

**RANCANG BANGUN SISTEM INFORMASI PEMROSESAN
LINE DAMAGE CARD PADA BAGIAN PRODUKSI *TRIMMING*
BERBASIS WEB
PADA PT KRAMA YUDHA RATU MOTOR**

TUGAS AKHIR

Diajukan Untuk Memenuhi Syarat Program Sarjana Terapan Pada Program Studi
Sistem Informasi Industri Otomotif Politeknik STMI Jakarta

**OLEH
DANIEL RAY ELBRAM
1311059**



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2018**

LEMBAR PERSETUJUAN PEMBIMBING
POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN R.I

TANDA PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : **RANCANG BANGUN SISTEM INFORMASI
PEMROSESAN LINE DAMAGE CARD PADA
BAGIAN PRODUKSI TRIMMING BERBASIS
WEB PADA PT KRAMA YUDHA RATU MOTOR**

Disusun Oleh

Nama : Daniel Ray Elbram
Nim : 1311059
Program Studi : Sistem Informasi Industri Otomotif

Telah diperiksa dan disetujui untuk diajukan dalam sidang Tugas Akhir Politeknik
STMI Jakarta

Jakarta, 3 September 2018

Pembimbing

Drs. Jacob Saragih, MM
NIP. 195404281986031002

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Daniel Ray Elbram

Nim : 1311059

Program Studi : Sistem Informasi Industri Otomotif

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

**“RANCANG BANGUN SISTEM INFORMASI PEMROSESAN
LINE DAMAGE CARD PADA BAGIAN PRODUKSI TRIMMING
BERBASIS WEB PADA PT KRAMA YUDHA RATU MOTOR”**

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

Jika terbukti saya tidak memenuhi apa yang telah dinyatakan di atas, maka saya bersedia menerima sanksi atas apa yang telah saya lakukan sesuai peraturan yang berlaku.

Jakarta, 3 September 2018

Daniel Ray Elbram

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Daniel Ray Elbram
NIM : 1311059
Judul TA : RANCANG BANGUN SISTEM INFORMASI PEMROSESAN LINE
DAMAGE CARD PADA BAGIAN PRODUKSI TRIMMING
BERBASIS WEB PADA PT KRAMA YUDHA RATU MOTOR
Dosen Pembimbing : Drs. Jacob Saragih, MM

Tanggal	Keterangan	Paraf
9 Juli 2018	Bimbingan BAB I	
12 Juli 2018	Revisi BAB I	
13 Juli 2018	Bimbingan BAB II	
17 Juli 2018	Bimbingan BAB III	
19 Juli 2018	Revisi BAB III	
20 Juli 2018	Bimbingan BAB IV	
23 Juli 2018	Revisi BAB IV	
26 Juli 2018	Bimbingan BAB V & Demo Program	
30 Juli 2018	Revisi BAB V	
3 Agustus 2018	Evaluasi BAB I - VI	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Dosen Pembimbing

(Drs. Jacob Saragih, MM)
NIP. 195404281986031002

(Drs. Jacob Saragih, MM)
NIP. 195404281986031002

ABSTRAK

PT Krama Yudha Ratu Motor adalah perusahaan yang bergerak di bidang perakitan kendaraan bermotor. Saat ini sistem pemrosesan *line damage card* yang berjalan pada PT Krama Yudha Ratu Motor masih dilakukan secara manual. Sistem pemrosesan *line damage card* ini melakukan pembuatan data dengan cara tulis tangan. Selain itu, sistem ini juga mengandalkan data fisik berupa formulir yang dapat hilang atau rusak sewaktu-waktu. Untuk mengatasi permasalahan tersebut maka perlu dirancang sebuah sistem informasi yang nantinya dapat memudahkan dalam mengadakan serta merancang pemrosesan *line damage card* pada PT Krama Yudha Ratu Motor. Untuk membuat sistem tersebut maka diperlukan pengumpulan data dengan cara pengamatan pada bagian *Part Control* di PT Krama Yudha Ratu Motor. Data yang telah diperoleh selanjutnya akan diolah menjadi informasi yang diperlukan. Metodologi pengembangan sistem informasi yang digunakan pada penelitian ini adalah metodologi *evolutionary prototype*. Kemudian dalam merancang sistem informasi pemrosesan *line damage card* pemodelan sistem menggunakan *unified modelling language* (UML). Sistem informasi pemrosesan *line damage card* dibangun dengan menggunakan PHP 5.4.4 sebagai bahasa pemrograman, dan MySQL 5.5.25 sebagai basis datanya. Sistem informasi pemrosesan *line damage card* ini nantinya dapat memudahkan dalam proses pembuatan laporan dan memberikan informasi yang berhubungan dengan pemrosesan *line damage card*.

Kata Kunci: sistem informasi, pemrosesan line damage card, prototipe evolusioner, UML, PHP, MySQL.

KATA PENGANTAR

Segala puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan nikmat-Nya, terutama nikmat iman dan kesehatan sehingga Tugas Akhir ini dapat diselesaikan dengan baik dan lancar.

Tugas Akhir ini disusun untuk memenuhi persyaratan akademis yang telah ditetapkan oleh Politeknik STMI dengan bobot 4 SKS. Penulis mengambil pokok bahasan untuk laporan dengan judul “Rancang Bangun Sistem Informasi Pemrosesan *Line Damage Card* Pada Bagian Produksi Trimming Berbasis Web Pada PT Krama Yudha Ratu Motor”. Penulisan ini juga merupakan penerapan atau latihan untuk mengaplikasikan teori-teori yang pernah penulis dapatkan pada mata kuliah yang dipelajari dibangku kuliah dalam kenyataan dilapangan.

Dalam penulisan laporan ini, penulis banyak mendapatkan bantuan, bimbingan, petunjuk dan dukungan dari berbagai pihak. Dengan ketulusan dan kerendahan hati, perkenankanlah penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa yang telah memberikan berkat dan kekuatan.
2. Kedua orang tua, bapak Tihar Hamonangan Napitupulu dan ibu Elfrida Hutagalung serta keluarga saya yang telah memberikan doa, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
3. Bapak Dr. Mustofa, ST, MT selaku Direktur Politeknik STMI Jakarta.
4. Bapak Drs. Jacob Saragih, MM selaku dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
5. Ibu Noveriza Yuliasari, S.Si, MT Selaku sekretaris Program Studi Sistem Informasi di Sekolah Tinggi Manajemen Industri yang telah membantu mahasiswanya.
6. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.

7. Seluruh teman-teman jurusan Sistem Informasi angkatan 2010 - 2014 yang telah memberikan semangat dan motivasi kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
8. Seluruh teman-teman Bori Pictures, Bienvenidos, Rangers, yang telah memberikan motivasi.
9. Bapak Fega, Bapak Rusdi dan Bapak Anjar serta seluruh pegawai PT Krama Yudha Ratu Motor yang telah memberi arahan serta informasi untuk pengumpulan data
10. Serta semua pihak yang telah membantu penulis dan yang tidak dapat disebutkan satu per satu.

Dengan segala kemampuan dan keterbatasan, penulis menyadari segala kekurangan yang ada dalam penulisan, karena itu penulis sangat mengharapkan segala kritik atau saran yang dapat membangun dari semua pihak. Dan juga berharap penulisan ini dapat berguna bagi diri pribadi maupun pihak-pihak lain yang membacanya.

Jakarta, 3 September 2018

Penulis

(Daniel Ray Elbram)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di era yang serba canggih dan cepat ini membuat persaingan semakin ketat. Perusahaan dalam mempertahankan produknya agar tetap bisa bersaing, dituntut untuk bisa lebih kreatif dan inovatif. Perusahaan harus memiliki cara masing-masing agar produk-produk mereka dapat terus bertahan ditengah persaingan yang ketat. Produksi di dalam sebuah perusahaan merupakan jantung dari perusahaan. Oleh karena itu proses produksi harus lebih diperhatikan dalam sebuah perusahaan. Karena produksi merupakan elemen yang penting pada perusahaan. Proses produksi merupakan kegiatan yang bisa menambah nilai guna suatu benda atau menciptakan benda baru sehingga lebih bermanfaat bagi semua orang.

PT KRM (Krama Yudha Ratu Motor) merupakan perusahaan yang bergerak di bidang manufaktur, yang kegiatannya bergerak dibidang perakitan kendaraan roda 4 atau lebih. PT KRM (Krama Yudha Ratu Motor) terbagi ke dalam beberapa bagian. Masing-masing bagian akan menentukan keberhasilan proses produksi. Salah satu bagian yang dimiliki adalah bagian Produksi yang salah satunya bertanggung jawab terhadap kerusakan atau kehilangan *part* untuk kebutuhan produksi.

Dalam menjalankan kegiatannya, bagian Produksi tidak menggunakan aplikasi khusus untuk pemrosesan data kerusakan atau kehilangan *part*. Selama ini, bagian *Part Control* menerima *form* rencana produksi PC/PPC dari bagian PPC, *form* Rencana Produksi PC/PPC merupakan acuan bagi bagian *Part Control* untuk menyiapkan dan mengirimkan *part* yang dibutuhkan untuk perakitan kendaraan sesuai dengan jumlah kendaraan yang akan diproduksi setiap hari. Kemudian karyawan *Part Control* melakukan pengiriman *part* sesuai *form* Rencana Produksi PC/PPC serta mengirimkan *part* ke Bagian Produksi. *Form*

Rencana Produksi PC/PPC tersebut kemudian disimpan sebagai arsip pada lemari dokumen yang digunakan sebagai laporan perbulan. *Form Line Damage Card* merupakan *form* permintaan penggantian *part* yang hilang atau rusak pada saat proses perakitan, yang berisi catatan komplain komponen *part* yang rusak atau hilang yang diberikan untuk bagian *Part Control* untuk mengirimkan *Part* pengganti. Selanjutnya, karyawan *Part Control* akan memberikan *part* pengganti ke Bagian Produksi serta menyimpan *form Line Damage Card*.

Penyimpanan *form Line Damage Card* dilakukan dengan cara disimpan pada lemari dokumen. Sehingga, menyulitkan saat pencarian dan riskan akan terjadinya kehilangan data. Selain itu, Proses Pembuatan *form line damage card* masih menggunakan tulis tangan, sehingga membutuhkan waktu yang lama untuk pembuatannya. dan masih dalam bentuk dokumen fisik yang membutuhkan persetujuan dari pihak-pihak yang terlibat seperti meminta persetujuan dari Kepala Bagian Produksi dan Manager Bagian Produksi. Mengingat lokasi yang berbeda antara *line* produksi dan *office* Departemen Produksi, merupakan sebuah kendala karena untuk mengantarkan *form line damage card* memerlukan waktu yang cukup lama, sehingga akan menyita lebih banyak waktu. Proses Pembuatan *form line damage card* masih menggunakan tulis tangan, sehingga membutuhkan waktu yang lama untuk pembuatannya.

Untuk membantu dalam mendukung pemrosesan *line damage card* pada bagian *Trimming*, diperlukan suatu sistem informasi yang dapat menangani permasalahan tersebut. Adapun judul Tugas Akhir ini adalah “RANCANG BANGUN SISTEM INFORMASI PEMROSESAN *LINE DAMAGE CARD* PADA BAGIAN *TRIMMING* BERBASIS WEB PADA PT KRAMA YUDHA RATU MOTOR”.

1.2 Pokok Permasalahan

Berdasarkan latar belakang di atas, maka dapat diidentifikasi permasalahan yang terjadi pada bagian Produksi *Trimming* pada PT KRM (Krama Yudha Ratu Motor) adalah sebagai berikut:

1. Proses penyimpanan *form line damage card* pada bagian PPC masih manual menggunakan lemari dokumen sehingga memungkinkan data akan rusak atau hilang.
2. Proses pengisian data yang dilakukan oleh *foreman* produksi pada *form line damage card* yang memakan waktu dan masih menggunakan tulisan tangan.
3. Proses persetujuan *form Line Damage Card* masih menggunakan cara manual seperti datang ke tempat tujuan untuk menyampaikan data yang diperlukan dan meminta persetujuan dari kepala produksi dan manager produksi sehingga memperlambat proses penggantian *part* yang baru.

1.3 Tujuan Penelitian

Adapun tujuan penelitian Tugas Akhir pada bagian Produksi *Trimming* di PT KRM (Krama Yudha Ratu Motor) adalah merancang dan membangun suatu sistem informasi yang mampu:

1. Menjadikan *database* sebagai tempat untuk penyimpanan dan pencarian data historis *line damage card*.
2. Untuk membantu dan mempercepat proses pengisian data pada *form line damage card*.
3. Untuk membantu dan mempercepat proses dalam meminta persetujuan pada *form line damage card*.

1.4 Batasan Masalah

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

1. Penelitian dilakukan di bagian *Part Control* pada PT KRM (Krama Yudha Ratu Motor).
2. Penelitian ini dilakukan sejak tanggal 1 Desember – 23 Desember 2016.
3. Sistem informasi ini dimulai dari diterimanya Form Rencana Produksi PC/PPC dari bagian *PPC* ke bagian *Part Control* sampai ke penggantian

Part.

4. Sistem Informasi ini tidak membahas pelaporan *part* rawan hilang atau rusak, yang terkait dengan *form line damage card* pada bagian Produksi *Trimming*.
5. Rancang bangun sistem informasi pemrosesan *line damage card* ini menggunakan PHP 5.4.4 dan MySQL 5.5.25.

1.5 Manfaat Tugas Akhir

Beberapa manfaat yang diperoleh dalam pelaksanaan penelitian dan penyusunan Tugas Akhir adalah:

1. Bagi perusahaan
Rancang Bangun Sistem Informasi Pemrosesan *Line Damage Card* diharapkan dapat mempermudah kegiatan pelaporan data *part* rusak atau hilang khususnya pada bagian Produksi *Trimming* di PT Krama Yudha Ratu Motor serta dapat memberikan kontribusi untuk perbaikan dan pengembangan sistem informasi *part* pada perusahaan itu ataupun pada perusahaan sejenis lainnya.
2. Bagi penulis
Rancang Bangun Sistem Informasi Pemrosesan *Line Damage Card* diharapkan dapat menjadi saran untuk kebenaran teori selama perkuliahan terhadap aplikasinya pada perusahaan sehingga penulis dapat merumuskan dan mencari solusi penyelesaian masalah yang terjadi di dalam perusahaan secara praktis dan menerapkannya di dalam dunia kerja.
3. Bagi pihak lain
Pihak lain dalam hal ini adalah para pengambil keputusan sehingga rancang Bangun Sistem Informasi Pemrosesan *Line Damage Card* diharapkan dapat digunakan sebagai sumbangan pemikiran dan juga sebagai referensi bagi peneliti lain yang mengadakan penelitian serupa.

1.6 Sistematika Penulisan

Sistematika penulisan ini ditujukan untuk memberikan gambaran sistematis yang terdiri dari:

BAB I PENDAHULUAN

Bab ini merupakan bagian pendahuluan yang membahas tentang latar belakang masalah, permasalahan, tujuan penelitian, pembatasan masalah, manfaat penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan berbagai teori yang diperoleh dari buku-buku literatur ataupun berbagai macam referensi. Teori yang dipaparkan pada Rancang Bangun Sistem Informasi Pemrosesan *Line Damage Card* seperti pengertian sistem informasi, pengembangan sistem informasi, bahan baku, Produksi, bagan alir (*flowchart*), *Unified Modelling Language* (UML), ERD, kamus data, HIPO, konsep dasar *Hypertext Preprocessing (PHP)*, *MySQL* dan *XAMPP*.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang jenis dan sumber data serta kerangka penelitian. Adapun kerangka penelitian ini sebagai berikut: identifikasi masalah, studi pendahuluan, penggunaan *prototypeevolutionary* serta kesimpulan dan saran.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini membahas tentang profil perusahaan, jenis produk, proses sistem *Line Damage Card* yang sedang berjalan menggunakan *flowchart* dan UML.

BAB V ANALISIS DAN PEMBAHASAN

Bab ini menguraikan analisis yang dirinci tentang proses sistem *Line Damage Card* ke *line* produksi yang diusulkan, model sistem yang digunakan *Unified Modelling Language* (UML), dan perancangan *interface* program.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil pengembangan sistem dan saran-saran yang sekiranya perlu disampaikan sebagai masukan bagi perusahaan.

BAB II

LANDASAN TEORI

2.1 Sistem

Menurut Jogiyanto (2005), sistem adalah “kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”. Suatu sistem mempunyai maksud tertentu, ada yang menyebutkan maksud dari sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai suatu sasaran (*objectives*). Tujuan biasanya dihubungkan dengan ruang lingkup yang lebih luas dan sasaran dalam ruang lingkup yang lebih sempit.

Bila merupakan suatu sistem utama, seperti misalnya bisnis, maka istilah *goal* lebih tepat diterapkan. Untuk sistem akuntansi atau sistem-sistem yang lainnya merupakan bagian atau subsistem dari sistem bisnis, maka istilah *objectives* yang lebih tepat. Jadi tergantung dari ruang lingkup dan cara memandang sistem tersebut, seringkali tujuan (*goal*) dan sasaran (*objectives*) digunakan bergantian dan tidak dibedakan.

2.1.1 Karakteristik Sistem

Menurut Jogiyanto (2005), suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu yaitu komponen (*components*), batas sistem (*boundary*), lingkungan sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), sasaran (*objectives*), dan pengolah (*process*), antara lain:

1. 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 subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan sistem (*boundary*)

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

3. Lingkungan luar (*environment*)

Lingkungan dari sistem adalah segala sesuatu di luar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat 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, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem (*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 (*output*) dari suatu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya dengan 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 A adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya, dan data adalah *signal input* untuk diolah menjadi informasi. Pada sistem informasi, masukan dapat berupa data transaksi, data non transaksi (contoh: surat pemberitahuan) dan instruksi.

6. Keluaran (*output*)

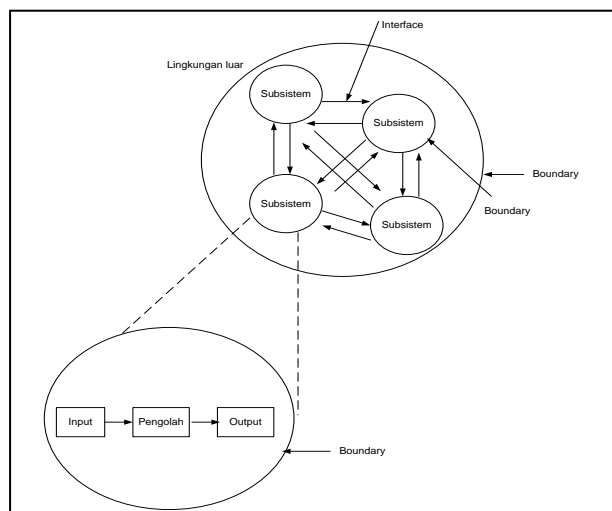
Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran merupakan masukan untuk subsistem yang lain atau kepada suprasistem. Hasil dari pemrosesan, dapat berupa keluaran yang berguna (informasi, produk) atau keluaran yang tidak berguna (limbah). Pada sistem informasi, keluaran dapat berupa: informasi, saran, dan cetakan laporan.

7. Sasaran (*objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Apabila 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.

8. Pengolahan (*process*)

Suatu sistem yang mempunyai suatu bagian 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. Karakteristik sistem dapat dilihat pada Gambar II.1.



Gambar II.1 Karakteristik Sistem
(Sumber:Jogiyanto, 2005)

2.1.2 Klasifikasi Sistem

Menurut Jogiyanto (2005), sistem dapat diklasifikasikan dari beberapa sudut pandang diantaranya adalah sebagai berikut:

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem akuntansi, dan sistem produksi.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak dibuat manusia, misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasi sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

2.2 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 merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

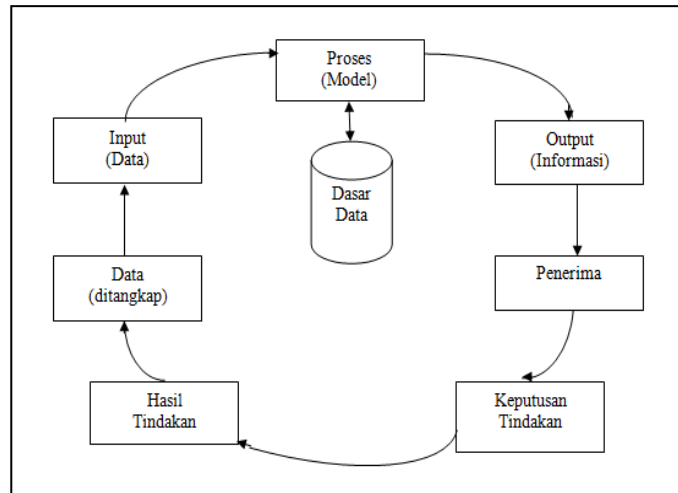
2.2.1 Kualitas Informasi

Kualitas Informasi tergantung pada 3 hal, antara lain (Jogiyanto, 2005):

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*).
2. Tetap pada waktunya, berarti informasi yang datang kepada penerima tidak boleh terlambat. Hal itu disebabkan karena informasi merupakan dasar atau landasan dalam pengambilan keputusan.
3. Relevan, berarti Informasi yang disampaikan mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang atau pemakai satu dengan lainnya dapat berbeda.

2.2.2 Siklus Informasi

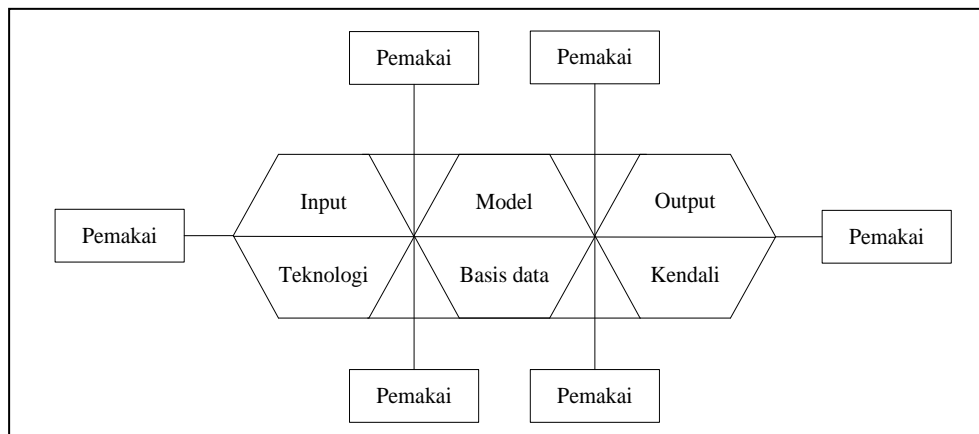
Menurut Jogiyanto (2005), agar data menjadi lebih berarti dan berguna dalam bentuk informasi maka perlu diolah menjadi suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan 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*). Siklus informasi dapat dilihat pada Gambar II.2



Gambar II.2 Siklus Informasi
(Sumber:Jogiyanto, 2005)

2.3 Sistem Informasi

Menurut Jogiyanto (2005), sistem informasi adalah suatu sistem di dalam 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. Komponen-komponen blok sistem informasi dapat dilihat pada Gambar II.3.



Gambar II.3 Blok Sistem Informasi yang Berinteraksi
(Sumber:Jogiyanto, 2005)

Keterangan:

1. Blok *input*
Input mewakili data yang masuk kedalam sistem informasi, *input* disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.
2. Blok model
Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.
3. Blok *output*
Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.
4. Blok teknologi
Teknologi merupakan *tool box* dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.
5. Blok basis data
Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut.
6. Blok kendali
Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, dan air. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah dan ataupun bila terlanjur terjadi kesalahan-kesalahan dapat cepat diatasi.

2.4 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem adalah metode-metode, prosedur-prosedur, konsep-konsep pekerjaan, aturan-aturan yang akan digunakan sebagai pedoman bagaimana dan apa yang harus dikerjakan selama pengembangan.

Pendekatan sistem merupakan sebuah metodologi. Metodologi adalah cara yang direkomendasikan dalam melakukan sesuatu. Pendekatan sistem adalah metodologi dasar dalam memecahkan segala jenis masalah. Siklus hidup pengembangan sistem (*systems development life cycle*) adalah aplikasi dari pendekatan sistem bagi pengembang suatu sistem informasi (McLeod & Schell, 2008). Menurut Rosa dan Shalahuddin (2011), proses perancangan atau pengembangan sistem informasi dimulai dari pemilihan konsep sampai dengan implementasinya yang biasa disebut dengan istilah *Software Development Life Cycle* atau dapat disebut juga *System Development Life Cycle (SDLC)*. 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.

Tahapan-tahapan yang ada pada *Sistem Development Life Cycle* secara umum 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.

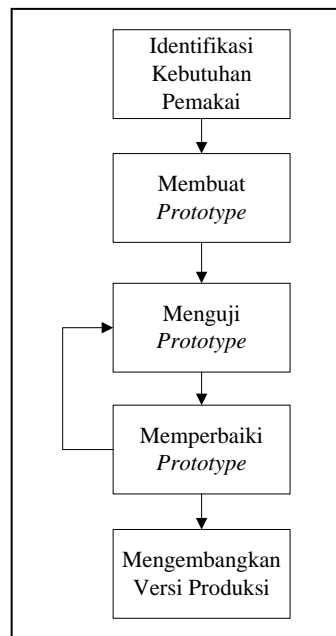
4. Analisis kebutuhan (*requirements analysis*)
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user* serta membuat dokumen-dokumen kebutuhan fungsional.
5. Desain (*design*)
Mentransformasikan kebutuhan *detail* menjadi kebutuhan yang sudah lengkap, dokumen 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 aktivitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

Dalam pengembangan sistem terdapat beberapa metodologi yang sudah dikenal. Berikut adalah macam-macam metodologi yang biasa digunakan pengembang yang diambil dari berbagai sumber:

1. *Prototype*

Menurut McLeod & Schell (2008), *prototype* adalah suatu versi dari sebuah sistem potensial yang memberikan ide bagi para pengembang dan calon pengguna, bagaimana sistem akan berfungsi dalam bentuk yang telah selesai. Dasar pemikirannya adalah membuat *prototype* secepat mungkin, bahkan dalam waktu semalam lalu memperoleh umpan balik dari pengguna yang akan memungkinkan *prototype* tersebut diperbaiki kembali dengan sangat cepat.

Mekanisme pengembangan sistem dengan *prototype* dapat dilihat pada Gambar II.4.



Gambar II.4 Mekanisme Pengembangan Sistem Dengan *Prototype*
(Sumber:McLeod & Schell, 2008)

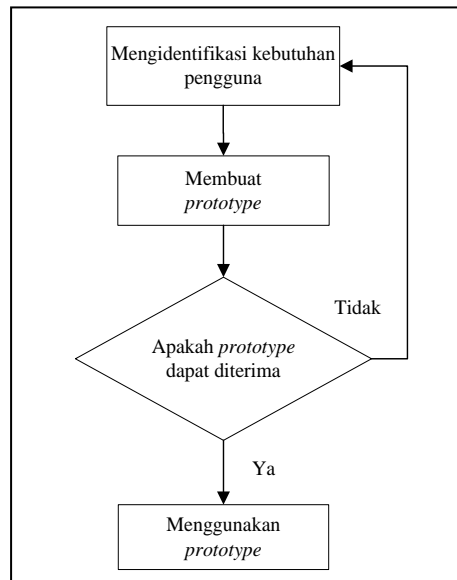
Keterangan:

a. Identifikasi kebutuhan pemakai

Tahap ini merupakan tahap dimana pengembang dan pemakai bertemu.

Pada tahap ini juga pemakai akan menjelaskan kebutuhan sistemnya.

- b. *Membuat prototype*
Pada tahap ini, pengembang akan mulai membuat *prototype* sesuai dengan kebutuhan pemakai.
 - c. *Menguji prototype*
Pada tahap ini, pemakai akan menguji *prototype* dan memberikan kritikan atau saran.
 - d. *Memperbaiki prototype*
Pada tahap ini, pengembang akan melakukan modifikasi sesuai dengan masukan pemakai. Setelah modifikasi selesai dilakukan, maka akan dilanjutkan ke tahap menguji *prototype*. Tahap ini akan terus berulang sampai *prototype* yang dibuat sesuai dengan kebutuhan pemakai.
 - e. *Mengembangkan versi produksi*
Tahap ini merupakan tahap terakhir dalam pembuatan *prototype* dimana pengembang merampungkan sistem sesuai dengan masukan terakhir dari pemakai.
2. *Evolutionary Prototype*
- Menurut McLeod dan Schell (2008), *Evolutionary prototype* adalah *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 daripada dibuang. *Prototype* yang dibuang biasanya digunakan dengan aspek sistem yang dimengerti secara luas dan dibangun atas kekuatan tahapan *evolutionary prototype*, Langkah-langkah dari *evolutionary prototype* dapat dilihat pada Gambar II.5.



Gambar II.5 *Evolutionary Prototype Model*
(Sumber: McLeod dan Schell, 2008)

Keterangan:

- 1) Mengidentifikasi kebutuhan pengguna
Pengembang mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.
- 2) Membuat *prototype*
Pengembang membuat *prototype* dari sistem yang telah dijelaskan oleh pengguna atau pemilik sistem.
- 3) Menentukan apakah *prototype* dapat diterima
Pengembang mendemonstrasikan *prototype* kepada pengguna atau pemilik sistem tentang *prototype* yang sudah dibuat untuk mengetahui apakah sesuai atau tidak dengan kebutuhan sistem.
- 4) Menggunakan *prototype*
Sistem mulai dikembangkan dengan *prototype* yang sudah dibuat.

2.5 Pengertian Produksi

Produksi adalah sesuatu yang dihasilkan oleh suatu perusahaan baik berbentuk barang (*goods*) maupun jasa (*services*) dalam suatu periode waktu yang selanjutnya dihitung sebagai nilai tambah bagi perusahaan. Bentuk hasil produksi dengan kategori barang (*goods*) dan jasa (*services*) sangat tergantung pada

kategori aktivitas bisnis yang dimiliki perusahaan yang bersangkutan. jika perusahaan *manufacture* (pabrik) sudah jelas produksi yang dihasilkan dalam bentuk barang, sedangkan untuk bisnis perhotelan, *travel*, pendidikan adalah berbentuk jasa.

