

No. Dok : 7042

D₃ 658.562

Asm
R.

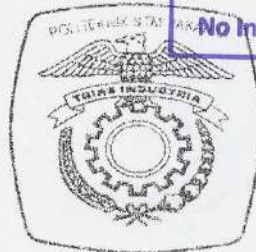
**RANCANG BANGUN SISTEM INFORMASI
PENGENDALIAN KUALITAS BAHAN BAKU
MENGUNAKAN PHP 7.2.9 DAN MYSQL 10.1.35
PADA PT GAYA MOTOR**

TUGAS AKHIR

Untuk Memenuhi Salah Satu Syarat Penyelesaian Program Sarjana Terapan
Program Studi Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta

**OLEH
GALIH ASMORO**

1312022



DATA BUKU PERPUSTAKAAN

Tgl Terima

21/10/22.

No Induk Buku

989/510/56/DA/22

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA
JAKARTA
2019**

SUMBANGAN ALUMNI

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

LEMBAR PENGESAHAN

Judul Tugas Akhir :

**RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN
KUALITAS BAHAN BAKU MENGGUNAKAN PHP 7.2.9 DAN MYSQL
10.1.35 PADA PT GAYA MOTOR**

Disusun oleh

Nama : Galih Asmoro

NIM : 1312022

Program Studi : Sistem Informasi Industri Otomotif

Telah diuji oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia pada hari Senin tanggal 09 September 2019.

Jakarta, 19 September 2019

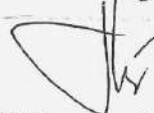
Dosen Pembimbing



Dedy Trisanto, S.Kom., MMSI

NIP. 197805052005021002

Ketua Penguji



Uli Hamida, ST, MT

NIP. 198103272005022001

Dosen Penguji



Fifi Lailasari H., S.Kom, M.Kes

NIP. 197310162005022001

Dosen Penguji



Ahmad Juniar, S.Kom, M.T.

NIP. 197906052006041002

**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA**

LEMBAR PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR :

**RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN
KUALITAS BAHAN BAKU MENGGUNAKAN PHP 7.2.9 DAN MYSQL
10.1.35 PADA PT GAYA MOTOK**

Disusun Oleh:

Nama : Galih Asmoro
NIM : 1312022
Program Studi : Sistem Informasi Industri Otomotif
Tanggal Seminar : 02 Agustus 2019
Tanggal Sidang : 09 September 2019
Tanggal Lulus : 19 September 2019

Jakarta, 19 September 2019

Dosen Pembimbing



Dedy Trisanto, S.Kom., MMSI

NIP. 197805052005021002

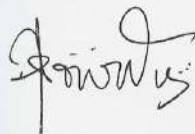


LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Galih Asmoro
NIM : 1312022
Judul TA : Rancang Bangun Sistem Informasi Pengendalian Kualitas Bahan Baku Menggunakan PHP 7.2.9 dan MySQL 10.1.35 Pada PT Gaya Motor
Pembimbing : Dedy Trisanto, SKom. MMSI

Tanggal	BAB	Keterangan	Paraf
14 Januari 2019	Cover, Bab I	Pengajuan	
17 Januari 2019	Cover, Bab I	Revisi	
22 Januari 2019	Bab II	Pengajuan, dan Revisi	
25 Januari 2019	Bab III	Revisi Kerangka Penelitian	
30 Januari 2019	Bab IV	Revisi <i>Flowmap</i> Berjalan	
06 Februari 2019	Bab IV	Revisi Usecase Berjalan	
06 Februari 2019	Bab IV	Revisi <i>Flowmap</i> usulan dan <i>Use Case</i> usulan	
14 Februari 2019	Bab V	Revisi <i>Flowmap</i> usulan dan <i>Use Case</i> usulan	
19 Februari 2019	Bab V	Revisi <i>Activity Diagram</i>	
28 Februari 2019	Bab V	Revisi <i>Class Diagram</i>	
19 Maret 2019	Bab V	Revisi <i>Sequence Diagram</i>	
25 Juni 2019	Bab I, III, IV, V, VI	Kesimpulan dan Saran	
01 Juli 2019	Demo Program	Revisi	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif



Noveriza Yuliasari, SSi, MT
NIP : 197811212009012003

Pembimbing



Dedy Trisanto, SKom. MMSI
NIP : 197805052005021002



POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Galih Asmoro

Nim : 1312022

Program Studi : Sistem Informasi Industri Otomotif

Dengan ini menyatakan bahwa karya Tugas Akhir yang saya buat dengan judul:

“RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN KUALITAS BAHAN BAKU MENGGUNAKAN PHP 7.2.9 DAN MYSQL 10.1.35 PADA PT GAYA MOTOR”. Merupakan dibuat dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survei lapangan, dosen pembimbing, melalui tanya jawab maupun asistensi serta buku-buku acuan yang tertera dalam referensi pada karya Tugas Akhir ini.

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

Jakarta, 10 Juli 2019

Yang Membuat Pernyataan,


Galih Asmoro

ABSTRAK

PT Gaya Motor merupakan perusahaan yang bergerak di bidang industri manufaktur yang berfokus dalam pembuatan perakitan mobil. PT Gaya Motor membutuhkan pengembangan sebuah sistem informasi pengendalian kualitas bahan baku mengenai pelaporan kualitas bahan baku. Sistem informasi pengendalian kualitas bahan baku yang ada pada PT Gaya Motor dalam pengolahan datanya masih kurang maksimal karena masih dilakukan secara manual dengan pencatatan tangan sehingga membutuhkan waktu yang lama dalam prosesnya. Selain itu, terjadi kesulitan dalam penyimpanan data *delivery note* dan monitor *problem* karena dokumen hanya disimpan dalam lemari arsip. Rancang bangun sistem informasi pengendalian kualitas bahan baku sangat diperlukan dalam perusahaan untuk melakukan proses pengolahan data kualitas bahan baku. Sistem informasi yang diusulkan akan mempermudah dan mengintegrasikan proses pengolahan data sehingga membantu bagian di dalam sistem untuk saling bertukar informasi dan mengambil keputusan dengan cepat. Pengembangan sistem informasi pengendalian kualitas bahan baku menggunakan metode *evolutionary prototype*. Pemodelan sistem menggunakan *unified modeling language* (UML). Perancangan sistem dengan *hierarchy plus input-process-output* (HIPO), *flowchart*. Pembuatan aplikasi sistem informasi pengendalian kualitas bahan baku ini menggunakan PHP 7.2.9 dan MySQL 10.1.35. Sistem informasi pengendalian kualitas bahan baku diharapkan dapat memudahkan perusahaan dalam proses pengolahan data dan memudahkan dalam menyimpan data pada bagian produksi. Untuk penerapan sistem baru, disarankan untuk melakukan sosialisasi kepada bagian terkait dan pemeliharaan aplikasi agar dapat berjalan dengan baik.

Kata Kunci : sistem informasi, pengendalian kualitas, pelaporan kualitas, *evolutionary prototype*.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah.SWT, yang atas berkat dan kasih-Nya sehingga penulis mampu menyelesaikan Tugas Akhir dengan judul **“Rancang Bangun Sistem Informasi Pengendalian Kualitas Bahan Baku Menggunakan PHP 7.2.9 dan MySQL 10.1.35 pada PT Gaya Motor”**.

Penulisan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian program studi D-IV Sistem Informasi pada Politeknik STMI Jakarta.

Terselesaikannya tugas akhir ini tidak terlepas dari bantuan moril maupun materil, bimbingan, dan petunjuk dari berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih yang sebesar-besarnya kepada Alm. Bapak Lilik Suwarno dan Ibu Wiwik Pudji Utami orang tua penulis dan juga kakak penulis yang selalu mendoakan. Selain itu, penulis juga berterimakasih kepada:

1. Bapak Dr. Mustofa, ST, MT. selaku Direktur Politeknik STMI Jakarta.
2. Ibu Noveriza Yuliasari, SSi, MT. selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
3. Bapak Dedy Trisanto, S.Kom, MMSI. Selaku dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
4. Bapak Gandung, dan Bapak Bambang selaku pembimbing di PT Gaya Motor serta seluruh pekerja khususnya *Production Department* yang telah membimbing dan memberikan informasi yang berguna kepada penulis dalam penyusunan Tugas Akhir.
5. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.
6. Serta semua pihak yang baik langsung maupun tidak langsung memberikan kritik, saran dan bantuan dalam pembuatan laporan ini yang tidak bisa penulis sebutkan satu per satu.

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

Jakarta, Juli 2019

Penulis,

DAFTAR ISI

Halaman

HALAMAN JUDUL	
LEMBAR PERSETUJUAN DOSEN PEMBIMBING	
LEMBAR BIMBINGAN DENGAN DOSEN PEMBIMBING	
LEMBAR PERNYATAAN KEASLIAN	
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Pokok Permasalahan	3
1.3. Tujuan Penelitian	3
1.4. Batasan Masalah	4
1.5. Manfaat Tugas Akhir	4
1.6. Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
2.1. Pengertian Sistem	7
2.1.1 Elemen Sistem	9
2.1.2 Karakteristik Sistem	10
2.2. Pengertian dan Konsep Dasar Sistem Informasi	11
2.2.1 Tipe Informasi	13
2.2.2 Karakteristik Informasi yang baik	14
2.3. Pengertian Sistem Informasi	15
2.3.1 Komponen Sistem Informasi	15
2.3.2 Klasifikasi dan Tujuan Sistem Informasi	16
2.4. Pengertian Pengendalian	17

2.5	Konsep Dasar Kualitas	18
2.6	Konsep Dasar Pengendalian Kualitas	18
2.7	Pengertian Bahan Baku	20
2.8	<i>System Development Life Cycle</i> (SDLC)	20
2.9	Model <i>Prototype</i>	22
2.9.1	<i>Prototype Evolutioner</i>	22
2.9.2	<i>Prototype Requirement</i>	23
2.10	<i>Flowchart</i>	23
2.11	<i>Unified Modelling Language</i> (UML)	25
2.11.1	<i>Use Case Diagram</i>	26
2.11.2	<i>Activity Diagram</i>	28
2.11.3	<i>Sequence Diagram</i>	28
2.11.4	<i>Class Diagram</i>	30
2.11.5	<i>Deployment Diagram</i>	33
2.11.6	<i>Component Diagram</i>	33
2.12	Kamus Data	34
2.13	HIPO (<i>Hierarchy plus Input-Process-Output</i>)	35
2.14	PHP (<i>Hypertext Pre-Processor</i>)	38
2.15	MySQL	39
2.15.1	Keunggulan MySQL	39
2.15.2	Tipe Data My SQL	41
BAB III	METODOLOGI PENELITIAN	42
3.1.	Metodologi Penelitian	42
3.2.	Jenis dan Sumber Data	42
3.3.	Metode Pengumpulan Data	43
3.4.	Metode Pengembangan Sistem	44
3.5.	Kerangka Penelitian	45
BAB IV	PENGUMPULAN DAN PENGOLAHAN DATA	48
4.1.	Sekilas Tentang Perusahaan	48
4.2.	Lokasi Pabrik	48
4.3.	Visi, Misi, dan Kebijakan Mutu Perusahaan	48

4.4.	Struktur Organisasi Perusahaan	49
4.5.	Struktur Organisasi Departemen Quality Control	49
4.6.	Dokumen Pengendalian Kualitas	50
4.7.	Prosedur Pengendalian Bahan Baku	51
BAB V	ANALISIS DAN PEMBAHASAN	53
5.1.	Analisis Kebutuhan Sistem	53
5.2.	Prosedur Sistem Informasi Pengendalian Kualitas Bahan Baku	54
5.3.	Analisis dan Perancangan Sistem Usulan	55
5.3.1.	<i>Use Case Diagram</i>	56
5.3.2.	<i>Activity Diagram</i>	62
5.3.3.	<i>Sequence Diagram</i>	66
5.3.4.	<i>Class Diagram</i>	70
5.3.5.	Kamus Data	71
5.3.6.	<i>Deployment Diagram</i>	74
5.4.	Analisis Desain Program	75
5.4.1.	HIPO (<i>Hierarchy plus Input-Process-Output</i>).....	75
5.4.2.	<i>Flowchart</i> Program	76
5.4.3.	Perancangan <i>Interface</i> Program	76
5.5.	Implementasi Sistem <i>Software</i> dan <i>Hardware</i>	81
BAB VI	KESIMPULAN DAN SARAN	83
6.1.	Kesimpulan	83
6.2.	Saran	83
DAFTAR PUSTAKA	84

DAFTAR GAMBAR

	Halaman
Gambar II.1 Karakteristik Sistem	11
Gambar II.2 Pengembangan <i>Prototype Evolutioner</i>	23
Gambar II.3 Pengembangan <i>Prototype Requirement</i>	23
Gambar II.4 UML Diagram	26
Gambar II.5 <i>Visual Table of Contents</i>	37
Gambar II.6 <i>Overview Diagram</i>	37
Gambar III.1 Kerangka Penelitian	47
Gambar IV.1 Struktur Organisasi PT Gaya Motor	49
Gambar IV.2 Struktur Organisasi Departemen Quality Control	49
Gambar IV.3 <i>Delivery Note</i>	51
Gambar IV.4 <i>Monitor Problem</i>	51
Gambar IV.5 <i>Flowmap</i> Proses Pengendalian Kualitas Bahan Baku yang berjalan	52
Gambar V.I <i>Flowmap</i> Sistem Informasi Pengendalian Kualitas Bahan Baku Usulan	55
Gambar V.2 <i>Use Case Diagram</i> Sistem Informasi Pengendalian Kualitas Bahan Baku Usulan	56
Gambar V.3 <i>Activity Diagram Login</i>	62
Gambar V.4 <i>Activity Diagram</i> Menginput Data Pesanan	63
Gambar V.5 <i>Activity Diagram</i> Menginput Data Stok Bahan Baku	63
Gambar V.6 <i>Activity Diagram</i> Menginput Surat Jalan	64
Gambar V.7 <i>Activity Diagram</i> Menginput <i>Monitor Problem</i>	64
Gambar V.8 <i>Activity Diagram</i> Mengupdate <i>Status Problem</i>	65
Gambar V.9 <i>Activity Diagram</i> Menginput <i>Receiving Check</i>	65
Gambar V.10 <i>Sequence Diagram Login</i>	66
Gambar V.11 <i>Sequence Diagram</i> Menginput Data Pesanan	67
Gambar V.12 <i>Sequence Diagram</i> Data Stok Bahan Baku	67

Gambar V.13	<i>Sequence Diagram</i> Membuat Data Surat Jalan	68
Gambar V.14	<i>Sequence Diagram</i> Menginput Monitor <i>Problem</i>	69
Gambar V.15	<i>Sequence Diagram</i> Update Status <i>Problem</i>	69
Gambar V.16	<i>Sequence Diagram</i> Receiving Check.....	70
Gambar V.17	<i>Class Diagram</i> Usulan	70
Gambar V.18	<i>Deployment Diagram</i>	74
Gambar V.19	HIPO Sistem Informasi Pengendalian Kualitas Bahan Baku	75
Gambar V.20	<i>Program Logic Flowchart</i> Aplikasi Sistem Informasi Pengendalian Kualitas	76
Gambar V.21	<i>Interface Form</i> Login	77
Gambar V.22	<i>Interface Form</i> Menu Utama.....	77
Gambar V.23	<i>Interface Form</i> Data Master User	78
Gambar V.24	<i>Interface Form</i> Input Data Stok Bahan Baku.....	78
Gambar V.25	<i>Interface Form</i> Transaksi Surat Jalan	79
Gambar V.26	<i>Interface Form</i> Monitor <i>Problem</i>	79
Gambar V.27	<i>Interface Form</i> Receiving Check	80
Gambar V.28	<i>Interface Form</i> Laporan Pengendalian Kualitas Bahan Baku..	81

DAFTAR TABEL

	Halaman
Tabel II.1 Simbol <i>Flow Direction</i>	24
Tabel II.2 Simbol Proses	24
Tabel II.3 Simbol <i>Input dan Output</i>	25
Tabel II.4 Simbol-Simbol <i>Use Case Diagram</i>	27
Tabel II.5 Simbol-Simbol <i>Activity Diagram</i>	28
Tabel II.6 Simbol-Simbol <i>Sequence Diagram</i>	29
Tabel II.7 Simbol-Simbol <i>Class Diagram</i>	31
Tabel II.8 Tipe <i>Multiplicity</i>	32
Tabel II.9 Simbol-Simbol <i>Deployment Diagram</i>	33
Tabel II.10 Simbol-Simbol <i>Component Diagram</i>	34
Tabel II.11 Contoh Kamus Data untuk Tabel Pemasok	35
Tabel II.12 Jenis Data pada MySQL	41
Tabel V.1 Kebutuhan Sistem informasi Pengendalian Kualitas	53
Tabel V.2 Definisi Aktor <i>Use Case Diagram</i> Sistem Usulan	57
Tabel V.3 Definisi <i>Use Case Diagram</i> Sistem Usulan	57
Tabel V.4 Skenario <i>Use Case Login</i>	58
Tabel V.5 Skenario <i>Use Case</i> Proses Menginput Data Pesanan	58
Tabel V.6 Skenario <i>Use Case</i> Proses data stok Bahan Baku	59
Tabel V.7 Skenario <i>Use Case</i> Proses Membuat Surat Jalan	59
Tabel V.8 Skenario <i>Use Case</i> Proses Monitor <i>Problem</i>	60
Tabel V.9 Skenario <i>Use Case Update Status Problem</i>	61
Tabel V.10 Skenario <i>Use Case</i> Proses menginput <i>Receiving Check</i>	61
Tabel V.11 Tabel <i>User</i>	71
Tabel V.12 Tabel Data Pesanan	71
Tabel V.13 Tabel Data Data Stok Bahan Baku	72
Tabel V.14 Tabel Data Surat Jalan	72
Tabel V.15 Tabel Data Monitor <i>Problem</i>	73

Tabel V.16	Tabel Data <i>Receiving Check</i>	73
------------	---	----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suatu perusahaan memiliki sistem yang dapat mengelola segala aktivitas guna menjalankan organisasinya, salah satunya adalah sistem informasi. Perusahaan memerlukan sistem informasi yang mampu mengkoordinasikan informasi internal dan eksternal perusahaan. Sebagai contoh internal perusahaan seperti bagian produksi memberikan informasi kerusakan pada proses penerimaan bahan baku, kemudian informasi tersebut disampaikan kepada pihak penyedia bahan baku oleh bagian gudang agar penyediaan bahan baku selanjutnya lebih memperhatikan kualitas bahan baku.

Kualitas suatu bahan baku sangat penting khususnya bagi penilaian pelanggan terhadap produk itu sendiri. Kualitas bahan baku yang baik pasti memiliki dampak positif yang dapat menarik pelanggan untuk menggunakan produk tersebut, sehingga produksi produk meningkat dan secara otomatis pendapatan perusahaan juga ikut meningkat.

Setiap perusahaan memiliki kebijakan sendiri atas standar kualitas suatu bahan baku dan menyediakan departemen tersendiri untuk mengelola kualitas produk, dalam hal ini yang dimaksud adalah Departemen produksi. Departemen produksi bertugas untuk mengelola dan memperhatikan kualitas suatu produk sehingga kualitas suatu produk memenuhi standar yang telah ditetapkan perusahaan. Dalam mengerjakan tugasnya departemen produksi memiliki sistem yang dapat memudahkan kinerja yaitu dengan cara memberikan laporan rutin per periode. Laporan ini kemudian akan dijadikan bahan untuk evaluasi standar kualitas selanjutnya. Laporan yang dibuat oleh departemen produksi terdiri dari laporan kerusakan per periode.

PT Gaya Motor merupakan salah satu perusahaan yang bergerak di bidang manufaktur dalam pembuatan perakitan mobil BMW. Perusahaan ini memproduksi berbagai jenis mobil BMW antaranya berupa tipe sedan dan jeep. Dalam mendukung kinerjanya, PT Gaya Motor telah menerapkan komitmen untuk menghasilkan produk yang bermutu sesuai dengan persyaratan dan harapan pelanggan, dengan cara melaksanakan dan memelihara sistem manajemen mutu. Salah satu cara untuk menciptakan produk dengan mutu yang berkualitas adalah dengan memperhatikan kualitas dari bahan baku yang digunakan untuk proses produksinya. Hal ini diwujudkan dengan tanda tangan komitmen dan membentuk departemen produksi yang bertugas untuk memantau standar kualitas bahan baku yang digunakan.

Departemen produksi memiliki sistem pengolahan data hasil inspeksi kualitas bahan baku yang terdiri dari *delivery note* dan monitor *problem*. Laporan jumlah kerusakan bahan baku masih dicatat secara manual, yaitu dengan penulisan tangan oleh bagian produksi dengan laporan dicatat pada monitor *problem*. Selanjutnya monitor *problem* disimpan oleh kepala produksi untuk dijadikan arsip tanpa dibuatkan sebuah aplikasi terkomputerisasi yang dapat membantu dan mempermudah kegiatan pengendalian kualitas. Hal ini sering menimbulkan terjadinya kehilangan dan kerusakan data. Sistem pengolahan data hasil inspeksi kualitas bahan baku ini juga belum memiliki laporan. Hal tersebut mengakibatkan acuan kerusakan bahan baku untuk periode berikutnya kurang akurat, karena hanya melihat dari monitor *problem* yang dijadikan arsip. Hal ini juga berdampak pada proses pencarian data hasil inspeksi kerusakan bahan baku pada suatu periode tertentu yang membutuhkan waktu lama, karena harus mencari satu per satu.

Untuk mengatasi masalah tersebut, diperlukan aplikasi untuk membantu perusahaan dalam mengatasi masalah tersebut. Adapun judul Tugas Akhir ini adalah “RANCANG BANGUN SISTEM INFORMASI PENGENDALIAN KUALITAS BAHAN BAKU MENGGUNAKAN PHP 7.2.9 DAN MYSQL 10.1.35 PADA PT GAYA MOTOR”.

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada bagian produksi pada PT Gaya Motor adalah sebagai berikut:

1. Pencatatan data kualitas bahan baku pada monitor *problem* masih berupa kertas. Hal tersebut dapat menimbulkan kerusakan dan kehilangan data hasil inspeksi atas kualitas bahan baku.
2. Belum adanya aplikasi terintegrasi dengan *database* sebagai media penyimpanan data, yang mengakibatkan terjadinya kesulitan mencari hasil inspeksi kualitas bahan baku karena masih berupa dokumen-dokumen berbentuk kertas.
3. Belum adanya sistem pelaporan data hasil inspeksi kualitas bahan baku, hal ini berdampak pada proses pencarian hasil inspeksi kualitas bahan baku pada suatu periode tertentu yang membutuhkan waktu lama, karena harus mencari satu persatu.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. Merancang sistem informasi pencatatan data kualitas bahan baku, proses pencatatan data monitor *problem* di-*input* secara komputerisasi.
2. Membuat sistem basis data pengendalian kualitas sehingga memudahkan dalam penyimpanan data, pencarian data dan pelaporan serta mencegah terjadinya kerusakan dan kehilangan data.
3. Merancang sistem informasi yang dapat menghasilkan laporan data kualitas bahan baku. Sehingga acuan data inspeksi kualitas bahan baku untuk periode selanjutnya lebih akurat serta memudahkan dalam pencarian data.

1.4 Batasan Masalah

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

1. Tempat penelitian adalah PT Gaya Motor pabrik Sunter, Jakarta.
2. Penelitian dilakukan pada bagian produksi selama satu bulan, dari bulan Oktober s.d November 2017.
3. Analisis dan penelitian yang dilakukan hanya sebatas menangani masalah pengelolaan data pengendalian kualitas pada bagian produksi yang menangani pengendalian kualitas bahan baku.
4. Pengendalian kualitas bahan baku hanya terbatas pada komponen-komponen kecil.

