



**APLIKASI PENILAIAN TINGKAT KEMATANGAN
TATA KELOLA TI BERDASARKAN DOMAIN *PLAN AND ORGANISE* (PO)
DAN *ACQUIRE AND IMPLEMENT* (AI) FRAMEWORK COBIT 4.1
(STUDI KASUS SISTEM INFORMASI TERPADU UNIVERSITAS JEMBER)**

SKRIPSI

Oleh

Dhevi Indriawati

NIM 122410101065

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016



**APLIKASI PENILAIAN TINGKAT KEMATANGAN
TATA KELOLA TI BERDASARKAN DOMAIN *PLAN AND ORGANISE* (PO)
DAN *ACQUIRE AND IMPLEMENT* (AI) *FRAMEWORK* COBIT 4.1
(STUDI KASUS SISTEM INFORMASI TERPADU UNIVERSITAS JEMBER)**

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

Oleh

Dhevi Indriawati

NIM 122410101065

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Damsiri dan Ibunda Suharlin tercinta;
2. Kakak perempuan satu-satunya Siti Yunia Adchawati, kakak ipar Edi Susanto, dan keponakan tersayang Kenzie Kaesang Adinata Sayyidina;
3. Kakek H. Habib dan kakek Kartono beserta keluarga besar;
4. Sahabatku bersama dukungan dan doanya;
5. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
6. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“Niscaya Allah akan mengangkat (derajat) orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu beberapa derajat”. (QS. Al-Mujadalah : 11)¹



¹ Kementerian Agama Republik Indonesia. 2014. *Ummul Mukminin : Al Qur'an dan Terjemahannya untuk Wanita*. Jakarta: OASIS Terrace Resident

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Dhevi Indriawati

NIM : 122410101066

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Aplikasi Penilaian Tingkat Kematangan Tata Kelola TI Berdasarkan Domain *Plan and Organise* (PO) dan *Acquire and Implement* (AI) Framework COBIT 4.1 (Studi Kasus Sistem Informasi Terpadu Universitas Jember)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 13 Mei 2016

Yang menyatakan,

Dhevi Indriawati

NIM 122410101065

SKRIPSI

**APLIKASI PENILAIAN TINGKAT KEMATANGAN
TATA KELOLA TI BERDASARKAN DOMAIN *PLAN AND ORGANISE* (PO)
DAN *ACQUIRE AND IMPLEMENT* (AI) *FRAMEWORK* COBIT 4.1
(STUDI KASUS SISTEM INFORMASI TERPADU UNIVERSITAS JEMBER)**

Oleh
Dhevi Indriawati
NIM 122410101065

Pembimbing :

Dosen Pembimbing Utama : Dr. Saiful Bukhori, ST., M.Kom

Dosen Pembimbing Pemdamping : Windi Eka Yulia Retnani, S.Kom., M.T

PENGESAHAN PEMBIMBING

Skripsi berjudul “Aplikasi Penilaian Tingkat Kematangan Tata Kelola TI Berdasarkan Domain *Plan and Organise* (PO) dan *Acquire and Implement* (AI) *Framework* COBIT 4.1 (Studi Kasus Sistem Informasi Terpadu Universitas Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 13 Mei 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Dr. Saiful Bukhori, ST., M.Kom
NIP 196811131994121001

Windi Eka Yulia Retnani, S.Kom, MT
NIP 198403052010122002

PENGESAHAN

Skripsi berjudul “Aplikasi Penilaian Tingkat Kematangan Tata Kelola TI Berdasarkan Domain *Plan and Organise* (PO) dan *Acquire and Implement* (AI) Framework COBIT 4.1 (Studi Kasus Sistem Informasi Terpadu Universitas Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 13 Mei 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST., MT
NIP 19690615 1997021002

Fahrobby Adnan, S.Kom., M.MSI
NIP 19870619 2014041001

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamin, M.Comp.Sc.,Ph.D
NIP. 19670420 1992011001

RINGKASAN

Aplikasi Penilaian Tingkat Kematangan Tata Kelola TI Berdasarkan Domain *Plan and Organise (PO)* dan *Acquire and Implement (AI)* Framework COBIT 4.1 (Studi Kasus Sistem Informasi Terpadu Universitas Jember)

Penerapan teknologi di Universitas Jember tidak selalu selaras dengan tujuan dari Universitas Jember. TI menjadi bagian yang penting dalam menjalankan kegiatannya, maka harus dipastikan bahwa TI yang berjalan pada Universitas Jember berjalan dengan baik dan lancar. Untuk mengetahui seberapa sejauh pengelolaan TI yang telah dijalankan dalam sebuah perusahaan, sebagai bahan evaluasi untuk meningkatkan tata kelola TI kedepannya dan mengetahui hal-hal apa saja yang perlu diperbaiki. Maka perlu dilakukan analisis penilaian tingkat kematangan terhadap infrastruktur dan pengelolaan TI menggunakan *framework* COBIT 4.1. COBIT adalah *framework* yang digunakan untuk manajemen proses TI yang berfokus pada kontrol objektif yang dapat mengukur pengelolaan TI suatu perusahaan menggunakan *maturity model*. Domain dari *framework* COBIT yang digunakan dalam penelitian ini adalah domain PO dan AI.

Penelitian ini dilaksanakan dalam 3 tahap yaitu tahap pengumpulan data, tahap analisis, dan tahap pengembangan sistem. Tahap pengumpulan data dilakukan melalui studi literatur, obserasi, dan menyebarkan kuesioner kepada responden. Tahap analisis dilakukan dengan mengolah data hasil kuesioner sampai menemukan nilai dari tingkat kematangan tata kelola TI. Tahap pengembangan dilakukan dengan membangun sebuah aplikasi penilaian tingkat kematangan tata kelola TI yang berbasis web. Hasil dari penelitian ini adalah nilai tingkat kematangan tata kelola TI pada SISTER Universitas Jember berapa pada level 3.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Aplikasi Penilaian Tingkat Kematangan Tata Kelola TI Berdasarkan Domain *Plan and Organise* (PO) dan *Acquire and Implement* (AI) Framework COBIT 4.1 (Studi Kasus Sistem Informasi Terpadu Universitas Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamir, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Dr. Saiful Bukhori, ST.,M.Kom., selaku Dosen Pembimbing Utama dan Windi Eka Yulia Retnani S.Kom., MT., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Nelly Oktavia Adiwijaya, S.Si., MT. sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
4. seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ayahanda Damsiri dan Ibunda Suharlin yang selalu mendukung dan mendoakan;
6. Kakak perempuan Siti Yunia Adchawati beserta keluarga;
7. Kakek H. Habib dan kakek Kartono beserta keluarga besar;
8. sahabat seperjuangan yang selalu menemani dan memberikan semangat serta doa Rifta, Inez, Alfi, Dinda, Tiara, Ratih, Indira, Yudha, Yudi, Nindi, Vadil, Affan, Marcelli, dan Afendi;
9. Lukman Hakim beserta keluarga yang selalu memberikan motivasi;

10. teman-teman seperjuanganku FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
11. keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) Periode 2013-2014 dan periode 2014-2015 Yudha Herlambang C. P., Safitri Febryanti A., Nindi Norya Tindra, Yudi Candra K., Rizki Vadilla, Dimas Arifianto, Diah Ayu S., dan anggota lain yang tidak dapat disebutkan satu persatu;
12. keluarga besar asisten laboratorium Rekayasa Perangkat Lunak (RPL) tahun 2014-2015 dan 2015-2016 Mbak Qilba, Mas Nano, Mas Selamat, Mas Anwar, Mas Pandu, Haris, Rifta, Nindi, Yudi, Yudha, Affan, Herda, Helma, Oni, Widi, Elviera, Ridlo, Ghiffari, dan Alam;
13. keluarga besar Koperasi Mahasiswa Sistem Informasi (KOPMASI) periode 2012-2013 dan 2013-2014;
14. teman sekamar Rohmatul Isnaini beserta seluruh penghuni rumah singgah Jalan Jawa 4 no.2c yang telah menjadi rumah singgah dan keluarga penulis selama di Jember;
15. UPT TI Universitas Jember;
16. semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 13 Mei 2016

Penulis

DAFTAR ISI

SKRIPSI.....	i
SKRIPSI.....	i
PERSEMBAHAN.....	ii
MOTO.....	iii
PERNYATAAN.....	iv
SKRIPSI.....	v
PENGESAHAN PEMBIMBING.....	vi
PENGESAHAN.....	vii
RINGKASAN.....	viii
PRAKATA.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xix
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
BAB 2. TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terdahulu.....	5
2.2 Tata Kelola TI.....	6
2.3 <i>Framework</i> COBIT.....	7
2.3.1 Prinsip Dasar COBIT.....	7
2.3.2 Kontrol Objektif.....	11
2.3.4 Maturity Level.....	14

2.4	Model <i>Waterfall</i>	19
BAB 3. METODOLOGI PENELITIAN.....		20
3.1	Tahapan Penelitian	20
3.1.1	Tahapan Analisis Kebutuhan	20
3.1.2	Tahapan Desain Sistem	26
3.1.3	Tahapan Implementasi Sistem	28
3.1.4	Tahapan Pengujian Sistem	28
3.1.5	Tahapan Pemeliharaan Sistem	29
BAB 4. PENGEMBANGAN SISTEM.....		30
4.1	Analisis Kebutuhan Sistem	30
4.1.1	Kebutuhan Fungsional	30
4.1.2	Kebutuhan Non-Fungsional	31
4.2	Desain Sistem	31
4.1.1	<i>Business Process</i>	31
4.1.2	<i>Use Case Diagram</i>	32
4.1.3	<i>Use Case Scenario</i>	36
4.1.4	<i>Sequence Diagram</i>	44
4.1.5	<i>Activity Diagram</i>	49
4.1.6	<i>Class Diagram</i>	53
4.1.7	<i>Entity Relationship Diagram (ERD)</i>	54
4.3	Penulisan Kode Program	54
4.3.1	Kode Program <i>Login</i>	54
4.3.2	Kode Program Pengelolaan Data <i>User</i>	55
4.3.3	Kode Program Pengelolaan Data Domain	55
4.3.4	Kode Program Pengelolaan Data Kontrol Objektif	55
4.3.5	Kode Program Pengelolaan Data <i>Maturity Level</i>	55
4.3.6	Kode Program Pengelolaan Data Pernyataan	55
4.3.7	Kode Program Pengelolaan Data Periode	56
4.3.8	Kode Program <i>View COBIT</i>	56

4.3.9	Kode Program Mengisi Kuesioner.....	56
4.3.9	Kode Program <i>View</i> Hasil Perhitungan.....	67
4.4	Pengujian Sistem	67
4.4.1	Pengujian <i>White Box</i>	67
4.4.2	Pengujian <i>Black Box</i>	87
BAB 5. HASIL DAN PEMBAHASAN.....		91
5.1	Hasil Penelitian.....	91
5.2	Hasil Perhitungan <i>Maturity Level</i>	91
5.2.1	Konversi Data Kuesioner	92
5.2.2	Hasil Konversi Data Kuesioner.....	155
5.2.3	Nilai <i>Maturity Level</i> Tata Kelola TI.....	168
5.3	Hasil Pembangunan Aplikasi	171
5.3.1	Halaman <i>Login</i>	171
5.3.2	Halaman Dashboard	172
5.3.3	Halaman <i>User</i>	174
5.3.4	Halaman Domain	176
5.3.5	Halaman Kontrol Objektif.....	178
5.3.7	Halaman Kuesioner	180
5.3.6	Halaman <i>Maturity</i>	183
5.3.8	Halaman Periode	184
5.3.9	Halaman COBIT	186
5.3.10	Halaman Hasil Audit.....	187
BAB 6. PENUTUP		189
6.1	Kesimpulan.....	189
6.2	Saran	190
DAFTAR PUSTAKA		191
LAMPIRAN		192

DAFTAR GAMBAR

Gambar 2.1 Prinsip Dasar COBIT (ITGI, 2007).....	8
Gambar 2.2 Hubungan dari Komponen-komponen COBIT (ITGI, 2007)	9
Gambar 2.3 Hubungan 4 Domain COBIT (ITGI, 2007).....	11
Gambar 2.4 Keseluruhan Framework COBIT 4.1 (ITGI, 2007).....	11
Gambar 2.5 Representasi Grafik Maturity Model (ITGI, 2007).....	14
Gambar 2.6 Nilai Konversi Pernyataan (Pederiva, 2003).....	15
Gambar 2.7 <i>Flowchart</i> Perhitungan Nilai Kematangan.....	18
Gambar 2.8 Model Waterfall (Sommerville, 2003).....	19
Gambar 4.1 Business Process	32
Gambar 4.2 Use Case Diagram.....	33
Gambar 4.3 Sequence Diagram Pengelolaan Data Periode	46
Gambar 4.4 Sequence Diagram Mengisi Kuesioner Staff TI	47
Gambar 4.5 Sequence Diagram Mengisi Kuesioner Manajerial.....	47
Gambar 4.6 Sequence Diagram View Hasil Perhitungan	48
Gambar 4.7 Activity Diagram Mengisi Kuesioner Staff TI.....	50
Gambar 4.8 Activity Diagram Mengisi Kuesioner Manajerial	51
Gambar 4.9 Activity Diagram View Hasil Perhitungan	52
Gambar 4.10 Class Diagram	53
Gambar 4.11 Entity Relationship Diagram.....	54
Gambar 4.12 Kode Program function jawaban_ya	56
Gambar 4.13 Kode Program function simpan_jawaban_tidak	56
Gambar 4.14 Kode Program function get_jawaban	57
Gambar 4.15 Kode Program function simpan_konversi.....	57
Gambar 4.16 Kode Program Function set_jml_pernyataan	58
Gambar 4.17 Kode Program Function simpan_jml_pernyataan.....	58
Gambar 4.18 Kode Program Function set_avg_konversi	59
Gambar 4.19 Kode Program Function simpan_avg_konversi	59

Gambar 4.20 Kode Program Function set_tot_avg_konversi.....	59
Gambar 4.21 Kode Program Function simpan_tot_avg_konversi.....	60
Gambar 4.22 Kode Program Function set_jml_avg_konversi.....	60
Gambar 4.23 Kode Program Function simpan_jml_avg_konversi.....	61
Gambar 4.24 Kode Program Function set_normalisasi	61
Gambar 4.25 Kode Program Function simpan_normalisasi	61
Gambar 4.26 Kode Program Function set_kontribusi	61
Gambar 4.27 Kode Program Function simpan_kontribusi	62
Gambar 4.28 Kode Program Function hasil.....	62
Gambar 4.29 Kode Program Function simpan_hasil	63
Gambar 4.30 Kode Program Function get_periode	63
Gambar 4.31 Kode Program Function simpan_periode.....	63
Gambar 4.32 Kode Program Function simpan_jawaban_ya.....	63
Gambar 4.33 Kode Program simpan_jawaban_tidak.....	64
Gambar 4.34 Kode Program Function step1	64
Gambar 4.35 Kode Program Function step2.....	64
Gambar 4.36 Kode Program Function step3.....	65
Gambar 4.37 Kode Program Function step4.....	65
Gambar 4.38 Kode Program Function get_nilaico17	66
Gambar 4.39 Kode Program Function simpan_hasil	66
Gambar 4.40 Kode Program Function step5.....	66
Gambar 4.41 Diagram Alir Function simpan_jawaban_ya.....	67
Gambar 4.42 Diagram Alir Function simpan_jawaban_tidak	68
Gambar 4.43 Diagram Alir Function get_jawaban	68
Gambar 4.44 Diagram Alir Function simpan_konversi.....	70
Gambar 4.45 Diagram Alir Function set_jml_pernyataan.....	70
Gambar 4.46 Diagram Alir Function simpan_jml_pernyataan.....	72
Gambar 4.47 Diagram Alir Function set_avg_konversi	72
Gambar 4.48 Diagram Alir Function simpan_avg_konversi	73

Gambar 4.49 Diagram Alir Function set_tot_avg_konversi.....	74
Gambar 4.50 Diagram Alir Function simpan_tot_avg_konversi.....	75
Gambar 4.51 Diagram Alir Function set_jml_avg_konversi.....	76
Gambar 4.52 Diagram Alir Function simpan_jml_avg_konversi.....	77
Gambar 4.53 Diagram Alir Function set_normalisasi	78
Gambar 4.54 Diagram Alir Function simpan_normalisasi	78
Gambar 4.55 Diagram Alir Function set_kontribusi	79
Gambar 4.56 Diagram Alir Function simpan_kontribusi	79
Gambar 4.57 Diagram Alir Function hasil.....	80
Gambar 4.58 Diagram Alir Function simpan_hasil	81
Gambar 4.59 Diagram Alir Function get_periode	81
Gambar 4.60 Diagram Alir Function simpan_periode.....	82
Gambar 4.61 Diagram Alir Function simpan_jawaban_ya.....	82
Gambar 4.62 Diagram Alir Function simpan_jawaban_tidak	83
Gambar 4.63 Diagram Alir Function step1	83
Gambar 4.64 Diagram Alir Function step2.....	84
Gambar 4.65 Diagram Alir Function step3.....	84
Gambar 4.66 Diagram Alir Function step4.....	85
Gambar 4.67 Diagram Alir Function get_nilaico	85
Gambar 4.68 Diagram Alir Function simpan_hasil	86
Gambar 4.69 Diagram Alir Function step5.....	86
Gambar 5.1 Tampilan Hasil Maturity Level PO1	155
Gambar 5.2 Tampilan Hasil Maturity Level PO2.....	156
Gambar 5.3 Tampilan Hasil Maturity Level PO3.....	157
Gambar 5.4 Tampilan Hasil Maturity Level PO4.....	158
Gambar 5.5 Tampilan Hasil Maturity Level PO5.....	158
Gambar 5.6 Tampilan Hasil Maturity Level PO6.....	159
Gambar 5.7 Tampilan Hasil Maturity Level PO7.....	160
Gambar 5.8 Tampilan Hasil Maturity Level PO8.....	160

Gambar 5.9 Tampilan Hasil Maturity Level PO9.....	161
Gambar 5.10 Tampilan Hasil Maturity Level PO10.....	162
Gambar 5.11 Tampilan Hasil Maturity Level AI1.....	162
Gambar 5.12 Tampilan Hasil Maturity Level AI2.....	163
Gambar 5.13 Tampilan Hasil Maturity Level AI3.....	164
Gambar 5.14 Tampilan Hasil Maturity Level AI4.....	164
Gambar 5. 15 Tampilan Hasil Maturity Level AI5.....	165
Gambar 5.16 Tampilan Hasil Maturity Level AI6.....	166
Gambar 5.17 Tampilan Hasil Maturity Level AI7.....	167
Gambar 5.18 Tampilan Hasil Maturity Level.....	169
Gambar 5.19 Halaman Utama Aplikasi.....	171
Gambar 5.20 Halaman Login.....	172
Gambar 5.21 Halaman Dashboard Admin.....	172
Gambar 5.22 Halaman Dashboard Staff TI.....	173
Gambar 5.23 Halaman Dashboard Manajerial.....	173
Gambar 5.24 Halaman Dashboard PR 1.....	174
Gambar 5.25 Halaman User.....	175
Gambar 5.26 Halaman Tambah User.....	175
Gambar 5.27 Halaman Edit User.....	176
Gambar 5.28 Halaman Domain.....	177
Gambar 5.29 Halaman Tambah Domain.....	177
Gambar 5.30 Halaman Edit Domain.....	178
Gambar 5.31 Halaman Kontrol Objektif.....	179
Gambar 5.32 Halaman Tambah Kontrol Objektif.....	179
Gambar 5.33 Halaman Edit Kontrol Objektif.....	180
Gambar 5.34 Halaman Kuesioner Admin.....	181
Gambar 5.35 Halaman Tambah Kuesioner.....	181
Gambar 5.36 Halaman Edit Kuesioner.....	182
Gambar 5.37 Halaman Kuesioner Responden.....	182

Gambar 5.38 Halaman untuk Pengisian Kuesioner	183
Gambar 5.39 Halaman Maturity	183
Gambar 5.40 Halaman Tambah Maturity	184
Gambar 5.41 Halaman Edit Maturity.....	184
Gambar 5.42 Halaman Periode	185
Gambar 5.43 Halaman Tambah Periode	185
Gambar 5.44 Halaman Edit Periode.....	185
Gambar 5.45 Halaman COBIT	186
Gambar 5.46 Halaman Detail Kontrol Objektif.....	186
Gambar 5.47 Halaman Hasil Audit.....	187
Gambar 5.48 Halaman Hasil Maturity Level Kontrol Objektif.....	187
Gambar 5.49 Halaman Hasil Maturity Level.....	188

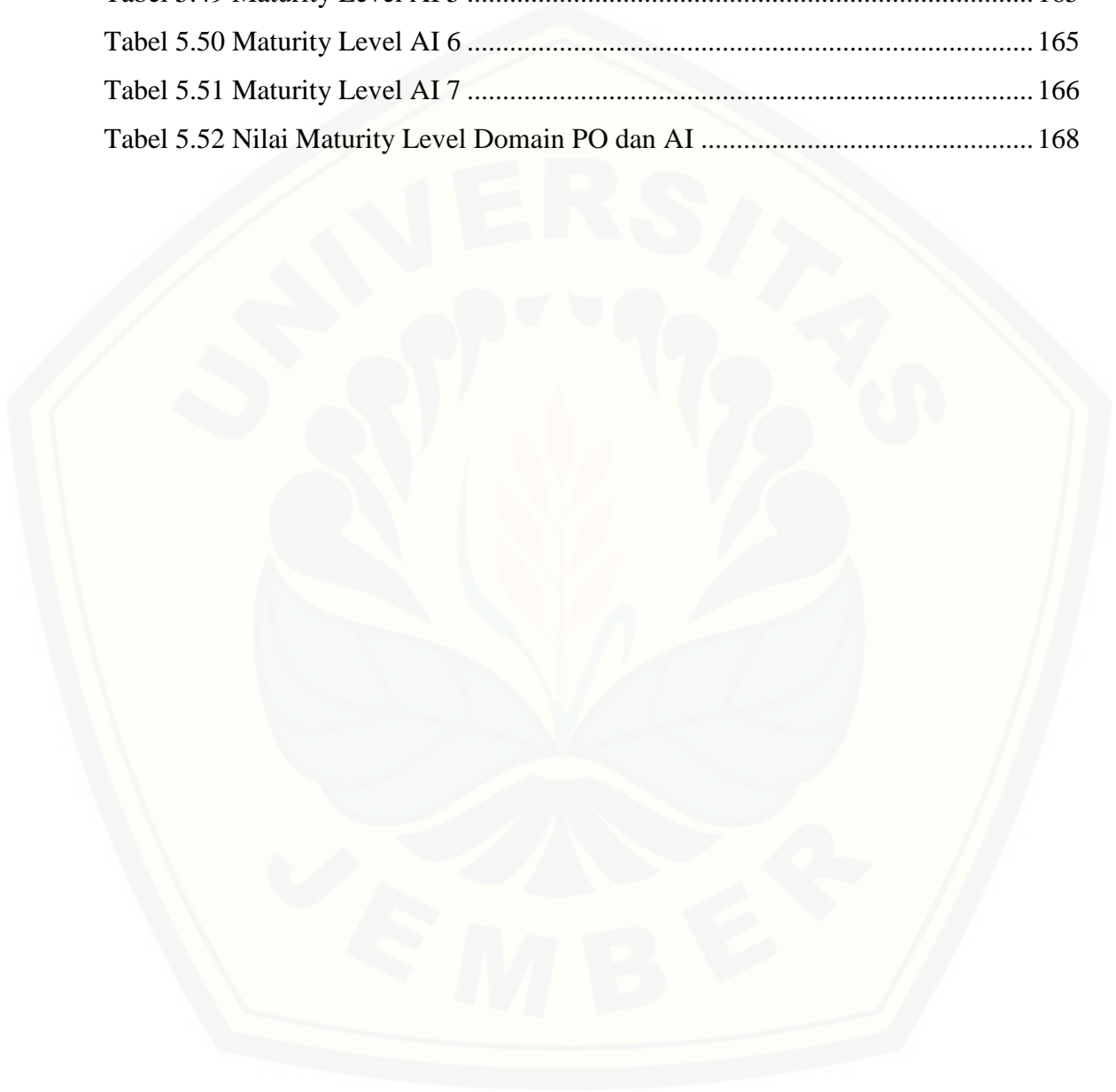
DAFTAR TABEL

Tabel 3.1 Kuesioner PO 1	22
Tabel 4.1 Definsi Aktor.....	34
Tabel 4.2 Definsi Use Case.....	35
Tabel 4.3 Skenario Use Case Mengisi Kuesioner Staff TI	37
Tabel 4.4 Skenario Use Case Mengisi Kuesioner Manajerial.....	39
Tabel 4.5 Skenario Use Case View Hasil Perhitungan	40
Tabel 4.6 Test Case Function simpan_jawaban_ya	67
Tabel 4.7 Test Case Function simpan_jawaban_tidak	68
Tabel 4.8 Test Case Function get_jawaban.....	69
Tabel 4.9 Test Case Function simpan_konversi	70
Tabel 4.10 Test Case Function set_jml_pernyataan	71
Tabel 4.11 Test Case Function simpan_jml_pernyataan	72
Tabel 4.12 Test Case Function set_avg_konversi.....	73
Tabel 4.13 Test Case Function simpan_avg_konversi.....	73
Tabel 4.14 Test Case Function set_tot_avg_konversi.....	74
Tabel 4.15 Test Case Function simpan_tot_avg_konversi	75
Tabel 4.16 Test Case Function set_jml_avg_konversi	76
Tabel 4.17 Test Case Function simpan_jml_avg_konversi	77
Tabel 4.18 Test Case Function set_normalisasi.....	78
Tabel 4.19 Test Case Function simpan_normalisasi.....	78
Tabel 4.20 Test Case Function set_kontribusi	79
Tabel 4.21 Test Case Function simpan_kontribusi.....	79
Tabel 4.22 Test Case Function hasil	80
Tabel 4.23 Test Case Function simpan_hasil.....	81
Tabel 4.24 Test Case Function get_periode.....	82

Tabel 4.25 Test Case Function simpan_periode	82
Tabel 4.26 Test Case Function simpan_jawaban_ya	83
Tabel 4.27 Test Case Function simpan_jawaban_tidak	83
Tabel 4.28 Test Case Function step1	83
Tabel 4.29 Test Case Function step2	84
Tabel 4.30 Test Case Function step3	84
Tabel 4.31 Test Case Function step4	85
Tabel 4.32 Test Case Function get_nilaico	85
Tabel 4.33 Test Case Function simpan_hasil.....	86
Tabel 4.34 Test Case Function step5	86
Tabel 4.35 Pengujian Black Box Mengisi Kuesioner	87
Tabel 4.36 Pengujian Black Box View Hasil Perhitungan	88
Tabel 5.1 Hasil Pengolahan Data R1 PO 1	93
Tabel 5.2 Hasil Pengolahan Data R1 PO 2	94
Tabel 5.3 Hasil Pengolahan Data R1 PO 3	96
Tabel 5.4 Hasil Pengolahan Data R1 PO 4	99
Tabel 5.5 Hasil Pengolahan Data R1 PO 5	101
Tabel 5.6 Hasil Pengolahan Data R1 PO 6	102
Tabel 5.7 Hasil Pengolahan Data R1 PO 7	104
Tabel 5.8 Hasil Pengolahan Data R1 PO 8	105
Tabel 5.9 Hasil Pengolahan Data R1 PO 9	106
Tabel 5.10 Hasil Pengolahan Data R1 PO 10	109
Tabel 5.11 Hasil Pengolahan Data R1 AI 1	111
Tabel 5.12 Hasil Pengolahan Data R1 AI 2	113
Tabel 5.13 Hasil Pengolahan Data R1 AI 3	114
Tabel 5.14 Hasil Pengolahan Data R1 AI 4	116
Tabel 5.15 Hasil Pengolahan Data R1 AI 5	118
Tabel 5.16 Hasil Pengolahan Data R1 AI 6	120
Tabel 5.17 Hasil Pengolahan Data R1 AI 7	122

Tabel 5.18 Hasil Pengolahan Data R2 PO 1	124
Tabel 5.19 Hasil Pengolahan Data R2 PO 2	126
Tabel 5.20 Hasil Pengolahan Data R2 PO 3	128
Tabel 5.21 Hasil Pengolahan Data R2 PO 4	130
Tabel 5.22 Hasil Pengolahan Data R2 PO 5	132
Tabel 5.23 Hasil Pengolahan Data R2 PO 6	133
Tabel 5.24 Hasil Pengolahan Data R2 PO 7	135
Tabel 5.25 Hasil Pengolahan Data R2 PO 8	136
Tabel 5.26 Hasil Pengolahan Data R2 PO 9	138
Tabel 5.27 Hasil Pengolahan Data R2 PO 10	140
Tabel 5.28 Hasil Pengolahan Data R2 AI 1	142
Tabel 5.29 Hasil Pengolahan Data R2 AI 2	144
Tabel 5.30 Hasil Pengolahan Data R2 AI 3	146
Tabel 5.31 Hasil Pengolahan Data R2 AI 4	147
Tabel 5.32 Hasil Pengolahan Data R2 AI 5	149
Tabel 5.33 Hasil Pengolahan Data R2 AI 6	151
Tabel 5.34 Hasil Pengolahan Data R2 AI 7	153
Tabel 5.35 Maturity Level PO 1	155
Tabel 5.36 Maturity Level PO 2	156
Tabel 5.37 Maturity Level PO 3	156
Tabel 5.38 Maturity Level PO 4	157
Tabel 5.39 Maturity Level PO 5	158
Tabel 5.40 Maturity Level PO 6	159
Tabel 5.41 Maturity Level PO 7	159
Tabel 5.42 Maturity Level PO 8	160
Tabel 5.43 Maturity Level PO 9	161
Tabel 5.44 Maturity Level PO 10	161
Tabel 5.45 Maturity Level AI 1	162
Tabel 5.46 Maturity Level AI 2	163

Tabel 5.47 Maturity Level AI 3	163
Tabel 5.48 Maturity Level AI 4	164
Tabel 5.49 Maturity Level AI 5	165
Tabel 5.50 Maturity Level AI 6	165
Tabel 5.51 Maturity Level AI 7	166
Tabel 5.52 Nilai Maturity Level Domain PO dan AI	168



BAB 1. PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, dan sistematika penulisan.

1.1 Latar Belakang

Universitas Jember sebagai salah satu universitas yang menerapkan Teknologi Informasi (TI) hampir pada semua aspek transaksinya. Teknologi yang diterapkan oleh Universitas Jember bertujuan untuk memberikan pelayanan proses transaksi yang efektif dan efisien. Salah satu teknologi yang telah diterapkan di Universitas Jember antara lain Sistem Informasi Terpadu (SISTER) yang digunakan untuk mengelola manajemen akademik Universitas Jember. Teknologi di Universitas Jember dikelola oleh Unit Pelaksana Teknis Teknologi Informasi (UPT TI) Universitas Jember. Pengelolaan yang dilakukan oleh pihak UPT TI berupa perencanaan mengenai hal-hal yang akan dilakukan untuk mencapai tujuannya.

Teknologi menjadi bagian yang penting dalam menjalankan kegiatan maka harus dipastikan bahwa teknologi yang berjalan pada Universitas Jember berjalan dengan baik dan lancar. Untuk mengetahui seberapa sejauh pengelolaan teknologi yang telah dijalankan dalam sebuah perusahaan sebagai bahan evaluasi untuk meningkatkan tata kelola TI kedepannya dan mengetahui hal-hal apa saja yang perlu diperbaiki. Maka perlu dilakukan analisis penilaian tingkat kematangan terhadap infrastruktur dan pengelolaan TI menggunakan metode yang tepat sehingga dapat diketahui kondisi TI yang ada.

Proses perencanaan yang selama ini telah diterapkan yaitu dengan mendiskusikan perencanaan pada sebuah pertemuan namun perencanaan yang

dilakukan tidak secara terstruktur yang didokumentasikan dan diketahui oleh semua staf. Perencanaan yang dilakukan menjadi tanggung jawab tingkat atasan. Tidak hanya pada perencanaan, tetapi juga pada penerapan dari perencanaannya. Salah satu contoh penerapan yang tidak selaras dengan perencanaannya yaitu dalam perencanaan ingin mendokumentasikan kegiatan pengembangan dan perubahan yang berlangsung namun perencanaan itu belum dapat diterapkan dengan baik bahkan untuk *Standart Operational Procedur* (SOP) sebagai dasar untuk mengatur pelaksanaan transaksi atau kegiatan belum ada. Disebabkan tidak ada dokumentasi maka kegiatan yang selama ini berlangsung, berjalan sesuai dengan komunikasi dan kebiasaan yang telah dilakukan.

Metode yang dapat digunakan untuk menilai tingkat kematangan TI adalah *framework Control Objective for Information and Related Technology* (COBIT) 4.1. COBIT adalah *framework* yang digunakan untuk manajemen proses TI yang berfokus pada kontrol objektif yang dapat mengukur pengelolaan TI suatu perusahaan menggunakan *maturity model* (ITGI, 2007). Model ini sangat tepat jika digunakan untuk melihat kesesuaian antara proses bisnis dengan pengembangan TI dalam sebuah organisasi. Apabila model ini diterapkan maka dapat diketahui apakah TI yang dijalankan pada Universitas Jember sudah sesuai dengan visi dan misi Universitas Jember. *Maturity model* yang digunakan dapat menentukan tingkat kematangan TI Universitas Jember berada pada level tertentu.

Penilaian tingkat kematangan tata kelola TI pada Universitas Jember, pada pelaksanaannya peneliti akan menggunakan *framework* COBIT 4.1 dengan domain *Plan and Organise* (PO) dan *Acquire and Implement* (AI) sesuai dengan permasalahan mengenai manajemen perencanaan dan implementasi yang kurang selaras. Pemilihan domain PO karena domain PO membahas mengenai manajemen perencanaan dan domain AI membahas mengenai solusi yang diberikan sesuai dengan perencanaan yang telah dilakukan dari domain PO. Penelitian ini akan membutuhkan data yang akan diambil dari pihak UPT TI Universitas Jember sebagai responden dari kuesioner yang akan disebar. Hasil dari analisa kebutuhan akan dibangun sebuah aplikasi yang menggambarkan alur dari kerja *framework* COBIT 4.1 berdasarkan domain PO dan AI.

1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang mendefinisikan beberapa permasalahan yang harus diselesaikan dalam penulisan ini adalah bagaimana membangun aplikasi penilaian tingkat kematangan tata kelola TI pada Universitas Jember berdasarkan domain PO dan AI *framework* COBIT 4.1?

1.3 Tujuan

Tujuan dalam penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah membangun aplikasi penilaian tingkat kematangan tata kelola TI pada Universitas Jember berdasarkan domain PO dan AI *framework* COBIT 4.1.

1.4 Batasan Masalah

Penulis memberikan batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penulisan dan pembuatan aplikasi. Berikut adalah batasan masalah yang dicantumkan:

1. Tata kelola TI yang diteliti yaitu diambil dari pihak staff TI dan manajerial Sistem Informasi Terpadu (SISTER) Universitas Jember.
2. Domain *framework* COBIT 4.1 yang digunakan pada penelitian ini adalah domain *Plan and Organise* (PO) dan *Acquire and Implement* (AI).
3. Aplikasi yang dikembangkan menampilkan nilai tingkat kematangan tata kelola TI.

1.5 Sistematika Penulisan

Sistematika penulisan dan keruntutan skripsi ini disusun sebagai berikut:

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, perumusan masalah, tujuan, dan manfaat, ruang lingkup studi dan sistematika penulisan.

2. Tinjauan Pustaka

Bab ini menjelaskan tentang materi, informasi, tinjauan pustaka, dan studi terdahulu yang menjadi kerangka pemikiran dalam penelitian.

3. Metodologi Penelitian

Bab ini menjelaskan tentang metode penelitian yang digunakan dalam penelitian.

4. Pengembangan Sistem

Bab ini menjelaskan tentang pengembangan sistem yang dikembangkan.

5. Hasil dan Pembahasan

Bab ini menjelaskan tentang hasil dan pembahasan dari penelitian yang dilakukan.

6. Penutup

Bab ini berisi tentang kesimpulan dari penelitian dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, kerangka pemikiran yang merupakan sintesis dari kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori-teori ini diambil dari buku, literature, jurnal, dan internet.

