

**RANCANG BANGUN SISTEM INFORMASI PELAPORAN
DAILY DEFECT UNIT KENDARAAN MOBIL TIPE
TDMENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 DI PT
KRAMA YUDHA RATU MOTOR**

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Syarat Penyelesaian Program Diploma
Empat (D- IV) Program Studi Sistem Informasi Industri Otomotif pada
Politeknik STMI Jakarta

OLEH

NAMA : HALIMATUS SYAKDIYA

NIM : 1310034



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2017**

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Halimatus Syakdiya

NIM : 1310034

Berstatus sebagai mahasiswa jurusan program studi Sistem Informasi Industri Otomotif pada Politeknik STMIK Kementerian Perindustrian R.I Jakarta. Dengan ini menyatakan bahwa hasil karya Tugas Akhir yang saya buat dengan judul:

“RANCANG BANGUN SISTEM INFORMASI PELAPORAN DAILY DEFECT UNIT KENDARAAN MOBIL TIPE TDMENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 DI PT KRAMA YUDHA RATU MOTOR”.

Dibuat dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, *survey* 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 atau 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 saya nyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai dengan peraturan yang berlaku.

Jakarta, November 2017

Yang Membuat Pernyataan,

Halimatus Syakdiya

ABSTRAK

PT Krama Yudha Ratu Motor (KRM) adalah perusahaan yang memproduksi mobil niaga jenis truk tipe TD. Setiap komponen truk tipe TD didapatkan dari PT Krama Yudha Tiga Berlian (KTB) yang dikirim ke PT Krama Yudha Ratu Motor(KRM) akan melalui proses pengecekan sebelum dirakit menjadi satu unit kepala truck. Proses pengecekan ini disebut dengan pengecekan *daily defect*. Pengecekan *daily defect* merupakan suatu pengumpulan data kerusakan perhari yang dilakukan setelah pengecekan mobil tipe TD oleh bagian Trimming. Setelah penyebab *defect* diketahui, *defect* dicatat ke dalam dokumen Countermeasure dan diberikan selambat-lambatnya satu hari ke bagian Quality Inspection (QI). Selama periode satu bulan, dokumen Countermeasure yang telah terkumpul di Quality Inspection (QI), dijadikan dasar pembuatan laporan bulanan Countermeasure dengan menggunakan *Microsoft Office Excel*. Setelah laporan bulanan Countermeasure selesai dibuat, pihak Quality Inspection (QI) mengirim laporan ke pihak Trimming 1 lewat *email*. Pada proses penemuan *daily defect*, menemukan beberapa kendala diantaranya pembuatan laporan bulanan Countermeasure di PT Krama Yudha Ratu Motor (KRM) dinilai kurang efisien karena sering terjadi keterlambatan penyerahan dokumen Countermeasure oleh pihak Trimming 1. Kendala lain yang ditemukan pihak Trimming kesulitan dalam penerimaan laporan bulanan lewat email dikarenakan terlalu banyak *email* yang masuk di bagian Trimming 1. Oleh karena itu, dibangun sebuah sistem informasi dengan metode pengembangan sistem *evolutionary prototype*. Sistem informasi ini dirancang dengan menggunakan *Unified Modelling Language* (UML). Sistem Informasi Pelaporan *Daily Defect* ini dibangun dengan menggunakan PHP 5.4.7 dan MySQL 5.5.27. Sistem Informasi ini dapat mempercepat dan mempermudah proses pembuatan laporan bulanan oleh pihak Quality Inspection tanpa harus menunggu penyerahan dokumen Countermeasure dari pihak Trimming 1.

Kata kunci: Sistem Informasi Pelaporan, *Daily Defect*, *Evolutionary Prototype*, *Flowchart*, PHP, MySQL, HIPO, UML.

KATA PENGANTAR

Segala puji dan syukur atas kasih sayang dan karunia yang diberikan Allah SWT sehingga dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Sistem Informasi Pelaporan *Daily Defect* Unit Kendaraan Mobil Tipe TD Menggunakan PHP 5.4.7 dan MySQL 5.5.27 di PT Krama Yudha Ratu Motor”**.

Tujuan penulisan tugas akhir ini untuk memenuhi salah satu syarat penyelesaian Program Studi Diploma Empat (D-4) Sistem Informasi Industri Otomotif pada Politeknik STMI Kementerian Perindustrian R.I Jakarta

Tugas Akhir ini dapat disusun dengan baik karena mendapat banyak masukan dan dukungan dari berbagai pihak yang berupa informasi, arahan dan bimbingan. Oleh karena itu pada kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Kedua orang tua dan adikku tercinta yang dengan penuh kasih selalu mendoakan, memberi semangat dan dukungan untuk menyelesaikan Tugas Akhir.
2. Bapak Dr. Mustofa, ST, MT. selaku Direktur Politeknik STMI Jakarta.
3. Bapak Drs. Jacob Saragih, MM. selaku Ketua Program Studi Sistem Informasi Industri Otomotif yang memberi arahan dan masukan dalam penyusunan tugas akhir
4. Bapak Dedy Trisanto, S.Kom, MMSI selaku Dosen Pembimbing yang senantiasa mengarahkan, memberikan saran dan membantu dalam penyusunan Tugas Akhir.
5. Ibu Fifi L. Hadianastuti S.Kom, M.Kes selaku Asisten Dosen Pembimbing yang senantiasa mengarahkan, memberikan saran dan membantu dalam penyusunan Tugas Akhir.

6. Bapak Dr. Ridzky Kramanandita, S.Kom, MT selaku Pudir 1 yang telah membantu mengkoordinasi dan memberikan kesempatan dalam perpanjangan waktu penyelesaian Praktek Kerja Lapangan dan Laporan Tugas Akhir.
7. Ibu Noveriza Yuliasari, S.Si, MT. Selaku Sekretaris prodi SIIO dan seluruh pengurus prodi SIIO yang telah membantu mengkoordinasi dan memfasilitasi mahasiswa SIIO 2010 dalam penyelesaian Tugas Akhir.
8. Seluruh dosen Politeknik STMI yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang berguna.
9. Bapak Adang selaku staff Quality Inspection, Bapak Surya selaku staff Quality Assurance, dan Bapak Fadly selaku staff Trimming 1 yang telah memberikan izin dan kesempatan serta memberikan informasi selama Praktek Kerja Lapangan di PT Krama Yudha Ratu Motor.
10. Dzul Faqar, Khairunnisa, Balqis, Annisha, Trisno, Tina, Dimas selaku teman kelompok belajar yang selalu memotivasi dan membantu baik dalam suka maupun duka selama penyelesaian Tugas Akhir.
11. Siti Rachmah, Desi Lusiana Pakpahan, Alfredo dan Triaselaku teman yang memberikan support dan motivasi selama penyelesaian Tugas Akhir.
12. Semua teman-teman yang senantiasa saling mendukung terutama teman jurusan SIIO pada Politeknik STMI Jakarta angkatan 2010 serta semua pihak yang baik langsung maupun tidak langsung memberikan kritik, saran dalam pembuatan Tugas Akhir.

Dalam penyusunan Tugas Akhir ini, terdapat banyak kekurangan dan kelemahan yang dimiliki, baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu diharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan Tugas Akhir.

Jakarta, November 2017

Halimatus Syakdiya

DAFTAR ISI

Halaman

Halaman Judul	
Lembar Persetujuan Dosen Pembimbing	
Lembar Persetujuan Asisten Dosen Pembimbing	
Lembar Pengesahan	
Lembar Bimbingan Tugas Akhir Dosen Pembimbing	
Lembar Bimbingan Tugas Akhir Asisten Dosen Pembimbing	
Lembar Pernyataan Keaslian	
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL.....	ix
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Pokok Permasalahan.....	3
1.3 . Tujuan Penelitian.....	3
1.4 . Batasan Masalah.....	3
1.5. Manfaat Penelitian.....	4
1.6. Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1. Pengertian Rancang Bangun.....	6
2.2. Pengertian Sistem	6
2.3. Karakteristik Sistem	6
2.3.1 Informasi.....	9
2.3.1 Siklus Informasi.....	9
2.4. Kualitas Informasi	10
2.5. Sistem Informasi.....	11
2.6 . Pengembangan Sistem.....	13

2.6.1	<i>Model Prototype</i>	15
2.6.2	<i>Evolutionary Prototype</i>	17
2.7 .	Pengertian Quality Assurance	158
2.8 .	Pengertian Quality Inspection	15
2.9.	Definisi Daily Defect.....	21
2.10.	Flowchart.....	21
2.10.1	Bagan Alir atau Flowchart.....	21
2.11.	Unified Modelling Language (UML).....	25
2.11.1	Bagian-bagian UML.....	26
2.12.	Kamus Data (Data Dictionary)	35
2.13.	Hierarchy Plus Input-Process-Output (HIPO).....	36
2.14.	Hypertext Preprocessor (PHP)	38
2.15.	MySQL	38
BAB III METODOLOGI PENELITIAN.....		41
3.1.	Metodologi Penelitian	41
3.2.	Pengumpulan Data.....	41
3.3.	Metode Pengembangan Sistem.....	42
3.4.	Kerangka Penelitian.....	43
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA.....		47
4.1.	Hubungan Kerja Sama PT Krama Yudha Ratu Motor.....	47
4.2.	Struktur Organisasi.....	48
4.3.	Jenis-Jenis Produk	51
4.4.	Sistem Pengecekan Kualitas.....	53
4.5.	Quality Assurance	53
4.6.	Quality Inspection (QI).....	54
4.7.	Trimming 1 (Perakitan).....	55
4.8.	<i>Daily Defect</i>	56
4.9.	Observasi <i>Daily Defect</i>	60
4.10.	Faktor Pengaruh Masalah pada Kendaraan Mobil Tipe TD.....	60
4.11.	<i>Flowmap</i> Proses <i>Daily Defect</i> PT Krama Yudha Ratu Motor	60
4.12.	Proses Alur Kerja <i>Daily Defect</i> Unit Kendaraan Mobil Tipe TD ..	62

4.13. Penggambaran Sistem <i>Daily Defect</i> Unit Kendaraan Mobil Tipe TD Dengan <i>Unified Modelling Language</i> (UML)	62
BAB V ANALISIS DAN PEMBAHASAN	68
5.1. Analisis Kebutuhan User	68
5.2. Perancangan Sistem Informasi <i>Daily Defect</i>	70
5.3. <i>Flowmap</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan	71
5.4. <i>Use Case Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan.....	72
5.5. <i>Usecase Description</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan.....	72
5.6. <i>Activity Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan	76
5.7. <i>Sequence Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan	81
5.8. <i>Class Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan..	86
5.9. Kamus Data	87
5.10. <i>Deployment Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan.....	90
5.11. <i>Hierarchy Plus-Input-Process-Output</i> (HIPO) Sistem Informasi <i>Daily Defect</i> yang Diusulkan.....	90
5.12. Perancangan Interface Sistem Informasi <i>Daily Defect</i>	91
5.13. Spesifikasi Kebutuhan <i>Software</i> dan <i>Hardware</i>	96
BAB VI KESIMPULAN DAN SARAN	96
6.1. Kesimpulan	97
6.2. Saran	97

DAFTAR GAMBAR

	Halaman
Gambar II.1	Karakteristik Sistem9
Gambar II.2	Siklus Informasi10
Gambar II.3	Blok Sistem Informasi yang Berinteraksi11
Gambar II.4	Mekanisme Pengembangan Sistem Dengan <i>Prototype</i>16
Gambar II.5	<i>EvolutionaryPrototype</i> Model.....18
Gambar II.6	Klasifikasi <i>Diagram</i> UML26
Gambar II.7	<i>Visual Table Of Contents</i>37
Gambar II.8	<i>Overview Diagram</i>37
Gambar III.1	Kerangka Penelitian45
Gambar IV.1	Hubungan Kerja PT Krama Yudha Ratu Motor di dalam dan di luar Krama Yudha Mitsubishi Group48
Gambar IV.2	Struktur Organisasi PT Krama Yudha Ratu Motor49
Gambar IV.3	Jenis-Jenis Produksi PT Krama Yudha Ratu Motor52
Gambar IV.4	Varian Produk tipe TD di PT Krama Yudha ratu Motor53
Gambar IV.5	Ruang Lingkup Quality Inspection54
Gambar IV.6	Stasiun Kerja Trimming 1 Tipe TD pada PT Krama Yudha Ratu Motor.....56
Gambar IV.7	Dokumen Countermeasure.....58
Gambar IV.8	<i>Flowmap</i> Alur Kerja <i>Daily Defect</i> Unit Kendaraan Mobil <i>Type</i> TD PT Krama Yudha Ratu Motor.....61
Gambar IV.9	<i>Use Case</i> Proses <i>Daily Defect</i> PT Krama Yudha Ratu Motor63
Gambar V.1	<i>Flowmap</i> Sistem Informasi <i>Daily Defect</i> PT Krama Yudha Ratu Motor.....71
Gambar V.2	<i>Use Case Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan72
Gambar V.3	<i>Activity Diagram</i> <i>Login</i>77
Gambar V.4	<i>Activity Diagram</i> untuk Mengelola Data Master78
Gambar V.5	<i>Activity Diagram</i> untuk Membuat Data Countermeasure80

Gambar V.6	<i>Activity Diagram</i> untuk Membuat laporan bulanan Countermeasure.....	81
Gambar V.7	<i>Sequence Diagram Login</i>	82
Gambar V.8	<i>Sequence Diagram</i> Membuat Master <i>User</i>	82
Gambar V.9	<i>Sequence Diagram</i> Membuat Master Komponen	83
Gambar V.10	<i>Sequence Diagram</i> Mengelola Data Dokumen Countermeasure.....	84
Gambar V.11	<i>Sequence Diagram</i> Laporan Bulanan Countermeasure	85
Gambar V.12	<i>Class Diagram</i> Sistem Informasi <i>Daily Defect</i>	86
Gambar V.13	<i>Deployment Diagram</i> Sistem Informasi <i>Daily Defect</i> yang Diusulkan	89
Gambar V.14	HIPO Sistem Informasi <i>Daily Defect</i> yang Diusulkan	90
Gambar V.15	Rancangan <i>Form Login</i>	91
Gambar V.16	Rancangan Menu Utama	91
Gambar V.17	Rancangan Master <i>User</i>	92
Gambar V.18	Rancangan Master <i>User</i>	93
Gambar V.19	Menu Countermeasure	94

DAFTAR TABEL

		Halaman
Tabel II.1	Simbol Bagan Alir Sistem (<i>Systems Flowchart</i>).....	22
Tabel II.2	Simbol Bagan Alir Program (<i>Program Flowchart</i>).....	23
Tabel II.3	Simbol Bagan Alir Proses (<i>Process Flowchart</i>)	24
Tabel II.4	Simbol-Simbol <i>Use Case Diagram</i>	28
Tabel II.5	Simbol-Simbol <i>Class Diagram</i>	30
Tabel II.6	<i>Multiplicity</i>	31
Tabel II.7	Simbol-Simbol <i>Activity Diagram</i>	31
Tabel II.8	Simbol-Simbol <i>Sequence Diagram</i>	32
Tabel II.9	Simbol-Simbol <i>Component Diagram</i>	34
Tabel II.10	Simbol-Simbol <i>Deployment Diagram</i>	35
Tabel II.11	Beberapa tipe data pada MySQL	39
Tabel IV.1	Jenis-Jenis Produk PT Krama Yudha Ratu Motor	52
Tabel IV.2	Klasifikasi <i>Defect</i> dan 3 <i>Worst Problem</i>	60
Tabel IV.3	Definisi Aktor dan <i>Use Case</i> Mencetak Dokumen Countermeasure.....	64
Tabel IV.4	Definisi Aktor dan <i>Use Case</i> Menerima Dokumen <i>Layout</i> Countermeasure.....	64
Tabel IV.5	Definisi Aktor dan <i>Use Case</i> Pengambilan <i>Sample</i> Mobil <i>Type TD</i>	64
Tabel IV.6	Definisi Aktor dan <i>Use Case</i> Pengecekan <i>Daily Defect</i> Keseluruhan <i>Body Part Type TD</i>	65
Tabel IV.7	Definisi Aktor dan <i>Use Case</i> Mengembalikan Dokumen Countermeasure / Menulis Data Temuan <i>Daily Defect</i> (5W+1H) diCountermeasure	65
Tabel IV.8	Definisi Aktor dan <i>Use Case</i> Menerima Dokumen Countermeasure.....	65
Tabel IV.9	Definisi Aktor dan <i>Use Case</i> Membuat Laporan Bulanan (<i>Month</i> <i>Defect</i>) Di MS Excel	66

Tabel IV.10	Definisi Aktor dan <i>Use Case</i> Mengirim Laporan Bulanan (<i>Month Defect</i>) Lewat Email	66
Tabel IV.11	Definisi Aktor dan <i>Use Case</i> Menerima Laporan Bulanan (<i>Month Defect</i>) Lewat Email	66
Tabel V.1	Identifikasi Kebutuhan <i>User</i>	69
Tabel V.2	<i>Use Case Description Login</i>	73
Tabel V.3	<i>Use Case Description Master User</i>	74
Tabel V.4	<i>Use Case Description Master Komponen</i>	74
Tabel V.5	<i>Use Case Description Mengelola Data Dokumen Countermeasure</i>	75
Tabel V.6	<i>Use Case Description Mencetak Data Laporan Bulanan Countermeasure</i>	75
Tabel V.7	Spesifikasi Tabel Master <i>User</i>	87
Tabel V.8	Spesifikasi Tabel Master Komponen	87
Tabel V.9	Spesifikasi Tabel Countermeasure	88
Tabel V.10	Spesifikasi Tabel Detail Countermeasure	88

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Seiring dengan perkembangan ilmu pengetahuan dan teknologi, perkembangan komputer saat ini sudah sangat maju, serta kemampuan kinerja komputer sudah semakin berkembang. Dengan Berkembangnya teknologi komputer, perusahaan memanfaatkan internet untuk berkomunikasi dengan konsumen-konsumen prospektif, dengan cara yang tepat sasaran, personal, langsung, dan ekonomis. Kebutuhan terhadap komputer saat ini bukan lagi kebutuhan sekunder tapi sudah menjadi kebutuhan pokok. Sehingga teknologi komputer bisa meringankan proses kerja masyarakat, agar bisa melakukan aktivitas secara efektif dan cepat terhadap lingkungan yang terus berubah-ubah.

Teknologi sistem informasi dapat digunakan masyarakat untuk meringankan pekerjaan yang rumit. Sistem informasi dapat menciptakan pasar-pasar baru bahkan industri-industri baru. Seperti industri internet, industri jasa, industri multimedia interaktif, industri manufaktur dan industri otomotif. Salah satu contoh proses industri otomotif adalah harus bisa menjamin kualitas produk industri yang dihasilkan.

Manufaktur adalah suatu cabang industri yang mengaplikasikan mesin, peralatan dan tenaga kerja pada medium proses untuk mengubah bahan mentah menjadi barang jadi untuk dijual. Sedangkan industri otomotif, industri yang merancang, mengembangkan, memasarkan, dan menjual kendaraan bermotor termasuk mobil dan kendaraan komersial yang diproduksi ke seluruh dunia. Dalam industri otomotif terdapat kegiatan *daily defect* untuk menjaga dan menjamin kualitas produknya.

PT Krama Yudha Ratu Motor (KRM) adalah perusahaan yang memproduksi mobil niaga jenis truk tipe TD. Komponen tipe TD *diassembly* di bagian Trimming 1 Komponen-komponen truck tipe TD didapatkan dari PT Krama Yudha Tiga Berlian (KTB) yang dikirim ke PT Krama Yudha Ratu Motor

(KRM) untuk dirakit menjadi satu unit kepala truck. Sesudah unit tersebut melewati *lineprocess* atau alur kerja Welding, Painting, dan Trimming barang diinspeksi oleh tim *audit*, jika dinyatakan OK kemudian dikirim kembali ke PT KTB.

Daily Defect yang dilakukan oleh PT Krama Yudha Ratu Motor (KRM) merupakan suatu pengumpulan data kerusakan perhari, yang setelah dilakukan pengecekan mobil tipe TD oleh bagian Trimming. Setelah penyebab *defect* diketahui, *defect* dicatat ke dalam dokumen Countermeasure dan diberikan selambat-lambatnya satu hari ke bagian Quality Inspection (QI). Selama periode satu bulan dokumen Countermeasure terkumpul di Quality Inspection (QI), seluruh data Countermeasure dijadikan dasar pembuatan laporan bulanan Countermeasure ke dalam Microsoft Office Excel. Setelah laporan bulanan Countermeasure selesai dibuat, pihak Quality Inspection (QI) mengirim laporan ke pihak Trimming 1 lewat *email*. Data laporan bulanan Countermeasure ini digunakan untuk pengambilan keputusan dan mengurangi kesalahan produksi kedepannya oleh pihak Trimming 1 dan Quality Inspection (QI).

Pembuatan laporan bulanan Countermeasure di PT Krama Yudha Ratu Motor (KRM) dinilai kurang efektif karena sering terjadi keterlambatan penyerahan dokumen Countermeasure oleh pihak Trimming 1. Kendala lain yang ditemukan pihak Trimming dalam penerimaan laporan bulanan lewat *email* adalah kesulitan dalam pengecekan *email* karena terlalu banyak *email* yang masuk di bagian Trimming 1. Berdasarkan uraian yang telah dipaparkan diatas dengan latar belakang permasalahan maka penulisan Tugas akhir ini mengambil judul **“RANCANG BANGUN SISTEM INFORMASI PELAPORAN DAILY DEFECT UNIT KENDARAAN MOBIL TIPE TD MENGGUNAKAN PHP 5.4.7 DAN MYSQL 5.5.27 DI PT KRAMA YUDHA RATU MOTOR”**.

1.2. Pokok Permasalahan

Permasalahan yang terjadi pada PT Krama Yudha Ratu Motor dalam sistem *daily defect* adalah sebagai berikut:

1. Pembuatan laporan bulanan *defect* mobil oleh Quality Inspection (QI) dilakukan dengan menunggu keseluruhan dokumen Countermeasure. Kendala yang ditemukan adalah sering terjadi keterlambatan dalam penyerahan dokumen Countermeasure oleh bagian Trimming 1.
2. Laporan bulanan yang telah selesai diproses oleh Quality Inspection (QI), dikembalikan ke bagian Trimming lewat *email* setiap bulannya. Namun, dalam pengecekan lewat *email* menemukan kendala, diantaranya kesulitan dalam pengecekan *email* karena terlalu banyak *email* yang masuk di bagian Trimming 1.

1.3. Tujuan Penelitian

Adapun tujuan dari Tugas Akhir adalah sebagai berikut:

1. Merancang sistem informasi pelaporan *daily defect* untuk memudahkan pembuatan laporan bulanan.
Pembuatan laporan bulanan oleh Quality Inspection (QI) dapat dilakukan tanpa harus menunggu penyerahan dokumen Countermeasure.
2. Merancang sistem informasi pelaporan *daily defect* untuk memudahkan pengecekan laporan bulanan.
Trimming 1 dapat mengakses laporan bulanan *daily defect* tanpa melalui *email* dan dapat diakses berdasarkan data bulanan yang dibutuhkan.

1.4. Batasan Masalah

Agar dalam penulisan Tugas Akhir ini lebih fokus dan lebih terarah, maka perlu adanya batasan-batasan sebagai berikut :

1. Dalam Tugas Akhir ini, pembahasan hanya terbatas pada sistem informasi pelaporan *daily defect* yang dilakukan pada bagian Trimming khususnya Trimming 1.

2. Sistem informasi pelaporan *daily defect* diamati pada produk kendaraan tipe TD yang berada di bagian Trimming 1 .
3. Sistem informasi pelaporan *daily defect*, laporan dokumen yang terkait mengenai penemuan kecacatan hanya membahas dokumen Countermeasure.

1.5. Manfaat Penelitian

Hasil dari pembuatan Tugas Akhir diharapkan dapat bermanfaat:

1. Bagi Mahasiswa
Rancang Bangun Sistem Informasi Pelaporan *Daily Defect* sebagai pengujian untuk menerapkan teori-teori yang telah diperoleh selama perkuliahan.
2. Bagi Perusahaan
Rancang Bangun Sistem Informasi Pelaporan *Daily Defect* diharapkan dapat mempermudah mengakses laporan *daily defect* yang dibuat oleh Quality Inspection (QI) untuk pihak Trimming .
3. Bagi Pembaca
Rancang Bangun Sistem Informasi Pelaporan *Daily Defect* diharapkan dapat memberikan pengetahuan kepada pembaca khususnya dalam hal pengembangan sistem informasi.

