

20.526/H/04

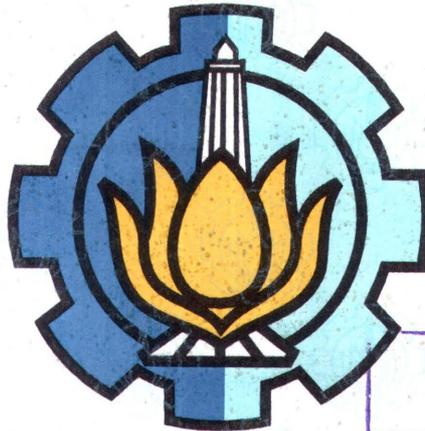
20.337/H/04



PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

PERANCANGAN DAN PEMBUATAN SISTEM MANAGEMENT OF CHANGE MENGUNAKAN TEKNOLOGI J2EE: STUDI KASUS PADA UNOCAL INDONESIA CO.

TUGAS AKHIR



RSIF
005-1
kur
P-1
2004

PERPUSTAKAAN ITS	
Tgl. Terima	20-2-2004
Terima Dari	H
No. Agenda Prp.	29581

Disusun oleh :

RIZKI AGUS ZANDRA KURNIAWAN

5198 100 034

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2004**

**PERANCANGAN DAN PEMBUATAN SISTEM
MANAGEMENT OF CHANGE
MENGUNAKAN TEKNOLOGI J2EE:
STUDI KASUS PADA UNOCAL INDONESIA CO.**

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer
Pada
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya

Mengetahui / Menyetujui,

Dosen Pembimbing I



Yudhi Purwananto, S.Kom, M.Kom

NIP. 132 172 210

Dosen Pembimbing II

all.
 5/2 '04

Rully Soelaiman, S.Kom, M.Kom

NIP. 132 085 802

**SURABAYA
Januari, 2004**



*kupersembahkan
kepada ibunda, ayahanda, dik rully, dik reny, dik rosy tersayang
serta semua yang telah mendukung dan membantu hingga tugas akhir ini selesai*



ABSTRAK

ABSTRAK

Sebagai perusahaan besar dengan berbagai macam unit bisnis dan pekerjaan, UNOCAL Indonesia Co memiliki banyak proses-proses bisnis. Tuntutan akan adanya informasi yang cepat dan akurat juga semakin meningkat demi mengoptimalkan kinerja dari perusahaan. Karena itu, perusahaan memerlukan adanya sistem-sistem elektronik yang akan membantu menjadikan proses-proses bisnis yang ada dapat berjalan secara lebih baik dan efisien, terutama dalam hal waktu. Salah satu diantara sistem-sistem tersebut adalah sistem pengajuan perubahan elektronik (e-MOC).

Dalam tugas akhir ini dikembangkan sistem Management of Changes yang berbasis web dengan menggunakan teknologi J2EE. Dengan dikembangkannya sistem e-MOC maka proses pengajuan dan persetujuan MOC yang selama ini masih manual, menjadi sistem elektronik yang mudah diakses dari intranet perusahaan. Sistem ini dibangun berdasarkan studi literatur dan permintaan pengguna, serta disesuaikan dengan batasan-batasan yang ditentukan oleh perusahaan. Berdasarkan hasil studi dan penggalan requirement tersebut, maka dapat dikembangkan desain sistem e-MOC. Setelah desain ini dikembangkan, selanjutnya dilakukan simulasi dengan beberapa data contoh dari perusahaan.

Hasil yang diperoleh dengan dikembangkannya sistem ini berupa dokumen pengajuan perubahan, hasil analisis risiko (hazard screening) dan penutupan proyek perubahan di Unocal Indonesia Co. Sistem ini diharapkan secara khusus agar dapat mempermudah karyawan dalam melakukan pengajuan perubahan dan secara umum dapat membantu meningkatkan kinerja perusahaan.

KATA PENGANTAR

Segala puji syukur ke hadirat Allah SWT, karena hanya dengan kehendak dan kuasa-Nya, Penulis dapat menyelesaikan Tugas Akhir yang berjudul

"PERANCANGAN DAN PEMBUATAN SISTEM

MANAGEMENT OF CHANGE

MENGGUNAKAN TEKNOLOGI J2EE :

STUDI KASUS PADA UNOCAL INDONESIA CO. "

Tugas Akhir dengan beban 4 SKS ini disusun dan diajukan sebagai salah satu syarat untuk menyelesaikan program Strata Satu (S1) pada jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa tugas akhir ini jauh dari sempurna, masih banyak kekurangan-kekurangan. Oleh karena itu sangat diharapkan kritik dan saran yang membangun untuk perbaikan kedepan.

Akhirnya, penulis mengucapkan terima kasih kepada pihak-pihak yang telah membantu terselesaikannya tugas akhir ini.

Surabaya, Januari 2004

Penulis



UCAPAN TERIMA KASIH

UCAPAN TERIMA KASIH

Segala puji bagi Allah Swt yang Maha Pengasih lagi Maha Penyayang. Pada kesempatan ini, penulis menyampaikan penghargaan dan rasa terima kasih yang sebesar-besarnya atas bimbingan, dukungan, doa, serta bantuan yang diberikan kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini kepada:

1. Ibunda, Ayahanda, dik Rully, dik Reny dan dik Rosy , yang senantiasa memberi semangat, dukungan, dan doa, sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik.
2. Bapak Yudhi Purwananto, S.Kom, M.Kom selaku dosen pembimbing, yang telah memberikan banyak masukan dan bimbingan. Semoga Allah membalasnya dengan kebaikan yang lebih besar.
3. Bapak Rully Soelaiman, S.Kom, M.Kom selaku dosen pembimbing, yang telah memberikan begitu banyak dukungan, bimbingan dan pengertian selama pembuatan Tugas Akhir. Semoga Allah membalasnya dengan kebaikan yang lebih besar.
4. Bapak dan Ibu dosen, untuk segala ilmu dan pengetahuan yang diberikan.
5. Ita, terima kasih atas segala motivasi, bantuan dan dukungannya yang tak akan sanggup untuk penulis balas.
6. Bapak dan Ibu Munawir, dik Ferdy dan dik Lia yang telah begitu banyak memberikan dukungan dan bantuan, terutama selama berada di Balikpapan. Mohon maaf selalu penulis repotkan dalam begitu banyak hal.

7. Kendy dan Nisa', yang begitu penulis repotkan dengan berbagai urusan terkait dengan Tugas Akhir dan Yudisium. Terima kasih yang tak terhingga penulis ucapkan. Semoga Allah membalasnya dengan kebaikan yang lebih besar.
8. Aby, Indie, Kamal, Lintang, dan teman-teman yang terlibat banyak secara langsung pada detik-detik menjelang ujian. Terima kasih yang tak terhingga penulis ucapkan. Semoga Allah memudahkan urusan kalian juga.
9. Pihak UNOCAL Indonesia Co yang telah memberi kesempatan untuk dapat mengerjakan Tugas Akhir di sana.
10. Para *stakeholder* e-MOC: Ibu Tribudi Lestyaningsih, Bapak Steve Adrianto, Bapak Agus M. Manggala, Mas Dwi dan Mas Adriyan. Terima kasih atas semua kesempatan dan bantuan yang telah diberikan.
11. Teman-teman Admin Lab-Prog Informatika ITS: Kendy, Hariedi "Didiet" Muhammad, Aby, Cphee, Dimas, Wongky, mas Hoedan, mas Royyan, adik Iin kecil, Joko "sang saingan", Jakka, Gunna, Rozy, Gayuh dan teman-teman lainnya atas kebersamaan, hangatnya persaudaraan dan bantuan anda sekalian.
12. Adik-adikku semua: Cici', Nanum, Miemiek, Lidya. Terima kasih banyak atas motivasi dan doanya. Khusus dik Cici', selamat ya atas pernikahannya.
13. Teman-teman di UNOCAL Indonesia Co: Andri, Ita, Mas Dwi, Mas Yogie, Mas Gunadi, Mas Ma'mun, Mbak Mala, Mbak Puji, Mbak Inung, Mas Ferdi, Mas FIRRAR, Mas Aris, makasih atas masukan-masukan dan bantuannya selama pengerjaan TA. Makasih juga atas kebersamaannya, terutama saat melepas stress dengan bermain game.

14. Teman-teman penghuni HMTC: Adhiaxa, Kamal, Liga, Eli, mas Ajun, Angga, Kholimi, Anip, dan kawan-kawan. Terima kasih atas bantuan dan kebersamaannya.
15. Anak-Anak Don Arjo, terima kasih untuk kebersamaannya.
16. Teman-teman c0e yang tak dapat penulis sebut satu persatu, terima kasih atas persahabatan dan dukungannya selama ini. Terima kasih atas momen-momen yang begitu indah bersama anda semua.
17. Mas Yudhi, Pak Narno, Mbak Erna, Mbak Eva, Pak Kadir, Mas Sugeng dan pegawai TU lain, yang sudah sering membantu penulis dalam urusan administrasi.
18. Teman-temanku para pendekar Counter-Strike: Ade “Rambo”, Cphee “Segopecelus”, Heri “VeteranAfghan”, Dimas “Gegana”, Didiet “Lumpia_Wrapper”, Joko, Jakka, Ajun, dan kawan-kawan atas kebersamaannya dalam menemani penulis mengusir stress semasa kuliah dan tugas akhir. Juga untuk semua pemain AOE dan Starcraft, penulis ucapkan terima kasih. Salam hangat buat anda dari FFoxie a.k.a Pasingsingan.
19. Buat teman-teman kos lama GL 10 makasih atas kebersamaannya.
20. Semua pihak yang telah mendukung penulis hingga Tugas Akhir ini selesai. Mohon maafkan, penulis tidak dapat menuliskan nama-nama anda semuanya.

Akhir kata, penulis memohon maaf yang sebesar-besarnya apabila penulis tidak sengaja pernah melakukan atau mengatakan hal-hal yang tidak berkenan.



DAFTAR ISI

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR.....	ii
UCAPAN TERIMA KASIH.....	iii
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan dan Manfaat	3
1.4 Ruang Lingkup dan Batasan Masalah.....	3
1.5 Metodologi Pembuatan Tugas Akhir	4
1.6 Sistematika Penulisan.....	5
BAB II Sistem Persetujuan Elektronik MOC (<i>Management of Changes Electronic Approval System</i>).....	8
2.1 Latar Belakang Pengembangan Sistem <i>MOC</i>	8
2.2 Pengertian dan Tujuan MOC	10
2.2.1 Pengertian.....	10
2.2.2 Tujuan MOC	11
2.3 <i>Management of Change (MOC)</i>	11
2.3.1 Ruang Lingkup MOC.....	11
2.3.2 Jenis Perubahan.....	12
2.3.3 Koordinasi MOC.....	15
2.3.4 Koordinasi Proyek (terkait MOC).....	16
2.3.5 Aturan-aturan dalam MOC	17
2.4 Alur Proses Pengajuan MOC	17
2.5 Struktur Organisasi Perusahaan	22
2.6 Sistem MOC Electronic Approval (sistem <i>e-MOC</i>)	24
2.5.1 Form	25
2.5.2 Report.....	28
2.5.3 Kebutuhan Lain.....	28
2.5.4 Aturan-aturan dalam proses bisnis	29
BAB III TEKNOLOGI J2EE DAN MVC DESIGN PATTERN.....	30
3.1 Java 2 Platform, Enterprise Edition (J2EE)	30
3.1.1 Enterprise JavaBeans (EJB)	31
3.1.2 Java Servlets.....	39
3.1.3 Java Server Pages (JSP)	41
3.1.4 Servlet Filters	42
3.1.5 Application Programming Interface J2SE dan J2EE	44
3.2 Lightweight Directory Access Protocol (LDAP).....	47
3.3 Model-View-Controller (MVC) Design Pattern.....	49

3.3.1 Definisi Pattern.....	50
3.3.2 Komponen Arsitektur MVC.....	50
3.4 Pattern pendukung lain yang digunakan	53
3.4.1 Single Access Point Pattern	53
3.4.2 Intercepting Filter.....	53
3.4.3 Service Locator Pattern.....	54
Bab IV PERANCANGAN dan PEMBUATAN SISTEM e-MOC.....	55
4.1 Elemen Sistem MOC.....	55
4.2 Perancangan Perangkat Lunak	60
4.2.1 Perancangan Aplikasi dengan MVC Pattern.....	60
4.2.2 Use Case Diagram.....	69
4.2.3 Perancangan Data.....	76
4.2.5 Perancangan Proses (Realisasi Use-Case)	86
4.2.6 Perancangan Form.....	115
4.2.7 Perancangan Report	122
4.2.8 Perancangan Menu Antarmuka	123
4.3 Pembuatan Perangkat Lunak.....	125
4.3.1 Implementasi Model-View-Controller Pattern	126
4.3.2 Implementasi Data	132
4.3.3 Implementasi Keamanan Sistem	138
4.3.4 Implementasi Antar Muka.....	140
BAB V UJI COBA PERANGKAT LUNAK.....	141
5.1 Uji Coba Aplikasi Webman	141
5.2 Uji Coba Sistem e-MOC	146
5.2.1 Skenario Uji Coba 1	147
5.2.2 Skenario Uji Coba 2	161
5.2.3 Skenario Uji Coba 3	164
BAB VI PENUTUP	169
6.1 Kesimpulan	169
6.2 Kemungkinan Pengembangan.....	169
DAFTAR PUSTAKA	171
LAMPIRAN.....	172

DAFTAR GAMBAR

Gambar 2.1 Alur Persetujuan MOC	19
Gambar 2.1 Struktur Jabatan Departemen IS.....	24
Gambar 3.1 Logical view dari filter	42
Gambar 3.2 Model hierarki tradisional.....	48
Gambar 3.3 Model hierarki internet.....	48
Gambar 4.1 Dispatcher View diagram sequence e-MOC	62
Gambar 4.3 Diagram <i>class</i> dari Controller	65
Gambar 4.4 Originator	71
Gambar 4.5 Supervisor of Originator	71
Gambar 4.6 Chief E&C Engineer.....	72
Gambar 4.7. Manager & Lead Process Engineer	72
Gambar 4.8. Project Leader.....	74
Gambar 4.9 Superintendent.....	74
Gambar 4.10. Supervisor of PL	75
Gambar 4.11. MOC Coordinator	76
Gambar 4.12 Diagram Sequence Pengajuan MOC	89
Gambar 4.13 Diagram Class Pengajuan MOC	90
Gambar 4.14 Diagram Sequence Persetujuan Proposal MOC.....	93
Gambar 4.15 Diagram Class Pengajuan MOC	94
Gambar 4.16 Diagram Sequence Penentuan Approver proposal MOC.....	96
Gambar 4.17 Diagram Class Penentuan Approver proposal MOC.....	97
Gambar 4.18 Diagram Sequence Penunjukan Project Leader.....	99
Gambar 4.19 Diagram Class Penunjukan Project Leader.....	100
Gambar 4.20 Diagram Sequence Screening oleh Project Leader.....	102
Gambar 4.21 Diagram Class Screening oleh Project Leader.....	103
Gambar 4.22 Diagram Sequence Persetujuan terhadap Form 2 dan 3	105
Gambar 4.23 Diagram Class Persetujuan terhadap Form 2 dan 3.....	106
Gambar 4.24 Diagram Sequence Pengisian MOC Form 4	107
Gambar 4.25 Diagram Class Pengisian MOC Form 4	108
Gambar 4.26 Diagram Sequence Persetujuan Terhadap MOC Form 4	110
Gambar 4.27 Diagram Class Persetujuan Terhadap MOC Form 4.....	110
Gambar 4.28 Diagram Sequence Pengajuan Penutupan Proyek Perubahan	112
Gambar 4.29 Diagram Class Pengajuan Penutupan Proyek Perubahan	112
Gambar 4.30 Diagram Sequence Persetujuan terhadap penutupan proyek perubahan	114
Gambar 4.31 Diagram Class Persetujuan terhadap pengajuan penutupan proyek perubahan	114
Gambar 4.32 MOC Proposal (Form 1)	115
Gambar 4.33 Hazard Analysis (Form 2).....	117
Gambar 4.34 Form 3.....	118

Gambar 4.35 Form 4.....	119
Gambar 4.36 Form Persetujuan	120
Gambar 4.37 Form MOC Navigator.....	121
Gambar 4.38 Form 3 Siap Cetak.....	122
Gambar 4.39 Grafik Statistik MOC.....	123
Gambar 4.40 Rancangan Menu Antarmuka.....	124
Gambar 4.41 Lingkungan Implementasi Perancangan.....	125
Gambar 5.1 Login ke Aplikasi Webman.....	141
Gambar 5.2 Menambahkan Aplikasi Baru	142
Gambar 5.3 Form Pendaftaran Modul-Modul	142
Gambar 5.4 Menu Memasukkan Data Role.....	143
Gambar 5.5 Menentukan Hak Tiap Role	143
Gambar 5.6 Form Menambah User Baru.....	144
Gambar 5.7 Searching Pengguna dalam Webman	144
Gambar 5.8 Edit Data Pengguna Webman	145
Gambar 5.9 Data Pengguna “tutukor”	145
Gambar 5.10 Menentukan Role Pengguna Webman	145
Gambar 5.11 Login ke sistem e-MOC.....	146
Gambar 5.12 Proposal MOC (Form 1).....	148
Gambar 5.13 Proposal MOC (Form 1).....	148
Gambar 5.13 E-mail pemberitahuan terhadap Supervisor	149
Gambar 5.14 Daftar tugas (MyTask) dari Supervisor	149
Gambar 5.15 Formulir Persetujuan Proposal oleh Supervisor	150
Gambar 5.16 Daftar tugas MOC Coordinator.....	150
Gambar 5.17 Formulir Manage MOC	151
Gambar 5.18 Daftar tugas Superintendent	152
Gambar 5.19 Formulir Persetujuan Proposal bagi Superintendent.....	152
Gambar 5.20 Penetapan Project Leader.....	153
Gambar 5.21 Daftar tugas bagi Project Leader.....	153
Gambar 5.22 HA Screening Form (MOC Form 2).....	154
Gambar 6.23 MOC Plan.....	155
Gambar 6.24 Daftar tugas bagi Supervisor dari Project Leader	156
Gambar 6.25 Persetujuan terhadap Form 2 dan 3	157
Gambar 6.26 MOC Form 4.....	158
Gambar 6.27 Persetujuan terhadap Form 4.....	158
Gambar 6.28 Startup Check Form.....	159
Gambar 6.29 Closure Status Form.....	160
Gambar 5.30 Proposal MOC (Form 1).....	162
Gambar 5.31 Formulir Persetujuan Proposal oleh Supervisor	162
Gambar 5.32 Formulir Manage MOC	163
Gambar 4.33 Penolakan Proposal MOC oleh LPE.....	163
Gambar 5.34 Proposal MOC (Form 1).....	164
Gambar 5.35 Formulir Persetujuan Proposal oleh Supervisor	165
Gambar 5.36 Formulir Manage MOC	166
Gambar 5.37 Penetapan Project Leader.....	166
Gambar 5.38 Daftar tugas bagi Project Leader	167

Gambar 5.39 Daftar tugas bagi Project Leader setelah Submit ditekan.....	167
Gambar 5.40 Penolakan hasil screening oleh Supervisor of PL.....	168
Gambar 5.38 Daftar tugas Project Leader setelah penolakan.....	168

DAFTAR TABEL

Tabel 2.1 Daftar Departemen.....	23
Tabel 4.1. Daftar Peran dalam Sistem MOC.....	55
Tabel 4.2. Jalur Persetujuan	57
Tabel 4.3. Status MOC.....	59
Tabel 4.4 Master Area	78
Tabel 4.5 Master Field	78
Tabel 4.6 Screening Form Entries	79
Tabel 4.7 Record.....	79
Tabel 4.8 Master Status	80
Tabel 4.9 Attachment.....	81
Tabel 4.10 Nominated Project Leader	81
Tabel 4.11 Approval.....	82
Tabel 4.12 Master Action	83
Tabel 4.13 Remarks	83
Tabel 4.14 Persons.....	84
Tabel 4.15 Master Roles	85
Tabel 4.16 Reminder List.....	85
Tabel 5.1 Daftar User Percobaan	146





BAB I
PENDAHULUAN

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai sebuah perusahaan besar yang bergerak dalam industri pertambangan, khususnya pertambangan minyak dan gas bumi, UNOCAL Indonesia Company terlibat dalam berbagai kegiatan dalam skala besar maupun kecil di berbagai sektor untuk menunjang operasionalnya. Dalam berbagai kegiatan tersebut, tak dapat dielakkan bahwa ada perubahan yang dilakukan untuk mencapai tujuan perusahaan. Perubahan yang dimaksud dapat berupa perubahan fisik seperti penggantian alat dan sebagainya; namun juga perubahan yang bersifat non fisik seperti perubahan prosedur operasi standar (SOP).

Langkah perubahan yang diambil senantiasa menimbulkan konsekuensi, baik yang bersifat positif maupun negatif. Dalam lingkungan berisiko tinggi seperti pertambangan minyak dan gas bumi, sebuah langkah perubahan yang tidak dipertimbangkan dengan seksama dapat menimbulkan akibat yang fatal, seperti hilangnya nyawa, terlukanya personil, pencemaran lingkungan, rusaknya aset dan nama baik perusahaan.

Hal ini mendorong perusahaan untuk menyediakan sebuah sarana pengkajian terhadap perubahan (*change*) yang akan dilakukan beserta dampak-dampaknya, yaitu melalui prosedur MOC (*Management of Changes*).

Selama ini proses MOC ditangani secara manual, yaitu meminta persetujuan satu-per-satu kepada masing-masing pemberi persetujuan (approvers) dengan menggunakan form kertas (*hard-copy routing*). Hal ini menyebabkan proses pengajuan MOC menjadi tidak efisien, terutama dalam hal waktu proses. Sedangkan seringkali proses MOC terkait dengan hal yang perlu segera diselesaikan untuk menghindari kerugian dan bahaya yang mungkin timbul.

Sebab itu, diperlukan adanya sebuah aplikasi yang dapat mengotomatisasi proses pengajuan sebuah MOC. Dengan sistem ini proses pengajuan MOC sejak dari *originator* sampai MOC mendapatkan persetujuan agar dapat dijalankan, akan dilaksanakan secara otomatis sehingga mengurangi kelambatan proses yang sebelumnya terjadi.

Aplikasi tersebut disebut dengan *Management of Change (MOC) Electronic Approval System* atau disebut *sistem e-MOC*, yang akan dibangun sesuai dengan kebutuhan pengguna dan peraturan tentang MOC di Unocal Indonesia Company.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, permasalahan yang akan diselesaikan dalam tugas akhir adalah sebagai berikut:

1. Bagaimana melakukan identifikasi kebutuhan pengguna dari sistem MOC, yang meliputi identifikasi proses bisnis beserta aturan-aturan yang berlaku di dalamnya.
2. Bagaimana melakukan perancangan sistem pengajuan elektronik MOC untuk dapat memenuhi kebutuhan yang sudah teridentifikasi.

3. Bagaimana mengimplementasikan rancangan yang telah dibuat menjadi sebuah solusi dengan menggunakan teknologi Java 2 Enterprise Edition (J2EE), khususnya menggunakan teknologi EJB CMP 2.0, JavaBeans, JSP, Servlet 2.3, JNDI dan Java Mail.

1.3 Tujuan dan Manfaat

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Merancang dan membangun sistem pengajuan elektronik untuk Management of Change (MOC) di perusahaan Unocal Indonesia Company yang bersifat *paperless*, sehingga dapat lebih efisien dan relatif lebih aman dari kemungkinan hilang.
2. Mengimplementasikan sistem dengan teknologi J2EE yang bersifat multiplatform, sehingga perusahaan nantinya tidak perlu terikat pada platform tertentu. Selain itu teknologi J2EE telah dikenal sebagai teknologi yang cukup teruji dan berkapabilitas tinggi.

1.4 Ruang Lingkup dan Batasan Masalah

Sistem pengajuan elektronik ini dirancang dan dibangun dengan batasan-batasan berikut ini:

1. Studi kasus dari Tugas Akhir ini adalah sistem pengajuan manajemen perubahan (MOC) di perusahaan Unocal Indonesia Company yang berkedudukan di Balikpapan, Kalimantan Timur.
2. Proses bisnis dan aturan-aturan dalam sistem ini sesuai dengan yang berlaku di perusahaan tersebut.

3. Sistem ini dirancang dan dibangun dengan teknologi berikut ini, sesuai dengan permintaan perusahaan:
 - Sybase PowerDesigner 9, Rational Rose 2002 dan Microsoft Visio untuk permodelan sistem
 - Oracle Container for J2EE (OC4J) versi 1.0.2.2 sebagai J2EE Container. OC4J menyediakan dukungan terhadap penggunaan teknologi EJB CMP 2.0, JSP, Servlet 2.3, dan JNDI yang digunakan.
 - Oracle 8.1.7 sebagai database server

1.5 Metodologi Pembuatan Tugas Akhir

Tugas Akhir ini disusun menggunakan metodologi berikut ini:

1. Studi Literatur

Pada tahap ini dilakukan aktivitas mencari, mengumpulkan, dan mempelajari informasi yang berkaitan dengan *Management of Change*, peraturan perusahaan yang terkait dengan hal tersebut, dan teknologi yang dipergunakan.

2. Pengumpulan data dan kebutuhan pengguna

Pada tahap ini dilakukan pengumpulan data tentang sistem pengajuan MOC secara manual, keterkaitan dengan sistem lain yang sudah ada, dan mendefinisikan ekspektasi dari pengguna yang akan berinteraksi dengan sistem (*requirements engineering*).

3. Perancangan Sistem

Pada tahap ini, semua informasi tentang sistem MOC yang akan dibangun dituangkan menjadi sebuah rancangan. Rancangan ini mencakup aturan-

aturan(*rules*), kelas-kelas hak dari pengguna (*roles*), alur persetujuan umum dari MOC, dan prototipe sistem yang dapat dijadikan parameter untuk mengetahui sejauh mana kebutuhan pengguna sistem telah dapat terakomodasi.

4. Implementasi Rancangan

Pada tahap ini, rancangan-rancangan yang telah diverifikasi oleh *user* pada tahap sebelumnya akan diimplementasikan dengan teknologi yang akan dipergunakan.

5. Uji Coba Sistem

Pada tahap ini sistem akan diuji-coba untuk mengetahui apakah sistem yang dibangun sudah memenuhi spesifikasi yang didefinisikan pada perancangan sistem.

6. Penulisan Laporan

Pada tahap terakhir ini disusun sebuah buku sebagai laporan dan dokumentasi pelaksanaan Tugas Akhir dari awal sampai akhir.

1.6 Sistematika Penulisan

Sebagai gambaran umum dari laporan Tugas Akhir yang disusun, berikut ini akan dijelaskan tentang sistematika penyusunan laporan Tugas Akhir ini secara garis besar.

Bab I Pendahuluan

Bab ini membahas tentang latar belakang pembuatan sistem, permasalahan yang dihadapi, tujuan dan manfaat dari Tugas Akhir ini, ruang lingkup, batasan-batasan dan metodologi yang dipergunakan.

Bab II Management of Change (MOC) Electronic Approval System

Bab ini membahas tentang sistem yang akan dibangun. Hal yang dijelaskan antara lain tentang Management of Change, fitur-fitur sistem yang akan diimplementasikan sesuai dengan kebutuhan pengguna, aturan-aturan sistem, dan kebutuhan sistem itu sendiri.

Bab III Teknologi J2EE dan Design Pattern

Bab ini membahas tentang teori-teori penunjang yang dipergunakan dalam penyelesaian Tugas Akhir. Teori penunjang ini mencakup penjelasan tentang teknologi J2EE yaitu diantaranya EJB, JSP, Servlets, JNDI. Pada bab ini juga akan dibahas *design pattern* yang dipergunakan.

Bab IV Perancangan dan Penerapan Perangkat Lunak

Bab ini membahas tahapan-tahapan dari proses perancangan dan pembuatan perangkat lunak. Bagian ini terdiri dari perancangan data dan perancangan proses bisnis serta fungsi-fungsi yang dipergunakan dalam perancangan dan pembuatan perangkat lunak.

Bab V Implementasi dan Ujicoba Perangkat Lunak

Bab ini menjelaskan proses uji coba dan evaluasi perangkat lunak yang telah dikembangkan dengan mempergunakan data yang telah disiapkan.

Bab VI Penutup

Bab ini berisi kesimpulan dan saran bagi pengembangan sistem e-MOC.



BAB II

SISTEM PERSETUJUAN ELEKTRONIKA MOC (*MANAGEMENT OF CHANGES ELECTRONIC APPROVAL SYSTEM*)

BAB II

Sistem Persetujuan Elektronik MOC

(Management of Changes Electronic Approval System)

Dalam bab ini akan dibahas tentang *Management of Change* (MOC) *Electronic Approval System*, selanjutnya disebut sebagai Sistem e-MOC. Bab ini menjelaskan antara lain tentang latar belakang dan tujuan dari dikembangkannya sistem ini, serta beberapa peraturan perusahaan yang menjadi landasan operasionalnya.

Bab ini juga membahas tentang kebutuhan pengguna (*user requirements*) dari sistem e-MOC dan struktur organisasi yang sangat penting sebagai acuan perancangan sistem nantinya.

2.1 Latar Belakang Pengembangan Sistem MOC

Selama ini proses pengajuan MOC ditangani secara manual, yaitu meminta persetujuan satu-per-satu kepada masing-masing pemberi persetujuan (*hard-copy routing*). Hal ini menyebabkan proses pengajuan MOC menjadi tidak efisien, terutama dalam hal waktu proses. Sedangkan seringkali proses MOC terkait dengan hal yang perlu segera diselesaikan untuk menghindari kerugian dan bahaya yang mungkin timbul.

Sebab itu diperlukan adanya sebuah aplikasi yang dapat mengotomatisasi proses persetujuan sebuah MOC. Dengan sistem ini proses pengajuan MOC, akan

dilaksanakan secara otomatis sehingga dapat mengurangi kelambatan proses yang sebelumnya terjadi.

Sistem MOC bertujuan untuk memberikan solusi terhadap masalah yang timbul dalam proses pengajuan dan persetujuan MOC. Sistem ini dibangun dengan tujuan untuk menangani prosedur MOC secara lengkap, diantaranya untuk mempermudah proses pengajuan, permintaan persetujuan proposal MOC sesuai dengan jalur wewenang persetujuan (*approval routing*), penelusuran status proposal, pengarsipan dokumen MOC, dan lain-lain.

Sistem MOC dibangun sebagai sebuah aplikasi berbasis web agar lebih mudah diakses oleh pengguna sistem yang lokasinya tersebar secara geografis. Sebagai aplikasi web, sistem ini tidak membutuhkan instalasi perangkat lunak khusus pada masing-masing komputer klien yang akan mengaksesnya, kecuali browser web yang sudah termasuk instalasi standar.

Terdapat beberapa aplikasi yang berhubungan dan mendukung sistem MOC. Aplikasi-aplikasi ini antara lain:

1. HRIS (*Human Resources Information System*)

Sistem ini mengelola informasi seluruh pegawai yang terdapat di UNOCAL Indonesia Co dan pada skema HRIS terdapat satu tabel yang sangat bagi sistem MOC, yaitu tabel HRIS_EMPLOYEE. Tabel inilah yang menyimpan data seluruh pegawai perusahaan.



2. *Webman*

Dari aplikasi ini seseorang dapat didaftarkan untuk memiliki role dan hak tertentu terhadap suatu sistem. Karena itulah Webman digunakan untuk membantu proses autentikasi dan otorisasi. Webman akan membantu mengenali role dan hak setiap pegawai itu kembali, sesuai dengan yang telah didaftarkan. Seseorang dapat memiliki role dalam beberapa aplikasi dan dalam satu aplikasi dapat pula memiliki dua role yang berbeda.

2.2 Pengertian dan Tujuan MOC

Sub-bab ini menjelaskan tentang pengertian dan tujuan MOC yang sesuai dengan *Guideline* MOC milik perusahaan [UIC-2001].

2.2.1 Pengertian

Secara formal Unocal Indonesia Co. memberikan dua definisi tentang *Management of Change* (MOC), yaitu :

1. MOC adalah cara yang formal yang dipergunakan untuk mengidentifikasi, mengatur dan mengontrol risiko yang mungkin terjadi berkaitan dengan sebuah perubahan yang akan dilakukan. (*“MOC is a formalized way for identifying, managing, and controlling hazards associated with changes”*)
2. MOC merupakan istilah dalam bidang teknik yang merujuk pada pengontrolan terhadap perubahan yang memiliki kemungkinan akan adanya resiko. (*“MOC is an engineering term that refers to controlling changes that have any possible relation with hazards”*)

MOC diimplementasikan sebagai sekumpulan prosedur yang dipergunakan untuk mengontrol perubahan (*change*) yang akan dilaksanakan, mencakup persetujuan terhadap proposal MOC, analisis dan pengkajian terhadap dampak perubahan, kontrol atas status proyek pelaksanaan perubahan, persiapan penutupan proyek perubahan, dan sebagainya.

2.2.2 Tujuan MOC

Tujuan dari MOC adalah untuk memberikan sarana yang sistematis dan terkontrol sehingga dapat secara efektif mengidentifikasi, mengevaluasi, mencatat, dan mengontrol risiko pada saat terjadi penambahan atau perubahan terhadap fasilitas atau prosedur yang ada, guna meminimalkan risiko (*hazard*) yang mungkin terjadi di Unocal Indonesia Co.

2.3 Management of Change (MOC)

Sub-bab ini menjelaskan tentang beberapa hal yang terkait erat dengan sistem MOC, diantaranya landasan operasional MOC, ruang lingkup, jenis-jenis MOC dan aturan-aturan dasar lain yang terkait dengan proses dalam sistem MOC. Operasional MOC berlandaskan pada dokumen standar perusahaan, yaitu *Management of Changes Guideline*. Dokumen ini memberikan berbagai spesifikasi tentang berbagai hal dalam MOC, yang akan dijelaskan kemudian pada bagian selanjutnya.

2.3.1 Ruang Lingkup MOC

Management of Change diterapkan ke semua fasilitas perusahaan, baik fasilitas produksi maupun non-produksi yang terkena pengaruh perubahan

tersebut. Perubahan-perubahan yang dimaksud dapat berupa perubahan yang disebabkan oleh suatu proyek atau bisa juga perubahan yang diperlukan oleh proses operasional.

Untuk saat ini, Management of Change masih merupakan kebutuhan spesifik dari departemen Engineering & Construction (E&C).

2.3.2 Jenis Perubahan

Terdapat beberapa tipe perubahan yang harus ditangani dengan menggunakan MOC, antara lain:

1) Proyek Baru (*New Project*)

MOC dengan tipe “New Project” ini mencakup proses pembangunan atau instalasi sebuah sistem atau item baru.

Contoh :

- Instalasi pompa baru
- Instalasi modul peralatan terintegrasi (*integrated equipment modules*)
- Pembangunan struktur atau bangunan baru
- Percobaan terhadap peralatan baru dan *pilot testing*

2) Perubahan (*Changes*)

MOC jenis ini mencakup penggantian peralatan (*equipment change*), perubahan terhadap proses (*process change*), dan perubahan terhadap prosedur dan rencana kerja (*procedure and work plan changes*).

a. Perubahan terhadap peralatan (*Equipment Changes*)

Perubahan peralatan (*equipment changes*) ini mencakup perubahan terhadap struktur bangunan, pemipaan, kelistrikan, instrumentasi, mesin dan peralatan mekanis, serta peralatan keselamatan kerja. Jenis perubahan ini dapat berupa penambahan, pengurangan, modifikasi atau pemindahan terhadap peralatan.

Contoh :

- Meningkatkan kapasitas peralatan lift
- Mengganti sistem catu daya
- Menambahkan atau melepas *shutdown switch*
- Melakukan perubahan terhadap bangunan atau jalur pipa yang sudah ada

b. Perubahan Proses (*Process Changes*)

Perubahan proses (*process changes*) adalah perubahan yang melibatkan proses-proses yang berkaitan dengan produksi atau layanan. Perubahan ini mencakup modifikasi terhadap perangkat lunak komputer dan sistem kontrol, termasuk juga sistem pengontrolan proses dan pengukuran.

Contoh:

- Mengubah sebuah parameter operasi ke luar batas aman operasi yang selama ini dipergunakan
- Mengubah komposisi produk
- Mengubah logika pengontrol (*control logic*)
- Melakukan *bypass* terhadap *shutdown switch*

c. Perubahan Prosedur dan Rencana Kerja (*Procedure and work plan changes*)

Perubahan terhadap prosedur dan rencana kerja yang sudah ditetapkan harus juga dilaksanakan melalui prosedur MOC.

Contoh:

- Mengubah prosedur pengoperasian, perawatan, atau penanganan keadaan darurat yang sudah ditetapkan
- Mengubah urutan dari rencana kerja konstruksi

3) Replacement in Kind (RIK)

Replacement In Kind didefinisikan sebagai penggantian peralatan atau sistem dengan peralatan atau sistem pengganti yang sama dengan spesifikasi desain. Proses RIK terdiri dari dua jenis, yaitu RIK rutin dan RIK non rutin. MOC hanya perlu diterbitkan untuk *RIK non-rutin* saja.

Contoh RIK non rutin:

- Perbaikan tak rutin terhadap pompa atau kompresor karena satu dan lain hal.
- Mengganti *vessel* yang sudah usang sebelum masa berlakunya habis
- Penggantian *spool* pipa karena telah berlubang

2.3.3 Koordinasi MOC

Koordinasi MOC akan ditangani oleh MOC Coordinator E&C dan Area Coordinator dari masing-masing wilayah (*South Area* atau *North Area*), kecuali untuk departement Drilling yang ditangani secara terpisah. Pengarsipan dokumen MOC dilakukan secara terpusat di departemen E&C.

1) MOC Coordinator E & C

Bertanggung jawab atas MOC di lingkungan Pasir Ridge (kantor pusat Unocal Balikpapan) dan MOC secara umum di wilayah lain. Selain itu, MOC Coordinator E&C juga bertanggung jawab atas:

- Instalasi fasilitas baru yang dapat memberikan beban tambahan terhadap fasilitas yang sudah ada.
- Penambahan, pengurangan, pemindahan ataupun modifikasi terhadap struktur baja dan bangunan.
- Penambahan, pengurangan, pemindahan ataupun perbaikan terhadap *pressure vessel*.
- Penambahan, pengurangan, pemindahan terhadap katup pengaman (*safety valve*), termasuk juga perubahan konfigurasi.
- Penambahan, pengurangan, pemindahan ataupun modifikasi terhadap pipa pemrosesan utama (*main process piping*)
- Chemical treatments
- Pencegahan korosi

2) Area MOC Coordinator

Bertanggung jawab atas MOC di area masing-masing, yaitu North Area dan South Area. Masing-masing area memiliki beberapa *field* (lapangan minyak atau daerah kerja), seperti *Attaka*, *Melahin* dan *Kerindingan* di *North Area*; sedangkan *Sepinggan field* dan berbagai lapangan minyak lain di Teluk Balikpapan tergabung dalam *South Area*.