1.5 Manfaat Tugas Akhir

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

1. Bagi perusahaan
 - a. Hasil penelitian ini agar dapat diimplementasikan di perusahaan untuk membantu kinerja perusahaan dalam melakukan pengolahan pengendalian kualitas yang dilakukan.
 - b. Mendapatkan informasi yang lengkap terhadap setiap laporan proses pengendalian kualitas yang dilakukan setiap harinya.
2. Bagi mahasiswa
 - a. Memberikan kemampuan dalam mengaplikasikan teori secara jelas terhadap masalah yang diamati.
 - b. Memberikan wawasan dan pengalaman kepada mahasiswa dalam menganalisis suatu sistem dan diharapkan dapat memberikan suatu solusi permasalahan.
3. Bagi pihak lain

Hasil penelitian ini diharapkan dapat memberikan sumbangan pemikiran dan sebagai referensi bagi peneliti lain yang melakukan penelitian serupa.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis guna memberikan gambaran yang jelas mengenai isi dan pembahasan yang ada di dalamnya. Agar lebih mempermudah perumusan yang akan dibahas pada Tugas Akhir ini, maka diuraikan tahapan-tahapan dalam penyusunan Tugas Akhir ini sebagai berikut:

BAB I PENDAHULUAN

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

BAB II LANDASAN TEORI

Bab ini menjelaskan tentang berbagai teori yang diperoleh dari buku-buku literatur ataupun berbagai macam referensi. Teori-teori yang dipaparkan pada laporan ini adalah seputar pengertian sistem, jenis sistem, pengertian informasi, jenis-jenis informasi, konsep dasar pengendalian kualitas, pengertian bahan baku, metodologi pengembangan sistem, *flowmap*, *Unified Modelling Language* (UML), *Hypertext Preprocessor* (PHP), dan MYSQL.

BAB III METODOLOGI PENELITIAN

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

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini akan membahas mengenai data yang telah diperoleh, pengolahan data diagram alir, dan *use case* sistem pengendalian kualitas bahan baku yang sedang berjalan berdasarkan pengamatan selama melaksanakan penelitian di PT Gaya Motor.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari analisis kebutuhan sistem, memodelkan sistem dengan *Unified Modelling Language* (UML), kamus data, perancangan *flowchart* program dan tampilan layar, sampai kebutuhan *software* dan *hardware* yang diperlukan.

BAB V

PENUTUP

Dalam bab penutup ini dikemukakan kesimpulan-kesimpulan dari hasil penelitian dan saran-saran untuk pihak perusahaan dalam berbagai hal yang berhubungan dengan sistem informasi pengendalian kualitas pada bagian produksi.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Kata sistem berasal dari bahasa latin (*systēma*) dan bahasa Yunani (*sustēma*) yaitu suatu kesatuan yang terdiri dari komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Pada dasarnya manusia hidup di dunia penuh dengan sistem. Apa yang terlihat di sekeliling kita dalam kehidupan sehari-hari sebenarnya merupakan kumpulan dari sebuah sistem, dan sistem tersebut tersusun atau terorganisir dengan kompleks. Dikatakan kompleks karena dunia ini memang tersusun dari beberapa subsistem yang berbeda satu sama lain dan mereka berinteraksi pada tingkat tertentu. Sedangkan sistem dalam sudut pandang perusahaan atau organisasi adalah tempat sumber daya manusia yang berinteraksi untuk merealisasikan formulasi tujuan yang telah ditetapkan. Oleh karena itu dengan sendirinya, sebuah sistem dapat dikatakan sangat rumit.

Berikut ini beberapa pengertian sistem menurut para ahli:

1. Menurut Hall (2001), “Sistem adalah sekelompok dua atau lebih komponen-komponen yang saling berkaitan (*interrelated*) atau subelemen-subelemen yang bersatu untuk mencapai tujuan yang sama (*common purpose*) ”.
2. Menurut McLeod (2011), “Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan”.
3. Menurut Kristanto (2008), “Sistem merupakan jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu”.
4. Menurut Nugroho (2008), “Sistem adalah sesuatu yang memiliki bagian-bagian yang saling berinteraksi untuk mencapai tujuan tertentu melalui tiga tahapan yaitu input, proses, dan output”.

5. Menurut O'brien (2005), yang dialih bahasakan oleh Dewi Fitriansari dan Deni Arnos Kwary "Sistem adalah sekelompok komponen yang saling berhubungan, bekerjasama untuk mencapai tujuan bersama dengan menerima input serta menghasilkan output dalam transformasi yang teratur.

Suatu sistem dapat terdiri dari sistem-sistem bagian (*subsystem*). Sebagai contoh, sistem komputer dapat terdiri dari subsistem perangkat keras dan perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem yang lebih kecil lagi atau terdiri dari komponen-komponen. Subsistem perangkat keras dapat terdiri dari alat masukan, alat pemroses, alat keluaran, dan simpanan luar. Subsistem-subsistem saling berinteraksi dan saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai interaksi dari subsistem-subsistem sedemikian rupa sehingga dicapai suatu kesatuan yang terpadu atau terintegrasi (Jogiyanto, 1999).

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan ada yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedurnya mendefinisikan sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Kedua kelompok definisi tersebut adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya.

Pendekatan sistem yang merupakan kumpulan elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang lebih luas. Definisi ini lebih banyak diterima, karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem-subsistem atau sistem bagian. Misalnya sistem akuntansi pembelian, subsistem akuntansi penggajian, subsistem akuntansi biaya dan lain sebagainya

2.1.1 Elemen Sistem

Di dalam suatu sistem terdapat beberapa elemen atau unsur yang membentuk sebuah sistem, yaitu:

1. Tujuan (*Goal*)

Setiap sistem memiliki tujuan. Tujuan bisa lebih dari satu. Tujuan inilah yang menjadi pendorong atau motivasi yang mengarahkan ke arah mana sistem bergerak. Tujuan antara satu sistem dengan sistem yang lain berbeda termasuk berlaku juga dalam sistem informasi. Sekalipun dari setiap sistem berbeda, secara umum ada tiga macam tujuan utama (Hall, 2001), yaitu:

- a. Untuk mendukung fungsi kepengurusan manajemen.
- b. Untuk mendukung pengambilan keputusan manajemen.
- c. Untuk mendukung kegiatan operasi perusahaan.

Secara khusus, tujuan sistem informasi bergantung pada kegiatan yang ditangani. Namun, kecenderungan penggunaan sistem informasi lebih ditunjukan pada usaha menuju keunggulan kompetitif, yaitu mampu bersaing mengungguli pesaing.

2. Masukan (*Input*)

Masukan adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan untuk diproses.

3. Proses (*Process*)

Merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

4. Keluaran (*Output*)

Merupakan hasil dari pemrosesan yang bisa berupa suatu informasi, sasaran cetakan laporan dan lain-lain.

5. Mekanisme Pengendalian dan Umpan Balik

Mekanisme pengendalian dilakukan dengan menggunakan umpan balik (*feedback*) dari keluaran.

6. Batas (*Boundary*)

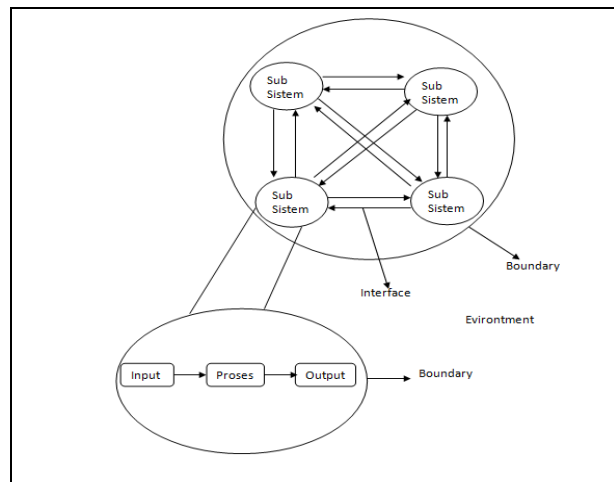
Batas adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkungan atau kemampuan sistem.

2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, antara lain (Mustakini 2005):

1. Suatu sistem mempunyai komponen-komponen sistem (*components*) atau sub-sistem-sub-sistem. Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerjasama dalam membentuk suatu kesatuan. Komponen sistem tersebut dapat berupa suatu bentuk sub-sistem.
2. Suatu sistem mempunyai batas sistem (*boundary*). Batasan sistem membatasi antara sistem yang satu dengan yang lainnya atau sistem dengan lingkungan luarnya.
3. Suatu sistem mempunyai lingkungan luar (*environment*). Lingkungan luar sistem adalah suatu bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut.
4. Suatu sistem mempunyai penghubung (*interface*). Penghubung sistem merupakan media yang menghubungkan sistem dengan sub-sistem yang lain, dengan demikian dapat terjadi suatu integrasi sistem yang membentuk suatu kesatuan.
5. Suatu sistem mempunyai tujuan (*goal*) atau sasaran sistem (*objective*). Sebuah sistem dikatakan berhasil apabila mengenai sasaran atau tujuannya, jika suatu sistem tidak mempunyai tujuan maka operasi sistem tidak akan ada gunanya.

Untuk lebih jelasnya, karakteristik sistem dapat dilihat pada gambar dibawah ini:



Gambar II.1. Karakteristik Sistem
(Sumber: Jogiyanto, 1999)

2.2 Pengertian dan Konsep Dasar Informasi

Kata informasi berasal dari kata Perancis kuno *informacion* tahun 1387 yang diambil dari bahasa latin *informationem* yang berarti “garis besar, konsep, ide”. Informasi merupakan kata benda dari *informare* yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan”. Nilai informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya (Jogiyanto, 2005).

Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* dan *cost benefit*. Terdapat beberapa pendapat mengenai pengertian informasi. Berikut ini beberapa pengertian informasi menurut para ahli:

1. Informasi adalah data yang dapat diolah yang lebih berguna dan berarti bagi yang menerimanya (Jogiyanto, 2005).
2. Menurut Edward (Eddy, 2005), informasi adalah analisis dan sintesis terhadap data, atau informasi adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai kebutuhan seseorang, manajer, staf, atau orang lain di dalam suatu organisasi atau perusahaan.

Jadi secara umum informasi adalah data yang sudah diolah menjadi suatu bentuk lain yang lebih berguna yaitu pengetahuan atau keterangan yang ditujukan bagi penerima dalam pengambilan keputusan, baik masa sekarang atau yang akan

datang. Untuk memperoleh informasi yang berguna, tindakan yang pertama adalah pengumpulan data, kemudian mengolahnya sehingga menjadi informasi. Dari kedua data tersebut informasi yang didapatkan lebih terarah dan penting karena telah dilalui berbagai tahap dalam pengolahannya diantaranya yaitu pengumpulan data, data apa yang terkumpul dan menemukan informasi yang diperlukan.

Kualitas dari suatu informasi (*quality of information*) tergantung dari tiga hal, yaitu (Jogiyanto, 2005):

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya.
2. Tepat pada waktunya, berarti informasi yang datang ke penerima tidak boleh terlambat. Informasi yang sudah usang tidak ada nilainya lagi. Karena informasi merupakan landasan dalam pengambilan keputusan.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

Darmawan (2007), menjelaskan enam ciri dari informasi yang bisa memberikan makna bagi pengguna, diantaranya:

1. Kuantitas Informasi (*Amount of Information*), dalam arti bahwa informasi yang diolah oleh suatu prosedur pengolahan informasi mampu memenuhi kebutuhan banyaknya informasi.
2. Kualitas Informasi (*Quality of Information*), dalam arti bahwa informasi yang diolah oleh sistem pengolahan tertentu mampu memenuhi kebutuhan kualitas informasi.
3. Informasi Aktual (*Recency of Information*), dalam arti bahwa informasi yang diolah oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi baru.
4. Informasi yang relevan atau sesuai (*Relevace of Information*), dalam arti bahwa informasi yang oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi.
5. Ketepatan Informasi (*Accuracy of Information*), dalam arti bahwa informasi yang oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi.

6. Kebenaran informasi (*Authenticity of Information*), dalam arti bahwa informasi yang dikelola oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi yang benar.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, melakukan keputusan berdasarkan informasi tersebut dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap kembali sebagai input, diproses kembali melalui suatu model dan seterusnya yang merupakan suatu siklus.

2.2.1 Tipe Informasi

Sistem informasi sekarang perannya tidak hanya sebagai pengumpulan data dan mengolahnya menjadi informasi berupa laporan-laporan keuangan saja, tetapi mempunyai peranan yang lebih penting di dalam menyediakan informasi bagi manajemen untuk fungsi-fungsi perencanaan, alokasi-alokasi sumber daya, pengukuran dan pengendalian. Laporan-laporan dari sistem informasi memberikan informasi kepada manajemen mengenai permasalahan-permasalahan yang terjadi di dalam organisasi untuk menjadi sebuah bukti yang berguna di dalam menentukan tindakan yang diambil.

Sistem informasi dapat menyediakan tiga macam tipe informasi, masing-masing mempunyai arti yang berbeda untuk tingkatan manajemen yang berbeda, yaitu:

1. Informasi Pengumpulan Data (*Scorekeeping of Information*), merupakan informasi yang berupa akumulasi atau pengumpulan data, informasi ini berguna bagi manajer bawah untuk mengevaluasi kinerja personil-personilnya.
2. Informasi Pengarahan Perhatian (*Attention Directing of Information*), merupakan informasi untuk membantu manajemen memusatkan perhatian pada masalah-masalah yang menyimpang, ketidakberesan, ketidakefisienan dan kesempatan-kesempatan yang dapat dilakukan. Informasi ini akan

membantu manajemen untuk melihat penyimpangan-penyimpangan yang terjadi.

3. Informasi Pemecahan Masalah (*Problem Solving of Information*), merupakan informasi untuk membantu manajer dalam pengambilan keputusan dalam memecahkan masalah yang dihadapinya.

Untuk mendukung keputusan yang akan dilakukan oleh manajemen, maka manajemen membutuhkan informasi yang berguna untuk tiap-tiap tingkatan karakteristik yang berbeda pula. karakteristik informasi misalnya adalah kepadatan informasinya, luas informasinya, waktu informasinya, akses informasinya dan sumber informasinya.

2.2.2 Karakteristik Informasi yang Baik

Informasi dapat dikatakan baik jika memiliki kriteria atau karakteristik sebagai berikut:

1. *Information must be patient*

Informasi harus saling berhubungan. Pernyataan informasi harus berhubungan dengan urusan dan masalah yang penting bagi penerima informasi (orang yang membutuhkan informasi tersebut).

2. *Information must be accurate*

Informasi harus bebas dari kesalahan-kesalahan dan tidak memiliki bias atau menyesatkan. Informasi yang dihasilkan harus mencerminkan maksudnya. Keakuratan informasi seringkali bergantung pada keadaan.

3. *Information must be timely*

Informasi harus ada ketika dibutuhkan. Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan di dalam pengambilan keputusan.

4. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang yang satu dengan yang lainnya pasti berbeda.

2.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005).

2.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari komponen *input*, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran. Berikut ini adalah penjelasan dari masing-masing komponen yang terlibat di dalamnya, yaitu:

1. Komponen *input*

Input mewakili data yang masuk ke dalam sistem informasi. *Input* disini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Komponen model

Komponen ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Komponen *output*

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen 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. Komponen *hardware*

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi. Yang berfungsi sebagai tempat untuk menampung basis data atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung, dan memanipulasi data yang diambil dari *hardware* untuk menciptakan suatu informasi.

7. Komponen basis data

Merupakan blok yang berisi definisi basis data yang disediakan untuk menyimpan data-data yang akan disimpan dalam media penyimpan. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

8. Komponen kontrol

Merupakan sebuah komponen yang bertugas mendefinisikan bagaimana kontrol terhadap sistem dilakukan sehingga sistem dapat berjalan dengan baik. Dalam blok kontrol ini misalnya didefinisikan bagaimana melindungi data yang ada di basis data agar selalu sama dengan kenyataan yang dicatat.

2.3.2 Klasifikasi dan Tujuan Sistem Informasi

Ada beberapa cara untuk mengelompokkan sistem informasi. Klasifikasi yang umum dipakai antara lain didasarkan pada:

1. Level Organisasi, berdasarkan level organisasi sistem informasi dikelompokkan menjadi sistem informasi departemen, sistem informasi perusahaan, dan sistem informasi antar organisasi.

2. Area Fungsional, sebagaimana diketahui bahwa dalam sebuah organisasi memiliki sejumlah bidang fungsional seperti akuntansi, pemasaran, produksi, dan sebagainya.

Tujuan dari sistem informasi adalah:

1. Menyediakan informasi untuk membantu pengambilan keputusan manajemen.
2. Membantu petugas di dalam melaksanakan operasi perusahaan dari hari ke hari.
3. Menyediakan informasi yang layak untuk pemakai pihak luar perusahaan.

2.4 Pengertian Pengendalian

Pengendalian berasal dari kata “kendali” yang berarti kekang. Menurut Kamus Besar Bahasa Indonesia (2000), pengendalian adalah proses, cara, perbuatan mengendalikan, atau pengawasan atas kemajuan (tugas) dengan membandingkan hasil dan sasaran secara teratur serta menyesuaikan usaha (kegiatan) dengan hasil pengawasan.

Pengendalian menurut Glenn A. Welsch, Hilton, dan Gordon yang diterjemahkan oleh Purwatiningsih dan Maudy Warouw (2003) adalah suatu proses untuk menjamin terciptanya kinerja yang efisien yang memungkinkan tercapainya tujuan perusahaan. Menurut Syamsi (1983), Pengendalian adalah fungsi manajemen yang mengusahakan agar pekerjaan atau kegiatan terlaksana sesuai dengan rencana, instruksi, pedoman, patokan, pengaturan, atau hasil yang telah ditetapkan sebelumnya.

Berdasarkan pengertian diatas, dapat ditarik kesimpulan bahwa pengendalian adalah usaha untuk membandingkan prestasi kerja dengan rencana, pemeriksaan, evaluasi dan pemantauan atas pekerjaan yang telah dijalankan sesuai dengan rencana atau tujuan yang telah ditetapkan sebelumnya, dilakukan secara terus menerus agar dapat berfungsi sebaik mungkin untuk mencapai tujuan perusahaan.

2.5 Konsep Dasar Kualitas

Tidak dapat dipungkiri bahwa kualitas merupakan salah satu komponen yang dapat menjadi modal dan alat yang tangguh bagi organisasi maupun agar dapat bertahan dan bahkan menjadi unggul dalam kompetisi pada era manapun. Berbagai kalangan beranggapan bahwa kualitas hanya dapat diidentikan dengan produk atau mutu produk. Ketika industri telah memasuki era produk massal, banyak yang berpandangan skeptis bahwa masa sebuah organisasi menjunjung tinggi semangat untuk menghasilkan produk dengan kualitas yang baik sudah usai.

Dikutip dari buku Nursya'bani Purnama (2006), menurut Evan dan Dean (2003) kualitas adalah keseluruhan ciri-ciri dan karakteristik dari suatu produk atau layanan menyangkut kemampuan untuk memenuhi kebutuhan-kebutuhan yang telah ditentukan atau yang bersifat laten. Sedangkan menurut filsafat Jepang, kualitas adalah *zero defect* mengerjakan pertama kali dengan benar.

2.6 Konsep Dasar Pengendalian Kualitas

Pengendalian kualitas merupakan teknik yang sangat bermanfaat agar suatu perusahaan dapat mengetahui kualitas produknya sebelum dipasarkan kepada konsumen. Teknik pengendalian kualitas dapat membantu perusahaan dalam mengetahui kelayakan kualitas produk berdasarkan batas-batas kontrol yang telah ditentukan.

Berikut ini merupakan pengertian pengendalian menurut para ahli:

1. Menurut Assauri (1998)

Pengendalian dan pengawasan adalah kegiatan yang dilakukan untuk menjamin agar kegiatan produksi dan operasi yang dilaksanakan sesuai dengan apa yang direncanakan dan apabila terjadi penyimpangan, maka penyimpangan tersebut dapat dikoreksi sehingga apa yang diharapkan dapat tercapai.

2. Menurut Gasperz (2005)

Pengendalian adalah sebagai kegiatan yang dilakukan untuk memantau aktivitas dan memastikan kinerja sebenarnya yang dilakukan telah sesuai dengan yang direncanakan.

Berdasarkan pengertian tersebut, maka dapat ditarik kesimpulan bahwa pengendalian kualitas adalah suatu teknik dan aktivitas/tindakan yang terencana yang dilakukan untuk mencapai, mempertahankan dan meningkatkan kualitas suatu produk dan jasa agar sesuai dengan standar yang telah ditetapkan dan memenuhi kepuasan konsumen.

Pengendalian kualitas memiliki tujuan tertentu. Adapun tujuan dari pengendalian kualitas menurut Assauri (1998) adalah:

1. Agar barang hasil produksi dapat mencapai standar kualitas yang telah ditetapkan.
2. Mengusahakan agar biaya inspeksi dapat menjadi sekecil mungkin.
3. Mengusahakan agar biaya desain dari produk dan proses dengan menggunakan kualitas produksi tertentu dapat menjadi serendah mungkin.
4. Mengusahakan agar biaya produksi dapat menjadi serendah mungkin.

Tujuan utama pengendalian kualitas adalah untuk mendapatkan jaminan bahwa kualitas produk atau jasa yang dihasilkan sesuai dengan standar kualitas yang telah ditetapkan dengan mengeluarkan biaya yang ekonomis atau serendah mungkin. Pengendalian kualitas tidak dapat dilepaskan dari pengendalian produksi karena pengendalian kualitas merupakan bagian dari pengendalian produksi. Pengendalian produksi baik secara kualitas maupun kuantitas merupakan kegiatan yang sangat penting dalam suatu perusahaan. Hal ini disebabkan karena semua kegiatan produksi yang dilaksanakan dan dikendalikan, supaya barang dan jasa yang dihasilkan sesuai dengan rencana yang telah ditetapkan, dimana penyimpangan-penyimpangan yang terjadi diusahakan serendah-rendahnya.

Pengendalian kualitas juga menjamin barang atau jasa yang dihasilkan dapat dipertanggungjawabkan seperti halnya pada pengendalian produksi. Dengan demikian antara pengendalian kualitas dan pengendalian produksi erat kaitannya dalam pembuatan barang.

2.7 Pengertian Bahan Baku

Bahan baku adalah bahan yang sebagian besar membentuk produk setengah jadi (barang jadi) atau menjadi bagian wujud dari suatu produk yang dapat ditelusuri ke produk tersebut. Sedangkan menurut Mulyadi (2005), bahan baku merupakan bahan yang membentuk bagian menyeluruh produk jadi. Bahan baku yang diolah dalam perusahaan manufaktur dapat diperoleh dari pembelian lokal, impor, atau dari pengolahan sendiri. Didalam memperoleh bahan baku, perusahaan tidak hanya mengeluarkan sejumlah harga beli bahan baku saja, tetapi juga mengeluarkan biaya-biaya pembelian, pergudangan, dan biaya-biaya perolehan lain. Adapun point pokok pembahasan teori konsep para ahli atau pakar sebagai berikut :

1. Hanggana (2006)

Bahan baku adalah sesuatu yang digunakan membuat barang jadi, bahan pasti menempel menjadi satu dengan barang jadi.

2. Baroto (2002)

Bahan baku adalah barang-barang yang terwujud untuk digunakan perusahaan dalam proses produksi.

2.8 *System Development Life Cycle (SDLC)*

System Development Life Cycle (SDLC) atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji baik (Rosa dan Shalahuddin, 2014).

Tahapan yang ada pada SDLC secara global adalah sebagai berikut:

1. Inisiasi (*initiation*)
Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.
2. Pengembangan konsep sistem (*system concept development*)
Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana dan pembelajaran kemudahan sistem.
3. Perencanaan (*planning*)
Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.
4. Analisis kebutuhan (*requirements analysis*)
Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.
5. Desain (*design*)
Mentransformasikan kebutuhan *detail* menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.
6. Pengembangan (*development*)
Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean, pengkompilasian, memperbaiki dan membersihkan program serta peninjauan pengujian.
7. Integrasi dan pengujian (*integration and test*)
Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user* sehingga menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

Termasuk pada persiapan implementasi, implementasi perangkat lunak pada lingkungan produksi (lingkungan luar *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.