1.6. Sistematika Penulisan

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

BAB I PENDAHULUAN

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

BAB II LANDASAN TEORI

Bab ini membahas tentang Rancang Bangun, Pengertian Quality Assurance (QA), Pengertian Quality Inspection (QI), Pengertian Sistem, Sistem Informasi, Sistem *Daily Defect*, *Flowchart*, *Unified Modelling Language*(UML), *Hierarchy Plus Input-Process-Output* (HIPO), sebagai alat bantu untuk membuat rancangan konseptual, teori-teori yang berhubungan dengan perancangan sistem, *Hypertext Preprocessor*(PHP), dan *My Structure Query Language* (MySQL).

BAB III METODELOGI PENELITIAN

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

BAB IV PENGUMPULAN DAN PENGGOLAHAN DATA

Bab ini akan mengumpulkan dan mengolah data hasil pengamatan yang telah dilakukan di PT Krama Yudha Ratu Motor, yang menjelaskan *process problem daily defect* pada PT Krama Yudha Ratu Motor. Penggambaran *process problem daily defect* menggunakan UML.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini berisi tentang perancangan sistem yang dimulai dari analisis, yakni mulai dari analisis sistem yang meliputi perancangan proses, perancangan UML, perancangan basis data, perancangan antarmuka, perancangan hierarki menu, dan pembuatan spesifikasi proses melalui metode yang diterapkan.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan secara keseluruhan yang diperoleh dari hasil penelitian serta saran-saran dalam penerapan Sistem Informasi *Daily Defect* untuk perusahaan dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1. Pengertian Rancang Bangun

Rancang berarti mengatur segala sesuatu sebelum bertindak, mengerjakan atau melakukan sesuatu untuk merencanakan. Sedangkan kata bangun berarti sesuatu yang didirikan. Rancang bangun berarti merencanakan atau merancang sesuatu yang akan dibuat (Kamus Besar Bahasa Indonesia, 2002).

2.2. Pengertian Sistem

Kata sistem sendiri berasal dari bahasa latin “systema” dan bahasa Yunani “sustema” adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Berikut merupakan pendapat beberapa ahli mengenai sistem:

1. Sistem adalah kumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai tujuan (Kadir, 2003).
2. Sistem adalah suatu kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata (Jogiyanto, 2005).
3. Sistem adalah suatu kumpulan atau himpunan dari unsur, komponen atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu (Sutabri, 2012).

Dari uraian diatas, dapat disimpulkan sistem adalah satu kesatuan yang terdiri dari dua atau lebih komponen yang berinteraksi untuk mencapai suatu tujuan.

2.3. Karakteristik Sistem

Menurut Jogiyanto (2005), suatu sistem mempunyai sifat atau karakteristik tertentu, yaitu antara lain sebagai berikut:

1. Komponen-komponen (*components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu yang lebih besar lagi yang disebut *supra system*.

2. Batasan sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem (*environments*)

Lingkungan luar dari suatu sistem adalah luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung (*interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran dari satu subsistem akan menjadi masukan untuk subsistem yang lainnya melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan (*input*)

Masukan adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh di dalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran (*output*)

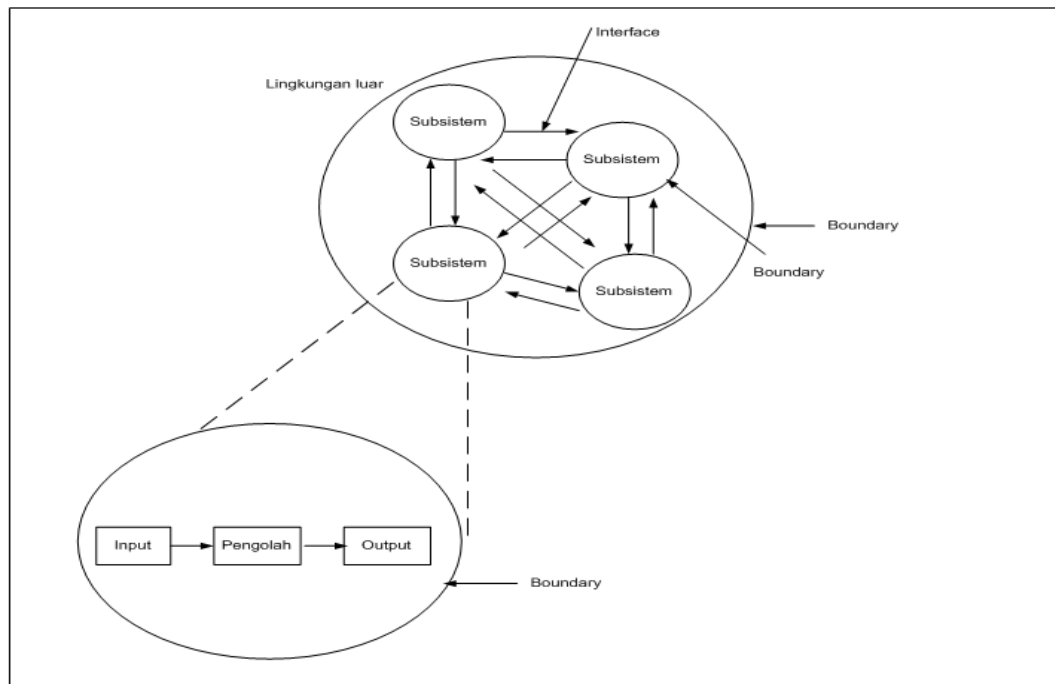
Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan adalah keluaran yang tidak berguna merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

7. Pengolah (*process*)

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran berupa barang jadi.

8. Sasaran (*objectives*) atau tujuan (*goal*)

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.



Gambar II.1 Karakteristik Sistem

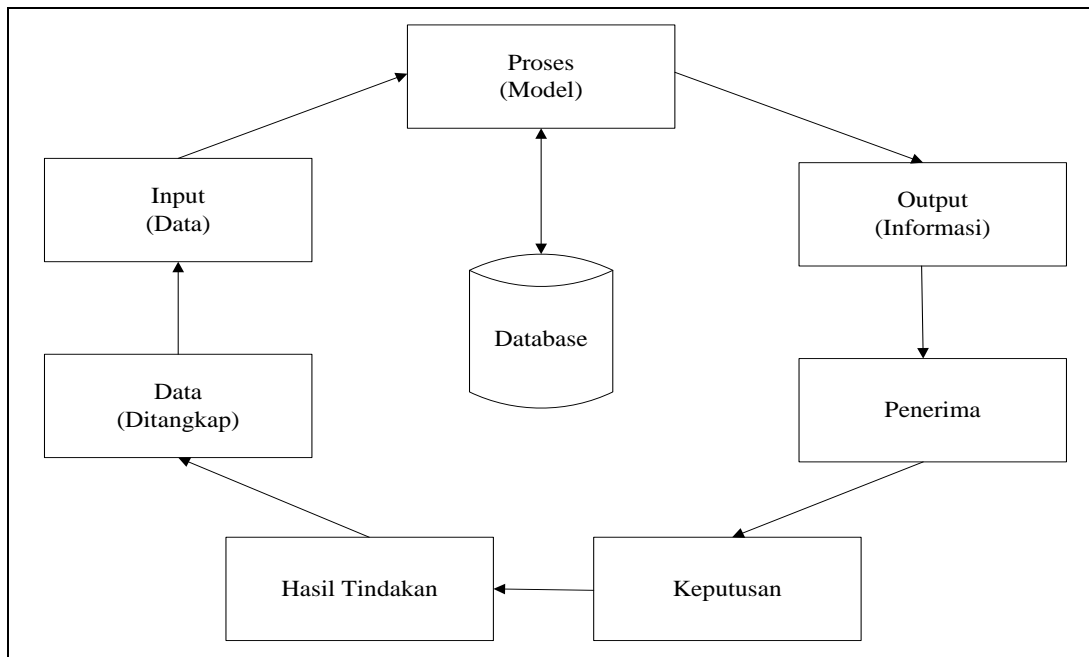
(Sumber: Jogiyanto, 2005)

2.4. Informasi

Menurut Jogiyanto (2005) informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

2.4.1 Siklus Informasi

Data agar menjadi lebih berarti dan berguna dalam bentuk informasi maka perlu diolah menjadi suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, dan diproses kembali lewat suatu model dan seterusnya sehingga membentuk suatu siklus. Siklus ini disebut dengan siklus informasi (*information cycle*) atau disebut pula siklus pengolahan data (*processing cycles*) (Jogiyanto, 2005).



Gambar II.2 Siklus Informasi

(Sumber: Jogiyanto, 2005)

2.4.2. Kualitas Informasi

Kualitas Informasi sangat dipengaruhi oleh 3 hal, yaitu (Jogiyanto, 2005):

1. Relevan (*Relevancy*)

Relevan berarti informasi harus memberikan manfaat bagi pemakainya. Relevansi informasi berbeda-beda untuk tiap-tiap orang satu dengan yang lainnya.

2. Akurat (*Accuracy*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak bisa menyesatkan dan harus jelas mencerminkan maksudnya. Ketidakakuratan dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak atau merubah data-data asli tersebut. Komponen akurat: lengkap (*completeness*), benar (*correctness*), dan aman (*security*).

3. Tepat Waktu (*timeliness*)

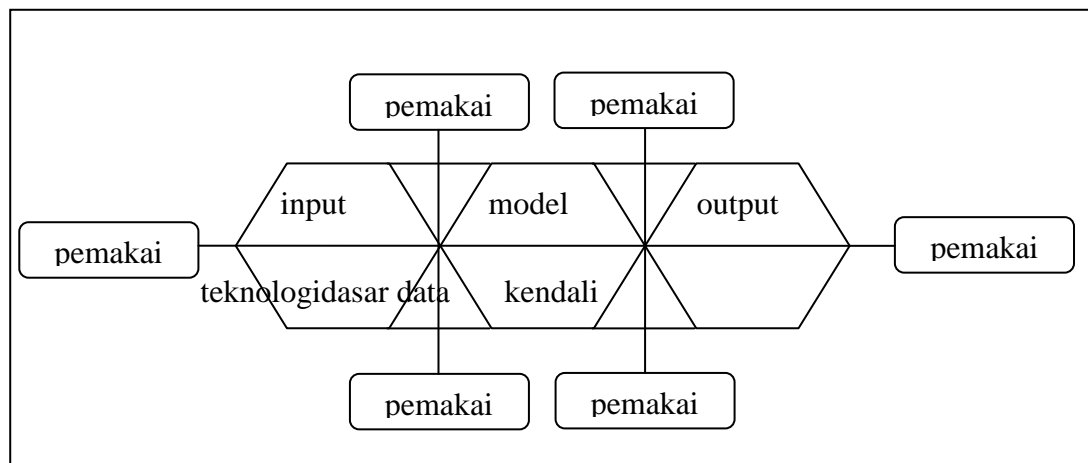
Informasi yang dihasilkan atau dibutuhkan tidak boleh terlambat. Informasi yang terlambat tidak mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar dalam pengambilan keputusan akan berakibat

fatal atau kesalahan dalam keputusan dan tindakan. Kondisi demikian menyebabkan mahalnya suatu nilai informasi, sehingga kecepatan untuk mendapatkan, mengolah, dan mengirimkannya memerlukan teknologi-teknologi terbaru.

2.5. Sistem Informasi

Dalam buku Jogiyanto (2005), sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Robert A. Leitch/K. Roscoe Davis, 1983).

Sistem informasi dapat terdiri dari beberapa komponen yang disebut blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), dan blok teknologi (*technology block*), blok dasar data (*database block*), dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarnya.



Gambar II.3. Blok Sistem Informasi yang Berinteraksi

(Sumber: Jogiyanto, 2005)

Keterangan:

1. Blok Masukan
Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, dapat berupa dokumen-dokumen dasar.
2. Blok Model
 Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data *input* dan data yang tersimpan di dasar data dengan cara tertentu untuk menghasilkan keluaran yang diinginkan.
3. Blok Keluaran
 Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
4. Blok Teknologi
 Teknologi merupakan “kotak alat” (*tool-box*) dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi terdiri dari 2 bagian utama, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*).
 Perangkat lunak berupa program yang membuat perangkat keras dapat bekerja dengan mengintruksikannya untuk memroses sesuai dengan model yang diterapkan. Perangkat keras terdiri dari bermacam-macam alat yang menyediakan dukungan fisik untuk blok-blok lainnya. Untuk blok *input*, disediakan perangkat keras untuk memasukkan data. Untuk blok model guna memroses berdasarkan model dibutuhkan alat proses yang disebut CPU (*Central Processing Unit*). Untuk blok keluaran diperlukan alat untuk menampilkan hasil pengolahan. Untuk blok basis data digunakan teknologi *diskette*, *harddisk*, *magnetic tape* maupun *optical disc* untuk menyimpan data guna keperluan proses lebih lanjut.

5. Blok Basis Data

Basis data merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam dasar data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut DBMS (*Database Management Systems*).

6. Blok Kendali

Agar sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu adanya pengendalian-pengendalian di dalamnya. Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, debu, air, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidakefisienan dan sabotase. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.6 . Pengembangan Sistem

Menurut Jogiyanto (2005), pengembangan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah ada. Sistem yang lama perlu diperbaiki oleh karena beberapa hal, yaitu:

1. Munculnya masalah pada sistem yang lama.
2. Untuk meraih kesempatan.
3. Adanya instruksi.

Proses perancangan atau pengembangan sistem informasi, mulai dari konsep sampai dengan implementasinya disebut dengan istilah *Software Development Life Cycle* atau sering disebut juga *System Development Life*

Cycle(SDLC) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik) (Rosa dan Shalahuddin, 2013).

Menurut Rosa dan Shalahuddin (2013), tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)
Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*)
Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.
3. Perencanaan (*planning*)
Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.
4. Analisis kebutuhan (*requirements analysis*)
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.
5. Desain (*design*)
Mentransformasikan kebutuhan *detail* menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
6. Pengembangan (*development*)
Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean,

pengkompilasian, memperbaiki dan membersihkan program serta peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)
Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (Quality Assurance) dan *user* sehingga menghasilkan laporan analisis pengujian.
8. Implementasi (*implementation*)
Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan luar *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.
9. Operasi dan pemeliharaan (*operations and maintenance*)
Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.
10. Disposisi (*disposition*)
Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

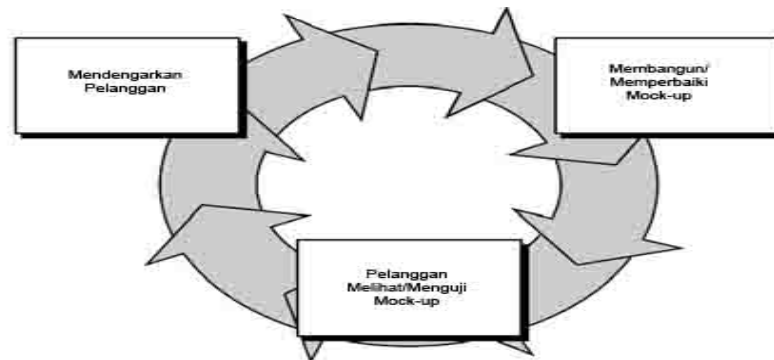
2.6.1. Model *Prototype*

Model *prototype* dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis yang memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak (Rosa dan Shalahuddin, 2013).

Tahapan-tahapan pada model *prototype* (*prototype model*) adalah sebagai berikut (Rosa dan Shalahuddin, 2013):

1. Mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat.
2. Membuat *prototype* agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program *prototype* biasanya merupakan program yang belum jadi.

3. Program *prototype* selanjutnya dievaluasi oleh pelanggan atau *user* sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*.



Gambar II.4 Ilustrasi Model *Prototype*

(Sumber: Rosa dan Shalahuddin, 2013)

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi atau keperluan lain. Sebuah *mock-up* disebut sebagai *prototype* perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak. Iterasi terjadi pada pembuatan *prototype* sampai sesuai dengan keinginan pelanggan atau *user* (Rosa dan Shalahuddin, 2013).

Kelemahan model *prototype* adalah sebagai berikut (Rosa dan Shalahuddin, 2013):

1. *User* dapat sering mengubah-ubah atau menambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan *user* karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.
2. Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan *prototype* dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan *prototype* untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas

perangkat lunak yang kurang baik atau bahkan menyebabkan iteratif tanpa akhir.

Permasalahan dapat terjadi pada model *prototype*, hal ini dapat diatasi dengan melakukan perjanjian antara pengembang perangkat lunak dengan pelanggan atau *user* agar model *prototype* hanya digunakan untuk mendefinisikan spesifikasi kebutuhan perangkat lunak, tapi tidak untuk seluruh proses pengembangan seluruh sistem perangkat lunak.

Model *prototype* cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena pelanggan sering merasa kesulitan menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas. Untuk mengantisipasi agar proyek dapat berjalan sesuai dengan waktu dan biaya di awal, maka sebaiknya spesifikasi oleh pengembang dengan pelanggan secara tertulis. Dokumen tersebut akan menjadi patokan agar spesifikasi kebutuhan sistem masih dalam ruang lingkup proyek.

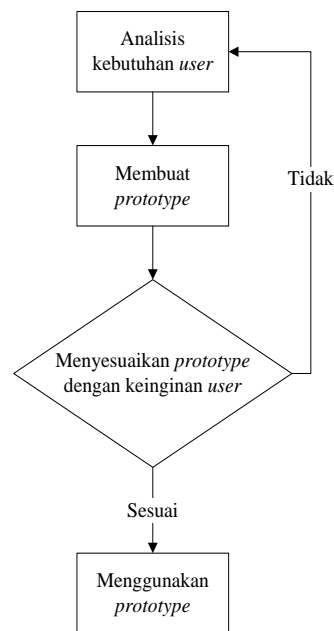
2.6.2 Evolutionary Prototype

Evolutionary prototype yaitu, *prototype* yang secara terus menerus dikembangkan hingga *prototype* tersebut memenuhi fungsi dan prosedur yang dibutuhkan oleh sistem. Pada pendekatan evolusioner, suatu *prototype* dibangun berdasarkan pada kebutuhan dan pemahaman secara umum. *prototype* kemudian diubah dan dievolusikan dari pada dibuang. *prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *Evolutionary prototype* (McLeod, 2008).

Tahapan-tahapan model *Evolutionary Prototype* menurut McLeod (2008) adalah sebagai berikut:

1. Analisis kebutuhan *user*, pengembang dan *user* atau pemilik sistem melakukan diskusi dimana *user* atau pemilik sistem menjelaskan kepada pengembang tentang kebutuhan sistem yang mereka inginkan.
2. Membuat *prototype*, pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh *user* atau pemilik sistem.

3. Menyesuaikan *prototype* dengan keinginan *user* atau pemilik sistem, pengembang menanyakan kepada *user* atau pemilik sistem tentang *prototype* yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.
4. Menggunakan *prototype*, sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.



Gambar II.5 *Evolutionary Prototype Model*

(Sumber: McLeod, 2008)

2.7. Pengertian Quality Assurance

Quality assurance (QA) adalah seluruh rencana dan tindakan penting yang menyediakan kepercayaan untuk memuaskan kebutuhan kualitas. Dimana kebutuhan merupakan kebutuhan pelanggan. Quality assurance (QA) biasanya membutuhkan evaluasi secara terus menerus dan biasanya digunakan sebagai alat manajemen (Elliot, G., 1993).

Quality assurance (QA) bertujuan meningkatkan proses pengembangan dan testing agar tidak terjadi kesalahan pada produk. Manfaat Quality assurance (QA) untuk memastikan proses pengembangan produk berjalan dengan baik untuk menghasilkan produk dan dapat mencapai suatu kualitas (*defect*

prevention).

Concepts of quality assurance (konsep QA) adalah sebuah tindakan untuk pencegahan suatu cacat terhadap suatu produk. *Quality assurance* (QA) menyediakan metode yang efisien untuk mengumpulkan informasi tentang kualitas produk saat operasi.

Quality assurance (QA) berfungsi memelihara manajemen agar dapat mengambil tindakan yang tepat untuk mencapai hal-hal berikut:

1. Meningkatkan kualitas
2. Meningkatkan lingkungan kerja
3. Mendistribusikan informasi teknis

Quality assurance (QA) berperan sebagai *analyst* untuk memperbaiki kualitas dari produk, *Quality assurance* (QA) juga masih memiliki peran untuk memahami spesifikasi pelanggan dan spesifikasi standar yang berhubungan dengan produk, kemudian membuat atau menentukan cara untuk melakukan tindakan inspeksi yang berupa prosedur dan membuat hasil dokumentasi hasil tindakan inspeksi sehingga dapat diketahui *Quality assurance* (QA) bersifat *proactive*.

2.8. Pengertian Quality Inspection

Quality inspection (QI) adalah pemeriksaan kualitas yang dilakukan dengan cara observasi secara langsung. Kata *inspection* atau inspeksi kita aplikasikan kedalam pengendalian kualitas yang diartikan pemeriksaan suatu produk yang dihasilkan apakah sudah sesuai aturan yang ditetapkan (Salmah, 2006).

Quality inspection (QI) merupakan metode paling umum yang digunakan oleh perusahaan manufaktur untuk mencapai kualitas produk sesuai standarisasi produk. Jika produk yang dihasilkan tidak sesuai dengan ketentuan standard spesifikasi maka produk akan ditolak oleh pihak yang bertanggung jawab dan harus melakukan tindakan perbaikan (*corrective Countermeasure*).

Tujuan *Quality inspection* (QI) dalam pengendalian kualitas:

1. Untuk mendeteksi dan menghilangkan bahan baku yang cacat sebelum masuk ke proses produksi.
2. Untuk mendeteksi produk cacat dan produk yang berkualitas rendah terkirim ke pelanggan.
3. Untuk memberikan pemberitahuan kepada manajemen sebelum masalah kualitas menjadi serius sehingga manajemen dapat mengambil tindakan yang diperlukan.
4. Untuk mencegah keterlambatan pengiriman yang dikarenakan masalah kualitas dan mengurangi keluhan dari pelanggan.
5. Untuk meningkatkan kualitas dan realibilitas produk.

Manfaat Quality inspection (QI):

1. Membedakan unit produk yang baik dan unit produk yang cacat.
2. Untuk mengetahui perubahan pada proses produksi.
3. Untuk mengetahui apakah proses produksi berada atau mendekati spesifikasi.
4. Untuk menilai kualitas produk.
5. Untuk mengukur ketepatan alat ukur diproduksi.

Jenis-jenis Quality inspection (QI) dalam pengendalian kualitas:

1. *Floor inspection* adalah inspeksi yang dilakukan dalam proses produksi.
2. *Centralized inspection* adalah inspeksi yang dilakukan pada lokasi tertentu atau terpusat pada tempat yang ditentukan.
3. *Functional inspection* adalah inspeksi terhadap fungsional produk.
4. *First piece inspection* adalah inspeksi yang dilakukan terhadap unit pertama.
5. *Pilot piece inspection* inspeksi yang dilakukan terhadap produk baru ataupun model-model baru.
6. *Final inspection* adalah inspeksi yang dilakukan pada produk jadi.

2.9. Definisi Daily Defect

Defect diartikan dalam bahasa Indonesia adalah cacat. *Daily* diartikan dalam bahasa Indonesia adalah sehari-hari. *Daily defect* adalah sejumlah cacat yang muncul dari atribut produk. Biasanya muncul dari spesifikasi produk. Sejalan dengan berkembangnya jaman, kegiatan pemeriksaan tidak lagi dilakukan sendiri, tetapi diserahkan kepada orang yang ahli dan kompeten atau dilakukan oleh pihak yang mencatat kerusakan perhari pada *Daily defect* disebut dengan *check man*, kata *defect* diterjemahkan sebagai jumlah produk yang cacat. Berkenaan dengan masalah cacat (*defect*) produk yang cacat (*defective product*) menyebabkan produsen harus bertanggung jawab.

2.10. Flowchart

Flowchart adalah bagan yang menunjukkan alir di dalam program atau prosedur sistem secara logika. Bagan alir digunakan sebagai alat bantu komunikasi dan dokumentasi. Notasi yang digunakan dalam suatu *Flowmap* merupakan penggabungan notasi *Flowchart* program.

