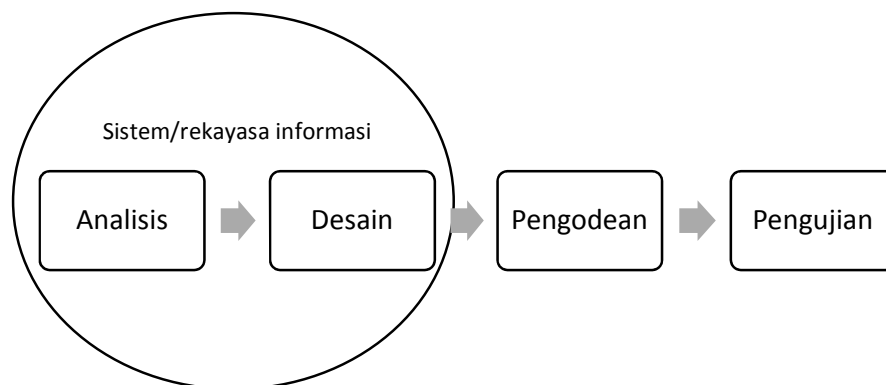
Berikut adalah penjelasan dari alur kerja mesin inferensi berdasarkan gambar 3.4 di atas:

1. Saat *user* mengakses sistem pakar, *user* diminta untuk memasukan *device name* melalui formulir *input* pada antarmuka sistem pakar.
2. Sistem pakar memeriksa apakah *device name* tersebut ada di dalam basis pengetahuan atau tidak. Jika tidak ada maka sistem mengeluarkan pesan *error*. Tetapi jika *device name* tersebut ada maka proses berlanjut ke tahap berikutnya.
3. Sistem pakar menampilkan *fail category* sesuai dengan *device name* yang dimasukkan oleh *user*.
4. *User* menjawab apakah *fail category* yang dialami sesuai dengan yang ditampilkan sistem pakar atau tidak. Apabila *user* menjawab “TIDAK” maka sistem mengeluarkan pesan *error*, tetapi apabila *user* menjawab “YA” maka proses berlanjut ke tahap berikutnya.
5. Sistem pakar menampilkan *fail name* sesuai dengan *device name* yang dimasukkan oleh *user*.
6. *User* menjawab apakah *fail name* yang dialami sesuai dengan yang ditampilkan sistem pakar atau tidak. Apabila *user* menjawab “TIDAK” maka sistem mengeluarkan pesan *error*, tetapi apabila *user* menjawab “YA” maka sistem pakar akan mencocokkan jawaban *user* dengan aturan yang terdapat pada sistem pakar. Apabila kombinasi jawaban *user* tidak cocok dengan aturan, sistem akan mengeluarkan pesan *error*, tetapi jika cocok maka sistem akan menampilkan penyebab kesalahan dan solusi untuk mengatasinya.

3.4 Perancangan Sistem

Menurut A.S. dan Shalahuddin (2011: 21), perancangan sistem merupakan upaya untuk membangun sebuah sistem yang memberikan kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performa maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat.

Dalam penelitian ini, peneliti menggunakan perancangan sistem model *waterfall* karena model ini adalah model yang sederhana dan mudah dalam pengerjaannya. Model *waterfall* menyediakan pendekatan alur hidup perangkat lunak secara berurutan dimulai dari analisis, desain, pengodean dan pengujian (A.S. dan Shalahuddin, 2011: 26). Ilustrasi model *waterfall* ditunjukkan pada gambar 3.5 berikut ini.

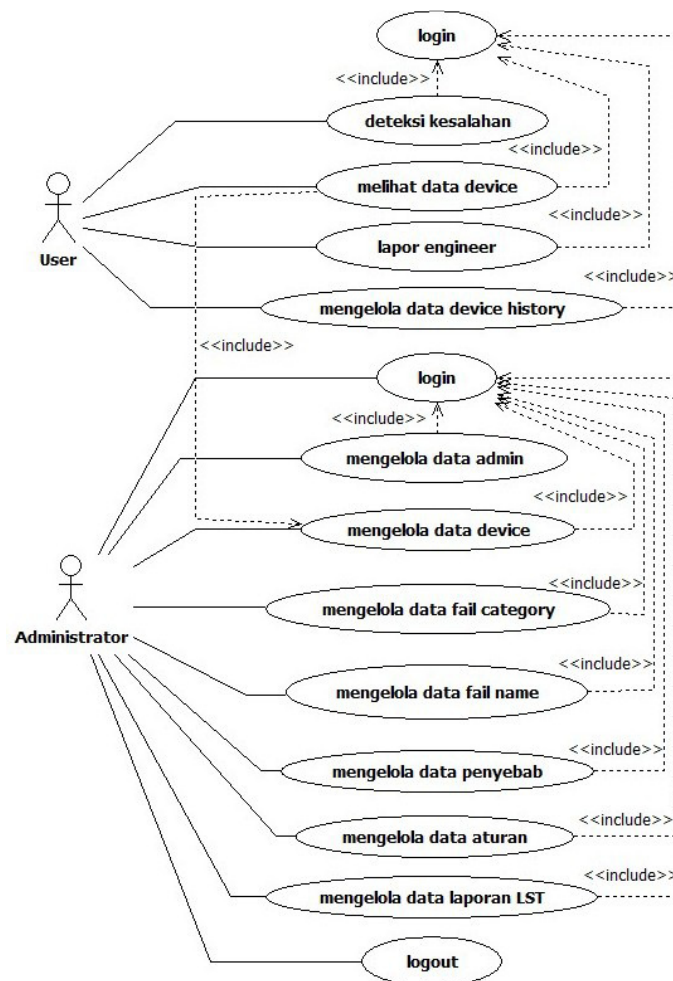


Gambar 3.5 Perancangan Sistem Model *Waterfall*
(Sumber: A.S. dan Shalahuddin, 2011: 27)

3.4.1 Desain UML (*Unified Modeling Language*)

Dalam penelitian ini, sistem pakar dimodelkan dengan menggunakan perangkat pemodelan *Unified Modelling Language (UML)*. Adapun diagram UML yang akan digunakan dalam penelitian ini ada 4 yaitu: *use case diagram*, *activity diagram*, *sequence digram* dan *class diagram*.

3.4.1.1 Use Case Diagram



Gambar 3.6 Use Case Diagram
(Sumber: Data Penelitian, 2016)

Pada gambar 3.6 disajikan gambar *use case diagram* dari sistem pakar pada penelitian ini. Terdapat 2 aktor yang digambarkan dalam *use case diagram* yaitu *administrator* dan *user*. *Administrator* diperankan oleh *test engineer* sedangkan *user* adalah *Line Sustaining Technician (LST)* atau biasa disebut dengan teknisi. *Use case* yang terdapat dalam sistem pakar ini menggambarkan fungsi yang disediakan oleh sistem pakar bagi masing-masing aktor.

Deskripsi dari *use case diagram* pada gambar 3.6 disajikan pada tabel 3.17 berikut ini.

Tabel 3.17 Deskripsi *Use Case Diagram*

No	Use Case	Deskripsi
1	deteksi kesalahan	Proses untuk mencari solusi bagi permasalahan pada proses tes korelasi
2	melihat data <i>device</i>	Proses untuk melihat data <i>device</i> yang terdaftar pada sistem pakar. Pada fungsi ini <i>user</i> hanya dapat melihat data <i>device</i> dan tidak dapat menambah atau mengubah data
3	lapor <i>engineer</i>	Proses untuk membuat laporan kepada <i>test engineer</i> jika ada permasalahan baru yang belum dapat dicarikan solusinya oleh sistem pakar
4	mengelola data <i>device history</i>	Proses untuk menyimpan data <i>device history</i> . <i>Device history</i> adalah sebuah catatan tentang semua <i>device</i> yang pernah diproses. <i>Device history</i> berguna sebagai referensi bagi <i>user</i> (teknisi) untuk memudahkan proses tes korelasi dengan melihat catatan dari masa lalu
5	<i>login</i>	Proses untuk validasi sebelum <i>user/administrator</i> menggunakan sistem pakar
6	mengelola data <i>admin</i>	Proses untuk menambah, mengubah atau menghapus data <i>administrator</i> yang terdapat pada sistem pakar
7	mengelola data <i>device</i>	Proses untuk menambah, mengubah atau menghapus data <i>device</i> yang terdapat pada sistem pakar
8	mengelola data <i>fail category</i>	Proses untuk menambah, mengubah atau menghapus data <i>fail category</i> yang terdapat pada sistem pakar
9	mengelola data <i>fail name</i>	Proses untuk menambah, mengubah atau menghapus data <i>fail name</i> yang terdapat pada sistem pakar
10	mengelola data penyebab	Proses untuk mengubah data penyebab yang terdapat pada sistem pakar
11	mengelola data aturan	Proses untuk menambah, mengubah atau menghapus data aturan yang terdapat pada sistem pakar

Tabel 3.17 Lanjutan

No	Use Case	Deskripsi
12	mengelola data laporan lst	Proses untuk memberikan solusi atau menghapus data laporan LST yang terdapat pada sistem pakar
13	logout	Proses untuk keluar dari halaman <i>administrator</i>

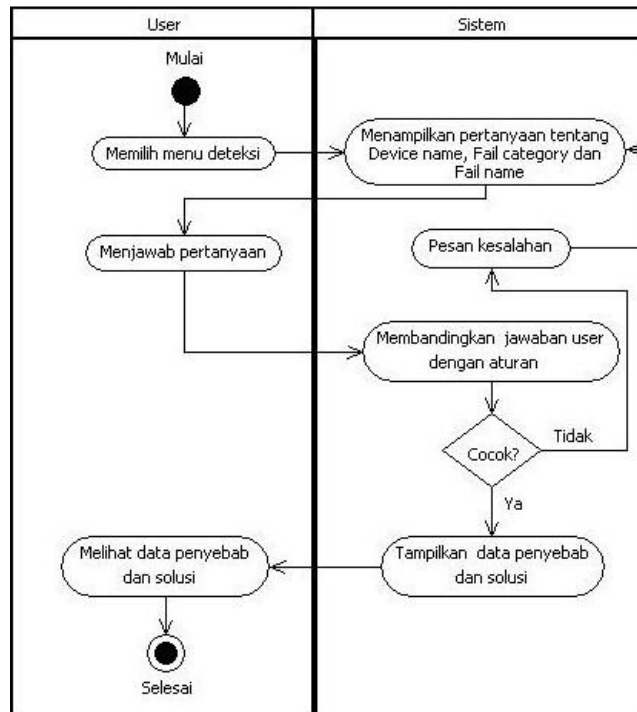
Sumber: Data Penelitian (2016)

3.4.1.2 Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktifitas dari sebuah sistem atau proses yang ada pada perangkat lunak (A.S. dan Shalahuddin, 2011: 134). *Activity diagram* pada sistem pakar ini dibagi menjadi 2 kategori yaitu *activity diagram* untuk menu *user* dan *activity diagram* untuk menu *administrator*. *Activity diagram* untuk menu *user* disajikan melalui gambar-gambar berikut.

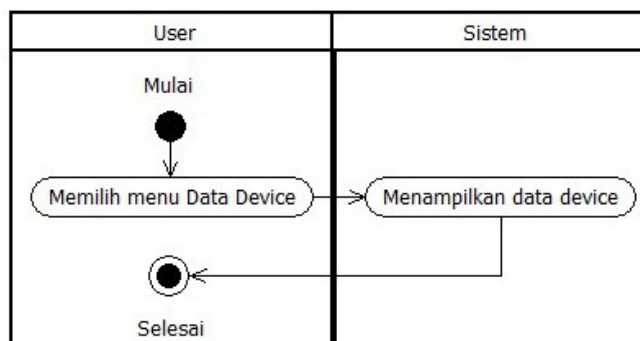
1. *Activity diagram* deteksi kesalahan

Pada gambar 3.7 disajikan *activity diagram* menu deteksi kesalahan. Melalui menu ini *user* dapat melakukan proses deteksi kesalahan pada proses tes korelasi dengan cara menjawab pertanyaan-pertanyaan yang muncul pada sistem pakar. Jawaban dari *user* akan dicocokkan dengan aturan yang ada dan sistem pakar akan menampilkan hasil deteksi berupa penyebab dan solusi. Apabila jawaban *user* tidak cocok dengan aturan maka sistem pakar akan ditampilkan pesan kesalahan.



