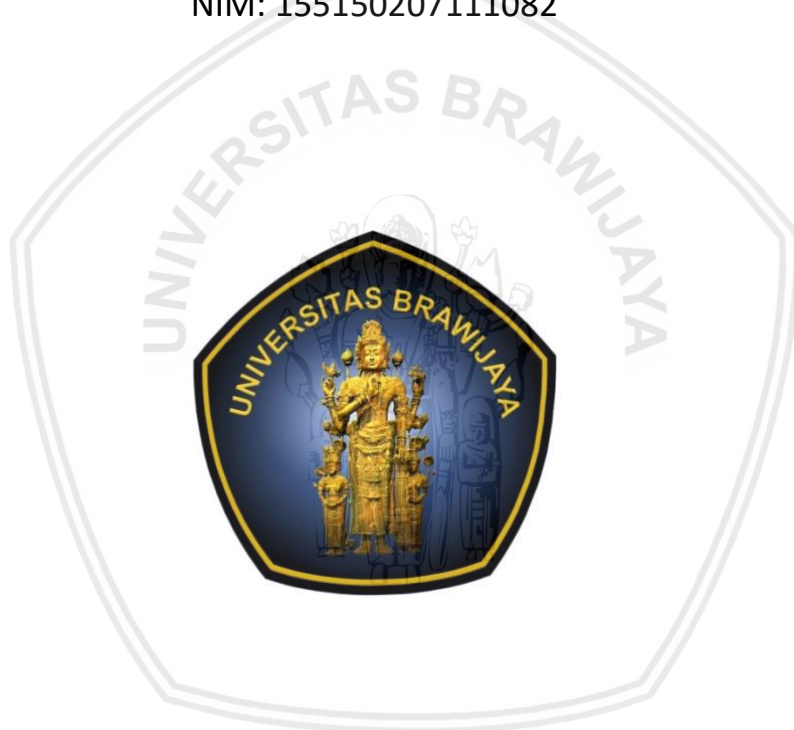


**SISTEM PENGEMBANGAN *WORKFLOW* MANAJEMEN KERJA (STUDI
KASUS: PT. CHEVRON PACIFIC INDONESIA)**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Lazuardi Dio Ramadhan
NIM: 155150207111082



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

SISTEM PENGEMBANGAN *WORKFLOW* MANAJEMEN KERJA
(STUDI KASUS: PT. CHEVRON PACIFIC INDONESIA)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Lazuardi Dio Ramadhan
NIM: 155150207111082

Skripsi ini telah diuji dan dinyatakan lulus pada
22 April 2019

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Fajar Pradana, S.ST., M.Eng.
NIP: 19871121 201504 1 004

Dosen Pembimbing II

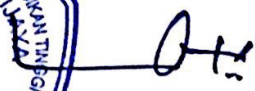


Faizatul Amalia S.Pd., M.Pd.
NIK: 201309 860821 2 001

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 22 April 2019



Lazuardi Dio Ramadhan
NIM: 155150207111082



PRAKATA

Puji dan syukur penulis panjatkan atas rahmat dan karunia Allah SWT sehingga penulis dapat menyelesaikan skripsi yang berjudul Sistem Pengembangan *Workflow* Manajemen Kerja (Studi Kasus: PT. Chevron Pacific Indonesia). Skripsi ini diajukan sebagai salah satu syarat untuk mendapatkan gelar Sarjana Komputer di Fakultas Ilmu Komputer, Universitas Brawijaya.

Dalam penyusunan skripsi ini penulis mendapat banyak dukungan dan bantuan dari berbagai pihak. Oleh sebab itu, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Fajar Pradana, S.ST., M.Eng. selaku Pembimbing I, yang telah membimbing dan mengarahkan penulis dalam proses penyelesaian skripsi.
2. Ibu Faizatul Amalia S.Pd., M.Pd. selaku Pembimbing II, yang telah membimbing dan mengarahkan penulis sehingga skripsi ini dapat penulis selesaikan.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya dan Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Bayudita atas dukungan dan menyediakan kebutuhan yang dibutuhkan pada topik skripsi ini.
5. Kedua orang tua dan keluarga yang telah memberikan doa dan dukungan selama proses pendidikan hingga penyelesaian skripsi.
6. Teman-teman angkatan Teknik Informasi 2015 yang selalu memberikan bantuan, informasi, dan motivasi.
7. Semua pihak yang tidak dapat disebutkan satu – persatu yang terlibat demi terselesaikannya penyusunan skripsi ini.

Penulis menyadari dalam proses penyusunan skripsi masih terdapat kekurangan, oleh karena itu kritik dan saran yang membangun diperlukan dalam penulisan selanjutnya. Penulis berharap semoga skripsi ini memberikan manfaat kepada seluruh pihak yang membutuhkannya.

Malang, 22 April 2019



Penulis

diormdn@gmail.com

ABSTRAK

Lazuardi Dio Ramadhan, Sistem Pengembangan *Workflow* Manajemen Kerja (Studi Kasus: PT. Chevron Pacific Indonesia)

Pembimbing: Fajar Pradana, S.ST., M.Eng. dan Faizatul Amalia S.Pd., M.Pd.

Teknologi informasi merupakan hal diperlukan untuk mendapatkan keunggulan serta lebih maju dan berkembang pada suatu perusahaan. Keunggulan teknologi informasi yaitu dapat membantu manajemen di suatu perusahaan karena menghasilkan informasi dengan cepat dan bermanfaat. Pada suatu perusahaan seperti PT Chevron Pacific Indonesia keberhasilan ditentukan salah satunya dengan kinerja manajemen pekerjaan yang baik. Manajemen pekerjaan pasti terkait dengan pekerjaan yang memiliki tenggat waktu tertentu yang akan menentukan kinerja dari perusahaan tersebut. Perusahaan PT Chevron Pacific Indonesia mempunyai banyak teknologi informasi pada bidang manajemen kerja, namun saat ini dalam manajemen kerja untuk membuat suatu proses pekerjaan harus membuat proses bisnis terlebih dahulu, kemudian setelah selesai dengan sistem yang berbeda baru dapat melakukan penugasan terhadap karyawannya. Hal tersebut menyebabkan masalah yaitu sulitnya melakukan penugasan secara langsung terhadap karyawan dan pelacakan proses pekerjaan. Saat ini dalam melakukan hal tersebut harus membandingkan ke dua sistem yang berbeda. PT Chevron Pacific Indonesia membutuhkan sistem untuk membantu pembuatan proses bisnis dan penugasan karyawan sekaligus dan dapat melacak suatu kendala pada proses bisnis tersebut termasuk karyawan yang bertanggung jawab pada proses tersebut. Dalam membuat sistem tersebut menggunakan pendekatan *workflow* yang merupakan alur proses kerja dan dapat dilakukan pelacakan yang di dalamnya dapat memberikan tugas ke partisipan atau karyawan tertentu. Dalam merepresentasikan *workflow* dilakukan pendekatan *Business Process Model Notation* (BPMN) yang merupakan suatu standar untuk memodelkan proses bisnis. Pada akhirnya sistem dapat melakukan pembuatan proses bisnis, penugasan karyawan, dan melakukan pelacakan tugas. Sistem ini dibangun pada *platform web* dan telah diuji dengan pengujian *whitebox* dan *blackbox*. Selanjutnya juga telah dilakukan pengujian *usability* dengan metode *System Usability Scale* dan sistem memasuki kategori *acceptable* yaitu sistem sudah dibuat sesuai keinginan pengguna.

Kata kunci: Manajemen Kerja, Proses Bisnis, Penugasan Kerja, *Workflow*, *Business Process Model Notation*

ABSTRACT

Lazuardi Dio Ramadhan, *Work Management Development Workflow System (Case Study: PT. Chevron Pacific Indonesia)*

Adviser: Fajar Pradana, S.ST., M.Eng, and Faizatul Amalia S.Pd., M.Pd.

Information technology is a necessary thing to get superiority and more advanced and developing in a company. The superiority of information technology is that it can help management in a company because it produces information quickly and is useful. In a company like PT Chevron Pacific Indonesia success is determined by one of them with good job management performance. Job management is definitely related to work that has a certain deadline that will determine the performance of the company. The company PT Chevron Pacific Indonesia has a lot of information technology in the field of work management, but now in work management to make a work process must make the business process first, then after finishing with a different system can only be assigned to its employees. This causes problems, namely the difficulty of directly assigning employees and tracking work processes. At this time in doing so, it must compare it to two different systems. PT Chevron Pacific Indonesia needs a system to help create business processes and assign employees at the same time and can track a problem in the business process including employees who are responsible for the process. In making the system use a workflow approach which is a workflow process and tracking can be done in which assignments to certain participants or employees. In representing workflows, a Business Process Model Notation (BPMN) approach is carried out which is a standard for modeling business processes. In the end the system can make business processes, assign employees, and carry out tracking tasks. This system is built on the web platform and has been tested with whitebox and blackbox testing. Furthermore, usability testing has been carried out using the System Usability Scale method and the system enters the acceptable category, namely the system has been made according to the user's wishes.

Keyword: *Work Management, Business Process, Job Assignment, Workflow, Business Process Model Notation*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xxi
DAFTAR LAMPIRAN	xxiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Rekayasa Perangkat Lunak	5
2.3 Pengembangan Perangkat Lunak	6
2.3.1 Model <i>Waterfall</i>	6
2.3.2 <i>Object Oriented Programming (OOP)</i>	7
2.3.3 <i>Unified Modeling Language (UML)</i>	8
2.3.4 <i>Model View Controller (MVC)</i>	14
2.3.5 <i>Workflow</i>	15
2.3.6 <i>Business Process Modelling Notation (BPMN)</i>	15
2.4 Teknologi Pengembangan Sistem.....	17
2.4.1 <i>Gojs Diagram</i>	17
2.4.2 Kendo UI.....	17
2.4.3 <i>.NET Framework</i>	17



2.4.4 <i>Entity Framework</i>	17
2.4.5 Microsoft SQL Server.....	18
2.4.6 Pengujian Perangkat Lunak.....	18
BAB 3 METODOLOGI	20
3.1 Studi Literatur	21
3.2 Analisis Kebutuhan	21
3.3 Perancangan Sistem.....	21
3.4 Implementasi Sistem	21
3.5 Pengujian	22
3.6 Kesimpulan dan Saran	22
BAB 4 ANALISIS KEBUTUHAN	23
4.1 Deskripsi Sistem.....	23
4.2 Identifikasi Aktor.....	23
4.3 Kebutuhan Fungsional	24
4.4 Kebutuhan Non Fungsional.....	37
4.5 Pemodelan Kebutuhan	37
4.5.1 <i>Use Case Diagram</i>	37
4.5.2 <i>Use Case Scenario</i>	38
BAB 5 PERANCANGAN DAN IMPLEMENTASI	75
5.1 Perancangan Sistem.....	75
5.1.1 Perancangan <i>Sequence Diagram</i>	75
5.1.2 Perancangan <i>Class Diagram</i>	78
5.1.3 Perancangan Komponen	79
5.1.4 Perancangan Basis Data	82
5.1.5 Perancangan Antarmuka.....	84
5.2 Implementasi Sistem	86
5.2.1 Spesifikasi Sistem	86
5.2.2 Implementasi Basis Data	87
5.2.3 Implementasi Kode Program	90
5.2.4 Implementasi Antarmuka	94
BAB 6 PENGUJIAN	96
6.1 Pengujian Unit.....	96

6.1.1 Pengujian Unit <i>TemplateWorkflowDetails()</i>	96
6.1.2 Pengujian Unit <i>TaskView()</i>	98
6.1.3 Pengujian Unit <i>MyWorkflowStatistics()</i>	103
6.2 Pengujian Integrasi	106
6.3 Pengujian Validasi	109
6.3.1 Pengujian Validasi Login	109
6.3.2 Pengujian Validasi Register	110
6.3.3 Pengujian Validasi Buat <i>Workflow</i>	112
6.3.4 Pengujian Validasi Lihat Petunjuk Cepat	114
6.3.5 Pengujian Validasi Pilih Tenggat Waktu	114
6.3.6 Pengujian Validasi <i>Undo</i> Objek	115
6.3.7 Pengujian Validasi <i>Redo</i> Objek	115
6.3.8 Pengujian Validasi <i>Cut</i> Objek	115
6.3.9 Pengujian Validasi <i>Copy</i> Objek	116
6.3.10 Pengujian Validasi <i>Paste</i> Objek	116
6.3.11 Pengujian Validasi <i>Delete</i> Objek	117
6.3.12 Pengujian Validasi Pilih Seluruh Objek	117
6.3.13 Pengujian Validasi Pindah Posisi Objek Kiri	117
6.3.14 Pengujian Validasi Pindah Posisi Objek Kanan	118
6.3.15 Pengujian Validasi Pindah Posisi Objek Atas	118
6.3.16 Pengujian Validasi Pindah Posisi Objek Bawah	119
6.3.17 Pengujian Validasi Tampil Kisi	119
6.3.18 Pengujian Lihat Jumlah <i>Template Workflow</i>	119
6.3.19 Pengujian Lihat Jumlah <i>Implemented Workflow</i>	120
6.3.20 Pengujian Validasi Lihat Daftar <i>Template Workflow</i>	120
6.3.21 Pengujian Validasi Lihat Cari <i>Template Workflow</i>	121
6.3.22 Pengujian Validasi Lihat <i>Detail</i> Daftar <i>Template Workflow</i>	122
6.3.23 Pengujian Validasi Lihat Daftar <i>Implemented Workflow</i>	122
6.3.24 Pengujian Validasi Lihat Cari <i>Implemented Workflow</i>	123
6.3.25 Pengujian Validasi Lihat Lihat <i>Detail</i> Daftar <i>Implemented Workflow</i>	124
6.3.26 Pengujian Validasi Lihat <i>Pie Chart</i> Persentase <i>Workflow</i>	124

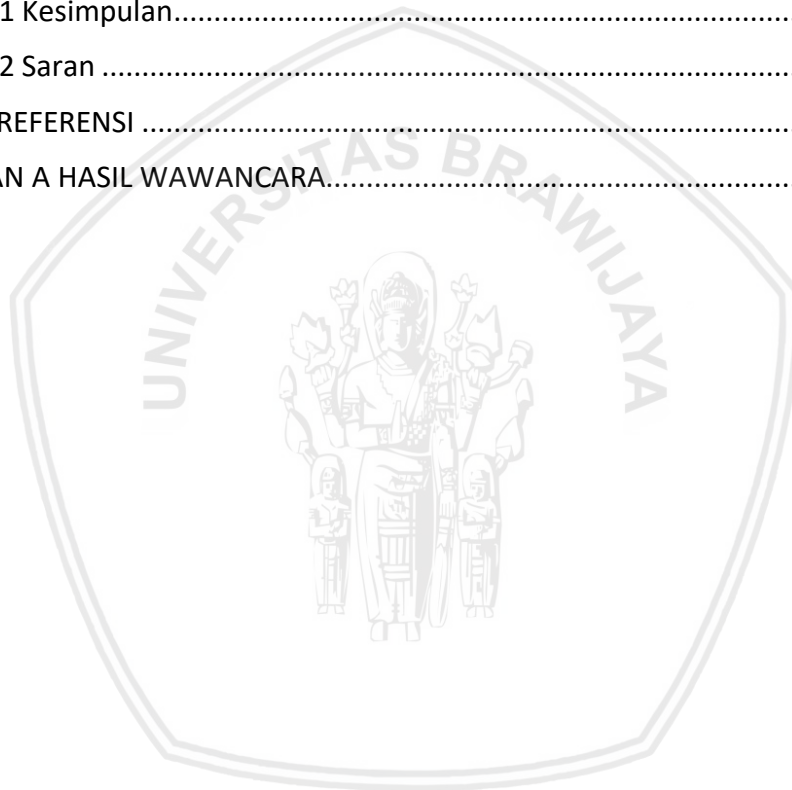


6.3.27 Pengujian Validasi Lihat Jumlah <i>Finished Workflow</i>	125
6.3.28 Pengujian Validasi Lihat Daftar <i>Finished Workflow</i>	125
6.3.29 Pengujian Validasi Lihat Cari <i>Finished Workflow</i>	126
6.3.30 Pengujian Validasi Lihat <i>Detail Finished Workflow</i>	127
6.3.31 Pengujian Validasi Lihat Statistik <i>Finished Workflow</i>	127
6.3.32 Pengujian Validasi Lihat Jumlah <i>Workflow By Template</i>	127
6.3.33 Pengujian Validasi Lihat Daftar <i>Workflow By Template</i>	128
6.3.34 Pengujian Validasi Lihat Cari <i>Workflow By Template</i>	128
6.3.35 Pengujian Validasi Lihat Statistik <i>Workflow By Template</i>	129
6.3.36 Pengujian Validasi Lihat Jumlah <i>Pending User</i>	130
6.3.37 Pengujian Validasi Lihat Jumlah <i>All User</i>	130
6.3.38 Pengujian Validasi Lihat Daftar <i>Pending Users</i>	130
6.3.39 Pengujian Validasi Lihat Cari <i>Pending Users</i>	131
6.3.40 Pengujian Validasi Verifikasi <i>Pending Users</i>	132
6.3.41 Pengujian Validasi Tolak <i>Pending Users</i>	133
6.3.42 Pengujian Validasi Lihat Daftar <i>All Users</i>	134
6.3.43 Pengujian Validasi Lihat Cari <i>All Users</i>	134
6.3.44 Pengujian Validasi Berhasil Lihat <i>Profile Admin</i>	136
6.3.45 Pengujian Validasi Berhasil Ubah <i>Password Admin</i>	136
6.3.46 Pengujian Validasi Berhasil Lihat <i>Template Workflow Admin</i> 137	
6.3.47 Pengujian Validasi Lihat Cari <i>Template Workflow Admin</i>	138
6.3.48 Pengujian Validasi Berhasil Pilih <i>Template Workflow Admin</i> .	139
6.3.49 Pengujian Validasi Berhasil Mulai <i>Template Workflow Admin</i>	139
6.3.50 Pengujian Validasi Berhasil Pilih <i>Person In Charge</i>	140
6.3.51 Pengujian Validasi Berhasil Unggah <i>File Workflow</i>	141
6.3.52 Pengujian Validasi Berhasil Lihat Daftar <i>My Workflow</i>	141
6.3.53 Pengujian Validasi Cari <i>My Workflow</i>	142
6.3.54 Pengujian Validasi Berhasil Lihat <i>Detail My Workflow</i>	143
6.3.55 Pengujian Validasi Berhasil Lihat Statistik <i>My Workflow</i>	143
6.3.56 Pengujian Validasi Berhasil Lihat Daftar <i>Task</i>	144
6.3.57 Pengujian Validasi Cari Daftar <i>Task</i>	144
6.3.58 Pengujian Validasi Berhasil Lihat Isi <i>Form Task</i>	145

6.3.59 Pengujian Validasi Berhasil Unduh <i>File</i> individu <i>Person In Charge</i>	146
6.3.60 Pengujian Validasi Berhasil Unduh Seluruh <i>File Person In Charge</i>	147
6.3.61 Pengujian Validasi Berhasil Kirim Notifikasi.....	147
6.3.62 Pengujian Validasi Berhasil Lihat Daftar <i>File</i>	148
6.3.63 Pengujian Validasi Berhasil Unduh <i>File</i> Individu	148
6.3.64 Pengujian Validasi Berhasil Unduh Seluruh <i>File</i>	149
6.3.65 Pengujian Validasi Berhasil Terima Notifikasi <i>Workflow</i> Selesai	149
6.3.66 Pengujian Validasi Berhasil Lihat Daftar Notifikasi <i>Workflow Manager</i>	150
6.3.67 Pengujian Validasi Berhasil Lihat <i>Profile Workflow Manager</i> 150	
6.3.68 Pengujian Validasi Berhasil Ubah <i>Password Workflow Manager</i>	150
6.3.69 Pengujian Validasi Berhasil Lihat Daftar Tugas	152
6.3.70 Pengujian Validasi Cari Tugas.....	152
6.3.71 Pengujian Validasi Berhasil Lihat <i>Shared Files</i>	153
6.3.72 Pengujian Validasi Berhasil Unduh <i>File</i> Individu <i>Shared Files</i> . 153	
6.3.73 Pengujian Validasi Berhasil Unduh Seluruh <i>File Shared Files</i> . 154	
6.3.74 Pengujian Validasi Berhasil Lihat Jumlah Tugas Sebelum.....	154
6.3.75 Pengujian Validasi Berhasil Lihat Daftar Tugas Sebelum	155
6.3.76 Pengujian Validasi Cari Tugas Sebelum.....	155
6.3.77 Pengujian Validasi Berhasil Lihat Isi <i>Form</i> Tugas Sebelum	156
6.3.78 Pengujian Validasi Berhasil Unduh <i>File</i> Individu <i>Person In Charge</i>	157
<i>Files</i>	157
6.3.79 Pengujian Validasi Berhasil Unduh Seluruh <i>File Person In Charge</i>	157
<i>Files</i>	157
6.3.80 Pengujian Validasi Berhasil Menyelesaikan Tugas.....	158
6.3.81 Pengujian Validasi Berhasil Unggah <i>File</i> Tugas	158
6.3.82 Pengujian Validasi Berhasil Lihat <i>Workflow</i> Tugas	159



6.3.83 Pengujian Validasi Berhasil Lihat Daftar Notifikasi <i>Person In Charge</i>	159
6.3.84 Pengujian Validasi Berhasil Lihat <i>Profile Person In Charge</i>	159
6.3.85 Pengujian Validasi Berhasil Ubah <i>Password Person In Charge</i>	160
6.3.86 Pengujian Validasi Berhasil <i>Logout</i>	161
6.4 Pengujian <i>Usability</i>	161
6.4.1 Hasil Pengujian <i>Usability</i>	162
BAB 7 PENUTUP	165
7.1 Kesimpulan.....	165
7.2 Saran	165
DAFTAR REFERENSI	166
LAMPIRAN A HASIL WAWANCARA.....	168



DAFTAR TABEL

Tabel 2.1 Simbol <i>Activity Diagram</i>	13
Tabel 4.1 Identifikasi Aktor Sistem.....	23
Tabel 4.2 Kebutuhan Fungsional	24
Tabel 4.3 Kebutuhan Non Fungsional	37
Tabel 4.4 <i>Use Case Scenario Login</i>	38
Tabel 4.5 <i>Use Case Scenario Register</i>	39
Tabel 4.6 <i>Use Case Scenario Buat Workflow</i>	40
Tabel 4.7 <i>Use Case Scenario Lihat Petunjuk Cepat</i>	40
Tabel 4.8 <i>Use Case Scenario Pilih Tenggat Waktu</i>	41
Tabel 4.9 <i>Use Case Scenario Undo Objek</i>	41
Tabel 4.10 <i>Use Case Scenario Redo Objek</i>	41
Tabel 4.11 <i>Use Case Scenario Cut Objek</i>	42
Tabel 4.12 <i>Use Case Scenario Copy Objek</i>	42
Tabel 4.13 <i>Use Case Scenario Paste Objek</i>	42
Tabel 4.14 <i>Use Case Scenario Delete Objek</i>	43
Tabel 4.15 <i>Use Case Scenario Pilih Seluruh Objek</i>	43
Tabel 4.16 <i>Use Case Scenario Pindah Posisi Objek Kiri</i>	43
Tabel 4.17 <i>Use Case Scenario Pindah Posisi Objek Kanan</i>	44
Tabel 4.18 <i>Use Case Scenario Pindah Posisi Objek Atas</i>	44
Tabel 4.19 <i>Use Case Scenario Pindah Posisi Objek Bawah</i>	45
Tabel 4.20 <i>Use Case Scenario Tampil Kisi</i>	45
Tabel 4.21 <i>Use Case Scenario Lihat Jumlah Template Workflow</i>	45
Tabel 4.22 <i>Use Case Scenario Lihat Jumlah Implemented Workflow</i>	46
Tabel 4.23 <i>Use Case Scenario Lihat Daftar Template Workflow</i>	46
Tabel 4.24 <i>Use Case Scenario Cari Template Workflow</i>	46
Tabel 4.25 <i>Use Case Scenario Lihat Detail Template Workflow</i>	47
Tabel 4.26 <i>Use Case Scenario Lihat Implemented Workflow</i>	47
Tabel 4.27 <i>Use Case Scenario Cari Implemented Workflow</i>	48
Tabel 4.28 <i>Use Case Scenario Lihat Detail Daftar Implemented Workflow</i>	48
Tabel 4.29 <i>Use Case Scenario Lihat Pie Chart Persentase Workflow</i>	48

Tabel 4.30 Use Case Scenario Lihat Jumlah <i>Finished Workflow</i>	49
Tabel 4.31 Use Case Scenario Lihat Daftar <i>Finished Workflow</i>	49
Tabel 4.32 Use Case Scenario Cari <i>Finished Workflow</i>	49
Tabel 4.33 Use Case Scenario Lihat <i>Detail Finished Workflow</i>	50
Tabel 4.34 Use Case Scenario Lihat Statistik <i>Finished Workflow</i>	50
Tabel 4.35 Use Case Scenario Lihat Jumlah <i>Workflow By Template</i>	51
Tabel 4.36 Use Case Scenario Lihat Daftar <i>Workflow By Template</i>	51
Tabel 4.37 Use Case Scenario Cari <i>Workflow By Template</i>	51
Tabel 4.38 Use Case Scenario Lihat Statistik <i>Workflow By Template</i>	52
Tabel 4.39 Use Case Scenario Lihat Jumlah <i>Pending User</i>	52
Tabel 4.40 Use Case Scenario Lihat Jumlah <i>All User</i>	53
Tabel 4.41 Use Case Scenario Lihat Daftar <i>Pending Users</i>	53
Tabel 4.42 Use Case Scenario Cari <i>Pending Users</i>	53
Tabel 4.43 Use Case Scenario Verifikasi <i>Pending Users</i>	54
Tabel 4.44 Use Case Scenario Tolak <i>Pending Users</i>	54
Tabel 4.45 Use Case Scenario Lihat Daftar <i>All Users</i>	55
Tabel 4.46 Use Case Scenario Cari <i>All Users</i>	55
Tabel 4.47 Use Case Scenario Lihat <i>Profile Admin</i>	56
Tabel 4.48 Use Case Scenario Ubah <i>Password Admin</i>	56
Tabel 4.49 Use Case Scenario Lihat <i>Template Workflow Admin</i>	57
Tabel 4.50 Use Case Scenario Cari <i>Template Workflow Admin</i>	57
Tabel 4.51 Use Case Scenario Pilih <i>Template Workflow Admin</i>	58
Tabel 4.52 Use Case Scenario Mulai <i>Template Workflow Admin</i>	58
Tabel 4.53 Use Case Scenario Pilih <i>Person In Charge</i>	58
Tabel 4.54 Use Case Scenario Unggah <i>File Workflow</i>	59
Tabel 4.55 Use Case Scenario Lihat Daftar <i>My Workflow</i>	60
Tabel 4.56 Use Case Scenario Cari <i>My Workflow</i>	60
Tabel 4.57 Use Case Scenario Lihat <i>Detail My Workflow</i>	60
Tabel 4.58 Use Case Scenario Lihat Statistik <i>My Workflow</i>	61
Tabel 4.59 Use Case Scenario Lihat Daftar <i>Task</i>	61
Tabel 4.60 Use Case Scenario Cari Daftar <i>Task</i>	62
Tabel 4.61 Use Case Scenario Lihat Isi <i>Form Task</i>	62



Tabel 4.62 <i>Use Case Scenario</i> Unduh File Individu <i>Person In Charge</i>	63
Tabel 4.63 <i>Use Case Scenario</i> Unduh Seluruh File <i>Person In Charge</i>	63
Tabel 4.64 <i>Use Case Scenario</i> Kirim Notifikasi	63
Tabel 4.65 <i>Use Case Scenario</i> Lihat Daftar File	64
Tabel 4.66 <i>Use Case Scenario</i> Unduh File Individu	64
Tabel 4.67 <i>Use Case Scenario</i> Unduh Seluruh File	65
Tabel 4.68 <i>Use Case Scenario</i> Lihat Daftar Notifikasi <i>Workflow Manager</i>	65
Tabel 4.69 <i>Use Case Scenario</i> Lihat <i>Profile Workflow Manager</i>	66
Tabel 4.70 <i>Use Case Scenario</i> Ubah <i>Password Workflow Manager</i>	66
Tabel 4.71 <i>Use Case Scenario</i> Lihat Daftar Tugas	66
Tabel 4.72 <i>Use Case Scenario</i> Cari Tugas.....	67
Tabel 4.73 <i>Use Case Scenario</i> Lihat <i>Shared Files</i>	67
Tabel 4.74 <i>Use Case Scenario</i> Unduh File Individu <i>Shared Files</i>	68
Tabel 4.75 <i>Use Case Scenario</i> Unduh Seluruh File <i>Shared Files</i>	68
Tabel 4.76 <i>Use Case Scenario</i> Lihat Jumlah Tugas Sebelum	68
Tabel 4.77 <i>Use Case Scenario</i> Lihat Daftar Tugas Sebelum	69
Tabel 4.78 <i>Use Case Scenario</i> Cari Tugas Sebelum	69
Tabel 4.79 <i>Use Case Scenario</i> Lihat Isi <i>Form</i> Tugas Sebelum.....	70
Tabel 4.80 <i>Use Case Scenario</i> Unduh File Individu <i>Person In Charge Files</i>	70
Tabel 4.81 <i>Use Case Scenario</i> Unduh Seluruh File <i>Person In Charge Files</i>	71
Tabel 4.82 <i>Use Case Scenario</i> Menyelesaikan Tugas.....	71
Tabel 4.83 <i>Use Case Scenario</i> Unggah File Tugas	71
Tabel 4.84 <i>Use Case Scenario</i> Lihat <i>Workflow</i> Tugas.....	72
Tabel 4.85 <i>Use Case Scenario</i> Lihat Daftar Notifikasi <i>Person In Charge</i>	72
Tabel 4.86 <i>Use Case Scenario</i> Lihat <i>Profile Person In Charge</i>	73
Tabel 4.87 <i>Use Case Scenario</i> Ubah <i>Password Person In Charge</i>	73
Tabel 4.88 <i>Use Case Scenario</i> <i>Logout</i>	73
Tabel 5.1 Algoritme <i>TemplateWorkflowDetail</i>	80
Tabel 5.2 Algoritme <i>TaskView</i>	80
Tabel 5.3 Algoritme <i>MyWorkflowStatistics</i>	81
Tabel 5.4 Spesifikasi Perangkat Keras	86
Tabel 5.5 Spesifikasi Perangkat Lunak	87



Tabel 5.6 Implementasi Tabel DataStorage	87
Tabel 5.7 Implementasi Tabel Form.....	87
Tabel 5.8 Implementasi Tabel LinkdataArray	88
Tabel 5.9 Implementasi Tabel LinkdataArrayIM.....	88
Tabel 5.10 Implementasi Tabel NodedataArray.....	88
Tabel 5.11 Implementasi Tabel NodedataArrayIM	89
Tabel 5.12 Implementasi Tabel Notifications	89
Tabel 5.13 Implementasi Tabel RootObject.....	89
Tabel 5.14 Implementasi Tabel RootObjectIM	90
Tabel 5.15 Implementasi Tabel Users.....	90
Tabel 5.16 Kode Program <i>TemplateWorkflowDetail()</i>	91
Tabel 5.17 Kode Program MyWorkflowStatistics	92
Tabel 5.18 Kode Program TaskView.....	92
Tabel 6.1 <i>Pseudocode</i> Algoritme <i>TemplateWorkflowDetails</i>	96
Tabel 6.2 Hasil Pengujian Unit Method <i>TemplateWorkflowDetails</i>	98
Tabel 6.3 <i>Pseudocode</i> Algoritme <i>TaskView</i>	99
Tabel 6.4 Hasil Pengujian Unit Method <i>TaskView</i>	101
Tabel 6.5 Algoritme MyWorkflowStatistics().....	104
Tabel 6.6 Hasil Pengujian Unit Method <i>MyWorkflowStatistics</i>	105
Tabel 6.7 <i>Pseudocode</i> Algoritme MyWorkflowStatistics ().....	106
Tabel 6.8 Hasil Pengujian Integrasi	108
Tabel 6.9 Kasus Uji Berhasil <i>Login</i>	109
Tabel 6.10 Kasus Uji Login Alternatif 1	109
Tabel 6.11 Kasus Uji Login Alternatif 2	110
Tabel 6.12 Kasus Uji Berhasil <i>Register</i>	110
Tabel 6.13 Kasus Uji <i>Register</i> Alternatif 1	111
Tabel 6.14 Kasus Uji <i>Register</i> Alternatif 2	111
Tabel 6.15 Kasus Uji <i>Register</i> Alternatif 3	112
Tabel 6.16 Kasus Uji Berhasil Buat <i>Workflow</i>	112
Tabel 6.17 Kasus Uji Buat <i>Workflow</i> Alternatif 1.....	113
Tabel 6.18 Kasus Uji Berhasil Lihat Petunjuk Cepat.....	114
Tabel 6.19 Kasus Uji Berhasil Pilih Tenggat Waktu	114

Tabel 6.20 Kasus Uji Berhasil <i>Undo</i> Objek	115
Tabel 6.21 Kasus Uji Berhasil <i>Redo</i> Objek	115
Tabel 6.22 Kasus Uji Berhasil <i>Cut</i> Objek.....	115
Tabel 6.23 Kasus Uji Berhasil <i>Copy</i> Objek	116
Tabel 6.24 Kasus Uji Berhasil <i>Paste</i> Objek	116
Tabel 6.25 Kasus Uji Berhasil <i>Delete</i> Objek.....	117
Tabel 6.26 Kasus Uji Berhasil Pilih Seluruh Objek	117
Tabel 6.27 Kasus Uji Berhasil Pindah Posisi Objek Kiri.....	117
Tabel 6.28 Kasus Uji Berhasil Pindah Posisi Objek Kanan	118
Tabel 6.29 Kasus Uji Berhasil Pindah Posisi Objek Atas.....	118
Tabel 6.30 Kasus Uji Berhasil Pindah Posisi Objek Bawah	119
Tabel 6.31 Kasus Uji Berhasil Tampil Kisi	119
Tabel 6.32 Kasus Uji Berhasil Lihat Jumlah <i>Template Workflow</i>	119
Tabel 6.33 Kasus Uji Berhasil Lihat Jumlah <i>Implemented Workflow</i>	120
Tabel 6.34 Kasus Uji Berhasil Lihat Daftar <i>Template Workflow</i>	120
Tabel 6.35 Kasus Uji Berhasil Cari <i>Template Workflow</i>	121
Tabel 6.36 Kasus Uji Berhasil Cari <i>Template Workflow</i> Alternatif 1.....	121
Tabel 6.37 Kasus Uji Berhasil Lihat <i>Detail</i> Daftar <i>Template Workflow</i>	122
Tabel 6.38 Kasus Uji Berhasil Lihat Daftar <i>Implemented Workflow</i>	122
Tabel 6.39 Kasus Uji Berhasil Cari <i>Implemented Workflow</i>	123
Tabel 6.40 Kasus Uji Berhasil Cari <i>Implemented Workflow</i> Alternatif 1.....	123
Tabel 6.41 Kasus Uji Berhasil Lihat <i>Detail</i> Daftar <i>Implemented Workflow</i>	124
Tabel 6.42 Kasus Uji Berhasil Lihat <i>Pie Chart</i> Persentase <i>Workflow</i>	124
Tabel 6.43 Kasus Uji Berhasil Lihat Jumlah <i>Finished Workflow</i>	125
Tabel 6.44 Kasus Uji Berhasil Lihat Daftar <i>Finished Workflow</i>	125
Tabel 6.45 Kasus Uji Berhasil Cari <i>Finished Workflow</i>	126
Tabel 6.46 Kasus Uji Berhasil Cari <i>Finished Workflow</i> Alternatif 1.....	126
Tabel 6.47 Kasus Uji Berhasil Lihat <i>Detail Finished Workflow</i>	127
Tabel 6.48 Kasus Uji Berhasil Lihat Statistik <i>Finished Workflow</i>	127
Tabel 6.49 Kasus Uji Berhasil Lihat Jumlah <i>Workflow By Template</i>	127
Tabel 6.50 Kasus Uji Berhasil Lihat Daftar <i>Workflow By Template</i>	128
Tabel 6.51 Kasus Uji Berhasil Cari <i>Workflow By Template</i>	128



Tabel 6.52 Kasus Uji Berhasil Cari <i>Workflow By Template</i> Alternatif 1	129
Tabel 6.53 Kasus Uji Berhasil Lihat Statistik <i>Workflow By Template</i>	129
Tabel 6.54 Kasus Uji Berhasil Lihat Jumlah <i>Pending User</i>	130
Tabel 6.55 Kasus Uji Berhasil Lihat Jumlah <i>All User</i>	130
Tabel 6.56 Kasus Uji Berhasil Lihat Daftar <i>Pending Users</i>	130
Tabel 6.57 Kasus Uji Berhasil Cari <i>Pending Users</i>	131
Tabel 6.58 Kasus Uji Berhasil Cari <i>Pending Users</i> Alternatif 1	132
Tabel 6.59 Kasus Uji Berhasil Verifikasi <i>Pending Users</i>	132
Tabel 6.60 Kasus Uji Berhasil Cari <i>Pending Users</i> Alternatif 1	133
Tabel 6.61 Kasus Uji Berhasil Tolak <i>Pending Users</i>	133
Tabel 6.62 Kasus Uji Tolak <i>Pending Users</i> Alternatif 1	133
Tabel 6.63 Kasus Uji Berhasil Lihat Daftar <i>All Users</i>	134
Tabel 6.64 Kasus Uji Berhasil Cari <i>All Users</i>	134
Tabel 6.65 Kasus Uji Cari <i>All Users</i> Alternatif 1	135
Tabel 6.66 Kasus Uji Berhasil Lihat <i>Profile Admin</i>	136
Tabel 6.67 Kasus Uji Berhasil Ubah <i>Password Admin</i>	136
Tabel 6.68 Kasus Uji Berhasil Ubah <i>Password Admin</i>	137
Tabel 6.69 Kasus Uji Berhasil Lihat <i>Template Workflow Admin</i>	137
Tabel 6.70 Kasus Uji Berhasil Cari <i>Template Workflow Admin</i>	138
Tabel 6.71 Kasus Uji Cari <i>All Users</i> Alternatif 1	138
Tabel 6.72 Kasus Uji Berhasil Pilih <i>Template Workflow Admin</i>	139
Tabel 6.73 Kasus Uji Berhasil Mulai <i>Template Workflow Admin</i>	139
Tabel 6.74 Kasus Uji Mulai <i>Template Workflow Admin</i> Alternatif 1	140
Tabel 6.75 Kasus Uji Berhasil Pilih <i>Person In Charge</i>	140
Tabel 6.76 Kasus Uji Berhasil Unggah <i>File Workflow</i>	141
Tabel 6.77 Kasus Uji Berhasil Lihat Daftar <i>My Workflow</i>	142
Tabel 6.78 Kasus Uji Berhasil Cari <i>My Workflow</i>	142
Tabel 6.79 Kasus Uji Cari <i>All Users</i> Alternatif 1	143
Tabel 6.80 Kasus Uji Berhasil Lihat <i>Detail My Workflow</i>	143
Tabel 6.81 Kasus Uji Berhasil Lihat Statistik <i>My Workflow</i>	143
Tabel 6.82 Kasus Uji Berhasil Lihat Daftar <i>Task</i>	144
Tabel 6.83 Kasus Uji Berhasil Cari Daftar <i>Task</i>	144



Tabel 6.84 Kasus Uji Cari Daftar <i>Task</i> Alternatif 1	145
Tabel 6.85 Kasus Uji Berhasil Lihat Isi <i>Form Task</i>	145
Tabel 6.86 Kasus Uji Berhasil Lihat Isi <i>Form Task</i> Alternatif 1	146
Tabel 6.87 Kasus Uji Berhasil Unduh <i>File</i> individu <i>Person In Charge</i>	146
Tabel 6.88 Kasus Uji Berhasil Unduh Seluruh <i>File Person In Charge</i>	147
Tabel 6.89 Kasus Uji Berhasil Kirim Notifikasi	147
Tabel 6.90 Kasus Uji Berhasil Lihat Daftar <i>File</i>	148
Tabel 6.91 Kasus Uji Berhasil Unduh <i>File</i> Individu	148
Tabel 6.92 Kasus Uji Berhasil Unduh Seluruh <i>File</i>	149
Tabel 6.93 Kasus Uji Berhasil Terima Notifikasi <i>Workflow</i> Selesai	149
Tabel 6.94 Kasus Uji Berhasil Lihat Daftar Notifikasi <i>Workflow Manager</i>	150
Tabel 6.95 Kasus Uji Berhasil Lihat <i>Profile Workflow Manager</i>	150
Tabel 6.96 Kasus Uji Berhasil Ubah <i>Password Workflow Manager</i>	150
Tabel 6.97 Kasus Uji Berhasil Ubah <i>Password Workflow Manager</i>	151
Tabel 6.98 Kasus Uji Berhasil Lihat Daftar Tugas	152
Tabel 6.99 Kasus Uji Berhasil Cari Tugas	152
Tabel 6.100 Kasus Cari Tugas Alternatif 1.....	153
Tabel 6.101 Kasus Uji Berhasil Lihat <i>Shared Files</i>	153
Tabel 6.102 Kasus Uji Berhasil Unduh <i>File</i> Individu <i>Shared Files</i>	153
Tabel 6.103 Kasus Uji Berhasil Unduh Seluruh <i>File Shared Files</i>	154
Tabel 6.104 Kasus Uji Berhasil Lihat Jumlah Tugas Sebelum.....	154
Tabel 6.105 Kasus Uji Berhasil Lihat Daftar Tugas Sebelum	155
Tabel 6.106 Kasus Uji Berhasil Cari Tugas Sebelum.....	155
Tabel 6.107 Kasus Cari Tugas Sebelum Alternatif 1.....	156
Tabel 6.108 Kasus Uji Berhasil Lihat Isi <i>Form</i> Tugas Sebelum	156
Tabel 6.109 Kasus Uji Berhasil Unduh <i>File</i> Individu <i>Person In Charge Files</i>	157
Tabel 6.110 Kasus Uji Berhasil Unduh Seluruh <i>File Person In Charge Files</i>	157
Tabel 6.111 Kasus Uji Berhasil Menyelesaikan Tugas.....	158
Tabel 6.112 Kasus Uji Unggah <i>File</i> Tugas	158
Tabel 6.113 Kasus Uji Lihat <i>Workflow</i> Tugas	159
Tabel 6.114 Kasus Uji Lihat Daftar Notifikasi <i>Person In Charge</i>	159
Tabel 6.115 Kasus Uji Berhasil Lihat <i>Profile Person In Charge</i>	159

Tabel 6.116 Kasus Uji Berhasil Ubah <i>Password Person In Charge</i>	160
Tabel 6.117 Kasus Uji Berhasil Ubah <i>Password Person In Charge</i>	160
Tabel 6.118 Kasus Uji Berhasil <i>Logout</i>	161
Tabel 6.122 Daftar Pertanyaan SUS	162
Tabel 6.123 Hasil Kuesioner Pertanyaan SUS	163
Tabel 6.124 Hasil Perhitungan Skor Pertanyaan SUS.....	163



DAFTAR GAMBAR

Gambar 2.1 Metode <i>Waterfall</i>	7
Gambar 2.2 <i>Use Case Diagram</i>	9
Gambar 2.3 Hubungan <i>Association</i>	9
Gambar 2.4 Hubungan <i>Extends</i>	10
Gambar 2.5 Hubungan <i>Depends On</i>	10
Gambar 2.6 <i>Class</i>	11
Gambar 2.7 <i>Association</i>	11
Gambar 2.8 <i>Composition</i>	11
Gambar 2.9 <i>Agregation</i>	11
Gambar 2.10 <i>Class Diagram</i>	12
Gambar 2.11 <i>Sequence Diagram</i>	14
Gambar 2.12 <i>Model View Controller</i>	15
Gambar 2.13 <i>Event BPMN</i>	16
Gambar 2.14 <i>Activity dan Gateway BPMN</i>	16
Gambar 2.15 <i>Sequence BPMN</i>	16
Gambar 2.16 <i>Swimlanes BPMN</i>	16
Gambar 3.1 Metodologi Penelitian.....	20
Gambar 4.1 Diagram Use Case	38
Gambar 5.1 <i>Sequence Diagram</i> Lihat <i>Detail Template Workflow</i>	76
Gambar 5.2 <i>Sequence Diagram</i> Lihat <i>Shared Files</i>	77
Gambar 5.3 <i>Sequence Diagram</i> Lihat <i>Statistik My Workflow</i>	78
Gambar 5.4 Perancangan Umum <i>Class Diagram</i>	78
Gambar 5.5 Perancangan <i>Class Diagram Model</i>	79
Gambar 5.6 Perancangan <i>Class Diagram Controller</i>	79
Gambar 5.7 <i>Entity Relationship Diagram</i> Sistem.....	82
Gambar 5.8 <i>Physical Data Model (PDM)</i>	83
Gambar 5.9 Antarmuka <i>Lihat Detail Template Workflow</i>	84
Gambar 5.10 Antarmuka <i>Lihat Detail Tugas</i>	85
Gambar 5.11 Antarmuka <i>Lihat Statistik My Workflow</i>	86
Gambar 5.12 Implementasi Antarmuka <i>Lihat Detail Template Workflow</i>	94

Gambar 5.13 Implementasi Antarmuka Lihat *Detail* Tugas..... 95
Gambar 5.14 Implementasi Antarmuka Lihat Statistik *My Workflow*..... 95
Gambar 6.1 *Flowgraph Method TemplateWorkflowDetails*..... 97
Gambar 6.2 *Flowgraph Method TaskView*..... 100
Gambar 6.3 *Flowgraph Method MyWorkflowStatistics* 104
Gambar 6.4 *Flowgraph Method MyWorkflowStatistics* 107
Gambar 6.5 *Rating dan Skala Konversi SUS*..... 163



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA..... 168



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada suatu perusahaan, teknologi informasi merupakan hal yang sangat diperlukan. Saat ini salah satu usaha untuk mendapatkan keunggulan dalam persaingan bisnis yaitu dengan melakukan penerapan teknologi informasi yang berperan sebagai alat bantu perusahaan menjadi lebih maju dan juga berkembang (Shaqiri, 2015). Keberadaan inovasi teknologi tentu akan membuat perubahan baru bagi karyawan dan juga perusahaan. Pada dasarnya dengan kemunculan teknologi informasi pada setiap bisnis sudah dapat dirasakan manfaatnya (Nuskiya, 2018). Kemudian dengan adanya inovasi teknologi tentu dapat membantu manajemen dalam suatu pekerjaan di perusahaan. Inovasi teknologi dalam hal manajemen dapat lebih banyak menghasilkan informasi dibandingkan manusia pada umumnya, oleh karena itu teknologi manajemen dalam pekerjaan diperlukan pada saat ini karena dapat dirasakan teknologi tersebut lebih cepat dalam hal interdependensi daripada siapa pun mengelola dan juga kecepatan dalam perubahan (Beyrouiti, 2006).