Dalam menggambar suatu bagan alir, analisis sistem atau pemrograman dapat mengikuti pedoman-pedoman sebagai berikut:

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misal:
 - a. "Persiapkan" dokumen
 - b. "Hitung" gaji
5. Masing-masing kegiatan di dalam bagan alir harus di dalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung di tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol-simbol bagan alir yang standar.





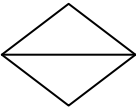
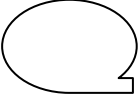

2.10.1. Bagan Alir atau *Flowchart*


Flowchart adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Jogiyanto, 2005). Terdapat beberapa jenis bagan alir (*flowchart*) antara lain sebagai berikut:

1. Bagan alir sistem (*systems flowchart*)

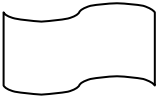
Systems flowchart merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. Bagan ini menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. *Systems flowchart* digambar dengan menggunakan simbol-simbol yang tampak pada tabel II.1.

Tabel II.1 Simbol Bagan Alir Sistem (*Systems Flowchart*)

Simbol	Nama	Keterangan
	Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> , baik untuk proses manual, mekanik atau computer
	Kegiatan manual	Menunjukkan pekerjaan yang dilakukan secara manual
	Kartu plong	Menunjukkan <i>input</i> atau <i>output</i> yang menggunakan kartu plong (<i>punched card</i>)
	Proses	Menunjukkan kegiatan proses dari operasi program komputer
	Pengurutan <i>offline</i>	Menunjukkan proses pengurutan data di luar proses computer
	Pita magnetic	Menunjukkan <i>input</i> atau <i>output</i> yang menggunakan pita magnetic
	<i>Hard disk</i>	Menunjukkan <i>input</i> atau <i>output</i>

		yang menggunakan <i>hard disk</i>
	Disket	Menunjukkan <i>input</i> atau <i>output</i> yang menggunakan disket

Tabel II.1 Simbol Bagan Alir Sistem (*Systems Flowchart*)(Lanjutan)

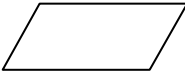
Simbol	Nama	Keterangan
	Pita kertas berlubang	Menunjukkan <i>input</i> atau <i>output</i> yang menggunakan pita kertas berlubang

(Sumber: Jogiyanto, 2005)

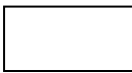
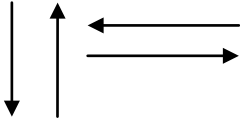
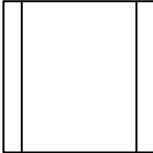
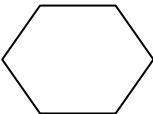
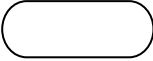
2. Bagan alir dokumen (*document flowchart*)
 Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusan. Bagan alir dokumen menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem (*systems flowchart*).
3. Bagan alir skematik (*schematic flowchart*)
 Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaan bagan alir skematik dari bagan alir sistem adalah selain menggunakan simbol-simbol bagan alir sistem digunakan pula gambar-gambar komputer dan peralatan lain. Maksud dari penggunaan gambar tersebut adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar mudah untuk dipahami, tetapi sulit serta lama pada pengerjaan.
4. Bagan alir program (*program flowchart*)
 Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dibuat

dengan menggunakan simbol yang terdapat pada tabel II.2.

Tabel II.2 Simbol Bagan Alir Program (Program *Flowchart*)

Simbol	Nama	Keterangan
	<i>Input/ output</i>	Simbol <i>input / output</i> digunakan untuk mewakili data <i>input/ output</i>

Tabel II.2 Simbol Bagan Alir Program (Program *Flowchart*)


Simbol	Nama	Keterangan
	Proses	Simbol proses digunakan untuk mewakili suatu proses
	Garis alir	Simbol garis alir digunakan untuk menunjukkan arus dari proses
	Proses terdefinisi	Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rincian ditunjukkan di tempat lain
	Persiapan	Simbol persiapan digunakan untuk memberi nilai awal suatu sasaran
	Titik terminal	Simbol titik terminal digunakan untuk menunjukkan awal dan akhir dari suatu proses

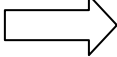

(Sumber: Jogiyanto, 2005)

5. Bagan alir proses (*process flowchart*)



Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan di teknik industri. Bagan alir ini juga berguna bagi analisis sistem untuk menggambarkan proses dalam suatu prosedur. Bagan alir proses memiliki simbol yang terdapat pada tabel II.3.

Tabel II.3 Simbol Bagan Alir Proses (*Process Flowchart*)

Simbol	Nama	Keterangan
	<i>Operation</i>	Menunjukkan suatu operasi

	<i>Movement</i>	Menunjukkan suatu pemindahan
	<i>Storage</i>	Menunjukkan suatu simpanan

Tabel II.3 Simbol Bagan Alir Proses (*Process Flowchart*)(Lanjutan)

Simbol	Nama	Keterangan
	<i>Inspection</i>	Menunjukkan suatu inspeksi
	<i>Delay</i>	Menunjukkan suatu penundaan

(Sumber: Jogiyanto, 2005)

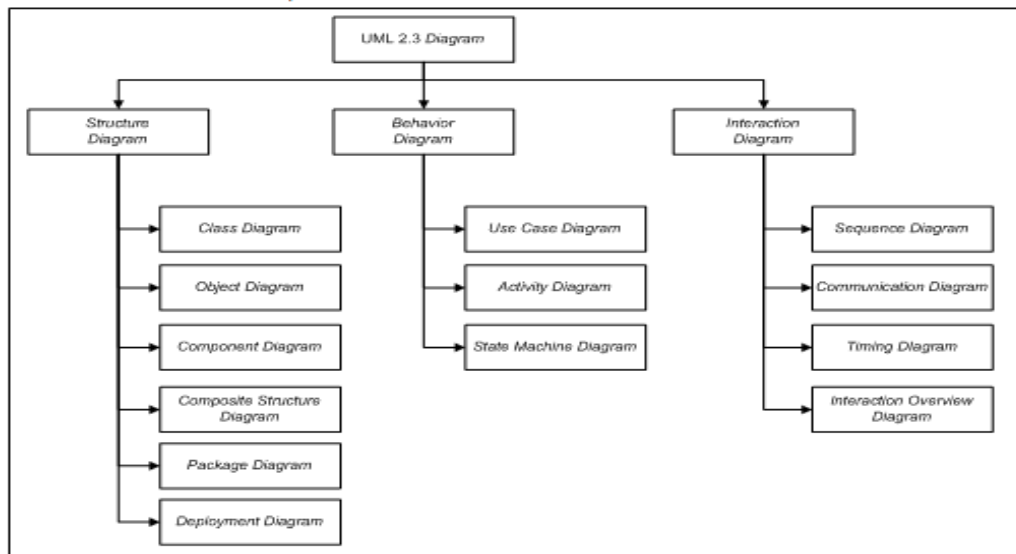
2.11. *Unified Modelling Language*(UML)

Menurut Munawar (2005), UML (*Unified Modeling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain.

Tujuan dari pemodelan UML adalah sebagai berikut:

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktik-praktik terbaik yang terdapat dalam pemodelan.

UML II.3 terdiri dari 13 macam *Diagram* yang dikelompokkan dalam 3 kategori. Pembagian kategori tersebut dapat dilihat pada gambar berikut ini (Rosa dan Shalahuddin, 2013):



Gambar II.6 Klasifikasi *Diagram* UML

(Sumber: Rosa dan Shalahuddin, 2013)

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

1. *Structure Diagrams*, yaitu kumpulan *Diagram* yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior Diagrams*, yaitu kumpulan *Diagram* yang digunakan untuk menggambarkan proses sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction Diagrams*, yaitu kumpulan *Diagram* yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar sub sistem pada suatu sistem.

2.11.1. Bagian-bagian UML

Bagian-bagian utama dari UML yaitu:

1. *View*
View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah *Diagram*. Beberapa jenis *view* dalam UML antara lain:

a. *Use Case View*

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya. *View* ini digambarkan dalam *Use Case Diagrams* dan kadang-kadang dengan *Activity Diagrams*. *View* ini digunakan terutama untuk pelanggan, perancang, pengembang, dan penguji sistem.

b. *Logical View*

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object, dan relationship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *View* ini digambarkan dalam *class Diagrams* untuk struktur statis dan dalam *state, sequence, collaboration, dan Activity Diagram* untuk model dinamis. *View* ini digunakan untuk perancang dan pengembang.

c. *Component View*

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lain dari *code module* diperlihatkan dengan struktur dan ketergantungan alokasi sumber daya komponen dan informasi administratif lain. *View* ini digambarkan dalam *component view* dan digunakan oleh pengembang.

d. *Concurrency View*

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam *diagram* dinamis (*state, sequence, collaboration, dan activity diagrams*) dan *diagram* implementasi (*component dan deployment diagrams*) serta digunakan untuk pengembang, pengintegrasian, dan penguji.

e. *Deployment View*

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan lainnya. *View* ini

digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang, pengintegrasian, dan pengujian.

2. Diagram

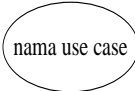
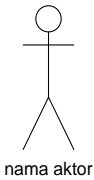
Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah *diagram* merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis *diagram* antara lain (akakom.ac.id, 2015):

a. Use Case Diagram


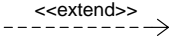
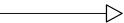
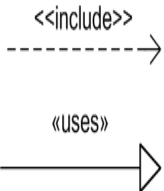
Use case diagram adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *Use case symbol* namun dapat juga dilakukan dalam *Activity diagram*.

Use case digambarkan hanya yang dilihat dari luar *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bukan bagaimana fungsi yang ada di dalam sistem. *Use case diagram* digambar dengan menggunakan simbol-simbol yang tampak pada Tabel II.4

Tabel II.4 Simbol-Simbol *Use Case Diagram*

Simbol	Nama	Deskripsi
	<i>Use case</i>	Fungsionalitas yang disediakan oleh sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>Use Case</i> .
	Aktor/ <i>actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambaran orang, tapi aktor belum tentu merupakan orang.

Tabel II.4 Simbol-Simbol *Use Case Diagram*(Lanjutan)

Simbol	Nama	Deskripsi
		Biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	Asosiasi/ <i>association</i>	Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> atau <i>Use Case</i> memiliki interaksi dengan aktor.
	Ekstensi/ <i>extend</i>	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan dapat berdiri sendiri tanpa <i>Use Case</i> tambahan. Seperti prinsip <i>inheritance</i> pada pemrograman berorientasi objek. Arah panah mengarah pada <i>Use Case</i> yang ditambahkan.
	Generalisasi/ <i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>Use Case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
	Menggunakan <i>/Include/uses</i>	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> .

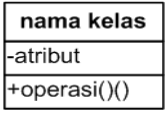
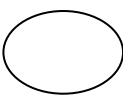


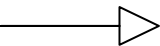

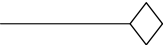
(Sumber: Rosa dan Shalahuddin, 2013)

b. *Class Diagram*

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

metode atau operasi. *diagram* kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam *diagram* kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. *class diagram* digambar menggunakan simbol-simbol yang tampak pada Tabel II.5.

Tabel II.5 Simbol-Simbol *Class Diagram*

Simbol	Nama	Keterangan
	Kelas / <i>class</i>	Kelas pada struktur sistem.
	Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	Asosiasi/ <i>association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
	Asosiasi berarah	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus).
	Kebergantungan <i>/despency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas.
	<i>Bidirectional</i> <i>Association</i>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

(Sumber: Rosa dan Shalahuddin, 2013)

Pada relasi di dalam *class diagram* terdapat suatu penanda yang disebut *multiplicity*. *Multiplicity* mengindikasikan berapa banyak obyek dari suatu kelas terelasi ke obyek lain. *Multiplicity* digambar dengan menggunakan simbol-simbol yang tampak pada Tabel II.6.

Tabel II.6 *Multiplicity*



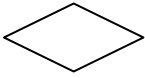

Simbol	Keterangan
0..1	Nol atau satu
1	Satu
0..*	Nol atau lebih
1..*	Satu atau lebih
N	Hanya n (dengan $n > 1$)
0..n	Nol sampai n (dengan $n > 1$)
1..n	Satu sampai n (dengan $n > 1$)

(Sumber: *Ftass.telkomuniversity.ac.id*, 2015)


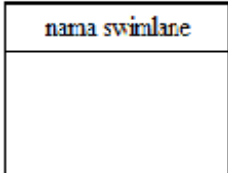
c. *Activity Diagram*

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa *diagram* aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem *activity diagram* digambar dengan menggunakan simbol-simbol yang tampak pada Tabel II.7.

Tabel II.7 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Keterangan
	<i>Initial Node</i>	Status awal aktivitas sistem, sebuah <i>Diagram</i> aktivitas memiliki sebuah status awal.
	<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas yang digabungkan menjadi satu.

Tabel II.7 Simbol-Simbol *Activity Diagram*(lanjutan)

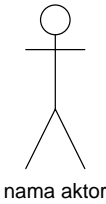
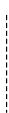
Simbol	Nama	Keterangan
	<i>Activity Final Node</i>	Status akhir yang dilakukan sistem, sebuah <i>Diagram</i> aktivitas memiliki sebuah status akhir.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi.

(Sumber: Rosa dan Shalahuddin, 2013)


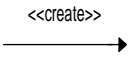

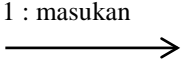
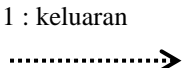
d. *Sequence Diagram*

Diagram sekuen 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. Membuat *Diagram* sekuen juga dibutuhkan untuk melihat skenario yang ada pada *Use case*. *Sequence diagram* digambar dengan menggunakan simbol-simbol yang tampak pada Tabel II.8

Tabel II.8 Simbol-Simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	Aktor	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 gambaran orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	Garis hidup/ <i>lifeline</i>	Menyatakan kehidupan suatu objek.

Tabel II.8 Simbol-Simbol *Sequence Diagram*(Lanjutan)

Simbol	Nama	Keterangan
	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif.
	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi/metode.
	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkandata atau masukaninformasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
	Pesantipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.

(Sumber: Rosa dan Shalahuddin, 2013)

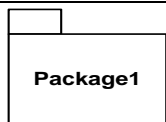

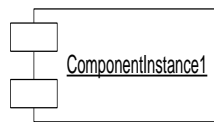


e. *Collaboration Diagram*

Collaboration diagram menggambarkan interaksi antar objek seperti *Sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*.

f. *Component Diagram*

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*.

Tabel II.9 Simbol-Simbol *ComponentDiagram*

Simbol	Nama	Keterangan
	<i>Package</i>	Merupakan sebuah kumpulan dari satu atau lebih komponen
	<i>Link</i>	Relasi antar objek
	<i>Component</i>	Komponen system
	<i>Dependency</i>	Hubungan suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	<i>Interface</i>	Sebagai antarmuka komponen

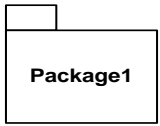
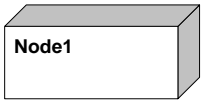
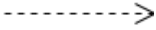

(Sumber: Rosa dan Shalahuddin, 2013)

g. *Deployment Diagram*

Deployment diagram merupakan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal yang bersifat fisik. Sebuah *node* adalah server, *workstation*, atau piranti

keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya.

Tabel II.10 Simbol-Simbol *Deployment Diagram*

Simbol	Nama	Keterangan
	<i>Package</i>	Merupakan sebuah kumpulan dari satu atau lebih komponen
	<i>Node</i>	Perangkat keras dan perangkat lunak
	<i>Dependency</i>	Hubungan pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Link</i>	Relasi antar objek

(Sumber: Rosa dan Shalahuddin, 2013)

2.12. Kamus Data (*Data Dictionary*)

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.

2.13. Hierarchy Plus Input-Proses-Output (HIPO)

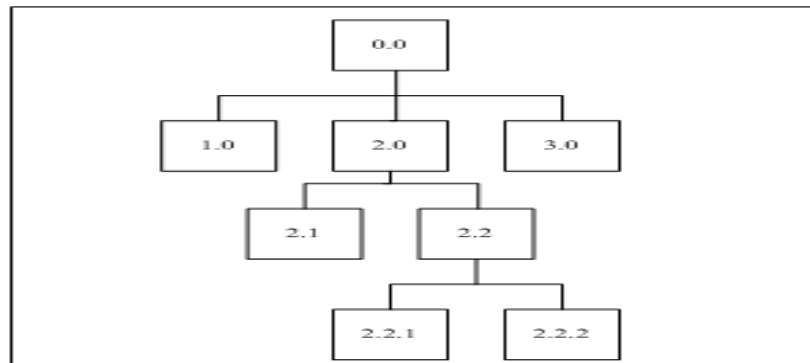
Hierarchy Plus-Input-Process-Output(HIPO) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program, akan tetapi sekarang HIPO banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO mempunyai sasaran utama sebagai berikut:

1. Untuk menyediakan suatu struktur guna memahami fungsi-fungsi dari sistem.
2. Untuk lebih menekankan fungsi-fungsi yang harus diselesaikan oleh program, bukannya menunjukkan statemen program yang digunakan untuk melaksanakan fungsi tersebut.
3. Untuk menyediakan penjelasan yang jelas dari *input* yang harus digunakan dan *output* yang harus dihasilkan oleh masing-masing fungsi pada tiap-tiap tingkatan dari *diagram-diagram* HIPO.
4. Untuk menyediakan *output* yang tepat dan sesuai dengan kebutuhan kebutuhan pemakai.

HIPO dapat digunakan sebagai alat pengembangan sistem dan teknik dokumentasi program, fungsi-fungsi dari sistem digambarkan oleh HIPO dalam tiga tingkatan (Jogiyanto, 2005). Untuk masing-masing tingkatan digambarkan dalam bentuk *diagram* tersendiri, dengan demikian HIPO menggunakan tiga macam *diagram* untuk masing-masing tingkatannya, yaitu sebagai berikut:

1. *Visual Table Of Contents* (VTOC)

Visual table of contents 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 sebagai berikut:

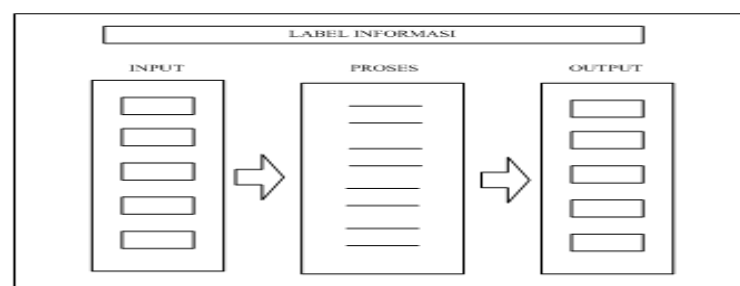


Gambar II.7 Visual Table Of Contents

(Sumber: Jogiyanto, 2005)

2. *Overview Diagram*

Overview diagram menunjukkan secara garis besar hubungan dari *input*, proses dan *output*. Bagian *input* menunjukkan *item-item* data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan *item-item* data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *Output* berisi dengan *item-item* data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses. *Overview diagram* ini dapat dilihat pada Gambar II.8.



Gambar II.8 Overview Diagram

(Sumber: Jogiyanto, 2005)

3. *Detail Diagram*

Detail diagram merupakan *diagram* tingkatan yang paling rendah pada *diagram HIPO*. *Detail diagram* berisi elemen-elemen dasar dari paket yang menggambarkan secara rinci kerja dari fungsi.

2.14. Hypertext Preprocessor (PHP)

Hypertext preprocessor (PHP) diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. PHP terus mengalami perkembangan dan perubahan hingga saat ini dalam berbagai versi. PHP adalah bahasa *scripting* yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang pengguna berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke *browser* hanya hasilnya saja (Widigdo, 2003). Dalam PHP setiap nama variabel diawali tanda dolar (\$). Misalnya nama variabel dalam PHP ditulis dengan \$. Jenis suatu variabel ditentukan pada saat jalannya program dan tergantung pada konteks yang digunakan. Adapun kelebihan dari penggunaan PHP antara lain (Saputra, 2012):

1. Bahasa pemrograman PHP dapat diaplikasikan ke berbagai *platform OS* (*Operating System*).
2. Bersifat *open source*.
3. Mudah dipelajari dan digunakan.
4. PHP didukung oleh beberapa macam *webserver*, seperti Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
5. PHP mendukung beberapa *database*, sistem basis data yang dapat didukung dengan PHP seperti MySQL, PostgreSQL, mSQL, Informix, SQL Sever dan Oracle.

2.15. MySQL

MySQL merupakan salah satu jenis program *database server* yaitu sebuah program yang berfungsi mengolah, menyimpan dan memanipulasi data di *server*. MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GNU *General Public Licence* (GPL). Setiap pengguna dapat secara bebas menggunakan MySQL. MySQL merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, yaitu SQL (*Structured Query Language*), merupakan sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan

mudah. Berikut merupakan beberapa kelebihan yang dimiliki MySQL (Saputra, 2012):

1. Berifat *opensource*, yang memiliki kemampuan untuk dikembangkan lagi.
2. Menggunakan bahasa SQL (*Structure Query Language*), yang merupakan standar bahasa dunia dalam pengolahan data.
3. *Upper performance* dan *reliable*, pemrosesan *database* sangat cepat.
4. Mudah dipelajari (*easy of use*).
5. Memiliki dukungan *support (group)* pengguna MySQL.
6. Dapat berjalan di berbagai sistem operasi.
7. Bersifat *multiuser*, MySQL dapat digunakan oleh berbagai pengguna dalam waktu yang bersamaan tanpa mengalami konflik.

Database MySQL memiliki beberapa tipe yang dapat digunakan sesuai kebutuhan pengguna dalam merancang tabel. Berikut merupakan beberapa tipe data yang dimiliki database MySQL (Unair.ac.id, 2015):

Tabel II.11 Beberapa tipe data pada MySQL

Tipe Data	Keterangan
<i>Char</i>	String yang memiliki lebar tetap. Jangkauan nilai dari 1 hingga 255 karakter.
<i>Varchar</i>	String dengan lebar bervariasi. Jangkauan nilai dari 1 hingga 255 karakter.
<i>Tinyblob, Tinytext</i>	Sebuah blob (semacam catatan) atau <i>text</i> dengan lebar maksimum 255 ($2^8 - 1$) karakter.
<i>Blob, Text</i>	Sebuah blob atau <i>text</i> dengan lebar maksimum 65535 ($2^{16} - 1$) karakter.
<i>Tinyint</i>	<i>Integer</i> yang sangat kecil jangkauan nilainya, yaitu -128 hingga 127. Jangkauan <i>unsigned</i> adalah 0 hingga 255.
<i>Mediumint</i>	<i>Integer</i> tingkat menengah. Jangkauan nilainya adalah -8388608 hingga 8388607. Jangkauan <i>unsigned</i> adalah 0 hingga 16777215.
<i>Int</i>	<i>Integer</i> yang berukuran normal. Jangkauan nilainya adalah -2147483648 hingga 2147483647. Jangkauan <i>unsigned</i> adalah 0 hingga 4294967295.

Tabel II.11 Beberapa tipe data pada MySQL (Lanjutan)

Tipe Data	Keterangan
<i>Float</i>	Bilangan <i>floating-point</i> presisi tunggal. Tidak dapat bersifat <i>unsigned</i> . Nilai yang diijinkan adalah -3.402823466E+38 hingga 1.175494351E-38 untuk nilai negatif, 0 dan 1.175494351E-38 hingga 3.402823466E+38 untuk nilai positif.
<i>Decimal</i> (M, D)	Jangkauan nilai dari <i>decimal</i> sama dengan <i>double</i> , tetapi juga tergantung dai nilai atribut M dan D yang disertakan. Jika D tidak diisi, akan dianggap 0. Jika M tidak diisi maka akan dianggap 10.
<i>Date</i>	MySQL menampilkan tanggal dalam format 'YYYY-MM-DD'. Jangkauan nilainya adalah '1000-01-01' hingga '9999-12-31'.
<i>Time</i>	MySQL menampilkan <i>TIME</i> dalam format 'HH:MM:SS'. Jangkauannya adalah '-838:59:59' hingga '838:59:59'.

(Sumber: unair.ac.id, 2015)

BAB III

METODOLOGI PENELITIAN

3.1. Metodologi Penelitian

Metodologi penelitian adalah sekumpulan peraturan, kegiatan dan prosedur yang digunakan dalam melakukan penyelidikan yang sistematis untuk meningkatkan sejumlah pengetahuan. Metodologi penelitian juga merupakan suatu usaha yang sistematis dan terorganisasi untuk menyelidiki masalah tertentu yang membutuhkan jawaban.