Produksi diartikan sebagai kegiatan yang dapat menimbulkan tambahan manfaat atau faedah baru (Ahyari, 2011). Berikut pengertian produksi menurut beberapa ahli:

1. Produksi adalah kegiatan yang mentransformasikan masukan (*input*) menjadi keluaran (*output*), tercakup semua aktivitas atau kegiatan yang menghasilkan barang dan jasa, serta kegiatan-kegiatan lain yang mendukung atau menunjang usaha untuk menghasilkan produk tersebut yang berupa barang-barang atau jasa (Assauri, 2008).
2. Produksi merupakan fungsi pokok dalam setiap organisasi, yang mencakup aktivitas yang bertanggung jawab untuk menciptakan nilai tambah produk yang merupakan output dari setiap organisasi industri itu (Gaspersz, 2005).
3. Kegiatan produksi merupakan suatu kegiatan yang berhubungan dengan penciptaan atau pembuatan barang, jasa, atau kombinasinya, melalui proses transformasi dari masukan sumber daya produksi menjadi keluaran yang diinginkan (Herjanto, 2003).
4. Produksi adalah proses penciptaan barang baru (Heizer and Render, 2012)

2.5.1 Pengertian Proses Produksi

Proses produksi adalah Proses pengolahan *input* menjadi output yang dimaksud adalah bahan baku langsung, tenaga kerja langsung, dan biaya *overhead* pabrik yang diproses menjadi bahan produk selesai (Assauri, 2008).

Adapun konsep dasar sistem produksi terdiri dari (Ishak, 2010):

1. Elemen *input* dalam sistem produksi
 Pada dasarnya *input* dalam sistem produksi dapat diklasifikasikan ke dalam dua jenis, yaitu: *input* tetap dan *input variable*. *Input* tetap

didefinisikan sebagai suatu input bagi sistem produksi yang tingkat penggunaannya tidak tergantung pada jumlah output yang akan diproduksi. *Input variable* didefinisikan sebagai suatu input bagi sistem produksi yang tingkat penggunaannya bergantung pada jumlah *output* yang akan diproduksi.

2. Proses dalam sistem produksi

Proses dalam sistem produksi didefinisikan sebagai integrasi sekuensial dari tenaga kerja, material informasi, metode kerja, dan mesin atau peralatan, dalam suatu lingkungan guna menghasilkan nilai tambah bagi produk agar dapat dijual dengan harga kompetitif di pasar.

3. Elemen *output* dalam sistem produksi

Output dari proses dalam sistem produksi dapat berbentuk barang dan atau jasa, yang disebut produk.

2.5.2 Sifat Proses Produksi

Penggolongan proses produksi menurut sifat ini akan menentukan jenis atau bentuk pokok yang dipakai dalam pengolahan suatu produk. Berdasarkan sifatnya, proses produksi dapat dibedakan menjadi lima macam, yaitu (Sutabri, 2010) :

1. Proses Ekstraktif

Suatu proses produksi yang mengambil bahan-bahan langsung dari sumber alam. Proses ekstraktif ini terdapat dalam industri proses produksi dasar. Karena itu, pertanian dan perikanan juga disebut industri ekstraktif.

2. Proses Analitik

Suatu proses pemisahan dari suatu bahan menjadi beberapa macam barang yang hampir menyerupai bentuk atau jenis aslinya. Termasuk dalam kategori ini adalah penyulingan minyak.

3. Proses Fabrikasi

Proses fabrikasi atau sering disebut proses perubahan adalah suatu proses yang mengubah suatu bahan menjadi beberapa bentuk. Perubahan bentuk

tersebut dapat dilakukan dengan menggunakan berbagai macam mesin. Contohnya adalah proses pembuatan pakaian, sepatu, mebel dan sebagainya.

4. Proses Sintetik

Proses sintetik menunjukkan metode pengkombinasian beberapa bahan ke dalam suatu bentuk produk. Dalam pengolahan baja dan kaca, produk akhirnya sangat berbeda dengan jenis aslinya karena ada perubahan fisik atau kimia.

5. Proses *Assembling*

Proses *assembling* berarti merangkaikan beberapa produk jadi atau setengah jadi menjadi produk baru (barang baru) tanpa merubah bentuk fisik susunan kimiawinya. Termasuk dalam kategori ini adalah perusahaan alat listrik.

2.5.3 *Line Damage Card*

Line damage card adalah *form* yang berisi data kerusakan atau kehilangan *part* pada saat proses produksi. *Form* ini digunakan di PT Krama Yudha Ratu Motor sebagai syarat untuk dilakukannya penggantian terhadap *part* yang hilang atau rusak. Isi dari *form ini* berupa tanggal, nomor *part*, tipe kendaraan, nama *part*, kuantitas *part* yang rusak, penyebab kerusakan *part* dan penanggulangan untuk *part* yang rusak yang dilakukan sesuai dengan jenis kerusakan yang terjadi pada *part* tersebut.

2.6 **Bahan Baku**

Bahan baku adalah barang yang dibuat menjadi barang lain. Bahan baku merupakan bahan langsung (*direct material*), yaitu bahan yang membentuk suatu kesatuan yang tidak terpisahkan dari produk jadi. Bahan baku adalah bahan utama atau bahan pokok dan merupakan komponen utama dari suatu produk. Bahan baku biasanya lebih mudah ditelusuri dalam suatu produk dan harganya relatif tinggi dibandingkan dengan bahan pembantu. Misalnya produk kursi rotan bahan bakunya rotan. Adapun bahan pembantu dari produk kursi rotan, seperti: paku,

lem kayu dan lain-lain. Bahan pembantu (*indirect material*) merupakan bahan pelengkap yang melekat pada suatu produk. Bahan pembantu biasanya tidak mudah ditelusuri dalam suatu produk dan harganya relatif rendah dibandingkan dengan bahan baku (Nafarin, 2007).

2.6.1 Persediaan Bahan Baku

Menurut Rangkuti (2007), Pada prinsipnya persediaan mempermudah atau memperlancar jalannya operasi perusahaan yang harus dilakukan secara berturut-turut.

Sistem informasi persediaan barang adalah struktur interaksi manusia, peralatan metode-metode, dan kontrol-kontrol yang disusun untuk mencapai tujuan berikut :

- 1 Mendukung rutinitas kerja dalam suatu bagian di dalam suatu perusahaan.
- 2 Mendukung pembuatan keputusan untuk personil-personil yang mengatur dan bagian kontrol persediaan.
- 3 Mendukung persiapan laporan-laporan *internal* dan laporan *eksternal*.

Sistem persediaan mendukung rutin kerja dalam bagian kontrol persediaan, yaitu dengan menangkap dan mencatat data yang berhubungan dengan sistem persediaan, misalnya transaksi penerimaan bahan baku dan transaksi penggunaan bahan baku. Sistem persediaan bahan baku mendukung pembuatan keputusan untuk personil-personil yang mengatur gudang dan bagian kontrol persediaan bahan baku. Sistem persediaan bahan baku merupakan suatu sistem yang menjelaskan bagaimana transaksi penerimaan bahan baku dan transaksi penggunaan bahan baku yang berisi tentang status stok bahan baku itu sendiri yang dapat membantu meningkatkan produktifitas perusahaan.

2.6.2 Fungsi Persediaan Bahan Baku

Persediaan barang menurut fungsinya dibagi menjadi 3 (Tiga) jenis yaitu :

1. *Batch Stock* atau *Lot Size Inventory*
Persediaan yang diadakan karena membeli atau membuat bahan-bahan atau barang-barang dalam jumlah yang lebih besar dari jumlah yang dibutuhkan saat itu.
2. *Fluctuation Stock*
Persediaan yang diadakan untuk menghadapi fluktuasi permintaan konsumen yang tidak dapat diramalkan.
3. *Anticipation Stock*
Persediaan yang diadakan untuk menghadapi fluktuasi permintaan yang dapat diramalkan, berdasarkan pola musiman yang terdapat dalam satu tahun dan untuk menghadapi penggunaan atau penjualan atau permintaan yang meningkat.

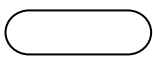
2.6.3 Tujuan Persediaan Bahan Baku

Persediaan bahan baku panel listrik dalam penelitian ini adalah suatu bagian proses awal dari serangkaian sistem proses administrasi untuk mempermudah karyawan menyelesaikan pekerjaannya dengan baik, efisien dan tepat sehingga hasil yang diinginkan tercapai dan memuaskan.

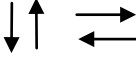

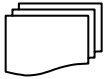



2.7 Diagram Alir (*Flowchart*)

Menurut Jogiyanto (2005), *flowchart* adalah bagan (*chart*) yang menunjukkan alur (*flow*) di dalam program atau prosedur sistem secara logika. Diagram alur digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Suatu *flowchart* memberi gambaran dua dimensi berupa simbol-simbol grafis, masing-masing simbol memiliki fungsi dan arti tersendiri. Simbol-simbol pada *flowchart* dapat dilihat pada Tabel II.1.

Tabel II.1 Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	Titik terminal	Menunjukkan awal dan akhir dari suatu proses.

Tabel II.1 Simbol-simbol *Flowchart* (Lanjutan)

Simbol	Nama	Keterangan
	<i>Arus/Flow</i>	Menunjukkan jalannya suatu arus proses.
	Dokumen	Menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik dan, computer
	Dokumen rangkap	Menggambarkan dokumen atau formulir asli dan tembusannya.
	Keputusan	Menggambarkan keputusan yang harus dibuat dalam proses pengolahan data.
	Tampilan <i>Output</i>	Menunjukkan <i>output</i> yang ditampilkan pada <i>monitor</i> .
	Arsip sementara	Menunjukkan tempat penyimpanan dokumen.

(Sumber: Jogyanto, 2005)

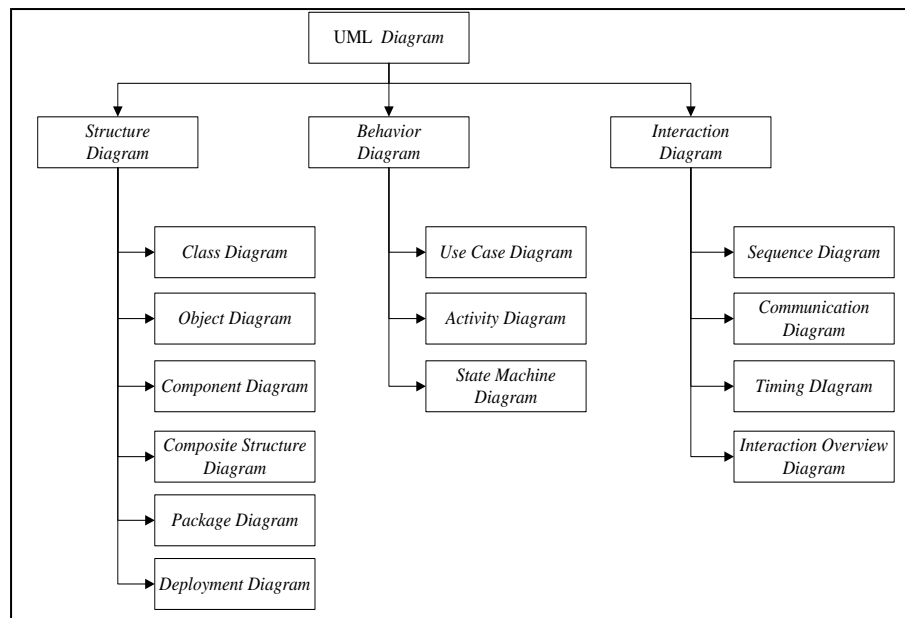
2.8 Unified Modelling Language (UML)

Menurut Rosa dan Shalahudin (2011), UML (*Unified Modelling Language*) merupakan 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. Adapun 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 terdiri dari 13 macam diagram yang dikelompokkan dalam tiga kategori, pembagian kategori tersebut dapat dilihat pada Gambar II.6.



Gambar II.6 Klasifikasi Diagram UML
(Sumber: Rosa dan Shalahudin, 2011)

Berikut ini penjelasan dari pembagian kategori tersebut, antara lain:

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

2.9 Kelompok Diagram pada UML

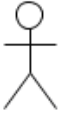


UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram

adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut.

2.9.1 Use Case Diagram

Menurut Rosa dan Shalahuddin (2011), *use case diagram* merupakan pemodelan untuk perilaku (*behavior*) sistem informasi yang akan dibuat. *Use case* menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Simbol-simbol pada *use case diagram* dapat dilihat pada Tabel II.2.

Tabel II.2 Simbol-simbol *Use Case Diagram*

Simbol	Nama	Keterangan
	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; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
	<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> .
	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.

Tabel II.2 Simbol-simbol *Use Case Diagram* (Lanjutan)

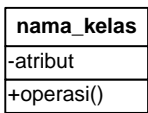


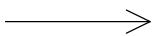



<p><<include>> -----></p>	<p><i>Include</i></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p>
<p>————></p>	<p>Generalisasi/ <i>Generalization</i></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p><<extend>> -----></p>	<p>Ekstensi/ <i>Extend</i></p>	<p>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 itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek.</p>

(Sumber:Rosa dan Shalahudin ,2011)

2.9.2 *Class Diagram*

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain.

Tabel II.3 Simbol-Simbol *Class Diagram*

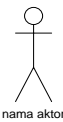
No.	Elemen	Simbol
1		Kelas
2		Antarmuka / <i>interface</i>
3		Asosiasi / <i>association</i>
4		Asosiasi berarah / <i>directed association</i>
5		Generalisasi
6		Kebergantungan / <i>dependency</i>
7		Agregasi / <i>aggregation</i>

(Sumber: Rosa dan Shalahudin , 2011)

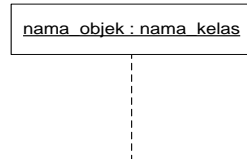

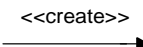
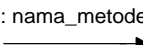
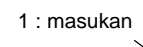
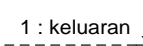
2.9.3 *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *use case*. Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Setiap *participant* terhubung dengan garis titik-titik yang disebut *lifeline*. Sepanjang *line* ada kotak yang disebut *activation*). Berikut ini adalah simbol-simbol yang digunakan pada *sequence diagram*:

Tabel II.4 Simbol-Simbol *Sequence Diagram*

No.	Elemen	Simbol
1		Aktor

Tabel II.4 Simbol-Simbol *Sequence Diagram* (Lanjutan)






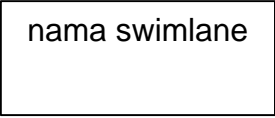
2		Objek
3		Waktu aktif
4		Pesan tipe create
5		Pesan tipe call
6		Pesan tipe send
7		Pesan tipe <i>return</i>

(Sumber: Rosa dan Shalahuddin, 2011)

2.9.4 Activity Diagram

Activity diagram adalah teknik untuk mendeskripsikan logika *procedural*, proses bisnis, dan aliran kerja dalam banyak kasus. Diagram ini dapat digunakan untuk memodelkan alir kerja, *use case* individual atau logika keputusan yang terkandung dalam metode individual. Berikut adalah simbol-simbol yang dapat digunakan pada saat pembuatan *activity diagram*:

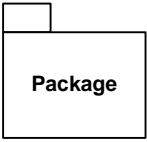
Tabel II.5 Simbol-Simbol *Activity Diagram*

No.	Elemen	Simbol
1		Status awal
2		Aktivitas
3.		Percabangan / <i>decision</i>
4.		Penggabungan / <i>join</i>
5.		Status akhir
6.		<i>Swimlane</i>

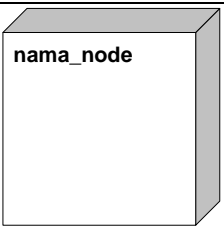


2.9.5 *Deployment Diagram*

Deployment diagram menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem terdiri dari *node-node* dimana setiap *node* diwakili oleh sebuah kubus dan garis yang menghubungkan antara kubus tersebut menunjukkan hubungan antara kedua *node* tersebut. Berikut ini adalah simbol-simbol yang digunakan pada *deployment diagram* (lihat Tabel II.8):

Tabel II.6. Simbol-Simbol *Deployment Diagram*

No.	Elemen	Simbol
1		<i>Package</i>

Tabel II.6. Simbol-Simbol *Deployment Diagram* (Lanjutan)





No.	Elemen	Simbol
2		<i>Node</i>
3		Kebergantungan / <i>dependency</i>
4		<i>Link</i>

(Sumber: Rosa dan Shalahuddin, 2011)

2.10 ERD (*Entity Relationship Diagram*)

Terdapat sebuah model perancangan hubungan antar entitas (tabel) dari sebuah basis data (*database*). Istilah untuk frase ini biasa dikenal dengan nama *Entity Relationship Model*. Model hubungan ini seterusnya akan berlanjut menjadi sebuah Diagram Hubungan Antar Entitas yang biasa dikenal dengan nama *Entity Relationship Diagram* (ERD) (Jeffrey A. Hoffer, 2007). Simbol-simbol ERD dapat dilihat pada Tabel II.12 berikut :

Tabel II.7 Simbol-Simbol ERD

Simbol	Nama	Deskripsi
	<i>Strong Entity</i>	Setiap hal dunia nyata (orang, tempat, objek, konsep, aktivitas) tentang suatu perusahaan mencatat data
	<i>Attribute</i>	Properti atau karakteristik tipe entitas.
	<i>Multivalued Attribute</i>	Karakteristik tipe entitas yang dapat memiliki nilai lebih dari satu.
	<i>Relationship</i>	Hubungan antar entitas.

(Sumber: Jeffrey A. Hoffer, 2007)

2.11 Kamus Data

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

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

Spesifikasi Tabel pemasok

Nama Tabel : Pemasok

Tipe : File master

Tabel II.8 Contoh Kamus Data Untuk Tabel Pemasok

No.	Nama Elemen	Akronim	Tipe	Width	Keterangan
1.	ID pemasok	ID_pemasok	Varchar	10	<i>Primary Key</i>
2.	Nama pemasok	Nama_pemasok	Char	40	
3.	Alamat pemasok	Alamat	Varchar	100	
4.	Nomor telepon	Telepon	Varchar	12	

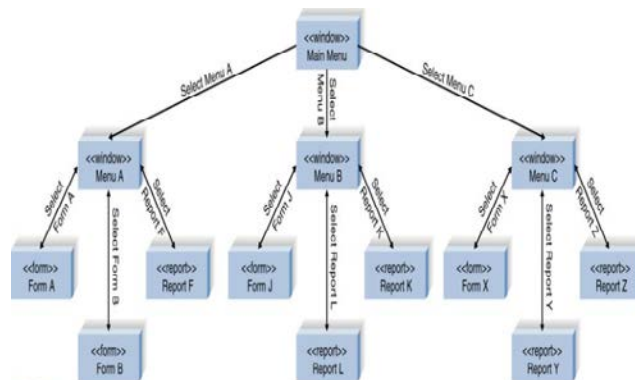
Sumber: Jogiyanto (2005)

2.12 *Windows Navigation Diagram (WND)*

Windows Navigation Diagram (WND) Menurut Dennis (2015):

- ❖ Menunjukkan bagaimana semua layar, formulir dan laporan terkait.
- ❖ Menunjukkan bagaimana pengguna bergerak dari satu ke yang lain.
- ❖ Seperti diagram keadaan untuk *user interface*
 - Kotak mewakili komponen
 - Panah mewakili transisi
 - Stereotip menunjukkan tipe antarmuka

Contoh Diagram WND menurut Dennis dapat dilihat seperti gambar dibawah ini:



Gambar II.7 Contoh *Windows Navigation Diagram*

(Sumber: Dennis, 2015)

2.13 XAMPP Local Server

XAMPP adalah perangkat lunak yang berfungsi sebagai *server* yang dapat berdiri sendiri (*localhost*) yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. XAMPP merupakan *web server* yang mudah digunakan dan dapat melayani tampilan halaman *web* yang dinamis, perangkat lunak ini tersedia dalam versi *GNU (General Public License)* yaitu bebas didapatkan tanpa harus membayar.

Menurut Yogi Wicaksono (2008), “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolah data *MySQL* dikomputer local”. XAMPP berperan sebagai *server web* pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan internet.

Ada beberapa bagian XAMPP yang biasa digunakan pada umumnya yaitu sebagai berikut:

1. *htdocs*, adalah *folder* tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas *PHP*, *HTML* dan *script* lainnya.

2. *phpMyAdmin*, merupakan bagian untuk mengelola basis data MySQL yang ada di komputer.
3. *Control Panel*, yang berfungsi untuk mengelola layanan (*service*) XAMPP. Layanan (*service*) ini seperti menghentikan (*stop*) layanan, ataupun memulai (*start*) layanan.

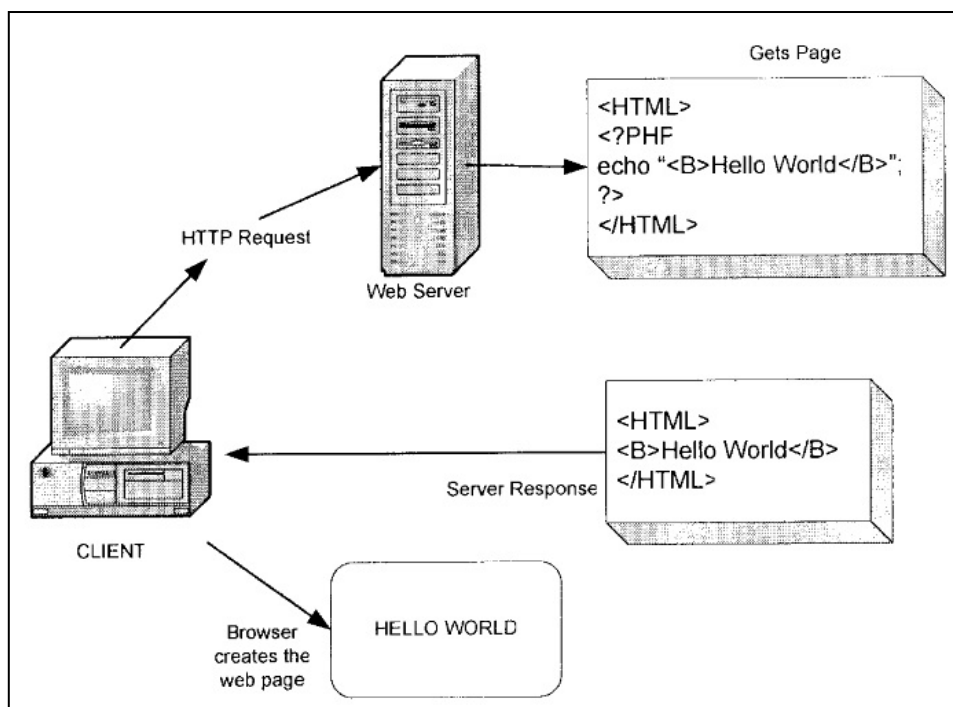
2.14 *Hypertext Preprocessor(PHP)*

PHP adalah singkatan dari *hypertext preprocessor* yaitu bahasa pemrograman yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. PHP dapat dikatakan sebagai sebuah *server-side embedded script language* yang artinya semua sintaks dan perintah program yang ditulis akan sepenuhnya dijalankan oleh *server* dan dapat juga diletakkan pada *script hyper text markup language* (HTML).

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*, maka dalam hal ini *client* menggunakan kode-kode PHP untuk mengirim permintaan ke *server* ketika menggunakan PHP sebagai *server-side embedded script language*, maka *server* akan melakukan beberapa hal sebagai berikut (Wahana Komputer, 2010):

1. Membaca permintaan dengan *script* PHP yang berasal dari *browser*.
2. Mencari halaman atau *page* di *server* (*server pages*).
3. Melakukan processing melalui instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman atau *page*.
4. Mengirim kembali halaman tersebut kepada *client* melalui *internet* atau *intranet* yang merupakan proses *echo* atau *print*.

Siklus PHP dalam *web server* dapat dijelaskan pada Gambar II.8.



Gambar II.8 Siklus PHP dalam *Web Server*
(Sumber: Wahana Komputer, 2010)

PHP adalah bahasa pemrograman web atau *scripting language* yang didesain untuk web. Awalnya PHP merupakan kependekan dari *Personal Home Page*. PHP dibuat pertama kali oleh Rasmus Lerdorf pada tahun 1994, yang pada awalnya dibuat untuk menghitung jumlah pengunjung pada *homepage*-nya. Selanjutnya diganti menjadi FI ("*Forms Interpreter*"). Sejak versi 3.0, namabahasa ini diubah menjadi "PHP: *Hypertext Preprocessor*" dengan singkatannya "PHP". PHP versi terbaru adalah versi ke -5 (Rosa dan Shalahuddin, 2011).

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai *apache*, *IIS*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.

4. Dalam sisi pemahaman, php adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.

Beberapa kelemahan PHP dari bahasa pemrograman lain, antara lain:

1. Tidak detail untuk pengembangan skala besar
2. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya (sampai versi 4).
3. Tidak bisa memisahkan antara tampilan dengan *logic* dengan baik.
4. PHP memiliki kelemahan *security* tertentu apabila *programmer* tidak jeli dalam memperhatikan isi konfigurasi PHP.

2.15 *My Structure Query Language (MySQL)*

MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan *multi user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public Licence* (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL (Rosa dan Shalahuddin, 2011).

Fitur-fitur MySQL antara lain (Rosa dan Shalahuddin, 2011):

1. Relational Database System
Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS (*Relational DataBase Management System*).
2. Arsitektur Client-Server
MySQL memiliki arsitektur *client-server* dimana *server database* MySQL terinstal di-*server*. *Client* MySQL dapat berada di komputer yang sama

dengan *server* dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet.

3. Mengetahui perintah SQL standar
SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.
4. Mendukung *Sub Select*
Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.
5. Mendukung *Views*
MySQL mendukung *views* sejak versi 5.0
6. Mendukung *Stored Prosedured (SP)*
MySQL mendukung *SP* sejak versi 5.0
7. Mendukung *Triggers*
MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
8. Mendukung *replication*.
9. Mendukung transaksi.
10. Mendukung *foreign key*.
11. Tersedia fungsi GIS.
12. *Free* (bebas diunduh)
13. Stabil dan tangguh
14. Fleksibel dengan berbagai pemrograman
15. *Security* yang baik
16. Dukungan dari banyak komunitas
17. Perkembangan software yang cukup cepat.
Tipe-tipe data yang ada di dalam My SQL antara lain :

1. Tipe Data *Numeric*

Tipe data *numeric* berisi sekumpulan tipe data sejenis yang mampu menangani data-data *numeric*. Berikut ini beberapa tipe data yang digunakan dalam MySQL:

Tabel II.9 Daftar Tipe Data *Numeric*

<i>Type</i>	Keterangan
TINYINT	Tipe ini merupakan bentuk <i>numeric</i> yang paling kecil dalam menangani data mulai dari angka -128 sampai dengan 127 (sampai dengan 4 digit).
SMALLINT	Memiliki kemampuan menyimpan data lebih besar dari TINYINT, yaitu mampu untuk menangani data mulai dari -32768 sampai dengan 32767 (sampai dengan 5 digit).
MEDIUMINT	Mampu menangani data mulai dari -83887608 sampai dengan 83887607 (sampai dengan 8 digit) .
INT	Tipe INT merupakan tipe yang sangat sering digunakan dalam pembuatan database, karena tipe ini dirasakan sangat cukup menampung dalam menangani data, yaitu mampu menyimpan data mulai dari -2147483648 sampai dengan 2147483647 (sampai dengan 11 digit).
BIGINT	Bentuk terbesar dalam menangani data <i>numeric</i> , mampu menangani data mulai dari -9223372036854775808 sampai dengan 9223372036854775807.
FLOAT (M, D)	Sebuah angka <i>floating-point</i> yang tidak dapat <i>unsigned</i> . Dapat menentukan panjang tampilan (M) dan jumlah desimal (D). Desimal presisi bisa pergi ke 24 tempat untuk sebuah <i>Float</i> .

Tabel II.9 Daftar Tipe Data *Numeric* (Lanjutan)

<i>Type</i>	Keterangan
DOUBLE (M, D)	Sebuah presisi angka <i>floating-point</i> ganda yang tidak dapat <i>unsigned</i> . Dapat menentukan panjang tampilan (M) dan jumlah desimal (D). Desimal presisi bisa pergi ke 53 tempat untuk sebuah DOUBLE. REAL adalah sinonim untuk DOUBLE.

(Sumber: Rosa dan Shalahuddin, 2011)

2. Tipe Data untuk Penanggalan dan Waktu

Untuk menangani data-data yang berkaitan dengan waktu dan penanggalan. Berikut tipe data untuk penanggalan dan waktu:

Tabel II.10 Daftar Tipe Data untuk Penanggalan dan Waktu

<i>Type</i>	Keterangan
DATE TIME	Bentuk ini dapat menyimpan dua buah bentuk tipe data sekaligus, yaitu penanggalan dan waktu. Bentuk yang dapat diciptakan oleh DATE TIME adalah '0000-00-00' merupakan bentuk penanggalan yang dimulai dari tahun, bulan dan tanggal. Sedangkan '00:00:00' adalah tempat menyimpan waktu atau jam.
DATE	Bentuk ini digunakan untuk menyimpan data penanggalan saja, dengan bentuk penulisan '0000-00-00'.
TIME STAMP	Bentuk penanggalan dengan TIME STAMP ditampilkan berjajar tanpa ada pembatasnya, dengan bentuk penulisan '00000000'.

Tabel II.10 Daftar Tipe Data untuk Penanggalan dan Waktu (Lanjutan)

TIME	Bentuk TIME hanya digunakan untuk menyimpan data berbentuk jam. Yaitu dimulai dari tahun yang dibaca dari dua karakter terakhir dan selanjutnya diikuti bulan dan tanggal, bentuk penulisannya ada '00:00:00'.
YEAR	Bentuk yang paling sederhana adalah YEAR, yang hanya menyimpan data berupa tahun saja. Ditulis secara lengkap 4 digit.

(Sumber: Rosa dan Shalahuddin, 2011)

3. Tipe Data *String*

Dengan menyesuaikan banyaknya data, MySQL telah membagi datanya menjadi beberapa tipe, sehingga penggunaannya dapat disesuaikan.

Berikut tipe data *string*:

Tabel II.11 Tipe Data *String*

<i>Type</i>	<i>Byte</i>	Keterangan
TINYTEXT	255	Tipe ini merupakan bentuk terkecil dari datat <i>string</i> yang mampu menangani data sampai dengan 2^8-1 data.
TINYBLOB	255	Bentuk <i>tinytext</i> adalah bentuk yang sama dengan <i>tinyblob</i> , yaitu mampu menangani data sampai dengan 2^8-1
TEXT	65535	Bentuk <i>text</i> salah satu bentuk tipe <i>string</i> yang mampu menanganini data sampai dengan berukuran $2^{16}-1$ data.
BLOB	65535	Memiliki kemampuan sama dengan <i>text</i> , yaitu sampai dengan $2^{16}-1$ data.
MEDIUM TEXT	16777215	Dapat menyimpan data dengan ukuran cukup besar, sampai dengan $2^{24}-1$ data.