Keberhasilan suatu perusahaan ditentukan dari kinerja manajemen yang telah diatur suatu proses pekerjaan (Syarif, 2014). Begitu pula pada perusahaan PT Chevron Pacific Indonesia yang merupakan salah satu perusahaan minyak kontraktor terbesar di Indonesia. PT Chevron Pacific Indonesia di dalamnya memiliki berbagai macam departemen dan proses bisnis yang tentu diperlukan manajemen pekerjaan di dalamnya. Dalam suatu proses pekerjaan pasti terdapat tenggat waktu tertentu yang telah ditentukan dan adanya proses pekerjaan yang melebihi tenggat waktu yang bisa memengaruhi kinerja pekerjaan. Hasil kinerja dari suatu proses pekerjaan dapat memengaruhi produktivitas perusahaan (Febijyantini et al, 2015). Suatu perusahaan jika manajemen proses pekerjaan dapat diatur dengan baik maka akan membuat perusahaan tersebut menjadi lebih optimal (Chandra dan Prasetya, 2015). Pada dasarnya suatu keberhasilan perusahaan dapat dinilai dari kinerjanya.

Saat ini PT Chevron Pacific Indonesia memiliki berbagai macam teknologi informasi pada perusahaannya termasuk pada bidang manajemen kerja. Pada bidang tersebut, PT Chevron Pacific Indonesia dalam membuat program pekerjaan harus membuat proses bisnis dengan notasi BPMN (*Business Process Model Notation*) yang di dalamnya terdapat alur proses pekerjaan pada suatu sistem terlebih dahulu, kemudian pada sistem yang berbeda melakukan penugasan kepada karyawannya. Terkait dengan perpindahan sistem tersebut ditemukannya suatu masalah yakni sulitnya melakukan penugasan terhadap karyawan dan pelacakan performa alur proses pekerjaan karena pada sistem penugasan tidak terdapat alur proses pekerjaannya sehingga untuk melakukan penugasan ataupun pelacakan perlu membandingkan dengan sistem pembuatan proses bisnis yang dapat berpengaruh pada performa perusahaan dari segi efektivitas dan efisiensi. Pada PT Chevron Pacific Indonesia Karena dalam mendukung kinerja perusahaan

dalam pelaksanaannya dibutuhkan efektivitas dan efisiensi (Suhartono, 2010). Dampak yang sangat dirasakan oleh PT Chevron Pacific Indonesia yaitu dalam memeriksa program pekerjaan yang memiliki performa yang buruk karena untuk melacak hal tersebut harus melihat kedua sistem yang berbeda dan juga jika terdapat program pekerjaan yang sejenis maka untuk melakukan rekapitulasi program pekerjaan yang sejenis tersebut harus membandingkannya kembali ke sistem yang berbeda. Sebagai contoh program pekerjaan wawancara terhadap subjek calon karyawan yang banyak maka disini akan merekapitulasi seluruh wawancara setiap subjek, tidak hanya satu subjek saja. Perusahaan PT Chevron Pacific Indonesia membutuhkan sistem untuk membantu dalam pembuatan proses bisnis sekaligus penugasan sehingga mampu melacak suatu kendala pada suatu proses bisnis dan mengetahui siapa yang bertanggung jawab pada proses bisnis tersebut. Dengan adanya informasi tersebut tentu perusahaan PT Chevron Pacific Indonesia dapat mengevaluasi proses tersebut dan memperbaiki proses pekerjaan untuk kedepannya.

Terkait dengan masalah sebelumnya dibutuhkan suatu pendekatan *workflow* yang berfungsi untuk membantu proses bisnis perusahaan yang berisi tentang prosedur dan alur proses pekerjaan yang sudah disepakati sebelumnya dalam suatu perusahaan. *Workflow* merupakan alur proses kerja yang di dalamnya dapat dilakukan pelacakan dan memberikan suatu tugas kepada partisipan dengan aturan tertentu (Chaffey, 1998). Pada *workflow* partisipan bisa berarti suatu manusia, aplikasi ataupun proses yang lain (Fischer, 2002). *Workflow* dapat membantu suatu perusahaan sehingga seluruh alur kerjanya dapat terlacak dan juga teratur sehingga bisa mengontrol suatu alur proses pekerjaan.

Dalam merepresentasikan sebuah *workflow* dapat menggunakan pendekatan BPMN. BPMN merupakan suatu standar untuk memodelkan suatu proses bisnis (Dewi, Indahyanti dan Hari, 2010). Dalam melakukan pemodelan proses bisnis menggunakan BPMN dapat membantu dan mengoptimalkan suatu *workflow* (Altuhhov, Matulevičius and Ahmed, 2013). Suatu *tools* untuk membuat BPMN seperti bizagi hanya bisa memodelkan suatu proses bisnis secara notasi namun tidak dapat melakukan penugasan terhadap karyawan yang bertanggung jawab dalam suatu proses itu sekaligus. Pada dasarnya BPMN hanya berfungsi sebagai pemodelan proses bisnis bukan sebagai sistem yang dapat melakukan penugasan terhadap karyawan, untuk melakukan hal tersebut diperlukan aplikasi lain yang berfungsi untuk melakukan penugasan.

Dari pendekatan sebelumnya penulis tertarik menyelesaikan masalah yang ada dengan pendekatan tersebut dalam mengimplementasikan sistem *workflow* manajemen kerja yang akan digunakan oleh PT Chevron Pacific Indonesia. Sistem *workflow* ini berorientasi pada kegiatan manajemen pekerjaan. Pada sistem ini dapat melakukan kegiatan seperti pembuatan proses kerja dengan *workflow* mengikuti aturan BPMN, penugasan karyawan, dan juga melakukan rekapitulasi suatu *workflow* dan proses bisnis. Sistem ini akan dikembangkan berbasis web. Diharapkan dengan adanya sistem ini mampu meningkatkan kinerja proses kerja dari PT Chevron Pacific Indonesia.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang mengenai penelitian ini, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimanakah hasil analisis kebutuhan sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia?
2. Bagaimanakah hasil perancangan sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia?
3. Bagaimanakah hasil implementasi sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia?
4. Bagaimanakah hasil pengujian sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia?

1.3 Tujuan

Berdasarkan pada rumusan masalah yang disebutkan, tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mengetahui hasil analisis kebutuhan sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia.
2. Mengetahui hasil perancangan sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia.
3. Mengetahui hasil implementasi sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia.
4. Mengetahui hasil pengujian dari sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini bagi PT. Chevron Pacific Indonesia yaitu dengan adanya sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia dapat membantu manajemen kerja seperti memberikan tugas, mengatur alur kerja, dan melakukan evaluasi alur kerja pada perusahaan.

1.5 Batasan masalah

Dari permasalahan yang telah dijelaskan sebelumnya, maka berikut ini diberikan batasan permasalahan tersebut, diantaranya:

1. Pada penelitian ini sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia digunakan untuk proses manajemen kerja pada PT. Chevron Pacific Indonesia.
2. Kebutuhan sistem didapatkan dari PT. Chevron Pacific Indonesia.
3. Pada aplikasi *workflow* manajemen kerja PT. Chevron Pacific ini digunakan hanya di *platform web* dengan bahasa *C#* pada *framework* ASP. NET MVC .

1.6 Sistematika Pembahasan

Pada sub bab sistematika pembahasan digunakan untuk memberikan gambaran secara umum yang akan dibahas sebagai berikut:

- BAB I : PENDAHULUAN**
Pada bab ini berisi latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika pembahasan *Workflow* manajemen kerja PT. Chevron Pacific Indonesia.
- BAB II : LANDASAN KEPUSTAKAAN**
Pada bab ini berisi teori atau penelitian yang sudah dilakukan sebelumnya yang berhubungan dengan permasalahan dalam penelitian tersebut.
- BAB III : METODE PENELITIAN**
Bab ini menjelaskan metode yang digunakan dan langkah kerja yang dilakukan dalam penulisan yang terdiri dari studi literatur, analisis kebutuhan, pengumpulan data, perancangan sistem, implementasi sistem, pengujian dan analisis, serta penulisan laporan
- BAB IV : ANALISIS KEBUTUHAN**
Pada bab ini berisi bagaimana analisis kebutuhan dilakukan oleh penulis pada sistem *Workflow* manajemen kerja PT. Chevron Pacific Indonesia.
- BAB V : PERANCANGAN DAN IMPLEMENTASI**
Pada bab ini berisi perancangan dari hasil analisis kebutuhan dan proses implementasi dari sistem *Workflow* manajemen kerja PT. Chevron Pacific Indonesia.
- BAB VII : PENGUJIAN**
Pada bab ini berisi proses uji dan hasil uji yang dihasilkan untuk memenuhi kriteria sistem *Workflow* manajemen kerja PT. Chevron Pacific Indonesia.
- BAB VIII : PENUTUP**
Pada bab ini berisi uraian kesimpulan dan saran dari hasil penelitian untuk digunakan pada pengembangan selanjutnya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka berfungsi sebagai referensi dalam penelitian yang dilakukan. Dalam hal ini kajian pustaka yang digunakan untuk referensi terkait dengan aplikasi *workflow* manajemen kerja PT. Chevron Pacific Indonesia. Terdapat beberapa penelitian tentang penggunaan manajemen *workflow* tetapi dari penelitian yang ada, hanya dua yang akan digunakan sebagai referensi. Dua penelitian tersebut akan dikaji agar dapat menganalisis kebutuhan sistemnya. Kajian pustaka pertama diperoleh dari Dulce Domingos et al. (2016) dan yang kedua diperoleh dari Azof Ghazali Sujono, Hanung Adi Nugroho, dan Sukiyo (2015).

Pada penelitian pertama, telah dilakukan penelitian sejenis yang masih relevan dengan topik tersebut. Penelitian tersebut dilakukan oleh Dulce Domingos et al pada tahun 2016 dengan penggunaan *resource reliability* pada proses BPMN. Kemudian pada penelitian tersebut, melakukan studi kasus terhadap dua *resource* yaitu sensor dan juga manusia. Penambahan *resource* tersebut dapat ditambahkan pada notasi BPMN, maka dengan adanya penambahan *resource* tersebut dapat dilakukan proses *assignment* dalam BPMN.

Penelitian kedua dilakukan oleh Azof Ghazali Sujono, Hanung Adi Nugroho, dan Sukiyo dengan membuat Sistem Informasi Administrasi Pelatihan dengan Manajemen *Workflow* pada tahun 2015. Pada penelitian tersebut dilakukan manajemen *workflow* dalam melakukan administrasi pelatihan mulai dari awal hingga akhir. Kemudian penelitian tersebut membuat kerangka kerja atau *workflow* dilakukan dengan pendekatan tabel. Dengan adanya *workflow* sistem tersebut dapat mengintegrasikan tugas yang terdapat pada administrasi pelatihan.

Dari kedua penelitian tersebut dapat disimpulkan bahwa pada penelitian pertama BPMN bisa memberikan notasi *resource* manusia sebagai penanggung jawab namun pada penelitian tersebut masih dalam bentuk notasi BPMN tidak dapat dilakukan penugasan secara langsung. Selanjutnya pada penelitian kedua didapati alur *workflow* atau proses kerja sistem tersebut direpresentasikan pada bentuk tabel. Oleh karena itu pada sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia, *workflow* akan direpresentasikan dalam bentuk BPMN. Kemudian pada BPMN tersebut dilakukan juga penggunaan *resource* dan dapat melakukan *assignment* pada sistem.

2.2 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak merupakan suatu ilmu atau prinsip tentang rekayasa yang memiliki hubungan dengan segala aspek diawali dari spesifikasi pembuatan perangkat lunak sampai *maintenance* sistem perangkat lunak (Sommerville, 2011). Dalam tahapan rekayasa perangkat lunak terdapat beberapa tahapan yaitu, analisis, desain, implementasi, pengujian, dan pemeliharaan dokumentasi (Sommerville, 2011). Dengan adanya proses pengembangan rekayasa perangkat

lunak diharapkan menghasilkan suatu perangkat dengan cara yang efektif dan efisien.

2.3 Pengembangan Perangkat Lunak

Pengembangan perangkat lunak merupakan suatu tahapan proses yang dikerjakan untuk mengembangkan perangkat lunak begitu pula pada sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia. Dalam pengembangan perangkat lunak perlu adanya ketelitian dalam merancang dan menjalankan dalam tujuan memenuhi tujuan yang ingin dituju. Ulasan perangkat lunak, seperti inspeksi, penelusuran, dan tinjauan teknis, merupakan salah satu pendekatan yang dipraktikkan secara luas untuk meningkatkan kualitas dari perangkat lunak tersebut (Sauer *et al.*, 2000). Ada beberapa bahasan tentang pengembangan perangkat lunak contohnya adalah model pengembangan perangkat lunak itu sendiri dengan menggunakan *platform* web dengan model *waterfall*, OOP (*Object Oriented Programming*), dan menggunakan konsep MVC (Model View Controller).

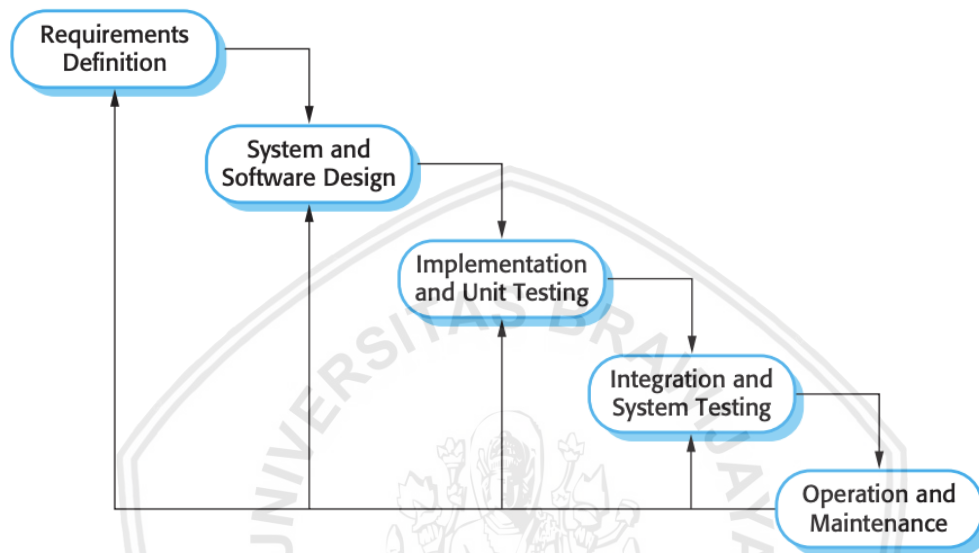
2.3.1 Model *Waterfall*

Pada sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia digunakan model *waterfall* karena sudah terdefiniskan secara rinci kebutuhannya di awal dan tidak berubah, sehingga metode *waterfall* akan cocok digunakan dalam pengembangan sistem ini. Terdapat beberapa tahapan dalam metode *waterfall* yang digambarkan pada Gambar 2.1. Dalam pengembangannya model *waterfall* memiliki tahapan sebagai berikut (Sommerville, 2011):

1. *Requirement Analysis*
Pada tahap ini mendefinisikan secara rinci layanan sistem, masalah, dan juga tujuan dari konsultasi pengguna sistem yang akan berfungsi sebagai spesifikasi sistem. Kemudian pada sistem yang akan dikembangkan akan digunakan *requirement analysis* dengan metode wawancara guna mendapatkan kebutuhan.
2. *Design*
Kemudian kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Desain sistem membantu dalam menentukan perangkat keras sistem persyaratan dan juga membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.
3. *Implementation dan Unit Testing*
Selanjutnya pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Pada tahap ini setiap unit dilakukan verifikasi sesuai spesifikasi sebelumnya.
4. *Integration dan Testing*
Seluruh unit yang dikembangkan dalam tahap implementasi diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Setelah integrasi seluruh sistem diuji untuk memeriksa tiap kegagalan maupun kesalahan.

5. *Operation dan Maintenance*

Tahap akhir dalam model *waterfall*. Perangkat lunak dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Pada penelitian ini pengembangan *waterfall* hanya dilakukan hingga tahap *Intergration dan Testing*.



Gambar 2.1 Metode *Waterfall*

(Sumber : Sommerville, 2011)

2.3.2 *Object Oriented Programming (OOP)*

OOP diperlukan pada pengembangan sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia dalam pembuatan *class*. Beberapa karakteristik yang terdapat dalam OOP adalah (Clark *et al.*, 2011):

a. *Object*

Objek adalah suatu struktur data untuk menggabungkan data dan prosedur untuk bekerja dengan data. Sebagai contoh, jika akan memiliki kemampuan berkendara, maka harus bekerja dengan objek mobil yang mewakili atas data dan metode yang digunakan dengan mobil tersebut.

b. *Inheritance*

Dalam OOP *inheritance* digunakan untuk mengklasifikasikan objek dalam program sesuai dengan karakteristik dan fungsi yang umum. Hal itu membuat bekerja dengan objek menjadi lebih mudah dan intuitif. Hal itu juga membuat program menjadi lebih mudah, karena memungkinkan pembuat untuk menggabungkan karakteristik umum

ke dalam objek *parent* dan mewarisi karakteristik tersebut di dalam objek *child*.

c. *Abstraction*

Abstraksi yaitu menyaring sifat benda asing sehingga hanya memproses sejumlah informasi dan konsentrasi penting pada tugas yang sedang dilakukan.

d. *Encapsulation*

Enkapsulasi adalah proses untuk mengatur suatu class yang didalamnya tidak terdapat akses langsung yang diberikan untuk data, namun tersembunyi. Pada enkapsulasi untuk mendapatkan akses ke data, diperlukan interaksi dengan objek yang bertanggung jawab untuk data tersebut.

e. *Aggregation*

Agregasi merupakan keadaan ketika sebuah objek terdiri dari gabungan dari benda-benda lain yang bekerja sama. Sebagai contoh, mesin pemotong rumput adalah gabungan dari objek roda, mesin, dan sebagainya. Objek mesin sendiri adalah gabungan dari berbagai objek. Kemampuan untuk menggunakan agregasi di dalam OOP merupakan fitur canggih yang memungkinkan implementasi proses bisnis dan model secara akurat di dalam suatu program.

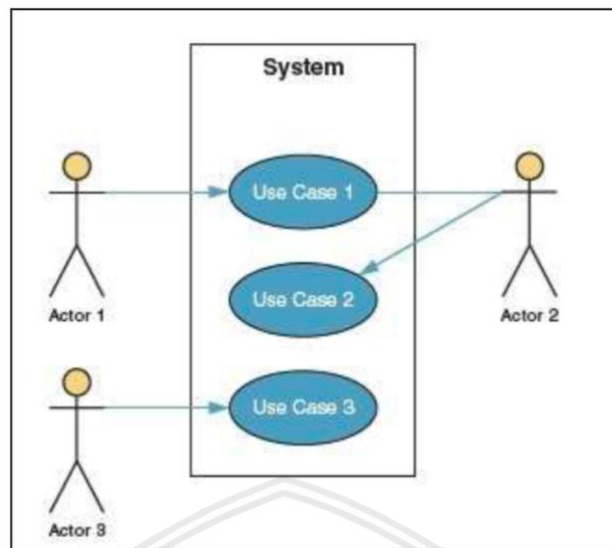
2.3.3 Unified Modeling Language (UML)

Dalam memodelkan spesifikasi sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia diperlukan adanya UML. UML adalah suatu konvensi pemodelan yang digunakan untuk menspesifikasikan atau mendeskripsikan sebuah sistem piranti lunak yang terkait dengan objek (Whitten & Bentley, 2007). UML sendiri terdiri dari beberapa tipe diagram, yaitu:

1. *Use Case Diagram*

Use case diagram adalah diagram yang menggambarkan interaksi antara sistem dan pengguna (Whitten & Bentley, 2007). Dengan kata lain, diagram ini mendeskripsikan siapa yang akan menggunakan sistem itu dengan cara apa pengguna berinteraksi dengan sistem.

- a. *Use Case* adalah urutan langkah-langkah yang secara tindakan saling terkait, baik terotomatisasi maupun secara manual untuk melengkapi suatu tugas bisnis tunggal yang digambarkan pada Gambar 2.2.



Gambar 2.2 Use Case Diagram

(Sumber : Whitten & Bentley, 2007)

Aktor adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi.

Hubungan (*Relationship*) pada diagram *use case* digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan hubungan berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Beberapa tipe hubungan yang terdapat pada diagram *use case*, yaitu sebagai berikut:

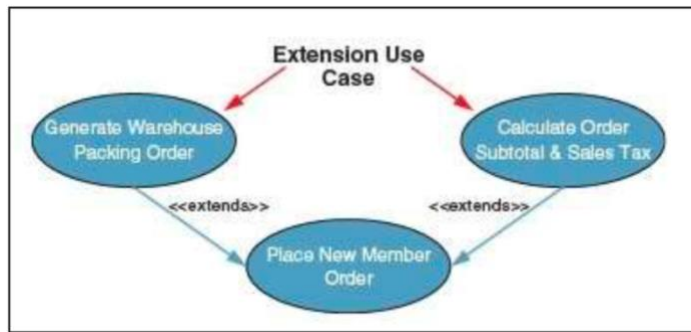
- 1) Gabungan (*Association*): hubungan antara pelaku dengan *use case* yang sama dan terjadi interaksi antara pelaku tersebut. Contoh dari *association* digambarkan pada gambar 2.3.



Gambar 2.3 Hubungan Association

(Sumber : Whitten & Bentley, 2007)

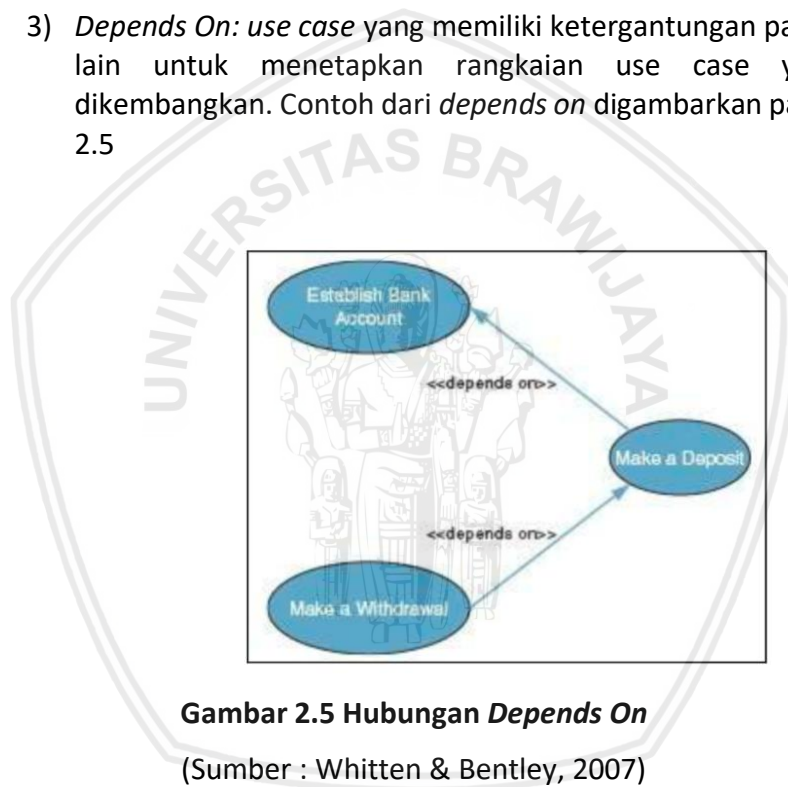
- 2) *Extends*: *use case* yang terdiri dari langkah yang diekstraksi dari *use case* yang lebih kompleks untuk menyederhanakan masalah orisinal dan karena itu memperluas fungsinya. Contoh dari *extends* digambarkan pada Gambar 2.4.



Gambar 2.4 Hubungan Extends

(Sumber : Whitten & Bentley, 2007)

- 3) *Depends On*: use case yang memiliki ketergantungan pada use case lain untuk menetapkan rangkaian use case yang perlu dikembangkan. Contoh dari *depends on* digambarkan pada Gambar 2.5



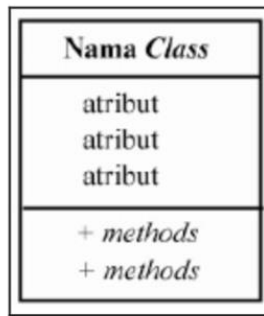
Gambar 2.5 Hubungan Depends On

(Sumber : Whitten & Bentley, 2007)

2. Class Diagram

Pada sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia perlu digambarkan *class diagram* untuk menggambarkan hubungan antar *class*. *Class diagram* menggambarkan struktur objek yang terdapat pada sistem (Whitten & Bentley, 2007). Diagram ini menunjukkan objek yang terdapat pada suatu sistem serta relasi antara objek tersebut.

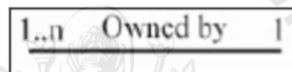
- 1) *Class* digambarkan sebagai sebuah kotak yang terbagi atas tiga bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan atribut *class*. Bagian akhir mendefinisikan fungsi dari sebuah *class*. Contoh dari *class* digambarkan pada gambar 2.6.



Gambar 2.6 Class

(Sumber : Whitten & Bentley, 2007)

- 2) *Association* adalah merupakan sebuah hubungan yang paling umum antara dua *class* dan dilambangkan oleh sebuah garis yang menghubungkan antara dua *class*. Garis ini bisa melambangkan tipe-tipe hubungan. Contohnya *one-to-one*, *one-to-many*, *many-to-many*. Contoh dari *association* digambarkan pada Gambar 2.7.



Gambar 2.7 Association

(Sumber : Whitten & Bentley, 2007)

- 3) *Composition* adalah jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *composition* terhadap *class* tempat *class* tersebut bergantung. Sebuah hubungan *composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang genjang berisi/solid. Contoh dari *composition* digambarkan pada Gambar 2.8.



Gambar 2.8 Composition

(Sumber : Whitten & Bentley, 2007)

- 4) *Agregation* adalah jika sebuah *class* bisa berdiri sendiri walaupun merupakan bagian dari *class* yang lain. Contoh dari *aggregation* digambarkan pada Gambar 2.9.



Gambar 2.9 Agregation

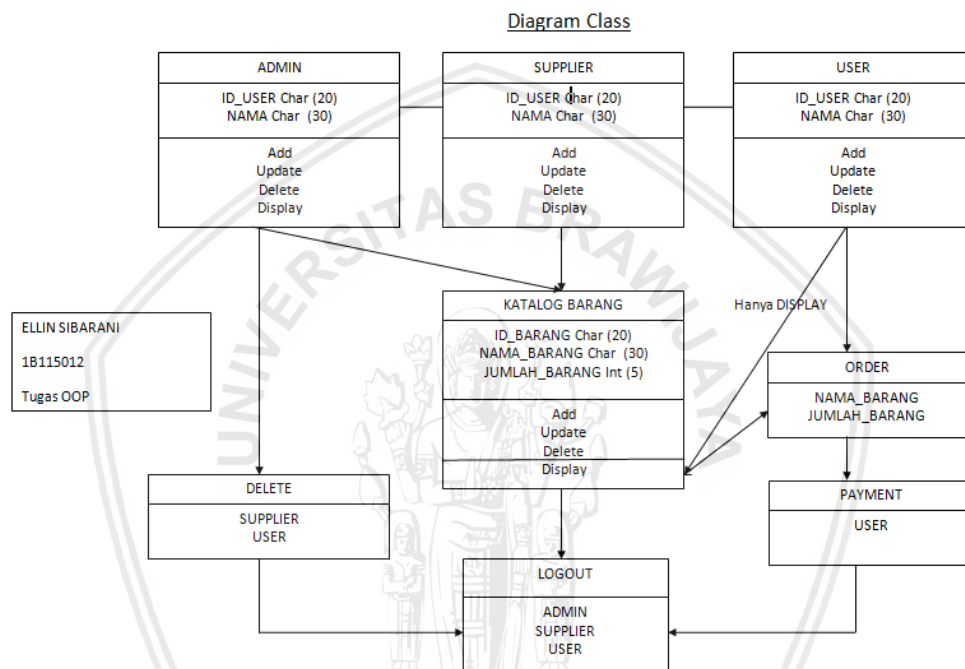
(Sumber : Whitten & Bentley, 2007)



5) Bagaimana cara mengakses *visibility* UML menyediakan tiga tingkat dari *visibility*, yaitu:

1. *Public* : ditandai dengan simbol “+”
2. *Protected* : ditandai dengan simbol “#”
3. *Private* : ditandai dengan simbol “-”

Sebagai contoh terdapat *class diagram* mengenai katalog barang yang digambarkan pada Gambar 2.10.



Gambar 2.10 Class Diagram
(Sumber : Whitten & Bentley, 2007)


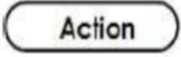


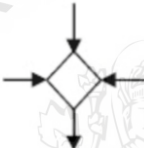

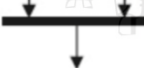

3. Activity Diagram

Activity diagram adalah sebuah diagram yang dapat digunakan untuk mendeskripsikan secara grafis aliran kerja dari sebuah proses bisnis, langkah-langkah dari sebuah *use case*, atau logika dari sebuah perilaku objek (*object behavior/method*) (Whitten & Bentley, 2007). *Activity diagram* merupakan representasi grafis dari aliran kerja (*workflow*) atas aktivitas-aktivitas yang bertahap dan aksi-aksi di dalam sebuah sistem.

Diagram ini juga mendukung proses seleksi, pengulangan (*iterasi*), dan konkurensi. *Activity diagram* sejenis dengan *state diagram/statechart diagram*. Diagram ini mendeskripsikan/menggambarkan kumpulan aktivitas dengan menunjukkan sekuensial dari aktivitas yang dilakukan. Diagram ini juga bisa menunjukkan aktivitas yang bersifat kondisional atau paralel. Berikut notasi yang digunakan pada *activity diagram* yang dijabarkan pada Tabel 2.1:



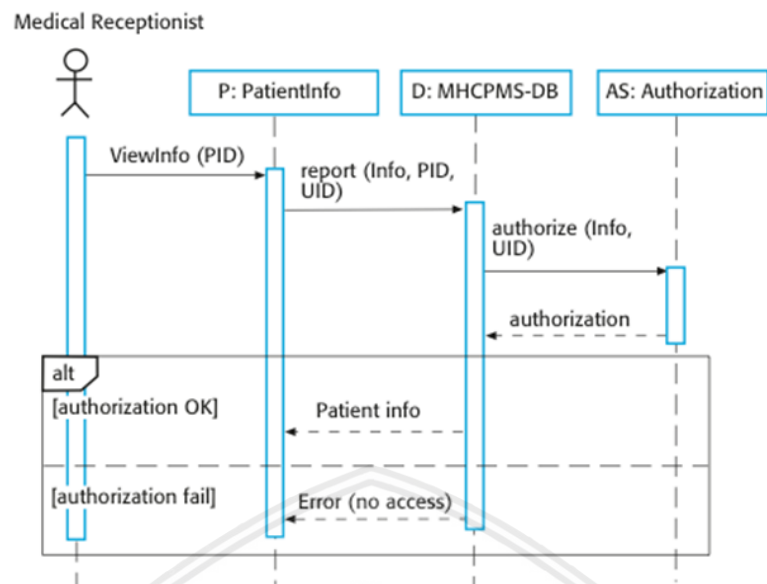
Tabel 2.1 Simbol *Activity Diagram*

No	Simbol	Gambar	Keterangan
1	<i>Initial Node</i>		Merupakan awal dari proses.
2	<i>Action</i>		Menunjukkan aktivitas yang terjadi.
3	<i>Flow</i>		Untuk berlanjut ke aksi yang lain dengan dilengkapi kata-kata bila berasal dari <i>decision</i> .
4	<i>Decision</i>		Memiliki satu buah <i>flow</i> masuk dan dua atau lebih <i>flow</i> keluar menunjukkan bahwa lebih dari satu kemungkinan terjadi.
5	<i>Merge</i>		Menyatukan <i>flow</i> yang sebelumnya terpisah oleh <i>decision</i> .
6	<i>Fork</i>		Menunjukkan aksi atau proses paralel yang dapat terjadi secara bersamaan.
7	<i>Join</i>		Semua aksi yang masuk ke <i>join</i> harus telah diselesaikan sebelum proses berlanjut.
8	<i>Final Activity</i>		Merupakan akhir dari proses.

4. *Sequence Diagram*

Sequence diagram menggambarkan bagaimana objek-objek berinteraksi satu sama lain melalui pesan dalam eksekusi dari sebuah *use case* atau operasi (Whitten & Bentley, 2007). Sebuah *sequence diagram* membantu untuk mengidentifikasi pesan tingkat tinggi yang masuk dan keluar dari sistem. Berikut ini merupakan contoh gambar dari *sequence diagram* yang ditujukan pada gambar 2.11.





Gambar 2.11 Sequence Diagram

(Sumber : Sommerville, 2011)

Beberapa notasi yang terdapat di dalam *sequence diagram*, yaitu:

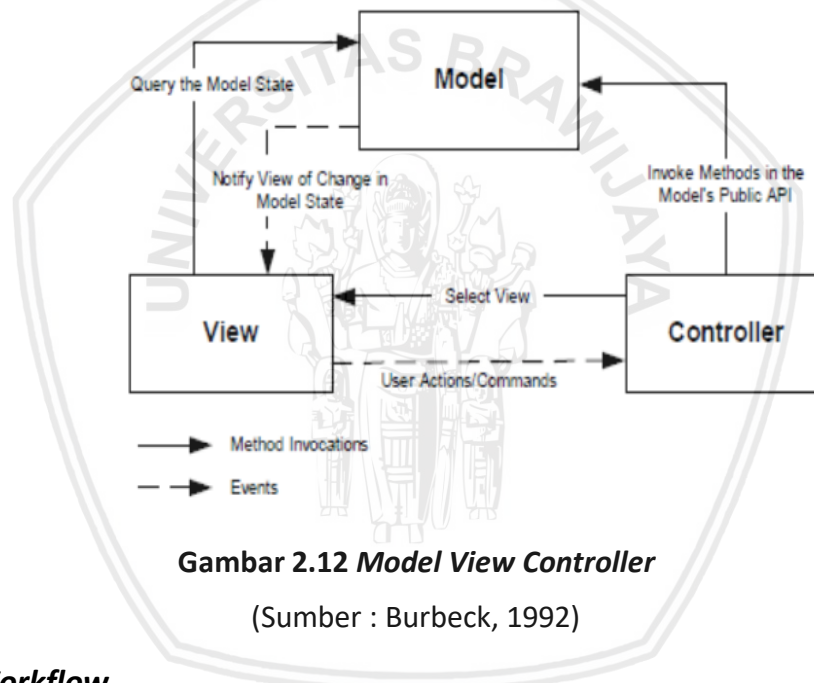
- Actor*, berupa aktor yang memulai pada *use case* ditunjukkan dengan simbol *use case* aktor.
- System*, berupa kotak yang menunjukkan sistem yang sedang berjalan, ditandai dengan notasi titik dua (:) pada awal nama sistem.
- Lifelines*, berupa garis putus-putus menurun dari simbol aktor dan sistem.
- Activation bars*, berupa balok yang terletak di atas *lifelines*, menunjukkan periode waktu ketika terdapat interaksi secara aktif.
- Input messages*, berupa anak panah mendarat dari aktor ke sistem yang menunjukkan pesan masuk dengan menulis kata pertama diawali dengan huruf kecil, sedangkan kata berikutnya diawali dengan huruf besar dan tidak ada spasi, diikuti dengan tanda kurung yang berisi parameter yang dibutuhkan.
- Output messages*, berupa anak panah mendarat dengan garis putus-putus dari sistem ke aktor yang menunjukkan pesan keluar.
- Receiver Actor*, berupa aktor lain yang menerima pesan dari sistem.
- Frame*, berupa kotak yang menambahkan pesan terpisah untuk menunjukkan perulangan, *alternative*, atau pilihan.