Gambar 3.7 Activity Diagram Deteksi Kesalahan
(Sumber: Data Penelitian, 2016)

2. Activity diagram melihat data device

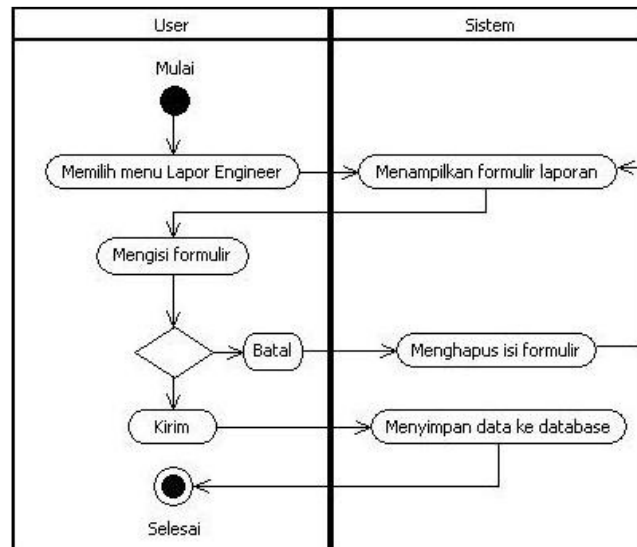


Gambar 3.8 Activity Diagram Melihat Data Device
(Sumber: Data Penelitian, 2016)

Pada gambar 3.8 disajikan *activity diagram* menu melihat data *device*. Melalui menu ini *user* hanya dapat melihat data *device* dan tidak dapat

menambah atau merubah data. Menambah atau mengubah data hanya dapat dilakukan oleh *administrator*.

3. Activity diagram lapor engineer

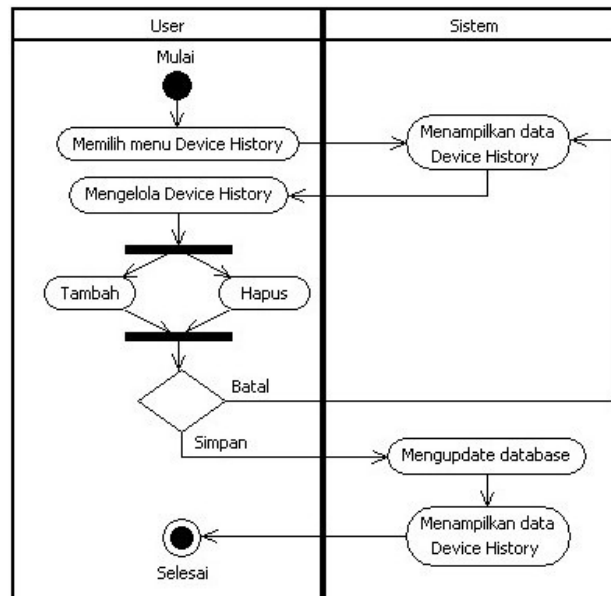


Gambar 3.9 Activity Diagram Lapor Engineer
(Sumber: Data Penelitian, 2016)

Pada gambar 3.9 disajikan *activity diagram* menu lapor engineer. Melalui menu ini *user* dapat melaporkan kesalahan yang belum bisa dideteksi oleh sistem pakar. Untuk mengirim laporan, *user* mengisi formulir dan data laporan akan disimpan ke *database*.

4. Activity diagram mengelola data device history

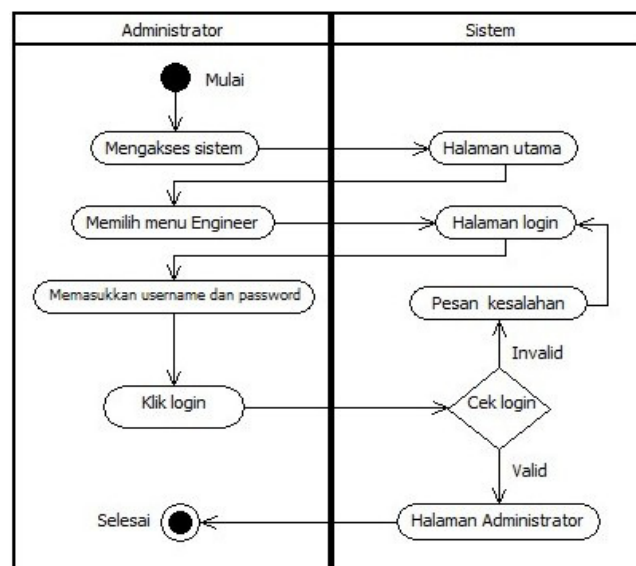
Pada gambar 3.10 disajikan *activity diagram* menu mengelola data *device history*. *Device history* berguna sebagai referensi bagi *user* (teknisi) untuk memudahkan proses tes korelasi dengan melihat catatan dari masa lalu. Pada menu ini *user* dapat melihat, menambah atau menghapus data *device history*. Hasil penambahan atau penghapusan data akan disimpan ke *database*.



Gambar 3.10 Activity Diagram Mengelola Data Device History
(Sumber: Data Penelitian, 2016)

Selanjutnya *activity diagram* untuk menu *administrator* disajikan melalui gambar-gambar berikut.

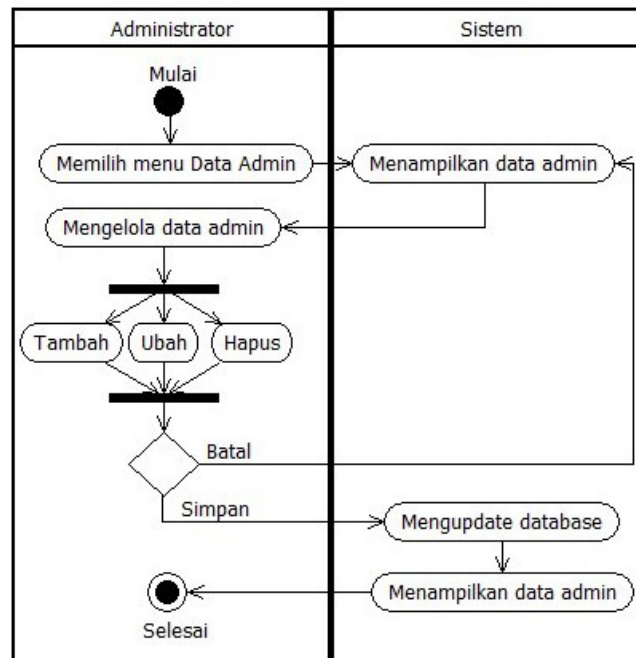
1. Activity diagram login



Gambar 3.11 Activity Diagram Login
(Sumber: Data Penelitian, 2016)

Pada gambar 3.11 disajikan gambar *activity diagram* menu *login*. Untuk mengakses halaman *administrator*, pada halaman utama sistem pakar, seorang *administrator* harus memilih menu *Engineer* kemudian memasukkan *username* dan *password*. Sistem akan melakukan validasi terhadap *username* dan *password* tersebut, apabila valid maka sistem akan mengarahkan ke halaman *administrator* tetapi apabila tidak valid maka sistem akan menampilkan pesan kesalahan dan mengarahkan kembali ke halaman *login*.

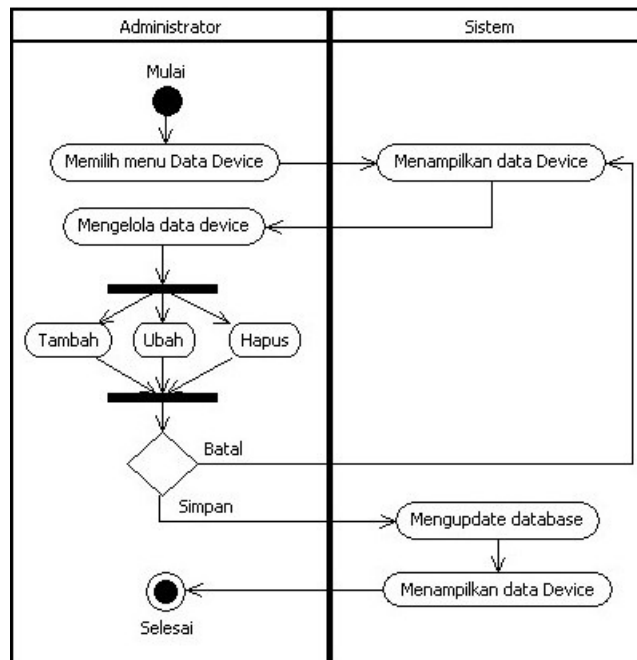
2. Activity diagram mengelola data admin



Gambar 3.12 *Activity Diagram* Mengelola Data Admin
(Sumber: Data Penelitian, 2016)

Pada gambar 3.12 disajikan *activity diagram* menu mengelola data *admin*. Melalui menu ini *administrator* dapat melihat data *admin* yang terdaftar pada sistem pakar. Selain itu *administrator* dapat melakukan tambah, ubah dan hapus data *admin*. Semua perubahan yang terjadi akan disimpan ke *database*.

3. Activity diagram mengelola data *device*

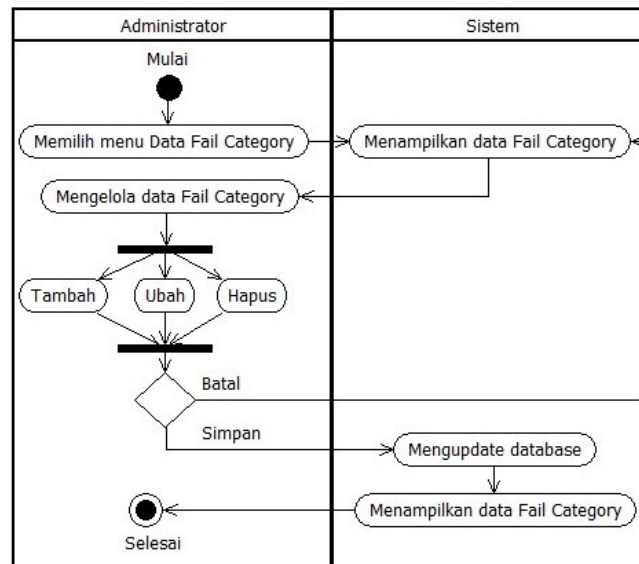


Gambar 3.13 Activity Diagram Mengelola Data Device
(Sumber: Data Penelitian, 2016)

Pada gambar 3.13 disajikan *activity diagram* menu mengelola data *device*. Melalui menu ini *administrator* dapat melihat data *device* yang terdaftar pada sistem pakar. Selain itu *administrator* dapat melakukan tambah, ubah dan hapus data *device*. Semua perubahan yang terjadi akan disimpan ke *database*.

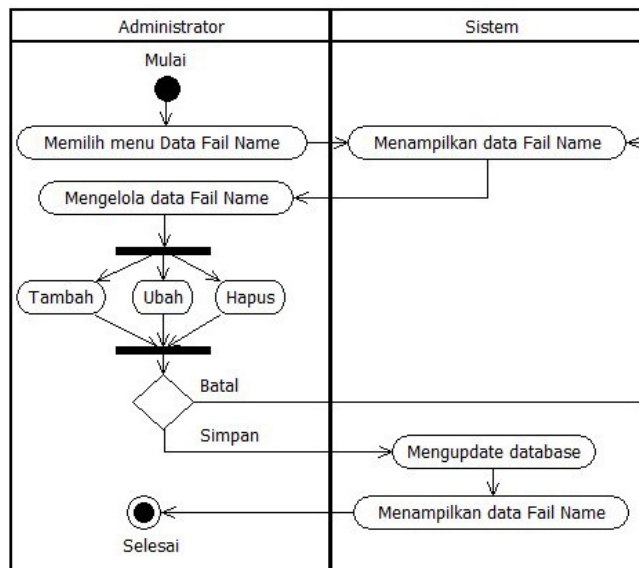
4. Activity diagram mengelola data *fail category*

Pada gambar 3.14 disajikan *activity diagram* menu mengelola data *fail category*. Melalui menu ini *administrator* dapat melihat data *fail category* yang terdaftar pada sistem pakar. Selain itu *administrator* dapat melakukan tambah, ubah dan hapus data *fail category*. Semua perubahan yang terjadi akan disimpan ke *database*.



