

No. Dole: 6760
Copy: 1

D3.658.562
Nas
S



**SISTEM INFORMASI PENGENDALIAN KUALITAS
UNTUK MEMBANTU MENGURANGI JUMLAH
CACAT PRODUKSI K 81 DENGAN METODE
STATISTICAL PROCESS CONTROL (SPC) PADA PT
NAGASAKTI COMPONENT PARTS**

TUGAS AKHIR

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

**OLEH
LUTHFI NASRULLAH
1315053**



**POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI
JAKARTA**

2019

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI

LEMBAR PENGESAHAN

Judul Tugas Akhir :

**SISTEM INFORMASI PENGENDALIAN KUALITAS UNTUK
MEMBANTU MENGURANGI JUMLAH CACAT PRODUKSI K 81
DENGAN METODE *STATISTICAL PROCESS CONTROL* (SPC) PADA
PT NAGASAKTI COMPONENT PARTS**

Disusun Oleh :

Nama : Luthfi Nasrullah
Nim : 1315053
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 R.I pada hari Senin tanggal 19 Agustus 2019.

Jakarta, 23 Agustus 2019

Dosen Pembimbing



Ahmad Juniar, S.Kom, MT
NIP: 197906052006041002

Dosen Penguji



Ahlan Ismono, S.Kom, MMSI
NIP: 197901072006041002

Ketua Penguji



Fifi Lailasari Hadianastuti, S.Kom, M.Kes
NIP: 197310162005022001

Dosen Penguji



Noveriza Yuliasari, S.Si, MT
NIP: 197811212009012003

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN R.I

TANDA PERSETUJUAN DOSEN PEMBIMBING

Judul Tugas Akhir : SISTEM INFORMASI PENGENDALIAN
KUALITAS UNTUK MEMBANTU
MENGURANGI CACAT PRODUKSI K 81
DENGAN METODE *STATISTICAL PROCESS*
CONTROL (SPC) PADA PT NAGASAKTI
COMPONENT PARTS

Disusun Oleh

Nama : Luthfi Nasrullah
Nim : 1315053
Program Studi : Sistem Informasi Industri Otomotif
Seminar : 04 Agustus 2019
Sidang : 19 Agustus 2019
Lulus : 19 Agustus 2019

Jakarta, 03 September 2019

Menyetujui
Dosen Pembimbing



Ahmad Juniar, S.Kom, MT
NIP. 197906052006041002

LEMBAR BIMBINGAN PENYUSUNAN TUGAS AKHIR

Nama : Luthfi Nasrullah
NIM : 1315053
Judul Laporan PKL II : Sistem Pengendalian Kualitas Untuk Mengurangi Cacat Produksi K 81 Dengan Metode Statistical Process Control (SPC) Pada PT Nagasakti Component Parts
Pembimbing : Ahmad Juniar, S.Kom, MT

Tanggal	BAB	Keterangan	Paraf
22 April 2019	Bab I	Perbaiki hal 1 & 2.	Ah
24 April 2019	Bab II	Perbaiki hal 6 & 7.	Ah
13 Mei 2019	Bab III	Perbaiki sub bab 3.6.	Ah
21 Mei 2019	Bab IV	Perbaiki Tulisan QA	Ah
24 Mei 2019	Bab IV	Alur proses berjalan & analisis	Ah
12 Juli 2019	-	Demo Aplikasi ke-1	Ah
13 Juli 2019	Bab V	Perbaiki <i>non functional</i> .	Ah
22 Juli 2019	Bab V	<i>Sequence Diagram, Class Diagram, WND, ERD, Kamus Data.</i>	Ah
23 Juli 2019	Bab V	<i>Deployment Diagram, Perancangan User Interface.</i>	Ah
24 Juli 2019	Bab VI	Kesimpulan, Susunan lap. TA.	Ah

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif



Noveriza Yuliasari, S.Si, MT
NIP : 197811212009012003

Pembimbing



Ahmad Juniar, S.Kom, MT
NIP : 197906052006041002



LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Luthfi Nasrullah

NIM : 1315053

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

“SISTEM INFORMASI PENGENDALIAN KUALITAS UNTUK MEMBANTU MENGURANGI JUMLAH CACAT PRODUKSI K 81 DENGAN METODE *STATISTICAL PROCESS CONTROL* (SPC) PADA PT NAGASAKTI COMPONENT PARTS”

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

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

Jakarta, 5 Agustus 2019


Luthfi Nasrullah

ABSTRAK

PT Nagasaki Component Parts merupakan salah satu perusahaan manufaktur yang bergerak di bidang otomotif dan memproduksi berbagai jenis *rubber*. Pada proses produksinya PT Nagasaki Component Parts memproduksi jenis *rubber* berdasarkan peramalan yang sudah dibuat. Salah satu produk unggulan dari PT Nagasaki Component Parts adalah K 81 merupakan sebuah komponen otomotif yang berada di bagian *speedometer* honda beat, fungsinya adalah untuk mencegah korsleting listrik kabel di bagian *speedometer*. Dalam melakukan pengendalian kualitas masih terdapat beberapa masalah yaitu, seperti pengelolaan data masih masih dicatat dalam kertas sehingga mengakibatkan terjadinya kehilangan kertas, kehilangan *file* yang disimpan dalam *flashdisk* akibat virus dan *output* yang digunakan tidak bisa dijadikan acuan dalam pengambilan keputusan banyak terjadinya cacat produksi. Pengendalian kualitas dapat dilakukan secara statistik atau disebut dengan pengendalian kualitas statistik SPC (*Statistical Process Control*). Keunggulan dari metode ini adalah menentukan cara penyelesaian masalah, mengendalikan, membuat produk tetap sesuai dengan standar spesifikasi dari awal proses hingga akhir proses, dan menentukan faktor – faktor utama yang berpengaruh pada penyebab utama terjadinya kegagalan dalam proses produksi. Dari masalah tersebut maka dibuatlah sistem aplikasi berbasis komputer yang menggunakan metode SPC (*Statistical Process Control*) dan disimpan menggunakan database MariaDB, sehingga menghasilkan *output* yang bisa dijadikan acuan pengambilan keputusan dalam pengendalian kualitas produksi K 81 bagi divisi *Quality Control* itu sendiri. Untuk penelitian selanjutnya, sistem informasi ini dapat diimplementasikan pada proses pengendalian kualitas di bagian *Quality Control* pada PT Nagasaki Component Parts, dan dapat diintegrasikan dengan bagian pegawai,

Kata Kunci: Cacat Produksi, Pengendalian kualitas, *Statistical Process Control*.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul **“SISTEM INFORMASI PENGENDALIAN KUALITAS UNTUK MEMBANTU MENGURANGI JUMLAH CACAT PRODUKSI K 81 DENGAN METODE *STATISTICAL PROCESS CONTROL* (SPC) PADA PT NAGASAKTI COMPONENT PARTS”**

Penulisan Tugas Akhir ini bertujuan untuk memenuhi sebagian syarat penyelesaian program Sarjana Terapan pada program studi Sistem Informasi Industri Otomotif 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 dan penghargaan kepada:

1. Allah SWT yang telah memberikan segala sesuatu serta kelancaran dalam menyelesaikan Tugas Akhir ini.
2. Kedua orang tua serta kakak dan adik yang selalu mendoakan dan memberikan dukungan keberhasilan penulis.
3. Bapak Dr. Mustofa, S.T, MT selaku Direktur Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, S.Si, MT selaku Ketua Program Studi Sistem Informasi Industri Otomotif.
5. Bapak Ahmad Juniar, S.Kom, MT selaku dosen pembimbing yang telah memberikan arahan, bimbingan serta penjelasan dalam penyusunan Tugas Akhir.
6. Seluruh dosen Politeknik STMI Jakarta yang telah memberikan ilmunya guna menambah pengetahuan dan pengalaman yang sangat bermanfaat.

7. Seluruh anggota organisasi Forum Olahraga Mahasiswa yang selalu memberikan semangat, motivasi dan saran sehingga Tugas Akhir ini dapat diselesaikan secara bertahap.
8. Ibu Juliana Burmawi, SE selaku direktur PT Nagasakti Component Parts.
9. Bapak Maman dan Bapak Asep selaku pembimbing yang telah membantu mengarahkan dan membimbing selama Kerja Lapangan di PT Nagasakti Component Parts.
10. Teman-teman mahasiswa/i Program Studi Sistem Informasi Industri Otomotif 2015 yang telah membantu memberikan saran bantuan dalam penyusunan Tugas Akhir ini.
11. Serta semua pihak yang tidak disebutkan satu persatu yang telah memberikan kritik, saran dan bantuan dalam penyusunan Tugas Akhir ini.

Semoga segala bantuan yang telah diberikan kepada penulis mendapat balasan yang setimpal dari Allah SWT. Penulis menyadari masih banyak kekurangan dalam penyusunan Tugas Akhir ini. Semoga Tugas Akhir ini dapat memberikan manfaat dan informasi yang berguna bagi kami dan para pembaca.

Jakarta, 5 Agustus 2019

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN	ii
LEMBAR PERSETUJUAN DOSEN PEMBIMBING.....	iii
LEMBAR BIMBINGAN TUGAS AKHIR.....	iv
LEMBAR PERNYATAAN KEASLIAN	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Pokok Permasalahan	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
1.5 Manfaat Tugas Akhir	4
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	6
2.1 Pengertian Sistem	6
2.1.1 Pengertian Sub Sistem.....	6
2.1.2 Karakteristik Sistem	7
2.1.3 Klasifikasi Sistem.....	8
2.1.4 Daur Hidup Sistem	9
2.2 Informasi	11
2.2.1 Fungsi Informasi.....	11
2.2.2 Nilai Informasi	12

2.2.3	Kualitas Informasi	13
2.3	Pengertian Sistem Informasi	14
2.4	Konsep Dasar Informasi	15
2.5	Komponen dan Tipe Sistem Informasi	15
2.6	Pengertian Pengendalian	18
2.7	Pengertian Kualitas	18
2.8	Pengertian Pengendalian Kualitas	18
2.8.1	Tujuan Pengendalian Kualitas	19
2.8.2	Faktor Pengendalian Kualitas	20
2.9	<i>Statistical Process Control (SPC)</i>	21
2.9.1	Contoh Kasus <i>Statistical Process Control (SPC)</i>	26
2.9.2	Diagram Pareto	28
2.9.3	Peta Kedali	29
2.10	<i>System Development Life Cycle (SDLC)</i>	31
2.11	Model <i>Waterfall</i>	32
2.12	Analisis PIECES	32
2.13	<i>System Requirement</i>	34
2.14	<i>Flowmap</i>	35
2.15	<i>Unified Modelling Language (UML)</i>	36
2.17.1	<i>Use Case Diagram</i>	38
2.17.2	<i>Use Case Description</i>	39
2.17.3	<i>Diagram Aktivitas (Activity Diagram)</i>	40
2.17.4	<i>Sequence Diagram</i>	42
2.17.5	<i>Class Diagram</i>	44
2.16	<i>Deployment Diagram</i>	45
2.17	ERD (<i>Entity Relationship Diagram</i>)	46
2.18	Sistem Manajemen Basis Data	47
2.18.1	Basis data (<i>Database</i>)	47
2.18.2	<i>Conceptual Data Modelling (CDM)</i>	47
2.18.3	Kamus Data	50
2.18.4	MariaDB	50

2.18.5	Tipe Data.....	51
2.19	Windows Navigation Diagram.....	52
2.20	Hypertext Processor (PHP)	53
2.21	Framework.....	54
2.22	Codeigniter	55
2.23	Black-Box Testing	58
BAB III METODOLOGI PENELITIAN		59
3.1	Metodologi Penelitian.....	59
3.2	Jenis dan Sumber Data.....	59
3.3	Metode Pengumpulan Data.....	59
3.4	Metode Pengembangan Sistem.....	60
3.5	Literature Review.....	61
3.6	Kerangka Penelitian.....	65
BAB IV PENGUMPULAN DAN PENGOLAHAN DATA		71
4.1	Sejarah Perusahaan	71
4.2	Profil Perusahaan	72
4.3	Visi dan Misi Perusahaan	72
4.4	Struktur Organisasi Perusahaan	73
4.5	Job Description.....	76
4.6	Sertifikat Perusahaan	82
4.7	K 81.....	85
4.8	Alur Proses Yang Berjalan	86
4.9	Dokumen-Dokumen Terkait Pengendalian Kualitas Proses Produksi	87
4.10	Analisis Permasalahan	92
BAB V ANALISIS DAN PEMBAHASAN.....		95
5.1	Analisis Kebutuhan Sistem.....	95
5.1.1	Functional Requirment	97
5.1.2	Non Functional Requirment.....	97
5.2	Alur Proses Pengendalian Kualitas Usulan	98
5.3	Pemodelan Sistem Informasi Pengendalian Kualitas Usulan	101

5.3.1	<i>Use Case Diagram</i>	101
5.3.2	<i>Use Case Description</i>	102
5.4	<i>Activity Diagram</i>	108
5.5	<i>Sequence Diagram</i>	118
5.6	<i>Class Diagram</i>	126
5.7	<i>Deployment Diagram</i>	127
5.8	<i>Windows Navigation Diagram</i>	128
5.9	Pemodelan Data Sistem Usulan	129
5.1	<i>Entity Relationship Diagram (ERD)</i>	129
5.2	Kamus Data.....	129
5.3	Perancangan <i>Interface</i> Sistem.....	131
5.10	Contoh Kasus SPC	139
BAB VI PENUTUP		144
6.1	Kesimpulan	144
6.2	Saran	144
DAFTAR PUSTAKA		145
Lampiran		

DAFTAR GAMBAR

	Halaman
Gambar II.1 Daur Hidup Sistem	11
Gambar II.2 Komponen Sistem Informasi	17
Gambar II.3 Contoh Jenis Cacat	26
Gambar II.4 Contoh Hasil Perhitungan Cacat.....	26
Gambar II.5 Contoh Perhitungan Diagram Pareto	27
Gambar II.6 Contoh Diagram Pareto	27
Gambar II.7 Contoh Perhitungan Peta Kendali.....	27
Gambar II.8 Contoh Peta Kendali	28
Gambar II.9 Diagram Pareto	29
Gambar II.10 Peta Kendali.....	30
Gambar II.11 Pola Melingkar dari Siklus Hidup Sistem	31
Gambar II.12 Penggambaran Metode <i>Waterfall</i>	32
Gambar II.13 Contoh <i>Windows Navigation Diagram</i>	52
Gambar II.14 Cara Kerja <i>Codeigniter</i>	56
Gambar III.1 Kerangka Penelitian	69
Gambar IV.1 Lambang Perusahaan PT Nagasakti Component Parts	71
Gambar IV.2 Struktur Organisasi PT Nagasakti Component Parts	75
Gambar IV.3 <i>Good Quality & Delivery Performance PT Hi-Lex Indonesia</i>	82
Gambar IV.4 <i>FMEA and Control Plan PT Hi-Lex Indonesia</i>	83
Gambar IV.5 <i>Good Quality & Delivery Performance PT Hi-Lex Indonesia</i>	84
Gambar IV.6 <i>Reward</i> PT Astra Honda Motor	84
Gambar IV.7 ISO TS 16949 : 2009	85
Gambar IV.8 K 81	85
Gambar IV.9 Proses Bisnis Sistem Pengendalian Kualitas Proses Produksi	87
Gambar IV.10 <i>Check Sheet</i>	88
Gambar IV.11 Laporan Hasil Kerja Produksi PT Nagasakti Component Parts.....	89

Gambar IV.12	<i>Form Data Part Reject</i>	90
Gambar IV.13	<i>Form Countermeasure</i>	91
Gambar V.1	Alur Proses Pengendalian Kualitas Usulan	97
Gambar V.2	<i>Use Case</i> Usulan	101
Gambar V.3	<i>Activity Diagram</i> Melakukan <i>Login</i>	109
Gambar V.4	<i>Activity Diagram</i> Mengelola Data Master Produk	110
Gambar V.5	<i>Activity Diagram</i> Mengelola Data Master Penyebab Cacat....	111
Gambar V.6	<i>Activity Diagram</i> Mengelola Data Master <i>User</i>	112
Gambar V.7	<i>Activity Diagram</i> Mengelola Data Transaksi Hasil Produksi .	113
Gambar V.8	<i>Activity Diagram</i> Mengelola Data Transaksi Cacat Produksi.	114
Gambar V.9	<i>Activity Diagram</i> Mencetak Laporan Grafik.....	115
Gambar V.10	<i>Activity Diagram</i> Mencetak Laporan Hasil Produksi.....	116
Gambar V.11	<i>Activity Diagram</i> Mencetak Laporan Cacat Produksi.....	117
Gambar V.12	<i>Sequence Diagram</i> Melakukan <i>Login</i>	118
Gambar V.13	<i>Sequence Diagram</i> Mengelola Data Master Produk	119
Gambar V.14	<i>Sequence Diagram</i> Mengelola Data Master Penyebab Cacat .	120
Gambar V.15	<i>Sequence Diagram</i> Mengelola Data Master <i>User</i>	121
Gambar V.16	<i>Sequence Diagram</i> Mengelola Data Transaksi Hasil Produksi.....	122
Gambar V.17	<i>Sequence Diagram</i> Mengelola Data Transaksi Cacat Produksi.....	123
Gambar V.18	<i>Sequence Diagram</i> Mencetak Laporan Grafik.....	124
Gambar V.19	<i>Sequence Diagram</i> Mencetak Laporan Hasil Produksi.....	125
Gambar V.20	<i>Sequence Diagram</i> Mencetak Laporan Cacat Produksi.....	126
Gambar V.21	<i>Class Diagram</i> Sistem Informasi Pengendalian Kualitas	127
Gambar V.22	<i>Deployment Diagram</i> Sistem Informasi Pengendalian Kualitas	128
Gambar V.23	WND Sistem Informasi Pengendalian Kualitas	128
Gambar V.24	ERD Sistem Informasi Pengendalian Kualitas.....	129
Gambar V.25	<i>Interface Login</i>	132
Gambar V.26	<i>Interface Menu Utama</i>	133

Gambar V.27	<i>Interface</i> Data Master Produk	134
Gambar V.28	<i>Interface</i> Data Master Penyebab Cacat	135
Gambar V.29	<i>Interface</i> Data Master <i>User</i>	136
Gambar V.30	<i>Interface</i> Data Transaksi Hasil Produksi.....	136
Gambar V.31	<i>Interface</i> Data Transaksi Cacat Produksi	137
Gambar V.32	<i>Interface</i> Laporan Grafik.....	137
Gambar V.33	<i>Interface</i> Laporan Hasil Produksi.....	138
Gambar V.34	<i>Interface</i> Laporan Cacat Produksi.....	138
Gambar V.35	Laporan Grafik Diagram Pareto	141
Gambar V.36	Laporan Grafik Peta Kendali.....	143

DAFTAR TABEL

	Halaman
Tabel II.1 Simbol-simbol <i>Flowmap</i>	35
Tabel II.2 Simbol-simbol <i>Use-Case Diagram</i>	38
Tabel II.3 Simbol-Simbol <i>Activity Diagram</i>	41
Tabel II.4 Simbol-Simbol <i>Sequence Diagram</i>	43
Tabel II.5 Simbol-Simbol <i>Class Diagram</i>	44
Tabel II.6 Simbol-Simbol <i>Deployment Diagram</i>	45
Tabel II.7 Simbol-Simbol ERD	46
Tabel II.8 Simbol-Simbol <i>Conceptual Data Model</i> (CDM)	48
Tabel II.9 Simbol-Simbol Aturan <i>Conceptual Data Model</i> (CDM)	48
Tabel II.10 Contoh Kamus Data	50
Tabel II.11 Tipe Data MySQL	51
Tabel III.1 <i>Literature Review</i>	63
Tabel IV.1 Analisis Permasalahan	92
Tabel V.1 Analisis Kebutuhan Sistem	98
Tabel V.2 <i>Use Case Description</i> Melakukan <i>Login</i>	101
Tabel V.4 <i>Use Case Description</i> Mengelola Master Data Produk	102
Tabel V.5 <i>Use Case Description</i> Mengelola Master Data Penyebab cacat .	103
Tabel V.5 <i>Use Case Description</i> Mengelola Master Data <i>User</i>	104
Tabel V.6 <i>Use Case Description</i> Mengelola Data Transaksi Hasil Produksi	104
Tabel V.7 <i>Use Case Description</i> Mengelola Data Transaksi Cacat Produksi	105
Tabel V.8 <i>Use Case Description</i> Mencetak Laporan Grafik	106
Tabel V.9 <i>Use Case Description</i> Mencetak Laporan Hasil Produksi	107
Tabel V.10 <i>Use Case Description</i> Mencetak Laporan Cacat Produksi	108
Tabel V.11 Spesifikasi Tabel User	130
Tabel V.12 Spesifikasi Tabel Produk	130

Tabel V.13	Spesifikasi Tabel Penyebab Cacat	130
Tabel V.14	Spesifikasi Tabel Produksi.....	131
Tabel V.15	Spesifikasi Tabel Cacat Produksi.....	131
Tabel V.16	Contoh Penyebab Cacat PT Nagasakti Component Parts	139
Tabel V.17	Contoh Jumlah Penyebab Cacat PT Nagasakti Component Parts.....	139
Tabel V.18	Contoh Jumlah Penyebab Cacat Yang Sudah Dikategorikan PT Nagasakti Component Parts	140
Tabel V.19	Perhitungan Peta Kendali PT Nagasakti Component Parts.....	141

DAFTAR LAMPIRAN

	Halaman
Lampiran A Coding.....	L-1
Lampiran B Black-Box Testing	L-21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perusahaan adalah tempat terjadinya kegiatan produksi dan berkumpulnya semua faktor produksi. Manufaktur adalah suatu cabang industri yang mengaplikasikan mesin, peralatan dan tenaga kerja dan suatu medium proses untuk mengubah bahan mentah menjadi barang jadi yang memiliki nilai jual, sehingga perusahaan manufaktur merupakan suatu perusahaan yang aktivitasnya mengelola bahan mentah atau bahan baku sehingga menjadi barang jadi lalu menjualnya kepada konsumen. *Inspection* merupakan tempat/area pemeriksaan produk yang telah diproduksi (produk jadi), lalu akan diperiksa sesuai dengan kriteria/SOP yang telah dicantumkan, dan produk itu akan menghasilkan kriteria *NG/Good*.

Pengendalian kualitas proses produksi merupakan suatu hal yang harus dilakukan setelah melakukan proses produksi, guna untuk mengetahui kualitas produknya sebelum dipasarkan kepada konsumen. dengan berkembangnya teknologi informasi yang semakin pesat sistem pengendalian kualitas harus memenuhi standar mutu yang telah ditentukan, untuk mengurangi terjadinya kecacatan produk.

PT Nagasakti Component Parts merupakan salah satu perusahaan manufaktur yang bergerak di bidang otomotif dan memproduksi berbagai jenis *rubber*. Pada proses produksinya PT Nagasakti Component Parts memproduksi jenis *rubber* berdasarkan peramalan yang sudah dibuat. *Rubber* yang telah selesai diproduksi akan diperiksa di area *inspection* yang menghasilkan kategori *NG/Good*, lalu produk yang menghasilkan *Good* akan disimpan ke dalam gudang barang jadi yang nantinya akan dikirimkan ke *customer*.

PT Nagasakti Component Parts memiliki Divisi *Quality Control* yang bertugas melakukan pengendalian kualitas di setiap proses produksinya. Proses produksi komponen K 81 di PT Nagasakti Component Parts terdiri dari enam proses produksi, yaitu *Incoming Material*, *Material Compound*, *Mixing*, Produksi,

Finishing, dan *Inspection* dimana masing-masing setiap proses produksi memiliki standar acuan melakukan pemeriksaan. Dalam menjalankan kegiatan pengendalian kualitas tidak terlepas dari permasalahan-permasalahan yang terjadi, diantaranya kesulitan dalam pencarian data karena banyaknya dokumen yang tersimpan di dalam arsip tertumpuk. Selain itu, belum adanya aplikasi yang terintegrasi dengan *database* yang mengakibatkan kesulitan dalam melakukan pembaruan atau pencarian informasi dalam pengendalian kualitas, serta sering terjadinya kehilangan data. Untuk pembuatan laporannya, masih menggunakan *Microsoft Excel*, sehingga laporan cacat produksi kurang informatif untuk dijadikan bahan evaluasi kinerja.

Berdasarkan penjelasan di atas maka, diperlukan pengembangan suatu aplikasi yang dapat membantu perusahaan dalam mengelola setiap data pengendalian kualitas yang menjadi suatu laporan yang berguna bagi perusahaan, Adapun judul Tugas Akhir ini adalah “SISTEM INFORMASI PENGENDALIAN KUALITAS PROSES PRODUKSI UNTUK MENGURANGI JUMLAH CACAT PRODUKSI K 81 DENGAN MENGGUNAKAN METODE *STATISTICAL PROCESS CONTROL* (SPC) PADA PT NAGASAKTI COMPONENT PARTS”.

1.2. Pokok Permasalahan

Permasalahan yang terjadi di Divisi Quality Control pada PT Nagasaki Component Parts adalah sebagai berikut:

1. Pencatatan data pengendalian kualitas masih menggunakan kertas, sehingga mengakibatkan terjadinya penumpukan kertas dan sering terjadi kehilangan kertas, media penyimpanan menggunakan *flashdisk*, sehingga terjadinya *file corrupt* akibat terkena virus, dan pembuatan Laporan masih menggunakan *Microsoft Excel*, *output* yang dihasilkan hanya berupa tabel, tidak bisa dijadikan acuan untuk pengambilan keputusan penyebab terjadinya cacat produksi.
2. Banyak terjadinya cacat produksi dalam proses produksi mengakibatkan pengiriman produksi terhambat.

1.3. Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini yang dilakukan terhadap Divisi *Quality Control* PT Nagasaki Component Parts dalam melakukan pengendalian kualitas proses produksi adalah untuk merancang dan membangun sistem informasi pengendalian kualitas proses produksi yang mampu:

1. Membuat pengolahan data menjadi terkomputerisasi, menyediakan penyimpanan aplikasi berupa *database*, dan pembuatan aplikasi *user friendly*.
2. Menerapkan metode *Statistical Process Control* (SPC), yang dapat membantu mengurangi jumlah cacat produksi yang dihasilkan.

1.4. Batasan Masalah

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

1. Tempat melakukan penelitian adalah PT Nagasaki Component Parts pada divisi *Quality Control* selama satu bulan Sembilan belas hari, mulai dari 11 Juli sampai 30 Agustus 2019.
2. Analisis dan penelitian hanya sebatas mengenai penyajian informasi pengendalian kualitas proses produksi K 81 menggunakan metode *Statistical Process Control* (SPC).
3. Menggunakan *Framework Codeigniter* 3.1.10, PHP 7.3.7, dan MariaDB 10.3.16.
4. Sistem ini hanya di rancang bangun tetapi belum diimplementasikan.

1.5 Manfaat Tugas Akhir

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

1. Sebagai bahan untuk pengambilan keputusan yang membantu kinerja perusahaan dalam meningkatkan kualitas produk.
2. Membantu perusahaan dalam menurunkan jumlah cacat produksi yang dihasilkan menggunakan metode *Statistical Process Control* (SPC).

1.6 Sistematika Penulisan

Tugas Akhir ini disusun secara sistematis guna memberikan gambaran yang jelas mengenai isi dan pembahasan yang ada di dalamnya. Adapun sistematika penulisan Tugas Akhir ini diuraikan dalam enam bab, yaitu:

BAB I PENDAHULUAN

Pada bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat tugas akhir, dan sistematika penulisan.

BAB II LANDASAN TEORI

Pada bab ini membahas tentang berbagai teori yang diperoleh dari buku-buku, literatur, ataupun berbagai macam referensi yang berkaitan dengan tema yang diambil. Teori-teori yang dipaparkan pada laporan ini adalah seputar sistem informasi, pengendalian kualitas, *Unified Modelling Language* (UML), serta teori-teori lain yang berhubungan dengan perancangan sistem informasi.

BAB III METODOLOGI PENELITIAN

Pada bab ini menjelaskan metode ilmiah dalam mencari dan menjelaskan kerangka pemecahan masalah yang menguraikan tahap-tahap untuk mengatasi permasalahan yang telah diidentifikasi.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini membahas tentang data yang telah diperoleh selama melaksanakan penelitian di PT Nagasakti Component Parts seperti alur proses produksi K 81, proses bisnis yang berjalan, dan hal lain yang mendukung sistem informasi pengendalian kualitas proses produksi K 81.

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi analisis rinci dari pengolahan data, yakni mulai dari analisis kebutuhan sistem, pemodelan sistem dengan *Unified Modelling Language* (UML), pemodelan data dan *Entity Relationship Diagram* (ERD), perancangan antarmuka, sampai kebutuhan *software* dan *hardware* yang diperlukan.

BAB VI PENUTUP

Pada bab ini berisikan kesimpulan dari hasil penelitian yang dilakukan, serta saran-saran dalam penerapan sistem informasi pengendalian kualitas cacat produksi K 81 untuk perusahaan dan pengembangan selanjutnya.

BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Sutabri, 2012). Sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan (Yakub, 2012). Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran tertentu dan juga sistem merupakan kumpulan dari komponen-komponen yang memiliki unsur keterkaitan antara satu dan yang lain (Hutahaean, 2015). Sistem adalah kumpulan elemen yang saling berinteraksi untuk mencapai satu tujuan tertentu (Ladjmudin, 2005). Sistem adalah kumpulan/grup/komponen apapun baik fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu (Setyorini, 2016). Sistem adalah suatu susunan yang teratur dari kegiatan-kegiatan yang saling berkaitan dan susunan prosedur prosedur yang saling berhubungan, yang melaksanakan dan mempermudah kegiatan-kegiatan utama organisasi/institusi (Setyorini, 2016).

2.1.1 Pengertian Subsistem

Suatu sistem dapat terdiri dari bagian-bagian sistem atau subsistem. Contohnya, sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem lagi atau terdiri komponen-komponen pendukung sistem itu sendiri. Subsistem perangkat keras dapat terdiri dari alat masukan, alat pemroses, alat keluaran, dan media penyimpanan. Subsistem-subsistem ini saling berinteraksi. dan saling berhubungan membentuk satu kesatuan sehingga tujuan dan sasaran tersebut dapat tercapai (Sutabri, 2012).

Subsistem sebenarnya hanyalah sistem didalam suatu sistem, ini berarti bahwa sistem berada pada lebih dari satu tingkat. Permisalan lainnya, mobil adalah suatu sistem yang terdiri dari sistem-sistem bawahan seperti sistem mesin, sistem badan mobil dan sistem rangka. Masing-masing sistem ini terdiri dari sistem tingkat yang lebih rendah (Setyorini, 2016).

2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen, lingkungan luar sistem, penghubung, masukan, keluaran, pengolah dan sasaran atau tujuan (Ladjmudin, 2005).

- **Komponen Sistem**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai karakteristik dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

- **Batasan Sistem**

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

- **Lingkungan Luar Sistem**

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

- **Penghubung Sistem**

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

- **Masukan Sistem**

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal *maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat berjalan. Sinyal input adalah energi yang diproses untuk mendapatkan keluaran dari sistem.

- **Keluaran Sistem**

Keluaran sistem adalah energi yang diolah dan diklasifikasikan menjadi keluaran berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain.

- **Pengolahan Sistem**

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

- **Sasaran Sistem**

Suatu sistem mempunyai tujuan atau sasaran, kalau sistem tidak mempunyai sasaran maka sistem tidak akan ada. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya. Sasaran sangat berpengaruh pada masukan dan keluaran yang dihasilkan.

2.1.3 Klasifikasi Sistem

Suatu sistem dapat diklasifikasikan dari beberapa sudut pandang, yaitu (Setyorini, 2016):

1. **Sistem Abstrak (*Abstract System*)**, adalah sistem yang tidak tampak secara fisik, karena hanya berupa pemikiran atau ide-ide. Contoh, sistem Teologia

yang merupakan suatu sistem yang menggambarkan hubungan manusia dengan Tuhan.

2. Sistem Fisik (*Physical System*), adalah sistem yang tampak secara fisik. Contoh, sistem komputer, sistem produksi, sistem pendidikan, dll.
3. Sistem Alamiah (*Natural System*), adalah sistem yang terdiri dari proses-proses alam. Contoh sistem Geologi.
4. Sistem buatan Manusia (*Human made System*), adalah suatu sistem yang dirancang atau didesain oleh manusia. Contoh sistem informasi.
5. Sistem Deterministik (*Deterministic System*), adalah sistem yang beroperasi dengan tingkah laku yang dapat diramalkan. Interaksi antar elemen-elemen dapat dideteksi, sehingga *ouputnya* juga dapat diramalkan. Contoh sistem komputer.
6. Sistem Probabilitas (*Probabilistic System*), adalah sistem yang tidak bisa diramalkan. Contoh sistem manusia.
7. Sistem Tertutup (*Closed System*), adalah sistem yang tidak berhubungan dengan lingkungan luarnya.
8. Sistem Terbuka (*Open System*), adalah sistem yang berhubungan atau dipengaruhi oleh lingkungan luarnya.

2.1.4 Daur Hidup Sistem

Siklus hidup sistem (*system life cycle*) adalah proses evolusioner yang diikuti dalam penerapan sistem atau subsistem informasi berbasis komputer (Sutabri, 2012). Siklus hidup sistem terdiri dari serangkaian tugas yang mengikuti yang mengikuti langkah-langkah pendekatan sistem, karena tugas-tugas tersebut mengikuti pola yang teratur dan dilakukan secara *top down*. Siklus hidup sistem sering disebut sebagai pendekatan air terjun (*waterfall approach*) bagi pembangunan dan pengembangan sistem. Pembangunan sistem hanyalah salah satu dari rangkaian daur hidup suatu sistem. Meskipun demikian proses ini merupakan aspek yang sangat penting. Ada beberapa tahapan daur hidup suatu sistem adalah sebagai berikut:

1. Mengenali adanya kebutuhan

Sebelum segala sesuatu terjadi, pastilah terlebih dahulu timbul suatu kebutuhan atau problema yang harus dapat dikenali sebagaimana adanya. Kebutuhan dapat terjadi sebagai hasil perkembangan organisasi. Volume kebutuhan itu meningkat melebihi kapasitas dari sistem yang ada. Semua kebutuhan itu harus didefinisikan dengan jelas. Tanpa adanya kejelasan mengenai kebutuhan yang ada, pembangunan sistem akan kehilangan arah dan efektivitasnya.

2. Pembangunan sistem

Suatu proses atau seperangkat prosedur yang harus diikuti guna menganalisis kebutuhan yang timbul dan membangun suatu sistem untuk memenuhi kebutuhan tersebut.

3. Pemasangan sistem

Setelah tahap pembangunan sistem selesai, kemudian sistem akan dioperasikan. Pemasangan sistem merupakan tahap yang penting dalam daur hidup sistem, di mana peralihan dari tahap pembangunan menuju tahap operasional adalah pemasangan sistem, yang merupakan langkah akhir dari suatu pembangunan sistem.

4. Pengoperasian sistem

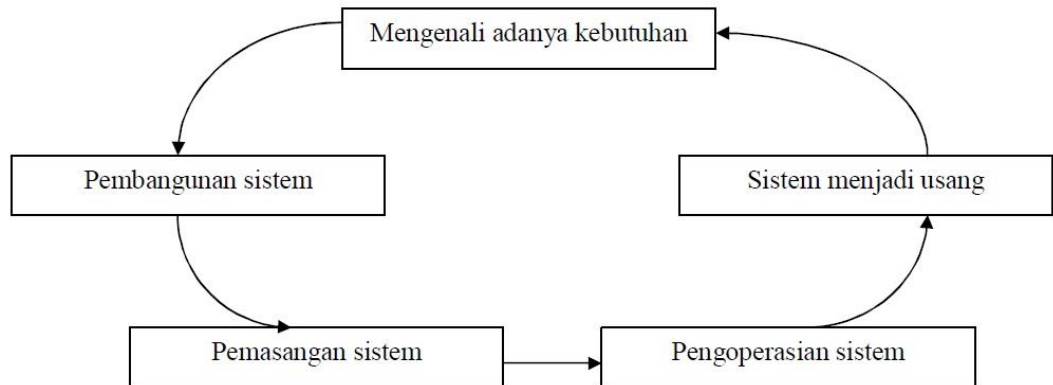
Program-program komputer dan prosedur-prosedur pengoperasian yang membentuk suatu sistem informasi semuanya bersifat statis, sedangkan organisasi yang ditunjang oleh sistem informasi selalu mengalami perubahan karena pertumbuhan kegiatan, perubahan peraturan dan kebijaksanaan, ataupun kemajuan teknologi. Untuk mengatasi perubahan-perubahan tersebut, sistem harus diperbaharui atau diperbaiki.

5. Sistem menjadi usang

Kadang-kadang perubahan yang terjadi begitu drastis sehingga tidak dapat diatasi hanya dengan melakukan perbaikan sistem yang sedang berjalan. Pada saat itu, sistem tersebut sudah tidak layak lagi untuk dioperasikan dan sistem yang baru perlu dibangun untuk menggantikannya.

Sistem informasi kemudian akan melanjutkan daur hidupnya. Sistem dibangun untuk memenuhi kebutuhan. Sistem beradaptasi terhadap aneka perubahan lingkungan yang dinamis hingga kemudian sampai pada kondisi di mana

sistem tidak dapat lagi beradaptasi. Sistem baru kemudian dibangun untuk menggantikannya. Daur sistem ini dapat dilihat pada Gambar II.1



Gambar II.1 Daur Hidup Sistem
Sumber: (Sutabri, 2012)

2.2 Informasi

Informasi adalah sebuah istilah yang tepat dalam pemakaian umum mengenai data mentah, data tersusun, kapasitas sebuah saluran komunikasi dan lain sebagainya (Sutabri, 2012). Informasi juga mencakup mengenai data yang telah diklasifikasikan atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan.

Informasi atau dalam bahasa Inggrisnya adalah *information*, berasal dari kata *informacion* bahasa Prancis. Kata tersebut diambil dari bahasa Latin, yaitu *informationem* yang artinya konsep, ide, garis besar. Informasi adalah suatu data yang sudah diolah atau diproses sehingga menjadi suatu bentuk yang memiliki arti bagi penerima informasi yang memiliki nilai bermanfaat (Rusdiana, 2014).

2.2.1 Fungsi Informasi

Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Ketika berbekal informasi seseorang dapat mengambil keputusan dengan baik. Akan tetapi, dalam pengambilan keputusan yang kompleks, informasi hanya dapat menambah kemungkinan kepastian atau mengurangi berbagai macam pilihan (Sutabri, 2012).

2.2.2 Nilai Informasi

Nilai dari informasi ditentukan dari dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dikatakan bernilai apabila manfaat yang diperoleh lebih berharga dibandingkan dengan biaya untuk mendapatkannya. Berikut ini adalah nilai informasi berdasarkan atas 10 (sepuluh) sifat, yaitu (Sutabri, 2012):

1. Mudah diperoleh
Sifat ini menunjukkan kemudahan dan kecepatan untuk memperoleh informasi. Kecepatannya dapat diukur, misalnya 1 menit versus 24 jam. Akan tetapi berapa nilainya bagi pemakai informasi sulit untuk mengukurnya.
2. Luas dan lengkap
Sifat ini menunjukkan kelengkapan isi informasi. Hal ini tidak hanya mengenai volumenya, akan tetapi juga mengenai keluaran informasinya. Sifat ini sangat kabur dan karena itu sulit untuk mengukurnya.
3. Ketelitian
Sifat ini berhubungan dengan tingkat kebebasan dari kesalahan keluaran informasi. Pada volume data yang besar biasanya terdapat dua jenis kesalahan, yakni kesalahan pencatatan dan kesalahan perhitungan.
4. Kecocokan
Sifat ini menunjukkan seberapa baik keluaran informasi dalam hubungannya dengan permintaan para pemakai. Isi informasi harus ada hubungannya dengan masalah yang sedang dihadapi sedangkan semua keluaran yang lainnya tidak berguna. Sifat ini sulit mengukurnya.
5. Ketepatan waktu
Sifat ini berhubungan dengan waktu yang dilalui, yang lebih pendek dari siklus untuk mendapatkan informasi. Masukan pengolahan dan pelaporan keluaran kepada para pemakai biasanya tepat waktu. Dalam beberapa hal, ketepatan waktu dapat diukur. Misalnya berapa banyak penjualan dapat ditingkatkan dengan menanggapi permintaan pelanggan mengenai ketersediaan barang-barang inventaris.

6. Kejelasan

Sifat ini menunjukkan tingkat kejelasan informasi. Informasi hendaknya terbebas dari istilah-istilah yang tidak jelas.

7. Keluwesan

Sifat ini berhubungan dengan apakah informasi tersebut dapat digunakan untuk membuat lebih dari satu keputusan, tetapi apakah juga dapat digunakan untuk lebih dari seorang pengambil keputusan. Sifat ini sulit mengukurnya, akan tetapi dalam beberapa hal dapat diukur dengan suatu nilai tertentu.

8. Dapat dibuktikan

Sifat ini menunjukkan sejauh mana informasi itu dapat diuji oleh beberapa pemakai hingga sampai didapatkan kesimpulan yang sama.

9. Tidak ada prasangka

Sifat ini berhubungan dengan ada tidaknya keinginan untuk mengubah informasi tersebut guna mendapatkan kesimpulan yang telah diarahkan sebelumnya.

10. Dapat diukur

Sifat ini menunjukkan hakikat informasi yang dihasilkan oleh sistem informasi formal. Meskipun kabar angin, desas-desus, dugaan-dugaan, klenik, dan lainnya juga sering dianggap sebagai informasi, namun hal-hal tersebut berada di luar lingkup pembahasan.

Informasi bernilai sempurna apabila pengambil keputusan dapat mengambil keputusan secara optimal dalam setiap hal, dan bukan keputusan yang rata-rata akan menjadi optimal dan untuk menghindari kejadian-kejadian yang akan mendatangkan kerugian.

2.2.3 Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timeliness*) dan relevan (*relevance*) (Sutabri, 2012).

1. Akurat (*Accurate*)

Informasi harus bebas dari kesalahan–kesalahan dan tidak menyesatkan. Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya.

2. Tepat Waktu (*Timelines*)

Informasi yang datang kepada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi, karena informasi merupakan suatu landasan dalam mengambil sebuah keputusan dimana bila pengambilan keputusan terlambat maka akan berakibat fatal untuk organisasi.

3. Relevan (*Relevance*)

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda. Menyampaikan informasi tentang penyebab kerusakan mesin produksi kepada akuntan perusahaan tentunya kurang relevan. Akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi disampaikan untuk ahli teknik merupakan informasi yang kurang relevan, tetapi akan sangat relevan untuk seorang akuntan perusahaan.

4. Kelengkapan

Para pengguna harus memperoleh informasi yang menyajikan sesuatu gambaran lengkap atas suatu masalah tertentu atau solusinya.

2.3 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen pembentuk sistem yang memiliki keterkaitan antara satu komponen dan komponen lain yang bertujuan menghasilkan informasi dalam bidang tertentu (Rusdiana, 2014).