2.1 Penelitian Terdahulu

Penelitian sebelumnya yang pernah dilakukan dengan mengimplementasikan COBIT 4.1 yaitu dengan judul “Evaluasi *IT Governance* Berdasarkan Cobit 4.1 dengan studi kasus di PT Timah (Persero) Tbk” yang dilakukan oleh Dwi Rizki Kesumawardhani dari Fakultas Ekonomi Universitas Indonesia pada tahun 2012. Pada penelitian ini dijelaskan bagaimana menerapkan tata kelola TI yang baik sesuai dengan COBIT 4.1. Kemudian dengan studi kasus PT. Timah, dilakukan penelitian seberapa jauh PT. Timah telah menerapkan tata kelola TI yang sesuai dengan Cobit 4.1 yang dapat dilihat dari hasil *maturity levelnya*. Hasil dari penelitian ini adalah diketahuinya level-level dari penerapan proses TI yang ada pada PT Timah. Proses-proses tersebut memiliki nilai rata-rata 3 yang berada pada level kematangan. Pada penelitian tersebut dapat diketahui pula bahwa tidak semua rekomendasi proses menurut COBIT 4.1 dapat diterapkan pada perusahaan karena perusahaan hanya dapat menerapkan yang sesuai dengan kebutuhannya saja. Oleh karena itu, perusahaan dapat mencari *tools-tools* lainnya yang disesuaikan dengan kebutuhan perusahaan. Selain itu juga dapat diketahui kelebihan dan kekurangan yang dimiliki oleh operasional TI pada perusahaan tersebut (Kesumawardhani, 2012).

Penelitian lainnya dengan judul “Penilaian Tingkat Kematangan Tata Kelola TI Pada Sistem Informasi Manajemen Akademik” oleh Fitroh dari jurusan Sistem

Informasi Fakultas Sains dan Teknologi Universitas Islam Negeri Jakarta pada tahun 2012. Pada penelitian ini dilakukan penilaian tingkat kematangan tata kelola TI pada sistem informasi akademik menggunakan framework COBIT 4.0 yang berfokus pada domain *Planning and Organise* (PO) dan *Acquire and Implement* (AI). Hasil dari penelitian ini yaitu Sistem Informasi Manajemen Akademik berada pada level 2 (*repeatable but intuitive*) untuk domain PO dan level 1 (*initial/ad-hoc*) untuk domain AI. Sedangkan *maturity* berada pada level 3 (*define process*) dan dapat dijadikan rekomendasi untuk setiap *control objective* yang memiliki *gap maturity level* berdasarkan *control objective* (Fitroh, 2012).

Berdasarkan penelitian tersebut dapat disimpulkan bahwa dalam penilaian tata kelola TI menggunakan *framework* COBIT dapat diketahui tingkat kematangan dari TI berada pada level tertentu. Kemudian dalam melakukan analisa, dapat menggunakan beberapa domain dari *framework* COBIT. Jadi tidak harus selalu menggunakan 4 domain yang dimiliki oleh *framework* COBIT. Pemilihan domain yang digunakan untuk penelitian disesuaikan dengan tujuan dari penelitian.

2.2 Tata Kelola TI

TI merupakan faktor yang penting dalam pencapaian sukses pada ekonomi informasi dan pusat pada sebuah kesatuan operasional dan manajemen keuangan (ITGI, 2004). Penerapan TI pada suatu perusahaan memerlukan biaya yang cukup besar disertai dengan kemungkinan resiko kegagalan yang cukup besar pula, sehingga penerapan TI dapat dilakukan dengan baik apabila ditunjang dengan suatu tata kelola dari mulai perencanaan sampai dengan implementasinya (Surendro, 2008).

Tata kelola TI adalah suatu struktur hubungan dan proses untuk mengatur dan mengontrol perusahaan yang bertujuan untuk mencapai tujuan perusahaan yang telah ditetapkan dengan pertambahan nilai dengan tetap menyeimbangkan resiko-resiko dengan nilai yang didapatkan dari penerapan TI dan proses-prosesnya (*IT Governance*

Institute dalam Surendro, 2008). Tata kelola TI bukan bidang yang terpisah dari pengelolaan perusahaan, melainkan merupakan komponen pengelolaan perusahaan secara keseluruhan, dengan tanggung jawab sebagai berikut:

1. Memastikan kepentingan *stakeholder* diikutsertakan dalam penyusunan strategi perusahaan.
2. Memberikan arahan-arahan kepada proses-proses yang menerapkan strategi perusahaan.
3. Memastikan proses-proses tersebut menghasilkan keluaran yang terukur.
4. Memastikan adanya informasi mengenai hasil yang diperoleh dan mengukurnya.
5. Memastikan keluaran yang dihasilkan sesuai dengan yang diharapkan. (*IT Governance Institute* dalam Surendro, 2008)

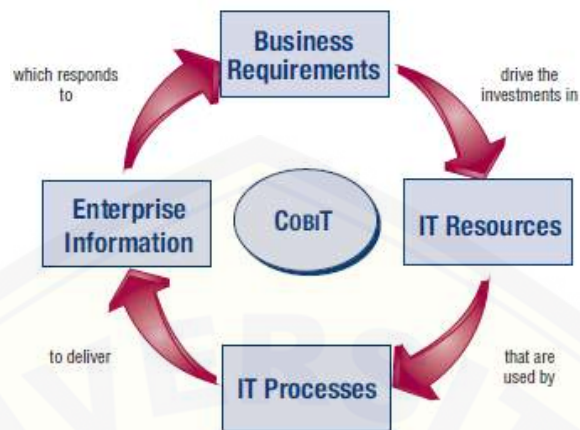
Tata kelola TI menyediakan struktur yang menghubungkan IT processes, IT resources serta informasi untuk tujuan dan strategi perusahaan (ITGI, 2004).

2.3 Framework COBIT

COBIT adalah sebuah framework yang merupakan kumpulan dari alat yang mendukung manajer untuk menjembatani *gap* antara kebutuhan kontrol, persoalan teknis dan resiko bisnis, dan mengomunikasikan level dari control ke *stakeholder* (ITGI, 2007). Framework COBIT adalah model Proses Oriented (Zandhessami & Kargar, 2010).

2.3.1 Prinsip Dasar COBIT

Prinsip dasar COBIT meliputi kebutuhan bisnis, sumber daya TI, proses TI, dan informasi perusahaan. Prinsip dasar COBIT selengkapnya dapat dilihat pada Gambar 2.1. Gambar 2.1

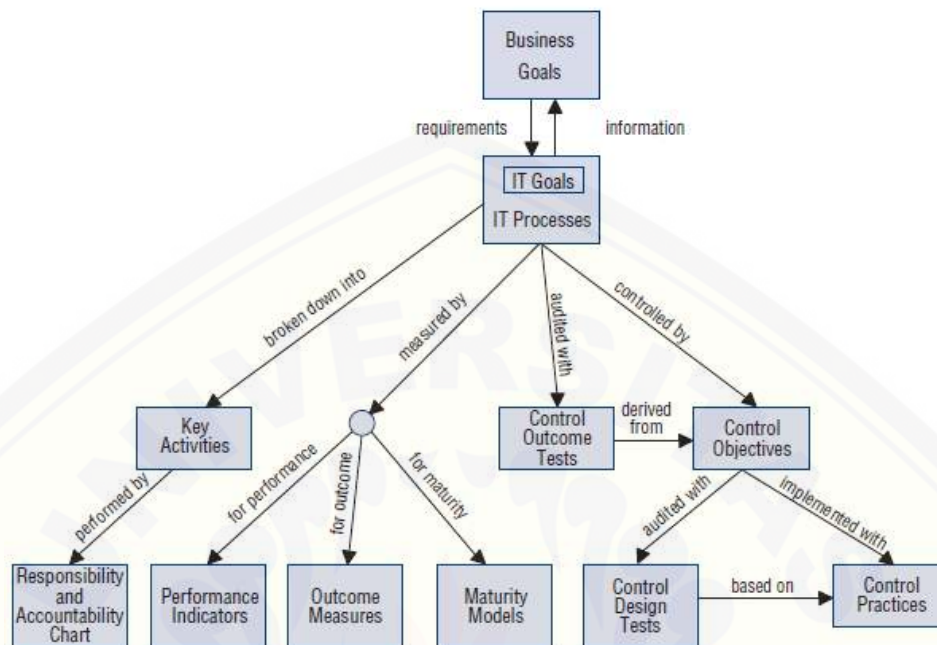


Gambar 2.1 Prinsip Dasar COBIT (ITGI, 2007)

COBIT memiliki komponen-komponen, yaitu:

1. *Board Briefing on IT Governance*
2. *Management Guidelines/maturity models*
3. *Frameworks*
4. *Control objectives*
5. *IT Governance Implementation Guide*
6. *COBIT Control Practices*
7. *IT Assurance Guide*

Semua komponen dari COBIT terhubung, menyediakan dukungan terhadap perusahaan, manajemen, control dan kebutuhan jaminan dari pengguna yang berbeda, seperti yang digambarkan dalam Gambar 2.2.



Gambar 2.2 Hubungan dari Komponen-komponen COBIT (ITGI, 2007)

COBIT mendefinisikan aktifitas TI dalam model proses umum dengan 4 domain, yaitu (ITGI, 2007):

1. *Plan and Organise (PO)*

Domain PO memberikan pengarahannya untuk solusi yang berhubungan dengan domain AI dan pelayanan yang berhubungan dengan domain DS seperti dijelaskan pada Gambar 2.3. Domain ini mencakup strategi dan taktik dan menyangkut identifikasi kontribusi TI terhadap tujuan pencapaian bisnis. Perealisisian visi strategis dari sebuah perusahaan perlu direncanakan, dikomunikasikan, dan dikelola agar dapat mencapai tujuan bisnis sesuai yang diharapkan. Ada beberapa hal yang dibahas domain ini, meliputi keselarasan antara TI dan strategi bisnis, optimalisasi penggunaan sumber daya oleh organisasi, pemahaman setiap orang yang ada dalam perusahaan, pengelolaan dan pemahaman mengenai risiko TI, dan kualitas TI yang tepat untuk kebutuhan bisnis.

2. *Acquire and Implement (AI)*

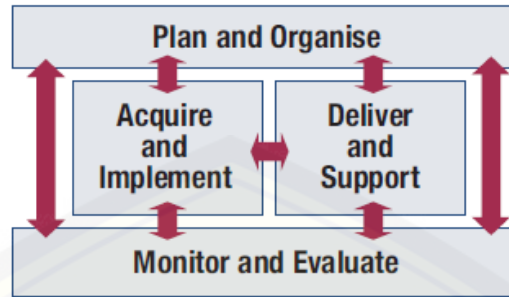
Solusi TI perlu diidentifikasi, dikembangkan atau diperoleh, serta diimplementasikan dan diintegrasikan dalam proses bisnis untuk merealisasikan strategi TI. Domain ini memberikan solusi terhadap perubahan dan pemeliharaan sistem yang terjadi pada perusahaan. Ada beberapa hal yang dibahas domain ini, meliputi keberadaan proyek baru yang memberikan solusi untuk memenuhi kebutuhan bisnis, pengadaan proyek baru yang tepat waktu sesuai dengan anggaran perusahaan, cara kerja sistem baru saat diimplementasikan, dan kemungkinan perubahan yang dilakukan dapat mengganggu proses bisnis.

3. *Deliver and Support (DS)*

Domain ini berkaitan dengan pengiriman aktual dari layanan yang diperlukan, meliputi pelayanan, pengelolaan keamanan dan kontinuitas, dukungan layanan untuk pengguna, dan penggunaan data dan fasilitas operasional. Ada beberapa hal yang dibahas dalam domain ini, meliputi kesesuaian antara layanan TI dan prioritas bisnis, optimalisasi biaya TI, kemampuan tenaga kerja perusahaan untuk menggunakan sistem TI secara produktif dan aman, dan ketersediaan tempat untuk keamanan informasi.

4. *Monitor and Evaluate (ME)*

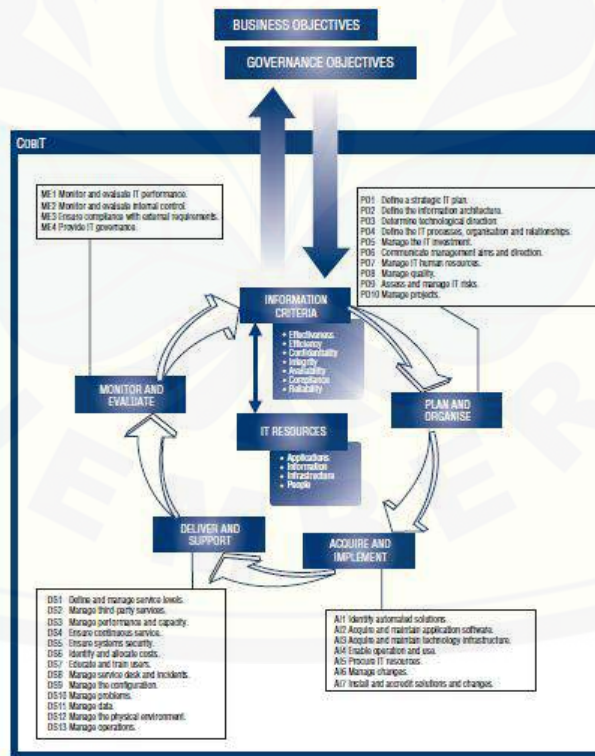
Semua proses TI perlu untuk dinilai secara berkala dari waktu ke waktu untuk mengetahui kualitas dan kepatuhan terhadap kebutuhan kontrol. Ada beberapa hal yang dibahas domain ini, meliputi pengukuran kinerja TI dalam mendeteksi masalah, memastikan pengendalian internal yang efektif dan efisien, menghubungkan kinerja TI dengan tujuan bisnis kembali, dan ketersediaan tempat untuk keamanan informasi. Hubungan antara keempat domain tersebut dapat dilihat pada Gambar 2.3.



Gambar 2.3 Hubungan 4 Domain COBIT (ITGI, 2007)

2.3.2 Kontrol Objektif

Setiap domain dari COBIT memiliki masing-masing kontrol objektif. Jumlah kontrol objektif pada setiap domain berbeda seperti yang dijelaskan pada Gambar 2.4.



Gambar 2.4 Keseluruhan Framework COBIT 4.1 (ITGI, 2007)

Kontrol objektif sesuai dengan Gambar 2.4 adalah sebagai berikut (ITGI, 2007):

1. *Plan and Organise* (PO)

Domain PO merupakan domain yang digunakan dalam penelitian ini dan memiliki 10 objektif kontrol, yaitu:

- a. PO1 : *Define a strategic IT plan* (Menetapkan rencana strategis TI).
- b. PO2 : *Define the information architecture* (Menetapkan arsitektur informasi).
- c. PO3 : *Determine technological direction* (Menentukan arahan teknologi).
- d. PO4 : *Define the IT processes, organisation and relationships* (Menetapkan proses TI, organisasi dan keterhubungannya).
- e. PO5 : *Manage the IT investment* (Mengelola investasi TI).
- f. PO6 : *Communicate management aims and direction* (Mengkomunikasikan tujuan dan arahan manajemen).
- g. PO7 : *Manage IT human resources* (Mengelola sumber daya TI).
- h. PO8 : *Manage quality* (Mengelola kualitas).
- i. PO9 : *Asses and manage IT risks* (Menaksir dan mengelola resiko TI).
- j. PO10 : *Manage projects* (Mengelola proyek).

2. *Acquire and Implement* (AI)

Domain AI merupakan domain yang digunakan dalam penelitian ini dan memiliki 7 objektif kontrol, yaitu:

- a. AI1 : *Identify automated solutions* (Mengidentifikasi solusi otomatis).
- b. AI2 : *Acquire and mantain application software* (Memperoleh dan memelihara perangkat lunak aplikasi).
- c. AI3 : *Acquire and mantain technology infrastructure* (Memperoleh dan memelihara infrastruktur teknologi).
- d. AI4 : *Enable operation and use* (Memungkinkan operasional dan penggunaan).
- e. AI5 : *Procure IT resources* (Memenuhi sumber daya TI).
- f. AI6 : *Manage changes* (Mengelola perubahan).

- g. AI7 : *Install and accredit solutions and changes* (Instalasi dan akreditasi solusi beserta perubahannya).

3. *Deliver and Support* (DS)

Domain DS memiliki 13 kontrol objektif, yaitu:

- a. DS1 : *Define and manage service levels* (Menetapkan dan mengelola tingkat layanan).
- b. DS2 : *Manage third-party services* (Mengelola layanan pihak ketiga).
- c. DS3 : *Manage performance and capacity* (Mengelola kinerja dan kapasitas).
- d. DS4 : *Ensure continous service* (Memastikan layanan berkelanjutan).
- e. DS5 : *Ensure systems security* (Memastikan sistem keamanan).
- f. DS6 : *Identify and allocated costs* (Mengidentifikasi dan mengalokasikan biaya).
- g. DS7 : *Educate and train users* (Mengajarkan pengguna).
- h. DS8 : *Manage service desk and incidents* (Mengelola layanan dan insiden).
- i. DS9 : *Manage the configuration* (Mengelola informasi).
- j. DS10 : *Manage problems* (Mengelola masalah).
- k. DS11 : *Manage data* (Mengelola data).
- l. DS12 : *Manage the physical environment* (Mengelola lingkungan fisik).
- m. DS13 : *Manage operations* (Mengelola operasi).

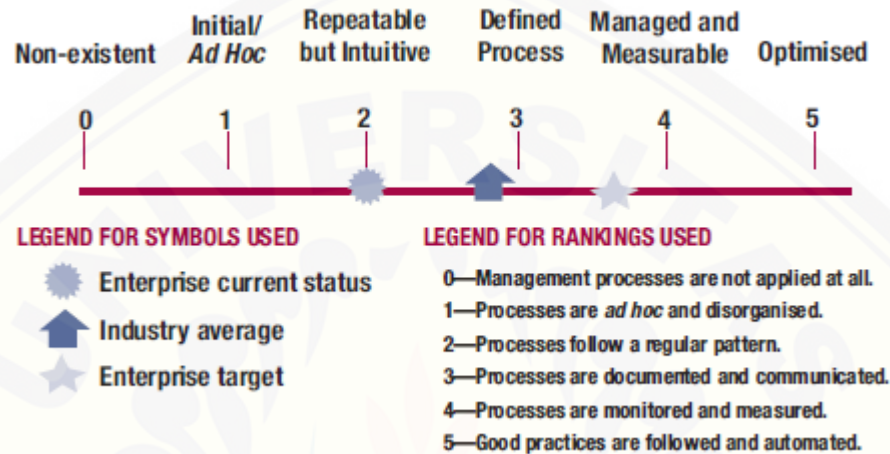
4. *Monitor and Evaluate* (ME)

Domain ME memiliki 4 kontrol objektif, yaitu:

- a. ME1 : *Monitor and evaluate IT performance* (Memantau dan mengevaluasi kinerja TI).
- b. ME2 : *Monitor and evaluate internal control* (Memantau dan mengevaluasi pengendalian eksternal).
- c. ME3 : *Ensure compliance with external requirements* (Memastikan kepatuhan dengan ketaatan eksternal).
- d. ME4 : *Provide IT governance* (Menyediakan *IT governance*)

2.3.4 Maturity Level

Maturity Models adalah model penilaian untuk proses TI yang terdiri dari metode perhitungan dari skala 0 (*nonexistent*) sampai 5 (*optimised*) (ITGI, 2004). Skala *maturity model* dijelaskan pada Gambar 2.5.



Gambar 2.5 Representasi Grafik *Maturity Model* (ITGI, 2007)

Tahapan perhitungan *maturity level* dilakukan dengan mengonversi pernyataan pada kuisisioner, menghitung rata-rata konversi, normalisasi, dan nilai kontribusinya (Pederiva, 2003). Perhitungan ini dilakukan pada setiap level pada objektif kontrol untuk setiap responden. Selanjutnya nilai kematangan dari setiap objektif kontrol dari setiap responden akan dirata-rata sehingga menjadi nilai kematangan sebuah domain.

1. Konversi Jawaban Pernyataan Kuisisioner

Jawaban pernyataan dari kuisisioner yang telah diperoleh, diberi nilai sesuai dengan standart yang ada. Pilihan jawaban yang digunakan pada penelitian ini yaitu ya (*completely*) dan tidak (*not at all*) (Pederiva, 2003). Nilai untuk jawaban setiap responden dapat dilihat pada Gambar 2.6.

Agreement with Statement	Compliance Value
Not at all	0
A little	0.33
Quite a lot	0.66
Completely	1

Gambar 2.6 Nilai Konversi Pernyataan (Pederiva, 2003)

2. Rata-rata Konversi

Perhitungan rata-rata konversi dilakukan pada setiap level di tiap kontrol objektif dari tiap responden. Rumus perhitungan konversi dapat dilihat pada Rumus 2.1.

$$\text{avgKonversi} = \frac{\text{konversi}}{\text{jmlPertanyaan}} \quad \dots (2.1)$$

Keterangan:

avgKonversi = rata-rata konversi dari jawaban tiap pernyataan kuisioner

konversi = nilai konversi jawaban tiap pernyataan

jmlPernyataan = jumlah pernyataan yang pada setiap level

Setelah menghitung rata-rata konversi pada setiap pernyataan di tiap level, rata-rata dari semua pernyataan di tiap level dijumlahkan. Rumus perhitungan jumlah rata-rata konversi dapat dilihat pada Rumus 2.2.

$$\text{jmlAvgKonversi} = \sum \text{avgKonversi} \quad \dots (2.2)$$

Keterangan:

jmlAvgKonversi = jumlah rata-rata konversi semua pernyataan di tiap level

avgKonversi = rata-rata konversi dari jawaban tiap pernyataan kuisioner

Setelah mendapatkan hasil dari jumlah rata-rata konversi setiap level, kemudian jumlah rata-rata konversi di setiap level dijumlahkan dengan level yang lain pada setiap domain dari setiap responden. Rumus perhitungan total rata-rata konversi dapat dilihat pada Rumus 2.3.

$$\text{totAvgKonversi} = \sum \text{jmlAvgKonversi} \quad \dots (2.3)$$

Keterangan:

totAvgKonversi = jumlah rata-rata konversi dari semua level

jmlAvgKonversi = jumlah rata-rata konversi semua pernyataan di tiap level

3. Normalisasi

Normalisasi yaitu mencari nilai untuk setiap level dengan menghitung hasil jumlah rata-rata konversi setiap level yang dibagi dengan jumlah rata-rata konversi dari semua level. Rumus perhitungan normalisasi dapat dilihat pada Rumus 2.4.

$$\text{normalisasi} = \frac{\text{jmlAvgKonversi}}{\text{totAvgKonversi}} \quad \dots (2.4)$$

Keterangan:

normalisasi = nilai untuk setiap level

jmlAvgKonversi = jumlah rata-rata konversi semua pernyataan di tiap level

totAvgKonversi = jumlah rata-rata konversi di setiap level

4. Kontribusi

Nilai kontribusi merupakan nilai pencapaian dari setiap level yang nantinya akan dijumlahkan sehingga menghasilkan nilai kematangan (*maturity level*) untuk setiap domain dari setiap responden (Rosalika). Rumus perhitungan nilai kontribusi dapat dilihat pada Rumus 2.5.

$$\text{kontribusi} = \text{normalisasi} \times \text{level} \quad \dots (2.5)$$

Keterangan:

kontribusi = nilai pencapaian setiap level

normalisasi = nilai untuk setiap level

level = level dari domain setiap kontrol objektif

Setelah diketahui nilai kontribusi dari masing-masing level, nilai kontribusi tersebut dijumlahkan untuk mendapatkan nilai kematangan dari setiap kontrol objektif tiap responden. Rumus perhitungan *maturity level* dari setiap responden dapat dilihat pada Rumus 2.6.

$$\text{maturityRes} = \sum \text{kontribusi} \quad \dots (2.6)$$

Keterangan:

maturityRes = nilai kematangan dari setiap kontrol objektif tiap responden

kontribusi = nilai pencapaian setiap level

5. Maturity Level

Nilai kematangan domain dapat diketahui dari penjumlahan nilai kematangan semua kontrol objektif dari semua responden yang dibagi dengan jumlah kontrol objektif setiap domain. Rumus perhitungan *maturity level* dari setiap kontrol objektif dapat dilihat pada Rumus 2.7.

$$\text{maturityOC} = \sum \text{maturityRes} \quad \dots (2.7)$$

Keterangan:

maturityOC = jumlah dari maturity level semua kontrol objektif

maturityRes = nilai kematangan dari setiap kontrol objektif tiap responden

Setelah *maturity level* dari setiap kontrol objektif dari setiap responden ditemukan, maka dapat dihitung *maturity level* dari setiap domain. Rumus perhitungan *maturity level* dari setiap domain dapat dilihat pada Rumus 2.8.

$$\text{maturity} = \frac{\sum \text{maturityOC}}{\text{jmIOC}} \quad \dots (2.8)$$

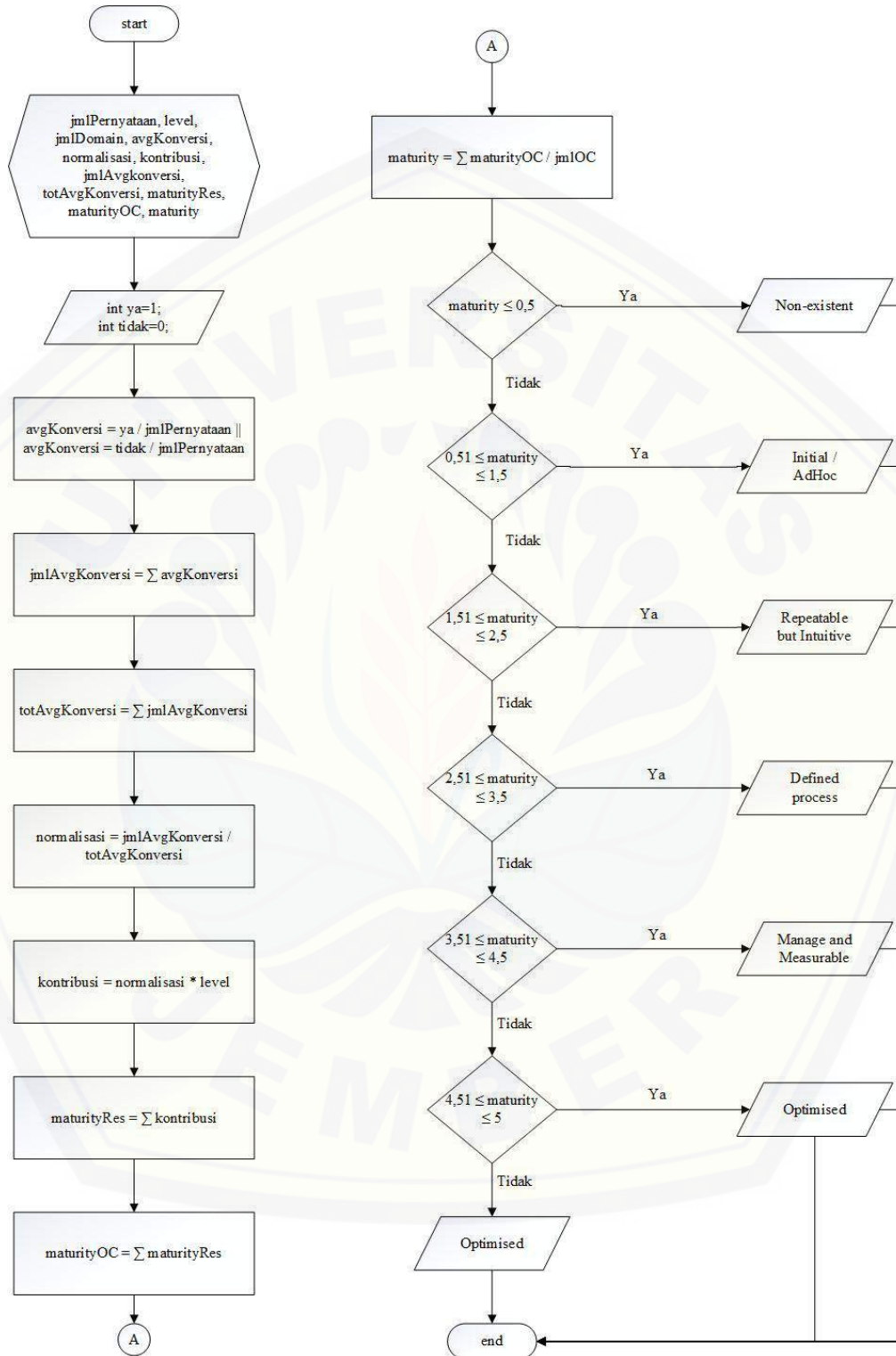
Keterangan :

maturity = nilai kematangan domain

maturityOC = jumlah dari maturity level semua kontrol objektif

jmIOC = jumlah kontrol objektif setiap domain

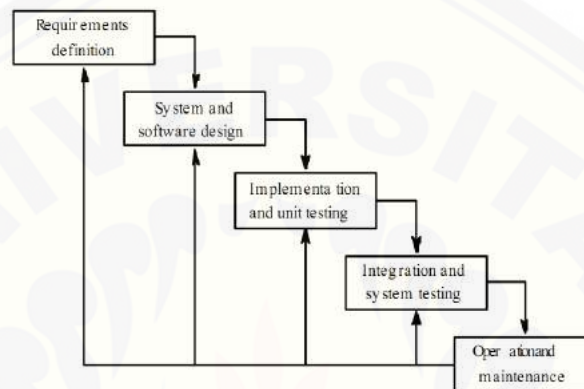
Untuk lebih lengkapnya, perhitungan nilai kematangan tersebut dapat dijelaskan pada *flowchart* Gambar 2.7.



Gambar 2.7 Flowchart Perhitungan Nilai Kematangan

2.4 Model Waterfall

Model *waterfall* merupakan metode yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test dan pemeliharaan (Sommerville, 2003). Gambaran model *waterfall* dapat dilihat pada Gambar 2.8.



Gambar 2.8 Model *Waterfall* (Sommerville, 2003)

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tahapan dari penelitian.

3.1 Tahapan Penelitian

Penelitian mengenai pembangunan aplikasi implementasi *framework* COBIT untuk menentukan nilai kematangan tata kelola teknologi informasi pada Universitas Jember ini menggunakan *System Development Life Cycle (SDLC) waterfall* seperti yang telah dijelaskan pada bab tinjauan pustaka subbab 2.4. Pengembangan aplikasi ini menggunakan model *waterfall* karena disesuaikan dengan kebutuhan aplikasi. Aplikasi yang akan dibangun merupakan aplikasi dalam skala yang kecil, Sumber Daya Manusia (SDM) yang membangun aplikasi hanya 1 orang, dan menyesuaikan dengan *user* yang akan menjadi pengguna aplikasi.

3.1.1 Tahapan Analisis Kebutuhan

Penelitian yang dilakukan pada UPT TI Universitas Jember yang beralamat di Jalan Kalimantan No. 37 Kampus Tegalboto Jember, Jawa Timur. Penelitian yang dilaksanakan dalam waktu 2 bulan dimulai pada bulan Februari 2016 sampai dengan bulan April 2016. Tahapan analisis kebutuhan pada penelitian yang telah dilakukan tersebut dimulai dengan tahap pengumpulan data meliputi studi pustaka, observasi, dan menyebarkan kuesioner. Tahapan yang selanjutnya yaitu tahap pengolahan data dan gambaran umum sistem yang akan dibangun.

1. Tahap Pengumpulan Data

Tahap pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian. Pengumpulan data pada penelitian ini dilakukan dalam beberapa tahap pengumpulan data sebagai berikut:

a. Studi Literatur

Studi pustaka ini dilakukan dengan tujuan sebagai dasar pembahasan penyusunan dasar teori yang digunakan dalam penelitian. Sumber yang digunakan sebagai studi literatur berupa buku, jurnal, dan karya ilmiah yang berkaitan dengan *framework* COBIT 4.1, domain *Plan and Organize* (PO), domain *Acquire and Implement* (AI), *maturity level*, dan perhitungan *maturity level*.

b. Observasi

Observasi dilakukan dengan datang ke UPT TI Universitas Jember. Observasi dilakukan dengan tujuan untuk mengetahui TI yang ada di Universitas Jember dan mengetahui pihak-pihak yang berkaitan dengan TI di Universitas Jember. Pihak-pihak tersebut yang nantinya akan menjadi responden dari penyebaran kuesioner. Data responden berdasarkan hasil observasi yaitu sekretaris UPT TI sebagai pihak manajerial dan programmer di UPT TI sebagai staf TI.

c. Menyebarakan Kuesioner

Kuesioner menjadi media untuk mengetahui sejauh mana pengetahuan dan efek yang dirasakan oleh responden terhadap TI di Universitas Jember. Kuesioner yang digunakan dalam penelitian ini menggunakan kuesioner dari penelitian yang berjudul *Penilaian Tata Kelola Teknologi Informasi Pada Aplikasi CSBO dengan Menggunakan Framework COBIT 4.0 Domain PO dan AI* oleh Mirza Hasan Siraji (Siraji, 2011) sehingga penulis tidak melakukan uji validitas terhadap kuesioner tersebut. Proses untuk memahami kuesioner dan memperdalam pemahaman dari kuesioner tersebut menggunakan literatur dari buku COBIT 4.1 (ITGI, 2007). Beberapa bagian dari kuesioner yang akan disebarakan dapat dilihat pada Tabel 3.1 sedangkan untuk kuesioner yang selengkapnya dapat dilihat pada lampiran A. Hasil

dari kuesioner ini akan menjadi parameter dalam menentukan nilai kematangan (*maturity level*) tata kelola TI pada Universitas Jember.

Tabel 3.1 Kuesioner PO 1 (Siraji, 2011)

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Perencanaan strategi tidak bekerja.		
	2. Tidak ada kesadaran manajemen bahwa perencanaan strategi TI diperlukan untuk mendukung tujuan-tujuan bisnis.		
1 <i>Initial/Ad- hoc</i>	1. Kebutuhan untuk perencanaan strategi TI dikenal oleh manajemen TI.		
	2. Perencanaan TI dilakukan sebagai kebutuhan dasar sebagai jawaban atas kebutuhan perusahaan tertentu.		
	3. Perencanaan strategi TI adakalanya didiskusikan pada pertemuan manajemen TI.		
	4. Penjajaran kebutuhan bisnis, aplikasi, dan teknologi terjadi dengan reaktif dibanding dengan strategi seluruh perusahaan.		
	5. Posisi resiko strategi dikenali dengan tidak formal dalam proyek ke proyek.		
2 <i>Repeatable but Intuitive</i>	1. Perencanaan strategi TI terbagi dengan manajemen bisnis sebagai kebutuhan dasar.		
	2. Membaharui rencana TI terjadi sebagai jawaban atas permintaan manajemen.		
	3. Keputusan yang strategis digerakkan dalam proyek ke proyek, tanpa konsistensi dengan seluruh strategi perusahaan.		

LEVEL	KRITERIA	YA	TIDAK
	4. Resiko-resiko dan keuntungan-keuntungan <i>user</i> dari keputusan strategi utama diakui dengan cara intuitif.		
3 <i>Define Process</i>	1. Sebuah kebijakan menjelaskan kapan dan bagaimana untuk melakukan perencanaan strategi TI.		
	2. Perencanaan strategi TI mengikuti pendekatan yang terstruktur yang didokumentasikan dan diketahui oleh semua staff.		
	3. Proses perencanaan TI layak bunyi dan menjamin bahwa perencanaan sesuai seperti yang dilakukan, namun kebijaksanaan diberikan kepada manajer individual berkenaan dengan proses implementasi dan tidak ada prosedur untuk menguji proses.		
	4. Seluruh strategi TI meliputi penjelasan yang konsisten dari resiko-resiko yang mana perusahaan rela ambil sebagai sebuah pembaharuan atau penyokong.		
	5. Strategi teknis, keuangan, dan sumber daya manusia sangat mempengaruhi tambahan produk dan teknologi baru.		
	6. Perencanaan strategi TI didiskusikan saat pertemuan manajemen bisnis.		
4	1. Perencanaan strategi TI adalah praktik standart dan eksepsi yang diumumkan oleh manajemen.		

LEVEL	KRITERIA	YA	TIDAK
<i>Manage and Measurable</i>	2. Perencanaan strategi TI adalah fungsi manajemen yang baik dengan tanggung jawab tingkat atasan.		
	3. Manajemen dapat mengawasi proses perencanaan strategi TI, membuat perencanaan jelas dan mengukur keefektifannya.		
	4. Perencanaan TI jangka pendek dan jangka panjang terjadi dan mengalir ke bawah dalam perusahaan dengan pembaharuan yang dilakukan seperti yang dibutuhkan.		
	5. Strategi TI dan strategi seluruh perusahaan semakin menjadi lebih serasi dengan menunjukkan proses bisnis dan kemampuan nilai tambah dan mengungkit penggunaan aplikasi dan teknologi melalui merekayasa ulang proses bisnis.		
	6. Ada proses yang baik untuk menentukan penggunaan sumber internal dan eksternal yang dibutuhkan dalam operasi dan pengembangan sistem.		
	5	1. Perencanaan strategi TI terdokumentasi, proses hidup secara terus-menerus dipertimbangkan dalam <i>setting</i> tujuan bisnis dan berakibat pada nilai bisnis yang terlihat melalui investasi dalam TI.	