Tabel II.11 Tipe Data *String* (Lanjutan)

<i>Type</i>	<i>Byte</i>	Keterangan
MEDIUM BLOB	16777215	Bentuk ini mampu menyimpan data sampai dengan $2^{24}-1$ data.
LOB	4294967295	Tipe data <i>longblob</i> adalah bentuk tipe data yang paling besar dalam menangani data. Data yang disimpan sampai dengan berukuran <i>GigaByte</i> . Tipe ini memiliki batasan penyimpanan sampai dengan $2^{32}-1$ data.

(Sumber: Rosa dan Shalahuddin, 2011)

4. Tipe Data *String* yang Sering Digunakan

Berikut macam-macam tipe data *string* yang sering digunakan:

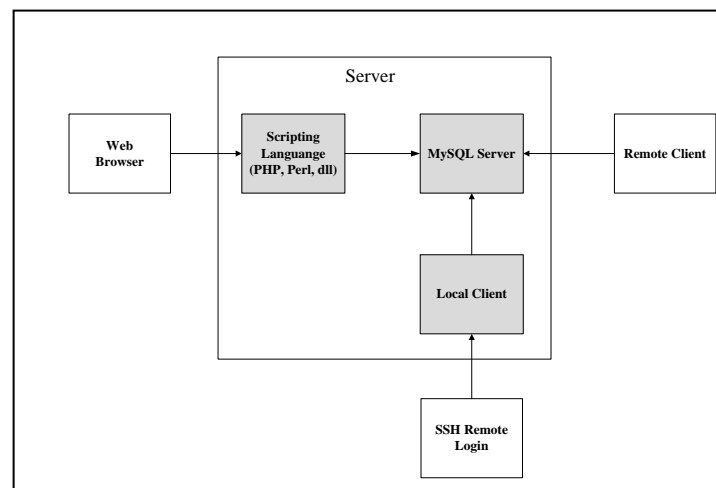
Tabel II.12 Tipe Data *String* yang Sering Digunakan

<i>Type</i>	Keterangan
<i>VARCHAR</i>	Bentuk ini dapat menyimpan data sampai dengan 225 karakter. Tipe ini dapat digunakan apabila data yang dimasukkan tidak lebih dari batasan tersebut.
<i>CHAR</i>	Bentuk <i>char</i> hampir sama dengan <i>varchar</i> , mampu menangani data sampai dengan 225 karakter. Namun, kedua tipe tersebut sangat signifikan dalam menyimpan data.
<i>ENUM</i>	Tipe data ini digunakan untuk validasi. Tipe data seperti ini, biasanya kolom ditentukan terlebih dahulu.
<i>SET</i>	Tipe data <i>set</i> sebenarnya memiliki fungsi yang sama dengan tipe <i>enum</i> , yaitu dengan mendeklarasikan anggota dari isi kolom yang mungkin akan menjadi anggotanya.

(Sumber: Rosa dan Shalahuddin, 2011)

Keunggulan MySQL adalah kemampuannya dalam menyediakan berbagai fasilitas atau fitur-fitur yang dapat digunakan oleh bermacam-macam *user* seperti *administrator database*, *programmer aplikasi*, *manager*, sampai dengan *end user* (pemakai akhir).

Pada mulanya MySQL bekerja pada platform *unix* dan *linux*, namun seiring perkembangannya sekarang banyak bermunculan beberapa versi yang mampu berjalan pada beberapa *platform* yang bersifat *shareware* dan *corporate*. Skema penggunaan MySQL dapat dilihat pada Gambar II.9.



Gambar II.9 Skema *Prototype Server MySQL*

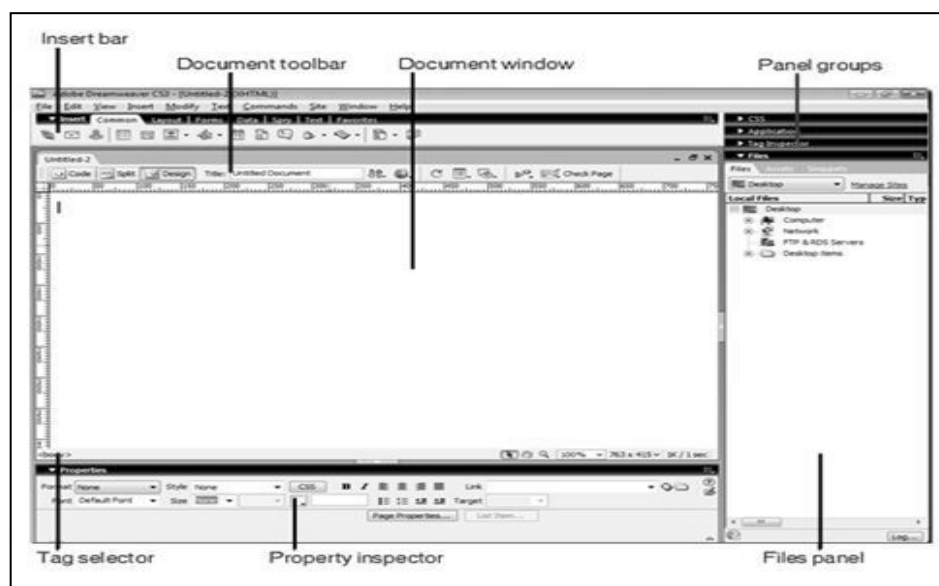
(Sumber: Wahana Komputer, 2010)

MySQL mempunyai prosedur keamanan non-standar yang salah satunya adalah *privillage system* (pengaturan hak akses pengguna), fungsi utama dari sistem ini ialah melakukan pengecekan *user* yang mengakses dari sebuah *host* dan memberikan beberapa hak seperti *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*. MySQL juga mendukung penggunaan koneksi enkripsi yang aman antara *server* dan *client* MySQL menggunakan *secure sockets layer (SSL) protocol*. Konfigurasi standar dari MySQL menerapkan kecepatan yang paling tinggi sehingga sambungan enkripsi tidak digunakan secara *default*. Menggunakan koneksi enkripsi akan menentukan porsi CPU yang besar yang membutuhkan beberapa pekerjaan tambahan sehingga memperlambat kerja MySQL.

MySQL mengizinkan penggunaan enkripsi secara personal, koneksi biasa, atau enkripsi bisa dipilih tergantung dari kebutuhan aplikasi itu sendiri. SSL adalah sebuah protokol yang menggunakan algoritma enkripsi untuk memastikan data yang ditranser melalui *public network* aman. SSL memiliki mekanisme untuk mendeteksi perubahan, kehilangan, atau pengiriman ulang data sehingga SSL memiliki algoritma yang bisa melakukan verifikasi identitas.

2.16 Adobe Dreamweaver

Adobe dreamweaver merupakan program penyunting halaman *web* dari *adobe systems* yang dulu dikenal sebagai *Macromedia Dreamweaver* dari *Macromedia*. *Adobe dreamweaver* dilengkapi dengan kemampuan manajemen situs yang memudahkan untuk mengelola keseluruhan elemen yang ada dalam sebuah situs. Pengguna juga dapat melakukan evaluasi situs dengan melakukan pengecekan *broken link*, kompatibilitas *browser*, termasuk validasi tag-tag HTML (*Hyper Text Markup Language*) dan CSS (*Cascading Style Sheet*) yang tidak sesuai dengan pedoman secara otomatis serta perkiraan waktu *download* pada sebuah halaman *web*. Tampilan antarmuka *adobe dreamweaver* dapat dilihat pada Gambar II.10.



Gambar II.10 Ruang Kerja *Adobe Dreamweaver*
(Sumber: Wahana Komputer, 2010)

Keterangan:

1. *Document Window* berfungsi untuk menampilkan dokumen yang sedang dikerjakan.
2. *Insert Bar* mengandung tombol-tombol untuk menyisipkan berbagai macam objek seperti *image*, *table* dan *layer* ke dalam dokumen.
3. *Document Toolbar* berisikan tombol-tombol dan menu *pop-up* yang menyediakan tampilan berbeda dari *Document Window*.
4. *Panel Groups* merupakan kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan dibawah satu judul.
5. *Tag Selector* berfungsi menampilkan hirarki tag disekitar pilihan yang aktif pada *Design View*.
6. *Property Inspector* digunakan untuk melihat dan mengubah berbagai *property objek* atau teks.
7. *Files Panel* digunakan untuk mengatur file dan folder-folder yang membentuk situs.

Adobe Dreamweaver CS6 secara default menyediakan beberapa *shortcut* untuk menjalankan perintah yang ada. *Adobe Dreamweaver CS6* terhubung dengan *browser Firefox* dan *browser Internet Explorer* agar bisa menampilkan *preview* desain melalui salah satu *browser* tersebut. *Adobe Dreamweaver* merupakan software web desain yang menawarkan cara mendesain *website* dengan cara sekaligus dalam satu waktu yaitu mendesain dan memprogram. *Adobe Dreamweaver* memiliki dua area kerja, berupa kode-kode HTML tertulis. *Adobe Dreamweaver* juga mempunyai kelebihan yaitu mampu mengenali tag-tag lain diluar HTML seperti ASP, PHP, serta mendukung *script-script dinamis* HTML dan *CSS Style*.

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian yang penulis gunakan dimulai dari mengidentifikasi masalah sampai dengan kesimpulan yang membentuk sebuah alur yang sistematis. Metodologi penelitian ini juga digunakan sebagai pedoman dalam pelaksanaan penelitian Tugas Akhir agar hasil yang dicapai tidak menyimpang dari tujuan yang telah ditentukan sebelumnya.

3.1.1 Jenis dan Sumber Data

Data yang dibutuhkan dalam penelitian ini dikelompokkan menjadi dua jenis, yaitu sumber data primer dan data sekunder.

1. Data primer

Data primer adalah data yang diperoleh secara langsung dari perusahaan dimana pengumpulan data atau informasi dilakukan langsung dari objek yang diteliti. Data-data tersebut adalah data yang digunakan dalam proses berlangsungnya persediaan bahan baku diantaranya struktur organisasi, sistem yang telah berjalan, dan kebutuhan pengguna sistem.

2. Data Sekunder

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

3.1.2 Metode Pengumpulan Data

Pengumpulan data merupakan sebuah sarana yang menampung dan mengumpulkan data-data dari berbagai sumber dan bentuk yang akan digunakan

dalam keperluan perancangan sistem usulan. Sumber data yang diperoleh untuk penyelesaian dalam penelitian ini, adalah sebagai berikut:

1. Studi lapangan

Pada metode pengumpulan data diperlukan turun langsung ke lapangan untuk memastikan benar adanya suatu sistem yang sedang berjalan, pengumpulan data di lapangan terbagi menjadi dua teknik yaitu:

1) Observasi

Merupakan pengamatan langsung terhadap kegiatan yang sedang berlangsung di PT KramaYudha Ratu Motor khususnya pada divisi produksi dan hrd, hasil observasi yang dilakukan adalah sebagai berikut:

1) Profil mengenai PT KramaYudha Ratu Motor.

2) Mengamati sistem pemrosesan *line damage card* yang sedang berjalan.

2) Wawancara

Merupakan pengumpulan data melalui tanya jawab secara langsung kepada pihak-pihak yang terkait dengan objek penelitian. Hal pertama yang dilakukan dalam wawancara ialah membuat daftar pertanyaan yang berkaitan dengan penelitian yang akan dibahas, pihak yang diwawancarai merupakan orang yang terlibat didalam kegiatan pemrosesan *form line damage card* seperti Kepala Bagian Gudang, *Staff* Produksi, dan *Staff Part Control*.

2. Studi kepustakaan

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

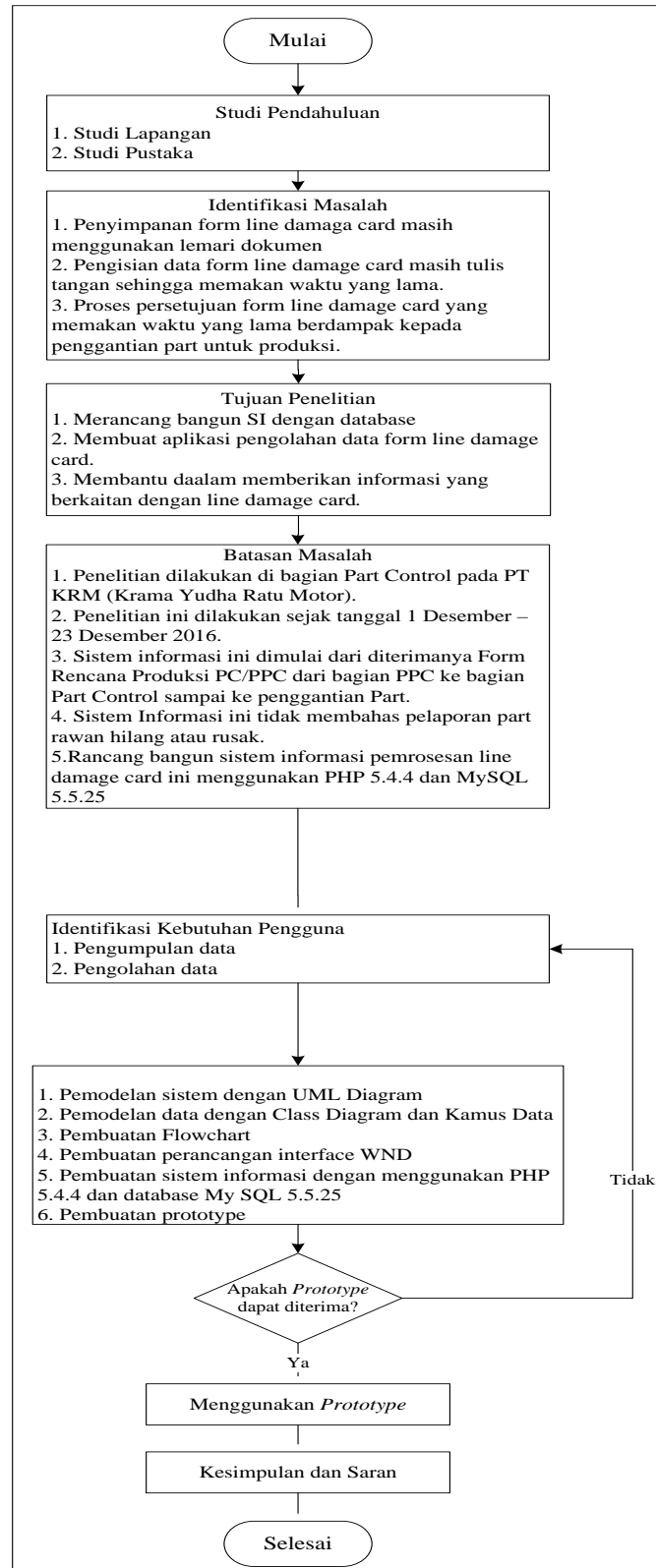
3.2 Metode Pengembangan Sistem

Prototype adalah versi sistem informasi atau bagian dari sistem yang sudah dapat berfungsi, tetapi dimaksudkan hanya sebagai model awal saja. Setelah beroperasi, *prototype* akan dikembangkan kembali hingga *prototype* tersebut menjadi cocok dengan kebutuhan penggunanya. Ketika rancangannya telah mencapai tahap final, *prototype* dapat dikonversi menjadi sistem yang lebih baik.

Pembuatan *prototype* meliputi pengembangan sistem uji coba yang cepat dan murah untuk dapat dievaluasi oleh pengguna akhir (*end user*). Lewat interaksi dengan *prototype*, para pengguna dapat memperoleh gagasan yang lebih baik mengenai kebutuhan informasi mereka. *Prototype* yang telah disetujui oleh pengguna dapat digunakan sebagai patokan untuk membuat sistem versi finalnya. *Prototype* terdapat dua jenis yaitu evolusioner dan persyaratan. Untuk penelitian Tugas Akhir ini menggunakan metode pengembangan sistem *evolutionary prototype*. *Prototype* terus menerus disempurnakan sampai memiliki seluruh fungsionalitas yang dibutuhkan pengguna dari sistem yang baru. Jadi, *evolutionary prototype* akan menjadi sistem aktual yang dapat dikembangkan lebih lanjut. Dalam perancangan *prototype* pemodelan sistem yang digunakan yaitu *Unified Modelling Language* (UML).

3.3 Kerangka Penelitian

Dari permasalahan yang telah diidentifikasi melalui penelitian, maka dibuat sebuah kerangka yang menjelaskan tahap-tahap dalam memecahkan permasalahan yang terdapat pada sistem informasi pemrosesan *line damage card* pada PT KramaYudha Ratu Motor. Kerangka penelitian dibuat dalam bentuk *flowchart* yang menggambarkan tahap-tahap kegiatan mulai dari awal hingga akhir dapat dilihat pada Gambar III.1.



Gambar III.1 Kerangka Penelitian
(Sumber: Pengolahan Data, 2018)

Penjelasan langkah-langkah atau tahapan dalam penelitian tersebut adalah sebagai berikut:

1. Studi Pendahuluan

Studi pendahuluan yang dilakukan terbagi menjadi dua, antara lain:

1) Studi Lapangan

Studi Lapangan yang dilakukan berupa observasi langsung dan wawancara dengan pihak-pihak terkait yang berada di PT Krama Yudha Ratu Motor

2) Studi Pustaka

Studi Literatur yang dilakukan yakni membaca buku dan literatur serta sumber-sumber lain dalam lingkup perkuliahan maupun di luar lingkup perkuliahan yang berhubungan dengan judul dan permasalahan yang ada. Studi kepustakaan yang penulis lakukan adalah dengan mengunjungi perpustakaan dan mencari data yang diperlukan melalui internet.

2. Pokok Permasalahan

Tahap ini merupakan tahap perumusan-perumusan masalah yang terjadi pada sistem yang sedang berjalan.

3. Tujuan Penelitian

Berdasarkan pokok permasalahan yang telah ditetapkan maka tujuan penelitian dari Tugas Akhir ini yaitu:

1. Merancang bangun SI dengan *database*.
2. Membuat aplikasi pemrosesan *line damage card*.
3. Membantu dalam memberikan informasi yang berhubungan dengan *line damage card*.

4. Batasan Masalah

Penentuan batasan masalah dilakukan agar pembahasan tetap fokus dan tidak melenceng dari topik yang dibahas.

5. Pengembangan sistem menggunakan *Evolutionary Prototype*

Penggunaan *evolutionary prototype* dimulai dengan empat tahapan yang diantaranya adalah sebagai berikut:

1) Identifikasi kebutuhan pengguna

Pada tahap ini dimulai dari analisis kebutuhan pengguna untuk mengetahui sistem yang akan dibangun.

2) Pembuatan *prototype*

Membuat *prototype* terdiri dari enam macam antara lain:

1. Pemodelan sistem dengan UML.
2. Pemodelan data dengan *class diagram* dan kamus data.
3. *Prototype Flowchart*.
4. *Prototype* perancangan *interface* WND.
5. Pembuatan *prototype* dengan menggunakan PHP 5.4.4 dan *database* MySQL 5.5.25.

3) Menentukan apakah *prototype* dapat diterima

Pada tahap ini pula akan dilihat apakah *prototype* yang dibuat dapat diterima oleh pengguna atau tidak. Jika tidak, akan dilakukan kembali mengidentifikasi kebutuhan pengguna, tetapi jika ya, tahap selanjutnya adalah menggunakan *prototype* tersebut.

4) Menggunakan *prototype*

Pada tahap ini merupakan hasil dari keputusan yang diambil berdasarkan tahap evaluasi yang berarti bahwa sistem yang telah dirancang sudah sesuai dengan kebutuhan penggunaan akan digunakan.

6. Kesimpulan dan Saran

Mengambil kesimpulan dari perbandingan hasil sistem yang berjalan dengan sistem yang diusulkan, serta dapat memberikan saran kepada PT Krama Yudha Ratu Motor.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Hubungan Kerja Sama PT Krama Yudha Ratu Motor

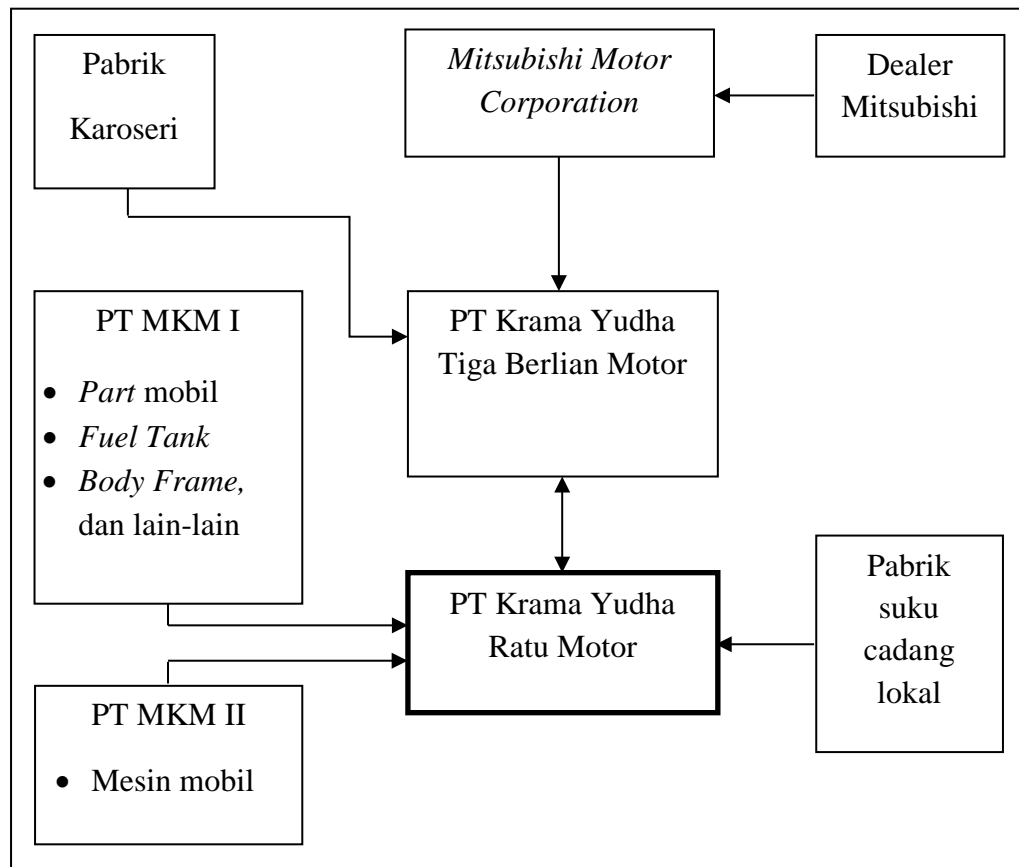
PT Krama Yudha Tiga Berlian yang tergabung dengan *Mitsubishi Motor Corporation (MMC)* merupakan perusahaan induk (*Holding Company*) PT Krama Yudha Ratu Motor.

Dalam proses produksinya, PT Krama Yudha Ratu Motor terikat kontrak kerja dengan PT Krama Yudha Tiga Berlian. PT KTB mengimpor komponen-komponen *Complete Knock Down (CKD)* dari *MMC*. Komponen-komponen yang telah di terima oleh PT KTB kemudian diserahkan kepada PT KRM untuk dirakit. Selain menerima *CKD* dari *MMC* melalui PT KTB, PT 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 Manufaktur I dan rangka mobil disediakan PT Mitsubishi Krama Yudha Motor dan Manufaktur II (MKM I dan MKM II).

Dalam proses produksinya, PT 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 KRM kepada *supplier* yang bersangkutan.

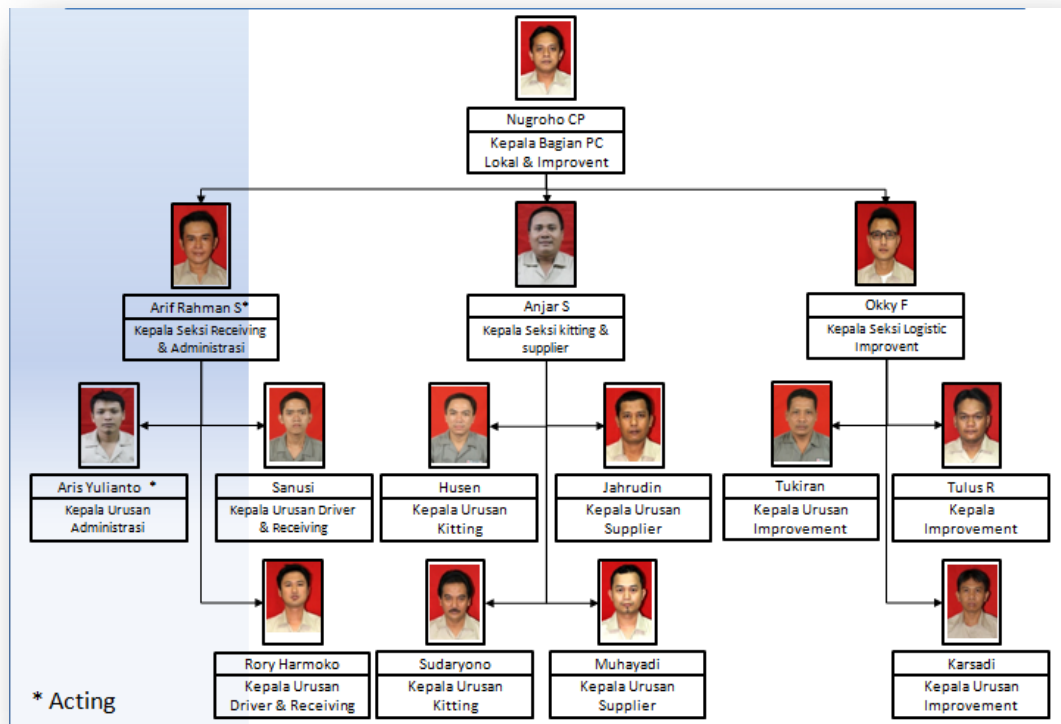
Setelah proses perakitan dan pengujian selesai dilakukan, kendaraan diserahkan kepada PT KTB untuk didistribusikan kepada para *dealer* Mitsubishi. Adapun hubungan kerjasama dapat dijabarkan pada Gambar IV.1 berikut ini.



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, 2016)

4.2 *Part Control Local Organisation*

Karena yang penulis teliti adalah bagian *part control* maka kita perlu mengetahui struktur organisasi dalam bagian *part control* tersebut. masing-masing bagian memegang peranan yang sangat penting dalam mendukung pencapaian tujuan suatu perusahaan. Karena merupakan cetak biru perusahaan, didalamnya menunjukkan dan menjelaskan pembagian tugas, tanggung jawab maupun wewenang setiap bagian yang ada dalam bagian *part control* perusahaan, serta sekaligus sebagai alat koordinasi antar bagian, baik secara vertikal maupun secara horizontal. Berikut gambar Struktur organisasi dalam part control dapat dilihat pada Gambar IV.2.



Gambar IV.2 Struktur Organisasi *Part Control Lokal*

(Sumber: PT Krama Yudha Ratu Motor, 2016)

1. KEPALA SEKSI *KITTING* DAN *SUPPLIER*

Bertanggung Jawab terhadap

- Proses *supply part* dari warehouse ke main line
- Proses setting *box kitting* dan *supply box kitting* ke main line
- Pengelolaan sumber daya manusia di supplier dan *kitting*
- Proses *line damage* akibat *handling operator supplier & kitting*
- Kerapihan area gudang
- Fasilitator 5r area gudang
- Kesesuaian penggunaan alat angkut dan angkat
- Menjaga keselamatan dan kesehatan kerja karyawan PC Lokal

Wewenang

- Rotasi di bagian supplier dan kitting
- *Reward* dan *punishment* operator supplier dan *kitting*
- Penilaian operator supplier dan *kitting*

2. KEPALA URUSAN SUPPLIER

Bertanggung Jawab terhadap

- Pengawasan langsung proses *Supply part* ke *line* produksi
- Kontrol kesesuaian pekerjaan terhadap SOP
- Kontrol kesesuaian *stock part* di *line* produksi
- Kontrol kesesuaian tipe *part* di *line* produksi
- Kontrol ketepatan waktu *supply part* di *line* produksi
- Proses *line damage* yang diakibatkan *handling* operator supplier
- Investigasi part *line damage* yang ada di area produksi
- Kerapihan box kosong di area produksi dan area box kosong
- Koordinator 5r di bagian supplier
- Menjaga kondisi alat angkut

Wewenang

- Rotasi operator di bagian supplier
- *Reward* dan *punishment* operator supplier
- Penilaian operator supplier
- Pengaturan *job* untuk *back up* absensi

3. KEPALA URUSAN *KITTING*

Bertanggung Jawab terhadap

- Pengawasan langsung proses *setting box Kitting*
- Kontrol kesesuaian pekerjaan terhadap SOP
- Kontrol kesesuaian jumlah part di box *kitting*
- Kontrol kesesuaian type part di box *kitting*

- Kontrol ketepatan waktu supply box *kitting* ke line produksi
- Proses *line damage* operator *kitting*
- Investigasi *part line damage* di box *kitting*(kembali dari produksi)
- Kerapihan *part* di area setting *kitting*
- Kerapihan box kosong di area setting *kitting*
- Koordinator 5r di bagian setting *kitting*
- Menjaga kondisi alat angkut , box *kitting* , *dolly kitting*

Wewenang

- Rotasi operator di bagian setting *kitting*
- *Reward* dan *punishment* operator setting *kitting*
- Penilaian operator setting *kitting*
- Pengaturan *job* untuk *back up* absensi

4. KEPALA SEKSI *RECEIVING* DAN *ADMINISTRATION*

Bertanggung Jawab terhadap

- Proses kesesuaian *receiving part* dari vendor.
- Proses *supply part* yang menggunakan *forklift*.
- Kontrol *stock part* dan ketersediaan part.
- Pusat administrasi.
- Proses *claim part*.
- Proses *line damage* akibat *handling* operator *checker* dan *driver*
- Kerapihan area gudang.
- Fasilitator 5r area gudang.
- Kesesuaian penggunaan alat angkat & angkut.

