No. Dok : 6592

D 658.562 Pra S.

SISTEM INFORMASI *MONITORING QUALITY CONTROL SAMPLING PRODUCT* MENGGUNAKAN *CODE IGNITER* 3.1.4 DAN *POSTGRESQL* PADA DIVISI *QUALITY CONTROL* PT NUSA INDAH JAYA UTAMA

SUMBANGAN ALUMNI

TUGAS AKHIR

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

OLEH
HAFIYYAN ARIF PRAKOSO
1315113

DATA BUKU PERPUSTAKAAN

Tgl Terima	27/07/2022
No Induk Buku	508/S110/58/TA/22

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN RI
JAKARTA
2019

POLITEKNIK STMI JAKARTA
KEMENTERIAN PERINDUSTRIAN R.I

TANDA PERSETUJUAN DOSEN PEMBIMBING

JUDUL TUGAS AKHIR:

SISTEM INFORMASI *MONITORING QUALITY CONTROL SAMPLING PRODUCT* MENGGUNAKAN *CODE IGNITER* 3.1.4 DAN *POSTGRESQL* PADA DIVISI *QUALITY CONTROL* PT NUSA INDAH JAYA UTAMA

Disusun Oleh:

Nama	: Hafiyyan Arif Prakoso
NIM	: 1315113
Program Studi	: Sistem Informasi Industri Otomotif
Tanggal Seminar	: 14 Agustus 2019
Tanggal Sidang	: 23 Agustus 2019
Tanggal Lulus	: 23 Agustus 2019

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

Jakarta, 23 Agustus 2019
Dosen Pembimbing

a.n

(Noveriza Yuliasari, S.Si, M.T)
NIP. 197811212009012003

POLITEKNIK STMI JAKARTA

KEMENTERIAN PERINDUSTRIAN REPUBLIK INDONESIA

LEMBAR PENGESAHAN

JUDUL TUGAS AKHIR:

SISTEM INFORMASI *MONITORING QUALITY CONTROL SAMPLING PRODUCT* MENGGUNAKAN *CODE IGNITER* 3.1.4 DAN *POSTGRESQL* PADA DIVISI *QUALITY CONTROL* PT NUSA INDAH JAYA UTAMA

Disusun Oleh:

Nama : Hafiyyan Arif Prakoso

NIM : 1315113

Program Studi : Sistem Informasi Industri Otomotif

Telah diuji oleh Tim Penguji Sidang Tugas Akhir Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta Kementerian Perindustrian Republik Indonesia pada hari Kamis tanggal 23 Agustus 2019.

Jakarta, 23 Agustus 2019

Dosen Pembimbing	Ketua Penguji
a.n	
Noveriza Yulfasari, S.Si, M.T.	**Dr. Ridzky Kramanandita, S.Kom, M.T.**
NIP.197811212009012003	NIP. 197403022002121001
Dosen Penguji	Dosen Penguji
Ulil Hamida, S.T, M.T.	**Lucky Heriyanto, S.T, M.T.I.**
NIP. 198103272005022001	NIP. 197908202009011009

Kementerian Perindustrian
REPUBLIK INDONESIA

BADAN PENGEMBANGAN SUMBER DAYA MANUSIA INDUSTRI
POLITEKNIK STMI JAKARTA
Jl. Letjen Suprapto No. 26 Cempaka Putih, Jakarta 10510
Telp: (021) 42886064 Fax: (021) 42888206
www.stmi.ac.id

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Hafiyyan Arif Prakoso
NIM : 1315113
Judul Tugas Akhir : Sistem Informasi *Monitoring Quality Control Sampling Product* Menggunakan *Code Igniter* 3.1.4 Dan *Postgresql* Pada Divisi *Quality Control* PT Nusa Indah Jaya Utama
Pembimbing : Noveriza Yuliasari, S.Si, M.T

Tanggal	BAB	Keterangan	Paraf
27-05-2019	1,2, dan 3	Pengajuan bab 1,2, dan 3	
29-05-2019	1,2, dan 3	Revisi bab 1,2, dan 3	
13-06-2019	2,3, dan 4	Revisi bab 2,3 dan mengajukan bab 4	
21-06-2019	3,4, dan 5	Revisi bab 3,4 dan mengajukan bab 5	
25-06-2019	5	Revisi bab 5	
3-07-2019	5	Revisi bab 5	
12-07-2019	5	Revisi bab 5	
23-07-2019	5	Revisi bab 5 dan demo program	
25-07-2019	5	Revisi bab 5 dan demo program	
30-07-2019	5 dan 6	Revisi bab 5 dan mengajukan bab 6	
31-07-2019	6	Revisi bab 6	
1-08-2019		Periksa kelengkapan	

Mengetahui,
Ketua Program Studi
Sistem Informasi Industri Otomotif

Noveriza Yuliasari, S.Si, M.T
NIP.197811212009012003

Dosen Pembimbing

Noveriza Yuliasari, S.Si, M.T
NIP.197811212009012003

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini:

Nama : Hafiyyan Arif Prakoso

NIM : 1315113

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

" SISTEM INFORMASI *MONITORING QUALITY CONTROL SAMPLING PRODUCT* MENGGUNAKAN *CODE IGNITER* 3.1.4 DAN *POSTGRESQL* PADA DIVISI *QUALITY CONTROL* PT NUSA INDAH JAYA UTAMA "

- **Dibuat** dan diselesaikan sendiri, dengan menggunakan literatur hasil kuliah, survey lapangan, dibantu oleh dosen pembimbing maupun asisten pembimbing, serta buku-buku maupun jurnal-jurnal ilmiah yang menjadi bahan acuan yang tertera dalam referensi karya tugas akhir ini.
- **Bukan** merupakan hasil duplikasi hasil 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

Hafiyyan Arif Prakoso

ABSTRAK

PT Nusa Indah Jaya Utama merupakan perusahaan yang bergerak di bidang manufaktur yang memproduksi komponen otomotif yang dipesan dari berbagai pelanggan. Sebagai pelaku bisnis dalam bidang otomotif dituntut untuk terus melakukan inovasi yang membuat keunggulan kompetitif dari usaha bisnisnya. Banyak perusahaan yang sudah menerapkan strategi–strategi bisnis tertentu tetapi masih belum dapat mencapai hasil yang optimal. Salah satu penyebabnya adalah kualitas produk yang kurang diperhatikan. Oleh karena itu, perusahaan harus mampu melakukan perbaikan-perbaikan secara berkelanjutan dalam menjaga dan meningkatkan kualitas produknya. Sebagai perusahaan yang sedang berkembang, PT Nusa Indah Jaya Utama tentunya tidak terlepas dari permasalahan-permasalahan yang terjadi di antaranya kesulitan dalam pencarian data karena banyaknya dokumen yang menumpuk di dalam arsip penyimpanan. Selain itu, proses penghitungan laporan masih menggunakan *Microsoft Excel* yang kurang informatif dan tidak ada sistem yang terintegrasi sehingga Divisi *Quality Control* membutuhkan waktu yang lama dalam menghitung jumlah kualitas produk harian karena menunggu seluruh produk selesai dicek. Untuk mengatasi masalah tersebut, diperlukan pengembangan sistem informasi berdasarkan pada analisis dokumen dan proses bisnis yang dapat menyimpan data ke dalam *database* sebagai media penyimpanannya. Aplikasi yang dibangun diharapkan dapat membantu perusahaan dalam mengelola manajemen data yang ada, serta membantu membuat laporan *monitoring reject* menjadi lebih informatif. Sistem informasi juga menyediakan informasi jumlah kualitas produk yang dapat menjadi acuan untuk peningkatan kualitas pada periode berikutnya.

Kata Kunci : Sistem Informasi, *Quality Control Monitoring*, *Sampling Product*, *Reject Monitoring*.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, yang diberikan kepada penulis agar dapat menyelesaikan Tugas Akhir dengan judul **“ Sistem Informasi *Monitoring Quality Control Sampling Product* Menggunakan *Code Igniter* 3.1.4 Dan *Postgresql* Pada Divisi *Quality Control* PT Nusa Indah Jaya Utama”**.

Tugas akhir ini disusun guna memenuhi sebagian syarat yang harus dipenuhi dalam menempuh program sarjana terapan jurusan Sistem Informasi Industri Otomotif pada Politeknik STMI Jakarta.

Dalam penyusunan Tugas Akhir ini, penulis banyak mendapat bantuan serta bimbingan dari berbagai pihak. Pada kesempatan ini tidak lupa penulis sampaikan ucapan terima kasih kepada:

1. Allah SWT atas rahmat serta kemudahan yang diberikan.
2. Kedua orang tua serta keluarga tercinta yang telah memberikan do’a, dukungan, pengorbanan, semangat dan kasih sayang hingga saat ini.
3. Bapak Dr. Mustofa, S.T, M.T selaku Direktur Politeknik STMI Jakarta.
4. Ibu Noveriza Yuliasari, S.Si, M.T selaku Ketua Program Studi Sistem Informasi Industri Otomotif Politeknik STMI Jakarta dan dosen pembimbing yang memberi arahan dan masukan dalam penyusunan Tugas Akhir.
5. Dosen-dosen Politeknik STMI Jakarta, khususnya untuk dosen Program Studi Sistem Informasi Industri Otomotif yang telah memberikan mata kuliah serta pengarahan selama perkuliahan.
6. Bapak H.Saipudin Dipl.Eng selaku pembimbing yang telah memberikan pengetahuan baik secara lisan, tulisan, maupun dokumen terkait sehingga penulis dapat menyelesaikan Tugas Akhir.

7. Bapak H.Asrul Nasution, Bapak H.Hermawan, Bapak H.Soetarman dan Seluruh karyawan PT Nusa Indah Jaya Utama yang telah memberikan izin kepada penulis untuk melakukan pengamatan, memberikan motivasi dan ilmunya, serta membatu dan mempermudah penulis dalam menyelesaikan Tugas Akhir.
8. Semua teman-teman yang senantiasa saling mendukung satu sama lain terutama teman-teman dari Program Studi Sistem Informasi Industri Otomotif angkatan 2015.

Semoga Allah SWT membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan Tugas Akhir ini dengan melimpahkan rahmat dan karunia-Nya. Besar harapan Penulis bahwa Tugas Akhir ini dapat memberikan manfaat serta pengetahuan bagi pembacanya. Terima kasih.

Jakarta, 5 Agustus 2019

Penulis

DAFTAR ISI

Halaman

TANDA PERSETUJUAN DOSEN PEMBIMBING ..... **Error! Bookmark not defined.**

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 Penelitian 4

1.6 Sistematika Penulisan 4

BAB II LANDASAN TEORI 6

2.1 Pengertian Rancang Bangun 6

2.2 Konsep Dasar Sistem 6

2.2.1 Tujuan Sistem 7

2.2.2 Karakteristik Sistem 7

2.2.3 Klasifikasi Sistem 10

2.3 Hakikat Informasi 11

2.3.1 Pengertian Informasi 11

2.3.2 Tujuan Informasi 11

2.3.3 Klasifikasi Informasi 11
2.3.4 Kualitas Informasi 12
2.4 Konsep Dasar Sistem Informasi 13
2.5 Pengertian *Monitoring* 13
2.5.1 Sistem *Monitoring* 14
2.5.2 Tujuan Sistem *Monitoring* 14
2.6 Pengertian Pengendalian Kualitas 14
2.6.1 Tujuan Pengendalian Kualitas 15
2.6.2 Faktor Pengendalian Kualitas 16
2.7 Pengertian Kualitas Produk 16
2.8 Siklus Deming 18
2.8.1 Tujuh Alat Pemecahan Masalah 18
2.9 *System Development Life Cycle* (SDLC) 19
2.10 Metode *Waterfall* 21
2.11 *Flowmap* 22
2.12 *Unified Modelling Language* (UML) 24
2.12.1 *Use Case Diagram* 26
2.12.2 *Activity Diagram* 27
2.12.3 *Sequence Diagram* 28
2.12.4 *Class Diagram* 30
2.12.5 *Deployment Diagram* 31
2.13 Basis Data 32
2.14 *Entity Relationship Diagram* (ERD) 32
2.15 *Conceptual Data Model* (CDM) 34
2.16 *Physical Data Model* (PDM) 35
2.17 Kamus Data 35
2.18 *Windows Navigation Diagram* (WND) 35
2.19 *Hypertext Preprocessor* (PHP) 36
2.20 *PostgreSQL* 37
2.21 *Code Igniter* 38
2.22 *Black Box Testing* 38

BAB III METODOLOGI PENELITIAN 40
3.1 Metodologi Penelitian 40
3.2 Jenis Dan Sumber Data 40
3.3 Metode Pengumpulan Data 41
3.4 Kerangka Penelitian 41

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA 46
4.1 Sekilas Perusahaan 46
4.2 Profil Perusahaan 47
4.3 Logo Perusahaan 47
4.4 Visi dan Misi Perusahaan 48
4.5 Struktur Organisasi 49
4.6 Tugas dan Wewenang Divisi QC 51
4.7 Analisis Dokumen Kualitas Produk 52
4.8 Analisis Proses Bisnis 57

BAB V ANALISIS DAN PEMBAHASAN 51
5.1 Analisis Kebutuhan Sistem 51
5.2 Prosedur Sistem Informasi *Monitoring Quality Control* 64
5.3 Analisis Sistem Usulan 65
5.3.1 *Use Case Diagram* 66
5.3.2 *Activity Diagram* 78
5.3.3 *Sequence Diagram* 93
5.3.4 *Class Diagram* 106
5.3.5 *Deployment Diagram*..... 107
5.4 Pemodelan Data 107
5.4.1 *Entity Relationship Diagram* 107
5.4.2 *Conseptual Data Model* 108
5.4.3 Kamus Data 109
5.5 Perancangan Sistem 112
5.5.1 *Windows Navigation Diagram* 112
5.5.2 Perancangan *Interface* Sistem 114

5.6 Implementasi Sistem 127
5.7 Pengujian Sistem 127

BAB VI KESIMPULAN DAN SARAN 114
6.1 Kesimpulan 114
6.2 Saran 114

DAFTAR PUSTAKA 114

DAFTAR GAMBAR

Halaman

Gambar II.1 Karakteristik Sistem 9
Gambar II.2 Metode *Waterfall* Menurut Dennis..... 22
Gambar II.3 Contoh *Windows Navigation Diagram* 36
Gambar III.1 Kerangka Penelitian 45
Gambar IV.1 Logo PT Nusa Indah Jaya Utama 48
Gambar IV.2 Struktur Orgsnisasi PT Nusa Indah Jaya Utama..... 51
Gambar IV.3 Struktur Orgsnisasi Divisi QC pada PT Nusa Indah Jaya Utama... 51
Gambar IV.4 Dokumen *Check Sheet* 52
Gambar IV.5 Dokumen *Problem Countermeasure* 53
Gambar IV.6 Dokumen *Repair Card*..... 54
Gambar IV.7 Dokumen *Part Not Good* 55
Gambar IV.8 Dokumen *Monitoring Reject* 56
Gambar IV.9 *Flowmap* Sistem Informasi *Monitoring Quality Control*…….59
Gambar IV.10 *Use Case Sistem Informasi Monitoring Quality Control*..... 60
Gambar V.1 *Flowmap* Sistem Informasi *Monitoring Quality Control* usulan 65
Gambar V.2 *Use Case* Sistem Informasi *Monitoring Quality Control*..... 66
Gambar V.3 *Activity Diagram Login*..... 79
Gambar V.4 *Activity Diagram* Mengelola Data Pengguna..... 80
Gambar V.5 *Activity Diagram* Mengelola Data Barang..... 81
Gambar V.6 *Activity Diagram* Mengelola Data Pelanggan……...82
Gambar V.7 *Activity Diagram* Membuat Data *Check Sheet*..... 83
Gambar V.8 *Activity Diagram* Memeriksa Data *Check Sheet*..... 84
Gambar V.9 *Activity Diagram* Mengkonfirmasi Sampel sebagai Produk Baik ... 85
Gambar V.10 *Activity Diagram* Membuat Data *Problem Countermeasure*..... 86
Gambar V.11 *Activity Diagram* Memeriksa *Data Problem Countermeasure*..... 87
Gambar V.12 *Activity Diagram* Membuat Data *Repair Card* 88
Gambar V.13 *Activity Diagram* Memeriksa Data *Repair Card* 89

Gambar V.14 *Activity Diagram* Memperbarui Status Perbaikan..... 90
Gambar V.15 *Activity Diagram* Membuat Data *Part Not Good* 91
Gambar V.16 *Activity Diagram* Membuat Laporan *Monitoring Reject* 92
Gambar V.17 *Activity Diagram* Mencetak Laporan *Monitoring Reject*..... 93
Gambar V.18 *Sequence Diagram Login*..... 94
Gambar V.19 *Sequence Diagram* Mengelola Data Pengguna..... 95
Gambar V.20 *Sequence Diagram* Mengelola Data Barang..... 96
Gambar V.21 *Sequence Diagram* Mengelola Data Pelanggan..... 97
Gambar V.22 *Sequence Diagram* Membuat Data *Check Sheet*..... 98
Gambar V.23 *Sequence Diagram* Memeriksa Data *Check Sheet*..... 99
Gambar V.24 *Sequence Diagram* Mengkonfirmasi Sampel..... 99
Gambar V.25 *Sequence Diagram* Membuat Data *Problem Countermeasure*.... 100
Gambar V.26 *Sequence Diagram* Memeriksa Data *Problem Countermeasure* . 101
Gambar V.27 *Sequence Diagram* Membuat Data *Repair Card* 102
Gambar V.28 *Sequence Diagram* Memeriksa Data *Repair Card*..... 102
Gambar V.29 *Sequence Diagram* Mengkonfirmasi Hasil Perbaikan 103
Gambar V.30 *Sequence Diagram* Membuat Data *Part Not Good*..... 104
Gambar V.31 *Sequence Diagram* Membuat Laporan *Monitoring Reject*…….....104
Gambar V.32 *Sequence Diagram* Mencetak Laporan *Monitoring Reject*..... 105
Gambar V.33 *Class Diagram Monitoring Quality Control*..... 106
Gambar V.34 *Deployment Diagram Monitoring Quality Control* 107
Gambar V.35 *Entity Relationship Diagram Monitoring Quality Control*..... 108
Gambar V.36 *Conceptual Data Model Monitoring Quality Control* 109
Gambar V.37 *Wimdows Navigation Diagram Monitoring Quality Control* 115
Gambar V.38 *Interface Form Login* 114
Gambar V.39 *Interface Form* Menu Utama 115
Gambar V.40 *Interface Form* Data Master Pengguna..... 115
Gambar V.41 *Interface Form* Tambah Data Pengguna..... 116
Gambar V.42 *Interface Form* Ubah Data Pengguna 116
Gambar V.43 *Interface Form* Data Master Barang 117
Gambar V.44 *Interface Form* Tambah Data Barang 117

Gambar V.45 *Interface Form* Ubah Data Barang 118
Gambar V.46 *Interface Form* Data Master Pelanggan 118
Gambar V.47 *Interface Form* Tambah Data Pelanggan 119
Gambar V.48 *Interface Form* Ubah Data Barang 119
Gambar V.49 *Interface Form* Data *Check sheet* 120
Gambar V.50 *Interface Form* Tambah Data *Check sheet* 120
Gambar V.51 *Interface Form* Ubah Data *Check sheet* 121
Gambar V.52 *Interface Form* Data *Problem countermeasure* 121
Gambar V.53 *Interface Form* Tambah Data *Problem countermeasure* 122
Gambar V.54 *Interface Form* Ubah Data *Problem Countermeasure* 122
Gambar V.55 *Interface Form* Data *Repair Card* 123
Gambar V.56 *Interface Form* Tambah Data *Repair Card* 123
Gambar V.57 *Interface Form* Ubah Data *Repair Card* 124
Gambar V.58 *Interface Form* Data *Part Not Good* 124
Gambar V.59 *Interface Form* Tambah Data *Part Not Good* 125
Gambar V.60 *Interface Form* Ubah Data *Part Not Good* 125
Gambar V.61 *Interface Form* Data Info *Part Not Good* 126
Gambar V.62 *Interface Form Monitoring Reject* 126

DAFTAR TABEL

Halaman

Tabel II.1 Simbol-simbol *Flowmap* 23
Tabel II.2 *Structure Diagram* 24
Tabel II.3 *Behavior Diagram* 25
Tabel II.4 Simbol-Simbol *Use Case Diagram* 26
Tabel II.5 Simbol-Simbol *Activity Diagram* 28
Tabel II.6 Simbol-Simbol *Sequence Diagram* 29
Tabel II.7 Simbol-Simbol Lain *Sequence Diagram* 30
Tabel II.8 Simbol-Simbol *Class Diagram* 30
Tabel II.9 Simbol-Simbol *Deployment Diagram* 31
Tabel II.10 Simbol-Simbol ERD 33
Tabel II.11 Simbol-Simbol *Conceptual Data Model* 34
Tabel II.12 Contoh Kamus Data 35
Tabel IV.1 Definisi Aktor Sistem Informasi *Monitoring Quality Control* 61
Tabel IV.2 Definisi *Use Case* Sistem Informasi *Monitoring Quality Control* 61
Tabel V.1 Analisis Kebutuhan Fungsional Sistem 51
Tabel V.2 Definisi Aktor Sistem Informasi *Monitoring Quality Control* 67
Tabel V.3 Definisi *Use Case* Sistem Informasi *Monitoring Quality Control* 67
Tabel V.4 *Use Case Description Login* 68
Tabel V.5 *Use Case Description* Mengelola Data Pengguna 69
Tabel V.6 *Use Case Description* Mengelola Data Barang 70
Tabel V.7 *Use Case Description* Mengelola Data Pelanggan 70
Tabel V.8 *Use Case Description* Membuat Data *Check Sheet* 71
Tabel V.9 *Use Case Description* Memeriksa Data *Check Sheet* 72
Tabel V.10 *Use Case Description* Mengkonfirmasi Sampel 72
Tabel V.11 *Use Case Description* Membuat Data *Problem Countermeasure* 73
Tabel V.12 *Use Case Description* Memeriksa Data *Problem Countermeasure* ... 74
Tabel V.13 *Use Case Description* Membuat Data *Repair Card* 74

Tabel V.14 *Use Case Description* Memeriksa Data *Repair Card* 75
Tabel V.15 *Use Case Description* Memperbarui Status Perbaikan 76
Tabel V.16 *Use Case Description* Membuat Data *Part Not Good* 76
Tabel V.17 *Use Case Description* Membuat Laporan *Monitoring Reject* 77
Tabel V.18 *Use Case Description* Mencetak Laporan *Monitoring Reject* 78
Tabel V.19 Tabel tb_pengguna 109
Tabel V.20 Tabel tb_barang 110
Tabel V.21 Tabel tb_pelanggan 110
Tabel V.22 Tabel *Check Sheet* 110
Tabel V.23 Tabel *Problem Countermeasure* 111
Tabel V.24 Tabel *Repair Card* 111
Tabel V.25 Tabel *Part Not Good* 112
Tabel V.26 Tabel Monitoring Reject 112

DAFTAR LAMPIRAN

Halaman

Lampiran A Kode Program L-1
Lampiran B *Black Box Testing* L-9

BAB I
PENDAHULUAN

1.1 Latar Belakang

Faktor teknologi sangat diperhatikan pada perusahaan mengingat teknologi merupakan aspek yang mempengaruhi proses bisnis pada perusahaan. Dengan adanya perkembangan teknologi menjadikan pengolahan data menjadi informasi semakin mudah. Pengolahan data dan informasi yang tersedia secara cepat, tepat dan akurat merupakan hal penting yang dibutuhkan dalam suatu perusahaan. Perusahaan harus menerapkan sistem informasi yang dapat digunakan dalam proses bisnis perusahaan agar mampu bersaing dalam dunia global. Dengan penerapan sistem informasi, secara tidak langsung perusahaan dapat meningkatkan produktivitas dan efisiensi dalam memenuhi informasi mengenai semua informasi yang ada di dalam perusahaan seperti pada bagian Produksi, *Purchasing*, *Inventory*, *Quality Control* dan lainnya.

Pada Divisi *Quality Control* sistem informasi dapat diterapkan untuk mengetahui informasi proses pengendalian kualitas. Divisi *Quality Control* berperan penting dalam menghasilkan produk yang berkualitas dan menjamin agar hasil dari pelaksanaan sesuai dengan rencana yang telah ditetapkan. Dengan adanya sistem informasi pada *Quality Control* dalam suatu perusahaan, perusahaan tersebut akan dapat lebih mudah mengendalikan kualitas dan standar yang ditentukan untuk memenuhi kepuasan pelanggan.

PT Nusa Indah Jaya Utama merupakan perseroan terbatas yang bergerak dalam bidang otomotif yang memproduksi *sparepart* kendaraan bermotor. Kinerja Divisi *Quality Control* PT Nusa Indah Jaya Utama terbilang cukup baik, namun terdapat beberapa kelemahan yaitu masalah manajemen data yang berjalan. Dengan pengolahan data yang masih konvensional, beberapa dokumen mudah hilang dan rusak karena dokumen yang diarsipkan dalam bentuk kertas. Petugas masih mencatat satu per satu produk pada *check sheet.* Pada saat mengubah data, petugas harus mencari secara manual. Hal tersebut cukup memakan waktu dalam

persebaran informasi ke tahap selanjutnya. Proses penghitungan laporan *Monitoring Reject* masih menggunakan *Microsoft Excel* selama proses pengelolaan data sehingga Divisi *Quality Control* membutuhkan waktu yang lama dalam menghitung persentase kualitas produk dan tidak jarang hasil perhitungan kurang akurat. Secara keseluruhan pengelolaan manajemen data yang ada saat ini kurang optimal karena masih belum menggunakan *software* yang mampu mengolah data dengan baik.

Untuk meningkatkan kinerja dalam pengelolaan manajemen data *quality control*, maka perlu diterapkan sistem yang terkomputerisasi dengan cara merancang sistem informasi menggunakan basis data yang baik. Adapun judul Tugas Akhir ini adalah "Sistem Informasi *Monitoring Quality Control Sampling Product* Menggunakan *Code Igniter* 3.1.4 dan *PostgreSQL* Pada Divisi *Quality Control* PT Nusa Indah Jaya Utama".

1.2 Pokok Permasalahan

Permasalahan yang terjadi pada pengelolaan manajemen data *quality control* di PT Nusa Indah Jaya Utama adalah sebagai berikut:

1. Dokumen mengenai kualitas produk mudah hilang dan rusak karena dokumen-dokumen yang beredar dan diarsipkan masih dalam bentuk kertas. Hal ini berisiko terhadap kerusakan dan kehilangan dokumen-dokumen tersebut. Jika terdapat kesalahan penulisan atau pencarian data produk harus melakukan pengecekan secara manual.
2. Penyebaran informasi yang relatif memakan waktu lama karena bagian-bagian yang terkait dengan proses pengendalian kualitas pada setiap harinya harus menulis secara manual dan menyebabkan *input* data membutuhkan waktu yang lama.
3. Tidak akuratnya data kualitas produk karena proses pembuatan laporan *Monitoring Reject* tidak efisien disebabkan sulitnya pencarian dokumen terkait perhitungan jumlah kualitas produk harian maupun bulanan menyebabkan ada beberapa perhitungan yang kurang tepat.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian Tugas Akhir ini yang dilakukan terhadap Divisi *Quality Control* pada PT Nusa Indah Jaya Utama dalam pengelolaan manajemen data *quality control* adalah menganalisis, merancang, dan membangun sistem informasi yang mampu:

1. Mengelola manajemen data *Quality Control* dengan bantuan sistem yang terkomputerisasi.
2. Mempersingkat waktu *input* data sehingga mempercepat proses pengendalian kualitas dalam penyebaran informasi dan dapat mengurangi tingkat kesalahan.
3. Mengelola laporan *Monitoring Reject* dengan menggunakan basis data agar mempercepat penghitungan jumlah kualitas produk harian maupun bulanan.

1.4 Batasan Masalah

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

1. Tempat Praktik Kerja Lapangan ini adalah di Divisi *Quality Control* PT Nusa Indah Jaya Utama yang terletak di Jl. Laskar Dalam No.49, Pekayon Jaya, Bekasi Selatan, Kota Bekasi 17148, Jawa Barat, Indonesia.
2. Jangka waktu untuk Praktik Kerja Lapangan selama satu bulan mulai dari 3 Januari sampai dengan 31 Januari 2019.
3. Ruang lingkup yang dianalisis hanya sebatas menangani proses pengelolaan manajemen data *quality control* yang meliputi pemeriksaan pada sampel produk. Secara detil, ruang lingkup yang diamati adalah proses pengecekan barang jadi, proses *repairing* produk cacat, dan mengetahui jumlah produk NG yang terdeteksi selama satu bulan.

1.5 Manfaat Penelitian

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

1. Bagi Perusahaan
   a. Mempermudah dan mengurangi kesalahan karyawan dalam pengelolaan manajemen data *quality control* pada PT Nusa Indah Jaya Utama.
   b. Mempercepat menghasilkan informasi dalam hal kualitas produk jadi yang telah selesai dicek.
   c. Mampu melaporkan kualitas produk harian maupun bulanan yang sesuai dengan target yang ditargetkan oleh perusahaan sehingga dapat bersaing dengan industri otomotif lainnya.
2. Bagi Mahasiswa
   a. Melatih mahasiswa untuk menganalisis, merancang, dan membangun sebuah sistem informasi.
   b. Memberikan pembelajaran untuk menerapkan ilmu yang diperoleh selama perkuliahan serta menambah wawasan dan pengetahuan, khususnya dalam analisis dan perancangan sistem informasi.

1.6 Sistematika Penulisan

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

BAB I PENDAHULUAN

Pada bab ini memuat latar belakang, pokok permasalahan, tujuan penelitian, batasan masalah, manfaat penelitian, 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, konsep dasar *monitoring product*, serta *tools* pemodelan sistem khususnya *Unified Modelling Language* (UML), *Hypertext Preprocessor* (PHP), *Code Igniter* dan *PostgreSQL*.

BAB III METODOLOGI

Pada bab ini membahas tentang langkah-langkah atau tahapan yang akan dilakukan dalam pemecahan masalah termasuk metodologi pengembangan sistem yang digunakan.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini membahas tentang data yang telah diperoleh berdasarkan pengamatan pada PT Nusa Indah Jaya Utama terutama dalam Sistem *Monitoring Quality Control*

BAB V ANALISIS DAN PEMBAHASAN

Pada bab ini berisi tentang analisis dan perancangan sistem informasi usulan yang berhubungan dengan Sistem Informasi *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini diuraikan beberapa kesimpulan dari hasil penulisan Tugas Akhir dan saran untuk pihak perusahaan dalam berbagai hal yang berhubungan dengan proses bisnis, khususnya di bidang pengolahan sistem informasi.

BAB II
LANDASAN TEORI

2.1 Pengertian Rancang Bangun

Apabila bangunan dianalogikan sebagai suatu sistem, jadi rancang bangun adalah desain sistem. Desain sistem dapat diartikan sebagai berikut ini (Jogiyanto, 2010):

1. Tahap setelah analisis dari siklus pengembangan sistem.
2. Pendefinisian dari kebutuhan-kebutuhan fungsional.
3. Persiapan untuk rancang bangun implementasi.
4. Menggambarkan bagaimana suatu sistem dibentuk.
5. Dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
6. Menyangkut mengonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

Berdasarkan penganalogian dan penjelasan di atas dapat disimpulkan bahwa rancang bangun adalah persiapan perancangan berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen terpisah juga konfigurasi dari komponen perangkat lunak dan perangkat keras dari suatu sistem.

2.2 Konsep Dasar Sistem

Jogiyanto (2010) mengemukakan bahwa sistem adalah kumpulan dari elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem menggambarkan suatu kejadian dan kesatuan yang nyata seperti tempat, benda, serta orang-orang yang ada dan terjadi. Pendapat lain mengatakan sistem merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditunjukan kepada sistem tersebut dan

mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang dihasilkan (Kristanto, 2008).

Dari kedua pandangan di atas, dapat diambil kesimpulan bahwa sistem dapat diartikan sebagai sekumpulan elemen yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi serta bekerja sama untuk mencapai tujuan tertentu.

2.2.1 Tujuan Sistem

Susanto (2013) mengemukakan bahwa tujuan sistem adalah target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai suatu sasaran tanpa mengetahui ciri-ciri atau kriteria darisasaran tersebut kemungkinan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengendalian.

2.2.2 Karakteristik Sistem

Menurut Sutabri (2012) sebuah sistem memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik sistem di antaranya adalah:

1. Komponen Sistem (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan Sistem (*Boundary*)

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

   Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar sistem yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luat tersebut harus selalu dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

   Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsitem ke subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

   Merupakan komponen sistem, yaitu segala sesuatu yang perlu dimasukkan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran (*output*) yang berguna.

6. Pengolah Sistem (*Process*)

   Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

7. Keluaran Sistem (*Output*)

   Merupakan komponen sistem yang berupa berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan.

8. Sasaran sistem (*Objective*)

   Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

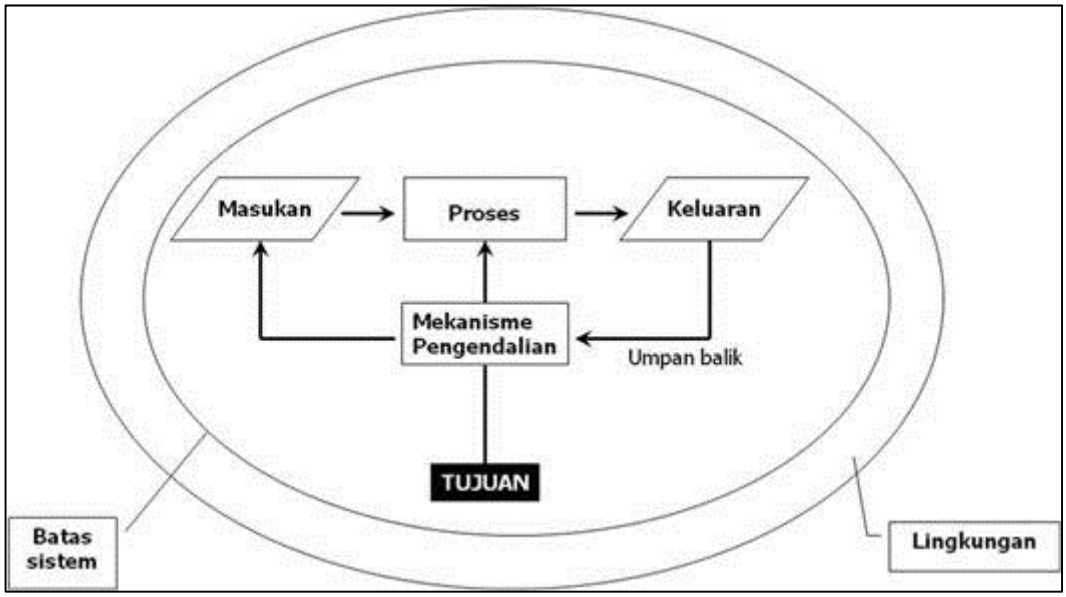
9. Kendali (*Control*)

   Setiap komponen dalam sistem perlu dijaga agar tetap bekerja sesuai dengan peran dan fungsinya masing-masing.

10. Umpan Balik (*Feedback*)

    Diperlukan oleh bagian kendali sistem untuk mengecek terjadinya penyimpanan proses dalam sistem dan mengembalikannya pada kondisi normal.

Sebuah sistem memiliki karakteristik tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem (Sutabri, 2012). Untuk lebih jelasnya dapat dilihat pada Gambar II.1 berikut:



Gambar II.1 Karakteristik Sistem
Sumber: Sutabri (2012)

2.2.3 Klasifikasi Sistem

Menurut Sutabri (2012) sistem dapat diklasifikasikan dalam beberapa sudut pandang. Adapun klasifikasi sistem diantaranya adalah:

1. Sistem Abstrak dan Sistem Fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi, dan sebagainya.

2. Sistem Alamiah dan Sistem Buatan Manusia

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia merupakan sistem yang melibatkan interaksi antara manusia dan mesin, disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*. Sistem informasi akuntansi merupakan contoh *man-machine system* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem Tertentu dan Sistem Tidak Tentu

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi di antara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankan. Sistem tidak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Sistem Terbuka

   Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem lainnya. Karena sistem bersifat terbuka dan terpengaruh oleh lingkungan luarnya, suatu sistem harus mempunyai sistem pengendalian yang baik.

2.3 Hakikat Informasi

Pada hakikatnya, informasi adalah salah satu sumber utama dari perusahaan dan dapat dikelola seperti halnya sumber-sumber lain. Informasi adalah sumber konseptual yang menggambarkan sumber-sumber fisik yang harus dikelola oleh manajer (Rusdiana dan Irfan, 2014).

2.3.1 Pengertian Informasi

Sutabri (2012) menyatakan bahwa informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Menurut Jogiyanto (2010), informasi adalah sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata yang digunakan untuk pengambilan keputusan.

Dari kedua pandangan di atas, dapat diambil kesimpulan bahwa informasi adalah suatu data yang telah diolah sedemikian rupa sehingga dapat tersusun dan terklasifikasi dengan baik sehingga memiliki arti bagi penerimanya, yang selanjutnya menjadi pengetahuan bagi penerima tentang suatu hal tertentu yang membantu pengambilan keputusan

2.3.2 Tujuan Informasi

Tujuan utama informasi sebagai penambah 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).

2.3.3 Klasifikasi Informasi

Sutabri (2012) mengemukakan bahwa informasi merupakan proses lebih lanjut dari data yang sudah memiliki nilai tambah. Informasi dapat diklasifikasikan menjadi 3 bagian, yaitu:

1. Informasi Strategis

   Informasi ini digunakan untuk mengambil keputusan jangka panjang, yang mencakup informasi eksternal, rencana perluasan perusahaan, dan sebagainya.
2. Informasi Taktis

   Informasi ini dibutuhkan untuk mengambil keputusan jangka menengah, seperti informasi tren penjualan yang dapat dimanfaatkan untuk menyusun rencana penjualan.
3. Informasi Teknis

   Informasi ini dibutuhkan untuk keperluan operasional sehari-hari, seperti informasi persediaan stok, retur penjualan dan laporan kas harian.

2.3.4 Kualitas Informasi

Menurut Sutabri (2012) kualitas dari suatu informasi tergantung dari 3 hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timeliness*) dan relevan (*relevance*).

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.

2.4 Konsep Dasar 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 dan Irfan, 2014).

Pratama (2014) mengemukakan bahwa sistem informasi merupakan gabungan dari empat bagian utama yaitu perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur dan Sumber Daya Manusia (SDM) yang terlatih. Keempat bagian utama ini saling berkaitan untuk menciptakan sebuah sistem yang dapat mengolah sebuah data menjadi informasi yang bermanfaat.

2.5 Pengertian *Monitoring*

Monitoring (pemantauan) menurut Mulyono dan Yumari (2017) dalam bukunya merupakan proses pengumpulan dan analisis informasi (berdasarkan indikator yang ditetapkan) secara sistematis dan kontinu tentang kegiatan program sehingga dapat dilakukan tindakan koreksi untuk penyempurnaan program itu selanjutnya. Secara umum *Monitoring* dapat juga diartikan sebagai fungsi manajemen yang dilakukan pada saat kegiatan sedang berlangsung mencangkup aspek-aspek diantaranya:

- Penelusuran pelaksanaan kegiatan dan keluarannya (fokus pada *input*, proses, *output*)
- Pelaporan tentang kemajuan
- Identifikasi masalah-masalah pengelolaan dan pelaksanaan

Menurut Nasir dan Zakiyah (2013) dalam bukunya menjelaskan bahwa *Monitoring* merupakan pengumpulan informasi secara teratur yang akan membantu menjaga agar pekerjaan tetap pada jalurnya dan dapat memperingatkan anda ketika terjadi sesuatu yang salah. Suatu penilaian yang dilaksanakan terus menerus (berkelanjutan) dalam suatu kegiatan. *Monitoring* bertujuan untuk meningkatkan cara dan kinerja kelompok atau organisasi dalam menjalankan kegiatan.

2.5.1 Sistem *Monitoring*

Menurut Wrihatnolo (2006) sistem *monitoring* dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan. Pada pelaksanaannya, *monitoring* dilakukan ketika suatu proses sedang berlangsung. Level kajian sistem *monitoring* mengacu pada kegiatan dalam suatu bagian proses transaksi maupun kegiatan struktural.

2.5.2 Tujuan Sistem *Monitoring*

Terdapat beberapa tujuan sistem *monitoring*. Tujuan sistem *monitoring* dapat ditinjau dari beberapa segi, misalnya segi obyek dan subyek yang dipantau, serta hasil dari proses *monitoring* itu sendiri. Adapun beberapa tujuan dari sistem monitoring (Amsler, 2009) yaitu:

1. Memastikan suatu proses dilakukan sesuai prosedur yang berlaku. Sehingga, proses berjalan sesuai jalur yang disediakan (*on the track*).
2. Menyediakan probabilitas tinggi akan keakuratan data bagi pelaku *monitoring*.
3. Mengidentifikasi hasil yang tidak diinginkan pada suatu proses dengan cepat (tanpa menunggu proses selesai).
4. Menumbuh kembangkan motivasi dan kebiasaan positif pekerja.

2.6 Pengertian Pengendalian Kualitas

Menurut Assauri dikutip Kartika (2013) pengendalian kualitas merupakan usaha untuk mempertahankan mutu atau kualitas dari barang yang dihasilkan, agar sesuai dengan spesifikasi produk yang telah ditetapkan berdasarkan kebijaksanaan pimpinan perusahaan.

Menurut Bakhtiar (2013) pengendalian kualitas dapat diartikan sebagai kegiatan yang dilakukan untuk memantau aktivitas dan memastikan kinerja sebenarnya.

Berdasarkan pengertian-pengertian di atas, maka dapat disimpulkan bahwa pengendalian kualitas adalah 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.

2.6.1 Tujuan Pengendalian Kualitas

Pengendalian kualitas merupakan kegiatan yang terpadu dalam perusahaan untuk menjaga dan mempertahankan kualitas produk yang dihasilkan agar dapat berjalan baik dan sesuai standar yang ditetapkan. Menurut Heizer dan Render (2013) ada beberapa tujuan pengendalian kualitas, yaitu:

1. Peningkatan kepuasan pelanggan.
2. Penggunaan biaya yang serendah-rendahnya.
3. Selesai tepat pada waktunya.

Tujuan pokok pengendalian kualitas adalah untuk mengetahui sampai sejauh mana proses dan hasil produk atau jasa yang dibuat sesuai dengan standar yang ditetapkan perusahaan. Adapun tujuan pengendalian kualitas secara umum menurut Heizer dan Render (2013) sebagai berikut:

1. Produk akhir mempunyai spesifikasi sesuai dengan standar mutu atau kualitas yang telah ditetapkan.
2. Agar biaya desain produk, biaya inspeksi, dan biaya proses produksi dapat berjalan secara efisien.
3. Prinsip pengendalian kualitas merupakan upaya untuk mencapai dan meningkatkan proses dilakukan secara terus-menerus untuk dianalisis agar menghasilkan informasi yang dapat digunakan untuk mengendalikan dan meningkatkan proses, sehingga proses tersebut memiliki kemampuan (kapabilitas) untuk memenuhi spesifikasi produk yang diinginkan oleh pelanggan.

2.6.2 Faktor Pengendalian Kualitas

Menurut Yamit (2013) menyebutkan bahwa faktor-faktor yang mempengaruhi pengendalian kualitas yang dilakukan perusahaan adalah:

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 proses adalah dapat mengurangi produk yang berada di bawah standar seminimal mungkin. Tingkat pengendalian tergantung pada banyaknya produk yang berada di bawah standar.

4. Biaya kualitas

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

2.7 Pengertian Kualitas Produk

Menurut Fandy Tjiptono (2008) Kualitas produk merupakan suatu penilaian konsumen terhadap keunggulan atau keistimewaan suatu produk. Dimensi kualitas produk menurut Fandy Tjiptono (2008) yaitu:

1. *Performance* (Kinerja)

   Merupakan karakteristik operasi dan produk inti (*core product*) yang dibeli. Misalnya kecepatan, kemudahan dan kenyamanan dalam penggunaan.

2. *Durability* (Daya tahan)

   Yaitu jumlah pemakaian suatu produk sebelum produk itu digantikan atau rusak. Semakin lama daya tahannya tentu semakin awet, produk yang awet akan dipersepsikan lebih berkualitas dibanding produk yang cepat ganti.

3. *Conformance to specifications* (Kesesuaian dengan spesifikasi)

   Yaitu sejauh mana karakteristik desain dan operasi memenuhi standar yang telah ditetapkan sebelumnya. Misalnya pengawasan kualitas dan desain, Standar karakteristik operasional adalah kesesuaian kinerja produk dengan standar yang dinyatakan suatu produk. Ini semacam "janji" yang harus dipenuhi oleh produk. Produk yang memiliki kualitas dari dimensi ini berarti sesuai dengan standarnya.

4. *Features* (Fitur)

   Merupakan karakteristik atau ciri-ciri tambahan yang melengkapi manfaat dasar suatu produk. Fitur bersifat pilihan atau option bagi konsumen. Fitur bisa meningkatkan kualitas produk jika kompetitor tidak memiliki fitur tersebut.