LEVEL	KRITERIA	YA	TIDAK
<i>Optimised</i>	2. Pertimbangan nilai tambah dan resiko terus-menerus diperbaharui dalam proses perencanaan strategi TI.		
	3. Perencanaan TI realistis dikembangkan dan tetap diperbaharui untuk mencerminkan perubahan teknologi dan pengembangan terkait bisnis.		
	4. <i>Benchmarking</i> terhadap pemahaman yang baik dan norma-norma industri dapat dipercaya terjadi dan terintegrasi dengan proses perumusan strategi.		
	5. Perencanaan yang strategis meliputi bagaimana pengembangan teknologi baru dapat menggerakkan penciptaan kemampuan bisnis baru dan meningkatkan keuntungan kompetitif perusahaan.		

2. Tahap Pengolahan Data

Tahap pengolahan data dimulai dengan menelaah data secara keseluruhan yang telah diperoleh dari tahap pengumpulan data. Data yang telah terkumpul digunakan untuk mengidentifikasi kebutuhan fungsional dan kebutuhan non-fungsional sistem. Data hasil kuesioner dikelompokkan sesuai dengan domain *Acquire and Implement* (AI) dan *Plan and Organize* (PO) sampai menentukan nilai kematangan (*maturity level*). Data yang diperoleh berupa jawaban dari pernyataan di kuesioner. Jawaban tersebut dibagi menjadi dua, yaitu jawaban dari responden 1 (staff TI) dan responden 2 (manajerial). Jawaban tersebut kemudian diolah sesuai dengan tahapan dan rumusnya seperti yang telah dijelaskan pada tinjauan pustaka 2.3.4. setelah

pengelompokkan data, langkah selanjutnya yaitu mengonversikan jawaban, menghitung nilai normalisasi, dan terakhir menghitung nilai *maturity level* pada setiap kontrol objektif dan nilai *maturity level* secara keseluruhan (dari domain PO dan AI).

3. Gambaran Umum Sistem

Aplikasi yang akan dibuat yaitu aplikasi penilaian tata kelola teknologi informasi berdasarkan domain *Plan and Organize* (PO) dan *Acquire and Implement* (AI) *framework* COBIT 4.1 berbasis web dengan studi kasus yaitu Sistem Informasi Terpadu (SISTER) Universitas Jember. Aplikasi ini dibangun dengan tujuan untuk mempermudah suatu organisasi dalam mengetahui tingkat kematangan tata kelola teknologi informasinya. Aktor yang terlibat dalam aplikasi ini yaitu admin yang bertugas untuk memanjamen *user*, pertanyaan kuesioner, dan periode. Kuesioner yang ada akan diisi oleh para responden, seperti staff TI dan pihak manajerial. Hasil dari penilaian tingkat kematangan tata kelola teknologi informasi tersebut dapat dilihat oleh Pembantu Rektor I Universitas Jember.

3.1.2 Tahapan Desain Sistem

Tahap yang selanjutnya yaitu desain sistem menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang akan digunakan sebagai berikut:

1. *Business Process*

Business process digunakan untuk menggambarkan masukan data yang dibutuhkan oleh sistem (*input*), keluaran data yang dihasilkan oleh sistem (*output*), media dari sistem (*uses*), dan tujuan dari pembuatan sistem (*goal*). *Input* yang dibutuhkan berupa data *user*, data domain, data kontrol objektif, data *maturity level*, data kuesioner, dan data periode. *Output* yang dihasilkan oleh sistem berupa rekap data

user, rekap data domain, rekap data kontrol objektif, rekap data *maturity level*, rekap data kuesioner, rekap data periode, dan hasil perhitungan *maturity level*.

2. Use Case Diagram

Use case diagram merupakan dokumentasi yang menggambarkan fitur dan aktor yang dapat mengakses fitur tersebut pada sistem yang akan dibangun. Aktor yang dapat mengakses aplikasi yaitu admin, staf TI, manajerial, dan Pembantu Rektor I. Fitur yang disediakan oleh aplikasi yaitu fitur login, pengelolaan data *user*, pengelolaan data domain, pengelolaan data kontrol objektif, pengelolaan data *maturity level*, pengelolaan data kuesioner, pengelolaan data periode, melihat domain dan kontrol objektif COBIT 4.1, mengisikan kuesioner, dan melihat hasil perhitungan *maturity level*.

3. Use Case Scenario

Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *use case diagram* dan menjelaskan keadaan yang akan terjadi pada suatu *event* tertentu. *Use case scenario* menggambarkan keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada suatu kondisi tertentu.

4. Sequence Diagram

Sequence diagram adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antarobjek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu. Interaksi antarobjek tersebut meliputi, aktor yang mengakses fitur tersebut sesuai dengan yang ada pada *use case diagram*, tampilan sistem, kontroller, model, dan pesan yang disampaikan jika terjadi suatu *event* tertentu.

5. Activity Diagram

Activity diagram menggambarkan alur aktivitas dalam sistem yang sedang dirancang, meliputi awal alur terjadi, *decision* yang terjadi, dan bagaimana alur berakhir.

6. Class Diagram

Class diagram adalah sebuah spesifikasi yang menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram*

menggambarkan kelas, *function*, dan atribut yang dibutuhkan oleh seorang programmer dalam membangun sistem.

7. *Entity Relationship Diagram* (ERD)

Entity relationship diagram merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antarrelasi.

3.1.3 Tahapan Implementasi Sistem

Pada tahap ini desain yang telah dibuat akan diimplementasikan ke dalam kode program. Hal yang dilakukan dalam tahap implementasi antara lain:

- a. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor* (PHP) dengan bantuan *framework Code Igniter* (CI).
- b. Manajemen basisdata menggunakan MySQL.

3.1.4 Tahapan Pengujian Sistem

Pengujian aplikasi dilakukan dengan 2 cara yaitu:

1. *White Box Testing* merupakan cara pengujian dengan melihat modul yang telah dibuat dengan program yang ada. Pengujian *white box* menggunakan *cyclomatic Complexity* (CC) atau biasa disebut juga dengan *conditional complexity*. *Cyclomatic complexity* adalah alat pengukuran untuk mengindikasikan kekompleksitasan dari program dengan cara menelusuri nomor dari jalur yang independen melalui *source codenya*. Dikembangkan oleh Thomas J. McCabe, Sr. pada tahun 1976, *cyclomatic complexity* digunakan di semua fase pada *software lifecycle*, dimulai dari fase desain untuk menjaga agar software dapat dipercaya dan mudah untuk di tes, dan juga agar bisa terorganisir. CC merupakan metrik tradisional yang menghitung tingkat kompleksitas suatu *method/procedure*. Metrik ini bisa diterapkan pada pemrograman berorientasi objek untuk menghitung kompleksitas suatu method. CC secara langsung tidak bisa digunakan untuk mengukur kompleksitas kelas, karena

adanya pewarisan dalam code. Namun CC bisa menghitung kompleksitas kelas jika dikombinasikan dengan pengukuran lain. Rumus CC dapat dilihat pada rumus 3.1.

$$CC = (E - N) + 2 \quad \dots (3.1)$$

Keterangan :

E= Edge

N=Node

2. *Black Box Testing* merupakan cara pengujian perangkat lunak yang memeriksa fungsionalitas dari aplikasi yang berkaitan dengan struktur internal kerja. Pengujian dilaksanakan dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Pengujian *black box* dilakukan oleh *user* yang mengoperasikan aplikasi.

3.1.5 Tahapan Pemeliharaan Sistem

Sistem yang sudah selesai akan mengalami perubahan. Perubahan biasanya berupa *error* sehingga diperlukan perbaikan dan pemeliharaan pada sistem.

BAB 4. PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan aplikasi penilaian tingkat kematangan tata kelola TI berdasarkan domain PO dan AI *framework* COBIT 4.1. Tahap pengembangan sistem dilaksanakan berdasarkan mode *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, dan pengujian sistem.

4.1 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem merupakan tahapan yang penting dalam pengembangan sebuah sistem informasi. Analisis kebutuhan yang dilakukan sesuai dengan data yang diperoleh pada tahapan yang telah dijelaskan pada bab metodologi penelitian subbab 3.1.1. Seluruh kebutuhan didefinisikan pada tahap kebutuhan fungsional dan kebutuhan non-fungsional.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses yang akan dilakukan oleh sistem. Kebutuhan fungsional dari aplikasi ini adalah sebagai berikut:

1. Sistem mampu mengelola data *user* meliputi (*view, insert, edit, delete*).
2. Sistem mampu mengelola data domain meliputi (*view, insert, edit, delete*).
3. Sistem mampu mengelola data kontrol objektif meliputi (*view, insert, edit, delete*).
4. Sistem mampu mengelola data *maturity level* meliputi (*view, insert, edit, delete*).
5. Sistem mampu mengelola data pernyataan meliputi (*view, insert, edit, delete*).
6. Sistem mampu mengelola data periode meliputi (*view, insert, edit, delete*).
7. Sistem mampu menampilkan data pernyataan berupa kuesioner.

8. Sistem mampu menampilkan hasil dari penilaian tingkat kematangan tata kelola TI berdasarkan *framework* COBIT 4.1.

4.1.2 Kebutuhan Non-Fungsional

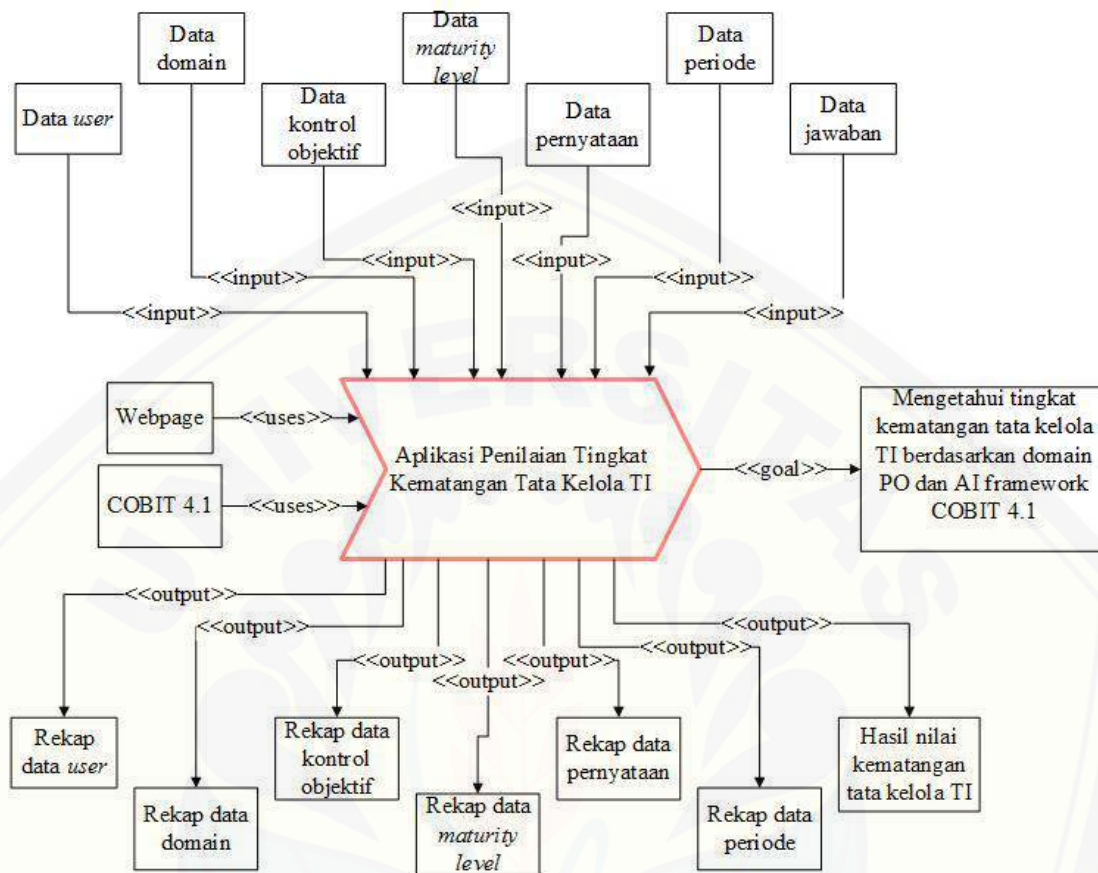
Kebutuhan non-fungsional merupakan hal yang dibutuhkan oleh sistem untuk mendukung aktivitas sistem sesuai dengan kebutuhan fungsional yang telah disusun. Kebutuhan non-fungsional menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan non-fungsional aplikasi ini yaitu sistem menggunakan *username* dan *password* untuk autentifikasi akses terhadap sistem.

4.2 Desain Sistem

Desain sistem yang dibuat berdasarkan gambaran sistem yang telah dijelaskan pada bab metodologi penelitian subbab 3.1.1. Desain yang dibuat meliputi *business process*, *use case diagram*, *use case scenario*, *sequence diagram*, *activity diagram*, *class diagram*, dan ERD.

4.1.1 Business Process

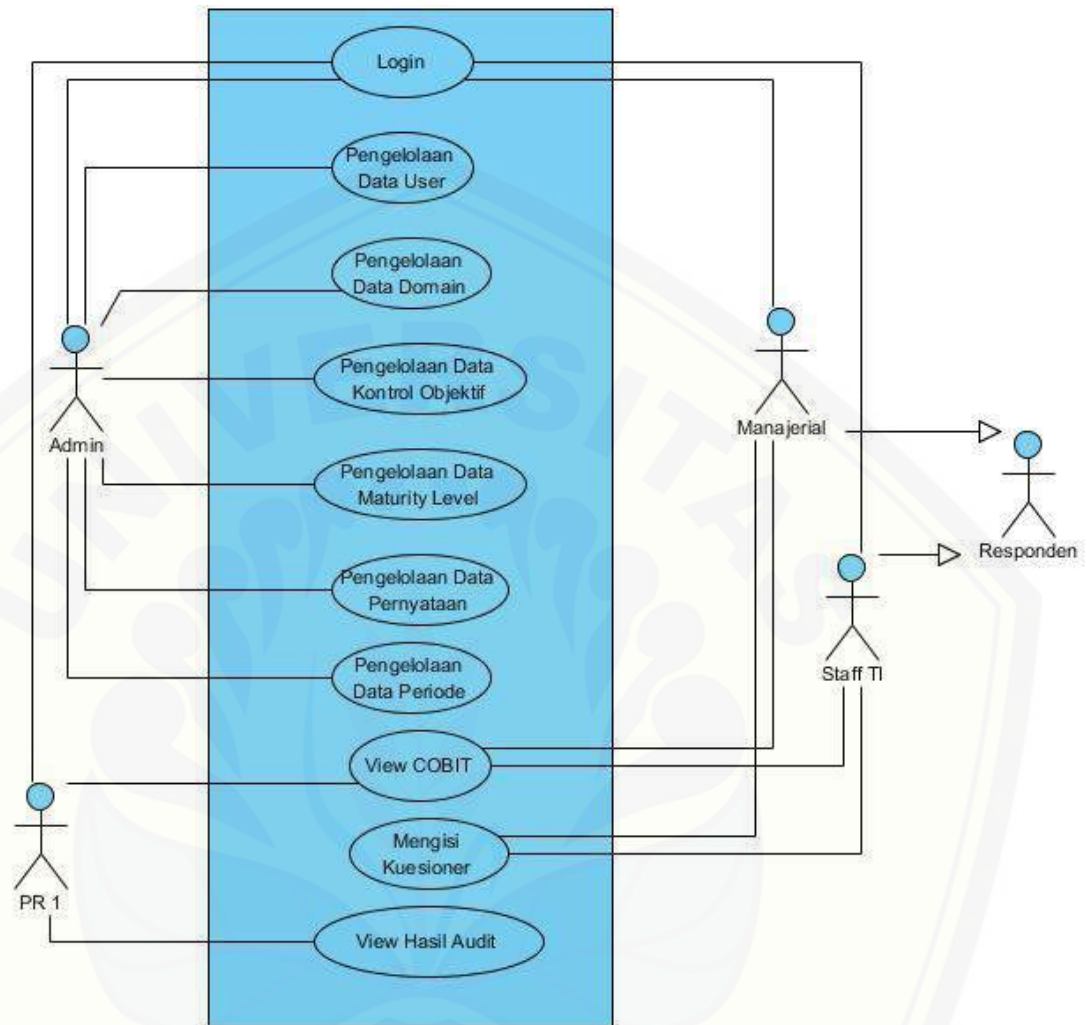
Business Process adalah suatu kumpulan aktivitas yang terstruktur untuk mencapai suatu tujuan tertentu atau untuk menghasilkan sebuah produk. Ada beberapa komponen di dalamnya, meliputi data yang menjadi masukan (*input*), data masukan yang kemudian diolah menjadi data keluaran (*output*), media yang digunakan (*uses*), dan tujuan yang ingin dicapai (*goal*). *Business Process* aplikasi ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 Business Process

4.1.2 Use Case Diagram

Use Case Diagram merupakan dokumentasi yang menggambarkan fitur dan aktor yang dapat mengakses fitur tersebut pada aplikasi penilaian tingkat kematangan tata kelola TI. *Use Case Diagram* aplikasi ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Use Case Diagram*

Penjelasan tentang definisi aktor dan definisi *use case* dalam *use case diagram* pada Gambar 4.2 akan dijelaskan di bawah ini.

1. Definisi Aktor

Definisi aktor merupakan penjelasan tentang aktor-aktor sebagai pengguna dari aplikasi penilaian tingkat kematangan tata kelola TI yang akan dibangun. Terdapat 4 aktor seperti yang dijelaskan pada Tabel 4.1.

Tabel 4.1 Definsi Aktor

No.	Aktor	Deskripsi
1.	Admin	Aktor admin memiliki hak akses secara penuh. Admin dapat melakukan <i>login</i> dan mengelola data sistem secara keseluruhan, meliputi data <i>user</i> , data domain, data kontrol objektif, data <i>maturity level</i> , data pernyataan, dan data periode.
2.	Staff TI	Aktor staff TI digeneralisasi sebagai responden yang memiliki hak akses untuk mengisi kuesioener dengan melihat data pernyataan. Staff TI juga memiliki hak akses melihat COBIT meliputi melihat data domain dan data kontrol objektif.
3.	Manajerial	Aktor manajerial digeneralisasi sebagai responden yang memiliki hak akses untuk mengisi kuesioener dengan melihat data pernyataan. Manajerial juga memiliki hak akses melihat COBIT meliputi melihat data domain dan data kontrol objektif.
4.	Pembantu Rektor 1 (PR 1)	Aktor PR 1 memiliki hak akses untuk melihat hasil perhitungan tingkat kematangan tata kelola TI berdasarkan domain PO dan AI <i>framework</i> COBIT 4.1 sesuai dengan periode yang ada.

2. Definisi *Use Case*

Definisi *use case* merupakan penjelasan dari masing-masing *use case* atau fitur-fitur dari aplikasi penilaian tingkat kematangan tata kelola TI yang akan dibangun. Terdapat 4 aktor seperti yang dijelaskan pada Tabel 4.2.

Tabel 4.2 Definisi Use Case

No.	Use Case	Deskripsi
1.	<i>Login</i>	Menggambarkan proses autentifikasi untuk masuk ke sistem.
2.	Pengelolaan Data <i>User</i>	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data <i>user</i> .
3.	Pengelolaan Data Domain	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data domain.
4.	Pengelolaan Data Kontrol Objektif	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data kontrol objektif.
5.	Pengelolaan Data <i>Maturity Level</i>	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data <i>maturity level</i> .
6.	Pengelolaan Data Pernyataan	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data pernyataan.
7.	Pengelolaan Data Periode	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data periode.
8.	<i>View COBIT</i>	Menggambarkan proses melihat data COBIT meliputi data domain dan data kontrol objektif.
9.	Mengisi Kuesioner	Menggambarkan proses pengisian kuesioner.
10.	<i>View Hasil Audit</i>	Menggambarkan proses melihat hasil audit yang merupakan hasil dari kuesioner yang sudah diolah dan menghasilkan hasil perhitungan penilaian tata kelola TI sesuai dengan kuesioner yang telah diisi.

4.1.3 *Use Case Scenario*

Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* seperti pada Gambar 4.2.

1. Skenario *Use Case Login*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case login* dijelaskan pada lampiran A.

2. Skenario *Use Case Pengelolaan Data User*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data user* dapat dilihat pada lampiran B.

3. Skenario *Use Case Pengelolaan Data Domain*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data domain* dapat dilihat pada lampiran B.

4. Skenario *Use Case Pengelolaan Data Kontrol Objektif*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Pengelolaan Data kontrol objektif* dapat dilihat pada lampiran B.

5. Skenario *Use Case Pengelolaan Data Maturity Level*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data maturity level* dapat dilihat pada lampiran B.

6. Skenario *Use Case Pengelolaan Data Pernyataan*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data pernyataan* dapat dilihat pada lampiran B.

7. Skenario *Use Case Pengelolaan Data Periode*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data periode* dapat dilihat pada lampiran B.

8. Skenario *Use Case View COBIT*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* Pengelolaan Data COBIT dapat dilihat pada lampiran B.

9. Skenario *Use Case* Mengisi Kuesioner

Use case mengisi kuesioner menghubungkan dengan 2 aktor, yaitu staff TI dan manajerial. Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* mengisi kuesioner dapat dilihat pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Skenario *Use Case* Mengisi Kuesioner Staff TI

Nama Use Case	Mengisi Kuesioner
Aktor	Staff TI
Deskripsi Singkat	Staff TI akan menginputkan penilaian
Prekondisi	Data pernyataan yang telah diinputkan
Prakondisi	Mengisi kuesioner berhasil
Flow Events	
Skenario Normal : View Domain	
Aksi Aktor	Reaksi Sistem
1. Klik menu kuesioner	
	2. Menampilkan halaman kuesioner meliputi daftar domain dan kontrol objektif serta status penginputan
Skenario Normal : View Kontrol Objektif	
Aksi Aktor	Reaksi Sistem

3. Klik tombol kontrol objektif	
	4. Menampilkan halaman pernyataan beserta tombol jawaban ya dan tidak
Skenario Normal : Memilih Tombol Ya	
Aksi Aktor	Reaksi Sistem
5. Klik tombol ya	
	6. Menyimpan jawaban ya ke database
	7. Menampilkan halaman pernyataan
Skenario Alternatif : Memilih Tombol Tidak	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menyimpan jawaban tidak ke database
	7a. Menampilkan halaman pernyataan
Skenario Normal : Memilih Tombol Selesai	
Aksi Aktor	Reaksi Sistem
8. Klik tombol selesai	
	9. Menampilkan halaman kuesioner

Tabel 4.4 Skenario *Use Case* Mengisi Kuesioner Manajerial

Nama Use Case	Mengisi Kuesioner
Aktor	Manajerial
Deskripsi Singkat	Manajerial akan menginputkan penilaian
Prekondisi	Data pernyataan yang telah diinputkan
Prakondisi	Mengisi kuesioner berhasil
Flow Events	
Skenario Normal : View Domain	
Aksi Aktor	Reaksi Sistem
1. Klik menu kuesioner	
	2. Menampilkan halaman kuesioner meliputi daftar domain dan kontrol objektif serta status penginputan
Skenario Normal : View Kontrol Objektif	
Aksi Aktor	Reaksi Sistem
3. Klik tombol kontrol objektif	
	4. Menampilkan halaman pernyataan beserta tombol jawaban ya dan tidak
Skenario Normal : Memilih Tombol Ya	

Aksi Aktor	Reaksi Sistem
5. Klik tombol ya	
	6. Menyimpan jawaban ya ke database
	7. Menampilkan halaman pernyataan
Skenario Normal : Memilih Tombol Tidak	
Aksi Aktor	Reaksi Sistem
5. Klik tombol tidak	
	6. Menyimpan jawaban tidak ke database
	7. Menampilkan halaman pernyataan
Skenario Normal : Memilih Tombol Selesai	
Aksi Aktor	Reaksi Sistem
8. Klik tombol selesai	
	9. Menampilkan halaman kuesioner

10. Skenario *Use Case View* Hasil Perhitungan

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case view* hasil perhitungan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Skenario Use Case View Hasil Perhitungan

Nama Use Case	<i>View</i> Hasil Perhitungan
Aktor	PR 1

Deskripsi Singkat	PR 1 akan melihat hasil perhitungan <i>maturity level</i>
Prekondisi	Data jawaban yang telah diinputkan
Prakondisi	Data <i>maturity level</i>
Flow Events	
Skenario Normal : View Domain	
Aksi Aktor	Reaksi Sistem
1. Klik menu hasil audit	
	2. Menampilkan halaman hasil audit meliputi tabel periode yang telah diinputkan dan tombol hitug step 1, hitung step 2, hitung step 3, hitung step 4, hitung step 5, dan maturity level
Skenario Normal : Perhitungan Tahap 1	
Aksi Aktor	Reaksi Sistem
3. Klik tombol hitung step 1	
	4. Melakukan perhitungan tahap 1 dan menyimpan hasilnya ke database
	5. Menampilkan kembali halaman hasil audit

Skenario Normal : Perhitungan Tahap 2	
Aksi Aktor	Reaksi Sistem
6. Klik tombol hitung step 2	
	7. Melakukan perhitungan tahap 2 dan menyimpan hasilnya ke database
	8. Menampilkan kembali halaman hasil audit
Skenario Normal : Perhitungan Tahap 3	
Aksi Aktor	Reaksi Sistem
9. Klik tombol hitung step 3	
	10. Melakukan perhitungan tahap 3 dan menyimpan hasilnya ke database
	11. Menampilkan kembali halaman hasil audit
Skenario Normal : Perhitungan Tahap 4	
Aksi Aktor	Reaksi Sistem
12. Klik tombol hitung step 4	
	13. Melakukan perhitungan tahap 4 dan menyimpan hasilnya ke database

	14. Menampilkan kembali halaman hasil audit
Skenario Normal : Perhitungan Tahap 5	
Aksi Aktor	Reaksi Sistem
15. Klik tombol hitung step 5	
	16. Melakukan perhitungan tahap 5 dan menyimpan hasilnya ke database
	17. Menampilkan kembali halaman hasil audit
Skenario Normal : View Maturity Level Kontrol Objektif	
Aksi Aktor	Reaksi Sistem
18. Klik tombol kontrol objektif	
	19. Menampilkan halaman maturity level kontrol objektif meliputi tabel maturity level setiap responden pada kontrol objektif dan tombol hitung maturity
Skenario Normal : View Hasil Maturity Level	
Aksi Aktor	Reaksi Sistem
20. Klik tombol hitung maturity	
	21. Menampilkan halaman hasil perhitungan tingkat

	kematangan tata kelola teknologi informasi dan grafik nilai <i>maturity level</i> setiap kontrol objektif
Skenario Normal : Memilih Tombol Selesai	
Aksi Aktor	Reaksi Sistem
22. Klik tombol kembali	
	23. Menampilkan kembali halaman daftar periode

4.1.4 *Sequence Diagram*

Sequence diagram adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antarobjek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu pada aplikasi penilaian tingkat kematangan tata kelola TI.

1. *Sequence Diagram Login*

Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

2. *Sequence Diagram Pengelolaan Data User*

Penggambaran *sequence diagram pengelolaan data user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

3. *Sequence Diagram Pengelolaan Data Domain*

Penggambaran *sequence diagram pengelolaan data domain* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

4. *Sequence Diagram* Pengelolaan Data Kontrol Objektif

Penggambaran *sequence diagram* pengelolaan data kontrol objektif digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

5. *Sequence Diagram* Pengelolaan Data *Maturity Level*

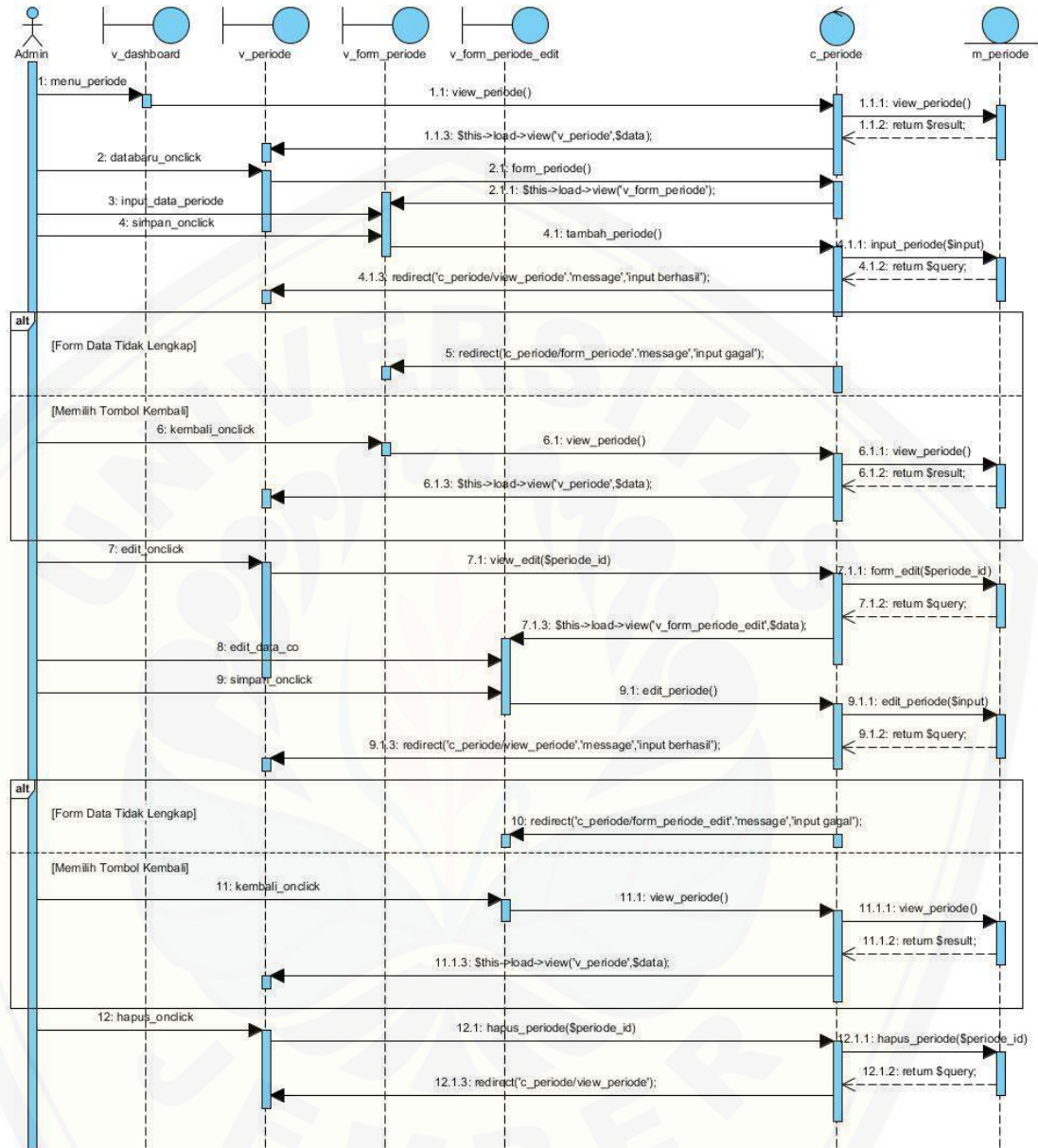
Penggambaran *sequence diagram* pengelolaan data *maturity level* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

6. *Sequence Diagram* Pengelolaan Data Pernyataan

Penggambaran *sequence diagram* pengelolaan data pernyataan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

7. *Sequence Diagram* Pengelolaan Data Periode

Penggambaran *sequence diagram* pengelolaan data periode digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3.



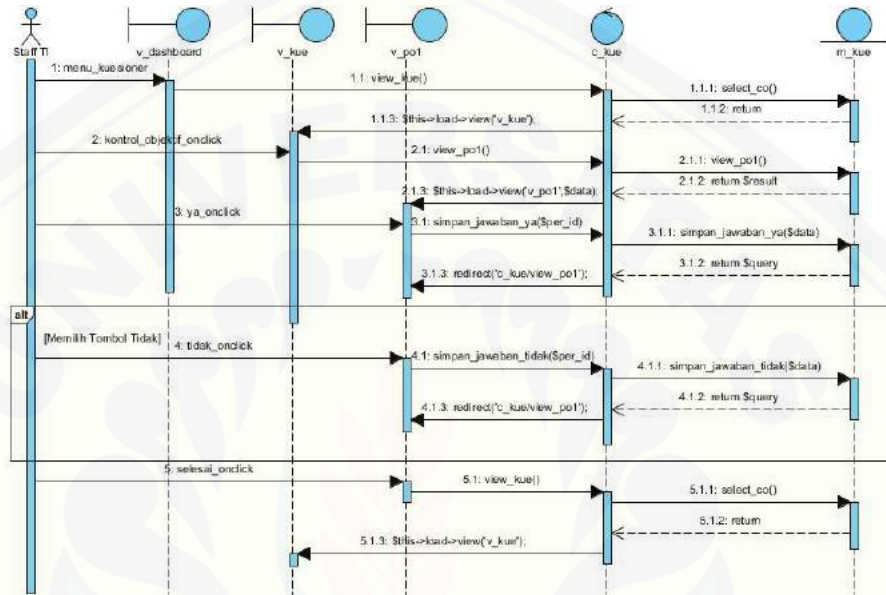
Gambar 4.3 Sequence Diagram Pengelolaan Data Periode

8. Sequence Diagram View COBIT

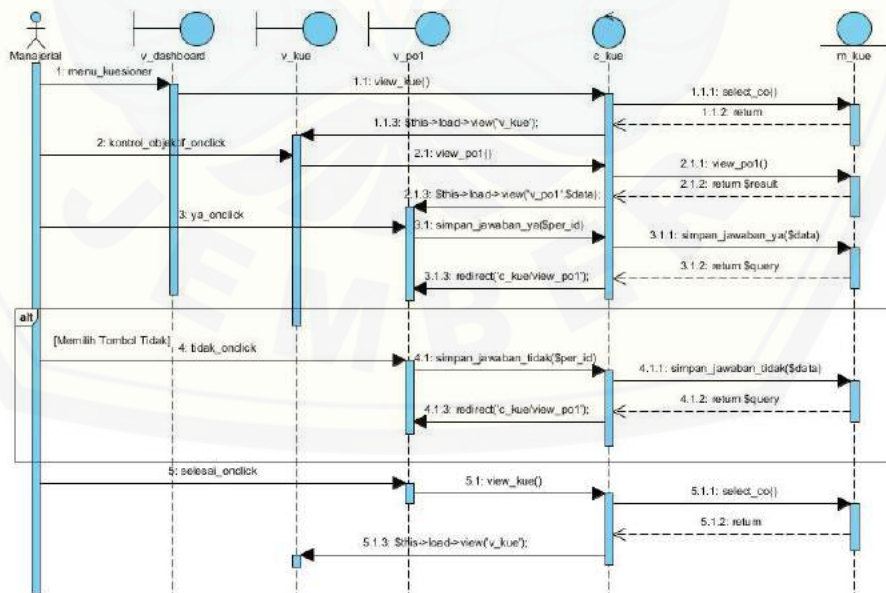
Penggambaran *sequence diagram view* COBIT digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

9. Sequence Diagram Mengisi Kuesioner

Penggambaran *sequence diagram* mengisi kuesioner digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.4 dan Gambar 4.5.