Wewenang

- Rotasi di bagian *Checker, driver & Administrasi*
- *Reward* dan *punishment* operator *Checker, driver & Administrasi*
- Penilaian operator *Checker, driver & Administrasi*

5. KEPALA URUSAN *CHECKER DRIVER*

Bertanggung Jawab terhadap

- Pengawasan langsung proses *receiving part* dari vendor
- Kontrol kesesuaian SOP proses *receiving part*
- Kontrol *supply part* yang menggunakan *forklift*
- Kontrol *loading unloading* vendor
- Kontrol *receiving* dan penempatan *part milkrun*
- Kerapihan *part* di *warehouse*
- Kebersihan area *part* dari sampah di *warehouse*
- Kerapihan di area box kosong
- Koordinator 5r di bagian *Checker Driver*
- Menjaga kondisi alat angkut

Wewenang

- Rotasi operator di bagian *Checker Driver*
- *Reward* dan *punishment* operator *Checker Driver*
- Penilaian operator *Checker Driver*
- Pengaturan *job* untuk *back up* absensi

6. KEPALA URUSAN *ADMINISTRASI*

Bertanggung Jawab terhadap

- Pengawasan proses input surat jalan di admistrasi
- Kontrol *balance stock part* lokal
- Kontrol *claim part*
- Pengadaan *stock check* untuk *part* yang di anggap kurang
- Kontrol kerapihan dan kebersihan di area *office pc* lokal
- *Central* administrasi

- Koordinator 5r di bagian administrasi

Wewenang

- Rotasi operator di bagian admistrasi
- *Reward* dan *punishment* operator admistrasi
- Penilaian operator admistrasi
- Pengaturan *job* untuk *back up* absensi

4.3 Jenis-Jenis Produk

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

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

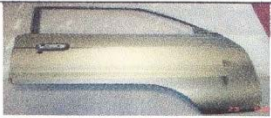


Model Kendaraan	Type	Spesifikasi	Berat Kotor Kendaraan (Ton)
CJM	CJ-M G-PU	78 PS/ 1343 cc	1,7
	CJ-M G-MB	78 PS/ 1343 cc	1,7
SL	SL D-PU	72 PS/ 2500 cc	2,5
	SL D-MB	72 PS/ 2500 cc	2,5
Truck Diesel (TD)	<i>Colt Diesel</i> FE-304	100 PS/ 3298 cc	5
	<i>Colt Diesel</i> FE-334	100 PS/ 3298 cc	7
	<i>Colt Diesel</i> FE-334 HD	100 PS/ 3298 cc	7
Fuso	<i>Fuso</i> FM-517H	190 PS/ 7545 cc	14
	<i>Fuso</i> FM-517	190 PS/ 7545 cc	21
	<i>Fuso</i> FN-527	220 PS/ 7545 cc	23,5
	<i>Fuso</i> FN-527 HD	220 PS/ 7545 cc	23,5

(Sumber: PT Krama Yudha Ratu Motor, 2016)

4.3.1 Contoh Jenis Part

Part yang terdapat pada masing-masing bagian *Part Control* berbeda sesuai dengan jenis kendaraan yang diproduksi. Contoh *part* dapat dilihat pada gambar berikut ini:

1. CJM

<i>PART CODE</i>	<i>SKETCH</i>	<i>PART NAME</i>
TB-018		Panel A Syy Front Door Rh
TB-021		Bracket Front Bumper Ctr
TB-023		Reinf Front Pilar Pedal

Gambar IV.3 Contoh *Part* Jenis Kendaraan CJM

(Sumber: PT Krama Yudha Ratu Motor, 2016)

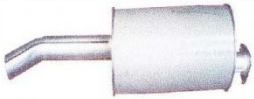

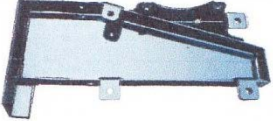
2. SL

<i>PART CODE</i>	<i>SKETCH</i>	<i>PART NAME</i>
D-003		Frame Assy MB
D-010		Post Side Lh
D-080		Bek. Bumper

Gambar IV.4 Contoh *Part* Jenis Kendaraan SL

(Sumber: PT Krama Yudha Ratu Motor, 2016)




3. *Truck Diesel*

<i>PART CODE</i>	<i>SKETCH</i>	<i>PART NAME</i>
TDP-002		MUFLER
TDP-008		PIPE AIR
TDP-016		BRACKET FUEL TANK REAR

Gambar IV.5 Contoh *Part* Jenis Kendaraan *Truck Diesel*

(Sumber: PT Krama Yudha Ratu Motor, 2016)

4. Fuso

<i>PART CODE</i>	<i>SKETCH</i>	<i>PART NAME</i>
U-02		Dash Board Assy
U-031		Panel Back inner Lh
U-036		Fender.Sd.Lower Assy Lh

Gambar IV.6 Contoh *Part* Jenis Kendaraan Fuso

(Sumber: PT Krama Yudha Ratu Motor, 2016)

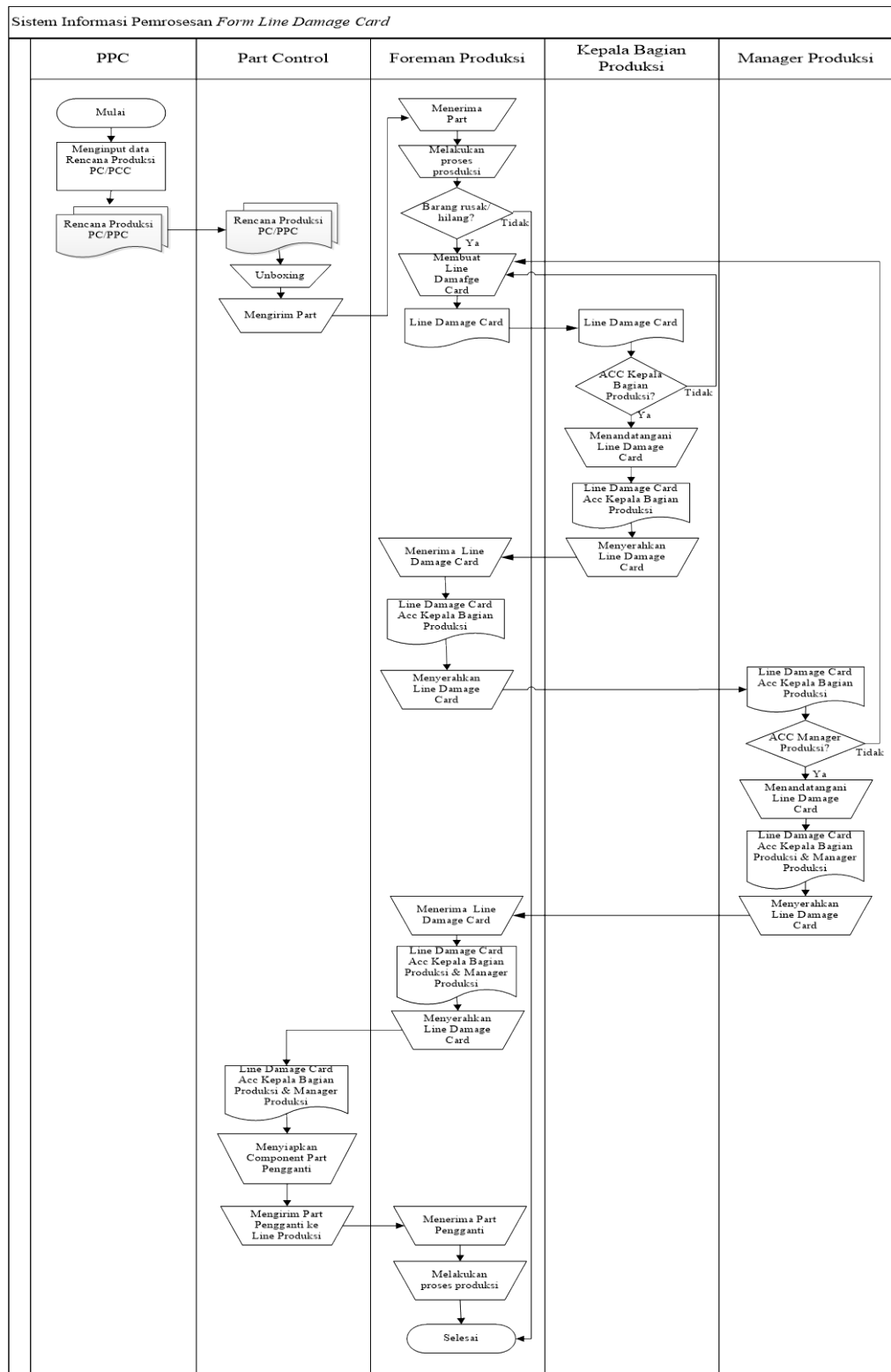
4.4 Sistem Informasi Pemrosesan *Form Line Damage Card* Yang Sedang Berjalan.

Pada PT Krama Yudha Ratu Motor proses yang digunakan untuk pemrosesan *line damage card* diuraikan sebagai berikut:

1. Bagian PPC mennginput data rencana produksi PC/PPC dan memberikannya ke bagian *Part Control*.

2. Bagian *Part Control Unboxing* sesuai dengan Form Rencana Produksi PC/PPC.
3. Bagian *Part Control* mengirimkan *Part* ke *Foreman* Produksi.
4. *Foreman* Produksi Menerima *part* yang diserahkan oleh bagian *Part Control*.
5. *Foreman* produksi melakukan proses produksi.
6. *Foreman* produksi melakukan pengecekan terhadap *part* apabila ada kerusakan atau hilang pada *part/part*.
7. Jika ditemukan *part* hilang atau rusak maka foreman produksi membuat *form Line Damage Card* dan menyerahkan *form* tersebut ke Kepala Bagian Produksi.
8. Kepala Bagian Produksi memvalidasi *form Line Damage Card* jika tidak sesuai maka *form* akan dikembalikan ke foreman produksi untuk dibuat ulang dan apabila sesuai kepala bagian produksi akan ACC *form Line Damage Card*.
9. Kepala bagian produksi menyerahkan *form Line Damage Card ACC* ke *foreman* produksi.
10. *Foreman* Produksi menerima *form Line Damage Card ACC* dari Kepala Bagian Produksi.
11. *Foreman* produksi menyerahkan *form Line Damage Card* ke *Manager* Produksi.
12. *Manager* Produksi melakukan validasi pada *form Line Damage Card* yang sudah di ACC oleh Kepala Bagian Produksi, apabila tidak sesuai maka *Manager* Produksi akan mengembalikan *form* kepada *Foreman* Produksi untuk dibuat yang baru, apabila sesuai manager melakukan ACC terhadap *form Line Damage Card*.
13. *Manager* Produksi menyerahkan *form Line Damage Card* yang sudah di ACC kepada *Foreman* Produksi.
14. *Foreman* Produksi menerima *form Line Damage Card* yang sudah di ACC oleh *Manager* Produksi.

15. *Foreman* Produksi menyerahkan *form Line Damage Card* yang sudah di ACC oleh Kepala Bagian Produksi dan *Manager* Produksi ke bagian *Part Control* untuk meminta pengganti *part* yang hilang atau rusak.
16. Bagian *Part Control* menyiapkan *part* pengganti sesuai dengan *form Line Damage Card*.
17. *Bagian Part Control* mengirimkan *part* pengganti ke *Foreman* Produksi.
18. *Foreman* Produksi menerima *part* pengganti yang diberikan oleh bagian *Part Control*.
19. *Foreman* Produksi melakukan proses produksi.



Gambar IV.7 Flowmap Pemrosesan Line Damage Card yang Sedang Berjalan

(Sumber: Pengolahan Data, 2018)

4.4.1 Dokumen yang Terkait dalam Pemrosesan *Line Damage Card*

Dokumen-dokumen yang digunakan dalam proses persediaan dan pengiriman *part* ke *line* produksi:

1. Rencana Produksi PC/PPC

Rencana Produksi PC/PPC merupakan acuan bagi bagian part control untuk menyiapkan dan mengirimkan *part* yang dibutuhkan untuk perakitan kendaraan sesuai dengan jumlah kendaraan yang akan diproduksi setiap hari. Pada *form* ini terdapat tanggal produksi, jenis kendaraan, tipe kendaraan, jumlah kendaraan, dan tipe perakitan.

Keterangan Gambar:

Kode Dokumen	: berguna untuk menunjukkan kode dokumen Rencana Produksi PC/PPC.
Tanggal Berlaku	: tanggal berlakunya dokumen Rencana Produksi PC/PPC.
Revisi	: menunjukkan jumlah revisi pada dokumen Rencana Produksi PC/PPC.
TANGGAL	: tanggal dilakukannya proses produksi <i>Trimming</i> .
TITIK HITUNG	:Menjelaskan tempat mulainya dilakukan proses produksi <i>Trimming</i> .
TYPE	: Menjelaskan tipe dari proses produksi yang dilakukan.
CJM EURO 2	: proses produksi kendaraan jenis <i>Colt</i> .
L300 EURO 2	: proses produksi kendaraan jenis <i>pick up</i> .
TD	: Proses produksi kendaraan jenis <i>Truck Diesel</i> .
FUSO M/N EURO 2	: proses produksi kendaraan jenis <i>Fuso</i> .
TOTAL	: jumlah kendaraan dari setiap jenis produksi yang dilakukan di bagian.
WELDING	: proses produksi pada bagian <i>welding</i> .
FRONT AXLE	: proses produksi <i>front axle</i> (mesin dan transmisi).
T.ON	: proses <i>trimming on</i> .
T.OFF	: proses <i>trimming off</i> .
DOCKING	: proses pemasangan <i>docking</i> pada kendaraan.
PRA CONVEYOR	: proses pengiriman kendaraan untuk di uji coba dengan alat pembawa kendaraan (<i>conveyor</i>).

2. Line Damage Card

Form Line Damage Card merupakan data *part* yang rusak atau hilang pada saat produksi. berupa nomor, nama, tipe, dan jenis kerusakan *part* serta cara pengulangan yang dilakukan sesuai dengan jenis kerusakan yang terjadi pada *part* tersebut.

PT. KRAMA YUDHA RATU MOTOR		NO. DOKUMEN : FRM-PC-715-019
LINE DAMAGE CARD		TANGGAL BERLAKU : 15-05-2003
		REVISI : 00
A : REPAIR B : PARTS. CHANGE C : SCRAP		TYPE : FUGO KRM. MMC. INS
DATE	16-6-2017	
PART NO.	MC 908743Y	
PART NAME	MIRROR, REAR VIE	
QUANTITY	1-PCS	
PENYEBAB : - MIRROR PECAH PADA SAAT DI BUKA UNTUK REPAIR BOLT STAY MIRROR YANG PATAH.		
PENANGGULANGAN : - PADA SAAT PEMASANGAN BOLT STAY BACK MIRROR DI LAKUKAN DENGAN CARA MAHUAL TERLEBIH DAHULU. SEBELUM DI KENCANGKAN DENGAN IMPACK.		
MANAGER	SUPERVISOR	FOREMAN
<i>[Signature]</i> Timmet	<i>[Signature]</i>	<i>[Signature]</i>
BAGIAN : TRIMING.		SEKSI : SUB ASSY.

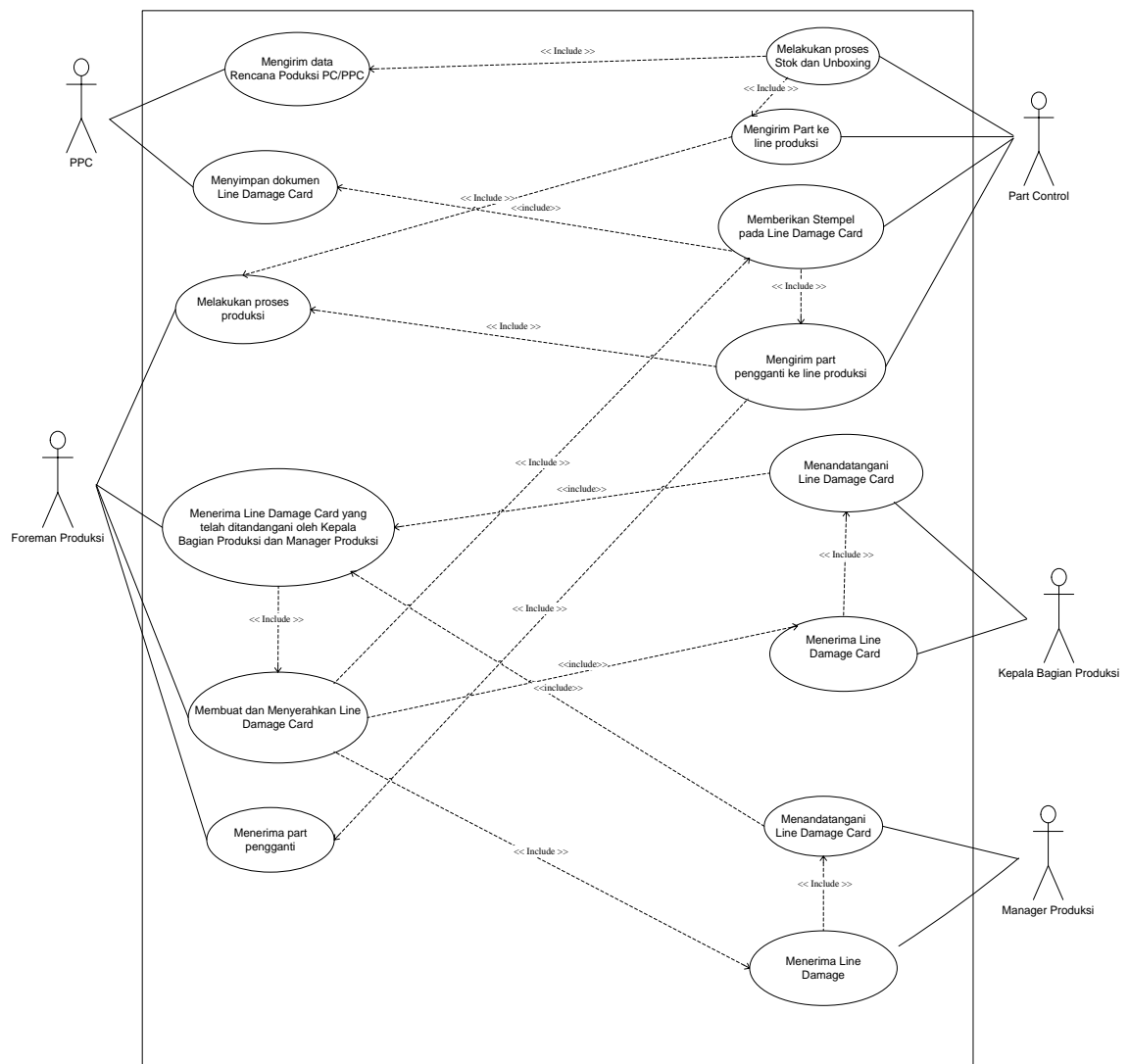
Gambar IV.9 Line Damage Card

(Sumber: PT Krama Yudha Ratu Motor, 2016)

4.4.2 Sistem Pemrosesan *Line Damage Card* dengan *Unified Modelling Language (UML)*

Penggambaran *Unified Modelling Language (UML)* Sistem Pemrosesan *Line Damage Card* di *line* produksi pada PT Krama Yudha Ratu Motor dengan menggunakan *Use Case Diagram* adalah sebagai berikut :

1. *Use Case Diagram*



Gambar IV.10 *Use Case Diagram* Pemrosesan *Line Damage Card*

(Sumber: Pengolahan Data, 2018)

Tabel IV.2 Definisi *Actor* dalam Pemrosesan *Line Damage Card*

<i>Actor</i>	Keterangan
PPC	Berperan dalam Mengirim data Rencana Produksi PC/PPC
<i>Part Control</i>	Berperan dalam melakukan proses <i>unboxing</i> , stok, mengirim <i>part</i> , dan stempel <i>form Line Damage Card</i> .
<i>Foreman</i> Produksi	Berperan dalam menerima <i>part</i> , melakukan proses produksi, membuat <i>form Line Damage Card</i> , menerima dan menyerahkan <i>Line Damage Card</i>
Kepala Bagian Produksi	Berperan dalam menandatangani <i>form Line Damage Card</i> , menyerahkan <i>form Line Damage Card</i> kepada <i>foreman</i> produksi.
Manager Produksi	Berperan dalam menandatangani <i>form Line Damage Card</i> , menyerahkan <i>form Line Damage Card</i>

(Sumber: Hasil Pengolahan Data, 2018)

Tabel IV.3 Definisi *Use Case* dalam Pemrosesan *Line Damage Card*

<i>Use Case</i>	Keterangan
Mengirim data Rencana Produksi PC/PPC	Merupakan kegiatan pengiriman dokumen Rencana Produksi PC/PPC sebagai acuan <i>Part Control</i> dalam pengiriman part. Kegiatan ini dilakukan oleh <i>Actor</i> PPC.
<i>Unboxing</i>	Merupakan kegiatan membongkar <i>part</i> dari gudang <i>Part Control</i> untuk dikelompokan sesuai jumlah dan jenis <i>part</i> sebelum dikirim ke <i>line</i> produksi. Kegiatan ini dilakukan oleh <i>Actor Part Control</i> .
Mengirim <i>part</i>	Merupakan kegiatan pengiriman <i>part</i> ke line produksi. Kegiatan ini dilakukan oleh <i>Actor Part Control</i> .
Menerima <i>part</i>	Merupakan kegiatan penerimaan <i>part</i> yang dikirim dari <i>Part Control</i> . Kegiatan ini dilakukan oleh <i>Actor Foreman</i> Produksi.

Tabel IV.3 Definisi *Use Case* dalam Pemrosesan *Line Damage Card* (Lanjutan)

<i>Use Case</i>	Keterangan
Membuat <i>Line Damage Card</i>	Merupakan kegiatan pembuatan <i>form</i> informasi temuan <i>part</i> yang rusak saat diproduksi berupa jenis dan cara penanggulangan kerusakan. Kegiatan ini dilakukan oleh <i>Actor Foreman</i> Produksi..
Menandatangani <i>Line Damage Card</i>	Merupakan kegiatan penyetujuan <i>form</i> <i>Line Damage Card</i> yang dibuat oleh foreman produksi. Kegiatan ini dilakukan oleh <i>Actor</i> Kepala Bagian Produksi dan <i>Manager</i> Produksi.
Stempel <i>Line Damage Card</i>	Merupakan kegiatan pemberian stempel pada <i>form</i> <i>Line Damage Card</i> sebagai persetujuan untuk mengeluarkan <i>part</i> baru sebagai pengganti <i>part</i> rusak. Kegiatan ini dilakukan oleh <i>Actor</i> Part Control.

(Sumber: Hasil Pengolahan Data, 2018)

BAB V

ANALISIS DAN PEMBAHASAN

PT. Krama Yudha Ratu Motor memiliki sistem informasi pemrosesan *line damage card* yang merupakan pengembangan dari sistem sebelumnya di mana sistem yang dirancang diharapkan akan memberikan kemudahan pada pihak perusahaan khususnya dalam pemrosesan *line damage card*.

Pada bab ini akan dijelaskan analisis rinci dari pengolahan sistem informasi pemrosesan *line damage card* pada PT Krama Yudha Ratu Motor. Analisis tersebut dimulai dari analisis kebutuhan sistem, perancangan UML, perancangan basis data, perancangan *interface*, perancangan WND dan pembuatan spesifikasi proses melalui metode yang diterapkan.

5.1 Analisis Kebutuhan Fungsional Sistem

Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan perancangan sistem informasi pemrosesan *line damage card* yang terintegrasi sebagai sarana untuk memberikan informasi mengenai *form line damage card* sebagai *form* pelaporan *part* yang rusak atau hilang saat proses produksi di PT Krama Yudha Ratu Motor. Perancangan aplikasi sistem informasi pemrosesan *line damage card* akan menampilkan data yang terhubung dengan informasi rencana produksi dan *line damage card*. Sistem informasi pemrosesan *line damage card* ini dibuat berbasis *web* sehingga bersifat *user friendly* dalam penggunaannya dan dapat mempermudah dan tepat guna bagi *user* dalam mengelola informasi.

Adapun kebutuhan fungsional sistem yang diidentifikasi untuk sistem informasi pemrosesan *line damage card* pada PT Krama Yudha Ratu Motor adalah sebagai berikut:

Tabel V.1 Kebutuhan Fungsional Sistem

Kebutuhan Sistem	
<i>Project Name</i>	Sistem Informasi Pemrosesan <i>Line Damage Card</i>
<i>Business Need</i>	<ol style="list-style-type: none"> 1. Terwujudnya sebuah sistem yang membantu proses pembuatan <i>line damage card</i>. 2. Terwujudnya sebuah sistem proses pembuatan form rencana produksi dan <i>line damage card</i>. 3. Terwujudnya sebuah sistem yang membantu mempermudah proses pelaporan <i>form line damage card</i>. 4. Terwujudnya sebuah sistem yang mempercepat proses penggantian <i>part</i> baru untuk diproduksi.
<i>Business Requirement</i>	<ol style="list-style-type: none"> 1. Memberikan sistem informasi pemrosesan <i>line damage card</i> yang dapat membantu proses kerja <i>Part Control</i>.
<i>Business Value</i>	<ol style="list-style-type: none"> 1. Mempermudah proses pengiriman <i>part</i>. 2. Mempermudah proses penerimaan <i>part</i>. 3. Mempercepat dalam pembuatan laporan rencana produksi. 4. Memudahkan dalam melihat data <i>part</i> yang dikirim. 5. Memudahkan proses yang berhubungan dengan <i>line damage card</i>.
<i>Special Issues or Constrains</i>	Pembuatan sistem dilakukan selama empat bulan.

(Sumber: Hasil Pengolahan Data 2018)

5.2 Perancangan Sistem

Tahap ini adalah tahap kedua pada metodologi *prototype evolutioner* yaitu membuat *prototype* berupa prototipe sistem setelah sebelumnya melakukan tahap mengidentifikasi kebutuhan. Perancangan proses sistem informasi pemrosesan *line damage card* menggunakan *modeling system* UML (*unified modeling language*) yang meliputi pembuatan *use case ciagram*, *activity diagram*, *sequence diagram*, *class diagram*, *deployment diagram* dan juga pembuatan kamus data. Tahapan perancangan ini akan memberikan gambaran mengenai aliran informasi dan proses perancangan sistem informasi pemrosesan *line damage card*. *Prototype* sistem informasi pemrosesan *line damage card* yang telah selesai akan digunakan oleh pengguna untuk diketahui kekurangannya agar dapat diperbaiki.

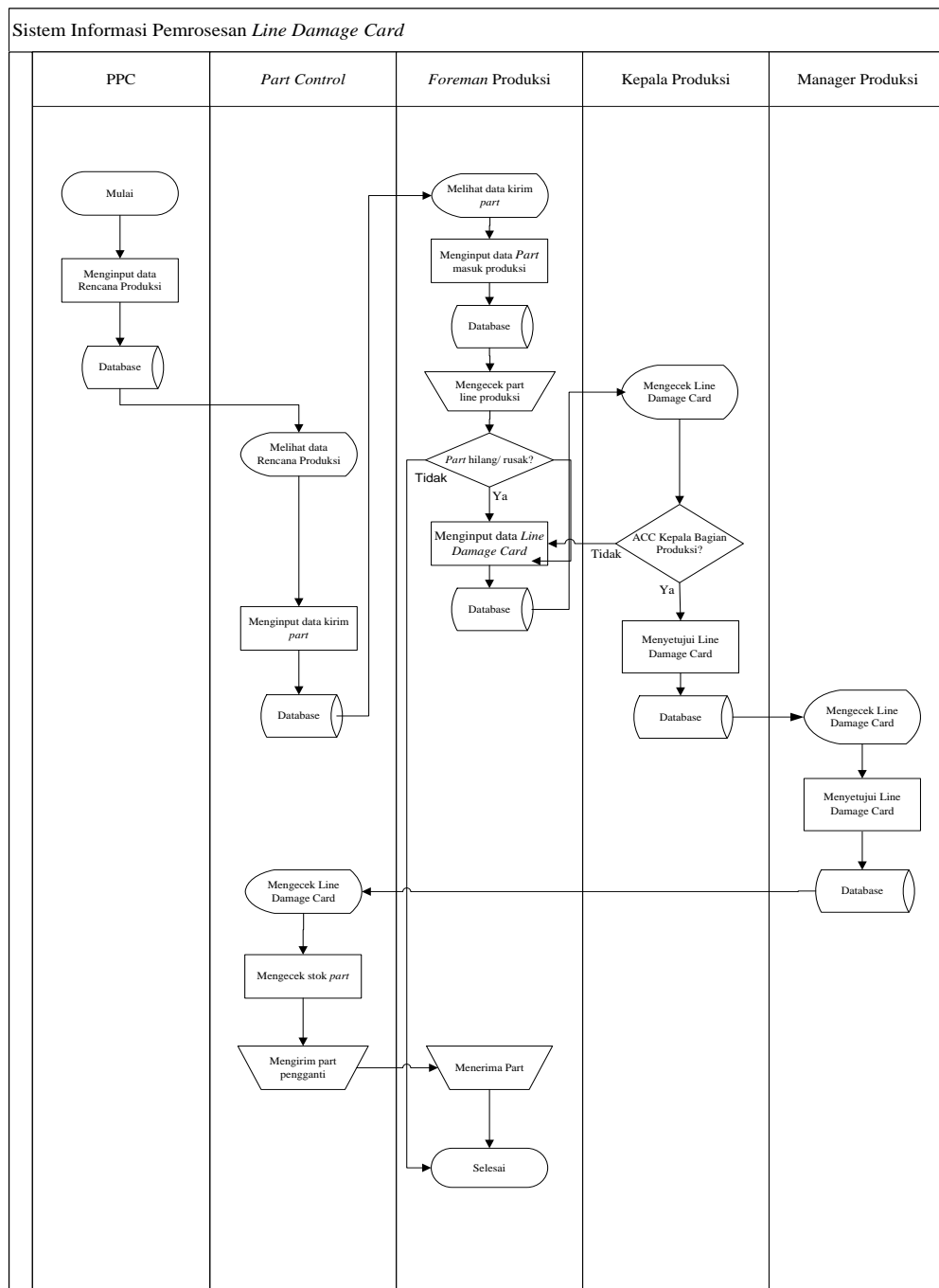
5.2.1 *Flowmap* Sistem Informasi Pemrosesan *Line Damage Card* yang *Diusulkan*

Alur proses sistem yang diusulkan berisi perancangan proses sistem. Perancangan dibuat untuk meminimalkan kekurangan dari sistem lama yang kurang efektif dan efisien ke sistem baru yang lebih terprogram dan cepat. Berikut ini adalah uraian dari alur proses sistem yang diusulkan:

1. Bagian PPC melakukan pengiputan data Rencana Produksi.
2. Bagian PPC mencetak data Rencana Produksi.
3. Bagian *Part Control* mengecek stok dan mengirim *part* ke bagian *Foreman* Produksi.
4. Bagian *Foreman* Produksi menerima *part* dan melakukan pengecekan *part* yang rusak atau hilang.
5. Jika ada *part* yang hilang atau rusak pada saat proses produksi, maka *Foreman* Produksi menginput data untuk *Line Damage Card*.
6. Kepala Produksi melihat *Line Damage Card*, lalu melakukan ACC oleh Kepala Bagian Produksi, dan meyetujui *Line Damage Card*.
7. Manager Produksi melihat data *Line Damage Card*, lalu melakukan ACC oleh Manager Produksi, dan menyetujui *Line Damage Card*.
8. Bagian *Part Control* melihat data *Line Damage Card*.