MOC Coordinator akan melakukan verifikasi terhadap proposal perubahan yang diajukan, dimana proposal tersebut akan diperiksa apakah sudah sesuai prosedur dan tidak melanggar Standar Teknis Perusahaan (*Company Engineering Standards*), proses kontrol risiko, dan bahwa perubahan tersebut membawa keuntungan terhadap perusahaan.

2.3.4 Koordinasi Proyek (terkait MOC)

Dalam MOC Guideline, dikatakan bahwa keberhasilan pengajuan perubahan sebagaimana telah diproses sesuai dengan prosedur MOC, tergantung pada seberapa jelas tugas yang diberikan kepada pemimpin proyek (*Project Leader*). Pemimpin proyek harus memiliki kemampuan perhitungan untuk memastikan bahwa MOC dilaksanakan sesuai dengan pekerjaan yang telah diberikan.

2.3.5 Aturan-aturan dalam MOC

Terdapat beberapa aturan yang terkait dengan prosedur pengajuan MOC, diantaranya yaitu:

1. Tidak seorangan yang akan memperkenalkan sebuah perubahan atau perbaikan Non routine, kecuali telah melalui semua prosedur MOC dan telah disetujui.
2. Semua perubahan harus yang akan diajukan, haruslah disaring secara tepat oleh orang-orang yang terbiasa dengan standar teknis, spesifikasi, operasi lapangan, dan peraturan yang berlaku.
3. Semua perubahan harus tercatat dan terlacak untuk memastikan registrasi yang tepat, proses yang benar sesuai dengan *request* awal, persetujuan sampai dengan penutupan proses.
4. MOC Coordinator (E & C) dan/atau Area MOC Coordinator dari masing-masing *field* akan memastikan penyaringan yang tepat dari semua perubahan dimana prosedur MOC telah diterapkan.
5. Fungsi koordinasi akan dikelola oleh Area MOC Coordinator dari masing-masing fields dan MOC Coordinator (E & C).

2.4 Alur Proses Pengajuan MOC

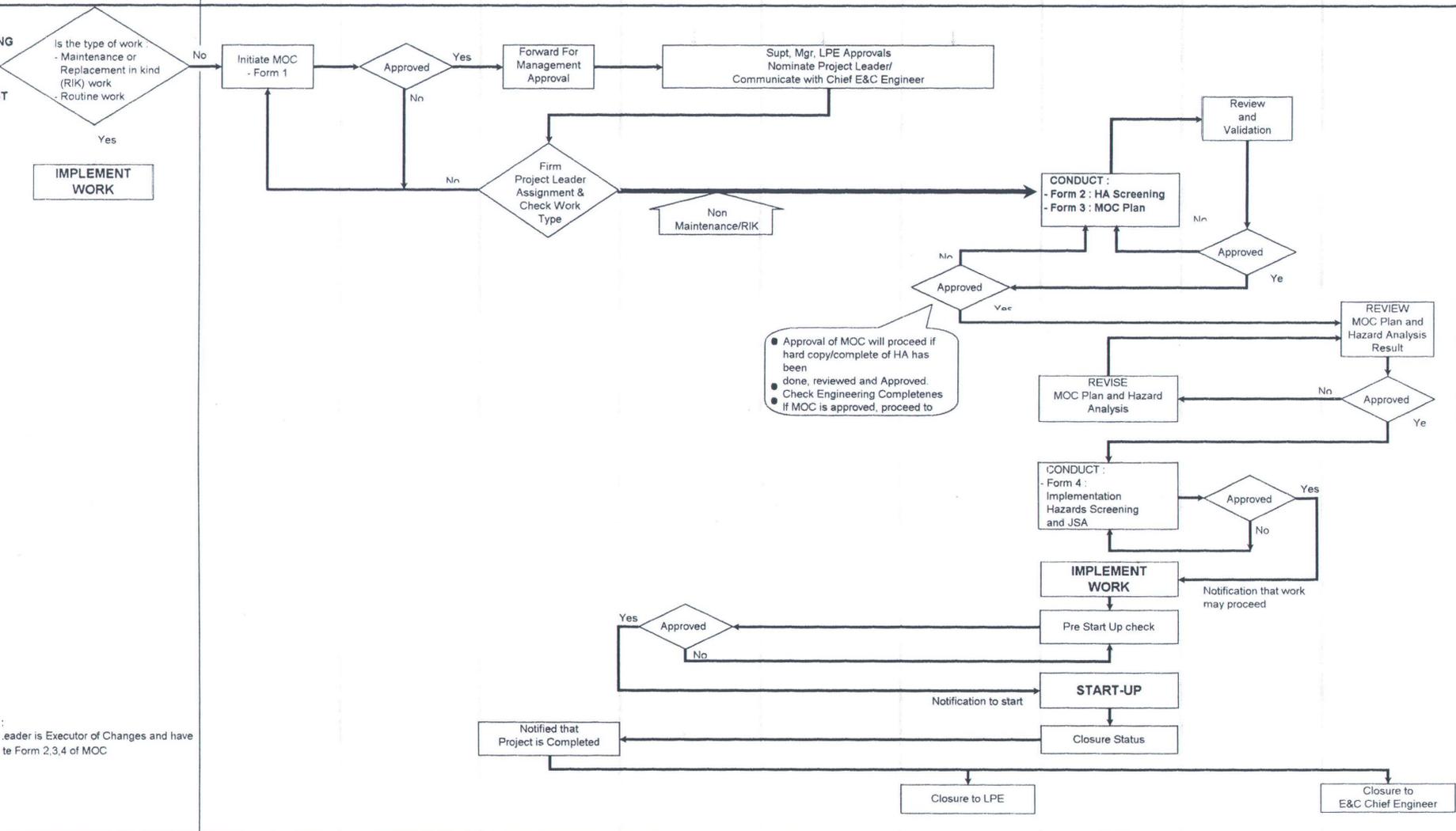
Pada diagram alur berikut ini akan terlihat bagaimana sebuah permohonan MOC diajukan, mulai dari proses pengajuan awal oleh Originator sampai dengan proses penutupan MOC oleh Lead Process Engineer dan Chief E&C Engineer. Proses ini telah menjadi kesepakatan bersama antara *Vice President* dan manager serta ahli-ahli terkait dibidangnya dan mengikuti aliran proses yang sesuai dengan

kondisi nyata dilapangan. Hal ini sangatlah penting karena suatu perubahan akan memberikan pengaruh bagi keseluruhan proses bisnis yang terjadi di perusahaan.

Aliran proses tersebut dapat dilihat pada gambar 2.1.

E - MOC PROCESS - Attachment 5 A

LPE (Lead Process Engineer) MOC Originator Supv of Originator MOC Coord as Navigator Supt (Field, Etc) Manager LPE (Lead Process Engineer) PL (Project Leader) Supervisor of PL Chief E&C Engineer



Gambar 2.1 Alur Persetujuan MOC

Penjelasan secara singkat dari alur pengajuan tersebut adalah sebagai berikut:

1. Originator mengajukan proposal MOC baru dengan mengisi formulir MOC dan melengkapinya dengan dokumen-dokumen pendukung sebagai attachment.
2. Supervisor dari Originator menilai kelayakan proposal MOC. Apabila disetujui, maka proposal akan diserahkan ke MOC Coordinator untuk diproses lebih lanjut. Sebaliknya, jika tidak disetujui proposal tidak dapat dilanjutkan.
3. MOC Coordinator akan menentukan orang-orang yang berhak untuk memberikan persetujuan (*approver*). Dalam tahap ini, orang-orang yang mempunyai kemungkinan dipilih untuk memberikan persetujuan antara lain manajer, *Superintendent*, *Lead Process Engineer* (LPE). MOC Coordinator juga berhak menentukan tiga orang pemberi persetujuan tambahan sesuai kebutuhan. MOC Coordinator dapat memilih semua atau hanya sebagian saja dari semua *approver* yang mungkin. Pertimbangan penentuan ini sepenuhnya diserahkan kepada MOC Coordinator.
4. Para *approver* akan memberikan persetujuan atau penolakannya terhadap proposal MOC. Apabila salah satu saja menolak, maka proposal tidak akan diproses lebih lanjut. Sebaliknya apabila semua menyetujui, maka proposal akan ditindak lanjuti dan kontrol dikembalikan kepada MOC Coordinator. Para pemberi persetujuan ini juga berhak untuk mencalonkan seorang *Project Leader* yang nanti diharapkan akan memimpin pelaksanaan proyek perubahan (*change project*) yang diminta dalam proposal MOC.

5. MOC Coordinator akan menentukan Project Leader untuk pelaksanaan proyek perubahan ini. MOC Coordinator dapat mempertimbangkan pilihan dari masing-masing pemberi persetujuan, namun dapat pula menunjuk orang lain.
6. Project Leader terpilih akan melakukan analisis risiko (*Hazard Analysis Screening*) dan membuat rencana pelaksanaan MOC (*MOC Plan*). Hasil penilaian Project Leader terhadap kedua hal tersebut akan dituangkan dalam MOC Form 2 dan 3. Project Leader kemudian dapat mengajukan form 2 dan 3 ini ke Supervisornya untuk meminta persetujuan. Kendati demikian, dapat juga dia meminta persetujuan dari orang lain dengan pertimbangan kemampuan teknis terkait dengan proyek perubahan yang dijalankan.
7. Supervisor dari Project Leader akan mempelajari Form 2 dan 3. Apabila persetujuan diberikan, maka kontrol akan dialihkan ke Lead Process Engineer (LPE). Sebaliknya jika tidak disetujui, maka Project Leader harus melakukan revisi agar mendapatkan persetujuan.
8. Lead Process Engineer (LPE) akan mempelajari Form 2 dan 3. Apabila persetujuan diberikan, maka kontrol akan dialihkan ke Chief E&C Engineer. Sebaliknya jika tidak disetujui, maka Project Leader harus melakukan revisi agar mendapatkan persetujuan.
9. Chief E&C Engineer akan mempelajari Form 2 dan 3. Apabila persetujuan diberikan, maka Project Leader diperkenankan melakukan *Implementation Hazard Screening* dan mengisi MOC Form 4. Sebaliknya jika tidak disetujui, maka Project Leader harus melakukan revisi agar mendapatkan persetujuan.

10. Project Leader mengisi MOC Form 4, kemudian diserahkan ke Supervisor dari Project Leader untuk dimintakan persetujuan.
11. Supervisor dari Project Leader akan mempelajari MOC Form 4. Apabila disetujui, maka Project Leader dapat memulai proyek perubahan. Sebaliknya jika tidak disetujui, Project Leader harus melakukan *screening* ulang.
12. Project Leader mengisi form StartUp Check, kemudian meminta persetujuan dari Superintendent.
13. Project Leader melaksanakan proyek perubahan. Selama proyek berlangsung, Project Leader diharapkan untuk memperbarui status proyek perubahan yang dijalankannya.
14. Apabila telah selesai, Project Leader akan memberitahu MOC Coordinator bahwa proyek telah selesai. Selain itu, Project Leader akan meminta persetujuan penutupan proyek dari LPE dan Chief E&C Engineer.
15. Apabila LPE dan Chief E&C Engineer sudah memberikan persetujuannya, maka proyek perubahan dapat secara resmi dinyatakan selesai.

2.5 Struktur Organisasi Perusahaan

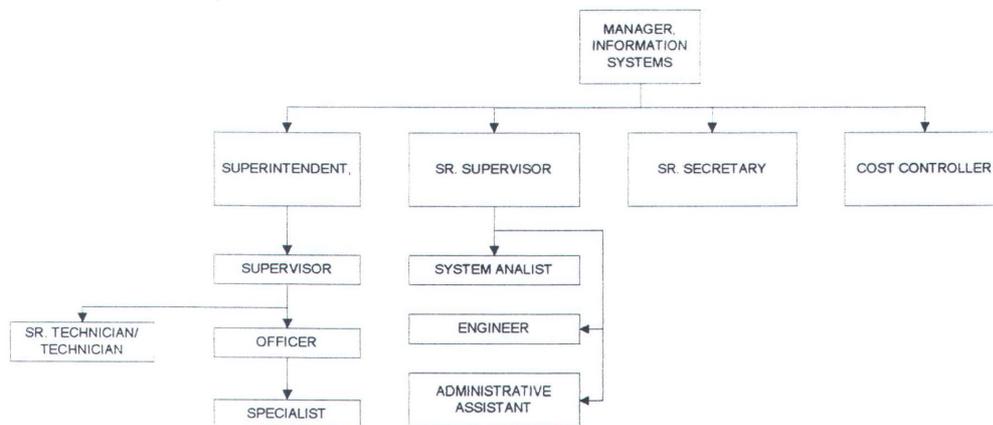
Struktur organisasi perusahaan sangatlah mempengaruhi alur dari persetujuan pada sistem ini. Dengan mengetahui hierarki jabatan yang ada, sistem akan dapat menentukan kepada siapa email ditujukan. Email ini akan memberikan peringatan kepada atasan pemohon untuk memberikan persetujuan atau penolakan terhadap permohonan yang dilakukan. Berikut ini adalah departemen-departemen yang ada di UNOCAL Indonesia Co.

Tabel 2.1 Daftar Departemen

	Departemen		Departemen
1	A/P, A/R, TAX & PAYROLL	25	LOGISTICS
2	ATTAKA TEAM	26	LOST CONTROL & SECURITY
3	COMMUNICATIONS AND GOVERNMENT RELATIONS	27	MANAGER DEEPWATER DEVELOPMENT
4	COMMUNITY RELATIONS	28	MANAGER REGIONAL, PLANNING
5	COMPLIANCE AUDITING	29	NAT. GAS & GAS LIQUIDS
6	DRILLING	30	NIB TEAM
7	ENGINEERING & CONSTRUCTION	31	NORTH AREA PRODUCTION
8	ESTATE & OPERATIONS SUPPORT	32	OMS & ESS
9	EXPLOITATION	33	OPERATIONS DEEPWATER DEVELOPMENT
10	EXPLOITATION NEW AREAS	34	PETROLEUM ENGINEERING
11	EXPLOITATION SOUTH	35	PLANNING
12	EXTERNAL AFFAIRS	36	PROCUREMENT
13	FIELD	37	PROCUREMENT & LOGISTIC
14	FINANCE	38	PRODUCTION ENGINEERING
15	G & G PROJECT	39	PROJECT DEEPWATER FACILITIES
16	GAS MARKETING	40	REVENUE PSC AND JV ACCOUNTING
17	GEOSCIENTIST	41	SECURITY & EMERGENCY RESPONSE
18	GOVERNMENT RELATIONS	42	SENIOR VP. & GENERAL
19	HUMAN RESOURCES & GENERAL AFFAIRS	43	SOUTH AREA PRODUCTION
20	HUMAN RESOURCES - EAST KALIMANTAN	44	SPECIAL PROJECT
21	HUMAN RESOURCES DEVELOPMENT	45	SYSTEMS AND SPECIAL PROJECT
22	INFORMATION SYSTEMS	46	VP. & GENERAL
23	LNG CONTRACT	47	WEST SENO CFT
24	LNG/JMG CONTRACT		

Di UNOCAL Indonesia Co, hierarki dari jabatan-jabatan diatur dalam dalam *Chain of Command* (COC) yang disusun berdasarkan `box_id` dan `parent_box_id` masing-masing pegawai. `Box_id` menunjukkan posisi jabatan pemohon. Sedangkan `parent_box_id` merujuk kepada `box_id` dari atasan langsung pemohon. Jika digambarkan, setiap departemen akan memiliki struktur organisasi yang

berbeda-beda. Sebagai contoh, berikut adalah struktur organisasi yang ada di departemen *Information System*.



Gambar 2.1 Struktur Jabatan Departemen IS

Kendati demikian, perlu diperhatikan bahwa tidak semua peran (*role*) dalam sistem e-MOC ini dapat ditentukan melalui struktur organisasi ini. Dalam prosedur MOC terdapat banyak peran yang lebih bersifat fungsional, seperti *MOC Coordinator, Lead Process Engineer, Project Leader, Chief E&C Engineer* dan sebagainya.

2.6 Sistem MOC Electronic Approval (sistem *e-MOC*)

Sistem e-MOC ini akan dibangun dengan berbasis *web* dan akan dijalankan bersama dengan sistem lain yang sejenis dalam Intranet perusahaan, yaitu di myUnocal Indonesia Portal. Terdapat beberapa kebutuhan pengguna yang perlu disediakan dan dibangun di dalam sistem MOC agar proses pengajuan MOC dapat berjalan sebagaimana harusnya. Berikut adalah penjelasan dari masing-masing kebutuhan.



2.5.1 Form

Terdapat beberapa form yang harus ada pada aplikasi ini. Form-form tersebut yaitu:

1. MOC Proposal (Form 1)

Form ini adalah form utama bagi alur proses pengajuan MOC. Setiap orang yang ingin mengajukan sebuah perubahan harus mengisi form ini terlebih dahulu. Form ini berisi nomor MOC, tanggal pengajuan, deskripsi, dan beberapa isian lainnya. Selain itu, dokumen-dokumen lain yang terkait dengan proposal MOC juga dapat dilampirkan di MOC Form 1 ini sebagai *attachment*.

2. Hazard Analysis Screening (Form 2)

Form ini dipergunakan oleh Project Leader untuk melakukan analisis risiko atau bahaya yang dapat timbul karena proses perubahan (*change*) yang diusulkan.

3. MOC Plan (Form 3)

Form ini digunakan sebagai parameter terhadap kelayakan perencanaan suatu proyek perubahan. Berisi semua hal yang seharusnya ada agar sebuah pengajuan perubahan dapat dikatakan layak untuk dilakukan. Form ini diisi oleh Project Leader dan akan dilakukan pemeriksaan oleh supervisor dari Project Leader dan Chief E&C engineer.

4. Implementation Hazard Screening (Form 4)

Form ini digunakan untuk penilaian terhadap analisa resiko yang mungkin terjadi pada saat implementasi perubahan. Hal ini diperlukan untuk sedapat mungkin meminimalisasi bahaya atau kerugian lain yang mungkin terjadi.

Form 2, 3 dan 4 merupakan form utama dalam proses pengajuan sebuah perubahan.

5. Form Persetujuan Proposal MOC (*Approval Form*)

Pada form ini para pemberi persetujuan (selanjutnya disebut *approver*) dapat memberikan persetujuan terhadap proposal MOC yang ditujukan pada dirinya.

6. Form Persetujuan terhadap hasil *screening* (*Screening Approval form*)

Form ini dipergunakan oleh *Supervisor* dari *Project Leader*, *Lead Process Engineer* dan *Chief E&C Engineer* untuk memberikan persetujuan terhadap hasil *screening* (form 2 dan 3) yang dilakukan oleh *Project Leader*.

7. Form *MOC Navigator*

Form ini digunakan untuk menentukan para pemberi persetujuan (*Approver*) bagi sebuah pengajuan MOC. Peran-peran yang berhak memberikan persetujuan antara lain *Manager* departemen, *Superintendent*, dan *Lead Process Engineer*. Dalam form ini, MOC Coordinator juga diberikan fasilitas untuk menambahkan tiga pemberi persetujuan tambahan ke alur persetujuan sebanyak maksimum tiga orang.

8. Form Persetujuan terhadap Form 4 (*Implementation Hazard Screening*)

Form ini dipergunakan oleh *Supervisor* dari *Project Leader* untuk memberikan persetujuan terhadap hasil *screening* terhadap rencana implementasi oleh *Project Leader*.

9. Form *Attachment Manager*

Form ini diperlukan untuk melakukan operasi-operasi yang berkaitan dengan *attachment* seperti menambah, menghapus atau melihat semua *attachment* untuk proposal MOC tertentu. Selain itu, *attachment manager* juga diharapkan dapat juga dipergunakan untuk keperluan lain yang serupa, yaitu mengelola *attachment* untuk form penutupan MOC (*MOC Closure*).

10. Form *Start-up Check*

Form ini berisi parameter-parameter yang digunakan oleh *Project Leader* untuk memastikan bahwa semua rencana perubahan telah siap.

11. Form *MOC Closure*

Form ini akan diisi oleh *Project Leader* untuk mengajukan persetujuan status penutupan proyek perubahan.

12. Daftar tugas (*myTask*)

Daftar ini diperlukan untuk memberikan kemudahan agar orang-orang yang terlibat dalam pengajuan MOC ini dapat mengetahui tugas-tugas yang harus diselesaikannya dalam sistem e-MOC.

13. Form Persetujuan *MOC Closure*

Form ini dipergunakan untuk memberikan persetujuan terhadap penutupan proyek perubahan yang telah dilaksanakan oleh *Project Leader*.

2.5.2 Report

Fasilitas report dalam e-MOC merupakan fitur yang bersifat pilihan. Apabila fitur ini tidak ada, maka hal ini tidak akan mengganggu jalannya proses pengajuan permohonan. Tetapi jika tersedia di dalam sistem, fitur tersebut akan mendukung proses pengajuan permohonan. Report yang ada pada sistem MOC meliputi report-report sebagai berikut:

1. Report untuk form 1, 2, 3 dan 4.

Report yang diperlukan adalah form 1, 2, 3, dan 4 dalam format HTML yang siap dicetak (*printer friendly version*). Hal ini antara lain sekedar untuk mengantisipasi budaya penggunaan kertas yang masih cukup kuat di lingkungan kerja perusahaan.

2. Grafik (*chart*) statistik MOC

Grafik ini diperlukan untuk mendeskripsikan jumlah rata-rata proposal MOC yang dapat diajukan dan dilaksanakan setiap bulannya. Fitur grafik ini termasuk fitur opsional yang tidak akan mengganggu fungsionalitas sistem apabila tidak dibuat.

2.5.3 Kebutuhan Lain

Dalam aplikasi e-MOC terdapat beberapa kebutuhan tambahan yang tidak termasuk dalam kategori form maupun report. Terdapat beberapa kebutuhan yang termasuk dalam kebutuhan lain ini, diantaranya:

1. Notification

Peringatan berupa email yang dikirimkan oleh sistem. Berisi peringatan tentang tugas-tugas yang harus dilakukan terhadap suatu permohonan MOC.

2. Reminder

Reminder dikirimkan jika notification diabaikan, atau tidak dilakukan. Reminder akan berhenti dikirim jika orang tersebut telah menjalankan tugasnya.

3. Popup User

Ini adalah popup window yang digunakan untuk mencari seseorang dan mengganti nilai *default* tertentu. Misalnya untuk mengganti *default* supervisor.

2.5.4 Aturan-aturan dalam proses bisnis

Proses bisnis disini mencakup semua proses yang terjadi di dalam sistem beserta aturan-aturan yang berpengaruh pada proses tersebut. Aturan-aturan itu antara lain:

1. Alur persetujuan

Alur persetujuan pengajuan proposal MOC adalah sesuai dengan alur yang digambarkan di bab 2 sub bab 4.

2. Segala kewenangan dan pertimbangan keputusan persetujuan sepenuhnya ada pada para *approver*.

BAB III



TEKNOLOGI J2EE DAN
MVC DESIGN PATTERN

BAB III

TEKNOLOGI J2EE DAN MVC DESIGN PATTERN

Bab ini akan membahas tentang teknologi yang akan dipergunakan dalam pengembangan sistem, yaitu teknologi **Java 2 Enterprise Edition (J2EE)** beserta beberapa Java API penting lainnya yang dipergunakan. Selain itu akan dibahas pula teknologi pendukung sistem lainnya, yaitu **Lightweight Directory Access Protocol (LDAP)**.

Aplikasi MOC ini akan menerapkan salah satu dari beberapa design pattern dalam penerapan teknologi J2EE. Design pattern adalah sekumpulan solusi yang terbukti dapat menyelesaikan permasalahan desain pada konteks tertentu [CJM-2002]. Tugas Akhir ini menggunakan **Model View Controller (MVC)** pattern dengan pendekatan *Dispatcher View* dan *Intercepting Filter* sebagai solusi yang dipilih.

3.1 Java 2 Platform, Enterprise Edition (J2EE)

Teknologi Java 2 Enterprise Edition (J2EE) diperkenalkan oleh perusahaan Sun Microsystems sebagai solusi pengembangan aplikasi berskala *enterprise*. Setelah sekian lama sejak diperkenalkan, teknologi ini telah berkembang menjadi model pengembangan aplikasi terdistribusi yang dianggap paling matang dan memiliki skalabilitas yang baik. Teknologi ini juga menawarkan berbagai fitur untuk pengembangan aplikasi yang besar, aman,

transaksional, dan berskalabilitas tinggi [RFK-2001]. Teknologi ini mendukung pengembangan aplikasi yang berbasis komponen yang dapat memaksimalkan proses pengembangan dengan penggunaan ulang (*reuse*). Sebagai salah satu teknologi yang dikembangkan dari Java, teknologi ini juga dapat dijalankan pada berbagai macam server karena bersifat tidak tergantung pada *platform* perangkat keras maupun sistem operasi tertentu.

J2EE menawarkan beberapa teknologi untuk pengembangan aplikasi, diantaranya Enterprise JavaBeans (EJB), Servlets dan Java Server Pages (JSP). Teknologi J2EE ini pada dasarnya adalah spesifikasi standar saja, sedangkan implementasinya dapat dilakukan oleh pihak manapun, sehingga tidak terbatas pada server, *platform*, *middleware*, implementasi *application* server atau perusahaan tertentu.

3.1.1 Enterprise JavaBeans (EJB)

Enterprise JavaBeans (EJB) merupakan sebuah arsitektur model komponen terdistribusi yang disediakan oleh J2EE. Teknologi ini memberikan kemudahan untuk mengembangkan komponen-komponen logik bisnis aplikasi yang bersifat aman, transaksional, berskalabilitas tinggi, dan dapat dipergunakan oleh banyak pengguna sekaligus (*multi user*). Teknologi ini memungkinkan untuk memisahkan logika aplikasi dari layanan sistem, sehingga programmer dapat lebih berkonsentrasi pada proses bisnis, sedangkan layanan sistem seperti akses ke basis data akan ditangani oleh Enterprise JavaBeans.

Secara teknis, EJB berupa sekumpulan *class-class* java yang mengikuti aturan-aturan yang telah ditetapkan dalam standar EJB, menyediakan method

pemanggilan (*finder method*) tertentu dan sebuah file konfigurasi berbentuk XML, dan dikombinasikan menjadi satu komponen [RFK-2001].

Terdapat perbedaan antara Enterprise JavaBeans (EJB) dan JavaBeans, antara lain:

- JavaBeans mendefinisikan suatu model untuk mengembangkan komponen multi guna yang biasanya dipergunakan di sisi *client*. Komponen tersebut dapat mempunyai antar muka pengguna (*user interface*), dapat juga tidak.
- Enterprise JavaBeans mendefinisikan model komponen pada sisi server (*server side*) untuk mengembangkan komponen dengan fungsi spesifik untuk logika bisnis. Komponen EJB tidak menyertakan antar muka pengguna (*user interface*) [RFK-2001].

A. Container EJB

Container EJB merupakan lingkungan eksekusi (*execution environment*) dimana komponen-komponen EJB dijalankan, yang berjalan dalam sebuah EJB server. *Container* menyediakan berbagai layanan sebagai berikut:

❖ *Component pooling* dan manajemen siklus hidup *bean*

Teknik *component pooling* ini dimaksudkan untuk melakukan pengelolaan sumber daya sistem secara efisien. Dengan teknik ini maka sejumlah kecil *instance* dari *bean* dapat melayani sejumlah besar aplikasi *client* dengan cara mengalihkan penggunaan *bean* oleh *client* tertentu yang pasif ke *client* lain yang membutuhkannya.

- ❖ Manajemen *Client Session*

Container bertanggung jawab untuk menyimpan dan mengembalikan *state* dari *bean*. Sebagaimana dibahas pada poin sebelumnya, *container* dapat memberikan sebuah *bean* ke sebuah *client* baru. Pada saat itu, *container* secara otomatis akan menyimpan *state* dari *client* yang lama. Ketika *client* lama tadi kembali memanggil *method* lain dalam *bean*, maka *state* yang disimpan tadi akan dimuat kembali dan dikopi ulang ke *bean*, sebelum *bean* tersebut diberikan kepada *client* lama.

- ❖ *Connetion pooling* basis data

Container EJB memiliki mekanisme untuk mengelola sumber daya koneksi basis data. Mekanisme ini memungkinkan banyak *client* dapat terhubung ke koneksi basis data yang terbatas, sehingga akan memberikan keuntungan berupa penghematan sumber daya.

- ❖ Manajemen transaksi (*Transaction management*)

EJB mendukung *declarative transaction*. *Declarative transaction* memungkinkan *programmer* dapat menentukan agar pemanggilan *method* EJB tertentu dilaksanakan secara transaksional. *Container* akan secara otomatis menangani proses transaksi, termasuk memulai transaksi, melakukan *commit* transaksi atau bahkan melakukan *rollback* secara otomatis, misalnya pada saat proses *recovery* setelah server mati secara tiba-tiba.

- ❖ Pengelolaan Persistensi

Container dapat juga melakukan penyimpanan *state* dari EJB dalam basis data, dengan cara menyalin isi variabel *member* dari EJB ke kolom basis data.

Container akan menyusun semua pernyataan SQL yang diperlukan untuk proses *update* basis data. *Container* juga akan memastikan integritas basis data.

B. Entity Beans

Entity Bean merupakan salah satu jenis teknologi EJB yang disediakan oleh Java 2 Enterprise Edition (J2EE). *Entity Bean* merepresentasikan data dalam basis data relasional sebagai obyek Java. Biasanya setiap *entity bean* mempunyai sebuah tabel dalam basis data relasional dan setiap *instance* dari *bean* terhubung dengan sebuah baris di dalam tabel tersebut. *Entity bean* bersifat persisten, dapat diakses oleh beberapa *client* secara konkuren (*shared access*), mempunyai *primary key* dan dapat berpartisipasi dalam relasi dengan *entity bean* yang lain. Pendekatan ini dikenal sebagai *object/relational mapping*. [RFK-2001]

Entity Bean diklasifikasikan menjadi dua tipe berdasarkan cara menangani *persistensi* data *bean* tersebut. Dalam tugas akhir ini teknologi EJB yang dipilih adalah *Entity Beans Container-Managed-Persistence* (EJB-CMP).

Dalam *Container-Managed-Persistence*, *container* bertanggung jawab untuk menangani semua akses basis data yang diperlukan oleh *entity bean*. Dalam hal ini, program *bean* tidak perlu secara eksplisit melakukan panggilan mengakses basis data. Teknologi ini juga memungkinkan pemrograman *bean* menjadi tidak tergantung pada mekanisme penyimpanan khusus. Karena kefleksibelannya ini, jika dilakukan *deploy* ulang terhadap *entity bean* yang sama, pada server J2EE dan basis data yang berbeda, maka tidak perlu memodifikasi atau melakukan kompilasi ulang terhadap *bean*.

C. Elemen Enterprise JavaBeans

Enterprise JavaBeans terdiri dari 4 (empat) elemen penting, antara lain *home interface*, *remote interface*, *bean implementation class*, dan *deployment descriptor*.

1. Home Interface

Home interface bertanggung jawab untuk mengontrol operasi-operasi yang berkaitan dengan siklus hidup *bean*, misalnya untuk mencari, menghapus atau membuat *instance* bean baru. *Home interface* memungkinkan *client* EJB dapat melakukan hal-hal sebagai berikut:

- Membuat *instance* baru dari *bean*

Pembuatan *instance* baru dari *bean* ini secara otomatis juga akan membuat satu record baru di tabel basis data.

- Menghapus *instance* *bean*

Penghapusan *instance* *bean* akan menghapus juga record yang berkaitan dalam tabel basis data.

- Mencari *instance* *entity* *bean* yang sudah ada

Home interface dari sebuah *entity bean* menyediakan satu atau lebih *method* untuk menemukan *instance* dari *entity bean* yang ada dalam *home*.

Setiap *home* dari *entity bean* mempunyai sebuah *method* `findByPrimaryKey`. *Client* yang mengetahui primary key dari sebuah objek dapat memperoleh sebuah *reference* dari objek tersebut dengan memanggil *method* `findByPrimaryKey` pada *home interface* dari sebuah

entity bean. *Bean provider*, dalam hal ini OC4J, dapat juga mendefinisikan *finder method* lain di dalam *home interface* sebuah entity bean. Hasil pencarian oleh *finder method* adalah sebuah referensi ke *remote interface*.

2. *Remote Interface* atau *Component Interface*

Remote interface dari sebuah EJB mendefinisikan *method-method* bisnis EJB yang dapat diakses oleh *client*. *Remote interface* mendefinisikan beberapa *method* sehingga *client* dapat melakukan hal-hal sebagai berikut:

- Menghapus *instance enterprise bean*
- Membaca data yang tersimpan dalam suatu *instance enterprise bean*
- Memperbarui (update) data yang tersimpan dalam suatu *instance enterprise bean*
- Memeriksa apakah suatu *instance enterprise bean* sama dengan *instance enterprise bean* lainnya
- Mendapatkan nilai *primary key* dari suatu *instance enterprise bean*

3. *Bean implementation class*

Class ini berisi semua implementasi aktual logika bisnis sebuah EJB. *Method-method* bisnis yang didefinisikan pada class enterprise bean akan secara otomatis dipanggil oleh container ketika client memanggil method terkait yang terdaftar dalam remote interface.

4. *Deployment descriptor*

Setelah membuat *home interface*, *remote interface*, dan *bean implementation class*, selanjutnya diperlukan sebuah *deployment descriptor* sebelum proses



deploy komponen EJB ke *container* dapat dilaksanakan. *Deployment descriptor* adalah sebuah file XML (harus bernama *ejb-jar.xml*) yang berisi instruksi proses *deployment* dan daftar sumber daya yang diperlukan oleh komponen-komponen EJB untuk proses *deploy* tersebut. Beberapa isi *deployment descriptor* penting antara lain:

- *<ejb-jar>*

Elemen ini merupakan elemen akar (*root*) dari sebuah *deployment descriptor* yang bersifat wajib. Semua elemen lain akan berada di dalam elemen ini.

- *<enterprise-beans>*

Elemen *<enterprise-beans>* ini berisi deklarasi semua EJB yang akan didaftarkan untuk proses *deployment* ini.

- *<ejb-name>*

Elemen *<ejb-name>* menspesifikasikan sebuah nama dari enterprise bean. Nama ini harus unik di dalam satu file *ejb-jar.xml* yang sama. Pemrograman enterprise bean tidak tergantung pada namanya, karena itu *<ejb-name>* dapat diganti selama proses pengembangan tanpa mengganggu fungsi dari enterprise bean. Namun, nama inilah yang akan dicari oleh *client* di JNDI sehingga apabila sudah ada aplikasi *client* yang menggunakannya, maka sebaiknya nama ini tidak diubah lagi.

- *<home>*

Elemen *<home>* berisi nama lengkap dari java *class* home interface sebuah enterprise bean.

- `<remote>`
 Elemen `<remote>` berisi nama lengkap dari java *class* remote interface sebuah enterprise bean.
- `<ejb-class>`
 Elemen `<ejb-class>` berisi nama lengkap dari java *class implementation bean* sebuah enterprise bean.
- `<persistence-type>`
 Elemen `<persistence-type>` menspesifikasikan sebuah tipe pengaturan penyimpanan data entity bean. Terdapat dua tipe pengaturan penyimpanan, yaitu Bean dan Container, masing-masing untuk *Bean Managed Persistence* (BMP) dan *Container Managed Persistence* (CMP). Sesuai dengan teknologi yang dipilih dalam tugas akhir, untuk selanjutnya maka nilai elemen `<persistence-type>` adalah “Container”.
- `<primary-key-class>`
 Elemen `<primary-key-class>` berisi nama *class* lengkap dari tipe data primary key sebuah entity bean. Apabila *primary key* dari table bertipe *Integer*, maka `<primary-key-class>` akan dituliskan seperti contoh berikut ini:

```
<prim-key-class>java.lang.Integer</prim-key-class>
```
- `<reentrant>`
 Elemen `<reentrant>` dapat bernilai false atau true. Jika `<reentrant>` bernilai false, maka sebuah entity bean tidak akan dapat dipanggil

ulang selama proses bisnis entity bean tersebut berlangsung. Dan elemen ini sebaiknya di set false untuk setiap entity bean.

- *<abstract-schema-name>*

Elemen ini berisi nama dari bean yang mengabstraksi table tertentu.

- *<cmp-field>* dan *<field-name>*

Elemen *<cmp-field>* memberitahukan kepada Container, *field-field* apa yang diperlukan untuk mengatur penyimpanan sebagaimana telah ditentukan di elemen anak *<field-name>*. *<field-name>* haruslah sebuah variabel *public* dari class enterprise bean.

- *<primkey-field>*

Elemen *<primkey-field>* digunakan untuk menentukan nama dari field primary key untuk sebuah entity dengan pengaturan penyimpanan Container. *<primkey-field>* harus merupakan salah satu dari field yang dideklarasikan didalam elemen *<cmp-field>* dan tipe dari field harus sama dengan tipe dari primary key. Elemen *<primkey-field>* tidak digunakan jika primary key memetakan ke multiple field. Pada kasus seperti ini, field dari primary key haruslah public dan nama dari field-field tersebut haruslah sesuai dengan nama field dari class entity bean yang mengandung key tadi.

3.1.2 Java Servlets

Servlet adalah aplikasi yang berjalan disisi *server* yang mengijinkan *business logic* dari aplikasi disisipkan di dalam proses *request-response* HTTP [RFK-2001]. Dapat juga dikatakan bahwa *Servlet* adalah sebuah komponen yang

- `<servlet-name>`

Elemen ini digunakan untuk mendaftarkan sebuah nama atau pengenal bagi servlet. Nama ini haruslah unik dalam file *web.xml*.

- `<servlet-class>`

Elemen ini digunakan untuk mendaftarkan *class* java dari servlet.

- `<servlet-mapping>`

Elemen ini digunakan untuk memetakan servlet ke URL tertentu. Sebuah servlet tidak dapat dipanggil oleh client bila belum dipetakan ke sebuah URL tertentu. Elemen ini berisi elemen anak `<servlet-name>` dan `<url-pattern>`.

- `<url-pattern>`

Elemen ini dipergunakan untuk menentukan URL yang akan dipetakan ke servlet tertentu.

3.1.3 Java Server Pages (JSP)

JavaServer Pages (JSP) menyediakan cara yang fleksibel untuk menyisipkan kode Java dalam kode HTML untuk menghasilkan halaman yang dinamis. JSP dapat berisi HTML, kode Java dan komponen-komponen JavaBean. Pada kenyataannya JSP adalah penyederhanaan dari model pemrograman servlet. Ketika seorang pengguna melakukan *request* terhadap sebuah halaman JSP, server web akan mengkompilasi halaman tersebut ke dalam sebuah servlet. Kemudian server web memanggil servlet tadi dan mengembalikan isi hasil kompilasi ke web *browser*. Sekali servlet dikompilasi dari halaman JSP, server web dapat dengan mudah menjalankan servlet tersebut tanpa perlu melakukan kompilasi ulang.