5. *Reliability* (Reabilitas)

   Merupakan kemungkinan kecil akan mengalami kerusakan atau gagal pakai. Misalnya pengawasan kualitas dan desain, standar karakteristik operasional kesesuaian dengan spesifikasi.

6. *Aesthetics* (Estetika)

   Merupakan daya tarik produk terhadap panca indera, misalkan bentuk fisik, model atau desain yang artistik, warna dan sebagainya.

7. *Perceived quality* (Kesan kualitas)

   Merupakan persepsi konsumen terhadap keseluruhan kualitas atau keunggulan suatu produk. Biasanya karena kurangnya pengetahuan pembeli akan atribut atau ciri-ciri produk yang akan dibeli, maka pembeli mempersepsikan kualitasnya dari aspek harga, nama merek, iklan, reputasi perusahaan, maupun negara pembuatnya.

8. *Serviceability* (Kemampuan melayani)

   Merupakan kualitas produk ditentukan atas dasar kemampuan diperbaiki: mudah, cepat, dan kompeten. Produk yang mampu diperbaiki tentu kualitasnya lebih tinggi dibandingkan dengan produk yang tidak atau sulit diperbaiki.

2.8 Siklus Deming

Siklus Deming adalah model perbaikan berkesinambungan yang dikembangkan oleh Dr. Edward Deming seorang pionir TQM (Tjiptono, 2008). Siklus ini terbagi atas 4 komponen utama dan dibagi menjadi beberapa langkah yaitu:

a. Mengembangkan rencana perbaikan (*Plan*)
b. Melaksanakan rencana yang dibuat (*Do*)
c. Memeriksa hasil yang dicapai (*Check*)
d. Melakukan penyesuaian bila diperlukan (*Action*)

Model ini dimulai dengan menentukan tujuan kemudian membuat rencana perbaikan dan selanjutnya melaksanakan apa yang telah direncanakan itu. Hasilnya dianalisis dan diketahui penyebabnya. Jika hasilnya tidak sesuai dengan yang diharapkan maka perlu dimodifikasi dengan kembali ke langkah awal sehingga perbaikan yang diharapkan di rencana awal dapat tercapai.

2.8.1 Tujuh Alat Pemecahan Masalah

Tujuh Alat Pemecahan Masalah adalah alat-alat (*tools*) yang dipakai dalam setiap tahapan perbaikan dalam mengelompokkan masalah, menampilkan data sehingga memudahkan analisis data dalam proses penyelesaian masalah dan peningkatan kinerja. Tujuh alat tersebut adalah:

1. Stratifikasi

   Stratifikasi adalah teknik mengelompokkan data ke dalam kategori tertentu, agar data dapat menggambarkan permasalahan secara jelas sehingga kesimpulan mudah diambil (Tjiptono, 2008).

2. Diagram Pareto

   Diagram Pareto diciptakan oleh Alfred Pareto (1848-1923) yang melakukan studi tentang distribusi kekayaan di Eropa. Ia menemukan bahwa hanya sedikit orang yang memiliki banyak uang dan banyak orang yang memiliki sedikit uang. Distribusi kekayaan yang tidak merata ini yang menjadi bagian integral dari teori ekonomi. Diagram ini mengklasifikasi data dengan urutan menurun dari kiri ke kanan (Besterfield, 2009).

3. Diagram Sebab-akibat

   Salah satu alat yang digunakan untuk melihat permasalahan dan penyebab terjadinya permasalahan adalah Diagram Tulang Ikan atau *Fishbone Diagram* atau biasanya dikenal dengan nama Diagram Sebab-Akibat (*Cause and Effect Diagram*). Diagram ini diciptakan oleh Dr Kaoru Ishikawa pada tahun 1943 sehingga disebut juga Ishikawa Diagram (Achyari, 2013).

4. Histogram

   Histogram adalah representasi grafis dari distribusi data. Histogram secara grafis menunjukkan kapabilitas proses (Besterfield,2009).

5. Diagram *Scatter*

   Cara termudah dalam menentukan hubungan sebab-akibat diatara dua variabel adalah dengan cara mem-plot ke dalam Diagram Scatter (Besterfield, 2009). Ada beberapa hubungan (korelasi) antar dua variabel yaitu:

   - Korelasi positif
   - Korelasi negatif
   - Tidak ada korelasi

6. Peta Kendali

   Peta Kendali adalah sebuah alat grafik yang digunakan untuk melakukan pengawasan dari sebuah proses yang sedang berjalan. (Besterfield, 2009).

7. *Check Sheet*

   Check Sheet adalah form yang digunakan untuk mengumpulkan data. Tujuan utama dari check sheet adalah untuk memastikan bahwa data dikumpulkan secara akurat oleh personel dalam melakukan *process control* dan *problem solving* (Besterfield, 2009).

2.9 *System Development Life Cycle* (SDLC)

Menurut Dennis (2015) *System Development Life Cycle* (SDLC) memiliki empat perangkat fase dasar yaitu *planning*, *analysis*, *design*, dan *implementation*. Beberapa proyek dalam membangun sebuah sistem infromasi mungkin dapat berbeda-beda sesuai dengan caranya masing-masing, tetapi hampir semua proyek memiliki elemen dari empat fase tersebut. Masing-masing fase tersebut tersusun

dari beberapa langkah yang menghasilkan *deliverable* atau hasil kegiatan seperti beberapa dokumen spesifik dan *file* yang menjelaskan pemahaman tentang proyek.

Dokumen yang dihasilkan dalam tahap analisis, memberikan ide umum dari suatu bagian dari sistem baru. Dokumen dari *deliverable* ini digunakan sebagai *input* pada tahap *design*, yang kemudian disempurnakan untuk menghasilkan dokumen yang menjelaskan secara detil dari sistem yang dibuat. Dokumen ini akan digunakan dalam tahap implementasi untuk menghasilkan sistem yang sebenarnya. Berikut merupakan penjabaran dari setiap fase dalam SDLC:

1. Perencanaan (*Planning*)

   Dalam tahapan ini, menjelaskan dan mengargumentasikan untuk melanjutkan proyek yang telah dipilih, Rencana kerja yang matang juga disusun untuk menjalankan tahapan-tahapan lainnya. Pada tahap ini ditentukan secara detail rencana kerja yang harus dikerjakan, durasi yang diperlukan masing-masing tahap, sumber daya manusia, perangkat lunak, dokumentasi, perangkat keras, maupun financialdiestimasi. Pembuatan perencanaan ini bukan langkah mudah karena untuk mengestimasi beban kerja dan durasi dari masing-masing tahap dibutuhkan pengalaman yang cukup banyak. Kesalahan pada tahap ini akan mengakibatkan keuntungan yang diperolah tidak maksimal, bahkan bisa rugi. Pada tahapan ini peran manajemen sistem informasi berpengalaman sangat dibutuhkan.

2. Analisis (*Analysis*)

   Tahap kedua, adalah tahap analisis, yaitu berusaha mengenali segenap permasalahan yang muncul pada pengguna dengan mendekomposisi dan merealisasikan komponen-komponen sistem. Tujuan utama dari tahap analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem baru. Menganalisis kebutuhan sebagai bahan dalam membuat spesifikasi di tahapan selanjutnya.

3. Perancangan (*Design*)

   Tahap Perancangan (*design*), mencoba mencari solusi permasalahan yang didapat dari tahap analisis. Tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang nyata untuk diimplementasikan. Jika pada tahapan analisis (*form requirement to specification*), maka tahapan desain adalah (*form specification to implementation*). Jadi, bagaimana pembuatan spesifikasi yang detail untuk bisa diimplementasikan.

4. Implementasi (*Implementation*)

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

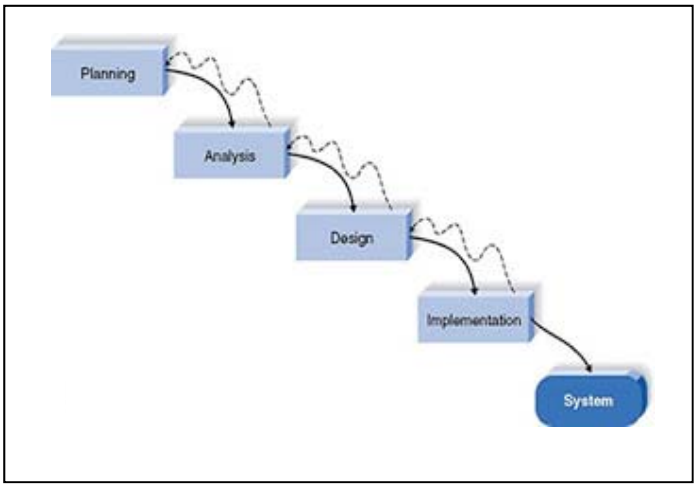
5. Sistem

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

2.10 Metode *Waterfall*

Menurut Dennis (2010) pada penggunaan metode pengembangan *waterfall*, seorang analis dan *user* memproses pengembangan secara bertahap dari satu fase ke fase berikutnya. Setiap fase biasanya berlangsung cukup lama dan setiap fase yang dilewati akan dipresentasikan kepada sponsor untuk mendapatkan persetujuan. Jika sponsor belum menyetujui suatu fase maka pengembangan sistem tidak dapat dilanjutkan ke fase berikutnya. Metodologi ini menyerupai air terjun atau *waterfall* karena bergerak maju dari satu fase ke fase berikutnya secara bertahap seperti cara kerja air terjun, walaupun dalam SDLC memungkinkan

untuk kembali ke fase sebelumnya, namun hal ini akan sulit dilakukan dalam Metode *Waterfall*. Penggambaran Metode *Waterfall* menurut Dennis (2010) dapat dilihat pada Gambar II.2 berikut:



Gambar II.2 Metode *Waterfall* Menurut Dennis
Sumber: Dennis (2010)

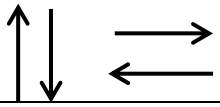
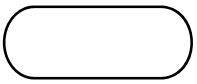
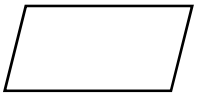
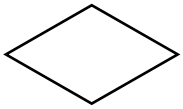
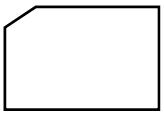
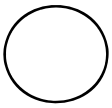
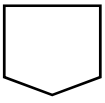
Keuntungan yang didapat dari pengembangan dengan Metode *Waterfall* yaitu, dapat mengidentifikasi kebutuhan sistem jauh sebelum proses pemrograman berlangsung sehingga meminimalisir perubahan yang dapat terjadi pada kebutuhan sistem saat proyek berjalan. Kelemahan dari Metode *Waterfall* adalah desain harus benar-benar ditentukan sebelum pemrograman dimulai dan lamanya proses yang dibutuhkan untuk menyelesaikan sistem.

2.11 *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 ke dalam segmen yang lebih kecil dan menolong dalam menganalisis alternatif lain dalam pengoperasian (Jogiyanto, 2010).

Tujuan dari *flowmap* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas menggunakan simbol-simbol. Simbol-simbol yang terdapat dalam sebuah *flowmap* dijelaskan pada Tabel II.1

Tabel II.1 Simbol-simbol *Flowmap*

Simbol	Nama Simbol	Fungsi
	*Flow*	Untuk menyatakan jalannya arus suatu proses
	*Terminal*	Untuk permulaan atau akhir dari sebuah program
	Predifined Process	Untuk menyatakan penyediaan tempat penyimpanan suatu sebagai pemberi harga awal
	*Input – Output* *Read - Write*	Untuk menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya
	Process	Untuk menunjukan pengolahan yang dilakukan oleh perangkat computer
	*Decision*	Untuk pengkondisian yang akan menghasilkan beberapa kemungkin-an jawaban
	Document	Untuk menyatakan *input* yang berasal dari dokumen dalam bentuk kertas atau *output* dicetak di kertas
	*Punched Card*	Untuk menyatakan *input* berasal dari kartu atau *output* ditulis ke kartu
	*Connector*	Untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
	*Off-line Connector*	Untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda

Sumber : Jogiyanto (2010)

2.12 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Tujuan dari UML adalah untuk menyediakan kosakata yang umum dari istilah-istilah berbasis objek dan teknik yang cukup banyak untuk memodelkan proyek pengembangan sistem dari analisis ke desain.

Menurut Dennis (2012), diagram-diagram yang ada dalam UML terbagi menjadi dua bagian utama yaitu, *Structure Diagram* dan *Behavior Diagram*. *Structure Diagram* biasanya digunakan untuk mepresentasikan data dan hubungan statik yang ada di dalam sebuah sistem informasi. Sedangkan *Behavior Diagram* menyediakan para analis dengan sebuah gambaran hubungan yang dinamis antara instansi atau objek yang merepresentasikan sistem informasi bisnis. Berikut adalah tabel dari jenis diagram yang ada dan fungsi-fungsinya:

1. *Structure Diagram*

   Structure Diagram adalah kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure Diagram* dapat dilihat pada Tabel II.2 berikut:

Tabel II.2 *Structure Diagram*

Nama Diagram	**Fungsi**
Class Diagram	Menggambarkan hubungan antara kelas di dalam sistem.
Object Diagram	Menggambarkan hubungan antara objek di dalam sistem.
Package Diagram	Mengelompokkan elemen UML lainya menjadi satu untuk membentuk tingkat konstruksi yang lebih tinggi.

Tabel II.2 *Structure Diagram* (lanjutan)

Nama Diagram	Fungsi
Deployment Diagram	Menampilkan arsitektur fisik dari suatu sistem. Bisa juga digunakan untuk menunjukkan komponen perangkat lunak yang digunakan di dalam arsitektur fisik.
Component Diagram	Menggambarkan hubungan fisik antara komponen perangkat lunak.
Composite Diagram	Menggambarkan struktur internal dari suatu kelas dan hubungan antara bagian-bagian dari suatu kelas.

Sumber: Dennis (2012)

2. *Behavior Diagram*

Behaviour diagram adalah kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada suatu sistem. *Behavior Diagram* dapat dilihat pada Tabel II.3 berikut:

Tabel II.3 *Behavior Diagram*

Nama Diagram	Fungsi
Activity Diagram	Menggambarkan proses bisnis dari suatu kelas dan alur kegiatan dalam *use case*.
Sequence Diagram	Memodelkan perilaku objek dalam suatu *use case*, berfokus berdasarkan urutan waktu
Communication Diagram	Memodelkan perilaku objek dalam suatu *use case*, berfokus pada komunikasi antara satu set objek yang berkolaborasi dari suatu kegiatan.
Interaction Overview Diagram	Menggambarkan ikhtisar alur kontrol dari suatu proses.
Timing Diagram	Menggambarkan interaksi antara satu set objek dan perubahan keadaan selama perjalanan waktu.

Tabel II.3 *Behavior Diagram* (lanjutan)

Nama Diagram	Fungsi
Behavioral State Machine Diagram	Mengkaji perilaku dari suatu kelas.
Protocol State Machine Diagram	Menggambarkan ketergantungan antara antarmuka yang berbeda dari suatu kelas.
Use Case Diagram	Menangkap kebutuhan bisnis untuk sistem dan menggambarkan interaksi antara sistem dan lingkungannya.

Sumber: Dennis (2012)

2.12.1 *Use Case Diagram*

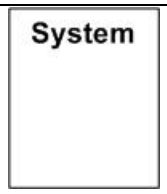
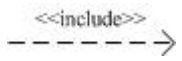
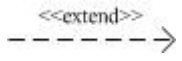
Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana" sistem melakukannya. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Use Case Diagram menggambarkan kebutuhan sistem dari sudut pandang *user*, yaitu proses yang dilakukan oleh sistem dalam melayani *user* yang berinteraksi dengan sistem tersebut. Diagram ini sangat tepat untuk menggambarkan hubungan yang terjadi antara sistem dengan penggunanya. Simbol-simbol *Use Case Diagram* dapat dilihat pada Tabel II.4.

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

Simbol	Nama	Deskripsi
	Actor	- Seseorang atau sistem yang mendapatkan keuntungan dari sistem. - Diletakkan di luar batas sistem.
	Use Case	- Merepresentasikan bagian utama dari sistem secara fungsional. - Diletakkan di dalam batas sistem.

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

Simbol	Nama	Deskripsi
	*System Boundary*	- Nama dari sistem terdapat di dalam atau di atas. - Merepresentasikan ruang lingkup dari sistem.
	Association	- Menghubungkan suatu aktor dengan *use case* dengan apa interaksinya.
	*Include*	- Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya.
	*Extend*	- Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan ini.
	Generalization	- Hubungan generalisasi dan spesialisasi antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.

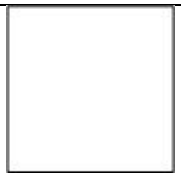
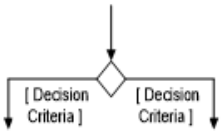
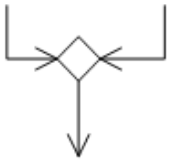
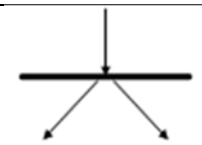
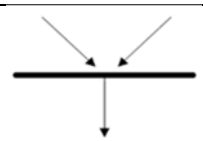
Sumber: Dennis (2012)

2.12.2 *Activity Diagram*

Digunakan untuk model perilaku dalam independen proses bisnis objek. Dalam banyak hal, *Activity Diagram* dapat dipandang sebagai diagram aliran data yang canggih yang digunakan dalam hubungannya dengan analisis terstruktur. Namun, tidak seperti aliran data diagram, *Activity Diagram* termasuk notasi yang membahas pemodelan paralel, kegiatan bersamaan, dan proses (Dennis, 2012).

Yang perlu diperhatikan di sini adalah bahwa *Activity Diagram* menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, tetapi aktivitas yang dapat dilakukan oleh sistem. Adapun simbol-simbol *Activity Diagram* dapat dilihat pada Tabel II.5 berikut:

Tabel II.5 Simbol-Simbol *Activity Diagram*

Simbol	Nama	Deskripsi
	Initial State	Menggambarkan awal dari serangkaian aktivitas.
	Final State	Menghentikan semua aliran kontrol dan aliran objek dalam sebuah aktivitas.
	Activity	Merepresentasikan serangkaian aktivitas yang terjadi.
	Control Flow	Menunjukkan urutan eksekusi.
	*Swimlane*	Memisahkan individu atau objek yang bertanggung jawab untuk melaksanakan aktivitas.
	*Decision*	Merepresentasikan sebuah kondisi tes untuk memastikan bahwa aliran kontrol atau aliran objek hanya menuju satu jalur.
	*Merge*	Membawa kembali jalur keputusan bersama yang berbeda yang dibuat oleh *Decision*.
	*Fork*	Membagi perilaku menjadi satu set arus paralel dalam satu kondisi waktu yang sama.
	*Join*	Membawa kembali bersama satu set arus paralel yang sebelumnya dibentuk oleh *Fork*.

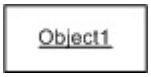
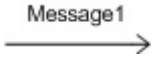
Sumber: Dennis (2012)

2.12.3 *Sequence Diagram*

Sequence Diagram adalah salah satu dari dua jenis diagram interaksi. Mereka menggambarkan objek-objek yang berpartisipasi dalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu untuk satu *use case*.

Sebuah *Sequence Diagram* adalah model dinamis yang menunjukkan urutan eksplisit pesan yang lewat di antara objek dalam interaksi yang didefinisikan. Karena urutan diagram menekankan pemesanan berbasis waktu kegiatan yang terjadi di antara set objek, mereka sangat membantu untuk memahami spesifikasi *real-time* dan kompleks menggunakan kasus (Dennis, 2012). Adapun simbol-simbol *Sequence Diagram* dapat dilihat pada Tabel II.6 berikut:

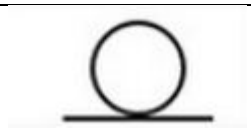
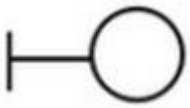
Tabel II.6 Simbol-Simbol *Sequence Diagram*

Simbol	Nama	Deskripsi
	Actor	Orang atau sistem yang berasal dari luar sistem yang memberikan manfaat yang berpartisipasi secara berurutan dengan mengirim dan/atau menerima pesan.
	*Object*	Berpartisipasi secara berurutan dengan mengirim dan/atau menerima pesan yang ditempatkan di atas diagram.
	Lifeline	Menyatakan kehidupan suatu objek.
	*Message*	Menyampaikan informasi dari satu objek ke objek yang lain.
	Execcution Occurence	Menandakan ketika suatu objek mengirim dan/atau menerima pesan.
X	*Object Destruction*	Ditempatkan pada akhir dari suatu *object lifeline* untuk menunjukkan bahwa itu akan keluar dari eksistensi.

Sumber: Dennis (2012)

Selain simbol-simbol diatas *sequence diagram* memiliki *class* dengan fungsi yang masing-masing berbeda, berikut tabel II.7 merupakan *class* yang terdapat dalam *sequence diagram* menurut Richardson dan Thies:

Tabel II.7 Simbol-Simbol Lain *Sequence Diagram*

Simbol	Nama	Deskripsi
	*Entity Class*	Menggambarkan hubungan kegiatan yang akan dilakukan
	*Boundary Class*	Menggambarkan sebuah gambaran dari form
	Control Class	Menggambarkan penghubung antara boundary dengan tabel

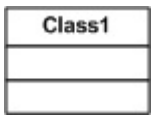
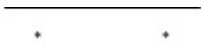
Sumber: Richardson dan Thies (2013)

2.12.4 *Class Diagram*

Class Diagram adalah ilustrasi antara kelas yang dimodelkan di dalam sistem. 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, 2012).

Simbol-simbol *Class Diagram* dapat dilihat pada Tabel II.8 berikut:

Tabel II.8 Simbol-Simbol *Class Diagram*

Simbol	Nama	Deskripsi
	*Class*	Mewakili sejenis orang, tempat, atau suatu hal yang mana sistem akan butuhkan untuk menyimpan informasi.
	*Association*	Merepresentasikan sebuah hubungan antara beberapa kelas atau sebuah kelas dan dirinya sendiri.

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

Simbol	Nama	Deskripsi
	Generalization	Merepresentasikan sebuah hubungan antara beberapa kelas.
	Aggregation	Menggambarkan suatu *class* terdiri dari *class* lain atau suatu *class* adalah bagian dari *class* lain.
	Composition	Merepresentasikan sebuah bagian dari hubungan antara beberapa kelas atau sebuah kelas dan dirinya sendiri.

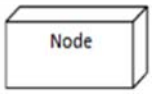
Sumber: Dennis (2012)

2.12.5 *Deployment Diagram*

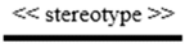
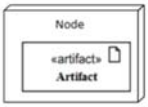
Deployment Diagram digunakan untuk mewakili hubungan antara komponen-komponen *hardware* yang digunakan dalam infrastruktur fisik sistem informasi. Misalnya, ketika mengarang suatu sistem informasi terdistribusi yang akan menggunakan jaringan luas, *Deployment Diagram* dapat digunakan untuk menunjukkan 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, 2012).

Simbol-simbol *Deployment Diagram* dapat dilihat pada Tabel II.9 berikut:

Tabel II.9 Simbol-Simbol *Deployment Diagram*

Simbol	Nama	Deskripsi
	*Node*	Menggambarkan sumber daya komputasi dalam sebuah sistem.
	Artifact	Menggambarkan spesifikasi dari *software* atau *database*, misalnya *file* sumber, tabel *database*, *executable file*.

Tabel II.9 Simbol-Simbol *Deployment Diagram* (lanjutan)

Simbol	Nama	Deskripsi
	*Communication Path*	Menggambarkan hubungan antara dua *node* untuk bertukar pesan.
	*Node With A Deployed Artifact Tiplicity*	Menggambarkan *artifact* yang ditempatkan pada *node* fisik. Mendukung pemodelan distribusi perangkat lunak melalui jaringan.

(Sumber: Dennis, 2012)

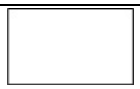
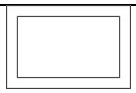
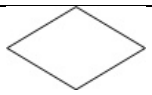
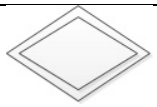
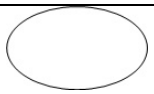
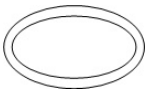
2.13 Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan oleh perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data dapat diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan *Database Management System* (DBMS) (Puspitawati dan Anggadini, 2014).

2.14 *Entity Relationship Diagram* (ERD)

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 dan Jeffrey, 2007). Simbol-simbol ERD dapat dilihat pada Tabel II.10 berikut:

Tabel II.10 Simbol-Simbol ERD

Simbol	Nama	Deskripsi
	*Strong Entity*	Setiap hal dunia nyata (orang, tempat, objek, konsep, aktivitas).
	*Weak Entity*	Bergantung pada *Strong Entity*.
	*Relationship*	Sebuah hubungan antara dua atau lebih entitas.
	*Identifying Relationship*	Menghubungkan *Strong Entity* dengan *Weak Entity*.
	*Attribute*	Properti atau karakteristik dari sebuah tipe entitas.
	*Multivalued Attribute*	Karakteristik tipe entitas yang dapat memiliki nilai lebih dari satu.

Sumber: Jeffrey A. Hoffer (2007)

Menurut Mulyani (2016), ERD berfungsi untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu:

a. Entity(Entitas)

   Entity merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entity ini biasanya digambarkan dengan persegi panjang.

b. Atribut

   Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

c. Hubungan/Relasi

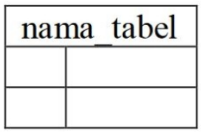
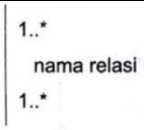
Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut: Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Satu ke satu (*One to one*) Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.
2) Satu ke banyak (*One to many*) Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitasB dapat berhubungan dengan satu entitas pada himpunan entitas A.
3) Banyak ke banyak (*Many to many*) Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

2.15 *Conceptual Data Model* (CDM)

Menurut Rosa dan Shalahuddin (2018), *Conceptual Data Model* (CDM) atau model konsep data merupakan konsep yang berkaitan dengan pandangan pemakai terhadap data yang disimpan dalam basis data. CDM sudah di buat dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan relasi antartabel untuk keperluan implementasi ke basis data. CDM merupakan hasil penjabaran dari ERD. Berikut adalah simbol-simbol yang ada pada CDM:

Tabel II.11 Simbol-Simbol *Conceptual Data Model*

Simbol	Nama	Deskripsi
	Entitas/tabel	Entitas atau tabel yang menyimpan data dalam basis data.
	Relasi	Relasi antar tabel yang terdiri atas nama relasi dan *multiplicity*

Sumber: Rosa dan Shalahuddin (2018)

2.16 *Physical Data Model* (PDM)

Menurut Rosa dan Shalahuddin (2018), *Physical Data Model* (PDM) adalah jenis model data yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik.

2.17 Kamus Data

Menurut Jogiyanto (2010), kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, diharapkan analisis sistem dapat mendefinisikan data yang mengalir dalam sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem. Adapun contoh dari kamus data dapat dilihat pada Tabel II.12 berikut:

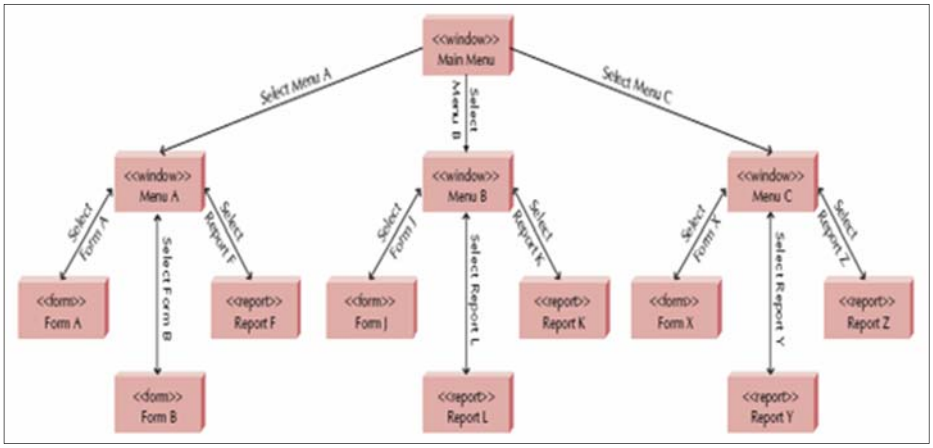
Tabel II.12 Contoh Kamus Data

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

Sumber: Jogiyanto (2010)

2.18 *Windows Navigation Diagram* (WND)

Menurut Dennis (2015), desain struktur navigasi mendefinisikan komponen dasar antarmuka dan bagaimana mereka bekerja sama untuk menyediakan fungsionalitas kepada pengguna. *Windows Navigation Diagram* (WND) digunakan untuk menunjukkan bagaimana semua layar, bentuk, dan laporan yang digunakan oleh sistem terkait dan bagaimana pengguna berpindah dari satu menu ke menu lainnya. Kebanyakan sistem memiliki beberapa WND, satu untuk setiap bagian utama dari sistem. Adapun contoh dari *Windows Navigation Diagram* dapat dilihat pada Gambar II.3 berikut:



Gambar II.3 Contoh *Windows Navigation Diagram*
Sumber: Dennis (2015)

2.19 *Hypertext Preprocessor* (PHP)

Menurut Anhar (2010), PHP singkatan dari *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. 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.

Kode-kode PHP memiliki tata aturan, yaitu diawali dengan tanda <?php dan diakhiri dengan tanda ?>. Tiap akhir baris harus selalu diberi tanda titik koma (;). PHP bersifat *case sensitive*, artinya penulisan huruf besar dan kecil pada kode PHP sangat berpengaruh.

Menurut Anhar (2010), beberapa kelebihan PHP dari bahasa pemrograman lain adalah sebagai berikut:

1. PHP adalah bahasa *scripting* yang memiliki referensi yang banyak dan sederhana sehingga mudah untuk dimengerti pemula.
2. *Web server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.

4. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin seperti Linux, Unix, Macintosh, dan Windows serta dapat dijalankan secara *runtime* melalui *console* dan dapat menjalankan perintah-perintah sistem.
6. PHP juga dilengkapi dengan berbagai macam pendukung lain seperti *support* langsung ke berbagai macam *database* yang populer, misalnya Oracle, PostgreSQL, dan lain-lain.

2.20 *PostgreSQL*

Menurut Adil (2017) *PostgreSQL* atau sering disebut *Postgres* merupakan salah satu dari sejumlah basis data besar yang menawarkan skalabilitas, keluwesan, dan kinerja yang tinggi. Penggunaannya begitu meluas diberbagai *platform* dan didukung oleh banyak bahasa pemrograman. Bagi masyarakat IT (Teknologi Informasi) di Indonesia, *Postgres* sudah digunakan untuk berbagai aplikasi seperti *web, billing system,* dan sistem informasi besar lainnya.

SQL di *Postgres* tidaklah seperti yang kita temui pada RDBMS umumnya. Perbedaan penting antara *Postgres* dengna sistem relasional standar adalah arsitektur *Postgres* yang memungkinkan *user* untuk mendefinisikan sendiri SQL-nya, terutama pada pembuatan *function* atau biasa disebut sebagai *stored procedure*. Hal ini dimungkinkan karena informasi yang disimpan oleh *Postgres* bukan hanya tabel dan kolom, melainkan tipe, fungsi, metode akses, dan banyak lagi yang terkait dengan tabel dan kolom tersebut. Semuanya terhimpun dalam bentuk *class* yang bisa diubah *user*.

Dengan demikian, *Postgres* memiliki dua kekuatan besar yaitu *source code* dan arsitektur yang luwes, tentunya disamping *feature* penting lainnya seperti dokumentasi yang lengkap, dan sebagainya. Disamping itu, *Postgres* juga didukung oleh banyak antar muka ke berbagai bahasa pemrograman, seperti C++, Java, Perl, PHP, Python dan Tcl.

2.21 *Code Igniter*

Menurut Widodo (2013) "*Codeigniter* merupakan *framework* PHP yang diklaim memiliki eksekusi tercepat dibandingkan dengan *framework* lainnya. *Codeigniter* bersifat *open source* dan menggunakan model basic MVC (*Model View Controller*), yang merupakan model konsep *modern framework* yang digunakn saat ini". Sedangkan menurut Purbadian (2016) "*Codeigniter* adalah sebuah *framework* yang dibuat menggunakan bahasa pemoraman PHP yang bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*".

Berdasarkan pengertian diatas dapat ditarik kesimpulan bahwa *Codeigniter* adalah aplikasi bersifat *open source* yang merupakan *framework* PHP dengan eksekusi tercepat dan menggunakan model basic MVC (*Model View Controller*) bertujuan untuk memudahkan para *programmer web* untuk membuat atau mengembangkan aplikasi berbasis *web*.

2.22 *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 daripada Metode *White-Box* (Pressman dalam Subri, 2015).

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.

BAB III
METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Menurut Sugiyono (2013), Metode penelitian pada dasarnya merupakan cara ilmiah untuk mendapatkan data dengan tujuan dan kegunaan tertentu. Berdasarkan hal tersebut terdapat empat kata kunci yang perlu diperhatikan yaitu cara ilmiah, data, tujuan dan kegunaan. Menurut Darmadi (2013), Metode penelitian adalah suatu cara ilmiah untuk mendapatkan data dengan tujuan kegunaan tertentu. Cara ilmiah berarti kegiatan penelitian itu didasarkan pada ciri-ciri keilmuan yaitu rasional, empiris, dan sistematis. Berdasarkan pemaparan di atas dapat disimpulkan bahwa metode penelitian adalah suatu cara ilmiah untuk memperoleh data dengan tujuan dan kegunaan tertentu.

3.2 Jenis Dan Sumber Data

Menurut Arikunto (2010) sumber data dalam penelitian adalah subjek dari mana data dapat diperoleh. Dalam suatu penelitian terdapat dua sumber data yang dipakai, data tersebut adalah sebagai beikut :

1. Data Primer

   Data primer adalah data yang diperoleh secara langsung melalui pengamatan langsung terhadap sistem yang sedang berjalan dan wawancara dengan karyawan dari Divisi *Quality Control* sebagai sumber informasinya. Dalam penelitian ini data yang diperoleh berupa alur proses pelaporan kualitas yang berjalan.

2. Data Sekunder

   Data sekunder adalah data yang diperoleh secara tidak langsung dari sumber yang sudah ada melalui media perantara, buku-buku, internet, dan referensi lainnya. Dalam penelitian ini data tersebut berupa data umum perusahaan seperti sejarah perusahaan dan struktur organisasi perusahaan.

3.3 Metode Pengumpulan Data

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

1. Studi Lapangan

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

   a. Observasi

      Observasi adalah melakukan pengamatan langsung pada sistem yang berjalan di Divisi *Quality Control* pada PT Nusa Indah Jaya Utama, yang kemudian dilakukan pecatatan terhadap informasi yang dianggap perlu. Hasil pengamatan yang dilakukan menjadi landasan dalam melakukan pengembangan sistem yang akan dibuat.

   b. Wawancara

      Wawancara adalah metode pengumpulan data melalui tanya jawab secara langsung kepada narasumber mengenai segala hal yang diperlukan untuk menunjang penelitian yang dilakukan. Pihak yang diwawancarai adalah karyawan di Divisi *Quality Control*.

2. Studi Kepustakaan

   Studi kepustakaan adalah metode pengumpulan data yang dilakukan dengan membaca buku dan literatur dalam lingkup penelitian maupun di luar lingkup penelitian yang berhubungan dengan judul dan permasalahan sehingga dapat menunjang penelitian.

3.4 Kerangka Penelitian

Kerangka penelitian merupakan gambaranlangkah-langkah kegiatan mulai dari awal hingga akhir. Adapun langkah-langkah atau tahapan dalam kerangka penelitian tersebut adalah sebagai berikut:

1. Studi Pendahuluan

   Pada tahap ini merupakan tahapan awal yang dilakukan untuk memulai penelitian. Dengan melakukan studi lapangan meliputi pengamatan langsung pada Divisi *Quality Control,* wawancara dengan divisi terkait dan melakukan studi pustaka.

2. Identifikasi Masalah

   Pada tahap ini peneliti melakukan penelitian dengan menganalisis sistem pengelolaan manajemen data *quality control* yang sedang berjalan sehingga dapat diidentifikasi permasalahan yang ada pada proses pengendalian kualitas di PT Nusa Indah Jaya Utama. Permasalahan yang teridentifikasi yaitu masih menggunakan sistem konvensional dalam mencatat data kualitas produk, persebaran informasi yang relatif lama, tidak jarang terjadi kesalahan data pata laporan *Monitoring Reject* karena sistem yang tidak terintegrasi.

3. Tujuan Penelitian

   Pada tahap ini dimaksudkan agar tujuan yang akan dilakukan oleh peneliti dapat tercapaiyaitu:

   4. Menganalisis Sistem Informasi Pengelolaan Manajemen Data *Quality Control* yang mampu berjalan dengan bantuan sistem yang terkomputerisasi.
   5. Merancang Sistem Informasi *Quality Control* pada PT Nusa Indah Jaya Utama untuk menghasilkan data yang akurat sehingga mempercepat dalam proses pengendalian kualitas dalam persebaran informasi dan dapat mengurangi tingkat kesalahan.
   6. Membangun Sistem Informasi yang mampu mengelola laporan *Monitoring Reject* dengan menggunakan basis data agar mempercepat penghitungan persentase kualitas produk harian maupun bulanan.

4. Batasan Masalah

   Pada tahap ini bertujuan untuk membatasi masalah yang ada untuk dapat terarah dan lebih spesifik mengenai pemecahan masalahnya.Berikut merupakan batasan-batasan masalah:

a. Ruang lingkup yang dianalisis hanya sebatas menangani proses pengelolaan manajemen data *quality control* yang meliputi pemeriksaan pada sampel produk. Meliputi proses pengecekan barang jadi, proses perbaikan produk cacat, dan mengetahui jumlah produk NG yang tedeteksi selama satu bulan. Dilakukan selama satu bulan, mulai dari 3 Januari sampai dengan 31 Januari 2019.

5. Pengembangan Sistem

Pengembangan sistem yang digunakan dalam penelitian Tugas Akhir ini adalah menggunakan metode waterfall. Metode *waterfall* ini memiliki beberapa tahapan yaitu:

a. Perencanaan (*Planning*)

Pada tahap ini akan dilakukan perencanaan untuk membangun sistem informasidan menentukan bagaimana akan membangunnya dengan mengumpulkan data kemudian akan diolah.

b. Analisis (*Analysis*)

Pada tahap ini akan dilakukan analisis kebutuhan sistem dengan melakukan pengamatan, wawancara dan membuat analisis permasalahan.

c. Desain (*Design*)

Pada tahap ini akan membuat pemodelan sistem dengan menggunakan:

1) Pemodelan sistem menggunakan UML
   a) *Usecase diagram*
   b) *Activity diagram*
   c) Sequence diagram
   d) *Class diagram*
   e) *Deployment diagram*
2) Pemodelan data
   a) *Entity Relationship Diagram* (ERD)
   b) Kamus data
3) Desain sistem
   a) Windows Navigation Diagram (WND)
   b) Perancangan antar muka

d. Implementasi (*Implementation*)

Pada tahap ini akan mengembangkan dan membangun sebuah sistem dengan rancangan menggunakan *software CodeIgniter* 3.1.4, *database PostgreSQL*, dan kemudian akan diimplementasikan menggunakan metode *Total Quality Management* (TQM).

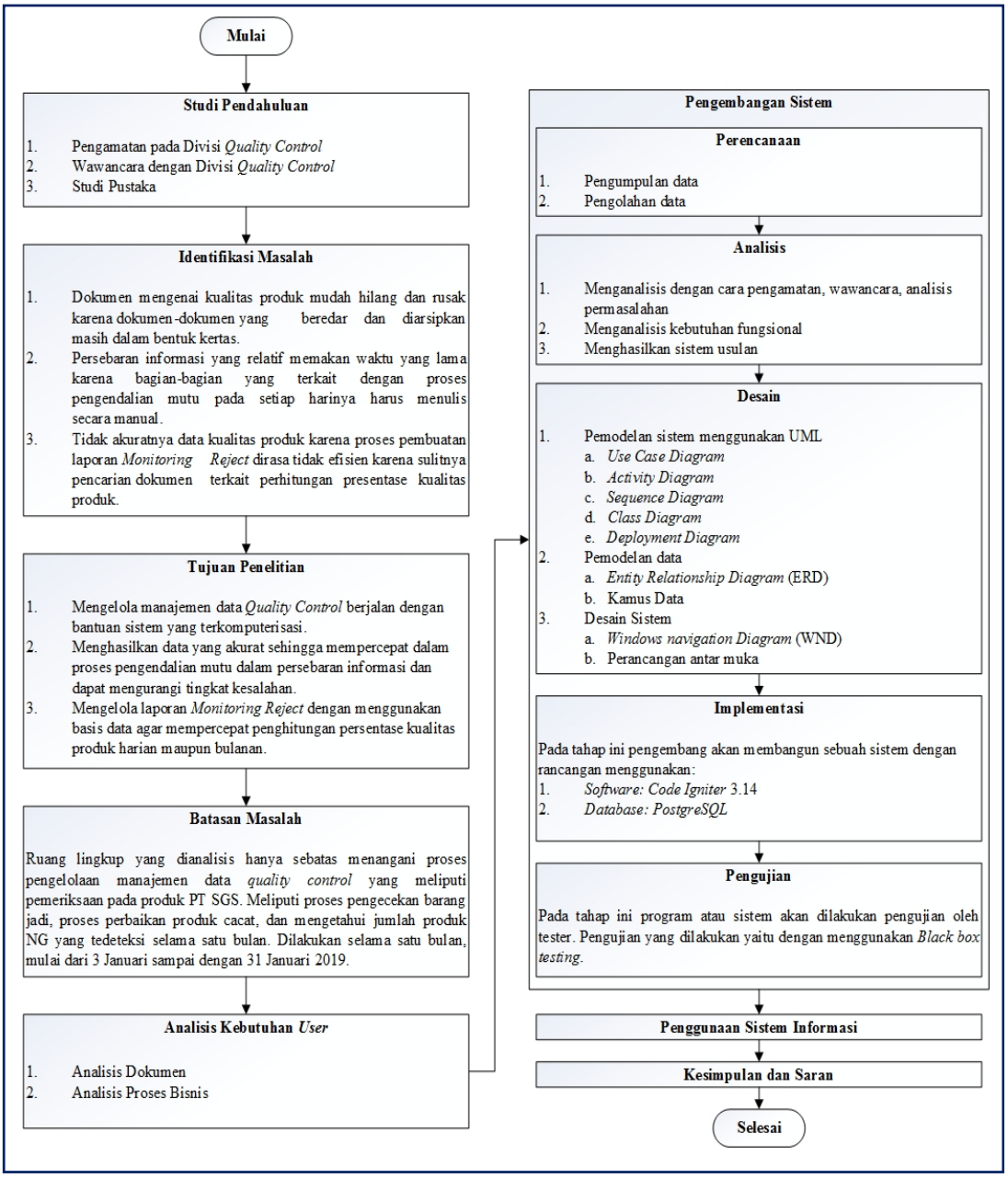
e. Pengujian (*Testing*)

Pada tahap terakhir dalam metode *waterfall* ini adalah pengujian (*testing*), digunakan untuk menentukan apakah sistem yang dibuat sesuai dengan kebutuhan pengguna atau belum. Pada tahap ini program atau sistem akan dilakukan pengujian oleh tester. Pengujian yang dilakukan yaitu dengan menggunakan *Black box testing*.

6. Kesimpulan dan Saran

Pada tahap ini bertujuan untuk membuat kesimpulan yang berisi tentang hasil penelitian atau pengembangan sistem yang ada. Sedangkan saran berisi tentang masukan terhadap pengembangan sistem selanjutnya untuk dapat membangun sistem pada PT Nusa Indah Jaya Utama.

Berikut gambar III.1 merupakan langkah-langkah dalam kerangka penelitian:



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

BAB IV
PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Sekilas Perusahaan

PT Nusa Indah Jaya Utama berdiri pada tahun 1974 didirikan oleh bapak H. M. Kusnadi dengan nama CV Nusa Indah dan bergerak dalam penanganan limbah besi dari kapal-kapal tua yang berlokasi di Jakarta. Pada tahun 1976 CV Nusa Indah bekerja sama dengan PT Mitsubishi Kramayudha Motors (PT MKM) dalam penanganan limbah (*Scrap Stamping*) mobil *Colt* yang diproduksi pertama di Indonesia dan bergerak aktif dalam pengadaan *manpower* untuk PT MKM yang berada di Jl. Rawa Teratai, Pulogadung, Jakarta Timur. Pada tahun 1978 telah menjadi rekanan tetap PT MKM khususnya di bagian penanganan pengerjaan *part* komponen mobil *Colt* yang bertempat di *workshop*. Pada tahun 1980 CV Nusa Indah mendapatkan pekerjaan tambahan dalam pembuatan *packing part* komponen dalam bentuk CKD (*Complete Knock Down*) yang dikirim untuk *supplier* PT MKM. Pada tahun 1987 PT MKM membuat lokaslisasi komponen untuk semua jenis kendaraannya (mobil L300, *Colt Diesel*, dan Fuso).