9. Bagian *Part Control* mengirim *part* pengganti kebagian *Foreman Produksi*.
10. Bagian *Foreman Produksi* menerima *part*.

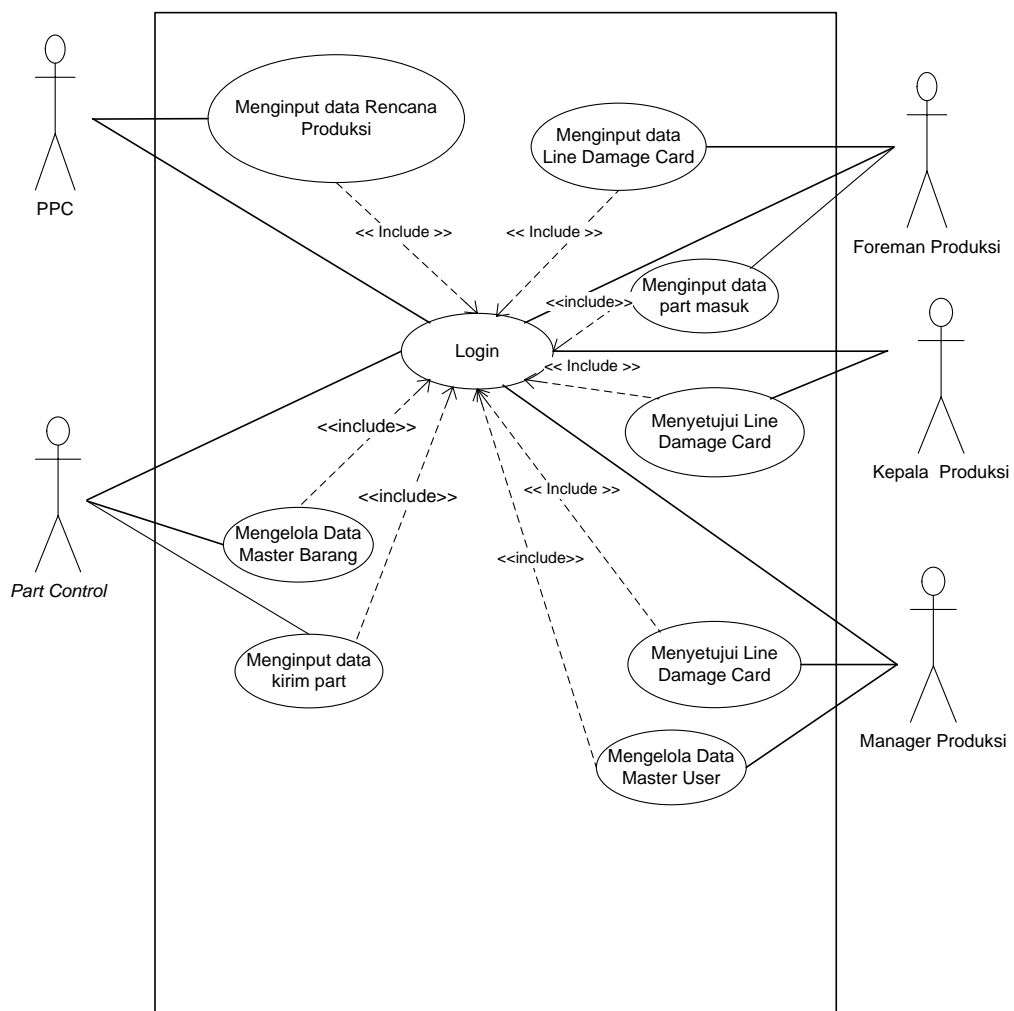
Berikut adalah perancangan alur proses sistem informasi pemrosesan *line damage card* yang diusulkan:



Gambar V.1 *Flowmap* Sistem Informasi Pemrosesan *Line Damage Card* yang Diusulkan
(Sumber: Hasil Pengolahan Data 2018)

5.2.2 Use Case Diagram Sistem Informasi Pemrosesan *Line Damage Card* yang Diusulkan

Use case diagram digunakan untuk mendeskripsikan sebuah interaksi antara aktor dengan sistem informasi yang akan dibuat. Berdasarkan analisis yang telah dilakukan maka *use case diagram* yang diusulkan pada sistem informasi pemrosesan *Line Damage Card* pada PT Krama Yudha Ratu Motor sebagai berikut:



Gambar V.2 Use Case Diagram Sistem Informasi Pemrosesan *Line Damage Card* yang Diusulkan

Sumber: Hasil Pengolahan Data (2018)

5.2.3 Use Case Description Sistem Informasi Pemrosesan Line Damage Card 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
<i>Primary Actor</i>	<i>PPC, Part Control, Foreman Produksi, Kepala Bagian Produksi, Manager Produksi.</i>
<i>Brief Description</i>	Sistem ini menggambarkan proses aktor masuk ke dalam sistem.
<i>Relationship</i>	<i>Association: PPC, Part Control, Foreman Produksi, Kepala Produksi, Manager Produksi.</i> <i>Include: -</i> <i>Extends: -</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>Login</i>. 2. Aktor memasukkan <i>user name</i> dan <i>password</i> pada <i>Login Form</i>. 3. Sistem mengecek ke <i>database</i>, apakah <i>user name</i> dan <i>password</i> sudah benar. 4. Jika <i>user name</i> dan <i>password</i> benar (valid), akan muncul tampilan menu utama sistem. 5. Jika <i>user name</i> dan <i>password</i> salah (tidak valid), maka muncul <i>message box</i> “Maaf, <i>user name</i> dan <i>password</i> Anda SALAH!” dan kembali ke <i>Login Form</i>.

(Sumber: Hasil Pengolahan Data 2018)

Tabel V.3 *Use Case Description* Menginput Data Rencana Produksi

Nama Use Case	Menginput Data Rencana Produksi
<i>Primary Actor</i>	PPC
<i>Brief Description</i>	Bagian PPC dapat membuat data rencana produksi.
<i>Relationship</i>	<i>Include: Login</i> <i>Generalisasi: Data User, id Kendaraan, Tipe Kendaraan, jumlah kendaraan</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Klik menu Rencana Produksi 2. Input ID Kendaraan. 3. Klik cari untuk menampilkan form Rencana Produksi. 4. Melakukan proses tambah, ubah, hapus data rencana produksi ke <i>database</i>
<i>Precondition</i>	<i>Login ke dalam sistem informasi Pemrosesan Line Damage Card</i>

(Sumber: Hasil Pengolahan Data 2018)

Tabel V.4 *Use Case Description* Menginput Data Part Kirim

Nama Use Case	Menginput Data Part Kirim
<i>Primary Actor</i>	<i>Part Control</i>
<i>Brief Description</i>	Bagian <i>Part Control</i> dapat membuat data kirim <i>part</i> .
<i>Relationship</i>	<i>Include: Login</i> <i>Generalisasi: Data User, id part, Tipe part, jumlah part.</i>

Tabel V.4 *Use Case Description* Menginput Data *Part* Kirim (Lanjutan)

Nama Use Case	Menginput Data <i>Part</i> Kirim
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. Klik menu Kirim Part 2. Memilih part. 3. Klik cari untuk menampilkan form Kirim Part 4. Melakukan proses tambah, ubah, hapus data kirim part ke <i>database</i>
<i>Precondition</i>	<i>Login ke dalam sistem informasi pemrosesan line damage card.</i>

(Sumber: Hasil Pengolahan Data 2018)

Tabel V.5 *Use Case Description* Menginput Data *Part* Masuk

Nama Use Case	Menginput Data <i>Part</i> Masuk
<i>Primary Actor</i>	<i>Foreman</i> Produksi
<i>Brief Description</i>	<i>Foreman</i> Produksi dapat menginput Data <i>Part</i> Masuk.
<i>Relationship</i>	<i>Include: Login</i> <i>Generalisasi: Data User, Part, Tipe Part, ID Part.</i>

Tabel V.5 *Use Case Description* Menginput Data *Part* Masuk

Nama Use Case	Menginput Data <i>Part</i> Masuk
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Foreman</i> Produksi dapat memilih menu Terima <i>Part</i>. 2. Masukkan kode kirim <i>part</i>. 3. Klik cari untuk menampilkan rincian data kirim <i>part</i>. 4. Pilih <i>part</i> yang akan ditambahkan ke data <i>part</i> masuk. 5. Klik Tambah Data Item Masuk. 6. Isi data jumlah <i>part</i> yang masuk sesuai dengan jumlah <i>part</i> yang dikirim.

Tabel V.5 *Use Case Description* Menginput Data *Part* Masuk (Lanjutan)

Nama Use Case	Menginput Data <i>Part</i> Masuk
<i>Normal Flow of Events</i>	7. Klik Save untuk menyimpan detail data <i>part</i> yang masuk. 8. Klik Selesai Menambahkan untuk menyimpan Data <i>Part</i> Masuk.
<i>Precondition</i>	<i>Login</i> ke dalam sistem informasi pemrosesan <i>Line Damage Card</i> .

(Sumber: Hasil Pengolahan Data 2018)

Tabel V.6 *Use Case Description* Master *User*

Nama Use Case	Master <i>User</i>
<i>Primary Actor</i>	Manager Produksi
<i>Brief Description</i>	Manager Produksi dapat melakukan Tambah, Hapus dan Ubah terhadap data Master <i>User</i> .
<i>Relationship</i>	<i>Include: Login</i>
<i>Normal Flow of Events</i>	1. Manager Produksi membuka menu Data Master. 2. Pilih submenu <i>User</i> . 3. Pilih Tambah untuk menambah data <i>user</i> . 4. Pilih Ubah untuk mengubah data <i>user</i> . 5. Pilih Hapus untuk menghapus data <i>user</i> . 6. Klik tombol Simpan untuk menyimpan data.
<i>Precondition</i>	<i>Login</i> ke dalam sistem informasi pemrosesan <i>line damage card</i> .

(Sumber: Hasil Pengolahan Data, 2018)

Tabel V.7 *Use Case Description Master Barang*

Nama Use Case	Master Part
<i>Primary Actor</i>	<i>Part Control</i>
<i>Brief Description</i>	<i>Part Control</i> dapat melakukan Tambah, Hapus, dan Ubah terhadap data Master Stok Barang
<i>Relationship</i>	<i>Include: Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Part Control</i> membuka menu Data Master. 2. Pilih submenu Barang. 3. Pilih Tambah untuk menambah data barang. 4. Pilih Ubah untuk mengubah data barang. 5. Pilih Hapus untuk menghapus data barang. 6. Klik tombol Simpan untuk menyimpan data.
<i>Precondition</i>	<i>Login</i> ke dalam sistem informasi pemrosesan <i>line damage card</i>

(Sumber: Hasil Pengolahan Data, 2018)

Tabel V.8 *Use Case Description Membuat Line Damage Card*

Nama Use Case	Membuat Line Damage Card
<i>Primary Actor</i>	<i>Foreman Produksi</i>
<i>Brief Description</i>	<i>Foreman produksi</i> dapat membuat data <i>part</i> rusak pada <i>form Line Damage</i>
<i>Relationship</i>	<i>Association: Kepala Bagian Produksi, Manager Produksi, Part Control.</i> <i>Include: Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>Foreman Produksi</i> membuka menu <i>form Line Damage Card</i>. 2. Menginput data yang berkaitan dengan <i>part</i> rusak. 3. Klik tombol Simpan untuk menyimpan data.

Tabel V.8 *Use Case Description* Membuat *Line Damage Card* (Lanjutan)

Nama Use Case	Membuat <i>Line Damage Card</i>
<i>Precondition</i>	<i>Login</i> ke dalam sistem informasi pemrosesan <i>line damage card</i> .

(Sumber: Hasil Pengolahan Data 2018)

Tabel V.9 *Use Case Description* Persetujuan *Line Damage Card*

Nama Use Case	Persetujuan <i>Line Damage Card</i>
<i>Primary Actor</i>	Kepala Produksi, Manager Produksi.
<i>Brief Description</i>	Menggambarkan proses memberi persetujuan pada <i>Line Damage Card</i> yang dilakukan oleh Kepala Produksi, dan Manager Produksi.
<i>Relationship</i>	<i>Association:</i> Kepala Produksi, Manager Produksi. <i>Include:</i> <i>Login</i> , Membuat <i>Line Damage Card</i>
<i>Normal Flow of Events</i>	1. Kepala Produksi, dan Manager Produksi membuka menu <i>Form Line Damage</i> . 2. Pilih Setuju untuk menyetujui/ Tidak Setuju untuk menolak persetujuan. 3. Pilih Simpan
<i>Precondition</i>	<i>Login</i> ke dalam sistem informasi pemrosesan <i>line damage card</i> .

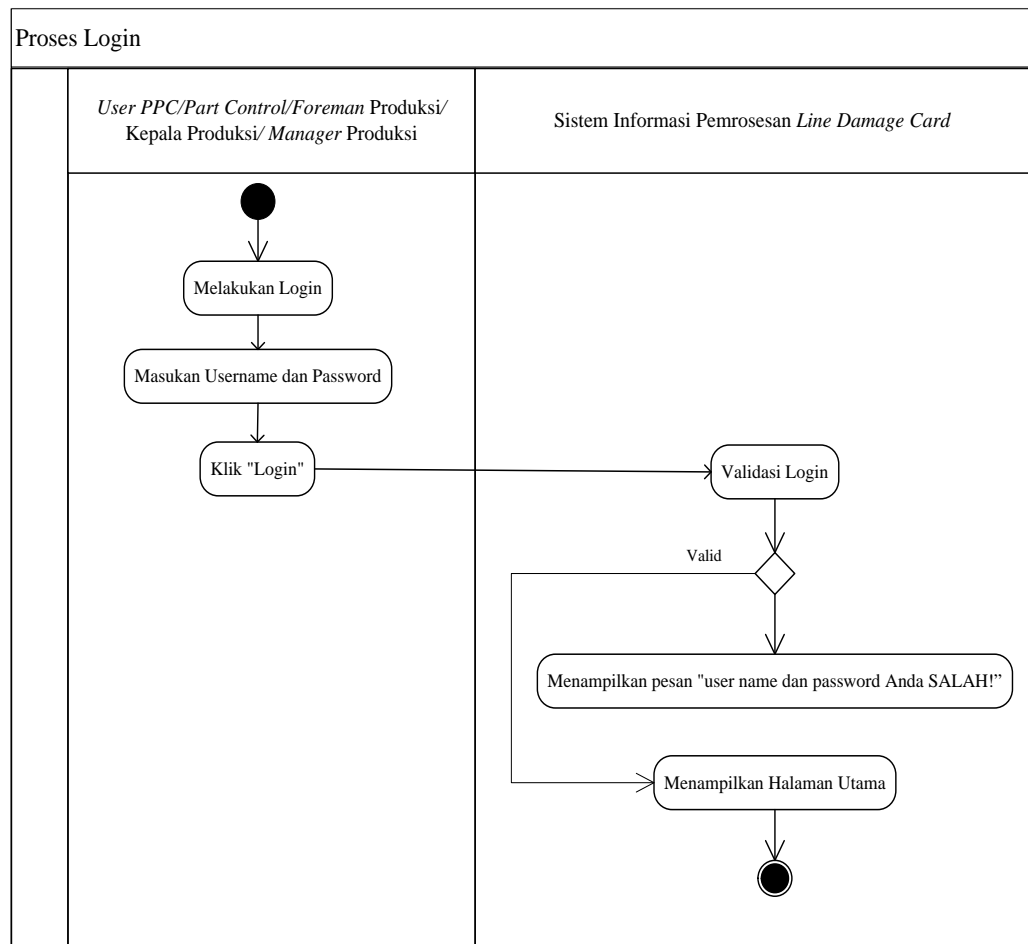
(Sumber: Hasil Pengolahan Data 2018)

5.2.4 *Activity Diagram* Yang Diusulkan

Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam sistem. *Activity diagram* bertujuan supaya lebih memahami sistem yang akan dibuat.

1. *Activity Diagram* Proses *Login*

Diagram ini menggambarkan aktivitas kegiatan akses *login* terhadap sistem. Berikut adalah kegiatan proses *login* yang diusulkan:



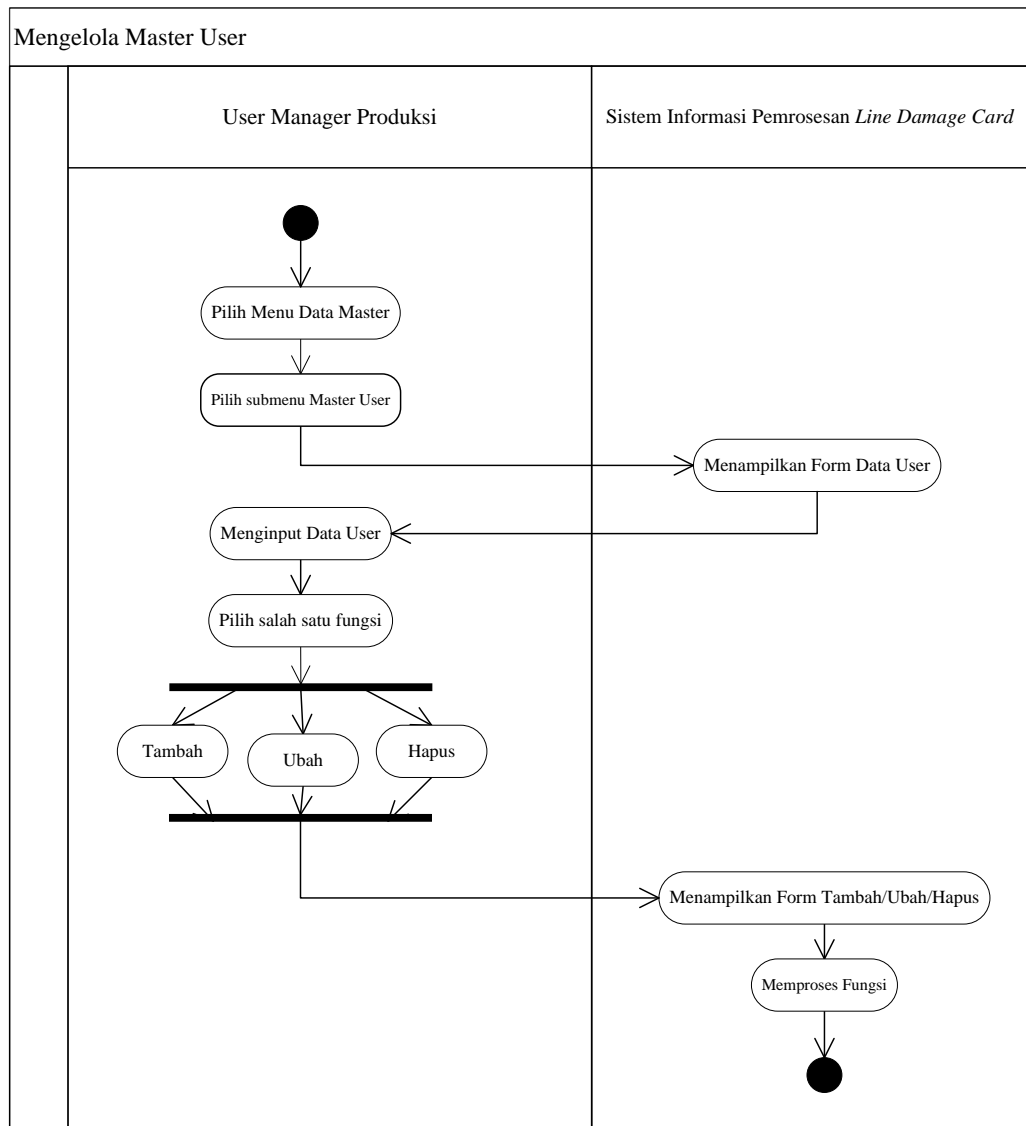
Gambar V.3 Activity Diagram Proses Login

(Sumber: Hasil Pengolahan Data 2018)

2. Activity Diagram Mengelola Master User

Diagram ini menggambarkan kegiatan pengelolaan data pengguna sistem (*user*) pada usulan sistem informasi persediaan dan pengiriman *part*.

Berikut kegiatan pengelolaan data *user* yang diusulkan:

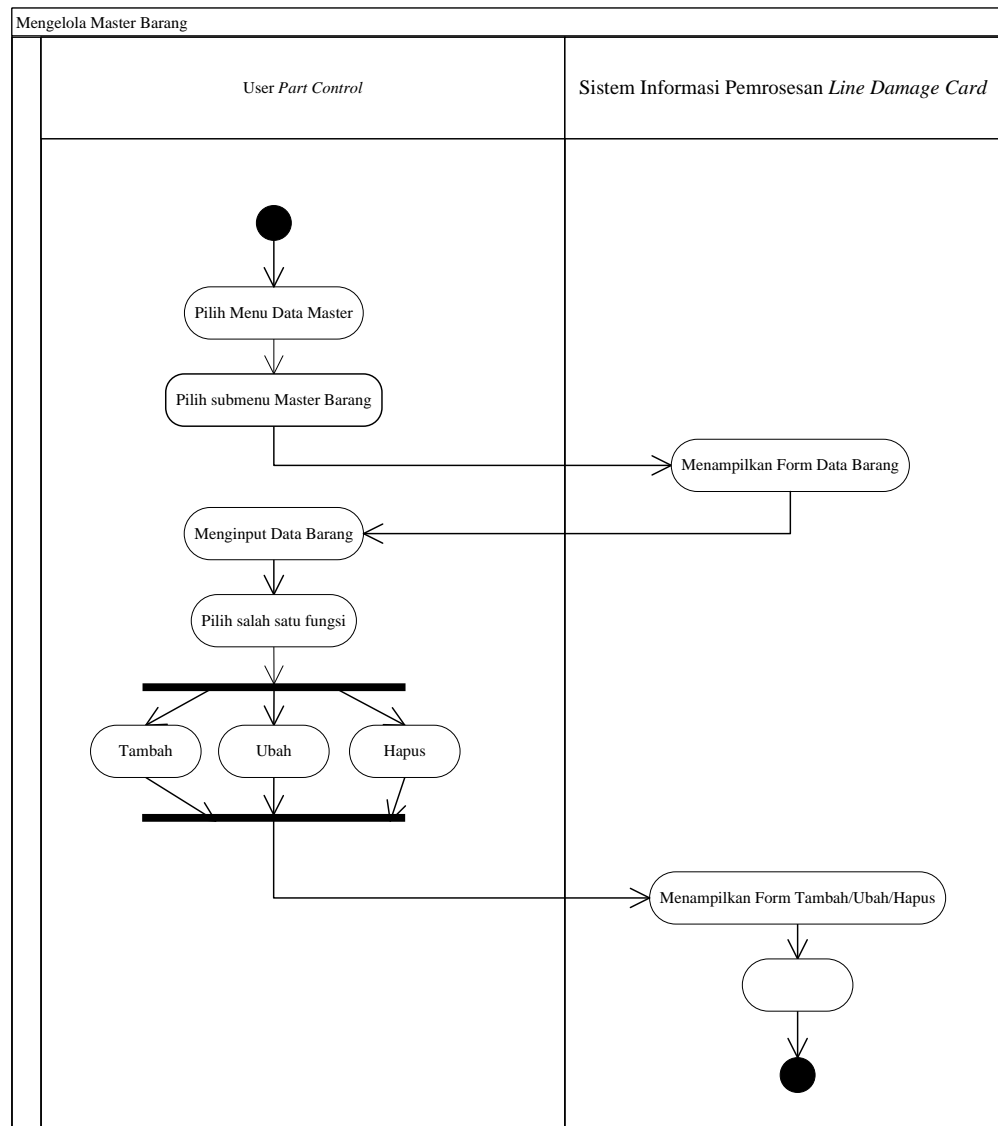


Gambar V.4 *Activity Diagram* Mengelola Master User

Sumber: Hasil Pengolahan Data (2018)

3. *Activity Diagram* Mengelola Master Barang

Diagram ini menggambarkan kegiatan pengelolaan data barang pada usulan sistem informasi persediaan dan pengiriman *part*. Kegiatan pengelolaan data barang merupakan interaksi antara bagian user dan sistem. Berikut kegiatan pengelolaan data barang yang diusulkan:

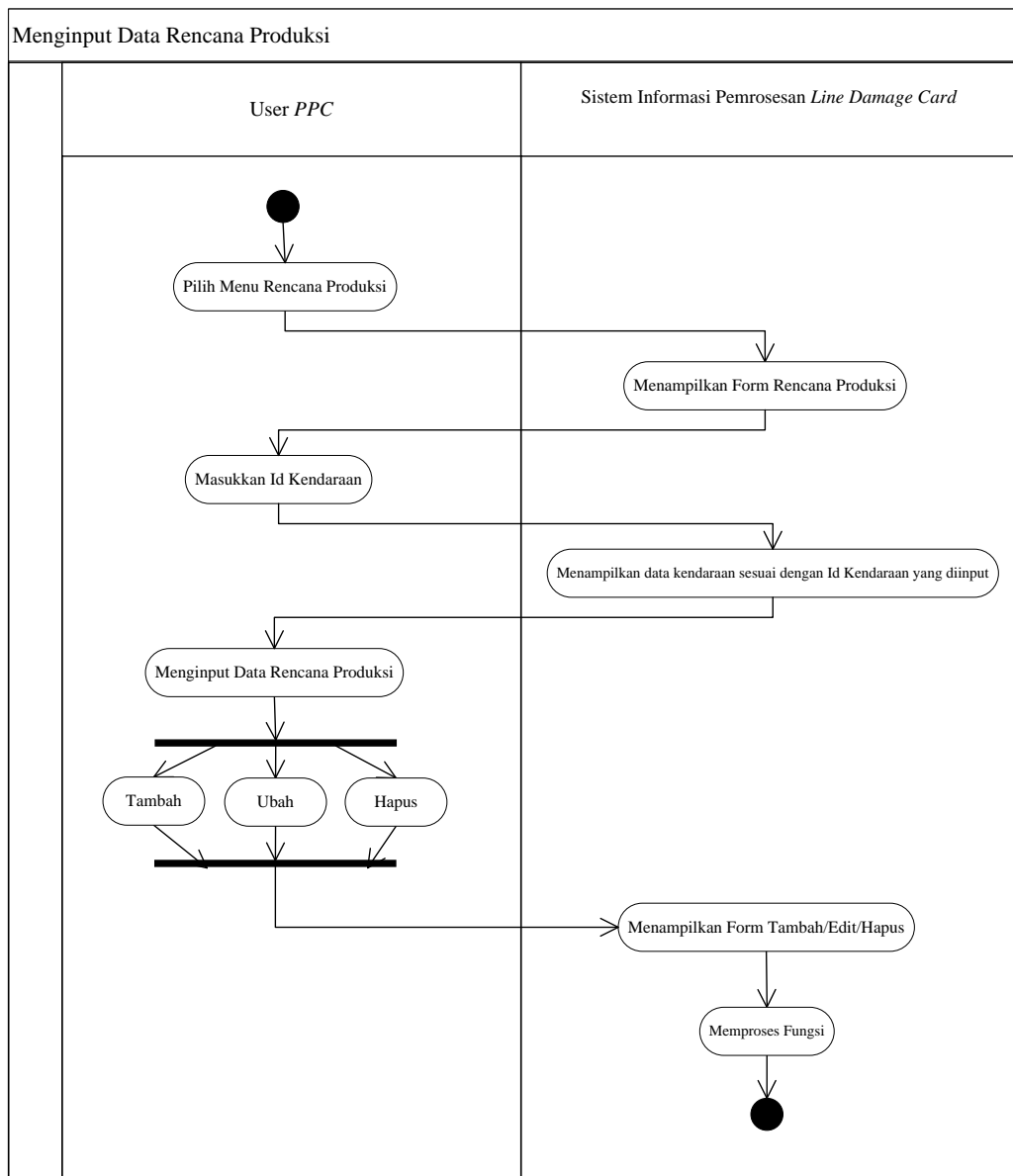


Gambar V.5 Activity Diagram Mengelola Master Barang

Sumber: Hasil Pengolahan Data (2018)

4. Activity Diagram Menginput Data Rencana Produksi.

Diagram ini menggambarkan kegiatan menginput data rencana produksi pada usulan sistem informasi persediaan dan pengiriman *part*. Kegiatan pengelolaan data barang merupakan interaksi antara bagian user dan sistem. Berikut kegiatan pengelolaan data barang yang diusulkan:

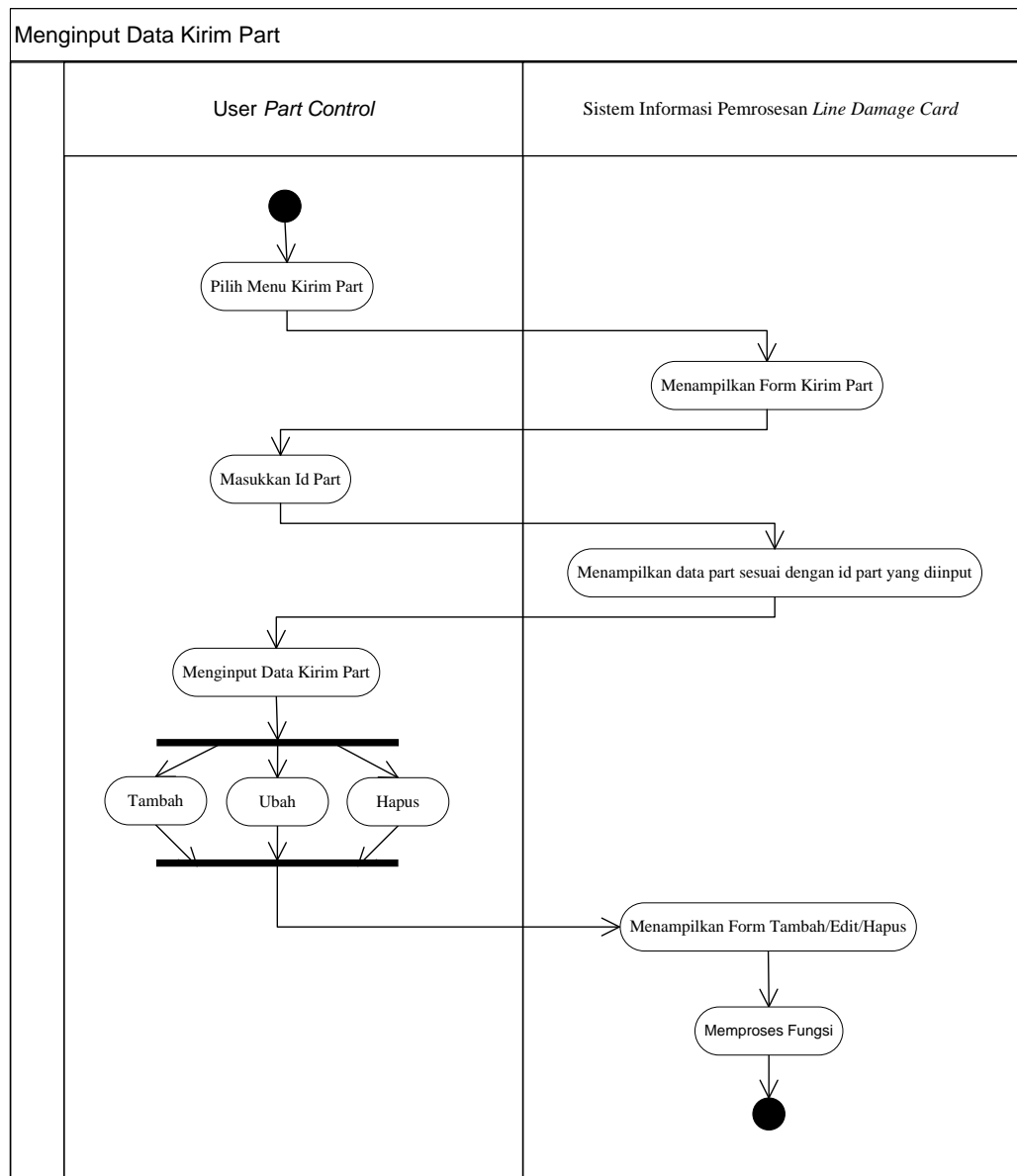


Gambar V.6 *Activity Diagram* Menginput Data Rencana Produksi

Sumber: Hasil Pengolahan Data (2018)

5. *Activity Diagram* Menginput Data Kirim Part.