3.1.4 Servlet Filters

Teknologi *Servlet Filters* diperkenalkan pada rilis spesifikasi servlet terbaru, yaitu versi 2.3 keatas. Berbeda dengan servlet biasa, teknologi ini tidak dimaksudkan untuk membangkitkan sebuah halaman dinamis berdasarkan logika bisnis tertentu. Teknologi ini mendukung pengembangan komponen web yang dapat mengakses dan memodifikasi HTTP *request* sebelum diproses oleh *container* dan juga HTTP *response* sebelum dikirim ke *client*. Dalam Tugas Akhir ini, teknologi *servlet filter* (sering juga disebut dengan istilah *filter* saja) dipergunakan untuk mengimplementasikan komponen *Controller* pada MVC pattern.

Pada saat dijalankan, *web container* akan memanggil *servlet filter* dan menempatkannya dalam alur proses (*processing pipeline*) *request* dan *response* HTTP, misalnya pada saat terjadi pemanggilan terhadap JSP, servlet, atau file HTML. Secara konseptual, posisi *filter* dalam alur proses tersebut adalah seperti yang digambarkan dalam ilustrasi berikut ini [JTS-2002]:



Gambar 3.1 Logical view dari filter

Seperti halnya *servlet*, *filter* juga harus didaftarkan dalam *deployment descriptor* agar dapat dikenali oleh *web container*. *Filter* juga didaftarkan pada *deployment descriptor* yang sama yaitu *web.xml*. Elemen-elemen *web.xml* untuk mendaftarkan *filter* antara lain:

- `<filter>`

Elemen ini merupakan elemen induk yang berisi informasi definisi filter.

Elemen ini berisi beberapa elemen yang mendefinisikan nama filter dan nama lengkap dari class java yang merupakan implementasi dari filter ini.

- `<filter-name>`

Elemen ini mendefinisikan nama atau pengenal (*identifier*) sebuah filter.

Nama ini harus bersifat unik dalam file *web.xml*.

- `<filter-class>`

Elemen ini mendefinisikan nama *class* java lengkap yang merupakan implementasi filter ini.

- `<filter-mapping>`

Elemen ini berisi informasi pemetaan komponen filter ke URL atau pola URL tertentu. Elemen ini mempunyai elemen-elemen anak, antara lain `<filter-name>` dan `<url-mapping>`.

- `<url-pattern>`

Elemen ini mendefinisikan URL atau pola URL tertentu yang akan melalui pemrosesan filter. Setiap ada *request* HTTP ke URL yang telah didefinisikan tersebut, maka secara otomatis *container* akan menyerahkannya terlebih dahulu ke filter untuk diproses. Demikian juga untuk *response* HTTP, *container* akan melewati *response* tersebut melalui filter, sebelum dikirimkan ke *client*.

3.1.5 Application Programming Interface J2SE dan J2EE

Selain teknologi-teknologi yang telah disebutkan, Java juga menyediakan berbagai teknologi lain yang menunjang pengembangan aplikasi yang kompleks. Teknologi-teknologi tersebut dapat diakses melalui *Application Programming Interface* (API) yang disediakan oleh Java 2 Standard Edition (J2SE) maupun Java 2 Enterprise Edition (J2EE). Pada sub-bab ini akan dijelaskan beberapa API yang dipergunakan untuk pengembangan sistem.

A. Java Database Connectivity (JDBC)

JDBC merupakan pustaka java yang menyediakan akses basis data. Dengan JDBC *client* dapat membuat koneksi ke basis data, membuat *statement*, menjalankan perintah query *insert*, *delete*, atau *update*, menjalankan prosedur penyimpanan dan lain sebagainya.

B. JavaMail API

JavaMail API menyediakan satu set class abstrak dan interface untuk berinteraksi dengan sistem email. JavaMail bersifat *protocol independent*, sehingga dapat kompatibel dengan berbagai standar mail yang ada. JavaMail API mengabstraksi detail dari implementasi protokol e-mail, seperti SMTP, IMAP, POP, dan POP3 yang dipergunakan untuk mengirim dan menerima e-mail.

JavaMail menggunakan JavaBeans Activation Framework (JAF) untuk mengidentifikasi *handler* dari berbagai *content-type* e-mail.

Proses pengiriman e-mail menggunakan JavaMail terdiri dari tiga langkah yang harus dilakukan, antara lain:

1. Membuat *Session* yang merepresentasikan hubungan ke penyedia layanan e-mail. Sebuah *Session* menangani konfigurasi dan otentifikasi ke sebuah layanan e-mail.
2. Membuat dan menginisialisasi objek *Message* yang berisi data yang akan dikirimkan melalui e-mail. Sebuah objek *Message* merepresentasikan e-mail tunggal, yang termasuk didalamnya *body text* pesan, subjek email, alamat tujuan dan sebagainya.
3. Mengirimkan pesan e-mail yang sudah disiapkan.

Dalam lingkup Tugas Akhir ini, JavaMail dipergunakan untuk mengirimkan pemberitahuan (*notification*) dan pengingat (*reminder*).

C. Java Naming and Directory Interface (JNDI)

Layanan penamaan dan direktori sudah menjadi hal yang umum dalam dunia komputasi. Layanan ini disediakan untuk mempermudah pencarian sebuah informasi, bahkan sebuah objek yang kompleks. Dengan layanan ini, informasi tentang sumberdaya seperti data login pengguna, koneksi ke basis data, atau bahkan lokasi printer di jaringan, didaftarkan di sebuah layanan direktori yang terpusat. *Client* yang membutuhkannya dapat memperoleh informasi tersebut dari server menggunakan protokol tertentu. Diantara beberapa layanan direktori yang ada, sistem e-MOC akan menggunakan LDAP untuk layanan otentifikasinya.

LDAP adalah layanan standar direktori yang diadopsi oleh banyak pihak. LDAP memiliki banyak implementasi dalam dunia industri, diantaranya

OpenLDAP, Novell Directory Services (NDS) dan Microsoft Active Directory (AD).

Java Naming and Directory Interface (JNDI) menyediakan antarmuka terhadap layanan penamaan dan direktori. JNDI juga menyediakan sekumpulan API dengan *method-method* untuk menjalankan operasi *standard directory*, seperti mengasosiasikan atribut dengan objek dan mencari objek menggunakan atribut-atributnya. Dengan menggunakan JNDI, sebuah aplikasi dapat menyimpan dan mengambil tipe dari penamaan Java object. JNDI tidak bergantung pada implementasi layanan spesifik, namun bersifat mengabstraksi layanan direktori yang sudah ada selain menyediakan layanan direktori sendiri. Aplikasi Java dapat menggunakan JNDI untuk mengakses beberapa layanan direktori, termasuk penamaan yang ada dan layanan direktori seperti LDAP, NDS, DNS, dan NIS.

Dalam Tugas Akhir ini, JNDI dipergunakan untuk:

- Untuk mendapatkan *Home Interface* dari Enterprise JavaBeans CMP Container J2EE meletakkan referensi terhadap Home interface dari Entity Beans di JNDI. Sebuah EJB *client* dapat memperoleh referensi tersebut dengan mencarinya di JNDI.
- Untuk mendapatkan koneksi ke basis data
OC4J merekomendasikan agar semua basis data yang akan dipergunakan dalam aplikasi didefinisikan secara global dalam *data-sources.xml*. Semua data-source tersebut nantinya akan didaftarkan ke JNDI oleh OC4J, sehingga untuk mendapatkannya harus dengan memanggil layanan API dari JNDI. Metode ini memberi keuntungan besar dalam pengembangan aplikasi, karena

perubahan definisi informasi basis data untuk keseluruhan aplikasi dapat dilakukan hanya dengan mengubah di satu tempat saja.

- Untuk melakukan otentifikasi ke LDAP server

Layanan LDAP berfungsi sebagai penyedia layanan otentifikasi sentral di Unocal Indonesia, sehingga setiap aplikasi di perusahaan dikonfigurasi untuk melakukan pengecekan data penggunaanya di LDAP server tersebut. Layanan LDAP dapat diakses dari Java menggunakan antarmuka yang disediakan oleh JNDI.

3.2 Lightweight Directory Access Protocol (LDAP)

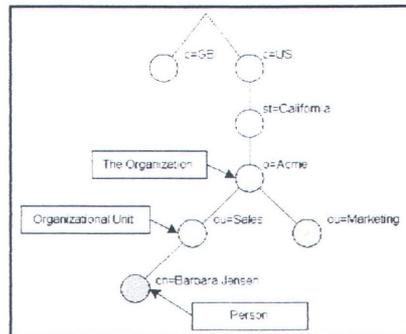
LDAP dikembangkan oleh Universitas Michigan sebagai adaptasi dari layanan direktori yang kompleks yang disebut dengan standar X.500 Directory Access Protocol (DAP). Standar X.500 dianggap terlalu berat untuk digunakan di internet, sehingga dikembangkanlah *Lightweight Directory Access Protocol* (LDAP) yang lebih ringan.

Standar LDAP telah diadopsi oleh banyak pihak, antara lain oleh Microsoft dalam produknya Active Directory (AD), dan Novell dalam produknya Novell Directory Services (NDS).

Informasi dalam LDAP disusun dalam bentuk pohon hierarki (*hierachical tree*). Ada dua model penyusunan hierarki yang dapat dipergunakan, yaitu model tradisional dan model internet [OPL-2003].

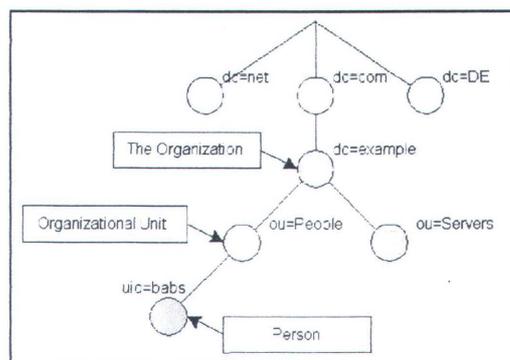
Model hierarki tradisional disusun dengan berdasarkan lokasi geografis. Puncak hierarki adalah negara, kemudian pada tingkat dibawahnya adalah negara bagian atau provinsi, disusul dengan nama organisasi, sub organisasi, lalu diikuti

dengan berbagai macam entri lain yang ada pada sub organisasi tersebut. Entri-entri tersebut dapat berupa user, dokumen, dan lain-lain.



Gambar 3.2 Model hierarki tradisional

Selain itu, hierarki LDAP dapat disusun dengan berdasarkan domain internet. Dengan model ini, puncak hierarki bermula dari *top-level domain* (misalnya com, edu, dan sebagainya), diikuti dengan nama domain, lalu nama sub domain, kemudian diikuti dengan entri lain yang ada dibawahnya, seperti identitas pengguna (*user ID*) dan sebagainya.



Gambar 3.3 Model hierarki internet

Dalam Tugas Akhir ini, LDAP model hierarki internet dipergunakan untuk melakukan otentifikasi terpusat sesuai dengan ketentuan perusahaan. Dengan menggunakan layanan otentifikasi terpusat ini maka akses terhadap berbagai sumberdaya informasi seperti Domain Logon (Windows), Oracle Portal, Livelink,

dan semua aplikasi intranet yang berbasis web dapat dilakukan dengan *username* dan *password* yang sama.

3.3 Model-View-Controller (MVC) Design Pattern

Design pattern mula-mula dikenal di bidang teknik sipil dan arsitektur. Pada era 70-an, teknik arsitektur dianggap sebagai salah satu disiplin ilmu yang memerlukan banyak pengalaman. Pandangan tersebut mulai berubah sejak diterbitkannya beberapa buku oleh *Christopher Alexander et.al.*, dan menjadikan bidang ilmu ini dapat diterapkan oleh orang-orang yang belum memiliki pengalaman yang luas. Mereka mengidentifikasi kemiripan-kemiripan dari berbagai arsitektur yang terbukti dapat memecahkan berbagai permasalahan dalam konteks yang sama. Kemiripan-kemiripan tersebut kemudian disimpulkan menjadi prinsip-prinsip umum yang dapat dipergunakan untuk menyelesaikan permasalahan desain secara umum. Solusi umum ini kemudian disebut sebagai "*pattern*" dalam bidang arsitektur.

Evolusi *pattern* dalam disiplin ilmu arsitektur ini memberikan pengaruh yang besar terhadap berbagai disiplin ilmu lainnya. Jelas bahwa setiap disiplin ilmu yang matang harus memiliki dokumentasi solusi untuk permasalahan-permasalahan umum. Demikian juga pada bidang rekayasa perangkat lunak, dimana *pattern* telah terbukti sukses dalam mendeskripsikan solusi untuk permasalahan umum perangkat lunak. Sub bab ini akan menjelaskan beberapa *pattern* dalam bidang rekayasa perangkat lunak yang diterapkan dalam pengembangan sistem e-MOC ini.

3.3.1 Definisi Pattern

Pattern secara sederhana dapat didefinisikan sebagai sebuah solusi yang telah terbukti dapat menyelesaikan sebuah permasalahan dalam konteks tertentu [CJM-2002]. Lebih jauh, Alexander menyatakan bahwa:

“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution of that problem in such way that you can use this solution a million times over, without doing it the same way twice.”

Pattern disusun berdasarkan pengalaman dan berdasarkan solusi yang sudah terbukti dapat menyelesaikan permasalahan. *Pattern* dapat dipergunakan ulang untuk menyelesaikan berbagai skenario permasalahan dengan konteks yang sama. Penggunaan *pattern* membantu mempercepat proses desain perangkat lunak dengan menggunakan ulang solusi-solusi yang sudah terbukti berhasil.

3.3.2 Komponen Arsitektur MVC

Sesuai dengan namanya, *pattern* ini terdiri dari tiga komponen. Berikut adalah penjelasan dari tiap-tiap komponen MVC:

A. Model

Model merepresentasikan logika bisnis dari sebuah aplikasi. Enkapsulasi aturan bisnis ke dalam komponen mempermudah penelusuran *bug*, memungkinkan pelaksanaan testing secara terpisah dari aplikasi keseluruhan dan mendukung pemakaian kembali komponen yang sudah ada. *Model* dalam praktiknya dapat diimplementasikan dengan teknologi JavaBeans atau Enterprise

Java Beans, atau keduanya. *Model* dalam *MVC pattern* diharapkan dapat mengembalikan data olahan, sehingga sedapat mungkin data tersebut dapat langsung ditampilkan oleh layer *View* tanpa pemrosesan lebih lanjut. *Model* juga diharapkan dapat membantu proses penyimpanan data ke basis data dengan cara yang lebih sederhana.

B. View

View merupakan modul sistem yang mewakili fungsi *presentation logic* dari sebuah aplikasi. Komponen *View* mendapatkan data yang diperlukan dari *Model* dan menyediakan antar muka pengguna untuk protokol khusus yang terlibat. Sebagai bagian dari proses menampilkan antar muka pengguna yang dinamis, *View* bertanggung jawab untuk merespon serangkaian *event* yang dapat pengguna lakukan pada setiap saat. *View* dapat diimplementasikan menggunakan teknologi JSP ataupun Servlet. Pemisahan *View* dengan *Model* memungkinkan untuk dapat mengembangkan antar muka pengguna yang bervariasi dengan mudah.

C. Controller

Pada arsitektur MVC, *controller* bertanggung jawab untuk menerima *event*, menentukan *handler* yang tepat sekaligus memanggil *handler* tersebut dan akhirnya mengembalikan *response* yang sesuai dengan *event* yang ditangkap. *Controller* juga dapat berfungsi untuk memeriksa validitas parameter yang dikirim dan memeriksa hak akses pengguna terhadap operasi tertentu.

Controller memiliki beberapa peran penting dalam arsitektur MVC, yaitu:

- **Fungsi keamanan (security) terpusat**

Controller bertanggung jawab untuk melakukan pemeriksaan keamanan melalui mekanisme otentifikasi dan otorisasi. *Controller* mula-mula memeriksa status *login* pengguna dengan cara memeriksa *session*. Apabila ternyata *session login* pengguna tidak valid, maka *Controller* akan menampilkan halaman login.

Selain itu, *Controller* juga bertanggung jawab melakukan kontrol akses. Pengguna mempunyai hak akses yang berbeda-beda terhadap sistem. *Controller* akan memeriksa otorisasi seorang pengguna terhadap fitur yang akan diaksesnya dan akan menolaknya apabila pengguna tersebut tidak memiliki hak akses (*access rights*).

- **Identifikasi event**

Controller bertanggung jawab untuk menangkap event yang dikirim oleh pengguna, mengidentifikasinya, kemudian menentukan cara penanganan dari event tersebut.

- **Validasi data**

Apabila diinginkan, *Controller* dapat juga didesain untuk dapat melakukan validasi data yang disertakan pada setiap event yang dikirim. *Controller* dapat memeriksa apakah parameter yang dikirimkan sudah lengkap, sudah sesuai formatnya, dan sebagainya.

3.4 Pattern pendukung lain yang digunakan

Selain MVC sebagai *pattern* inti dalam perancangan sistem e-MOC ini, terdapat beberapa *pattern* lain yang dipergunakan sebagai pendukung.

3.4.1 Single Access Point Pattern

Single Access Point pattern merupakan salah satu *pattern* yang berkaitan dengan penanganan masalah keamanan. *Pattern* ini mendeskripsikan bahwa sebuah sistem harus hanya mempunyai satu jalan masuk (*point of entry*). Jalan akses masuk ini harus dapat menangani semua *request* yang masuk. Dengan mengimplementasikan *pattern* ini, diharapkan bahwa semua pengguna telah melakukan proses otentifikasi dasar dan tidak dapat melewati begitu saja (*bypass*) semua pemeriksaan identitas [CJM-2002].

Pada tugas akhir ini, *Single Access Point* pattern diterapkan bersama-sama dengan *Intercepting Filter* pattern untuk mengembangkan *controller* pada arsitektur MVC yang telah dijelaskan sebelumnya. Berbeda dengan implementasi *controller* yang umum ditemui dimana *controller* dipetakan ke sekelompok URL semu, implementasi *controller* dengan menggunakan kedua *pattern* tersebut dapat dipetakan ke file JSP secara langsung. Model implementasi ini diharapkan lebih memperkecil kemungkinan akan adanya akses yang tidak melalui proses otorisasi.

3.4.2 Intercepting Filter

Intercepting Filter pattern memungkinkan aplikasi web untuk menangkap *request* yang dikirimkan oleh *client* sebelum diproses oleh target yang bersangkutan [CJM-2002]. *Pattern* ini biasanya dipergunakan untuk melakukan

validasi terhadap data atau parameter yang dikirimkan oleh *client* dan pada Tugas Akhir ini, dipergunakan bersama-sama dengan *Single Access Point* pattern untuk mengembangkan *Controller*. Secara teknis, *pattern* ini dapat diimplementasikan dengan menggunakan teknologi *Filter* yang disediakan J2EE.

3.4.3 Service Locator Pattern

Dalam aplikasi yang dikembangkan menggunakan teknologi J2EE, peranan Java Naming and Directory Interface (JNDI) cukup dominan. JNDI banyak digunakan antara lain untuk mencari referensi ke Enterprise JavaBeans (EJB) maupun untuk mendapatkan koneksi ke basis data. Namun demikian, akses ke JNDI ini dapat cukup membebani sistem apabila dilakukan berulang-ulang.

Apabila sebuah program memerlukan layanan yang diperoleh melalui JNDI, maka sebelumnya harus disusun dulu parameter yang dibutuhkan oleh JNDI, membuat koneksi ke *server* JNDI, lalu kemudian baru melakukan operasi pembacaan (*look up*) [CJM-2002].

Service Locator Pattern mengabstraksi semua panggilan JNDI yang rumit tersebut menjadi sebuah *method* yang sederhana. Selain itu, *Service Locator pattern* juga meminimalisasi akses JNDI dengan cara meletakkan semua objek yang pernah diakses sebelumnya dalam sebuah “*cache*”.

Service Locator diimplementasikan sebagai sebuah class yang bersifat *singleton instance*, yaitu hanya ada satu *instance* dari class ini di memory yang berjalan untuk melayani semua program pemanggil. Tujuan dari hal ini antara lain agar *cache* dapat berjalan dengan efektif.



**PERANCANGAN dan PEMBUATAN
SISTEM e- MOC**

BAB IV

Bab IV

PERANCANGAN dan PEMBUATAN

SISTEM e-MOC

Bab ini menjelaskan tentang elemen-elemen dalam sistem MOC dan tahapan-tahapan perancangan serta implementasi perangkat lunak sistem MOC. Meliputi penjelasan perancangan desain *pattern* yang digunakan, perancangan data, perancangan proses dan perancangan antarmuka sistem. Perancangan yang telah dibuat kemudian diimplementasikan ke dalam pemrograman agar sistem MOC dapat berjalan seperti yang seharusnya.

4.1 Elemen Sistem MOC

Elemen-elemen sistem ini adalah hal-hal yang akan terlibat atau menjadi bagian dari sistem MOC. Terdiri dari grup pengguna yang akan terlibat, jalur persetujuan yang perlu dilalui dan status pengajuan. Grup pengguna ini dibagi berdasarkan peran-peran yang akan mereka wakili. Berikut ini adalah daftar dari peran-peran yang terlibat, beserta penjelasannya masing-masing.

Tabel 4.1. Daftar Peran dalam Sistem MOC

Peran	Keterangan
Originator	Peran ini berhak untuk mengajukan proposal MOC baru. Terdiri dari semua karyawan di Unocal Indonesia Co.
Supervisor of Originator	Atasan langsung dari Originator.

	Fungsi ini dijabat oleh tiga orang, yaitu satu orang MOC Coordinator E&C yang bertindak sebagai koordinator tertinggi dan dua orang Area Coordinator.
Project Leader	Bertanggung jawab untuk melaksanakan <i>Hazard Analysis</i> , <i>Change Plan</i> dan <i>Implementation Hazard Screening</i> , kemudian melaksanakan implementasi perubahan.
Supv of Project Leader	Atasan langsung dari Project Leader.
Chief E&C Engineer	Kepala Engineer departemen E&C. Bertanggung jawab untuk melakukan verifikasi <i>MOC Plan</i> dan <i>Hazard Analysis Results</i> dan menyetujui penutupan proyek perubahan.

Sedangkan jalur-jalur persetujuan yang harus dilalui oleh sebuah permohonan MOC adalah sebagai berikut:

Tabel 4.2. Jalur Persetujuan

Rute	Keterangan
Originator -> Supervisor of Originator	<i>Originator</i> membuat proposal MOC (MOC Form 1), kemudian diserahkan kepada penyeliaanya (<i>supervisor</i>) untuk mendapatkan persetujuan.
<i>Supervisor</i> -> <i>MOC Coordinator</i>	<i>Supervisor</i> dari <i>Originator</i> dapat menyetujui atau menolak proposal MOC yang diajukan. Apabila proposal disetujui, proposal akan dialihkan ke <i>MOC Coordinator</i> . Selain itu, Supervisor juga berhak mencalonkan seorang <i>Project Leader</i> untuk MOC yang diajukan. Apabila ditolak, maka proposal akan dibatalkan.
MOC Coordinator -> {Superintendent, Manager, Lead Process Engineer,	MOC Coordinator akan mengirimkan proposal MOC ke <i>Superintendent</i> , <i>Manager</i> , dan <i>Lead Process Engineer</i> untuk meminta persetujuan proposal MOC. <i>MOC Coordinator</i> juga dapat menentukan tiga <i>approver</i>

Other Approvers}	tambahan sesuai keperluan. Masing-masing dapat juga mencalonkan seorang <i>Project Leader</i> .
{Superintendent, Manager, Lead Process Engineer, Other Approvers}-> MOC Coordinator	Proposal MOC yang disetujui akan dikembalikan kepada <i>MOC Coordinator</i> , sedangkan yang ditolak akan dibatalkan.
<i>MOC Coordinator</i> -> <i>Project Leader</i>	<i>MOC Coordinator</i> kemudian menentukan <i>Project Leader</i> . <i>MOC Coordinator</i> dapat menunjuk orang lain diluar calon yang diajukan oleh para <i>approver</i> .
Project Leader-> Supervisor of Project Leader	<i>Project Leader</i> melakukan <i>Hazard Analysis Screening</i> (MOC Form 2) dan menyusun rencana proyek perubahan (MOC Plan - Form 3). Hasilnya kemudian diserahkan kepada penyeliannya. Alur persetujuan MOC baru dapat dilanjutkan setelah mendapatkan persetujuan dari <i>Supervisor of Project Leader</i> .
Supervisor of Project Leader -> Lead Process Engineer (LPE)	Hasil <i>Hazard Analysis</i> dan <i>MOC Plan</i> yang telah disetujui kemudian diserahkan kepada <i>Lead Process Engineer (LPE)</i> untuk dimintakan persetujuan. Alur persetujuan MOC baru dapat dilanjutkan setelah mendapatkan persetujuan dari <i>LPE</i> .
LPE -> Project Leader	Project Leader diminta mengisi <i>Implementation Hazards Screening</i> (MOC Form 4).
Project Leader-> Supv of Project Leader	MOC Form 4 kemudian diajukan kepada penyeliannya untuk mendapatkan persetujuan.
Supv of Project Leader-> Project Leader	Project Leader memulai implementasi proyek perubahan (<i>change project</i>).
Project Leader-> MOC Coordinator, LPE, Chief E&C Engineer	Project Leader menyatakan proyek selesai dan siap ditutup.
MOC Coordinator->	MOC Coordinator (dalam hal ini akan ditangani secara

{LPE, Chief E&C Engineer}	otomatis) meminta LPE dan Chief E&C Engineer memeriksa status proyek sebelum ditutup. Proyek MOC ditutup setelah keduanya memberikan persetujuan penutupan.
------------------------------	--

Ketika sebuah permohonan MOC diajukan dan diproses, setiap tahapan yang telah dilalui oleh sebuah permohonan, memiliki sebuah status. Setiap status yang ada memiliki maksud dan arti yang berbeda-beda. Berikut ini adalah status yang terdapat dalam sistem MOC.

Tabel 4.3. Status MOC

Status	Keterangan
Proposed	Proposal MOC sudah diajukan dan sedang menunggu persetujuan <i>Supervisor</i> dari Originator.
Supervisor of Originator Approved	Proposal MOC disetujui oleh <i>Supervisor of Originator</i> dan sedang menunggu MOC Coordinator melakukan <i>forward</i> proposal ke para pemberi persetujuan (<i>approver</i>).
Waiting Management Approval	MOC Coordinator sudah melakukan <i>forward</i> proposal ke <i>Superintendent, Manager, LPE</i> dan para <i>approver</i> tambahan. Apabila ada satu saja dari para <i>Approver</i> ini tidak menyetujui, maka status MOC secara otomatis menjadi <i>Rejected</i> .
Management Approved	Manajemen sudah menyetujui proposal MOC, kemudian proposal dikembalikan ke MOC Coordinator. Proses MOC menunggu MOC Coordinator untuk memilih Project Leader
Rejected	Proposal MOC ditolak oleh salah satu <i>approver</i> , yaitu <i>Supervisor of Originator, Superintendent, Manager, LPE</i> dan para <i>approver</i> tambahan. Proposal yang sudah ditolak akan dianggap gugur.

Screening	Project Leader sedang melakukan proses <i>screening</i> dengan mengisi MOC Form 2 dan 3.
SPL Approved screening	Supervisor dari Project Leader menyetujui hasil <i>screening</i> , kemudian dikirimkan ke LPE. Proses MOC menunggu persetujuan hasil <i>screening</i> dari LPE.
Implementation Hazard screening	LPE menyetujui hasil <i>screening</i> . Project Leader melaksanakan <i>Implementation Hazards screening</i> (MOC Form 4).
Change Project Ongoing	<i>Supervisor of Project Leader</i> menyetujui MOC Form 4. <i>Project Leader</i> melaksanakan proyek perubahan.
PL Initiate Closure	Project Leader menyatakan proyek perubahan siap ditutup. Proses MOC menunggu LPE dan Chief E&C Engineer melakukan persetujuan penutupan proyek.
Closed	Proyek MOC dinyatakan selesai dan ditutup.

4.2 Perancangan Perangkat Lunak

Pada bagian perancangan perangkat lunak ini untuk pemodelan sistem digunakan Rational Rose versi 2002, sedangkan untuk pemodelan data digunakan Power Designer versi 9. Diagram-diagram yang digunakan dalam proses perancangan ini antara lain, Entity Relational Database (ERD) berupa Physical Data Model, use case diagram, class diagram, dan sequence diagram.

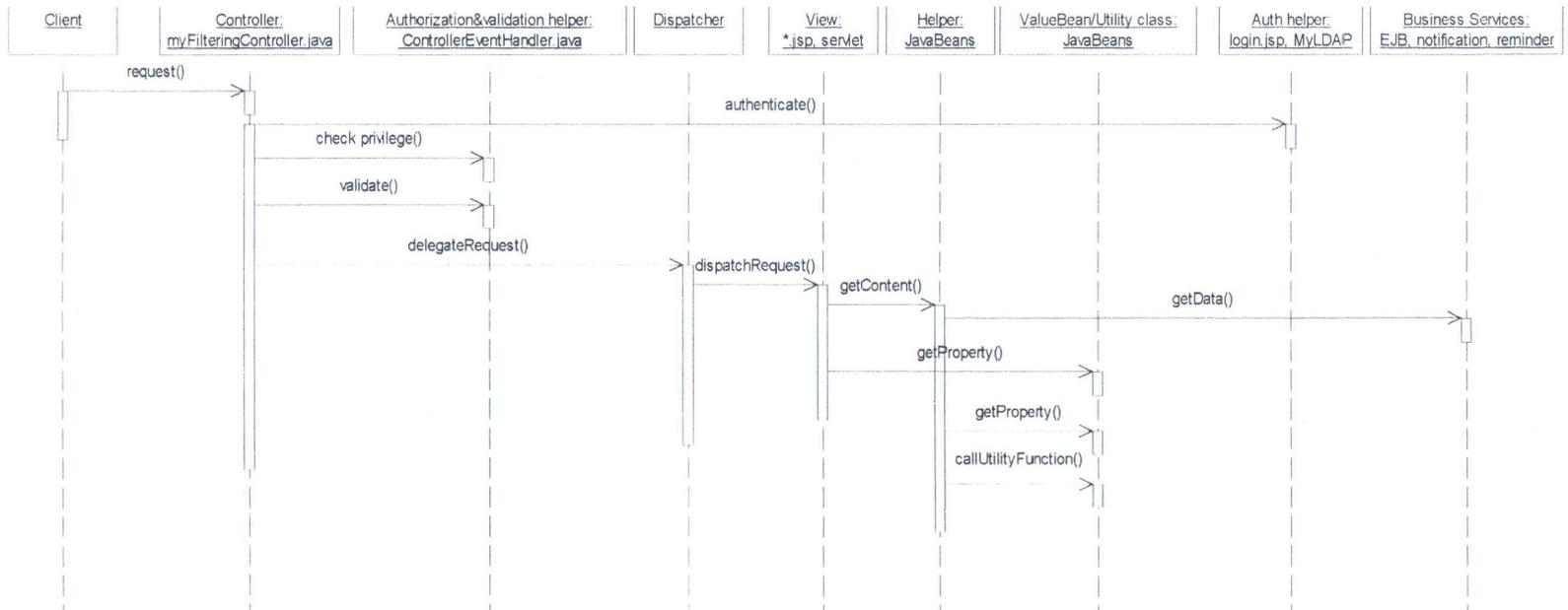
4.2.1 Perancangan Aplikasi dengan MVC Pattern

UNOCAL Indonesia Company memenuhi sebagian kebutuhan perangkat lunaknya dengan cara mengembangkannya sendiri. Departemen yang menangani hal ini adalah departemen Information System, khususnya bagian Applications Support. Karena pengembangannya ditujukan bagi kepentingan intern perusahaan,

segala kebutuhan sistem juga disesuaikan dengan kebutuhan pengguna di perusahaan. Pengguna memiliki keleluasaan untuk menentukan apa-apa yang mereka butuhkan, bahkan terkadang desain yang telah disepakati sebelumnya masih dapat diubah-ubah sesuai dengan keinginan pengguna. Akibatnya sering terjadi proses pengembangan perangkat lunak menjadi proses yang tidak kunjung selesai.

Dengan memperhatikan kondisi tersebut, perancangan perangkat lunak harus juga memberikan aspek fleksibilitas pengembangan dan dapat memberikan kemudahan kepada para pengembang untuk mengubah sistem sesuai permintaan pengguna tanpa harus melakukan perubahan terhadap rancangan dasar sistem. Sebab itu maka perlu untuk memilih *pattern* yang tepat agar menghasilkan rancangan aplikasi cukup fleksibel untuk memenuhi permintaan pengguna yang cenderung berubah-ubah. Dengan mempertimbangkan hal tersebut, maka dipilihlah desain *pattern* Model-View-Controller (MVC) *pattern* dengan pendekatan *Dispatcher View* sebagai solusinya.

Berdasarkan penjelasan pada bab tiga tentang MVC *pattern*, rancangan perangkat lunak e-MOC dalam bentuk diagram *sequence* adalah sebagai berikut [CJM-2002]:



Gambar 4.1 Dispatcher View diagram sequence e-MOC

Berikut ini penjelasan dari masing-masing partisipan dari pattern MVC:

A. Controller

Controller merupakan titik awal dalam menangani permintaan dari *client*. *Controller* bertanggung jawab untuk memeriksa status login, hak akses, melakukan validasi parameter, dan memutuskan bagaimana menangani *event* dari permintaan tersebut. Dalam aplikasi e-MOC, *Controller* dibangun dengan menggunakan teknologi *filter* yang dipetakan ke pola URL tertentu. Dengan penggunaan teknologi *filter* maka dapat menjamin bahwa semua permintaan (*request*) yang masuk akan secara otomatis diproses oleh *Controller* terlebih dahulu.

Pada perancangan aplikasi e-MOC, *Controller* dipetakan ke semua file .JSP dalam file *web.xml*, seperti contoh berikut ini:

```

    <filter>
        <filter-name>myFilteringController</filter-name>
        <filter-class>
com.unocal.moc.servlet.myFilteringController2
        </filter-class>
    </filter>
    <filter-mapping>
        <filter-name>myFilteringController</filter-name>
        <url-pattern>
/for/*.jsp
        </url-pattern>
    </filter-mapping>

```

Dengan demikian maka akses ke semua file .JSP akan diproses terlebih dahulu oleh filter (yang berfungsi sebagai *Controller*) sebelum file .JSP tersebut dijalankan.

Controller kemudian melakukan *parsing* terhadap URL untuk mendapatkan *event* yang dimaksud oleh pengguna. Dalam *Controller* ini juga,

setiap session dari pengguna akan dikenali. Jika seorang pengguna berusaha mengakses sistem e-MOC tanpa login terlebih dahulu, yang berarti tidak mempunyai session, maka Controller akan mengarahkan tampilan langsung ke halaman login.

Jika session telah dikenali, maka Controller akan memuat secara dinamis Authorization & Validation Helper berdasarkan *event* yang diterima tersebut. Kemudian Controller menentukan *event handler* dari permintaan yang diterima untuk kemudian menyerahkan kontrol ke Dispatcher.

B. Authorization & Validation Helpers

Sesuai namanya, Authorization & Validation Helpers terdiri dari beberapa java *class* yang berfungsi untuk membantu Controller dalam melakukan pemeriksaan otorisasi pengguna terhadap halaman web yang akan diakses dan sekaligus juga memeriksa *request* parameter yang dikirimkan. Controller dapat memanggil *helper* ini secara dinamis berdasarkan sebuah file konfigurasi (*ControllerEventHandlers.properties*).

Penggunaan helper semacam ini memudahkan pengembang dalam membuat komponen pada *presentation layer* karena tidak perlu memasukkan kode untuk pengecekan hak akses serta validasi parameter di setiap halaman web.

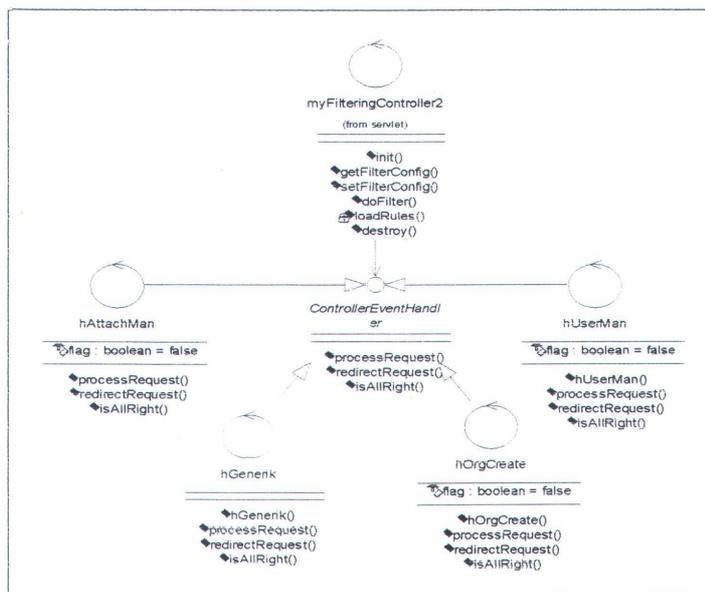
C. Dispatcher

Dispatcher mempunyai fungsi untuk mengelola *view* dan navigasi. *Dispatcher* bertugas untuk mengalihkan pengguna ke *view* berikutnya, sesuai dengan keputusan Controller tentang bagaimana menangani *event* yang diterima tersebut.

Setelah Controller mengenali session dari pengguna, memeriksa hak akses dan memeriksa keabsahan parameter *request*, maka Controller akan meminta dispatcher untuk memanggil *view* yang diminta.

Dalam tugas akhir ini, dispatcher dikembangkan menjadi satu dengan Controller dengan memanfaatkan teknologi *filtering* yang dipergunakan untuk mengembangkan Controller. Dengan teknologi filtering ini maka tugas Dispatcher dapat dialihkan ke *Web Container* dari J2EE Application Server (OC4J).

Berikut ini adalah diagram *class* dari Controller, Authorization helpers dan Dispatcher:



Gambar 4.3 Diagram *class* dari Controller

Penjelasan dari masing-masing *class* adalah sebagai berikut:

❖ **myFilteringController2.java**

`myFilteringController2.java` merupakan implementasi inti dari *Controller* yang dibangun menggunakan teknologi *servlet filters*. *Controller* ini akan menggunakan bean-bean turunan dari *ControllerEventHandler.java* untuk membantu menangani *event* yang diterimanya. Bean-bean tersebut akan dimuat pada saat inisialisasi *Controller*, kemudian diletakkan di sebuah *HashMap* yang berfungsi sebagai *cache*, sehingga tidak perlu memuatnya secara berulang-ulang nantinya.

`myFilteringController2.java` dapat mengenali bean *event handler* baru tanpa memerlukan kompilasi ulang. Hal ini karena bean-bean tersebut dimuat secara dinamis berdasarkan sebuah file konfigurasi yang mudah diubah-ubah.

❖ **ControllerEventHandler.java**

Merupakan sebuah *class* abstrak yang merupakan *class* induk bagi semua bean *event handler* yang berfungsi sebagai pembantu *Controller* untuk menangani otorisasi dan validasi. *Class* ini diperlukan agar *Controller* dapat dengan mudah memuat bean-bean tersebut secara dinamis. Karena fungsinya, maka bean-bean tersebut disebut sebagai *Authorization & Validation helper* bagi *Controller*.