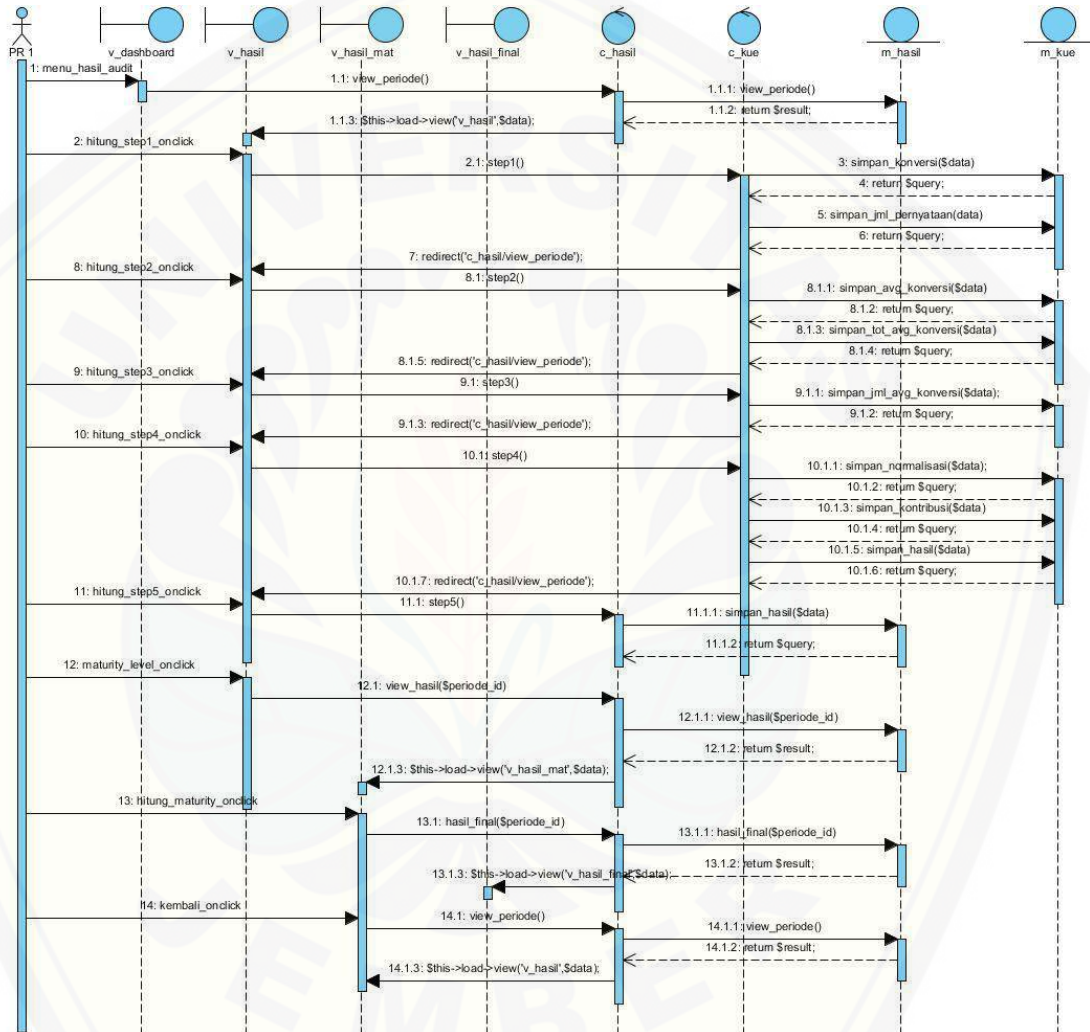
Gambar 4.4 Sequence Diagram Mengisi Kuesioner Staff TI



Gambar 4.5 Sequence Diagram Mengisi Kuesioner Manajerial

10. Sequence Diagram View Hasil Perhitungan

Penggambaran *sequence diagram view* hasil perhitungan digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 Sequence Diagram View Hasil Perhitungan

4.1.5 *Activity Diagram*

Activity diagram menggambarkan alur aktivitas pada aplikasi penilaian tingkat kematangan tata kelola TI.

1. *Activity Diagram Login*

Activity diagram login dapat dilihat pada lampiran D.

2. *Activity Diagram Pengelolaan Data User*

Activity diagram pengelolaan data *user* dapat dilihat pada lampiran D.

3. *Activity Diagram Pengelolaan Data Domain*

Activity diagram pengelolaan data domain dapat dilihat pada lampiran D.

4. *Activity Diagram Pengelolaan Data Kontrol Objektif*

Activity diagram pengelolaan data kontrol objektif dapat dilihat pada lampiran D.

5. *Activity Diagram Pengelolaan Data Maturity Level*

Activity diagram pengelolaan data *maturity* dapat dilihat pada lampiran D.

6. *Activity Diagram Pengelolaan Data Pernyataan*

Activity diagram pengelolaan data pernyataan dapat dilihat pada lampiran D.

7. *Activity Diagram Pengelolaan Data Periode*

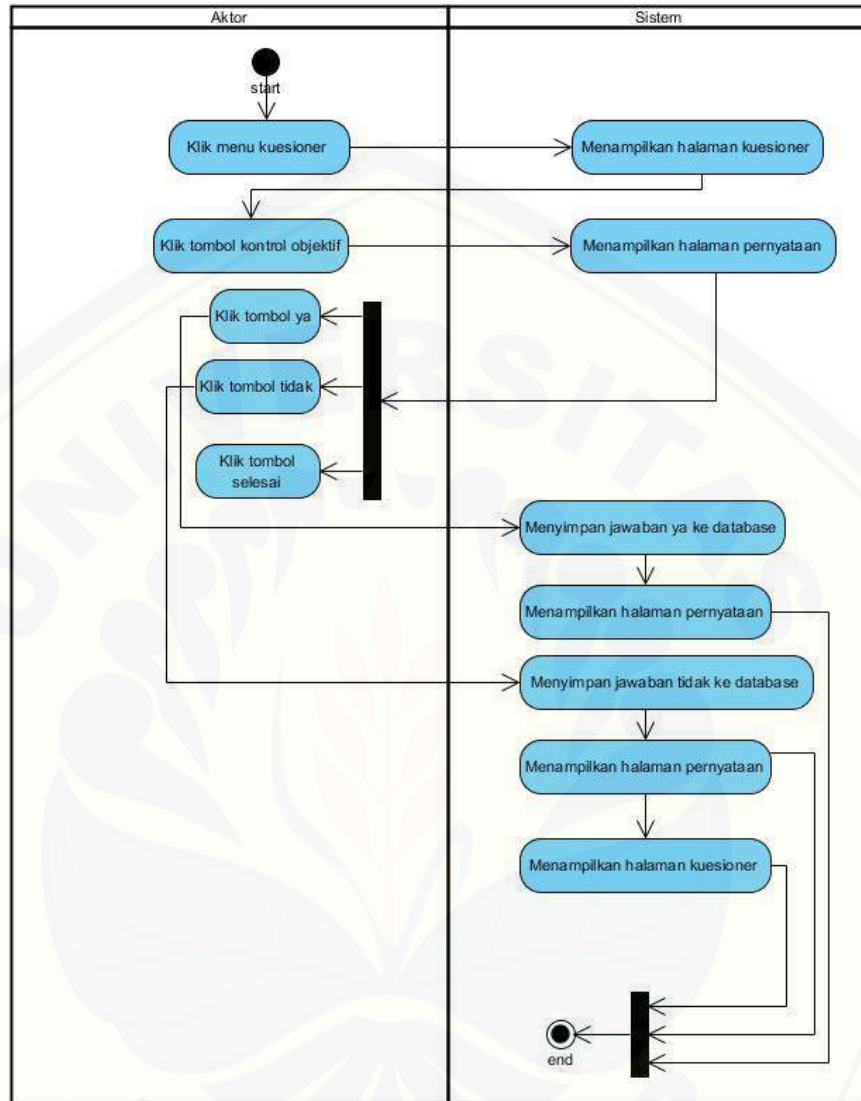
Activity diagram pengelolaan data periode dapat dilihat pada lampiran D.

8. *Activity Diagram View COBIT*

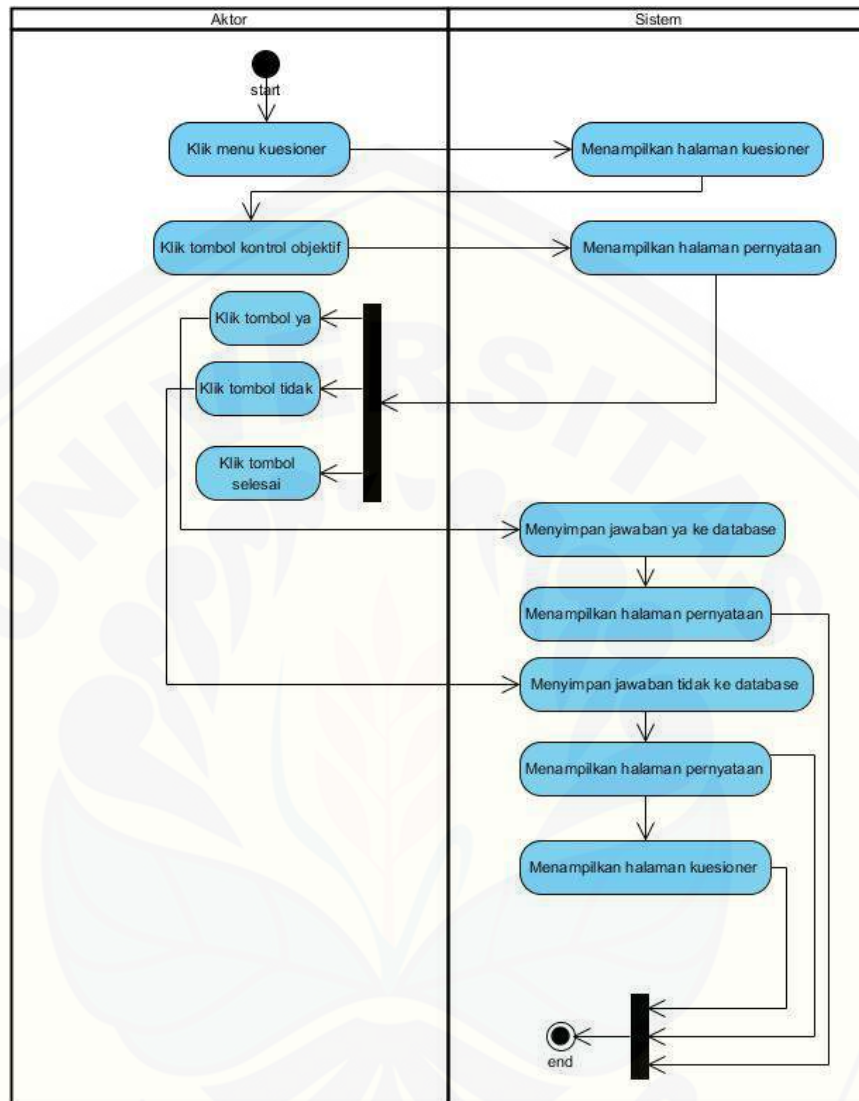
Activity diagram view COBIT dapat dilihat pada lampiran D.

9. *Activity Diagram Mengisi Kuesioner*

Activity diagram mengisi kuesioner dapat dilihat pada Gambar 4.7 dan Gambar 4.8.



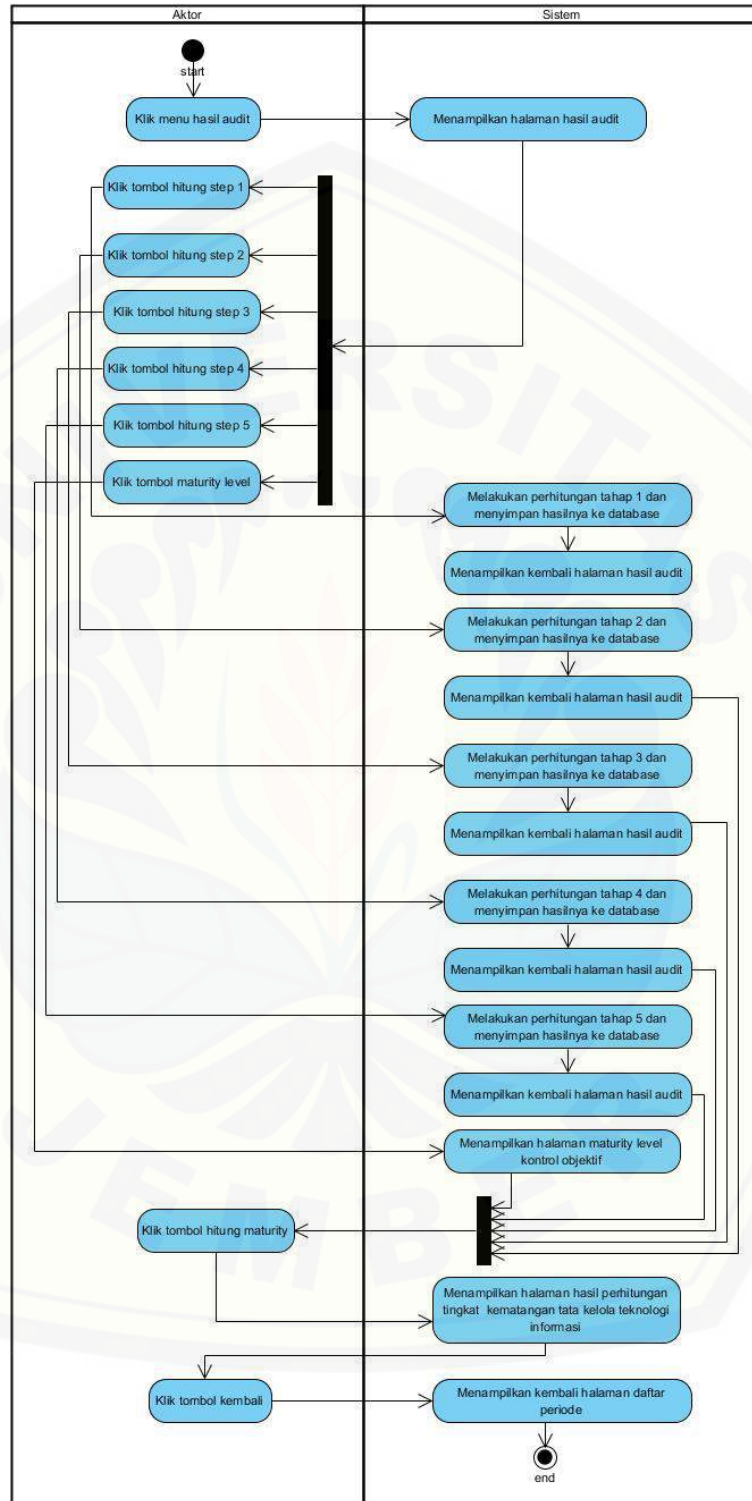
Gambar 4.7 Activity Diagram Mengisi Kuesioner Staff TI



Gambar 4.8 Activity Diagram Mengisi Kuesioner Manajerial

10. Activity Diagram View Hasil Perhitungan

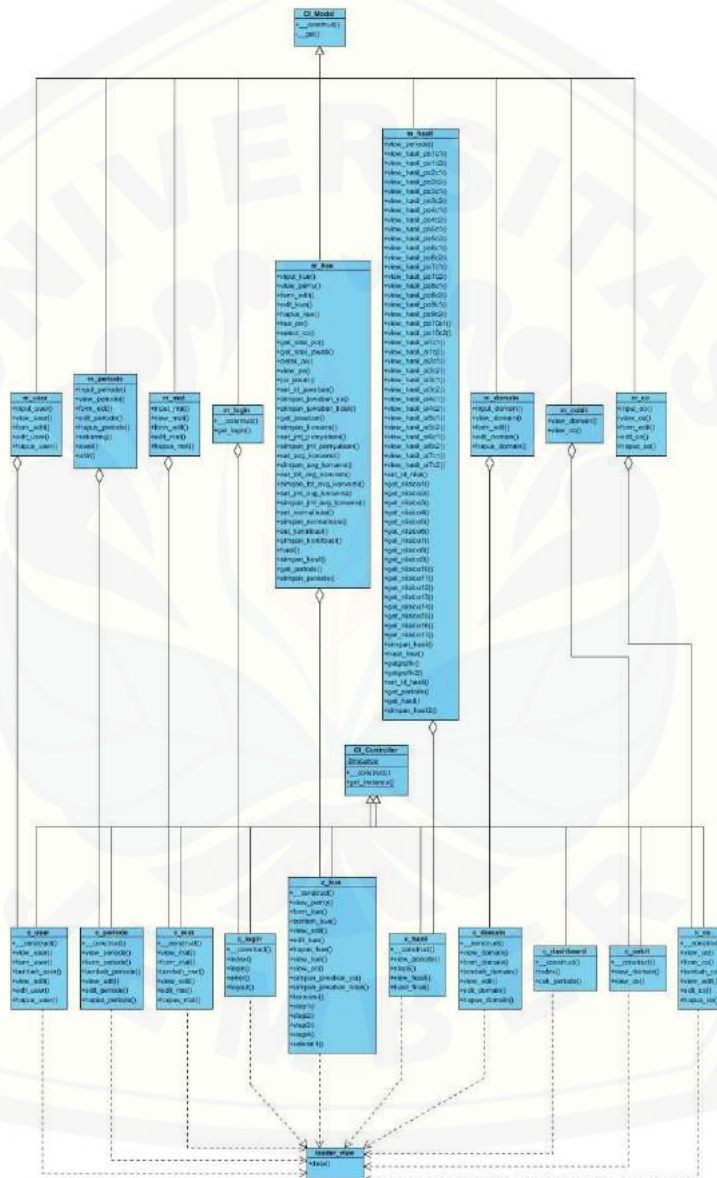
Activity diagram pengelolaan data view hasil perhitungan dapat dilihat pada Gambar 4.9.



Gambar 4.9 Activity Diagram View Hasil Perhitungan

4.1.6 Class Diagram

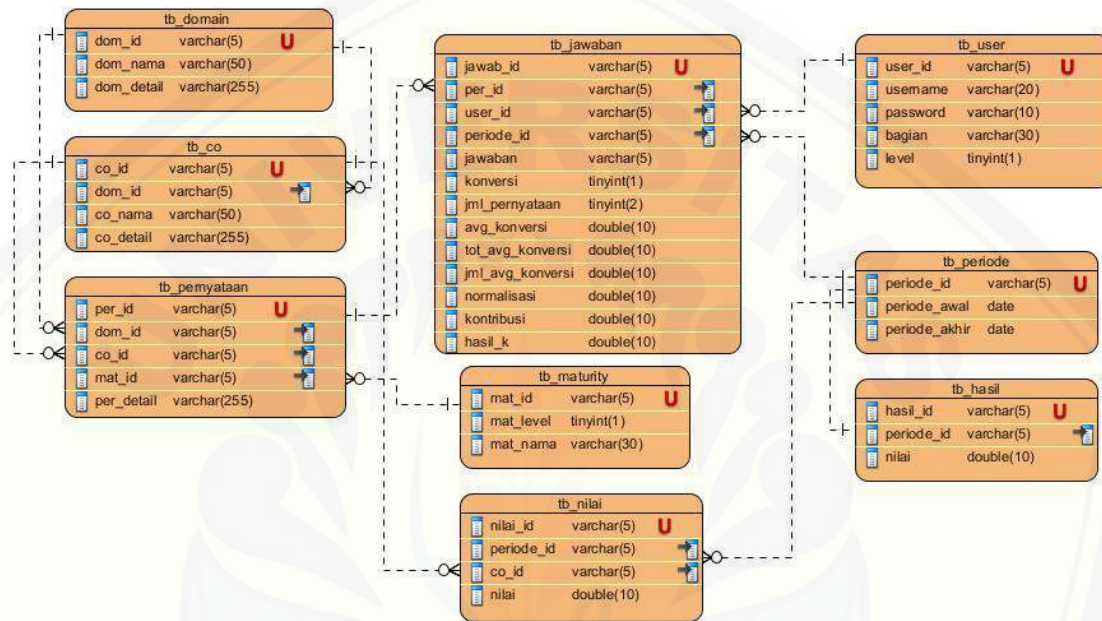
Class Diagram menggambarkan hubungan antarkelas yang digunakan untuk membangun suatu sistem. Class Diagram aplikasi penilaian tingkat kematangan tata kelola TI dapat dilihat pada Gambar 4.10.



Gambar 4.10 Class Diagram

4.1.7 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD aplikasi penilaian tingkat kematangan tata kelola TI dapat dilihat pada Gambar 4.11.



Gambar 4.11 Entity Relationship Diagram

4.3 Penulisan Kode Program

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Penulisan kode program untuk setiap fitur seperti yang telah digambarkan pada *use case diagram* Gambar 4.2 ditulis dalam 3 bagian kelas, meliputi kelas *view*, *controller*, dan *model*.

4.3.1 Kode Program Login

Kode program *login* terletak pada kelas *v_login*, *c_login*, dan *m_login*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.2 Kode Program Pengelolaan Data *User*

Kode program *login* terletak pada kelas *v_user_adm*, *v_form_user*, *v_form_user_edit*, *c_user*, dan *m_user*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.3 Kode Program Pengelolaan Data Domain

Kode program *login* terletak pada kelas *v_domain_adm*, *v_form_domain*, *v_form_domain_edit*, *c_domain*, dan *m_domain*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.4 Kode Program Pengelolaan Data Kontrol Objektif

Kode program kontrol objektif terletak pada kelas *v_co_adm*, *v_form_co*, *v_form_co_edit*, *c_co*, dan *m_co*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.5 Kode Program Pengelolaan Data *Maturity Level*

Kode program *maturity level* terletak pada kelas *v_mat*, *v_form_mat*, *v_form_mat_edit*, *c_mat*, dan *m_mat*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.6 Kode Program Pengelolaan Data Pernyataan

Kode program pernyataan terletak pada kelas *v_kue_adm*, *v_form_kue*, *v_form_kue_edit*, *c_kue*, dan *m_kue*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.7 Kode Program Pengelolaan Data Periode

Kode program periode terletak pada kelas `v_periode`, `v_form_periode`, `v_form_periode_edit`, `c_periode`, dan `m_periode`. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.8 Kode Program View COBIT

Kode program *view* COBIT terletak pada kelas `v_cobit`, `v_cobit_co`, `c_cobit`, dan `m_cobit`. Penulisan kode program ini dapat dilihat pada lampiran E.

4.3.9 Kode Program Mengisi Kuesioner

Kode program mengisi kuesioner terletak pada kelas `v_kue`, `v_po1`, `c_kue`, dan `m_kue`. Implementasi COBIT 4.1 terletak pada kelas `m_kue`. Penulisan kode program dapat dilihat pada Gambar 4.12 sampai dengan Gambar 4.34.

```
public function simpan_jawaban_ya($data){  
    return $query = $this->db->insert('tb_jawaban', $data);  
}
```

Gambar 4.12 Kode Program function jawaban_ya

Gambar 4.12 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk menyimpan jawaban ya yang dipilih oleh responden.

```
public function simpan_jawaban_tidak($data){  
    return $query = $this->db->insert('tb_jawaban', $data);  
}
```

Gambar 4.13 Kode Program function simpan_jawaban_tidak

Gambar 4.13 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk menyimpan jawaban tidak yang dipilih oleh responden.

```
public function get_jawaban(){
    $query = $this->db->query("SELECT jawab_id, jawaban FROM tb_jawaban");
    $konversi=array();
    $indeks=0;
    if($query->num_rows() > 0){
        foreach($query->result_array() as $jawab){
            if($jawab['jawaban'] == 'ya'){
                $konversi[$indeks]['jawaban'] = 1;
            }else if($jawab['jawaban'] == 'tidak'){
                $konversi[$indeks]['jawaban'] = 0;
            }
            $konversi[$indeks]['jawab_id'] = $jawab['jawab_id'];
            $indeks++;
        }
    }
    return $konversi;
}
```

Gambar 4.14 Kode Program function get_jawaban

Gambar 4.14 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk menyimpan jawaban yang telah dipilih oleh responden. Jika jawaban ya dikonversikan menjadi 1 dan jika jawaban tidak dikonversikan menjadi 0.

```
public function simpan_konversi($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET konversi='".$row['jawaban']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}
```

Gambar 4.15 Kode Program function simpan_konversi

Gambar 4.15 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk menyimpan jawaban yang telah dikonversikan pada *function* Gambar 4.14.


```

public function set_jml_pernyataan(){
    $query = $this->db->query("SELECT j.jawab_id, j.per_id, p.dom_id, p.co_id, p.mat_id FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id");
    $jml = array();
    $indeks = 0;
    if($query->num_rows() > 0){
        foreach($query->result_array() as $hasil){
            if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M1'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P01' AND mat_id='M1'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M2'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P01' AND mat_id='M2'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M3'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P01' AND mat_id='M3'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M4'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P01' AND mat_id='M4'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M5'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P01' AND mat_id='M5'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M6'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P01' AND mat_id='M6'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }

            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P02' && $hasil['mat_id'] == 'M1'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P02' AND mat_id='M1'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P02' && $hasil['mat_id'] == 'M2'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P02' AND mat_id='M2'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P02' && $hasil['mat_id'] == 'M3'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P02' AND mat_id='M3'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P02' && $hasil['mat_id'] == 'M4'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P02' AND mat_id='M4'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P02' && $hasil['mat_id'] == 'M5'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P02' AND mat_id='M5'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P02' && $hasil['mat_id'] == 'M6'){
                $sq = $this->db->query("SELECT COUNT(per_id) as jml FROM tb_pernyataan WHERE dom_id='PO' AND co_id='P02' AND mat_id='M6'");
                $jml[$indeks]['per_id'] = $sq[0]['jml'];
            }
        }
        $jml[$indeks]['jawab_id'] = $hasil['jawab_id'];
        $indeks++;
    }
    return $jml;
}

```

Gambar 4.16 Kode Program Function set_jml_pernyataan

Gambar 4.16 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk mengambil nilai jumlah pernyataan pada setiap level di kontrol objektifnya domain.

```

public function simpan_jml_pernyataan($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET jml_pernyataan='".$row['per_id']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}

```

Gambar 4.17 Kode Program Function simpan_jml_pernyataan

Gambar 4.17 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk menyimpan nilai jumlah pernyataan sesuai dengan jawaban yang dipilih oleh responden pada setiap level di kontrol objektifnya domain yang akan digunakan untuk

perhitungan.

```
public function set_avg_konversi(){
    $query = $this->db->query("SELECT jawab_id, (konversi/jml_pernyataan) as avg FROM tb_jawaban")->result_array();
    return $query;
}
```

Gambar 4.18 Kode Program Function set_avg_konversi

Gambar 4.18 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk membuat nilai rata-rata konversi dari jawaban yang dipilih responden.

```
public function simpan_avg_konversi($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET avg_konversi='".$row['avg']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}
```

Gambar 4.19 Kode Program Function simpan_avg_konversi

Gambar 4.19 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk menyimpan nilai rata-rata konversi yang telah dibuat pada *function* Gambar 4.18.

```
public function set_tot_avg_konversi(){
    $query = $this->db->query("SELECT j.jawab_id, j.per_id, j.avg_konversi, j.user_id, p.dom_id, p.co_id FROM tb_jawaban j join tb_pernyataan p
    on j.per_id=p.per_id");
    $total = array();
    $indeks = 0;
    if($query->num_rows() > 0){
        foreach($query->result_array() as $hasil){
            if($hasil['dom_id'] == 'P0' && $hasil['co_id'] == 'P01' && $hasil['user_id'] == 'C1'){
                $sq = $this->db->query("SELECT SUM(j.avg_konversi) as total FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='P0' AND p.co_id='P01' AND j.user_id='C1'")->result_array();
                $total[$indeks]['per_id'] = $sq[0]['total'];
            }
            else if($hasil['dom_id'] == 'P0' && $hasil['co_id'] == 'P01' && $hasil['user_id'] == 'C2'){
                $sq = $this->db->query("SELECT SUM(j.avg_konversi) as total FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='P0' AND p.co_id='P01' AND j.user_id='C2'")->result_array();
                $total[$indeks]['per_id'] = $sq[0]['total'];
            }
            else if($hasil['dom_id'] == 'P0' && $hasil['co_id'] == 'P02' && $hasil['user_id'] == 'C1'){
                $sq = $this->db->query("SELECT SUM(j.avg_konversi) as total FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='P0' AND p.co_id='P02' AND j.user_id='C1'")->result_array();
                $total[$indeks]['per_id'] = $sq[0]['total'];
            }
            else if($hasil['dom_id'] == 'P0' && $hasil['co_id'] == 'P02' && $hasil['user_id'] == 'C2'){
                $sq = $this->db->query("SELECT SUM(j.avg_konversi) as total FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='P0' AND p.co_id='P02' AND j.user_id='C2'")->result_array();
                $total[$indeks]['per_id'] = $sq[0]['total'];
            }
        }
        $total[$indeks]['jawab_id'] = $hasil['jawab_id'];
        $indeks++;
    }
    return $total;
}
```

Gambar 4.20 Kode Program Function set_tot_avg_konversi

Gambar 4.20 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk membuat nilai total dari jumlah rata-rata konversi pada setiap kontrol objektif.

```

public function simpan_tot_avg_konversi($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET tot_avg_konversi='".$row['per_id']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}

```

Gambar 4.21 Kode Program Function simpan_tot_avg_konversi

Gambar 4.21 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk menyimpan nilai total total jumlah rata-rata konversi yang telah dibuat pada *function* Gambar 4.20.

```

public function set_jml_avg_konversi(){
    $query = $this->db->query("SELECT j.jawab_id, j.per_id, j.user_id, p.dom_id, p.co_id, p.mat_id FROM tb_jawaban j join tb_pernyataan p
    on j.per_id=p.per_id");
    $jml = array();
    $indeks = 0;
    if($query->num_rows() > 0){
        foreach($query->result_array() as $hasil){
            if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M1' && $hasil['user_id'] == 'C1'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M1' AND j.user_id='C1'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M2' && $hasil['user_id'] == 'C1'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M2' AND j.user_id='C1'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M3' && $hasil['user_id'] == 'C1'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M3' AND j.user_id='C1'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M4' && $hasil['user_id'] == 'C1'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M4' AND j.user_id='C1'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M5' && $hasil['user_id'] == 'C1'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M5' AND j.user_id='C1'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M6' && $hasil['user_id'] == 'C1'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M6' AND j.user_id='C1'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M1' && $hasil['user_id'] == 'C2'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M1' AND j.user_id='C2'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M2' && $hasil['user_id'] == 'C2'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M2' AND j.user_id='C2'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M3' && $hasil['user_id'] == 'C2'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M3' AND j.user_id='C2'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
            else if($hasil['dom_id'] == 'PO' && $hasil['co_id'] == 'P01' && $hasil['mat_id'] == 'M4' && $hasil['user_id'] == 'C2'){
                $q = $this->db->query("SELECT SUM(j.avg_konversi) as jml_avg FROM tb_jawaban j join tb_pernyataan p on j.per_id=p.per_id
                WHERE p.dom_id='PO' AND p.co_id='P01' AND p.mat_id='M4' AND j.user_id='C2'");
                $jml[$indeks]['per_id'] = $q[0]['jml_avg'];
            }
        }
        $jml[$indeks]['jawab_id'] = $hasil['jawab_id'];
        $indeks++;
    }
    return $jml;
}

```

Gambar 4.22 Kode Program Function set_jml_avg_konversi

Gambar 4.22 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk membuat nilai jumlah rata-rata konversi pada setiap level di kontrol objektif.

```
public function simpan_jml_avg_konversi($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET jml_avg_konversi='".$row['per_id']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}
```

Gambar 4.23 Kode Program Function `simpan_jml_avg_konversi`

Gambar 4.23 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk menyimpan nilai jumlah rata-rata konversi yang telah dibuat pada *function* Gambar 4.22.

```
public function set_normalisasi(){
    $query = $this->db->query("SELECT jawab_id, (jml_avg_konversi/tot_avg_konversi) as normalisasi FROM tb_jawaban")->result_array();
    return $query;
}
```

Gambar 4.24 Kode Program Function `set_normalisasi`

Gambar 4.24 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk membuat nilai normalisasi.

```
public function simpan_normalisasi($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET normalisasi='".$row['normalisasi']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}
```

Gambar 4.25 Kode Program Function `simpan_normalisasi`

Gambar 4.25 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk menyimpan nilai normalisasi yang telah dibuat pada *function* Gambar 4.24.

```
public function set_kontribusi(){
    $query = $this->db->query("SELECT j.jawab_id, (j.normalisasi*m.mat_level) as kontribusi FROM tb_jawaban j join tb_pernyataan p
    on j.per_id=p.per_id join tb_maturity m on p.mat_id=m.mat_id")->result_array();
    return $query;
}
```

Gambar 4.26 Kode Program Function `set_kontribusi`

Gambar 4.26 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk membuat nilai kontribusi.

```
public function simpan_kontribusi($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET kontribusi='".$row['kontribusi']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}
```

Gambar 4.27 Kode Program Function simpan_kontribusi

Gambar 4.27 merupakan *function* yang terletak di kelas *m_kue* yang digunakan untuk menyimpan nilai kontribusi yang telah dibuat pada *function* Gambar 4.26.

```
public function hasil() {
    $query = $this->db->query("SELECT jawab_id, per_id, user_id, kontribusi FROM tb_jawaban");
    $k = array();
    $indeks = 0;
    if($query->num_rows() > 0){
        foreach($query->result_array() as $hasil){
            if($hasil['per_id'] == 'P01' && $hasil['user_id']=='C1'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P01'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P03' && $hasil['user_id']=='C1'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P03'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P08' && $hasil['user_id']=='C1'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P08'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P12' && $hasil['user_id']=='C1'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P12'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P18' && $hasil['user_id']=='C1'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P18'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P24' && $hasil['user_id']=='C1'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P24'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P01' && $hasil['user_id']=='C2'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P01'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P03' && $hasil['user_id']=='C2'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P03'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P08' && $hasil['user_id']=='C2'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P08'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P12' && $hasil['user_id']=='C2'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P12'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P18' && $hasil['user_id']=='C2'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P18'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }else if($hasil['per_id'] == 'P24' && $hasil['user_id']=='C2'){
                $q = $this->db->query("SELECT kontribusi as k FROM tb_jawaban WHERE per_id='P24'")->result_array();
                $k[$indeks]['per_id'] = $q[0]['k'];
            }
        }
    }else{
        $k[$indeks]['per_id'] = 0;
    }
    $k[$indeks]['jawab_id'] = $hasil['jawab_id'];
    $indeks++;
}
return $k;
}
```

Gambar 4.28 Kode Program Function hasil

Gambar 4.28 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk mengambil nilai kontribusi pada setiap level kontrol objektif.

```
public function simpan_hasil($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET hasil_k='".$row['per_id']."' WHERE jawab_id='".$row['jawab_id']."'");
    }
    return $query;
}
```

Gambar 4.29 Kode Program Function `simpan_hasil`

Gambar 4.29 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk menyimpan nilai kontribusi yang telah dibuat pada *function* Gambar 4.28.

```
public function get_periode(){
    $query = $this->db->query("SELECT periode_id as periode FROM tb_periode order by periode_id desc limit 1")->result_array();
    return $query;
}
```

Gambar 4.30 Kode Program Function `get_periode`

Gambar 4.30 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk mengambil nilai periode.

```
public function simpan_periode($data){
    foreach($data as $row){
        $query = $this->db->query("UPDATE tb_jawaban SET periode_id='".$row['periode']."' ");
    }
    return $query;
}
```

Gambar 4.31 Kode Program Function `simpan_periode`

Gambar 4.31 merupakan *function* yang terletak di kelas `m_kue` yang digunakan untuk menyimpan nilai periode yang telah diambil pada *function* Gambar 4.30.

```
public function simpan_jawaban_ya($per_id){
    $id_jawab = $this->m_kue->set_id_jawaban();
    $user = $this->session->userdata('user_id');

    $data = array(
        'jawab_id' => $id_jawab,
        'per_id' => $per_id,
        'user_id' => $user,
        'periode_id' => $periode,
        'jawaban' => 'ya'
    );

    $logic = $this->m_kue->simpan_jawaban_ya($data);

    if ($logic==true) {
        redirect('c_kue/view_poi');
    }
}
```

Gambar 4.32 Kode Program Function `simpan_jawaban_ya`

Gambar 4.32 merupakan *function* yang terletak di kelas `c_kue` yang digunakan untuk menyimpan jawaban ya saat responden memilih jawaban ya.

```
public function simpan_jawaban_tidak($per_id){
    $id_jawab = $this->m_kue->set_id_jawaban();
    $user = $this->session->userdata('user_id');
    $periode = $this->m_kue->get_periode()[0]['periode'];

    $data = array(
        'jawab_id' => $id_jawab,
        'per_id' => $per_id,
        'user_id' => $user,
        'periode_id' => $periode,
        'jawaban' => 'tidak'
    );

    $logic = $this->m_kue->simpan_jawaban_tidak($data);
    if ($logic==true) {
        redirect('c_kue/view_pol');
    }
}
```

Gambar 4.33 Kode Program `simpan_jawaban_tidak`

Gambar 4.33 merupakan *function* yang terletak di kelas `c_kue` yang digunakan untuk menyimpan jawaban tidak saat responden memilih jawaban tidak.

```
public function step1(){
    // SET KONVERSI JAWABAN
    $konversi = $this->m_kue->get_jawaban();
    $this->m_kue->simpan_konversi($konversi);

    // SET JUMLAH PERNYATAAN TIAP LEVEL KONTROL OBJEKTIF
    $jml = $this->m_kue->set_jml_pernyataan();
    $this->m_kue->simpan_jml_pernyataan($jml);

    redirect('c_hasil/view_periode');
}
```

Gambar 4.34 Kode Program Function `step1`

Gambar 4.34 merupakan *function* yang terletak di kelas `c_kue` yang digunakan untuk mengambil dan menyimpan nilai perhitungan tahap 1 yang dilakukan oleh kelas `m_kue`.

```
public function step2(){
    // // SET RATA2 KONVERSI
    $avg = $this->m_kue->set_avg_konversi();
    $this->m_kue->simpan_avg_konversi($avg);

    // SET TOTAL AVG KONVERSI
    $total = $this->m_kue->set_tot_avg_konversi();
    $this->m_kue->simpan_tot_avg_konversi($total);

    redirect('c_hasil/view_periode');
}
```

Gambar 4.35 Kode Program Function `step2`

Gambar 4.35 merupakan *function* yang terletak di kelas `c_kue` yang digunakan untuk mengambil dan menyimpan nilai perhitungan tahap 2 yang dilakukan oleh kelas `m_kue`.

```
public function step3(){  
    // SET JUMLAH RATA2 KONVERSI TIAP LEVEL KONTROL OBJEKTIF  
    $jml_avg = $this->m_kue->set_jml_avg_konversi();  
    $this->m_kue->simpan_jml_avg_konversi($jml_avg);  
  
    redirect('c_hasil/view_periode');  
}
```

Gambar 4.36 Kode Program Function step3

Gambar 4.36 merupakan *function* yang terletak di kelas `c_kue` yang digunakan untuk mengambil dan menyimpan nilai perhitungan tahap 3 yang dilakukan oleh kelas `m_kue`.

```
public function step4(){  
    // SET NORMALISASI  
    $normalisasi = $this->m_kue->set_normalisasi();  
    $this->m_kue->simpan_normalisasi($normalisasi);  
  
    // SET KONTRIBUSI  
    $kontribusi = $this->m_kue->set_kontribusi();  
    $this->m_kue->simpan_kontribusi($kontribusi);  
  
    // NILAI KONTRIBUSI  
    $hasil = $this->m_kue->hasil();  
    $this->m_kue->simpan_hasil($hasil);  
  
    redirect('c_hasil/view_periode');  
}
```

Gambar 4.37 Kode Program Function step4

Gambar 4.37 merupakan *function* yang terletak di kelas `c_kue` yang digunakan untuk mengambil dan menyimpan nilai perhitungan tahap 4 yang dilakukan oleh kelas `m_kue`.


```

public function get_nilaico17($periode_id) {
    $query = $this->db->query("SELECT SUM(j.hasil_k/2) as mat_co FROM tb_jawaban j join
    tb_pernyataan p on j.per_id=p.per_id WHERE p.co_id='AI7' AND j.periode_id='$periode_id'")->result_array();
    return $query;
}

```

Gambar 4.38 Kode Program Function get_nilaico17

Gambar 4.38 merupakan *function* yang terletak di kelas m_hasil yang digunakan untuk menghitung nilai *maturity level* kontrol objektif.

```

public function simpan_hasil($data){
    return $this->db->insert('tb_nilai', $data);
}

```

Gambar 4.39 Kode Program Function simpan_hasil

Gambar 4.39 merupakan *function* yang terletak di kelas m_hasil yang digunakan untuk menyimpan nilai *maturity level* kontrol objektif.

```

public function step5($periode_id){
    $id_nilai = $this->m_hasil->set_id_nilai();
    $periode = $this->m_hasil->get_periode()[0]['periode'];
    $nilai1 = $this->m_hasil->get_nilaico1($periode_id)[0]['mat_co'];
    $data1 = array(
        'nilai_id' => $id_nilai,
        'periode_id' => $periode,
        'co_id' => 'P01',
        'nilai' => $nilai1
    );
    $logic1 = $this->m_hasil->simpan_hasil($data1);
}

```

Gambar 4.40 Kode Program Function step5

Gambar 4.38 merupakan *function* yang terletak di kelas c_hasil yang digunakan untuk mengambil dan menyimpan nilai perhitungan tahap 5 yang dilakukan oleh kelas m_hasil.