3.2. Pengumpulan Data

Metode pengumpulan data merupakan teknik atau cara yang dilakukan untuk mengumpulkan data. Metode menunjuk suatu cara pengumpulan data dari berbagai sumber dan bentuk yang akan digunakan dalam keperluan menganalisis sistem untuk perancangan sistem usulan.

Data yang diperoleh dari responden secara langsung yang dikumpulkan melalui survei lapangan dengan menggunakan teknik pengumpulan data. Teknik pengumpulan data merupakan faktor penting demi keberhasilan penelitian. Teknik yang digunakan antara lain:

1. **Observasi**

Tahap ini merupakan tahap untuk mengumpulkan data dengan cara mengamati secara langsung keadaan maupun kegiatan dalam perusahaan yang kemudian dilakukan pencatatan informasi terhadap objek yang dianggap perlu. Observasi ini dilakukan di Departemen Produksi di bagian Trimming 1 dan di Departemen Teknik di bagian Quality Inspection(QI) pada PT Krama Yudha Ratu Motor (KRM). Pada tahap ini juga dilakukan pengamatan langsung terhadap pengguna sistem dan pengamatan langsung terhadap sistem yang sedang berjalan.

2. Wawancara

Pada tahap ini penulis melakukan wawancara dengan para pengguna sistem tersebut untuk memperoleh keterangan mengenai alur kerja *Daily defect* unit kendaraan mobil tipe TD berfungsi untuk melaporkan kecacatan pada kendaraan dalam periode satu hari.

3. Studi Pustaka.

Studi pustaka yaitu metode pengumpulan data dengan melakukan pencarian referensi yang berkaitan dengan pemrograman PHP dan *databaseMySQL* dari berbagai referensi.

3.3. Metode Pengembangan Sistem

Perancang suatu sistem informasi dibutuhkan metode pengembangan sistem agar dapat menuntun analisis sistem untuk menghasilkan sistem yang standar. Metodologi yang digunakan dalam perancangan aplikasi pembuatan dokumen Countermeasure dengan metode *prototyping model* dan menggunakan *evolutionary prototype*.

Metode *prototyping model* merupakan sebuah metode pengembangan sistem yang dapat memberikan gambaran atau ide bagi seorang analis sistem untuk menyajikan gambaran secara lengkap tentang sistem. Alasan digunakannya model ini dalam Rancang Bangun Pelaporan Sistem Informasi *Daily Defect Unit Kendaraan Mobil Tipe TD* adalah sebagai berikut:

1. Pengumpulan kebutuhan

Pengembang dan bagian Trimming 1 pada PT Krama Yudha Ratu Motor (KRM) mendefinisikan perangkat lunak, mengidentifikasi semua kebutuhan dan garis besar sistem yang akan dibuat.

2. Membuat *Prototype*

Pengembang membuat *prototype* sementara yaitu dengan membuat *input* dan format *output*.

3. **Evaluasi *Prototype***
Pengembang menanyakan kepada bagian Trimming 1 mengenai *prototype* yang sudah dibuat, apakah sesuai atau tidak dengan kebutuhan sistem.
4. **Menggunakan *Prototype***
Mempersiapkan kebutuhan sistem dan konfigurasi sistem, *user*, *password* dan basis data.

3.4. Kerangka Penelitian

Dalam penelitian yang dibahas dalam tugas akhir ini, dilakukan langkah-langkah atau tahapan dalam penelitian yang ada pada Gambar III.1 sesuai dengan metodologi penelitian yang digunakan dalam penelitian tersebut. Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

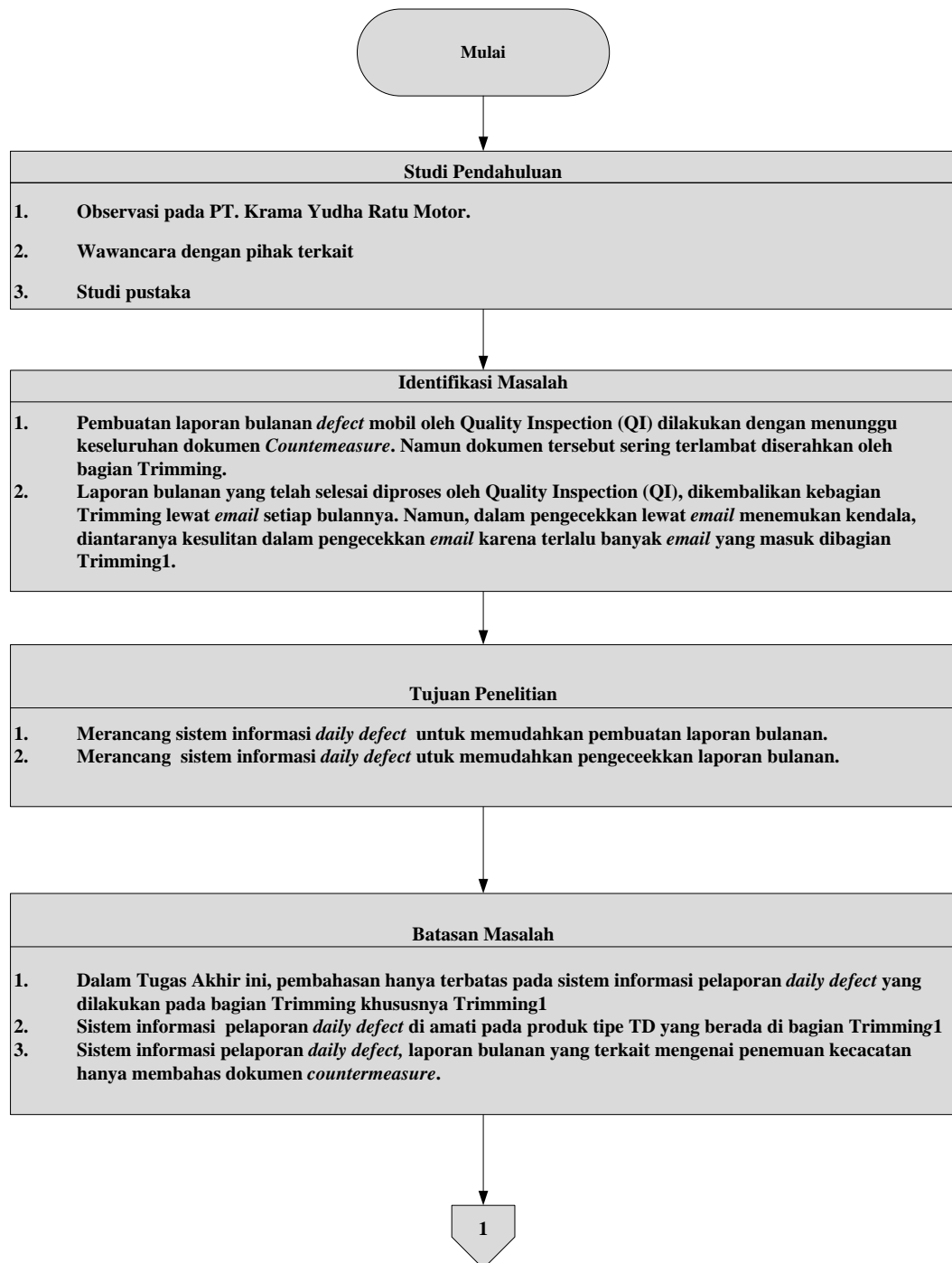
1. **Identifikasi Masalah**
Mengidentifikasi dan merumuskan masalah yang terjadi selama penelitian pada sistem *daily defect*.
2. **Tujuan Penelitian**
Tujuan merupakan arah dari suatu penelitian. Tujuan penelitian harus disesuaikan dengan identifikasi masalah. Jika permasalahan sudah terjawab maka dapat dikatakan tujuan penelitian sudah tercapai.
3. **Batasan Masalah**
Batasan masalah adalah suatu proses pembatasan terhadap cakupan suatu masalah yang diamati. Batasan masalah dilakukan agar penelitian lebih terfokus dan tidak keluar dari jalur cakupan masalah.
4. **Identifikasi Kebutuhan**
Menganalisis kebutuhan pengguna terhadap sistem yang akan dirancang.
5. **Perancangan *Prototype***
Merancang *prototype* sesuai dengan analisis dan perancangan yang dilakukan yaitu menggunakan PHP 5.4.7 dan MySQL 5.5.27 sebagai basis data.

6. Menggunakan *Prototype*

Berdasarkan *prototype* dari sistem yang diusulkan, dapat diketahui sistem usulan tersebut sudah sesuai dengan keinginan pengguna atau belum. Jika belum, maka harus membuat kembali *prototype*. Namun jika sudah sesuai, maka *prototypetersebut* dapat digunakan. Dan hal selanjutnya yang dilakukan adalah mempersiapkan kebutuhan *hardware* dan *software*.

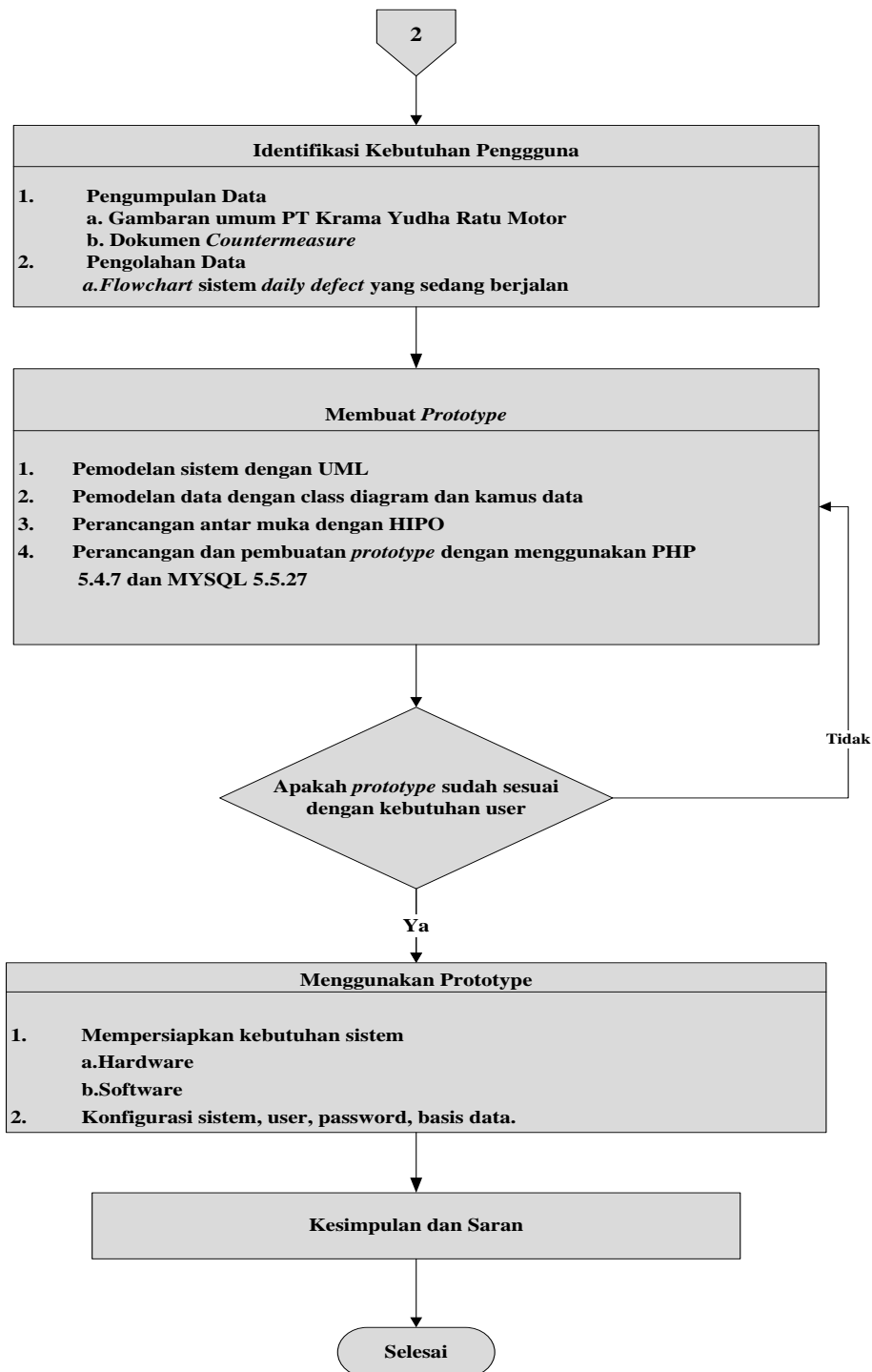
7. Kesimpulan dan Saran

Memberikan kesimpulan mengenai sistem yang diusulkan dan aplikasi yang dirancang serta memberikan saran kepada PT Krama Yudha Ratu Motor dan masukan kepada pengembang selanjutnya.



Gambar III.1 Kerangka Penelitian

(Sumber: Pengolahan Data, 2017)



Gambar III.1 Kerangka Penelitian (lanjutan)
(Sumber: Pengolahan Data, 2017)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1. Hubungan Kerja Sama PT Krama Yudha Ratu Motor

PT Krama Yudha Ratu Motor (KRM) merupakan pabrik perakitan kendaraan bermotor Mitshubshi jenis niaga yang berdiri pada tanggal 1 juni 1973. PT. Krama Yudha Ratu Motor (KRM) ini merupakan bagian dari Krama Yudha Mitshubshi Group (KYMG) yang berstatus Penanaman Modal Dalam Negeri (PMDN) agar selalu percaya terhadap diri sendiri dan kemandirian bangsa, khususnya pengusaha nasional.

Dalam proses produksinya, PT Krama Yudha Ratu Motor (KRM) terikat kontrak kerja dengan PT Krama Yudha Tiga Berlian (KTB). PT Krama Yudha Tiga Berlian (KTB) mengimpor komponen-komponen *Complete Knock Down (CKD)* dari *Mitsubishi Motor Corporation (MMC)*. Komponen-komponen yang telah di terima oleh PT KramaYudha Tiga Berlian (KTB) kemudian diserahkan kepada PT Krama Yudha Ratu Motor (KRM) untuk dirakit.

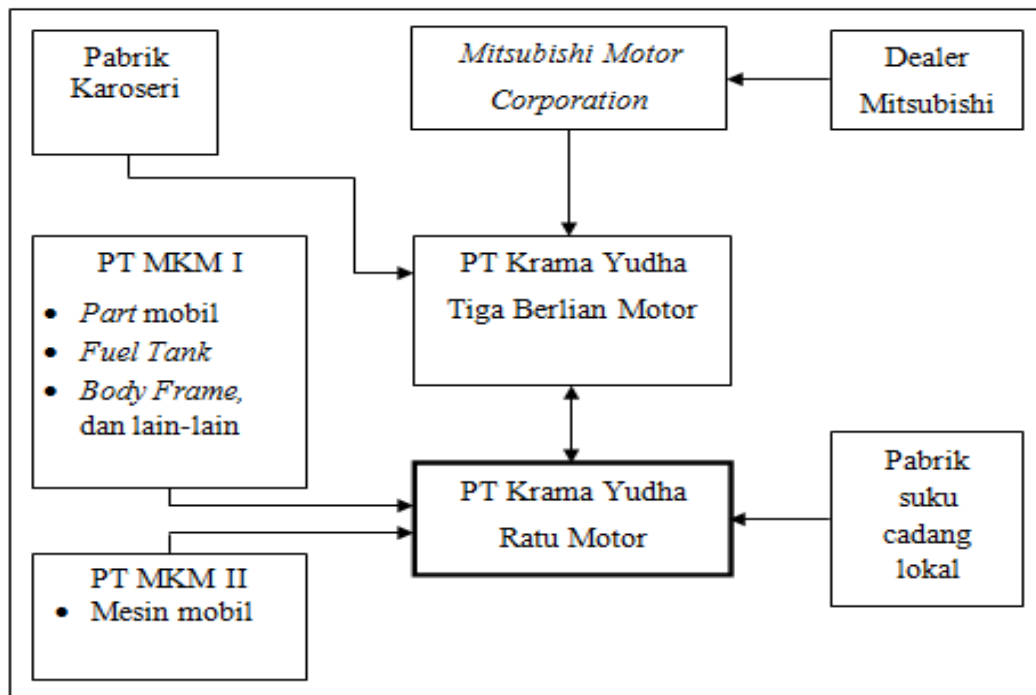
Selain menerima *Complete Knock Down (CKD)* dari *Mitsubishi Motor Corporation (MMC)* melalui PT KramaYudha Tiga Berlian (KTB), PT Krama Yudha Ratu Motor (KRM) juga melakukan pembelian suku cadang lokal untuk komponen-komponen tertentu, seperti ban dari PT Bridgestone, dan lain-lain.

Beberapa tipe mesin disediakan oleh PT Mitsubishi Krama Yudha Motor dan ManufakturI dan rangka mobil disediakan PT Mitsubishi Krama Yudha Motor dan ManufakturII (MKM I dan MKM II).

Dalam proses produksinya, PT Krama Yudha Ratu Motor (KRM) juga memerlukan bahan-bahan penunjang seperti cat, amplas, bahan bakar minyak, kawat las, dan lain-lain. Pembelian bahan-bahan penunjang dilakukan langsung oleh PT Krama Yudha Ratu Motor (KRM) kepada *supplier* yang bersangkutan.

Setelah proses perakitan dan pengujian selesai dilakukan, kendaraan diserahkan kepada PT KramaYudha Tiga Berlian (KTB) untuk didistribusikan

kepada para *dealer* Mitsubishi. Adapun hubungan kerjasama dapat dijabarkan pada Gambar IV.1 berikut ini.

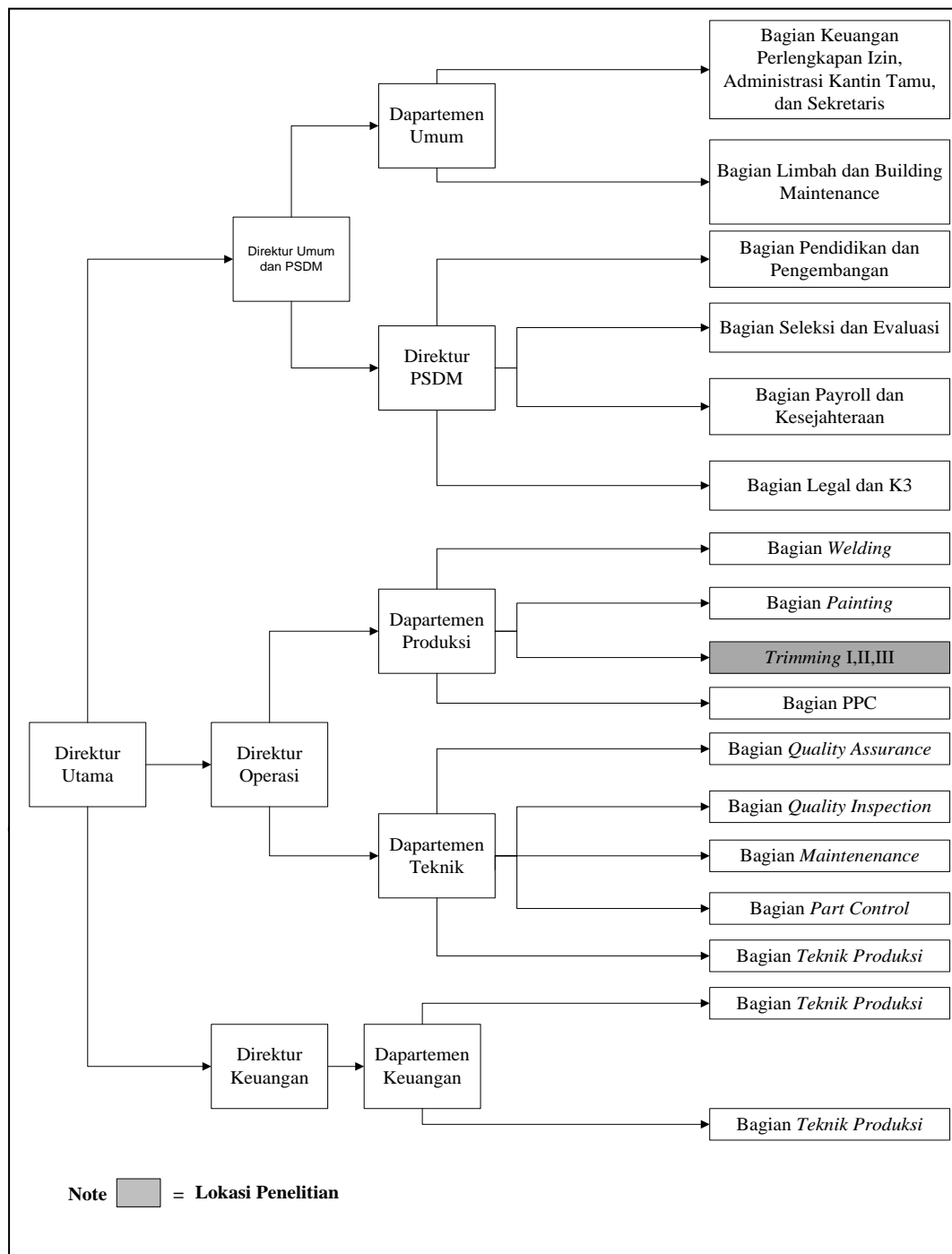


Gambar IV.1 Hubungan Kerja PT Krama Yudha Ratu Motor di dalam dan di luar Krama Yudha Mitsubishi Group

(Sumber: PT Krama Yudha Ratu Motor, 2013)

4.2. Struktur Organisasi

Struktur organisasi memegang peranan yang sangat penting dalam mendukung pencapaian tujuan suatu perusahaan. Karena merupakan cetak biruperusahaan, didalamnya menunjukkan dan menjelaskan pembagian tugas, tanggung jawab maupun wewenang setiap bagian yang ada dalam perusahaan, serta sekaligus sebagai alat koordinasi antar bagian, baik secara vertikal maupun secara horizontal. Struktur organisasi dapat dilihat pada Gambar IV.2.



Gambar IV.2 Struktur Organisasi PT Krama Yudha Ratu Motor

(Sumber: PT Krama Yudha ratu Motor, 2013)

Struktur organisasi PT Krama Yudha Ratu Motor memiliki tugas dan wewenang mengawasi dan mengontrol jalannya operasional perusahaan sesuai dengan tujuan yang telah disepakati sehingga tidak terjadi penyimpangan, terdiri dari:

1. Direktur Utama

Memiliki tugas dan wewenang dalam mengontrol jalannya operasional perusahaan sesuai dengan tujuan yang telah disepakati sehingga tidak terjadi penyimpangan.

2. Direktur Keuangan

Memiliki tugas dan tanggung jawab:

- a. Memegang tanggung jawab mengenai penggunaan keuangan dan surat-surat berharga.
- b. Mengumpulkan dan menganalisa biaya produksi dan pendapatan.
- c. Pengawasan terhadap penerimaan dan pengeluaran atas biaya-biaya produksi dan gaji karyawan.
- d. Menyusun laporan keuangan perusahaan.

Dalam melaksanakan tugasnya, Direktur Keuangan dibantu oleh Departemen Keuangan yang membawahi 3 bagian, yaitu Bagian Keuangan, Bagian Akuntansi.

3. Direktur Operasi

Memiliki tugas dan tanggung jawab:

- a. Mengkoordinir kelancaran jalannya proses produksi
- b. Mengesahkan rencana kerja masing-masing kepala bagian pada departemen teknik, departemen Quality Control dan departemen produksi.
- c. Bertanggung jawab atas hasil jumlah produksi yang telah ditetapkan.

Dalam melaksanakan tugasnya Direktur Operasi dibantu oleh:

- a. Departemen Teknik yang membawahi 5 bagian, yaitu Bagian Teknik produksi, bagian *part control*, bagian *maintenance*, Bagian Quality Inspection (QI), Bagian Quality Assurance (QA).

