

**RANCANG BANGUN SISTEM *EXPERT LOCATOR* UNTUK MENDUKUNG
SHARING KNOWLEDGE PADA JEJARING PARA AHLI PADA
PT. PETROKIMIA GRESIK**

SKRIPSI

Laboratorium Sistem Informasi

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh:

DODY YUDHA PRAWIRA

125150209111022

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

LEMBAR PERSETUJUAN

**RANCANG BANGUN SISTEM *EXPERT LOCATOR* UNTUK
MENDUKUNG *SHARING KNOWLEDGE* PADA JEJARING PARA AHLI
PADA PT. PETROKIMIA GRESIK**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh:

DODY YUDHA PRAWIRA
NIM. 125150209111022

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Aditya Rachmadi, S.ST., MTL.
NIP. 860421 16 1 1 0426

Diah Priharsari, ST., MT.
NIP.



LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM *EXPERT LOCATOR* UNTUK
MENDUKUNG *SHARING KNOWLEDGE* PADA JEJARING PARA AHLI
PADA PT. PETROKIMIA GRESIK**

SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh:

DODY YUDHA PRAWIRA
NIM. 125150209111022

Skripsi ini telah diuji dan dinyatakan lulus pada
Tanggal 13 Maret 2015

Penguji I

Denny Sagita R., S.Kom, M.Kom
NIK. 85112406110250

Penguji II

Adharul Muttaqin, S.T., M.T.
NIK. 19760121 200501 1 001

Penguji III

Aryo Pinandito, S.T., M.MT.
NIK. 830519161110374

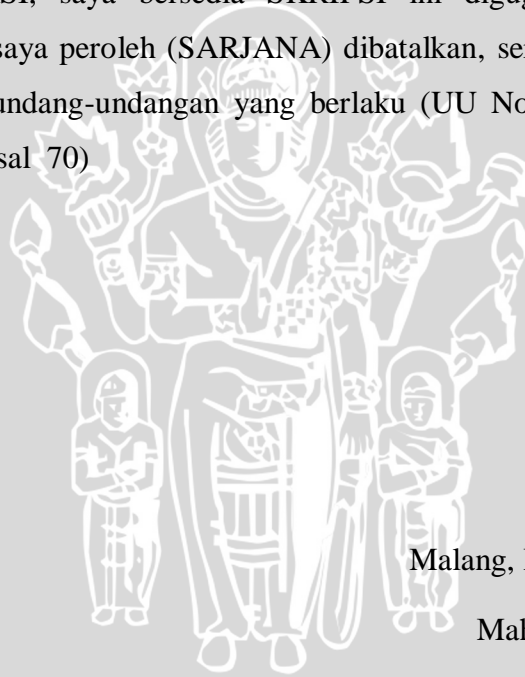
Mengetahui
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS SKRIPSI

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

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



Malang, Maret 2015

Mahasiswa,

Dody Yudha Prawira

NIM. 125150209111022

KATA PENGANTAR

Puji syukur kita panjatkan kepada Tuhan Yang Maha Esa karena atas segala rahmat, taufik serta hidayah-Nya, sehingga skripsi dengan judul:

Rancang Bangun Sistem Expert Locator Untuk Mendukung Sharing Knowledge Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik, dapat diselesaikan dengan baik.

Skripsi ini disusun berdasarkan kondisi dan situasi yang diteliti serta dari beberapa literatur, dimaksudkan untuk memenuhi syarat untuk memperoleh gelar Sarjana Komputer pada Program Teknologi Informasi & Ilmu Komputer.

