

SKRIPSI

ANALISA DAN PERANCANGAN SISTEM INFORMASI MONITORING HASIL INTERNAL AUDIT BERBASIS WEB PADA PT. ABC



Disusun Oleh :

NAMA : NURAMANAH

NPM : 2015804296

JURUSAN : SISTEM INFORMASI

**Untuk Memenuhi Sebagian Dari Syarat-syarat
Guna Mencapai Gelar Sarjana Komputer**

STMIK INSAN PEMBANGUNAN

Jl. Raya Serang Km. 10 Bitung-Tangerang

Website: <http://www.stmik.ipem.ac.id>

Email: info@ipem.ac.id

Telp. (021) 59492836

Fax. (021) 59492837

Th. Akademik 2019/2020

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
INSAN PEMBANGUNAN
SARJANA KOMPUTER JURUSAN SISTEM INFORMASI
2019/2020**

TANDA PERSETUJUAN SKRIPSI

**NAMA : NURAMANAH
NPM : 2015804296
JURUSAN : SISTEM INFORMASI
JUDUL SKRIPSI : ANALISA DAN PERANCANGAN SISTEM
INFORMASI MONITORING HASIL
INTERNAL AUDIT BERBASIS WEB PADA
PT. ABC**

SKRIPSI INI TELAH DIPERIKSA DAN DISETUJUI :

STMIK INSAN PEMBANGUNAN

Pembimbing Materi

Pembimbing Teknis

Jainuri, S.Kom., M.Kom
NIDN : 0401018505

Nurasiah, S.Kom., MMSI
NIDN : 0418077801

Ketua Jurusan Sistem Informasi

Nurasiah, S.Kom., MMSI
NIDN : 0418077801



**SEKOLAH TINGGI MANAJEMEN INFORMATIKA
DAN KOMPUTER INSAN PEMBANGUNAN**
Jl. Raya Serang KM. 10 Bitung – Tangerang
Tahun Akademik 2019/2020

TANDA PERSETUJUAN SETELAH LULUS UJIAN SKRIPSI

NAMA : NURAMANAH
NPM : 2015804296
JURUSAN : SISTEM INFORMASI
JUDUL SKRIPSI : ANALISA DAN PERANCANGAN SISTEN
INFORMASI MONITORING HASIL INTERNAL
AUDIT BERBASIS WEB PADA PT. ABC

Tangerang, 16 Februari 2020

Penguji I

Penguji II

Winanti, S.Kom., MM., M.Kom.
M.Kom

NIDN : 0405057702

Mustar Aman, S.Kom.,

NIDN : 0421088002

Mengetahui
Ketua STMIK Insan Pembangunan

Dr. Drs. Karnawi Kamar, MM.

NIDN: 0304086803

BERITA ACARA KOMPREHENSIF / SKRIPSI

Berdasarkan Surat Keputusan Ketua STMIK Insan Pembangunan No. STMIK 176/K-I/STMIK-IP/II/2020 tanggal 16 Februari 2020, pada hari ini telah dilaksanakan Ujian Komprehensif / Skripsi program S1 STMIK Insan Pembangunan Jurusan Sistem Informasi untuk Tahun Akademik 2019/2020

- 
1. Nama : NURAMANA H
2. NPM : 2015804296
3. Program Studi : Strata Satu
4. Jurusan : Sistem Informasi
5. Judul Skripsi : Analisa dan Perancangan Sisten Informasi
Monitoring Hasil Internal Audit Berbasis WEB pada
PT. ABC
6. Ruang / Tempat : 203 / Kampus STMIK Insan Pembangunan
7. Kelulusan dengan nilai : B / 77,40
8. Keterangan : Lulus

PANITIA UJIAN

- | | | | |
|---|----------------------------------|-------|------------|
| 1 | Dr. Dra. Francisca Sestri G., MM | _____ | Ketua |
| 2 | Winanti, S.Kom., MM., M.Kom | _____ | Sekretaris |
| 3 | Nurasiah, S.Kom., MMSI | _____ | Anggota |
| 4 | Supriyanto, SE., MM | _____ | Anggota |

Mengetahui
Ketua STMIK Insan Pembangunan

Dr. Drs. Karnawi Kamar, MM.
NIDN: 0304086803

PERSEMBAHAN

Alhamdulillah, puji syukur kehadiran Allah SWT, karena berkat pertolongan-Nya dan izin-Nya skripsi ini dapat terselesaikan. Dengan penuh rasa bangga saya persembahkan skripsi ini kepada:

1. Kedua orang tuaku tercinta yang selalu memberikan kasih sayang, selalu mendoakan untuk keberhasilanku serta dukungan sehingga semuanya berjalan dengan lancar.
2. Kakak dan adikku yang selalu menyemangati, membantu, dan mendoakan dalam penyusunan skripsi ini.
3. Untuk kawan lamaku (Diana Zuhriyatul Fauziyah, S.Pd) yang telah membantu, mendo'akan dan memberi semangat selama menyusun skripsi ini.
4. Untuk sahabat-sahabat seperjuanganku (*The Game Bell*) yang selalu bersama dalam menuntut ilmu di STMIK Insan Pembangunan.
5. Kepada Bpk. M. Ikhsan terima kasih telah mengizinkan penulis melakukan penelitian serta rekan-rekan kerja yang telah membantu dan membimbing selama melakukan penelitian sehingga skripsi ini dapat terselesaikan.
6. Teman-teman Sistem Informasi Kelas D angkatan 2015 yang selalu mewarnai kehidupanku sehari-hari selama proses perkuliahan
7. Almamaterku STMIK Insan Pembangunan tercinta yang telah membimbing, mendidik dan mendewasakanku dalam berpikir dan bertindak.

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan segala rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Analisa dan Perancangan Sistem Informasi Internal Audit pada PT. ABC”.

Adapun maksud dari penyusunan skripsi ini adalah untuk memenuhi syarat guna menyelesaikan Program Studi Strata Satu (S1) pada STMIK Insan Pembangunan. Dalam menyusun skripsi ini, banyak pihak yang telah membantu dalam penyusunan skripsi ini. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dalam penulisan dan penggunaan tata bahasa Indonesia yang digunakan, untuk itu kritik dan saran yang sifatnya membangun dari berbagai pihak sangat diharapkan dalam rangka penyempurnaan penulisan skripsi ini.

Dalam menyelesaikan skripsi ini, penulis banyak menerima bantuan dan bimbingan yang sangat berharga dari berbagai pihak. Untuk itu penulis mengucapkan terimakasih kepada :

1. Bapak H. Soebari Hadi Prayitno, selaku Ketua Yayasan Pendidikan Insan Pembangunan.
2. Bapak Dr. Drs. Karnawi Kamar, M.M. selaku Ketua STMIK Insan Pembangunan.
3. Ibu Dr. Dra Francisca Sesri, G., MM. selaku Pembantu Ketua 1 Bidang Akademik STMIK Insan Pembangunan.

4. Ibu Nurasiah, S.Kom., MMSI. selaku Ketua Jurusan Sistem Informasi STMIK Insan Pembangunan sekaligus dosen pembimbing teknis
5. Bapak Jainuri, S.Kom., M.Kom selaku dosen pembimbing materi yang selalu memberi masukan-masukan dalam proses penyusunan skripsi ini.
6. Seluruh Dosen STMIK Insan Pembangunan.
7. Kedua Orang Tua, yang selalu mengiringi dengan doa yang sangat mulia dalam hidup saya.
8. Kakanda dan Adinda yang selalu memberikan support.
9. Dan rekan-rekan maupun pihak-pihak yang tidak bisa saya sebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini. Untuk itu penulis memerlukan kritik dan saran untuk perbaikan dikemudian hari. Penulis berharap skripsi ini dapat memberikan manfaat untuk kita semua khususnya bagi penulis.

Tangerang, 16 Februari 2020

NURAMANA H

SURAT PERNYATAAN

Yang bertanda tangan dibawah ini :

NAMA : NUR AMANAH

NPM : 2015804296

JURUSAN : SISTEM INFORMASI

JUDUL SKRIPSI : ANALISA DAN PERANCANGAN SISTEM

INFORMASI *MONITORING* HASIL *INTERNAL*

AUDIT BERBASIS *WEB* PADA PT. ABC

Dengan ini menerangkan bahwa Skripsi dengan judul tersebut di atas adalah benar hasil karya tulis dan penelitian saya. Oleh karena itu saya bersedia untuk mempertanggungjawabkannya. Apabila dikemudian hari ternyata Skripsi tersebut bukan hasil karya tulis dan penelitian saya maka saya siap menerima sanksi dari kampus.

Demikian surat pernyataan ini saya buat untuk persyaratan Sidang Skripsi.

Terima kasih.

Tangerang, 16 Februari 2020

Yang membuat pernyataan,

(NURAMANAH)

ABSTRAK

PT. ABC merupakan salah satu industri makanan. Salah satu permasalahan yang terjadi pada proses internal audit adalah *monitoring* hasil *audit* yang kurang efektif dan efisien, karena *internal audit* yang terjadi saat ini masih manual yaitu hasil *report* dan *monitoring audit* masih menggunakan *form* yang diolah menggunakan *Microsoft Excel* lalu mengirimkan hasil *audit* tersebut melalui *email*. Seiring meningkatnya perkembangan teknologi informasi, saat ini perusahaan membutuhkan *monitoring internal audit* yang efektif dan efisien. Dengan permasalahan tersebut akan dirancang Sistem Informasi *Monitoring Hasil Internal Audit* Berbasis Web. Metode penelitian yang digunakan berupa metode observasi, wawancara, *Study Literatur*, analisa sistem dan perancangan sistem. Sistem informasi ini dibuat dengan menggunakan bahasa pemrograman *PHP*, *Framework Laravel*, *DBMS MySQL*, metode *waterfall*. Salah satu manfaat dan keuntungan dalam penggunaan aplikasi *monitoring* hasil *internal audit* ini dapat melakukan *monitoring* hasil *internal audit* yang lebih efektif, efisien dan akurat.

Kata Kunci : *Internal Audit, PHP, Framework Laravel, MySQL, Waterfall*



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN DOSEN PEMBIMBING	ii
HALAM PERSETUJUAN TEAM PENGUJI SKRIPSI	iii
BERITA ACARA UJIAN KOMPREHENSIF	iv
PERSEMBAHAN	v
KATA PENGANTAR	vi
SURAT PERNYATAAN	viii
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xv

BAB I PENDAHULUAN

1.1. Latar Belakang Masalah	1
1.2. Identifikasi Masalah	3
1.3. Ruang Lingkup / Batasn Masalah	4
1.4. Perumusan Masalah	5
1.5. Tujuan Penelitian.....	5
1.6. Manfaat Penelitian.....	5

BABA II LANDASAN TEORI

2.1. Tinjauan Pustaka	7
2.1.1. Pengertian Dasar Sistem	7
2.1.2. Konsep Dasar Informasi	9
2.1.3. Konsep Dasar Sistem Informasi	12
2.1.4. Sistem Informasi Manajemen	13
2.1.5. Pengertian Analisa dan Perancangan	14
2.1.6. Pengertian Monitoring	18
2.1.7. Konsep Dasar <i>Internal Audit</i>	20
2.1.8. Pengertian SDLC (<i>System Devolepment Life Cycle</i>)	21

2.1.9. Model SDLC (<i>System Devolepment Life Cycle</i>)	24
2.2. Penelitian Sebelumnya	34
2.3. Tools yang digunakan	37
2.3.1. UML (<i>Unified Modelling Language</i>)	37
2.3.2. XAMPP	46
2.3.3. MySQL	47
2.3.4. <i>Sublime Text</i>	47
2.3.5. <i>Bootstrap</i>	47
2.3.6. <i>Testing</i>	49
2.4. Kerangka Kerja Teoritis	54

BAB III METODOLOGI PENELITIAN

3.1. Desain Penelitian	58
3.2. Data dan Sumber Data	59
3.3. Metode Pengumpulan Data	60
3.3.1. Wawancara	60
3.3.2. Observasi	61
3.3.3. Studi Pustaka	61
3.4. Metode Analisis Data	62
3.5. Metode Perancangan/Pengembangan Sistem	63
3.5.1. Tahap Analisis Kebutuhan Sistem	65
3.5.2. Desain Diagram yang digunakan	66
3.5.3. <i>Software</i> yang digunakan	68
3.5.4. Metode Pengujian Sistem yang digunakan	68

BAB IV ANALISA DAN PERANCANGAN SISTEM

4.1. Analisa Sistem Berjalan	70
4.1.1. Latar Belakang Perusahaan	70
4.1.2. Visi dan Misi Perusahaan	71
4.1.3. Struktur Organisasi PT. ABC	72

4.1.4. Tata Laksana Sistem yang Berjalan pada Audit Mutu Internal (AMI) dan <i>Audit Good Manufacturing Practices</i> (GMP).....	79
4.1.5. Tata Laksana Sistem yang Berjalan pada Audit Morning Round	85
4.1.6. Masalah yang dihadapi	90
4.1.7. Alternatif Pemecah Masalah	91
4.2. Rancangan Sistem Usulan	92
4.2.1. <i>Use Case</i> Usulan	92
4.2.2. <i>Activity Diagram</i>	116
4.2.3. <i>Sequence Diagram</i>	148
4.2.4. <i>Class Diagram</i>	165
4.2.5. Perancangan Basis Data	165
4.2.6. Rancangan Tampilan Program yang diusulkan	172
4.3. Spesifikasi <i>Hardware</i> dan <i>Software</i>	185
4.3.1. Kebutuhan <i>Hardware</i>	185
4.3.2. Kebutuhan <i>Software</i>	186
4.4. Jadwal Pengembangan Sistem	186

BAB V Penutup

5.1. Kesimpulan	189
5.2. Saran	190

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

TABEL 2.1. Penelitian Sebelumnya	36
TABEL 2.2. Simbol <i>Use Case</i>	38
TABEL 2.3. Simbol <i>Activity Diagram</i>	41
TABEL 2.4. Simbol <i>Sequence Diagram</i>	42
TABEL 2.5. Simbol <i>Class Diagram</i>	45
TABEL 2.6. Jenis-jenis <i>Multiplicity</i>	46
TABEL 4.1. Deskripsi Aktor Berjalan pada <i>Audit Mutu Internal</i> (AMI) dan <i>Audit GMP</i>	80
TABEL 4.2. Deskripsi <i>Use Case</i> Berjalan pada <i>Audit Mutu Internal</i> (AMI) dan <i>Audit GMP</i>	80
TABEL 4.3. Deskripsi Aktor Sistem Berjalan pada <i>Audit Morning Round</i>	86
TABEL 4.4. Deskripsi <i>Use Case</i> Sistem Berjalan pada <i>Audit Morning Round</i> ..	86
TABEL 4.5. Deskripsi Aktor Sistem Usulan	93
TABEL 4.6. Deskripsi <i>Use Case</i> Sistem Usulan	93
TABEL 4.7. <i>Scenario Use Case Login</i>	94
TABEL 4.8. <i>Scenario Use Case</i> Kelola Data User	95
TABEL 4.9. <i>Scenario Use Case</i> Kelola Data Departemen	96
TABEL 4.10. <i>Scenario Use Case</i> Kelola Data Area	98
TABEL 4.11. <i>Scenario Use Case</i> Kelola Data Klasifikasi	99
TABEL 4.12. <i>Scenario Use Case</i> Kelola Data Klausul	101
TABEL 4.13. <i>Scenario Use Case</i> Periode Audit AMI	102
TABEL 4.14. <i>Scenario Use Case</i> Periode Audit GMP	103
TABEL 4.15. <i>Scenario Use Case</i> Periode Audit Morning Round	104
TABEL 4.16. <i>Scenario Use Case</i> Audit Mutu Internal (AMI)	106
TABEL 4.17. <i>Scenario Use Case</i> Audit (GMP)	107
TABEL 4.18. <i>Scenario Use Case</i> Audit Morning Round	108
TABEL 4.19. <i>Scenario Use Case</i> CAR AMI	110
TABEL 4.20. <i>Scenario Use Case</i> CAR GMP	111
TABEL 4.21. <i>Scenario Use Case</i> CAR Morning Round	112

TABEL 4.22. <i>Scenario Use Case Approve CAR AMI</i>	113
TABEL 4.23. <i>Scenario Use Case Approve CAR GMP</i>	114
TABEL 4.24. <i>Scenario Use Case Approve CAR Morning Round</i>	114
TABEL 4.25. <i>Scenario Use Case Laporan AMI</i>	115
TABEL 4.26. <i>Scenario Use Case Laporan GMP</i>	115
TABEL 4.27. <i>Scenario Use Case Laporan Morning Round</i>	116
TABEL 4.28. Spesifikasi Tabel <i>Users</i>	165
TABEL 4.29. Spesifikasi Tabel <i>Tb_departemens</i>	166
TABEL 4.30. Spesifikasi Tabel <i>Tb_areas</i>	166
TABEL 4.31. Spesifikasi Tabel <i>Tb_klausuls</i>	167
TABEL 4.32. Spesifikasi Tabel <i>Tb_klasifikasi</i>	167
TABEL 4.33. Spesifikasi Tabel <i>Tb_periode_amis</i>	167
TABEL 4.34. Spesifikasi Tabel <i>Tb_periode_gmp</i>	168
TABEL 4.35. Spesifikasi Tabel <i>Tb_periode_mrs</i>	168
TABEL 4.36. Spesifikasi Tabel <i>Tb_audit_amis</i>	169
TABEL 4.37. Spesifikasi Tabel <i>Tb_audit_gmps</i>	169
TABEL 4.38. Spesifikasi Tabel <i>Tb_audit_mrs</i>	170
TABEL 4.39. Spesifikasi Tabel <i>Tb_car_amis</i>	170
TABEL 4.40. Spesifikasi Tabel <i>Tb_car_gmps</i>	171
TABEL 4.41 Spesifikasi Tabel <i>TB_car_mrs</i>	171
TABEL 4.42. Jadwal Pengembangan Sistem	185

DAFTAR GAMBAR

GAMBAR 2.1. Ilustrasi Model <i>Waterfall</i>	24
GAMBAR 2.2. Ilustrasi Model Prototipe	27
GAMBAR 2.3. Ilustrasi Model RAD	29
GAMBAR 2.4. Ilustrasi Model Iteratif	32
GAMBAR 2.5. Kerangka Kerja Teoritis	54
GAMBAR 4.1. Struktur Organisasi PT. ABC	72
GAMBAR 4.2. <i>Use Case Diagram</i> Sistem Berjalan pada <i>Audit Mutu Internal</i> (AMI) dan <i>Audit GMP</i>	79
GAMBAR 4.3. <i>Activity Diagram</i> Sistem Berjalan pada <i>Audit Mutu Internal</i> dan <i>Audit GMP</i>	82
GAMBAR 4.4. <i>Use Case Diagram</i> Sistem Berjalan pada <i>Audit Morning Round</i>	85
GAMBAR 4.5. <i>Activity Diagram</i> Berjalan pada <i>Audit Morning Round</i>	88
GAMBAR 4.6. <i>Use Case Diagram</i> Usulan	92
GAMBAR 4.7. <i>Activity Diagram</i> Login	116
GAMBAR 4.8. <i>Activity Diagram</i> Kelola Data User	117
GAMBAR 4.9. <i>Activity Diagram</i> Kelola Data Departemen	119
GAMBAR 4.10. <i>Activity Diagram</i> Kelola Data Area	121
GAMBAR 4.11. <i>Activity Diagram</i> Kelola Data Klasifikasi	123
GAMBAR 4.12. <i>Activity Diagram</i> Kelola Data Klausul	125
GAMBAR 4.13. <i>Activity Diagram</i> Periode AMI	127
GAMBAR 4.14. <i>Activity Diagram</i> Periode GMP	129
GAMBAR 4.15. <i>Activity Diagram</i> Periode Morning Round	131
GAMBAR 4.16. <i>Activity Diagram</i> Audit Mutu Internal (AMI)	133
GAMBAR 4.17. <i>Activity Diagram</i> Audit GMP	135
GAMBAR 4.18. <i>Activity Diagram</i> Audit Morning Round	137
GAMBAR 4.19. <i>Activity Diagram</i> Membuat CAR AMI	138
GAMBAR 4.20. <i>Activity Diagram</i> Membuat CAR GMP	140
GAMBAR 4.21. <i>Activity Diagram</i> Membuat CAR Morning Round	141

GAMBAR 4.22. <i>Activity Diagram Approve CAR AMI</i>	143
GAMBAR 4.23. <i>Activity Diagram Approve CRA GMP</i>	144
GAMBAR 4.24. <i>Activity Diagram Approve Morning Round</i>	145
GAMBAR 4.25. <i>Activity Diagram Laporan AMI</i>	146
GAMBAR 4.26. <i>Activity Diagram Laporan GMP</i>	147
GAMBAR 4.27. <i>Activity Diagram Laporan Morning Round</i>	148
GAMBAR 4.28. <i>Sequence Diagram Login</i>	149
GAMBAR 4.29. <i>Sequence Diagram Kelola Data User</i>	150
GAMBAR 4.30. <i>Sequence Diagram Kelola Data Departemen</i>	151
GAMBAR 4.31. <i>Sequence Diagram Kelola Data Area</i>	151
GAMBAR 4.32. <i>Sequence Diagram Kelola Data Klasifikasi</i>	152
GAMBAR 4.33. <i>Sequence Diagram Kelola Data Klausul</i>	153
GAMBAR 4.34. <i>Sequence Diagram Periode AMI</i>	153
GAMBAR 4.35. <i>Sequence Diagram Periode GMP</i>	154
GAMBAR 4.36. <i>Sequence Diagram Periode Morning Round</i>	155
GAMBAR 4.37. <i>Sequence Diagram AMI</i>	156
GAMBAR 4.38. <i>Sequence Diagram Audit GMP</i>	156
GAMBAR 4.39. <i>Sequence Diagram Audit Morning Round</i>	157
GAMBAR 4.40. <i>Sequence Diagram Membuat CAR AMI</i>	158
GAMBAR 4.41. <i>Sequence Diagram Membuat CAR GMP</i>	159
GAMBAR 4.42. <i>Sequence Diagram Membuat CAR Morning Round</i>	159
GAMBAR 4.43. <i>Sequence Diagram Approve CAR AMI</i>	160
GAMBAR 4.44. <i>Sequence Diagram Approve CAR GMP</i>	160
GAMBAR 4.45. <i>Sequence Diagram Approve CAR Morning Round</i>	161
GAMBAR 4.46. <i>Sequence Diagram Laporan AMI</i>	161
GAMBAR 4.47. <i>Sequence Diagram Laporan GMP</i>	162
GAMBAR 4.48. <i>Sequence Diagram Laporan Morning Round</i>	162
GAMBAR 4.49. <i>Class Diagram</i>	165
GAMBAR 4.50. <i>Tampilan Form Login</i>	172
GAMBAR 4.51. <i>Tampilan Menu Utama</i>	172
GAMBAR 4.52. <i>Tampilan Daftar User</i>	173

GAMBAR 4.53. Tampilan Tambah Data <i>User</i>	173
GAMBAR 4.54. Tampilan Daftar Departemen	174
GAMBAR 4.55. Tampilan Tambah Data Departemen	174
GAMBAR 4.56. Tampilan Daftar Area	174
GAMBAR 4.57. Tampilan Tambah Data Area	175
GAMBAR 4.58. Tampilan Daftar Klausul	175
GAMBAR 4.59. Tampilan Tambah Data Klausul	175
GAMBAR 4.60. Tampilan Daftar Klasifikasi	176
GAMBAR 4.61. Tampilan Tambah Data Klasifikasi	176
GAMBAR 4.62. Tampilan Daftar Periode <i>Audit</i> AMI	176
GAMBAR 4.63. Tampilan Tambah Data Periode <i>Audit</i> AMI	177
GAMBAR 4.64. Tampilan Daftar Periode <i>Audit</i> GMP	177
GAMBAR 4.65. Tampilan Tambah Data Periode <i>Audit</i> GMP	177
GAMBAR 4.66. Tampilan Daftar Periode <i>Audit Morning Round</i>	178
GAMBAR 4.67. Tampilan Tambah Data Periode <i>Audit Morning Round</i>	178
GAMBAR 4.68. Tampilan Menu Utama <i>Auditor</i>	178
GAMBAR 4.69. Tampilan <i>Form Audit</i> AMI	179
GAMBAR 4.70. Tampilan Daftar <i>Audit</i> AMI	179
GAMBAR 4.71. Tampilan <i>Form Audit</i> GMP	179
GAMBAR 4.72. Tampilan Daftar <i>Audit</i> GMP	180
GAMBAR 4.73. Tampilan <i>Form Audit Morning Round</i>	180
GAMBAR 4.74. Tampilan Daftar <i>Audit Morning Round</i>	181
GAMBAR 4.75. Tampilan Menu Utama <i>Auditee</i>	181
GAMBAR 4.76. Tampilan <i>Form CAR</i> AMI	181
GAMBAR 4.77. Tampilan <i>Form CAR</i> GMP	182
GAMBAR 4.78. Tampilan <i>Form CRA Morning Round</i>	182
GAMBAR 4.79. Tampilan <i>Approve CAR</i> AMI	182
GAMBAR 4.80. Tampilan <i>Approve CAR Audit</i> GMP	183
GAMBAR 4.81. Tampilan <i>Approve CAR Audit Morning Round</i>	183
GAMBAR 4.82. Tampilan Laporan AMI	183
GAMBAR 4.83. Tampilan Laporan AMI yang Akan Dicetak	184

GAMBAR 4.84. Tampilan Laporan <i>Audit</i> GMP	184
GAMBAR 4.85. Tampilan Laporan <i>Audit</i> GMP yang Akan Dicitak	184
GAMBAR 4.86. Tampilan Laporan <i>Morning Round</i>	185
GAMBAR 4.87. Tampilan Laporan <i>Morning Round</i> yang Akan Dicitak	185



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan teknologi yang semakin canggih kini dunia memasuki era revolusi industri 4.0, dimana era ini sangat mempengaruhi banyak aspek kehidupan baik dalam bidang politik, budaya, ekonomi bahkan sampai pada dunia perindustrian. Perkembangan teknologi saat ini telah memberi pengaruh yang sangat besar bagi dunia teknologi informasi. Berbagai perkembangan teknologi ini banyak membuat perusahaan melakukan pergeseran dari sistem *manual* ke komputerisasi sehingga penyimpanan data menjadi efisien, informasi yang *up to date*, dan cepat dalam penyajian informasi.

Suatu perusahaan memiliki faktor kontrol, informasi dan dokumentasi merupakan hal penting yang dapat mempengaruhi terhadap kualitas mutu pada organisasi atau perusahaan. Salah satunya adalah aktivitas penjaminan mutu perusahaan yang dilakukan dengan cara melakukan *internal audit* pada suatu organisasi atau perusahaan. Tujuan dari pelaksanaan *internal audit* yang dilakukan yaitu untuk memastikan implementasi sistem yang ditetapkan oleh organisasi atau perusahaan berjalan dengan efektif.

PT. ABC merupakan salah satu industri makanan di Indonesia, dimana untuk memastikan implementasi sistem yang telah ditetapkan oleh

perusahaan berjalan dengan efektif atau tidak maka perusahaan ini melakukan suatu aktifitas yaitu *internal audit*. Ada tiga jenis *internal audit* yang dilakukan oleh PT. ABC yaitu *Audit Mutu Internal* (AMI), *Good Manufacturing Practices* (GMP), dan *Morning Round*, dimana ketiga audit ini frekuensi pelaksanaannya berbeda-beda. *Audit Mutu Internal* (AMI) dilakukan setiap 6 bulan sekali, *audit Good Manufacturing Practices* (GMP) dilakukan setiap 2 bulan sekali, dan *audit Morning Round* dilakukan setiap 1 minggu sekali. Sampai saat ini *internal audit* yang dilakukan pada PT. ABC masih menggunakan cara manual yaitu hasil *report* dan *monitoring audit* masih menggunakan *form* atau yang diolah menggunakan *office* standar seperti perangkat lunak *Mc. Office Excel*, kemudian hasil *audit* tersebut dikirim kepada semua departemen melalui *email* sehingga dalam pencarian informasi dan *monitoring* status mengenai *internal audit* kurang efektif dan efisien karena PIC yang bersangkutan miss komunikasi serta email yang dikirim terdapat kemungkinan tidak tersampaikan atau terbaca.

Dalam permasalahan yang terjadi selanjutnya adalah pada saat admin menginput data hasil *audit*, tidak sedikit tulisan dari *auditor* tidak terbaca sehingga menyulitkan atau menghambat admin dalam proses penginputan hasil *audit*. Selanjutnya untuk *memonitoring* status *internal audit* secara keseluruhan seperti *Audit Mutu Internal* (AMI), *Good Manufacturing Practices* (GMP), dan *Morning Round*, departemen yang berkaitan (*auditee*) harus menghubungi PIC yang bersangkutan melalui

pesawat telepon karena penyimpanan masing-masing hasil *audit* dikelola oleh PIC/admin yang berbeda. Kendala lain PIC tersebut tidak selalu berada ditempat sehingga kurang efektif dan efisien dalam pencarian informasi hasil *audit* secara keseluruhan. Dengan kurangnya efektif dan efisien baik dalam proses *audit* maupun dalam menyajikan informasi mengenai hasil *audit* maka akan mempengaruhi mutu atau kualitas produk, mempengaruhi citra perusahaan, *break down system*, dan dapat mengurangi target produksi.

Oleh karena itu, diperlukan integrasi antar proses *audit* dan sistem informasi pengolahan hasil *audit* secara terkomputerisasi, sehingga diharapkan agar bisa memperoleh informasi mengenai pengendalian *internal* dan tercapainya efektifitas serta efisiensi yang tinggi serta keakuratan data dan kecepatan dalam mencari informasi hasil audit. Dari uraian diatas penulis tertarik mengambil judul “Analisa dan Perancangan Sistem Informasi *Monitoring* Hasil *Internal Audit* Berbasis *Web* pada PT. ABC”.

1.2. Identifikasi Masalah

Berdasarkan latar belakang masalah yang penulis paparkan diatas, maka penulis dapat mengidentifikasi masalah sebagai berikut :

1. Laporan hasil audit masih dilakukan dengan menggunakan *form* yang diolah menggunakan *Ms. Office Excel*, kemudian hasil audit dikirim kepada semua departemen yang bersangkutan melalui *email* sehingga dalam pencarian informasi dan *monitoring* status mengenai *internal audit* kurang efektif dan efisien karena PIC yang bersangkutan kurang

komunikasi serta *email* yang dikirim terdapat kemungkinan tidak tersampaikan atau terbaca.

2. Pada saat admin melakukan *input* data hasil *audit*, tidak sedikit tulisan dari *auditor* tidak terbaca yang menyulitkan atau menghambat admin dalam proses *input* hasil *audit*.
3. Dalam melakukan *monitoring* status *internal audit* secara keseluruhan, departemen yang berkaitan (*auditee*) harus menghubungi PIC yang bersangkutan melalui pesawat telepon karena penyimpanan masing-masing hasil *audit* dikelola oleh PIC/admin yang berbeda.
4. PIC/admin yang mengelola hasil audit tidak selalu berada ditempat sehingga kurang efektif dan efisien dalam pencarian informasi hasil *audit* secara keseluruhan.

1.3. Ruang Lingkup / Batasan Masalah

Ruang lingkup dibuat agar penulis tahu batasan-batasan yang akan dibahas dan tidak ke hal yang lain, dan masalah yang akan dibahas oleh penulis saat ini yaitu :

1. Sistem yang dibuat adalah berbasis web.
2. Sistem yang dibuat lebih ditekankan kepada *monitoring* hasil audit.
3. Sistem yang dibuat hanya untuk 3 *internal audit* yaitu *Audit Mutu Internal* (AMI), *Good Manufacturing Practices* (GMP), dan *Morning Round*.
4. Tidak mengelola *external audit*.

1.4. Perumusan Masalah

Berdasarkan latar belakang masalah yang penulis paparkan diatas, maka penulis dapat merumuskan masalah sebagai berikut :

1. Bagaimana sistem pengelolaan *monitoring* hasil *internal audit* pada PT. ABC saat ini?
2. Bagaimana merancang dan membangun sistem *monitoring* hasil *Internal Audit* berbasis *web* pada PT. ABC yang efektif dan efisien?

1.5. Tujuan Penelitian

Tujuan kegiatan penelitian ini adalah :

1. Untuk mengetahui sistem pengelolaan *monitoring* hasil *internal audit* pada PT. ABC saat ini.
2. Untuk merancang dan membangun sistem *monitoring* hasil *internal audit* berbasis *web* pada PT. ABC yang efektif dan efisien.

1.6. Manfaat Penelitian

Manfaat kegiatan penelitian ini adalah :

1. Bagi Perusahaan
 - a) Penelitian ini diharapkan dapat memberikan kemudahan pada proses *internal audit* dalam mendapatkan data atau informasi semua hasil *internal audit* yang dilakukan di dalam perusahaan secara efektif dan efisien.
 - b) Memberikan kemudahan dalam *monitoring* hasil *internal audit* yang dilakukan di dalam perusahaan.

- c) Melatih atau membiasakan tentang integrasi data antara *auditee* dengan *auditor*.

2. Bagi STMIK Insan Pembangunan

- a) Hasil penelitian ini diharapkan dapat menjadi literatur yang bermanfaat dalam memperkaya ilmu yang berkaitan dengan sistem informasi.
- b) Bahan informasi atau referensi bagi mahasiswa untuk penelitian tentang audit.

3. Bagi Penulis

- a) Sebagai implementasi dari hasil belajar atau penerapan ilmu selama belajar di STMIK Insan Pembangunan.
- b) Sebagai pengalaman langsung tentang merancang dan membangun sistem informasi *monitoring internal audit* dengan menggunakan bahasa pemrograman *PHP* dan *MySQL* sesuai dengan ilmu yang diperoleh selama belajar di STMIK Insan Pembangunan.

4. Bagi Peneliti Selanjutnya

Manfaat bagi peneliti selanjutnya yaitu sebagai referensi untuk menganalisa dan mengembangkan sistem yang telah dibuat, sehingga sistem yang ada bisa lebih ter-*update* lagi.

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1. Pengertian Dasar Sistem

a. Pengertian Sistem

Menurut Lani Sidharta dalam Yosy Arisandi dkk (2017:48) “sistem merupakan sekumpulan unsur-unsur yang saling berkaitan yang secara bersama beroperasi untuk meraih tujuan yang sama”.

Menurut Jogianto dalam Jeperson Hutahaeen (2014:1) mengemukakan bahwa “sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

Menurut Lani Sidharta dalam Jeperson Hutahaeen (2014:2) “sistem adalah himpunan dari bagian-bagian yang saling berhubungan yang secara bersama-sama mencapai tujuan-tujuan yang sama”.

Menurut Jeffry Fitz Gerald dalam Yosi Arisandi dkk (2017:49) sistem merupakan sebuah jaringan kerja dari pada prosedur-prosedur yang saling berkaitan, berkumpul secara bersama-sama agar bisa beroperasi sebuah kegiatan atau menyelesaikan tujuan tertentu.

Berdasarkan beberapa pendapat diatas, dapat penulis simpulkan bahwa sistem merupakan suatu unsur-unsur atau jaringan kerja dari prosedur yang saling berkaitan, berkumpul hingga membentuk satu kesatuan yang utuh untuk mencapai suatu tujuan tertentu.

b. Karakteristik Sistem

Menurut Jeperson Hutahaeen (2014:3), agar sistem itu dikatakan baik memiliki karakteristik yaitu :

1) Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan, komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2) Batasan sistem.

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3) Lingkungan luar sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah luar batas dari sistem yang berpengaruh operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap dijaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

4) Penghubung sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini

memungkinkan sumber-sumber daya mengalir dari subsistem yang lainnya. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5) Masukan sistem (*input*)

Masukan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*maintenance input*), dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem *computer program* adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

6) Keluaran sistem (*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasi menjadi keluaran yang berguna dan sisa pembuangan. Contoh komputer menghasilkan panas dan merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

2.1.2. Konsep Dasar Informasi

a. Definisi Informasi

Menurut George H. Bordnar dalam Yosy Arisandy dkk (2017:77) “informasi adalah data yang diolah sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat”.

Menurut Sri Mulyani (2016:17) informasi merupakan data yang sudah diolah yang ditunjukkan untuk seseorang, organisasi ataupun siapa saja yang membutuhkan informasi akan menjadi berguna apabila objek yang menerima informasi membutuhkan informasi tersebut.

Menurut Andri Kristanto (2018:7) “informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima”.

Menurut Davis dalam Abdul Kadir (2014:45) “informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti sebagai penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang”.

Menurut Mc Fadden dkk, dalam Abdul Kadir (2014:45) mendefinisikan “informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut”.

Berdasarkan beberapa pendapat diatas, dapat penulis simpulkan bahwa informasi merupakan suatu data yang telah diolah menjadi sesuatu yang berguna, dapat diterima dan lebih berarti untuk mengambil suatu keputusan yang tepat baik untuk saat ini ataupun dimasa yang akan datang.

b. Definisi Data

Menurut Abdul Kadir (2014:44) menyatakan bahwa “secara konsep data adalah deskripsi tentang benda, kejadian, aktifitas, dan transaksi yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai”.

c. Kualitas Informasi

Menurut Abdul Kadir (2014:55) menyatakan bahwa istilah kualitas informasi (*quality of information*) dipakai untuk menyatakan informasi yang baik. Kualitas informasi sering kali diukur berdasarkan :

- 1) Akurasi
- 2) Ketepatan waktu
- 3) Relevansi

Kualitas informasi “dapat dianalogikan sebagai pilar-pilar dalam bangunan dan menentukan baik tidaknya pengambilan keputusan” (Burch dan Grudnitski dalam Abdul kadir, 2014:56).

Akurat berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Tepat waktu berarti informasi yang datang pada penerima tidak boleh terlambat karena informasi yang sudah usang tidak mempunyai nilai lagi. Relevansi berarti informasi tersebut mempunyai manfaat untuk pemakainya.

d. Nilai Informasi

Menurut Jogiyanto dalam Abdul Kadir (2014:56) “nilai informasi (*value of information*) ditentukan oleh dua hal yaitu, manfaat dan biaya untuk mendapatkannya”.

Suatu informasi dianggap bernilai kalau manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya.

2.1.3. Konsep Dasar Sistem Informasi

Menurut O'Brien dalam Yosy Arisandy dkk (2017:83) sistem informasi merupakan suatu kombinasi dari setiap unit yang dikelola oleh user atau manusia, hardware (perangkat keras komputer), software (perangkat lunak), jaringan komputer dan jaringan komunikasi data (komunikasi), dan juga database (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi tentang suatu organisasi. Jadi, pada dasarnya sistem informasi memang harus memiliki elemen-elemen tersebut agar dapat berguna dan juga bekerja dengan optimal.

Menurut Hutahean Jeperson (2014: 13) sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan.

Menurut Andri Kristanto (2018:12) Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut.

Menurut Abdul Kadir (2014:8) sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi dan prosedur kerja) ada sesuatu yang diproses (data menjadi informasi) dan dimaksudkan untuk mencapai suatu sasaran atau tujuan.

Menurut Hall dalam Abdul Kadir (2014:9) "sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai".

Berdasarkan beberapa pendapat diatas dapat penulis simpulkan bahwa informasi merupakan gabungan dari inputan, *hardware*, *software*, prosedur, jaringan komputer dan jaringan komunikasi data (komunikasi), dan juga database (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi yang berguna untuk mendukung dalam pengambilan keputusan didalam suatu organisasi.

Dalam Abdul Kadir (2014:71) sistem informasi mengandung komponen-komponen sebagai berikut :

- a. Perangkat keras (*hardware*), yang mencakup peranti-peranti fisik seperti komputer dan printer.
- b. Perangkat lunak (*software*) atau program, yaitu sekumpulan instruksi yang memungkinkan perangkat keras memproses data.
- c. Prosedur, yaitu sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikendaki.
- d. Orang, yaitu semua pihak yang bertanggungjawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.
- e. Basis data (*database*) yaitu, kumpulan *table*, hubungan dan lain-lain yang berkaitan dengan penyimpanan data
- f. Jaringan komputer dan komunikasi data, yaitu sistem penghubung yang memungkinkan sumber (*resources*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

2.1.4. Sistem Informasi Manajemen

Menurut Gordon Joseph F. Kelly dalam Danang Sunyoto (2014:5) sistem informasi manajemen adalah proses komunikasi dimana informasi masukan (*input*) direkam, disimpan, dan diperoleh kembali (diproses) bagi keputusan (*output*) mengenai perencanaan, pengoperasian, dan pengawasan.

Dalam Yosy Arisandy dkk (2017:11) sistem informasi manajemen adalah suatu sistem antar manusia-mesin (komputer dan instrument didalamnya) yang terpadu guna menyediakan informasi yang mendukung proses, manajemen, dan fungsi pengambilan keputusan dalam suatu organisasi.

Abdul Kadir (2014:106) menyatakan bahwa sistem informasi manajemen adalah sistem informasi yang digunakan untuk menyajikan informasi yang digunakan untuk mendukung operasi, manajemen, dan pengambilan keputusan dalam suatu organisasi.

Menurut Sri Mulyani (2016:23) “Sistem informasi manajemen merupakan sistem informasi yang sudah terkomputerisasi yang bekerja karena adanya interaksi manusia dan komputer”.

Dari beberapa pendapat diatas dapat penulis simpulkan bahwa sistem informasi manajemen merupakan suatu sistem antar manusia-mesin yang saling terpadu dalam menyediakan informasi yang dibutuhkan untuk mendukung fungsi operasi, fungsi manajemen, serta pengambilan keputusan dalam suatu organisasi.

2.1.5. Pengertian Analisis dan Perancangan

a. Pengetian Analisis Sistem

Menurut Rosa. A. S dan M. Shalahuddin (2018:18) kegiatan untuk melihat sistem yang sudah berjalan, melihat bagian mana yang bagus dan tidak bagus, dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru.

Menurut Jaluanto Sunu Punjul Tyoso (2016:18) “analisis sistem merupakan contoh yang baik dari pendekatan sistem untuk memecahkan masalah”

Berdasarkan beberapa pendapat diatas, dapat penulis simpulkan bahwa analisa sistem merupakan suatu kegiatan yang mendefinisikan hal-hal apa yang akan dilakukan oleh aplikasi dan pengembangan sistem

yang berjalan kemudian mengevaluasi semua kekurangan agar dapat terpenuhi sesuai dengan yang diharapkan.

Prinsip-prinsip analisis sistem menurut Juluanto Sunu Punjul Tyso (2016:18) adalah :

1) Mendefinisikan Masalah

Masalah yang akan dipecahkan dengan sistem diatur berkenaan dengan lingkungan tempat sistem berinteraksi

2) Menyatakan Sasaran Sistem

Tujuan umum dan khusus yang ingin dicapai yang berkaitan dengan keefektifan ditetapkan dan diumumkan kepada pihak-pihak yang berkepentingan.

3) Menetapkan Batasan Sistem (*system boundaries*)

Pembatas antara sistem yang baru dengan lingkungannya harus terperinci hubungan sistem (*interface*) yang berkaitan dengan masukan dan keluaran harus ditegaskan.