❖ **hAttachMan.java**

Bean ini merupakan turunan dari *class* abstrak *Controller-EventHandler.java*. Fungsi dari bean ini adalah menjadi *helper* bagi *Controller* untuk melakukan validasi akses terhadap *pop-up Attachment Manager* (*Popup_AttachManager.jsp*).

• **hGenerik.java**

Bean ini merupakan turunan dari *class* abstrak *Controller-EventHandler.java*. Fungsi dari bean ini adalah menjadi *helper* generik bagi *Controller* untuk melakukan validasi akses semu terhadap halaman-halaman JSP yang sebenarnya tidak memerlukan pemeriksaan, misalnya untuk halaman login.

Bean-bean *helper* lainnya mempunyai fungsi yang serupa, sebab itu tidak perlu dijelaskan lebih lanjut.

D. View

View akan merepresentasikan dan menampilkan informasi ke *client*. View adalah semua file .JSP dan servlet yang digunakan sebagai antarmuka sistem e-MOC.

E. View Helper (Helper)

Helper ini diwakili oleh bean-bean yang akan membantu View atau *Controller* dalam menyelesaikan proses bisnis, dimana nilai yang diperoleh dari

bean dapat langsung ditampilkan ke halaman web. Selain itu, helper juga dapat membantu *view* menangani penyimpanan data. Contoh dari view helper adalah `helperMocRecord.java` yang berfungsi menangani proses penampilan data proposal MOC sekaligus juga menangani proses penyimpanan data proposal tersebut.

F. Value Bean dan Utility classes

`ValueBean` dan `Utility` adalah jenis lain dari *helper* dimana nilai yang diperoleh dari bean tidak dapat langsung ditampilkan ke halaman web. Bean yang termasuk dalam tipe ini misalnya, bean untuk memperoleh nomor proposal MOC yaitu `MocSequence.java`.

G. Authentication Helper

`Auth Helper` berfungsi untuk menangani layanan otentikasi. Fungsi otentikasi dilaksanakan oleh bean yang akan mengenali apakah seorang pengguna diperbolehkan untuk mengakses sistem e-MOC. Proses ini dilaksanakan dengan cara mengirimkan permintaan otentikasi ke server LDAP yang berfungsi sebagai penyedia layanan login terpusat, baik bagi sistem e-MOC, aplikasi berbasis web lainnya, maupun akses ke sistem operasi (Windows).

H. Business Services (Layanan bisnis)

Layanan bisnis ini merupakan komponen dari `Dispatcher View pattern` yang bertugas memberikan layanan bisnis terhadap proses. Termasuk didalamnya

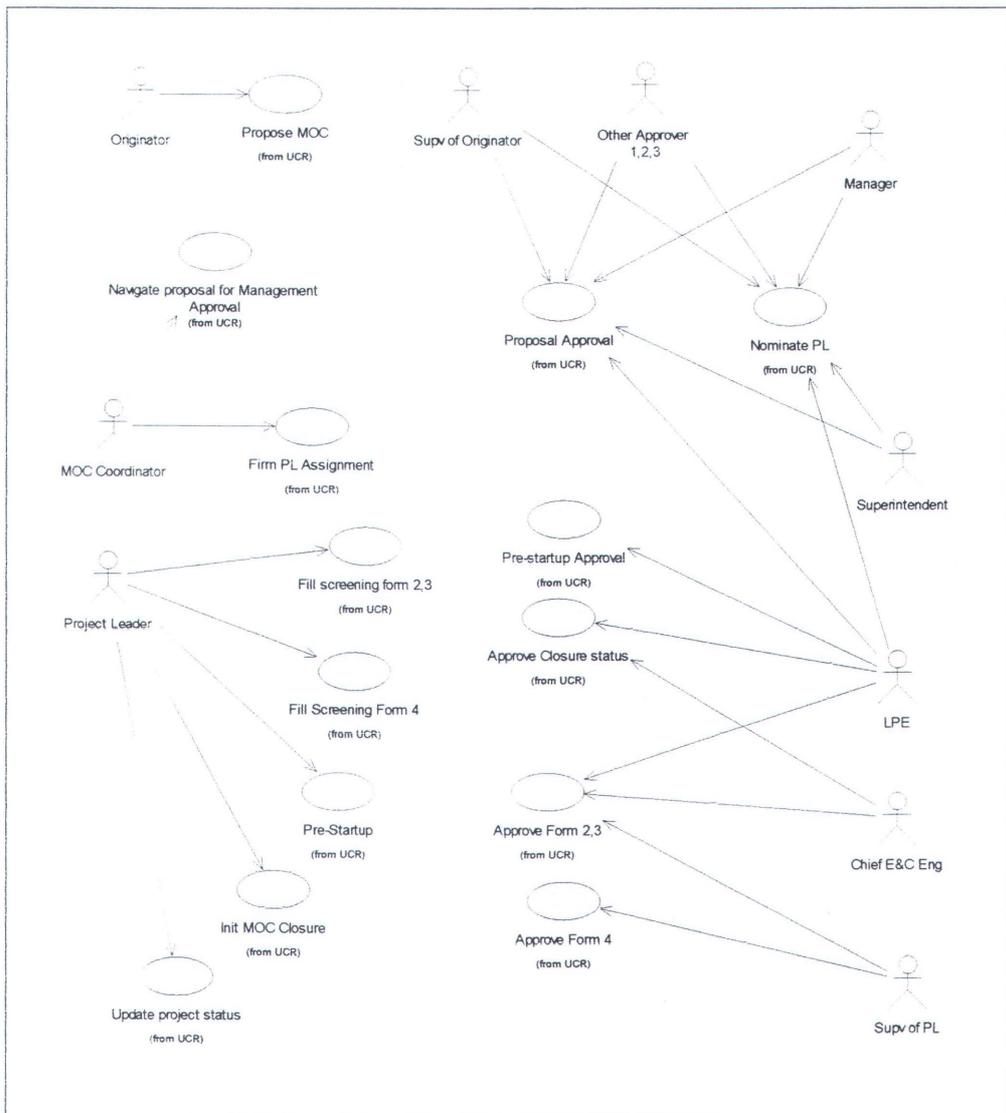
yaitu pengelolaan alur persetujuan (*approval flow*), layanan untuk pengiriman *notification* dan *reminder*, serta semua EJB sebagai abstraksi dari sumber data.

4.2.2 Use Case Diagram

Sebelum membahas perancangan sistem yang lain, akan dibahas terlebih dahulu tentang diagram use case yang ada. Dari diagram ini akan terlihat transaksi-transaksi apa yang dapat dilakukan oleh setiap aktor yang terlibat.

Pada diagram use-case ini dapat dilihat bahwa terdapat beberapa aktor yang mempunyai tugas atau transaksi yang sama. Kesamaan tugas ini dapat dipergunakan sebagai pedoman pada saat implementasi sehingga proses pengembangan dapat menjadi efisien.

Diagram use case selengkapnya dari sistem e-MOC digambarkan sebagai berikut.

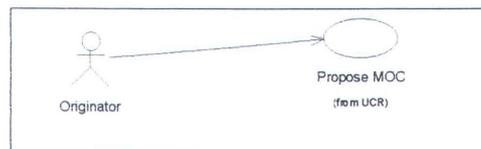


Gambar 4.3 Diagram Use Case sistem e-MOC

Penjelasan yang lebih mendalam tentang masing-masing use case dan aktor yang terlibat adalah sebagai berikut.

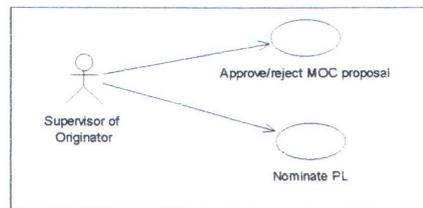
1. Use Case Diagram: Originator

Seorang Originator hanya dapat mengajukan permohonan MOC. Setelah proposalnya dikirimkan kepada *Supervisor of Originator*, Originator tidak bisa mengubahnya lagi



Gambar 4.4 Originator

2. Use Case Diagram: Supervisor of Originator



Gambar 4.5 Supervisor of Originator

Pada gambar 4.5 dapat dilihat bahwa Supervisor dari Originator mempunyai dua use case sebagai berikut:

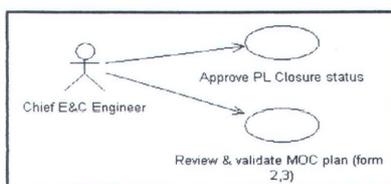
- ❖ Approve or Reject MOC proposal

Seorang Supervisor dari Originator dapat memberikan persetujuan atau penolakan terhadap proposal MOC.

❖ Nominate a Project Leader

Supervisor dari Originator berhak mencalonkan seorang Project Leader. Penentuan Project Leader itu sendiri merupakan tanggung jawab MOC Coordinator.

3. Use Case Diagram: Chief E&C Engineer

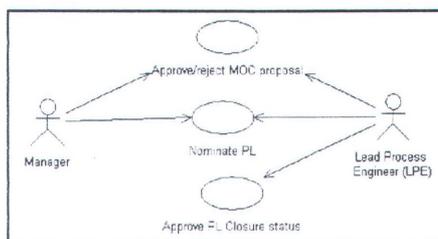


Gambar 4.6 Chief E&C Engineer

Chief E&C Engineer dapat melakukan persetujuan atau penolakan terhadap hasil *screening* yang dibuat oleh Project Leader. Selain itu juga berhak memberikan persetujuan terhadap penutupan proyek perubahan yang diajukan oleh Project Leader.

4. Use Case Diagram: Manager

Manager adalah salah satu dari beberapa Approver dalam MOC. Approver ini dapat memberikan persetujuan atau penolakan terhadap MOC yang diajukan. Selain itu Manager juga dapat mengajukan calon Project Leader dari MOC ini.



Gambar 4.7. Manager & Lead Process Engineer

5. Use Case Diagram: Lead Process Engineer (LPE)

LPE juga dapat melakukan apa yang dilakukan Manager. Selain itu LPE juga dapat memberikan persetujuan atas status *closure* yang diajukan oleh Project Leader. Gambar use case diagram dari LPE dapat dilihat pada gambar 4.7.

6. Use Case Diagram: Project Leader (PL)

Project Leader dapat melakukan hal-hal sebagai berikut:

- ❖ Do screening

Project Leader bertanggung jawab untuk melakukan analisis sebelum proyek perubahan benar-benar dilaksanakan.

- ❖ Fill form 2,3 based on screening

Project Leader mengisi MOC *screening* form 2 dan 3 berdasarkan hasil analisis yang dilakukannya. Hasil screening ini nantinya diajukan kepada *Supervisornya* untuk meminta persetujuan.

- ❖ Conduct *implementation hazard screening* (MOC form 4)

Project Leader juga bertanggung jawab untuk melaksanakan analisis terhadap rencana pelaksanaan proyek perubahan. Hasil analisis ini dituangkan dalam MOC form 4.

- ❖ Do the Change project dan Update project status

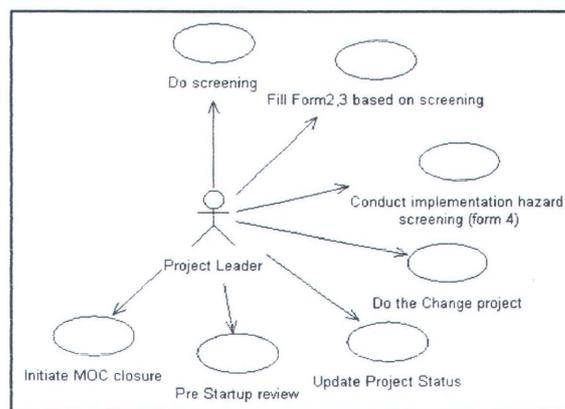
Project Leader bertugas memimpin pelaksanaan proyek perubahan. Selama proyek perubahan berlangsung, Project Leader akan melakukan *update* terhadap status proyek (*project status*).

❖ Pre start-up review

Sebelum proyek perubahan benar-benar dimulai, Project Leader harus mengisi *checklist* yang disebut “pre startup check”.

❖ Initiate MOC closure

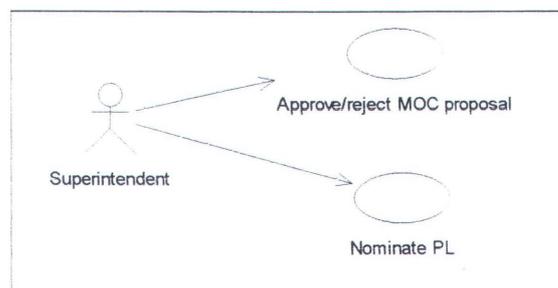
Setelah proyek perubahan selesai dilaksanakan, Project Leader akan mengajukan penutupan proyek perubahan.



Gambar 4.8. Project Leader

7. Use Case Diagram: Superintendent

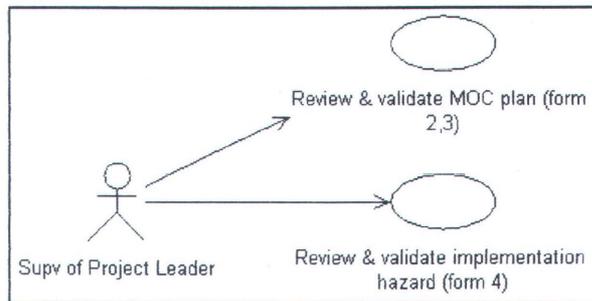
Superintendent dapat memberikan persetujuannya atas sebuah permohonan MOC dan mengajukan calon PL bagi permohonan tersebut.



Gambar 4.9 Superintendent

8. Use Case Diagram: Supervisor of PL

Supervisor dari PL dapat memberikan review dan melakukan validasi terhadap rencana MOC (form 2 dan 3) dan *implementation hazard* (form 4).



Gambar 4.10. Supervisor of PL

9. Use Case Diagram: MOC Coordinator

MOC Coordinator sering juga disebut Navigator. Seorang MOC Coordinator dapat melakukan hal-hal sebagai berikut:

- ❖ Navigate MOC proposal

Melakukan forward proposal MOC to *Superintendent, Manager, LPE* dan mungkin juga ke orang-orang lain (*Other approvers*).

- ❖ Assign Project Leader

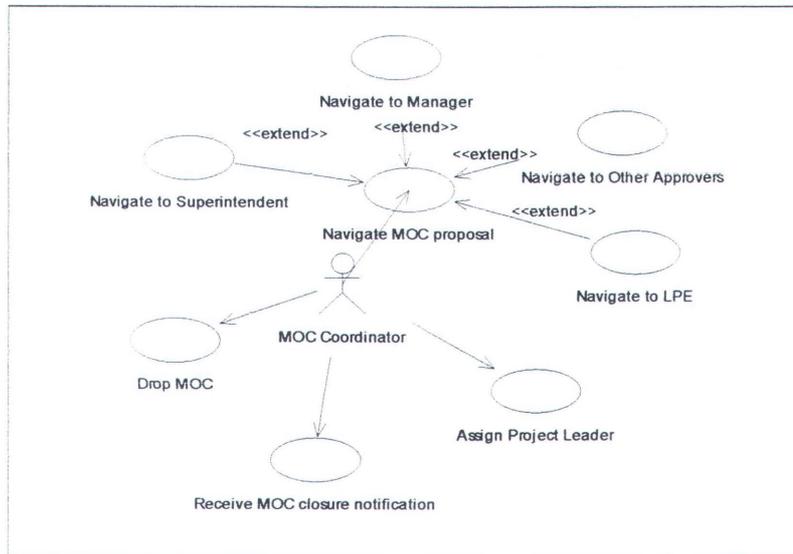
Menentukan Project Leader. Project Leader dapat dipilih baik dari calon-calon yang diajukan para *approver* maupun orang lain.

- ❖ Receive Notification of MOC Closure status

MOC Coordinator akan menerima pemberitahuan apabila Project Leader mengajukan penutupan proyek perubahan.

❖ Drop MOC

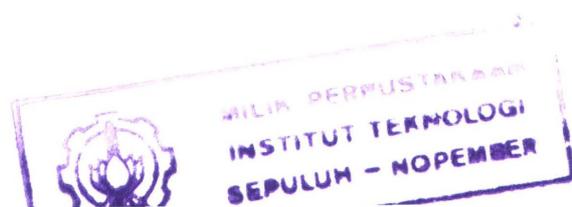
MOC Coordinator dapat menghapus MOC yang sudah berjalan secara paksa apabila ternyata Project Leader lalai tidak mengajukan penutupan proyek, walaupun sebenarnya proyek sudah selesai.



Gambar 4.11. MOC Coordinator

4.2.3 Perancangan Data

Dari perancangan data ini dapat diketahui tabel-tabel apa saja yang dipergunakan untuk menyimpan data-data dari sistem MOC dan keterkaitan antara tabel satu dengan tabel lainnya. Perancangan data ini menggunakan Physical Data Model (PDM). Hasil pembangkitan kode dari PDM ini akan digunakan untuk membuat tabel-tabel dalam Oracle Database Server. Tabel-tabel ini sendiri ditempatkan dalam skema *production*. Untuk mengetahui lebih jelas tentang rancangan data yang dibuat, dapat dilihat pada gambar di bawah ini.



Dari gambar diatas terlihat tabel-tabel yang akan digunakan untuk menyimpan data-data yang dipergunakan dalam sistem MOC.

1. MOC2_MST_AREA

Tabel ini merupakan tabel master dari Area. Pada tabel ini menyimpan data area beserta MOC Koordinator dari masing-masing area.

Tabel 4.4 Master Area

MOC2_MST_AREA		
Nama Atribut	Tipe data	Keterangan
area_id	Integer	ID dari area kerja
area	Varchar2	Nama area kerja
moc_coord	Char2	Badge dari MOC Coordinator untuk suatu area kerja

2. MOC2_MST_FIELD

Tabel ini merupakan tabel master field. Tabel ini akan menyimpan data field beserta pada area mana field tersebut berada.

Tabel 4.5 Master Field

MOC2_MST_FIELD		
Nama Atribut	Tipe data	Keterangan
area_id	Integer	ID dari area kerja
field_id	Varchar2	ID dari field
field	Varchar2	Nama field pada area tertentu

3. MOC2_SCREENINGFORMENTRIES

Tabel ini digunakan untuk menyimpan isi setiap item pilihan dari form 2, form 3 dan form 4, untuk setiap permohonan MOC.

Tabel 4.6 Screening Form Entries

MOC2_SCREENINGFORMENTRIES		
Nama Attribut	Tipe data	Keterangan
item_ans	Varchar2	Jawaban dari form 2,3, 4 dan startup
item_id	Integer	ID dari item
moc_id	Varchar2	Nomer permohonan MOC

4. MOC2_RECORD

Tabel ini merupakan tabel yang paling penting diantara yang lainnya. Dalam tabel ini tersimpan hampir semua informasi penting tentang MOC yang diajukan. Hampir semua data masukan dari form 1 disimpan dalam tabel ini.

Tabel 4.7 Record

MOC2_RECORD		
Nama Attribut	Tipe data	Keterangan
moc_id	Varchar2	Nomer permohonan MOC
status_id	Char	ID dari status
area_id	Integer	ID dari area kerja
field_id	Varchar2	ID dari field
date_propose	Date	Tanggal pengajuan permohonan MOC
afe_cc	Varchar2	Kode AFE/ <i>cost control</i> untuk proposal MOC yang diajukan.
workorder	Varchar2	Kode <i>work order</i> /perintah kerja terkait dengan MOC yang diajukan
change_desc	Varchar2	Deskripsi singkat dari perubahan yang

		diusulkan
typeofchange	Varchar2	Tipe perubahan yang diusulkan
temp_week	Varchar2	Jangka waktu perubahan (khusus yg bertipe <i>temporary</i>), satuan mingguan
department	Varchar2	Departemen dari Originator.
aims	Varchar2	Nomer AIMS record untuk moc proposal ini, jika ada
pl_project_status	Varchar2	status dari kemajuan proyek perubahan, diperbarui oleh <i>Project Leader</i> selama proyek berlangsung
org_position	Varchar2	Posisi/jabatan dari Originator
facility	Varchar2	Nama fasilitas dimana perubahan akan dilakukan
budgeted	Varchar2	Flag ada tidaknya dana untuk perubahan yang diusulkan

5. MOC2_STATUS

Tabel ini merupakan tabel master yang menyimpan data status dari MOC. Status ini diperlukan terutama untuk masalah penelusuran status persetujuan MOC.

Tabel 4.8 Master Status

MOC2_STATUS		
Nama Atribut	Tipe data	Keterangan
status_id	Char	ID dari status
status_desc	Varchar2	Deskripsi dari status

6. MOC2_ATTACHMENT

Tabel ini digunakan untuk menyimpan file-file attachment yang dikirimkan pada saat pengajuan permohonan MOC dan penutupan MOC. File-file ini dapat dalam berbagai macam format. Di dalam tabel ini file-file tersebut akan disimpan dengan tipe BLOB.

Tabel 4.9 Attachment

MOC2_ATTACHMENT		
Nama Attribut	Tipe data	Keterangan
attachment	BLOB	File <i>attachment</i> dalam format BLOB
attach_id	Number	ID dari <i>attachment</i>
moc_id	Varchar2	Nomer permohonan MOC
filename	Varchar2	Nama file <i>attachment</i>
content_type	Varchar2	Tipe dari file <i>attachment</i>
sessionid	Varchar2	ID dari session
forwhat	Number	Flag tujuan penggunaan attachment ini.

7. MOC2_NOMINATED_PL

Tabel ini digunakan untuk menyimpan data pencalonan Project Leader oleh para Approver. Project Leader yang terpilih akan memiliki flag yang berbeda dengan yang lain.

Tabel 4.10 Nominated Project Leader

MOC2_NOMINATED_PL		
Nama Attribut	Tipe data	Keterangan
moc_id	Varchar2	Nomor permohonan MOC
person_id	Number	Nomor ID di tabel MOC2_PERSON

		dari orang yang dicalonkan sebagai Project Leader
moc_person_id	Number	Nomor ID di tabel MOC2_PERSON dari orang yang mencalonkan Project Leader
flag_selected	Integer	Flag untuk menandai bahwa orang ini terpilih sebagai Project Leader

8. MOC2_APPROVAL

Tabel ini berfungsi sebagai tabel yang menyimpan data approval yang dilakukan setiap Approver dan sekaligus sebagai log bagi proses pengajuan MOC. Data yang tersimpan dalam tabel ini meliputi data rencana persetujuan, yaitu siapa saja orang-orang yang akan memberikan persetujuan pada permohonan ini dan data persetujuan yang sudah dilakukan.

Tabel 4.11 Approval

MOC2_APPROVAL		
Nama Atribut	Tipe data	Keterangan
approval_level	Integer	Level dari <i>approval</i>
date_approval	Date	Tanggal dilakukannya proses approval
host	Varchar2	Nama komputer atau alamat IP dari orang yang melakukan <i>approval</i>
forward_to	Varchar2	Mencatat pengalihan penanganan MOC
log_id	Number	ID dari catatan log yang terkait
moc_id	Varchar2	Nomor permohonan MOC
person_id	Number	Nomor kode identitas orang yang melakukan aktivitas <i>approval</i>
remark_id	Number	ID dari remark (catatan yang disertakan)

		dalam persetujuan)
AREA_ID	Number	ID dari tindakan (action) yang diambil
role_id	Integer	ID dari role orang yang melakukan aktivitas <i>approval</i>

9. MOC2_ACTION

Tabel ini merupakan tabel master yang menyimpan data tindakan-tindakan yang dapat dilakukan oleh setiap orang yang terlibat dalam proses permohonan MOC.

Tabel 4.12 Master Action

MOC2_ACTION		
Nama Atribut	Tipe data	Keterangan
AREA_ID	Char	ID dari action
action_desc	Varchar2	Deskripsi dari action

10. MOC2_REMARKS

Tabel ini menyimpan catatan-catatan atau komentar dari para Approver terhadap setiap permohonan MOC.

Tabel 4.13 Remarks

MOC2_REMARKS		
Nama Atribut	Tipe data	Keterangan
remark_id	Number	ID dari <i>remark</i>
remark	Varchar2	Komentar atau catatan dari setiap <i>approver</i>

11. MOC2_PERSONS

Tabel ini digunakan untuk menyimpan data pegawai yang terlibat dalam sistem MOC. Data dalam tabel ini digunakan sebagai rujukan untuk tabel-tabel lain, seperti pada tabel MOC2_NOMINATED_PL untuk mengetahui siapa yang menjadi calon PL dan MOC_CLOSURE_APPL untuk mengetahui orang yang melakukan closure. Sistem e-MOC tidak menggunakan nomor registrasi karyawan atau *badge* sebagai identitas (ID) pegawai karena pada kenyataannya di lapangan sering terjadi satu orang memiliki lebih dari satu nomor *badge*, bahkan dimungkinkan juga adanya satu nomor *badge* dimiliki oleh lebih dari satu orang.

Tabel 4.14 Persons

MOC2_PERSONS		
Nama Atribut	Tipe data	Keterangan
person_userid	Varchar2	Nama login atau ID pengguna seperti pada LDAP.
person_badge	Varchar2	Nomor registrasi pegawai
person_flag	Varchar2	Flag untuk menandai bahwa pengguna ini berstatus <i>disabled</i> dalam e-MOC.
person_id	Number	Nomor ID karyawan yang bersifat unik dalam e-MOC
email	Varchar2	Alamat e-mail dari karyawan
position	Varchar2	Jabatan karyawan

12. MOC2_ROLES

Tabel ini tabel master untuk menyimpan id-id untuk setiap group peran (*role*) yang terlibat dalam sistem MOC.

Tabel 4.15 Master Roles

MOC2_ROLES		
Nama Attribut	Tipe data	Keterangan
role_id	Integer	ID dari role
role_name	Varchar2	Nama role yang ada

13. MOC2_REMINDERLIST

Tabel ini berisi daftar orang-orang yang akan dikirim pengingat (*reminder*).

Tabel 4.16 Reminder List

MOC2_REMINDERLIST		
Nama Attribut	Tipe data	Keterangan
remind_count	Number	Jumlah pengingat (<i>reminder</i>) yang sudah dikirimkan ke pengguna ini
remind_id	Number	Kode ID dari isian reminder
moc_id	Varchar2	Kode ID dari proposal MOC yang terkait dengan pengingat ini
log_id	Number	Kode ID log di tabel MOC2_APPROVAL
remind_limit	Number	Jumlah maksimal pengingat yang akan dikirimkan. Apabila <i>remind_count</i> sama atau melebihi nilai ini, maka pengingat tidak akan dikirimkan lagi.
remind_subject	Varchar2	Subjek e-mail pengingat yang akan dikirimkan
remind_extranotes	Varchar2	Isi e-mail pengingat yang akan dikirimkan
flag-removed	Varchar2	Flag penanda bahwa pengingat ini sudah dihapus.

		Kendati demikian, catatan pengiriman pengingat ini tetap disimpan untuk keperluan di masa mendatang.
startdate	Date	Tanggal mulai pengiriman pengingat
usermail	Varchar2	Alamat e-mail dari pengguna yang akan menerima kiriman pengingat.

4.2.5 Perancangan Proses (Realisasi Use-Case)

Perancangan proses perangkat lunak ini merupakan realisasi dari diagram use-case, yang digambarkan dengan Diagram Sequence dan Diagram Class. Berikut ini merupakan penjelasan tentang perancangan proses yang terdapat dalam aplikasi e-MOC.

A. Pengajuan proposal MOC (MOC Form 1)

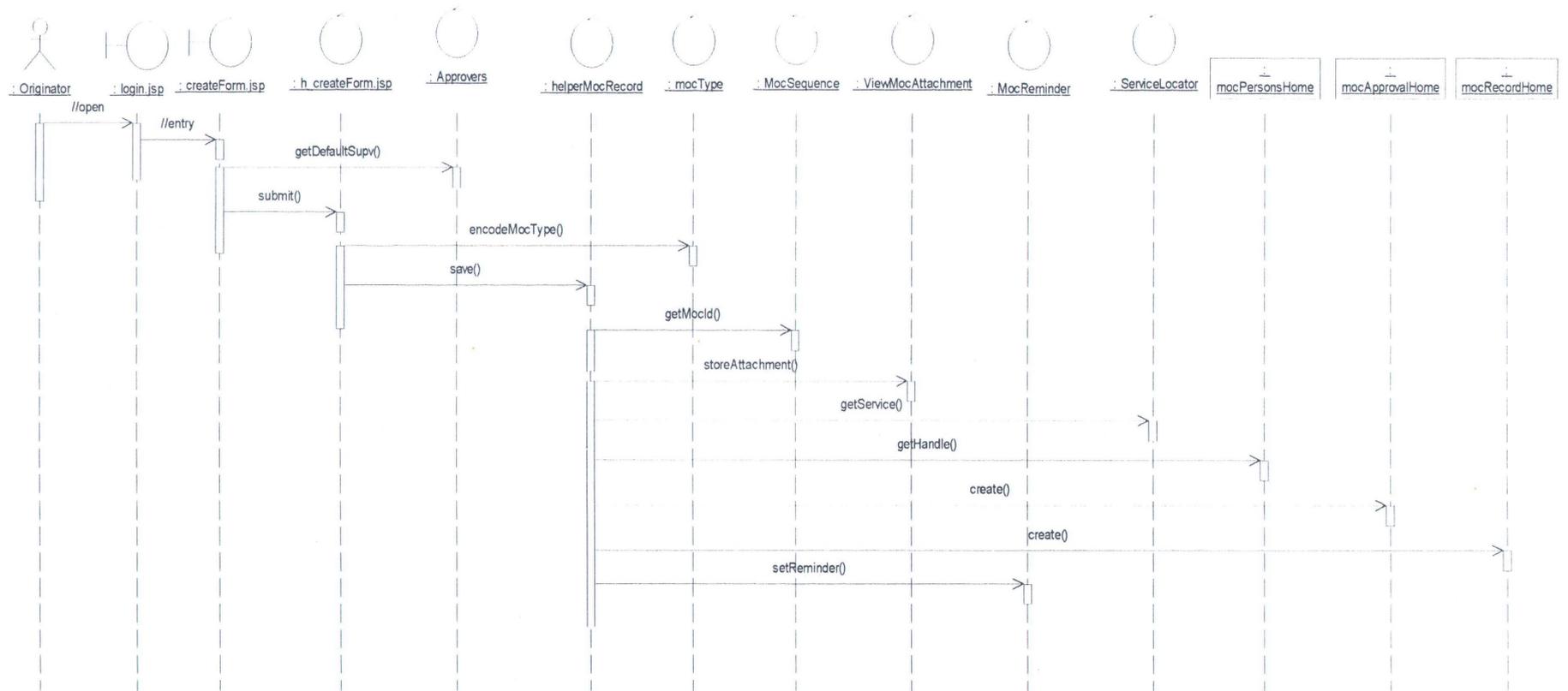
Pada proses pengajuan proposal MOC, aktor yang berperan adalah Originator. Proses dimulai saat Originator membuka halaman login, kemudian mengisi formulir pengajuan proposal MOC (MOC Form 1). Originator kemudian mengisi semua kolom isian yang diperlukan, dan menambahkan berkas yang akan dilampirkan dalam bentuk *attachment*.

Setelah Originator menekan tombol *Submit*, maka kemudian proses beralih pada *h_createForm.jsp*. Pada saat *script* JSP ini dipanggil, sistem akan menjalankan beberapa bean dan EJB lain, antara lain:

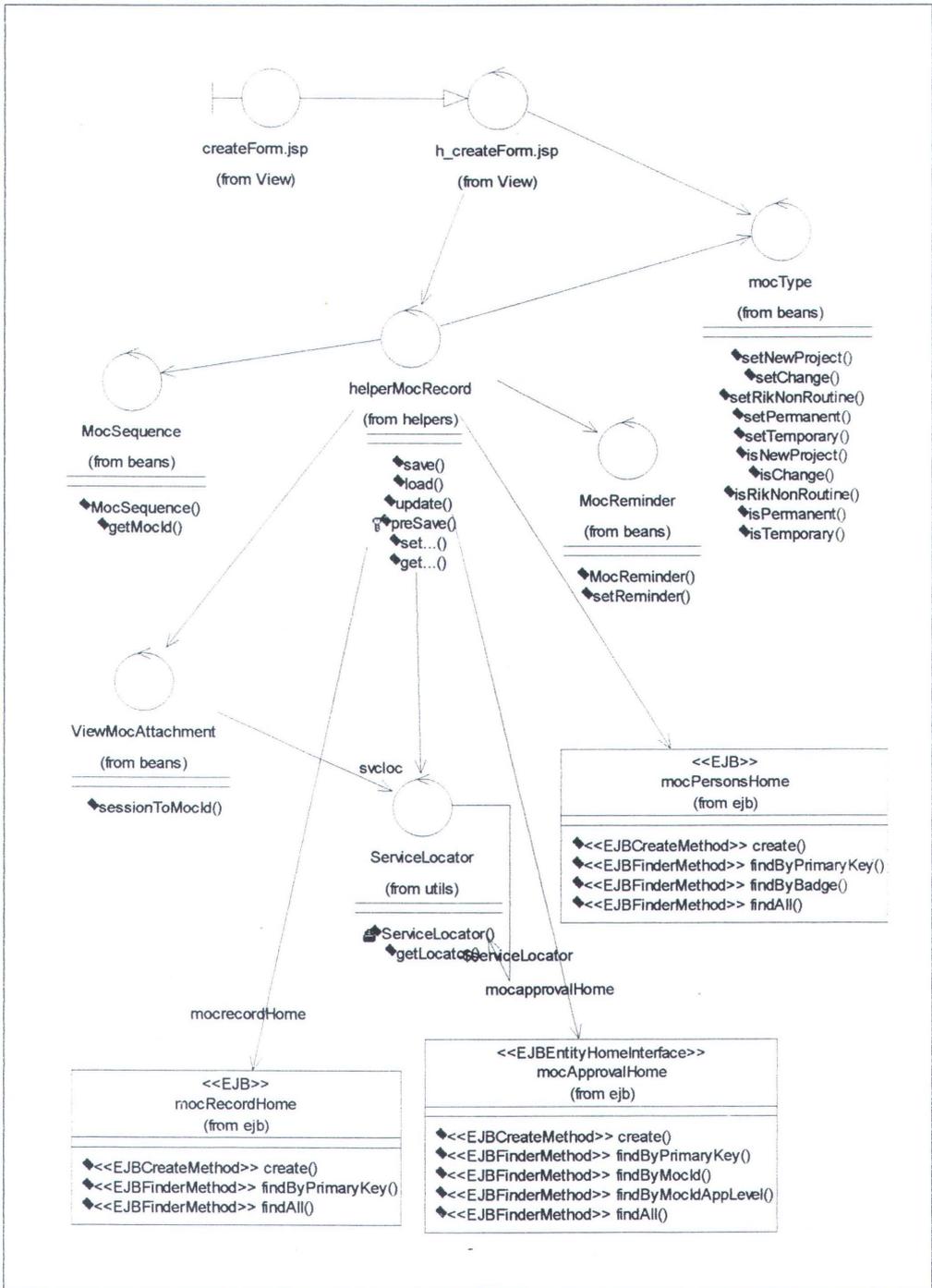
- *Approvers.java* dipergunakan untuk memperoleh *default* Supervisor dari Originator.

- *mocType.java* dipergunakan untuk membantu melakukan penyimpanan informasi kode tipe MOC. Bean ini diperlukan karena tipe MOC dapat lebih dari satu, sedangkan pada perancangan tabel basis data sendiri hanya satu kolom, sehingga diperlukan proses pengkodean terlebih dahulu.
- *helperMocRecord.java* merupakan sebuah bean yang berfungsi untuk membantu menyimpan, memuat dan memperbarui isi MOC Form 1.
- *MocSequence.java* digunakan oleh *helperMocRecord.java* untuk mendapatkan kode ID untuk proposal MOC yang akan dibuat.
- *ViewMocAttachment.java* berfungsi untuk menyimpan file-file yang dilampirkan dalam MOC. Selain itu, bean ini juga dapat mengembalikan URL atau daftar URL dari semua file yang terkait dengan nomor ID MOC tertentu.
- *MocReminder.java* berfungsi untuk menyiapkan pengingat (*reminder*) pada Supervisor dari Originator untuk mengisi formulir persetujuan proposal MOC.
- *ServiceLocator.java* berfungsi untuk membantu dalam mencari referensi ke *home interface* EJB di JNDI. Setelah referensi yang dimaksud didapatkan, maka *helperMocRecord.java* dapat memanggil EJB yang dimaksud.
- *mocRecordHome.java* merupakan *home interface* dari EJB *mocRecord*. EJB ini berfungsi untuk mengelola data MOC Form 1 di tabel MOC2_RECORD.
- *mocPersonsHome.java* merupakan *home interface* dari EJB *mocPerson*. EJB ini berfungsi untuk menangani data karyawan dalam sistem e-MOC di tabel MOC2_PERSONS.

- *mocApprovalHome.java* merupakan *home interface* dari EJB *mocApproval*. EJB ini dipergunakan untuk menangani data persetujuan MOC di tabel `MOC2_APPROVAL`.



Gambar 4.12 Diagram Sequence Pengajuan MOC



Gambar 4.13 Diagram Class Pengajuan MOC

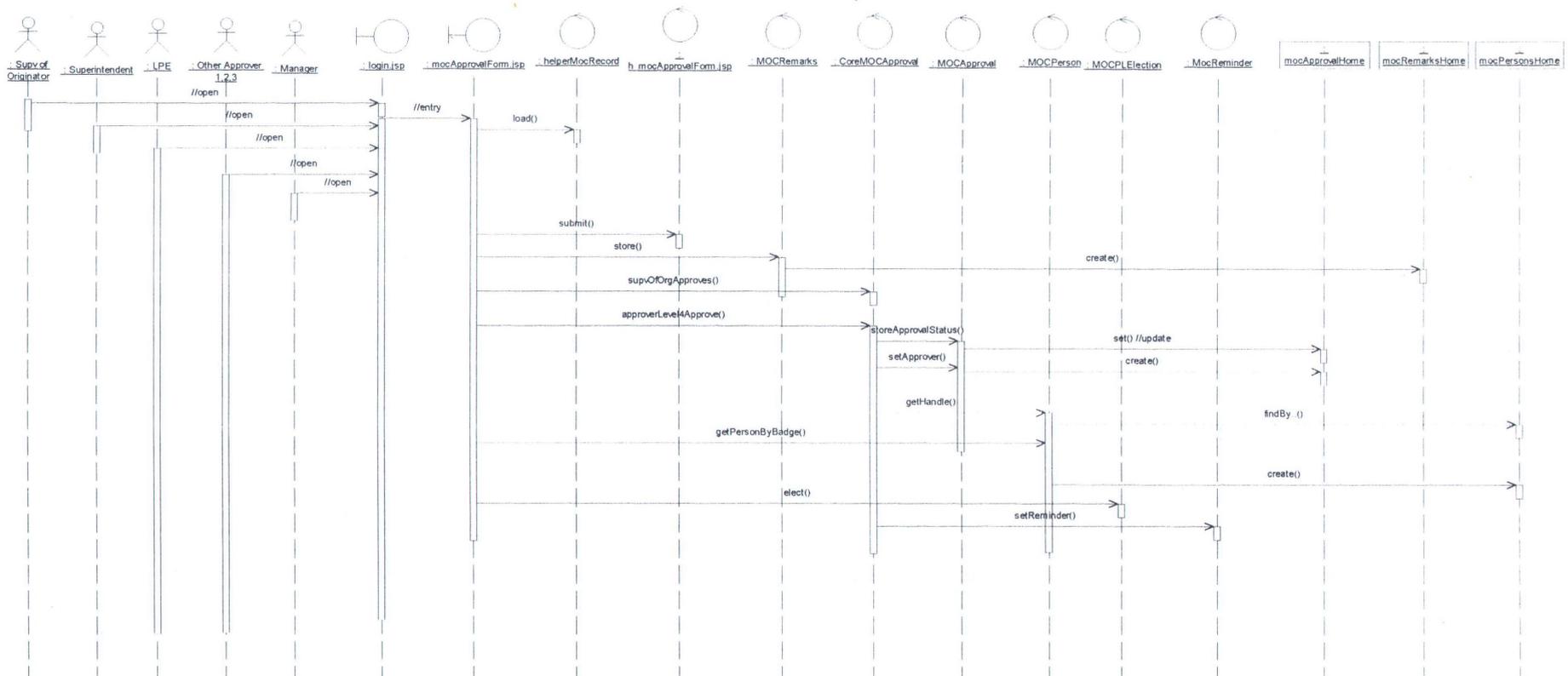
B. Persetujuan Proposal MOC

Pada diagram sequence persetujuan proposal MOC ini terdapat beberapa aktor yang terlibat, antara lain Supervisor, Superintendent, Manager dan Lead Process Engineer(LPE) yang secara kolektif dapat disebut sebagai para *approver*. Proses dimulai saat para *approver* ini membuka halaman persetujuan (*mocApprovalForm.jsp*). Selanjutnya Approver dapat mengisi keterangan atau catatan tambahan pada form tersebut dan mengaktifkan fungsi persetujuan (*approve*) dengan menekan tombol *approve*.