Diagram ini menggambarkan kegiatan menginput data kirim *part* pada usulan sistem informasi persediaan dan pengiriman *part*. Kegiatan pengelolaan data barang merupakan interaksi antara bagian user dan sistem. Berikut kegiatan pengelolaan data barang yang diusulkan:

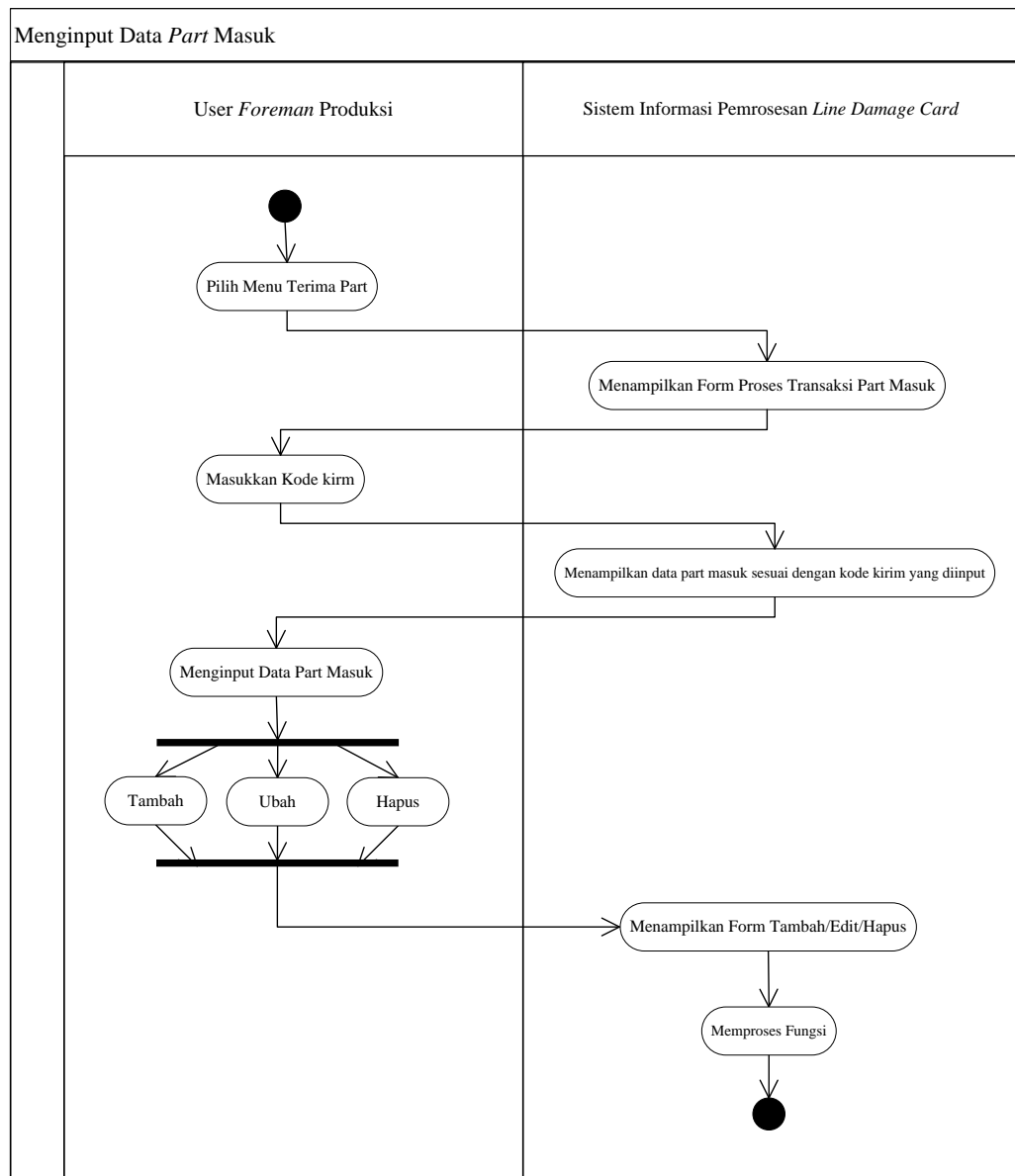


Gambar V.7 *Activity Diagram* Menginput Data Kirim Part

Sumber: Hasil Pengolahan Data (2018)

6. *Activity Diagram* Menginput Data *Part* Masuk

Diagram ini menggambarkan kegiatan penginputan data *part* yang masuk ke line produksi seperti: id, tipe, dan nama part.

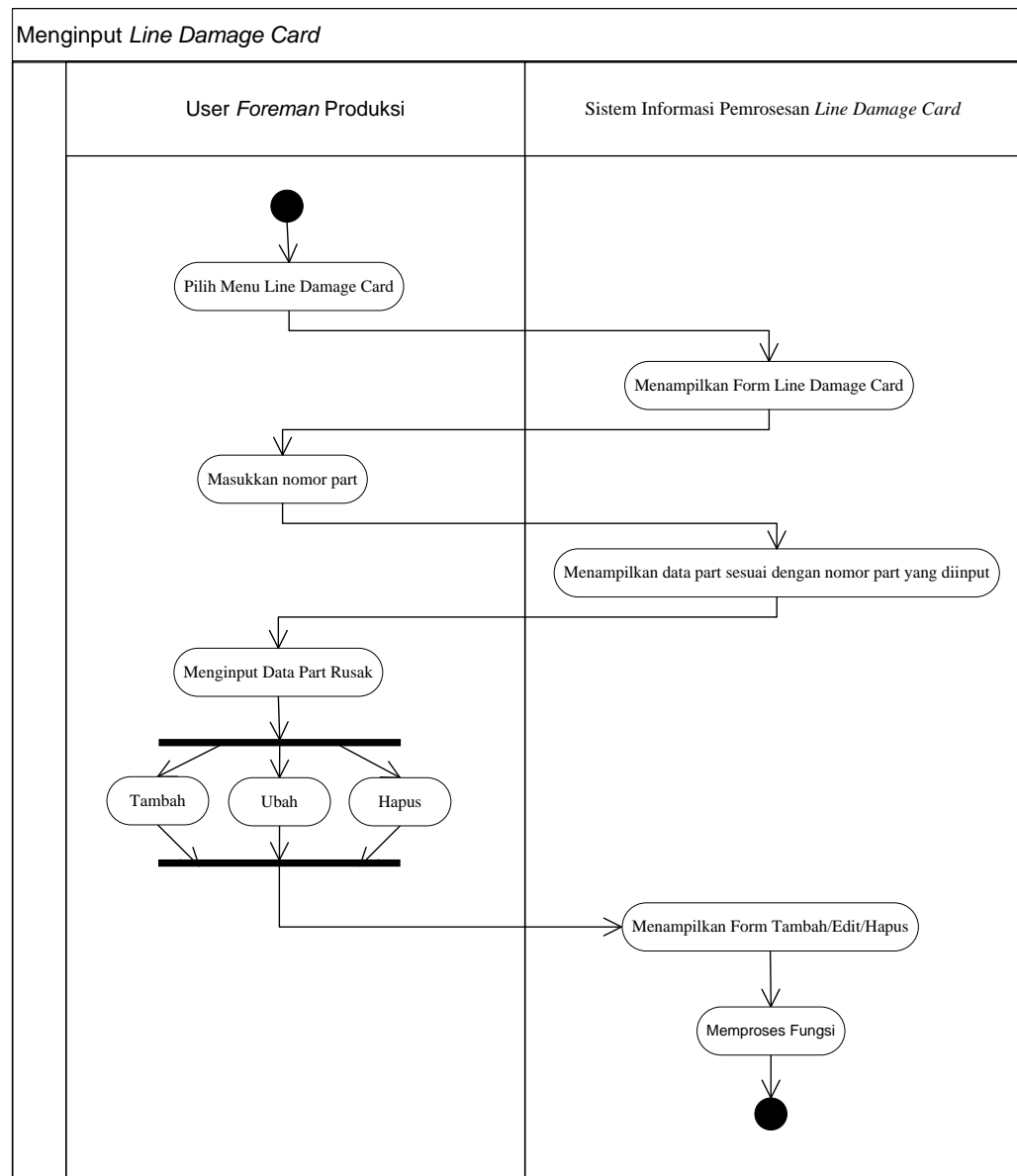


Gambar V.8 *Activity Diagram* Menginput Data *Part* Masuk

Sumber: Hasil Pengolahan Data (2018)

7. *Activity Diagram* Menginput *Line Damage Card*

Diagram ini menggambarkan kegiatan pembuatan data part yang rusak seperti: nomor, tipe, dan nama part serta penyebab kerusakan part dan cara penanggulangan yang harus dilakukan pada kerusakan part tersebut.

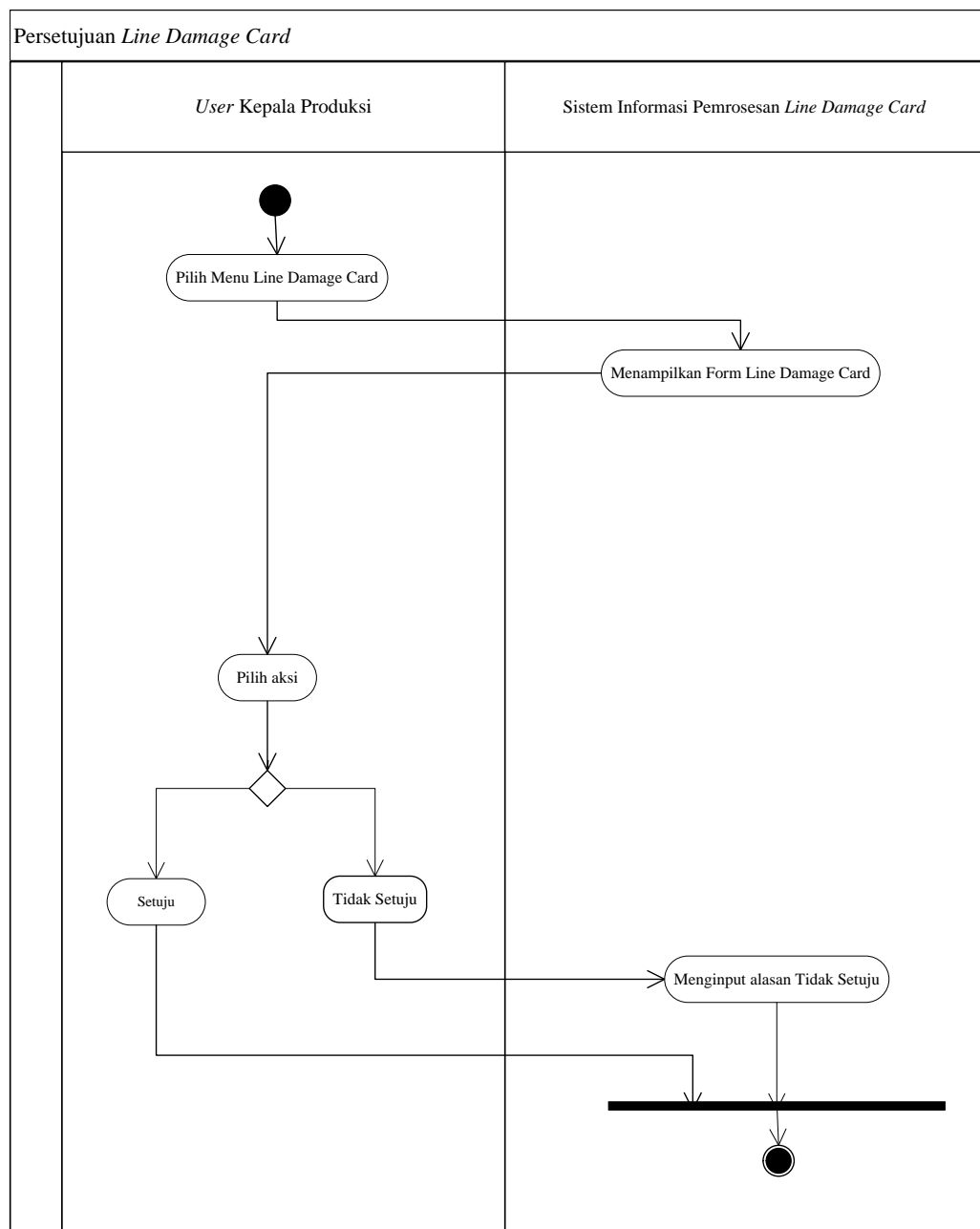


Gambar V.9 Activity Diagram Menginput *Line Damage Card*

Sumber: Hasil Pengolahan Data (2018)

8. Activity Diagram Persetujuan *Line Damage Card* Oleh Kepala Bagian Produksi

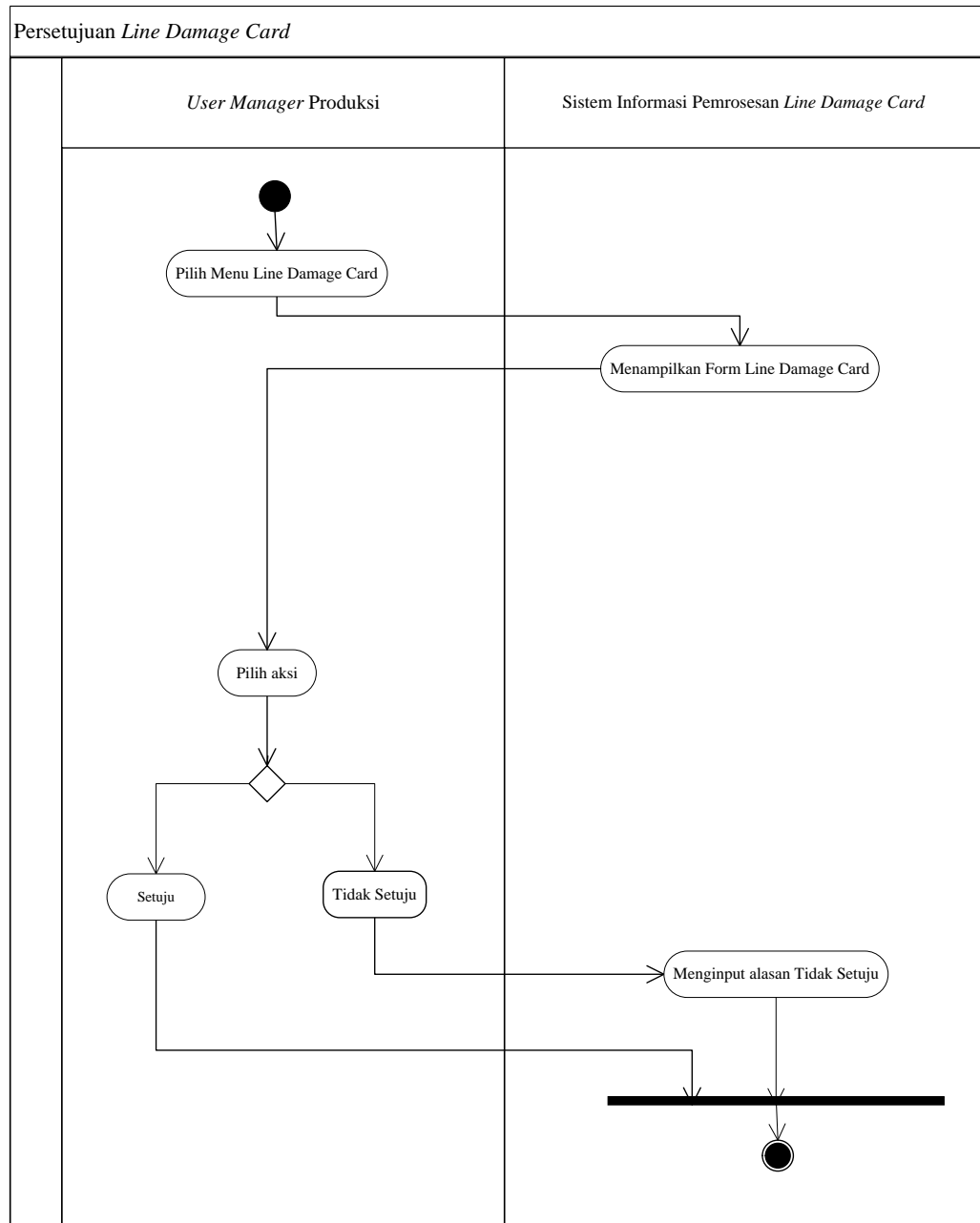
Diagram ini menggambarkan kegiatan memberikan persetujuan pada *Line Damage Card* sebagai tahapan proses untuk penggantian part. Kegiatan ini dilakukan oleh Kepala Produksi.



Gambar V.10 *Activity Diagram* Persetujuan *Line Damage Card* Oleh Kepala Bagian Produksi

Sumber: Hasil Pengolahan Data (2018)

9. *Activity Diagram Persetujuan Line Damage Card Oleh Manager Produksi*
 Diagram ini menggambarkan kegiatan memberikan persetujuan pada *Line Damage Card* sebagai tahapan proses untuk penggantian part. Kegiatan ini dilakukan oleh Manager Produksi.



Gambar V.11 *Activity Diagram Persetujuan Line Damage Card Oleh Manager Produksi*

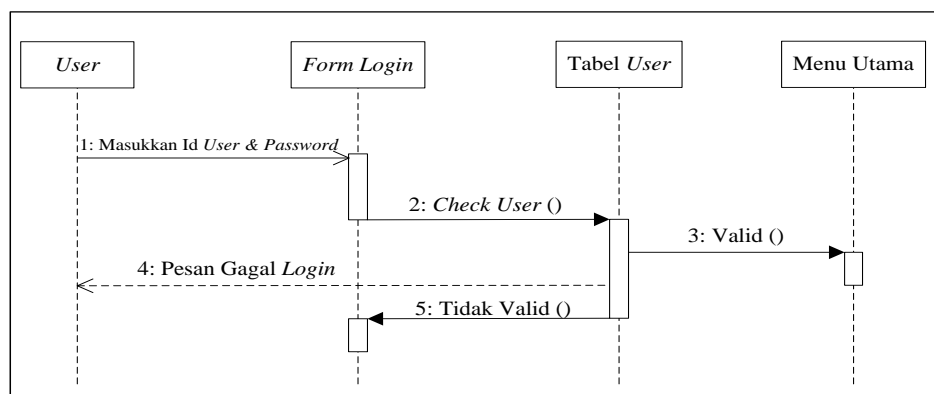
Sumber: Hasil Pengolahan Data (2018)

5.2.5 Sequence Diagram Yang Diusulkan

Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*. Berikut ini merupakan beberapa gambar *sequence diagram* dari Sistem Informasi Pemrosesan *Line Damage Card*.

1. Sequence Diagram Login

User memulai login dengan mengakses *form login*, kemudian menginputkan *username* dan *password*. Sistem melakukan validasi dengan mengecek hasil inputan *User* apakah sesuai dengan yang tersimpan didalam basis data *User*. Hasil pengecekan dapat menghasilkan data valid atau tidak. Sistem akan menampilkan *display menu* atau pesan jika inputan tidak valid kepada pengguna sistem.



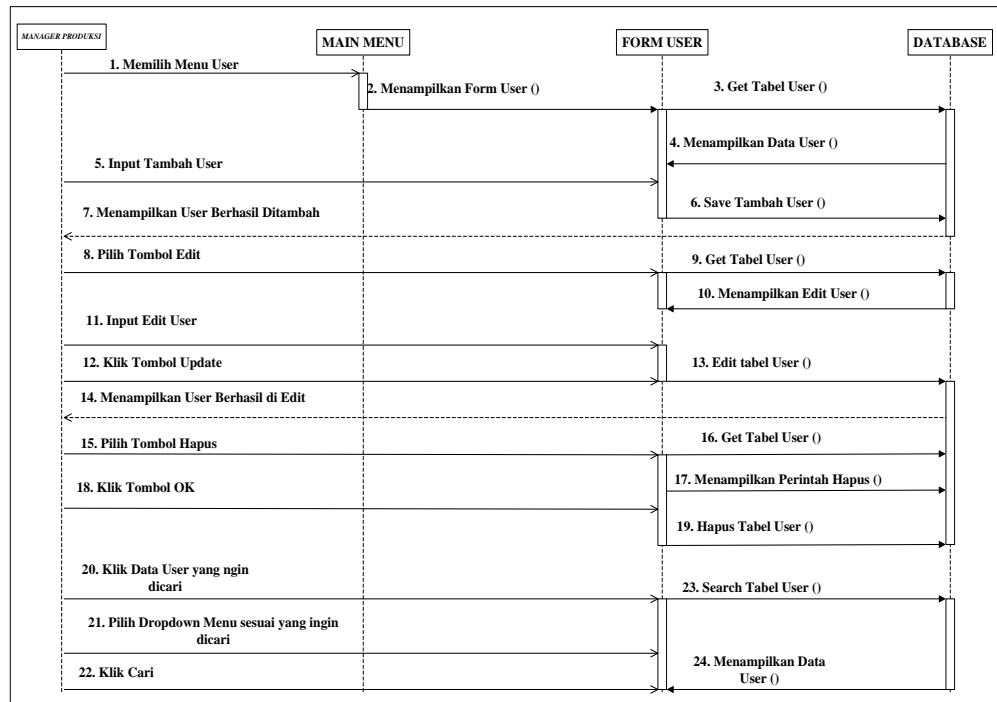
Gambar V.12 Sequence Diagram Login

(Sumber: Hasil Pengolahan Data, 2018)

2. Sequence Diagram Data User

Proses *sequence* dimulai ketika *administrator* memilih menu *user*, kemudian melengkapi *field* yang disediakan pada halaman data *user*. Setelah data dianggap valid, *simpan*, *edit* dan *hapus* di-*execute* oleh

sistem, lalu manipulasi data simpan, ubah dan hapus pada tabel *user* di *database* terjadi.

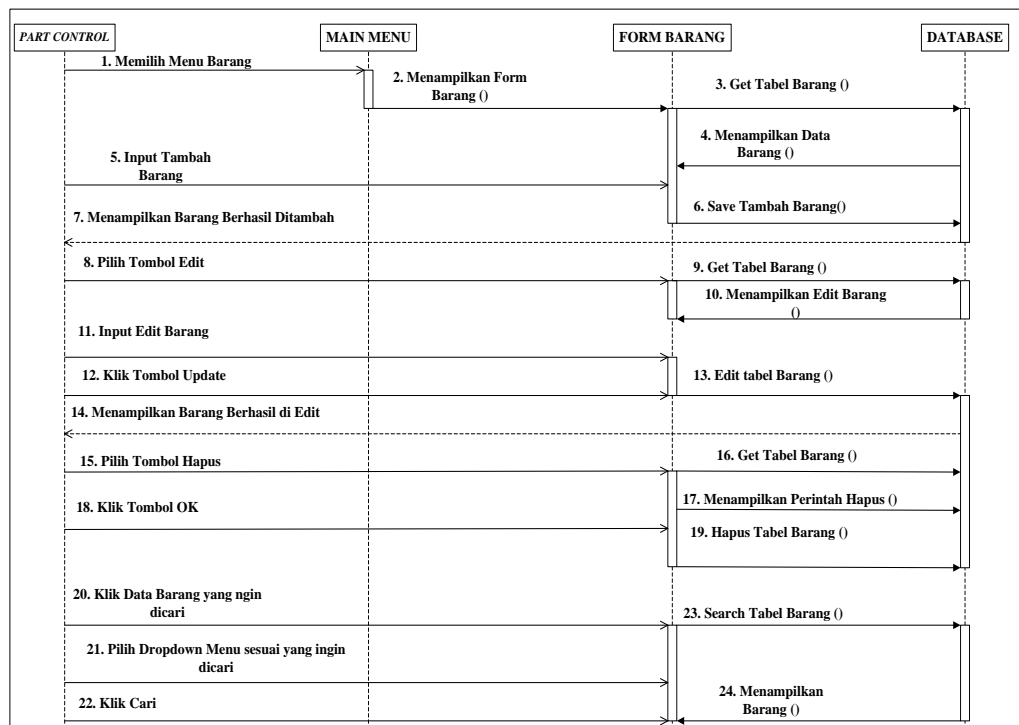


Gambar V.13 *Sequence Diagram Data User*

(Sumber: Hasil Pengolahan Data, 2018)

3. Sequence Diagram Data Barang

Proses *sequence* dimulai ketika *administrator* memilih menu part, kemudian melengkapi *field* yang disediakan pada halaman data barang. Setelah data dianggap valid, simpan, ubah dan hapus di-*execute* oleh sistem, lalu manipulasi data simpan, edit dan hapus pada tabel barang di *database* terjadi.

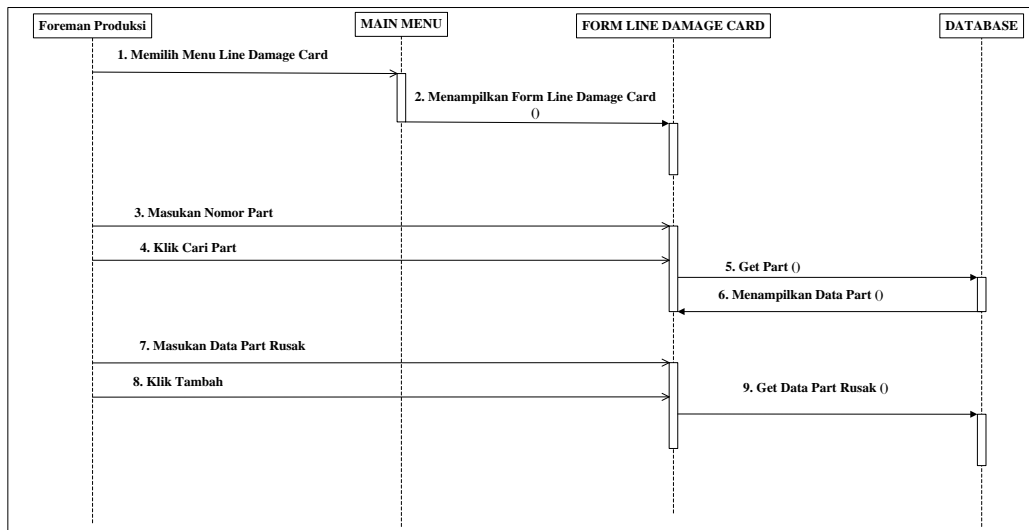


Gambar V.14 Sequence Diagram Data Barang

(Sumber: Hasil Pengolahan Data, 2018)

4. Sequence Diagram Rencana Produksi

Proses *sequence* dimulai ketika PPC memilih menu Rencana Produksi, kemudian melengkapi *field* yang disediakan pada *form* Rencana Produksi. Setelah data diinput klik tombol cari, lalu klik centang selesai menambahkan. Lalu data yang diinput akan tampil dalam bentuk *form*.

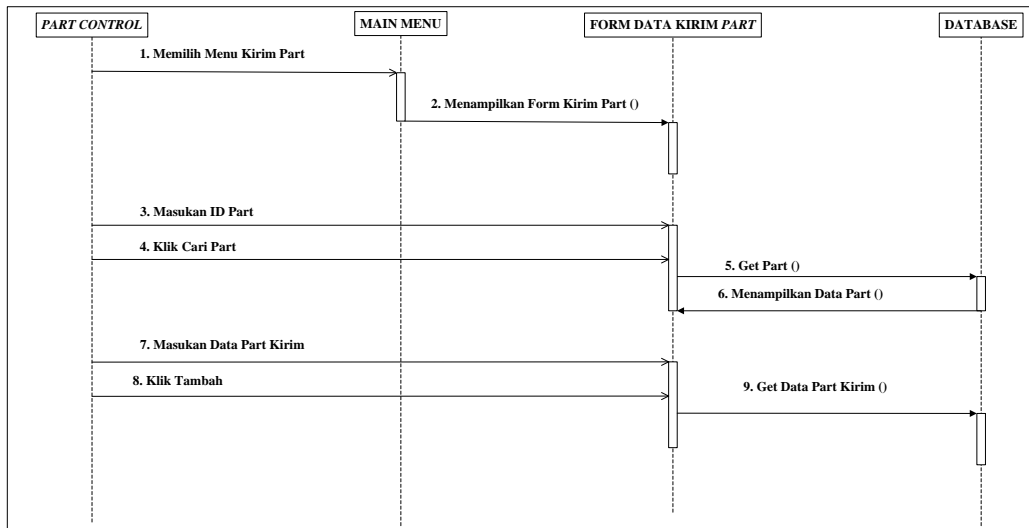


Gambar V.15 Sequence Diagram Rencana Produksi

(Sumber: Hasil Pengolahan Data, 2018)

5. Sequence Diagram Data Kirim Part

Proses *sequence* dimulai ketika *Part Control* memilih menu Kirim Part, kemudian melengkapi *field* yang disediakan pada *form* Kirim Part. Setelah data diinput klik tombol cari, lalu klik centang selesai menambahkan. Lalu data yang diinput akan tampil dalam bentuk *form*.