4) Menetapkan Kendala Sistem

Kendala pada sistem dan proses pengembangan, seperti biaya dan jangka waktu untuk pengembangan sistem, harus dipastikan.

5) Dekomposisi Sistem

Sistem dipecah kedalam sub-sub sistem yang saling terkait dan berhubungan dengan lingkungannya. Hubungan antar subsistem ditentukan sehingga seorang analis mampu melihat sistem dengan

terperinci. Subsistem yang berada pada tingkat bawahlah yang nantinya dirancang dan menjadi bagian sistem yang ditetapkan.

b. Pengertian Perancangan Sistem

Menurut Soetam Rizky Wicaksono (2017:193) menyatakan bahwa perancangan perangkat lunak merupakan sebuah proses yang berkelanjutan dari analisa dan didalamnya melakukan identifikasi hasil analisa serta menghasilkan konsep dasar untuk kepentingan pengembangan perangkat lunak.

Menurut Rosa A. S dan M. Shalahuddin (2018:23), desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengkonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun pengguna sumber daya, kepuasan batasan pada proses dari segi biaya, waktu, dan perangkat.

Menurut Jhon dan Gary dalam Rosa A.S (2014:196), desain sistem dapat didefinisikan sebagai gambaran, perancangan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

Dari beberapa pendapat diatas dapat penulis simpulkan bahwa perancangan sistem merupakan sebuah proses yang berkelanjutan dari analisa sistem untuk mendefinisikan sesuatu yang akan diimplementasikan sesuai dengan kebutuhan perangkat lunak.

Menurut Soetam Rizky Wicaksono (2017:197-199) sebuah perancangan perangkat lunak yang baik diwajibkan memenuhi syarat-syarat berikut :

1) Fleksibel

Hasil perancangan harus dapat menyesuaikan diri dengan kebutuhan pengguna yang sewaktu-waktu dapat berubah.

2) Mudah ditransfer

Dalam ruang lingkup perancangan perangkat lunak, yang dimaksud dengan mudah di *transfer* adalah hasil perancangan yang dapat dengan mudah diterapkan dilingkungan perangkat keras yang berbeda.

3) Mudah dimodifikasi

Telah lazim diketahui, dan juga telah dijelaskan di dalam bab mengenai siklus hidup, bahwa perangkat lunak akan mengalami masa modifikasi ulang. Dalam kasus tersebut, maka hasil perancangan yang sudah ada sebelumnya harus dapat dengan mudah di modifikasi untuk kepentingan versi perangkat lunak yang baru.

4) Mudah digunakan

Sebuah hasil perancangan yang baik harus mampu menghasilkan pengerjaan perangkat lunak yang mudah digunakan oleh pengguna.

5) Handal

Syarat handal dalam perancangan berarti bahwa hasil perancangan perangkat lunak mampu meminimalkan kesalahan yang dibuat oleh pengembang perangkat lunak, sehingga hasil perancangan mampu diimplementasikan dengan baik.

6) Aman

Hasil perancangan yang baik juga harus memperhatikan segi keamanan perangkat lunak yang dirancang sehingga tidak akan membuat pengguna menjadi cemas.

7) Tidak mahal

Tentu saja tidak semua pengguna memiliki dana yang berlebih dalam mengimplementasikan sebuah perangkat lunak. Oleh karena itu, perancangan yang dibuat juga harus menyesuaikan dengan anggaran yang telah disediakan oleh pengguna.

2.1.6. Pengertian Monitoring

Monitoring adalah kegiatan memantau yang dilakukan secara rutin mengenai kemajuan proses yang sedang berjalan. *Monitoring* merupakan bagian dari pengamatan, pengumpulan informasi, dan pengawasan.

Menurut Peraturan Pemerintah Nomor 39 Tahun 2006, *monitoring* merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu, dengan tujuan agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan tindakan selanjutnya yang diperlukan. *Monitoring*, bertujuan untuk mengamati/ mengetahui perkembangan dan kemajuan, identifikasi dan permasalahan serta antisipasinya/ upaya pemecahannya.

Sedangkan Menurut Humas (2015) menyatakan bahwa *monitoring* adalah “aktifitas yang ditujukan untuk memberikan informasi tentang sebab dan akibat dari suatu kebijakan yang sedang dilaksanakan”.

Monitoring (pemantauan) adalah “kegiatan untuk mengamati program atau proyek. Pada suatu rencana kemudian diikuti dengan “(Priambodo, 2014:9). Sedangkan *monitoring* memiliki fungsi sebagai berikut :

a. Ketaatan (*compliance*)

Monitoring menentukan apakah tindakan *administrator*, staff, dan semua yang terlibat mengikuti standar dan prosedur yang ditetapkan.

b. Pemeriksaan (*auditing*)

Monitoring menetapkan apakah sumber dan layanan yang diperuntukkan bagi pihak tertentu (target) telah mencapai mereka.

c. Laporan

Monitoring menghasilkan informasi yang membantu “menghitung” hasil perubahan sosial dan masyarakat sebagai akibat *implementasi* kebijaksanaan sesudah periode waktu tertentu.

d. Penjelasan (*explanation*)

Menghasilkan informasi yang membantu menjelaskan bagaimana akibat kebijaksanaan dan mengapa antara perencanaan dan pelaksanaan tidak cocok.

Sedangkan *monitoring* dibagi menjadi 2 jenis *monitoring*, sebagai berikut :

a. *Monitoring* aspek proses

Proses merupakan jembatan untuk menghasilkan *output*. Dari pemanfaatan *input*, diproses maka menghasilkan *output*. Pilihan proses yang tepat dan mampu memanfaatkan *input* secara efisien, merupakan persoalan yang harus diperhatikan tim manajemen.

b. *Monitoring* aspek hasil (*output*)

Monitoring aspek hasil (*output*) difokuskan untuk mengamati perkembangan pencapaian *output*. Suatu perencanaan yang baik, sejak awal harus menetapkan target/*output* serta tahapan pencapaian.

Dari penjelasan teori-teori diatas dapat penulis simpulkan bahwa *monitoring* adalah pemantau suatu proses secara periodik mengenai perkembangan dari objek yang sedang dikerjakan, dengan tujuan untuk mengetahui informasi aktivitas atau hambatan apa saja yang terjadi untuk mengambil suatu langkah-langkah yang tepat.

2.1.7. Konsep Dasar *Internal Audit*

a. Pengertian Audit Internal

Menurut *Institut Of Internal Auditors* (IIA) audit internal merupakan kegiatan *assurance* dan konsultasi yang dilakukan secara independent dan objektif yang dirancang untuk memberikan nilai tambah dan meningkatkan kegiatan operasi organisasi. Audit internal membantu organisasi mencapai tujuannya melalui suatu pendekatan yang sistematis dan teratur untuk mengevaluasi dan meningkatkan efektivitas pengelolaan resiko, pengendalian, dan tata kelola.

Menurut Mathius Tandionong (2016: 67-68) menyatakan bahwa audit internal adalah kegiatan penilaian independen yang dibentuk dalam suatu organisasi untuk memeberikan jasa kepada manajemen dalam bentuk penelaahan kegiatan organisasi. Audit internal merupakan pengendalian manajerial yang fungsinya mengukur dan mengevaluasi keefektifan sistem pengendalian lain.

Tujuan audit internal adalah membantu semua anggota manajemen dalam mengelola secara efektif pertanggungjawabannya dengan cara menyediakan analisis, penilaian, rekomendasi, dan komentar-komentar yang berhubungan dengan kegiatan yang ditelaah.

b. Pengertian Audit Mutu Internal (AMI)

Menurut Ida Widaningsih selaku *Quality System* pada PT. ABC menyatakan bahwa audit mutu internal (AMI) adalah suatu sistem yang dijalankan atau diterapkan pada suatu organisasi untuk memastikan sistem berjalan efektif dan sesuai dengan persyaratan yang telah ditetapkan (Wawancara, 31 Juli 2019).

c. Penertian Audit *Good Manufacturing Practices* (GMP)

Menurut Ida Widaningsih selaku *Quality System* pada PT. ABC menyatakan bahwa audit *good manufacturing practices* (GMP) adalah suatu sistem yang dijalankan atau diterapkan pada suatu organisasi untuk memastikan system keamanan pangan mulai dari sumber daya manusia, fasilitas produksi, hygiene dan sanitasi, serta 5R secara periodik sesuai jadwal yang telah ditetapkan (Wawancara, 31 Juli 2019).

d. Pengertian Audit *Morning Round*

Menurut Ida Widaningsih selaku *Quality System* pada PT. ABC menyatakan bahwa audit *morning round* adalah audit yang dilaksanakan oleh *department head* untuk melihat kesesuaian kondisi setiap departemen yang bersifat insfeksi mendadak setiap pagi yang dilakukan setiap 1 minggu sekali (Wawancara, 31 Juli 2019).

e. Pengertian *Auditor* dan *Auditee*

Menurut Arens dkk dalam buku Mathius Tandiontong (2016:71) menyatakan bahwa “*auditor* adalah seorang independen dan kompeten yang melaksanakan audit dan *auditee* adalah organisasi yang dilakukan proses audit”.

2.1.8. Pengertian *SDLC* (*System Devolepment Life Cycle*)

SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* “adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya” (Rosa A. S dan M. Shalahuddin, 2018:26).

Dalam Rosa A. S dan M. Shalahuddin (2018:26-28) tahapan-tahapan yang ada pada SDLC secara global adalah sebagai berikut :

1. Inisiasi (*initiation*)

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perancangan (*planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransformasikan kebutuhan detail menjadi kebutuhan yang sudah lengkap, dokumen desain sistem focus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)

Mengkonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data dan mempersiapkan prosedur kasus pengujian, mempersiapkan berkas atau *file* pengujian, pengkodean, pengompilasian, memperbaiki dan membersihkan program, peninjauan pengujian.

7. Integrasi dan pengujian (*integration and test*)

Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh staf penjamin kualitas (*quality assurance*) dan *user*. Menghasilkan laporan analisis pengujian.

8. Implementasi (*implementation*)

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

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

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

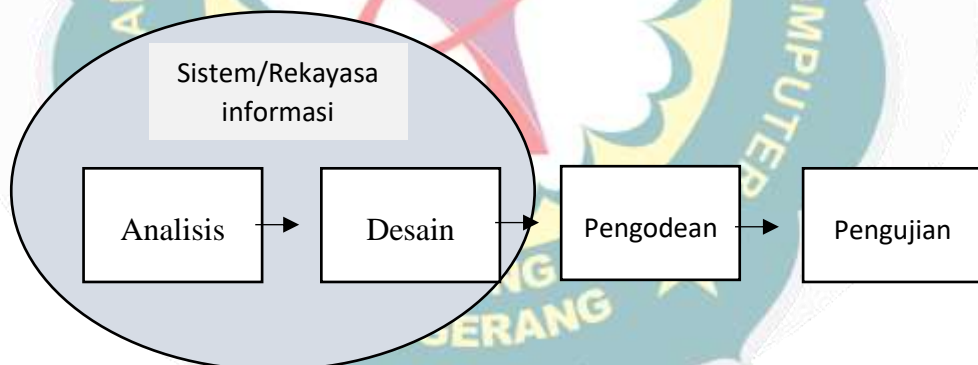
10. Disposisi (*disposition*)

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

2.1.9. Model SDLC (*System Development Life Cycle*)

a. Model *Waterfall*

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*) (Rosa A. S dan M. Shalahuddin, 2018:29-30).



GAMBAR 2.1. Ilustrasi Model *Waterfall*

Sumber: Buku Rekayasa Perangkat Lunak, (Rosa A.S dan M. Shalahuddin, 2018:29)

1) Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi

kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2) Desain

Desain perangkat lunak adalah proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini menranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3) Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4) Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5) Pendukung (*support*) atau pemeliharaan (*maintenance*)

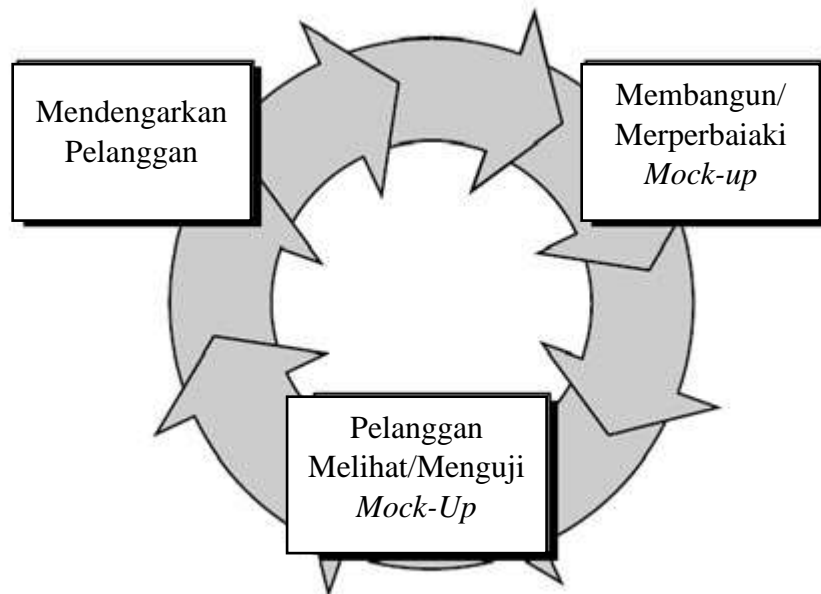
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena

adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian, atau perangkat lunak harus beradaptasi terhadap lingkungan baru, pada tahap ini diharapkan dapat mengulangi proses pengembangan mulai dari analisa spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak yang baru.

b. Model Prototipe

Model prototipe “dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak” (Rosa A. S dan M. Shalahuddin, 2018:31).

Model prototipe dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. “Program prototipe ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau *user*” (Rosa A. S dan M. Shalahuddin, 2018:31).



GAMBAR 2.2. Ilustrasi Model Prototipe

Sumber: Buku Rekayasa Perangkat Lunak, (Rosa A.S dan M. Shalahuddin, 2018:32)

Mock-up adalah sesuatu yang digunakan sebagai model desain yang digunakan untuk mengajar, demonstrasi, evaluasi desain, promosi, atau keperluan lain. Sebuah *mock-up* disebut sebagai prototipe perangkat lunak jika menyediakan atau mampu mendemonstrasikan sebagian fungsi sistem perangkat lunak. “Iterasi terjadi pada pembuatan prototipe sampai sesuai dengan keinginan pelanggan (*customer*) atau *user*” (Rosa A. S dan M. Shalahuddin, 2018:32).

Seiring dengan mengembangkan prototipe maka sistem perangkat lunak yang sebenarnya dikembangkan juga sehingga sesuai dengan kebutuhan pelanggan (*customer*) atau *user* (Rosa A. S dan M. Shalahuddin, 2018:32).

Model prototipe juga memiliki kelemahan sebagai berikut :

1) Pelanggan dapat sering mengubah-ubah atau menambah-tambah spesifikasi kebutuhan karena menganggap aplikasi sudah dengan cepat dikembangkan, karena adanya iterasi ini dapat menyebabkan pengembang banyak mengalah dengan pelanggan karena perubahan atau penambahan spesifikasi kebutuhan perangkat lunak.

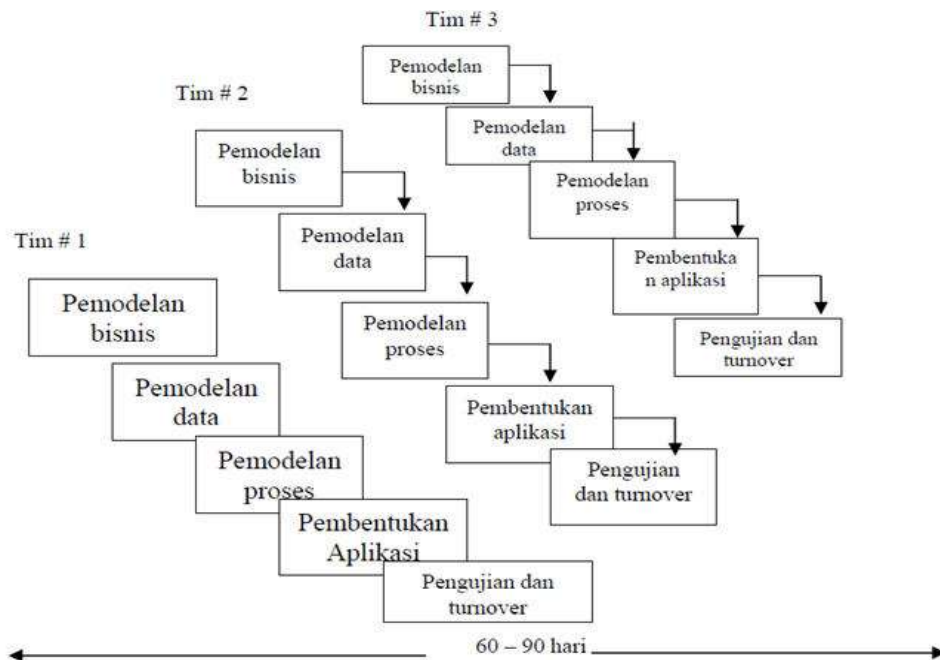
2) Pengembang lebih sering mengambil kompromi dengan pelanggan untuk mendapatkan prototipe dengan waktu yang cepat sehingga pengembang lebih sering melakukan segala cara (tanpa idealis) guna menghasilkan prototipe untuk didemonstrasikan. Hal ini dapat menyebabkan kualitas perangkat lunak yang kurang baik atau bahkan menyebabkan iteratif tanpa akhir

Permasalahan dapat terjadi pada model prototipe, hal ini dapat diatasi dengan melakukan perjanjian antara pengembang perangkat lunak dengan pelanggan (*customer*) atau *user* agar model prototipe hanya digunakan untuk mendefinisikan spesifikasi kebutuhan perangkat lunak, tapi tidak untuk seluruh proses pengembangan seluruh sistem perangkat lunak (Rosa A. S dan M. Shalahuddin, 2018:33).

c. Model *Rapid Application Development* (RAD)

Rapid Application Development (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun versi kecepatan tinggi dengan menggunakan model air terjun untuk

pengembangan setiap komponen perangkat lunak (Rosa A. S dan M. Shalahuddin, 2018:34). Berikut adalah gambar dari model RAD :



GAMBAR 2.3. Ilustrasi Model RAD

Sumber: Buku Rekayasa Perangkat Lunak, (Rosa A.S dan M. Shalahuddin, 2018:35)

1) Pemodelan bisnis

Pemodelan yang dilakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait proses bisnis, informasi apa saja yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yang terkait informasi itu.

2) Pemodelan data

Pemodelan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta realisasinya dengan data-data yang lain.

3) Pemodelan proses

Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data.

4) Pembuatan aplikasi

Mengimplementasikan pemodelan proses dan data menjadi program. Model RAD sangat menganjurkan pemakaian komponen yang sudah ada jika dimungkinkan.

5) Pengujian dan pergantian

Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembang komponen dapat beranjak untuk mengembangkan komponen berikutnya.

Model RAD memiliki kelemahan sebagai berikut :

- 1) Untuk pembuatan sistem perangkat lunak untuk skala besar maka model RAD akan memerlukan sumber daya manusia yang cukup besar untuk membentuk tim-tim yang mengembangkan komponen-komponen.
- 2) Jika tidak ada persetujuan untuk mengembangkan perangkat lunak secara dengan cepat (*rapid*) maka proyek dengan model ini akan gagal karena hanya akan bingung mendefinisikan kebutuhan pelanggan (*customer*) atau *user*.
- 3) Jika sistem perangkat lunak yang akan dibuat tidak bisa dimodulkan (dibagi-bagi menjadi beberapa komponen) maka model RAD tidak

dapat digunakan untuk membuat sistem perangkat lunak ini karena terlalu banyak campur tangan antar tim.

- 4) Model RAD tidak cocok digunakan untuk sistem perangkat lunak yang memiliki resiko teknis sangat tinggi, misalnya menggunakan teknologi baru yang belum banyak dikenal dan dikuasai pengembang.

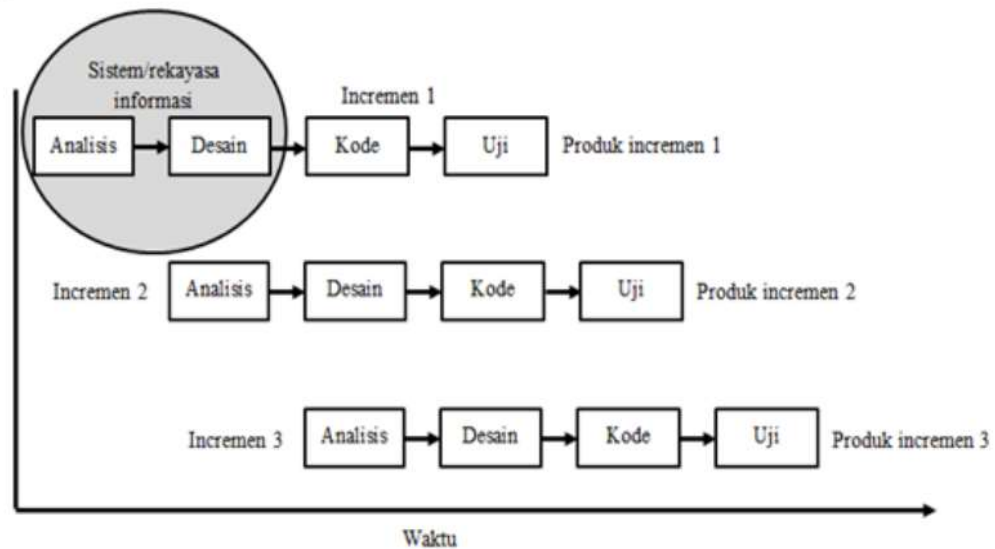
Model RAD cocok diterapkan apabila memenuhi kriteria proyek sebagai berikut :

- 1) Anggota tim sudah berpengalaman mengembangkan perangkat lunak yang sejenis.
- 2) Pengembang sudah memiliki komponen-komponen sistem yang bisa digunakan kembali dalam proyek-proyek tersebut.

d. Model Iteratif

Model Iteratif (*iterative model*) mengkombinasikan proses-proses pada model air terjun dan iteratif pada model prototipe. Model inkremental akan menghasilkan versi-versi perangkat lunak yang sudah mengalami penambahan fungsi untuk setiap pertambahannya (inkremen/*increment*)

(Rosa A. S dan M. Shalahuddin, 2018:38). Berikut adalah gambar dari model iteratif :



GAMBAR 2.4. Ilustrasi Model Iteratif

Sumber: Buku Rekayasa Perangkat Lunak, (Rosa A.S dan M. Shalahuddin, 2018:38)

Model inkremental dibuat untuk mengatasi kelemahan dari model air terjun yang tidak mengakomodasi iterasi, dan mengatasi kelemahan dari metode prototipe yang memiliki proses terlalu pendek dan setiap iteratif prosesnya tidak selalu menghasilkan produk (bisa jadi hanya prototipe). “Model inkremental menghasilkan produk/aplikasi untuk setiap tahapan inkremental” (Rosa A. S dan M. Shalahuddin, 2018:38).

Model inkremental sangat cocok digunakan jika staf yang dimiliki memiliki pergantian (*turnover*) yang tinggi sehingga staf tidak dapat terus ikut dalam pengembangan perangkat lunak. “Mekanisme tahapan inkremental perlu direncanakan terlebih dahulu agar hasil produk dan pengerjaan setiap tahapan inkremental menjadi lebih baik” (Rosa A. S dan M. Shalahuddin, 2018:38).

e. Model Spiral

Model spiral (*spiral model*) memasangkan iteratif pada model prototipe dengan kontrol dan aspek sistematis yang diambil dari model air terjun. Model spiral menyediakan pengembangan dengan cara cepat dengan perangkat lunak yang memiliki versi yang terus bertambah fungsinya (*increment*) (Rosa A. S dan M. Shalahuddin, 2018:39).

Pada iterasi awal maka yang dihasilkan adalah prototipe sedangkan pada iterasi akhir yang dihasilkan adalah perangkat lunak yang sudah lengkap. Model spiral dibagi menjadi beberapa kerangka aktifitas atau disebut juga wilayah kerja (*task region*). Banyaknya wilayah kerja biasanya diantara tiga sampai enam wilayah sebagai berikut :

1) Komunikasi dengan pelanggan (*customer communication*)

Aktifitas ini diperlukan untuk membangun komunikasi yang efektif antara pengembang (*developer*) dan pelanggan (*customer*).

2) Perencanaan (*planning*)

Aktifitas ini diperlukan untuk mendefinisikan sumber daya, waktu, dan informasi yang terkait dengan proyek.

3) Analisa Resiko (*risk analysis*)

Aktifitas ini diperlukan untuk memperkirakan resiko dari segi teknis maupun manajemen.

4) Rekayasa (*engineering*)

Aktifitas ini diperlukan untuk membangun satu atau lebih representasi dari aplikasi perangkat lunak.

5) Konstruksi dan peluncuran (*contruction and release*)

Aktifitas ini dibutuhkan untuk mengontruksi, menguji, melakukan instansi, dan menyediakan dukungan terhadap *user* (misalnya dari segi dokumentasi dan pelatihan).

6) Evaluasi pelanggan (*customer evaluation*)

Aktifitas ini dibutuhkan untuk mendapatkan umpan balik berdasarkan evaluasi representasi perangkat lunak yang dihasilkan dari proses rekayasa dan diimplemntasikan pada tahap instalasi.

2.2. Penelitian Sebelumnya

Penelitian yang dilakukan oleh Dina Istriawan (2018). Penelitian ini berjudul “Analisis dan Perancangan Sistem Informasi *Monitoring* Laporan Harian Pada Area Produksi *Wrapping* di PT. Torabika Eka Semesta Ground 2”. Dengan masalah-masalah yang sering dihadapi adalah data *monitoring* laporan harian dalam manual *form* sehingga bisa rusak dan hilang, serta pembuatan laporan dan pencarian data *monitoring* harian pada area *wrapping* yang dilakukan oleh pengawas membutuhkan waktu yang lama dalam penyajian. *Tools* yang digunakan yaitu permodelan objek menggunakan *UML* (*Unified Modeling Language*). Bahasa program yang digunakan yaitu *Visual Basic* 6.0, dan *Microsoft Access* sebagai *database*-nya. Metode pengembangan sistem yang digunakan adalah model *waterfall*. Hasil penelitian ini menunjukkan bahwa sistem informasi ini mampu mengolah data *monitoring* laporan harian pada produksi *wrapping* di PT. Torabika Eka Semesta Ground 2 dengan baik dibandingkan dengan sistem manualnya.

Penyajian informasi laporan monitoring area produksi *wrapping* PT. Torabika Eka Semesta Ground 2 lebih cepat dan akurat.

Penelitian yang dilakukan oleh Sugeng Arif Barianto (2018). Penelitian dengan judul “Analisa dan Perancangan Sistem Informasi *Monitoring Pelaksanaan Improvement* di PT. 123 Divisi *Candy* Departemen Teknik *Project and Improvement*”. Latar belakang masalah yang mendasari penelitian ini adalah *user* maupun staff teknik tidak dapat mengetahui *update progress* dari pelaksanaan *improvement* serta informasi yang ada pada papan *jobdesc* kurang informatif dan kurang *up to date* karena tidak sesuai dengan aktual yang ada di lapangan. Model pengembangan sistem yang dipakai yaitu *waterfall*. Tools yang digunakan yaitu *UML (Unified Modelling Language)*. Sedangkan bahasa pemrograman yang dipakai adalah *PHP* dan *MySQL* sebagai *database*-nya. Dari penelitian yang dilakukan Sugeng Arif Barianto ini dihasilkan Sistem Informasi *Monitoring Pelaksanaan Improvement* sehingga proses dan prosedur *update* status dari progres pelaksanaan *improvement* lebih efektif dan efisien, serta informasi yang ditampilkan lebih akurat dan *realtime*.

Penelitian yang dilakukan oleh Andi Yulianto (2015). Penelitian dengan judul “Analisa dan Perancangan Sistem Informasi *Maintenance* untuk *Monitoring* Mesin Produksi Pada PT. Cingluh Indonesia 2”. Penelitian ini diangkat oleh Andi Yulianto berdasarkan latar belakang masalah yaitu sulitnya petugas *maintenance* dalam memonitor setiap mesin karena sistem yang digunakan belum terintegrasi dan masih *manual* dengan kata lain masih

menggunakan buku *record* mesin yang ada disetiap mesin. Metode pengembangan sistem menggunakan *SDLC* model *waterfall*. Sedangkan *tools* yang digunakan adalah permodelan objek *UML* (*Unified Modelling Language*). Bahasa pemrograman yang dipakai yaitu *Visual Basic.Net* dengan *SQL Server* sebagai *database*-nya. Dari penelitian tersebut dihasilkan aplikasi yang dapat berfungsi sebagai alat bantu dalam proses *monitoring* mesin produksi mulai dari perawatan, perbaikan serta penyampaian informasi kerusakan mesin.

TABEL 2.1. Penelitian Sebelumnya

Deskripsi	Penelitian Sebelumnya			Penelitian Saat Ini
Penulis	Dina Istriawan	Sugeng Arif Barianto	Andi Yulianto	Nuramanah
Tahun	2018	2018	2015	2019
Judul	Analisis dan Perancangan Sistem Informasi <i>Monitoring</i> Laporan Harian Pada Area Produksi <i>Wrapping</i> di PT. Torabika Eka Semesta Ground 2	Analisa dan Perancangan Sistem Informasi <i>Monitoring</i> Pelaksanaan <i>Improvement</i> di PT. 123 Divisi <i>Candy</i> Departemen Teknik <i>Project and Improvement</i>	Analisa dan Perancangan Sistem Informasi <i>Maintenance</i> untuk <i>Monitoring</i> Mesin Produksi Pada PT. Cingluh Indonesia 2	Analisa dan Perancangan Sistem Informasi <i>Monitoring</i> Hasil <i>Internal Audit</i> berbasis <i>Web</i> pada PT. ABC
Metode Pengembangan	SDLC	SDLC	SDLC	SDLC
Model	<i>Waterfall</i>	<i>Waterfall</i>	<i>Waterfall</i>	<i>Waterfall</i>
Bahasa Pemrograman	<i>Visual Basic 6</i>	<i>PHP</i>	<i>Visual Basic. Net</i>	<i>PHP</i>

<i>Database</i>	<i>Microsoft Access</i>	<i>MySQL</i>	<i>SQL Server</i>	<i>MySQL</i>
Objek	<i>Monitoring Laporan Harian Pada Area Produksi Wrapping di PT. Torabika Eka Semesta Ground 2</i>	<i>Monitoring Pelaksanaan Improvement di PT. 123 Divisi Candy Departemen Teknik Project and Improvement</i>	<i>Monitoring Mesin Produksi Pada PT. Cingluh Indonesia 2</i>	<i>Monitoring Hasil Internal Audit PT. ABC</i>

2.3. Tools yang digunakan

2.3.1. UML (*Unified Modelling Language*)

Menurut Philippe Kruchten dalam Astria Firman, dkk (2016:31), *UML* adalah bahasa grafis untuk visualizing, specifying, constructing and documenting setiap artifak dari sistem perangkat lunak. Alistair Cockburn dalam Astria Firman dkk (2016:31) menjelaskan bahwa *use case* merupakan kontrak untuk perilaku. Kontrak mendefinisikan dengan cara bagaimana suatu aktor menggunakan sistem berbasis komputer untuk menyelesaikan permasalahan-permasalahan yang dihadapinya.

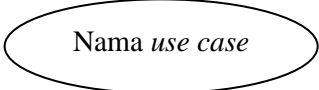
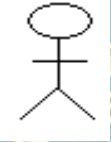

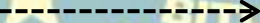
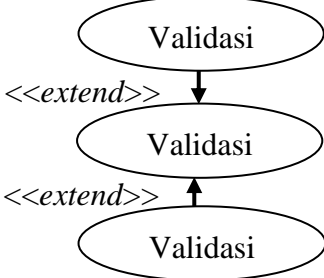
UML (unified modelling language) adalah salah satu standar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A. S dan M. Shalahuddin, 2018:133).

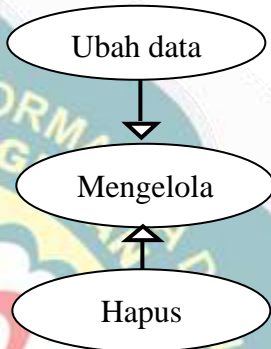

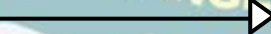
a. *Use Case Diagram*

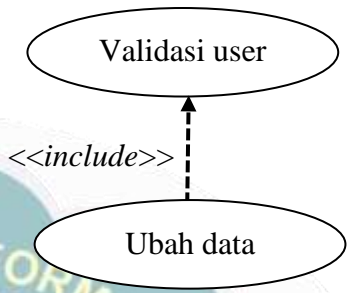
Use case diagram atau diagram *use case* merupakan permodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu (Rosa A. S dan M. Shalahuddin, 2018:133).

Simbol-simbol yang ada pada diagram *use case*:

TABEL 2.2. Simbol *Use Case*

No	Simbol	Deskripsi
1.	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit-unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i>
2.	<p>Aktor/<i>actor</i></p> 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3.	<p>Asosiasi/<i>association</i></p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
4.	<p>Ekstensi/<i>extend</i></p> <p><< <i>extend</i> >></p> 	<p>Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, missal</p> 

		Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya
5.	<p>Generalisasi/ <i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasi (umum)</p>
6.	<p>Menggunakan/ <i>include/uses</i> <code><<include>></code></p>  <p><code><<uses>></code></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan atau sebagai syarat dijalankan <i>use case</i> ini</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut: 


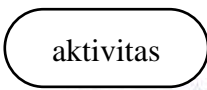




		<p>2. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut :</p>  <pre> graph BT A([Ubah data]) -.-> <<include>> B([Validasi user]) </pre> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>
--	--	--

Sumber: Buku Rekayasa Perangkat Lunak (Rosa A. S dan M. Shalahuddin, 2018:156-158)

b. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

TABEL 2.3. Simbol *Activity Diagram*

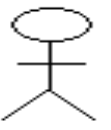

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih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


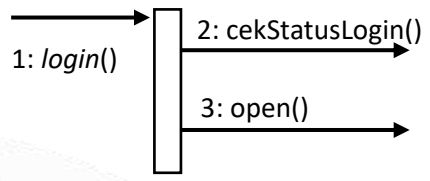




Sumber: Buku Rekayasa Perangkat Lunak (Rosa A. S dan M. Shalahuddin, 2018:162-163)

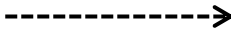
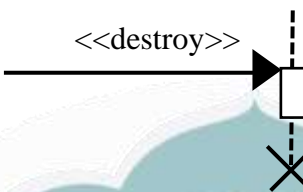
c. *Sequence Diagram*

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

TABEL 2.4. Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Nama aktor Atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama aktor</u> </div> <p>Tanpa waktu aktif</p>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
2.	<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek
3.	<p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>Nama objek: nama kelas</u> </div>	Menyatakan objek yang berinteraksi pesan

4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya</p>  <p>Maka cekStatusLogin() dan open() dilakukan di dalam metode login()</p> <p>Aktor tidak memiliki waktu aktif</p>
5.	<p>Pesan tipe <i>create</i></p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
7.	<p>Pesan tipe <i>send</i></p> <p>1: masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>

8.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>
10.	Kelas entitas 	Menangai informasi yang mungkin akan disimpan secara permanen
11.	Kelas Control 	Mengkoordinasikan kegiatan-kegiatan terhadap kelas lainnya
12.	Kelas Pembatas 	Kelas yang terletak diantara sistem dan sekelilingnya

Sumber: Buku Rekayasa Perangkat Lunak (Rosa A. S dan M. Shalahuddin, 2018:165-167)

d. *Class Diagram*

Diagram *class* atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem (Rosa A. S dan M. Shalahuddin, 2018:141)

TABEL 2.5. Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktur sistem
2.	<p>Antarmuka/<i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
6.	<p>Kebergantungan/<i>dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antarkelas
7.	<p>Agregasi/<i>aggregation</i></p> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)

Sumber: Buku Rekayasa Perangkat Lunak (Rosa A. S dan M. Shalahuddin, 2018:146-147)

e. Nilai Kardinalitas

Nilai kardinalitas atau *multiplicity* atau multiplisitas menunjukkan jumlah suatu objek yang dapat berhubungan dengan objek lain. *multiplicity* biasanya ditunjukkan dengan “satu” atau

“banyak”, tetapi secara khusus dapat ditunjukkan pula dengan bilangan integer lebih besar atau sama dengan nol.

TABEL 2.6 Jenis-jenis *Multiplicity*

No.	Indikator	Deskripsi
1.	0...1	Nol atau satu
2.	1	Hanya satu
3.	0...*	Nol atau lebih
4.	1...*	Satu atau lebih
5.	N	Hanya n (dengan $n > 1$)
6.	0...n	Nol sampai n (dengan $n > 1$)
7.	1...n	Satu sampai n (dengan $n > 1$)

Sumber: Buku Analisa Serta Perancangan Sistem Informasi Melalui Pendekatan UML (Hamim Tohari, 2014:86)

2.3.2. XAMPP

Dalam buku Yosef Murya Kusuma Ardhana (2016:14) *XAMPP* merupakan sebuah perangkat lunak gratis sehingga bebas digunakan. *XAMPP* berfungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari apache *HTTP server*, *MySQL database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP*.

Toosl pengembangan *web* ini mendukung teknologi *web* populer seperti *PHP*, *MYSQL* dengan menggunakan perangkat lunak *XAMPP*, *programmer web* dapat menguji aplikasi *web* yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer tanpa perlu terkoneksi ke *internet*. *XAMPP* juga memiliki fitur manajemen *database PHPMyAdmin* seperti pada *server hosting* sungguhan, sehingga pengembang aplikasi *web* berbasis database dengan mudah.

Istilah *XAMPP* diambil dari kata X yang berarti empat sistem operasi apapun seperti *windows*, *linux*, *mac OS*, dan *Solaris* sedangkan A diambil dari kata *Apache*, kemudia M singkatan dari kata *MySQL*

kemudian huruf P singkatan dari *PHP* dan huruf P yang terakhir adalah Perl.

2.3.3. MySQL

Dalam buku Yosef Murya Kusuma Ardhana (2016:14), *MySQL* adalah aplikasi *database server*. *SQL* merupakan kependekan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah *database*. *MySQL* dapat digunakan untuk membuat dan mengelola *database* beserta isinya. *Programmer* dan *user* dapat memanfaatkan *MySQL* untuk menambahkan (*Create*), membaca atau menampilkan (*Read*), mengubah (*Update*), dan menghapus (*Delete*) data yang berada dalam *database*.

2.3.4. Sublime Text

Sublime Text adalah teks *editor* untuk berbagai bahasa pemrograman termasuk pemrograman *PHP*. *Sublime Text* merupakan teks *editor* lintas platform dengan *Phyon Application Interface (API)*. *Sublime Text* juga mendukung bahasa pemrograman dan bahasa markup, dan fungsinya dapat ditambah dengan *plugin*.

2.3.5. Bootstrap

Dalam buku Jubilee Enterprise (2016:1) “*Bootstrap* adalah *framework front-end* yang intuitif dan *powerfull* untuk pengembangan aplikasi *web* yang lebih cepat dan mudah. *Bootstrap* menggunakan *HTML*, *CSS* dan *Javascript*”.

Bootstrap dikembangkan oleh Mark Otto dan Jacob Thorton dari *Twitter*. *Framework* ini diluncurkan sebagai produk *opensource*. *Bootstrap* memiliki fitur–fitur komponen *interface* yang bagus seperti *Typography*,

Forms, Buttons, Tables, Navigation, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel, dan lain sebagainya.

Salah satu kelebihan *Bootstrap* adalah *framework* ini berisi kumpulan *tool* yang gratis untuk membuat *layout web* yang fleksibel dan responsive. Berikut ini adalah kelebihan-kelebihan *bootstrap* lainnya :

1) Menghemat waktu

Menghemat waktu dan tenaga dengan menggunakan berbagai desain template dan kelas yang sudah ada dalam *bootstrap*.

2) Fitur yang *responsif*

Dengan *bootstrap*, dapat dengan mudah membuat desain yang *responsif*. Dengan fitur-fitur ini, halaman *web* akan tampil *responsif* pada perangkat yang berbeda tanpa perlu adanya perubahan kode *markup*.

3) Desain yang konsisten

Semua komponen *bootstrap* mempunyai desain *template* dan *style* yang sama sehingga membuat tampilan *web* menjadi konsisten.

4) Mudah digunakan

Bootstrap sangat mudah digunakan, siapapun dengan pengetahuan dasar HTML dan CSS dapat menggunakan *bootstrap*.

5) Didukung oleh semua *browser* populer

Bootstrap dapat digunakan pada semua *browser* modern seperti *Mozilla Firefox, Google Chrome, Safari, Internet Explorer*, dan *Opera*.

2.3.6. *Testing*

a. Pengertian *Testing*

Testing adalah sebuah proses yang diejawantahkan sebagai siklus hidup yang merupakan bagian dari proses rekayasa perangkat lunak secara terintegrasi demi memastikan kualitas dari perangkat lunak serta memenuhi kebutuhan teknis yang telah disepakati dari awal (Soetam Rizky Wicaksono, 2017:315).

Dalam Soetam Rizky Wicaksono (2017:318-320) detail dari tahapan yang harus dilampaui dalam kaitan kebutuhan perangkat lunak dari sudut pandang *testing* perangkat lunak adalah :

1) *Verifikasi*

Verifikasi adalah proses pemeriksaan untuk memastikan bahwa perangkat lunak telah menjalankan apa yang harus dilakukan dari kesepakatan awal antara pengembang perangkat lunak dan pengguna.

2) *Validasi*

Validasi adalah sebuah proses yang melakukan konfirmasi bahwa perangkat lunak dapat di eksekusi secara baik.

Definisi dari standar yang harus dipenuhi oleh kebutuhan perangkat lunak adalah pembebasan perangkat lunak dari *failure*, *fault*, dan *error* serta *incident* dijelaskan dalam detail berikut:

1) *Failure*

Failure adalah sesuatu yang terjadi jika sebuah perilaku diluar lingkup perangkat lunak tidak sesuai dengan kebutuhan perangkat lunak. *Failure* juga dapat didefinisikan sebagai penyimpangan dari fungsi perangkat lunak yang sesungguhnya.

2) *Fault*

Fault didefinisikan sebagai *the adjudged cause of an error* atau akibat yang dihasilkan oleh *error*. Dalam *behaviour chain* proses *testing*, *fault* adalah akar permasalahan dari kegagalan sebuah perangkat lunak. Dari sisi bahasa, *fault* atau kerusakan diartikan sebagai *something done wrongly* atau sesuatu yang dikerjakan dengan salah. *Fault* dalam lingkup *testing* perangkat lunak, bisa jadi tidak akan terdeteksi kecuali jika suatu tindakan akan membuatnya muncul dalam proses perangkat lunak. Ini berarti bahwa *fault* adalah potensi dari terjadinya sebuah *error*, dan saat *error* tersebut terjadi akibat tindakan pengguna, maka akan timbul *failure* atau kegagalan dalam proses perangkat lunak.