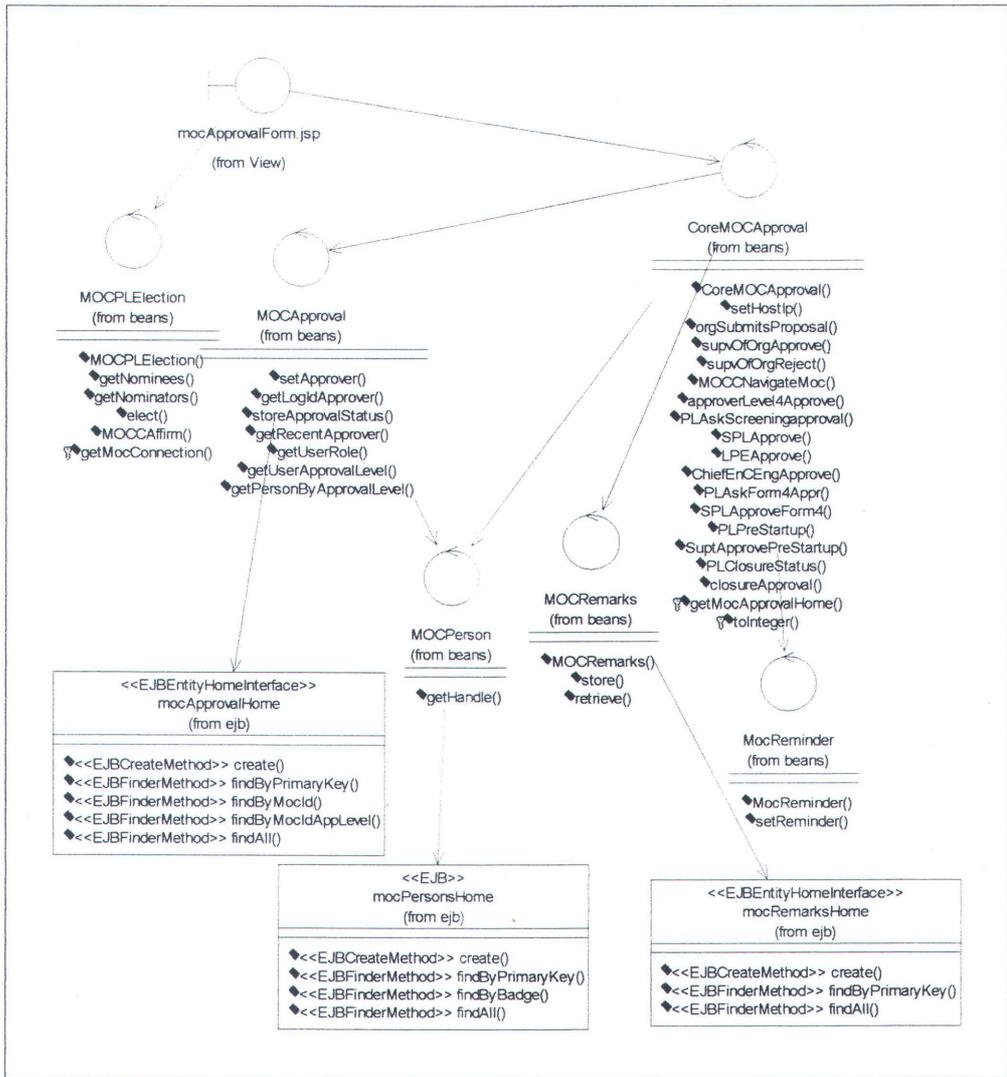
Selanjutnya sistem akan memanggil *h_mocApprovalForm.jsp*. Script JSP ini kemudian memanggil beberapa bean dan EJB lain.

- *helperMocRecord.java* dipergunakan untuk mendapatkan data proposal MOC untuk ditampilkan ke formulir persetujuan.
- *MOCRemarks.java* untuk menyimpan komentar atau catatan dari para *approver* yang disertakan dalam formulir persetujuan.
- *CoreMOCApproval.java* untuk menyimpan status persetujuan (*approve* atau *reject*), menentukan *approver* berikutnya, dan menyiapkan pengingat (*reminder*) bagi *approver* berikutnya.
- *MOCApproval.java* dipergunakan oleh bean *CoreMOCApproval.java* untuk menyimpan status persetujuan, menentukan *approver* berikutnya.
- *MOCPerson.java* dipergunakan oleh bean *CoreMOCApproval.java* untuk mendapatkan kode ID karyawan dalam sistem e-MOC.

- *MOCPLElection.java* berfungsi untuk menyimpan pilihan calon *Project Leader* yang diajukan oleh para *approver*.
- *MocReminder.java* dipergunakan oleh *CoreMOCApproval.java* untuk menyiapkan pengingat (*reminder*).
- *mocApprovalHome.java* merupakan *home interface* dari EJB *mocApproval*. EJB ini dipergunakan untuk menangani data persetujuan MOC di tabel *MOC2_APPROVAL*.
- *mocRemarksHome.java* merupakan *home interface* dari EJB *mocRemarks*. EJB ini berfungsi untuk menangani pengelolaan data catatan atau komentar yang berhubungan dengan MOC ID tertentu di tabel *MOC2_REMARKS*.
- *mocPersonsHome.java* merupakan *home interface* dari EJB *mocPersons*. EJB ini berfungsi untuk menangani data karyawan dalam sistem e-MOC di tabel *MOC2_PERSONS*.



Gambar 4.14 Diagram Sequence Persetujuan Proposal MOC

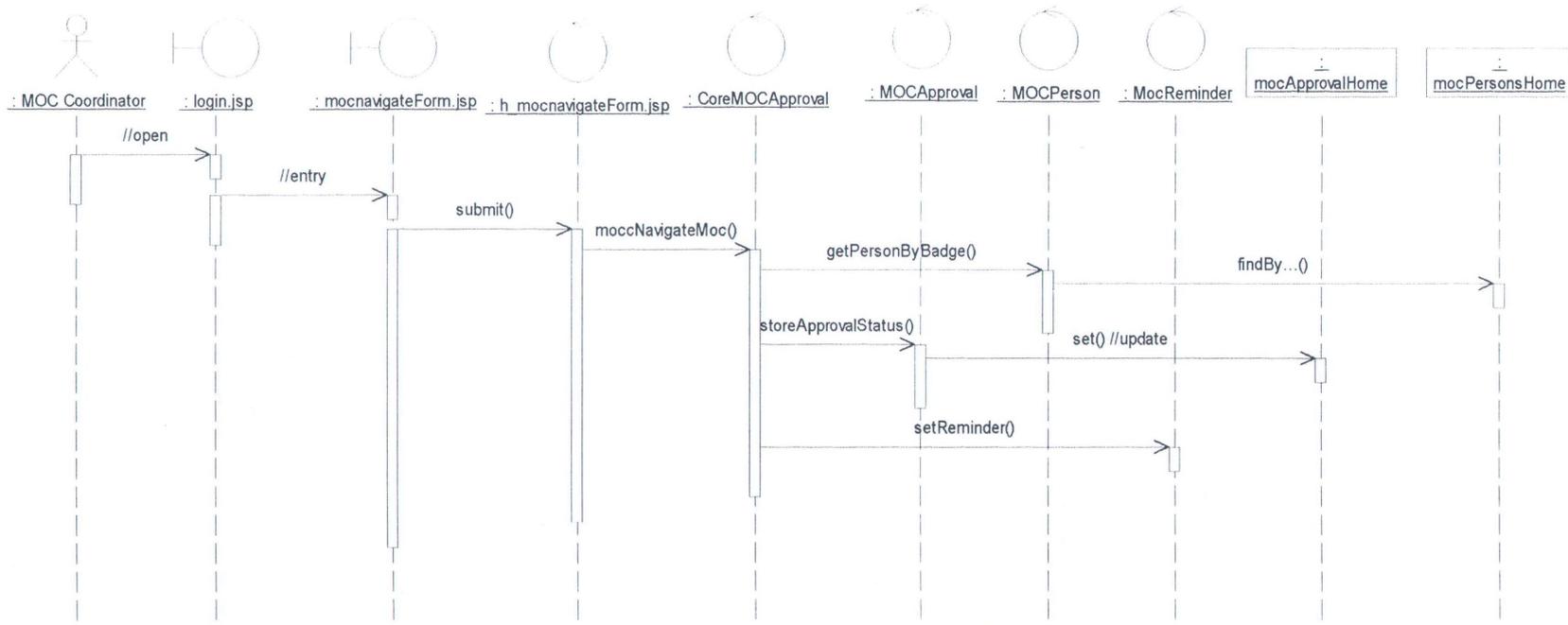


Gambar 4.15 Diagram Class Pengajuan MOC

C. Penentuan *Approver* proposal MOC

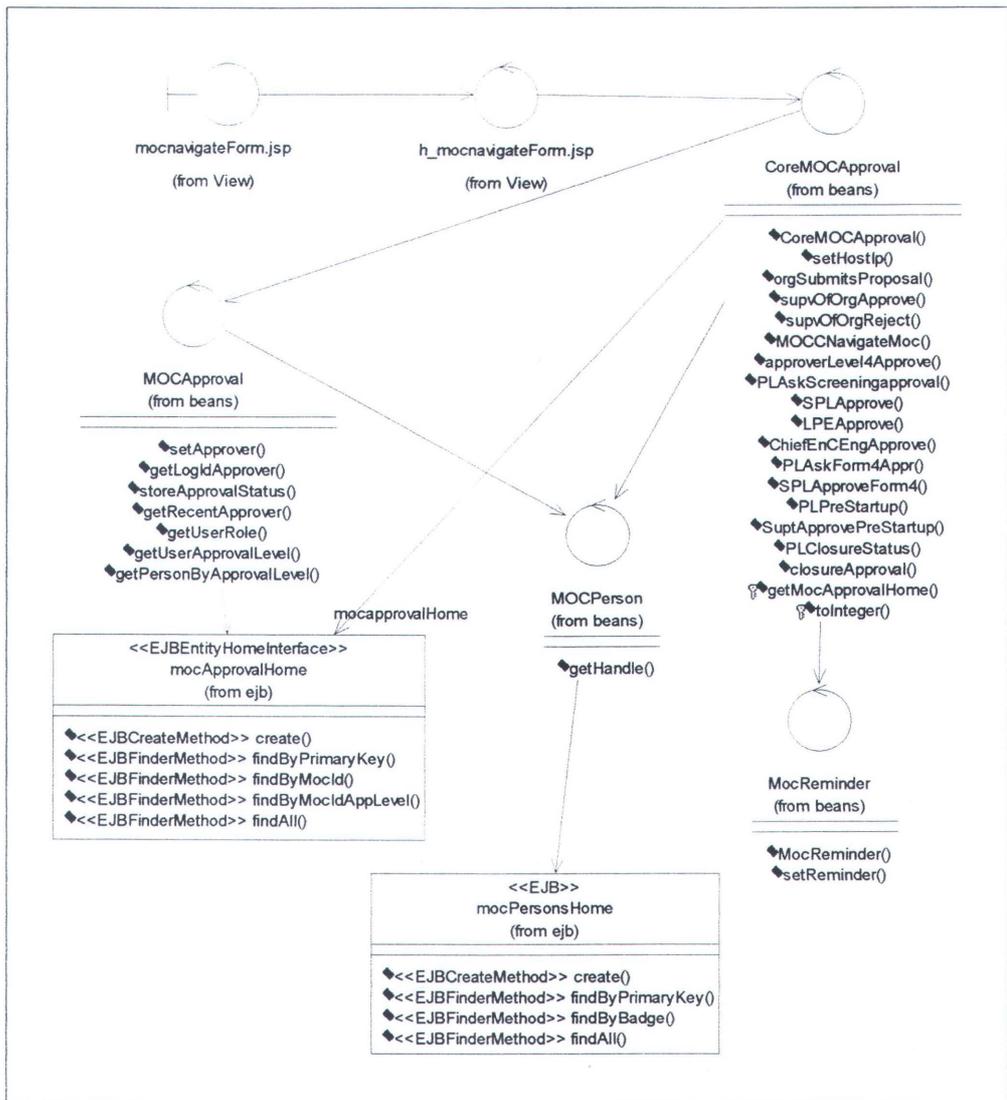
Pada diagram sequence penentuan *approver* bagi proposal MOC, aktor yang berperan adalah MOC Coordinator. Disini MOC Coordinator akan membuka formulir *mocNavigateForm.jsp*, kemudian menentukan orang-orang yang akan diberikan kewenangan untuk memberikan persetujuan terhadap proposal MOC yang diajukan. Setelah MOC Coordinator menekan tombol *dispatch MOC*, sistem akan memanggil *h_mocNavigateForm.jsp*. JSP ini akan mengakses beberapa bean dan EJB lain sebagai berikut.

- *CoreMOCApproval.java* dipergunakan untuk menyimpan daftar *approver* yang dipilih dan mengirimkan e-mail pemberitahuan dan pengingat (*reminder*).
- *MOCApproval.java* dalam hal ini dipergunakan oleh bean *CoreMOCApproval.java* untuk menyimpan data *approver*.
- *MocReminder.java* dipergunakan oleh *CoreMOCApproval.java* untuk menyiapkan pengingat (*reminder*).
- *MOCPerson.java* dipergunakan oleh bean *CoreMOCApproval.java* untuk mendapatkan kode ID karyawan dalam sistem e-MOC.
- *mocApprovalHome.java* merupakan *home interface* dari EJB *mocApproval*. EJB ini dipergunakan untuk menangani data persetujuan MOC di tabel MOC2_APPROVAL.
- *mocRemarksHome.java* merupakan *home interface* dari EJB yang berfungsi untuk menangani data catatan atau komentar yang berhubungan dengan MOC ID tertentu di tabel MOC2_REMARKS.



Gambar 4.16 Diagram Sequence Penentuan Approver proposal MOC



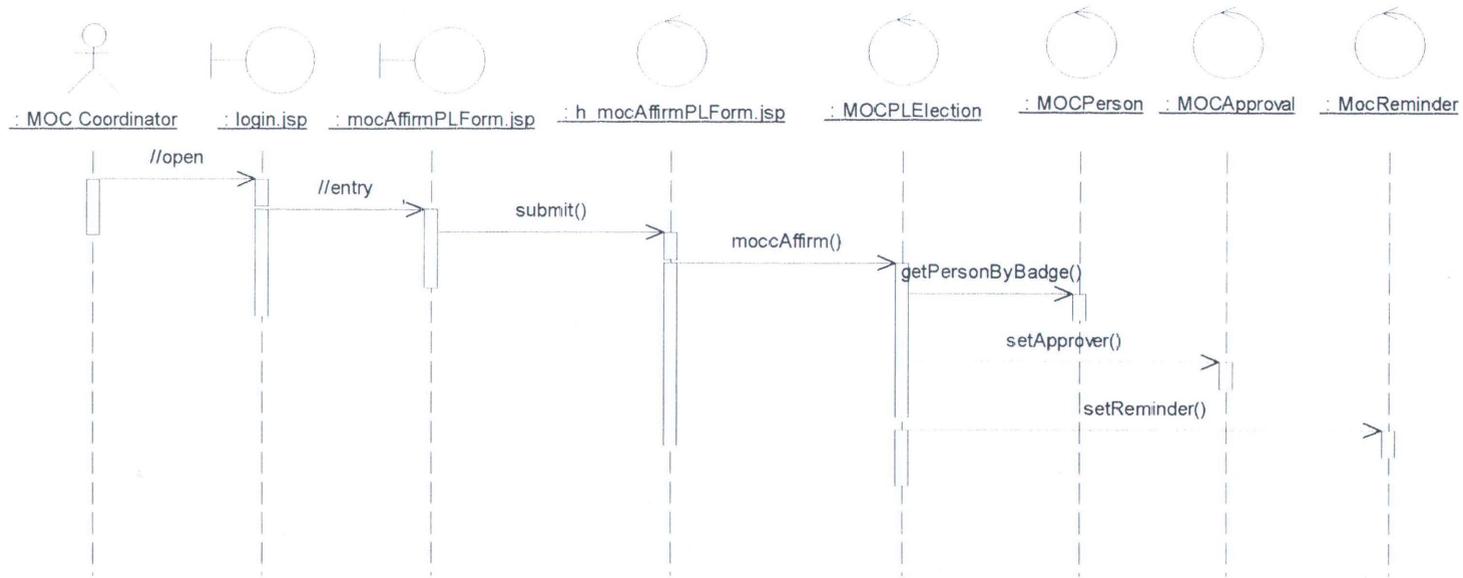


Gambar 4.17 Diagram Class Penentuan Approver proposal MOC

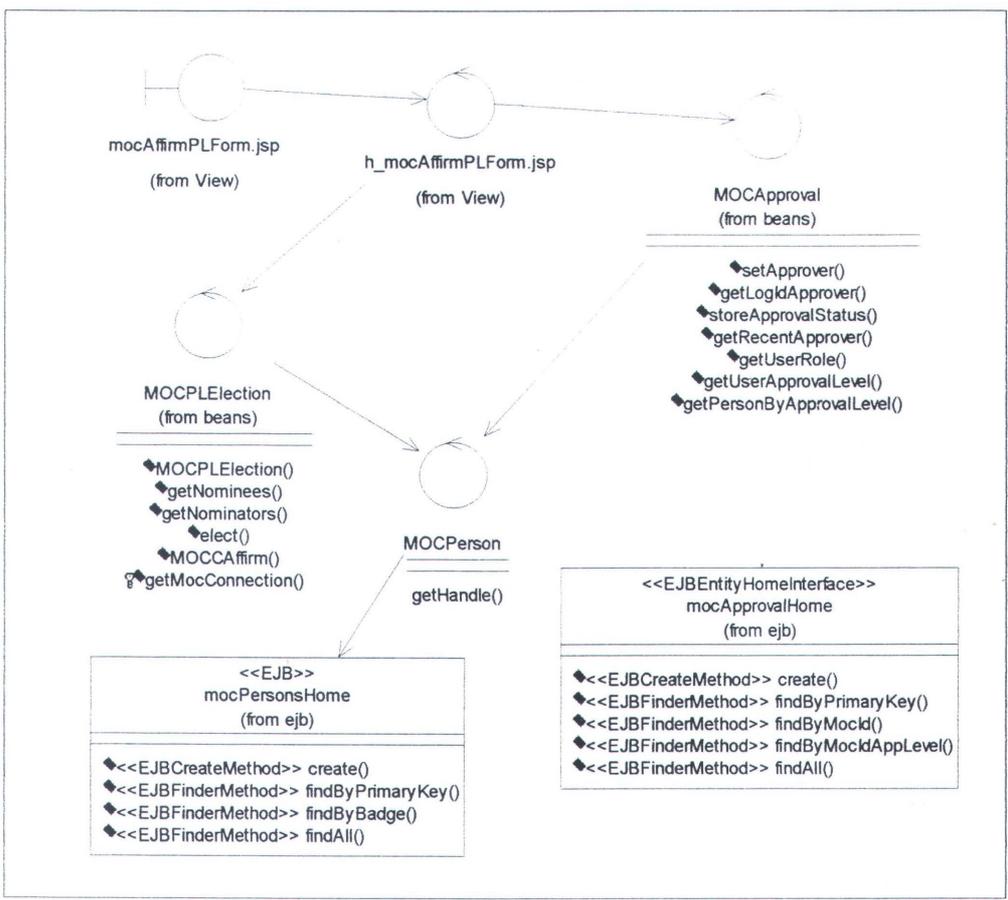
D. Penunjukan Project Leader

Pada diagram sequence penunjukan Project Leader ini, aktor yang memegang peranan adalah MOC Coordinator. MOC Coordinator akan membuka halaman *mocAffirmPLForm.jsp*, memilih seorang Project Leader dan kemudian mengaktifkan fungsi pemilihan tersebut dengan menekan tombol *Assign PL*. Sistem kemudian akan memanggil *h_mocAffirmPLForm.jsp*. JSP ini kemudian akan mengakses beberapa bean lain. Bean yang dipanggil adalah sebagai berikut:

- *MOCPLElection.java* berfungsi untuk menetapkan pilihan Project Leader. Bean ini akan mengakses beberapa bean lain untuk melaksanakan fungsinya.
- *MOCApproval.java* dalam hal ini dipergunakan oleh bean *MOCPLElection.java* untuk memasukkan nama Project Leader terpilih ke alur proses MOC.
- *MocReminder.java* dipergunakan oleh *MOCPLElection.java* untuk menyiapkan pengingat (*reminder*).
- *MOCPerson.java* dipergunakan oleh bean *MOCPLElection.java* untuk mendapatkan kode ID karyawan dalam sistem e-MOC.



Gambar 4.18 Diagram Sequence Penunjukan Project Leader

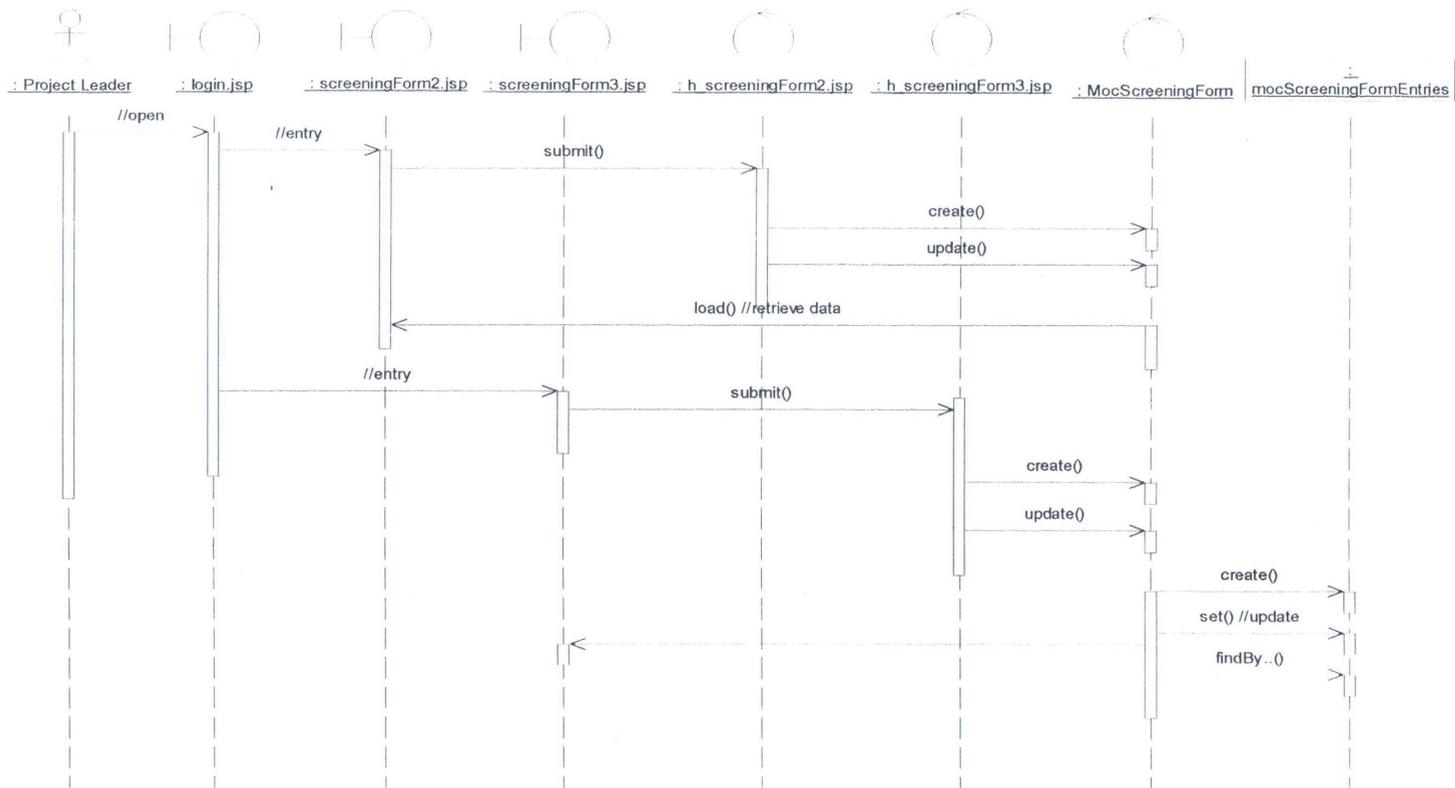


Gambar 4.19 Diagram Class Penunjukan Project Leader

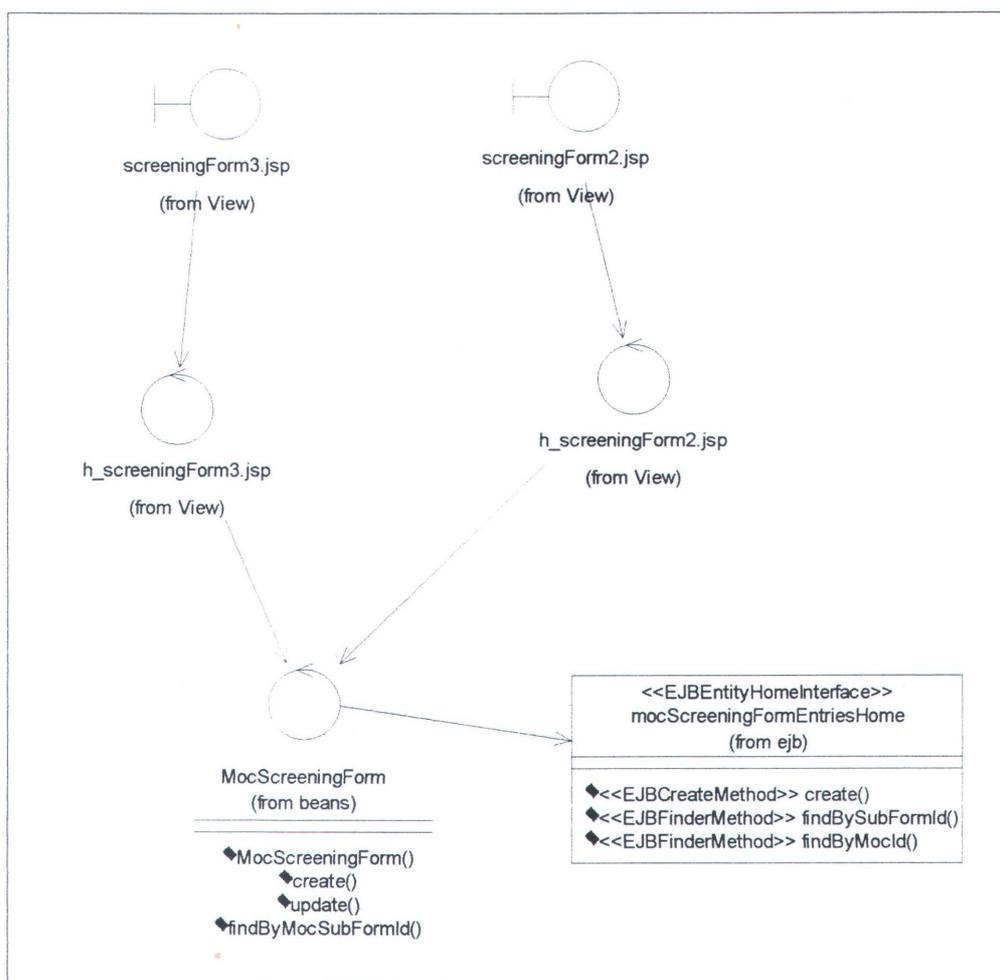
E. Pengisian MOC *Screening* (MOC Form 2 dan 3)

Pada diagram sequence MOC *screening* ini, aktor yang berperan adalah Project Leader. Project Leader akan mengisi MOC Form 2 dengan hasil analisis terhadap proyek perubahan yang direncanakan untuk memenuhi proposal MOC. Project Leader juga akan mengisi MOC Form 3 dengan hasil analisis terhadap rencana pelaksanaan. Selanjutnya setelah Project Leader menekan *submit*, sistem akan mengakses sebuah bean dan EJB untuk melakukan penyimpanan hasil *screening* yang telah dilakukan. Bean dan EJB tersebut adalah sebagai berikut:

- *MocScreeningForm.java* dipergunakan untuk melakukan penyimpanan data *screening* form 2 dan 3. Bean ini melakukan enkapsulasi terhadap akses ke komponen EJB sehingga lebih mudah dipergunakan.
- *mocScreeningFormentries.java* merupakan sebuah komponen EJB yang dipergunakan oleh *MocScreeningForm.java* untuk melakukan pengelolaan data isi *screening form*.



Gambar 4.20 Diagram Sequence Screening oleh Project Leader

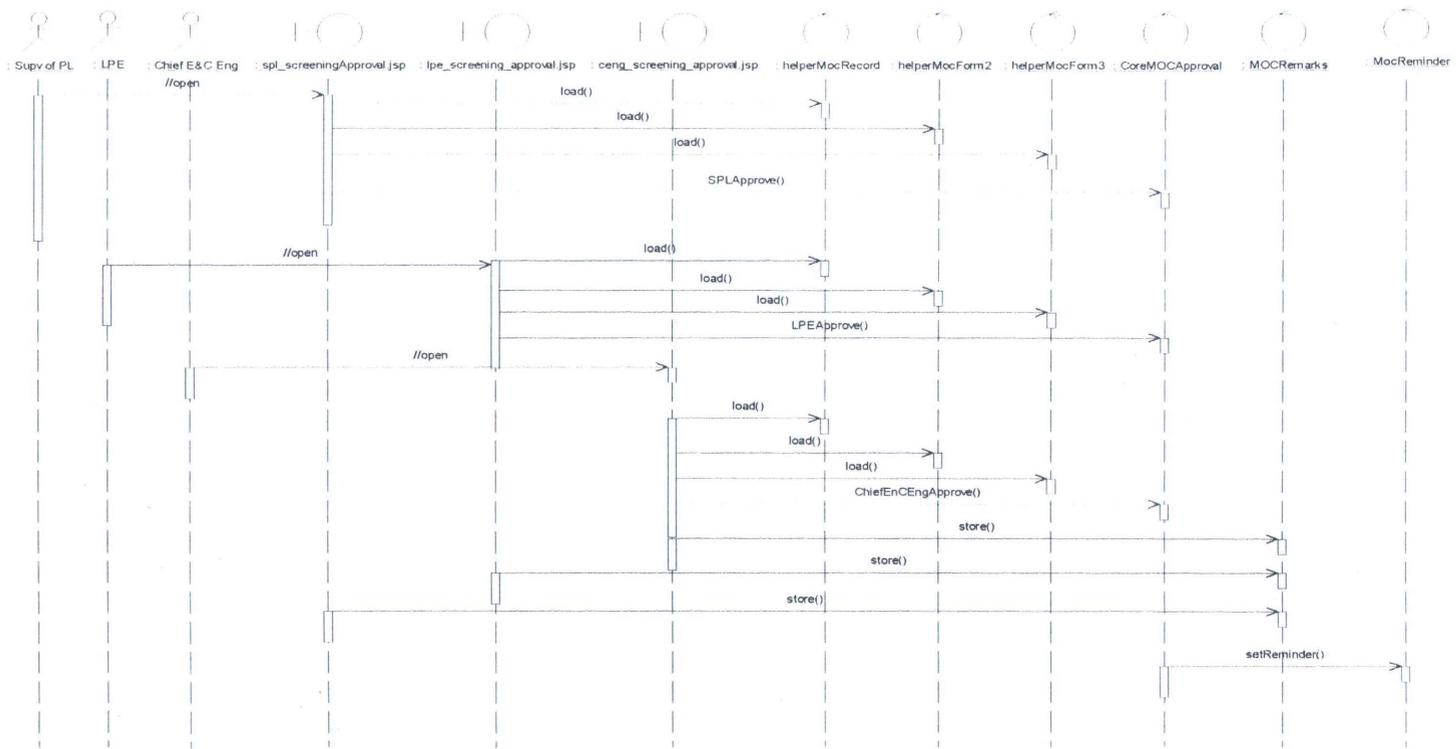


Gambar 4.21 Diagram Class Screening oleh Project Leader

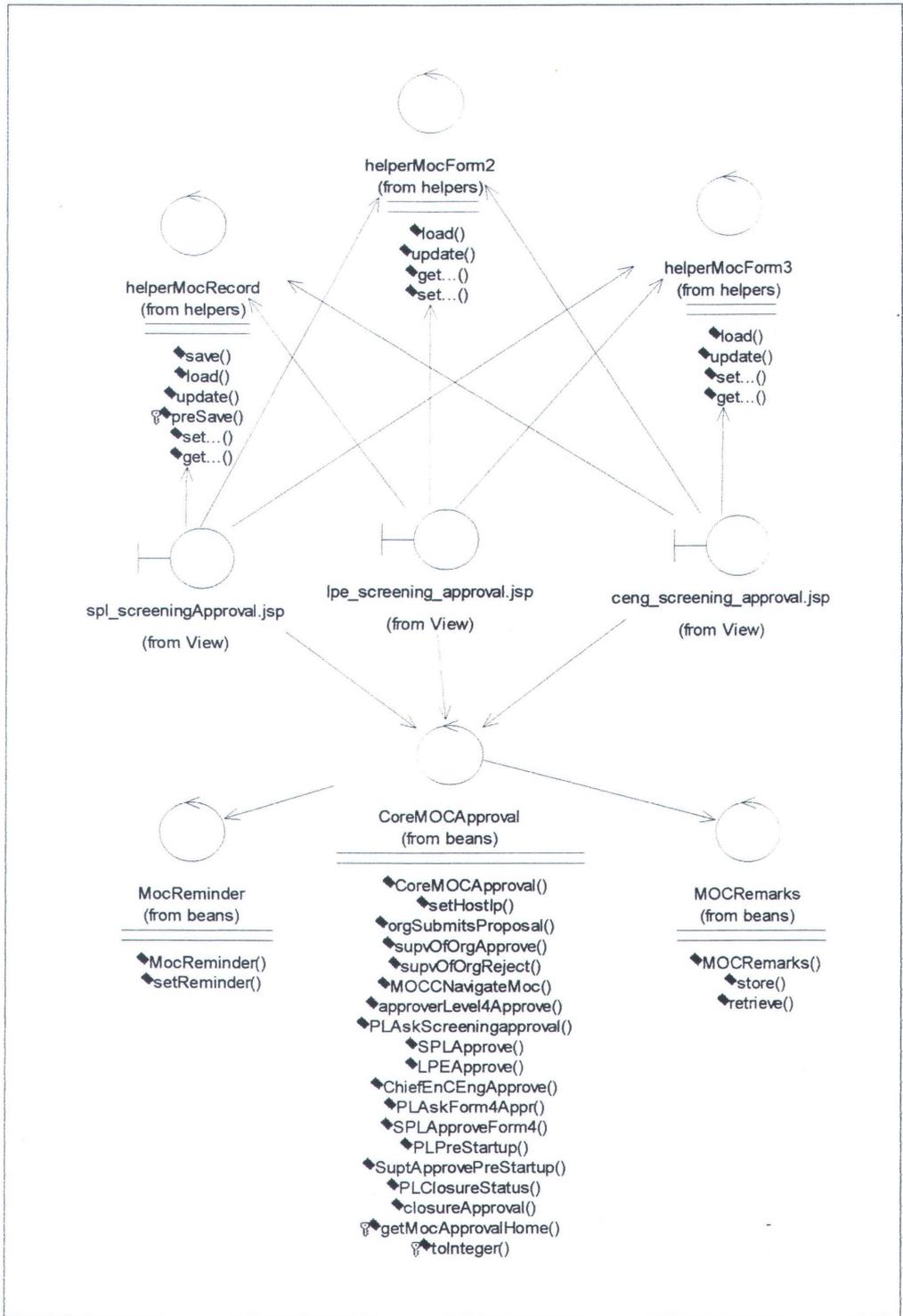
F. Persetujuan Terhadap Form 2 dan 3

Pada diagram sequence persetujuan terhadap hasil screening (form 2 dan 3), aktor yang berperan adalah Supervisor dari Project Leader, Lead Process Engineer (LPE) dan Chief E&C Engineer. Proses dimulai saat aktor membuka halaman persetujuan dan memeriksa Form 2 dan 3 yang dikirimkan oleh Project Leader. Setelah aktor memberikan persetujuan dengan menekan tombol *approve*, sistem akan memanggil script JSP yang sesuai untuk masing-masing aktor. Script JSP ini kemudian akan mengakses beberapa bean lain. Bean-bean yang diakses adalah sebagai berikut:

- *helperMocRecord.java* merupakan sebuah bean yang berfungsi untuk membantu menyimpan, memuat dan memperbarui isi MOC Form 1.
- *helperMocForm2.java* merupakan sebuah bean yang berfungsi untuk membantu menyimpan, memuat dan memperbarui isi MOC Form 2.
- *helperMocForm3.java* merupakan sebuah bean yang berfungsi untuk membantu menyimpan, memuat dan memperbarui isi MOC Form 3.
- *MOCRemarks.java* untuk menyimpan komentar atau catatan dari para *approver* yang disertakan dalam formulir persetujuan.
- *CoreMOCApproval.java* untuk menyimpan status persetujuan (*approve* atau *reject*), menentukan *approver* berikutnya, dan menyiapkan pengingat (*reminder*) bagi *approver* berikutnya.