Pada tahun 1991 CV Nusa Indah pindah lokasi di Jl. Pulogebang dengan luas area 1.350 m^2 dan menjadi *home industry* dalam pengerjaan komponen Mitsubishi serta bekerja sama dengan PT Daihatsu Motor Indonesia. Pada tahun 2007 berubah statusnya menjadi PT Nusa Indah Jaya Utama dan berpindah alamat di Jl. Laskar 49 Pekayon Jaya, Bekasi Selatan dengan luas tanah sebesar 3800m^2. Pada tahun 2013 PT Nusa Indah Jaya Utama mulai menggunakan mesin *medium press* dengan kapasitas 110-160 ton yang didatangkan dari negara China dan Korea dalam mengerjakan pesanan PT Garmak Motor, mobil *Chevrolet* serta motor TVS. Pada tahun 2014 PT Nusa Indah Jaya Utama mulai menggunakan mesin *high press* dengan kapasitas 200-315 ton dan memperluas hubungan pekerjaan dengan menjadi *vendor* PT Gemala Kempa Daya, PT Fuji Teknika Indonesia dan PT Inti Pantja Press Industri.

Berdasarkan keputusan Menteri Hukum dan Hak Asasi Manusia Republik Indonesia CV Nusa Indah berubah menjadi PT Nusa Indah Jaya Utama dan juga merubah anggaran dasar perseroan dengan Nomor: AHU-12765.AH.01.02.Tahun 2012. Kemudian Surat Izin Usaha Perdagangan (SIUP) Menengah untuk PT Nusa Indah Jaya Utama sendiri keluar pada tahun 2013 dengan Nomor: 510/656-BPPT/PM/IX/2013.

4.2 Profil Perusahaan

Profil perusahaan merupakan sebuah media atau alat yang bermanfaat untuk memberikan informasi secara lengkap mengenail detail perusahaan. Fungsi profil perusahaan antara lain memuat informasi detail mengenai perusahaan, berperan sebagai media representasi dan *tools* marketing, pelengkap dalam proposal, serta berperan sebagai media untuk branding dan publikasi.

Adapun profil usaha lebih rinci sebagai berikut:

Nama Perusahaan	: PT Nusa Indah Jaya Utama
Nama Pemilik	: H. M. Kusnadi
Jenis Usaha	: *Stamping and Manufacturing* komponen otomotif
Jumlah Karyawan	: 83 orang
Alamat	: Jl. Laskar 49 Pekayon Jaya, Bekasi Selatan
Telepon	: (021) 82411782
Tahun Berdiri	: 1974, berubah menjadi PT pada tahun 2007
Nomor TDP	: 102614611073
Email	: pt.niju@gmail.com

4.3 Logo Perusahaan

Logo adalah lambang dari sebuah perusahaan dan logo juga merupakan simbol yang memberi penjelasan tentang citra dari perusahaan atau yang lebih dikenal dengan sebutan *branding* atau *corporate identity*. Dengan memiliki logo perusahaan menempatkan dirinya secara berbeda dalam masyarakat dan konsumen. Logo dari PT Nusa Indah Jaya Utama dapat dilihat pada Gambar IV.1 berikut:

Gambar IV.1 Logo PT Nusa Indah Jaya Utama
Sumber : PT Nusa Indah Jaya Utama (2019)

4.4 Visi dan Misi Perusahaan

Visi merupakan rangkaian kalimat yang menyatakan cita-cita atau impian sebuah organisasi atau perusahaan yang ingin dicapai di masa depan. Adapun visi PT Nusa Indah Jaya Utama adalah "Menjadi perusahaan penyedia jasa *stamping and manufacturing* untuk komponen otomotif, baik roda dua maupun roda empat".

Sedangkan misi merupakan rangkaian kalimat yang menyatakan tujuan atau alasan eksistensi organisasi, yang memuat apa yang disediakan oleh perusahaan kepada masyarakat, baik berupa produk ataupun jasa. Adapun misi PT Nusa Indah Jaya Utama adalah membangun perusahaan yang mampu menciptakan lapangan kerja berkualitas bagi sebanyak mungkin rakyat Indonesia dengan mengusung nilai-nilai berikut ini:

1. Pengembangan kompetensi karyawan secara berkelanjutan
2. Mengupayakan pertumbuhan finansial, intelektual dan citra perusahaan yang konsisten serta melakukan investasi kembali ke dalam bisnis yang dijalankan
3. Mempertahankan standar kode etik yang tinggi dalam aktivitas bisnis.

4.5 Struktur Organisasi

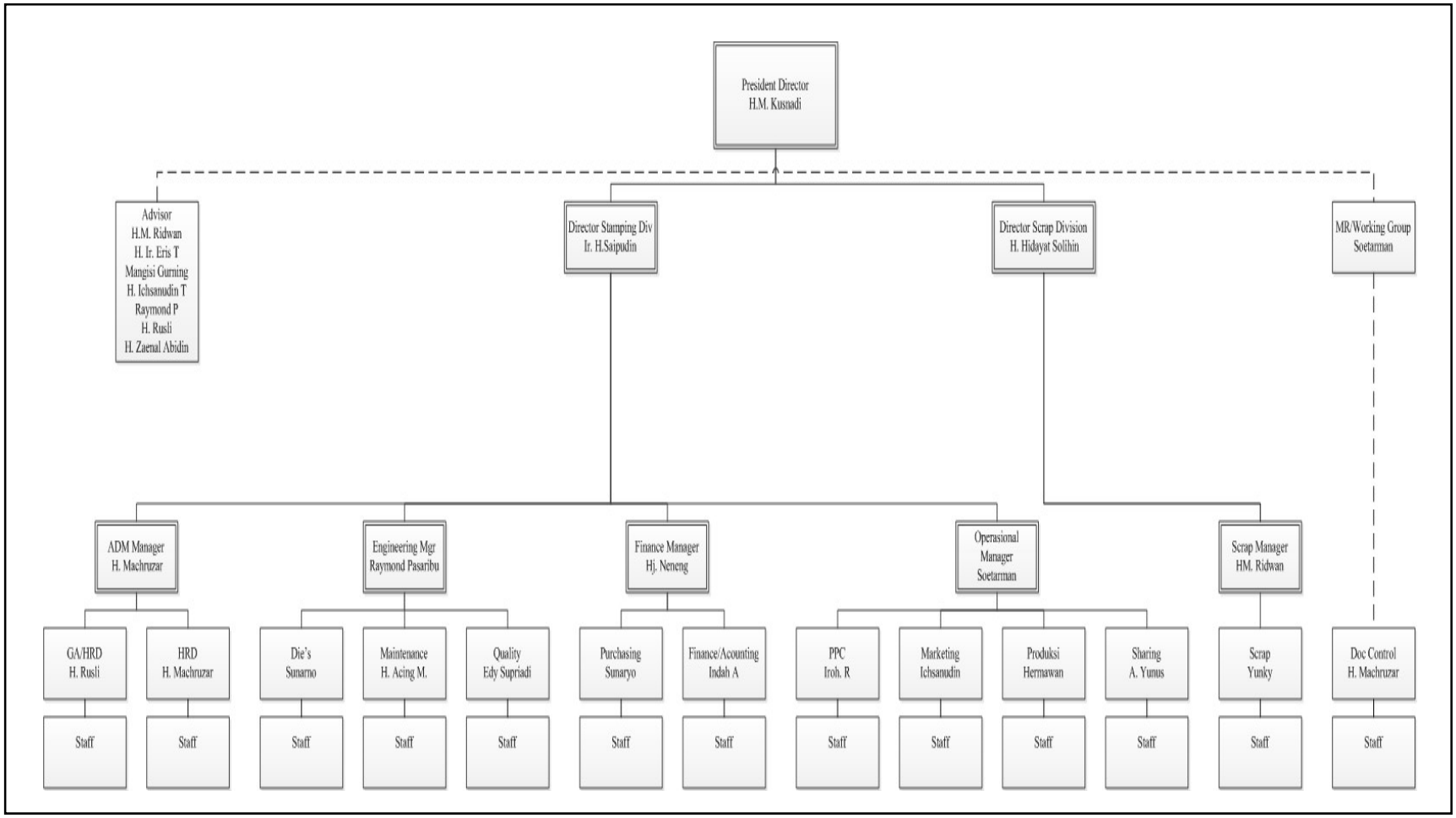
Struktur organisasi adalah suatu susunan unit-unit kerja dalam sebuah organisasi. Struktur organisasi menunjukan bahwa adanya pembagian kerja dan bagaimana fungsi atau kegiatan-kegiatan berbeda yang dikoordinasikan. Dan selain itu struktur organisasi juga menunjukkan mengenai spesialisasi-spesialisasi dari pekerjaan, saluran perintah maupun penyampaian laporan. Struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektivitas dan efisiensi suatu perusahaan didalam mencapai tujuannya.

Berikut ini merupakan fungsi dari struktur organisasi:

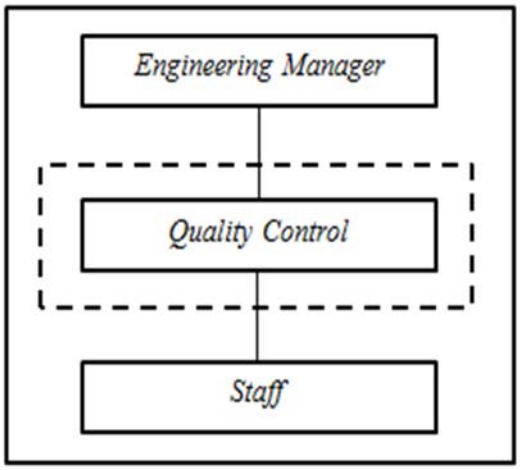
1. Kejelasan tanggung jawab
2. Kejelasan kedudukan dan kordinasi
3. Kejelasan dalam jalur hubungan
4. Pengendalian dan pengawasan

Struktur organisasi pada PT Nusa Indah Jaya Utama memakai struktur organisasi lini/garis. Dengan alasan dalam struktur organisasi PT Nusa Indah Jaya Utama adanya perintah terpusat pada atasan secara langung dan koordinasi dengan bawahannya secara langsung sehingga perintah dapat diterima dengan jelas dan mudah dilaksanakan berdasarkan tanggung jawabnya masing-masing.

Pada struktur organisasi lini/garis ini, wewenang dari atasan disalurkan secara vertikal kepada bawahan, begitu juga sebaliknya. Pertanggung jawaban dari bawahan secara langsung ditujukan kepada atasan yang memberi perintah. Struktur organisasi yang tersusun dengan baik akan memudahkan koordinasi, integrasi, serta meningkatkan efektivitas dan efisiensi suatu perusahaan didalam mencapai tujuannya. Berikut Gambar IV.2 merupakan struktur organisasi PT Nusa Indah Jaya Utama secara keseluruhan. Sedangkan Gambar IV.3 merupakan struktur organisasi PT Nusa Indah Jaya Utama pada bagian QC



Gambar IV.2 Struktur Organisasi PT Nusa Indah Jaya Utama
Sumber : PT Nusa Indah Jaya Utama (2019)



Gambar IV.3 Struktur Orgsnisasi Divisi QC pada PT Nusa Indah Jaya Utama
Sumber : PT Nusa Indah Jaya Utama (2019)

4.6 Tugas dan Wewenang Divisi QC

Dengan adanya suatu organisasi yang baik, maka memungkinkan terselenggaranya suatu kelancaran dalam proses usaha yang dilakukan oleh perusahaan dikarenakan dalam organisasi tersebut sudah ditetapkan dengan jelas mengenai wewenang, tugas, dan tanggung jawab seseorang. Berikut ini merupakan tugas dan wewenang divisi QC pada PT Nusa Indah Jaya Utama adalah sebagai berikut:

1. *Quality Control*

   Divisi QC berperan melakukan perencanaan dan mengendalikan kegiatan *Quality Control* dalam menunjang, dan memastikan kualitas produk, sistem manajemen kualitas dan manajemen lingkungan.

   Tugas dan wewenang:

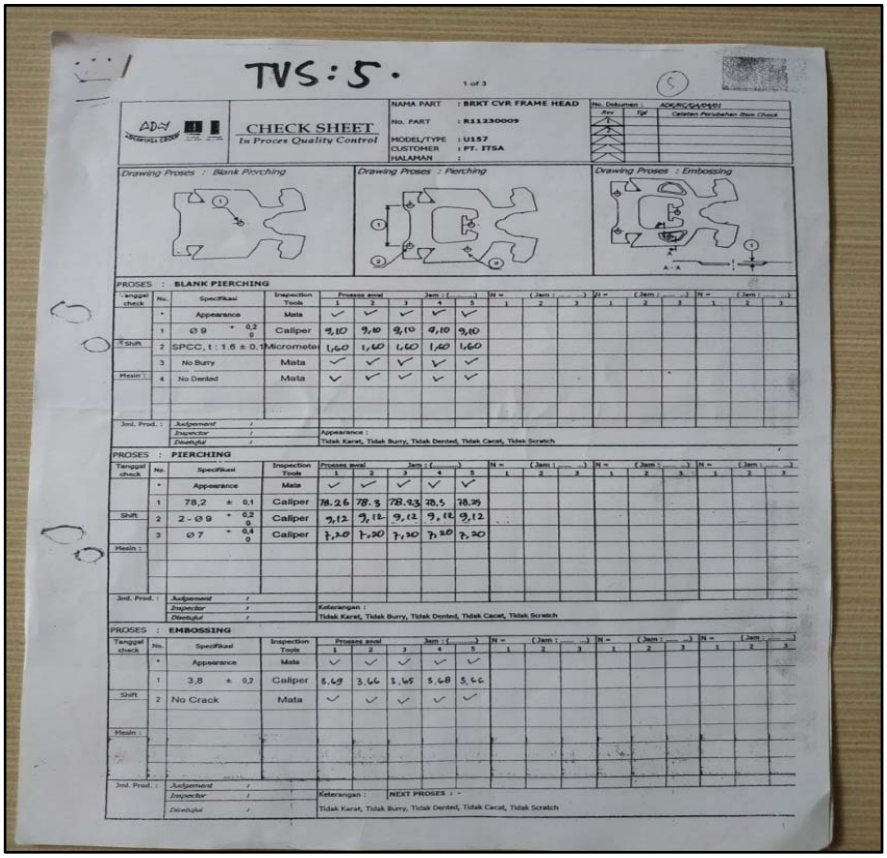
   a. Bertanggung jawab kepada *Engineering Manager.*
   b. Melaporkan hasil temuan kualitas produk *Incoming, Inprocess & Out Going* kepada *Engineering Manager*.
   c. Melakukan identifikasi terhadap masalah kualitas *part Incoming, Inprocess & Out Going*

4.7 Analisis Dokumen Kualitas Produk

Divisi *Quality Control* mempunyai tanggung jawab penuh terhadap pengendalian kualitas yang dilaksanakan setiap hari untuk mencapai kepuasan pelanggan Analisis dokumen berikut akan menjelaskan dokumen-dokumen apa saja yang digunakan dalam Sistem Informasi *Monitoring Quality Control*. Adapun dokumen-dokumen yang terkait dalam proses bisnis Divisi *Quality Control* terdiri dari *Check Sheet, Repair Card,* Data *Part Not Good, Problem Countermeasure,* dan *Monitoring Reject.* Yang dijelaskan seperti berikut:

1. *Check Sheet*

   Merupakan sebuah *form* yang yang digunakan sebagai acuan ukuran suatu produk yang dihasilkan, apakah telah sesuai dengan standar kualitas atau tidak. *Check sheet* dicatat oleh *inspector* tiga kali sehari. Dokumen ini dibuat oleh divisi *quality control* sebagai ukuran lolos atau tidaknya suatu produk. (dapat dilihat pada gambar IV.4).


TVS : 5.

CHECK SHEET
In Proces Quality Control

NAMA PART : BRKT CVR FRAME HEAD
No. PART : R11230009
MODEL/TYPE : U157
CUSTOMER : PT. ITSA
HALAMAN :

Drawing Proses : Blank Piercing | Drawing Proses : Piercing | Drawing Proses : Embossing

PROSES : BLANK PIERCHING

No.	Spesifikasi	Inspection Tools	Proses awal 1	2	3	4	5
*	Appearance	Mata	✓	✓	✓	✓	✓
1	Ø 9 +0,3 0	Caliper	9,10	9,10	9,10	9,10	9,10
2	SPCC, t : 1.6 ± 0.1	Micrometer	1,60	1,60	1,60	1,60	1,60
3	No Burry	Mata	✓	✓	✓	✓	✓
4	No Dented	Mata	✓	✓	✓	✓	✓

Judgement / Inspector / Diketahui /
Appearance : Tidak Karat, Tidak Burry, Tidak Dented, Tidak Cacat, Tidak Scratch

PROSES : PIERCHING

No.	Spesifikasi	Inspection Tools	Proses awal 1	2	3	4	5
*	Appearance	Mata	✓	✓	✓	✓	✓
1	78,2 ± 0,1	Caliper	78.26	78.3	78.23	78.5	78.29
2	2 - Ø 9 +0,3 0	Caliper	9,12	9,12	9,12	9,12	9,12
3	Ø 7 +0,4 0	Caliper	7,20	7,20	7,20	7,20	7,20

Judgement / Inspector / Diketahui /
Keterangan : Tidak Karat, Tidak Burry, Tidak Dented, Tidak Cacat, Tidak Scratch

PROSES : EMBOSSING

No.	Spesifikasi	Inspection Tools	Proses awal 1	2	3	4	5
*	Appearance	Mata	✓	✓	✓	✓	✓
1	3,8 ± 0,2	Caliper	5,68	3,66	5,65	5,68	5,66
2	No Crack	Mata	✓	✓	✓	✓	✓

Judgement / Inspector / Diketahui /
Keterangan : NEXT PROSES : -
Tidak Karat, Tidak Burry, Tidak Dented, Tidak Cacat, Tidak Scratch

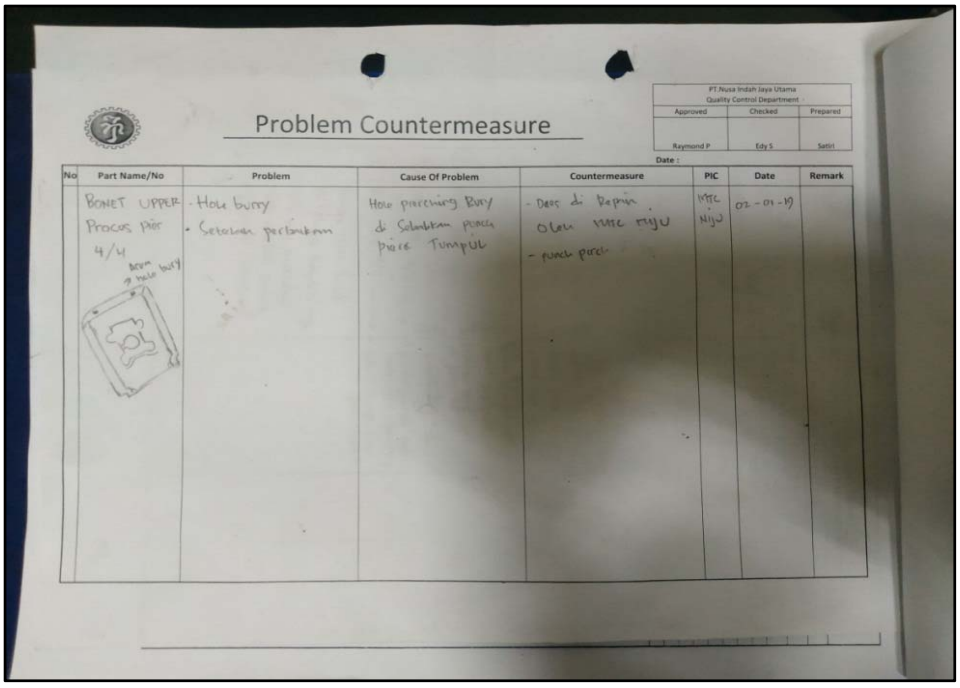
Gambar IV.4 Dokumen *Check Sheet*
Sumber : PT Nusa Indah Jaya Utama (2019)

Keterangan dari field-field pada Gambar IV.4 adalah:

- No Dokumen: berisi nomor dokumen *check sheet*
- Nama *Part*: berisi nama *part*
- *Customer*: berisi nama pelanggan
- *Drawing Process*: berisi gambar tahapan proses
- Proses: berisi proses yang dikerjakan
- Tanggal *Check*: berisi tanggal pengecekan dilakukan
- Mesin: berisi mesin yang digunakan dalam proses
- Jumlah Produksi: berisi jumlah produksi pada tahapan proses
- Spesifikasi: berisi perincian pengecekan
- *Inspection Tools*: berisi alat yang digunakan untuk pengecekan

2. *Problem Countermeasure*

Merupakan sebuah *form* yang digunakan untuk mencatat penyebab kecacatan dan membuat perencanaan penanggulangan masalah pada setiap produk yang terindikasi cacat. Dokumen ini dibuat oleh divisi *quality control* sebagai evaluasi cara penanggulangan masalah secara efektif dan efisien (dapat dilihat pada gambar IV.5).



Gambar IV.5 Dokumen *Problem Countermeasure*

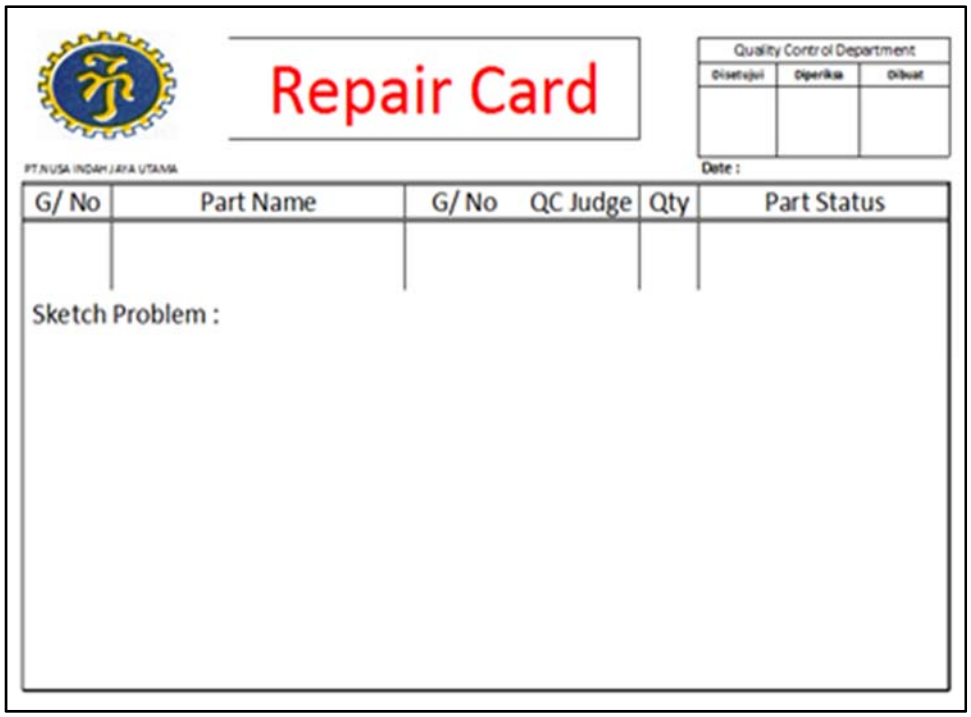
Sumber : PT Nusa Indah Jaya Utama (2019)

Keterangan dari field-field pada Gambar IV.5 adalah:

- No: berisi nomor urutan produk yang dicatat
- *Part Name*: berisi nama-nama produk
- *Problem*: berisi masalah produk
- *Cause of problem*: berisi penyebab masalah produk
- *Countermeasure*: berisi cara penanggulangan masalah
- *Person in charge*: berisi penanggung jawab
- *Date*: berisi tanggal penanggulangan masalah
- Dibuat: berisi tanda tangan *Inspector*
- Diperiksa: berisi tanda tangan *Leader QC*
- Disetujui: berisi tanda tangan *Engineering Manager*

3. *Repair Card*

Merupakan sebuah *form* yang digunakan sebagai perintah perbaikan sesuai jenis dan tingkat kecacatan produk. dokumen ini dibuat oleh divisi *quality control* ditujukan *repairing* yang berisi tentang informasi penanganan produk NG (dapat dilihat pada gambar IV.6).



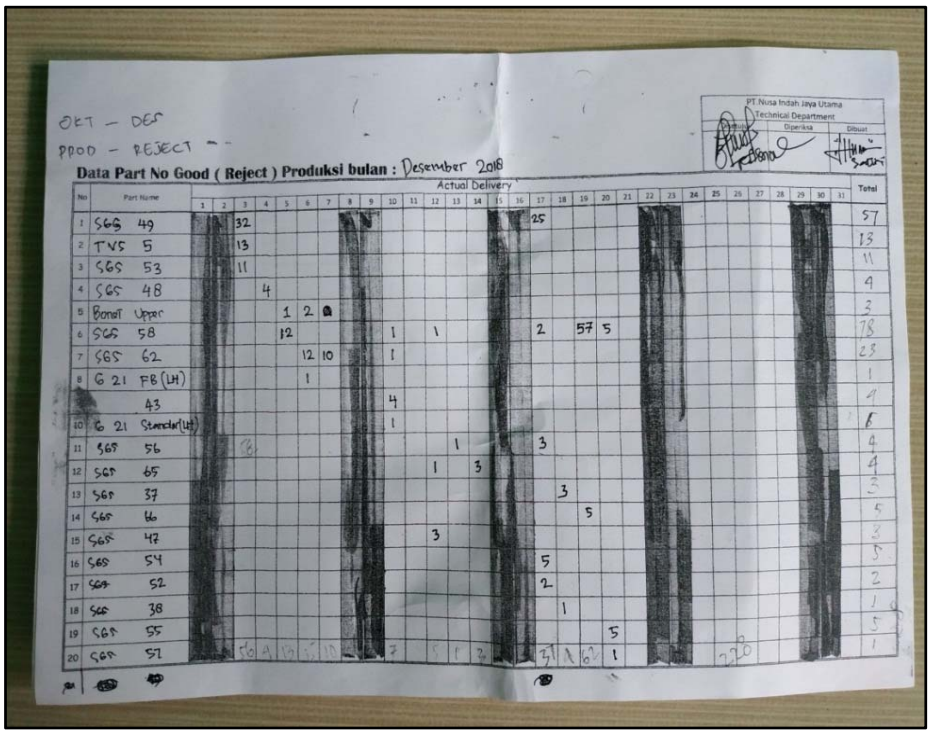
Gambar IV.6 Dokumen *Repair Card*
Sumber : PT Nusa Indah Jaya Utama (2019)

Keterangan dari field-field pada Gambar IV.6 adalah:

- *Date*: berisi tanggal kartu ini dibuat
- No: beirisi nomor urutan produk yang dicatat
- *Part Name*: berisi nama-nama produk
- QC *Judge*: berisi dugaan/penilaian produk cacat
- Qty: berisi jumlah produk yang perlu perbaikan
- *Part Status*: berisi status/tingkat kecacatan produk
- Dibuat: berisi tanda tangan *Inspector*
- Diperiksa: berisi tanda tangan *Leader QC*
- Disetujui: berisi tanda tangan *Engineering Manager*

4. Data *Part Not Good*

Merupakan sebuah *form* yang digunakan untuk mencatat jumlah produk NG yang dihasilkan setiap harinya selama satu bulan. Dokumen ini dibuat oleh divisi *quality control* sebagai bahan rekap produk NG selama satu bulan produksi (dapat dilihat pada gambar IV.7).



Gambar IV.7 Dokumen *Part Not Good*
Sumber : PT Nusa Indah Jaya Utama (2019)

Keterangan dari field-field pada Gambar IV.7 adalah:

- Bulan: beirisi nama bulan
- Tanggal: berisi tanggal produksi
- No: berisi nomor urutan produk yang dicatat
- *Part Name*: berisi nama-nama produk
- Total: berisi total produk NG yang dihasilkan selama satu bulan
- Dibuat: berisi tanda tangan *Inspector*
- Diperiksa: berisi tanda tangan *Leader QC*
- Disetujui: berisi tanda tangan *Engineering Manager*

5. *Monitoring Reject*

Merupakan sebuah laporan yang berisikan jumlah produk NG yang dihasilkan setiap harinya selama satu bulan. Dokumen ini dibuat oleh divisi *quality control* sebagai hasil rekap dari Data *Part Not Good* selama satu bulan yang akan diserahkan ke bagian manajemen sebagai laporan. (dapat dilihat pada gambar IV.8).

PT. GEMALA KEMPA DAYA — MONITORING INTERNAL REJECT

P/C : PT. GEMALA KEMPA DAYA
QUALITY DEPARTEMENT

MEDIUM PRESS
MONITORING REJECT

Jun-17

No	No Part	Name Part	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	JUMLAH
1	GKD-1	PLATE RR AXLE HOUSING						4																										4
2	GKD-2	WEB D01N																																0
3	GKD-3	BRKT UPPER ARM LH																																0
4	GKD-4	BRKT UPPER ARM RH																																0
5	GKD-5	PAD REAR SPRING																																0
6	GKD-6	CUP AD01N					40	43	46									3																132
7	GKD-7	CUP BD01N		1														1																2
8	GKD-8	SEAT RR SPRING																																0
9	GKD-9	COVER DUST																																0
10	GKD-10	PLATE BUFFLE RH																																0
11	GKD-11	PLATE BUFFLE LH																																0
12	GKD-12	DEFLECTOR DUST																																0
13	GKD-13	COVER PLUG CU																																0
14	GKD-14	COVER PLUG SLD																																0
15	GKD-15	PLATE RH						1									14	1																16
16	GKD-16	PLATE LH						1									14	2																17
17	GKD-17	CUP A D17		28				16	23		30	18		28			23	23	22			17	15											237
18	GKD-18	CUP B D17	35	28			40					27		28		39	40																	234
19	GKD-19	REINFORCEMENT B RH																																0
20	GKD-20	REINFORCEMENT B LH																																0
21	GKD-21	RBO																																0
22	GKD-22	PLATE RR SPRING SEAT SUB BY																																0
23	GKD-23	REINF SIDE RAIL RH																																0
24	GKD-24	REINF SIDE RAIL LH																																0
TOTAL REJECT			35	51	0	0	80	65	69	0	30	45	0	53	0	39	91	30	22	0	0	17	15	0	0	0	0	0	0	0	0	0	0	642

Gambar IV.8 Dokumen *Monitoring Reject*
Sumber : PT Nusa Indah Jaya Utama (2019)

Keterangan dari field-field pada Gambar IV.8 adalah:

- Bulan: beirisi nama bulan
- Tanggal: berisi tanggal produksi
- Mesin: berisi mesin yang digunakan dalam proses
- No: berisi nomor urutan produk yang dicatat
- No *part*: berisi *id part*
- *Part Name*: berisi nama-nama produk
- Jumlah: jumlah satu produk cacat yang dihasilkan selama satu bulan
- Total *Reject*: total produk cacat yang dihasilkan selama satu bulan

4.8 Analisis Proses Bisnis

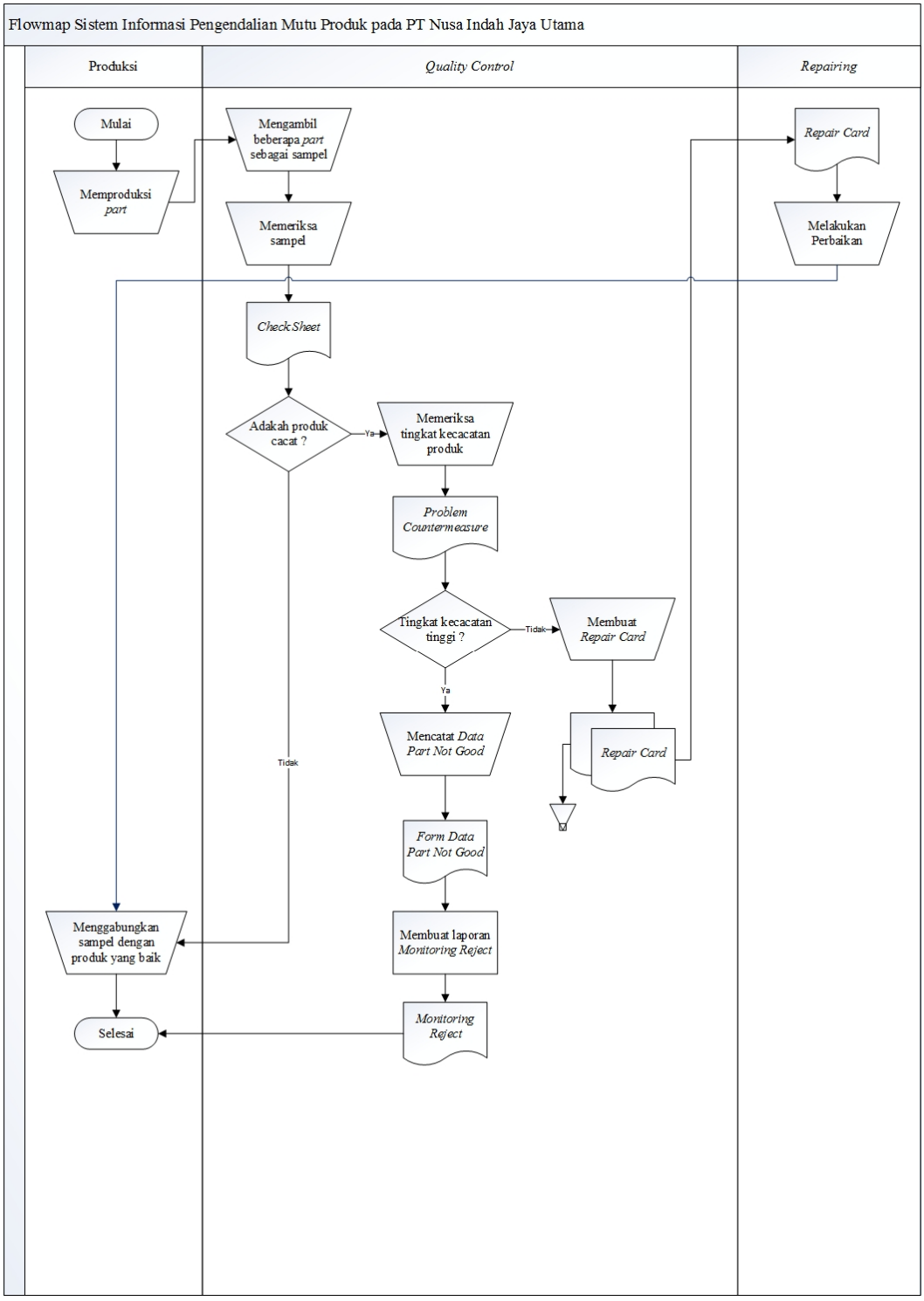
Kegiatan pengendalian kualitas pada divisi *quality control* PT Nusa Indah Jaya Utama terdiri dari pengendalian kualitas bahan baku (*check in coming*) dan pengendalian kualitas produk (*check in process, check out going*). Divisi QC akan melakukan *check in coming* pada saat bahan baku datang. Proses pengecekan meliputi penampilan, ukuran, dan kuantitas bahan baku. Setelah bahan baku dinilai sesuai standar maka dilanjut pada proses produksi.

Sedangkan pengendalian kualitas produk dilakukan pada saat proses produksi. Pada saat produk masuk ke proses produksi, tugas divisi QC melakukan pengawasan dan pengecekan produk pada tiap tahapan proses produksi secara manual menggunakan *check sheet*. Bila ditemukan penyimpangan pada produk yang dicek maka produk tersebut akan diperbaiki sesuai jenis kerusakannya. Produk yang tidak sesuai standar akan dicatat dalam *form data part not good*. Dikarenakan dalam *update* data pada pengendalian kualitas produk masih dilakukan secara manual menggunakan *check sheet* dan pencatatan hasil pengecekan masih manual menggunakan *form data part not good*, hal ini dirasa kurang efektif sehingga seringkali data mengenai produk tidak ter-*update* dengan baik.

Adapun proses Sistem *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama sebagai berikut:

1. Sistem *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama dimulai ketika Divisi Produksi telah selesai melakukan proses produksi.
2. Kemudian Divisi *Quality Control* akan mengambil beberapa unit dari hasil produksi tersebut untuk dijadikan sampel dan kemudian melakukan pemeriksaan terhadap sampel yang hasilnya dicatat pada *check sheet*.
3. Jika sampel tersebut memenuhi standar kualitas maka keseluruhan hasil produksi tersebut dinyatakan sebagai produk yang baik dan kemudian sampel tersebut akan digabungkan dengan produk yang baik lainnya.
4. Namun jika sampel tersebut tidak sesuai dengan standar kualitas maka Divisi *Quality Control* akan mengecek tingkat kecacatan yang ada pada produk tersebut untuk memastikan apakah sampel tersebut dapat diperbaiki atau tidak yang hasilnya dicatat pada *problem countermeasure*.
5. Jika tingkat kecacatan produk tersebut rendah artinya produk tersebut masih dapat diperbaiki. Kemudian Divisi *Quality Control* akan membuat *Repair Card* sebagai perintah perbaikan sesuai jenis dan tingkat kecacatan produk. Divisi *Repair* akan melakukan perbaikan terhadap produk tersebut berdasarkan informasi yang diberikan oleh Divisi *Quality Control*.
6. Setelah selesai diperbaiki maka sampel tersebut akan diperiksa kembali oleh Divisi *Quality Control*. Jika sampel tersebut telah memenuhi standar kualitas maka akan digabungkan dengan produk yang baik.
7. Sedangkan jika tingkat kecacatan produk tersebut tinggi artinya produk tersebut tidak dapat diperbaiki, dan akan dipisahkan untuk dijadikan *scrap*.
8. Kemudian Divisi *Quality Control* akan mencatat data produk NG yang dihasilkan setiap harinya selama satu bulan ke dalam sebuah *form* yang dinamakan dengan *Form Data Part Not Good*.
9. Setelah itu Divisi *Quality Control* akan merekap data yang ada pada *Form Data Part Not Good* tersebut menjadi sebuah laporan bulanan yang dinamakan dengan *Monitoring Reject*.

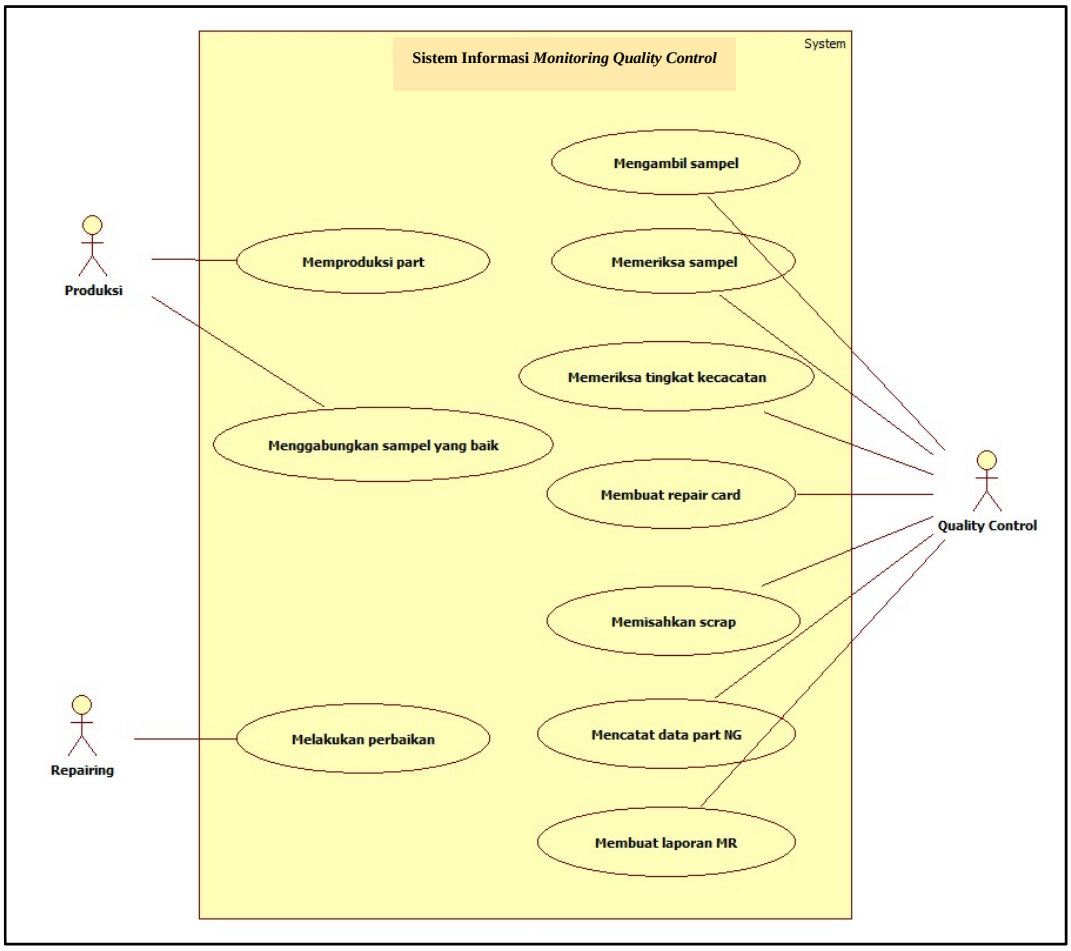
Berikut adalah kegiatan *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama diuraikan sebagai berikut (dapat dilihat pada gambar IV.9).



Gambar IV.9 *Flowmap* Sistem Informasi *Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

4.9 Pemodelan Sistem *Monitoring Quality Control* dengan UML

Penggambaran *Unified Modeling Language* (UML) Sistem Informasi *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama dengan menggunakan *Use Case Diagram* dapat dilihat pada gambar IV.10.



Gambar IV.10 *Use Case Diagram Sistem Informasi Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

Berikut penjelasan *Use Case Diagram* Sistem Informasi *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama adalah sebagai berikut:

1. Definisi Aktor

   Pendefinisian aktor pada *Use Case Diagram* Sistem Informasi *Monitoring Quality Control* yang sedang berjalan pada PT Nusa Indah Jaya Utama dapat dilihat pada tabel IV.1.

Tabel IV.1 Definisi Aktor Sistem Informasi *Monitoring Quality Control*

No	Aktor	Deskripsi
1	Produksi	Divisi yang memproduksi produk/*part* dan menggabungkan sampel dengan produk yang baik.
2	*Quality Control*	Divisi yang memeriksa sampel dan tingkat kecacatannya serta mengambil keputusan penanganan produk NG, juga membuat laporan bulanan terkait produk NG.
3	*Repairing*	Divisi yang memperbaiki produk cacat sesuai informasi dari QC.

Sumber:Hasil Analisis (2019)

2. Definisi *Use Case*

Pendefinisian *Use Case* pada *Use Case Diagram* Sistem Informasi *Monitoring Quality Control* pada PT Nusa Indah Jaya Utama dapat dilihat pada tabel IV.2.

Tabel IV.2 Definisi *Use Case* Sistem Informasi *Monitoring Quality Control*

No	*Use Case*	Deskripsi
1	Memproduksi *part*	Kegiatan yang dilakukan oleh divisi produksi dalam membuat *part* untuk memenuhi permintaan pelanggan.
2	Mengambil sampel	Kegiatan yang dilakukan oleh divisi QC dalam mengambil beberapa sampel dari produk yang telah selesai melewati tiap tahapan proses produksi.
3	Memeriksa sampel	Kegiatan yang dilakukan oleh divisi QC dalam memeriksa beberapa sampel yang dipilih dari setiap proses produksi.
4	Menggabungkan sampel yang baik	Kegiatan yang dilakukan oleh divisi produksi dalam menggabungkan sampel yang baik untuk menyelesaikan proses produksi.

Tabel IV.2 Definisi *Use Case* Sistem Informasi *Monitoring Quality Control* (lanjutan)

No	*Use Case*	Deskripsi
5	Memeriksa tingkat kecacatan	Kegiatan yang dilakukan oleh divisi QC dalam memeriksa tingkat kecacatan dan mengambil keputusan apakah produk bisa diperbaiki atau tidak.
6	Membuat *repair card*	Kegiatan yang dilakukan oleh divisi QC dalam membuat *repair card* yang berisi informasi sebagai acuan perbaikan bagi divisi *repairing.*
7	Melakukan perbaikan	Kegiatan yang dilakukan oleh divisi *repairing* dalam melakukan perbaikan sesuai informasi yang diberikan dalam *repair card*.
8	Memisahkan *scrap*	Kegiatan yang dilakukan oleh divisi QC dalam memisahkan beberapa sampel yang tidak bias diperbaiki dan akan dijadikan *scrap.*
9	Mencatat *data part NG*	Kegiatan yang dilakukan oleh divisi QC dalam mencatat produk NG yang ditemukan setiap hari.
10	Membuat laporan *MR*	Kegiatan yang dilakukan oleh divisi QC dalam membuat laporan *monitoring reject* untuk direkap setiap bulan.