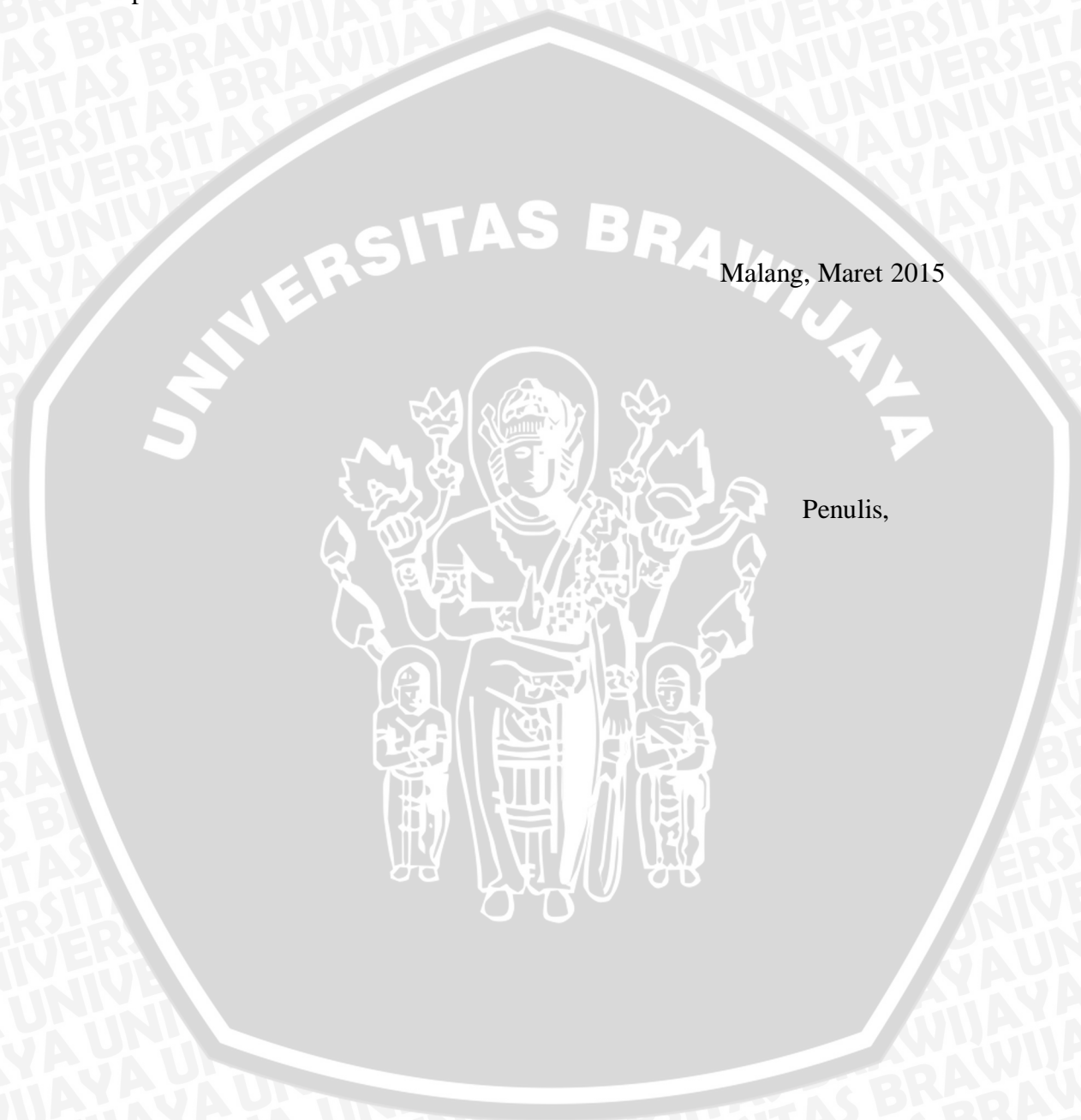
Penulis menyadari bahwa skripsi ini tidak akan mencapai titik akhir penyelesaian tanpa bantuan berbagai pihak, karenanya penulis mengucapkan terima kasih kepada:

1. Allah SWT yang telah membantu secara spiritual sehingga dapat terselesaikannya skripsi ini.
2. Kedua orang tua yang selalu mengalirkan do'a tiada hentinya dan memberikan dukungan moral maupun material.
3. Bapak Drs. Marji, MT. dan Bapak Issa Arwani, ST., MT., selaku Ketua dan Sekretaris Program Studi Informatika
4. Bapak Aditya Rachmadi, S.ST., MTI. dan Ibu Diah Priharsari, ST., MT. selaku dosen pembimbing I dan pembimbing II pada penyusunan Skripsi ini
5. Mas Aldy Sefan Rezanaldy, S.Kom. dan Mas Gary Eka Luviano, ST. selaku pembina di PT. Petrokimia Gresik yang memberikan bantuan dalam penyusunan Skripsi ini.
6. Ardyanto Hermawan dan juga Danu Hendarto yang telah turut membantu dalam penyelesaian Skripsi ini.
7. Sahabat-sahabat SAP 2012 yang banyak memberikan informasi dan bantuan dalam penyusunan Skripsi ini.
8. Serta semua pihak yang tidak dapat disebutkan satu persatu yang telah turut membantu baik secara langsung maupun tidak langsung dalam penyelesaian Skripsi ini.

Akhir kata semoga Skripsi ini dapat bermanfaat bagi kita semua dan menjadi bahan renungan dalam rangka mengembangkan ilmu pengetahuan sebagai makna regenerasi bangsa untuk mencapai suatu keberhasilan khususnya untuk penulis sendiri.

Malang, Maret 2015

Penulis,



ABSTRAK

Dody Yudha Prawira. 2015. Rancang Bangun Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya, Malang.

Dosen Pembimbing: Bapak Aditya Rachmadi, S.ST., MTI. dan Ibu Diah Priharsari, ST., MT.

Aplikasi *expert locator* merupakan aplikasi yang mempunyai fungsi sebagai wadah untuk membedakan *talent* setiap masing-masing karyawan. Aplikasi ini juga akan membantu setiap karyawan dalam hal berinteraksi antara karyawan satu dengan yang lainnya, serta memberikan peluang bagi karyawan untuk bisa dapat berkonsultasi dengan para ahli yang sudah dikategorikan sesuai dengan bidang pekerjaannya masing-masing dalam bentuk forum diskusi. Aplikasi ini dibuat menggunakan bahasa pemrograman PHP dengan CI (Codeigniter) sebagai *frameworknya*, MySQL sebagai database dan AJAX (*Asynchronous JavaScript and XML*) sebagai bahasa pemrograman *web* agar membuat aplikasi menjadi interaktif. Metode yang digunakan dalam perancangan aplikasi ini adalah metode prototyping untuk memudahkan pengguna mengetahui kesesuaian antara sistem yang dihasilkan dengan kebutuhan tanpa harus menunggu sampai sistem diimplementasikan. Aplikasi ini dapat diakses oleh semua karyawan yang telah terdaftar pada sistem. Dari hasil uji coba dengan menggunakan metode *functional testing* dihasilkan bahwa fungsi aplikasi ini secara keseluruhan berfungsi 100% valid dan berjalan dengan baik dan dari hasil pengujian *usability testing* menunjukkan bahwa 53% pengguna setuju terhadap kemudahan penggunaan aplikasi ini.

Kata kunci: *Talent, Expert, Knowledge Management System, Expert Locator*

ABSTRACT

Dody Yudha Prawira. 2015., Information Technology and Computer Science Program, Brawijaya University, Malang.

Supervisor: Mr. Aditya Rachmadi, S.ST., MTI. and Mrs. Diah Priharsari, ST., MT.

Expert locator application is an application that has function as a container for distinguishing talent of each individual employee. This application will also help every employee in terms of interaction among employees, as well as providing opportunities for employees to be able to consult with experts that have been categorized according to their respective areas of work in the form of a discussion forum. This application is created using PHP programming language with a CI (Codeigniter) as a framework, MySQL as the database and AJAX (Asynchronous JavaScript and XML) as programming language to make this application interactive. The method used in the design of this application is a prototyping method to allow users to determine the suitability of the resulting system needs without having to wait until the system is implemented. This application can be accessed by all employees who have registered on the system. From the test results by using the method of functional testing is generated that the overall function of this application works 100% valid and running properly and usability testing of test results showed that 53% of users agree to the ease of use of this application.