3) *Error*

Error adalah sebuah keadaan dari sistem yang disebabkan oleh tindakan pengguna yang pada akhirnya menyebabkan kegagalan dalam pelaksanaan fungsi sebuah perangkat lunak.

4) *Incident*

Incident atau kecelakaan merupakan hasil akhir yang terjadi akibat dari *error* berkelanjutan dan tidak diperbaiki atau tidak terdeteksi dalam proses pengembangan perangkat lunak.

b. Tipe dan Teknik *Testing*

Secara teoritis, *testing* dapat dilakukan dengan berbagai jenis tipe dan teknik. Namun, secara garis besar, terdapat dua jenis tipe *testing* yang paling umum digunakan di dalam lingkup rekayasa perangkat lunak. Dua jenis tersebut adalah *white box testing* dan *black box testing* (Soetam Rizky Wicaksono, 2017:347).

1) *White Box Testing*

White box testing secara umum merupakan jenis *testing* yang lebih berkonsentrasi terhadap “isi” dari perangkat lunak itu sendiri. Prinsip dari keluaran tipe *testing* ini adalah :

- a) Menjamin bahwa semua alur program yang *independen* (dalam bentuk modul, *form*, prosedur, *class*, dan lainnya) telah dites minimal satu kali.
- b) Telah melakukan *testing* terhadap semua kondisi percabangan dengan nilai *true* dan *false*.
- c) Telah melakukan *testing* terhadap semua jenis perulangan dengan kondisi normal dan kondisi yang dianggap melampaui batas perulangan (umumnya kondisi yang melampaui batas harus diatasi oleh prosedur tertentu).
- d) Telah melakukan *testing* terhadap struktur data internal (seperti variabel) agar terjaga validitasnya.

Beberapa teknik yang terdapat dalam jenis *white box testing* adalah :

a) *Decision (branch) Covarage*

Sesuai dengan namanya, teknik *testing* ini focus terhadap hasil dari tiap skenario yang dijalankan terhadap bagian perangkat lunak yang mengandung percabangan (if... then... else).

b) *Condition Coverage*

Teknik ini hampir mirip dengan teknik yang pertama, tetapi dijalankan terhadap percabangan yang dianggap kompleks atau percabangan majemuk

c) *Path Analysis*

Merupakan teknik *testing* yang berusaha menjalankan kondisi yang ada dalam perangkat lunak serta berusaha mengoreksi apakah kondisi yang dijalankan telah sesuai dengan alur diagram yang terdapat dalam proses perancangan.

d) *Execution Time*

Pada teknik ini perangkat lunak berusaha dijalankan atau dieksekusi kemudian dilakukan pengukuran waktu pada saat *input* dimasukan hingga *output* dikeluarkan. Waktu eksekusi yang dihasilkan kemudian dijadikan bahan evaluasi dan dianalisa lebih lanjut untuk melihat apakah perangkat lunak telah berjalan sesuai dengan kondisi yang dimaksud oleh *tester*.

e) *Algorithm Analysis*

Teknik ini umumnya jarang dilakukan jika perangkat lunak yang dibuat berjenis sistem informasi. Sebab teknik ini membutuhkan kemampuan matematis yang cukup tinggi dari para *tester*, karena didalamnya berusaha melakukan analisa terhadap algoritma yang diimplementasikan pada perangkat lunak tersebut.

2) *Black Box Testing*

Black box testing adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tetapi cukup dikenal proses *testing* dibagian luar.

Beberapa teknik *testing* yang tergolong dalam tipe ini antara lain:

a) *Equivalence Partitioning*

Pada teknik ini, tiap *inputan* data dikelompokkan ke dalam grup tertentu, yang kemudian dibandingkan *outputnya*.

b) *Boundary Value Analysis*

Merupakan teknik yang sangat umum digunakan pada saat awal sebuah perangkat lunak selesai dikerjakan. Pada teknik ini, dilakukan *inputan* yang melebihi dari batasan sebuah data.

c) *Cause Effect Graph*

Dalam teknik ini, dilakukan proses *testing* yang menghubungkan sebab dari sebuah *inputan* dan akibatnya pada *output* yang dihasilkan.

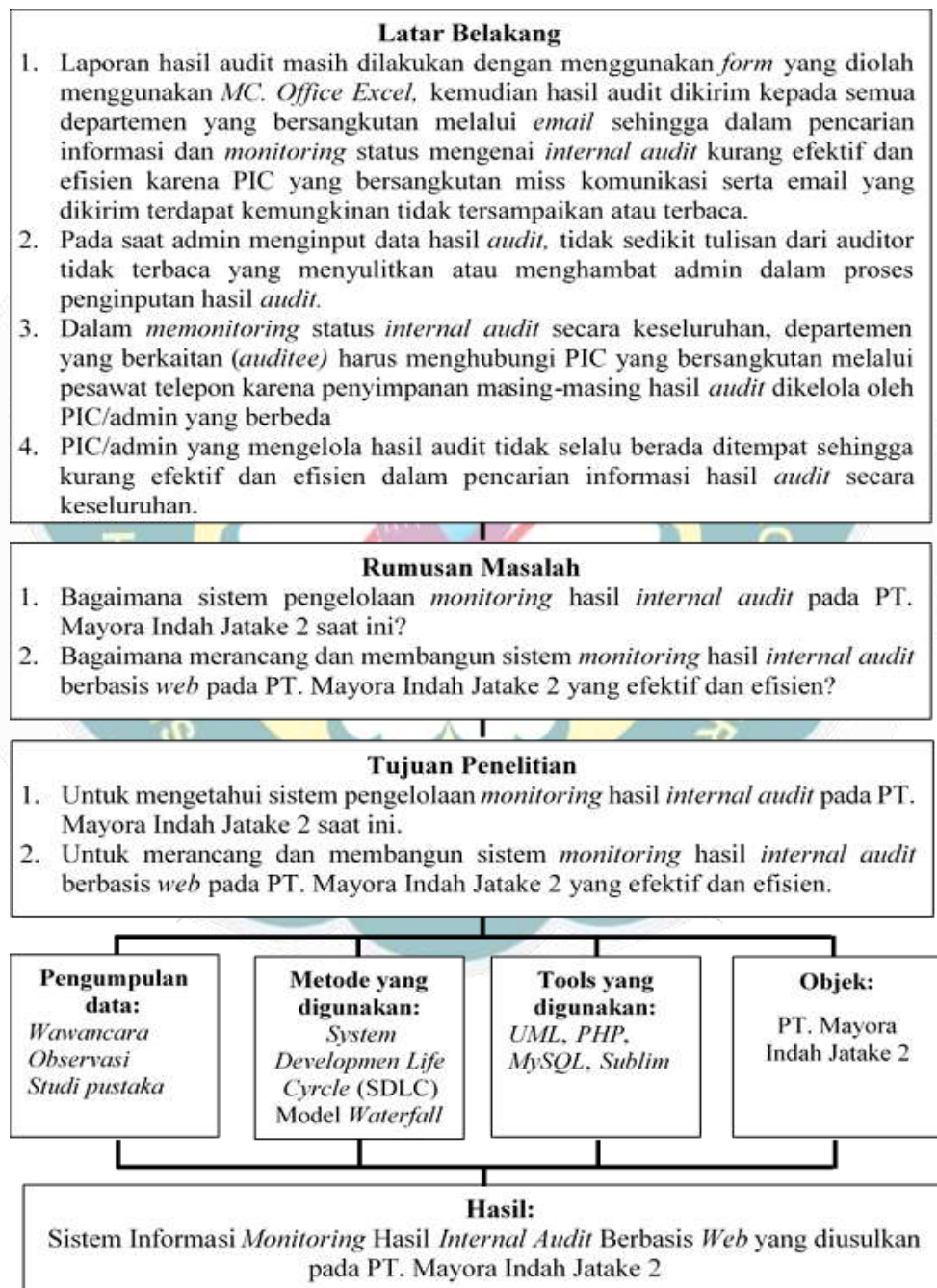
d) *Random Data Selection*

Teknik ini berusaha melakukan proses *inputan* data dengan menggunakan nilai acak. Dari hasil *inputan* tersebut kemudian dibuat sebuah tabel yang menyatakan validitas dari *output* yang dihasilkan.

e) *Feature Test*

Pada teknik ini, dilakukan proses *testing* terhadap spesifikasi dari perangkat lunak yang telah selesai dikerjakan

2.4. Kerangka Kerja Teoritis



GAMBAR 2.5. Kerangka Kerja Teoritis

Berikut merupakan uraian dari kerangka kerja teoritis :

1. PT. ABC merupakan salah satu industri makanan, permasalahan yang terdapat pada PT. ABC ini adalah pada saat melakukan internal audit masih menggunakan *form* atau *checklist* kemudian data hasil *audit* diolah menggunakan *office* standar seperti perangkat lunak *Mc. Office Excel*, kemudian hasil *audit* tersebut dikirim kepada semua departemen melalui *email* sehingga dalam pencarian informasi mengenai *internal audit* kurang efektif dan efisien karena *email* lebih rentan tertutup oleh *email* lainnya atau *email* tidak terbaca. Dalam permasalahan yang terjadi selanjutnya adalah pada saat admin melakukan *input* data hasil *audit*, tidak sedikit tulisan dari auditor tidak terbaca sehingga menyulitkan atau menghambat admin dalam proses penginputan hasil *audit*. Selanjutnya untuk informasi *internal audit* secara keseluruhan seperti *Audit Mutu Internal (AMI)*, *Good Manufacturing Practices (GMP)*, dan *Morning Round*, untuk melihat informasi tersebut departemen yang berkaitan harus menghubungi admin yang berbeda melalui pesawat telepon karena penyimpanan hasil *audit* tidak pada satu admin tetapi dua admin dan admin tidak selalu berada ditempat atau diruangan sehingga kurang efektif dan efisien dalam pencarian informasi hasil *audit* secara keseluruhan. Dengan kurangnya efektif dan efisien baik dalam proses *audit* maupun dalam menyajikan informasi mengenai hasil *audit* maka akan mempengaruhi mutu atau kualitas produk, mempengaruhi citra perusahaan, *break down system*, dan dapat mengurangi target produksi.

2. Pada tahap selanjutnya adalah merumuskan masalah berdasarkan dari latar belakang yang telah dijelaskan yaitu: bagaimana merancang dan membangun aplikasi sistem *internal audit* berbasis web pada PT. ABC dan bagaimana menyajikan laporan hasil *internal audit* secara efektif dan efisien.
3. Pada tahap selanjutnya adalah menentukan tujuan dari penelitian yang dilakukan, tujuan dari penelitian ini yaitu :
 - a. Untuk mengetahui sistem pengelolaan *monitoring* hasil *internal audit* pada PT. ABC saat ini.
 - b. Untuk merancang dan membangun sistem *monitoring* hasil *internal audit* berbasis *web* pada PT. ABC yang efektif dan efisien.
4. Tahap selanjutnya yaitu bagaimana cara membuat program aplikasi perancangan sistem informasi *monitoring* hasil *internal audit* agar terkomputerisasi dengan melakukan beberapa tahap sebagai berikut :
 - a) Pengumpulan data dengan metode observasi, wawancara serta studi pustaka. Kemudian dilakukan analisa terhadap data yang dikumpulkan untuk mendapatkan sebuah solusi atau perbaikan.
 - b) Menentukan metode perancangan sistem dengan menggunakan *System Development Life Cycle* (SDLC) dengan model *waterfall*.
 - c) Menentukan *tools* yang digunakan dalam proses pembuatan diagram dengan menggunakan bahasa pemodelan *Unified Modelling Language* (UML), bahasa pemrograman PHP, *database* menggunakan MySQL dan *text editor* menggunakan *sublime*.
 - d) Objek penelitian yaitu PT. ABC

5. Pada tahap akhir dari penelitian ini didapatkan solusi atau hasil dari masalah yang terjadi yaitu terbentuknya sistem informasi *internal audit* berbasis web.



BAB III

METODOLOGI PENELITIAN

3.1. Desain Penelitian

Dalam melakukan suatu penelitian diperlukan perencanaan penelitian agar penelitian yang dilakukan dapat berjalan dengan baik, sistematis, dan efektif

Dalam buku Endang Widi (2018:213) menerangkan bahwa “kadang-kadang peneliti tidak dapat membedakan arti “rencana” dengan “desain”. Rencana dalam tulisan ini berarti seperangkat kegiatan yang ditata secara sistematis dan runtut yang akan dilaksanakan oleh peneliti untuk mencapai tujuan penelitian”.

Langkah-langkah yang dilakukan untuk melakukan penelitian proses *internal audit* pada PT. ABC adalah sebagai berikut :

- a. Mencari dan membaca literature mengenai *internal audit*.
- b. Mempelajari proses *internal audit*, untuk mengetahui kekurangan dan kelebihan proses yang berjalan.
- c. Melakukan pengumpulan data yang berkaitan dengan *internal audit*
- d. Memberikan solusi agar hasil *internal audit* menjadi lebih mudah, efektif dan efisien.
- e. Membuat rancangan sistem informasi *internal audit* berbasis *web* pada PT. ABC.

Pada penelitian yang dilakukan oleh penulis langkah pertama adalah penulis melakukan observasi langsung ke lapangan kemudian melakukan wawancara dengan orang yang terkait guna untuk mengumpulkan data-data yang diperlukan untuk kemudian diolah dan dilakukan pembahasan sehingga tercapai kesimpulan dapat dibuat suatu laporan tentang kegiatan penelitian yang dilakukan pada PT. ABC.

3.2. Data dan Sumber Data

Data adalah bahan keterangan berupa himpunan fakta, angka, huruf, grafik, tabel, lambang, objek, kondisi, situasi. Data merupakan bahan baku informasi. Untuk mencapai tujuan penelitian, peneliti memerlukan data yang benar yang dapat diperoleh di lapangan sesuai dengan topik dalam penelitiannya. Dalam penelitian ini sumber data yang digunakan oleh penulis adalah data primer dan data sekunder.

1. Data primer adalah sumber data yang diperoleh peneliti secara langsung dari sumber asli dari lokasi penelitian. Dalam penelitian ini data primer diperoleh langsung dari tempat penelitian melalui observasi maupun melalui kegiatan tanya jawab yang langsung kepada *user* yang terkait mengenai sistem *internal audit* yang berjalan saat ini.
2. Data sekunder adalah data penelitian yang diperoleh secara tidak langsung melalui media perantara diperoleh dan dicatat oleh pihak lain. Data sekunder umumnya berupa bukti, catatan atau laporan yang telah tersusun dalam arsip.

a. Tempat Penelitian

Tempat penelitian adalah tempat dimana peneliti melakukan penelitian dalam mengumpulkan data. Dalam penelitian ini penulis melakukan penelitian *internal audit* pada PT. ABC yang beralamatkan Jl. Telesonik Blok C No. 2 Kec. Jatiuwung, Kab. Tangerang, Banten.

b. Waktu Penelitian

Waktu penelitian adalah waktu yang dibutuhkan penulis dalam melakukan penelitian untuk pengumpulan data. Dalam penelitian ini penulis melakukan penelitian pada saat bekerja dimulai dari bulan Agustus sampai dengan November 2019 di PT. ABC.

3.3. Metode Pengumpulan Data

Teknik pengumpulan data “merupakan langkah yang paling strategis dalam penelitian karena tujuan utama dari penelitian adalah mendapatkan data” (Endang Widi Winarti, 2018:158). Dalam melakukan penelitian ini penulis melakukan pengumpulan data dengan cara sebagai berikut :

3.3.1. Wawancara

Wawancara adalah suatu percakapan yang diarahkan pada suatu masalah tertentu dan merupakan proses Tanya jawab lisan dimana dua orang atau lebih berhadapan secara fisik (Setyadin dalam Imam Gunawan, 2017:160).

Wawancara yang dilakukan dalam penelitian ini yaitu tanya jawab antara peneliti dengan *user* terkait *internal audit* pada PT. ABC dengan

tujuan untuk menggali data atau informasi yang berhubungan dengan penelitian. Pertanyaan yang diajukan harus berkaitan erat dengan sistem yang akan dibuat agar dapat menghasilkan sistem yang lebih baik dari sistem sebelumnya.

3.3.2. Observasi

Observasi merupakan suatu teknik pengumpulan data yang dilakukan dengan cara mengadakan penelitian secara teliti, serta pencatatan secara sistematis (Arikunto dalam Imam Gunawan, 2017:143).

Observasi yang dilakukan dalam penelitian ini yaitu :

- a. Peneliti menetapkan terlebih dahulu apa saja yang ingin diobservasi atau diamati. Hal yang diamati dalam penelitian ini adalah proses *internal audit* dan *monitoring hasil internal audit* yang dilakukan pada PT. ABC
- b. Peneliti mencari data dengan datang langsung ke PT. ABC, kemudian peneliti melakukan pengamatan pada kebiasaan-kebiasaan yang dilakukan selama proses *audit dan monitoring hasil audit* berlangsung.

3.3.3. Studi Pustaka

Teknik studi pustaka adalah salah satu metode pengumpulan data yang digunakan dalam penelitian sosial untuk menelusuri data histori (Bungin dalam Imam Gunawan, 2017:177).

Dokumen merupakan catatan peristiwa yang sudah berlalu yang berbentuk tulisan, gambar, atau karya monumental dari seseorang (Sugiono dalam Imam Gunawan, 2017:176).

Dokumentasi yang dilakukan dalam penelitian ini yaitu berupa SOP (Standar Operasional Prosedur) yang terkait dengan *internal audit*.

3.4. Metode Analisis Data

Analisis data adalah proses pencarian dan pengaturan secara sistematis hasil wawancara, catatan-catatan, dan bahan-bahan yang dikumpulkan untuk meningkatkan pemahaman terhadap semua hal yang dikumpulkan dan memungkinkan menyajikan apa yang ditemukan (Bogdan & Biklen dalam Imam Gunawan, 2017:210).

Metode analisis data yang digunakan pada penelitian ini yaitu menggunakan metode kualitatif. Untuk menganalisa proses *internal audit* yang berjalan saat ini akan dilakukan tahapan sebagai berikut :

- a. Pengumpulan data hasil penelitian, pada tahap ini peneliti melakukan wawancara kepada *user* yang terkait, masalah yang sedang dihadapi serta pengumpulan dokumen-dokumen yang berkaitan dengan proses *internal audit* di PT. ABC
- b. Analisa terhadap data yang dikumpulkan, pada tahapan ini peneliti mempelajari serta menganalisa proses dokumentasi yang sedang berjalan untuk mengetahui gambaran keseluruhan sistem yang sedang berjalan.

- c. Identifikasi atas kebutuhan pengguna akan informasi, pada tahapan ini peneliti melakukan analisa kebutuhan sistem usulan sesuai dengan kebutuhan serta permasalahan yang berkaitan dengan proses *internal audit*.
- d. Identifikasi sistem *internal audit* yang akan dibangun beserta *spesifikasi hardware* yang dibutuhkan.

3.5. Metode Perancangan/Pengembangan Sistem

Adapun dalam penelitian ini metode yang akan digunakan adalah metode SDLC dengan menggunakan *fase waterfall* atau sering juga disebut model sekuensial *linier* atau alur hidup klasik. *Fase* ini dipakai karena menyediakan pendekatan alur hidup perangkat lunak sekuensial atau terurut dari analisa, desain, pengodean, *testing*, *implementasi* dan tahap pendukung (*support*).

a. Analisa

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini menranslasi kebutuhan perangkat lunak dari tahap

analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

c. Pengkodean

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. *Testing*

Pengujian atau *testing* fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. *Implementasi*

Tahapan ini dapat dikatakan final dalam pembuatan sebuah sistem, setelah melakukan analisa, desain, pengodean maka sistem yang telah selesai akan digunakan oleh *user*.

f. Pendukung (*support*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian, atau perangkat lunak harus beradaptasi terhadap lingkungan baru, pada tahap ini diharapkan dapat mengulangi proses pengembangan mulai dari analisa spesifikasi untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak yang baru.

3.5.1. Tahap Analisis Kebutuhan Sistem

Analisa kebutuhan sebuah perangkat lunak atau *requirement analysis* adalah “proses untuk mempelajari kebutuhan pengguna yang datang pada defines dari sistem, perangkat keras serta kebutuhan perangkat lunak” (Soetam Rizky Wicaksono 2017:155).

Dalam metode analisis kebutuhan data penulis menggunakan teknik analisis data deskriptif. Untuk proses analisis data dalam penelitian ini, penulis melakukan analisa sistem yang berjalan, menjelaskan mengenai hasil dari evaluasi sistem yang sudah ada dan mengembangkan sistem yang belum ada sehingga menghasilkan sistem baru yang lebih efektif.

a. *Data Reduction* (Reduksi Data)

Reduksi data dapat diartikan sebagai suatu proses merangkum, memilih hal-hal yang pokok, memfokuskan pada hal-hal yang penting, mencari tema, dan polanya. “Dalam mereduksi data, setiap peneliti akan dipandu oleh tujuan yang akan dicapai” (Endang Widi Winarti, 2018:172).

b. *Data Display* (Penyajian Data)

Dalam penelitian kualitatif, penyajian data bias dilakukan dalam bentuk uraian singkat, bagan, hubungan antarkategori, *flowchart*, dan sejenisnya. “Data yang paling sering digunakan untuk menyajikan

data dalam penelitian kualitatif adalah data berupa teks yang bersifat naratif” (Endang Widi Winarti, 2018:173).

c. *Conclusion Drawing/verification* (Penarikan Kesimpulan dan Verifikasi)

Kesimpulan awal yang dikemukakan masih bersifat sementara sehingga akan berubah bila tidak ditemukan bukti-bukti yang kuat untuk mendukung tahap pengumpulan data berikutnya. “Apabila kesimpulan yang dikemukakan pada tahap awal didukung oleh bukti-bukti yang valid dan konsisten, saat peneliti kembali ke lapangan untuk mengumpulkan data maka kesimpulan yang dikemukakan bersifat kredibel” (Endang Widi Winarti, 2018:174).

3.5.2. Desain Diagram yang digunakan

Pembuatan diagram merupakan implementasi dari hasil analisa kebutuhan sistem, pada tahap ini penulis memodelkan kebutuhan sistem kedalam bentuk diagram menggunakan bahasa pemodelan *Unified Modelling Language* (UML). Adapun diagram yang digunakan adalah sebagai berikut :

a. *Use Case Diagram*

Use case diagram atau diagram *use case* merupakan permodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor

dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Komponen-komponen yang terdapat pada *use case* adalah :

1. *Use Case*

Merupakan fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit-unit atau aktor

2. *Actor*

Merupakan Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor

3. *Asosiasi/association*

Merupakan Komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor

4. *Ekstensi/extend*

Merupakan relasi *use case* tambahan kesebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu, mirip dengan prinsip *inheritance* pada

pemrograman berorientasi objek, biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan

5. Generalisasi/*generalization*

Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya

6. Menggunakan / *include* / *uses*

Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan atau sebagai syarat dijalankan *use case* ini.

3.5.3. **Software yang digunakan**

Software adalah suatu intruksi yang berfungsi untuk menjalankan *hardware*, mengelola data sehingga dapat menghasilkan informasi yang bermanfaat dalam proses pengambilan keputusan. Dalam hal ini *software* yang dibutuhkan antara lain :

1. *Operation System* : *Microsoft Windows*
2. *Program Aplikasi* : *PHP Laravel*
3. *Database server* : *MySQL*
4. *Scrip Editor* : *Sublime Text*

3.5.4. **Metode Pengujian Sistem yang digunakan**

Metode *testing* yang digunakan oleh penulis dalam penelitian ini adalah metode *black box testing*.

Black box testing adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester* memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya, tetapi cukup dikenal proses *testing* dibagian luar (Soetam Rizky Wicaksono, 2017:353).

Penulis menggunakan metode ini untuk mengetes program yang merupakan kegiatan pengujian sistem yang diusulkan pada tahap ini penulis melakukan pengujian terhadap sistem yang telah dibangun sebagai usulan perbaikan sistem yang ada.



BAB 4

ANALISA DAN PERANCANGAN SISTEM

4.1. Analisa Sistem Berjalan

4.1.1. Latar Belakang Perusahaan

Berdirinya PT. ABC diawali dari produksi skala *home industry* pada tahun 1948 yang memproduksi biskuit dengan cita rasa yang khas dan banyak disukai pelanggan, sehingga permintaan pelanggan semakin bertambah.

4.1.2. Visi dan Misi Perusahaan

PT. ABC memiliki visi dan misi sebagai berikut :

- a. Menjadi produsen makanan dan minuman yang berkualitas dan terpercaya dimata konsumen domestik maupun internasional dan menguasai pangsa pasar terbesar dalam kategori produk sejenis.
- b. Dapat memperoleh laba bersih operasional diatas rata-rata industri dan memberikan *value added* yang baik seluruh *stakeholders* perseroan.
- c. Dapat memberikan kontribusi positif terhadap lingkungan dan negara dimana perseroan berada.

4.1.3. Struktur PT. ABC



Sumber : Dokumen Struktur Organisasi PT. ABC

PT. ABC memiliki struktur organisasi yang memiliki ciri kekuasaan dan tanggung jawab mengalir dalam satu garis puncak ke bawah. Pimpinan pada anak perusahaan PT. ABC dipimpin oleh seorang *Factory Manajer*. Posisi dan kewenangan masing-masing jabatan di PT. ABC diuraikan sebagai berikut :

a. *Factory Manajer*

Factory Manajer (FM) adalah pimpinan yang bertanggung jawab dalam mengatur operasional suatu *plant* perusahaan agar berjalan dengan baik.

b. *Group Departement Head*

Group Departement Head (GDH) memiliki fungsi dasar dalam membantu menjalankan kegiatan perusahaan, merencanakan dan mengendalikan proses produksi dan kualitas produk untuk mencapai target produksi yang telah ditetapkan. *Group Departement Head* di PT. ABC membawahi langsung 4 *departement head* yang bertanggung jawab terhadap masing-masing departemennya, yaitu *Production Packing Departement Head*, *Production Process Departement Head*, *Engineering Departement Head* dan *Energy, Project dan Utility Departemen Head*.

c. *Continous Improvement Departement Head*

Continous Improvement (CI) *Departement Head* memiliki fungsi dasar dalam melakukan perbaikan terus menerus dan berkelanjutan untuk seluruh aspek-aspek produksi dan bertanggung jawab terhadap

pengendalian, pengawasan, perencanaan, maupun perbaikan terhadap seluruh teknik baik mesin, kebutuhan daya, lingkungan maupun energi dalam industri.

d. *PDQC Departement Head*

Product Development and Quality Control (PDQC) Departement Head

memiliki fungsi dasar dalam memastikan dan mengevaluasi seluruh kegiatan proses produksi berdasarkan kebijakan sistem keamanan pangan yang telah ditentukan dan menjamin keamanan serta kualitas produk yang dihasilkan sesuai standar yang berlaku dan efisien.

e. *Production Planning Inventory Control Departement Head*

Production Planning Inventory Control (PPIC) Departement Head

memiliki fungsi dasar dalam mengkoordinasikan dan mengendalikan rencana produksi terkait dengan jumlah produk, pengorganisasian sarana produksi, hingga tenaga kerja yang ada mulai dari persiapan bahan baku, kemasan hingga jumlah produk yang dibutuhkan. Selain itu departemen ini bertugas dalam mengkoordinasikan produksi kebutuhan bahan baik lokal maupun impor, melayani dan mengatur ketentuan produksi untuk ekspor sesuai keinginan eksportir, mengkoordinasikan pembuatan rencana dan realisasi produksi baik harian, mingguan, maupun bulanan, serta mengendalikan standar *output*, kapasitas dan parameter hasil produksi.

f. *Industrial Relation General Affair Departement Head*

Industrial Relation General Affair (IRGA) Departement Head

memiliki fungsi dasar dalam mengendalikan peraturan perundang-undangan ketenagakerjaan yang berlaku didalam perusahaan dan mengendalikan sarana umum diperusahaan.

g. *Production Process Departemen Head*

Production Process Departemen Head memiliki fungsi dasar dalam membantu menjalankan kegiatan perusahaan, merencanakan dan mengendalikan proses produksi dan kualitas produk untuk mencapai target produksi yang telah ditetapkan dimulai pada saat proses preparasi bahan baku hingga proses pemanggangan.

h. *Production Packing Departement Head*

Production Packing Departement Head memiliki fungsi dasar dalam membantu menjalankan kegiatan perusahaan, merencanakan dan mengendalikan proses produksi dan kualitas produk untuk mencapai target produksi yang telah ditetapkan dimulai pada saat proses pendinginan dengan *cooling conveyor* hingga proses *finish good* atau produk jadi.

i. *Engineering Departement Head*

Engineering Departement Head memiliki fungsi dasar dalam pengembangan dan perencanaan mesin, peralatan, dan teknik manufaktur suku cadang, perencanaan dan pengendalian pelaksanaan program pemeliharaan mesin produksi, perencanaan dan pengendalian

biaya pemeliharaan teknis, peningkatan produktivitas mesin-mesin produksi, peningkatan kemampuan SDM, pelaksanaan *improvement* mesin-mesin dan sistem kerja mesin.

j. *Energy, Project dan Utility Departement Head*

Energy, Project dan Utility Departement Head memiliki fungsi dasar dalam melakukan pengolahan pengadaan barang melalui perencanaan secara sistematis dan terkontrol, melakukan seleksi rekanan pengadaan sesuai kriteria perusahaan, dan bekerjasama dengan departemen terkait untuk memastikan kelancaran operasional perusahaan terkait tentang sumber energi dan sumber daya buatan yang digunakan untuk kelancaran proses produksi.

k. *Warehouse Departement Head*

Warehouse Departement Head bertanggung jawab terhadap keseluruhan pergudangan yang terdapat dalam industri. Departemen ini terbagi menjadi beberapa bagian diantaranya :

- 1) Gudang *Raw Material* (GDRM) berperan dalam menangani segala bahan baku yang digunakan saat proses produksi.
- 2) Gudang *Packaging Material* (GDPM) berperan menangani segala macam bahan pengemas produk.
- 3) Gudang *Finish Good* (GDFG) berperan dalam menangani produk-produk jadi yang telah siap dipasarkan.
- 4) Gudang *Transit Packing* (GDTP) berperan dalam menangani barang pengemas yang akan digunakan oleh produksi.

5) Gudang *Spare Part* (GDSP) berperan dalam menangani barang-barang *spare part* atau suku cadang.

1. *Quality Control Departement Head*

1) *Quality Control (QC) Departement Head* memiliki fungsi dasar yaitu memantau proses produksi, membantu sistem yang berperan dalam mengendalikan dan mengontrol kualitas dari produk yang dihasilkan sesuai dengan standar yang telah ditetapkan, analisa organoleptik dan laboratorium, serta berkoordinasi dengan bagian produksi dan teknik untuk menghasilkan *output* produk yang berkualitas.

2) *Quality System (QS)* memiliki fungsi dasar yaitu mengkoordinasikan kelancaran sistem manajemen keamanan pangan dan halal selain itu bertanggung jawab dalam memastikan sistem manajemen mutu dan keamanan pangan, halal, GMP, dan 5R dijalankan secara efektif, terkontrol, *terupdate*, dan tersosialisasi sesuai dengan standar dan ketentuan yang telah ditetapkan oleh perusahaan.

m. *Purchasing Lokal Section Head*

Purchasing Lokal Section Head memiliki fungsi dasar dalam melaksanakan pembelian barang dan jasa di lingkungan pabrik, memastikan kedatangan *spare part* sesuai dengan target, dan memastikan kebenaran harga dan jenis barang yang dibeli harus sesuai permintaan.

n. *Factory Accounting Departement Head*

Factory Accounting (FA) Departement Head memiliki fungsi dasar dalam seluruh aspek *financial* dan *accounting* di pabrik khususnya pemantauan biaya produksi, melakukan pemantauan dan mengatur proses *stock opname* bahan baku dan bungkus secara berkala, berkoordinasi dengan pihak lain yang terlibat dalam proses *manufactur* terkait dengan pengendalian biaya produksi.

o. *Kasir Staff*

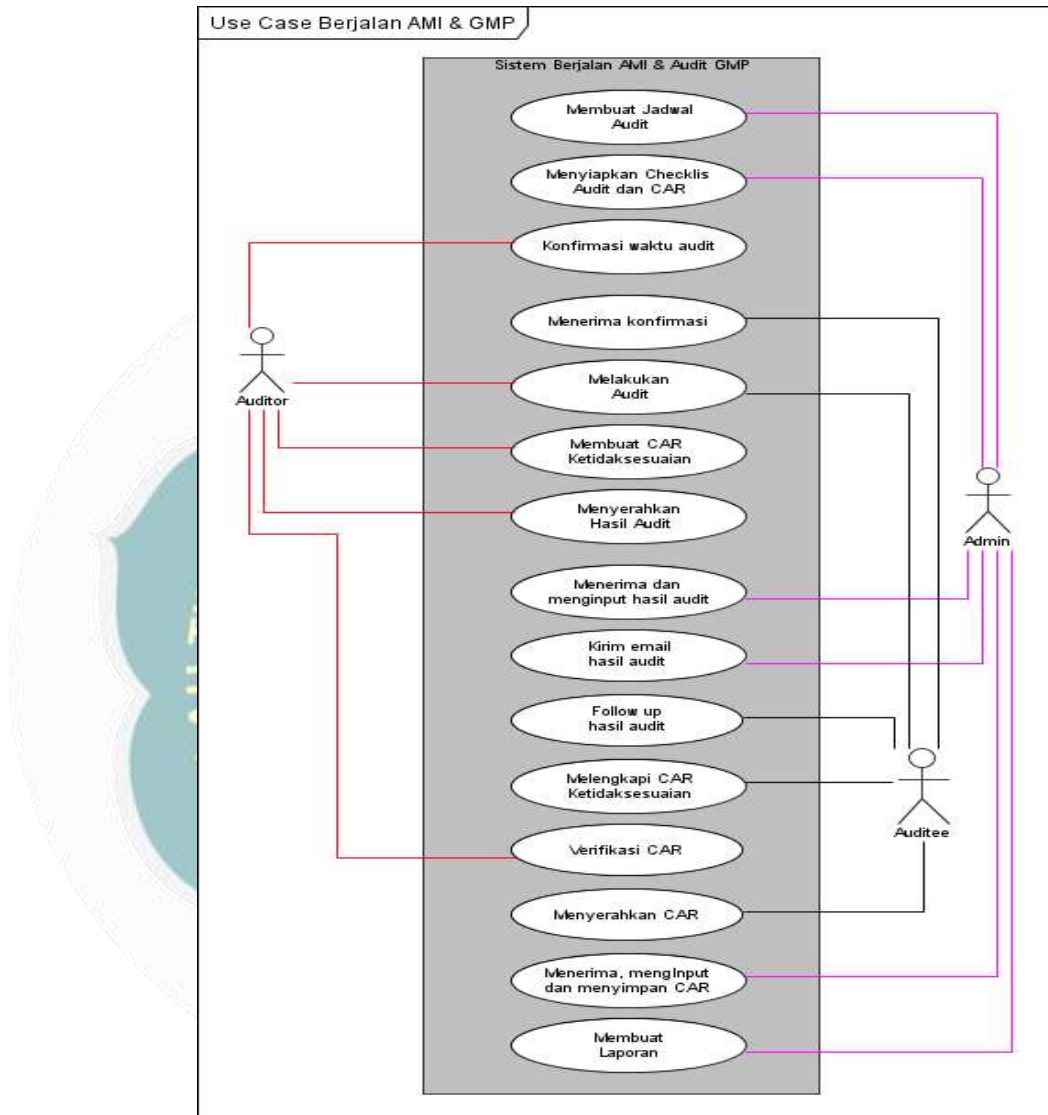
Kasir Staff bertanggung jawab atas fisik uang sehari-hari dan bertanggung jawab atas tukar faktur untuk *vendor* kas kecil pabrik, mengontrol kas bon dan membuat laporan keluar masuknya uang.

p. *Human Resource Development Departement Head*

Human Resource Developmen (HRD) Development Departement Head berperan dalam mengatur masalah administrasi yang berkaitan dengan sumber daya manusia seperti perekrutan dan seleksi karyawan, pengembangan dan evaluasi karyawan, mengkoordinasikan fungsi *recruitment* dan *training*, mengkoordinasikan fungsi pengembangan *talented people*, dan mengeksekusi program-program HR *Corporate*. Berperan dalam pengendalian jumlah tenaga kerja sesuai dengan kebutuhan industri.

4.1.4. Tata Laksana Sistem yang Berjalan pada *Audit Mutu Internal (AMI)* dan *Audit Good Manufacturing Praticice (GMP)*

a. *Use Case Diagram* Sistem Berjalan



GAMBAR 4.2. *Use Case Diagram* Sistem Berjalan pada *Audit Mutu Internal (AMI)* dan *Audit GMP*

b. Deskripsi Aktor

Berikut ini adalah deskripsi aktor berjalan pada *Audit Mutu Internal (AMI)* dan *Audit Good Manufacturing Praticice (GMP)* pada PT. ABC

TABEL 4.1. Deskripsi Aktor Berjalan pada *Audit Mutu Internal* dan *Audit GMP*

NO	Aktor	Deskripsi
1	Admin	Bertugas membuat jadwal periode <i>audit</i> , menyiapkan <i>checklist audit</i> , <i>form CAR</i> (<i>Corective Action Report</i>), dan menginput hasil <i>audit</i> dan <i>CAR</i> serta membuat laporan hasil <i>audit</i> .
2	<i>Auditor</i>	Bertugas melakukan <i>audit</i> , membuat hasil <i>audit</i> , menuliskan hasil ketidaksesuaian pada <i>form CAR</i> (<i>Corective Action Report</i>), menyerahkan hasil <i>audit</i> kepada admin, memberikan <i>form CAR</i> kepada <i>auditee</i> dan memverifikasi tindakan perbaikan yang dilakukan oleh <i>auditee</i> .
3	<i>Auditee</i>	Bertugas melakukan <i>audit</i> bersama <i>auditor</i> , melengkapi <i>form CAR</i> (<i>Corective Action Report</i>) yang dibuat oleh <i>auditor</i> terkait <i>root cause</i> dan tindakan perbaikan atas ketidaksesuaian yang ditemukan, kemudian menyerahkan <i>CAR</i> kepada <i>auditor</i> untuk diverifikasi dan selanjutnya <i>auditee</i> menyerahkan <i>CAR</i> tersebut kepada admin.

c. Deskripsi *Use Case*

Berikut ini adalah deskripsi *Use Case* berjalan pada *Audit Mutu Internal* (AMI) dan *Audit GMP* pada PT. ABC

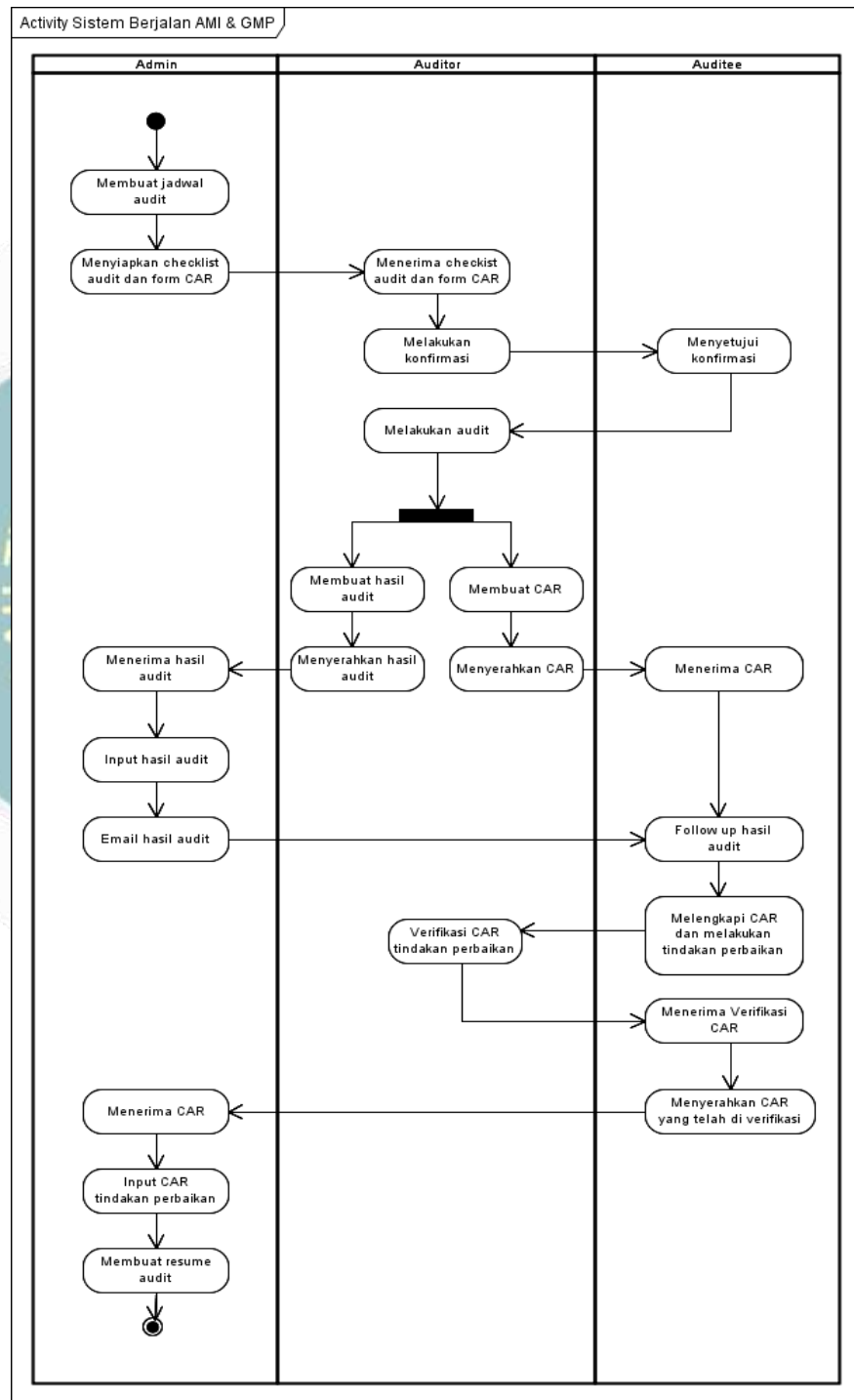
TABEL 4.2. Deskripsi *Use Case* Berjalan pada *Audit Mutu Internal* dan *Audit GMP*

NO	<i>Use Case</i>	Deskripsi
1	Membuat jadwal periode <i>audit</i>	Admin melakukan pembuatan jadwal periode <i>audit</i> .
2	Menyiapkan <i>checklist audit</i> & <i>form CAR</i> (<i>Corrective Action Report</i>)	Admin menyiapkan <i>checklist audit</i> dan <i>form CAR</i> (<i>Corrective Action Report</i>) untuk diberikan kepada <i>auditor</i> .
3	Konfirmasi waktu <i>audit</i>	<i>Auditor</i> melakukan konfirmasi kepada <i>auditee</i> sesuai dengan

		departemen yang akan di-audit terkait waktu pelaksanaan <i>audit</i> yang akan dilakukan.
4	Menerima konfirmasi	<i>Auditee</i> mengkonfirmasi waktu pelaksanaan <i>audit</i> .
5	Melakukan <i>audit</i>	<i>Auditor</i> bersama <i>auditee</i> melakukan <i>audit</i> sesuai dengan waktu yang telah disepakati antara <i>auditor</i> dengan <i>auditee</i> .
6	Membuat CAR (<i>Corrective Action Report</i>) ketidaksesuaian	<i>Auditor</i> membuat CAR (<i>Corrective Action Report</i>) ketidaksesuaian dan kemudian memberikan CAR (<i>Corrective Action Report</i>) kepada <i>auditee</i> untuk dilengkapi terkait tindakan perbaikannya.
7	Menyerahkan hasil <i>audit</i>	<i>Auditor</i> menyerahkan hasil <i>audit</i> kepada admin.
8	Menerima dan melakukan <i>input</i> hasil <i>audit</i>	Admin melakukan <i>input</i> hasil <i>audit</i> pada <i>Ms. Excel</i> .
9	Kirim <i>email</i> hasil <i>audit</i>	Setelah hasil <i>audit</i> di- <i>input</i> kemudian admin mengirimkan hasil <i>audit</i> melalui <i>email</i> kepada semua departemen.
10	<i>Follow up</i> hasil <i>audit</i>	Setelah admin mengirimkan <i>email</i> kemudian <i>auditee</i> mem- <i>follow up</i> terkait hasil temuan dimasing-masing departemen.
11	Melengkapi CAR (<i>Corrective Action Report</i>) ketidaksesuaian	<i>Auditee</i> melengkapi <i>form</i> CAR (<i>Corrective Action Report</i>) yang telah diberikan oleh <i>auditor</i> terkait tindakan perbaikan yang akan dilakukan.
12	Verifikasi CAR (<i>Corrective Action Report</i>)	<i>Auditor</i> memverifikasi tindakan perbaikan yang dilakukan oleh <i>auditee</i> .
13	Menyerahkan CAR (<i>Corrective Action Report</i>)	Jika <i>form</i> CAR (<i>Corrective Action Report</i>) sudah di-verifikasi oleh <i>auditor</i> , maka <i>auditee</i> akan menyerahkan <i>form</i> CAR (<i>Corrective Action Report</i>) kepada admin.
14	Menerima dan melakukan <i>input form</i>	Admin melakukan <i>input form</i> CAR (<i>Corrective Action Report</i>).