- b. Departemen Produksi yang membawahi:
 - 1) Bagian PPC
 - 2) Bagian Trimming 1 (*Production Unit SLM & TD*)
 - 3) Bagian Trimming 2 (*Production Unit TD & FMN*)
 - 4) Bagian Trimming 3 (*Production Unit CJM*)
 - 5) Bagian Welding
 - 6) Bagian Painting
4. Direktur Umum dan PSDM
- Direktur umum bertanggung jawab dibagian umum perusahaan.sedangkan PSDM bertanggung jawab mengenai peningkatan kemampuan dari para operator yang berada ditiap-tiap bagian.Dalam melaksanakan Tugasnya mereka dibantu oleh seorang General Manager.
- Dapartemen Umum membawahi:
- a. Bagian keamanan, perlengkapan izin, administrasi kantin tamu dan sekretaris.
 - b. Bagian limbah dan building.
- Dapartemen PSDM membawahi:
- a. Bagian legal dan K3.
 - b. Bagian payroll dan kesejahteraan.
 - c. Bagian seleksi dan evaluasi.
 - d. Bagian pendidikan dan pengembangan.

4.3. Jenis-Jenis Produk

Ada beberapa jenis kendaraan niaga dan mobil *passanger* yang dirakit di PT Krama Yudha Ratu Motor yang kemudian diklasifikasikan menjadi beberapa tipe model kendaraan antara lain sebagai berikut:

Tabel IV.1 Jenis-Jenis Produk PT Krama Yudha Ratu Motor

Tipe	Jenis-Jenis Kendaraan	Spesifikasi
CJM	CJ-M G-PU	78 PS
	CJ-M G-MB	78 PS
SL	SL D-PU	72 PS
	SL D-MB	72 PS
<i>Truck Diesel(TD)</i>	FE 71	110PS
	FE73	110PS
	FE73HD	110PS
	FE74	125PS
	FE74HD	125PS
	FE75	136PS
	FE84	136PS
Fuso	<i>Fuso</i> FM-517H	190 PS
	<i>Fuso</i> FM-517	190 PS
	<i>Fuso</i> FN-527	220 PS
	<i>Fuso</i> FN-527 HD	220 Ps
Outlander Sport	<i>Outlander Sport</i> GLX	150 PS

(Sumber: Data Historis PT Krama Yudha Ratu Motor, 2013)

Berikut ini merupakan gambar jenis-jenis produksi PT Krama Yudha Ratu Motor:



Gambar IV.3 Jenis-Jenis Produksi PT Krama Yudha Ratu Motor

(Sumber: PT Krama Yudha Ratu Motor, 2013)

Berikut merupakan gambar varian produk tipe TD pada PT Krama Yudha Ratu Motor:



Gambar IV.4 Varian Produk tipe TD di PT Krama Yudha Ratu Motor

(Sumber: PT Krama Yudha Ratu Motor, 2013)

4.4. Sistem Pengecekan Kualitas

Pengecekan kualitas PT Krama Yudha Ratu Motor dilakukan oleh Departemen *Quality Control* yang membawahi 2 bagian, yaitu:

1. Quality Inspection (QA)
2. Quality Assurance (QI)

4.5. Quality Assurance

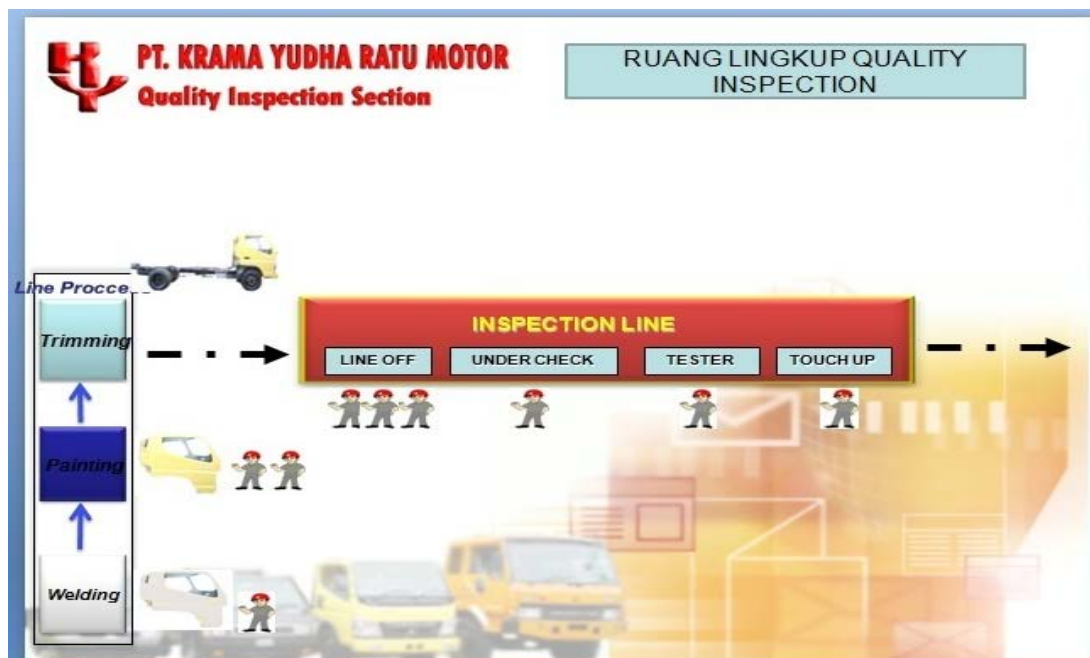
Quality Assurance (QA) adalah salah satu Departemen dari Quality Control yang melakukan kegiatan di perusahaan dalam melihat dan meningkatkan proses produksi yang sedang berlangsung. Quality Assurance (QA) juga berperan merencanakan dan mengendalikan kualitas hasil produk dan proses kerja welding, painting, dan Trimming 1 dengan melakukan audit ke 3 proses tersebut.

4.6. Quality Inspection (QI)

Quality Inspection (QI) adalah salah satu Departemen dari Quality Control yang melakukan kegiatan di perusahaan dalam melihat dan menginspeksi proses produksi yang sedang berlangsung. Quality Inspection (QI) juga berperan merencanakan dan mengendalikan kualitas hasil produk dari proses kerja *welding*, *painting*, dan *Trimming* agar terpeliharanya lingkungan kerja serta keselamatan dan kesehatan kerja. Ruang lingkup Quality Inspection (QI) proses kerja dimulai dari *welding* (pengelasan), *painting* (pengecatan), *Trimming* 1 (perakitan). Sesudah *line proses* berjalan melewati *Trimming* 1, unit kendaraan melakukan *inspection line*. *Inspection line* terdiri dari beberapa tahap antara lain:

1. Line Off : Pengecekan pada mesin kendaraan.
2. Under Check : Pengecekan pada bawah kendaraan.
3. Tester : Tes kecepatan dan tes pengereman
4. Touch UP : Kendaraan Sudah dinyatakan siap berjalan untuk delivery.

Berikut merupakan gambar ruang lingkup Quality Inspection (QI):



Gambar IV.5 Ruang Lingkup Quality Inspection

(Sumber: PT Krama Yudha Ratu Motor, 2013)

4.7. Trimming 1 (Perakitan)

Trimming 1 (perakitan) adalah salah satu Departemen dari produksi yang melakukan kegiatan di perusahaan dengan pemasangan komponen yang sudah menjadi *cabin* (kepala) dan *body* (badan). Proses kerja Trimming 1 adalah dimulai dari perakitan komponen-komponen yang telah melalui proses pengecatan. Dimana dalam perakitan 1 terdapat 16 stasiun kerja, yaitu masing-masing stasiun bertugas memasang komponen bagian luar maupun dalam dari sebuah mobil. Setelah selesai kemudian diangkat ke dalam proses perakitan lalu ke kerangka dan dilanjutkan ke pemasangan komponen lainnya, sehingga menjadi mobil. Pada Trimming ini, tipe produk yang dirakit adalah tipe TD. Berikut ini adalah gambar Stasiun kerja Trimming 1 tipe TD yang terdapat di PT Krama Yudha Ratu Motor (KRM):

layout dokumen yang sudah dicetak oleh Quality Inspection (QI) dan diberikan pada pihak Trimming untuk pencatatan kerusakan unit kendaraan tipe TD yang telah ditemukan. Berikut merupakan dokumen Countermeasure:

Penjelasannya dari gambar adalah sebagai berikut:

1. Kode Dokumen
Kode Dokumen Countermeasure.
2. Tanggal berlaku
Tanggal pencetakan dokumen
3. Revisi
No Revisi dokumen Countermeasure jika adanya kesalahan.
4. *Type*
Tipe produk diPT Krama Yudha Ratu Motor.
5. *To*
Bagian yang ditunjukkan kendala adanya *defect*.
6. *Date*
Tanggal penyerahan layout dokumen Countermeasure ke pihak Trimming 1.
7. *What/Where*
Penjabaran nama product dan tempat kejadian ditemukan kecacatan part.
8. *Quantity*
Jumlah part yang mengalami kecacatan.
9. *Why*
Penjabaran kenapa terjadi kerusakan pada part.
10. *How*
Penjabaran menangani kerusakan atau penanggulangannya.
11. *When*
Jadwal pelaksanaan perbaikan.
12. *Who*
Orang yang ditunjuk untuk menangani perbaikan
13. *Evaluation*
Keterangan yang ditulis berisi tentang hasil evaluasi perbaikan, setelah diperbaiki apakah sudah *Ok* atau *Not Good*.

4.9. Observasi *Daily Defect*

Berdasarkan hasil observasi dari penemuan kecacatan pada bagian Trimming yang berhasil dihimpun, dapat disimpulkan kedalam 1 tabel sebagai berikut:

Tabel IV.2 Klasifikasi *Defect* dan 3 *Worst Problem*

Jenis	<i>Defect</i>
Trimming	<i>W/nut cabin hinge LH not centre</i>
	<i>Brake pedal deep operation</i>
	<i>W/nut post mounting LH damage</i>

Sumber: PT. Krama Yudha Ratu Motor (2013)

4.10. Faktor Pengaruh Masalah pada Kendaraan Mobil Tipe TD

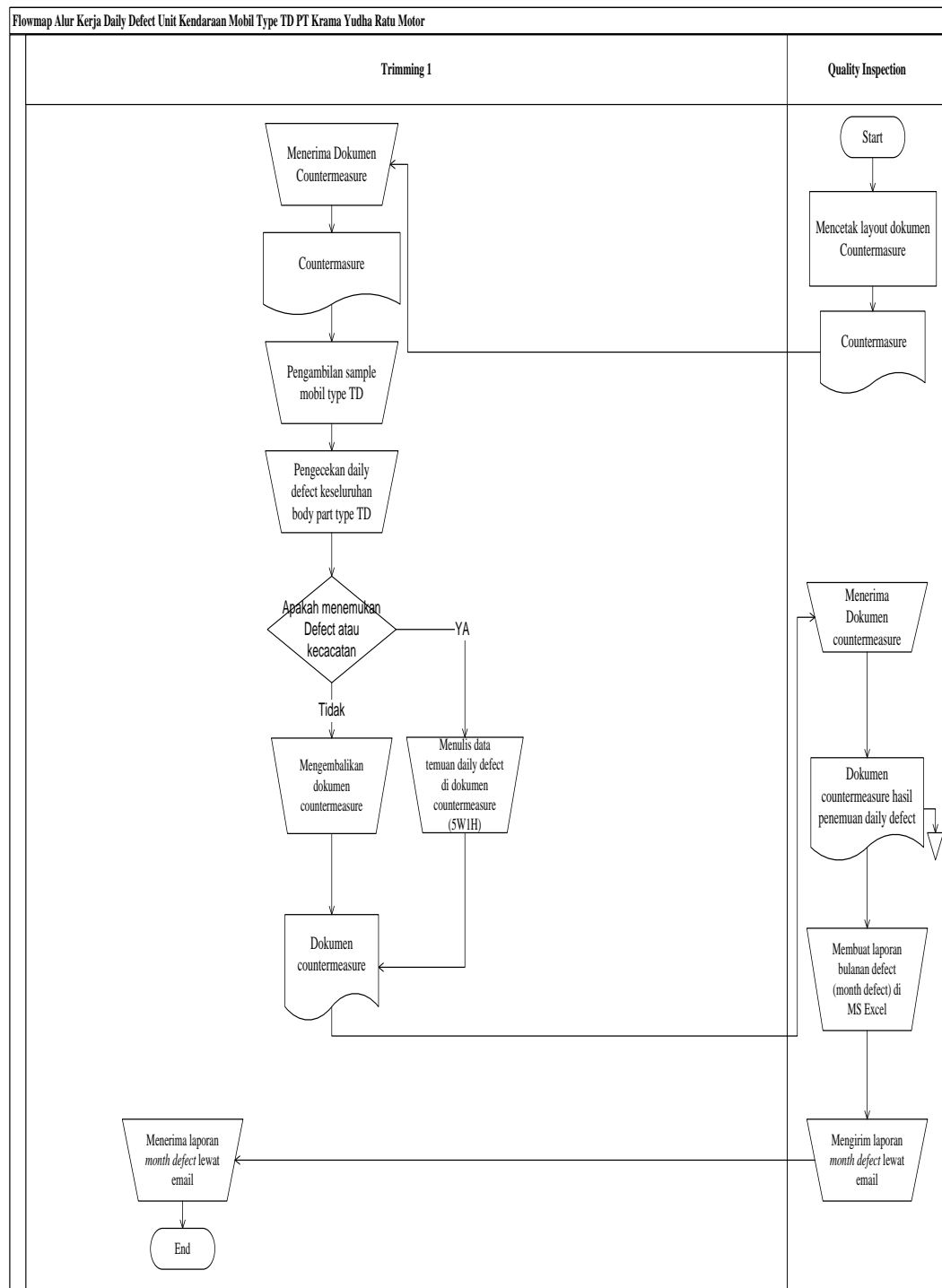
Faktor yang mempengaruhi masalah pada kendaraan mobil tipe TD adalah:

1. Faktor *Machines* Penyebabnya adalah kurang *maintenance* dan mesin sudah tua.
2. Faktor *Methods* Penyebabnya adalah pemasangan mesin kurang benar , mesin belum panas saat dilakukan pengujian emisi.
3. Faktor *Materials* Penyebabnya adalah mesin dari mobil supplier cacat, tidak dilakukan penyortiran saat penerimaan barang, tidak ada pengawasan terhadap supplier.
4. Faktor *Man* Penyebabnya adalah kurang terlatih, tidak memahami SOP, becanda saat bekerja.
5. Faktor *Environment* Penyebabnya adalah suhu ruangan panas, tempat kerja bocor saat hujan, lantai licin karena oli.
6. Faktor *System* Penyebabnya adalah prosedur pemasangan mesin belum jelas, prosedur penyortiran *material* mesin mobil belum jelas.

4.11. *Flowmap* Proses *Daily Defect* PT Krama Yudha Ratu Motor

Flowmap proses *daily defect* di PT KRM terdiri dari dua aktor, Trimming dan Quality Inspection(QI). *Flowmap* ini menjelaskan alur kerja pengecekan *defect*

/ kecacatan kendaraan mobil tipe TD. Berikut merupakan *Flowmap* alur kerja *Daily Defect* unit kendaraan tipe TD :



Gambar IV.8 Flowmap Alur Kerja *Daily Defect* Unit Kendaraan Mobil Type TD PT

Krama Yudha Ratu Motor

(Sumber: Pengolahan Data, 2017)

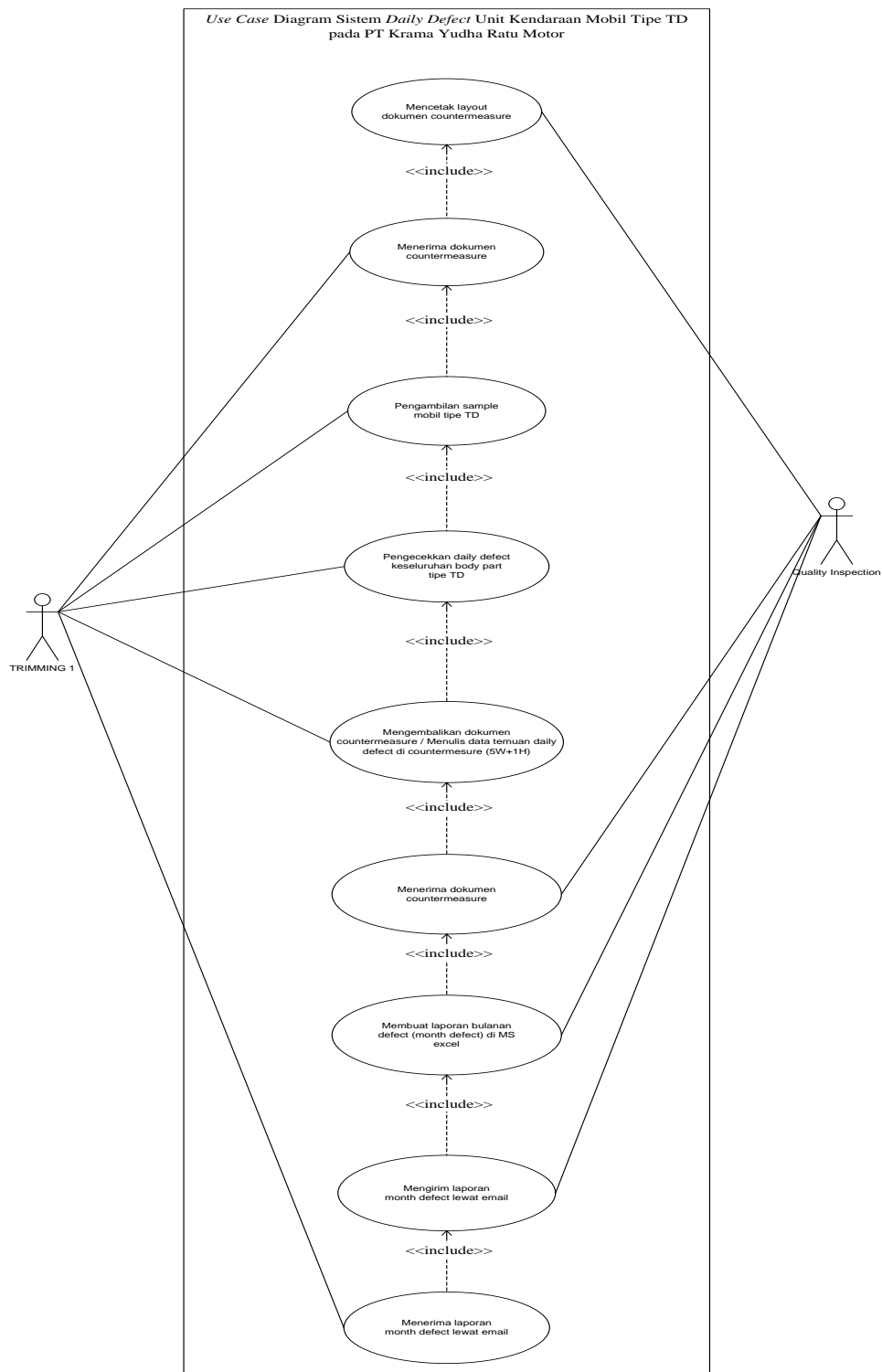
4.12. Proses Alur Kerja *Daily Defect* Unit Kendaraan Mobil Tipe TD

Proses alur kerja *Daily Defect* unit kendaraan mobil tipe TD adalah sebagai berikut:

1. Quality Inspection (QI) mencetak layout dokumen Countermeasure dan memberikannya ke bagian Trimming .
2. Setelah menerima dokumen Countermeasure, bagian Trimming melakukan pengecekan dengan pengambilan sample mobil tipe TD.
3. Berdasarkan hasil pada unit kendaraan mobil tipe TD, bagian Trimming menuliskan kecacatan di Countermeasure, jika tidak ada kecacatan dokumen Countermeasure dikembalikan ke Quality Inspection (QI)
4. Dokumen Countermeasure hasil penemuan *Daily Defect* diberikan ke Quality Inspection (QI) untuk diarsipkan dan dijadikan data untuk pembuatan laporan bulanan *defect* di *Microsoft Excel*.
5. Setelah laporan bulanan selesai laporan akan dikirim lewat *email* ke pihak Trimming .
6. Proses terakhir, bagian Trimming menerima laporan bulanan.

4.13. Penggambaran Sistem *Daily Defect* Unit Kendaraan Mobil Tipe TD Dengan *Unified Modelling Language*(UML)

Penggambaran dengan menggunakan *Use case diagram* bertujuan untuk menggambarkan fungsionalitas dari sebuah proses yang merepresentasikan sebuah interaksi antara aktor dengan aktor lain. Gambar IV.9 menunjukkan *Use case diagram* Sistem Informasi Pelaporan *Daily Defect* Unit Kendaraan Mobil Tipe TD Pada PT Krama Yudha Ratu Motor.



Gambar IV.9 *Use Case* Proses *Daily Defect* PT Krama Yudha Ratu Motor

(Sumber: Pengolahan Data, 2017)

Sistem *Daily Defect* pada PT KRM juga dapat digambarkan dalam bentuk aktor dan *Use Case*. Berikut ini adalah tabel definisi aktor dan definisi *Use Case* dalam sistem *Daily Defect*.

Tabel IV.3 Definisi Aktor dan Usecase Mencetak Dokumen Countermeasure

<i>Use Case</i>	Mencetak Dokumen Countermeasure
<i>Actor</i>	Quality Inspection (QI)
<i>Brief Description</i>	Kegiatan mencetak dokumen layout Countermeasure yang dilakukan oleh Quality Inspection (QI).
<i>Normal Flow of Events</i>	Quality Inspection (QI) mencetak dokumen layoutCountermeasure.
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.4 Definisi Aktor dan Usecase Menerima Dokumen *Layout Countermeasure*

<i>Use Case</i>	Menerima Dokumen Countermeasure
<i>Actor</i>	Trimming 1
<i>Brief Description</i>	Kegiatan menerima dokumen layout Countermeasure yang dilakukan oleh Quality Inspection (QI).
<i>Normal Flow of Events</i>	Trimming 1 menerima dokumen layoutCountermeasure.
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.5 Definisi Aktor dan Usecase Pengambilan *Sample Mobil Type TD*

<i>Use Case</i>	Pengambilan Sample Mobil Type TD
<i>Actor</i>	Trimming 1
<i>Brief Description</i>	Kegiatan megambil sample mobil type TD oleh Trimming 1 .
<i>Normal Flow of Events</i>	Trimming 1 melakukan pengambilan sample mobil type TD
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.6 Definisi Aktor Dan Usecase Pengecekan *Daily Defect* Keseluruhan *Body Part* Type TD

<i>Use Case</i>	Pengecekan <i>Daily Defect</i> keseluruhan <i>body part</i> type TD
<i>Actor</i>	Trimming 1
<i>Brief Description</i>	Kegiatan pengecekan <i>Daily Defect</i> keseluruhan <i>body part</i> type TD oleh Trimming 1
<i>Normal Flow of Events</i>	Trimming 1 melakukan pengecekan <i>Daily Defect</i> keseluruhan <i>body part</i> type TD
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.7 Definisi Aktor dan Usecase Mengembalikan Dokumen Countermeasure / Menulis Data Temuan *Daily Defect* (5W+1H) di Countermeasure

<i>Use Case</i>	Mengembalikan Dokumen Countermeasure / Menulis Data Temuan <i>Daily Defect</i> (5W+1H) di Countermeasure
<i>Actor</i>	Trimming 1
<i>Brief Description</i>	Kegiatan mengembalikan dokumen Countermeasure / menulis data temuan (5W+1H) di Countermeasure yang dilakukan oleh Trimming 1
<i>Normal Flow of Events</i>	Trimming 1 mengembalikan dokumen Countermeasure / menulis data temuan (5W+1H) di Countermeasure
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.8 Definisi Aktor dan Usecase Menerima Dokumen Countermeasure

<i>Use Case</i>	Menerima Dokumen Countermeasure
<i>Actor</i>	Quality Inspection (QI)
<i>Brief Description</i>	Kegiatan Menerima Dokumen Countermeasure yang dilakukan oleh Quality Inspection (QI)
<i>Normal Flow of Events</i>	Quality Inspection (QI) Menerima Dokumen Countermeasure
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.9 Definisi Aktor dan Usecase Membuat Laporan Bulanan (*Month Defect*) Di MS Excel

<i>Use Case</i>	Membuat Laporan Bulanan (<i>Month Defect</i>) Di MS Excel
<i>Actor</i>	Quality Inspection (QI)
<i>Brief Description</i>	Kegiatan membuat laporan bulanan (<i>month defect</i>) di MS Excel yang dilakukan oleh Quality Inspection (QI)
<i>Normal Flow of Events</i>	Quality Inspection (QI) membuat laporan bulanan (<i>month defect</i>) di MS Excel
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.10 Definisi Aktor dan Usecase Mengirim Laporan Bulanan (*Month Defect*) Lewat Email

<i>Use Case</i>	Mengirim Laporan Bulanan (<i>Month Defect</i>) Lewat Email
<i>Actor</i>	Quality Inspection (QI)
<i>Brief Description</i>	Kegiatan mengirim laporan bulanan (<i>month defect</i>) lewat email yang dilakukan oleh Quality Inspection (QI)
<i>Normal Flow of Events</i>	Quality Inspection (QI) mengirim laporan bulanan (<i>month defect</i>) lewat email ke Trimming 1
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

Tabel IV.11 Definisi Aktor dan Usecase Menerima Laporan Bulanan (*Month Defect*) Lewat Email

<i>Use Case</i>	Menerima Laporan Bulanan (<i>Month Defect</i>) Lewat Email
<i>Actor</i>	Trimming 1
<i>Brief Description</i>	Kegiatan menerima laporan bulanan (<i>month defect</i>) lewat email yang dilakukan oleh Trimming 1
<i>Normal Flow of Events</i>	Trimming 1 menerima laporan bulanan (<i>month defect</i>) lewat email

Tabel IV.11 Definisi Aktor dan Usecase Menerima Laporan Bulanan (*Month Defect*)
Lewat Email (lanjutan)

<i>Use Case</i>	Menerima Laporan Bulanan (<i>Month Defect</i>) Lewat Email
<i>Alternate/exceptional Flows</i>	-

(Sumber: Pengolahan Data, 2017)

BAB V

ANALISIS DAN PEMBAHASAN

Sistem *daily defect* pada PT Krama Yudha Ratu Motor merupakan pengembangan dari sistem sebelumnya, dimana sistem yang dirancang diharapkan dapat memberikan kemudahan pada pihak perusahaan khususnya dalam proses pembuatan dokumen countermeasure atau laporan bulanan countermeasure. Dalam bab ini, akan dijelaskan analisis rinci dari pengolahan data Sistem Informasi *daily defect* pada PT Krama Yudha Ratu Motor. Analisis tersebut dimulai dari identifikasi kebutuhan *user*, perancangan UML, perancangan basis data, perancangan *interface*, perancangan HIPO dan pembuatan spesifikasi proses melalui metode yang diterapkan.

