

**RANCANG BANGUN SISTEM INFORMASI
PEMERIKSAAN PRODUK KLAIM BATERAI AKI
PADA PT NIPRESS TBK DENGAN MENGGUNAKAN
J2SE DAN MYSQL 5.6.16**

TUGAS AKHIR

Untuk Memenuhi Sebagian Syarat Penyelesaian Jenjang Diploma Empat (D-4)
Program Studi Sistem Informasi Pada Politeknik STMI Jakarta

OLEH

VICKI WAHYU DIANTO

(1312018)



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2016**

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PEMERIKSAAN PRODUK KLAIM BATERAI AKI PADA
PT NIPRESS TBK DENGAN MENGGUNAKAN J2SE
DAN MYSQL 5.6.16

Disusun Oleh :
Nama : Vicki Wahyu Dianto
Nim : 1312018
Program Studi : Sistem Informasi

Telah Diuji Oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi
Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia pada hari Rabu
tanggal 2 November 2016.

Jakarta, 14 November 2016

Dosen Pembimbing

Ketua Penguji

Ulil Hamida, ST, MT
NIP. 198103272005022001

Noveriza Yuliasari, S.Si, MT
NIP. 197811212009012003

Dosen Penguji

Dosen Penguji

Dedy Trisanto, S.Kom. MMSI
NIP. 197805052005021002

Dr. Ridzky Kramanandita, S.Kom, MT
NIP. 197403022002121001

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PEMERIKSAAN PRODUK KLAIM BATERAI AKI PADA
PT NIPRESS TBK DENGAN MENGGUNAKAN J2SE
DAN MYSQL 5.6.16

Disusun Oleh :
Nama : Vicki Wahyu Dianto
Nim : 1312018
Program Studi : Sistem Informasi
Tanggal Seminar : 12 Oktober 2016
Tanggal Sidang : 2 November 2016
Tanggal Lulus : 2 November 2016

Jakarta, 14 November 2016

Menyetujui
Dosen Pembimbing

Ulil Hamida ST, MT
NIP. 198103272005022001

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PEMERIKSAAN PRODUK KLAIM BATERAI AKI PADA
PT NIPRESS TBK DENGAN MENGGUNAKAN J2SE
DAN MYSQL 5.6.16

Disusun Oleh :
Nama : Vicki Wahyu Dianto
Nim : 1312018
Program Studi : Sistem Informasi
Tanggal Seminar : 12 Oktober 2016
Tanggal Sidang : 2 November 2016
Tanggal Lulus : 2 November 2016

Jakarta, 14 November 2016

Menyetujui
Asisten Dosen Pembimbing

Ahmad Juniar, S.Kom, MT
NIP. 197906052006041002

LEMBAR PERSETUJUAN DOSEN PEMBIMBING
POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

TANDA PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PEMERIKSAAN PRODUK KLAIM BATERAI AKI
PADA PT NIPRESS TBK DENGAN
MENGUNAKAN J2SE DAN MYSQL 5.6.16

Disusun Oleh

Nama : Vicki Wahyu Dianto
Nim : 1312018
Program Studi : Sistem Informasi

Telah diperiksa dan disetujui untuk diajukan dalam sidang Tugas Akhir Sekolah Tinggi
Manajemen Industri.

Jakarta, 14 November 2016

Dosen Pembimbing

Ulil Hamida, ST, MT
NIP. 198103272005022001

**LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING
POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

TANDA PERSETUJUAN ASISTEN DOSEN PEMBIMBING

Judul Tugas Akhir : RANCANG BANGUN SISTEM INFORMASI
PEMERIKSAAN PRODUK KLAIM BATERAI AKI
PADA PT NIPRESS TBK DENGAN
MENGUNAKAN J2SE DAN MYSQL 5.6.16

Disusun Oleh

Nama : Vicki Wahyu Dianto
Nim : 1312018
Program Studi : Sistem Informasi

Telah diperiksa dan disetujui untuk diajukan dalam sidang Tugas Akhir Sekolah Tinggi
Manajemen Industri.

Jakarta, 14 November 2016
Asisten Dosen Pembimbing

Ahmad Juniar, S.Kom, MT
NIP. 197906052006041002

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Vicki Wahyu Dianto

Nim : 1312018

Berstatus sebagai mahasiswa Program Studi Sistem Informasi di Politeknik STMI Jakarta Kementerian Perindustrian RI. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul:

**“RANCANG BANGUN SISTEM INFORMASI PEMERIKSAAN PRODUK
KLAIM BATERAI AKI PADA PT NIPRESS TBK DENGAN MENGGUNAKAN
J2SE DAN MYSQL 5.6.16”**

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei lapangan, dibantu oleh dosen pembimbing maupun asisten dosen pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya Tugas Akhir ini.
- **Bukan** merupakan hasil duplikasi karya tulis yang sudah dipublikasikan atau yang pernah dipakai sebelumnya untuk mendapatkan gelar sarjana di Universitas/Perguruan Tinggi lain, kecuali yang telah disebutkan sumbernya dan dicantumkan pada referensi karya Tugas Akhir ini.
- **Bukan** merupakan karya tulis hasil terjemahan dari kumpulan buku atau jurnal acuan yang tertera dalam referensi karya Tugas Akhir ini.

Jika terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka saya bersedia karya Tugas Akhir ini dibatalkan.

Jakarta, 14 November 2016

Yang Membuat Pernyataan,

Vicki Wahyu Dianto

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Vicki Wahyu Dianto

NIM : 1312018

Judul Tugas Akhir : Rancang Bangun Sistem Informasi Pemeriksaan Produk Klaim
Baterai Aki

5.6.16 : Pada PT Nipress Tbk Dengan Menggunakan J2SE Dan MySQL

Pembimbing : Ulil Hamida ST, MT

Asisten Pembimbing : Ahmad Juniar, S.Kom, MT

Tanggal	BAB	Keterangan	Paraf
11-03-2016	I	Revisi pendahuluan	
14-03-2016	I	Revisi dan penyerahan BAB II	
18-03-2016	II	Koreksi Bab II, penyerahan revisi Bab I	
21-03-2016	II	Tata letak banyak yang salah	
01-04-2016	III	Perubahan kerangka pemecahan masalah	
04-04-2016	II & III	Revisi Bab II dan III	
13-04-2016	IV	Revisi Bab IV	
18-04-2016	IV	Penambahan bahan pengumpulan data	
20-04-2016	IV	Cek akhir Bab IV	
22-04-2016	V	Revisi flowchart, usecase, class diagram usulan	
09-05-2016	V	Flowmaap dan usecase direvisi	
12-05-2016	V	Revisi Activity diagram	
30-09-2016	V & VI	Program, kesimpulan dan saran	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Pembimbing

Drs. Jacob Saragih, MM
NIP. 195404281986031002

Ulil Hamida ST, MT
NIP. 198103272005022001

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Vicki Wahyu Dianto
NIM : 1312018
Judul Tugas Akhir : Rancang Bangun Sistem Informasi Pemeriksaan Produk Klaim
Baterai Aki Pada PT Nipress Tbk Dengan Menggunakan J2SE
Dan MySQL 5.6.16
Pembimbing : Ulil Hamida ST, MT
Asisten Pembimbing : Ahmad Juniar, S.Kom, MT

Tanggal	BAB	Keterangan	Paraf
06-04-2016	I	Cek BAB I	
11-04-2016	II	Cek BAB II dan Revisi BAB I	
19-04-2016	III	Cek BAB III dan Revisi BAB II	
31-04-2016	IV	Cek BAB IV, perbaikan halaman 52 & 62	
24-06-2016	V	Cek BAB V, pemeriksaan sampai UseCase	
22-06-2016	V	Cek BAB V bagian activity, sequence, dan halaman 89	
10-08-2016	V	Perbaikan class diagram dan HIPO	
24-08-2016	V	Perbaikan flowchart dan program	
19-09-2016	V	Program dan rancangan interface harus sama	
04-10-2016	VI	Cek cover, daftar isi dan daftar pustaka	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Asisten Pembimbing

Drs. Jacob Saragih, MM
NIP. 195404281986031002

Ahmad Juniar, S.Kom, MT
NIP. 197906052006041002

ABSTRAK

PT Nipress Tbk merupakan perusahaan yang bergerak di bidang industri manufaktur yang berfokus dalam produksi baterai aki. PT Nipress Tbk membutuhkan pengembangan sebuah sistem informasi klaim produk baterai aki mengenai pengolahan data dan pelaporan kerusakan baterai. Sistem informasi klaim produk baterai aki yang ada pada PT Nipress Tbk dalam pengolahan datanya masih kurang maksimal karena menggunakan *form* manual sehingga banyak sekali resiko kehilangan, rusak atau rangkapnya data serta lamanya proses pelaporan yang dibuat. Rancang bangun sistem informasi pemeriksaan produk klaim baterai aki sangat diperlukan dalam perusahaan untuk melakukan proses pengolahan data klaim produk. Sistem informasi yang diusulkan akan mempermudah dan mengintegrasikan proses pengolahan data sehingga membantu bagian di dalam sistem untuk saling bertukar informasi dan mengambil keputusan dengan cepat. Pengembangan sistem informasi klaim produk baterai aki menggunakan metode prototipe evolusioner. Pemodelan sistem dan data menggunakan *unified modeling language* (UML) dan *flowchart*. Perancangan aplikasi sistem informasi penerimaan klaim produk baterai aki ini menggunakan *hierarchy plus input-process-output* (HIPO), *flowchart* program dan perancangan tampilan antarmuka. Pembuatan sistem informasi penerimaan klaim produk baterai aki menggunakan J2SE dan MySQL 5.6.16. Sistem informasi klaim produk baterai aki diharapkan dapat memudahkan perusahaan dalam proses pengolahan data terutama dalam pengolahan laporan kerusakan baterai bulanan dan memudahkan dalam menyimpan data pada bagian *Claim Handling Sub Section*. Untuk penerapan sistem baru, disarankan untuk melakukan sosialisasi kepada bagian terkait dan pemeliharaan aplikasi agar dapat berjalan dengan baik.

Kata Kunci : Sistem informasi, klaim produk baterai aki, pelaporan kerusakan bulanan, prototipe evolusioner, J2SE, MySQL 5.6.16, UML.

KATA PENGANTAR

Puji syukur penulis panjatkan bagi Tuhan Yang Maha Esa, yang atas berkat dan kasih-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Pada PT Nipress Tbk Dengan Menggunakan J2SE Dan Mysql 5.6.16”**.

Penulisan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian jenjang Diploma Empat (D-4) program studi Sistem Informasi pada Politeknik STMI Jakarta.

Terselesainya tugas akhir ini tidak terlepas dari bantuan moril maupun materil, bimbingan, dan petunjuk dari berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih yang sebesar-besarnya dan penghargaan kepada:

1. ALLAH Subhanahu Wa Ta'ala Tuhan Yang Maha Esa.
2. Bapak Dr. Mustofa, MT. selaku Direktur Politeknik STMI Jakarta.
3. Bapak Drs. Jacob Saragih, MM. selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
4. Ibu Ulil Hamida, ST, MT. selaku dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
5. Bapak Ahmad Juniar, S.Kom, MT. selaku asisten dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
6. Bapak Khosim Mardiyanto dan Ibu Lasih Al Sulasih selaku orang tua penulis yang selalu mendoakan, memberikan motivasi dan pengorbanannya baik segi moril, materi kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
7. Gilang Putra Haryoko dan Zulfa Zakiyah selaku teman baik penulis yang tidak segan memberikan kritik dan saran serta bantuan moril dan materil kepada penulis.
8. Bustomi Raharjo, Hikmah Jameela, Nur Wahyudi, Dimas Viannur Rahman dan kawan-kawan kelompok *Unknown* yang selalu mendukung, memberikan nasihat dan mau membagi pengalamannya untuk penulis.

9. Bapak Maman selaku pembimbing di PT Nipress Tbk. Ibu Yunita, Bapak Moko, Bapak Puji serta seluruh pekerja khususnya di bagian *Claim Handling Sub Section* yang telah membimbing dan memberikan informasi yang berguna kepada penulis dalam penyusunan Tugas Akhir.
10. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
11. Seluruh teman-teman jurusan Sistem Informasi angkatan 2012 yang telah memberikan semangat dan motivasi kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
12. Serta semua pihak yang telah membantu penulis dan yang tidak dapat disebutkan satu per satu.

Semoga Tuhan Yang Maha Esa membalas kebaikan dan ketulusan semua pihak yang telah membantu dan penulis berharap semoga Tugas Akhir ini dapat memberikan manfaat dan informasi yang berguna bagi pembaca. Terima kasih.

Jakarta, 14 November 2016

Penulis,
Vicki Wahyu Dianto

DAFTAR ISI

	Halaman
HALAMAN JUDUL	
LEMBAR PENGESAHAN	
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	
LEMBAR PERSETUJUAN ASISTEN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN ASISTEN DOSEN PEMBIMBING	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Konsep Dasar Sistem	6
2.1.1 Pengertian Sistem	6
2.1.1.1 Pendekatan Prosedur	7
2.1.1.2 Pendekatan Komponen	8
2.1.2 Karakteristik Sistem	8
2.1.3 Elemen Sistem	10
2.2 Konsep Dasar Informasi	12
2.2.1 Siklus Informasi	13

2.2.2	Jenis-Jenis Informasi	14
2.3	Konsep Dasar Sistem Informasi	15
2.4	Siklus Pengembangan Perangkat Lunak (SDLC)	17
2.4.1	Tahap SDLC	18
2.4.2	Prototipe	19
2.4.2.1	Prototipe Evolusioner	19
2.4.2.2	Prototipe <i>Requirement</i>	20
2.5	Pengertian Klaim	21
2.6	Pengertian Produk	22
2.6.1	Produk Konsumsi (<i>Consumer Products</i>)	22
2.6.2	Produk Industri (<i>Business Product</i>)	23
2.7	Pengertian Baterai Aki	23
2.8	<i>Flowchart</i>	23
2.9	<i>Unified Modeling Language (UML)</i>	26
2.9.1	<i>Use Case Diagram</i>	27
2.9.2	<i>Activity Diagram</i>	28
2.9.3	<i>Sequence Diagram</i>	28
2.9.4	<i>Class Diagram</i>	30
2.9.5	<i>Component Diagram</i>	31
2.9.6	<i>Deployment Diagram</i>	32
2.10	Kamus Data	33
2.11	<i>Hierarchy plus Input-Proses-Output (HIPO)</i>	34
2.12	<i>Java</i>	35
2.12.1	Netbeans	36
2.12.2	Fitur-Fitur Netbeans	36
2.13	MySQL	37
BAB III METODOLOGI PENELITIAN		42
3.1	Metodologi Penelitian	42
3.2	Jenis dan Sumber Data	42
3.3	Metode Pengumpulan Data	43
3.4	Metode Pengembangan Sistem	44

3.5	Kerangka Pemecahan Masalah	45
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	49
4.1	Sekilas Perusahaan	49
4.2	Visi Misi dan Filosofi Perusahaan	50
4.3	Struktur Organisasi Perusahaan	51
4.4	Peran dan Fungsi Departemen <i>Quality Assurance</i>	53
4.5	Tugas dan Tanggung Jawab <i>Claim Handling</i> dan <i>Quality Assurance Section</i>	53
4.6	Produk Perusahaan	54
4.7	Komponen Baterai dan Fungsinya	54
4.8	Proses Produksi Baterai Aki	56
4.9	Jenis-Jenis Kerusakan Baterai Aki	57
4.10	Dokumen Proses Klaim	59
4.11	Bentuk Informasi Klaim Baterai	63
4.12	Prosedur Klaim	65
4.13	Sistem Berjalan Dengan <i>Use Case Diagram</i>	68
BAB V	ANALISIS DAN PEMBAHASAN	73
5.1	Analisis Kebutuhan Sistem	73
5.2	Prosedur Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan	74
5.3	Analisis dan Perancangan Sistem Usulan	76
5.3.1	<i>Use Case Diagram</i>	76
5.3.2	<i>Activity Diagram</i>	88
5.3.3	<i>Sequence Diagram</i>	98
5.3.4	<i>Class Diagram</i>	108
5.3.5	Kamus Data	109
5.3.5.1	Rincian Elemen	115
5.3.5.2	Perhitungan <i>Charge</i>	117
5.3.6	<i>Deployment Diagram</i>	119
5.4	Analisis Desain Program	120
5.4.1	HIPO	120

5.4.2	<i>Flowchart</i> Program	121
5.4.3	Perancangan <i>Interface</i> Program	122
5.5	Implementasi Sistem <i>Software</i> dan <i>Hardware</i>	130
BAB VI	KESIMPULAN DAN SARAN	132
6.1	Kesimpulan	132
6.2	Saran	132
DAFTAR PUSTAKA	133

DAFTAR GAMBAR

	Halaman
Gambar II.1	Karakteristik Sistem 10
Gambar II.2	Hubungan Elemen Sistem 12
Gambar II.3	Pemrosesan Data Menjadi Informasi 13
Gambar II.4	Siklus Informasi 13
Gambar II.5	Interaksi Blok Sistem Informasi 16
Gambar II.6	Siklus Hidup Sistem 17
Gambar II.7	Pengembangan Prototipe Evolusioner 20
Gambar II.8	Pengembangan Prototipe <i>Requirement</i> 21
Gambar II.9	Diagram UML 26
Gambar II.10	<i>Visual Table Of Contents</i> 35
Gambar III.1	Kerangka Pemecahan Masalah 45
Gambar IV.1	Struktur Organisasi PT Nipress Tbk 52
Gambar IV.2	Struktur Organisasi <i>Quality Assurance Departement</i> 52
Gambar IV.3	Struktur Baterai Aki 55
Gambar IV.4	Alur Proses Produksi Baterai Aki 56
Gambar IV.5	Contoh Klaim Dengan Email 59
Gambar IV.6	Memo <i>External</i> 60
Gambar IV.7	Lembar Penerimaan Barang 61
Gambar IV.8	Lembar Analisis Kondisi Baterai 62
Gambar IV.9	Grafik Fenomena Kerusakan Baterai (<i>Customer Fault</i>) 63
Gambar IV.10	Grafik Fenomena Kerusakan Baterai (<i>Factory Fault</i>) 64
Gambar IV.11	Aliran Dokumen Sistem yang Berjalan 67
Gambar IV.12	<i>Use Case</i> Sistem yang Berjalan 69
Gambar V.1	<i>Flowmap</i> Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan 75
Gambar V.2	<i>Use Case Diagram</i> Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan..... 77
Gambar V.3	<i>Activity Diagram Login</i> 88

Gambar V.4	<i>Activity Diagram</i> Mengelola Data <i>User</i>	89
Gambar V.5	<i>Activity Diagram</i> Mengelola Data Produk.....	90
Gambar V.6	<i>Activity Diagram</i> Mengelola Data Kerusakan Produk.....	91
Gambar V.7	<i>Activity Diagram</i> Mengelola <i>Form</i> LPB.....	92
Gambar V.8	<i>Activity Diagram</i> Mengelola Informasi Produk Klaim.....	93
Gambar V.9	<i>Activity Diagram</i> Mengelola Informasi Pemeriksaan Awal Baterai	94
Gambar V.10	<i>Activity Diagram</i> Mengelola Informasi Tes Kemampuan Baterai	95
Gambar V.11	<i>Activity Diagram</i> Mengelola Informasi Data Bedah Baterai	96
Gambar V.12	<i>Activity Diagram</i> Validasi.....	97
Gambar V.13	<i>Activity Diagram</i> View dan Mencetak Laporan AKB	98
Gambar V.14	<i>Sequence Diagram</i> Proses <i>Login</i>	99
Gambar V.15	<i>Sequence Diagram</i> Mengelola Data <i>User</i>	99
Gambar V.16	<i>Sequence Diagram</i> Mengelola Data Produk	100
Gambar V.17	<i>Sequence Diagram</i> Mengelola Data Kerusakan	101
Gambar V.18	<i>Sequence Diagram</i> Mengelola <i>Form</i> LPB	102
Gambar V.19	<i>Sequence Diagram</i> Mengelola Informasi Produk Klaim	103
Gambar V.20	<i>Sequence Diagram</i> Mengelola Informasi Pemeriksaan Awal Baterai	104
Gambar V.21	<i>Sequence Diagram</i> Mengelola Informasi Tes Kemampuan Baterai	105
Gambar V.22	<i>Sequence Diagram</i> Mengelola Informasi Data Bedah Baterai	106
Gambar V.23	<i>Sequence Diagram</i> Validasi	107
Gambar V.24	<i>Sequence Diagram</i> View Dan Mencetak Laporan Kerusakan Baterai	108
Gambar V.25	<i>Class Diagram</i> Usulan	109
Gambar V.26	<i>Deployment Diagram</i>	119
Gambar V.27	HIPO Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan	121

Gambar V.28	<i>Program Logic Flowchart Aplikasi Sistem Usulan.....</i>	122
Gambar V.29	<i>Interface Form Login</i>	123
Gambar V.30	<i>Interface Form Menu Utama</i>	123
Gambar V.31	<i>Interface Form Data User</i>	124
Gambar V.32	<i>Interface Form Data Produk</i>	124
Gambar V.33	<i>Interface Form Data Kerusakan</i>	125
Gambar V.34	<i>Interface Form Laporan Penerimaan Barang</i>	125
Gambar V.35	<i>Interface Form Detil Laporan Penerimaan Barang</i>	126
Gambar V.36	<i>Interface Form Pemeriksaan Awal Tab Info Produk Klaim.....</i>	127
Gambar V.37	<i>Interface Form Pemeriksaan Awal Tab Cek Tampilan Luar</i>	127
Gambar V.38	<i>Interface Form Pasca Charge dan Load Test</i>	128
Gambar V.39	<i>Interface Form Bedah Baterai</i>	128
Gambar V.40	<i>Interface Form Validasi</i>	129
Gambar V.41	<i>Interface Form Pilih Tanggal & Laporan Penerimaan Baterai</i>	130
Gambar V.42	<i>Interface Form Pilih Tanggal & Laporan Kerusakan Baterai</i>	130

DAFTAR TABEL

	Halaman
Tabel II.1	Simbol-Simbol <i>Flowchart</i> Menurut Sitorus 24
Tabel II.2	Simbol-Simbol <i>Flowchart</i> Menurut Jogiyanto 25
Tabel II.3	Simbol-Simbol <i>Use Case Diagram</i> 27
Tabel II.4	Simbol-Simbol <i>Activity Diagram</i> 28
Tabel II.5	Simbol-Simbol <i>Sequence Diagram</i> 29
Tabel II.6	Simbol-Simbol <i>Class Diagram</i> 30
Tabel II.7	Notasi <i>Multiplicity</i> 31
Tabel II.8	Simbol-Simbol <i>Component Diagram</i> 32
Tabel II.9	Simbol-Simbol <i>Deployment Diagram</i> 33
Tabel II.10	Contoh Kamus Data Untuk Tabel Barang 34
Tabel II.11	Tipe Data Numerik <i>Integer</i> 39
Tabel II.12	Tipe Data Numerik <i>Floating Point</i> 39
Tabel II.13	Tipe Data <i>String Text/Blob</i> 40
Tabel II.14	Tipe Data <i>String</i> Selain <i>Text/Blob</i> 40
Tabel II.15	Tipe Data Tanggal dan Waktu 41
Tabel IV.1	Komponen Baterai Aki dan Fungsinya 55
Tabel IV.2	Kerusakan Akibat Kesalahan Pabrik 57
Tabel IV.3	Kerusakan Akibat Kesalahan <i>Customer</i> 58
Tabel IV.4	Definisi Aktor Proses Klaim 70
Tabel IV.5	Definisi <i>Use Case</i> Proses Klaim 71
Tabel V.1	Kebutuhan Sistem Informasi Klaim Pemeriksaan Produk Klaim Baterai Aki 73
Tabel V.2	Definisi Aktor <i>Use Case Diagram</i> Sistem Usulan 78
Tabel V.3	Definisi <i>Use Case Diagram</i> Sistem Usulan..... 78
Tabel V.4	Skenario <i>Use Case Login</i> 80
Tabel V.5	Skenario <i>Use Case</i> Mengelola Data <i>User</i> 80
Tabel V.6	Skenario <i>Use Case</i> Mengelola Data Produk 81

Tabel V.7	Skenario <i>Use Case</i> Mengelola Data Kerusakan	81
Tabel V.8	Skenario <i>Use Case</i> Mengelola <i>Form</i> LPB	82
Tabel V.9	Skenario <i>Use Case</i> Mengelola Informasi Produk Klaim	82
Tabel V.10	Skenario <i>Use Case</i> Melakukan Pemeriksaan Awal Baterai	83
Tabel V.11	Skenario <i>Use Case</i> Mengelola Informasi Pemeriksaan Awal Baterai	84
Tabel V.12	Skenario <i>Use Case</i> Melakukan Tes Kemampuan Baterai	84
Tabel V.13	Skenario <i>Use Case</i> Mengelola Informasi Tes Kemampuan Baterai	85
Tabel V.14	Skenario <i>Use Case</i> Melakukan Pembedahan Baterai	85
Tabel V.15	Skenario <i>Use Case</i> Mengelola Informasi Data Bedah Baterai	86
Tabel V.16	Skenario <i>Use Case</i> Validasi	86
Tabel V.17	Skenario <i>Use Case</i> View Laporan Kerusakan Baterai	87
Tabel V.18	Skenario <i>Use Case</i> Mencetak Laporan Kerusakan Baterai	88
Tabel V.19	Tabel <i>User</i>	110
Tabel V.20	Tabel Produk.....	110
Tabel V.21	Tabel Kerusakan	111
Tabel V.22	Tabel LPB	111
Tabel V.23	Tabel Detil LPB	111
Tabel V.24	Tabel Info Produk Klaim	112
Tabel V.25	Tabel Pemeriksaan Awal Baterai.....	112
Tabel V.26	Tabel Data <i>Cell</i> Baterai.....	113
Tabel V.27	Tabel <i>Load Test</i>	113
Tabel V.28	Tabel Bedah Baterai.....	114
Tabel V.29	Tabel Detil Kerusakan Produk.....	114
Tabel V.30	Tabel Validasi	115
Tabel V.31	Tabel Rincian Elemen.....	115
Tabel V.32	Tabel <i>Level Charge</i>	118

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era industri yang semakin maju, persaingan di bidang tersebut semakin ketat seiring meningkatnya kebutuhan masyarakat terhadap sejumlah produk barang dan jasa. Hal ini mendorong suatu organisasi atau perusahaan untuk lebih kreatif, inovatif dan efisien dalam menghasilkan produk yang berkualitas tinggi serta membangun kepercayaan masyarakat terhadap produk yang dihasilkan. Untuk menciptakan produk yang berkualitas tinggi, perusahaan harus memperhatikan proses produksi yang sedang berjalan hingga menjadi produk yang mempunyai nilai jual dan sampai ke tangan konsumen. Sementara itu, untuk membangun kepercayaan masyarakat terhadap suatu produk, perusahaan harus memperhatikan metode pemasaran hingga kebijakan pemberian garansi pada setiap produk.

Bagi suatu perusahaan, memberikan garansi pada produk merupakan suatu bentuk kebijakan yang penting, karena kegiatan tersebut pada kenyataannya tidak saja memberikan manfaat kepada konsumen tetapi juga kepada produsen. Bagi konsumen, garansi melindungi dari pembelian produk yang cacat. Sementara itu, bagi produsen garansi membatasi klaim yang tidak rasional dari konsumen. Di samping itu, produsen juga dapat memanfaatkan garansi sebagai alat promosi yang efektif untuk produknya, serta dapat membangun suatu kepercayaan masyarakat terhadap produk yang dihasilkan.

PT Nipress Tbk yaitu sebuah perusahaan industri yang bergerak di bidang manufaktur pembuatan baterai aki. Dalam industri ini, jenis baterai yang diproduksi meliputi AMB (*Auto Mobile Battery*), MCB (*Motor Cycle Battery*) dan *Industry Battery*. Adapun baterai aki merupakan jenis baterai sekunder yang dapat digunakan berulang-ulang dengan mengubah energi kimia menjadi energi listrik melalui reaksi kimia kelistrikan.

Pada saat ini sistem informasi klaim baterai PT Nipress Tbk masih ditulis di atas *form* kertas sehingga memunculkan berbagai permasalahan, seperti data produk klaim yang tercatat ulang, *form* rusak akibat keteledoran karyawan, *form* yang hilang sehingga

harus mencatat ulang, penghitungan masa *charge* baterai yang terpisah dan merepotkan, pencarian data klaim yang memakan waktu lama, dan terkadang terjadi salah tanggap antara karyawan akibat tulisan yang kurang jelas. Permasalahan yang lain juga muncul karena pembuatan laporan klaim yang sulit, bahkan terkadang laporan tidak dibuat sehingga bagian klaim hanya menggunakan *form* yang ada sebagai bahan laporan.

Berdasarkan pemaparan di atas, maka dibuatlah Tugas Akhir dengan judul “RANCANG BANGUN SISTEM INFORMASI PEMERIKSAAN PRODUK KLAIM BATERAI AKI PADA PT NIPRESS TBK DENGAN MENGGUNAKAN J2SE DAN MYSQL 5.6.16”. Aplikasi yang dihasilkan diharapkan menjadi alat untuk membantu bagian *Claim Handling Sub Section* di *Departement Quality Assurance* agar dapat meningkatkan kinerja kerja.

1.2 Rumusan Masalah

Adapun rumusan masalah yang didapat adalah sebagai berikut:

1. Sistem informasi pemeriksaan produk klaim baterai aki di perusahaan tersebut masih manual dengan cara ditulis di atas lembar *form*. Hal itu menyebabkan proses klaim menjadi memakan waktu lama dan menimbulkan beberapa masalah, seperti *form* rusak, *form* hilang, sulit mengolah data, terjadi kerangkapan data dan pencarian data yang memakan waktu lama.
2. Proses pembuatan laporan pemeriksaan produk klaim yang sulit, bahkan terkadang laporan tidak dibuat sehingga pada bagian klaim hanya menggunakan *form* yang sudah ada sebagai bahan presentasi untuk menyampaikan informasi. Laporan tersebut dinilai kurang informatif, karena akan menyulitkan bagi yang membutuhkan informasi dalam menentukan sebuah keputusan.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah merancang dan membangun sistem informasi untuk:

1. Membantu proses pemeriksaan produk klaim baterai yang lebih cepat, mudah dan akurat dengan menggunakan MySQL sebagai pengolah *database*.

2. Memfasilitasi pembuatan laporan pemeriksaan produk klaim baterai sehingga memudahkan bagi *Quality Assurance Section Head*.

1.4 Batasan Masalah

Agar dalam penulisan Tugas Akhir ini lebih fokus dan terarah, maka perlu diberikan batasan-batasan di antaranya adalah sebagai berikut:

1. Analisis dan penelitian yang dilakukan hanya sebatas menangani proses pemeriksaan produk klaim hingga menghasilkan laporan.
2. Tidak membahas laporan akhir yang berbentuk *Investigation and Countermeasure Report* karena akar permasalahan tersebut tidak bisa diputuskan oleh *Departement Quality Assurance* sendiri.
3. Penelitian dilakukan di PT Nipress Tbk dengan masa penelitian selama dua bulan yang terhitung pada tanggal 3 Agustus 2015 s.d. 2 Oktober 2015.

1.5 Manfaat Penelitian

Adapun manfaat yang dapat diambil dari penelitian pada Tugas Akhir ini adalah sebagai berikut:

1. Bagi penulis
 - a. Untuk memberikan pengalaman kepada penulis dalam menganalisis suatu sistem dan diharapkan dapat memberikan suatu solusi permasalahan.
 - b. Untuk memberikan kemampuan dalam menggambarkan suatu literatur sistem secara jelas terhadap masalah yang diamati.
2. Bagi institusi pendidikan
 - a. Menjalinkan kerja sama dalam rangka pengembangan antara dunia pendidikan dan dunia kerja.
 - b. Untuk menghasilkan lulusan yang sesuai dengan kebutuhan dunia kerja.
3. Bagi perusahaan
 - a. Memberikan partisipasi dalam pengembangan di dunia pendidikan.

- b. Mendapatkan saran yang diharapkan mampu menyelesaikan permasalahan yang terjadi.

1.6 Sistematika Penulisan

Laporan ini disusun berdasarkan hal-hal yang berhubungan erat dengan hasil pengamatan sehingga dapat memperoleh gambaran yang lebih jelas mengenai isi laporan dengan praktik kerja lapangan yang dilaksanakan. Adapun tahapan-tahapan dalam laporan ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat latar belakang, permasalahan, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan laporan.

BAB II LANDASAN TEORI

Bab ini membahas tentang pengertian sistem, analisis sistem, jenis sistem, metode pengembangan sistem informasi, informasi, sistem informasi, klaim, produk, baterai aki, *flowchart*, dan *Unified Modeling Language (UML)* sebagai alat bantu untuk membuat rancangan konseptual. Selain itu, penulis akan menggunakan J2SE dengan editor Netbeans 8.0.2 dan MySQL 5.6.16 sebagai alat bantu untuk membuat implementasinya.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang jenis dan sumber data, metode pengumpulan data serta langkah-langkah yang akan dilakukan dalam perumusan dan pemecahan masalah termasuk metodologi pengembangan sistem yang digunakan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini akan menguraikan tentang hasil pengamatan dalam kerja lapangan yang telah dilakukan, seperti fungsi sistem klaim baterai, alur proses sistem klaim, alur proses pemeriksaan baterai, dan transaksi-transaksi yang terjadi di sistem klaim baterai perusahaan.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari identifikasi kebutuhan sistem, perancangan basis data, perancangan tampilan layar, perancangan sistem dengan menggunakan UML yang meliputi *structure diagrams* (*class diagram* dan *deployment diagram*), *behavior diagrams* (*use case diagram* dan *activity diagram*) dan *interaction diagrams* (*sequence diagram*), pembuatan HIPO aplikasi, hingga pembuatan *flowchart* program dan spesifikasi kebutuhan *software* dan *hardware* untuk mengimplementasi sistem yang dibangun.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan secara keseluruhan yang diperoleh dari hasil analisis dan perancangan sistem informasi serta saran-saran untuk penerapan dan pengembangan lebih lanjut pada sistem yang bersangkutan.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Dalam kehidupan sehari-hari di era teknologi dan komputerisasi saat ini, kita sering mendengar istilah mengenai sistem. Misalkan sistem komputer, sistem operasi, sistem informasi, sistem geografis, sistem akademis dan lainnya. Bahkan disadari maupun tidak, kita sering menggunakan atau memanfaatkan layanan yang diberikan oleh sistem tersebut.

Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama. Secara garis besar, sebuah sistem informasi terdiri atas tiga komponen utama. Ketiga komponen tersebut mencakup *software*, *hardware* dan *brainware*. Ketiga komponen ini saling berkaitan satu sama lain (Pratama, 2014).

2.1.1 Pengertian Sistem

Pengertian tentang sistem pertama kali dapat diperoleh dari definisi sistem itu sendiri. Pendekatan sistem memberikan banyak manfaat dalam memahami lingkungan. Pendekatan sistem berusaha menjelaskan sesuatu yang dipandang dari sudut pandang sistem serta berusaha untuk menemukan struktur unsur yang membentuk sistem tersebut.

Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai suatu tujuan. Organisasi terdiri dari sejumlah sumber daya manusia, material, mesin, uang dan informasi. Sumber daya tersebut bekerja sama menuju tercapainya suatu tujuan tertentu yang ditentukan oleh pemilik atau manajemen. Terdapat dua kelompok pendekatan sistem di dalam mendefinisikan sistem yaitu pendekatan prosedur dan pendekatan pada komponen atau elemen (Yakub, 2012).

2.1.1.1 Pendekatan Prosedur

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau tujuan tertentu. Berikut ini adalah beberapa pengertian dari prosedur.

- a. Prosedur adalah urutan-urutan operasi klerikal atau tulis menulis yang melibatkan beberapa orang di dalam suatu departemen untuk menjamin penanganan yang seragam.
- b. Prosedur adalah urutan-urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya.

Sebagai perincian langkah-langkah dari sistem dan rangkaian kegiatan yang saling berhubungan erat satu sama lainnya untuk mencapai tujuan tertentu, maka sistem dan prosedur memiliki manfaat sebagai berikut.

- a. Memberikan keseragaman dalam melakukan tindakan.
- b. Menyediakan pandangan yang menyeluruh pada situasi dan persoalan yang dihadapi dengan realita.
- c. Dapat menyederhanakan pelaksanaan dalam pengambilan keputusan.
- d. Dapat mengurangi kesalahan-kesalahan yang timbul pada pelaksanaan pekerjaan.
- e. Dapat dilaksanakan dengan cepat.
- f. Keputusan yang salah dan terburu-buru dapat dikurangi.
- g. Membantu usaha-usaha latihan karyawan dengan diterapkan syarat-syarat kerja, ditentukannya hubungan kerja, serta diuraikan secara lengkap aliran kerja.

Untuk menjelaskan sebuah mobil, pendekatan prosedur didefinisikan sebagai kumpulan dari prosedur-prosedur untuk menghidupkan mesin, memasukan gigi persneling, menjalankan dan lain-lain. Penggunaan pendekatan prosedur ini akan lebih menyulitkan pengguna untuk memahami sebuah mobil (Yakub, 2012).

2.1.1.2 Pendekatan Komponen

Pendekatan pada komponen akan lebih mudah dalam mempelajari sistem untuk tujuan analisis dan perancangan sistem. Untuk menganalisis dan merencanakan suatu sistem, seharusnya analisis sistem mengerti dahulu komponen-komponen atau subsistem-

subsistemnya. Sebagai contoh subsistem akuntansi terdiri dari subsistem akuntansi penjualan, akuntansi pembelian, akuntansi penggajian dan akuntansi biaya.

Subsistem dalam sistem tidak dapat berdiri sendiri, subsistem tersebut saling berinteraksi dan saling berhubungan membentuk satu kesatuan. Sebuah mobil dapat didefinisikan sebagai kumpulan dari komponen kerangka mobil, bodi mobil, tempat duduk, mesin, kemudi dan sebagainya yang bekerja bersama-sama membentuk satu kesatuan untuk mencapai tujuan. Jika memahami komponen-komponen mobil, maka akan lebih mudah merangkai bersama-sama untuk membentuk suatu mobil (Yakub, 2012).

2.1.2 Karakteristik Sistem

Sebuah sistem harus memiliki sifat tertentu untuk mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem (Hutahaean, 2014). Agar suatu sistem menjadi sistem yang baik maka sistem tersebut memiliki karakteristik, yaitu:

a. **Komponen Sistem (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

b. **Batasan Sistem (*Boundary*)**

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. **Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar sistem adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dikendalikan, kalau hal tersebut diabaikan maka akan mengganggu kelangsungan hidup sistem.

d. **Penghubung Sistem (*Interface*)**

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

e. Masukan Sistem (*input*)

Masukkan adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem komputer, program adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna atau pembuangan. Sebagai contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

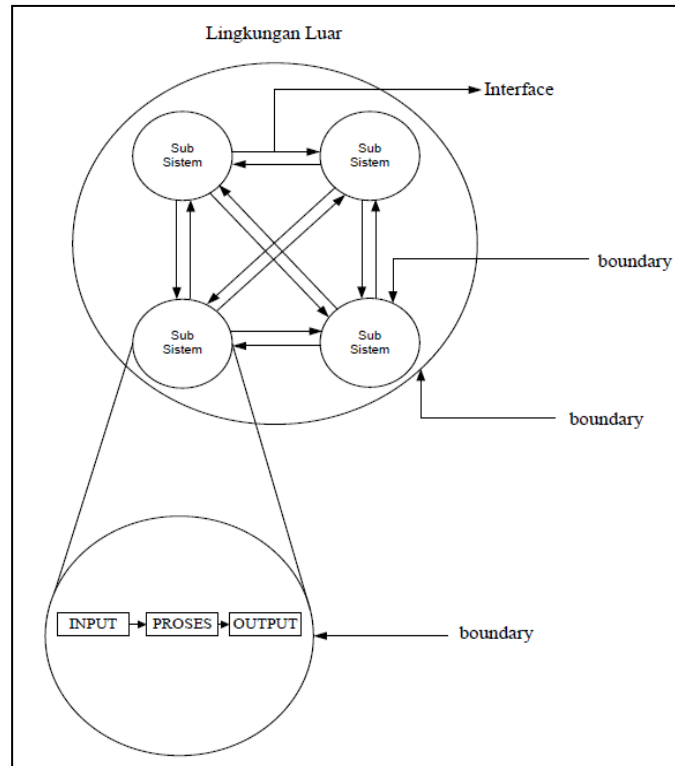
g. Pengolah Sistem (*Procces*)

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, sistem akuntansi akan mengolah data menjadi laporan-laporan keuangan.

h. Sasaran Sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

Untuk memudahkan dalam membayangkan dan memahami penjelasan mengenai karakteristik sistem yang disampaikan, maka disajikan sebuah bagan sederhana untuk mengilustrasikannya. Gambar II.1 berikut ini menunjukkan bagan sederhana mengenai karakteristik sistem.



Gambar II.1 Karakteristik Sistem
Sumber: Hutahaean (2014)

2.1.3 Elemen Sistem

Menurut Kadir (2014) Elemen yang membentuk sebuah sistem yaitu:

a. Tujuan

Setiap sistem memiliki tujuan (*goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tidak terarah dan tidak terkendali. Tentu saja tujuan antara satu sistem dengan sistem lain berbeda-beda.

b. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak berwujud. Contoh masukan yang berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa dari pelanggan).

c. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa pemanasan bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.

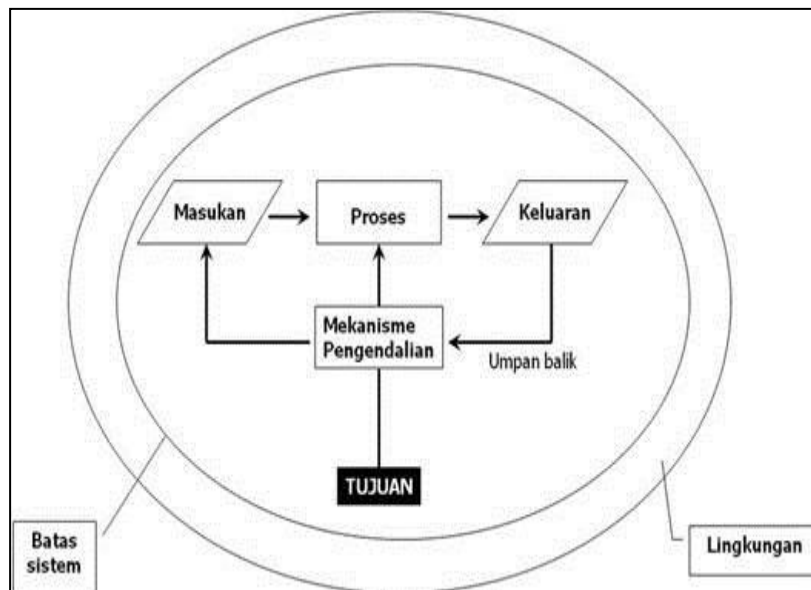
d. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan dan sebagainya.

e. Metode Pengendalian dan Umpan Balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feed back*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.

Selain itu, sistem juga berinteraksi dengan lingkungan dan memiliki batas. Gambar II.2 memperlihatkan hubungan antara elemen dan juga kaitannya dengan lingkungan.



Gambar II.2 Hubungan Elemen Sistem

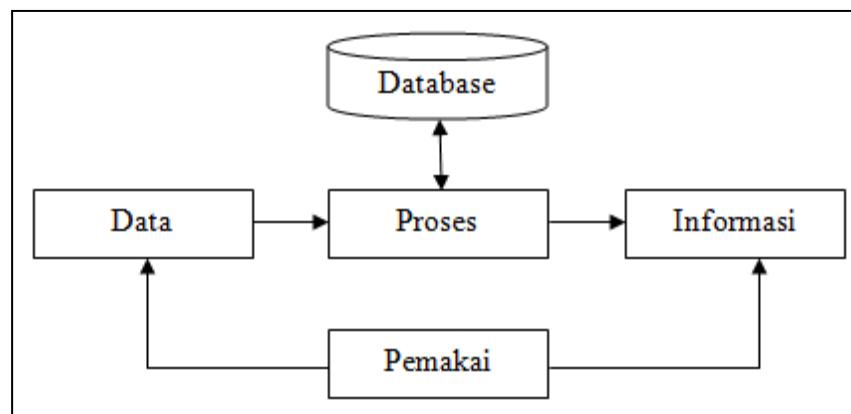
Sumber: Kadir (2014)

2.2 Konsep Dasar Informasi

Informasi merupakan salah satu sumber daya penting dalam manajemen modern. Banyak keputusan strategis yang bergantung kepada informasi. Sebagaimana diketahui, sumber

daya yang dimaksud mencakup manusia, material, mesin, modal dan informasi merupakan sumber daya vital bagi kelangsungan suatu organisasi bisnis.

Informasi didefinisikan sebagai data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Informasi juga disebut data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakannya. Para pembuat keputusan memahami bahwa informasi menjadi faktor kritis dalam menentukan kesuksesan atau kegagalan dalam suatu bidang usaha. Sistem apapun tanpa ada informasi tidak akan berguna, karena sistem tersebut akan mengalami kemacetan dan akhirnya berhenti. Informasi dapat berupa data mentah, data tersusun, kapasitas sebuah saluran informasi dan sebagainya (Yakub, 2012). Hubungan antara data informasi digambarkan dalam Gambar II.3 berikut ini.

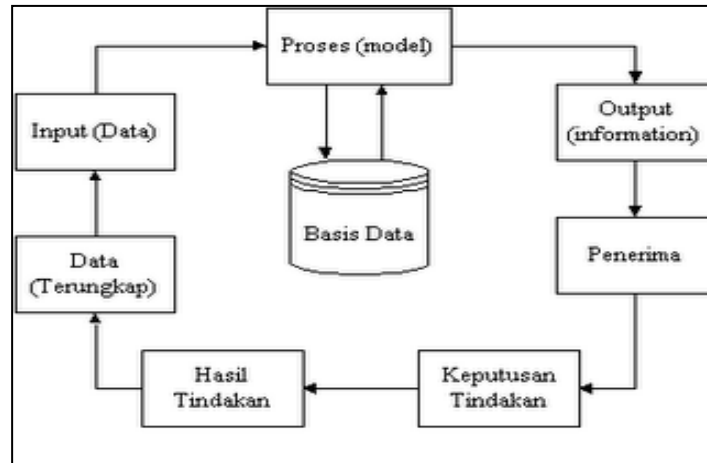


Gambar II.3 Pemrosesan Data Menjadi Informasi

Sumber: Yakub (2012)

2.2.1 Siklus Informasi

Siklus informasi (*information cycle*) atau siklus pengolahan data (*data processing cycle*) adalah gambaran secara umum mengenai proses terhadap data sehingga menjadi informasi yang bermanfaat bagi pengguna. Informasi yang menghasilkan informasi berikutnya, demikian seterusnya proses pengolahan data menjadi informasi. Untuk memperoleh informasi yang bermanfaat bagi penerimanya, perlu untuk dijelaskan bagaimana siklus yang terjadi atau dibutuhkan dalam menghasilkan informasi. Siklus informasi digambarkan pada Gambar II.4 sebagai berikut (Yakub, 2012).



Gambar II.4 Siklus Informasi

Sumber: Yakub (2012)

Pertama data dimasukkan dalam model yang umumnya memiliki urutan proses tertentu dan pasti, setelah diproses akan menghasilkan nilai yang bermanfaat bagi penerima sebagai dasar dalam membuat keputusan atau melakukan tindakan tertentu.

kemudian dari keputusan atau tindakan tersebut akan menghasilkan atau diperoleh kejadian-kejadian tertentu yang akan digunakan kembali sebagai data yang nantinya akan dimasukkan ke dalam (proses) dan akan begitu seterusnya.

2.2.2 Jenis-Jenis Informasi

Informasi jika dilihat dari sifat dan sumbernya dapat dibedakan dari beberapa jenis. Jenis-jenis informasi tersebut dibedakan menjadi informasi manajerial, sumber dan rutinitas, serta fisik (Yakub, 2012).

- a. Informasi manajerial, yaitu strategis untuk manajerial tingkat atas, informasi taktis untuk manajerial tingkat menengah dan informasi operasional untuk manajerial tingkat bawah.
- b. Sumber informasi, dibagi menjadi informasi internal dan eksternal. Informasi internal adalah informasi yang menggambarkan keadaan (*profile*), sedangkan informasi eksternal adalah informasi yang menggambarkan ada tidaknya perubahan di luar organisasi. Informasi ini biasanya lebih banyak digunakan untuk kegiatan-kegiatan manajerial tingkat atas.
- c. Informasi rutinitas, dibagi menjadi informasi rutin dan tidak rutin. Informasi rutin digunakan secara periodik terjadwal dan digunakan untuk penanggulangan masalah

rutin, sedangkan informasi tidak rutin diperlukan untuk penanggulangan masalah khusus.

- d. Informasi fisik, dapat diartikan susunan yang terdiri dari perangkat keras, perangkat lunak dan tenaga pelaksananya yang secara bersama-sama saling mendukung untuk menghasilkan suatu produk dan sistem informasi dari segi fungsi merupakan suatu proses berurutan dimulai dari pengumpulan data dan diakhiri dengan komunikasi.

2.3 Konsep Dasar Sistem Informasi

Sistem informasi (*information system*) merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi serta menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Sistem informasi juga dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk menyajikan informasi. Sistem informasi merupakan sistem pembangkit informasi, kemudian dengan integrasi yang dimiliki antara subsistem, maka sistem informasi akan mampu menyediakan informasi yang berkualitas, tepat, cepat dan akurat dengan manajemen yang membutuhkannya.