2.3.4 Model View Controller (MVC)

Dalam sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia dikembangkan dalam *platform* web yang membutuhkan MVC untuk memisahkan antara *model*, *view* dan *controller*. MVC merupakan konsep untuk mengenkapsulasi data dengan pemrosesan (*model*), yang diisolasi dari proses manipulasi (*controller*) dan pada tampilan (*view*) untuk ditampilkan pada suatu UI (*User Interface*) (Deacon, 2009). Lapisan pada MVC digambarkan pada gambar

2.12 yang menjelaskan hubungan antar lapisan. Arsitektur MVC dibagi menjadi tiga lapisan (Burbeck, 1992):

1. *Model*, berfungsi untuk memberitahu *controller* jika ada perubahan informasi. Pada model mengandung pemrosesan data yang terdiri dari data dan juga fungsi.
2. *View*, berfungsi untuk merepresentasikan grafis ke sebuah perangkat. Pada dasarnya *view* memiliki hubungan langsung ke layar. Jika model berubah maka *View* otomatis akan mengubah ulang tampilan yang terkena perubahan oleh model.
3. *Controller*, berfungsi untuk menerima masukan dari *user* sekaligus memberi instruksi kepada *view* dan juga model untuk melakukan tugasnya. Maka *controller* memiliki tugas untuk melakukan pemetaan aksi terhadap respon *user*.



2.3.5 Workflow

Workflow merupakan alur proses kerja yang dapat berfungsi untuk membantu suatu perusahaan ataupun individu sehingga seluruh alur kerjanya dapat terlacak dan juga teratur sehingga bisa mengatur suatu alur proses pekerjaan (Chaffey, 1998). *Workflow* juga merupakan otomatisasi suatu prosedur seperti tugas ataupun dokumen untuk bergerak sesuai aturan tertentu untuk mencapai tujuan bisnis. Dalam sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia *workflow* akan menggambarkan alur dari suatu proses pekerjaan. Selanjutnya *workflow* tersebut akan digambarkan dengan pemodelan BPMN.

2.3.6 Business Process Modelling Notation (BPMN)

BPMN merupakan suatu standar dalam pemodelan suatu proses bisnis. BPMN selain digunakan untuk mudah dipahami namun juga memiliki kelebihan dalam

pemodelan proses bisnis yang kompleks. Pada BPMN terdapat notasi yang disediakan (Dewi, Indahyanti dan Hari, 2010) yaitu:

- *Event* yaitu menggambarkan sesuatu yang terjadi pada suatu proses bisnis. Terdapat tiga jenis yaitu *start*, *intermediate*, dan *end* yang digambarkan pada gambar 2.13.



Gambar 2.13 Event BPMN

(Sumber : Dewi, Indahyanti dan Hari, 2010)

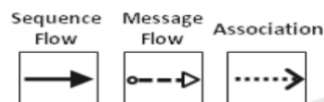
- *Activity* merupakan suatu aktivitas yang dapat berdiri secara sendiri ataupun gabungan jika bukan merupakan suatu *task* atau merupakan *sub process* maka akan diberikan tanda + pada bagian bawah tengah. *Gateway* merupakan suatu proses keputusan yang menentukan suatu kondisi tertentu pada aktivitas dari suatu hasil proses. Gambar *activity* dan *gateway* digambarkan pada gambar 2.14.



Gambar 2.14 Activity dan Gateway BPMN

(Sumber : Dewi, Indahyanti dan Hari, 2010)

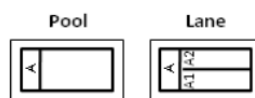
- *Sequence Flow* merupakan alur menunjukkan kegiatan yang akan dilakukan. *Message Flow* merupakan alur pesan antara penerima dan pengirim dua entitas. Asosiasi merupakan asosiasi data dan juga informasi. Gambar dari *sequence flow*, *message flow*, dan asosiasi digambarkan pada gambar 2.15.



Gambar 2.15 Sequence BPMN

(Sumber : Dewi, Indahyanti dan Hari, 2010)

- Terdapat pula *Swimlanes* pada BPMN yaitu terdapat *pool* yang mewakili proses dan *lane* merupakan bagian sub dari *pool* dan berguna untuk mengkategorikan dan mengatur suatu aktivitas. Gambar dari *pool* dan juga *lane* digambarkan pada gambar 2.16.



Gambar 2.16 Swimlanes BPMN

(Sumber : Dewi, Indahyanti dan Hari, 2010)

2.4 Teknologi Pengembangan Sistem

2.4.1 GoJs Diagram

Pada pengembangan sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia menggunakan library Gojs untuk menampilkan *workflow* BPMN. Gojs merupakan suatu *library* javascript yang berfungsi membuat diagram secara interaktif sehingga dapat mempermudah *user* dalam pembuatan diagram agar menjadi lebih interaktif (Shahzad *et al.*, 2016).

Pada GoJs terdapat *canvas* yang di dalamnya terdapat proses dalam bentuk *nodes* dan *links* yang menghubungkan *nodes* tersebut. Dalam menyimpan suatu *canvas* besar beserta *nodes* dan juga *links*-nya disimpan dalam bentuk *JavaScript Object Notation* (JSON). Pada diagram GoJs dapat dilakukan *resizing*, pewarnaan dan melakukan *drag and drop*.

2.4.2 Kendo UI

Dalam pengembangan sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia terdapat rekapitulasi *workflow* yang akan direpresentasikan dalam bentuk grafik dan juga membutuhkan tabel untuk menampilkan daftar yang dibutuhkan. Dalam merepresentasikan grafik dan tabel tersebut dibutuhkan *library* yaitu Kendo UI. Kendo UI merupakan suatu *framework library* yang berfungsi untuk data visualisasi secara *client side* dalam tampilan yang interaktif (Bhandari & Nair, 2013). Pada kendo UI ini sendiri dapat diimplementasikan dalam bentuk MVC ASP .NET. Pada Kendo UI sangat cocok untuk bisnis tingkat perusahaan untuk membuat *website* profesional yang memerlukan dukungan teknis dan tepat waktu.

2.4.3 .NET Framework

Pada pengembangan sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia dalam implementasinya menggunakan *framework* .NET. *Framework* .NET merupakan produk dari *Microsoft* yang bertujuan untuk membantu melakukan pengembangan pada pemrograman *web* ASP.NET yang memiliki kecocokan arsitektur yang digunakan pada perusahaan besar (Grimmer, 2004). Disini .NET berfungsi untuk membangun, merilis, dan menjalankan *web service* dan aplikasi. Pada *framework* .NET terdapat pengaksesan data subsistem atau mengambil data ke dalam *database* dengan menggunakan ADO.NET.

2.4.4 Entity Framework

Pada pengembangan sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia dalam pengaksesan *database* membutuhkan *entity framework* karena sebelumnya menggunakan *framework* .NET. *Entity Framework* adalah *Object Relational Framework* untuk .NET yang merupakan produk dari *Microsoft* yang dalam pengaksesan *database* menggunakan *library* ADO.NET (Grimmer, 2004). *Entity framework* memudahkan penggunaannya untuk menggunakan objek dari *class* tanpa harus fokus terhadap tabel dan kolom yang

ada pada *database* yang menyimpan data tersebut. Adanya *entity framework* memudahkan pengguna untuk berurusan dengan data dengan menggunakan kode yang lebih sedikit dibandingkan aplikasi tradisional.

2.4.5 Microsoft SQL Server

Dalam pengembangan sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia DBMS yang digunakan adalah Microsoft SQL Server. Microsoft SQL Server merupakan produk *desktop server side* yang memiliki fungsi untuk melakukan process *Create Read Update Delete* (CRUD) pada *database* (Mistry, 2012). Pada penggunaannya manajemen operasinya dilakukan oleh *administrator*. Pada SQL server ini *administrator* hanya bisa melakukan pada *device* yang sudah terdapat Microsoft SQL Server. Pada Microsoft SQL Server memiliki kelebihan seperti pengendalian *database* yang terpusat, manajemen *password* yang baik aman serta mudah digunakan baik untuk *client* ataupun *user*.

2.4.6 Pengujian Perangkat Lunak

Pada pengujian perangkat lunak berfungsi sebagai prosedur untuk mencari kesalahan atau *error* pada suatu perangkat lunak. Pengujian perangkat lunak bertujuan untuk melakukan pengujian kemampuan dari perangkat lunak dan memvalidasi bahwa perangkat lunak memenuhi hasil yang diharapkan (Rouf, 2012). Dengan adanya pengujian perangkat lunak dalam perangkat lunak akan ditemukannya kekurangan dan kesalahan sistem maka *developer* akan mengetahui apa yang akan dilakukan selanjutnya.

2.4.6.1 White Box Testing

White box testing merupakan tata acara pengujian untuk menganalisis kesalahan pada kode program yang ada. Pada *white box testing* dilakukan agar mengetahui secara internal cara kerja kode program dan dipastikan bahwa telah berjalan sesuai kebutuhan sebelumnya. *White box testing* adalah bagian dari suatu *component level design* yang akan menghasilkan sebuah *test cases*. Dalam pengujian *white box* akan dilakukan pengujian terhadap masukan dan keluaran dari program tersebut. Pada penelitian ini metode yang digunakan adalah *basis path testing* yang digunakan untuk memperoleh *independent path* yang berfungsi sebagai pengecekan jalur pada algoritme dan menggunakan *cyclomatic complexity* yang berfungsi untuk memeriksa tingkat kompleksitas pada sistem.

2.4.6.2 Black Box Testing

Black box testing merupakan pengujian untuk melihat kepada kebutuhan perangkat lunak dari sistem yang disebut juga dengan *behavioral testing* (Pressman, 2010). *Black box testing* berfokus kepada fungsionalitas sistem yang telah didefinisikan sebelumnya. Pada *black box testing* biasa ditemukan hal seperti fungsi yang tidak ada, kesalahan pada *interface*, kesalahan pada struktur data, kesalahan *performance*, dan kesalahan inialisasi terminasi (Pressman, 2010). Tahapan *black box testing* dilakukan setelah *white box testing*. Pada penelitian ini metode yang digunakan adalah *equivalence partitioning* yang akan membagi

masuk dari sistem ke dalam kelas data dan kemudian *test cases* dapat didapatkan. Selanjutnya masukkan dan keluaran dipartisi ke dalam kelas-kelas sesuai dengan spesifikasi komponen tersebut yang akan diperlakukan sama dan dikatakan berhasil bila masukan akan menghasilkan keluaran yang diharapkan.

2.4.6.3 Integration Testing

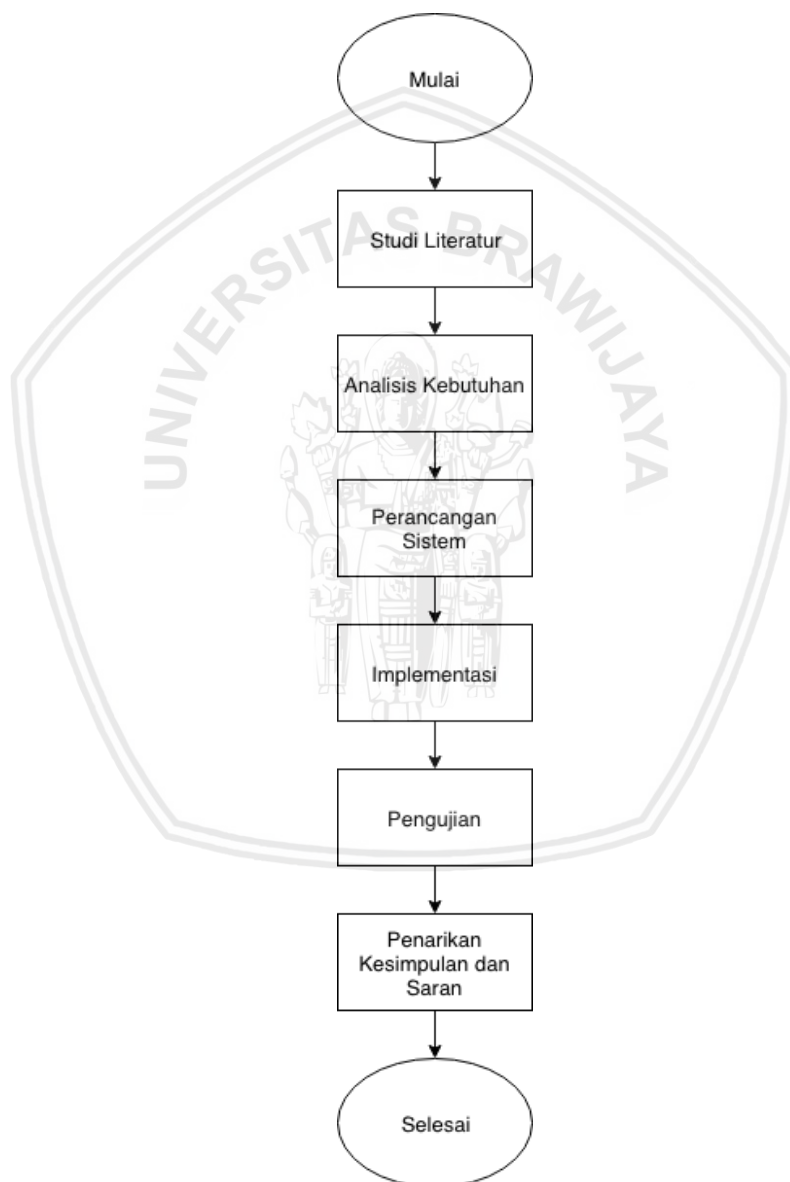
Integration testing merupakan pengujian yang bertujuan untuk menentukan suatu kecacatan sistem dari suatu unit ataupun komponen yang jika ditemukan akan diperbaiki kemudian (Orellana, 2017). Terdapat dua metode yang bisa digunakan pada saat melakukan *integration testing* yaitu pengujian *Bottom Up*, dan *Top Down*. Pengujian *Bottom Up* yaitu pengujian integrasi yang dilakukan dari komponen paling bawah hingga ke level atas dan jika didapati level paling atas belum selesai maka dapat melakukan simulasi komponen level atas tersebut dengan menggunakan *test driver*. Selanjutnya untuk pengujian *Top Down* yaitu pengujian integrasi yang dilakukan dari komponen paling atas hingga ke level bawah dan jika didapati level bawah belum selesai maka dapat melakukan simulasi komponen level bawah tersebut dengan menggunakan *test stub*. Dalam penelitian ini *integration testing* akan dilakukan dengan metode *Bottom Up*.

2.4.6.4 Usability Testing

Usability testing merupakan pengujian yang dilakukan untuk mengetahui kemudahan dalam penggunaan sistem. Pengujian ini dilakukan ketika sistem telah siap digunakan pengguna dan setelah diujikan oleh pengguna diketahui bagaimana tingkat dari kemudahan sistem. Pada pengujian ini digunakan metode *System Usability Scale (SUS)* yaitu pengujian yang melibatkan responden dengan skala kegunaan yang andal dan berbiaya rendah yang dapat digunakan untuk penilaian umum kegunaan sistem (Brooke, 2011). Saat menggunakan SUS ini pengguna atau responden akan diberikan sepuluh pertanyaan sederhana terkait penggunaan sistem. Selanjutnya dari sepuluh pertanyaan tersebut dijawab dengan ukuran terkecil sangat tidak setuju sampai dengan sangat setuju yang dimulai dari skala satu sampai dengan lima. Terakhir jika sudah terjawab seluruhnya maka akan diberi penilaian secara keseluruhan dari nilai nol sampai dengan seratus.

BAB 3 METODOLOGI

Pada bab ini berisi tentang metode penelitian yang digunakan dalam proses pengembangan aplikasi *workflow* manajemen kerja PT. Chevron Pacific Indonesia. Metodologi yang digunakan dalam penelitian ini melalui beberapa tahapan diantaranya studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pengambilan kesimpulan dan saran yang digambarkan pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian

3.1 Studi Literatur

Studi literatur merupakan penjelasan dari dasar teori yang digunakan dalam penelitian yang pernah dilakukan sebelumnya. Berikut ini merupakan teori – teori pendukung yang digunakan pada penelitian yang akan dilakukan yaitu *Waterfall*, *Object Oriented Programming* (OOP), *Model View Controller* (MVC), GoJs, Kendo UI, *NET Framework*, *Entity Framework*, dan Microsoft SQL Server.

3.2 Analisis Kebutuhan

Analisis kebutuhan berfungsi untuk mengidentifikasi kebutuhan yang dibutuhkan oleh aplikasi *workflow* manajemen kerja PT Chevron Pacific Indonesia. Tahapan yang dilakukan melalui elisitasi kebutuhan yaitu dengan teknik wawancara untuk mengumpulkan data-data yang terkait dengan PT Chevron Pacific Indonesia. Dengan adanya hasil wawancara tersebut maka didapatkan gambaran umum mengenai sistem seperti fitur. Selanjutnya dilakukan spesifikasi kebutuhan dengan mendapatkan jenis aktor sekaligus yang dilakukan aktor tersebut dan didapatkan kebutuhan fungsional dan non fungsional. Kemudian setelah spesifikasi kebutuhan dilakukan manajemen kebutuhan yaitu dari hasil fungsional dan non fungsional sebelumnya ditulis dengan kode yang merepresentasikan kebutuhan fungsional, non fungsional dan urutannya. Terakhir dilakukan pemodelan kebutuhan yaitu identifikasi dari suatu aktor pada sistem yang terdapat pada *use case diagram*, dan *use case scenario*.

3.3 Perancangan Sistem

Pada tahap perancangan sistem dilakukan ketika seluruh kebutuhan dari sistem telah didefinisikan dengan jelas sehingga proses perancangan sistem dapat dilakukan sesuai kebutuhan pengguna. Setelah didapati kebutuhan, maka selanjutnya kebutuhan sistem akan dimodelkan dalam bentuk UML (*Unified Modeling Language*). Pada UML ini terdapat perancangan, *sequence diagram*, *class diagram*, perancangan komponen, basis data, dan perancangan antarmuka.

3.4 Implementasi Sistem

Pada Implementasi sistem adalah fase pembangunan sistem yang mengacu pada hasil dari perancangan sistem. Implementasi sistem ini dibangun dengan menggunakan *platform web* dengan framework .NET. Proses-proses yang ada dalam implementasi sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia ini antara lain adalah:

1. Implementasi sistem dilakukan dengan mengacu pada spesifikasi sistem yang digunakan untuk membangun aplikasi ini.
2. Implementasi *database* dilakukan berdasarkan data yang telah dianalisis pada proses analisis data.
3. Implementasi kode program mengacu pada perancangan komponen yang memuat algoritme pada perancangan sistem dan juga perancangan antarmuka pada proses perancangan sistem.

3.5 Pengujian

Pada saat pengujian dilakukan untuk menemukan apakah terdapat kesalahan pada perangkat lunak dan menguji kebutuhan yang didefinisikan sebelumnya. Ketika pengujian dilakukan empat metode yaitu *black box testing*, *white box testing*, *integration testing*, dan pada pengujian dilakukan juga pengujian *usability testing* untuk mengetahui kemudahan dalam menggunakan sistem. Kemudian selanjutnya jika terdapat kesalahan pada perangkat lunak maka akan dilakukan perbaikan selanjutnya.

3.6 Kesimpulan dan Saran

Proses pengambilan kesimpulan dilakukan setelah semua tahapan pengembangan perangkat lunak telah selesai dilakukan. Kesimpulan didapatkan dari hasil pengujian dan analisis terhadap aplikasi *workflow* manajemen kerja PT. Chevron Pacific Indonesia. Saran ditulis untuk memperbaiki kesalahan-kesalahan yang terjadi dan menyempurnakan penulisan dan juga untuk memberikan pertimbangan untuk melakukan pengembangan lebih lanjut terhadap perangkat lunak yang telah dibuat.



BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan adalah tahapan awal dalam suatu pembangunan sistem. Analisis kebutuhan berfungsi untuk menentukan kebutuhan yang ada pada *workflow* manajemen kerja PT Chevron Pacific Indonesia. Kebutuhan ini diperoleh dari studi lapangan di PT Chevron Pacific Indonesia dengan teknik wawancara. Disini analisis kebutuhan menjelaskan aktor-aktor yang terlibat dalam sistem, yang dapat dilakukan oleh aktor-aktor, dan interaksi antara aktor dengan sistem.

4.1 Deskripsi Sistem

Sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia adalah sistem yang bertujuan membantu dalam melakukan manajemen kerja dengan cara membuat suatu alur *workflow* yang berorientasikan BPMN. Pada sistem ini dibangun berbasis web agar mudah diakses oleh penggunanya. Sistem ini memiliki beberapa fungsi seperti membuat *workflow* kemudian dalam *workflow* tersebut dapat digambarkan suatu *template* alur proses pekerjaan berikut dengan deskripsi tiap proses pekerjaan seperti nama pekerjaan, tenggat waktu dan memilih penanggung jawab proses pekerjaan. Selanjutnya *workflow* dapat mulai dilaksanakan dan di selesaikan oleh setiap orang yang bertanggung jawab. Jika *workflow* sudah selesai maka dapat dilihat laporan dari *workflow* tersebut.

4.2 Identifikasi Aktor

Dalam identifikasi aktor dijelaskan bagaimana peran dari aktor-aktor yang terlibat dalam sistem. Aktor disini didapatkan dari hasil wawancara ketika studi lapangan. Berdasarkan hasil wawancara, dalam sistem ini didapati empat aktor yaitu Pengunjung, *Admin*, *Workflow Manager*, dan *Person in Charge*. Identifikasi dari aktor dijelaskan pada Tabel 4.1 yang terdiri dari Nomor, nama aktor, dan deskripsi dari masing-masing aktor.

Tabel 4.1 Identifikasi Aktor Sistem

No	Nama Aktor	Deskripsi
1	Pengunjung	Pengunjung merupakan aktor yang dapat mengakses halaman <i>web</i> awal untuk melakukan <i>login</i> dan juga melakukan registrasi ke dalam sistem.
2	<i>Admin</i>	<i>Admin</i> merupakan pengguna yang dapat membuat <i>template workflow</i> dengan notasi BPMN yang kemudian di dalamnya bisa membuat proses-proses dan saling menghubungkan, kemudian admin juga dapat memberikan tenggat waktu tiap prosesnya. <i>Admin</i> juga dapat melihat seluruh <i>template workflow</i> yang telah dibuat dan juga bisa melihat <i>template workflow</i> yang sudah

Tabel 4.1 Identifikasi Aktor Sistem (lanjutan)

No	Nama Aktor	Deskripsi
		diimplementasikan oleh <i>Workflow Manager</i> dan di dalamnya dapat melihat seluruh status detail dan rekapitulasi dari seluruh <i>workflow</i> yang ada. Kemudian <i>admin</i> juga dapat melihat seluruh pengguna yang ada pada sistem.
3	<i>Workflow Manager</i>	<i>Workflow Manager</i> merupakan pengguna yang dapat memilih <i>template workflow</i> yang sudah dibuat sebelumnya oleh admin lalu menentukan subjek dari suatu <i>workflow</i> dan menggunakannya, kemudian di dalamnya dapat menentukan <i>person in charge</i> yang bertanggung jawab, memulai suatu <i>workflow</i> dan dapat melihat status detail dan rekapitulasi dari <i>workflow</i> yang dimiliki.
4	<i>Person in Charge</i>	<i>Person in Charge</i> merupakan pengguna yang menerima notifikasi dari sistem setelah suatu <i>workflow</i> dimulai dan didalam <i>workflow</i> tersebut sudah terpilih oleh <i>workflow manager</i> untuk bertanggung jawab dalam suatu proses. Kemudian <i>person in charge</i> dapat melakukan konfirmasi jika proses yang didapati sudah selesai atau belum.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus terdapat pada sistem yang dapat diakses oleh pengguna. Berdasarkan hasil elisitasi kebutuhan yang dilakukan sebelumnya diperoleh kebutuhan fungsional yang dibutuhkan pengguna. Pada kebutuhan fungsional didefinisikan dalam kode WMK_F_XX. WMK merupakan singkatan dari *workflow* manajemen kerja, F merupakan singkatan dari fungsional, dan XX merupakan representasi dari kode urutan kebutuhan fungsional. Kebutuhan fungsional dari sistem ini terdapat 84 buah kebutuhan yang dijelaskan dalam bentuk Tabel 4.2 yang terdiri dari nomor, kode, *use case*, aktor, dan deksripsi kebutuhan.

Tabel 4.2 Kebutuhan Fungsional

No	Kode	<i>Use Case</i>	Aktor	Deskripsi Kebutuhan
1	WMK_F_1	<i>Login</i>	Pengunjung	Sistem dapat menyediakan fungsi <i>login</i> kepada pengunjung sehingga bisa mengakses sistem baik

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				sebagai admin, <i>workflow manager</i> , dan <i>person in charge</i> dengan mengisikan <i>email</i> dan <i>password</i> .
2	WMK_F_2	<i>Register</i>	Pengunjung	Sistem dapat menyediakan fungsi <i>register</i> bagi pengunjung dengan mengisikan <i>full name</i> , <i>department</i> , <i>role</i> , <i>email</i> , <i>password</i> , dan <i>password confirmation</i> .
3	WMK_F_3	Buat <i>Workflow</i>	Admin	Admin dapat membuat <i>template workflow</i> dengan <i>tools</i> yang ditampilkan oleh sistem yang akan disimpan dengan memberikan nama <i>workflow</i> .
4	WMK_F_4	Lihat Petunjuk Cepat	Admin	Admin dapat melihat petunjuk cepat cara membuat <i>workflow</i> .
5	WMK_F_5	Pilih Tenggat Waktu	Admin	Admin dapat menentukan tenggat waktu suatu tugas dalam <i>workflow</i> .
6	WMK_F_6	<i>Undo</i> Objek	Admin	Admin dapat melakukan <i>undo</i> perubahan pada saat pembuatan <i>workflow</i> .
7	WMK_F_7	<i>Redo</i> Objek	Admin	Admin dapat melakukan <i>redo</i> perubahan pada saat pembuatan <i>workflow</i> .
8	WMK_F_8	<i>Cut</i> Objek	Admin	Admin dapat melakukan <i>cut</i> pada objek pada saat pembuatan <i>workflow</i> .
9	WMK_F_9	<i>Copy</i> Objek	Admin	Admin dapat melakukan <i>copy</i> pada objek pada saat pembuatan <i>workflow</i> .
10	WMK_F_10	<i>Paste</i> Objek	Admin	Admin dapat melakukan <i>paste</i> pada objek pada

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				saat pembuatan <i>workflow</i> .
11	WMK_F_11	Delete Objek	Admin	Admin dapat melakukan <i>delete</i> pada objek pada saat pembuatan <i>workflow</i> .
12	WMK_F_12	Pilih Seluruh Objek	Admin	Admin dapat memilih seluruh objek sekaligus pada saat pembuatan <i>workflow</i> .
13	WMK_F_13	Pindah Posisi Objek Kiri	Admin	Admin dapat melakukan pemindahan posisi objek di sisi kiri pada saat pembuatan <i>workflow</i> .
14	WMK_F_14	Pindah Posisi Objek Kanan	Admin	Admin dapat melakukan pemindahan posisi objek di sisi kanan pada saat pembuatan <i>workflow</i> .
15	WMK_F_15	Pindah Posisi Objek Atas	Admin	Admin dapat melakukan pemindahan posisi objek di sisi atas pada saat pembuatan <i>workflow</i> .
16	WMK_F_16	Pindah Posisi Objek Bawah	Admin	Admin dapat melakukan pemindahan posisi objek di sisi bawah pada saat pembuatan <i>workflow</i> .
17	WMK_F_17	Tampil Kisi	Admin	Admin dapat menampilkan kisi pada saat pembuatan <i>workflow</i> .
18	WMK_F_18	Lihat Jumlah <i>Template Workflow</i>	Admin	Admin dapat melihat jumlah daftar <i>template workflow</i> .
19	WMK_F_19	Lihat Jumlah <i>Implemented Workflow</i>	Admin	Admin dapat melihat jumlah daftar <i>implemented workflow</i> .
20	WMK_F_20	Lihat Daftar <i>Template Workflow</i>	Admin	Admin dapat melihat daftar <i>template workflow</i> yang telah dibuat oleh admin sebelumnya berdasarkan <i>workflow</i>

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				<i>name, task name, dan deadline</i> dalam bentuk tabel.
21	WMK_F_21	Cari <i>Template Workflow</i>	Admin	Admin dapat melakukan pencarian <i>template workflow</i> berdasarkan <i>workflow name</i> .
22	WMK_F_22	Lihat <i>Detail Template Workflow</i>	Admin	Admin dapat melihat <i>detail</i> daftar <i>template workflow</i> yang telah dibuat oleh admin sebelumnya dalam bentuk <i>workflow</i> .
23	WMK_F_23	Lihat <i>Implemented Workflow</i>	Admin	Admin dapat melihat daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> berdasarkan <i>workflow name, task name, deadline, dan status</i> dalam bentuk tabel.
24	WMK_F_24	Cari <i>Implemented Workflow</i>	Admin	Admin dapat melakukan pencarian <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> berdasarkan <i>workflow name</i> .
25	WMK_F_25	Lihat <i>Detail Daftar Implemented Workflow</i>	Admin	Admin dapat melihat <i>detail</i> daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> dan sistem menampilkannya dalam bentuk <i>workflow</i> .
26	WMK_F_26	Lihat <i>Pie Chart Persentase Workflow</i>	Admin	Admin dapat melihat <i>pie chart</i> mengenai persentase penggunaan <i>workflow</i> baik <i>workflow</i> yang telah selesai maupun

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				<i>workflow</i> yang sedang berjalan.
27	WMK_F_27	Lihat Jumlah <i>Finished Workflow</i>	Admin	Admin dapat melihat jumlah <i>workflow</i> yang telah selesai.
28	WMK_F_28	Lihat Daftar <i>Finished Workflow</i>	Admin	Admin dapat melihat daftar <i>workflow</i> yang telah selesai berdasarkan <i>workflow name, task name, workflow manager, pic, deadline, start task</i> , dan <i>finish task</i> dalam bentuk tabel.
29	WMK_F_29	Cari <i>Finished Workflow</i>	Admin	Admin dapat melakukan pencarian <i>workflow</i> yang telah selesai berdasarkan <i>workflow name</i> .
30	WMK_F_30	Lihat <i>Detail Finished Workflow</i>	Admin	Admin dapat melihat <i>detail</i> daftar <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>workflow</i> .
31	WMK_F_31	Lihat Statistik <i>Finished Workflow</i>	Admin	Admin dapat melihat statistik <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
32	WMK_F_32	Lihat Jumlah <i>Workflow By Template</i>	Admin	Admin dapat melihat jumlah <i>workflow</i> berdasarkan jenis <i>template workflow</i> yang ada.
33	WMK_F_33	Lihat Daftar <i>Workflow By Template</i>	Admin	Admin dapat melihat daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> dalam bentuk tabel.

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
34	WMK_F_34	Cari <i>Workflow By Template</i>	Admin	Admin dapat melakukan pencarian <i>template workflow</i> yang telah selesai berdasarkan <i>template workflow name</i> .
35	WMK_F_35	Lihat Statistik <i>Workflow By Template</i>	Admin	Admin dapat melihat statistik rata-rata <i>template workflow</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
36	WMK_F_36	Lihat Jumlah <i>Pending User</i>	Admin	Admin dapat melihat jumlah <i>pending users</i> yang ada pada sistem.
37	WMK_F_37	Lihat Jumlah <i>All User</i>	Admin	Admin dapat melihat jumlah <i>all users</i> yang ada pada sistem.
38	WMK_F_38	Lihat Daftar <i>Pending Users</i>	Admin	Admin dapat melihat daftar <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name</i> , <i>email</i> , <i>role</i> , dan <i>department</i> dalam bentuk tabel.
39	WMK_F_39	Cari <i>Pending User</i>	Admin	Admin dapat melakukan pencarian daftar <i>pending user</i> berdasarkan <i>full name</i> , <i>email</i> , <i>role</i> , dan <i>department</i> .
40	WMK_F_40	Verifikasi <i>Pending User</i>	Admin	Admin dapat melakukan verifikasi akun <i>user</i> yang mendaftar.
41	WMK_F_41	Tolak <i>Pending User</i>	Admin	Admin dapat menolak akun <i>user</i> yang mendaftar.
42	WMK_F_42	Lihat Daftar <i>All User</i>	Admin	Admin dapat melihat daftar seluruh <i>user</i> berdasarkan <i>full name</i> ,

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				<i>email, role, dan department.</i>
43	WMK_F_43	Cari <i>All User</i>	Admin	Admin dapat melakukan pencarian daftar seluruh <i>user</i> berdasarkan <i>full name, email, role, dan department.</i>
44	WMK_F_44	Lihat <i>Profile Admin</i>	Admin	Admin dapat melihat <i>profile</i> berdasarkan <i>full name, department, role, dan email.</i>
45	WMK_F_45	Ubah <i>Password Admin</i>	Admin	Admin dapat mengubah <i>password</i> untuk masuk ke dalam sistem dengan memasukkan <i>password</i> baru dan konfirmasi <i>password</i> baru.
46	WMK_F_46	Lihat <i>Template Workflow Admin</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat daftar <i>template workflow</i> yang sudah dibuat oleh admin sebelumnya berdasarkan <i>workflow name, task name, dan deadlines</i> dalam bentuk tabel.
47	WMK_F_47	Cari <i>Template Workflow Admin</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melakukan pencarian <i>template workflow</i> yang sudah dibuat oleh admin sebelumnya berdasarkan <i>workflow name.</i>
48	WMK_F_48	Pilih <i>Template Workflow Admin</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat memilih <i>template workflow</i> kemudian sistem akan menampilkan <i>workflow</i> yang sudah dipilih.

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode		Aktor	Deskripsi Kebutuhan
49	WMK_F_49	Mulai <i>Template Workflow Admin</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat memulai <i>template workflow</i> yang sudah dipilih oleh <i>workflow manager</i> itu sendiri.
50	WMK_F_50	Pilih <i>Person In Charge</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat menentukan <i>person in charge</i> dalam suatu <i>workflow</i> .
51	WMK_F_51	Unggah <i>File Workflow</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melakukan <i>upload file</i> dari <i>template workflow admin</i> yang disediakan pada <i>data storage</i> .
52	WMK_F_52	Lihat Daftar <i>My Workflow</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri berdasarkan <i>workflow name</i> , <i>task name</i> , <i>person in charge</i> , <i>deadline</i> , <i>status</i> , <i>start date</i> , dan <i>finish date</i> dalam bentuk tabel.
53	WMK_F_53	Cari <i>My Workflow</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mencari <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri berdasarkan <i>workflow name</i> .
54	WMK_F_54	Lihat <i>Detail My Workflow</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat <i>detail</i> daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri dan sistem

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				menampilkannya dalam bentuk <i>workflow</i> .
55	WMK_F_55	Lihat Statistik <i>My Workflow</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat statistik <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
56	WMK_F_56	Lihat Daftar <i>Task</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel.
57	WMK_F_57	Cari Daftar <i>Task</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mencari daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
58	WMK_F_58	Lihat Isi <i>Form Task</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat isi <i>form task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan yang berisi <i>subject</i> , <i>description</i> , dan daftar <i>file</i> .
59	WMK_F_59	Unduh <i>File Individu Person In Charge</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mengunduh <i>file</i> secara individu pada daftar <i>file</i> pada <i>form task</i> yang diunggah <i>person in charge</i> .
60	WMK_F_60	Unduh <i>Seluruh File Person In Charge</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i>

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
				pada <i>form task</i> yang diunggah <i>person in charge</i> .
61	WMK_F_61	Kirim Notifikasi	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat memberikan notifikasi kepada <i>person in charge</i> pada <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan dengan mengisi isi notifikasi.
62	WMK_F_62	Lihat Daftar File	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat daftar <i>file</i> yang diunggah oleh <i>workflow manager</i> itu sendiri pada <i>workflow</i> yang dipilih.
63	WMK_F_63	Unduh File Individu	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mengunduh <i>file</i> yang di <i>upload</i> oleh <i>workflow manager</i> itu sendiri pada <i>shared files</i> secara individu pada <i>file</i> yang dibagikan pada <i>workflow</i> yang telah diimplementasikan.
64	WMK_F_64	Unduh Seluruh File	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mengunduh <i>file</i> yang diunggah oleh <i>workflow manager</i> itu sendiri pada <i>shared files</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i> pada <i>file</i> yang dibagikan pada <i>workflow</i> yang telah diimplementasikan.
65	WMK_F_65	Lihat Daftar Notifikasi <i>Workflow Manager</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi diterima.

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
66	WMK_F_66	Lihat <i>Profile Workflow Manager</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat melihat <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
67	WMK_F_67	Ubah <i>Password Workflow Manager</i>	<i>Workflow Manager</i>	<i>Workflow Manager</i> dapat mengubah <i>password</i> untuk masuk ke dalam sistem dengan memasukkan <i>password</i> baru dan konfirmasi <i>password</i> baru.
68	WMK_F_68	Lihat <i>Daftar Tugas</i>	<i>Person In Charge</i>	<i>Person In Charge</i> dapat melihat daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name, task name, manager, deadline, task start, dan must complete</i> .
69	WMK_F_69	Cari Tugas	<i>Person In Charge</i>	<i>Person In Charge</i> dapat mencari daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name, task name, manager, deadline, task start, dan must complete</i> .
70	WMK_F_70	Lihat <i>Shared Files</i>	<i>Person In Charge</i>	<i>Person In Charge</i> dapat melihat daftar <i>file</i> yang diunggah oleh <i>workflow manager</i> pada <i>shared files</i> .
71	WMK_F_71	Unduh <i>File Individu Shared Files</i>	<i>Person In Charge</i>	<i>Person In Charge</i> dapat mengunduh <i>file</i> secara individu pada daftar <i>file</i> pada <i>shared files</i> yang diunggah oleh <i>workflow manager</i> .

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
72	WMK_F_72	Lihat Jumlah Tugas Sebelum	<i>Person In Charge</i>	<i>Person In Charge</i> dapat melihat jumlah tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri.
73	WMK_F_73	Lihat Daftar Tugas Sebelum	<i>Person In Charge</i>	<i>Person In Charge</i> dapat melihat daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
74	WMK_F_74	Cari Tugas Sebelum	<i>Person In Charge</i>	<i>Person In Charge</i> dapat mencari daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
75	WMK_F_75	Lihat Isi Form Tugas Sebelum	<i>Person In Charge</i>	<i>Person In Charge</i> dapat melihat isi <i>form</i> tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri yang terdiri dari <i>subject</i> , <i>description</i> dan daftar <i>files</i> .
76	WMK_F_76	Unduh File Individu <i>Person In Charge Files</i>	<i>Person In Charge</i>	<i>Person In Charge</i> dapat mengunduh <i>file</i> secara individu pada daftar <i>files</i> .
77	WMK_F_77	Unduh Seluruh File <i>Person In Charge Files</i>	<i>Person In Charge</i>	<i>Person In Charge</i> dapat mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>files</i> .

Tabel 4.2 Kebutuhan Fungsional (lanjutan)

No	Kode	Use Case	Aktor	Deskripsi Kebutuhan
78	WMK_F_78	Menyelesaikan Tugas	Person In Charge	Person In Charge dapat menyelesaikan tugas yang diberikan workflow manager dengan mengisi subject, description, dan files yang ingin diunggah.
79	WMK_F_79	Unggah File Tugas	Person In Charge	Person In Charge dapat mengunggah files dalam form tugas.
80	WMK_F_80	Lihat Workflow Tugas	Person In Charge	Person In Charge dapat melihat workflow pada tugas yang diberikan workflow manager.
81	WMK_F_81	Lihat Daftar Notifikasi	Person In Charge	Person In Charge dapat melihat daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi masuk.
82	WMK_F_82	Lihat Profile Person	Person In Charge	Person in charge dapat melihat profile berdasarkan full name, department, role, dan email.
83	WMK_F_83	Ubah Password	Person In Charge	Person in charge dapat mengubah password untuk masuk ke dalam sistem dengan memasukan password baru dan konfirmasi password baru.
84	WMK_F_84	Logout	Admin, Workflow Manager, Person In Charge	Sistem dapat menyediakan fungsi untuk logout kepada admin, workflow manager, person in charge untuk keluar dari sistem.

4.4 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan daftar kebutuhan yang berhubungan dengan batasan dan kualitas sistem. Pada kebutuhan non fungsional pada umumnya mencakup pada hal *usability*, *availability*, *performance*, dan *reliability*. Dalam kebutuhan non-fungsional ini diberikan kode WMK-NF-XX. WMK merupakan singkatan dari *workflow* manajemen kerja, NF merupakan singkatan dari non fungsional dan XX merupakan sebuah representasi dari nomor urutan kebutuhan non fungsional. Pada Tabel 4.3 dijabarkan kebutuhan non fungsional yang ada pada sistem pengembangan *workflow* manajemen kerja PT Chevron Pacific Indonesia.