Keyword: *Talent, Expert, Knowledge Management System, Expert Locator*

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xx
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat	4
A. Bagi Perusahaan	4
B. Bagi Penulis	4
1.6 Sistematika Penulisan	5
BAB II DASAR TEORI	6
2.1 Kajian Pustaka	6
2.2 Kamus Kompetensi	8
2.3 <i>Knowledge Management System</i> Saat Ini	10
2.4 Profil Perusahaan	13
2.5 Visi dan Misi	13
2.5.1 Visi	13
2.5.2 Misi	13



2.6	Knowledge	14
2.6.1	Tipe Knowledge.....	14
2.7	Knowledge Management.....	15
2.8	Knowledge Management Life Cycle	15
2.9	Knowledge Management System	18
2.10	Rekayasa Perangkat Lunak	18
2.11	Unified Modeling Language (UML)	19
2.11.1	Konsep Dasar UML (Unified Modelling Language)	19
2.11.2	Use Case Diagram.....	19
2.11.3	Activity Diagram	19
2.11.4	Sequence Diagram	20
2.11.5	Class Diagram	20
2.12	Software Development Life Cycle (SDLC).....	20
2.12.1	Model SDLC.....	21
2.12.1.1	Model Prototipe.....	21
2.13	PHP.....	22
2.14	Framework CodeIgniter.....	22
2.15	MYSQL.....	23
2.16	Pengujian Perangkat Lunak	23
2.16.1	Teknik Pengujian Perangkat Lunak	24
BAB III METODE PENELITIAN DAN PERANCANGAN		25
3.1	Metodologi Penelitian.....	25
3.1.1	Penentuan Masalah.....	26
3.1.2	Studi Literatur dan Penyusunan Dasar Teori.....	26
3.1.3	Analisa Kondisi Saat Ini	27
3.1.4	Analisa dan Perancangan Perangkat Lunak.....	27
3.1.5	Membuat <i>Prototype</i> Sistem.....	28
3.1.6	Pengujian Sistem.....	28
3.1.7	Pengambilan Kesimpulan dan Saran	28
3.2	Analisa Kondisi Saat Ini	29

3.2.1	Analisis Untuk Pengembangan Fitur <i>Expert Locator</i> Pada KMS PT. Petrokimia Gresik	29
3.2.2	Bisnis Proses <i>Expert Locator</i>	31
3.3	Analisa dan Perancangan Perangkat Lunak	32
3.3.1	Analisis Sistem	33
3.3.2	Analisis Kebutuhan Fungsional	33
3.3.3	Analisis Kebutuhan Non Fungsional	36
3.3.4	Hasil Dari Proses Penerapan Metode Prototyping	37
3.3.5	Diagram <i>Use Case</i>	47
3.4	Desain Sistem	84
3.4.1	Rancangan Database	84
3.4.2	<i>Mind Mapping</i>	84
3.4.3	<i>Activity Diagram</i>	87
3.4.4	<i>Class Diagram</i>	98
3.4.5	<i>Sequence Diagram</i>	99
3.5	Desain Interface	104
BAB IV IMPLEMENTASI		106
4.1	Spesifikasi Sistem	106
4.1.1	Spesifikasi Kebutuhan Perangkat Lunak	106
4.1.2	Spesifikasi Kebutuhan Perangkat Keras	106
4.2	Batasan – Batasan Implementasi	107
4.3	Implementasi Basis Data	107
4.3.1	DDL untuk Membuat Basis Data kms	107
4.3.2	DDL untuk Membuat Tabel karyawan	107
4.3.3	DDL untuk Membuat Tabel pendidikan	108
4.3.4	DDL untuk Membuat Tabel pengalaman_kerja	108
4.3.5	DDL untuk Membuat Tabel training	109
4.3.6	DDL untuk Membuat Tabel sertifikasi	109
4.3.7	DDL untuk Membuat Tabel m_user	110
4.3.8	DDL untuk Membuat Tabel m_akses	110
4.3.9	DDL untuk Membuat Tabel m_unit_kerja	111