4.3.9 Kode Program *View* Hasil Perhitungan

Kode program *view* hasil perhitungan terletak pada kelas *v_hasil*, *v_hasil_mat*, *v_hasil_final*, *c_hasil*, dan *m_hasil*. Penulisan kode program ini dapat dilihat pada lampiran E.

4.4 Pengujian Sistem

Pengujian dilakukan untuk mengevaluasi aplikasi yang telah dibuat. Proses pengujian dilakukan dengan pengujian *whitebox* terlebih dahulu kemudian akan dilanjutkan dengan pengujian *blackbox*.

4.4.1 Pengujian *White Box*

Pengujian *white box* pada aplikasi penilaian tingkat kematangan tata kelola TI ini dengan cara menggambar diagram alir, menghitung kompleksitas siklomatiknya (CC), dan membuat tabel pengujian *test case*. Pengujian kompleksitas siklomatik pada fitur mengisi kuesioner dapat dilihat pada Gambar 4.41 sampai dengan Gambar 4.69 dan tabel pengujian *test case* dapat dilihat pada Tabel 4.6 sampai dengan Tabel 2.34. Sedangkan untuk fitur yang lain dapat dilihat pada lampiran F.

69

Gambar 4.41 Diagram Alir *Function* *simpan_jawaban_ya*

$$CC = E - N + 2 = 0 - 1 + 2 = 1$$

Tabel 4.6 *Test Case Function* *simpan_jawaban_ya*

Test case	Jika berhasil menyimpan jawaban “ya”
Target yang Diharapkan	Menyimpan jawaban ke <i>database</i>
Hasil pengujian	Benar

Path/jalur	69
-------------------	----

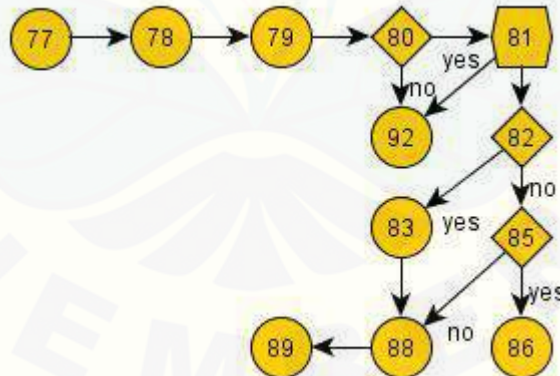
73

Gambar 4.42 Diagram Alir *Function* *simpan_jawaban_tidak*

$$CC = E - N + 2 = 0 - 1 + 2 = 1$$

Tabel 4.7 *Test Case Function* *simpan_jawaban_tidak*

Test case	Jika berhasil menyimpan jawaban “tidak”
Target yang Diharapkan	Menyimpan jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	73

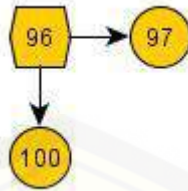


Gambar 4.43 Diagram Alir *Function* *get_jawaban*

$$CC = E - N + 2 = 13 - 12 + 2 = 3$$

Tabel 4.8 Test Case Function get_jawaban

Test case	Jika berhasil mengambil nilai jawaban yang telah disimpan di <i>database</i> dan jawaban adalah “ya”
Target yang Diharapkan	Jawaban terkonversi menjadi 1 untuk jawaban “ya”
Hasil pengujian	Benar
Path/jalur	77-78-79-80-81-82-83-88-89
Test case	Jika berhasil mengambil nilai jawaban yang telah disimpan di <i>database</i> dan jawaban adalah “tidak”
Target yang Diharapkan	Jawaban terkonversi menjadi 0 untuk jawaban “tidak”
Hasil pengujian	Benar
Path/jalur	77-78-79-80-81-82-85-88-89
Test case	Jika tidak ada nilai jawaban yang bisa diambil atau kolom jawaban pada <i>database</i> kosong
Target yang Diharapkan	Nilai jawaban kosong
Hasil pengujian	Benar
Path/jalur	77-78-79-80-92

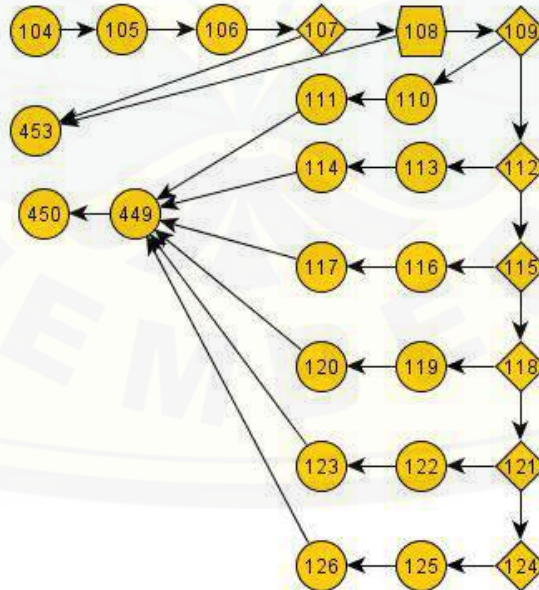


Gambar 4.44 Diagram Alir *Function* *simpan_konversi*

$$CC = E - N + 2 = 2 - 2 + 2 = 2$$

Tabel 4.9 *Test Case Function* *simpan_konversi*

Test case	Jika berhasil menyimpan hasil konversi jawaban
Target yang Diharapkan	Menyimpan hasil konversi jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	96-97, 97-100



Gambar 4.45 Diagram Alir *Function* *set_jml_pernyataan*

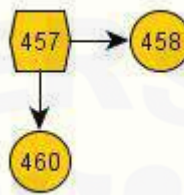
Proses pada Gambar 4.40 jalur 109 – 110 – 111 – 112 – 113 – 114 – 115 – 116 – 117 – 118 – 119 – 120 – 121 – 123 – 124 – 125 – 126 dan *edge* yang mengarah ke node 109 dan *edge* yang mengarah ke node 449 akan diulangi sebanyak 17 kali pada baris berikutnya sampai pada baris ke 666 sesuai dengan *listing program* pada Gambar 4.17.

$$CC = E - N + 2 = 415 - 314 + 2 = 103$$

Tabel 4.10 *Test Case Function set_jml_pernyataan*

Test case	Jika berhasil mengambil nilai id pernyataan dari <i>database</i>
Target yang Diharapkan	Ada nilai id pernyataan yang diambil
Hasil pengujian	Benar
Path/jalur	104-105-106-107-108-453
Test case	Jika tidak ada nilai id pernyataan yang bisa diambil dari <i>database</i> atau kolom id pernyataan kosong
Target yang Diharapkan	Nilai id pernyataan kosong
Hasil pengujian	Benar
Path/jalur	104-105-106-107-453
Test case	Menghitung nilai <i>count</i> (jumlah pernyataan)
Target yang Diharapkan	Menghasilkan nilai jumlah pernyataan
Hasil pengujian	Benar
Path/jalur	104-105-106-108-109-110-111-449-450

	104-105-106-108-109-112-113-114-449-450
	104-105-106-108-109-112-115-116-117-449-450

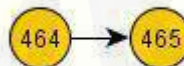


Gambar 4.46 Diagram Alir *Function* simpan_jml_pernyataan

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.11 *Test Case Function* simpan_jml_pernyataan

Test case	Jika berhasil menyimpan hasil count jumlah pernyataan
Target yang Diharapkan	Menyimpan hasil count jumlah pernyataan ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	457-458, 457-460

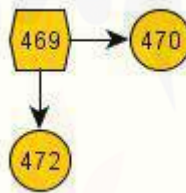


Gambar 4.47 Diagram Alir *Function* set_avg_konversi

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 4.12 *Test Case Function set_avg_konversi*

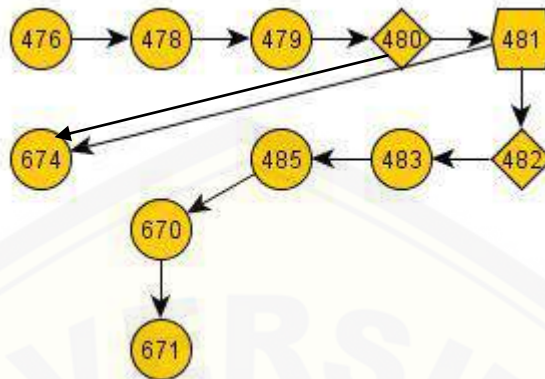
Test case	Jika berhasil membuat nilai rata-rata konversi jawaban
Target yang Diharapkan	Menghasilkan nilai rata-rata konversi jawaban
Hasil pengujian	Benar
Path/jalur	464-465

Gambar 4.48 Diagram Alir *Function simpan_avg_konversi*

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.13 *Test Case Function simpan_avg_konversi*

Test case	Jika berhasil menyimpan nilai rata-rata konversi jawaban
Target yang Diharapkan	Menyimpan nilai rata-rata konversi jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	469-470, 469-472



Gambar 4.49 Diagram Alir *Function* set_tot_avg_konversi

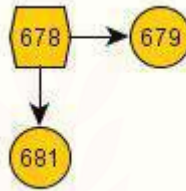
Proses pada Gambar 4.44 jalur 482 – 483 – 485 dan *edge* yang mengarah ke node 482 dan *edge* yang mengarah ke node 670 akan diulangi sebanyak 34 kali pada baris berikutnya sampai baris ke 666 sesuai dengan *listing program* pada Gambar 4.21.

$$CC = E - N + 2 = 142 - 110 + 2 = 34$$

Tabel 4.14 *Test Case Function* set_tot_avg_konversi

Test case	Jika berhasil mengambil nilai rata-rata konversi dari <i>database</i>
Target yang Diharapkan	Ada nilai rata-rata konversi yang diambil
Hasil pengujian	Benar
Path/jalur	476-477-478-479-480-481-674
Test case	Jika tidak ada nilai rata-rata konversi yang bisa diambil dari <i>database</i> atau kolom rata-rata konversi kosong
Target yang Diharapkan	Nilai rata-rata konversi kosong
Hasil pengujian	Benar

Path/jalur	476-477-478-479-480-674
Test case	Menghitung nilai sum(rata-rata konversi)
Target yang Diharapkan	Menghasilkan nilai total rata-rata konversi
Hasil pengujian	Benar
Path/jalur	476-477-478-479-480-481-482-483-484-485-670-671

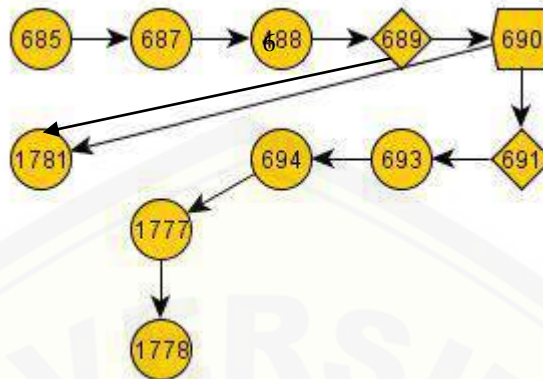


Gambar 4.50 Diagram Alir *Function* simpan_tot_avg_konversi

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.15 *Test Case Function* simpan_tot_avg_konversi

Test case	Jika berhasil menyimpan nilai total rata-rata konversi jawaban
Target yang Diharapkan	Menyimpan nilai total rata-rata konversi jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	678-679, 678-681



Gambar 4.51 Diagram Alir *Function* set_jml_avg_konversi

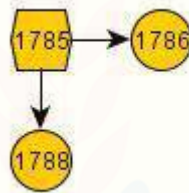
Proses pada Gambar 4.46 jalur 691 – 693 - 694 dan *edge* yang mengarah ke node 691 dan *edge* yang mengarah ke node 1777 akan diulangi sebanyak 204 kali pada baris berikutnya sampai pada baris ke 1774 sesuai dengan *listing program* pada Gambar 4.23.

$$CC = E - N + 2 = 822 - 620 + 2 = 204$$

Tabel 4.16 *Test Case Function* set_jml_avg_konversi

Test case	Jika berhasil mengambil nilai rata-rata konversi dari <i>database</i>
Target yang Diharapkan	Ada nilai rata-rata konversi yang diambil
Hasil pengujian	Benar
Path/jalur	685-686-687-688-689-690-1781
Test case	Jika tidak ada nilai rata-rata konversi yang bisa diambil dari <i>database</i> atau kolom rata-rata konversi kosong
Target yang Diharapkan	Nilai rata-rata konversi kosong

Hasil pengujian	Benar
Path/jalur	685-686-687-688-689-1781
Test case	Menghitung nilai sum(rata-rata konversi) pada setiap level
Target yang Diharapkan	Menghasilkan nilai jumlah rata-rata konversi
Hasil pengujian	Benar
Path/jalur	685-686-687-688-689-690-691-693-694-1777-1778

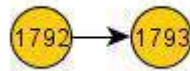


Gambar 4.52 Diagram Alir *Function* simpan_jml_avg_konversi

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.17 *Test Case Function* simpan_jml_avg_konversi

Test case	Jika berhasil menyimpan nilai jumlah rata-rata konversi jawaban
Target yang Diharapkan	Menyimpan nilai jumlah rata-rata konversi jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	1785-1786, 1785-1788

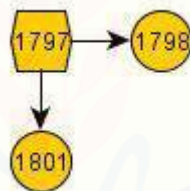


Gambar 4.53 Diagram Alir *Function* set_normalisasi

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 4.18 *Test Case Function* set_normalisasi

Test case	Jika berhasil membuat nilai normalisasi
Target yang Diharapkan	Menghasilkan nilai normalisasi
Hasil pengujian	Benar
Path/jalur	1792 – 1793

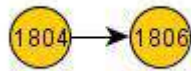


Gambar 4.54 Diagram Alir *Function* simpan_normalisasi

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.19 *Test Case Function* simpan_normalisasi

Test case	Jika berhasil menyimpan nilai normalisasi
Target yang Diharapkan	Menyimpan nilai normalisasi ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	1797-1798, 1797-1801

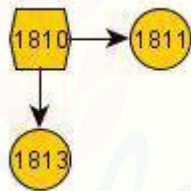


Gambar 4.55 Diagram Alir *Function* set_kontribusi

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 4.20 *Test Case Function* set_kontribusi

Test case	Jika berhasil membuat nilai kontribusi
Target yang Diharapkan	Menghasilkan nilai kontribusi
Hasil pengujian	Benar
Path/jalur	1804-1806

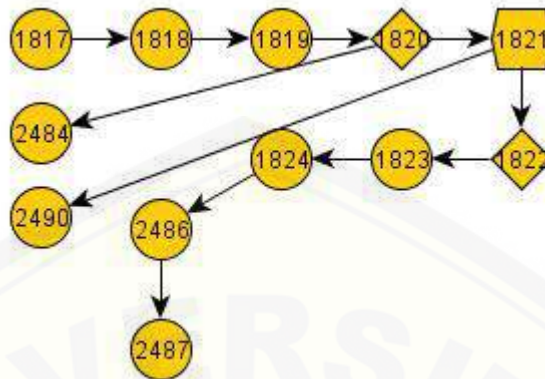


Gambar 4.56 Diagram Alir *Function* simpan_kontribusi

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.21 *Test Case Function* simpan_kontribusi

Test case	Jika berhasil menyimpan nilai kontribusi
Target yang Diharapkan	Menyimpan nilai kontribusi ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	1810-1811, 1810-1813



Gambar 4.57 Diagram Alir *Function* hasil

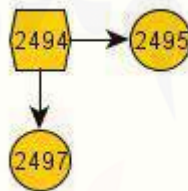
Proses pada Gambar 4.52 jalur 1822 – 1823 - 1824 dan *edge* yang mengarah ke node 1822 dan *edge* yang mengarah ke node 2486 akan diulangi sebanyak 204 kali pada baris berikutnya sampai dengan baris ke 2481 sesuai dengan *listing program* pada Gambar 4.29.

$$CC = E - N + 2 = 823 - 621 + 2 = 204$$

Tabel 4.22 *Test Case Function* hasil

Test case	Jika berhasil mengambil nilai kontribusi dari <i>database</i>
Target yang Diharapkan	Ada nilai kontribusi yang diambil
Hasil pengujian	Benar
Path/jalur	1817-1818-1819-1820-1821-2490
Test case	Jika tidak ada nilai kontribusi yang bisa diambil dari <i>database</i> atau nilai tidak sesuai dengan kondisi
Target yang Diharapkan	Kolom hasil bernilai 0
Hasil pengujian	Benar

Path/jalur	1817-1818-1819-1820-2484
Test case	Mengambil nilai kontribusi setiap level
Target yang Diharapkan	Menghasilkan nilai kontribusi
Hasil pengujian	Benar
Path/jalur	1817-1818-1819-1820-1821-1822-1823-1824-2486-2487

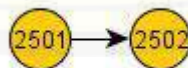


Gambar 4.58 Diagram Alir *Function* *simpan_hasil*

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.23 *Test Case Function* *simpan_hasil*

Test case	Jika berhasil menyimpan nilai hasil
Target yang Diharapkan	Menyimpan nilai hasil rata-rata konversi jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	2494-2495, 2494-2497

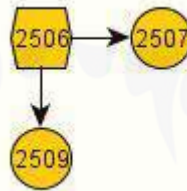


Gambar 4.59 Diagram Alir *Function* *get_periode*

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 4.24 *Test Case Function* get_periode

Test case	Jika berhasil mengambil nilai periode
Target yang Diharapkan	Menghasilkan nilai periode
Hasil pengujian	Benar
Path/jalur	2501-2502

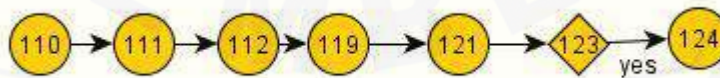


Gambar 4.60 Diagram Alir *Function* simpan_periode

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.25 *Test Case Function* simpan_periode

Test case	Jika berhasil menyimpan nilai periode
Target yang Diharapkan	Menyimpan nilai periode ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	2506-2507, 2506-2509

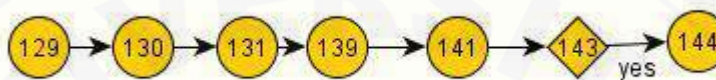


Gambar 4.61 Diagram Alir *Function* simpan_jawaban_ya

$$CC = E - N + 2 = 6 - 7 + 2 = 1$$

Tabel 4.26 Test Case Function simpan_jawaban_ya

Test case	Jika berhasil menyimpan nilai jawaban “ya”
Target yang Diharapkan	Menyimpan nilai jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	110-111-112-119-123-124



Gambar 4.62 Diagram Alir Function simpan_jawaban_tidak

$$CC = E - N + 2 = 6 - 7 + 2 = 1$$

Tabel 4.27 Test Case Function simpan_jawaban_tidak

Test case	Jika berhasil menyimpan nilai jawaban “tidak”
Target yang Diharapkan	Menyimpan nilai jawaban ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	129-130-131-139-141-143-144



Gambar 4.63 Diagram Alir Function step1

$$CC = E - N + 2 = 4 - 5 + 2 = 1$$

Tabel 4.28 Test Case Function step1

Test case	Jika berhasil menyimpan nilai perhitungan tahap 1
------------------	---

Target yang Diharapkan	Menyimpan nilai perhitungan tahap 1 ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	196-197-200-201-203



Gambar 4.64 Diagram Alir *Function* step2

$$CC = E - N + 2 = 4 - 5 + 2 = 1$$

Tabel 4.29 *Test Case Function* step2

Test case	Jika berhasil menyimpan nilai perhitungan tahap 2
Target yang Diharapkan	Menyimpan nilai perhitungan tahap 2 ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	208-209-212-215



Gambar 4.65 Diagram Alir *Function* step3

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 4.30 *Test Case Function* step3

Test case	Jika berhasil menyimpan nilai perhitungan tahap 3
Target yang Diharapkan	Menyimpan nilai perhitungan tahap 3 ke <i>database</i>

Hasil pengujian	Benar
Path/jalur	220-221-223



Gambar 4.66 Diagram Alir *Function* step4

$$CC = E - N + 2 = 4 - 5 + 2 = 1$$

Tabel 4.31 *Test Case Function* step4

Test case	Jika berhasil menyimpan nilai perhitungan tahap 4
Target yang Diharapkan	Menyimpan nilai perhitungan tahap 4 ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	228-229-232-233-239



Gambar 4.67 Diagram Alir *Function* get_nilaico

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 4.32 *Test Case Function* get_nilaico

Test case	Jika berhasil mengambil nilai maturity level tiap kontrol objektif
Target yang Diharapkan	Menghasilkan nilai maturity level
Hasil pengujian	Benar
Path/jalur	358-360

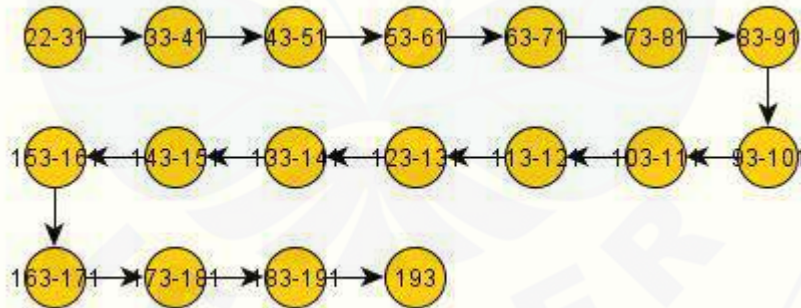
376

Gambar 4.68 Diagram Alir *Function* simpan_hasil

$$CC = E - N + 2 = 0 - 1 + 2 = 1$$

Tabel 4.33 *Test Case Function* simpan_hasil

Test case	Jika berhasil menyimpan nilai maturity level tiap kontrol objektif
Target yang Diharapkan	Menyimpan nilai mengambil nilai maturity level tiap kontrol objektif ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	376



Gambar 4.69 Diagram Alir *Function* step5

$$CC = E - N + 2 = 17 - 18 + 2 = 1$$

Tabel 4.34 *Test Case Function* step5

Test case	Jika berhasil menyimpan nilai perhitungan tahap 5
------------------	---

Target yang Diharapkan	Menyimpan nilai perhitungan tahap 5 ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	(22-31) – (33-41) – (43-51) – (53-61) – (63-71) – (73-81) – (83-91) – (93-101) – (103-111) – (113-121) – (123-131) – (133-141) – (143-151) – (153-161) – (163-171) – (173-181) – (183-191) – 193

4.4.2 Pengujian *Black Box*

Pengujian *black box* merupakan pengujian yang dilakukan oleh *user* dengan menjalankan program secara langsung dan menganalisis *input* dan *output* yang dihasilkan aplikasi. Pengujian *black box* untuk fitur mengisi kuesioner dapat dilihat pada Tabel 4.35 dan untuk fitur melihat hasil perhitungan dapat dilihat pada Tabel 4.36. Sedangkan untuk pengujian fitur yang lain dapat dilihat pada lampiran G.

Tabel 4.35 Pengujian *Black Box* Mengisi Kuesioner

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Mengisi Kuesioner	Memilih menu kuesioner	Menampilkan tabel daftar domain dan kontrol objektif	Berhasil
		Memilih kontrol objektif	a. Mengambil data pernyataan dari <i>database</i> b. Menampilkan data pernyataan sesuai kontrol objektif pada tabel	Berhasil

		Memilih jawaban	<ul style="list-style-type: none"> a. Menyimpan data ke <i>database</i> b. Menampilkan data pernyataan kembali c. Memberi status pada pernyataan yang sudah dipilih 	Berhasil
		Memilih tombol selesai	Menampilkan halaman data domain dan kontrol objektif	Berhasil

Tabel 4.36 Pengujian *Black Box View* Hasil Perhitungan

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melakukan perhitungan nilai <i>maturity level</i>	Memilih menu hasil audit	Menampilkan tabel daftar periode	Berhasil
		Memilih tombol hitung step 1	<ul style="list-style-type: none"> a. Melakukan perhitungan tahap 1 b. Menyimpan hasil ke <i>database</i> c. Menampilkan halaman hasil perhitungan kembali 	Berhasil

		Memilih tombol hitung step 2	a. Melakukan perhitungan tahap 2 b. Menyimpan hasil ke <i>database</i> c. Menampilkan halaman hasil perhitungan kembali	Berhasil
		Memilih tombol hitung step 3	a. Melakukan perhitungan tahap 3 b. Menyimpan hasil ke <i>database</i> c. Menampilkan halaman hasil perhitungan kembali	
		Memilih tombol hitung step 4	a. Melakukan perhitungan tahap 4 b. Menyimpan hasil ke <i>database</i> c. Menampilkan halaman hasil perhitungan kembali	
		Memilih tombol hitung step 5	a. Melakukan perhitungan tahap 5 b. Menyimpan hasil ke <i>database</i> c. Menampilkan halaman hasil perhitungan kembali	

2.	Melihat nilai <i>maturity level</i> kontrol objektif	Memilih tombol <i>maturity level</i>	a. Menampilkan nilai <i>maturity level</i> tiap user pada kontrol objektif	Berhasil
3.	Melihat nilai <i>maturity level</i>	Memilih tombol <i>maturity level</i>	a. Menampilkan nilai <i>maturity level</i> b. Menampilkan grafik nilai <i>maturity level</i> kontrol objektif	Berhasil
		Memilih tombol selesai	Menampilkan halaman data domain dan kontrol objektif	Berhasil

BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang dilakukan adalah:

1. *Framework* COBIT digunakan untuk melakukan penilaian tingkat kematangan tata kelola TI. Domain PO berfokus pada strategi TI yang diterapkan pada perusahaan dan domain AI berfokus pada pemberian solusi untuk perubahan yang terjadi pada perusahaan berhubungan dengan penerapan TI pada domain PO.
2. Pengembangan aplikasi menggunakan SDLC *waterfall* yang pada penerapannya memiliki kelemahan yaitu terjadinya iterasi yang menyebabkan proses pengembangan kurang mengikuti urutan sekuensial seperti pada teori.
3. Aplikasi penilaian tingkat kematangan tata kelola TI merupakan pengembangan aplikasi untuk penilaian tingkat kematangan yang berbentuk kuesioner berbasis web.
4. Tingkat kematangan tata kelola TI pada SISTER Universitas Jember yang penelitiannya dilakukan pada UPT TI Universitas Jember berada pada level 3 yang menunjukkan bahwa proses perencanaan dan penerapan solusi yang diterapkan pada UPT TI untuk mencapai tujuan UPT TI termasuk dalam kategori yang sudah didokumentasikan dan dikomunikasikan. Kontrol objektif dengan nilai tertinggi yaitu kontrol objektif PO 10 dengan nilai 3.58 yang membahas mengenai manajemen proses pengelolaan proyek yang memenuhi kebutuhan bisnis TI dan kontrol objektif AI 3 dengan nilai 3.51 yang membahas mengenai manajemen

proses dalam memperoleh dan merawat infrastruktur teknologi yang memenuhi bisnis TI. Sedangkan kontrol objektif dengan nilai terendah yaitu kontrol objektif PO 2 dengan nilai 2.68 yang membahas mengenai penetapan arsitektur informasi dan kontrol objektif AI 4 dengan nilai 2.29 yang membahas mengenai proses operasi dan penggunaannya.

5. Perolehan nilai dari hasil kuesioner dipengaruhi oleh kesamaan jawaban antarresponden. Semakin banyak jawaban yang sama pada setiap kontrol objektif, maka nilai *maturity level* yang dihasilkan akan semakin tinggi dan begitu pula sebaliknya.
6. *Framework* COBIT 4.1 dengan domain PO dan AI dapat diterapkan pada proses penilaian tingkat kematangan tata kelola TI di UPT TI Universitas Jember yang merupakan organisasi di bawah naungan sebuah instansi pendidikan.

6.2 Saran

Beberapa saran berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu:

1. Kuesioner yang digunakan untuk penelitian hendaknya lebih terfokus dan disesuaikan untuk setiap responden yang akan mengisi.
2. Aplikasi dibuat yang lebih mudah dimengerti oleh *user*.

DAFTAR PUSTAKA

- Fitroh. (2012). *Penilaian Tingkat Kematangan Tata Kelola TI Pada Sistem Informasi Manajemen Akademik*.
- ITGI. (2004). *COBIT Student Book*. IT Governance Institute.
- ITGI. (2007). *COBIT 4.1*. IT Governance Institute.
- Kesumawardhani, D. (2012). *Evaluasi IT Governance Berdasarkan COBIT 4.1 (Studi Kasus di PT Timah (Persero) Tbk)*.
- Pederiva, A. (2003). *The COBIT Maturity Model in a Vendor Evaluation Case. Information System Control Journal : Volume 3*.
- Pradana, D. (n.d.). *Evaluasi Kinerja Sistem Informasi I-POS 4.0.3 Menggunakan Framework COBIT 4.1 (Studi Kasus PT. POS Indonesia MPC Semarang)*.
- Rosalika, L. M. (n.d.). *Audit Sistem Informasi Perpustakaan Universitas Kristen Petra Berdasarkan Standar Control Objectives For Information and Related Technology (COBIT 4.0)*.
- Siraji, M. H. (2011). *Penilaian Tata Kelola Teknologi Informasi Pada Aplikasi CSBO Dengan Menggunakan Framework COBIT 4.0 Domain PO dan AI*.
- Sommerville, I. (2003). *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.
- Surendro, K. (2008). *Rancangan Tata Kelola Teknologi Informasi untuk Pabrik Pupuk. Jurnal Informatika Vol. 9, 115-121*.
- Zandhessami, H., & Kargar, M. H. (2010). *Prioritization of COBIT Framework processes based on Sazehgostar Saipa's Information Technology Strategies*.

LAMPIRAN**A. Hasil Kuesioner****A.1 Hasil Kuesioner Responden 1**

Hasil kuesioner dari responden 1 (bagian manajerial) akan dijelaskan pada Tabel 1.

Tabel 1 Hasil Kuesioner Responden 1

**ANALISA TINGKAT KEMATANGAN TATA KELOLA TI SISTEM
INFORMASI TERPADU (SISTER) UNIVERSITAS JEMBER**

Kuesioner ini dikembangkan dari standart pengelolaan teknologi informasi dari *Control Objectives for Information and Related Technology* (COBIT) dengan fokus pada domain *Plan and Organize* (PO) dan *Acquire and Implement* (AI). Untuk itu mohon kesediaan Bapak/Ibu untuk memberikan pendapatnya akan pernyataan-pernyataan yang diberikan dalam kuesioner ini sesuai dengan teknologi informasi Sistem Informasi Terpadu (SISTER) yang di kelola di Universitas Jember.

Petunjuk Pengisian

Bacalah pernyataan kriteria dari tingkat kematangan dengan seksama, lalu berikan centang (✓) pada pilihan jawaban (Ya/Tidak) untuk setiap pernyataan yang diberikan.