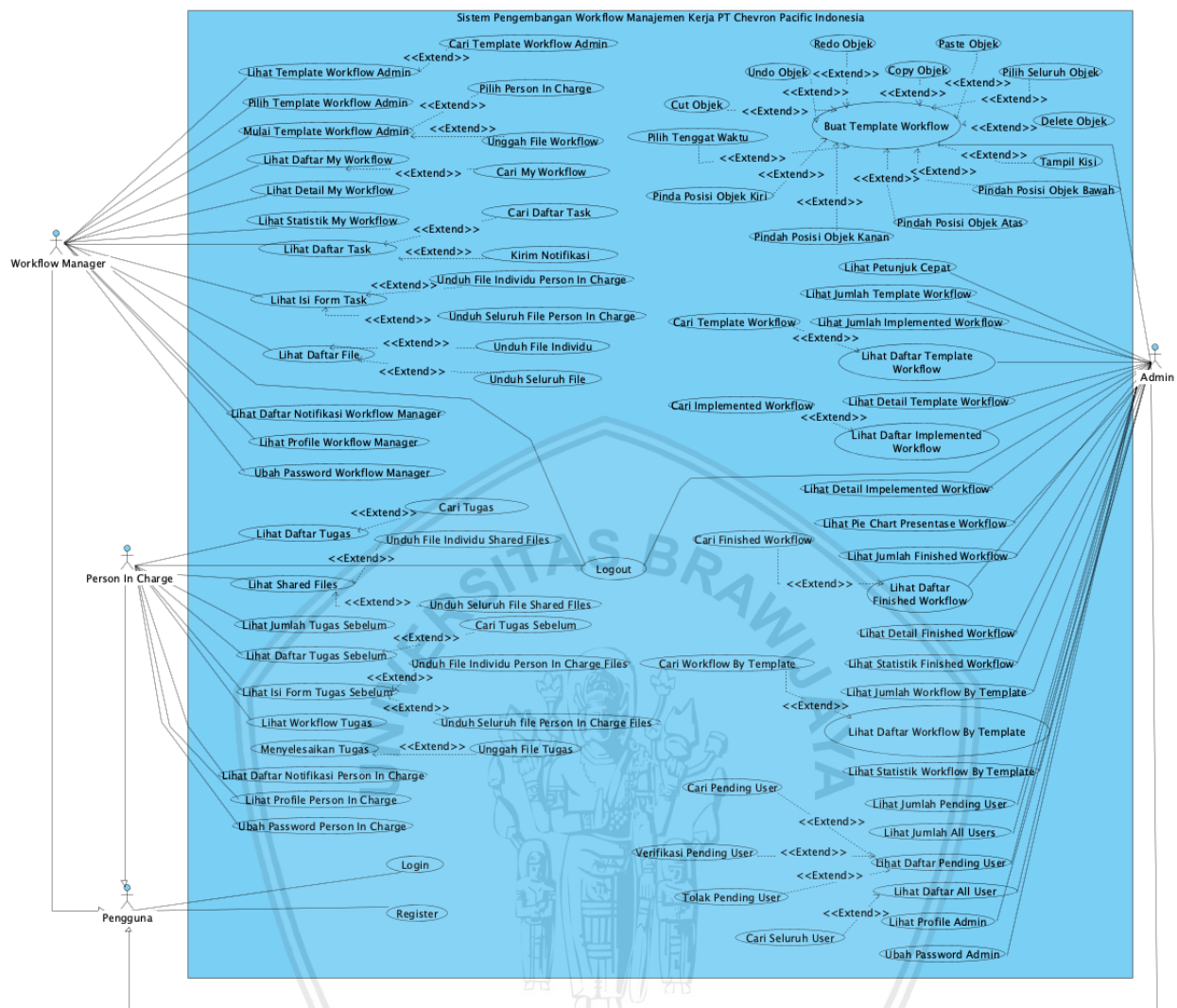
Tabel 4.3 Kebutuhan Non Fungsional

No	Kode	Parameter	Deskripsi Kebutuhan
1	WMK_NF_01	<i>Usability</i>	Sistem dapat digunakan setiap segmen pengguna dengan kemudahan.

4.5 Pemodelan Kebutuhan

4.5.1 Use Case Diagram

Pada diagram *use case* akan memodelkan perilaku aktor terhadap sistem kemudian akan menggambarkan yang bisa dilakukan kepada sistem oleh aktor. Disini *use case diagram* akan mendeskripsikan siapa yang akan menggunakan sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia. Pada sistem *workflow* manajemen kerja PT Chevron Pacific Indoneisa memiliki empat aktor yaitu pengunjung, *admin*, *workflow manager*, dan *person in charge* dan pada sistem ini terdapat *use case* berjumlah 85 yang diantara *use case* tersebut memiliki hubungan *extends*. Tujuan dari *use case diagram* agar dapat terlihat gambaran fungsionalitas dari sistem. Pada gambar 4.1 dapat dilihat *use case diagram* sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia.



Gambar 4.1 Diagram Use Case

4.5.2 Use Case Scenario

Use case scenario adalah penjelasan dari alur kegiatan ketika pengguna menggunakan sistem. Use case scenario kemudian dijabarkan dengan tabel yang terdiri dari actor, objective, pre-condition, main flow, alternative flow, dan post-condition. Pada tabel 4.4 sampai dengan tabel 4.88 akan dijabarkan 84 use case scenario.

Tabel 4.4 Use Case Scenario Login

Actor	Pengunjung
Objective	Aktor dapat mendapatkan izin hak akses sistem dengan melakukan login sehingga aktor dikenali.
Pre-Condition	Halaman login ditampilkan.
Main Flow	1. Aktor mengisi email dan password dengan lengkap.

Tabel 4.4 Use Case Scenario Login (lanjutan)

<i>Main Flow</i>	<ol style="list-style-type: none"> 2. Aktor mengisi <i>email</i> dan <i>password</i> dengan lengkap. 3. Aktor menekan tombol <i>login</i>. 4. Sistem memeriksa <i>email</i> dan <i>password</i> aktor 5. Sistem menampilkan halaman utama aktor.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika <i>email</i> atau <i>password</i> salah maka sistem menampilkan pesan <i>Wrong email or Password</i> dan menampilkan halaman <i>login</i> kembali. 2. Jika akun belum di verifikasi oleh <i>administrator</i> maka sistem menampilkan pesan <i>Waiting for Administrator Confirmation</i>.
<i>Post-Condition</i>	Sistem menampilkan halaman utama aktor.

Tabel 4.5 Use Case Scenario Register

<i>Actor</i>	Pengunjung
<i>Objective</i>	Aktor dapat melakukan registrasi akun sistem dengan melakukan <i>register</i> sehingga akun akan terdaftar pada sistem.
<i>Pre-Condition</i>	Halaman <i>register</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi <i>field full name, email, password, password confirmation</i>, memilih <i>department</i> dan <i>role</i> dengan lengkap. 2. Aktor menekan tombol <i>register</i>. 3. Sistem memeriksa seluruh <i>field, email, password</i> aktor. 4. Sistem menampilkan halaman <i>register</i> kembali.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika terdapat <i>field</i> yang tidak terisi maka sistem akan menampilkan pesan <i>You must fill all data</i>. 2. Jika <i>email</i> akun sudah terdaftar maka sistem akan menampilkan pesan <i>Email already taken</i>. 3. Jika <i>field password</i> dan <i>password confirmation</i> tidak sesuai maka sistem akan menampilkan pesan <i>password confirmation didn't match</i>.
<i>Post-Condition</i>	Sistem menampilkan pesan <i>Waiting for Administator Confirmation</i> .

Tabel 4.6 Use Case Scenario Buat Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mendapatkan izin untuk membuat <i>template workflow</i> BPMN.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menggeser <i>template</i> BPMN ke dalam <i>field</i> pekerjaan. 2. Aktor membuat dan menghubungkan setiap proses dalam BPMN. 3. Aktor memberikan nama proses dan mengisikan tenggat waktu pada setiap proses. 4. Aktor menekan tombol <i>save as</i>. 5. Sistem menampilkan <i>pop up field</i> nama <i>template workflow</i>. 6. Aktor menekan tombol OK. 7. Sistem menampilkan pesan <i>The name that you entered is 'nama template workflow'</i>. 8. Aktor menekan tombol OK.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika nama <i>template workflow</i> sudah ada, maka akan memunculkan pesan <i>Workflow name already used</i>.
<i>Post-Condition</i>	Sistem menyimpan <i>template workflow</i> yang telah dibuat oleh aktor.

Tabel 4.7 Use Case Scenario Lihat Petunjuk Cepat

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat petunjuk cepat cara membuat <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan kalimat <i>quick guide</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan petunjuk cepat membuat <i>workflow</i> .

Tabel 4.8 Use Case Scenario Pilih Tenggat Waktu

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat menentukan tenggat waktu suatu tugas dalam <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan objek <i>note</i> yang sudah dibuat pada <i>field</i> pekerjaan. 2. Sistem menampilkan tampilan untuk memilih jumlah tenggat waktu dalam satuan hari. 3. Aktor memilih banyaknya tenggat waktu. 4. Aktor menekan tombol <i>select</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengubah tenggat waktu pada objek <i>note</i> yang telah ditentukan oleh admin.

Tabel 4.9 Use Case Scenario Undo Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan <i>undo</i> pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>edit</i>. 2. Aktor menekan menu <i>undo</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan melakukan <i>undo</i> pekerjaan pada <i>field</i> pekerjaan.

Tabel 4.10 Use Case Scenario Redo Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan <i>redo</i> pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>edit</i>. 2. Aktor menekan menu <i>redo</i>.

Tabel 4.10 Use Case Scenario Redo Objek (lanjutan)

<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan melakukan <i>redo</i> pekerjaan pada <i>field</i> pekerjaan.

Tabel 4.11 Use Case Scenario Cut Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan <i>cut</i> pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>edit</i>. 3. Aktor menekan menu <i>cut</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan melakukan <i>cut</i> pada objek yang telah dipilih.

Tabel 4.12 Use Case Scenario Copy Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan <i>copy</i> pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>edit</i>. 3. Aktor menekan menu <i>copy</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan melakukan <i>copy</i> pada objek yang telah dipilih.

Tabel 4.13 Use Case Scenario Paste Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan <i>paste</i> pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>edit</i>.



Tabel 4.13 Use Case Scenario Paste Objek (lanjutan)

<i>Main Flow</i>	2. Aktor menekan menu <i>paste</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan melakukan <i>paste</i> dari hasil <i>copy</i> atau <i>cut</i> objek sebelumnya.

Tabel 4.14 Use Case Scenario Delete Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan <i>delete</i> pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>edit</i>. 3. Aktor menekan menu <i>delete</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan melakukan <i>delete</i> pada objek yang dipilih sebelumnya.

Tabel 4.15 Use Case Scenario Pilih Seluruh Objek

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat memilih seluruh objek pada saat pembuatan <i>workflow</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>edit</i>. 2. Aktor menekan menu <i>select all</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan memilih seluruh objek yang ada pada <i>field</i> pekerjaan.

Tabel 4.16 Use Case Scenario Pindah Posisi Objek Kiri

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mengubah posisi objek ke sisi kiri <i>field</i> pekerjaan.

Tabel 4.16 Use Case Scenario Pindah Posisi Objek Kiri (lanjutan)

<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>left sides</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengubah posisi objek yang dipilih ke sebelah kiri <i>field</i> pekerjaan.

Tabel 4.17 Use Case Scenario Pindah Posisi Objek Kanan

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mengubah posisi objek ke sisi kanan <i>field</i> pekerjaan.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>right sides</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengubah posisi objek yang dipilih ke sebelah kanan <i>field</i> pekerjaan.

Tabel 4.18 Use Case Scenario Pindah Posisi Objek Atas

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mengubah posisi objek ke sisi atas <i>field</i> pekerjaan.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>tops</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengubah posisi objek yang dipilih ke sebelah atas <i>field</i> pekerjaan.

Tabel 4.19 Use Case Scenario Pindah Posisi Objek Bawah

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mengubah posisi objek ke sisi bawah <i>field</i> pekerjaan.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>bottoms</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengubah posisi objek yang dipilih ke sebelah bawah <i>field</i> pekerjaan.

Tabel 4.20 Use Case Scenario Tampil Kisi

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat menampilkan kisi pada <i>field</i> pekerjaan.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>options</i>. 2. Aktor menekan menu <i>grid</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan kisi pada <i>field</i> pekerjaan.

Tabel 4.21 Use Case Scenario Lihat Jumlah Template Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat jumlah <i>template workflow</i> yang ada pada sistem.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>workflow</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah <i>template workflow</i> yang ada pada sistem.

Tabel 4.22 Use Case Scenario Lihat Jumlah Implemented Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat jumlah <i>workflow</i> yang sudah dikerjakan oleh <i>workflow manager</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>workflow</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah <i>workflow</i> yang sudah diselesaikan <i>workflow manager</i> pada bagian <i>implemented workflow</i> .

Tabel 4.23 Use Case Scenario Lihat Daftar Template Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>workflow</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>view</i> pada menu <i>template workflow</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan halaman <i>template workflow</i> yang berisi daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.24 Use Case Scenario Cari Template Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat hasil pencarian <i>template workflow</i> .
<i>Pre-Condition</i>	Halaman <i>template workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.



Tabel 4.24 Use Case Scenario Cari Template Workflow (lanjutan)

<i>Alternative Flow</i>	1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i> .
<i>Post-Condition</i>	Sistem menampilkan daftar <i>template workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.25 Use Case Scenario Lihat Detail Template Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mendapatkan izin untuk melihat secara <i>detail template workflow</i> dalam bentuk <i>workflow</i> .
<i>Pre-Condition</i>	Halaman <i>template workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar template workflow</i>. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>workflow</i> dari <i>template workflow</i> yang telah dipilih aktor.

Tabel 4.26 Use Case Scenario Lihat Implemented Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat daftar <i>implemented workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>workflow</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>view</i> pada menu <i>implemented workflow</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan halaman <i>implemented workflow</i> yang berisi daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.



Tabel 4.27 Use Case Scenario Cari Implemented Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat hasil pencarian <i>implemented workflow</i> .
<i>Pre-Condition</i>	Halaman <i>implemented workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i>.
<i>Post-Condition</i>	Sistem menampilkan daftar <i>implemented workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.28 Use Case Scenario Lihat Detail Daftar Implemented Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mendapatkan izin untuk melihat secara <i>detail implemented workflow</i> dalam bentuk <i>workflow</i> .
<i>Pre-Condition</i>	Halaman <i>template workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar implemented workflow</i>. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>workflow</i> dari <i>implemented workflow</i> yang telah dipilih aktor.

Tabel 4.29 Use Case Scenario Lihat Pie Chart Persentase Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat <i>pie chart</i> mengenai persentase penggunaan <i>workflow</i> baik <i>workflow</i> yang telah selesai maupun <i>workflow</i> yang sedang berjalan.
<i>Pre-Condition</i>	Halaman <i>template workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>reports</i>.

Tabel 4.29 Use Case Scenario Lihat Pie Chart Persentase Workflow (lanjutan)

<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>pie chart</i> mengenai persentase penggunaan <i>workflow</i> baik <i>workflow</i> yang telah selesai maupun <i>workflow</i> yang sedang berjalan.

Tabel 4.30 Use Case Scenario Lihat Jumlah Finished Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat jumlah <i>workflow</i> yang sudah diselesaikan oleh <i>workflow manager</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>reports</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah <i>workflow</i> yang sudah diselesaikan <i>workflow manager</i> pada bagian <i>finished workflow</i> .

Tabel 4.31 Use Case Scenario Lihat Daftar Finished Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat daftar <i>finished workflow</i> berdasarkan <i>workflow name</i> , <i>task name</i> , <i>workflow manager</i> , <i>PIC</i> , <i>deadline</i> , <i>start task</i> dan <i>finish task</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>workflow</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>view</i> pada menu <i>finished workflow</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>finished workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , <i>workflow manager</i> , <i>PIC</i> , <i>deadline</i> , <i>start task</i> dan <i>finish task</i> dalam bentuk tabel.

Tabel 4.32 Use Case Scenario Cari Finished Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mencari <i>finished workflow</i> berdasarkan <i>workflow name</i> .

Tabel 4.32 Use Case Scenario Cari Finished Workflow (lanjutan)

<i>Pre-Condition</i>	Halaman <i>workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i>.
<i>Post-Condition</i>	Sistem menampilkan daftar <i>finished workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name, task name, workflow manager, PIC, deadline, start task</i> dan <i>finish task</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.33 Use Case Scenario Lihat Detail Finished Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mendapatkan izin untuk melihat secara <i>detail finished workflow</i> dalam bentuk <i>workflow</i> .
<i>Pre-Condition</i>	Halaman <i>finished workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar finished workflow</i>. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>workflow</i> dari <i>finished workflow</i> yang telah dipilih aktor.

Tabel 4.34 Use Case Scenario Lihat Statistik Finished Workflow

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat statistik <i>workflow</i> yang telah dipilih aktor dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
<i>Pre-Condition</i>	Halaman <i>finished workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar finished workflow</i>. 2. Aktor menekan tombol <i>statistics</i>.
<i>Alternative Flow</i>	-

Tabel 4.34 Use Case Scenario Lihat Statistik Finished Workflow (lanjutan)

<i>Post-Condition</i>	Sistem menampilkan statistik <i>workflow</i> yang telah dipilih aktor sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
-----------------------	---

Tabel 4.35 Use Case Scenario Lihat Jumlah Workflow By Template

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat jumlah <i>workflow</i> berdasarkan jenis <i>template workflow</i> yang ada.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>reports</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah <i>workflow</i> berdasarkan jenis <i>template workflow</i> yang ada.

Tabel 4.36 Use Case Scenario Lihat Daftar Workflow By Template

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> .
<i>Pre-Condition</i>	Halaman <i>reports</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>view</i> pada menu <i>workflow by template</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> .

Tabel 4.37 Use Case Scenario Cari Workflow By Template

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mencari <i>workflow by template</i> berdasarkan <i>template workflow name</i> .
<i>Pre-Condition</i>	Halaman <i>workflow by template</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menulis kata yang ingin dicari berdasarkan <i>template workflow name</i> pada kolom pencarian.

Tabel 4.37 Use Case Scenario Cari Workflow By Template (lanjutan)

<i>Main Flow</i>	2. Aktor menekan <i>enter</i> pada <i>keyboard</i> .
<i>Alternative Flow</i>	1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i> .
<i>Post-Condition</i>	Sistem menampilkan daftar <i>template by workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>template workflow name</i> dan <i>status</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.38 Use Case Scenario Lihat Statistik Workflow By Template

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat statistik <i>workflow by template</i> yang telah dipilih aktor dalam bentuk <i>bar chart</i> rata-rata <i>template workflow</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
<i>Pre-Condition</i>	Halaman <i>finished workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan salah satu daftar <i>finished workflow</i>. 2. Aktor menekan tombol <i>statistics</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan melihat statistik <i>workflow by template</i> yang telah dipilih aktor dalam bentuk <i>bar chart</i> rata-rata <i>template workflow</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .

Tabel 4.39 Use Case Scenario Lihat Jumlah Pending User

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat jumlah <i>pending user</i> yang ada pada sistem.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>users</i>
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah <i>pending user</i> yang ada pada sistem.

Tabel 4.40 Use Case Scenario Lihat Jumlah All User

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat jumlah seluruh <i>user</i> yang ada pada sistem pada bagian <i>all user</i> .
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>users</i>
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah seluruh <i>user</i> yang ada pada sistem pada bagian <i>all user</i> .

Tabel 4.41 Use Case Scenario Lihat Daftar Pending Users

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat daftar <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>users</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menekan tombol <i>manage</i> pada menu <i>pending users</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel.

Tabel 4.42 Use Case Scenario Cari Pending Users

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mencari <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>users</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>full name, email, role, atau department workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.

Tabel 4.42 Use Case Scenario Cari Pending Users (lanjutan)

<i>Alternative Flow</i>	Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i> .
<i>Post-Condition</i>	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.43 Use Case Scenario Verifikasi Pending Users

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melakukan verifikasi terhadap akun yang mendaftar pada sistem dalam <i>pending users</i> sehingga dapat masuk ke dalam sistem.
<i>Pre-Condition</i>	Halaman <i>manage users</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan salah satu baris <i>user</i>. 2. Aktor menekan tombol <i>verify</i>. 3. Sistem menampilkan pesan <i>Are you sure that you want to verify this user?</i> 4. Aktor menekan tombol OK.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol <i>cancel</i> maka sistem akan menampilkan pesan <i>You choose to cancel</i>.
<i>Post-Condition</i>	Sistem menampilkan pesan <i>user verified</i> dan mengizinkan <i>user</i> untuk masuk ke dalam sistem.

Tabel 4.44 Use Case Scenario Tolak Pending Users

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat menolak terhadap akun yang mendaftar pada sistem dalam <i>pending users</i> sehingga dapat masuk ke dalam sistem.
<i>Pre-Condition</i>	Halaman <i>manage users</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan salah satu baris <i>user</i>. 2. Aktor menekan tombol <i>decline</i>.



Tabel 4.44 Use Case Scenario Tolak Pending Users (lanjutan)

<i>Main Flow</i>	<ol style="list-style-type: none"> 3. Sistem menampilkan pesan <i>Are you sure that you want to decline this user?</i> 4. Aktor menekan tombol OK.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol <i>cancel</i> maka sistem akan menampilkan pesan <i>You choose to cancel.</i>
<i>Post-Condition</i>	Sistem menampilkan pesan <i>user rejected.</i>

Tabel 4.45 Use Case Scenario Lihat Daftar All Users

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat daftar <i>all users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>users</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>view</i> pada menu <i>all users.</i>
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>all users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel.

Tabel 4.46 Use Case Scenario Cari All Users

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mencari <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>users</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan jenis <i>template workflow, full name, email, role, department</i> atau <i>status workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard.</i>
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display.</i>



Tabel 4.46 Use Case Scenario Cari All Users (lanjutan)

<i>Post-Condition</i>	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
-----------------------	--

Tabel 4.47 Use Case Scenario Lihat Profile Admin

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat melihat <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown admin</i>. 2. Aktor menekan menu <i>profile</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .

Tabel 4.48 Use Case Scenario Ubah Password Admin

<i>Actor</i>	Admin
<i>Objective</i>	Aktor dapat mengubah <i>password</i> sebelumnya menjadi <i>password</i> yang baru.
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown admin</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i>. 5. Aktor menekan tombol <i>change password</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika isi kolom <i>password</i> baru dan konfirmasi <i>password</i> baru kosong atau tidak sama maka sistem akan menampilkan pesan <i>password empty or didn't match</i>.

Tabel 4.48 Use Case Scenario Ubah Password Admin (lanjutan)

<i>Post-Condition</i>	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.
-----------------------	--

Tabel 4.49 Use Case Scenario Lihat Template Workflow Admin

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat daftar <i>template workflow</i> yang sudah dibuat oleh admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>workflow</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan tabel <i>template workflow</i> berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam bentuk tabel.

Tabel 4.50 Use Case Scenario Cari Template Workflow Admin

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mencari daftar <i>template workflow</i> yang sudah dibuat oleh admin sebelumnya berdasarkan <i>workflow name</i> .
<i>Pre-Condition</i>	Halaman <i>workflow</i> ditampilkan.
<i>Main Flow</i>	1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i> .
<i>Alternative Flow</i>	1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i> .
<i>Post-Condition</i>	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.51 Use Case Scenario Pilih Template Workflow Admin

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat memilih <i>template workflow</i> kemudian sistem akan menampilkan <i>workflow</i> yang sudah dipilih.
<i>Pre-Condition</i>	Halaman <i>workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>template workflow</i> yang akan diinginkan. 2. Aktor menekan tombol <i>use</i> pada baris tersebut.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam bentuk tabel.

Tabel 4.52 Use Case Scenario Mulai Template Workflow Admin

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat memulai <i>template workflow</i> yang sudah dipilih oleh <i>workflow manager</i> itu sendiri.
<i>Pre-Condition</i>	Halaman <i>workflow use</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi <i>subject</i> pada <i>field</i> objek yang disediakan. 2. Aktor menekan tombol <i>start workflow</i>. 3. Sistem menampilkan pesan <i>are you sure that you want to start the workflow?</i> 4. Aktor menekan tombol OK.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika aktor menekan tombol <i>cancel</i> maka sistem akan menampilkan pesan <i>You choose to Cancel the Workflow</i>.
<i>Post-Condition</i>	Sistem menampilkan pesan <i>Workflow Started</i> yang sekaligus memulai <i>workflow</i> tersebut dan mengirimkan notifikasi ke <i>person in charge</i> yang terlibat.

Tabel 4.53 Use Case Scenario Pilih Person In Charge

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat menentukan <i>person in charge</i> dalam suatu <i>workflow</i> .



Tabel 4.53 Use Case Scenario Pilih Person In Charge (lanjutan)

<i>Pre-Condition</i>	Halaman <i>workflow use</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan bagian kata PIC pada objek yang disediakan. 2. Sistem akan menampilkan daftar <i>person in charge</i> berdasarkan <i>full name</i>, dan <i>email</i> dalam bentuk tabel. 3. Aktor memilih dengan menekan baris <i>person in charge</i> yang akan dipilih. 4. Aktor menekan tombol <i>choose</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem mengubah kata PIC sebelumnya menjadi <i>person in charge</i> yang telah ditentukan aktor sebelumnya.

Tabel 4.54 Use Case Scenario Unggah File Workflow

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melakukan mengunggah <i>file</i> dari <i>template workflow admin</i> yang disediakan pada <i>data storage</i> .
<i>Pre-Condition</i>	Halaman <i>workflow use</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan klik kanan pada objek <i>data storage</i>. 2. Aktor menekan tombol <i>upload a file</i>. 3. Sistem menampilkan tampilan untuk melakukan unggah <i>file</i>. 4. Aktor menekan tombol <i>select files</i>. 5. Sistem akan menampilkan direktori <i>folder</i> aktor. 6. Aktor memilih <i>file</i> yang akan diunggah. 7. Sistem menampilkan nama <i>file</i>, tipe <i>file</i>, dan ukuran <i>file</i>. 8. Aktor menekan tombol <i>submit</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menyimpan <i>file</i> yang sudah dipilih oleh aktor sebelumnya yang akan terunggah ketika <i>workflow</i> dijalankan.

Tabel 4.55 Use Case Scenario Lihat Daftar My Workflow

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>my workflow</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel.

Tabel 4.56 Use Case Scenario Cari My Workflow

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mencari daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name</i> .
<i>Pre-Condition</i>	Halaman <i>template workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
<i>Alternative Flow</i>	1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i> .
<i>Post-Condition</i>	Sistem menampilkan daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.57 Use Case Scenario Lihat Detail My Workflow

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat <i>detail</i> daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri dan sistem menampilkannya dalam bentuk <i>workflow</i> .



Tabel 4.57 Use Case Scenario Lihat Detail My Workflow (lanjutan)

<i>Pre-Condition</i>	Halaman <i>my workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>detail</i> daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri dan sistem menampilkannya dalam bentuk <i>workflow</i> .

Tabel 4.58 Use Case Scenario Lihat Statistik My Workflow

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat statistik <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
<i>Pre-Condition</i>	Halaman <i>my workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>statistics</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan statistik <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .

Tabel 4.59 Use Case Scenario Lihat Daftar Task

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel.
<i>Pre-Condition</i>	Halaman <i>my workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>statistics</i>.

Tabel 4.59 Use Case Scenario Lihat Daftar Task (lanjutan)

<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel.

Tabel 4.60 Use Case Scenario Cari Daftar Task

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mencari daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
<i>Pre-Condition</i>	Halaman <i>statistics</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i>.
<i>Post-Condition</i>	Sistem menampilkan daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.61 Use Case Scenario Lihat Isi Form Task

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat isi <i>form task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan yang berisi <i>subject</i> , <i>description</i> , dan daftar <i>file</i> .
<i>Pre-Condition</i>	Halaman <i>statistics</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika tidak ada <i>file</i> yang tersedia maka akan terdapat kalimat <i>No Files Available</i>.



Tabel 4.61 Use Case Scenario Lihat Isi Form Task (lanjutan)

<i>Post-Condition</i>	Sistem menampilkan isi <i>form task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan yang berisi <i>subject</i> , <i>description</i> , dan daftar <i>file</i> .
-----------------------	---

Tabel 4.62 Use Case Scenario Unduh File Individu Person In Charge

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mengunduh <i>file</i> secara individu pada daftar <i>file</i> pada <i>form task</i> yang diunggah <i>person in charge</i> .
<i>Pre-Condition</i>	Halaman <i>form</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor memilih dengan menekan <i>file</i> yang akan diunduh.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.

Tabel 4.63 Use Case Scenario Unduh Seluruh File Person In Charge

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i> pada <i>form task</i> yang diunggah <i>person in charge</i> .
<i>Pre-Condition</i>	Halaman <i>form</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor menekan tombol <i>download</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i> pada <i>form task</i> yang diunggah <i>person in charge</i> .

Tabel 4.64 Use Case Scenario Kirim Notifikasi

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat memberi notifikasi kepada <i>person in charge</i>

Tabel 4.64 Use Case Scenario Kirim Notifikasi (lanjutan)

	pada <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan.
<i>Pre-Condition</i>	Halaman <i>statistics</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>notification</i>. 3. Sistem menampilkan tampilan untuk menuliskan isi notifikasi pada sebuah <i>textfield</i>. 4. Aktor menuliskan pesan yang akan dikirimkan. 5. Aktor menekan tombol <i>send</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem mengirimkan notifikasi ke <i>person in charge</i> yang telah dipilih kemudian mengembalikan tampilan ke halaman <i>statistics</i> .

Tabel 4.65 Use Case Scenario Lihat Daftar File

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat daftar <i>file</i> yang diunggah oleh aktor itu sendiri pada <i>workflow</i> yang dipilih.
<i>Pre-Condition</i>	Halaman <i>my workflow</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>statistics</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>file</i> yang diunggah oleh aktor itu sendiri pada <i>workflow</i> yang dipilih.

Tabel 4.66 Use Case Scenario Unduh File Individu

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat dapat mengunduh <i>file</i> yang diunggah oleh aktor sendiri secara individu pada <i>shared files</i> yang dibagikan pada <i>workflow</i> yang telah diimplementasikan.



Tabel 4.66 Use Case Scenario Unduh File Individu (lanjutan)

<i>Pre-Condition</i>	Halaman <i>statistics</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor memilih dengan menekan <i>file</i> yang akan diunduh.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.

Tabel 4.67 Use Case Scenario Unduh Seluruh File

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mengunduh <i>file</i> yang diunduh oleh aktor sendiri secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i> pada <i>shared files</i> yang dibagikan di <i>workflow</i> yang telah diimplementasikan.
<i>Pre-Condition</i>	Halaman <i>form</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor menekan tombol <i>download all</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .

Tabel 4.68 Use Case Scenario Lihat Daftar Notifikasi Workflow Manager

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi diterima.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>notifications</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi diterima.



Tabel 4.69 Use Case Scenario Lihat Profile Workflow Manager

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat melihat <i>profile</i> berdasarkan <i>full name</i> , <i>department</i> , <i>role</i> , dan <i>email</i> .
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown workflow manager</i>. 2. Aktor menekan menu <i>profile</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>profile</i> berdasarkan <i>full name</i> , <i>department</i> , <i>role</i> , dan <i>email</i> .

Tabel 4.70 Use Case Scenario Ubah Password Workflow Manager

<i>Actor</i>	<i>Workflow Manager</i>
<i>Objective</i>	Aktor dapat mengubah <i>password</i> sebelumnya menjadi <i>password</i> yang baru.
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown workflow manager</i>. 2. Aktor menekan <i>dropdown workflow manager</i>. 3. Aktor menekan menu <i>profile</i>. 4. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 5. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i>. 6. Aktor menekan tombol <i>change password</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika isi kolom <i>password</i> baru dan konfirmasi <i>password</i> baru kosong atau tidak sama maka sistem akan menampilkan pesan <i>password empty or didn't match</i>.
<i>Post-Condition</i>	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.

Tabel 4.71 Use Case Scenario Lihat Daftar Tugas

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat daftar tugas yang diberikan <i>workflow</i>



Tabel 4.71 Use Case Scenario Lihat Daftar Tugas (lanjutan)

	<i>manager</i> berdasarkan <i>workflow name</i> , <i>task name</i> , <i>manager</i> , <i>deadline</i> , <i>task start</i> , dan <i>must complete</i> .
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	1. Aktor membuka menu <i>task</i> .
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name</i> , <i>task name</i> , <i>manager</i> , <i>deadline</i> , <i>task start</i> , dan <i>must complete</i> .

Tabel 4.72 Use Case Scenario Cari Tugas

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat mencari daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name</i> , <i>task name</i> , <i>manager</i> , <i>deadline</i> , <i>task start</i> , dan <i>must complete</i> .
<i>Pre-Condition</i>	Halaman <i>task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
<i>Alternative Flow</i>	1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>No items to display</i> .
<i>Post-Condition</i>	Sistem menampilkan hasil pencarian daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name</i> , <i>task name</i> , <i>manager</i> , <i>deadline</i> , <i>task start</i> , dan <i>must complete</i> kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 4.73 Use Case Scenario Lihat Shared Files

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat daftar <i>file</i> yang diunggah oleh <i>workflow manager</i> pada <i>shared files</i> .
<i>Pre-Condition</i>	Halaman <i>task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.

Tabel 4.73 Use Case Scenario Lihat Shared Files (lanjutan)

<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar <i>file</i> yang diunggah oleh <i>workflow manager</i> pada <i>shared files</i> .

Tabel 4.74 Use Case Scenario Unduh File Individu Shared Files

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat mengunduh <i>file</i> secara individu pada daftar <i>file</i> pada <i>shared files</i> yang diunggah oleh <i>workflow manager</i> .
<i>Pre-Condition</i>	Halaman <i>task detail</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>shared files</i>. 2. Aktor memilih dengan menekan <i>file</i> yang akan diunduh.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.

Tabel 4.75 Use Case Scenario Unduh Seluruh File Shared Files

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat jumlah tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri.
<i>Pre-Condition</i>	Halaman <i>task detail</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>shared files</i>. 2. Aktor menekan tombol <i>download all</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .

Tabel 4.76 Use Case Scenario Lihat Jumlah Tugas Sebelum

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat jumlah tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri.

Tabel 4.76 Use Case Scenario Lihat Jumlah Tugas Sebelum (lanjutan)

<i>Pre-Condition</i>	Halaman <i>task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan jumlah tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri pada bagian <i>previous task</i> .

Tabel 4.77 Use Case Scenario Lihat Daftar Tugas Sebelum

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
<i>Pre-Condition</i>	Halaman <i>task detail</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>view previous task</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .

Tabel 4.78 Use Case Scenario Cari Tugas Sebelum

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat mencari daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
<i>Pre-Condition</i>	Halaman <i>previous task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika kata pencarian dicari tidak ada maka status pada tabel akan menampilkan <i>no items to display</i>.

Tabel 4.78 Use Case Scenario Cari Tugas Sebelum (lanjutan)

<i>Post-Condition</i>	Sistem menampilkan hasil pencarian daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
-----------------------	---

Tabel 4.79 Use Case Scenario Lihat Isi Form Tugas Sebelum

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat isi <i>form</i> tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri yang terdiri dari <i>subject</i> , <i>description</i> dan daftar <i>files</i> .
<i>Pre-Condition</i>	Halaman <i>previous task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas sebelum yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan isi <i>form</i> tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri yang terdiri dari <i>subject</i> , <i>description</i> dan daftar <i>files</i> .

Tabel 4.80 Use Case Scenario Unduh File Individu Person In Charge Files

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat dapat mengunduh <i>file</i> secara individu pada daftar <i>files</i> .
<i>Pre-Condition</i>	Halaman <i>previous task detail</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>form</i> 2. Aktor memilih dengan menekan <i>file</i> yang akan diunduh.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.



Tabel 4.81 Use Case Scenario Unduh Seluruh File Person In Charge Files

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>files</i> .
<i>Pre-Condition</i>	Halaman <i>previous task detail</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>form</i> 2. Aktor menekan tombol <i>download</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .

Tabel 4.82 Use Case Scenario Menyelesaikan Tugas

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat menyelesaikan tugas yang diberikan <i>workflow manager</i> dengan mengisi <i>subject</i> , <i>description</i> , dan <i>files</i> yang ingin diunggah.
<i>Pre-Condition</i>	Halaman <i>task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor mengisi isi <i>subject</i> pada <i>field</i> yang disediakan. 2. Aktor mengisi isi <i>description</i> pada <i>field</i> yang disediakan. 3. Aktor mengunggah <i>file</i> yang ingin diunggah. 4. Aktor menekan tombol <i>complete</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Aktor dapat menyelesaikan tugas yang diberikan <i>workflow manager</i> dengan mengisi <i>subject</i> , <i>description</i> , dan <i>files</i> yang ingin di unggah.

Tabel 4.83 Use Case Scenario Unggah File Tugas

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat mengunggah <i>files</i> dalam <i>form</i> tugas.
<i>Pre-Condition</i>	Halaman <i>task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>select files</i>.

Tabel 8.83 Use Case Scenario Unggah File Tugas (lanjutan)

	<ol style="list-style-type: none"> 2. Sistem akan menampilkan direktori <i>folder</i> aktor. 3. Aktor memilih <i>file</i> yang akan diunggah. 4. Sistem menampilkan nama <i>file</i>, tipe <i>file</i>, dan ukuran <i>file</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menyimpan <i>file</i> yang sudah dipilih oleh aktor sebelumnya yang akan terunggah ketika tugas diselesaikan.

Tabel 4.84 Use Case Scenario Lihat Workflow Tugas

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat <i>workflow</i> yang pada tugas yang diberikan <i>workflow manager</i> pada bagian <i>workflow preview</i> .
<i>Pre-Condition</i>	Halaman <i>task</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan <i>workflow</i> pada tugas yang diberikan <i>workflow manager</i> pada bagian <i>workflow preview</i> .

Tabel 4.85 Use Case Scenario Lihat Daftar Notifikasi Person In Charge

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi masuk.
<i>Pre-Condition</i>	Halaman utama ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>notifications</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan menampilkan daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi masuk.



Tabel 4.86 Use Case Scenario Lihat Profile Person In Charge

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat melihat <i>profile</i> berdasarkan <i>full name</i> , <i>department</i> , <i>role</i> , dan <i>email</i> .
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown person in charge</i>. 2. Aktor menekan menu <i>profile</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem menampilkan <i>profile</i> berdasarkan <i>full name</i> , <i>department</i> , <i>role</i> , dan <i>email</i> .

Tabel 4.87 Use Case Scenario Ubah Password Person In Charge

<i>Actor</i>	<i>Person In Charge</i>
<i>Objective</i>	Aktor dapat mengubah <i>password</i> sebelumnya menjadi <i>password</i> yang baru.
<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown workflow manager</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i>. 5. Aktor menekan tombol <i>change password</i>.
<i>Alternative Flow</i>	<ol style="list-style-type: none"> 1. Jika isi kolom <i>password</i> baru dan konfirmasi <i>password</i> baru kosong atau tidak sama maka sistem akan menampilkan pesan <i>password empty or didn't match</i>.
<i>Post-Condition</i>	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.

Tabel 4.88 Use Case Scenario Logout

<i>Actor</i>	<i>Admin, Workflow Manager, Person In Charge</i>
<i>Objective</i>	Sistem dapat menyediakan fungsi untuk <i>logout</i> kepada <i>aktor</i> untuk keluar dari sistem.

Tabel 4.88 Use Case Scenario Logout (lanjutan)

<i>Pre-Condition</i>	Halaman <i>utama</i> ditampilkan.
<i>Main Flow</i>	<ol style="list-style-type: none">1. Aktor menekan <i>dropdown</i> pada nama <i>role</i> aktor.2. Aktor menekan tombol <i>logout</i>.
<i>Alternative Flow</i>	-
<i>Post-Condition</i>	Sistem akan mengeluarkan aktor dari sistem menuju ke halaman <i>login</i> .



BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

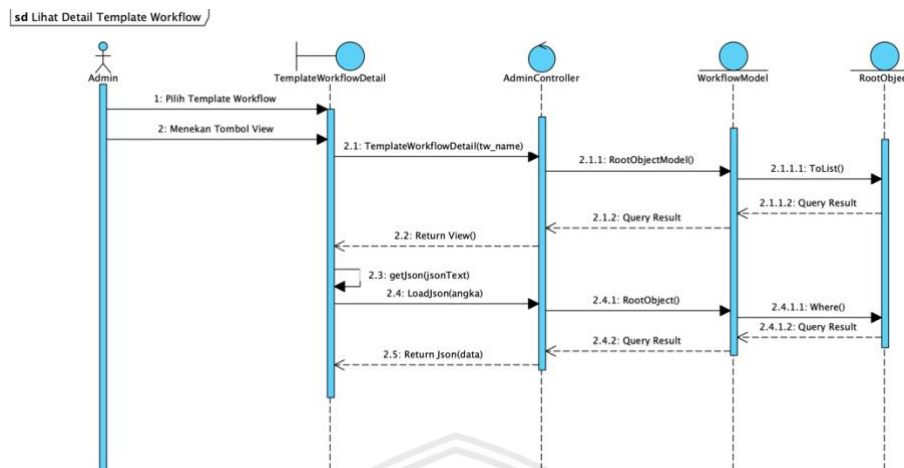
Pada perancangan sistem akan dibahas mengenai sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia. Sebelumnya telah ditentukan kebutuhan yang ada pada sistem, selanjutnya akan dilanjutkan pada tahapan perancangan yaitu pengembangan perangkat lunak yang berdasarkan hasil analisis kebutuhan. Pada perancangan sistem dilakukan melalui tahapan seperti perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, perancangan komponen, dan perancangan antarmuka.

5.1.1 Perancangan *Sequence Diagram*

Pada *sequence diagram* berfungsi untuk merepresentasikan alur antara objek pada sistem yang saling melakukan interaksi. *Sequence diagram* di dalamnya terdapat objek yang didapati dari identifikasi *use case* dan spesifikasi kebutuhan sebelumnya. Selanjutnya terdapat tiga *sequence diagram* yang akan digunakan sebagai sampel.