4.3.10 DDL untuk Membuat Tabel m_group	111
4.3.11 DDL untuk Membuat Tabel m_group_status	111
4.3.12 DDL untuk Membuat Tabel m_group_komentar.....	111
4.3.13 DDL untuk Membuat Tabel m_status	112
4.3.14 DDL untuk Membuat Tabel m_komentar	112
4.3.15 DDL untuk Membuat Tabel notifikasi	112
4.3.16 DDL untuk Membuat Tabel permintaan_group.....	112
4.3.17 DDL untuk Membuat Tabel permintaan_teman	113
4.3.18 DDL untuk Membuat Tabel pesan	113
4.3.19 DDL untuk Membuat Tabel user_album.....	113
4.4 Implementasi Antarmuka.....	114
4.4.1 Implementasi Antarmuka Halaman Utama.....	114
4.4.2 Implementasi Antarmuka Halaman Administrator	114
4.4.3 Implementasi Antarmuka Halaman User.....	121
BAB V PENGUJIAN	126
5.1 Pengujian	126
5.1.1 <i>Functional Testing</i>	126
5.1.2 <i>Usability Testing</i>	150
5.2 Analisa Hasil.....	153
5.2.1 Analisa Hasil <i>Functional Testing</i>	153
5.2.2 Analisa Hasil <i>Usability Testing</i>	153
BAB VI PENUTUP	155
6.1 Kesimpulan.....	155
6.2 Saran.....	155
DAFTAR PUSTAKA	157
LAMPIRAN	159



DAFTAR GAMBAR

Gambar 2.1 Usulan Penerapan Kerangka <i>Knowledge Management</i> untuk Facebook	6
Gambar 2.2 Jejaring Facebook digunakan untuk <i>Knowledge Management</i>	7
Gambar 2.3 Kamus Kompetensi	9
Gambar 2.4 Knowledge Flow	11
Gambar 2.5 <i>Knowledge Mapping</i>	12
Gambar 2.6 Pengkonversian <i>Tacit dan Explicit Knowledge</i>	15
Gambar 2.7 <i>Model Prototipe</i>	21
Gambar 3.1 Alur Pengerjaan Secara Umum	25
Gambar 3.2 Gambaran Umum Aplikasi <i>Expert Locator</i>	31
Gambar 3.3 Bisnis Proses dari <i>Expert Locator</i>	32
Gambar 3.4. <i>Use Case Diagram</i> Pengolahan Data Pribadi Karyawan	49
Gambar 3.5. <i>Use Case Diagram</i> Pengolahan Data Jejaring Sosial	61
Gambar 3.6. <i>Use Case Diagram</i> Pengolahan Data Akun	67
Gambar 3.7. <i>Use Case Diagram</i> Modul Karyawan	75
Gambar 3.9 <i>Mind Mapping</i>	84
Gambar 3.8 Rancangan Database	86
Gambar 3.10 <i>Activity</i> karyawan melakukan edit <i>profil</i>	87
Gambar 3.11 <i>Activity</i> karyawan menulis status	88
Gambar 3.12 <i>Activity</i> karyawan melakukan <i>update</i> status	88
Gambar 3.13 <i>Activity</i> karyawan melakukan hapus status	89
Gambar 3.14 <i>Activity</i> karyawan menulis komentar	89
Gambar 3.15 <i>Activity</i> karyawan melakukan <i>update</i> komentar	90
Gambar 3.16 <i>Activity</i> karyawan melakukan hapus komentar	90
Gambar 3.17 <i>Activity</i> karyawan melakukan <i>private message</i>	91
Gambar 3.18 <i>Activity</i> karyawan melakukan <i>join group</i>	91
Gambar 3.19 <i>Activity</i> karyawan melakukan add friend	92
Gambar 3.20 <i>Activity</i> admin kelola data karyawan	92
Gambar 3.21 <i>Activity</i> admin kelola data pendidikan	93
Gambar 3.22 <i>Activity</i> admin kelola data training	93