Gambar 3.14 *Activity Diagram Mengelola Data Fail Category*
(Sumber: Data Penelitian, 2016)

5. *Activity diagram mengelola data fail name*



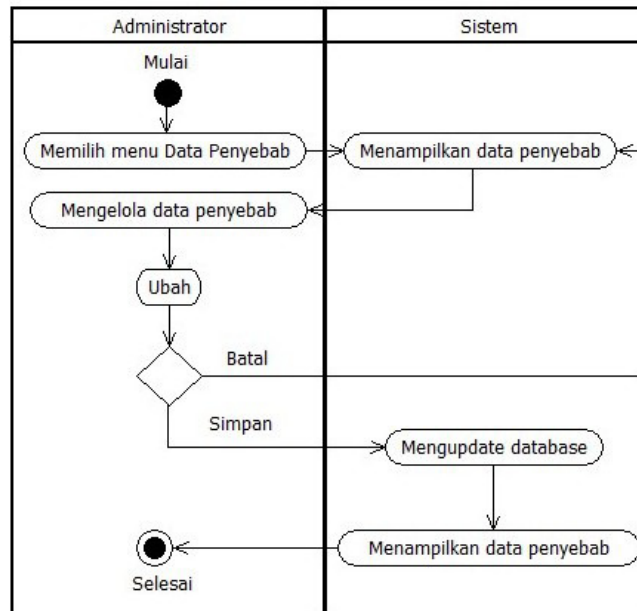
Gambar 3.15 *Activity Diagram Mengelola Data Fail Name*
(Sumber: Data Penelitian, 2016)

Pada gambar 3.15 disajikan *activity diagram* menu mengelola data *fail name*.

Melalui menu ini *administrator* dapat melihat data *fail name* yang terdaftar

pada sistem pakar. Selain itu *administrator* dapat melakukan tambah, ubah dan hapus data *fail name*. Semua perubahan yang terjadi akan disimpan ke *database*.

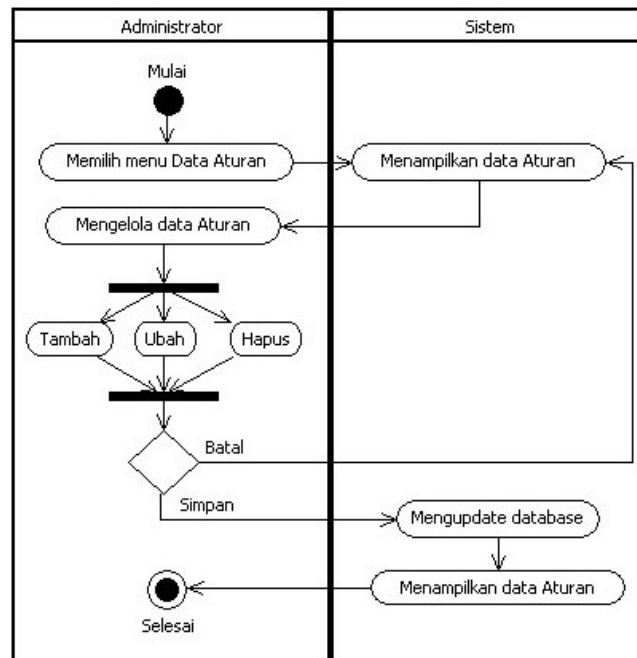
6. *Activity diagram* mengelola data penyebab



Gambar 3.16 *Activity Diagram* Mengelola Data Penyebab
(Sumber: Data Penelitian, 2016)

Pada gambar 3.16 disajikan *activity diagram* menu mengelola data penyebab. Melalui menu ini *administrator* dapat melihat dan mengubah data penyebab yang terdaftar pada sistem pakar. Perubahan yang terjadi akan disimpan ke *database*. Pada menu ini tidak disediakan fungsi untuk menambah dan menghapus penyebab. Untuk menambah dan menghapus data penyebab dapat dilakukan melalui menu Data Aturan.

7. *Activity diagram* mengelola data aturan

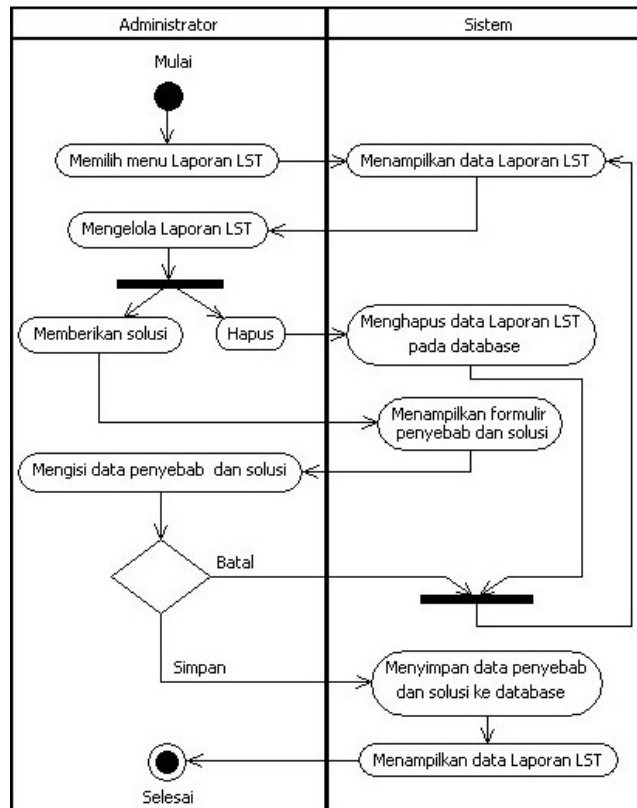


Gambar 3.17 *Activity Diagram* Mengelola Data Aturan
(Sumber: Data Penelitian, 2016)

Pada gambar 3.17 disajikan *activity diagram* menu mengelola data aturan. Melalui menu ini *administrator* dapat melihat data aturan yang terdaftar pada sistem pakar. Selain itu *administrator* dapat melakukan tambah, ubah dan hapus data aturan. Semua perubahan yang terjadi akan disimpan ke *database*.

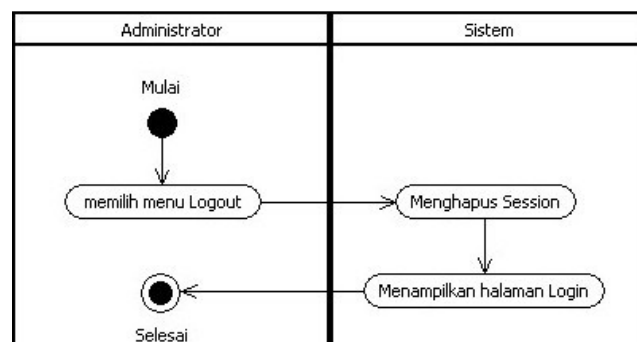
8. *Activity diagram* mengelola data laporan LST (*Line Sustaining Technician*)

Pada gambar 3.18 disajikan *activity diagram* menu mengelola data laporan LST atau teknisi. Melalui menu ini *administrator* dapat melihat data laporan kesalahan proses tes korelasi yang dikirim oleh LST. Pada menu ini *administrator* juga dapat memberikan solusi untuk kesalahan dengan cara mengisi formulir penyebab dan solusi atau menghapus laporan tersebut. Semua perubahan yang terjadi akan disimpan ke *database*.



Gambar 3.18 Activity Diagram Mengelola Data Laporan LST
(Sumber: Data Penelitian, 2016)

9. Activity diagram logout



Gambar 3.19 Activity Diagram Logout
(Sumber: Data Penelitian, 2016)

Pada gambar 3.19 disajikan *activity diagram* menu *logout*. Pada saat administrator memilih menu *logout*, sistem akan menghapus *session* dan mengarahkan ke halaman *login*.

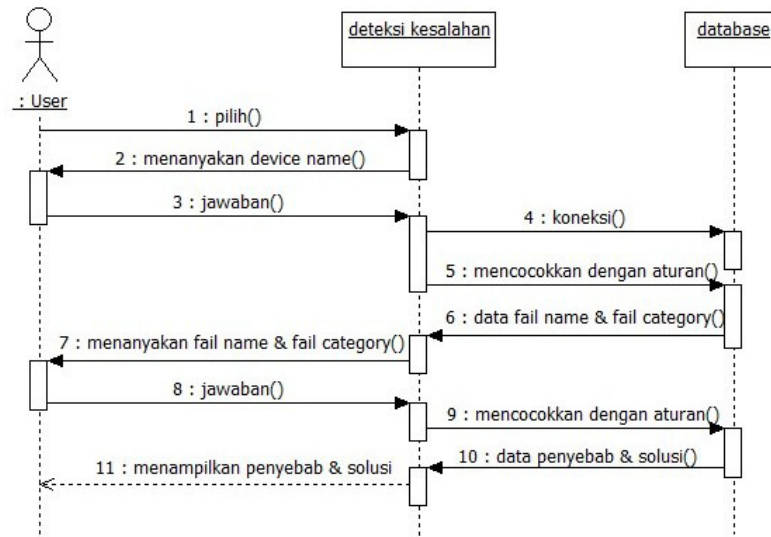
3.4.1.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup (*life cycle*) objek dan pesan (*message*) yang dikirimkan dan diterima antar objek. (A.S dan Shalahuddin, 2011: 137). Dalam penelitian ini jumlah *sequence diagram* yang dibuat adalah sebanyak proses pada *use case diagram* yang sudah dibuat sebelumnya.

Selanjutnya *sequence diagram* pada sistem pakar ini dibagi menjadi 2 kategori yaitu *sequence diagram* untuk menggambarkan aksi yang dilakukan oleh *user* serta *sequence diagram* untuk menggambarkan aksi yang dilakukan oleh *administrator*. Berikut ini adalah *sequence diagram* untuk menggambarkan aksi yang dilakukan oleh *user*.

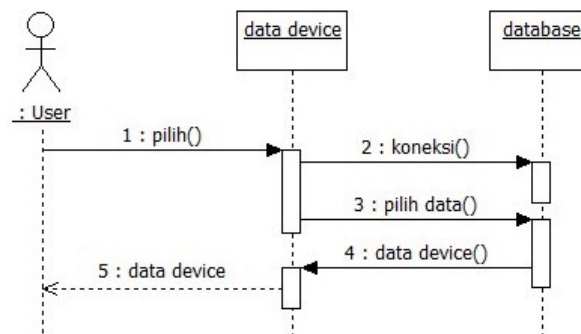
1. *Sequence diagram* deteksi kesalahan

Pada gambar 3.20 disajikan *sequence diagram* deteksi kesalahan. Pada saat *user* memilih menu deteksi kesalahan, sistem mengajukan pertanyaan kemudian *user* memberikan jawaban. Selanjutnya sistem berinteraksi dengan *database* untuk menampilkan solusi sesuai dengan kondisi yang dialami oleh *user*.



Gambar 3.20 *Sequence Diagram* Deteksi Kesalahan
(Sumber: Data Penelitian, 2016)

2. *Sequence diagram* melihat data device



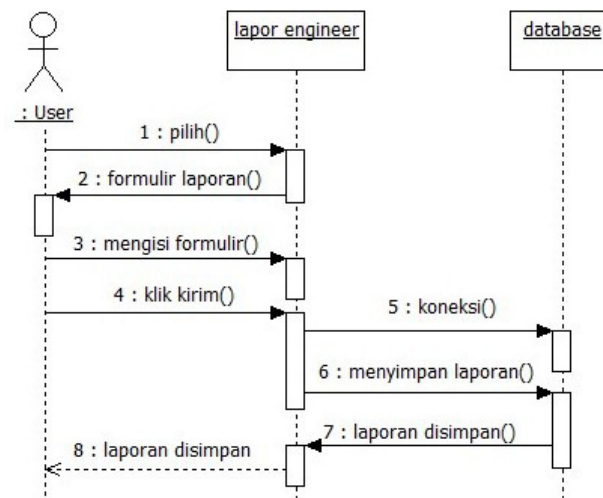
Gambar 3.21 *Sequence Diagram* Melihat Data Device
(Sumber: Data Penelitian, 2016)

Pada gambar 3.21 disajikan *sequence diagram* melihat data device. Pada saat *user* memilih menu data device, sistem menampilkan data device yang terdaftar pada sistem pakar.