Sistem informasi (*information system*) merupakan kombinasi teratur dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data yang mengumpulkan, mengubah, menyebarkan informasi dalam sebuah organisasi (Yakub, 2012).

2.4 Konsep Dasar Sistem Informasi

Suatu organisasi terdiri atas sejumlah unsur, orang-orang yang mempunyai berbagai peran, kegiatan ataupun tugas yang harus diselesaikan, tempat kerja, wewenang, serta hubungan komunikasi yang mengikat organisasi tersebut. Sistem informasi merupakan penerapan sistem di dalam organisasi untuk mendukung informasi yang dibutuhkan oleh semua tingkat manajemen. Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Informasi diperoleh dari sistem informasi (Sutabri, 2012).

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan oleh pihak luar tertentu (Sutabri, 2012).

2.5 Komponen dan Tipe Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data, dan blok kendali (Sutabri, 2012). Sebagai suatu sistem, keenam blok tersebut saling berinteraksi satu dengan yang lain membentuk satu kesatuan untuk mencapai sasaran.

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Yang dimaksud dengan input di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok model (*model block*)

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

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

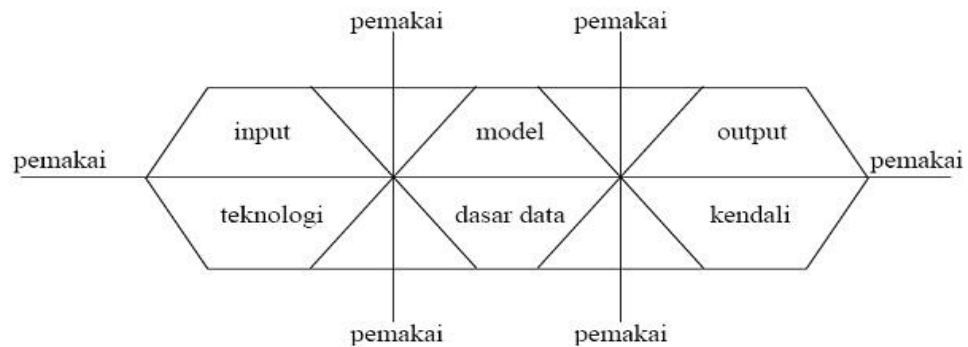
Teknologi merupakan tool box dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian sistem secara keseluruhan. Teknologi terdiri dari 3(tiga) bagian utama, yaitu teknis (*brainware*), perangkat lunak (*software*), perangkat keras (*hardware*).

5. Blok basis data (*database block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan perangkat lunak digunakan untuk memanipulasinya. 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 penyimpanan. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*database management system*).

6. Blok kendali (*control block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan pada sistem itu sendiri, ketidak-efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk menyakinkan bahwa hal-hal yang dapat merusak sistem dicegah dan bila terlanjur terjadi maka kesalahan-kesalahan dapat dengan cepat diatasi (dapat dilihat pada Gambar II.2).



Gambar II.2 Komponen Sistem Informasi

Sumber: (Sutabri, 2012)

Manajemen membutuhkan informasi untuk mendukung pengambilan keputusan yang akan dilakukan. Sumber informasi untuk pengambilan keputusan manajemen bisa didapatkan dari informasi eksternal dan informasi internal. Informasi internal dapat diperoleh dari sistem informasi yang berupa hasil pengolahan data elektronik (PDE) atau data non PDE.

Secara teori, komputer tidak harus digunakan di dalam sistem, akan tetapi kenyataannya tidaklah mungkin suatu sistem yang kompleks dapat melibatkan elemen non komputer dan elemen komputer.

Tipe sistem informasi adalah sebagai berikut:

- a. Sistem informasi Akuntansi
- b. Sistem informasi Pemasaran
- c. Sistem informasi Manajemen Persediaan
- d. Sistem informasi Personalia
- e. Sistem informasi Distribusi
- f. Sistem informasi Pembelian
- g. Sistem informasi Kekayaan
- h. Sistem informasi Analisis Kredit
- i. Sistem informasi Penelitian dan Pengembangan
- j. Sistem informasi Teknik

Semua sistem informasi tersebut dimaksudkan untuk memberikan informasi kepada semua tingkat manajemen, mulai manajemen tingkat bawah, manajemen tingkat menengah, hingga manajemen tingkat atas (Sutabri, 2012).

2.6 Pengertian Pengendalian

Pengendalian merupakan kegiatan yang dilakukan untuk menjamin agar kegiatan produksi dan operasi yang dilaksanakan sesuai dengan apa yang telah di rencanakan (Assauri, 2008).

Pengendalian merupakan ketentuan apa yang harus dilaksanakan, menilai dan mengoreksi pelaksanaannya bila perlu dengan maksud supaya pelaksanaan pekerjaan sesuai dengan rencana semula (Ayuni, 2012).

2.7 Pengertian Kualitas

Kualitas adalah keseluruhan fitur dan karakteristik sebuah barang atau jasa yang menggunakan kemampuannya untuk memenuhi kebutuhan yang dijanjikan dan tersirat (Heizer & Render, 2014).

Kualitas terpenuhi atau terlampauinya espetasi pelanggan melalui produk yang dihasilkan perusahaan kualitas yang baik adalah produk yang dihasilkan oleh perusahaan telah sesuai dengan spesifikasi yang telah ditentukan oleh perusahaan (Harsanto, 2013).

Kualitas adalah sebuah istilah subjektif yang masing-masing setiap orang memiliki definisi sendiri-sendiri. Dalam penggunaan teknis, kualitas dapat memiliki dua makna (Russell & Taylor, 2011):

1. Karakteristik dari produk atau jasa yang menanggung pada kemampuannya untuk memuaskan dinyatakan atau tersirat kebutuhan.
2. Sebuah produk atau layanan gratis dari kekurangan, jelas kualitas dapat didefinisikan dalam banyak cara, tergantung pada siapa yang mendefinisikan dan kemana produk atau jasa itu mengacu.

2.8 Pengertian Pengendalian Kualitas

Pengendalian kualitas adalah suatu proses yang pada intinya adalah menjadikan entitas sebagai peninjau kualitas dari semua faktor yang terlibat dalam kegiatan produksi. Pengertian pengendalian kualitas menurut para ahli adalah sebagai berikut:

1. Pengawasan mutu merupakan usaha untuk mempertahankan mutu atau kualitas dari barang yang dihasilkan, agar sesuai dengan spesifikasi produk yang telah ditetapkan berdasarkan kebijaksanaan kepemimpinan perusahaan (Assauri, 2008).
2. *“Quality control is the operational techniques and activities used to fulfill requirements for quality (Gaspersz, 2005).”*

Berdasarkan pengertian-pengertian di atas, maka dapat disimpulkan bahwa pengendalian kualitas merupakan suatu teknik dan aktivitas atau tindakan yang terencana yang dilakukan untuk mencapai, mempertahankan, dan meningkatkan kualitas suatu produk dan jasa agar sesuai dengan standar yang telah ditetapkan dan dapat memenuhi kepuasan konsumen.

2.8.1 Tujuan Pengendalian Kualitas

Pengendalian kualitas memiliki tujuan tertentu. Adapun tujuan dari pengendalian kualitas adalah (Assauri, 2008):

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.

2.8.2 Faktor Pengendalian Kualitas

Faktor yang mempengaruhi pengendalian kualitas adalah (Yamit, 2013):

1. Kemampuan proses

Batas-batas yang ingin dicapai haruslah disesuaikan dengan kemampuan proses yang ada. Tidak ada gunanya mengendalikan suatu proses dalam batas-batas yang melebihi kemampuan atau kesanggupan proses yang ada.

2. Spesifikasi yang berlaku

Spesifikasi hasil produksi yang ingin dicapai harus dapat berlaku, bila ditinjau dari segi kemampuan proses dan keinginan atau kebutuhan konsumen yang ingin dicapai dari hasil produksi tersebut.

3. Tingkat ketidaksesuaian yang dapat diterima

Tujuan dilakukan pengendalian suatu proses adalah dapat mengurangi produk yang berada di bawah standar seminimal mungkin. Tingkat pengendalian yang diberlakukan tergantung pada banyaknya produk yang berada di bawah standar yang dapat diterima.

4. Biaya kualitas

Biaya kualitas sangat mempengaruhi tingkat pengendalian kualitas dalam menghasilkan produk dimana biaya kualitas mempunyai hubungan yang positif dengan terciptanya produk yang berkualitas.

- a. Biaya Pencegahan (*Prevention Cost*)

Biaya yang terjadi untuk mencegah terjadinya kerusakan produk.

- b. Biaya Deteksi/Penilaian (*Detection/Appraisal Cost*)

Biaya yang timbul untuk menentukan apakah produk atau jasa yang dihasilkan telah sesuai dengan persyaratan-persyaratan kualitas sehingga dapat menghindari kesalahan dan kerusakan sepanjang proses produksi.

- c. Biaya Kegagalan Internal (*Internal Failure Cost*)

Biaya yang terjadi karena adanya ketidaksesuaian dengan persyaratan dan terdeteksi sebelum barang atau jasa tersebut dikirim ke pihak luar (pelanggan).

d. Biaya Kegagalan Eksternal (*External Failure Cost*)

Biaya yang terjadi karena produk atau jasa tidak sesuai dengan persyaratan-persyaratan yang diketahui setelah produk tersebut dikirimkan kepada pelanggan.

2.9 Pengertian SPC (*Statistical Process Control*)

Statistical Process Control (SPC) merupakan ilmu yang mempelajari tentang teknik/metode pengendalian kualitas berdasarkan prinsip statistik (Montgomery, 2009). Hal ini bertujuan untuk membuat produk agar sesuai dengan spesifikasi yang sudah ditetapkan dari awal proses hingga akhir proses. SPC juga digunakan untuk mengumpulkan dan menganalisis data hasil pemeriksaan terhadap sampel dalam kegiatan pengawasan kualitas produk (Arifianti, 2013). Selain itu SPC juga digunakan untuk mengukur kualitas sekarang dari produk atau jasa dan mendeteksi apakah proses barang atau jasa mengalami perubahan yang akan mempengaruhi kualitas (Haizer & Render, 2006).

Statistical Process Control (SPC) adalah teknik penyelesaian masalah yang digunakan untuk menganalisis data dalam rangka mengendalikan dan memperbaiki kualitas suatu produksi sehingga sesuai dengan yang diharapkan (Montgomery, 2009). Alat ini digunakan untuk mengendalikan proses produksi secara berkesinambungan dan mengidentifikasi kerusakan yang terjadi ketika proses produksi berlangsung (Sultana, Razive, & Azeem, 2009). Metode ini merupakan salah satu metode yang akurat, yang mampu meminimalkan dan meniadakan cacat (*zero defect*) pada produk serta mempertahankan dan memaksimalkan kesuksesan suatu perusahaan (Kartika, 2013).

Manfaat/keuntungan melakukan pengendalian kualitas secara *statistic* (Assauri, 2008) adalah :

1. Pengendalian (*control*), di mana penyelidikan yang diperlukan untuk dapat menetapkan *statistical control* mengharuskan bahwa syarat-syarat

kualitas pada situasi itu dan kemampuan prosesnya telah dipelajari hingga mendetail. Hal ini akan menghilangkan beberapa titik kesulitan tertentu, baik dalam spesifikasi maupun dalam proses.

2. Pengerjaan kembali barang-barang yang telah *scrap-rework*. Dengan dijalankan pengontrolan, maka dapat dicegah terjadinya penyimpangan-penyimpangan dalam proses. Sebelum terjadi hal-hal yang serius dan akan diperoleh kesesuaian yang lebih baik antara kemampuan proses (*process capability*) dengan spesifikasi, sehingga banyaknya barang-barang yang diapkir (*scrap*) dapat dikurangi sekali. Dalam perusahaan pabrik sekarang ini, biaya-biaya bahan sering kali mencapai 3 sampai 4 kali biaya buruh, sehingga dengan perbaikan yang telah dilakukan dalam hal pemanfaatan bahan dapat memberikan penghematan yang menguntungkan.
3. Biaya-biaya pemeriksaan, karena *Statistical Process Control* dilakukan dengan jalan mengambil sampel-sampel dan mempergunakan *sampling techniques*, maka hanya sebagian saja dari hasil produksi yang perlu untuk diperiksa. Akibatnya maka hal ini akan dapat menurunkan biaya-biaya pemeriksaan.

Pengendalian kualitas secara *statistic* dengan menggunakan SPC (*Statistical Processing Control*) mempunyai 7 alat statistik utama yang dapat digunakan sebagai alat bantu untuk mengendalikan kualitas, antara lain yaitu; *check Sheet*, histogram, *control chart*, diagram pareto, diagram sebab akibat, *scatter diagram*, dan diagram proses (Haizer & Render, 2006).

1. Lembar Pemeriksaan (*Check Sheet*).

Check Sheet atau lembar pemeriksaan merupakan alat pengumpul dan penganalisis data yang disajikan dalam bentuk tabel yang berisi data jumlah barang yang diproduksi dan jenis ketidaksesuaian beserta dengan jumlah yang dihasilkannya. Tujuan digunakannya check sheet ini adalah untuk mempermudah proses pengumpulan data dan analisis, serta untuk mengetahui area permasalahan berdasarkan frekuensi dari jenis atau penyebab dan mengambil keputusan untuk melakukan perbaikan atau tidak. Pelaksanaannya dilakukan dengan cara mencatat frekuensi munculnya

karakteristik suatu produk yang berkenaan dengan kualitasnya. Data tersebut digunakan sebagai dasar untuk mengadakan analisis masalah kualitas.

Adapun manfaat dipergunakannya check sheet yaitu sebagai alat untuk :

- a. Mempermudah pengumpulan data terutama untuk mengetahui bagaimana suatu masalah terjadi.
- b. Mengumpulkan data tentang jenis masalah yang sedang terjadi.
- c. Menyusun data secara otomatis sehingga lebih mudah untuk dikumpulkan.
- d. Memisahkan antara opini dan fakta.

2. Diagram Sebar (*Scatter Diagram*).

Scatter diagram atau disebut juga dengan peta korelasi adalah grafik yang menampilkan hubungan antara dua variabel apakah hubungan antara dua variabel tersebut kuat atau tidak, yaitu antara faktor proses yang mempengaruhi proses dengan kualitas produk. Pada dasarnya diagram sebar (*scatter diagram*) merupakan suatu alat interpretasi data yang digunakan untuk menguji bagaimana kuatnya hubungan antara dua variabel dan menentukan jenis hubungan dari dua variabel tersebut, apakah positif, negatif, atau tidak ada hubungan. Dua variabel yang ditunjukkan dalam diagram sebar dapat berupa karakteristik kuat dan faktor yang mempengaruhinya.

3. Diagram Sebab-Akibat (*Cause and Effect Diagram*)

Diagram ini disebut juga diagram tulang ikan (*fishbone chart*) dan berguna untuk memperlihatkan faktor-faktor utama yang berpengaruh pada kualitas dan mempunyai akibat pada masalah yang kita pelajari. Selain itu, kita juga dapat melihat faktor-faktor yang lebih terperinci yang berpengaruh dan mempunyai akibat pada faktor utama tersebut yang dapat kita lihat pada panah-panah yang berbentuk tulang ikan. Diagram sebab-akibat ini pertama kali dikembangkan pada tahun 1950 oleh seorang pakar kualitas dari Jepang yaitu Dr. Kaoru Ishikawa yang menggunakan uraian grafis dari unsur-unsur proses untuk menganalisa

sumber-sumber potensial dari penyimpangan proses. Faktor-faktor penyebab utama ini dapat dikelompokkan dalam :

- a. Material (bahan baku).
- b. Machine (mesin).
- c. Man (tenaga kerja).
- d. Method (metode).
- e. Environment (lingkungan)

Adapun kegunaan dari diagram sebab akibat adalah:

- a. Membantu mengidentifikasi akar penyebab masalah.
- b. Menganalisa kondisi yang sebenarnya yang bertujuan untuk memperbaiki peningkatan kualitas.
- c. Membantu membangkitkan ide-ide untuk solusi suatu masalah.
- d. Membantu dalam pencarian fakta lebih lanjut.
- e. Mengurangi kondisi-kondisi yang menyebabkan ketidaksesuaian produk dengan keluhan konsumen.
- f. Menentukan standarisasi dari operasi yang sedang berjalan atau yang akan dilaksanakan.
- g. Merencanakan tindakan perbaikan.

Adapun langkah-langkah dalam membuat diagram sebab akibat adalah sebagai berikut:

- a. Mengidentifikasi masalah utama.
- b. Menempatkan masalah utama tersebut disebelah kanan diagram.
- c. Mengidentifikasi penyebab minor dan meletakkannya pada diagram utama.
- d. Mengidentifikasi penyebab minor dan meletakkannya pada penyebab mayor.
- e. Diagram telah selesai, kemudian dilakukan evaluasi untuk menentukan penyebab sesungguhnya.

4. Diagram Pareto (*Pareto Analysis*)

Diagram pareto pertama kali diperkenalkan oleh Alfredo Pareto dan digunakan pertama kali oleh Joseph Juran. Diagram pareto adalah grafik balok dan grafik baris yang menggambarkan perbandingan masing-masing jenis data terhadap keseluruhan. Dengan memakai diagram pareto, dapat terlihat masalah mana yang dominan sehingga dapat mengetahui prioritas penyelesaian masalah. Fungsi Diagram pareto adalah untuk mengidentifikasi atau menyeleksi masalah utama untuk peningkatan kualitas dari yang paling besar ke yang paling kecil.

5. Diagram Alir/ Diagram Proses (*Process Flow Chart*)

Diagram alir secara grafis menunjukkan sebuah proses atau sistem dengan menggunakan kotak dan garis yang saling berhubungan. Diagram ini cukup sederhana, tetapi merupakan alat yang sangat baik untuk mencoba memahami sebuah proses atau menjelaskan langkah-langkah sebuah proses.

6. Histogram

Histogram adalah suatu alat yang membantu untuk menentukan variasi dalam proses. Berbentuk diagram batang yang menunjukkan tabulasi dari data yang diatur berdasarkan ukurannya. Tabulasi data ini umumnya dikenal dengan distribusi frekuensi. Histogram menunjukkan karakteristik-karakteristik dari data yang dibagi-bagi menjadi kelas-kelas. Histogram dapat berbentuk “normal” atau berbentuk seperti lonceng yang menunjukkan bahwa banyak data yang terdapat pada nilai rata-ratanya. Bentuk histogram yang miring atau tidak simetris menunjukkan bahwa banyak data yang tidak berada pada nilai rata-ratanya tetapi kebanyakan data nya berada pada batas atas atau bawah.

7. Peta Kendali (*Control Chart*)

Peta kendali adalah suatu alat yang secara grafis digunakan untuk memonitor dan mengevaluasi apakah suatu aktivitas/proses berada dalam pengendalian kualitas secara statistika atau tidak sehingga dapat memecahkan masalah dan menghasilkan perbaikan kualitas. Peta kendali menunjukkan adanya perubahan data dari waktu ke waktu, tetapi tidak

menunjukkan penyebab penyimpangan meskipun penyimpanan itu akan terlihat pada peta kendali.

2.9.1 Contoh Kasus *Statistical Process Control* (SPC)

Dalam Jurnal Amanda Nur Cahyawati dan Suci Septiasari disini menjelaskan terjadinya kecacatan pada produk lampu, berikut deskripsi hal apa saja yang membuat produks lampu bisa terjadi cacat:

Jenis Cacat	Deskripsi Produk Cacat
Salah potong	Plat yang dipotong tidak sesuai dengan spesifikasi produk.
Salah <i>bending/punch</i>	Plat yang sudah dipotong akan <i>dibending/punch</i> . Plat yang di tekuk (<i>bending</i>) tidak sesuai desain produk atau pemberian lubang (<i>punch</i>) plat salah (tidak sesuai spesifikasi produk).
Pengelasan komponen tidak rekat/kuat	Plat yang di las tidak kuat sehingga bagian komponen terlepas atau tidak menyatu layaknya kerangka rumah lampu.
Pelapisan fosfat kurang merata	Produk yang tidak dilapisi fosfat mengakibatkan mudahnya terkena korosi yang akan berdampak pada tidak bertahan lamanya penggunaan produk
Pengecatan kurang merata	Permukaan produk yang kasar karena tidak meratanya pengecatan.
Komponen rusak/lecet	Bagian komponen produk rumah lampu rusak/lecet karena benturan saat membawa komponen menuju ke <i>station</i> perakitan atau saat proses <i>assembly</i>

Gambar II.3 Contoh Jenis Cacat

Sumber: (Amanda Nur Cahyawati & Suci Septiasari, 2019)

Dan berikut hasil cacat produk lampu yang dihasilkan pada setiap bulannya:

Bulan	Jenis Cacat (<i>Defect</i>)						Total produk <i>Defect</i>
	Salah Potong	Salah <i>bending/punch</i>	Las tidak rekat	Pelapisan tidak merata	Pengecatan tidak merata	Komponen rusak/lecet	
Januari	11	8	5	10	9	5	48
Februari	10	5	8	4	6	8	41
Maret	9	5	9	6	9	9	47
April	5	4	2	4	4	3	22
Mei	7	2	6	5	5	12	37
Juni	4	2	8	4	3	4	25
Juli	12	5	10	8	9	9	53
Agustus	15	8	12	9	10	15	69
September	19	3	5	16	9	16	68
Oktober	21	1	9	18	12	18	79
November	8	3	10	2	19	9	51
Desember	9	2	6	8	2	10	37
Total	130	48	90	94	97	118	577

Gambar II.4 Contoh Hasil Perhitungan Cacat

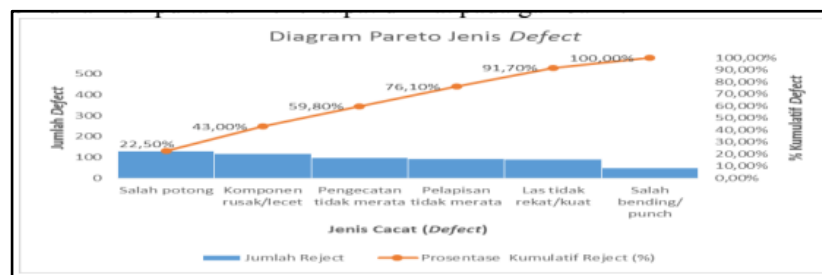
Sumber: (Amanda Nur Cahyawati & Suci Septiasari, 2019)

Pengolahan data defect diurutkan dari jumlah terbesar hingga terkecil lalu dihitung prosentase defect dan prosentase kumulatif defect. Prosentase data defect dapat dilihat pada Gambar dibawah ini.

No	Jenis Cacat (<i>Defect</i>)	Jumlah <i>Defect</i>	Prosentase <i>Defect</i>	Prosentase Kumulatif <i>Defect</i>
1	Salah potong	130	22,50%	22,50%
2	Komponen rusak/lecet	118	20,50%	43,00%
3	Pengecatan tidak merata	97	16,80%	59,80%
4	Pelapisan tidak merata	94	16,30%	76,10%
5	Las tidak rekat/kuat	90	15,60%	91,70%
6	Salah <i>bending/punch</i>	48	8,30%	100,00%
	Total	577	100 %	

Gambar II.5 Contoh Perhitungan Diagram Pareto
Sumber: (Amanda Nur Cahyawati & Suci Septiasari, 2019)

Data *defect* diidentifikasi menggunakan diagram pareto. Analisis penggunaan diagram pareto dilakukan untuk mengetahui tingkat prioritas bagian cacat yang memiliki jumlah dan presentase terbanyak untuk dapat menentukan langkah pencegahan atau perbaikannya. Diagram pareto dari data *defect* produksi rumah lampu tahun 2017 dapat dilihat pada gambar dibawah ini:



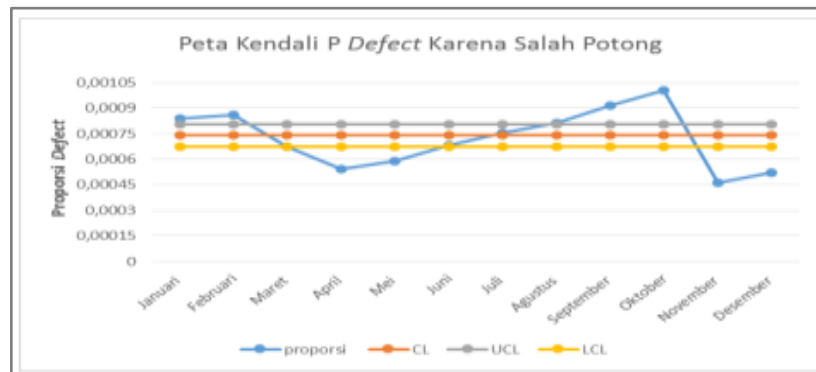
Gambar II.6 Contoh Diagram Pareto
Sumber: (Amanda Nur Cahyawati & Suci Septiasari, 2019)

Langkah selanjutnya melakukan perhitungan dengan cara membuat peta kendali p. Pembuatan peta kendali perlu mengetahui *Center Line* (CL), *Upper Control Limit* (UCL), dan *Lower Control Limit* (LCL) dengan menggunakan rumus yang sudah ada. Berikut contoh gambar dari kasus *defect* yang salah potong:

Bulan	Jumlah Produksi	Jenis Cacat (<i>Defect</i>) Salah Potong	Proporsi <i>Defect</i>	UCL	CL	LCL
Januari	13144	11	0,00084	0,000805	0,00074	0,000675
Februari	11606	10	0,00086	0,000805	0,00074	0,000675
Maret	13384	9	0,00067	0,000805	0,00074	0,000675
April	9207	5	0,00054	0,000805	0,00074	0,000675
Mei	11871	7	0,00059	0,000805	0,00074	0,000675
Juni	5871	4	0,00068	0,000805	0,00074	0,000675
Juli	15934	12	0,00075	0,000805	0,00074	0,000675
Agustus	18437	15	0,00081	0,000805	0,00074	0,000675
September	20799	19	0,00091	0,000805	0,00074	0,000675
Oktober	20933	21	0,001	0,000805	0,00074	0,000675
November	17270	8	0,00046	0,000805	0,00074	0,000675
Desember	17187	9	0,00052	0,000805	0,00074	0,000675
Total	175643	130				

Gambar II.7 Contoh Perhitungan Peta Kendali
Sumber: (Amanda Nur Cahyawati & Suci Septiasari, 2019)

Berikut contoh gambar peta kendali yang telah dibuat dari kasus *defect* yang salah potong:



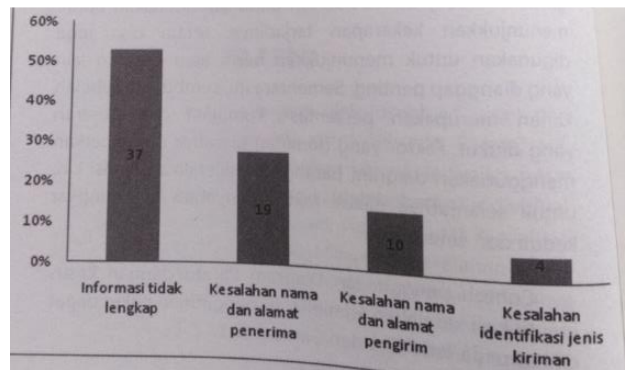
Gambar II.8 Contoh Peta Kendali

Sumber: (Amanda Nur Cahyawati & Suci Septiasari, 2019)

2.9.2 Diagram Pareto

Diagram Pareto dikembangkan oleh Joseph M. Juran, dan diberi nama sesuai dengan nama Vilfredo Pareto, ahli ekonomi Italia menemukan bahwa sebagian besar kekayaan di dunia hanya dimiliki oleh beberapa individual. Dengan menggunakan Diagram Pareto, dapat dievaluasi hal-hal yang menyebabkan terjadinya permasalahan secara spesifik berdasarkan dampak atau frekuensi terjadinya permasalahan. Selanjutnya, dari analisis tersebut dapat ditentukan faktor-faktor dominan yang memiliki pengaruh paling besar yang menyebabkan terjadinya permasalahan, untuk kemudian dibuat prioritas perbaikannya. Kegunaan dari Diagram Pareto sebagai berikut.

- Menunjukkan masalah utama/pokok masalah yang dominan.
- Menyatakan perbandingan masing-masing masalah terhadap permasalahan secara keseluruhan.
- Menunjukkan perbandingan masalah sebelum dan sesudah perbaikan.



Gambar II.9 Diagram Pareto
Sumber: (Soemohadiwidjojo, 2017)

2.9.3 Peta Kendali

Control Chart (Peta Kendali), *Statistical Process Control* (SPC), atau diagram *Shewart* adalah alat statistik yang digunakan untuk mengukur dan mengevaluasi variasi yang terjadi dalam sebuah proses, dengan tujuan melihat apakah sebuah proses berada dalam kondisi terkendali (Soemohadiwidjojo, 2017). *Control Chart* biasanya digunakan untuk data yang diukur seiring dengan waktu, dan bermanfaat untuk mengidentifikasi masalah-masalah yang mungkin terjadi, atau insiden-insiden yang tidak biasa sehingga dapat dilakukan tindakan yang tepat sebagai atas permasalahan tersebut. *Control Chart* terdiri atas hal-hal berikut.

1. *Upper Control Limit* / batas kendali atas (UCL)
Menggunakan garis batas untuk suatu penyimpangan yang masih diijinkan.
2. *Central line* / garis pusat atau tengah (CL)
Merupakan garis yang melambangkan tidak adanya penyimpangan dari karakteristik sampel
3. *Lower Control limit* / batas kendali bawah (LCL)
Merupakan garis batas bawah untuk suatu penyimpangan dari karakteristik sampel

Berikut merupakan rumus perhitungan peta kendali.

1. Menghitung persentase ketidaksesuaian

$$p = \frac{np}{n}$$

p = presentase ketidak sesuaian (cacat)

np = jumlah ketidaksesuaian dalam subgrup

n = jumlah yang diperiksa dalam subgroup

subgroup : hari yang diperiksa ke-

2. Batas Pusat (CL)

$$CL = \bar{P} = \frac{\sum np}{\sum n}$$

CL = *Central Line* (Garis Pusat Rata Rata Kerusakan Produk)

$\sum np$ = jumlah cacat

$\sum n$ = jumlah yang diperiksa

3. Batas Atas (UCL)

$$UCL = \bar{P} + 3 \sqrt{\frac{\bar{P}(1 - \bar{P})}{n}}$$

UCL = *Upper Control Limit* (Batas Kendali Atas)

\bar{P} = Rata Rata Ketidak Sesuaian Produk

n = Jumlah Produksi

4. Batas Bawah (LCL)

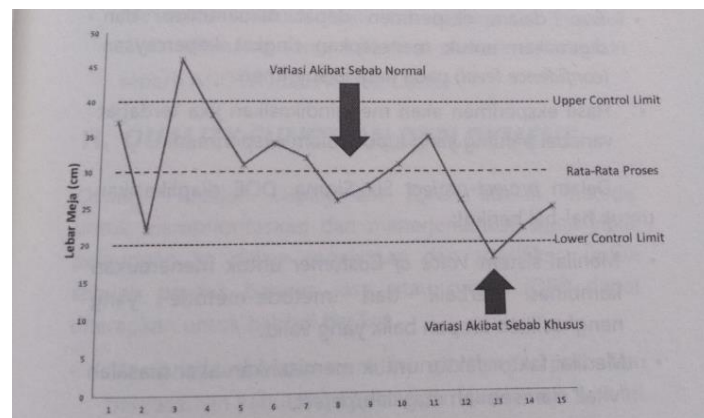
$$LCL = \bar{P} - 3 \sqrt{\frac{\bar{P}(1 - \bar{P})}{n}}$$

LCL = *Lower Control Limit* (Batas Kendali Atas)

\bar{P} = Rata Rata Ketidak Sesuaian Produk

n = Jumlah Produksi

5. Membuat Peta kendali



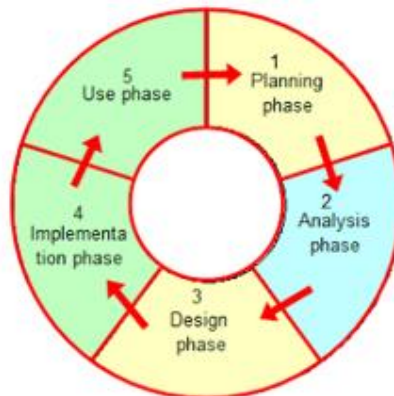
Gambar II.10 Peta Kendali
Sumber: (Soemohadiwidjojo, 2017)

2.10 *System Development Life Cycle (SDLC)*

System Development Life Cycle atau yang disingkat SDLC adalah aplikasi dari pendekatan sistem bagi pengembangan suatu sistem informasi (Yakub, 2012). Tidak dibutuhkan waktu lama bagi seorang pengembang sistem yang pertama untuk mengetahui bahwa terdapat beberapa tahapan pekerjaan pengembangan yang perlu dilakukan dalam urutan tertentu jika suatu proyek ingin memiliki kemungkinan berhasil yang paling besar. Tahapan-tahapan tersebut adalah:

1. Perencanaan
2. Analisis
3. Desain
4. Implementasi
5. Penggunaan

Proyek direncanakan dan sumber-sumber daya yang dibutuhkan untuk melakukan pekerjaan kemudian disatukan. Sistem yang ada juga dianalisis untuk memahami masalah dan menentukan persyaratan fungsionalitas dari sistem yang baru. Sistem baru ini kemudian dirancang dan diimplementasikan. Setelah implementasi, sistem kemudian digunakan idealnya untuk jangka waktu yang lama.

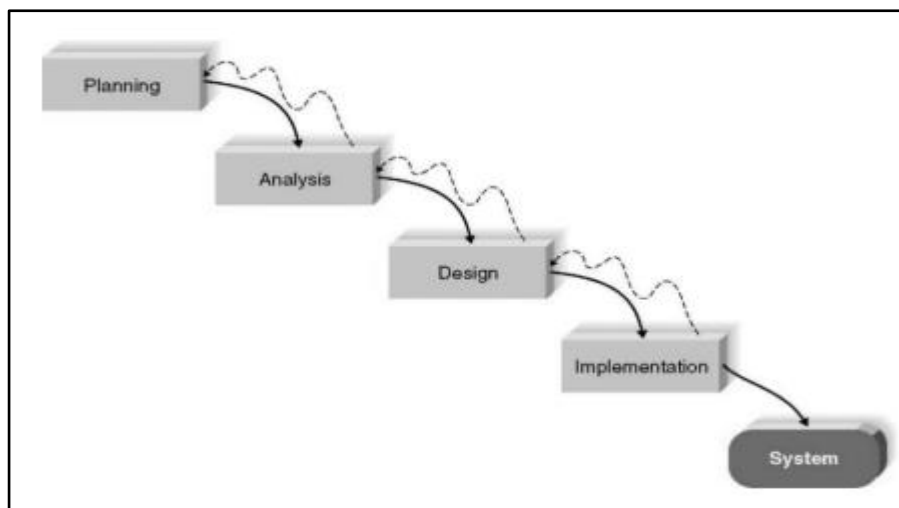


Gambar II.11 Pola Melingkar dari Siklus Hidup Sistem
Sumber: (Yakub, 2012)

Gambar di atas mengilustrasikan sifat melingkar dari siklus hidup sistem. Ketika sebuah sistem telah melampaui masa manfaatnya dan harus diganti, satu siklus hidup baru akan dimulai dengan diawali oleh tahap perencanaan.

2.11 Model Waterfall

Contoh model yang masih digunakan hingga saat ini adalah *waterfall development* atau disebut juga dengan model pengembangan “air mancur” (Dennis, Wixom, & Tegarden, 2015). Dengan metode pengembangan *waterfall*, analis dan pengguna atau *user* memproses secara berurutan dari satu fase ke fase berikutnya. Biasanya proses dalam satu fase sangat panjang dan ditampilkan dalam sponsor proyek atau *project sponsor* untuk disetujui sehingga dapat berlanjut ke fase selanjutnya. Suatu fase atau tahapan akan berlanjut jika sponsor proyek telah menyetujui tahapan tersebut. Model ini disebut dengan *waterfall development*, karena tahap demi tahapan bergerak maju dengan cara yang sama seperti air mancur. Tetapi, dalam model ini suatu tahapan sulit untuk bergerak mundur, (contohnya dari tahap desain mundur ke tahap analis). Berikut penggambaran model *waterfall* seperti pada gambar II.6 berikut:



Gambar II.12 Penggambaran Metode *Waterfall*
sumber: (Dennis, Wixom, & Tegarden, 2015)

2.12 Analisis PIECES

Metode PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik (Ragil, 2010). Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi, dan pelayanan pelanggan. Analisis ini disebut dengan *PIECES Analysis (Performance, Information, Economy, Control, Efficiency, and Service)*.

Analisis PIECES ini sangat penting untuk dilakukan sebelum mengembangkan sebuah sistem informasi karena dalam analisis ini biasanya akan ditemukan beberapa masalah utama maupun masalah yang bersifat gejala dari masalah utama. Metode ini menggunakan enam variabel evaluasi, yaitu:

1. *Performance* (kinerja)
Jumlah kerja selama periode tertentu. Pada bagian ini dideskripsikan situasi saat ini tentang jumlah kerja yang dibutuhkan untuk melakukan serangkaian kerja tertentu dalam satuan orang jam, orang hari, orang bulan.
2. *Information* (informasi)
Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki sehingga kualitas informasi yang dihasilkan menjadi semakin baik. Informasi yang disajikan haruslah benar-benar mempunyai nilai yang berguna. Hal ini dapat diukur dengan :
Keluaran (*outputs*): Suatu sistem dalam memproduksi keluaran.
Masukan (*inputs*): Dalam memasukkan suatu data sehingga kemudian diolah untuk menjadi informasi yang berguna.
3. *Economic* (ekonomi)
Pada bagian ini dideskripsikan situasi saat ini tentang biaya yang dikeluarkan untuk memproduksi informasi, melakukan proses bisnis, dan mengambil keputusan tidak diketahui jumlahnya dan pembiayaannya. Selain itu juga di deskripsikan penyebab biaya tidak diketahui, dan dampak yang ditimbulkan ketika hal tersebut terjadi.
4. *Control* (pengendalian)
Menilai apakah prosedur yang ada saat ini masih dapat ditingkatkan sehingga kualitas pengendalian menjadi semakin baik, dan kemampuannya untuk mendeteksi kesalahan/ kecurangan menjadi semakin baik pula.
5. *Efficiency* (efisiensi)
Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki, sehingga tercapai peningkatan efisiensi operasi, dan harus lebih unggul dari pada sistem manual.
6. *Service* (layanan)

Menilai apakah prosedur yang ada saat ini masih dapat diperbaiki kemampuannya untuk mencapai peningkatan kualitas layanan. Buatlah kualitas layanan yang sangat *user friendly* untuk *end-user* (pengguna) sehingga pengguna mendapatkan kualitas layanan yang baik.

2.13 *System Requirement*

System requirements sekumpulan layanan dan batasan-batasan kebutuhan sistem adalah semua aktivitas yang harus dilakukan atau didukung oleh sistem baru dan batasan-batasan yang harus dicapai sistem baru, harus dijelaskan secara detil. (Satzinger, Jackson, & Burd, 2012). *System requirements* dibagi menjadi 2 kategori, yaitu:

1. *Functional requirements* atau kebutuhan fungsional.
Functional requirements merupakan penjelasan tentang layanan yang perlu disediakan oleh sistem, bagaimana sistem menerima dan mengolah masukan, dan bagaimana sistem mengatasi situasi-situasi tertentu. Selain itu kadang-kadang juga secara jelas menentukan apa yang tidak dikerjakan oleh sistem. *Functional Requirement* menggambarkan *system requirement* secara detil seperti *input*, *output* dan pengecualian yang berlaku. Contoh: Sistem mampu menginput dan menampilkan data pelamar kerja
2. *Non-functional requirements* atau kebutuhan non-fungsional. *Non-functional requirements* secara umum berisi batasan-batasan pada pelayanan atau fungsi yang disediakan oleh sistem. Termasuk di dalamnya adalah batasan waktu, batasan proses pembangunan, standar-standar tertentu. Karena berkaitan dengan kebutuhan sistem secara keseluruhan, maka kegagalan memenuhi kebutuhan jenis ini berakibat pada sistem secara keseluruhan. Contoh: Hak akses admin seleksi penerimaan pegawai dapat melakukan semua kegiatan pada sistem sementara hak akses untuk kepala departemen hanya bisa melihat hasil akhir seleksi.

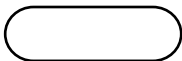
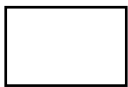

2.14 Flowmap

Flowmap adalah campuran peta dan *flowchart* yang menunjukkan pergerakan benda dari suatu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan. *Flowmap* menolong analis dan programmer untuk memecahkan masalah kedalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif lain dalam pengoperasian (Hartono, 2010). Adapun pedoman-pedoman dalam pembuatan *flowmap* adalah sebagai berikut:


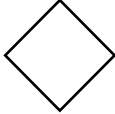


1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam *flowmap* harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Berikut simbol-simbol *flowmap* dapat dilihat pada tabel II.1 berikut ini:

Tabel II.1 Simbol-simbol *Flowmap*

Gambar	Nama	Keterangan
	Mulai/Akhir	Menunjukkan dimulainya/akhir dari sebuah proses.
	Proses	Menunjukkan kegiatan proses dari operasi komputer.
	Kegiatan Manual	Menunjukkan kegiatan manual.

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

Gambar	Nama	Keterangan
	Garis Alur	Menunjukkan arus dari proses.
	Keputusan	Keputusan dalam suatu program.
	Dokumen	Menunjukkan dokumen <i>input/output</i> baik untuk proses manual, mekanik, atau komputer.
	Keyboard	Menunjukkan <i>input</i> yang menggunakan <i>on-line keyboard</i> .

Sumber: (Hartono, 2010)

2.15 *Unified Modelling Language (UML)*

UML adalah notasi pemodelan yang banyak digunakan dalam metodologi berorientasi objek (Sarosa, 2017). Tujuan UML adalah (Sarosa, 2017):

1. Bagi pengguna, tersedia bahasa pemodelan visual yang siap digunakan dan ekspresif sehingga dapat digunakan bersama dalam pemodelan suatu sistem informasi.
2. Memberikan mekanisme pemodelan yang dapat diperluas dan dikhususkan untuk memperluas konsep dasarnya.
3. Tidak tergantung pada bahasa pemrograman tertentu maupun proses pengembangan (metodologi) tertentu.
4. Memberikan dasar pemahaman secara formal terhadap bahasa pemodelan.
5. Mendorong pertumbuhan alat-alat bantu berorientasi objek.
6. Mendukung konsep pengembangan sistem pada tingkatan tinggi (tidak terlalu teknis).
7. Mengintegrasikan praktik dan metodologi terbaik.