9. Operasi dan pemeliharaan (*operations and maintenance*)

Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi akhir dan masuk pada proses peninjauan.

10. Disposisi (*disposition*)

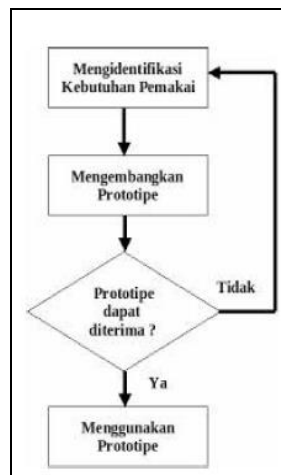
Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

2.9 Model Prototype

Menurut McLeod (2011) prototype adalah suatu versi sistem potensial yang disediakan bagi pengembang dan calon pengguna yang dapat memberikan gambaran bagaimana kira-kira sistem tersebut akan berfungsi bila telah disusun dalam bentuk yang lengkap. Proses dalam memproduksi suatu prototype disebut *prototyping*. Tujuannya adalah menghasilkan prototype secepat mungkin dan memperoleh umpan balik dari pengguna yang akan memungkinkan prototype untuk ditingkatkan sampai sistem dianggap sempurna. Adapun jenis *prototype* terdapat dua jenis prototype yaitu prototype evolusioner (*evolutionary prototype*) dan prototype *requirement* (*requirement prototype*)

2.9.1 Prototype Evolusioner

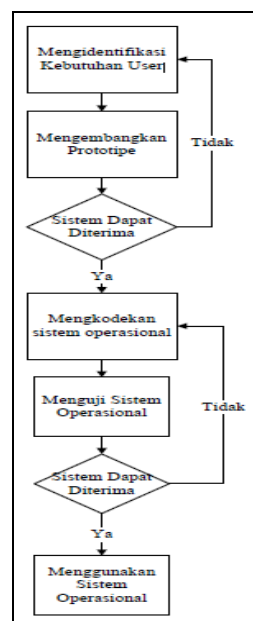
Prototype evolusioner adalah prototype yang terus menerus diperbaiki sampai semua kriteria sistem yang baru terpenuhi. Ada empat langkah yang diambil dalam mengembangkan suatu prototype evolusioner yaitu identifikasi kebutuhan pengguna, mengembangkan prototype, menentukan prototype dapat diterima atau tidak, dan penggunaan prototype.



Gambar II.2. Pengembangan Prototipe Evolusioner
(Sumber: McLeod, 2011)

2.9.2 *Prototype Requirement*

Prototype requirement adalah suatu pengembangan untuk menentukan kebutuhan fungsional dari sistem baru pada saat para pengguna tidak mampu mengungkapkan dengan tepat apa yang mereka butuhkan. Saat kebutuhan telah ditentukan *prototype requirement* dapat mulai dikerjakan dan proyek siap untuk mengembangkan suatu sistem yang baru.



Gambar II.3. Pengembangan Prototipe *Requirement*
(Sumber: McLeod, 2011)

2.10 Flowchart

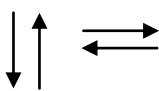


Flowchart merupakan penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analisis dan programmer untuk memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. (Febriani, 2015).

Simbol-simbol *flowchart* yang digunakan merupakan simbol-simbol *flowchart* standar yang dikeluarkan oleh ANSI dan ISO. Berikut simbol-simbol standar yang digunakan untuk menggambarkan diagram alir sesuai kegunaan simbol. (Febriani, 2015):

1. Flow Direction Symbols

Simbol yang dipakai untuk menghubungkan antara simbol yang satu dengan simbol lainnya atau disebut juga *connecting line*. Tabel II.1 menjelaskan tentang simbol-simbol penghubung dalam menggambarkan diagram alir.

Tabel II.1 Simbol *Flow Direction*


Simbol	Nama	Keterangan
	<i>Arus/Flow</i>	Penghubung antara prosedur/proses.
	<i>Connector</i>	Simbol keluar/masuk prosedur atau proses dalam lembar/halaman yang sama.
	<i>Off-line Connector</i>	Simbol keluar/masuk prosedur atau proses dalam lembar/halaman yang lain.



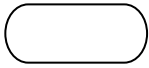

Sumber: Febriani (2015)

2. Simbol Proses

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu prosedur. Tabel II.2 menjelaskan tentang simbol-simbol proses dalam menggambarkan diagram alir.

Tabel II.2 Simbol Proses

Simbol	Nama	Keterangan
	<i>Process</i>	Simbol yang menunjukkan pengolahan yang dilakukan komputer.




	<i>Decision</i>	Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi.
	<i>Predafined Process</i>	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam <i>storage</i> .
	Terminal	Simbol untuk permulaan atau akhir dari suatu program.
	Manual Proses	Simbol untuk pemasukan data secara manual <i>on-line</i> keyboard.

Sumber: Febriani (2015)

3. Simbol *Input* dan *Output*

Simbol yang dipakai untuk menyatakan jenis peralatan yang digunakan sebagai media *input* atau *output*. Tabel II.3 menjelaskan tentang simbol-simbol *input* dan *output* dalam menggambarkan diagram alir.

Tabel II.3 Simbol *Input* dan *Output*

Simbol	Nama	Keterangan
	<i>Input-Output</i>	Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<i>Document</i>	Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> dicetak di kertas.
	<i>Disk and On-line Storage</i>	Simbol untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i> .

Sumber: Febriani (2015)

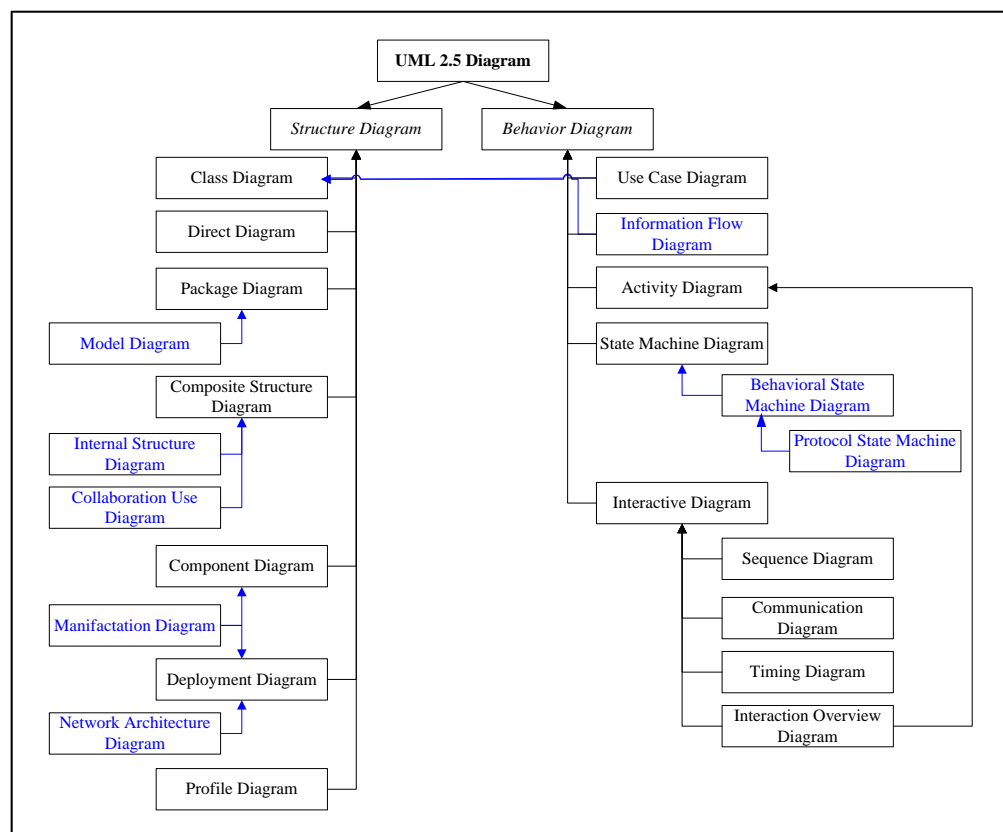
2.11 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. UML merupakan standar terbuka yang menjadikannya sebagai

bahasa pemodelan yang umum dalam industri piranti lunak dan pengembangan sistem (Rama dan Jones, 2008).

Dimulai pada bulan Oktober 1994, Booch, Rumbaugh dan Jacobson yang merupakan tiga tokoh yang boleh dikatakan metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi perancangan berorientasi objek. Pada tahun 1995 dirilis pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group (OMG)* (Dharwiyanti, 2015).

UML dikelompokkan menjadi 2 kategori, yaitu *Structure Diagrams* dan *Behavior Diagrams*. Pengelompokan UML dapat dilihat pada Gambar II.4.



Gambar II.4 UML Diagram
Sumber: uml-diagrams.org (2015)

Berikut ini penjelasan singkat dari pembagian kategori UML tersebut (Rosa dan Shalahuddin, 2014):


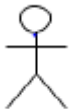


1. *Structure diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.



2. *Behavior diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan proses sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.11.1 Use Case Diagram

Use case adalah pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

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

No.	Simbol	Deskripsi
1.	<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
2.	<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 gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	<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.
4.	<i>Extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri







		walaupun tanpa <i>use case</i> tambahan itu.
5.	<p><i>Generalization</i></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.
6.	<p><i>Include</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Sumber: Rosa dan Shalahuddin (2014)

2.11.2 Activity Diagram

Activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel II.5 Simbol-Simbol *Activity Diagram*

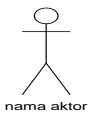
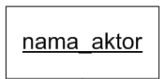

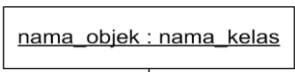

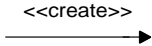
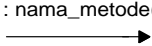
No.	Simbol	Deskripsi
1.	<p>Status awal</p> 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	<p>Aktivitas</p> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	<p>Percabangan/decision</p> 	Asosiasi percabangan dimana jika ada pilihan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

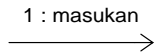
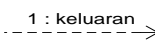
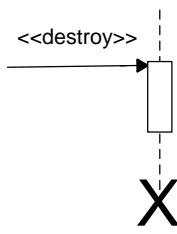
Sumber: Rosa dan Shalahuddin (2014)

2.11.3 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case*.

Tabel II.6 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama aktor</p>  <p>Tanpa waktu aktif</p>	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 gambar orang, tapi aktor belum tentu merupakan orang.
2.	<p>Garis Hidup/<i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	<p>Pesan tipe create</p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	<p>Pesan tipe call</p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode.

7.	<p>Pesan tipe send</p> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	<p>Pesan tipe return</p> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	<p>Pesan Tipe Destroy</p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Sumber: Rosa dan Shalahuddin(2014)

2.11.4 Class Diagram

Class diagram merupakan diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas sebagai berikut (Rosa dan Shalahuddin, 2011):

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case*

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.

4. Kelas yang diambil dari pendefinisian data

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Di dalam kelas terdapat beberapa *keys* yang berfungsi untuk membedakan semua basis data dalam tabel secara unik. Pengertian *keys* menurut Connolly dan Begg (2010) dalam Bina Nusantara (2012) yaitu sebagai berikut:

1. *Candidate key*

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

2. *Primary key*

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

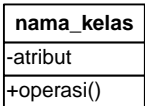



3. *Composite key*

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

4. *Foreign key*

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

Tabel II.7 Simbol-Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktur sistem.
2.	<p>Antarmuka/interface</p> 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi/association</p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

5.	Generalisasi —————▷	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
6.	Kebergantungan ----->	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi/aggregation —————◊	Relasi antar kelas dengan makna semua-bagian (whole-part).

Sumber: Rosa dan Shalahuddin (2011)

Multiplicity pada UML memungkinkan untuk menentukan kardinalitas yaitu jumlah elemen dari beberapa koleksi elemen. *Multiplicity* adalah definisi kardinalitas yaitu jumlah elemen dari beberapa koleksi elemen dengan memberikan interval termasuk bilangan bulat non-negatif untuk menentukan jumlah yang diijinkan contoh elemen dijelaskan. *Interval* banyaknya telah terikat beberapa *lower bound* dan (mungkin tak terbatas) *upper bound* (uml-diagrams, 2015):

Multiplicity-range ::= [*lower-bound* '..'] *upper-bound*

Lower-bound ::= *natural-value-specification*

Upper-bound ::= *natural-value-specification* | '*'

Lower dan *upper bounds* bisa konstanta alam atau ekspresi konstan dievaluasi untuk alam (non negatif) nomor. *Upper bound* bisa juga ditetapkan sebagai tanda '*' yang menunjukkan jumlah yang tidak terbatas elemen. *Upper bound* harus lebih besar dari atau sama dengan *lower bound*. Tabel II.8 menjelaskan beberapa contoh tipe *multiplicity* (uml-diagrams, 2015).

Tabel II.8 Tipe *Multiplicity*

<i>Multiplicity</i>	<i>Option</i>	<i>Cardinality</i>
0..0	0	Nol
0..1		Antara nol sampai satu
1..1	1	Satu
0..*	*	Antara nol sampai banyak
1..*		Antara satu sampai banyak
5..5	5	Tepat lima
m..n		Sedikitnya m tetapi tidak boleh lebih dari n

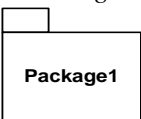
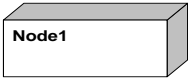
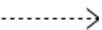

Sumber: uml-diagrams (2015)

2.11.5 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. *Deployment diagram* juga dapat digunakan untuk memodelkan hal-hal berikut (Rosa dan Shalahuddin, 2014):

1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node* dan *hardware*.
2. Sistem *client/server*.
3. Sistem terdistribusi murni.
4. Rekayasa ulang aplikasi.

Tabel II.9 Simbol-simbol *Deployment Diagram*

No.	Simbol	Deskripsi
1.		Package merupakan sebuah bungkus dari satu atau lebih node.
2.		Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka, komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3.		Kebergantungan antar node, arah panah mengarah pada node yang dipakai.
4.		Relasi antar node.

Sumber: Rosa dan Shalahuddin (2014)

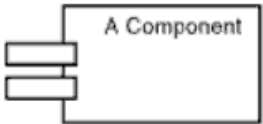



2.11.6 Component Diagram

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam

sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut:

1. *Source code* program perangkat lunak
2. Komponen *executable* yang dilepas ke *user*
3. Basis data secara fisik
4. Sistem yang harus beradaptasi dengan sistem lain
5. *Framework* sistem, *framework* pada perangkat lunak merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi.

Tabel II.10 Simbol-simbol *Component Diagram*

No.	Simbol	Deskripsi
1.	<p>Komponen</p> 	Komponen system
2.	<p>Kebergantungan/dependency</p> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
3.	<p>Antarmuka/interface</p> 	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen.
	<p>Link</p> 	Relasi antar komponen

Sumber: Rosa dan Shalahuddin (2014)

2.12 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.

Keuntungan penggunaan kamus data adalah (Sommerville, 2003):

1. Kamus data merupakan mekanisme untuk manajemen nama. Banyak orang yang harus menciptakan nama untuk entitas dan relasi ketika mengembangkan model sistem yang besar. Nama-nama ini harus dipakai secara konsisten dan tidak boleh bentrok. Kamus data dapat memeriksa keunikan nama dan memberitahu analis persyaratan sekiranya terjadi duplikasi nama.
2. Kamus data sebagai tempat penyimpanan informasi yang dapat menghubungkan analis, desain, implementasi dan evolusi. Sementara sistem dikembangkan, informasi diambil untuk memberitahu perkembangan informasi baru ditambahkan pada sistem. Semua informasi mengenai entitas berada pada satu tempat. Berikut adalah contoh penulisan kamus data:

Spesifikasi Tabel pemasok

Nama Tabel : Pemasok

Tipe : File master

Tabel II.11 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)

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

HIPO (*Hierarchy plus Input-Process-Output*) aslinya dibuat oleh IBM sebagai alat untuk mendokumentasikan program. Bagan HIPO merupakan bagan

yang memperagakan apa yang dikerjakan suatu program, data apa yang digunakan, dan keluaran yang dihasilkannya (Zulkifli, 2005).

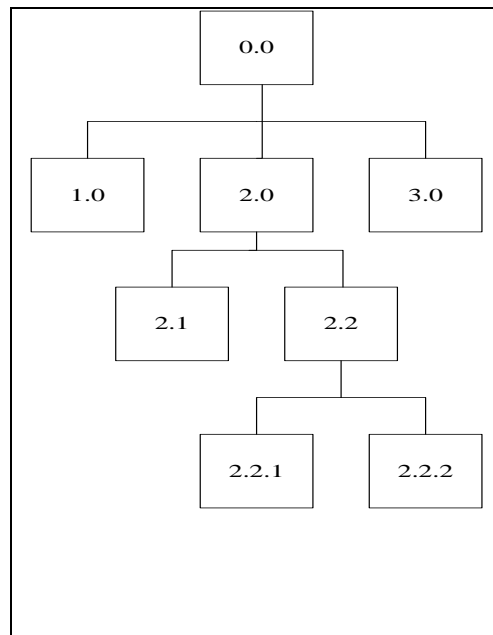
HIPO (*Hierarchy plus Input-Process-Output*) mempunyai sasaran utama sebagai berikut (Jogiyanto, 2005):

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

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

1. *Visual Table Of Contents* (VTOC)

Visual table of contents menggambarkan hubungan fungsi-fungsi di *system* secara berjenjang, *VTOC* menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada diagram ini nama dan nomor dari program HIPO, struktur paket diagram dan hubungan fungsi diidentifikasi dalam bentuk hirarki. Keterangan masing-masing fungsi diberikan pada bagian penjelasan yang diikutsertakan dalam diagram ini. Berikut adalah Gambar II.5 *Visual table of contents*,

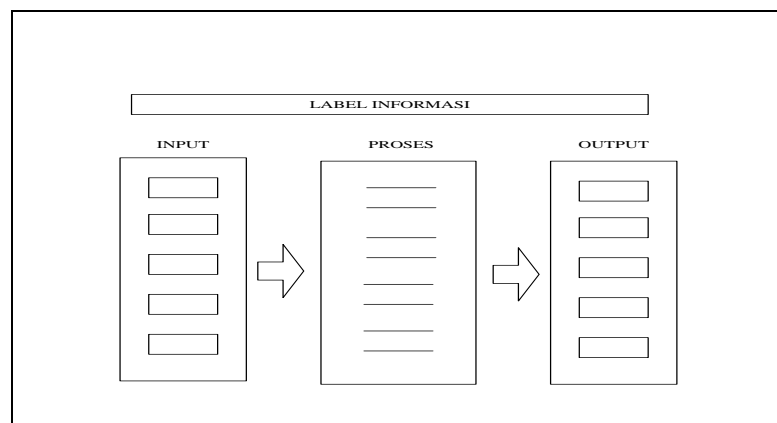


Gambar II.5 *Visual Table Of Contents*
Sumber: Jogiyanto (2005)

2. *Overview Diagram*

Overview Diagram menunjukkan secara garis besar hubungan dari *input*, proses dan *output*. Bagian *input* menunjukkan item-item data yang akan digunakan oleh bagian proses. Bagian proses berisi sejumlah langkah-langkah yang menggambarkan kerja dari fungsi. Bagian *output* berisi dengan item-item data yang dihasilkan atau dimodifikasi oleh langkah-langkah proses.

Berikut adalah Gambar II.6 *Overview diagram*.



Gambar II.6 *Overview Diagram*
Sumber: Jogiyanto (2005)

3. *Detail Diagram*

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

2.14 **PHP (*Hypertext Pre-Processor*)**

PHP Hypertext Preprocessor (PHP) adalah bahasa *server-side scripting* yang digunakan untuk aplikasi web yang dinamis dan interaktif. Sebuah halaman PHP adalah sebuah halaman *HTML* yang memiliki *server-side scripts* yang ditempatkan dalam *server* dan diproses oleh web *server* sebelum dikirim ke *browser* pemakai (Welling dan Thomson, 2003).

Server-side scripts dijalankan ketika *browser* melakukan permintaan *file.php* dari *server*. PHP dipanggil oleh web *server*, dimana proses script perintah yang ada di suatu halaman dieksekusi mulai dari awal sampai akhir di dalam mesin PHP. Setelah *script* PHP tersebut diolah, hasilnya akan ditampilkan kepada *client* melalui web *browser* berupa tampilan *HTML*. Menurut Welling dan Thomson (2003), beberapa keunggulan PHP adalah:

1. *High Performance*

PHP sangat efisien. Dengan menggunakan *server* tunggal yang tidak mahal, *user* dapat melakukan banyak pekerjaan setiap harinya.

2. *Database Integration*

PHP mempunyai sambungan ke banyak sistem basis data, antara lain MySQL, PostgreSQL, Oracle, Informix, dan Sysbase databases.

3. *Built-in-Libraries*

PHP dirancang khusus untuk web, dan mempunyai banyak *built-in-function* untuk menampilkan banyak fungsi di dalam web.

4. Harga yang murah

PHP adalah perangkat lunak gratis.

5. Mudah dalam pembelajaran dan penggunaan

Sintaks PHP berdasarkan bahasa pemrograman lainnya, terutama C dan Java.

6. *Portability*

PHP dapat digunakan di banyak sistem operasi yang berbeda.

7. Ketersediaan *Source Code*

Kode PHP dapat langsung diakses dan dimodifikasi secara bebas.

2.15 MySQL

My Structured Query Language (MySQL) adalah *Relational Database Management System* (RDBMS) yang sangat cepat dan kuat. Sebuah basis data memungkinkan *user* untuk melakukan penyimpanan yang efisien, pencarian dan pengurutan data. MySQL *server* memiliki kemampuan melakukan kontrol akses terhadap data untuk memastikan bahwa setiap *user* dapat bekerja dengan sesuai, menyediakan akses yang cepat, dan meyakinkan bahwa hanya *user* yang mempunyai hak akses yang dapat mengaksesnya. Oleh karena itu, MySQL merupakan *database server* yang yang dapat digunakan banyak *user* dan banyak pekerjaan (Welling dan Thomson, 2003).

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, pengubahan, dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis (Sutaji, 2012).

2.15.1 Keunggulan MySQL

Berikut keunggulan dari MySQL, di antaranya adalah (Sutaji, 2012):

1. *Portability*

Dapat berjalan stabil pada berbagai sistem operasi, di antaranya: Windows, Linux, FreeBSD, MacOS X Server, Solaris, Asigma.

2. *Open source*

Didistribusikan secara gratis di bawah lisensi dari *General Public License* (GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan bersifat *close source* (komersial).

3. *Multi User*

Dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan.

4. *Performance tuning*

Mempunyai kecepatan yang tinggi dalam menangani *query*.

5. *Column types*

Memiliki tipe data yang sangat kompleks, seperti: *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year* dan *enum*.

6. *Command dan function*

Memiliki operator dan fungsi penuh yang mendukung *select* dan *where* dalam *query*.

7. *Security*

Memiliki beberapa lapisan sekuritas seperti tingkat *subnet mask, hostname, privilege user* dengan sistem perijinan yang mendetail serta *password* yang ter-enkripsi.

8. *Scalability dan limits*

Mampu menangani basis data dalam jumlah besar, dengan jumlah *field* lebih dari 50 juta, 60 ribu tabel, dan 5 miliar *record*. Batas indeks mencapai 32 buah per tabel.

9. *Localization*

Dapat mendeteksi pesan kesalahan (*error code*) pada klien dengan menggunakan lebih dari 20 bahasa.

10. *Connectivity*

Dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix Socket, Named Pipes*.

11. *Interface*

Memiliki antarmuka terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan API.

12. *Client dan tools*

Dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi basis data sekaligus dokumen petunjuk *online*.

13. Struktur tabel

Memiliki struktur tabel yang lebih fleksibel dalam menangani *alter* tabel dibandingkan dengan PostgreSQL dan Oracle.

2.15.2 Tipe Data MySQL

MySQL mendukung banyak tipe data yang dapat disimpan pada sebuah kolom. Penjelasan singkat kegunaan masing-masing jenis data akan sangat bermanfaat dalam memilih suatu jenis data yang dipakai dalam merancang *table*. Beberapa jenis data yang tersedia pada MySQL dapat dilihat pada Tabel II.12 (Sutaji, 2012).