3. *Sequence diagram* lapor engineer

Pada gambar 3.22 disajikan *sequence diagram* lapor engineer. Pada saat *user* memilih menu lapor engineer, sistem menampilkan formulir laporan. Untuk

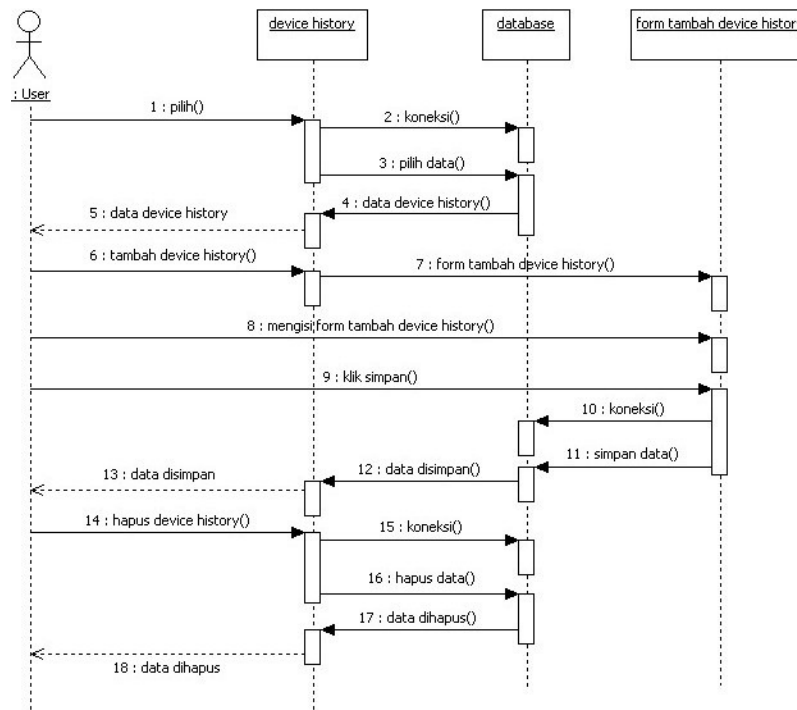
mengirim laporan maka *user* harus mengisi formulir tersebut. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan laporan yang dikirim oleh *user*.



Gambar 3.22 *Sequence Diagram* Lapor Engineer
(Sumber: Data Penelitian, 2016)

4. *Sequence diagram* mengelola data *device history*

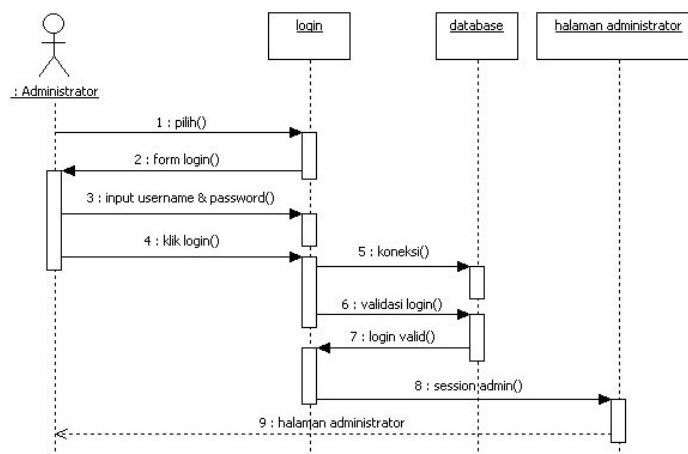
Pada gambar 3.23 disajikan *sequence diagram* mengelola data *device history*. Pada saat *user* memilih menu *device history*, sistem menampilkan data *device history*. *User* dapat melakukan tambah data dengan memilih menu tambah data dan mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan data tersebut. Selain itu, jika *user* memilih menu hapus data, sistem akan menghapus data dari *database*.



Gambar 3.23 *Sequence Diagram* Mengelola Data Device History
(Sumber: Data Penelitian, 2016)

Adapun *sequence diagram* untuk menggambarkan aksi yang dilakukan oleh administrator yaitu:

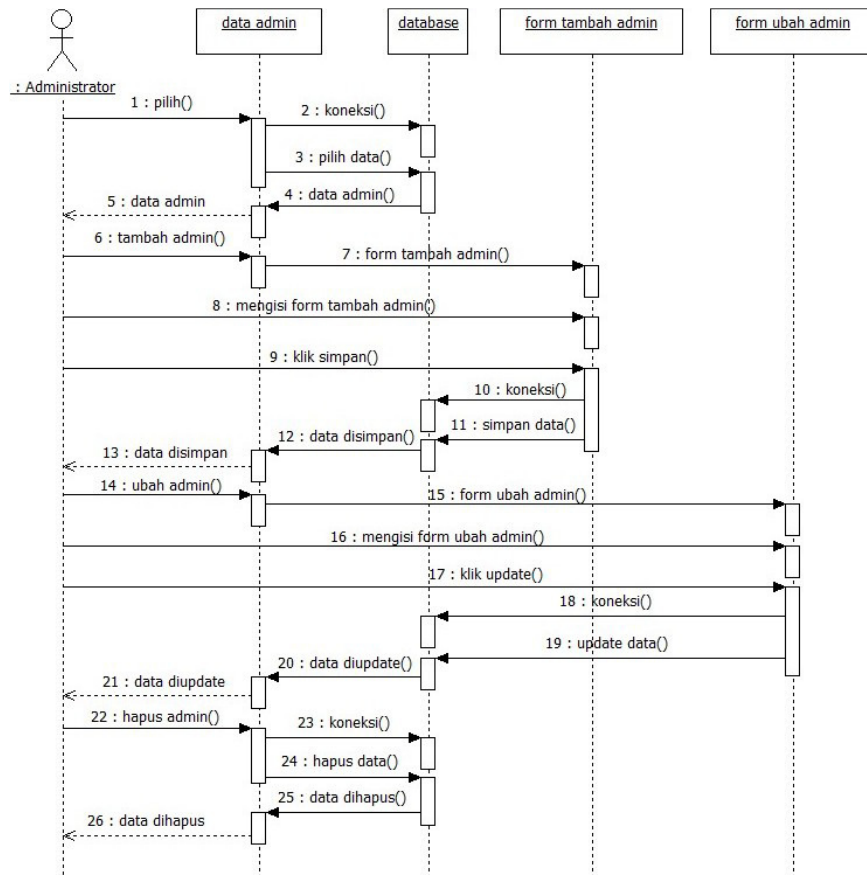
1. *Sequence diagram login*



Gambar 3.24 *Sequence Diagram Login*
(Sumber: Data Penelitian, 2016)

Pada gambar 3.24 disajikan *sequence diagram login*. Pada saat *administrator* memilih menu *login*, sistem menampilkan formulir *login* kemudian *administrator* mengisi formulir tersebut. Selanjutnya sistem berinteraksi dengan *database* untuk melakukan validasi *login* dan membuat *session*.

2. *Sequence diagram* mengelola data *admin*

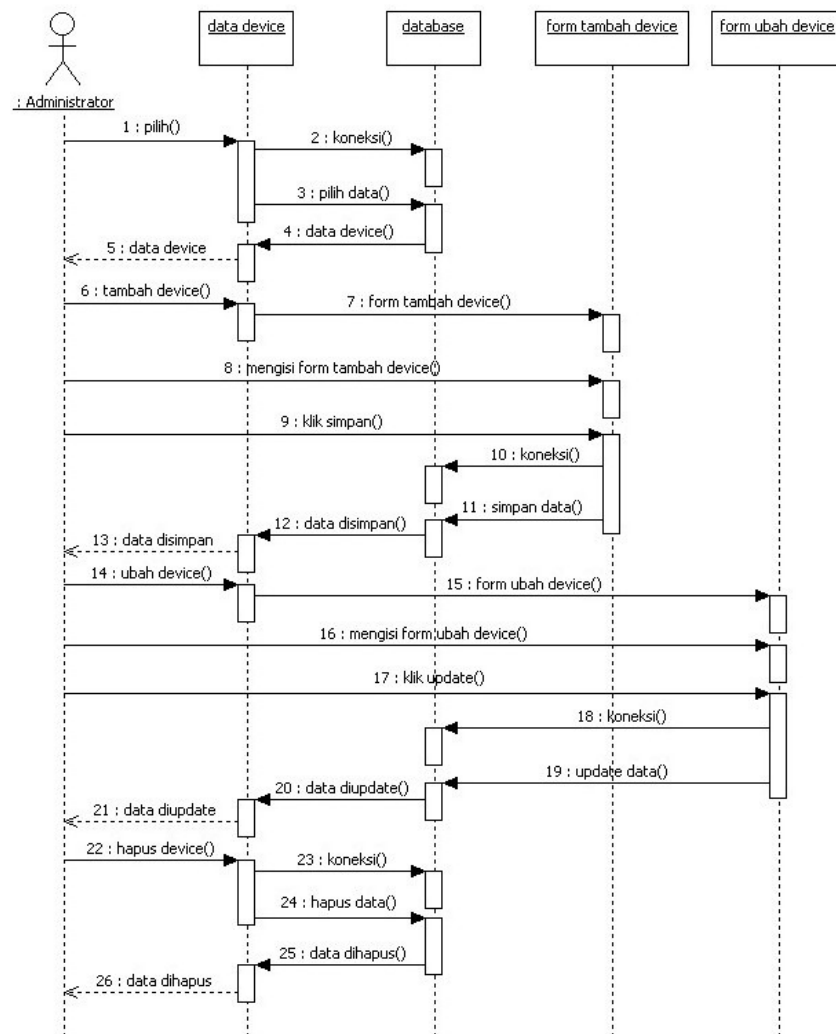


Gambar 3.25 *Sequence Diagram* Mengelola Data Admin
(Sumber: Data penelitian, 2016)

Pada gambar 3.25 disajikan *sequence diagram* mengelola data *admin*. Pada saat *administrator* memilih menu data *admin*, sistem menampilkan data *admin* yang diambil dari *database*. *Administrator* dapat melakukan tambah data atau ubah data dengan memilih menu tambah data atau ubah data dan

mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan atau perubahan data tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data dari *database*.

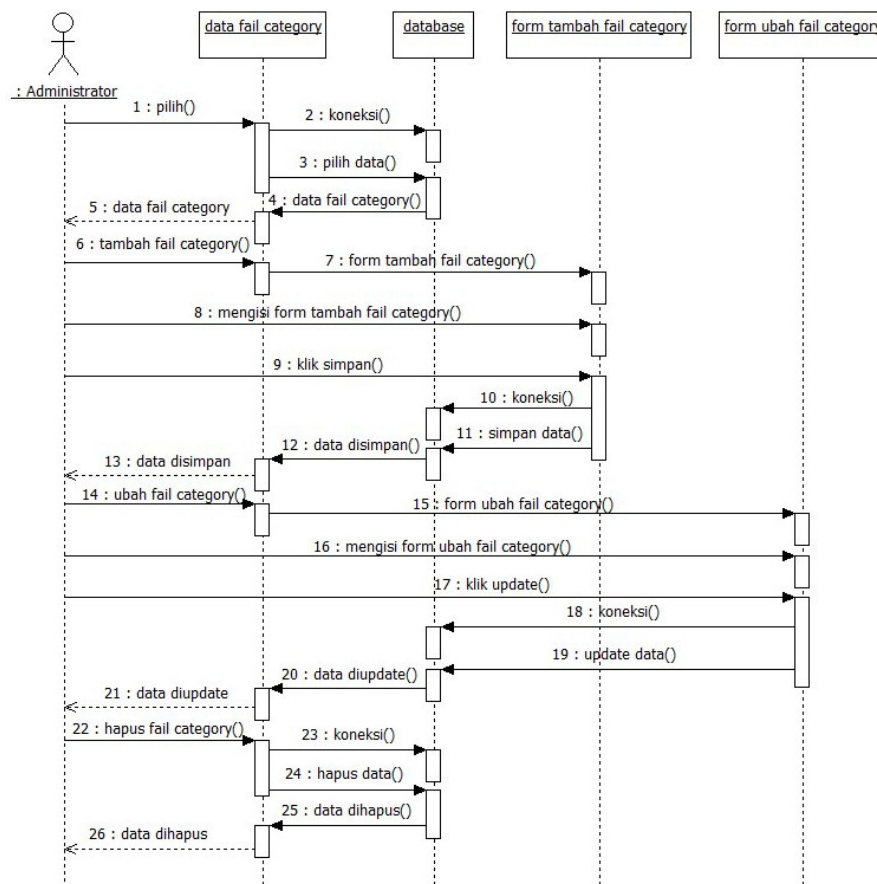
3. *Sequence diagram* mengelola data device



Gambar 3.26 *Sequence Diagram* Mengelola Data Device
(Sumber: Data Penelitian, 2016)

Pada gambar 3.26 disajikan *sequence diagram* mengelola data *device*. Pada saat *administrator* memilih menu data *device*, sistem menampilkan data *device* yang diambil dari *database*. *Administrator* dapat melakukan tambah data atau ubah data dengan memilih menu tambah data atau ubah data dan mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan atau perubahan data tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data dari *database*.