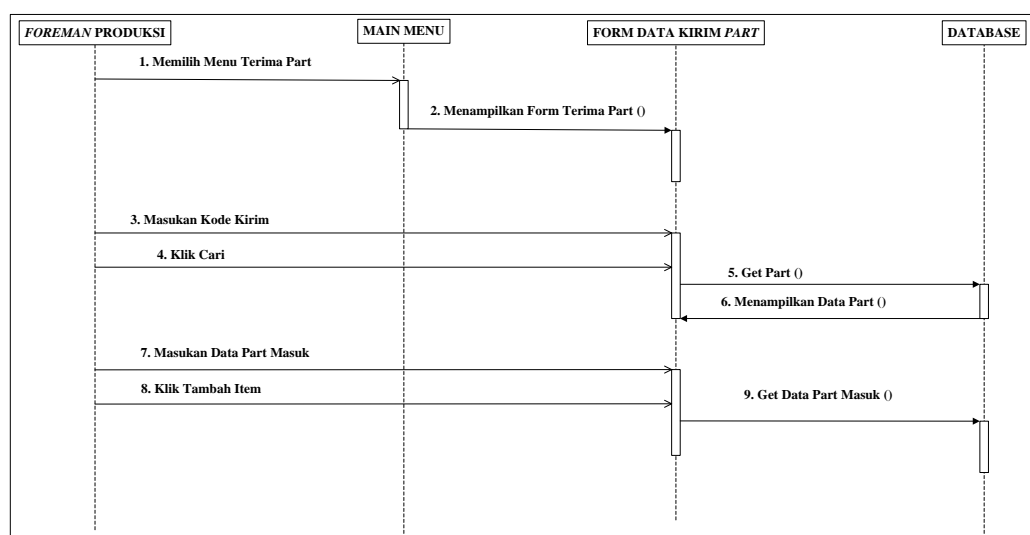


Gambar V.16 Sequence Diagram Data Kirim Part

(Sumber: Hasil Pengolahan Data, 2018)

6. Sequence Diagram Data Part Masuk

Proses *sequence* dimulai ketika *Foreman* Produksi memilih menu Terima Part, kemudian melengkapi *field* yang disediakan pada *form* Terima Part. Setelah data diinput klik tombol cari, lalu klik centang selesai menambahkan. Lalu data yang diinput akan tampil dalam bentuk *form*.

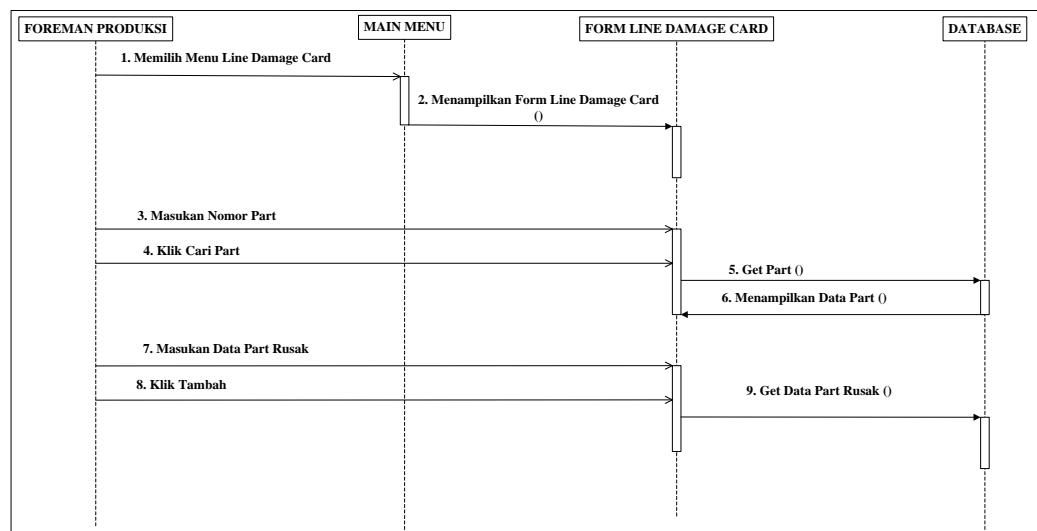


Gambar V.17 *Sequence Diagram* Data Part Masuk

(Sumber: Hasil Pengolahan Data, 2018)

7. Sequence Diagram Menginput Line Damage Card

Proses *sequence* dimulai ketika *Foreman* Produksi memilih menu *Line Damage Card*, kemudian memilih nomor *part* dan menginput data *part* yang rusak. Setelah data dianggap valid, simpan, edit dan hapus di-*execute* oleh sistem, lalu manipulasi data simpan, edit dan hapus pada tabel barang di *database* terjadi.

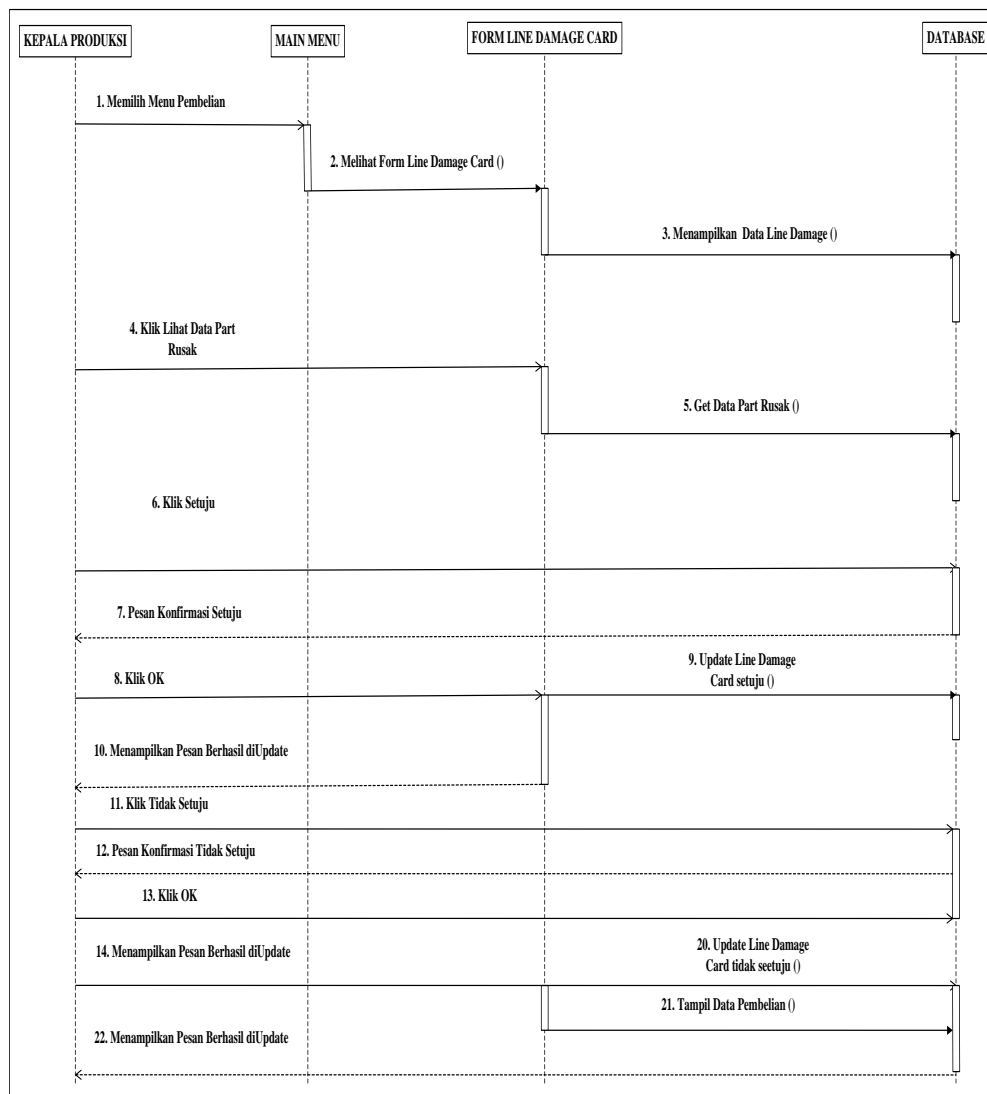


Gambar V.18 *Sequence Diagram* Menginput *Line Damage Card*

(Sumber: Hasil Pengolahan Data, 2018)

8. *Sequence Diagram* Persetujuan *Line Damage Card* oleh Kepala Bagian Produksi

Proses *sequence* dimulai ketika Kepala Produksi memilih menu *Line Damage Card*, kemudian melengkapi *field* yang disediakan pada *form Line Damage Card*. Setelah data dianggap valid, simpan, edit dan hapus di-*execute* oleh sistem, lalu manipulasi data simpan, edit dan hapus pada tabel barang di *database* terjadi.

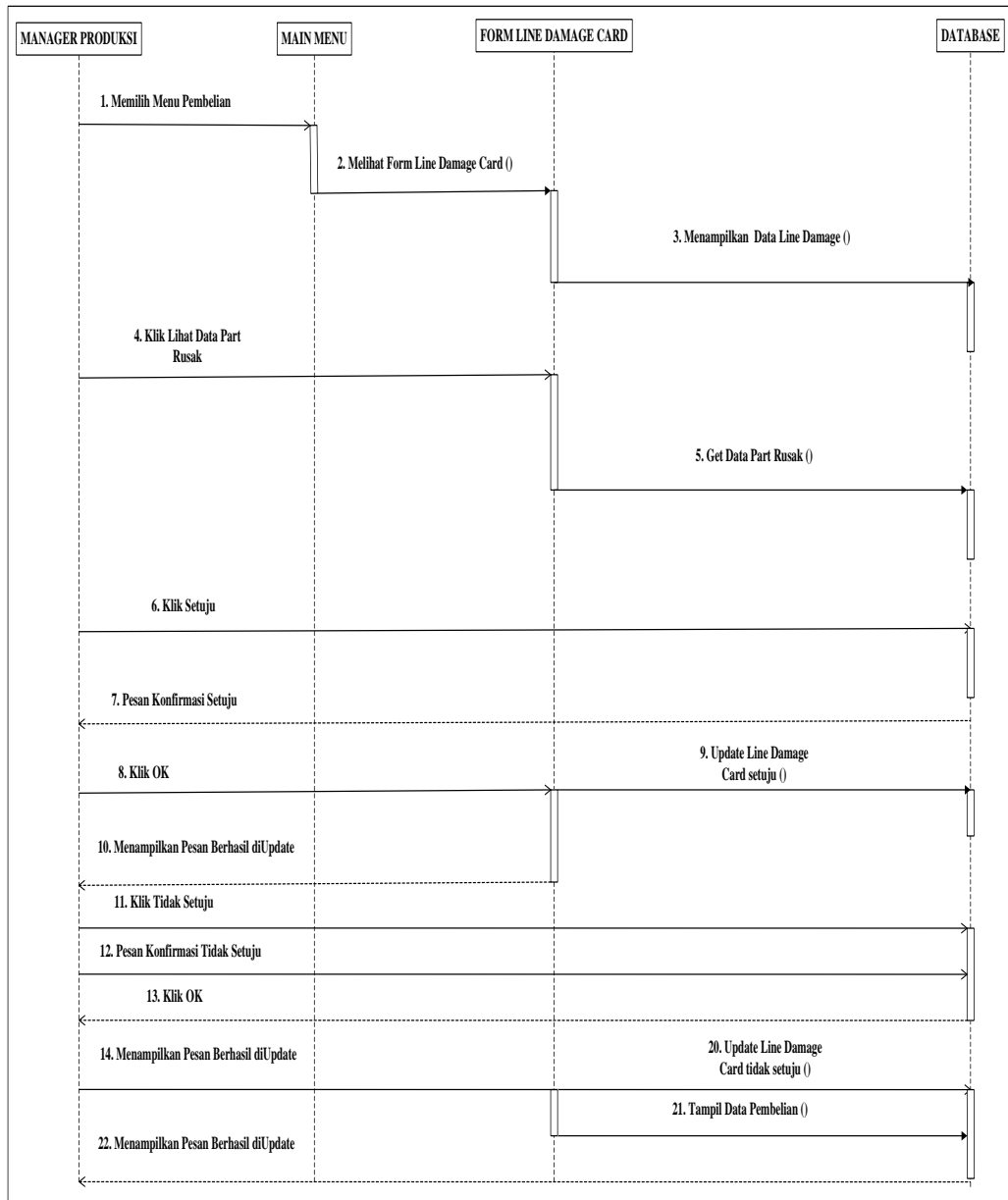


Gambar V.19 *Sequence Diagram* Persetujuan *Line Damage Card* oleh Kepala Bagian Produksi

(Sumber: Hasil Pengolahan Data, 2018)

9. *Sequence Diagram* Persetujuan *Line Damage Card* oleh Manager Produksi
 Proses *sequence* dimulai ketika Manager Produksi memilih menu *Line Damage Card*, kemudian melengkapi *field* yang disediakan pada *form Line Damage Card*. Setelah data dianggap valid, simpan, ubah dan hapus di-

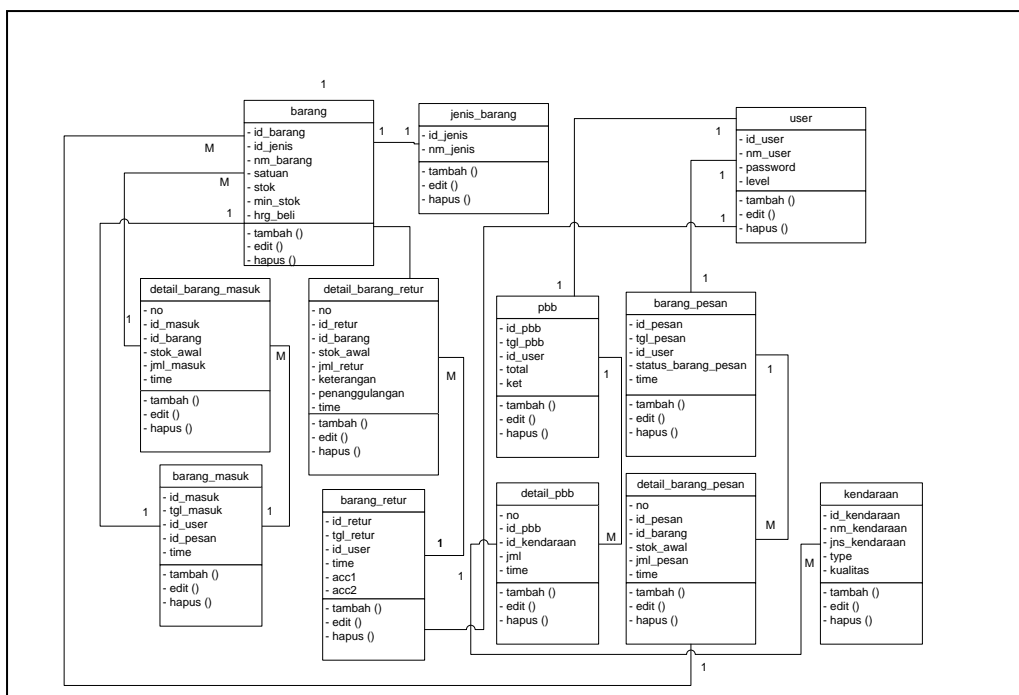
execute oleh sistem, lalu manipulasi data simpan, ubah, dan hapus pada tabel barang di *database* terjadi.



Gambar V.20 Sequence Diagram Persetujuan Line Damage Card oleh Manager Produksi (Sumber: Hasil Pengolahan Data, 2018)

5.2.6 Class Diagram Yang Diusulkan

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem informasi pemrosesan *line damage card*. Class diagram memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. Class diagram sistem informasi pemrosesan *line damage card* yang diusulkan dapat dilihat pada Gambar V.13 sebagai berikut:



Gambar V.21 Class Diagram Yang Diusulkan

(Sumber: Hasil Pengolahan Data, 2018)

5.3 Entity Relationship Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Untuk menggambarkannya digunakan beberapa notasi dan simbol. ERD sistem informasi pemrosesan *line damage card* yang diusulkan dapat dilihat pada Gambar V.22.

5.4 Kamus Data

Kamus data membantu dalam pendefinisian data agar pendefinisian data tersebut dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan). Berikut ini adalah kamus data yang terdapat dalam aplikasi sistem informasi pemrosesan *line damage card*:

1. Spesifikasi Tabel *User*

Nama Tabel : *user*

Fungsi : Untuk menyimpan data *user*

Tipe : File data *master*

Tabel V.10 Tabel Data *User*

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
Id_user	<i>Integer</i>	16	<i>Primary key</i>	Nomor Id pengguna
nm_user	<i>Varchar</i>	25		Nama pengguna
<i>password</i>	<i>Varchar</i>	15		User Name pengguna
Akses Level	<i>Varchar</i>	15		Password Pengguna

(Sumber: Hasil Analisis Data, 2018)

2. Spesifikasi Tabel Data Barang

Nama Tabel : Data Barang

Fungsi : Untuk menyimpan data barang

Tipe : File data *Master*

Tabel V.11 Tabel Data Barang

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
id_barang	<i>Varchar</i>	16	<i>Primary key</i>	Kode Part
Id_jenis	<i>Varchar</i>	10	<i>Foreign Key</i>	Nama Part
nm_barang	<i>Varchar</i>	30		Tipe Part
Satuan	<i>Varchar</i>	15		Jenis Part

Tabel V.11 Tabel Data Barang (Lanjutan)

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
Stok	<i>Integer</i>	4		Stok Part
min_stok	<i>Integer</i>	4		Minimal Stok
Hrg_beli	<i>Integer</i>	11		Harga Beli

(Sumber: Hasil Analisis Data, 2018)

3. Spesifikasi Tabel Jenis Barang

Nama Tabel : Jenis_Barang

Fungsi : Untuk menyimpan data jenis barang

Tipe : File data Transaksi

Tabel V.12 Tabel Jenis Barang

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
id_jenis	<i>Varchar</i>	10	<i>Primary key</i>	Id Jenis
nm_jenis	<i>Varchar</i>	30		Nama Jenis

(Sumber: Hasil Analisis Data, 2018)

4. Spesifikasi Tabel Kendaraan

Nama Tabel : kendaraan

Fungsi : Untuk menyimpan data kendaraan

Tipe : File data *master*

Tabel V.13 Tabel Data Kendaraan

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
Id_kendaraan	<i>Varchar</i>	10	<i>Primary key</i>	Id Kendaraan
nm_kendaraan	<i>Varchar</i>	30		Nama Kendaraan
Jns_kendaraan	<i>Varchar</i>	10		Jenis Kendaraan
<i>type</i>	<i>Varchar</i>	20		Tipe Kendaraan
kualitas	<i>Varchar</i>	10		Kualitas Kendaraan

(Sumber: Hasil Analisis Data, 2018)

5. Spesifikasi Tabel Pemesanan Bahan Baku

Nama Tabel : Pbb

Fungsi : Untuk menyimpan data rencana produksi

Tipe : File data Transaksi

Tabel V.14 Tabel Data Pemesanan Bahan Baku

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
Id_pbb	<i>Varchar</i>	30	<i>Primary key</i>	Id Pbb
Tgl_pbb	<i>Date</i>			Tanggal Pbb
Id_user	<i>Varchar</i>	16	<i>Foreign Key</i>	Id User
total	<i>Integer</i>	11		Total
ket	<i>Varchar</i>	20		Keterangan

(Sumber: Hasil Analisis Data, 2018)

6. Spesifikasi Tabel Detail Pemesanan Bahan Baku

Nama Tabel : Detail_Pbb

Fungsi : Untuk menyimpan data detail rencana produksi

Tipe : File data Transaksi

Tabel V.15 Tabel Detail Pemesanan Bahan Baku

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
no	<i>Integer</i>	11	<i>Primary key</i>	Nomor Pbb
id_pbb	<i>Varchar</i>	30	<i>Foreign Key</i>	Id Pbb
Id_kendaraan	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Kendaraan
jml	<i>Integer</i>	11		Total
time	<i>Timestamp</i>			Waktu

(Sumber: Hasil Analisis Data, 2018)

7. Spesifikasi Tabel Barang Pesan

Nama Tabel : barang_pesanan

Fungsi : Untuk menyimpan data barang pesanan

Tipe : File data Transaksi

Tabel V.16 Tabel Data Barang Pesan

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
id_pesanan	<i>Varchar</i>	16	<i>Primary key</i>	Id Pesan
tgl_pesanan	<i>Date</i>			Tanggal Pesan

Tabel V.16 Tabel Data Barang Pesan (Lanjutan)

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
id_barang	<i>Varchar</i>	16		Id Barang
Status_barang_pesan	<i>Varchar</i>	15		Status Barang Pesan
time	<i>Timestamp</i>			Waktu

(Sumber: Hasil Analisis Data, 2018)

8. Spesifikasi Tabel Detail Barang Pesan

Nama Tabel : detail_pesan

Fungsi : Untuk menyimpan data detail pesan barang

Tipe : File data Transaksi

Tabel V.17 Tabel Detail Barang Pesan

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
no	<i>Integer</i>	11	<i>Primary key</i>	Nomor
Id_pesan	<i>Varchar</i>		<i>Foreign Key</i>	Id Pesan
Id_barang	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Barang
Stok_awal	<i>Integer</i>	11		Stok Awal
Jml_pesan	<i>Integer</i>	11		Jumlah Pesan
time	<i>Timestamp</i>			Waktu

(Sumber: Hasil Analisis Data, 2018)

9. Spesifikasi Tabel Barang Masuk

Nama Tabel : Barang_Masuk

Fungsi : Untuk menyimpan data barang masuk

Tipe : File data Transaksi

Tabel V.18 Tabel Barang Masuk

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
Id_masuk	<i>Varchar</i>	16	<i>Primary key</i>	Id Masuk
Tgl_masuk	<i>Date</i>			Tanggal Masuk
Id_user	<i>Varchar</i>	16	<i>Foreign Key</i>	Id User

Tabel V.18 Tabel Barang Masuk (Lanjutan)

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
Id_pesan	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Pesan
time	<i>Timestamp</i>			Waktu

(Sumber: Hasil Analisis Data, 2018)

10. Spesifikasi Tabel Detail Barang Masuk

Nama Tabel : Detail Barang_Masuk

Fungsi : Untuk menyimpan data detail barang masuk

Tipe : File data Transaksi

Tabel V.19 Tabel Detail Barang Masuk

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
no	<i>Integer</i>	11	<i>Primary key</i>	Nomor
Id_masuk	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Masuk Barang
Id_barang	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Barang
Stok_awal	<i>Integer</i>	11		Stok Awal Barang
Jml_masuk	<i>Integer</i>	11		Jumlah Barang Masuk
<i>Time</i>	<i>Timestamp</i>			Waktu

(Sumber: Hasil Analisis Data, 2018)

11. Spesifikasi Tabel Barang Retur

Nama Tabel : barang_retur

Fungsi : Untuk menyimpan data barang retur

Tipe : *Line Damage Card*

Tabel V.20 Tabel Data Barang Retur

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
id_retur	<i>Varchar</i>	16	<i>Primary key</i>	Kode Part
tgl_retur	<i>Date</i>			Nama Part
id_user	<i>Varchar</i>	16		Tipe Part
time	<i>Timestamp</i>	15		Jenis Part
Acc1	<i>Varchar</i>	10		Stok Part
Acc2	<i>Varchar</i>	10		Minimal Stok

(Sumber: Hasil Analisis Data, 2018)

12. Spesifikasi Tabel Detail Barang Retur

Nama Tabel : detail_barang_retur

Fungsi : Untuk menyimpan data detail barang retur

Tipe : *Line Damage Card*

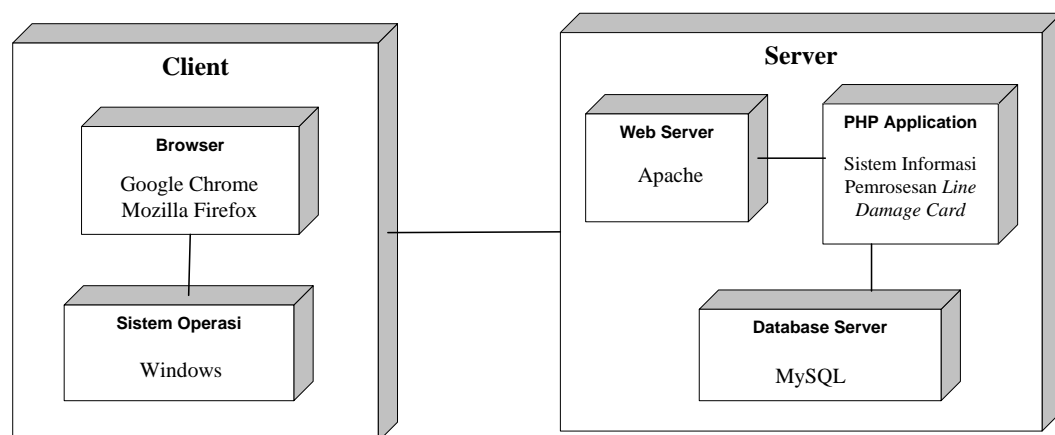
Tabel V.21 Tabel Detail Barang Retur

Nama Field	Tipe	Panjang	Keterangan	Deskripsi
no	<i>Integer</i>	11	<i>Primary key</i>	Nomor Retur
Id_retur	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Retur
Id_barang	<i>Varchar</i>	16	<i>Foreign Key</i>	Id Barang
Stok_awal	<i>Integer</i>	11		Stok Awal Barang
Jml_retur	<i>Integer</i>	11		Jumlah Retur
keterangan	<i>Varchar</i>	50		Keterangan
penanggulangan	<i>Varchar</i>	100		Penanggulangan
time	<i>Timestamp</i>			Waktu

(Sumber: Hasil Analisis Data, 2018)

5.5 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Seperti yang dapat dilihat pada gambar V.15 sebagai berikut:

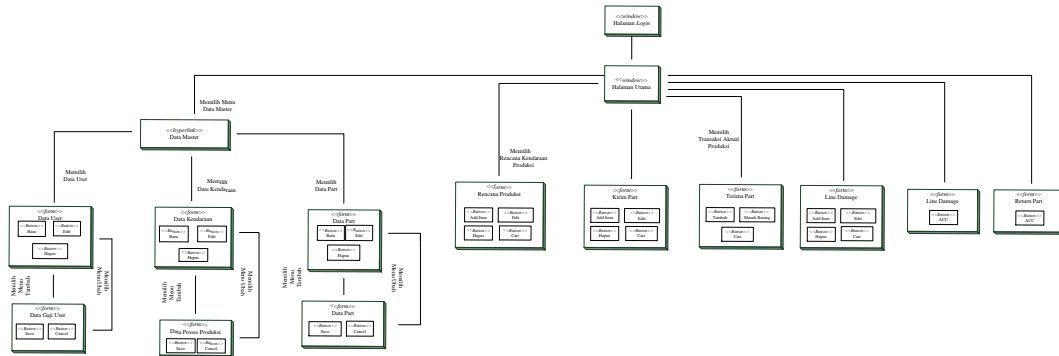


Gambar V.22 Deployment Diagram yang Diusulkan

(Sumber: Hasil Pengolahan Data, 2017)

5.6 Windows Navigation Diagram (WND)

Dengan *Windows Navigation Diagram* kita dapat dengan mudah melihat skema sistem, sehingga akan memudahkan menganalisa sistem. Berikut ini merupakan contoh *Windows Navigation Diagram* usulan pada sistem informasi pemrosesan *line damage card*. Dapat dilihat pada Gambar V.23



Gambar V.23 WND yang Diusulkan

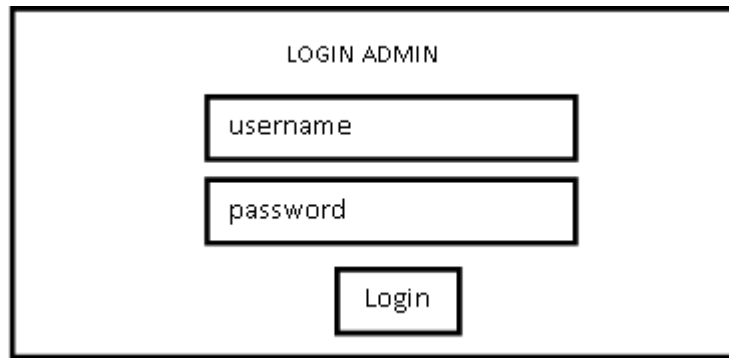
(Sumber: Hasil Pengolahan Data, 2018)

5.7 Perancangan Interface Sistem Informasi Pemrosesan Line Damage Card

Perancangan *interface* merupakan tahapan untuk membuat tampilan atau *design* dari Sistem Informasi Pemrosesan *Line Damage Card*. Rancangan tampilan yang dibuat meliputi beberapa rancangan *input* (berupa *form input*) dan rancangan *output* (berupa laporan). Rancangan *interface* dalam Sistem Informasi Pemrosesan *Line Damage Card* tersebut dapat dilihat berikut ini:

1. Form Login

Form Login digunakan untuk membedakan hak akses pengguna (*user*). Melalui *Form Login* ini pengguna yang boleh masuk sistem adalah pengguna yang mengetahui *username* dan *password* atau pengguna yang memiliki wewenang untuk menggunakan sistem yaitu Sistem Informasi Pengiriman *Part*.



LOGIN ADMIN

Gambar V.24 Rancangan *Form Login*

(Sumber: Hasil Pengolahan Data, 2018)

Keterangan:


Username : Untuk Menginputkan *username*.

Password : Untuk menginputkan *password*.

Button Login : Digunakan untuk proses masuk ke dalam menu utama.

2. 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 PEMROSESAN LINE DAMAGE CARD
<div style="text-align: center; margin-bottom: 5px;"><input style="width: 60px; height: 15px;" type="button" value="HOME"/></div> <div style="text-align: center; margin-bottom: 5px;"><input style="width: 60px; height: 15px;" type="button" value="DATA MASTER"/></div> <div style="text-align: center; margin-bottom: 5px;"><input style="width: 60px; height: 15px;" type="button" value="DATA BARANG"/></div> <div style="text-align: center; margin-bottom: 5px;"><input style="width: 60px; height: 15px;" type="button" value="USER"/></div> <div style="text-align: center;"><input style="width: 60px; height: 15px;" type="button" value="LOGOUT"/></div>	SELAMAT DATANG
FOOTER	

Gambar V.25 Rancangan Menu Utama

(Sumber: Hasil Pengolahan Data, 2018)

3. Data Master

Data master ini terdiri dari Master *User*, Master *Barang*. Berikut penjelasan dari Data Master:

- 1) Master Data *User* merupakan *form* master *user* dan digunakan untuk menginput data master tentang hak akses dalam sistem informasi pengiriman part.