Nama Responden	: Andrei Ramani, S.KM., M.KM.
Bagian	: Manajerial
Jabatan	: Sekretaris

Plan and Organize (PO)**PO 1 - Menetapkan Perencanaan Strategi TI**

Manajemen proses mendefinisikan rencana strategi TI yang memenuhi kebutuhan bisnis TI dari dukungan dan perluasan strategi bisnis dan ketentuan penguasaan ketika transparan tentang keuntungan, biaya, dan resiko-resiko, adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Perencanaan strategi tidak bekerja.		√
	2. Tidak ada kesadaran manajemen bahwa perencanaan strategi TI diperlukan untuk mendukung tujuan-tujuan bisnis.		√
1 <i>Initial/Ad- hoc</i>	1. Kebutuhan untuk perencanaan strategi TI dikenal oleh manajemen TI.	√	
	2. Perencanaan TI dilakukan sebagai kebutuhan dasar sebagai jawaban atas kebutuhan perusahaan tertentu.	√	
	3. Perencanaan strategi TI adakalanya didiskusikan pada pertemuan manajemen TI.	√	
	4. Penjajaran kebutuhan bisnis, aplikasi, dan teknologi terjadi dengan reaktif dibanding dengan strategi seluruh perusahaan.		√
	5. Posisi resiko strategi dikenali dengan tidak formal dalam proyek ke proyek.	√	
2	1. Perencanaan strategi TI terbagi dengan manajemen bisnis sebagai kebutuhan dasar.	√	

<i>Repeatable but Intuitive</i>	2. Membaharui rencana TI terjadi sebagai jawaban atas permintaan manajemen.	√	
	3. Keputusan yang strategis digerakkan dalam proyek ke proyek, tanpa konsistensi dengan seluruh strategi perusahaan.		√
	4. Resiko-resiko dan keuntungan-keuntungan <i>user</i> dari keputusan strategi utama diakui dengan cara intuitif.	√	
3 <i>Define Process</i>	1. Sebuah kebijakan menjelaskan kapan dan bagaimana untuk melakukan perencanaan strategi TI.	√	
	2. Perencanaan strategi TI mengikuti pendekatan yang terstruktur yang didokumentasikan dan diketahui oleh semua staff.		√
	3. Proses perencanaan TI layak bunyi dan menjamin bahwa perencanaan sesuai seperti yang dilakukan, namun kebijaksanaan diberikan kepada manajer individual berkenaan dengan proses implementasi dan tidak ada prosedur untuk menguji proses.		√
	4. Seluruh strategi TI meliputi penjelasan yang konsisten dari resiko-resiko yang mana perusahaan rela ambil sebagai sebuah pembaharuan atau penyokong.	√	
	5. Strategi teknis, keuangan, dan sumber daya manusia sangat mempengaruhi tambahan produk dan teknologi baru.	√	

	6. Perencanaan strategi TI didiskusikan saat pertemuan manajemen bisnis.	√	
4 <i>Manage and Measurable</i>	1. Perencanaan strategi TI adalah praktik standart dan eksepsi yang diumumkan oleh manajemen.	√	
	2. Perencanaan strategi TI adalah fungsi manajemen yang baik dengan tanggung jawab tingkat atasan.	√	
	3. Manajemen dapat mengawasi proses perencanaan strategi TI, membuat perencanaan jelas dan mengukur keefektifannya.	√	
	4. Perencanaan TI jangka pendek dan jangka panjang terjadi dan mengalir ke bawah dalam perusahaan dengan pembaharuan yang dilakukan seperti yang dibutuhkan.	√	
	5. Strategi TI dan strategi seluruh perusahaan semakin menjadi lebih serasi dengan menunjukkan proses bisnis dan kemampuan nilai tambah dan mengungkit penggunaan aplikasi dan teknologi melalui merekayasa ulang proses bisnis.	√	
	6. Ada proses yang baik untuk menentukan penggunaan sumber internal dan eksternal yang dibutuhkan dalam operasi dan pengembangan sistem.	√	
5	1. Perencanaan strategi TI terdokumentasi, proses hidup secara terus-menerus dipertimbangkan dalam <i>setting</i> tujuan bisnis dan berakibat pada nilai bisnis yang terlihat melalui investasi dalam TI.	√	

<i>Optimised</i>	2. Pertimbangan nilai tambah dan resiko terus-menerus diperbaharui dalam proses perencanaan strategi TI.	√	
	3. Perencanaan TI realistis dikembangkan dan tetap diperbaharui untuk mencerminkan perubahan teknologi dan pengembangan terkait bisnis.	√	
	4. <i>Benchmarking</i> terhadap pemahaman yang baik dan norma-norma industri dapat dipercaya terjadi dan terintegrasi dengan proses perumusan strategi.	√	
	5. Perencanaan yang strategis meliputi bagaimana pengembangan teknologi baru dapat menggerakkan penciptaan kemampuan bisnis baru dan meningkatkan keuntungan kompetitif perusahaan.	√	

PO 2 - Menetapkan Arsitektur Informasi

Manajemen proses mendefinisikan arsitektur informasi yang kebutuhan bisnis TI menjadi tangkas dalam menjawab kebutuhan, memberikan keandalan, informasi yang konsisten tanpa kliem terintegrasi dalam aplikasi dalam proses bisnis adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Tidak ada kesadaran pentingnya arsitektur informasi bagi perusahaan.		√
	2. Pengetahuan, keahlian, dan tanggung jawab perlu untuk mengembangkan arsitektur ini yang tidak ada dalam perusahaan.	√	
1	1. Manajemen mengakui kebutuhan untuk arsitektur informasi.	√	

<i>Initial/Ad-hoc</i>	2. Pengembangan beberapa komponen dari sebuah arsitektur terjadi pada dasar tertentu.	√	
	3. Definisi menunjukkan data dibanding informasi dan digerakkan oleh tawaran vendor aplikasi <i>software</i> .		√
	4. Ada komunikasi yang jarang-jarang dan tidak konsisten dari kebutuhan bagi arsitektur informasi.		√
2 <i>Repeatable but Intuitive</i>	1. Proses arsitektur informasi muncul dan mirip, meskipun informal dan intuitif, prosedur-prosedur diikuti dengan individu-individu yang berbeda di dalam perusahaan.		√
	2. Orang-orang mendapat skill mereka dengan membangun arsitektur informasi melalui pengalaman yang dimiliki dan aplikasi teknik-teknik yang diulang.	√	
	3. Kebutuhan taktis menggerakkan pengembangan komponen arsitektur informasi oleh individu-individu.	√	
3 <i>Define Process</i>	1. Pentingnya arsitektur informasi dipahami dan diterima, dan pertanggungjawaban bagi penyampaian ditetapkan dan dikomunikasikan dengan jelas.	√	
	2. Prosedur, tools, dan teknik terkait meskipun tidak canggih telah distandarisasi dan didokumentasi dan menjadi bagian aktivitas <i>training</i> informal.		√

	3. Kebijakan-kebijakan arsitektur informasi dasar telah dikembangkan, meliputi beberapa kebutuhan yang strategis tetapi sesuai dengan kebijakan-kebijakan, standar-standar, dan <i>tools</i> dengan konsisten ditekankan.	√	
	4. Secara formal fungsi administrasi data yang baik adalah pada tempatnya, <i>setting</i> standar seluruh perusahaan, dan mulai untuk melaporkan penyampaian dan penggunaan arsitektur informasi.	√	
	5. <i>Tools</i> yang otomatis mulai dikerjakan, tetapi penggunaan proses dan peraturan ditetapkan dengan tawaran vendor <i>software</i> database.		√
	6. Aktivitas-aktivitas <i>training</i> formal ditetapkan, didokumentasi, dan terus diterapkan.	√	
4 <i>Manage and Measurable</i>	1. Pengembangan dan pelaksanaan arsitektur informasi secara penuh didukung dengan metode dan teknik formal.		√
	2. Akuntabilitas bagi kinerja proses pengembangan arsitektur dilaksanakan dan keberhasilan arsitektur informasi dapat diukur.		√
	3. Dukungan <i>tools</i> yang otomatis tersebar luas, tetapi belum terintegrasi.		√
	4. Matrik dasar telah dikenali dan sebuah sistem pengukuran adalah pada tempatnya.	√	
	5. Proses penjelasan arsitektur informasi adalah proaktif dan terfokus pada tujuan kebutuhan bisnis masa depan.	√	

	6. Organisasi administrasi data secara aktif terlibat dalam semua usaha-usaha pengembangan aplikasi, untuk menjamin kemantapan.	√	
	7. Penyimpanan yang otomatis dengan penuh diterapkan.	√	
	8. Model data lebih kompleks diterapkan untuk mengungkit isi informasi dari database.	√	
	9. Sistem informasi eksekutif dan sistem pendukung keputusan mengungkit ketersediaan informasi.	√	
5 <i>Optimised</i>	1. Arsitektur informasi secara konsisten dilaksanakan pada semua level.		√
	2. Nilai arsitektur informasi pada bisnis terus-menerus ditekankan.		√
	3. Personal IT mempunyai keahlian dan skill yang perlu untuk membangun dan menjaga arsitektur informasi yang kuat dan responsif yang mencerminkan kebutuhan semua bisnis.	√	
	4. Informasi diberikan oleh arsitektur informasi diterapkan dengan tetap dan luas.	√	
	5. Penggunaan luas dilakukan dari praktek industri dalam pengembangan dan perawatan arsitektur informasi mencakup proses kemajuan berlanjut.	√	
	6. Strategi untuk mengungkit informasi melalui data pergudangan dan teknologi <i>data mining</i> yang baik.	√	

7. Arsitektur informasi sedang meningkat terus-menerus dan mempertimbangkan dengan seksama informasi non-tradisional pada proses-proses, organisasi-organisasi, dan sistem-sistem.	√	
--	---	--

PO 3 - Menentukan Arah Teknologi

Manajemen proses mendefinisikan arah teknologi yang memenuhi kebutuhan bisnis TI menjadi stabil dan terintegrasi hemat biaya dan sistem biaya dan sistem aplikasi standart, sumber daya, dan kemampuan yang diperlukan bisnis sekarang dan yang akan datang adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Tidak ada kesadaran pentingnya perencanaan insfrastruktur teknologi yang sungguh-sungguh ada.		√
	2. Pengetahuan dan keahlian perlu untuk mengembangkan sebagaimana perencanaan teknologi yang tidak ada.	√	
	3. Ada jarak pemahaman bahwa perencanaan bagi perubahan teknologi adalah penting untuk menyediakan sumber daya dengan efektif.		√
1 <i>Initial/Ad- hoc</i>	1. Manajemen mengenali kebutuhan bagi perencanaan insfrastruktur teknologi.	√	
	2. Pengembangan komponen teknologi dan implementasi teknologi baru adalah khusus dan asing.		√

	3. Ada reaktif dan pendekatan terfokus secara operasional pada perencanaan insfrastruktur.		√
	4. Arah teknologi digerakkan oleh perencanaan evolusi produk sering bertentangan diantara <i>hardware</i> , sistem <i>software</i> , dan vendor <i>software</i> aplikasi.		√
	5. Komunikasi dari dampak potensial perubahan dalam teknologi adalah tidak konsisten.		√
2 <i>Repeatable</i> <i>but</i> <i>Intuitive</i>	1. Kebutuhan untuk dan pentingnya perencanaan teknologi dikomunikasikan.	√	
	2. Perencanaan adalah taktis dan terfokus pada solusi teknis pembangkit pada masalah-masalah teknis, dibanding penggunaan teknologi untuk kebutuhan bisnis.		√
	3. Evaluasi perubahan teknologi ditinggalkan untuk membedakan individu-individu yang mengikuti intuitif serupa proses.		√
	4. Orang-orang mendapat skillnya dalam perencanaan teknologi melalui pembelajaran dan teknik-teknik aplikasi yang diulang.	√	
	5. Teknik dan standar umum muncul untuk pengembangan komponen infrastruktur.	√	
3 <i>Define</i> <i>Process</i>	1. Manajemen sadar akan pentingnya perencanaan insfrastruktur teknologi.	√	
	2. Proses pengembangan perencanaan infrastruktur teknologi layak bunyi dan dilakukan bersama dengan rencana strategi TI.	√	

	3. Perencanaan infrastruktur teknologi yang baik, terdokumentasi, dan berkomunikasi baik, tetapi tidak konsisten diterapkan.		√
	4. Arah infrastruktur teknologi meliputi pemahaman dimana perusahaan ingin untuk mendorong atau meninggalkan penggunaan teknologi, berdasar resiko-resiko dan persekutuan strategi perusahaan.	√	
	5. Vendor utama dipilih berdasar pemahaman teknologi jangka panjang dan rencana pengembangan produk, konsisten dengan arah perusahaan.	√	
	6. Ada komunikasi dan training formal dari peran dan tanggung jawab.	√	
4 <i>Manage and Measurable</i>	1. Manajemen menjamin pengembangan dan perawatan perencanaan infrastruktur teknologi.	√	
	2. Staff TI mempunyai keahlian dan skill penting untuk mengembangkan perencanaan infrastruktur teknologi.	√	
	3. Dampak potensial perubahan dan penemuan teknologi dipertimbangkan.	√	
	4. Manajemen dapat mengenali penyimpangan dari perencanaan dan antisipasi masalah-masalah.	√	
	5. Tanggung jawab untuk pengembangan dan perawatan perencanaan infrastruktur teknologi dikerjakan.	√	

	6. Proses pengembangan perencanaan infrastruktur teknologi yang canggih dan tanggap terhadap perubahan.	√	
	7. Praktek internal yang baik dikenalkan dalam proses.	√	
	8. Strategi sumber daya manusia berjalan bersama arah teknologi untuk memastikan bahwa staff TI dapat mengelola perubahan teknologi.	√	
	9. Perencanaan migrasi untuk mengenalkan teknologi baru yang baik.	√	
	10. <i>Outsourcing</i> dan kerjasama sedang diungkit untuk mengakses keahlian dan <i>skill</i> penting.		√
	11. Manajemen telah menganalisis penerimaan resiko terkait mendorong atau meninggalkan penggunaan teknologi dalam mengembangkan kesempatan bisnis baru atau efisiensi operasional.	√	
5 <i>Optimised</i>	1. Fungsi penelitian ada untuk maninjau kemunculan dan keterlibatan teknologi dan <i>benchmark</i> perusahaan terhadap norma industri.	√	
	2. Arah perencanaan infrastruktur teknologi diarahkan oleh industri, standar internasional, dan pengembangan, dibanding digerakkan oleh vendor teknologi.	√	
	3. Dampak bisnis potensial dari perubahan teknologi ditinjau kembali pada level manajemen senior.	√	
	4. Ada persetujuan eksekutif formal baru dan arah teknologi.	√	

5. Yang sungguh-sungguh ada mempunyai perencanaan infrastruktur teknologi yang hebat yang mencerminkan kebutuhan bisnis, responsif dan dapat dimodifikasi untuk mencerminkan perubahan-perubahan dalam lingkungan bisnis.	√	
6. Ada proses yang dilakukan dan terus-menerus pada tempatnya untuk meningkatkan perencanaan infrastruktur teknologi.	√	
7. Praktek terbaik industri secara ekstensif digunakan dalam menentukan arah teknis.	√	

PO 4 - Mendefinisikan Proses, Organisasi, dan Hubungan TI

Manajemen proses mendefinisikan proses, organisasi, dan hubungan TI yang memenuhi kebutuhan bisnis TI menjadi kuat dalam menjawab strategi bisnis ketika menyetujui dengan ketentuan penguasaan dan memberikan yang baik dan komponen titik kontak adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Organisasi TI tidak secara efektif terbangun untuk memfokuskan prestasi sasaran bisnis.		√
1 <i>Initial/Ad- hoc</i>	1. Fungsi dan aktivitas TI reaktif dan tidak konsisten diterapkan.		√
	2. TI terlibat dalam proyek bisnis hanya dalam langkah kemudian.		√

	3. Fungsi TI dipertimbangkan sebagai fungsi pendukung, tanpa perspektif seluruh perusahaan.		√
	4. Ada sebuah pemahaman jelas dari kebutuhan untuk organisasi TI, namun demikian peran dan tanggung jawab tidak formal dan tidak dilakukan.	√	
2 <i>Repeatable but Intuitive</i>	1. Fungsi TI dibuat untuk menjawab secara taktis, tetapi dengan tidak konsisten, pada kebutuhan pelanggan dan hubungan vendor.		√
	2. Kebutuhan untuk organisasi yang terstruktur dan manajemen vendor dikomunikasikan, tetapi keputusan masih bergantung pada pengetahuan dan skill utama individu.		√
	3. Kemunculan teknis-teknis umum untuk mengelola organisasi TI dan hubungan vendor.	√	
3 <i>Define Process</i>	1. Peran dan tanggung jawab yang baik bagi organisasi TI dan pihak ketiga yang ada.	√	
	2. Organisasi TI dikembangkan, didokumentasi, an dikomunikasikan bersama dengan strategi TI.	√	
	3. Lingkungan kontrol internal baik adanya.	√	
	4. Ada formulasi hubungan dengan kelompok-kelompok lain, meliputi <i>steering committee/SC</i> (komisi pengendali), internal audit, an manajemen vendor.	√	
	5. Organisasi TI secara lengkap sempurna.		√
	6. Ada definisi fungsi yang dilakukan oleh personel TI dan itu oleh <i>user</i> .	√	
	7. Kebutuhan staff TI penting dan keahlian yang baik dan memuaskan.	√	

	8. Ada definisi hubungan formal dengan <i>user</i> dan pihak ketiga.	√	
	9. Divisi peran dan tanggung jawab yang baik dan diterapkan.	√	
4 <i>Manage and Measurable</i>	1. Organisasi TI dengan proaktif menjawab untuk merubah dan memasukkan semua peran penting untuk mengejar kebutuhan bisnis.	√	
	2. Manajemen TI, kepemilikan proses, akuntabilitas, dan tanggung jawab jelas dan berimbang.	√	
	3. Praktek internal yang baik telah diterapkan dalam perusahaan dalam fungsi TI.	√	
	4. Manajemen TI disesuaikan dengan keahlian dan <i>skill</i> untuk menetapkan, menerapkan, dan mengawasi hubungan dan perusahaan yang disukai.	√	
	5. Matrik terukur untuk mendukung sasaran-sasaran bisnis dan faktor-faktor keberhasilan penting user dibakukan.		√
	6. Inventarisasi <i>skill</i> tersedia untuk mendukung proyek pengembangan profesional dan staff.	√	
	7. Keseimbangan antara <i>skill</i> dan sumber daya tersedia secara internal dan itu diperlukan dari perusahaan luar yang baik dan dilakukan.	√	
	8. Struktur organisasi TI mencerminkan kebutuhan bisnis dengan memberikan layanan bersama dengan proses bisnis yang strategis, bukannya dengan teknologi-teknologi asing.	√	
5	1. Struktur organisasi TI fleksibel dan adaktif.	√	

<i>Optimised</i>	2. Praktek terbaik industri disebarakan.	√	
	3. Ada penggunaan teknologi yang luas untuk membantu dalam mengawasi kinerja proses-proses dan organisasi TI.	√	
	4. Teknologi diungkit sejalan untuk mendukung kompleksitas dan distribusi geografis organisasi.	√	
	5. Ada proses kemajuan berlanjut pada tempatnya.	√	

PO 5 - Mengelola Investasi TI

Penerapan TI di perusahaan harus disertai dengan evaluasi/penilaian pembiayaan dan keuntungan yang menyertainya.

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Tidak ada kekhawatiran terhadap kepentingan pemilihan investasi TI dan besaran dana.		√
	2. Tidak ada pelacakan atau pengawasan terhadap investasi TI.		√
1 <i>Initial/Ad- hoc</i>	1. Organisasi mengenali/mengakui kebutuhan pengaturan investasi TI tetapi kebutuhan ini masih terkomunikasi secara tidak konsisten.	√	
	2. Alokasi tanggung jawab pemilihan investasi TI dan besaran dana pembangunan telah dikerjakan oleh sebuah basis <i>ad hoc</i> .		√
	3. Implementasi pemilihan TI dan pendanaan yang terisolasi muncul dengan dokumentasi tidak resmi.		√
	4. Investasi TI dihakimi pada sebuah proses <i>ad hoc</i> .		√

	5. Muncul keputusan yang relatif dan secara operasional terfokus pada pendanaan.	√	
2 <i>Repeatable but Intuitive</i>	1. Terdapat sebuah sedikit pengertian yang implisit terhadap kebutuhan pemilihan investsi TI dan pendanaan.		√
	2. Kebutuhan untuk yang terpilih dan pendanaannya telah terkomunikasi.	√	
	3. Perbedaannya tergantung pada inisiatif perseorangan dalam organisasi.	√	
	4. Terdapat sebuah teknik umum yang darurat untuk membangun komponen TI.		√
	5. Muncul keputusan reaktif dan taktik pendanaan.	√	
3 <i>Define Process</i>	1. Kebijakan dan proses untuk investasi dan pendanaan terdeskripsi, terdokumentasi dan dikomunikasikan, dan menutupi kunci bisnis dan isu/berita teknologi.	√	
	2. Pendanaan TI selaras dengan strategi TI dan rencana bisnis.	√	
	3. Proses pendanaan dan pemilihan TI terformula, didokumentasikan dan dikomunikasikan.	√	
	4. <i>Training</i> formal adalah darurat namun merupakan kebutuhan dasar tiap iniatif individu.	√	
	5. Pendekatan formal dari pemilihan investasi TI dan pendanaan menjadi faktor utama.	√	
	6. Staff TI mempunyai pengalaman dan keterampilan yang dibutuhkan untuk membuat pendanaan TI.	√	

	7. Merekomendasikan investasi TI yang sesuai.	√	
4 <i>Manage and Measurable</i>	1. Tanggung jawab dan akuntabilitas untuk pemilihan investasi TI dan pendanaan ditugaskan kepada individu tertentu.		√
	2. Variasi pendanaan teridentifikasi dan <i>resolved</i> .	√	
	3. Analisa pengeluaran formal diketahui meliputi pengeluaran langsung dan tidak langsung dari operasi yang ada termasuk investasi yang dituju, dan juga mempertimbangkan seluruh pengeluaran daur hidup.		√
	4. Digunakan proses proaktif dan standar pendanaan.	√	
	5. Sumber daya manusia bidang TI diakui pada rencana investasi.	√	
	6. Keuntungan dan pengembalian dihitung di <i>terms</i> finansial dan non-finansial.		√
5 <i>Optimised</i>	1. Praktek terbaik industri biasanya digunakan sebagai contoh/standar dan pendekatan identifikasi untuk meningkatkan efektifitas investasi analisa pembangunan teknologi digunakan di pemilihan investasi dan proses pendanaan.	√	
	2. Proses manajemen investasi bertambah secara terus berdasar pelajaran yang dipelajari dari analisa performa investasi yang nyata/aktual.	√	
	3. Keputusan-keputusan investais menyertakan kecenderungan kenaikan harga.	√	

4. Alternatif-alternatif pembiayaan di evaluasi dalam konteks organisasi.	√	
5. Analisa biaya jangka panjang dan keuntungan dari semuanya disatukan dengan keputusan organisasi.	√	

PO 6 - Mengkomunikasikan Tujuan Manajemen dan Direksi

Penerapan TI harus didukung oleh kebijakan manajemen perusahaan, dan manajemen harus berperan aktif dalam menjadikan kebijakan terkait TI menjadi kebijakan perusahaan secara umum.

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Manajemen tidak menyediakan/terdapat sebuah informasi kontrol lingkungan yang positif.		√
	2. Tidak ada pengakuan akan kebutuhan ketersediaan satu set kebijaksanaan, prosedur, standarisasi dan proses komplain.		√
1 <i>Initial/Ad- hoc</i>	1. Manajemen berekasi terhadap persyaratan kontrol lingkungan informasi.	√	
	2. Kebijaksanaan, prosedur, dan standarisasi dibangun dan dikomunikasikan kepada sebuah basis Ad Hoc yang didasarkan pada isu berkembang.	√	
	3. Proses pembangunan, komunikasi, dan komplain dilakukan secara informal dan tidak <i>consistent</i> .		√

2	<i>Repeatable but Intuitive</i>	1. Manajemen mempunyai pengertian implisit (tersirat) dari kebutuhan dan persyaratan sebuah kontrol lingkungan informasi yang efektif, tetapi secara praktikal sebagian besar dilakukan secara informal.	√	
		2. Manajemen telah mengkomunikasikan kebutuhan akan kebijaksanaan kontrol, prosedur dan standarisasi, tetapi pembangunan diserahkan kepada manajer secara perseorangan dan area bisnis.	√	
		3. Kualitas dikenali sebagai filosofi yang diinginkan untuk diikuti tetapi kenyataannya diserahkan kepada manajer perseorangan.		√
		4. Training dilakukan kepada perseorangan sebagai dasar persyaratan.	√	
3	<i>Define Process</i>	1. Manajemen membangun, mendokumentasikan sebuah kontrol informasi yang komplit, dan juga membuat manajemen kualitas yang termasuk didalamnya adalah kebijakan-kebijakan, prosedur dan standarisasi.		√
		2. Proses kebijakan yang telah ada di dalam departemen dilakukan secara terstruktur, terawat, dan diketahui seluruh staff prosedur dan standarisasi yang ada dan dapat diterima dan meliputi beberapa isu elemen.	√	
		3. Manajemen telah mengirimkan peringatan pesan keamanan TI yang penting.	√	

	4. Training resmi dapat mensupport kontrol lingkungan informasi tetapi sulit diterima.		√
	5. Selama ada pembangunan kembang api untuk menghormati kebijaksanaan kontrol dan standar, terdapat ketidakkonsistenan pada pengawasan terhadap kebijakan dan standarisasi ini.		√
4 <i>Manage and Measurable</i>	1. Manajemen menerima tanggung jawab untuk kebijakan kontrol komunikasi internal dan mendelegasikan tanggung jawab dan mengalokasikan sumber daya yang cukup untuk mempertahankan lingkungan berada pada jalurnya di perubahan secara signifikan.	√	
	2. Kontrol posesif, lingkungan kontrol informasi, termasuk sebuah komitmen untuk kualitas dan peringatan keamanan TI telah disediakan.	√	
	3. Satu set lengkap kebijaksanaan, prosedur dan sandarisasi telah disiapkan, dirawat dan dikomunikasikan dan membutuhkan praktis internal yang baik.		√
5 <i>Optimised</i>	1. Lingkungan kontrol informasi adalah sejajar dengan strategi manajemen <i>framework</i> dan pandangan dan secara teratur diperhatikan, diupdate dan berkembang secara periodik.	√	
	2. Ahli urusan dalam dan luar ditunjuk untuk meyakinkan bahwa praktik industri terbaik akan segera diadopsi dengan penghormatan untuk mengontrol arah dan teknik komunikasi. Pengawasan, kontrol diri dan pengecekan		√

<i>compliance</i> dilakukan secara teratur oleh organisasi.		
3. Teknologi digunakan untuk mempertahankan kebijaksanaan dan dasar pengetahuan kewaspadaan dan untuk mengoptimalkan komunikasi menggunakan <i>automation</i> kantor dan alat komputer <i>training</i> .	√	

PO 7 - Mengelola SDM TI

Manajemen proses mengelola sumber daya manusia TI yang memenuhi kebutuhan bisnis TI yang kompeten dan memotivasi orang untuk membuat dan menyampaikan layanan TI adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Tidak ada kesadaran bahwa pentingnya persekutuan manajemen sumber daya manusia TI dengan proses perencanaan teknologi bagi perusahaan.		√
	2. Tidak ada orang atau kelompok secara formal bertanggung jawab bagi manajemen sumber daya manusia TI.		√
1 <i>Initial/Ad- hoc</i>	1. Manajemen mengenali kebutuhan bagi manajemen sumber daya manusia TI.	√	
	2. Proses manajemen sumber daya manusia TI reaktif dan informal.	√	

	3. Proses manajemen sumber daya manusia TI secara operasional terfokus menggunakan dan mengelola personel TI.	√	
	4. Kesadaran berkembang mengenai dampak bahwa perubahan teknologi dan bisnis cepat solusi kompleks terus meningkat berakibat pada kebutuhan skill baru dan tingkat kompetensi.	√	
2 <i>Repeatable but Intuitive</i>	1. Ada pendekatan taktis untuk menggunakan dan mengelola sumber daya manusia TI digerakkan oleh kebutuhan proyek tertentu, bukannya oleh keseimbangan pemahaman dari ketersediaan internal dan eksternal staff yang terampil.	√	
	2. <i>Training</i> informal terjadi untuk personel baru, yang lalu menerima <i>training</i> pada dasar yang dibutuhkan.	√	
3 <i>Define Process</i>	1. Ada proses dokumentasi dan baik untuk mengelola sumber daya manusia TI.		√
	2. Perencanaan manajemen sumber daya manusia TI ada.		√
	3. Ada pendekatan strategi untuk menggunakan dan mengelola personel TI.	√	
	4. Perencanaan <i>training</i> formal dirancang untuk mencapai kebutuhan sumber daya manusia TI.	√	
	5. Program pemutaran, dirancang untuk memperluas kemampuan manajemen bisnis yang mantap.		√

4 <i>Manage and Measurable</i>	1. Tanggung jawab pengembangan dan pemeliharaan perencanaan manajemen sumber daya manusia TI telah ditugaskan pada individu dan kelompok tertentu dengan dibuthkan keahlian dan <i>skill</i> yang penting untuk mengembangkan dan memelihara perencanaan.	√	
	2. Proses perkembangan dan pengelolaan perencanaan manajemen sumber daya manusia TI responsif pada perubahan.	√	
	3. Perusahaan telah menstandarisasi ukuran yang memperbolehkan untuk mengenali penyimpangan dari perencanaan manajemen sumber daya manusia TI, dengan penekanan tertentu dan pengelolaan perkembangan personel TI dan mutasi.	√	
	4. Manajemen sumber daya manusia TI proaktif mempertimbangkan arah karir.	√	
5 <i>Optimised</i>	1. Perencanaan manajemen sumber daya manusia TI diperbarui terus-menerus untuk mencapai kebutuhan bisnis.	√	
	2. Manajemen sumber daya manusia TI terintegrasi dengan perencanaan teknologi, menjamin pengembangan optimal, dan penggunaan ketersediaan <i>skill</i> TI.		√
	3. Manajemen sumber daya manusia TI terintegrasi dengan responsif pada kesatuan arah yang strategis.	√	

4. Komponen perencanaan manajemen sumber daya manusia TI konsisten dengan praktek terbaik industri, seperti kompensasi, penilaian prestasi, partisipasi dalam forum industri, transfer pengetahuan, training dan pengawasan.		√
5. Program-program <i>training</i> dikembangkan untuk semua standar teknologi baru dan produk lebih dulupada penyebarannya di perusahaan.	√	

PO 8 - Mengelola Mutu

Manajemen proses mengelola mutu yang memenuhi kebutuhan bisnis untuk TI dengan kemajuan berlanjut dan terukur dari mutu layanan TI yang diberikan adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Perusahaan kekurangan proses perencanaan QMS dan metodologi daur hidup pengembangan sistem (SDLC).	√	
	2. Manajemen senior dan staff TI tidak mengenali bahwa mutu program mutu memang perlu.		√
	3. Proyek dan operasi tidak pernah ditinjau mutunya.		√
1 <i>Initial/Ad- hoc</i>	1. Ada kesadaran manajemen kebutuhan untuk QMS.	√	
	2. QMS digerakkan oleh individu-individu dimana itu terjadi.		√
	3. Manajemen membuat keputusan informal pada mutu.	√	

2 <i>Repeatable but Intuitive</i>	1. Program yang sedang dibangun untuk menetapkan dan mengawasi aktivitas QMS dalam TI.	√	
	2. Aktivitas-aktivitas QMS yang tidak terjadi terfokus pada proyek TI dan inisiatif berorientasi proses, tidak ada proses seluruh perusahaan.	√	
3 <i>Define Process</i>	1. Proses QMS yang baik telah dikomunikasikan oleh manajemen dan mencakup manajemen TI dan <i>end-user</i> .	√	
	2. Program <i>training</i> dan pendidikan muncul untuk mengejar semua level perusahaan tentang mutu.	√	
	3. Harapan mutu dasar menjadi jelas dan terbagi diantara proyek-proyek dan dalam perusahaan TI.	√	
	4. Praktek dan <i>tools</i> sederhana bagi manajemen mutu muncul.	√	
4 <i>Manage and Measurable</i>	1. QMS ditujukan pada semua proses, meliputi proses-prose dengan kepercayaan pada pihak-pihak ketiga.	√	
	2. <i>Knowledge base</i> berstandart sedang dibangun untuk matrik mutu.	√	
	3. Metode analisis biaya/keuntungan digunakan untuk membenarkan inisiatif QMS.		√
	4. <i>Benchmarking</i> terhadap industri dan pesaing muncul.	√	
	5. Program <i>training</i> dan pendidikan telah didirikan untuk mengejar semua level perusahaan <i>training</i> mutu.		√

	6. Praktek dan <i>tools</i> sedang distandarisasi dan analisis sebab utama secara periodik dilakukan.	√	
	7. Survei kepuasan mutu secara konsisten dilakukan.		√
	8. Program berstandar untuk mengukur mutu adalah pada tempatnya dan terstruktur dengan baik.	√	
	9. Manajemen TI sedang memebangun <i>knowledge base</i> untuk matrik mutu.		√
5 <i>Optimised</i>	1. QMS terintegrasi dan dilakukan pada semua aktivitas-aktivitas TI.		√
	2. Proses-proses QMS <i>fleksible</i> dan dapat menyesuaikan diri pada perubahan-perubahan lingkungan TI.	√	
	3. <i>Knowledge base</i> untuk matrik mutu ditingkatkan dengan praktik terbaik eksternal.		√
	4. <i>Benchmarking</i> terhadap standar eksternal secara rutin dilakukan.		√
	5. Survei kepuasan mutu adalah proses terus-menerus dan mendorong analisis sebab utama dan tindakan perbaikan.	√	

PO 9 - Menilai dan Mengelola Resiko-resiko TI

Manajemen proses dari menilai dan mengalola resiko-resiko TI yang memenuhi kebutuhan bisnis TI dari analisis dan komunikasi resiko-resiko TI dan dampak potensialnya pada proses dan tujuan bisnis adalah:

LEVEL	KRITERIA	YA	TIDAK
-------	----------	----	-------

0 <i>Non Existent</i>	1. Penilaian resiko untuk proses dan keputusan bisnis tidak terjadi.		√
	2. Perusahaan tidak mempertimbangkan dampak bisnis terkait dengan celah keamanan dan ketidakpastian pengembangan proyek.		√
	3. Manajemen resiko tidak dikenali terkait dengan perolehan solusi TI dan penyampaian layanan TI.		√
1 <i>Initial/Ad- hoc</i>	1. Resiko-resiko TI dipertimbangkan dalam cara khusus.	√	
	2. Penilaian informal resiko proyek terjadi ditentukan oleh setiap proyek.	√	
	3. Penilaian resiko kadang-kadang dikenali dalam perencanaan proyek tetapi jarang ditugaskan pada manajer tertentu.		√
	4. Resiko-resiko terkait TI tertentu seperti keamanan, ketersediaan, dan integritas adalakangan dipertimbangkan dalam proyek-proyek.	√	
	5. Operasi hari ke hari mempengaruhi resiko-resiko terkait TI jarang dibicarakan pada pertemuan manajemen.		√
	6. Dimana resiko-resiko telah dipertimbangkan, kelonggaran tidak konsisten.		√
	7. Muncul pemahaman bahwa resiko-resiko TI penting dan perlu dipertimbangkan.	√	
2	1. Pendekatan penilaian resiko berkembang dan belum matang ada dua diterapkan pada kebijakan manajer proyek.		√

<i>Repeatable but Intuitive</i>	2. Manajemen resiko selalu pada level tinggi dan diterapkan hanya pada proyek utama atau sebagai jawaban atas masalah.	√	
	3. Proses kelonggaran resiko mulai diterapkan dimana resiko dikenali.	√	
3 <i>Define Process</i>	1. Kebijakan manajemen resiko seluruh perusahaan menetapkan kapan dan bagaimana untuk melakukan penilaian resiko.	√	
	2. Manajemen resiko mengikuti proses yang baik yang terdokumentasi.		√
	3. <i>Training</i> manajemen resiko tersedia untuk semua staff.		√
	4. Keputusan-keputusan untuk mengikuti proses manajemen resiko dan untuk menerima <i>training</i> tertinggal pada keleluasaan individu.	√	
	5. Metodologi untuk penilaian resiko meyakinkan dan bersuara dan memastikan bahwa resiko utama pada bisnis dikenali.		√
	6. Sebuah proses untuk mengurangi resiko-resiko utama selalu diadakan sekali saat resiko dikenali.	√	
	7. Deskripsi-deskripsi pekerjaan mempertimbangkan tanggung jawab manajemen resiko.	√	
4 <i>Manage and Measurable</i>	1. Manajemen dan penilaian resiko adalah prosedur standart.		√
	2. Pengecualian pada proses manajemen resiko dilaporkan pada manajemen TI.	√	

3. Manajemen resiko TI adalah tanggung jawab manajemen senior.		√
4. Resiko dinilai dan dikurangi pada level proyek individu dan juga secara teratur terkait seluruh operasi TI.	√	
5. Manajemen disarankan pada perubahan dalam bisnis dan lingkungan TI yang dengan mantap mempengaruhi resiko terkait TI.	√	
6. Manajemen dapat mengawasi posisi resiko dan membuat keputusan terang mengenai pembukaan dan rela untuk menerima.		√
7. Semua resiko dikenali pemilik yang diangkat dan manajemen senior dan manajemen TI telah menentukan tingkat resiko yang perusahaan akan hadapi.		√
8. Manajemen TI telah mengembangkan ukuran-ukuran standart untuk menilai resiko dan melukiskan rasio resiko/keuntungan.		√
9. Manajemen mengatur untuk proyek manajemen resiko operasional untuk menaksir kembali resiko-resiko secara reguler.	√	
10. Database manajemen resiko dibangun dan bagian dari proses manajemen resiko mulai diotomasikan.		√
11. Manajemen TI mempertimbangkan strategi-strategi kelonggaran resiko.	√	

5 <i>Optimised</i>	1. Manajemen resiko telah mengembangkan pada langkah dimana terstruktur, proses seluruh perusahaan dilakukan dan dikelola dengan baik.		√
	2. Praktek-praktek yang baik ditetapkan seluruh perusahaan.	√	
	3. Penangkapan , analisis, dan pelaporan data yang manajemen resiko sangat otomatis.		√
	4. Petunjuk ditarik dari pimpinan-pimpinan dalam bidang itu dan perusahaan TI mengambil bagian dalam kelompok sejawat untuk bertukar pikiran.	√	
	5. Manajemen resiko adalah sungguh terintegrasi dalam semua bisnis dan operasi TI diterima dengan baik, secara ekstensif mencakup <i>user</i> dari layanan TI.		√
	6. Manajamen akan menemukan dan bertindak ketika operasional TI utama dan keputusan investasi dibuat tanpa pertimbangan perencanaan manajemen resiko.		√
	7. Manajemen terus-menerus menaksir strategi-strategi kelonggaran resiko.		√

PO 10 - Mengelola Proyek-proyek

Manajemen proses mengelola proyek-proyek yang memenuhi kebutuhan bisnis TI dari penyampaian hasil proyek di dalam kerangka waktu yang disetujui, biaya, dan mutu adalah:

LEVEL	KRITERIA	YA	TIDAK
-------	----------	----	-------

0 <i>Non Existent</i>	1. Teknik-teknik manajemen proyek tidak digunakan dan perusahaan tidak mempertimbangkan dampak bisnis terkait salah urus proyek dan kegagalan pengembangan proyek.		√
1 <i>Initial/Ad- hoc</i>	1. Penggunaan teknik dan pendekatan manajemen proyek di dalam TI meninggalkan keputusan pada individu manajer.		√
	2. Ada jarak komitmen pada kepemilikan proyek dan manajemen proyek.		√
	3. Keputusan penting pada manajemen proyek dibuat tanpa manajemen <i>user</i> atau input pelanggan.		√
	4. Sedikit atau tidak ada pelanggan dan keterlibatan <i>user</i> dalam menggambarkan proyek-proyek TI.		√
	5. Tidak ada kejelasan perusahaan di dalam TI untuk manajemen proyek.		√
	6. Peran dan tanggung jawab untuk manajemen proyek tidak jelas.		√
	7. Proyek-proyek, rencana-rencana, dan kejadian penting kurang jelas.		√
	8. Waktu dan biaya pegawai proyek tidak sejalan dan dibanding anggaran-anggaran.		√
2 <i>Repeatable but Intuitive</i>	1. Manajemen senior telah memperoleh dan berkomunikasi kesadaran kebutuhan untuk manajemen proyek TI.	√	
	2. Perusahaan adalah dalam proses pengembangan dan pemanfaatan beberapa teknik dan metode dari proyek ke proyek.	√	