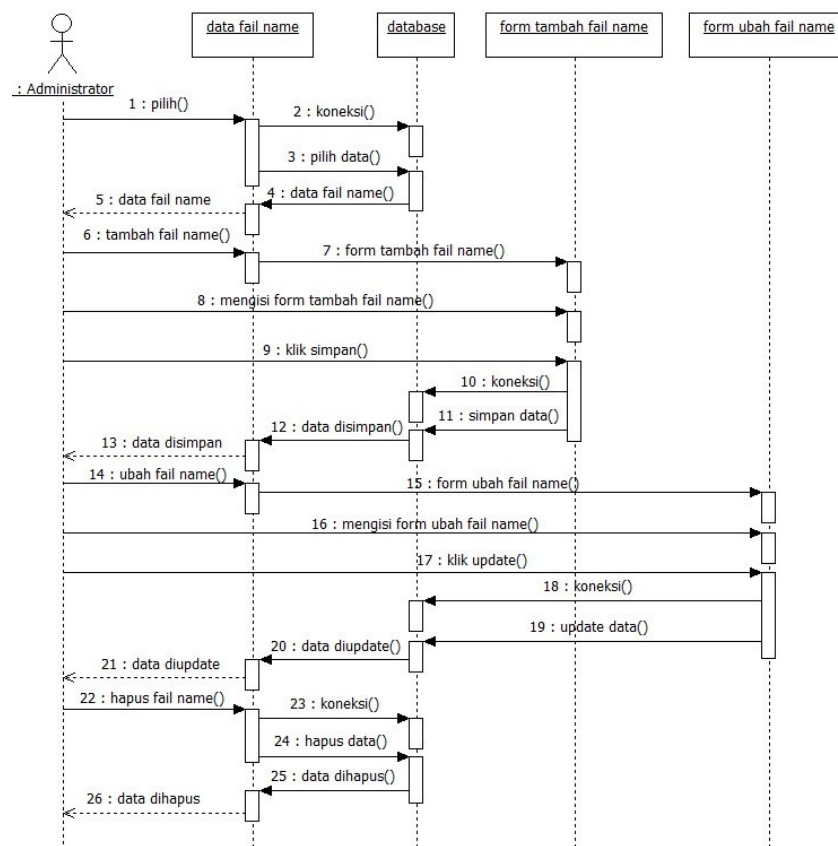
4. *Sequence diagram* mengelola data *fail category*



Gambar 3.27 *Sequence Diagram* Mengelola Data *Fail Category*
(Sumber: Data Penelitian, 2016)

Pada gambar 3.27 disajikan *sequence diagram* mengelola data *fail category*. Pada saat *administrator* memilih menu data *fail category*, sistem menampilkan data *fail category* yang diambil dari *database*. *Administrator* dapat melakukan tambah data atau ubah data dengan memilih menu tambah data atau ubah data dan mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan atau perubahan data tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data dari *database*.

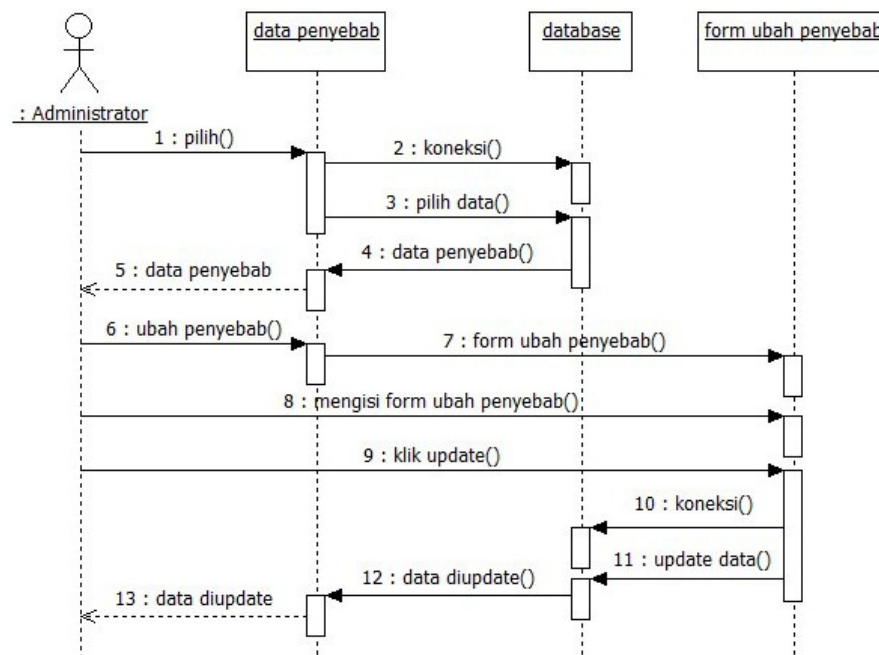
5. *Sequence diagram* mengelola data *fail name*



Gambar 3.28 *Sequence Diagram* Mengelola Data *Fail Name*
(Sumber: Data Penelitian, 2016)

Pada gambar 3.28 disajikan *sequence diagram* mengelola data *fail name*. Pada saat *administrator* memilih menu data *fail name*, sistem menampilkan data *fail name* yang diambil dari *database*. *Administrator* dapat melakukan tambah data atau ubah data dengan memilih menu tambah data atau ubah data dan mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan atau perubahan data tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data dari *database*.

6. *Sequence diagram* mengelola data penyebab

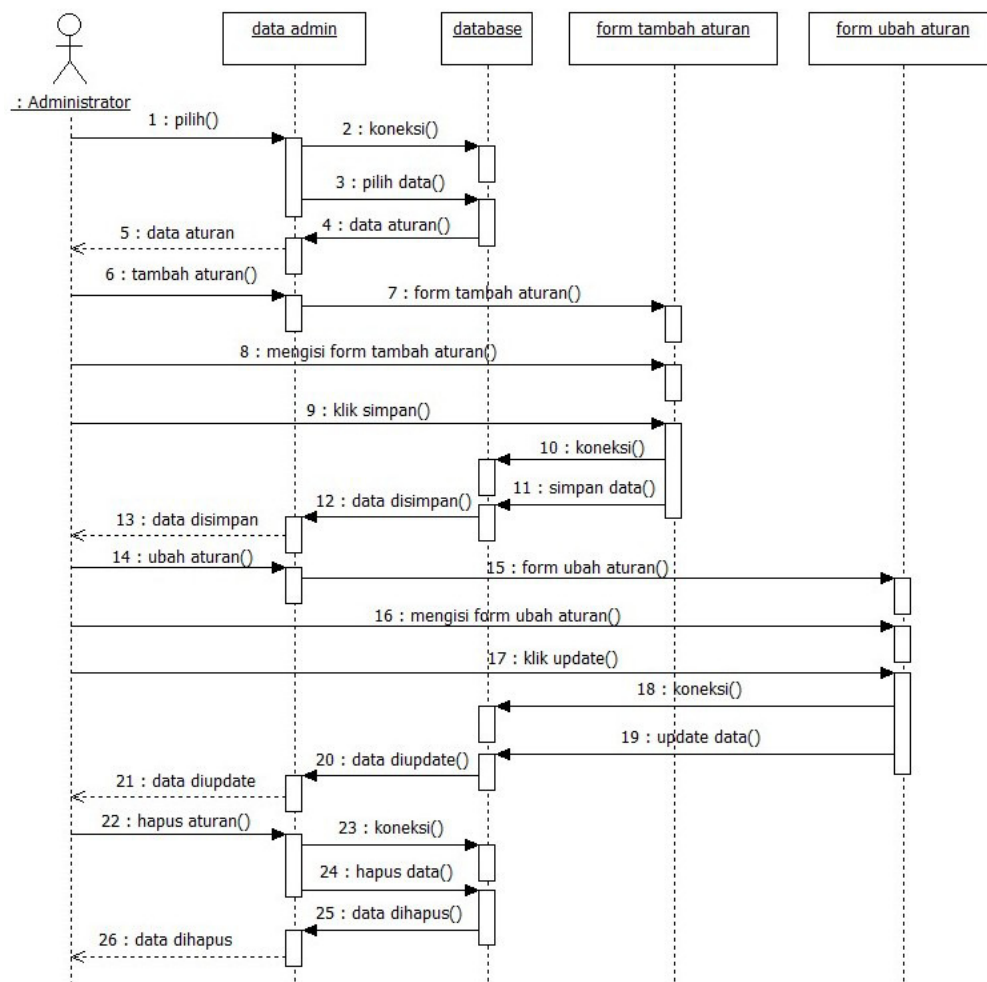


Gambar 3.29 *Sequence Diagram* Mengelola Data Penyebab
(Sumber: Data Penelitian, 2016)

Pada gambar 3.29 disajikan *sequence diagram* mengelola data penyebab. Pada saat *administrator* memilih menu data penyebab, sistem menampilkan data penyebab yang diambil dari *database*. *Administrator* dapat melakukan

tambah data atau ubah data dengan memilih menu tambah data atau ubah data dan mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan atau perubahan data tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data dari *database*.

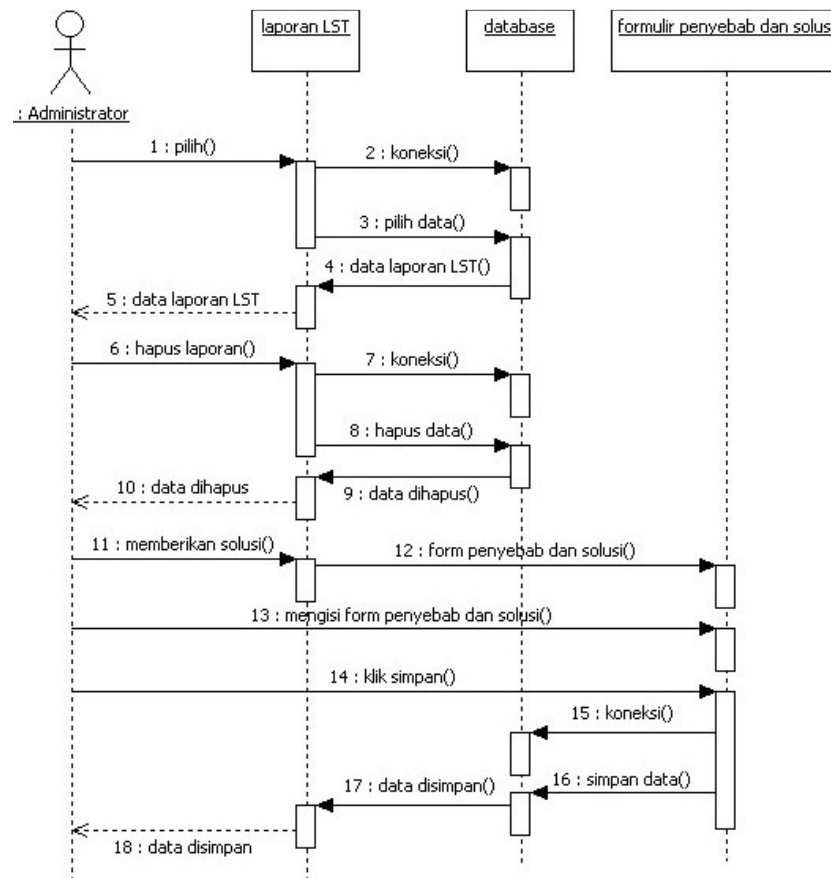
7. Sequence diagram mengelola data aturan



Gambar 3.30 Sequence Diagram Mengelola Data Aturan
(Sumber: Data Penelitian, 2016)

Pada gambar 3.30 disajikan *sequence diagram* mengelola data aturan. Pada saat *administrator* memilih menu data aturan, sistem menampilkan data aturan yang diambil dari *database*. *Administrator* dapat melakukan tambah data atau ubah data dengan memilih menu tambah data atau ubah data dan mengisi formulir yang ditampilkan oleh sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan penambahan atau perubahan data tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data dari *database*.