5.1.1.1 *Sequence Diagram* Lihat *Detail Template Workflow*

Sequence Diagram lihat *detail* daftar *template workflow* yang terdapat pada Gambar 5.1 menggambarkan proses pada saat admin memilih *template workflow* lalu sistem akan menampilkan *workflow* dari *template workflow* yang dipilih tersebut. Pada *sequence diagram* ini terdapat empat objek dengan aktor *admin* yaitu dengan *boundary TemplateWorkflowDetail*, terdapat satu *controller AdminController*, dan dua model yaitu *WorkflowModel* dan *RootObject*. Dalam *sequence* ini diawali dengan aktor memilih *workflow* kemudian setelah dipilih aktor menekan tombol *view* yang memanggil *method* dari *AdminController* yaitu *TemplateWorkflowDetail(tw_name)* yang berfungsi untuk mengambil nama dari *template workflow* yang dipilih aktor sebelumnya lalu dibandingkan dengan data yang diperoleh dari hasil *query model workflow* lalu memanggil *model RootObject* dengan membandingkan nama tersebut, jika sama maka akan disimpan id *workflow* tersebut. Selanjutnya dari *AdminController* akan melakukan *return* ke *boundary TemplateWorkflowDetail* dengan data id yang telah didapatkan lalu memanggil *method* *getJSON(JsonText)* pada *javascript* dengan *JsonText* adalah id sebelumnya, kemudian memanggil *LoadJSON(angka)* dengan angka adalah id lalu dibandingkan kembali ke *RootObject* jika terdapat id yang sesuai pada *database* lalu akan dikembalikan hasil tersebut dalam bentuk *Json* yang akan menampilkan tampilan *workflow* yang dipilih admin sebelumnya.

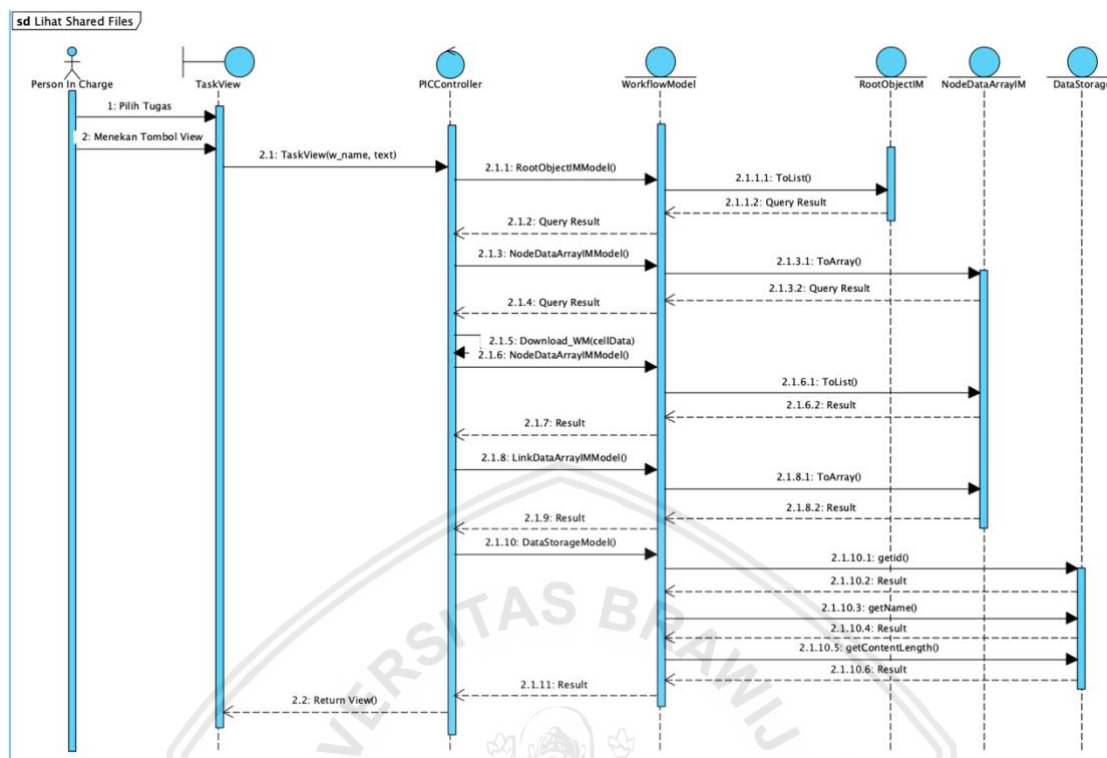


Gambar 5.1 Sequence Diagram Lihat Detail Template Workflow

5.1.1.2 Sequence Diagram Lihat Shared Files

Sequence Diagram lihat *shared files* yang terdapat pada Gambar 5.2 menggambarkan proses pada saat *person in charge* melihat isi dari tugas yang kemudian terdapat daftar *file* yang dibagikan oleh *workflow manager*. Pada *sequence diagram* ini terdapat enam objek dengan aktor *person in charge* yaitu dengan *boundary TaskView*, terdapat satu *controller PICController*, dan tiga model yaitu *WorkflowModel*, *RootObjectIM*, *NodeDataArrayIM*, dan *DataStorage*. Dalam *sequence* ini diawali dengan aktor memilih *tugas* kemudian setelah dipilih aktor menekan tombol *view* yang memanggil *method* dari *PICController* yaitu *TaskView(w_name, text)* yang berfungsi untuk mengambil nama dari *workflow* dan *text* objek dari *workflow* yang dipilih aktor sebelumnya lalu dibandingkan dengan data yang diperoleh dari hasil *query model workflow* lalu memanggil *model RootObjectIM* dan *NodeDataArrayIM* dengan membandingkan nama tersebut, jika sama maka akan disimpan *id workflow* tersebut. Selanjutnya dari *PICController* akan memanggil *method Download_WM(cellData)* dengan data *cellData* yang telah didapatkan dari perbandingan nama sebelumnya. Kemudian pada *method* tersebut mengambil data dari *files* melalui model *DataStorage* dan akan mendapatkan data *file* untuk ditampilkan pada *view* tampilan tersebut.



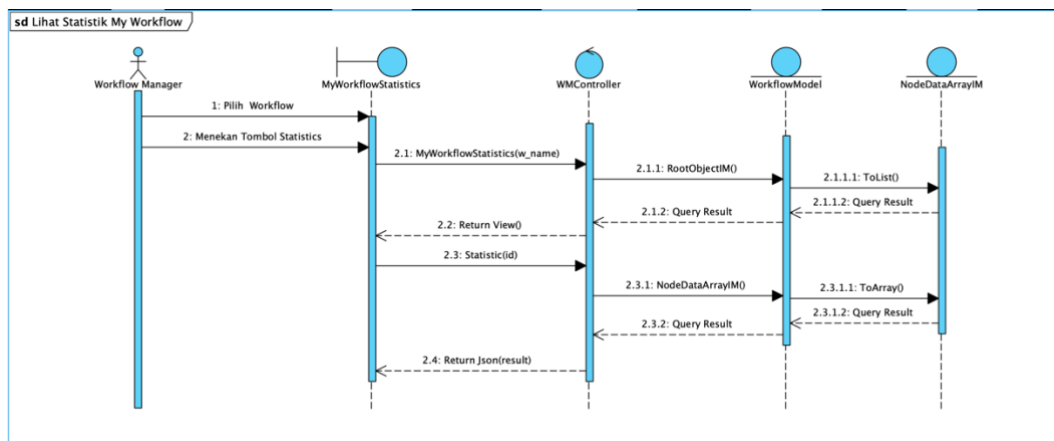


Gambar 5.2 Sequence Diagram Lihat Shared Files

5.1.1.3 Sequence Diagram Lihat Statistik My Workflow

Sequence Diagram lihat detail daftar template workflow yang terdapat pada Gambar 5.3 menggambarkan proses pada saat *workflow manager* melihat statistik dari *workflow* yang sudah diimplementasikan. Pada *sequence diagram* ini terdapat empat objek dengan aktor *workflow manager* yaitu dengan *boundary MyWorkflowStatistics*, terdapat satu *controller WMController*, dan dua model yaitu *WorkflowModel* dan *NodeDataArrayIM*. Dalam *sequence* ini diawali dengan aktor memilih *workflow* kemudian setelah dipilih aktor menekan tombol *statistics* yang memanggil *method* dari *WMController* yaitu *MyWorkflowStatistics(w_name)* yang berfungsi untuk mengambil nama dari *workflow* yang dipilih aktor sebelumnya lalu dibandingkan dengan data yang diperoleh dari hasil *query model workflow* lalu memanggil *model NodeDataArrayIM* dengan membandingkan nama tersebut, jika sama maka akan disimpan id *workflow* tersebut. Selanjutnya dari *WMController* akan melakukan *return* ke *boundary MyWorkflowStatistics* dengan data id yang telah didapatkan lalu memanggil *method Statistic(id)* dengan id adalah id yang sebelumnya didapatkan pada *query* lalu dibandingkan kembali ke *NodeDataArrayIM* jika terdapat id yang sesuai pada *database* lalu akan dikembalikan hasil tersebut dalam bentuk *Json* yang akan menampilkan statistik *bar chart* dari *workflow* tersebut.

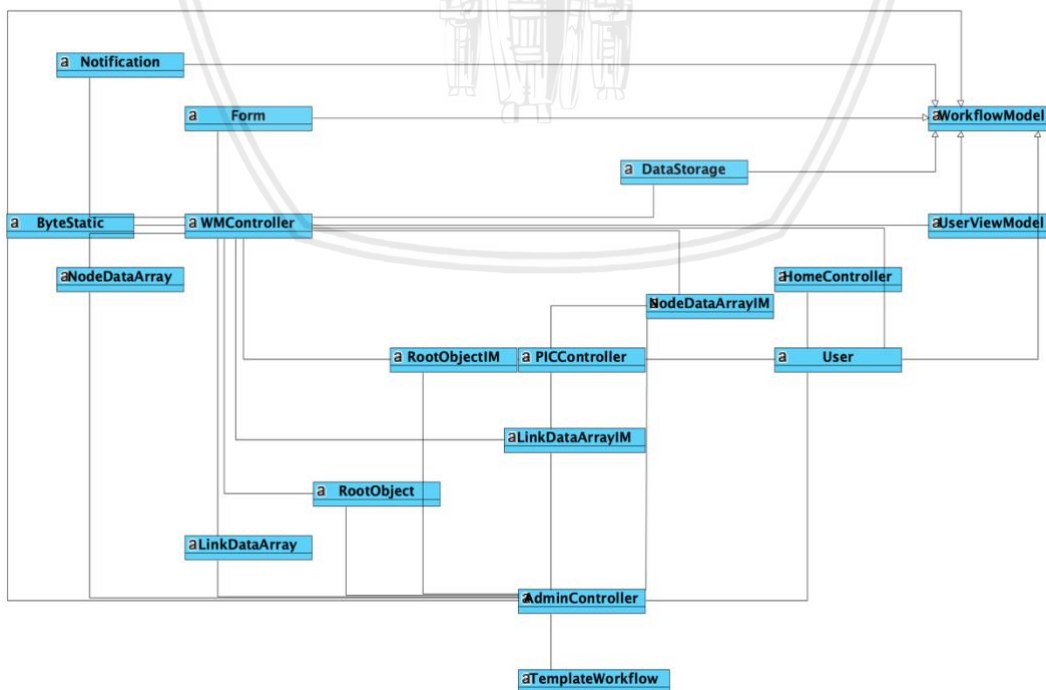




Gambar 5.3 Sequence Diagram Lihat Statistik My Workflow

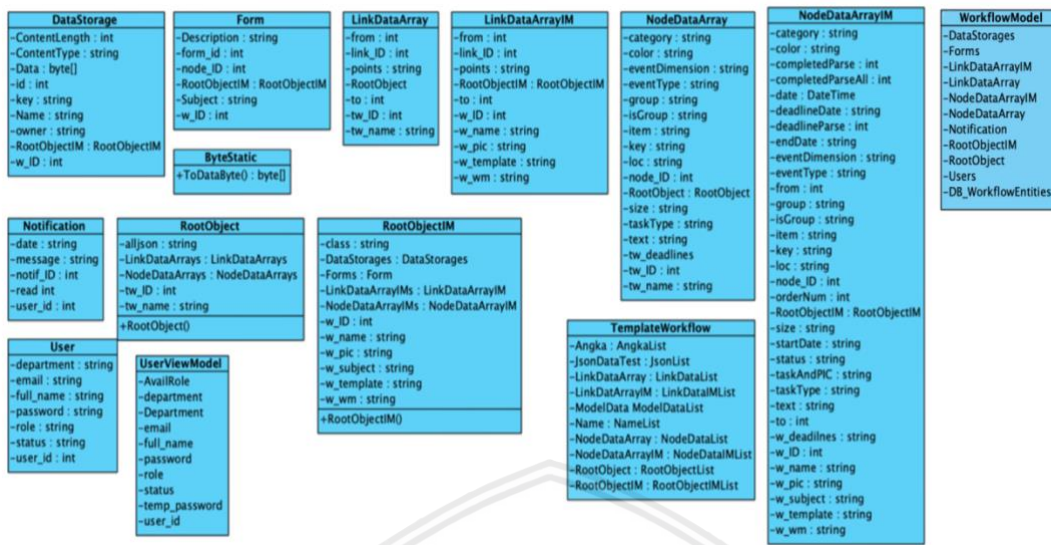
5.1.2 Perancangan Class Diagram

Pada perancangan *class diagram* akan dilakukan identifikasi mengenai *class*, *object*, dan interkasi pada sistem tersebut. Pada *sequence diagram* terdapat objek yang kemudian dijadikan menjadi sebuah *class*. Terdapat gambar perancangan umum *class diagram* untuk melihat hubungan antar *class* atau asosiasi. Pada setiap model dilakukan generalisasi terhadap *WorkflowModel*. Perancangan umum dapat dilihat pada Gambar 5.4 *Class diagram* yang terdapat pada sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia memiliki 14 *class* pada *model* dan 4 *class* pada *controller*. Sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia memiliki *class diagram model* yang digambarkan pada Gambar 5.5 dan *class diagram controller* pada Gambar 5.6.

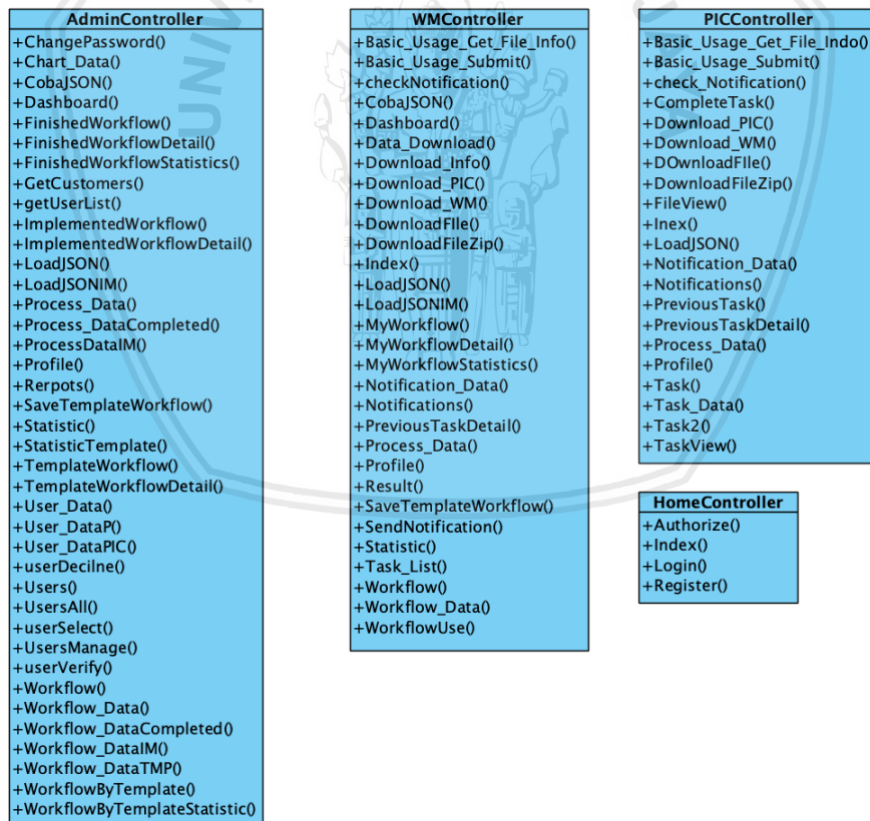


Gambar 5.4 Perancangan Umum Class Diagram





Gambar 5.5 Perancangan *Class Diagram Model*



Gambar 5.6 Perancangan *Class Diagram Controller*

5.1.3 Perancangan Komponen

Pada perancangan komponen terdapat komponen pada sistem yang terdapat rincian subsistem yang akan digambarkan. Selanjutnya pada

perancangan komponen terdapat algoritme pada sistem yang akan dijelaskan. Pada sistem ini terdapat sampel sebanyak 3 algoritme diantaranya adalah algoritme *TemplateWorkflowDetail*, *TaskView*, dan *MyWorkflowStatistics*.

5.1.3.1 Algoritme *TemplateWorkflowDetail*

Algoritme *TemplateWorkflowDetail* adalah algoritme yang berfungsi untuk menampilkan *template workflow* yang sebelumnya telah dibuat oleh administrator. Algoritme *TemplateWorkflowDetail* dapat dilihat pada Tabel 5.1.

Nama *Class* : AdminController

Nama *Method* : *TemplateWorkflowDetail()*

Algoritme :

Tabel 5.1 Algoritme *TemplateWorkflowDetail*

1	Inisialisasi objek <i>DB_WorkflowEntities</i> dengan nama <i>db</i>
2	Inisialisasi objek <i>RootObject</i> dengan nama <i>tempRoot</i>
3	Inisialisasi list objek <i>RootObject</i> dengan nama <i>result</i>
4	Mengisi var <i>root</i> dengan nilai dari method
5	<i>db.RootObjects.ToList()</i>
6	Melakukan perulangan sebanyak variable item dalam <i>root</i>
7	Melakukan cek kondisi jika <i>item.tw.name = tw.name</i>
8	Mengisi variable <i>ViewData["click"]</i> dengan nilai <i>item.tw_ID</i>
9	Mengisi variable <i>ViewData["Name"]</i> dengan nilai <i>item.tw_name</i>
10	Akhir cek kondisi
11	Akhir perulangan
12	Menampilkan halaman <i>templateWorkflowDetail</i>

5.1.3.2 Algoritme *TaskView*

Algoritme *TaskView* adalah algoritme yang berfungsi untuk menampilkan isi dari suatu task *person in charge* pada suatu *workflow*. Algoritme *TaskView* dapat dilihat pada Kode Program 5.2.

Nama *Class* : *PICController*

Nama *Method* : *TaskView()*

Algoritme :

Tabel 5.2 Algoritme *TaskView*

1	Mulai
2	Memanggil method <i>checkNotification</i>
3	Inisialisasi objek <i>DB_WorkflowEntities</i> dengan nama <i>db</i>
4	Inisialisasi objek <i>RootObjectIM</i> dengan nama <i>tempRoot</i>
5	Inisialisasi list objek <i>RootObjectIM</i> dengan nama <i>result</i>
6	Mengisi nilai var <i>root</i> dengan nilai <i>db.NodeDataArrayIms.ToList()</i>
7	Inisialisasi objek <i>NodeDataArrayIM</i> dengan nama <i>cellData</i>
8	Mengisi nilai var <i>tempID</i> dengan nilai 0
9	Melakukan perulangan sebanyak variable item di dalam <i>root</i>
10	Melakukan cek kondisi jika <i>item.w_name = w_name</i>
11	Mengisi nilai <i>ViewData["click"] = itwm.w_ID</i>
12	Mengisi nilai <i>ViewData["Name"] = item.w_name</i>
13	Mengisi nilai <i>cellData</i> dengan nilai <i>item</i>
14	Akhir cek kondisi
15	Akhir perulangan
16	Mengisi nilai <i>ViewData["Task"]</i> dengan nilai variable <i>text</i>



Tabel 5.2 Algoritme TaskView (lanjutan)

17	Inisialisasi objek NodeDataArrayIM dengan nama result2
18	Mengisi nilai var db.Configuration.ProxyCreationEnabled = false
19	Mengisi nilai var itemTemp dengan nilai db.NodeDataArrayIMs.ToArray()
20	Mengisi nilai tempwid dengan nilai 0
21	Mengisi nilai tempid dengan nilai 0
22	Melakukan perulangan sebanyak item di dalam db.NodeDaraArrayIMs
23	Melakukan cek kondisi jika item.text = text
24	Mengisi nilai tempwid dengan nilai item.w_ID
25	Mengisi nilai tempid dengan nilai item.orderNum
26	Akhir cek kondisi
27	Akhir perulangan
28	Mengisi nilai countPrevious dengan nilai 0
29	Melakukan perulangan sampai nilai var I bernilai kurang dari db.NodeDataArrayIMs.Count()
30	Melakukan cek kondisi jika itemTemp.ElementAt(i).orderNum lebih
31	kecil dari itemTemp.ElementAt(i).w_ID = tempwid dan itemTemp.ElementAt(i).category = "activity"
32	Melakukan increment variable countPrevious
33	Akhir kondisi cek kondisi
34	Akhir cek perulangan
35	Mengisi nilai ViewData["countPrev"] dengan nilai countPrevious
36	Memanggil method Download_WM dengan parameter cellData
37	Mengembalikan halaman TaskView

5.1.3.3 Algoritme MyWorkflowStatistics

Algoritme MyWorkflowStatistics adalah algoritme yang berfungsi untuk menampilkan statistik dalam bentuk *bar chart* berdasarkan *deadline* dan *day complete* dari suatu tugas *workflow* yang sudah diimplementasikan oleh *workflow manager*. Algoritme MyWorkflowStatistics dapat dilihat pada Kode Program 5.3.

Nama *Class* : WMController

Nama *Method* : MyWorkflowStatistics()

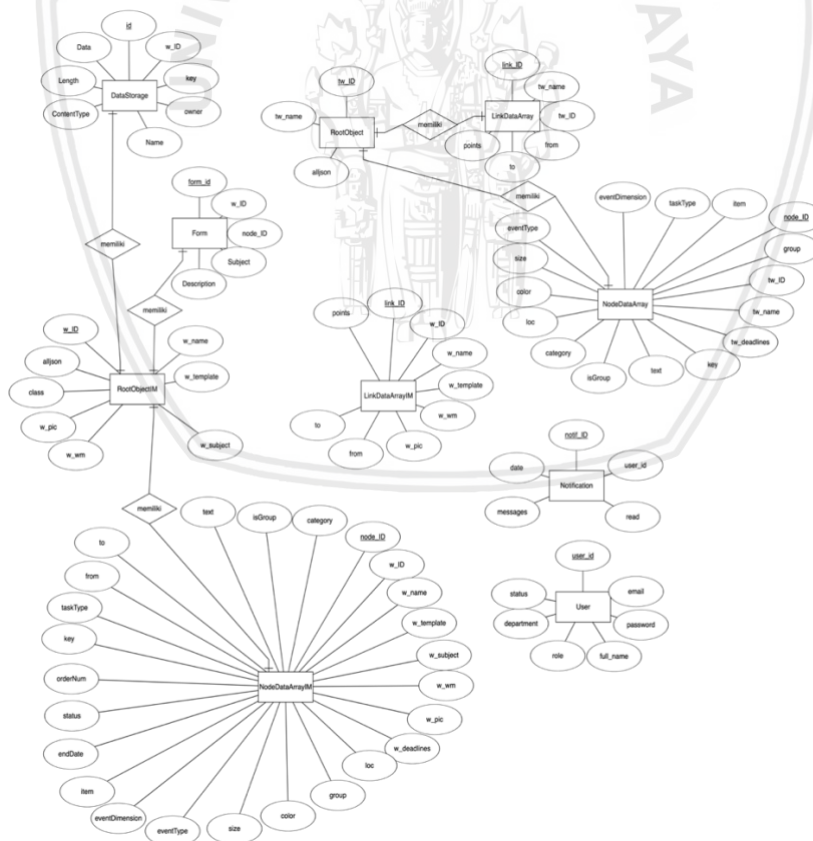
Algoritme :

Tabel 5.3 Algoritme MyWorkflowStatistics

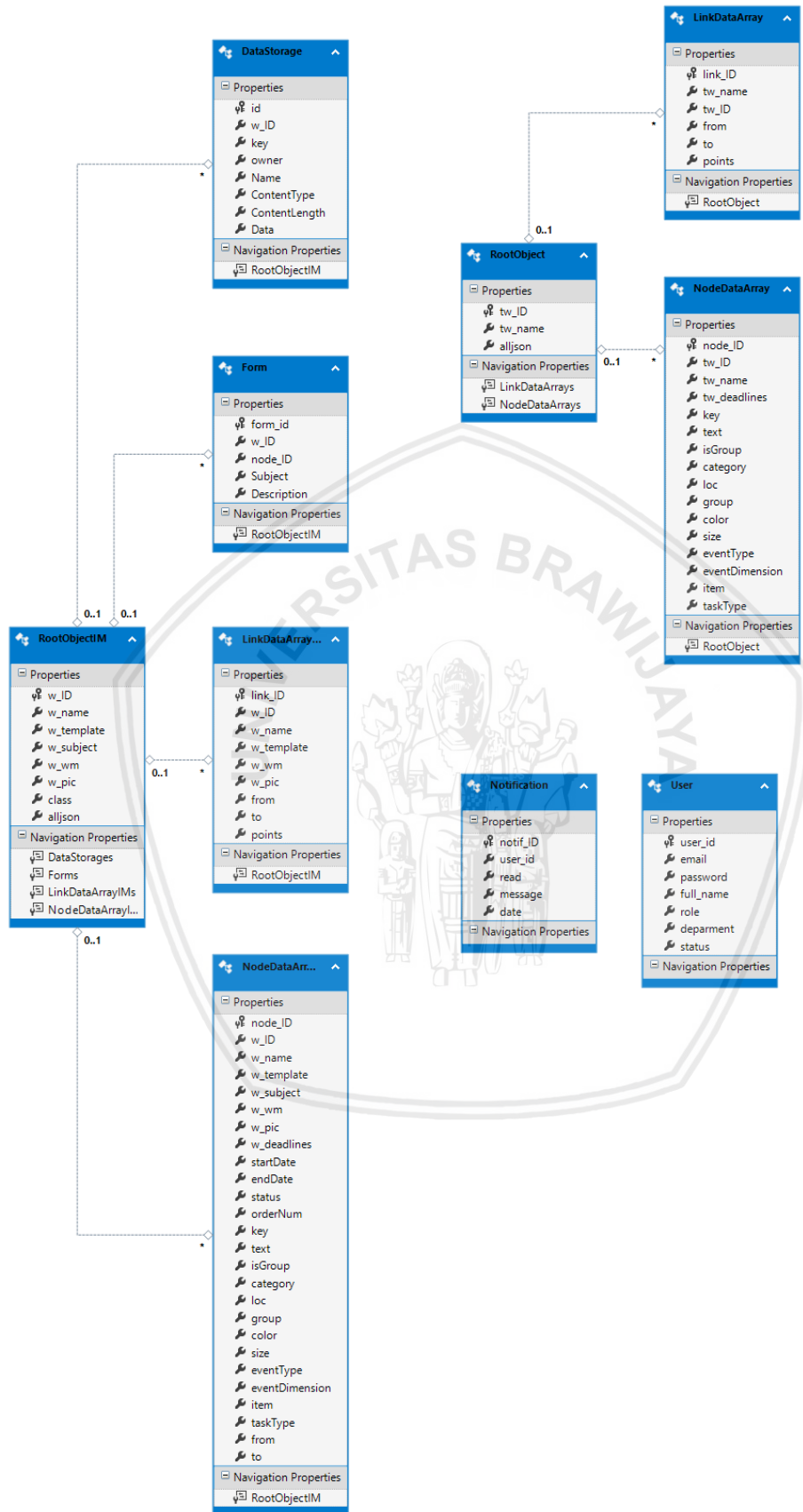
1	Mulai
2	Memanggil method session
3	Memanggil method checkNotification
4	Inisialisasi objek DB_WorkflowEntities dengan nama db
5	Inisialisasi list objek RootObjectIM dengan nama tempRoot
6	Inisialisasi list objek RootObjectIM dengan nama result
7	Mengisi nilai var root dengan nilai db.NodeDataArrayIMs.ToList()
8	Inisialisasi objek NodeDataArrayIM dengan nama cellData
9	Melakukan perulangan sebanyak variable item di dalam root
10	Melakukan cek kondisi jika item.w_name = w_name
11	Mengisi nilai ViewData["click"] = item.w_ID
12	Mengisi nilai ViewData["Name"] = item.w_name
13	Mengisi nilai cellData dengan nilai item
14	Akhir cek kondisi
15	Akhir perulangan
16	Menampilkan halaman MyWorkflowStatistics

5.1.4 Perancangan Basis Data

Perancangan basis data akan dibuat menggunakan *Entity Relationship Diagram* (ERD) dan juga *Physical Data Model*. Pada ERD ini terdapat entitas sebanyak sepuluh yaitu *DataStorage* yang akan menyimpan *file* berdasarkan *id*, *Form* yang berfungsi untuk menyimpan *form* yang diisi oleh *person in charge*, *LinkDataArray* untuk menyimpan data JSON dari posisi objek *template workflow*, *LinkDataArrayIM* untuk menyimpan data JSON dari posisi objek *workflow* yang terimplementasi, *NodeDataArray* untuk menyimpan data JSON dari bentuk dan jenis objek dari *template workflow*, *NodeDataArrayIM* untuk menyimpan data JSON dari bentuk dan jenis objek dari *workflow* terimplementasi, *Notifications* untuk menyimpan notifikasi dari *person in charge* dan *workflow manager*, *RootObject* berfungsi untuk menyimpan daftar *template workflow*, *RootObjectIM* berfungsi untuk menyimpan daftar *workflow* terimplementasi, dan *Users* untuk menyimpan daftar *user* yang ada pada sistem. Setelah perancangan ERD telah dilakukan maka akan dilakukan perancangan PDM yang memiliki jumlah tabel seperti pada perancangan ERD sebelumnya yaitu memiliki sebanyak sepuluh tabel. Selanjutnya perancangan basis data pada sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia digambarkan pada Gambar 5.7 dan *Physical Data Model* digambarkan pada Gambar 5.8.



Gambar 5.7 Entity Relationship Diagram Sistem



Gambar 5.8 Physical Data Model (PDM)

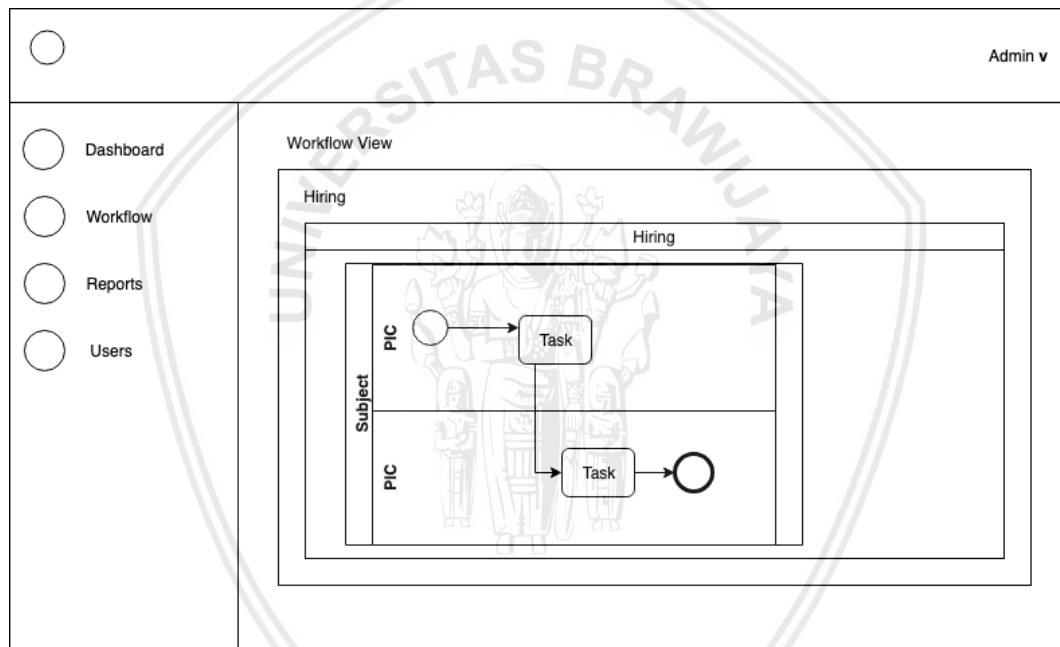


5.1.5 Perancangan Antarmuka

Pada perancangan antarmuka akan dijelaskan hasil perancangan dari halaman *website* pada sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia. Pada perancangan antarmuka ini terdapat tiga model antarmuka untuk dijadikan sampel. Gambar dari perancangan antarmuka digambarkan pada Gambar 5.9 Sampai Gambar 5.11.

5.1.5.1 Perancangan Antarmuka Lihat *Detail Template Workflow*

Pada perancangan antarmuka Lihat *Detail Template Workflow* adalah halaman untuk segmen admin untuk melihat tampilan dari *workflow* yang sebelumnya sudah dipilih dari daftar *template workflow*. Pada halaman ini *template workflow* sebelumnya telah dibuat oleh *admin*. Hasil perancangan antarmuka Lihat *Detail Template Workflow* digambarkan pada Gambar 5.9.



Gambar 5.9 Antarmuka *Lihat Detail Template Workflow*

5.1.5.2 Perancangan Antarmuka Lihat *Detail Tugas*

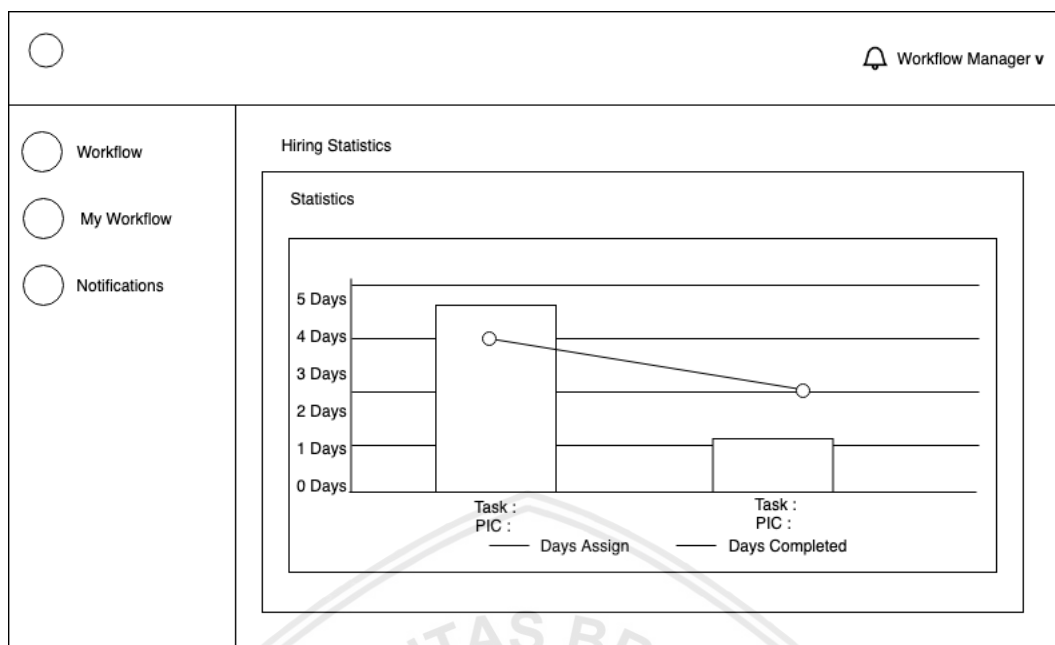
Pada perancangan antarmuka Lihat *Detail Tugas* adalah halaman untuk peran *person in charge* yang berfungsi untuk melihat daftar *file* yang diberikan oleh *workflow manager*, tugas sebelumnya, *workflow* dari tugas tersebut, dan *form* untuk menyelesaikan tugas tersebut. Pada halaman ini terdapat tombol untuk mengunduh *file* secara keseluruhan, tombol untuk melihat daftar tugas sebelumnya, dan tombol untuk melakukan *submit form* yang berisi *subject*, *description*, dan *file* sekaligus menyelesaikan tugas tersebut. Hasil dari perancangan antarmuka Lihat *Detail Tugas* digambarkan pada Gambar 5.10.

The screenshot shows a web interface for task management. At the top right, there is a notification bell icon and the text 'Person In Charge v'. On the left side, there is a sidebar with two radio button options: 'Task' and 'Notifications'. The main content area is titled 'Task Details' and is divided into several sections. The top section is 'Shared Files', which contains a 'Download All' button. To its right is the 'Previous Task' section, which contains a 'View Previous Task' button. Below these are two text input fields labeled 'Subject' and 'Description'. Underneath is an 'Upload File' section with a 'Select Files..' button. At the bottom right of the main content area, there is a 'Complete' button.

Gambar 5.10 Antarmuka Lihat *Detail* Tugas

5.1.5.3 Perancangan Antarmuka Lihat Statistik *My Workflow*

Pada perancangan antarmuka Lihat Statistik *My Workflow* adalah halaman untuk peran *workflow manager* yang berfungsi untuk menampilkan laporan dengan bentuk *bar chart* tentang tugas yang terdapat di dalam *workflow* yang diimplementasikan oleh *workflow manager*. Halaman statistik tersebut menampilkan *chart* berdasarkan hari penugasan, hari penyelesaian, dan informasi tentang *person in charge* dan nama tugas. Dari tampilan tersebut bisa terlihat proses mana yang sesuai tenggat waktu ataupun melebihi tenggat waktu sekaligus dapat melihat *person in charge* yang dipilih untuk suatu proses tugas. *Days Assign* adalah tenggat waktu yang diberikan sebelumnya oleh *workflow manager*, sedangkan untuk *days completed* adalah hari saat *person in charge* menyelesaikan tugas tersebut. Pada halaman ini jika garis pada *days completed* melebihi garis *days assign* maka proses tersebut melebihi tenggat waktu, namun jika dibawah garis *days assign* maka proses tersebut tepat waktu atau lebih cepat dari waktu yang telah diberikan oleh *workflow manager* sebelumnya. Hasil dari perancangan antarmuka Lihat Statistik *My Workflow* digambarkan pada Gambar 5.11.



Gambar 5.11 Antarmuka Lihat Statistik *My Workflow*

5.2 Implementasi Sistem

Setelah proses perancangan selesai maka selanjutnya adalah tahap implementasi sistem. Pada implementasi sistem akan dibahas mengenai hasil implementasi sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia. Pada bagian implementasi akan dibahas mengenai spesifikasi sistem, implementasi basis data, kode program dan implementasi antarmuka.

5.2.1 Spesifikasi Sistem

5.2.1.1 Spesifikasi Perangkat Keras

Pada sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia terdapat spesifikasi perangkat keras yang digunakan pada saat pembuatan sistem. Spesifikasi yang digunakan pada sistem ini dijelaskan pada tabel 5.4.

Tabel 5.4 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>System Model</i>	Acer Aspire E 14
<i>Processor</i>	Intel(R) Core(TM) i7-5500U CPU @2.40GHz
<i>Memory (RAM)</i>	4,00 GB DDR3 L
<i>Display</i>	NVIDIA GeForce 940m
<i>Hardisk</i>	1000 GB HDD



5.2.1.2 Spesifikasi Perangkat Lunak

Dalam sistem ini juga terdapat spesifikasi perangkat lunak untuk membantu proses pembuatan sistem *workflow* manajemen kerja PT Chevron Pacific Indonesia. Spesifikasi yang digunakan pada sistem ini dijelaskan pada tabel 5.5.

Tabel 5.5 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Windows 10 Home 64-Bit
Bahasa Pemrograman	HTML, Javascript, C#
Perkakaks Bantu	Visual Studio Community 2017 Microsoft SQL Server Management Studio

5.2.2 Implementasi Basis Data

Setelah tahap perancangan basis data dilakukan implementasi data dari hasil rancangan sebelumnya. Pada implementasi basis data ini terdapat tujuh buah tabel. *Query* dari pembuatan implementasi ke tujuh tabel tersebut dapat dilihat pada tabel 5.6 sampai 5.12.