Tabel II.12 Jenis Data pada MySQL

Jenis Data	Keterangan
CHAR	Sebuah <i>string</i> dengan panjang tetap. Sisa Jumlah karakter yang belum terisi akan diisi dengan spasi, akan tetapi spasi ini dibuang jika data dipanggil. Jangkauan nilai M adalah 1-255 karakter
VARCHAR	<i>String</i> dengan panjang berupa variabel. M bisa mencapai 65535
DATE	Data berupa tanggal. Format tanggal dalam bentuk 'YYYY-MM-DD'
TIME	Data berupa waktu. Format waktu dalam bentuk 'HH:MM:SS'
TINYINT	Bilangan antara -128 sampai dengan +127
SMALLINT	Bilangan antara -32768 sampai dengan +32767
INT	Bilangan antara -2147683648 sampai dengan +2147683647
FLOAT	Bilangan <i>floating point</i> yang kecil (presisi tunggal). Jangkauan nilai yang diizinkan adalah -3.402823466E+38 s/d -1.175494351e-38,0 dan 1.175494351E-38 s/d 3.402823466E+38
DOUBLE	Bilangan <i>floating point</i> dengan ukuran normal (presisi ganda). Jangkauan nilai yang diizinkan adalah -1.7976931348623157E+308 s/d -2.225073858507201E-308,0 dan 2.225073858507201E-308 s/d 1.7976931348623157E+308
ENUM	Sebuah <i>enumeration</i> . Sebuah obyek string yang hanya boleh memiliki satu nilai, yang terambil dari 'value1','value2',...,NULL atau nilai spesial "" <i>error</i> . Sebuah <i>enum</i> dapat menampung 65535 pilihan nilai
TEXT, BLOB	Sebuah TEXT atau BLOB dengan panjang karakter maksimum 65535 karakter

Sumber: Sutaji (2012)

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian atau *methodology of research* berasal dari kata “*metoda*” yang berarti cara atau teknik dan “*logos*” yang berarti ilmu. Untuk mendapatkan gambaran yang lebih jelas tentang arti metodologi penelitian, maka beberapa pengertian atau definisi metodologi penelitian adalah sebagai berikut:

1. Menurut Komarudin (1997), metodologi penelitian diartikan sebagai suatu proses mencari jawaban terhadap suatu pertanyaan atau masalah melalui prosedur yang sistematis dan terawasi.
2. Menurut Sutrisno Hadi (1987), pengertian metodologi penelitian lebih menekankan pada suatu usaha untuk menemukan, mengembangkan dan menguji kebenaran suatu pengetahuan, dimana dilakukan dengan menggunakan metoda-metoda ilmiah.

Dari beberapa pengertian dan definisi yang tersebut diatas maka dapat diperoleh kesimpulan pengertian metodologi penelitian adalah sebagai suatu usaha atau proses mencari jawaban atas suatu pertanyaan atau masalah dengan cara sabar, hati-hati, terencana, sistematis atas fakta atau prinsip-prinsip, mengembangkan dan menguji kebenaran ilmiah suatu pengetahuan.

3.2 Jenis dan Sumber Data

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

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 pengendalian produksi diantaranya struktur organisasi, analisis sistem yang telah berjalan, proses bisnis sistem saat ini dan yang akan diusulkan, dan kebutuhan pengguna sistem.

2. Data Sekunder

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

3.3 Metode Pengumpulan Data

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

1. Studi Lapangan

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

- a. Pengamatan, tahap ini dilakukan secara langsung di Bagian Produksi PT Gaya Motor dengan mengamati prosedur pengendalian kualitas yang sedang berjalan mulai dari penerimaan bahan baku oleh gudang sampai dengan pencatatan data kerusakan bahan baku pada monitor *problem*. Melalui teknik ini data yang dibutuhkan untuk diamati, dikumpulkan dan diolah sebagai bahan dalam penelitian.
- b. Wawancara, yaitu mencari data yang dibutuhkan secara langsung melalui memberikan pertanyaan spontan terhadap segala hal yang diperlukan pada penyusunan Tugas Akhir ini. Wawancara ini dilakukan kepada karyawan pada Bagian produksi di PT Gaya Motor.

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.

3.4 Metode Pengembangan Sistem

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

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

1. Mengidentifikasi kebutuhan pengguna

Pengembangan mewawancarai pengguna untuk mendapatkan ide mengenai apa yang diminta dari sistem.

2. Membuat satu prototipe

Pengembangan mempergunakan satu alat *prototyping* atau lebih untuk membuat prototipe. Contoh dari alat-alat *prototyping* tersebut adalah generator aplikasi terintegrasi dan *toolkit prototyping*. Generator aplikasi terintegrasi adalah sistem piranti lunak siap pakai yang mampu membuat seluruh fitur yang diinginkan dari sistem baru. Sedangkan *toolkit prototyping* meliputi sistem-sistem piranti lunak terpisah, seperti sistem manajemen basis data, yang

masing-masing mampu membuat sebagian dari fitur-fitur sistem yang diinginkan.

3. Menentukan apakah prototipe dapat diterima

Pengembangan mendemonstrasikan prototipe kepada pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika iya, akan dilakukan langkah selanjutnya, dan jika tidak, prototipe akan direvisi dari tahap awal dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.

4. Menggunakan prototipe

Menjadikan prototipe sebagai sistem yang dijalankan.

3.5 Kerangka Penelitian

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

1. Pendahuluan

Pendahuluan merupakan langkah awal yang dilakukan untuk memulai penelitian. Dengan melakukan studi pustaka, observasi, dan wawancara.

2. Identifikasi Masalah

Pada tahap ini bertujuan untuk mengetahui sistem yang sedang berjalan. Serta mengetahui masalah yang ada pada sistem tersebut.

3. Rumusan Masalah

Tahap ini bertujuan untuk mencari jalan keluar dari masalah yang ada pada sistem.

4. Tujuan Penelitian

Tujuan penelitian dimaksudkan agar dapat mencapai tujuan yang hendak dicapai dari penelitian.

5. Batasan Masalah

Pada tahap ini bertujuan untuk membatasi masalah agar penelitian lebih terarah sehingga dapat menunjukkan gambaran yang lebih spesifik mengenai arah pemecahannya.

6. Penerapan Metode *evolutionary prototype*

a. Mengidentifikasi kebutuhan pelanggan

Menganalisis kebutuhan *user* terhadap program yang akan dibuat. Dengan melakukan metode pengumpulan data dan pengolahan data.

b. Membuat prototipe

Membuat sebuah prototipe sesuai dengan analisis yang dilakukan. Pada tahap ini akan dirancang aplikasi tentang sistem pengendalian kualitas bahan baku. Membuat prototipe nantinya akan memuat tentang:

- 1) Memodelkan sistem menggunakan *Unified Modeling Language* (UML).
- 2) Memodelkan basis data dengan menggunakan *class diagram* dan kamus data.
- 3) Merancang sistem dengan menggunakan HIPO, *Flowchart*, dan *Interface*.
- 4) Membuat prototipe dengan menggunakan PHP 7.2.9 sebagai *software* dan MySQL 10.1.35 sebagai *databasenya*.

c. Mengevaluasi prototipe

Memutuskan untuk menggunakan prototipe yang telah divalidasi menggunakan metode *evolutionary prototype*. Apabila prototipe tidak sesuai dengan kebutuhan *user*, maka akan kembali ke tahap identifikasi.

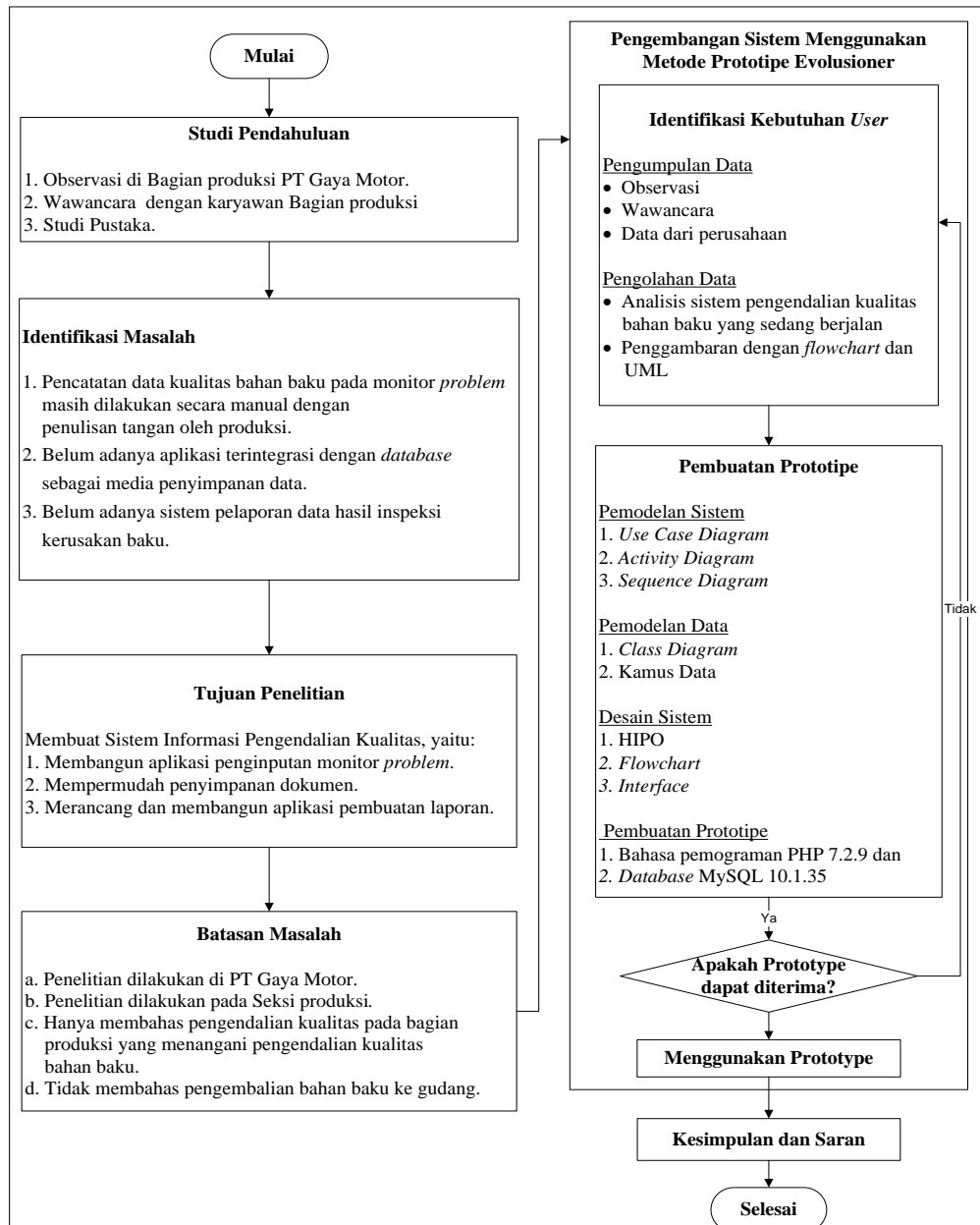
d. Menggunakan prototipe

Tahap ini merupakan tahap akhir dari pembuatan sistem dengan metode *evolutionary prototype*. Pada tahap ini perangkat lunak yang sudah jadi dan sudah lulus uji, siap untuk digunakan oleh *user*.

7. Kesimpulan dan Saran

Kesimpulan digunakan untuk membandingkan hasil penelitian atau pengembangan sistem dengan sistem sebelumnya. Sedangkan saran berisi masukan untuk pengembangan sistem selanjutnya.

Berikut adalah Gambar III.1 *Flowchart* kerangka penelitian untuk menyelesaikan masalah dalam Tugas Akhir ini:



Gambar III.1 Kerangka Penelitian
(Sumber: Pengolahan Data, 2019)

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sekilas Tentang Perusahaan

Dalam hal produk, ruang lingkup usaha PT Gaya Motor hanya bergerak di bidang usaha industri perakitan kendaraan, dimana untuk produknya pada masa lalu dan sekarang ini sangat sering dijumpai di masyarakat. PT Gaya Motor akan selalu menjadi milik yang bermanfaat bagi Bangsa dan Negara. Serta memberikan pelayanan yang terbaik kepada pelanggan. Saling menghargai dan membina kerjasama. Selalu berusaha mencapai yang terbaik, baik untuk Indonesia maupun untuk luar negeri sehingga tidak memungkinkan PT Gaya Motor merakit, dan mengeluarkan produk otomotif dengan jenis baru lainnya demi tetap bertahan dan bersaing dengan perusahaan otomotif lainnya.

4.2 Lokasi Pabrik

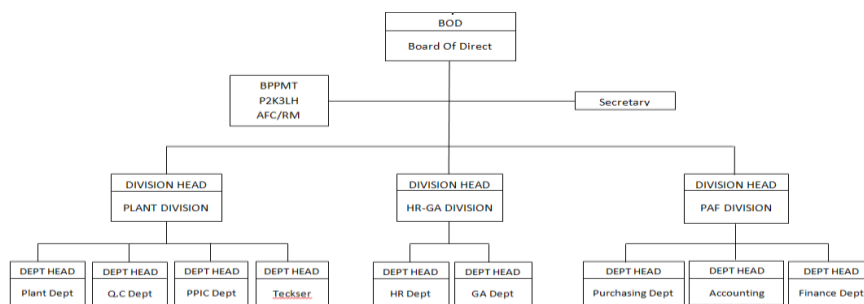
PT. Gaya Motor yang waktu itu berlokasi di jalan Sulawesi No. 22 Tanjung Priok Jakarta Utara menghentikan aktivitasnya karena dalam keadaan likuidasi, dan kebetulan sekali pada waktu itu PT. Astra Internasional (Inc) tidak mempunyai fasilitas assembling yang semula belum banyak bergerak dalam bidang automotive bussines lalu bertekad untuk terjun lebih dalam lagi pada bidang ini.

4.3 Visi, Misi dan Kebijakan Mutu Perusahaan

PT Gaya Motor adalah perusahaan yang bergerak di bidang Supplier sparpert Otomotif, mempunyai visi yaitu Menjadi pelopor industri yang efisien dan produktif untuk menunjang Pembangunan Nasional. Sedangkan misi dari PT Gaya Motor yaitu unggul dalam mutu, bersaing dalam harga, harmonis dalam bekerja.

4.4 Struktur Organisasi Perusahaan

Organisasi adalah suatu bentuk atau wadah dari kelompok manusia dalam usaha untuk mencapai tujuan. Agar organisasi dapat bekerja dengan baik maka diperlukan struktur organisasi. Struktur organisasi merupakan suatu tingkatan dari suatu fungsi dan tanggung jawab dalam sebuah instansi, dimana memperlihatkan hubungan garis wewenang dan pertanggungjawaban antara jabatan dan peranan dalam suatu organisasi.

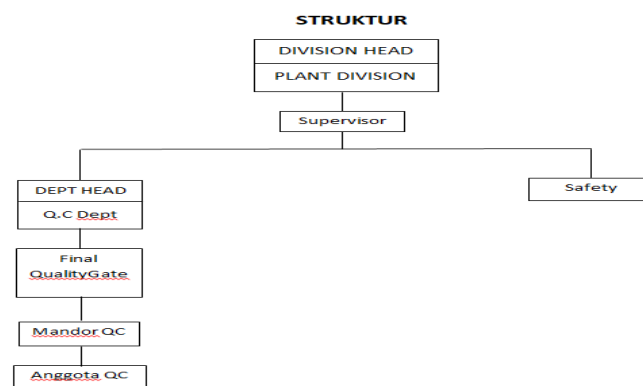


Gambar IV.1 Struktur Organisasi PT GM

Sumber: PT Gaya Motor (2019)

4.5 Struktur Organisasi Departemen Quality Control

Berikut struktur organisasi departemen quality control di PT Gaya Motor, pada gambar IV.2.



Gambar IV.2 Struktur Organisasi Departemen Quality Control

Sumber: PT Gaya Motor (2019)

Berikut *job desk* struktur organisasi departemen quality control :

1. *Supervisor*
 - Bertanggung jawab atas pengawasan pekerjaan dilapangan.
2. *Quality Control*
 - Bertanggung jawab atas mutu pekerjaan
 - Memeriksa kualitas material
 - Mengarahkan mandor untuk melakukan pekerjaan
3. Mandor QC
 - Bertanggung jawab atas mutu pekerjaan
 - Memeriksa kualitas material
 - Mengarahkan anggota qc untuk melakukan pekerjaan
4. Anggota QC
 - Bertanggung jawab atas mutu pekerjaan
5. *Safety*
 - Bertanggung jawab atas keselamatan para anggota qc
 - Mengawasi langsung para anggota

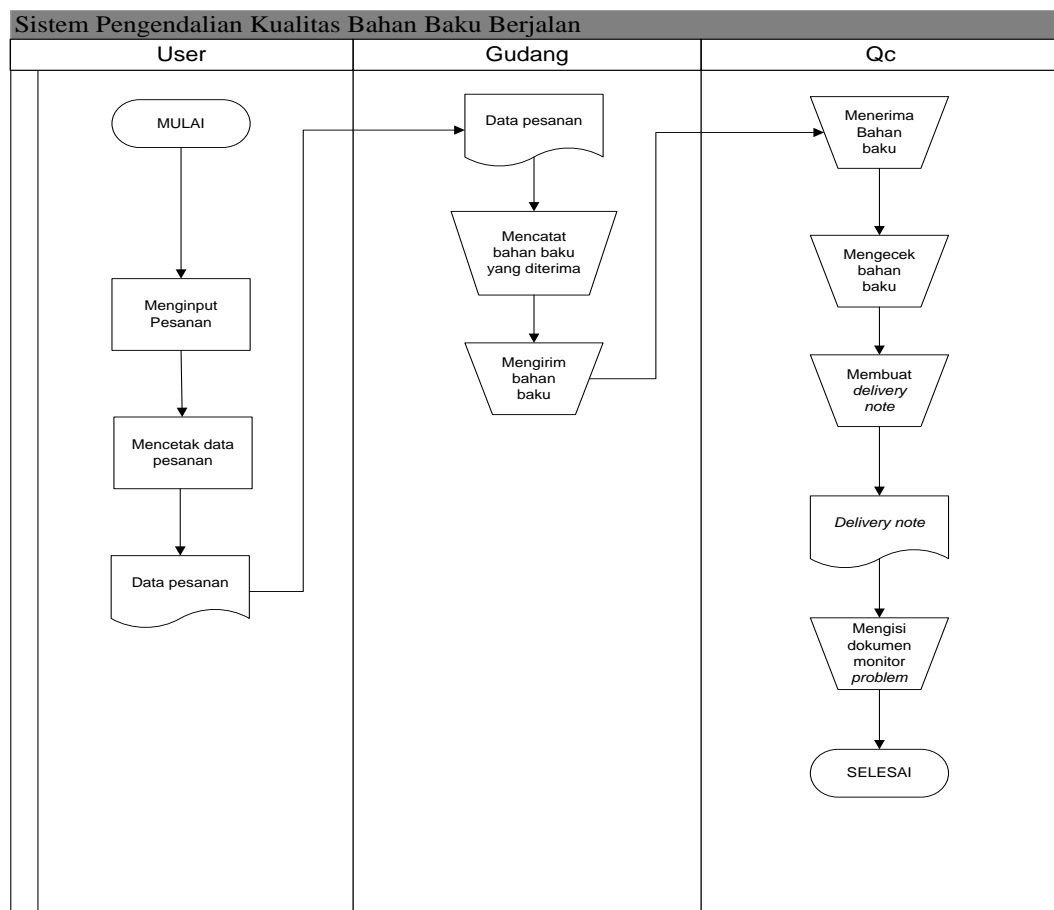
4.6 Dokumen Pengendalian Kualitas

Sistem Informasi Pengendalian kualitas bahan baku pada PT Gaya Motor dilakukan secara manual yaitu pencatatan hasil inspeksi kerusakan bahan baku dengan tulis tangan. Dokumen yang terlibat dalam sistem informasi pengendalian kualitas bahan baku adalah:

1. *Delivery Note*

Delivery Note merupakan lembar pengiriman bahan baku yang dibuat oleh bagian produksi saat menerima akan mengirim bahan baku. Lembar pengecekan tersebut bertujuan untuk mencatat data bahan baku yang diterima dari gudang dan akan dikirim. *Delivery Note* yang telah dibuat selanjutnya akan diserahkan kepada bagian produksi untuk ditandatangani sebagai validasi. Berikut adalah contoh *Delivery Note* yang terdapat di PT Gaya Motor pada Gambar IV.3:

3. Mencatat bahan baku yang diterima pada kartu stok berdasarkan jumlah barang yang diterima dari *supplier*.
4. Mengirim bahan baku kepada bagian *quality control*, guna untuk diproduksi.
5. Menerima barang bahan baku untuk proses produksi, maka pada bagian *quality control* langsung mengecek bahan baku tersebut. Dan membuat form *delivery note*.
6. Mengisi dokumen monitor *problem* untuk dijadikan sebagai arsip pada bagian *quality control*.



Gambar IV.5 Flowmap Proses Pengendalian kualitas Bahan Baku yang berjalan
(Sumber: Hasil Analisis Data, 2019)

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Berdasarkan hasil pengolahan data dan analisis sistem, dibutuhkan perancangan aplikasi sistem informasi pengendalian kualitas bahan baku sebagai sarana untuk memberikan informasi mengenai pelaporan kualitas bahan baku pada bagian produksi. Berikut adalah daftar kebutuhan sistem untuk aplikasi sistem informasi pengendalian kualitas bahan baku.

Tabel V.1 Kebutuhan Sistem Informasi Pengendalian Kualitas

Kebutuhan Sistem	
<i>Project Name</i>	Sistem Informasi Pengendalian Kualitas bahan baku.
<i>Project Sponsor</i>	Bagian Produksi
<i>Business Need</i>	<ol style="list-style-type: none"> 1. Terwujudnya aplikasi yang dapat menginput data hasil pemeriksaan kualitas bahan baku. Sehingga dapat mencegah terjadinya kerusakan dan kehilangan data hasil inspeksi kualitas bahan baku. 2. Terwujudnya suatu aplikasi berbasis data yang dapat memudahkan dalam penyimpanan data dan pelaporan, serta mencegah terjadinya kerusakan dan kehilangan data.
<i>Business Requirement</i>	Memberikan sistem informasi pengendalian kualitas bahan baku yang dapat membantu bagian produksi dalam membuat laporan.
<i>Bussiness Value</i>	Mampu menyimpan laporan kualitas bahan baku dengan baik di basis data.
<i>Special Issues or Constrains</i>	Hanya membahas pengendalian kualitas pada bagian produksi yang menangani pengendalian kualitas bahan baku.