UML dikelola oleh *Object Management Group* (OMG) dan menjadi standar umum. UML terbaru saat ini adalah UML versi 2.5 (OMG, 2015). UML versi 2.5 terdiri atas dua kelompok diagram, yaitu:

1. Diagram Struktur (*Structure Diagram*) yang menggambarkan struktur statis suatu sistem. Dalam diagram struktur digunakan objek, atribut, operasi dan relasi antar objek. Diagram struktur terdiri dari (OMG, 2015):
 - a. *Class Diagram*
 - b. *Component Diagram*
 - c. *Object Diagram*
 - d. *Composite Structure Diagram*
 - e. *Deployment Diagram*
 - f. *Package Diagram*
 - g. *Profile Diagram*
2. Diagram Perilaku (*Behavior Diagram*) menunjukkan dinamika suatu sistem. Di dalam diagram perilaku ditunjukkan bagaimana kolaborasi dan interaksi antar objek dan kondisi internal suatu objek.
 - a. *Activity Diagram*
 - b. *Interaction Diagram*
 1. *Communication Diagram*
 2. *Interaction Overview Diagram*
 3. *Sequence Diagram*
 4. *Timing Diagram*
 - c. *State Machine Diagram*
 - d. *Use Case Diagram*

Berbagai diagram tersebut dapat menggambarkan hal-hal sebagai berikut:

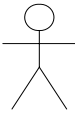

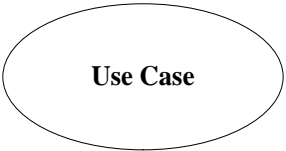
1. Berbagai aktifitas atau pekerjaan dalam sistem, baik yang dilakukan manusia maupun mesin.
2. Berbagai komponen data sistem bagaimana interaksinya.
3. Bagaimana sistem berjalan.
4. Bagaimana berbagai entitas berinteraksi dengan komponen lain.

5. Antar muka pengguna eksternal.

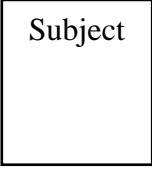

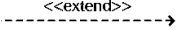
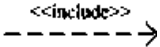

2.15.1 Use Case Diagram

Use-case diagram dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam *use-case diagram* adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya. *Use-case diagram* menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut.

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

Simbol	Nama	Deskripsi
 Actor/Role 	<i>Actor</i>	<ul style="list-style-type: none"> Seseorang atau sistem yang memperoleh manfaat dari sistem dan bersifat eksternal terhadap subjek. Digambarkan sebagai tongkat (<i>default</i>) atau jika yang terlibat bukan manusia, digambarkan dengan sebuah kotak dengan <code><<actor>></code> di dalamnya (alternatif). Dilabelkan dengan peran (<i>role</i>). Dapat dikaitkan dengan <i>actor</i> lain menggunakan <i>specialization</i> atau <i>superclass</i> (dilambangkan dengan panah dengan panah berongga). Ditempatkan di dalam batas sistem.
	<i>Use Case</i>	<ul style="list-style-type: none"> Merupakan bagian utama dari fungsionalitas sistem. Dapat memperpanjang <i>use-case</i> lain. Ditempatkan di dalam batas sistem. Diberi label dengan frasa kata kerja deskriptif-kata benda.

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

Simbol	Nama	Deskripsi
	<i>Subject Boundary</i>	<ul style="list-style-type: none"> • Terdapat nama subjek di dalam atau di atas. Merupakan ruang lingkup subjek. • Merupakan ruang lingkup subjek.
	<i>Association</i>	<ul style="list-style-type: none"> • Menghubungkan <i>actor</i> dengan <i>use-case</i> yang berinteraksi dengannya.
	<i>Extend</i>	<ul style="list-style-type: none"> • Merepresentasikan perluasan (<i>extend</i>) dari <i>use-case</i> lain untuk menyertakan perilaku opsional. • Disimbolkan dengan anak panah yang digambarkan dari perluasan <i>use-case</i> ke <i>use-case</i> pusat.
	<i>Include</i>	<ul style="list-style-type: none"> • Merupakan penyertaan fungsi dari satu <i>use-case</i> dengan <i>use-case</i> lainnya. • Disimbolkan panah yang ditarik dari pusat <i>use-case</i> ke <i>use-case</i> yang digunakan.
	<i>Generalization</i>	<ul style="list-style-type: none"> • Disimbolkan dengan anak panah yang digambarkan dari <i>use-case</i> khusus ke <i>use-case</i> umum.

Sumber: Sumber: (Dennis, Wixom, & Tegarden, 2015)

2.15.2 *Use Case Description*

Use Case Description memiliki urutan yang lengkap dari tahapan-tahapan untuk menyelesaikan suatu proses bisnis. Elemen yang dimiliki Use Case Description adalah sebagai berikut (Dennis, Wixom, & Tegarden, 2010):

1. *Basic Information*

a. Nama

Nama digunakan untuk menjelaskan *use case* dan membedakan *use case* satu dengan yang lainnya. Nama yang digunakan harus sederhana namun mampu menjelaskan kondisi yang memungkinkan.

b. *Description*

Description berarti uraian singkat untuk menyampaikan tujuan dari *use case*.

c. *Aktor*

Aktor merujuk kepada orang, sistem, perangkat lunak ataupun perangkat keras yang berinteraksi langsung dengan sistem untuk mencapai tujuan.

2. *Normal Courses*

Bagian utama dari *use case* berikutnya adalah normal courses. Pada *normal courses* menjelaskan mengenai langkah-langkah yang dilakukan untuk mengeksekusi respon terhadap masukan dan keluaran.

3. *Alternative Courses*

Untuk beberapa *use case* memiliki jalur alternatif pada setiap kasusnya. Pada penggambaran jalur alternatif ini terdapat cabang yang juga akan menghasilkan kesimpulan dari *use case* tersebut.



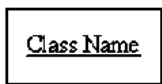
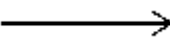




2.15.3 Diagram Aktivitas (*Activity Diagram*)

Secara grafis untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity Diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat operasi dieksekusi, dan memodelkan hasil dari *action* tersebut (Dennis, Wixom, & Tegarden, 2015). Pengertian *Activity diagram* adalah yang menggambarkan alur kerja bisnis independen dari *class*, aliran kegiatan dalam *use case*, atau desain rinci sebuah metode. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, diagram aktivitas juga banyak digunakan untuk mendefinisikan hal berikut:

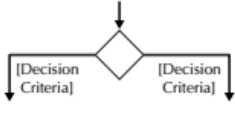
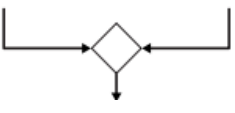


1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan hasil dari proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang akan ditampilkan pada perangkat lunak.

Simbol-simbol *Activity Diagram* dapat dilihat pada Tabel II.3

Tabel II.3 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	<i>Action</i>	<ul style="list-style-type: none"> • Merupakan notasi yang simple, dan tidak untuk dianalisis perilakunya.
	<i>Activity</i>	<ul style="list-style-type: none"> • Digunakan untuk melakukan satu <i>set</i> aksi.
	<i>Object Node</i>	<ul style="list-style-type: none"> • Digunakan untuk mewakili objek yang terhubung ke satu set arus objek.
	<i>Control Flow</i>	<ul style="list-style-type: none"> • Menunjukkan urutan eksekusi.
	<i>Object Flow</i>	<ul style="list-style-type: none"> • Menunjukkan aliran dari sebuah objek dari suatu aktivitas ke aktivitas lainnya.
	<i>Initial State</i>	<ul style="list-style-type: none"> • Menggambarkan permulaan dari suatu set aktivitas atau aksi.
	<i>Final-activity Node</i>	<ul style="list-style-type: none"> • Menggamarkan akhir dari setiap aktivitas.
	<i>Final-flow Node</i>	<ul style="list-style-type: none"> • Menggambarkan akhir aliran kontrol spesifik atau aliran objek.

Tabel II.3 Simbol-Simbol *Activity Diagram* (Lanjutan)

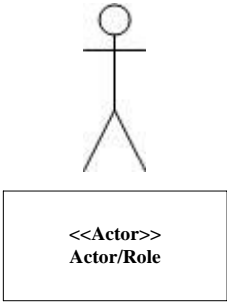
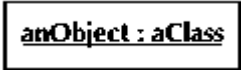

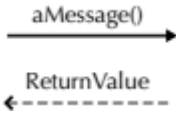
Simbol	Nama	Deksripsi
	<i>Decision Node</i>	<ul style="list-style-type: none"> Digunakan untuk mewakili kondisi pengujian untuk memastikan bahwa aliran kontrol atau aliran objek hanya turun satu jalur.
	<i>Merge Node</i>	<ul style="list-style-type: none"> Digunakan untuk membawa aliran keputusan yang berada ke satu <i>decision node</i>.
	<i>Join Node</i>	<ul style="list-style-type: none"> Menyatukan beberapa percabangan dari aliran aktivitas.
	<i>Swimlane</i>	<ul style="list-style-type: none"> Digunakan untuk memecah sebuah diagram aktivitas dalam baris dan kolom untuk menetapkan aktivitas individu (atau tindakan) kepada individu atau benda yang bertanggung jawab untuk melaksanakan kegiatan (atau tindakan).

Sumber: (Dennis, Wixom, & Tegarden, 2015)

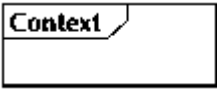
2.15.4 *Sequence Diagram*

Sequence Diagram merupakan urutan model dinamis yang menggambarkan contoh *class* yang berpartisipasi dalam *use case* dan pesan yang lewat di antara mereka dari waktu ke waktu (Dennis, Wixom, & Tegarden, 2015). *Sequence Diagram* merupakan diagram interaksi yang disusun berdasarkan urutan waktu. Cara membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan satu *flow* dari beberapa *flow* di dalam *use case*, dapat dilihat pada Tabel II.4

Tabel II.4 Simbol-Simbol *Sequence Diagram*

Simbol	Nama	Deskripsi
	<i>Actor</i>	<ul style="list-style-type: none"> • Orang atau sistem yang berasal dari luar sistem yang dapat memberikan manfaat. • Berpartisipasi secara berurutan oleh mengirim dan/atau menerima pesan. • Di tempatkan dibagian atas diagram. • Digambarkan sebagai gambar <i>stick</i>/gambar orang (<i>default</i>) atau jika bukan seorang aktor manusia, digambarkan dengan suatu kotak dengan tanda <i><<actor>></i> di dalamnya (alternatif).
	<i>Object</i>	<ul style="list-style-type: none"> • Berpartisipasi secara berurutan oleh mengirim dan/atau menerima pesan. • Ditempatkan dibagian atas diagram.
	<i>Lifeline</i>	<ul style="list-style-type: none"> • Menandakan kehidupan sebuah objek selama <i>Sequence</i>. • Berisi sebuah X pada titik dimana kelas tidak lagi berinteraksi
	<i>Message</i>	<ul style="list-style-type: none"> • Menyampaikan informasi dari satu objek ke objek lainnya. • Panggilan operasi diberi label dengan pesan yang dikirim dan panah padat, sedangkan pengembalian ditandai dengan nilai yang dikembalikan dan ditampilkan sebagai panah putus-putus

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


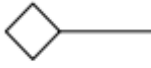
Simbol	Nama	Deskripsi
X	<i>Object Destruction</i>	<ul style="list-style-type: none"> Merupakan sebuah X ditempatkan pada akhir suatu garis hidup untuk menunjukkan bahwa itu akan keluar dari eksistensi.
	<i>Frame</i>	<ul style="list-style-type: none"> Menunjukkan konteks dalam <i>sequence diagram</i>

Sumber: (Dennis, Wixom, & Tegarden, 2015)

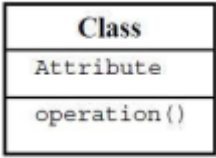
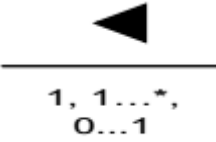

2.15.5 Class Diagram

Class Diagram merupakan salah satu diagram dalam konsep *Unified Modelling Language* (UML) yang menjelaskan mengenai berbagai jenis objek yang terdapat dalam sistem beserta beberapa hubungan antar objek tersebut. Sebuah *class* merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah *class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) (Dennis, Wixom, & Tegarden, 2015). Berikut simbol-simbol yang ada pada *class diagram*:

Tabel II.5 Simbol-Simbol *Class Diagram*

Simbol	Nama	Deskripsi
	<i>Generalization</i>	<ul style="list-style-type: none"> Merupakan sebuah <i>taxonomic relationship</i> antara <i>class</i> yang lebih umum dengan <i>class</i> yang lebih khusus.
	<i>Aggregation</i>	<ul style="list-style-type: none"> Mewakili hubungan <i>logic</i> antara <i>class</i> dan beberapa <i>class</i>.

Tabel II.5 Simbol-Simbol *Class Diagram* (Lanjutan)

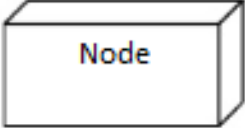
Simbol	Nama	Deskripsi
	<i>Class</i>	<ul style="list-style-type: none"> Kelas pada struktur sistem.
	<i>Association</i>	<ul style="list-style-type: none"> Asosiasi yang menghubungkan <i>class</i> dengan beberapa <i>class</i>. Bisa ada di antara satu atau lebih kelas.
	<i>Composition</i>	<ul style="list-style-type: none"> Mewakili hubungan fisik antara <i>class</i> dan beberapa <i>class</i>

Sumber: (Dennis, Wixom, & Tegarden, 2015)

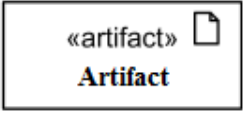
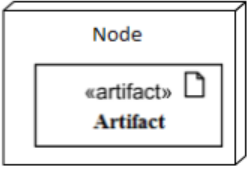
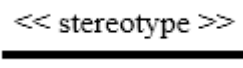
2.16 *Deployment Diagram*

Deployment diagram digunakan untuk mewakili hubungan antara komponen-komponen *hardware* yang digunakan dalam infrastruktur fisik sistem informasi. Misalnya, ketika membuat suatu sistem informasi terdistribusi yang akan menggunakan jaringan luas, *deployment diagram* dapat digunakan untuk menunjukkan hubungan komunikasi antara node yang berbeda dalam jaringan. *Deployment diagram* juga dapat digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi. Dalam hal ini, *deployment diagram* mewakili lingkungan pembuatan *software* (Dennis, Wixom, & Tegarden, 2015). Berikut simbol-simbol yang digunakan pada *deployment diagram*:

Tabel II.6 Simbol-Simbol *Deployment Diagram*

Simbol	Nama	Deskripsi
	<i>Node</i>	<ul style="list-style-type: none"> Untuk menggambarkan sumber daya komputasi dalam sebuah sistem (misalnya, <i>computer klien</i>, <i>server</i>, jaringan yang terpisah, atau individu perangkat jaringan).

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





Simbol	Nama	Deskripsi
	<i>Artifact</i>	<ul style="list-style-type: none"> Untuk menggambarkan spesifikasi dari <i>software</i> atau <i>database</i>, misalnya <i>file</i> sumber, tabel <i>database</i>, <i>executable file</i>.
	<i>Node with a Deployed Artifact</i>	<ul style="list-style-type: none"> Untuk menggambarkan <i>artifact</i> yang ditempatkan pada node fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.
	<i>Communication Path</i>	<ul style="list-style-type: none"> Untuk menggambarkan hubungan antara dua node untuk bertukar pesan.

Sumber: (Dennis, Wixom, & Tegarden, 2015)

2.17 ERD (*Entity Relationship Diagram*)

Terdapat sebuah model perancangan hubungan antar entitas (tabel) dari sebuah basis data (*database*). Istilah untuk frase ini biasa dikenal dengan nama *Entity Relationship Model*. Model hubungan ini seterusnya akan berlanjut menjadi sebuah Diagram Hubungan Antar Entitas yang biasa dikenal dengan nama *Entity Relationship Diagram* (ERD) (Hoffer, Presscott, & McFadden, 2007). Simbol-simbol ERD dapat dilihat pada Tabel II.7.

Tabel II.7 Simbol-Simbol ERD

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

Sumber: (Hoffer, Presscott, & McFadden, 2007)

2.18 Sistem Manajemen Basis Data

Sistem manajemen basis data merupakan sebuah sistem perangkat lunak yang digunakan untuk membuat, memelihara, dan menyediakan basis data yang dikendalikan oleh pengguna. SMBD menyediakan metode sistematis untuk membuat, memperbarui, menyimpan, dan mengambil data dalam database. Sistem manajemen basis data memungkinkan pengguna akhir dan pemrogram aplikasi untuk membagikan data (Hoffer, Presscott, & McFadden, 2007).

2.18.1 Basis Data (*Database*)

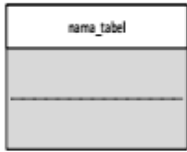
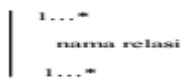
Basis data (*database*) merupakan kumpulan *file* / tabel / arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik (Fathansyah, 2015). Basis data dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.

2.18.2 *Conceptual Data Modelling* (CDM)

CDM (*Conceptual Data Model*) atau model konsep data adalah konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antar tabel untuk keperluan implementasi ke basis data (Sukanto & Shalahuddin, 2015). CDM merupakan hasil penjabaran lebih lanjut dari ERD. Berikut adalah simbol-simbol yang ada pada CDM:

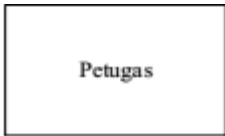


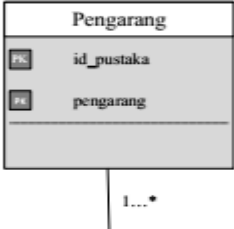
Tabel II.8 Simbol-Simbol *Conceptual Data Model* (CDM)

Simbol	Deskripsi
Entitas/tabel 	<ul style="list-style-type: none"> Entitas atau tabel yang menyimpan data dalam basis data.
Relasi 	<ul style="list-style-type: none"> Relasi antar tabel yang terdiri atas namarelasi dan <i>multiplicity</i>.

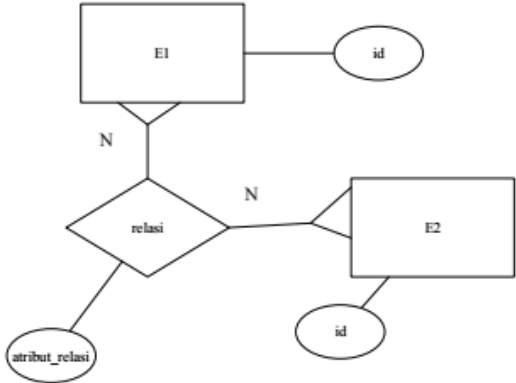
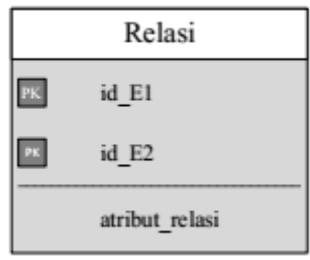
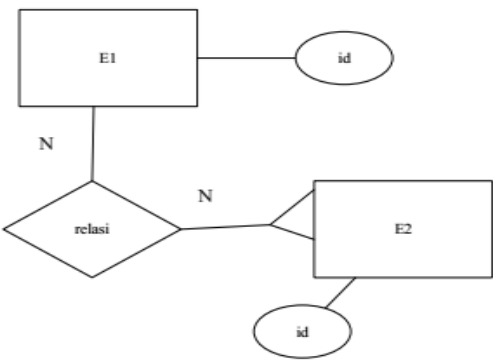
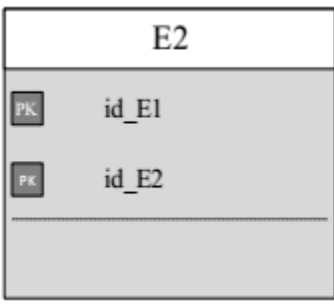
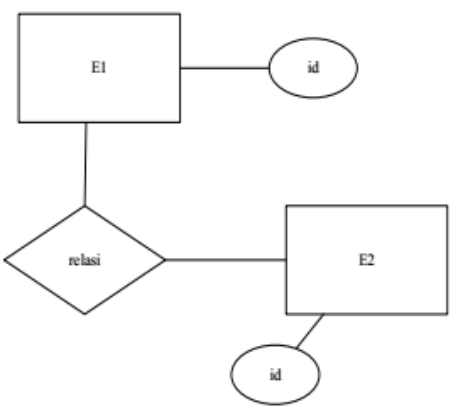
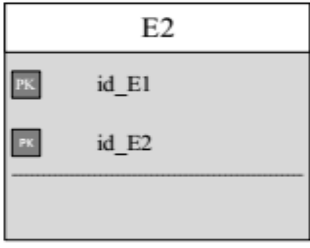
Sumber: (Rosa & Shalahuddin, 2015)

Aturan untuk mengubah ERD menjadi CDM secara umum adalah sebagai berikut:

Tabel II.9 Simbol-Simbol Aturan *Conceptual Data Model* (CDM)

ERD	CDM
 Entitas	 Menjadi sebuah tabel tersendiri.
 Atribut <i>multivalue</i>	 Menjadi sebuah tabel tersendiri dengan kunci primer (<i>primary key</i>) adalah kunci primer pada entitas dan memiliki atribut dengan nama seperti pada atribut entitas.

Tabel II.9 Simbol-Simbol Aturan *Conceptual Data Model* (CDM) (Lanjutan)

ERD	CDM
 <p>Relasi dengan kardinalitas <i>many to many</i></p>	 <p>Menjadi sebuah tabel tersendiri dengan kunci primer adalah atribut yang menjadi kunci primer di kedua entitas yang direlasikannya.</p>
 <p>Relasi dengan kardinalitas <i>one to many</i></p>	 <p>Kunci primer entitas yang memiliki hubungan <i>one</i> akan dijadikan kunci primer di entitas yang memiliki hubungan <i>many</i> dengan kata lain, relasi tidak menjadi tabel sendiri</p>
	 <p>Kunci primer salah satu entitas akan dijadikan kunci asing (<i>foreign key</i>) pada tabel yang lain dan kunci asing itu dijadikan kunci primer juga, dengan kata lain, relasi tidak menjadi tabel sendiri.</p>

Sumber: (Sukamto & Shalahuddin, 2015)

2.18.3 Kamus Data

Kamus Data merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi (Hartono, 2010).

Berdasarkan definisi di atas maka dapat disimpulkan bahwa kamus data merupakan suatu bantuan yang berguna untuk kebutuhan informasi dari suatu sistem informasi. Berikut adalah contoh penulisan kamus data:

Nama Tabel : Pemasok

Tipe : *File master*

Contoh dari kamus data dapat dilihat pada Tabel II.10 berikut:

Tabel II.10 Contoh Kamus Data

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

Sumber: (Hartono, 2010)

2.18.4 MariaDB

Didalam Maria.com MariaDB adalah sistem manajemen database relasional yang dikembangkan dari MySQL. MariaDB dikembangkan oleh komunitas pengembang yang sebelumnya berkontribusi untuk database MySQL.

MariaDB tetap mempertahankan kompatibilitas dan API layaknya MySQL dulu. Jika di MySQL ada InnoDB maka di MariaDB ada XtraDB yang menjadi mesin penyimpanan baru. Adapun Aria digunakan untuk transaksi database transaksional maupun non-transaksional.

Pengembangan MariaDB sekarang dipimpin oleh Michael "Monty" Widenius, salah satu founder MySQL AB dan Monty ProgramAB. Setelah MySQL diakusisi, Michael membangun sistem manajemen database baru dengan nama MariaDB. Penamaan MariaDB menggunakan salah satu anaknya Maria. Tak beda jauh dengan MySQL, MySQL juga dinamai dengan salah satu nama anaknya yaitu My.

Untuk awal mula penomoran versi, MariaDB mengikuti skema penomoran MySQL yakni 5.5. Setelah versi 5.5, pengembang MariaDB memutuskan untuk 'lompat jauh' dengan memberi versi terbaru mereka dengan penomoran 10. Tidak hanya penomoran versinya saja, fitur-fitur major pun dibangun dalam MariaDB. Saat ini versi terbaru MariaDB yang stabil adalah MariaDB 10.1.

Tak ketinggalan XAMPP pun mengganti MySQL dengan MariaDB semenjak bulan Oktober 2015. Memilih database untuk pengembangan perangkat lunak merupakan hal penting. Apalagi menyangkut lisensi aplikasi database yang digunakan. Jika pembaca sudah mulai sungkan dengan MySQL yang kini diakusisi oleh Oracle. Pembaca bisa mencoba alternatif database MySQL yakni MariaDB. Untuk API dan dukungan klien sudah sangat mendukung dan kompatibel, jadi tidak ada salahnya mencoba MariaDB yang bersifat open source.

2.18.5 Tipe Data

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* (Sutaji, 2012).

Beberapa jenis data yang tersedia pada MariaDB dapat dilihat pada Tabel II.12 berikut:

Tabel II.11 Tipe Data MySQL

Tipe Data	Keterangan
<i>CHAR</i>	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.
<i>VARCHAR</i>	<i>String</i> dengan panjang berupa variabel. M bisa mencapai 65535.
<i>DATE</i>	Data berupa tanggal. Format tanggal dalam bentuk 'YYYY-MM-DD'.
<i>TIME</i>	Data berupa waktu. Format waktu dalam bentuk 'HH:MM:SS'.
<i>TINYINT</i>	Bilangan antara -128 sampai dengan +127.
<i>SMALLINT</i>	Bilangan antara -32768 sampai dengan +32767.

Tabel II.11 Tipe Data MySQL (Lanjutan)

Tipe Data	Keterangan
<i>MEDIUMINT</i>	Bilangan antara -8388608 sampai dengan +8388607.
<i>INT</i>	Bilangan antara -2147683648 sampai dengan +2147683647
<i>FLOAT</i>	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.
<i>DOUBLE</i>	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.

Sumber: (Sutaji, 2012)

2.19 Windows Navigation Diagram

Windows Navigation Diagram merupakan statechart diagram khusus yang berfokus pada *user interface*. *Windows Navigation Diagram* (WND) adalah sebagai berikut (Dennis, Wixom, & Tegarden, 2015):

1. Menunjukkan bagaimana semua layar, formulir, dan laporan terkait.
2. Menunjukkan bagaimana pengguna bergerak dari satu ke yang lain.
3. Seperti diagram keadaan untuk *user interface*
 - a. Kotak mewakili komponen.
 - b. Panah mewakili transisi.
 - c. Stereotipe menunjukkan tipe antarmuka.

Contoh dari *Windows Navigation Diagram* dapat dilihat pada Gambar II.13



Gambar II.13 Contoh *Windows Navigation Diagram*

Sumber: (Dennis, Wixom, & Tegarden, 2015)

2.20 *Hypertext Preprocessor (PHP)*

PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source* (Anhar, 2010). PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru atau *up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script* tersebut dijalankan.

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1994. PHP (PHP: *Hypertext Preprocessor*) adalah bahasa *server-sidescripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *serverside scripting* maka sintaks dan perintah perintah PHP akan dieksekusi di *server* kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan dalam web lebih terjamin (Sutaji, 2012). Untuk menjalankan sistem PHP dibutuhkan 3 komponen, diantaranya adalah sebagai berikut (Sutaji, 2012):

1. *Webserver*
2. Program PHP
3. *Database Server*

Adapun kelebihan-kelebihan dari bahasa PHP diantaranya adalah sebagai berikut:

1. PHP mudah dibuat dan kecepatan akses tinggi.
2. PHP dapat berjalan dalam *webserver* yang berbeda dalam sistem operasi yang berbeda pula.
3. PHP diterbitkan secara gratis.
4. PHP merupakan bahasa yang dapat diletakkan dalam tag HTML.
5. Sistem *database* yang didukung PHP cukup banyak.
6. PHP termasuk *server side programming*.

2.21 Framework

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal. Seorang *programmer* tidak perlu membuat dari awal fungsi-fungsi seperti fungsi koneksi ke *database*, fungsi *string*, dan lainnya (Supono & Putratama, 2016).

Berdasarkan Wikipedia Indonesia, *PHP framework* adalah sebagai sebuah kerangka kerja yang disusun oleh berbagai komunitas pengembang web diseluruh dunia. Kerangka ini bertujuan untuk mempermudah dalam membuat sebuah aplikasi web bagi yang sering menulis *script* PHP secara keseluruhan dan itu pun akan diulang pada halaman yang lain. *Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi atau prosedur-prosedur dari *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal. Contoh *framework* PHP seperti *CakePHP*, *Codeigniter*, *Laravel*, *PRADO*, *Symfony*, *Zend Framework*, *Yii*, *Akelos*, *QPHP*, *ZooP*. Ada beberapa alasan mengapa menggunakan *framework* yaitu:

- a. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- b. Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat *programmer* mengikuti pola standar yang ada)
- c. Umumnya *framework* menyediakan fasilitas-fasilitas umum yang dipakai sehingga *programmer* tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dan lain-lain).

Framework merupakan banyak kode, yang disimpan dalam beberapa file yang terpisah, dan memudahkan dalam penggunaan kode yang digunakan secara berulang-ulang. Dengan *framework*, *programmer* tidak perlu menulis baris kode

yang panjang untuk fungsi tertentu. *Programmer* hanya perlu menggunakan fungsi yang sudah dituliskan ke dalam *framework*.

2.22 Codeigniter

Codeigniter adalah aplikasi *open source* berupa *framework* dengan *model* MVC (Model, View, Controller) untuk membangun website dinamis dengan PHP (Supomo & Putratama, 2016). Ada 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu:

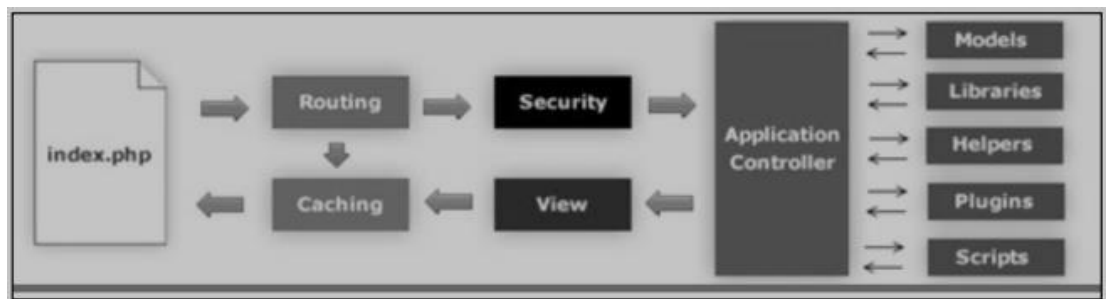
1. *View*, merupakan bagian yang menangani *presentation logic*. *View* berfungsi untuk menerima dan mempresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian *model*.
2. *Model*, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert, update, delete, search*) menangani validasi dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang diproses oleh aplikasi.

CI merupakan framework yang dibuat oleh Rick Ellis, CEO pMachine. CI dibuat dan terus dikembangkan, dan dapat digunakan secara gratis (Jubilee, 2015). CI juga mempunyai forum yang dapat digunakan untuk berbagi tips, mencari, dan mengatasi bug dalam kode. Tersedia pada situs www.codeigniter.com.

Codeigniter menggunakan pendekatan *Model-View-Controller*, yang bertujuan untuk memisahkan logika dan presentasi. Konsep ini mempunyai keunggulan dimana desainer dapat bekerja pada *template file*, sehingga kode presentasi dapat diperkecil (Subagia, 2018). Berikut ini konsep *Model-View-Controller* yang diterapkan pada *codeigniter*:

1. *Model*, menggambarkan struktur data. Biasanya kelas *model* akan berisi fungsi yang digunakan untuk mengambil, menambah dan memperbaharui informs yang ada di *database*.
2. *View*, informasi yang diperlihatkan kepada *user*. *View* adalah halaman web yang terdiri dari HTML, CSS dan Javascript, akan tetapi pada *codeigniter*, *view* dapat juga sebagai potongan halaman seperti *header* atau *footer*. Bahkan dapat juga halaman RSS tipe halaman lainnya.
3. *Controller*, prantara *Model*, *View*, dan *resource* lainnya yang dibutuhkan untuk menangani HTTP request dan menghasilkan halaman web.

Cara kerja codeigniter berdasarkan (Subagia, 2018) dapat dilihat pada Gambar II.14.



Gambar II.14 Cara Kerja *Codeigniter*

Sumber: (Subagia, 2018)

Berikut ini adalah penjelasan cara kerja codeigniter:

1. *Index.php* bertindak sebagai *controller* terdepan, dan menganalisis *resource* yang diperlukan untuk menjalankan *codeigniter*.
2. *Router*, memeriksa HTTP *request* untuk menentukan apa yang harus dikerjakan.
3. Jika *file cache file* ada, maka akan ditampilkan langsung, dengan melewati eksekusi normal sistem.
4. Sebelum memuat *controller*, HTTP *request* akan memeriksa apa yang *disubmit user* dan mem-*filter*nya untuk keamanan.
5. *Controller* membuat *model*, *core libraries*, *plugin*, *helper*, dan *resource* lainnya untuk memproses permintaan tertentu.

6. *View* ditampilkan di browser sesuai proses yang dikerjakan *controller*. Jika *caching* dijalankan, *view* akan *dicache* terlebih dahulu agar dapat ditampilkan di *request* selanjutnya.

Sebagai *web framework* populer yang menggunakan bahasa pemrograman PHP, *Codeigniter* mempunyai beberapa keunggulan seperti yang telah disebutkan di dokumentasinya, keunggulan-keunggulan tersebut meliputi:

1. *Free*, Karena berada di bawah lisensi *open source*, kita dapat melakukan apapun dengan *Codeigniter*. Lisensi lengkapnya dapat dilihat di halaman dokumentasi.
2. *Light weight*, sistem inti *Codeigniter* memerlukan *library* yang sedikit. Sangat berbeda dengan framework lainnya yang membutuhkan banyak sumber daya tambahan. *Library* tambahan akan digunakan ketika *request* secara dinamis, membuat *system* yang dibangun menjadi efisien dan cukup cepat.
3. *Fast*, menurut dokumentasi, performa yang dimiliki *Codeigniter* terbukti cepat setelah dibandingkan dengan *framework* lainnya.
4. Menggunakan kaidah MVC, dengan menggunakan *Model-View-Controller*, kita dapat memisahkan bagian *logic* dan *presentation* dari aplikasi yang kita bangun. Hal ini tentu sangat cocok dan bagus untuk proyek yang memfokuskan. *Designer* fokus pada *template file* dan *programmer* fokus pada pembangunan *logic* dari aplikasi yang dibangun.
5. Menghasilkan URL yang bersih, URL yang dihasilkan oleh *Codeigniter* bersih dan ramah terhadap search engine. *Codeigniter* menggunakan pendekatan *segment-based* dibandingkan dengan *query string* yang biasa digunakan oleh *programmer* yang tidak menggunakan *web framework*.
6. *Pack a Punch*, *Codeigniter* hadir dengan berbagai *library* yang akan membantu tugas-tugas di pengembangan *web* yang sudah umum dan sering dilakukan seperti mengakses *database*, mengirim *email*, validasi data dari *form*, mengelola *session*, memanipulasi gambar, dan masih banyak lagi.

7. *Extensible*, kita dapat menambahkan *library* atau *helper* yang kita ciptakan sendiri ke dalam *Codeigniter*. Selain itu kita dapat juga menambahkan fitur lewat *classs extension* atau *system hooks*.
8. *Thoroughly Documented*, hampir semua fitur, *library* dan *helper* yang ada di *Codeigniter* telah terdokumentasi dengan lengkap dan tersusun dengan baik. Ketika mendapatkan unduhan *Codeigniter*, dokumentasinya sudah tersedia dan siap digunakan.
9. Mempunyai komunitas yang ramah, bergabung di komunitas *Codeigniter* tentunya akan membantu sekali para pengguna *Codeigniter* yang masih pemula atau yang sudah mahir untuk saling berbagi ilmu pengetahuan. Komunitas ini dapat ditemui di codeigniter.com/forums.

2.23 *Black-Box Testing*

Pengujian *Black Box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan merekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black box* bukan merupakan alternatif dari teknik *white box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan dari pada metode *white box* (Pressman, 2011). Pengujian *black box* berusaha menemukan kesalahan dengan kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

Tidak seperti *white box*, yang dilakukan pada saat awal proses pengujian, pengujian *black box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black box* memperhatikan struktur kontrol, maka perhatian berfokus pada domain informasi (Pressman, 2011).

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metodologi penelitian adalah suatu cara atau prosedur yang dipergunakan untuk melakukan penelitian sehingga mampu menjawab rumusan masalah dan tujuan penelitian. Metodologi penelitian pada dasarnya merupakan cara ilmiah untuk mengumpulkan data dengan tujuan dan kegunaan tertentu. Cara ilmiah berarti kegiatan penelitian itu didasarkan pada ciri-ciri keilmuan rasional, empiris dan sistematis. Metodologi penelitian juga membuat penelitian lebih terarah.

3.2 Jenis Dan Sumber Data

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

1. Data Primer

Data primer merupakan data yang di peroleh secara langsung dari sumber asli atau pihak pertama. Perusahaan tempat dimana pengambilan data secara langsung terutama pada divisi *Quality Control*. Data yang didapat adalah alur proses pengendalian kualitas proses produksi yang sedang berjalan.

2. Data Sekunder

Data sekunder merupakan sumber data yang diperoleh secara tidak langsung atau melalui media perantara. Data sekunder pada umumnya seperti bukti catatan orang lain yang telah dilaporkan. Biasanya data tersebut adalah data umum perusahaan, profil perusahaan, dan struktur organisasi perusahaan.

3.3 Metode Pengumpulan Data

Metode pengumpulan data merupakan suatu faktor penting demi keberhasilan penelitian, untuk menganalisis suatu sistem yang berjalan untuk membuat sistem usulan. Berkaitan dengan berbagai jenis sumber data baik data

primer ataupun data sekunder. *Instrument* pengumpulan data merupakan alat yang biasa digunakan untuk melakukan pengumpulan data, adapun beberapa instrumen pengumpulan data yaitu:

1. Observasi

Observasi yaitu melakukan pengamatan situasi langsung pada sistem yang berjalan di Divisi *Quality Control* pada PT Nagasaki Component Parts. Hasil pengamatan yang dilakukan menjadi landasan dalam melakukan pengembangan sistem yang akan dibuat.

2. Wawancara

Metode wawancara adalah metode pengumpulan data melalui tanya jawab secara langsung kepada pihak-pihak yang terkait dengan objek penelitian. Pihak yang diwawancarai adalah karyawan di Divisi *Quality Control* dan Bagian *Repair*.

3. Studi Pustaka

Studi pustaka yaitu metode pengumpulan data dengan melakukan pencarian referensi yang berkaitan dengan pemrograman PHP dan *database* MySQL dari berbagai referensi, baik itu referensi elektronik yang didapat dari internet maupun referensi dari buku teks. Referensi yang diperoleh, kemudian dikaji sebagai dasar dalam menyelesaikan penelitian.

3.4 Metode Pengembangan Sistem

Pada penelitian ini untuk mengatasi masalah yang ada pada sistem, diputuskan untuk membuat pengembangan sistem. Dalam pengembangan sistem ini digunakan metode *waterfall*. Metode *Waterfall* sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Metode *Waterfall* menyediakan pendekatan alur hidup perangkat lunak dengan terurut dari fase awal (*planning*) hingga fase akhir (*implementation*). Berikut ini akan dijelaskan secara singkat mengenai tahapan-tahapan yang dilakukan dalam pengembangan sistem dengan menggunakan metode *waterfall*.

1. Perencanaan

Melakukan perencanaan tentang kebutuhan sistem yang diinginkan oleh staff *Quality Control* dengan melakukan wawancara secara intensif.

2. Analisis

Pengembang melakukan analisis kebutuhan sistem yang diperlukan oleh divisi *Quality Control* dengan cara melihat secara langsung sistem yang berjalan sehingga bisa diketahui apa permasalahannya.

3. Desain

Pengembang membuat desain program perangkat lunak seperti struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean serta dokumentasinya. Jadi, membuat bagaimana spesifikasi yang detail untuk bisa diimplementasikan.

4. Implementasi

Tahap implementasi, dimana pengembang mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Di sini pengembang mulai berurusan dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (pengkodean/*coding*).

5. Sistem

Pada tahapan sistem dilakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem/perangkat lunak yang kita buat sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya adalah bersifat *interactive*, yaitu kembali ke tahap sebelumnya. Tahap pemeliharaan dan perawatan dimana kita mulai melakukan pengoperasian sistem dan jika diperlukan melakukan perbaikan-perbaikan kecil. Kemudian jika waktu pengguna sistem habis, maka kita akan masuk lagi pada tahap perencanaan (*design*).

3.5 Literature Review

Literature review tidak hanya bermakna membaca literatur, tapi lebih ke arah evaluasi yang mendalam dan kritis tentang penelitian sebelumnya pada suatu

topik. *Literature review* memiliki peran penting dalam membuat suatu tulisan ataupun karangan ilmiah, karena dapat memberikan ide dan tujuan tentang topik penelitian yang akan dilakukan. Pada umumnya berisi ulasan, rangkuman & pemikiran penulis tentang beberapa pustaka (buku, jurnal, majalah) yang berkaitan dengan topik yang dibahas.

Literature review memiliki tujuan untuk mendapatkan landasan teori yang dapat mendukung pemecahan masalah yang sedang diteliti. Pada umumnya berisi ulasan, rangkuman, dan pemikiran penulis tentang beberapa pustaka (buku, jurnal, majalah) yang berkaitan dengan topik yang dibahas. Adapun langkah-langkah dalam *literature review* antara lain:

1. Formulasi Permasalahan
Penulis memilih topik yang sesuai, selain itu permasalahan yang diangkat harus sesuai, dalam hal ini sesuai dengan penelitian yang dilakukan.
2. Mencari Literatur
Literature yang dicari harus relevan dengan penelitian, sehingga dapat membantu untuk mendapatkan gambaran (*overview*) dari suatu topik penelitian.
3. Evaluasi Data
Melihat dari *literature* yang ada, apa saja yang menjadi persamaan dan perbedaan dengan topik yang dibahas.

Berikut ini merupakan *literature review* dari beberapa jurnal ilmiah dapat Dilihatpada Tabel III.1

Tabel III.1 *Literature Review*

No	Pengarang/Tahun	Topik Penelitian	Permasalahan	Metode yang digunakan	Kesimpulan Penelitian	Hasil Review
1.	Aldik Himawan. 2016	Sistem Pengendalian Kualitas Produksi	Banyaknya faktor yang menyebabkan cacat pada genteng seperti kualitas bahan baku, cuaca yang buruk, suhu udara yang panas, sirkulasi udara yang buruk, human eror dan keadaan peralatan yang sudah tidak baik.	Metode <i>Statistical Process Control</i> (SPC)	Penerapan metode Statistik <i>Process Control</i> (SPC) menghasilkan analisis dan penyelesaian masalah yang terjadi pada perusahaan tersebut.	Didalam jurnal penelitian pengendalian kualitas Pada genteng terdapat beberapa persamaan dalam pembahasan dengan Tugas Akhir yang dibuat oleh penulis. Salah satunya adalah metode yang digunakan pada jurnal milik Aldik Himawan adalah metode SPC (<i>Statistical Process Control</i>). Dimana tujuan dari jurnal ini sama dengan penulis yaitu meminimalis terjadinya produk gagal. Adapun perbedaan yang terdapat didalam jurnal Aldik Himawan dengan penulis Tugas Akhir. Jurnal berfokus kepada mengurangi jumlah cacat genetng sedangkan Tugas Akhir mengurangi jumlah cacat K 81. Tugas akhir yang dibuat oleh si penulis berupa suatu sistem informasi berbasis komputer.