Tabel 5.6 Implementasi Tabel DataStorage

1	CREATE TABLE DataStorage (
2	[id] [int] IDENTITY(1,1) NOT NULL,
3	[w_ID] INT,
4	[key] VARCHAR(20),
5	[owner] VARCHAR(20),
6	[Name] [varchar](50) NOT NULL,
7	[ContentType] [nvarchar](200) NOT NULL,
8	[ContentLength] INT NOT NULL,
9	[Data] [varbinary](max) NOT NULL,
10	PRIMARY KEY (id),
11	FOREIGN KEY (w_ID) REFERENCES RootObjectIM(w_ID)
12);

Tabel 5.7 Implementasi Tabel Form

1	CREATE TABLE Form (
2	[form_id] [int] IDENTITY(1,1) NOT NULL,
3	[w_ID] INT,
4	[node_ID] INT,
5	[Subject] VARCHAR (20),
6	[Description] VARCHAR (1000),
7	PRIMARY KEY (form_id),
8	FOREIGN KEY (w_ID) REFERENCES RootObjectIM(w_ID),
9);



Tabel 5.8 Implementasi Tabel LinkdataArray

1	CREATE TABLE LinkdataArray (
2	[link_ID] INT IDENTITY(1,1) NOT NULL,
3	[tw_name] VARCHAR (35),
4	[tw_ID] INT,
5	[from] INT,
6	[to] INT,
7	[points] VARCHAR(max),
8	PRIMARY KEY (link_ID),
9	FOREIGN KEY (tw_ID) REFERENCES RootObject(tw_ID)
10);

Tabel 5.9 Implementasi Tabel LinkdataArrayIM

1	CREATE TABLE LinkdataArrayIM (
2	[link_ID] INT IDENTITY(1,1) NOT NULL,
3	[w_ID] INT,
4	[w_name] VARCHAR (35),
5	[w_template] VARCHAR (35),
6	[w_wm] VARCHAR (35),
7	[w_pic] VARCHAR (35),
8	[from] INT,
9	[to] INT,
10	[points] VARCHAR(max),
11	PRIMARY KEY (link_ID),
12	FOREIGN KEY (w_ID) REFERENCES RootObjectIM(w_ID)
13);

Tabel 5.10 Implementasi Tabel NodedataArray

1	CREATE TABLE NodedataArray (
2	[node_ID] INT IDENTITY(1,1) NOT NULL,
3	[tw_ID] INT,
4	[tw_name] VARCHAR (35),
5	[tw_deadlines] VARCHAR (15),
6	[key] VARCHAR(20),
7	[text] VARCHAR(20),
8	[isGroup] VARCHAR(20),
9	[category] VARCHAR(20),
10	[loc] VARCHAR(37),
11	[group] VARCHAR(20),
12	[color] VARCHAR(20),
13	[size] VARCHAR(22),
14	[eventType] VARCHAR(20),
15	[eventDimension] VARCHAR(20),
16	[item] VARCHAR(12),
17	[taskType] VARCHAR(20),
18	PRIMARY KEY (node_ID),
19	FOREIGN KEY (tw_ID) REFERENCES RootObject(tw_ID)
20);

Tabel 5.11 Implementasi Tabel NodeDataArrayIM

1	C CREATE TABLE NodeDataArrayIM (
2	[node_ID] INT IDENTITY(1,1) NOT NULL,
3	[w_ID] INT,
4	[w_name] VARCHAR (35),
5	[w_template] VARCHAR (35),
6	[w_subject] VARCHAR (35),
7	[w_wm] VARCHAR (35),
8	[w_pic] VARCHAR (35),
9	[w_deadlines] VARCHAR (15),
10	[startDate] VARCHAR (15),
11	[endDate] VARCHAR (15),
12	[status] VARCHAR (20),
13	[orderNum] INT,
14	[key] VARCHAR(20),
15	[text] VARCHAR(20),
16	[isGroup] VARCHAR(20),
17	[category] VARCHAR(20),
18	[loc] VARCHAR(37),
19	[group] VARCHAR(20),
20	[color] VARCHAR(20),
21	[size] VARCHAR(22),
22	[eventType] VARCHAR(20),
23	[eventDimension] VARCHAR(20),
24	[item] VARCHAR(12),
25	[taskType] VARCHAR(20),
26	[from] INT,
27	[to] INT,
28	PRIMARY KEY (node_ID),
29	FOREIGN KEY (w_ID) REFERENCES RootObjectIM(w_ID)
30);

Tabel 5.12 Implementasi Tabel Notifications

1	CREATE TABLE Notifications(
2	[notif_ID] INT IDENTITY(1,1) NOT NULL,
3	[user_id] [int],
4	[read] [int],
5	[message] [varchar] (50),
6	[date] [varchar] (15),
7	PRIMARY KEY (notif_ID),
8)

Tabel 5.13 Implementasi Tabel RootObject

1	CREATE TABLE RootObject (
2	[tw_ID] INT IDENTITY(1,1) NOT NULL,
3	[tw_name] VARCHAR (35),
4	[alljson] VARCHAR(max),
5	PRIMARY KEY (tw_ID),
6);

Tabel 5.14 Implementasi Tabel RootObjectIM

1	CREATE TABLE RootObjectIM (
2	[w_ID] INT IDENTITY(1,1) NOT NULL,
3	[w_name] VARCHAR (35),
4	[w_template] VARCHAR (35),
5	[w_subject] VARCHAR (35),
6	[w_wm] VARCHAR (35),
7	[w_pic] VARCHAR (35),
8	[class] VARCHAR(18),
9	[alljson] VARCHAR(max),
10	PRIMARY KEY (w_ID),
11);

Tabel 5.15 Implementasi Tabel Users

1	CREATE TABLE Users (
2	[user_id] [int] IDENTITY(1,1) NOT NULL,
3	[email] [varchar] (50) NULL,
4	[password] [varchar] (50) NULL,
5	[full_name] [varchar] (50) NULL,
6	[role] [varchar] (50) NULL,
7	[deparment] [varchar] (50) NULL,
8	[status] [varchar] (10) NULL,
	PRIMARY KEY (user_id),
)

5.2.3 Implementasi Kode Program

5.2.3.1 Implementasi Kode Program TemplateWorkflowDetail()

Pada kode program *TemplateWorkflowDetail* adalah fungsi pada class AdminController yang berfungsi untuk menyimpan *template workflow* ke dalam *database*. Algoritme *saveTemplateWorkflow* dapat dilihat pada Tabel 5.13.

Nama *Class* : AdminController

Nama *Method* : TemplateWorkflowDetail()

Kode Program :

Tabel 5.16 Kode Program *TemplateWorkflowDetail()*

1	<code>public ActionResult TemplateWorkflowDetail(String tw_name)</code>
2	<code>{</code>
3	
4	
5	<code> DB_WorkflowEntities db = new DB_WorkflowEntities();</code>
6	<code> RootObject tempRoot = new RootObject();</code>
7	<code> List<RootObject> result = new List<RootObject>();</code>
8	<code> List<RootObject> result = new List<RootObject>();</code>
9	<code> var root = db.RootObjects.ToList();</code>
10	
11	
12	<code> foreach (var item in root)</code>
13	<code> {</code>
14	
15	<code> if (item.tw_name == tw_name)</code>
16	<code> {</code>
17	<code> ViewData["click"] = item.tw_ID;</code>
18	<code> ViewData["Name"] = item.tw_name;</code>
19	<code> ViewData["Name"] = item.tw_name;</code>
20	<code> }</code>
21	
22	
23	<code> }</code>
24	<code> return View();</code>
25	<code>}</code>

5.2.3.2 Implementasi Kode Program *MyWorkflowStatistics*

Pada kode program *MyWorkflowStatistics* adalah fungsi pada class *WMController* yang berfungsi untuk mengimplementasikan *template workflow* yang dibuat oleh admin. Fungsi *MyWorkflowStatistics* dapat dilihat pada Tabel 5.14.

Nama *Class* : *WMController*

Nama *Method* : *MyWorkflowStatistics()*

Kode Program :

Tabel 5.17 Kode Program MyWorkflowStatistics

1	<code>public ActionResult MyWorkflowStatistics(String w_name)</code>
2	<code>{</code>
3	
4	<code> checkSession();</code>
5	<code> checkNotification();</code>
6	<code> DB_WorkflowEntities db = new DB_WorkflowEntities();</code>
7	<code> RootObjectIM tempRoot = new RootObjectIM();</code>
8	<code> List<RootObjectIM> result = new List<RootObjectIM>();</code>
9	<code> var root = db.NodeDataArrayIMs.ToList();</code>
10	<code> NodeDataArrayIM cellData = new NodeDataArrayIM();</code>
11	
12	<code> foreach (var item in root)</code>
13	<code> {</code>
14	<code> if (item.w_name == w_name)</code>
15	<code> {</code>
16	<code> ViewData["click"] = item.w_ID;</code>
17	<code> ViewData["Name"] = item.w_name;</code>
18	<code> cellData = item;</code>
19	<code> }</code>
20	
21	<code> }</code>
22	<code> Download_WM(cellData);</code>
23	
24	<code> return View();</code>
25	<code>}</code>
26	

5.2.3.3 Implementasi Kode Program TaskView

Pada kode program CompleteTask adalah fungsi pada class PICController yang berfungsi untuk menyelesaikan suatu proses yang didapatkan *person in charge* kemudian akan memberikan notifikasi kepada *person in charge* selanjutnya ataupun memberikan notifikasi kepada *workflow manager*. Fungsi taskDone dapat dilihat pada Tabel 5.15.

Nama Class : PICController

Nama Method : TaskView()

Kode Program :

Tabel 5.18 Kode Program TaskView

1	<code>public ActionResult TaskView(string w_name, string text)</code>
2	<code>{</code>
3	<code> checkNotification();</code>
4	<code> DB_WorkflowEntities db = new DB_WorkflowEntities();</code>
5	<code> RootObjectIM tempRoot = new RootObjectIM();</code>
6	<code> List<RootObjectIM> result = new List<RootObjectIM>();</code>
7	<code> var root = db.NodeDataArrayIMs.ToList();</code>
8	<code> NodeDataArrayIM cellData = new NodeDataArrayIM();</code>
9	<code> var tempId = 0;</code>
10	<code> foreach (var item in root)</code>

Tabel 5.18 Kode Program TaskView (lanjutan)

```
11      {
12          if (item.w_name == w_name)
13          {
14              ViewData["click"] = item.w_ID;
15              ViewData["Name"] = item.w_name;
16              cellData = item;
17          }
18      }
19      ViewData["Task"] = text;
20
21          List<NodeDataArrayIM> result2 = new
22 List<NodeDataArrayIM>();
23      db.Configuration.ProxyCreationEnabled = false;
24
25      var itemTemp = db.NodeDataArrayIMs.ToArray();
26
27      int? tempwid = 0;
28      int? tempid = 0;
29
30      foreach (var item in db.NodeDataArrayIMs)
31      {
32          if (item.text == text)
33          {
34              tempwid = item.w_ID;
35              tempid = item.orderNum;
36          }
37      }
38
39      var countPrevious = 0;
40
41      for (int i = 0; i < db.NodeDataArrayIMs.Count(); i++)
42      {
43          if (itemTemp.ElementAt(i).orderNum < tempid &&
44 itemTemp.ElementAt(i).w_ID == tempwid &&
45 itemTemp.ElementAt(i).category == "activity")
46          {
47              countPrevious++;
48          }
49      }
50
51      ViewData["countPrev"] = countPrevious;
52
53
54      Download_WM(cellData);
55      return View();
56  }
57
```

5.2.4 Implementasi Antarmuka

5.2.4.1 Implementasi Antarmuka Lihat *Detail Template Workflow*

Pada implementasi antarmuka Lihat *Detail Template Workflow* adalah halaman untuk segmen admin untuk melihat tampilan dari *workflow* yang sebelumnya sudah dipilih dari daftar *template workflow*. Pada halaman ini *template workflow* sebelumnya telah dibuat oleh *admin*. Hasil perancangan antarmuka Lihat *Detail Template Workflow* digambarkan pada Gambar 5.12

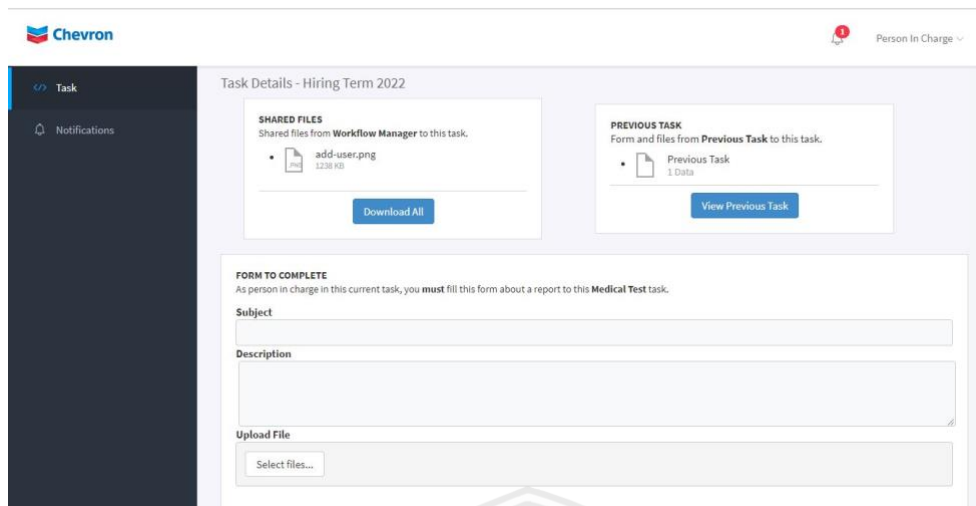


Gambar 5.12 Implementasi Antarmuka Lihat *Detail Template Workflow*

5.2.4.2 Implementasi Antarmuka Lihat *Detail Tugas*

Pada implmentasi antarmuka Lihat *Detail Tugas* adalah halaman untuk peran *person in charge* yang berfungsi untuk melihat daftar *file* yang diberikan oleh *workflow manager*, tugas sebelumnya, *workflow* dari tugas tersebut, dan *form* untuk menyelesaikan tugas tersebut. Pada halaman ini terdapat tombol untuk mengunduh *file* secara keseluruhan, tombol untuk melihat daftar tugas sebelumnya, dan tombol untuk melakukan *submit form* yang berisi *subject*, *description*, dan *file* sekaligus menyelesaikan tugas tersebut. Hasil dari perancangan antarmuka Lihat *Detail Tugas* digambarkan pada Gambar 5.13.

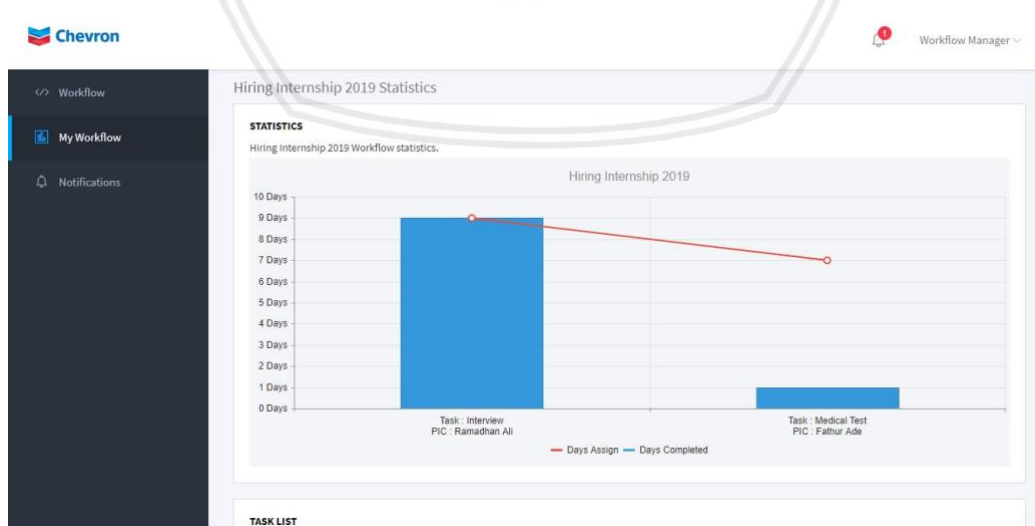




Gambar 5.13 Implementasi Antarmuka Lihat *Detail* Tugas

5.2.4.3 Implementasi Antarmuka Lihat Statistik *My Workflow*

Pada implementasi antarmuka Lihat Statistik *My Workflow* adalah halaman untuk peran *workflow manager* yang berfungsi untuk menampilkan laporan dengan bentuk *bar chart* tentang tugas yang terdapat di dalam *workflow* yang diimplementasikan oleh *workflow manager*. Halaman statistik tersebut menampilkan *chart* berdasarkan hari penugasan, hari penyelesaian, dan informasi tentang *person in charge* dan nama tugas. Dari tampilan tersebut bisa terlihat proses mana yang sesuai tenggat waktu ataupun melebihi tenggat waktu sekaligus dapat melihat *person in charge* yang dipilih untuk suatu proses tugas. *Days Assign* adalah tenggat waktu yang diberikan sebelumnya oleh *workflow manager*, sedangkan untuk *days completed* adalah hari saat *person in charge* menyelesaikan tugas tersebut. Hasil dari perancangan antarmuka Lihat Statistik *My Workflow* digambarkan pada Gambar 5.14.



Gambar 5.14 Implementasi Antarmuka Lihat Statistik *My Workflow*



BAB 6 PENGUJIAN

6.1 Pengujian Unit

Pengujian unit bertujuan untuk melakukan uji komponen, objek, dan klas dari hasil perancangan. Pada tahap ini dilakukan pengujian untuk menyesuaikan hasil *algoritme* yang telah diimplementasikan dengan hasil analisis kebutuhan sebelumnya. Dalam melakukan pengujian unit dilakukan dengan metode *basis path testing*. Pengujian unit dilakukan pada tiga sampel yaitu *TemplateWorkflowDetails*, *TaskView*, dan *MyWorkflowStatistics*.

6.1.1 Pengujian Unit *TemplateWorkflowDetails()*

1. Pseudocode

Pengujian unit *method TemplateWorkflowDetails* yang berasal dari *class AdminController* akan dijelaskan pada Tabel 6.1.

Tabel 6.1 Pseudocode Algoritme *TemplateWorkflowDetails*

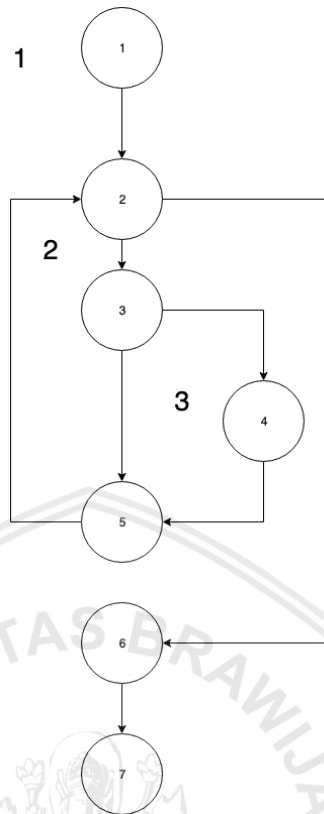
1	Inisialisasi objek <i>DB_WorkflowEntities</i> dengan nama <i>db</i>	}	①
2	Inisialisasi objek <i>RootObject</i> dengan nama <i>tempRoot</i>		
3	Inisialisasi list objek <i>RootObject</i> dengan nama <i>result</i>		
4	Mengisi var <i>root</i> dengan nilai dari method <i>db.RootObjects.ToList()</i>		
5	Melakukan perulangan sebanyak variable <i>item</i> dalam <i>root</i>	}	②
6	Melakukan cek kondisi jika <i>item.tw.name = tw.name</i>		
7	Mengisi variable <i>ViewData["click]</i> dengan nilai <i>item.tw_ID</i>	}	④
8	Mengisi variable <i>ViewData["Name]</i> dengan nilai <i>item.tw_name</i>		
9	Akhir cek kondisi	}	⑤
10	Akhir perulangan		
11	Menampilkan halaman <i>TemplateWorkflowDetails</i>		⑦

2. Basis Path Testing

2.1. Flow Graph

Berdasarkan algoritme *TemplateWorkflowDetails* akan menjadi dasar untuk pembuatan *flowgraph* untuk *method TemplateWorkflowDetails* yang dapat dilihat pada Gambar 6.1.





Gambar 6.1 Flowgraph Method Template Workflow Details

2.2. Cyclomatic Complexity

- $V(G) = \text{jumlah region} = 3$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 8 - 7 + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

2.3. Independent Path

- Jalur 1: 1-2-6-7
- Jalur 2: 1-2-3-5-2-6-7
- Jalur 3: 1-2-3-4-5-2-6-7

Tabel 6.2 Hasil Pengujian Unit Method *TemplateWorkflowDetails*

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
1	<i>Class Driver</i> memanggil method <i>TemplateWorkflowDetails()</i> dengan data pada <i>tw_name = null</i> yang dicocokkan dengan data pada <i>database RootObject</i> dan belum terdapat data pada <i>database RootObject</i> .	Menjalankan method <i>TemplateWorkflowDetails()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop".	Menjalankan method <i>TemplateWorkflowDetails()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop".	Valid
2	<i>Class Driver</i> memanggil method <i>TemplateWorkflowDetails()</i> dengan data pada <i>tw_name = "retirement"</i> yang dicocokkan dengan data pada <i>database RootObject</i> dan tidak ada data "retirement" pada <i>database RootObject</i> .	Menjalankan method <i>TemplateWorkflowDetails()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".	Menjalankan method <i>TemplateWorkflowDetails()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".	Valid
3	<i>Class Driver</i> memanggil method <i>TemplateWorkflowDetails()</i> dengan data pada <i>tw_name = "Hiring"</i> yang dicocokkan dengan data pada <i>database RootObject</i> dan terdapat data "Hiring" pada <i>database RootObject</i> .	Menjalankan method <i>TemplateWorkflowDetails()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1". "Memasuki cek kondisi 1"	M Menjalankan method <i>TemplateWorkflowDetails()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1". "Memasuki cek kondisi 1"	Valid

6.1.2 Pengujian Unit *TaskView()*

1. Pseudocode

Pengujian unit method *MyWorkflowStatistics* yang berasal dari class *PICController* akan dijelaskan pada Tabel 6.3.

Tabel 6.3 Pseudocode Algoritme TaskView

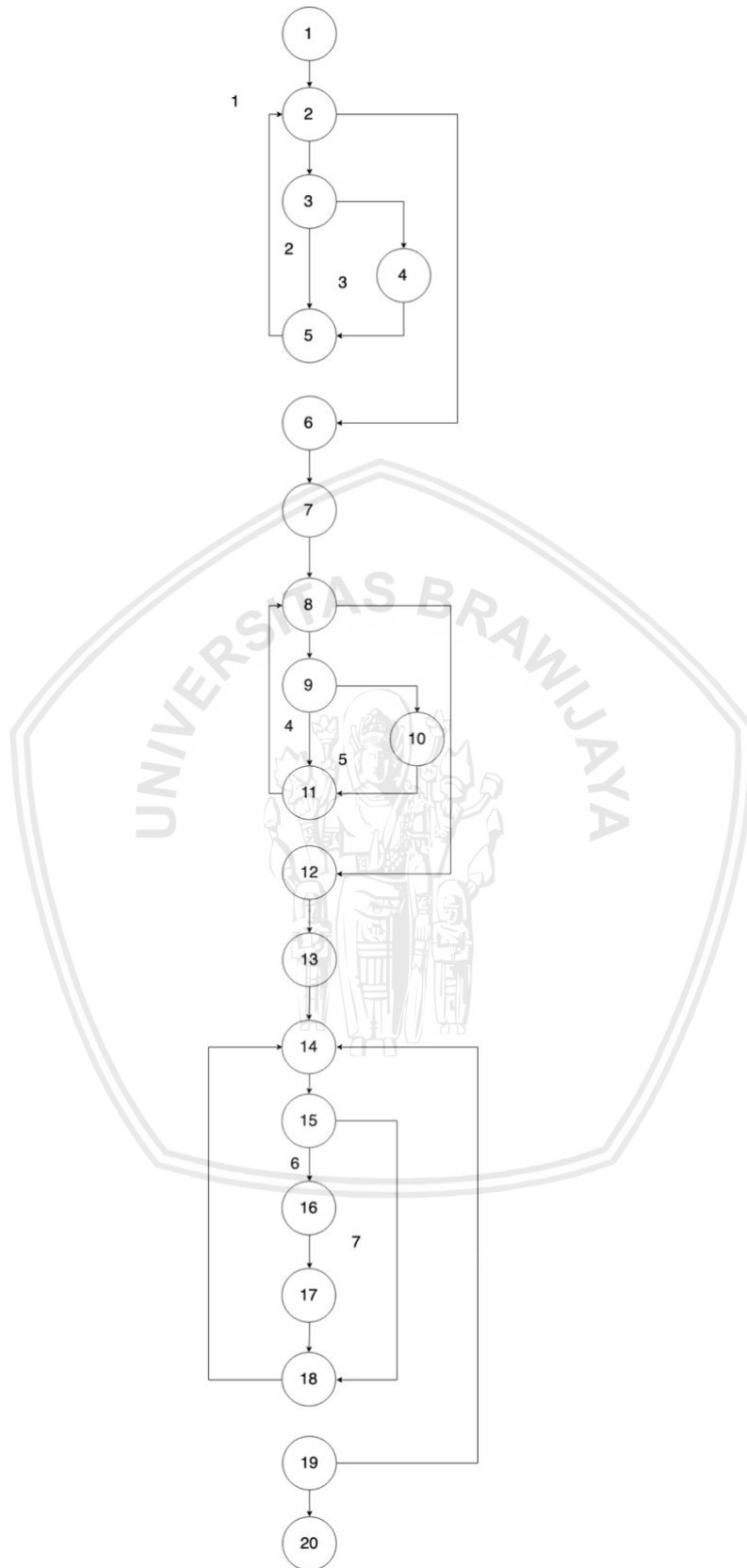
1	Memanggil method checkNotification	}	①
2	Inisialisasi objek DB_WorkflowEntities dengan nama db		
3	Inisialisasi objek RootObjectIM dengan nama tempRoot		
4	Inisialisasi list objek RootObjectIM dengan nama result		
5	Mengisi nilai var root dengan nilai db.NodeDataArrayIms.ToList()		
6	Inisialisasi object NodeDataArrayIM dengan nama cellData	}	②
7	Mengisi nilai var tempId dengan nilai 0		
8	Melakukan perulangan sebanyak variable item di dalam root	}	③
9	Melakukan cek kondisi jika item.w_name = w_name		
10	Mengisi nilai ViewData["click"] = itwm.w_ID		
11	Mengisi nilai ViewData["Name"] = item.w_name		
12	Mengisi nilai cellData dengan nilai item	}	④
13	Akhir cek kondisi		
14	Akhir perulangan	}	⑤
15	Mengisi nilai ViewData["Task"] dengan nilai variable text		
16	Inisialisasi objek NodeDataArrayIM dengan nama result2	}	⑥
17	Mengisi nilai var db.Configuration.ProxyCreationEnabled = false		
18	Mengisi nilai var itemTemp dengan nilai db.NodeDataArrayIMs.ToArray()		
19	Mengisi nilai tempwid dengan nilai 0		
20	Mengisi nilai tempid dengan nilai 0		
21	Melakukan perulangan sebanyak item di dalam db.NodeDaraArrayIMs	}	⑦
22	Melakukan cek kondisi jika item.text = text		
23	Mengisi nilai tempwid dengan nilai item.w_ID	}	⑧
24	Mengisi nilai tempid dengan nilai item.orderNum		
25	Akhir cek kondisi	}	⑨
26	Akhir perulangan		
27	Mengisi nilai countPrevious dengan nilai 0	}	⑩
28	Melakukan perulangan sampai nilai var I bernilai kurang dari db.NodeDataArrayIMs.Count()		
29	Melakukan cek kondisi jika	}	⑪
30	itemTemp.ElementAt(i).orderNum lebih kecil dari itemTemp.ElementAt(i).w_ID = tempwid dan itemTemp.ElementAt(i).category = "activity"		
31	Melakukan increment variable countPrevious	}	⑫
32	Melakukan increment variable countPrevious		
33	Akhir kondisi cek kondisi	}	⑬
34	Akhir cek perulangan		
35	Mengisi nilai ViewData["countPrev"] dengan nilai countPrevious	}	⑭
36	Memanggil method Download_WM dengan parameter cellData		
37	Mengembalikan halaman TaskView		
38			

2. Basis Path Testing

2.1. Flow Graph

Berdasarkan algoritme *TaskView* akan menjadi dasar untuk pembuatan *flowgraph* untuk method *TaskView* yang dapat dilihat pada Gambar 6.2.





Gambar 6.2 Flowgraph Method TaskView

2.2. Cyclomatic Complexity

- $V(G) = \text{jumlah region} = 7$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 24 - 19 + 2 = 7$
- $V(G) = \text{jumlah predicate node} + 1 = 6 + 1 = 7$

2.3. Independent Path

- Jalur 1: 1-2-6-8-12-13-14-18-19
- Jalur 2: 1-2-3-5-2-6-7-8-12-13-14-18-19
- Jalur 3: 1-2-3-4-5-2-6-7-8-12-113-14-18-19
- Jalur 4: 1-2-6-7-8-9-11-8-12-13-14-18-19
- Jalur 5: 1-2-6-7-8-9-10-11-8-12-13-14-18-19
- Jalur 6: 1-2-6-7-8-12-13-14-15-17-14-18-19
- Jalur 7: 1-2-6-7-8-12-13-14-15-16-17-14-18-19

Tabel 6.4 Hasil Pengujian Unit Method *TaskView*

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
1	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = null</i> yang dicocokkan dan <i>text = null</i> dengan data pada <i>database NodeDataArrayIM</i> dan belum terdapat data pada <i>database NodeDataArrayIM</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Sebelum Loop Kedua" "Sebelum Loop ketiga".	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Sebelum Loop Kedua" "Sebelum Loop ketiga".	Valid
2	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = null</i> yang dicocokkan dan <i>text = null</i> dengan data pada <i>database NodeDataArrayIM</i> dan sudah terdapat data pada <i>database NodeDataArrayIM</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop Kedua" "Sebelum Loop ketiga"	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop Kedua" "Sebelum Loop ketiga"	Valid

Tabel 6.4 Hasil Pengujian Unit Method *TaskView* (lanjutan)

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
3	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = "Hiring"</i> yang dicocokkan dan <i>text = null</i> dengan data pada <i>database NodeDataArrayIM</i> dan sudah terdapat data pada <i>database NodeDataArrayIM</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "cek kondisi loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "Sebelum Loop ketiga"	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "cek kondisi loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "Sebelum Loop ketiga"	Valid
4	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = "Hiring"</i> yang dicocokkan dan <i>text = null</i> dengan data pada <i>database NodeDataArrayIM</i> dan belum terdapat data pada <i>database NodeDataArrayIM</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "Sebelum Loop ketiga"	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "Sebelum Loop ketiga"	Valid
5	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = "Hiring"</i> yang dicocokkan dan <i>text = "Interview Test"</i> dengan data pada <i>database NodeDataArrayIM</i> dan sudah terdapat data pada <i>database NodeDataArrayIM</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "cek kondisi loop kedua"	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "cek kondisi loop kedua" "Sebelum Loop ketiga"	Valid



Tabel 6.4 Hasil Pengujian Unit Method *TaskView* (lanjutan)

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
		"Sebelum Loop ketiga"		
6	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = "Hiring"</i> yang dicocokkan dan <i>text = "Interview Test"</i> dengan data pada <i>database NodeDataArrayIM</i> , sudah terdapat data pada <i>database NodeDataArrayIM</i> dan data pada <i>itemTemp.category = null</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "Sebelum Loop ketiga" "1 Loop kedua"	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "Sebelum Loop ketiga" "1 Loop kedua"	Valid
7	<i>Class Driver</i> memanggil method <i>TaskView()</i> dengan data pada <i>w_name = "Hiring"</i> yang dicocokkan dan <i>text = "Interview Test"</i> dengan data pada <i>database NodeDataArrayIM</i> , sudah terdapat data pada <i>database NodeDataArrayIM</i> dan data pada <i>itemTemp.category = activity</i> .	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "Sebelum Loop ketiga" "1 Loop ketiga" "cek kondisi loop kedua"	Menjalankan <i>method TaskView()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "1 Loop pertama" "Sebelum Loop" "Sebelum Loop Kedua" "1 Loop kedua" "Sebelum Loop ketiga" "1 Loop ketiga" "cek kondisi loop kedua"	Valid

6.1.3 Pengujian Unit *MyWorkflowStatistics()*

1. Pseudocode

Pengujian unit *method MyWorkflowStatistics* yang berasal dari *class WMController* akan dijelaskan pada Tabel 6.5.



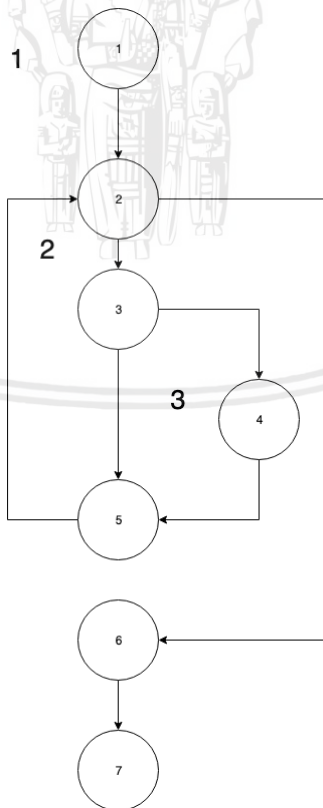
Tabel 6.5 Algoritme MyWorkflowStatistics()

1	Memanggil method session	}	①
2	Memanggil method checkNotification		
3	Inisialisasi objek DB_WorkflowEntities dengan nama db		
4	Inisialisasi list objek RootObjectIM dengan nama tempRoot		
5	Inisialisasi list objek RootObjectIM dengan nama result		
6	Mengisi nilai var root dengan nilai db.NodedataArrayIMs.ToList()	}	②
7	Inisialisasi objek NodedataArrayIM dengan nama cellData		
8	Melakukan perulangan sebanyak variable item di dalam root		③
9	Melakukan cek kondisi jika item.w_name = w_name	}	④
10	Mengisi nilai ViewData["click"] = itwm.w_ID		
11	Mengisi nilai ViewData["Name"] = item.w_name		⑤
12	Mengisi nilai cellData dengan nilai item		⑥
13	Akhir cek kondisi		⑦
14	Akhir perulangan		⑧
15	Menampilkan halaman MyWorkflowStatistics		⑨

2. Basis Path Testing

2.1. Flow Graph

Berdasarkan algoritme *TemplateWorkflowDetails* akan menjadi dasar untuk pembuatan *flowgraph* untuk method *TemplateWorkflowDetails* dapat dilihat pada Gambar 6.3.



Gambar 6.3 Flowgraph Method MyWorkflowStatistics

2.2. Cyclomatic Complexity

- $V(G) = \text{jumlah region} = 3$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 8 - 7 + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

2.3. Independent Path

- Jalur 1: 1-2-6-7
- Jalur 2: 1-2-3-5-2-6-7
- Jalur 3: 1-2-3-4-5-2-6-7

Tabel 6.6 Hasil Pengujian Unit Method *MyWorkflowStatistics*

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
1	<i>Class Driver</i> memanggil method <i>MyWorkflowStatistics()</i> dengan data pada <i>w_name = null</i> yang dicocokkan dengan data pada <i>database NodeDataArrayIM</i> dan belum terdapat data pada <i>database RootObject</i> .	Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop".	Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop".	Valid
2	<i>Class Driver</i> memanggil method <i>MyWorkflowStatistics()</i> dengan data pada <i>w_name = "retirement"</i> yang dicocokkan dengan data pada <i>database NodeDataArrayIM</i> dan tidak ada data "retirement" pada <i>database RootObject</i> .	Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".	Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".	Valid



Tabel 6.6 Hasil Pengujian Unit Method *MyWorkflowStatistics* (lanjutan)

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
3	<p><i>Class Driver</i> memanggil method <i>MyWorkflowStatistics ()</i> dengan data pada <i>w_name = "Hiring"</i> yang dicocokkan dengan data pada <i>database NodeDataArrayIM</i> dan terdapat data "<i>Hiring</i>" pada <i>database RootObject</i>.</p>	<p>Menjalankan method <i>MyWorkflowStatistics ()</i> dan menampilkan pesan di <i>console "Sebelum Loop"</i></p> <p>"Loop 1".</p> <p>"Memasuki cek kondisi 1"</p>	<p>Menjalankan method <i>MyWorkflowStatistics ()</i> dan menampilkan pesan di <i>console "Sebelum Loop"</i></p> <p>"Loop 1".</p> <p>"Memasuki cek kondisi 1"</p>	Valid

6.2 Pengujian Integrasi

1. Pseudocode

Pengujian integrasi method *TemplateWorkflowDetails* yang berasal dari class *WMController* akan dijelaskan pada Tabel 6.7.

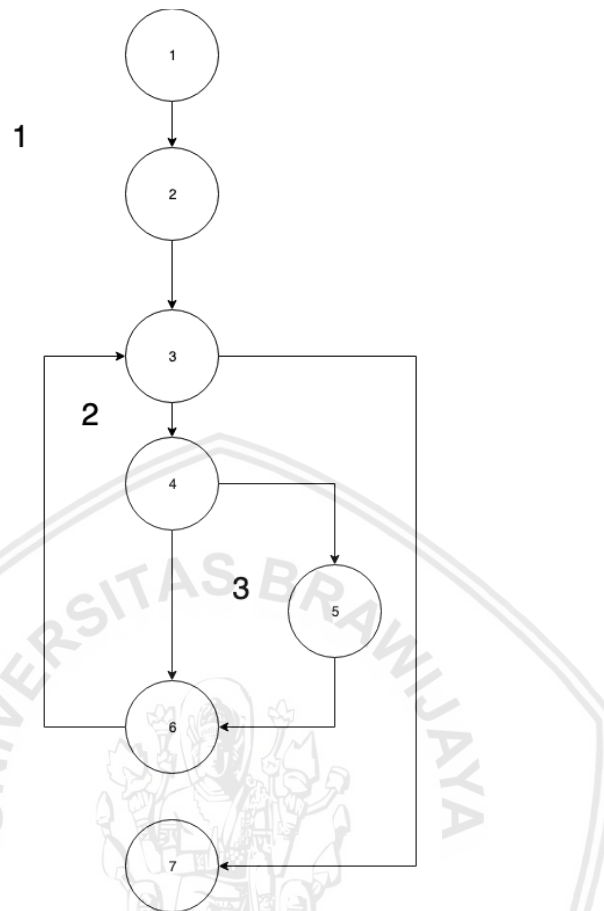
Tabel 6.7 Pseudocode Algoritme *MyWorkflowStatistics ()*

1	Memanggil method <i>session</i>	
2	Memanggil method <i>checkNotification</i>	
3	Inisialisasi objek <i>DB_WorkflowEntities</i> dengan nama <i>db</i>	
4	Inisialisasi list objek <i>RootObjectIM</i> dengan nama <i>tempRoot</i>	①
5	Inisialisasi list objek <i>RootObjectIM</i> dengan nama <i>result</i>	
6	Mengisi nilai <i>var root</i> dengan nilai <i>db.NodeDataArrayIMs.ToList ()</i>	
7	Inisialisasi objek <i>NodeDataArrayIM</i> dengan nama <i>cellData</i>	
8	Melakukan perulangan sebanyak variable <i>item</i> di dalam <i>root</i>	②
9	Melakukan cek kondisi jika <i>item.w_name = w_name</i>	③
10	Mengisi nilai <i>ViewData["click"] = itm.w_ID</i>	④
11	Mengisi nilai <i>ViewData["Name"] = item.w_name</i>	
12	Mengisi nilai <i>cellData</i> dengan nilai <i>item</i>	
13	Akhir cek kondisi	⑤
14	Akhir perulangan	⑥
15	Menampilkan halaman <i>MyWorkflowStatistics</i>	⑦

2. Basis Path Testing

2.1. Flow Graph

Berdasarkan algoritme *MyWorkflowStatistics* akan menjadi dasar untuk pembuatan *flowgraph* untuk method *MyWorkflowStatistics* yang dapat dilihat pada Gambar 6.4.



Gambar 6.4 Flowgraph Method MyWorkflowStatistics

2.2. Cyclomatic Complexity

- $V(G) = \text{jumlah region} = 3$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 8 - 7 + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

2.3. Independent Path

- Jalur 1: 1-2-6-7
- Jalur 2: 1-2-3-5-2-6-7
- Jalur 3: 1-2-3-4-5-2-6 -7

Tabel 6.8 Hasil Pengujian Integrasi

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
1	<p><i>Class Driver</i> memanggil method <i>MyWorkflowStatistics()</i> yang memanggil method <i>checkSession()</i> dan method <i>checkNotification()</i> dengan data pada <i>tw_name = null</i> yang dicocokkan dengan data pada <i>database RootObject</i> dan belum terdapat data pada <i>database RootObject</i>.</p>	<p>Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop".</p>	<p>Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop".</p>	Valid
2	<p><i>Class Driver</i> memanggil method <i>MyWorkflowStatistics()</i> yang memanggil method <i>checkSession()</i> dan method <i>checkNotification()</i> dengan data pada <i>tw_name = "retirement"</i> yang dicocokkan dengan data pada <i>database RootObject</i> dan tidak ada data "retirement" pada <i>database RootObject</i>.</p>	<p>Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".</p>	<p>Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".</p>	Valid
3	<p><i>Class Driver</i> memanggil method <i>MyWorkflowStatistics()</i> yang memanggil method <i>checkSession()</i> dan method <i>checkNotification()</i> dengan data pada <i>tw_name = "Hiring"</i></p>	<p>Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1". "Memasuki cek kondisi 1"</p>	<p>Menjalankan method <i>MyWorkflowStatistics()</i> dan menampilkan pesan di <i>console</i> "Sebelum Loop" "Loop 1".</p>	Valid



Tabel 6.8 Hasil Pengujian Integrasi (lanjutan)

Jalur	Prosedur Uji	Hasil pengujian yang diharapkan	Hasil Pengujian	Status
	pada <i>database RootObject</i> .		"Memasuki cek kondisi 1"	Valid

6.3 Pengujian Validasi

6.3.1 Pengujian Validasi Login

1. Kasus Uji Berhasil Login

Tabel 6.9 Kasus Uji Berhasil *Login*

Nama Kasus Uji	<i>Login</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>email</i> dan <i>password</i> dengan lengkap. 2. Aktor menekan tombol <i>login</i>. 3. Sistem memeriksa <i>email</i> dan <i>password</i> aktor.
Hasil yang diharapkan	Sistem menampilkan halaman utama sesuai aktor yang dikenali.
Hasil	Sistem menampilkan halaman utama sesuai aktor yang dikenali.
Status	Valid.

2. Kasus Uji Login Alternatif 1: Jika *email* atau *password* salah

Tabel 6.10 Kasus Uji Login Alternatif 1

Nama Kasus Uji	Login Alternatif 1: Jika <i>email</i> atau <i>password</i> salah
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>email</i> dan <i>password</i> dengan salah. 2. Aktor menekan tombol <i>login</i>. 3. Sistem memeriksa <i>email</i> dan <i>password</i> aktor.
Hasil yang diharapkan	Sistem menampilkan menampilkan pesan <i>Wrong email or Password</i> dan menampilkan halaman <i>login</i> kembali.

Tabel 6.10 Kasus Uji Login Alternatif 1 (lanjutan)

Hasil	Sistem menampilkan pesan <i>Wrong email or Password</i> dan menampilkan halaman <i>login</i> kembali.
Status	Valid.