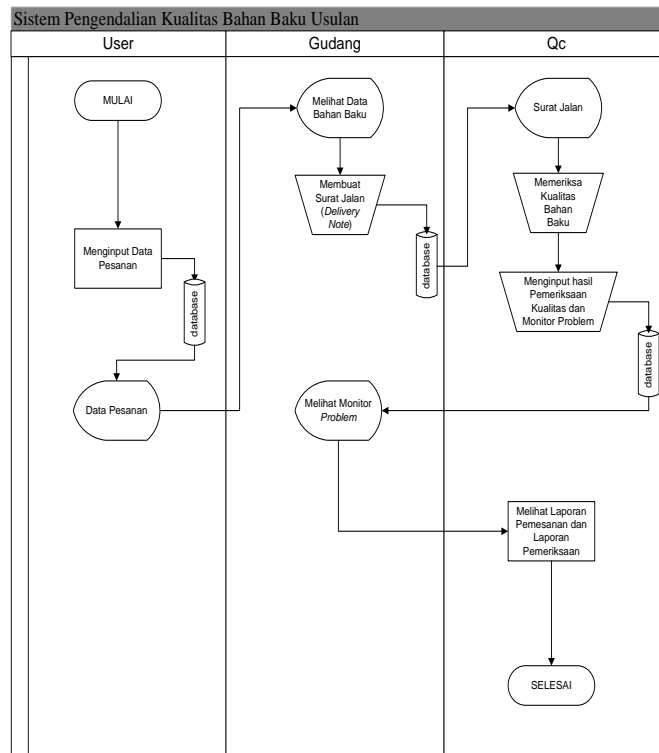
Sumber: Hasil Analisis Data (2019)

5.2 Prosedur Sistem Informasi Pengendalian Kualitas Bahan Baku

Prosedur sistem informasi pengendalian kualitas bahan baku adalah suatu kegiatan yang dilakukan untuk menjamin agar kegiatan produksi yang dilaksanakan agar sesuai dengan yang direncanakan dan apabila terjadi problem/masalah maka dapat dikoreksi sehingga apa yang diharapkan tercapai. Kemudian pada prosedur system informasi diusulkan melalui tahapan sebagai berikut:

1. Bagian User menginput data costumer dan data pesanan, kemudian tersimpan ke dalam *database* yang nantinya akan diserahkan kepada bagian Gudang.
2. Bagian Gudang akan menerima laporan data pesanan, kemudian akan membuat data stok bahan baku dan surat jalan, serta tersimpan ke dalam *database*, yang nantinya akan diserahkan kepada bagian QC.
3. Setelah itu, pada bagian QC akan menerima bahan baku dan surat jalan yang nantinya akan melakukan penginputan hasil pemeriksaan kualitas bahan baku dan tersimpan ke dalam *database*.
 1. Jika YA nantinya akan dilakukan *menginput* hasil pemeriksaan kualitas bahan baku dan dimasukkan dalam *database* sebagai arsip.
 2. Jika TIDAK maka bagian QC akan melaporkan bahan baku dan surat jalan kepada bagian Gudang sebagai monitor *problem* untuk dilakukan pengecekan agar sesuai dengan standar.
 3. Kemudian pada bagian Gudang juga akan melakukan *update status problem*, bila nantinya akan ada monitor *problem* pada bahan baku maka dimasukkan ke dalam *database* sebagai arsip.

Berikut *Flowmap* Sistem Informasi Pengendalian Kualitas Bahan Baku yang diusulkan dapat dilihat pada Gambar V.1.



Gambar V.1 *Flowmap* Sistem Informasi Pengendalian Kualitas Bahan Baku Usulan
Sumber: Hasil Analisis Data (2019)

5.3 Analisis dan Perancangan Sistem Usulan

Untuk mengatasi permasalahan dalam pengendalian kualitas bahan baku yang ada pada bagian produksi, diajukan usulan sistem baru dengan menerapkan penggunaan aplikasi komputer untuk pengolahan data yang akan mendukung beberapa proses bisnis yang sedang berjalan pada perusahaan. Dengan menggunakan aplikasi ini membuat sistem menjadi terintegrasi sehingga dapat menyajikan informasi secara cepat dan dapat meningkatkan fungsionalitas proses pengendalian kualitas bahan baku yang ada pada bagian produksi di PT Gaya Motor.

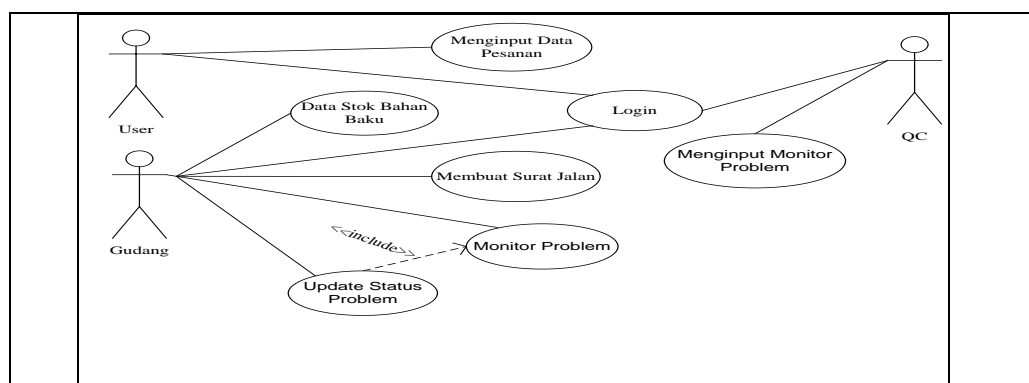
Sistem yang diusulkan yaitu proses *penginputan* hasil pemeriksaan kualitas bahan baku secara terkomputerisasi, membangun basis data penyimpanan data kualitas bahan baku, dan pembuatan laporan data kualitas bahan baku yang kemudian digunakan untuk bahan evaluasi pengendalian kualitas bahan baku

selanjutnya. Analisis dan perancangan sistem ini menggunakan model pengembangan perangkat lunak *prototype evolusioner*. Model *prototype* cocok digunakan untuk menjabarkan kebutuhan *user* secara lebih terperinci karena *user* seringkali kesulitan menyampaikan kebutuhannya secara terperinci tanpa melihat gambaran yang jelas (Rosa dan Shalahuddin, 2014). Kelebihan model *prototype evolusioner* adalah program *prototype* yang telah dievaluasi oleh *user* dapat digunakan untuk proses selanjutnya tanpa harus dibuang, sehingga dapat mempercepat pembuatan program. Pemodelan sistem dengan UML melewati beberapa tahapan yaitu:

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

5.3.1 *Use Case Diagram*

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.



Gambar V.2 *Use Case Diagram* Sistem Informasi Pengendalian Kualitas Bahan Baku Usulan

Sumber: Hasil Analisis Data (2019)

Perancangan *use case diagram* perancangan sistem informasi pengendalian kualitas bahan baku dapat dijelaskan sebagai berikut:

1. Definisi Aktor

Pendefinisian aktor pada *use case* perancangan sistem informasi pengendalian kualitas bahan baku usulan dapat dilihat pada Tabel V.2 berikut:

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

No.	Aktor	Deskripsi
1.	User	User adalah yang melakukan <i>Login</i> , dan menginput Data Pesanan.
2.	Gudang	Gudang adalah yang melakukan <i>Login</i> , Data stok Bahan Baku, membuat Surat Jalan, monitor <i>Problem</i> , dan <i>update Status Problem</i> .
3.	QC	QC adalah yang melakukan <i>Login</i> , menginput hasil pemeriksaan kualitas bahan baku.

Sumber: Hasil Analisis Data (2019)

2. Definisi *Use Case*

Pendefinisian *use case* pada sistem informasi pengendalian kualitas bahan baku usulan dapat dilihat pada Tabel V.3 berikut:

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

No.	<i>Use Case</i>	Deskripsi
1.	<i>Login</i>	Proses untuk melakukan <i>login</i> pada aplikasi.
2.	Menginput Data Pesanan	Proses untuk melakukan penginputan bahan baku yang nantinya akan di kirimkan kepada gudang.
3.	Data Stok Bahan Baku	Proses untuk persiapan perakitan produksi.
4.	Membuat Surat Jalan	Proses membuat Surat Jalan dari gudang yang telah diterima.
5.	Monitor <i>Problem</i>	Proses untuk memonitor, serta memasukkan data kerusakan bahan baku.
6..	Update Status <i>Problem</i>	Proses untuk <i>update</i> data kerusakan bahan baku untuk dijadikan arsip.

7.	Menginput hasil pemeriksaan kualitas bahan baku	Proses untuk memasukkan data hasil pengecekan bahan baku.
----	---	---

Sumber: Hasil Analisis Data (2019)

3. Skenario *Use Case*

Skenario jalannya masing-masing *use case* pada sistem informasi pengendalian kualitas bahan baku usulan dapat dilihat pada poin berikut:

1. *Use Case Login*

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

Tabel V.4 Skenario *Use Case Login*

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

Sumber: Hasil Analisis Data (2019)

2. *Use Case Menginput Data Pesanan*

Berikut adalah skenario *use case* menginput Data Pesanan yang terdapat pada Tabel V.5:

Tabel V.5 Skenario *Use Case* Proses Menginput Data Pesanan

Nama Use Case	Menginput Data Pesanan
<i>Primary Actor</i>	<i>User</i>
<i>Use Case Description</i>	menggambarkan proses pengolahan data pesanan.
<i>Relationship</i>	<i>Association: User</i> <i>Include: - Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 1. <i>User Login</i> dan masuk ke tampilan menu utama. 2. <i>User</i> memilih menu data pesanan. 3. Sistem menampilkan data pesanan. 4. <i>User</i> melakukan proses tambah, ubah, hapus pesanan di basis data.

Sumber: Hasil Analisis Data (2019)

3. *Use Case* Data Stok Bahan Baku

Berikut adalah skenario *use case* data stok Bahan Baku yang terdapat pada Tabel V.6:

Tabel V.6 Skenario *Use Case* Proses data stok Bahan Baku

Nama Use Case	Data Stok Bahan Baku
<i>Primary Actor</i>	Gudang
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan proses pengolahan data bahan baku.
<i>Relationship</i>	<i>Association: Gudang</i> <i>Include: - Login</i>
<i>Normal Flow of Events</i>	<ol style="list-style-type: none"> 5. Gudang <i>Login</i> dan masuk ke tampilan menu utama. 6. Gudang memilih menu data bahan baku. 7. Sistem menampilkan data bahan baku. 8. <i>User</i> melakukan proses tambah, ubah, hapus bahan baku di basis data.

Sumber: Hasil Analisis Data (2019)

4. *Use Case* Membuat Surat Jalan

Berikut adalah skenario *use case* membuat surat jalan yang terdapat pada Tabel V.7:

Tabel V.7 Skenario *Use Case* Proses Membuat Surat Jalan

Nama Use Case	Memasukkan Data Surat Jalan
<i>Primary Actor</i>	Gudang
<i>Use Case Description</i>	<i>Use case</i> ini menggambarkan membuat surat jalan.
<i>Relationship</i>	<i>Association:</i> Gudang <i>Include:</i> - Login - Membuat Surat Jalan
<i>Normal Flow of Events</i>	1. Gudang <i>Login</i> dan masuk ke tampilan menu utama. 2. Gudang memilih menu transaksi, pilih sub menu surat jalan . 3. Sistem menampilkan <i>form</i> surat jalan. 4. Gudang memasukkan data surat jalan.

Sumber: Hasil Analisis Data (2019)

5. *Use Case Monitor Problem*

Berikut adalah skenario *use case* monitor *problem* yang terdapat pada Tabel V8:

Tabel V.8 Skenario *Use Case* Proses Monitor *Problem*

Nama Use Case	Monitor Problem
<i>Primary Actor</i>	Gudang
<i>Use Case Description</i>	<i>Use case</i> ini Monitor <i>Problem</i>
<i>Relationship</i>	<i>Association:</i> Gudang <i>Include:</i> - Login - Membuat <i>Problem</i>
<i>Normal Flow of Events</i>	1. Gudang <i>Login</i> , masuk ke tampilan menu utama. 2. Gudang memilih menu laporan dan menampilkan menu laporan. 3. Gudang Memilih data <i>Problem</i> yang ingin di <i>input</i> . 4. Gudang Memasukkan data yang ingin di <i>input</i> .

	5. Gudang Menampilkan data yang sudah di <i>input</i> .
--	---

Sumber: Hasil Analisis Data (2019)

6. *Use Case Update Status Problem*

Berikut adalah skenario *use case Update Status Problem* yang terdapat pada Tabel V.9:

Tabel V.9 Skenario *Use Case Update Status Problem*

Nama Use Case	<i>Update Status Problem</i>
<i>Primary Actor</i>	Gudang
<i>Use Case Description</i>	<i>Use case ini Update Status Problem</i>
<i>Relationship</i>	<i>Association:</i> Gudang <i>Include:</i> - Login - <i>Update Status Problem</i>
<i>Normal Flow of Events</i>	1. Gudang <i>Login</i> , masuk ke tampilan menu utama. 2. Gudang memilih menu laporan dan menampilkan menu laporan. 3. Gudang Memilih data <i>Problem</i> yang ingin di <i>update</i> . 4. Gudang Memasukkan data yang ingin di <i>update</i> . 5. Gudang Menampilkan data yang sudah di <i>update</i> .

Sumber: Hasil Analisis Data (2019)

7. *Use Case Menginput Hasil Pemeriksaan Kualitas Bahan Baku*

Berikut adalah skenario *use case* menginput hasil pemeriksaan kualitas bahan baku yang terdapat pada Tabel V.10:

Tabel V.10 Skenario *Use Case* Proses menginput Hasil Pemeriksaan Kualitas Bahan Baku

Nama Use Case	Menginput Hasil Pemeriksaan Kualitas Bahan Baku
<i>Primary Actor</i>	QC

<i>Use Case Description</i>	<i>Use case</i> ini Menginput hasil pemeriksaan kualitas bahan baku
<i>Relationship</i>	<i>Association:</i> QC <i>Include:</i> - Login - Membuat hasil pemeriksaan kualitas bahan baku
<i>Normal Flow of Events</i>	1. QC <i>Login</i> , masuk ke tampilan menu utama. 2. QC memilih menu laporan dan menampilkan menu laporan. 3. QC Memilih data hasil pemeriksaan kualitas bahan baku yang ingin di <i>input</i> . 4. QC Memasukkan data yang ingin di <i>input</i> . 5. QC Menampilkan data yang sudah di <i>input</i> .

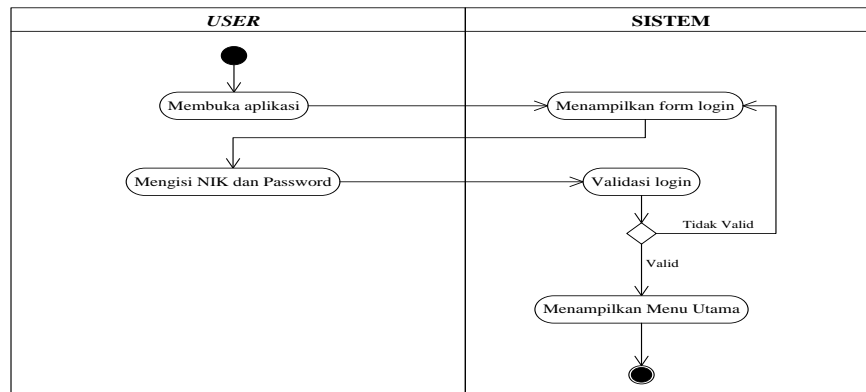
Sumber: Hasil Analisis Data (2019)

5.3.2 Activity Diagram

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

1. Activity Diagram Login

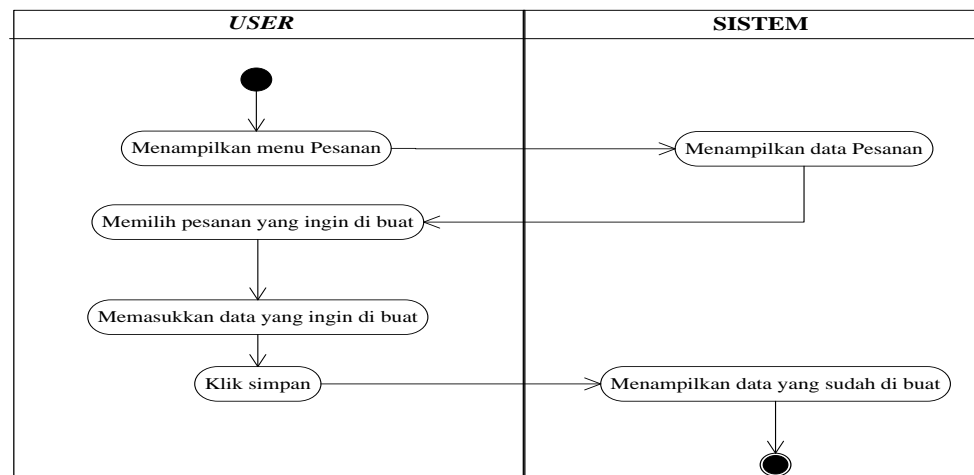
Activity diagram login ini menjelaskan alur aktivitas yang dilakukan oleh *user*, Gudang, dan QC untuk dapat masuk ke dalam Sistem Informasi Pengendalian Kualitas. *Activity diagram* dapat dilihat pada Gambar V.3 sebagai berikut:



Gambar V.3 Activity Diagram Login
Sumber: Hasil Analisis Data (2019)

2. Activity Diagram Menginput Data Pesanan

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

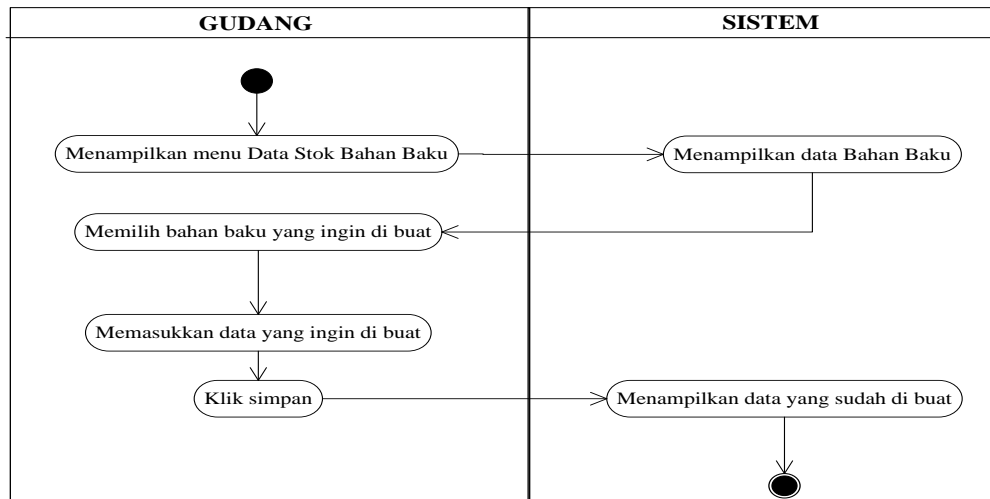


Gambar V.4 Activity Diagram Menginput Data Pesanan
Sumber: Hasil Analisis Data (2019)

3. Activity Diagram Data Stok Bahan Baku

Activity diagram menginput bahan baku berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. Activity diagram

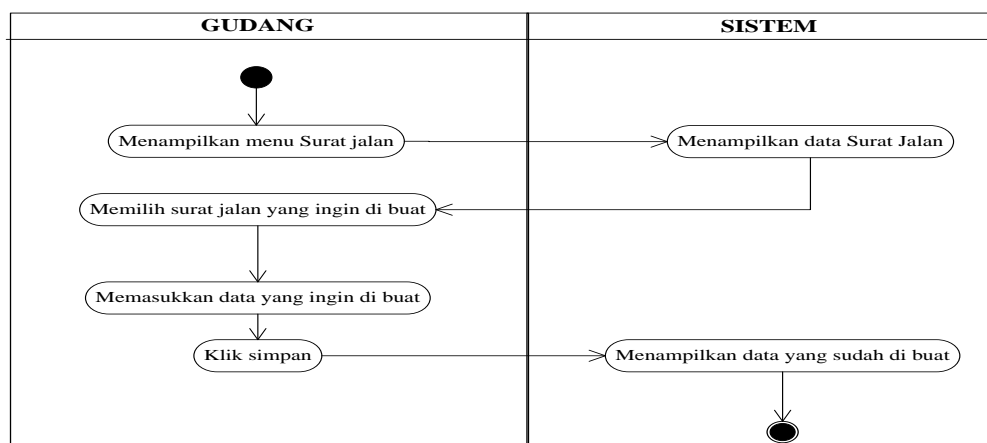
data stok bahan baku yang diusulkan dapat dilihat pada Gambar V.5 sebagai berikut:



Gambar V.5 Activity Diagram Data Stok Bahan Baku
Sumber: Hasil Analisis Data (2019)

4. Activity Diagram Menginput Surat Jalan

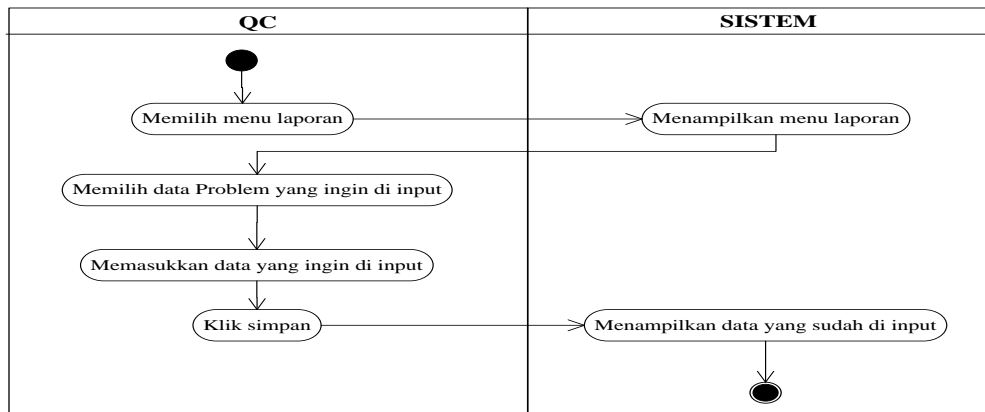
Activity diagram menginput data surat jalan berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. Activity diagram menginput surat jalan yang diusulkan dapat dilihat pada Gambar V.6 sebagai berikut:



Gambar V.6 Activity Diagram Menginput Surat Jalan
Sumber: Hasil Analisis Data (2019)

5. Activity Diagram Monitor Problem

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

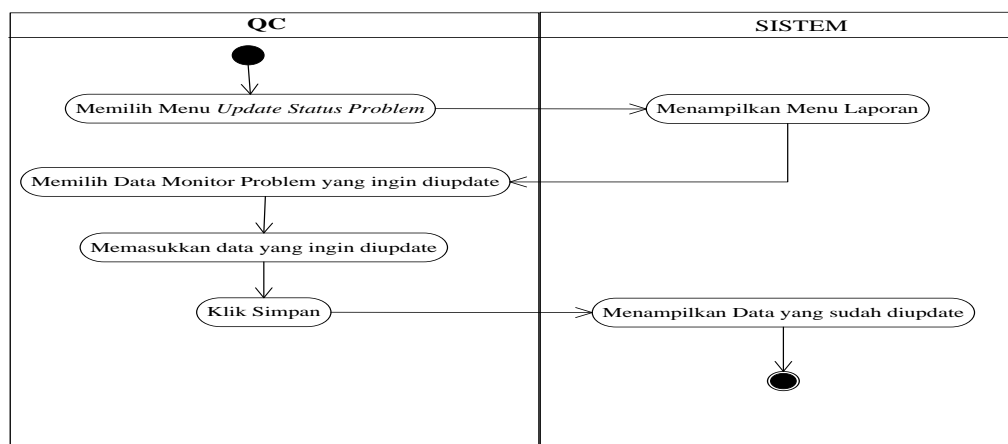


Gambar V.7 Activity Diagram Monitor Problem

Sumber: Hasil Analisis Data (2019)

6. Activity Diagram Update Status Problem

Activity Diagram Update Status Problem berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity Diagram Update Status Problem* yang diusulkan dapat dilihat pada Gambar V.8 sebagai berikut:

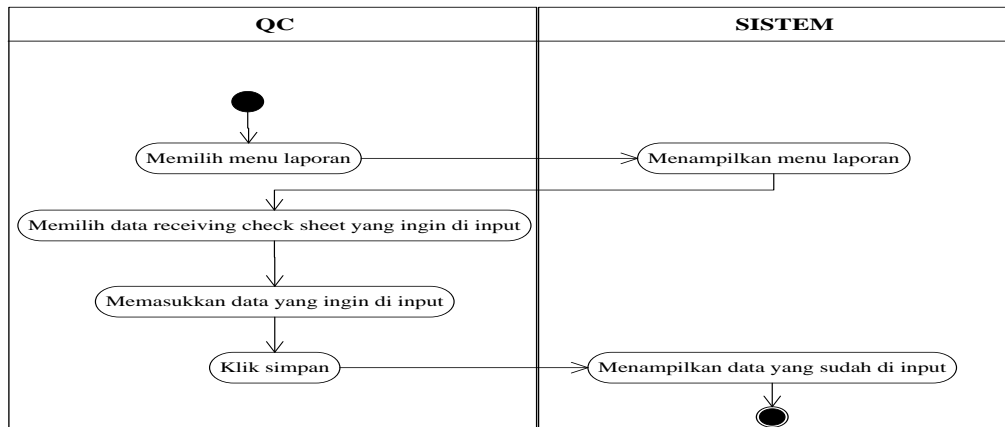


Gambar V.8 Activity Diagram Update Status Problem

Sumber: Hasil Analisis Data (2019)