Tabel III.1 *Literature Review* (lanjutan)

2.	Ida Bagus Suryaningrat, Noer Novijianto, dan Nur Faidah 2015	Sistem Pengendalian Kualitas Proses Produksi	minimnya sarana pengolahan, lemahnya pengawasan mutu, serta penerapan teknologi pengolahan biji kakao yang belum berorientasi pada mutu	Metode <i>Statistical Process Control</i> (SPC)	Penerapan metode <i>Statistical Process Control</i> (SPC) berhasil menganalisis faktor utama terjadinya kecacatan dalam proses pengolahan biji kakao.	Didalam jurnal penelitian analisis pengendalian kualitas Pengolahan Biji Kakao Pada PERUSAHAAN BIJI KAKAODI terdapat beberapa persamaan dalam pembahasan dengan Tugas Akhir yang dibuat oleh penulis. Salah satunya adalah metode yang digunakan pada jurnal milik Ida Bagus Suryaningrat, Noer Novijianto, dan Nur Faidah adalah metode SQC (<i>Statistic Process Control</i>). Dimana tujuan dari jurnal ini sama dengan penulis yaitu meminimalis terjadinya produk gagal. Adapun perbedaan yang terdapat didalam jurnal Ida Bagus Suryaningrat, Noer Novijianto, dan Nur Faidah dengan penulis Tugas Akhir. Tugas akhir yang dibuat oleh si penulis berupa suatu sistem informasi berbasis <i>computer</i> . juga ada beberapa <i>tools</i> yang tidak digunakan oleh penulis terhadap jurnal tersebut.
----	--	--	---	---	---	---

3.6 Kerangka Penelitian

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

1. Perancangan dan pengembangan sistem

- Studi Pendahuluan

Proses pemagangan yang dilakukan dengan mengetahui terlebih dahulu alur proses bisnis yang berjalan pada PT Nagasakti Component Parts. Studi pendahuluan ini dimaksud untuk mengetahui gambaran alur bisnis yang terjadi pada Divisi *Quality Control* dan produksi pada saat ini, dan juga mengetahui masalah yang terjadi pada divisi tersebut. Langkah langkah yang dilakukan dalam studi pendahuluan untuk mendapatkan informasi adalah melakukan observasi dan wawancara terhadap pihak pihak terkait yang berada di Divisi *Quality Control* dan produksi. Selain itu juga studi pendahuluan dilakukan dengan cara membaca buku, literatur, serta sumber lain guna untuk mendapatkan beberapa referensi untuk mengerjakan Tugas Akhir.

- Identifikasi Masalah

Masalah yang terjadi di Divisi *Quality Control* kurang rapihnya penyimpanan data mengakibatkan banyak *file* yang hilang di karenakan proses penginputan data masih manual dan sistem yang digunakan masih menggunakan *Microsoft excel*, pernah terjadi kasus data *Quality Control* disimpan didalam *flashdisk* dan *flashdisk* tersebut hilang, mengakibatkan sulitnya membuat laporan *Quality*. Adapun cara mengidentifikasi masalah dalam penelitian ini dengan melakukan

a. Wawancara dan Observasi

Wawancara Salah satu metode pengumpulan data dengan cara melakukan tanya jawab secara langsung kepada narasumber yang bersangkutan mengenai berbagai macam hal yang dibutuhkan oleh si penulis untuk melakukan pengamatan dan pengembangan sistem.

Observasi yaitu melakukan pengamatan langsung dari Divisi *Quality Control* dan Produksi pada PT Nagasaki Component Parts. Hasil dari pengamatan tersebut akan menjadi landasan utama dalam melakukan pengembangan sistem.

b. Analisis hasil Observasi dan Wawancara

Analisis dari hasil observasi dan wawancara yang telah dilakukan oleh penulis dengan menggunakan tabel observasi yang dilampirkan, serta transkrip wawancara yang juga akan dilampirkan sebagai analisis dari hasil wawancara yang dilakukan oleh penulis.

c. Analisis *PIECES*

Analisis *PIECES* ini merupakan identifikasi permasalahan dengan menggunakan metode analisis sebagai untuk memperoleh pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem biasanya akan dilakukan beberapa aspek diantaranya adalah kinerja, informasi, ekonomi, keamanan aplikasi, efisiensi dan pelayanan pelanggan. Analisis ini akan mengidentifikasi permasalahan berdasarkan *PIECES* yaitu aspek kinerja, aspek informasi, aspek ekonomi, aspek kontrol, aspek efisiensi, dan aspek pelayanan.

- Identifikasi Solusi

Dalam mengidentifikasi solusi pada penelitian ini ada beberapa hal yang harus dilakukan yaitu:

a. Menentukan tujuan dan batasan

Dalam proses ini penulis harus menentukan tujuan yang akan dianalisis dan dibuat pengembangan sistemnya, dan juga penulis harus menentukan batasan – batasan dalam penelitian yang akan dilakukan, dalam hal ini penulis melakukan penelitian di PT Nagasaki Component Parts di bagian Divisi *Quality Control* sebagai batasan melakukan penelitian dan bertujuan untuk mengurangi tingkat barang K 81.

b. Menggunakan Metode *Waterfall*

Metode *Waterfall* adalah suatu pengembangan sistem informasi berurutan, dimana kemajuan dipandang terus mengalir ke bawah (seperti air terjun) melewati *fase-fase* perencanaan, pemodelan, implementasi (konstruksi) dan pengujian.

Kelebihan metode ini adalah:

- Memiliki proses yang urut, mulai dari analisa hingga *support*.
- Setiap proses memiliki spesifikasinya sendiri, sehingga sebuah sistem dapat dikembangkan sesuai dengan apa yang dikehendaki (tepat sasaran).
- Setiap proses tidak dapat saling tumpang tindih.

c. Menggunakan Metode SPC

Metode SPC adalah salah satu metode pengendalian kualitas dilakukan menggunakan teknik penyelesaian masalah yang digunakan secara statistik kelebihan metode ini adalah untuk memonitor, mengendalikan, menganalisis, mengelola dan memperbaiki standar dari kualitas hasil produksi pada tingkat biaya minimum yang didesain untuk mengevaluasi kualitas ditinjau dari kesesuaian dengan spesifikasinya.

2. Analisis Sistem

a. Analisis Sistem Berjalan

- Analisis dokumen masuk (*input*) dan dokumen keluaran (*output*).
- Analisis proses bisnis sistem yang berjalan dalam bentuk *Flowmap*

b. Analisis Sistem Usulan

Identifikasi kebutuhan sistem, baik *functional* maupun *non-functional*.

3. Desain (Perancangan Sistem)

Merancang sistem sesuai dengan analisis yang dilakukan, meliputi:

a. Perancangan alur sistem yang diusulkan menggunakan *flowmap*.

b. Desain sistem menggunakan:

- Diagram *use case* dibuat untuk menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Suatu landasan pola pikir yang ditekankan dalam diagram *use case* ini adalah “apa” yang dapat diperbuat oleh sistem, dan bukan “bagaimana” sistem melakukannya.

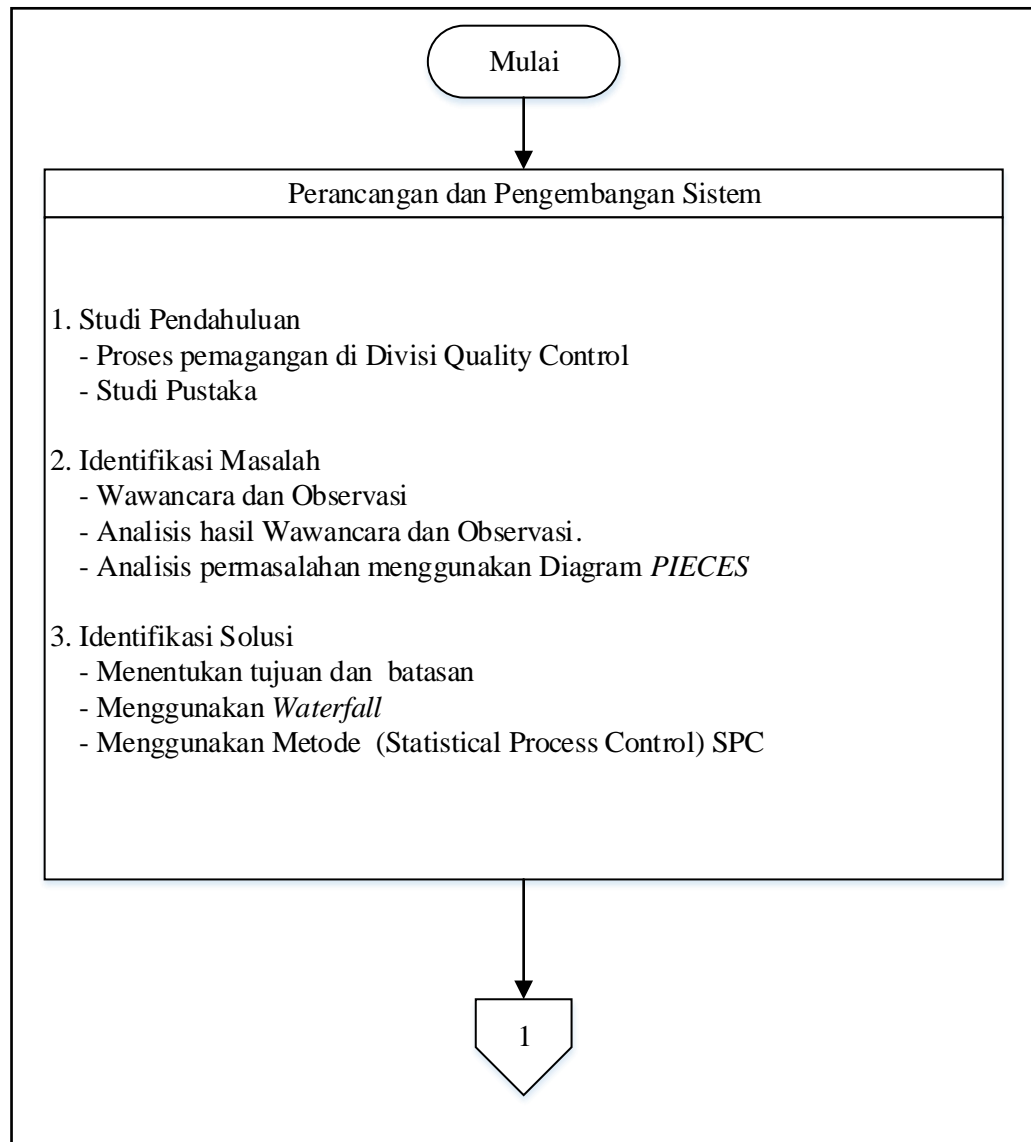
Diagram ini sangat tepat untuk menggambarkan hubungan yang terjadi antara sistem dengan *user*nya dalam hal ini *user* adalah *staff Quality Control* dan Produksi.

- *Activity Diagram* adalah Digunakan untuk model perilaku dalam independen proses bisnis. Dalam banyak hal, diagram aktivitas dapat dipandang sebagai diagram aliran data yang canggih yang digunakan dalam hubungannya dengan analisis terstruktur. Diagram aktivitas digunakan untuk menggambarkan aliran data yang terjadi pada proses pengendalian kualitas proses produksi di PT Nagasaki Component Parts.
 - *Class Diagram* *Class Diagram* merupakan salah satu diagram dalam konsep *Unified Modelling Language* (UML) yang menjelaskan mengenai berbagai jenis objek yang terdapat dalam sistem beserta beberapa hubungan antar objek tersebut.
 - *Sequence Diagram* berfungsi untuk menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi didefinisikan. Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara set, sehingga sangat membantu untuk memahami spesifikasi *real-time* dan kompleks.
- c. Desain data menggunakan *Entity Relationship Diagram* digunakan untuk pemodelan basis data relasional dan kamus data yang digunakan untuk membantu dalam menggambarkan atau pengidentifikasian setiap *file/field* didalam sistem yang dibangun.
- d. Desain *interface* menggunakan *Windows Navigation Diagram* yang berfungsi untuk menggambarkan perpindahan atau transisi *window* (jendela) ke *window* yang lain, berdasarkan *interface* dasarnya dan tombol apa saja atau *event* apa saja yang menyebabkan perpindahan dari satu *window* ke *window* yang lain dan perancangan antarmukanya.

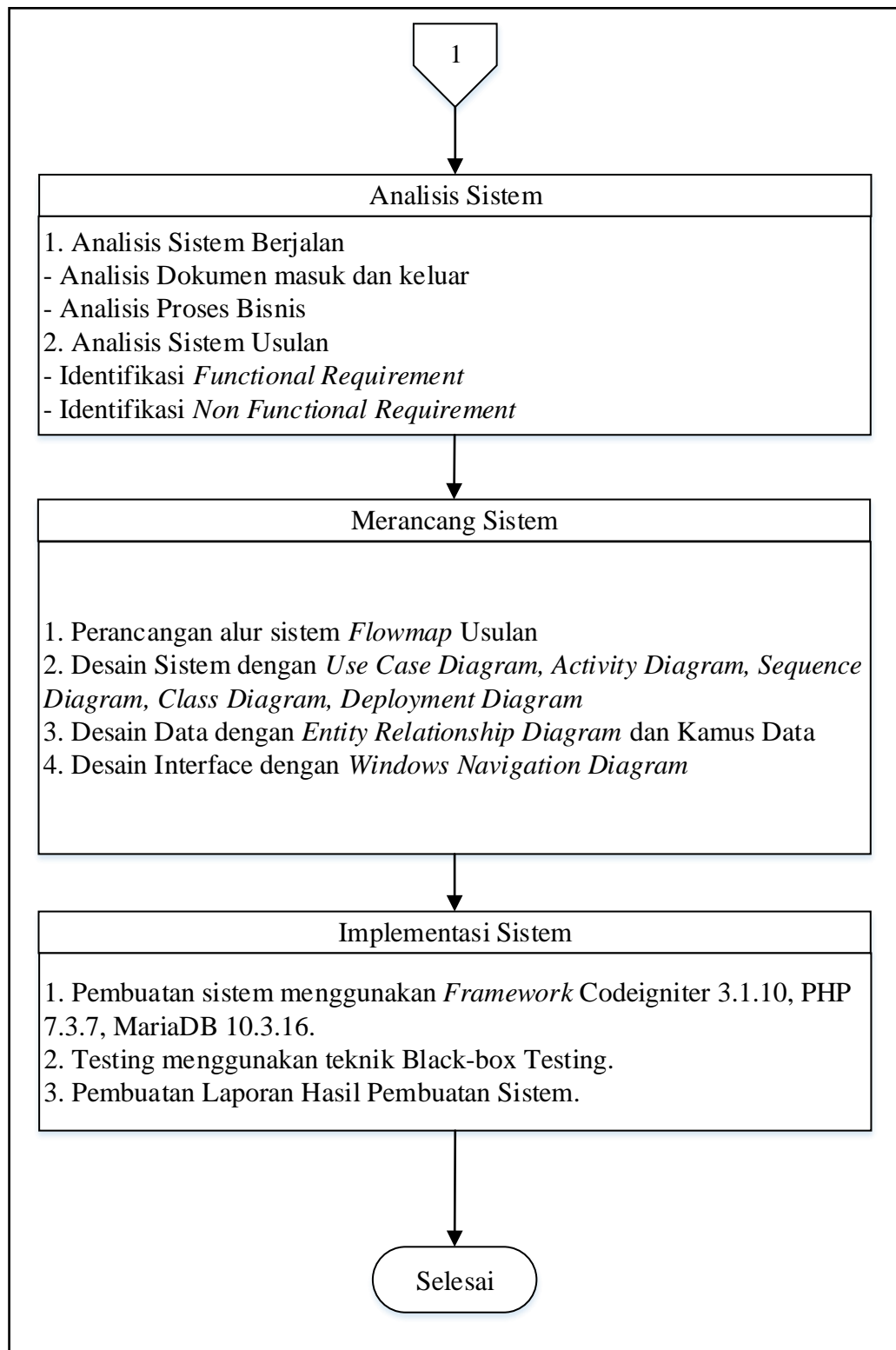
4. Implementasi Sistem

- a. Pembuatan sistem menggunakan *Framework CodeIgniter* 3.1.10, PHP 7.3.7, dan MariaDB 10.3.16.
- b. Dalam proses implementasi sistem, penulis melakukan proses pemrograman untuk merancang sistem usulan yang dibutuhkan dengan menggunakan *Framework CodeIgniter* 3.1.10, PHP 7.3.7, dan MariaDB 10.3.16.
- c. *Testing*
Melakukan *Testing* atau pengujian terhadap sistem yang telah dibuat. Pada penelitian ini menggunakan teknik *Black-Box Testing*.
- d. Pembuatan Laporan
Membuat laporan hasil pembuatan sistem informasi pengendalian kualitas proses produksi.

Kerangka penelitian dari laporan Tugas Akhir ini dapat dilihat pada Gambar III.1.



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



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

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sejarah Umum Perusahaan

PT Nagasaki Component Parts didirikan sejak tahun 1980, telah berpengalaman di bidang manufaktur dan pemasaran industri karet utama untuk produksi dan pengolahan komponen karet. Berbagai komponen diproduksi untuk pengembangan proyek. Produk yang dihasilkan yaitu komponen karet otomotif, bagian karet industri.

PT Nagasaki Component Parts memproduksi komponen yang terbuat dari karet alam dan sintesis. Perusahaan dalam menjalankan usahanya dengan keterampilan tenaga kerja, penelitian, dan pengembangan, sistem jaminan kualitas dan sistem distribusi yang terdefinisi dengan baik. Saat ini, produk diserap semua untuk mobil, sepeda motor, dan bagian manufaktur. Kesuksesan yang ditingkatkan oleh perusahaan saat ini didasarkan pada pendekatan inovatif, penciptaan sistem inovasi baru dan efisiensi produksi. Karena produk yang perusahaan hasilkan memiliki peran yang berharga di bidang otomotif dan industri. Perusahaan percaya bahwa sekarang ini waktu terbaik untuk PT Nagasaki Component Parts masuk ke produsen global produk karet. Berikut adalah lambang PT Nagasaki Component Parts Gambar IV.1.



Gambar IV.1. Lambang Perusahaan PT Nagasaki Component Parts
Sumber: PT Nagasaki Component Parts (2018)

4.2 Profil Perusahaan

Nama Perusahaan	: PT Nagasakti Component Parts
Lokasi	: Jl. Bintang Mas No. 12 KM.47,5, Nanggewer Mekar, Cibinong Bogor, 16912
Telepon	: 021-3801790, 8753729
Fax	: 021-3811347
Direktur Utama	: Bpk. Monsantoso Wijaya Lie
Email	: nagasakti24@yahoo.com
Status Perusahaan	: Industri Otomotif
Produksi	: Rubber dan Komponen Otomotif

4.3 Visi dan Misi Perusahaan

Visi merupakan pandangan jauh tentang suatu perusahaan ataupun lembaga dan lain-lain yang ingin dicapai pada masa yang akan datang. Visi juga dapat diartikan sebagai tujuan perusahaan atau lembaga dan apa yang harus dilakukan untuk mencapai tujuannya tersebut pada masa depan.

Misi merupakan suatu pernyataan tentang apa yang harus dikerjakan oleh perusahaan atau lembaga dalam usaha mewujudkan visi tersebut. Misi perusahaan diartikan sebagai tujuan dan alasan mengapa perusahaan atau lembaga itu dibuat. Misi juga akan memberikan arah sekaligus batasan-batasan proses pencapaian tujuan.

PT Nagasakti Component Parts merupakan perusahaan industri yang membuat komponen dan spare part pada bagian mobil dan motor. PT Nagasakti Component Parts mempunyai moto *on time delivery*. PT Nagasakti Component Parts mempunyai Visi dan Misi Sebagai berikut:

Visi:

1. Menjadi competitor untuk pembuatan suku cadang berbasis material karet kompon.
2. Menjadi mitra bagi perusahaan Nasional maupun Regional terutama yang memerlukan suku cadang karet.
3. Menjadi salah satu perusahaan yang menciptakan efisiensi bagi rekanan.

Misi:

1. Dapat dipercaya sebagai penyedia keperluan peralatan suku cadang berbasis material karet bagi partner usaha.
2. Sebagai harapan bagi tumpuan karya dan kehidupan bagi karyawan.

4.4 Struktur Organisasi Perusahaan

Struktur Organisasi adalah suatu susunan dan hubungan antar bagian-bagian, komponen dan posisi dalam suatu perusahaan. Dengan adanya struktur organisasi maka akan memberikan informasi kepada seluruh manusia yang menjadi anggotanya untuk mengetahui kegiatan atau pekerjaan yang harus ia kerjakan, berkonsultasi atau bertanggung jawab kepada siapa, sehingga proses kerjasama menuju pencapaian tujuan organisasi dapat terwujud sesuai dengan perencanaan yang telah ditetapkan sebelumnya.

Struktur organisasi dapat diartikan sebagai kerangka kerja formal organisasi yang dengan kerangka kerja itu tugas-tugas pekerjaan dibagi-bagi, dikelompokkan, dan dikoordinasikan (Robbins & Coulter, 2007).

Struktur Organisasi pada PT Nagasakti Component Parts memakai struktur organisasi lini/garis. Dengan alasan dalam struktur organisasi PT Nagasakti Component Parts adanya perintah terpusat pada atasan langsung dan koordinasi dengan bawahannya langsung sehingga perintah jelas dan mudah di laksanakan berdasarkan tanggung jawabnya masing-masing. Ciri-ciri struktur organisasi lini/garis adalah sebagai berikut:

1. Mempunyai wewenang dan tanggung jawab secara langsung secara vertical yang dikaitkan dengan jabatan dan tugas tiap tingkatan.
2. Adanya kesatuan komando atau perintah.
3. Hanya cocok untuk organisasi yang kecil atau sederhana.
4. Bawahan hanya mempunyai satu atasan.

Keuntungan dari struktur organisasi lini / garis adalah:

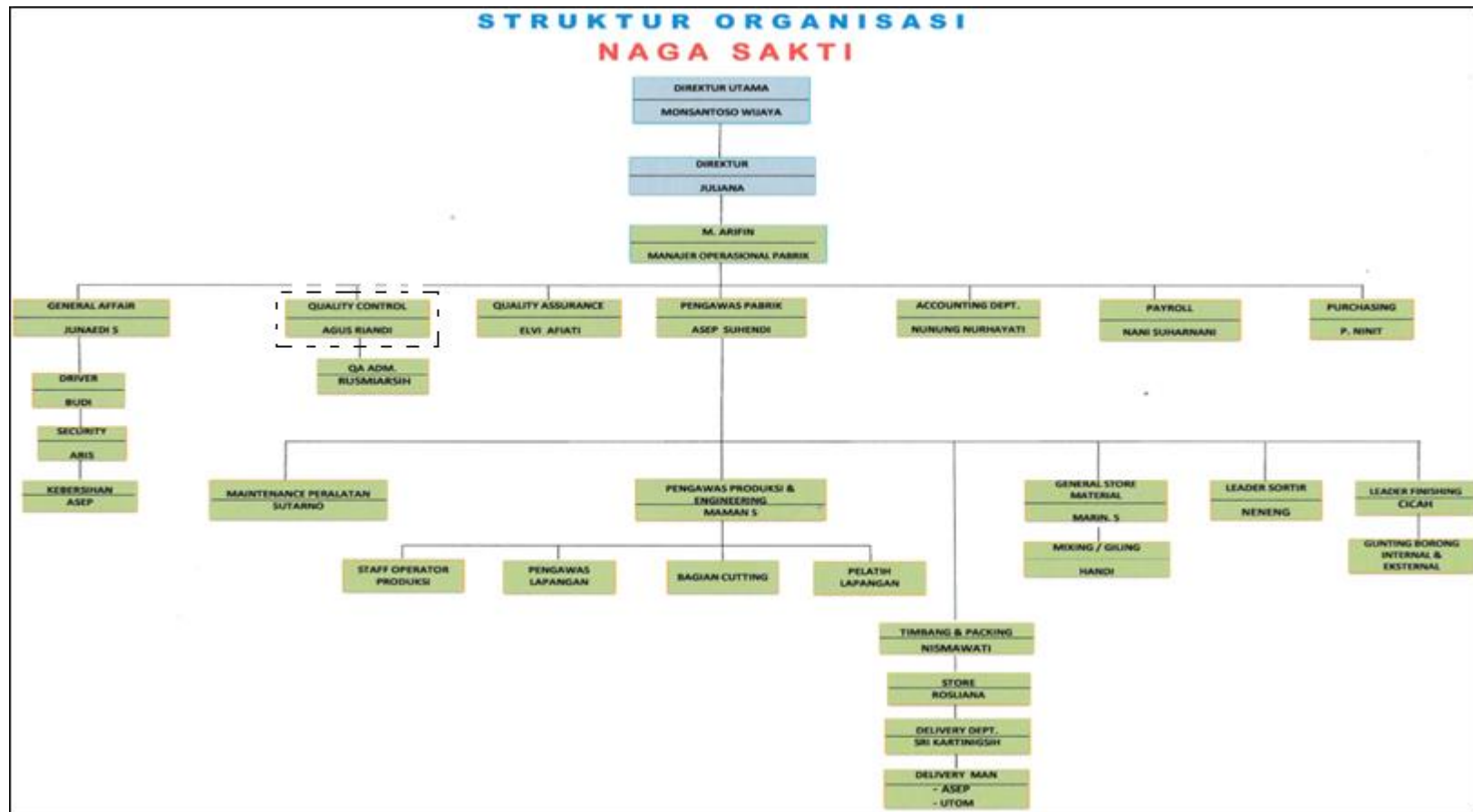
1. Sederhana dan jelas.
2. Wewenang dan tanggung jawabnya mengalir dengan jelas.

3. Mata rantai perintah menghubungkan semua pekerjaan beserta pejabat satu dengan yang lainnya secara menyeluruh.
4. Ada kesatuan pengarahannya sehingga tidak membingungkan bawahannya.

Kerugian dari struktur organisasi lini / garis adalah:

1. Struktur ini hanya cocok untuk organisasi yang masih sederhana.
2. Pucuk pimpinan akan mengalami kesulitan apabila organisasi ini berkembang.

Berikut ini adalah struktur organisasi PT Nagasakti Component Parts (Gambar IV.2:



Gambar IV.1 Struktur Organisasi PT Nagasakti Component Parts
Sumber: PT Nagasakti Component Parts (2018)

4.5 *Job Description*

Job description adalah sebuah kumpulan informasi jabatan dan disusun secara sistematis yang dapat mengidentifikasi dan menguraikan suatu jabatan atau posisi tertentu. *Job description* membuat status setiap jabatan menjadi jelas akan fungsi dan perannya, hasilnya, serta tanggung jawabnya. Dari struktur organisasi diatas, maka *job description* dari masing-masing pekerjaan adalah:

1. *Direktur*

Pemimpin perusahaan, memiliki tanggung jawab CV atas semua kepentingan perusahaan.

2. *Manajer Operasional Pabrik*

Bertanggungjawab untuk memastikan organisasi berjalan sebaik mungkin dalam memberikan pelayanan dan memenuhi harapan para pelanggan dan *client* dengan cara yang efektif dan efisien.

3. *General Affair*

Bertanggungjawab atas pengadaan barang dan jasa yang mendukung seluruh aktivitas operasional kantor dan melakukan pemeliharaan asset fisik kantor serta bekerjasama dengan bagian bisnis, operasional, dan keuangan untuk melakukan pengelolaan anggaran atas biaya pengadaan barang atau jasa, pemeliharaan serta biaya-biaya lain yang terkait.

4. *Quality Control*

- a. Membantu perkembangan semua produk yang diproduksi oleh perusahaan.
- b. Bertanggungjawab untuk memantau, menganalisis, meneliti, dan menguji suatu produk.
- c. Memverifikasi kualitas produk.
- d. Bertanggungjawab memonitor setiap proses yang terlibat dalam produksi produk.
- e. Memastikan kualitas barang produksi sesuai standar.
- f. Merekomendasikan pengolahan ulang produk-produk berkualitas rendah.

- g. Bertanggungjawab untuk dokumentasi inspeksi dan tes yang dilakukan pada produk dari sebuah perusahaan.
- h. Membuat analisis catatan sejarah perangkat dan dokumentasi produk sebelumnya untuk referensi di masa mendatang.

5. *Quality Assurance*

- a. Bertanggungjawab memonitoring, uji-tes, dan memeriksa semua proses produksi yang terlibat dalam produksi suatu produk dan memastikan semua standar kualitas dipenuhi oleh setiap komponen dari produk atau layanan yang disediakan oleh perusahaan untuk memberikan jaminan kualitas sesuai standar yang diberikan oleh perusahaan.
- b. Menerima keluhan *customer* apabila produk dibawah standar dan menganalisa akar masalahnya untuk menjadi dan memastikan tidak terjadi lagi masalah yang sama pada produk tersebut.

6. *Accounting Department*

- a. Mengatur sirkulasi keuangan dalam perusahaan.
- b. Memantau *profit* dan *loss* dari perusahaan agar dapat mengetahui apabila terjadi kerugian dan *loss profit*.

7. *Payroll*

- a. Bertanggungjawab menangani segala hal yang berkaitan dengan administrasi penggajian.
- b. Bertanggungjawab membuat *report* laporan nilai gaji ke badan yang berwenang.

8. *Purchasing*

Bertanggungjawab atas pemesanan barang atau keperluan perusahaan.

9. *Maintenance Peralatan*

Bertanggungjawab untuk mengawasi pelaksanaan pekerjaan-pekerjaan pemeliharaan terhadap seluruh peralatan proses dan penggunaan alat sampai dengan penggunaannya.

10. *Pengawas Produksi dan Engineering*

Bertanggungjawab memastikan produksi berjalan dengan baik dan mengatur persiapan peralatan.

11. *General Store Material*

Bertanggungjawab penuh atas segala hal mengenai material atau *compound* baik dari segi perawatan dan kondisi.

12. *Leader Sortir*

Bertanggungjawab penuh atas hasil barang sortir dan memenuhi jenis-jenis barang apa saja yang harus diutamakan untuk di sortir terlebih dahulu.

13. *Leader Finishing*

Bertanggungjawab penuh atas hasil barang *finishing*.

14. *Section Head Compounding*, mempunyai tugas dan tanggung jawab sebagai berikut:

- a. Merealisasikan/menterjemahkan *planning* produksi ke dalam aktivitas produksi.
- b. Mengkoordinasikan penyelesaian masalah produktifitas dan kualitas produksi.
- c. Mengkoordinir penekanan *rejection rate*.
- d. Mengendalikan dokumen, data dan catatan yang terkait.
- e. Membuat prosuder, intruksi kerja dan standar di seksinya.
- f. Melakukan perbaikan dan perkembangan proses produksi.
- g. Berperan aktif sebagai *Circle leader* dalam aktivitas QCC/SS.
- h. Melakukan identifikasi aspek lingkungan, potensi kecekaan kerja dan keadaan darurat dan melaporkan kepada atasannya,
- i. Memonitor dan melaksanakan 5R diarea kerjanya.

15. *Section Head Metal Treatment*, mempunyai tugas dan tanggung jawab sebagai berikut:

- a. Mengatur dan memonitor proses *Metal Treatment*.
- b. Mengevaluasi dan menganalisa hasil proses Metal Treatment.
- c. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
- d. Mengawasi kesesuaian proses pembuangan limbah sesuai jenisnya dan melakukan retur metal (NG) ke gudang.

- e. Memberikan pelatihan prosedur, petunjuk kerja baru atau oprator baru.
 - f. Membuat prosedur, intruksi kerja dan standar di seksinya.
 - g. Melakukan perbaikan dan pengembangan proses produksi.
 - h. Melakukan identifikasi aspek lingkungan, potensi kecelakaan kerja dan kedaan darurat dan melaporkan kepada atasaanya.
 - i. Memonitor dan melaksanakan 5R diarea kerjanya.
16. *Section Head Molding*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Merealisasikan/menterjemahkan *planning* produksi ke dalam aktivitas produksi.
 - b. Mengkoordinir penyelesaian masalah produktifitas dan kualitas produksi.
 - c. Mengkoordinir penekanan *rejection rate*.
 - d. Mengendalikan dokumen, data dan catatan yang terkait.
 - e. Membuat prosedur, intruksi kerja dan standar di seksinya.
 - f. Melakukan perbaikan dan pengembangan proses produksi.
 - g. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - h. Melakukan identifikasi aspek lingkungan, potensi kecelakaan kerja dan kedaan darurat dan melaporkan kepada atasaanya.
 - i. Memonitor dan melaksanakan 5R diarea kerjanya.
17. *Section Head Finishing*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Merealisasikan/menterjemahkan *planning* produksi ke dalam aktivitas produksi.
 - b. Mengkoordinir penyelesaian masalah produktifitas dan kualitas produksi.
 - c. Mengkoordinir penekanan *rejection rate*.
 - d. Mengendalikan dokumen, data dan catatan yang terkait.
 - e. Membuat prosedur, intruksi kerja dan standar di seksinya.
 - f. Melakukan perbaikan dan pengembangan proses produksi.
 - g. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - h. Melakukan identifikasi aspek lingkungan, potensi kecelakaan kerja dan kedaan darurat dan melaporkan kepada atasaanya.
 - i. Memonitor dan melaksanakan 5R diarea kerjanya.

18. *Section Head Inspection I*, mempunyai tugas dan tanggung jawab sebagai berikut:
 - a. Merealisasikan/menterjemahkan *planning* produksi ke dalam aktivitas produksi.
 - b. Mengkoordinir penyelesaian masalah produktifitas dan kualitas produksi.
 - c. Mengkoordinir penekanan *rejection rate*.
 - d. Mengendalikan dokumen, data dan catatan yang terkait.
 - e. Membuat prosedur, intruksi kerja dan standar di seksinya.
 - f. Melakukan perbaikan dan pengembangan proses produksi.
 - g. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - h. Melakukan identifikasi aspek lingkungan, potensi kecelakaan kerja dan keadaan darurat dan melaporkan kepada atasanya.
 - i. Memonitor dan melaksanakan 5R diarea kerjanya.
19. *Section Head Inspection II*, mempunyai tugas dan tanggung jawab sebagai berikut:
 - a. Merealisasikan/menterjemahkan *planning* produksi ke dalam aktivitas produksi.
 - b. Mengkoordinir penyelesaian masalah produktifitas dan kualitas produksi.
 - c. Mengkoordinir penekanan *rejection rate*.
 - d. Mengendalikan dokumen, data dan catatan yang terkait.
 - e. Membuat prosedur, intruksi kerja dan standar di seksinya.
 - f. Melakukan perbaikan dan pengembangan proses produksi.
 - g. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - h. Melakukan identifikasi aspek lingkungan, potensi kecelakaan kerja dan keadaan darurat dan melaporkan kepada atasanya.
 - i. Memonitor dan melaksanakan 5R diarea kerjanya.
20. *Foreman/Pengawas Molding*, mempunyai tugas dan tanggung jawab sebagai berikut:
 - a. Mengatur dan memonitor proses molding.
 - b. Mengevaluasi dan menganalisa hasil proses Molding.

- c. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - d. Mengawasi pelatihan prosedur, petunjuk kerja baru atau operator baru.
 - e. Memonitor dan melaksanakan 5R di area kerjanya.
21. *Operator Molding*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Melakukan proses Molding pada mesin Molding (*General Press, Vacuum, Injection*).
 - b. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - c. Memonitor dan melaksanakan 5R di area kerjanya.
22. *Operator Mixing*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Melakukan proses giling *compound* MB/SM pada M/C *Knreader* dan *Open Mill*.
 - b. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - c. Memonitor dan melaksanakan 5R di area kerjanya.
23. *Setting Mold*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Melakukan pekerjaan *Setting Mold, Supply gomu, control temperature* dan produk.
 - b. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - c. Memonitor dan melaksanakan 5R di area kerjanya.
24. *Mold Washing (Cleaning Mold)*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Melakukan pekerjaan *Washing Mold, Setting Mold* dan persiapan *Pre Heat Mold*.
 - b. Berperan aktif sebagai *Circle Leader* dalam aktivitas QCC/SS.
 - c. Memonitor dan melaksanakan 5R di area kerjanya.
25. *Foreman Maintenance*, mempunyai tugas dan tanggung jawab sebagai berikut:
- a. Melakukan identifikasi dan registrasi semua jenis mesin yang digunakan dalam proses produksi.
 - b. Melakukan perawatan mesin.
 - c. Melakukan perawatan berkala.
 - d. Melakukan evaluasi minimal 3 bulan dari data perbaikan.

- e. Memberikan informasi estimasi waktu penyelesaian perbaikan mesin bila lebih dari 3 hari secara tertulis ke PPIC dan produksi.
- f. Melakukan pencatatan tindakan perbaikan mesin.

4.6 Sertifikat Perusahaan

Sertifikasi memegang peranan penting dalam mengukur bagaimana kredibilitas perusahaan yang ingin bersaing secara global dan merupakan salah satu cara untuk meningkatkan secara signifikan sistem operasional perusahaan, kepercayaan pelanggan, kualitas produk, daya kompetitif, tingkat kepuasan pelanggan dan sistem monitoring atau kontrol. Namun sertifikasi belum mampu meningkatkan secara maksimal keuntungan finansial, image perusahaan, tingkat keberterimaan produk ekspor ke negara lain, tingkat pengetahuan standar bagi pegawai dan jumlah pelanggan dan manajemen mutunya. Berikut penghargaan-penghargaan yang dimiliki PT Nagasakti Component Parts:

- a. 2011: *Good Quality & Delivery Performance* PT Hi-Lex Indonesia



Gambar IV.3 2011: *Good Quality & Delivery Performance* PT Hi-Lex Indonesia
Sumber: PT Nagasakti Component Parts (2018)

- b. 2007: *Failure Mode Effect Analysis (FMEA) And Control Plan* PT Hi-Lex Indonesia



Gambar IV.4 2007: *FMEA and Control Plan* PT Hi-Lex Indonesia
Sumber: PT Nagasakti Component Parts (2018)

c. 2013: *Good Quality & Delivery Performance* PT Hi-Lex Indonesia



Gambar IV.5 2013: *Good Quality & Delivery Performance* PT Hi-Lex Indonesia
Sumber: PT Nagasaki Component Parts (2018)

d. *Reward* PT Astra Honda Motor



Gambar IV.6 *Reward* PT Astra Honda Motor
Sumber: PT Nagasaki Component Parts (2018)

e. ISO TS 16949 : 2009



Gambar IV.7 ISO TS 16949 : 2009
Sumber: PT Nagasaki Component Parts (2018)

4.7 K 81

K 81 merupakan salah satu komponen otomotif yang berada di *speedometer* Honda Beat yang berfungsi melindungi kabel terhadap air agar tidak terjadi korslet listrik pada kabel tersebut. Dapat dilihat pada Gambar IV.4 berikut:



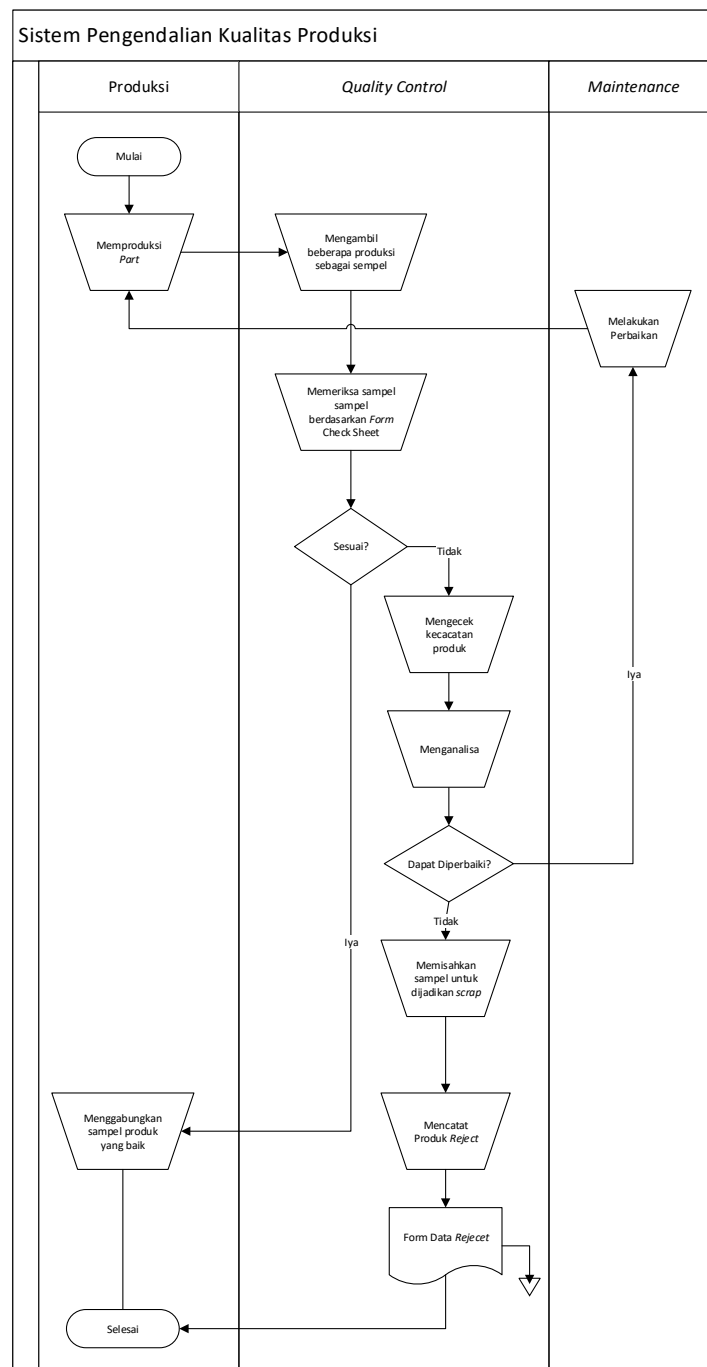
Gambar IV.8 K 81
Sumber: PT Nagasaki Component Parts (2018)

4.8 Alur Proses yang Berjalan

Alur proses pengendalian kualitas produksi yang berjalan pada PT Nagasakti Component Parts antara lain sebagai berikut:

1. Sistem pengendalian kualitas produksi pada PT Nagasakti Component Parts dimulai ketika Divisi Produksi akan melakukan proses produksi.
2. Kemudian Divisi *Quality Control* akan mengambil 2 unit/shift dari hasil produksi tersebut untuk dijadikan sampel dan kemudian melakukan pemeriksaan terhadap sampel tersebut berdasarkan *Check Sheet*.
3. Jika sampel tersebut memenuhi standar mutu maka keseluruhan hasil produksi tersebut dinyatakan sebagai produk yang baik dan kemudian sampel tersebut akan digabungkan dengan produk yang baik lainnya.
4. Tetapi jika sampel tersebut tidak sesuai dengan standar mutu maka Divisi *Quality Control* akan mengecek kecacatan yang ada pada produk tersebut untuk memastikan apakah sampel tersebut dapat diperbaiki atau tidak.
5. Jika produk tersebut tidak sesuai dengan *Check Sheet* artinya ada yang harus diperbaiki, dan akan dibawa ke Bagian *Maintenance*.
6. Menganalisa penyebab terjadinya Kecacatan produk.
7. Jika penyebab terjadi karena kerusakan mesin atau cetakan. Selanjutnya Bagian *Maintenance* akan melakukan perbaikan terhadap mesin atau cetakan tersebut berdasarkan informasi yang diberikan oleh Divisi *Quality Control*.
8. Setelah mesin atau cetakan selesai diperbaiki maka akan dibuat sampel baru yang akan diperiksa kembali oleh Divisi *Quality Control*.
9. Setelah diperiksa, jika sampel tersebut telah memenuhi standar mutu maka akan digabungkan dengan produk yang baik.
10. Sedangkan jika produk tersebut tidak dapat diperbaiki, maka produk tersebut akan dipisahkan untuk dijadikan *scrap*.
11. Kemudian Divisi *Quality Control* akan mencatat data produk cacat yang dihasilkan setiap harinya selama satu bulan ke dalam sebuah *form* yang dinamakan dengan *Form Data Part Reject*.