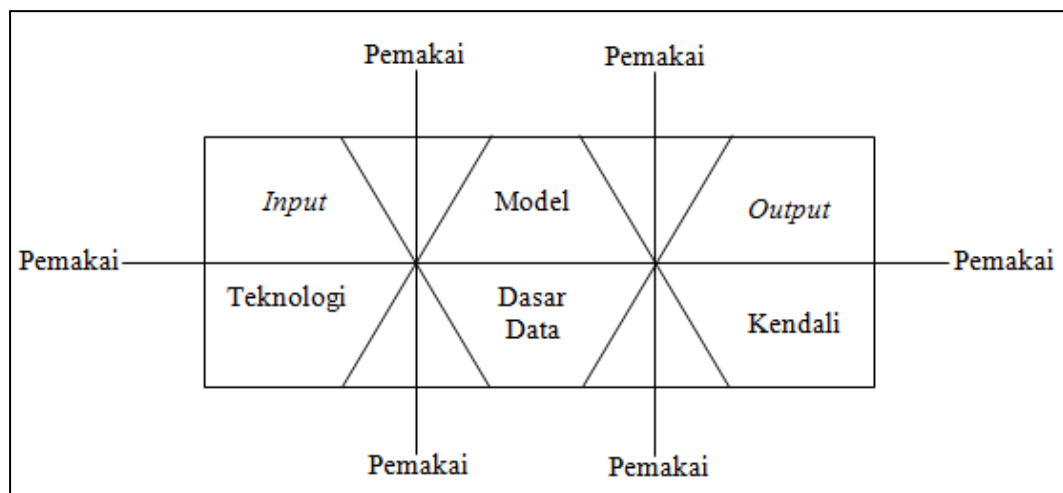
Sistem informasi juga merupakan suatu kumpulan dari komponen-komponen dalam organisasi yang berhubungan dengan proses penciptaan dan aliran informasi. Pada lingkungan berbasis komputer, sistem informasi menggunakan perangkat keras dan lunak komputer, jaringan telekomunikasi, manajemen basis data dan berbagai bentuk teknologi informasi yang lain dengan tujuan untuk mengubah sumber data menjadi berbagai macam informasi yang dibutuhkan oleh pemakai.

Sistem informasi merupakan sebuah susunan yang terdiri dari beberapa komponen atau elemen. Komponen sistem informasi disebut dengan istilah blok bangunan (*building block*). Komponen sistem informasi tersebut terdiri dari blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technology block*) dan basis data (*database block*) (Yakub, 2012).

- a. Blok masukan (*input block*), *input* memiliki data yang masuk ke dalam sistem informasi, juga metode-metode untuk menangkap data yang dimasukkan.
- b. Blok model (*model block*), blok ini terdiri dari kombinasi prosedur logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di basis data.

- c. Blok keluaran (*output block*), produk dari sistem informasi adalah keluaran yang merupakan informasi berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
- d. Blok teknologi (*technology block*), blok teknologi digunakan untuk menerima *input*, menyimpan, mengakses data, menghasilkan dan mengirimkan keluaran dari sistem secara keseluruhan. Teknologi terdiri dari tiga bagian utama yaitu teknisi (*brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).
- e. Basis data (*database block*), basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak (*software*) untuk memanipulasinya.

Sistem informasi memiliki lima blok utama yang saling tersusun yaitu blok masukan, model, keluaran, teknologi dan basis data. Untuk mempermudah dalam memahami blok-blok ini, berikut adalah Gambar II.5 yang menunjukkan adanya interaksi antara blok sistem informasi.



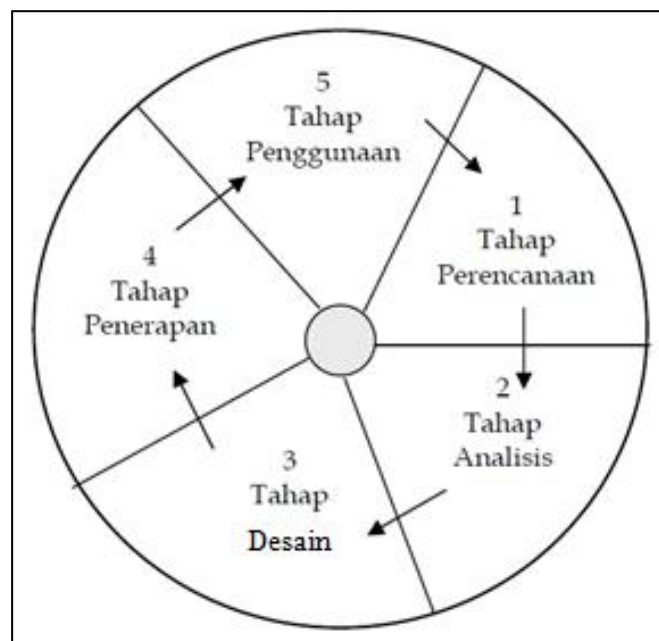
Gambar II.5 Interaksi Blok Sistem Informasi
 Sumber: Yakub (2012)

2.4 Siklus Pengembangan Perangkat Lunak (SDLC)

Siklus pengembangan perangkat lunak merupakan penerapan pendekatan sistem untuk mengembangkan dan menggunakan sistem berbasis komputer. Siklus hidup sistem itu sendiri merupakan metodologi, tetapi polanya lebih dipengaruhi oleh kebutuhan untuk

mengembangkan sistem yang lebih cepat. Pengembangan sistem yang lebih responsif dapat dicapai dengan peningkatan siklus hidup dan penggunaan peralatan pengembangan berbasis komputer.

Contoh siklus tersebut terdapat dalam perusahaan yang menggunakan komputer. Pada kenyataannya siklus ini adalah sarana yang digunakan oleh manajemen untuk melaksanakan rencana strategis. Konsep siklus hidup sistem menjadikan segala sesuatu yang tumbuh, menjadi dewasa setiap waktu dan akhirnya mati. Pola ini digunakan untuk sistem dasar komputer seperti subsistem pemrosesan data atau SSD. Berikut ini merupakan Gambar II.6 yang menunjukkan bagan siklus hidup sistem (Djahir dan Pratita, 2014).



Gambar II.6 Siklus Hidup Sistem
Sumber: Djahir dan Pratita (2014)

2.4.1 Tahap SDLC

SDLC memiliki lima tahapan sebelum mengulangi siklusnya seperti yang dijelaskan pada Gambar II.6. Berikut ini merupakan uraian secara rinci dari lima fase SDLC:

a. Tahap Perencanaan

Fase ini memiliki delapan langkah yang dimulai dari menyadari masalah, mendefinisikan masalah, menentukan tujuan sistem, mengidentifikasi kendala-kendala sistem,

membuat studi kelayakan, mempersiapkan usulan penelitian sistem, menyetujui atau menolak penelitian proyek dan menetapkan mekanisme pengendalian. Tanpa adanya kejelasan dari kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektivitasnya.

b. Tahap Analisis

Analisis sistem adalah penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbaharui. Ada enam tahap analisis yang meliputi mengumumkan penelitian sistem, mengorganisasikan tim proyek, mendefinisikan kebutuhan informasi, mendefinisikan kriteria kinerja sistem, menyiapkan usulan rancangan dan menyetujui atau menolak rancangan proyek.

c. Tahap Rancangan

Fase Desain ini meliputi menyiapkan rancangan sistem yang terinci, mengidentifikasi berbagai alternatif konfigurasi sistem, mengevaluasi berbagai alternatif konfigurasi sistem, memilih konfigurasi yang terbaik, menyiapkan usulan penerapan dan menyetujui atau menolak penerapan sistem. Proses rancangan akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*.

d. Tahap Penerapan

Merupakan kegiatan memperoleh dan mengintegrasikan sumber daya fisik dan konseptual yang menghasilkan suatu sistem yang bekerja. Langkah-langkah pada tahap penerapan yaitu merancang penerapan, mengumumkan penerapan, mendapatkan sumber daya perangkat keras dan lunak, menyiapkan *database*, menyiapkan fasilitas fisik, mendidik peserta maupun pemakai dan masuk ke sistem baru.

e. Tahap Penggunaan

Tahap penggunaan memiliki tiga langkah yang meliputi penggunaan sistem, audit sistem dan pemeliharaan sistem. Tahap ini akan berlangsung hingga tiba waktunya untuk merancang sistem itu kembali jika diperlukan. Proses merancang kembali akan mengakibatkan berulangnya siklus hidup sistem secara keseluruhan.

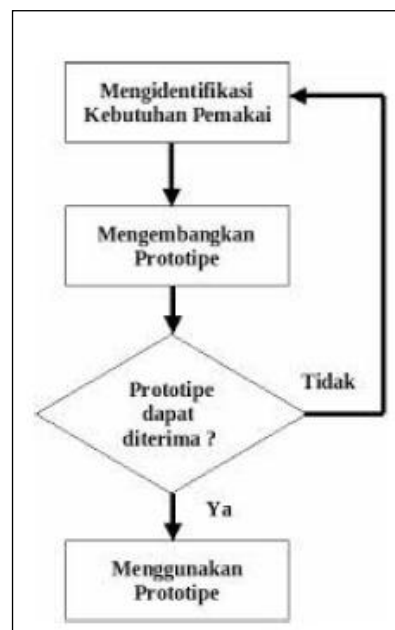
2.4.2 Prototipe

Menurut McLeod (2011) prototipe adalah suatu versi sistem potensial yang disediakan bagi pengembang dan calon pengguna yang dapat memberikan gambaran bagaimana kira-kira

sistem tersebut akan berfungsi bila telah disusun dalam bentuk yang lengkap. Proses dalam memproduksi suatu prototipe disebut *prototyping*. Tujuannya adalah menghasilkan prototipe secepat mungkin dan memperoleh umpan balik dari pengguna yang akan memungkinkan prototipe untuk ditingkatkan sampai sistem dianggap sempurna. Adapun terdapat dua jenis prototipe yaitu prototipe evolusioner dan prototipe *requirement* (McLeod, 2011).

2.4.2.1 Prototipe Evolusioner

Prototipe evolusioner adalah prototipe yang terus menerus diperbaiki sampai semua kriteria sistem yang baru terpenuhi. Ada empat langkah yang yang diambil dalam mengembangkan suatu prototipe evolusioner yaitu identifikasi kebutuhan pengguna, mengembangkan prototipe, menentukan prototipe dapat diterima atau tidak, dan penggunaan prototipe yang dapat dilihat pada Gambar II.7.

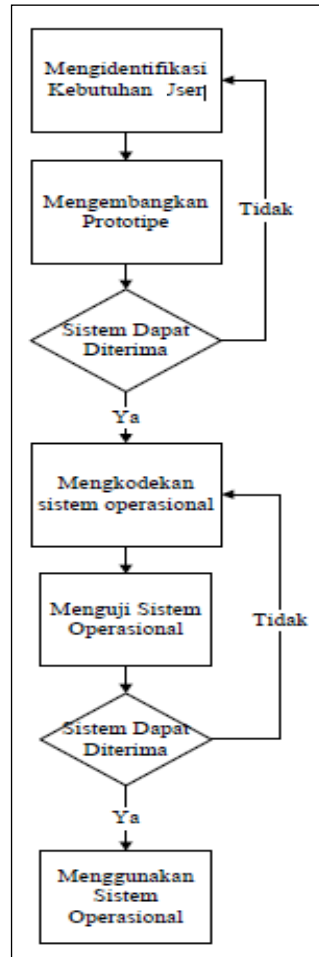


Gambar II.7 Pengembangan Prototipe Evolusioner
Sumber: McLeod (2011)

2.4.2.2 Prototipe Requirement

Prototipe *requirement* adalah suatu pengembangan untuk menentukan kebutuhan fungsional dari sistem baru pada saat para pengguna tidak mampu mengungkapkan dengan

tepat apa yang mereka butuhkan. Saat kebutuhan telah ditentukan prototipe *requirement* dapat mulai dikerjakan dan proyek siap untuk mengembangkan suatu sistem yang baru. Berikut ini adalah tahapan prototipe *requirement* yang dapat dilihat pada Gambar II.8.



Gambar II.8 Pengembangan *Prototipe Requirement*
Sumber: McLeod (2011)

2.5 Pengertian Klaim

Klaim menurut Kamus Besar Bahasa Indonesia (www.kbbi.co.id, 2016) adalah tuntutan pengakuan atas suatu fakta bahwa seseorang berhak (memiliki atau mempunyai) atas sesuatu. Ada beberapa contoh klaim yang dapat ditemukan dalam kehidupan sehari-hari, seperti klaim asuransi dan klaim garansi terhadap produk.

Perusahaan manufaktur yang memproduksi suatu barang biasanya akan menerapkan kebijakan pemberian garansi terhadap konsumen. Garansi tidak saja memberikan manfaat kepada konsumen tetapi juga kepada produsen. Bagi konsumen, garansi melindungi dari pembelian produk yang cacat. Sementara itu, bagi produsen garansi membatasi klaim yang tidak rasional dari konsumen. Disamping itu, produsen juga dapat memanfaatkan garansi sebagai alat promosi yang efektif untuk produknya, serta dapat membangun suatu kepercayaan masyarakat terhadap produk yang dihasilkan.

Konsumen memiliki hak klaim terhadap barang yang dibeli, apalagi klaim tersebut masih dalam masa garansi. Hal tersebut menunjukkan bahwa konsumen sangat dilindungi terhadap produk yang dibeli. Bahkan peraturan mengenai garansi dan klaim sudah diatur negara dalam Undang-Undang Dasar RI No. 8 tahun 1999 tentang Perlindungan Konsumen.

2.6 Pengertian Produk

Produk adalah sesuatu yang dapat ditawarkan ke pasar untuk diperhatikan, dipakai, dimiliki, atau dikonsumsi sehingga dapat memuaskan keinginan atau kebutuhan. Dari pengertian ini dapat disimpulkan bahwa hampir semua yang termasuk produksi adalah benda nyata yang dapat dilihat, diraba, dan dirasakan. Karena produk adalah benda ril, maka jenisnya cukup banyak. Secara garis besar jenis-jenis produk bisa kita perinci menjadi dua jenis, yaitu produk konsumsi dan produk industri (Royan, 2011).

2.6.1 Produk Konsumsi (*Consumer Products*)

Produk konsumsi adalah barang yang dipergunakan oleh konsumen akhir atau rumah tangga dengan maksud tidak untuk dibisniskan atau dijual lagi. Barang-barang yang termasuk jenis produk konsumsi ini antara lain sebagai berikut:

- a. Barang kebutuhan sehari-hari (*convenience goods*), yaitu barang yang umumnya sering kali dibeli, segera dan memerlukan usaha yang sangat kecil untuk memilikinya, misalnya barang kelontong dan sebagainya.
- b. Barang belanja (*shopping goods*), yaitu barang yang dalam proses pembelian dibeli oleh konsumen dengan cara membandingkan berdasarkan kesesuaian mutu, harga, dan model, misalnya pakaian, sepatu, sabun, dan lain sebagainya.

- c. Barang khusus (*speciality goods*), yaitu barang yang memiliki ciri-ciri unik atau merk khas dimana kelompok konsumen berusaha untuk memiliki atau membelinya, misalnya mobil, kamera, dan lain sebagainya.

2.6.2 Produk Industri (*Business Product*)

Produk industri adalah barang yang akan menjadi begitu luas dipergunakan dalam program pengembangan pemasaran. Barang industri juga dapat dirinci lebih lanjut jenisnya antara lain sebagai berikut:

- a. Bahan mentah, yaitu barang yang akan menjadi bahan baku secara fisik untuk memproduksi produk lain, seperti hasil hutan, gandum, dan lain sebagainya.
- b. Bahan baku dan suku cadang pabrik, yaitu barang industri yang digunakan untuk suku cadang yang aktual bagi produk lain, misalnya mesin, pasir, dan lain sebagainya.
- c. Perbekalan operasional, yaitu barang kebutuhan sehari-hari bagi sektor industri, misalnya alat-alat kantor, dan lain-lain.

2.7 Pengertian Baterai Aki

Baterai aki, atau bisa juga disebut *accu* adalah sebuah sel listrik dimana di dalamnya berlangsung proses elektrokimia yang *reversible* (dapat berbalikan) dengan efisiennya yang tinggi. Hal yang dimaksudkan dengan proses elektrokimia *reversible* adalah di dalam baterai dapat berlangsung proses pengubahan kimia menjadi tenaga listrik (proses pengosongan) dan sebaliknya dari tenaga listrik menjadi tenaga kimia. Pengisian kembali dengan cara regenerasi dari elektroda-elektroda yang dipakai, yaitu dengan melewatkan arus listrik dalam arah (polaritas) yang berlawanan didalam sel (Daryanto, 2011).

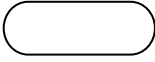

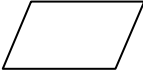

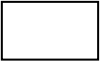

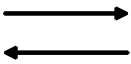




2.8 Flowchart

Untuk menggambarkan sebuah algoritma yang terstruktur dan mudah dipahami oleh orang lain maka dibutuhkan alat bantu yang berbentuk diagram alir (*flowchart*). *Flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah, sehingga *flowchart* merupakan langkah-langkah penyelesaian masalah yang dituliskan dalam simbol-simbol

tertentu. Diagram alir ini selain dibutuhkan sebagai alat komunikasi, juga diperlukan sebagai dokumentasi (Sitorus, 2015).

Tujuan dari *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas menggunakan simbol-simbol seperti yang dijelaskan pada Tabel II.1 berikut ini.

Tabel II.1 Simbol-Simbol *Flowchart* Menurut Sitorus


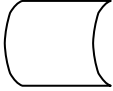
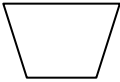






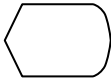

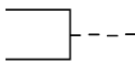



Simbol	Keterangan	Simbol	Keterangan
Terminal 	Menyatakan permulaan atau akhir suatu program	<i>Offline Connector</i> 	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
<i>Input / output</i> 	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya	<i>Connector</i> 	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
Proses 	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer	<i>Predifined Process</i> 	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
<i>Flow</i> 	Menyatakan jalannya arus suatu proses	<i>Punched Card</i> 	Menyatakan <i>input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu
<i>Decision</i> 	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban: ya/tidak	Dokumen 	Menunjukkan dokumen input dan output manual, mekanik atau komputer
<i>Punch Tape</i> 	Menunjukkan <i>input/output</i> menggunakan pita kertas berlubang		

Sumber: Sitorus (2015)

Menurut Jogiyanto (2005) *flowchart* merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Untuk melengkapi simbol-simbol *flowchart*

sebelumnya, berikut adalah simbol-simbol *flowchart* menurut Jogiyanto yang tampak pada tabel II.2.

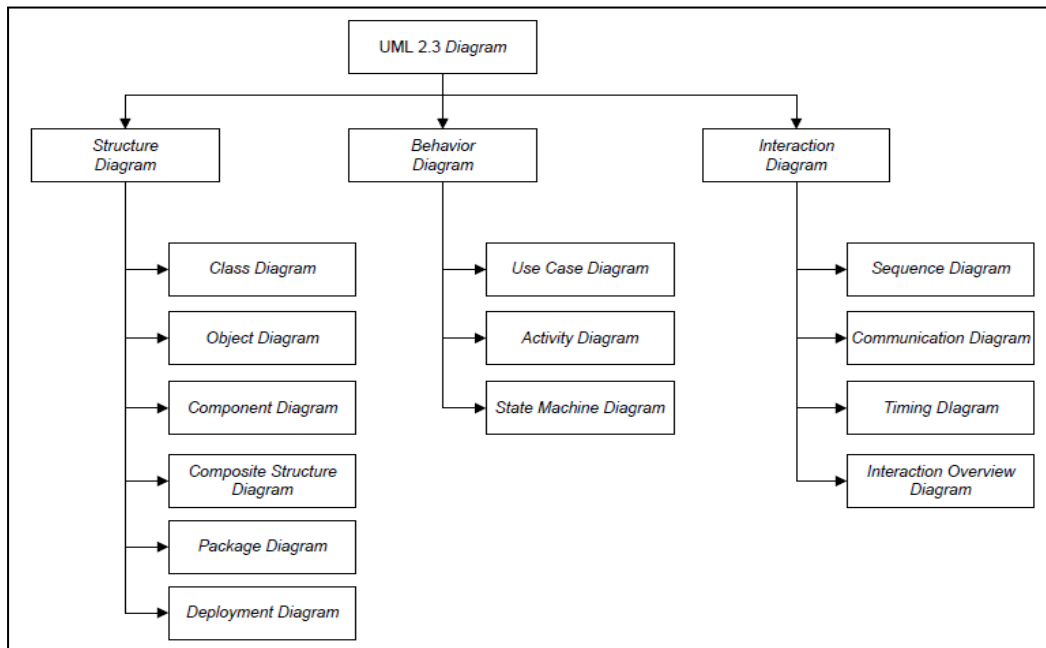
Tabel II.2 Simbol-Simbol *Flowchart* Menurut Jogiyanto

Simbol	Keterangan	Simbol	Keterangan
Pengurutan <i>Offline</i> 	Menunjukkan proses pengurutan data di luar proses komputer	Diskette 	Menunjukkan <i>input/output</i> menggunakan <i>diskette</i>
Kegiatan Manual 	Menunjukkan pekerjaan manual	Drum Magnetik 	Menunjukkan <i>input/output</i> menggunakan drum magnetik
Simpanan <i>Offline</i> 	File <i>non</i> -komputer yang diarsip menurut angka (<i>Numerical</i>)	Pita Kontrol 	Penggunaan pita kontrol dalam <i>batch control total</i> untuk pencocokan dalam <i>batch processing</i>
Simpanan <i>Offline</i> 	File <i>non</i> -komputer yang diarsip menurut huruf (<i>Alphabetical</i>)	Keyboard 	Menunjukkan <i>input</i> yang menggunakan <i>online keyboard</i>
Simpanan <i>Offline</i> 	File <i>non</i> -komputer yang diarsip menurut Tanggal (<i>Cronological</i>)	Display 	Menunjukkan <i>output</i> yang ditampilkan di monitor
Operasi Luar 	Menunjukkan operasi yang dilakukan di luar proses operasi komputer	Penjelasan 	Menunjukkan penjelasan dari suatu proses
Pita Magnetik 	Menunjukkan <i>input/output</i> menggunakan pita magnetik	Hubungan Komunikasi 	Menunjukkan proses transmisi data melalui <i>channel</i> komunikasi
Hard Disk 	Menunjukkan <i>input/output</i> menggunakan <i>hard disk</i>		

Sumber: Jogiyanto (2005)

2.9 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks tertentu. Pada Gambar II.9 dijelaskan bahwa UML terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori (Rosa dan Shalahuddin, 2013).



Gambar II.9 Diagram UML

Sumber: Rosa dan Shalahuddin (2013)

Berikut ini penjelasan singkat dari pembagian kategori tersebut (Rosa dan Shalahuddin, 2013):

- a. *Structure diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- b. *Behavior diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan proses sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- c. *Interaction diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub sistem pada suatu sistem.

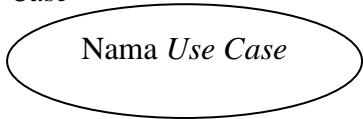


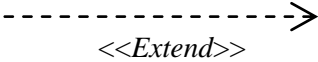

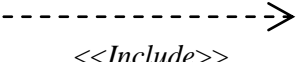
2.9.1 *Use Case Diagram*

Use Case Diagram merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau

lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa dan Shalahuddin, 2013).

Berikut adalah simbol-simbol yang ada pada *use case* diagram yang terdapat pada Tabel II.3:

Tabel II.3 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2	<p>Aktor/actor</p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
3	<p>Asosiasi/association</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.
4	<p>Ekstensi/Extend</p> 	Relasi <i>use case</i> tambahan ke <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.
5	<p>Generalisasi/generalization</p> 	Hubungan generalisasi dan spesifikasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
6	<p>Include</p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.



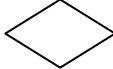


Sumber: Rosa dan Shalahuddin (2013)

2.9.2 *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa dan Shalahuddin, 2013).

Berikut adalah simbol-simbol yang ada pada *activity diagram* yang terdapat pada Tabel II.4:

Tabel II.4 Simbol-Simbol *Activity Diagram*

No	Simbol	Deskripsi
1	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir.


Sumber: Rosa dan Shalahuddin (2013)

2.9.3 *Sequence Diagram*

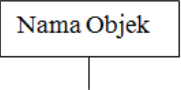


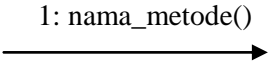
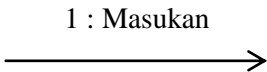
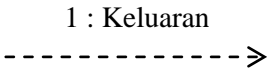
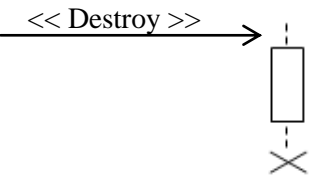
Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu (Rosa dan Shalahuddin, 2013).

Berikut ini adalah komponen-komponen dasar yang ada di dalam *Sequence Diagram* yang terdapat pada Tabel II.5:

Tabel II.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1	Aktor  Nama Aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.

Tabel II.5 Simbol-Simbol *Sequence Diagram*

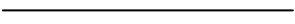

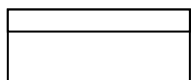

No	Simbol	Deskripsi
2	Objek 	Menyatakan objek yang berinteraksi pesan.
3	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
4	Pesan tipe <i>create</i> 	Menyatakan suatu objek membuat objek lain, arah panah mengarah kepada objek yang dibuat.
5	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi atau metode yang ada pada objek lain atau dirinya sendiri.
6	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah, mengarah kepada objek yang dikirim.
7	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah dijalankan suatu operasi dan menghasilkan kembalian dari objek tertentu, arah panah mengarah kepada objek yang menerima kembalian.
8	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah kepada objek yang diakhiri, sebaiknya jika ada create maka ada <i>destroy</i> .

Sumber: Rosa dan Shalahuddin (2013)

2.9.4 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. (Rosa dan Shalahudin, 2013). Berikut Tabel II.6 tentang simbol- simbol *class diagram*:

Tabel II.6 Simbol-Simbol *Class Diagram*

No	Simbol	Keterangan
1	<p><i>Generalization</i></p> 	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2	<p><i>Nary Association</i></p> 	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
3	<p><i>Class</i></p> 	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4	<p><i>Collaboration</i></p> 	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

Sumber: Rosa dan Shalahuddin (2013)

Dalam kelas terdapat beberapa *keys* yang berfungsi untuk membedakan semua basis data dalam tabel secara unik. Pengertian *keys* menurut Yanto (2016) yaitu sebagai berikut:

1. *Candidate key*

Candidate key adalah sejumlah kecil *attribute* yang secara unik mengidentifikasi setiap kejadian dari setiap tipe *entity*.

2. *Primary key*

Primary key adalah *candidate key* yang terpilih untuk mendefinisikan secara unik pada setiap kejadian dari sebuah tipe *entity*.

3. *Composite key*

Composite key adalah sebuah *candidate key* yang terdiri dari dua atau banyak *attribute*.

4. *Foreign key*

Foreign key adalah himpunan *attribute* dalam data *relationship* yang cocok dengan *candidate key* dari beberapa *relationship* lainnya.

Pada relasi *class diagram* terdapat suatu penanda yang disebut *multiplicity*. *Multiplicity* ini akan mengindikasikan berapa banyak obyek dari suatu kelas terelasi ke obyek lain. Berikut Tabel II.7 tentang notasi UML untuk *multiplicity*:

Tabel II.7 Notasi *Multiplicity*

Multiplicity	Keterangan
*	Banyak
0	Nol
1	Satu
0..*	Antara nol sampai banyak
1..*	Antara satu sampai banyak
0..1	Nol atau satu
1..1	Tepat satu

Sumber: Britton dan Doake (2005)

2.9.5 *Component Diagram*

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus kepada komponen sistem yang dibutuhkan dan ada dalam sistem. Diagram komponen digunakan untuk memodelkan hal- hal berikut:




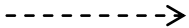

- a. *Source code* program perangkat lunak.
- b. Komponen *executable* yang dilepas ke user.
- c. Sistem yang beradaptasi dengan sistem lain.
- d. *Framework system*

Berikut ini adalah komponen-komponen dasar yang ada di dalam *Component Diagram* yang terdapat pada Tabel II.8:

Tabel II.8 Simbol-Simbol *Component Diagram*

No.	Simbol	Keterangan
------------	---------------	-------------------

Tabel II.8 Simbol-Simbol *Component Diagram*



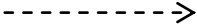

No.	Simbol	Keterangan
1	<i>Package</i> 	Merupakan sebuah bungkus dari satu atau lebih komponen.
2	<i>Link</i> 	Relasi antar komponen.
3	<i>Component</i> 	Komponen sistem.
4	<i>Dependency</i> 	Kebergantungan antar komponen.
5	<i>Interface</i> 	Sebagai antarmuka komponen.

Sumber: Rosa dan Shalahuddin (2013)

2.9.6 *Deployment Diagram*

Deployment diagram merupakan gambaran proses-proses berbeda pada suatu sistem yang berjalan dan bagaimana relasi di dalamnya. Hal inilah yang mempermudah *user* dalam pemakaian sistem yang telah dibuat dan diagram tersebut merupakan diagram yang statis. *Deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi (Rosa dan Shalahuddin, 2013). Berikut mengenai simbol-simbol *deployment diagram* yang akan dijelaskan pada Tabel II.9:

Tabel II.9 Simbol-Simbol *Deployment Diagram*

No	Simbol	Keterangan
1	<i>Package</i> 	Merupakan sebuah kumpulan dari satu atau lebih komponen.
2	<i>Link</i> 	Relasi antar objek.
3	<i>Dependency</i> 	Hubungan pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
4	<i>Node</i> 	Perangkat keras dan perangkat lunak.

Sumber: Rosa dan Shalahuddin (2013)

2.10 Kamus Data

Menurut Jogiyanto (2005), kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data diharapkan, analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Kamus data dapat berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan di dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses. Berikut ini adalah contoh penggunaan kamus data yang tertera pada Tabel II.10.

Tabel II.10 Contoh Kamus Data Untuk Tabel Barang

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
----	-------------	---------	------	---------	------------

Tabel II.10 Contoh Kamus Data Untuk Tabel Barang

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	ID Barang	ID_barang	Varchar	10	<i>Primary Key</i>
2	Nama Barang	Nama_barang	Char	40	
3	Jumlah	Jumlah	Integer	10	
4	Asal Barang	Asal_barang	Varchar	20	

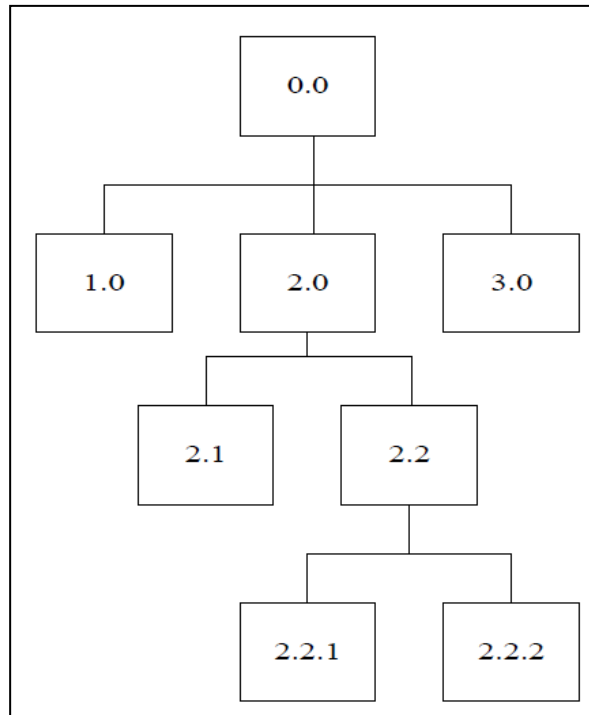
Sumber: Jogiyanto (2005)

2.11 *Hierarchy plus Input-Proses-Output (HIPO)*

HIPO (*Hierarchy plus Input-Process-Output*) aslinya dibuat oleh IBM sebagai alat untuk mendokumentasikan program. Bagan HIPO merupakan bagan yang memperagakan apa yang dikerjakan suatu program, data apa yang digunakan, dan keluaran yang dihasilkannya.

Menurut Jogiyanto (2005), HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program, fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan. Bagan HIPO lebih mudah dibaca dibanding dengan bagan arus, sangat rinci, fleksibel, mudah dimodifikasi, dan dikelola. Dalam membuat bagan HIPO, terdapat tiga jenis diagram, yaitu: daftar isi visual (*the visualtable of contents / VTOC*), diagram peninjauan dan rincian diagram.

Visual table of contents adalah salah satu jenis dari HIPO yang digunakan untuk menggambarkan hubungan fungsi-fungsi di sistem secara berjenjang, *visual table of contents* menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada diagram ini nama dan nomor dari program HIPO diidentifikasi. Struktur paket diagram dan hubungan fungsi juga diidentifikasi dalam bentuk hirarki. Keterangan masing-masing fungsi diberikan pada bagian penjelasan yang diikutsertakan dalam diagram ini. *Visual table of contents* ini dapat digambarkan pada Gambar II.10 sebagai berikut.



Gambar II.10 *Visual Table Of Contents*

Sumber: Jogiyanto (2005)

2.12 Java

Java merupakan pemrograman yang menanjak popularitasnya dewasa ini. Selain gratis, java mudah didapatkan dan juga tangguh. Java sendiri lahir pada tahun 1991 yang diprakasai oleh tim Sun melalui proyek bernama *Green* yang dipimpin oleh James Gosling.

Java menerapkan fitur-fitur dari bahasa pemrograman yang lain, di mana fitur itu dianggap merupakan suatu kelebihan oleh tim Sun. Misalnya JVM/JRE yang merupakan mesin maya pada bahasa *Pascal*, *Sintaks* dan *Exception Handling* diambil dari C/C++ dan lain sebagainya.

Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, *desktop*, *web*, *mobile* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Bahasa Pemrograman Java ini berorientasi objek (OOP-*Object Oriented Programming*), dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi

dikembangkan untuk berbagai sistem operasi dan bersifat *open source* (Jubilee Enterprise, 2015)

2.12.1 Netbeans

Netbeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasiskan Java dari *Sun Microsystems* yang berjalan di atas *swing*. *Swing* merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam *platform* seperti windows, linux, Mac OS dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface* (GUI), suatu kode editor atau teks, suatu *compiler* dan suatu *debugger*.

Netbeans juga digunakan oleh sang programmer untuk menulis, meng-*compile*, mencari kesalahan dan menyebarkan program netbeans yang ditulis dalam bahasa pemrograman java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat *professional desktop, enterprise, web, and mobile applications* dengan *Java language, C/C++,* dan bahkan *dynamic languages* seperti PHP, *JavaScript, Groovy,* dan *Ruby* (Jubilee Enterprise, 2015).

2.12.2 Fitur-Fitur Netbeans

Netbeans memiliki banyak kelebihan dari segi *code* sampai pada penggunaan aplikasinya. Hal tersebut tidak terlepas dari fitur yang dimiliki oleh Netbeans. Fitur-fitur yang terdapat di dalamnya antara lain:

- a. *Smart Code Completion*: untuk mengusulkan nama variabel dari suatu tipe, melengkapi *keyword* dan mengusulkan tipe parameter dari sebuah *method*.
- b. *Bookmarking*: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
- c. *Go to commands*: fitur yang digunakan untuk *jump* ke deklarasi variabel, *source code* atau file yang ada pada project yang sama.
- d. *Code generator*: jika kita menggunakan fitur ini kita dapat meng-*generate constructor, setter* dan *getter method* dan yang lainnya.

- e. *Error stripe*: fitur yang akan menandai baris yang eror dengan memberi *highlight* merah.

2.13 MySQL

Menurut Anhar (2010) MySQL adalah sebuah perangkat lunak sistem manajemen basis data (DBMS) dari sekian banyak DBMS seperti Oracle, MSSQL, Postgre SQL, dan lain-lain. MySQL merupakan DBMS yang multithread, multi-user yang bersifat gratis dibawah GNU *General Public Licence* (GPL).

MySQL memiliki beberapa kelebihan antara lain:

- a. *Portabilitas*
MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
- b. *Open Source*
MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
- c. *MultiUser*
MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- d. *Performance tuning*
MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- e. *Jenis Kolom*
MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed* atau *unsigned integer*, *float*, *double*, *char*, *text*, *date*, *timestamp*, dan lain-lain.
- f. *Perintah dan Fungsi*
MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).
- g. *Keamanan*
MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- h. *Konektivitas*

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).

i. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

j. Antar Muka

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

k. Klien dan Peralatan

MySQL dilengkapi dengan berbagai peralatan (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Terdapat tiga kategori tipe data yang didukung oleh MySQL, yaitu tipe data numerik, string, serta penganggulan dan waktu. Sebuah data yang akan disimpan harus sesuai dengan tipe data yang bersangkutan (Wahana Komputer, 2010).

1. Data Numerik

MySQL mendukung penyimpanan data standar numerik. Data numerik adalah salah satu bentuk data berupa angka, baik berupa bilangan bulat atau bilangan real. Bilangan bulat dapat berupa tipe data integer/int, tinyint, smallint, dan lainnya. Sebaliknya bilangan real dapat menyimpan data berupa angka pecahan. Pada Tabel II.11 dan Tabel II.12 dijelaskan macam tipe data yang termasuk dalam data numerik.

Tabel II.11 Tipe Data Numerik *Integer*

Tipe Data	Byte	Keterangan
<i>Tinyint</i>	1	-128 s/d 127 <i>signed</i> 0-255 <i>unsigned</i>
<i>Smallint</i>	2	-32768 s/d 32767 <i>signed</i> 0 s/d 65535 <i>unsigned</i>
<i>Mediumint</i>	3	-8388608 s/d 8388607 <i>signed</i> 0 s/d 16777215 <i>unsigned</i>
<i>Integer/int</i>	4	-2147483648 s/d 2147483647 <i>signed</i> 0 s/d 4294967295 <i>unsigned</i>

Tabel II.11 Tipe Data Numerik *Integer*

Tipe Data	Byte	Keterangan
<i>Bigint</i>	8	-9223372036854775808 s/d 9223372036854775807 <i>signed</i> 0 s/d 18446744073709551615 <i>unsigned</i>

Sumber: Wahana Komputer (2010)

Tabel II.12 Tipe Data Numerik *Floating Point*

Tipe Data	Byte	Keterangan
<i>Float (p)</i>	4 jika $0 \leq p \leq 24$ 8 jika $25 \leq p \leq 53$	P merepresentasikan presisi bit.
<i>Float</i>	4	Angka <i>floating point</i> kecil (<i>single-precision</i>)
<i>Double [Presisi], real</i>	8	Ukuran normal angka <i>floating point</i> (<i>double-precision</i>)
<i>Decimal/dec (M,D), Numeric (M<D)</i>	variasi	M adalah jumlah angka digit desimal dan D adalah angka di belakang tanda desimal.
<i>Bit (M)</i>	$(M+7)/8$	M adalah banyaknya bit setiap nilai. Bit ditambahkan pada versi 5.0.5, sebelumnya merupakan sinonim <i>tinyint</i> (1).

Sumber: Wahana Komputer (2010)

2. Data String

Tipe data String, dapat menyimpan semua data baik karakter, angka, waktu, maupun tanggal. Data dapat pula merupakan kombinasi karakter dan angka. Tipe data string pada MySQL dapat digolongkan menjadi dua, yaitu kelompok data yang berbentuk *Text/Blob* dan selain *Text/Blob*. Pada Tabel II.13 dan Tabel II.14 dijelaskan macam tipe data yang termasuk dalam data String.

Tabel II.13 Tipe Data *String Text/Blob*

Tipe Data	Byte	Kapasitas Penyimpanan
<i>Tinytext</i> <i>Tinyblob</i>	2^8-1	L+1 $0 \leq L \leq 2^8-1$
<i>Text</i> <i>Blob</i>	$2^{16}-1$	L+2 $0 \leq L \leq 2^{16}-1$
<i>Mediumtext</i> <i>Mediumblob</i>	$2^{24}-1$	L+3 $0 \leq L \leq 2^{24}-1$
<i>Longtext</i> <i>Longblob</i>	$2^{32}-1$	L+4 $0 \leq L \leq 2^{32}-1$

Sumber: Wahana Komputer (2010)

Tabel II.14 Tipe Data *String Selain Text/Blob*

Tipe Data	Byte	Keterangan
<i>Varchar</i>	225	Tipe <i>varchar</i> menyimpan data sebanyak yang dimasukkan.
<i>Char</i>	225	Tipe <i>char</i> sama dengan tipe <i>varchar</i> , hanya tempat penyimpanan selalu tetap.
<i>Binary</i>	255	<i>Binary</i> mirip dengan <i>char</i> , hanya saja yang disimpan adalah nilai biner (byte) dari data yang disimpan.
<i>Varbinary</i>	255	Menyimpan nilai biner sebanyak data yang dimasukkan.
<i>Enum</i>	N	Tipe data ini disebut juga tipe data validasi. Pada tipe data ini, data input telah dideklarasikan terlebih dahulu.
<i>Set</i>	N	Tipe <i>Set</i> memiliki fungsi yang sama dengan <i>enum</i> . N adalah banyaknya daftar input, tiap nilai bernilai 1 <i>byte</i> .

Sumber: Wahana Komputer (2010)

3. Tipe Data Penganggaran dan Waktu

Dalam menangani data tanggal dan waktu, MySQL memiliki tipe data tersendiri. Pada Tabel II.15 dijelaskan tipe data yang termasuk dalam data tanggal dan waktu.

Tabel II.15 Tipe Data Tanggal dan Waktu

Tipe Data	Byte	Keterangan
<i>Datetime</i>	8	Merupakan tipe data yang menyimpan tanggal dan jam. Formatnya YYYY-MM-DD HH:MM:SS.
<i>Date</i>	3	Tipe ini hanya menyimpan data tanggal. Formatnya YYYY-MM-DD.
<i>Timestamp</i>	4	Tipe data ini ditulis berjajar tanpa ada pembatas, menyimpan tanggal dan jam. Formatnya adalah YYYYMMDDHHMMSS.
<i>Time</i>	3	Tipe data ini hanya menyimpan data jam dengan format HH:MM:SS.
<i>Year</i>	1	Tipe data ini hanya menyimpan data tahun dengan format YYYY.

Sumber: Wahana Komputer (2010)

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Menurut Sugiyono (2013) metode penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Metodologi penelitian juga dikenal sebagai metode ilmiah dalam mencari, mengembangkan dan menguji suatu kebenaran pengetahuan. Metodologi penelitian juga membuat penelitian lebih terarah.

3.2 Jenis dan Sumber Data

Data yang diperlukan dalam penelitian ini ada dua jenis, yaitu data primer dan data sekunder. Sumber dari data-data ini berasal dari tempat yang diamati pada praktek kerja lapangan di PT Nipress Tbk.

1. Data Primer

Data primer adalah data yang diperoleh secara langsung dari perusahaan dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses klaim produk baterai diantaranya struktur organisasi, analisis sistem yang telah berjalan, proses bisnis sistem saat ini dan yang akan diusulkan, dan kebutuhan pengguna sistem.

2. Data Sekunder

Data sekunder adalah data yang diperoleh dari data yang tersedia dan telah terlebih dahulu dikumpulkan dan dilaporkan oleh orang lain, buku-buku dan kajian ilmiah dari berbagai sumber yang berkaitan dengan penelitian. Data tersebut adalah data umum perusahaan, profil perusahaan dan struktur organisasi perusahaan.

3.3 Metode Pengumpulan Data

Menurut Sugiyono (2013) teknik pengumpulan data merupakan langkah yang paling strategis dalam penelitian, karena tujuan utama dari penelitian adalah mendapatkan data. Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

a. Studi lapangan

Studi lapangan adalah usaha melakukan pengumpulan data secara langsung pada objek yang dilakukan dengan cara sebagai berikut:

1. Pengamatan, tahap ini dilakukan secara langsung di bagian *Claim Handling* PT Nipress Tbk dengan mengamati sistem klaim yang sedang berlangsung, mulai dari penerimaan produk baterai, analisis produk sampai dibuatnya laporan untuk kebutuhan pencarian akar masalah. Melalui teknik ini, data yang dibutuhkan diamati, dikumpulkan dan diolah sebagai bahan dalam penelitian.
2. Wawancara, yaitu mencari data yang dibutuhkan secara langsung melalui memberikan pertanyaan spontan terhadap segala hal yang diperlukan pada penyusunan tugas akhir ini. Wawancara ini dilakukan kepada pegawai di bagian *Claim Handling, Quality Assurance* dan beberapa pegawai di bagian produksi PT Nipress Tbk.

b. Studi kepustakaan

Studi kepustakaan dilakukan dengan membaca buku dan literatur dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan sehingga dapat menunjang dalam penulisan tugas akhir ini. Studi kepustakaan yang dilakukan adalah dengan menggunakan buku yang dimiliki, buku yang dipinjam dari perpustakaan dan mencari data yang diperlukan melalui internet.

3.4 Metode Pengembangan Sistem

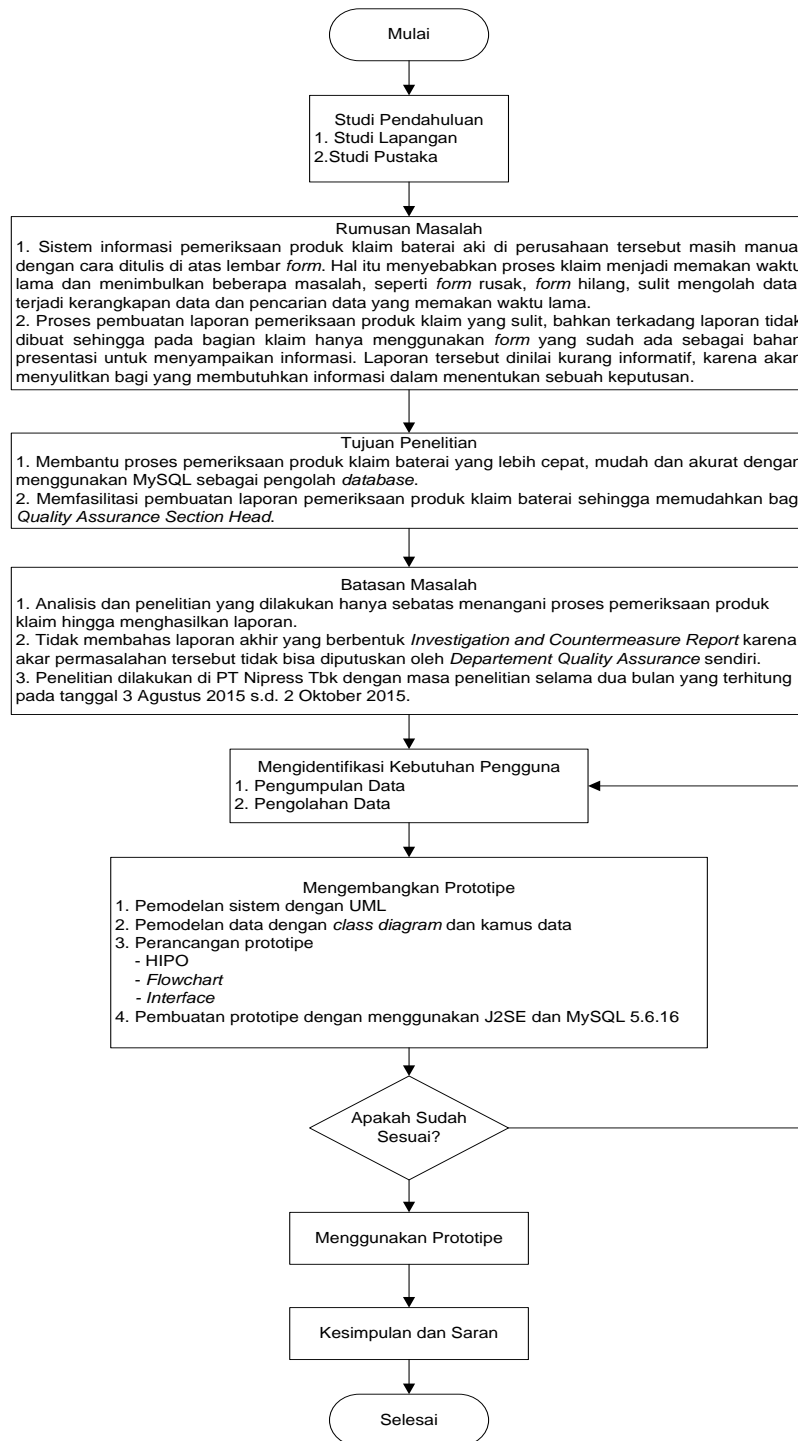
Masalah yang ada pada sistem saat ini telah teridentifikasi pada tahap sebelumnya, dan akan menjadi gambaran untuk mengetahui tindakan yang harus dilakukan untuk mengatasi masalah tersebut. Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk membuat pengembangan sistem. Sistem dikembangkan menjadi lebih terkomputerisasi dan berbasis objek. Pengembangan sistem dilakukan dengan menggunakan metode prototipe evolusioner. Metode prototipe ini terdiri dari tahap identifikasi kebutuhan pengguna, membuat prototipe, menentukan prototipe diterima atau tidak dan menggunakan prototipe.

Berikut ini akan dibahas secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem.