Gambar 4.22 Diagram Sequence Persetujuan terhadap Form 2 dan 3

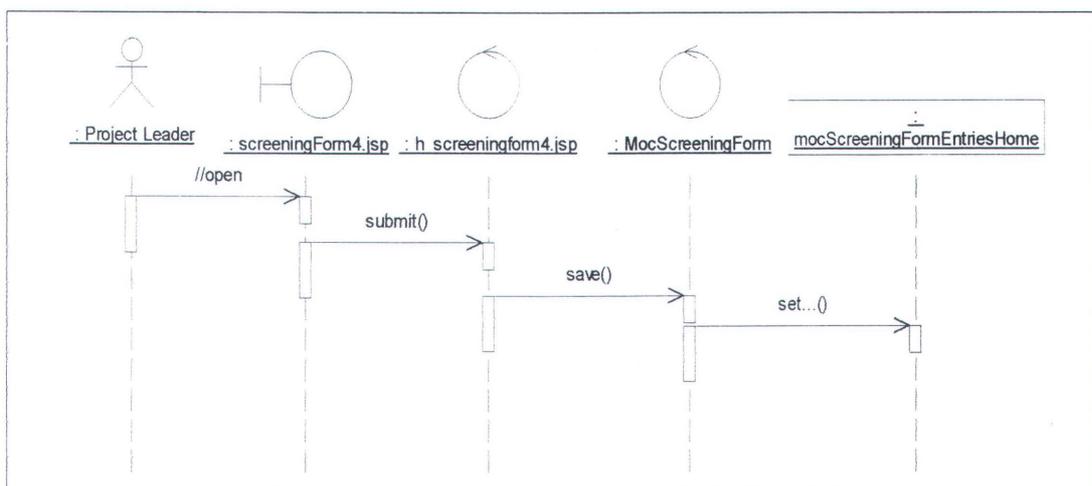


Gambar 4.23 Diagram Class Persetujuan terhadap Form 2 dan 3

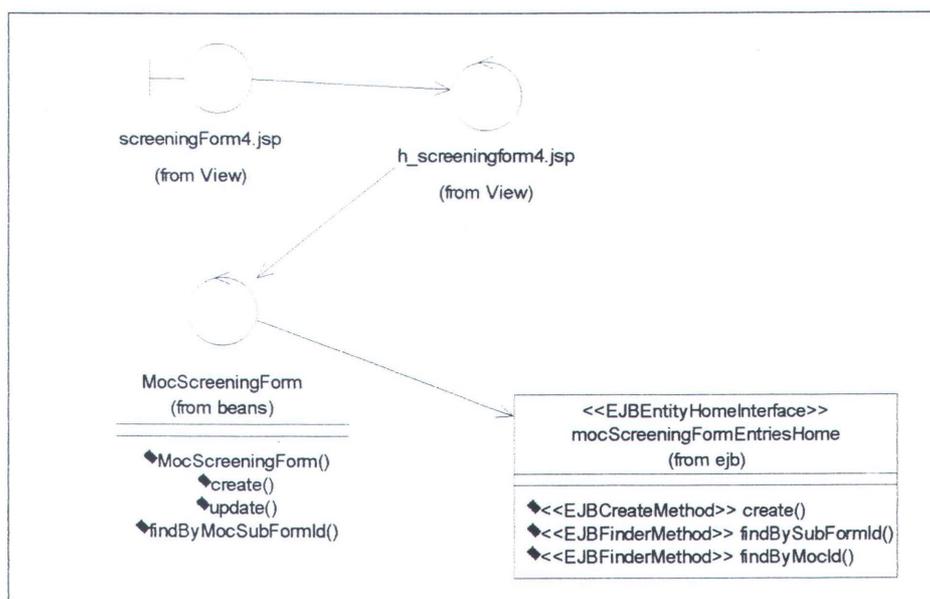
G. Pengisian Implementation Hazard Screening (MOC Form 4)

Pada diagram sequence pengisian MOC Form 4 ini, aktor yang berperan adalah Project Leader. Project Leader akan mengisi MOC Form 4 dengan hasil analisis terhadap rencana pelaksanaan proyek perubahan yang direncanakan untuk memenuhi proposal MOC. Selanjutnya setelah Project Leader menekan *submit*, sistem akan mengakses sebuah bean dan EJB untuk melakukan penyimpanan hasil *screening* yang telah dilakukan. Bean dan EJB tersebut adalah sebagai berikut:

- *MocScreeningForm.java* dipergunakan untuk melakukan penyimpanan data *screening* form 4. Bean ini melakukan enkapsulasi terhadap akses ke komponen EJB sehingga lebih mudah dipergunakan.
- *mocScreeningFormentries.java* merupakan sebuah komponen EJB yang dipergunakan oleh *MocScreeningForm.java* untuk melakukan pengelolaan data isi *screening form*.



Gambar 4.24 Diagram Sequence Pengisian MOC Form 4



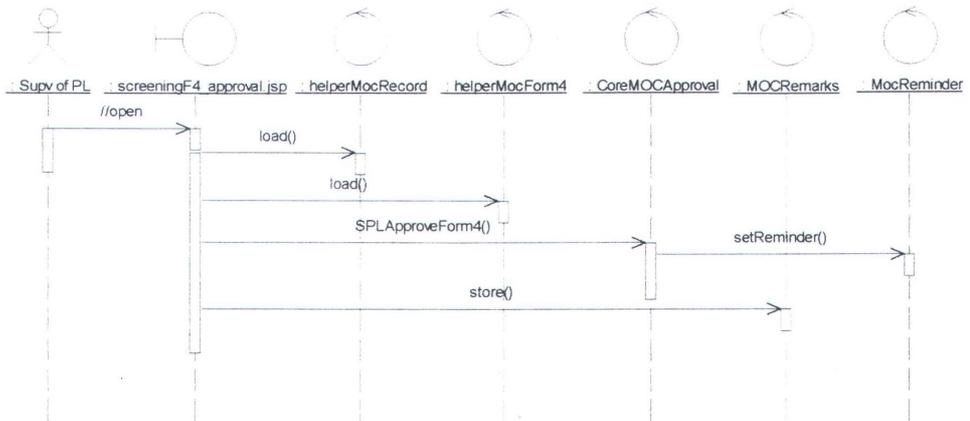
Gambar 4.25 Diagram Class Pengisian MOC Form 4

H. Persetujuan Terhadap MOC Form 4

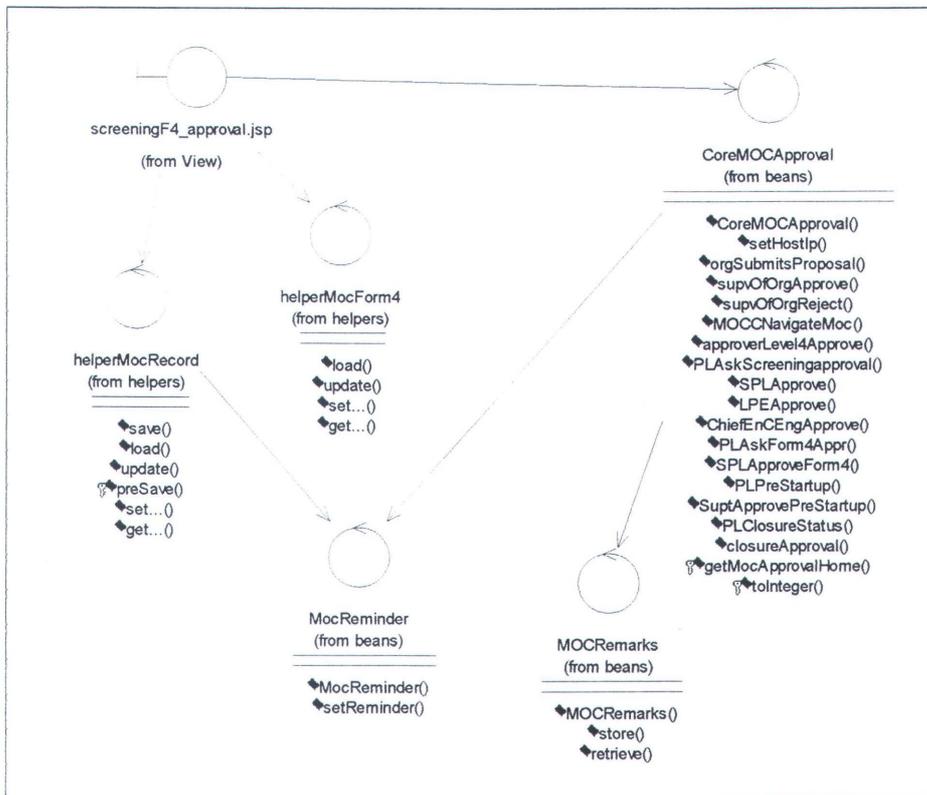
Pada diagram sequence persetujuan terhadap MOC Form 4, aktor yang berperan adalah Supervisor dari Project Leader Proses dimulai saat aktor membuka halaman persetujuan dan memeriksa Form 4 yang dikirimkan oleh Project Leader. Setelah aktor memberikan persetujuan dengan menekan tombol *approve*, sistem akan memanggil script JSP yang sesuai. Script JSP ini kemudian akan mengakses beberapa bean lain. Bean-bean yang diakses adalah sebagai berikut:

- *helperMocRecord.java* merupakan sebuah bean yang berfungsi untuk membantu menyimpan, memuat dan memperbarui isi MOC Form 1.
- *helperMocForm4.java* merupakan sebuah bean yang berfungsi untuk membantu menyimpan, memuat dan memperbarui isi MOC Form 4.

- *MOCRemarks.java* untuk menyimpan komentar atau catatan dari para *approver* yang disertakan dalam formulir persetujuan.
- *CoreMOCApproval.java* untuk menyimpan status persetujuan (*approve* atau *reject*), menentukan *approver* berikutnya, dan menyiapkan pengingat (*reminder*) bagi *approver* berikutnya.



Gambar 4.26 Diagram Sequence Persetujuan Terhadap MOC Form 4

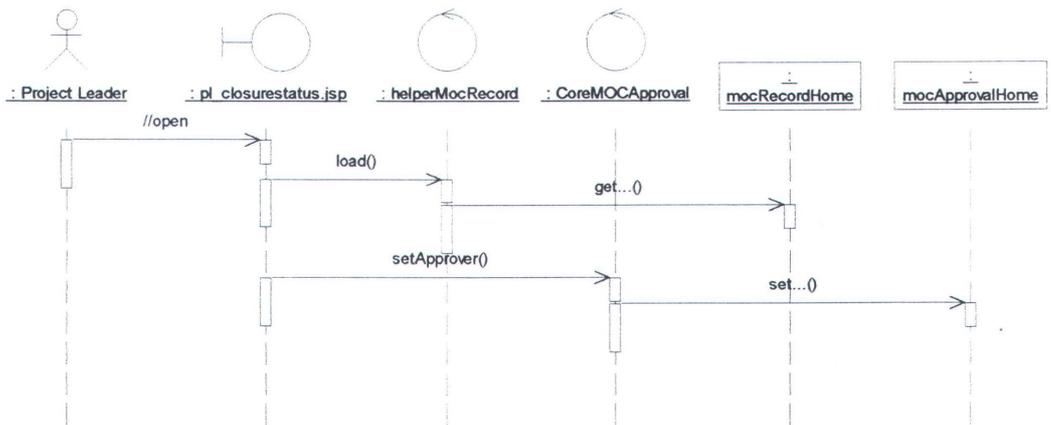


Gambar 4.27 Diagram Class Persetujuan Terhadap MOC Form 4

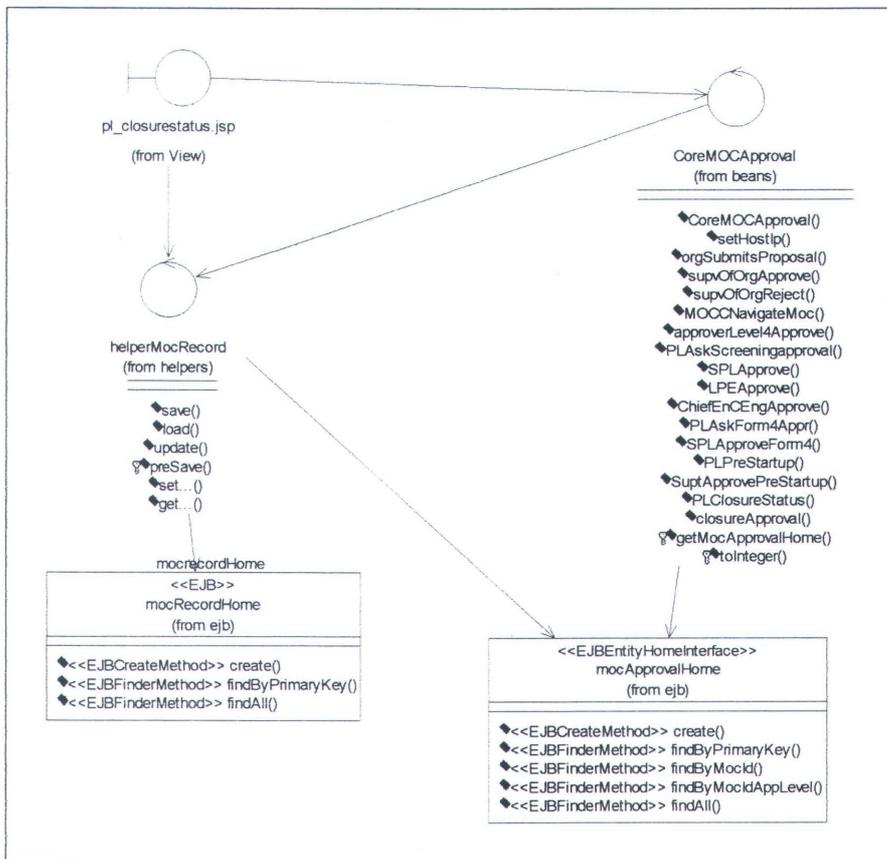
I. Pengajuan Penutupan Proyek Perubahan

Pada diagram sequence persetujuan penutupan proyek perubahan, aktor yang berperan adalah Project LEader. Proses dimulai saat aktor membuka halaman persetujuan penutupan proyek (*pl_closure_status.jsp*) dan menyertakan dokumen-dokumen yang terkait dengan usulan penutupan proyek. Setelah Project Leader menekan tombol Submit, sistem akan memanggil *h_closurestatus.jsp*. Script JSP ini kemudian akan mengakses beberapa bean lain. Bean-bean yang diakses adalah sebagai berikut:

- *CoreMOCApproval.java* untuk menyimpan status persetujuan terhadap permintaan penutupan proyek perubahan yang diusulkan. Method yang dipanggil dalam hal ini adalah *closureApproval()*.
- *helperMocRecord.java* untuk menyimpan status penutupan terhadap proposal MOC.
- *mocRecordHome.java* merupakan *home interface* dari EJB *mocRecord*. EJB ini berfungsi untuk mengelola data MOC Form 1 di tabel *MOC2_RECORD*.
- *mocApprovalHome.java* merupakan *home interface* dari EJB *mocApproval*. EJB ini dipergunakan untuk menangani data persetujuan MOC di tabel *MOC2_APPROVAL*.



Gambar 4.28 Diagram Sequence Pengajuan Penutupan Proyek Perubahan

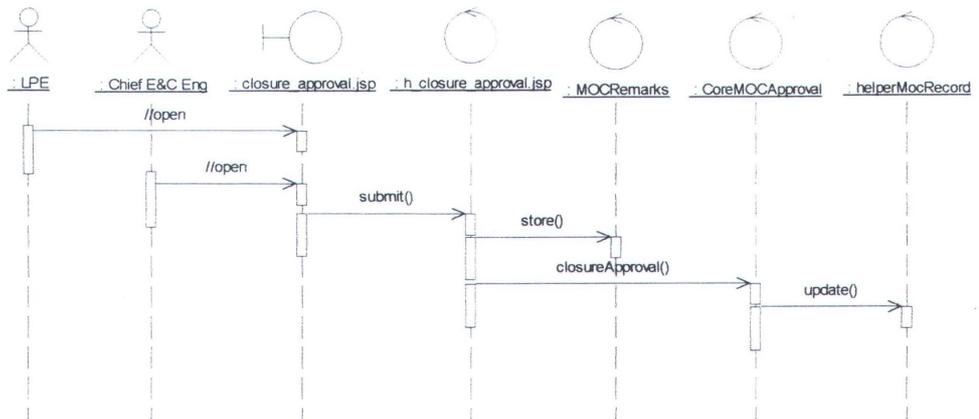


Gambar 4.29 Diagram Class Pengajuan Penutupan Proyek Perubahan

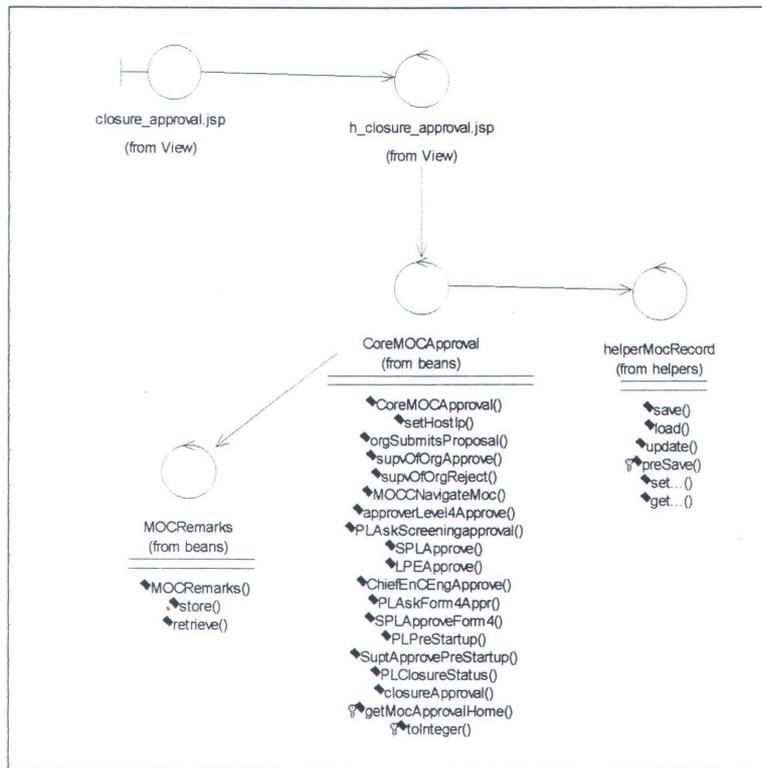
J. Persetujuan Penutupan Proyek Perubahan

Pada diagram sequence persetujuan penutupan proyek perubahan, aktor yang berperan adalah Lead Process Engineer (LPE) dan Chief E&C Engineer. Proses dimulai saat aktor membuka halaman persetujuan penutupan proyek (*closure_approval.jsp*) dan memeriksa dokumen-dokumen yang disertakan oleh Project Leader dalam usulan penutupan proyek. Setelah aktor memberikan persetujuan dengan menekan tombol *approve*, sistem akan memanggil *h_closureapproval.jsp*. Script JSP ini kemudian akan mengakses beberapa bean lain. Bean-bean yang diakses adalah sebagai berikut:

- *MocRemarks.java* untuk menyimpan komentar atau catatan yang diberikan oleh LPE atau Chief E&C Engineer terkait persetujuan perubahan yang diajukan.
- *CoreMOCApproval.java* untuk menyimpan status persetujuan terhadap permintaan penutupan proyek perubahan yang diusulkan. Method yang dipanggil dalam hal ini adalah *closureApproval()*.
- *helperMocRecord.java* untuk menyimpan status penutupan terhadap proposal MOC.



Gambar 4.30 Diagram Sequence Persetujuan terhadap penutupan proyek perubahan



Gambar 4.31 Diagram Class Persetujuan terhadap pengajuan penutupan proyek perubahan

4.2.6 Perancangan Form

Form-form ini merupakan objek penting dalam sebuah permohonan MOC, terutama form 1, 2, 3 dan 4. Berikut adalah rancangan dari tiap-tiap form yang ada dalam sistem MOC.

A. Proposal MOC (Form 1)

Merupakan form utama dan pertama bagi sistem MOC. Dengan diajukan form ini, maka dimulailah proses-proses berikutnya. Pada form ini, originator memasukkan deskripsi selengkapny tentang perubahan (change) yang diusulkan. Originator juga dapat menyertakan berkas-berkas pendukung sebagai *attachment*.

Nama originator, badge, dan penyalianya ditentukan secara otomatis oleh sistem berdasarkan informasi login. Pada tahap ini, nomor MOC akan ditentukan oleh sistem pada saat proposal MOC disimpan ke basis data.

Berikut adalah rancangan dari form 1.

UNOCAL Management of Changes			
CHANGE REQUEST			MOC Number
Field	Work Order	Issued Date	<%=Current
Facility	AFE/CC	Budgeted	
Originator	Title / Position		
CHANGE DESCRIPTION		Attachment(s)	
		Manage Attachments	
TYPE OF CHANGE			
<input type="checkbox"/> New Project <input type="checkbox"/> Change <input type="checkbox"/> RIK Non Routine <input type="checkbox"/> Permanent <input type="checkbox"/> Temporary			
Change duration (week)		Entry for temporary changes	
AIMS record		Entry for temporary changes	
APPROVAL SHEET			
RCLE	NAME	POSITION	ACTION
Supervisor of Originator	<%=approver.getSupName()%>	<%=approver.getSupJobTitle()%>	
Create MOC >>			Start over Cancel

Gambar 4.32 MOC Proposal (Form 1)

Action-action yang ada dalam form ini adalah:

a. Manage Attachments

Tombol ini digunakan untuk menambahkan atau menghapus file attachment yang akan disertakan dalam sebuah pengajuan MOC.

b. Create MOC

Tombol ini ditekan jika Originator ingin memproses pengajuan sebuah MOC.

c. Start Over

Digunakan untuk mengosongkan kembali form yang telah diisi.

d. Cancel

Tombol ini ditekan jika Originator tidak jadi mengajukan MOC.

e. *Icon* User

Icon ini digunakan untuk menampilkan popup user.

B. Hazard Analysis (Form 2)

Pada form ini apabila ada satu saja pertanyaan yang dijawab “Yes”, maka proses *Hazard Analysis* secara mendalam perlu dilakukan. Proses ini akan dilakukan secara eksternal, namun Project Leader diharapkan menandai metode analisis yang akan dipergunakan. Pada gambar dibawah ini akan terlihat rancangan form 2.

UNOCAL 76
Management of Changes

MOC no. 016.01.2002

FORM 2 : HAZARD ANALYSIS SCREENING FORM

Instruction:
A Hazard Analysis should be performed if any screening questions are answered "Yes". The potential for loss, including the impact on worker and public safety is considered minimal and a Hazard Analysis need not to be performed if all questions are answered "No".

FOR THE PROPOSED WORK		Y/N	REMARKS
1	Should a Hazard Analysis / Risk Assessment be performed as a precautionary measure because regulatory agencies, the local community, or others may perceive that an unacceptable hazard to the public or environment exists?	<input type="checkbox"/>	
2	Is a critical facility, process or piece of equipment involved (i.e. those that could result in significant harm to people, property or the environment, and associated economic losses, caused by the materials, locations or operating conditions involved)?	<input type="checkbox"/>	
3	Is the system considered complex (i.e. involves a large number of components or interacts with multiple or altered)?	<input type="checkbox"/>	
4	Is a safeguarding system, critical control or communications system added, removed, bypassed or altered?	<input type="checkbox"/>	
11	Are there significant changes or critical issues associated with the transport of personnel, materials, or equipment to and from the work site?	<input type="checkbox"/>	
12	Will the implementation/construction phase present new or unique hazards that require special controls?	<input type="checkbox"/>	
13	Is the change considered "temporary" ?	<input type="checkbox"/>	
RESULTS : Hazard Analysis to be performed ?		<input type="checkbox"/>	
Technique(s) to be used			<input type="checkbox"/> WHAT-IF <input type="checkbox"/> HAZOP <input type="checkbox"/> CHECK LIST <input type="checkbox"/> FMEA

Update >> Start Over

Gambar 4.33 Hazard Analysis (Form 2)

Action-action yang ada dalam form ini adalah:

- a. Update

Tombol ini digunakan untuk menyimpan hasil isian ke dalam basis data.

- b. Start Over

Tombol ini digunakan untuk mengembalikan keadaan form ke kondisi semula (*reset*).



C. MOC Plan (Form 3)

Pada form ini setiap parameter diisi sesuai dengan kenyataannya agar rencana perubahan (*change*) dapat benar-benar teruji kelayakannya. Setiap item, dapat dipilih apakah “YES” atau “NO”. Berikut adalah potongan rancangan dari form 3.

CLOSE OUT	PROGRAMS	
	<input type="checkbox"/> Maintenance	<input type="checkbox"/> Calibrations
	<input type="checkbox"/> Vessel Inspections	<input type="checkbox"/> PSV Testing
	<input type="checkbox"/> Safety Device Testing	<input type="checkbox"/> Others ...
	COMPLETION DOCUMENT	
	<input type="checkbox"/> As built drawing to E&C Drafting Supervisor	<input type="checkbox"/> Notify Affected Personnelnull
	<input type="checkbox"/> Operating / Maintenance Manuals	<input type="checkbox"/> Others ..
<input style="border: none;" type="button" value=" << Back to Form 1. "/> <input style="border: none;" type="button" value=" Submit "/> <input style="border: none;" type="button" value=" Print "/>		

Gambar 4.34 Form 3

Action-action yang ada dalam form ini adalah:

- a. Back to form 1

Tombol ini digunakan jika ingin kembali ke form 1.

- b. Submit

Tombol ini digunakan untuk menyimpan ke dalam basis data, semua pilihan untuk setiap item yang ada.

- c. Print

Tombol ini digunakan untuk menampilkan halaman versi cetak ke printer dan otomatis mengeluarkan dialog box untuk perintah *print*.

D. Implementation Hazards Screening (Form 4)

Seperti halnya form 2 dan 3, form ini juga harus diisi sesuai dengan kenyataan yang sebenarnya, berdasarkan analisa-analisa yang telah dilakukan sebelumnya. Gambar berikut ini adalah potongan rancangan form 4.

12	Are simultaneous operations involved?	NO	
RESULTS: Are there concerns to be resolved?		NO	
Item #	CONCERN TO RESOLVE	ACTION TAKEN BY	
1			
2			
3			
JOB SAFETY ANALYSIS (JSA) REQUIREMENT			
1	Does the work, if not done properly, result in major loss while being performed in terms of how serious the loss likely to be (the event of injury, cost of damage, cost of quality, cost of production loss, cost of environmental damage.)	NO	
2	COMMENTS: Mention if any concern		
JOB SAFETY ANALYSIS TO BE PERFORMED			NO
Submit		Start Over	

Gambar 4.35 Form 4

Action-action yang terdapat pada form ini adalah:

- a. Submit

Tombol ini digunakan untuk memproses form 4 yang telah diisikan.

- b. Start Over

Digunakan untuk mengosongkan kembali form yang telah diisi.

E. Form Persetujuan Proposal MOC

Pada form ini, masing-masing Approver juga dapat mengisikan calon Project Leader. Berikut gambar rancangan form persetujuan.

UNOCAL 76 Management of Changes		
NOMINATE PROJECT LEADER		
Proposed Project Leader		
REMARKS		
Please type your remarks about this MOC here		
APPROVAL SHEET		
ROLE	NAME	ACTION
Supervisor of Originator		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Superintendent		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Manager		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Lead Process Engineer		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Other Approver		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Other Approver		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
Other Approver		<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>

Gambar 4.36 Form Persetujuan

Action-action yang ada pada form ini adalah:

- a. Approve (icon cek)

Icon ini dipilih apabila Approver menyetujui pengajuan MOC.

- b. Reject (icon X)

Icon ini dipilih apabila Approver menolak pengajuan MOC.

F. Form MOC Navigator

Form ini berisi detail dari form 1 dan form untuk menentukan Approver yang akan memberikan persetujuan suatu MOC. Gambar berikut adalah rancangan dari form Navigator.

Manage MOC

UNOCAL 76

Management of Changes

UNOCAL 76

CHANGE REQUEST			MOC Number
Field	Work Order		Issued Date
Facility	AFE/CC		Budgeted
Originator	Title / Position		
CHANGE DESCRIPTION			Attachment(s)
TYPE OF CHANGE :			
Change duration (weeks) :			
AIMS record :			

UNOCAL 76

SET APPROVERS		
ROLE	NAME	ACTION
Supervisor of Originator	<%=approvers.getSupvName()%>	
Superintendent	<input type="text"/>	
Manager	<%=approvers.getMgrName()%>	
Lead Process Engineer	<input type="text"/>	
Other Approver 1	<input type="text"/>	
Other Approver 2	<input type="text"/>	
Other Approver 3	<input type="text"/>	

Submit | Cancel

Gambar 4.37 Form MOC Navigator

Action-action yang ada pada form ini adalah:

- a. Submit

Tombol ini ditekan jika MOC Coordinator ingin memproses form yang telah diisinya.

- b. Cancel

Tombol ini dipilih jika MOC Coordinator membatalkan pemrosesan form yang telah diisinya.

4.2.7 Perancangan Report

Selain form-form diatas, dibutuhkan pula sejumlah report yang akan berisi informasi yang dibutuhkan bagi pengguna kendati bersifat opsional. Berikut ini merupakan perancangan report yang ada dalam sistem e-MOC:

A. MOC Form 1, 2, 3 dan 4

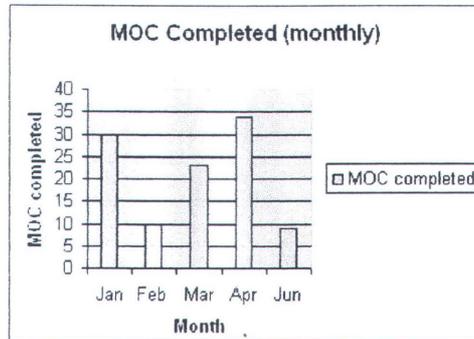
Sistem e-MOC akan memberikan fasilitas untuk menampilkan MOC form 1, 2, 3, dan 4 dalam bentuk halaman HTML yang siap cetak. Pada format siap cetak ini, warna form akan disesuaikan sehingga dapat dicetak dengan jelas. Fasilitas ini diberikan untuk mengantisipasi apabila versi cetak dari formulir elektronik e-MOC masih diperlukan.

MOC Plan			
UNOCAL 76 Management of Changes			
		MOC no. 003.012004	
FORM 3 : MOC PLAN			
DESIGN	<input checked="" type="checkbox"/> PFD	<input checked="" type="checkbox"/> Electrical 1-Line	<input checked="" type="checkbox"/> Operating Procedure
	<input checked="" type="checkbox"/> P&ID	<input checked="" type="checkbox"/> Metering Schematic	<input checked="" type="checkbox"/> Maintenance Procedure
	<input checked="" type="checkbox"/> SAFE Chart	<input checked="" type="checkbox"/> Control Block Diagram	<input checked="" type="checkbox"/> Pre-Commissioning Procedure
	<input checked="" type="checkbox"/> Layouts	<input checked="" type="checkbox"/> Electrical Area Classification	<input checked="" type="checkbox"/> Ref. Standards
	<input checked="" type="checkbox"/> Structural	<input checked="" type="checkbox"/> Fire and Gas Detection	<input checked="" type="checkbox"/> Ref. UIC Specs.
	<input checked="" type="checkbox"/> Permits :	<input checked="" type="checkbox"/> Safe Operating Limits	<input checked="" type="checkbox"/> Other
REVIEW	<input checked="" type="checkbox"/> Sand and Fluid Production	<input checked="" type="checkbox"/> Constructability	<input checked="" type="checkbox"/> Andal, UKL + UPL
	<input checked="" type="checkbox"/> API 14-C	<input checked="" type="checkbox"/> Field Based Reviewed	<input checked="" type="checkbox"/> Others
	<input checked="" type="checkbox"/> Peer Review	<input checked="" type="checkbox"/> Health, Medical, Environment, MSOS	
HAZARD ANALYSIS	<input checked="" type="checkbox"/> Project information prepared (PFD, P&ID, layout, procedures, etc.)	<input checked="" type="checkbox"/> Engineer's response to recommendations (requires management approval)	
	<input checked="" type="checkbox"/> Hazards Analysis team formed (leader, scribe, field representation, etc.)	<input checked="" type="checkbox"/> Develop Action Item Table for monitoring (input into AIMS)	
	<input checked="" type="checkbox"/> Hazard Analysis conducted (recommendations identified)	<input checked="" type="checkbox"/> Complete all Action Items (close in AIMS)	

Gambar 4.38 Form 3 Siap Cetak

B. Grafik statistik MOC

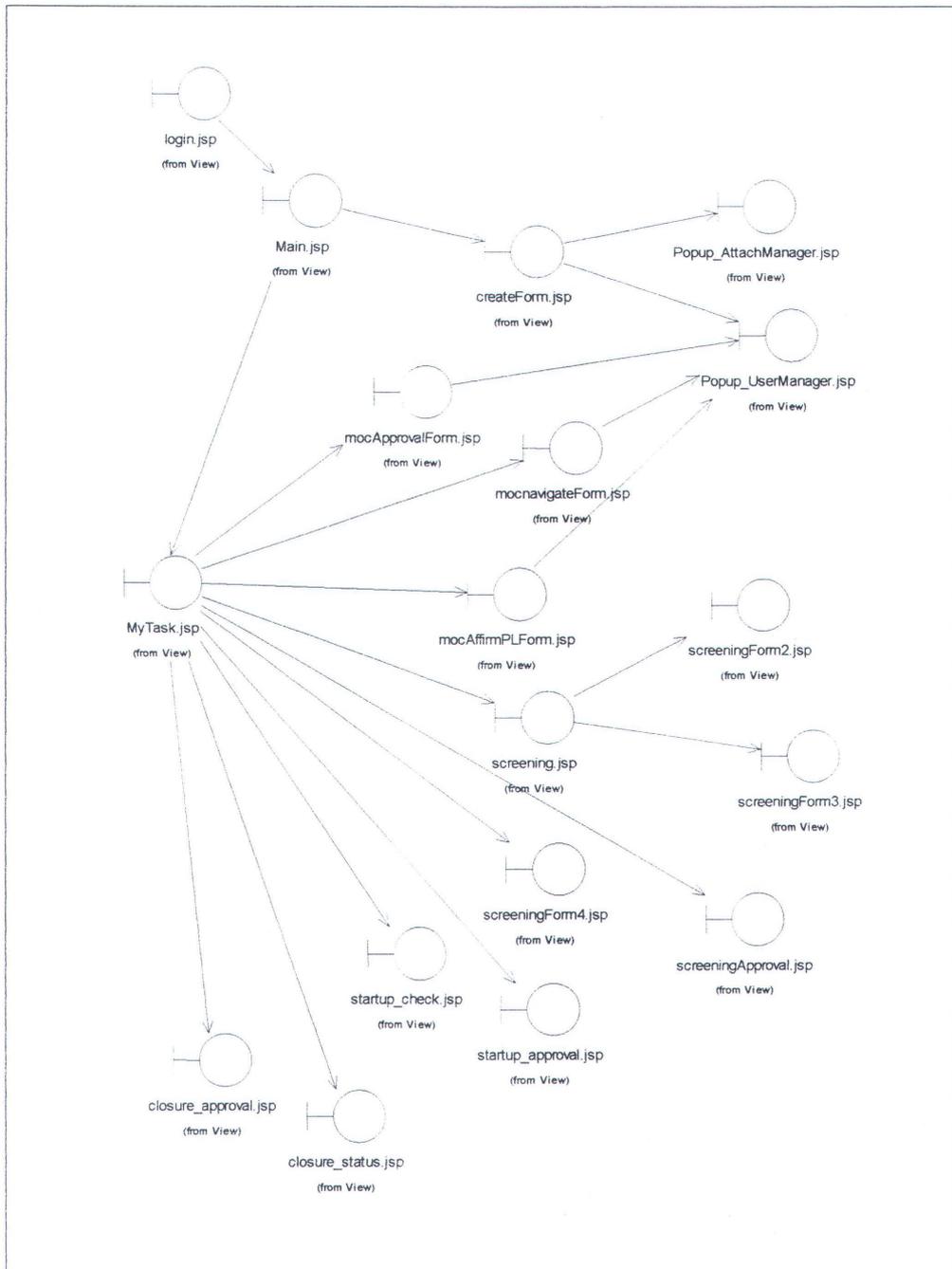
Grafik statistik ini akan menampilkan jumlah proposal MOC yang telah berhasil dilaksanakan pada setiap bulannya.



Gambar 4.39 Grafik Statistik MOC

4.2.8 Perancangan Menu Antarmuka

Sub bab ini membahas mengenai perancangan antarmuka dari perangkat lunak Sistem e-MOC. Perancangan menu ini direpresentasikan dalam bentuk diagram class seperti terlihat pada gambar 4.40. dibawah ini:

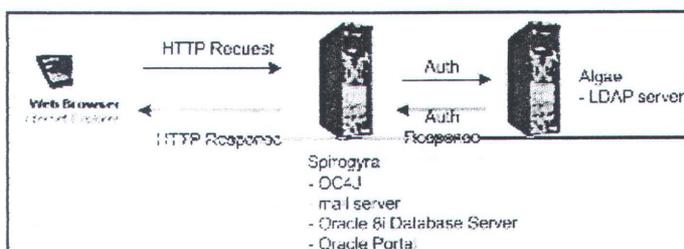


Gambar 4.40 Rancangan Menu Antarmuka

4.3 Pembuatan Perangkat Lunak

Berdasarkan pada rancangan yang telah dibuat, berikut merupakan penjelasan mengenai lingkungan pengembangan sistem yang digunakan. Terdiri dari konfigurasi dan *setting* dari lingkungan pengembangan sistem, implementasi perancangan Model View Controller Pattern yang terkait pula dengan implementasi keamanan sistem, implementasi perancangan data dan implementasi perancangan antarmuka. Sebelum membahas lebih lanjut tentang implementasi dari perancangan yang telah dibuat, akan dijelaskan terlebih dahulu tentang lingkungan dimana semua perancangan sistem tersebut akan diimplementasikan.

Pada Tugas Akhir ini semua perancangan yang telah dibuat akan diimplementasikan pada lingkungan sistem seperti terlihat dalam gambar 4.41. berikut ini.



Gambar 4.41 Lingkungan Implementasi Perancangan

Spesifikasi perangkat keras dari masing-masing server yang digunakan adalah sebagai berikut:

- Spirogyra
 - Duron 1,2 GHz
 - Memori 256 Mbyte, DDR
 - Hardisk 70 GB

Dengan spesifikasi perangkat lunak:

- Sistem Operasi Windows XP Professional
- OC4J versi 2.1
- PostCast Mail server versi 3.0
- Oracle Database Server versi 8.1.7
- Oracle Portal Release 1

- Pancala

- Pentium II 400 MHz
- Memori 384 Mbyte SDRAM
- Hardisk 4 GB

Dengan spesifikasi perangkat lunak:

- Debian Linux
- OpenLDAP
- phpLDAPAdmin

4.3.1 Implementasi Model-View-Controller Pattern

Sebagaimana telah dijelaskan pada bagian perancangan, berikut ini adalah implementasi dari perancangan tersebut.