Gambaran umum mengenai proses bisnis sistem penanganan cacat produksi pada PT Nagasakti Component Parts dapat dilihat pada Gambar IV.9 berikut:



Gambar IV.9 Proses Bisnis Sistem Pengendalian Kualitas Proses Produksi
Sumber: Hasil Analisis (2019)

4.9 Dokumen-Dokumen Terkait Pengendalian Kualitas Proses Produksi

Divisi *Quality Control* dalam pengecekan produk cacat memiliki beberapa dokumen, yaitu:

1. Check Sheet

Merupakan sebuah *form* yang digunakan sebagai acuan ukuran suatu produk yang dihasilkan, apakah telah sesuai dengan standar mutu atau tidak.

Check sheet dapat dilihat pada Gambar IV.10 *Check Sheet*:

BAGIAN		QUALITY		CHECKSHEET PROSES PRODUKSI				Page 1 / 1	
PART NO.	32108-K81A-N300-HI	TGL PRODUKSI		MATERIAL		PRESSURE			
PART NAME	COVER CONNECTOR	LOT NO.		BERAT MATERIAL		TEMPERATURE			
MODEL		NO. MESIN		TENSILE STRENGTH		CURRING			
CUSTOMER	PT. DEM	CAVITY		ELONGATION					

NO.	PROSES	CHECK POINT	STANDAR	ALAT UKUR	SHIFT KE :		SHIFT KE :		SHIFT KE :	
					SAMPLING	SAMPLING	SAMPLING	SAMPLING	SAMPLING	SAMPLING
1.	MATERIAL	1. Tensile Stength	N	Strength machine						
		2. Elongation	%	Strength machine						
		3. Berat Material	gr	Timbangan						

Gambar IV.10 *Check Sheet*

Sumber: PT Nagasaki Component Parts (2019)

Keterangan:

1. Asal : *Customer*
2. Tujuan : Divisi *Quality Control*
3. Frekuensi : Dokumen ini diterbitkan sebanyak satu kali sesuai

Dengan barang yang akan di produksi tiap hari, lalu akan digandakan satu kali dalam sehari sesuai dengan aktivitas produksi *part* tersebut.
4. Kolom - kolom
 - *Part No* : Nomor dari bagian dokumen
 - *Part Name* : Nama Produk

- Hari, berisi keterangan dalam sehari (tiga *shift*) bagian produksi dapat memproduksi berapa banyak barang

3. Laporan *Part Not Good*

Merupakan sebuah *form* yang digunakan untuk mencatat jumlah produk cacat yang dihasilkan setiap harinya selama satu bulan, dari gambar dibawah banyaknya produksi K 81 yang dihasilkan sejumlah 4321 *pieces* dan cacat produksinya sejumlah 360 *pieces* pada tanggal 1-30 Juli 2018. Laporan Data *Part Not Good* dapat dilihat pada Gambar IV.12 berikut:

DATA REJECT COMPONENT HARIAN										
PERIODE :			DIVISI :							
NO	NAMA PART	TANGGAL		NO REGISTER	QTY	NO/QTY	CHECKER	MASALAH	TINDAKAN PERBAIKAN	
		PRODUKSI	CHECK							
1	Ac 28008	01-11-17	19-01-18	17110102	7420	97	Teri	Cadit/Revisi		
2	Ac 21069	10-01-18	19-01-18	18010102	9408	19	Teri	Cadit		
3	K-81	18-01-17	18-01-18	17114389	9371	360	Teri	Cadit		
4	Ac 26084	18-01-17	18-01-18	17120801	11760	11	Putra	Cadit		
5	Ap 31055	10-01-18	19-01-18	18010103	3042	179	Putra	Cadit		
6	Ac 20991	08-01-18	19-01-18	18010079	2182	17	Putra	Cadit		
7	Ac 10028	13-12-17	18-01-18	17120799	2898	3	Teri	Cadit		
8	Ac 30008	12-11-17	19-01-18	011802	520	59	Putra	Cadit		
9	Ac 10071	16-12-17	-11-	012806	520	180	Sumi	Cadit		
10	Ac 20175	21-12-17	-11-	17120825	7082	70+113	Teri	Cadit		
11	Ac 53069	11-01-18	-11-	18010114	9664	70	Roda/100	Putra		
12	Ac 10071	14-12-17	-11-	012801	455	69	Teri	Cadit		
13	Ac 26004	2-01-18	-11-	18010009	13000	72	Sumi	Cadit		
14	Ap 31055	10-01-18	-11-	18010099	3046	170	Putra	Cadit		
15	Ac 21040	8-01-18	-11-	0100015	624	5	Sumi	Cadit		

Gambar IV.12 *Form Data Part Reject*
Sumber: PT Nagasaki Component Parts (2019)






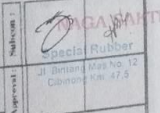
Keterangan:

1. Asal : Divisi *Quality Control*
2. Tujuan : *Manager*
3. Frekuensi : Dokumen ini diterbitkan sebanyak satu kali dalam sebulan.
4. Kolom-kolom
 - No : Nomor urutan produk yang dicatat
 - Nama *Part* : Nama produksi yang cacat

- *No Register* : No produk yang akan dikirim

4. Laporan *Problem Countermeasure*

Merupakan *form* yang digunakan untuk mencatat permasalahan dalam proses produksi. Seperti gambar IV.13 dibawah ini:

PT. DEM		PART CLAIM MEMO		Felix				Helm				Rahmat				Dulgo					
To Subcon	PT. Naga Sakti	Attention	Mrs. Juli																		
PCM No	003PCM-QA/DEM/V18	PO New Lot No	71GAD0149	Problem Occurance																	
Part Name	Cover	Qty Defect	49 pcs	Receiving	<input checked="" type="checkbox"/>	DEM															
Part No	32106-K81-N300-M1	Reject / Return to Supplier	49 pcs	Production	<input type="checkbox"/>	Customer	<input type="checkbox"/>														
Issued Date	18-Jan-18	Sortir at DEM	Yes <input checked="" type="checkbox"/> No <input type="checkbox"/>	Costumer	<input type="checkbox"/>																
Problem Date	18-Jan-18	Is this a repeating problem	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>	1	<input type="checkbox"/>	2	<input type="checkbox"/>	3	<input type="checkbox"/>	4	<input type="checkbox"/>										
Description of Defect/Problem																					
Cover Sobek																					
  <p>Label packingan</p>																					
Note : Mohon diganti dengan part Ok dan disertakan label IPP untuk pengiriman berikutnya																					
Root Cause																					
Why Defect Occur ?																					
Part sobek pada bagian tengah part	Part sobek diakibatkan , jatuh pada saat injector naik terlalu	Akibat kejatuhan injector, maka part mengalami gejala sobek	Part tidak terlihat operator sehingga proses tetap berjalan	Why	Why																
Why Defect flow out?	Proses pengecekan didivisi inspeksi kurang maksimal	seharusnya setelah pengecekan, dilakukan sampling	tujuan dari pengecekan sampling untuk mendeteksi	Why	Why																
Corrective / Preventive Action:	 			Injector untuk proses turun naik.	core pin yang jatuh menimpa dies																
Verification:				Dies bawah																	
Validation Date : 22 Januari 2018	Effective Date: 22 Jan 2018																				

Gambar IV.13 *Form Countermeasure*
Sumber: PT Nagasakti Component Parts (2019)

Keterangan:

1. Asal : Customer
2. Tujuan : PT. Nagasakti Component Parts

3. Frekuensi : Dokumen ini diterbitkan sebanyak satu kali apabila terjadi masalah dalam 1 hari.

4. Kolom – kolom

- No. : Nomor urutan produk yang dicatat
- *Name part* : Nama produk
- *Problem* : Permasalahan yang terjadi.
- *Qty Delivery* : Total kiriman barang
- *Qty Defect* : Total barang yang cacat

4.10 Analisis Permasalahan

Permasalahan yang ada pada sistem pengendalian kualitas proses produksi yang berjalan pada PT Nagasaki Component Parts dapat dilihat pada Tabel IV.1 berikut:

Tabel IV.1 Analisis Permasalahan

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
Kinerja	-	-	-	-
Informasi	Laporan data pada laporan produk cacat kurang informatif dikarenakan hanya menampilkan data jumlah produk cacat dalam sebulan.	Masih Menggunakan <i>Microsoft Excel</i> .	Laporan data cacat produksi tidak bisa dijadikan acuan dalam pengambilan keputusan untuk menentukan penyebab cacatnya produksi.	Membuat laporan cacat produksi menjadi lebih informatif dengan Menggunakan metode SPC
	Proses penginputan dan pengolahan data cacat produksi dan laporan hasil proses produksi masih dilakukan secara manual dengan menggunakan kertas.	Belum adanya sistem berbasis <i>computer</i> untuk mengolah data cacat produksi dan laporan hasil proses produksi	Kesulitan dalam mengolah data dikarenakan data yang menumpuk.	Membuat suatu sistem informasi pengendalian kualitas produksi dan cacat produksi menggunakan <i>database</i> .

Tabel IV.1 Analisis Permasalahan (Lanjutan)

Analisis	Uraian Masalah	Sebab	Akibat	Solusi
Pengendalian	Keamanan data yang lemah dikarenakan penyimpanan data berupa <i>file</i> yang terpisah dan disimpan menggunakan <i>flash disk</i> .	Belum adanya penyimpanan menggunakan <i>database</i> .	Sering terjadi kehilangan data karena <i>flashdisk</i> rentan terkena virus dan mengalami kehilangan	Membuat sistem informasi berbasis komputer menggunakan <i>database</i> sebagai media penyimpanan
	Keamanan data yang lemah dikarenakan pengelolaan data cacat produksi dilakukan menggunakan <i>Microsoft Excel</i> sehingga mudah di akses oleh siapapun.	Belum adanya sistem yang memiliki fitur hak akses.	Keamanan data yang terlalu lemah rentan terjadinya kehilangan data dan ketidak sesuaian data.	Membuat sistem informasi berbasis komputer yang memiliki fitur hak akses.
	Banyak Terjadinya produk cacat dalam proses Produksi.	Kurangnya Pengawasan dalam proses produksi	Banyaknya Jumlah produk cacat.	Membuat sistem menggunakan Metode SPC.
Ekonomi	-	-	-	-
Efisiensi	-	-	-	-
Layanan	Sistem yang ada saat ini tidak <i>user friendly</i> , banyaknya <i>table</i> dan <i>user</i> kesulitan dalam mencari mengubah dan menambah data.	Masih menggunakan <i>Microsoft Excel</i>	<i>Quality Control</i> yang tidak nyaman menggunakan sistem tersebut sehingga proses tersebut memakan waktu yang lama.	Membangun suatu sistem informasi yang <i>user friendly</i> , sehingga <i>Quality Control</i> nyaman menggunakannya

Sumber: Hasil Analisis (2019)

Rangkuman dari analisis permasalahan pada Tabel IV.1 di atas adalah belum adanya sistem berbasis komputer yang terintegrasi dengan *database* yang mengakibatkan kesulitan dalam mengolah data produksi, cacat produksi, kesulitan dalam pencarian data yang telah dicatat, serta sering terjadinya kehilangan data

dikarenakan keamanan data yang terlalu lemah. Selain itu, tingginya jumlah cacat produksi yang dihasilkan yang disebabkan oleh kurangnya pengawasan yang dilakukan oleh Divisi *Quality Control* terhadap kualitas produk.

Pada analisis di bidang Kinerja, Ekonomi dan Efisiensi tidak diketahui permasalahannya, dikarenakan kurangnya data yang dibutuhkan pada saat observasi di PT Nagasakti Component Parts.

BAB V

ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Kebutuhan-kebutuhan dari sistem informasi pengendalian kualitas cacat produksi yang diusulkan dapat dilihat pada Tabel V.1 berikut:

Tabel V.1 Analisis Kebutuhan Sistem

Masalah	Kebutuhan <i>User</i>	Solusi	Kebutuhan Sistem	
			<i>Functional Requirement</i>	<i>Non Functional Requirement</i>
Proses penginputan dan pengolahan data cacat produksi masih dilakukan secara <i>manual</i> , dengan cara mencatat di kertas dan disimpan pada <i>file-file</i> terpisah	Sistem yang dapat membantu mengelola data cacat produksi	Merancang dan membangun suatu sistem informasi pengendalian kualitas menggunakan basis data	Mengelola data cacat produksi	- Data cacat produksi memiliki penyebab cacat yang berbeda pada tiap harinya, dan memiliki penyebab cacat minimal 1 penyebab dan maksimal 6 penyebab.
			Mengelola <i>Master Data</i> Produk	Data produk dapat dikelola pada setiap harinya secara terus menerus berdasarkan hasil produksi.

Tabel V.1 Analisis Kebutuhan Sistem (Lanjutan)

Masalah	Kebutuhan <i>User</i>	Solusi	Kebutuhan Sistem	
			<i>Functional Requirement</i>	<i>Non Functional Requirement</i>
Laporan cacat produksi kurang informatif	Laporan cacat produksi yang lebih informatif	Membuat laporan grafik, cacat produksi, hasil produksi menjadi lebih informatif	Mencetak laporan grafik, cacat produksi, hasil produksi	- Laporan grafik menghasilkan diagram pareto dan peta kendali, laporan cacat produksi menghasilkan penyebab apa saja terjadinya kecacataan, dan laporan hasil produksi untuk mengetahui perbandingan cacat dan ok pada produk yang dihasilkan.
Keamanan data yang terlalu lemah	Sistem yang aman digunakan sesuai dengan	Membangun suatu sistem informasi yang dapat	- Melakukan <i>login</i>	- Setiap <i>user</i> memiliki hak akses yang

	hak aksesnya masing-masing	mengelola hak akses	- Mengelola <i>Master Data User</i>	berbeda pada setiap <i>actor</i>
--	----------------------------	---------------------	-------------------------------------	----------------------------------

Sumber: Hasil Analisis (2019)

5.1.1 *Functional Requirement*

Functional Requirement dari sistem informasi pengendalian kualitas cacat produksi yang diusulkan antara lain:

1. Admin dan *Quality Control* mengelola *Master Data* Penyebab Cacat.
2. Admin dan *Quality Control* mengelola data cacat produksi.
3. Admin dan *Quality Control* mencetak laporan diagram.
4. Admin dan *Quality Control* mencetak laporan cacat produksi.
5. Admin dan *Quality Control* mencetak laporan hasil produksi.
6. Admin, Produksi, *Quality Control* melakukan *login*.
7. Admin mengelola *Master Data User*.
8. Admin dan Produksi mengelola *Master Data Produk*.
9. Admin dan Produksi mengelola Data Transaksi Hasil Produksi.

5.1.2 *Non Functional Requirement*

Non Functional Requirement dari sistem informasi pengendalian kualitas cacat produksi yang diusulkan antara lain:

1. *Login* dengan memasukkan *username* dan *password*.
2. *Username* dan *password* terdiri dari gabungan huruf dan angka.
3. Data Laporan Hasil Produksi dan Cacat Produksi bisa dilihat dan dicetak berdasarkan tanggal mulai dan tanggal akhir yang akan mau dilaporkan.
4. Hasil Laporan akan muncul CL, UCL, LCL dan proporsi, data tersebut bisa digunakan sebagai acuan untuk pembuatan peta kendali.
5. Hasil Laporan akan muncul penyebab mesin, metode, manusia, material data tersebut bisa digunakan sebagai acuan untuk diagram pareto.

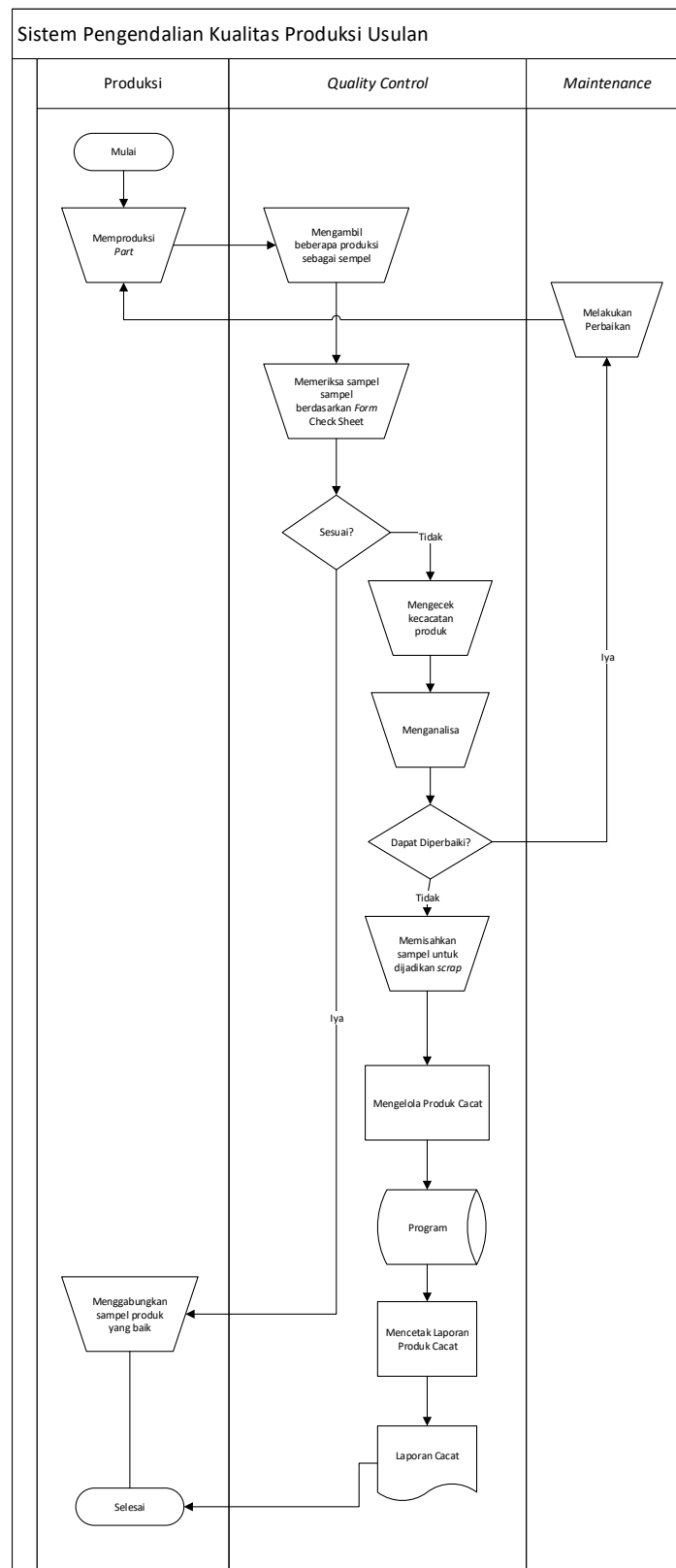
5.2 Alur Proses Pengendalian Kualitas Usulan

Alur proses pengendalian kualitas cacat produksi yang diusulkan antara lain sebagai berikut:

1. Sistem pengendalian kualitas produksi pada PT Nagasaki Component Parts dimulai ketika Divisi Produksi akan melakukan proses produksi.
2. Kemudian Divisi *Quality Control* akan mengambil 2 unit/*shift* dari hasil produksi tersebut untuk dijadikan sampel dan kemudian melakukan pemeriksaan terhadap sampel tersebut berdasarkan *Check Sheet*.
3. Jika sampel tersebut memenuhi standar mutu maka keseluruhan hasil produksi tersebut dinyatakan sebagai produk yang baik dan kemudian sampel tersebut akan digabungkan dengan produk yang baik lainnya.
4. Tetapi jika sampel tersebut tidak sesuai dengan standar mutu maka Divisi *Quality Control* akan mengecek kecacatan yang ada pada produk tersebut untuk memastikan apakah sampel tersebut dapat diperbaiki atau tidak.
5. Jika produk tersebut tidak sesuai dengan *Check Sheet* artinya ada yang harus diperbaiki, dan akan dibawa ke Bagian *Maintenance*.
6. Menganalisa penyebab terjadinya Kecacatan produk.
7. Jika penyebab terjadi karena kerusakan mesin atau cetakan. Selanjutnya Bagian *Maintenance* akan melakukan perbaikan terhadap mesin atau cetakan tersebut berdasarkan informasi yang diberikan oleh Divisi *Quality Control*.
8. Setelah mesin atau cetakan selesai diperbaiki maka akan dibuat sampel baru yang akan diperiksa kembali oleh Divisi *Quality Control*.
9. Setelah diperiksa, jika sampel tersebut telah memenuhi standar mutu maka akan digabungkan dengan produk yang baik.
10. Sedangkan jika produk tersebut tidak dapat diperbaiki, maka produk tersebut akan dipisahkan untuk dijadikan *scrap*.
11. Kemudian Divisi *Quality Control* akan mengelola data produk cacat yang dihasilkan setiap harinya selama satu bulan ke dalam sebuah *form* yang dinamakan dengan *Form Data Cacat Produksi*.

12. Lalu data produk cacat tersebut akan dikelola, yang akan menghasilkan laporan cacat produksi.
13. Kemudian *Quality Control* akan mencetak laporan cacat produksi selama periode tertentu yang kemudian akan diserahkan kepada Produksi untuk dijadikan bahan evaluasi.

Gambaran mengenai proses pengendalian kualitas yang diusulkan dapat dilihat pada Gambar V.1 berikut:



Gambar V.1 Alur Proses Pengendalian Kualitas Usulan

Sumber: Hasil Analisis (2019)

5.3 Pemodelan Sistem Informasi Pengendalian Kualitas Usulan

Pada tahap ini dilakukan pemodelan sistem dengan menggunakan UML (*Unified Modelling Language*) di antaranya adalah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*, dan *Deployment Diagram*.

5.3.1 Use Case Diagram

Use Case Diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case Diagram* sistem informasi pengendalian kualitas cacat produksi pada PT Nagasaki ComponentvParts yang diusulkan dapat dilihat pada Gambar V.2.



Gambar V.2 *Use Case* Usulan
Sumber: Hasil Analisis (2019)

5.3.2 Use Case Description

Use Case Description dari sistem informasi pengendalian kualitas cacat produksi yang diusulkan antara lain sebagai berikut:

1. Use Case Description Melakukan Login

Tabel V.2 *Use Case Description* Melakukan Login

Nama Use Case	Melakukan Login
Actor	Admin, Produksi, dan Quality Control
Normal Flow	<ol style="list-style-type: none"> 1. Actor membuka aplikasi. 2. Sistem akan menampilkan halaman login. 3. Actor memasukkan <i>username</i> dan <i>password</i>. 4. Actor mengklik tombol Login. 5. Sistem akan memvalidasi <i>username</i> dan <i>password</i>. 6. Jika <i>username</i> dan <i>password</i> valid, sistem akan menampilkan halaman utama. 7. Jika <i>username</i> dan <i>password</i> tidak valid, sistem akan menampilkan pesan kesalahan.

Sumber: Hasil Analisis (2019)

2. Use Case Description Mengelola Master Data Produk

Tabel V.3 *Use Case Description* Mengelola Master Data Produk

Nama Use Case	Mengelola Master Data Produk
Actor	Admin, Produksi
Normal Flow	<ol style="list-style-type: none"> 1. Admin atau Produksi memilih menu <i>Master Data</i>, lalu memilih submenu Data Produk. 2. Sistem akan menampilkan Data Produk. 3. Admin atau Produksi dapat mengklik tombol tambah data, ubah dan hapus. 4. Jika mengklik tombol tambah data maka sistem akan menampilkan <i>form</i> tambah data.

Tabel V.3 *Use Case Description* Mengelola *Master Data* Produk (Lanjutan)

<i>Normal Flow</i>	<p>5. <i>Admin</i> atau Produksi memasukkan data, lalu mengklik tombol simpan. Sistem akan memperbarui data di dalam <i>database</i>.</p> <p>6. Jika mengklik tombol ubah maka sistem akan menampilkan <i>form</i> ubah data.</p> <p>7. <i>Admin</i> atau Produksi mengubah data, lalu mengklik tombol ubah. Sistem akan memperbarui data di dalam <i>database</i>.</p> <p>8. Jika mengklik tombol hapus maka sistem akan menghapus data di dalam <i>database</i>.</p>
--------------------	--

Sumber: Hasil Analisis (2019)

3. *Use Case Description* Mengelola *Master Data* Penyebab CacatTabel V.4 *Use Case Description* Mengelola *Master Data* Penyebab cacat

Nama <i>Use Case</i>	Mengelola <i>Master Data</i> Penyebab Cacat
<i>Actor</i>	<i>Admin, Quality Control</i>
<i>Normal Flow</i>	<p>1. <i>Admin</i> atau <i>Quality Control</i> memilih menu <i>Master Data</i>, lalu memilih submenu Data Penyebab Cacat.</p> <p>2. Sistem akan menampilkan Data Penyebab Cacat.</p> <p>3. <i>Admin</i> atau <i>Quality Control</i> dapat mengklik tombol tambah data, ubah dan hapus.</p> <p>4. Jika mengklik tombol tambah data maka sistem akan menampilkan <i>form</i> tambah data.</p> <p>5. <i>Admin</i> atau <i>Quality Control</i> memasukkan data, lalu mengklik tombol simpan. Sistem akan memperbarui data di dalam <i>database</i>.</p> <p>6. Jika mengklik tombol ubah maka sistem akan menampilkan <i>form</i> ubah data.</p>

Tabel V.4 *Use Case Description* Mengelola Master Data Penyebab cacat (Lanjutan)

<i>Normal Flow</i>	<p>7. <i>Admin, Quality Control</i> mengubah data, lalu mengklik tombol ubah. Sistem akan memperbarui data di dalam <i>database</i>.</p> <p>8. Jika mengklik tombol hapus maka sistem akan menghapus data di dalam <i>database</i>.</p>
--------------------	---

Sumber: Hasil Analisis (2019)

4. *Use Case Description* Mengelola Master Data *User*Tabel V.5 *Use Case Description* Mengelola Master Data *User*

Nama <i>Use Case</i>	Mengelola <i>Master Data</i> Pengguna
<i>Actor</i>	<i>Admin</i>
<i>Normal Flow</i>	<p>1. <i>Admin</i> memilih menu <i>Master Data</i>, lalu memilih submenu <i>Data User</i>.</p> <p>2. Sistem akan menampilkan <i>Data User</i>.</p> <p>3. <i>Admin</i> dapat mengklik tombol tambah data, ubah dan hapus.</p> <p>4. <i>Admin</i> memilih menu <i>Master Data</i>, lalu memilih submenu <i>Data User</i>.</p> <p>5. Sistem akan menampilkan <i>Data User</i>.</p> <p>6. <i>Admin</i> dapat mengklik tombol tambah data, ubah dan hapus.</p> <p>7. Jika mengklik tombol tambah data maka sistem akan menampilkan <i>form</i> tambah data.</p> <p>8. <i>Admin</i> memasukkan data, lalu mengklik tombol simpan. Sistem akan memperbarui data di dalam <i>database</i>.</p> <p>9. Jika mengklik tombol ubah maka sistem akan menampilkan <i>form</i> ubah data.</p> <p>10. <i>Admin</i> mengubah data, lalu mengklik tombol ubah. Sistem akan memperbarui data di dalam <i>database</i>.</p>

Tabel V.5 *Use Case Description* Mengelola Master Data User (Lanjutan)

<i>Normal Flow</i>	11. Jika mengklik tombol hapus maka sistem akan menghapus data di dalam <i>database</i> .
--------------------	---

Sumber: Hasil Analisis (2019)

5. *Use Case Description* Mengelola Data Transaksi Hasil ProduksiTabel V.6 *Use Case Description* Mengelola Data Transaksi Hasil Produksi

Nama <i>Use Case</i>	Mengelola Master Data Hasil Produksi
<i>Actor</i>	<i>Admin</i> , Produksi
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. <i>Admin</i> atau Produksi memilih menu Data Transaksi, lalu memilih submenu Hasil Produksi. 2. Sistem akan menampilkan Hasil Produksi. 3. <i>Admin</i> atau Produksi dapat mengklik tombol tambah data, ubah dan hapus. 4. Jika mengklik tombol tambah data maka sistem akan menampilkan <i>form</i> tambah data. 5. <i>Admin</i> atau Produksi memilih menu Data Transaksi, lalu memilih submenu Hasil Produksi. 6. Sistem akan menampilkan Hasil Produksi. 7. <i>Admin</i> atau Produksi dapat mengklik tombol tambah data, ubah dan hapus. 8. Jika mengklik tombol tambah data maka sistem akan menampilkan <i>form</i> tambah data. 9. <i>Admin</i> atau Produksi memasukkan data, lalu mengklik tombol simpan. Sistem akan memperbarui data di dalam <i>database</i>. 10. Jika mengklik tombol ubah maka sistem akan menampilkan <i>form</i> ubah data. 11. <i>Admin</i> atau Produksi mengubah data, lalu mengklik tombol ubah. Sistem akan memperbarui data di dalam <i>database</i>.

Tabel V.6 *Use Case Description* Mengelola Data Transaksi Hasil Produksi (Lanjutan)

<i>Normal Flow</i>	12. Jika mengklik tombol hapus maka sistem akan menghapus data di dalam <i>database</i> .
--------------------	---

Sumber: Hasil Analisis (2019)

6. *Use Case Description* Mengelola Data Transaksi Cacat ProduksiTabel V.7 *Use Case Description* Mengelola Data Transaksi Cacat Produksi

Nama <i>Use Case</i>	Mengelola <i>Master</i> Data Transaksi Cacat Produksi
<i>Actor</i>	<i>Admin, Quality Control</i>
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. <i>Admin</i> atau <i>Quality Control</i> memilih menu <i>Master Data</i>, lalu memilih submenu Data Produk. 2. Sistem akan menampilkan Data Barang. 3. <i>Admin</i> atau <i>Quality Control</i> dapat mengklik tombol <i>input</i> data, dan <i>detail</i>. 4. Jika mengklik tombol <i>input</i> data maka sistem akan menampilkan <i>form</i> tambah data. 5. <i>Admin</i> atau <i>Quality Control</i> memilih menu <i>Master Data</i>, lalu memilih submenu Data Produk. 6. Sistem akan menampilkan Data Barang. 7. <i>Admin</i> atau <i>Quality Control</i> dapat mengklik tombol <i>input</i> data, dan <i>detail</i>. 8. Jika mengklik tombol <i>input</i> data maka sistem akan menampilkan <i>form</i> tambah data. 9. <i>Admin</i> atau <i>Quality Control</i> memasukkan data, lalu mengklik tombol simpan. Sistem akan memperbarui data di dalam <i>database</i>. 10. Jika mengklik tombol <i>detail</i> maka sistem akan menampilkan <i>form detail</i> data.

Sumber: Hasil Analisis (2019)

7. *Use Case Description* Mencetak Laporan Grafik

Tabel V.8 *Use Case Description* Mencetak Laporan Grafik

Nama <i>Use Case</i>	Mencetak Laporan Grafik
<i>Actor</i>	<i>Admin, Quality Control</i>
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. <i>Admin</i> atau <i>Quality Control</i> memilih menu Laporan Grafik. 2. <i>Admin</i> atau <i>Quality Control</i> memilih nama produk, tanggal mulai dan tanggal akhir pada laporan yang ingin dicetak. 3. <i>Admin</i> atau <i>Quality Control</i> memilih menu Laporan Grafik. 4. <i>Admin</i> atau <i>Quality Control</i> memilih nama produk, tanggal mulai dan tanggal akhir pada laporan yang ingin dicetak. 5. Lalu <i>Admin</i> atau <i>Quality Control</i> mengklik tombol lihat. 6. Sistem akan menampilkan laporan dari tanggal mulai dan tanggal akhir yang dipilih. 7. <i>Admin</i> atau <i>Quality Control</i> mengklik tombol cetak. 8. Sistem akan mencetak laporan.

Sumber: Hasil Analisis (2019)

8. *Use Case Description* Mencetak Laporan Hasil Produksi

Tabel V.9 *Use Case Description* Mencetak Laporan Hasil Produksi

Nama <i>Use Case</i>	Mencetak Laporan Hasil Produksi
<i>Actor</i>	<i>Admin, Produksi</i>
<i>Normal Flow</i>	<ol style="list-style-type: none"> 1. <i>Admin</i> atau <i>Produksi</i> memilih menu Hasil Produksi. 2. <i>Admin</i> atau <i>Produksi</i> memilih nama produk, tanggal mulai dan tanggal akhir pada laporan yang ingin dicetak. 3. Lalu <i>Admin</i> atau <i>Produksi</i> mengklik tombol lihat.

	<ol style="list-style-type: none"> 4. Sistem akan menampilkan laporan dari tanggal mulai dan tanggal akhir yang dipilih. 5. <i>Admin</i> atau Produksi mengklik tombol cetak. 6. Sistem akan mencetak laporan.
--	---

Sumber: Hasil Analisis (2019)

9. *Use Case Description* Mencetak Laporan Cacat Produksi

Tabel V.10 *Use Case Description* Mencetak Laporan Cacat Produksi

Nama <i>Use Case</i>	Mencetak Laporan Cacat Produksi
Actor	<i>Admin, Quality Control</i>
Normal Flow	<ol style="list-style-type: none"> 1. <i>Admin</i> atau <i>Quality Control</i> memilih menu Laporan Cacat Produksi. 2. <i>Admin</i> atau <i>Quality Control</i> memilih tanggal mulai dan tanggal akhir pada laporan yang ingin dicetak. 3. Lalu <i>Admin</i> atau Produksi mengklik tombol lihat. 4. Sistem akan menampilkan laporan dari tanggal mulai dan tanggal akhir yang dipilih. 5. <i>Admin</i> atau <i>Quality Control</i> mengklik tombol cetak. 6. Sistem akan mencetak laporan.

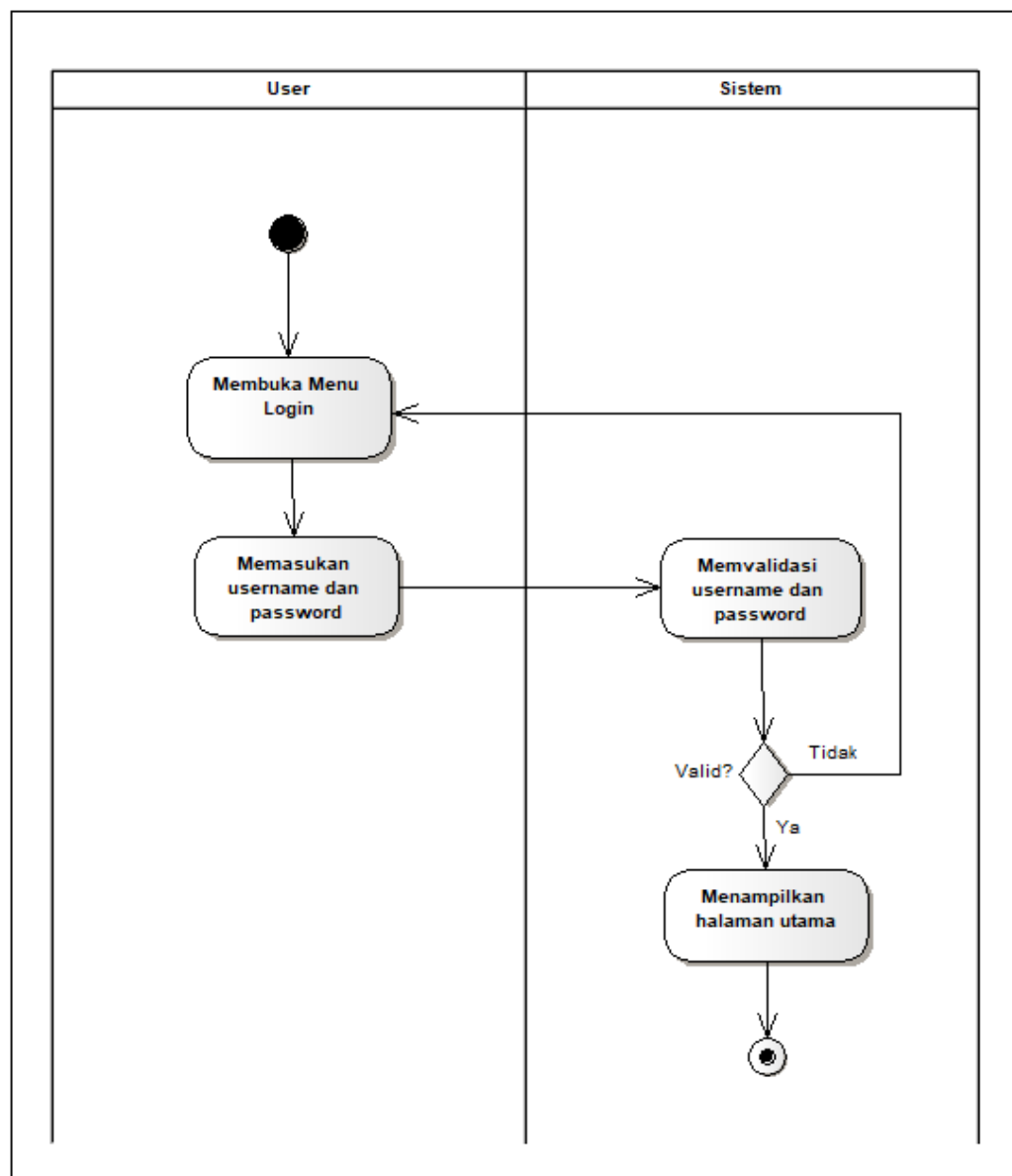
Sumber: Hasil Analisis (2019)

5.4 *Activity Diagram*

Activity Diagram menggambarkan urutan aktivitas proses bisnis pada suatu sistem. *Activity Diagram* sistem informasi pengendalian kualitas cacat produksi pada PT Nagasakti Component Parts yang diusulkan antara lain sebagai berikut:

1. *Activity Diagram* Melakukan Login

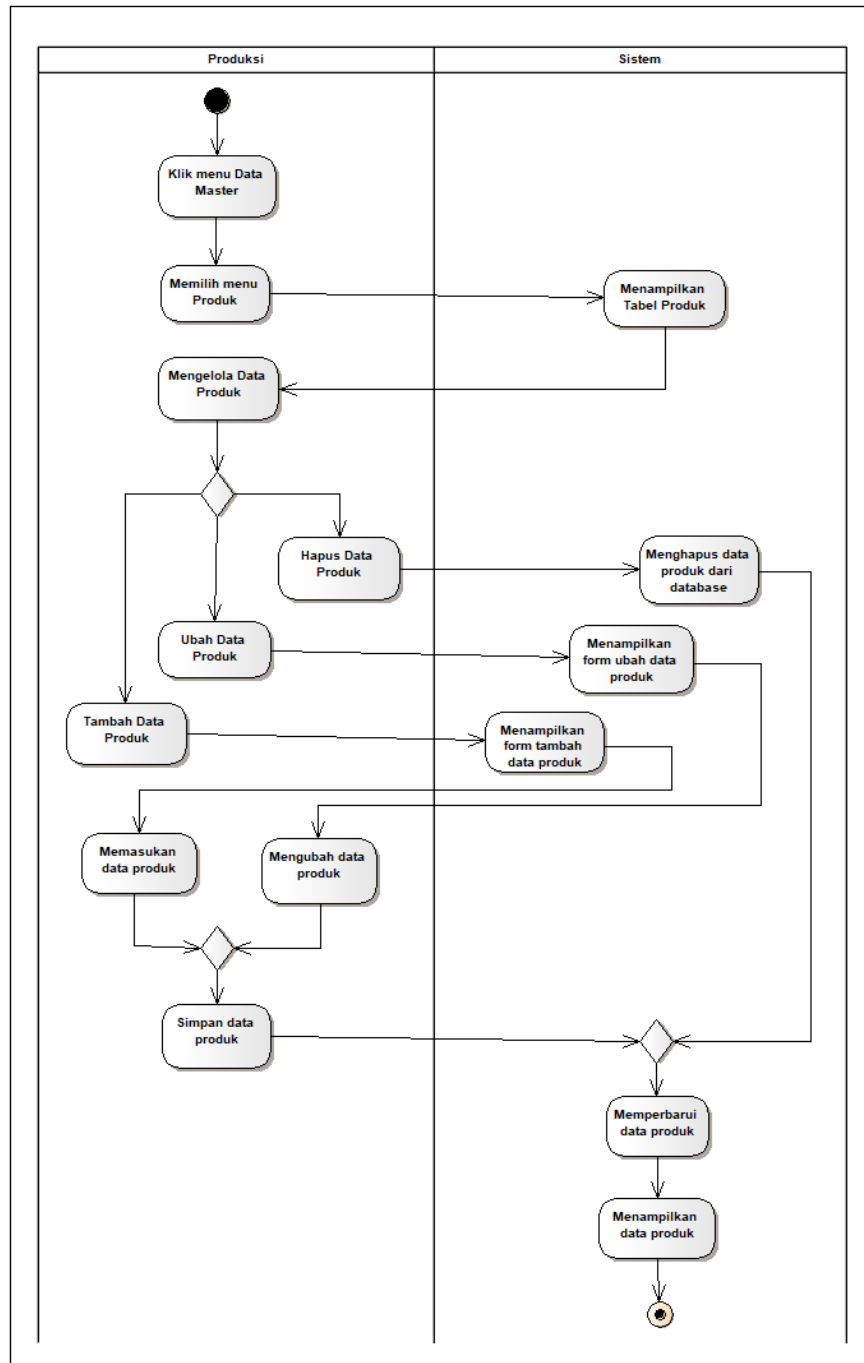
Activity Diagram melakukan login menjelaskan tentang aktivitas melakukan login bagi user, sesuai dengan hak akses yang ada di sistem tersebut Berikut Gambar V.3:



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

2. Activity Diagram Mengelola Master Data Produk

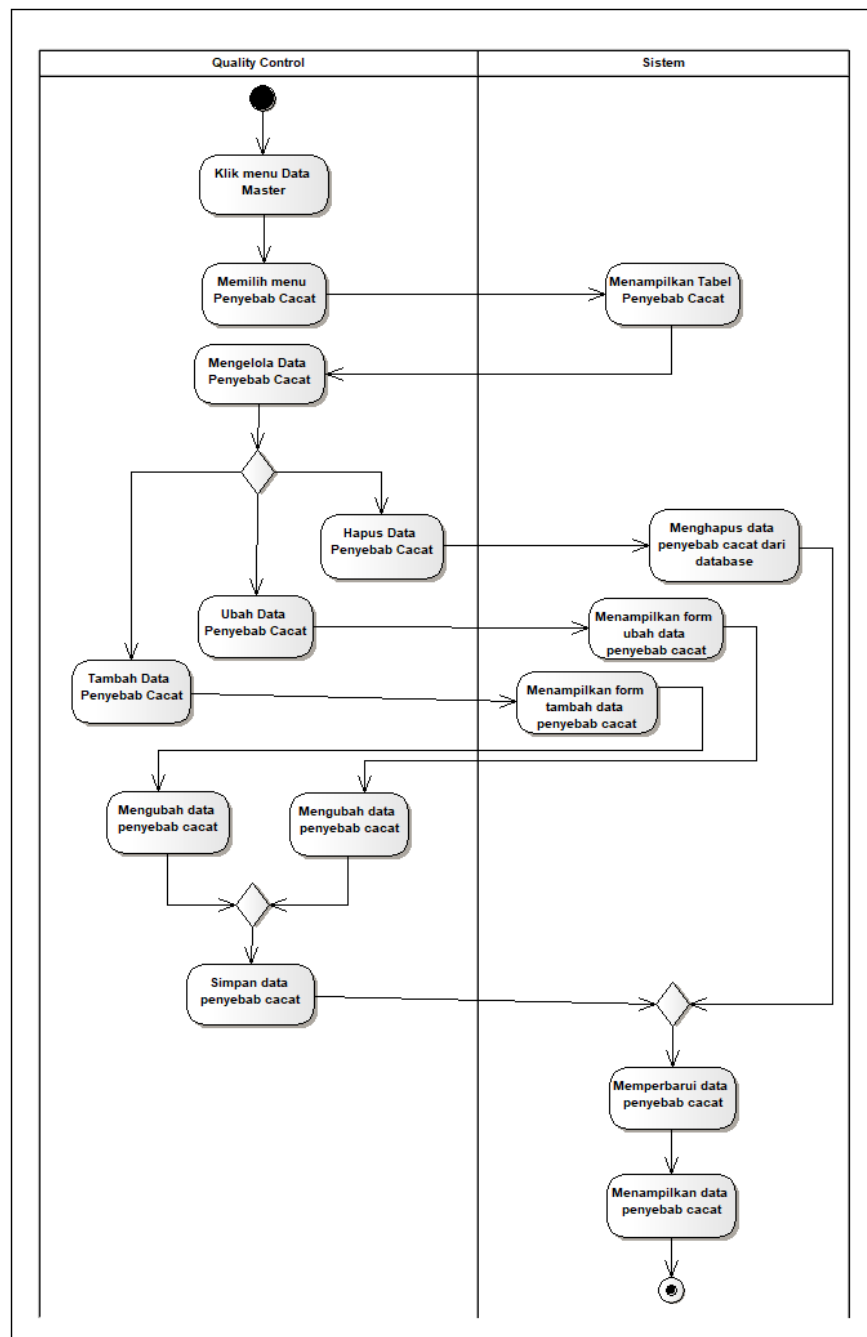
Activity Diagram mengelola Data Master Produk menjelaskan tentang aktivitas mengelola Data Master Produk bagi *user* Produksi dan *Admin*, kegiatan dalam mengelola Data Master Produk adalah Menambah, Mengubah, Menghapus. Berikut Gambar V.4:



Gambar V.4 Activity Diagram Mengelola Master Data Produk
Sumber: Hasil Analisis (2019)

3. *Activity Diagram* Mengelola Master Data Penyebab Cacat

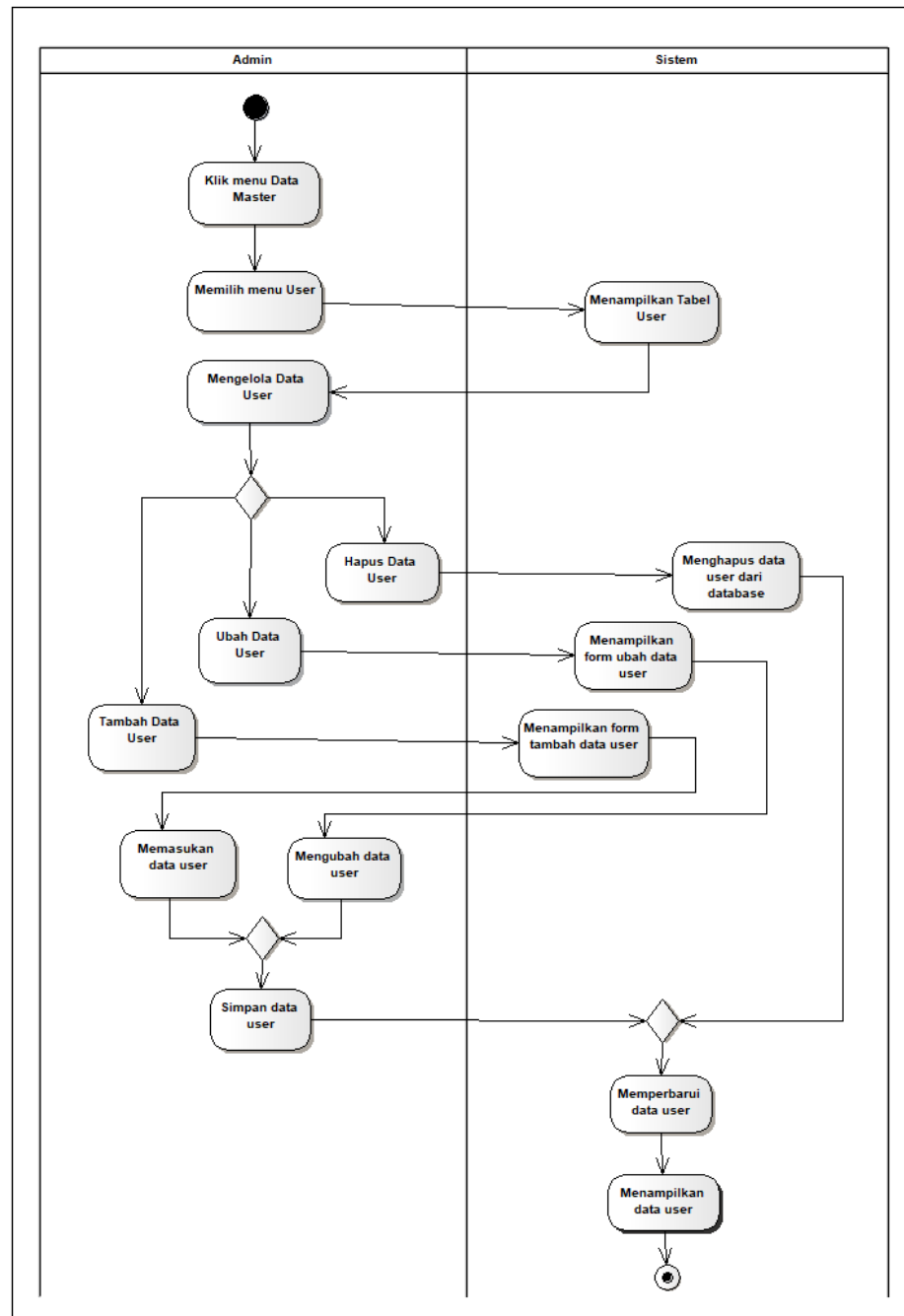
Activity Diagram mengelola Data Master Penyebab Cacat menjelaskan tentang aktivitas mengelola Data Master Penyebab Cacat bagi *user Quality Control* dan *Admin*, kegiatan dalam mengelola Data Master Penyebab Cacat adalah Menambah, Mengubah, Menghapus. Berikut Gambar V.5:



Gambar V.5 *Activity Diagram* Mengelola Master Data Penyebab Cacat
Sumber: Hasil Analisis (2019)

4. Activity Diagram Mengelola Master Data User

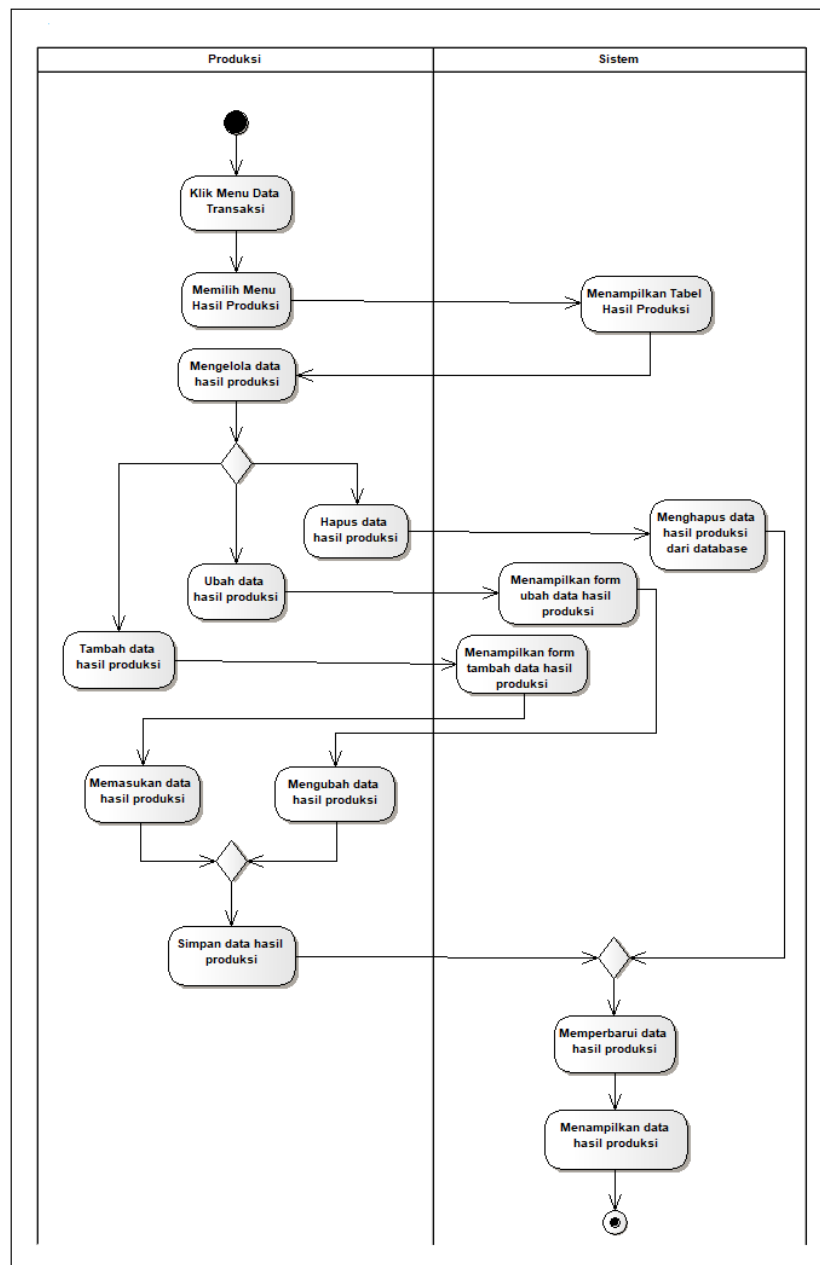
Activity Diagram mengelola Data Master User menjelaskan tentang aktivitas mengelola Data Master Penyebab Cacat bagi *user Quality CAdmin*, kegiatan dalam mengelola Data Master User adalah Menambah, Mengubah, Menghapus. Berikut Gambar V.6:



Gambar V.6 Activity Diagram Mengelola Master Data User
Sumber: Hasil Analisis (2019)

5. Activity Diagram Mengelola Data Transaksi Hasil Produksi

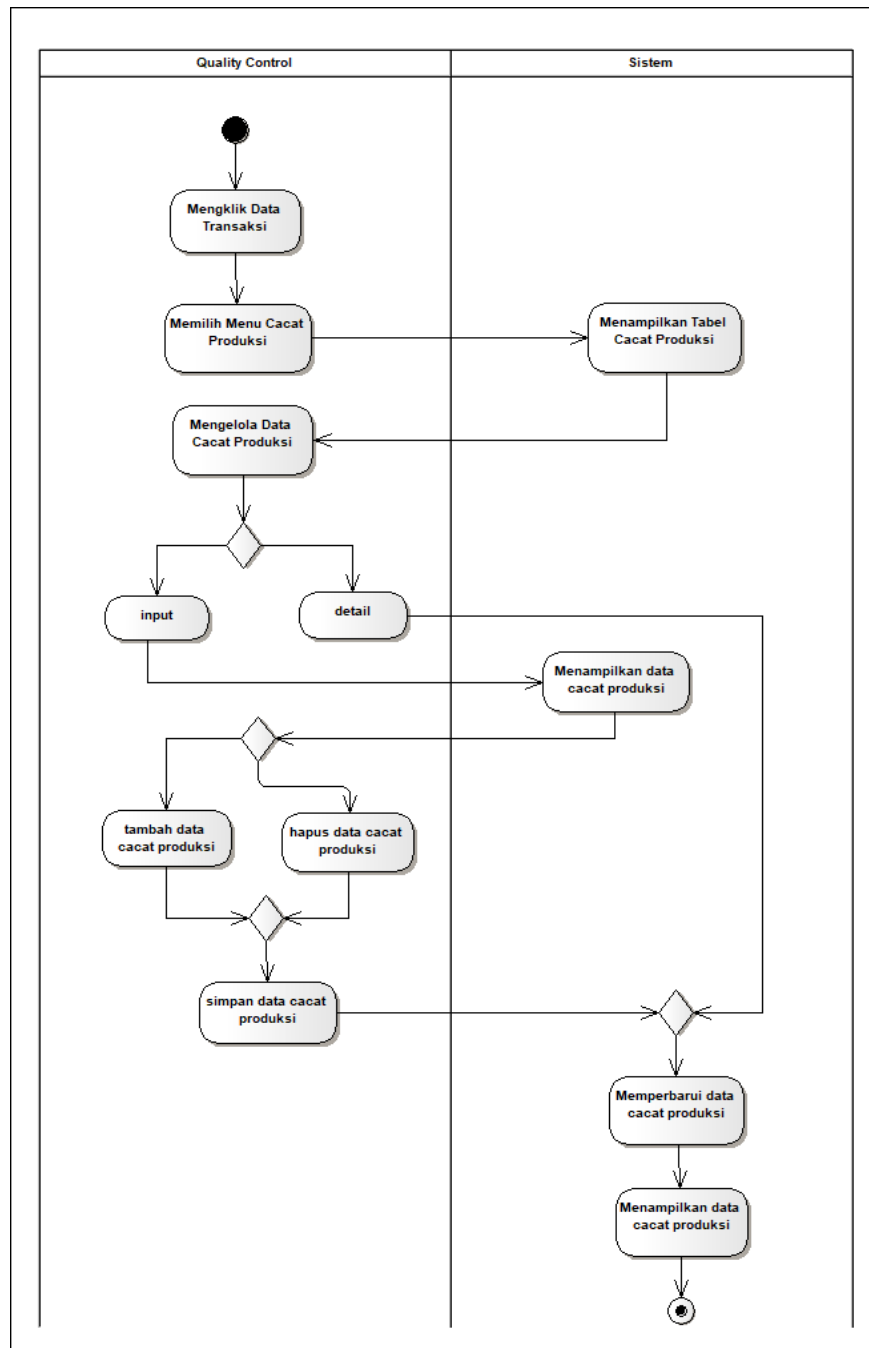
Activity Diagram mengelola Data Transaksi Hasil Produksi menjelaskan tentang aktivitas mengelola Data Transaksi Hasil Produksi bagi *user* Produksi dan *Admin*, kegiatan dalam mengelola Data Transaksi Hasil Produksi adalah Menambah, Mengubah, Menghapus. Berikut Gambar V.7:



Gambar V.7 Activity Diagram Mengelola Data Transaksi Hasil Produksi
Sumber: Hasil Analisis (2019)

6. Activity Diagram Mengelola Data Transaksi Cacat Produksi

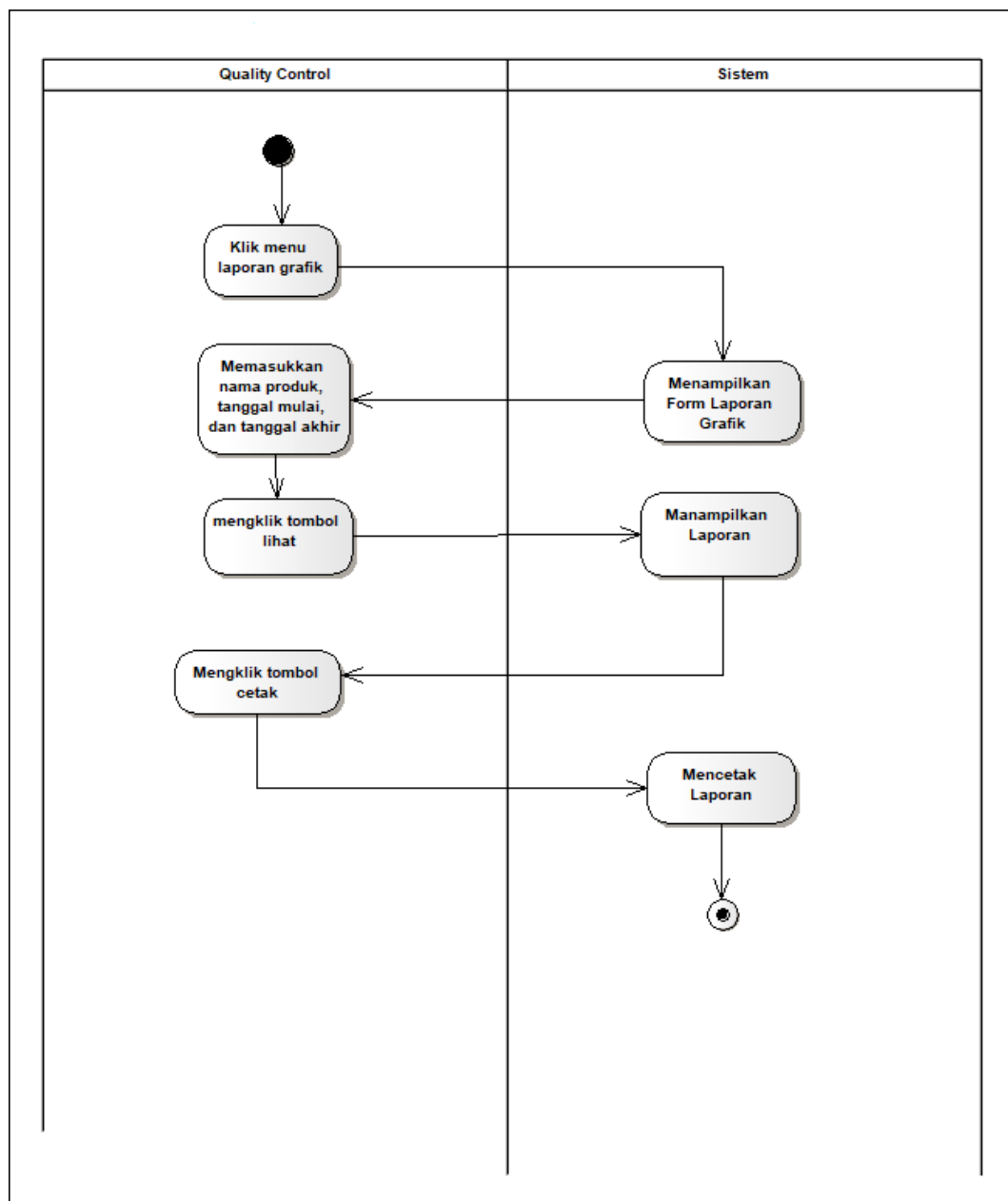
Activity Diagram mengelola Data Transaksi Cacat Produksi menjelaskan tentang aktivitas mengelola Data Transaksi Cacat Produksi bagi *user Quality Control* dan *Admin*, kegiatan dalam mengelola Data Transaksi Cacat Produksi adalah *Input*, *Detail*. Berikut Gambar V.8:



Gambar V.8 Activity Diagram Mengelola Data Transaksi Cacat Produksi
Sumber: Hasil Analisis (2019)

7. Activity Diagram Mencetak Laporan Grafik

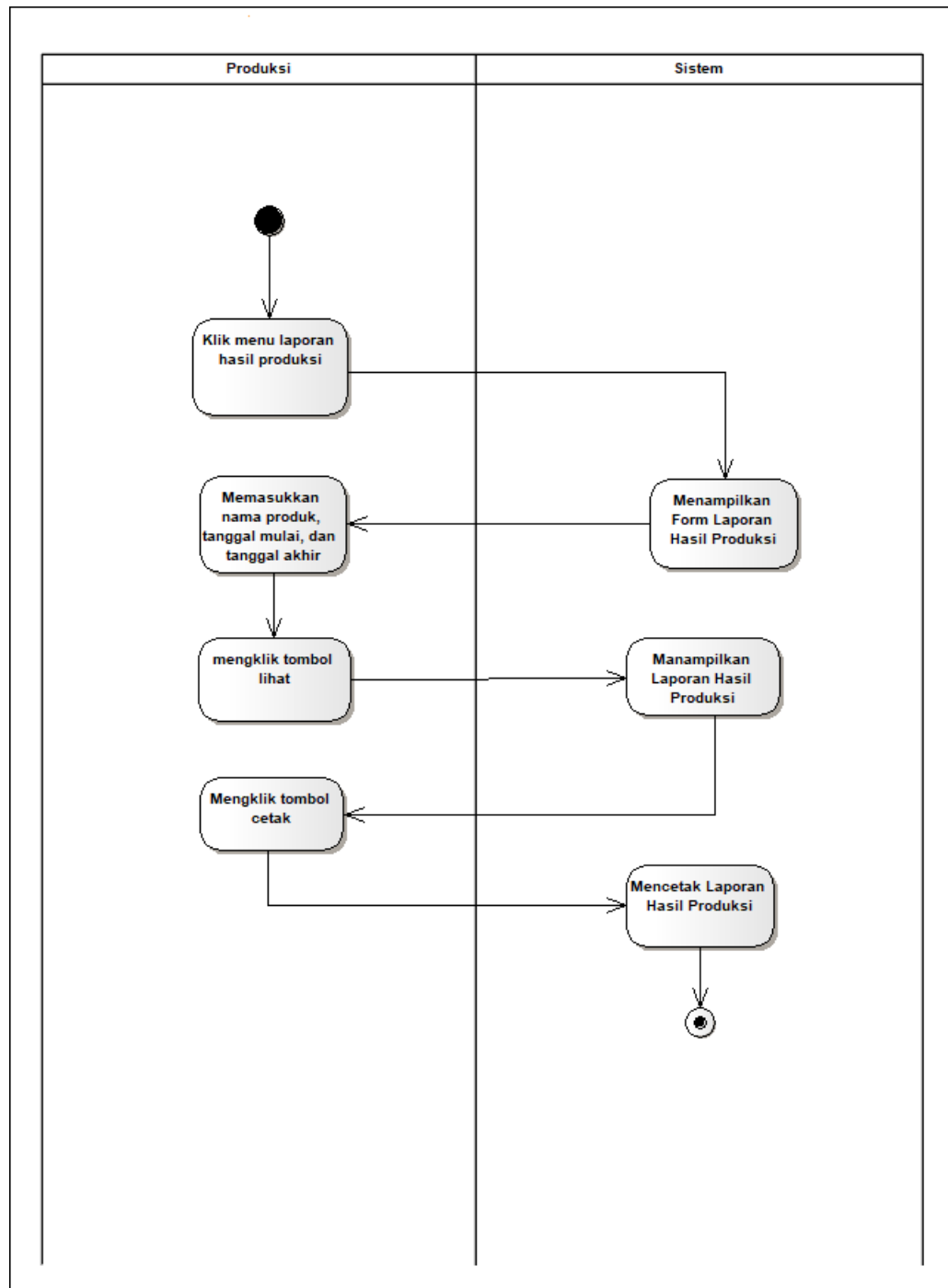
Activity Diagram Mencetak Laporan Grafik satu kegiatan untuk mencetak dan melihat yang diinput sesuai dengan nama produk, tanggal mulai, dan tanggal akhir yang dipanggil dalam *form* tersebut. Berikut Gambar V.9:



Gambar V.9 Activity Diagram Mencetak Laporan Grafik
Sumber: Hasil Analisis (2019)

8. Activity Diagram Mencetak Laporan Hasil Produksi

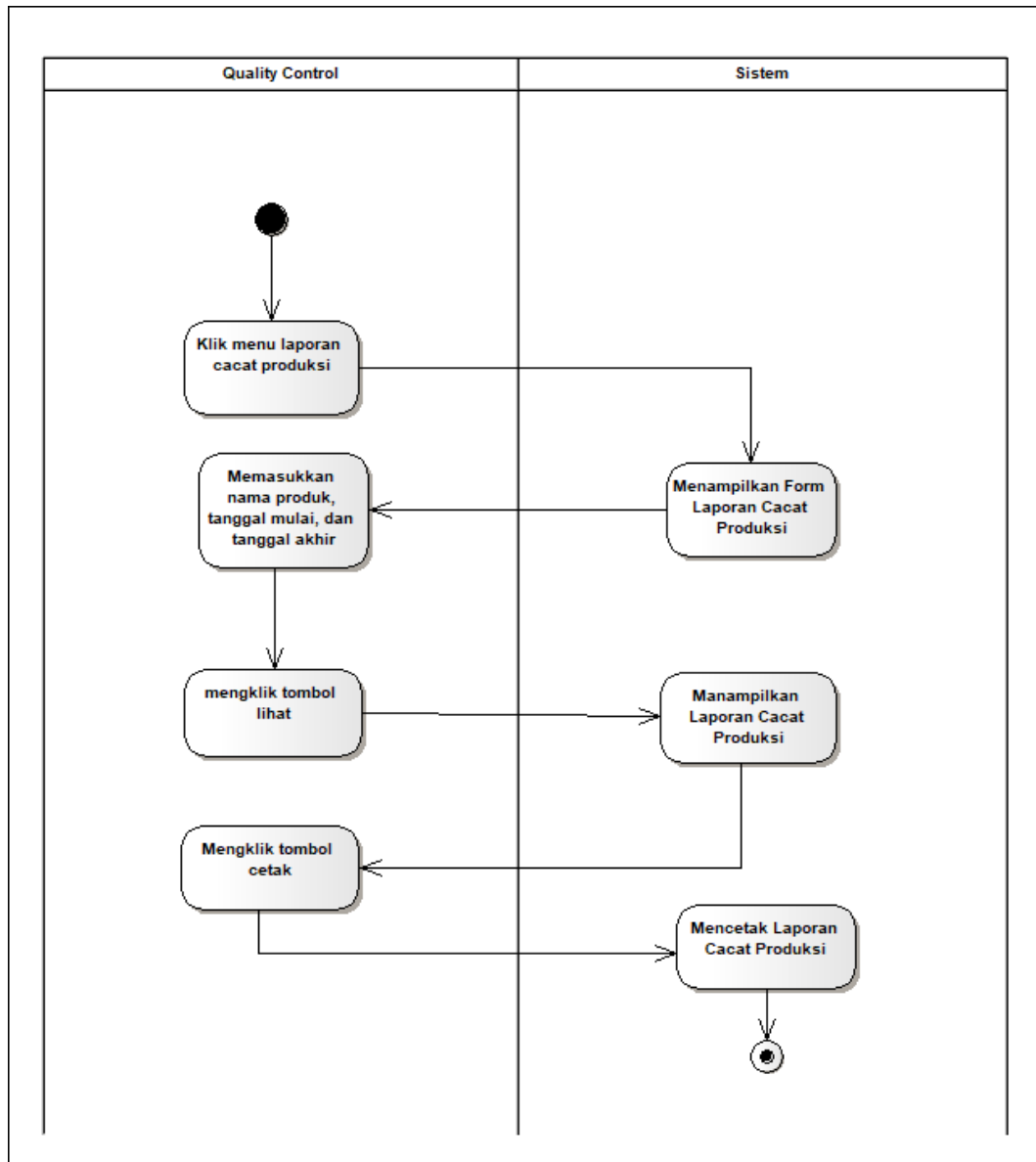
Activity Diagram Mencetak Laporan Hasil Produksi satu kegiatan untuk mencetak dan melihat yang diinput sesuai dengan nama produk, tanggal mulai, dan tanggal akhir yang dipanggil dalam form tersebut. Berikut Gambar V.10:



Gambar V.10 Activity Diagram Mencetak Laporan Hasil Produksi
Sumber: Hasil Analisis (2019)

9. *Activity Diagram* Mencetak Laporan Cacat Produksi

Activity Diagram Mencetak Laporan Cacat Produksi satu kegiatan untuk mencetak dan melihat yang *diinput* sesuai dengan nama produk, tanggal mulai, dan tanggal akhir yang dipanggil dalam *form* tersebut. Berikut Gambar V.11:



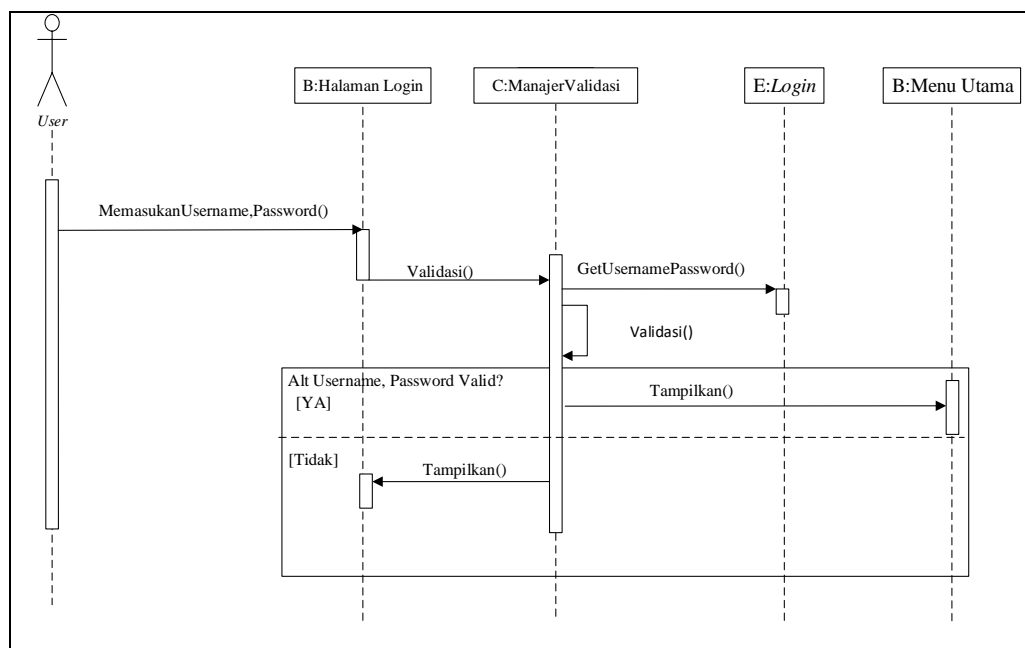
Gambar V.11 *Activity Diagram* Mencetak Laporan Cacat Produksi
Sumber: Hasil Analisis (2019)

5.5 Sequence Diagram

Sequence diagram merupakan diagram untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

1. Sequence Diagram Melakukan Login

Sequence Diagram Melakukan Login adalah jenis interaksi yang menggambarkan tentang proses *login* untuk karyawan sesuai dengan hak akses masing masing jabatan karyawan tersebut dalam *use case* melakukan *login*, berikut adalah *sequence* diagram melakukan *login*. Berikut Gambar V.12:

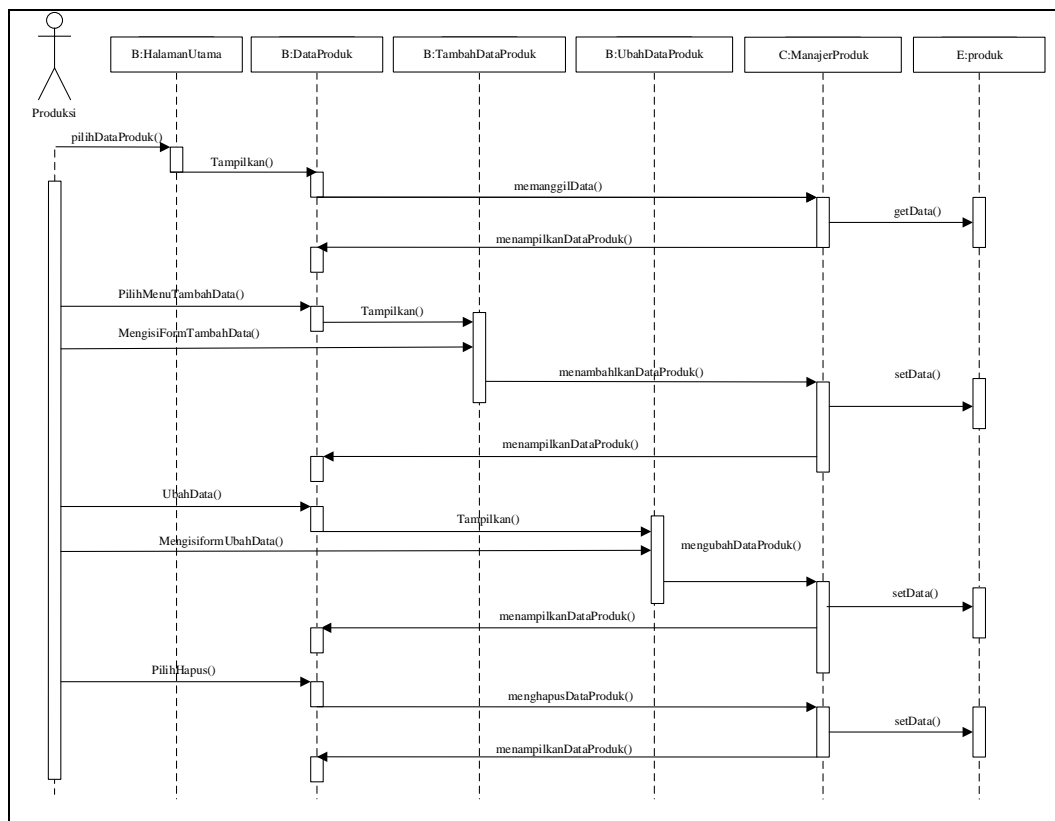


Gambar V.12 *Sequence Diagram* Melakukan Login

Sumber: Hasil Analisis (2019)

2. Sequence Diagram Mengelola Data Master Produk

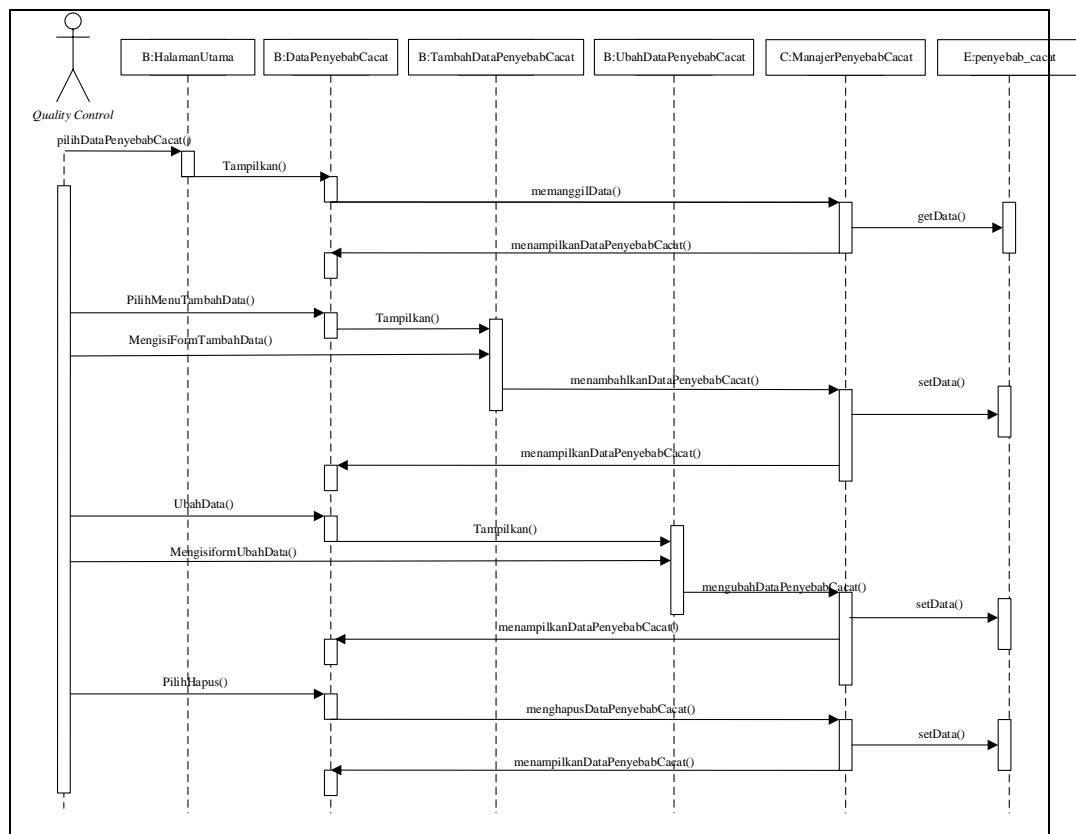
Sequence Diagram Mengelola Data Master Produk adalah jenis interaksi yang menggambarkan tentang proses Mengelola Data Master Produk, aktivitas yang dilakukan berupa menambah, mengubah dan menghapus data, berdasarkan *use case* *Sequence Diagram* Mengelola Data Master Produk, berikut adalah *Sequence Diagram* Mengelola Data Master Produk. Berikut Gambar V.13:



Gambar V.13 *Sequence Diagram* Mengelola Data Master Produk
Sumber: Hasil Analisis (2019)

3. *Sequence Diagram* Mengelola Data Master Penyebab Cacat

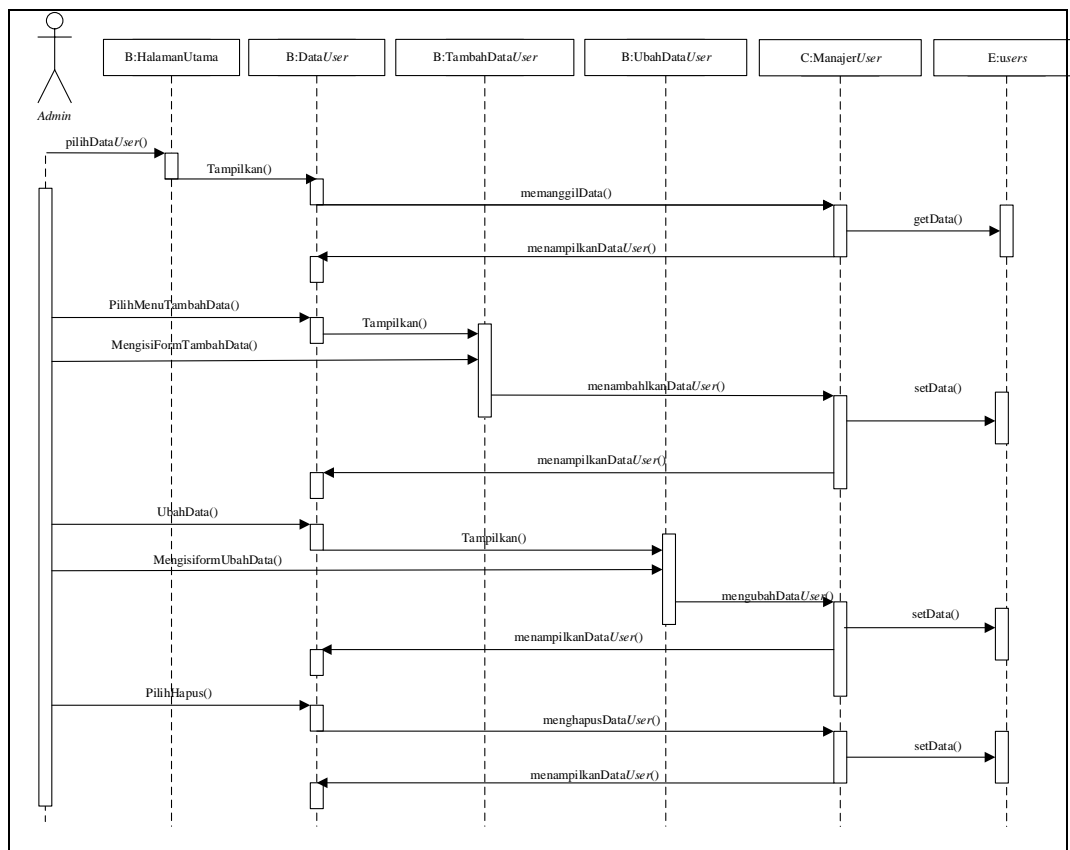
Sequence Diagram Mengelola Data Master Penyebab Cacat adalah jenis interaksi yang menggambarkan tentang proses Mengelola Data Master Penyebab Cacat, aktivitas yang dilakukan berupa menambah, mengubah dan menghapus data, berdasarkan *use case Sequence Diagram* Mengelola Data Master Penyebab Cacat, berikut adalah *Sequence Diagram Sequence Diagram* Mengelola Data Master Penyebab Cacat. Berikut Gambar V.14:



Gambar V.14 *Sequence Diagram* Mengelola Data Master Penyebab Cacat
Sumber: Hasil Analisis (2019)