7. *Activity Diagram* Menginput Hasil Pemeriksaan Kualitas Bahan Baku

Activity diagram membuat hasil pemeriksaan kualitas bahan baku berikut ini menunjukkan fungsi-fungsi yang dapat dilakukan pada aplikasi pengolahan data. *Activity diagram* menginput hasil pemeriksaan kualitas bahan baku yang diusulkan dapat dilihat pada Gambar V.9 sebagai berikut:



Gambar V.9 *Activity Diagram* Menginput Hasil Pemeriksaan Kualitas Bahan Baku

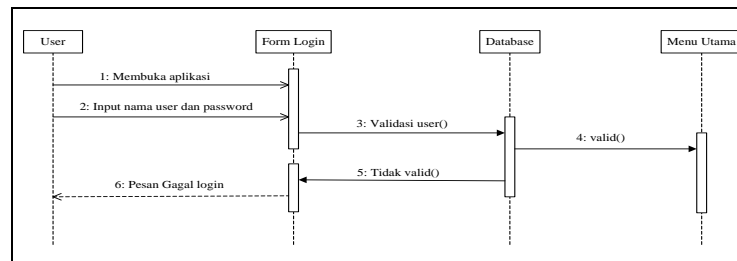
Sumber: Hasil Analisis Data (2019)

5.3.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, dimana urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada diagram. Hubungan yang ada pada gambar di bawah ini adalah proses yang dilakukan oleh sistem ketika melakukan proses yang sesuai dengan suatu objek *use case diagram*, berikut adalah *sequence diagram* pada sistem informasi pengendalian kualitas bahan baku:

1. *Sequence Diagram Login*

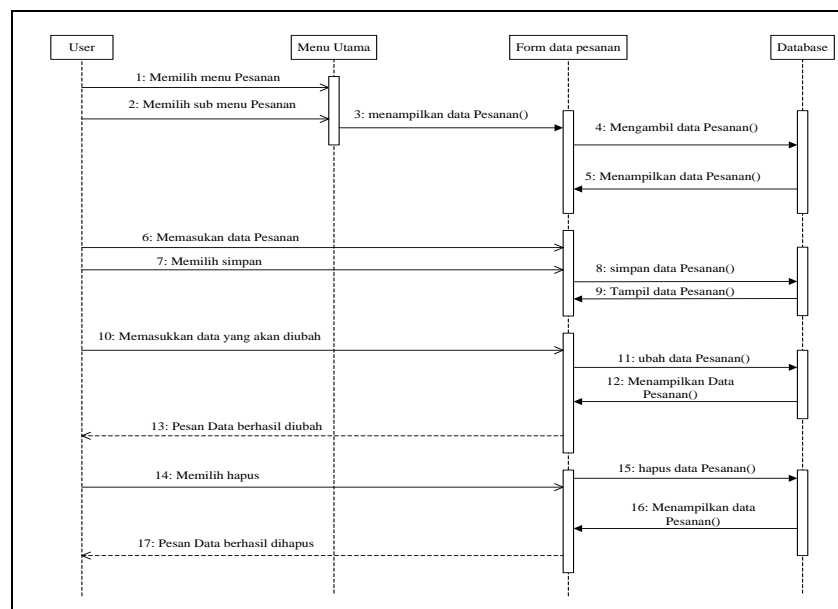
Sequence diagram login menjelaskan sebuah *sequence diagram* dalam proses *login*. Proses ini dilakukan oleh user sebelum masuk sistem. Adapun *sequence diagram* dari *use case login* dapat dilihat pada Gambar V.10.



Gambar V.10 *Sequence Diagram Login*
Sumber: Hasil Analisis Data (2019)

2. *Sequence Diagram* Menginput Data Pesanan

Sequence diagram menginput data pesanan menjelaskan sebuah *sequence diagram* dalam proses yang dimasukkan oleh *User*. Adapun *sequence diagram* dari *use case* Menginput data pesanan dapat dilihat pada Gambar V.11.

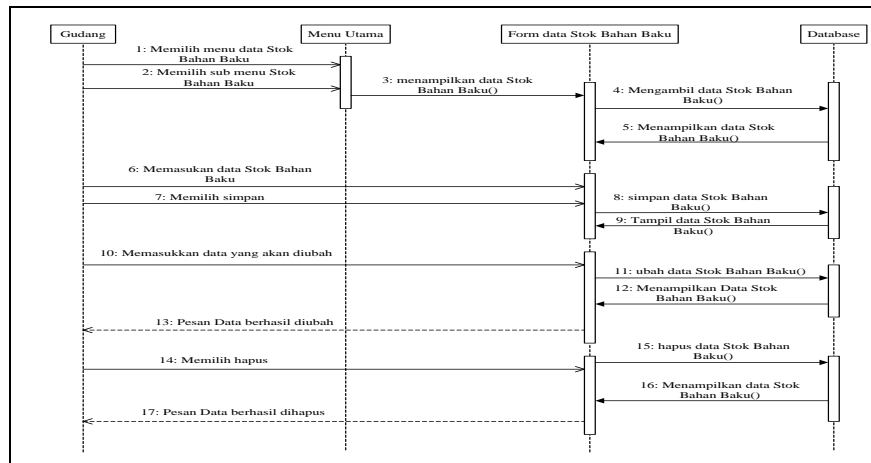


Gambar V.11 *Sequence Diagram* Menginput Data Pesanan
Sumber: Hasil Analisis Data (2019)

3. *Sequence Diagram* Data Stok Bahan Baku

Sequence diagram Data Stok Bahan Baku menjelaskan sebuah *sequence diagram* dalam proses menginput data stok bahan baku yang dimasukkan

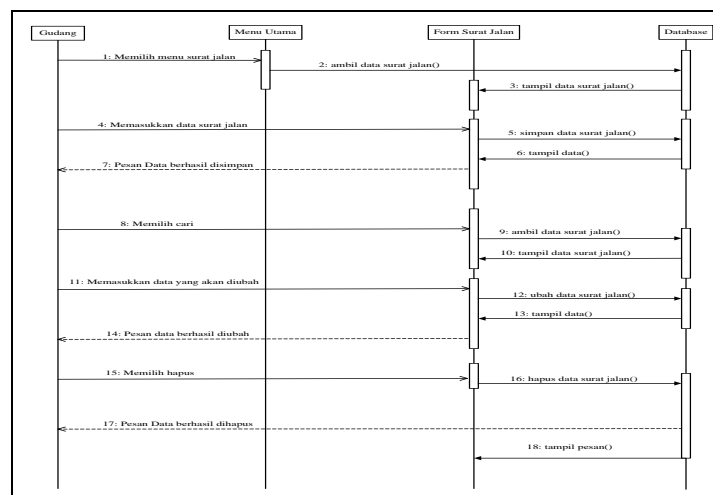
oleh Gudang. Adapun *sequence diagram* dari *use case* Mengelola Bahan Baku dapat dilihat pada Gambar V.12.



Gambar V.12 *Sequence Diagram* Data Stok Bahan Baku
Sumber: Hasil Analisis Data (2019)

4. *Sequence Diagram* Menginput Data Surat Jalan

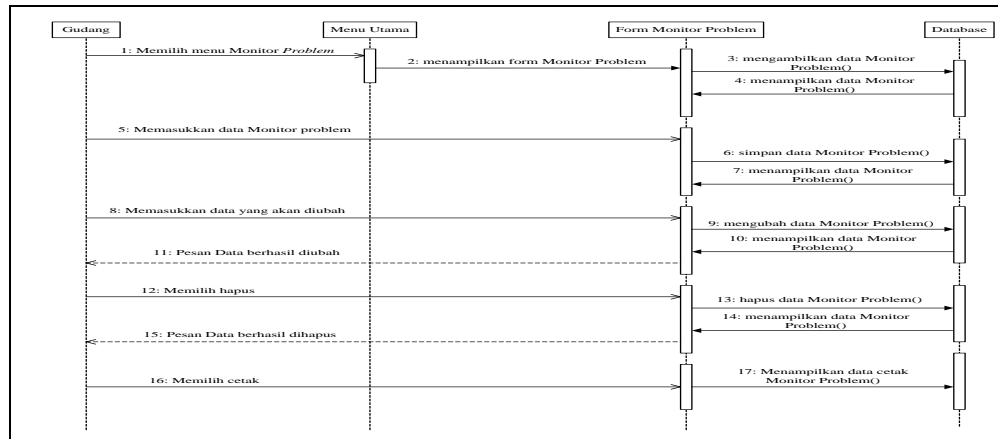
Sequence diagram menginput data surat jalan menjelaskan sebuah *sequence diagram* dalam proses transaksi menambah, menghapus, dan mengubah data surat jalan yang di masukkan oleh Gudang. Adapun *sequence diagram* dari *use case* menginput data surat jalan dapat dilihat pada Gambar V.13.



Gambar V.13 *Sequence Diagram* Menginput Data Surat Jalan
Sumber: Hasil Analisis Data (2019)

5. *Sequence Diagram Monitor Problem*

Sequence diagram Monitor Problem menjelaskan sebuah *sequence diagram* dalam proses menginput data kerusakan bahan baku yang dimasukkan oleh Gudang. Adapun *sequence diagram* dari *use case Monitor Problem* dapat dilihat pada Gambar V.14.

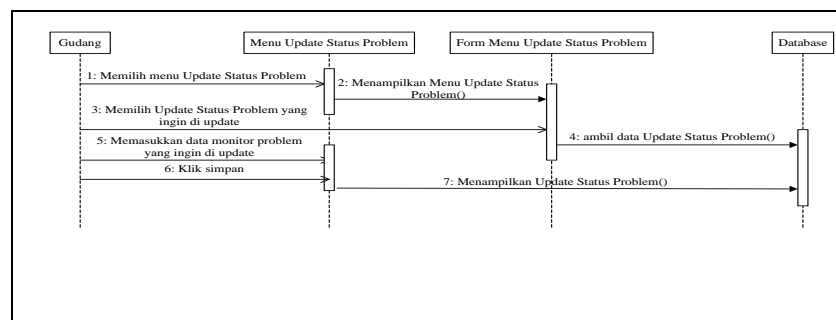


Gambar V.14 *Sequence Diagram Monitor Problem*

Sumber: Hasil Analisis Data (2019)

6. *Sequence Diagram Update Status Problem*

Sequence diagram Update Status Problem menjelaskan sebuah *sequence diagram* dalam proses *update* laporan *status problem* yang dimasukkan oleh Gudang. Adapun *sequence diagram* dari *use case Update Status Problem* dapat dilihat pada Gambar V.15.

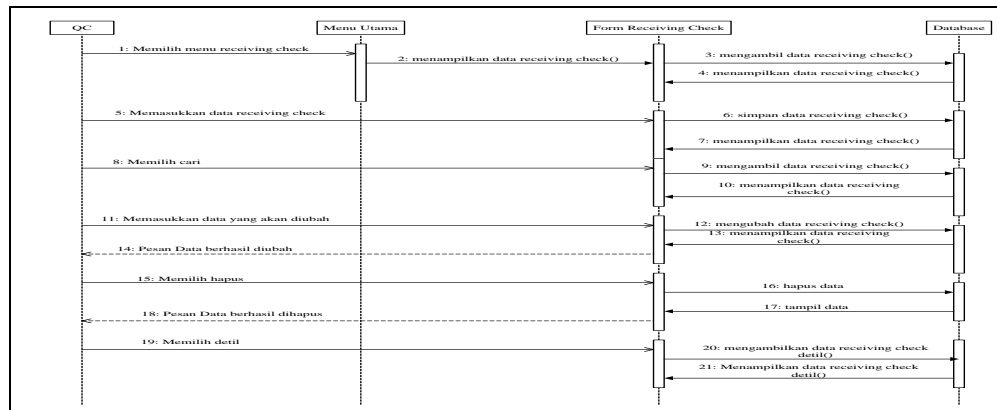


Gambar V.15 *Sequence Diagram Mengupdate Problem Sheet*

Sumber: Hasil Analisis Data (2019)

7. *Sequence Diagram Menginput hasil pemeriksaan kualitas bahan baku*

Sequence diagram menginput hasil pemeriksaan kualitas bahan baku menjelaskan sebuah *sequence diagram* dalam proses memasukkan data kualitas bahan baku yang dilakukan oleh QC. Adapun *sequence diagram* dari *use case* menginput hasil pemeriksaan kualitas bahan baku dapat dilihat pada Gambar V.16.

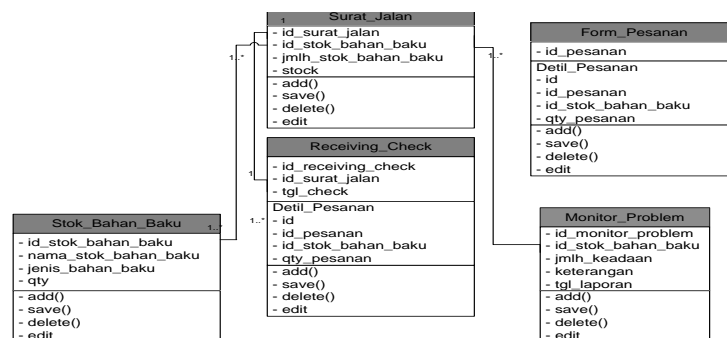


Gambar V.16 *Sequence Diagram* Membuat Hasil Pemeriksaan Kualitas Bahan Baku

Sumber: Hasil Analisis Data (2019)

5.3.4 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem informasi pengendalian kualitas bahan baku yang diusulkan dapat dilihat pada Gambar V.17 sebagai berikut:



Gambar V.17 *Class Diagram* Usulan
Sumber: Hasil Analisis Data (2019)

5.3.5 Kamus Data

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

1. Spesifikasi Tabel *User*

Nama Tabel : *User*

Fungsi : Untuk menyimpan data *user*

Tipe : File data *user*

Tabel V.11 Tabel *User*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	User	id_user	Char	5	Primary key
2.	Nama	nama_user	Varchar	25	
3.	Password	Password	Varchar	15	
4.	Kode Akses	id_akses	Char	35	Primary key
5.	Deskripsi Akses	dk_akses	Varchar	30	

Sumber: Hasil Analisis Data (2019)

2. Spesifikasi Tabel Data Pesanan

Nama Tabel : Pesanan

Fungsi : Untuk menyimpan data Pesanan

Tipe : File data admin

Tabel V.12 Tabel Pesanan

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

1.	Pesanan	kd_pesanan	<i>Char</i>	10	<i>Primary key</i>
2.	Invoice	no_invoice	<i>Varchar</i>	25	
3.	Tanggal Pesanan	tgl_pesanan	<i>Date</i>		
4.	Tanggal Terima	tgl_terima	<i>Date</i>		
5.	Supplier	kd_supplier	<i>Char</i>	5	<i>Primary key</i>
6.	<i>User</i>	id_user	<i>Char</i>	5	<i>Primary key</i>
7.	<i>Total Quality</i>	total_qty	<i>Integer</i>	11	
8.	<i>Created</i>	created_at	<i>Date</i>		
9.	<i>Updated</i>	updated_at	<i>Date</i>		

Sumber: Hasil Analisis Data (2019)

3. Spesifikasi Tabel Data Stok Bahan Baku

Nama Tabel : Bahan Baku

Fungsi : Untuk menyimpan data Bahan Baku

Tipe : File data gudang

Tabel V.13 Tabel Data Stok Bahan Baku

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kode BB	kd_bb	<i>Char</i>	10	<i>Primary key</i>
2.	Nama BB	nama_bb	<i>Varchar</i>	25	
3.	Tipe BB	tipe_bb	<i>Varchar</i>	25	
4.	Kode kategori	kd_kategori	<i>Char</i>	5	<i>Primary key</i>

Sumber: Hasil Analisis Data (2019)

4. Spesifikasi Tabel Surat Jalan

Nama Tabel : Surat Jalan

Fungsi : Untuk menyimpan data surat jalan

Tipe : File data gudang

Tabel V.14 Tabel Surat Jalan

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

1.	Kode Surat Jalan	kd_sj	Char	10	Primary key
2.	Nama Surat Jalan	nama_sj	Varchar	35	
3.	Tanggal	tgl_sj	Date		
4.	User	kd_user	Char	7	Primary key
5.	Lokasi	Lokasi	Varchar	25	
6.	Created	created_at	Date		
7.	Updated	updated_at	Date		

Sumber: Hasil Analisis Data (2019)

5. Spesifikasi Tabel Monitor *Problem*

Nama Tabel : Monitor *Problem*

Fungsi : Untuk menyimpan data Monitor *problem*

Tipe : File data QC

Tabel V.15 Tabel Monitor *Problem*

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kode pengecekan	kd_pengecekan	Char	10	Primary key
3.	Kode SJ	kd_sj	Char	10	Primary key
4.	Kode BB	kd_bb	Char	10	Primary key
5.	Nama BB	nama_bb	Varchar	25	
6.	Keterangan	Ket	Text	100	

Sumber : Hasil Analisis Data (2019)

6. Spesifikasi Tabel Hasil Pemeriksaan Kualitas Bahan Baku

Nama Tabel : Hasil Pemeriksaan Kualitas Bahan Baku

Fungsi : Untuk menyimpan data hasil pemeriksaan kualitas bahan baku

Tipe : File data QC

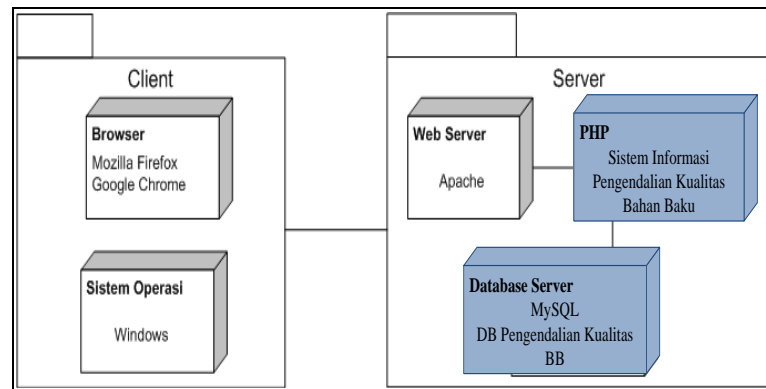
Tabel V.16 Tabel Hasil Pemeriksaan Kualitas Bahan Baku

No.	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1.	Kode Pengecekan	kd_pengecekan	<i>Char</i>	10	<i>Primary key</i>
2.	Kode Stock	id_stock	<i>Integer</i>	11	<i>Primary key</i>
3.	Kode SJ	kd_sj	<i>Char</i>	10	
4.	Kode Detil SJ	kd_dtl_sj	<i>Integer</i>	11	
5.	Kode BB	id_bb	<i>Char</i>	10	<i>Primary Key</i>
6.	<i>Quality Good</i>	Qty_good	<i>Integer</i>	11	
7.	<i>Quality Bad</i>	Qty_bad	<i>Integer</i>	11	
8.	Kode User	kd_user	<i>Char</i>	5	
9.	Kode Lvl Problem	kode_lvl_problem	<i>Char</i>	3	
10.	Status	Status	<i>Tinyinteger</i>	1	
11.	Tgl Pengecekan	tgl_pengecekan	<i>Date</i>		
12.	Keterangan	keterangan	<i>Text</i>		
13.	<i>Created</i>	<i>created_at</i>	<i>Date</i>		
14.	<i>Updated</i>	<i>updated_at</i>	<i>Date</i>		

Sumber: Hasil Analisis Data (2019)

5.3.6 Deployment Diagram

Deployment diagram digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem. *Deployment diagram* sistem informasi pengendalian kualitas bahan baku yang diusulkan dapat dilihat pada Gambar V.18.



Gambar V.18 *Deployment Diagram*

Sumber: Hasil Analisis Data (2019)

Berikut adalah penjelasan Gambar V.18 *deployment diagram* sistem informasi pengendalian kualitas bahan baku:

1. *Client* adalah komputer *client* yang harus terinstal sebuah *web browser* untuk menjalankan aplikasi sistem informasi pengendalian kualitas bahan baku dan terhubung dengan *server*.
2. *Server* aplikasi sistem informasi pengendalian kualitas bahan baku terdiri dari *web server* berupa *apache*, bahasa pemrograman PHP berbasis *web*, dan *database* MySQL.