- a. Mengidentifikasi kebutuhan pengguna. Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
- b. Membuat satu prototipe. Pengembang mempergunakan satu alat *prototyping* atau lebih untuk membuat prototipe. Contoh dari alat-alat *prototyping* tersebut adalah generator aplikasi terintegrasi dan *toolkit prototyping*. Generator aplikasi terintegrasi adalah sistem piranti lunak yang siap pakai yang mampu membuat seluruh fitur yang diinginkan dari sistem baru. Sedangkan *toolkit prototyping* meliputi sistem-sistem piranti lunak terpisah, seperti sistem manajemen basis data, yang masing-masing mampu membuat sebagian dari fitur-fitur sistem yang diinginkan.
- c. Menentukan apakah prototipe dapat diterima. Pengembang mendemonstrasikan prototipe kepada pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika iya, akan dilakukan langkah selanjutnya, dan jika tidak, prototipe akan direvisi dari tahap awal dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.
- d. Menggunakan prototipe. Menjadikan prototipe sebagai sistem yang dijalankan.

3.5 Kerangka Pemecahan Masalah

Adapun dalam pembuatan Tugas Akhir ini akan menggunakan kerangka pemecahan masalah seperti yang tertera pada Gambar III.1 sebagai berikut:



Gambar III.1 Kerangka Pemecahan masalah
(Sumber: Pengolahan Data, 2016)

Berikut ini adalah penjelasan dari tahapan kerangka pemecahan masalah adalah sebagai berikut:

- a. Studi Pendahuluan

Pada tahap ini dilakukan analisis sistem informasi klaim yang ada saat ini pada PT Nipress Tbk. Dalam tahap ini terdapat tiga metode yang dilakukan dalam studi pendahuluan pengumpulan data, yaitu:

1. Observasi, yaitu usaha untuk mengumpulkan data dengan cara mengamati secara langsung keadaan maupun kegiatan dalam perusahaan yang kemudian dilakukan pencatatan informasi terhadap objek yang dianggap perlu.
2. Wawancara, yaitu penelitian secara langsung yang dilakukan melalui wawancara terhadap segala hal yang diperlukan pada penyusunan Tugas Akhir ini berdasarkan pada tujuan penelitian.
3. Studi kepustakaan, yaitu dengan membaca buku dan literatur serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan yang ada. Studi kepustakaan yang dilakukan adalah dengan mengunjungi perpustakaan dan mencari data yang diperlukan melalui internet.

b. Identifikasi Masalah

Pada tahap ini melakukan perumusan-perumusan masalah pada sistem yang sedang berjalan, yaitu:

1. Sistem informasi pemeriksaan produk klaim baterai aki di perusahaan tersebut masih manual dengan cara ditulis di atas lembar *form*. Hal itu menyebabkan proses klaim menjadi memakan waktu lama dan menimbulkan beberapa masalah, seperti *form* rusak, *form* hilang, sulit mengolah data, terjadi kerangkapan data dan pencarian data yang memakan waktu lama.
2. Proses pembuatan laporan pemeriksaan produk klaim yang sulit, bahkan terkadang laporan tidak dibuat sehingga pada bagian klaim hanya menggunakan *form* yang sudah ada sebagai bahan presentasi untuk menyampaikan informasi. Laporan tersebut dinilai kurang informatif, karena akan menyulitkan bagi yang membutuhkan informasi dalam menentukan sebuah keputusan.

c. Rumusan Masalah

Rumusan masalah dilakukan dengan menyaring masalah-masalah yang timbul dan sering terjadi pada sistem yang berjalan, dalam penelitian ini dilakukan dengan observasi secara langsung pada sistem yang berjalan.

d. Tujuan Penelitian

Tujuan penelitian yang dilakukan yaitu:

1. Membantu proses pemeriksaan produk klaim baterai yang lebih cepat, mudah dan akurat dengan menggunakan MySQL sebagai pengolah *database*.
2. Memfasilitasi pembuatan laporan pemeriksaan produk klaim baterai sehingga memudahkan bagi *Quality Assurance Section Head*.

e. Batasan Masalah

Batasan masalah yang ada pada laporan Tugas Akhir ini adalah untuk membatasi permasalahan pada ruang lingkup pembuatan program aplikasi dan pembuatan laporan.

f. Identifikasi Kebutuhan Pengguna

Identifikasi kebutuhan pengguna adalah suatu sistem yang sesuai dan diinginkan oleh pengguna (*user*). Setelah data diolah maka perlu diperhatikan identifikasi pengguna yang akan menjalankan sistem tersebut apakah yang *user* inginkan sudah sesuai atau belum, dengan melalui tinjauan studi literatur dan hasil observasi dengan tujuan agar memenuhi kebutuhan *user* dan sekaligus memperkecil kemungkinan-kemungkinan kesalahan dan mendapatkan hasil yang sesuai bila ada kekurangan yang akan terjadi setelah sistem berjalan.

g. Mengembangkan Prototipe

Merancang prototipe sesuai dengan analisis dan perancangan yang dilakukan sebagai berikut:

1. Pemodelan sistem dengan UML.
2. Pemodelan data dengan *Class Diagram* dan kamus data.
3. Perancangan prototipe dengan HIPO, *Flowchart* dan *Interface*.
4. Pembuatan prototipe dengan menggunakan J2SE dan MySQL 5.6.16.

h. Menggunakan Prototipe

Pada tahap ini, prototipe dari sistem yang diusulkan akan digunakan dan dilihat apakah sistem usulan tersebut sudah sesuai dengan keinginan pengguna atau belum. Jika belum, maka akan dilakukan pengumpulan dan analisis data kembali, serta melakukan perancangan sistem ulang. Sedangkan jika sudah sesuai dengan keinginan pengguna, maka prototipe tersebut dapat digunakan.

i. Kesimpulan dan Saran

Setelah semua tahapan selesai, dibuat kesimpulan tentang penelitian sistem informasi klaim. Kesimpulan berkaitan dengan tujuan awal penelitian sistem informasi klaim dilakukan serta usulan yang dapat diberikan kepada penelitian selanjutnya.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sekilas Perusahaan

PT Nipress Tbk adalah salah satu perusahaan manufaktur yang bergerak di bidang pembuatan baterai aki. PT Nipress Tbk didirikan pada tanggal 24 april 1975 sebagai salah satu produsen aki motor dan produsen aki mobil terbesar di indonesia, dengan keputusan Menteri Kehakiman RI pada suratnya No. 72/1975 A.N.P Th. 1975. Tambahan Negeri RI tanggal 25 Mei 1976 / No. 42 Tambahan No. 394.

PT Nipress Tbk memulai operasionalnya sebagai produsen dan supplier aki motor dan aki mobil di pabrik seluas 479.5 m² yang dibangun di atas tanah seluas 6.556 m². PT Nipress Tbk memulai produksi dengan kapasitas 58.500 aki per tahun. Saat ini luas bangunan sudah mencapai 22.650 m² di luas tanah 99.576 m². Perusahaan ini memulai produksi *plate* positif dan negatif untuk baterai motor pada tahun 1982. Pada tahun 1985 diperkenalkanlah indikator baterai untuk mobil yang telah memiliki dukungan teknisi industri ahli dari Jepang sampai 1985, dan PT Nipress Tbk melakukan pengeksporan baterai ke negara ASEAN dan ini merupakan PT pertama yang melakukan pengeksporan ke ASEAN.

Pada tahun 1991, PT Nipress Tbk mengubah statusnya menjadi sebuah perusahaan publik yang tercatat pada bursa efek Jakarta dan bursa efek Surabaya dengan modal ditempatkan dan disetor penuh sebesar Rp 20 miliar. PT Nipress Tbk merupakan satu-satunya perusahaan aki yang menjadi perusahaan publik.

Saat ini, PT Nipress Tbk sudah menjadi penyedia baterai *motive battery* (baterai yang menggerakkan alat) selain untuk kendaraan sehari-hari. PT Nipress Tbk sudah berkolaborasi dengan TNI AL untuk mengembangkan baterai kapal selam. Produksinya sudah berjalan dan rencananya pada tahun 2025 sudah selesai memproduksi 18 baterai masing-masing seberat 240 ton untuk 18 kapal selam. Kerjasama juga dijalin dengan TNI AD dan AU dengan penyediaan baterai khusus untuk tank, kendaraan amfibi, dan pesawat terbang.

Persiapan pabrik baru khusus untuk produksi aki industri dimulai di tahun 2014 dengan luas bangunan sekitar 6.679 m² di luas tanah sekitar 20.000 m² dengan kapasitas produksi sebesar 54.400 unit aki. Sesuai rencana, pabrik ini akan berproduksi secara keseluruhan di kuartal ke-4 tahun 2015.

Selain itu, perusahaan juga mengembangkan *stationary battery* (baterai untuk penyimpanan energi), misalnya cadangan energi untuk *emergency lighting* gedung, BTS berbagai *provider*, atau untuk mem-*back up* energi di data *centre* yang tak boleh sekali pun berkedip. PT Nipress Tbk juga sudah siap dengan penyediaan baterai untuk energi terbarukan seperti piranti penyimpan energi matahari atau pun angin. Pada industri otomotif sendiri (mobil dan sepeda motor), Aki NS sudah berhasil meraih pangsa pasar 20 persen.

4.2 Visi Misi dan Filosofi Perusahaan

PT Nipress Tbk adalah perusahaan manufaktur yang bergerak di bidang pembuatan baterai aki. Perusahaan ini mempunyai visi, misi dan filosofi perusahaan sebagai berikut:

Visi:

Menjadi pemain global dalam solusi energi yang tersimpan.

Misi:

- a. Memberikan kontribusi untuk lingkungan yang lebih hijau.
- b. Memberikan solusi energi yang terjangkau dan dapat diandalkan.
- c. Menyediakan lingkungan belajar yang aman dan nyaman.

Filosofi Perusahaan:

- a. Budaya Hijau

PT Nipress Tbk berusaha untuk mencapai keunggulan di semua bidang, dengan demikian perusahaan menjamin bahwa proses produksi tidak akan membawa kerugian bagi lingkungan, dengan budaya hijau kami, perusahaan tegas berfokus

pada menciptakan sistem produksi yang baik dan efisien tanpa mengorbankan keberadaan Alam.

b. *Grand Inovasi*

Kami terus menerus meningkatkan dan menciptakan produk dengan metode pengembangan menggunakan teknologi canggih.

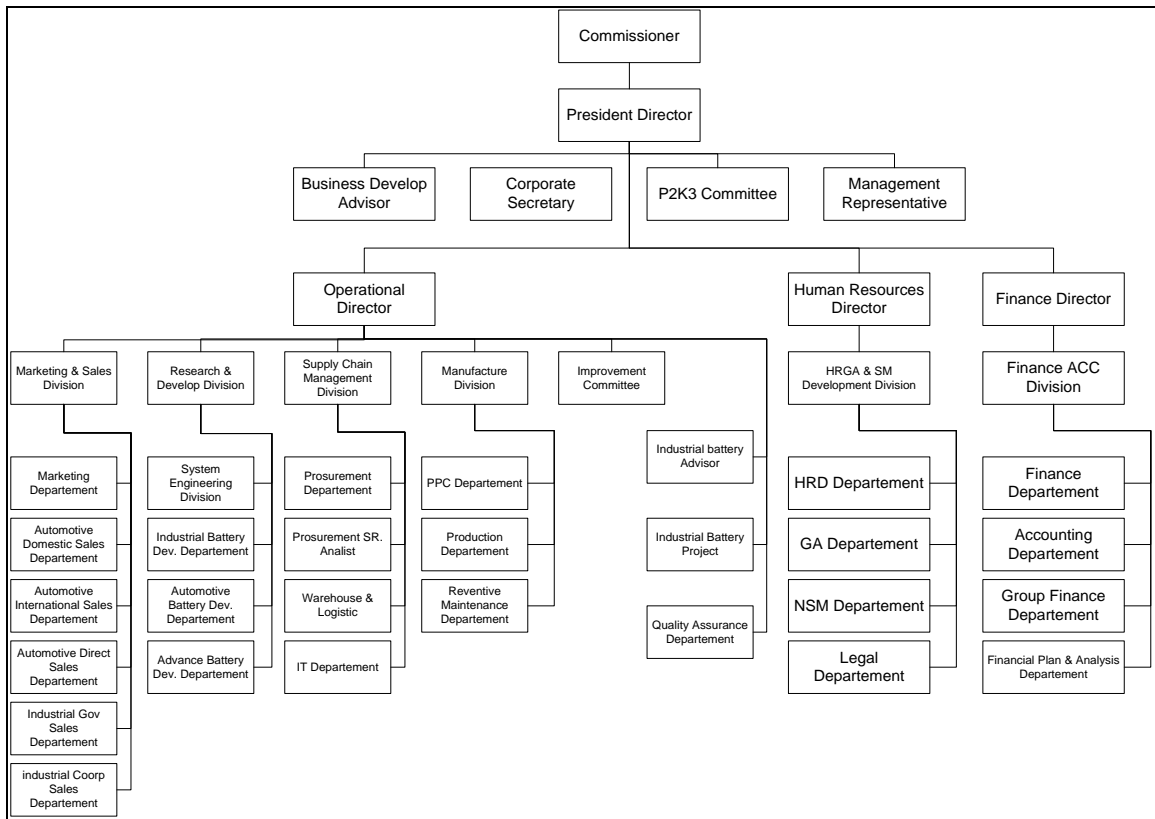
c. *Kehadiran Global*

PT Nipress Tbk memberikan pelayanan yang terbaik untuk melayani kebutuhan pelanggan dengan jaringan pasokan kami di seluruh dunia, dan kami terus memberikan produk-produk berkualitas tinggi kepada pelanggan.

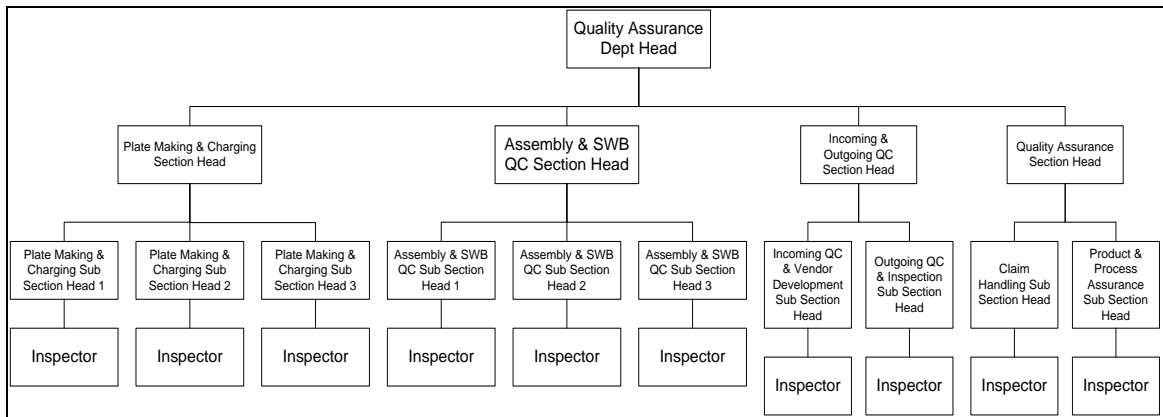
4.3 Struktur Organisasi Perusahaan

Untuk menjalankan usahanya, setiap perusahaan memerlukan suatu struktur organisasi yang baik. Struktur organisasi dapat diartikan sebagai susunan dan hubungan antar bagian dan posisi dalam suatu perusahaan. Suatu struktur organisasi menggambarkan pembagian kerja, pelimpahan wewenang, kesatuan perintah dan tanggung jawab yang jelas.

Struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektivitas dan efisiensi suatu perusahaan di dalam mencapai tujuannya. Berikut adalah struktur organisasi PT Nipress Tbk secara umum dan *Quality Assurance Departement* yang digambarkan pada Gambar IV.1 dan Gambar IV.2.



Gambar IV.1 Struktur Organisasi PT Nipress Tbk
(Sumber: PT Nipress Tbk, 2015)



Gambar IV.2 Struktur Organisasi *Quality Assurance Departement*
(Sumber: PT Nipress Tbk, 2015)

4.4 Peran dan Fungsi Departemen *Quality Assurance*

Berikut ini adalah peran dan fungsi dari bagian Departemen *Quality Assurance* pada PT Nipress Tbk:

1. Menjamin semua material yang diterima, proses produksi, *half finish* produk, *finish good* produk sesuai dengan standar PT Nipress Tbk dan *customer requirement*.
2. Memastikan bahwa produk baru ataupun perubahan produk sesuai dengan standar PT Nipress Tbk dan atau sesuai dengan *customer requirement*.
3. Melakukan *vendor development* bersama dengan *departement procurement*.
4. Mengidentifikasi dan mengendalikan barang yang tidak sesuai (*non conforming product*) di setiap proses produksi.

4.5 Tugas dan Tanggung Jawab *Claim Handling* dan *Quality Assurance Section*

1. Melakukan analisis klaim/*complaint handling customer* (OEM, *fleet*, *after market* dan industrial) sesuai dengan prosedur klaim.
2. Melakukan *meeting* koordinasi (*meeting claim*) dengan departemen lain untuk mencari akar permasalahan dan penetapan *countermeasure*-nya.
3. Melakukan *support* pelaksanaan prosedur PDPP (*Product Develop Planning Process*), melakukan *initial stage control* dan *vendor development* bersama dengan *purchasing departement*.
4. Melakukan *training customer* klaim/*complain handling* dan *product knowledge*.
5. Melakukan evaluasi dan pengembangan terhadap sistem klaim/*complaint handling* yang disesuaikan dengan *customer requirement*, serta memonitoring *countermeasure* yang dilakukan.

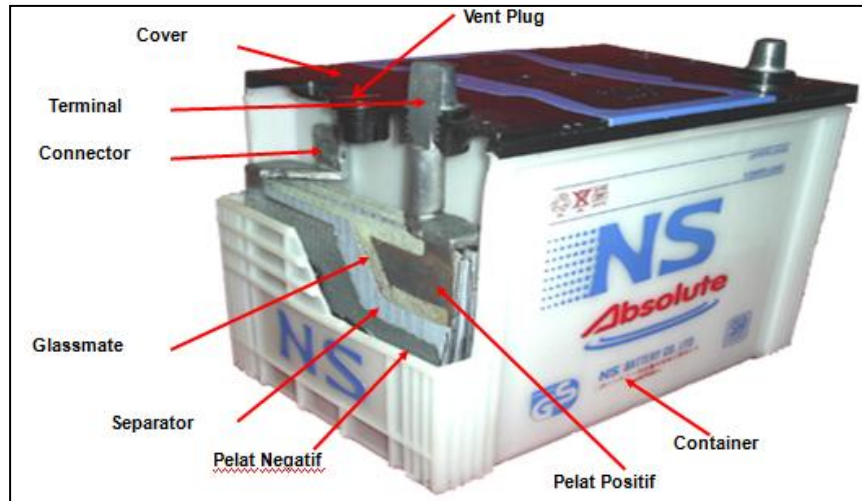
4.6 Produk Perusahaan

PT Nipress Tbk menghasilkan berbagai macam jenis baterai aki. Berikut ini adalah hasil produksi baterai aki yang digolongkan berdasarkan peruntukannya sebagai berikut:

1. AMB (*Automobile Battery*)
 - a. *Dry Charge Battery*
 - b. *Hybird Battery*
 - c. *Maintenance Free*
 - d. *Lithium Battery Electric Vehicle*
2. MCB (*Motorcycle Battery*)
 - a. *Dry Charge Battery*
 - b. *VRLA Battery*
 - c. *Spiral Battery*
 - d. *Spiral Single Battery Electric Bike (Pengembangan)*
3. Baterai Industri (*Industry Battery*)
 - a. *PLTS dan Telecommunication Battery*
 - b. *Defense dan Aviation Battery*
 - c. *Renewable Battery*
 - d. *Marine Defense Battery (Pengembangan)*
 - e. *Golf Cart Battery*
 - f. *Forklift Battery*
 - g. *Telecommunication Battery*

4.7 Komponen Baterai dan Fungsinya

Komponen-komponen baterai aki terdiri atas beberapa bagian yang masing-masing memiliki fungsi tertentu, antara lain sebagai berikut (lihat Gambar IV.3 dan Tabel IV.1):



Gambar IV.3 Struktur Baterai Aki
(Sumber: PT Nipress Tbk, 2015)

Tabel IV.1 Komponen Baterai Aki dan Fungsinya

No	Komponen	Fungsi
1	<i>Vent Plug</i>	Untuk menutup tiap lubang pengisian cairan elektrolit dan berfungsi juga untuk memisahkan gas hidrogen dengan uap asam sulfat pada saat baterai melakukan pengisian.
2	<i>Cover</i>	Sebagai penutup atas baterai aki.
3	Terminal	Untuk menyalurkan atau menghubungkan arus listrik ke dalam rangkaian.
4	<i>Connector</i>	Berfungsi untuk menghubungkan antara sel negatif dengan sel negatif dan antara sel positif dengan sel positif.
5	<i>Glassmate</i>	Sebagai pelapis dan pemisah agar plat tidak bersentuhan langsung dengan separator.
6	<i>Separator</i>	Untuk memisahkan antara plat positif baterai dengan plat negatif baterai.
7	Plat Negatif dan	Salah satu yang mempengaruhi kemampuan baterai dalam mengalirkan arus adalah jumlah dan ukuran plat. Semakin besar atau banyak platnya maka semakin besar pula arus yang dihasilkan.
	Plat Positif	

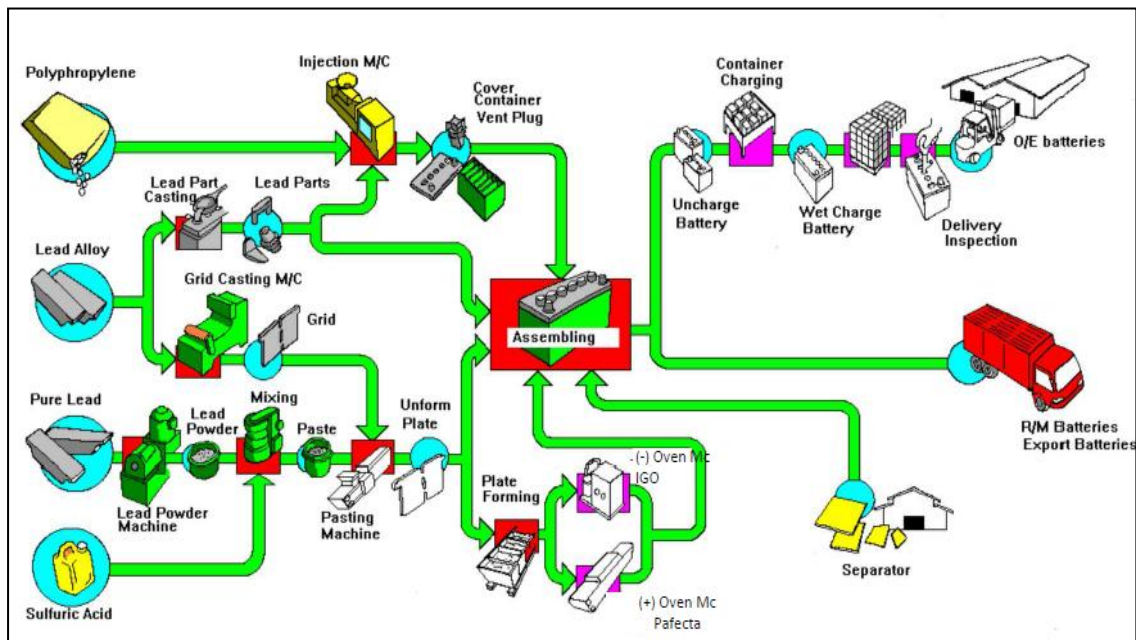
Tabel IV.1 Komponen Baterai Aki dan Fungsinya (lanjutan)

No	Komponen	Fungsi
8	Container	Sebagai wadah untuk menampung seluruh komponen baterai dan melindungi cairan elektrolit ke lingkungan luar.

Sumber: PT Nipress Tbk (2015)

4.8 Proses Produksi Baterai Aki

Dalam proses pembuatan baterai aki, terdapat lima tahapan produksi yang harus dilalui sebelum produk dapat dikirimkan kepada konsumen. Tahapan tersebut meliputi *plastic casting*, *small part casting*, *plate making*, *assembling* dan *charging*. Secara garis besar alur produksinya telah digambarkan secara lengkap pada Gambar IV.4 sebagai berikut.



Gambar IV.4 Alur Proses Produksi Baterai Aki
(Sumber: PT Nipress Tbk, 2015)

4.9 Jenis-Jenis Kerusakan Baterai Aki

Kualitas baterai merupakan penilaian akan mutu baterai yang akan diproduksi, salah satunya adalah kerusakan. Kerusakan menjadikan landasan atau pelajaran akan kesalahan yang mungkin terjadi pada proses produksi baterai yang dapat menjadi alat untuk mengatasi permasalahan yang terjadi sehingga memudahkan untuk meminimalisasi kerugian. Kerusakan yang dialami baterai aki per-tahun 2015 pada PT Nipress Tbk dibagi menjadi dua jenis yaitu kesalahan pabrik dan kesalahan *customer*. Hal itu dapat dilihat pada Tabel IV.2 dan Tabel IV.3 berikut ini:

Tabel IV.2 Kerusakan Akibat Kesalahan Pabrik

No	Kerusakan
1	Terminal Pole (putus (pole inside tidak terbakar))
2	Terminal Pole (putus (pole inside terbakar))
3	Terminal Pole Hitam (bocor / leakage pole)
4	Short (plate (+) dan (-) naik)
5	Short (plate bengkok/bent plate)
6	Short (separator sobek/ripped separator)
7	Short (lead run strap / busbar)
8	Short (plate flashing)
9	Short (pasta tebal (nempel)/material paste)
10	Short (Sedding paste/rontokan pasta)
11	Press Welder (cold weld)
12	Press Welder (no weld)
13	Press Welder (lead splash)
14	Las lasan Plate Lepas pada strap (disconnecting lug)
15	Las lasan Bush Bar/strap Patah / Lepas (strap damage crack/broken)
16	Lug Plate Patah (lug of plate broken)
17	Plate Negative Rontok (negative plate fall out)
18	Plate Positive Crack (positive plate broken/crack)
19	Plate Undercharge (wet charge battery)
20	Plate Belang Kuning (dry charge battery)
21	Polaritas Terbalik (reverse element)
22	Heat Seal Bocor (leakage heat sealing machine)
23	Container Bocor sisi runner gate
24	Salah Cover / Type
25	Fold Handle Belum diultrasonic
26	Fold Handle Lepas / Putus
27	Alluminium Foil Rusak / Lepas
28	Indicator / Mata Kucing Tidak Berfungsi
29	Printing / Sticker : Cacat / Salah
30	Box dengan Battery Tidak Sesuai (Salah isi battery) (lanjutan)
31	Separator motor Kotor (Battery belum dipakai/diisi elektrolit)
32	Separator Motor Tidak Rapih/Miss alignment separator (Battery belum dipakai/diisi

Tabel IV.2 Kerusakan Akibat Kesalahan Pabrik

No	Kerusakan
	elektrolit)
33	Sambungan Kabel Lepas / Putus
34	Kit Air Zuur Battery Tawar (salah isi kit air accu zuur)
35	Battery Meledak, Sumber Api dari Dalam
36	Short separator panjang/pendek sebelah (Un Squaring Plate)
37	Vent Plug Doll
38	Short separator terlipat
39	Leakage (Untuk MF dan Gel)
40	Lain Lain (Kesalahan Pabrik)

Sumber: PT Nipress Tbk (2015)

Tabel IV.3 Kerusakan Akibat Kesalahan Customer

No	Kerusakan
1	Battery Meledak, Sumber Api dari Luar
2	Terminal Pole Cacat Pakai
3	Container Pecah / Cacat Pakai
4	Cover Pecah / Cacat Pakai
5	Accu Lama / Bekas Pakai /Lama Pakai
6	Stock Lama Simpan
7	Undercharge (charge ulang)
8	Overcharge
9	Sulfation/Over discharge
10	Elektrolite Cell Kosong/Kering/Kurang/sudah diisi elektrolit tidak dijaga Level
11	Salah Isi Elektrolite
12	Kardus Rusak / Cacat pakai
13	Lubang Vent Plug Rusak / Cacat Pakai
14	Tegangan battery terbalik (Charge terbalik/pemasangan terbalik)
15	SpGr Tinggi (≥ 1.300)
16	Terminal Pole Berubah Bentuk (Bekas dicor ulang)
17	Cover pecah kering/isi terminal pole amblas
18	Battery belum diisi elektrolit, Battery baru (Jika Tidak jelas Claim nya)
19	Lain Lain Kesalahan Customer

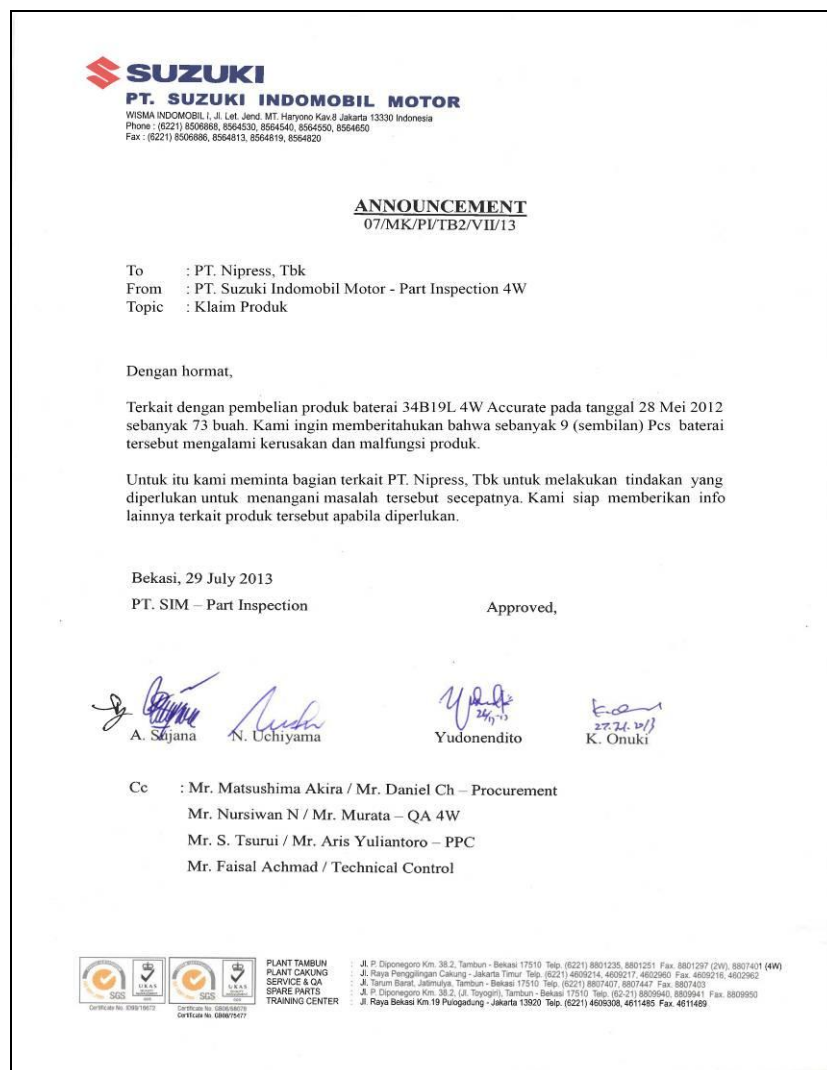
Sumber: PT Nipress Tbk (2015)

4.10 Dokumen Proses Klaim

Dalam proses klaim yang dilakukan oleh *Claim Handling Sub Section* pada PT Nipress Tbk masih dilakukan secara manual dengan menggunakan *form-form* kertas sehingga menimbulkan beberapa masalah. Dokumen yang terlibat dalam proses klaim baterai aki adalah sebagai berikut:

1. *Klaim Customer*


Klaim yang diajukan oleh *customer* terdiri dari dua bentuk yaitu email dan telpon. Klaim yang diajukan dengan telpon akan direkam percakapannya oleh bagian *marketing*. Sedangkan bentuk klaim yang diajukan dengan email salah satunya seperti Gambar IV.5 berikut.



Gambar IV.5 Contoh Klaim Dengan Email
(Sumber: PT Nipress Tbk, 2015)

2. *Memo External*

Memo *external* merupakan sebuah alat yang digunakan untuk menuliskan sebuah pemberitahuan antar bagian di PT Nipress Tbk. Dalam proses klaim, memo *external* digunakan bagian *marketing* untuk mengkonfirmasi adanya klaim produk ke bagian *claim handling* (lihat Gambar IV.6).

		MEMO EXTERNAL	
FROM : Marketing		TO : Claim Handling Sub Section	
Copy To	QAS	CHS	⇒
	MKT	FRW	
	PPD	FFG	
	HRD	ACC	
	PRD	PRC	
	BOD	PPIC	
	QSM	WorkShop	
No. Memo	: MM/MKT/001/15		
Date	: 8 Januari 2015		
Page	: 1 to 1		

Hal : Pengajuan Klaim

Dengan hormat,

Terkait dengan diajukannya klaim produk dari PT Astra Honda Motor-Sunter. Untuk itu kami team Marketing meminta bantuan kepada team Claim Handling agar menerima produk terkait pada tanggal 10 Januari 2015/11 Januari 2015 untuk dilakukan proses klaim.

Note:
12N20 Absolute = 5 Pcs
12N5-3B Absolute = 2 Pcs

Atas perhatian dan kerja samanya kami ucapkan terima kasih.

Mengetahui, Hormat Saya,

() ()

Gambar IV.6 Memo *External*
(Sumber: PT Nipress Tbk, 2015)

3. LPB (Lembar Penerimaan barang)


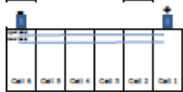
Lembar Penerimaan Barang merupakan lembar kertas yang berisikan data baterai aki serta perusahaan atau distributor yang ingin melakukan klaim baterai (lihat Gambar IV.7).

LAPORAN PENERIMAAN BARANG (LPB) ACCU COMPLAIN					
Tanggal terima :		NO LPB :		/ /	
Dari :					
No Surat Jalan :		Tanggal :			
NO	MERK	JENIS	TYPE	QTY	KETERANGAN
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
YANG MENYERAHKAN			YANG MENERIMA		
Putih : Gudang Complain (QC)		Biru : Accounting	Merah : gudang jadi	Kuning : Marketing	
SD/FR/QAS/086 00 08 JUNI 2012					

Gambar IV.7 Lembar Penerimaan Barang
(Sumber: PT Nipress Tbk, 2015)

4. AKB (Analisis Kondisi Baterai)

Analisis Kondisi Baterai merupakan lembar kertas yang berisikan data analisis baterai aki untuk menentukan akar permasalahan sementara (Lihat Gambar IV.8).

	Laporan Claim Pelanggan PT. NIPRESS Distributor PT. Dealer			No Battery																										
				ASSY	AMB/MCB/IND																									
				No. LPB																										
Type Battery		Kode Produksi		SJ Claim																										
Km Kendaraan		Kode Delivery		No Claim Cust																										
Tanggal Pakai		Tanggal Turun		Tanggal terima Claim																										
I. Pemeriksaan Awal Penampilan Cover-Container : <input type="checkbox"/> Baik <input type="checkbox"/> Pecah dry <input type="checkbox"/> Pecah Chg <input type="checkbox"/> Kembung <input type="checkbox"/> Kotor <input type="checkbox"/> Sealing (HSM) : <input type="checkbox"/> Baik <input type="checkbox"/> Bocor sisi panjang <input type="checkbox"/> Bocor sisi Pendek <input type="checkbox"/> Bocor antar sel <input type="checkbox"/> Penampilan Terminal Pole : <input type="checkbox"/> Baik <input type="checkbox"/> Penyok <input type="checkbox"/> Bolong <input type="checkbox"/> Crack <input type="checkbox"/> Bocor <input type="checkbox"/> Electrolyte : <input type="checkbox"/> Kosong <input type="checkbox"/> Kurang <input type="checkbox"/> Cukup <input type="checkbox"/> Over Upper Level  Beri Tanda jika elektrolit lebih atau kurang dengan memberi garis pada gambar Sticker Aluminium Foil : <input type="checkbox"/> Sudah di buka <input type="checkbox"/> Belum di buka (Khusus battery dry baru) Tegangan : Volt <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><td>cca</td><td></td></tr> <tr><td>R</td><td></td></tr> </table> <table border="1" style="display: inline-table;"> <tr><td>Cell</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>Sp Gr/20°C</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>Voltage</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>						cca		R		Cell	1	2	3	4	5	6	Sp Gr/20°C							Voltage						
cca																														
R																														
Cell	1	2	3	4	5	6																								
Sp Gr/20°C																														
Voltage																														
II. Keadaan Setelah Di Recharge Tegangan tertutup Vlt <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>cca</td><td></td></tr> <tr><td>R</td><td></td></tr> </table> Voltage setelah Equalisasi Vlt <table border="1"> <tr><td>Cell</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> <tr><td>Sp Gr/20°C</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>Voltage</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>						cca		R		Cell	1	2	3	4	5	6	Sp Gr/20°C							Voltage						
cca																														
R																														
Cell	1	2	3	4	5	6																								
Sp Gr/20°C																														
Voltage																														
III. Test Kemampuan Battery Load Tester <table border="1"> <thead> <tr> <th>Test Parameter</th> <th>Standard</th> <th>Hasil</th> <th>Kesimpulan</th> </tr> </thead> <tbody> <tr> <td rowspan="3">BT Load Test (400 untuk mobil dan 350A untuk motor)</td> <td>Hijau</td> <td></td> <td rowspan="3">N. OK/OK</td> </tr> <tr> <td>Kuning</td> <td></td> </tr> <tr> <td>Merah</td> <td></td> </tr> </tbody> </table> <p style="text-align: center;">Load Test, jika battery keluar gelembung saat test beban maka langsung bedah / analisa</p>						Test Parameter	Standard	Hasil	Kesimpulan	BT Load Test (400 untuk mobil dan 350A untuk motor)	Hijau		N. OK/OK	Kuning		Merah														
Test Parameter	Standard	Hasil	Kesimpulan																											
BT Load Test (400 untuk mobil dan 350A untuk motor)	Hijau		N. OK/OK																											
	Kuning																													
	Merah																													
IV. HASIL CHECK POTONG BATTERY. Kondisi : 1 Plate (+) : <input type="checkbox"/> Bagus <input type="checkbox"/> Memutih <input type="checkbox"/> Rapuh/Rusak <input type="checkbox"/> Mengeras <input type="checkbox"/> Coklat muda / terang <input type="checkbox"/> Coklat tua / Gelap <input type="checkbox"/> Grid / Plate crack <input type="checkbox"/> 2 Plate (-) : <input type="checkbox"/> Bagus <input type="checkbox"/> Memutih <input type="checkbox"/> Mengeras <input type="checkbox"/> Lunak / Rontok <input type="checkbox"/> Tidak ada kilap metalik <input type="checkbox"/> 3 Separator : <input type="checkbox"/> Bagus <input type="checkbox"/> Rapuh <input type="checkbox"/> Warna lebih gelap / Menghitam <input type="checkbox"/> Lengket pada plate <input type="checkbox"/> 4 Endapan : <input type="checkbox"/> Ada <input type="checkbox"/> Tidak Ada <input type="checkbox"/>																														
V. KESIMPULAN / KETERANGAN <input type="checkbox"/> Kode Kesalahan Analisa				<input type="checkbox"/> Manufacturing Fault (1) <input type="checkbox"/> Ekspedition Fault (2) <input type="checkbox"/> Handling Fault (3)	Tanggal Petugas Diperiksa Diketahui																									
SD/FR/QAS/021 02 10/09/2012																														

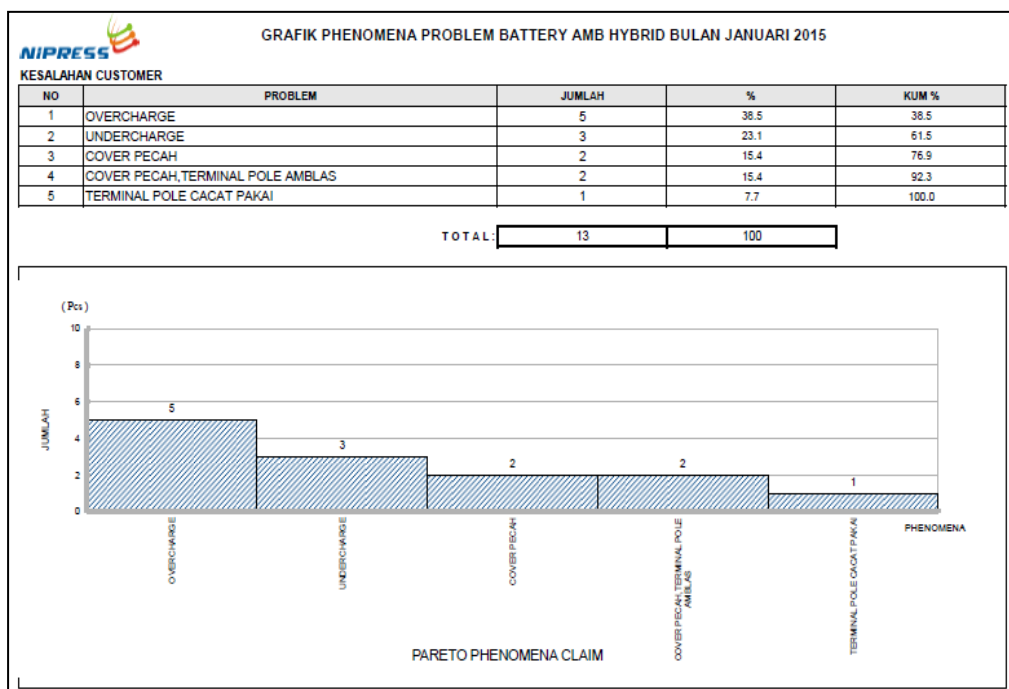
Gambar IV.8 Lembar Analisis Kondisi Baterai
(Sumber: PT Nipress Tbk, 2015)

4.11 Bentuk Informasi Klaim Baterai

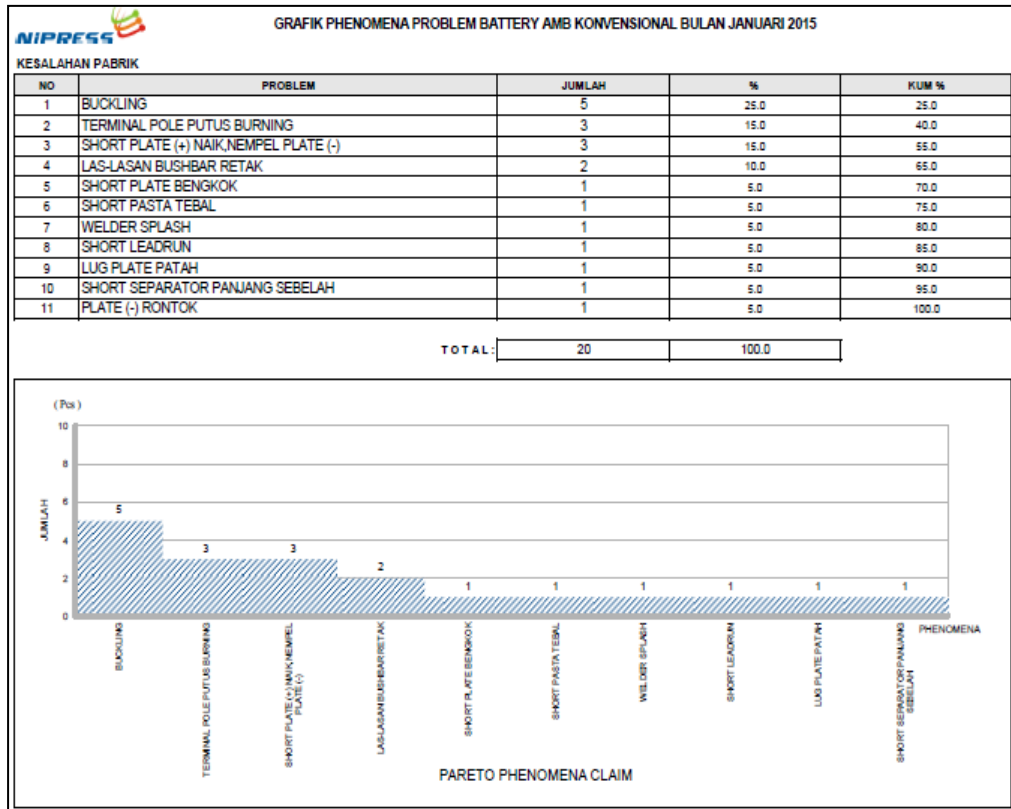
Bentuk informasi yang dihasilkan dari data *Claim Handling Sub Section* dibuat dalam bentuk tabel dan diagram berdasarkan jenis kerusakannya. Berikut ini merupakan bentuk informasi yang ada di dalam bagian *Quality Assurance Section Head*:

1. Menggunakan Grafik Fenomena Kerusakan Baterai

Grafik fenomena kerusakan baterai dibuat berdasarkan jenis kerusakan yang ditemukan. Laporan ini memiliki dua tipe yaitu laporan berdasarkan kesalahan *customer* dan kesalahan pabrik (Lihat Gambar IV.9 dan Gambar IV.10)



Gambar IV.9 Grafik Fenomena Kerusakan Baterai (*Customer Fault*)
(Sumber: PT Nipress Tbk, 2015)



Gambar IV.10 Grafik Fenomena Kerusakan Baterai (*Factory Fault*)
(Sumber: PT Nipress Tbk, 2015)

2. Menggunakan *Form* Analisis Kondisi Baterai

Form analisis kondisi baterai digunakan sebagai lembar pengisian untuk menganalisis kerusakan baterai dari pemeriksaan awal, keadaan setelah melakukan *charge*, tes kemampuan, hasil cek potong baterai, hingga kesimpulan. Selain itu, *form* tersebut secara langsung digunakan perusahaan sebagai alat untuk melaporkan kerusakan baterai aki yang bersifat sementara. Laporan tersebut nantinya akan dievaluasi dan dibicarakan kembali bersama-sama dengan departemen terkait lainnya.

4.12 Prosedur Klaim

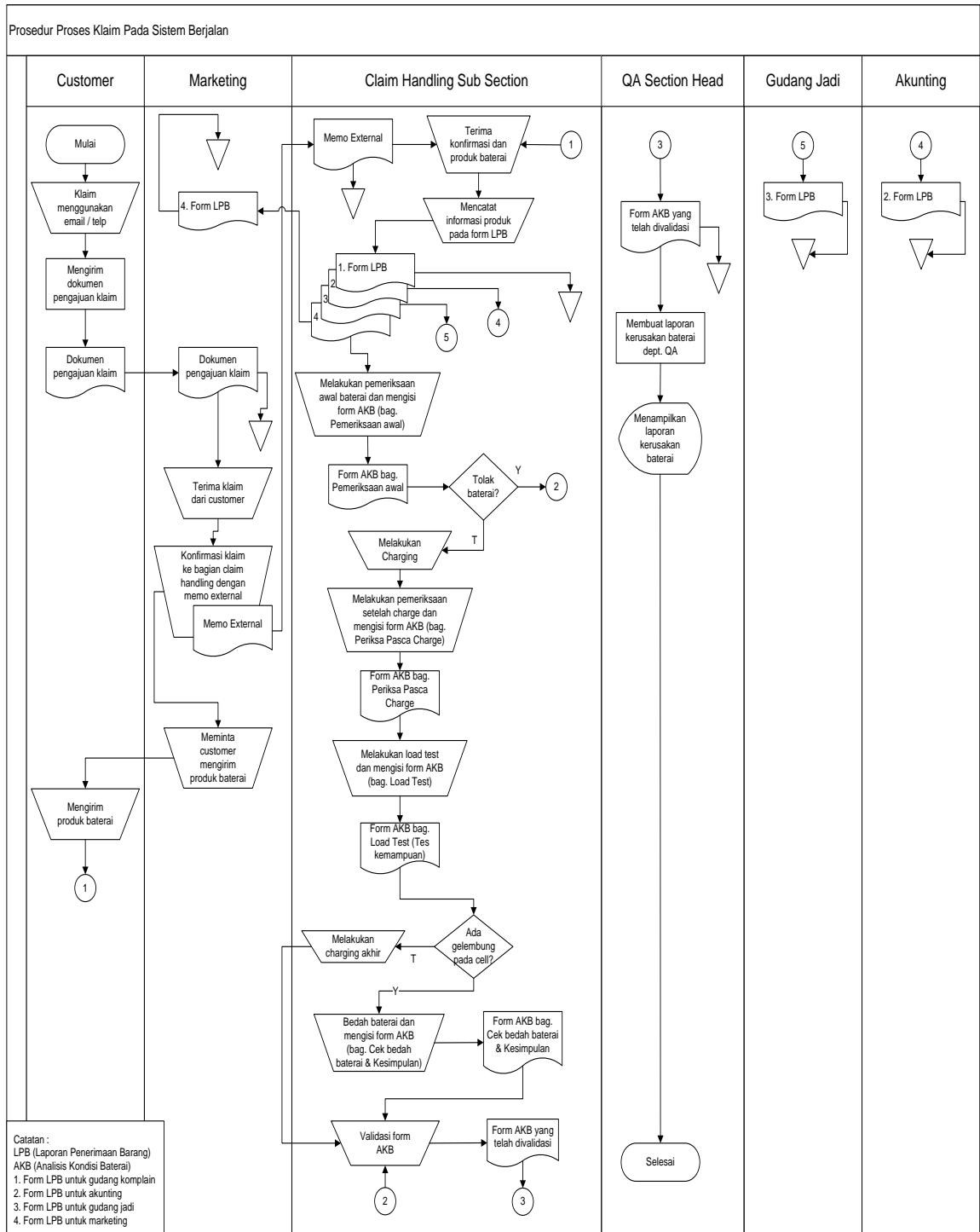
Prosedur klaim baterai aki pada PT Nipress Tbk memiliki prosedur yang ditangani oleh beberapa bagian seperti *Marketing*, *Claim Handling Sub Section*, dan *Quality Assurance Section Head*. Adapun prosedurnya terdiri dari beberapa tahapan, yaitu:

1. *Customer* melakukan klaim terhadap perusahaan dengan menggunakan email atau dengan melakukan panggilan telpon.
2. *Marketing* menerima klaim baterai aki dari *customer* dan meminta *customer* untuk mengirimkan produknya ke PT Nipress Tbk.
3. *Customer* mengirim produk baterai.
4. Bagian *Claim Handling Sub Section* menerima produk baterai dan melakukan pengisian *form* LPB (Lembar Penerimaan Barang) untuk mendapatkan data barang dan asal barang.
5. Bagian *Claim Handling Sub Section* melakukan pemeriksaan awal baterai untuk mengetahui kondisi fisik baterai. Selanjutnya, bagian *Claim Handling* mengisi *form* AKB (Analisis Kondisi Baterai) pada bagian I (pertama). Apabila kondisi baterai tidak memungkinkan untuk dilanjutkan pada proses kalim selanjutnya, maka baterai akan dikembalikan kepada *customer*.
6. Bagian *Claim Handling Sub Section* melakukan *charging* pada baterai aki tergantung pada tingkatan SpGr dan perhitungan pada masing-masing baterai.
7. Setelah masa *charging*, baterai harus didiamkan selama tiga hari.
8. Bagian *Claim Handling Sub Section* melakukan pemeriksaan tegangan kembali dan melakukan tes kemampuan baterai aki dengan menggunakan *Battery Load Tester*. Selanjutnya, mengisi *form* AKB (Analisis Kondisi Baterai) pada bagian II (kedua) dan III (ketiga).
9. Apabila pada tes kemampuan tidak ditemukan gelembung udara pada *cell* dan indikator menunjukkan warna hijau, maka baterai menunjukkan dalam keadaan baik lalu di-*charge* kembali selama lima jam dan selanjutnya dapat dikembalikan pada *customer*.
10. Apabila pada tes kemampuan ditemukan gelembung udara pada *cell* dan indikator menunjukkan warna merah, maka baterai masuk pada tahap pembedahan.