4. *Sequence Diagram* Mengelola Data Master User

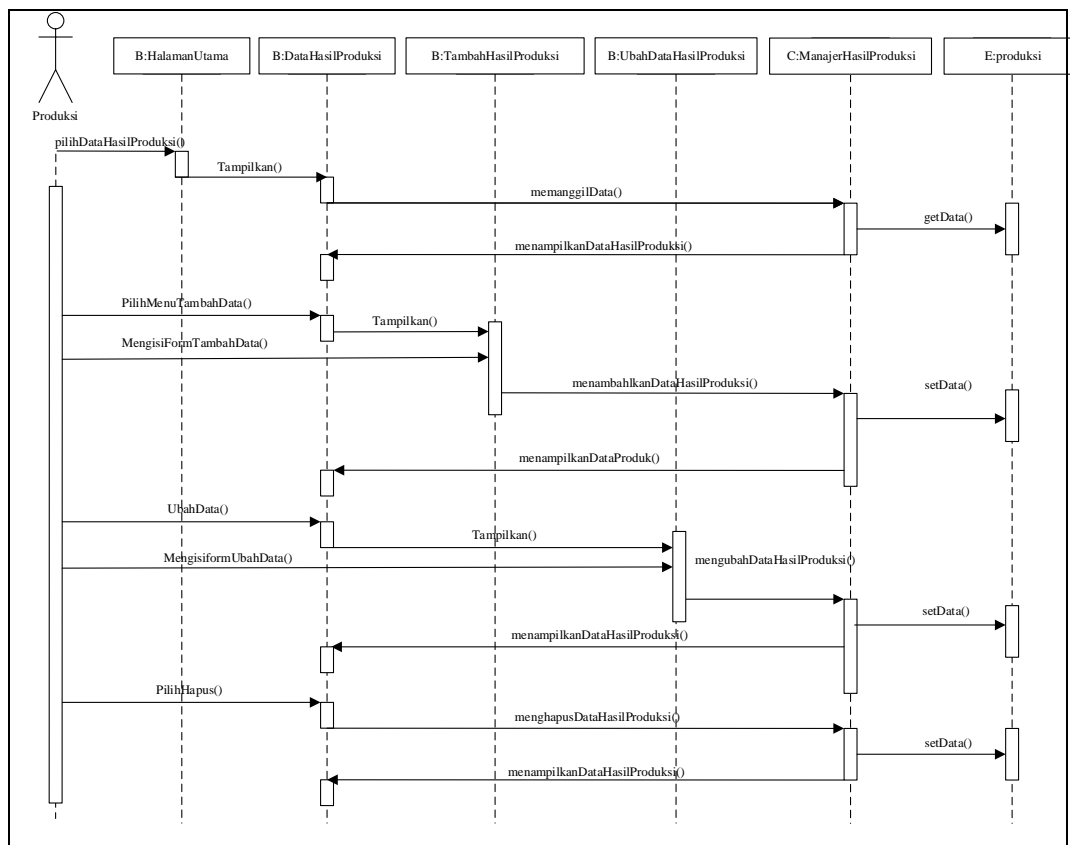
Sequence Diagram Mengelola Data Master User adalah jenis interaksi yang menggambarkan tentang proses Mengelola Data Master User, aktivitas yang dilakukan berupa menambah, mengubah dan menghapus data, berdasarkan *use case Sequence Diagram* Mengelola Data Master User, berikut adalah *Sequence Diagram* Mengelola Data User. Berikut Gambar V.15:



Gambar V.15 *Sequence Diagram* Mengelola Data Master User
 Sumber: Hasil Analisis (2019)

5. *Sequence Diagram* Mengelola Data Transaksi Hasil Produksi

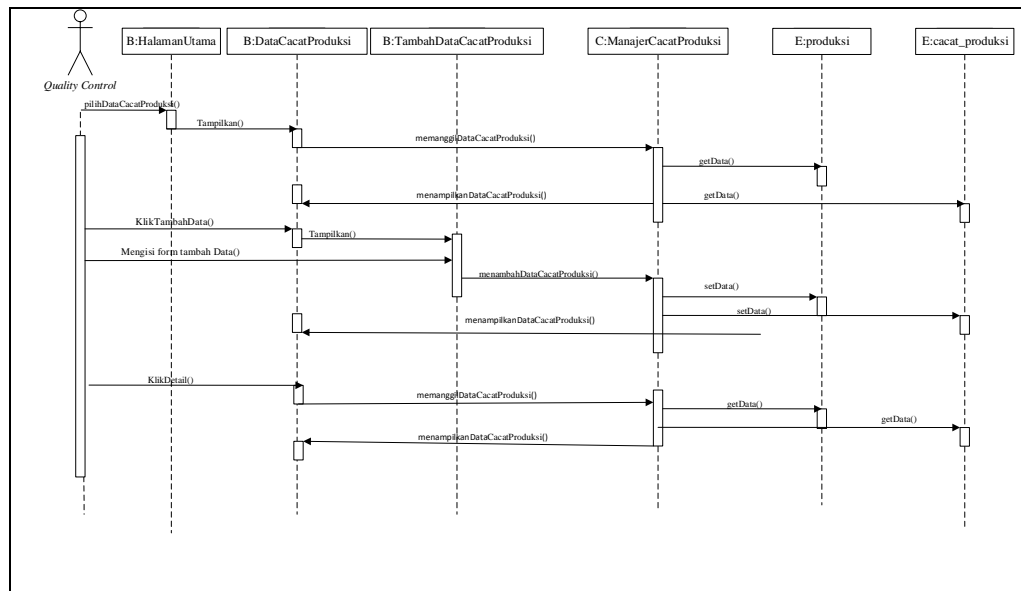
Sequence Diagram Mengelola Data Transaksi Hasil Produksi adalah jenis interaksi yang menggambarkan tentang proses Mengelola Data Transaksi Hasil Produksi, aktivitas yang dilakukan berupa menambah, mengubah dan menghapus data, berdasarkan *use case Sequence Diagram* Mengelola Data Data Transaksi Hasil Produksi, berikut adalah *Sequence Diagram Sequence Diagram* Mengelola Data Data Transaksi Hasil Produksi. Berikut Gambar V.16:



Gambar V.16 *Sequence Diagram* Mengelola Data Transaksi Hasil Produksi
Sumber: Hasil Analisis (2019)

6. *Sequence Diagram* Mengelola Data Transaksi Cacat Produksi

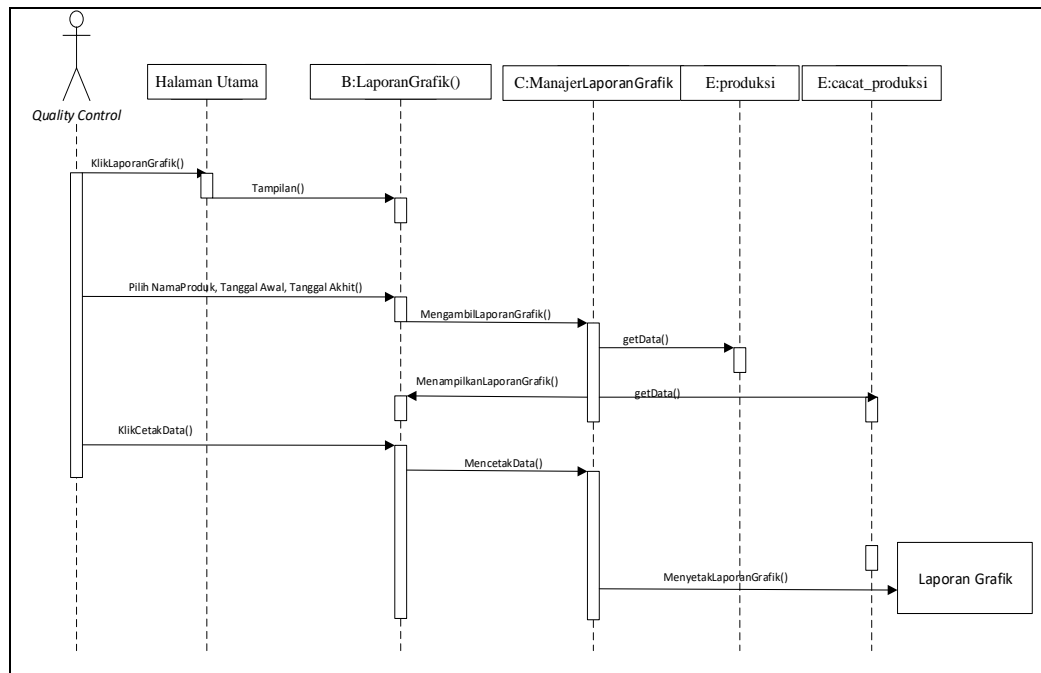
Sequence Diagram Mengelola Data Transaksi Cacat Produksi adalah jenis interaksi yang menggambarkan tentang proses Mengelola Data Transaksi Cacat Produksi, aktivitas yang dilakukan berupa *input*, dan *detail* data, berdasarkan *use case Sequence Diagram* Mengelola Data Transaksi Cacat Produksi, berikut adalah *Sequence Diagram Sequence Diagram* Mengelola Data Transaksi Cacat Produksi. Berikut Gambar V.17:



Gambar V.17 *Sequence Diagram* Mengelola Data Transaksi Cacat Produksi
Sumber: Hasil Analisis (2019)

7. *Sequence Diagram* Mencetak Laporan Grafik

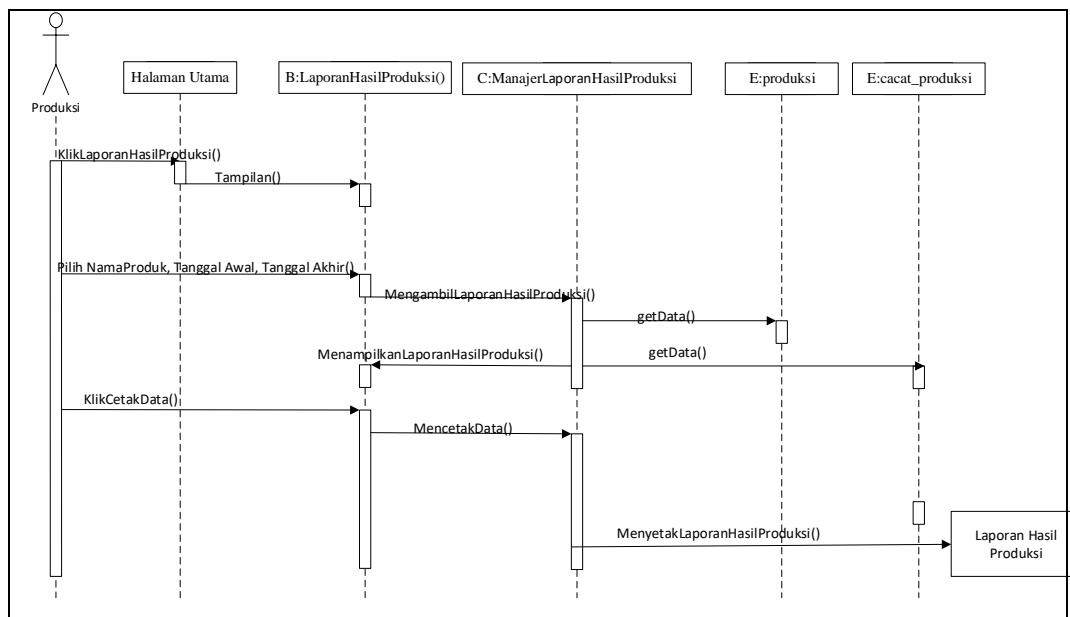
Sequence Diagram Mencetak Laporan Grafik adalah jenis interaksi yang menggambarkan tentang proses Mencetak Laporan, aktivitas yang dilakukan berupa menampilkan Laporan dalam bentuk grafik serta ada tombol untuk mencetak Laporan berdasarkan *use case Sequence Diagram* Mencetak Laporan, berikut adalah *Sequence Diagram Sequence Diagram* Mencetak Laporan. Berikut Gambar V.18:



Gambar V.18 *Sequence Diagram* Mencetak Laporan Grafik
 Sumber: Hasil Analisis (2019)

8. *Sequence Diagram* Mencetak Laporan Hasil Produksi

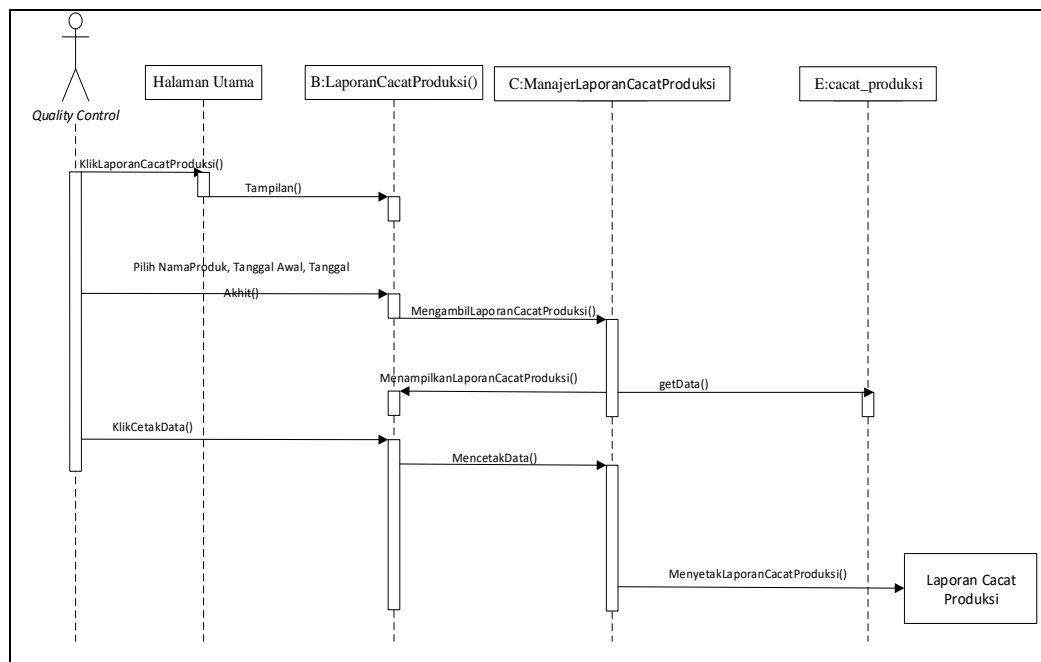
Sequence Diagram Mencetak Laporan Hasil Produksi adalah jenis interaksi yang menggambarkan tentang proses Mencetak Laporan Hasil Produksi, Hasil Produksi aktivitas yang dilakukan berupa menampilkan Laporan Hasil Produksi serta ada tombol untuk mencetak Laporan Hasil Produksi berdasarkan *use case Sequence Diagram* Mencetak Laporan Hasil Produksi, berikut adalah *Sequence Diagram Sequence Diagram* Mencetak Laporan Hasil Produksi. Berikut Gambar V.19:



Gambar V.19 *Sequence Diagram* Mencetak Laporan Hasil Produksi
 Sumber: Hasil Analisis (2019)

9. *Sequence Diagram* Mencetak Laporan Cacat Produksi

Sequence Diagram Mencetak Laporan Cacat Produksi adalah jenis interaksi yang menggambarkan tentang proses Mencetak Laporan Cacat Produksi, aktivitas yang dilakukan berupa menampilkan Laporan Cacat Produksi serta ada tombol untuk mencetak Laporan Cacat Produksi berdasarkan *use case* *Sequence Diagram* Mencetak Laporan Cacat Produksi, berikut adalah *Sequence Diagram* *Sequence Diagram* Mencetak Laporan Cacat Produksi. Berikut Gambar V.20:

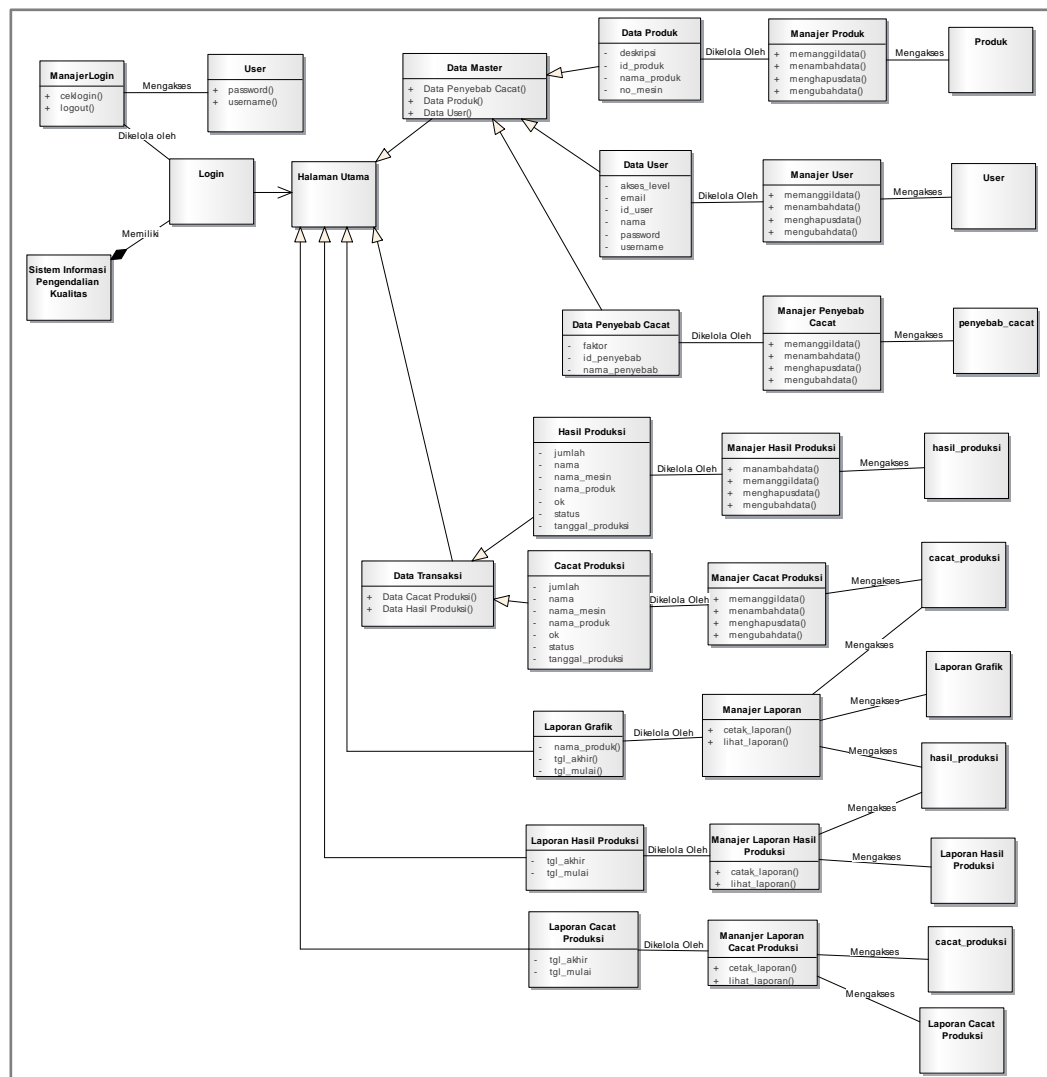


Gambar V.20 *Sequence Diagram* Mencetak Laporan Cacat Produksi

Sumber: Hasil Analisis (2019)

5.6 Class Diagram

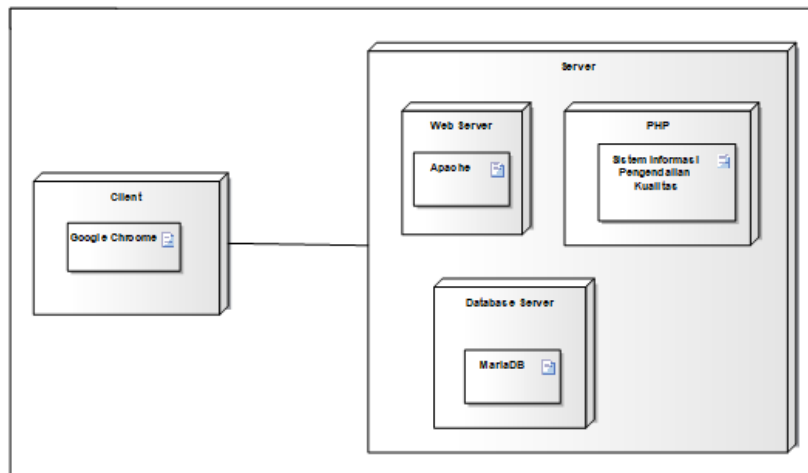
Sebuah diagram kelas adalah model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. Sebuah *class* merupakan deskripsi dari sekumpulan objek yang memiliki properti (*attribute*), operasi (*method*), relasi (*association*), dan tingkah laku (*behavior*) yang sama. Sebuah class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Berikut Gambar V.21:



Gambar V.21 Class Diagram Sistem Informasi Pengendalian Kualitas
Sumber: Hasil Analisis (2019)

5.7 Deployment Diagram

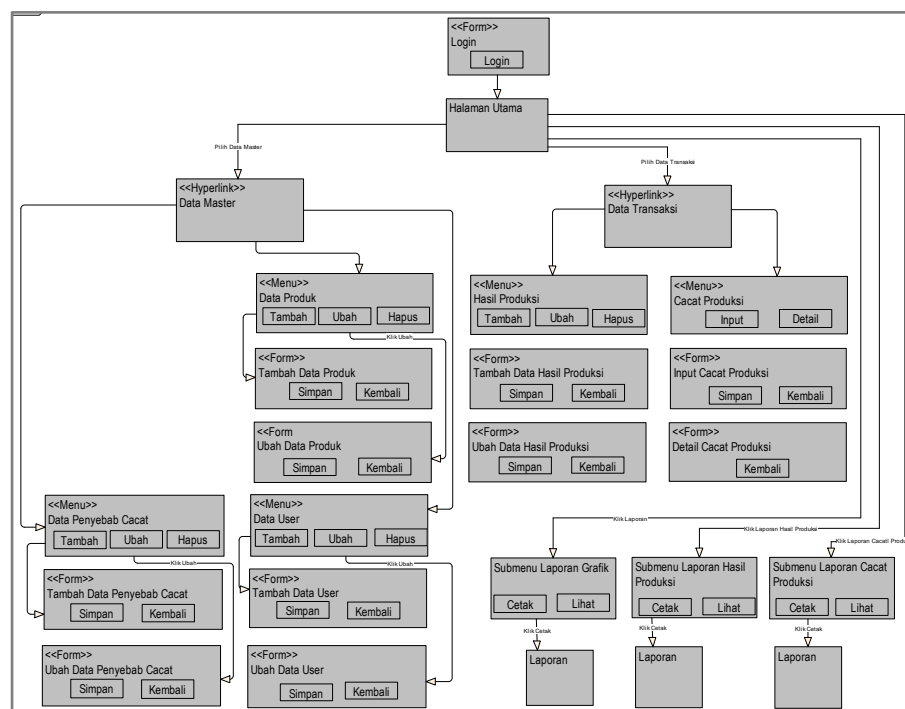
Deployment diagram pada usulan sistem informasi pengendalian kualitas digunakan untuk mewakili komponen-komponen *software* dan cara *software* ditempatkan pada arsitektur fisik atau infrastruktur sistem informasi, Berikut merupakan *Deployment Diagram* Sistem Informasi Pengendalian Kualitas. Berikut Adalah Gambar V.22:



Gambar V.22 *Deployment Diagram* Sistem Informasi Pengendalian Kualitas
Sumber: Hasil Analisis (2019)

5.8 Windows Navigation Diagram

Windows Navigation Diagram pada usulan sistem informasi pengendalian kualitas digunakan untuk menggambarkan perpindahan dari sebuah *window* ke *window* lain, beserta *interface* yang ada didalam sistem usulan pengendalian kualitas proses produksi, berikut adalah *Windows Navigation Diagram* usulan sistem pengendalian kualitas Seperti Pada Gambar V.23:



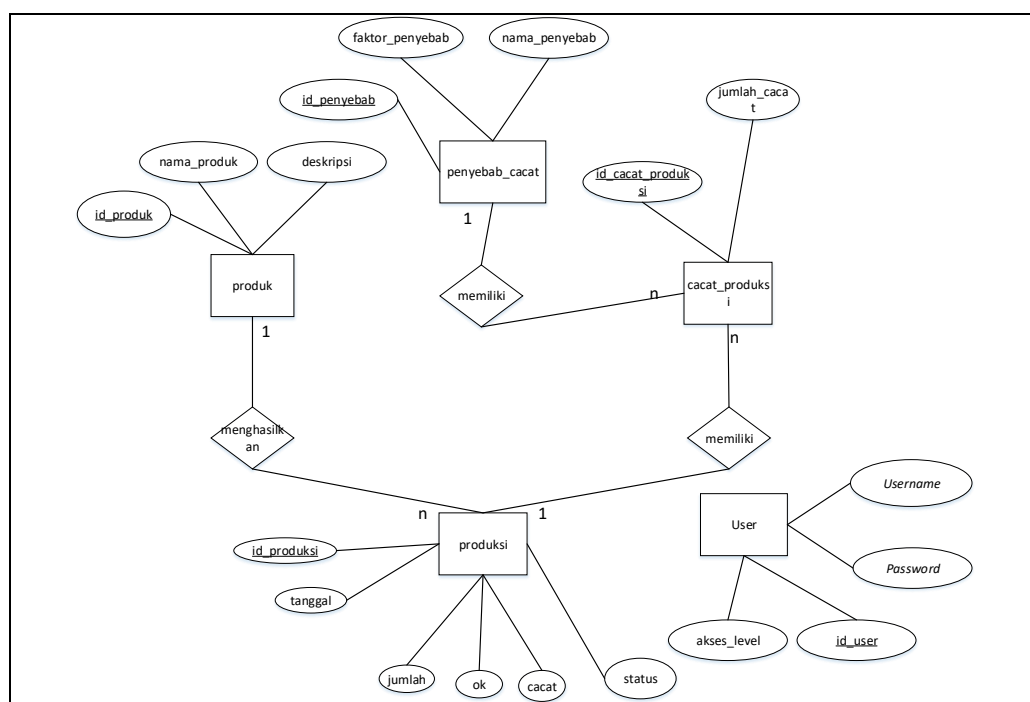
Gambar V.23 WND Sistem Informasi Pengendalian Kualitas
Sumber: Hasil Analisis (2019)

5.9 Pemodelan Data Sistem Usulan

Pemodelan data pada sistem informasi pengendalian kualitas menggunakan dua cara yaitu *Entity Relationship Diagram* (ERD) dan Kamus Data, ERD digunakan untuk menjelaskan hubungan antar data dalam basis data dan kamus data yang digunakan untuk menjelaskan isi dari *database* yang digunakan dalam sistem usulan pengendalian kualitas.

5.9.1 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram usulan sistem informasi pengendalian kualitas merupakan perancangan hubungan entitas antar tabel dari sebuah *database*, berikut adalah ERD usulan sistem informasi pengendalian kualitas:



Gambar V.24 ERD Sistem Informasi Pengendalian Kualitas

Sumber: Hasil Analisis (2019)

5.9.2 Kamus Data

Kamus data menggambarkan sebuah katalog data yang terdapat didalam sistem dengan maksud untuk mendefinisikan data yang ada di dalam sistem secara lengkap, sehingga *user* mampu mengerti tentang sistem yang dibuat oleh analis baik dari proses *penginputan* hingga mencetak *output*, berikut adalah kamus data dari sistem informasi pengendalian kualitas proses produksi.

1. Spesifikasi tabel user

- a. Nama tabel : users
- b. Fungsi : Menyimpan Data User
- c. Tipe : *File data master*

Tabel V.11 Spesifikasi Tabel User

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id User	id_user	Int	3	Primary key
2	Username	username	VarChar	24	
3	Password	password	VarChar	20	
4	Akses Level	akses_level	VarChar	30	

Sumber: Hasil Analisis (2019)

2. Spesifikasi tabel Produk

- a. Nama tabel : produk
- b. Fungsi : Untuk menyimpan data produk
- c. Tipe : *File data master*

Tabel V.12 Spesifikasi Tabel Produk

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id Produk	Id_produk	Int	3	Primary key
2	Nama Produk	nama_produk	VarChar	30	
3	Deskripsi	deskripsi	Text		

Sumber: Hasil Analisis (2019)

3. Spesifikasi tabel penyebab cacat

- a. Nama tabel : penyebab
- b. Fungsi : Untuk menyimpan data penyebab cacat
- c. Tipe : *File data master*

Tabel V.13 Spesifikasi Tabel Penyebab Cacat

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id Penyebab	id_penyebab	Int	2	Primary key
2	Faktor	faktor	VarChar	30	
3	Nama Penyebab	nama_penyebab	VarChar	50	

Sumber: Hasil Analisis (2019)

4. Spesifikasi tabel Produksi

- a. Nama tabel : produksi
- b. Fungsi : Untuk menyimpan data hasil produksi
- c. Tipe : *File* data transaksi

Tabel V.14 Spesifikasi Tabel Produksi

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id Produksi	id_penyebab	<i>Int</i>	5	<i>Primary key</i>
2	Tanggal Produksi	tanggal_produksi	<i>date</i>	10	
3	Id Produk	id_produk	<i>Int</i>	3	<i>Foreign key</i>
4	Jumlah	jumlah	<i>Int</i>	50	
5	Ok	ok	<i>Int</i>	11	
6	Cacat	cacat	<i>Int</i>	11	
7	Status	status	<i>Int</i>	11	
8	Id User	id_user	<i>Int</i>	11	

Sumber: Hasil Analisis (2019)

5. Spesifikasi tabel Cacat Produksi

- d. Nama tabel : cacat_produksi
- e. Fungsi : Untuk menyimpan data Cacat Produksi
- f. Tipe : *File* data transaksi

Tabel V.15 Spesifikasi Tabel Cacat Produksi

No.	Nama Elemen	Akronim	Tipe Data	Panjang	Keterangan
1	Id Cacat Produksi	id_cacat_produksi	<i>Int</i>	5	<i>Primary key</i>
2	Id Produksi	id_produksi	<i>Int</i>	5	<i>Foreign key</i>
3	Id Penyebab	id_penyebab	<i>Int</i>	2	<i>Foreign key</i>
4	Jumlah Cacat	jumlah_cacat	<i>Int</i>	4	

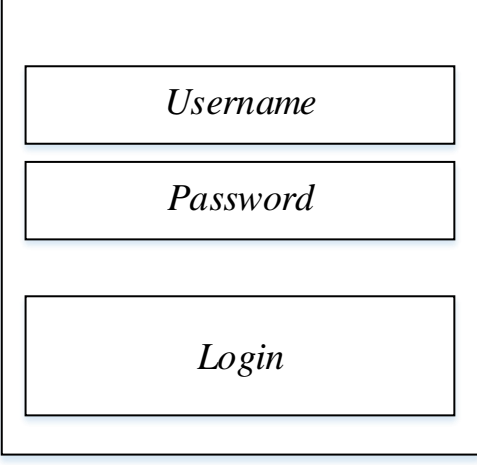
Sumber: Hasil Analisis (2019)

5.9.3 Perancangan Interface Sistem

Rancangan *interface* dari program pengendalian kualitas proses produksi ini sebagai berikut:

1. *Form Login*

Form login adalah *form* yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, karyawan harus memasukkan *username* dan *password* yang benar. Rancangan *form login* dapat dilihat pada gambar V.25.



The diagram illustrates a login form interface. It consists of a large rectangular container with a thin border. Inside this container, there are three smaller rectangular boxes stacked vertically. The top box is labeled 'Username', the middle box is labeled 'Password', and the bottom box is labeled 'Login'. Each box has a thin border and is centered within the container.

Gambar V.25 *Interface Login*
Sumber: Hasil Analisis (2019)

2. *Form Menu Utama*

Form menu utama adalah tampilan awal pada aplikasi yang berisi dari beberapa menu, seperti menu data *master*, data transaksi, menu laporan, menu laporan hasil produksi, dan menu laporan cacat produksi. Berikut adalah rancangan menu utama, Gambar V.26 berikut:

PT Nagasaki Component Parts		Logout
Halaman Utama	SELAMAT DATANG	
Data Master ▼		
Data Transaksi ▼		
Laporan Grafik		
Laporan Produksi		
Laporan Cacat Produksi		

Gambar V.26 *Interface* Menu Utama
Sumber: Hasil Analisis (2019)

3. *Form* Data Master Produk

Form data master produk adalah *form* yang digunakan untuk mengelola data master produk, Rancangan *interface* dari *form* data master produk dapat dilihat pada Gambar V.27 berikut.

PT Nagasaki Component Parts

Logout

Halaman Utama

Data Master ▼

Produk

Penyebab Cacat

User

Data Transaksi ▼

Laporan Grafik

Laporan Produksi

Laporan Cacat Produksi

Data Produk

Tambah

10 Records per page

Search :

Id Produk	Nama Produk	Deskripsi	Nomor Mesin	Aksi
				Ubah Hapus

Gambar V.27 *Interface* Data Master Produk
Sumber: Hasil Analisis (2019)

4. *Form* Data Master Penyebab Cacat

Form data master penyebab cacat adalah *form* yang digunakan untuk mengelola data *master* penyebab cacat, Rancangan *interface* dari *form* data *master* penyebab cacat dapat dilihat pada Gambar V.28 berikut.

PT Nagasaki Component Parts

Logout

Halaman Utama

Data Master ▼

Produk

Penyebab Cacat

User

Data Transaksi ▼

Laporan Grafik

Laporan Produksi

Laporan Cacat Produksi

Data Penyebab Cacat

Tambah

10

Records per page

Search :

Id Penyebab	Faktor	Nama Penyebab	Aksi
			<div>Ubah</div> <div>Hapus</div>

Gambar V.28 *Interface Data Master Penyebab Cacat*
Sumber: Hasil Analisis (2019)

5. *Form Data Master User*

Form data master user adalah *form* yang digunakan untuk mengelola data master *user*, Rancangan *interface* dari *form data master user* dapat dilihat pada Gambar V.28 berikut.

PT Nagasaki Component Parts

Logout

Halaman Utama

Data Master ▼

Produk

Penyebab Cacat

User

Data Transaksi ▼

Laporan Grafik

Laporan Produksi

Laporan Cacat Produksi

Data User

Tambah

10 Records per page

Search :

Id User	Nama	Email	Username	Password	Akses Level	Aksi
						Ubah Hapus

Gambar V.29 *Interface* Data Master user
Sumber: Hasil Analisis (2019)

6. *Form* Data Transaksi Hasil Produksi

Form data Transaksi Hasil Produksi adalah *form* yang digunakan untuk mengelola data Transaksi Hasil Produksi yang terjadi pada hari itu juga, Rancangan *interface* dari *form* data Transaksi Hasil Produksi dapat dilihat pada Gambar V.30 berikut.

PT Nagasaki Component Parts

Logout

Halaman Utama

Data Master ▼

Data Transaksi ▼

Hasil Produksi

Cacat Produksi

Laporan Grafik

Laporan Produksi

Laporan Cacat Produksi

Data Hasil Produksi

Tambah

10 Records per page

Search :

Tanggal Produksi	Nama Produk	Jumlah	Ok	Cacat	Operator	Status	Aksi
							Ubah Hapus

Gambar V.30 *Interface* Data Transaksi Hasil Produksi
Sumber: Hasil Analisis (2019)

7. *Form Data Transaksi Cacat Produksi*

Form data transaksi cacat produksi adalah *form* yang digunakan untuk mengelola data transaksi cacat produksi yang terjadi pada hari itu juga, Rancangan *interface* dari *form* data transaksi cacat produksi dapat dilihat pada Gambar V.31 berikut.

The screenshot shows a web application interface for PT Nagasaki Component Parts. The top header includes the company name and a 'Logout' button. A left sidebar contains a menu with items: 'Halaman Utama', 'Data Master', 'Data Transaksi', 'Hasil Produksi', 'Cacat Produksi' (highlighted), 'Laporan Grafik', 'Laporan Produksi', and 'Laporan Cacat Produksi'. The main content area is titled 'Data Cacat Produksi'. It features a 'Records per page' dropdown set to '10' and a 'Search' input field. Below these is a table with the following columns: 'Tanggal Produksi', 'Nama Produk', 'Jumlah', 'Ok', 'Cacat', 'Operator', 'Status', and 'Aksi'. The 'Aksi' column contains 'Input' and 'Detail' buttons. The table has three empty rows for data entry.

Gambar V.31 *Interface* Data Transaksi Cacat Produksi
Sumber: Hasil Analisis (2019)

8. *Form Laporan Grafik*

Form Laporan Grafik adalah form yang digunakan untuk membuat Laporan yang telah di *inputkan*, Rancangan *interface* dari *form* Laporan dapat dilihat pada Gambar V.32 berikut.

The screenshot shows the 'Form Laporan Grafik' interface. It has the same header and sidebar as the previous form. The main content area contains several input fields: 'Nama Produk' (text), a dropdown menu, 'Tanggal Mulai' (text with 'mm/dd/yyyy' placeholder), 'Tanggal Akhir' (text with 'mm/dd/yyyy' placeholder), and a 'Lihat' button.

Gambar V.32 *Interface* Laporan Grafik
Sumber: Hasil Analisis (2019)

9. *Form* Laporan Hasil Produksi

Form laporan hasil produksi adalah *form* yang digunakan untuk membuat laporan hasil produksi yang telah di *inputkan*, Rancangan *interface* dari *form* laporan hasil produksi dapat dilihat pada Gambar V.33 berikut.

PT Nagasaki Component Parts		Logout
Halaman Utama	Form Laporan Produksi	
Data Master ▼	<input type="text" value="Nama Produk"/> <input type="text" value="▼"/>	
Data Transaksi ▼	<input type="text" value="Tanggal Mulai"/> <input type="text" value="mm/dd/yyyy"/>	
Laporan Grafik	<input type="text" value="Tanggal Akhir"/> <input type="text" value="mm/dd/yyyy"/>	
Laporan Produksi	<input type="button" value="Lihat"/>	
Laporan Cacat Produksi		

Gambar V.33 *Interface* Laporan Hasil Produksi
Sumber: Hasil Analisis (2019)

10. *Form* Laporan Cacat Produksi

Form laporan cacat produksi adalah *form* yang digunakan untuk membuat laporan cacat produksi yang telah di *inputkan*, Rancangan *interface* dari *form* laporan cacat produksi dapat dilihat pada Gambar V.34 berikut.