5.1. Analisis Kebutuhan User

Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan perancangan Sistem Informasi Pelaporan *Daily Defect* sebagai sarana untuk memberikan kemudahan dalam memberi laporan bulanan pada PT Krama Yudha Ratu Motor. Untuk mendukung Sistem Informasi Pelaporan *Daily Defect* dibutuhkan juga kemampuan *user* dalam mengoperasikan sistem.

Kebutuhan sistem informasi dapat dijelaskan melalui proses-proses yang dibutuhkan pemakai dan kebutuhan sistem. Identifikasi kebutuhan sistem komputerisasi menjelaskan proses yang dilakukan sesuai dengan kebutuhan pemakai. Berikut adalah penjelasan identifikasi kebutuhan *user* Sistem Informasi Pelaporan *Daily Defect* pada PT Krama Yudha Ratu Motor.

Tabel V.1 Identifikasi Kebutuhan *User*

No.	Identifikasi Kebutuhan	Uraian
1.	Proses Sistem berjalan	<ol style="list-style-type: none"> 1. Quality Inspection (QI) mencetak layout dokumen Countermeasure. 2. Setelah mencetak dokumen Countermeasure, bagian Quality Inspection memberikan dokumen Countermeasure ke bagian Trimming 1. 3. Setelah menerima dokumen Countermeasure pihak Trimming 1 mengambil sample mobil tipe TD. 4. Setelah mengambil sample mobil tipe TD dilakukan pengecekan <i>Daily Defect</i> keseluruhan <i>body part</i> tipe TD. 5. Setelah dilakukan pengecekan <i>Daily Defect</i> apabila ditemukan kecacatan, jika iya, pihak Trimming 1 menulis data temuan <i>Daily Defect</i> di Countermeasure (5W1H). lalu apabila tidak, maka pihak Trimming 1 mengembalikan dokumen Countermeasure ke bagian Quality Inspection (QI). 6. Setelah Quality Inspection (QI) menerima dokumen Countermeasure, lalu dokumen temuan <i>Daily Defect</i> tersebut di arsipkan. Dan Quality Inspection (QI) membuat laporan bulanan <i>defect</i> di <i>Microsoft Excel</i>, lalu laporan bulanan dokumen Countermeasure tersebut dikirim melalui email. 7. Proses terakhir Trimming 1 menerima email laporan bulanan <i>defect</i> dari Quality Inspection (QI).
2.	Data masukan (Input)	<p>Data Countermeasure</p> <p>Data laporan bulanan Countermeasure</p>
3.	Data Keluaran (Output)	Data Laporan Bulanan Countermeasure

Tabel V.1 Identifikasi Kebutuhan *User* (lanjutan)

No.	Identifikasi Kebutuhan	Uraian
4.	Model Sistem Informasi	Input : Data Master Data User Data Komponen Data Countermeasure Data Laporan Bulanan Countermeasure Proses : Membuat data dokumen Countermeasure, melihat dan membuat laporan bulanan Countermeasure, kedalam sistem. Output: Mencetak laporan bulanan Countermeasure.

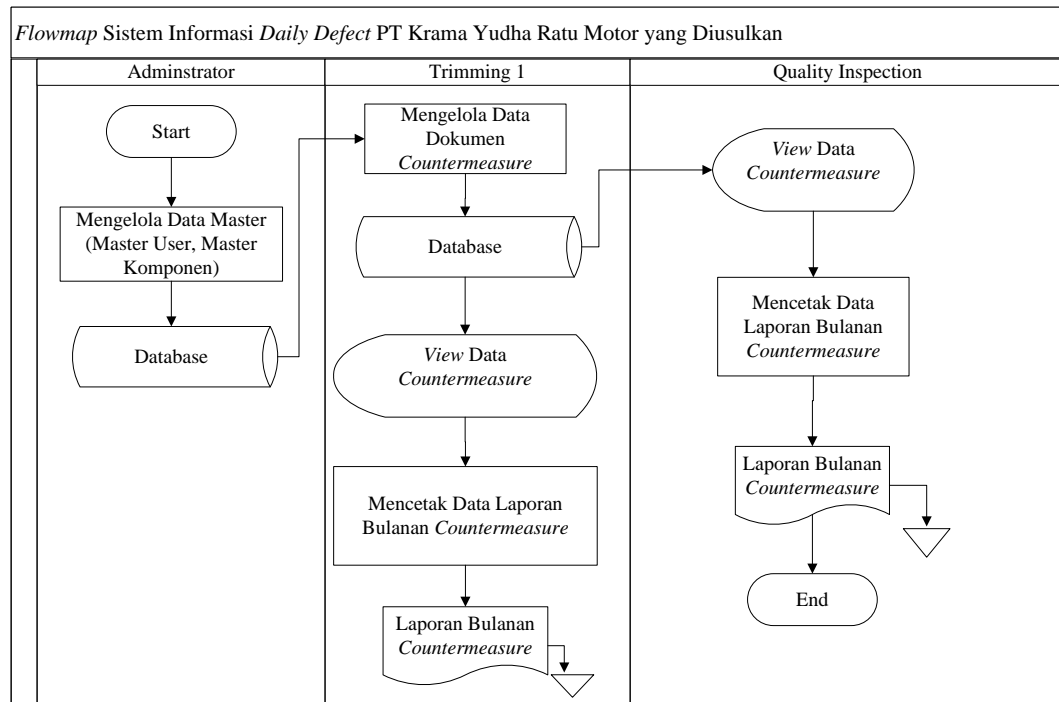
(Sumber: Hasil Analisis Data, 2017)

5.2. Perancangan Sistem Informasi *Daily Defect*

Dalam perancangan Sistem Informasi *Daily Defect* pada PT KramaYudha Ratu Motor ada tiga tahapan yang digunakan yaitu:

1. Pembuatan model sistem digunakan untuk mengetahui alur sistem yang diusulkan, yaitu dengan menggunakan *Unified Modelling Language* (UML).
2. Pembuatan model data dengan *Class Diagram* dan Kamus Data.
3. Perancangan program yang diusulkan dengan tahapan pembuatan HIPO dan perancangan antarmuka program.

5.3. Flowmap Sistem Informasi *Daily Defect* yang Diusulkan



Gambar V.1 *Flowmap* Sistem Informasi *Daily Defect* PT Krama Yudha Ratu Motor

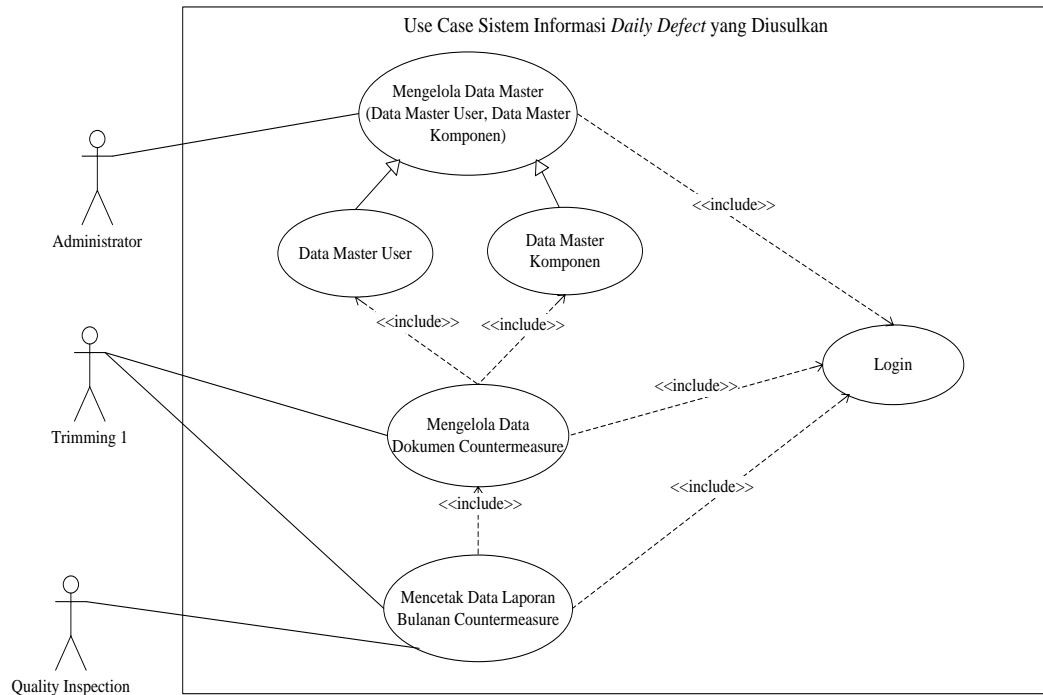
(Sumber: Hasil Pengolahan Data, 2017)

Berdasarkan pada Gambar V.1, penjelasan mengenai alursistem *Daily Defect* di PT Krama Yudha Ratu Motor yang diusulkan adalah sebagai berikut:

1. Administrator mengelola data master (master user dan master komponen) master user pengguna dapat mengelola keseluruhan data sedangkan master komponen keseluruhan data komponen dinspeksi oleh pihak Trimming 1.
2. Trimming 1 mengelola data dokumen Countermeasure. Setelah mengelola data dokumen Countermeasure dibuatlah database.
3. Setelah membuat data Countermeasure dan database pada sistem. Dibuat view data Countermeasure
4. Berikutnya pihak Trimming 1 dapat mencetak data laporan bulanan Countermeasure dan di arsipkan.
5. Proses akhir Quality Inspection (QI) dapat mencetak data laporan bulanan Countermeasure dan di arsipkan.

5.4. Use Case Diagram Sistem Informasi Daily Defect yang Diusulkan

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dalam sistem ini terdapat aktor *Administrator*, *Trimming1* dan *Quality Inspection (QI)*. Gambar V.2 merupakan *Use Case Diagram* Sistem Informasi *Daily Defect* yang diusulkan.



Gambar V.2 *Use Case Diagram* Sistem Informasi *Daily Defect* yang Diusulkan
(Sumber: Hasil Analisis Data, 2017)

5.5. Usecase Description Sistem Informasi Daily Defect yang Diusulkan

Use case Description merupakan gambaran *Use case* yang berisi nama *Use case*, primary actor, brief description, relationship, normal flow of events, precondition, subflows dan alternate/exceptional flows. Berikut ini adalah *Use Case description* dari *Use case diagram* dalam Gambar V.2.

Tabel V.2 *Use Case Description Login*

Nama Use Case	Login
Primary Actor	Administrator, Trimming 1, Quality Inspection
Brief Description	Sistem ini menggambarkan proses aktor masuk ke dalam sistem.
Relationship	Association: Administrator, Trimming 1, Quality Inspection . Include: - Extends: -
Normal Flow of Events	Aktor membuka Sistem Informasi <i>Daily Defect</i> . Aktor memasukan username dan password pada form login. Sistem mengecek ke database, apakah username dan password sudah benar. Jika username dan password benar (valid), akan muncul menu utama sistem. Dan sistem menampilkan data user dari aktor yang sedang login. Jika username dan password salah (tidak valid), maka muncul message box “Maaf, <i>username</i> dan password salah!” dan kembali ke form login.
Precondition	Membuka Sistem Informasi <i>Daily Defect</i> .
Subflows	-
Alternate/exceptional flows	-

(Sumber: Hasil Analisis Data, 2017)

Tabel V.3 *Use Case Description MasterUser*

Nama UseCase	Master User
Primary Actor	Administrator
Brief Description	Master user mengatur hak Akses Login
Relationship	Association: Administrator Include: - Extends: -
Normal Flow of Events	Administrator membuka master Pilih submenu Master User Klik Tambah Data untuk menambah data. Klik Ubah untuk mengubah data. Klik Hapus untuk menghapus data.
Precondition	Login ke dalam Sistem Informasi <i>Daily Defect</i> .
Subflows	-
alternate/exceptional flows	-

(Sumber: Hasil Analisis Data, 2017)

Tabel V.4 *Use Case Description MasterKomponen*

Nama UseCase	Master Komponen
Primary Actor	Administrator
Brief Description	Master Komponendata keseluruhan part2 yang diinpeksi
Relationship	Association: Administrator Include: - Extends: -
Normal Flow of Events	Administrator membuka master Pilih submenu Master Komponen Klik Tambah Data untuk menambah data. Klik Ubah untuk mengubah data. Klik Hapus untuk menghapus data.
Precondition	Login ke dalam Sistem Informasi <i>Daily Defect</i> .
Subflows	-
alternate/exceptional flows	-

(Sumber: Hasil Analisis Data, 2017)

Tabel V.5 *Use Case Description* Menegloa Data Dokumen Countermeasure

Nama UseCase	Master Menegloa Data Dokumen Countermeasure
Primary Actor	Trimming 1
Brief Description	Trimming 1 dapat melakukan tambah, hapus, ubah, dan cetak data terhadap Data Dokumen Countermeasure.
Relationship	Association: Trimming 1 Include: - Extends: -
Normal Flow of Events	Administrator membuka menu Data dokumen Countermeasure. Pilih Menu Countermeasure Klik Tambah Data untuk menambah data. Klik Ubah untuk mengubah data. Klik Hapus untuk menghapus data. Klik Cetak untuk mencetak
Precondition	Login ke dalam Sistem Informasi <i>Daily Defect</i> .
Subflows	-
alternate/exceptional flows	-

(Sumber: Hasil Analisis Data, 2017)

Tabel V.6 *Use Case Description* Mencetak Data Laporan Bulanan Countermeasure

Nama Use Case	Mencetak Data Laporan Bulanan Countermeasure
Primary Actor	Quality Inspection (QI)
Brief Description	Quality Inspection (QI) dapat melakukan tambah, hapus, ubah, dan cetak laporan bulanan Countermeasure.
Relationship	Association: Quality Inspection (QI) Include: - Extends: -

(Sumber: Hasil Analisis Data, 2017)

Tabel V.6 *Use Case Description* Membuat Data Countermeasure(lanjutan)

Nama <i>Use Case</i>	Countermeasure
Normal Flow of Events	Quality Inspection membuka menu laporan bulanan Countermeasure. Quality Inspection (QI) pilih laporan bulanan Countermeasure Klik Tambah Data untuk menambah data. Klik Ubah untuk mengubah data. Klik Hapus untuk menghapus data. Klik cetak untuk mencetak
Precondition	Login ke dalam Sistem Informasi <i>Daily Defect</i>
Subflows	-
alternate/exceptional flows	-

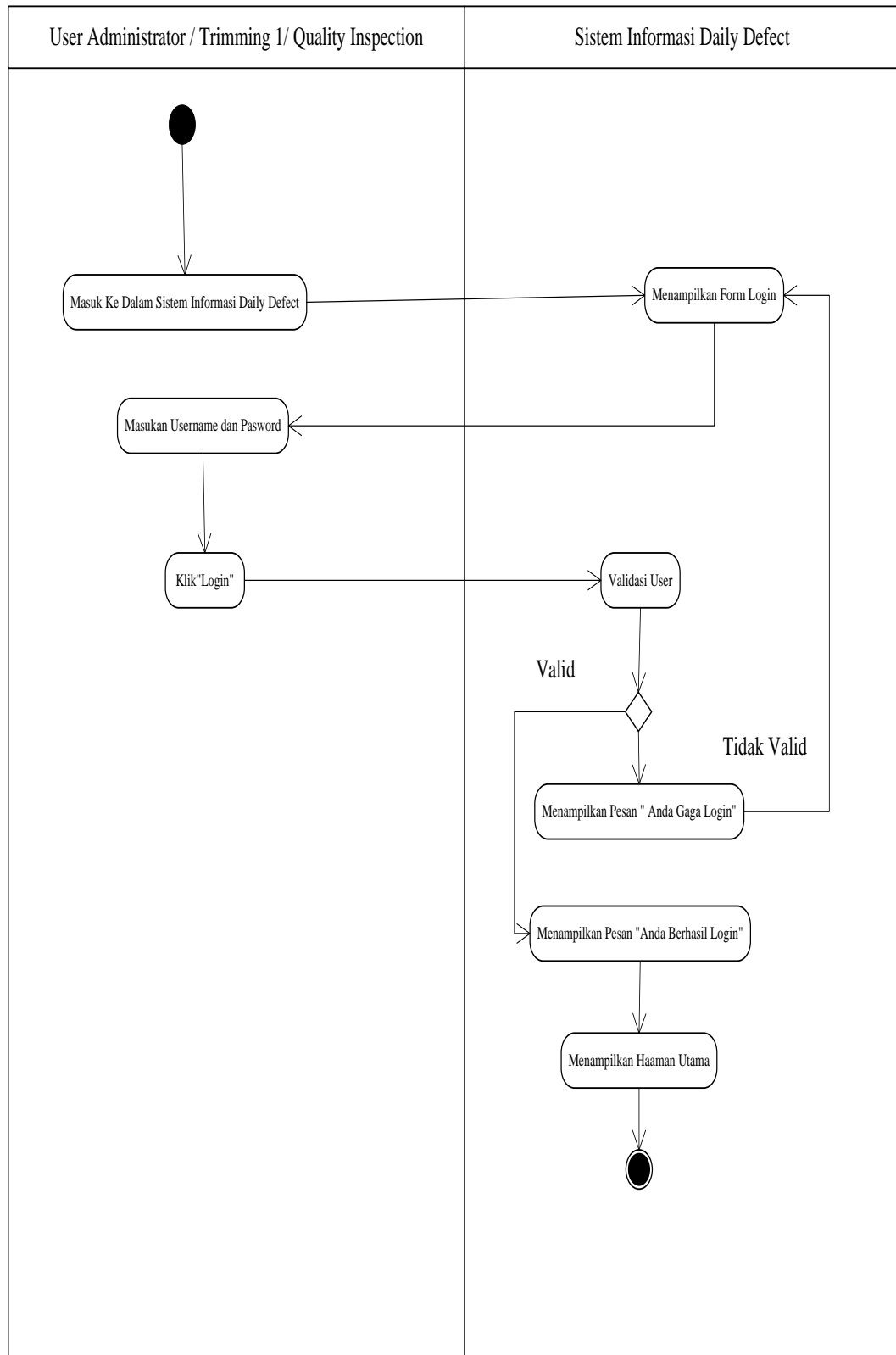
(Sumber: Hasil Analisis Data, 2017)

5.6. *Activity Diagram* Sistem Informasi *Daily Defect* yang Diusulkan

Activity Diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. Pembuatan *Activity Diagram* untuk lebih memahami sistem yang akan dibuat. Berikut ini adalah *Activity Diagram* yang terdapat dalam Sistem Informasi *Daily Defect*.

1. *Activity Diagram* Login

Activity Diagram login ini menjelaskan alur aktivitas yang dilakukan oleh *user*. *User* pada Sistem Informasi *Daily Defect* ini adalah Administrator, Trimming dan Quality Inspection. Gambar V.3 menunjukkan *Activity Diagram* login.

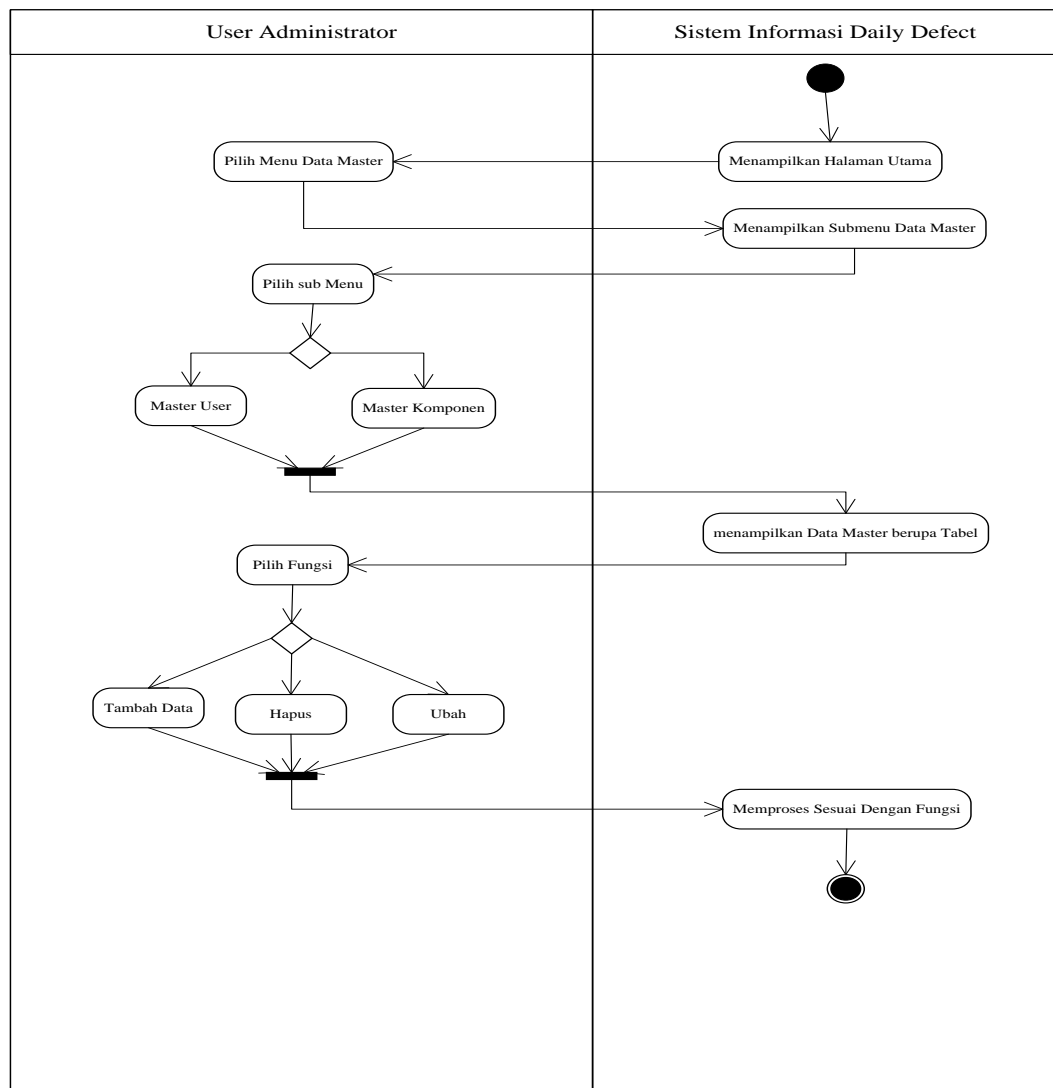


Gambar V.3 Activity Diagram Login

(Sumber: Hasil Analisis Data, 2017)

2. Activity Diagram Mengelola Data Master

User yang memiliki hak akses untuk mengelola data master adalah Administrator. Administrator dapat mengelola data master hingga menampilkan tabel dari keseluruhan data master. Data master meliputi master user, master komponen. Gambar V.4 menunjukkan Activity Diagram untuk mengelola data master.