A. Controller

Controller ini diimplementasikan oleh class *MyFilteringController2.java*. Secara lebih detail, beberapa bagian penting dari Controller ini akan dijelaskan sebagai berikut:

```
public class myFilteringController2 implements Filter {
    //cache for event handler objects
    protected static HashMap evtHandlerExecutors = new HashMap();
    protected static Properties evtHandlerTable =
    Utilities.loadParams2("ControllerEventHandlers");
```

Controller dikembangkan sebagai sebuah filter sehingga dapat masuk ke alur eksekusi semua file .JSP oleh Container. Sebagaimana dijelaskan pada sub bab perancangan, Controller memerlukan beberapa bean lain untuk melaksanakan validasi dan otorisasi akses. Bean-bean tersebut akan disimpan dalam *cache* sehingga Controller diharapkan dapat bekerja lebih cepat karena tidak perlu membuat *instance* baru dari bean-bean tersebut setiap saat. *Cache* ini diimplementasikan dengan menggunakan HashMap.

```
HttpServletRequest httpRequest = (HttpServletRequest)request;
HttpServletResponse httpResponse = (HttpServletResponse)response;
String sessionBadge = (String)httpRequest.getAttribute("session");
```

Controller akan menangkap objek HttpRequest dan HttpResponse yang dikirimkan oleh sistem. Hal ini diperlukan untuk mengidentifikasi *request* yang masuk dan juga untuk meneruskan semua parameter *request* ke halaman yang dimaksud, sehingga tidak ada parameter yang hilang dalam proses.

```
HttpSession session = httpRequest.getSession();
```

Selanjutnya pada potongan script diatas, Controller akan menangkap objek Session dari objek HttpRequest yang diterimanya. Hal ini dilakukan untuk mendapatkan akses ke session dari pengguna saat ini, sehingga dapat dipergunakan untuk pemeriksaan status login pengguna.

```
String evtTrigger = httpRequest.getRequestURI().toLowerCase();
ControllerEventHandler hEvent = null;
If ((session.getAttribute("userid")==null) ||
(session.getAttribute("userid")=="")) {
```

```

logger.log(appname,"user not logged on, redir to login page");
httpResponse.sendRedirect("/moc/sys/login.jsp?referrer="+
    httpRequest.getRequestURI());
} else {

```

Pada potongan kode diatas, Controller akan memeriksa status login seseorang. Apabila ternyata belum login, maka Controller akan segera menampilkan halaman login. Selain itu Controller juga melakukan identifikasi *event* berdasarkan *request* yang diterima.

Sebaliknya apabila ternyata pengguna mempunyai session yang valid, Controller akan mencari bean helper yang bertindak sebagai *event handler* yang sesuai dengan *event* yang ditangkap. Bean helper tersebut dipanggil dengan cara mengambilnya dalam *cache* kemudian dilakukan *casting* ke class induknya. Dengan demikian, semua bean helper tersebut dapat diakses dengan cara yang sama.

Pada kode dibawah ini, tampak bahwa Controller mendelegasikan penanganan *request* ke bean helper.

```

//determine other events
hEvent = (ControllerEventHandler) evtHandlerExecutors.get(evtTrigger);
hEvent.processRequest(httpRequest,httpResponse);

```

Selanjutnya Controller akan memeriksa status *request* yang ditangani helper. Apabila method *isAllRight()* dari helper mengembalikan nilai *false*, maka dapat diartikan bahwa *request* tidak valid. *Request* menjadi tidak valid antara lain karena parameter yang dikirim tidak sesuai atau pengguna tidak mempunyai hak untuk mengakses sumber daya yang dimaksud oleh *event*, misalnya form tertentu.

```

if (hEvent.isAllRight()) {
    chain.doFilter(request,response);
} else {
    logger.log(appname,"unable to pass filter: "+evtTrigger);
}

```

Apabila ternyata valid, maka Controller akan menyerahkan kontrol penanganan *request* kembali ke Container dengan menggunakan perintah `chain.doFilter(request,response)`. Dengan metode ini maka pemakaian *dispatcher* secara terpisah tidak diperlukan.

B. Authorization & Validation Helper

Bean-bean helper ini dikembangkan dengan menurun (*inherit*) dari sebuah class abstrak yang berfungsi sebagai class induk. Class abstrak tersebut didefinisikan sebagai berikut:

```

public abstract class ControllerEventHandler {
    public abstract void processRequest (HttpServletRequest request,
        HttpServletResponse response) ;
    public abstract void redirectRequest (HttpServletResponse response,
        String url) ;
    public abstract boolean isAllRight() ;
}

```

Method *processRequest()* akan didefinisikan ulang oleh class turunan untuk melakukan pemeriksaan terhadap *request*, baik parameter yang dikirim maupun hak akses dari session *request* yang bersangkutan. Method ini harus memberikan nilai *false* jika ada parameter yang tidak valid atau hak akses pengguna ternyata kurang.

Method *isAllRight()* akan dipanggil oleh Controller untuk mengetahui hasil pemeriksaan yang telah dilakukan.

C. View

Pada bagian perancangan telah disebutkan bahwa View disini adalah file-file .JSP yang berfungsi sebagai antarmuka. Seperti pada contoh sebelumnya, view disini adalah file createForm.jsp.

D. Helper

Helper dikembangkan sebagai java bean. Terdapat banyak helper dalam sistem e-MOC, diantaranya adalah *helperMocRecord.java*. Bean ini berfungsi untuk mempermudah view pengajuan proposal MOC (createForm.jsp) untuk menyimpan dan membaca kembali proposal MOC.

E. Auth Helper

Seperti telah disebutkan pada bagian perancangan, Auth Helper melakukan fungsi otentifikasi ke server LDAP. Implementasi dari fungsi otentikasi dilaksanakan oleh bean *MyLDAP.java*, yang akan mengenali apakah seorang pengguna diperbolehkan untuk mengakses sistem e-MOC. Bean ini akan dipanggil dari halaman login, yaitu fungsi *auth()*. Dalam fungsi *auth()*, terdapat beberapa hal penting yang harus diperhatikan. Untuk lebih jelasnya dapat dilihat pada potongan kode berikut ini.

Hal pertama yang diperlukan yaitu membuat sebuah HashTable. HashTable ini adalah format parameter pencarian yang diminta oleh JNDI.

```
Hashtable hashtable = new Hashtable();  
String s2 = "";
```

Potongan kode dibawah ini merupakan *setting* untuk driver dari LDAP dalam JNDI.

```
hashtable.put("java.naming.factory.initial", "com.sun.jndi.ldap.LdapCtxFactory");
```

Sedangkan potongan kode berikut digunakan untuk menentukan dimana server LDAP berada, yaitu di komputer Pancala dengan IP 10.126.13.93.

```
hashtable.put("java.naming.provider.url", "ldap://10.126.13.93");
```

Metode “simple” dipilih dengan maksud agar pada saat pengiriman user id dan password tidak dilakukan enkripsi untuk menghindari ketidakcocokan yang mungkin terjadi akibat implementasi LDAP yang berbeda-beda.

```
hashtable.put("java.naming.security.authentication", "simple");
```

Potongan kode berikut ini menunjukkan susunan atau hirarki dari LDAP, beserta user id dan password yang digunakan untuk dapat masuk ke server LDAP.

```
hashtable.put("java.naming.security.principal",
    "uid=rizki,CN=Users,DC=ad,DC=unocal,DC=com");
hashtable.put("java.naming.security.credentials", "bismillah");
```

Setelah berhasil masuk ke dalam server LDAP, proses selanjutnya adalah mencari dan membandingkan user id yang berasal dari *input*. Jika user id tidak ada, maka fungsi ini akan mengembalikan nilai “INVALID”.

```
try {
    InitialDirContext initialdircontext =
    new InitialDirContext(hashtable);
    String s3 = "(&(objectClass=person)(|(sAMAccountName="+
    .trim()+ ")))";
    String s4 = "CN=Users,dc=ad,dc=unocal,dc=com";
    String as[] = {
        "sAMAccountName", "userPassword"
    };
    SearchControls searchcontrols = new SearchControls();
```

```

searchcontrols.setSearchScope(2);
searchcontrols.setReturningAttributes(as);
NamingEnumeration namingenumeration = initialdircontext.search
(s4, s3, searchcontrols);
if(namingenumeration.hasMore())
{
    s2 = ((SearchResult)namingenumeration.next()).getName() + "," + s4;
} else
    return "INVALID";
initialdircontext.close();
} catch(NamingException namingexception) {
    return "LDAP: Problem getting attribute: " + namingexception;
}

```

Jika ada, maka user id awal akan diganti dengan user id sesuai nilai input dan dilakukan pengecekan ulang dengan user id dan password yang baru. Jika benar, maka fungsi akan mengembalikan nilai "VALID" dan "INVALID" untuk sebaliknya.

```

hashtable.put("java.naming.security.authentication", "simple");
hashtable.put("java.naming.security.principal", String.valueOf(s2));
hashtable.put("java.naming.security.credentials", s1.trim());
try {
    InitialDirContext initialdircontext1 =
        new InitialDirContext(hashtable);
    initialdircontext1.close();
    return "VALID";
} catch(AuthenticationException authenticationexception) {
    return "INVALID";
} catch(NamingException namingexception1) {
    return "ERROR" + namingexception1;
}

```

4.3.2 Implementasi Data

Data dalam sistem e-MOC disimpan dalam schema PRODUCTION dalam basis data. Server basis data yang digunakan adalah Oracle Database Server versi 8.1.7. Selanjutnya tabel-tabel tadi akan dipetakan ke dalam file *ejb-jar.xml* dan *orion-*ejb-jar.xml** berdasarkan abstraksinya. File *orion-*ejb-jar.xml** secara otomatis akan dibuat ketika dilakukan *deploy* terhadap EJB yang telah dibuat.

A. Abstraksi Basis Data

Sebagaimana dijelaskan dalam sub bab perancangan, dalam sistem e-MOC akses ke basis data diabstraksi menggunakan Enterprise Java Bean (EJB) yang terdiri dari file-file *home interface*, *remote interface* dan *enterprise bean class*. Sebagai contoh, untuk tabel MOC2_MSTAREA, berikut adalah pemrograman untuk file *home interface* yang diberi nama *mocMstAreaHome.java*:

```
import java.rmi.RemoteException;
import javax.ejb.*;
import java.lang.*;
public interface mocMstArea extends EJBObject {
    public Integer getAreaId() throws RemoteException;
    public void setAreaId(Integer areaId) throws RemoteException;
    public String getArea() throws RemoteException;
    public void setArea(String area) throws RemoteException;
    public String getMocCoord() throws RemoteException;
    public void setMocCoord(String mocCoord) throws
RemoteException;
}
```

Berikut ini adalah pemrograman *remote interface* yang diberi nama *mocMstArea.java*:

```
import java.rmi.RemoteException;
import javax.ejb.*;
import java.lang.*;
public interface mocMstArea extends EJBObject {
    public Integer getAreaId() throws RemoteException;
    public void setAreaId(Integer areaId) throws RemoteException;
    public String getArea() throws RemoteException;
    public void setArea(String area) throws RemoteException;
    public String getMocCoord() throws RemoteException;
    public void setMocCoord(String mocCoord) throws
RemoteException;
}
```

Sedangkan untuk *implementation bean*, contoh pemrogramannya adalah sebagai berikut:

```
import java.io.Serializable;
import java.util.Iterator;
import java.util.Collection;
```

```

import java.lang.*;
import javax.ejb.CreateException;
import javax.ejb.DuplicateKeyException;
import javax.ejb.EJBException;
import javax.ejb.EntityBean;
import javax.ejb.EntityContext;
import javax.ejb.FinderException;
import javax.ejb.NoSuchEntityException;
import javax.ejb.ObjectNotFoundException;
import javax.ejb.RemoveException;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;
abstract public class mocMstAreaBean implements EntityBean {
    private EntityContext ctx;
    public mocMstAreaBean() {};
    public void setEntityContext(EntityContext ctx) {
        this.ctx = ctx;
    }
    public void unsetEntityContext() {
        this.ctx = null;
    }
    abstract public Integer getAreaId();
    abstract public void setAreaId(Integer areaId);
    abstract public String getArea();
    abstract public void setArea(String area);
    abstract public String getMocCoord();
    abstract public void setMocCoord(String mocCoord);
    public void ejbActivate() {
    }
    public void ejbPassivate() {
    }
    public void ejbLoad() {
    }
    public void ejbStore() {
    }
    public void ejbRemove() throws RemoveException {
    }
    public Integer ejbCreate() throws CreateException {
        return null;
    }
    public void ejbPostCreate() {
    }
}

```

B. `ejb-jar.xml`

Dalam file `ejb-jar.xml` terdapat beberapa hal yang harus dikonfigurasi, diantaranya:

- `<ejb-name>`

Sebagai contoh untuk tabel action, `<ejb-name>` akan dituliskan sebagai berikut:

```
<ejb-name>JUocal.moc.MocMstArea</ejb-name>
```

- `<home>`

Sebagai contoh untuk tabel action, `<home>` akan dituliskan sebagai berikut:

```
<home>com.unocal.moc.ejb.MocMstAreaHome</home>
```

- `<remote>`

Sebagai contoh untuk table action, `<remote>` akan dituliskan sebagai berikut:

```
<remote>com.unocal.moc.ejb.MocMstArea</remote>
```

- `<ejb-class>`

Sebagai contoh untuk tabel action, `<ejb-class>` akan dituliskan sebagai berikut:

```
<ejb-class>com.unocal.moc.ejb.MocMstAreaBean</ejb-class>
```

- `<persistence-type>`

Pada tugas akhir ini yang dipergunakan adalah Container. Oleh karena itu dalam file `ejb-jar`, `<persistence-type>` akan dituliskan sebagai berikut:

```
<persistence-type>Container</persistence-type>
```

- `<primary-key-class>`

Karena primary key dari tabel MOC2_AREA bertipe String, maka `<primary-key-class>` akan dituliskan seperti contoh berikut ini:

```
<prim-key-class>java.lang.String</prim-key-class>
```

- `<reentrant>`

`<reentrant>` di set false untuk setiap entity bean.

```
<reentrant>False</reentrant>
```

- `<abstract-schema-name>`

Untuk abstraksi tabel MOC2_MSTAREA, `<abstract-schema-name>` akan dituliskan sebagai berikut:

```
<abstract-schema-name>MocAreaBean</abstract-schema-name>
```

- `<cmp-field>` dan `<field-name>`

Sebagai contoh pada tabel MOC2_MSTAREA, untuk mengatur penyimpanan bagi atribut action id, `<field-name>` akan berisi public field dari MocMstArea entity bean.

```
<cmp-field>
<field-name>areaId</field-name>
</cmp-field>
```

- `<primkey-field>`

Pada tabel MOC2_MSTAREA, yang menjadi primary key adalah area id, karena itu elemen `<primary-field>` diisi dengan areaId field.

```
<primkey-field>areaId</primkey-field>
```

C. orion-ejb-jar.xml

Setelah semua field dari entity bean yang ada dipetakan ke dalam *ejb-jar.xml*, selanjutnya setiap field tadi dipetakan kembali ke nama fisik dari tabel yang sesungguhnya. *orion-ejb-jar.xml* ini secara otomatis akan dibuat oleh *Container J2EE* dengan deklarasi awal sesuai nilai standar yang ditetapkan oleh

sistem. Untuk itu diperlukan beberapa konfigurasi ulang, terutama pada pemetaan ke tabel yang sesungguhnya.

- `<entity-deployment ...>`

Berisi informasi deployment dari sebuah entity bean. Informasi tersebut antara lain:

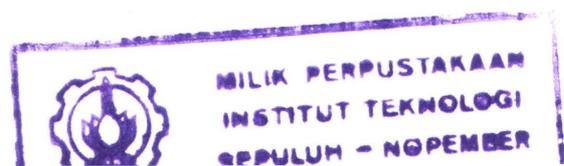
- ❖ **name** – berisi nama dari bean, nama ini sesuai dengan nama bean yang ada di dalam assembly descriptor (file ejb-jar.xml).
- ❖ **location** – nama JNDI dari bean.
- ❖ **wrapper** – Nama dari class Orion wrapper untuk bean ini. (atribut server internal attribute, jangan diubah)
- ❖ **table** – Nama dari tabel dalam basis data jika menggunakan CMP.
- ❖ **data-source** – Nama dari data-source, digunakan jika menggunakan CMP.

Berikut adalah contoh konfigurasi `<entity-deployment ...>` dari MocMstArea bean.

```
<entity-deployment name="JU nocal.moc.MocMstArea"
location="JU nocal.moc.moc.MstArea"
wrapper="MocMstAreaHome_EntityHomeWrapper2" tabel="MOC2_MSTAREA"
data-source="jdbc/PRODUCTION">
```

- `<primkey-mapping>` dan `<cmp-field-mapping..>`

Menentukan pemetaan dari primary key. Field primary key dari MocMstArea bean adalah areaId yang akan dipetakan ke kolom AREA_ID pada tabel MOC2_MSTAREA. Pemetaan areaId field ke AREA_ID kolom



dilakukan pada elemen `<cmp-field-mapping..>` dengan mengisi nama dari field pada *name* dan nama kolom pada *persistence-name*. Dengan demikian

`<primkey-mapping>` akan diatur sebagai berikut.

```
<primkey-mapping>
<cmp-field-mapping
name="areaId" persistence-name="AREA_ID" />
</primkey-mapping>
```

Pemetaan field-field selain primary key dilakukan langsung pada elemen `<cmp-field-mapping..>`. Sebagai contoh pemetaan untuk *actionDesc* berikut ini.

```
<cmp-field-mapping name="areaName" persistence-name="AREA_NAME"/>
```

Setelah semua tabel diabstraksi, selanjutnya file-file abstraksi dapat dipanggil untuk membuat file bean yang berisi proses bisnis dari perangkat lunak. Bean-bean ini selanjutnya dapat dipanggil melalui file-file JSP untuk menampilkan hasil proses kepada pengguna.

4.3.3 Implementasi Keamanan Sistem

Permasalahan keamanan sistem ini akan ditangani oleh Controller, Auth Helper dan antarmuka login. Dimana ketiganya akan menggabungkan fungsi-fungsi yang terdapat dalam Controller, bean MyLDAP, bean-bean helper dan antarmuka untuk login. Sebelum membahas lebih lanjut tentang masalah keamanan ini, akan dijelaskan terlebih dahulu tentang aplikasi Webman.

A. Webman (iSecure)

Webman adalah sebuah aplikasi yang di UNOCAL Indonesia Co. yang berjalan diatas Oracle Portal dan digunakan sebagai suatu sistem yang akan membantu dalam proses otorisasi pengguna. Dengan aplikasi ini, pengguna baru dapat didaftarkan untuk memiliki role dan hak atas suatu aplikasi.

Pada skema Webman tersimpan data-data penting sebagai berikut:

- aplikasi-aplikasi yang ada di UNOCAI Indonesia Co.
- user id pengguna
- role dan *privilege* dari pengguna terhadap aplikasi tertentu

Sistem elektronik lain, dapat menggunakan bean dari Webman untuk kepentingan lain, misalnya pengenalan user id pengguna dari badge.

B. Implementasi keamanan sistem e-MOC

Keamanan sistem haruslah dapat ditangani dengan baik untuk mencegah agar pihak-pihak yang tidak diinginkan dapat mengakses sistem ini. Ketika pengguna mengakses sistem, pertama kali Controller akan melakukan pengecekan terhadap pengguna, apakah sudah login atau belum. Jika pengguna belum melakukan login, Controller akan mengarahkan pengguna ke halaman login untuk melakukan otorisasi pengguna. Halaman ini juga akan menangkap *referrer* dari Controller. *Referrer* ini adalah URL yang tadi berusaha diakses oleh pengguna.

Selanjutnya pada proses otentikasi, halaman login akan memanggil fungsi *auth()* pada bean *MyLdap* dengan parameter user id dan password. Jika parameter yang dikirimkan invalid, maka akan dikembalikan ke halaman login kembali. Jika valid, maka login akan menyimpan session dari pengguna, yaitu berupa badge

pengguna dan redirect ke URL yang ada dalam *referrer*. Dari Controller, proses selanjutnya terjadi di dalam helper otorisasi dan validasi. Bean ini akan membantu Controller untuk mengetahui hak-hak dari tiap pengguna berdasarkan data yang terdapat dalam Webman. Setelah itu, tampilan menu pada bagian antarmuka (View) akan disesuaikan dengan hak-hak tersebut.

4.3.4 Implementasi Antar Muka

Sub-bab ini akan dijelaskan implementasi dari rancangan antarmuka sebagaimana yang telah dijelaskan sebelumnya pada sub-bab 4.2.6. Antar muka sistem ini terdiri dari satu halaman login dan satu halaman utama.

Pada halaman utama terdapat beberapa pilihan menu. Pilihan menu ini dapat berbeda antara pengguna satu dengan pengguna yang lain, tergantung pada hak yang dimiliki pengguna. Secara umum semua pengguna akan memiliki menu-menu sebagai berikut:

- Formulir Pengajuan Proposal MOC (*Create MOC*)
- Daftar tugas dalam sistem e-MOC (*MyTask*)
- Logout

Selanjutnya, tugas-tugas para aktor dalam sistem e-MOC dapat diakses melalui menu *MyTask*, kecuali untuk pengajuan proposal MOC yang sudah disediakan menu tersendiri.



BAB V
UJI COBA PERANGKAT LUNAK

BAB V

UJI COBA PERANGKAT LUNAK

Bab ini membahas tentang hasil uji coba aplikasi e-MOC. Uji coba dilakukan dengan simulasi terhadap proses pengajuan proposal MOC, persetujuan proposal oleh para *approver*, penunjukan Project Leader, pengisian *screening form*, persetujuan terhadap hasil *screening*, pengisian *startup check*, dan pengajuan penutupan proyek perubahan.

Tetapi sebelum membahas hal itu lebih lanjut, akan dilakukan uji coba terhadap aplikasi Webman. Pada tugas akhir ini, aplikasi tersebut dapat di akses melalui alamat URL: <http://10.126.11.10:7777/pls/webman>

5.1 Uji Coba Aplikasi Webman

Aplikasi ini hanya dapat diakses oleh Administrator Webman. Setelah melakukan login, Admin dapat memilih menu *Application*.



The image shows a login form with two input fields and a button. The first field is labeled 'User Id.' and contains the text 'rizkiag'. The second field is labeled 'Password' and contains a series of dots. Below these fields is a button labeled 'LOGIN'.

Gambar 5.1 Login ke Aplikasi Webman

Jika aplikasi sistem e-MOC belum terdaftar di dalam Webman, maka hal pertama yang dilakukan yaitu mendaftarkan aplikasi ini ke dalam Webman

Untuk menambahkan role dari sistem e-MOC, Admin dapat menekan tombol Roles, maka akan tampil menu baru seperti terlihat pada gambar 5.4.

Management of Changes Electronic Approval System					
No	Name	Description	Action		
1	EMOC_LPE	eMOC Lead Process Engineer			
2	EMOC_MOCC	eMOC Coordinator			
3	EMOC_ORG	eMOC Originator			
4	EMOC_PL	eMOC Project Leader			
5	EMOC_PROPAPPR	eMOC Proposal Approver			
6	EMOC_SPL	eMOC Supv of PL			
7	EMOC_SUPT	eMOC Superintendent			
New	OC_CHIENG	eMOC Chief E&C Engineer	Insert New		

Gambar 5.4 Menu Memasukkan Data Role

Untuk memasukkan data role yg baru, pengguna dapat mengetikkan role yang dimaksud pada baris new role yang diinginkan, kemudian menekan tombol Insert New. Role yang baru tadi akan muncul pada baris di atasnya. Misalnya apabila Administrator ingin menentukan hak-hak yang dimiliki oleh Project Leader, maka Admin dapat menekan tombol Detail pada baris 4. Kemudian tampilan berikut akan muncul.

EMOC_PL		
No	Prog.Name	Description
1	EMOC_APPROVAL	Approval form for MOC process
2	EMOC_ATTACHPOPUP	MOC Attachment Popup
3	EMOC_CLOSEUPAPPROVAL	Approval form for closing change project
4	EMOC_CREATE	Create new MOC
5	EMOC_MOC_ASSIGN	MOC Coordinator Assign Project Leader
6	EMOC_MOC_NAVIGATE	MOC Coordinator navigate MOC proposal
7	EMOC_MOC_UPDATE	MOC Coordinator update status
8	EMOC_PL_UPDATESTATUS	PL Update Status
9	EMOC_PLMOCGENAPPROVAL	Chief E&C Eng. Form 263 approval
10	EMOC_PLR200LREAPPROVAL	LRE Screening Form 200 approval
11	EMOC_PLR200PLWAPPROVAL	PL Screening form 200 ask approval
12	EMOC_PLR200PLAPPROVAL	PL Screening Form 200 approval
13	EMOC_PLR400APPROVAL	PL screening form 4 approval
14	EMOC_PLSCREENING2	Project Leader MOC Form 2 entry
15	EMOC_PLSCREENING3	Project Leader MOC Form 3 entry
16	EMOC_PLSCREENING4	Project Leader MOC Form 4 entry
17	EMOC_PLSTARTUP	PL MOC Startup
18	EMOC_PLSTARTUPAPPROVAL	MOC Startup Approval
19	EMOC_PLSTARTUPAPPR	Superintendent approval for MOC startup
20	EMOC_PLSTARTUPPOPUP	MOC Startup Popup

Gambar 5.5 Menentukan Hak Tiap Role

Untuk menentukan hak-hak yang dimiliki, Admin dapat memberikan tanda cek pada check box masing-masing modul. Setelah selesai Admin dapat menekan tombol Update untuk menyimpannya di dalam sistem.

Setelah Role dan hak tiap-tiap role ditentukan, langkah selanjutnya adalah menentukan orang-orang yang berhak atas role dan hak-hak tadi. Untuk mendaftarkan seseorang ke dalam suatu role tertentu, Admin dapat menekan pilihan Menu Users. Jika seseorang ini belum pernah terdaftar dalam Webman, maka Admin dapat memasukkan data dari orang tersebut pada form dibawah ini.

User Id.	dnugroho
User Name.	Dwi Nugroho
Department	ENGINEERING & TECHNOLOGY - EAST KALIMANTAN, INFORMATION SYSTEMS
Badge No.	02491
Email.	dnugroho@balikpapan.unocal.com
Location.	BALIKPAPAN ▾

Gambar 5.6 Form Menambah User Baru

Jika orang tersebut sudah pernah terdaftar dalam Webman, maka dapat dilakukan pencarian terhadap data orang tersebut dengan mengetikkan nama atau user id dari orang yang dimaksud, kemudian menekan tombol Filter. Misalnya jika akan menambahkan role dari sistem e-MOC kepada user “*tutukor*”.

Userid : tutukor	Filter
------------------	--------

Gambar 5.7 Searching Pengguna dalam Webman

Ketika tombol Filter ditekan, maka akan muncul tabel yang berisi daftar pengguna yang dicari. Untuk menampilkan data dari pengguna ini, Admin dapat menekan tombol Edit pada kolom Action.

No	User Id.	User Name	Dept Id	Badge No	Action
1	tutukor	Tutuko Resihadi		01859	Edit Delete

Gambar 5.8 Edit Data Pengguna Webman

Ketika tombol Edit ditekan, data dari *tutukor* akan muncul.

User Identification	
User Id.	tutukor
User Name.	Tutuko Resihadi
Department	ENGINEERING & TECHNOLOGY - EAST KALIMANTAN, INFORMATION SYSTEMS
Badge No.	01859
Email.	tutukor@orion.unocal.com
Location.	BALIKPAPAN ▾

Gambar 5.9 Data Pengguna “tutukor”

Baik untuk pengguna baru atau pengguna lama, untuk mendaftarkan mereka ke dalam suatu role, pada bagian User Roles admin dapat menambahkan role baru bagi pengguna ini, yaitu dengan memilih role yang sesuai berdasarkan lokasinya. Jika “tutukor” akan diberi role sebagai Superintendent saja pada sistem e-MOC, maka Admin dapat memilih “BLP” pada daftar pilihan MOC_SUPT. Untuk lebih jelasnya dapat dilihat pada gambar 5.9.

eMOC	EMOC_CHIENG (none) ▾		
	EMOC_LPE (none) ▾		
	EMOC_MOCC (none) ▾		
	EMOC_ORG (none) ▾		
	EMOC_PL (none) ▾		
	EMOC_PROPAPPR (none) ▾		
	EMOC_SPL (none) ▾		
	EMOC_SUPT BLP ▾		
	(none)		
	ALL		
BLP			
JKT			
			<input type="button" value="UPDATE DATA"/>

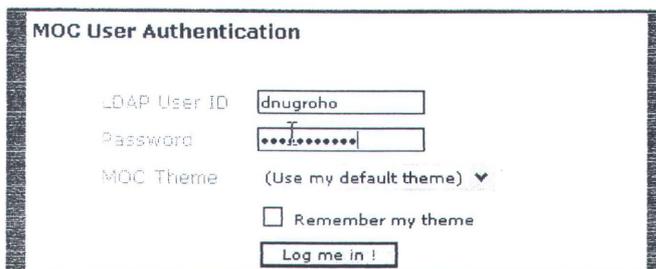
Gambar 5.10 Menentukan Role Pengguna Webman

Setelah tombol “update” ditekan, role akan segera efektif. Ini berarti bahwa pengguna dengan nama user “tutukor” akan berhak mengakses modul-modul yang telah ditentukan oleh Admin seperti terlihat pada gambar 5.5. diatas.

5.2 Uji Coba Sistem e-MOC

Untuk mengakses sistem MOC, setiap pengguna akan diminta untuk melakukan login terlebih dahulu. Pada saat login, sistem akan melakukan pemeriksaan nama *user* dan *password* yang diberikan di server *Active Directory* dengan protokol LDAP.

Selain itu sistem juga akan memeriksa hak dari *user* tersebut di Webman. Informasi tersebut akan disimpan dalam *session* untuk keperluan otorisasi akses di setiap halaman. Gambar 5.11 berikut ini merupakan tampilan antar muka untuk login.



Gambar 5.11 Login ke sistem e-MOC

Pelaksanaan uji coba ini terdiri atas beberapa skenario. Pada beberapa skenario uji coba berikut ini, peran-peran yang akan berpartisipasi adalah sebagai berikut.

Tabel 5.1 Daftar User Percobaan

Peran	Nama User	Login User
Originator	Dwi Nugroho	dnugroho
Supervisor of Originator	Agus M. Manggala	ammanggala
Superintendent	Tutuko Resihadi	tutukor
Manager	Chandra Murti	cepot3

Lead Process Engineer	Adriyan Putra	adriyanp
Project Leader (PL)	Hardiawan H. Triwanda	hardiawanht
Supervisor of PL	Chandra Murti	cepot3
Chief E&C Engineer	Cepot11	Cepot11
MOC Coordinator	Steve Adrianto	sadrianto

5.2.1 Skenario Uji Coba 1

Pada skenario ini, proposal MOC akan diajukan dan akan mendapatkan persetujuan dari para approver sampai selesai. Hal ini dimaksudkan untuk menguji kesesuaian alur persetujuan proses MOC dengan *requirement* yang ada.

A. Pengajuan proposal MOC

Pada proses pengajuan proposal MOC, aktor yang berperan adalah Originator. Dalam e-MOC, semua orang dapat bertindak sebagai Originator. Originator kemudian dapat mengisi formulir sesuai dengan deskripsi perubahan yang akan diajukan. Sebagai pendukung, Originator juga dapat menambahkan beberapa file lain yang akan disertakan sebagai lampiran (*attachment*).

Dalam studi kasus ini, peran Originator akan dijalankan oleh “Dwi Palumito Nugroho”. Untuk itu, sebelumnya pada saat login dimasukkan *user name* “*dnugroho*”. Setelah login berhasil dilaksanakan, secara otomatis sistem akan menampilkan formulir pengajuan proposal MOC (*createForm.jsp*).

Selanjutnya, Originator akan melampirkan dokumen pendukung yang diperlukan untuk mendukung proposal yang diajukan dengan cara menekan tombol “Manage Attachment”. Selanjutnya Originator dapat memilih file yang akan dilampirkan dengan menekan tombol “Browse”.

Create MOC

UNOCAL
Management of Changes

Upload Attachment - Microsoft Internet Explorer

Upload Attachment

Specify the file to attach :

Browse...

Upload Done

January 20, 2004

Yes

(s)

age Attachments

TYPE OF CHANGE New Project Change RIK Non Routine Permanent Temporary

Change duration (weeks) (only for temporary changes)

AMS record (only for temporary changes)

APPROVAL SHEET

ROLE	NAME	POSITION	ACTION
Supervisor of Originator	AGUS MUFNADI MANGGALA	SR. SUPERVISOR, APPLICATION SUPP	

Gambar 5.12 Proposal MOC (Form 1)

Create MOC

UNOCAL
Management of Changes

CHANGE REQUEST

Field: Serang Work Order: 001-aaa-2004 MOC Number: Issued Date: January 12, 2004

Facility: Pengeboran AFKOD: apcece Budgeted: Yes

Originator: DWI RAMLUTHI MUGROHO Title / Position: ANALYST SYSTEM

CHANGE DESCRIPTION: Adding valve

Attachment(s): Softload More

Manage Attachments

TYPE OF CHANGE New Project Change RIK Non Routine Permanent Temporary

Change duration (weeks) (only for temporary changes)

AMS record (only for temporary changes)

APPROVAL SHEET

ROLE	NAME	POSITION	ACTION
Supervisor of Originator	AGUS MUFNADI MANGGALA	SR. SUPERVISOR, APPLICATION SUPP	

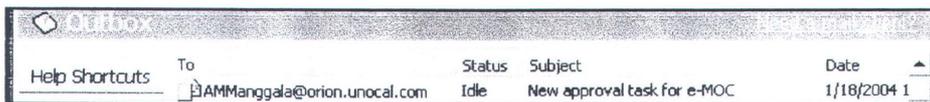
Create MOC >> Start over Cancel

Gambar 5.13 Proposal MOC (Form 1)

Terlihat pada gambar diatas form pengajuan MOC yang sudah diisi. Kemudian Originator dapat menekan tombol *Create MOC* untuk mengirimkan proposal ini ke Supervisornya.

B. Persetujuan dari Supervisor

Setelah Originator menekan tombol *Create MOC*, sistem secara otomatis akan mengirimkan pemberitahuan kepada Supervisor melalui e-mail. E-mail tersebut akan memuat pemberitahuan tentang adanya tugas baru di sistem e-MOC bagi Supervisor yang bersangkutan. Dalam studi kasus ini, *user name* yang menjalankan peran sebagai Supervisor dari Originator adalah “*Ammanggala*”.



Gambar 5.13 E-mail pemberitahuan terhadap Supervisor

Setelah Supervisor melakukan login, Supervisor dapat membuka MyTask untuk melihat tugas-tugas yang menantinya dalam sistem e-MOC. Untuk membuka sebuah tugas, Supervisor dapat melakukan klik pada nomor MOC yang diinginkan.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	January 18 2004	003.012004	Proposed	Supt. of Originator

Gambar 5.14 Daftar tugas (MyTask) dari Supervisor

Selanjutnya terlihat pada form persetujuan proposal di bawah ini bahwa Supervisor dapat mencalonkan seorang Project Leader bagi pelaksanaan proyek perubahan yang terkait dengan proposal ini. Supervisor juga dapat memberikan komentar atau catatan berkaitan dengan proposal yang diajukan. Setelah itu, Supervisor dapat memilih untuk menyetujui (*approve*) atau menolak (*reject*) proposal ini.

MOC Approval		UNOCAL 76 Management of Changes			
CHANGE REQUEST				MOC Number	003.012004
Field	Sarang	Work Order	111.aaa-2004	Issued Date	January 18, 2004
Facility	Pengedoran	AFC/CC	abcde	Budgeted	es
Originator	DW. PATIULTO, NUGROHO	Title / Position	ANAL. ST. S. STB	Attachment(s)	Soft.abcde
CHANGE DESCRIPTION					
Holding Valve					
TYPE OF CHANGE: New Project Permanent					
NOMINATE PROJECT LEADER					
Proposed Project Leader					
REMARKS					
Please type your remarks about this MOC here.		OK lanjut			
APPROVAL SHEET					
ROLE		NAME		ACTION	
Supervisor of Originator		AGUS HURFI ADI MANGGALA		<input checked="" type="checkbox"/> <input type="checkbox"/>	

Gambar 5.15 Formulir Persetujuan Proposal oleh Supervisor

C. MOC Coordinator melakukan forward proposal

Setelah Supervisor melakukan persetujuan, MOC Coordinator akan mendapatkan pemberitahuan tentang adanya sebuah proposal MOC yang baru diajukan. MOC Coordinator kemudian dapat membuka daftar tugasnya (MyTask). Untuk studi kasus ini, *user name* yang akan menjalankan peran sebagai MOC Coordinator adalah “*sadrianto*”.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	January 18, 2004	003.012004	Supv. of Originator Approved	MOC Coordinator

Gambar 5.16 Daftar tugas MOC Coordinator

MOC Coordinator kemudian dapat menentukan aktor-aktor lain yang akan diberikan tanggung jawab untuk melakukan persetujuan terhadap proposal MOC. Untuk memilih nama yang akan diberikan tanggung jawab sebagai pemberi

persetujuan, MOC Coordinator dapat menekan tombol *Select User* (icon bergambar orang).

Gambar 5.17 Formulir Manage MOC

Dalam studi kasus ini, MOC Coordinator akan memberikan peran Superintendent kepada “*Tutuko Resihadi*” (*user name: tutukor*), peran Manager kepada “*Chandra Murti*” (*user name: cepot3*), dan peran LPE kepada “*Adriyan Putra*” (*user name: adriyanp*). Selanjutnya, MOC Coordinator akan melakukan *forward* proposal tersebut kepada masing-masing *approver* yang telah dipilih dengan menekan tombol *Dispatch MOC*.

D. Persetujuan dari Manajemen

Setelah MOC Coordinator melakukan *forward* proposal MOC, alur proses persetujuan akan beralih ke manajemen. Beberapa aktor yang termasuk dalam kategori manajemen ini antara lain Superintendent, Manager, Lead Process Engineer (LPE) dan Other Approver. Proses persetujuan dari masing-masing pemberi persetujuan ini sama, sehingga cukup dibahas salah satu saja.

Berikut ini adalah proses persetujuan yang terjadi pada Superintendent. Untuk itu, sebelumnya *user name* yang dipergunakan untuk login adalah *tutukor*.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	January 18, 2004	003.012004	Supr. of Originator Approved	Superintendent

Gambar 5.18 Daftar tugas Superintendent

Setelah membuka MyTask, Superintendent dapat memberikan persetujuannya pada formulir ini. Selain itu, Superintendent dapat juga memberikan komentar atau memilih seorang calon Project Leader.