Sumber:Hasil Analisis (2019)

BAB V
ANALISIS DAN PEMBAHASAN

5.1 Analisis Kebutuhan Sistem

Pada tahap ini akan dilakukan analisis terhadap kebutuhan– kebutuhan sistem. Sistem yang dianalisis adalah sistem yang berisi informasi tentang segala sesuatu yang berkaitan dengan *monitoring quality control.* Berikut ini Tabel V.1 adalah daftar kebutuhan fungsional sistem untuk aplikasi sistem informasi *monitoring quality control* pada PT Nusa Indah Jaya Utama:

Tabel V.1 Analisis Kebutuhan Fungsional Sistem

Identifikasi Masalah	Kebutuhan Pengguna	Kebutuhan Sistem	Uraian
Dokumen mengenai kualitas produk mudah hilang dan rusak karena dokumen-dokumen yang beredar dan diarsipkan masih dalam bentuk kertas.	Pengelolaan manajemen data yang terkomputerisasi dan tertata rapi serta menyimpan data-data dokumen ke dalam basis data.	Terdapat proses penyimpanan data kualitas produk ke dalam basis data melalui sistem terkomputerasi.	Sistem informasi mampu mengelola data kualitas produk dengan menggunakan basis data.
Bagian yang terkait dengan QC setiap harinya harus menulis secara manual dan menyebabkan *input* data membutuhkan waktu yang lama.	Membuat sistem penginputan data secara terkomputerisasi dan terintegrasi ke bagian yang terkait dengan mudah dan cepat.	Terdapat proses penginputan data-data terkait pada bagian QC.	Sistem Informasi mampu mengelola data-data terkait bagian QC secara terkomputerisasi dan terintegrasi.

Tabel V.1 Analisis Kebutuhan Fungsional Sistem (lanjutan)

Identifikasi Masalah	Kebutuhan Pengguna	Kebutuhan Sistem	Uraian
Proses pembuatan laporan *Monitoring Reject* dirasa tidak efisien disebabkan sulitnya pencarian dokumen kualitas produk.	Membuat sistem pelaporan data *Monitoring Reject* yang dapat disimpan dan dicari dengan mudah melalui sistem berbasis web.	Terdapat menu yang berisi informasi terkait data laporan *Monitoring Reject.*	Sistem Informasi mampu memberikan informasi mengenai data-data yang dibutuhkan terkait laporan *Monitoring Reject.*

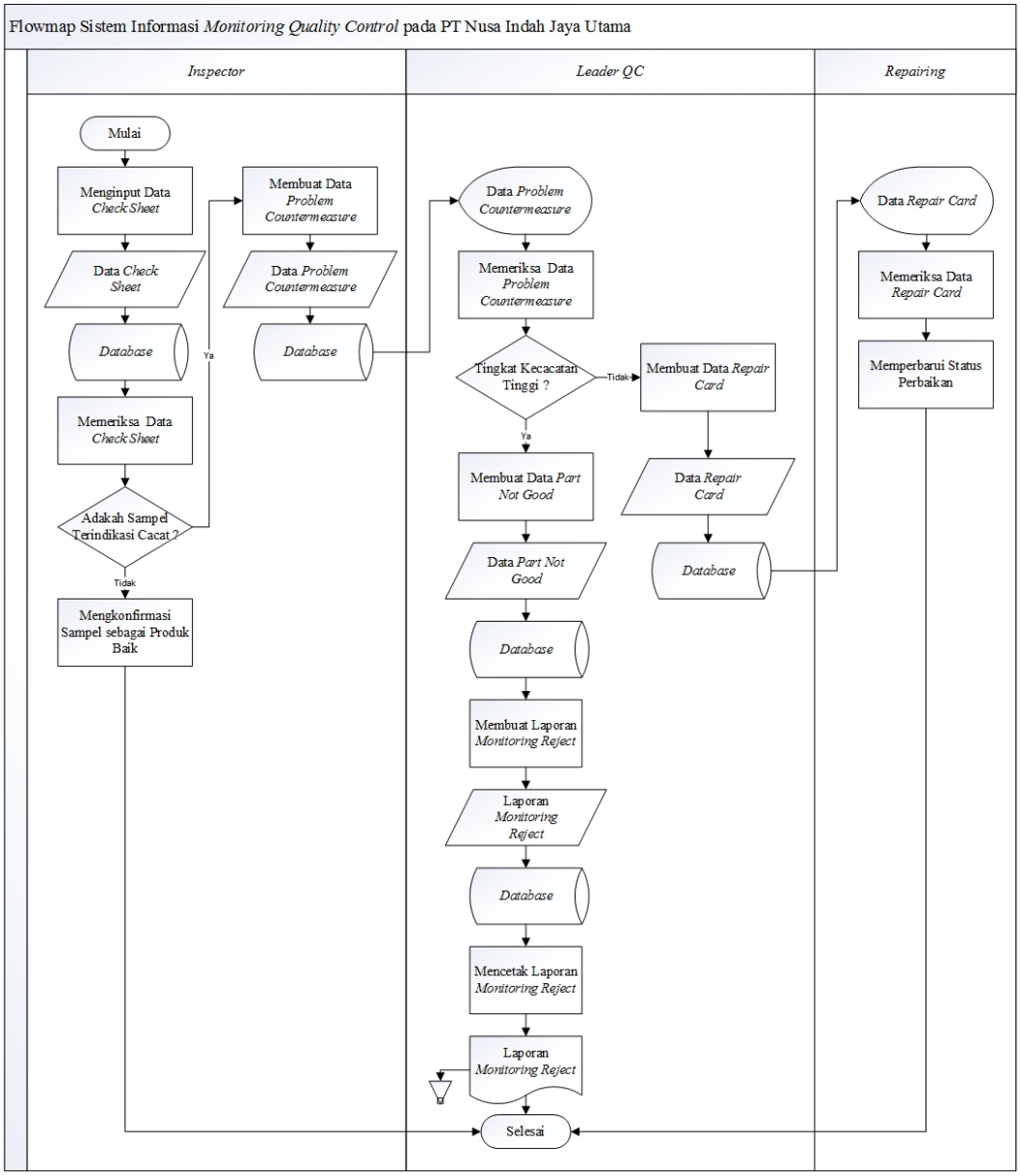
Sumber:Hasil Analisis (2019)

5.2 Prosedur Sistem Informasi *Monitoring Quality Control*

Berikut adalah prosedur sistem informasi *monitoring quality control* yang diusulkan melalui tahapan:

1. *Inspector* menginput data *Check Sheet* dan disimpan dalam basis data.
2. *Inspector* memeriksa data *Check Sheet* apakah ada sampel terindikasi cacat. Jika tidak ada *Inspector* akan menerapkan sampel sebagai produk yang baik. Tetapi jika ditemukan ada sampel yang cacat maka *Inspector* akan membuat data *Problem Countermeasure* dan disimpan dalam basis data.
3. *Leader QC* memeriksa data *Problem Countermeasure* apakah tingkat kecacatan tinggi. Jika tingkat kecacatan rendah maka *Leader QC* akan membuat data *Repair Card* dan disimpan dalam basis data.
4. *Repairing* memeriksa data *Repair Card* lalu melakukan perbaikan sesuai instruksi. Kemudian *Repairing* akan mengupdate status perbaikan apakah sedang menunggu pengerjaan, sedang dikerjakaan, atau selesai dikerjakan.
5. Sedangkan jika tingkat kecacatan produk tinggi artinya produk tersebut tidak dapat diperbaiki. Lalu *Leader QC* akan menginput data *Part Not Good* yang dihasilkan setiap harinya selama satu bulan dan disimpan dalam *database*.
6. *Leader QC* membuat dan mencetak laporan *Monitoring Reject* berdasarkan data *Part Not Good* yang direkap selama satu bulan.

Berikut merupakan Flowmap sistem informasi *monitoring quality control* usulan diuraikan sebagai berikut (dapat dilihat pada gambar V.1).



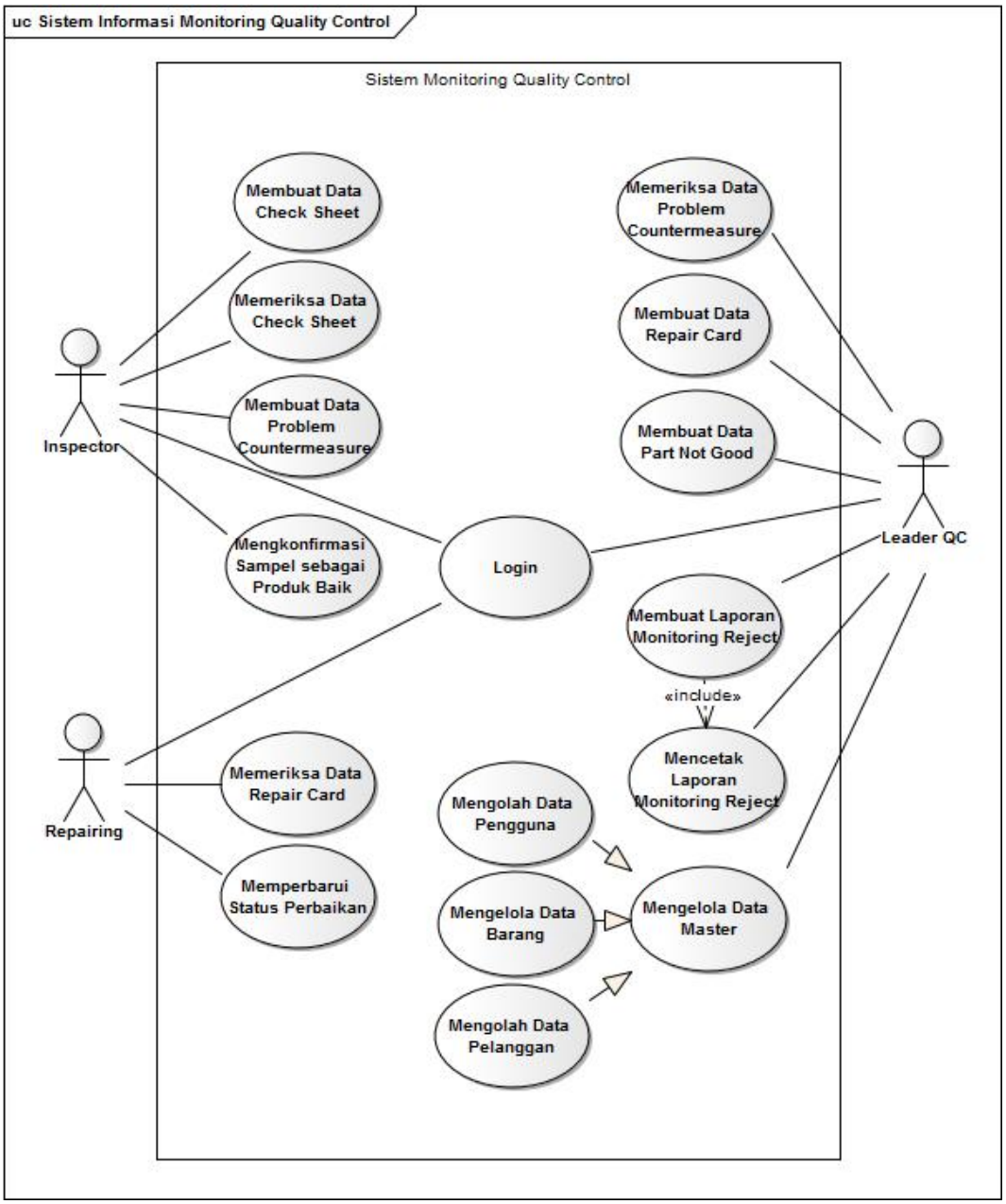
Gambar V.1 Flowmap Sistem Informasi *Monitoring Quality Control* usulan
Sumber:Hasil Analisis (2019)

5.3 Analisis Sistem Usulan

Analisis sistem informasi *monitoring quality control* menggunakan pemodelan sistem *Unified Modelling Language* (UML), berikut akan dimodelkan analisis menggunakan beberapa model yaitu *use case diagram*, *activity diagram*, *sequence diagram*, *class diagram*, dan *deployment diagram*. Tahapan analisis ini akan memberikan gambaran mengenai aliran informasi dan data pada sistem informasi *monitoring quality control* yang akan dibangun.

5.3.1 *Use Case Diagram*

Use case diagram Sistem Informasi *Monitoring Quality Control* yang yang diusulkan digambarkan pada Gambar V.2 dibawah ini:



Gambar V.2 Use Case Diagram Sistem Informasi Monitoring Quality Control
Sumber:Hasil Analisis (2019)

Berikut penjelasan *Use Case Diagram* Sistem Informasi *Monitoring Quality Control* usulan adalah sebagai berikut:

1. Definisi Aktor

   Pendefinisian aktor pada *Use Case Diagram* Sistem Informasi *Monitoring Quality Control* usulan dapat dilihat pada tabel V.2.

Tabel V.2 Definisi Aktor Sistem Informasi *Monitoring Quality Control*

No	Aktor	Deskripsi
1	*Inspector*	Bagian yang mengelola data sampel.
2	*Leader QC*	Bagian yang mengelola data master dan mengambil keputusan penanganan produk cacat, serta membuat laporan *Monitoring Reject* setiap bulan.
3	*Repairing*	Bagian yang melakukan proses perbaikan produk cacat sesuai instruksi.

Sumber:Hasil Analisis (2019)

2. Definisi *Use Case*

   Pendefinisian *Use Case* pada *Use Case Diagram* Sistem Informasi *Monitoring Quality Control* usulan dapat dilihat pada tabel V.3.

Tabel V.3 Definisi *Use Case* Sistem Informasi *Monitoring Quality Control*

No	*Use Case*	Deskripsi
1	*Login*	Proses melakukan *login* pada sistem informasi *monitoring quality control*
2	Mengelola Data Pengguna	Proses melakukan pengelolaan terhadapat data pengguna pada perusahaan
3	Mengelola Data Barang	Proses melakukan pengelolaan terhadapat data barang pada perusahaan
4	Mengelola Data Pelanggan	Proses melakukan pengelolaan terhadapat data pelanggan pada perusahaan
5	Membuat Data *Check Sheet*	Proses membuat data *check sheet* dari hasil pengukuran yang dilakukan oleh *inspector*
6	Memeriksa Data *Check Sheet*	Proses memeriksa data *check sheet* dari hasil pencatatan yang dilakukan oleh *inspector*

Tabel V.3 Definisi *Use Case* Sistem Informasi *Monitoring Quality Control* (lanjutan)

No	*Use Case*	Deskripsi
7	Mengkonfirmasi Sampel sebagai Produk Baik	Proses mengkonfirmasi sampel sebagai produk baik setelah melalui proses pemeriksaan
8	Membuat Data *Problem Countermeasure*	Proses membuat data *problem countermeasure* dengan menganalisis masalah yang dihadapi
9	Memeriksa Data *Problem Countermeasure*	Proses memeriksa data *problem countermeasure* dari hasil analisis
10	Membuat Data *Repair Card*	Proses membuat data *repair card* sesuai informasi
11	Memeriksa Data *Repair Card*	Proses memeriksa data *repair card* sesuai informasi yang tersedia
12	Memperbarui Status Perbaikan	Proses memperbarui status perbaikan guna memberitahu proses pengerjaan terbaru
13	Membuat Data *Part Not Good*	Proses membuat data *part not good* sesuai dengan data yang masuk setiap hari
14	Membuat Laporan *Monitoring Reject*	Proses membuat laporan *monitoring reject* dari hasil rekap data *part not good* selama satu bulan
15	Mencetak Laporan *Monitoring Reject*	Proses mencetak laporan *monitoring reject* yang dilakukan oleh *leader qc*

Sumber:Hasil Analisis (2019)

3. Skenario *Use Case*

Skenario jalannya masing-masing *use case* pada sistem informasi *monitoring quality control*, dapat dilihat pada poin berikut:

a. Use Case Login

Berikut adalah definisi *use case* login yang dapat dilihat pada (Tabel V.4):

Tabel V.4 *Use Case Description Login*

Use Case Login	
Deskripsi	*Use Case* ini menggambarkan proses *user* masuk ke sistem
Aktor	*Leader QC, Inspector, Repairing*
Pre-condition	*User* belum melakukan *login*

Tabel V.4 *Use Case Description Login* (lanjutan)

Use Case Login	
Basic Flow	1. *User* membuka aplikasi 2. Mengisi *Username* dan *Password* 3. Sistem memeriksa *user account* apakah valid 4. Sistem menampilkan menu utama
Alternative Flow	*User* gagal masuk ke menu utama dan sistem mengkonfirmasi *username* dan *password* salah
Post-condition	*User* berhasil masuk ke menu utama

Sumber:Hasil Analisis (2019)

b. *Use Case* Mengelola Data Pengguna

Berikut adalah definisi *use case* mengelola data pengguna yang dapat dilihat pada (Tabel V.5):

Tabel V.5 *Use Case Description* Mengelola Data Pengguna

Use Case Mengelola Data Pengguna	
Deskripsi	*Use Case* ini menggambarkan proses mengelola data pengguna
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu data master 2. Pilih submenu data pengguna 3. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah data - *User* mengisi *form* tambah data - Jika ubah, sistem menampilkan *form* ubah data - *User* mengisi *form* ubah data - Jika hapus, sistem menghapus data 4. Sistem menyimpan dan memperbarui data
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data *user* yang telah diperbarui

Sumber:Hasil Analisis (2019)

c. *Use Case* Mengelola Data Produk

Berikut adalah definisi *use case* mengelola data barang yang dapat dilihat pada (Tabel V.6):

Tabel V.6 *Use Case Description* Mengelola Data Barang

Use Case Mengelola Data Barang	
Deskripsi	*Use Case* ini menggambarkan proses mengelola data barang
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu data master 2. Pilih submenu data barang 3. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah data - *User* mengisi *form* tambah data - Jika ubah, sistem menampilkan *form* ubah data - *User* mengisi *form* ubah data - Jika hapus, sistem menghapus data 4. Sistem menyimpan dan memperbarui data
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data produk yang telah diperbarui

Sumber:Hasil Analisis (2019)

d. *Use Case* Mengelola Data Pelanggan

Berikut adalah definisi *use case* mengelola data pelanggan yang dapat dilihat pada (Tabel V.7):

Tabel V.7 *Use Case Description* Mengelola Data Pelanggan

Use Case Mengelola Data Pelanggan	
Deskripsi	*Use Case* ini menggambarkan proses mengelola data pelanggan
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*

Tabel V.7 *Use Case Description* Mengelola Data Pelanggan (lanjutan)

Use Case Mengelola Data Pelanggan	
Basic Flow	1. *User* memilih menu data master 2. Pilih submenu data pelanggan 3. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah data - *User* mengisi *form* tambah data - Jika ubah, sistem menampilkan *form* ubah data - *User* mengisi *form* ubah data - Jika hapus, sistem menghapus data 4. Sistem menyimpan dan memperbarui data
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data produk yang telah diperbarui

Sumber:Hasil Analisis (2019)

e. *Use Case* Membuat Data *Check Sheet*

Berikut adalah definisi *use case* membuat data *check sheet* yang dapat dilihat pada (Tabel V.8):

Tabel V.8 *Use Case Description* Membuat Data *Check Sheet*

Use Case Membuat Data *Check Sheet*	
Deskripsi	*Use Case* ini menggambarkan proses membuat data *check sheet* dari hasil pengukuran yang dilakukan oleh *inspector*
Aktor	*Inspector*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *check sheet* 2. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah *check sheet* - *User* mengisi *form check sheet* - Jika ubah, sistem menampilkan *form* ubah *check sheet* - *User* mengisi *form* ubah *check sheet* - Jika hapus, sistem menghapus data 3. Sistem menyimpan dan memperbarui data

Tabel V.8 *Use Case Description* Membuat Data *Check Sheet* (lanjutan)

Use Case Membuat Data *Check Sheet*	
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data *check sheet* yang telah diperbarui

Sumber:Hasil Analisis (2019)

f. *Use Case* Memeriksa Data *Check Sheet*

Berikut adalah definisi *use case* memeriksa data *check sheet* yang dapat dilihat pada (Tabel V.9):

Tabel V.9 *Use Case Description* Memeriksa Data *Check Sheet*

Use Case Memeriksa Data *Check Sheet*	
Deskripsi	*Use Case* ini menggambarkan proses memeriksa data *check sheet* dari hasil pencatatan yang dilakukan oleh *inspector*
Aktor	*Inspector*
Basic Flow	1. *User* memilih menu *check sheet* 2. Klik periksa 3. Memeriksa apakah ada data yang terindikasi cacat 4. Jika ada, *user* akan mengisi status *reject* 5. Jika tidak ada, *user* akan mengisi status *good*
Alternative Flow	Jika data yang dicari tidak ada sistem akan menunjukan pesan “Data yang dimaksud tidak ada”
Post-condition	Sistem menampilkan data *check sheet* yang telah diperbarui

Sumber:Hasil Analisis (2019)

g. *Use Case* Mengkonfirmasi Sampel sebagai Produk Baik

Berikut adalah definisi *use case* mengkonfirmasi sampel sebagai produk baik yang dapat dilihat pada (Tabel V.10):

Tabel V.10 *Use Case Description* Mengkonfirmasi Sampel sebagai Produk Baik

Use Case Mengkonfirmasi Sampel sebagai Produk Baik	
Deskripsi	*Use Case* ini menggambarkan proses mengkonfirmasi sampel sebagai produk baik setelah melalui proses pemeriksaan
Aktor	*Inspector*
Pre-condition	*User* telah melakukan *login*

Tabel V.10 *Use Case Description* Mengkonfirmasi Sampel sebagai Produk Baik (lanjutan)

Use Case Mengkonfirmasi Sampel sebagai Produk Baik	
Basic Flow	1. *User* memilih menu *check sheet* 2. Klik periksa 3. Memilih data yang akan dikonfirmasi *good*
Post-condition	Sistem menampilkan data *check sheet* yang telah diperbarui

Sumber:Hasil Analisis (2019)

h. *Use Case* Membuat Data *Problem Countermeasure*

Berikut adalah definisi *use case* membuat data *problem countermeasure* yang dapat dilihat pada (Tabel V.11):

Tabel V.11 *Use Case Description* Membuat Data *Problem Countermeasure*

Use Case Membuat Data *Problem Countermeasure*	
Deskripsi	*Use Case* ini menggambarkan proses membuat data *problem countermeasure* dengan menganalisis masalah yang dihadapi
Aktor	*Inspector*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *problem countermeasure* 2. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah *problem countermeasure* - *User* mengisi *form* tambah *problem countermeasure* - Jika ubah, sistem menampilkan *form* ubah *problem countermeasure* - *User* mengisi *form* ubah *problem countermeasure* - Jika hapus, sistem menghapus data 3. Sistem menyimpan dan memperbarui data
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data *problem countermeasure* yang telah diperbarui

Sumber:Hasil Analisis (2019)

i. *Use Case* Memeriksa Data *Problem Countermeasure*

Berikut adalah definisi *use case* memeriksa data *problem countermeasure* yang dapat dilihat pada (Tabel V.12):

Tabel V.12 *Use Case Description* Memeriksa Data *Problem Countermeasure*

Use Case Memeriksa Data *Problem Countermeasure*	
Deskripsi	*Use Case* ini menggambarkan proses memeriksa data *problem countermeasure* dari hasil analisis
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *problem countermeasure* 2. Mencari data yang akan dianalisis 3. Menganalisis apakah tingkat kecacatan tinggi 4. Jika iya, *user* akan mengisi status *scrap* 5. Jika tidak, *user* akan mengisi status *repair*
Alternative Flow	Jika data yang dicari tidak ada sistem akan menunjukan pesan "Data yang dimaksud tidak ada"
Post-condition	Sistem menampilkan data *problem countermeasure* yang telah diperbarui

Sumber:Hasil Analisis (2019)

j. *Use Case* Membuat Data *Repair Card*

Berikut adalah definisi *use case* membuat data *repair card* yang dapat dilihat pada (Tabel V.13):

Tabel V.13 *Use Case Description* Membuat Data *Repair Card*

Use Case Membuat Data *Repair Card*	
Deskripsi	*Use Case* ini menggambarkan proses membuat data *repair card* sesuai informasi
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*

Tabel V.13 *Use Case Description* Membuat Data *Repair Card* (lanjutan)

Use Case Membuat Data *Repair Card*	
Basic Flow	1. *User* memilih menu *repair card* 2. Memilih aksi tambah, ubah, atau hapus 3. Jika tambah, sistem menampilkan *form* tambah *repair card* - *User* mengisi *form* tambah *repair card* - Jika ubah, sistem menampilkan *form* ubah *repair card* - *User* mengisi *form* tambah *repair card* - Jika hapus, sistem menghapus data 4. Sistem menyimpan dan memperbarui data
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data *repair card* yang telah diperbarui

Sumber:Hasil Analisis (2019)

k. *Use Case* Memeriksa Data *Repair Card*

Berikut adalah definisi *use case* memeriksa data *repair card* yang dapat dilihat pada (Tabel V.14):

Tabel V.14 *Use Case Description* Memeriksa Data *Repair Card*

Use Case Memeriksa Data *Repair Card*	
Deskripsi	*Use Case* ini menggambarkan proses memeriksa data *repair card* sesuai informasi yang tersedia
Aktor	*Repairing*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *repair card* 2. Mencari data yang akan dikerjakan 3. Memasukan status dikerjakan
Alternative Flow	Jika data yang dicari tidak ada sistem akan menunjukan pesan “Data yang dimaksud tidak ada”
Post-condition	Sistem menampilkan data *repair card* yang telah diperbarui

Sumber:Hasil Analisis (2019)

l. *Use Case* Memperbarui Status Perbaikan

Berikut adalah definisi *use case* memperbarui status perbaikan yang dapat dilihat pada (Tabel V.15):

Tabel V.15 *Use Case Description* Memperbarui Status Perbaikan

Use Case Memperbarui Status Perbaikan	
Deskripsi	*Use Case* ini menggambarkan proses memperbarui status perbaikan guna memberitahu proses pengerjaan terbaru
Aktor	*Repairing*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *repair card* 2. Mencari data yang telah dikerjakan 3. Mengubah status menjadi selesai dikerjakan
Alternative Flow	Jika data yang dicari tidak ada sistem akan menunjukan pesan "Data yang dimaksud tidak ada"
Post-condition	Sistem menampilkan data *repair card* yang telah diperbarui

Sumber:Hasil Analisis (2019)

m. *Use Case* Membuat Data *Part Not Good*

Berikut adalah definisi *use case* membuat data *part not good* yang dapat dilihat pada (Tabel V.16):

Tabel V.16 *Use Case Description* Membuat Data *Part Not Good*

Use Case Membuat Data *Part Not Good*	
Deskripsi	*Use Case* ini menggambarkan proses membuat data *part not good* sesuai dengan data yang masuk setiap hari
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *part not good* 2. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah *part not good* - *User* mengisi *form* tambah *part not good* - Jika ubah, sistem menampilkan *form* ubah *part not good* - *User* mengisi *form* ubah *part not good* - Jika hapus, sistem menghapus data 3. Sistem menyimpan dan memperbarui data

Tabel V.16 *Use Case Description* Membuat Data *Part Not Good* (lanjutan)

Use Case Membuat Data *Part Not Good*	
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan data *part not good* yang telah diperbarui

Sumber:Hasil Analisis (2019)

n. *Use Case* Membuat Laporan *Monitoring Reject*

Berikut adalah definisi *use case* membuat laporan *monitoring reject* yang dapat dilihat pada (Tabel V.17):

Tabel V.17 *Use Case Description* Membuat Laporan *Monitoring Reject*

Use Case Membuat Laporan *Monitoring Reject*	
Deskripsi	*Use Case* ini menggambarkan proses membuat laporan *monitoring reject* dari hasil rekap data *part not good* selama satu bulan
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *monitoring reject* 2. Memilih aksi tambah, ubah, atau hapus - Jika tambah, sistem menampilkan *form* tambah *monitoring reject* - *User* mengisi *form* tambah *monitoring reject* - Jika hapus, sistem memperbarui data 3. Sistem menyimpan dan memperbarui data
Alternative Flow	Jika data yang diisi masih kosong maka sistem akan menunjukan pesan “Isi isian ini”
Post-condition	Sistem menampilkan laporan *monitoring reject* yang telah diperbarui

Sumber:Hasil Analisis (2019)

o. *Use Case* Mencetak Laporan *Monitoring Reject*

Berikut adalah definisi *use case* mencetak laporan *monitoring reject* yang dapat dilihat pada (Tabel V.18):

Tabel V.18 *Use Case Description* Mencetak Laporan *Monitoring Reject*

Use Case Mencetak Laporan *Monitoring Reject*	
Deskripsi	*Use Case* ini menggambarkan proses mencetak laporan *monitoring reject* yang dilakukan oleh *leader qc*
Aktor	*Leader QC*
Pre-condition	*User* telah melakukan *login*
Basic Flow	1. *User* memilih menu *monitoring reject* 2. Mencari data yang dicetak 3. Memilih data yang akan dicetak 4. Sistem menampilkan *print priview* dari laporan *monitoring reject* yang dipilih
Alternative Flow	Jika data yang dicari tidak ada sistem akan menunjukan pesan “Data yang dimaksud tidak ada”
Post-condition	Sistem mencetak laporan *monitoring reject*

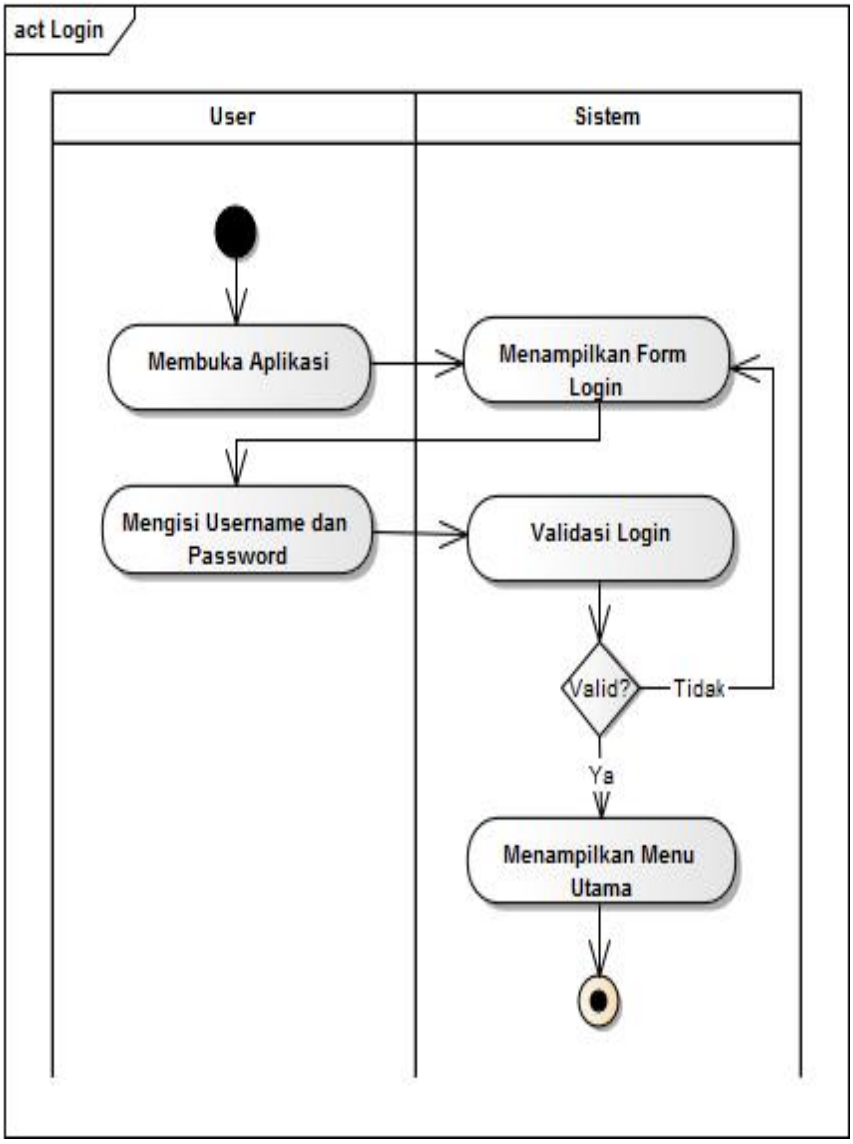
Sumber:Hasil Analisis (2019)

5.3.2 Activity Diagram

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

1. *Activity Diagram Login*

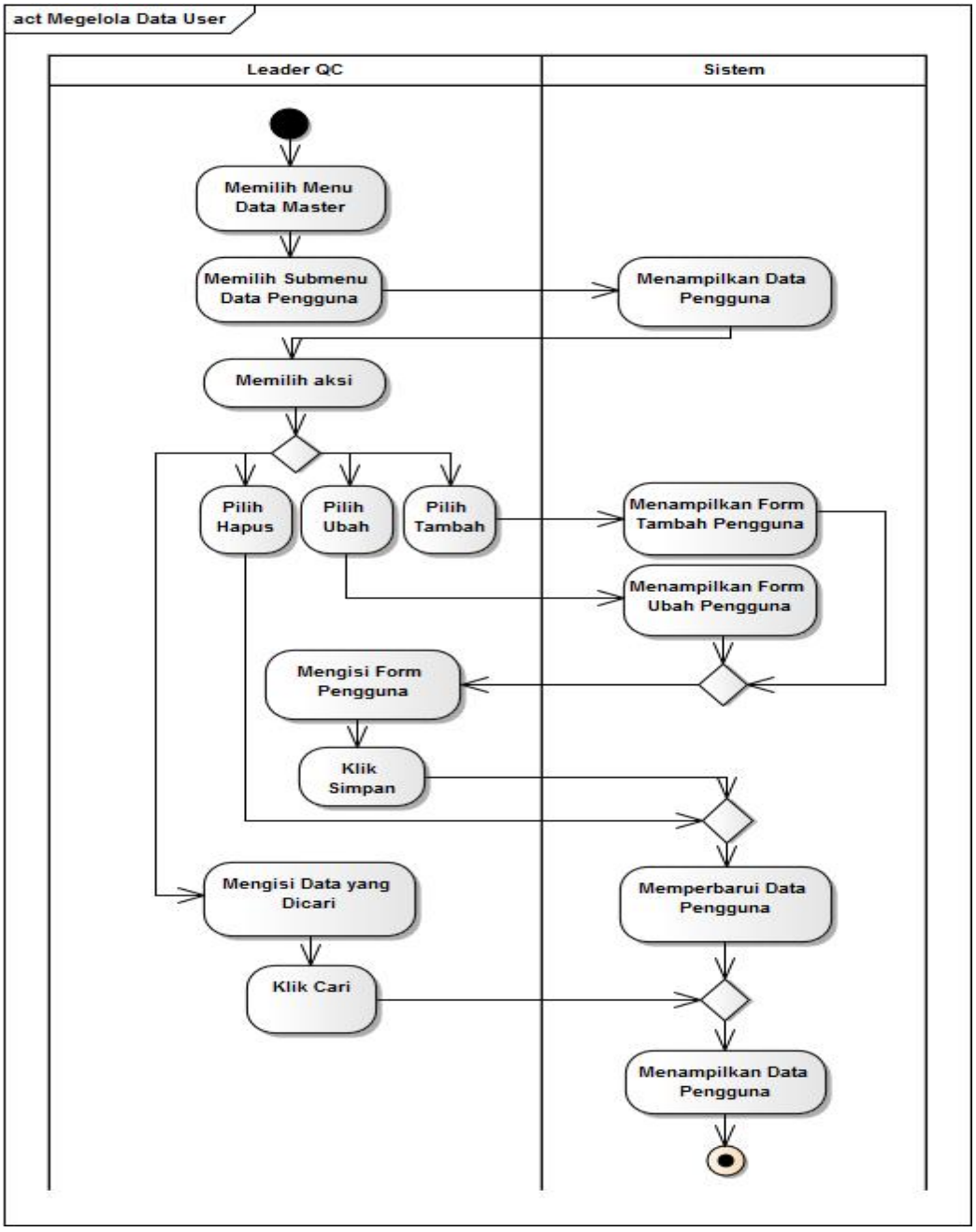
   Activity Diagram Login ini menjelaskan alur aktivitas yang dilakukan oleh *user* untuk dapat masuk ke dalam sistem informasi *monitoring quality control. Activity diagram login* dapat dilihat pada Gambar V.3:



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

2. *Activity Diagram* Mengelola Data Pengguna

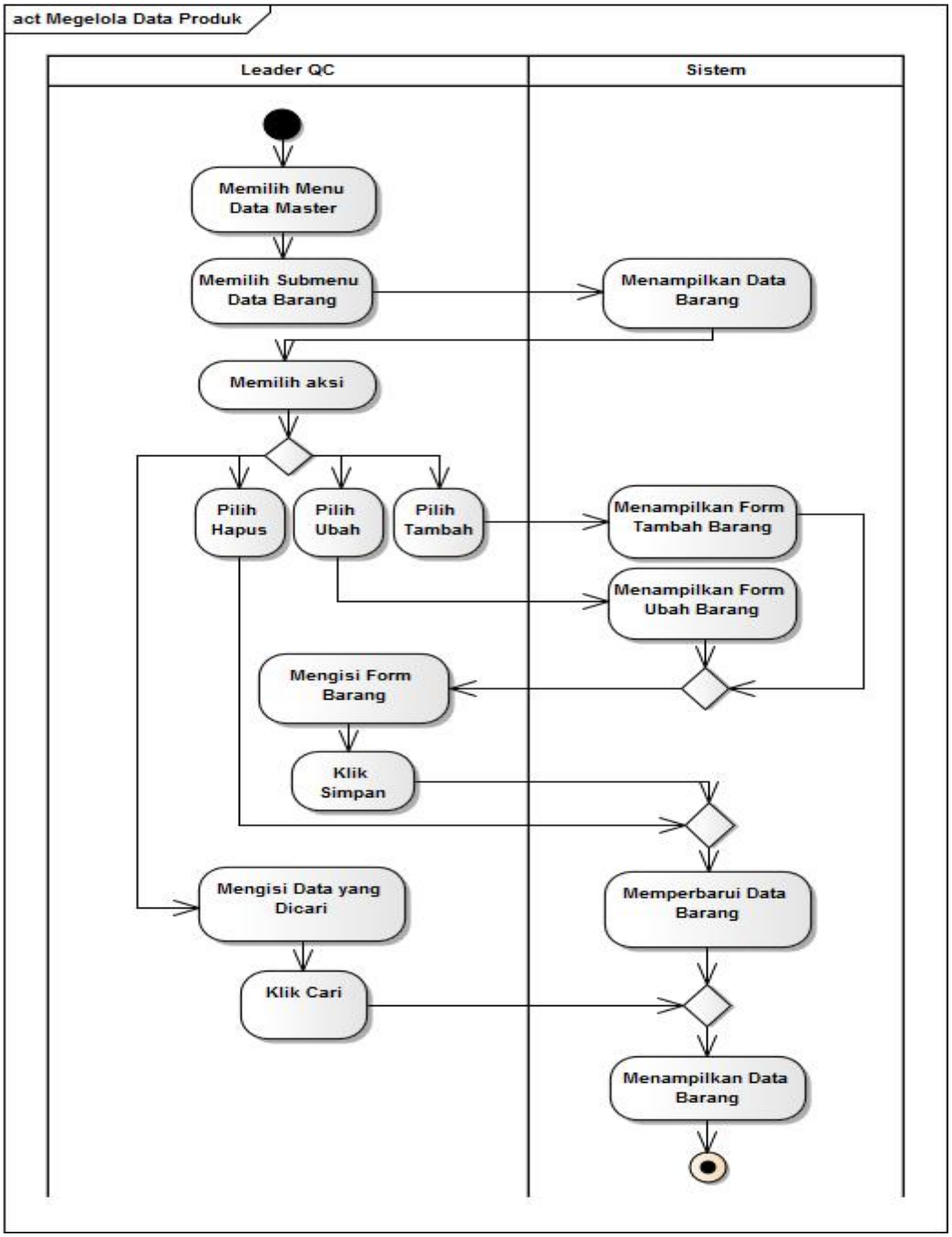
   Activity diagram mengelola data pengguna ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat melakukan proses pengelolaan data *user* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* mengelola data pengguna dapat dilihat pada Gambar V.4:



Gambar V.4 *Activity Diagram* Mengelola Data Pengguna
Sumber:Hasil Analisis (2019)

3. *Activity Diagram* Mengelola Data Barang

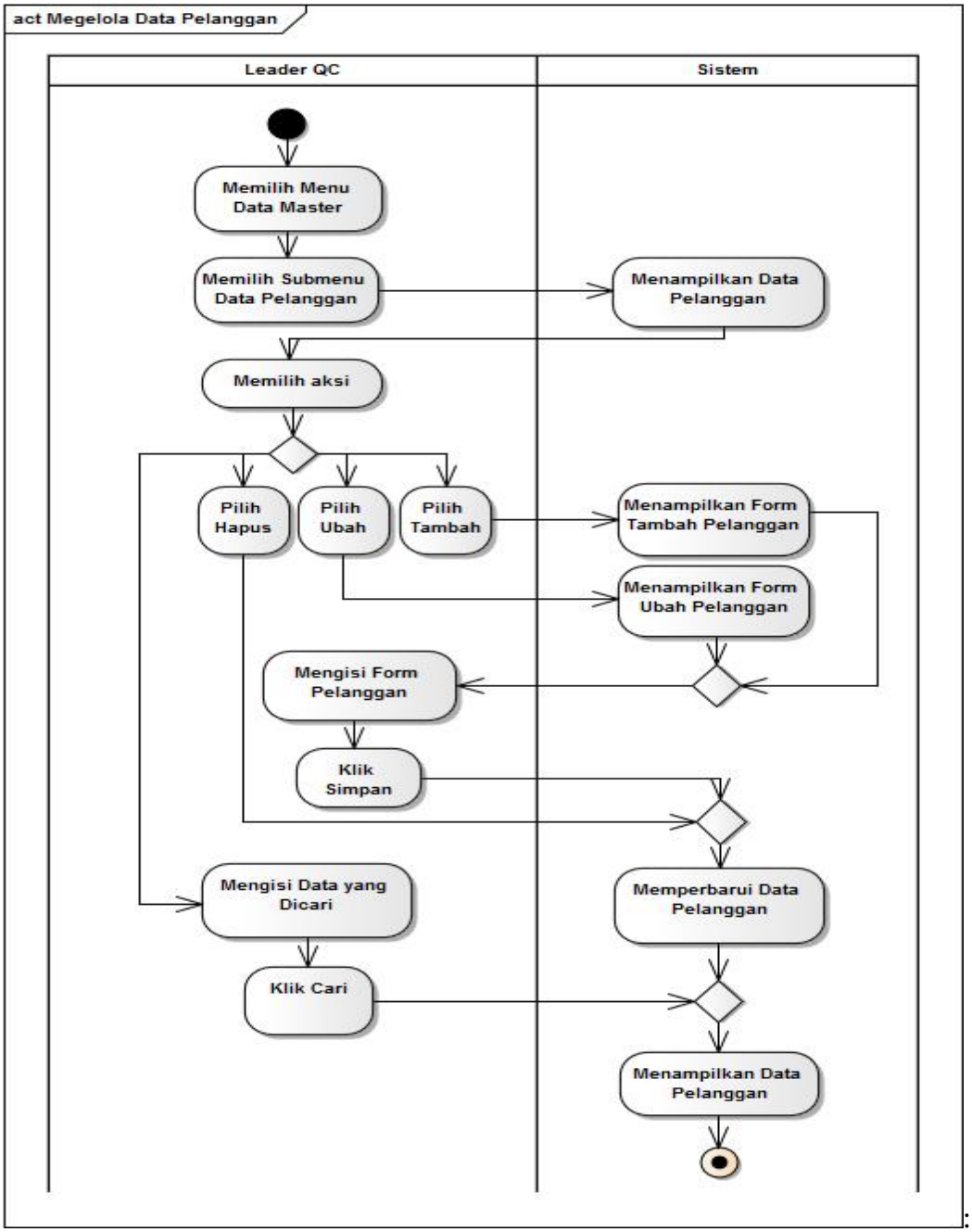
   Activity diagram mengelola data barang ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat melakukan proses pengelolaan data barang serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* mengelola data barang dapat dilihat pada Gambar V.5:



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

4. *Activity Diagram* Mengelola Data Pelanggan

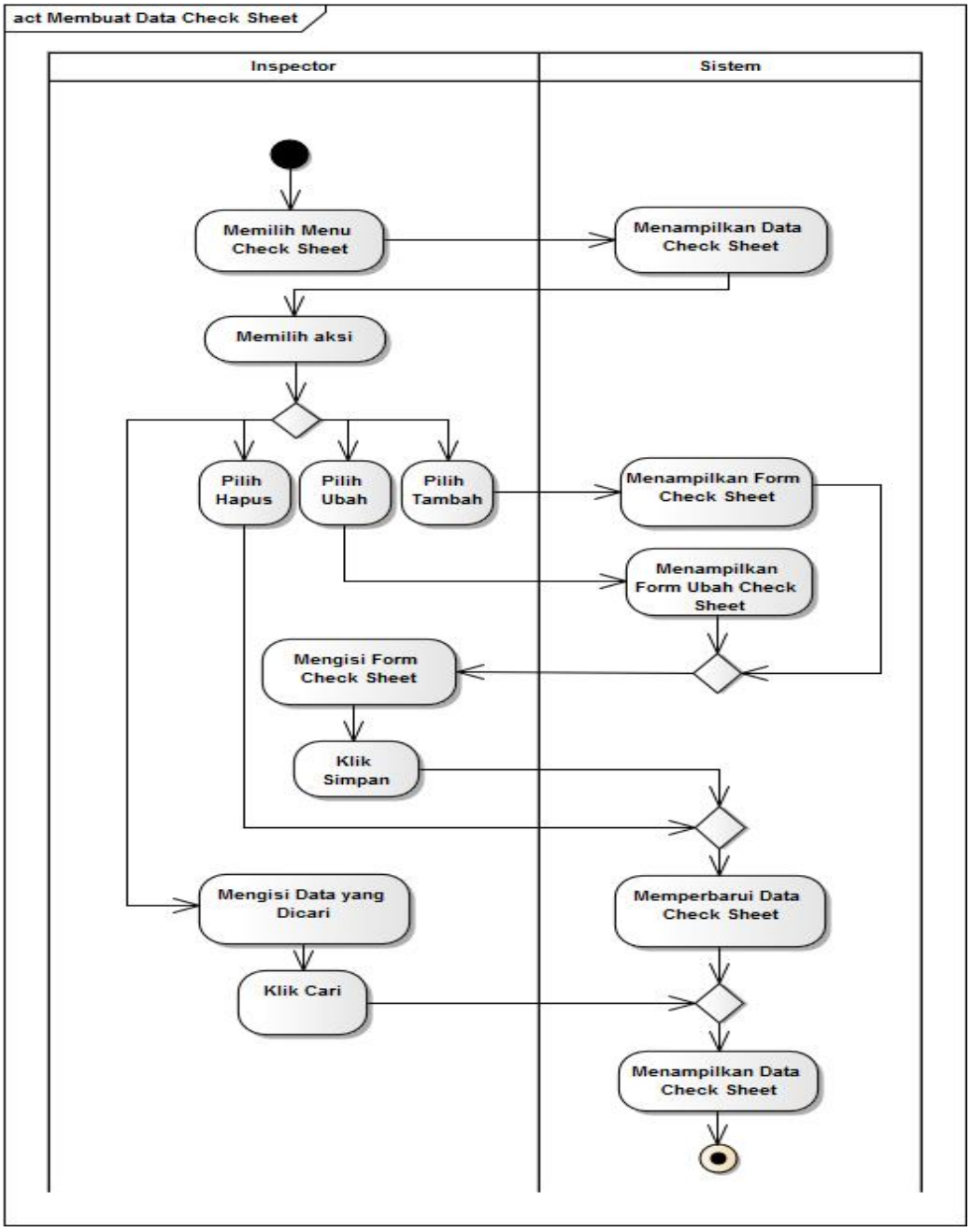
   Activity diagram mengelola data pelanggan ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat melakukan proses pengelolaan data pelanggan serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* mengelola data pelanggandapat dilihat pada Gambar V.6



Gambar V.6 *Activity Diagram* Mengelola Data Pelanggan
Sumber:Hasil Analisis (2019)

5. *Activity Diagram* Membuat Data *Check Sheet*

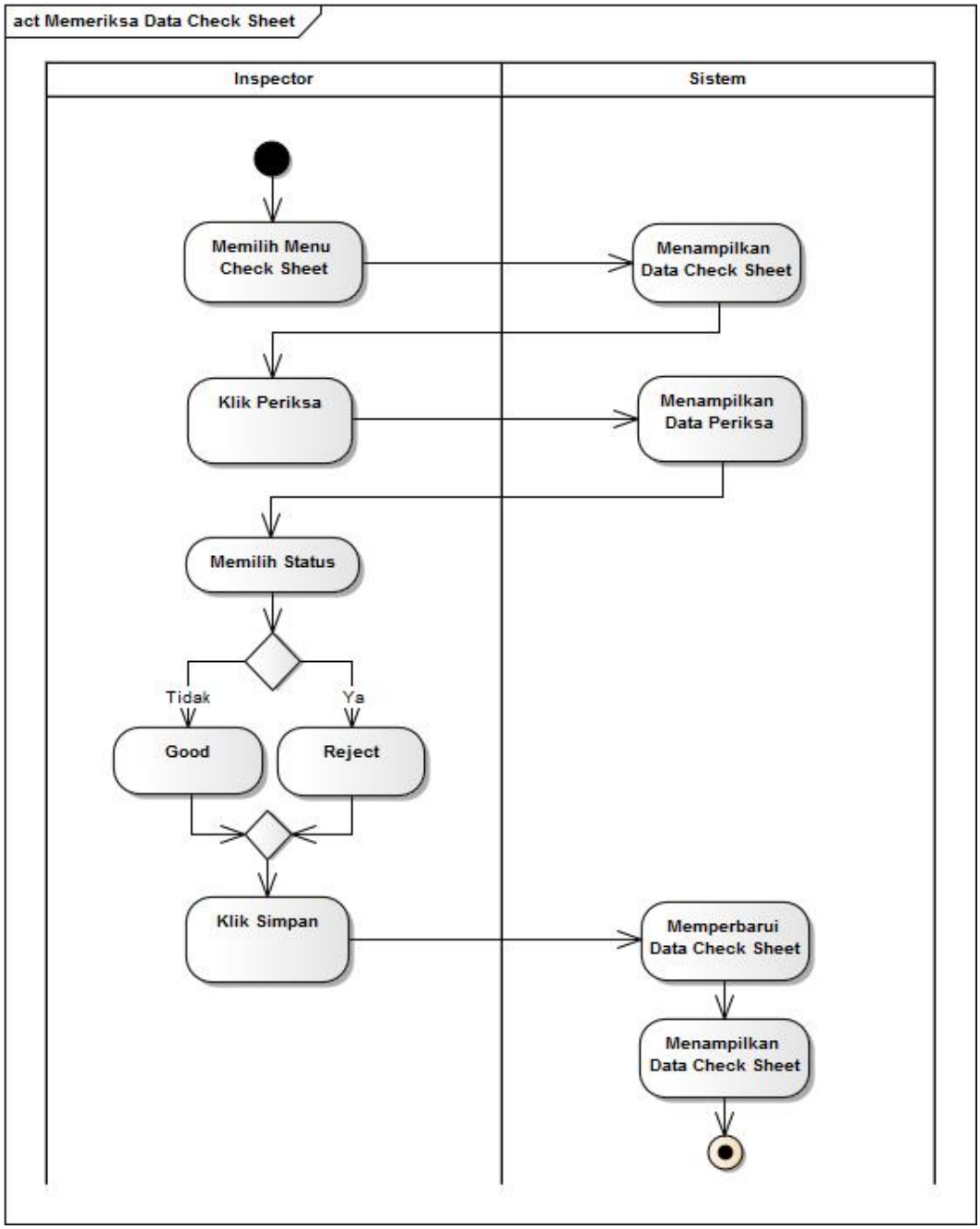
   Activity diagram membuat data *check sheet* ini mejelaskan alur aktivitas yang dilakukan oleh *inspector* untuk dapat menginput *check sheet* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* membuat data *check sheet* dapat dilihat pada Gambar V.7:



Gambar V.7 *Activity Diagram* Membuat Data *Check Sheet*
Sumber:Hasil Analisis (2019)

6. *Activity Diagram* Memeriksa Data *Check Sheet*

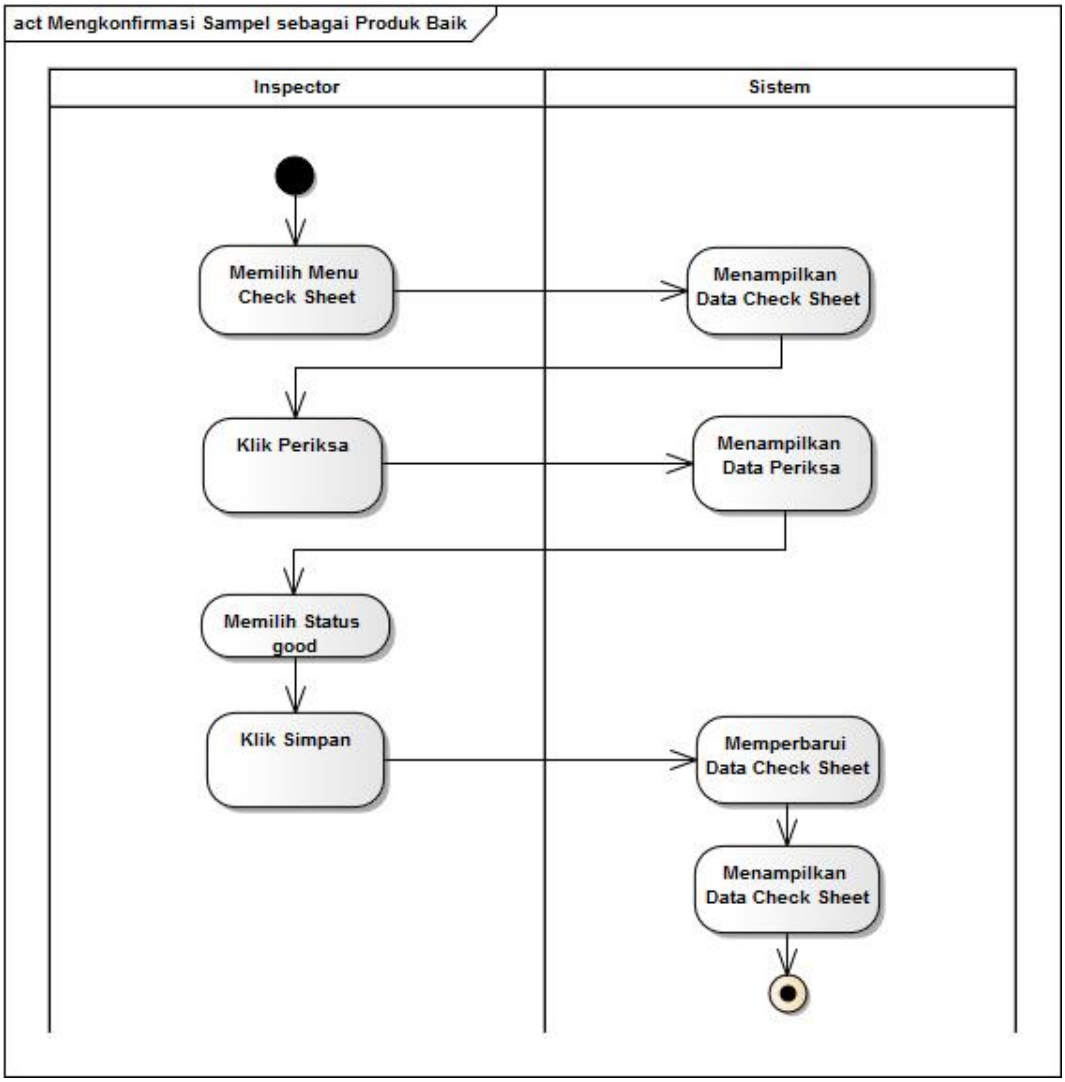
   Activity diagram memeriksa data *check sheet* ini mejelaskan alur aktivitas yang dilakukan oleh *inspector* untuk dapat memeriksa *check sheet* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* memeriksa data *check sheet* dapat dilihat pada Gambar V.8:



Gambar V.8 *Activity Diagram* Memeriksa Data *Check Sheet*
Sumber:Hasil Analisis (2019)

7. *Activity Diagram* Mengkonfirmasi Sampel sebagai Produk Baik

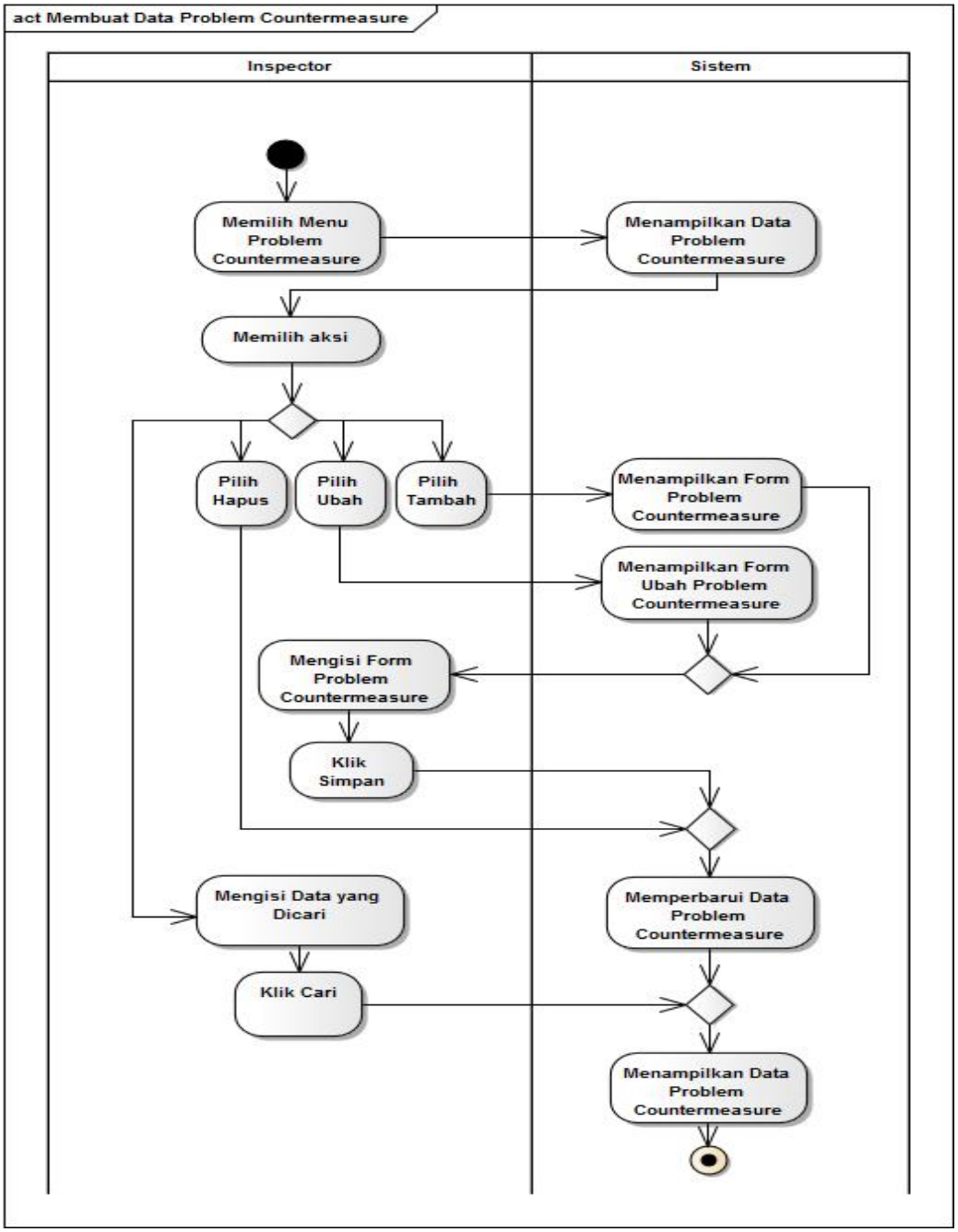
   Activity diagram mengkonfirmasi sampel sebagai produk baik ini mejelaskan alur aktivitas yang dilakukan oleh *inspector* untuk dapat mengkonfirmasi produk baik serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* membuat data *problem countermeasure* dapat dilihat pada Gambar V.9:



Gambar V.9 *Activity Diagram* Mengkonfirmasi Sampel sebagai Produk Baik
Sumber:Hasil Analisis (2019)

8. *Activity Diagram* Membuat Data *Problem Countermeasure*

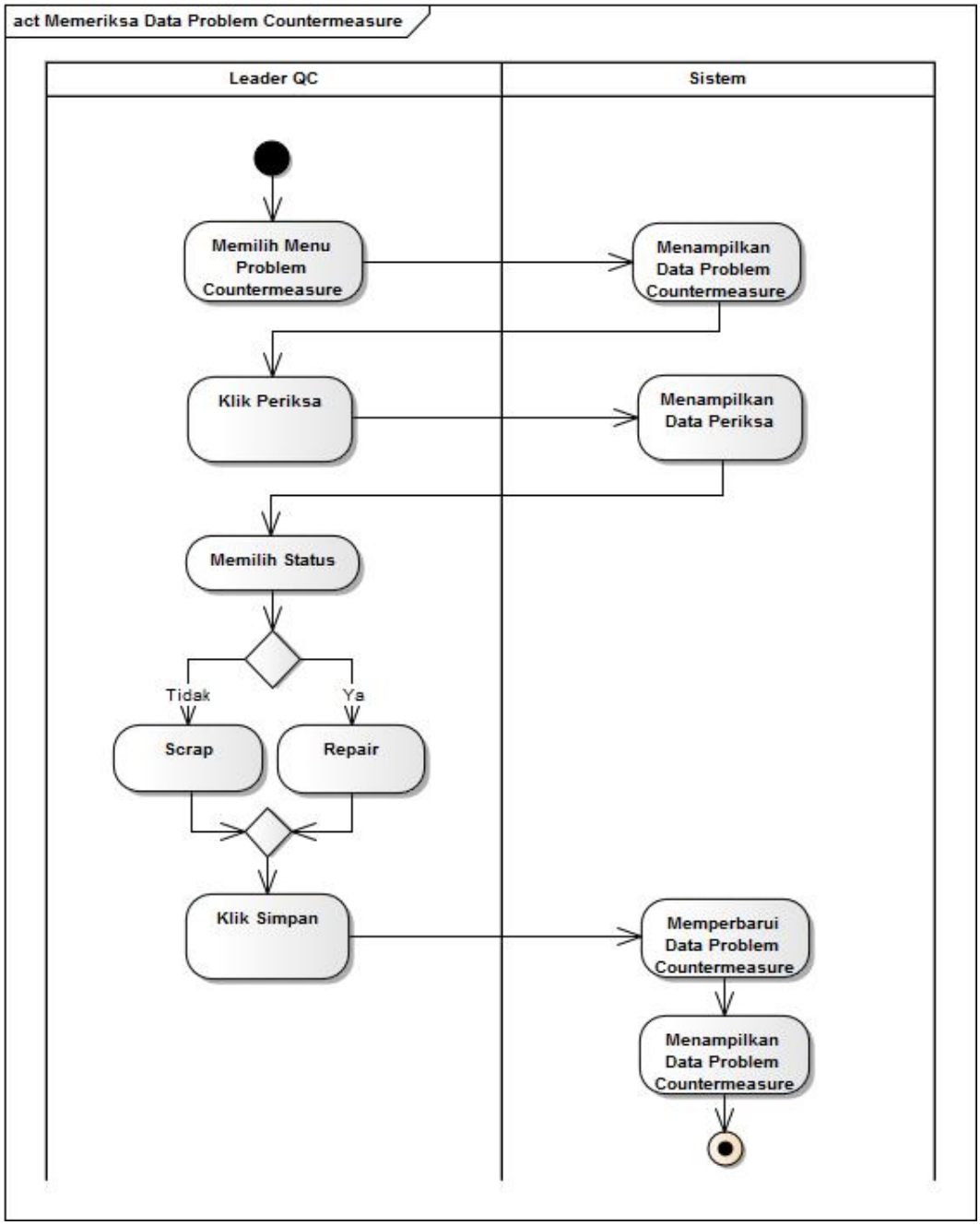
   Activity diagram membuat data *problem countermeasure* ini mejelaskan alur aktivitas yang dilakukan oleh *inspector* untuk dapat menginput *problem countermeasure* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* membuat data *problem countermeasure* dapat dilihat pada Gambar V.10:



Gambar V.10 *Activity Diagram* Membuat Data *Problem Countermeasure*
Sumber:Hasil Analisis (2019)

9. *Activity Diagram* Memeriksa Data *Problem Countermeasure*

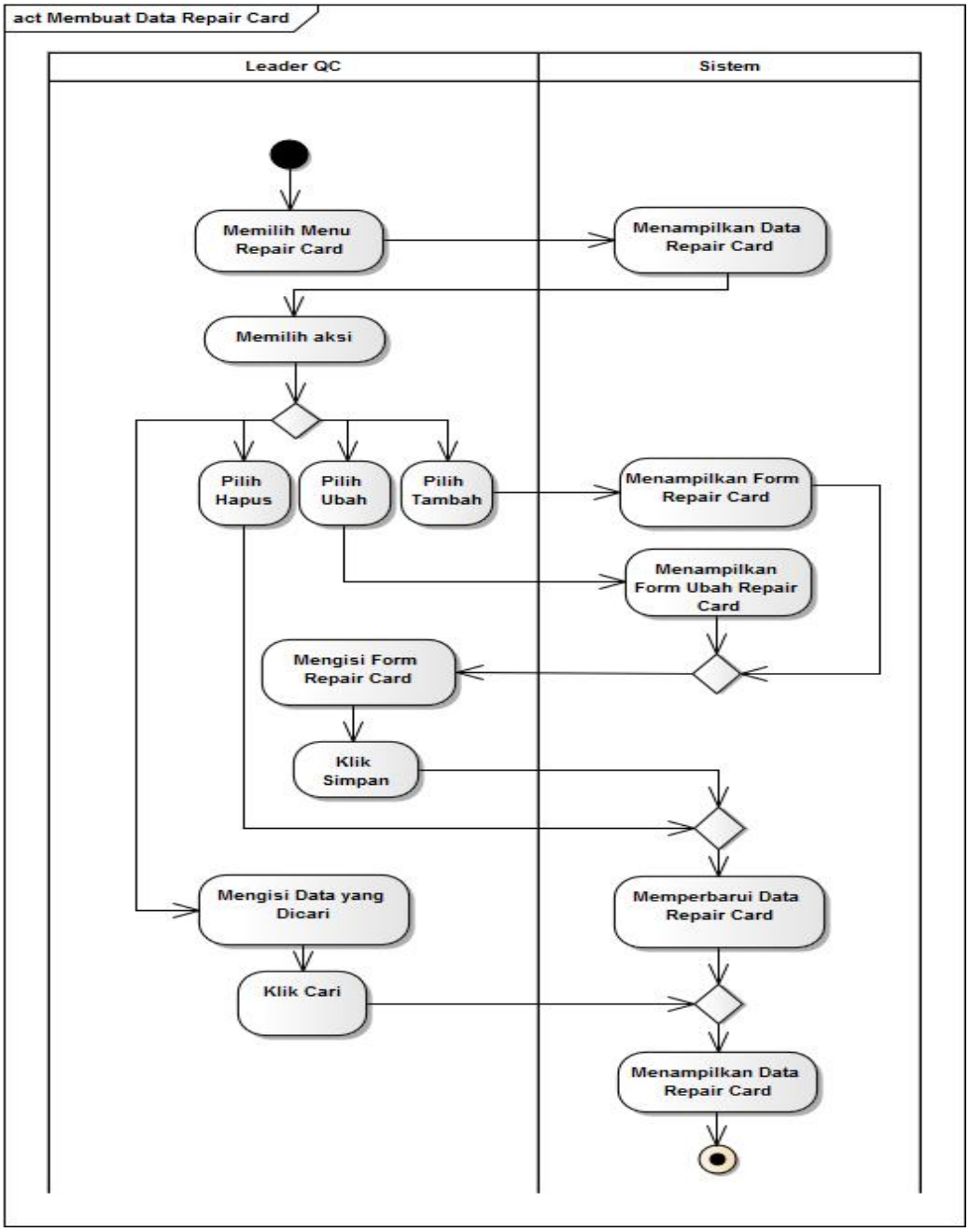
   Activity diagram memeriksa data *problem countermeasure* ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat memeriksa *problem countermeasure* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* memeriksa data *problem countermeasure* dapat dilihat pada Gambar V.11:



Gambar V.11 *Activity Diagram* Memeriksa *Data Problem Countermeasure*
Sumber:Hasil Analisis (2019)

10. *Activity Diagram* Membuat Data *Repair Card*

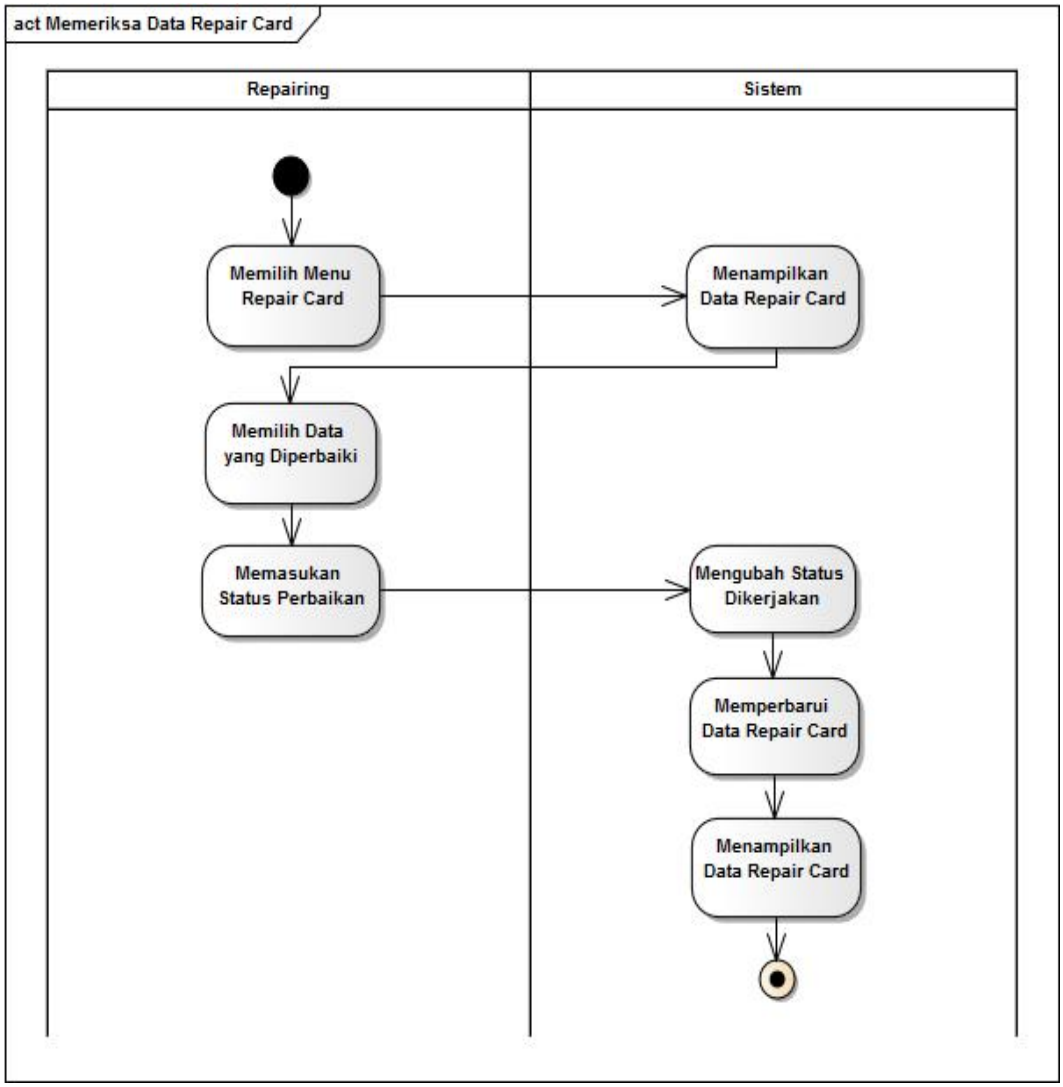
Activity diagram membuat data *repair card* ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat menginput *repair card* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* membuat data *repair card* dapat dilihat pada Gambar V.12:



Gambar V.12 *Activity Diagram* Membuat Data *Repair Card*
Sumber:Hasil Analisis (2019)

11. *Activity Diagram* Memeriksa Data *Repair Card*

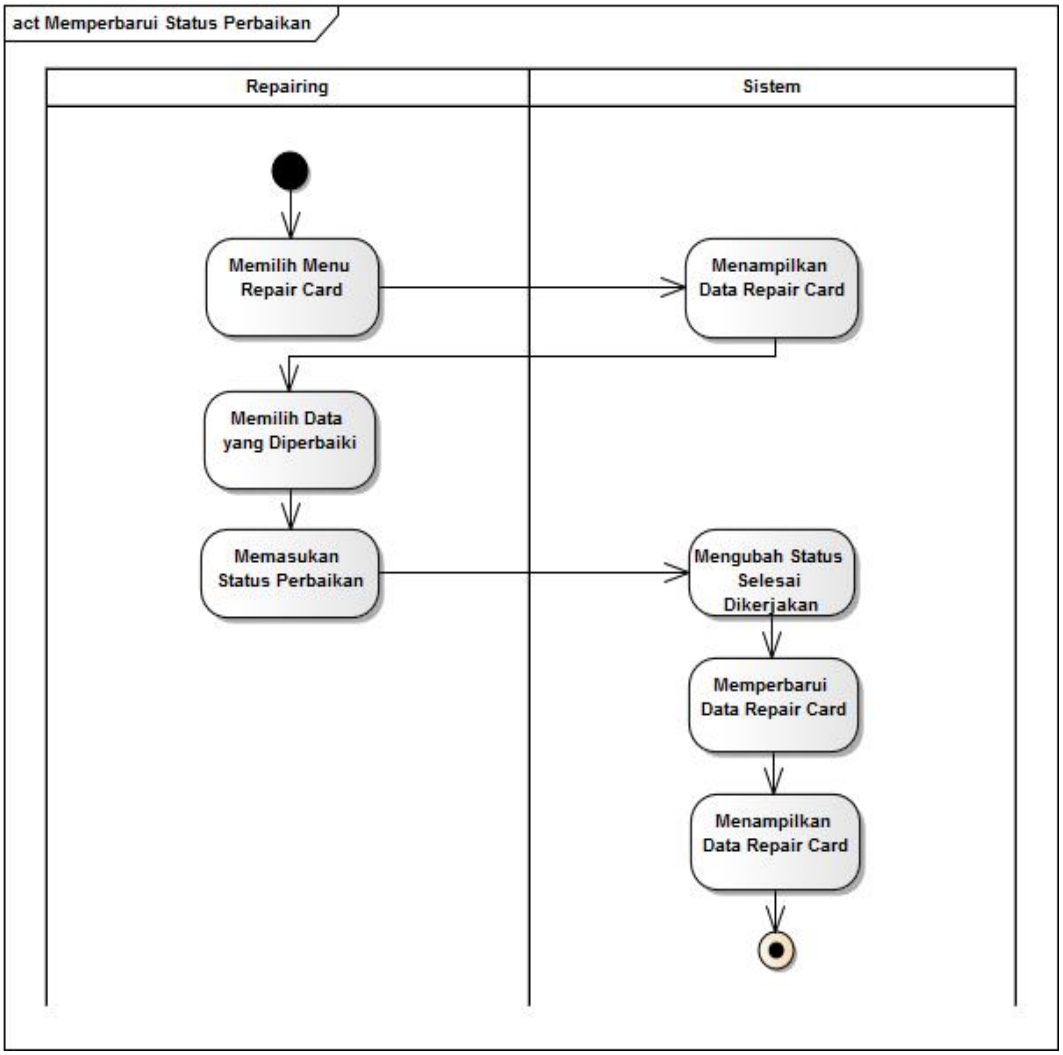
Activity diagram memeriksa data *repair card* ini mejelaskan alur aktivitas yang dilakukan oleh *repairing* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* memeriksa data *repair card* dapat dilihat pada Gambar V.13:



Gambar V.13 *Activity Diagram* Memeriksa Data *Repair Card*
Sumber:Hasil Analisis (2019)

12. *Activity Diagram* Memperbarui Status Perbaikan

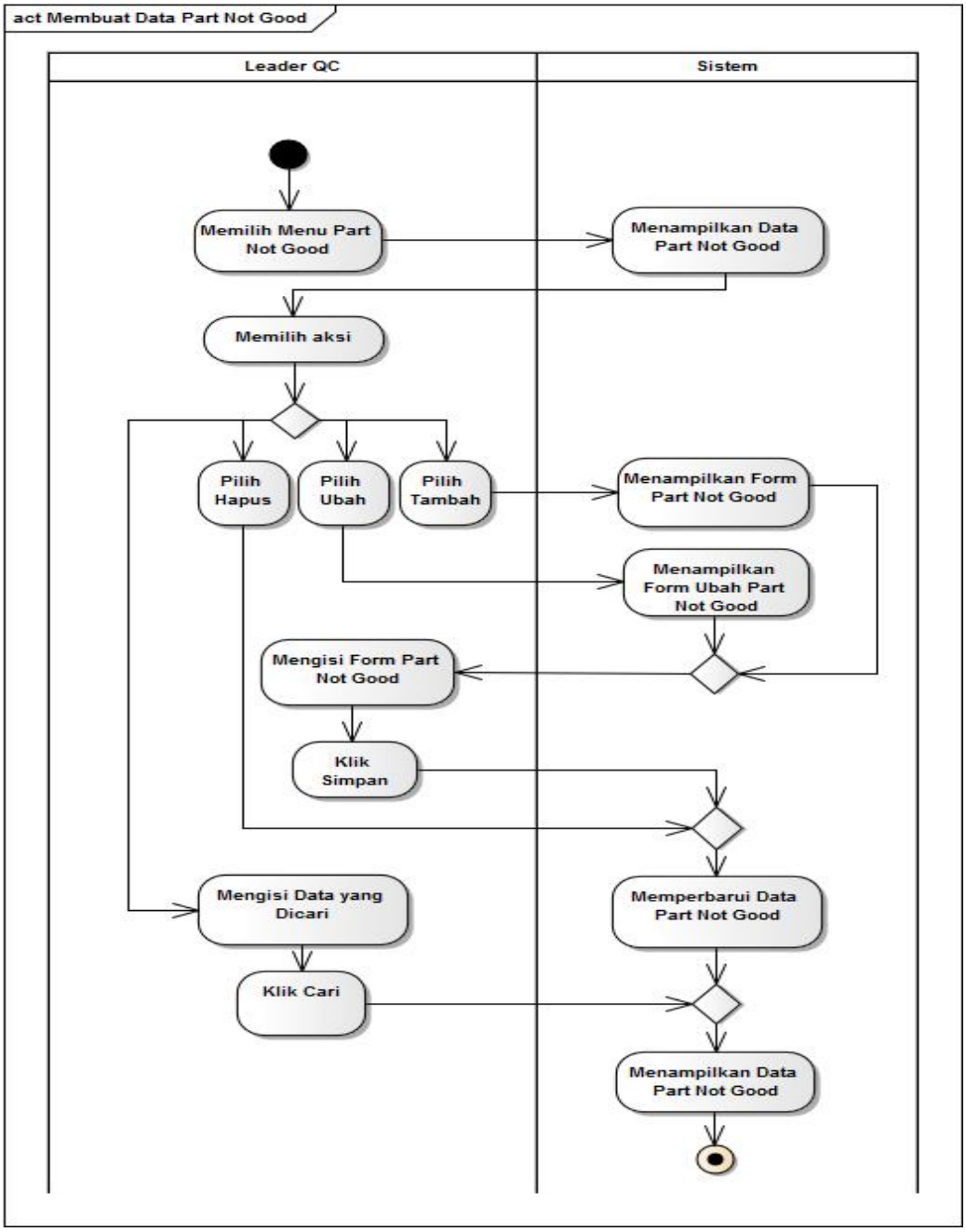
Activity diagram memperbarui status perbaikan ini mejelaskan alur aktivitas yang dilakukan oleh *repairing* untuk dapat merubah status perbaikan serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* memeriksa data *repair card* dapat dilihat pada Gambar V.14:



Gambar V.14 *Activity Diagram* Memperbarui Status Perbaikan
Sumber:Hasil Analisis (2019)

13. *Activity Diagram* Membuat Data *Part Not Good*

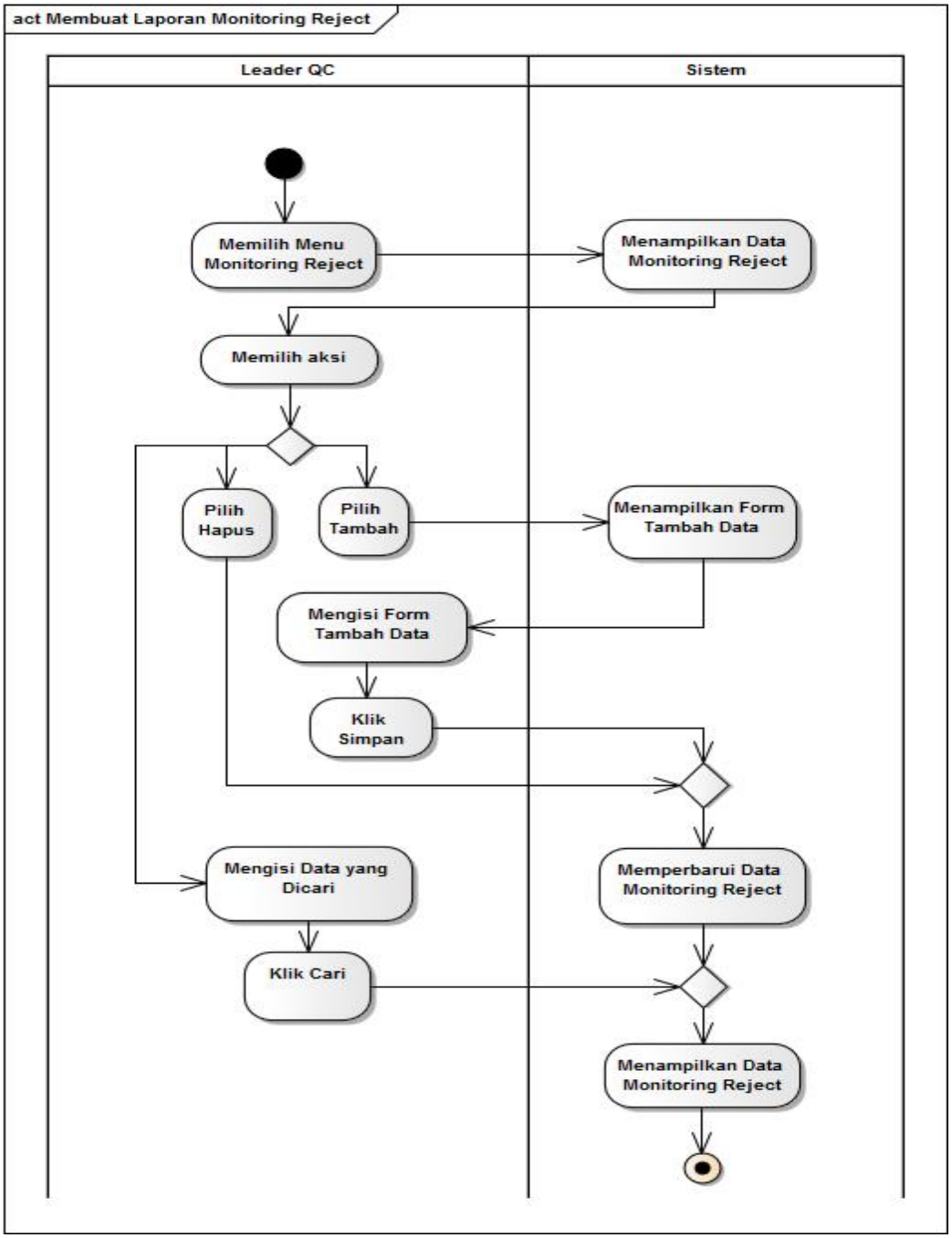
Activity diagram membuat data *part not good* ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat menginput data *part not good* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* membuat data *part not good* dapat dilihat pada Gambar V.15:



Gambar V.15 *Activity Diagram* Membuat Data *Part Not Good*
Sumber:Hasil Analisis (2019)

14. *Activity Diagram* Membuat Laporan *Monitoring Reject*

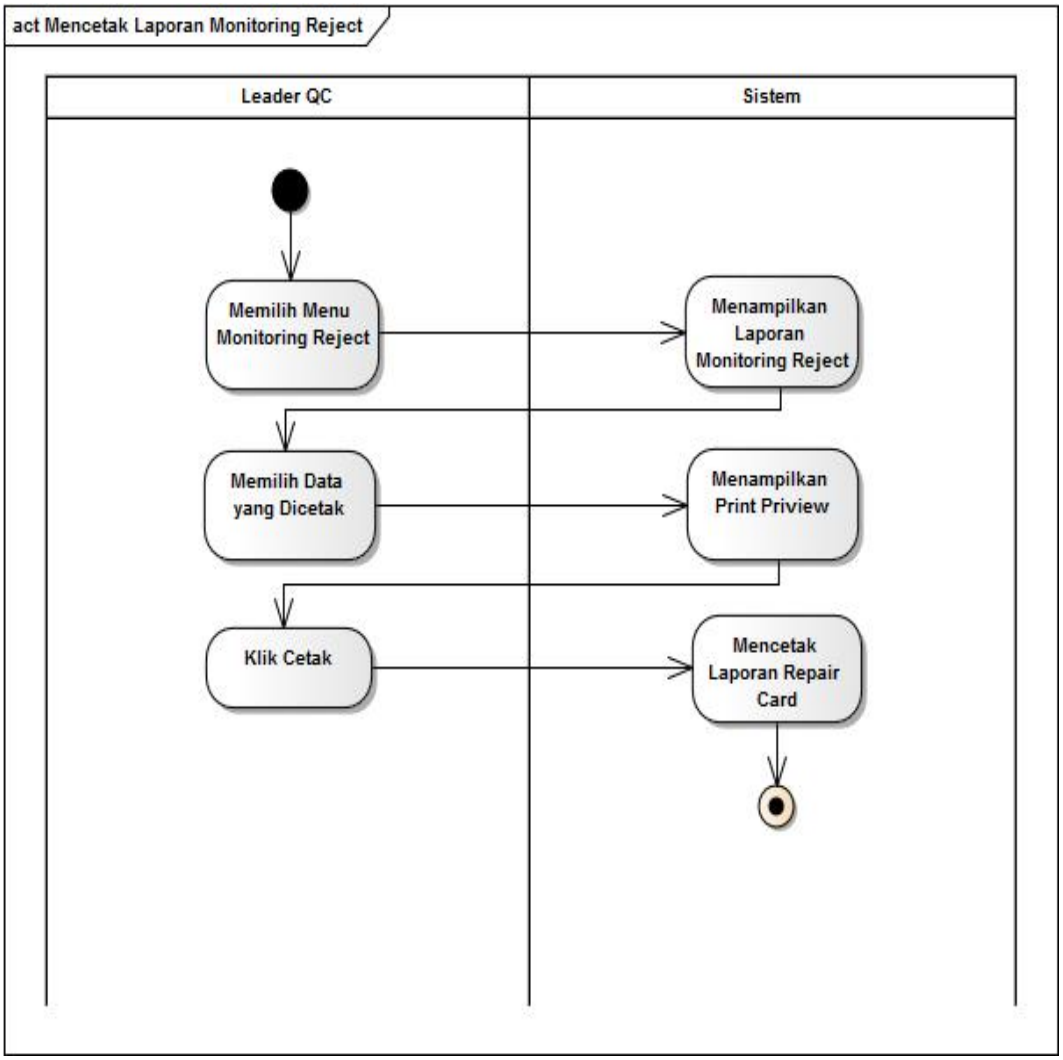
Activity diagram membuat laporan *monitoring reject* ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* untuk dapat merekap data *part not good* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* membuat laporan *monitoring reject* dapat dilihat pada Gambar V.16:



Gambar V.16 *Activity Diagram* Membuat Laporan *Monitoring Reject*
Sumber:Hasil Analisis (2019)

15. *Activity Diagram* Mencetak Laporan *Monitoring Reject*

Activity diagram mencetak laporan *monitoring reject* ini mejelaskan alur aktivitas yang dilakukan oleh *leader qc* serta fungsi-fungsi yang dapat dilakukan. *Activity diagram* mencetak laporan *monitoring reject* dapat dilihat pada Gambar V.17:



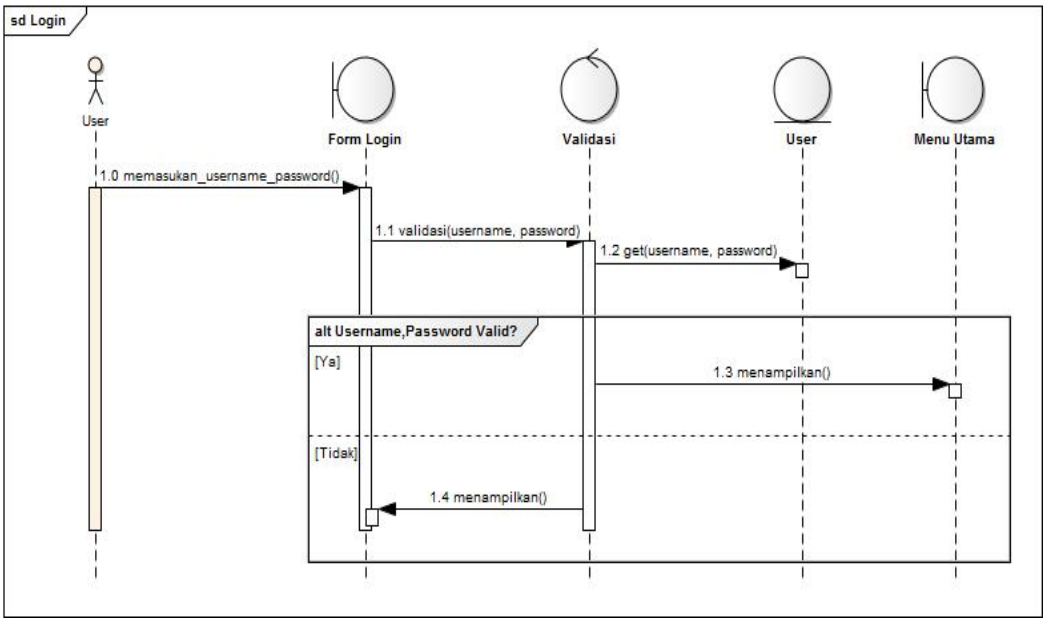
Gambar V.17 *Activity Diagram* Mencetak Laporan *Monitoring Reject*
Sumber:Hasil Analisis (2019)