	CAR (<i>Corrective Action Report</i>)	
15	Membuat resume hasil audit	Admin membuat resume hasil audit.

d. Activity Diagram Sistem Berjalan



GAMBAR 4.3. *Activity Diagram* sistem Berjalan pada *Audit Mutu Internal* dan *Audit GMP*

Keterangan gambar :

1. Mulai.
2. Admin membuat jadwal *audit*.
3. Admin menyiapkan *checklist audit* dan *form CAR (Corrective Action Report)*.
4. Auditor melakukan konfirmasi kepada *auditee* terkait waktu pelaksanaan *audit* yang akan dilakukan.
5. *Auditee* menerima dan menyetujui konfirmasi waktu pelaksanaan *audit*.
6. *Auditor* dan *auditee* melakukan proses *audit* sesuai dengan waktu yang telah disepakati.
7. *Auditor* membuat hasil *audit* dan menyerahkannya kepada admin, *auditor* juga membuat *CAR* dan menyerahkannya kepada *auditee*.
8. Admin menerima hasil *audit* dari *auditor* dan kemudian melakukan *input* hasil *audit* pada komputer.
9. Admin mengirim hasil *audit* kepada semua departemen melalui *email*.
10. Setelah admin mengirim *email* kemudian *auditee* melakukan *follow up* hasil *audit* terkait ketidaksesuaian yang ditemukan dimasing-masing departemen.

11. *Auditee* menerima CAR dari *auditor* untuk dilengkapi *root cause*, tindakan perbaikan dan melakukan perbaikan terhadap ketidaksesuaian yang ditemukan oleh *auditor* pada saat proses *audit*.

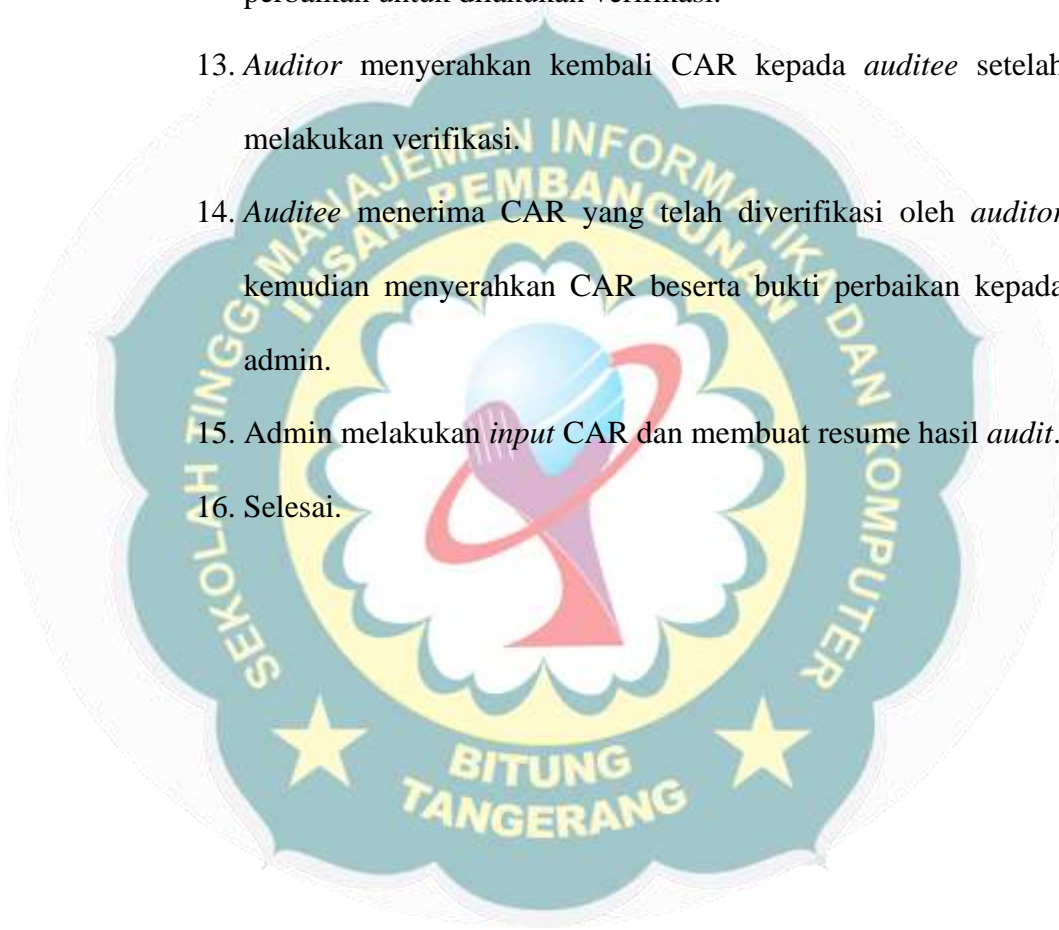
12. *Auditee* menyerahkan CAR kepada *auditor* beserta bukti perbaikan untuk dilakukan verifikasi.

13. *Auditor* menyerahkan kembali CAR kepada *auditee* setelah melakukan verifikasi.

14. *Auditee* menerima CAR yang telah diverifikasi oleh *auditor* kemudian menyerahkan CAR beserta bukti perbaikan kepada admin.

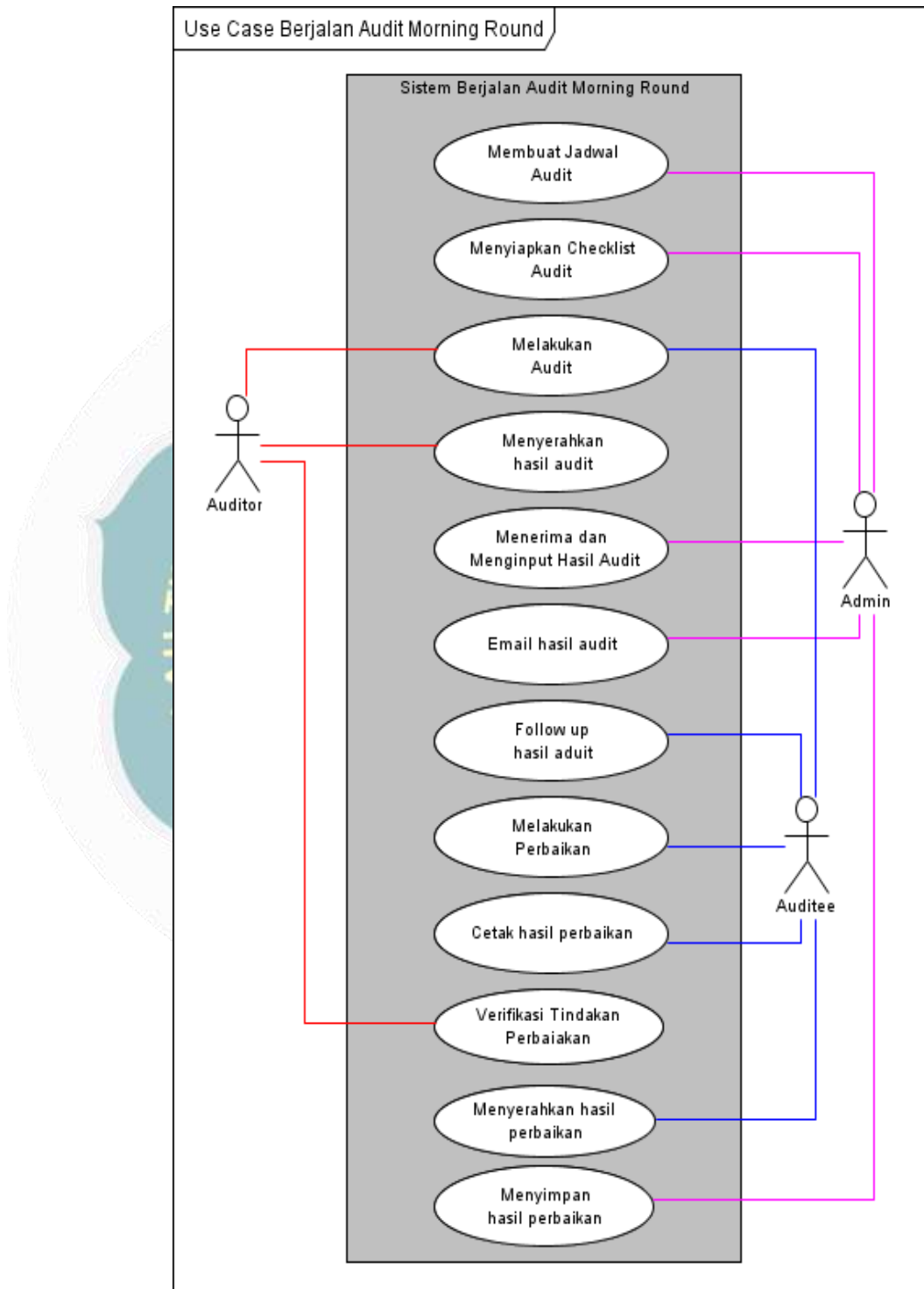
15. Admin melakukan *input* CAR dan membuat resume hasil *audit*.

16. Selesai.



4.1.5. Tata Laksana Sistem yang Berjalan pada *Audit Morning Round*

a. *Use Case Diagram* Sistem Berjalan



GAMBAR 4.4. *Use Case Diagram* Sistem Berjalan pada *Audit Morning Round*

b. Deskripsi Aktor

Berikut ini adalah deskripsi aktor sistem berjalan pada *Audit*

Morning Round PT. ABC

TABEL 4.3. Deskripsi Aktor Sistem Berjalan pada *Audit Morning Round*

NO	Aktor	Deskripsi
1	Admin	Bertugas membuat jadwal <i>audit</i> , menyiapkan <i>checklist audit</i> , melakukan <i>input</i> hasil <i>audit</i> , menyimpan hasil <i>audit</i> yang telah ditandatangani.
2	<i>Auditor</i>	Bertugas melakukan <i>audit</i> , membuat hasil <i>audit</i> , menyerahkan hasil <i>audit</i> kepada admin untuk dilakukan <i>input</i> .
3	<i>Auditee</i>	Bertugas melakukan proses <i>audit</i> bersama <i>auditor</i> , melengkapi hasil <i>audit</i> yang diinput oleh admin terkait tindakan perbaikan, mencetak hasil perbaikan untuk diverifikasi oleh <i>auditor</i> dan ditandatangani kemudian menyerahkan hasil <i>audit</i> yang telah diverifikasi kepada admin untuk disimpan oleh admin.

c. Deskripsi *Use Case*

Berikut ini adalah deskripsi *Use Case* sistem berjalan *Audit*

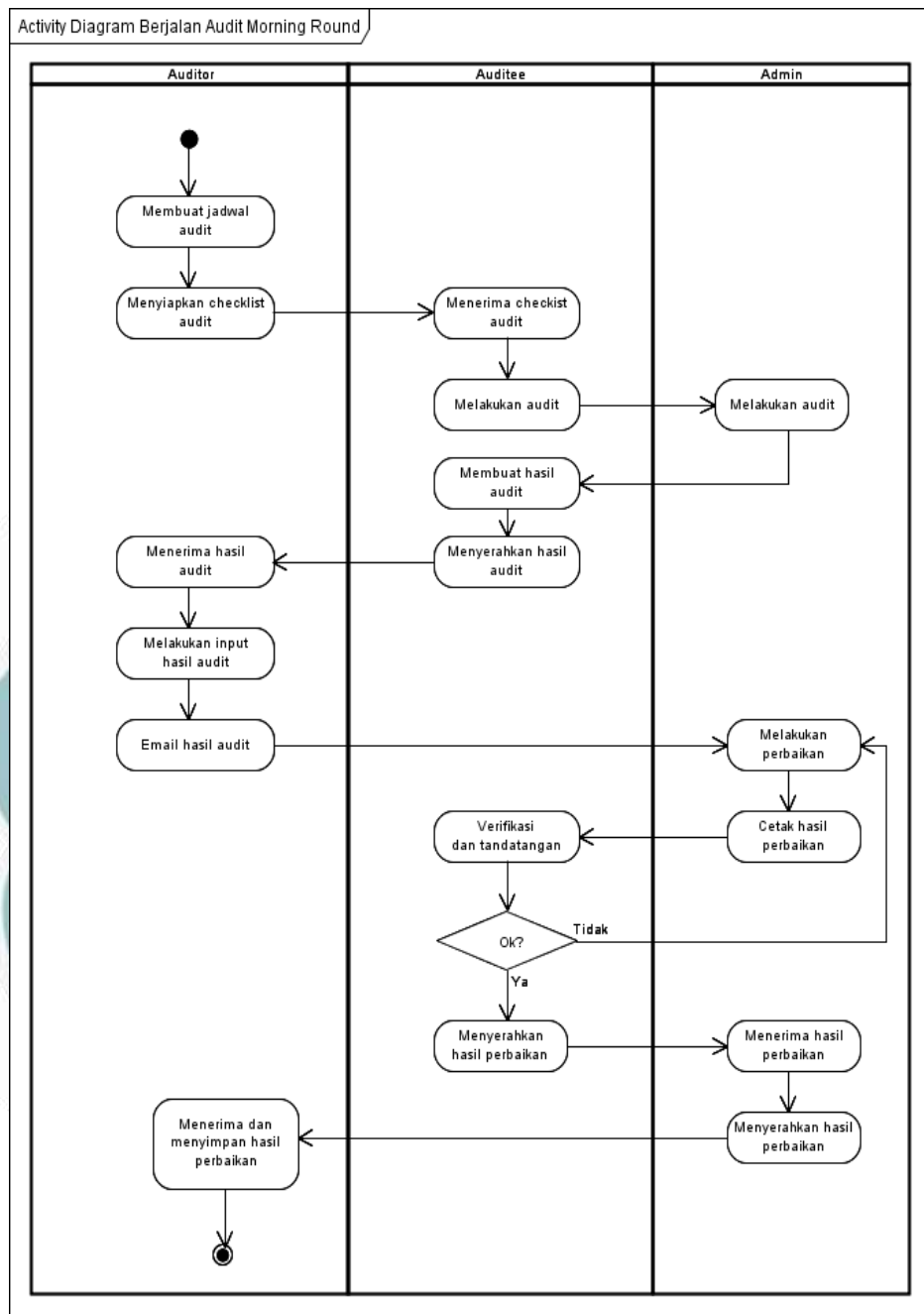
Morning Round pada PT. ABC

TABEL 4.4. Deskripsi *Use Case Diagram* Sistem Berjalan pada *Audit Morning Round*

NO	<i>Use Case</i>	Deskripsi
1	Membuat jadwal audit	Admin melakukan pembuatan jadwal audit <i>morning round</i> .
2	Menyiapkan <i>checklist audit</i>	Admin menyiapkan <i>checklist audit</i> untuk diberikan kepada <i>auditor</i> .
3	Melakukan <i>audit</i>	<i>Auditor</i> melakukan <i>audit</i> Bersama <i>auditee</i> sesuai dengan jadwal yang telah dibuat oleh admin.

4	Menyerahkan hasil <i>audit</i>	<i>Auditor</i> menyerahkan hasil <i>audit</i> kepada admin.
5	Menerima dan melakukan <i>input</i> hasil <i>audit</i>	Admin melakukan <i>input</i> hasil <i>audit</i> pada <i>Ms. Excel</i>
6	Kirim <i>email</i> hasil <i>audit</i>	Setelah hasil <i>audit</i> di- <i>input</i> kemudian admin mengirimkan hasil <i>audit</i> melalui <i>email</i> kepada departemen atau area yang di- <i>audit</i> .
7	Melakukan perbaikan	<i>Auditee</i> melengkapi resume hasil <i>audit</i> yang d- <i>input</i> oleh admin dengan melakukan tindakan perbaikan dan mencantumkan foto bukti perbaikan yang dilakukan.
8	Cetak hasil perbaikan	<i>Auditee</i> mencetak resume hasil <i>audit</i> yang telah dilengkapi dengan tindakan perbaikan untuk diverifikasi oleh <i>auditor</i> dan ditandatangani.
9	Verifikasi perbaikan	<i>Auditor</i> memverifikasi tindakan perbaikan yang dilakukan oleh <i>auditee</i> kemudian <i>auditor</i> menandatangani hasil <i>audit</i> yang telah dilengkapi dengan bukti perbaikan
10	Menyerahkan hasil perbaikan	<i>Auditee</i> menyerahkan hasil perbaikan yang telah diverifikasi oleh <i>auditor</i> kepada admin
11	Menyimpan hasil <i>audit</i>	Admin menerima dan menyimpan hasil <i>audit</i> yang telah dilengkapi dengan tindakan perbaikan dan sudah diverifikasi oleh auditor.

d. Activity Diagram



GAMBAR 4.5. Activity Diagram Berjalan pada Audit Morning Round

Keterangan gambar :

1. Mulai.
2. Admin membuat jadwal *audit*.

3. Admin menyiapkan *checklist audit*.
4. *Auditor* melakukan *audit* bersama *auditee* sesuai dengan jadwal yang telah dibuat oleh admin.
5. *Auditor* membuat hasil *audit* dan menyerahkannya kepada admin.
6. Admin menerima hasil *audit* dari *auditor* dan kemudian hasil *audit* di-input pada *komputer*.
7. Admin mengirimkan *email* hasil *audit* kepada departemen yang di-*audit*.
8. Setelah admin mengirim *email* kemudian *auditee* mem-follow up hasil *audit* terkait ketidaksesuaian yang ditemukan.
9. *Auditee* melengkapi *resume* hasil *audit* yang di-input oleh admin dengan melengkapi tindakan perbaikan dan mencantumkan foto bukti perbaikan.
10. *Auditee* mencetak *resume* hasil *audit* yang telah dilengkapi dengan tindakan perbaikan untuk diverifikasi oleh *auditor* dan ditandatangani.
11. *Auditor* memverifikasi tindakan perbaikan jika hasil verifikasi ok maka *auditor* menandatangani *resume* hasil *audit* yang telah dilengkapi dengan bukti perbaikan dan jika hasil verifikasi tidak ok maka *auditee* kembali melakukan perbaikan dan mencetak *resume* hasil perbaikan kemudian *auditor* memverifikasi ulang.
12. *Auditee* menyerahkan hasil *audit* yang telah ditandatangani oleh *auditor* kepada admin.

13. Admin menyimpan hasil perbaikan yang diberikan oleh *auditee*.

14. Selesai.

4.1.6. Masalah yang dihadapi

Adapun beberapa masalah yang sering dihadapi pada sistem *Internal Audit* PT. ABC antara lain :

1. Bagaimana cara merancang sistem informasi *monitoring* hasil *internal audit berbasis web* pada PT. ABC, karena sistem *audit* pada PT. ABC, karena laporan hasil *audit* masih dilakukan dengan menggunakan *form* yang diolah menggunakan *MS. Office Excel*, kemudian hasil *audit* dikirim kepada semua departemen yang bersangkutan melalui *email* sehingga dalam pencarian informasi dan *monitoring* status mengenai *internal audit* kurang efektif dan efisien karena PIC yang bersangkutan kurang komunikasi serta *email* yang dikirim terdapat kemungkinan tidak tersampaikan atau tidak terbaca.
2. Bagaimana cara merancang sistem informasi *monitoring* hasil *internal audit berbasis web* pada PT. ABC, karena pada saat admin melakukan *input* data hasil *audit*, tidak sedikit tulisan dari *auditor* tidak terbaca yang menyulitkan atau menghambat admin dalam melakukan *input* hasil *audit*.
3. Bagaimana cara merancang sistem informasi *monitoring* hasil *internal audit berbasis web* pada PT. ABC, karena dalam *memonitoring* status *internal audit* secara keseluruhan, departemen yang berkaitan (*auditee*) harus menghubungi PIC yang bersangkutan

melalui pesawat telepon karena penyimpanan masing-masing hasil *audit* dikelola oleh PIC/admin yang berbeda.

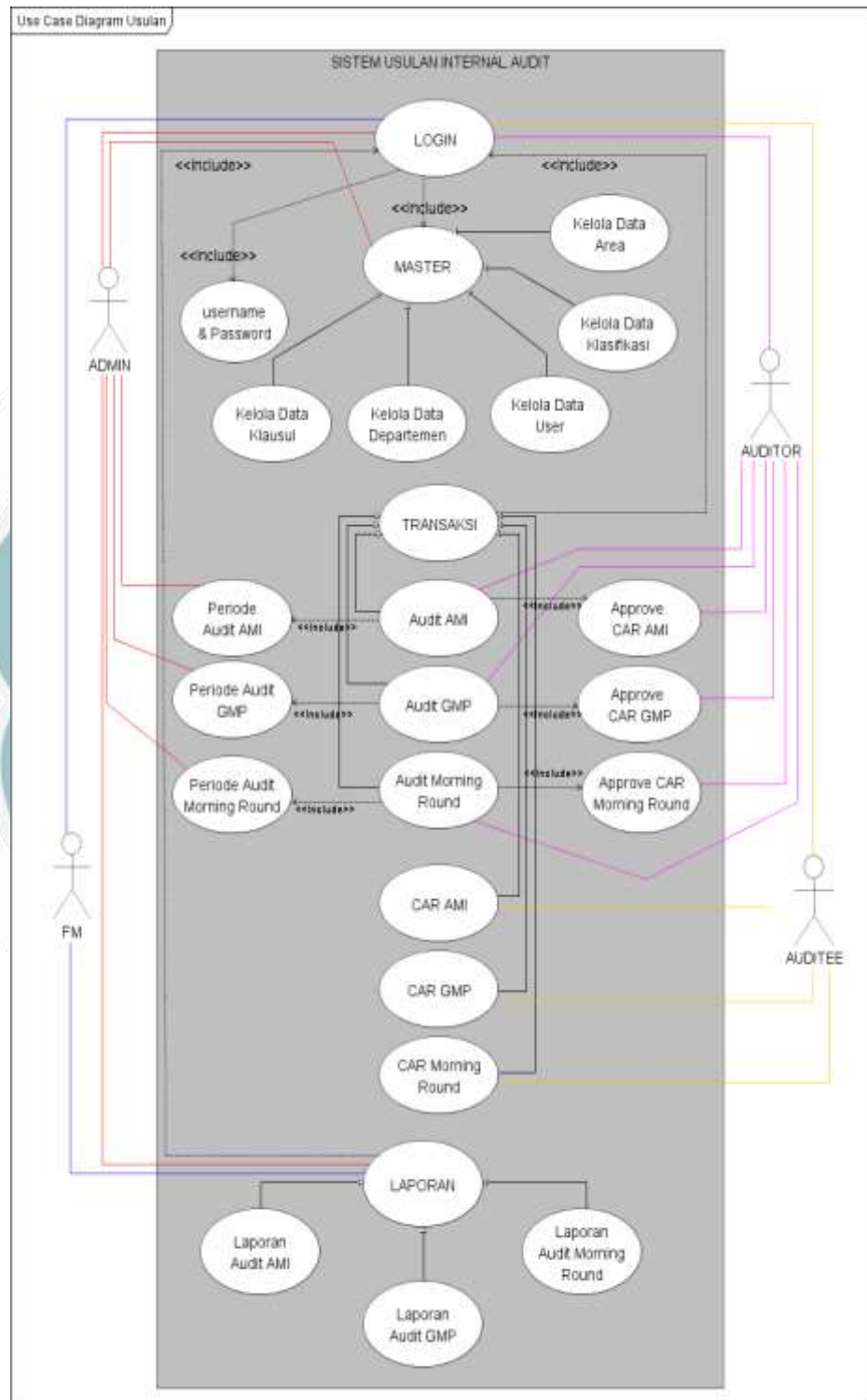
4. Bagaimana cara merancang sistem informasi monitoring hasil internal *audit* berbasis *web* pada PT. ABC, karena PIC/admin yang mengelola hasil *audit* tidak selalu berada ditempat sehingga kurang efektif dan efisien dalam pencarian informasi hasil *audit* secara keseluruhan.

4.1.7. Alternatif Pemecahan Masalah

Setelah mengetahui masalah yang dihadapi, maka penulis mencoba memecahkan masalah tersebut dengan membuat sistem informasi monitoring hasil internal *audit* berbasis *web*. Dengan adanya sistem monitoring hasil internal *audit* ini diharapkan dapat mempermudah semua pihak pada PT. ABC dalam melakukan *audit*, monitoring hasil *audit*, admin tidak lagi melakukan *input* hasil *audit* sehingga dapat mengurangi pekerjaan admin, dan semua data hasil *audit* dapat tersimpan dengan baik didalam *database*, sehingga untuk menyajikan laporan hasil *audit* bisa lebih efektif dan efisien.

4.2. Rancangan Sistem Usulan

4.2.1. Use Case Diagram



GAMBAR 4.6. Use Case Diagram Usulan

a. Deskripsi Aktor

TABEL 4.5. Deskripsi Aktor Sistem Usulan

No	Aktor	Deskripsi
1.	Admin	Aktor yang bertanggung jawab membuat, mengontrol sistem dan mempunyai hak akses penuh serta melakukan proses sistem yang akan diterapkan.
2.	<i>Auditor</i>	Aktor yang bertugas mengaudit, membuat CAR dan melakukan verifikasi CAR
3.	<i>Auditee</i>	Aktor yang diaudit, melengkapi CAR dan menindaklanjuti perbaikan
4.	<i>Factory Manajer (FM)</i>	Aktor yang mempunyai hak akses penuh atas sistem yang dibuat.

b. Deskripsi Use Case

TABEL 4.6. Deskripsi Use Case Sistem Usulan

No	Use Case	Deskripsi
1.	<i>Login</i>	Merupakan proses pengecekan hak akses pengguna dalam melakukan akses sistem.
2.	Master	Menu yang menampilkan tentang beberapa master data
3.	Kelola data <i>user</i>	Merupakan aktifitas mengelola data akun <i>user</i> .
4.	Kelola data departemen	Merupakan aktifitas mengelola data departemen.
5.	Kelola data area	Merupakan aktifitas mengelola data area.
6.	Kelola data klasifikasi	Merupakan aktifitas mengelola data klasifikasi.
7.	Kelola data klausul	Merupakan aktifitas mengelola data klausul.
8.	Transaksi	Merupakan menu yang berisi tentang proses pelaksanaan <i>audit</i> .
9.	Periode <i>Audit</i> AMI	Merupakan menu yang berisi tentang proses pembuatan periode <i>Audit</i> AMI

10.	Periode <i>Audit</i> GMP	Merupakan menu yang berisi tentang proses pembuatan periode <i>Audit</i> GMP
11.	Periode <i>Audit Morning Round</i>	Merupakan menu yang berisi tentang proses pembuatan periode <i>Audit Morning Round</i>
12.	<i>Audit</i> AMI	Merupakan menu yang berisi proses <i>Audit</i> AMI
13.	<i>Audit</i> GMP	Merupakan menu yang berisi proses <i>Audit</i> GMP
14.	<i>Audit Morning Round</i>	Merupakan menu yang berisi proses <i>Audit Morning Round</i>
15.	CAR GMP	Merupakan menu yang berisi proses pembuatan CAR tindakan perbaikan terhadap ketidaksesuaian yang ditemukan pada <i>Audit</i> GMP
16.	CAR AMI	Merupakan menu yang berisi proses pembuatan CAR tindakan perbaikan terhadap ketidaksesuaian yang ditemukan pada <i>Audit</i> AMI
17.	CAR <i>Morning Riund</i>	Merupakan menu yang berisi proses pembuatan CAR tindakan perbaikan terhadap ketidaksesuaian yang ditemukan pada <i>Audit Morning Round</i>
18.	Laporan	Merupakan menu yang berisi data-data laporan hasil <i>audit</i>
19.	Laporan AMI	Merupakan menu yang berisi laporan hasil <i>audit</i> AMI sesuai dengan periode <i>audit</i> yang dilakukan
20.	Laporan GMP	Merupakan menu yang berisi laporan hasil <i>audit</i> GMP sesuai dengan periode <i>audit</i> yang dilakukan
21.	Laporan <i>Morning Round</i>	Merupakan menu yang berisi laporan hasil <i>audit Morning Round</i> sesuai dengan periode audit yang dilakukan

c. Skenario *Use Case*

1) Nama *Use Case* : *Login*

Aktor : Admin, Auditor, Auditee, FM

TABEL 4.7 *Scenario Use Case Login*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. <i>Input username dan password yang valid</i>	
	2. Mengecek <i>valid</i> tidaknya data masukan
	3. Bila <i>valid</i> sistem akan menampilkan halaman menu utama
Skenario Alternatif	
4. <i>Input username dan password yang tidak valid</i>	
	5. Mengecek <i>valid</i> tidaknya data masukan
	6. Menampilkan pesan <i>login tidak valid</i>
7. <i>Input username dan password yang valid</i>	
	8. Mengecek <i>valid</i> tidaknya data masukan
	9. Masuk ke menu utama aplikasi

2) Nama *Use Case* : *Kelola Data User*

Aktor : Admin

TABEL 4.8 *Scenario Use Case Kelola Data User*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu master data	
2. Memilih menu <i>user</i>	
	3. Menampilkan menu daftar <i>user</i>
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data <i>user</i>
6. Memasukan data <i>nik, nama, username, level, departemen, dan password</i>	
7. Klik tombol submit	
	8. Menyimpan data <i>user</i> ke <i>database</i>

	9. Menampilkan daftar <i>user</i>
Skenario Alternatif	
10. Memilih menu master data	
11. Memilih menu <i>user</i>	
12. Memilih data <i>user</i> yang akan diedit	
	13. Menampilkan informasi data <i>user</i> yang akan di-edit
14. Mengganti data nik, nama, <i>username</i> , departemen, level, dan status	
15. Klik tombol <i>update</i>	
	16. Menampilkan informasi data <i>user</i> berhasil di- <i>update</i>
	17. Menyimpan perubahan data <i>user</i> ke- <i>database</i>
	18. Menampilkan daftar <i>user</i>
Skenario Alternatif	
19. Memilih menu master data	
20. Memilih menu <i>user</i>	
21. Memilih data <i>user</i> yang akan di- <i>delete</i>	
	22. Menampilkan informasi data <i>user</i> yang akan di- <i>delete</i>
	23. Menampilkan informasi <i>delete user</i>
24. Klik tombol <i>delete</i>	
	25. Menampilkan informasi <i>user</i> berhasil di- <i>delete</i>
	26. Menghapus data <i>user</i> yang di- <i>delete</i> dari <i>database</i>
	27. Menampilkan daftar <i>user</i>

3) Nama *Use Case* : Kelola Data Departemen

Aktor : Admin

TABEL 4.9 *Scenario Use Case* Kelola Data Departemen

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu master data	

2. Memilih menu departemen	
	3. Menampilkan menu daftar departemen
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data departemen
6. Memasukan data departemen	
7. Klik tombol <i>submit</i>	
	8. Menyimpan data departemen ke- <i>database</i>
	9. Menampilkan daftar departemen
Skenario Alternatif	
10. Memilih menu master data	
11. Memilih menu departemen	
12. Memilih data departemen yang akan <i>diedit</i>	
	13. Menampilkan informasi data departemen yang akan <i>di-edit</i>
14. Mengganti data departemen	
15. Klik tombol <i>update</i>	
	16. Menampilkan informasi data departemen berhasil <i>di-update</i>
	17. Menyimpan perubahan data departemen ke- <i>database</i>
	18. Menampilkan daftar departemen
Skenario Alternatif	
19. Memilih menu master data	
20. Memilih menu departemen	
21. Memilih data departemen yang akan <i>didelete</i>	
	22. Menampilkan informasi data departemen yang akan <i>di-delete</i>

	23. Menampilkan informasi <i>delete</i> departemen
24. Klik tombol <i>delete</i>	
	25. Menampilkan informasi departemen berhasil di- <i>delete</i>
	26. Menghapus data departemen yang di- <i>delete</i> dari <i>database</i>
	27. Menampilkan daftar departemen

4) Nama *Use Case* : Kelola Data Area

Aktor : Admin

TABEL 4.10 *Scenario Use Case* Kelola Data Area

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu master data	
2. Memilih menu area	
	3. Menampilkan menu daftar area
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data area
6. Memasukan data area	
7. Klik tombol <i>submit</i>	
	8. Menyimpan data area ke <i>database</i>
	9. Menampilkan daftar area
Skenario Alternatif	
10. Memilih menu master data	
11. Memilih menu area	
12. Memilih data area yang akan <i>diedit</i>	
	13. Menampilkan informasi data area yang akan di- <i>edit</i>
14. Mengganti data area	

15. Klik tombol <i>update</i>	
	16. Menampilkan informasi data area berhasil di- <i>update</i>
	17. Menyimpan perubahan data area ke <i>database</i>
	18. Menampilkan daftar area
Skenario Alternatif	
19. Memilih menu master data	
20. Memilih menu area	
21. Memilih data area yang akan di- <i>delete</i>	
	22. Menampilkan informasi data area yang akan di- <i>delete</i>
	23. Menampilkan informasi <i>delete</i> area
24. Klik tombol <i>delete</i>	
	25. Menampilkan informasi area berhasil di- <i>delete</i>
	26. Menghapus data area yang di- <i>delete</i> dari <i>database</i>
	27. Menampilkan daftar area

5) Nama *Use Case* : Kelola Data Klasifikasi

Aktor : Admin

TABEL 4.11 *Scenario Use Case* Kelola Data Klasifikasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu master data	
2. Memilih menu klasifikasi	
	3. Menampilkan menu daftar klasifikasi
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data klasifikasi
6. Memasukan data klasifikasi	

7. Klik tombol <i>submit</i>	
	8. Menyimpan data klasifikasi ke- <i>database</i>
	9. Menampilkan daftar klasifikasi
Skenario Alternatif	
10. Memilih menu master data	
11. Memilih menu klasifikasi	
12. Memilih data klasifikasi yang akan diedit	
	13. Menampilkan informasi data klasifikasi yang akan di- <i>edit</i>
14. Mengganti data klasifikasi	
15. Klik tombol <i>update</i>	
	16. Menampilkan informasi data klasifikasi berhasil di- <i>update</i>
	17. Menyimpan perubahan data klasifikasi ke- <i>database</i>
	18. Menampilkan daftar klasifikasi
Skenario Alternatif	
19. Memilih menu master data	
20. Memilih menu klasifikasi	
21. Memilih data klasifikasi yang akan di- <i>delete</i>	
	22. Menampilkan informasi data klasifikasi yang akan di- <i>delete</i>
	23. Menampilkan informasi <i>delete</i> klasifikasi
24. Klik tombol <i>delete</i>	
	25. Menampilkan informasi klasifikasi berhasil di- <i>delete</i>

	26. Menghapus data klasifikasi yang di-delete dari <i>database</i>
	27. Menampilkan daftar klasifikasi

6) Nama *Use Case* : Kelola Data Klausul

Aktor : Admin

TABEL 4.12 *Scenario Use Case* Kelola Data Klausul

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu master data	
2. Memilih menu klausul	
	3. Menampilkan menu daftar klausul
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data klausul
6. Memasukan data klausul	
7. Klik tombol <i>submit</i>	
	8. Menyimpan data klausul ke- <i>database</i>
	9. Menampilkan daftar klausul
Skenario Alternatif	
10. Memilih menu master data	
11. Memilih menu klausul	
12. Memilih data klausul yang akan diedit	
	13. Menampilkan informasi data klausul yang akan di- <i>edit</i>
14. Mengganti data klausul	
15. Klik tombol <i>update</i>	
	16. Menampilkan informasi data klausul berhasil di- <i>update</i>

	17. Menyimpan perubahan data klausul ke- <i>database</i>
	18. Menampilkan daftar klausul
Skenario Alternatif	
19. Memilih menu master data	
20. Memilih menu klausul	
21. Memilih data klausul yang akan di- <i>delete</i>	
	22. Menampilkan informasi data klausul yang akan di- <i>delete</i>
	23. Menampilkan informasi <i>delete</i> klausul
24. Klik tombol <i>delete</i>	
	25. Menampilkan informasi klausul berhasil di- <i>delete</i>
	26. Menghapus data klausul yang di- <i>delete</i> dari <i>database</i>
	27. Menampilkan daftar klausul

7) Nama *Use Case* : Periode Audit Mutu Internal (AMI)

Aktor : Admin

TABEL 4.13 *Scenario Use Case* Periode Audit AMI

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit</i> AMI	
	3. Menampilkan daftar periode <i>Audit</i> AMI
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data periode <i>Audit</i> AMI

6. Memasukan data periode <i>Audit</i> AMI	
7. Klik tombol <i>submit</i>	
	8. Menyimpan data periode <i>Audit</i> AMI ke- <i>database</i>
	9. Menampilkan daftar periode <i>Audit</i> AMI
Skenario Alternatif	
10. Memilih menu transaksi	
11. Memilih menu <i>Audit</i> AMI	
12. Memilih data periode <i>Audit</i> AMI yang akan di- <i>edit</i>	
	13. Menampilkan informasi data periode <i>Audit</i> AMI yang akan di- <i>edit</i>
14. Mengganti data periode <i>Audit</i> AMI	
15. Klik tombol <i>submit</i>	
	16. Menampilkan informasi data periode berhasil di- <i>update</i>
	17. Menyimpan perubahan data periode <i>Audit</i> AMI ke- <i>database</i>
	18. Menampilkan daftar periode <i>Audit</i> AMI

8) Nama *Use Case* : Periode *Audit* GMP

Aktor : Admin

TABEL 4.14 *Scenario Use Case* Periode *Audit* GMP

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit</i> GMP	
	3. Menampilkan daftar periode <i>Audit</i> GMP
4. Klik tombol tambah data	

	5. Menampilkan menu tambah data periode <i>Audit GMP</i>
6. Memasukan data periode <i>Audit GMP</i>	
7. Klik tombol <i>submit</i>	
	8. Menyimpan data periode <i>Audit GMP</i> ke <i>database</i>
	9. Menampilkan daftar periode <i>Audit GMP</i>
Skenario Alternatif	
10. Memilih menu transaksi	
11. Memilih menu <i>Audit GMP</i>	
12. Memilih data periode <i>Audit GMP</i> yang akan di- <i>edit</i>	
	13. Menampilkan informasi data periode <i>Audit GMP</i> yang akan di- <i>edit</i>
14. Mengganti data periode <i>Audit GMP</i>	
15. Klik tombol <i>submit</i>	
	16. Menampilkan informasi data periode berhasil di- <i>update</i>
	17. Menyimpan perubahan data periode <i>Audit GMP</i> ke- <i>database</i>
	18. Menampilkan daftar periode <i>Audit GMP</i>

9) Nama *Use Case* : Periode *Audit Morning Round*

Aktor : Admin

TABEL 4.15 *Scenario Use Case* Periode *Audit Morning Round*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	

2. Memilih menu <i>Audit Morning Round</i>	
	3. Menampilkan daftar periode <i>Audit Morning Round</i>
4. Klik tombol tambah data	
	5. Menampilkan menu tambah data periode <i>Audit Morning Round</i>
6. Memasukan data periode <i>Audit Morning Round</i>	
7. Klik tombol <i>submit</i>	
	8. Menyimpan data periode <i>Audit Morning Round</i> ke <i>database</i>
	9. Menampilkan daftar periode <i>Audit Morning Round</i>
Skenario Alternatif	
10. Memilih menu transaksi	
11. Memilih menu <i>Audit Morning Round</i>	
12. Memilih data periode <i>Audit Morning Round</i> yang akan di-edit	
	13. Menampilkan informasi data periode <i>Audit Morning Round</i> yang akan di-edit
14. Mengganti data periode <i>Audit Morning Round</i>	
15. Klik tombol <i>submit</i>	
	16. Menampilkan informasi data periode berhasil di- <i>update</i>
	17. Menyimpan perubahan data periode <i>Audit Morning Round</i> ke- <i>database</i>

	18. Menampilkan daftar periode <i>Audit Morning Round</i>
--	---

10) Nama *Use Case* : *Audit Mutu Internal (AMI)*

Aktor : *Auditor*

TABEL 4.16 *Scenario Use Case Audit Mutu Internal (AMI)*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit AMI</i>	
	3. Menampilkan daftar periode <i>Audit AMI</i>
4. Klik <i>list audit</i>	
	5. Menampilkan daftar <i>Audit AMI</i>
6. Klik <i>audit</i>	
	7. Menampilkan form <i>Checklis Audit AMI</i>
8. Memasukan data <i>audit</i>	
9. Klik tombol <i>submit</i>	
	10. Menyimpan data <i>audit</i> ke <i>database</i>
	11. Menampilkan daftar <i>Audit AMI</i>
Skenario Alternatif	
12. Memilih menu transaksi	
13. Memilih menu <i>Audit AMI</i>	
	14. Menampilkan daftar periode <i>Audit AMI</i>
15. Klik tambah pertanyaan	
	16. Menampilkan <i>form</i> pertanyaan <i>auditor</i>
17. Memasukan data pertanyaan	
18. Klik tombol <i>submit</i>	