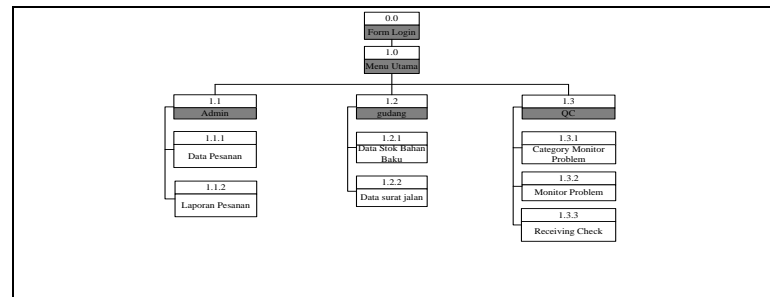
5.4 Analisis Desain Program

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

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

Hierarchy plus Input-Process-Output (HIPO) digunakan untuk mendokumentasikan sebuah struktur yang menggambarkan hubungan antar fungsi dalam program secara hierarkis. Diagram ini memuat semua modul yang ada dalam sistem beserta nama dan nomornya. Perancangan HIPO aplikasi usulan

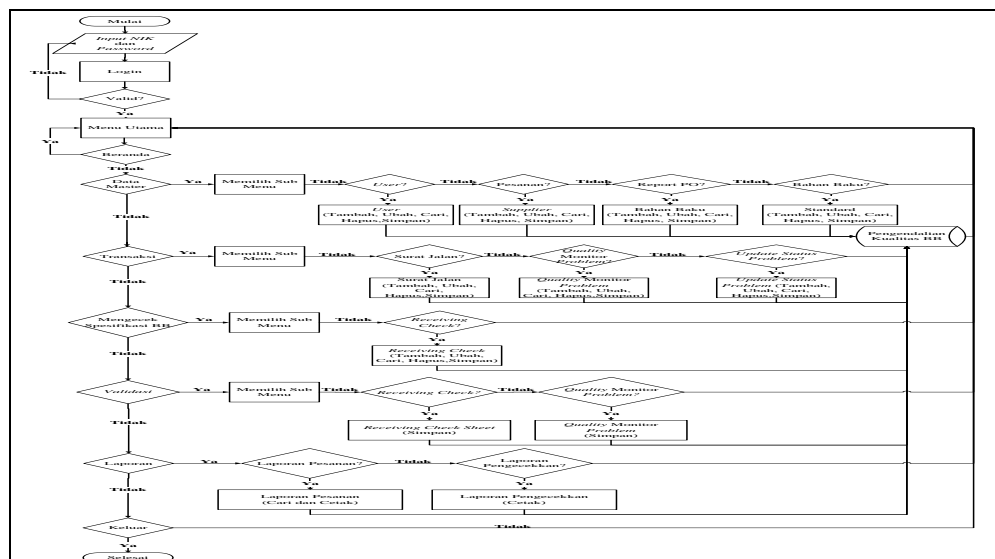
sistem pengendalian kualitas bahan baku dapat dilihat pada Gambar V.19 sebagai berikut:



Gambar V.19 HIPO Sistem Informasi Pengendalian Kualitas Bahan Baku
Sumber: Hasil Analisis Data (2019)

5.4.2 Flowchart Program

Flowchart yang digunakan untuk mendokumentasikan aplikasi sistem informasi pengendalian kualitas bahan baku ini menggunakan bagan alir logika program (*program logic flowchart*). Bagan alir ini digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika, bukan untuk menggambarkan instruksi-instruksi program komputer secara terinci. Gambar V.20 menunjukkan *program logic flowchart* aplikasi sistem informasi pengendalian kualitas bahan baku:



Gambar V.20 Program Logic Flowchart Aplikasi Sistem Informasi Pengendalian Kualitas

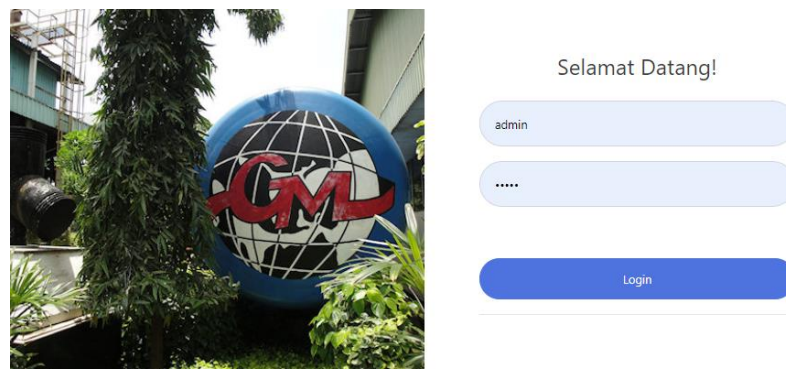
Sumber: Hasil Analisis Data (2019)

5.4.3 Perancangan *Interface* Program

Rancangan *interface* dari program pengendalian kualitas bahan baku ini adalah sebagai berikut;

1. *Form Login*

Form login adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, pengguna harus memasukkan nama pengguna dan sandi yang benar. Rancangan *form login* dapat dilihat pada Gambar V.21 berikut:



Gambar V.21 *Interface Form Login*
Sumber: Hasil Analisis Data (2019)

2. *Form Menu Utama*

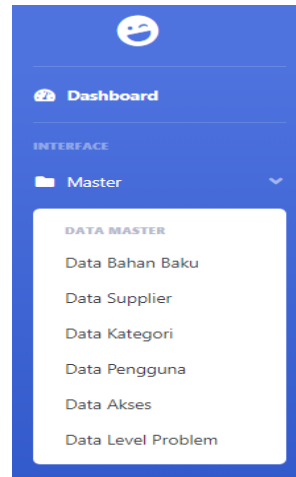
Form menu utama adalah tampilan awal pada aplikasi yang berisi empat menu. Menu tersebut adalah Beranda, Data Master, Transaksi, dan Mencetak Spesifikasi Laporan. Rancangan menu utama dapat dilihat pada Gambar V.22 berikut:



Gambar V.22 *Interface Form Menu Utama*
Sumber: Hasil Analisis Data (2019)

3. *Form Data Master User*

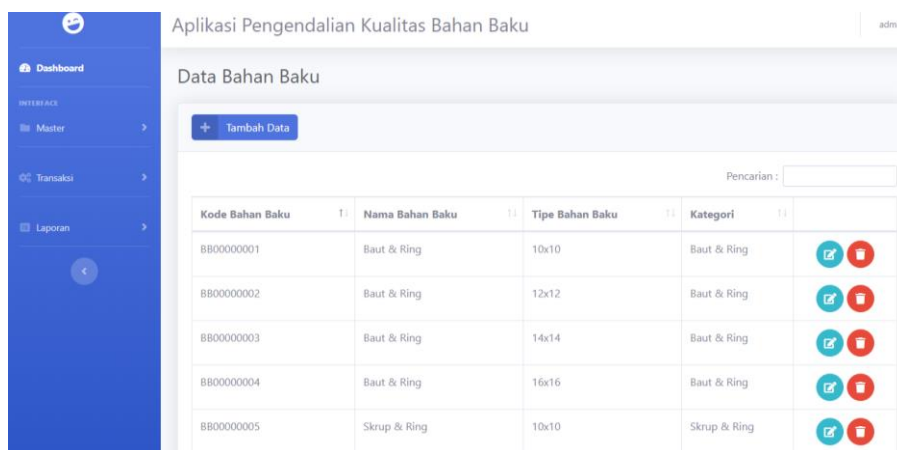
Form Data Master User adalah *form* yang digunakan untuk mengelola data *user*. Rancangan *interface* dari *form* data pengguna dapat dilihat pada Gambar V.23 berikut:



Gambar V.23 *Interface Form Data Master User*
Sumber: Hasil Analisis Data (2019)

4. *Form Data Master Data Stok Bahan Baku*

Form data master data bahan baku adalah *form* yang digunakan untuk mengelola data bahan baku. Rancangan *interface* dari *form* data bahan baku dapat dilihat pada Gambar V.24 berikut:



Gambar V.24 *Interface Form Data Stok Bahan Baku*
Sumber: Hasil Analisis Data (2019)

5. *Form Transaksi Surat Jalan*

Form transaksi surat jalan adalah *form* yang digunakan untuk mengelola data surat jalan yang diterima dari *Costumer*. Rancangan *interface* dari *form* transaksi surat jalan dapat dilihat pada Gambar V.25 berikut:

Kode Pesanan	Kode Bahan Baku	Nama Bahan Baku	Tipe Bahan Baku	Kategori	Qty	Status
P000000001	BB000000001	Baut & Ring	10x10	Baut & Ring	10	Problem
P000000002	BB000000001	Baut & Ring	10x10	Baut & Ring	50	Problem
Total:						

Gambar V.25 *Interface Form* Transaksi Surat Jalan
Sumber: Hasil Analisis Data (2019)

6. *Form* Transaksi Monitor Problem

Form transaksi monitor *problem* adalah *form* yang digunakan untuk mengelola data kerusakan bahan baku yang dicek secara visual. Rancangan *interface* dari *form* transaksi monitor *problem* dapat dilihat pada Gambar V.26 berikut:

Kode Pengecekan	Kode Surat Jalan	Nama Bahan Baku	Keterangan	Status
C000000001	2019061901	Baut & Ring	Good	OK
C000000002	2019070201	Baut & Ring	Bahan baku kurang	Problem
C000000003	2019070201	Baut & Ring	Good	OK

Gambar V.26 *Interface Form* Monitor Problem
Sumber: Hasil Analisis Data (2019)

Pada *interface form* transaksi monitor *problem* terdapat dua tombol, yaitu edit, dan simpan. Tombol simpan digunakan untuk menyimpan data monitor

problem ke dalam *database*. Tombol edit digunakan untuk mengedit keterangan pada data monitor *problem*.

7. *Form* Transaksi Hasil Pemeriksaan Kualitas Bahan Baku

Form transaksi hasil pemeriksaan kualitas bahan baku adalah *form* yang digunakan untuk mengelola data bahan baku yang dicek secara visual. Rancangan *interface* dari *form* transaksi hasil pemeriksaan kualitas bahan baku dapat dilihat pada Gambar V.27 berikut:



Kode Surat Jalan	Kode Bahan Baku	Nama Bahan Baku	Tanggal Surat Jalan	Qty	
2019061901	BB00000001	Baut & Ring	2019-06-20	15	Lihat Data
2019070201	BB00000001	Baut & Ring	2019-07-02	10	Lihat Data
2019070201	BB00000001	Baut & Ring	2019-07-02	50	Lihat Data

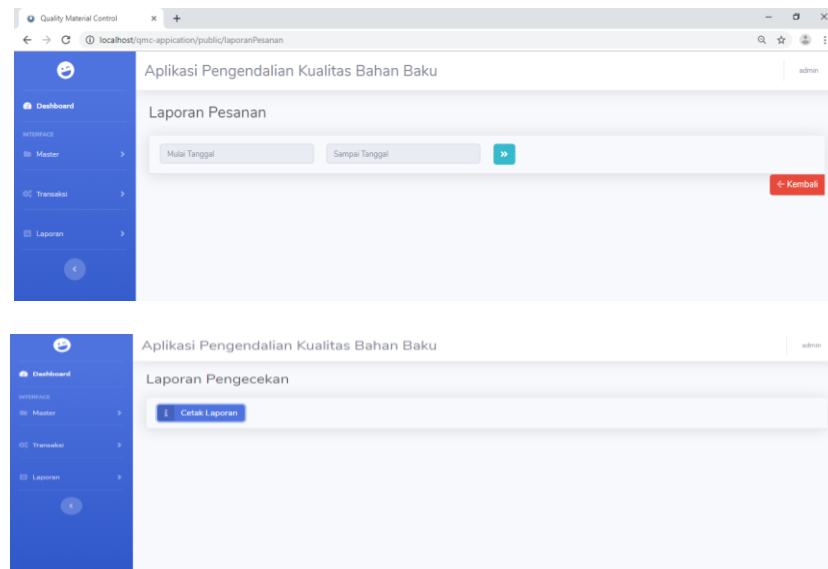
Gambar V.27 *Interface Form* Hasil Pemeriksaan Kualitas Bahan Baku
Sumber: Hasil Analisis Data (2019)

Pada *form* transaksi hasil pemeriksaan kualitas bahan baku ini terdapat dua tombol, yaitu tombol lihat data, dan kembali. Tombol lihat data digunakan untuk melihat data hasil pemeriksaan kualitas bahan baku ke dalam *database*. Tombol kembali digunakan untuk membatalkan melihat hasil pengecekan data hasil pemeriksaan kualitas bahan baku.

8. *Form* Laporan Pengendalian Kualitas Bahan Baku

Form laporan pengendalian kualitas bahan baku adalah *form* yang digunakan untuk melihat data kualitas dari bahan baku. Pada *form* laporan pesanan terdapat empat tombol yaitu, cari tanggal, tambah, kembali dan cetak. Pada *form* laporan pengecekan terdapat satu tombol yaitu, cetak. Dimana tombol cari tanggal digunakan untuk mencari laporan pesanan berdasarkan tanggal,

tombol tambah digunakan untuk menambah data laporan pesanan. Dan cetak digunakan untuk mencetak laporan pesanan. Sedangkan laporan pengecekan, hanya digunakan untuk mencetak laporan pengecekan. Rancangan *interface* dari *form* laporan pengendalian kualitas dapat dilihat pada Gambar V.28 berikut:



Gambar V.28 *Interface Form* Laporan Pengendalian Kualitas Bahan Baku
Sumber: Hasil Analisis Data (2019)

5.5 Implementasi Sistem *Software* dan *Hardware*

Tahap ini adalah tahap pengkodean program menggunakan bahasa pemrograman PHP dengan *Macromedia Dreamweaver 8* dan *Sublime* 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 Wndows 10*
- b. *Database Server* : MySQL versi 10.1.35
- c. Bahasa Pemrograman : PHP 7.2.9

d. *Web Browser* : *Google Chrome*

2. Analisis Kebutuhan *Hardware*

a. *Processor* : Minimal *Processor Pentium IV*

b. *RAM* : Minimal *RAM 512 MB*

c. *Harddisk* : Minimal *Harddisk 64 GB*

d. *Mouse, Keyboard, Monitor* sebagai peralatan antar muka.

e. *Printer* sebagai media pencetakan dokumen.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

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

1. Aplikasi sistem informasi pengendalian kualitas bahan baku ini bertujuan untuk memudahkan pembuatan hasil pemeriksaan kualitas bahan baku sehingga mencegah terjadinya kerusakan dan kehilangan data hasil inspeksi atas kualitas bahan baku.
2. Dengan sistem ini, data-data hasil inspeksi kualitas bahan baku dapat tersimpan dengan aman dan mudah di dalam basis data.
3. Dengan sistem ini, data hasil pemeriksaan kualitas bahan baku dapat diolah dengan mudah untuk dijadikan laporan data pengendalian kualitas. Sehingga acuan data inspeksi kualitas bahan baku untuk periode berikutnya lebih akurat.

6.2. Saran

Saran yang diberikan untuk pengembangan sistem informasi pengendalian kualitas bahan baku ini selanjutnya adalah sebagai berikut:

1. Untuk penerapan sistem baru terhadap sistem lama, sebaiknya dilakukan secara bertahap dan diperlukan sosialisasi penggunaan sistem ini kepada bagian yang terkait.
2. Sebaiknya dilakukan pemeliharaan aplikasi secara berkala, sehingga aplikasi dapat berjalan dengan baik.

DAFTAR PUSTAKA

- Amsyah, Zulkifli. 2005. *Manajemen Sistem Informasi*. PT Gramedia Pustaka Utama. Jakarta.
- Assauri, Sofjan. 1998. *Manajemen Operasi dan Produksi*. LP FE UI Dwiwinarno. Jakarta.
- Darmawan, Deni. 2007. *Teknologi Informasi dan Komunikasi*. Bandung. Arum Mandiri Press.
- Dharwiyanti, Sri. 2003. *Pengantar Unified Modeling Language (UML)*. IlmuKomputer.com.
- Eddy, Prahasta. 2005. *Konsep-Konsep Dasar Sistem Informasi Geografis*. Bandung: CV Informatika.
- Febriani. 2015. *Flowchart*. Febriani.staff.gunadarma.ac.id/Download/files/5616/flowchart.pdf. (Tanggal Akses: 5 April 2016).
- Gaspersz, Vincent. 2005. *Total Quality Management*. PT Gramedia Pusaka Utama. Jakarta.
- Hadi, Sutrisno. 1987. *Metodologi Research Jilid 2 / Sutrisno Hadi*. Andi. Yogyakarta.
- Hall, J.A. 2001. *Sistem Informasi Akuntansi*. Buku 1. Jakarta: Salemba Empat.
- Jogiyanto. 1999. *Analisis dan Desain Sistem Informasi, Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Andi. Yogyakarta.
- Jogiyanto. 2005. *Sistem Informasi Teknologi Edisi II. Pendekatan Terintegrasi Konsep Dasar Teknologi, Aplikasi Pengembangan dan Pengelolaannya*. Kanisius. Yogyakarta.
- Komaruddin. 1997. *Analisis dan Design Sistem Informasi*. Surabaya.

- Kristanto, Andri. 2008. *Perancangan Sistem Informasi*. Gava Media. Yogyakarta.
- McLeod, Raymond. 2001. *Sistem Informasi Manajemen, PT Prenhallindo*. Jakarta.
- Mustakini, Jogiyanto Hartono. 2005. *Analisis dan Desain*. Andi. Yogyakarta.
- Mulyadi. 2015. *Akuntansi Biaya*. Yogyakarta: UPP-STIM YKPN
- Nugroho, Widjajanto. 2008. *Sistem Informasi Akuntansi*. Jakarta. Erlangga.
- O'brien. 2005. *Pengantar Sistem Informasi: Perspektif Bisnis dan Manajerial*. Jakarta: Salemba Empat.
- Rama, D dan Jones, F. 2008. *Sistem Informasi Akuntansi*. Salemba.
- Rosa, A.S. dan M. Shalahuddin. 2011. *Modul Pembelajaran Rekayasa Perangkat Lunak*. Cetakan Pertama. Bandung: Modula.
- Rosa, A.S. dan M. Shalahuddin. 2014. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Edisi Kedua. Bandung : Informatika
- Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak)* Edisi-6. Jakarta. Erlangga.
- Sugiyono. 2013. *Metodologi Penelitian Manajemen*. Alfabeta. Bandung.
- Sutaji, Deni. 2012. "*Sistem Inventory Mini Market dengan PHP dan JQuery*". Yogyakarta : Lokomedia
- Syamsi, Ibnu, 1983, *Administrasi Perlengkapan Materil Pemerintahan Daerah*, Bina Aksara, Jakarta
- Welling, L. dan Thomson. 2003. *PHP and MySql Web Development, Second Edition*. Indianapolis : Sams Publishing
- Welsch, Glenn A, Ronald W. Hilton, 2003, *Anggaran, Perencanaan dan Pengendalian, alih bahasa Purwatiningsih*, Salemba Empat, Jakarta.
- _____. 2002. *Kamus Besar Bahasa Indonesia*. Jakarta: Balai Pustaka.

- _____. 2000. *UML Activity Diagrams: References*. <http://msdn.microsoft.co/en-us/library/dd409360.aspx>. (Tanggal Akses: 5 April 2016).
- _____. *UML Diagrams*, <http://uml-diagrams.org>. (Tanggal Akses: 20 Mei 2016)
- _____. *QC 7 Tools*, <http://thesis.binus.ac.id/Asli/Bab2/2007-1-00262-TI%20-Bab%202.pdf>. (Tanggal Akses: 11 November 2014)

LAMPIRAN A

TAMPILAN PROGRAM

1. Login



Selamat Datang!

admin

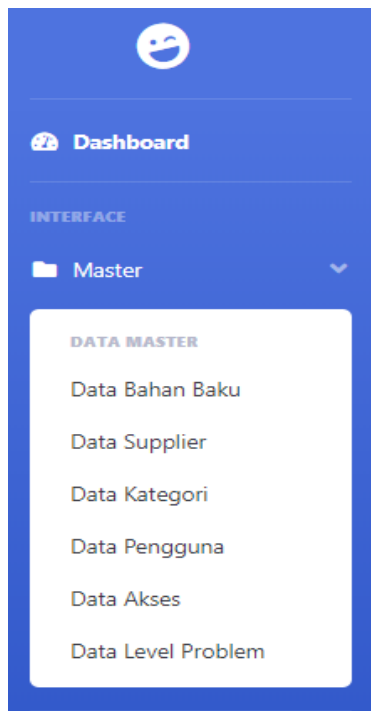
.....

Login

2. Menu Utama



3. Data Master *User*



4. Data Stok Bahan Baku

Dashboard

INTERFACE

Master

Transaksi

Laporan

Aplikasi Pengendalian Kualitas Bahan Baku

admin

Data Bahan Baku

+ Tambah Data

Pencarian :

Kode Bahan Baku	Nama Bahan Baku	Tipe Bahan Baku	Kategori	
BB00000001	Baut & Ring	10x10	Baut & Ring	
BB00000002	Baut & Ring	12x12	Baut & Ring	
BB00000003	Baut & Ring	14x14	Baut & Ring	
BB00000004	Baut & Ring	16x16	Baut & Ring	
BB00000005	Skrup & Ring	10x10	Skrup & Ring	

5. Transaksi Surat Jalan

Aplikasi Pengendalian Kualitas Bahan Baku

admin

Data Surat Jalan

Kode Surat Jalan: 2019070201 Tanggal Surat Jalan: 02-07-2019

Lokasi: AL Bunga Kumis Kuning Pembuat Surat Jalan: admin

Data Bahan Baku

Kode Pesanan... Bahan Baku... Qty...

Kode Pesanan	Kode Bahan Baku	Nama Bahan Baku	Tipe Bahan Baku	Kategori	Qty
P000000001	BB00000001	Baut & Ring	10x10	Baut & Ring	10
P000000002	BB00000001	Baut & Ring	10x10	Baut & Ring	50
Total :					

Data Mulai Dari 1 Sampai 2 , Total Data: 2

Sebelumnya 1 Berikutnya

[Kembali](#) [Cetak Ulang](#)

6. Monitor Problem

Aplikasi Pengendalian Kualitas Bahan Baku

admin

Data Problem

Pencarian :

Kode Pengecekan	Kode Surat Jalan	Nama Bahan Baku	Keterangan
C000000001	2019061901	Baut & Ring	Good
C000000002	2019070201	Baut & Ring	Bahan baku kurang
C000000003	2019070201	Baut & Ring	Good

Data Mulai Dari 1 Sampai 3 , Total Data: 3

Sebelumnya 1 Berikutnya

Copyright © by Galih Asmoro, Politeknik STMI Jakarta 2019

7. Receiving Check

Aplikasi Pengendalian Kualitas Bahan Baku

admin

Data Pengecekan

Pencarian :

Kode Surat Jalan	Kode Bahan Baku	Nama Bahan Baku	Tanggal Surat Jalan	Qty
2019061901	BB00000001	Baut & Ring	2019-06-20	15
2019070201	BB00000001	Baut & Ring	2019-07-02	10
2019070201	BB00000001	Baut & Ring	2019-07-02	50

Data Mulai Dari 1 Sampai 3 , Total Data: 3

Sebelumnya 1 Berikutnya

8. Laporan Pesanan

Quality Material Control x +

localhost/qmc-application/public/laporanPesanan

Aplikasi Pengendalian Kualitas Bahan Baku admin

Laporan Pesanan

Mulai Tanggal Sampai Tanggal »

← Kembali

9. Laporan Pengecekan

Quality Material Control x +

localhost/qmc-application/public/laporanPengecekan

Aplikasi Pengendalian Kualitas Bahan Baku admin

Laporan Pengecekan

i Cetak Laporan

LAMPIRAN B

KODE PROGRAM

1. Login

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
  <meta name="csrf-token" content="{{ csrf_token() }}">

  <title>Login</title>

  <link rel="icon" type="image/png" href="img/login_icon.png"/>
  <!-- Custom fonts for this template-->
  <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet"
type="text/css">
  <link
href="https://fonts.googleapis.com/css?family=Nunito:200,200i,300,300i,400,400
i,600,600i,700,700i,800,800i,900,900i" rel="stylesheet">

  <!-- Custom styles for this template-->
  <link href="css/sb-admin-2.min.css" rel="stylesheet">

  <style type="text/css">
```

```

body {
    background-image: url("img/body_background.jpg");
}

.error{
    color: red;
    font-size: 12px;
    font-weight: bold;
    margin-top: 0.5em;
    margin-left: 1em;
}
</style>
</head>

<body>

<div class="container">
    <!-- Outer Row -->
    <div class="row justify-content-center">
        <div class="col-xl-10 col-lg-12 col-md-9">
            <div class="card o-hidden border-0 shadow-lg my-5">
                <div class="card-body p-0">
                    <!-- Nested Row within Card Body -->
                    <div class="row">
                        <div class="col-lg-6 d-none d-lg-block bg-login-image"></div>
                        <div class="col-lg-6">
                            <div class="p-5">
                                <div class="text-center">
                                    <h1 class="h4 text-gray-900 mb-4">Selamat
Datang!</h1>
                                </div>

```

```
<form class="user" name="formLogin" id="formLogin">
  {{ csrf_field() }}
  <div class="form-group">
    <input type="text" class="form-control form-control-
user" name="username" id="username" aria-describedby="emailHelp"
placeholder="Masukkan Username...">

    <div class="error" id="errorUser"></div>
  </div>
  <div class="form-group">
    <input type="password" class="form-control form-
control-user" name="password" id="password" placeholder="Masukkan Kata
Sandi..">

    <div class="error" id="errorPwd"></div>
  </div>
  <div class="form-group">
    <div class="custom-control custom-checkbox small">
      <input type="checkbox" class="custom-control-
input" id="customCheck">

    </div>
  </div>
  <button class="btn btn-primary btn-user btn-block"
id="submitLogin">

    Login
  </button>
</form>
<hr>
</div>
</div>
</div>
</div>
```

</div>

</div>

</div>

<!-- Bootstrap core JavaScript-->

<script src="vendor/jquery/jquery.min.js"></script>

<script src="vendor/jquery/jquery.validate.min.js"></script>

<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="vendor/sweetalert/dist/sweetalert.min.js"></script>

<!-- Core plugin JavaScript-->

<script src="vendor/jquery-easing/jquery.easing.min.js"></script>

<!-- Custom scripts for all pages-->

<script src="js/sb-admin-2.min.js"></script>

<script>

\$(document).ready(function() {

});

\$('#submitLogin').click(function(e){

\$('#errorUser').html("");

\$('#errorPwd').html("");

e.preventDefault();

var form = \$('#formLogin');

var username = \$('#username').val();

var password = \$('#password').val();

if (username == "" && password == ""){

\$('#errorUser').html('* Username tidak boleh kosong');

```

        $("#errorPwd").html('* Kata sandi tidak boleh kosong');
    }else if (username == null || username == ""){
        $("#errorUser").html('* Username tidak boleh kosong');
        }else if (password == null || password == ""){
            $("#errorPwd").html('* Kata sandi tidak boleh
kosong');
        }else{
            jQuery.ajax({
                type:"POST",
                url: "{{ url('loginApp') }}",
                headers: {
                    'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
                },
                data: form.serialize(),
                beforeSend: function(){
                    swal({
                        title: "Loading...",
                        text: "Tunggu Sebentar..!",
                        icon:
'{{ url("img/loading.gif") }}',
                        button: false
                    });
                },
                success: function(data) {
                    if(data.success == true)
                    {
                        window.location.href = "{{
url('/dashboard') }}"
                    }
                    else
                    {

```