5.3.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi yang terjadi pada suatu objek *use case diagram* ketika melakukan suatu proses tertentu, dimana urutan proses ketika melakukan suatu proses tertentu dapat diketahui dengan melihat gambaran pada diagram. Hubungan yang ada pada gambar di bawah ini adalah proses yang dilakukan oleh sistem ketika melakukan proses yang seusai dengan suatu objek *use case diagram* pada sistem informasi *monitoring quality control.*

1. *Sequence Diagram Login*

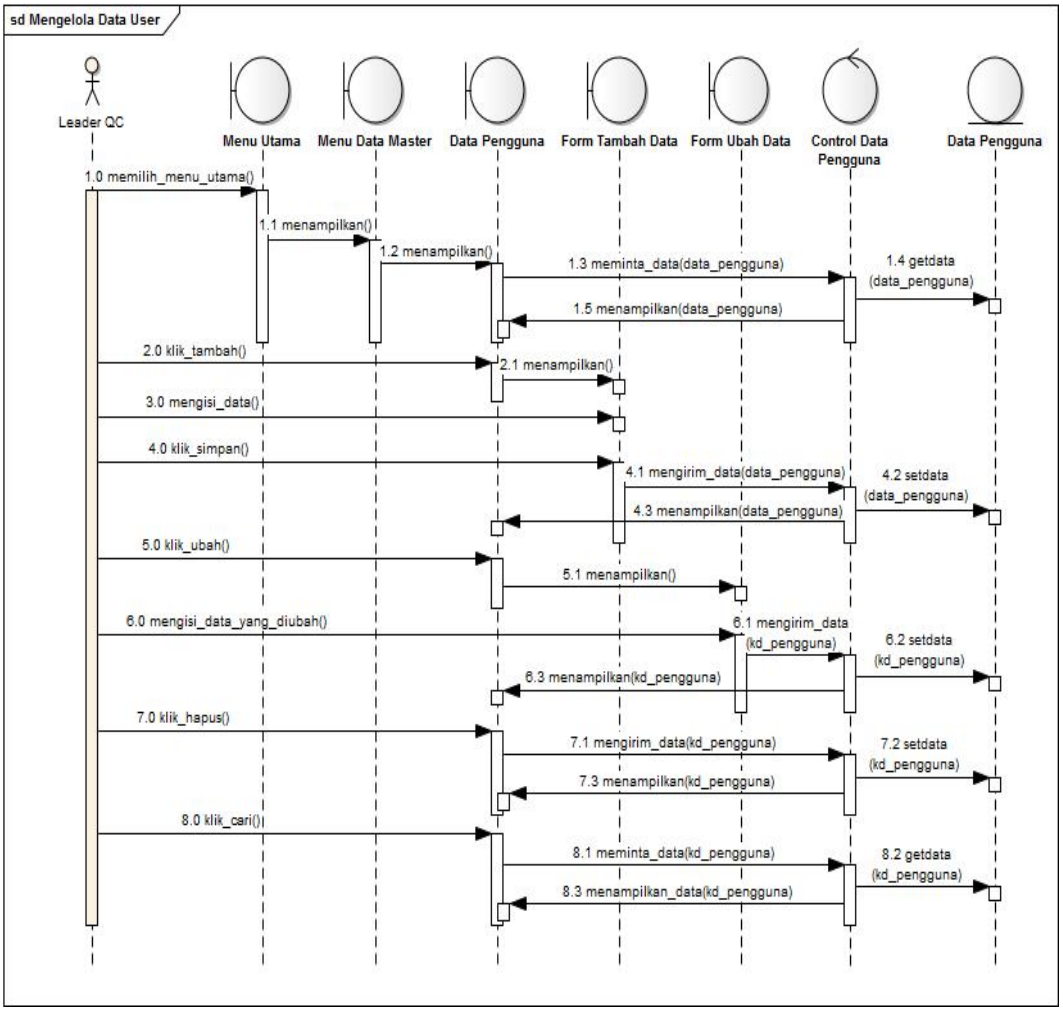
   Sequence diagram login menjelaskan interaksi objek-objek dalam sebuah proses *login.* Proses *login* dalam sistem informasi *monitoring quality control* langsung dilakukan oleh *user* agar dapat megakses sistem informasi. Berikut merupakan *sequence diagram login* dapat dilihat pada Gambar V.18:



Gambar V.18 *Sequence Diagram Login*
Sumber:Hasil Analisis (2019)

2. *Sequence Diagram* Mengelola Data Pengguna

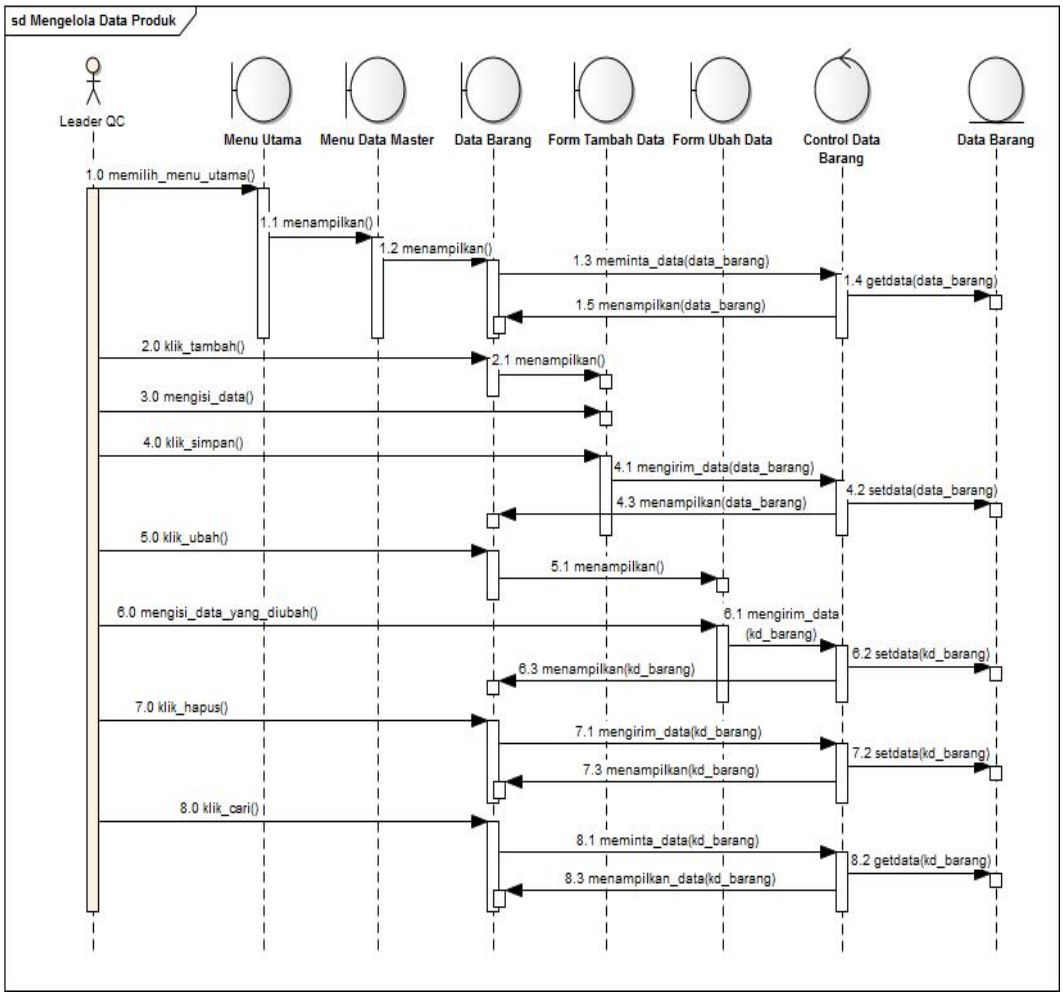
   Sequence diagram mengelola data pengguna menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk mengelola data pengguna. Berikut merupakan *sequence diagram* mengelola data pengguna dapat dilihat pada Gambar V.19:



Gambar V.19 *Sequence Diagram* Mengelola Data Pengguna
Sumber:Hasil Analisis (2019)

3. *Sequence Diagram* Mengelola Data Barang

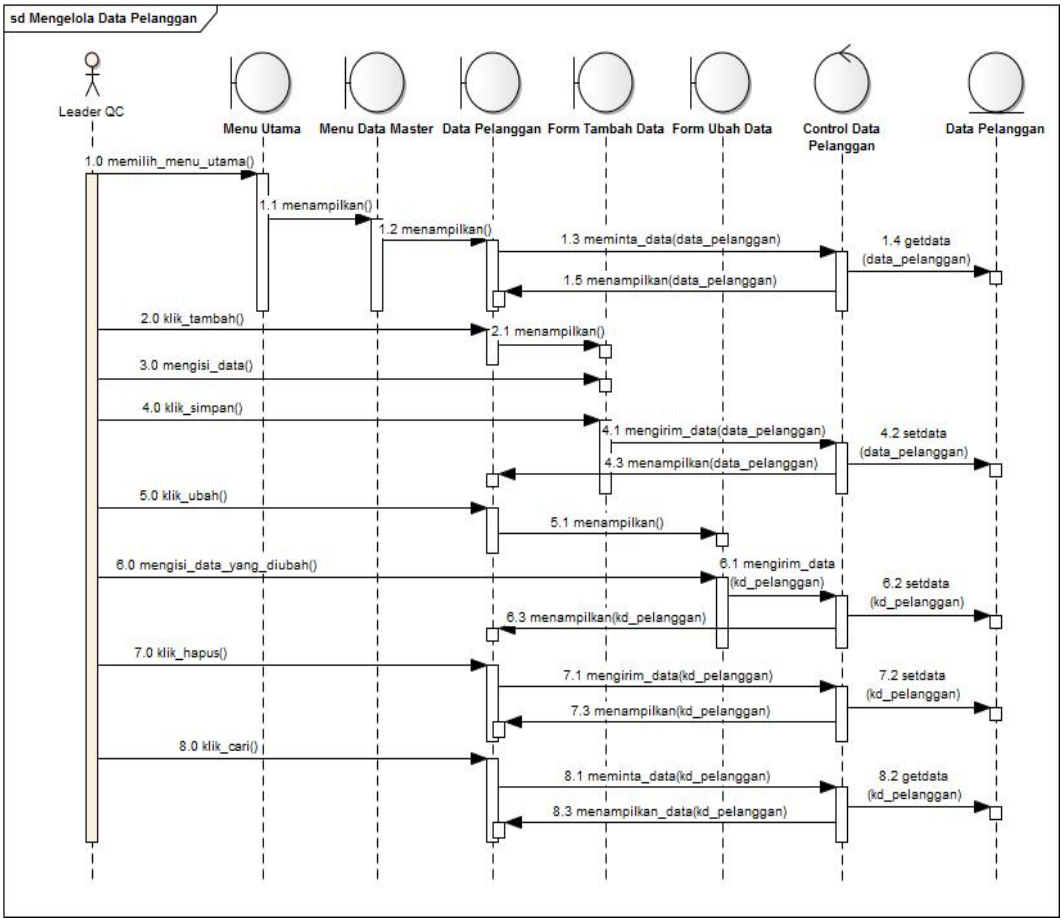
   Sequence diagram mengelola data barang menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk mengelola data barang. Berikut merupakan *sequence diagram* mengelola data barang dapat dilihat pada Gambar V.20:



Gambar V.20 *Sequence Diagram* Mengelola Data Barang
Sumber:Hasil Analisis (2019)

4. *Sequence Diagram* Mengelola Data Pelanggan

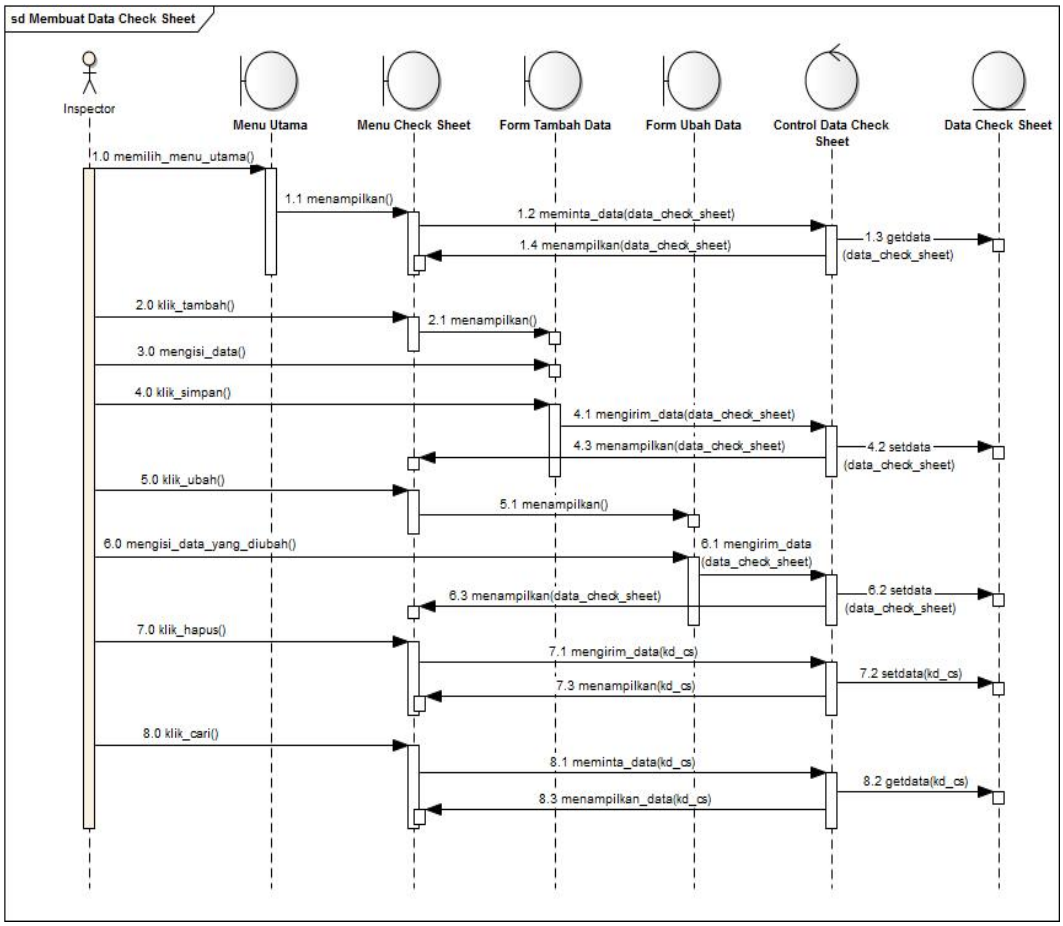
   Sequence diagram mengelola data pelanggan menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk mengelola data pelanggan. Berikut merupakan *sequence diagram* mengelola data pelanggan dapat dilihat pada Gambar V.21:



Gambar V.21 *Sequence Diagram* Mengelola Data Pelanggan
Sumber:Hasil Analisis (2019)

5. *Sequence Diagram* Membuat Data *Check Sheet*

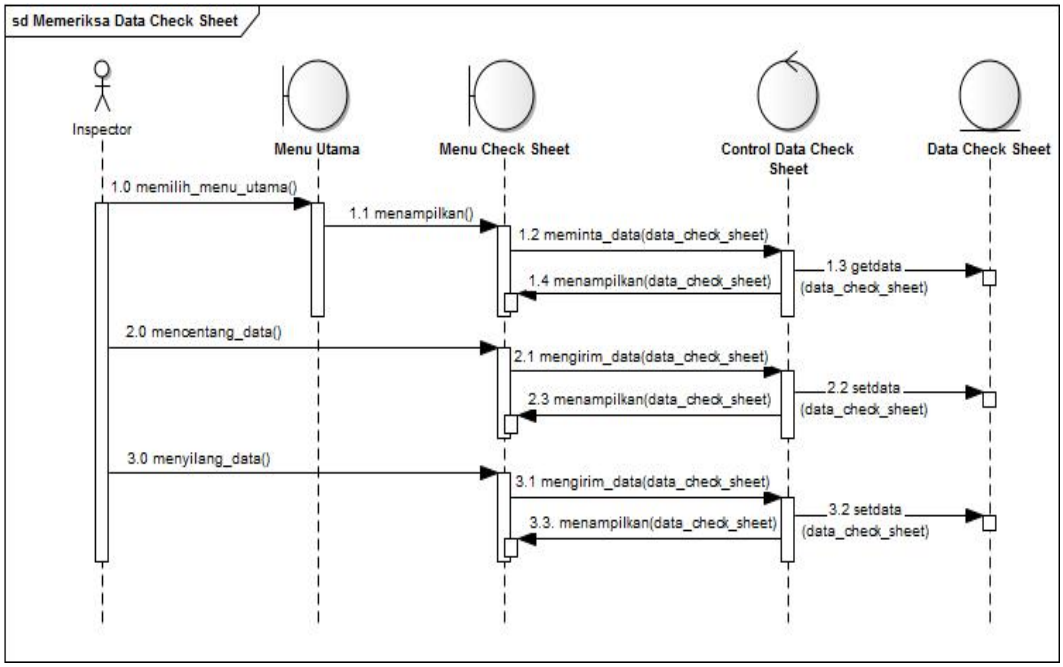
   Sequence diagram membuat data *check sheet* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *inspector* untuk menginput *check sheet.* Berikut merupakan *sequence diagram* membuat data *check sheet* dapat dilihat pada Gambar V.22:



Gambar V.22 *Sequence Diagram* Membuat Data *Check Sheet*
Sumber:Hasil Analisis (2019)

6. *Sequence Diagram* Memeriksa Data *Check Sheet*

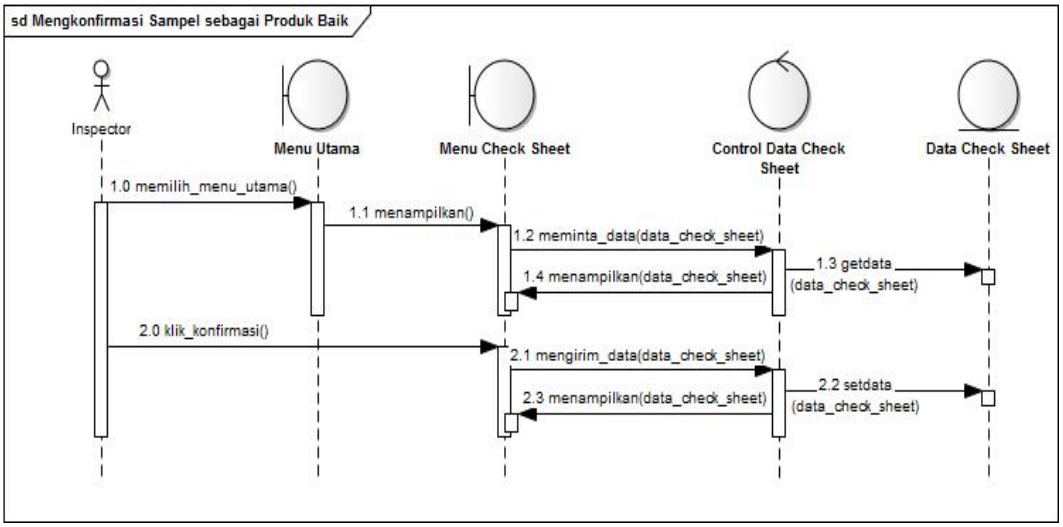
   Sequence diagram memeriksa data *check sheet* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *inspector* untuk memeriksa *check sheet.* Berikut merupakan *sequence diagram* memeriksa data *check sheet* dapat dilihat pada Gambar V.23:



Gambar V.23 *Sequence Diagram* Memeriksa Data *Check Sheet*
Sumber:Hasil Analisis (2019)

7. *Sequence Diagram* Mengkonfirmasi Sampel sebagai Produk Baik

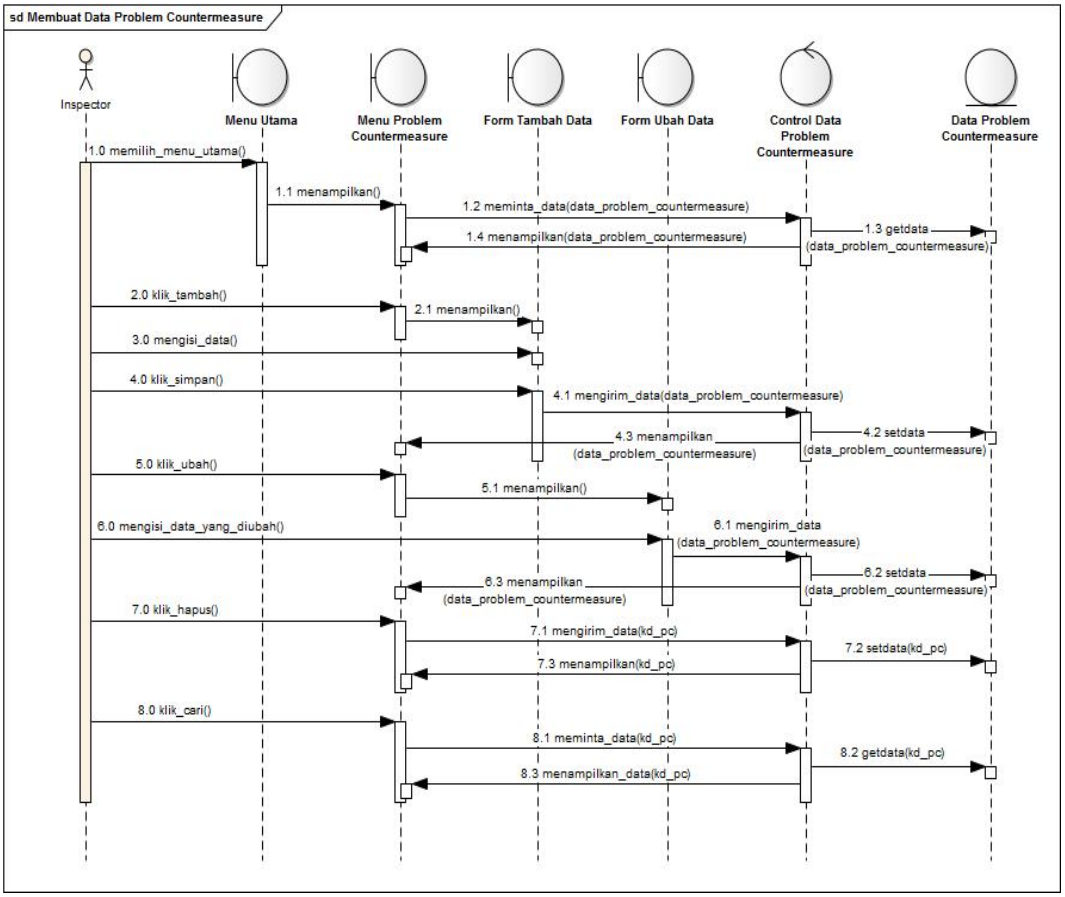
   Sequence diagram mengkonfirmasi sampel sebagai produk baik menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *inspector* untuk mengkonfirmasi produk yang baik. Berikut merupakan *sequence diagram* mengkonfirmasi sampel sebagai produk baik dapat dilihat pada Gambar V.24:



Gambar V.24 *Sequence Diagram* Mengkonfirmasi Sampel sebagai Produk Baik
Sumber:Hasil Analisis (2019)

8. *Sequence Diagram* Membuat Data *Problem Countermeasure*

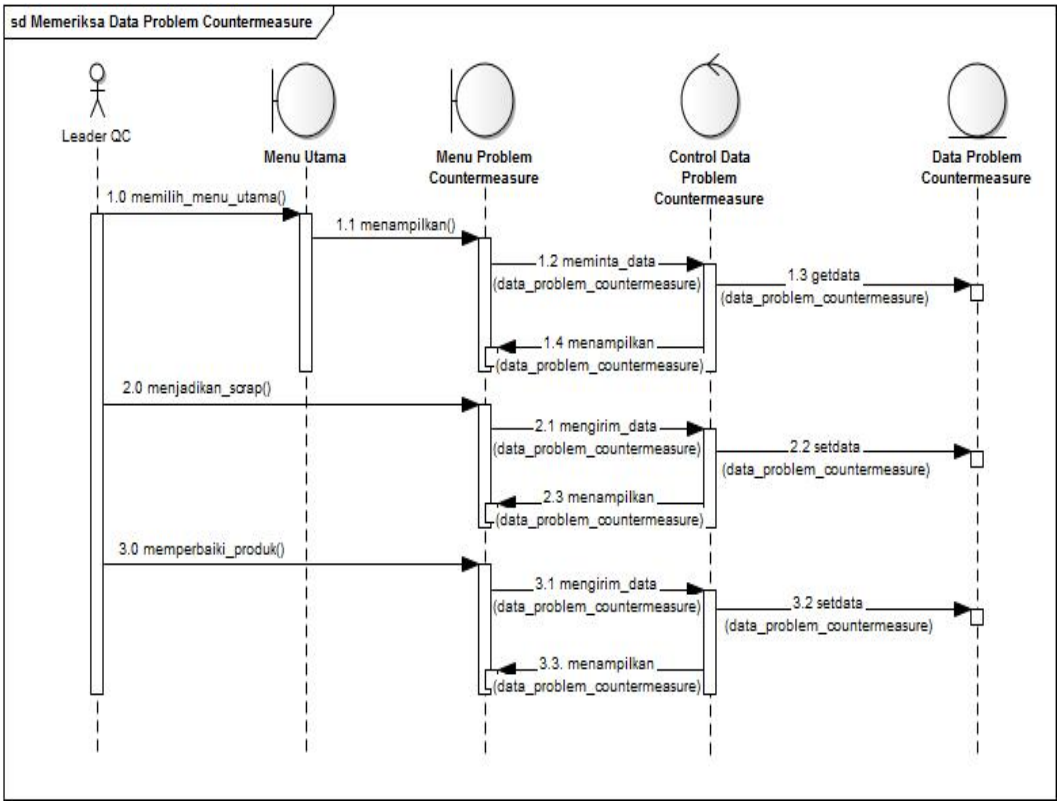
Sequence diagram membuat data *problem countermeasure* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *inspector* untuk menginput *problem countermeasure.* Berikut merupakan *sequence diagram* membuat data *problem countermeasure* dapat dilihat pada Gambar V.25:



Gambar V.25 *Sequence Diagram* Membuat Data *Problem Countermeasure*
Sumber:Hasil Analisis (2019)

9. *Sequence Diagram* Memeriksa Data *Problem Countermeasure*

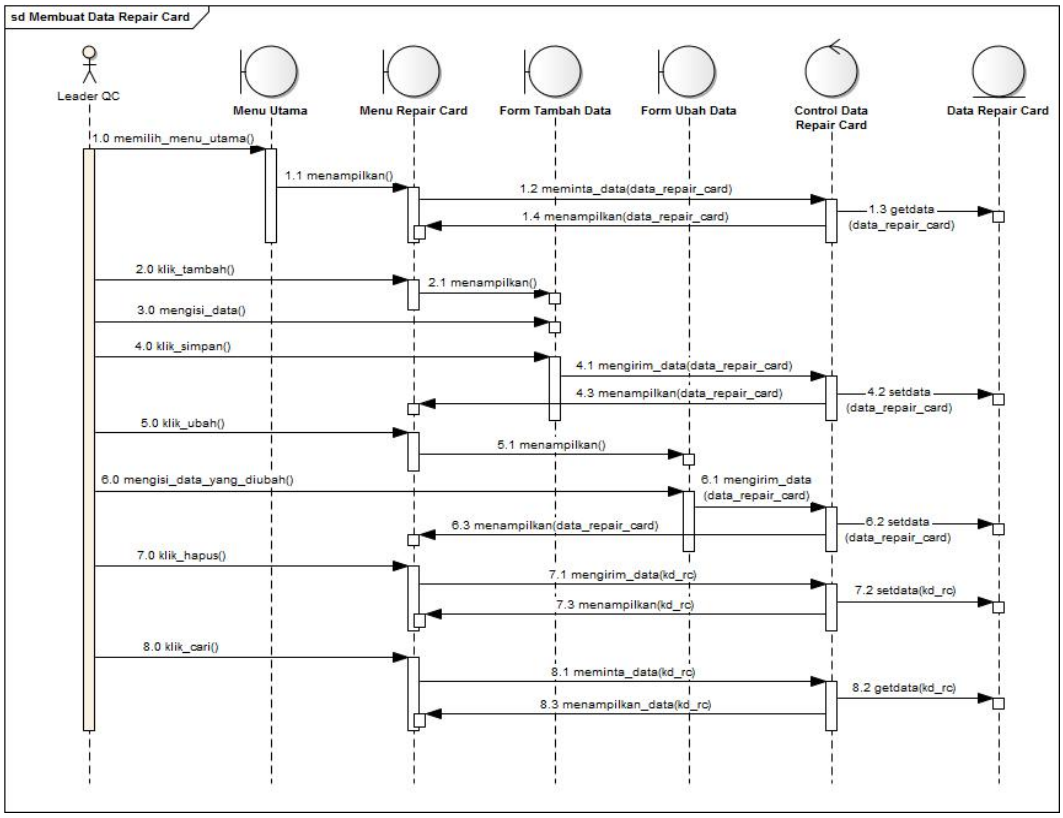
Sequence diagram memeriksa data *problem countermeasure* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk memeriksa *problem countermeasure.* Berikut merupakan *sequence diagram* memeriksa data *problem countermeasure* dapat dilihat pada Gambar V.26:



Gambar V.26 *Sequence Diagram* Memeriksa Data *Problem Countermeasure*
Sumber:Hasil Analisis (2019)

10. *Sequence Diagram* Membuat Data *Repair Card*

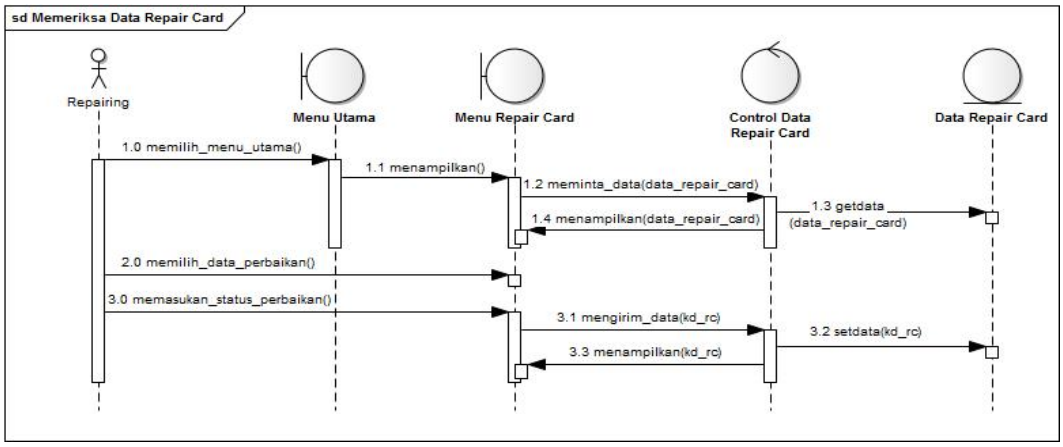
Sequence diagram membuat data *repair card* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk menginput *repair card.* Berikut merupakan *sequence diagram* membuat data *repair card* dapat dilihat pada Gambar V.27:



Gambar V.27 *Sequence Diagram* Membuat Data *Repair Card*
Sumber:Hasil Analisis (2019)

11. *Sequence Diagram* Memeriksa Data *Repair Card*

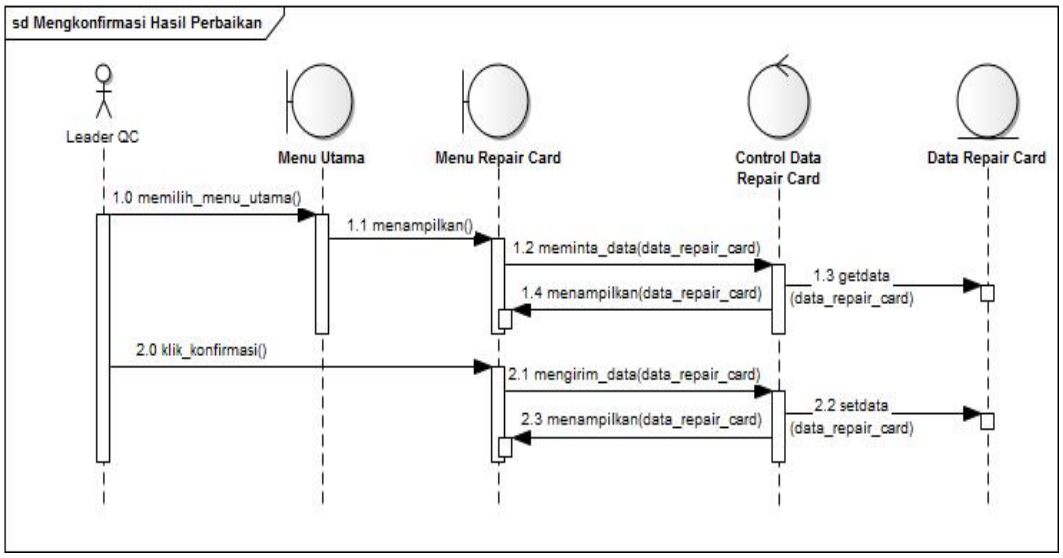
Sequence diagram memeriksa data *repair card* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *repairing* untuk memeriksa data *repair card.* Berikut merupakan *sequence diagram* memeriksa data *repair card* dapat dilihat pada Gambar V.28:



Gambar V.28 *Sequence Diagram* Memeriksa Data *Repair Card*
Sumber:Hasil Analisis (2019)

12. *Sequence Diagram* Mengkonfirmasi Hasil Perbaikan

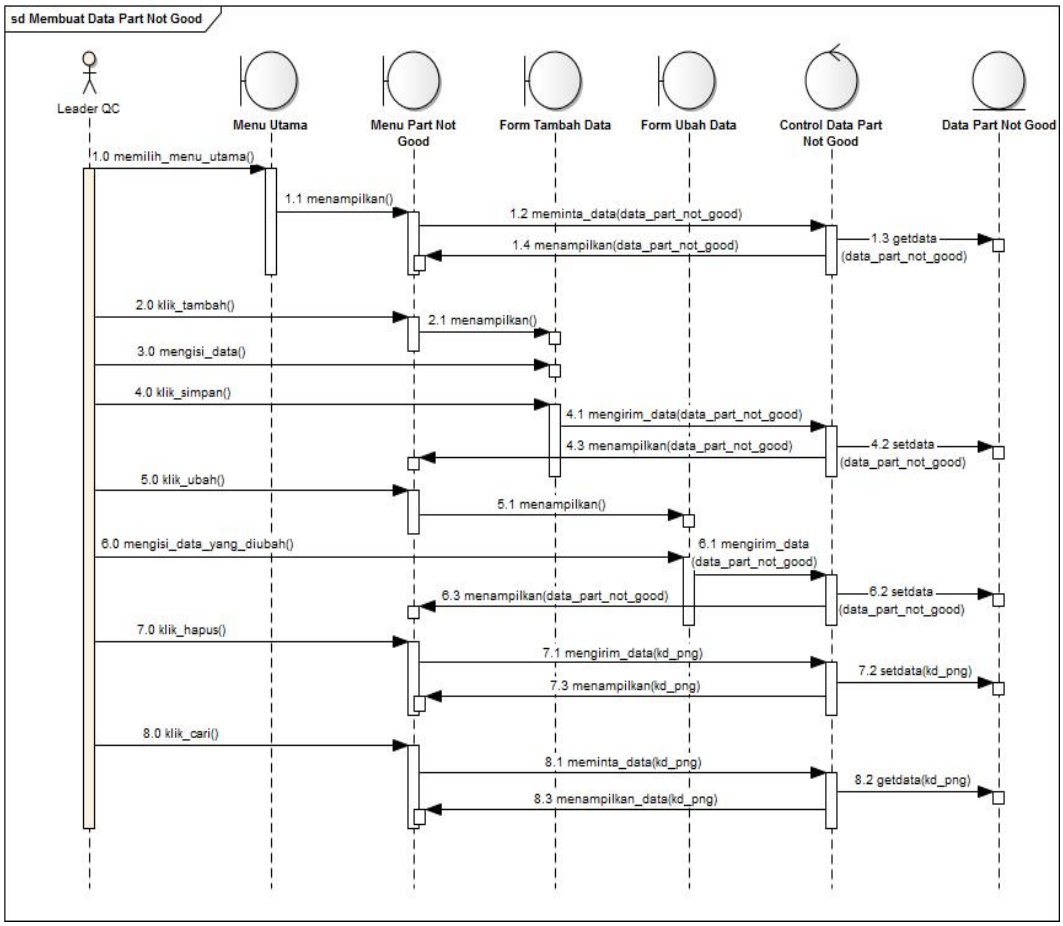
Sequence diagram mengkonfirmasi hasil perbaikan menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk mengkonfirmasi hasil perbaikan. Berikut merupakan *sequence diagram* mengkonfirmasi hasil perbaikan dapat dilihat pada Gambar V.29:



Gambar V.29 *Sequence Diagram* Mengkonfirmasi Hasil Perbaikan
Sumber:Hasil Analisis (2019)

13. *Sequence Diagram* Membuat Data *Part Not Good*

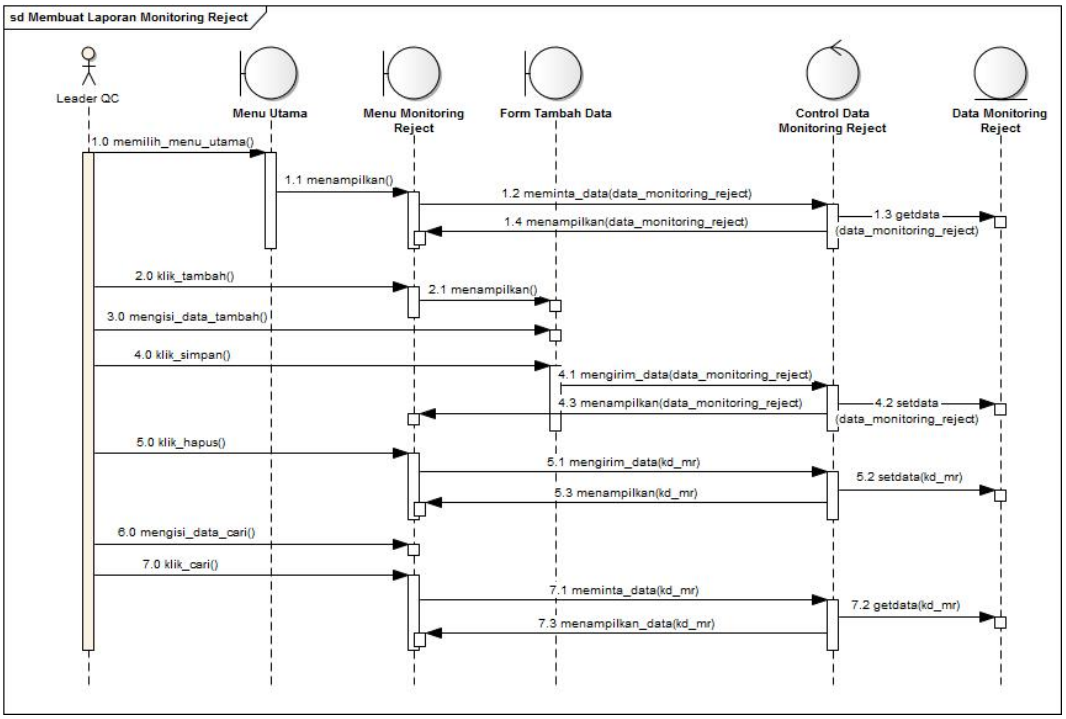
Sequence diagram membuat data *part not good* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk menginput data *part not good.* Berikut merupakan *sequence diagram* membuat data *part not good* dapat dilihat pada Gambar V.30:



Gambar V.30 *Sequence Diagram* Membuat Data *Part Not Good*
Sumber:Hasil Analisis (2019)

14. *Sequence Diagram* Membuat Laporan *Monitoring Project*

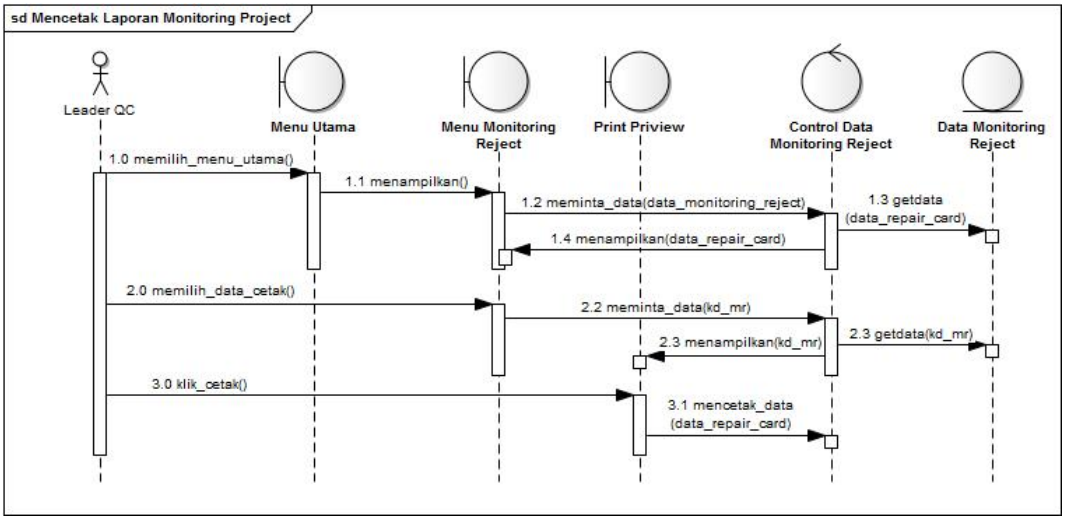
Sequence diagram membuat laporan *monitoring project* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk merekap data *part not good.* Berikut merupakan *sequence diagram* membuat laporan *Monitoring Project* dapat dilihat pada Gambar V.31:



Gambar V.31 *Sequence Diagram* Membuat Laporan *Monitoring Reject*
Sumber:Hasil Analisis (2019)

15. *Sequence Diagram* Mencetak Laporan *Monitoring Project*

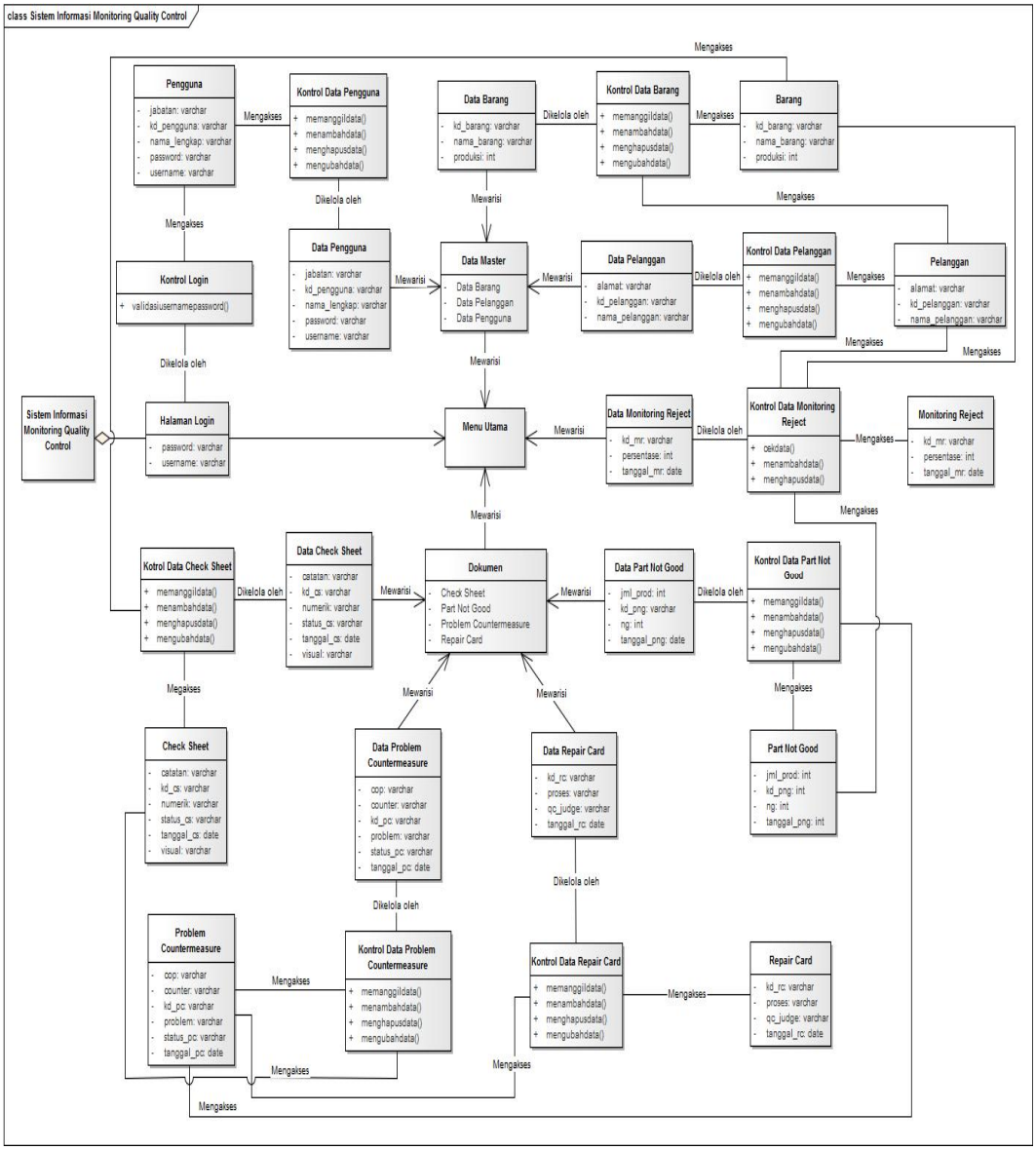
Sequence diagram mencetak laporan *monitoring project* menjelaskan interaksi objek-objek dalam sebuah proses yang dilakukan *leader qc* untuk mencetak laporan *monitoring project.* Berikut merupakan *sequence diagram* mencetak laporan *Monitoring Project* dapat dilihat pada Gambar V.32:



Gambar V.32 *Sequence Diagram* Mencetak Laporan *Monitoring Reject*
Sumber:Hasil Analisis (2019)

5.3.4 *Class Diagram*

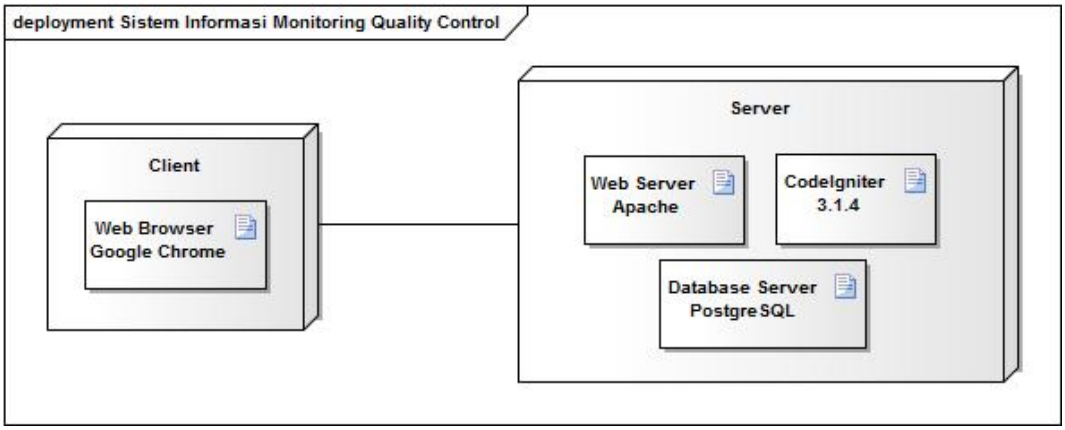
Class diagram pada usulan sistem informasi *monitoring quality control* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat, sistem *class diagram* memperlihatkan hubungan antar kelas dan penjelasan *detail* tiap-tiap kelas di dalam model desain dari suatu sistem. *Class diagram* sistem informasi *monitoring quality control* yang diusulkan dapat dilihat pada Gambar V.33:



Gambar V.33 *Class Diagram Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

5.3.5 *Deployment Diagram*

Deployment diagram pada usulan sistem informasi *monitoring quality control* digunakan untuk menggambarkan hubungan antar komponen *hardware* yang digunakan dan *software* dalam insfrastruktur fisik dari suatu sistem informasi. *Deployment diagram* sistem informasi *monitoring quality control* yang diusulkan dapat dilihat pada Gambar V.34:



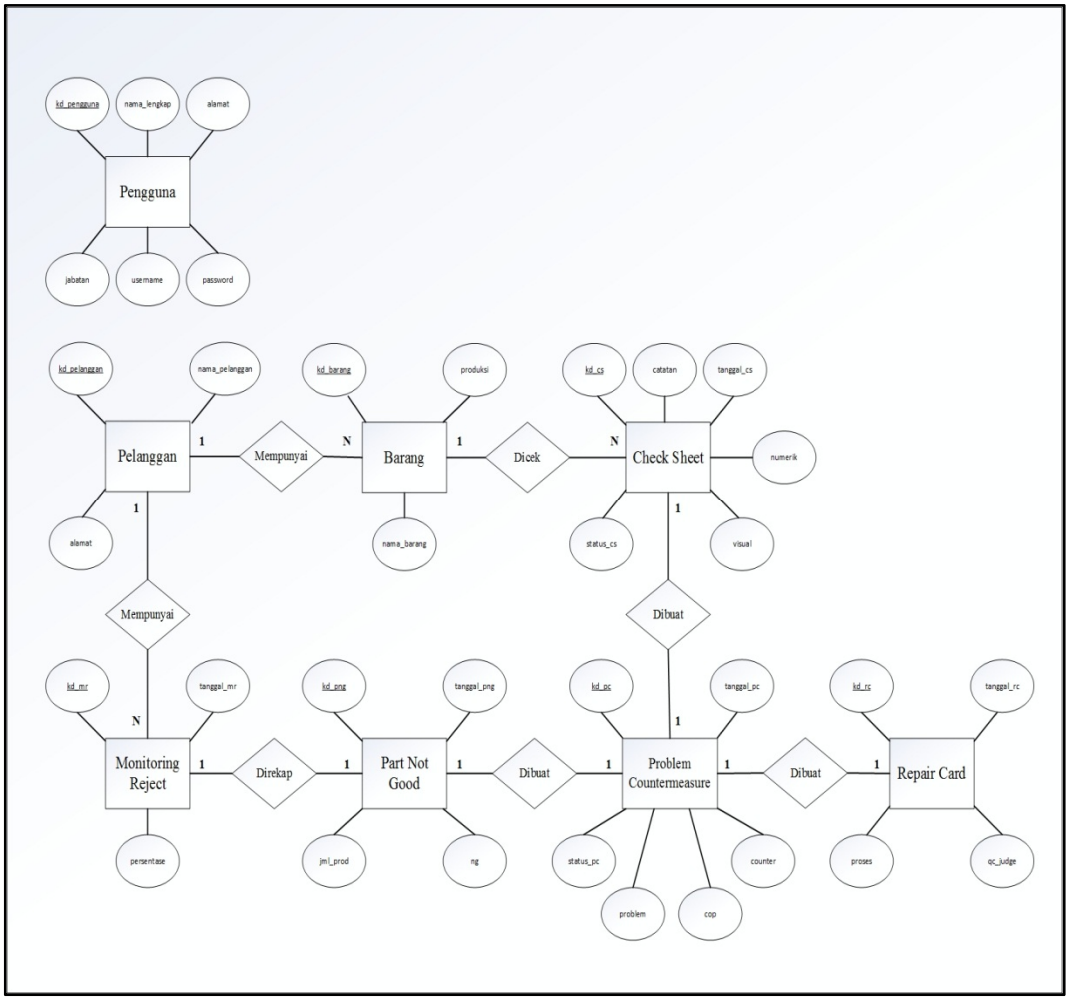
Gambar V.34 *Deployment Diagram Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

5.4 Pemodelan Data

Pemodelan data pada sistem sistem informasi *monitoring quality control* di PT Nusa Indah Jaya Utama yaitu menggunakan *Entity Relationship Diagram* (ERD) untuk menjelaskan hubungan antar data dalam basis data dan kamus data yang digunakan untuk menjelaskan isi daru basis data yang digunakan dalam sistem usulan.

5.4.1 *Entity Relationship Diagram*

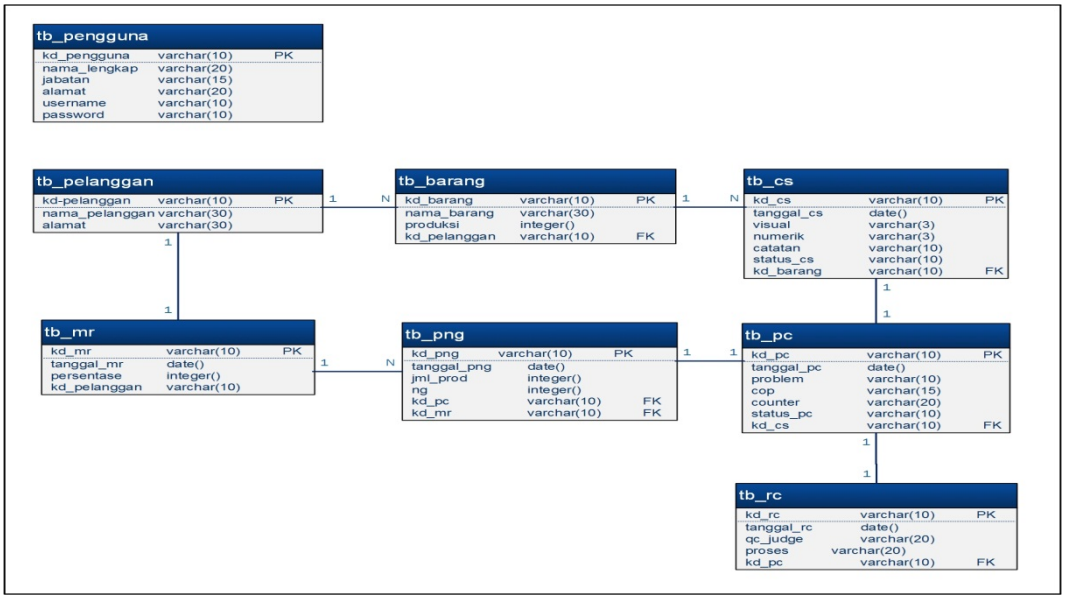
Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Untuk menggambarkannya digunakan beberapa notasi dan simbol. ERD sistem informasi *monitoring quality control* yang diusulkan dapat dilihat pada Gambar V.35:



Gambar V.35 *Entity Relationship Diagram Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

5.4.2 *Conseptual Data Model*

Conceptual Data Model digunakan untuk menggambarkan atau merepresentasikan seluruh muatan informasi yang dikandung oleh basis data, berikut ini (Gambar V.36) merupakan *Conceptual Data Model* dari sistem usulan:



Gambar V.36 *Conceptual Data Model Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

5.4.3 Kamus Data

Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analisis sistem mempumyai pengertian yang sama tentang *input*, *output* dan komponen *data store*. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input*, laporan dan basis data. Berikut adalah kamus data sistem informasi *monitoring quality control*

1. Spesifikasi Tabel tb_pengguna

   Nama Tabel : tb_pengguna

   Fungsi : Untuk menyimpan data pengguna

   Tipe : *File* data master

Tabel V.19 Tabel tb_pengguna

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode pengguna	kd_pengguna	*Varchar*	10	*Primary Key*
2	Nama lengkap	nama_lengkap	*Varchar*	20	*Not Null*
3	Alamat	alamat	*Varchar*	30	*Not Null*
4	Jabatan	jabatan	*Varchar*	15	*Not Null*
5	Username	username	*Varchar*	10	*Not Null*
6	Password	password	*Varchar*	10	*Not Null*

Sumber:Hasil Analisis (2019)

2. Spesifikasi Tabel tb_barang

   Nama Tabel : tb_barang

   Fungsi : Untuk menyimpan data barang

   Tipe : *File* data master

Tabel V.20 Tabel tb_barang

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode barang	kd_barang	*Varchar*	10	*Primary Key*
2	Nama barang	nama_barang	*Varchar*	30	*Not Null*
3	Produksi	produksi	*Integer*		*Not Null*
4	Kode Pelanggan	Kd_pelanggan	*Varchar*	10	*Foreign Key*

Sumber:Hasil Analisis (2019)

3. Spesifikasi Tabel tb_pelanggan

   Nama Tabel : tb_pelanggan

   Fungsi : Untuk menyimpan data pelanggan

   Tipe : *File* data master

Tabel V.21 Tabel tb_pelanggan

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode pelanggan	kd_pelanggan	*Varchar*	10	*Primary Key*
2	Nama pelanggan	nama_ pelanggan	*Varchar*	30	*Not Null*
3	Alamat	alamat	*Varchar*	30	*Not Null*

Sumber:Hasil Analisis (2019)

4. Spesifikasi Tabel tb_cs

   Nama Tabel : tb_cs

   Fungsi : Untuk menyimpan data check sheet

   Tipe : *File* data masukan

Tabel V.22 Tabel tb_cs

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode *Check Sheet*	kd_cs	*Varchar*	10	*Primary Key*
2	Tanggal	tanggal_cs	*Date*		*Not Null*
3	Pengukuran visual	visual	*Varchar*	3	*Not Null*
4	Pengukuran numeric	numerik	*Varchar*	3	*Not Null*
5	Catatan	catatan	*Varchar*	10	*Not Null*

Tabel V.22 Tabel tb_cs (lanjutan)

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
6	Status	status_cs	*Varchar*	10	*Not Null*
7	Kode barang	kd_barang	*Varchar*	10	*Foreign Key*

Sumber:Hasil Analisis (2019)

5. Spesifikasi Tabel tb_pc

   Nama Tabel : tb_pc

   Fungsi : Untuk menyimpan data *problem countermeasure*

   Tipe : *File* data masukan

Tabel V.23 Tabel tb_pc

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode *Problem Countermeasure*	kd_pc	*Varchar*	10	*Primary Key*
2	Tanggal	tanggal_pc	*Date*		*Not Null*
3	*Problem*	problem	*Varchar*	10	*Not Null*
4	*Couse of problem*	cop	*Varchar*	15	*Not Null*
5	*Countermeasure*	counter	*Varchar*	20	*Not Null*
6	Status	status_pc	*Varchar*	10	*Not Null*
7	Kode *Check Sheet*	kd_cs	*Varchar*	10	*Foreign Key*

Sumber:Hasil Analisis (2019)

6. Spesifikasi Tabel tb_rc

   Nama Tabel : tb_rc

   Fungsi : Untuk menyimpan data *repair card*

   Tipe : *File* data masukan

Tabel V.24 Tabel tb_rc

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode *Repair Card*	kd_rc	*Varchar*	10	*Primary Key*
2	Tanggal	tanggal_rc	*Date*		*Not Null*
3	QC *Judge*	qc_judge	*Varchar*	20	*Not Null*
4	Proses	proses	*Varchar*	20	*Not Null*
5	Kode *Problem Countermeasure*	kd_pc	*Varchar*	10	*Foreign Key*

Sumber:Hasil Analisis (2019)

7. Spesifikasi Tabel tb_png

   Nama Tabel : tb_png

   Fungsi : Untuk menyimpan data *part not good*

   Tipe : *File* data masukan

Tabel V.25 Tabel tb_png

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode *Part Not Good*	kd_png	*Varchar*	10	*Primary Key*
2	Tanggal	tanggal_png	*Date*		*Not Null*
3	Jumlah produksi	jml_prod	*Integer*		*Not Null*
4	*Not Good*	ng	*Integer*		*Not Null*
5	Kode *Problem Countermeasure*	kd_pc	*Varchar*	10	*Foreign Key*

Sumber:Hasil Analisis (2019)

8. Spesifikasi Tabel tb_mr

   Nama Tabel : tb_mr

   Fungsi : Untuk menyimpan data *monitoring reject*

Tabel V.26 Tabel tb_mr

No	Nama Elemen	Akronim	Tipe	Panjang	Keterangan
1	Kode *Monitoring Reject*	kd_mr	*Varchar*	10	*Primary Key*
2	Tanggal	tanggal_mr	*Date*		*Not Null*
3	Persentase	persentase	*Integer*	5	*Not Null*

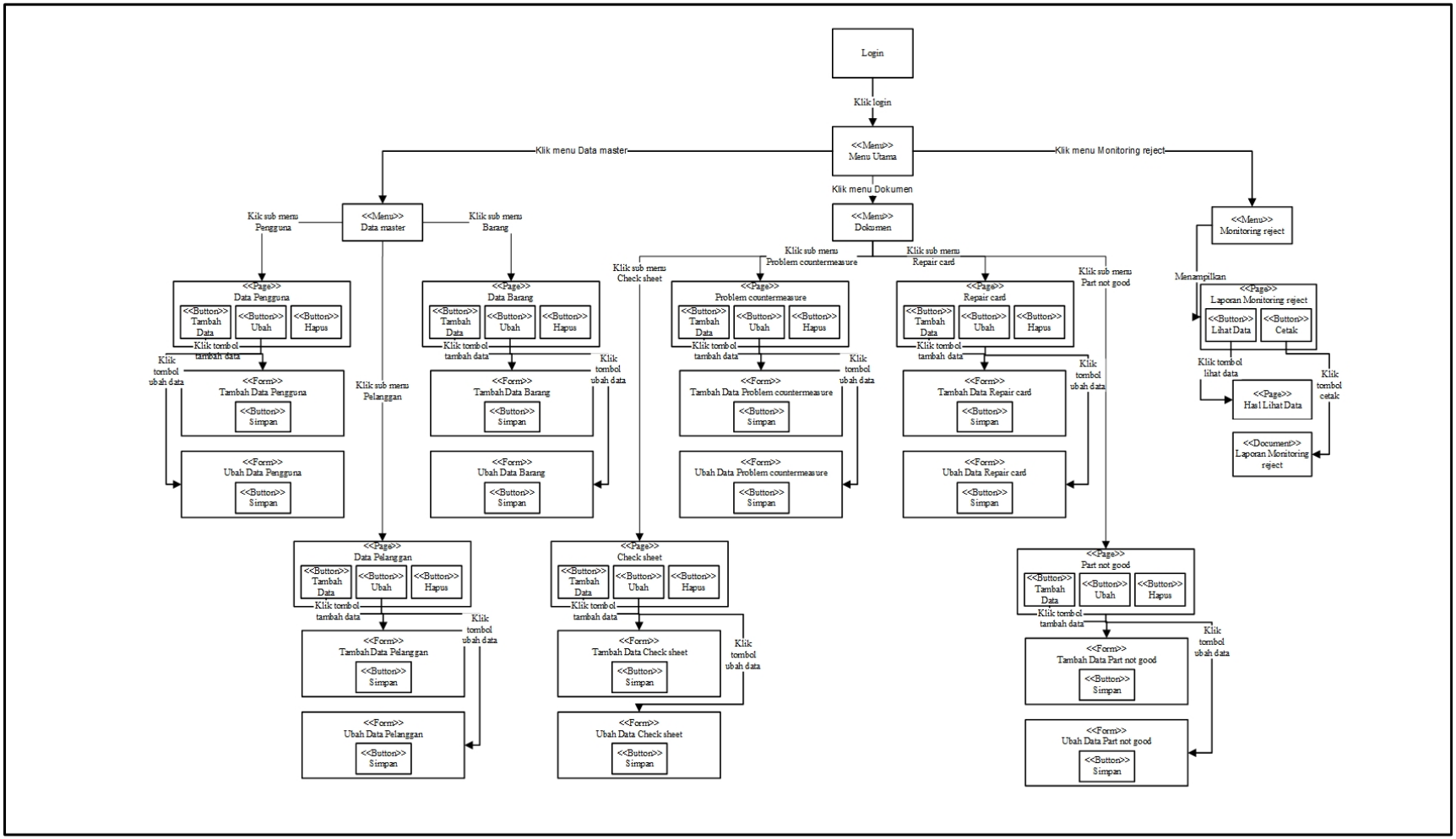
Sumber:Hasil Analisis (2019)

5.5 Perancangan Sistem

Perancangan sistem pada sistem usulan digunakan sebagai ilustrasi yang menggambarkan bagaimana aplikasi yang dibuat berjalan dan menggambarkan navigasi dari aplikasi tersebut dengan *tools* WND serta menggambarkan tampilan antar muka dari aplikasi.

5.5.1 *Windows Navigation Diagram*

Dengan *Windows Navigation Diagram* kita dapat dengan mudah melihat skema sistem, sehingga akan memudahkan menganalisa sistem. Berikut ini merupakan contoh *Windows Navigation Diagram* usulan pada sistem informasi *monitoring quality control* dapat dilihat pada Gambar V.37:



Gambar V.37 *Wimdows Navigation Diagram Monitoring Quality Control*
Sumber:Hasil Analisis (2019)

5.5.2 Perancangan *Interface* Sistem

Perancangan antarmuka dalam sistem usulan merupakan penggambaran antarmuka aplikasi yang hendak dibuat, antarmuka merupakan bagian yang berhubungan langsung dengan pengguna, berikut nerupakan rancangan antarmuka sistem informasi *monitoring quality control:*

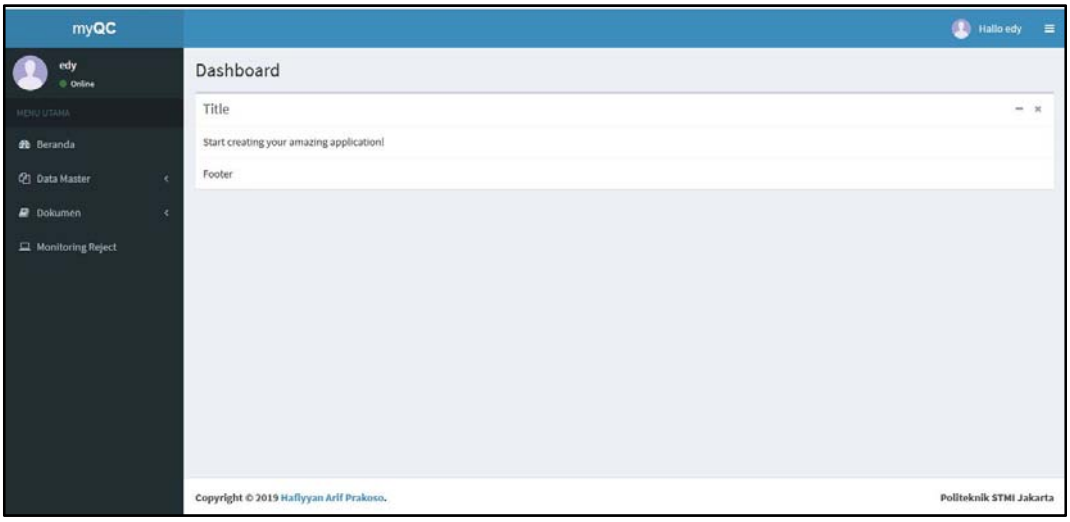
1. *Form Login*

   Form login adalah form yang digunakan untuk masuk ke dalam program aplikasi. Untuk masuk ke dalam aplikasi, pengguna harus memasukkan *username* dan *password* yang benar. Rancangan *interface* dari *form login* dapat dilihat pada Gambar V.38:

Gambar V.38 *Interface Form Login*
Sumber:Hasil Analisis (2019)

2. *Form* Menu Utama

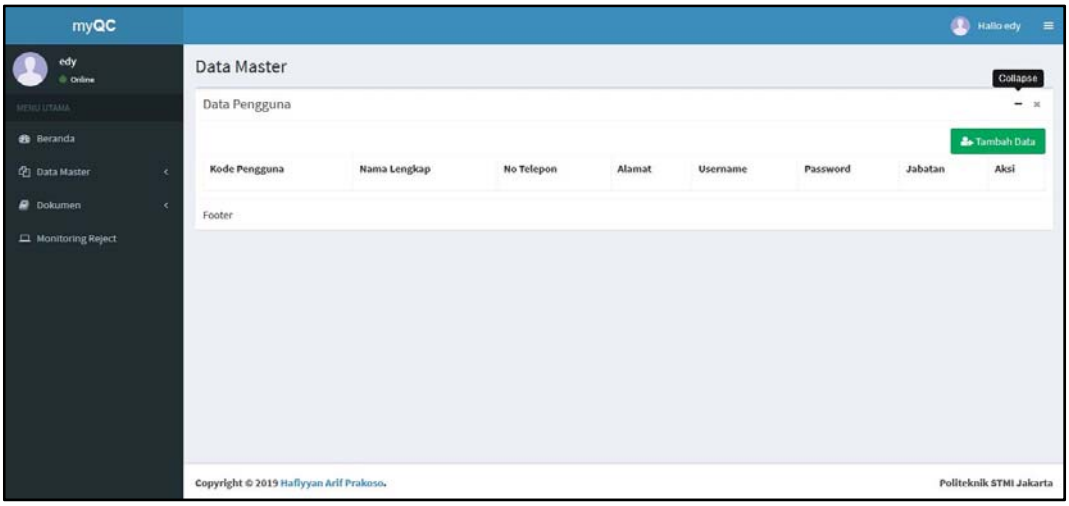
   Form menu utama adalah tampilan awal pada aplikasi yang berisi empat menu. Menu tersebut adalah beranda, data master, dokumen dan *monitoring reject*. Rancangan *interface* dari *form* menu utama dapat dilihat pada Gambar V.39:



Gambar V.39 *Interface Form* Menu Utama
Sumber:Hasil Analisis (2019)

3. *Form* Data Master Pengguna

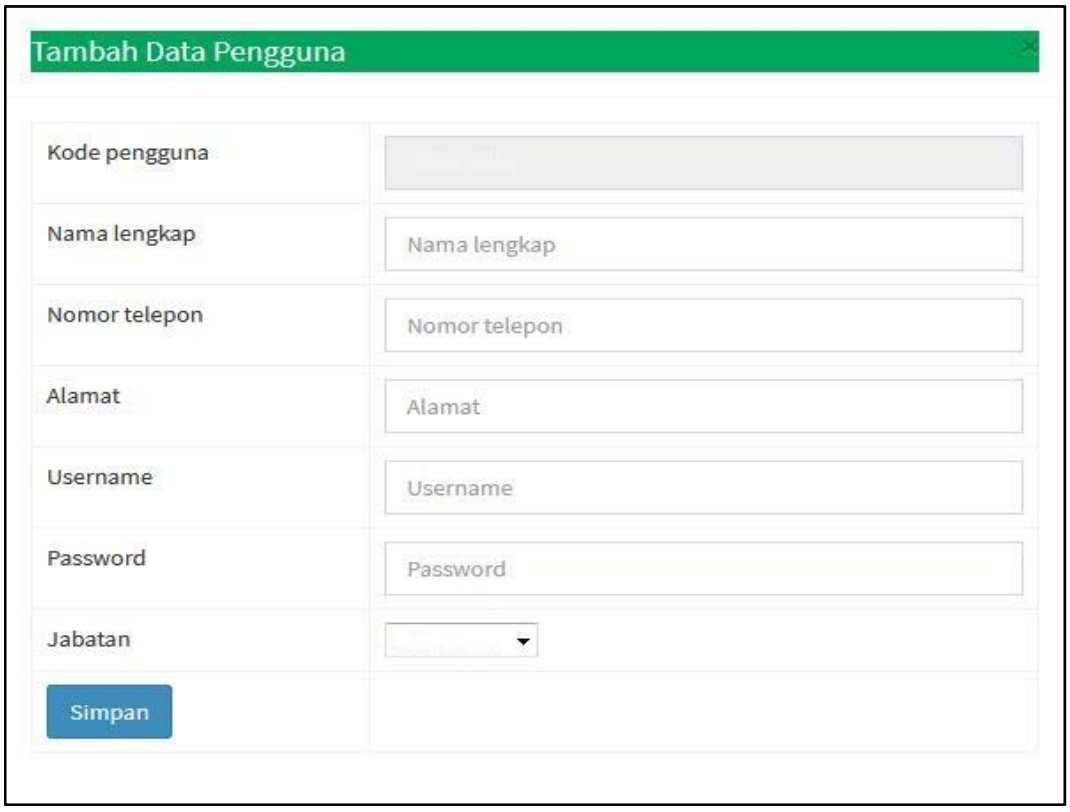
   Form data pengguna adalah form yang digunakan untuk mengelola data pengguna. Rancangan *interface* dari *form* data pengguna dapat dilihat pada Gambar V.40:



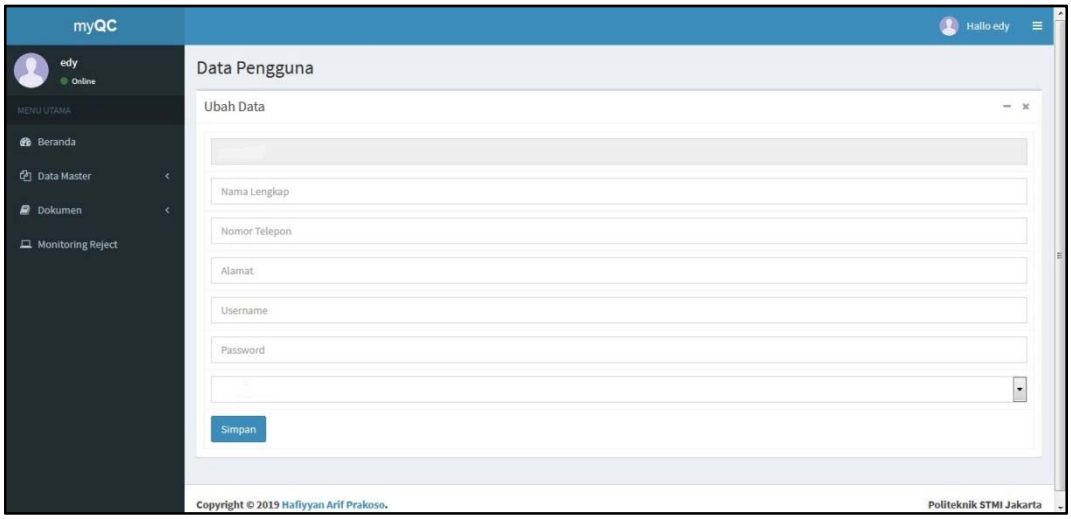
Gambar V.40 *Interface Form* Data Master Pengguna
Sumber:Hasil Analisis (2019)

4. *Form* Tambah dan Ubah Data Pengguna

   Form ini digunakan untuk menambah dan mengubah data pengguna. Rancangan *Form* Tambah dan Ubah Data Pengguna dapat dilihat pada Gambar V.41 dan V.42 berikut:



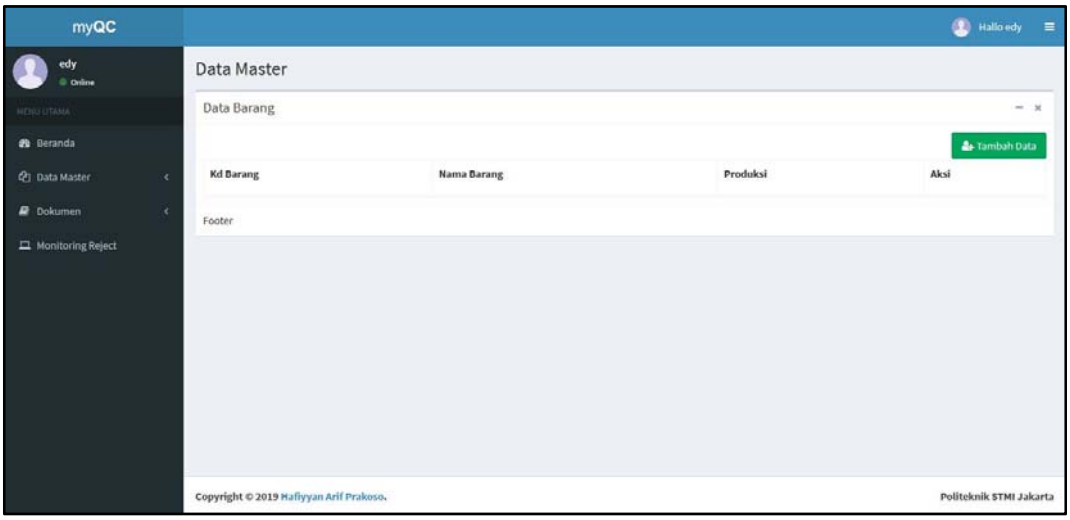
Gambar V.41 *Interface Form* Tambah Data Pengguna
Sumber:Hasil Analisis (2019)



Gambar V.42 *Interface Form* Ubah Data Pengguna
Sumber:Hasil Analisis (2019)

5. *Form* Data Master Barang

   Form data barang adalah form yang digunakan untuk mengelola data barang. Rancangan *interface* dari *form* data barang dapat dilihat pada Gambar V.43:

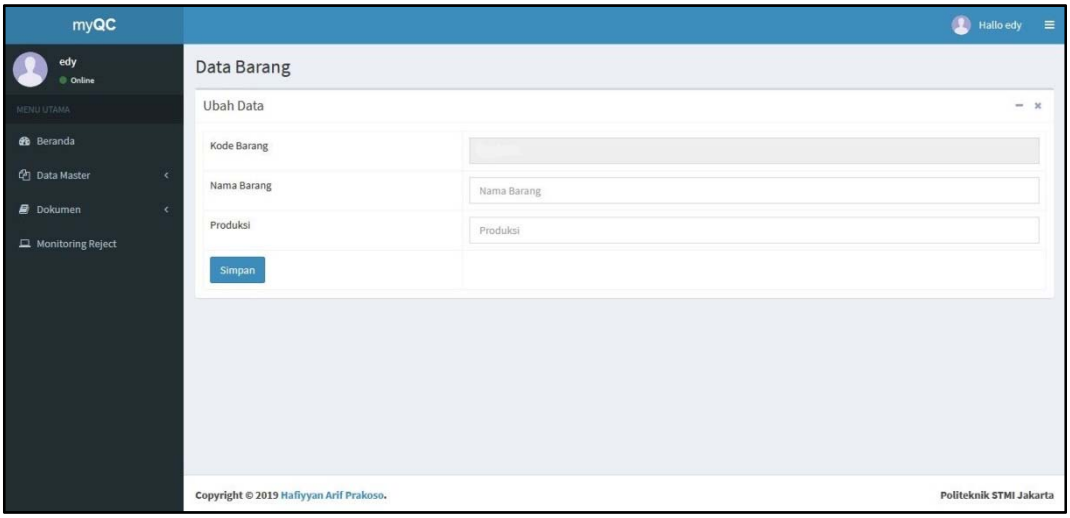


Gambar V.43 *Interface Form* Data Master Barang
Sumber:Hasil Analisis (2019)

6. *Form* Tambah dan Ubah Data Barang

   Form ini digunakan untuk menambah dan mengubah data barang. Rancangan *Form* Tambah dan Ubah Data Barang dapat dilihat pada Gambar V.44 dan V.45 berikut:

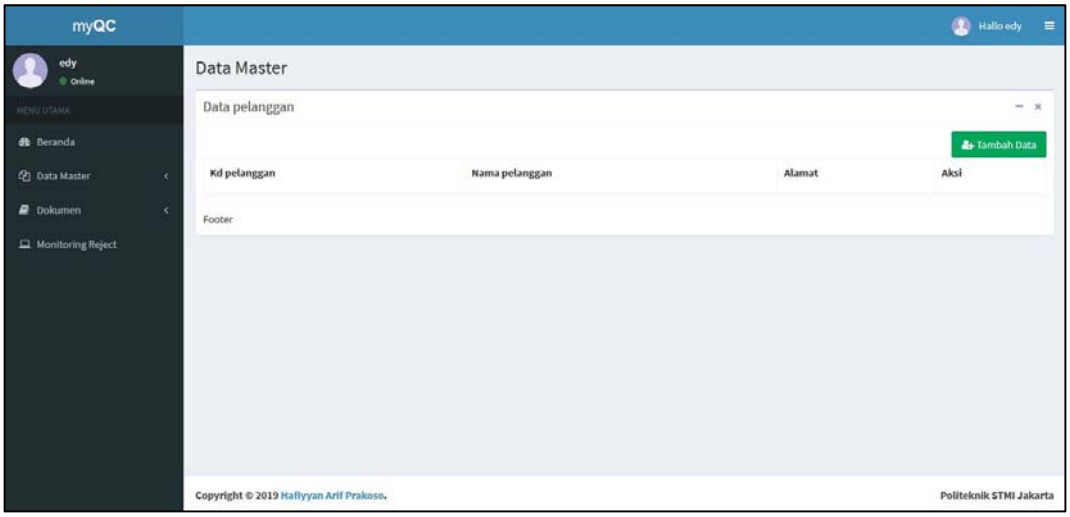
Gambar V.44 *Interface Form* Tambah Data Barang
Sumber:Hasil Analisis (2019)



Gambar V.45 *Interface Form* Ubah Data Barang
Sumber:Hasil Analisis (2019)

7. *Form* Data Master Pelanggan

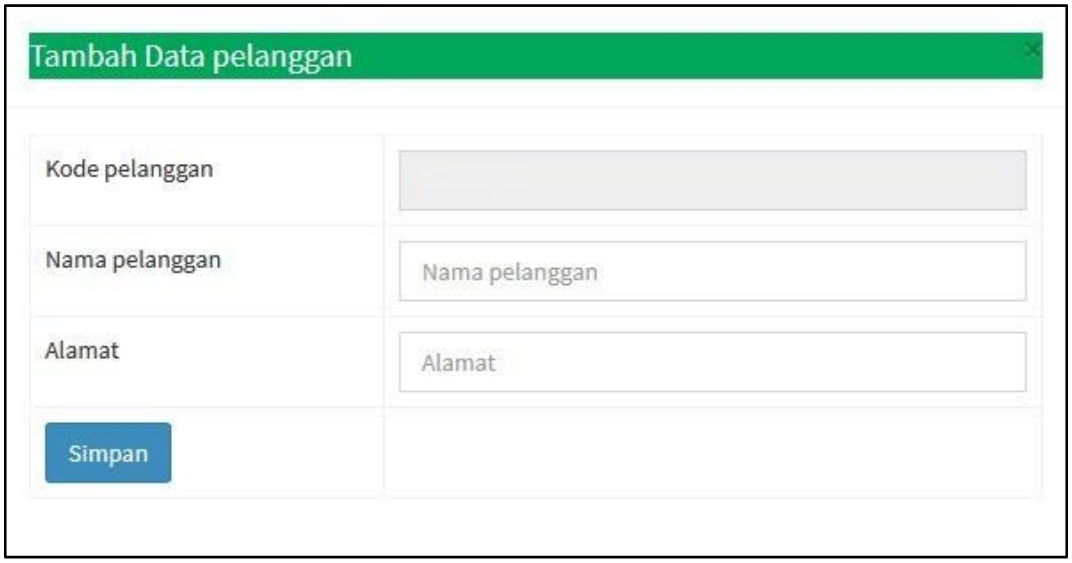
   Form data pelanggan adalah form yang digunakan untuk mengelola data pelanggan. Rancangan *interface* dari *form* data pelanggan dapat dilihat pada Gambar V.46:



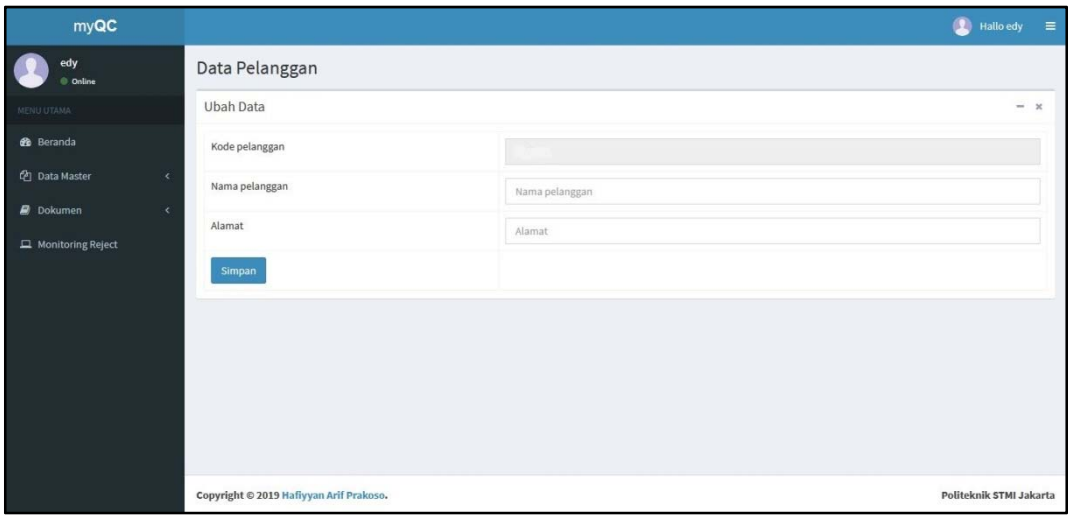
Gambar V.46 *Interface Form* Data Master Pelanggan
Sumber:Hasil Analisis (2019)

8. *Form* Tambah dan Ubah Data Pelanggan

   Form ini digunakan untuk menambah dan mengubah data pelanggan. Rancangan *Form* Tambah dan Ubah Data Pelanggan dapat dilihat pada Gambar V.47 dan V.48 berikut:



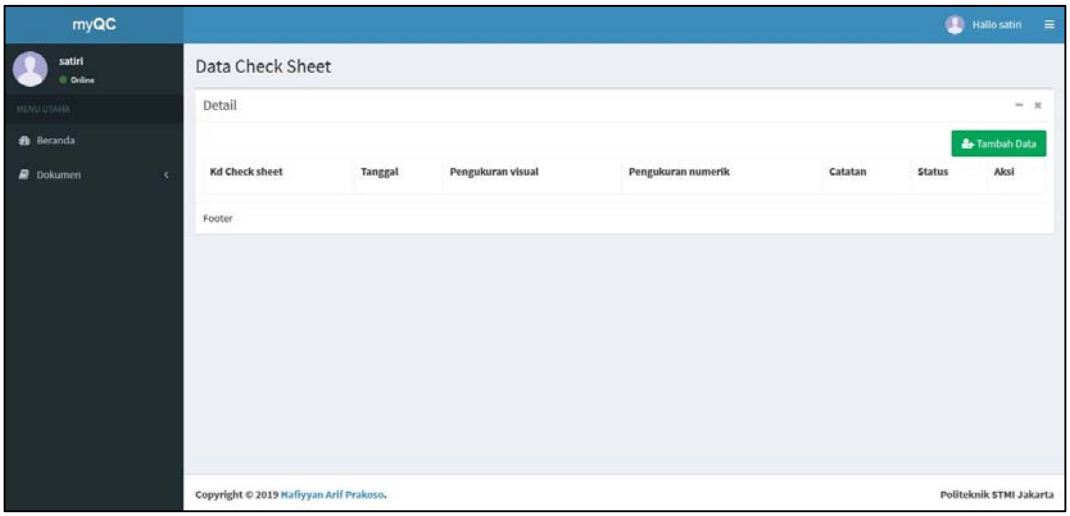
Gambar V.47 *Interface Form* Tambah Data Pelanggan
Sumber:Hasil Analisis (2019)



Gambar V.48 *Interface Form* Ubah Data Barang
Sumber:Hasil Analisis (2019)

9. *Form* Data *Check sheet*

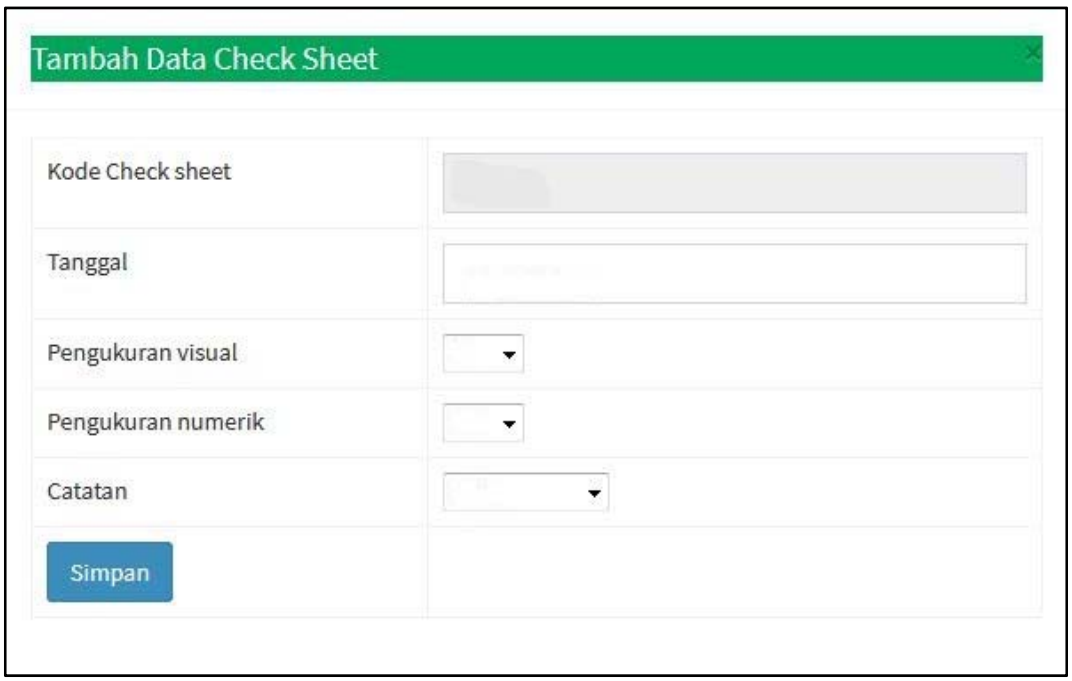
   Form data *check sheet* adalah form yang digunakan untuk mengelola data *check sheet*. Rancangan *interface* dari *form* data *check sheet* dapat dilihat pada Gambar V.49:



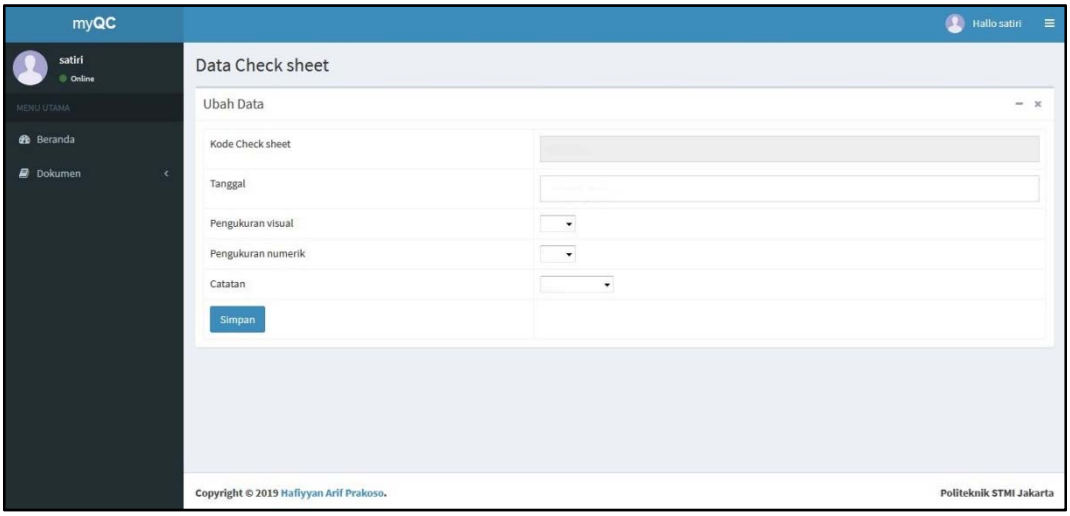
Gambar V.49 *Interface Form* Data *Check sheet*
Sumber:Hasil Analisis (2019)

10. *Form* Tambah dan Ubah Data *Check sheet*

    Form ini digunakan untuk menambah dan mengubah data *check sheet*. Rancangan *Form* Tambah dan Ubah Data *Check sheet* dapat dilihat pada Gambar V.50 dan V.51 berikut:



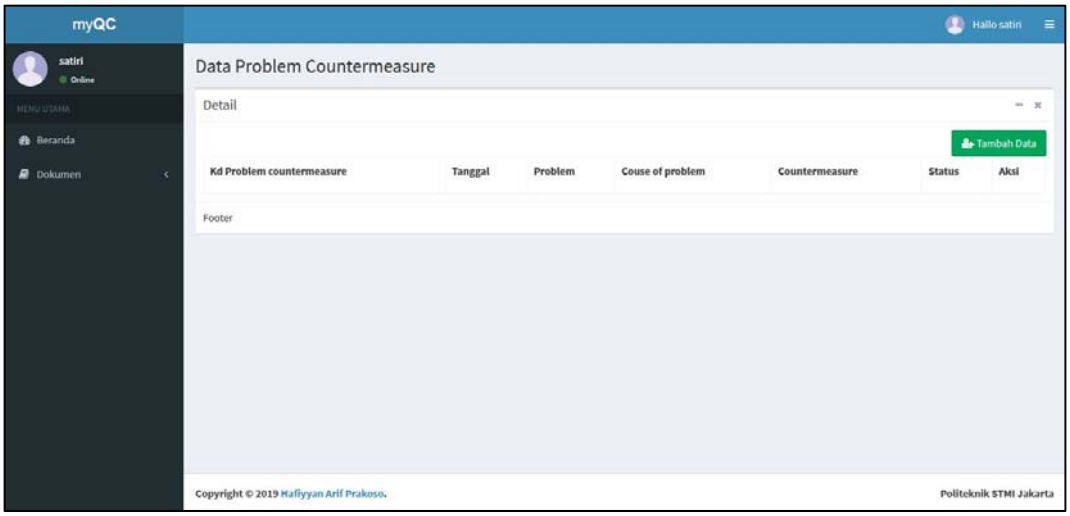
Gambar V.50 *Interface Form* Tambah Data *Check sheet*
Sumber:Hasil Analisis (2019)



Gambar V.51 *Interface Form* Ubah Data *Check sheet*
Sumber:Hasil Analisis (2019)

11. *Form* Data *Problem countermeasure*

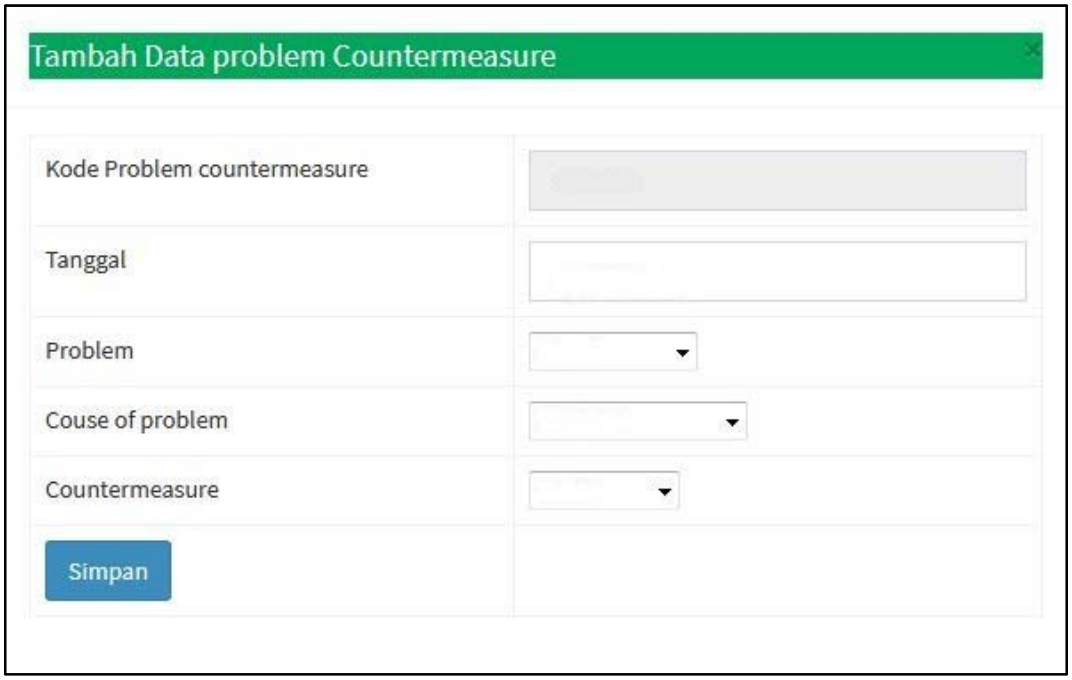
    Form data *problem countermeasure* adalah form yang digunakan untuk mengelola data *problem countermeasure*. Rancangan *interface* dari *form* data *problem countermeasure* dapat dilihat pada Gambar V.52:



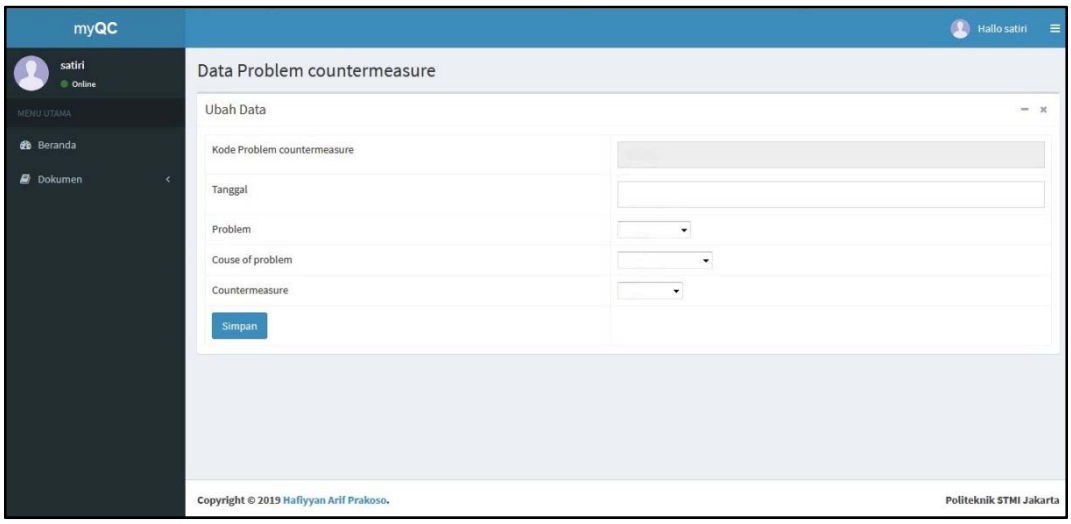
Gambar V.52 *Interface Form* Data *Problem countermeasure*
Sumber:Hasil Analisis (2019)

12. *Form* Tambah dan Ubah Data *Problem countermeasure*

    Form ini digunakan untuk menambah dan mengubah data *problem countermeasure*. Rancangan *Form* Tambah dan Ubah Data *Problem countermeasure* dapat dilihat pada Gambar V.53 dan V.54 berikut:



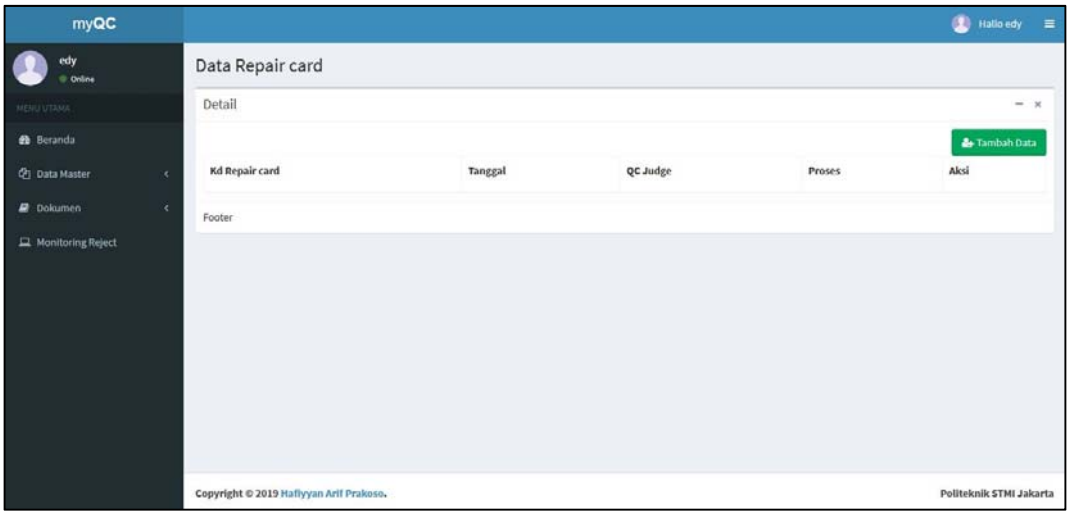
Gambar V.53 *Interface Form* Tambah Data *Problem countermeasure*
Sumber:Hasil Analisis (2019)



Gambar V.54 *Interface Form* Ubah Data *Problem Countermeasure*
Sumber:Hasil Analisis (2019)

13. *Form* Data *Repair card*

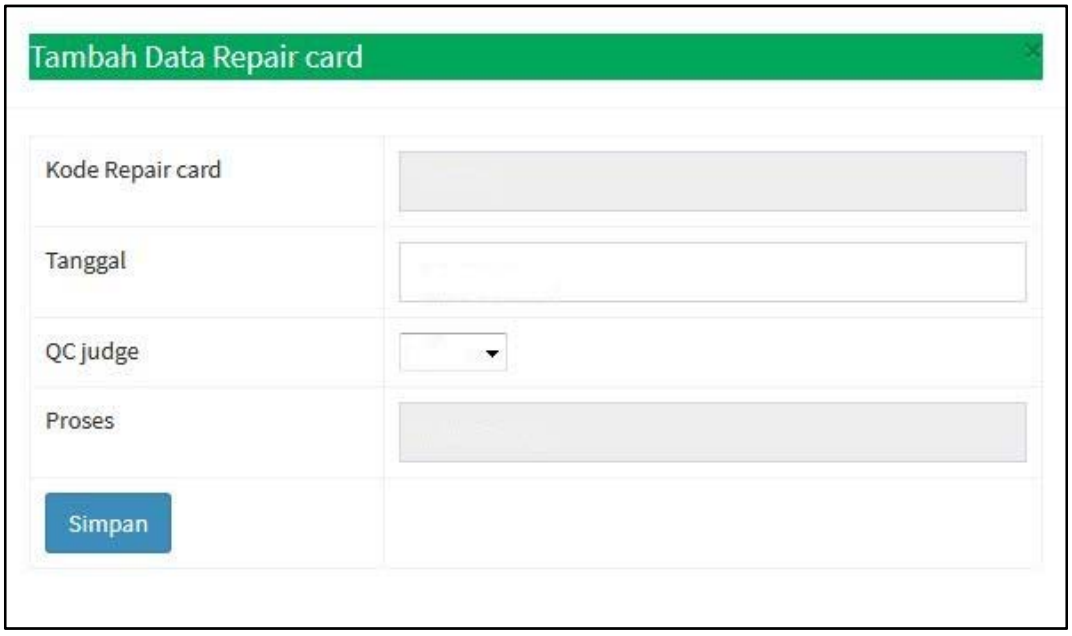
Form data *repair card* adalah form yang digunakan untuk mengelola data *repair card*. Rancangan *interface* dari *form* data *repair card* dapat dilihat pada Gambar V.55:



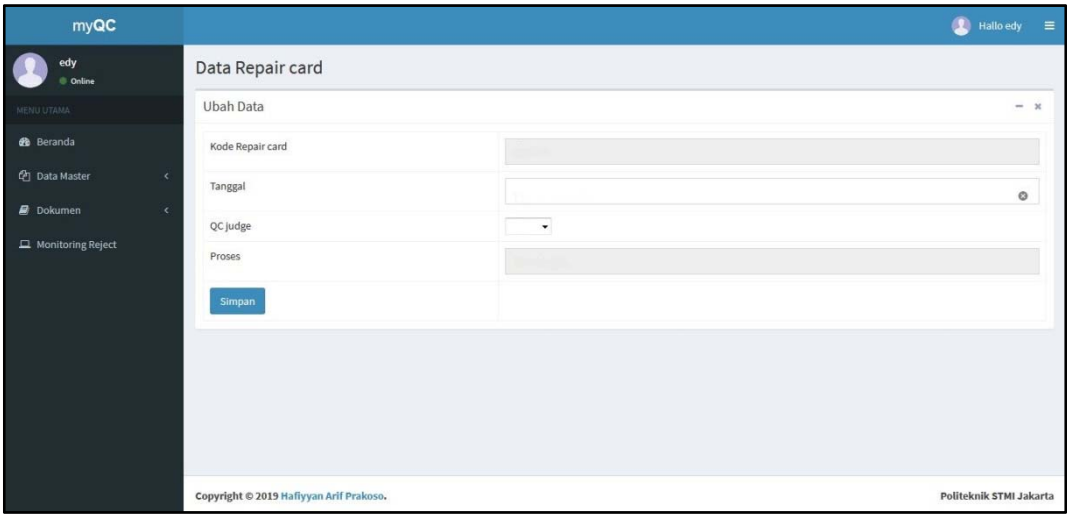
Gambar V.55 *Interface Form* Data *Repair Card*
Sumber:Hasil Analisis (2019)

14. *Form* Tambah dan Ubah Data *Repair card*

    Form ini digunakan untuk menambah dan mengubah data *repair card*. Rancangan *Form* Tambah dan Ubah Data *Repair card* dapat dilihat pada Gambar V.56 dan V.57 berikut:



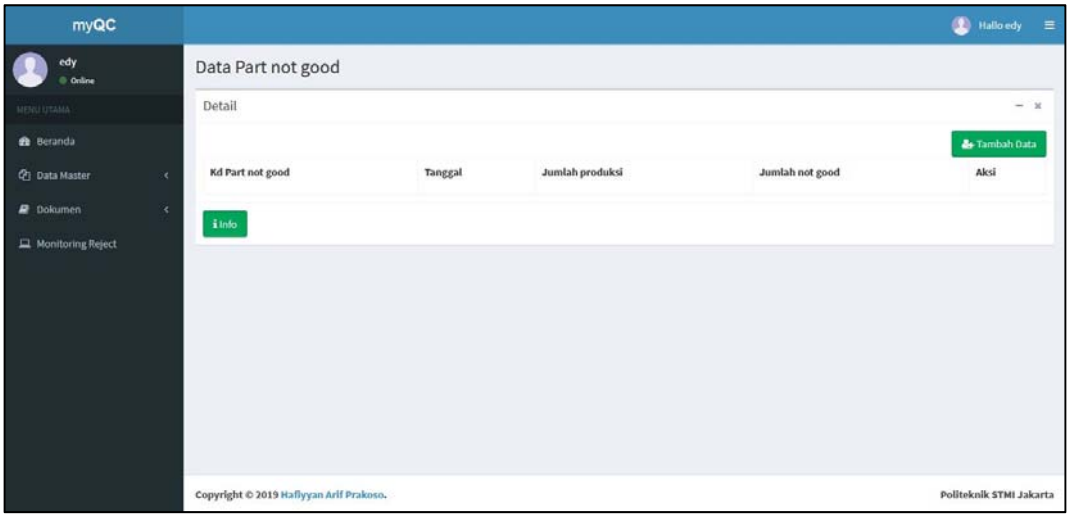
Gambar V.56 *Interface Form* Tambah Data *Repair Card*
Sumber:Hasil Analisis (2019)



Gambar V.57 *Interface Form* Ubah Data *Repair Card*
Sumber:Hasil Analisis (2019)

15. *Form* Data *Part not good*

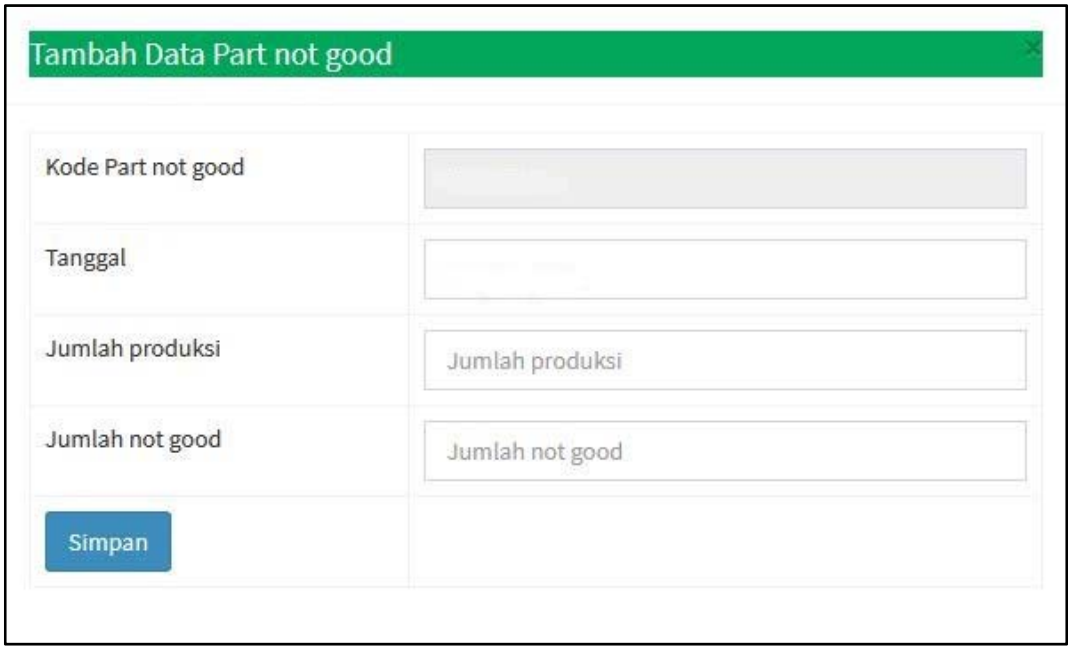
    Form data *part not good* adalah form yang digunakan untuk mengelola data *part not good*. Rancangan *interface* dari *form* data *part not good* dapat dilihat pada Gambar V.58:



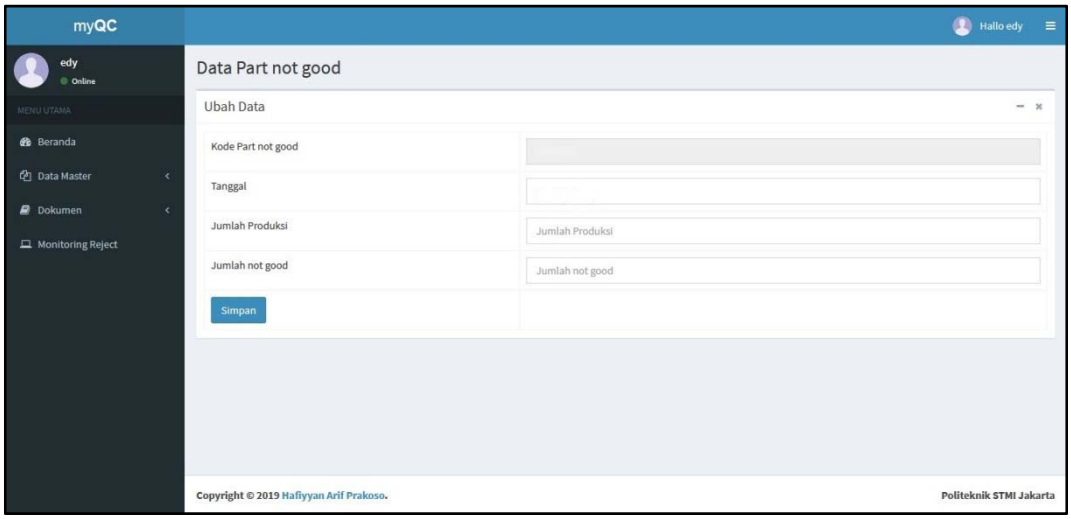
Gambar V.58 *Interface Form* Data *Part Not Good*
Sumber:Hasil Analisis (2019)

16. *Form* Tambah dan Ubah Data *Part not good*

    Form ini digunakan untuk menambah dan mengubah data *part not good*. Rancangan *Form* Tambah dan Ubah Data *Part not good* dapat dilihat pada Gambar V.59 dan V.60 berikut:



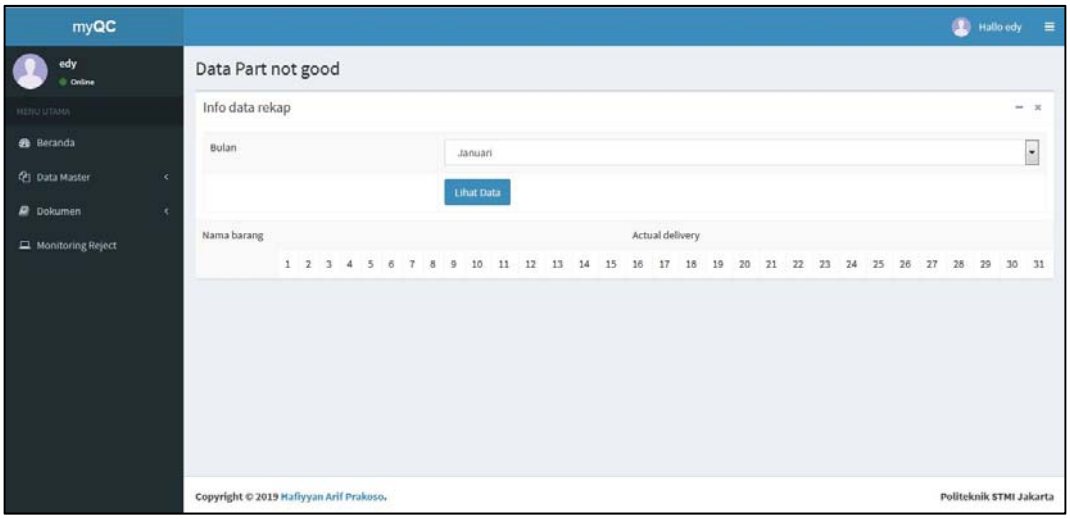
Gambar V.59 *Interface Form* Tambah Data *Part Not Good*
Sumber:Hasil Analisis (2019)



Gambar V.60 *Interface Form* Ubah Data *Part Not Good*
Sumber:Hasil Analisis (2019)

17. *Form* Data Info *Part not good*

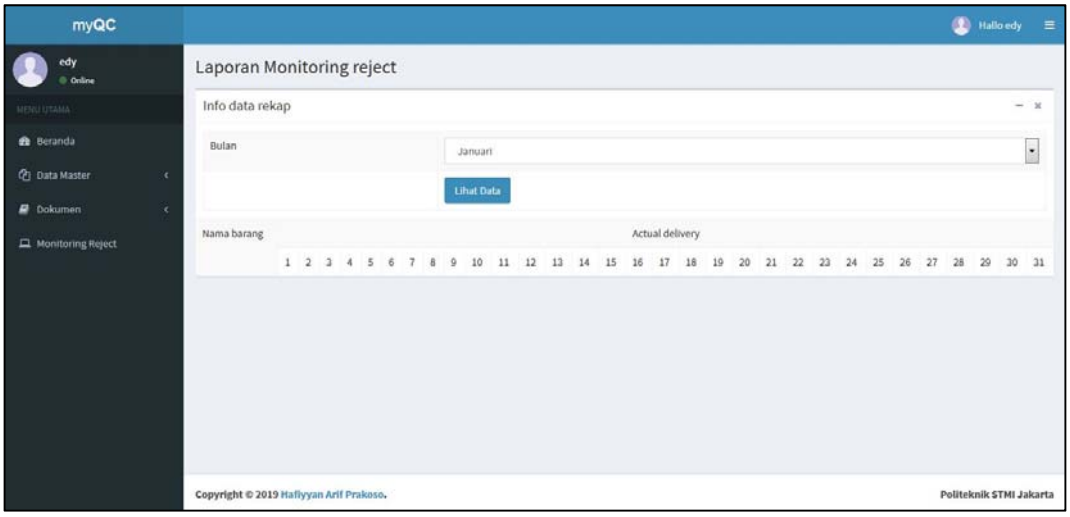
Form data info *part not good* adalah form yang digunakan untuk menlihat data *part not good*. Rancangan *interface* dari *form* datainfo *part not good* dapat dilihat pada Gambar V.61:



Gambar V.61 *Interface Form* Data Info *Part Not Good*
Sumber:Hasil Analisis (2019)

18. *Form Monitoring Reject*

Form data *monitoring reject* adalah form yang digunakan untuk menlihat data *monitoring reject*. Rancangan *interface* dari *form monitoring reject* dapat dilihat pada Gambar V.62:



Gambar V.62 *Interface Form Monitoring Reject*
Sumber:Hasil Analisis (2019)

5.6 Implementasi Sistem

Tahap ini adalah tahap pengkodean program menggunakan bahasa pemrograman PHP dengan menggunakan Visual Studio Code sebagai aplikasi *editor*. Untuk mendukung kebutuhan implementasi sistem diperlukan suatu spesifikasi perangkat lunak (*software*) maupun perangkat keras (*hardware*). Adapun spesifikasiknya antara lain sebagai berikut:

1. Analisis Kebutuhan *Software*
   a. Sistem Operasi : Microsoft Windows 7
   b. *Database Server* : PostgreSQL 10.8-4
   c. Bahasa Pemrograman : PHP 5.6.12
   d. *Web Browser* : Google Chrome
2. Analisis Kebutuhan *Hardware*
   a. *Processor* : Minimal *Processor* Intel Core 2 Duo
   b. RAM : Minimal RAM 2 GB
   c. *Harddisk* : Minimal *Harddisk* 64 GB
   d. *Mouse*, *Keyboard*, *Monitor* sebagai peralatan antarmuka.

5.7 Pengujian Sistem

Metode pengujian yang digunakan dalam penelitian adalah *black box testing*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Uji coba dengan *black box* pada sistem ini bertujuan untuk menentukan fungsi cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan. Pengujian sistem informasi *monitoring quality control* dapat dilihat pada lampiran.

BAB VI
KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan pengumpulan, pengolahan, dan analisis data yang telah dilakukan pada bab-bab sebelumnya pada penelitian ini maka dapat disimpulkan beberapa poin kesimpulan sebagai berikut:

7. Dengan bantuan sistem informasi *monitoring quality control* yang terkomputerisasi dapat menghasilkan data akurat sehingga mempercepat proses pengendalian kualitas dalam penyebaran informasi dan dapat mengurangi tingkat kesalahan.
8. Dengan bantuan sistem informasi *monitoring quality control* yang dibangun dengan basis data dapat membantu mengelola laporan *monitoring reject* untuk mempercepat penghitungan jumlah kualitas produk.

6.2 Saran

Saran yang diberikan untuk pengembangan sistem informasi *monitoring quality control* selanjutnya adalah sebagai berikut:

1. Aplikasi sistem informasi *monitoring quality control* diharapkan dapat di implementasikan pada divisi *Quality Control* pada PT Nusa Indah Jaya Utama.
2. Untuk pengembangan lebih lanjut dapat dikembangkan dengan menambahkan proses pengendalian kualitas bahan baku pada PT Nusa Indah Jaya Utama yang dapat melakukan pengecekan lebih detail dan akurat sehingga lebih meminimalisir tingkat kesalahan.

DAFTAR PUSTAKA

Achyari, A. (2013). *Manajemen Operasi*. Tanggerang: Universitas Terbuka.

Adil, A. (2017). *Sistem Informasi Geografis.* Yogyakarta: Andi.

Amsler. (2009). Performance monitoring: guidance for the modern workplace. Supervision.

Anhar. (2010). *PHP & MySQL Secara Otodidak.* Jakarta: PT Trans Media.

Arikunto, S. (2010). *Prosedur Penelitian : Suatu Pendekatan Praktik.* Jakarta : Rineka Cipta.

Bakhtiar, S. (2013). Analisa Pengendalian Kualitas dengan Menggunakan Metode Statistical Quality Control (SQC). Malikussaleh Industrial Engineering Journal.

Besterfield, D.H. (2009). *Quality Control – Fifth Edition*. Prentice Hall International.

Darmadi, H. (2013). *Metode Penelitian Pendidikan dan Sosial.* Bandung : Alfabeta.

Dennis, Alan. (2010). *System Analysis and Design with UML Version 2.0*. New York: John Wiley and Sons, Inc.

Dennis, Alan. (2012). *System Analysis and Design with UML Version 2.0, An. Object-Oriented Approach 4th Edition*. New York: John Wiley and Sons, Inc.

Dennis, Alan. (2015). *Systems Analysis & Design An Object-Oriented Approach with UML*. Danvers: John Wiley & Sons, Inc.

Heizer, Jay. dan Render, Barry. (2013). *Operations Management (Manajemen Operasi)*. Jakarta: Salemba Empat.

Hoffer, Jeffrey. A. (2007). *Modern Database Management*. New Jersey: Pearson Prentice Hall.

Jogiyanto. (2010). *Analisis & Desain Sistem Informasi*. Yogyakarta: Andi.

Kartika, Hayu. (2013). *Analisis Pengendalian Kualitas Produk CPE Film Dengan Metode Statistical Process Control*.

Kristanto, A. (2008). *Perancangan Sistem Informasi dan Aplikasinya*. Yogyakarta: Gava Media.

Mulyani, S. (2016). *Metode Analisis dan Perancangan Sistem.* Bandung: Abdi Sistematika.

Mulyono, dan Yumari. (2017). Strategi Monitoring dan Evaluasi Pelaksanaan Anggaran. Yogyakarta: Deepublish.

Nasir, dan Zakiyah. (2013). Panduan Penerapan Sistem Informasi Desa (SID) dan Monitoring Partisipatif. Yogyakarta: UNDP Indonesia.

Pratama, I, P. (2014). *Sistem Informasi dan Implementasinya.* Bandung: Informatika.

Pressman, R. S. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku.* Yogyakarta : Andi.

Puspitawati, Lilis., dan Anggadini, Sri Dewi. (2014). Sistem Informasi Akuntansi. Yogyakarta: Graha Ilmu.

Richardson, Theodor. dan Thies, Charles. (2013). *Secure Software Design.*

Rosa A.S dan Shalahuddin,M. (2018). *Rekayasa Perangkat Lunak:n Terstruktur dan Berorientasi Objek* (Revisi ed.). Jakarta: Informatika.

Rusdiana, H. A., dan Irfan, Moch. 2014. *Sistem Informasi Manajemen*. Bandung: Pustaka Setia.

Satzinger. (2010). *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.

Sugiyono. (2013). *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R&D.* Bandung : Alfabeta.

Susanto, A. (2013). *Sistem Informasi Akuntansi*. Bandung: Lingga Jaya.

Sutabri, T. (2012). *Analisis Sistem Informasi*. Yogyakarta: Andi .

Tjiptono, Fandy., dan Diana, Anastasia. (2008). *TQM (total quality management).* Yogyakarta: Andi offset.

Widodo, W. (2013). *Membangun Web Super epat dengan CodeIgniter Grocery CRUD dan Tank Auth*.

Wrihatlono, R. (2006). *Manajemen Pembangunan Indonesia*. Jakarta: Elex Media Komputindo.

Yamit, Zulian. (2013). *Manajemen Kualitas Produk dan Jasa*. Yogyakarta: Ekonisia.

LAMPIRAN A
KODE PROGRAM

1. *Controller* Data Barang

```
<?php

class Barang extends CI_Controller
{
	function __construct()
	{
		parent::__construct();
		$this->load->model(['model_barang','model_pelanggan']);
	}

	function index()
	{
		$data['record'] = $this->model_barang->lihat_data();
		$data['pelanggan'] = $this->model_pelanggan-
>lihat_data()->result();
		$data['kode'] = $this->model_barang->kode();
		$this->template->content-
>view('barang/data_barang',$data);
		$this->template->publish();
	}

	function post()
	{
		if (isset($_POST['submit'])){
			$kd_barang = $this->input-
>post('kd_barang');
			$nama_barang = $this->input-
>post('nama_barang');
			$produksi = $this->input-
>post('produksi');
			$kd_pelanggan = $this->input-
>post('kd_pelanggan');
			$data =
	array('kd_barang'=>$kd_barang,
```

```
        'nama_barang'=>$nama_barang,

        'produksi'=>$produksi,

        'kd_pelanggan'=>$kd_pelanggan);
                        $this->model_barang->post($data);
                        redirect('barang');
                }
                else {
                        $data['pelanggan']     =     $this->model_pelanggan-
>lihat_data()->result();
                        $this->template->content-
>view('barang/form_input',$data);
                        $this->template->publish();
                }
        }

        function edit()
        {
                if (isset($_POST['submit'])){
                        $kd_barang                =     $this->input-
>post('kd_barang');
                        $nama_barang        =     $this->input-
>post('nama_barang');
                        $produksi                =     $this->input-
>post('produksi');
                        $kd_pelanggan        =     $this->input-
>post('kd_pelanggan');
                        $data                        =
        array('kd_barang'=>$kd_barang,

        'nama_barang'=>$nama_barang,

        'produksi'=>$produksi,

        'kd_pelanggan'=>$kd_pelanggan);
                        $this->model_barang->edit($data,$kd_barang);
                        redirect('barang');
                }
```

```
                else {
                        $id=    $this->uri->segment(3);
                        $this->load->model('model_barang');
                        $data['barang']=        $this->model_barang-
>lihat_data()->result();
                        $data['pelanggan']    =    $this->model_pelanggan-
>lihat_data()->result();
                        $data['record']=        $this->model_barang-
>get_one($id)->row_array();
                        $this->template->content-
>view('barang/form_edit',$data);
                        $this->template->publish();
                }
        }

        function delete()
        {
                $id=    $this->uri->segment(3);
                $this->model_barang->delete($id);
                redirect('barang');
        }
}
```

2. *Model* Data Barang

```
<?php

class model_barang extends CI_Model
{
   function kode()
   {
      $q   =   $this->db->query("select   MAX(RIGHT(kd_barang,2))   as
code_max from tb_barang");
      $code = "";
      if($q->num_rows()>0){
         foreach($q->result() as $cd){
            $tmp = ((int)$cd->code_max)+1;
            $code = sprintf("%03s", $tmp);
         }
      }else{
         $code = "01";
```

```
        }
        return "BRG".$code;
    }

    function lihat_data()
        {
            return $this->db->get('tb_barang');
        }

    function post ($data)
    {
            $this->db->insert('tb_barang',$data);
        }

    function edit($data, $kd_barang)
    {
            $this->db->where('kd_barang',$kd_barang);
            $this->db->update('tb_barang',$data);
        }

    function get_one($id)
    {
            $param= array('kd_barang'=>$id);
            return $this->db->get_where('tb_barang',$param);
        }

    function delete($id)
    {
            $this->db->where('kd_barang',$id);
            $this->db->delete('tb_barang');
        }
}
```

3. *Views* Data Barang

```
<!-- Content Header (Page header) -->
<section class="content-header">
 <h1>
 Data Master
 </h1>
</section>

<!-- Main content -->
<section class="content">
    <!-- Default box -->
    <div class="box">
     <div class="box-header with-border">
      <h3 class="box-title">Data Barang</h3>

      <div class="box-tools pull-right">
       <button type="button" class="btn btn-box-tool" data-widget="collapse" data-toggle="tooltip"
            title="Collapse">
        <i class="fa fa-minus"></i></button>
       <button type="button" class="btn btn-box-tool" data-widget="remove" data-toggle="tooltip" title="Remove">
        <i class="fa fa-times"></i></button>
      </div>
     </div>

            <div class="box-body">
                <div class="pull-right">
             <button class="btn btn-success" type="button" data-target="#ModalAdd" data-toggle="modal">
                <i class="fa fa-user-plus"></i> Tambah Data
                </button>
          </div>
            <table class="table table-bordered table-hover table-striped cell-border" id="example1">
                <thead>
                <tr>
                    <th> <p> Kd Barang</th> </p>
                    <th> <p> Nama Barang</th> </p>
                    <th> <p> Produksi</th> </p>
```

```
                    <th> <p> Kd Pelanggan</th> </p>
                    <th > <p> Aksi</th> </p>
                </tr>
                </thead>
                <tr>
                <?php
                foreach ($record->result() as $d)
                {
                    echo "<tr>
                    <td>$d->kd_barang</td>
                    <td>$d->nama_barang</td>
                    <td>$d->produksi</td>
                    <td>$d->kd_pelanggan</td>
                    <td>".anchor('barang/edit/'.$d->kd_barang,'Ubah', array('class'=> 'btn btn-primary'))." "
                    .anchor('barang/delete/'.$d->kd_barang,'Hapus',array('class'=> 'btn btn-danger'))." </td>
                    </tr>";
                }
                ?>
            </table>
            </div>

<!-- ini modal -->
        <div id="ModalAdd" class="modal fade" tabindex="-1" role="dialog">
            <div class="modal-dialog">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                        <h4 class="modal-title bg-green">Tambah Data Barang</h4>
                    </div>
                <div class="modal-body">

                <?php
                echo form_open('barang/post', "name='modal_popup'");
```

```
                        ?>
<table class="table table-bordered">
        <tr>
                <td>Kode barang</td>
                <td>    <input type="text" class="form-control" name="kd_barang" placeholder="Kd barang" value="<?php echo $kode; ?>"
                readonly/></td>
        </tr>
        <tr>
                <td>Nama barang</td>
                <td>    <input type="text" class="form-control" name="nama_barang" placeholder="Nama barang" required autofocus></td>
        </tr>
        <tr>
                <td>Produksi</td>
                <td>    <input type="number" class="form-control" name="produksi" placeholder="Produksi" required></td>
        </tr>
        <tr>
                <td>Nama Pelanggan</td>
                <td>
                <select class="" name="kd_pelanggan" placeholder="kd_pelanggan">
                        <<?php
          foreach ($pelanggan as $k) {
             echo "<option value= '$k->kd_pelanggan'>$k->nama_pelanggan </option>";
          }
          ?>
                </select>
                </td>
        </tr>
        <tr>
                <td> <button type="submit" class ="btn btn-primary" name="submit">Simpan</button></td>
        </tr>
</table>
</form>
```

```
</div>
</div>
</div>
</div>
<div class="box-footer">
          Footer
        </div>
      </div>
</section>
```

LAMPIRAN B

BLACK BOX TESTING

1. *Form Login*

   Deskripsi : Menguji fungsi login pada *Form Login*

   Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol *Login* tanpa memasukkan *username* dan *password*	Sistem akan menampilkan pesan isi isian ini.	Sistem menampilkan pesan isi isian ini.	Valid
2	Memasukkan *username* dengan benar sedangkan *pasword* salah atau sebaliknya, lalu mengklik tombol *Login*	Sistem akan menampilkan pesan kombinasi *username* dan *password* salah	Sistem menampilkan pesan kombinasi *username* dan *password* salah	Valid
3	Memasukkan *username* dan *password* dengan benar lalu mengklik tombol *Login*	Sistem akan menerima akses *login* dan menampilkan halaman utama	Sistem menerima akses *login* dan menampilkan halaman utama	Valid

2. Menu Data Pengguna

   Deskripsi : Menguji fungsi tambah data pada Menu Data Pengguna

   Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Tambah Data	Sistem akan menampilkan *form* Tambah Data Pengguna	Sistem menampilkan *form* Tambah Data Pengguna	Valid

   Deskripsi : Menguji fungsi ubah pada Menu Data Pengguna

   Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Ubah	Sistem akan menampilkan *form* Ubah Data Pengguna	Sistem menampilkan *form* Ubah Data Pengguna	Valid

Deskripsi : Menguji fungsi hapus pada Menu Data Pengguna
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Hapus pada data yang dipilih	Sistem akan menghapus data yang dipilih	Sistem menghapus data yang dipilih	Valid

3. *Form* Tambah Data Pengguna
Deskripsi : Menguji fungsi simpan pada *Form* Tambah Data Pengguna
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Simpan dengan mengosongkan semua kolom	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
2	Mengklik tombol Simpan dengan kondisi salah satu kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
3	Mengisi seluruh kolom isian lalu mengklik tombol Simpan	Data dapat disimpan	Data dapat disimpan	Valid

4. *Form* Ubah Data Pengguna
Deskripsi : Menguji fungsi ubah pada *Form* Ubah Data Pengguna
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Ubah dengan kondisi semua kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
2	Mengklik tombol Ubah dengan kondisi salah satu kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
3	Mengisi seluruh kolom isian lalu mengklik tombol Ubah	Data dapat diubah	Data dapat diubah	Valid

5. Menu Data Barang

Deskripsi : Menguji fungsi tambah data pada Menu Data Barang
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Tambah Data	Sistem akan menampilkan *form* Tambah Data Barang	Sistem menampilkan *form* Tambah Data Barang	Valid

Deskripsi : Menguji fungsi ubah pada Menu Data Barang
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Ubah	Sistem akan menampilkan *form* Ubah Data Barang	Sistem menampilkan *form* Ubah Data Barang	Valid

Deskripsi : Menguji fungsi hapus pada Menu Data Barang
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Hapus pada data yang dipilih	Sistem akan menghapus data yang dipilih	Sistem menghapus data yang dipilih	Valid

6. *Form* Tambah Data Barang

Deskripsi : Menguji fungsi simpan pada *Form* Tambah Data Barang
Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Simpan dengan mengosongkan semua kolom	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
2	Mengklik tombol Simpan dengan kondisi salah satu kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
3	Mengisi seluruh kolom isian lalu mengklik tombol Simpan	Data dapat disimpan	Data dapat disimpan	Valid

7. *Form* Ubah Data Barang

Deskripsi : Menguji fungsi ubah pada *Form* Ubah Data Barang

Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Ubah dengan kondisi semua kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
2	Mengklik tombol Ubah dengan kondisi salah satu kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
3	Mengisi seluruh kolom isian lalu mengklik tombol Ubah	Data dapat diubah	Data dapat diubah	Valid

8. Menu Data Pelanggan

Deskripsi : Menguji fungsi tambah data pada Menu Data Pelanggan

Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Tambah Data	Sistem akan menampilkan *form* Tambah Data Pelanggan	Sistem menampilkan *form* Tambah Data Pelanggan	Valid

Deskripsi : Menguji fungsi ubah pada Menu Data Pelanggan

Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Ubah	Sistem akan menampilkan *form* Ubah Data Pelanggan	Sistem menampilkan *form* Ubah Data Pelanggan	Valid

Deskripsi : Menguji fungsi hapus pada Menu Data Pelanggan

Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Hapus pada data yang dipilih	Sistem akan menghapus data yang dipilih	Sistem menghapus data yang dipilih	Valid

9. *Form* Tambah Data Pelanggan

Deskripsi : Menguji fungsi simpan pada *Form* Tambah Data Pelanggan

Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Simpan dengan mengosongkan semua kolom	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
2	Mengklik tombol Simpan dengan kondisi salah satu kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
3	Mengisi seluruh kolom isian lalu mengklik tombol Simpan	Data dapat disimpan	Data dapat disimpan	Valid

10. *Form* Ubah Data Pelanggan

Deskripsi : Menguji fungsi ubah pada *Form* Ubah Data Pelanggan

Penguji : Herly Frisdiansyah (1315032)

No.	*Test Case*	*Expected Result*	*Actual Record*	*Result*
1	Mengklik tombol Ubah dengan kondisi semua kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
2	Mengklik tombol Ubah dengan kondisi salah satu kolom tidak diisi	Sistem akan menampilkan pesan isi isian ini.	Sistem akan menampilkan pesan isi isian ini.	Valid
3	Mengisi seluruh kolom isian lalu mengklik tombol Ubah	Data dapat diubah	Data dapat diubah	Valid