	19. Menampilkan daftar <i>Audit</i> AMI
Skenario Alternatif	
20. Memilih menu transaksi	
21. Memilih menu <i>Audit</i> AMI	
	22. Menampilkan daftar periode <i>Audit</i> AMI
23. Memilih <i>list audit</i> yang akan di- <i>edit</i>	
	24. Menampilkan form <i>checklis</i> <i>Audit</i> AMI yang akan di- <i>edit</i>
25. Mengganti data <i>list audit</i> yang akan di- <i>edit</i>	
26. Klik tombol <i>submit</i>	
	27. Menyimpan perubahan ke- <i>database</i>
	28. Menampilkan daftar <i>Audit</i> AMI

11) Nama *Use Case* : *Audit* GMP

Aktor : Auditor

TABEL 4.17 *Scenario Use Case Audit* GMP

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit</i> GMP	
	3. Menampilkan daftar periode <i>Audit</i> GMP
4. Klik <i>list audit</i>	
	5. Menampilkan daftar <i>Audit</i> GMP
6. Klik <i>audit</i>	
	7. Menampilkan form <i>checklis</i> <i>Audit</i> GMP
8. Memasukan data <i>audit</i>	

9. Klik tombol <i>submit</i>	
	10. Menyimpan data <i>audit</i> ke <i>database</i>
	11. Menampilkan daftar <i>Audit GMP</i>
Skenario Alternatif	
12. Memilih menu transaksi	
13. Memilih menu <i>Audit GMP</i>	
	14. Menampilkan daftar periode <i>Audit GMP</i>
15. Klik tambah pertanyaan	
	16. Menampilkan <i>form</i> pertanyaan <i>auditor</i>
17. Memasukan data pertanyaan	
18. Klik tombol <i>submit</i>	
	19. Menampilkan daftar <i>Audit GMP</i>
Skenario Alternatif	
20. Memilih menu transaksi	
21. Memilih menu <i>Audit GMP</i>	
	22. Menampilkan daftar periode <i>Audit GMP</i>
23. Memilih <i>list audit</i> yang akan di- <i>edit</i>	
	24. Menampilkan <i>form checklis Audit GMP</i> yang akan di- <i>edit</i>
25. Mengganti data <i>list audit</i> yang akan di- <i>edit</i>	
26. Klik tombol <i>submit</i>	
	27. Menyimpan perubahan ke- <i>database</i>
	28. Menampilkan daftar <i>Audit GMP</i>

12) Nama *Use Case* : *Audit Morning Round*

Aktor : *Auditor*

TABEL 4.18 *Scenario Use Case Audit Morning Round*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit Morning Round</i>	
	3. Menampilkan daftar periode <i>Audit Morning Round</i>
4. Klik <i>list audit</i>	
	5. Menampilkan daftar <i>Audit Morning Round</i>
6. Klik <i>audit</i>	
	7. Menampilkan form <i>Audit Morning Round</i>
8. Memasukan data <i>audit</i>	
9. Klik tombol <i>submit</i>	
	10. Menyimpan data <i>audit</i> ke <i>database</i>
	11. Menampilkan daftar <i>Audit Morning Round</i>
Skenario Alternatif	
12. Memilih menu transaksi	
13. Memilih menu <i>Audit Morning Round</i>	
	14. Menampilkan daftar periode <i>Audit Morning Round</i>
15. Memilih aksi <i>audit</i> yang akan di- <i>edit</i>	
	16. Menampilkan form <i>Audit Morning Round</i> yang akan di- <i>edit</i>
17. Mengganti data <i>audit</i> yang akan di- <i>edit</i>	
18. Klik tombol <i>submit</i>	

	19. Menyimpan perubahan ke- <i>database</i>
	20. Menampilkan daftar <i>Audit Morning Round</i>

13) Nama *Use Case* : CAR Audit Mutu Internal (AMI)

Aktor : *Auditee*

TABEL 4.19 *Scenario Use Case* CAR AMI

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih CAR AMI	
	3. Menampilkan periode <i>Audit AMI</i>
4. Memilih <i>list audit</i>	
	5. Menampilkan daftar <i>Audit AMI</i>
6. Memiliki aksi CAR	
	7. Menampilkan form CAR <i>auditee</i>
8. Mengisi form CAR	
9. Klik tombol <i>submit</i>	
	10. Menyimpan data CAR ke <i>database</i>
	11. Menampilkan daftar <i>Audit AMI</i>
Skenario Alternatif	
12. Memilih menu transaksi	
13. Memilih CAR AMI	
	14. Menampilkan periode <i>Audit AMI</i>
15. Memilih <i>list audit</i>	
	16. Menampilkan daftar <i>Audit AMI</i>
17. Memiliki aksi CAR yang telah diisi	

	18. Menampilkan form CAR <i>auditee</i> yang telah diisi
19. Mengganti data CAR yang akan di- <i>edit</i>	
20. Klik tombol <i>submit</i>	
	21. Menyimpan perubahan data CAR ke <i>database</i>
	22. Menampilkan daftar <i>Audit AMI</i>

14) Nama *Use Case* : CAR GMP

Aktor : *Auditee*

TABEL 4.20 *Scenario Use Case* CAR GMP

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih CAR GMP	
	3. Menampilkan periode <i>Audit GMP</i>
4. Memilih <i>list audit</i>	
	5. Menampilkan daftar <i>Audit GMP</i>
6. Memilih aksi CAR	
	7. Menampilkan form CAR <i>auditee</i>
8. Mengisi form CAR	
9. Klik tombol <i>submit</i>	
	10. Menyimpan data CAR ke <i>database</i>
	11. Menampilkan daftar <i>Audit GMP</i>
Skenario Alternatif	
12. Memilih menu transaksi	
13. Memilih CAR GMP	
	14. Menampilkan periode <i>Audit GMP</i>
15. Memilih <i>list audit</i>	

	16. Menampilkan daftar <i>Audit GMP</i>
17. Memiliki aksi CAR yang telah diisi	
	18. Menampilkan form CAR <i>auditee</i> yang telah diisi
19. Mengganti data CAR yang akan di-edit	
20. Klik tombol <i>submit</i>	
	21. Menyimpan perubahan data CAR ke <i>database</i>
	22. Menampilkan daftar <i>Audit GMP</i>

15) Nama *Use Case* : CAR Morning Round

Aktor : *Auditee*

TABEL 4.21 *Scenario Use Case CAR Morning Round*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih CAR Morning Round	
	3. Menampilkan periode <i>Audit Morning Round</i>
4. Memilih <i>list audit</i>	
	5. Menampilkan daftar <i>Audit Morning Round</i>
6. Memiliki aksi CAR	
	7. Menampilkan form CAR <i>auditee</i>
8. Mengisi form CAR	
9. Klik tombol <i>submit</i>	
	10. Menyimpan data CAR ke <i>database</i>
	11. Menampilkan daftar <i>Audit Morning Round</i>
Skenario Alternatif	

12. Memilih menu transaksi	
13. Memilih CAR <i>Morning Round</i>	
	14. Menampilkan periode <i>Audit Morning Round</i>
15. Memilih <i>list audit</i>	
	16. Menampilkan daftar <i>Audit Morning Round</i>
17. Memilih aksi CAR yang telah diisi	
	18. Menampilkan form CAR <i>auditee</i> yang telah diisi
19. Mengganti data CAR yang akan di-edit	
20. Klik tombol <i>submit</i>	
	21. Menyimpan perubahan data CAR ke <i>database</i>
	22. Menampilkan daftar <i>Audit Morning Round</i>

16) Nama *Use Case* : Approve CAR AMI

Aktor : Auditor

TABEL 4.22 *Scenario Use Case Approve CAR AMI*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit AMI</i>	
	3. Menampilkan daftar periode <i>Audit AMI</i>
4. Klik <i>list audit</i>	
	5. Menampilkan daftar <i>Audit AMI</i>
6. Klik CAR	
	7. Menampilkan <i>form CAR AMI</i>
8. Klik <i>Approve</i>	

	9. Menampilkan daftar <i>Audit</i> AMI
--	--

17) Nama *Use Case* : *Approve CAR GMP*

Aktor : *Auditor*

TABEL 4.23 *Scenario Use Case Approve CAR GMP*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit GMP</i>	
	3. Menampilkan daftar periode <i>Audit GMP</i>
4. Klik <i>list audit</i>	
	5. Menampilkan daftar <i>Audit GMP</i>
6. Klik CAR	
	7. Menampilkan <i>form CAR GMP</i>
8. Klik <i>Approve</i>	
	9. Menampilkan daftar <i>Audit GMP</i>

18) Nama *Use Case* : *Approve CAR Morning Round*

Aktor : *Auditor*

TABEL 4.24 *Scenario Use Case Approve CAR Morning Round*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu transaksi	
2. Memilih menu <i>Audit Morning Round</i>	
	3. Menampilkan daftar periode <i>Audit Morning Round</i>
4. Klik <i>list audit</i>	

	5. Menampilkan daftar <i>Audit Morning Round</i>
6. Klik CAR	
	7. Menampilkan <i>form CAR Morning Round</i>
8. Klik <i>Approve</i>	
	9. Menampilkan daftar <i>Audit Morning Round</i>

19) Nama *Use Case* : Laporan *Audit AMI*

Aktor : Admin dan *Factory Manajer* (FM)

TABEL 4.25 *Scenario Use Case* Laporan AMI

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu laporan	
2. Memilih <i>Audit AMI</i>	
	3. Menampilkan data laporan <i>Audit AMI</i>
4. Klik <i>export data</i>	
	5. Sistem <i>export data</i>

20) Nama *Use Case* : Laporan *Audit GMP*

Aktor : Admin dan *Factory Manajer* (FM)

TABEL 4.26 *Scenario Use Case* Laporan GMP

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu laporan	
2. Memilih <i>Audit GMP</i>	
	3. Menampilkan data laporan <i>Audit GMP</i>
4. Klik <i>export data</i>	
	5. Sistem <i>export data</i>

21) Nama *Use Case* : Laporan *Audit Morning Round*

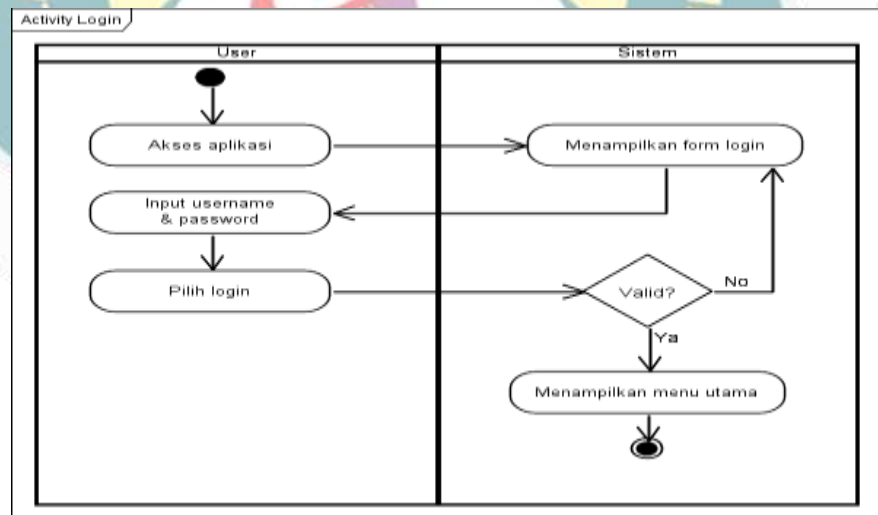
Aktor : Admin dan *Factory Manager* (FM)

TABEL 4.27 *Scenario Use Case* Laporan *Morning Round*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih menu laporan	
2. Memilih <i>Audit Morning Round</i>	
	3. Menampilkan data laporan <i>Audit Morning Round</i>
4. Klik <i>export</i> data	
	5. Sistem <i>export</i> data

4.2.2. Activity Diagram

a. Activity Diagram Login



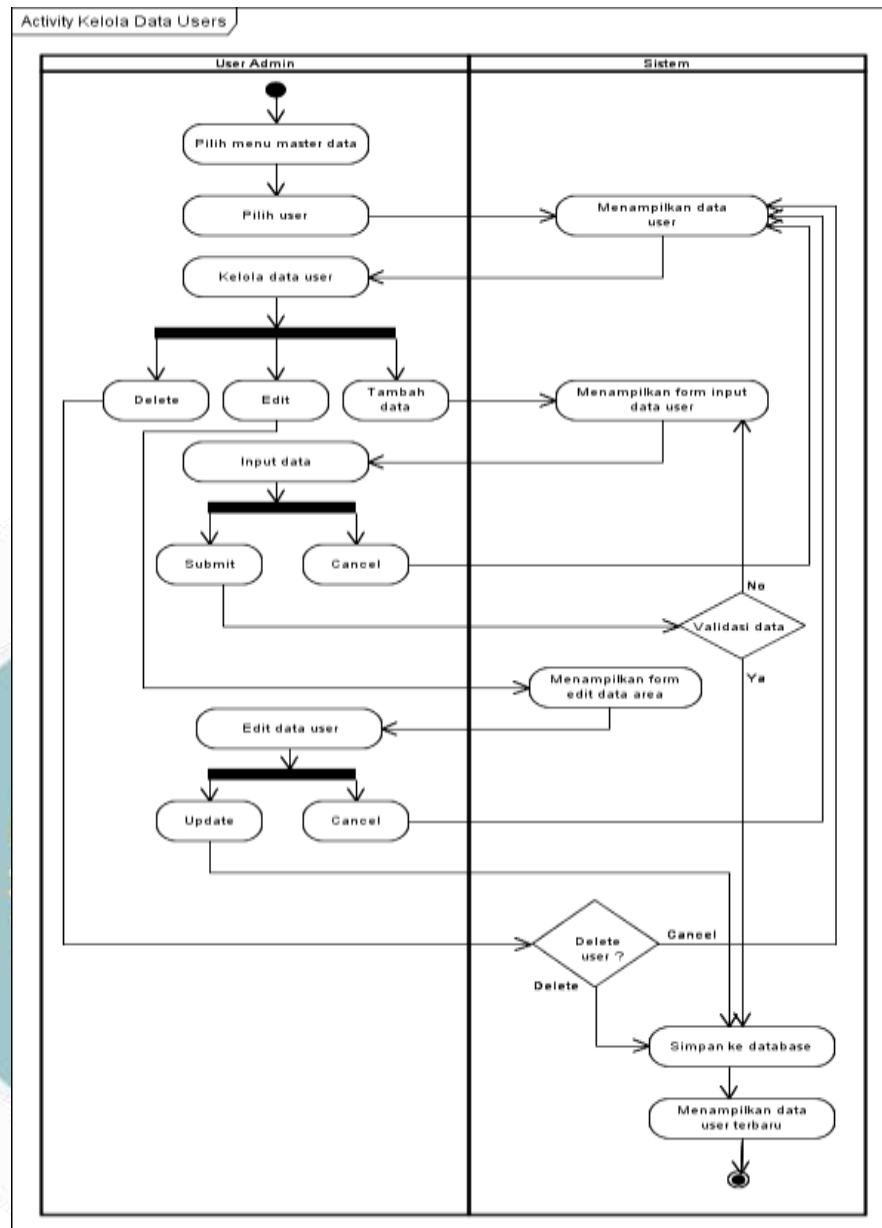
GAMBAR 4.7 Activity Diagram Login

Keterangan:

1) Mulai

- 2) *User* mengakses aplikasi.
 - 3) Sistem menampilkan *form Login*.
 - 4) *Input username* dan *password*, klik tombol *Login*
 - 5) Sistem memvalidasi apakah *username* dan *password* yang di *input valid* atau *invalid*, jika *valid* sistem akan menampilkan menu utama dan jika *invalid* sistem akan menampilkan form *input login*.
 - 6) Selesai.
- b. *Activity Diagram* Kelola Data *Users*





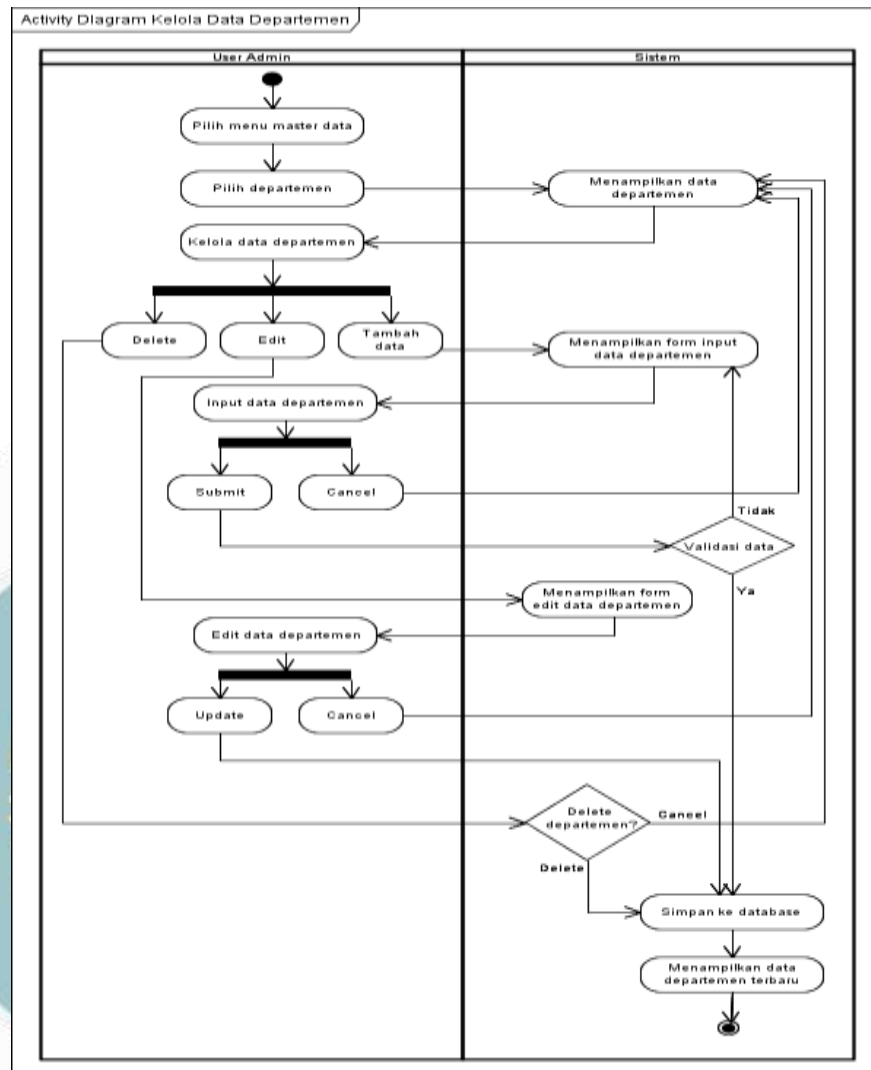
GAMBAR 4.8 Activity Diagram Kelola Data User

Keterangan:

- 1) Mulai
- 2) Admin memilih menu master data
- 3) Admin memilih master *user*
- 4) Sistem menampilkan data *user*
- 5) Admin kelola data *user* dengan memilih tambah data

- 6) Sistem menampilkan *form input* data
- 7) Admin *input* data *user*, kemudian pilih *button Submit*
- 8) Sistem akan memvalidasi data *inputan*, jika *inputan* sesuai kriteria *inputan* sistem akan menyimpan ke *database*, jika tidak sesuai kriteria *inputan* maka akan menampilkan pesan lalu admin kembali *input* data, kemudian sistem menampilkan data *user* terbaru
- 9) Admin *input* data *user*, kemudian pilih *button cancel*
- 10) Sistem akan menampilkan kembali data *user*
- 11) Admin kelola data *user* dengan memilih *edit*
- 12) Sistem menampilkan *form edit* data *user*
- 13) Admin mengubah data yang akan *diedit*
- 14) Admin memilih tombol *update*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan data *user* terbaru
- 15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data *user*
- 16) Admin kelola data *user* dengan memilih *delete*, maka sistem akan menampilkan validasi, jika memilih tombol *delete* maka data yang dipilih akan terhapus dari *database* dan kemudian perubahan akan disimpan pada *database* dan sistem menampilkan data *user* terbaru, jika memilih tombol *cancel* maka sistem akan kembali ke tampilan data *user*
- 17) Selesai

c. *Activity Diagram* Kelola Data Departemen

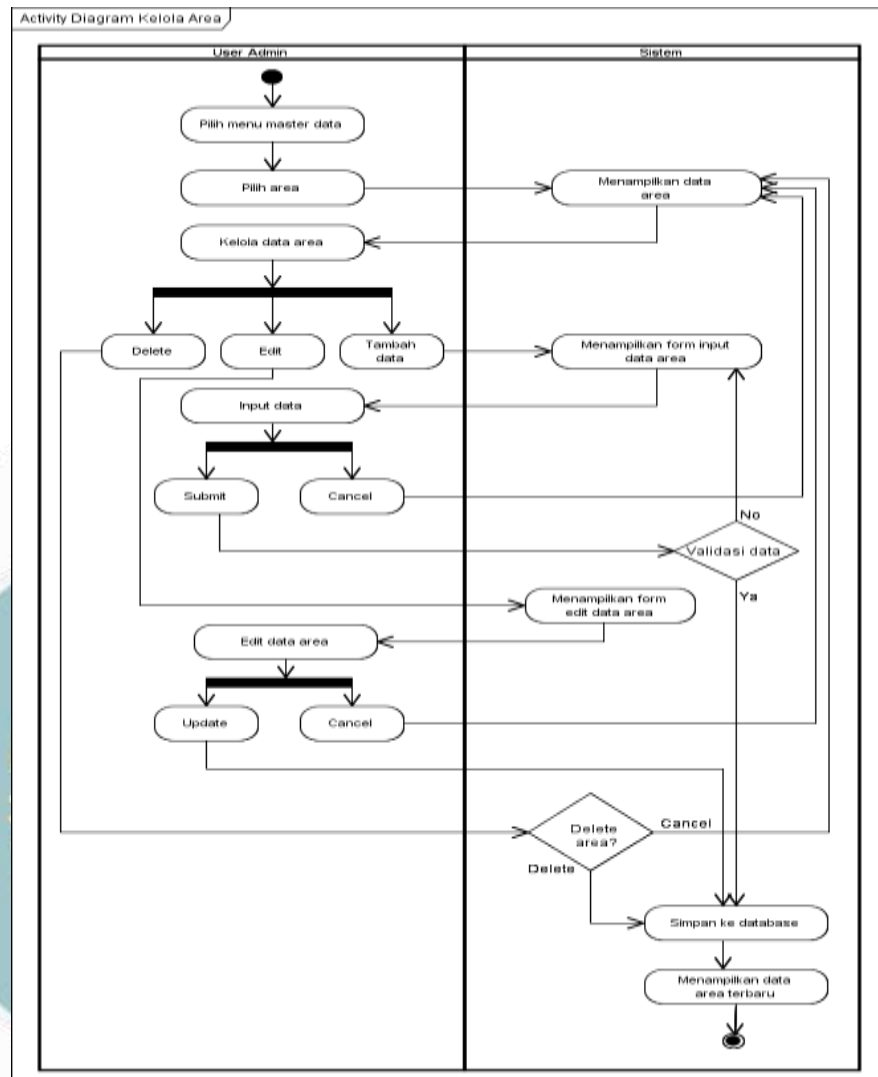


GAMBAR 4.9 *Activity Diagram* Kelola Data Departemen
Keterangan:

- 1) Mulai
- 2) Admin memilih menu master data
- 3) Admin memilih master departemen
- 4) Sistem menampilkan data departemen
- 5) Admin kelola data *user* dengan memilih tambah data
- 6) Sistem menampilkan *form input* data

- 7) Admin *input* data departemen, kemudian pilih *button Submit*
- 8) Sistem akan memvalidasi data *inputan*, jika *inputan* sesuai kriteria *inputan* sistem akan menyimpan ke *database*, jika tidak sesuai kriteria *inputan* maka akan menampilkan pesan lalu admin kembali *input* data, kemudian sistem menampilkan data departemen terbaru
- 9) Admin *input* data departemen, kemudian pilih *button cancel*
- 10) Sistem akan menampilkan kembali data departemen
- 11) Admin kelola data departemen dengan memilih *edit*
- 12) Sistem menampilkan *form edit* data departemen
- 13) Admin mengubah data yang akan *diedit*
- 14) Admin memilih tombol *update*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan data departemen terbaru
- 15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data departemen
- 16) Admin kelola data departemen dengan memilih *delete*, maka sistem akan menampilkan validasi, jika memilih tombol *delete* maka data yang dipilih akan terhapus dari *database* dan kemudian perubahan akan disimpan pada *database* dan sistem menampilkan data departemen terbaru, jika memilih tombol *cancel* maka sistem akan kembali ke tampilan data departemen
- 17) Selesai

d. *Activity Diagram* Kelola Data Area



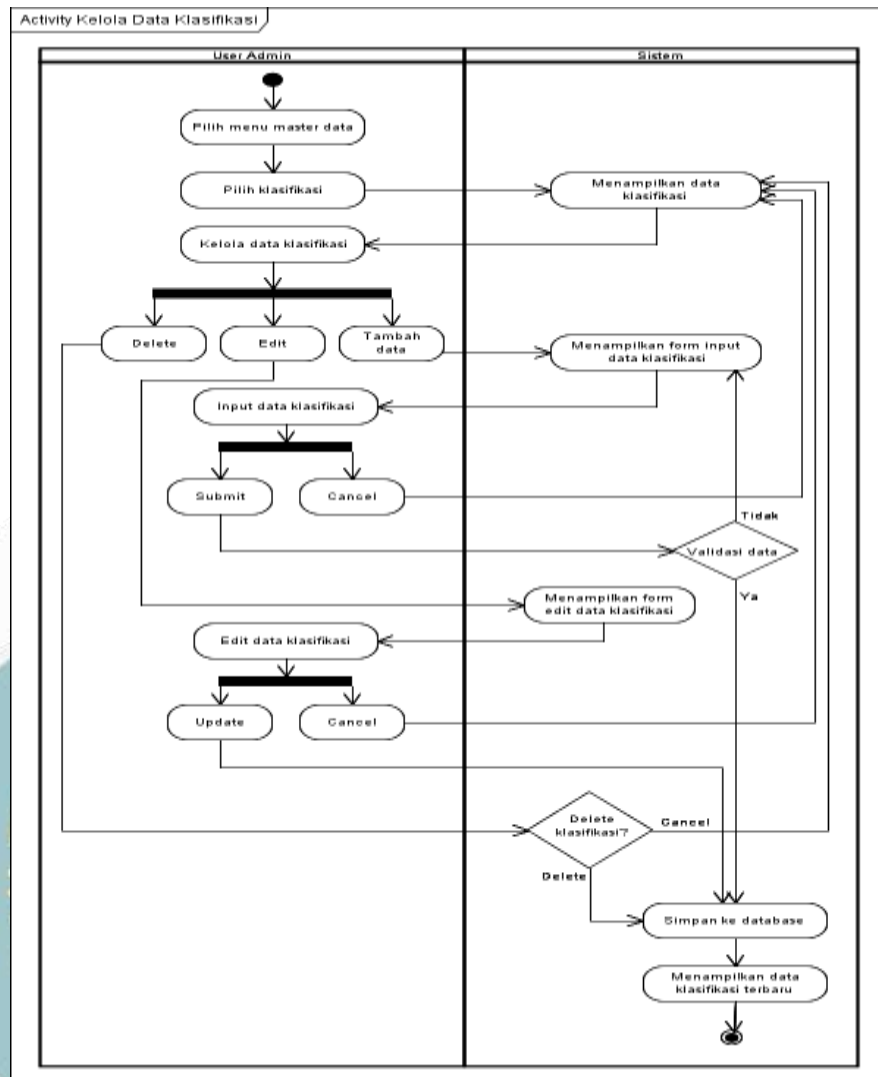
GAMBAR 4.10 *Activity Diagram* Kelola Data Area

Keterangan:

- 1) Mulai
- 2) Admin memilih menu master data
- 3) Admin memilih master area
- 4) Sistem menampilkan data area
- 5) Admin kelola data *user* dengan memilih tambah data
- 6) Sistem menampilkan *form input* data

- 7) Admin *input* data area, kemudian pilih *button Submit*
- 8) Sistem akan memvalidasi data *inputan*, jika *inputan* sesuai kriteria *inputan* sistem akan menyimpan ke *database*, jika tidak sesuai kriteria *inputan* maka akan menampilkan pesan lalu admin kembali *input* data, kemudian sistem menampilkan data area terbaru
- 9) Admin *input* data area, kemudian pilih *button cancel*
- 10) Sistem akan menampilkan kembali data area
- 11) Admin kelola data area dengan memilih *edit*
- 12) Sistem menampilkan form *edit* data area
- 13) Admin mengubah data yang akan *diedit*
- 14) Admin memilih tombol *update*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan data area terbaru
- 15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data area
- 16) Admin kelola data area dengan memilih *delete*, maka sistem akan menampilkan validasi, jika memilih tombol *delete* maka data yang dipilih akan terhapus dari *database* dan kemudian perubahan akan disimpan pada *database* dan sistem menampilkan data area terbaru, jika memilih tombol *cancel* maka sistem akan kembali ke tampilan data area
- 17) Selesai

e. *Activity Diagram* Kelola Data Klasifikasi



GAMBAR 4.11 Activity Diagram Kelola Data Klasifikasi
Keterangan:

- 1) Mulai
- 2) Admin memilih menu master data
- 3) Admin memilih master klasifikasi
- 4) Sistem menampilkan data klasifikasi
- 5) Admin kelola data klasifikasi dengan memilih tambah data
- 6) Sistem menampilkan *form input* data
- 7) Admin *input* data klasifikasi, kemudian pilih *button Submit*

8) Sistem akan memvalidasi data *inputan*, jika *inputan* sesuai kriteria *inputan* sistem akan menyimpan ke *database*, jika tidak sesuai kriteria *inputan* maka akan menampilkan pesan lalu admin kembali *input* data, kemudian sistem menampilkan data klasifikasi terbaru

9) Admin *input* data area, kemudian pilih *button cancel*

10) Sistem akan menampilkan kembali data klasifikasi

11) Admin kelola data klasifikasi dengan memilih *edit*

12) Sistem menampilkan form *edit* data klasifikasi

13) Admin mengubah data yang akan *diedit*

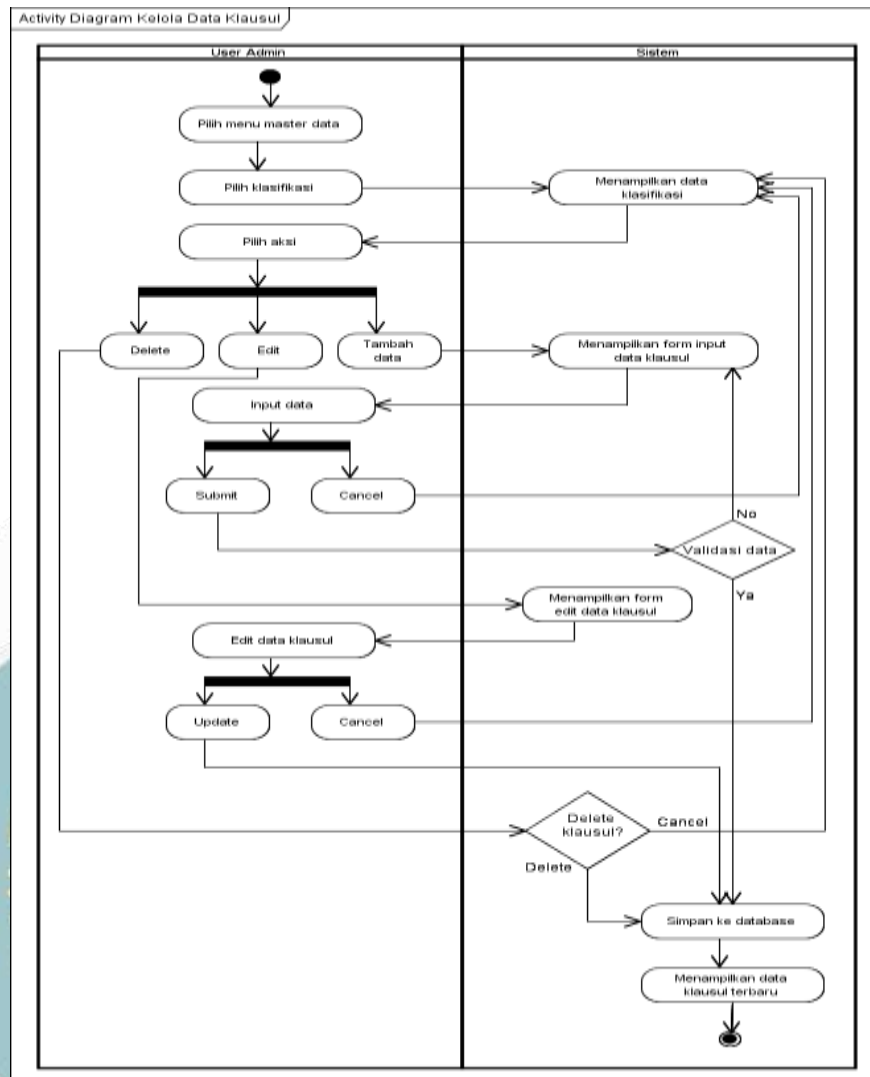
14) Admin memilih tombol *update*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan data klasifikasi terbaru

15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data klasifikasi

16) Admin kelola data area dengan memilih *delet*, maka sistem akan menampilkan validasi, jika memilih tombol *delete* maka data yang dipilih akan terhapus dari *database* dan kemudian perubahan akan disimpan pada *database* dan sistem menampilkan data klasifikasi terbaru, jika memilih tombol *cancel* maka sistem akan kembali ke tampilan data klasifikasi

17) Selesai

f. *Activity Diagram* Kelola Data Klausul



GAMBAR 4.12 *Activity Diagram* Kelola Data Klausul
Keterangan:

- 1) Mulai
- 2) Admin memilih menu master data
- 3) Admin memilih master klausul
- 4) Sistem menampilkan data klausul
- 5) Admin kelola data klausul dengan memilih tambah data
- 6) Sistem menampilkan *form input* data
- 7) Admin *input* data klausul, kemudian pilih *button Submit*

8) Sistem akan memvalidasi data *inputan*, jika *inputan* sesuai kriteria *inputan* sistem akan menyimpan ke *database*, jika tidak sesuai kriteria *inputan* maka akan menampilkan pesan lalu admin kembali *input* data, kemudian sistem menampilkan data klausul terbaru

9) Admin *input* data area, kemudian pilih *button cancel*

10) Sistem akan menampilkan kembali data klausul

11) Admin kelola data klasifikasi dengan memilih edit

12) Sistem menampilkan form edit data klausul

13) Admin mengubah data yang akan diedit

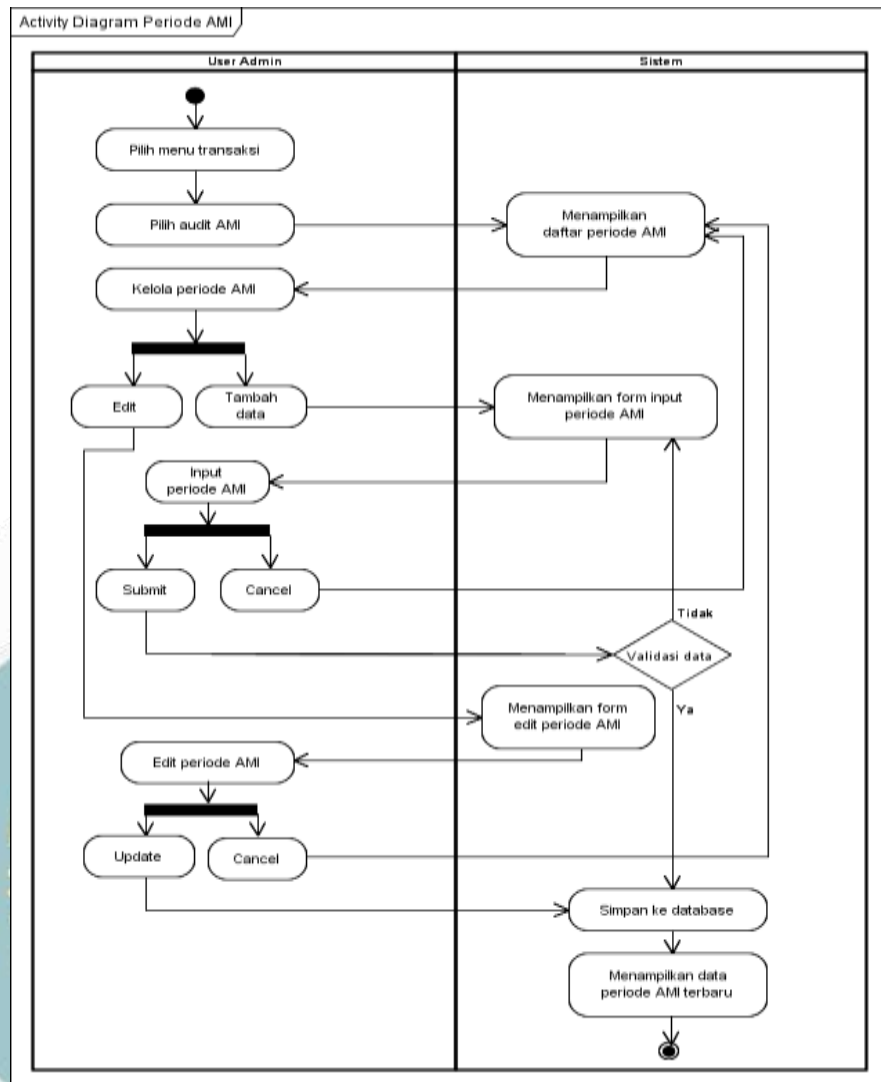
14) Admin memilih tombol *update*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan data klausul terbaru

15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data klausul

16) Admin kelola data area dengan memilih *delete*, maka sistem akan menampilkan validasi, jika memilih tombol *delete* maka data yang dipilih akan terhapus dari *database* dan kemudian perubahan akan disimpan pada *database* dan sistem menampilkan data klausul terbaru, jika memilih tombol *cancel* maka sistem akan kembali ke tampilan data klausul

17) Selesai

g. *Activity Diagram Periode Audit Mutu Internal (AMI)*



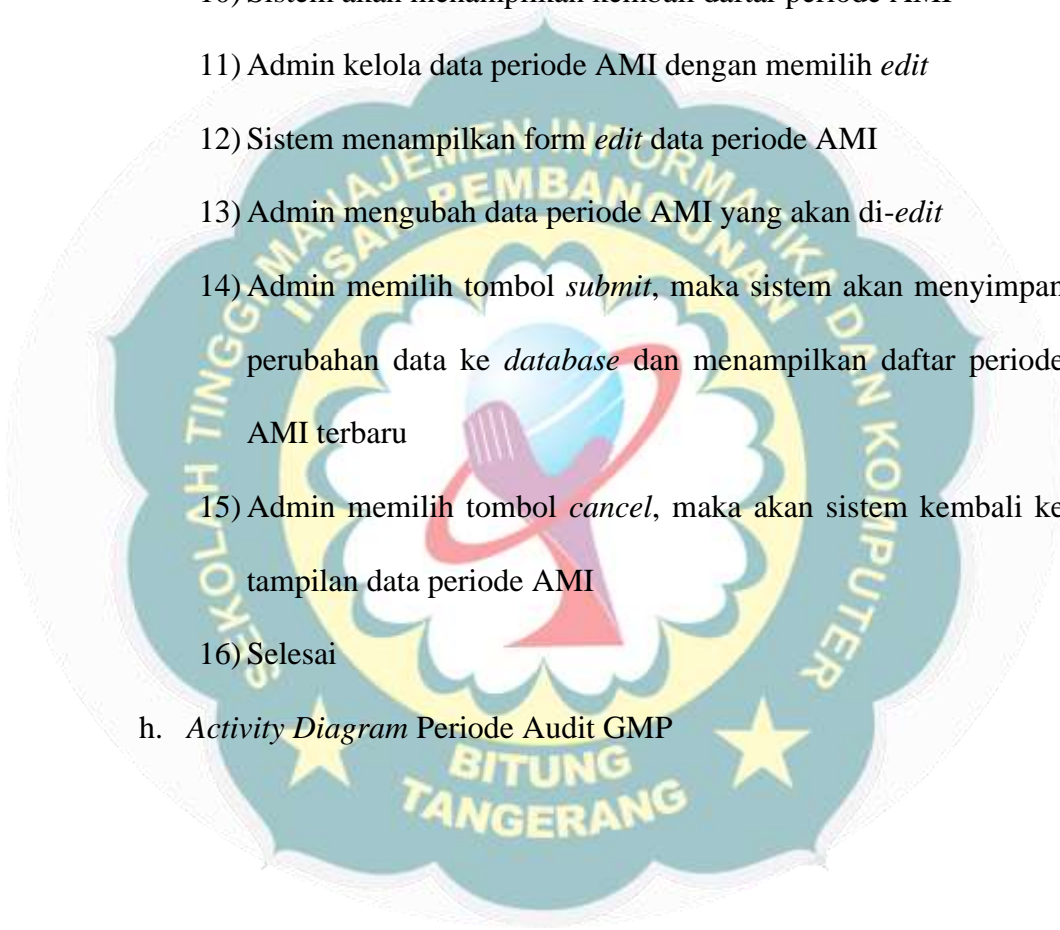
GAMBAR 4.13 Activity Diagram Periode AMI