Selanjutnya, bagian *Claim Handling* mengisi *form* AKB (Analisis Kondisi Baterai) pada bagian IV (keempat).

11. Bagian *Claim Handling Sub Section* mengisi kesimpulan pada *form* AKB (Analisis Kondisi Baterai) pada bagian V (kelima).
12. Setelah *form* AKB (Analisis Kondisi Baterai) divalidasi, maka selanjutnya bagian *Quality Assurance Section Head* membuat laporan permasalahan sementara.

Adapun aliran data klaim baterai aki dapat dilihat pada *flowchart* di bawah ini (Gambar IV.11):



Gambar IV.11 Aliran Dokumen Sistem yang Berjalan
 (Sumber: Pengumpulan dan Pengolahan Data, 2015)

4.13 Sistem Berjalan Dengan *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*behavior*) suatu sistem. *Use case* mendeskripsikan sebuah interaksi antara salah satu atau lebih aktor dengan suatu sistem (Rosa dan Shalahuddin, 2013).

Use case diagram klaim baterai aki yang sedang berjalan pada *Quality Assurance Section Head* terdiri dari empat aktor sebagaimana digambarkan pada Gambar IV.12 di bawah ini:



Gambar IV.12 Use Case Sistem yang Berjalan
 (Sumber: Pengumpulan dan Pengolahan Data, 2015)

Penjelasan *use case diagram* proses klaim yang sedang berjalan pada bagian *Claim Handling Sub Section* di PT Nipress Tbk adalah sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada *use case diagram* proses klaim yang sedang berjalan pada bagian *Claim Handling Sub Section* di PT Nipress Tbk dapat dilihat pada Tabel IV.4 berikut:

Tabel IV.4 Definisi Aktor Proses Klaim

No	Aktor	Definisi
1	<i>Customer</i>	<i>Customer</i> sebagai pemohon klaim baterai aki terhadap PT Nipress Tbk.
2	<i>Marketing</i>	<i>Marketing</i> sebagai penerima klaim dari <i>customer</i> , mengkonfirmasi ke bagian <i>claim handling</i> dan memverifikasi kelengkapan produk baterai aki yang akan dikirim nantinya. <i>Marketing</i> juga bertugas memberi informasi kepada <i>customer</i> mengenai apapun yang berhubungan dengan proses klaim baterai aki milik <i>customer</i> yang terkait.
3	<i>Claim Handling Sub Section</i>	<i>Claim Handling Sub Section</i> adalah bagian yang bertanggung jawab terhadap penerimaan baterai aki dari <i>customer</i> dan proses klaim yang berlangsung bersama dengan para <i>inspector</i> bagian <i>claim handling</i> .
4	<i>QA Section Head</i>	<i>QA Section Head</i> merupakan bagian yang bertanggung jawab terhadap bagian <i>Claim Handling Sub Section</i> . Bagian tersebut menerima <i>form</i> dari bagian <i>claim handling</i> lalu membuat laporan kerusakan baterai klaim, melakukan <i>meeting</i> dengan departemen terkait lainnya dan nantinya bertugas membuat laporan <i>countermeasure</i> akar permasalahan sebenarnya.

Sumber: Pengumpulan dan Pengolahan Data (2015)

2. Definisi Use Case

Pendefinisian *use case* pada *use case diagram* proses klaim baterai aki pada bagian *Claim Handling Sub Section* di PT Nipress Tbk dapat dilihat pada Tabel IV.5 berikut:

Tabel IV.5 Definisi *Use Case* Proses Klaim

No	<i>Use Case</i>	Definisi
1	Klaim menggunakan email / telp	Proses yang dilakukan <i>customer</i> saat ingin melakukan klaim terhadap produk baterai miliknya pada PT Nipress Tbk.
2	Mengirim dokumen pengajuan klaim	<i>Customer</i> Mengirim dokumen permohonan klaim menggunakan email.
3	Menerima dokumen pengajuan klaim	<i>Marketing</i> menerima dokumen pengajuan klaim dan melakukan pemanggilan balik ke <i>customer</i> untuk pemberian info klaim yang lebih lengkap.
4	Terima klaim dari <i>customer</i>	Dilakukan oleh bagian <i>marketing</i> sebagai tahap awal proses klaim sekaligus memberikan informasi yang dibutuhkan kepada <i>customer</i> mengenai proses klaim.
5	Konfirmasi klaim ke bagian <i>claim handling</i>	Dilakukan oleh <i>marketing</i> untuk mencegah adanya <i>miss communication</i> dan memberi informasi ke bagian terkait akan adanya baterai klaim dari <i>customer</i> .
6	Meminta <i>customer</i> mengirim produk baterai	<i>Marketing</i> meminta <i>customer</i> untuk mengirim produk baterai yang ingin diklaim.
7	Mengirim produk baterai	Proses pengirimam baterai aki yang akan diklaim oleh <i>customer</i> .
8	Terima konfirmasi dan produk baterai	Konfirmasi baterai <i>customer</i> telah diterima dari <i>marketing</i> , baterai aki telah sampai diperusahaan dan langsung ditangani oleh bagian <i>claim handling</i> .
9	Mencatat informasi produk pada <i>form</i> LPB	Bagian <i>claim handling</i> akan membuat <i>form</i> LPB (Laporan Penerimaan Barang) untuk mendata mengenai produk baterai yang diterima dan asal produk tersebut.
10	Melakukan pemeriksaan awal baterai dan mengisi	Memastikan kondisi awal baterai sebagai langkah

Tabel IV.5 Definisi *Use Case* Proses Klaim

No	<i>Use Case</i>	Definisi
	<i>form</i> AKB (bag. Pemeriksaan awal)	dasar proses klaim. Pemeriksaan SpGr dan voltase tiap <i>cell</i> pada baterai juga dilakukan untuk menentukan lama masa charge. Proses ini dicatat pada lembar AKB (Analisis Kondisi baterai) pada bagian I (Pertama). Apabila kondisi baterai tidak memungkinkan untuk dilanjutkan pada tahap selanjutnya, maka baterai aki dikembalikan pada <i>customer</i> . (lanjutan)
11	Melakukan <i>charging</i>	Proses mengisi daya baterai untuk memastikan produk baterai aki dalam keadaan <i>full charge</i> sebelum memasuki proses selanjutnya. Proses <i>charge</i> dapat berlangsung selama 5 s.d. 20 jam tergantung pada perhitungannya. selanjutnya, baterai akan didiamkan selama tiga hari. Proses <i>charging</i> juga dilakukan kembali apabila setelah melakukan <i>load test</i> tiap <i>cell</i> baterai tidak menunjukkan adanya gelembung udara.
12	Melakukan pemeriksaan setelah <i>charge</i> dan mengisi <i>form</i> AKB (bag. Periksa Pasca <i>Charge</i>)	Setelah dilakukan pendiaman selama tiga hari, baterai aki kembali diperiksa SpGr dan voltase tiap <i>cell</i> baterai. Proses ini dicatat pada lembar AKB (Analisis Kondisi baterai) pada bagian II (Kedua).
13	Melakukan <i>load test</i> dan mengisi <i>form</i> AKB (bag. <i>Load Test</i>)	Memastikan tiap <i>cell</i> baterai tidak terdapat gelembung udara dan memastikan produk baterai menunjukkan indikator hijau pada alat <i>load tester</i> . Hal itu menunjukkan bahwa baterai masih dalam keadaan baik. Apabila indikator menunjukkan warna merah, maka dapat dipastikan terdapat gelembung udara dan baterai harus dibedah. Proses ini dicatat pada lembar AKB (Analisis Kondisi baterai) pada bagian III (Ketiga).
14	Bedah baterai dan mengisi <i>form</i> AKB (bag. Cek bedah baterai & Kesimpulan)	Proses pembedahan dilakukan untuk mengetahui permasalahan pada baterai. Memastikan juga pada <i>part</i> mana yang membuat baterai tidak berfungsi sebagaimana mestinya. Proses ini dicatat pada lembar AKB (Analisis Kondisi Baterai) pada

Tabel IV.5 Definisi *Use Case* Proses Klaim

No	<i>Use Case</i>	Definisi
		bagian IV (Keempat).
15	Validasi <i>form</i> AKB	Menyetujui data produk baterai yang bermasalah pada lembar AKB oleh kepala bagian <i>claim handling</i> yang selanjutnya akan diserahkan kepada <i>QA Section Head</i> .
16	Membuat laporan departemen QA	Laporan akar permasalahan sementara dibuat oleh <i>QA Section Head</i> yang dibantu oleh admin pada bagian tersebut.
17	View laporan kerusakan	Departemen QA melakukan <i>view</i> laporan untuk melakukan persentasi di rapat klaim.

Sumber: Pengumpulan dan Pengolahan Data (2015)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan perancangan aplikasi sistem informasi pemeriksaan produk klaim baterai aki sebagai sarana untuk mengolah data pemeriksaan produk di bagian *Claim Handling* dan memberikan informasi mengenai pelaporan kerusakan baterai aki pada bagian *QA Section Head*. Berikut adalah daftar kebutuhan sistem untuk aplikasi sistem informasi pemeriksaan produk klaim baterai aki pada bagian *Claim Handling*.

Tabel V.1 Kebutuhan Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki

Kebutuhan Sistem	
<i>Project Name</i>	Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki.
<i>Project Sponsor</i>	<i>Claim Handling Sub Section</i>
<i>Business Need</i>	<ol style="list-style-type: none"> 1. Terwujudnya sebuah sistem yang dapat mengelola data kerusakan produk klaim pada bagian <i>Claim Handling</i> agar pengolahan data kerusakan bisa lebih baik dari sebelumnya. 2. Terwujudnya suatu aplikasi berbasis data yang dapat memberikan laporan kerusakan baterai setiap periodenya menjadi lebih cepat, mudah dan akurat.
<i>Business Requirement</i>	Memberikan sistem informasi pemeriksaan produk klaim baterai aki yang dapat membantu bagian <i>Claim Handling</i> dalam mengolah data kerusakan dan membuat laporan.
<i>Bussiness Value</i>	<ol style="list-style-type: none"> 1. Memudahkan dalam pembuatan laporan informasi kerusakan baterai sehingga laporan menjadi lebih informatif dan mudah dimengerti. 2. Mampu menyimpan laporan kerusakan baterai dengan baik di basis data.
<i>Special Issues or Constrains</i>	-

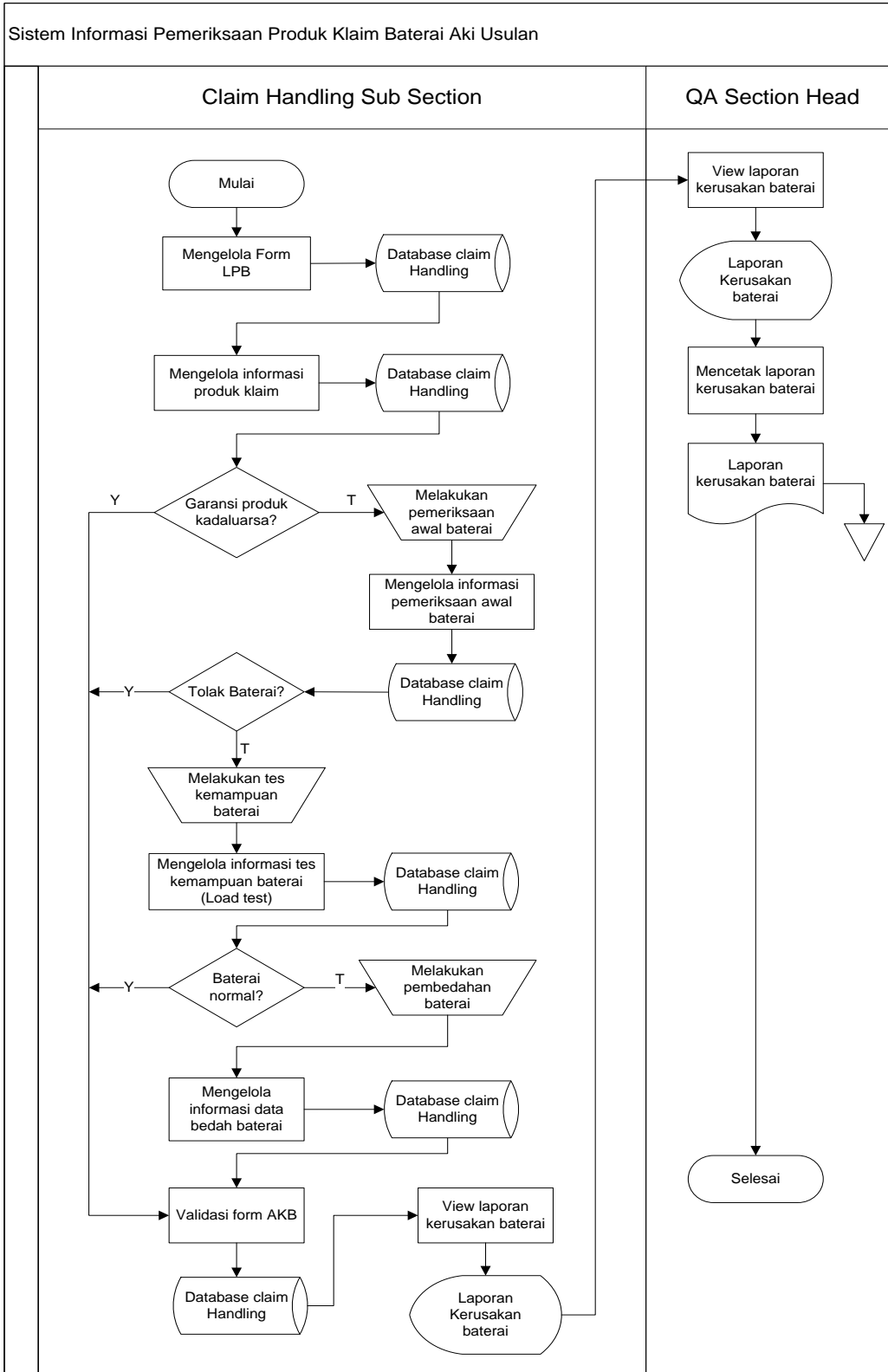
Sumber: Hasil Analisis Data (2016)

5.2 Prosedur Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan

Prosedur sistem informasi pemeriksaan produk klaim baterai aki yang diusulkan melalui tahapan sebagai berikut:

1. *Claim Handling Sub Section* menerima produk baterai aki yang dikirimkan oleh *customer*.
2. *Claim Handling Sub Section* melakukan pengisian data LPB (Laporan Penerimaan Barang).
3. *Claim Handling Sub Section* melakukan pemeriksaan awal untuk menentukan apakah klaim produk baterai tersebut bisa dilanjutkan.
4. *Claim Handling Sub Section* melakukan *charging* produk baterai sesuai dengan kondisi baterai.
5. *Claim Handling Sub Section* melakukan pemeriksaan setelah *charge* dan melakukan *load test*.
6. Apabila ditemukan gelembung pada *cell* baterai maka akan dilakukan pembedahan oleh bagian *Claim Handling Sub Section*. Apabila tidak ditemukan gelembung pada *cell*, dapat dipastikan indikator *load tester* berwarna hijau dan produk baterai akan di-*charge* kembali sebelum dikembalikan ke *customer*.
7. *Claim Handling Sub Section* akan melakukan validasi produk baterai aki yang ditolak, lolos uji *load test* atau produk yang telah dibedah untuk mencari sumber kerusakan.
8. *Claim Handling Sub Section* dan *QA Section Head* dapat melihat laporan kerusakan baterai. Laporan tersebut nantinya bisa digunakan oleh bagian *QA Section Head* untuk melakukan rapat klaim dengan bagian terkait lainnya untuk mencari akar permasalahan sebenarnya.

Seperti yang dapat dilihat pada Gambar V.1 adalah *Flowmap* Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan sebagai berikut:



Gambar

V.1 *Flowmap* Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan
 Sumber: Hasil Analisis Data (2016)

5.3 Analisis dan Perancangan Sistem Usulan

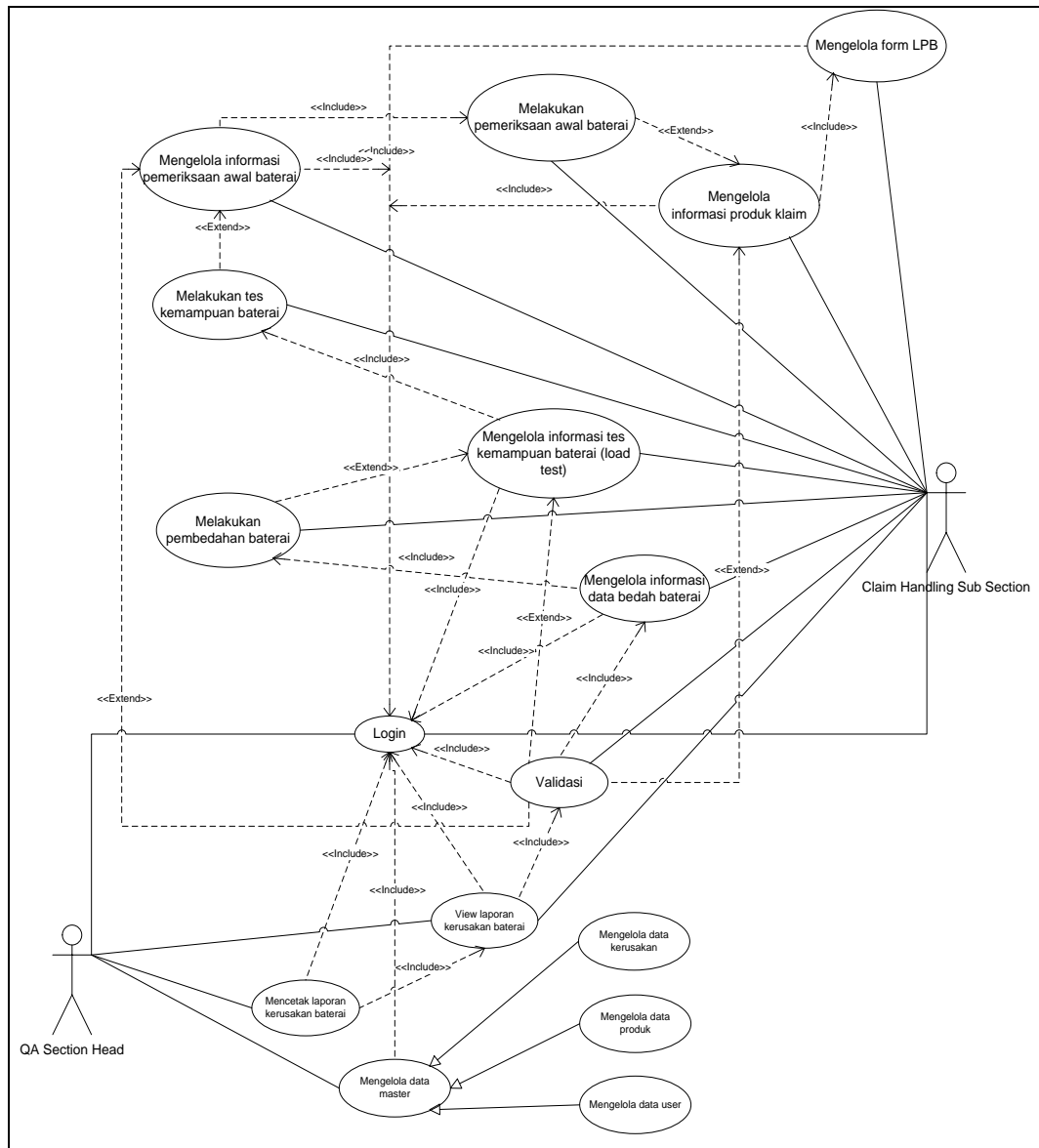
Untuk mengatasi permasalahan dalam proses pemeriksaan produk klaim baterai aki yang ada pada bagian *Claim Handling* diajukan usulan sistem baru dengan menerapkan penggunaan aplikasi komputer untuk pengolahan data yang akan mendukung beberapa proses bisnis yang sedang berjalan pada perusahaan. Dengan menggunakan aplikasi ini membuat sistem menjadi terintegrasi sehingga dapat menyajikan informasi secara cepat dan dapat meningkatkan fungsionalitas proses klaim baterai aki yang ada pada bagian *Claim Handling* di PT Nipress Tbk.

Sistem yang diusulkan yaitu mempercepat proses pembuatan laporan kerusakan baterai dan membangun basis data laporan kerusakan baterai yang kemudian digunakan untuk bahan evaluasi dan *meeting* oleh bagian *QA Section Head*. Analisis dan perancangan sistem ini menggunakan model pengembangan perangkat lunak prototipe evolusioner. Model prototipe cocok digunakan untuk menjabarkan kebutuhan *user* secara lebih terperinci karena *user* sering kali kesulitan menyampaikan kebutuhannya secara terperinci tanpa melihat gambaran yang jelas (Rosa dan Shalahuddin, 2013). Kelebihan model prototipe evolusioner adalah program prototipe yang telah dievaluasi oleh *user* dapat digunakan untuk proses selanjutnya tanpa harus dibuang, sehingga dapat mempercepat pembuatan program. Pemodelan sistem dengan UML melewati beberapa tahapan yaitu:

1. *Use Case Diagram*
2. *Activity Diagram*
3. *Sequence Diagram*
4. *Deployment Diagram*
5. *Class Diagram*

5.3.1 Use Case Diagram

Use case diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Rancangan *use case diagram* sistem informasi pemeriksaan produk klaim baterai aki yang diusulkan dapat dilihat pada Gambar V.2.



Gambar V.2 Use Case Diagram Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan
 Sumber: Hasil Analisis Data (2016)

Perancangan *use case diagram* sistem informasi klaim baterai aki dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada *use case* sistem informasi pemeriksaan produk klaim baterai aki usulan dapat dilihat pada Tabel V.2 berikut:

Tabel V.2 Definisi Aktor *Use Case Diagram* Sistem Usulan

No	Aktor	Deskripsi
1	<i>Claim Handling Sub Section</i>	<i>Claim Handling Sub Section</i> adalah bagian yang terdiri dari <i>Claim Handling Section Head</i> dan <i>inspector</i> . Bagian tersebut bertanggung jawab terhadap penerimaan baterai aki dari <i>customer</i> dan mengelola data pemeriksaan baterai klaim.
2	<i>QA Section Head</i>	<i>QA Section Head</i> adalah yang menerima hasil laporan kerusakan produk baterai aki, mengelola data master, memberikan saran dan kebijakan atas laporan pengendalian produk serta melakukan <i>meeting</i> dengan bagian terkait lainnya untuk mencari akar permasalahan sebenarnya.

Sumber: Hasil Analisis Data (2016)

2. Definisi Use Case

Pendefinisian *use case* pada sistem informasi pemeriksaan produk klaim baterai aki usulan dapat dilihat pada Tabel V.3:

Tabel V.3 Definisi *Use Case Diagram* Sistem Usulan

No	Use Case	Deskripsi
1	<i>Login</i>	Proses untuk melakukan <i>login</i> pada aplikasi
2	Mengelola data master	Proses generalisasi yang meliputi tiga buah proses pengolahan data master, yaitu data <i>user</i> , produk dan kerusakan baterai aki.
3	Mengelola data <i>user</i>	Proses mengelola data <i>user</i> yang terdiri dari tambah, ubah, hapus dan cari pada data <i>user</i> .
4	Mengelola data produk	Proses mengelola data produk untuk melakukan tambah, ubah, hapus, dan cari pada data produk.
5	Mengelola data kerusakan	Proses mengelola data kerusakan untuk melakukan tambah, ubah, hapus, dan cari.
6	Mengelola <i>form</i> LPB	Bagian <i>claim handling</i> akan meng- <i>input</i> data LPB (Laporan Penerimaan Barang) untuk mendata mengenai asal produk tersebut. Setelah itu bagian <i>claim handling</i> akan meng- <i>input</i> data detail LPB (Laporan Penerimaan Barang) untuk mendata mengenai produk baterai yang diterima.

Tabel V.3 Definisi *Use Case Diagram* Sistem Usulan

No	Use Case	Deskripsi
7	Mengelola informasi produk klaim	Bagian <i>claim</i> akan melakukan pemeriksaan informasi produk masing-masing baterai.
8	Melakukan pemeriksaan awal baterai	Bagian <i>claim</i> akan melakukan pemeriksaan fisik pada setiap produk, termasuk pemeriksaan SpGr dan tegangan baterai untuk mengukur lama <i>charge</i> . (Lanjutan)
9	Mengelola informasi pemeriksaan awal baterai	Selanjutnya, bagian <i>claim</i> akan menginput data hasil pemeriksaan tersebut.
10	Melakukan tes kemampuan baterai	Aki kembali diperiksa untuk memastikan kelistrikan baterai masih dalam kondisi baik dan memastikan produk baterai menunjukkan indikator hijau pada alat <i>load tester</i> . Hal itu menunjukkan bahwa baterai masih dalam keadaan baik.
11	Mengelola informasi tes kemampuan baterai (<i>Load Test</i>)	Apabila indikator menunjukkan warna merah, maka dapat dipastikan terdapat gelembung udara dan baterai harus dibedah. Selanjutnya petugas akan menginput data hasil pemeriksaan tersebut.
12	Melakukan pembedahan baterai	Proses pembedahan dilakukan untuk mengetahui permasalahan pada baterai. Memastikan juga pada <i>part</i> mana yang membuat baterai tidak berfungsi sebagaimana mestinya.
13	Mengelola informasi data bedah baterai	Selanjutnya petugas akan menginput data hasil pemeriksaan bedah baterai.
14	Validasi	Proses validasi produk baterai aki yang ditolak, lolos uji <i>load test</i> atau produk yang telah dibedah untuk mencari sumber kerusakan oleh <i>Claim Handling Sub Section Head</i> .
15	View laporan kerusakan baterai	Laporan dapat dilihat oleh <i>Claim handling Sub Section</i> dan <i>QA Section Head</i> .
16	Mencetak laporan kerusakan baterai	Laporan dapat dicetak untuk selanjutnya digunakan oleh bagian <i>QA Section Head</i> saat melakukan <i>meeting</i> dengan bagian terkait lainnya untuk mencari akar permasalahan sebenarnya.

Sumber: Hasil Analisis Data (2016)

3. Skenario Use Case

Skenario jalannya masing-masing *use case* pada sistem informasi pemeriksaan produk klaim baterai aki usulan dapat dilihat pada poin berikut:

1. *Use Case Login*

Berikut adalah skenario *use case login* yang terdapat pada Tabel V.4:

Tabel V.4 Skenario *Use Case Login*

Nama Use Case	Login
<i>Primary Actor</i>	<i>QA Section Head, Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan login sesuai dengan hak akses dari user. Dalam hal ini hak aksesnya sesuai jabatan.</i>
<i>Relationship</i>	<i>Association: QA Section Head, Claim Handling Sub Section</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"><i>1. User membuka Login Form.</i><i>2. User memasukkan user name dan password pada login form.</i><i>3. Sistem mengecek ke basis data, apakah user name dan password benar.</i><i>4. Jika user name dan password benar, maka muncul tampilan menu utama.</i><i>5. Jika user name dan password salah (tidak valid), maka muncul pesan “User Name dan Password Salah” pada login form.</i>

Sumber: Hasil Analisis Data (2016)

2. *Use Case Mengelola Data User*

Berikut skenario *use case* mengelola data *user* yang terdapat pada Tabel V.5:

Tabel V.5 Skenario *Use Case Mengelola Data User*

Nama Use Case	Mengelola Data User
<i>Primary Actor</i>	<i>QA Section Head</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses pengolahan data user.</i>
<i>Relationship</i>	<i>Association: QA Section Head</i> <i>Generalization: Mengelola data master</i> <i>Include: - Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Login</i> dan masuk ke tampilan menu utama. 2. <i>User</i> memilih menu file, kemudian memilih sub menu data <i>user</i>. 3. Sistem menampilkan data <i>user</i>. 4. <i>User</i> melakukan proses tambah, ubah dan hapus data <i>user</i> di basis data.

Sumber: Hasil Analisis Data (2016)

3. *Use Case* Mengelola Data Produk

Berikut adalah skenario *use case* mengelola data produk yang terdapat pada Tabel V.6:

Tabel V.6 Skenario *Use Case* Mengelola Data Produk

Nama Use Case	Mengelola Data Produk
<i>Primary Actor</i>	<i>QA Section Head</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses pengolahan data produk baterai aki.</i>
<i>Relationship</i>	<i>Association: QA Section Head</i> <i>Generalization: Mengelola data master</i> <i>Include: - Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Login</i> dan masuk ke tampilan menu utama. 2. <i>User</i> memilih menu file, kemudian memilih sub menu data produk. 3. Sistem menampilkan data produk. 4. <i>User</i> melakukan proses tambah, ubah dan hapus produk baterai di basis data.

Sumber: Hasil Analisis Data (2016)

4. *Use Case* Mengelola Data Kerusakan

Berikut adalah skenario *use case* mengelola data kerusakan yang terdapat pada Tabel V.7:

Tabel V.7 Skenario *Use Case* Mengelola Data Kerusakan

Nama Use Case	Mengelola Data Kerusakan
<i>Primary Actor</i>	<i>QA Section Head</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses pengolahan jenis kerusakan produk baterai aki.</i>
<i>Relationship</i>	<i>Association: QA Section Head Generalization: Mengelola data master Include: - Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Login</i> dan masuk ke tampilan menu utama. 2. <i>User</i> memilih menu file, kemudian memilih sub menu data kerusakan. 3. Sistem menampilkan data kerusakan. 5. <i>User</i> melakukan proses tambah, ubah dan hapus jenis kerusakan baterai aki di basis data.

Sumber: Hasil Analisis Data (2016)

5. *Use Case* Mengelola *Form* LPB

Berikut adalah skenario *use case* mengelola *form* LPB yang terdapat pada Tabel V.8:

Tabel V.8 Skenario *Use Case* Mengelola *Form* LPB

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses pada bagian claim handling akan meng-input data LPB (Laporan Penerimaan Barang) untuk mendata mengenai asal produk tersebut. Setelah itu meng-input data detil LPB untuk mendata mengenai produk baterai yang diterima</i>
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i>

	<i>Include: - Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Login</i> dan masuk ke tampilan menu utama. 2. <i>User</i> memilih menu daftar klaim, kemudian memilih sub menu laporan penerimaan barang. 3. <i>User</i> melengkapi data asal barang, lalu menambahkan data ke dalam basis data. 4. <i>User</i> memilih menu daftar klaim, kemudian memilih sub menu detil LPB. 5. <i>User</i> melengkapi data barang, lalu menambahkan data ke dalam basis data. 6. <i>User</i> juga dapat melakukan cari, ubah dan hapus pada <i>form</i> LPB.

Sumber: Hasil Analisis Data (2016)

6. *Use Case* Mengelola Informasi Produk Klaim

Berikut adalah skenario *use case* mengelola *form* detil LPB yang terdapat pada Tabel V.9:

Tabel V.9 Skenario *Use Case* Mengelola Informasi Produk Klaim

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses pada bagian <i>claim handling</i> akan melakukan pemeriksaan informasi produk masing-masing baterai.
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Include: - Login (Lanjutan)</i> <i>- Mengelola form LPB</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Login</i> dan masuk ke tampilan menu utama. 2. <i>User</i> memilih menu pemeriksaan produk, kemudian memilih sub menu pemeriksaan awal. 3. <i>User</i> mengisi kode produksi baterai. 4. <i>User</i> mengisi no LPB dan id produk baterai. 5. <i>User</i> melengkapi data produk lainnya, lalu menambahkan data ke dalam basis data. 6. <i>User</i> juga dapat melakukan cari, ubah dan hapus pada <i>form</i> LPB.

Sumber: Hasil Analisis Data (2016)

7. *Use Case* Melakukan Pemeriksaan Awal Baterai

Berikut adalah skenario *use case* melakukan pemeriksaan awal baterai yang terdapat pada Tabel V.10:

Tabel V.10 Skenario *Use Case* Melakukan Pemeriksaan Awal Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses manual pemeriksaan fisik pada setiap produk, termasuk pemeriksaan SpGr dan tegangan baterai untuk mengukur lama charge.</i>
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Extend: - Mengelola Informasi Produk Klaim</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none">1. <i>Inspector</i> mengambil baterai dari gudang klaim untuk dilakukan pemeriksaan.2. <i>Inspector</i> memeriksa tampilan produk baterai aki lalu mengisi hasil analisis ke dalam aplikasi.3. <i>Inspector</i> memeriksa tegangan dan masa jenis elektrolit baterai lalu mengisi hasil analisis ke dalam aplikasi.

Sumber: Hasil Analisis Data (2016)

8. *Use Case* Mengelola Informasi Pemeriksaan Awal Baterai

Berikut adalah skenario *use case* mengelola informasi pemeriksaan awal baterai yang terdapat pada Tabel V.11:

Tabel V.11 Skenario *Use Case* Mengelola Informasi Pemeriksaan Awal Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses melakukan input data pemeriksaan awal baterai aki pada aplikasi.</i>

Tabel V.11 Skenario *Use Case* Mengelola Informasi Pemeriksaan Awal Baterai

Nama <i>Use Case</i>	Terima Produk Baterai
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Include: - Login</i> - Melakukan pemeriksaan awal baterai
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> melakukan <i>login</i> untuk masuk ke tampilan menu utama. 2. <i>User</i> memilih menu pemeriksaan produk, lalu memilih sub menu pemeriksaan awal. 3. <i>User</i> memilih tab cek tampilan luar. 4. <i>User</i> mengisi data pemeriksaan produk dan mengisi pada tabel cell untuk melakukan penghitungan masa <i>charge</i> baterai, lalu data disimpan di basis data. 5. <i>User</i> dapat melakukan proses tambah, ubah dan cari di dalam sub menu pemeriksaan awal.

Sumber: Hasil Analisis Data (2016)

9. *Use Case* Melakukan Tes Kemampuan Baterai

Berikut adalah skenario *use case* melakukan tes kemampuan baterai (*load test*) yang terdapat pada Tabel V.12:

Tabel V.12 Skenario *Use Case* Melakukan Tes Kemampuan Baterai

Nama <i>Use Case</i>	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan kegiatan pemeriksaan fisik produk baterai aki pasca <i>charge</i> .
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Extend: - Mengelola informasi pemeriksaan awal baterai</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Inspector</i> mengambil baterai yang telah melewati proses <i>charging</i> awal. 2. <i>Inspector</i> mempersiapkan alat <i>BT Load Tester</i> untuk memeriksa kondisi tegangan masing-masing <i>cell</i> baterai. 3. <i>Inspector</i> melakukan tes fisik pasca <i>charge</i> dan melakukan tes kemampuan baterai (<i>load test</i>) lalu

Tabel V.12 Skenario *Use Case* Melakukan Tes Kemampuan Baterai

Nama Use Case	Terima Produk Baterai
	melakukan <i>input</i> data di aplikasi.

Sumber: Hasil Analisis Data (2016)

10. *Use Case* Mengelola Informasi Tes Kemampuan Baterai

Berikut adalah skenario *use case* mengelola informasi tes kemampuan baterai (*load test*) yang terdapat pada Tabel V.13:

Tabel V.13 Skenario *Use Case* Mengelola Informasi Tes Kemampuan Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan kegiatan input data tes kemampuan (<i>load test</i>) pada aplikasi.
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Include: - Login</i> - Melakukan tes kemampuan baterai
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> melakukan <i>login</i> untuk masuk ke tampilan menu utama. 2. <i>User</i> memilih menu pemeriksaan produk dan memilih sub menu pasca <i>charge</i> dan <i>load test</i>. 3. Pemeriksaan dilakukan petugas untuk memastikan daya baterai dan masa jenis elektrolit serta kemampuan baterai per unit. 4. <i>User</i> melakukan pengolahan data dan menyimpannya ke dalam basis data.

Sumber: Hasil Analisis Data (2016)

11. *Use Case* Melakukan Pembedahan Baterai

Berikut adalah skenario *use case* melakukan pembedahan baterai yang terdapat pada Tabel V.14:

Tabel V.14 Skenario *Use Case* Melakukan Pembedahan Baterai

Nama Use Case	Terima Produk Baterai
----------------------	------------------------------

Tabel V.14 Skenario *Use Case* Melakukan Pembedahan Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan kegiatan pemeriksaan fisik dengan cara bedah produk baterai.</i>
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Extend: - Mengelola informasi tes kemampuan baterai (load test)</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Inspector</i> mengambil baterai yang akan melalui proses bedah. 2. Pembedahan dilakukan untuk meneliti permasalahan yang diduga bersumber dari bagian dalam baterai, lalu <i>input</i> data ke dalam aplikasi.

Sumber: Hasil Analisis Data (2016)

12. *Use Case* Mengelola Informasi Data Bedah Baterai

Berikut adalah skenario *use case* mengelola informasi data bedah baterai yang terdapat pada Tabel V.15:

Tabel V.15 Skenario *Use Case* Mengelola Informasi Data Bedah Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Sub Section</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan input data bedah baterai pada aplikasi.</i>
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Include: - Login</i> <i>- Melakukan pembedahan baterai</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> melakukan <i>login</i> untuk masuk ke tampilan menu utama. 2. <i>User</i> memilih menu pemeriksaan produk. 3. <i>User</i> memilih sub menu bedah baterai. 4. <i>User</i> meng-<i>input</i> data kerusakan baterai yang ditemui. 5. <i>User</i> melakukan pengolahan data dan menyimpannya ke dalam basis data.

Sumber: Hasil Analisis Data (2016)

13. *Use Case Validasi*

Berikut adalah skenario *use case* validasi yang terdapat pada Tabel V.16:

Tabel V.16 Skenario *Use Case* Validasi

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>Claim Handling Section Head</i>
<i>Use Case Description</i>	<i>Use case ini menggambarkan proses validasi produk baterai aki yang telah dilakukan pemeriksaan untuk menemukan kerusakan produk.</i>
<i>Relationship</i>	<i>Association: Claim Handling Sub Section</i> <i>Include: - Login</i> - Mengelola informasi data bedah baterai <i>Extend: - Mengelola informasi tes kemampuan baterai</i> - Mengelola informasi pemeriksaan awal baterai - Mengelola informasi produk klaim
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> melakukan <i>login</i> untuk masuk ke tampilan menu utama. (Lanjutan) 2. <i>User</i> memilih menu file, lalu memilih sub menu validasi produk. 3. <i>User</i> melakukan validasi terhadap produk yang telah dilakukan klaim maupun produk yang ditolak untuk dilakukan kalim. 4. Validasi disimpan ke dalam basis data.

Sumber: Hasil Analisis Data (2016)

14. *Use Case View Laporan Kerusakan Baterai*

Berikut adalah skenario *use case* view laporan kerusakan baterai yang terdapat pada Tabel V.17:

Tabel V.17 Skenario *Use Case* View Laporan Kerusakan Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>QA Section Head, Claim Handling Section Head</i>
<i>Use Case</i>	<i>Use case ini menggambarkan kegiatan view laporan</i>

Tabel V.17 Skenario *Use Case* View Laporan Kerusakan Baterai

Nama Use Case	Terima Produk Baterai
<i>Description</i>	kerusakan baterai.
<i>Relationship</i>	<i>Association: Claim Handling Sub Section & QA Section Head</i> <i>Include: - Login</i> <i>- Validasi form AKB</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> melakukan <i>login</i> untuk masuk ke tampilan menu utama. 2. <i>User</i> memilih menu laporan. 3. Laporan kerusakan dan penerimaan dapat dilihat berdasarkan waktu kebutuhan. 4. <i>User</i> melakukan proses <i>view</i> laporan.

Sumber: Hasil Analisis Data (2016)

15. *Use Case* Mencetak Laporan Kerusakan Baterai

Berikut adalah skenario *use case* mencetak laporan kerusakan baterai yang terdapat pada Tabel V.18:

Tabel V.18 Skenario *Use Case* Mencetak Laporan Kerusakan Baterai

Nama Use Case	Terima Produk Baterai
<i>Primary Actor</i>	<i>QA Section Head</i>
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan kegiatan mencetak laporan kerusakan baterai.
<i>Relationship</i>	<i>Association: QA Section Head</i> <i>Include: - Login</i> <i>- View laporan kerusakan baterai</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User</i> melakukan <i>login</i> untuk masuk ke tampilan menu utama.

Tabel V.18 Skenario *Use Case* Mencetak Laporan Kerusakan Baterai

Nama <i>Use Case</i>	Terima Produk Baterai
	2. <i>User</i> memilih menu laporan. 3. <i>User</i> melihat laporan kerusakan atau penerimaan baterai. 4. <i>User</i> melakukan proses cetak laporan tersebut.

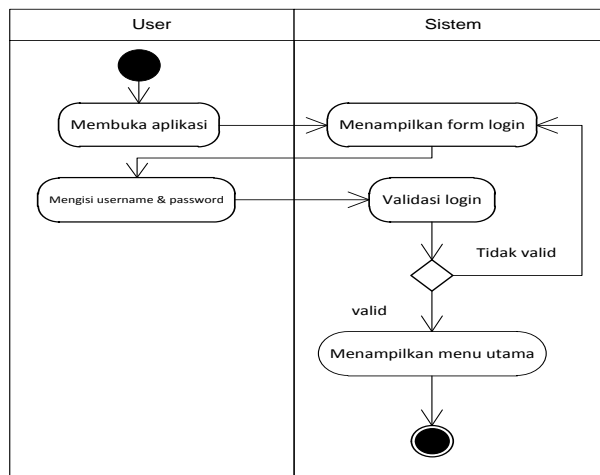
Sumber: Hasil Analisis Data (2016)

5.3.2 *Activity Diagram*

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. Agar lebih memahami sistem yang akan dibuat, maka perlu dibuatkan *activity diagram* tentang sistem informasi usulan ini.