Header									
<p>ID User <input style="width: 100%;" type="text"/></p> <p>User Name <input style="width: 100%;" type="text"/></p> <p>Password <input style="width: 100%;" type="text"/></p> <p>Akses Level <input style="width: 100%;" type="text"/></p> <p style="text-align: center;"> <input type="button" value="Tambah"/> <input type="button" value="Ubah"/> <input type="button" value="Hapus"/> </p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">ID User</th> <th style="width: 25%;">User Name</th> <th style="width: 25%;">Password</th> <th style="width: 25%;">Akses Level</th> </tr> </thead> <tbody> <tr> <td style="height: 150px;"> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	ID User	User Name	Password	Akses Level				
ID User	User Name	Password	Akses Level						
Footer									

Gambar V.26 Rancangan Master *User*

(Sumber: Hasil Pengolahan Data, 2018)

- 2) Master Data *Barang*

Form master *barang* digunakan untuk menginput data master tentang *barang* yang rusak saat di produksi.

Header											
Kode Barang	<input type="text"/>										
Nama Barang	<input type="text"/>										
Tipe Barang	<input type="text"/>										
Jenis Barang	<input type="text"/>										
Stok Barang	<input type="text"/>										
<input type="button" value="Tambah"/>	<input type="button" value="Ubah"/> <input type="button" value="Hapus"/>										
<table border="1"> <thead> <tr> <th>Kode Barang</th> <th>Nama Barang</th> <th>Tipe Barang</th> <th>Jenis Barang</th> <th>Stok Barang</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>		Kode Barang	Nama Barang	Tipe Barang	Jenis Barang	Stok Barang					
Kode Barang	Nama Barang	Tipe Barang	Jenis Barang	Stok Barang							
Footer											

Gambar V.27 Rancangan Master Barang

(Sumber: Hasil Pengolahan Data, 2018)

4. Menu *Line Damage Card*

Menu *Line Damage Card* berfungsi untuk mencatat data *part* rusak oleh *Foreman* Produksi.

Header	
LINE DAMAGE CARD	
<input type="text" value="Tanggal"/>	
<input type="text" value="Kode Part"/>	
<input type="text" value="Nama Part"/>	
<input type="text" value="JUmlah"/>	
<input type="text" value="Penyebab"/>	
<input type="text" value="Penanggulangan"/>	
Footer	

Gambar V.28 Rancangan Menu *Form Line Damage Card*

(Sumber: Hasil Pengolahan Data, 2018)

5. Persetujuan *Line Damage Card*

Menu Menyetujui dan Tidak Menyetujui *Line Damage Card*

berfungsi untuk meminta persetujuan mengenai part rusak dari Kepala Bagian Produksi, Manager Produksi, dan Controlling.

Header	
USER	LINE DAMAGE CARD
Tanggal	
Kode Part	
Nama Part	
JUmlah	
Penyebab	
Penanggulangan	
Setuju	Tidak Setuju
Footer	

Gambar V.29 Rancangan Persetujuan *Line Damage Card*

(Sumber: Hasil Pengolahan Data, 2018)

6. *Form* Rencana Produksi

Form Rencana Produksi berfungsi untuk memberikan informasi mengenai data Rencana Produksi yang dibuat oleh Bagian PPC.

PT Krama Yudha Ratu Motor

Form Permohonan Rencana Produksi
 No Nota
 Tgl Nota
 Tgl Transaksi

NO	ID KENDARAAN	NAMA KENDARAAN	JENIS KENDARAAN	TYPE	KUALITAS	JUMLAH

Gambar V.30 Rancangan *Form* Rencana Produksi PPC
 (Sumber: Hasil Pengolahan Data, 2018)

7. *Form Pengiriman Part*

Form Pengiriman Part berfungsi untuk memberikan informasi mengenai data *part* yang akan dikirim oleh bagian *Part Control* ke bagian Produksi, laporan ini dibuat oleh Bagian *Part Control*.

PT Krama Yudha Ratu Motor

Pengiriman Part Produksi
 NO.FAKTUR :
 TGL.FAKTUR :
 TGL.TRANSAKSI :
 PETUGAS YANG ENTRY
 ID USER :
 NAMA USER :

RINCIAN PART PRODUKSI INI YANG DIKIRIM

NO	ID PART	NAMA PART	JENIS PART	JUMLAH

Gambar V.31 Rancangan *Form* Pengiriman *Part* Produksi
 (Sumber: Hasil Pengolahan Data, 2018)

8. *Form Penerimaan Part*

Form Penerimaan Part berfungsi untuk memberikan informasi mengenai data *part* yang masuk ke bagian produksi untuk diproduksi oleh bagian Produksi, laporan ini dibuat oleh *Foreman* Produksi.

PT Krama Yudha Ratu Motor				
FAKTUR BARANG MASUK				
No Nota :				
Tgl Nota :				
PETUGAS YANG MENCATAT TRANSAKSI				
ID USER :				
NAMA USER :				
NO	ID PART	NAMA PART	JENIS PART	QTY

Gambar V.32 Rancangan *Form Penerimaan Part* Produksi
(Sumber: Hasil Pengolahan Data, 2018)

9. *Form Line Damage Card*

Form Line Damage Card berfungsi untuk memberikan informasi mengenai data *part* rusak/ hilang, form ini dibuat oleh *Foreman* Produksi.

PT Krama Yudha Ratu Motor

Nota Transaksi Retur Kerusakan Part
No Retur
Tanggal

TABEL DATA RINCIAN PART YANG DIRETUR

NO	ID PART	NAMA PART	ID JENIS	NAMA JENIS	ACC KABAG PRODUKSI	ACC MANAGER PRODUKSI	MASALAH	PENANGGULANGAN	QTY

Gambar V.33 Rancangan *Form Line Damage Card*
(Sumber: Hasil Pengolahan Data, 2018)

5.8 Implementasi Sistem

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

1. Analisis Kebutuhan *Software*
 - a. Sistem Operasi : *Microsoft Windows 10*
 - b. *Database Server* : *MySQL versi 5.5.25*
 - c. Bahasa Pemrograman : *PHP 5.4.4*
 - d. *Web Browser* : *Mozilla Firefox, Google Chrome*

2. Analisis Kebutuhan *Hardware*
 - a. *Processor* : Intel(R) Core i5
 - b. RAM : 8 GB
 - c. *Harddisk* : 750 GB
 - d. *Mouse, Keyboard, Monitor* sebagai peralatan antar muka.
 - e. *Printer* sebagai media pencetakan dokumen.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengumpulan, pengolahan dan analisis data yang telah dilakukan dalam penelitian ini dapat disimpulkan bahwa:

1. Diharapkan Sistem yang telah dibuat dapat membantu penyimpanan dan pencarian data historis *line damage card*.
2. Diharapkan sistem yang telah dibuat dapat membantu dan mempercepat proses pengisian data pada *form line damage card*
3. Merancang bangun sistem informasi yang berhubungan dengan *Line Damage Card* untuk mempersingkat waktu dalam meminta persetujuan pada *form line damage card*.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem informasi persediaan dan pengiriman part ini selanjutnya adalah sebagai berikut:

1. Dalam penerapan sistem baru terhadap sistem lama, sebaiknya dilakukan secara bertahap dan diperlukan sosialisasi penggunaan sistem ini kepada bagian yang terkait.
2. Pemeliharaan terhadap sistem yang telah dibuat agar tetap berjalan dengan baik wajib dilakukan.
3. Fitur-fitur yang lebih mendukung terhadap sistem yang baru seperti fitur pembuatan *Line Damage Card* dan fitur lainnya yang saling terhubung.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	
LEMBAR PENGESAHAN	
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN DOSEN PEMBIMBING	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Pokok Permasalahan	2
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	3
1.5. Manfaat Tugas Akhir	4
1.6. Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
2.1. Sistem	7
2.1.1 Karakteristik Sistem	7
2.1.2 Klasifikasi Sistem	10
2.2. Informasi	11
2.2.1 Kualitas Informasi	11
2.2.2 Siklus Informasi	11
2.3. Sistem Informasi	12
2.4. Metodologi Pengembangan Sistem	14
2.5. Pengertian Produksi	18

2.5.1	Pengertian Proses Produksi	19
2.5.2	Sifat Proses Produksi	20
2.5.2	<i>Line Damage Card</i>	21
2.6.	Bahan Baku	21
2.6.1	Persediaan Bahan Baku	22
2.6.2	Fungsi Persediaan Bahan Baku	22
2.6.3	Tujuan Persediaan Bahan Baku.....	23
2.7.	Diagram Alir (<i>Flowchart</i>)	23
2.8.	<i>Unified Modeling Language (UML)</i>	24
2.9.	Kelompok Diagram Pada UML	25
2.9.1	<i>Use Case Diagram</i>	26
2.9.2	<i>Class Diagram</i>	27
2.9.3	<i>Sequence Diagram</i>	28
2.9.4	<i>Activity Diagram</i>	29
2.9.8	<i>Deployment Diagram</i>	30
2.10.	ERD (<i>Entity Relationship Diagram</i>)	31
2.11.	Kamus Data	32
2.12.	<i>Windows Navigation Diagram (WND)</i>	32
2.13.	XAMPP <i>Local Server</i>	33
2.14.	<i>Hypertext Preprocessor (PHP)</i>	34
2.15.	<i>My Structure Query Language (MySQL)</i>	36
2.15.	<i>Adobe Dreamweaver</i>	43
BAB III	METODOLOGI PENELITIAN	45
3.1.	Metodologi Penelitian	45
3.1.1	Jenis dan Sumber Data	45
3.1.2	Metode Pengumpulan Data	45
3.2.	Metode Pengembangan Sistem	47
3.3.	Kerangka Penelitian	47
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	51
4.1.	Hubungan Kerjasama PT Krama Yudha Ratu Motor	51
4.2.	<i>Part Control Local Organisation</i>	52

4.3.	Jenis – Jenis Produk	57
4.3.1	Contoh Jenis <i>Part</i>	58
4.4.	Sistem Informasi Pemrosesan <i>Line Damage Card</i> Yang Sedang Berjalan	59
4.4.1	Dokumen yang Terkait dalam Pemrosesan <i>Line Damage Card</i>	63
4.4.2	Sistem Pemrosesan <i>Line Damage Card</i> dengan <i>Unified Modelling Language (UML)</i>	67
BAB V	ANALISIS DAN PEMBAHASAN	70
5.1.	Analisis Kebutuhan Sistem	70
5.2.	Perancangan Sistem	72
5.2.1	<i>Flowmap</i> Sistem Informasi Pemrosesan <i>Line Damage Card</i> yang Diusulkan	72
5.2.2	<i>Use Case Diagram</i> Sistem Informasi Pemrosesan <i>Line</i> <i>Damage Card</i> yang Diusulkan	75
5.2.3	<i>Use Case Description</i> Sistem Informasi Pemrosesan <i>Line Damage Card</i> yang Diusulkan	76
5.2.4	<i>Activity Diagram</i> Yang Diusulkan	81
5.2.5	<i>Sequence Diagram</i> Yang Diusulkan	91
5.2.6	<i>Class Diagram</i> Yang Diusulkan	99
5.3	<i>Entity Relationship Diagram (ERD)</i>	99
5.4	Kamus Data	101
5.5	<i>Deployment Diagram</i>	106
5.6	<i>Windows Navigation Diagram (WND)</i>	106
5.7	Perancangan <i>Interface</i> Sistem Informasi Persediaan dan Pengiriman <i>Part</i>	107
5.8	Implementasi Sistem	114
BAB VI	KESIMPULAN DAN SARAN	115
6.1.	Kesimpulan	115
6.2.	Saran	115

DAFTAR PUSTAKA	116
-----------------------------	------------

LAMPIRAN

DAFTAR GAMBAR

		Halaman
Gambar II.1	Karakteristik Sistem	9
Gambar II.2	Siklus Informasi	12
Gambar II.3	Blok Sistem Informasi Yang Berinteraksi	12
Gambar II.4	Mekanisme Pengembangan Sistem Dengan <i>Prototype</i>	16
Gambar II.5	<i>Evolutionary Prototype Model</i>	18
Gambar II.6	Klasifikasi Diagram UML.....	25
Gambar II.7	Contoh <i>Windows Navigation Diagram</i>	33
Gambar II.8	Siklus PHP dalam <i>Web Server</i>	35
Gambar II.9	Skema <i>Prototype Server MySQL</i>	42
Gambar II.10	Ruang Kerja <i>Adobe Dreamweaver</i>	43
Gambar III.1	Kerangka Penelitian	48
Gambar IV.1	Hubungan Kerja PT Krama Yudha Ratu Motor di dalam dan di luar Krama Yudha Mitsubishi Group.....	52
Gambar IV.2	Struktur Organisasi <i>Part Control</i> Lokal	53
Gambar IV.3	Contoh <i>Part</i> Jenis Kendaraan CJM.....	58
Gambar IV.4	Contoh <i>Part</i> Jenis Kendaraan SL.....	58
Gambar IV.5	Contoh <i>Part</i> Jenis Kendaraan <i>Truck Diesel</i>	59
Gambar IV.6	Contoh <i>Part</i> Jenis Kendaraan Fuso.....	59
Gambar IV.7	<i>Flowmap</i> Pemrosesan <i>Line Damage Card</i> yang Sedang Berjalan	62
Gambar IV.8	Rencana Produksi PC/PPC.....	64
Gambar IV.9	<i>Line Damage Card</i>	66
Gambar IV.10	<i>Use Case Diagram</i> Persediaan dan Pengiriman <i>Part</i>	67
Gambar V.1	<i>Flowmap</i> Sistem Informasi Pemrosesan <i>Line Damage Card</i> yang Diusulkan.....	74
Gambar V.2	<i>Use Case</i> Sistem Informasi Pemrosesan <i>Line Damage Card</i> yang Diusulkan.....	75

Gambar V.3	<i>Activity Diagram</i> Proses Login	82
Gambar V.4	<i>Activity Diagram</i> Mengelola Master User.....	83
Gambar V.5	<i>Activity Diagram</i> Mengelola Master Barang.....	84
Gambar V.6	<i>Activity Diagram</i> Menginput Data Rencana Produksi	85
Gambar V.7	<i>Activity Diagram</i> Menginput Data Kirim Part	86
Gambar V.8	<i>Activity Diagram</i> Menginput Data Part Masuk	87
Gambar V.9	<i>Activity Diagram</i> Menginput Line Damage Card	88
Gambar V.10	<i>Activity Diagram</i> Persetujuan Line Damage Card Oleh Kepala Bagian Produksi	89
Gambar V.11	<i>Activity Diagram</i> Persetujuan Line Damage Card Oleh Manager Produksi	90
Gambar V.12	<i>Sequence Diagram</i> Login	91
Gambar V.13	<i>Sequence Diagram</i> Data User.....	92
Gambar V.14	<i>Sequence Diagram</i> Data Barang	93
Gambar V.15	<i>Sequence Diagram</i> Rencana Produksi	94
Gambar V.16	<i>Sequence Diagram</i> Data Kirim Part	94
Gambar V.17	<i>Sequence Diagram</i> Data Part Masuk	95
Gambar V.18	<i>Sequence Diagram</i> Menginput Line Damage Card	96
Gambar V.19	<i>Sequence Diagram</i> Persetujuan Line Damage Card Oleh Kepala Bagian Produksi.....	97
Gambar V.20	<i>Sequence Diagram</i> Persetujuan Line Damage Card Oleh Manager Produksi.....	98
Gambar V.21	<i>Class Diagram</i> Yang Diusulkan.....	99
Gambar V.22	<i>Entity Relationship Diagram</i> yang Diusulkan	100
Gambar V.22	<i>Deployment Diagram</i> yang Diusulkan	106
Gambar V.23	WND yang Diusulkan	107
Gambar V.24	Rancangan Form Login	108
Gambar V.25	Rancangan Menu Utama	108
Gambar V.26	Rancangan Master User	109
Gambar V.27	Rancangan Master Barang	110
Gambar V.28	Rancangan Menu Form Line Damage Card	110

Gambar V.29	Rancangan Persetujuan <i>Line Damage Card</i>	111
Gambar V.30	Rancangan <i>Form</i> Rencana Produksi PPC	112
Gambar V.31	Rancangan <i>Form</i> Pengiriman <i>Part</i> Produksi	112
Gambar V.32	Rancangan <i>Form</i> Penerimaan <i>part</i> Produksi	113
Gambar V.33	Rancangan <i>Form Line Damage Card</i>	114

DAFTAR TABEL

		Halaman
Tabel II.1	Simbol-simbol <i>Flowchart</i>	23
Tabel II.2	Simbol-simbol <i>Use Case Diagram</i>	26
Tabel II.3	Simbol-Simbol <i>Class Diagram</i>	28
Tabel II.4	Simbol-Simbol <i>Sequence Diagram</i>	28
Tabel II.5	Simbol-Simbol <i>Activity Diagram</i>	30
Tabel II.6	Simbol-Simbol <i>Deployment Diagram</i>	30
Tabel II.7	Simbol-Simbol ERD	31
Tabel II.8	Contoh Kamus Data Untuk Tabel Pemasok	32
Tabel II.9	Daftar Tipe Data <i>Numeric</i>	38
Tabel II.11	Daftar Tipe Data Untuk Penanggalan dan Waktu	39
Tabel II.12	Tipe Data <i>String</i>	40
Tabel II.13	Tipe Data <i>String</i> Yang Sering Digunakan	41
Tabel IV.1	Jenis – Jenis Produk PT Krama Yudha Ratu Motor	57
Tabel IV.2	Definisi <i>Actor</i> dalam Pemrosesan <i>Line Damage Card</i>	68
Tabel IV.3	Definisi <i>Use Case</i> dalam Pemrosesan <i>Line Damage Card</i>	68
Tabel V.1	Kebutuhan Fungsional Sistem	71
Tabel V.2	<i>Use Case Description Login</i>	76
Tabel V.3	<i>Use Case Description</i> Menginput Data Rencana Produksi	77
Tabel V.4	<i>Use Case Description</i> Menginput Data <i>Part</i> Kirim.....	77
Tabel V.5	<i>Use Case Description</i> Menginput Data <i>Part</i> Masuk	78
Tabel V.6	<i>Use Case Description</i> Menginput <i>Master User</i>	79
Tabel V.7	<i>Use Case Description</i> Menginput <i>Master Barang</i>	80
Tabel V.8	<i>Use Case Description</i> Membuat <i>Line Damage Card</i>	80
Tabel V.9	<i>Use Case Description</i> Persetujuan <i>Line Damage Card</i>	81
Tabel V.10	Tabel Data <i>User</i>	101
Tabel V.11	Tabel Data Barang	101
Tabel V.12	Tabel Jenis Barang	102

Tabel V.13	Tabel Data Kendaraan.....	102
Tabel V.14	Tabel Data Pemesanan Bahan Baku	103
Tabel V.15	Tabel Detail Pemesanan Bahan Baku	103
Tabel V.16	Tabel Data Barang Pesan	103
Tabel V.17	Tabel Detail Barang Pesan	104
Tabel V.18	Tabel Barang Masuk	104
Tabel V.19	Tabel Detail Barang Masuk	105
Tabel V.20	Tabel Data Barang Retur.....	105
Tabel V.21	Tabel Detail Barang Retur	106

DAFTAR PUSTAKA

- Ahyari, Agus. 2011. *Perencanaan dan Pengendalian Produksi*. Yogyakarta: BPFE.
- Assauri, Sofjan. 2008. *Manajemen Produksi dan Operasi*. Jakarta: LPFEUI.
- Denis, et all. 2015. *System Analysis and Design With UML 2.0*, Wiley.
- Gasperz, Vincent. 2005. *Production Planning and Inventory Control Berdasarkan Sistem Terintegrasi MRP II dan JIT menuju Manufacturing 21 / Vincent Gasperz*. Jakarta, Gramedia Pustaka Utama
- Heizer, Jay dan Render, Berry. 2012. *Manajemen Operasi*. Edisi 7 Buku 1. Jakarta: Salemba Empat
- Herjanto, Eddy. 2003. *Manajemen Produksi dan Operasi*. Cetakan Ketiga, Jakarta PT Grasindo.
- Ishak. 2010. *Manajemen Produksi dan Operasi*. Yogyakarta: Graha Ilmu.
- Jeffrey A. Hoffer et.all, 2007. *Modern System Analysis and Design*, Edisi Keempat. India: *Pearson Education*.
- Jogiyanto, H. M. 2005. *Analisis dan Desain Sistem Informasi*. Andi Offset. Yogyakarta.
- McLeod, R. J, & Scholl. 2008. *Sistem Informasi Manajemen*. Edisi 10. Jakarta:
- Nafarin, M. (2007). *Penganggaran Perusahaan*. Salemba Empat. Jakarta.
- Rosa dan Shalahuddin, M. 2011. *Rekayasa Perangkat Lunak*. Bandung: Modula. Salemba Empat.

Rangkuti, Freddy 2007. *Manajemen Persediaan : Aplikasi di Bidang Bisnis*. Raja Grafindo. Jakarta.

Sutabri, Tata. 2010. *Analisa Sistem Informasi*. Yogyakarta: Andi Yogyakarta.

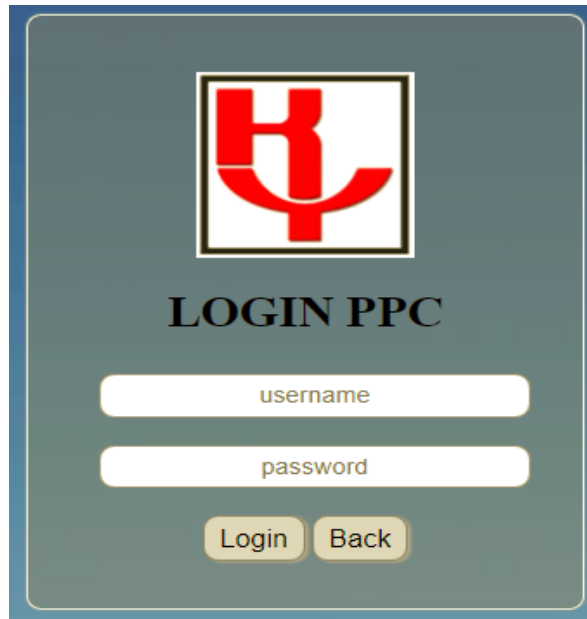
Wahana Komputer. 2010. *Panduan Belajar MySQL Database Server*. Media Kita. Jakarta Selatan.

Wicaksono, Yogi (2008). *Membangun Bisnis Online dg Mambo++ CD*. PT. Elex Media Komputindo, Jakarta

LAMPIRAN

KODE PROGRAM

1. Login



```
var action = $("#lg-form").attr('action');
var form_data = {
username: $("#username").val(),
password: $("#password").val(),
hakakses: $("#hakakses").val(),
is_ajax: 1
};

$.ajax({
type: "POST",
url: action,
data: form_data,
success: function(response)
{
if(response == "success")
$("#lg-form").slideUp('slow', function(){
$("#message").html('<p class="success">login anda
berhasil!</p><p>Redirecting....</p>');
window.location='index.php';
});
else
$("#message").html('<p class="error">Maaf Username dan Password yang
anda masukkan salah!</p>');
}
});
return false;
});
```

2. Data Kirim Part Produksi

The screenshot shows a web application interface for 'DATA KIRIM PART PRODUKSI'. At the top, there is a search bar with the text 'INPUT ID PART :', a search button labeled 'Cari', and the title 'DATA KIRIM PART PRODUKSI'. Below the search bar is a table with the following columns: ID PART, NAMA PART, ID JENIS, JENIS PART, STOK, JUMLAH, and PILIH. The table has a blue header and a yellow body. Below the table is a pagination bar showing 'Page 1 of 1' and 'Displaying 0 to 0 of 0 items'. There are also some navigation icons and a small text 'Selesai Menambahkan' at the bottom left.

```
<body>
<div id="cari">
  <form action="<?php echo $link_search; ?>" method="post"
  id="form_cari">
    INPUT ID PART :
    <input name="cari_id_brg" type="text" id="cari_id_brg" />
    <input type="submit" name="cari" id="cari2" value="Cari" />
  </form>
</div>
<div id="title" style="text-transform:uppercase">DATA KIRIM PART
PRODUKSI</div>
<div id="data">
<form action="save_item.php" method="post" name="form_item"
id="form_item">
  <table width="400" border="1" align="center" cellpadding="0"
cellspacing="0" id="tabel_item">
    <tr class="tr_item">
      <td align="center">ID PART</td>
      <td align="center">NAMA PART</td>
      <td align="center">ID JENIS</td>
      <td align="center">JENIS PART</td>
      <td align="center">STOK</td>
      <td align="center">JUMLAH</td>
      <td align="center">PILIH</td>
    </tr>
    <tr>
      <td><input name="id_barang" type="text" disabled="disabled"
id="id_barang" size="15" readonly="readonly" /></td>
      <td><input name="nm_barang" type="text" disabled="disabled"
id="nm_barang" size="30" readonly="readonly" /></td>
      <td><label for="id_jenis"></label>
```

```

        <input name="id_jenis" type="text" disabled="disabled"
id="id_jenis" size="8" readonly="readonly" /></td>
        <td><label for="nm_jenis"></label>
        <input name="nm_jenis" type="text" disabled="disabled"
id="nm_jenis" size="20" readonly="readonly" /></td>
        <td><input name="stok" type="text" disabled="disabled" id="stok"
size="5" readonly="readonly" /></td>
        <td><input name="jumlah" type="text" disabled="disabled"
id="jumlah" size="5" /></td>
        <td><input type="submit" name="tambah_item" id="tambah_item"
value="Add Item" /></td>
    </tr>
</table>
</form>
</div>
<div id="detail_request">
<table id="dg" title="DATA KIRIM PART PRODUKSI" class="easyui-
datagrid" style="height:250px"
        url="get_item.php"
        toolbar="#toolbar" pagination="true"
        rownumbers="true" fitColumns="true"
singleSelect="true">
    <thead>
        <tr>
            <th field="id_barang" width="50">ID PART</th>
            <th field="nm_barang" width="50">NAMA
PART</th>
            <th field="id_jenis" width="50">JENIS
PART</th>
            <th field="nm_jenis" width="50">NAMA
JENIS</th>
            <th field="jml" width="50">JUMLAH</th>
        </tr>
    </thead>
</table>
    <div id="toolbar">
        <a href="#" class="easyui-linkbutton" iconCls="icon-edit"
plain="true" onclick="editData()">Edit Data</a>
        <a href="#" class="easyui-linkbutton" iconCls="icon-
remove" plain="true" onclick="removeData()">Hapus Data</a>
    </div>
<div id="dlg" class="easyui-dialog"
style="width:400px;height:280px;padding:10px 20px"
        closed="true" buttons="#dlg-buttons">

    <div class="ftitle">EDIT JUMLAH KIRIM PART </div>

```

```

<form id="fm" method="post" novalidate>
<div class="fitem">
<label>Id Bahan baku:</label>
<input name="id_barang" disabled="disabled" class="easyui-validatebox"
>
</div>
<div class="fitem">
<label>Nama Bahan Baku:</label>
<input name="nm_barang" disabled="disabled" class="easyui-validatebox"
>
</div>
<div class="fitem">
<label>Jenis Bahan Baku:</label>
<input name="nm_jenis" disabled="disabled" class="easyui-
validatebox">
</div>
<div class="fitem">
<label>Jumlah : </label>
<input name="jml" class="easyui-numberbox" required="true" id="jml"
>
</div>
</form>
</div>
<div id="dlg-buttons">
<a href="#" class="easyui-linkbutton" iconCls="icon-ok"
onclick="saveData()">Save</a>
<a href="#" class="easyui-linkbutton" iconCls="icon-
cancel" onclick="javascript:$('#dlg').dialog('close')">Cancel</a>
</div>
<div id="proses">

<input type='checkbox' name='term' onClick="Javascript:disab(this,
1);"/>Selesai Menambahkan<br /><br />
<form action="proses_po.php?tb=<?php echo $tb; ?>" method="post"
id="form_proses" style="display:none">
<label class="label">Kode. RPP:<input name="id_trx" type="text"
id="id_trx" readonly="readonly" /></label>
<label class="label">Id User:<input name="id_user" type="text"
id="id_user" readonly="readonly" /></label><br />

<input type="submit" name="proses_request" id="proses_request"
value="Proses Transaksi" disabled="disabled" />
</form>
</div>
</div>
</body>

```

3. *Line Damage Card*



```

NO. RETUR : <?php echo $id_trx ?> <br />
TANGGAL: <?php echo date('d-m-Y'); ?> <br />
        <br />

</tr>
<tr>
    <td><br />
        TABEL RINCIAN DATA PART YANG DIRETUR<br />
        <br />
        <table width="730" border="1" align="center" cellpadding="0"
cellspacing="0">
            <tr>
                <td>NO</td>
                <td>ID PART</td>
                <td>NAMA PART</td>
                <td>ID JENIS</td>
                <td>JENIS PART</td>
                <td>ACC KABAG PRODUKSI</td>
                <td>ACC MANAGER PRODUKSI</td>
                <td>KETERANGAN</td>
                <td>PENANGGULANGAN</td>
                <td>QTY</td>

            </tr>

        </table>
    </td>
</tr>
<?php
        $i=0;
        $total_jumlah=0;
        $total_harga=0;
        while($rows=mysql_fetch_array($sql)){
            $id_barang=$rows['id_barang'];
    
```

```

        $nm_barang=$rows['nm_barang'];
        $jumlah=$rows['jml_retur'];
        $jenis=$rows['nm_jenis'];
        $id_jenis=$rows['id_jenis'];
        $acc1=$rows['acc1'];
        $acc2=$rows['acc2'];
        $keterangan=$rows['keterangan'];
        $penanggungan=$rows['penanggungan'];
        $total_jumlah=$total_jumlah+$jumlah;
        $i++;
    ?>
    <tr>
    <td><?php echo $i; ?></td>
    <td><?php echo $id_barang; ?></td>
    <td><?php echo $nm_barang; ?></td>
    <td><?php echo $id_jenis; ?></td>
    <td><?php echo $jenis; ?></td>
    <td><?php echo $acc1; ?></td>
    <td><?php echo $acc2; ?></td>
    <td><?php echo $keterangan; ?></td>
    <td><?php echo $penanggungan; ?></td>
    <td><?php echo $jumlah; ?></td>
    </tr>
    <?php } ?>
    <tr>
    <td colspan="8">&nbsp;</td>
    <td>TOTAL</td>
    <td><?php echo $total_jumlah; ?></td>
    </tr>
    </table></td>
</tr>
<tr>
<td>
    <a href="../../../pokaryawan.php" class="easyui-linkbutton"
    iconCls="icon-back" plain="true"></a>
    <a href="#" class="easyui-linkbutton" iconCls="icon-print" plain="true"
    onclick="print();"></a>
    </td>
</tr>
</table>
</body>
</html>

```