	3. Proyek TI secara informal menentukan bisnis dan sasaran teknis.		√
	4. Dibatasi keterlibatan pemegang saham dalam manajemen proyek TI.		√
	5. Petunjuk awal telah dikembangkan untuk beberapa aspek dari manajemen proyek.	√	
	6. Aplikasi petunjuk manajemen proyek diserahkan pada kebijaksanaan manajer proyek individu.		√
3 <i>Define Process</i>	1. Metodologi dan proses manajemen proyek TI telah dibangun dan dikomunikasikan.		√
	2. Proyek-proyek TI ditentukan dengan bisnis sesuai dengan sasaran teknis.	√	
	3. Manajemen bisnis dan senior TI mulai terikat dan terlibat dalam manajemen proyek-proyek TI.	√	
	4. Kantor manajemen proyek dibangun di dalam TI, dengan peran dan tanggung jawab awal tertentu.		√
	5. Proyek-proyek TI diawasi dengan baik dan memperbarui kejadian penting, rencana, biaya, dan ukuran kinerja.	√	
	6. <i>Training</i> manajemen proyek adalah hasil utama dari inisiatif staff individu.		√
	7. Prosedur jaminan mutu dan aktivitas penerapan pusat sistem telah ditentukan tetapi tidak luas diterapkan oleh manajer-manajer TI.	√	
	8. Proyek mulai dikelola seperti portofolio.		√
4	1. Manajemen memerlukan matrik proyek standart dan formal dan pelajaran yang ditinjau mengikuti penyelesaian proyek.		√

<i>Manage and Measurable</i>	2. Manajemen proyek diukur dan dievaluasi seluruh perusahaan tidak hanya di dalam TI.	√	
	3. Kemajuan pada proses manajemen proyek dirumuskan dan komunikasikan dengan anggota tim proyek mengarah ke kemajuan.	√	
	4. Manajemen TI telah diterapkan struktur organisasi proyek dengan peran dan tanggung jawab terdokumentasi dan kriteria kinerja staff.	√	
	5. Nilai dan resiko diukur dan dikelola lebih dulu, selama dan setelah penyelesaian proyek.	√	
	6. Proyek-proyek terus meningkat menunjukkan sasaran perusahaan daripada hanya satu TI tertentu.	√	
	7. Dukungan proyek kuat dan aktif dari sponsor manajemen senior maupun pemegang saham.	√	
	8. <i>Training</i> manajemen proyek yang relevan direncanakan untuk staff dalam kantor manajemen proyek dan seluruh pihak TI.	√	
	5 <i>Optimised</i>	1. Metodologi program dan proyek daur hidup penuh terbukti diterapkan, dilakukan, diintegrasikan dalam budaya seluruh perusahaan.	
2. Sebuah inisiatif terus-menerus untuk mengenali dan melembagakan praktek manajemen proyek terbaik yang diterapkan.		√	
3. Strategi TI untuk pengembangan sumber daya dan proyek operasional ditetapkan dan diterapkan.		√	

4. Kantor manajemen proyek terintegrasi bertanggung jawab untuk proyek-proyek dan program-program dari permulaan ke penerapan selanjutnya.		√
5. Perencanaan seluruh perusahaan dari program-program dan proyek-proyek memastikan bahwa <i>user</i> dan sumber daya TI dimanfaatkan terbaik untuk mendukung inisiatif-inisiatif yang strategis.	√	

Acquire and Implement (AI)

AI 1 - Identifikasi Solusi yang Otomatis

Manajemen proses identifikasi solusi yang otomatis yang memenuhi kebutuhan bisnis TI dalam mewujudkan fungsional bisnis dan kebutuhan kontrol dalam rancangan yang efektif dan efisien dari solusi yang otomatis adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Perusahaan tidak memerlukan identifikasi fungsional dan persyaratan operasional untuk pengembangan, implementasi atau modifikasi solusi, serti sistem, layanan, infrastruktur, <i>software</i> , dan data.		√
	2. Perusahaan tidak menjaga kesadaran ketersediaan solusi teknologi secara potensial sesuai pada bisnisnya.		√

1 <i>Initial/Ad-hoc</i>	1. Ada kesadaran kebutuhan menetapkan kebutuhan dan mengenali solusi teknologi.	√	
	2. Kelompok individu mendiskusikan kebutuhan secara tidak normal dan kebutuhan kadang-kadang didokumentasikan.	√	
	3. Solusi dikenali oleh individu-individu berdasar kesadaran pasar terbatas atau sebagai jawaban atau sebagai vendor.		√
	4. Ada sedikit penelitian terstruktur atau analisis ketersediaan teknologi.		√
2 <i>Repeatable but Intuitive</i>	1. Beberapa pendekatan intuitif mengenali keberadaan solusi TI dan bermacam-macam bisnis.	√	
	2. Solusi dikenali secara informal berdasar pengalaman dan pengetahuan internal pihak TI.		√
	3. Kesuksesan setiap proyek bergantung keahlian dari beberapa individu utama.		√
	4. Mutu dokumentasi dan pembuat keputusan dengan sangat berbeda.		√
	5. Pendekatan terstruktur digunakan untuk menetapkan kebutuhan dan mengenali solusi teknologi.	√	
3 <i>Define Process</i>	1. Pendekatan terstruktur dan jelas dalam menentukan solusi TI yang ada.	√	
	2. Pendekatan pada ketetapan solusi TI mensyaratkan pertimbangan alternatif pendidikan terhadap bisnis atau kebutuhan <i>user</i> , kesempatan	√	

	teknologi, kebutuhan ekonomi, penaksiran resiko, dan faktor lain.		
	3. Proses untuk menemukan solusi TI diterapkan bagi beberapa proyek berdasar faktor-faktor seperti keputusan yang dibuat oleh ketertiban staff individu, sejumlah keterikatan waktu manajemen, dan ukuran dalam prioritas keaslian kebutuhan bisnis.	√	
	4. Pendekatan terstruktur digunakan untuk menentukan kebutuhan-kebutuhan dan identifikasi solusi TI.	√	
4 <i>Manage and Measurable</i>	1. Metodologi kuat untuk identifikasi dan penilaian solusi TI yang ada dan digunakan sebagian besar proyek.		√
	2. Dokumentasi proyek dalam mutu yang baik dan setiap langkah layak disetujui.		√
	3. Kebutuhan-kebutuhan diartikulasikan dan sesuai dengan struktur yang sudah dikenal.		√
	4. Alternatif-alternatif solusi dipertimbangkan, meliputi analisis biaya dan keuntungan.	√	
	5. Metodologi jelas, tegas, umumnya dipahami, dan terukur.		√
	6. Ada <i>interface</i> jelas antara manajemen TI dengan bisnis dalam identifikasi dan penilaian solusi TI.		√

5 <i>Optimised</i>	1. Metodologi yang mantap untuk identifikasi dan penilaian solusi TI dilakukan untuk kemajuan seterusnya.		√
	2. Metodologi implementasi dan perolehan fleksibel untuk proyek skala besar dan kecil.		√
	3. Metodologi didukung oleh database dan <i>konowledge</i> internal dan eksternal berisi bahan solusi teknologi.	√	
	4. Metodologi itu sendiri menghasilkan dokumentasi dalam struktur yang sudah dikenal yang membuat produksi dan penawaran efisien.		√
	5. Kesempatan baru sering diidentifikasi untuk memanfaatkan teknologi untuk memperoleh keuntungan kompetitif, rekayasa ulang proses bisnis pengaruh dan memperbaiki seluruh efisiensi.	√	
	6. Manajemen akan menemukan dan bertindak jika solusi TI disetujui tanpa pertimbangan alternatif teknologi atau kebutuhan fungsional bisnis.		√

AI 2 - Memperoleh dan Merawat Aplikasi *Software*

Manajemen proses memperoleh dan merawat aplikasi *software* yang memenuhi kebutuhan bisnis TI dengan membuat aplikasi tersedia dengan kebutuhan bisnis dan dengan pada waktunya dan biaya yang pantas adalah:

LEVEL	KRITERIA	YA	TIDAK
0	1. Tidak ada proses untuk merancang dan menetapkan aplikasi.		√

<i>Non Existent</i>	2. Aplikasi diperoleh berdasarkan tawaran gerakan vendor, pengakuan <i>brand</i> atau keakraban staff dengan produk-produk tertentu, dengan sedikit atau tanpa mempertimbangkan kebutuhan sebenarnya.		√
1 <i>Initial/Ad-hoc</i>	1. Ada kesadaran bahwa proses untuk perolehan dan perawatan aplikasi diperlukan.	√	
	2. Pendekatan untuk memperoleh dan merawat aplikasi <i>software</i> berbeda-beda dari proyek ke proyek.		√
	3. Keragaman solusi individual untuk kebutuhan bisnis tertentu mungkin diperlukan dengan bebas, sebagai hasil efisiensi dengan perawatan dan dukungan.	√	
	4. Ada sedikit pertimbangan kemauan aplikasi dan ketersediaan dalam rancangan atau pemerolehan aplikasi <i>software</i> .		√
2 <i>Repeatable but Intuitive</i>	1. Ada perbedaan tapi sama proses pada perolehan dan perawatan aplikasi berdasar keahlian di dalam pihak TI.	√	
	2. Rasio keberhasilan dengan aplikasi bergantung besar pada <i>skill</i> dan pengalaman yang dipunyai di dalam TI.	√	
	3. Perawatan selalu problematik dan menderitakan ketika <i>knowledge</i> internal telah hilang dari perusahaan.		√
	4. Ada sedikit perkembangan keamanan aplikasi dan ketersediaan dalam rancangan atau perolehan aplikasi <i>software</i> .		√

3 <i>Define Process</i>	1. Proses pemahaman umumnya baik dan jelas ada dalam perolehan dan perawatan aplikasi <i>software</i> .	√	
	2. Proses ini bersama dengan TI dan strategi bisnis.	√	
	3. Usaha dibuat untuk menerapkan proses terdokumentasi secara konsisten di seluruh aplikasi dan proyek berbeda.		√
	4. Metodologi-metodologi umumnya tidak fleksibel dan sulit diterapkan dalam semua keadaan, jadi langkah yang mungkin adalah dengan melewatinya.		√
	5. Aktivitas perawatan direncanakan, dijadwal, dan dikoordinasikan.		√
4 <i>Manage and Measurable</i>	1. Ada metodologi dipahami dengan baik dan formal yang mencakup rancangan dan proses spesifikasi, kriteria untuk memperolehnya, proses untuk testing dan kebutuhan untuk dokumentasi.	√	
	2. Mekanisme persetujuan disetujui dan terdokumentasi ada untuk memastikan bahwa semua langkah diikuti dan pengecualian sah.		√
	3. Prosedur dan praktik telah berkembang cocok untuk perusahaan, digunakan oleh semua staff dan dapat diterapkan pada sebagian besar kebutuhan aplikasi.	√	
5 <i>Optimised</i>	1. Pratek perolehan dan perawatan <i>software</i> aplikasi bersama dengan proses-proses yang baik.	√	
	2. Pendekatan berdasar komponen yang sudah dikenal, sesuai aplikasi berstandar pada kebutuhan bisnis.	√	

3. Pendekatan adalah seluruh perusahaan.	√	
4. Metodologi perawatan dan perolehan maju dengan baik dan memungkinkan penyebaran cepat, mengijinkan kemampuan aksi dan fleksibilitas dalam menjawab perubahan kebutuhan bisnis.	√	
5. Metodologi implementasi dan perolehan aplikasi <i>software</i> telah diarahkan pada kemajuan berkelanjutan dan didukung oleh database <i>konowledge</i> internal dan eksternal yang berisi bahan dan praktek terbaik.	√	
6. Metodologi ini membuat dokumentasi dalam struktur yang sudah dikenal yang membuat produksi dan perawatan efisien.		√

AI 3 - Memperoleh dan Merawat Infrastruktur Teknologi

Manajemen proses memperoleh dan merawat infratruktur teknologi yang memenuhi kebutuhan bisnis TI dalam memperoleh dan merawat infrastruktur TI yang terintegrasi dan bersandart adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Mengelola infrastruktur teknologi tidak dikenal sebagai topik penting untuk dikejar.		√
1	1. Ada perubahan yang dibuat pada infrastruktur bagi setiap aplikasi baru, tanpa ada semua perencanaan.		√

<i>Initial/Ad-hoc</i>	2. Meskipun ada kesadaran bahwa infrastruktur TI penting, tidak ada pendekatan yang konsisten.		√
	3. Aktivitas perawatan memberi aksi pada kebutuhan jangka pendek.		√
	4. Lingkungan produksi adalah lingkungan uji.		√
2 <i>Repeatable but Intuitive</i>	1. Ada konsistensi antara pendekatan taktis ketika memperoleh dan merawat infrastruktur TI.	√	
	2. Peroleh dan perawatan infrastruktur TI tidak berdasar pada strategi yang baik dan tidak memprioritaskan kebutuhan aplikasi bisnis yang harus didukung.		√
	3. Ada pemahaman bahwa infrastruktur TI penting, didukung oleh beberapa praktek formal.	√	
	4. Perawatan dijadwal tetapi tidak terjadwal dan terkoordinasi penuh, misal beberapa lingkungan, lingkungan uji terpisah ada.	√	
3 <i>Define Process</i>	1. Proses pemahaman baik dan jelas ada untuk memperoleh dan merawat infrastruktur TI.	√	
	2. Kebutuhan dukungan proses dari aplikasi bisnis penting dan bersama dengan TI dan strategi bisnis tidak konsisten diterapkan.	√	
	3. Perawatan direncanakan terjadwal dan terkoordinasi.		√
	4. Ada lingkungan terpisah untuk uji dan produksi.	√	
4	1. Proses perolehan dan perawatan bagi infrastruktur teknologi telah berkembang pada poin dimana bekerja dengan baik pada semua situasi,	√	

<i>Manage and Measurable</i>	diperbolehkan secara konsisten dan terfokus pada <i>reusability</i> .		
	2. infrastruktur TI cukup mendukung aplikasi bisnis.	√	
	3. Proses proaktif dan terorganisir dengan baik.	√	
	4. Biaya dan waktu untuk mencapai tingkat skalabilitas, fleksibility, dan integrasi yang diharapkan secara parsial optimal.	√	
5 <i>Optimised</i>	1. Proses perolehan dan perawatan untuk infrastruktur teknologi proaktif dan bersama dengan aplikasi bisnis penting dan arsitektur teknologi.	√	
	2. Praktek yang baik mengenai solusi teknologi dibolehkan dan perusahaan sadar pengembangan <i>platform</i> terbaru dan manajemen <i>tools</i> . Biaya dikurangi dengan rasionalisasi dan standarisasi komponen infrastruktur dan dengan menggunakan automasi.	√	
	3. Kesadaran teknis tingkat tinggi dapat mengenali cara optimal untuk meningkatkan kinerja secara proaktif, meliputi pertimbangan pilihan <i>outsourcing</i> .		√
	4. Infrastruktur TI dilihat seperti pemungkin utama untuk mengungkit penggunaan TI.	√	

AI 4 – Memungkinkan Operasi dan Penggunaannya

Manajemen proses memungkinkan operasi dan penggunaannya yang memenuhi kebutuhan bisnis TI dalam memastikan pemenuhan dari *end-user* dengan tawaran layanan dan level layanan, integrasi aplikasi dan solusi teknologi dalam proses bisnis adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Ada proses pada tempatnya dengan mempertimbangkan produksi dari dokumentasi <i>user</i> , manual operasi dan bahan <i>training</i> .	√	
	2. Hanya bahan yang telah ada yang disuplai dengan produk terbeli.	√	
1 <i>Initial/Ad- hoc</i>	1. Ada kesadaran bahwa dokumentasi proses diperlukan.	√	
	2. Dokumentasi adakalanya diproduksi dan didistribusi pada kelompok tertentu.		√
	3. Banyak dokumentasi dan prosedur yang ketinggalan.		√
	4. Bahan <i>training</i> cenderung satu skema dari variabel mutu.		√
	5. Tidak ada integrasi virtual dari prosedur dari seluruh sistem dan unit bisnis yang berbeda.		√
	6. Tidak ada input dari unit bisnis dalam rancangan program-program <i>training</i> .		√
2	1. Pendekatan yang sama digunakan untuk menghasilkan prosedur dan dokumentasi, tetapi tidak berdasar pada pendekatan terstruktur atau <i>framework</i> .		√

<i>Repeatable but Intuitive</i>	2. Tidak ada pendekatan seragam untuk pengembangan prosedur operasi dan <i>user</i> .	√	
	3. Bahan <i>training</i> dihasilkan oleh individual dan kelompok proyek, dan mutu bergantung keterlibatan individu. Prosedur dan mutu dari dukungan <i>user</i> beragam dari buruk ke baik, dengan sangat kecil dan konsisten dan integrasi seluruh perusahaan.	√	
	4. Program <i>training</i> bagi bisnis dan <i>user</i> diberikan atau dimudahkan, tetapi tidak ada perencanaan untuk melicinkan dan menyampaikan <i>training</i> .	√	
3 <i>Define Process</i>	1. Ada kerangka pemahaman yang diterima dan jelas bagi dokumentasi <i>user</i> , manual operasi, dan bahan <i>training</i> .	√	
	2. Prosedur-prosedur disimpan dan dirawat dalam perpustakaan formal dan dapat diakses oleh setiap orang yang perlu tahu.		√
	3. Koreksi pada dokumentasi dan prosedur-prosedurnya dibuat dengan landasan reaktif.		√
	4. Prosedur-prosedur tersedia <i>offline</i> dan dapat diakses dan dirawat dalam keadaan bahaya.		√
	5. Ada proses yang menetapkan prosedur terbaru dan bahan <i>training</i> menjadi jelas sampai saat perubahan proyek.	√	
	6. Meskipun keberadaan pendekatan yang jelas, isi yang sebenarnya beragam, karena tidak ada kontrol untuk melakukan sesuai standart.		√

	7. <i>User</i> secara tidak formal terlibat dalam proses.	√	
	8. <i>Tools</i> yang otomatis secara terus-menerus digunakan dalam angkatan dan distribusi prosedur-prosedur.	√	
	9. <i>Training</i> bisnis dan user terencana dan terjadwal.		√
4 <i>Manage and Measurable</i>	1. Ada kerangka yang baik untuk prosedur perawatan dan bahan <i>training</i> yang mempunyai dukungan manajemen TI.	√	
	2. Pendekatan yang dibuat untuk prosedur perawatan dan manual <i>training</i> .	√	
	3. Melingkupi semua sistem dan unit bisnis, sehingga proses-proses dapat ditinjau dari perspektif bisnis.	√	
	4. Prosedur-prosedur dan bahan <i>training</i> terintegrasi mencakup dan saling ketergantungan dan menghubungkan.	√	
	5. Kontrol ada untuk memastikan bahwa standart terlekat dan prosedur-prosedur dikembangkan dan dirawat bagi semua proses.		√
	6. <i>Feedback user</i> dan bisnis pada dokumentasi dan <i>training</i> dikumpulkan dan dinilai sebagai bagian dari proses kemajuan berkelanjutan.	√	
	7. Dokumentasi dan bahan <i>training</i> bisa dapat diprediksi, tingkat yang baik <i>realibilitas</i> dan ketersediaan.		√

	8. Proses yang muncul untuk menggunakan dokumentasi dan manajemen prosedur yang otomatis diterapkan.		√
	9. Pengembangan prosedur yang otomatis terus-menerus terintegrasi dengan pengembangan sistem aplikasi, memudahkan konsistensi, dan akses <i>user</i> .	√	
	10. <i>Training</i> bisnis dan <i>user</i> responsif pada kebutuhan bisnis.	√	
	11. Manajemen TI sedang mengembangkan matrik bagi pengembangan dan penyampaian dari dokumentasi, bahan <i>training</i> , dan program <i>training</i> .		√
5 <i>Optimised</i>	1. Proses untuk dokumentasi operasional dan <i>user</i> ditingkatkan melalui adopsi metode atau <i>tools</i> baru. Bahan prosedur dan bahan <i>training</i> dilakukan seperti mempertimbangkan <i>knowledge base</i> secara konstan yang dirawat secara elektronik menggunakan manajemen <i>knowledgeterbaru</i> , alir kerja, dan didistribusi teknologi, membuat dapat diakses dan mudah untuk dirawat.		√
	2. Proses perolehan dan perawatan untuk infrastruktur teknologi proaktif dan bersekutu dengan aplikasi bisnis penting dan arsitektur teknologi.		√
	3. Bahan <i>training</i> dan dokumentasi diperbarui untuk mencerminkan organisasional, operasional, dan perubahan <i>software</i> .	√	

4. Pengembangan dokumentasi dan bahan <i>training</i> dan penyampaian program <i>training</i> secara penuh terintegrasi dengan bisnis dan dengan definisi proses bisnis, jadi mendukung kebutuhan seluruh perusahaan, daripada hanya prosedur-prosedur berorientasi TI.		√
---	--	---

AI 5 - Memperoleh Sumber Daya TI

Manajemen proses memperoleh sumber daya TI yang memenuhi kebutuhan bisnis TI dengan meningkatkan hemat biaya TI dan kontribusinya pada keuntungan bisnis adalah:

LEVEL	KRITERIA	YA	TIDAK
0 <i>Non Existent</i>	1. Tidak ada proses perolehan sumber daya TI yang baik pada tempatnya.		√
	2. Perusahaan tidak mengenali kebutuhan bagi kebijakan dan prosedur perolehan yang jelas untuk memastikan bahwa semua sumber daya TI tersedia tepat waktu dan efisiensi biaya.		√
1 <i>Initial/Ad- hoc</i>	1. Perusahaan telah mengenali kebutuhan kebijakan dan prosedur terdokumentasi yang menghubungkan perolehan TI pada proses perolehan seluruh perusahaan bisnis.	√	
	2. Kontrak untuk perolehan sumber daya TI dikembangkan dan dikelola dengan manajer proyek dan individu lainnya dengan melatih keputusan	√	

	profesional mereka bukannya sebagai hasil dari prosedur dan kebijakan formal.		
	3. Ada hubungan khusus antara perolehan perusahaan dan proses manajemen kontrak TI.		√
	4. Kontrak bagi perolehan dikelola pada akhir proyek bukannya terus-menerus.		√
2 <i>Repeatable but Intuitive</i>	1. Ada kesadaran perusahaan atas kebutuhan prosedur dan kebijakan dasar bagi perolehan TI.	√	
	2. Prosedur dan kebijakan terintegrasi dengan proses perolehan organisasi bisnis.	√	
	3. Proses perolehan dimanfaatkan untuk proyek sangat besar.		√
	4. Tanggung jawab dan akuntabilitas bagi perolehan TI dan manajemen kontrak ditentukan oleh pengalaman manajer kontrak individual.		√
	5. Pentingnya manajemen supplier dan hubungan manajer dikenal tetapi ditunjukkan berdasarkan inisiatif individu.	√	
	6. Proses kontrak sangat dimanfaatkan oleh proyek besar.	√	
3 <i>Define Process</i>	1. Manajemen telah mengadakan kebijakan dan prosedur bagi perolehan TI.		√
	2. Kebijakan dan prosedur diserahkan oleh proses perusahaan bisnis.		√

	3. Perolehan TI terintegrasi dengan sistem perolehan bisnis.	√	
	4. Standart TI bagi perolehan sumber daya TI yang ada.	√	
	5. Supplier sumber daya TI terintegrasi dalam mekanisme manajemen proyek perusahaan perspektif manajemen kontrak.		√
	6. Manajemen TI mengkomunikasikan kebutuhan bagi perolehan yang layak dan manajemen proyek seluruh fungsi TI.		√
4 <i>Manage and Measurable</i>	1. Perolehan TI terintegrasi secara penuh dengan sistem perolehan seluruh bisnis.	√	
	2. Standart TI bagi perolehan sumber daya TI digunakan untuk semua perolehan.	√	
	3. Ukuran kontrak dan manajemen perolehan diambil berkait dengan kasus bisnis bagi perolehan TI.	√	
	4. Laporan bahwa dukungan sasaran bisnis tersedia.	√	
	5. Manajemen selalu sadar pengecualian pada kebijakan dan prosedur bagi perolehan TI.	√	
	6. Manajemen yang strategis dari hubungan berkembang.	√	
	7. Manajemen TI melakukan penggunaan perolehan dan proses manajemen kontrak untuk semua perolehan dengan meninjau ukuran kinerja.	√	
5	1. Manajemen telah melakukan dan menghasilkan seluruh proses bagi perolehan TI.	√	

<i>Optimised</i>	2. Manajemen melakukan sesuai dengan kebijakan dan prosedur bagi perolehan TI.	√	
	3. Ukuran pada manajemen perolehan dan kontrak dilakukan berkaitan dengan kasus bisnis untuk perolehan TI.	√	
	4. Hubungan baik terbangun dari waktu ke waktu dengan banyak supplier dan partner dan mutu hubungan terukur dan terawasi.	√	
	5. Hubungan dikelola dengan strategis.	√	
	6. Prosedur, kebijakan, dan standart TI untuk perolehan sumber daya TI dikelola dengan strategis dan bereaksi terhadap ukuran proses.	√	
	7. Manajemen TI mengkomunikasikan pentingnya strategi perolehan yang layak dan manajemen kontrak seluruh fungsi TI.	√	

AI 6 - Mengelola Perubahan-Perubahan

Manajemen proses mengelola perubahan-perubahan yang memenuhi kebutuhan bisnis TI bereaksi terhadap kebutuhan bisnis bersama dengan strategi bisnis, selagi mengurangi solusi dan kerusakan penyampaian layanan da mengerjakan lagi adalah:

LEVEL	KRITERIA	YA	TIDAK
0	1. Tidak ada proses manajemen perubahan yang baik dan perubahan dapat dibuat dengan virtual tanpa kontrol.		√

<i>Non Existent</i>	2. Tidak ada kesadaran bahwa perubahan dapat mengacaukan bagi operasi bisnis dan TI dan tidak ada kesadaran keuntungan dari manajemen perubahan yang baik.		√
1 <i>Initial/Ad-hoc</i>	1. Dikenal bahwa perubahan harus dikelola dan diawasi.	√	
	2. Praktek beragam dan mungkin bahwa perubahan yang tidak sah terjadi.	√	
	3. Ada sedikit atau tidak ada dokumentasi dari perubahan, dan konfigurasi dokumentasi tidak lengkap dan tidak dapat dipercayai.	√	
	4. Kesalahan mungkin terjadi bersama dengan interupsi pada lingkungan produksi disebabkan oleh manajemen perubahan yang kurang bagus.	√	
2 <i>Repeatable but Intuitive</i>	1. Ada proses manajemen perubahan informal pada tempatnya dan perubahan mengikuti pendekatan ini, namun tidak terstruktur, mendasar, cenderung akan error.		√
	2. Akurasi dokkumentasi konfigurasi tidak konsekuen dan hanya perencanaan terbatas dan penilaian dampak terjadi lebih dulu sebelum perubahan.		√
3	1. Ada proses manajemen perubahan formal yang baik pada tempatnnya, mencakup kategorisasi, prioritasisasi, prosedur-prosedur darurat, otorisasi perubahan, dan manajemen pelepasan dan sesuai dengan perkembangan yang cepat.	√	

<i>Define Process</i>	2. <i>Workaround</i> terjadi dan proses-prosesnya sering dilewati.	√	
	3. Error mungkin terjadi dan perubahan tidak sah adakalanya terjadi.	√	
	4. Analisis dampak perubahan TI pada operasi bisnis menjadi terbentuk untuk mendukung perencanaan teknologi dan aplikasi baru.	√	
4 <i>Manage and Measurable</i>	1. Proses manajemen perubahan dikembangkan dengan baik dan secara konsisten diikuti untuk semua perubahan, dan manajemen yakin bahwa ada sedikit pengecualian.	√	
	2. Proses efisien dan efektif tetapi berstandar pada prosedur manual dan kontrol untuk memastikan bahwa mutu dicapai.	√	
	3. Semua perubahan adalah subjek perencanaan seksama dan dampak penilaian untuk memperkecil kemungkinan masalah tempat produksi.	√	
	4. Proses persetujuan untuk perubahan sudah pada tempatnya.	√	
	5. Dokumentasi manajemen perubahan adalah sekarang dan tepat, dengan perubahan, dengan merubah secara formal.		√
	6. Dokumentasi konfigurasi umumnya akurat.		√
	7. Perencanaan manajemen perubahan TI dan implementasi menjadi terintegrasi dengan perubahan dalam proses bisnis, menjamin bahwa training, perubahan organisasi, persoalan-persoalan kelancaran bisnis ditujukan.	√	

	8. Ada koordinasi yang meningkat antara manajemen perubahan TI dan perancangan kembali proses bisnis.	√	
	9. Ada proses konsisten untuk mengawasi mutu dan kinerja proses manajemen perubahan.	√	
5 <i>Optimised</i>	1. Proses manajemen perubahan secara teratur ditinjau dan diperbarui untuk tetap sejalan dengan praktik yang baik.	√	
	2. Proses meninjau mencerminkan hasil monitoring.	√	
	3. Informasi konfigurasi adalah berbasis komputer dan memberikan kontrol terjemahan.	√	
	4. Pelacakan perubahan adalah canggih dan mencakup <i>tools</i> untuk menemukan <i>software</i> yang tidak sah dan tidak berlisensi.		√
	5. Manajemen perubahan TI terintegrasi dengan manajemen perubahan bisnis untuk memastikan bahwa TI adalah pemungkin dalam meningkatkan produktivitas dan menciptakan kesempatan bisnis baru bagi perusahaan.	√	

AI 7 - Memasang dan Mengakui Solusi-Solusi dan Perubahan-Perubahan

Manajemen proses memasang dan mengakui solusi-solusi dan perubahan-perubahan yang memenuhi kebutuhan bisnis TI dengan bekerja sistem baru atau diubah tanpa masalah utama instalasi adalah:

LEVEL	KRITERIA	YA	TIDAK
-------	----------	----	-------

0 <i>Non Existent</i>	1. Ada jarak penuh instalasi formal atau proses pengangkatan dan tidak ada manajemen senior ataupun staff TI mengenali kebutuhan untuk membuktikan bahwa solusi patut untuk tujuan yang dimaksud.		√
1 <i>Initial/Ad- hoc</i>	1. Ada kesadaran kebutuhan untuk membuktikan dan menegaskan bahwa solusi-solusi terimplementasi melayani tujuan yang dimaksud.	√	
	2. Testing dilakukan pada beberapa proyek, tetapi inisiatif untuk testing ditinggal pada tim proyek individu dan pendekatan yang diambil beragam.	√	
	3. Pengangkatan formal dan mengakhiri adalah jarang dan tidak ada.		√
2 <i>Repeatable but Intuitive</i>	1. Ada beberapa konsistensi diantara pendekatan testing dan pengangkatan, tetapi tidak berdasar beberapa metodologi.	√	
	2. Tim pengembangan individu secara normal memutuskan pendekatan testing dan biasanya sebuah kemangkiran dari testing terintegrasi.		√
	3. Ada proses persetujuan informal.	√	
3 <i>Define Process</i>	1. Metodologi formal terkait pada instalasi, migrasi, konversi, dan penerimaan adalah pada tempatnya.	√	
	2. Proses pengangkatan dan instalasi TI terintegrasi dalam daur hidup sistem dan sampai taraf tertentu secara otomatis.	√	
	3. Training, testing, dan transisi pada status produksi dan pengangkatan mungkin untuk	√	

	membedakan dari proses yang baik, berdasar keputusan individu.		
	4. Mutu sistem memasuki produksi adalah tidak konsisten, dengan sistem baru sering membangkitkan level penting dari masalah-masalah implementasi utama.	√	
4 <i>Manage and Measurable</i>	1. Prosedur dirumuskan dan dikembangkan menjadi terorganisir baik dan praktis dengan menentukan prosedur pengangkatan dan lingkungan <i>test</i> yang baik.	√	
	2. Dalam praktik, semua perubahan utama, pada sistem mengikuti pendekatan terumus ini.	√	
	3. Evaluasi pertemuan kebutuhan user berstandart dan terukur, matrik produksi yang secara efektif ditinjau dan dianalisis oleh manajemen.		√
	4. Mutu sistem memasuki produksi memuaskan untuk manajemen, bahkan dengan level yang layak dari masalah implementasi utama.		√
	5. Otomasi proses adalah bergantung proyek dan tertentu.	√	
	6. Manajemen mungkin puas dengan level sekarang dari efisiensi meskipun kurang evaluasi dari implementasi utama.	√	
	7. Sistem <i>test</i> layak mencerminkan lima lingkungan.	√	

	8. Testing tegangan bagi sistem baru dan kemunduran testing bagi keberadaan sistem diterapkan untuk semua proyek.	√	
5 <i>Optimised</i>	1. Proses pengangkatan dan insatalasi telah dibersihkan pada tingkat praktek yang baik, berdasar hasil kemajuan yang berlanjut dan perbaikan.	√	
	2. Proses pengangkatan dan instalasi TI terintegrasi dalam daur hidup dan otomatis ketika layak, memfasilitasi sebagian besar training, testing dan transisi pada status produksi dari sistem baru.	√	
	3. Lingkungan <i>test</i> berkembang baik, daftar masalah dan resolusi kesalahan menjamin transisi efisien dan efektif pada lingkungan produksi.	√	
	4. Proses pengangkatan terjadi selalu tanpa mengolah lagi (<i>rework</i>), dan masalah implementasi utama secara normal terbatas pada koreksi kecil.		√
	5. Tinjauan implementasi utama berstandart, dengan pelajaran tersalur kembali pada proses untuk memastikan kemajuan mutu terus-menerus.	√	
	6. Testing tegangan bagi sistem baru dan testing lemunduran bagi sistem yang termodifikasi secara konsisten diterapkan.	√	

B. Use Case Scenario

B.1 Skenario Use Case Login

Use Case Login menghubungkan dengan 4 aktor, yaitu *admin*, staff TI, manajerial, dan PR 1. Penjelasan urutan aksi aktor dan reaksi sistem pada skenario

normal dan skenario alternatif skenario *use case login* dijelaskan pada Tabel 3 sampai dengan Tabel 6.

Tabel 3 Skenario *Use Case Login Admin*

Nama Use Case	<i>Login</i>
Aktor	<i>Admin</i>
Deskripsi Singkat	<i>Admin akan mengakses aplikasi</i>
Prekondisi	<i>Username dan password yang akan digunakan untuk login</i>
Prakondisi	<i>Berhasil login</i>
Flow Events	
Skenario Normal : <i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	
	2. Menampilkan halaman utama <i>login</i>
3. Klik tombol <i>login</i>	
	4. Menampilkan halaman <i>login</i>
5. Menginputkan <i>username</i> dan password	
6. Klik tombol <i>login</i>	
	7. Mengecek ke database
	8. Menampilkan halaman dashboard
Skenario Alternatif : <i>Username dan Password Salah</i>	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>login</i>	
	7a. Menampilkan span “ <i>username dan password salah</i> ”

Tabel 4 Skenario *Use Case Login* Staff TI

Nama Use Case	<i>Login</i>
Aktor	Staff TI
Deskripsi Singkat	Staff TI akan mengakses aplikasi
Prekondisi	<i>Username</i> dan <i>password</i> yang akan digunakan untuk <i>login</i>
Prakondisi	Berhasil <i>login</i>
Flow Events	
Skenario Normal : <i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	
	2. Menampilkan halaman utama <i>login</i>
3. Klik tombol <i>login</i>	
	4. Menampilkan halaman <i>login</i>
5. Menginputkan <i>username</i> dan <i>password</i>	
6. Klik tombol <i>login</i>	
	7. Mengecek ke database
	8. Menampilkan halaman dashboard
Skenario Alternatif : <i>Username dan Password Salah</i>	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>login</i>	
	7a. Menampilkan span " <i>username dan password salah</i> "

Tabel 4 Skenario *Use Case Login* Manajerial

Nama Use Case	<i>Login</i>
----------------------	--------------

Aktor	Manajerial
Deskripsi Singkat	Manajerial akan mengakses aplikasi
Prekondisi	<i>Username</i> dan <i>password</i> yang akan digunakan untuk <i>login</i>
Prakondisi	Berhasil <i>login</i>
Flow Events	
Skenario Normal : <i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	
	2. Menampilkan halaman utama <i>login</i>
3. Klik tombol <i>login</i>	
	4. Menampilkan halaman <i>login</i>
5. Menginputkan <i>username</i> dan <i>password</i>	
6. Klik tombol <i>login</i>	
	7. Mengecek ke database
	8. Menampilkan halaman dashboard
Skenario Alternatif : <i>Username dan Password Salah</i>	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>login</i>	
	7a. Menampilkan span " <i>username dan password salah</i> "

Tabel 5 Skenario *Use Case Login PR 1*

Nama Use Case	<i>Login</i>
Aktor	PR 1
Deskripsi Singkat	PR 1 akan mengakses aplikasi

Prekondisi	<i>Username dan password yang akan digunakan untuk login</i>
Prakondisi	Berhasil <i>login</i>
Flow Events	
Skenario Normal : <i>Login</i>	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	
	2. Menampilkan halaman utama <i>login</i>
3. Klik tombol <i>login</i>	
	4. Menampilkan halaman <i>login</i>
5. Menginputkan <i>username</i> dan <i>password</i>	
6. Klik tombol <i>login</i>	
	7. Mengecek ke database
	8. Menampilkan halaman dashboard
Skenario Alternatif : <i>Username dan Password Salah</i>	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>login</i>	
	7a. Menampilkan span “ <i>username dan password salah</i> ”

B.2 Skenario *Use Case* Pengelolaan Data *User*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case* pengelolaan data *user* dapat dilihat pada Tabel 7.