1. *Activity Diagram Login*

Activity diagram login ini menjelaskan alur aktivitas yang dilakukan oleh *user*, yaitu *QA Section Head* dan *Claim Handling Sub Section* untuk dapat masuk ke dalam sistem. *Activity diagram login* dapat dilihat pada Gambar V.3 sebagai berikut:

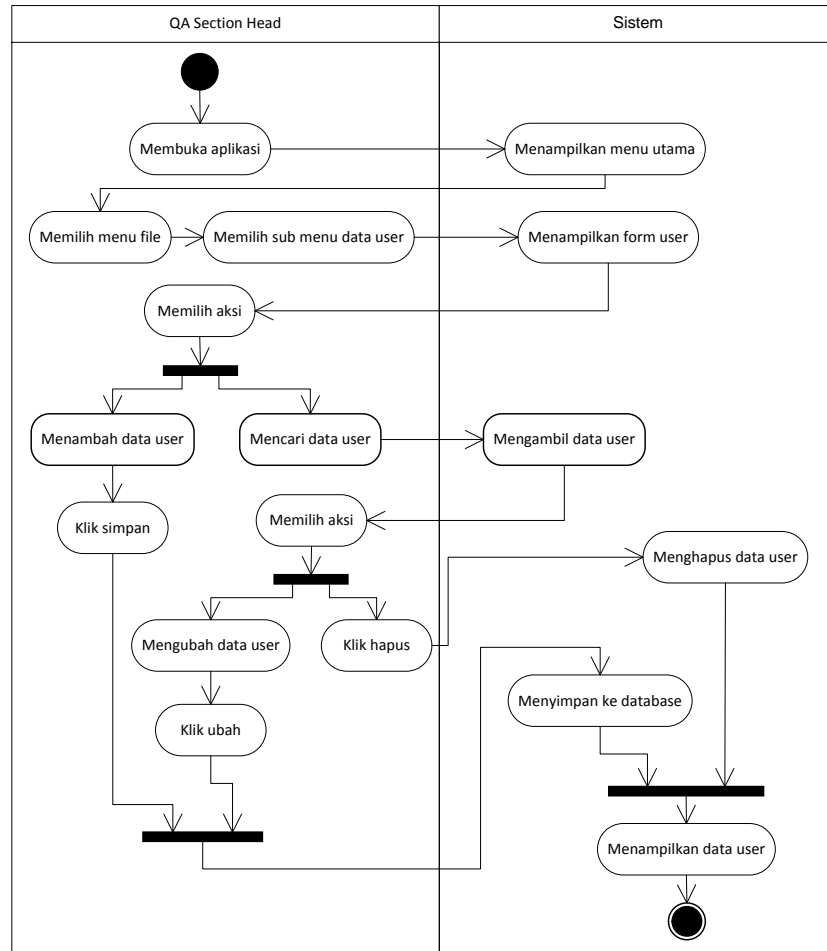


Gambar V.3 *Activity Diagram Login*

Sumber: Hasil Analisis Data (2016)

2. *Activity Diagram Mengelola Data User*

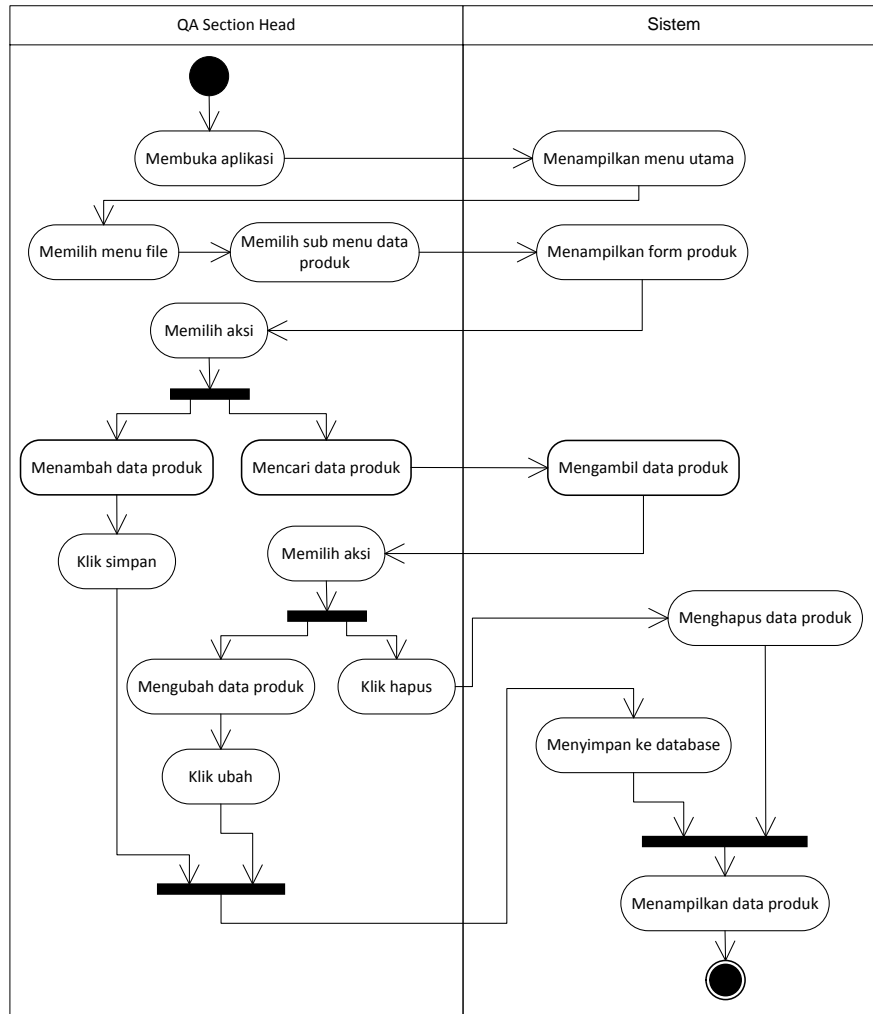
Activity diagram mengelola data *user* berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity diagram* mengelola data *user* yang diusulkan dapat dilihat pada Gambar V.4 sebagai berikut:



Gambar V.4 Activity Diagram Mengelola Data User
Sumber: Hasil Analisis Data (2016)

3. Activity Diagram Mengelola Data Produk

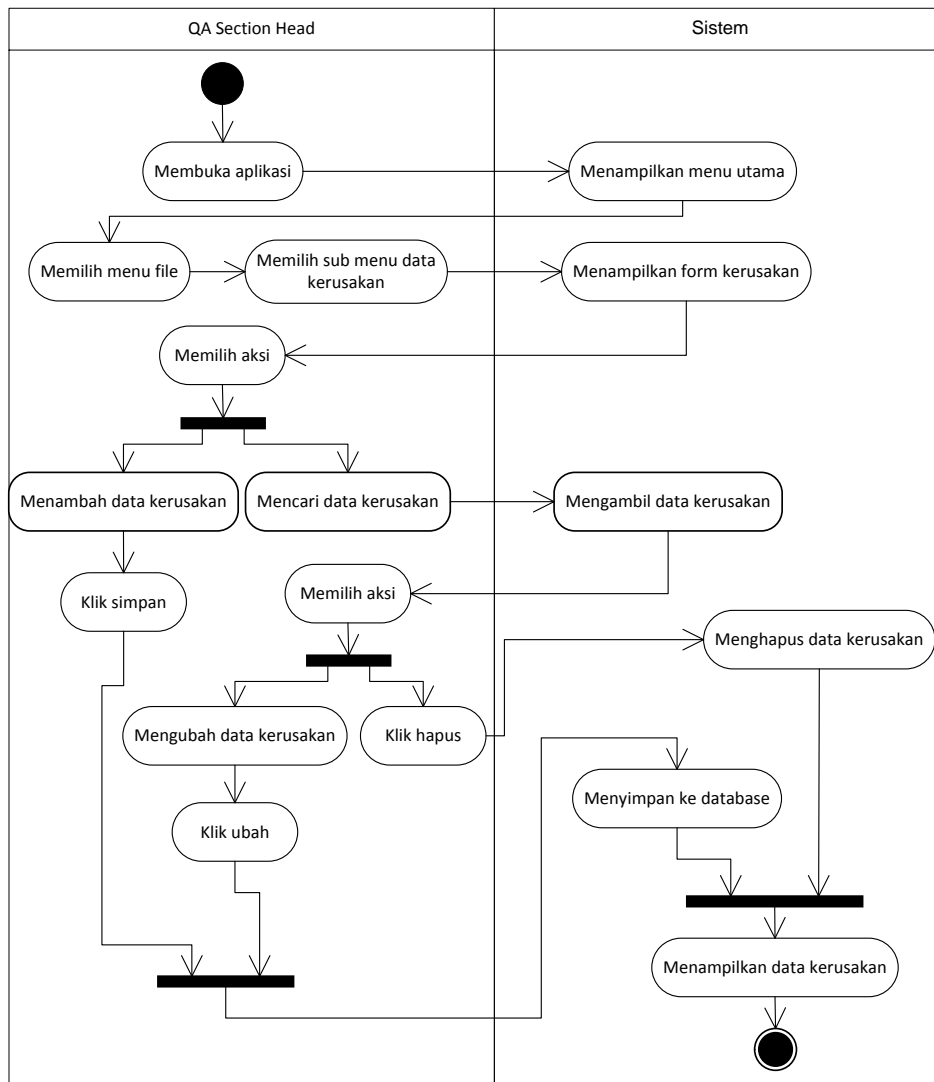
Activity diagram mengelola data produk berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. Activity diagram mengelola data produk yang diusulkan dapat dilihat pada Gambar V.5 sebagai berikut:



Gambar V.5 Activity Diagram Mengelola Data Produk
 Sumber: Hasil Analisis Data (2016)

4. Activity Diagram Mengelola Data Kerusakan

Activity diagram mengelola data kerusakan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. Activity diagram mengelola data tipe kerusakan baterai aki yang diusulkan dapat dilihat pada Gambar V.6 sebagai berikut:

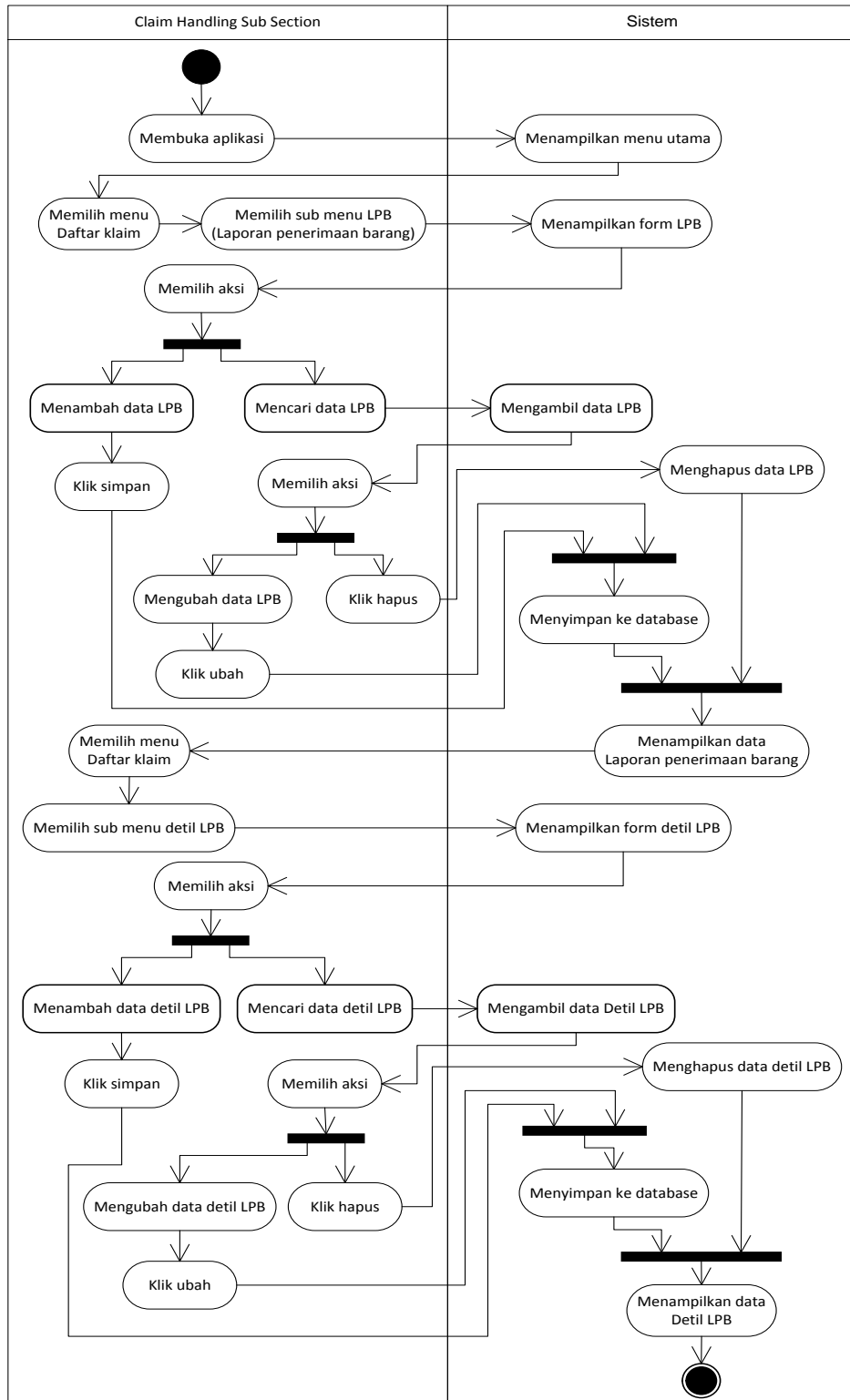


Gambar V.6

Activity Diagram Mengelola Data Kerusakan Produk
 Sumber: Hasil Analisis Data (2016)

5. *Activity Diagram Mengelola Form LPB*

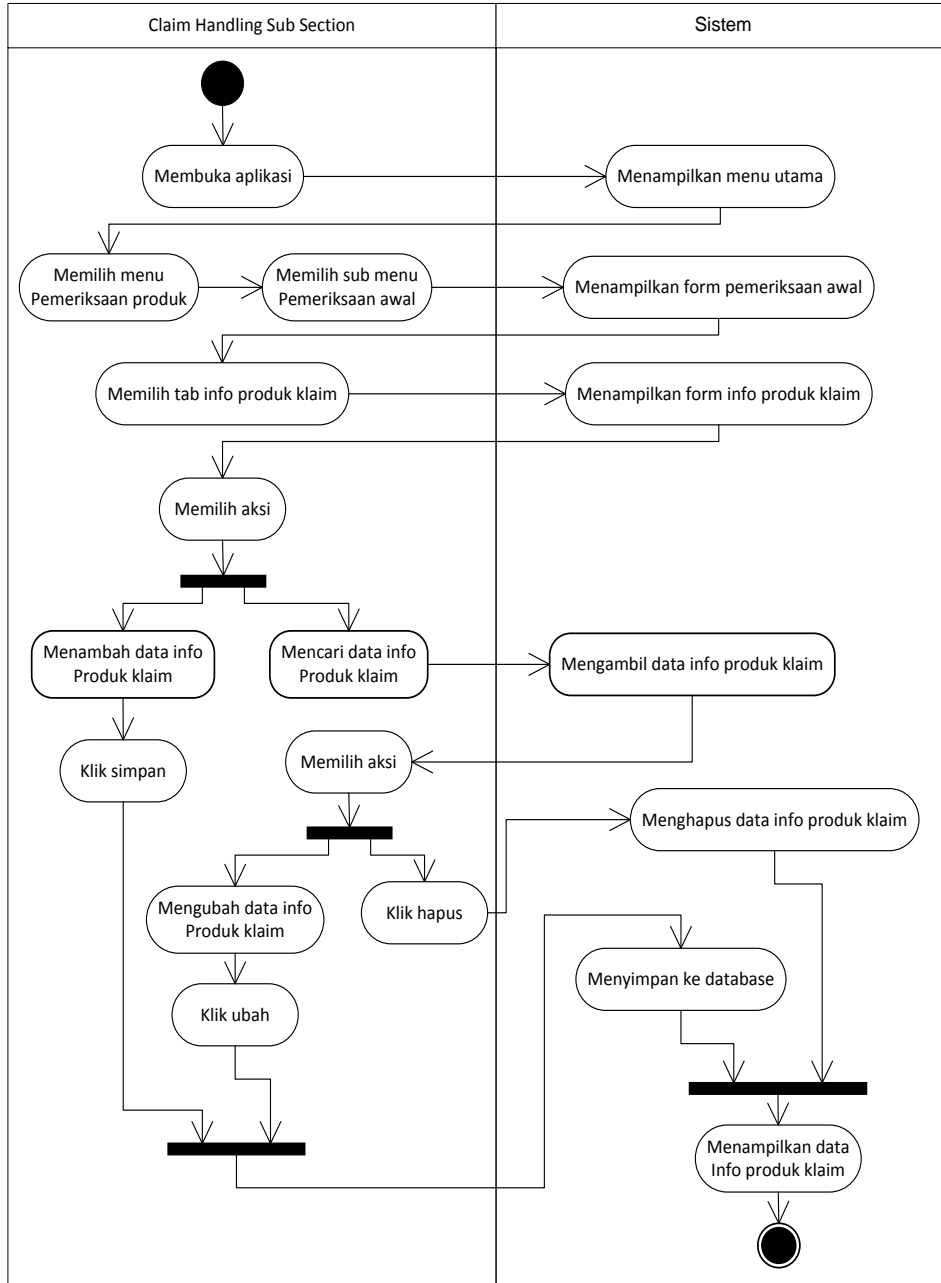
Activity diagram mengelola form LPB berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity diagram* mengelola form LPB yang diusulkan dapat dilihat pada Gambar V.7 sebagai berikut:



Gambar V.7 Activity Diagram Mengelola Form LPB
 Sumber: Hasil Analisis Data (2016)

6. *Activity Diagram* Mengelola Informasi Produk Klaim

Activity diagram mengelola informasi produk klaim berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity diagram* mengelola informasi produk klaim yang diusulkan dapat dilihat pada Gambar V.8 sebagai berikut:

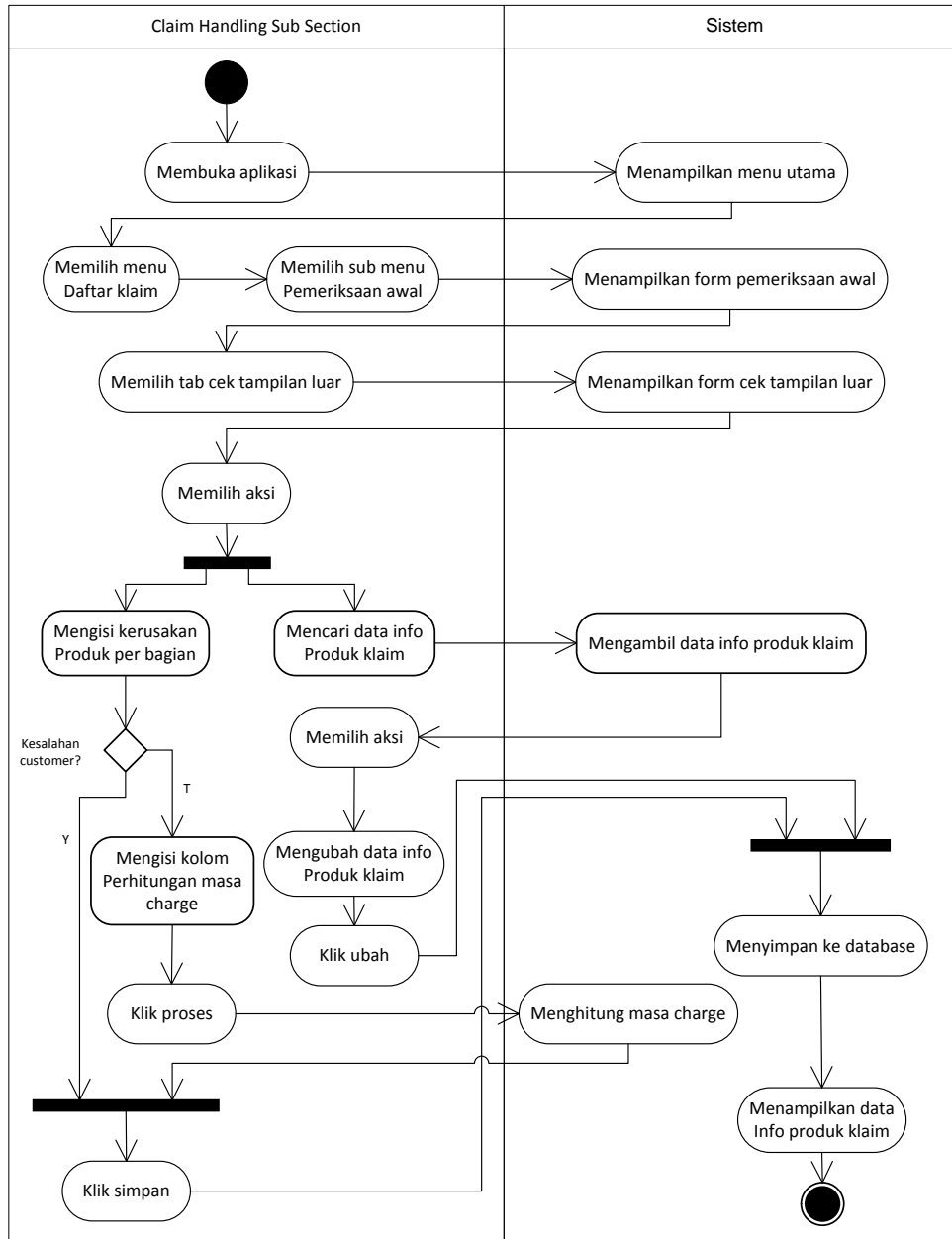


Gambar

V.8 *Activity Diagram* Mengelola Informasi Produk Klaim
 Sumber: Hasil Analisis Data (2016)

7. *Activity Diagram* Mengelola Informasi Pemeriksaan Awal Baterai

Activity diagram mengelola informasi pemeriksaan awal baterai berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity diagram* mengelola informasi pemeriksaan awal baterai yang diusulkan dapat dilihat pada Gambar V.9 sebagai berikut:

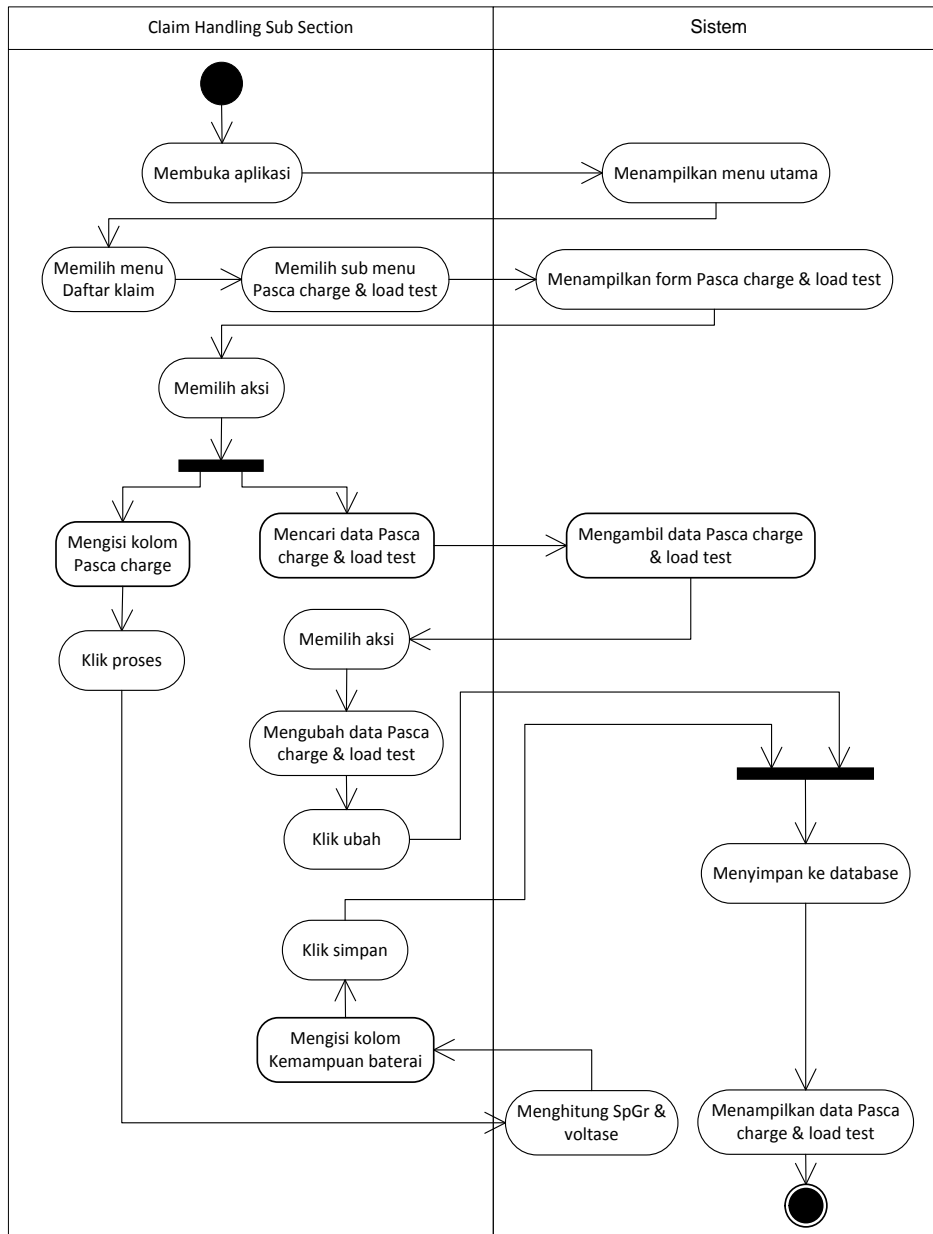


Gambar

V.9 *Activity Diagram* Mengelola Informasi Pemeriksaan Awal Baterai
Sumber: Hasil Analisis Data (2016)

8. *Activity Diagram* Mengelola Informasi Tes Kemampuan Baterai (*Load Test*)

Activity diagram mengelola informasi tes kemampuan baterai berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity diagram* yang diusulkan dapat dilihat pada Gambar V.10 sebagai berikut:



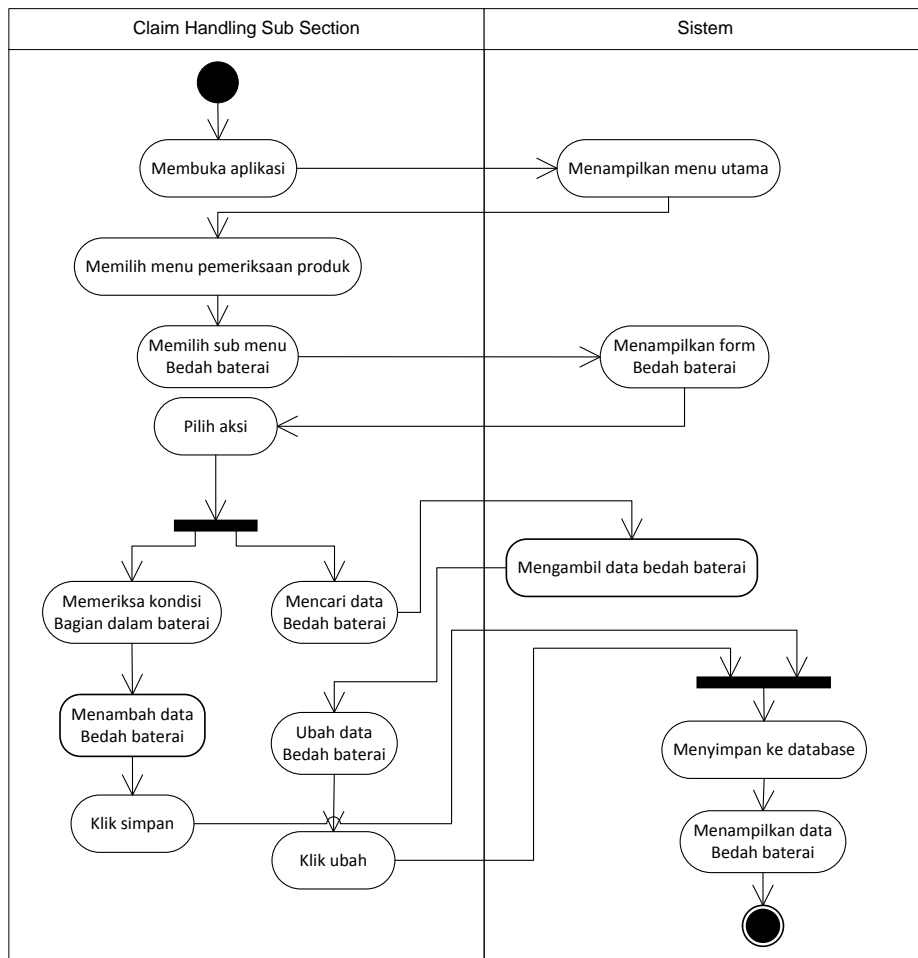
Gambar V.10

Activity Diagram Mengelola Informasi Tes Kemampuan Baterai
Sumber: Hasil Analisis Data (2016)

9. *Activity Diagram* Mengelola Informasi Data Bedah Baterai

Activity diagram mengelola informasi data bedah baterai berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity*

diagram mengelola informasi data bedah baterai yang diusulkan dapat dilihat pada Gambar V.11 sebagai berikut:

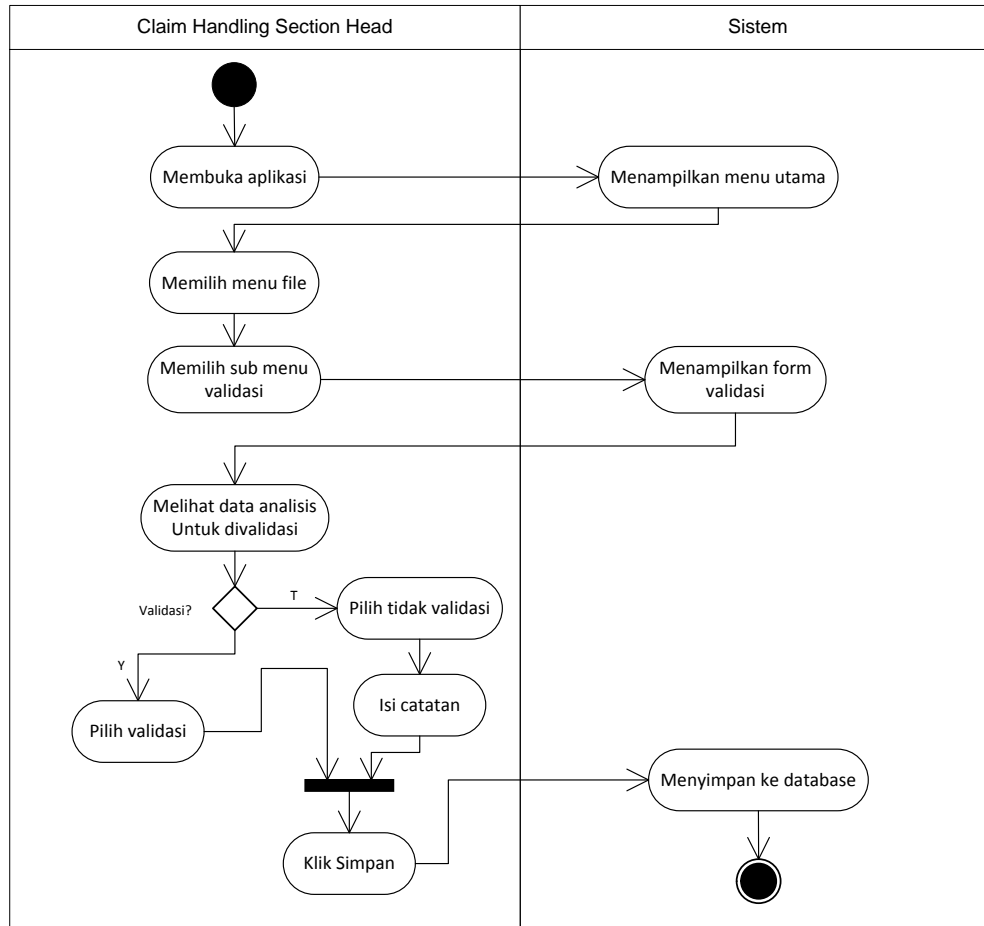


Gambar V.11

Activity Diagram Mengelola Informasi Data Bedah Baterai
Sumber: Hasil Analisis Data (2016)

10. Activity Diagram Validasi

Activity diagram validasi berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. Activity diagram validasi yang diusulkan dapat dilihat pada Gambar V.12 sebagai berikut:

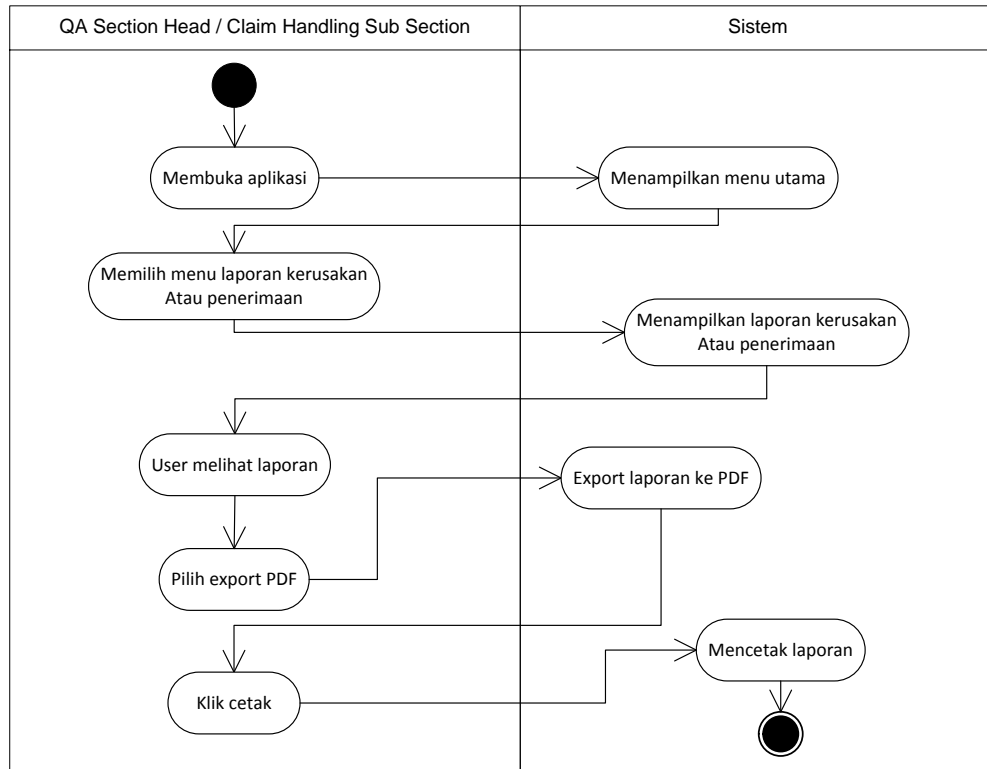


Gambar

V.12 Activity Diagram Validasi
 Sumber: Hasil Analisis Data (2016)

11. Activity Diagram View dan Mencetak Laporan Kerusakan Baterai

Activity diagram view dan mencetak laporan kerusakan baterai berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. Activity diagram view dan mencetak laporan kerusakan baterai yang diusulkan dapat dilihat pada Gambar V.13 sebagai berikut:



Gambar

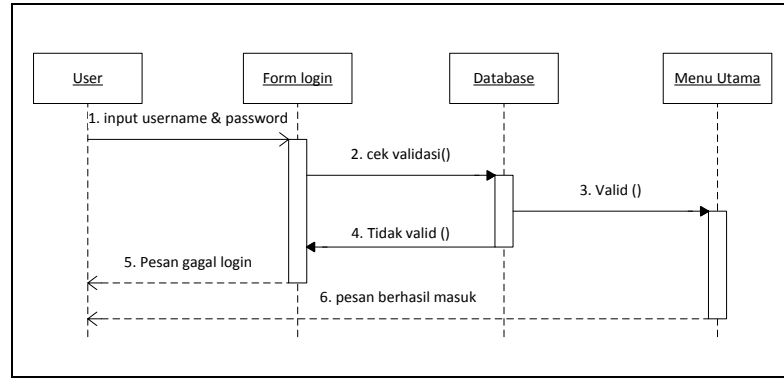
V.13 Activity Diagram View dan Mencetak Laporan AKB
 Sumber: Hasil Analisis Data (2016)

5.3.3 Sequence Diagram

Sequence Diagram menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, dimana urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada diagram. Hubungan yang ada pada gambar di bawah ini adalah proses yang dilakukan oleh sistem ketika melakukan proses yang sesuai dengan suatu objek *use case diagram*, berikut adalah *sequence diagram* pada sistem pemeriksaan produk klaim baterai aki:

1. *Sequence Diagram Proses Login*

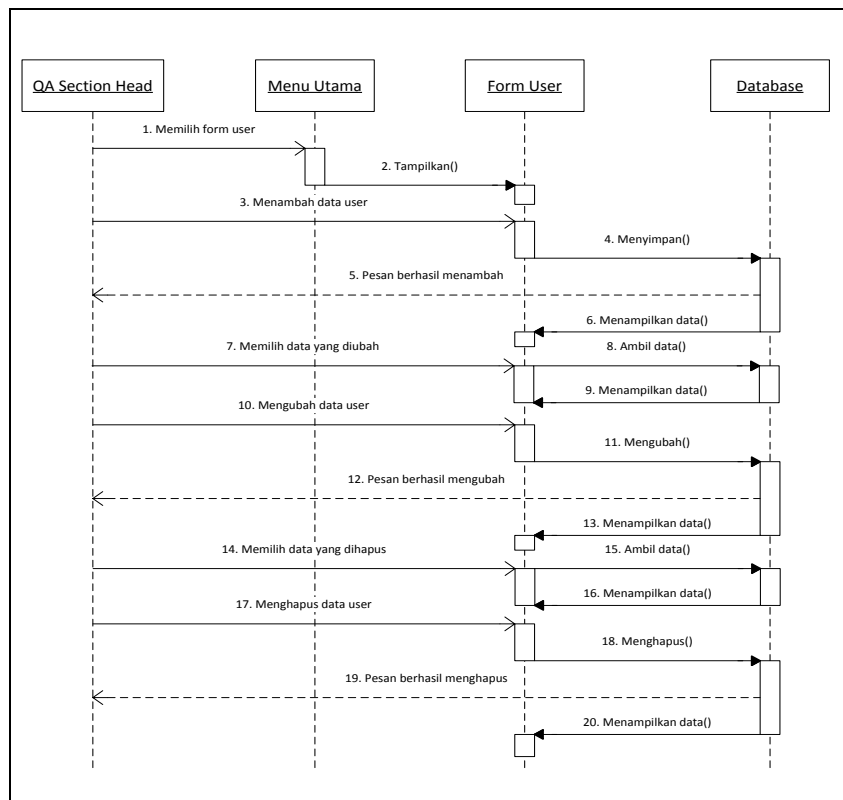
Sequence diagram login menjelaskan sebuah *sequence diagram* dalam proses *login*. Proses ini dilakukan oleh *user* sebelum masuk sistem. Adapun *sequence diagram* dari *use case login* dapat dilihat pada Gambar V.14.



Gambar V.14 *Sequence Diagram* Proses Login
 Sumber: Hasil Analisis Data (2016)

2. *Sequence Diagram* Mengelola Data User

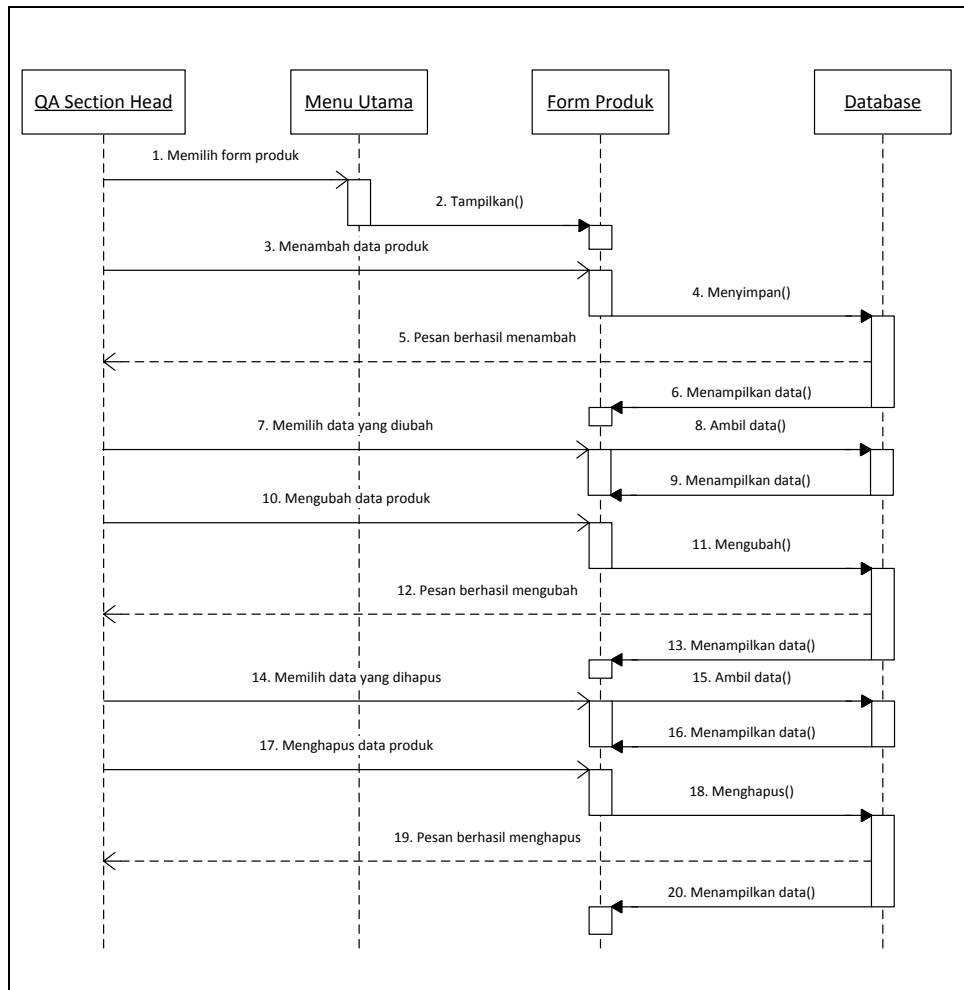
Sequence diagram mengelola data user menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus, mengubah dan mencari data user. Adapun *sequence diagram* dari use case mengelola data user dapat dilihat pada Gambar V.15 sebagai berikut:



Gambar V.15 *Sequence Diagram* Mengelola Data User
 Sumber: Hasil Analisis Data (2016)

3. *Sequence Diagram* Mengelola Data Produk

Sequence diagram mengelola data produk menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus dan mengubah data produk. Adapun *sequence diagram* dari *use case* mengelola data produk dapat dilihat pada Gambar V.16 sebagai berikut:



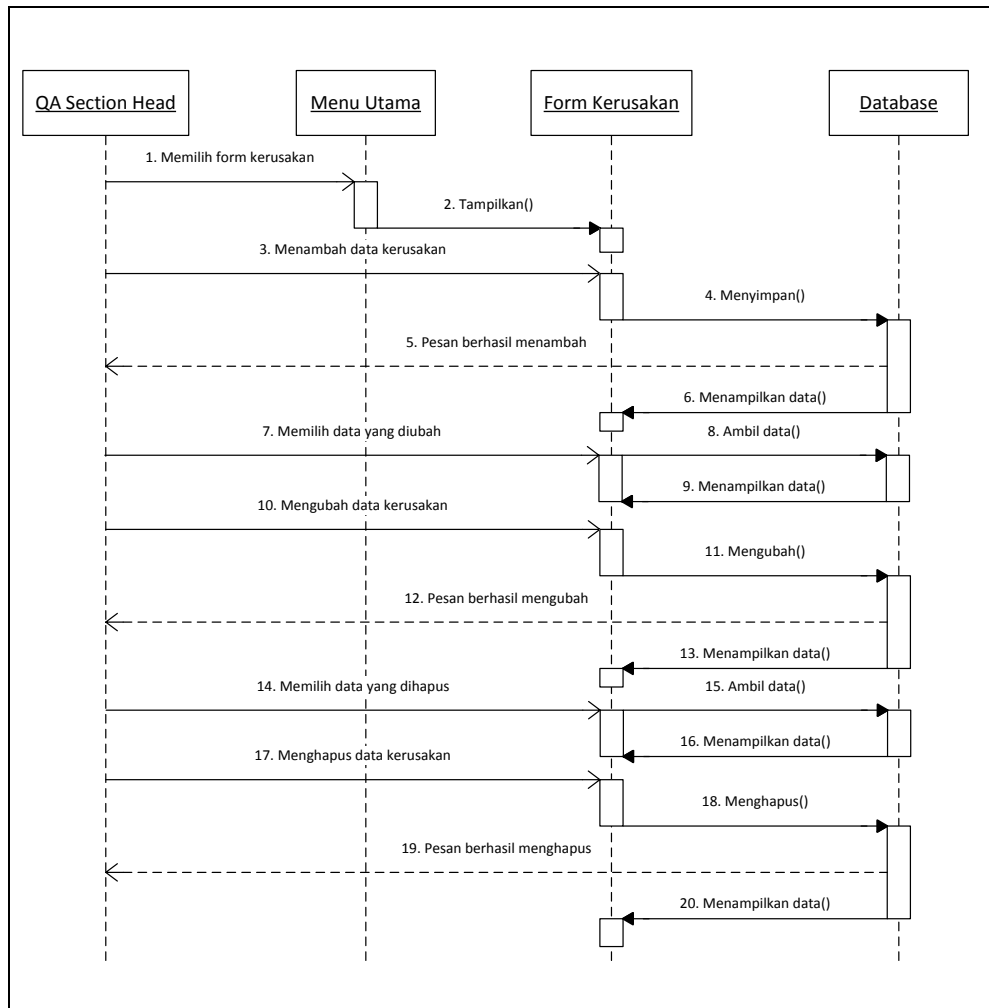
Gambar V.16

Sequence Diagram Mengelola Data Produk

Sumber: Hasil Analisis Data (2016)

4. *Sequence Diagram* Mengelola Data Kerusakan

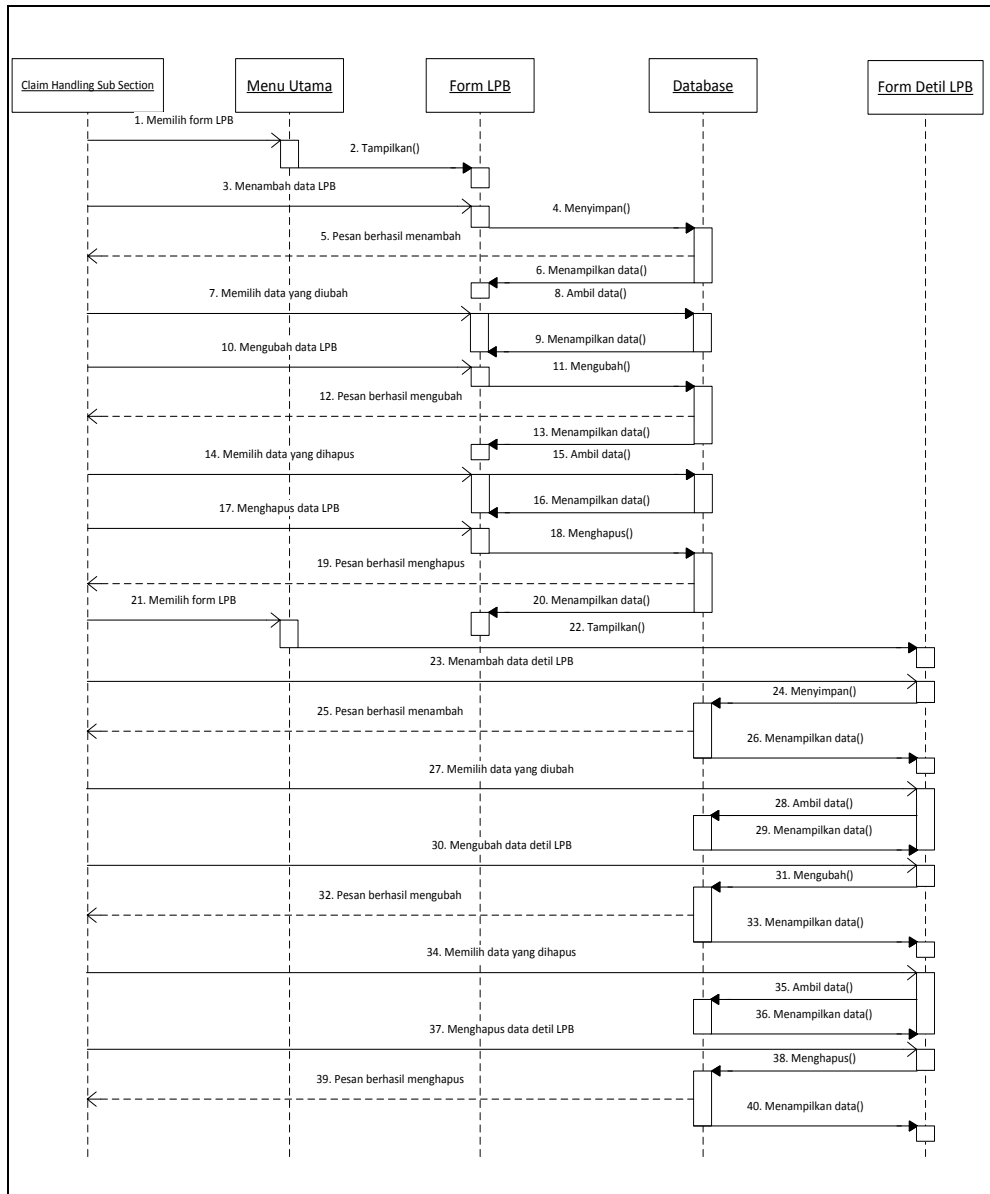
Sequence diagram mengelola data kerusakan menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus dan mengubah data kerusakan. Adapun *sequence diagram* dari *use case* mengelola data kerusakan dapat dilihat pada Gambar V.17 sebagai berikut:



Gambar V.17 *Sequence Diagram* Mengelola Data Kerusakan
 Sumber: Hasil Analisis Data (2016)

5. *Sequence Diagram* Mengelola *Form* LPB

Sequence diagram ini menjelaskan sebuah proses transaksi menambah, menghapus dan mengubah data pada *form* LPB dan *form* detail LPB. Adapun *sequence diagram* dari *use case* mengelola *form* LPB dapat dilihat pada Gambar V.18 sebagai berikut:

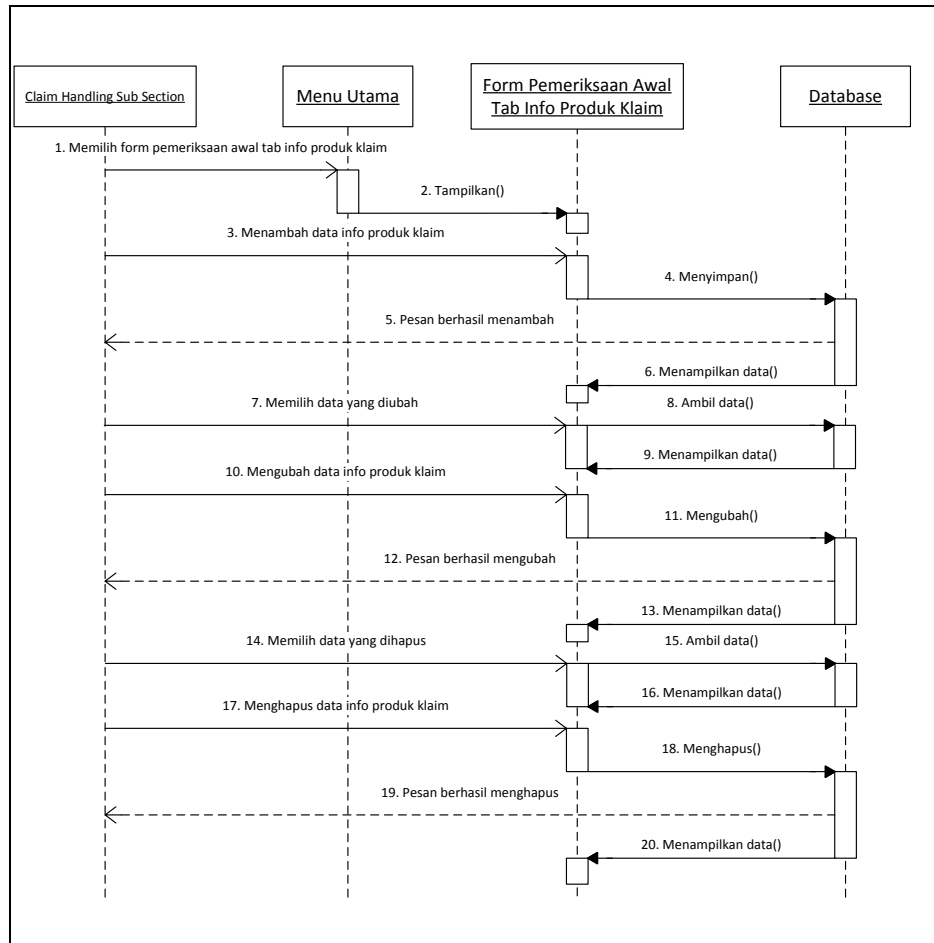


Gambar

V.18 Sequence Diagram Mengelola Form LPB
 Sumber: Hasil Analisis Data (2016)

6. Sequence Diagram Mengelola Informasi Produk Klaim

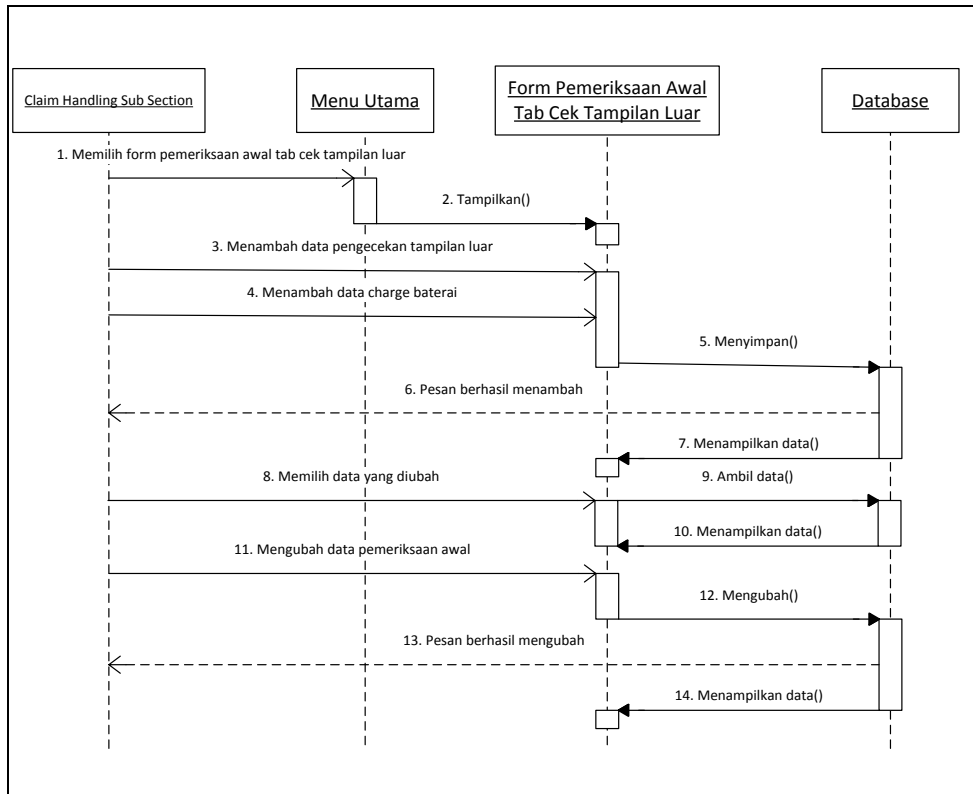
Sequence diagram mengelola informasi produk klaim menjelaskan sebuah sequence diagram dalam proses transaksi menambah, menghapus dan mengubah data pemeriksaan awal baterai. Adapun sequence diagram dari use case mengelola informasi produk klaim dapat dilihat pada Gambar V.19 sebagai berikut:



Gambar V.19 *Sequence Diagram* Mengelola Informasi Produk Klaim
 Sumber: Hasil Analisis Data (2016)

7. *Sequence Diagram* Mengelola Informasi Pemeriksaan Awal Baterai

Sequence diagram mengelola informasi pemeriksaan awal baterai menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah dan mengubah data pemeriksaan awal baterai. Adapun *sequence diagram* dari *use case* mengelola informasi pemeriksaan awal baterai dapat dilihat pada Gambar V.20 sebagai berikut:

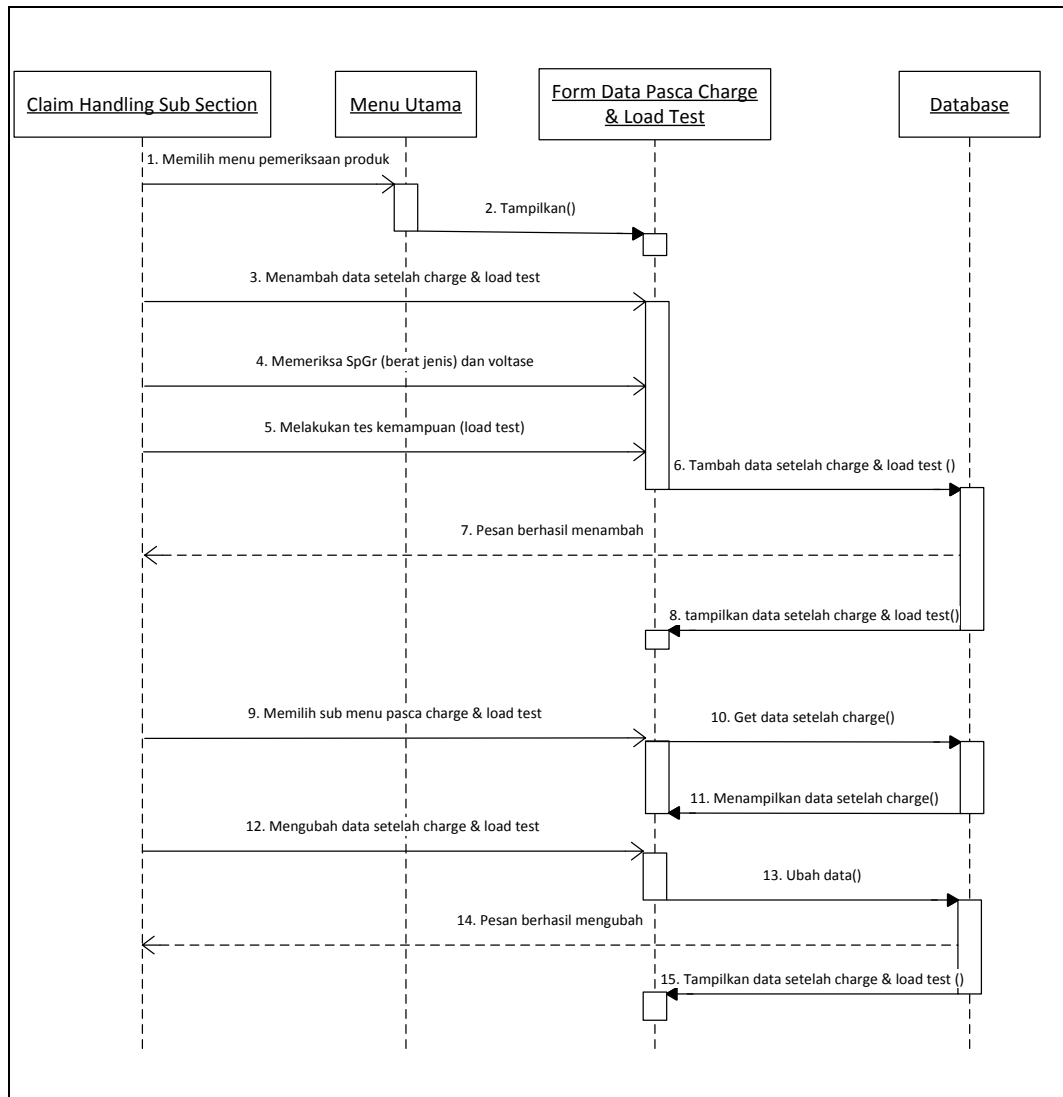


Gambar V.20

Sequence Diagram Mengelola Informasi Pemeriksaan Awal Baterai
 Sumber: Hasil Analisis Data (2016)

8. *Sequence Diagram* Mengelola Informasi Tes Kemampuan Baterai

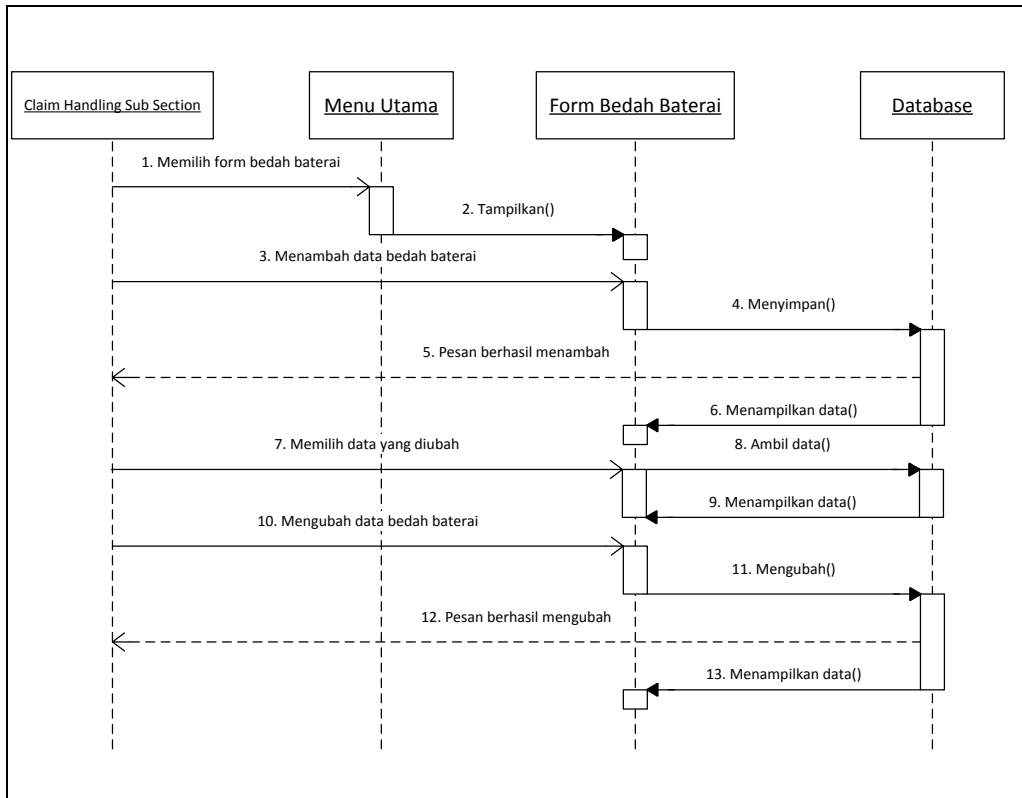
Sequence diagram mengelola informasi tes kemampuan baterai (*load test*) menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah dan mengubah data tersebut. Adapun *sequence diagram* dari *use case* mengelola informasi tes kemampuan baterai (*load test*) dapat dilihat pada Gambar V.21 sebagai berikut:



Gambar V.21 *Sequence Diagram* Mengelola Informasi Tes Kemampuan Baterai
 Sumber: Hasil Analisis Data (2016)

9. *Sequence Diagram* Mengelola Informasi Data Bedah Baterai

Sequence diagram mengelola informasi data bedah baterai menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah dan mengubah data bedah baterai. Adapun *sequence diagram* dari *use case* mengelola informasi data bedah baterai dapat dilihat pada Gambar V.22 sebagai berikut:

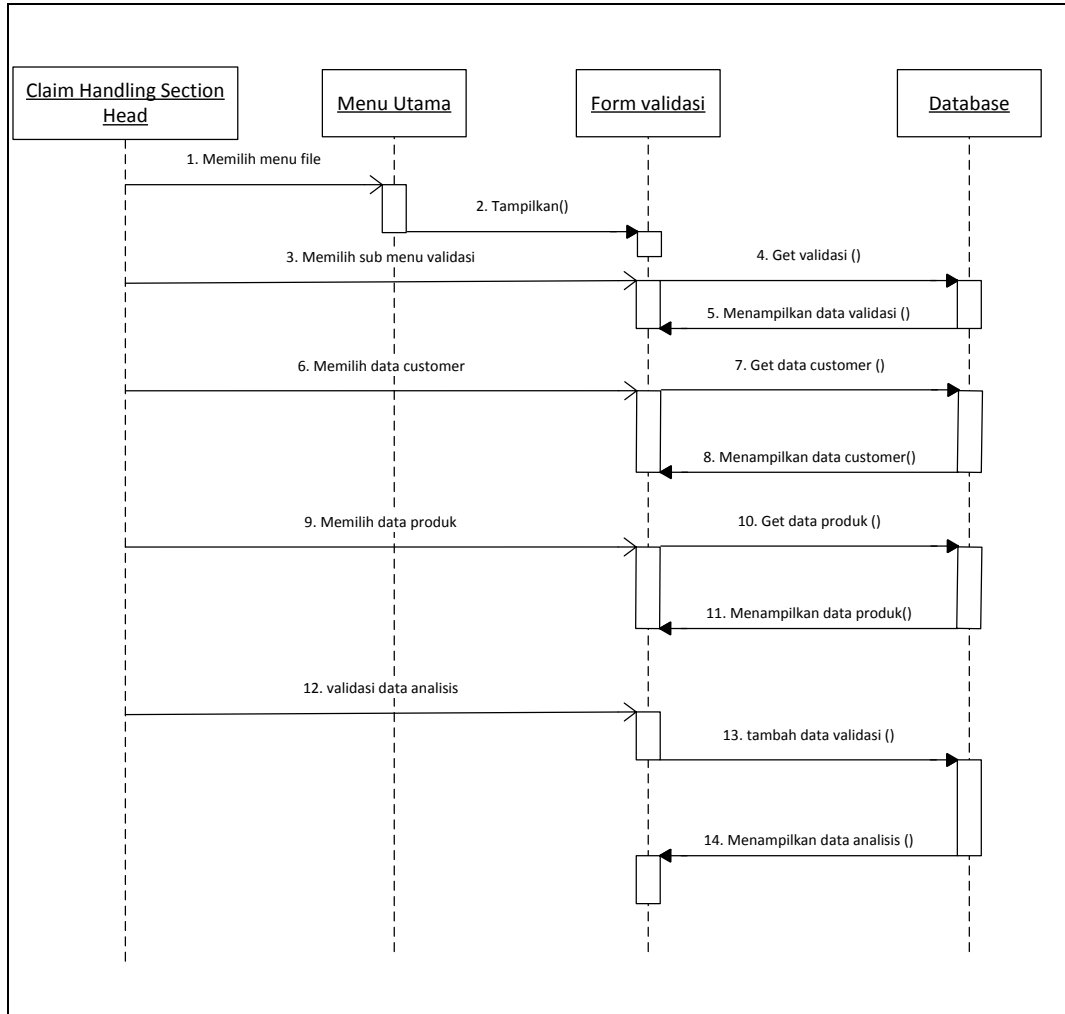


Gambar

V.22 Sequence Diagram Mengelola Informasi Data Bedah Baterai
 Sumber: Hasil Analisis Data (2016)

10. *Sequence Diagram* Validasi

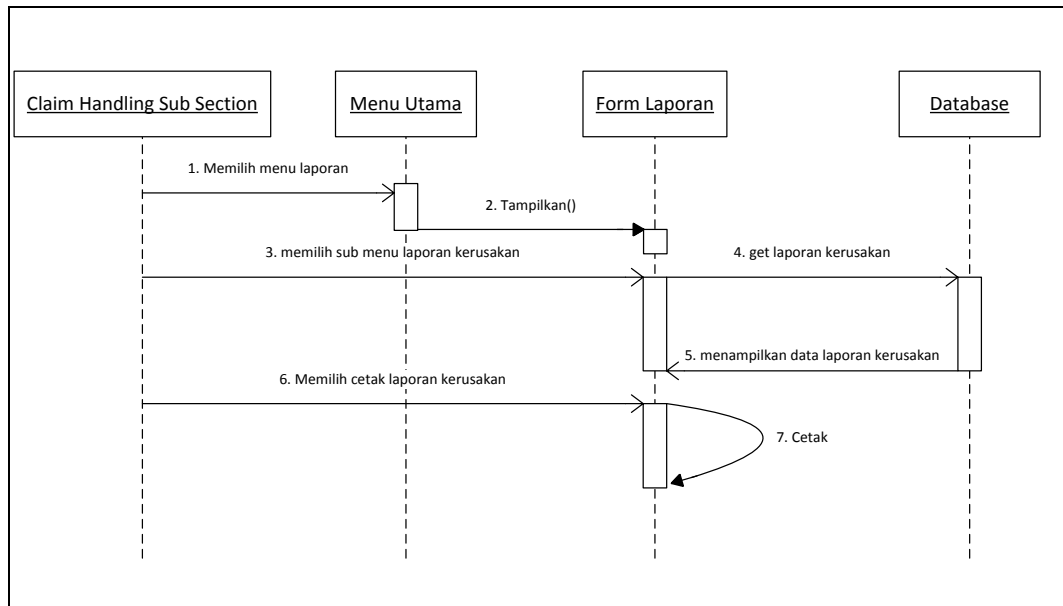
Sequence diagram validasi menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah atau mengubah data validasi baterai klaim. Adapun *sequence diagram* dari *use case* validasi dapat dilihat pada Gambar V.23 sebagai berikut:



Gambar V.23 *Sequence Diagram* Validasi
 Sumber: Hasil Analisis Data (2016)

11. *Sequence Diagram* View Dan Mencetak Laporan Kerusakan Baterai

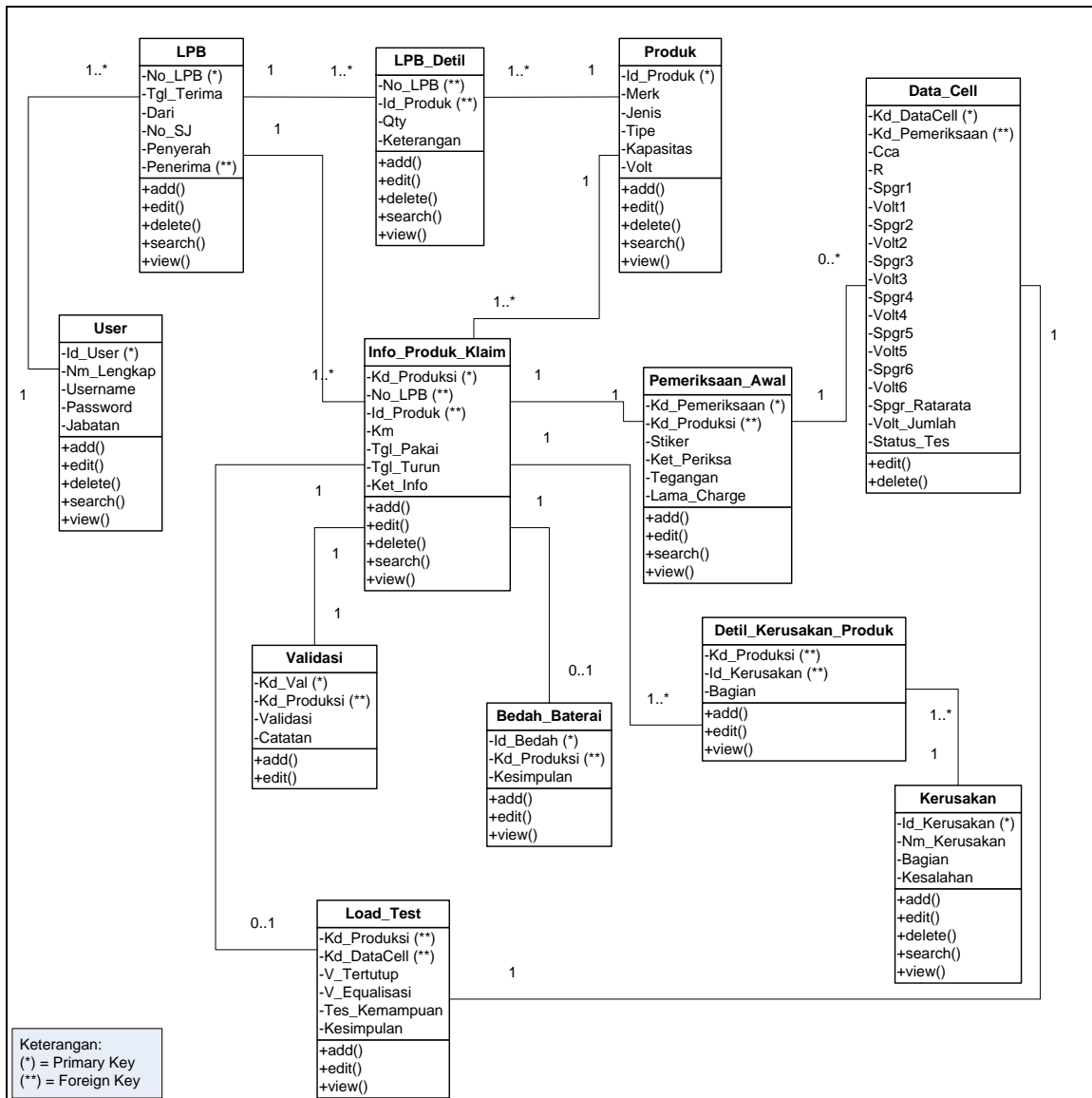
Sequence diagram view dan mencetak laporan kerusakan baterai menjelaskan sebuah *sequence diagram* dalam proses melihat dan mencetak data laporan kerusakan baterai. Adapun *sequence diagram* dari *use case* view dan mencetak laporan kerusakan baterai dapat dilihat pada Gambar V.24 sebagai berikut:



Gambar V.24 *Sequence Diagram View Dan Mencetak Laporan Kerusakan Baterai*
 Sumber: Hasil Analisis Data (2016)

5.3.4 *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem pemeriksaan produk klaim baterai aki yang diusulkan dapat dilihat pada Gambar V.25 sebagai berikut:



Gambar V.25 Class Diagram Usulan
 Sumber: Hasil Analisis Data (2016)

5.3.5 Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analisi sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen *data store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, laporan dan basis data. Berikut adalah kamus data sistem informasi pemeriksaan produk klaim baterai aki:

1. Spesifikasi Tabel *User*

Nama Tabel : *User*

Fungsi : Untuk menyimpan data *user*

Tipe : File data master

Tabel V.19 Tabel *User*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	<i>Id user</i>	<i>Id_User</i>	<i>Char</i>	3	<i>Primary key</i>
2.	Nama lengkap	<i>Nm_Lengkap</i>	<i>Varchar</i>	30	
3.	<i>Username</i>	<i>Username</i>	<i>Varchar</i>	30	
4.	<i>Password</i>	<i>Password</i>	<i>Varchar</i>	100	
5.	Jabatan	<i>Jabatan</i>	<i>Varchar</i>	30	

Sumber: Hasil Analisis Data (2016)

2. Spesifikasi Tabel Produk

Nama Tabel : Produk

Fungsi : Untuk menyimpan data produk

Tipe : File data master

Tabel V.20 Tabel Produk

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id produk	<i>Id_Produk</i>	<i>Char</i>	5	<i>Primary key</i>
2.	Merk Produk	<i>Merk</i>	<i>Varchar</i>	50	
3.	Jenis Produk	<i>Jenis</i>	<i>Varchar</i>	18	
4.	Tipe produk	<i>Tipe</i>	<i>Varchar</i>	10	
5.	Kapasitas Baterai	<i>Kapasitas</i>	<i>Integer</i>	3	
6.	Tegangan Baterai	<i>Volt</i>	<i>Integer</i>	3	

Sumber: Hasil Analisis Data (2016)

3. Spesifikasi Tabel Kerusakan

Nama Tabel : Kerusakan

Fungsi : Untuk menyimpan data kerusakan

Tipe : File data master

Tabel V.21 Tabel Kerusakan

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id kerusakan	Id_Kerusakan	<i>Char</i>	4	<i>Primary key</i>
2.	Nm kerusakan	Nm_Kerusakan	<i>Varchar</i>	100	
3.	Bagian kerusakan	Bagian	<i>Varchar</i>	10	
4.	Kesalahan	Kesalahan	<i>Varchar</i>	8	

Sumber: Hasil Analisis Data (2016)

4. Spesifikasi Tabel LPB (Laporan Penerimaan Barang)

Nama Tabel : LPB

Fungsi : Untuk menyimpan data penerimaan barang

Tipe : File data transaksi

Tabel V.22 Tabel LPB

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	No LPB	No_LPB	<i>Varchar</i>	10	<i>Primary key</i>
2.	Tanggal terima	Tgl_Terima	<i>Date</i>	-	
3.	Asal barang	Dari	<i>Varchar</i>	30	
4.	No surat jalan	No_SJ	<i>Varchar</i>	20	
5.	Penyerah barang	Penyerah	<i>Varchar</i>	20	
6.	Penerima barang	Penerima	<i>Char</i>	3	<i>Foreign key</i>

Sumber: Hasil Analisis Data (2016)

5. Spesifikasi Tabel Detil LPB

Nama Tabel : LPB_Detil

Fungsi : Untuk menyimpan data detil LPB

Tipe : File data transaksi

Tabel V.23 Tabel Detil LPB

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	No LPB	No_LPB	<i>Varchar</i>	10	<i>Composite Key</i>
2.	Id Produk	Id_Produk	<i>Char</i>	5	<i>Composite Key</i>
3.	Quantity	Qty	<i>Integer</i>	3	
4.	Keterangan	Keterangan	<i>Varchar</i>	100	

Sumber: Hasil Analisis Data (2016)

6. Spesifikasi Tabel Info Produk Klaim

Nama Tabel : Info_Produk_Klaim

Fungsi : Untuk menyimpan data info produk klaim baterai

Tipe : File data transaksi

Tabel V.24 Tabel Info Produk Klaim

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kd produksi	Kd_Produksi	<i>Char</i>	11	<i>Primary key</i>
2.	No LPB	No_LPB	<i>Varchar</i>	10	<i>Foreign key</i>
3.	Id produksi	Id_Produk	<i>Char</i>	5	<i>Foreign key</i>
4.	Km kendaraan	Km	<i>Integer</i>	5	
5.	Tanggal pakai	Tgl_Pakai	<i>Date</i>	-	
6.	Tanggal turun	Tgl_Turun	<i>Date</i>	-	
7.	Keterangan	Ket_Info	<i>Varchar</i>	20	

Sumber: Hasil Analisis Data (2016)

7. Spesifikasi Tabel Pemeriksaan Awal Baterai

Nama Tabel : Pemeriksaan_Awal

Fungsi : Untuk menyimpan data pemeriksaan awal baterai aki

Tipe : File data transaksi

Tabel V.25 Tabel Pemeriksaan Awal Baterai

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kd Periksa	Kd_Pemeriksaan	Char	6	Primary key
2.	Kd produksi	Kd_Produksi	Char	11	Foreign key
3.	Stiker produk	Stiker	Varchar	15	
4.	Keterangan	Ket_Periksa	Varchar	15	
5.	Tegangan	Tegangan	Integer	4	
6.	Charge (Jam)	Lama_Charge	Integer	2	

Sumber: Hasil Analisis Data (2016)

8. Spesifikasi Tabel Data Cell Baterai

Nama Tabel : Data_Cell

Fungsi : Untuk menyimpan data masing-masing cell baterai aki

Tipe : File data transaksi

Tabel V.26 Tabel Data Cell Baterai

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kd Data Cell	Kd_DataCell	Char	6	Primary key
2.	Kd Periksa	Kd_Pemeriksaan	Char	6	Foreign key
3.	Cold Cranking Amps	Cca	Integer	4	
4.	Reserve	R	Integer	4	
5.	Spgr cell 1	Spgr1	Integer	4	

Tabel V.26 Tabel Data *Cell* Baterai

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
6.	Volt <i>cell</i> 1	Volt1	<i>Float</i>		
7.	Spgr <i>cell</i> 2	Spgr2	<i>Integer</i>	4	
8.	Volt <i>cell</i> 2	Volt2	<i>Float</i>		
9.	Spgr <i>cell</i> 3	Spgr3	<i>Integer</i>	4	
10.	Volt <i>cell</i> 3	Volt3	<i>Float</i>		
11.	Spgr <i>cell</i> 4	Spgr4	<i>Integer</i>	4	
12.	Volt <i>cell</i> 4	Volt4	<i>Float</i>		
13.	Spgr <i>cell</i> 5	Spgr5	<i>Integer</i>	4	
14.	Volt <i>cell</i> 5	Volt5	<i>Float</i>		
15.	Spgr <i>cell</i> 6	Spgr6	<i>Integer</i>	4	
16.	Volt <i>cell</i> 6	Volt6	<i>Float</i>		
17.	Spgr Rata-rata	Spgr_Ratarata	<i>Float</i>		
18.	Jumlah Voltase	Volt_Jumlah	<i>Float</i>		
19.	Status tes	Status_Tes	<i>Char</i>	7	

Sumber: Hasil Analisis Data (2016)

9. Spesifikasi Tabel *Load Test*

Nama Tabel : Load_Test

Fungsi : Untuk menyimpan data *load test* baterai aki

Tipe : File data transaksi

Tabel V.27 Tabel *Load Test*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kd produksi	Kd_Produksi	<i>Char</i>	11	<i>Composite Key</i>
2.	Kd data <i>cell</i>	Kd_DataCell	<i>Char</i>	6	<i>Composite</i>

Tabel V.27 Tabel *Load Test*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
					<i>Key</i>
3.	Tegangan tertutup	V_Tertutup	<i>Char</i>	4 (Lanjutan)	
4.	Tegangan equalisasi	V_equalisasi	<i>Char</i>	4	
5.	Tes Kemampuan	Tes_Kemampuan	<i>Varchar</i>	6	
6.	Kesimpulan	Kesimpulan	<i>Varchar</i>	20	

Sumber: Hasil Analisis Data (2016)

10. Spesifikasi Tabel Bedah Baterai

Nama Tabel : Bedah_Baterai

Fungsi : Untuk menyimpan data pembedahan baterai aki

Tipe : File data transaksi

Tabel V.28 Tabel Bedah Baterai

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Id Bedah	Id_Bedah	<i>Char</i>	5	<i>Primary key</i>
2.	Kd produksi	Kd_Produksi	<i>Char</i>	11	<i>Foreign key</i>
3.	Kesimpulan	Kesimpulan	<i>Varchar</i>	100	

Sumber: Hasil Analisis Data (2016)

11. Spesifikasi Tabel Detil Kerusakan Produk

Nama Tabel : Detil_Kerusakan_Produk

Fungsi : Untuk menyimpan data detil kerusakan produk

Tipe : File data transaksi

Tabel V.29 Tabel Detil Kerusakan Produk

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kd produksi	Kd_Produksi	<i>Char</i>	11	<i>Composite</i>

Tabel V.29 Tabel Detil Kerusakan Produk

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
					<i>Key</i>
2.	Id Kerusakan	Id_Kerusakan	<i>Char</i>	4	<i>Composite Key</i>
3.	Bagian	Bagian	<i>Varchar</i>	10	

Sumber: Hasil Analisis Data (2016)

12. Spesifikasi Tabel Validasi

Nama Tabel : Validasi

Fungsi : Untuk menyimpan data validasi klaim

Tipe : File data transaksi

Tabel V.30 Tabel Validasi

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kd Validasi	Kd_Val	<i>Char</i>	5	<i>Primary key</i>
2.	Kd produksi	Kd_Produksi	<i>Char</i>	11	<i>Foreign key</i>
3.	Hasil Validasi	Validasi	<i>Varchar</i>	20	
4.	Catatan	Catatan	<i>Varchar</i>	100	

Sumber: Hasil Analisis Data (2016)

5.3.5.1 Rincian Elemen

Elemen atau *field* yang ada pada database terkadang membuat bingung orang yang membacanya. Kebingungan tersebut sering kali terkait dengan masalah kegunaan dari masing-masing elemen, apalagi kata yang digunakan tidaklah umum. Penjelasan rinci yang perlu diketahui dari beberapa elemen pada aplikasi usulan dapat dilihat pada Tabel V.33 berikut:

Tabel V.31 Tabel Rincian Elemen

No.	Nama Elemen	Penjelasan
1.	No LPB	Merupakan nomor untuk mendeskripsikan data LPB yang terdiri dari sepuluh digit. Tiga digit pertama menunjukkan bulan yang dikonotasikan dengan angka romawi, dua digit untuk tahun dan lima digit untuk nomor urutnya.
2.	Tanggal Terima	Tanggal penerimaan barang oleh bagian <i>claim handling</i> .
3.	Dari	Mendeskripsika nama customer atau instansi yang mengirim produk baterai.
4.	No Surat Jalan	Merupakan nomor yang didapatkan dari surat jalan perusahaan pemohon. Nomor tersebut bisa bervariasi digit dan komposisinya.
5.	Penyerah	Nama orang yang mengantar dan menyerahkan produk.
6.	Penerima	Nama petugas yang menerima produk.
7.	Id Produk	Merupakan nomor id untuk mendiskripsikan data produk baterai aki.
8.	Qty	Singkatan dari <i>quantity</i> untuk menjelaskan jumlah barang.
9.	Merk	Menjelaskan merk suatu produk baterai aki.
10.	Jenis	Menjelaskan jenis suatu produk baterai aki. Dapat berupa <i>motorcycle battery</i> , <i>automobile battery</i> dan <i>industry battery</i> . (Lanjutan)
11.	Tipe	Menjelaskan tipe suatu produk baterai aki.
12.	Kapasitas	Menjelaskan kapasitas suatu produk baterai aki dalam satuan <i>Ampere Hour (AH)</i> .
13.	Volt	Menjelaskan tegangan suatu produk baterai aki dalam satuan <i>Volt (V)</i> .
14.	Id User	Merupakan nomor id untuk mendiskripsikan data pengguna aplikasi.
15.	Kd Produksi	Merupakan kode untuk mendeskripsikan data produksi suatu produk baterai yang terdiri dari sebelas digit. satu digit huruf pertama untuk <i>shift</i> produksi, satu digit angka

Tabel V.31 Tabel Rincian Elemen

No.	Nama Elemen	Penjelasan
		untuk <i>line</i> produksi, dua digit angka untuk tahun, satu digit huruf untuk bulan, dua digit angka untuk tanggal dan empat digit terakhir untuk nomor urut produk.
16.	Km	Menunjukkan kilometer kendaraan yang telah memakai produk baterai aki.
17.	Tanggal Pakai	Menyatakan tanggal baterai baru pertama kali dipakai.
18.	Tanggal Turun	Menyatakan tanggal baterai di lepas dari kendaraan.
19.	Kd Pemeriksaan	Menyatakan kode periksa pada tabel pemeriksaan awal dan hanya sebagai <i>primary key</i> .
20.	<i>Cover</i>	Merupakan kesalahan produk pada bagian <i>cover</i> yang diimplementasikan dengan empat digit id kerusakan.
21.	<i>Sealing</i>	Merupakan kesalahan produk pada bagian <i>seal</i> yang diimplementasikan dengan empat digit id kerusakan.
22.	Terminal	Merupakan kesalahan produk pada bagian terminal yang diimplementasikan dengan empat digit id kerusakan.
23.	Elektrolit	Merupakan kesalahan produk pada bagian elektrolit yang diimplementasikan dengan empat digit id kerusakan.
24.	Stiker	Merupakan kesalahan produk pada bagian stiker yang diimplementasikan dengan empat digit id kerusakan.
25.	<i>Lama Charge</i>	Menyatakan lama <i>charge</i> yang didapatkan setelah adanya perhitungan tertentu.
26.	Kd Data Cell	Menyatakan kode data cell pada tabel pemeriksaan awal dan hanya sebagai <i>primary key</i> .
27.	Cca	Singkatan dari <i>Cold Cranking Ampere</i> yaitu nilai suatu baterai aki untuk menunjukkan kekuatan baterai melakukan stater awal dalam keadaan dingin (nol derajat <i>fahrenheit</i>). Nilai bisa dari 400 hingga 1000 Cca.
28.	R	Singkatan dari <i>Reserve Capacity</i> yaitu kemampuan baterai menyimpan listrik saat kelistrikan pada

Tabel V.31 Tabel Rincian Elemen

No.	Nama Elemen	Penjelasan
		kendaraan mengalami masalah dan tidak bisa membalikan arus ke dalam baterai.
29.	SpGr	Singkatan dari <i>Specific Gravity</i> yaitu nilai untuk mengukur berat jenis elektrolit setiap <i>cell</i> baterai. (Lanjutan)
30.	Volt	Yaitu nilai untuk mengukur tegangan tiap <i>cell</i> baterai.
31.	SpGr Rata-rata	Yaitu perhitungan rata-rata dari berat jenis elektrolit setiap <i>cell</i> .
32.	Voltase Jumlah	Yaitu perhitungan jumlah keseluruhan dari tegangan setiap <i>cell</i> .
33.	Id Kerusakan	Menyatakan id kerusakan pada tabel kerusakan dan hanya sebagai <i>primary key</i> .
34.	Nama Kerusakan	Menunjukkan nama dari kerusakan atau fenomena klaim yang pernah terjadi.
35.	Bagian	Menyatakan bagian yang menjadi sumber kerusakan. Terdiri dari <i>cover</i> , <i>sealing</i> , elektrolit, terminal, plat positif, plat negatif, separator dan endapan.
36.	Kesalahan	Menyatakan dugaan pelaku kerusakan yaitu akibat kesalahan pabrik atau kesalahan <i>customer</i> .
37.	Id Bedah	Menyatakan id bedah pada tabel bedah baterai dan hanya sebagai <i>primary key</i> .
38.	Plat Positif	Merupakan kesalahan produk pada bagian plat positif yang diimplementasikan dengan empat digit id kerusakan.
39.	Plat Negatif	Merupakan kesalahan produk pada bagian plat negatif yang diimplementasikan dengan empat digit id kerusakan.
40.	Separator	Merupakan kesalahan produk pada bagian separator yang diimplementasikan dengan empat digit id kerusakan.
41.	Endapan	Merupakan kesalahan produk pada bagian endapan yang

Tabel V.31 Tabel Rincian Elemen

No.	Nama Elemen	Penjelasan
		diimplementasikan dengan empat digit id kerusakan.
42.	Tes Kemampuan	Tes kemampuan atau <i>load test</i> yang di tampilkan dengan indikator warna hijau, kuning atau merah.
43.	Kd Validasi	Menyatakan kd validasi pada tabel validasi dan hanya sebagai <i>primary key</i> .
44.	Kesimpulan	Menyatakan kesimpulan akhir dari proses klaim yaitu akibat <i>manufacturing fault</i> atau <i>handling fault</i> .
45.	Validasi	Menyatakan hasil validasi dari suatu produk baterai apakah masih dalam proses, tidak divalidasi atau sudah divalidasi.

Sumber: Hasil Analisis Data (2016)

5.3.5.2 Perhitungan Charge

Dalam proses yang berlangsung, kegiatan *charge* yang dilakukan oleh bagian *claim handling* merupakan salah satu kegiatan yang penting. Berikut ini adalah tahap dan rumus perhitungan *charge* menurut standar PT Nipress Tbk:

1. Menghitung Rata-rata SpGr

$$\overline{SpGr} = \frac{(Spgr\ cell\ 1 + Spgr\ cell\ 2 + Spgr\ cell\ 3 + Spgr\ cell\ 4 + Spgr\ cell\ 5 + Spgr\ cell\ 6)}{6}$$

Keterangan:

- a. \overline{SpGr} = *Specific Gravity* rata-rata
- b. *Spgr cell n* = *Specific Gravity cell* ke-n

2. Menghitung Jumlah Voltase

$$V = (Volt\ cell\ 1 + Volt\ cell\ 2 + Volt\ cell\ 3 + Volt\ cell\ 4 + Volt\ cell\ 5 + Volt\ cell\ 6)$$

Keterangan:

- a. V = Jumlah tegangan keseluruhan

b. *Volt cell n* = Tegangan *cell* ke-n

3. Menentukan *Level Charge*

Dalam menentukan *level charge*, dibutuhkan SpGr rata-rata, jumlah voltase dan sebuah tabel yang sudah digunakan perusahaan sebagai standar acuan *level charge*. Berikut adalah tabel *level charge* yang tertera pada Tabel V.34.

Tabel V.32 Tabel *Level Charge*

Level Charge	SpGr	V
100%	1280	12,80
75%	1245	12,60
50%	1210	12,40
25%	1175	12,20
Discharge	1140	12,00

Sumber: PT Nipress Tbk, 2015

4. Menghitung *Loss Charge*

$$\text{Loss Charge (\%)} = 100 - \text{Level charge (\%)}$$

5. Menghitung Arus *Charging*

$$A = \frac{1}{10} \times \text{Capacity}$$

Keterangan:

a. A = Arus *Charging* (*Ampere*)

b. *Capacity* = Kapasitas Baterai (*Ampere Hour*)

6. Menghitung Lama *Charge*

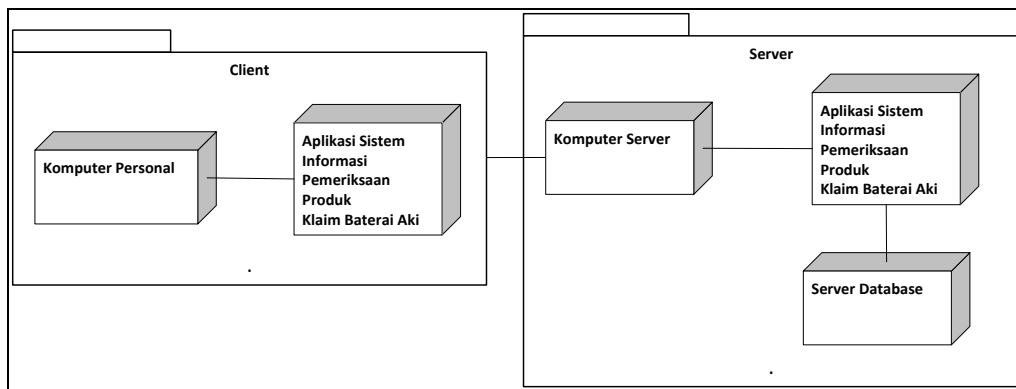
$$\text{Hour} = \frac{\text{Loss Charge (\%)} \times \text{Capacity} \times 1,2}{A}$$

Keterangan:

- a. *Hour* = Lama Charge (Jam)
- b. *Capacity* = Kapasitas Baterai (*Ampere Hour*)
- c. *A* = Arus Charging (*Ampere*)

5.3.6 Deployment Diagram

Deployment diagram digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. *Deployment diagram* sistem informasi pemeriksaan produk klaim baterai aki yang diusulkan dapat dilihat pada Gambar V.26.



Gambar

V.26 *Deployment Diagram*
Sumber: Hasil Analisis Data (2016)

Berikut adalah penjelasan Gambar V.26 *deployment diagram* sistem informasi pemeriksaan produk klaim baterai aki:

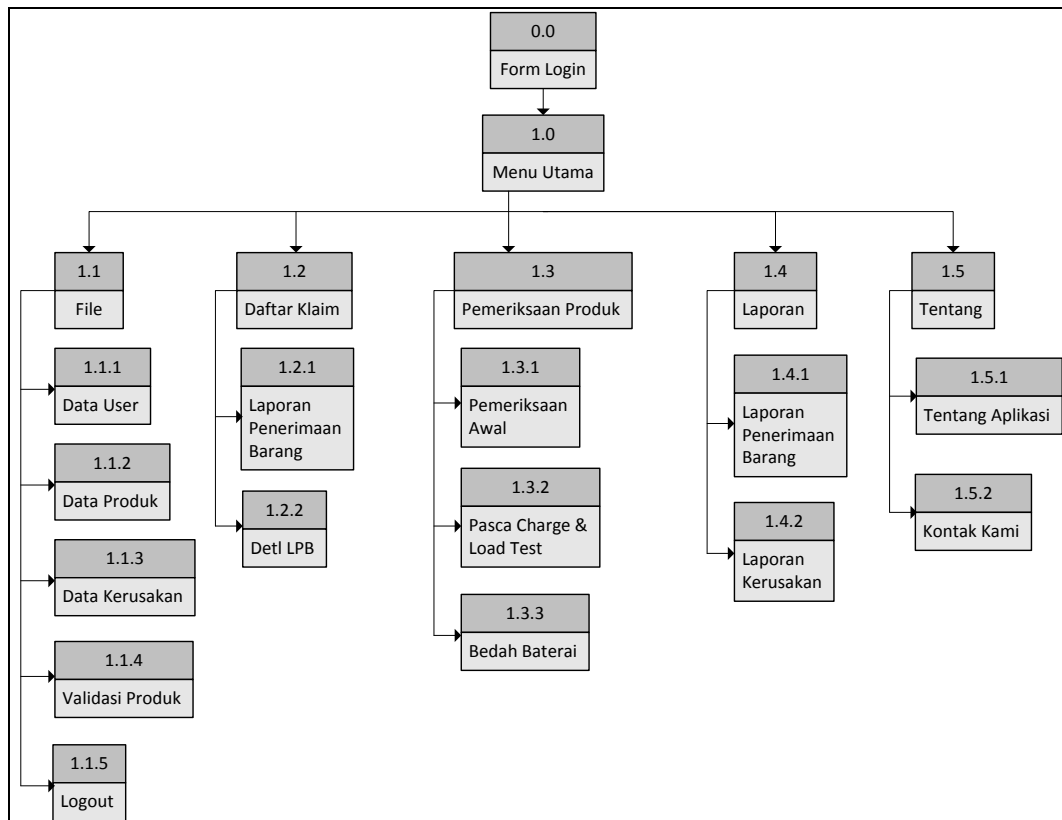
1. *Client* adalah komputer *client* yang terinstal dan hanya menggunakan aplikasi sistem informasi pemeriksaan produk klaim baterai aki. *Client* akan selalu terhubung dengan server.
2. *Server* aplikasi sistem informasi pemeriksaan produk klaim baterai aki terdiri dari komputer *server*, aplikasi sistem informasi pemeriksaan produk klaim baterai aki, dan *server* database.

5.4 Analisis Desain Program

Tahap ini merupakan tahap kedua dalam metodologi prototipe evolusioner, yaitu tahap membuat sebuah prototipe dari program/aplikasi. Dimulai dengan analisis desain usulan meliputi pembuatan struktur menu program, *flowchart* program, dan *interface* program sampai dengan program dapat dijalankan.

5.4.1 HIPO (*Hierarchy plus Input-Process-Output*)

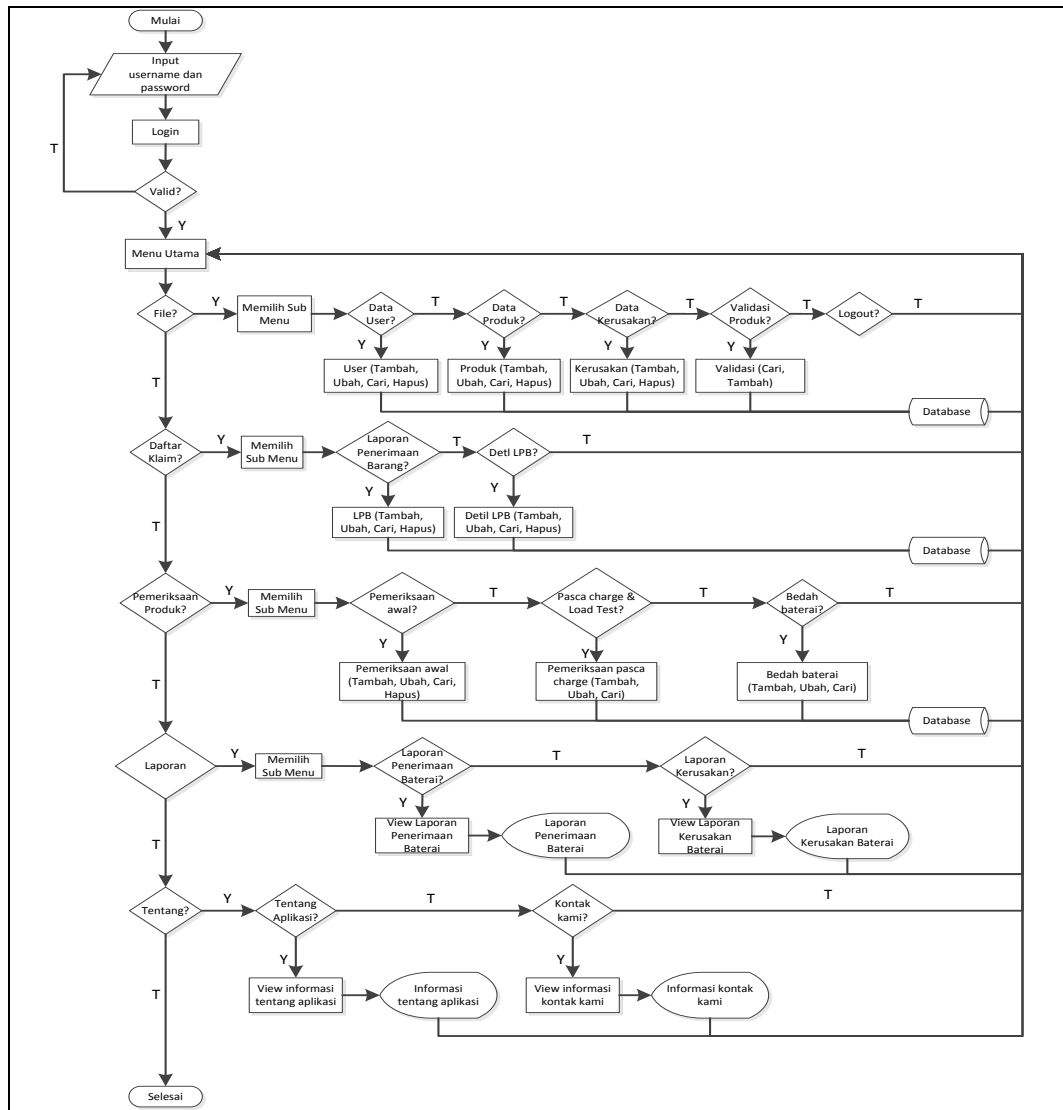
Hierarchy plus Input-Process-Output (HIPO) digunakan untuk mendokumentasikan sebuah struktur yang menggambarkan hubungan antar fungsi dalam program secara hierarkis. Diagram ini memuat semua modul yang ada dalam sistem beserta nama dan nomornya. Perancangan HIPO aplikasi usulan sistem informasi pemeriksaan produk klaim baterai aki dapat dilihat pada Gambar V.27 sebagai berikut:



Gambar V.27 HIPO Sistem Informasi Pemeriksaan Produk Klaim Baterai Aki Usulan
Sumber: Hasil Analisis Data (2016)

5.4.2 *Flowchart* Program

Flowchart yang digunakan untuk mendokumentasikan aplikasi sistem klaim baterai aki ini menggunakan bagan alir logika program (*program logic flowchart*). Bagan alir ini digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika, bukan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Gambar V.28 menunjukkan *program logic flowchart* aplikasi sistem informasi pemeriksaan produk klaim baterai aki:



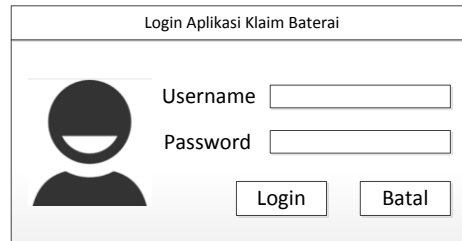
Gambar V.28 Program Logic Flowchart Aplikasi Sistem Usulan
Sumber: Hasil Analisis Data (2016)

5.4.3 Perancangan *Interface* Program

Rancangan *interface* dari program sistem informasi pemeriksaan produk klaim baterai aki ini adalah sebagai berikut:

1. *Form Login*

Form login adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, pengguna harus memasukkan *username* dan sandi yang benar. Rancangan *form login* dapat dilihat pada Gambar V.29.



The image shows a login window titled "Login Aplikasi Klaim Baterai". On the left is a simple black silhouette of a person's head and shoulders. To the right of the icon are two text input fields: "Username" and "Password". Below these fields are two buttons: "Login" and "Batal".