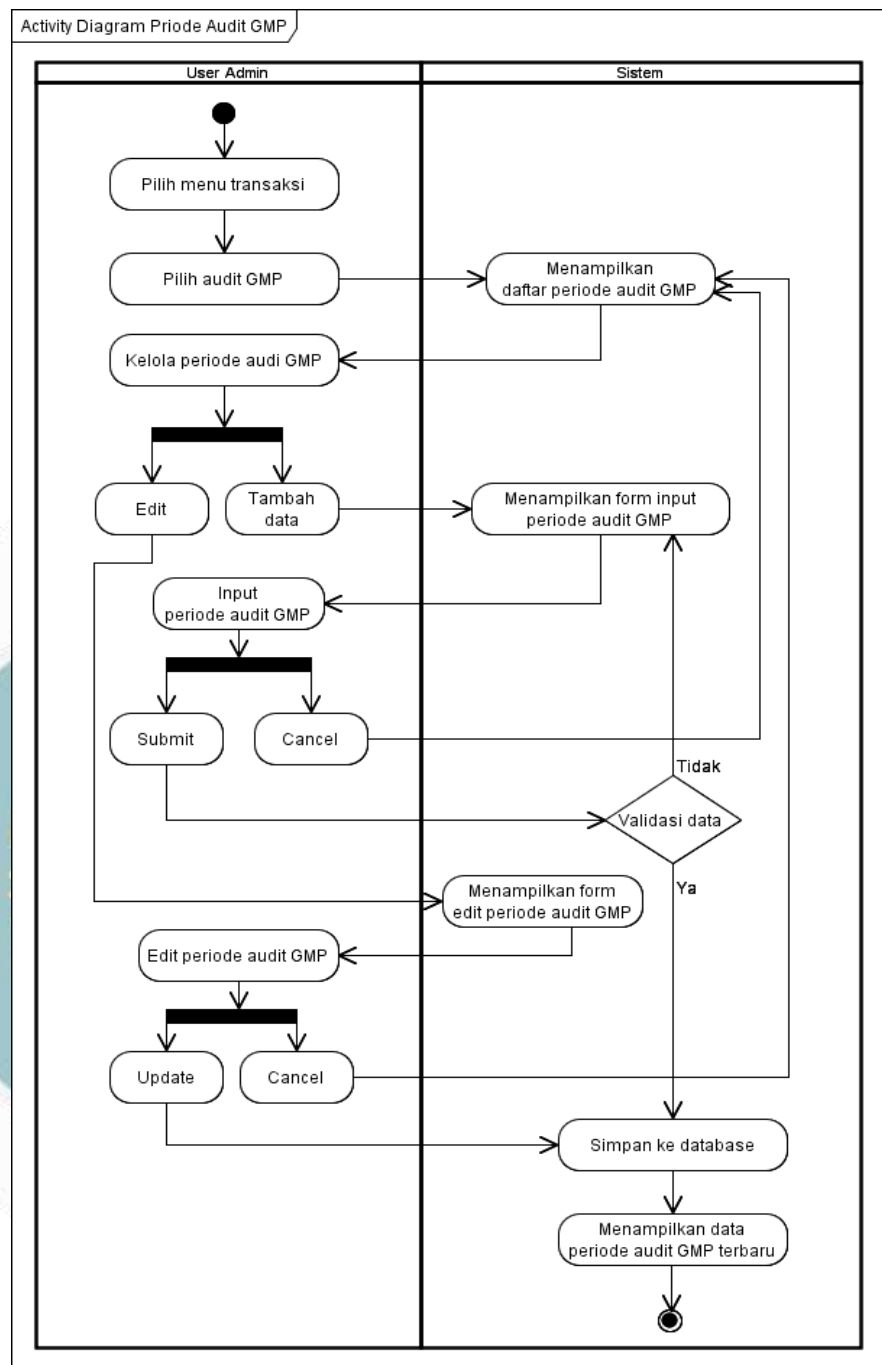
Keterangan:

- 1) Mulai
- 2) Admin memilih menu transaksi
- 3) Admin memilih *Audit* AMI
- 4) Sistem menampilkan daftar periode *Audit* AMI
- 5) Admin kelola periode *Audit* AMI dengan memilih tambah data
- 6) Sistem menampilkan *form* tambah periode AMI
- 7) Admin *input* data periode AMI, kemudian pilih *button Submit*

- 8) Sistem akan memvalidasi data *inputan*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu admin kembali *input* data *valid*, kemudian sistem menampilkan daftar periode AMI terbaru
- 9) Admin *input* data periode AMI, kemudian pilih *button cancel*
- 10) Sistem akan menampilkan kembali daftar periode AMI
- 11) Admin kelola data periode AMI dengan memilih *edit*
- 12) Sistem menampilkan form *edit* data periode AMI
- 13) Admin mengubah data periode AMI yang akan di-*edit*
- 14) Admin memilih tombol *submit*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan daftar periode AMI terbaru
- 15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data periode AMI
- 16) Selesai

h. *Activity Diagram* Periode Audit GMP





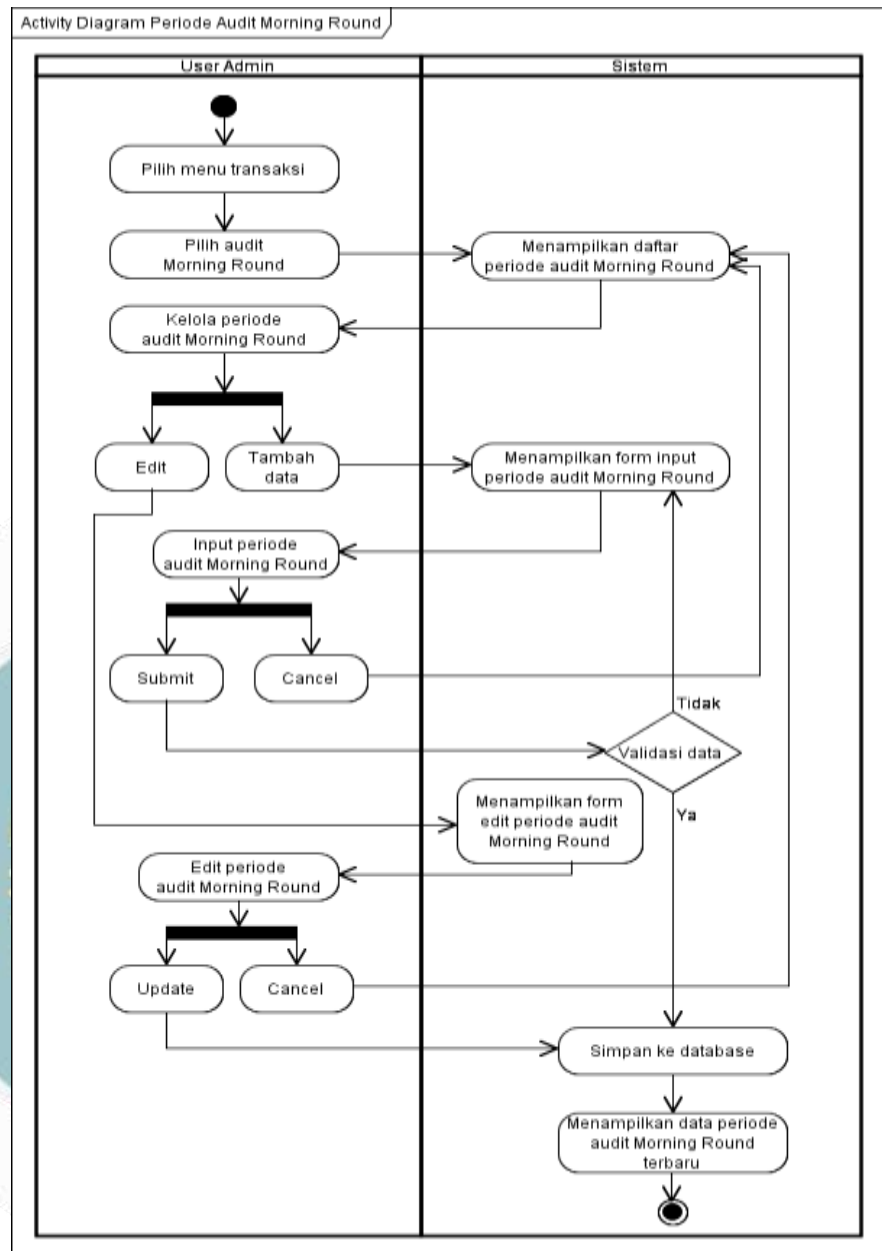
GAMBAR 4.14 Activity Diagram Periode Audit GMP

Keterangan:

- 1) Mulai
- 2) Admin memilih menu transaksi
- 3) Admin memilih *Audit* GMP

- 4) Sistem menampilkan daftar periode *Audit* GMP
- 5) Admin kelola periode *Audit* GMP dengan memilih tambah data
- 6) Sistem menampilkan *form* tambah periode *Audit* GMP
- 7) Admin *input* data periode *Audit* GMP, kemudian pilih *button Submit*
- 8) Sistem akan memvalidasi data *inputan*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu admin kembali *input* data *valid*, kemudian sistem menampilkan daftar periode *Audit* GMP terbaru
- 9) Admin *input* data periode *Audit* GMP, kemudian pilih *button cancel*
- 10) Sistem akan menampilkan kembali daftar periode *Audit* GMP
- 11) Admin kelola data periode *Audit* GMP dengan memilih *edit*
- 12) Sistem menampilkan *form edit* data periode *Audit* GMP
- 13) Admin mengubah data periode *Audit* GMP yang akan di-*edit*
- 14) Admin memilih tombol *submit*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan daftar periode *Audit* GMP terbaru
- 15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data periode *Audit* GMP
- 16) Selesai

i. *Activity Diagram* Periode *Audit Morning Round*



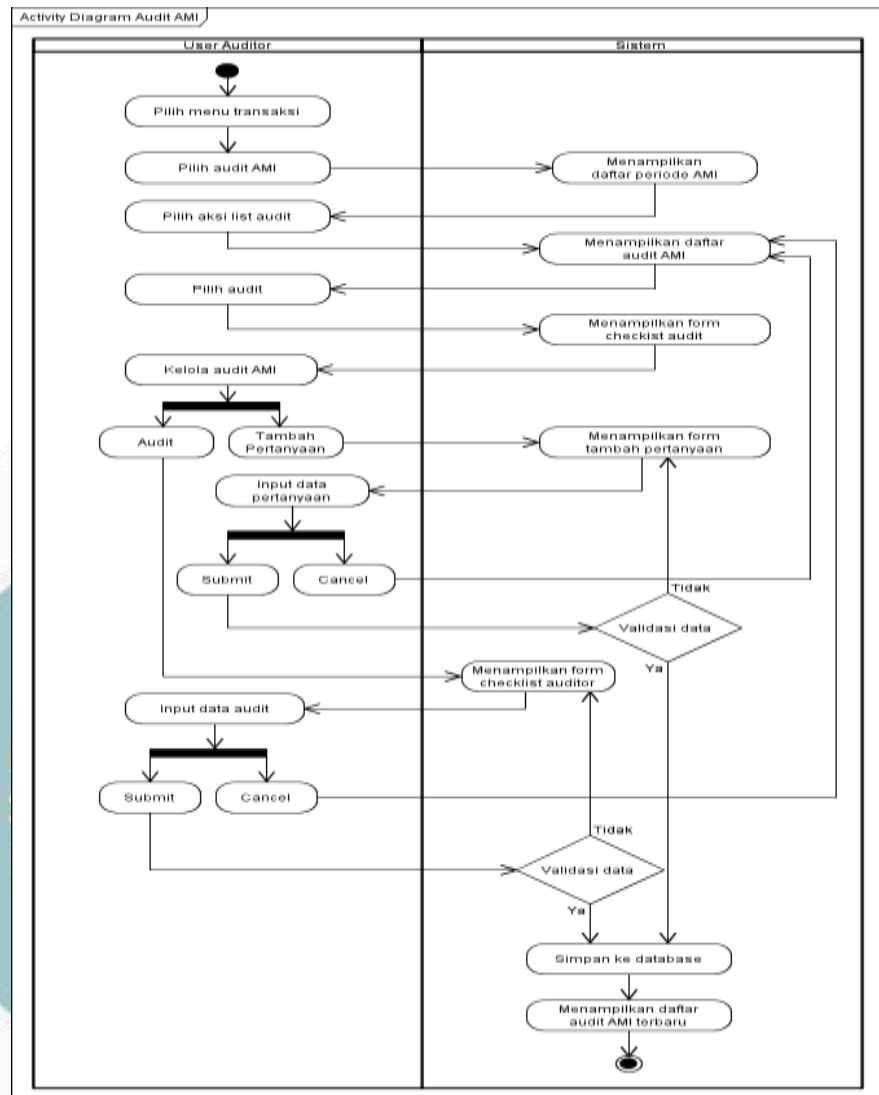
GAMBAR 4.15 Activity Diagram Periode Audit Morning Round

Keterangan:

- 1) Mulai
- 2) Admin memilih menu transaksi
- 3) Admin memilih *Audit MR*
- 4) Sistem menampilkan daftar periode *Audit MR*

- 5) Admin kelola periode *Audit MR* dengan memilih tambah data
- 6) Sistem menampilkan *form* tambah periode *Audit MR*
- 7) Admin *input* data periode *Audit MR*, kemudian pilih *button Submit*
- 8) Sistem akan memvalidasi data *inputan*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu admin kembali *input* data *valid*, kemudian sistem menampilkan daftar periode *Audit MR* terbaru
- 9) Admin *input* data periode *Audit MR*, kemudian pilih *button cancel*
- 10) Sistem akan menampilkan kembali daftar periode *Audit MR*
- 11) Admin kelola data periode *Audit MR* dengan memilih *edit*
- 12) Sistem menampilkan *form edit* data periode *Audit MR*
- 13) Admin mengubah data periode *Audit MR* yang akan di-*edit*
- 14) Admin memilih tombol *submit*, maka sistem akan menyimpan perubahan data ke *database* dan menampilkan daftar periode *Audit MR* terbaru
- 15) Admin memilih tombol *cancel*, maka akan sistem kembali ke tampilan data periode *Audit MR*
- 16) Selesai

j. *Activity Diagram Audit Mutu Internal (AMI)*



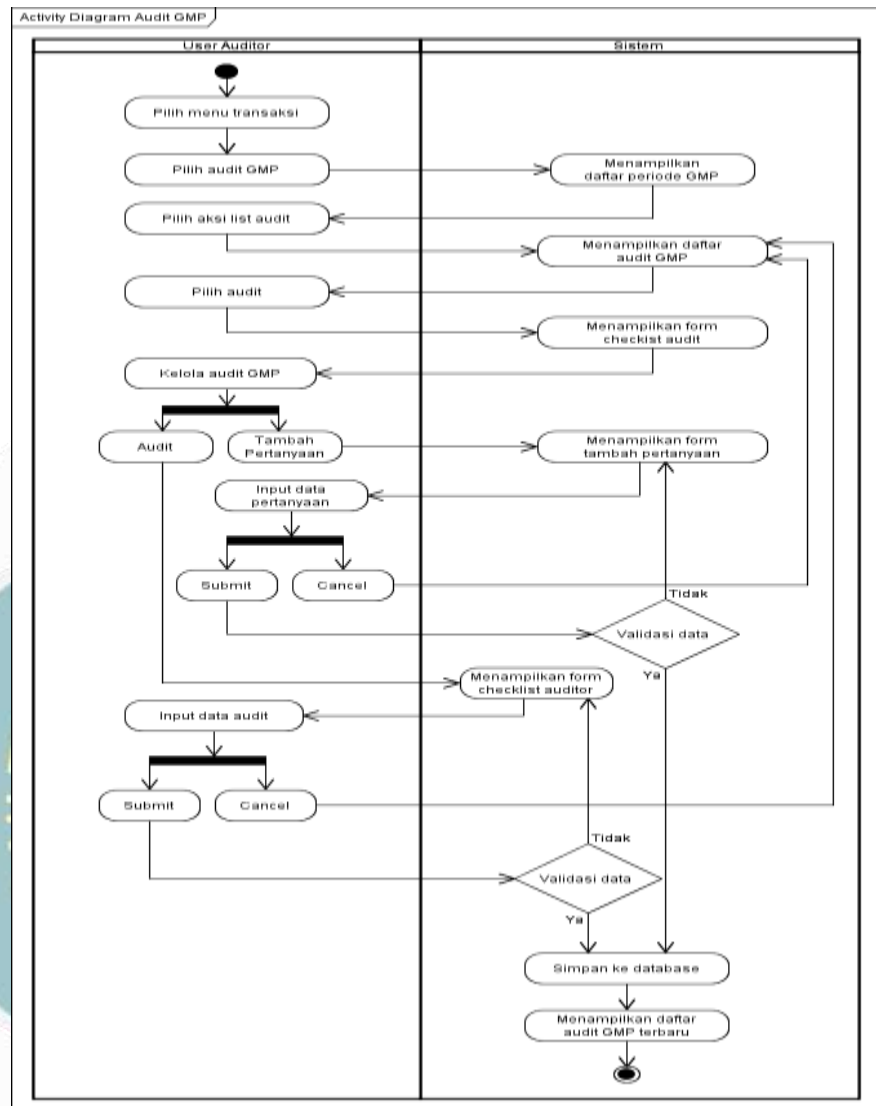
GAMBAR 4.16 *Activity Diagram Audit Mutu Internal (AMI)*

Keterangan:

- 1) Mulai
- 2) *Auditor* memilih menu transaksi
- 3) *Auditor* memilih menu *Audit AMI*
- 4) Sistem menampilkan daftar periode AMI
- 5) *Auditor* memilih *list audit*

- 6) Sistem menampilkan daftar *Audit AMI*
- 7) *Auditor* kelola *audit* dengan memilih tambah pertanyaan
- 8) Sistem menampilkan *form* tambah pertanyaan
- 9) *Auditor* melakukan *input* pertanyaan, kemudian pilih *submit*
- 10) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode AMI terbaru
- 11) *Auditor* melakukan *input* pertanyaan, kemudian pilih *cancel*
- 12) Sistem akan kembali menampilkan daftar AMI
- 13) *Auditor* kelola *audit* dengan memilih aksi *audit*
- 14) Sistem menampilkan *form checklist audit*
- 15) *Auditor* melakukan *input audit*, kemudian pilih *submit*
- 16) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode AMI terbaru
- 17) *Auditor* melakukan *input audit*, kemudian pilih *cancel*
- 18) Sistem akan kembali menampilkan daftar AMI
- 19) Selesai

k. *Activity Diagram Audit GMP*



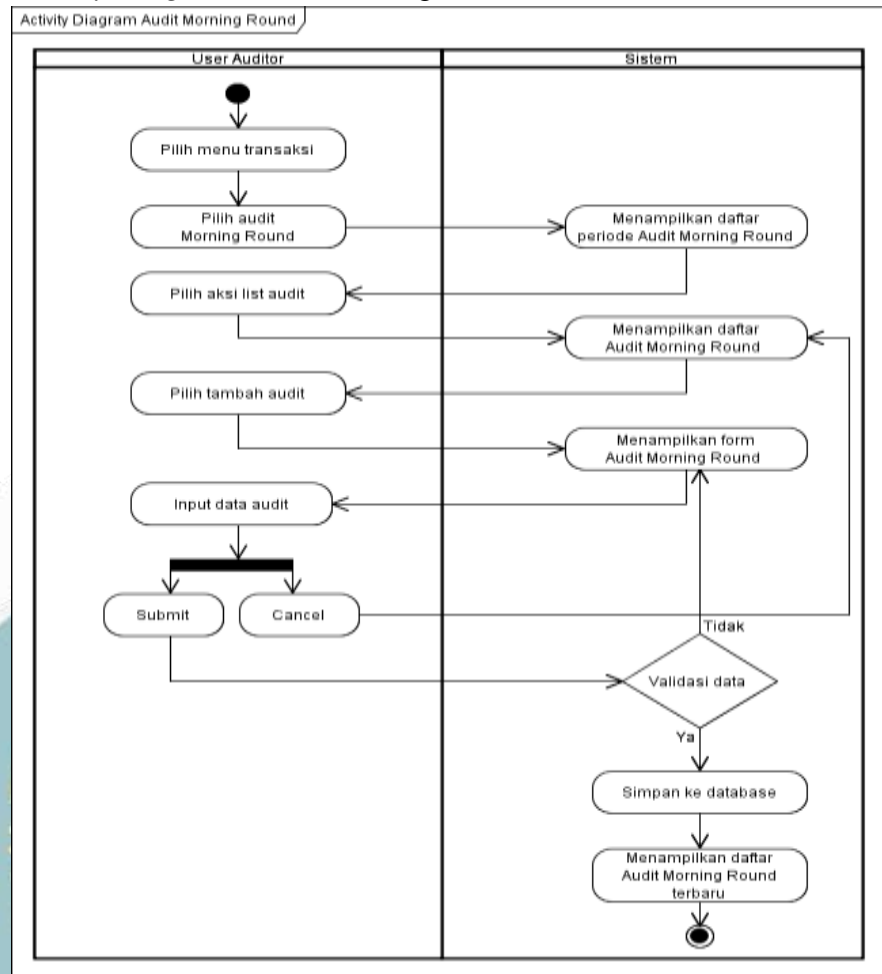
GAMBAR 4.17 *Activity Diagram Audit GMP*

Keterangan:

- 1) Mulai
- 2) *Auditor* memilih menu transaksi
- 3) *Auditor* memilih menu *Audit GMP*
- 4) Sistem menampilkan daftar periode *Audit GMP*
- 5) *Auditor* memilih *list audit*

- 6) Sistem menampilkan daftar *Audit* GMP
- 7) *Auditor* kelola *audit* dengan memilih tambah pertanyaan
- 8) Sistem menampilkan *form* tambah pertanyaan
- 9) *Auditor* melakukan *input* pertanyaan, kemudian pilih *submit*
- 10) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode *Audit* GMP terbaru
- 11) *Auditor* melakukan *input* pertanyaan, kemudian pilih *cancel*
- 12) Sistem akan kembali menampilkan daftar AMI
- 13) *Auditor* kelola *audit* dengan memilih aksi *audit*
- 14) Sistem menampilkan *form checklist audit*
- 15) *Auditor* melakukan *input audit*, kemudian pilih *submit*
- 16) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode *Audit* GMP terbaru
- 17) *Auditor* melakukan *input audit*, kemudian pilih *cancel*
- 18) Sistem akan kembali menampilkan daftar *Audit* GMP
- 19) Selesai

1. Activity Diagram Audit Morning Round



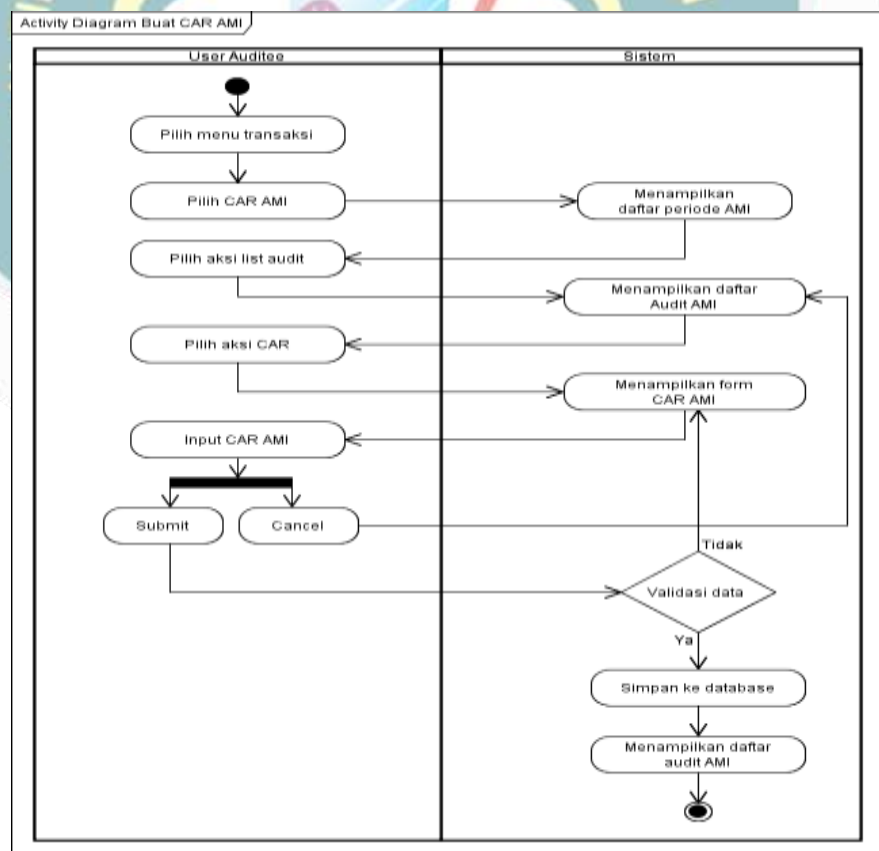
GAMBAR 4.18 Activity Diagram Audit Morning Round

Keterangan:

- 1) Mulai
- 2) Auditor memilih menu transaksi
- 3) Auditor memilih menu Audit Morning Round
- 4) Sistem menampilkan daftar periode Audit Morning Round
- 5) Auditor memilih aksi list audit
- 6) Sistem menampilkan daftar Audit Morning Round
- 7) Auditor memilih tambah audit

- 8) Sistem menampilkan *form Audit Morning Round*
- 9) *Auditor* melakukan *input* hasil audit, kemudian pilih *submit*
- 10) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode *Audit Morning Round* terbaru
- 11) *Auditor* melakukan *input* hasil audit, kemudian pilih *cancel*
- 12) Sistem akan kembali menampilkan daftar *Audit Morning Round*
- 13) Selesai

m. *Activity Diagram* Membuat CAR AMI

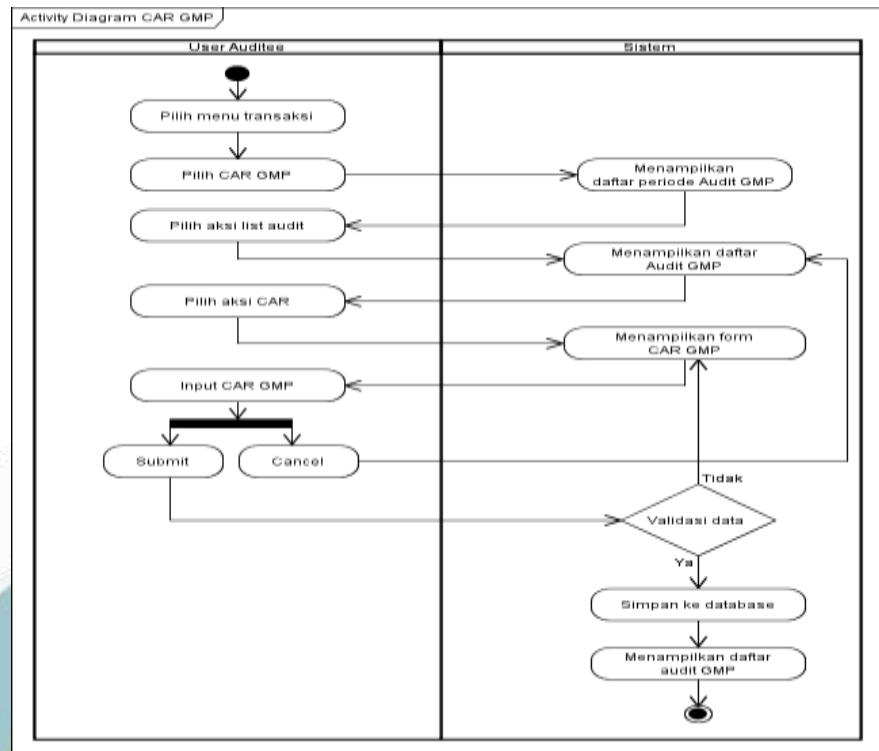


GAMBAR 4.19 *Activity Diagram* Membuat CAR AMI

Keterangan:

- 1) Mulai
- 2) *Auditee* memilih menu transaksi
- 3) *Auditee* memilih menu CAR AMI
- 4) Sistem menampilkan daftar periode AMI
- 5) *Auditee* memilih aksi *list audit*
- 6) Sistem menampilkan daftar AMI
- 7) *Auditee* memilih aksi CAR
- 8) Sistem menampilkan *form* CAR AMI
- 9) *Auditee* melakukan *input* CAR AMI, kemudian pilih *submit*
- 10) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode AMI
- 11) *Auditor* melakukan *input* CAR AMI, kemudian pilih *cancel*
- 12) Sistem akan kembali menampilkan daftar AMI
- 13) Selesai

n. *Activity Diagram Membuat CAR Audit GMP*



GAMBAR 4.20 *Activity Diagram Membuat CAR Audit GMP*

Keterangan:

- 1) Mulai
- 2) *Auditee* memilih menu transaksi
- 3) *Auditee* memilih menu CAR GMP
- 4) Sistem menampilkan daftar periode *Audit GMP*
- 5) *Auditee* memilih aksi *list audit*
- 6) Sistem menampilkan daftar *Audit GMP*
- 7) *Auditee* memilih aksi CAR
- 8) Sistem menampilkan *form* CAR GMP
- 9) *Auditee* melakukan *input* CAR GMP, kemudian pilih *submit*

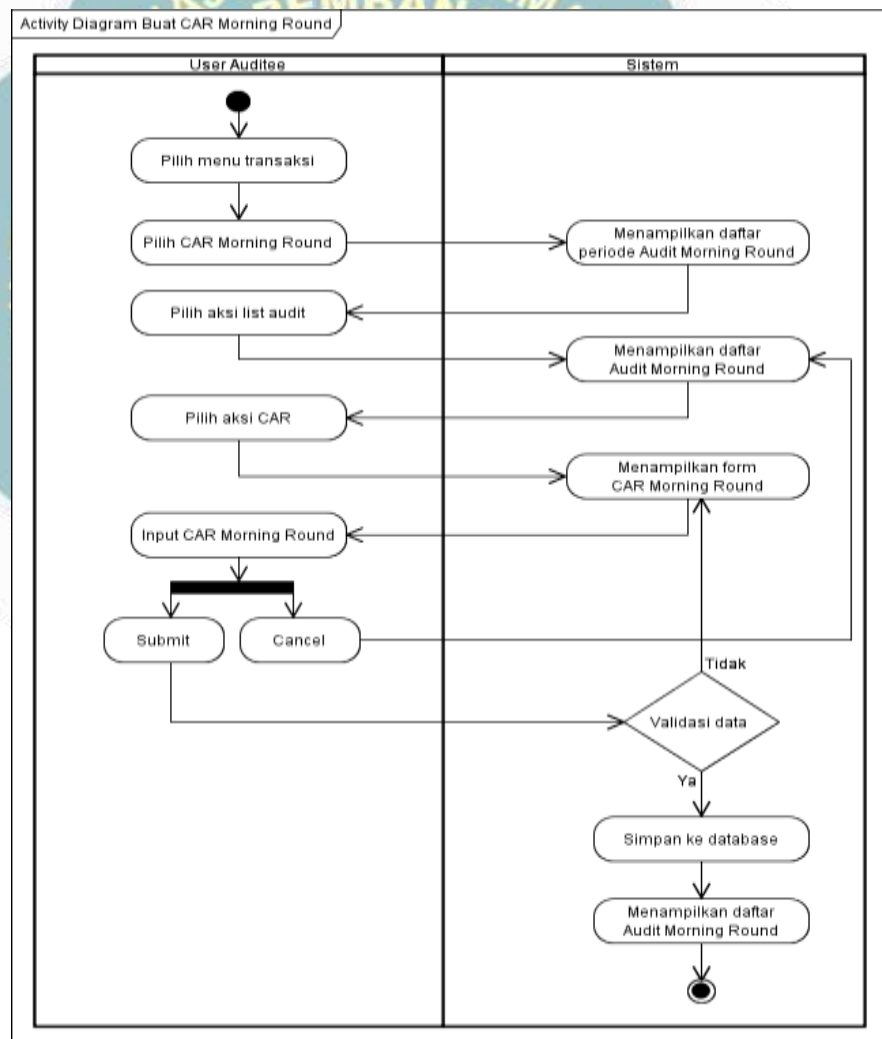
10) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode GMP

11) *Auditor* melakukan *input* CAR GMP, kemudian pilih *cancel*

12) Sistem akan kembali menampilkan daftar GMP

13) Selesai

o. *Activity Diagram* Membuat CAR Audit Morning Round

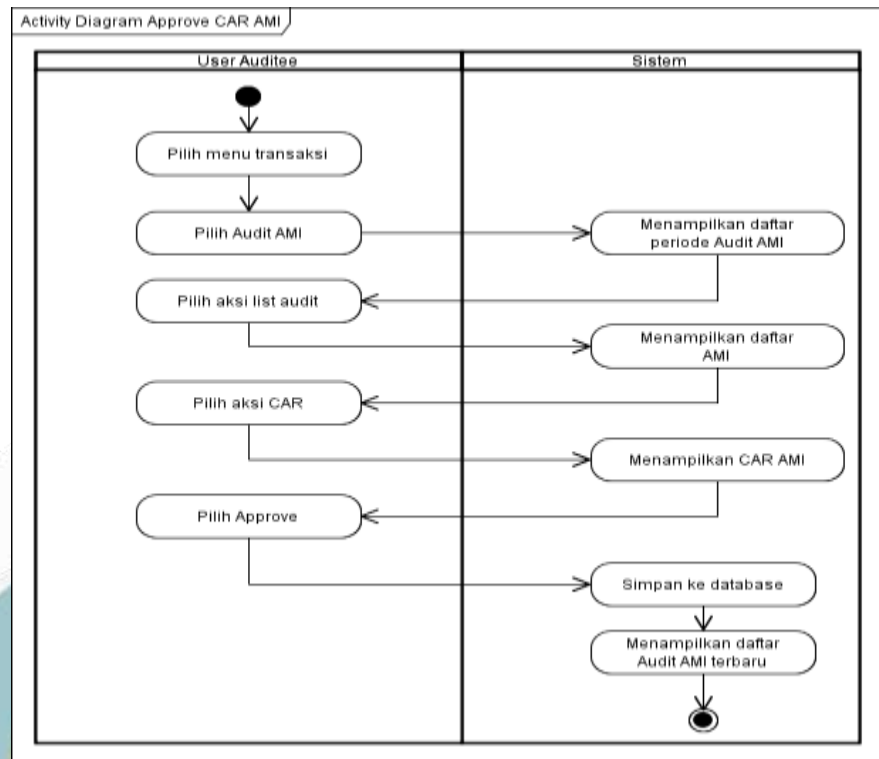


GAMBAR 4.21 *Activity Diagram* Membuat CAR Audit Morning Round

Keterangan:

- 1) Mulai
- 2) *Auditee* memilih menu transaksi
- 3) *Auditee* memilih menu *CAR Morning Round*
- 4) Sistem menampilkan daftar periode *Audit Morning Round*
- 5) *Auditee* memilih aksi *list audit*
- 6) Sistem menampilkan daftar *Audit Morning Round*
- 7) *Auditee* memilih aksi CAR
- 8) Sistem menampilkan *form CAR Morning Round*
- 9) *Auditee* melakukan *input CAR Morning Round*, kemudian pilih *submit*
- 10) Sistem akan validasi data *input*, jika *valid* sistem akan menyimpan ke *database*, jika tidak *valid* maka sistem akan menampilkan pesan lalu *auditor* kembali *input* data *valid*, kemudian sistem menampilkan daftar periode *Morning Round*
- 11) *Auditor* melakukan *input CAR Morning Round*, kemudian pilih *cancel*
- 12) Sistem akan kembali menampilkan daftar *Morning Round*
- 13) Selesai

p. *Activity Diagram Approve CAR AMI*



GAMBAR 4.22 *Activity Diagram Approve CAR AMI*

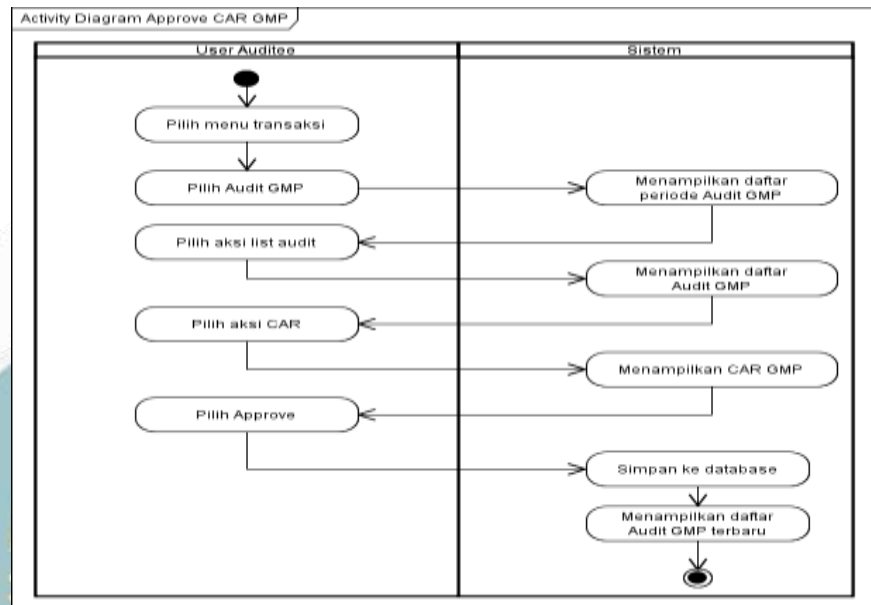
Keterangan:

- 1) Mulai
- 2) Auditor memilih menu transaksi
- 3) Auditor memilih menu *Audit AMI*
- 4) Sistem menampilkan daftar periode AMI
- 5) Auditor memilih aksi *list audit*
- 6) Sistem menampilkan daftar AMI
- 7) Auditor memilih aksi CAR
- 8) Sistem menampilkan CAR AMI
- 9) Auditor memilih *approve*

10) Sistem menyimpan data ke *database* dan kembali menampilkan daftar AMI terbaru

11) Selesai

q. *Activity Diagram Approve CAR GMP*



GAMBAR 4.23 *Activity Diagram Approve CAR GMP*

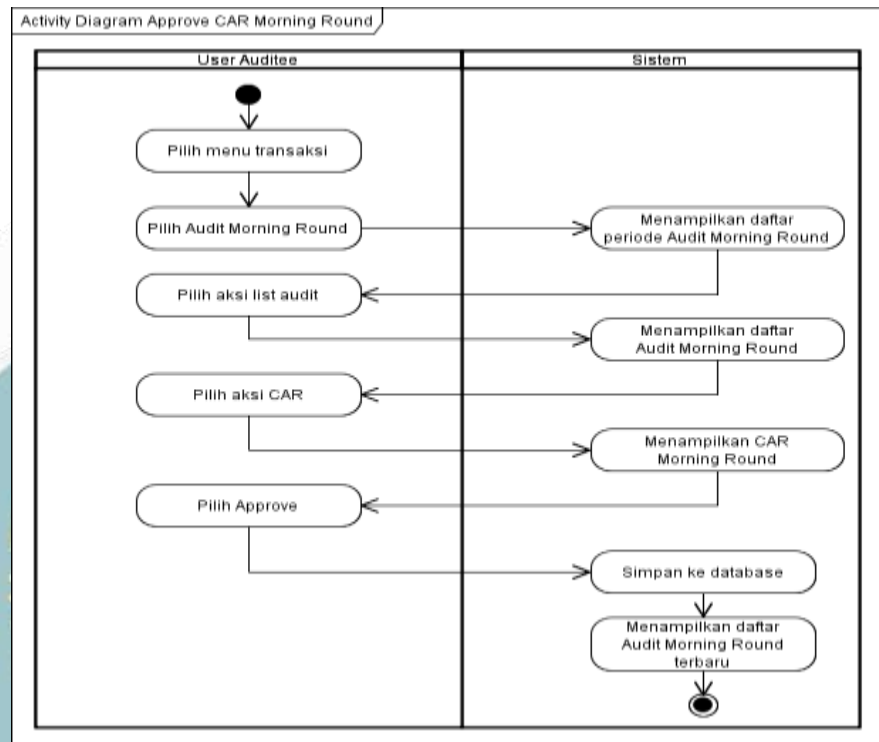
Keterangan:

- 1) Mulai
- 2) *Auditor* memilih menu transaksi
- 3) *Auditor* memilih menu *Audit GMP*
- 4) Sistem menampilkan daftar periode *Audit GMP*
- 5) *Auditor* memilih aksi *list audit*
- 6) Sistem menampilkan daftar *Audit GMP*
- 7) *Auditor* memilih aksi *CAR*
- 8) Sistem menampilkan *CAR Audit GMP*
- 9) *Auditor* memilih *approve*

10) Sistem menyimpan data ke *database* dan kembali menampilkan daftar *Audit GMP* terbaru

11) Selesai

r. *Activity Diagram Approve CAR Morning Round*



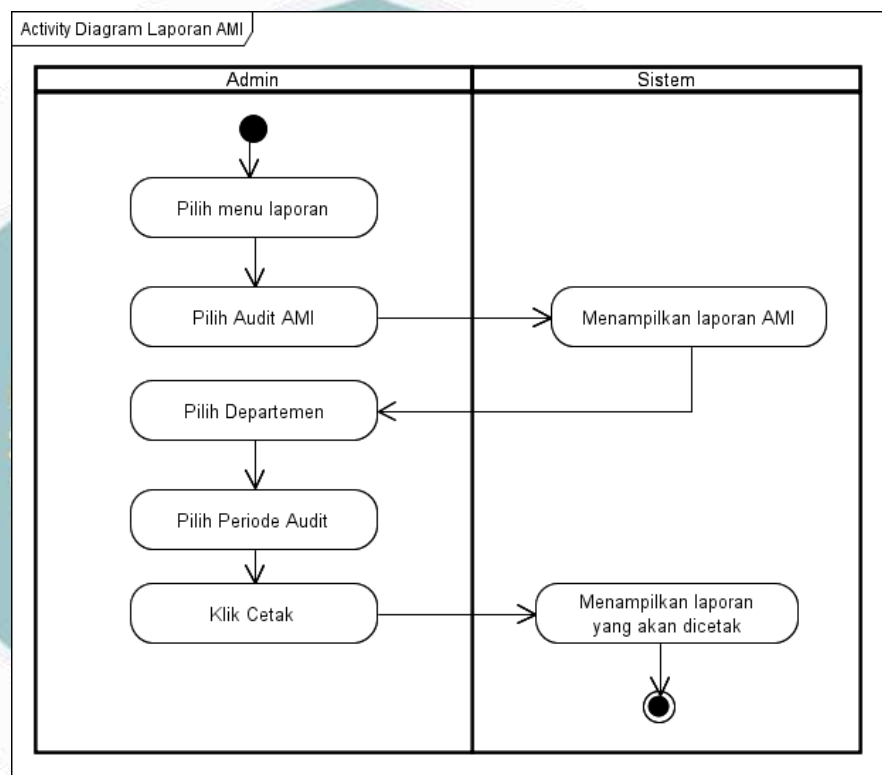
GAMBAR 4.24 *Activity Diagram Approve CAR Morning Round*

Keterangan:

- 1) Mulai
- 2) Auditor memilih menu transaksi
- 3) Auditor memilih menu *Audit Morning Round*
- 4) Sistem menampilkan daftar periode *Audit Morning Round*
- 5) Auditor memilih aksi *list audit*
- 6) Sistem menampilkan daftar *Audit Morning Round*
- 7) Auditor memilih aksi CAR