CHANGE REQUEST		
Field	Sarang	Work Order
Facility	Pengembangan	AFE-CC
Originator	DIV. PARTITIONING/GRHO	Title - Position
CHANGE DESCRIPTION		MOC Number
MOC No. 003.012004		Issued Date
Project Name		Budgeted
Project Description		Attachment(s)
Project Status		003.012004
Project Location		January 18, 2004
Project Budget		00
NOMINATE PROJECT LEADER		
Proposed Project Leader		
REMARKS		
Please type your remarks about the MOC here		
<< projectna class layout		
APPROVAL SHEET		
ROLE	NAME	ACTION
Supervisor of Originator	HQS. SUPRIADI MANTOGALU	Approved
Superintendent	TUTUKOR RISHADI	Approved
Manager	CHANDRA PURA	Approved
Lead Process Engineer	ADR. H. PUTRA	Approved

Gambar 5.19 Formulir Persetujuan Proposal bagi Superintendent

E. Penetapan Project Leader

Setelah semua *approver* proposal MOC memberikan persetujuannya, MOC Coordinator dapat melakukan penetapan Project Leader. MOC Coordinator berhak memilih Project Leader, baik dari pilihan *approver* maupun orang lain.

Affirm MOC Project Leader

UNOCAL 76
Management of Changes

CHANGE REQUEST

Field	Sarang	Work Order	111.aaa-2004	MOC Number	003.012004
Facility	Pengembangan	AFE/CC	abcedede	Issued Date	January 18, 2004
Originator	DR. PAMULOTO NUGROHO	Title/Position	MANAJ. ST. S. STEFI	Budgeted	es
CHANGE DESCRIPTION				Attachment(s)	
Adding Valve				Soft...stoccd	
TYPE OF CHANGE 111.aaa-2004 Project Permanent					

SELECT PROJECT LEADER

PROPOSED BY	NAME	ACTION
Select Other	HARDIAWAN, H TRIWANDA	 

Gambar 5.20 Penetapan Project Leader

Dalam studi kasus ini, MOC Coordinator akan menentukan Project Leader dari pilihannya sendiri. MOC Coordinator akan menetapkan “Hardiawan H Triwanda” (*user name: hardiawanht*) sebagai Project Leader.

F. Project Leader melaksanakan screening

Setelah Project Leader ditetapkan, sistem akan secara otomatis memberi tahu Project Leader terpilih bahwa ada tugas baru di daftar tugasnya (MyTask). Dalam studi kasus ini, *user name* yang dipergunakan untuk memerankan Project Leader ini adalah *hardiawanht*.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	January 18, 2004	003.012004	Management Approved	Project Leader

Gambar 5.21 Daftar tugas bagi Project Leader

Project Leader kemudian melaksanakan analisis terhadap rencana proyek perubahan yang diusulkan oleh proposal MOC. Hasil analisis tersebut dituangkan

dalam Hazard Analysis Screening Form (MOC Form 2) dan MOC Plan (MOC Form 3). Pelaksanaan analisis ini sepenuhnya diserahkan kepada Project Leader yang bersangkutan. Berikut ini adalah contoh potongan dari formulir-formulir tersebut.

MOC Hazard Analysis		
UNOCAL 76 Management of Changes		
		MOC no. 003.012004
FORM 2 : HAZARD ANALYSIS SCREENING FORM		
<i>Instruction:</i> Hazard analysis should be performed if any screening questions are answered "Yes". The potential for loss including the impact on worker and public safety is considered minimal and a Hazard Analysis need not be performed if all questions are answered "No".		
FOR THE PROPOSED WORK	Y/N	REMARKS
1. Should a Hazard Analysis, Risk Assessment be performed as a precautionary measure because regulatory agencies, the local community, or others may perceive that an unacceptable hazard to the public or environment exists?	Yes <input checked="" type="checkbox"/>	Perlu Hazard Analysis
2. Is a critical facility, process or piece of equipment involved (i.e. those that could result in significant harm to people, property or the environment, and associated economic losses, caused by the materials, locations or operating conditions involved)?	No <input type="checkbox"/>	
3. Is the system considered complex (i.e. involves a large number of components or interacts with multiple or altered)?	Yes <input checked="" type="checkbox"/>	Complex System
4. Is a safeguarding system, critical control or communications system added, removed, bypassed or altered?	No <input type="checkbox"/>	
5. Is a new or unfamiliar technology, process or piece of equipment involved?	No <input type="checkbox"/>	
6. Is there a significant increase in the inventory of toxic, flammable, or reactive materials, or is there an alteration in the manner in which these materials are handled, processed, stored, or disposed of?	No <input type="checkbox"/>	
7. Is there a potential impact on separate but interrelated upstream or downstream processes or facilities?	No <input type="checkbox"/>	
8. Is a critical procedure involved or required (i.e. those that could result in significant harm to people, property or the environment, and associated economic losses, caused by the materials, locations or operating condition involved)?	No <input type="checkbox"/>	
9. Is elaborate Personal Protective Equipment or decontamination procedures required to perform the work?	No <input type="checkbox"/>	

Gambar 5.22 HA Screening Form (MOC Form 2)

MOC Plan

UNOCAL 76
Management of Changes

MOC no. 003.012004

FORM 3 : MOC PLAN

<input checked="" type="checkbox"/> PFD	<input checked="" type="checkbox"/> Electrical 1-Line	<input checked="" type="checkbox"/> Operating Procedure
<input checked="" type="checkbox"/> P&ID	<input checked="" type="checkbox"/> Metering Schematic	<input checked="" type="checkbox"/> Maintenance Procedure
<input checked="" type="checkbox"/> SAFE Chart	<input checked="" type="checkbox"/> Control Block Diagram	<input checked="" type="checkbox"/> Pre-Commissioning Procedure
DESIGN	<input checked="" type="checkbox"/> Electrical Area Classification	<input checked="" type="checkbox"/> Ref. Standards
<input checked="" type="checkbox"/> Layouts	<input checked="" type="checkbox"/> Fire and Gas Detection	<input checked="" type="checkbox"/> Ref. UO Specs.
<input checked="" type="checkbox"/> Structural	<input checked="" type="checkbox"/> Safe Operating Limits	<input checked="" type="checkbox"/> Other
<input checked="" type="checkbox"/> Details	<input checked="" type="checkbox"/> Constructability	<input checked="" type="checkbox"/> Local U.K. U.P.U.
<input checked="" type="checkbox"/> Sand and Fluid Production	<input checked="" type="checkbox"/> Field Based Reviewed	<input checked="" type="checkbox"/> Others
REVIEW	<input checked="" type="checkbox"/> Health, Medical, Environment, MSDS	
<input checked="" type="checkbox"/> API 14-C		
<input checked="" type="checkbox"/> Peer Review		
<input checked="" type="checkbox"/> Project information prepared (PFD, P&ID, layout, procedures, etc.)	<input checked="" type="checkbox"/> Engineers response to recommendations / resources management approval	
HAZARD ANALYSIS	<input checked="" type="checkbox"/> Hazards Analysis team formed (leader, scribe, field representation, etc.)	<input checked="" type="checkbox"/> Develop Action Item Table for monitoring (input into AIMS)
<input checked="" type="checkbox"/> Hazard Analysis conducted/recommendations identified	<input checked="" type="checkbox"/> Complete all Action Items (close in AIMS)	

Gambar 6.23 MOC Plan

Project Leader dapat menyimpan hasil pengisian formulirnya dengan menekan tombol *Save*. Setelah semua proses *screening* selesai, Project Leader dapat mengirimkan kedua formulir ini ke Supervisor-nya dengan menekan tombol *Submit for Supervisor Approval*. Dalam studi kasus ini, kebetulan Supervisor dari Project Leader ini adalah “Chandra Murti” (*user name: cepot3*) juga berperan sebagai Manager.

G. Persetujuan Terhadap Hasil Screening

Persetujuan terhadap hasil *screening* dilakukan oleh tiga peran masing-masing secara berurutan, yaitu Supervisor dari Project Leader, Lead Process Engineer (LPE) dan Chief E&C Engineer (dalam studi kasus ini diperankan oleh *user name “cepot11”*). Proses persetujuan oleh ketiganya sangat mirip dari sisi

pengguna, sehingga dalam studi kasus ini akan dibahas salah satu saja, yaitu untuk Supervisor dari Originator.

Setelah Project Leader mengirimkan hasil *screening* ke Supervisor-nya, maka sistem akan segera memasukkan sebuah tugas baru dalam MyTask dari Supervisor dari Project Leader. Untuk melaksanakan uji coba, maka dilakukan login dengan menggunakan *user name* “*cepot3*” ke dalam sistem.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	Feb 27 2014	003 012004	Screening in progress	Supv of PL

Gambar 6.24 Daftar tugas bagi Supervisor dari Project Leader

Supervisor dari Project Leader kemudian melakukan klik pada nomor MOC untuk menampilkan formulir persetujuan terhadap hasil *screening*. Pada formulir ini Supervisor dapat melihat hasil *screening* di form 2 dan 3 dengan cara membuka *link* yang telah disediakan di halaman persetujuan.

Setelah memeriksa hasil *screening*, Supervisor dapat memberikan persetujuan terhadap hasil *screening* tersebut. Supervisor dapat memberikan komentar atau catatan sesuai keperluan.



MOC Screening

UNOCAL 76
Management of Changes

CHANGE REQUEST

Field	Unit 10	Work Order	101010101	MOC Number	101010101
Facility	Refinery	A/E/C/C	General	Issued Date	10/10/2010
Originator	Dr. P. M. T. N. S. P. H. I.	Title	Position	Budgeted	Yes
CHANGE DESCRIPTION				Attachments	
Title: 101010101				101010101	

Screenings:

Remarks:

REMARKS

Remarks:

Submit | Reset

Gambar 6.25 Persetujuan terhadap Form 2 dan 3

H. Pengisian MOC Form 4

Setelah MOC Form 2 dan 3 disetujui, Project Leader dapat melakukan analisis *Implementation Hazard Screening* yang bertujuan untuk mengidentifikasi masalah dan risiko yang mungkin timbul selama proyek implementasi perubahan berlangsung. Hasil analisis tersebut akan dituangkan dalam MOC Form 4 (*Implementation Hazard Screening*).

Setelah MOC Form 4 selesai diisi, Project Leader akan mengirimkan hasil *screening* tersebut kepada Supervisor-nya dengan cara menekan tombol *Submit*.

J. Pengisian Startup check

Setelah MOC Form 4 disetujui, maka Project Leader akan mengisi formulir *Startup check*. Formulir ini dipergunakan untuk mengidentifikasi persiapan Project Leader sebelum proyek perubahan benar-benar dilaksanakan. Setelah diisi, Project Leader akan melakukan klik pada tombol *Proceed Change Work*. Sistem kemudian akan mengirimkannya ke Superintendent untuk diberikan persetujuan.

MOC Pre-Startup Check		
UNOCAL 76 Management of Changes		
MOC No. 003.012004		
NO	THE FOLLOWING ITEMS MUST BE CHECKED	DONE
1	General Post-Construction Inspection has been performed by Field Supervisor and Project Leader in charge.	Yes
2	All software, programs and computer machines have been checked for correctness and screened for free from viruses.	Yes
3	Process simulation of all software and programs have been performed and corrected.	No
4	Fabrication and installation are in compliance with approved drawings, flow charts, SAFE Chart and Specifications.	No
5	Non-Conforming Records (NCR) have been followed up and closed.	No
6	Non-Destructive Test (NDT) have been completed and free from outstanding repair.	No
7	All other required Testing have been performed and acceptable (Load Test etc).	No
8	All Piping and Equipment have been pressure tested adequately.	No
9	Operating, Maintenance Procedures and P & ID have been updated and available on site.	No
10	All affected personnel and parties have been trained and notified.	Yes
11	All protective shipping brackets and packing have been removed from instrumentation and equipment.	No
12	Utility, fire fighting and personnel safety, equipment checked and functional.	No
13	All lockout and Tag out have been removed.	No
14	All Safety and Control devices have been set and installed in accordance with API RP 14C and any other Regulatory requirement.	No
15	Requirement of Hazard Analysis has been performed and Recommendation have been followed up.	No
16	Electrical system and devices have been checked and meet electrical Codes and Area Classification.	No
17	Check Valves, Check Valve Flappers and Strainers have been installed correctly in accordance with flow direction.	Yes
18	Normally Closed (NOC), Normally Opened (NO) and Locked Open (LO) Block Valves have been identified and checked.	No
19	PSV, PSH, PSL, LSH, LSL and TS have been tested and calibrated.	No
20	Internal of Tanks, Compressor Capsule, Pressure Vessels and Piping have been checked for correct installation and free from dirt or foreign metals.	Yes
21	Prime Movers Rotation have been checked.	No
22	Prime Movers Coupling have been Connected and safety guard installed.	No
23	Alignment and Lubrication have been checked on all Rotating Equipment.	No

Gambar 6.28 Startup Check Form

K. Persetujuan terhadap hasil Startup check

Superintendent, dalam studi kasus ini diperankan oleh *user name* “tutukor”, kemudian dapat memeriksa hasil *startup check* yang dilakukan oleh

M. Persetujuan terhadap Closure status

Selanjutnya, LPE dan Chief E&C Engineer akan mendapatkan pemberitahuan dari sistem tentang penutupan proyek perubahan. Setelah kedua peran ini memberikan persetujuan, maka proyek perubahan dinyatakan selesai.

5.2.2 Skenario Uji Coba 2

Pada skenario ini, proposal MOC akan diajukan dan akan mendapatkan persetujuan dari para approver, namun akan mendapatkan penolakan sehingga dibatalkan.

Alur awal dari pengajuan proposal MOC ini sama dengan sebelumnya, namun akan terhenti pada persetujuan manajemen karena LPE memberikan penolakan.

A. Pengajuan proposal MOC

Originator kemudian dapat mengisi formulir sesuai dengan deskripsi perubahan yang akan diajukan. Terlihat pada gambar di bawah ini form pengajuan MOC yang sudah diisi. Kemudian Originator dapat menekan tombol *Create MOC* untuk mengirimkan proposal ini ke Supervisornya.

UNOCAL 76
Management of Changes

CHANGE REQUEST

MOC Number: 000.012004

Field: Serang Work Order: 111.aaa-2004 Issued Date: January 18, 2004
 Facility: Pengeboran AFE/CC: abcdefg Budgeted: Yes
 Originator: DWI PARTI LUTU NUGROHO Title-Position: ANALYST SYSTEM

CHANGE DESCRIPTION: Adding valve

Attachment(s): SoftLat.000

New Project Change Routine
 Permanent Temporary
 Change duration in weeks: apply for temporary changes
 AFS record: apply for temporary changes

APPROVAL SHEET

ROLE	NAME	POSITION	ACTION
Supervisor of Originator	AGUS TRUFAD MANGGALA	SR SUPERVISOR APPLICATION SUP	<input type="button" value="Approve"/> <input type="button" value="Reject"/>

Gambar 5.30 Proposal MOC (Form 1)

B. Persetujuan dari Supervisor

Setelah Originator menekan tombol *Create MOC*, sistem secara otomatis akan mengirimkan pemberitahuan kepada Supervisor melalui e-mail.

Dalam uji coba ini Supervisor akan memilih untuk menyetujui (*approve*) proposal ini.

UNOCAL 76
Management of Changes

CHANGE REQUEST

MOC Number: 000.012004

Field: Serang Work Order: 111.aaa-2004 Issued Date: January 18, 2004
 Facility: Pengeboran AFE/CC: abcdefg Budgeted: ea
 Originator: DWI PARTI LUTU NUGROHO Title-Position: ANALYST SYSTEM

CHANGE DESCRIPTION: Adding valve

Attachment(s): SoftLat.000

New Project Permanent

NOMINATE PROJECT LEADER

Proposed Project Leader:

REMARKS

Please type your remarks about the MOC here:

APPROVAL SHEET

ROLE	NAME	ACTION
Supervisor of Originator	AGUS TRUFAD MANGGALA	<input checked="" type="checkbox"/> <input type="checkbox"/>

Gambar 5.31 Formulir Persetujuan Proposal oleh Supervisor

C. MOC Coordinator melakukan forward proposal

MOC Coordinator kemudian dapat menentukan aktor-aktor lain yang akan diberikan tanggung jawab untuk melakukan persetujuan terhadap proposal MOC. Selanjutnya, MOC Coordinator akan melakukan *forward* proposal tersebut kepada masing-masing *approver* yang telah dipilih dengan menekan tombol *Dispatch MOC*.

Gambar 5.32 Formulir Manage MOC

D. Penolakan dari Manajemen

Pada skenario ini, LPE akan memberikan penolakan sehingga proposal MOC harus dibatalkan.

ROLE	NAME	ACTION
Supervisor of Originator	AGUS MUFNADI MANGGALA	Approved
Superintendent	TUTUKO RESHADI	
Manager	Chandra Murti	
Lead Process Engineer	ADRIAN PUTRA	<input checked="" type="checkbox"/>

Gambar 4.33 Penolakan Proposal MOC oleh LPE

5.2.3 Skenario Uji Coba 3

Secara umum, skenario ini berjalan mirip dengan skenario-skenario sebelumnya. Pada skenario ini, proposal MOC akan diajukan dan akan mendapatkan persetujuan dari para approver. Selanjutnya, Project Leader akan mengajukan persetujuan atas hasil *screening* Form 2 dan 3 dari atasannya. Dalam tahap ini, Supervisor of PL akan memberikan penolakan sehingga proses *screening* harus diulang.

A. Pengajuan proposal MOC

Originator kemudian dapat mengisi formulir sesuai dengan deskripsi perubahan yang akan diajukan. Terlihat pada gambar di bawah ini form pengajuan MOC yang sudah diisi. Kemudian Originator dapat menekan tombol *Create MOC* untuk mengirimkan proposal ini ke Supervisornya.

CHANGE REQUEST		MOC Number	
Field	Serang	Work Order: 001-aaa-2004	
Facility	Pengebora	AFEMOC: acecece	
Originator	DWI RAMLUTIO	Title/Position: ANALYST SYSTEM	
CHANGE DESCRIPTION		Attachment(s)	
Adding valve		SoftList.doc	
TYPE OF CHANGE		More	
<input checked="" type="checkbox"/> New Project <input type="checkbox"/> Change <input type="checkbox"/> RK Non Routine <input checked="" type="checkbox"/> Permanent <input type="checkbox"/> Temporary		Manage Attachments	
Change duration (weeks)	only for temporary changes		
AFIS record	only for temporary changes		
APPROVAL SHEET			
ROLE	NAME	POSITION	ACTION
Supervisor of Originator	AGUS MURNADI MANGSALA	SR SUPERVISOR APPLICATION SUPP	[User Icon]

Gambar 5.34 Proposal MOC (Form 1)

B. Persetujuan dari Supervisor

Setelah Originator menekan tombol *Create MOC*, sistem secara otomatis akan mengirimkan pemberitahuan kepada Supervisor melalui e-mail.

Dalam uji coba ini Supervisor akan memilih untuk menyetujui (*approve*) proposal ini.

MOC Approval			
UNOCAL 76 Management of Changes			
CHANGE REQUEST		MOC Number	002.012004
Field	Sarang	Work Order	111.aaa-2004
Facility	Pengedaraan	AFE/CC	abcedede
Originator	DIV. PERMINTAAN/UGRO-00	Title / Position	AKAL STIS STEI
CHANGE DESCRIPTION	Meningkatkan		Attachment(s) Softlist.doc
TYPE OF CHANGE	New Project Permanent		
NOMINATE PROJECT LEADER			
Proposed Project Leader			
REMARKS			
Please type your remarks about this MOC here	<input data-bbox="413 1038 944 1079" type="text" value="OK"/>		
APPROVAL SHEET			
ROLE	NAME	ACTION	
Supervisor of Originator	AGUS MURNIADI MANGGALA	<input checked="" type="checkbox"/> <input type="checkbox"/>	

Gambar 5.35 Formulir Persetujuan Proposal oleh Supervisor

C. MOC Coordinator melakukan forward proposal

MOC Coordinator kemudian dapat menentukan aktor-aktor lain yang akan diberikan tanggung jawab untuk melakukan persetujuan terhadap proposal MOC. Selanjutnya, MOC Coordinator akan melakukan *forward* proposal tersebut kepada masing-masing *approver* yang telah dipilih dengan menekan tombol *Dispatch MOC*.

Manage MOC		UNOCAL 			
Management of Changes					
CHANGE REQUEST			MOC Number	000.012004	
Field	Sarang	Work Order	111.aaa-2004	Issued Date	January 18, 2004
Facility	Pengedraan	AFC/CC	abcdece	Budgeted	53
Originator	DIV. PAKLIT/TO/UGRO/0	Title - Position	AKAL ST. S STEI	Attachment(s)	5001.5000
CHANGE DESCRIPTION			Attachment(s)		
Adding Valve			5001.5000		
<input type="checkbox"/> PE OF CHANGE			New Project Permanent		
SET APPROVERS					
ROLE	NAME		ACTION		
Supervisor of Originator	AGUS TRIWANDA/HT/UGRO/0				
Superintendent	TUTUKO RESH/0				
Manager	Chandra Hurd				
Lead Process Engineer	ADRI AN PUTRA				
Other Approver 1					
Other Approver 2					
Other Approver 3					
			Dispatch MOC to		Cancel

Gambar 5.36 Formulir Manage MOC

D. Persetujuan dari Manajemen

Dalam skenario ini, semua approver memberikan persetujuan sehingga proposal MOC dapat dilanjutkan ke tahap pelaksanaan.

E. Penetapan Project Leader

Dalam skenario ini, MOC Coordinator menetapkan “Hardiawan H Triwanda” (*user name: hardiawanht*) sebagai Project Leader.

Affirm MOC Project Leader		UNOCAL 			
Management of Changes					
CHANGE REQUEST			MOC Number	000.012004	
Field	Sarang	Work Order	111.aaa-2004	Issued Date	January 18, 2004
Facility	Pengedraan	AFC/CC	abcdece	Budgeted	53
Originator	DIV. PAKLIT/TO/UGRO/0	Title - Position	AKAL ST. S STEI	Attachment(s)	5001.5000
CHANGE DESCRIPTION			Attachment(s)		
Adding Valve			5001.5000		
<input type="checkbox"/> PE OF CHANGE			New Project Permanent		
SELECT PROJECT LEADER					
PROPOSED BY	NAME		ACTION		
Select Other	HARDIAWAN/HT/UGRO/0				
			Assign PL		Cancel

Gambar 5.37 Penetapan Project Leader

F. Project Leader melaksanakan screening

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	January 18, 2004	003.012004	Management Approved	Project Leader

Gambar 5.38 Daftar tugas bagi Project Leader

Project Leader kemudian melaksanakan analisis terhadap rencana proyek perubahan yang diusulkan oleh proposal MOC. Setelah semua proses *screening* selesai, Project Leader dapat mengirimkan hasil screening ke Supervisor-nya dengan menekan tombol *Submit for Supervisor Approval*.

Terlihat bahwa setelah tombol ini ditekan, isi daftar tugas dari Project Leader menjadi kosong kembali.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role

Gambar 5.39 Daftar tugas bagi Project Leader setelah Submit ditekan

G. Persetujuan Terhadap Hasil Screening

Selanjutnya, dalam skenario ini Supervisor of PL akan memberikan penolakan terhadap hasil *screening* yang diajukan. Terlihat pada gambar berikut ini Supervisor of PL memberikan komentar dan kemudian menekan tombol *Reject* untuk memberikan penolakan.

CHANGE REQUEST				MOC Number	001 022004
Field	Serang	Work Order	1111.aaa-2004	Issued Date	January 18, 2004
Facility	Pengeboran	AFE/CC	abecede	Budgeted	Yes
Originator	DWM PAMILITO NUGROHO		Title : Position	ANALYST, SYSTEM	
CHANGE DESCRIPTION Adding valve				Attachment(s)	
TYPE OF CHANGE : New Project, Permanent					
Screenings: Screening shall be repeated if any change in work scope. Project Leader shall arrange MOC Plan as necessary and applicable. Such MOC Plan generated by Project Leader shall be approved by Supervisor.					
FORM 2			FORM 3		
REMARKS Please type your remarks about this MOC here.					
Poin 2 terlewat di form 3 dan berisiko tinggi.					
<input type="button" value="Approve"/> <input type="button" value="Reject"/>					

Gambar 5.40 Penolakan hasil screening oleh Supervisor of PL

H. Project Leader melaksanakan screening ulang

Setelah Supervisor of Project Leader memberika penolakan, alur proses kembali ke Project Leader. Project :Leader akan mendapatkan pemberitahuan melalui e-mail tentang hal tersebut.

Dalam daftar tugas Project Leader, MOC dengan ID 003.012004 kembali muncul sehingga Project Leader dapat melakukan *screening* ulang.

My Task				
No	Preparation Date	MOC Number	MOC Status	My Role
1	January 18 2004	003.012004	Management Approved	Project Leader

Gambar 5.38 Daftar tugas Project Leader setelah penolakan



BAB VI

PENUTUP

BAB VI

PENUTUP

Bab ini berisi kesimpulan dari tugas akhir yang dibuat berdasarkan uji coba yang telah dilakukan. Selain itu disertakan pula kemungkinan yang dapat dilakukan untuk pengembangan lebih lanjut dari tugas akhir ini.

6.1 Kesimpulan

Berdasarkan uji coba pada perangkat lunak sistem e-MOC dalam tugas akhir ini, dapat disimpulkan beberapa hal sebagai berikut:

1. Penggunaan *pattern* MVC membantu dalam proses desain pemrograman perangkat lunak pada tugas akhir ini, terutama dalam identifikasi komponen-komponen yang diperlukan dan menyusun interaksi antar komponen dalam sistem, sehingga pengembangan sistem dapat lebih terarah.
2. Perangkat lunak sistem e-MOC telah berhasil dibangun sesuai dengan kebutuhan yang telah didefinisikan sebelumnya.

6.2 Kemungkinan Pengembangan

Sistem ini dapat dikembangkan lebih lanjut agar dapat menangani sistem persetujuan serupa secara generik. Dengan demikian, sistem ini dapat dengan mudah dikonfigurasi agar dapat menangani keperluan lain seperti sistem

persetujuan pengadaan barang, sistem persetujuan perubahan jabatan karyawan, dan sebagainya.



DAFTAR PUSTAKA

DAFTAR PUSTAKA

- [RFK-2001] Rahim Adatia, Faiz Arni, Kyle Gabhart, Jhon Griffin, Matjaz B Juric, Jeremiah Lott, Tim McAllister, Aaron Mulder, Nathan Nagarajan, Daniel O'Connor, Ted Osborn, PG Sarang, Andre Tost, Dave Young, and Craig A Berry, "Professional EJB", *Programmer to Programmer*, Wrox Press Ltd., 2001.
- [CJM-2002] Craig A. Berry, John Cornell, Matjaz B. Juric, Meeraj Moidoo Kunnumpurath, Nadia Nashi, Sasha Romanosky, "J2EE Design Pattern Applied", *Real World Development with Pattern Frameworks*, Wrox Press Ltd., 2002.
- [IDS-2002] Inderjeet Singh, Beth Stearns, Mark Jhonson, and the Enterprise Team, "Designing Enterprise Applications with the J2EE™ Platform, 2nd Edition", *The Java™ Series*, Addison-Wesley, 2002.
- [JTS-2002] John Bell, Tony Loton, Subrahmanyam Allamajaru, Sam Dalton, Simon Brown, Andrew Harbourne-Thomas, Bjarki holm, Meeraj Moidoo Kunnumpurath, Sing Li, "Professional Java Servlets 2.3", Wrox Press, 2002
- [OPL-2003] "OpenLDAP 2.1 Administrator's Guide", *The OpenLDAP Project*, Available from: <http://www.openldap.org>, 10 Januari 2003.
- [UIC-2001] "Management of Changes(MOC) User Guide", Unocal Indonesia Co., 2001
- [MOC-2003] "e-MOC Process – Attachment 5A", Unocal Indonesia Co., 2003

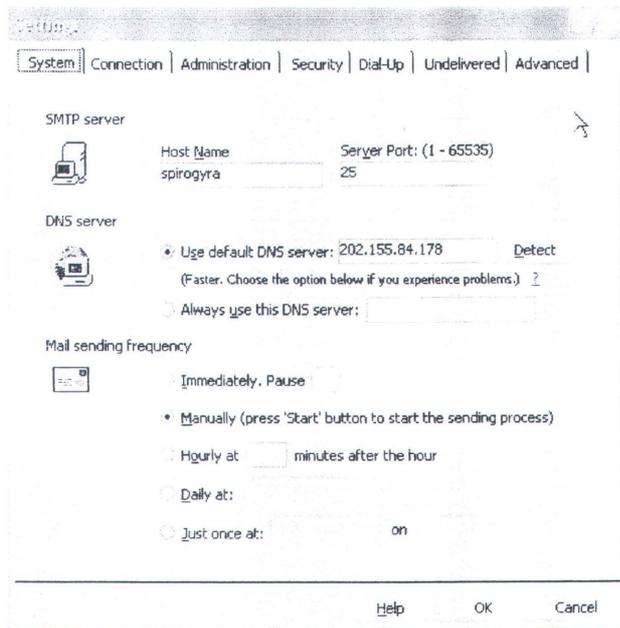


LAMPIRAN

LAMPIRAN

A. Konfigurasi Mail Server

Server email yang digunakan adalah PostCast Mail Server. Berikut adalah konfigurasi dari server tersebut.



Gambar B.1 Konfigurasi Mail Server

B. Konfigurasi OC4J

Berikut ini adalah beberapa file konfigurasi untuk OC4J.

1. *application.xml*

Berikut ini adalah konfigurasi untuk *application.xml* OC4J.

```

<orion-application autocreate-tables="false"
  default-data-source="jdbc/production">
  <web-module id="defaultWebApp" path="../default-web-app" />

  <persistence path="../persistence" />

  <!-- Path to the libraries that are installed on this server.
  These will accesible for the servlets, EJBs etc -->
  <library path="../lib" />

  <principals path="./principals.xml" />

  <log>
    <file path="../log/global-application.log" />
  </log>

  <data-sources path="data-sources.xml" />
  <namespace-access>
    <read-access>
      <namespace-resource root="">
        <security-role-mapping>
          <group name="administrators" />
        </security-role-mapping>
      </namespace-resource>
    </read-access>
    <write-access>
      <namespace-resource root="">
        <security-role-mapping>
          <group name="administrators" />
        </security-role-mapping>
      </namespace-resource>
    </write-access>
  </namespace-access>
</orion-application>

```

Konfigurasi tersebut berlaku secara global dalam instalasi OC4J. Terdapat beberapa hal penting yang harus diperhatikan. Potongan kode berikut menunjukkan setting OC4J untuk tidak membuat tabel-tabel baru yang belum ada dalam basis data secara otomatis.



```
<orion-application autocreate-tables="false"  
default-data-source="jdbc/production">
```

Untuk *default* koneksi basis data melalui JDBC bagi semua aplikasi di OC4J, terlihat pada setting dari **default-data-source**.

2. *server.xml*

File ini dipergunakan untuk mendaftarkan nama dan path dari aplikasi-aplikasi J2EE yang menggunakan OC4J ini. Setting utama yang perlu diperhatikan yaitu pada *application name* dan *path*. Sebagai contoh, file-file aplikasi sistem e-MOC diletakkan pada **e:\tea\mocproject\JUnocal\applications\moc** dengan nama “moc”, maka pada file *server.xml* ini dituliskan seperti pada potongan kode berikut.

```
<application name="moc" path="e:\tea\mocproject\JUnocal\applications\moc"  
auto-start="true" />
```

auto-start diberi nilai “true” agar setiap kali terjadi perubahan pada file *application.xml* yang berada dalam folder WEB-INF, OC4J akan melakukan start-up secara otomatis.

Berikut ini adalah kode pemrograman file *server.xml* secara lengkap.

```

<application-server
  application-directory="../applications"
  deployment-directory="../application-deployments"

  <rmi-config path="../rmi.xml" />
  <!-- JMS-server config link, uncomment to activate the JMS service -->
  <jms-config path="../jms.xml" />
  <log>
    <file path="../log/server.log" />
  </log>

  <global-application name="default" path="application.xml" />

  <global-web-app-config path="global-web-application.xml" />
  <!-- <web-site path="../secure-web-site.xml" /> -->
  <web-site path="../default-web-site.xml" />

  <application name="moc" path="e:\tea\mocproject\JUnocal\applications\moc"
  auto-start="true" />

  <application name="leave" path="e:\tea\mocproject\JUnocal\applications\leave"
  auto-start="true" />

  <application name="iinkecil"
  path="e:\tea\mocproject\JUnocal\applications\iinkecil" auto-start="true" />

  <!-- Uncomment the following, to deploy these apps.
  <application name="ejbsamples" path="../demo/ejb" />

  <application name="news" path="../applications/news.ear" />
  <application name="logger"
  path="../demo/messagelogger.ear" />
  <application name="petstore"
  path="../applications/estore-patched.ear" />
  -->
  <library path="../tools.jar" />

  <!-- Compiler, activate this to specify an alternative compiler such
  as jikes for EJB/JSP compiling. -->
  <!-- <compiler executable="jikes" classpath="/myjdkdir/jre/lib/rt.jar" /> -->
</application-server>

```

3. *data-sources.xml*

File ini diperlukan untuk mendaftarkan koneksi-koneksi basis data yang akan digunakan dalam aplikasi J2EE. Proses ini sekaligus mendaftarkan koneksi-koneksi tersebut ke JNDI.

```

<data-sources>
  <data-source
    class="com.evermind.sql.DriverManagerDataSource"
    name="production"
    location="jdbc/OracleCoreDS"
    xa-location="jdbc/xa/OracleXADS"
    ejb-location="jdbc/production"
    connection-driver="oracle.jdbc.driver.OracleDriver"
    username="production"
    password="prodx1"
    url="jdbc:oracle:thin:@10.126.11.10:1521:itarizki"
    inactivity-timeout="30"
  />
  <data-source
    class="com.evermind.sql.DriverManagerDataSource"
    name="WebmanDS"
    location="jdbc/OracleCoreDS"
    xa-location="jdbc/xa/OracleXADS"
    ejb-location="jdbc/WebmanDS"
    connection-driver="oracle.jdbc.driver.OracleDriver"
    username="webman"
    password="webman"
    url="jdbc:oracle:thin:@10.126.11.10:1521:itarizki"
    inactivity-timeout="30"
  />
</data-sources>

```

Terdapat beberapa hal yang harus diperhatikan, diantaranya:

- ❖ name menunjukkan nama koneksi
- ❖ ejb-location menunjukkan nama koneksi yang akan dipergunakan dalam pemrograman J2EE

- ❖ connection-driver menunjukkan driver JDBC yang dipergunakan untuk mengakses basis data
- ❖ username menunjukkan username dari skema
- ❖ password menunjukkan password dari skema
- ❖ url menunjukkan lokasi dari server basis data

e-MOC	Unocal 76
Version - 1	User Acceptance

e-MOC System

Version - 1

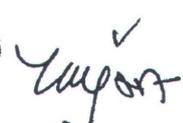
System Acceptance

Date	Reason for change(s)	Author(s)
January 03, 2004	First Draft	Rizki Agus Zandra

e-MOC	Unocal 76
Version - 1	User Acceptance

This document is purposed to obtain project acceptance. This document is related to acceptance test document that signed by project team member.

SYSTEM ACCEPTANCE

Position	Name	Date	Signature
System Owner	Stevanus Adrianto (MOC Coordinator)	S. ADRIANTO 20 JAN 2004	
System Owner	Tri Budi Lestyaningsih (IT Dept Manager)	TRI BUDI LESTYANINGSIH 21 JAN 2004	
Project Manager	for Agus M Manggala (Senior Supervisor, Application Support)	ADRIYAN PUTRA 20 JAN 2004	
Team Member	Dwi Nugroho (System Analyst)	DWI P. NUGROHO 19 JAN 2004	

e-MOC	Unocal 76
Version - 1	User Acceptance

List of Items to be accepted

1.	CREATE	Create MOC proposal
2.	CREATE	<p>MOC proposal requirements:</p> <ul style="list-style-type: none"> - Proposal date can be changed, with default value is today's date - Originator can be automatically determined from login - Can change default Supervisor of originator - Originator can attach files to the MOC proposal - Proposal submission works properly
3.	MOC Proposal APPROVAL	<p>Proposal approval requirements (for Supervisor, Superintendent, Manager and LPE):</p> <ul style="list-style-type: none"> - Nominate Project Leader works properly - Approver's remarks is stored properly - Approval (approve/reject) work properly
4.	MOCC FORWARD Proposal	<p>MOC Navigate Form requirements:</p> <ul style="list-style-type: none"> - Form has entry fields for assigning Superintendent, Manager, LPE - Manager is auto detected by default, but can be changed - When submitted, approval tasks will be distributed correctly to each assigned approver
5.	FIRM PL Assignment	<p>Firm Project Leader Assignment form requirements:</p> <ul style="list-style-type: none"> - PL nominees (by the approvers) are displayed on the list - "Other" option is provided to enable MOCC to select other person as PL - PL Assignment works properly
6.	PL Screening results	<p>Hazard Analysis screening (Form 2) requirements:</p> <ul style="list-style-type: none"> - Screening items are complete and correct. - Form can check if the user has selected "Yes" before choosing a HA technique - Form saves properly <p>MOC Plan screening (Form 3) requirements:</p> <ul style="list-style-type: none"> - Screening items are complete and correct. - Form saves properly
7.	Screening Approval	<p>Approval for Project Leader's screening results requirements (for Supv. of PL, LPE, Chief E&C engineer):</p> <ul style="list-style-type: none"> - Approval form can display MOC proposal and screening results properly for review

e-MOC	Unocal 76
Version - 1	User Acceptance

		<ul style="list-style-type: none"> - Approval functionality works properly
8.	MOC FORM 4	<p>Implementation Hazard Screening (Form 4) requirements:</p> <ul style="list-style-type: none"> - Screening items are complete and correct - Form saves properly when submitted and forwarded to the right person
9.	MOC FORM 4 APPROVAL	<p>Implementation Hazard Screening (Form 4) Approval requirements:</p> <ul style="list-style-type: none"> - Form can display MOC proposal, screening results, and Implementation Hazards screening results for review - Approval functionality works properly
10.	Pre STARTUP	<p>Pre Startup form requirements:</p> <ul style="list-style-type: none"> - Startup checklist items are complete and correct. - Pre startup check form works properly.
11.	Pre STARTUP APPROVAL	<p>Pre Startup form Approval requirements:</p> <ul style="list-style-type: none"> - Form can display Startup checklist results correctly - Approval functionality works properly
12.	PL Project Status	Project Leader is provided a way to update MOC project status during the change project
13.	Closure Status	<p>MOC Closure Status form requirements:</p> <ul style="list-style-type: none"> - Project Leader can attach documents required for change project closure - Initiate Closure works properly
14.	Closure APPROVAL	Change Project closure approval works properly
15.	OVERALL APPROVAL FLOW	Overall approval flow complies to the latest MOC Approval Flow specification (E-MOC Flow Attachment 5A.xls)
16.	MAIL REMINDER	<ul style="list-style-type: none"> - Mail notification and reminder are sent to the appropriate approvers - Reminders will be sent at 7 am every morning
17.	SECURITY	<p>Security requirements:</p> <ul style="list-style-type: none"> - Integration to Webman is implemented - e-MOC system uses the centralized LDAP service for user authentication