Gambar V.29 *Interface Form Login*
Sumber: Hasil Analisis Data (2016)

2. *Form Menu Utama*

Form menu utama adalah tampilan awal pada aplikasi yang berisi lima menu. Menu tersebut adalah File, Daftar Klaim, Pemeriksaan Produk, Laporan dan Tentang. Rancangan menu utama dapat dilihat pada Gambar V.30 berikut:



The image shows a main menu window titled "Aplikasi Claim Handling". At the top is a menu bar with "File", "Daftar Klaim", "Pemeriksaan Produk", "Laporan", and "Tentang". Below the menu bar, it says "Anda Login Sebagai: Username (Jabatan)". In the center is the "NS Battery" logo. To the right is a table titled "Produk Tidak Divalidasi" with columns "Customer", "Id Produk", "No LPB", and "Merk". Below the table is a "Catatan" field and a "Refresh" button. At the bottom left, it displays "Tanggal / Bulan / Tahun" and "Waktu AM/PM".

Gambar V.30 *Interface Form Menu Utama*
Sumber: Hasil Analisis Data (2016)

3. *Form Data User*

Form data user adalah *form* yang digunakan untuk mengelola data master *user*. Rancangan *interface* dari *form data user* dapat dilihat pada Gambar V.31 berikut:

Data Master User

Form User

Id User :

Nama Lengkap :

Username :

Password :

Jabatan : ▼

Id User	Nama Lengkap	Username	Password	Jabatan

Gambar V.31 *Interface Form Data User*
 Sumber: Hasil Analisis Data (2016)

4. *Form Data Produk*

Form data produk adalah form yang digunakan untuk mengelola data master produk baterai aki. Rancangan *interface* dari *form* data produk dapat dilihat pada Gambar V.32 berikut:

Data Master Produk Baterai Aki

Form Produk

Id Produk :

Merk :

Jenis : ▼

Tipe :

Kapasitas : AH

Cari berdasarkan Merk / Tipe baterai aki :

Id Produk	Merk	Jenis	Tipe	Kapasitas

Gambar

V.32 *Interface Form Data Produk*
 Sumber: Hasil Analisis Data (2016)

5. *Form Data Kerusakan*

Form data kerusakan adalah *form* yang digunakan untuk menyimpan data jenis kerusakan produk. Rancangan *interface* dari *form* data kerusakan produk dapat dilihat pada Gambar V.33 berikut:

Id Kerusakan	Nama Kerusakan	Bagian	Kesalahan

Gambar V.33 *Interface Form Data Kerusakan*
Sumber: Hasil Analisis Data (2016)

6. *Form Laporan Penerimaan Barang*

Form laporan penerimaan barang (LPB) adalah *form* yang digunakan untuk menyimpan data penerimaan produk baterai yang dikirimkan oleh *customer*. Rancangan *interface* dari *form* laporan penerimaan barang dapat dilihat pada Gambar V.34 berikut:

Id Kerusakan	Nama Kerusakan	Bagian	Kesalahan

Gambar V.34 *Interface Form Laporan Penerimaan Barang*
Sumber: Hasil Analisis Data (2016)

7. *Form* Detil Laporan Penerimaan Barang

Form detil laporan penerimaan barang adalah *form* yang digunakan untuk menyimpan data rincian LPB produk baterai aki. Rancangan *interface* dari *form* detil laporan penerimaan barang dapat dilihat pada Gambar V.35 berikut:

No LPB	Id Produk	Quantity	Keterangan

Gambar V.35 *Interface Form* Detil Laporan Penerimaan Barang

Sumber: Hasil Analisis Data (2016)

8. *Form* Pemeriksaan Awal

Form pemeriksaan awal adalah *form* yang digunakan untuk menyimpan info produk baterai dan menyimpan data yang berhubungan dengan tampilan luar produk. *Form* pemeriksaan awal dibagi menjadi dua *tab* yaitu *tab* info produk klaim dan *tab* cek tampilan luar. Rancangan *interface* dari *form* pemeriksaan awal dapat dilihat pada Gambar V.36 dan Gambar V.37 berikut:

Pemeriksaan Awal

Info Produk Klaim	Cek Tampilan Luar
Kode Produksi : <input type="text"/>	<input type="button" value="Pilih"/>
No LPB : <input type="text"/>	
No SJ : <input type="text"/>	<input type="button" value="Pilih"/>
Tanggal Terima : <input type="text"/>	
Id Produksi : <input type="text"/>	
Tipe : <input type="text"/>	
ASSY / Jenis : <input type="text"/>	
KM Kendaraan : <input type="text"/> KM	
Tanggal Pakai : <input type="text"/> <input type="checkbox"/>	
Tanggal Turun : <input type="text"/> <input type="checkbox"/>	
Keterangan : <input type="text"/>	
<input type="button" value="Simpan"/> <input type="button" value="Ubah"/> <input type="button" value="Hapus"/> <input type="button" value="Batal"/>	

Data Produk Klaim						
Kode Produksi : <input type="text"/> <input type="button" value="Cari"/>						
Kd Produksi	No LPB	Id Produk	KM	Tgl Pakai	Tgl urun	Keterangan

Data Produk Klaim						
Kode Produksi : <input type="text"/> <input type="button" value="Cari"/>						
Kd Produksi	No LPB	Id Produk	KM	Tgl Pakai	Tgl urun	Keterangan

Gambar V.36 Interface Form Pemeriksaan Awal Tab Info Produk Klaim
 Sumber: Hasil Analisis Data (2016)

Pemeriksaan Awal

Info Produk Klaim	Cek Tampilan Luar
Kode Produksi : <input type="text"/>	<input type="button" value="Pilih"/>
Penampilan Cover : <input type="text"/>	
Sealing : <input type="text"/>	
Terminal Pole : <input type="text"/>	
Elektrolit : <input type="text"/>	<input type="button" value="Pilih"/>
Stiker Alumunium : <input type="radio"/> Sudah dibuka <input type="radio"/> Belum dibuka	
Tegangan : <input type="text"/> Volt Cca : <input type="text"/> R : <input type="text"/>	
Cell 1 2 3 4 5 6	
SP Gr/20 <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> Rata-rata <input type="text"/>	
Voltase <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> Jumlah <input type="text"/>	
Masa Charge : Jam	
<input type="button" value="Simpan"/> <input type="button" value="Ubah"/> <input type="button" value="Batal"/>	

Data Produk Klaim						
Kode Produksi : <input type="text"/> <input type="button" value="Cari"/>						
Kd Produksi	No LPB	Id Produk	KM	Tgl Pakai	Tgl urun	Keterangan

Data Cek Tampilan						
Kode Produksi : <input type="text"/> <input type="button" value="Cari"/>						
Kd Produksi	Cover	Sealing	Terminal	Elektrolit	Charge	

Gambar V.37 Interface Form Pemeriksaan Awal Tab Cek Tampilan Luar
 Sumber: Hasil Analisis Data (2016)

9. *Form Pasca Charge dan Load Test*

Form pasca charge adalah *form* yang digunakan untuk menyimpan data SPGr dan voltase tiap *cell* baterai aki, sedangkan *form* kemampuan baterai adalah *form* yang digunakan untuk menyimpan hasil uji *load test* baterai dengan indikator warna. Rancangan *interface* dari *form* pasca charge dan *load test* dapat dilihat pada Gambar V.38 berikut:

Pemeriksaan Setelah Charge & Load Test

Form Pasca Charge

Tegangan Tertutup :

Tegangan Tertutup : Volt Cca :

Voltase Setelah Equalisasi : Volt R :

Cell 1 2 3 4 5 6

SP Gr/20 Rata-rata

Voltase Jumlah

Form Kemampuan Baterai

Catatan :
Parameter test berdasarkan BT Load Test (400 A untuk mobil dan 350 A untuk motor).

Parameter warna :
 Hijau Kuning Merah

Kesimpulan :

Data pemeriksaan

Kode Produksi :

Kd Produksi	SPGr Rata-rata	Jml volt	Parameter	Kesimpulan

SpGr (Specific Gravity) Volt (Tegangan)

Gambar V.38 Interface Form Pasca Charge dan Load Test
 Sumber: Hasil Analisis Data (2016)

10. *Form* Bedah Baterai

Form bedah baterai adalah *form* yang digunakan untuk menyimpan data bedah baterai aki. Rancangan *interface* dari *form* bedah baterai dapat dilihat pada Gambar V.39 berikut:

Bedah Baterai

Form Bedah Baterai

Kode Produksi :

Plate (+) :

Plate (-) :

Separator :

Endapan :

Kesimpulan Akhir :

Data Produk Klaim

Kode Produksi :

Kd Produksi	No LPB	Merk	Tipe	Cov	Seal	Term	Elk	P(+)	P(-)	Sptr	Endap	Kesimpulan

Kd Produksi	No LPB

Gambar V.39 Interface Form Bedah Baterai
 Sumber: Hasil Analisis Data (2016)

11. *Form Validasi*

Form validasi adalah *form* yang digunakan untuk menyimpan data validasi baterai yang telah melalui proses klaim. Rancangan *interface* dari *form* validasi dapat dilihat pada Gambar V.40 berikut:

Data Pemohon Klaim		
Ketik Nama Pemohon		Cari
No LPB	Nama Perusahaan	Jumlah

Data Produk Klaim		
Nama Perusahaan :		
Jumlah :		
Merk	Jenis	Tipe
1/5 < >		

Diketahui oleh :

Claim Handling Section Head

Tidak Divalidasi

Catatan

Simpan Batal

Form Validasi Klaim		
Petugas :		
Kode Produksi :	Tanggal Masuk :	
Merk :	Jenis :	
Tipe :	Kapasitas :	AH
Pemeriksaan Awal Produk		
Cover :		
Sealing :		
Terminal :		
Elektrolit :		
Stiker :	Lama Charge :	Jam
Tes Kemampuan Produk		
Tegangan Tertutup :		Tegangan Equalisasi :
SPGr Rata-rata :		Jumlah Voltase :
Indikator warna :		
Bedah baterai		
Plat Positif + :		
Plat Negatif - :		
Separator :		
Endapan :		
Hasil		
Kesimpulan :		

Gambar V.40 *Interface Form Validasi*
Sumber: Hasil Analisis Data (2016)

12. *Form Laporan*

Form laporan adalah *form* yang digunakan untuk melihat laporan penerimaan dan kerusakan produk. *Form* laporan terdiri dari dua jenis yaitu laporan penerimaan dan laporan kerusakan. Rancangan *interface* dari *form* laporan dapat dilihat pada Gambar V.41 dan Gambar V.42 berikut:

Gambar V.41 *Interface Form* Pilih Tanggal & Laporan Penerimaan Baterai
 Sumber: Hasil Analisis Data (2016)

Gambar V.42 *Interface Form* Pilih Tanggal & Laporan Kerusakan Baterai
 Sumber: Hasil Analisis Data (2016)

5.5 Implementasi Sistem *Software* dan *Hardware*

Tahap ini adalah tahap pengkodean program menggunakan bahasa pemrograman Java dengan Netbeans 8.0.2 sebagai aplikasi editor. Setiap *interface* berisikan kode program agar program dapat dijalankan sesuai dengan fungsinya. Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*). Adapun spesifikasinya sebagai berikut:

1. Analisis Kebutuhan *Software*

- a. Sistem Operasi : Minimal *Microsoft Windows 7*
- b. *Database Server* : MySQL versi 5.6.16
- c. Bahasa Pemrograman : J2SE (*Java 2 Standard Edition*)

2. Analisis Kebutuhan *Hardware*

- a. *Processor* : Minimal *Processor Core 2 Duo*
- b. RAM : Minimal RAM 512 MB
- c. *Harddisk* : Minimal *Harddisk 64 GB*
- d. *Mouse, Keyboard, Monitor* sebagai peralatan antar muka.
- e. *Printer* sebagai media pencetakan dokumen.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan hasil penelitian selama proses pengumpulan dan pengolahan data, analisis sistem, dan implementasi sistem usulan maka dapat diambil kesimpulan sebagai berikut:

1. Dengan dibangun sistem informasi pemeriksaan produk klaim baterai aki ini, proses pemeriksaan produk baterai dapat berjalan lebih mudah, akurat dan data klaim dapat tersimpan dengan aman di dalam basis data.
2. Aplikasi sistem informasi pemeriksaan produk klaim baterai aki ini memudahkan dalam mengelola data pelaporan klaim pada *Quality Assurance Section Head* secara otomatis dengan hanya memilih periode dan tipe kerusakan produk baterai.

6.2. Saran

Saran yang diberikan untuk pengembangan sistem informasi pemeriksaan produk klaim baterai aki ini selanjutnya adalah sebagai berikut:

1. Untuk penerapan sistem baru terhadap sistem lama, sebaiknya dilakukan secara bertahap dan diperlukan sosialisasi penggunaan sistem ini kepada bagian yang terkait.
2. Terus melakukan pengembangan pada aplikasi, sehingga aplikasi dapat terus diperbaharui dari segi fitur dan tampilan untuk meningkatkan kinerja kerja bagian terkait.

DAFTAR PUSTAKA

- Anhar. 2010. *PHP & MySQL Secara Otodidak*. Jakarta: PT Trans Media.
- Arifin, E. Zaenal dan Tasai, S. Amran. 2009. *Cermat Berbahasa Indonesia*. Cetakan Ke-11. Jakarta: Akademika Pressindo.
- Badudu, JS. 2003. *Kamus Kata-Kata Serapan Asing Dalam Bahasa Indonesia*. Cetakan Pertama. Jakarta: Kompas.
- Britton, Carol dan Doake, Jill. 2005. *A Student Guide To Object Oriented Development*. Inggris: Linacre House Jordan Hill.
- Daryanto. 2011. *Sistem Kelistrikan Motor*. Bandung: PT Sarana Tutorial Nurani Sejahtera.
- Djahir, Yulia dan Pratita, Dewi. 2014. *Bahan Ajar Sistem Informasi Manajemen*. Yogyakarta: Deepublish.
- Hutahaean, Japerson. 2014. *Konsep Sistem Informasi*. Yogyakarta: Deepublish.
- Ichwan, M. 2011. *Pemrograman Basis Data Delphi 7 dan MySQL*. Bandung: Informatika Bandung.
- Indrajani. 2015. *Database Design*. Jakarta: PT Elex Media Komputindo.
- Jogiyanto, HM. 2005. *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Edisi Ketiga. Yogyakarta: Andi Offset.
- Jubilee Enterprise. 2015. *Mengenal Java dan Database Dengan Netbeans*. Jakarta: PT Elex Media Komputindo.
- Kadir, Abdul. 2014. *Pengenalan Sistem Informasi*. Edisi Revisi. Yogyakarta: Andi Offset.
- McLeod, Raymond. dan P. Schell, George. 2011. *Sistem Informasi Manajemen*. Jakarta: PT. Indeks.
- Nugroho, Adi. 2009. *Rekayasa Perangkat Lunak Menggunakan UML dan Java*. Yogyakarta: Andi Offset.
- Pratama, I Putu Agus Eka. 2014. *Sistem Informasi dan Implementasinya*. Bandung: Informatika Bandung.

- Pusat Bahasa Departemen Pendidikan Nasional. *Kamus Besar Bahasa Indonesia*. Diambil dari situs resmi Badan Bahasa KEMDIKBUD: <http://www.kbbi.co.id/>. (Tanggal akses: 14 Juli 2016).
- Pusat Bahasa Departemen Pendidikan Nasional. 2005. *Pedoman umum Ejaan Bahasa Indonesia Yang Disempurnakan*. Cetakan Ke-28. Jakarta: Balai Pustaka.
- Rosa, A.S, dan Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Cetakan Kedua. Bandung: Modula.
- Royan, Frans M. 2011. *Strategi Mendirikan Perusahaan Distributor Baru*. Jakarta: PT Gramedia Pustaka Utama.
- Simarmata, Janner. 2010. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi Offset.
- Sitorus, Lamhot. 2015. *Algoritma dan Pemrograman*. Yogyakarta: Andi Offset.
- Sommerville, Ian. 2003. *Software Engineering Rekayasa Perangkat Lunak*. Jakarta: Erlangga.
- Sugiyono. 2013. *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta.
- Sutabri, Tata. 2004. *Analisa Sistem Informasi*. Yogyakarta: Andi Offset.
- Wahana Komputer. 2015. *Membangun Sistem Informasi Dengan Java Netbeans dan MySQL*. Yogyakarta: Andi Offset.
- Wahana Komputer. 2010. *Panduan Aplikatif dan Solusi (PAS) Membuat Aplikasi Client Server dengan Visual Basic 2008*. Andi offset, Yogyakarta.
- Wahana Komputer. 2010. *Pengembangan Aplikasi Database Berbasis Java DB Dengan Netbeans*. Yogyakarta: Andi Offset.
- Yakub. 2012. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Yanto, Robi. 2016. *Manajemen Basis Data Menggunakan MySQL*. Yogyakarta: Deepublish.

LAMPIRAN KODE PROGRAM

1. LPB

```
public View_LPB(String username) {
    initComponents();
    cl = new Ctr_LPB();
    makeCenter();
    tbl_LPB.setModel(cl.Daftar_LPB());
    aturKolom();
    txt_no_no.setDocument(new View_LPB.limitChar(5));
    txt_sj.setDocument(new View_LPB.limitChar(20));
    kd_otomatis();
    txt_penerima.setText(username);
    icon();
    tampilIdUser();
    Status(false);
}
private void btn_tambahActionPerformed(java.awt.event.ActionEvent evt) {
    if(txt_tanggal.getDate() == null){
        JOptionPane.showMessageDialog(null, "Tanggal harus diisi!");
    }else if(txt_dari.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Asal barang harus diketahui!");
    }else if(txt_sj.getText().equals("")){
        JOptionPane.showMessageDialog(null, "No surat jalan tidak boleh kosong!");
    }else if(txt_penyerah.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Nama penyerah harus ada!");
    }else{
        String no_bln = txt_no_bln.getText();
        String no_thn = txt_no_thn.getText();
        String no_no = txt_no_no.getText();
        Date tgl = txt_tanggal.getDate();
        String dari = txt_dari.getText();
        String sj = txt_sj.getText();
        String penyerah = txt_penyerah.getText();
        String penerima = txt_iduser.getText();
        cl.setNo_bln(no_bln);
        cl.setNo_thn(no_thn);
        cl.setNo_no(no_no);
        cl.setTanggal(tgl);
        cl.setDari(dari);
        cl.setSJ(sj);
        cl.setPenyerah(penyerah);
        cl.setPenerima(penerima);
        cl.Tambah();
    }
}
private void txt_tanggalPropertyChange(java.beans.PropertyChangeEvent evt) {
    if ("date".equals(evt.getPropertyName())) {
        Date d= (Date)evt.getNewValue();
        Calendar cal= Calendar.getInstance();
        cal.setTime(d);
        int month =cal.get(Calendar.MONTH)+1;
        int year =cal.get(Calendar.YEAR);
    }
}
```

```

String year2 = String.valueOf(year);
txt_no_bln.setText(String.valueOf(month));
txt_no_thn.setText(year2.substring(2, year2.length()));
switch (txt_no_bln.getText()) {
    case "1":txt_no_bln.setText("I");break;
    case "2":txt_no_bln.setText("II");break;
    case "3":txt_no_bln.setText("III");break;
    case "4":txt_no_bln.setText("IV");break;
    case "5":txt_no_bln.setText("V");break;
    case "6":txt_no_bln.setText("VI");break;
    case "7":txt_no_bln.setText("VII");break;
    case "8":txt_no_bln.setText("VIII");break;
    case "9":txt_no_bln.setText("IX");break;
    case "10":txt_no_bln.setText("X");break;
    case "11":txt_no_bln.setText("XI");break;
    case "12":txt_no_bln.setText("XII");break;
}
}
private void btn_ubahActionPerformed(java.awt.event.ActionEvent evt) {
    String no_bln = txt_no_bln.getText();
    String no_thn = txt_no_thn.getText();
    String no_no = txt_no_no.getText();
    Date tgl = txt_tanggal.getDate();
    String dari = txt_dari.getText();
    String sj = txt_sj.getText();
    String penyerah = txt_penyerah.getText();
    String penerima = txt_iduser.getText();
    cl.setNo_bln(no_bln);
    cl.setNo_thn(no_thn);
    cl.setNo_no(no_no);
    cl.setTanggal(tgl);
    cl.setDari(dari);
    cl.setSJ(sj);
    cl.setPenyerah(penyerah);
    cl.setPenerima(penerima);
    cl.Ubah();
}
private void tbl_LPBMousePressed(java.awt.event.MouseEvent evt) {
    Status(true);
    try {
        JTable tbl = (JTable) evt.getSource();
        int baris = tbl.getSelectedRow();
        String a = tbl.getModel().getValueAt(baris, 0).toString();
        a = a.substring(a.length()-5,a.length());
        txt_no_no.setText(a);
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
        Date dateUtil= sdf.parse((String)tbl.getModel().getValueAt(baris, 1));
        txt_tanggal.setDate(dateUtil);
        txt_dari.setText(tbl.getModel().getValueAt(baris, 2).toString());
        txt_sj.setText(tbl.getModel().getValueAt(baris, 3).toString());
    }
}

```

```

txt_penyerah.setText(tbl.getModel().getValueAt(baris, 4).toString());
String getPenerima = txt_penerima.getText();
String jabatan;
cl.setJabatan(getPenerima);
jabatan = cl.ambilJabatan();
if(jabatan.equals("Claim handling Section Head")||jabatan.equals("QA Section
Head")){
    refresh();
} else if(!getPenerima.equals(tbl.getModel().getValueAt(baris, 5).toString())){
    JOptionPane.showMessageDialog(null, "Anda tidak berhak merubah data ini");
    refresh();
    bersih();
    Status(false);}
} catch (ParseException ex) {}
}
private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) {
    String no_bln = txt_no_bln.getText();
    String no_thn = txt_no_thn.getText();
    String no_no = txt_no_no.getText();
    String penerima = txt_iduser.getText();
    cl.setNo_bln(no_bln);
    cl.setNo_thn(no_thn);
    cl.setNo_no(no_no);
    cl.setPenerima(penerima);
    cl.Hapus();
}
private void btn_pilihCusActionPerformed(java.awt.event.ActionEvent evt) {
    View_Pilih_Cus vpc = new View_Pilih_Cus(true);
    vpc.setVisible(true);
    if(vpc.getcustomer()!=null){
        txt_dari.setText(vpc.getcustomer().getDari());}
}
private void kd_otomatis() {
    try {
        String query = "select max(right(No_LPB,5)) as id from lpb";
        stm = Koneksi.getISNTANCE().connect(query);
        rs = stm.executeQuery();
        while(rs.next())
        {if(rs.first() == false)
            {txt_no_no.setText("00001");
            txt_no_no.setEditable(false);}
        else
            {rs.last();
            int auto_id = rs.getInt(1)+1;
            String no = String.valueOf(auto_id);
            int noLong = no.length();
            for(int a=0;a<5-noLong;a++)
            {no = "0" + no; }
            txt_no_no.setText(no);
            txt_no_no.setEditable(false);}
        }
    }
}

```

```

    }
    rs.close();
    stm.close();
  } catch (SQLException ex) {}
}}

```

2. LPB Controller

```

public DefaultTableModel CariLPB() {
    try {
        stm = Koneksi.getISNTANCE().connect(
            "SELECT No_LPB,\n"
            + "DATE_FORMAT(Tgl_Terima, '%d-%m-%Y') AS Tgl_Terima,\n"
            + "Dari,No_SJ,Penyerah,u>Nama_Lengkap FROM lpb AS l\n"
            + "LEFT JOIN `user` AS u ON l.Penerima = u.Id_User\n"
            + "WHERE EXISTS (SELECT No_LPB FROM lpb_detil AS ld WHERE
ld.No_LPB = l.No_LPB)\n"
            + "AND (No_LPB LIKE ? OR Dari LIKE ?)\n"
            + "ORDER BY No_LPB DESC");
        stm.setString(1, "%" + getCari() + "%");
        stm.setString(2, "%" + getCari() + "%");
        rs = stm.executeQuery();
        DefaultTableModel Judul_Tbl = new DefaultTableModel();
        Judul_Tbl.addColumn("No LPB");
        Judul_Tbl.addColumn("Tgl Terima");
        Judul_Tbl.addColumn("Dari");
        Judul_Tbl.addColumn("No Surat Jalan");
        Judul_Tbl.addColumn("Penyerah");
        Judul_Tbl.addColumn("Penerima");
        Object[] ok;
        while (rs.next()) {
            ok = new Object[6];
            ok[0] = rs.getString("No_LPB");
            ok[1] = rs.getString("Tgl_Terima");
            ok[2] = rs.getString("Dari");
            ok[3] = rs.getString("No_SJ");
            ok[4] = rs.getString("Penyerah");
            ok[5] = rs.getString("Nama_Lengkap");
            Judul_Tbl.addRow(ok);
        }return Judul_Tbl;
    } catch (SQLException ex) {}return null;
}

public TableModel Daftar_LPBB() {
    try {
        String query = "SELECT No_LPB,\n"
            + "DATE_FORMAT(Tgl_Terima, '%d-%m-%Y') AS Tgl_Terima,\n"
            + "Dari,No_SJ,Penyerah,u>Nama_Lengkap FROM lpb AS l\n"
            + "LEFT JOIN `user` AS u ON l.Penerima = u.Id_User\n"
            + "ORDER BY No_LPB DESC;";
        stm = Koneksi.getISNTANCE().connect(query);
        rs = stm.executeQuery();
    }
}

```

```

DefaultTableModel Judul_Tbl = new DefaultTableModel();
Judul_Tbl.addColumn("No LPB");
Judul_Tbl.addColumn("Tgl Terima");
Judul_Tbl.addColumn("Dari");
Judul_Tbl.addColumn("No Surat Jalan");
Judul_Tbl.addColumn("Penyerah");
Judul_Tbl.addColumn("Penerima");
Object[] ok;
while (rs.next()) {
    ok = new Object[6];
    ok[0] = rs.getString("No_LPB");
    ok[1] = rs.getString("Tgl_Terima");
    ok[2] = rs.getString("Dari");
    ok[3] = rs.getString("No_SJ");
    ok[4] = rs.getString("Penyerah");
    ok[5] = rs.getString("Nama_Lengkap");
    Judul_Tbl.addRow(ok);
}return Judul_Tbl;
} catch (SQLException ex) {}return null;
}
public void Tambah() {
    try {
        String query = "INSERT INTO lpb\n"
            + "      (No_LPB,Tgl_Terima,Dari,No_SJ,Penyerah,Penerima)\n"
            + "VALUES (?,?,?,?,?)";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, getNo_bln()+getNo_thn()+getNo_no());
        stm.setDate(2, new java.sql.Date(getTanggal().getTime()));
        stm.setString(3, getDari());
        stm.setString(4, getSJ());
        stm.setString(5, getPenyerah());
        stm.setString(6, getPenerima());
        stm.executeUpdate();
        JOptionPane.showMessageDialog(null, "LPB berhasil ditambahkan",
            "Berhasil", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {}
}
public void Ubah(){
    try {
        String query = "UPDATE lpb\n"
            + "SET No_LPB = ?,Tgl_Terima = ?,Dari = ?,No_SJ = ?,Penyerah = ?\n"
            + "WHERE No_LPB = ? AND Penerima = ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, getNo_bln()+getNo_thn()+getNo_no());
        stm.setDate(2, new java.sql.Date(getTanggal().getTime()));
        stm.setString(3, getDari());
        stm.setString(4, getSJ());
        stm.setString(5, getPenyerah());
        stm.setString(6, getNo_bln()+getNo_thn()+getNo_no());
        stm.setString(7, getPenerima());
    }
}

```

```

        stm.executeUpdate();
        JOptionPane.showMessageDialog(null, "Ubah Data LPB Berhasil",
            "Berhasil", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {}
}
public void Hapus(){
    try {
        String query = "DELETE\n"
            + "FROM LPB WHERE No_LPB = ? AND Penerima = ?;";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, getNo_bln()+getNo_thn()+getNo_no());
        stm.setString(2, getPenerima());
        stm.executeUpdate();
        JOptionPane.showMessageDialog(null, "Hapus Data LPB Berhasil",
            "Berhasil", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {}
}
public String tampil_iduser() {
    try {
        String query = "SELECT Id_User FROM `user` WHERE Nama_Lengkap = ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, getPenerima());
        rs = stm.executeQuery();
        String id = "";
        if(rs.next()){
            id = rs.getString("id_user");
        }return id;
    } catch (SQLException ex) {}return null;
}
public DefaultTableModel PilihLPB(String user) {
    try {
        stm = Koneksi.getISNTANCE().connect(
            "SELECT No_LPB, DATE_FORMAT(Tgl_Terima, '%d-%m-%Y') AS
Tgl_Terima,\n"
            + "Dari,No_SJ,Penyerah,u.Nama_Lengkap FROM lpb AS l\n"
            + "LEFT JOIN `user` AS u ON l.Penerima = u.Id_User\n"
            + "WHERE Nama_Lengkap LIKE ? ORDER BY No_LPB DESC");
        stm.setString(1, "%" + user+ "%");
        rs = stm.executeQuery();
        DefaultTableModel Judul_Tbl = new DefaultTableModel();
        Judul_Tbl.addColumn("No LPB");
        Judul_Tbl.addColumn("Tgl Terima");
        Judul_Tbl.addColumn("Dari");
        Judul_Tbl.addColumn("No Surat Jalan");
        Judul_Tbl.addColumn("Penyerah");
        Judul_Tbl.addColumn("Penerima");
        Object[] ok;
        while (rs.next()) {
            ok = new Object[6];
            ok[0] = rs.getString("No_LPB");

```

```

        ok[1] = rs.getString("Tgl_Terima");
        ok[2] = rs.getString("Dari");
        ok[3] = rs.getString("No_SJ");
        ok[4] = rs.getString("Penyerah");
        ok[5] = rs.getString("Nama_Lengkap");
        Judul_Tbl.addRow(ok);
    }return Judul_Tbl;
} catch (SQLException ex) {}return null;
}
}
public DefaultTableModel cariCus(){
    try {
        stm = Koneksi.getISNTANCE().connect(
            "SELECT Dari FROM lpb\n"
            + "WHERE Dari LIKE ?\n"
            + "GROUP BY Dari");
        stm.setString(1, "%" + getCari()+ "%");
        rs = stm.executeQuery();
        DefaultTableModel Judul_Tbl = new DefaultTableModel();
        Judul_Tbl.addColumn("Nama Customer");
        Object[] ok;
        while (rs.next()) {
            ok = new Object[1];
            ok[0] = rs.getString("Dari");
            Judul_Tbl.addRow(ok);}
        return Judul_Tbl;
    } catch (SQLException ex) {}
    return null;
}
}}

```

3. Pemeriksaan Awal

```

public class View_PeriksaAwal extends javax.swing.JInternalFrame {
    public View_PeriksaAwal(String user) {
        cpa = new Ctr_PeriksaAwal();
        tbl_ProdKlaim.setModel(cpa.Daftar_InfoProduk());
        aturKolomInfo();
        cpa2 = new Ctr_PeriksaAwal_Tab2();
        tbl_periksaAwal.setModel(cpa2.Daftar_PeriksaAwal());
        aturKolomInfo2();
        cval = new Ctr_Validasi();
        batasChar();
        Status("awal");
        Status("HitungChargeFalse");
    }
    private void btn_pilihProd_1ActionPerformed(java.awt.event.ActionEvent evt) {
        periksaAwal = new PeriksaAwal();
        user = "parameter_noLPB";
        nolpb = txt_nolpb_1.getText();
        String ambilLPB = nolpb;
        periksaAwal.setAmbilLPB(ambilLPB);
        View_Pilih_Produk vpp = new View_Pilih_Produk(user, true, nolpb);
    }
}

```

```

vpp.setVisible(true);
String ambilProd = (vpp.getProduk().getId_Produk());
periksaAwal.setAmbilprod(ambilProd);
String queryC = cpa.totalQtydetilPerMerk(periksaAwal);
String queryD = cpa.totalQtydataprodukPerMerk(periksaAwal);
if(queryC.equals(queryD)){
    JOptionPane.showMessageDialog(null, "Produk ini tidak bisa ditambah lagi !\n"+"Id
"+ambilProd+" sudah lengkap");
}
else{
    if(vpp.getProduk()!=null){
        txt_idpro_1.setText(vpp.getProduk().getId_Produk());
        lbl_tipe_1.setText(vpp.getProduk().getTipe());
        lbl_jenis_1.setText(vpp.getProduk().getJenis());}}
}
private void btn_pilihLPB_1ActionPerformed(java.awt.event.ActionEvent evt) {
    periksaAwal = new PeriksaAwal();
    user = "all";
    View_Pilih_LPB vpl = new View_Pilih_LPB(user, true);
    vpl.setVisible(true);
    if(vpl.getLpb()!=null){
        nolpb = vpl.getLpb().getNolpb();
        String ambilLPB = nolpb;
        periksaAwal.setAmbilLPB(ambilLPB);
        String queryA = cpa.totalQtydetil(periksaAwal);
        String queryB = cpa.totalQtydataproduk(periksaAwal);
        if(queryA.equals(queryB)){
            JOptionPane.showMessageDialog(null, "Produk telah LENGKAP !\n"+"Total
"+queryB+" Baterai");
        }
        else{
            txt_nolpb_1.setText(vpl.getLpb().getNolpb());
            lbl_noSJ_1.setText(vpl.getLpb().getSJ());
            SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
            String dateUtil=sdf.format(vpl.getLpb().getTanggal());
            lbl_tglterima_1.setText(dateUtil);
            Status("lpbSelected");}}
}
private void txt_kdpro_1KeyTyped(java.awt.event.KeyEvent evt) {
    boolean error=false;
    String errorMessage="";
    String kode = txt_kdpro_1.getText();
    if(txt_kdpro_1.getText().trim().length()==10){
        if(kode.length() == 10){
            String tahun = kode.substring(2, 4);
            String bulan = kode.substring(4, 5);
            Calendar calInput = Calendar.getInstance();
            if(isNumeric(tahun)){
                calInput.set(Calendar.YEAR, Integer.valueOf("20"+tahun));
            }
            else{
                errorMessage="Format kode salah!";
                error=true;}
        }
    }
}

```

```

if(convertMonth(bulan) != 0){
    calInput.set(Calendar.MONTH, convertMonth(bulan));
}else{
    errorMessage="Format kode salah!";
    error=true;}
if(error){
    lbl_kesimpulan_1.setText(errorMessage);return;
}
Calendar calNow = Calendar.getInstance();
int selisih = calNow.get(Calendar.YEAR) - calInput.get(Calendar.YEAR);
if(selisih > 2){
    errorMessage = "Produk Kadaluarsa";
    error=false;
}else {
    selisih = selisih * 12;
    int selisihBulan = (calNow.get(Calendar.MONTH)+2) -
calInput.get(Calendar.MONTH);
    int selisihTotal = selisih + selisihBulan;
    if(selisihTotal > 24){
        errorMessage = "Produk Kadaluarsa";
        error=false;
    }else{
        errorMessage = "Lanjut Klaim";
        error=false;}}
}else{
    errorMessage="Format kode salah!";
    error=true;}
lbl_kesimpulan_1.setText(errorMessage);
Status("kd_Notempty");
}else if(txt_kdpro_1.getText().trim().length() < 11){
    lbl_kesimpulan_1.setText("");
    Status("awal");}
}
private void tbl_ProdKlaimMousePressed(java.awt.event.MouseEvent evt) {
try {
    JTable tbl = (JTable) evt.getSource();
    int baris = tbl.getSelectedRow();
    txt_kdpro_1.setText(tbl.getModel().getValueAt(baris, 0).toString());
    txt_nolpb_1.setText(tbl.getModel().getValueAt(baris, 1).toString());
    txt_idpro_1.setText(tbl.getModel().getValueAt(baris, 2).toString());
    txt_km_1.setText(tbl.getModel().getValueAt(baris, 3).toString());
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
    Date dateUtil= sdf.parse((String)tbl.getModel().getValueAt(baris, 4));
    txt_tglPakai_1.setDate(dateUtil);
    Date dateUtil2= sdf.parse((String)tbl.getModel().getValueAt(baris, 5));
    txt_tglTurun_1.setDate(dateUtil2);
    lbl_kesimpulan_1.setText(tbl.getModel().getValueAt(baris, 6).toString());
} catch (ParseException ex) {}
periksaAwal = new PeriksaAwal();
String lpb = txt_nolpb_1.getText();

```

```

periksaAwal.setNolpb(lpb);
String SuratJalan = cpa.query_SuratJalan(periksaAwal);
lbl_noSJ_1.setText(SuratJalan);
String TglTerima = cpa.query_TglTerima(periksaAwal);
lbl_tglterima_1.setText(TglTerima);
String idProd = txt_idpro_1.getText();
periksaAwal.setIdprod(idProd);
String tipe = cpa.query_Tipe(periksaAwal);
lbl_tipe_1.setText(tipe);
String jenis = cpa.query_Jenis(periksaAwal);
lbl_jenis_1.setText(jenis);
Status("tblProdKlaimSelected");
}
private void btn_pilihElektrolit_2ActionPerformed(java.awt.event.ActionEvent evt) {
    user = "Elektrolit";
    View_Pilih_Kerusakan vpk = new View_Pilih_Kerusakan(user, true);
    vpk.setVisible(true);
    if(vpk.getKerusakan()!=null){
        user = vpk.getKerusakan().getNm_Kerusakan();
        String kesalahan = cpa2.cariKesalahan(user);
        txt_elektrolit_2.setText(vpk.getKerusakan().getNm_Kerusakan());
        if(kesalahan.equals("Customer"))
            status_4=Boolean.TRUE;
        else
            status_4=Boolean.FALSE;
        if(status_1 || status_2 || status_3 || status_4){
            lbl_keterangan_2.setText("Stop Klaim");
            Status("HitungChargeFalse");
            BersihHitungCas(); }
        else if(!status_1 && !status_2 && !status_3 && !status_4){
            lbl_keterangan_2.setText("Lanjut Klaim");
            Status("HitungChargeTrue");}
        user = vpk.getKerusakan().getNm_Kerusakan();}
    }
private void btn_prosesActionPerformed(java.awt.event.ActionEvent evt) {
    if(spgr1.getText().equals("")||volt1.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 1!");
    }else if(spgr2.getText().equals("")||volt2.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 2!");
    }else if(spgr3.getText().equals("")||volt3.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 3!");
    }else if(spgr4.getText().equals("")||volt4.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 4!");
    }else if(spgr5.getText().equals("")||volt5.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 5!");
    }else if(spgr6.getText().equals("")||volt6.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 6!");
    }else{
        DecimalFormat df = new DecimalFormat("##");
        float average; float sp1; float sp2; float sp3; float sp4; float sp5; float sp6;

```

```

sp1=Float.parseFloat(spgr1.getText());
sp2=Float.parseFloat(spgr2.getText());
sp3=Float.parseFloat(spgr3.getText());
sp4=Float.parseFloat(spgr4.getText());
sp5=Float.parseFloat(spgr5.getText());
sp6=Float.parseFloat(spgr6.getText());
average = (sp1+sp2+sp3+sp4+sp5+sp6)/6;
    if(average > 1280){
        average = 1280;
    }else if(average < 1140){
        average = 1140;}
    txt_average_2.setText(Float.valueOf(df.format(average)).toString());
float count; float vo1; float vo2; float vo3; float vo4; float vo5; float vo6;
vo1=Float.parseFloat(volt1.getText());
vo2=Float.parseFloat(volt2.getText());
vo3=Float.parseFloat(volt3.getText());
vo4=Float.parseFloat(volt4.getText());
vo5=Float.parseFloat(volt5.getText());
vo6=Float.parseFloat(volt6.getText());
    count = vo1+vo2+vo3+vo4+vo5+vo6;
    if(count > 12.8){
        count = (float) 12.8;
    }else if(count < 12){
        count = 12;}
    txt_jumlah_2.setText(Float.valueOf(df.format(count)).toString());
//Perhitungan losscharge
int levelCharge; float lossCharge;
levelCharge = (int) (((average-1140)/35)*12.5) + (((count-12)/0.2)*12.5));
lossCharge = (float) (100 - levelCharge)/100;
String getAH; int kapasitas; float arusCharge; int countlamaCharge; String
lamaCharge;
user = txt_kdpro_2.getText();
getAH = cpa2.getAmpereHour(user);
kapasitas = Integer.valueOf(getAH);
arusCharge = (float) (0.1*kapasitas);
countlamaCharge = (int) Math.ceil(((lossCharge*kapasitas*1.2)/arusCharge));
lamaCharge = String.valueOf(countlamaCharge);
lbl_lamaCharge.setText(lamaCharge);} }
}
public void ket_periksaAwal(String keterangan){
    try {
        stm = Koneksi.getISNTANCE().connect(
            "SELECT Kesalahan\n"
            + "FROM kerusakan\n"
            + "WHERE Nama_Kerusakan LIKE ?");
        stm.setString(1, keterangan);
        rs = stm.executeQuery();
        while (rs.next()){
            String lanjut = "Lanjut Klaim";
            String stop = "Stop Klaim";

```

```

        if(rs.getString("Kesalahan").equals("Pabrik")){
            lbl_keterangan_2.setText(lanjut);
        }else if(rs.getString("Kesalahan").equals("Customer")){
            lbl_keterangan_2.setText(stop);}
    } catch (SQLException ex) {}
}

```

4. Pemeriksaan Awal Controller

```

public class Ctr_PeriksaAwal {
    PreparedStatement stm;
    ResultSet rs;
    public void Tambah(PeriksaAwal modelPeriksaawal) {
        try {
            String query = "INSERT INTO info_produk_klaim\n"
                + "(Kd_Produksi,No_LPB,Id_Produk,Km,Tgl_Pakai,Tgl_Turun,Ket_Info)\n"
                + "VALUES (?,?,?,?,?,?);";
            stm = Koneksi.getISNTANCE().connect(query);
            stm.setString(1, modelPeriksaawal.getKd_prod());
            stm.setString(2, modelPeriksaawal.getNoLPB());
            stm.setString(3, modelPeriksaawal.getIdprod());
            stm.setString(4, modelPeriksaawal.getKm());
            stm.setDate(5, new java.sql.Date(modelPeriksaawal.getTglpakai().getTime()));
            stm.setDate(6, new java.sql.Date(modelPeriksaawal.getTglTurun().getTime()));
            stm.setString(7, modelPeriksaawal.getKeterangan());
            stm.executeUpdate();
            JOptionPane.showMessageDialog(null, "Tambah Info Produk Berhasil",
                "Berhasil", JOptionPane.INFORMATION_MESSAGE);
        } catch (SQLException ex) {}
    }
    public TableModel Daftar_InfoProduk() {
        try {
            String query = "SELECT Kd_Produksi,No_LPB,Id_Produk,Km,\n"
                + " DATE_FORMAT(Tgl_Pakai, '%d-%m-%Y') AS Tgl_Pakai,\n"
                + " DATE_FORMAT(Tgl_Turun, '%d-%m-%Y') AS Tgl_Turun,\n"
                + " Ket_Info FROM info_produk_klaim ORDER BY No_LPB DESC";
            stm = Koneksi.getISNTANCE().connect(query);
            rs = stm.executeQuery();
            DefaultTableModel Judul_Tbl = new DefaultTableModel();
            Judul_Tbl.addColumn("Kd Produksi");
            Judul_Tbl.addColumn("No LPB");
            Judul_Tbl.addColumn("Id Prod");
            Judul_Tbl.addColumn("Km");
            Judul_Tbl.addColumn("Tgl Pakai");
            Judul_Tbl.addColumn("Tgl Turun");
            Judul_Tbl.addColumn("Keterangan");
            Object[] ok;
            while (rs.next()) {
                ok = new Object[7];
                ok[0] = rs.getString("Kd_Produksi");
                ok[1] = rs.getString("No_LPB");

```

```

        ok[2] = rs.getString("Id_Produk");
        ok[3] = rs.getString("Km");
        ok[4] = rs.getString("Tgl_Pakai");
        ok[5] = rs.getString("Tgl_Turun");
        ok[6] = rs.getString("Ket_Info");
        Judul_Tbl.addRow(ok);}
    return Judul_Tbl;
} catch (SQLException ex) {}
return null;
}
public void Hapus(PeriksaAwal modelPeriksaawal){
    try {
        String query = "DELETE FROM info_produk_klaim WHERE Kd_Produksi = ?;";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelPeriksaawal.getKd_prod());
        stm.executeUpdate();
        JOptionPane.showMessageDialog(null, "Hapus Info Produk Berhasil",
            "Berhasil", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {}
}
public void Ubah(PeriksaAwal modelPeriksaawal){
    try {
        String query = "UPDATE info_produk_klaim SET No_LPB = ?,\n"
            + " Id_Produk = ?,Km = ?,Tgl_Pakai = ?,Tgl_Turun = ?\n"
            + "WHERE Kd_Produksi = ?;";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelPeriksaawal.getNolpb());
        stm.setString(2, modelPeriksaawal.getIdprod());
        stm.setString(3, modelPeriksaawal.getKm());
        stm.setDate(4, new java.sql.Date(modelPeriksaawal.getTglpakai().getTime()));
        stm.setDate(5, new java.sql.Date(modelPeriksaawal.getTglturun().getTime()));
        stm.setString(6, modelPeriksaawal.getKd_prod());
        stm.executeUpdate();
        JOptionPane.showMessageDialog(null, "Ubah Info Produk Berhasil",
            "Berhasil", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {}
}
public String query_SuratJalan(PeriksaAwal modelPeriksaawal) {
    try {
        String query = "SELECT No_SJ FROM lpb WHERE No_LPB LIKE ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelPeriksaawal.getNolpb());
        rs = stm.executeQuery();
        String sj = "";
        if(rs.next()){
            sj = rs.getString("No_SJ");}
        return sj;
    } catch (SQLException ex) {}
    return null;
}
}

```

```

public String query_Jenis(PeriksaAwal modelPeriksaawal) {
    try {
        String query = "SELECT Jenis FROM produk WHERE Id_Produk LIKE ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelPeriksaawal.getIdprod());
        rs = stm.executeQuery();
        String jenis = "";
        if(rs.next()){
            jenis = rs.getString("Jenis");}
        return jenis;
    } catch (SQLException ex) {}
    return null;
}
public String totalQtydetil(PeriksaAwal modelPeriksaawal){
    try {
        String query = "SELECT SUM(Qty) FROM lpb_detil WHERE No_LPB LIKE ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelPeriksaawal.getAmbilLPB());
        rs = stm.executeQuery();
        String total = "";
        if(rs.next()){
            total = rs.getString("SUM(Qty)");}
        return total;
    } catch (SQLException ex) {}
    System.out.println("gagal");
    return null;}
}

```

5. Load Test

```

public class View_Load extends javax.swing.JInternalFrame {
    public View_Load() {
        initComponents();
        cload = new Ctr_Load();
        tbl_dataPeriksa.setModel(cload.Daftar_Loadtes());
        makeCenter();
        icon();
        batasChar();
        status("awal");
        status("cekcellfalse");
        kd_otomatis_datacell();
    }
    private void tampilgrafikspgr(){
        CategoryDataset dataset = createDataspgr();
        JFreeChart chart = createChart(dataset);
        pan_spgr.setLayout(new java.awt.BorderLayout());
        ChartPanel CP = new ChartPanel(chart);
        CP.setPreferredSize(new java.awt.Dimension(500, 222));
        pan_spgr.add(CP, BorderLayout.CENTER);
        pan_spgr.validate();
    }
}

```

```

private CategoryDataset createDatasetspgr() {
    loadtes = new LoadTes();
    String kdpro = kdproduksi;
    loadtes.setKdproduksi(kdpro);
    int s1;int s2;int s3;int s4;int s5;int s6;
    int sp1 = Integer.valueOf(cload.ambilsp1(loadtes));
    int sp2 = Integer.valueOf(cload.ambilsp2(loadtes));
    int sp3 = Integer.valueOf(cload.ambilsp3(loadtes));
    int sp4 = Integer.valueOf(cload.ambilsp4(loadtes));
    int sp5 = Integer.valueOf(cload.ambilsp5(loadtes));
    int sp6 = Integer.valueOf(cload.ambilsp6(loadtes));

    if(spgr1.getText().equals("")||spgr2.getText().equals("")||spgr3.getText().equals("")||spgr4.get
Text().equals("")||spgr5.getText().equals("")||spgr6.getText().equals("")){
        s1 = 0;s2 = 0;s3 = 0; s4 = 0;s5 = 0;s6 = 0;
    }else{
        s1 = Integer.valueOf(spgr1.getText());
        s2 = Integer.valueOf(spgr2.getText());
        s3 = Integer.valueOf(spgr3.getText());
        s4 = Integer.valueOf(spgr4.getText());
        s5 = Integer.valueOf(spgr5.getText());
        s6 = Integer.valueOf(spgr6.getText());}
    String series1 = "Periksa Pertama";
    String series3 = "Periksa Kedua";
    String category1 = "Cell 1";String category2 = "Cell 2";String category3 = "Cell 3";
    String category4 = "Cell 4";String category5 = "Cell 5";String category6 = "Cell 6";
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    dataset.addValue(sp1, series1, category1);
    dataset.addValue(sp2, series1, category2);
    dataset.addValue(sp3, series1, category3);
    dataset.addValue(sp4, series1, category4);
    dataset.addValue(sp5, series1, category5);
    dataset.addValue(sp6, series1, category6);
    dataset.addValue(s1, series3, category1);
    dataset.addValue(s2, series3, category2);
    dataset.addValue(s3, series3, category3);
    dataset.addValue(s4, series3, category4);
    dataset.addValue(s5, series3, category5);
    dataset.addValue(s6, series3, category6);
    return dataset;
}

private JFreeChart createChart(CategoryDataset dataset) {
    JFreeChart chart = ChartFactory.createBarChart(
        "", //Judul
        "", // Sumbu Y
        "Nilai", // Sumbu X
        dataset, // data
        PlotOrientation.VERTICAL, // Orientasi
        true, // legend
        true, // tooltips