- 8) Sistem menampilkan CAR *Audit Morning Round*
- 9) Auditor memilih *approve*
- 10) Sistem menyimpan data ke *database* dan kembali menampilkan daftar *Audit Morning Round* terbaru
- 11) Selesai

s. *Activity Diagram* Laporan AMI



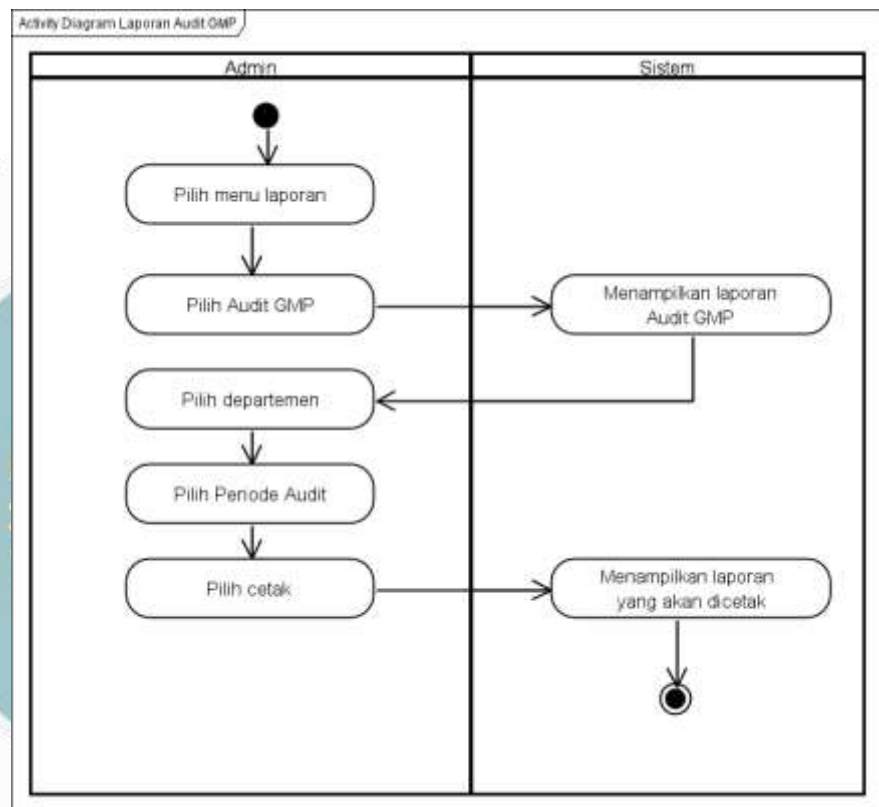
GAMBAR 4.25 *Activity Diagram* Laporan AMI

Keterangan:

- 1) Mulai
- 2) Admin memilih menu laporan
- 3) Admin memilih menu *Audit AMI*
- 4) Sistem menampilkan laporan *Audit AMI*

- 5) Admin memilih departemen dan periode laporan
- 6) Admin memilih cetak
- 7) Sistem akan menampilkan laporan yang akan di cetak
- 8) Selesai

t. *Activity Diagram* Laporan GMP



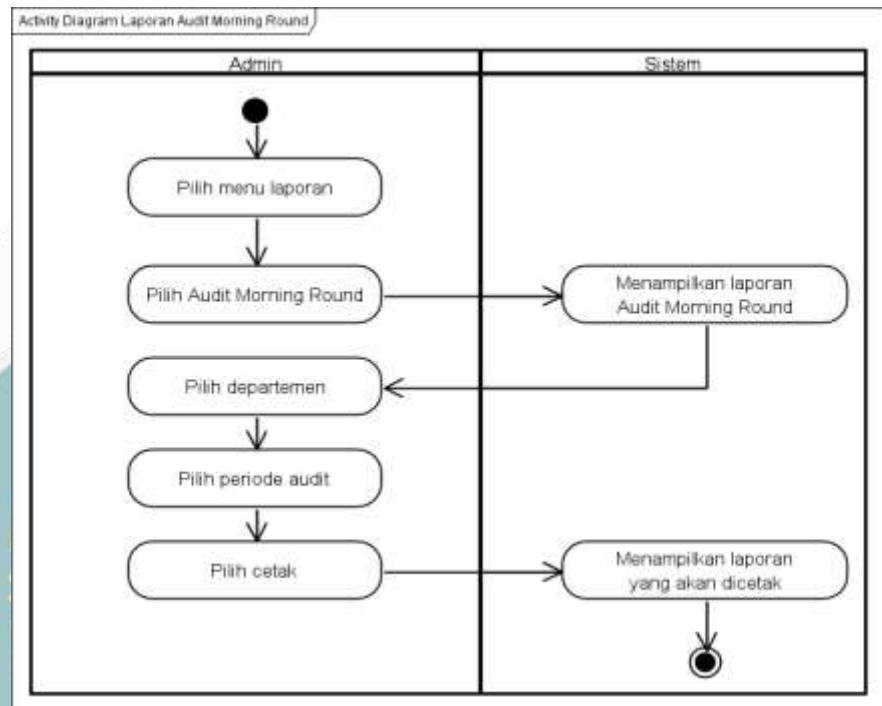
GAMBAR 4.26 *Activity Diagram* Laporan Audit GMP

Keterangan:

- 1) Mulai
- 2) Admin memilih menu laporan
- 3) Admin memilih menu *Audit* GMP
- 4) Sistem menampilkan laporan *Audit* GMP
- 5) Admin memilih departemen dan periode audit

- 6) Admin memilih cetak
- 7) Sistem akan menampilkan laporan yang akan dicetak
- 8) Selesai

u. *Activity Diagram Laporan Audit Morning Round*



GAMBAR 4.27 *Activity Diagram* Laporan Audit Morning Round

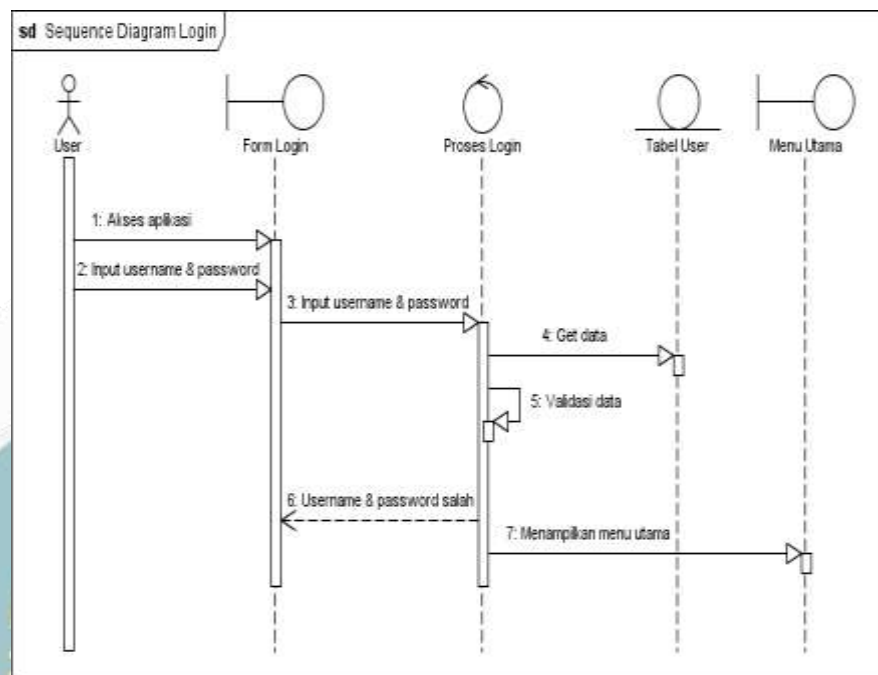
Keterangan:

- 1) Mulai
- 2) Admin memilih menu laporan
- 3) Admin memilih menu *Audit Morning Round*
- 4) Sistem menampilkan laporan *Audit Morning Round*
- 5) Admin memilih departemen dan periode audit
- 6) Admin memilih cetak
- 7) Sistem akan menampilkan laporan yang akan dicetak
- 8) Selesai

4.2.3. Sequence Diagram

Berikut adalah *sequence diagram* yang diusulkan :

a. Sequence Diagram Login

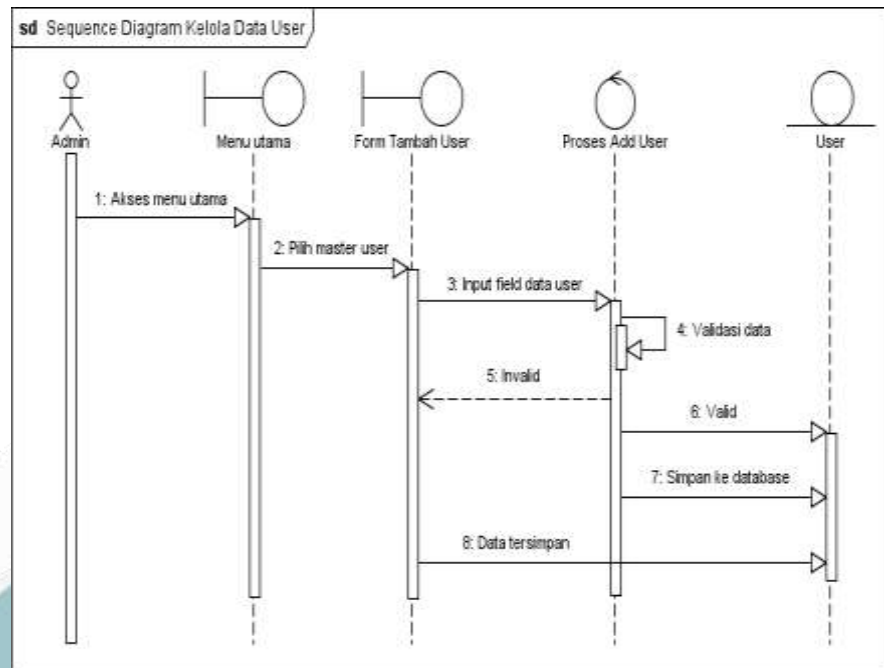


GAMBAR 4.28 Sequence Diagram Login

Keterangan :

- User* mengakses aplikasi dengan tampilan awal berupa *form login*.
- User* memasukan *username* dan *password*.
- Proses validasi, jika benar maka sistem akan menampilkan halaman menu utama, jika salah maka *user* kembali *input username* dan *password*.

b. *Sequence Diagram Kelola Data User*

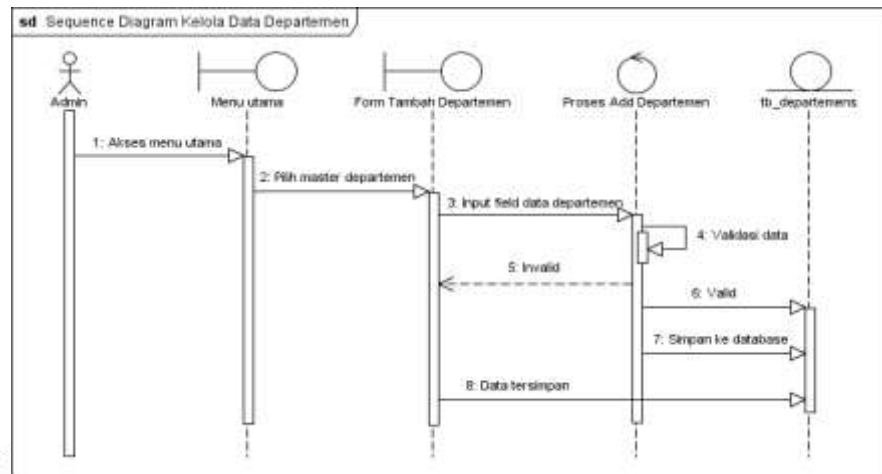


GAMBAR 4.29 *Sequence Diagram Kelola Data User*

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih master *user*
- 3) Admin melakukan *input* data *user*
- 4) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid, jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

c. *Sequence Diagram* Kelola Data Departemen

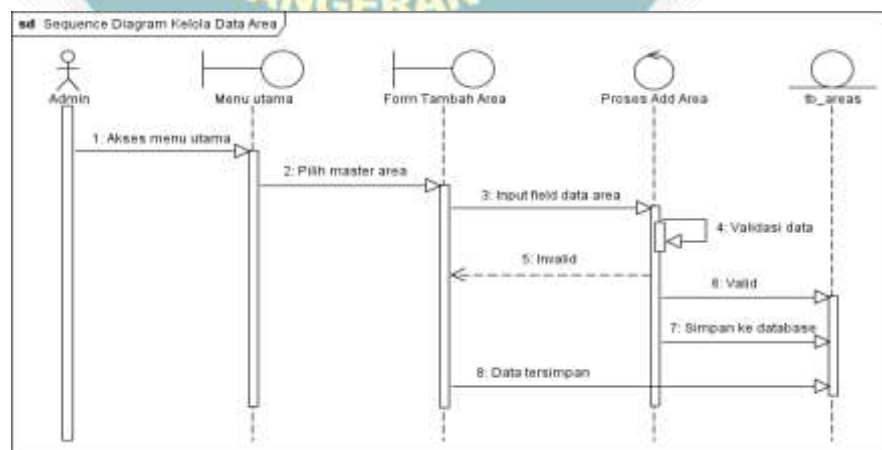


GAMBAR 4.30 *Sequence Diagram* Kelola Data Departemen

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih master departemen
- 3) Admin melakukan *input* data departemen
- 4) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

d. *Sequence Diagram* Kelola Data Area

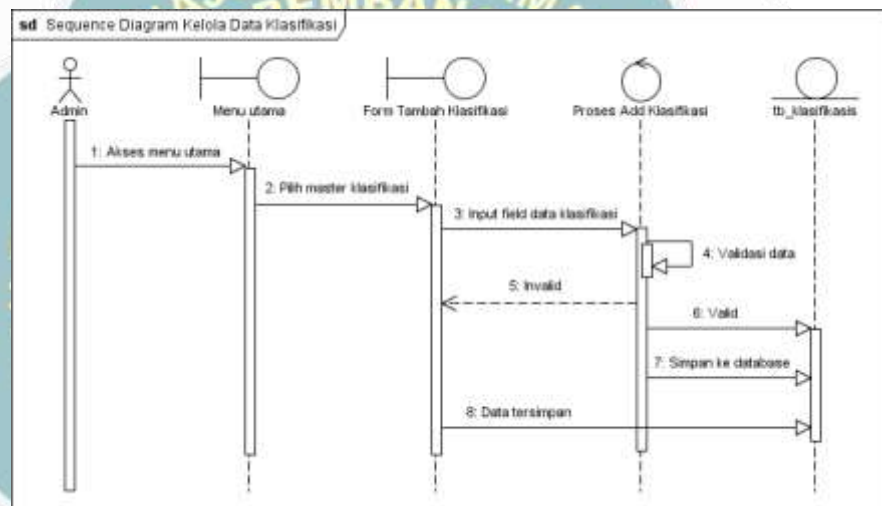


GAMBAR 4.31 *Sequence Diagram* Kelola Data Area

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih master area
- 3) Admin melakukan *input* data area
- 4) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

e. *Sequence Diagram* Kelola Data Klasifikasi

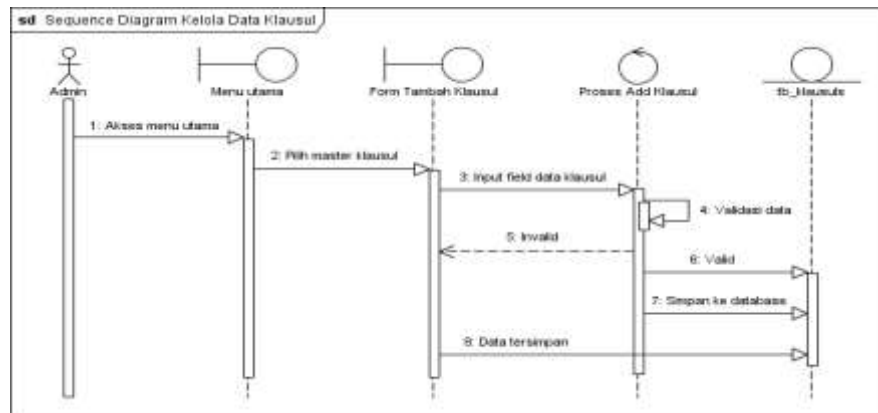


GAMBAR 4.32 *Sequence Diagram* Kelola Data Klasifikasi

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih master klasifikasi
- 3) Admin melakukan *input* data klasifikasi
- 4) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

f. *Sequence Diagram* Kelola Data Klausul

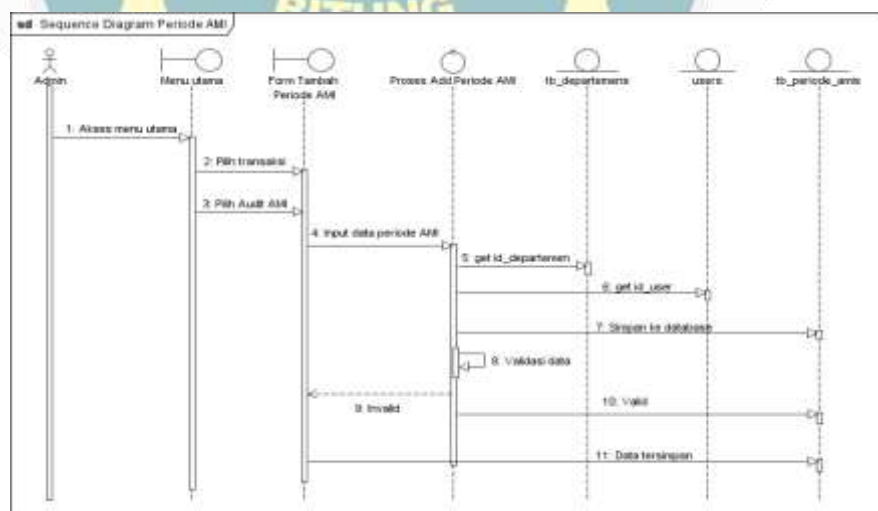


GAMBAR 4.33 *Sequence Diagram* Kelola Data Klausul

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih master klausul
- 3) Admin melakukan *input* data klausul
- 4) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

g. *Sequence Diagram* Periode Audit Mutu Internal (AMI)

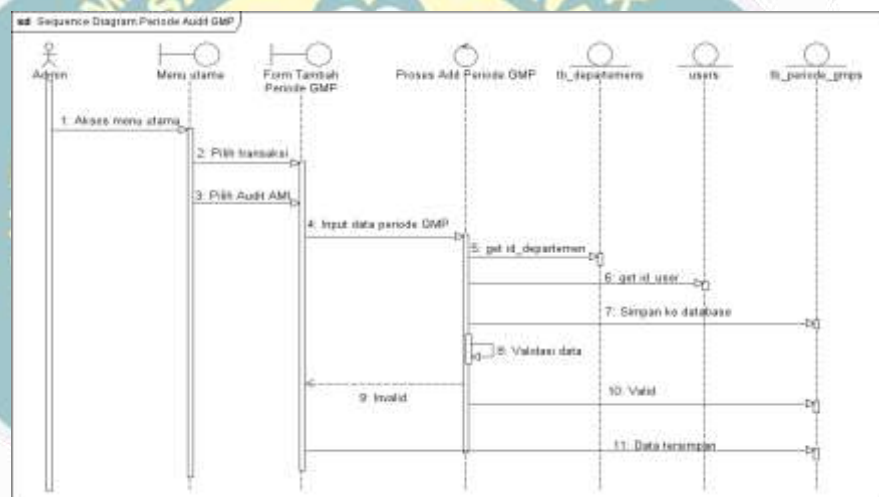


GAMBAR 4.34 *Sequence Diagram* Periode AMI

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih transaksi
- 3) Admin memilih audit AMI
- 4) Admin melakukan *input* data periode AMI
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

h. *Sequence Diagram* Periode Audit GMP



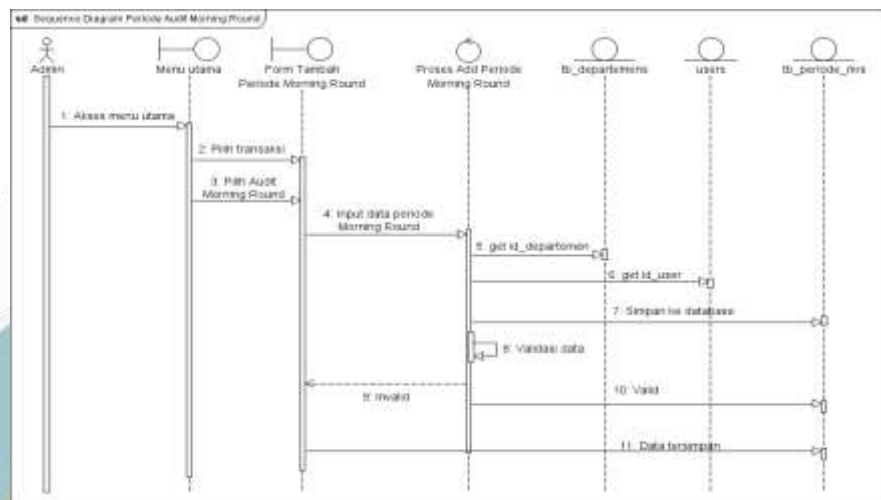
GAMBAR 4.35 *Sequence Diagram* Periode GMP

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih transaksi
- 3) Admin memilih audit GMP
- 4) Admin melakukan *input* data periode audit GMP

- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

i. *Sequence Diagram* Periode Audit Morning Round

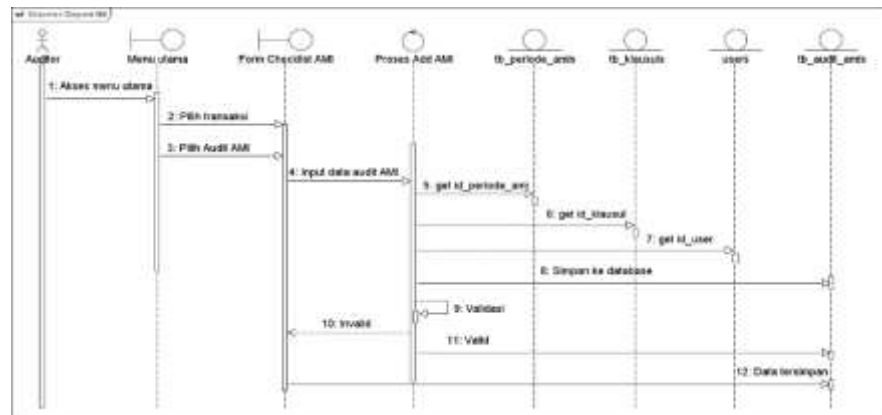


GAMBAR 4.36 *Sequence Diagram* Periode Morning Round

Keterangan :

- 1) Admin mengakses aplikasi menu utama
- 2) Admin memilih transaksi
- 3) Admin memilih audit *Morning Round*
- 4) Admin melakukan *input* data periode audit *Morning Round*
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

j. *Sequence Diagram Audit Mutu Internal (AMI)*

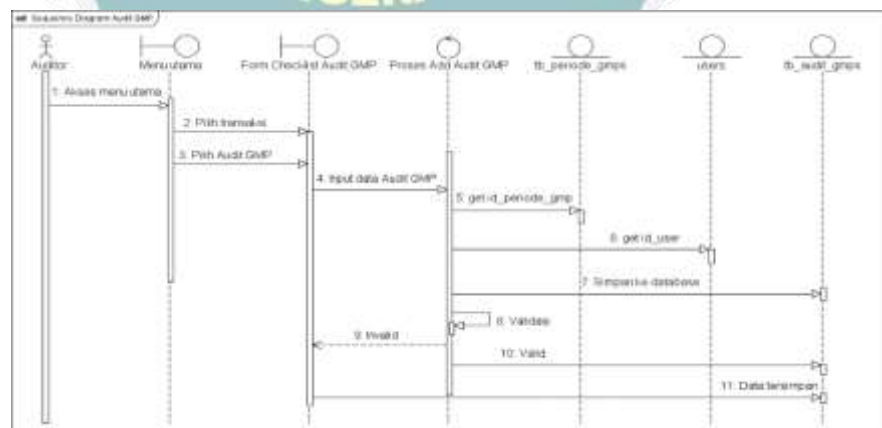


GAMBAR 4.37 *Sequence Diagram AMI*

Keterangan :

- 1) Auditor mengakses aplikasi menu utama.
- 2) Auditor memilih transaksi.
- 3) Auditor memilih Audit AMI.
- 4) Auditor melakukan *input* data Audit Mutu Internal (AMI)
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

k. *Sequence Diagram Audit Good Manufacturing Practices (GMP)*

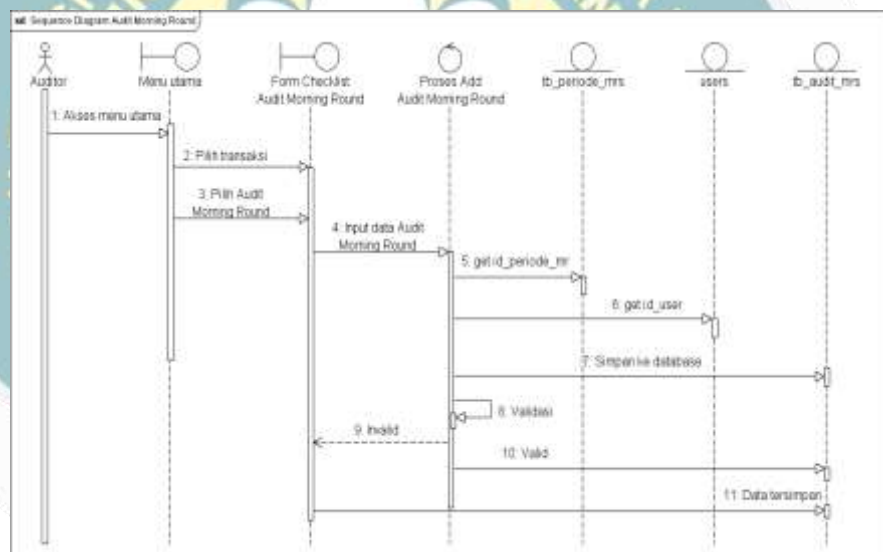


GAMBAR 4.38 *Sequence Diagram Audit GMP*

Keterangan :

- 1) *Auditor* mengakses aplikasi menu utama.
- 2) *Auditor* memilih transaksi.
- 3) *Auditor* memilih *Audit GMP*.
- 4) *Auditor* melakukan *input* data *Audit GMP*.
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

1. *Sequence Diagram Audit Morning Round*



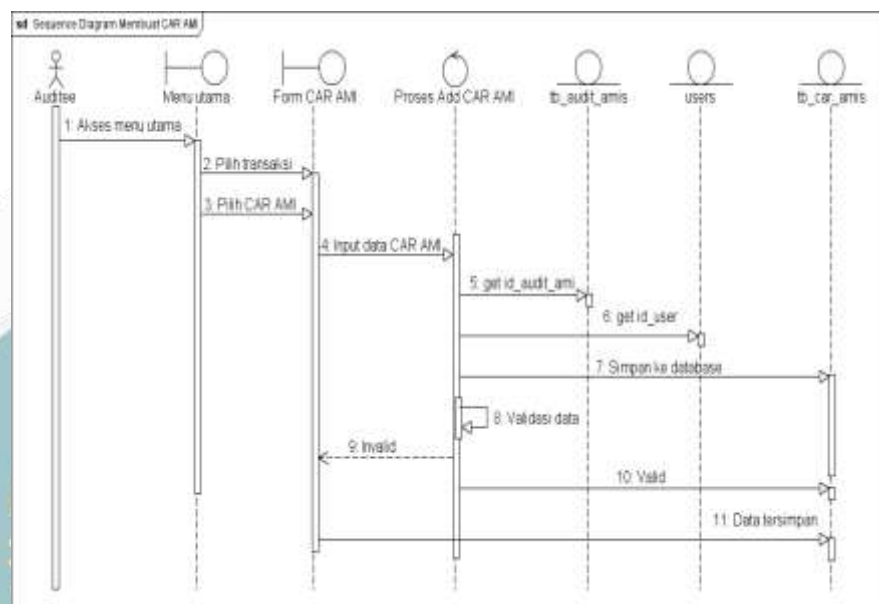
GAMBAR 4.39 *Sequence Diagram Audit Morning Round*

Keterangan :

- 1) *Auditor* mengakses aplikasi menu utama.
- 2) *Auditor* memilih transaksi.
- 3) *Auditor* memilih *Audit Morning Round*.
- 4) *Auditor* melakukan *input* data *Audit Morning Round*.

- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

m. *Sequence Diagram* Membuat CAR AMI

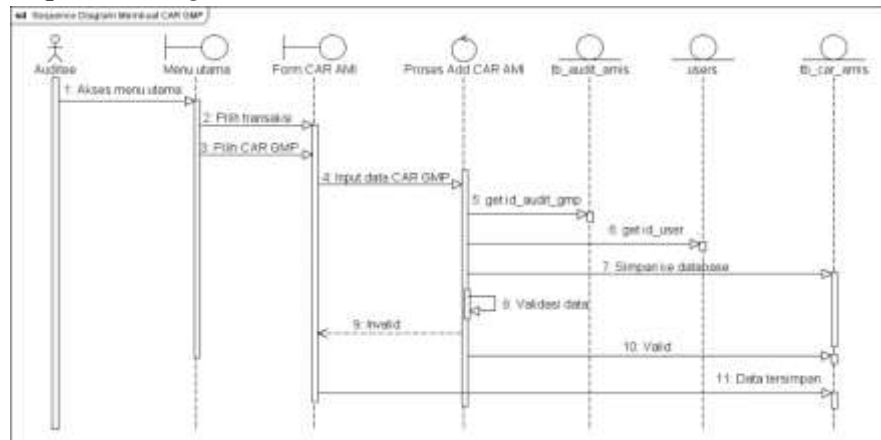


GAMBAR 4.40 *Sequence Diagram* Membuat CAR AMI

Keterangan :

- 1) *Auditee* mengakses aplikasi menu utama.
- 2) *Auditee* memilih transaksi.
- 3) *Auditee* memilih CAR AMI.
- 4) *Auditee* melakukan *input* CAR AMI.
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

n. *Sequence Diagram Membuat CAR Audit GMP*

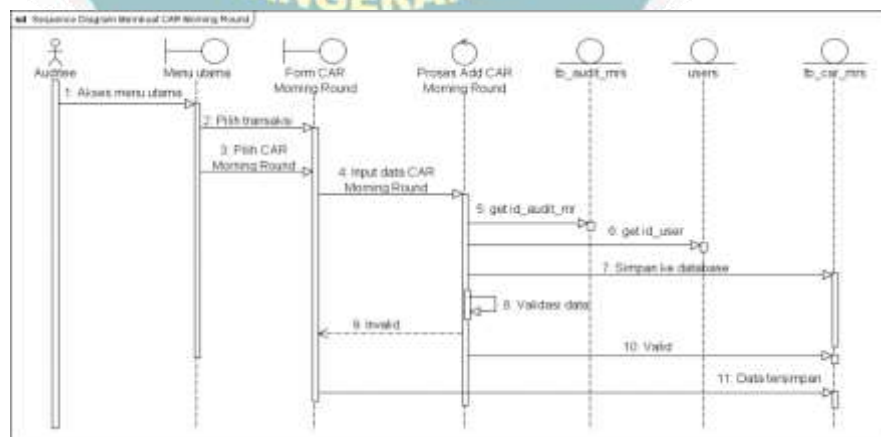


GAMBAR 4.41 *Sequence Diagram* Membuat CAR Audit GMP

Keterangan :

- 1) *Auditee* mengakses aplikasi menu utama.
- 2) *Auditee* memilih transaksi.
- 3) *Auditee* memilih CAR GMP.
- 4) *Auditee* melakukan *input* CAR GMP.
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

o. *Sequence Diagram Membuat CAR Morning Round*

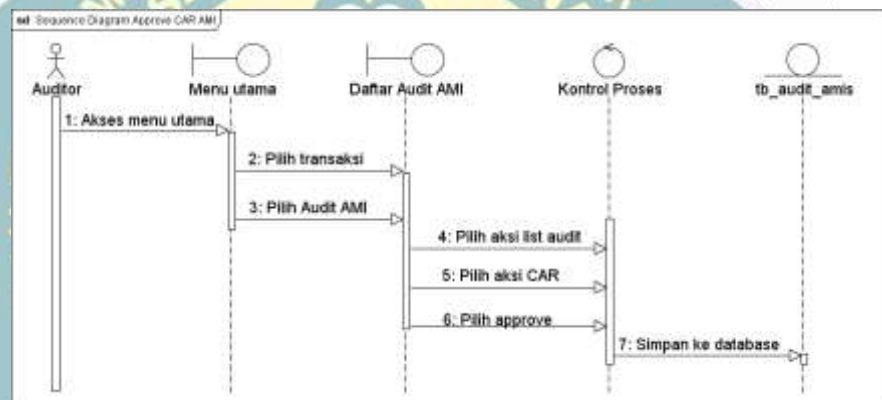


GAMBAR 4.42 *Sequence Diagram* Membuat CAR Morning Round

Keterangan :

- 1) *Auditee* mengakses aplikasi menu utama.
- 2) *Auditee* memilih transaksi.
- 3) *Auditee* memilih CAR Morning Round.
- 4) *Auditee* melakukan *input* CAR Morning Round.
- 5) Proses validasi, jika data tidak valid maka sistem akan menampilkan pesan data tidak valid jika data valid maka sistem akan menyimpan ke *database* dan menampilkan pesan berhasil.

p. *Sequence Diagram Approve CAR AMI*

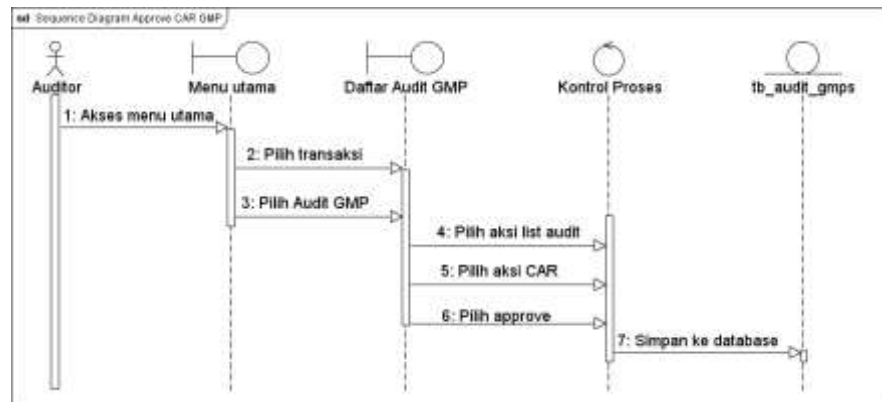


GAMBAR 4.43 *Sequence Diagram Approve CAR AMI*

Keterangan :

- 1) *Auditor* mengakses aplikasi menu utama.
- 2) *Auditor* memilih transaksi, kemudian pilih *Audit* AMI.
- 3) *Auditor* memilih aksi *list audit*, pilih aksi CAR AMI, lalu pilih approve.
- 4) Sistem memproses penyimpanan data.

q. *Sequence Diagram Approve CAR GMP*

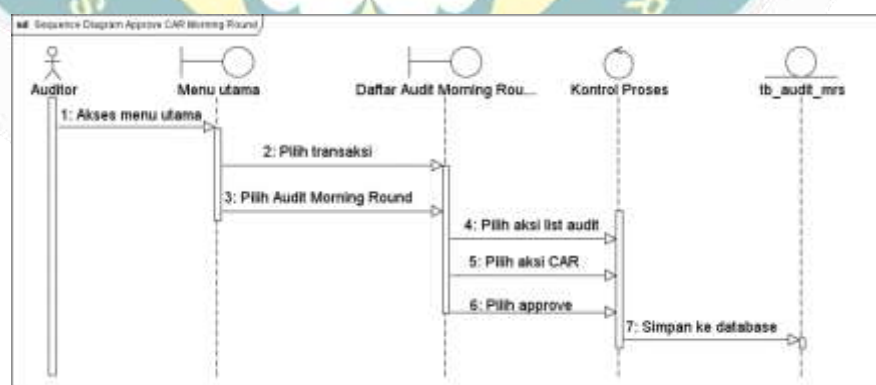


GAMBAR 4.44 *Sequence Diagram Approve CAR GMP*

Keterangan :

- 1) *Auditor* mengakses aplikasi menu utama.
- 2) *Auditor* memilih transaksi, kemudian pilih *Audit GMP*.
- 3) *Auditor* memilih aksi *list audit*, pilih aksi *CAR GMP*, lalu pilih approve.
- 4) Sistem memproses penyimpanan data.

r. *Sequence Diagram Approve CAR Morning Round*



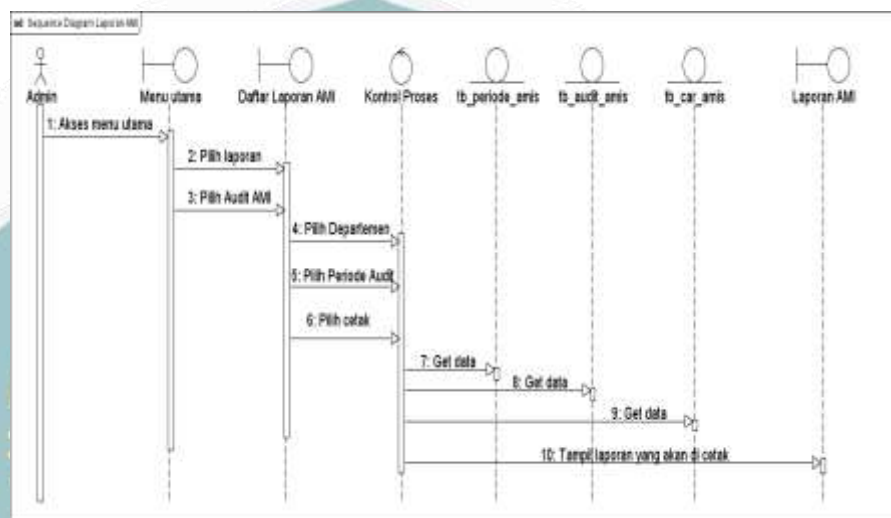
GAMBAR 4.45 *Sequence Diagram Approve CAR Morning Round*

Keterangan :

- 1) *Auditor* mengakses aplikasi menu utama.

- 2) Auditor memilih transaksi, kemudian pilih *Audit Morning Round*.
- 3) Auditor memilih aksi *list audit*, pilih aksi *CAR Morning Round*, lalu pilih approve.
- 4) Sistem memproses penyimpanan data.

s. *Sequence Diagram* Laporan AMI

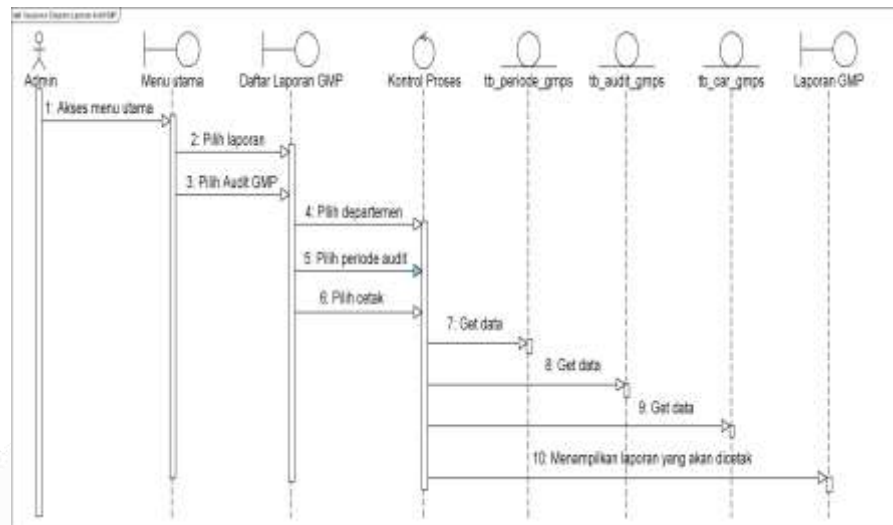


GAMBAR 4.46 *Sequence Diagram* Laporan AMI

Keterangan :

- 1) Admin mengakses aplikasi menu utama.
- 2) Admin memilih laporan, kemudian pilih *Audit AMI*.
- 3) Admin memilih departemen dan periode audit, kemudian pilih cetak
- 4) Sistem memproses penarikan data dan menampilkan laporan yang akan dicetak

t. *Sequence Diagram Laporan Audit GMP*

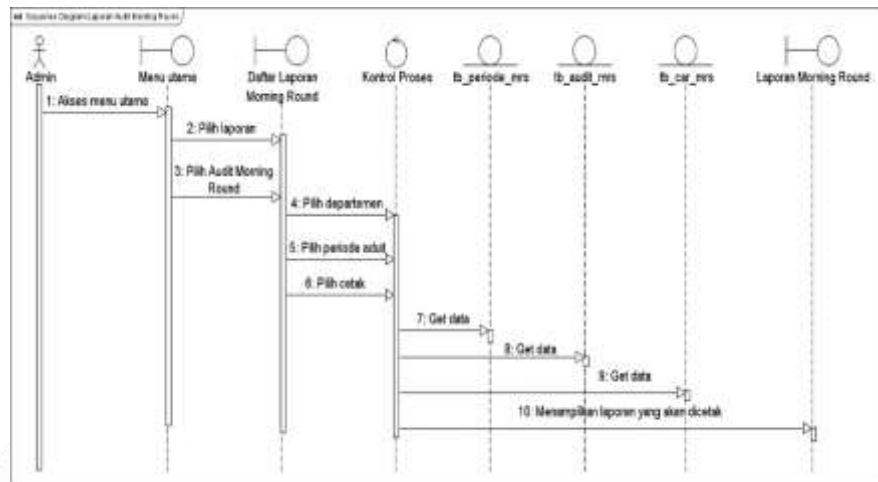


GAMBAR 4.47 *Sequence Diagram Laporan Audit GMP*
Berdasarkan Periode Audit

Keterangan :

- 1) Admin mengakses aplikasi menu utama.
- 2) Admin memilih laporan, kemudian pilih *Audit GMP*.
- 3) Admin memilih departemen dan periode audit, kemudian pilih cetak
- 4) Sistem memproses penarikan data dan menampilkan laporan yang akan dicetak

u. *Sequence Diagram Laporan Audit Morning Round*

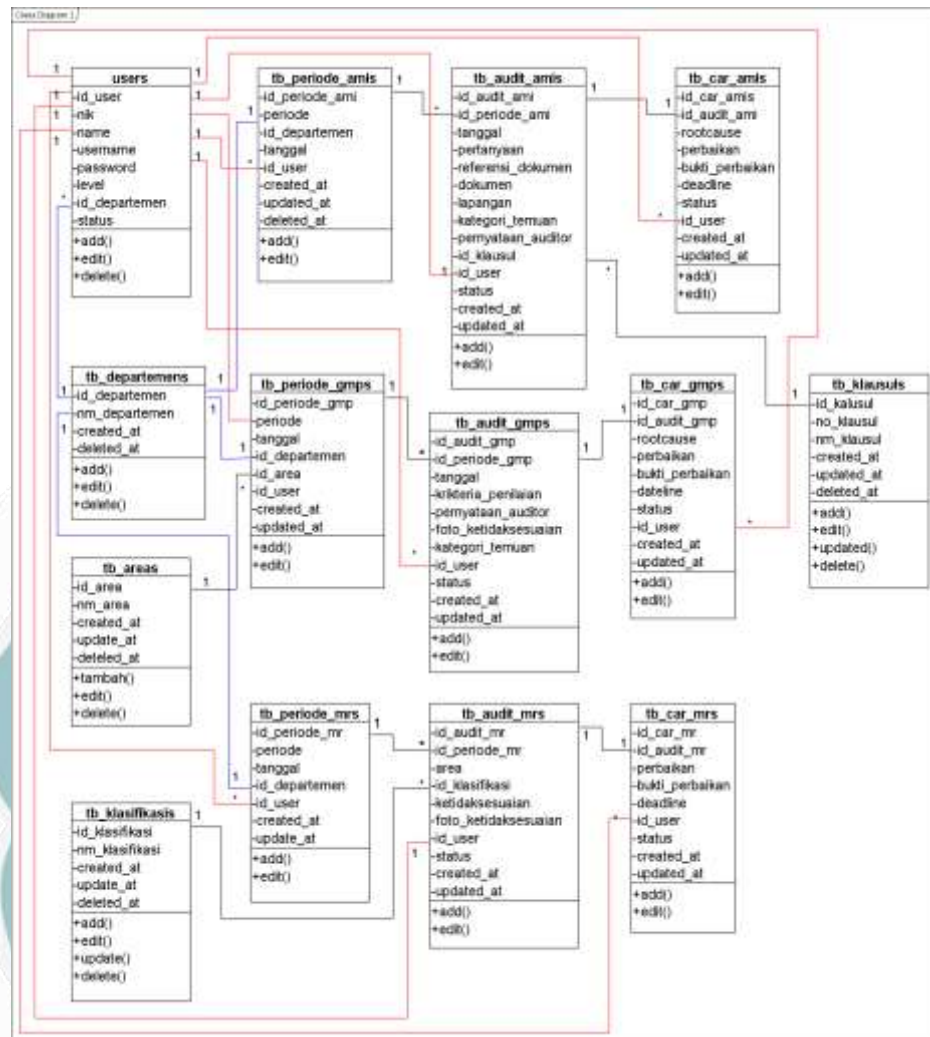


GAMBAR 4.48 *Sequence Diagram Laporan Audit Morning Round*
Berdasarkan Periode Audit

Keterangan :

- 1) Admin mengakses aplikasi menu utama.
- 2) Admin memilih laporan, kemudian pilih *Audit Morning Round*.
- 3) Admin memilih departemen dan periode *audit*, kemudian pilih cetak.
- 4) Sistem memproses penarikan data dan menampilkan laporan yang akan dicetak.