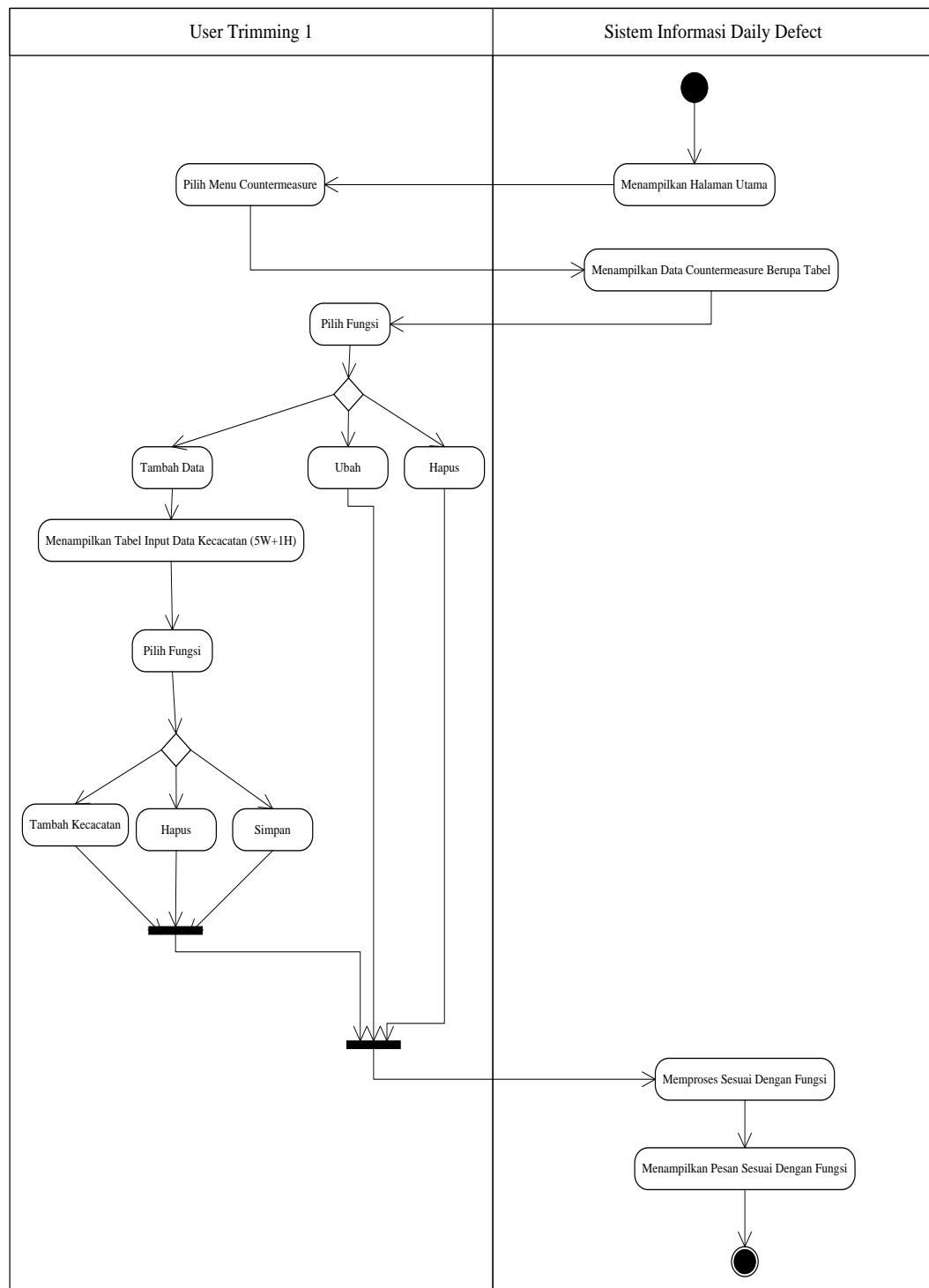


Gambar V.4 Activity Diagram untuk Mengelola Data Master

(Sumber: Hasil Analisis Data, 2017)

3. *Activity Diagram* untuk Mengelola Data Dokumen Countermeasure.

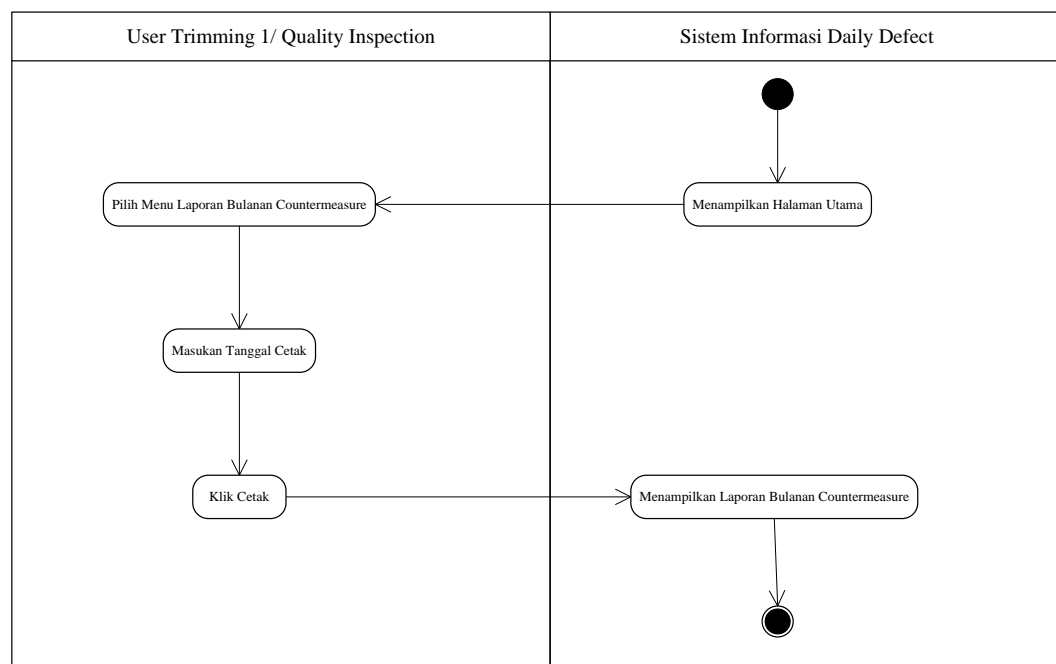
User yang memiliki hak akses untuk mengelola data dokumen Countermeasure adalah Trimming 1. Trimming 1 dapat mengelola data dokumen Countermeasure dan melihat dan mencetak Data dokumen Countermeasure. Gambar V.5 menunjukkan *Activity Diagram* untuk mengelola data dokumen Countermeasure.



Gambar V.5 Activity Diagram untuk Membuat Data Countermeasure
(Sumber: Hasil Analisis Data, 2017)

4. *Activity Diagram* Mencetak Laporan Countermeasure

User yang memiliki hak akses laporan bulanan Countermeasure menjelaskan dapat mencetak laporan bulanan Countermeasure. Bagian Quality Inspection dapat melihat dan mencetak bulanan Countermeasure. Gambar V.6 menunjukkan *Activity Diagram* untuk mencetak laporan bulanan *countemeasure*.



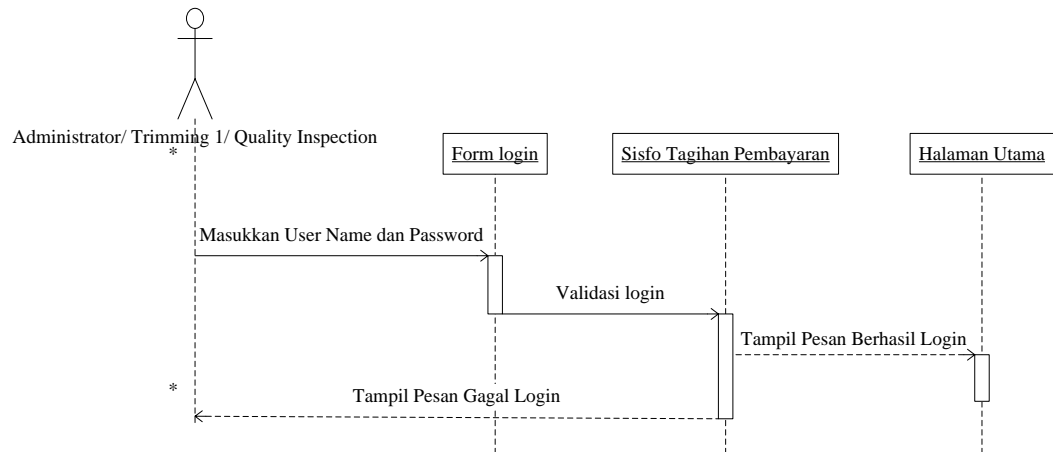
Gambar V.6 *Activity Diagram* untuk Membuat laporan bulanan Countermeasure

(Sumber: Hasil Analisis Data, 2017)

5.7. *Sequence Diagram* Sistem Informasi *Daily Defect* yang Diusulkan

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *Use case*. *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence Diagram* digunakan untuk memodelkan pengiriman *message* antar *objects*. Berikut ini merupakan beberapa gambar *Sequence diagram* dari Sistem Informasi *Daily defect*.

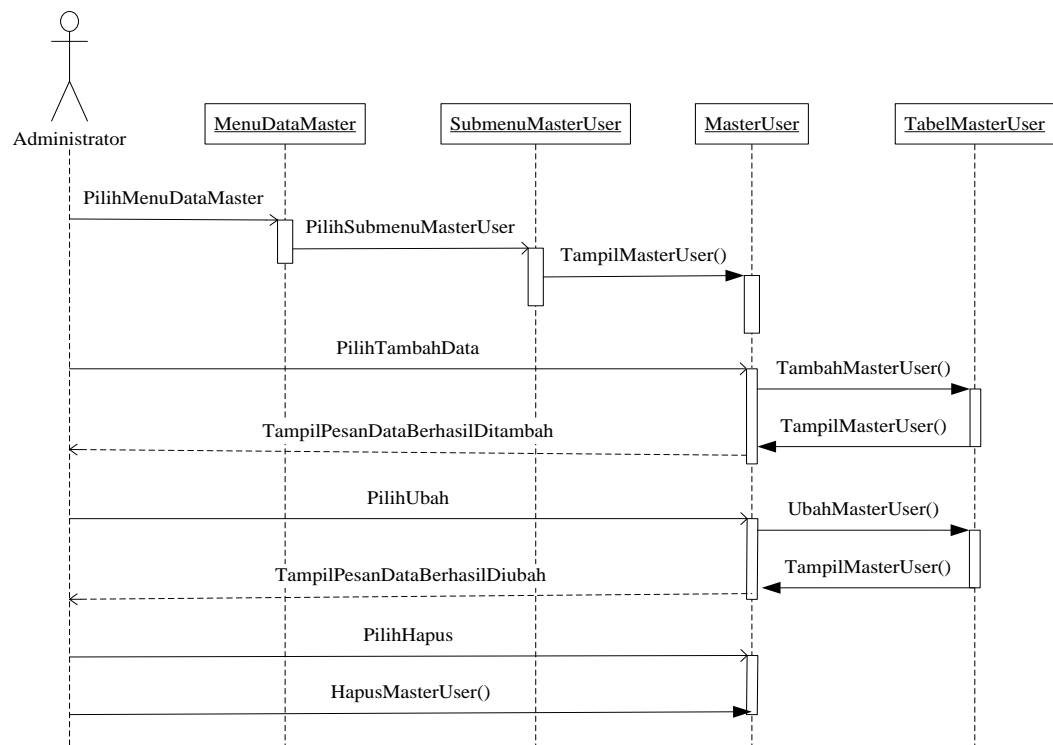
1. Sequence Diagram Login



Gambar V.7 Sequence Diagram Login

(Sumber: Hasil Analisis Data, 2017)

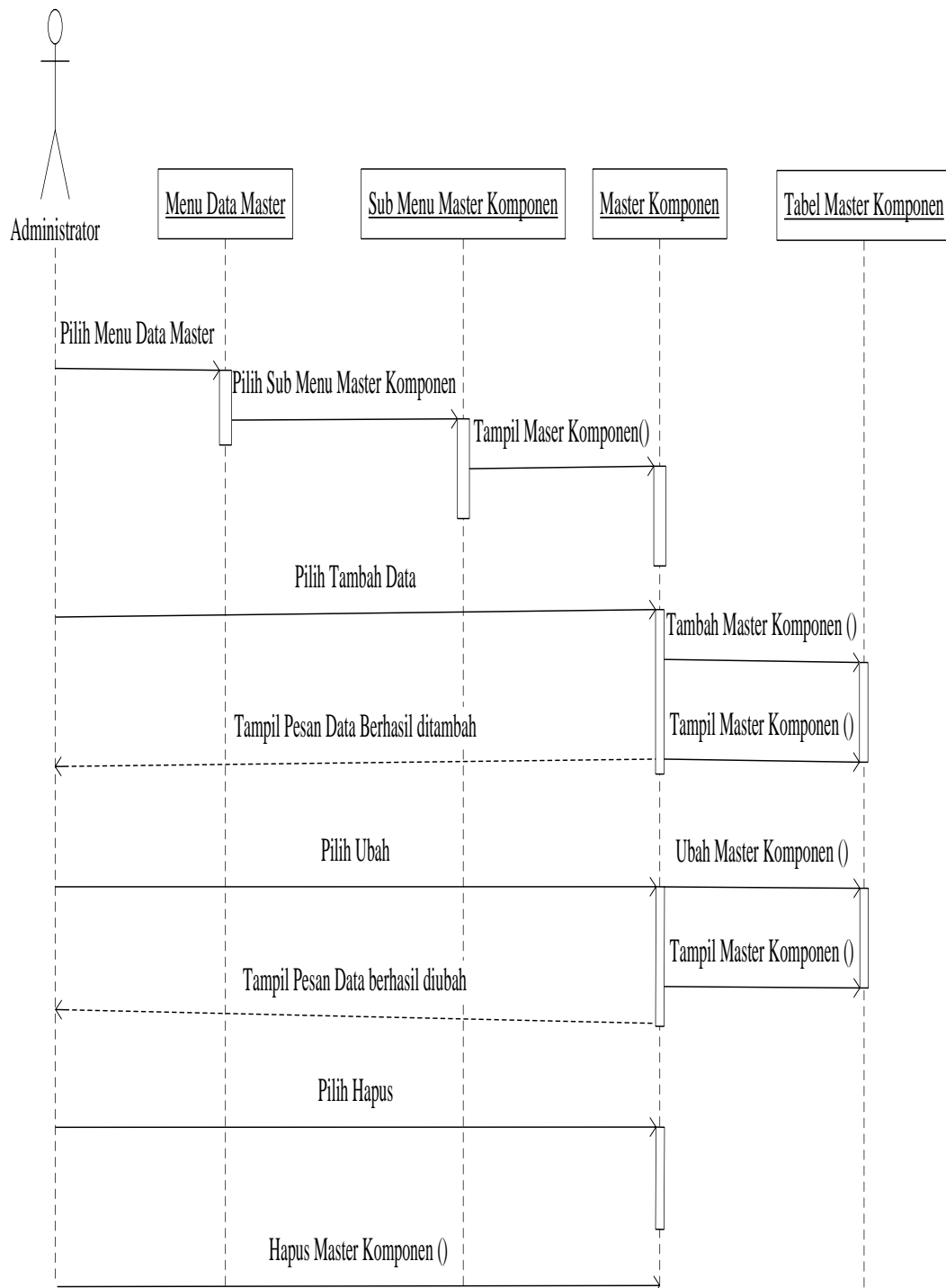
2. Sequence Diagram Membuat Master User



Gambar V.8 *Sequence Diagram* Membuat Master User

(Sumber: Hasil Analisis Data, 2017)

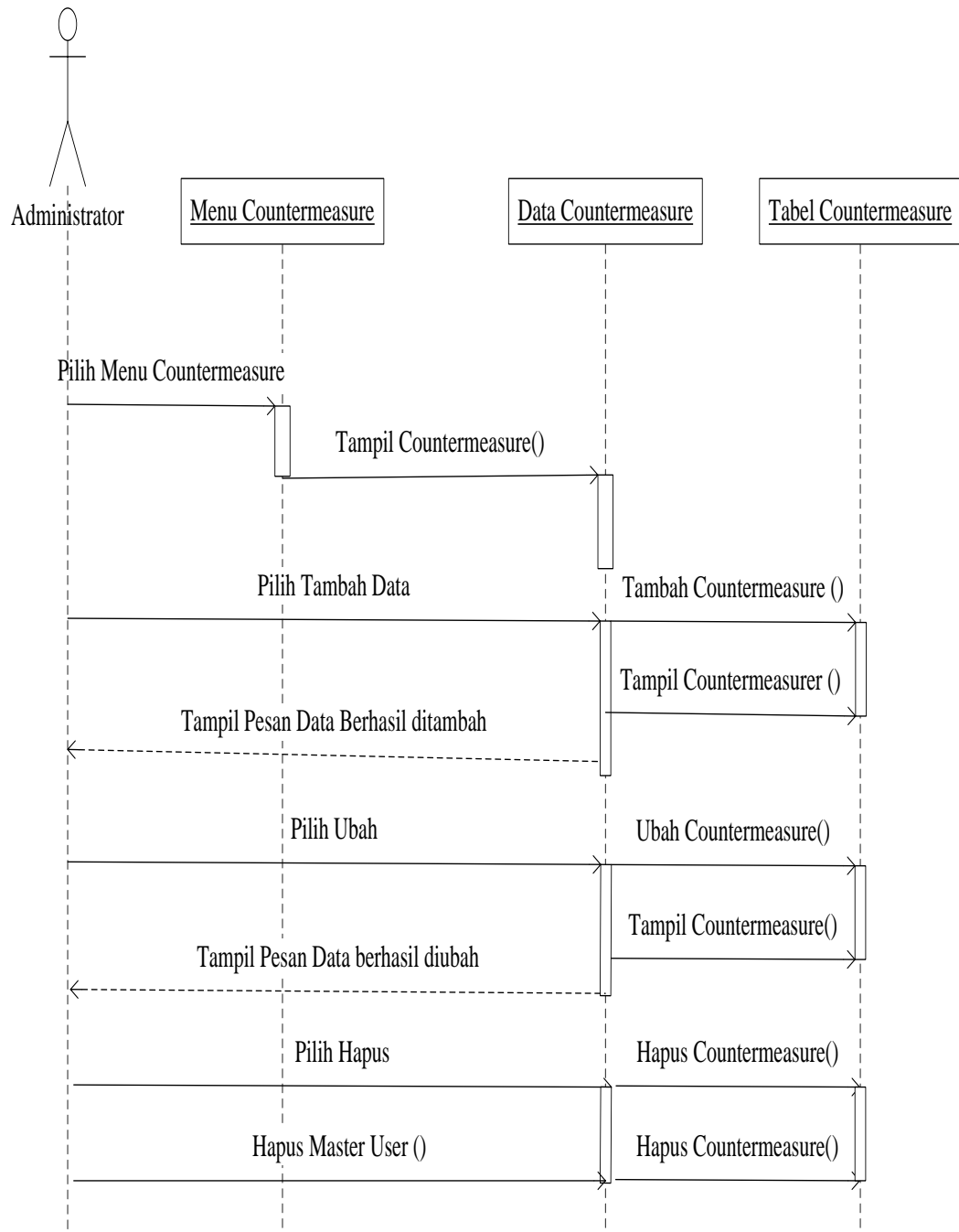
3. *Sequence Diagram* Master Komponen



Gambar V.9 *Sequence Diagram* Membuat Master Komponen

(Sumber: Hasil Analisis Data, 2017)

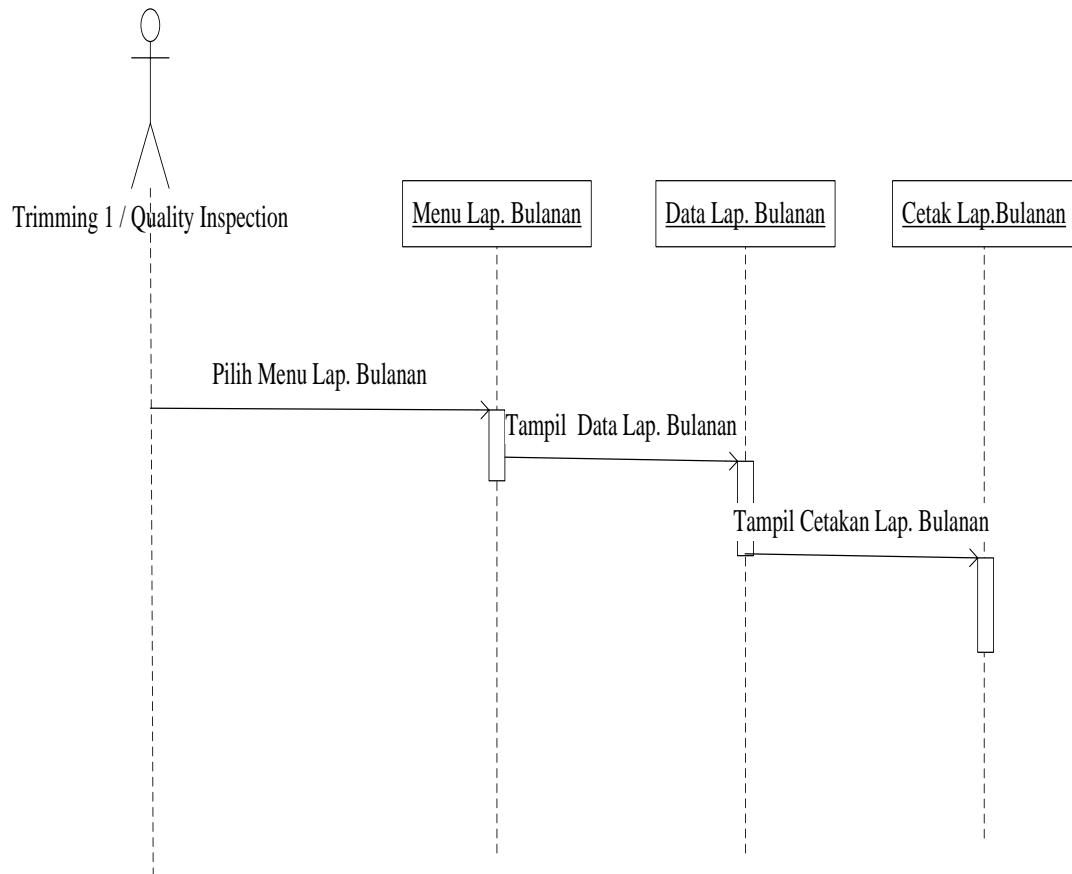
4. Mengelola Data DokumenCountermeasure



Gambar V.10 Sequence Diagram Mengelola Data DokumenCountermeasure

(Sumber: Hasil Analisis Data, 2017)

5. *Sequence Diagram* laporan bulanan Countermeasure

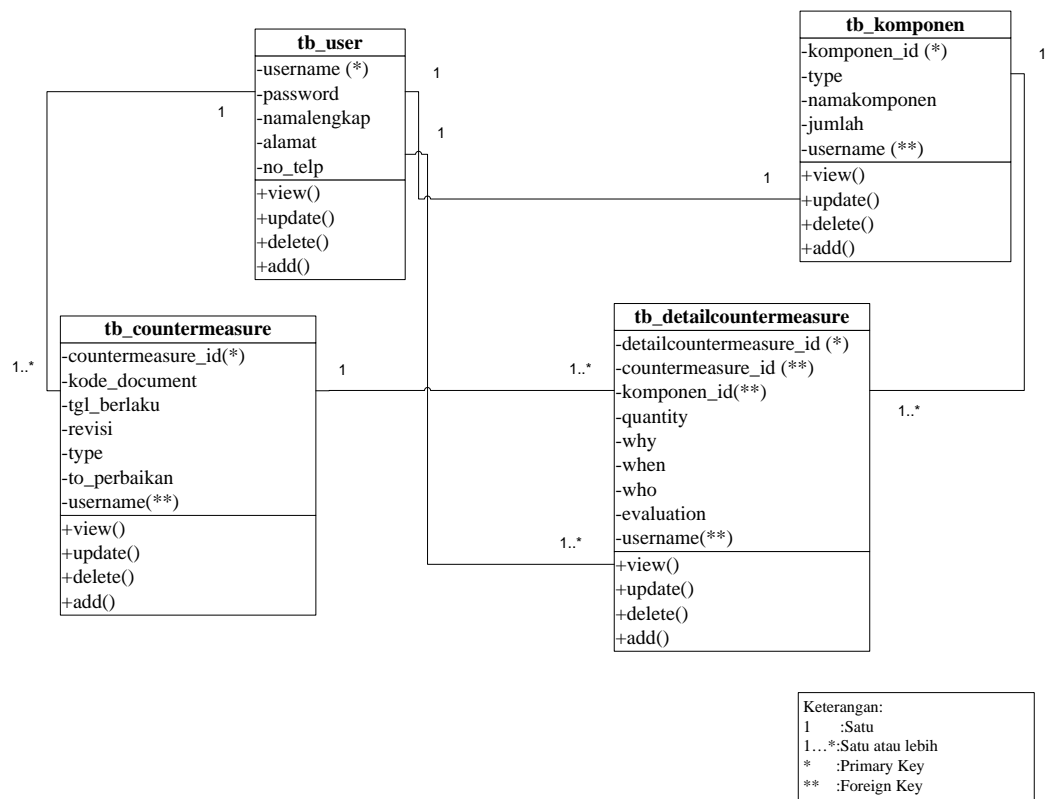


Gambar V.11 *Sequence Diagram* Laporan Bulanan Countermeasure

(Sumber: Hasil Analisis Data, 2017)

5.8. *Class Diagram* Sistem Informasi Pelaporan *Daily Defect* yang Diusulkan

Class diagram membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe *diagram* yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas dalam model desain dari suatu sistem. Gambar V.12 berikut ini merupakan *class diagram* dari Rancangan Bangun Sistem Informasi Pelaporan *Daily Defect*.