Tabel 7 Skenario *Use Case* Pengelolaan Data *User*

Nama <i>Use Case</i>	Pengelolaan Data <i>User</i>
-----------------------------	------------------------------

Aktor	<i>Admin</i>
Deskripsi Singkat	<i>Admin akan mengelola data user</i>
Prekondisi	<i>Data user</i>
Prakondisi	<ol style="list-style-type: none"> 1. <i>Data user</i> berhasil ditambahkan 2. <i>Data user</i> berhasil diubah 3. <i>Data user</i> berhasil dihapus
Flow Events	
Skenario Normal : View User	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>user</i>	
	2. Menampilkan halaman <i>user</i> meliputi tabel data <i>user</i> yang telah diinputkan beserta tombol data baru, edit, dan hapus
Skenario Normal : Tambah User	
Aksi Aktor	Reaksi Sistem
3. Klik tombol data baru	
	4. Menampilkan halaman form tambah <i>user</i> meliputi id <i>user</i> , <i>username</i> , password, bagian, dan level
5. Mengisi form tambah <i>user</i>	
6. Klik tombol simpan	
	7. Menyimpan ke database
	8. Menampilkan halaman <i>user</i>
Skenario Alternatif : Form Data Tidak Lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol simpan	

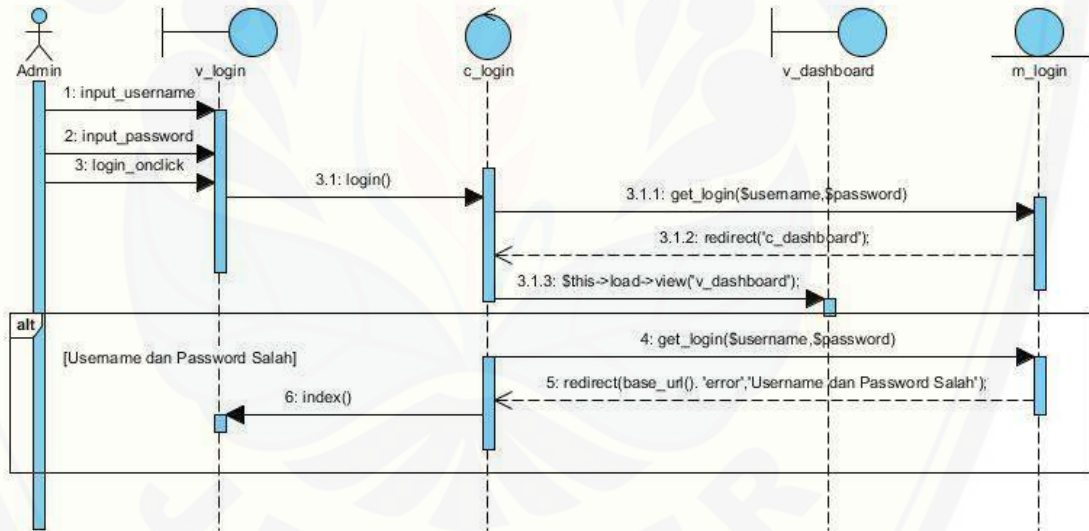
	7a. Menampilkan warning “please fill out this field”
Skenario Alternatif : Memilih Tombol Kembali	
Aksi Aktor	Reaksi Sistem
6b. Klik tombol kembali	
	7b. Menampilkan kembali halaman <i>user</i>
Skenario Normal : Edit User	
Aksi Aktor	Reaksi Sistem
3. Klik tombol edit	
	4. Menampilkan halaman form edit <i>user</i> meliputi id <i>user</i> , <i>username</i> , password, bagian, dan level
5. Mengisi form edit <i>user</i>	
6. Klik tombol simpan	
	7. Menyimpan ke database
	8. Menampilkan halaman <i>user</i>
Skenario Alternatif : Form Data Tidak Lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol simpan	
	7a. Menampilkan warning “please fill out this field”
Skenario Alternatif : Memilih Tombol Kembali	
Aksi Aktor	Reaksi Sistem
6b. Klik tombol kembali	
	7b. Menampilkan kembali halaman <i>user</i>
Skenario Normal : Hapus User	

Aksi Aktor	Reaksi Sistem
3. Klik tombol hapus	
	4. Menghapus data pada database
	5. Menampilkan halaman <i>user</i>

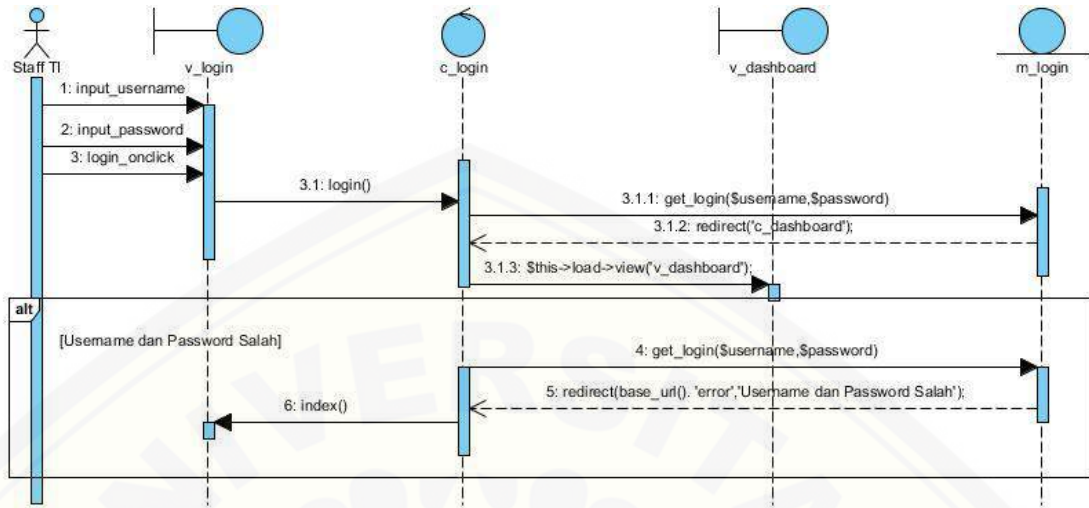
C. Sequence Diagram

C.1. Sequence Diagram Login

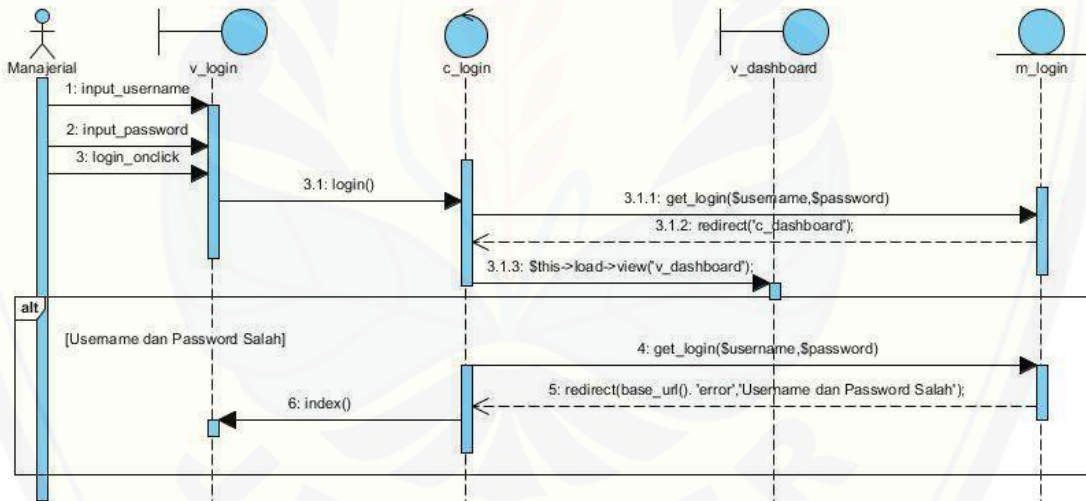
Penggambaran *sequence diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 1 sampai dengan Gambar 4.



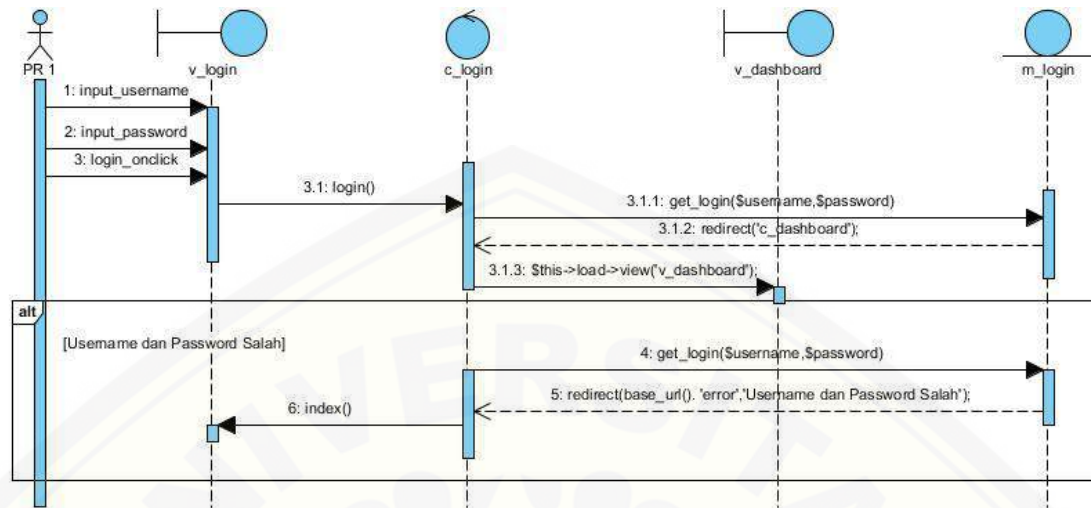
Gambar 1 Sequence Diagram Login Admin



Gambar 2 Sequence Diagram Login Staff TI



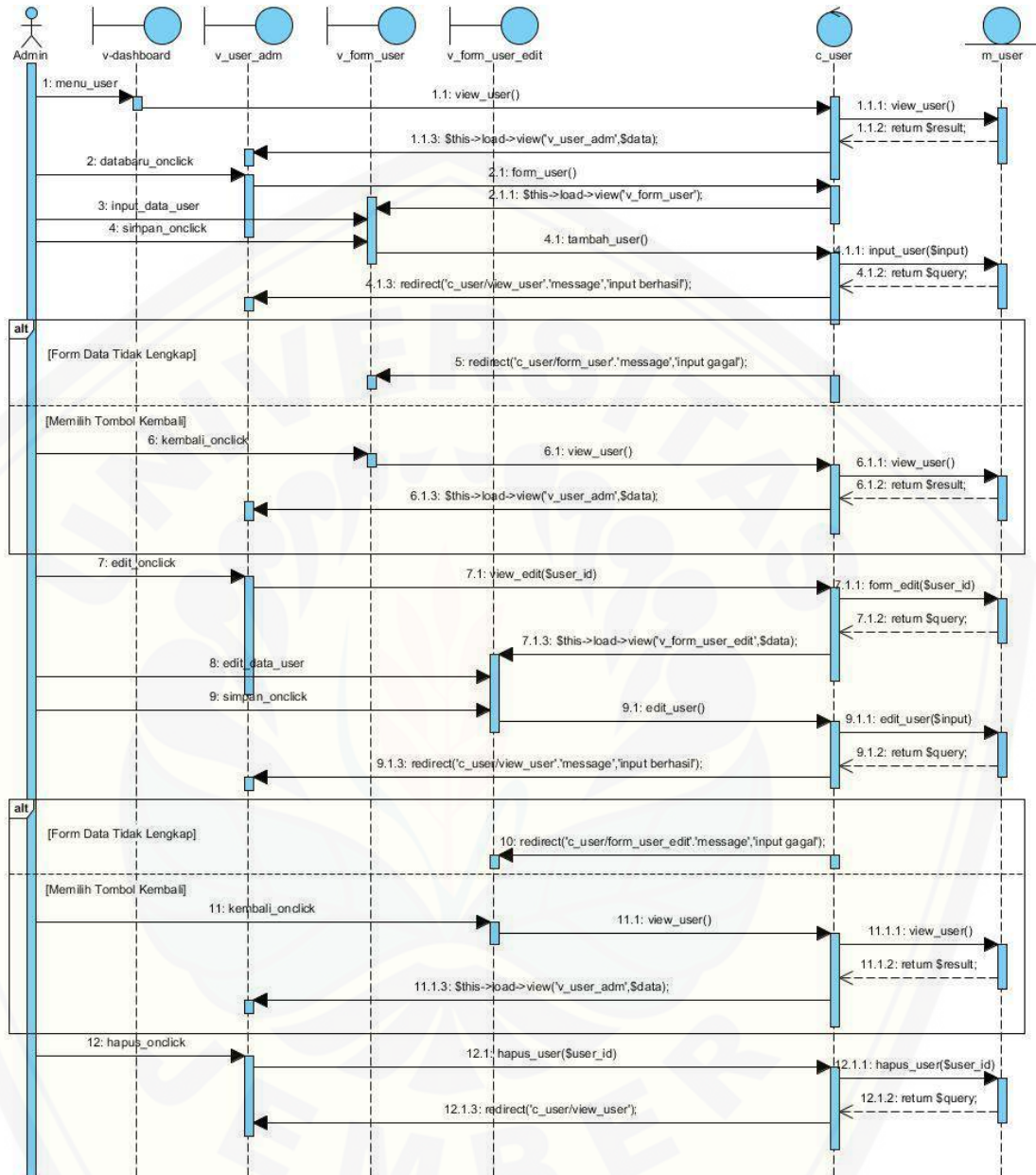
Gambar 3 Sequence Diagram Login Manajerial



Gambar 4 Sequence Diagram Login PR 1

C.2 Sequence Diagram Pengelolaan Data User

Penggambaran *sequence diagram* Pengelolaan Data *user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 5.



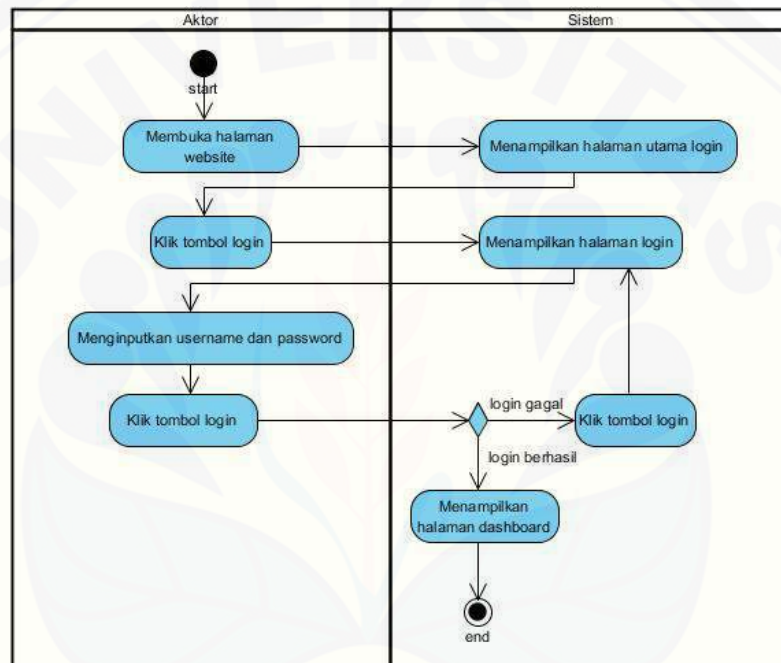
Gambar 5 Sequence Diagram Pengelolaan Data User

D. Activity Diagram

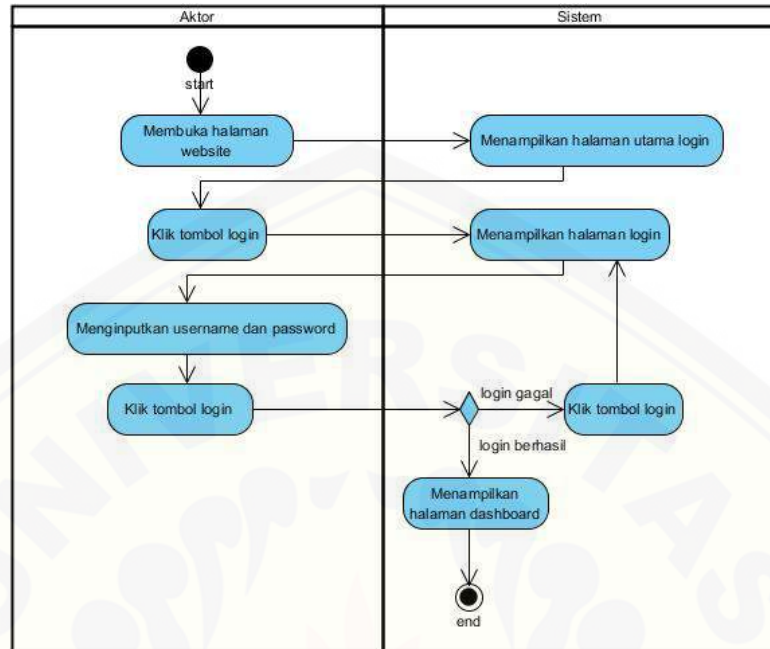
Activity diagram menggambarkan alur aktivitas pada aplikasi penilaian tingkat kematangan tata kelola TI.

D.1 Activity Diagram Login

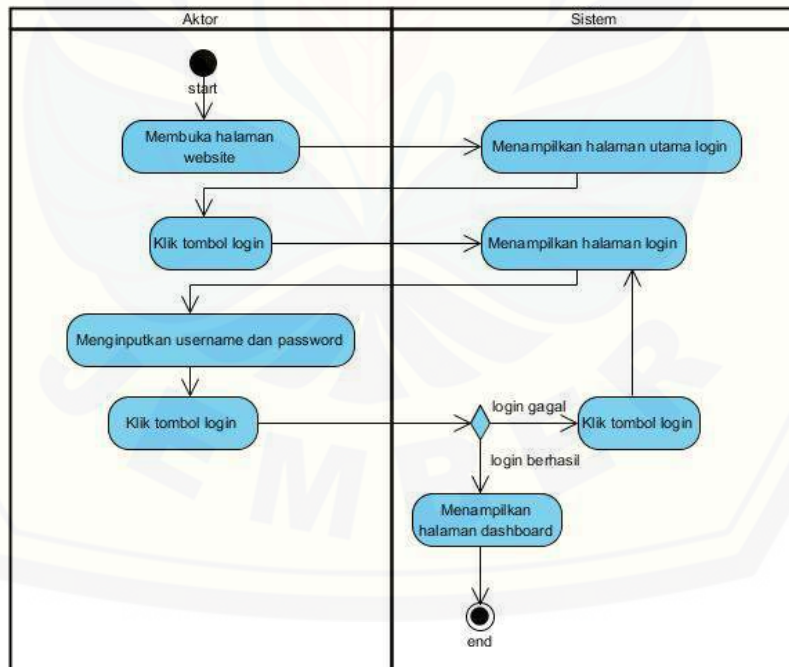
Activity diagram login dapat dilihat pada Gambar 14 sampai dengan Gambar 17.



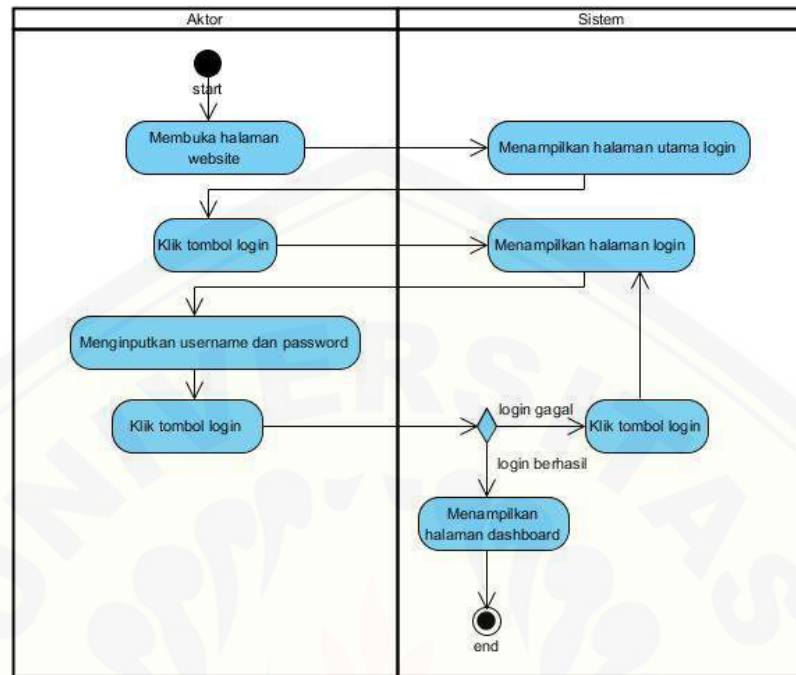
Gambar 14 Activity Diagram Login Admin



Gambar 15 Activity Diagram Login Staff TI



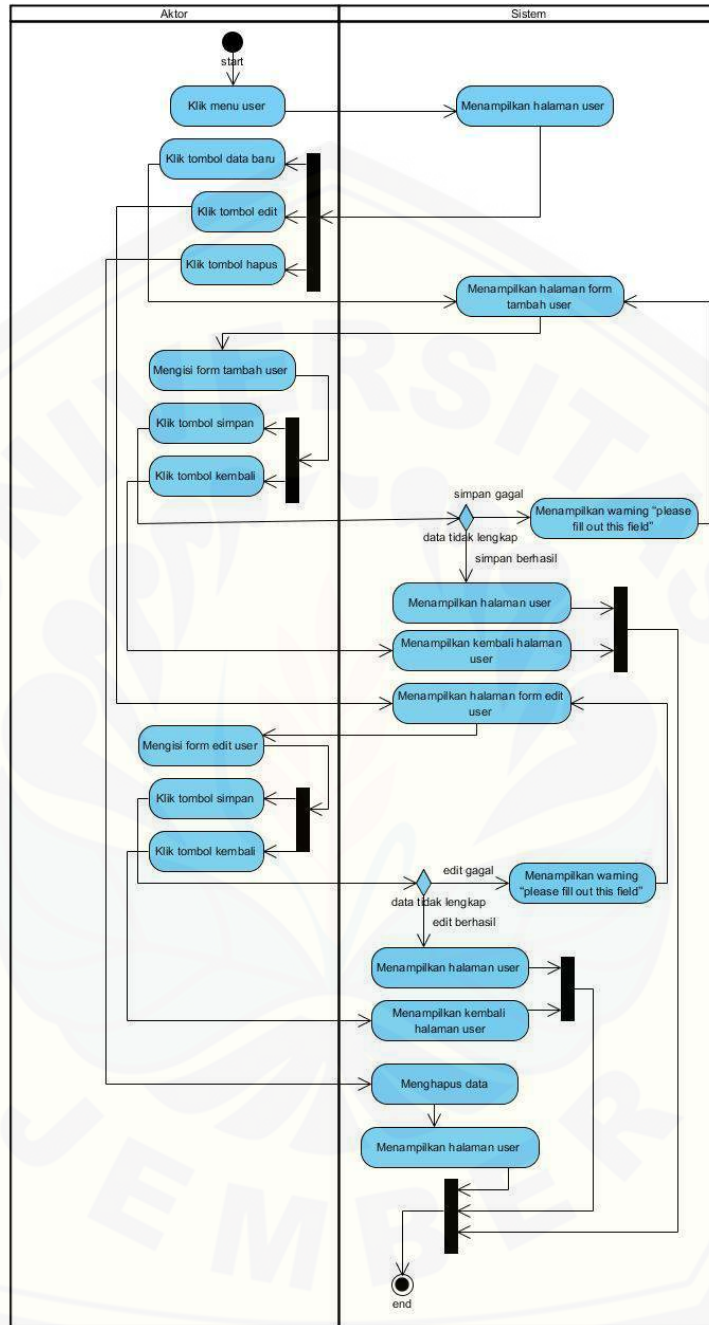
Gambar 16 Activity Diagram Login Manajerial



Gambar 17 Activity Diagram Login PR 1

D.2 Activity Diagram Pengelolaan Data User

Activity diagram pengelolaan data user dapat dilihat pada Gambar 18.



Gambar 18 Activity Diagram Pengelolaan Data User

E. Kode Program

E.1 Kode Program *Login*

Kode program *login* terletak pada kelas *v_login*, *c_login*, dan *m_login*.

1. Kelas *c_login*

Penulisan kode program *c_login* dapat dilihat pada Gambar 27 sampai dengan Gambar 30.

```
// Halaman Login
public function index(){
    $this->load->view('v_login');
}
```

Gambar 27 Kode Program function index

```
// Tombol Login
public function login(){
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $level = $this->m_login->get_login($username,$password);
    if($level > 0){
        redirect('c_dashboard');
    }else{
        $this->session->set_flashdata('error', 'Username dan Password Salah');
        redirect(base_url());
    }
}
```

Gambar 28 Kode Program function login

```
// Error Login
public function error(){
    $this->session->set_flashdata('error', 'Silahkan lakukan login terlebih dahulu');
    redirect(base_url());
}
```

Gambar 29 Kode Program function error

```
// Tombol Logout
public function logout(){
    session_destroy();
    redirect(base_url());
}
```

Gambar 30 Kode Program function logout

2. Kelas m_login

Penulisan kode program c_login dapat dilihat pada Gambar 31.

```
public function get_login($username,$password){
    $query = "SELECT * FROM tb_user WHERE username = '". $username.'" AND password = '". $password.'"";
    $result = $this->db->query($query);
    $level = "";
    foreach($result->result() as $row){
        $bagian = $row -> bagian;
        $level = $row -> level;
        $user_id=$row -> user_id;
    }
    $parameter = array(
        'user_id' => $user_id,
        'username' => $username,
        'bagian' => $bagian,
        'level' => $level
    );
    $this ->session->set_userdata($parameter);
    if($level > 0){
        return $level;
    }else{
        return 0;
    }
}
```

Gambar 31 Kode Program function get_login

E.2 Kode Program Pengelolaan Data *User*

Kode program pengelolaan data *user* terletak pada kelas v_user_adm, v_form_user, v_form_user_edit, c_user, dan m_user.

1. Kelas c_user

Kode program kelas c_user dapat dilihat pada Gambar 32 sampai dengan Gambar 37.

```
public function view_user(){
    $data['user'] = $this->m_user->view_user()->result();
    $this->load->view('header');
    $this->load->view('v_user_adm',$data);
    $this->load->view('footer');
}
```

Gambar 32 Function view_user

```
public function form_user(){
    $this->load->view('header');
    $this->load->view('v_form_user');
    $this->load->view('footer');
}
```

Gambar 33 Function form_user

```
public function tambah_user(){
    $input=array(
        'user_id' => $this->input->post('user_id'),
        'username' => $this->input->post('username'),
        'password' => $this->input->post('password'),
        'bagian' => $this->input->post('bagian'),
        'level' => $this->input->post('level')
    );

    $logic = $this->m_user->input_user($input);
    if($logic=true){
        $this->session->set_flashdata('message','input berhasil');
        redirect('c_user/view_user');
    } else {
        $this->session->set_flashdata('message','input gagal');
        redirect('c_user/form_user');
    }
}
```

Gambar 34 Function tambah_user

```
public function view_edit($user_id){
    $data['user_ed'] = $this->m_user->form_edit($user_id);
    $this->load->view('header');
    $this->load->view('v_form_user_edit',$data);
    $this->load->view('footer');
}
```

Gambar 35 Function view_edit

```

public function edit_user(){
    $input=array(
        'user_id' => $this->input->post('user_id'),
        'username' => $this->input->post('username'),
        'password' => $this->input->post('password'),
        'bagian' => $this->input->post('bagian'),
        'level' => $this->input->post('level')
    );

    $logic = $this->m_user->edit_user($input);
    if($logic=true){
        $this->session->set_flashdata('message','input berhasil');
        redirect('c_user/view_user');
    } else {
        $this->session->set_flashdata('message','input gagal');
        redirect('c_user/form_user_edit');
    }
}

```

Gambar 36 Function edit_user

```

public function hapus_user($user_id){
    $this->m_user->hapus_user($user_id);
    redirect('c_user/view_user');
}

```

Gambar 37 Function hapus_user

2. Kelas m_user

Kode program kelas m_user dapat dilihat pada Gambar 38 samapi dengan Gambar 42.

```

public function input_user($input){
    return $query = $this->db->query("INSERT INTO tb_user (user_id, username, password, bagian, level)
    VALUES ('$input[user_id]', '$input[username]', '$input[password]', '$input[bagian]', '$input[level]')");
}

```

Gambar 38 Function input_user

```

public function view_user(){
    $query = "SELECT * FROM tb_user";
    $result = $this->db->query($query);
    return $result;
}

```

Gambar 39 Function view_user

```

public function form_edit($user_id) {
    $query = $this->db->query("SELECT * FROM tb_user WHERE user_id='$user_id' ")->result_array();
    return $query;
}

```

Gambar 40 Function form_edit

```
public function edit_user($input){
    return $query = $this->db->query("UPDATE tb_user SET username='$input[username]', password='$input[password]',
    bagian='$input[bagian]', level='$input[level]' WHERE user_id='$input[user_id]'");
}
```

Gambar 41 Function edit_user

```
public function hapus_user($user_id) {
    return $query = $this->db->query("DELETE FROM tb_user WHERE user_id='$user_id'");
}
```

Gambar 42 Function hapus_user

F. Pengujian White Box

F.1 Kelas Controller

1. Kelas c_user



Gambar 102 Diagram Alir Function view_user

$$CC = E - N + 2 = 3 - 4 + 2 = 1$$

Tabel 16 Test Case Function view_user

Test case	Jika berhasil menampilkan halaman user
Target yang Diharapkan	Menampilkan halaman user
Hasil pengujian	Benar
Path/jalur	15-16-17-18

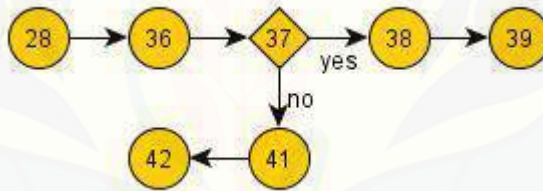


Gambar 103 Diagram Alir *Function* form_ user

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 17 *Test Case Function* form_ user

Test case	Jika berhasil menampilkan halaman form user
Target yang Diharapkan	Menampilkan halaman form user
Hasil pengujian	Benar
Path/jalur	22-23-24

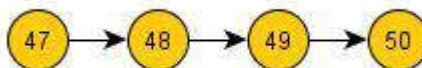


Gambar 104 Diagram Alir *Function* tambah_ user

$$CC = E - N + 2 = 6 - 7 + 2 = 1$$

Tabel 18 *Test Case Function* form_ user

Test case	Jika berhasil menyimpan data user
Target yang Diharapkan	Menyimpan data user ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	28-36-37-38-39, 28-36-37-41-42

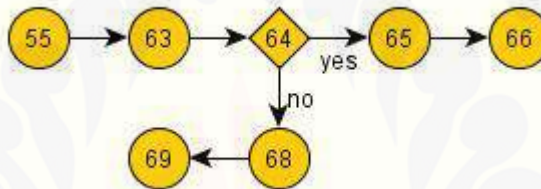


Gambar 105 Diagram Alir *Function* view_edit

$$CC = E - N + 2 = 3 - 4 + 2 = 1$$

Tabel 19 Test Case Function view_edit

Test case	Jika berhasil menampilkan halaman form edit user
Target yang Diharapkan	Menampilkan halaman form edit user
Hasil pengujian	Benar
Path/jalur	47-48-49-50

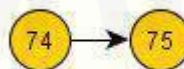


Gambar 106 Diagram Alir Function edit_user

$$CC = E - N + 2 = 6 - 7 + 2 = 1$$

Tabel 20 Test Case Function edit_user

Test case	Jika berhasil menyimpan data user yang telah diedit
Target yang Diharapkan	Menyimpan data user yang telah diedit
Hasil pengujian	Benar
Path/jalur	55-63-64-65-66, 55-63-64-68-69



Gambar 107 Diagram Alir Function hapus_user

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 21 Test Case Function hapus_user

Test case	Jika berhasil menghapus data user
Target yang Diharapkan	Data user pada <i>database</i> terhapus
Hasil pengujian	Benar
Path/jalur	74-75

F.2 Kelas Model

1. Kelas m_user



Gambar 128 Diagram Alir *Function* input_user

$$CC = E - N + 2 = 0 - 1 + 2 = 1$$

Tabel 42 *Test Case Function* input_user

Test case	Jika berhasil menyimpan data user
Target yang Diharapkan	Menyimpan data user ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	8



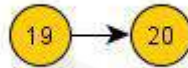
Gambar 129 Diagram Alir *Function* view_user

$$CC = E - N + 2 = 2 - 3 + 2 = 1$$

Tabel 43 *Test Case Function* view_user

Test case	Jika berhasil menampilkan data user
Target yang Diharapkan	Menampilkan data user
Hasil pengujian	Benar

Path/jalur	13-14-15
-------------------	----------



Gambar 130 Diagram Alir *Function* form_edit

$$CC = E - N + 2 = 1 - 2 + 2 = 1$$

Tabel 44 *Test Case Function* form_edit

Test case	Jika berhasil menampilkan data user yang akan diedit
Target yang Diharapkan	Menampilkan data user
Hasil pengujian	Benar
Path/jalur	19-20



Gambar 130 Diagram Alir *Function* edit_domain

$$CC = E - N + 2 = 0 - 1 + 2 = 1$$

Tabel 45 *Test Case Function* edit_user

Test case	Jika berhasil menyimpan data user yang diedit
Target yang Diharapkan	Menyimpan data user yang diedit ke <i>database</i>
Hasil pengujian	Benar
Path/jalur	24



Gambar 131 Diagram Alir *Function* hapus_user

$$CC = E - N + 2 = 0 - 1 + 2 = 1$$

Tabel 46 Test Case Function hapus_user

Test case	Jika berhasil menghapus data user
Target yang Diharapkan	Menghapus data user
Hasil pengujian	Benar
Path/jalur	29

G. Pengujian *Black Box*

G.1 Fitur Pengelolaan Data User

Tabel 64 Pengujian *Black Box* Pengelolaan Data User

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data user	Memilih menu user	Menampilkan tabel daftar user	Berhasil
2.	Menambah data user	Memilih tombol data baru	Menampilkan form tambah user	Berhasil
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i> b. Menampilkan halaman user kembali	Berhasil
3.	Mengubah data user	Memilih tombol edit	Menampilkan form edit user	Berhasil
		Memilih tombol simpan	a. Menyimpan data ke <i>database</i>	Berhasil

			b. Menampilkan halaman user kembali	
4.	Menghapus data user	Memilih tombol hapus	Menghapus data dari <i>database</i>	Berhasil

G.2 Fitur Pengelolaan Data Domain

Tabel 65 Pengujian *Black Box* Pengelolaan Data Domain

No.	Fitur	Aksi	Hasil	Kesimpulan
1.	Melihat data domain	Memilih menu domain	Menampilkan tabel daftar domain	Berhasil
2.	Menambah data domain	Memilih tombol data baru	Menampilkan form tambah domain	Berhasil
		Memilih tombol simpan	c. Menyimpan data ke <i>database</i> d. Menampilkan halaman domain kembali	Berhasil
3.	Mengubah data domain	Memilih tombol edit	Menampilkan form edit domain	Berhasil

		Memilih tombol simpan	c. Menyimpan data ke <i>database</i> d. Menampilkan halaman domain kembali	Berhasil
4.	Menghapus data domain	Memilih tombol hapus	Menghapus data dari <i>database</i>	Berhasil