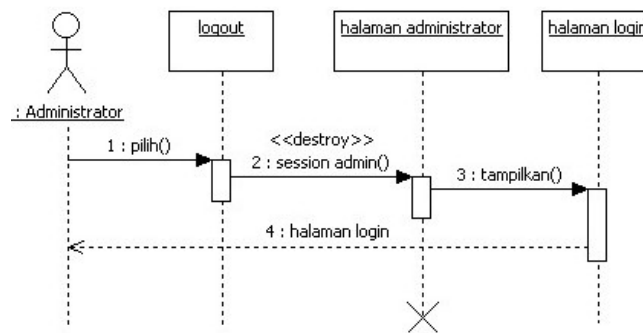
8. *Sequence diagram* mengelola data laporan LST



Gambar 3.31 *Sequence Diagram* Mengelola Data Laporan LST
(Sumber: Data Penelitian, 2016)

Pada gambar 3.31 disajikan *sequence diagram* mengelola data laporan LST. Pada saat *administrator* memilih menu data laporan LST, sistem menampilkan data laporan LST yang diambil dari *database*. *Administrator* menjawab laporan yaitu memberikan solusi dengan cara mengisi formulir yang ditampilkan sistem. Selanjutnya sistem berinteraksi dengan *database* untuk menyimpan data solusi tersebut. Selain itu, jika *administrator* memilih menu hapus data, sistem akan menghapus data laporan LST dari *database*.

9. *Sequence diagram logout*



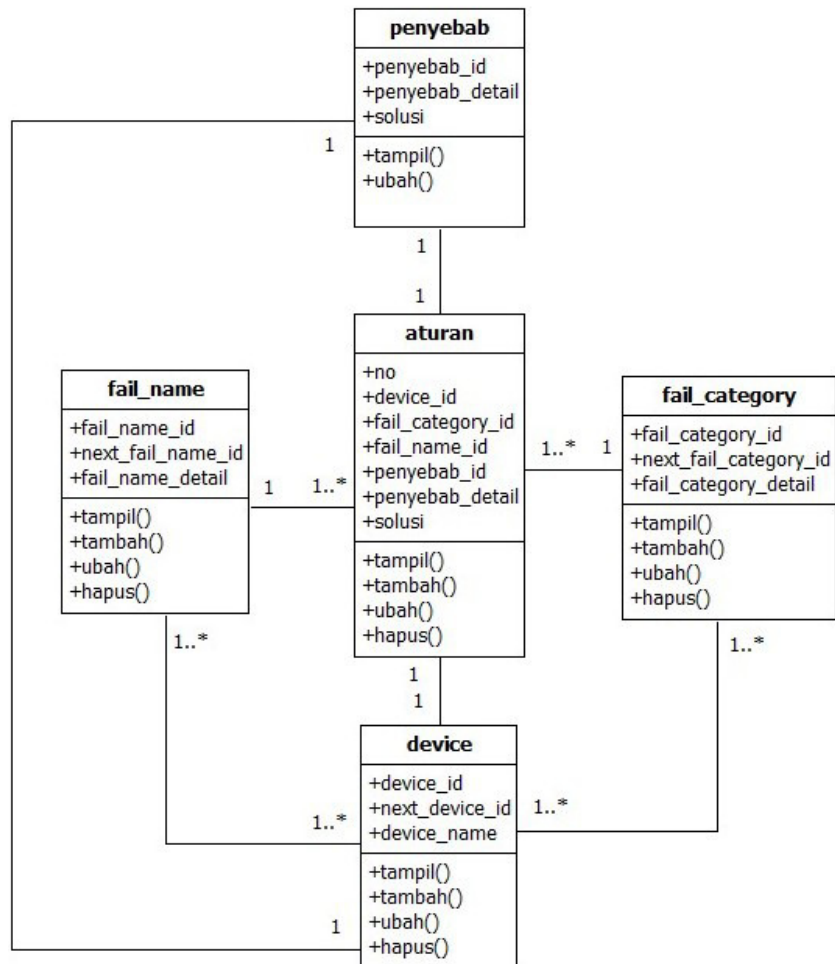
Gambar 3.32 *Sequence Diagram Logout*
(Sumber: Data Penelitian, 2016)

Pada gambar 3.32 disajikan *sequence diagram logout*. Pada saat administrator memilih menu *logout*, sistem akan menghapus *session* pada halaman administrator dan mengarahkan ke halaman *login*.

3.4.1.4 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode. Atribut yaitu variabel-variabel yang dimiliki oleh suatu kelas

sedangkan metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (A.S. dan Shalahuddin, 2011: 122).



Gambar 3.33 *Class Diagram*
(Sumber: Data Penelitian, 2016)

Gambar 3.33 menyajikan *class diagram* yang digunakan pada sistem pakar dalam penelitian ini. Deskripsi dari *class diagram* tersebut disajikan pada tabel 3.18 berikut.

Tabel 3.18 Deskripsi *Class Diagram*

No	Nama Kelas	Deskripsi
1	penyebab	Kelas proses yang diambil dari pendefinisian <i>use case</i> mengelola data penyebab
2	<i>fail_name</i>	Kelas proses yang diambil dari pendefinisian <i>use case</i> mengelola data <i>fail name</i>
3	aturan	Kelas proses yang diambil dari pendefinisian <i>use case</i> mengelola data aturan
4	<i>fail_category</i>	Kelas proses yang diambil dari pendefinisian <i>use case</i> mengelola data <i>fail category</i>
5	<i>device</i>	Kelas proses yang diambil dari pendefinisian <i>use case</i> melihat data <i>device</i> dan mengelola data <i>device</i>

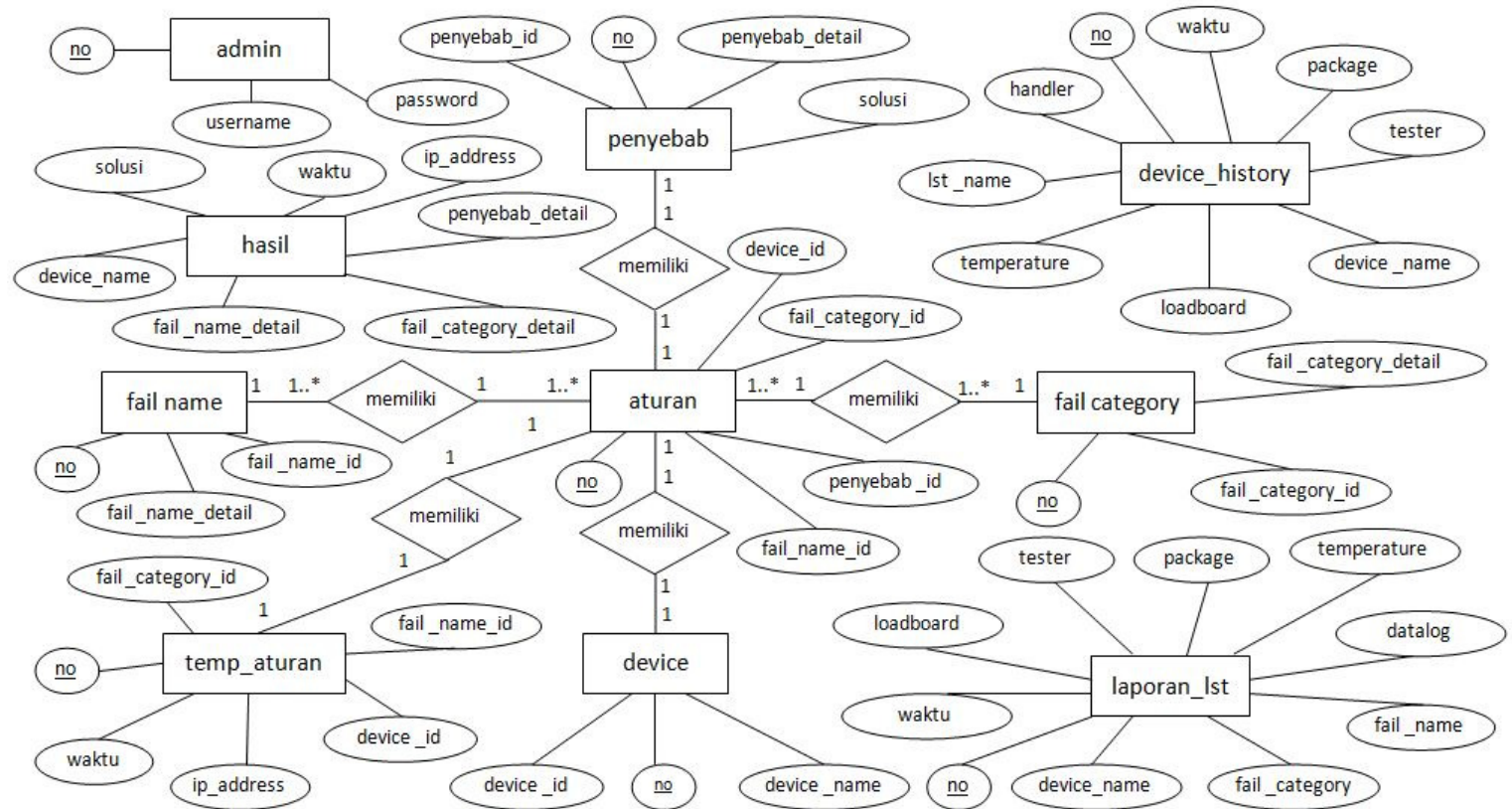
Sumber: Data Penelitian (2016)

Berdasarkan *class diagram* pada gambar 3.33 dapat dijelaskan relasi-relasi sebagai berikut:

- 1 *device* dapat memiliki lebih dari 1 *fail_category* dan begitu juga sebaliknya, 1 *fail_category* dapat dimiliki oleh lebih dari 1 *device* (*many to many*).
- 1 *device* dapat memiliki lebih dari 1 *fail_name* dan begitu juga sebaliknya, 1 *fail_name* dapat dimiliki oleh lebih dari 1 *device* (*many to many*).
- 1 *device* hanya memiliki 1 penyebab dan begitu juga sebaliknya, 1 penyebab hanya dimiliki oleh 1 *device* (*one to one*).
- 1 *device* hanya memiliki 1 aturan dan begitu juga sebaliknya, pada 1 aturan hanya terdapat 1 *device* (*one to one*).
- 1 aturan hanya memiliki 1 *fail_category*, tetapi 1 *fail_category* dapat dimiliki oleh lebih dari 1 aturan (*one to many*).
- 1 aturan hanya memiliki 1 *fail_name*, tetapi 1 *fail_name* dapat dimiliki oleh lebih dari 1 aturan (*one to many*).
- Dalam 1 aturan, hanya terdapat 1 penyebab dan begitu juga sebaliknya 1 penyebab hanya dimiliki oleh 1 aturan (*one to one*).

3.4.3 Desain *Database*

Dalam penelitian ini, peneliti membuat desain *database* menggunakan model *Entity Relationship Diagram (ERD)*. Adapun ERD pada sistem pakar ini dapat dilihat pada gambar 3.34.