```

swal("Error", data.msg,
"error");
    }
    }
});
}
});
</script>

```

```
</body>
```

```
</html>
```

2. Menu Utama

```
<!doctype html>
```

```
<html lang="{ { str_replace('_', '-', app()->getLocale()) } }">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<title>Laravel</title>
```

```
<!-- Fonts -->
```

```
<link href="https://fonts.googleapis.com/css?family=Nunito:200,600"
rel="stylesheet">
```

```
<!-- Styles -->
```

```
<style>
```

```
html, body {
```

```
background-color: #fff;
```

```
color: #636b6f;
```

```
font-family: 'Nunito', sans-serif;
```

```
    font-weight: 200;  
    height: 100vh;  
    margin: 0;  
}
```

```
.full-height {  
    height: 100vh;  
}
```

```
.flex-center {  
    align-items: center;  
    display: flex;  
    justify-content: center;  
}
```

```
.position-ref {  
    position: relative;  
}
```

```
.top-right {  
    position: absolute;  
    right: 10px;  
    top: 18px;  
}
```

```
.content {  
    text-align: center;  
}
```

```
.title {  
    font-size: 84px;
```



```
}
```

```
.links > a {  
  color: #636b6f;  
  padding: 0 25px;  
  font-size: 13px;  
  font-weight: 600;  
  letter-spacing: .1rem;  
  text-decoration: none;  
  text-transform: uppercase;  
}
```

```
.m-b-md {  
  margin-bottom: 30px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="flex-center position-ref full-height">
```

```
  @if (Route::has('login'))
```

```
    <div class="top-right links">
```

```
      @auth
```

```
        <a href="{{ url('/home') }}">Home</a>
```

```
      @else
```

```
        <a href="{{ route('login') }}">Login</a>
```

```
      @if (Route::has('register'))
```

```
        <a href="{{ route('register') }}">Register</a>
```

```
      @endif
```

```
    @endauth
```

```
  </div>
```

```

@endif

<div class="content">
    <div class="title m-b-md">
        Laravel
    </div>

    <div class="links">
        <a href="https://laravel.com/docs">Docs</a>
        <a href="https://laracasts.com">Laracasts</a>
        <a href="https://laravel-news.com">News</a>
        <a href="https://blog.laravel.com">Blog</a>
        <a href="https://nova.laravel.com">Nova</a>
        <a href="https://forge.laravel.com">Forge</a>
        <a href="https://github.com/laravel/laravel">GitHub</a>
    </div>
</div>
</body>
</html>

```

3. Data Master User

```

@extends('layout.main')

@section('title','Data User')

@section('section')
    <div class="card shadow">
        <div class="card-header py-3">
            <button href="#" class="btn btn-primary btn-icon-split" data-
toggle="modal" data-target="#modalUser-A"
onClick="getAksesList('{{ url("listAkses") }}')">
                <span class="icon text-white-50">

```

```

        <i class="fas fa-plus"></i>
    </span>
    <span class="text">Tambah Data</span>
</button>
</div>
<div class="card-body">
    <div class="table-responsive">
        <table class="table table-bordered" id="tableUser" width="100% "
cellspacing="0">
            <thead>
                <tr>
                    <th>Kode User</th>
                    <th>Nama</th>
                    <th>Username</th>
                    <th>Kata Sandi</th>
                    <th>Kode Akses</th>
                    <th>Deskripsi Akses</th>
                    <th></th>
                </tr>
            </thead>
        </table>
    </div>
</div>
</div>

```

```

<!-- MODALS Add User -->
<div class="modal" id="modalUser-A" tabindex="-1" role="dialog" aria-
labelledby="defModalHead" aria-hidden="true">
    <div class="modal-dialog modal-md">
        <div class="modal-content">
            <div class="modal-header">

```

```

        <h5 class="modal-title" id="defModalHead">Tambah User</h5>
        <button type="button" class="close" data-dismiss="modal"><span
aria-hidden="true">&times;</span><span class="sr-
only">Close</span></button>
    </div>
    <div class="modal-body">
        <form class="form-group" name="formUserAdd"
id="formUserAdd">
            {{ csrf_field() }}
            <div class="form-group row">
                <label class="col-md-5">Kode User</label>
                <div class="col-md-7">
                    <input type="text" name="userID" id="newUserID"
class="form-control" value="" readonly>
                </div>
            </div>
            <div class="form-group row">
                <label class="col-md-5">Nama</label>
                <div class="col-md-7">
                    <input type="text" name="name" id="newName"
class="form-control" value="" style="text-transform: capitalize;">
                </div>
            </div>
            <div class="form-group row">
                <label class="col-md-5">Username</label>
                <div class="col-md-7">
                    <input type="text" name="username" id="newUsername"
class="form-control" value="">
                </div>
            </div>
            <div class="form-group row">

```

```

        <label class="col-md-5">Kata Sandi</label>
        <div class="col-md-7">
            <input type="password" name="password"
id="newPassword" class="form-control" value="">
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Deskripsi Akses</label>
        <div class="col-md-7">
            <select class="form-control" name="access" id="newAccess"
required>
                </select>
            </div>
        </div>
    </form>
</div>
<div class="modal-footer">
    <div class="btn-group pull-right">
        <button class="btn btn-primary"
onclick="storeUser('{{ url("storeUser") }}','{{ url("img/loading.gif") }}','Insert')
"><span class="fa fa-save"></span> Save</button>
    </div>
</div>
</div>
</div>
<!-- MODALS Add User -->

<!-- MODALS Edit User -->
<div class="modal" id="modalUser-E" tabindex="-1" role="dialog" aria-
labelledby="defModalHead" aria-hidden="true">

```

```

<div class="modal-dialog modal-md">
  <div class="modal-content">
    <div class="modal-header">
      <h4 class="modal-title" id="defModalHead">Edit User</h4>
      <button type="button" class="close" data-dismiss="modal"><span
aria-hidden="true">&times;</span><span class="sr-
only">Close</span></button>
    </div>
    <div class="modal-body">
      <form class="form-group" name="formUserEdit"
id="formUserEdit">
        {{ csrf_field() }}
        <div class="form-group row">
          <label class="col-md-5">Kode User</label>
          <div class="col-md-7">
            <input type="text" name="userID" id="userID" class="form-
control" value="" readonly>
          </div>
        </div>
        <div class="form-group row">
          <label class="col-md-5">Nama</label>
          <div class="col-md-7">
            <input type="text" name="name" id="name" class="form-
control" value="" style="text-transform: capitalize;">
          </div>
        </div>
        <div class="form-group row">
          <label class="col-md-5">Username</label>
          <div class="col-md-7">
            <input type="text" name="username" id="username"
class="form-control" value="">

```

```

        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Kata Sandi</label>
        <div class="col-md-7">
            <input type="password" name="password" id="password"
class="form-control" value="">
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Deskripsi Akses</label>
        <div class="col-md-7">
            <select class="form-control" name="access" id="access"
required>
                </select>
        </div>
    </div>
</form>
</div>
<div class="modal-footer">
    <div class="btn-group pull-right">
        <button class="btn btn-primary"
onclick="storeUser('{{ url("updateUser") }}','{{ url("img/loading.gif") }}','Edit')
"><span class="fa fa-save"></span> Save</button>
    </div>
</div>
</div>
</div>
<!-- MODALS Edit Kategori -->
@endsection

```

4. Data Stok Bahan Baku

@extends ('layout.main')

@section('title','Data Bahan Baku')

@section('section')

```
<div class="card shadow">
  <div class="card-header py-3">
    <button href="#" class="btn btn-primary btn-icon-split" data-
toggle="modal" data-target="#modalBB-A"
onClick="getCatList('{{url("listKategori")}}')">
      <span class="icon text-white-50">
        <i class="fas fa-plus"></i>
      </span>
      <span class="text">Tambah Data</span>
    </button>
  </div>
  <div class="card-body">
    <div class="table-responsive">
      <table class="table table-bordered" id="tableBahanBaku"
width="100%" cellpadding="0">
        <thead>
          <tr>
            <th>Kode Bahan Baku</th>
            <th>Nama Bahan Baku</th>
            <th>Tipe Bahan Baku</th>
            <th>Kode Kategori</th>
            <th>Kategori</th>
            <th></th>
          </tr>
        </thead>
      </table>
```



```

        </div>
    </div>
</div>

<!-- MODALS Add BB -->
<div class="modal" id="modalBB-A" tabindex="-1" role="dialog" aria-
labelledby="defModalHead" aria-hidden="true">
    <div class="modal-dialog modal-md">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="defModalHead">Tambah Bahan
                Baku</h5>
                <button type="button" class="close" data-dismiss="modal"><span
                aria-hidden="true">&times;</span><span class="sr-
                only">Close</span></button>
            </div>
            <div class="modal-body">
                <form class="form-group" name="formBBAdd" id="formBBAdd">
                    {{ csrf_field() }}
                    <div class="form-group row">
                        <label class="col-md-5">Kode Bahan Baku</label>
                        <div class="col-md-7">
                            <input type="text" name="bbID" id="newBBID"
                            class="form-control" value="" readonly>
                        </div>
                    </div>
                    <div class="form-group row">
                        <label class="col-md-5">Nama Bahan Baku</label>
                        <div class="col-md-7">
                            <input type="text" name="bbName" id="newBBName"
                            class="form-control" value="" style="text-transform: capitalize;">

```

```

        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Tipe Bahan Baku</label>
        <div class="col-md-7">
            <input type="text" name="bbType" id="newBBType"
class="form-control" value="" style="text-transform: capitalize;">
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Kategori Deskripsi</label>
        <div class="col-md-7">
            <select class="form-control" name="catName"
id="newCatName" required>
                </select>
        </div>
    </div>
</form>
</div>
<div class="modal-footer">
    <div class="btn-group pull-right">
        <button class="btn btn-primary"
onclick="storeBB('{{url("storeBahanBaku")}}','{{url("img/loading.gif")}}','I
nsert')"><span class="fa fa-save"></span> Save</button>
    </div>
</div>
</div>
</div>
<!-- MODALS Add BB -->

```

```

<!-- MODALS Edit BB -->
<div class="modal" id="modalBB-E" tabindex="-1" role="dialog" aria-
labelledby="defModalHead" aria-hidden="true">
  <div class="modal-dialog modal-md">
    <div class="modal-content">
      <div class="modal-header">
        <h4 class="modal-title" id="defModalHead">Edit Bahan Baku</h4>
        <button type="button" class="close" data-dismiss="modal"><span
aria-hidden="true">&times;</span><span class="sr-
only">Close</span></button>
      </div>
      <div class="modal-body">
        <form class="form-group" name="formBBEdit" id="formBBEdit">
          {{ csrf_field() }}
          <div class="form-group row">
            <label class="col-md-5">Kode Bahan Baku</label>
            <div class="col-md-7">
              <input type="text" name="bbID" id="bbID" class="form-
control" value="" readonly>
            </div>
          </div>
          <div class="form-group row">
            <label class="col-md-5">Nama Bahan Baku</label>
            <div class="col-md-7">
              <input type="text" name="bbName" id="bbName"
class="form-control" value="" style="text-transform: capitalize;">
            </div>
          </div>
          <div class="form-group row">
            <label class="col-md-5">Tipe Bahan Baku</label>
            <div class="col-md-7">

```

```

        <input type="text" name="bbType" id="bbType"
class="form-control" value="" style="text-transform: capitalize;">
    </div>
</div>
<div class="form-group row">
    <label class="col-md-5">Deskripsi Kategori</label>
    <div class="col-md-7">
        <select class="form-control" name="catName" id="catName"
required>
            </select>
        </div>
    </div>
</form>
</div>
<div class="modal-footer">
    <div class="btn-group pull-right">
        <button class="btn btn-primary"
onclick="storeBB('{{ url("updateBahanBaku") }}','{{ url("img/loading.gif") }}','
Edit')"><span class="fa fa-save"></span> Save</button>
    </div>
</div>
</div>
</div>
</div>
<!-- MODALS Edit Kategori -->
@endsection

```

5. Transaksi Surat Jalan

```
@extends ('layout.main')
```

```
@section('title','Input Data Surat Jalan')
```

```
@section('section')
```

```

<form class="form-group" name="formSuratAdd" id="formSuratAdd">
    {{ csrf_field() }}
    <div class="card border-left-warning shadow h-100 py-2" style="margin-
bottom:20px">
        <div class="card-body">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group row">
                        <label class="col-md-4">Kode Surat Jalan</label>
                        <div class="col-md-6">
                            <input type="text" name="sjID" id="sjID" class="form-control"
value="{{ $data->ID }}" readonly>
                        </div>
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-md-4">Lokasi</label>
                    <div class="col-md-6">
                        <textarea name="location" id="location" style="height:100px"
class="form-control" value="" required></textarea>
                    </div>
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-group row">
                    <label class="col-md-4">Tanggal Surat Jalan</label>
                    <div class="col-md-4">
                        <input type="text" name="deliveryDate" id="deliveryDate"
class="form-control" value="" required readonly>
                    </div>
                </div>
            </div>
            <div class="form-group row">

```

```

        <label class="col-md-4">Pembuat Surat Jalan</label>
        <div class="col-md-6">
            <input type="hidden" name="userID" id="userID"
value="{{ Cache::get(config('aplikasi.codeRedis')).Cookie::get('api_token'))-
>kd_pengguna }}">
            <input type="text" name="userName" id="userName"
class="form-control"
value="{{ Cache::get(config('aplikasi.codeRedis')).Cookie::get('api_token'))-
>nama_pengguna }}" readonly>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</form>
<div class="card mb-4 shadow">
    <div class="card-header bg-gradient-warning text-gray-900">
        <b>Data Bahan Baku</b>
    </div>
    <div class="card-header py-3">
        <form class="form-group" name="formSuratDetail"
id="formSuratDetail">
            <div class="row">
                <div class="col-md-3">
                    <div class="input-group">
                        <input type="hidden" name="stockID" id="stockID"
class="form-control">
                        <input type="text" name="orderID" id="orderID" class="form-
control" placeholder="Kode Pesanan..." readonly>
                    </div class="input-group-append">

```

```

        <a href="#" class="btn btn-warning" data-toggle="modal"
data-target="#modalListStok" onclick="getListStok('{{url("getStok")}}')">
            <i class="fas fa-list"></i>
        </a>
    </div>
</div>
</div>
<div class="col-md-3">
    <input type="hidden" name="bbID" id="bbID">
    <input type="text" name="bbName" id="bbName" class="form-
control" placeholder="Bahan Baku..." readonly>
</div>
<div class="col-md-1">
    <input type="text" name="qty" id="qty" onkeypress="return
isNumberKey(event)" class="form-control" placeholder="Qty...">
</div>
<div class="col-md-2">
    <button class="btn btn-primary" id="tmpSurat">
        <i class="fas fa-plus"></i>
    </button>
</div>
</div>
</form>
</div>
<div class="card-body">
    <div class="col-sm-12">
        <table id="suratBBTable" class="table table-bordered"
style="width:100%">
            <thead>
                <tr>
                    <th>Kode Stok</th>

```

```

        <th>Kode Pesanan</th>
        <th>Kode Bahan Baku</th>
        <th>Nama Bahan Baku</th>
        <th>Tipe Bahan Baku</th>
        <th>Kategori</th>
        <th>Qty</th>
        <th></th>
    </tr>
</thead>
<tbody></tbody>
<tfoot>
    <tr class="success">
        <th colspan="5" style="text-align:right">Total :</th>
        <th style="text-align:right"></th>
        <th></th>
    </tr>
</tfoot>
</table>
</div>
</div>
</div>

```

```

<div class="btn-group float-right">
    <div class="row">
    </div>
    <a href="{ { url('surat') } }" class="btn btn-danger">
        <span class="icon text-white-50">
            <i class="fas fa-arrow-left"></i>
        </span>
        <span class="text">Kembali</span>
    </a>

```



```

        <th>Kode Bahan Baku</th>
        <th>Nama Bahan Baku</th>
        <th>Qty</th>
        <th>Tipe Bahan Baku</th>
        <th>Kategori</th>
    </tr>
</thead>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- MODALS Stok List -->
@endsection

6. Monitor Problem
@extends ('layout.main')
@section('title','Data Problem')

@section('section')
    <div class="card shadow">
        <div class="card-body">
            <div class="table-responsive">
                <table class="table table-bordered" id="tableProblem" width="100% "
cellspacing="0">
                    <thead>
                        <tr>
                            <th>Kode Pengecekan</th>
                            <th>Kode Surat Jalan</th>

```

```

        <th>Kode Bahan Baku</th>
        <th>Nama Bahan Baku</th>
        <th>Keterangan</th>
        <th></th>
    </tr>
</thead>
</table>
</div>
</div>
</div>

```

```

<!-- MODALS Edit Problem -->
<div class="modal" id="modalProblem" tabindex="-1" role="dialog" aria-
labelledby="defModalHead" aria-hidden="true">
    <div class="modal-dialog modal-md">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title" id="defModalHead">Edit Keterangan</h4>
                <button type="button" class="close" data-dismiss="modal"><span
aria-hidden="true">&times;</span><span class="sr-
only">Close</span></button>
            </div>
            <div class="modal-body">
                <form class="form-group" name="formProblem"
id="formProblem">
                    {{ csrf_field() }}
                    <div class="form-group row">
                        <label class="col-md-5">Kode Pengecekan</label>
                        <div class="col-md-7">
                            <input type="text" name="checkID" id="checkID"
class="form-control" value="" readonly>

```

```

        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Kode Surat Jalan</label>
        <div class="col-md-7">
            <input type="text" name="sjID" id="sjID" class="form-
control" value="" readonly>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Nama Bahan Baku</label>
        <div class="col-md-7">
            <input type="text" name="bbName" id="bbName"
class="form-control" value="" style="text-transform: capitalize;" readonly>
        </div>
    </div>
    <div class="form-group row">
        <label class="col-md-5">Keterangan</label>
        <div class="col-md-7">
            <textarea name="remark" id="remark" class="form-control"
value=""></textarea>
        </div>
    </div>
</form>
</div>
<div class="modal-footer">
    <div class="btn-group pull-right">
        <button class="btn btn-primary"
onclick="storeProblem('{{ url("updateProblem") }}','{{ url("img/loading.gif") }}')">
        <span class="fa fa-save"></span> Save</button>
    </div>
</div>

```

```

        </div>
    </div>
</div>
</div>
</div>
<!-- MODALS Edit Problem -->
@endsection

7. Receiving Check
@extends ('layout.main')
@section('title','Pengecekan Kualitas')

@section('section')
    <div class="card border-left-success shadow h-100 py-2" style="margin-
bottom:20px">
        <div class="card-body">
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group row">
                        <label class="col-md-4">Kode Pengecekan</label>
                        <div class="col-md-6">
                            <input type="text" name="checkID" class="form-control"
value="{{ $data['data']->id_pengecekan }}" readonly>
                        </div>
                    </div>
                </div>
                <div class="form-group row">
                    <label class="col-md-4">Kode Surat Jalan</label>
                    <div class="col-md-6">
                        <input type="text" name="sjID" class="form-control"
value="{{ $data['data']->kd_sj }}" readonly>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="form-group row">
  <label class="col-md-4">Nama Bahan Baku</label>
  <div class="col-md-6">
    <input type="text" name="nameBB" class="form-control"
value="{{ $data['data']->nm_bb }}" readonly>
  </div>
</div>
<div class="form-group row">
  <label class="col-md-4">Qty</label>
  <div class="col-md-6">
    <input type="text" name="qty" class="form-control"
value="{{ $data['data']->qty }}" readonly required>
  </div>
</div>
<div class="form-group row">
  <label class="col-md-4">QC</label>
  <div class="col-md-6">
    <input type="text" name="user" class="form-control"
value="{{ $data['data']->nama_pengguna }}" readonly required>
  </div>
</div>
<div class="col-md-6">
  <div class="form-group row">
    <label class="col-md-4">Tanggal Pengecekan</label>
    <div class="col-md-6">
      <input type="text" name="checkDate" class="form-control"
value="{{ $checkDate }}" required readonly>
    </div>
  </div>
</div>
<div class="form-group row">

```

```

<label class="col-md-4">Kualitas Baik (qty)</label>
<div class="col-md-6">
    <input type="text" name="qtyGood" class="form-control"
value="{{ $data['data']->qty_good }}" placeholder="0" readonly required>
</div>
</div>
<div class="card border-left-danger shadow" style="margin-
bottom:20px">
    <div class="card-body">
        <div class="form-group row">
            <label class="col-md-4">Kualitas Buruk (qty)</label>
            <div class="col-md-6">
                <input type="text" name="qtyBad" class="form-control"
value="{{ $data['data']->qty_bad }}" readonly required>
            </div>
        </div>
        <div class="form-group row">
            <label class="col-md-4">Level Problem</label>
            <div class="col-md-6">
                <input type="text" name="level" class="form-control"
value="{{ $data['data']->desk_level }}" readonly>
            </div>
        </div>
        <div class="form-group row">
            <label class="col-md-4">Keterangan</label>
            <div class="col-md-6">
                <textarea name="remark" style="height:100px"
class="form-control" value="" readonly>{{ $data['data']-
>keterangan }}</textarea>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</form>

```

```

<div class="btn-group float-right">
    <a href="{ {url("check")}} " class="btn btn-danger">
        <span class="icon text-white-50">
            <i class="fas fa-arrow-left"></i>
        </span>
        <span class="text">Kembali</span>
    </a>
</div>

```

```
@endsection
```

8. Laporan Pesanan

```
@extends ('layout.main')
```

```
@section('title','Laporan Pesanan')
```

```
@section('section')
```

```

<div class="card shadow">
    <div class="card-header py-3">
        <div class="row">
            <div class="col-md-3">
                <input type="text" name="sDate" id="sDate" class="form-control"
placeholder="Mulai Tanggal" readonly>
            </div>
            <div class="col-md-3">

```



```

        <input type="text" name="eDate" id="eDate" class="form-control"
placeholder="Sampai Tanggal" readonly>
    </div>
    <div class="col-md-2">
        <button class="btn btn-info"
onclick="getReportPesanan('{{ url("getReportPesanan") }}')">
            <i class="fas fa-angle-double-right"></i>
        </button>
    </div>
</div>
<div class="card-body" id="viewReportPesanan">
    <div class="table-responsive">
        <table class="table table-bordered" id="tableReportPesanan"
width="100%" cellpadding="0">
            <thead>
                <tr>
                    <th>Kode Bahan Baku</th>
                    <th>Nama Bahan Baku</th>
                    <th>Kode Pesanan</th>
                    <th>Invoice</th>
                    <th>Tanggal Terima</th>
                    <th>Supplier</th>
                    <th></th>
                </tr>
            </thead>
        </table>
    </div>
</div>
<div class="btn-group float-right">

```

```

<a href="{ { url("surat") } }" class="btn btn-danger">
  <span class="icon text-white-50">
    <i class="fas fa-arrow-left"></i>
  </span>
  <span class="text">Kembali</span>
</a>

<div id="buttonCetak">
  <button
onclick="printPesanan('{ { url("printLaporanPesanan") } }',sDate.value,eDate.value)
;" class="btn btn-primary">
    <span class="icon text-white-50">
      <i class="fas fa-print"></i>
    </span>
    <span class="text">Cetak</span>
  </button>
</div>
</div>

```

@endsection

9. Laporan Pengecekan

@extends ('layout.main')

@section('title','Laporan Pengecekan')

@section('section')

```

<div class="card shadow">
  <div class="card-header py-3">
    <button onclick="printCheck('{ { url("printLaporanPengecekan") } }')"
class="btn btn-primary btn-icon-split">
      <span class="icon text-white-50">
        <i class="fas fa-info"></i>
      </span>
      <span class="text">Cetak Laporan</span>
    </button>
  </div>
</div>

```

</button>

</div>

</div>

@endsection