```

```

true // URLs);
chart.setBackgroundPaint(Color.white);
CategoryPlot plot = (CategoryPlot) chart.getPlot();
plot.setBackgroundPaint(Color.WHITE);//warna Background
plot.setDomainGridlinePaint(Color.BLACK);//warna Grid
plot.setDomainGridlinesVisible(true);
plot.setRangeGridlinePaint(Color.BLACK);
final NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
BarRenderer renderer = (BarRenderer) plot.getRenderer();
renderer.setDrawBarOutline(false);
GradientPaint gp0 = new GradientPaint(0.0f, 0.0f, Color.CYAN,
0.0f, 0.0f, new Color(0, 0, 64));//warna untuk series 1
GradientPaint gp2 = new GradientPaint(0.0f, 0.0f, Color.PINK,
0.0f, 0.0f, new Color(64, 0, 0));//warna untuk series3
renderer.setSeriesPaint(0, gp0);
renderer.setSeriesPaint(2, gp2);
return chart;
}
private void btn_pilihActionPerformed(java.awt.event.ActionEvent evt) {
    user = "load tes";
    View_Pilih_KdProd vpkp = new View_Pilih_KdProd(user, null, true);
    vpkp.setVisible(true);
    if(vpkp.getperiksaAwal()!=null){
        txt_kdproduksi.setText(vpkp.getperiksaAwal().getKd_prod());
        status("pilihprod");
        status("cekcelltrue");
        this.kdproduksi = (vpkp.getperiksaAwal().getKd_prod());
        bersih();
        tampilgrafikspgr();
        tampilgrafikvolt();
        ambilKdPeriksa();
        nilai = "pilihKdTrue";}
}
private void btn_tambahActionPerformed(java.awt.event.ActionEvent evt) {
    if(txt_kdproduksi.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Kode produksi harus terisi!");
    }else if(txt_sprata.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa cell terlebih dahulu!");
    }else if(gr_warna.isSelected(null)){
        JOptionPane.showMessageDialog(null, "Periksa kemampuan baterai terlebih
dahulu!");
    }else{
        String kdprod = txt_kdproduksi.getText();
        String periksa = kdperiksa;
        String kddatacell = kd_datacell;
        String tegtutup = txt_volTertutup.getText();
        String tegequal = txt_volEqual.getText();
        String cca = txt_cca.getText();
        String R = txt_R.getText();

```

```

String sp1 = spgr1.getText();String sp2 = spgr2.getText();String sp3 = spgr3.getText();
String sp4 = spgr4.getText();String sp5 = spgr5.getText();String sp6 = spgr6.getText();
String v1 = vol1.getText();String v2 = vol2.getText();String v3 = vol3.getText();
String v4 = vol4.getText();String v5 = vol5.getText();String v6 = vol6.getText();
String sprata = txt_sprata.getText();
String vjml = txt_voljmlh.getText();
String warna = null;
    if(rb_hijau.isSelected()){
        warna = "Hijau";
    }else if(rb_kuning.isSelected()){
        warna = "Kuning";
    }else if(rb_merah.isSelected()){
        warna = "Merah";}
String kesimpulan = lbl_kesimpulan.getText();
loadtes = new LoadTes();
loadtes.setKdproduksi(kdprod);
loadtes.setTegTutup(tegtutup);
loadtes.setTegEqual(tegequal);
loadtes.setWarna(warna);
loadtes.setKesimpulan(kesimpulan);
datacell = new DataCell();
datacell.setKdDataCell(kddatacell);
datacell.setKdPeriksa(periksa);
datacell.setCca(cca);
datacell.setR(R);
datacell.setSpgr1(sp1);datacell.setSpgr2(sp2);datacell.setSpgr3(sp3);
datacell.setSpgr4(sp4);datacell.setSpgr5(sp5);datacell.setSpgr6(sp6);
datacell.setVolt1(v1);datacell.setVolt2(v2);datacell.setVolt3(v3);
datacell.setVolt4(v4);datacell.setVolt5(v5);datacell.setVolt6(v6);
datacell.setSpRatarata(sprata);
datacell.setVoltJml(vjml);
cload.Tambah(datacell, loadtes);}
}
private void btn_prosesActionPerformed(java.awt.event.ActionEvent evt) {
    if(spgr1.getText().equals("")||vol1.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 1!");
    }else if(spgr2.getText().equals("")||vol2.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 2!");
    }else if(spgr3.getText().equals("")||vol3.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 3!");
    }else if(spgr4.getText().equals("")||vol4.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 4!");
    }else if(spgr5.getText().equals("")||vol5.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 5!");
    }else if(spgr6.getText().equals("")||vol6.getText().equals("")){
        JOptionPane.showMessageDialog(null, "Periksa kembali kolom cell 6!");
    }else{
        DecimalFormat df = new DecimalFormat("###");
        float average; float sp1; float sp2; float sp3; float sp4; float sp5; float sp6;
        sp1=Float.parseFloat(spgr1.getText());

```

```

sp2=Float.parseFloat(spgr2.getText());
sp3=Float.parseFloat(spgr3.getText());
sp4=Float.parseFloat(spgr4.getText());
sp5=Float.parseFloat(spgr5.getText());
sp6=Float.parseFloat(spgr6.getText());
average = (sp1+sp2+sp3+sp4+sp5+sp6)/6;
    if(average > 1280){
        average = 1280;
    }else if(average < 1140){
        average = 1140;
    }
    txt_sprata.setText(Float.valueOf(df.format(average)).toString());
float count; float vo1; float vo2; float vo3; float vo4; float vo5; float vo6;
vo1=Float.parseFloat(vol1.getText());
vo2=Float.parseFloat(vol2.getText());
vo3=Float.parseFloat(vol3.getText());
vo4=Float.parseFloat(vol4.getText());
vo5=Float.parseFloat(vol5.getText());
vo6=Float.parseFloat(vol6.getText());
count = vo1+vo2+vo3+vo4+vo5+vo6;
    if(count > 12.8){
        count = (float) 12.8;
    }else if(count < 12){
        count = 12;}
    txt_voljmlh.setText(Float.valueOf(df.format(count)).toString());
tampilgrafikspgr();
tampilgrafikvolt();
status("prosesklik");}
}
private void rb_kuningMouseClicked(java.awt.event.MouseEvent evt) {
    lbl_kesimpulan.setText("Cas Kembali, amati beberapa faktor");
    lbl_link.setEnabled(true);
    lbl_link.setText(">>Link<<");
    btn_tambah.setEnabled(false);
    btn_ubah.setEnabled(false);
}
private void rb_merahMouseClicked(java.awt.event.MouseEvent evt) {
    lbl_kesimpulan.setText("Lanjut Klaim");
    lbl_link.setEnabled(false);
    lbl_link.setText("");
    if(nilai.equals("tableTrue")){
        btn_tambah.setEnabled(false);
        btn_ubah.setEnabled(true);
    }else{
        btn_tambah.setEnabled(true);
        btn_ubah.setEnabled(false);}
}
private void lbl_linkMouseClicked(java.awt.event.MouseEvent evt) {
    if(lbl_link.getText().equals(">>Link<<")){
        View_Faktor vf = new View_Faktor(null, true);

```

```

        vf.setVisible(true);}
    }
    private void tbl_dataPeriksaMouseClicked(java.awt.event.MouseEvent evt) {
        JTable tbl = (JTable) evt.getSource();
        int baris = tbl.getSelectedRow();
        txt_kdproduksi.setText(tbl.getModel().getValueAt(baris, 0).toString());
        gr_warna.getSelection();
        if("Hijau".equals(tbl.getModel().getValueAt(baris, 2).toString())){
            status("prosesklik");
            rb_hijau.setSelected(true);
        }else if("Kuning".equals(tbl.getModel().getValueAt(baris, 2).toString())){
            status("prosesklik");
            rb_kuning.setSelected(true);
        }else if("Merah".equals(tbl.getModel().getValueAt(baris, 2).toString())){
            status("prosesklik");
            rb_merah.setSelected(true);
        }
        lbl_kesimpulan.setText(tbl.getModel().getValueAt(baris, 3).toString());
        loadtes = new LoadTes();
        this.kdproduksi = tbl.getModel().getValueAt(baris, 0).toString();
        loadtes.setKdproduksi(kdproduksi);
        txt_volTertutup.setText(cload.ambilTegTutup(loadtes));
        txt_volEqual.setText(cload.ambilTegEqual(loadtes));
        txt_cca.setText(cload.ambilcca_2nd(loadtes));
        txt_R.setText(cload.ambilR_2nd(loadtes));
        spgr1.setText(cload.ambilsp1_2nd(loadtes));spgr2.setText(cload.ambilsp2_2nd(loadtes));
        spgr3.setText(cload.ambilsp3_2nd(loadtes));spgr4.setText(cload.ambilsp4_2nd(loadtes));
        spgr5.setText(cload.ambilsp5_2nd(loadtes));spgr6.setText(cload.ambilsp6_2nd(loadtes));
        vol1.setText(cload.ambilv1_2nd(loadtes));vol2.setText(cload.ambilv2_2nd(loadtes));
        vol3.setText(cload.ambilv3_2nd(loadtes));vol4.setText(cload.ambilv4_2nd(loadtes));
        vol5.setText(cload.ambilv5_2nd(loadtes));vol6.setText(cload.ambilv6_2nd(loadtes));
        txt_sprata.setText(cload.ambilsprata2nd(loadtes));
        txt_voljmlh.setText(cload.ambilvJml2nd(loadtes));
        nilai = "tableTrue";
        status("tblklik");
        tampilgrafikspgr();
        tampilgrafikvolt();
    }
    private void btn_ubahActionPerformed(java.awt.event.ActionEvent evt) {
        String kdprod = txt_kdproduksi.getText();
        String kdCell = cload.getKdDataCell(loadtes);
        String tegtutup = txt_volTertutup.getText();
        String tegequal = txt_volEqual.getText();
        String cca = txt_cca.getText();
        String R = txt_R.getText();
        String sp1 = spgr1.getText();String sp2 = spgr2.getText();String sp3 = spgr3.getText();
        String sp4 = spgr4.getText();String sp5 = spgr5.getText();String sp6 = spgr6.getText();
        String v1 = vol1.getText();String v2 = vol2.getText();String v3 = vol3.getText();
        String v4 = vol4.getText();String v5 = vol5.getText();String v6 = vol6.getText();
        String sprata = txt_sprata.getText();
        String vjml = txt_voljmlh.getText();
    }

```

```

String warna = null;
    if(rb_hijau.isSelected()){
        warna = "Hijau";
    }else if(rb_kuning.isSelected()){
        warna = "Kuning";
    }else if(rb_merah.isSelected()){
        warna = "Merah";}
String kesimpulan = lbl_kesimpulan.getText();
loadtes = new LoadTes();
loadtes.setKdproduksi(kdprod);
loadtes.setTegTutup(tegtutup);
loadtes.setTegEqual(tegequal);
loadtes.setWarna(warna);
loadtes.setKesimpulan(kesimpulan);
datacell = new DataCell();
datacell.setKdDataCell(kdCell);
datacell.setCca(cca);
datacell.setR(R);
datacell.setSpgr1(sp1);datacell.setSpgr2(sp2);datacell.setSpgr3(sp3);
datacell.setSpgr4(sp4);datacell.setSpgr5(sp5);datacell.setSpgr6(sp6);
datacell.setVolt1(v1);datacell.setVolt2(v2);datacell.setVolt3(v3);
datacell.setVolt4(v4);datacell.setVolt5(v5);datacell.setVolt6(v6);
datacell.setSpRatarata(sprata);
datacell.setVoltJml(vjml);
cload.ubah(loadtes, datacell);}
}

```

6. Load Test Controller

```

public class Ctr_Load {
    PreparedStatement stm;
    ResultSet rs;
    public void Tambah(DataCell modelDataCell, LoadTes modelLoad) {
        try {
            String query2 = "INSERT INTO data_cell\n"
                + "(Kd_DataCell,Kd_Pemeriksaan,Cca,R,\n"
                + "Spgr1, Volt1,Spgr2, Volt2,\n"
                + "Spgr3, Volt3,Spgr4, Volt4,\n"
                + "Spgr5, Volt5,Spgr6, Volt6,\n"
                + "Spgr_Ratarata,Volt_Jumlah,Status_Tes)\n"
                + "VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?, 'Sesudah)";
            stm = Koneksi.getINSTANCE().connect(query2);
            stm.setString(1, modelDataCell.getKdDataCell());
            stm.setString(2, modelDataCell.getKdPeriksa());
            stm.setString(3, modelDataCell.getCca());
            stm.setString(4, modelDataCell.getR());
            stm.setString(5, modelDataCell.getSpgr1());
            stm.setString(6, modelDataCell.getVolt1());
            stm.setString(7, modelDataCell.getSpgr2());
            stm.setString(8, modelDataCell.getVolt2());
            stm.setString(9, modelDataCell.getSpgr3());

```

```

stm.setString(10, modelDataCell.getVolt3());
stm.setString(11, modelDataCell.getSpgr4());
stm.setString(12, modelDataCell.getVolt4());
stm.setString(13, modelDataCell.getSpgr5());
stm.setString(14, modelDataCell.getVolt5());
stm.setString(15, modelDataCell.getSpgr6());
stm.setString(16, modelDataCell.getVolt6());
stm.setString(17, modelDataCell.getSpRatarata());
stm.setString(18, modelDataCell.getVoltJml());
stm.executeUpdate();
String query = "INSERT INTO load_test\n"
    + "(Kd_Produksi,Kd_DataCell,V_Tertutup,V_Equalisasi,\n"
    + "Tes_Kemampuan,Kesimpulan) VALUES (?, ?, ?, ?, ?)";
stm = Koneksi.getISNTANCE().connect(query);
stm.setString(1, modelLoad.getKdproduksi());
stm.setString(2, modelDataCell.getKdDataCell());
stm.setString(3, modelLoad.getTegTutup());
stm.setString(4, modelLoad.getTegEqual());
stm.setString(5, modelLoad.getWarna());
stm.setString(6, modelLoad.getKesimpulan());
stm.executeUpdate();
JOptionPane.showMessageDialog(null, "Tambah data pemeriksaan Berhasil",
    "Berhasil", JOptionPane.INFORMATION_MESSAGE);
} catch (SQLException ex) {}
}

public void ubah(LoadTes modelLoad, DataCell modelDataCell){
    try {
        String query = "UPDATE load_test SET V_Tertutup = ?,\n"
            + " V_Equalisasi = ?,Tes_Kemampuan = ?,Kesimpulan = ?\n"
            + "WHERE Kd_Produksi = ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelLoad.getTegTutup());
        stm.setString(2, modelLoad.getTegEqual());
        stm.setString(3, modelLoad.getWarna());
        stm.setString(4, modelLoad.getKesimpulan());
        stm.setString(5, modelLoad.getKdproduksi());
        stm.executeUpdate();
        String query2 = "UPDATE data_cell\n"
            + "SET Cca = ?,\n"
            + " R = ?,\n"
            + " Spgr1 = ?, Volt1 = ?,Spgr2 = ?, Volt2 = ?,\n"
            + " Spgr3 = ?, Volt3 = ?,Spgr4 = ?, Volt4 = ?,\n"
            + " Spgr5 = ?, Volt5 = ?,Spgr6 = ?, Volt6 = ?,\n"
            + " Spgr_Ratarata = ?,Volt_Jumlah = ?\n"
            + "WHERE Kd_DataCell = ? AND Status_Tes = 'Sesudah';";
        stm = Koneksi.getISNTANCE().connect(query2);
        stm.setString(1, modelDataCell.getCca());
        stm.setString(2, modelDataCell.getR());
        stm.setString(3, modelDataCell.getSpgr1());
        stm.setString(4, modelDataCell.getVolt1());
    }
}

```

```

stm.setString(5, modelDataCell.getSpgr2());
stm.setString(6, modelDataCell.getVolt2());
stm.setString(7, modelDataCell.getSpgr3());
stm.setString(8, modelDataCell.getVolt3());
stm.setString(9, modelDataCell.getSpgr4());
stm.setString(10, modelDataCell.getVolt4());
stm.setString(11, modelDataCell.getSpgr5());
stm.setString(12, modelDataCell.getVolt5());
stm.setString(13, modelDataCell.getSpgr6());
stm.setString(14, modelDataCell.getVolt6());
stm.setString(15, modelDataCell.getSpRatarata());
stm.setString(16, modelDataCell.getVoltJml());
stm.setString(17, modelDataCell.getKdDataCell());
stm.executeUpdate();
if(modelLoad.getWarna().equals("Hijau")){
    String query3 = "DELETE FROM bedah_baterai\n"
        + "WHERE EXISTS(SELECT Kd_Produksi\n"
        + "FROM load_test WHERE Kd_Produksi = ?)\n"
        + "AND Kd_Produksi = ?";
    stm = Koneksi.getISNTANCE().connect(query3);
    stm.setString(1, modelLoad.getKdproduksi());
    stm.setString(2, modelLoad.getKdproduksi());
    stm.executeUpdate();
}
JOptionPane.showMessageDialog(null, "Ubah Data Load Tes Berhasil",
    "Berhasil", JOptionPane.INFORMATION_MESSAGE);
} catch (SQLException ex) {}
}
public TableModel Daftar_Loadtes() {
    try {
        String query = "SELECT lt.Kd_Produksi, ip.No_LPB, \n"
            + "lt.Tes_Kemampuan AS Indikator, lt.Kesimpulan\n"
            + "FROM load_test AS lt LEFT JOIN info_produk_klaim AS ip\n"
            + "ON lt.Kd_Produksi = ip.Kd_Produksi ORDER BY No_LPB DESC";
        stm = Koneksi.getISNTANCE().connect(query);
        rs = stm.executeQuery();
        DefaultTableModel Judul_Tbl = new DefaultTableModel();
        Judul_Tbl.addColumn("Kd Produksi");
        Judul_Tbl.addColumn("No LPB");
        Judul_Tbl.addColumn("Indikator");
        Judul_Tbl.addColumn("Kesimpulan");
        Object[] ok;
        while (rs.next()) {
            ok = new Object[4];
            ok[0] = rs.getString("Kd_Produksi");
            ok[1] = rs.getString("No_LPB");
            ok[2] = rs.getString("Indikator");
            ok[3] = rs.getString("Kesimpulan");
            Judul_Tbl.addRow(ok);}
        return Judul_Tbl;
    }
}

```

```

    } catch (SQLException ex) {}
    return null;}
}

```

7. Laporan Penerimaan Viewer

```

public class View_Lap_Penerimaan extends javax.swing.JInternalFrame {
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
    public View_Lap_Penerimaan() {
        initComponents();
        makeCenter();
        icon();
    }
    private void makeCenter(){
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = this.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;}
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;}
        this.setLocation((screenSize.width - frameSize.width) / 2,
            (screenSize.height - frameSize.height) / 3);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        if(tgl_dari.getDateFormatString().trim().length() < 1){
            JOptionPane.showMessageDialog(null, "Tanggal Dari harus diisi!");
        }else if(tgl_sampai.getDateFormatString().trim().length() < 1){
            JOptionPane.showMessageDialog(null, "Tanggal Sampai harus diisi!");
        }else{
            try {
                InputStream inputStream =
View_Lap_Penerimaan.class.getResourceAsStream("/laporan/Laporan_Penerimaan.jasper");
                Ctr_Laporan laporan = new Ctr_Laporan();
                List<Laporan> list=laporan.laporanPenerimaan(sdf.format(tgl_dari.getDate()),
                    sdf.format(tgl_sampai.getDate()));
                JRDataSource dataSource = new JRBeanCollectionDataSource(list,false);
                Map<String, Object> map = new HashMap<>();
                map.put("tanggal", new Date());
                JasperPrint jasperPrint = JasperFillManager.fillReport(inputStream, map,
dataSource);
                Dialog_Laporan dialogpenerimaan = new Dialog_Laporan(jasperPrint);
                dialogpenerimaan.setVisible(true);
            } catch (JRException ex) {}
        }
    }
    private void icon(){
        ImageIcon icon = new ImageIcon(getClass().getResource("/gambar/lapTerima.png"));
        setFrameIcon(icon);}
}

```

8. Validasi Controller

```

public class Ctr_Validasi {

```

```

PreparedStatement stm;
ResultSet rs;
public void Tambah(Validasi modelValidasi) {
    try {
        String query = "INSERT INTO validasi\n"
            + "(Kd_Val,Kd_Produksi,Validasi) VALUES (?,?,'Menunggu Validasi)";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelValidasi.getKdval());
        stm.setString(2, modelValidasi.getKdprod());
        stm.executeUpdate();
    } catch (SQLException ex) {}
}
public void TambahAliasUbah(Validasi modelValidasi) {
    try {
        String query = "UPDATE validasi SET Validasi = ?,\n"
            + " Catatan = ? WHERE Kd_Produksi = ?";
        stm = Koneksi.getISNTANCE().connect(query);
        stm.setString(1, modelValidasi.getValidasi());
        stm.setString(2, modelValidasi.getCatatan());
        stm.setString(3, modelValidasi.getKdprod());
        stm.executeUpdate();
        JOptionPane.showMessageDialog(null, "Status Validasi Produk Telah Diubah!",
            "Berhasil", JOptionPane.INFORMATION_MESSAGE);
    } catch (SQLException ex) {}
}
public TableModel TabelPemohon() {
    try {
        String query = "SELECT Id.No_LPB, l.Dari, l.Tgl_Terima, SUM(l.D.Qty) AS qty
FROM lpb_detil AS ld LEFT JOIN lpb AS l ON ld.No_LPB = l.No_LPB GROUP BY
ld.No_LPB ORDER BY SUBSTRING(ld.No_LPB,7) DESC";
        stm = Koneksi.getISNTANCE().connect(query);
        rs = stm.executeQuery();
        DefaultTableModel Judul_Tbl = new DefaultTableModel();
        Judul_Tbl.addColumn("No LPB");
        Judul_Tbl.addColumn("Customer");
        Judul_Tbl.addColumn("Tgl Terima");
        Judul_Tbl.addColumn("Qty");
        Object[] ok;
        while (rs.next()) {
            ok = new Object[4];
            ok[0] = rs.getString("No_LPB");
            ok[1] = rs.getString("Dari");
            ok[2] = rs.getString("Tgl_Terima");
            ok[3] = rs.getString("qty");
            Judul_Tbl.addRow(ok);}
        return Judul_Tbl;
    } catch (SQLException ex) {}
    return null;
}
public TableModel tabelDataProduksi(String user, int start, int result){

```

```

try {
    String query = "SELECT ip.Kd_Produksi, p.Merk, p.Jenis, p.Tipe\n"
        + "FROM info_produk_klaim AS ip LEFT JOIN produk AS p\n"
        + "ON ip.Id_Produk = p.Id_Produk WHERE ip.No_LPB LIKE ?"
        + "LIMIT "+start+", "+result;
    stm = Koneksi.getInstANCE().connect(query);
    stm.setString(1, user);
    rs = stm.executeQuery();
    DefaultTableModel Judul_Tbl = new DefaultTableModel();
    Judul_Tbl.addColumn("Kd Produksi");
    Judul_Tbl.addColumn("Merk");
    Judul_Tbl.addColumn("Jenis");
    Judul_Tbl.addColumn("Tipe");
    Object[] ok;
    while (rs.next()) {
        ok = new Object[4];
        ok[0] = rs.getString("Kd_Produksi");
        ok[1] = rs.getString("Merk");
        ok[2] = rs.getString("Jenis");
        ok[3] = rs.getString("Tipe");
        Judul_Tbl.addRow(ok);}
    return Judul_Tbl;
} catch (SQLException ex) {}
return null;
}

public String [] getInfoProduk(String kode){
    try {
        String query = "SELECT u>Nama_Lengkap, ip.Kd_Produksi, p.Merk,\n"
            + "p.Tipe, l.Tgl_Terima, p.Jenis, p.Kapasitas, ip.Ket_Info\n"
            + "FROM `user` AS u JOIN lpb AS l ON u.Id_User = l.Penerima\n"
            + "JOIN info_produk_klaim AS ip ON l.No_LPB = ip.No_LPB\n"
            + "JOIN produk AS p ON ip.Id_Produk = p.Id_Produk\n"
            + "WHERE ip.Kd_Produksi = " + kode + """;
        stm = Koneksi.getInstANCE().connect(query);
        rs = stm.executeQuery();
        String [] result = null;
        while (rs.next()) {
            result = new String[8];
            result[0] = rs.getString("Nama_Lengkap");
            result[1] = rs.getString("Kd_Produksi");
            result[2] = rs.getString("Merk");
            result[3] = rs.getString("Tipe");
            result[4] = rs.getString("Tgl_Terima");
            result[5] = rs.getString("Jenis");
            result[6] = rs.getString("Kapasitas");
            result[7] = rs.getString("Ket_Info");}
        return result;
    } catch (SQLException ex) {}
    return null;
}
}

```

```

public String [] getHasilValidasi(String kode){
    try {
        String query = "SELECT Validasi, Catatan FROM validasi\n"
            + "WHERE Kd_Produksi = '" + kode + "'";
        stm = Koneksi.getISNTANCE().connect(query);
        rs = stm.executeQuery();
        String [] result = null;
        while (rs.next()) {
            result = new String[2];
            result[0] = rs.getString("Validasi");
            result[1] = rs.getString("Catatan");}
        return result;
    } catch (SQLException ex) {}
    return null;
}
public String [] getPeriksaAwal(String kode){
    try {
        String query = "SELECT \n"
            + "(SELECT k>Nama_Kerusakan AS cover FROM detil_kerusakan_produk AS
d JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan WHERE d.Kd_Produksi =
"+kode+" AND d.Bagian = 'Cover') AS cover,\n"
            + "(SELECT k>Nama_Kerusakan AS cover FROM detil_kerusakan_produk AS
d JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan WHERE d.Kd_Produksi =
"+kode+" AND d.Bagian = 'Sealing') AS sealing,\n"
            + "(SELECT k>Nama_Kerusakan AS cover FROM detil_kerusakan_produk AS
d JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan WHERE d.Kd_Produksi =
"+kode+" AND d.Bagian = 'Terminal') AS terminal,\n"
            + "(SELECT k>Nama_Kerusakan AS cover FROM detil_kerusakan_produk AS
d JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan\n"
            + "WHERE d.Kd_Produksi = '"+kode+"' AND d.Bagian = 'Elektrolit') AS
elektrolit, Stiker, Lama_Charge, Ket_Periksa FROM pemeriksaan_awal\n"
            + "WHERE Kd_Produksi = '"+kode+"'";
        stm = Koneksi.getISNTANCE().connect(query);
        rs = stm.executeQuery();
        String [] result = null;
        while (rs.next()) {
            result = new String[7];
            result[0] = rs.getString("cover");
            result[1] = rs.getString("sealing");
            result[2] = rs.getString("terminal");
            result[3] = rs.getString("elektrolit");
            result[4] = rs.getString("Stiker");
            result[5] = rs.getString("Lama_Charge");
            result[6] = rs.getString("Ket_Periksa");}
        return result;
    } catch (SQLException ex) {}
    return null;
}
public String [] getDataLoad(String kode){
    try {

```

```

String query = "SELECT lt.V_Tertutup, lt.V_Equalisasi, dc.Spgr_Ratarata, \n"
    + "dc.Volt_Jumlah, lt.Tes_Kemampuan, lt.Kesimpulan\n"
    + "FROM load_test AS lt JOIN data_cell AS dc\n"
    + "ON lt.Kd_DataCell = dc.Kd_DataCell\n"
    + "WHERE Kd_Produksi = '" + kode + "'";
stm = Koneksi.getISNTANCE().connect(query);
rs = stm.executeQuery();
String [] result = null;
while (rs.next()) {
    result = new String[6];
    result[0] = rs.getString("V_Tertutup");
    result[1] = rs.getString("V_Equalisasi");
    result[2] = rs.getString("Spgr_Ratarata");
    result[3] = rs.getString("Volt_Jumlah");
    result[4] = rs.getString("Tes_Kemampuan");
    result[5] = rs.getString("Kesimpulan");}
return result;
} catch (SQLException ex) {}
return null;
}
public String [] getDataBedah(String kode){
    try {
        String query = "SELECT \n"
            + "(SELECT k>Nama_Kerusakan AS PlateP FROM detil_kerusakan_produk AS
d JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan WHERE d.Kd_Produksi =
'+kode+' AND d.Bagian = 'Plate (+)' AS PlateP,\n"
            + "(SELECT k>Nama_Kerusakan AS PlateN FROM detil_kerusakan_produk AS
d JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan WHERE d.Kd_Produksi =
'+kode+' AND d.Bagian = 'Plate (-)' AS PlateN,\n"
            + "(SELECT k>Nama_Kerusakan AS sepa FROM detil_kerusakan_produk AS d
JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan WHERE d.Kd_Produksi =
'+kode+' AND d.Bagian = 'Separator') AS sepa,\n"
            + "(SELECT k>Nama_Kerusakan AS Endapan FROM detil_kerusakan_produk
AS d\n"
            + "JOIN kerusakan AS k ON d.Id_Kerusakan = k.Id_Kerusakan\n"
            + "WHERE d.Kd_Produksi = '"+kode+"' AND d.Bagian = 'Endapan') AS
Endapan,\n"
            + "Kesimpulan FROM bedah_baterai WHERE Kd_Produksi = '"+kode+"'";
stm = Koneksi.getISNTANCE().connect(query);
rs = stm.executeQuery();
String [] result = null;
while (rs.next()) {
    result = new String[5];
    result[0] = rs.getString("PlateP");
    result[1] = rs.getString("PlateN");
    result[2] = rs.getString("sepa");
    result[3] = rs.getString("Endapan");
    result[4] = rs.getString("Kesimpulan");}
return result;
} catch (SQLException ex) {}

```

```

return null;
}
public TableModel TabelTidakValid(String nama) {
try {
String query = "SELECT\n"
+ " l.Dari, v.Kd_Produksi, ip.No_LPB, p.Merk\n"
+ "FROM validasi AS v LEFT JOIN info_produk_klaim AS ip\n"
+ "ON v.Kd_Produksi = ip.Kd_Produksi LEFT JOIN lpb AS l\n"
+ "ON ip.No_LPB = l.No_LPB LEFT JOIN lpb_detil AS ld\n"
+ "ON l.No_LPB = ld.No_LPB LEFT JOIN produk AS p\n"
+ "ON ld.Id_Produk = p.Id_Produk LEFT JOIN `user` AS u\n"
+ "ON l.Penerima = u.Id_User WHERE Validasi = 'Tolak, Cek Kembali'\n"
+ "AND u>Nama_Lengkap = '" + nama + "'";
stm = Koneksi.getINSTANCE().connect(query);
rs = stm.executeQuery();
DefaultTableModel Judul_Tbl = new DefaultTableModel();
Judul_Tbl.addColumn("Customer");
Judul_Tbl.addColumn("Kode Produksi");
Judul_Tbl.addColumn("No LPB");
Judul_Tbl.addColumn("Merk");
Object[] ok;
while (rs.next()) {
ok = new Object[4];
ok[0] = rs.getString("Dari");
ok[1] = rs.getString("Kd_Produksi");
ok[2] = rs.getString("No_LPB");
ok[3] = rs.getString("Merk");
Judul_Tbl.addRow(ok);}
return Judul_Tbl;
} catch (SQLException ex) {}
return null;}
}

```

LAMPIRAN SCREEN SHOOT PROGRAM

1. Login



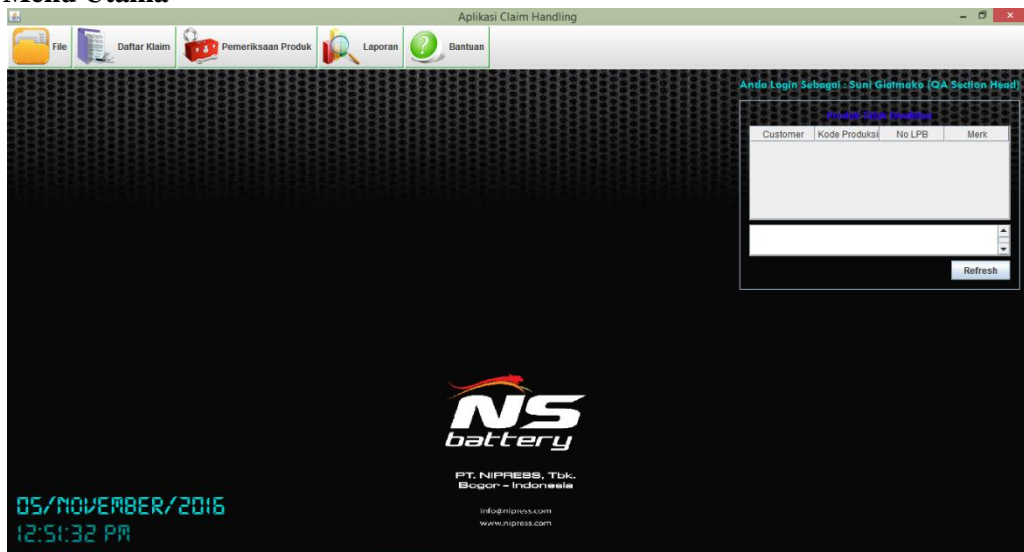
Silahkan Melakukan Login

Username :

Password :

Login Keluar

2. Menu Utama



Aplikasi Claim Handling

File Daftar Klaim Pemeriksaan Produk Laporan Bantuan

Anda Login Sebagai : Suni Giatmoko | QA Section Head

Customer Kode Produksi No LPB Merk

Refresh

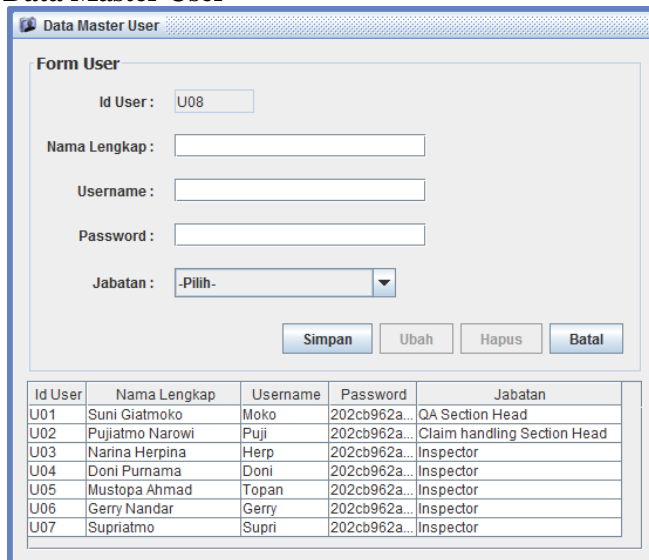
NS
battery

PT. NIPRESS, Tbk.
Bogor - Indonesia

05/NOVEMBER/2016
12:51:32 PM

info@nipress.com
www.nipress.com

3. Data Master User



Data Master User

Form User

Id User :

Nama Lengkap :

Username :

Password :

Jabatan :

Simpan Ubah Hapus Batal

Id User	Nama Lengkap	Username	Password	Jabatan
U01	Suni Giatmoko	Moko	202cb962a...	QA Section Head
U02	Pujiatmo Narowi	Puji	202cb962a...	Claim handling Section Head
U03	Narina Herpina	Herp	202cb962a...	Inspector
U04	Doni Purnama	Doni	202cb962a...	Inspector
U05	Mustopa Ahmad	Topan	202cb962a...	Inspector
U06	Gerry Nandar	Gerry	202cb962a...	Inspector
U07	Supriatmo	Supri	202cb962a...	Inspector

4. Data Master Produk

Data Master Produk Baterai AKI

Form Produk

Id Produk : P0066

Merk :

Jenis :

Tipe :

Kapasitas : AH

Tegangan : Volt

Cari berdasarkan Merk / Tipe baterai aki:

Id Produk	Merk Baterai	Jenis	Tipe	AH	V
P0065	NSGF 12-180 Accelerate	Industry	F9	160	12
P0064	NSGF 12-100H Accelerate	Industry	F9	100	12
P0063	NSGF 12-90 Accelerate	Industry	F6	80	12
P0062	OPzS 2-800 Accelerate	Industry	F10-M8	800	12
P0061	OPzS 2-300 Accelerate	Industry	F10-M8	300	12
P0060	OPzS 2-150 Accelerate	Industry	F10-M8	150	12
P0059	OPzS 2-100 Accelerate	Industry	F10-M8	100	12
P0058	NSAG 12-260 Accelerate	Industry	F14	231	12
P0057	NSAG 12-160 Accelerate	Industry	F12/F16	142	12
P0056	NSAG 12-145 Accelerate	Industry	F5/F12	129	12
P0055	FGB 12-100 Accelerate	Industry	F10-MB	100	12
P0054	NSAG 12-100 Accelerate	Industry	F5/F12	89	12
P0053	NSAG 12-70 Accelerate	Industry	F5/F11	62	12
P0052	NSAG 12-55 Accelerate	Industry	F11/F15	49	12
P0051	NSAG 12-40 Accelerate	Industry	F4/F11	36	12
P0050	N200 Advance	Automobile Battery	190H52	200	12
P0049	N150 Advance	Automobile Battery	145G51	150	12
P0048	NT200-12 Advance	Automobile Battery	150F51	135	12
P0047	N100 Advance	Automobile Battery	95E41R	100	12
P0046	N70Z Advance	Automobile Battery	75D31R	75	12
P0045	NS70 Advance	Automobile Battery	75D26R	65	12
P0044	N40L Advance	Automobile Battery	32C24L	40	12
P0043	NX100-S6 Advance	Automobile Battery	55B24R	45	12
P0042	NS60 Advance	Automobile Battery	50B24R	45	12
P0041	NS40ZLS Advance	Automobile Battery	45B19LS	35	12
P0040	NS40S Advance	Automobile Battery	40B19RS	33	12
P0039	N200 Accurate	Automobile Battery	190H52	200	12
P0038	N150 Accurate	Automobile Battery	145G51	150	12

5. Data Master Kerusakan

Data Master Kerusakan

Form Kerusakan

Id Kerusakan : K084

Nama Kerusakan:

Bagian :

Kesalahan : Pabrik Customer

Cari Berdasarkan Nama kerusakan/ Bagian kerusakan baterai aki:

Id Kerusakan	Nama Kerusakan	Bagian	Kesalahan
K083	Tidak Ada	-	-
K082	Cukup	-	-
K081	Baik	-	-
K080	Plate Negative lunak	Plate (-)	Customer
K079	Grid plate crack	Plate (-)	Customer
K078	Plate memulih	Plate (-)	Customer
K077	Plate Mengeras	Plate (-)	Pabrik
K076	Plate Belang Kuning (dry charge battery)	Plate (-)	Pabrik
K075	Plate Undercharge (wet charge battery)	Plate (-)	Pabrik
K074	Tidak ada kilap metalik	Plate (-)	Pabrik
K073	Plate Negative Rontok (negative plate f...	Plate (-)	Pabrik
K072	Lug Plate Patah (lug of plate broken)	Plate (-)	Pabrik
K071	Las lasan Plate Lepas pada strap (dis...	Plate (-)	Pabrik
K070	Short (plate flashing)	Plate (-)	Pabrik
K069	Short (plate bengkok/bent plate)	Plate (-)	Pabrik
K068	Short (plate (-) naik)	Plate (-)	Pabrik
K067	Endapan pada cell	Endapan	Customer
K066	Short (Sedding paste/rontokan pasta)	Endapan	Pabrik
K065	Short (pasta tebal (nempel)/material p...	Endapan	Pabrik
K064	Separator menghitam	Separator	Customer
K063	Separator lengket pada plate	Separator	Customer
K062	Las lasan Bush Bar/strap Patah / Lep...	Separator	Pabrik
K061	Short separator terlipat	Separator	Pabrik
K060	Short separator panjang/pendek sebe...	Separator	Pabrik
K059	Separator Tidak Rapih	Separator	Pabrik
K058	Separator motor Kotor (Battery belum ...	Separator	Pabrik
K057	Short (separator sobek/ripped separa...	Separator	Pabrik
K056	Plate Cokelat muda / Terang	Plate (+)	Customer
K055	Grid plate crack	Plate (+)	Customer
K054	Plate + rapuh/nusak	Plate (+)	Customer
K053	Plate memulih	Plate (+)	Customer
K052	Plate Belang Kuning (dry charge battery)	Plate (+)	Pabrik

6. Form Penerimaan Barang

Laporan Penerimaan Barang

Form LPB

No LPB:

Tanggal Terima:

Dari:

No Surat Jalan:

Penyerah:

Penerima:

Cari berdasarkan No LPB / Nama Perusahaan :

No LPB	Tgl Terima	Dari	No Surat Ja...	Penyerah	Penerima
X11600006	02-11-2016	PT Suzuki	PK/0211	Didik	Doni Purna...
X1600005	31-10-2016	PT Nissan Indonesia	23/D/OPE	Suhendar	Doni Purna...
X1600004	16-10-2016	PT Kereta Api Indonesia	SAK222A	Didin	Doni Purna...
X1600003	15-10-2016	PT Kereta Api Indonesia	S234JKL	Rony	Doni Purna...
X1600002	12-10-2016	PT Astra	Ky12345	Didik	Doni Purna...
X1600001	12-10-2016	PT Daihatsu	A1234	Juju	Doni Purna...

7. Form Detil LPB

Detail Laporan Penerimaan Barang

Form Detil LPB

No LPB :

Id Produk :

Quantity :

Keterangan :

No LPB	Id Produk	Quantity	Keterangan
X1600005	P0020	1	
X1600005	P0019	2	
X1600005	P0010	2	
X1600005	P0009	3	
X1600004	P0007	1	
X1600003	P0012	1	
X1600002	P0043	1	
X1600001	P0061	1	

8. Form Pemeriksaan Awal Tab Info Produk Klaim

Pemeriksaan Awal

Info Produk Klaim | Cek Tampilan Luar

Kode Produksi :

No LPB :

No SJ :

Tgl Terima :

Id Produk :

Tipe :

ASSY / Jenis :

Km Kendaraan : KM

Tanggal Pakai :

Tanggal Turun :

Keterangan :

Data Produk Klaim

Kd Produksi	No LPB	Id Prod	Km	Tgl Pakai	Tgl Turun	Keterangan
B315J170011	X1600005	P0009	1111	01-10-2016	31-10-2016	Lanjut Klaim
B315J170012	X1600005	P0009	1234	08-10-2016	14-10-2016	Lanjut Klaim
B315J170013	X1600005	P0009	2344	01-10-2016	15-10-2016	Lanjut Klaim
C112L030999	X1600005	P0010	3333	01-10-2016	31-10-2016	Produk Kadaluarsa
C116L030999	X1600005	P0010	2222	14-10-2016	28-10-2016	Lanjut Klaim
A115H200998	X1600004	P0007	1200	01-10-2016	15-10-2016	Lanjut Klaim
A115H200991	X1600003	P0012	2000	07-10-2016	08-10-2016	Lanjut Klaim
A115H200992	X1600002	P0043	6000	01-10-2016	08-10-2016	Lanjut Klaim
A110H200943	X1600001	P0061	2009	08-10-2016	14-10-2016	Produk Kadaluarsa

9. Form Pemeriksaan Awal Tab Cek Tampilan Luar

Pemeriksaan Awal

Info Produk Klaim | Cek Tampilan Luar

Kode Produksi :

Penampilan Cover : Baik

Sealing : Baik

Terminal Pole : Baik

Elektrolit : Cukup

Stiker Aluminium : Sudah dibuka Belum dibuka

Keterangan :

Hitung Masa Charge

Tegangan : Volt cca : R :

Cell 1 2 3 4 5 6

SP Gr20 Rata-rata

Voltase Jumlah

Lama Charge : Jam

Data Periksa Awal

Kd Produksi	No LPB	Merk	Tipe	Keterangan	Charge (jam)
C116L030999	X1600005	N200 Absolute	190H52	Stop Klaim	
B315J170013	X1600005	N150 Absolute	145G51	Lanjut Klaim	12
B315J170012	X1600005	N150 Absolute	145G51	Lanjut Klaim	12
B315J170011	X1600005	N150 Absolute	145G51	Lanjut Klaim	12
A115H200992	X1600002	INX100-S6 Advance	55B24R	Lanjut Klaim	12
A115H200991	X1600003	DIH45 Absolute	54533	Lanjut Klaim	12
A115H200998	X1600004	N100 Absolute	95E41R	Stop Klaim	

10. Form Pasca Charge dan Load Test

Pemeriksaan Setelah Charge & Load Test

Form Pasca Charge

Kode Produksi:

Tegangan Tertutup: Volt Cca:

Voltase Setelah Equalisasi: Volt R:

Cell	1	2	3	4	5	6	Rata-rata
SP Gri20	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Voltase	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Jumlah <input type="text"/>

Form Kemampuan Baterai

Catatan:
Parameter test berdasarkan BT Load Test (400A untuk mobil dan 350A untuk motor).

Parameter warna:
 Hijau Kuning Merah

Kesimpulan:

Data Pemeriksaan

Kode Produksi:

Kd Produksi	No LPB	Indikator	Kesimpulan
B315J170013	X1600005	Merah	Lanjut Klaim
B315J170011	X1600005	Hijau	Baterai normal. Sto...
B315J170012	X1600005	Merah	Lanjut Klaim
A115H200991	X1600003	Hijau	Baterai normal. Sto...
A115H200992	X1600002	Merah	Lanjut Klaim

11. Form Bedah Baterai

Bedah Baterai

Form Bedah Baterai

Kode Produksi:

Plate (+):

Plate (-):

Separator:

Endapan:

Kesimpulan Akhir:

Kd Prod	No LPB	Customer	Merk	Jenis	Tipe	Kapasitas	Volt	Kesimpulan
B315J170013	X1600005	PT Nissan Indonesia	N150 Absolute	Automobile Battery	145G51	150	12	Handling Fault
B315J170012	X1600005	PT Nissan Indonesia	N150 Absolute	Automobile Battery	145G51	150	12	Manufacturing Fault
A115H200992	X1600002	PT Astra	NX100-S6 Advance	Automobile Battery	55B24R	45	12	Manufacturing Fault

Kd Produksi	No LPB
B315J170012	X1600005
B315J170013	X1600005
A115H200992	X1600002

12. Form Pilih Laporan LPB

Laporan Penerimaan Barang

Dari:

Sampai:

13. View Laporan LPB

Export PDF

Laporan Penerimaan

Sel Nov 05 13:02:54 ICT 2018

No.	Tanggal Terima	Customer	No SJ	Qty
1	2018-10-12	PT Dahatsu	A1234	1
2	2018-10-12	PT Asia	KJ12345	1
3	2018-10-15	PT Kereta Api Indonesia	S214JKL	1
4	2018-10-18	PT Kereta Api Indonesia	SAK222A	1
5	2018-10-31	PT Nenas Indonesia	Z1DJKQE	5
6	2018-11-02	PT Suzuki	PK0211	3

Page 1 of 1

14. View Pilih Laporan Kerusakan

Laporan Kerusakan

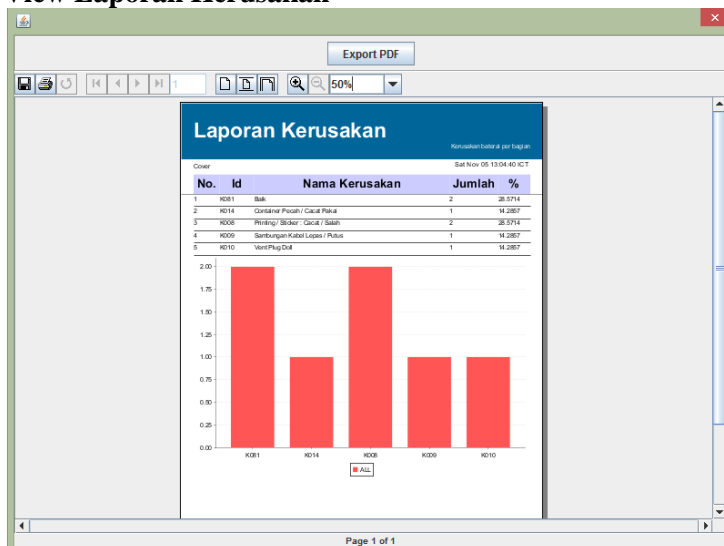
Bagian Baterai : -- Pilih --

Dari :

Sampai :

Tampilkan Laporan

15. View Laporan Kerusakan



16. Form Validasi Produk

Validasi Produk

Data Pemohon Klaim

No LPB	Customer	Tgl Ter
X1600005	PT Nissan Indonesia	2016-10-
X1600004	PT Kereta Api Indonesia	2016-10-
X1600003	PT Kereta Api Indonesia	2016-10-
X1600002	PT Astra	2016-10-
X1600001	PT Daihatsu	2016-10-
X1600006	PT Suzuki	2016-11-

Data Produk Klaim

Nm Customer :

Jumlah :

Title 1	Title 2	

Divalidasi Oleh :

Claim Handling Section Head

Tidak divalidasi

Form Falidasi Klaim

Petugas :

Kode Produksi : Tanggal Masuk :

Merk : Jenis :

Tipe : Kapasitas : AH

Pemeriksaan Awal Produk

Cover :

Sealing :

Terminal :

Elektrolit :

Stiker : Lama Charge : Jam

Tes Kemampuan Produk

Tegangan Tertutup : Tegangan Equalisasi :

SpGr Rata-rata : Jumlah Voltase :

Indikator Warna :

Bedah Baterai

Plat Positif (+) :

Plat Negatif (-) :

Separator :

Endapan :

Hasil

Kesimpulan :