Gambar 3.34 Desain *Entity Relationship Diagram (ERD)*
(Sumber: Data Penelitian, 2016)

Tabel yang digunakan pada sistem pakar ini berjumlah 10 tabel, yaitu:

1. Tabel *admin*

Nama tabel : *admin*

Primary key : no

Fungsi : menyimpan data *admin*

Tabel 3.19 Struktur Tabel *Admin*

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
username	varchar	32	-
password	varchar	64	-

Sumber: Data Penelitian (2016)

2. Tabel *device*

Nama tabel : *device*

Primary key : no

Fungsi : menyimpan data *device*

Tabel 3.20 Struktur Tabel *Device*

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
device_id	varchar	8	-
device_name	varchar	32	-

Sumber: Data Penelitian (2016)

3. Tabel *device_history*

Nama tabel : *device_history*

Primary key : no

Fungsi : menyimpan data *device_history*

Tabel 3.21 Struktur Tabel *Device_History*

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
waktu	datetime	-	-
device_name	varchar	32	-
package	varchar	16	-
tester	varchar	16	-
handler	varchar	16	-
loadboard	varchar	16	-
temperature	integer	8	-
lst_name	varchar	32	-

Sumber: Data Penelitian (2016)

4. Tabel laporan_lst

Nama tabel : laporan_lst

Primary key : no

Fungsi : menyimpan data laporan LST

Tabel 3.22 Struktur Tabel Laporan_LST

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
waktu	datetime	-	-
device_name	varchar	32	-
package	varchar	16	-
tester	varchar	16	-
loadboard	varchar	16	-
temperature	integer	8	-
fail_category	varchar	32	-
fail_name	varchar	32	-
datalog	varchar	64	-

Sumber: Data Penelitian (2016)

5. Tabel *fail_category*

Nama tabel : *fail_category*

Primary key : no

Fungsi : menyimpan data *fail_category*

Tabel 3.23 Struktur Tabel *Fail_Category*

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
fail_category_id	varchar	8	-
fail_category_detail	varchar	32	-

Sumber: Data Penelitian (2016)

6. Tabel *fail_name*

Nama tabel : *fail_name*

Primary key : no

Fungsi : menyimpan data *fail name*

Tabel 3.24 Struktur Tabel *Fail_Name*

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
fail_name_id	varchar	8	-
fail_name_detail	varchar	32	-

Sumber: Data Penelitian (2016)

7. Tabel penyebab

Nama tabel : penyebab

Primary key : no

Fungsi : menyimpan data penyebab

Tabel 3.25 Struktur Tabel Penyebab

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
penyebab_id	varchar	8	-
penyebab_detail	text	-	-
solusi	text	-	-

Sumber: Data Penelitian (2016)

8. Tabel aturan

Nama tabel : aturan

Primary key : no

Fungsi : menyimpan data aturan sistem pakar

Tabel 3.26 Struktur Tabel Aturan

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
device_id	varchar	8	-
fail_category_id	varchar	8	-
fail_name_id	varchar	8	-
penyebab_id	varchar	8	-

Sumber: Data Penelitian (2016)

9. Tabel temp_aturan

Nama tabel : temp_aturan

Primary key : no

Fungsi : menyimpan data aturan sementara selama proses deteksi

Tabel 3.27 Struktur Tabel Temp_Aturan

Kolom	Tipe Data	Panjang	Keterangan
no	integer	11	<i>primary key</i>
waktu	datetime	-	
ip_address	varchar	32	
device_id	varchar	8	-
fail_category_id	varchar	8	-
fail_name_id	varchar	8	-

Sumber: Data Penelitian (2016)

10. Tabel hasil

Nama tabel : hasil

Primary key : no

Fungsi : menyimpan data hasil deteksi

Tabel 3.28 Struktur Tabel Hasil

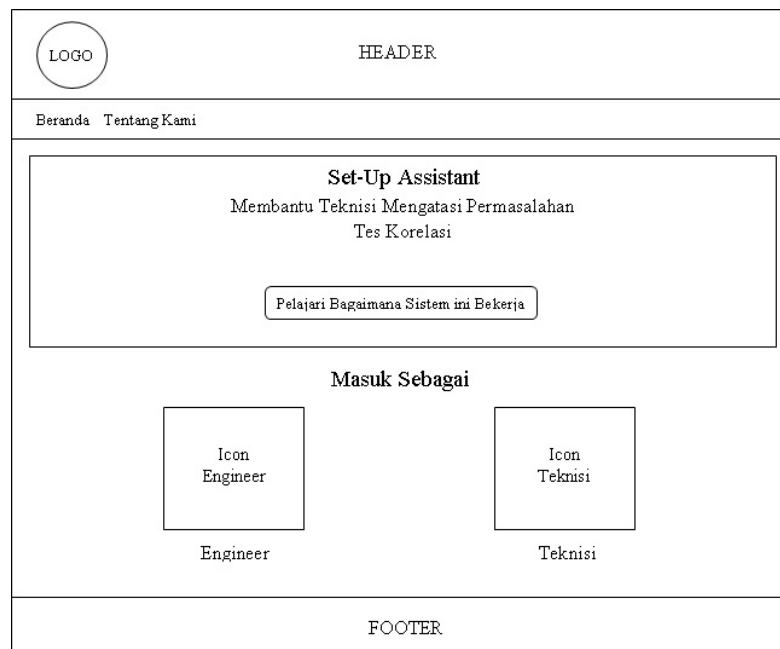
Kolom	Type Data	Panjang	Keterangan
waktu	datetime	-	-
ip_address	varchar	32	-
device_name	varchar	32	-
fail_category_detail	varchar	32	-
fail_name_detail	varchar	32	-
penyebab_detail	text	-	-
solusi	text	-	-

Sumber: Data Penelitian (2016)

3.4.4 Desain Antarmuka Pengguna (*User Interface*)

Berikut ini adalah desain-desain tampilan sistem pakar pada penelitian ini.

1. Desain halaman Beranda

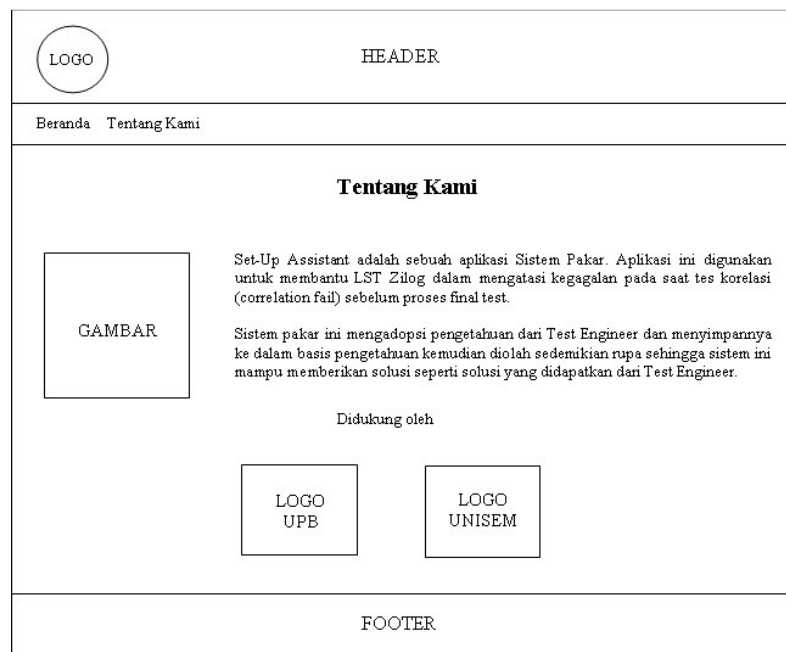


Gambar 3.35 Desain Halaman Beranda
(Sumber: Data Penelitian, 2016)

Gambar 3.35 merupakan desain halaman Beranda, yaitu halaman awal yang akan tampil pada saat *user/administrator* mengakses sistem pakar ini.

Halaman Beranda menampilkan informasi tentang sistem pakar dan pilihan untuk *login* sebagai *engineer* atau teknisi.

2. Desain halaman Tentang Kami



Gambar 3.36 Desain Halaman Tentang Kami
(Sumber: Data Penelitian, 2016)

Gambar 3.36 merupakan desain halaman Tentang Kami. Pada halaman ini ditampilkan informasi tentang manfaat sistem pakar dan pihak-pihak yang mendukung pembuatan sistem pakar ini.

3. Desain halaman Awal *User*

Gambar 3.37 merupakan desain halaman Awal *User*. Halaman ini akan ditampilkan pada saat *user* memilih masuk sebagai LST pada halaman Beranda.

LOGO	HEADER
Beranda Device History Lapor Engineer Data Device Tips	
Selamat Datang Set-Up Assistant membantu mencari solusi permasalahan tes korelasi Mulai Ikon Teknisi	
FOOTER	

Gambar 3.37 Desain Halaman Awal *User*
(Sumber: Data Penelitian, 2016)

4. Desain halaman Deteksi Kesalahan

LOGO	HEADER
Beranda Device History Lapor Engineer Data Device Tips	
Cari Solusi Untuk Permasalahan Tes Korrelasi Cari solusi permasalahan tes korelasi Ketik device name Contoh : Z86E6316VSG-AN CARI Petunjuk : Pastikan Anda mengisi device name dengan lengkap	
FOOTER	

Gambar 3.38 Desain Halaman Deteksi Kesalahan
(Sumber: Data Penelitian, 2016)

Gambar 3.38 merupakan desain halaman Deteksi Kesalahan. Pada halaman ini *user* dapat memulai menggunakan sistem pakar dengan cara memasukkan *device name* untuk mencari solusi dalam mengatasi permasalahan pada proses tes korelasi.

5. Desain halaman Hasil Deteksi

LOGO HEADER	
Beranda Device History Lapor Engineer Data Device Tips	
Hasil Deteksi Berikut ini adalah hasil deteksi untuk permasalahan yang Anda alami Device name XXXXXX Fail name XXXXXX Fail category XXXXXX Penyebab XXXXXX Solusi XXXXXX Deteksi ulang ? TIDAK YA	
FOOTER	

Gambar 3.39 Desain Halaman Hasil Deteksi
(Sumber: Data Penelitian, 2016)

Gambar 3.39 merupakan desain halaman Hasil Deteksi. Halaman ini menampilkan hasil deteksi kesalahan oleh sistem pakar. Hal penting yang ada pada halaman ini adalah informasi mengenai penyebab dan solusi yang akan menjadi panduan bagi *user* dalam mengatasi permasalahan tes korelasi. Selanjutnya *user* dapat memilih untuk melakukan deteksi ulang atau tidak.

6. Desain halaman *Device History*

Gambar 3.40 merupakan desain halaman *Device History*. Halaman ini menampilkan informasi tentang berbagai *device* yang pernah di proses pada mesin *tester* tertentu. Pada halaman ini juga *user* dapat menambah data baru atau menghapus data yang sudah ada.

LOGO		HEADER					
Beranda Device History Lapor Engineer Data Device Tips							
Data Device History							
Berikut ini data device history yang pernah running lancar							
Tambah Data Baru							
No	Device	Package	Tester	Handler	Loadboard	Temperature	Pilihan
1	Z86E1631VSG-AN	20L PDIP	J750-2	ZL-KUW-3	ZL-LB-016A	75	Hapus
2	Z86E1631VSG-AN	80L POFP	J750-4	SYNAX-3	ZL-LB-019A	105	Hapus
FOOTER							

Gambar 3.40 Desain Halaman Data *Device History*
(Sumber: Data Penelitian, 2016)

7. Desain halaman Lapor *Engineer*

LOGO		HEADER	
Beranda Device History Lapor Engineer Data Device Tips			
Lapor Engineer			
Isi formulir di bawah ini untuk membuat laporan kepada Test Engineer			
Device name	Ketik device name		
Package	Contoh : 44L PLCC		
Tester	Pilih ▼		
Loadboard	Contoh : ZL-LB-009A		
Temperature	Contoh : 75		
Fail name	Contoh : functional fail		
Fail category	Pilih ▼		
Datalog (file.txt)	Browse... no file selected		
Batal		Kirim	
FOOTER			

Gambar 3.41 Desain Halaman Lapor *Engineer*
(Sumber: Data Penelitian, 2016)

Gambar 3.41 merupakan desain halaman Laporan *Engineer*. Melalui halaman ini *user* dapat melaporkan permasalahan tes korelasi yang belum bisa dideteksi oleh sistem pakar. *User* harus mengisi formulir untuk membuat laporan.