PT Nagasaki Component Parts		Logout
Halaman Utama	Form Laporan Cacat Produksi	
Data Master ▼	<input type="text" value="Nama Produk"/> <input type="text" value="▼"/>	
Data Transaksi ▼	<input type="text" value="Tanggal Mulai"/> <input type="text" value="mm/dd/yyyy"/>	
Laporan Grafik	<input type="text" value="Tanggal Akhir"/> <input type="text" value="mm/dd/yyyy"/>	
Laporan Produksi		
Laporan Cacat Produksi	<input type="button" value="Lihat"/>	

Gambar V.34 *Interface* Laporan Cacat Produksi
Sumber: Hasil Analisis (2019)

5.10 Contoh Kasus SPC

Dalam PT Nagasakti Component Parts disini menjelaskan terjadinya kecacatan pada produk K 81, berikut deskripsi hal apa saja yang membuat produk K 81 bisa terjadi cacat:

Tabel V.16 Contoh Penyebab Cacat PT Nagasakti Component Parts

Penyebab Cacat	Alasan
Mesin	Suhu tidak Sesuai
Material	Jamur
Manusia	Kurang teliti
Manusia	Ceroboh
Metode	Alat yang digunakan tidak tajam

Sumber: Hasil Analisis (2019)

Berikut banyaknya kecacatan produk dari tanggal 1 - 30 juli 2018:

Tabel V.17 Contoh Jumlah Penyebab Cacat PT Nagasakti Component Parts

No	Penyebab	Alasan	Jumlah
1	Material	Jamur	5
2	Mesin	Suhu Tidak Sesuai	10
3	Metode	Alat yang digunakan tidak tajam	10
4	Manusia	Ceroboh	7
5	Mesin	Suhu Tidak Sesuai	9
6	Mesin	Suhu Tidak Sesuai	11
7	Manusia	Kurang Teliti	3
8	Mesin	Suhu Tidak Sesuai	15
9	Metode	Alat yang digunakan tidak tajam	4
10	Manusia	Ceroboh	9
11	Manusia	Terburu Buru	9
12	Mesin	Suhu Tidak Sesuai	3
13	Metode	Alat yang digunakan tidak tajam	5
14	Mesin	Suhu Tidak Sesuai	9
15	Metode	Alat yang digunakan tidak tajam	8
16	Manusia	Terburu Buru	7
17	Mesin	Suhu Tidak Sesuai	12
18	Material	Jamur	3
19	Metode	Alat yang digunakan tidak tajam	5
20	Material	Jamur	7
21	Manusia	Terburu Buru	8
22	Metode	Alat yang digunakan tidak tajam	9
23	Mesin	Suhu Tidak Sesuai	10

Tabel V.17 Contoh Jumlah Penyebab Cacat PT Nagasaki Component Parts (Lanjutan)

No	Penyebab	Alasan	Jumlah
24	Mesin	Terburu Buru	4
25	Manusia	Jamur	20
26	Material	Terburu Buru	3
27	Manusia	Jamur	15
28	Material	Suhu Tidak Sesuai	9
29	Mesin	Suhu Tidak Sesuai	3
30	Mesin	Jamur	10
31	Metode	Alat yang digunakan tidak tajam	15
32	Material	Jamur	22
33	Material	Jamur	11
34	Metode	Alat yang digunakan tidak tajam	4
35	Material	Jamur	9
36	Manusia	Ceroboh	3
37	Manusia	Terburu Buru	4
38	Material	Jamur	11
39	Manusia	Terburu Buru	2
40	Material	Jamur	4
41	Mesin	Suhu Tidak Sesuai	1
42	Manusia	Terburu Buru	9
43	Material	Jamur	4
44	Metode	Alat yang digunakan tidak tajam	12
45	Metode	Alat yang digunakan tidak tajam	4
46	Mesin	Suhu Tidak Sesuai	3

Sumber: Hasil Analisis (2019)

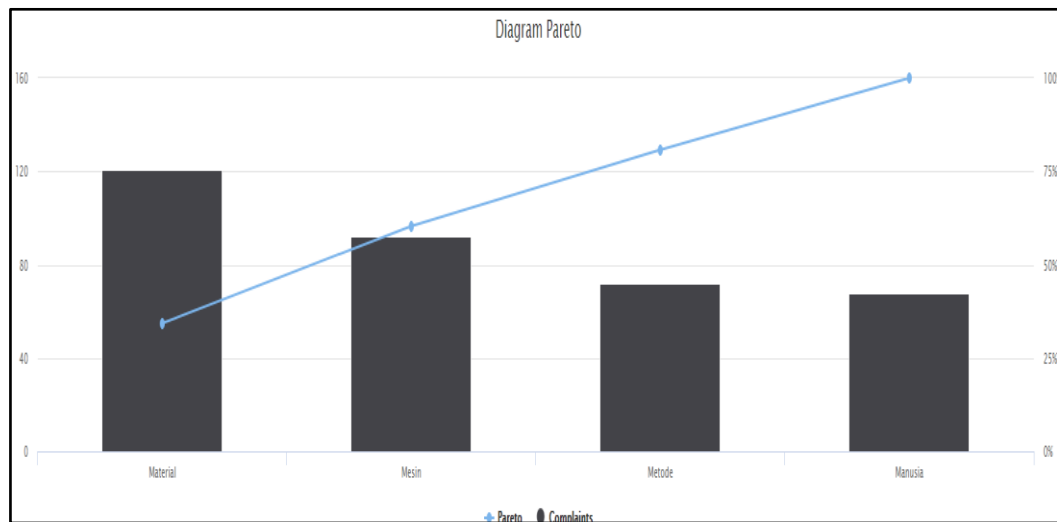
Berikut hasil presentase dan jumlah yang disebabkan oleh metode, mesin, material, manusia:

Tabel V.18 Contoh Jumlah Penyebab Cacat Yang Sudah Dikategorikan PT Nagasaki Component Parts

Sebab	Jumlah	Jumlah akumulasi	Presentase	Presentase akumulasi
Material	121	121	33.61	33.61
Mesin	95	216	26.39	60
Metode	76	292	21.11	81.1
Manusia	68	360	18.89	100
	360		100	

Sumber: Hasil Analisis (2019)

Berikut gambar yang dihasilkan dari diagram pareto yang telah dibuat:



Gambar V.35 Laporan Grafik Diagram Pareto
Sumber: Hasil Analisis (2019)

Deskripsi:

- Sumbu y yang berada disebelah kiri menjelaskan bahwa penyebab cacat disebelah kira berjumlah.
- Sumbu y sebelah kanan menjelaskan hasil akumulasi presentase
- Sumbu x yang berada dibawah mejelaskan penyebab terjadinya kecacatan produk ada 4 yaitu, material, mesin, metode, manusia

Berikut ini membuat dan cara menghitung proporsi dan hasil CL, UCL, dan LCL untuk jumlah sampel Bervariasi kasus perusahaan PT Nagasaki Component Parts, pada perusahaan *rubber* ditemukan cacat produk dalam sampel yang bervariasi setiap kali melakukan observasi. Adapun sampel yang diambil dan kesalahan yang terjadi sebagai berikut:

Tabel V.19 Perhitungan Peta Kendali PT Nagasaki Component Parts

Tanggal	Jumlah Produksi	Jumlah Cacat	Proporsi (p)	UCL	LCL
7/3/2018	224	15	0.067	0.139	0.028
7/3/2018	250	10	0.040	0.136	0.031
7/4/2018	200	16	0.080	0.142	0.025
7/5/2018	260	14	0.054	0.135	0.032
7/6/2018	180	15	0.083	0.145	0.022
7/7/2018	150	13	0.087	0.151	0.016

Tabel V.19 Perhitungan Peta Kendali PT Nagasaki Component Parts (Lanjutan)

Tanggal	Jumlah Produksi	Jumlah Cacat	Proporsi (p)	UCL	LCL
7/9/2018	300	17	0.057	0.131	0.036
7/10/2018	350	9	0.026	0.128	0.039
7/11/2018	166	15	0.090	0.148	0.019
7/12/2018	156	12	0.077	0.150	0.017
7/13/2018	100	8	0.080	0.166	0.001
7/14/2018	154	15	0.097	0.150	0.017
7/16/2018	120	19	0.158	0.159	0.008
7/17/2018	180	4	0.022	0.145	0.022
7/18/2018	132	23	0.174	0.156	0.011
7/19/2018	100	24	0.240	0.166	0.001
7/20/2018	140	13	0.093	0.154	0.013
7/21/2018	123	37	0.301	0.158	0.009
7/23/2018	144	15	0.104	0.153	0.014
7/24/2018	155	9	0.058	0.150	0.017
7/25/2018	156	7	0.045	0.150	0.017
7/26/2018	149	13	0.087	0.151	0.016
7/27/2018	111	5	0.045	0.162	0.005
7/28/2018	135	25	0.185	0.155	0.012
7/30/2018	177	7	0.040	0.146	0.021
Jumlah	4312	360			

Sumber: Hasil Analisis (2019)

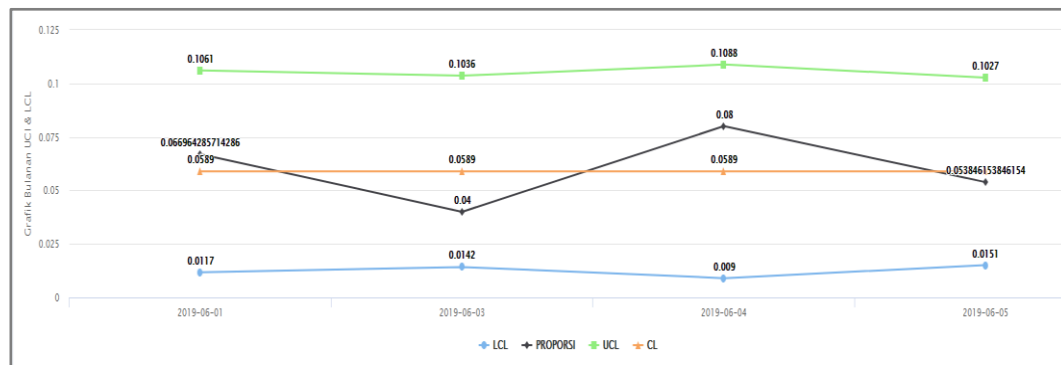
$$p = \frac{224}{15} = 0,067$$

$$CL = \frac{4312}{360} = 0.083$$

$$UCL = 0,083 + 3 \frac{\sqrt{0,083(1 - 0,083)}}{224} = 0,139$$

$$LCL = 0,083 - 3 \frac{\sqrt{0,083(1 - 0,083)}}{224} = 0,028$$

Berikut ini *output*/diagram yang dihasil untuk membuat peta kendali:



Gambar V.36 Laporan Grafik Peta Kendali
Sumber: Hasil Analisis (2019)

Deskripsi:

- Sumbu x yang berada dibawah menjelaskan tanggal diproduksinya K 81
- Sumbu y yang berada dikiri menjelaskan angka hasil perhitungan CL, UCL, LCL.
- Garis hijau itu hasil perhitungan UCL
- Garis orange itu hasil perhitungan CL
- Garis biru itu hasil perhitungan LCL
- Garis hitam itu hasil perhitungan proporsi
- Jika pernhitungan proporsi diantara garis UCL dan LCL dinyatakan variasi akibat sebab normal, jika perhitungan proporsi diluar garis UCL dan LCL dinyatakan variasi akibat sebab khusus.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

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

1. Dengan adanya sistem aplikasi berbasis komputer pada proses pengelolaan cacat produksi dan hasil produksi, dapat meminimalisir terjadinya penumpukan dan kehilangan data cacat produksi dan hasil produksi. Dengan adanya *database* dapat mengurangi terjadinya kehilangan data dan mempercepat kinerja karyawan dalam hal pencarian data. Dengan adanya sistem aplikasi pengendalian kualitas proses produksi dengan *output diagram* dapat membantu untuk dijadikan acuan pengambilan keputusan dan menentukan penyebab utama terjadinya cacat produksi.
2. Penggunaan metode *Statistical Process Control* (SPC) dalam sistem aplikasi pengendalian kualitas dapat memberikan informasi yang bertujuan untuk membantu mengurangi terjadinya cacat produksi.

6.2. Saran

Saran yang diberikan untuk pengembangan sistem informasi pengendalian kualitas produk ini selanjutnya adalah Sebagai berikut:

1. Sistem ini dapat diimplementasikan pada proses pengendalian kualitas di bagian *Quality Control* pada PT Nagasakti Component Parts.
2. Untuk penelitian selanjutnya, sistem informasi ini diharapkan dapat diintegrasikan dengan bagian pegawai.

DAFTAR PUSTAKA

- Arifianti, R. (2013). Analisis Kualitas Produk Sepatu Tomkins. *Jurnal Dinamika Manajemen.*, 46-58.
- Assauri, S. (2008). *Manajemen Operasi Dan Produksi*. Jakarta: LP FE UI.
- Ayuni, D., Siswandaru, K., & Nupikso, G. (2012). Analisis Penerapan Statistical Quality Control Pada Beban Usaha PT. PLN. *Jurnal Organisasi dan Manajemen*, 8, 22-31.
- Cahyawati, A. N., & Septiasari, S. (2019). Analisis Penyebab Defect Pada Produksi Rumah Lampu Dengan Metode Statistical Process Control. *ISSN 2085-4218*, 257-262.
- Dennis, A., Wixom, B. H., & Tegarden, D. (2010). *System Analysis and Design with UML 2.0*. New York: John Willey and Sons, Inc.
- Dennis, A., Wixom, B. H., & Tegarden, D. (2015). *System Analysis & Design. An Object-Oriented Approach with UML (Fifth Edition)*. USA: Wiley.
- Enterprise, J. (2015). *Membuat Website PHP dengan Codeigniter*. Jakarta: Elex Media Komputindo.
- Fathansyah. (2015). *Basis Data*. Bandung: Informatika Bandung.
- Gaspersz, V. (2005). *Total Quality Management*. Jakarta: Gramedia Pusaka Utama.
- Haizer, J., & Render, B. (2006). *Operations Management (Manajemen Operasi)*. Jakarta: Salemba Empat.
- Harsanto, B. (2013). *Dasar Ilmu Manajemen Operasi*. Sumedang: Unpad Press.
- Hartono, J. (2010). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Hoffer, J. A., George, J. F., & Valcich, J. S. (2007). *Modern System Analysis and Design*. Edisi Keempat. India: Pearson Education.

- Hutahaean, J. (2015). *Konsep Sistem Informasi*. Yogyakarta: Deepublish .
- Kartika, H. (2013). Analisis Pengendalian Kualitas Produk CPE Film Dengan Metode Statistical Process Control Pada PT MSI. *Jurnal Ilmiah Teknik Industri*, 50-58.
- Ladjmudin, A.-B. b. (2005). *Analisa Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Montgomery, D. C. (2009). *Introduction to Statistical Quality Control*. USA: John Wiley & Sons, Inc.
- OMG. (2015). *Unified Modeling Language*. Needham: Object Management Group.
- Pressman, R. S. (2010). *Software Engineering : a practitioner's approach 7th ed.USA*. New York: McGraw-Hil.
- Ragil, W. (2010). *Pedoman Sosialisai Prosedur Operasi Standar*. Jakarta: Mitra.
- Robbins, S. P., & Coulter, M. (2007). *Manajemen*. Edisi Kedelapan. Jakarta: PT Indeks.
- Rusdiana. (2014). *Sistem Informasi Manajemen*. Bandung: CV Pustaka Setia.
- Russell, R. S., & Taylor, B. W. (2011). *Operations and Supply Chain Management*. New York: John Wiley & Sons Inc.
- Sarosa, S. (2017). *Meotologi Pengembangan Sistem Informasi*. Jakarta: Indeks.
- Setyorini. (2016). *Konsep Dasar Sistem Informasi Manajemen*. Yogyakarta: Deepublish.
- Soemohadiwidjojo, A. T. (2017). *Six Sigma: Metode Pengukuran Kinerja Perusahaan Berbasis Statistik*. Jakarta: Raih Asa Sukses.
- Subagia, A. (2018). *Kolaborasi Codeigniter dan Ajax dalam Perancangan CMS*. Jakarta: PT Elex Media Komputindo.
- Sukamto, R. A., & Shalahuddin, M. (2015). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

- Sultana, F., Razive, N. I., & Azeem, A. (2009). Implementation Of Statistical Process Control (SPC) For Manufacturing Performance Improveent. *Journal of Mechanical Engineering*, 40, 15-21.
- Supono, & Putratama, V. (2016). *Pemograman Web Dengan Menggunakan PHP dan Framework Codeigniter*. Yogyakarta: Deepublish.
- Sutabri, T. (2012). *Analisis Sistem Informasi*. Yogyakarta: Andi.
- Sutaji, D. (2012). *Sistem Inventory mini market dengan PHP dan JQuery*. Yogyakarta: Lokomedia.
- Utami, E. (2008). *RDBMS Menggunakan MS SQL SERVER 2000*. Yogyakarta: Graha Ilmu.
- Yakub. (2012). *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Yamit, Z. (2013). *Manajemen Kualitas Produk dan Jasa*. Jakarta: Ekosinia.

Lampiran

Lampiran A

Coding

1. Auth.php

```
<?php

class Auth extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();
        $this->load->library('form_validation');
        $this->load->model('M_user');
    }

    public function index()
    {

        // jika button di klik (yg ada di view)
        if(isset($_POST['submit'])){
            //proses login disini
            $username    =    $this->input-
>post('username');
            $password    =    $this->input-
>post('password');
            $hasil        =    $this->M_user-
>login($username,$password);
```



```

        $jabatan      =      $this->db-
>get_where('users',array('username'=>$username))->row_array();
        if($hasil==1)
        {
            $this->session->set_userdata($jabatan);
            // $this->session-
>set_userdata(array('status_login'=>'oke','username'=>$username,'jabatan'
=>$jabatan));

            redirect ('home');
        }
        else{
            $this->session->set_flashdata('message',
'<div class="alert alert-danger" role="alert">
            Kombinasi Password dan Username Salah
<button type="button" class="close" data-dismiss="alert" aria-
label="Close">
            <span aria-hidden="true">&times;</span>
            </button>
            </div>');
            // die;
            redirect('auth/login');
        }
    }else{
        $this->load->view('form_login');
    }
}

public function logout()
{
    $this->session->unset_userdata('username');

```

```

        $this->session->unset_userdata('role_id');

        $this->session->set_flashdata('message', '<div class="alert
alert-danger" role="alert"> Anda Telah Keluar dari Aplikasi PT Nagasakti
Component Parts</div>');
        redirect('Auth');
    }
}

```

2. Produk.php

```

<?php

class Produk extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('M_produk');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data['record'] = $this->M_produk->tampilkan_data();
        $this->template->load('template', 'produk/lihat_data',
$data);
    }

    public function tambah()
    {
        $this->template->load('template', 'produk/form_input');
    }
}

```

```

public function simpan()
{
    $nama_produk= $this->input->post('nama_produk');
    $deskripsi      = $this->input->post('deskripsi');

    $data = array('nama_produk' => $nama_produk, 'deskripsi'
=> $deskripsi);
    $simpan = $this->M_produk->simpan($data);
    redirect('Produk');
}

public function hapus($id)
{
    $id = $this->uri->segment(3);
    $this->M_produk->hapus($id);
    redirect('Produk');
}

public function ubah()
{
    $id = $this->uri->segment(3);
    $data['record'] = $this->M_produk->get_data($id)-
>row_array();
    $this->template->load('template', 'produk/form_edit',
$data);
}

public function edit()
{

```

```

        $id = $this->input->post('id_produk');

        $nama_produk= $this->input->post('nama_produk');
        $deskripsi = $this->input->post('deskripsi');

        $data = array('nama_produk' => $nama_produk,'deskripsi'
=> $deskripsi);
        $simpan = $this->M_produk->update($data, $id);
        redirect('Produk');
    }
}

```

3. User.php

```

<?php

class User extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('M_user');
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data['record'] = $this->M_user->tampilkan_data();
        $this->template->load('template', 'user/lihat_data' , $data);
    }

    public function tambah()
    {

```

```

        $this->template->load('template', 'user/form_input');
    }

    public function simpan()
    {
        $nama            = $this->input->post('nama');
        $email            = $this->input->post('email');
        $username         = $this->input->post('username');
        $password         = $this->input->post('password');
        $akses_level     = $this->input->post('akses_level');

        $data = array('nama' => $nama, 'email' => $email,
'username' => $username, 'password' => $password, 'akses_level' =>
$akses_level);

        $simpan = $this->M_user->simpan($data);
        redirect('user');
    }

    public function hapus($id)
    {
        $id = $this->uri->segment(3);
        $this->M_user->hapus($id);
        redirect('user');
    }

    public function ubah()
    {
        $id = $this->uri->segment(3);
        $data['record'] = $this->M_user->get_data($id)-
>row_array();

```

```

        $this->template->load('template', 'user/form_edit', $data);
    }

    public function edit()
    {
        $id = $this->input->post('id');
        $nama = $this->input->post('nama');
        $email = $this->input->post('email');
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $akses_level = $this->input->post('akses_level');

        $data = array('nama' => $nama, 'email' => $email,
        'username' => $username, 'password' => $password, 'akses_level' =>
        $akses_level);

        $simpan = $this->M_user->update($data, $id);
        redirect('user');
    }
}

```

4. Produksi.php

```

<?php

class Produksi extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model(array('M_produk', 'M_produk'));
        $this->load->library('form_validation');
    }
}

```

```

public function index()
{
    $data['record'] = $this->M_produksi->tampilkan_data();
    $this->template->load('template', 'produksi/lihat_data',
$data);
}

public function tambah()
{
    $data['produk'] = $this->M_produk->tampilkan_data();
    $this->template->load('template', 'produksi/form_input',
$data);
}

public function simpan()
{
    $tanggal_produksi = $this->input-
>post('tanggal_produksi');
    $id_produk = $this->input->post('id_produk');
    $jumlah = $this->input->post('jumlah');
    $id_user = $this->session->userdata('id_user');

    $data = array('tanggal_produksi' => $tanggal_produksi,
'id_produk' => $id_produk, 'jumlah' => $jumlah, 'id_user' => $id_user);
    $simpan = $this->M_produksi->simpan($data);
    redirect('Produksi');
}

public function hapus($id)
{

```

```

        $id_produk = $this->uri->segment(3);
        $this->M_produksi->hapus($id);
        redirect('Produksi');
    }

    public function ubah()
    {
        $id = $this->uri->segment(3);
        $data['produk'] = $this->M_produk->tampilkan_data();
        $data['record'] = $this->M_produksi->get_data($id)-
>row_array();
        $this->template->load('template', 'produksi/form_edit',
        $data);
    }

    public function edit()
    {
        $id      = $this->input->post('id_produksi');
        $tanggal_produksi    = $this->input-
>post('tanggal_produksi');
        $id_produk    = $this->input->post('id_produk');
        $jumlah        = $this->input->post('jumlah');

        $data = array('tanggal_produksi' =>
        $tanggal_produksi, 'id_produk' => $id_produk, 'jumlah' => $jumlah);
        $simpan = $this->M_produksi->update($data, $id);
        redirect('Produksi');
    }
}

```


5. Qcontrol.php

```
<?php

class Qcontrol extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();

        $this->load->model(array('M_produk', 'M_mesin',
'M_produksi','M_qcontrol','M_penyebab'));
        $this->load->library('form_validation');
    }

    public function index()
    {
        $data['record'] = $this->M_produksi->tampilkan_data();
        $this->template->load('template',
'quality_control/lihat_data', $data);
    }

    public function tambah($id_produk)
    {
        $data['penyebab'] = $this->M_penyebab-
>tampilkan_data();
        $data['record'] = $this->M_qcontrol-
>tampilkan_data_detail($id_produk)->result();

        $data['produksi'] = $this->M_qcontrol-
>get_data_produk_by_id($id_produk)->row_array();
        $this->template->load('template',
'quality_control/form_input', $data);
    }
}
```

```

    }

    public function detail($id_produksi){

        $data['penyebab'] = $this->M_penyebab-
>tampilkan_data();

        $data['record'] = $this->M_qcontrol-
>tampilkan_data_detail($id_produksi)->result();

        $data['produksi'] = $this->M_qcontrol-
>get_data_produksi_by_id($id_produksi)->row_array();

        $this->template->load('template',
'quality_control/lihat_data_detail', $data);

    }

    public function simpan()
    {

        $id_produksi = $this->input->post('id_produksi');
        $id_penyebab = $this->input->post('id_penyebab');
        $jumlah_cacat = $this->input->post('jumlah_cacat');
        $data = array('id_produksi' => $id_produksi, 'id_penyebab'
=> $id_penyebab, 'jumlah_cacat' => $jumlah_cacat);
        $simpan = $this->M_qcontrol->simpan($data);

        $produksi = $this->M_qcontrol-
>get_data_produksi_by_id($id_produksi)->row_array();

        $cacat = $this->M_qcontrol-
>get_total_cacat($id_produksi)->row_array();

```

```

        $ok = $produksi['jumlah'] - $cacat['total_cacat'];

        $data = array('ok' => $ok, 'cacat' => $cacat['total_cacat']);
        $simpan = $this->M_produk->update($data,
        $id_produk);

        redirect('Qcontrol/tambah/'.$id_produk);
    }

    public function hapus($id_produk)
    {
        $id_produk = $this->uri->segment(3);
        $id_cacat_produk = $this->uri->segment(4);
        $this->M_qcontrol->hapus($id_cacat_produk);

        $produksi = $this->M_qcontrol->
        >get_data_produk_by_id($id_produk)->row_array();

        $cacat = $this->M_qcontrol->
        >get_total_cacat($id_produk)->row_array();

        $ok = $produksi['jumlah'] - $cacat['total_cacat'];

        $data = array('ok' => $ok, 'cacat' => $cacat['total_cacat']);
        $simpan = $this->M_produk->update($data,
        $id_produk);

        redirect('Qcontrol/tambah/'.$id_produk);
    }

```

```

public function selesai($id_produk){
    $id_produk = $this->uri->segment(3);
    $data = array('status' => '1');
    $simpan = $this->M_produk->update($data,
$id_produk);
    redirect('Qcontrol');
}

```

```

public function ubah()
{
    $id = $this->uri->segment(3);
    $data['produk'] = $this->M_produk->tampilkan_data();
    $data['mesin'] = $this->M_mesin->tampilkan_data();
    $data['record'] = $this->M_produk->get_data($id)-
>row_array();
    $this->template->load('template',
'quality_control/form_edit', $data);
}

```

```

public function edit()
{
    $id      = $this->input->post('id_produk');
    $tanggal_produk    = $this->input-
>post('tanggal_produk');
    $no_mesin    = $this->input->post('no_mesin');
    $id_produk   = $this->input->post('id_produk');
    $jumlah      = $this->input->post('jumlah');

```

```

        $data = array('tanggal_produksi' =>
$tanggal_produksi,'no_mesin' => $no_mesin,'id_produk' =>
$id_produk,'jumlah' => $jumlah);
        $simpan = $this->M_produksi->update($data, $id);
        redirect('Produksi');
    }
}

```

6. Laporan.php

```

<?php

class Laporan extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model(array('M_produk', 'M_mesin',
'M_produksi','M_qcontrol','M_penyebab','M_laporan'));
        $this->load->library('form_validation');
    }

    public function index()
    {

        $id_produk = $this->input->post('id_produk');
        $tgl_mulai = $this->input->post('tgl_mulai');
        $tgl_akhir = $this->input->post('tgl_akhir');
        // $tgl_mulai = '2019-06-01';
        // $tgl_akhir = '2019-06-31';
        // $id_produk = '4';
    }
}

```

```

        $data['produk'] = $this->M_produk->tampilkan_data();

        $data['record'] = $this->M_laporan-
>diagram_pareto($id_produk,$tgl_mulai,$tgl_akhir)->result();
        $data['data_total'] = $this->M_laporan-
>data_total($id_produk,$tgl_mulai,$tgl_akhir)->row_array();
        $data['detail_produksi'] = $this->M_laporan-
>detail_produksi($id_produk,$tgl_mulai,$tgl_akhir)->result();

        $data['tgl_mulai'] = $tgl_mulai;
        $data['tgl_akhir'] = $tgl_akhir;
        $data['id_produk'] = $id_produk;
        // echo "<pre>";
        // print_r($data);
        // echo "</pre>";
        // echo "<pre>";
        // print_r($_POST);
        // echo "</pre>";
        // die;

        $this->template->load('template',
'laporan/diagram_pareto',$data);
        // die;

    }

    public function tambah($id_produksi)
    {
        $data['penyebab'] = $this->M_penyebab-
>tampilkan_data();
        $data['record'] = $this->M_qcontrol-
>tampilkan_data_detail($id_produksi)->result();

```

```

        $data['produksi'] = $this->M_qcontrol-
>get_data_produksi_by_id($id_produksi)->row_array();
        $this->template->load('template',
'quality_control/form_input', $data);
    }

    public function simpan()
    {

        $id_produksi = $this->input->post('id_produksi');
        $id_penyebab = $this->input->post('id_penyebab');
        $jumlah_cacat = $this->input->post('jumlah_cacat');
        $data = array('id_produksi' => $id_produksi, 'id_penyebab'
=> $id_penyebab, 'jumlah_cacat' => $jumlah_cacat);
        $simpan = $this->M_qcontrol->simpan($data);

        $produksi = $this->M_qcontrol-
>get_data_produksi_by_id($id_produksi)->row_array();

        $cacat = $this->M_qcontrol-
>get_total_cacat($id_produksi)->row_array();

        $ok = $produksi['jumlah'] - $cacat['total_cacat'];

        $data = array('ok' => $ok, 'cacat' => $cacat['total_cacat']);
        $simpan = $this->M_produksi->update($data,
$id_produksi);

        redirect('Qcontrol/tambah/'.$id_produksi);
    }

```

```

public function hapus($id_produk)
{
    $id_produk = $this->uri->segment(3);
    $id_cacat_produk = $this->uri->segment(4);
    $this->M_qcontrol->hapus($id_cacat_produk);

    $produksi = $this->M_qcontrol-
>get_data_produk_by_id($id_produk)->row_array();

    $cacat = $this->M_qcontrol-
>get_total_cacat($id_produk)->row_array();

    $ok = $produksi['jumlah'] - $cacat['total_cacat'];

    $data = array('ok' => $ok, 'cacat' => $cacat['total_cacat']);
    $simpan = $this->M_produk->update($data,
$id_produk);

    redirect('Qcontrol/tambah/'.$id_produk);
}

public function selesai($id_produk){
    $id_produk = $this->uri->segment(3);
    $data = array('status' => '1');
    $simpan = $this->M_produk->update($data,
$id_produk);
    redirect('Qcontrol');
}

```



```

public function ubah()
{
    $id = $this->uri->segment(3);
    $data['produk'] = $this->M_produk->tampilkan_data();
    $data['mesin'] = $this->M_mesin->tampilkan_data();
    $data['record'] = $this->M_produksi->get_data($id)-
>row_array();
    $this->template->load('template',
'quality_control/form_edit', $data);
}

public function edit()
{
    $id      = $this->input->post('id_produksi');
    $tanggal_produksi    = $this->input-
>post('tanggal_produksi');
    $no_mesin    = $this->input->post('no_mesin');
    $id_produk   = $this->input->post('id_produk');
    $jumlah      = $this->input->post('jumlah');

    $data = array('tanggal_produksi' =>
$tanggal_produksi, 'no_mesin' => $no_mesin, 'id_produk' =>
$id_produk, 'jumlah' => $jumlah);
    $simpan = $this->M_produksi->update($data, $id);
    redirect('Produksi');
}

public function cetak_diagram_pareto(){
    $tgl_mulai = $this->uri->segment(3);

```

```

        $tgl_akhir = $this->uri->segment(4);
        $id_produk = $this->uri->segment(5);

        $data['produk'] = $this->M_produk->tampilkan_data();

        $data['record'] = $this->M_laporan-
>diagram_pareto($id_produk,$tgl_mulai,$tgl_akhir)->result();
        $data['tgl_mulai'] = $tgl_mulai;
        $data['tgl_akhir'] = $tgl_akhir;
        $data['id_produk'] = $id_produk;
        $this->template->load('template',
'laporan/cetak_diagram_pareto',$data);
    }

```

```

public function laporan_produksi(){
    $tgl_mulai = $this->input->post('tgl_mulai');
    $tgl_akhir = $this->input->post('tgl_akhir');
    $data['tgl_mulai'] = $tgl_mulai;
    $data['tgl_akhir'] = $tgl_akhir;
    $data['id_produk'] = $id_produk;
    $data['record'] = $this->M_laporan-
>laporan_produksi($tgl_mulai,$tgl_akhir)->result();
    $this->template->load('template',
'laporan/laporan_produksi',$data);
}

```

```

public function cetak_laporan_produksi(){
    $tgl_mulai = $this->uri->segment(3);
    $tgl_akhir = $this->uri->segment(4);
    $data['tgl_mulai'] = $tgl_mulai;
    $data['tgl_akhir'] = $tgl_akhir;

```

```

        $data['id_produk'] = $id_produk;
        $data['record'] = $this->M_laporan-
>laporan_produk($tgl_mulai,$tgl_akhir)->result();
        $this->load->view('laporan/cetak_laporan_produk',$data);

    }

```

```

    public function laporan_cacat_produk(){
        $tgl_mulai = $this->input->post('tgl_mulai');
        $tgl_akhir = $this->input->post('tgl_akhir');
        $data['tgl_mulai'] = $tgl_mulai;
        $data['tgl_akhir'] = $tgl_akhir;
        $data['record'] = $this->M_laporan-
>laporan_cacat_produk($tgl_mulai,$tgl_akhir)->result();
        $this->template->load('template',
'laporan/laporan_cacat_produk',$data);
    }

```

```

    public function cetak_laporan_cacat_produk(){
        $tgl_mulai = $this->uri->segment(3);
        $tgl_akhir = $this->uri->segment(4);
        $data['tgl_mulai'] = $tgl_mulai;
        $data['tgl_akhir'] = $tgl_akhir;
        $data['record'] = $this->M_laporan-
>laporan_cacat_produk($tgl_mulai,$tgl_akhir)->result();
        $this->load-
>view('laporan/cetak_laporan_cacat_produk',$data);
    }
}

```

Lampiran B

Black-Box Testing

1. *Form Login*

Test case ID : Login001.

Function : Operasi validasi saat melakukan *login*.

Data Assumption : Fungsi operasi validasi saat melakukan login sudah berjalan dengan baik, penggunaan huruf kapital dan huruf kecil tidak mempengaruhi pada validasi *login*.

Deskripsi : Melakukan *login* ke dalam sistem dengan menguji kesalahan *username* atau *password*.

<i>Test ID</i>	<i>Test Case</i>	<i>Expected</i>	<i>Actual</i>	<i>Result</i>
001	<i>Login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang benar, lalu klik tombol <i>login</i> atau <i>enter</i> .	Masuk kedalam program	Masuk kedalam program	Valid
002	<i>Login</i> dengan memasukkan <i>username</i> dan <i>password</i> yang salah, lalu klik tombol <i>login</i> atau <i>enter</i> .	Tidak masuk kedalam program	Tidak masuk kedalam program	Valid
003	<i>Login</i> tidak memasukkan salah satu antara <i>username</i> dan <i>password</i> . lalu klik tombol <i>login</i> atau <i>enter</i> .	Tidak masuk kedalam program	Tidak masuk kedalam program	Valid

2. *Menu Produk*

Test case ID : Produk001.

Function : Operasi mengelola data produk.

Data Assumption : Fungsi operasi data produk digunakan untuk mengelola data produk.

Deskripsi : Mengklik tambah, ubah, hapus pada data produk.

Test ID	Test Case	Expected	Actual	Result
001	Mengklik tambah pada <i>form</i> produk	Program akan menampilkan <i>form</i> inputan produk	Menampilkan <i>form</i> inputan produk	Valid
002	Mengklik edit pada <i>form</i> produk	Program akan menampilkan <i>form</i> edit produk	Menampilkan <i>form</i> edit produk	Valid
003	Mengklik tombol delete pada <i>form</i> produk	Program akan menghapus data	Menghapus data	Valid

3. **Form tambah data produk**

Test case ID : Tambahproduk001.

Function : Operasi tambah data produk.

Data Assumption : Fungsi tambah data produk digunakan untuk menambah data produk.

Deskripsi : Mengklik tambah, lalu akan menampilkan *form* data tambah produk.

Test ID	Test Case	Expected	Actual	Result
001	Mengisi semua data sesuai dengan inputan lalu mengklik simpan atau enter	Program akan menambah data kedalam <i>database</i> dan akan kembali ke <i>form</i> produk	Program menambah data kedalam <i>database</i> dan kembali ke menu <i>form</i> produk	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> produk	Program kembali ke menu <i>form</i> produk	Valid

4. **Form ubah data produk**

Test case ID : Ubahproduk001.

Function : Operasi ubah data produk.

Data Assumption : Fungsi Mengubah data produk digunakan untuk mengubah data produk.

Deskripsi : Mengklik ubah pada *field* aksi, lalu akan menampilkan *form* data mengubah produk.

Test ID	Test Case	Expected	Actual	Result
001	Mengubah data sesuai dengan keinginan user lalu mengklik simpan atau enter	Program akan mengubah data didalam <i>database</i> produk	Program mengubah data didalam <i>database</i> produk	Valid
002	Mengubah data dan mengosongkan data lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> produk	Program kembali ke menu <i>form</i> produk	Valid

5. Menu Penyebab Cacat

Test case ID : Penyebabcacat001.

Function : Operasi mengelola data Penyebab cacat.

Data Assumption : Fungsi operasi data penyebab digunakan untuk mengelola data penyebab cacat yang akan digunakan di menu/tabel cacat produksi.

Deskripsi : Mengklik tambah, ubah, hapus pada data penyebab cacat.

Test ID	Test Case	Expected	Actual	Result
001	Mengklik tambah pada <i>form</i> penyebab cacat	Program akan menampilkan <i>form</i> inputan penyebab cacat	Menampilkan <i>form</i> inputan penyebab cacat	Valid
002	Mengklik edit pada <i>form</i> penyebab cacat	Program akan menampilkan <i>form</i> edit penyebab cacat	Menampilkan <i>form</i> edit penyebab cacat	Valid
003	Mengklik tombol delete pada <i>form</i> penyebab cacat	Program akan menghapus data	Menghapus data	Valid

6. Form tambah data penyebab cacat

Test case ID : TambahPenyebabcacat001.

Function : Operasi tambah data produk.

Data Assumption : Fungsi tambah data penyebab cacat digunakan untuk menambah data penyebab cacat.

Deskripsi : Mengklik tambah, lalu akan menampilkan *form* data penyebab cacat.

No	Test Case	Expected	Actual	Result
001	Mengisi semua data sesuai dengan inputan lalu mengklik simpan atau enter	Program akan menambah data kedalam <i>database</i> dan akan kembali ke <i>form</i> penyebab cacat	Program menambah data kedalam <i>database</i> dan kembali ke menu <i>form</i> penyebab cacat	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> penyebab cacat	Program kembali ke menu <i>form</i> penyebab cacat	Valid

7. Form Ubah Data Penyebab Cacat

Test case ID : Ubahpenyebabcacat001.

Function : Operasi ubah data penyebab cacat.

Data Assumption : Fungsi Mengubah data penyebab cacat digunakan untuk mengubah data penyebab cacat.

Deskripsi : Mengklik ubah pada *field* aksi, lalu akan menampilkan *form* data mengubah penyebab cacat.

Test ID	Test Case	Expected	Actual	Result
001	Mengubah data sesuai dengan keinginan user lalu mengklik simpan atau enter	Program akan mengubah data didalam <i>database</i> penyebab cacat	Program mengubah data didalam <i>database</i> penyebab cacat	Valid
002	Mengubah data dan mengosongkan data lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> penyebab cacat	Program kembali ke menu <i>form</i> penyebab cacat	Valid

8. Menu User

Test case ID : User001.

Function : Operasi mengelola data user.

Data Assumption : Fungsi operasi data user digunakan untuk mengelola data hak akses pada menu.

Deskripsi : Mengklik tambah, ubah, hapus pada data produk.

Test ID	Test Case	Expected	Actual	Result
001	Mengklik tambah pada <i>form</i> user	Program akan menampilkan <i>form</i> inputan user	Menampilkan <i>form</i> inputan user	Valid
002	Mengklik edit pada <i>form</i> user	Program akan menampilkan <i>form</i> edit user	Menampilkan <i>form</i> edit user	Valid
003	Mengklik tombol delete pada <i>form</i> user	Program akan menghapus data	Menghapus data	Valid

9. Form tambah user

Test case ID : Tambahuser001.

Function : Operasi tambah data user.

Data Assumption : Fungsi tambah data user digunakan untuk menambah data produk.

Deskripsi : Mengklik tambah, lalu akan menampilkan *form* data tambah user.

Test ID	Test Case	Expected	Actual	Result
001	Mengisi semua data sesuai dengan inputan lalu mengklik simpan atau enter	Program akan menambah data kedalam <i>database</i> dan akan kembali ke <i>form</i> user	Program menambah data kedalam <i>database</i> dan kembali ke menu <i>form</i> user	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> user	Program kembali ke menu <i>form</i> user	Valid

10. *Form* ubah data user

Test case ID : Ubahuser001.

Function : Operasi ubah data user.

Data Assumption : Fungsi Mengubah data user digunakan untuk mengubah data produk.

Deskripsi : Mengklik ubah pada *field* aksi, lalu akan menampilkan *form* data mengubah user.

Test ID	Test Case	Expected	Actual	Result
001	Mengubah data sesuai dengan keinginan user lalu mengklik simpan atau enter	Program akan mengubah data didalam <i>database</i> user	Program mengubah data didalam <i>database</i> user	Valid
002	Mengubah data dan mengosongkan data lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> user	Program kembali ke menu <i>form</i> user	Valid

11. Menu Hasil Produksi

Test case ID : Hasilproduksi001.

Function : Operasi mengelola data hasil produksi.

Data Assumption : Fungsi operasi data produk digunakan untuk mengelola data produk.

Deskripsi : Mengklik tambah, ubah, hapus pada data hasil produksi.

<i>Test ID</i>	Test Case	Expected	Actual	Result
001	Mengklik tambah pada <i>form</i> hasil produksi	Program akan menampilkan <i>form</i> inputan hasil produksi	Menampilkan <i>form</i> inputan hasil produksi	Valid
002	Mengklik edit pada <i>form</i> hasil produksi	Program akan menampilkan <i>form</i> edit hasil produksi	Menampilkan <i>form</i> edit hasil produksi	Valid
003	Mengklik tombol delete pada <i>form</i> hasil produksi	Program akan menghapus data	Menghapus data	Valid

12. Form tambah data hasil produksi

Test case ID : Tambahhasilproduksi 001.

Function : Operasi tambah data hasil produksi.

Data Assumption : Fungsi tambah data hasil produksi digunakan untuk menambah data hasil produksi.

Deskripsi : Mengklik tambah, lalu akan menampilkan *form* data tambah hasil produksi.

Test ID	Test Case	Expected	Actual	Result
001	Mengisi semua data sesuai dengan inputan lalu mengklik simpan atau enter	Program akan menambah data kedalam <i>database</i> dan akan kembali ke <i>form</i> hasil produksi	Program menambah data kedalam <i>database</i> dan kembali ke menu <i>form</i> hasil produksi	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> hasil produksi	Program kembali ke menu <i>form</i> hasil produksi	Valid

13. Form ubah data hasil produksi

Test case ID : Ubahhasilproduksi 001.

Function : Operasi ubah data hasil produksi.

Data Assumption : Fungsi Mengubah data hasil produksi digunakan untuk mengubah data hasil produksi.

Deskripsi : Mengklik ubah pada *field* aksi, lalu akan menampilkan *form* data mengubah hasil produksi.

Test ID	Test Case	Expected	Actual	Result
001	Mengubah data sesuai dengan keinginan user lalu mengklik simpan atau enter	Program akan mengubah data didalam <i>database</i> hasil produksi	Program mengubah data didalam <i>database</i> hasil produksi	Valid
002	Mengubah data dan mengosongkan data lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik tombol kembali	Program akan kembali ke menu <i>form</i> hasil produksi	Program kembali ke menu <i>form</i> hasil produksi	Valid

14. Menu Cacat Produksi

Test case ID : Cacatproduksi001.

Function : Operasi mengelola data cacat produksi.

Data Assumption : Fungsi operasi data produk digunakan untuk mengelola data cacat produksi.

Deskripsi : Mengklik *input, detail* pada data cacat produksi.

Test ID	Test Case	Expected	Actual	Result
001	Mengklik <i>input</i> pada <i>form</i> hasil produksi	Program akan menampilkan <i>form</i> inputan cacat produksi	Menampilkan <i>form</i> inputan cacat produksi	Valid
002	Mengklik <i>detail</i> pada <i>form</i> cacat produksi	Program akan menampilkan <i>form detail</i> cacat produksi	Menampilkan <i>form detail</i> cacat produksi	Valid

15. *Form input cacat produksi*

Test case ID : Inputcacatproduksi001.

Function : Operasi mengelola data cacat produksi.

Data Assumption : Fungsi operasi data produk digunakan untuk mengelola data cacat produksi.

Deskripsi : Mengklik *input, detail* pada data cacat produksi.

Test ID	Test Case	Expected	Actual	Result
001	Mengklik <i>input</i> pada <i>form</i> cacat produksi	Program akan menampilkan <i>form</i> inputan hasil produksi	Menampilkan <i>form</i> inputan cacat produksi	Valid
002	Mengklik hapus pada <i>form</i> cacat produksi	Program akan menghapus data <i>form</i> cacat produksi	Program menghapus data <i>form</i> cacat produksi	Valid
003	Mengklik <i>simpan</i> pada <i>form</i> cacat produksi	Program akan menyimpan data <i>form</i> cacat produksi	Program menyimpan data <i>form</i> cacat produksi	Valid
004	Mengklik <i>kembali</i> pada <i>form</i> cacat produksi	Program akan kembali kemenu cacat produksi	Program kembali kemenu cacat produksi	Valid

16. *Form detail cacat produksi*

Test case ID : Detailcacatproduksi001.

Function : Operasi mengelola data cacat produksi.

Data Assumption : Fungsi operasi data produk digunakan untuk mengelola data cacat produksi.

Deskripsi : Mengklik *input, detail* pada data cacat produksi.

Test ID	Test Case	Expected	Actual	Result
001	Mengklik <i>detail</i> pada <i>form detail</i> cacat produksi	Program akan menampilkan <i>form detail</i> cacat produksi	Menampilkan <i>form detail</i> cacat produksi	Valid
002	Mengklik kembali pada <i>form detail</i> cacat produksi	Program akan kembali kemenu cacat produksi	Program kembali kemenu cacat produksi	Valid

17. Menu Laporan Grafik

Test case ID : Laporangrafik001.

Function : Menampilkan dan mencetak grafik.

Data Assumption : Fungsi operasi data laporan cetak digunakan untuk menampilkan dan mencetak grafik.

Deskripsi : Mengklik lihat dan cetak pada data laporan grafik.

Test ID	Test Case	Expected	Actual	Result
001	Menginput lalu mengklik lihat	Program akan menampilkan grafik.	Program menampilkan grafik.	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik cetak	Program akan menampilkan halaman cetak	Program menampilkan halaman cetak	Valid

18. Menu Laporan Hasil Produksi

Test case ID : Laporanhasilproduksi001.

Function : Menampilkan dan mencetak hasil produksi.

Data Assumption : Fungsi operasi data laporan cetak digunakan untuk menampilkan dan mencetak hasil produksi.

Deskripsi : Mengklik lihat dan cetak pada data laporan hasil produksi.

Test ID	Test Case	Expected	Actual	Result
001	Menginput lalu mengklik lihat	Program akan menampilkan hasil produksi.	Program menampilkan hasil produksi.	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik cetak	Program akan menampilkan halaman cetak	Program menampilkan halaman cetak	Valid

19. Menu Laporan Cacat Produksi

Test case ID : Laporancacatproduksi001.

Function : Menampilkan dan mencetak cacat produksi.

Data Assumption : Fungsi operasi data laporan cetak digunakan untuk menampilkan dan mencetak cacat produksi.

Deskripsi : Mengklik lihat dan cetak pada data laporan cacat produksi.

Test ID	Test Case	Expected	Actual	Result
001	Menginput lalu mengklik lihat	Program akan menampilkan cacat produksi.	Program menampilkan cacat produksi.	Valid
002	Mengosongkan salah satu inputan lalu mengklik simpan atau enter	Program akan memberi tahu bahwa data harus diisi	Program memberi tahu bahwa data harus diisi	Valid
003	Mengklik cetak	Program akan menampilkan halaman cetak	Program menampilkan halaman cetak	Valid