4.2.4. Class Diagram



GAMBAR 4.49 Class Diagram

4.2.5. Perancangan Basic Data

a) Tabel Users

Nama : User

Primary Key : id_User

TABEL 4.28 Spesifikasi Tabel Users

No	Nama Field	Type	Size	Keterangan
1	id_user	Int	5	Primary Key

2	nik	Varchar	5	
3	nama	Varchar	30	
4	username	Varchar	20	
5	password	Varchar	255	
6	level	Varchar	5	
7	id_departemen	Int	5	Foreign Key
8	status	Int	5	
9	remember_token	Varchar	100	
10	created_at	Timestamp		
11	update_at	Timestamp		
12	delete_at	Timestamp		

b) Tabel Departemen

Nama : *tb_departemens*

Primary Key : *id_departemen*

TABEL 4.29 Spesifikasi Tabel *tb_departemens*

No	Nama Field	Type	Size	Keterangan
1	id_departemen	Int	11	Primary Key
2	nm_departemen	Varchar	20	
3	created_at	Timestamp		
4	update_at	Timestamp		
	delete_at	Timestamp		

c) Tabel Area

Nama : *tb_areas*

Primary Key : *id_area*

TABEL 4.30 Spesifikasi Tabel *tb_areas*

No	Nama Field	Type	Size	Keterangan
1	id_area	Int	11	Primary Key
2	nm_area	Varchar	20	
3	created_at	Timestamp		
4	update_at	Timestamp		
5	delete_at	Timestamp		

d) Tabel Klausul

Nama : *tb_klausuls*

Primary Key : *id_klausul*

TABEL 4.31 Spesifikasi Tabel *tb_klausuls*

No	Nama Field	Type	Size	Keterangan
1	id_klausul	Int	10	Primary Key
2	nm_klausul	Varchar	20	
3	created_at	Timestamp		
4	update_at	Timestamp		
5	delete_at	Timestamp		

e) Tabel Klasifikasi

Nama : *tb_klasifikasis*

Primary Key : *id_klasifikasi*

TABEL 4.32 Spesifikasi Tabel *tb_klasifikasi*

No	Nama Field	Type	Size	Keterangan
1	id_klasifikasi	Int	11	Primary Key
2	nm_klasifikasi	Varchar	20	
3	created_at	Timestamp		

4	<i>update_at</i>	<i>Timestamp</i>		
5	<i>delete_at</i>	<i>Timestamp</i>		

f) Tabel Periode AMI

Nama : *tb_periode_amis*

Primary Key : *id_periode_ami*

TABEL 4.33 Spesifikasi Tabel *tb_periode_amis*

No	Nama Field	Type	Size	Keterangan
1	<i>id_periode_ami</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>periode</i>	<i>Date</i>		
3	<i>tanggal</i>	<i>Date</i>		
4	<i>id_departemen</i>	<i>Int</i>	11	<i>Foreign Key</i>
5	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
6	<i>created_at</i>	<i>Timestamp</i>		
7	<i>update_at</i>	<i>Timestamp</i>		
8	<i>delete_at</i>	<i>Timestamp</i>		

g) Tabel Periode GMP

Nama : *tb_periode_gmps*

Primary Key : *id_periode_gmp*

TABEL 4.34 Spesifikasi Tabel *tb_periode_gmps*

No	Nama Field	Type	Size	Keterangan
1	<i>id_periode_gmp</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>periode</i>	<i>Date</i>		
3	<i>tanggal</i>	<i>Date</i>		

4	<i>id_departemen</i>	<i>Int</i>	11	<i>Foreign Key</i>
5	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
6	<i>created_at</i>	<i>Timestamp</i>		
7	<i>update_at</i>	<i>Timestamp</i>		
8	<i>delete_at</i>	<i>Timestamp</i>		

h) Tabel Periode *Morning Round*

Nama : *tb_periode_mrs*

Primary Key : *id_periode_mr*

TABEL 4.35 Spesifikasi Tabel *tb_periode_mrs*

No	Nama <i>Field</i>	Type	Size	Keterangan
1	<i>id_periode_mr</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>periode</i>	<i>Date</i>		
3	<i>tanggal</i>	<i>Date</i>		
4	<i>id_departemen</i>	<i>Int</i>	11	<i>Foreign Key</i>
5	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
6	<i>created_at</i>	<i>Timestamp</i>		
7	<i>update_at</i>	<i>Timestamp</i>		
8	<i>delete_at</i>	<i>Timestamp</i>		

i) Tabel Audit AMI

Nama : *tb_audit_amis*

Primary Key : *id_audit_ami*

TABEL 4.36 Spesifikasi Tabel *tb_audit_amis*

No	Nama Field	Type	Size	Keterangan
1	<i>id_audit_ami</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>id_periode_mr</i>	<i>Int</i>	11	<i>Foreign Key</i>
3	<i>rootcause</i>	<i>Varchart</i>	225	
4	<i>perbaikan</i>	<i>Varchart</i>	225	
5	<i>bukti_perbaikan</i>	<i>Varchart</i>	225	
6	<i>deadline</i>	<i>Date</i>		
7	<i>Status</i>	<i>Int</i>	11	
8	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
9	<i>created_at</i>	<i>Timestamp</i>		
10	<i>update_at</i>	<i>Timestamp</i>		

j) Tabel Audit GMP

Nama : *tb_audit_gmps*

Primary Key : *id_audit_gmp*

TABEL 4.37 Spesifikasi Tabel *tb_audit_gmps*

No	Nama Field	Type	Size	Keterangan
1	<i>id_audit_gmp</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>id_periode_gmp</i>	<i>Int</i>	11	<i>Foreign Key</i>
3	<i>tanggal</i>	<i>Date</i>		
4	<i>kriteria_penilaian</i>	<i>Varchar</i>	255	
5	<i>pernyataan_auditor</i>	<i>Varchar</i>	225	
6	<i>foto_ketidaksesuaian</i>	<i>Varchar</i>	225	
7	<i>kategori_temuan</i>	<i>Varchar</i>	225	
8	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>

9	<i>Status</i>	<i>Int</i>	11	
10	<i>created_at</i>	<i>Timestamp</i>		
11	<i>update_at</i>	<i>Timestamp</i>		

k) Tabel *Audit Morning Round*

Nama : *tb_audit_mrs*

Primary Key : *id_audit_mr*

TABEL 4.38 Spesifikasi Tabel *tb_audit_mrs*

No	Nama Field	Type	Size	Keterangan
1	<i>id_audit_mr</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>id_periode_mr</i>	<i>Int</i>	11	<i>Foreign Key</i>
3	<i>area</i>	<i>Varchar</i>	100	
4	<i>id_klasifikasi</i>	<i>Int</i>	11	<i>Foreign Key</i>
5	<i>ketidaksesuaian</i>	<i>Varchar</i>	225	
6	<i>foto_ketidaksesuaian</i>	<i>Varchar</i>	225	
7	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
8	<i>Status</i>	<i>Int</i>	11	
9	<i>created_at</i>	<i>Timestamp</i>		
10	<i>update_at</i>	<i>Timestamp</i>		

l) Tabel CAR AMI

Nama : *tb_car_amis*

Primary Key : *id_car_ami*

TABEL 4.39 Spesifikasi Tabel *tb_car_amis*

No	Nama Field	Type	Size	Keterangan
1	<i>id_car_ami</i>	<i>Int</i>	11	<i>Primary Key</i>

2	<i>id_audit_ami</i>	<i>Int</i>	11	<i>Foreign Key</i>
3	<i>rootcause</i>	<i>Varchar</i>	225	
4	<i>perbaikan</i>	<i>Varchar</i>	225	
5	<i>bukti_perbaikan</i>	<i>Varchar</i>	225	
6	<i>deadline</i>	<i>Date</i>		
7	<i>Status</i>	<i>Int</i>	11	
8	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
9	<i>created_at</i>	<i>Timestamp</i>		
10	<i>update_at</i>	<i>Timestamp</i>		

m) Tabel CAR GMP

Nama : *tb_car_gmps*

Primary Key : *id_car_gmp*

TABEL 4.40 Spesifikasi Tabel *tb_car_gmps*

No	Nama Field	Type	Size	Keterangan
1	<i>id_car_gmp</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>id_audit_gmp</i>	<i>Int</i>	11	<i>Foreign Key</i>
3	<i>rootcause</i>	<i>Varchar</i>	225	
4	<i>perbaikan</i>	<i>Varchar</i>	225	
5	<i>bukti_perbaikan</i>	<i>Varchar</i>	225	
6	<i>deadline</i>	<i>Varchar</i>	225	
7	<i>Status</i>	<i>Int</i>	11	
8	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
9	<i>created_at</i>	<i>Timestamp</i>		
10	<i>update_at</i>	<i>Timestamp</i>		

n) Tabel CAR *Morning Round*

Nama : *tb_car_mrs*

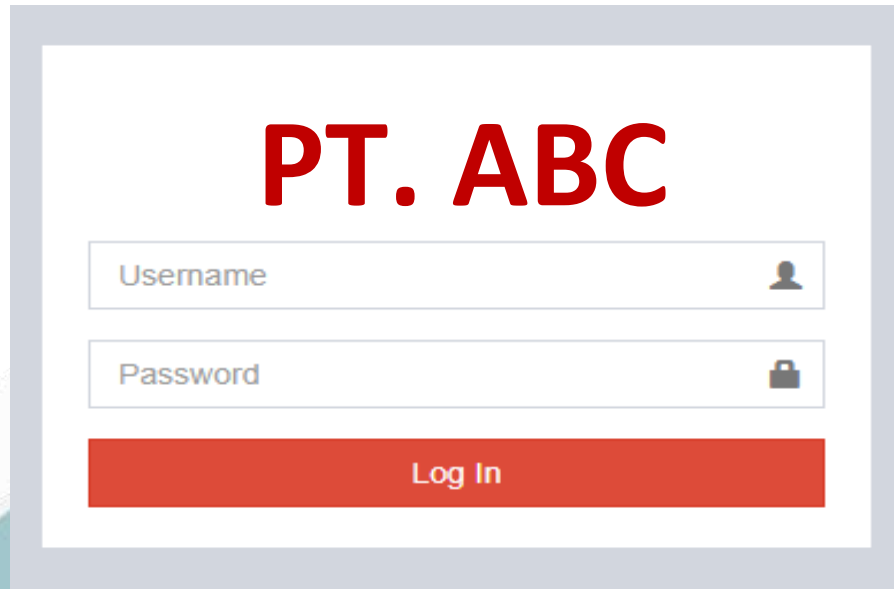
Primary Key : *id_car_mr*

TABEL 4.41 Spesifikasi Tabel *tb_car_mrs*

No	Nama <i>Field</i>	Type	Size	Keterangan
1	<i>id_car_mr</i>	<i>Int</i>	11	<i>Primary Key</i>
2	<i>id_audit_mr</i>	<i>Int</i>	11	<i>Foreign Key</i>
3	perbaikan	<i>Varchar</i>	225	
4	bukti_perbaikan	<i>Varchar</i>	225	
5	<i>deadline</i>	<i>Varchar</i>	225	
6	Status	<i>Int</i>	11	
7	<i>id_user</i>	<i>Int</i>	11	<i>Foreign Key</i>
8	<i>created_at</i>	<i>Timestamp</i>		
9	<i>update_at</i>	<i>Timestamp</i>		

4.2.6. Rancangan Tampilan Program yang diusulkan

a) Tampilan *Form Login User*



The image shows a user login form for PT. ABC. It features a large red title "PT. ABC" at the top. Below the title are two input fields: "Username" with a user icon and "Password" with a lock icon. A red "Log In" button is positioned below the password field. The form is set against a light gray background with a faint watermark of a circular logo.

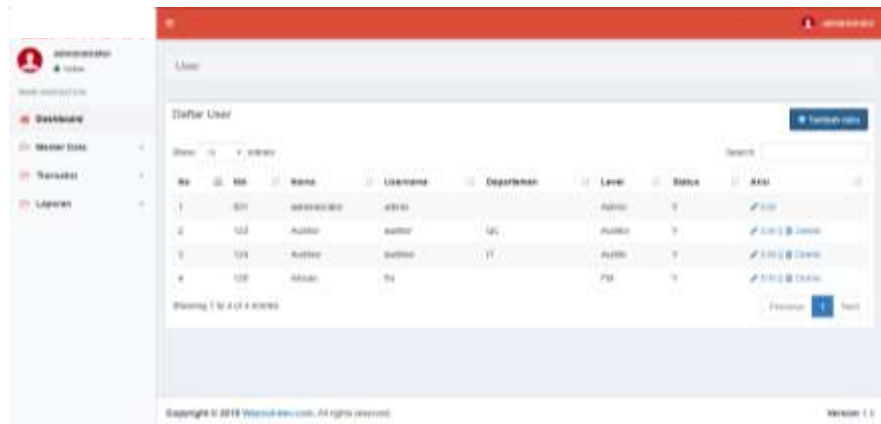
GAMBAR 4.50 Tampilan *Form Login*

b) Tampilan Menu Utama Admin



GAMBAR 4.51 Tampilan Menu Utama

c) Tampilan Menu Master *user*



GAMBAR 4.52 Tampilan Daftar *User*

Tambah User

NIK (No Induk Karyawan)

Nama

Username

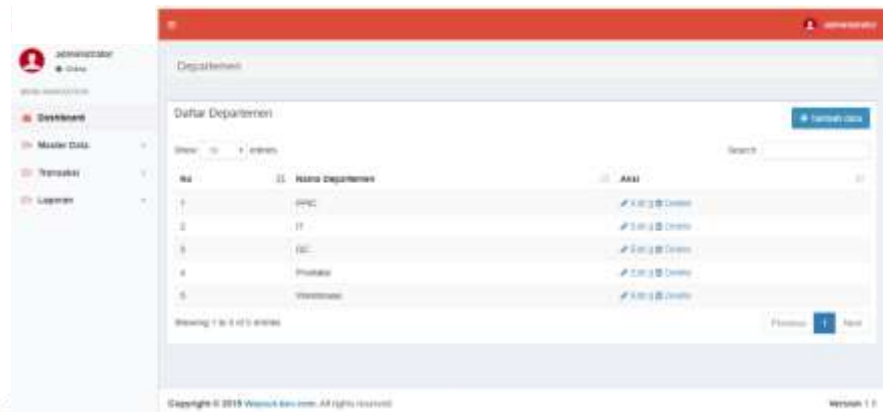
Level

Password

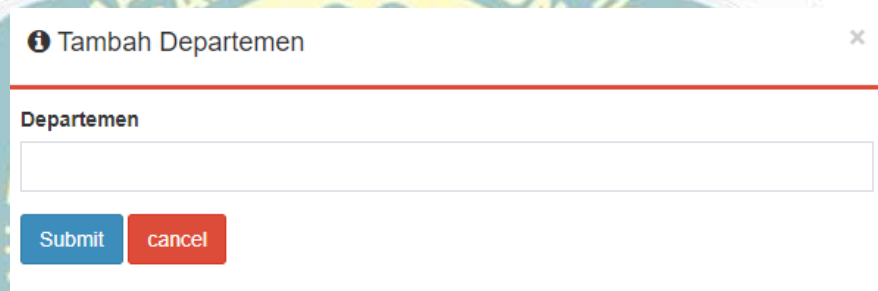
Submit cancel

GAMBAR 4.53 Tampilan Tambah Data *User*

d) Tampilan Menu Master Departemen



GAMBAR 4.54 Tampilan Daftar Departemen



GAMBAR 4.55 Tampilan Tambah Data Departemen

e) Tampilan Menu Master Area



GAMBAR 4.56 Tampilan Daftar Area

Tambah Area

Area

Submit

cancel

GAMBAR 4.57 Tampilan Tambah Data Area

f) Tampilan Menu Master Klausul

administrator

Daftar

Daftar

Daftar

Daftar

Daftar

Daftar

Daftar Klausul

Daftar Klausul

Daftar Klausul

Daftar Klausul

Daftar Klausul

Daftar Klausul

Daftar Klausul

GAMBAR 4.58 Tampilan Daftar Klausul

Tambah Klausul

No Klausul

Deskripsi Klausul

Submit

cancel

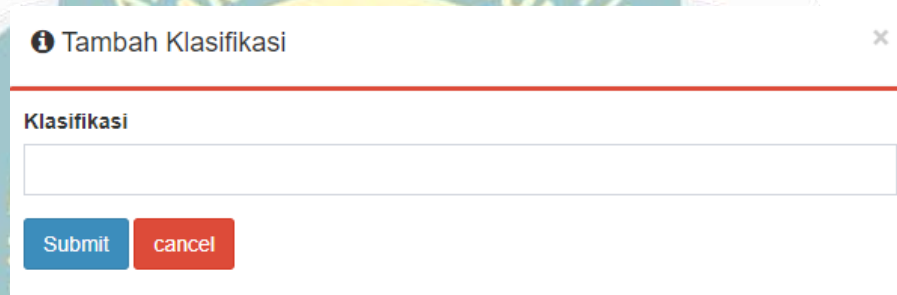
GAMBAR 4.59 Tampilan Tambah Data Klausul

177

g. Tampilan Menu Master Klasifikasi



GAMBAR 4.60 Tampilan Daftar Klasifikasi



GAMBAR 4.61 Tampilan Tambah Data Klasifikasi

h. Tampilan Menu Periode Audit



GAMBAR 4.62 Tampilan Daftar Periode Audit AMI

Tambah Periode AMI

×

Periode

mm/dd/yyyy

Departemen

--pilih--

Submit

cancel

GAMBAR 4.63 Tampilan Tambah Data Periode *Audit* AMI

administrator

• User

2019-10-20 10:10:10

Dashboard

Master Data

Transaksi

Laporan

Periode Audit GMP

Daftar Periode Audit GMP

Show 15 + items

Search

No	Periode	Departemen	Area	Aksi
1	September 2019	Engineering A	Packing Line 1	Edit Delete
2	November 2019	PPIC	Plant 4	Edit Delete
3	Desember 2019	Proteksi	Packing Line 6	Edit Delete
4	November 2019	Proteksi	SCADA	Edit Delete
5	November 2019	Proteksi	Packing Line 3	Edit Delete

Showing 1 to 5 of 5 entries

Previous

Next

Copyright © 2019 All rights reserved.

Version 1.0

GAMBAR 4.64 Tampilan Daftar Periode *Audit* GMP

Tambah Periode AMI

➤

Periode

mm/dd/yyyy

Departemen

--pilih--

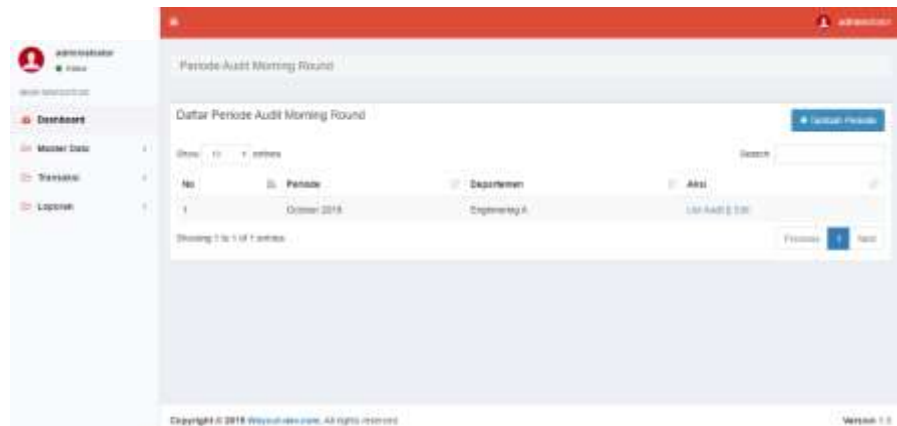
Area

--pilih--

Submit

cancel

GAMBAR 4.65 Tampilan Tambah Data Periode *Audit* GMP



GAMBAR 4.66 Tampilan Daftar Periode *Audit Morning Round*

Tambah Periode Audit Morning Round

Periode

mm/dd/yyyy

Departemen

--pilih--

Submit

cancel

GAMBAR 4.67 Tampilan Tambah Data Periode *Audit Morning Round*

i. Tampilan Menu Utama *Auditor*



GAMBAR 4.68 Tampilan Menu Utama *Auditor*

j. Tampilan Menu *Audit AMI*

The screenshot shows the 'Form Audit AMI' interface. It features a sidebar with 'Dashboard' and 'Transaksi' options. The main form area contains several input fields: 'Klasifikasi Akun' (dropdown), 'Referensi Dokumen' (text), 'Referensi Dokumen' (text), 'Kategori Laporan' (dropdown), 'Pembuat Audit' (text), and 'Author' (text). At the bottom, there are 'Submit' and 'Cancel' buttons.

GAMBAR 4.69 Tampilan *Form Audit AMI*

The screenshot shows the 'Daftar Audit AMI' interface. It features a sidebar with 'Dashboard' and 'Transaksi' options. The main area displays a table of audit records. The table has the following columns: No, Partisipan, Klasifikasi Akun, Referensi Dokumen, Dokumen, Laporan, Kategori Laporan, Pembuat Audit, Status, and Aksi. A 'Tambah Pendaftaran' button is located at the top right. The table contains one row of data. Below the table, there is a 'Kirim' button and a 'Filter' button.

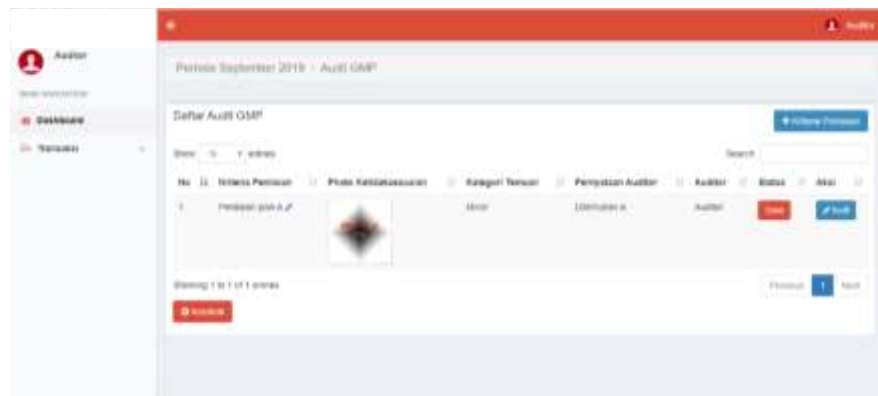
No	Partisipan	Klasifikasi Akun	Referensi Dokumen	Dokumen	Laporan	Kategori Laporan	Pembuat Audit	Status	Aksi
1	Partisipan A	Pengelolaan Dokumen	Dokumen A	Tidak	Tidak	Tidak	Dikirimkan A	Ya	Audit

GAMBAR 4.70 Tampilan Daftar *Audit AMI*

k. Tampilan Menu *Audit GMP*

The screenshot shows the 'Form Audit GMP' interface. It features a sidebar with 'Dashboard' and 'Transaksi' options. The main form area contains several input fields: 'Pembuat audit' (text), 'Foto Hasil Audit' (image upload), 'Kategori Laporan' (dropdown), and 'Author' (text). At the bottom, there are 'Submit' and 'Kirim' buttons.

GAMBAR 4.71 Tampilan *Form Audit GMP*



GAMBAR 4.72 Tampilan Daftar *Audit GMP*

1. Tampilan Menu *Audit Morning Round*

Form Audit Morning Round

Area

Klasifikasi

Temuan

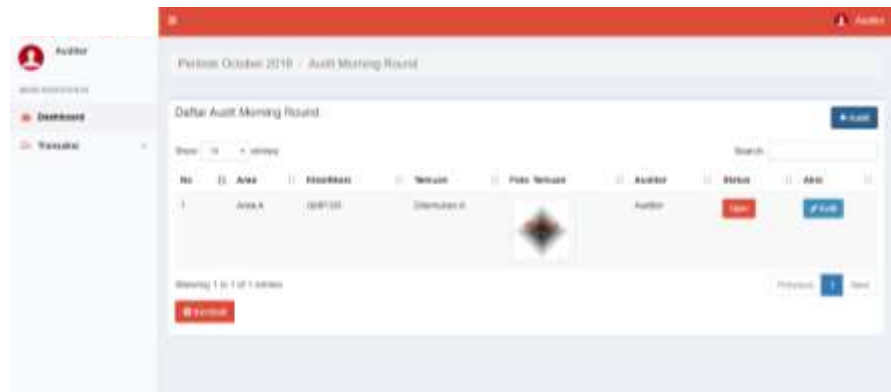
Foto Temuan

Unggah Foto

Submit

cancel

GAMBAR 4.73 Tampilan *Form Audit Morning Round*



GAMBAR 4.74 Tampilan Daftar *Audit Morning Round*

m. Tampilan Menu Utama *Auditee*



GAMBAR 4.75 Tampilan Menu Utama *Auditee*

n. Tampilan *Form CAR AMI*



GAMBAR 4.76 Tampilan *Form CAR AMI*

o. Tampilan *Form* CAR GMP

The screenshot shows the AWS IAM console 'Groups' page. The 'Groups' table has one entry: 'johndoe' with 1 member. The 'Users' table has one entry: 'johndoe' with a 'Groups' column containing 'johndoe'. The 'Groups' column in the 'Users' table is highlighted with a red box.

GAMBAR 4.77 Tampilan *Form* CAR GMP

p. Tampilan *Form CAR Morning Round*

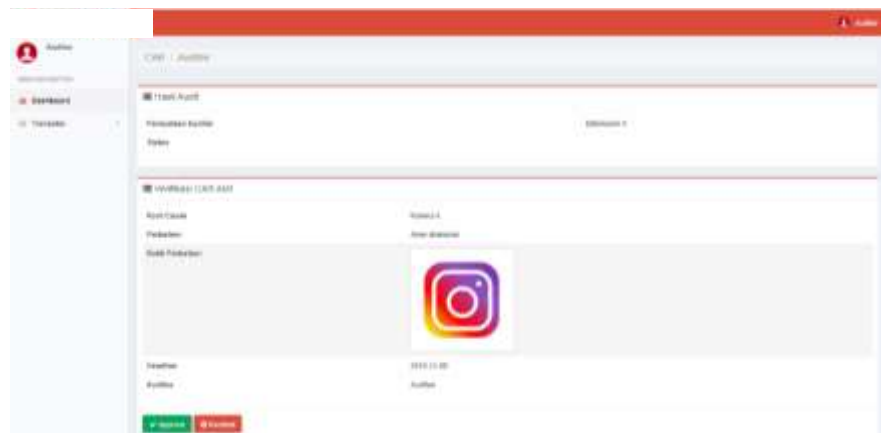
The screenshot shows the AWS IAM console interface. On the left, a navigation sidebar contains links for 'Overview', 'Users', 'Groups', 'Roles', and 'Policies'. The 'Groups' link is selected. The main panel displays a table of IAM groups. One group, 'Programmer Group', is listed and highlighted with a red border. Below the table, the 'Details' tab for 'Programmer Group' is open, showing fields for 'Name', 'Display Name', and 'Description', all of which are currently empty.

GAMBAR 4.78 Tampilan Form CAR *Morning Round*

q. Tampilan *Approve CAR Audit* AMI

GAMBAR 4.79 Tampilan *Approve CAR Audit* AMI

r. Tampilan *Approve* CAR Audit GMP



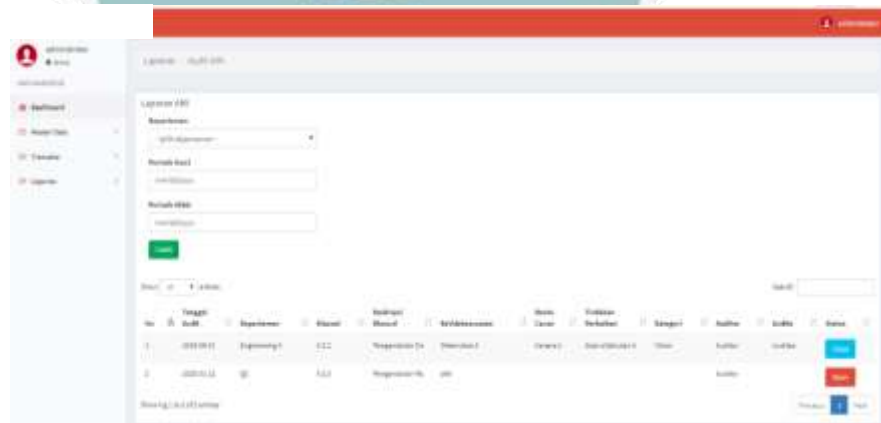
GAMBAR 4.80 Tampilan *Approve CAR Audit GMP*

s. Tampilan *Approve CAR Audit Morning Round*



GAMBAR 4.81 Tampilan *Approve CAR Audit Morning Round*

t. Tampilan Laporan *Audit* AMI



GAMBAR 4.82 Tampilan Laporan AMI

Print Date : 24-02-2020

LAPORAN AUDIT AMI
Departemen Engineering A Periode June 2019 s/d Periode February 2020

No	Tanggal	Departemen	Area	Revisi/Revisi	Kategori Temuan	Risiko Cause	Perbaikan	Deadline	Auditor	Audit	Status
1	2019-09-05	Engineering A	S.2.2	Revisi/Revisi	Operasional A	Kategori A	Revisi/Revisi A	Minor	Auditor	Audit	Close

Printed By

administrator

GAMBAR 4.83 Tampilan Laporan AMI yang Akan Dicitak

u. Tampilan Laporan *Audit* GMP

Laporan GMP

Revisi/Revisi

Kategori Temuan

Risiko Cause

Perbaikan

Print

No	Tanggal	Departemen	Area	Revisi/Revisi	Kategori Temuan	Risiko Cause	Perbaikan	Deadline	Auditor	Audit	Status
1	2019-09-05	Engineering A	Packing Line 1	Operasional A	Minor	Kategori A	Revisi/Revisi A	2019-12-09	Auditor	Audit	Close

Showing 2 of 12 items

GAMBAR 4.84 Tampilan Laporan *Audit* GMP

Print Date : 24-02-2020

LAPORAN AUDIT GMP
Departemen Engineering A Periode June 2019 s/d Periode February 2020

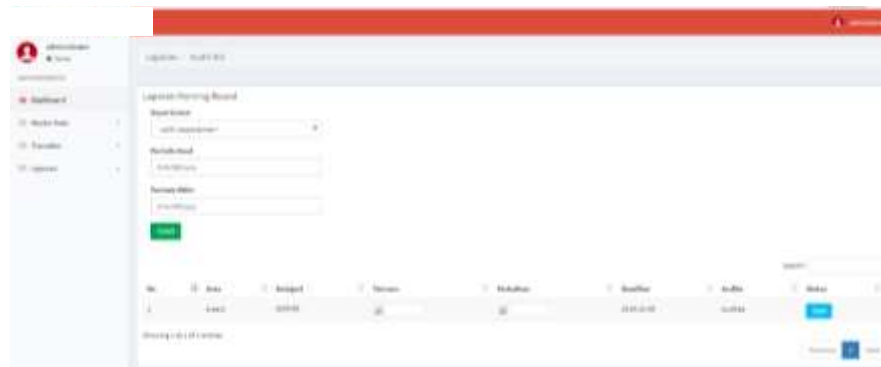
No	Tanggal	Departemen	Area	Revisi/Revisi	Kategori Temuan	Risiko Cause	Perbaikan	Deadline	Auditor	Audit	Status
1	2019-09-05	Engineering A	Packing Line 1	Operasional A	Minor	Kategori A	Revisi/Revisi A	2019-12-09	Auditor	Audit	Close

Printed By

administrator

GAMBAR 4.85 Tampilan Laporan *Audit* GMP yang Akan Dicitak

v. Tampilan Laporan *Audit Morning Round*



GAMBAR 4.86 Tampilan Laporan *Audit Morning Round*



GAMBAR 4.87 Tampilan Laporan *Audit Morning Round* yang Akan Dicetak

4.3. Spesifikasi *Hardware* dan *Software*

4.3.1. Kebutuhan *Hardware*

Perangkat keras (*Hardware*) minimal yang dibutuhkan dalam menjalankan sistem antara lain :

- a) *Processor* : Intel Core 2 Duo 2.0 Ghz
- b) *Harddisk* : 250 GB
- c) *RAM* : 2 GB

4.3.2. Kebutuhan *Software*

Perangkat lunak (*software*) minimal yang dibutuhkan dalam menjalankan sistem antara lain :

- a) Sistem Operasi : *Windows 7*
- b) Program Aplikasi : *XAMPP, Sublim Text 3*
- c) *DBMS* : *MYSQL*
- d) Laporan : *Microsoft Excel*

4.4. Jadwal Pengembangan Sistem

Pengembangan ini dibutuhkan jadwal yang berfungsi sebagai panduan atau pedoman terhadap kegiatan atau tahapan yang harus dilakukan dalam pengembangan sistem untuk mencapai keinginan yang diharapkan dalam perencanaan pengembangan sistem.

TABEL 4.42 Jadwal Pengembangan Sistem

NO	Kegiatan	Agustus 2019				September 2019				Oktober 2019				November 2019			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Perencanaan																
2	Analisa Kebutuhan																
3	Desain																
4	Pengembangan																
5	Implementasi																
6	Pemeliharaan																
7	Dokumentasi																

Keterangan :

a. Perencanaan

Kegiatan yang menyangkut dasar untuk mendapatkan sumber daya yang dibutuhkan untuk mendukung pengembangan sistem dan memperoleh solusi. Kegiatan ini direncanakan membutuhkan waktu dua minggu.

b. Analisa Kebutuhan

Tahap yang digunakan untuk menganalisis kebutuhan pemakaian sistem untuk menemukan kelemahan-kelemahan pada sistem yang ada sehingga dapat diusulkan perbaikannya. Analisa ini membutuhkan waktu selama tiga minggu

c. Desain

Tahap untuk membentuk sistem yang baru untuk memenuhi kebutuhan yang dibutuhkan pengguna. Desain membutuhkan waktu selama lima minggu.

d. Pengembangan

Tahap mengkonversi desain ke sistem informasi yang lengkap seperti bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan, membuat basis data, pengkodean, pengaplikasian dan pengujian. Waktu yang dibutuhkan selama lima minggu.

e. Implementasi

Tahap menerapkan sistem ke dalam kegiatan sebenarnya. Kegiatan yang dilakukan pada tahapan ini yaitu pemilihan dan pelatihan pengguna, pemilihan tempat dan instalasi *hardware* dan *software*, pengetesan program. Waktu yang dibutuhkan selama tiga minggu.

f. Pemeliharaan

Tahap yang dilakukan untuk menjaga dan atau memperbaikinya sampai suatu kondisi yang bisa diterima. Pemeliharaan diperlukan karena beberapa hal, yaitu :

1. Sistem memiliki kesalahan yang dulunya belum terdeteksi, sehingga kesalahan-kesalahan sistem perlu diperbaiki.
2. Sistem mengalami perubahan karena permintaan baru dari pemakai sistem.
3. Sistem mengalami perubahan karena perubahan lingkungan luar (perubahan bisnis).
4. Sistem terinfeksi program virus.

g. Dokumentasi

Tahap pencatatan dokumentasi sebagai *record* pelaksanaan selama jadwal pengembangan sistem sedang berjalan.



BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan uraian pada bab-bab sebelumnya maka dapat ditarik kesimpulan bahwa :

1. Sistem pengelolaan *monitoring* hasil *Internal Audit* pada PT. ABC saat ini yaitu :

- a) Laporan hasil *audit* masih dilakukan dengan menggunakan *form* yang diolah menggunakan *Ms. Office Excel*, kemudian hasil *audit* dikirim kepada semua departemen yang bersangkutan melalui *email* sehingga dalam pencarian informasi dan *monitoring* status mengenai *internal audit* kurang efektif dan efisien karena PIC yang bersangkutan kurang komunikasi serta *email* yang dikirim terdapat kemungkinan tidak tersampaikan atau terbaca.
- b) Pada saat admin melakukan *input* data hasil *audit*, tidak sedikit tulisan dari *auditor* tidak terbaca yang menyulitkan atau menghambat admin dalam proses *input* hasil *audit*.
- c) Dalam *memonitoring* status *internal audit* secara keseluruhan, departemen yang berkaitan (*auditee*) harus menghubungi PIC yang bersangkutan melalui pesawat telepon karena penyimpanan masing-masing hasil *audit* dikelola oleh PIC/admin yang berbeda.

d) PIC/admin yang mengelola hasil *audit* tidak selalu berada ditempat sehingga kurang efektif dan efisien dalam pencarian informasi hasil *audit* secara keseluruhan.

2. Sistem Monitoring Internal *Audit* ini lebih efisien dalam pelaksanaan monitoring, berikut beberapa kemudahannya :

- a) Internal *Audit* tidak lagi menggunakan kertas yang beresiko cepat rusak atau hilang. Internal *Audit* dibuat dengan menggunakan web aplikasi secara *online*.
- b) Admin tidak lagi melakukan *input* hasil *audit* yang beresiko salah *input* karena hasil *audit* sudah masuk kedalam sistem yang dibuat.
- c) Departemen yang terkait dalam *memonitoring* hasil *audit* tidak perlu lagi menghubungi admin atau PIC melalui pesawat telepon karena dapat langsung melakukan *monitoring* pada sistem yang dibuat sehingga lebih efektif dan efisien

Sistem *Monitoring Internal Audit* ini dirancang dengan menggunakan metode pengembangan *System Development Life Cycle (SDLC)* dengan model *Waterfall*, *Unified Modeling Language (UML)* dengan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*, serta menggunakan *database MySQL* dan dengan beberapa *software* antara lain *XAMPP*, *PHP*, dan *Sublime Text*.

5.2. Saran

Berdasarkan kesimpulan yang telah dikemukakan, dapat diajukan beberapa saran untuk pengembangan lebih lanjut antara lain :

- a. Bagi peneliti selanjutnya disarankan dapat mengembangkan aplikasi ini dengan menambahkan konten-konten yang masih kurang dan dapat menggunakan metode yang terkini seperti android.
- b. Bagi PT. ABC, sebelum aplikasi ini diterapkan disarankan untuk melakukan sosialisasi kepada semua karyawan yang terkait bagaimana cara menggunakan aplikasi.
- c. Perlu adanya pengawasan serta pengontrolan dalam pelaksanaan sistem ini agar sistem ini dapat berjalan dengan baik.
- d. Perlu dilakukan pemeliharaan sistem secara berkala agar sistem ini dapat berjalan dengan baik.



DAFTAR PUSTAKA

- Arif, Sugeng. (2018). Analisa dan Perancangan Sistem Informasi *Monitoring Pelaksanaan Improvement* di PT. 123 Divisi *Candy* Departemen Teknik *Project and Improvement*. Skripsi Sarjana. Tangerang : STMIK Insan Pembangunan
- Arysandy, Yosy dkk. (2017). Sistem Informasi Manajemen. Yogyakarta : Pustaka Pelajar
- Gunawan, Imam. (2017). Metode Penelitian Kualitatif. Jakarta Timur: Bumi Aksara.
- Hutahaean, Jeperson. (2014). Konsep Sistem Informasi. Yogyakarta: Deepublish
- Istriawan, Dina. (2018). Analisis dan Perancangan Sistem Informasi *Monitoring Laporan Harian Pada Area Produksi Wrapping* di PT. Torabika Eka Semesta Ground 2. Skripsi Sarjana. Tangerang : STMIK Insan Pembangunan.
- Kadir, Abdul. (2014). Pengenalan Sistem Informasi. Yogyakarta: CV. Andi Offset.
- Kristanto, Andri. (2018). Perancangan Sistem Informasi dan Aplikasinya. Yogyakarta: Gava Media.
- Mulyani, Sri. (2016). Sistem Informasi Manajemen. Bandung : Abdi Sistematika
- Prijambodo, (2014). Monitoring dan Evaluasi. Bogor: IPB Press.
- Shalahuddin, M. and A, S Rosa. (2018). Rekayasa Perangkat Lunak. Bandung: Informatika.
- Sunyoto, Danang. (2014). Sistem Informasi Manajemen. Yogyakarta: CAPS (Center of Academic Publishing Service).
- Wicaksono, Soetam Rizky. (2017). Rekayasa Perangkat Lunak. Seribu Bintang.
- Winarti, Endang Widi. (2018). Penelitian Kuantitatif Kualitatif. Jakarta Timur: Bumi Aksara.
- Yosi, Andi dan Adhadi. (2017). Sistem Informasi Manajemen. Yogyakarta : Pustaka Pelajar.
- Yulianto, Andi (2015). Analisa dan Perancangan Sistem Informasi *Maintenance* untuk *Monitoring* Mesin Produksi Pada PT. Cingluh Indonesia 2. Skripsi Sarjana. Tangerang : STMIK INsan Pembangunan

- Tandiontong, Mathius. (2016). Kualitas Audit dan Pengukurannya. Bandung: Alfabeta.
- Tohari, Hamim. (2014). Analisa Serta Perancangan Sistem Informasi Melalui Pendekatan UML. Yogyakarta: Andi Publisher.
- Tyoso, Juluanto Sunu Punjul. (2016). Sistem Informasi Manajemen. Yogyakarta: Deepublish.