Gambar V.12 Class Diagram Sistem Informasi Daily Defect

(Sumber: Hasil Analisis Data, 2017)

5.9. Kamus Data

Kamus data dapat berfungsi membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan. Penjelasan mengenai kamus data tiap tabel pada Sistem *Daily Defect* dapat dijelaskan berikut ini.

Tabel V.7 Spesifikasi Tabel Master *User*

Nama Tabel : tb_user			
Fungsi : untuk menyimpan data user.			
Tipe : file master user			
No	Coloumn	Tipe Data	Keterangan
1	Username	Varchar (20)	Primary Key
2	Password	Varchar (50)	-
3	Nama lengkap	Varchar (30)	-
4	Bagian	Varchar (10)	-
4	Alamat	Varchar (50)	-
5	No_telp	Varchar (12)	-

(Sumber: Hasil Analisis Data, 2017)

Tabel V.8 Spesifikasi Tabel Master Komponen

Nama Tabel : tb_Komponen			
Fungsi : untuk menyimpan data-data Komponen.			
Tipe : file master Komponen			
No	Coloumn	Tipe Data	Keterangan
1.	Komponen_id	Char (5)	Primary Key
2.	Type	Varchar (10)	-
3.	Nama Komponen	Varchar (30)	-
4.	Username	Varchar (20)	Foreign Key dari Master User

(Sumber: Hasil Analisis Data, 2017)

Tabel V.9 Spesifikasi Tabel Countermeasure

Nama Tabel : tb_Countermeasure			
Fungsi : untuk menyimpan data-data temuan defect dalam periode 1 hari.			
Tipe : file Countermeasure			
No	Coloumn	Tipe Data	Keterangan
1.	Countermeasure_id	Char(5)	Primary Key
2.	Kode_Document	Varchar(20)	-
3.	Tgl_berlaku	Date	-
4.	Revisi	Varchar(4)	-
5.	Type	Varchar(10)	-
6.	To_perbaikan	Varchar(20)	-
7.	Username	Varchar(20)	Foreign key dari master user

(Sumber: Hasil Analisis Data, 2017)

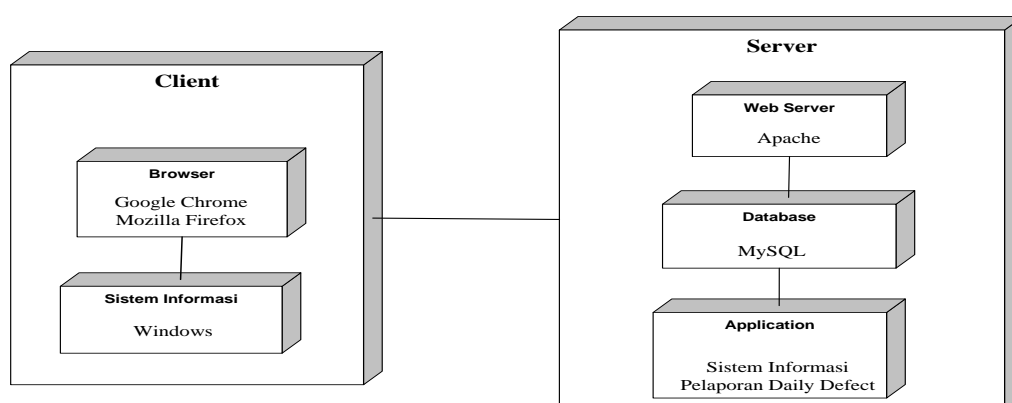
Tabel V.10 Spesifikasi Tabel DetailCountermeasure

Nama Tabel : tb_detailCountermeasure			
Fungsi : untuk menyimpan temuan <i>defect</i> , karena satu Countermeasurememiliki satu atau lebih detail Countermeasure			
Tipe : file detailCountermeasure			
No	Coloumn	Tipe Data	Keterangan
1.	DetailCountermeasure_id	Char(5)	Primary Key
2.	Countermeasure_id	Char(5)	Foreign Key master user
3.	Komponen_id	Char(5)	Foreign Key master user
4.	Quantity	Int(4)	-
5.	Why	Text	-
6.	Who	Varchar(30)	-
7.	Evaluation	Varchar(10)	-
8.	Username	Varchar(20)	Foreign key dari master user

(Sumber: Hasil Analisis Data, 2017)

5.10. *Deployment Diagram* Sistem Informasi *Daily Defect* yang Diusulkan

Deployment diagram ini digunakan untuk membuat gambaran susunan fisik sebuah sistem, dan gambaran bagian perangkat lunak mana yang berjalan pada perangkat keras. Berikut ini adalah *deployment diagram* dari Sistem Informasi Pelaporan *Daily Defect* yang diusulkan.

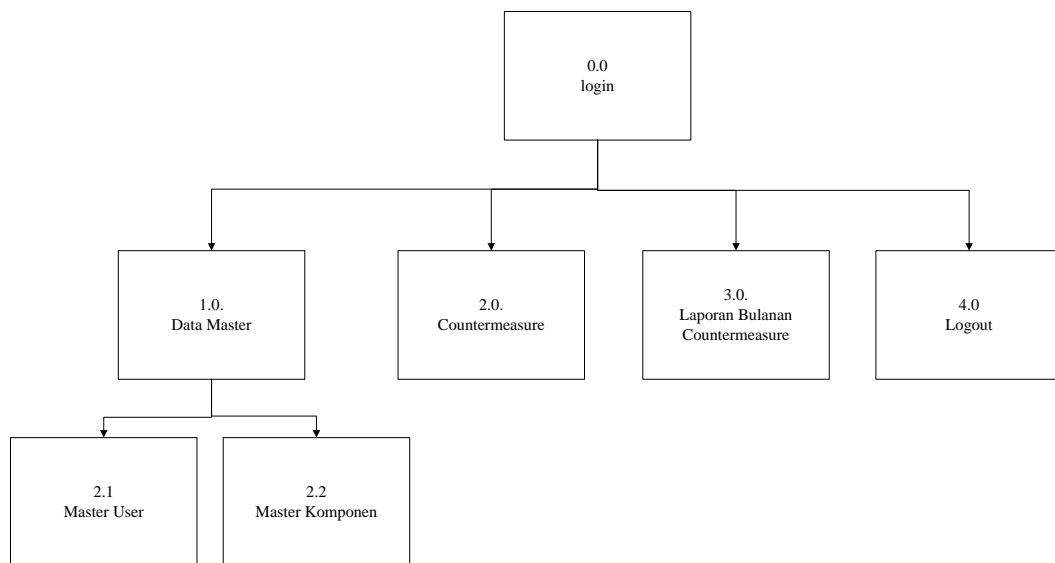


Gambar V.13 *Deployment Diagram* Sistem Informasi Pelaporan *Daily Defect* yang Diusulkan

(Sumber: Hasil Analisis Data, 2017)

5.11. *Hierarchy Plus-Input-Process-Output (HIPO)* Sistem Informasi Pelaporan *Daily Defect* yang Diusulkan

Pembuatan aplikasi dilakukan menggunakan perangkat lunak PHP *Designer* dan menggunakan bahasa pemrograman PHP 5.4.7 serta MySQL 5.5.27 sebagai perangkat lunak basis data. Berikut adalah struktur menu hirarki program yang digambarkan dalam *Hierarchy plus Input-Process Output (HIPO)*:



Gambar V.14 HIPO Sistem Informasi *Daily Defect* yang Diusulkan


(Sumber: Hasil Analisis Data, 2017)

5.12. Perancangan Interface Sistem Informasi Pelaporan *Daily Defect*

Perancangan *interface* merupakan tahapan untuk membuat tampilan atau *design* dari Sistem Informasi Pelaporan *Daily Defect*. Rancangan tampilan yang dibuat meliputi beberapa rancangan *input* (berupa *form input*) dan rancangan *output* (berupa dokumen). Rancangan *interface* dalam Sistem Informasi Pelaporan *Daily Defect* tersebut adalah sebagai berikut :

1. Rancangan Form Login

Form Login digunakan untuk membedakan hak akses pengguna. Melalui *Form Login* ini pengguna yang boleh masuk sistem adalah pengguna yang mengetahui *username* dan *password* atau pengguna yang memiliki wewenang untuk menggunakan sistem. Rancangan *Form Login* ini dapat dilihat pada Gambar V.13.


PT KRAMA YUDHA RATU MOTOR
 Username
 Password

Gambar V.15 Rancangan Form Login

(Sumber: Hasil Analisis Data, 2017)

2. Rancangan Menu Utama

Menu utama ini dirancang untuk memudahkan user dalam mengakses aplikasi dengan pilihan-pilihan menu yang sudah disediakan sesuai dengan kebutuhan dan hak akses karena tidak semua menu dapat digunakan.

Sistem Informasi <i>Daily Defect</i> Unit Kendaraan Tipe TD di PT Krama Yudha Ratu Motor			
Data Master	<i>Countermeasure</i>	Laporan <i>Countermeasure</i>	<i>Logout</i>
<i>Copyright@2017</i>			

Gambar V.16 Rancangan Menu Utama

(Sumber: Hasil Analisis Data, 2015)

3. Rancangan Data Master

Menu Data master terdiri dari submenu Master User dan Master Komponen. Berikut ini penjelasannya :

- 1.) Submenu Master User merupakan form yang digunakan untuk menginput data mengenai master user. Master user ini menjelaskan aktor yang mempunyai hak akses untuk bisa masuk ke dalam sistem. Adapun rancangan data user pada Sistem Informasi *Daily Defect* dapat dilihat pada Gambar V.15.

MASTER USER

username :

Nama Lengkap :

Bagian :

Alamat :

No telp :

username	Nama lengkap	Bagian	Alamat	No Telp

Gambar V.17 Rancangan Master User

(Sumber: Hasil Analisis Data, 2017)

- 2.) Submenu Master Komponen merupakan form yang digunakan untuk menginput data mengenai master komponen. Master komponen menjelaskan aktor yang mempunyai hak akses untuk bisa masuk ke dalam sistem. Adapun rancangan data Komponen pada Sistem Informasi *Daily Defect* dapat dilihat pada Gambar V.16.

MASTER KOMPONEN

Id_Komponen :

Nama Komponen :

Type Komponen :

Id_komponen	Nama_Komponen	Type Komponen

Gambar V.18 Rancangan Master Komponen

(Sumber: Hasil Analisis Data, 2017)

3.) Menu Countermeasure

Menu Countermeasure terdiri Data Dokumen Countermeasure Berikut ini penjelasannya:

Form ini diinput oleh bagian Trimming untuk memasukkan data Countermeasure ke dalam database sistem. Gambar V.16 menunjukkan rancangan Countermeasure.

COUNTERMEASURE DAILY DEFECT FORM

No Dokumen :

Kode Dokumen :

Tanggal Berlaku :

Revisi :

Tipe :

To :

Tambah Kecacatan

(What/Where) Problem (Scetch) :

QTY :

Why Investigation :

(How) Countermeasure or Action :

(When) Schedule :

(WHO) Incharge :

Evaluation :

(What/Where) Problem (Scetch)	QTY	Why Investigation	(How) Countermeasure or Action	(When) Schedule	(WHO) Incharge	Evaluation

Simpan

Copyright@2017

Gambar V.19 Menu Countermeasure

(Sumber: Hasil Analisis Data, 2017)

5.13. Spesifikasi Kebutuhan *Software* dan *Hardware*

Untuk mendukung kelancaran sistem informasi di atas diperlukan perangkat lunak (*software*) dan perangkat keras (*hardware*) yang terdiri dari sistem operasi dan aplikasi *database*. Adapun perangkat lunak (*software*) yang diperlukan adalah sebagai berikut:

1. Sistem Operasi: *Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7*
2. *Web Server: Apache*
3. *Database Server: MySQL 5.5.27*
4. *Processor: Pentium IV*
5. *Main Memory: 512 MB*
6. *Harddisk: 80 GB*
7. *Graphic Adapter (VGA) : 256 MB*
8. Bahasa Pemrograman: PHP
9. *Design Interface: PHP Designer 8*
10. *Web Browser: Internet Explorer, Mozilla Firefox, Google Chrome*

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berdasarkan penelitian yang dilakukan, penulis dapat memberikan kesimpulan sebagai berikut:

1. Sistem Informasi Pelaporan *Daily Defect* dapat memudahkan proses pembuatan laporan bulanan dokumen Countermeasure oleh pihak Trimming 1 tanpa harus menunggu penyerahan dokumen Countermeasure dari pihak Quality Inspection (QI).
2. Sistem Informasi Pelaporan *Daily Defect* juga dapat memudahkan pihak Trimming 1 dalam mengakses laporan bulanan Countermeasure sehingga dapat meminimalisir kendala dari pihak Trimming 1 tanpa harus mencari laporan Countermeasure periode bulanan di email yang masuk.

6.2. Saran

Berdasarkan analisis terhadap berbagai permasalahan yang ada pada bagian Trimming 1 dan bagian Quality Inspection, penulis membuat beberapa saran yang diharapkan untuk pemecahan masalah tersebut yaitu:

1. Untuk menerapkan Sistem Informasi Pelaporan *Daily Defect* dokumen Countermeasure dan laporan bulanan Countermeasure.
2. Menerapkan Sistem Informasi pada Trimming 1 dan Quality Inspection agar dapat lebih mudah dalam melakukan proses pengecekan *Daily Defect*.

DAFTAR PUSTAKA

Abdul Kadir.2009.*Membuat Aplikasi Web dengan PHP+Database MySQL*.Andi.
Yogyakarta.

Akakom, STMIK. 2015. *Modul 1 Mengenal Unified Modeling Language*.
(<http://ns.akakom.ac.id/.../Microsoft%20Word%20-%20MODUL%20I.pdf>,
diakses tanggal 2 Agustus 2017).

Departemen Pendidikan Nasional.2002.*Kamus Besar Bahasa Indonesia*. Jakarta:
Gramedia.

Elliot.(1993). “*Management of Quality in Computing Systems Education: ISO
9000 series Quality Standards Applied*”.*Journal of System Management*.
September, 6-11.

Jogiyanto. 2005. *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur
Teori dan Praktek Aplikasi Bisnis*. Edisi Ketiga. Yogyakarta: Andi Offset.

Kadir, Abdul. 2009. *Pengenalan Sistem Informasi*. Yogyakarta: Andi Offset.

Mustak ini.2009.*Sistem Informasi Teknologi*. Yogyakarta: Andi Offset.

Munawar. 2005. *Pemodelan dengan UML*. Yogyakarta: Graha Ilmu.

McLeod Pearson.2008.*Sistem Informasi Manajemen*.Salemba. Jakarta.

Niswonger. 1999.*Prinsip-Prinsip Akuntansi*.

(<http://repository.usu.ac.id/bitstream/123456789/22583/3/Chapter%20II.pdf>, diakses tanggal 2 September 2017).

Robert K. Leitch dan K. Roscoe Davis, 1983, *Accounting Information Systems*, Prentice-Hall, New Jersey.

Rosa, A.S., & Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Cetakan Pertama. Bandung: Modula.

Saputra, Agus. 2012. *Sistem Informasi Nilai Akademik untuk Panduan Skripsi*. Cetakan Kedua. Jakarta: PT Elex Media Komputindo.

Sutabri, Tata. 2012. *Konsep Sistem Informasi*. Yogyakarta: Andi Offset.

Telkomuniversity. 2017. *Class Diagram*.

(<http://ftass.telkomuniversity.ac.id>, diakses tanggal 3 Agustus 2017).

Unair, DSI. 2015. *Tipe-Tipe Data di MySQL*.

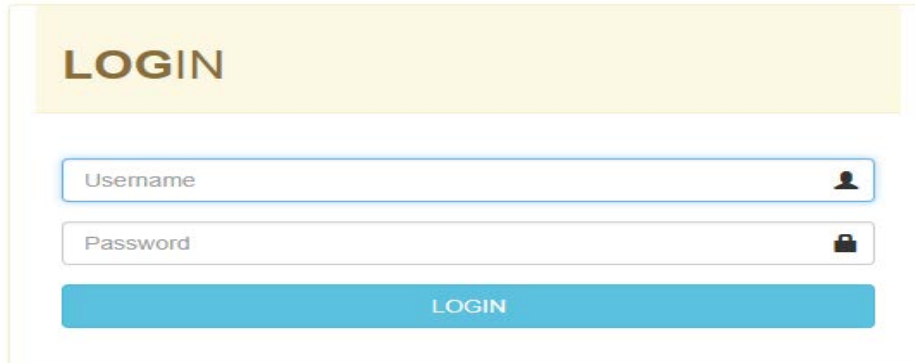
(http://nimos-fisip11.web.unair.ac.id/artikel_detail-59844-kuliah-Tipe%20%20tipe%20data%20di%20My%20SQL.html, diakses tanggal 7 Agustus 2017).

Widigdo, Anon K. 2003.*Dasar Pemrograman PHP dan MySQL*.

(<http://ikc.dinus.ac.id/umum/anon-phpmysql.php>, diakses tanggal 7 Agustus 2017).

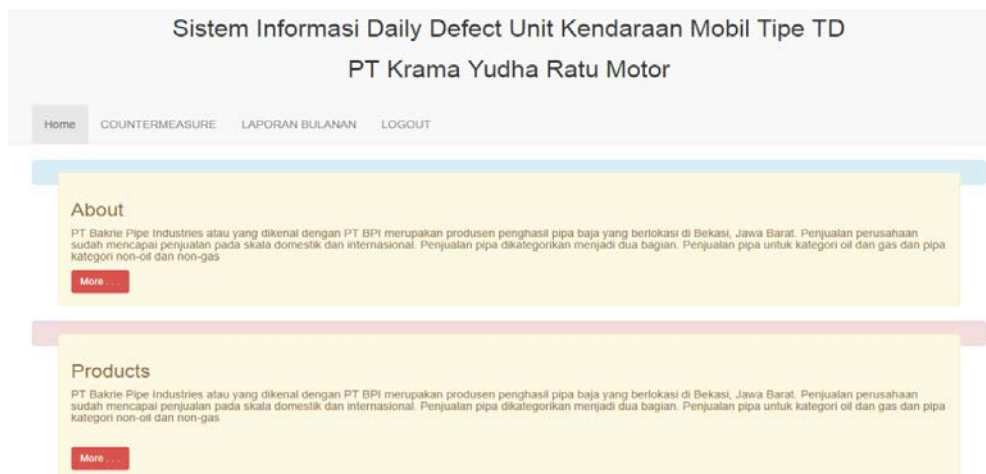
LAMPIRAN

1. Tampilan Login



The login page features a yellow header with the word "LOGIN" in bold black text. Below the header is a white form with two input fields: "Username" and "Password". The "Username" field has a user icon on the right, and the "Password" field has a lock icon. A blue "LOGIN" button is positioned below the form.

2. Tampilan Menu Utama



The main menu page has a grey header with the title "Sistem Informasi Daily Defect Unit Kendaraan Mobil Tipe TD PT Krama Yudha Ratu Motor". Below the header is a navigation bar with links: "Home", "COUNTERMEASURE", "LAPORAN BULANAN", and "LOGOUT". The main content area is divided into two sections: "About" and "Products". Each section contains a brief description of PT Bakrie Pipe Industries and a "More" button.

About

PT Bakrie Pipe Industries atau yang dikenal dengan PT BPI merupakan produsen penghasil pipa baja yang berlokasi di Bekasi, Jawa Barat. Penjualan perusahaan sudah mencapai penjualan pada skala domestik dan internasional. Penjualan pipa dikategorikan menjadi dua bagian. Penjualan pipa untuk kategori oil dan gas dan pipa kategori non-oil dan non-gas

[More](#)

Products

PT Bakrie Pipe Industries atau yang dikenal dengan PT BPI merupakan produsen penghasil pipa baja yang berlokasi di Bekasi, Jawa Barat. Penjualan perusahaan sudah mencapai penjualan pada skala domestik dan internasional. Penjualan pipa dikategorikan menjadi dua bagian. Penjualan pipa untuk kategori oil dan gas dan pipa kategori non-oil dan non-gas

[More](#)

3. Tampilan Master User

Sistem Informasi Daily Defect Unit Kendaraan Mobil Tipe TD
PT Krama Yudha Ratu Motor

Home MASTER ▾ LOGOUT

Data Master User

Tambah

Show 10 ▾ entries

Search:

No	Username	Nama Lengkap	Bagian	Alamat	No Telepon	ACTION
1	ijul	panjul	Quality Inspection	bunga	89005678	edit hapus
2	ika	ika lestari	Trimming 1	ketapang	083889992	edit hapus
3	miko	Miko Suhendra	Administrator	Jalan Pengangsaan Timur IX. No.13	0987655555	edit hapus

Showing 1 to 3 of 3 entries

Previous 1 Next

4. Tampilan Master Komponen

Sistem Informasi Daily Defect Unit Kendaraan Mobil Tipe TD
PT Krama Yudha Ratu Motor

Home MASTER ▾ LOGOUT

Data Master Komponen

Tambah

Show 10 ▾ entries

Search:

No	ID Komponen	Nama Komponen	Type Komponen	ACTION
1	K0001	Stir Mobil	TD	edit hapus
2	K0002	Jok Mobil	TD	edit hapus
3	K0003	pintu mobil	CJM	edit hapus

Showing 1 to 3 of 3 entries

Previous 1 Next

5. Tampilan Countermeasure

Sistem Informasi Daily Defect Unit Kendaraan Mobil Tipe TD
PT Krama Yudha Ratu Motor

Home COUNTERMEASURE LAPORAN BULANAN COUNTERMEASURE LOGOUT

Data Countermeasure

Tambah

No	ID		Tgl Berlaku	Revisi	Type Barang	To Perbaikan	ACTION			
	Countermeasure	No Document					edit	detail	hapus	cetak
1	C0001	FRM-QC.17.0001	2017-09-08	0	TD	Trimming 1	edit	detail	hapus	cetak
2	C0002	FRM-QC.17.0002	2017-09-09	0	TD		edit	detail	hapus	cetak

6. Tampilan Laporan Bulanan Countermeasure

Sistem Informasi Daily Defect Unit Kendaraan Mobil Tipe TD
PT Krama Yudha Ratu Motor

Home LAPORAN BULANAN COUNTERMEASURE LOGOUT

LAPORAN BULANAN COUNTERMEASURE X

Tgl From

Tgl To