8. Desain halaman Tips

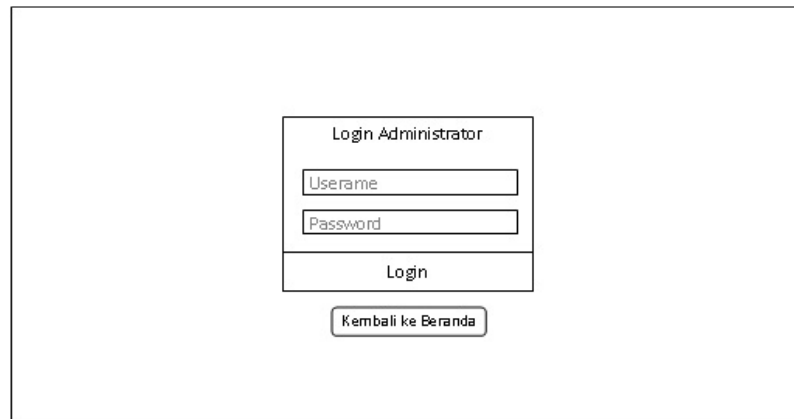
	HEADER
Beranda Device History Lapor Engineer Data Device Tips	
Tips Mengatasi Permasalahan Tes Korelasi Berikut ini tips mengatasi permasalahan tes korelasi 1. 2. 3. 4.	
FOOTER	

Gambar 3.42 Desain Halaman Tips
(Sumber: Data Penelitian, 2016)

Gambar 3.42 merupakan desain halaman Tips. Halaman ini menampilkan berbagai tips untuk mengatasi permasalahan pada tes korelasi. Tips ini diambil dari *Standard Operating Procedure (SOP)* yang berlaku di PT Unisem.

9. Desain halaman Login

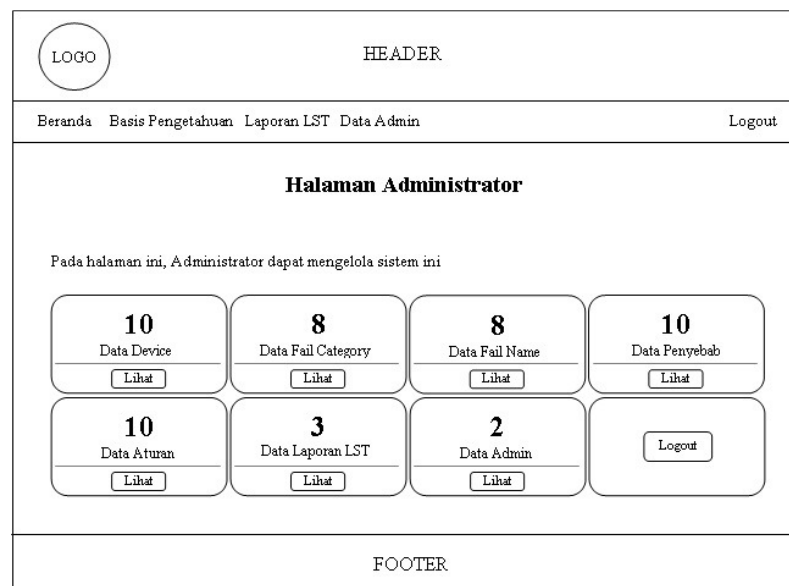
Gambar 3.43 merupakan desain halaman *login*. Halaman ini menampilkan formulir *login* untuk masuk ke halaman *Administrator*.



The image shows a login form titled "Login Administrator". It contains two input fields: "Username" and "Password". Below these fields is a "Login" button. At the bottom of the form is a button labeled "Kembali ke Beranda" (Return to Home).

Gambar 3.43 Desain Halaman *Login*
(Sumber: Data Penelitian, 2016)

10. Desain halaman Administrator



The image shows the design of the Administrator dashboard. It features a header with a "LOGO" placeholder and the word "HEADER". Below the header is a navigation bar with links: "Beranda", "Basis Pengetahuan", "Laporan LST", "Data Admin", and "Logout". The main content area is titled "Halaman Administrator" and includes a description: "Pada halaman ini, Administrator dapat mengelola sistem ini". Below this description are seven data cards arranged in two rows. Each card displays a count, a label, and a "Lihat" button. The cards are: Data Device (10), Data Fail Category (8), Data Fail Name (8), Data Penyebab (10), Data Aturan (10), Data Laporan LST (3), and Data Admin (2). A "Logout" button is located in the bottom right card. The footer area is labeled "FOOTER".

Gambar 3.44 Desain Halaman Awal *Administrator*
(Sumber: Data Penelitian, 2016)

Gambar 3.44 merupakan desain halaman *Administrator*. Pada halaman ini, *administrator* dapat mengelola sistem pakar. Tampilan utama halaman *Administrator* menampilkan data-data yang tersimpan pada sistem pakar seperti data *device*, data penyebab, data aturan dan sebagainya.

11. Desain halaman Data *Device*

LOGO		HEADER	
Beranda	Basis Pengetahuan	Laporan LST	Data Admin Logout
Data Device			
Berikut ini data device yang terdaftar pada sistem ini			
Tambah Data Baru			
No	Device ID	Device Name	Pilihan
1	DEV1	Z86E1631VSG-AN	Ubah Hapus
2	DEV2	Z86E1631VSG-AN	Ubah Hapus
FOOTER			

Gambar 3.45 Desain Halaman Data *Device*
(Sumber: Data Penelitian, 2016)

Gambar 3.45 merupakan desain halaman Data *Device*. Pada halaman ini *administrator* dapat mengelola data *device* yang tersimpan pada sistem pakar.

12. Desain halaman Data *Fail Category*

LOGO		HEADER	
Beranda	Basis Pengetahuan	Laporan LST	Data Admin Logout
Data Fail Category			
Berikut ini data fail category yang terdaftar pada sistem ini			
Tambah Data Baru			
No	Failure ID	Failure Name / Category	Pilihan
1	FC2	BIN-2	Ubah Hapus
2	FC3	BIN-3	Ubah Hapus
FOOTER			

Gambar 3.46 Desain Halaman Data *Fail Category*
(Sumber: Data Penelitian, 2016)

Gambar 3.46 adalah desain halaman Data *Fail Category*. Pada halaman ini *administrator* dapat mengelola data *fail category* yang ada pada sistem pakar.

13. Desain halaman Data *Fail Name*

LOGO		HEADER	
Beranda	Basis Pengetahuan	Laporan LST	Data Admin Logout
Data Fail Name			
Berikut ini data fail name yang terdaftar pada sistem ini			
Tambah Data Baru			
No	Failure ID	Failure Name / Category	Pilihan
1	FN4	ICC FAIL	Ubah Hapus
2	FN5	GROSS FAIL	Ubah Hapus
FOOTER			

Gambar 3.47 Desain Halaman Data *Fail Name*
(Sumber: Data Penelitian, 2016)

Gambar 3.47 merupakan desain halaman Data *Fail Name*. Pada halaman ini *administrator* dapat mengelola data *fail name* yang ada pada sistem pakar.

14. Desain halaman Data Penyebab

LOGO		HEADER		
Beranda	Basis Pengetahuan	Laporan LST	Data Admin Logout	
Data Penyebab dan Solusi				
Berikut ini data penyebab yang terdaftar pada sistem ini				
No	ID	Detail Penyebab	Solusi	Pilihan
1	PYB1	Fail ini disebabkan oleh test program...	Gunakan test program yang telah...	Ubah
2	PYB2	Fail ini disebabkan oleh delay pada....	Atur delay pada test program menjadi...	Ubah
FOOTER				

Gambar 3.48 Desain Halaman Data Penyebab
(Sumber: Data Penelitian, 2016)

Gambar 3.48 merupakan desain halaman Data Penyebab. Pada halaman ini *administrator* dapat melihat dan mengubah data penyebab dan data solusi yang tersimpan pada sistem pakar.

15. Desain halaman Data Aturan

Gambar 3.49 merupakan desain halaman Data Aturan. Pada halaman ini *administrator* dapat mengelola data aturan yang tersimpan pada sistem pakar.

Logo: LOGO

HEADER

Beranda Basis Pengetahuan Laporan LST Data Admin Logout

Data Aturan

Berikut ini data aturan yang terdaftar pada sistem ini

Tambah Aturan Baru

No	Device	Fail Category	Fail Name	Penyebab	Pilihan
1	DEV1	FC2	FN2	PYB1	Lihat
2	DEV2	FC4	FN6	PYB2	Lihat

FOOTER

Gambar 3.49 Desain Halaman Data Aturan
(Sumber: Data Penelitian, 2016)

16. Desain halaman Laporan LST

Gambar 3.50 adalah desain halaman Data Laporan LST. Pada halaman ini *administrator* dapat mengelola laporan permasalahan tes korelasi dari teknisi.

LOGO		HEADER					
Beranda		Basis Pengetahuan Laporan LST Data Admin				Logout	
Data Laporan LST							
Berikut ini data laporan masalah tes korelasi dari LST							
No	Device	Tester	Loadboard	Fail category	Fail Name	Datalog	Pilihan
1	Z86E1631VSG-AN	J750-2	ZL-LB-016A	BIN-4	ICC FAIL	↓	✓ ✕
2	Z86E1631VSG-AN	J750-4	ZL-LB-019A	BIN-7	DC FAIL	↓	✓ ✕
FOOTER							

Gambar 3.50 Desain Halaman Data Laporan LST
(Sumber: Data Penelitian, 2016)

17. Desain halaman Data Admin

LOGO		HEADER	
Beranda		Basis Pengetahuan Laporan LST Data Admin	
		Logout	
Data Admin			
Berikut ini data admin yang terdaftar pada sistem ini			
Tambah Admin Baru			
No	Username	Password	Pilihan
1	admin	Ubah	Hapus
2	romli	Ubah	Hapus
FOOTER			

Gambar 3.51 Desain Halaman Data Admin
(Sumber: Data Penelitian, 2016)

Gambar 3.51 merupakan desain halaman Data Admin. Pada halaman ini administrator dapat mengelola data admin yang tersimpan pada sistem pakar.

3.4.5 Penulisan Kode Program

Desain yang telah dibuat harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahapan ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahapan desain (A.S. dan Shalahuddin, 2011: 27). Kode program sistem pakar dalam penelitian ini ditulis dengan bahasa pemrograman PHP untuk mengolah data. Sebagai pendukung, digunakan juga bahasa pemrograman HTML. Semua kode program ditulis menggunakan aplikasi Sublime Text. Untuk mengatur tampilan antarmuka pengguna, peneliti menggunakan *framework* Bootstrap. Kemudian *database* dibuat dengan bantuan aplikasi phpMyadmin. Sistem pakar yang dihasilkan dijalankan pada *server localhost* dengan menginstal aplikasi Xampp dan diakses menggunakan *browser* Mozilla Firefox.

3.5 Lokasi dan Jadwal Penelitian

3.5.1 Lokasi Penelitian

Penelitian ini dilakukan di PT Unisem yang beralamat di Jl. S. Parman Kav. 201 Batamindo Industrial Park kota Batam. Alasan peneliti memilih PT Unisem sebagai tempat penelitian adalah karena kemudahan mendapatkan data serta efisiensi biaya dan waktu.

3.5.2 Jadwal Penelitian

Setiap rancangan penelitian perlu dilengkapi dengan jadwal kegiatan yang akan dilaksanakan yang berisi jadwal kegiatan apa saja yang akan dilakukan selama penelitian (Sugiyono, 2014: 286). Penelitian dilaksanakan pada semester ganjil 2016/2017 yaitu antara bulan September 2016 sampai dengan bulan Februari 2017. Jadwal kegiatan penelitian selengkapnya dapat dilihat pada tabel 3.29 berikut.

Tabel 3.29 Jadwal Penelitian

KEGIATAN	WAKTU KEGIATAN																					
	Sep 2015				Okt 2015				Nov 2015				Des 2015				Jan 2016				Feb 2017	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Pengajuan Judul																						
Penyusunan BAB I																						
Penyusunan BAB II																						
Penyusunan BAB III																						
Penyusunan BAB IV																						
Penyusunan BAB V																						

Sumber: Data Penelitian (2016)