3 Kasus Uji Login Alternatif 2: Jika *email* atau *password* belum diverifikasi *admin*

Tabel 6.11 Kasus Uji Login Alternatif 2

Nama Kasus Uji	Uji Login Alternatif 2: Jika <i>email</i> atau <i>password</i> belum diverifikasi <i>admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>email</i> dan <i>password</i> dengan yang belum diverifikasi oleh <i>admin</i>. 2. Aktor menekan tombol <i>login</i>. 3. Sistem memeriksa <i>email</i> dan <i>password</i> aktor.
Hasil yang diharapkan	Sistem menampilkan halaman utama sesuai aktor yang dikenali.
Hasil	Sistem menampilkan pesan <i>Waiting for Administrator Confirmation</i> dan menampilkan halaman <i>login</i> kembali.
Status	Valid.

6.3.2 Pengujian Validasi Register

1. Kasus Uji Berhasil Register

Tabel 6.12 Kasus Uji Berhasil Register

Nama Kasus Uji	<i>Register</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>field full name, email, password, password confirmation</i>, memilih <i>department</i> dan <i>role</i> dengan lengkap. 2. Aktor menekan tombol <i>register</i>. 3. Sistem memeriksa seluruh <i>field, email, password</i> aktor 4. Sistem menampilkan halaman <i>register</i> kembali.

Tabel 6.12 Kasus Uji Berhasil Register (lanjutan)

Hasil yang diharapkan	Sistem menampilkan pesan <i>Waiting for Administator Confirmation</i> .
Hasil	Sistem menampilkan pesan <i>Waiting for Administator Confirmation</i> .
Status	Valid.

2. Kasus Uji Register Alternatif 1 : Jika terdapat *field* yang tidak terisi

Tabel 6.13 Kasus Uji Register Alternatif 1

Nama Kasus Uji	Register Alternatif 1 : Jika terdapat <i>field</i> yang tidak terisi
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>field full name, email, password, password confirmation</i>, memilih <i>department</i> dan <i>role</i> tidak lengkap. 2. Aktor menekan tombol <i>register</i>. 3. Sistem memeriksa seluruh <i>field, email, password</i> aktor 4. Sistem menampilkan halaman <i>register</i> kembali.
Hasil yang diharapkan	Sistem menampilkan pesan <i>You must fill all data</i> .
Hasil	Sistem menampilkan pesan <i>You must fill all data</i> .
Status	Valid.

3 Kasus Uji Register Alternatif 2: Jika *email* akun sudah terdaftar.

Tabel 6.14 Kasus Uji Register Alternatif 2

Nama Kasus Uji	Register Alternatif 2: Jika <i>email</i> akun sudah terdaftar
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>field full name, email, password, password confirmation</i>, memilih <i>department</i> dan <i>role</i> tidak lengkap. 2. Aktor menekan tombol <i>register</i>. 3. Sistem memeriksa seluruh <i>field, email, password</i> aktor.



Tabel 6.14 Kasus Uji Register Alternatif 2 (lanjutan)

	4. Sistem menampilkan halaman <i>register</i> kembali.
Hasil yang diharapkan	Sistem menampilkan pesan <i>Email already taken</i> .
Hasil	Sistem menampilkan pesan <i>Email already taken</i> .
Status	Valid.

4 Kasus Uji Register Alternatif 3: Jika *field password* dan *password confirmation* tidak sesuai.

Tabel 6.15 Kasus Uji Register Alternatif 3

Nama Kasus Uji	Register Alternatif 3: Jika <i>field password</i> dan <i>password confirmation</i> tidak sesuai.
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>field full name, email, password, password confirmation</i>, memilih <i>department</i> dan <i>role</i> dengan <i>password</i> dan <i>password confirmation</i> tidak sesuai. 2. Aktor menekan tombol <i>register</i>. 3. Sistem memeriksa seluruh <i>field, email, password</i> aktor 4. Sistem menampilkan halaman <i>register</i> kembali.
Hasil yang diharapkan	Sistem menampilkan pesan <i>password confirmation didn't match</i> .
Hasil	Sistem menampilkan pesan <i>password confirmation didn't match</i> .
Status	Valid.

6.3.3 Pengujian Validasi Buat *Workflow*

1. Kasus Uji Berhasil Buat *Workflow*

Tabel 6.16 Kasus Uji Berhasil Buat *Workflow*

Nama Kasus Uji	Buat <i>Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menggeser <i>template</i> BPMN ke dalam <i>field</i> pekerjaan. 2. Aktor membuat dan menghubungkan setiap proses dalam BPMN.



Tabel 6.16 Kasus Uji Berhasil Buat *Workflow* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 3. Aktor memberikan nama proses dan mengisikan tenggat waktu pada setiap proses. 4. Aktor menekan tombol <i>save as</i> 5. Sistem menampilkan <i>pop up field</i> nama <i>template workflow</i>. 6. Aktor menekan tombol OK. 7. Sistem menampilkan pesan <i>The name that you entered is 'nama template workflow'</i> 8. Aktor menekan tombol OK.
Hasil yang diharapkan	Sistem meyimpan <i>template workflow</i> yang telah dibuat oleh aktor.
Hasil	Sistem meyimpan <i>template workflow</i> yang telah dibuat oleh aktor.
Status	Valid.

2. Kasus Uji Buat *Workflow* Alternatif 1: Jika nama *workflow template* sudah ada

Tabel 6.17 Kasus Uji Buat *Workflow* Alternatif 1

Nama Kasus Uji	Buat <i>Workflow</i> Alternatif 1: Jika nama <i>workflow template</i> sudah ada
Prosedur	<ol style="list-style-type: none"> 1. Aktor menggeser <i>template BPMN</i> ke dalam <i>field</i> pekerjaan. 2. Aktor membuat dan menghubungkan setiap proses dalam BPMN. 3. Aktor memberikan nama proses dan mengisikan tenggat waktu pada setiap proses. 4. Aktor menekan tombol <i>save as</i>. 5. Sistem menampilkan <i>pop up field</i> nama <i>template workflow</i>. 6. Aktor menekan tombol OK. 7. Sistem menampilkan pesan <i>The name that you entered is 'nama template workflow'</i>. 8. Aktor menekan tombol OK.



Tabel 6.17 Kasus Uji Buat *Workflow* Alternatif 1 (lanjutan)

Hasil yang diharapkan	Jika nama <i>template workflow</i> sudah ada, maka akan memunculkan pesan <i>Workflow name already used</i> .
Hasil	Jika nama <i>template workflow</i> sudah ada, maka akan memunculkan pesan <i>Workflow name already used</i> .
Status	Valid.

6.3.4 Pengujian Validasi Lihat Petunjuk Cepat

1. Kasus Uji Berhasil Lihat Petunjuk Cepat

Tabel 6.18 Kasus Uji Berhasil Lihat Petunjuk Cepat

Nama Kasus Uji	Lihat Petunjuk Cepat
Prosedur	1. Aktor menekan kalimat <i>quick guide</i> .
Hasil yang diharapkan	Sistem akan menampilkan petunjuk cepat membuat <i>workflow</i> .
Hasil	Sistem akan menampilkan petunjuk cepat membuat <i>workflow</i> .
Status	Valid.

6.3.5 Pengujian Validasi Pilih Tenggat Waktu

1. Kasus Uji Berhasil Pilih Tenggat Waktu

Tabel 6.19 Kasus Uji Berhasil Pilih Tenggat Waktu

Nama Kasus Uji	Pilih Tenggat Waktu
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan objek <i>note</i> yang sudah dibuat pada <i>field</i> pekerjaan. 2. Sistem menampilkan tampilan untuk memilih jumlah tenggat waktu dalam satuan hari. 3. Aktor memilih banyaknya tenggat waktu. 4. Aktor menekan tombol <i>select</i>.
Hasil yang diharapkan	Sistem akan mengubah tenggat waktu pada objek <i>note</i> yang telah ditentukan oleh admin.
Hasil	Sistem akan mengubah tenggat waktu pada objek <i>note</i> yang telah ditentukan oleh admin.
Status	Valid.

6.3.6 Pengujian Validasi *Undo* Objek

1. Kasus Uji Berhasil *Undo* Objek

Tabel 6.20 Kasus Uji Berhasil *Undo* Objek

Nama Kasus Uji	<i>Undo</i> Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu edit. 2. Aktor menekan menu <i>undo</i>.
Hasil yang diharapkan	Sistem akan melakukan <i>undo</i> pekerjaan pada <i>field</i> pekerjaan.
Hasil	Sistem akan melakukan <i>undo</i> pekerjaan pada <i>field</i> pekerjaan.
Status	Valid.

6.3.7 Pengujian Validasi *Redo* Objek

1. Kasus Uji Berhasil *Redo* Objek

Tabel 6.21 Kasus Uji Berhasil *Redo* Objek

Nama Kasus Uji	<i>Undo</i> Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu edit. 2. Aktor menekan menu <i>redo</i>.
Hasil yang diharapkan	Sistem akan melakukan <i>redo</i> pekerjaan pada <i>field</i> pekerjaan.
Hasil	Sistem akan melakukan <i>redo</i> pekerjaan pada <i>field</i> pekerjaan.
Status	Valid.

6.3.8 Pengujian Validasi *Cut* Objek

1. Kasus Uji Berhasil *Cut* Objek

Tabel 6.22 Kasus Uji Berhasil *Cut* Objek

Nama Kasus Uji	<i>Cut</i> Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu edit. 3. Aktor menekan menu <i>cut</i>.
Hasil yang diharapkan	Sistem akan melakukan <i>cut</i> pada objek yang telah dipilih.

Tabel 6.22 Kasus Uji Berhasil *Cut* Objek (lanjutan)

Hasil	Sistem akan melakukan <i>cut</i> pada objek yang telah dipilih.
Status	Valid.

6.3.9 Pengujian Validasi *Copy* Objek

1. Kasus Uji Berhasil *Copy* Objek

Tabel 6.23 Kasus Uji Berhasil *Copy* Objek

Nama Kasus Uji	<i>Copy</i> Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu edit. 3. Aktor menekan menu <i>copy</i>.
Hasil yang diharapkan	Sistem akan melakukan <i>copy</i> pada objek yang telah dipilih.
Hasil	Sistem akan melakukan <i>copy</i> pada objek yang telah dipilih.
Status	Valid.

6.3.10 Pengujian Validasi *Paste* Objek

1. Kasus Uji Berhasil *Paste* Objek

Tabel 6.24 Kasus Uji Berhasil *Paste* Objek

Nama Kasus Uji	<i>Paste</i> Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu edit. 2. Aktor menekan menu <i>paste</i>.
Hasil yang diharapkan	Sistem akan melakukan <i>paste</i> dari hasil <i>copy</i> atau <i>cut</i> objek sebelumnya.
Hasil	Sistem akan melakukan <i>paste</i> dari hasil <i>copy</i> atau <i>cut</i> objek sebelumnya.
Status	Valid.

6.3.11 Pengujian Validasi *Delete* Objek

1. Kasus Uji Berhasil *Delete* Objek

Tabel 6.25 Kasus Uji Berhasil *Delete* Objek

Nama Kasus Uji	<i>Delete</i> Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu edit. 3. Aktor menekan menu <i>delete</i>.
Hasil yang diharapkan	Sistem akan melakukan <i>delete</i> pada objek yang dipilih sebelumnya.
Hasil	Sistem akan melakukan <i>delete</i> pada objek yang dipilih sebelumnya.
Status	Valid.

6.3.12 Pengujian Validasi Pilih Seluruh Objek

1. Kasus Uji Berhasil Pilih Seluruh Objek

Tabel 6.26 Kasus Uji Berhasil Pilih Seluruh Objek

Nama Kasus Uji	Pilih Seluruh Objek
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu edit. 2. Aktor menekan menu <i>select all</i>.
Hasil yang diharapkan	Sistem akan memilih seluruh objek yang ada pada <i>field</i> pekerjaan.
Hasil	Sistem akan memilih seluruh objek yang ada pada <i>field</i> pekerjaan.
Status	Valid.

6.3.13 Pengujian Validasi Pindah Posisi Objek Kiri

1. Kasus Uji Berhasil Pindah Posisi Objek Kiri

Tabel 6.27 Kasus Uji Berhasil Pindah Posisi Objek Kiri

Nama Kasus Uji	Pindah Posisi Objek Kiri
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>left sides</i>.

Tabel 6.27 Kasus Uji Berhasil Pindah Posisi Objek Kiri (lanjutan)

Hasil yang diharapkan	Sistem akan mengubah posisi objek yang dipilih ke sebelah kiri <i>field</i> pekerjaan.
Hasil	Sistem akan mengubah posisi objek yang dipilih ke sebelah kiri <i>field</i> pekerjaan.
Status	Valid.

6.3.14 Pengujian Validasi Pindah Posisi Objek Kanan

1. Kasus Uji Berhasil Pindah Posisi Objek Kanan

Tabel 6.28 Kasus Uji Berhasil Pindah Posisi Objek Kanan

Nama Kasus Uji	Pindah Posisi Objek Kanan
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>right sides</i>.
Hasil yang diharapkan	Sistem akan mengubah posisi objek yang dipilih ke sebelah kanan <i>field</i> pekerjaan.
Hasil	Sistem akan mengubah posisi objek yang dipilih ke sebelah kanan <i>field</i> pekerjaan.
Status	Valid.

6.3.15 Pengujian Validasi Pindah Posisi Objek Atas

1. Kasus Uji Berhasil Pindah Posisi Objek Atas

Tabel 6.29 Kasus Uji Berhasil Pindah Posisi Objek Atas

Nama Kasus Uji	Pindah Posisi Objek Atas
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>tops</i>.
Hasil yang diharapkan	Sistem akan mengubah posisi objek yang dipilih ke sebelah atas <i>field</i> pekerjaan.
Hasil	Sistem akan mengubah posisi objek yang dipilih ke sebelah atas <i>field</i> pekerjaan.
Status	Valid.

6.3.16 Pengujian Validasi Pindah Posisi Objek Bawah

1. Kasus Uji Berhasil Pindah Posisi Objek Bawah

Tabel 6.30 Kasus Uji Berhasil Pindah Posisi Objek Bawah

Nama Kasus Uji	Pindah Posisi Objek Bawah
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan satu atau lebih objek. 2. Aktor memilih menu <i>align</i>. 3. Aktor menekan menu <i>bottoms</i>.
Hasil yang diharapkan	Sistem akan mengubah posisi objek yang dipilih ke sebelah bawah <i>field</i> pekerjaan.
Hasil	Sistem akan mengubah posisi objek yang dipilih ke sebelah atas <i>bawah</i> pekerjaan.
Status	Valid.

6.3.17 Pengujian Validasi Tampil Kisi

1. Kasus Uji Berhasil Tampil Kisi

Tabel 6.31 Kasus Uji Berhasil Tampil Kisi

Nama Kasus Uji	Tampil Kisi
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih menu <i>options</i>. 2. Aktor menekan menu <i>grid</i>.
Hasil yang diharapkan	Sistem akan menampilkan kisi pada <i>field</i> pekerjaan.
Hasil	Sistem akan menampilkan kisi pada <i>field</i> pekerjaan.
Status	Valid.

6.3.18 Pengujian Lihat Jumlah *Template Workflow*

1. Kasus Uji Berhasil Lihat Jumlah *Template Workflow*

Tabel 6.32 Kasus Uji Berhasil Lihat Jumlah *Template Workflow*

Nama Kasus Uji	Lihat Jumlah <i>Template Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>workflow</i>.
Hasil yang diharapkan	Sistem menampilkan jumlah <i>template workflow</i> yang ada pada sistem.

Tabel 6.32 Kasus Uji Berhasil Lihat Jumlah *Template Workflow* (lanjutan)

Hasil	Sistem menampilkan jumlah <i>template workflow</i> yang ada pada sistem.
Status	Valid.

6.3.19 Pengujian Lihat Jumlah *Implemented Workflow*

1. Kasus Uji Berhasil Lihat Jumlah *Implemented Workflow*

Tabel 6.33 Kasus Uji Berhasil Lihat Jumlah *Implemented Workflow*

Nama Kasus Uji	Lihat Jumlah <i>Implemented Workflow</i>
Prosedur	1. Aktor membuka menu <i>workflow</i> .
Hasil yang diharapkan	Sistem menampilkan jumlah <i>workflow</i> yang sudah diselesaikan <i>workflow manager</i> pada bagian <i>implemented workflow</i> .
Hasil	Sistem menampilkan jumlah <i>workflow</i> yang sudah diselesaikan <i>workflow manager</i> pada bagian <i>implemented workflow</i> .
Status	Valid.

6.3.20 Pengujian Validasi Lihat Daftar *Template Workflow*

1. Kasus Uji Berhasil Lihat Daftar *Template Workflow*

Tabel 6.34 Kasus Uji Berhasil Lihat Daftar *Template Workflow*

Nama Kasus Uji	Lihat Daftar <i>Template Workflow</i>
Prosedur	1. Aktor menekan tombol <i>view</i> pada menu <i>template workflow</i> .
Hasil yang diharapkan	Sistem menampilkan halaman <i>template workflow</i> yang berisi daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 6.34 Kasus Uji Berhasil Lihat Daftar *Template Workflow* (lanjutan)

Hasil	Sistem menampilkan halaman <i>template workflow</i> yang berisi daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

6.3.21 Pengujian Validasi Lihat Cari *Template Workflow*

1. Kasus Uji Berhasil Cari *Template Workflow*

Tabel 6.35 Kasus Uji Berhasil Cari *Template Workflow*

Nama Kasus Uji	Cari <i>Template Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>template workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>template workflow</i> yang ada sesuai kata yang tulis aktor sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari *Template Workflow* Alternatif 1: Jika kata pencarian dicari tidak ada.

Tabel 6.36 Kasus Uji Berhasil Cari *Template Workflow* Alternatif 1

Nama Kasus Uji	Cari <i>Template Workflow</i> Alternatif 1: Jika kata pencarian dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom

Tabel 6.36 Kasus Uji Berhasil Cari *Template Workflow* Alternatif 1 (lanjutan)

Prosedur	<p>pengisian yang tidak terdapat pada <i>database</i> sistem.</p> <p>2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.</p>
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i> .
Hasil	Sistem tabel akan menampilkan <i>No items to display</i> .
Status	Valid.

6.3.22 Pengujian Validasi Lihat *Detail* Daftar *Template Workflow*

1. Kasus Uji Berhasil Lihat *Detail* Daftar *Template Workflow*

Tabel 6.37 Kasus Uji Berhasil Lihat *Detail* Daftar *Template Workflow*

Nama Kasus Uji	Lihat <i>Detail</i> Daftar <i>Template Workflow</i>
Prosedur	<p>1. Aktor menekan salah satu <i>daftar template workflow</i>.</p> <p>2. Aktor menekan tombol <i>view</i>.</p>
Hasil yang diharapkan	Sistem menampilkan <i>workflow</i> dari <i>template workflow</i> yang telah dipilih aktor.
Hasil	Sistem menampilkan <i>workflow</i> dari <i>template workflow</i> yang telah dipilih aktor.
Status	Valid.

6.3.23 Pengujian Validasi Lihat Daftar *Implemented Workflow*

1. Kasus Uji Berhasil Lihat Daftar *Implemented Workflow*

Tabel 6.38 Kasus Uji Berhasil Lihat Daftar *Implemented Workflow*

Nama Kasus Uji	Lihat Daftar <i>Implemented Workflow</i>
Prosedur	1. Aktor menekan tombol <i>view</i> pada menu <i>implemented workflow</i> .
Hasil yang diharapkan	Sistem menampilkan halaman <i>implemented workflow</i> yang berisi daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 6.38 Kasus Uji Berhasil Lihat Daftar *Implemented Workflow* (lanjutan)

Hasil	Sistem menampilkan halaman <i>implemented workflow</i> yang berisi daftar <i>template workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

6.3.24 Pengujian Validasi Lihat Cari *Implemented Workflow*

1. Kasus Uji Berhasil Cari *Implemented Workflow*

Tabel 6.39 Kasus Uji Berhasil Cari *Implemented Workflow*

Nama Kasus Uji	Cari <i>Implemented Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>implemented workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>implemented workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadline</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari *Implemented Workflow* Alternatif 1: Jika kata pencarian dicari tidak ada.

Tabel 6.40 Kasus Uji Berhasil Cari *Implemented Workflow* Alternatif 1

Nama Kasus Uji	Cari <i>Implemented Workflow</i> Alternatif 1: Jika kata pencarian tidak ada
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom



Tabel 6.40 Kasus Uji Berhasil Cari *Implemented Workflow* Alternatif 1 (lanjutan)

Prosedur	<p>pengambilan yang tidak terdapat pada <i>database</i> sistem.</p> <p>2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.</p>
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i>
Hasil	Sistem tabel akan menampilkan <i>No items to display</i>
Status	Valid.

6.3.25 Pengujian Validasi Lihat Lihat *Detail Daftar Implemented Workflow*

1. Kasus Uji Berhasil Lihat *Detail* Daftar *Implemented Workflow*

Tabel 6.41 Kasus Uji Berhasil Lihat *Detail* Daftar *Implemented Workflow*

Nama Kasus Uji	Lihat <i>Detail</i> Daftar <i>Implemented Workflow</i>
Prosedur	<p>1. Aktor menekan salah satu <i>daftar implemented workflow</i>.</p> <p>2. Aktor menekan tombol <i>view</i>.</p>
Hasil yang diharapkan	Sistem menampilkan <i>workflow</i> dari <i>implemented workflow</i> yang telah dipilih aktor.
Hasil	Sistem menampilkan <i>workflow</i> dari <i>implemented workflow</i> yang telah dipilih aktor.
Status	Valid.

6.3.26 Pengujian Validasi Lihat *Pie Chart* Persentase *Workflow*

1. Kasus Uji Berhasil Lihat *Pie Chart* Persentase *Workflow*

Tabel 6.42 Kasus Uji Berhasil Lihat *Pie Chart* Persentase *Workflow*

Nama Kasus Uji	Lihat <i>Pie Chart</i> Persentase <i>Workflow</i>
Prosedur	1. Aktor membuka menu <i>reports</i> .
Hasil yang diharapkan	Sistem menampilkan <i>pie chart</i> mengenai persentase penggunaan <i>workflow</i> baik <i>workflow</i> yang telah selesai maupun <i>workflow</i> yang sedang berjalan.
Hasil	Sistem menampilkan <i>pie chart</i> mengenai persentase penggunaan <i>workflow</i> baik <i>workflow</i> yang telah selesai maupun <i>workflow</i> yang sedang berjalan.
Status	Valid,



6.3.27 Pengujian Validasi Lihat Jumlah *Finished Workflow*

1. Kasus Uji Berhasil Lihat Jumlah *Finished Workflow*

Tabel 6.43 Kasus Uji Berhasil Lihat Jumlah *Finished Workflow*

Nama Kasus Uji	Lihat Jumlah <i>Finished Workflow</i>
Prosedur	1. Aktor membuka menu <i>reports</i> .
Hasil yang diharapkan	Sistem menampilkan jumlah <i>workflow</i> yang sudah diselesaikan <i>workflow manager</i> pada bagian <i>finished workflow</i> .
Hasil	Sistem menampilkan jumlah <i>workflow</i> yang sudah diselesaikan oleh <i>workflow manager</i> pada bagian <i>finished workflow</i> .
Status	Valid

6.3.28 Pengujian Validasi Lihat Daftar *Finished Workflow*

1. Kasus Uji Berhasil Lihat Daftar *Finished Workflow*

Tabel 6.44 Kasus Uji Berhasil Lihat Daftar *Finished Workflow*

Nama Kasus Uji	Lihat Daftar <i>Finished Workflow</i>
Prosedur	1. Aktor menekan tombol <i>view</i> pada menu <i>finished workflow</i> .
Hasil yang diharapkan	Sistem menampilkan daftar <i>finished workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , <i>workflow manager</i> , <i>PIC</i> , <i>deadline</i> , <i>start task</i> dan <i>finish task</i> dalam bentuk tabel.
Hasil	Sistem menampilkan daftar <i>finished workflow</i> yang telah dibuat admin sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , <i>workflow manager</i> , <i>PIC</i> , <i>deadline</i> , <i>start task</i> dan <i>finish task</i> dalam bentuk tabel.
Status	Valid.

6.3.29 Pengujian Validasi Lihat Cari *Finished Workflow*

1. Kasus Uji Berhasil Cari *Finished Workflow*

Tabel 6.45 Kasus Uji Berhasil Cari *Finished Workflow*

Nama Kasus Uji	Cari <i>Finished Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>finished workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name, task name, workflow manager, PIC, deadline, start task</i> dan <i>finish task</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>finished workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya berdasarkan <i>workflow name, task name, workflow manager, PIC, deadline, start task</i> dan <i>finish task</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Berhasil Cari *Finished Workflow* Alternatif 1: Jika kata pencarian yang dicari tidak ada.

Tabel 6.46 Kasus Uji Berhasil Cari *Finished Workflow* Alternatif 1

Nama Kasus Uji	Cari <i>Finished Workflow</i> Alternatif 1: Jika kata pencarian yang dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i> .
Hasil	Sistem tabel akan menampilkan <i>No items to display</i> .
Status	Valid.



6.3.30 Pengujian Validasi Lihat *Detail Finished Workflow*

1. Kasus Uji Berhasil Lihat *Detail Finished Workflow*

Tabel 6.47 Kasus Uji Berhasil Lihat *Detail Finished Workflow*

Nama Kasus Uji	Lihat <i>Detail Finished Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar finished workflow</i>. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan <i>workflow</i> dari <i>finished workflow</i> yang telah dipilih aktor.
Hasil	Sistem menampilkan <i>workflow</i> dari <i>finished workflow</i> yang telah dipilih aktor.
Status	Valid.

6.3.31 Pengujian Validasi Lihat Statistik *Finished Workflow*

1. Kasus Uji Berhasil Lihat Statistik *Finished Workflow*

Tabel 6.48 Kasus Uji Berhasil Lihat Statistik *Finished Workflow*

Nama Kasus Uji	Lihat Statistik <i>Finished Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar finished workflow</i>. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan statistik <i>workflow</i> yang telah dipilih aktor sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
Hasil	Sistem menampilkan statistik <i>workflow</i> yang telah dipilih aktor sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
Status	Valid.

6.3.32 Pengujian Validasi Lihat Jumlah *Workflow By Template*

1. Kasus Uji Berhasil Lihat Jumlah *Workflow By Template*

Tabel 6.49 Kasus Uji Berhasil Lihat Jumlah *Workflow By Template*

Nama Kasus Uji	Lihat Jumlah <i>Workflow By Template</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>reports</i>.

Tabel 6.49 Kasus Uji Berhasil Lihat Jumlah *Workflow By Template* (lanjutan)

Hasil yang diharapkan	Sistem menampilkan jumlah <i>workflow</i> berdasarkan jenis <i>template workflow</i> yang ada.
Hasil	Sistem menampilkan jumlah <i>workflow</i> berdasarkan jenis <i>template workflow</i> yang ada.
Status	Valid.

6.3.33 Pengujian Validasi Lihat Daftar *Workflow By Template*

1. Kasus Uji Berhasil Lihat Daftar *Workflow By Template*

Tabel 6.50 Kasus Uji Berhasil Lihat Daftar *Workflow By Template*

Nama Kasus Uji	Lihat Daftar <i>Workflow By Template</i>
Prosedur	1. Aktor menekan tombol <i>view</i> pada menu <i>workflow by template</i> .
Hasil yang diharapkan	Sistem menampilkan daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> .
Hasil	Sistem menampilkan daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> .
Status	Valid.

6.3.34 Pengujian Validasi Lihat Cari *Workflow By Template*

1. Kasus Uji Berhasil Cari *Workflow By Template*

Tabel 6.51 Kasus Uji Berhasil Cari *Workflow By Template*

Nama Kasus Uji	Cari <i>Workflow By Template</i>
Prosedur	1. Aktor menulis kata yang ingin dicari berdasarkan <i>template workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i> .
Hasil yang diharapkan	Sistem menampilkan daftar <i>template by workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.

Tabel 6.51 Kasus Uji Berhasil Cari *Workflow By Template* (lanjutan)

Hasil	Sistem menampilkan daftar <i>template by workflow</i> yang ada sesuai kata yang ditulis aktor sebelumnya daftar <i>workflow</i> dari jenis <i>template workflow</i> berdasarkan <i>template workflow name</i> dan <i>status</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari *Workflow By Template* Alternatif 1: Jika kata pencarian yang dicari tidak ada.

Tabel 6.52 Kasus Uji Berhasil Cari *Workflow By Template* Alternatif 1

Nama Kasus Uji	Cari <i>Workflow By Template</i> Alternatif 1: Jika kata pencarian yang dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>template workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i> .
Hasil	Sistem tabel akan menampilkan <i>No items to display</i> .
Status	Valid.

6.3.35 Pengujian Validasi Lihat Statistik *Workflow By Template*

1. Kasus Uji Berhasil Lihat Statistik *Workflow By Template*

Tabel 6.53 Kasus Uji Berhasil Lihat Statistik *Workflow By Template*

Nama Kasus Uji	Lihat Statistik <i>Workflow By Template</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan salah satu <i>daftar finished workflow</i>. 2. Aktor menekan tombol <i>statistics</i>.
Hasil yang diharapkan	Sistem menampilkan melihat statistik <i>workflow by template</i> yang telah dipilih aktor dalam bentuk <i>bar chart</i> rata-rata <i>template workflow</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .

Tabel 6.53 Kasus Uji Berhasil Lihat Statistik *Workflow By Template* (lanjutan)

Hasil	Sistem menampilkan melihat statistik <i>workflow by template</i> yang telah dipilih aktor dalam bentuk <i>bar chart</i> rata-rata <i>template workflow</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
Status	Valid.

6.3.36 Pengujian Validasi Lihat Jumlah *Pending User*

1. Kasus Uji Berhasil Lihat Jumlah *Pending User*

Tabel 6.54 Kasus Uji Berhasil Lihat Jumlah *Pending User*

Nama Kasus Uji	Lihat Jumlah <i>Pending User</i>
Prosedur	1. Aktor membuka menu <i>users</i> .
Hasil yang diharapkan	Sistem menampilkan jumlah <i>pending user</i> yang ada pada sistem.
Hasil	Sistem menampilkan jumlah <i>pending user</i> yang ada pada sistem.
Status	Valid.

6.3.37 Pengujian Validasi Lihat Jumlah *All User*

1. Kasus Uji Berhasil Lihat Jumlah *All User*

Tabel 6.55 Kasus Uji Berhasil Lihat Jumlah *All User*

Nama Kasus Uji	Lihat Jumlah <i>All User</i>
Prosedur	1. Aktor membuka menu <i>users</i> .
Hasil yang diharapkan	Sistem menampilkan jumlah seluruh <i>user</i> yang ada pada sistem pada bagian <i>all user</i> .
Hasil	Sistem menampilkan jumlah seluruh <i>user</i> yang ada pada sistem pada bagian <i>all user</i> .
Status	Valid.

6.3.38 Pengujian Validasi Lihat Daftar *Pending Users*

1. Kasus Uji Berhasil Lihat Daftar *Pending Users*

Tabel 6.56 Kasus Uji Berhasil Lihat Daftar *Pending Users*

Nama Kasus Uji	Lihat Daftar <i>Pending Users</i>
-----------------------	-----------------------------------

Tabel 6.56 Kasus Uji Berhasil Lihat Daftar *Pending Users* (lanjutan)

Prosedur	1. Aktor menekan tombol <i>manage</i> pada menu <i>pending users</i> .
Hasil yang diharapkan	Sistem menampilkan daftar <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel.
Hasil	Sistem menampilkan daftar <i>pending users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel.
Status	Valid.

6.3.39 Pengujian Validasi Lihat Cari *Pending Users*

1. Kasus Uji Berhasil Cari *Pending Users*

Tabel 6.57 Kasus Uji Berhasil Cari *Pending Users*

Nama Kasus Uji	Cari <i>Pending Users</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>full name, email, role, atau department workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, dan department</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari *Pending Users* Alternatif 1: Jika kata pencarian dicari tidak ada.

Tabel 6.58 Kasus Uji Berhasil Cari *Pending Users* Alternatif 1

Nama Kasus Uji	Cari <i>Pending Users</i> Alternatif 1: Jika kata pencarian dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan berdasarkan <i>full name, email, role</i>, atau <i>department workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i>
Hasil	Sistem tabel akan menampilkan <i>No items to display</i>
Status	Valid.

6.3.40 Pengujian Validasi Verifikasi *Pending Users*

1. Kasus Uji Berhasil Verifikasi *Pending Users*

Tabel 6.59 Kasus Uji Berhasil Verifikasi *Pending Users*

Nama Kasus Uji	Verifikasi <i>Pending Users</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan salah satu baris <i>user</i>. 2. Aktor menekan tombol <i>verify</i>. 3. Sistem menampilkan pesan <i>Are you sure that you want to verify this user?</i> 4. Aktor menekan tombol OK.
Hasil yang diharapkan	Sistem menampilkan pesan <i>user verified</i> dan mengizinkan <i>user</i> untuk masuk ke dalam sistem.
Hasil	Sistem menampilkan pesan <i>user verified</i> dan mengizinkan <i>user</i> untuk masuk ke dalam sistem.
Status	Valid.

2. Kasus Uji Verifikasi *Pending Users* Alternatif 1: Jika aktor menekan tombol *cancel*.

Tabel 6.60 Kasus Uji Berhasil Cari *Pending Users* Alternatif 1

Nama Kasus Uji	Verifikasi <i>Pending Users</i> Alternatif 1: Jika aktor menekan tombol <i>cancel</i> .
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan salah satu baris <i>user</i>. 2. Aktor menekan tombol <i>verify</i>. 3. Sistem menampilkan pesan <i>Are you sure that you want to verify this user?</i> 4. Aktor menekan tombol <i>cancel</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>You choose to cancel</i> .
Hasil	Sistem menampilkan pesan <i>You choose to cancel</i> .
Status	Valid.

6.3.41 Pengujian Validasi Tolak *Pending Users*

1. Kasus Uji Berhasil Tolak *Pending Users*

Tabel 6.61 Kasus Uji Berhasil Tolak *Pending Users*

Nama Kasus Uji	Tolak <i>Pending Users</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan salah satu baris <i>user</i>. 2. Aktor menekan tombol <i>decline</i>. 3. Sistem menampilkan pesan <i>Are you sure that you want to decline this user?</i> 4. Aktor menekan tombol OK.
Hasil yang diharapkan	Sistem menampilkan pesan <i>user rejected</i> .
Hasil	Sistem menampilkan pesan <i>user rejected</i> .
Status	Valid.

2. Kasus Uji Verifikasi Tolak *Users* Alternatif 1: Jika aktor menekan tombol *cancel*.

Tabel 6.62 Kasus Uji Tolak *Pending Users* Alternatif 1

Nama Kasus Uji	Verifikasi Tolak <i>Users</i> Alternatif 1: Jika aktor menekan tombol <i>cancel</i> .
-----------------------	---



Tabel 6.62 Kasus Uji Tolak *Pending Users* Alternatif 1 (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan salah satu baris <i>user</i>. 2. Aktor menekan tombol <i>decline</i>. 3. Sistem menampilkan pesan <i>Are you sure that you want to decline this user?</i> 4. Aktor menekan tombol <i>cancel</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>You choose to cancel</i> .
Hasil	Sistem menampilkan pesan <i>You choose to cancel</i> .
Status	Valid.

6.3.42 Pengujian Validasi Lihat Daftar *All Users*

1. Kasus Uji Berhasil Lihat Daftar *All Users*

Tabel 6.63 Kasus Uji Berhasil Lihat Daftar *All Users*

Nama Kasus Uji	Lihat Daftar <i>All Users</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>view</i> pada menu <i>all users</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>all users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel.
Hasil	Sistem menampilkan daftar <i>all users</i> berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel.
Status	Valid.

6.3.43 Pengujian Validasi Lihat Cari *All Users*

1. Kasus Uji Berhasil Cari *All Users*

Tabel 6.64 Kasus Uji Berhasil Cari *All Users*

Nama Kasus Uji	Cari <i>All Users</i>
-----------------------	-----------------------



Tabel 6.64 Kasus Uji Berhasil Cari *All Users* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> atau <i>status workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> dan <i>status</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari *All Users* Alternatif 1 : Jika kata pencarian dicari tidak ada.

Tabel 6.65 Kasus Uji Cari *All Users* Alternatif 1

Nama Kasus Uji	Cari <i>All Users</i> Alternatif 1 : Jika kata pencarian dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan jenis <i>template workflow</i> berdasarkan <i>full name, email, role, department</i> atau <i>status workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i>
Hasil	Sistem tabel akan menampilkan <i>No items to display</i>
Status	Valid.

6.3.44 Pengujian Validasi Berhasil Lihat *Profile Admin*

1. Kasus Uji Berhasil Lihat *Profile Admin*

Tabel 6.66 Kasus Uji Berhasil Lihat *Profile Admin*

Nama Kasus Uji	Lihat <i>Profile Admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown admin</i>. 2. Aktor menekan menu <i>profile</i>
Hasil yang diharapkan	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
Hasil	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
Status	Valid.

6.3.45 Pengujian Validasi Berhasil Ubah *Password Admin*

1. Kasus Uji Berhasil Ubah *Password Admin*

Tabel 6.67 Kasus Uji Berhasil Ubah *Password Admin*

Nama Kasus Uji	Ubah <i>Password Admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown admin</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i>. 5. Aktor menekan tombol <i>change password</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem merubah <i>password</i> dari aktor.
Hasil	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem merubah <i>password</i> dari aktor.
Status	Valid

2. Kasus Uji Ubah *Password Admin* Alternatif 1: Jika kolom *password* baru dan konfirmasi *password* baru kosong atau tidak sama

Tabel 6.68 Kasus Uji Berhasil Ubah *Password Admin*

Nama Kasus Uji	Ubah <i>Password Admin</i> Alternatif 1: Jika kolom <i>password</i> baru dan konfirmasi <i>password</i> baru kosong atau tidak sama
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown admin</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i> tidak sama dengan kolom <i>new password</i> sebelumnya. 5. Aktor menekan tombol <i>change password</i>.
Hasil yang diharapkan	Sistem menampilkan menampilkan pesan <i>password empty or didn't match</i> .
Hasil	Sistem menampilkan menampilkan pesan <i>password empty or didn't match..</i>
Status	Valid.

6.3.46 Pengujian Validasi Berhasil Lihat *Template Workflow Admin*

1. Kasus Uji Berhasil Lihat *Template Workflow Admin*

Tabel 6.69 Kasus Uji Berhasil Lihat *Template Workflow Admin*

Nama Kasus Uji	Lihat <i>Template Workflow Admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>workflow</i>.
Hasil yang diharapkan	Sistem menampilkan tabel <i>template workflow</i> berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam
Hasil	Sistem menampilkan tabel <i>template workflow</i> berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam
Status	Valid.

6.3.47 Pengujian Validasi Lihat Cari *Template Workflow Admin*

1. Kasus Uji Berhasil Cari *Template Workflow Admin*

Tabel 6.70 Kasus Uji Berhasil Cari *Template Workflow Admin*

Nama Kasus Uji	Cari <i>Template Workflow Admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan <i>workflow name</i> , <i>task name</i> , dan <i>deadlines</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari *Template Workflow Admin* Alternatif 1 : Jika kata pencarian dicari tidak ada.

Tabel 6.71 Kasus Uji Cari *All Users* Alternatif 1

Nama Kasus Uji	Cari <i>Template Workflow Admin</i> Alternatif 1 : Jika kata pencarian dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i>
Hasil	Sistem tabel akan menampilkan <i>No items to display</i>
Status	Valid.

6.3.48 Pengujian Validasi Berhasil Pilih *Template Workflow Admin*

1. Kasus Uji Berhasil Pilih *Template Workflow Admin*

Tabel 6.72 Kasus Uji Berhasil Pilih *Template Workflow Admin*

Nama Kasus Uji	Pilih <i>Template Workflow Admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>template workflow</i> yang akan diinginkan. 2. Aktor menekan tombol <i>use</i> pada baris tersebut.
Hasil yang diharapkan	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan <i>workflow name, task name, dan deadlines</i> dalam bentuk tabel.
Hasil	Sistem menampilkan daftar <i>pending users</i> dari kata yang ditulis sebelumnya berdasarkan <i>workflow name, task name, dan deadlines</i> dalam bentuk tabel.
Status	Valid.

6.3.49 Pengujian Validasi Berhasil Mulai *Template Workflow Admin*

1. Kasus Uji Berhasil Mulai *Template Workflow Admin*

Tabel 6.73 Kasus Uji Berhasil Mulai *Template Workflow Admin*

Nama Kasus Uji	Mulai <i>Template Workflow Admin</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>subject</i> pada <i>field</i> objek yang disediakan. 2. Aktor menekan tombol <i>start workflow</i>. 3. Sistem menampilkan pesan <i>are you sure that you want to start the workflow?</i> 4. Aktor menekan tombol OK.
Hasil yang diharapkan	Sistem menampilkan pesan <i>Workflow Started</i> yang sekaligus memulai <i>workflow</i> tersebut dan mengirimkan notifikasi ke <i>person in charge</i> yang terlibat.
Hasil	Sistem menampilkan pesan <i>Workflow Started</i> yang sekaligus memulai <i>workflow</i> tersebut dan mengirimkan notifikasi ke <i>person in charge</i> yang terlibat.
Status	Valid.

2. Kasus Uji Mulai *Template Workflow Admin* Alternatif 1: Jika aktor menekan tombol *cancel*.

Tabel 6.74 Kasus Uji Mulai *Template Workflow Admin* Alternatif 1

Nama Kasus Uji	Verifikasi Mulai <i>Template Workflow Admin</i> Alternatif 1: Jika aktor menekan tombol <i>cancel</i> .
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisi <i>subject</i> pada <i>field</i> objek yang disediakan. 2. Aktor menekan tombol <i>start workflow</i>. 3. Sistem menampilkan pesan <i>are you sure that you want to start the workflow?</i> 4. Aktor menekan tombol <i>cancel</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>You choose to cancel</i> .
Hasil	Sistem menampilkan pesan <i>You choose to cancel</i> .
Status	Valid.

6.3.50 Pengujian Validasi Berhasil Pilih *Person In Charge*

1. Kasus Uji Berhasil Pilih *Person In Charge*

Tabel 6.75 Kasus Uji Berhasil Pilih *Person In Charge*

Nama Kasus Uji	Pilih <i>Person In Charge</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan bagian kata PIC pada objek yang disediakan. 2. Sistem akan menampilkan daftar <i>person in charge</i> berdasarkan <i>full name</i>, dan <i>email</i> dalam bentuk tabel. 3. Aktor memilih dengan menekan baris <i>person in charge</i> yang akan dipilih. 4. Aktor menekan tombol <i>choose</i>.
Hasil yang diharapkan	Sistem mengubah kata PIC sebelumnya menjadi <i>person in charge</i> yang telah ditentukan aktor sebelumnya.



Tabel 6.75 Kasus Uji Berhasil Pilih *Person In Charge* (lanjutan)

Hasil	Sistem mengubah kata PIC sebelumnya menjadi <i>person in charge</i> yang telah ditentukan aktor sebelumnya.
Status	Valid.

6.3.51 Pengujian Validasi Berhasil Unggah *File Workflow*

1. Kasus Uji Berhasil Unggah *File Workflow*

Tabel 6.76 Kasus Uji Berhasil Unggah *File Workflow*

Nama Kasus Uji	Unggah <i>File Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan klik kanan pada objek <i>data storage</i> 2. Aktor menekan tombol <i>upload a file</i>. 3. Sistem menampilkan tampilan untuk melakukan unggah <i>file</i>. 4. Aktor menekan tombol <i>select files</i>. 5. Sistem akan menampilkan direktori <i>folder</i> aktor. 6. Aktor memilih <i>file</i> yang akan diunggah. 7. Sistem menampilkan nama <i>file</i>, tipe <i>file</i>, dan ukuran <i>file</i>. 8. Aktor menekan tombol <i>submit</i>.
Hasil yang diharapkan	Sistem akan menyimpan <i>file</i> yang sudah dipilih oleh aktor sebelumnya yang akan terunggah ketika <i>workflow</i> dijalankan.
Hasil	Sistem akan menyimpan <i>file</i> yang sudah dipilih oleh aktor sebelumnya yang akan terunggah ketika <i>workflow</i> dijalankan.
Status	Valid.

6.3.52 Pengujian Validasi Berhasil Lihat Daftar *My Workflow*

1. Kasus Uji Berhasil Lihat Daftar *My Workflow*

Tabel 6.77 Kasus Uji Berhasil Lihat Daftar *My Workflow*

Nama Kasus Uji	Lihat Daftar <i>My Workflow</i>
-----------------------	---------------------------------

Tabel 6.77 Kasus Uji Berhasil Lihat Daftar *My Workflow* (lanjutan)

Prosedur	1. Aktor membuka menu <i>my workflow</i> .
Hasil yang diharapkan	Sistem menampilkan daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel.
Hasil	Sistem menampilkan daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel.
Status	Valid.

6.3.53 Pengujian Validasi Cari *My Workflow*

1. Kasus Uji Berhasil Cari *My Workflow*

Tabel 6.78 Kasus Uji Berhasil Cari *My Workflow*

Nama Kasus Uji	Cari <i>My Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>workflow</i> yang telah diimplementasikan oleh aktor sebelumnya berdasarkan <i>workflow name, task name, person in charge, deadline, status, start date, dan finish date</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.



2. Kasus Uji Cari *Template Workflow Admin* Alternatif 1 : Jika kata pencarian yang dicari tidak ada.

Tabel 6.79 Kasus Uji Cari *All Users* Alternatif 1

Nama Kasus Uji	Cari <i>Template Workflow Admin</i> Alternatif 1 : Jika kata pencarian yang dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i>
Hasil	Sistem tabel akan menampilkan <i>No items to display</i>
Status	Valid.

6.3.54 Pengujian Validasi Berhasil Lihat *Detail My Workflow*

1. Kasus Uji Berhasil Lihat *Detail My Workflow*

Tabel 6.80 Kasus Uji Berhasil Lihat *Detail My Workflow*

Nama Kasus Uji	Lihat <i>Detail My Workflow</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan <i>detail</i> daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri dan sistem menampilkannya dalam bentuk <i>workflow</i> .
Hasil	Sistem menampilkan <i>detail</i> daftar <i>workflow</i> yang sudah diimplementasikan oleh <i>workflow manager</i> itu sendiri dan sistem menampilkannya dalam bentuk <i>workflow</i> .
Status	Valid.

6.3.55 Pengujian Validasi Berhasil Lihat Statistik *My Workflow*

1. Kasus Uji Berhasil Lihat Statistik *My Workflow*

Tabel 6.81 Kasus Uji Berhasil Lihat Statistik *My Workflow*

Nama Kasus Uji	Lihat Statistik <i>My Workflow</i>
-----------------------	------------------------------------

Tabel 6.81 Kasus Uji Berhasil Lihat Statistik *My Workflow* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>statistics</i>.
Hasil yang diharapkan	Sistem menampilkan statistik <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
Hasil	Sistem menampilkan statistik <i>workflow</i> yang telah selesai sebelumnya dalam bentuk <i>bar chart</i> berdasarkan <i>days assign</i> dan <i>days completed</i> .
Status	Valid.

6.3.56 Pengujian Validasi Berhasil Lihat Daftar *Task*

1. Kasus Uji Berhasil Lihat Daftar *Task*

Tabel 6.82 Kasus Uji Berhasil Lihat Daftar *Task*

Nama Kasus Uji	Lihat Daftar <i>Task</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>statistics</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel.
Hasil	Sistem menampilkan daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel.
Status	Valid.

6.3.57 Pengujian Validasi Cari Daftar *Task*

1. Kasus Uji Berhasil Cari Daftar *Task*

Tabel 6.83 Kasus Uji Berhasil Cari Daftar *Task*

Nama Kasus Uji	Cari Daftar <i>Task</i>
-----------------------	-------------------------



Tabel 6.83 Kasus Uji Berhasil Cari Daftar *Task* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan daftar <i>task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> dalam bentuk tabel kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari Daftar *Task* Alternatif 1 : Jika kata pencarian yang dicari tidak ada.

Tabel 6.84 Kasus Uji Cari Daftar *Task* Alternatif 1

Nama Kasus Uji	Cari Daftar <i>Task</i> Alternatif 1 : Jika kata pencarian yang dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i> .
Hasil	Sistem tabel akan menampilkan <i>No items to display</i> .
Status	Valid.

6.3.58 Pengujian Validasi Berhasil Lihat Isi *Form Task*

1. Kasus Uji Berhasil Lihat Isi *Form Task*

Tabel 6.85 Kasus Uji Berhasil Lihat Isi *Form Task*

Nama Kasus Uji	Lihat Isi <i>Form Task</i>
-----------------------	----------------------------

Tabel 6.85 Kasus Uji Berhasil Lihat Isi *Form Task* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih.. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan isi <i>form task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan yang berisi <i>subject</i> , <i>description</i> , dan daftar <i>file</i> .
Hasil	Sistem menampilkan isi <i>form task</i> yang ada pada <i>workflow</i> yang telah diimplementasikan yang berisi <i>subject</i> , <i>description</i> , dan daftar <i>file</i> .
Status	Valid.

2. Kasus Uji Lihat Isi Form *Task* Alternatif 1: Jika tidak ada *file* yang tersedia

Tabel 6.86 Kasus Uji Berhasil Lihat Isi *Form Task* Alternatif 1

Nama Kasus Uji	Lihat Isi Form <i>Task</i> Alternatif 1: Jika tidak ada <i>file</i> yang tersedia
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan isi daftar <i>file</i> dengan pesan <i>No Files Available</i> .
Hasil	Sistem menampilkan isi daftar <i>file</i> dengan pesan <i>No Files Available</i> .
Status	Valid.

6.3.59 Pengujian Validasi Berhasil Unduh *File* individu *Person In Charge*

1. Kasus Uji Berhasil Unduh *File* individu *Person In Charge*

Tabel 6.87 Kasus Uji Berhasil Unduh *File* individu *Person In Charge*

Nama Kasus Uji	Unduh <i>File</i> individu <i>Person In Charge</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor memilih dengan menekan <i>file</i> yang akan di unduh.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.

Tabel 6.87 Kasus Uji Berhasil Unduh *File* individu *Person In Charge* (lanjutan)

Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Status	Valid.

6.3.60 Pengujian Validasi Berhasil Unduh Seluruh *File Person In Charge*

1. Kasus Uji Berhasil Unduh Seluruh *File Person In Charge*

Tabel 6.88 Kasus Uji Berhasil Unduh Seluruh *File Person In Charge*

Nama Kasus Uji	Unduh Seluruh <i>File Person In Charge</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor menekan tombol <i>download</i>.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i> pada <i>form task</i> yang di unggah <i>person in charge</i> .
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>file</i> pada <i>form task</i> yang di unggah <i>person in charge</i> .
Status	Valid.

6.3.61 Pengujian Validasi Berhasil Kirim Notifikasi

1. Kasus Uji Berhasil Kirim Notifikasi

Tabel 6.89 Kasus Uji Berhasil Kirim Notifikasi

Nama Kasus Uji	Kirim Notifikasi
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>notification</i> 3. Sistem menampilkan tampilan untuk menuliskan isi notifikasi pada sebuah <i>textfield</i>. 4. Aktor menuliskan pesan yang akan dikirimkan. 5. Aktor menekan tombol <i>send</i>.

Tabel 6.89 Kasus Uji Berhasil Kirim Notifikasi (lanjutan)

Hasil yang diharapkan	Sistem mengirimkan notifikasi ke <i>person in charge</i> yang telah dipilih kemudian mengembalikan tampilan ke halaman <i>statistics</i> .
Hasil	Sistem mengirimkan notifikasi ke <i>person in charge</i> yang telah dipilih kemudian mengembalikan tampilan ke halaman <i>statistics</i> .
Status	Valid.

6.3.62 Pengujian Validasi Berhasil Lihat Daftar *File*

1. Kasus Uji Berhasil Lihat Daftar *File*

Tabel 6.90 Kasus Uji Berhasil Lihat Daftar *File*

Nama Kasus Uji	Lihat Daftar <i>File</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris <i>workflow</i> yang akan dipilih. 2. Aktor menekan tombol <i>statistics</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>file</i> yang diunggah oleh aktor itu sendiri pada <i>workflow</i> yang dipilih.
Hasil	Sistem menampilkan daftar <i>file</i> yang diunggah oleh aktor itu sendiri pada <i>workflow</i> yang dipilih.
Status	Valid.

6.3.63 Pengujian Validasi Berhasil Unduh *File* Individu

1. Kasus Uji Berhasil Unduh *File* Individu

Tabel 6.91 Kasus Uji Berhasil Unduh *File* Individu

Nama Kasus Uji	Unduh <i>File</i> Individu
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor memilih dengan menekan <i>file</i> yang akan di unduh.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Status	Valid.

6.3.64 Pengujian Validasi Berhasil Unduh Seluruh *File*

1. Kasus Uji Berhasil Unduh Seluruh *File*

Tabel 6.92 Kasus Uji Berhasil Unduh Seluruh *File*

Nama Kasus Uji	Unduh Seluruh <i>File</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> yang tersedia. 2. Aktor menekan tombol <i>download all</i>.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Status	Valid.

6.3.65 Pengujian Validasi Berhasil Terima Notifikasi *Workflow* Selesai

1. Kasus Uji Berhasil Terima Notifikasi *Workflow* Selesai

Tabel 6.93 Kasus Uji Berhasil Terima Notifikasi *Workflow* Selesai

Nama Kasus Uji	Terima Notifikasi <i>Workflow</i> Selesai
Prosedur	<ol style="list-style-type: none"> 1. <i>Person in charge</i> pada tugas terakhir dalam <i>workflow</i> yang dimiliki aktor telah menyelesaikan tugasnya. 2. Sistem mengirimkan pesan ke aktor nama <i>workflow has finished</i>.
Hasil yang diharapkan	Sistem akan mengirimkan notifikasi jika suatu <i>workflow</i> sudah diselesaikan seluruh prosesnya oleh seluruh <i>person in charge</i> yang terlibat.
Hasil	Sistem akan mengirimkan notifikasi jika suatu <i>workflow</i> sudah diselesaikan seluruh prosesnya oleh seluruh <i>person in charge</i> yang terlibat.
Status	Valid.

6.3.66 Pengujian Validasi Berhasil Lihat Daftar Notifikasi *Workflow Manager*

1. Kasus Uji Berhasil Lihat Daftar Notifikasi *Workflow Manager*

Tabel 6.94 Kasus Uji Berhasil Lihat Daftar Notifikasi *Workflow Manager*

Nama Kasus Uji	Lihat Daftar Notifikasi <i>Workflow Manager</i>
Prosedur	1. Aktor membuka menu <i>notifications</i> .
Hasil yang diharapkan	Sistem akan menampilkan daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi diterima.
Hasil	Sistem akan menampilkan daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi diterima.
Status	Valid.

6.3.67 Pengujian Validasi Berhasil Lihat *Profile Workflow Manager*

1. Kasus Uji Berhasil Lihat *Profile Workflow Manager*

Tabel 6.95 Kasus Uji Berhasil Lihat *Profile Workflow Manager*

Nama Kasus Uji	Lihat <i>Profile Workflow Manager</i>
Prosedur	1. Aktor menekan <i>dropdown Workflow Manager</i> . 2. Aktor menekan menu <i>profile</i> .
Hasil yang diharapkan	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
Hasil	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
Status	Valid.

6.3.68 Pengujian Validasi Berhasil Ubah *Password Workflow Manager*

1. Kasus Uji Berhasil Ubah *Password Workflow Manager*

Tabel 6.96 Kasus Uji Berhasil Ubah *Password Workflow Manager*

Nama Kasus Uji	Ubah <i>Password Admin</i>
Prosedur	1. Aktor menekan <i>dropdown Workflow Manager</i> . 2. Aktor menekan menu <i>profile</i> .

Tabel 6.96 Kasus Uji Berhasil Ubah *Password Workflow Manager* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i>. 5. Aktor menekan tombol <i>change password</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.
Hasil	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.
Status	Valid.

2. Kasus Uji Ubah *Password Workflow Manager* Alternatif 1: Jika kolom *password* baru dan konfirmasi *password* baru kosong atau tidak sama

Tabel 6.97 Kasus Uji Berhasil Ubah *Password Workflow Manager*

Nama Kasus Uji	Ubah <i>Password Workflow Manager</i> Alternatif 1: Jika kolom <i>password</i> baru dan konfirmasi <i>password</i> baru kosong atau tidak sama
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown Workflow Manager</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i> tidak sama dengan kolom <i>new password</i> sebelumnya. 5. Aktor menekan tombol <i>change password</i>.
Hasil yang diharapkan	Sistem menampilkan menampilkan pesan <i>password empty or didn't match</i> .
Hasil	Sistem menampilkan menampilkan pesan <i>password empty or didn't match</i> .
Status	Valid.



6.3.69 Pengujian Validasi Berhasil Lihat Daftar Tugas

1. Kasus Uji Berhasil Lihat Daftar Tugas

Tabel 6.98 Kasus Uji Berhasil Lihat Daftar Tugas

Nama Kasus Uji	Lihat Daftar Tugas
Prosedur	1. Aktor membuka menu <i>task</i> .
Hasil yang diharapkan	Sistem menampilkan daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name, task name, manager, deadline, task start, dan must complete</i> .
Hasil	Sistem menampilkan daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name, task name, manager, deadline, task start, dan must complete</i> .
Status	Valid.

6.3.70 Pengujian Validasi Cari Tugas

1. Kasus Uji Berhasil Cari Tugas

Tabel 6.99 Kasus Uji Berhasil Cari Tugas

Nama Kasus Uji	Cari Tugas
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan hasil pencarian daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name, task name, manager, deadline, task start, dan must complete</i> kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Hasil	Sistem menampilkan hasil pencarian daftar tugas yang diberikan <i>workflow manager</i> berdasarkan <i>workflow name, task name, manager, deadline, task start, dan must complete</i> kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari Tugas Alternatif 1 : Jika kata pencarian yang dicari tidak ada.

Tabel 6.100 Kasus Cari Tugas Alternatif 1

Nama Kasus Uji	Cari Tugas Alternatif 1 : Jika kata pencarian yang dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i> .
Hasil	Sistem tabel akan menampilkan <i>No items to display</i> .
Status	Valid.

6.3.71 Pengujian Validasi Berhasil Lihat *Shared Files*

1. Kasus Uji Berhasil Lihat *Shared Files*

Tabel 6.101 Kasus Uji Berhasil Lihat *Shared Files*

Nama Kasus Uji	Lihat <i>Shared Files</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan daftar <i>file</i> yang diunggah oleh <i>workflow manager</i> pada <i>shared files</i> .
Hasil	Sistem menampilkan daftar <i>file</i> yang diunggah oleh <i>workflow manager</i> pada <i>shared files</i> .
Status	Valid

6.3.72 Pengujian Validasi Berhasil Unduh *File Individu Shared Files*

1. Kasus Uji Berhasil Unduh *File Individu Shared Files*

Tabel 6.102 Kasus Uji Berhasil Unduh *File Individu Shared Files*

Nama Kasus Uji	Unduh <i>File Individu Shared Files</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>shared files</i>.

	2. Aktor memilih dengan menekan <i>file</i> yang akan diunduh.
--	--

Tabel 6.102 Kasus Uji Berhasil Unduh *File Individu Shared Files* (lanjutan)

Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Status	Valid.

6.3.73 Pengujian Validasi Berhasil Unduh Seluruh *File Shared Files*

1. Kasus Uji Berhasil Unduh Seluruh *File Shared Files*

Tabel 6.103 Kasus Uji Berhasil Unduh Seluruh *File Shared Files*

Nama Kasus Uji	Unduh Seluruh <i>File Shared Files</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>shared files</i>. 2. Aktor menekan tombol <i>download all</i>.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Status	Valid.

6.3.74 Pengujian Validasi Berhasil Lihat Jumlah Tugas Sebelum

1. Kasus Uji Berhasil Lihat Jumlah Tugas Sebelum

Tabel 6.104 Kasus Uji Berhasil Lihat Jumlah Tugas Sebelum

Nama Kasus Uji	Lihat Jumlah Tugas Sebelum
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan jumlah tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri pada bagian <i>previous task</i> .

Tabel 6.104 Kasus Uji Berhasil Lihat Jumlah Tugas Sebelum (lanjutan)

Hasil	Sistem menampilkan jumlah tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri pada bagian <i>previous task</i> .
Status	Valid.

6.3.75 Pengujian Validasi Berhasil Lihat Daftar Tugas Sebelum

1. Kasus Uji Berhasil Lihat Daftar Tugas Sebelum

Tabel 6.105 Kasus Uji Berhasil Lihat Daftar Tugas Sebelum

Nama Kasus Uji	Lihat Daftar Tugas Sebelum
Prosedur	Aktor menekan tombol <i>view previous task</i> .
Hasil yang diharapkan	Sistem menampilkan daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
Hasil	Sistem menampilkan daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> .
Status	Valid.

6.3.76 Pengujian Validasi Cari Tugas Sebelum

1. Kasus Uji Berhasil Cari Tugas Sebelum

Tabel 6.106 Kasus Uji Berhasil Cari Tugas Sebelum

Nama Kasus Uji	Cari Tugas Sebelum
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem menampilkan hasil pencarian daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.



Tabel 6.106 Kasus Uji Berhasil Cari Tugas Sebelum (lanjutan)

Hasil	Sistem menampilkan hasil pencarian daftar tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri berdasarkan <i>task name</i> , <i>person in charge</i> , dan <i>complete date</i> kemudian status pada tabel menampilkan banyak <i>item</i> yang ada.
Status	Valid.

2. Kasus Uji Cari Tugas Sebelum Alternatif 1 : Jika kata pencarian yang dicari tidak ada.

Tabel 6.107 Kasus Cari Tugas Sebelum Alternatif 1

Nama Kasus Uji	Cari Tugas Sebelum Alternatif 1 : Jika kata pencarian yang dicari tidak ada.
Prosedur	<ol style="list-style-type: none"> 1. Aktor menulis kata yang ingin dicari berdasarkan <i>workflow name</i> pada kolom pencarian yang tidak terdapat pada <i>database</i> sistem. 2. Aktor menekan <i>enter</i> pada <i>keyboard</i>.
Hasil yang diharapkan	Sistem tabel akan menampilkan <i>No items to display</i> .
Hasil	Sistem tabel akan menampilkan <i>No items to display</i> .
Status	Valid.

6.3.77 Pengujian Validasi Berhasil Lihat Isi *Form* Tugas Sebelum

1. Kasus Uji Berhasil Lihat Isi *Form* Tugas Sebelum

Tabel 6.108 Kasus Uji Berhasil Lihat Isi *Form* Tugas Sebelum

Nama Kasus Uji	Lihat Isi <i>Form</i> Tugas Sebelum
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas sebelum yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem menampilkan isi <i>form</i> tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri yang terdiri dari <i>subject</i> , <i>description</i> dan daftar <i>files</i> .



Tabel 6.108 Kasus Uji Berhasil Lihat Isi *Form* Tugas Sebelum (lanjutan)

Hasil	Sistem menampilkan isi <i>form</i> tugas sebelum tugas yang diberikan pada <i>person in charge</i> itu sendiri yang terdiri dari <i>subject</i> , <i>description</i> dan daftar <i>files</i> .
Status	Valid.

6.3.78 Pengujian Validasi Berhasil Unduh *File Individu Person In Charge Files*

1. Kasus Uji Berhasil Unduh *File Individu Person In Charge Files*

Tabel 6.109 Kasus Uji Berhasil Unduh *File Individu Person In Charge Files*

Nama Kasus Uji	Unduh <i>File Individu Person In Charge Files</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>form</i> 2. Aktor memilih dengan menekan <i>file</i> yang akan diunduh.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> yang dipilih oleh aktor sebelumnya.
Status	Valid.

6.3.79 Pengujian Validasi Berhasil Unduh Seluruh *File Person In Charge Files*

1. Kasus Uji Berhasil Unduh Seluruh *File Person In Charge Files*

Tabel 6.110 Kasus Uji Berhasil Unduh Seluruh *File Person In Charge Files*

Nama Kasus Uji	Unduh Seluruh <i>File Person In Charge Files</i>
Prosedur	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar <i>file</i> pada <i>form</i>. 2. Aktor menekan tombol <i>download</i>.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Hasil	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Status	Valid.



6.3.80 Pengujian Validasi Berhasil Menyelesaikan Tugas

1. Kasus Uji Berhasil Menyelesaikan Tugas

Tabel 6.111 Kasus Uji Berhasil Menyelesaikan Tugas

Nama Kasus Uji	Menyelesaikan Tugas
Prosedur	<ol style="list-style-type: none"> 1. Aktor mengisikan isi <i>subject</i> pada <i>field</i> yang disediakan. 2. Aktor mengisikan isi <i>description</i> pada <i>field</i> yang disediakan. 3. Aktor mengunggah <i>file</i> yang ingin diunggah. 4. Aktor menekan tombol <i>complete</i>.
Hasil yang diharapkan	Sistem akan mengizinkan <i>web browser</i> aktor untuk mengunduh <i>file</i> secara keseluruhan dengan format <i>zip</i> pada daftar <i>shared files</i> .
Hasil	Sistem dapat menyelesaikan tugas yang diberikan <i>workflow manager</i> dengan mengisikan <i>subject</i> , <i>description</i> , dan <i>files</i> yang ingin diunggah.
Status	Valid.

6.3.81 Pengujian Validasi Berhasil Unggah File Tugas

1. Kasus Uji Berhasil Unggah File Tugas

Tabel 6.112 Kasus Uji Unggah File Tugas

Nama Kasus Uji	Unggah File Tugas
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan tombol <i>select files</i>. 2. Sistem akan menampilkan direktori <i>folder</i> aktor. 3. Aktor memilih <i>file</i> yang akan diunggah. 4. Sistem menampilkan nama <i>file</i>, tipe <i>file</i>, dan ukuran <i>file</i>.
Hasil yang diharapkan	Sistem akan menyimpan <i>file</i> yang sudah dipilih oleh aktor sebelumnya yang akan terunggah ketika tugas diselesaikan.
Hasil	Sistem akan menyimpan <i>file</i> yang sudah dipilih oleh aktor sebelumnya yang akan terunggah ketika tugas diselesaikan.

Status	Valid.
--------	--------

6.3.82 Pengujian Validasi Berhasil Lihat *Workflow* Tugas

1. Kasus Uji Berhasil Lihat *Workflow* Tugas

Tabel 6.113 Kasus Uji Lihat *Workflow* Tugas

Nama Kasus Uji	Lihat <i>Workflow</i> Tugas
Prosedur	<ol style="list-style-type: none"> 1. Aktor memilih dengan menekan baris tugas yang akan dipilih. 2. Aktor menekan tombol <i>view</i>.
Hasil yang diharapkan	Sistem akan menampilkan <i>workflow</i> yang pada tugas yang diberikan <i>workflow manager</i> pada bagian <i>workflow preview</i> .
Hasil	Sistem akan menampilkan <i>workflow</i> yang pada tugas yang diberikan <i>workflow manager</i> pada bagian <i>workflow preview</i> .
Status	Valid.

6.3.83 Pengujian Validasi Berhasil Lihat Daftar Notifikasi *Person In Charge*

1. Kasus Uji Berhasil Lihat Daftar Notifikasi *Person In Charge*

Tabel 6.114 Kasus Uji Lihat Daftar Notifikasi *Person In Charge*

Nama Kasus Uji	Terima Notifikasi <i>Workflow Manager</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor membuka menu <i>notifications</i>.
Hasil yang diharapkan	Sistem akan menampilkan daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi masuk.
Hasil	Sistem akan menampilkan daftar notifikasi berdasarkan pesan notifikasi dan tanggal notifikasi masuk.
Status	Valid.

6.3.84 Pengujian Validasi Berhasil Lihat *Profile Person In Charge*

1. Kasus Uji Berhasil Lihat *Profile Person In Charge*

Tabel 6.115 Kasus Uji Berhasil Lihat *Profile Person In Charge*

Nama Kasus Uji	Lihat <i>Profile Admin</i>
----------------	----------------------------



Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown Person In Charge</i>. 2. Aktor menekan menu <i>profile</i>
-----------------	--

Tabel 6.115 Kasus Uji Berhasil Lihat *Profile Person In Charge* (lanjutan)

Hasil yang diharapkan	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
Hasil	Sistem menampilkan <i>profile</i> berdasarkan <i>full name, department, role, dan email</i> .
Status	Valid.

6.3.85 Pengujian Validasi Berhasil Ubah *Password Person In Charge*

1. Kasus Uji Berhasil Ubah *Password Person In Charge*

Tabel 6.116 Kasus Uji Berhasil Ubah *Password Person In Charge*

Nama Kasus Uji	Ubah <i>Password Person In Charge</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown Person In Charge</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i>. 5. Aktor menekan tombol <i>change password</i>.
Hasil yang diharapkan	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.
Hasil	Sistem menampilkan pesan <i>password changed</i> sekaligus sistem mengubah <i>password</i> dari aktor.
Status	Valid.

2. Kasus Uji Ubah *Password Person In Charge* Alternatif 1: Jika kolom *password* baru dan konfirmasi *password* baru kosong atau tidak sama

Tabel 6.117 Kasus Uji Berhasil Ubah *Password Person In Charge*

Nama Kasus Uji	Ubah <i>Password Person In Charge</i> Alternatif 1: Jika kolom <i>password</i> baru dan konfirmasi <i>password</i> baru kosong atau tidak sama
-----------------------	--

Tabel 6.117 Kasus Uji Berhasil Ubah *Password Person In Charge* (lanjutan)

Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown Person In Charge</i>. 2. Aktor menekan menu <i>profile</i>. 3. Aktor mengisi <i>password</i> baru pada kolom <i>new password</i>. 4. Aktor mengisi konfirmasi <i>password</i> baru pada kolom <i>new password (again)</i> tidak sama dengan kolom <i>new password</i> sebelumnya. 5. Aktor menekan tombol <i>change password</i>
Hasil	Sistem menampilkan menampilkan pesan <i>password empty or didn't match</i> .
Status	Valid.

6.3.86 Pengujian Validasi Berhasil *Logout*

1. Kasus Uji Berhasil *Logout*

Tabel 6.118 Kasus Uji Berhasil *Logout*

Nama Kasus Uji	<i>Logout</i>
Prosedur	<ol style="list-style-type: none"> 1. Aktor menekan <i>dropdown</i> pada nama <i>role</i> aktor. 2. Aktor menekan tombol <i>logout</i>
Hasil yang diharapkan	Sistem akan mengeluarkan aktor dari sistem menuju ke halaman <i>login</i> .
Hasil	Sistem akan mengeluarkan aktor dari sistem menuju ke halaman <i>login</i> .
Status	Valid.

6.4 Pengujian *Usability*

Pada pengujian *usability* diuji dengan metode *System Usability Scale (SUS)* yaitu sistem yang telah dibuat sebelumnya diberikan kepada pengguna untuk dijadikan responden. Selanjutnya setelah sistem diujikan oleh responden tersebut maka responden diberikan pertanyaan sebanyak sepuluh pertanyaan yang berisi tentang kemudahan penggunaan sistem yang kemudian diukur dengan skala *likert*. Skala *likert* merupakan skala yang digunakan pada *SUS*

untuk mengukur tingkat kemudahan dalam menggunakan sistem yang terdapat lima skala dari satu sampai lima. Skala satu sampai lima pada *likert* masing-masing yaitu sangat tidak setuju, tidak setuju, netral, setuju dan sangat setuju. Dari sepuluh pertanyaan sebelumnya yang diberikan untuk nomor ganjil akan berisi pertanyaan positif dan sebaliknya genap akan berisi pertanyaan negatif dalam penggunaan sistem. Jika menjawab pertanyaan ganjil nilai dikurangi satu, dan untuk genap dikurangi lima. Selanjutnya dari total hasil seluruhnya akan dikalikan dengan 2,5 yang kemudian diperoleh ukuran tingkat *usability* dari sistem tersebut. Pada penilaian akhir jika nilai tersebut bernilai 0 – 50,9 untuk SUS, 51 – 70,9 marginal, dan 71 – 100 untuk *acceptable*. Selanjutnya terdapat daftar pertanyaan yang diberikan kepada responden yang dapat dilihat pada Tabel 6.122.

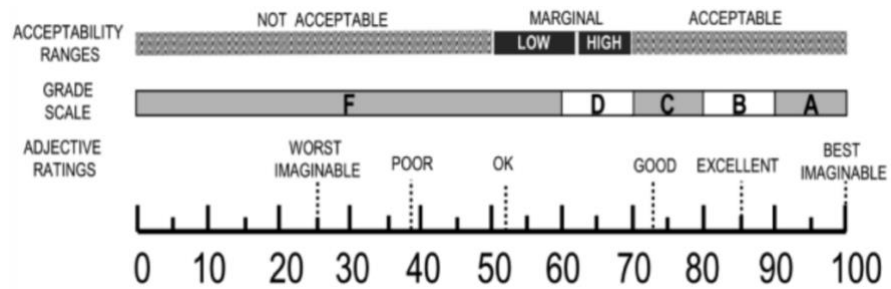
Tabel 6.119 Daftar Pertanyaan SUS

No	Pertanyaan
1	Sistem ini dapat mempermudah pekerjaan saya.
2	Saya perlu diberikan pengajaran di awal untuk menggunakan sistem ini.
3	Informasi yang diperlukan pada sistem sudah lengkap.
4	Bahasa yang digunakan sistem sulit dipahami.
5	Sistem ini sangat diperlukan pada perusahaan agar manajemen kerja lebih teratur.
6	Saya sangat sulit memahami sistem ini.
7	Tampilan dan instruksi pada sistem mudah dipahami.
8	Saya menemukan banyak ketidak konsistenan pada sistem ini.
9	Saya merasa fitur dalam sistem sudah terintegrasi dengan baik.
10	Saya merasa ada fitur pada sistem yang sulit digunakan.

6.4.1 Hasil Pengujian *Usability*

Setelah responden menjawab berdasarkan pertanyaan yang telah diberikan, maka selanjutnya akan dilakukan analisa dari hasil tersebut. Dari jawaban seuruh responden didapatkan hasil skala *likert* yang akan dilakukan perhitungan hasil skor untuk mendapatkan hasil akhir dari pengujian *usability* metode SUS. Hasil jawaban dari responden dapat dilihat pada Tabel 6.123. Dari hasil responden dihitung dengan melihat urutan pertanyaan jika menjawab pertanyaan ganjil nilai dikurangi satu, untuk genap dikurangi lima dan terakhir total seluruhnya dikalikan dengan 2,5. Selanjutnya hasil rata-rata dari total keseluruhan didapatkan dan dicocokkan dengan skala konversi SUS yang akan menentukan kategori dari usabilitas sistem. Skala konversis SUS dapat dilihat

pada Gambar 6.5. Selanjutnya hasil perhitungan pertanyaan pengujian *usability* SUS dapat dilihat pada Tabel 6.124.



Gambar 6.5 Rating dan Skala Konversi SUS

(Sumber : Brooke, 2011)

Tabel 6.120 Hasil Kuesioner Pertanyaan SUS

No	Nama		
	Bayudita	Nando	Adi
1	5	5	4
2	3	3	3
3	4	3	4
4	2	2	2
5	4	5	5
6	2	2	2
7	3	4	3
8	2	2	2
9	5	5	4
10	2	2	2

Tabel 6.121 Hasil Perhitungan Skor Pertanyaan SUS

No	Nama		
	Bayudita	Nando	Adi
1	4	4	3
2	2	2	2
3	3	2	3
4	3	3	3



5	3	4	4	
6	3	3	3	

Tabel 6.124 Hasil Perhitungan Skor Pertanyaan SUS (lanjutan)

No	Nama			
	Bayudita	Nando	Adi	
7	4	3	2	
8	2	3	3	
9	3	4	3	
10	3	3	3	
Total	30	31	29	Rata - Rata
Total x 2.5	75	77,5	72,5	75

Berdasarkan hasil perhitungan dari jawaban responden maka didapatkan hasil total skor *likert* yang telah dianalisis. Hasil yang diperoleh yaitu dengan skor akhir 75. Berdasarkan rating dan skala konversi SUS maka sistem ini termasuk dalam kategori *acceptable* yang diartikan sistem ini sudah memenuhi kebutuhan pengguna.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil dari setiap tahapan pada fase analisis kebutuhan, perancangan, implementasi, dan pengujian sistem maka dapat disimpulkan seperti berikut:

1. Berdasarkan dari hasil analisis kebutuhan didapatkan hasil yaitu 84 kebutuhan fungsional dan 1 kebutuhan non fungsional. Pada tahapan tersebut juga didapatkan aktor berjumlah 4 yaitu, pengunjung, admin, *workflow manager*, dan *person in charge*. Hasil pada tahapan analisis kebutuhan didapatkan dari hasil studi literatur dan hasil wawancara yang sesuai dengan permintaan klien.
2. Berdasarkan hasil pada tahap perancangan dari hasil analisis kebutuhan didapatkan hasil yaitu perancangan *sequence diagram*, perancangan *class diagram*, perancangan komponen, perancangan basis data yang berisi *Entity Relationship Diagram* (ERD) dan *Physical Data Model* (PDM), dan perancangan antarmuka.
3. Berdasarkan hasil pada tahap implementasi didapatkan hasil implementasi sistem, spesifikasi sistem, implementasi basis data, implementasi kode program, dan implementasi antarmuka.
4. Berdasarkan hasil pada tahap pengujian didapatkan hasil 100% valid pada tiga pengujian unit, satu pengujian integrasi, dan 84 pengujian validasi. Kemudian pada pengujian *usability* didapatkan nilai 75 dengan metode *System Usability Scale* yang berarti mendapatkan kategori *acceptable* yaitu sudah sesuai dengan kebutuhan pengguna.

7.2 Saran

Saran yang diberikan untuk pengembangan lanjut sistem *workflow* manajemen kerja PT. Chevron Pacific Indonesia adalah:

1. Menambahkan kebutuhan pada saat sudah dilakukan penugasan bisa mengganti *person in charge* yang bertanggung jawab saat proses sudah berjalan.
2. Menambahkan fitur untuk memperbaiki tenggat waktu secara otomatis sehingga akan mengurangi kemungkinan *person in charge* dalam menyelesaikan tugasnya melebihi tenggat waktu yang telah diberikan oleh *workflow manager* sebelumnya.
3. Penambahan aktor baru yang dapat melakukan verifikasi suatu tugas dari *person in charge*.
4. Menambahkan fitur untuk memungkinkan penugasan secara parallel yang sebelumnya hanya bisa dilakukan secara sekuensial.

DAFTAR REFERENSI

- Altuhhov, O., Matulevičius, R. and Ahmed, N., 2013. An Extension of Business Process Model and Notation for Security Risk Management. *International Journal of Information System Modeling and Design*, 4(4), pp.93–113.
- Beyrouiti, N., 2006 The Impact of Technological Innovation on Organizations, Work Environment and Personal Lives. Istanbul: Technology Management for the Global Future - PICMET 2006 Conference.
- Bhandari, R.K. and Nair, N., 2013. Building Mobile Applications Using Kendo UI Mobile and ASP.NET Web API.
- Brooke John. 2011. Measuring Usability With The System Usability Scale (SUS). <http://www.measuringu.com/sus.php>. Diakses pada tanggal 26 Desember 2018.
- Burbeck, Steven., 1992. Application Programmings in Smaltalk's 80™ : How To Use MVC.
- Chaffey, D., 1998. Groupware, Workflow and Intranets : Reengineering the Enterprise with Collaborative Software
- Chandra, A. and Prasetya, W., 2015. Peningkatan produktivitas melalui peningkatan kualitas kerja, kepemimpinan, dan kompensasi pada PT. KMK Global Sports. *Jurnal WIDYA Ekonomika*, 1(1), hal.58–63.
- Clark, D. et al., 2011. *Beginning C# Object- Oriented Programming Beginning C# Object-Oriented Programming*. New York: Springer Science.
- Deacon, John., 2009. Model-View-Controller (MVC Architecture), pp.1–6.
- Dewi, L. P., Indahyanti, U. dan S, Y. H., 2010. Pemodelan Proses Bisnis Menggunakan *Activity Diagram* UML Dan BPMN Studi Kasus FRS Online , hal.1-9.
- Domingos, D., Respício, A. and Martinho, R., 2016. Using Resource Reliability in BPMN Processes. *Procedia Computer Science*.
- Febijyantini, P. D. et al., 2015. Faktor-Faktor Yang Mempengaruhi Produktivitas Pada UD Wahyu Artha Di Desa Menyali Kabupaten Buleleng Tahun 2013.
- Fischer, 2002. *Workflow Handbook*. Florida: Future Strategies.
- Grimmer, T., 2004 Student's Essential Guide to .NET.
- Mistry, R., 2012. Microsoft® SQL Server 2012 Management and Administration, Second Edition.
- Nuskiya, A. F., 2018. The Effect of Information Technology on Employees Performance in the Banking Industry in Sri Lanka . Empirical Study Based on the Banks in Ampara District.
- Orellana, G., Laghari, G., Murgia, A., & Demeyer, S. 2017. On the Differences between Unit and Integration Testing in the TravisTorrent Dataset.
- Pressman RS, Ph.D., Maxim BR, Ph.D., 2015. *Software Engineering: Horn, S., Microsoft Silverlight 3: A Beginner's Guide*. New York : McGraw-Hill Companies.
- Rouf, A., 2012. Pengujian Perangkat Lunak dengan Menggunakan Metode



- Whitebox dan Blackbox*.pp.1–7.
- Sauer, C. et al., 2000. The Effectiveness of Software Development Technical Reviews: A Behaviorally Motivated Program of Research. *IEEE Transactions on Software Engineering*, 26(1), pp.1–14.
- Shaqiri, A.B., 2015. Impact of Information Technology and Internet in Businesses. *Academic Journal of Business, Administration, Law and Social Sciences*, 1(1), hal.73–79.
- Shahzad, F. et al., 2016. A Review of Latest Web Tools and Libraries for State-of-the-art Visualization. *Procedia Computer Science*, 58(Euspn), pp.100–106. Available at: <<http://dx.doi.org/10.1016/j.procs.2016.09.017>>.
- Sommerville., 2011. *Software engineering, 9th Edition*. USA: Pearson Education .
- Suhartono, I., 2009. Manajemen Kinerja Pada Perusahaan Bisnis Dari Manajemen Kinerja Tradisional Ke Manajemen Kinerja Baru.pp.1–14.
- Sujono, A.G. and Nugroho, H.A., 2015. Sistem Informasi Administrasi Pelatihan dengan Manajemen Workflow Berbasis Web Information System of Training Administration using Web Based Workflow Management. 18(2), pp.83–92.
- Syarif, A.A., Sinulingga, S. and Nazaruddin, 2014. Penentuan faktor yang mempengaruhi produktivitas tenaga kerja di pt. intan suar kartika dan rancangan strategi perbaikan. 1(1)hal.8–14.
- Whitten, J.L. and Bentley, L.D., 2007. *System Analysis & Design Methods. 7th Edition*. New York.