Gambar 3.23 <i>Activity</i> admin kelola data pengalaman kerja.....	94
Gambar 3.24 <i>Activity</i> admin kelola data unit kerja	94
Gambar 3.25 <i>Activity</i> admin kelola data sertifikasi.....	95
Gambar 3.26 <i>Activity</i> admin kelola data user.....	95
Gambar 3.27 <i>Activity</i> admin kelola data akses.....	96
Gambar 3.28 <i>Activity</i> admin kelola data group	96
Gambar 3.29 <i>Activity</i> admin kelola data status karyawan	97
Gambar 3.30 <i>Activity</i> admin kelola data pesan	97
Gambar 3.31 <i>Activity</i> admin kelola data album	98
Gambar 3.32 <i>Class Diagram</i>	99
Gambar 3.33 <i>Sequence</i> diagram login	100
Gambar 3.34 <i>Sequence</i> diagram edit profile	100
Gambar 3.35 <i>Sequence</i> diagram menulis status	101
Gambar 3.36 <i>Sequence</i> diagram menulis komentar	101
Gambar 3.37 <i>Sequence</i> diagram mengirim pesan	102
Gambar 3.38 <i>Sequence</i> diagram <i>join group</i>	102
Gambar 3.39 <i>Sequence</i> diagram permintaan teman	103
Gambar 3.40 <i>Sequence</i> diagram kelola data <i>user</i>	103
Gambar 3.41 Desain interface tampilan utama user	104
Gambar 3.42 Desain interface tampilan home user.....	105
Gambar 3.43 Desain interface tampilan utama admin.....	105
Gambar 4.1 Halaman Utama.....	114
Gambar 4.2 Halaman Administrator.....	115
Gambar 4.3 Halaman Modul Karyawan	115
Gambar 4.4 Halaman Modul Pendidikan.....	116
Gambar 4.5 Halaman Modul Pengalaman	116
Gambar 4.6 Halaman Modul Training	117
Gambar 4.7 Halaman Modul Sertifikasi	117
Gambar 4.8 Halaman Modul Unit Kerja.....	118
Gambar 4.9 Halaman Modul User.....	118
Gambar 4.10 Halaman Modul Akses.....	119
Gambar 4.11 Halaman Modul Group	119

Gambar 4.12 Halaman Modul Status.....	120
Gambar 4.13 Halaman Modul Pesan.....	120
Gambar 4.14 Halaman Modul Album.....	121
Gambar 4.15 Halaman User.....	121
Gambar 4.16 Halaman Timeline.....	122
Gambar 4.17 Halaman Profile Saya.....	122
Gambar 4.18 Halaman Teman.....	123
Gambar 4.19 Halaman Cari Teman.....	123
Gambar 4.20 Halaman Group.....	124
Gambar 4.21 Halaman Pesan.....	124
Gambar 4.22 Halaman Akun Saya.....	125
Gambar 5.1. Rekapitulasi Hasil dari Pengujian <i>Usability Testing</i> dalam bentuk persentase (%).....	154



DAFTAR TABEL

Tabel 1.1 Tabel <i>Knowledge Inventory</i>	10
Tabel 3.1 Definisi <i>Actor</i>	47
Tabel 3.2 Definisi <i>Use Case</i>	48
Tabel 3.3. Skenario <i>Use Case Login</i>	49
Tabel 3.4. Skenario <i>Use Case</i> Kelola Data Karyawan	50
Tabel 3.5. Skenario <i>Use Case</i> Kelola Data Pendidikan.....	52
Tabel 3.6. Skenario <i>Use Case</i> Kelola Data Training.....	54
Tabel 3.7. Skenario <i>Use Case</i> Kelola Data Pengalaman Kerja.....	56
Tabel 3.8. Skenario <i>Use Case</i> Kelola Data Sertifikasi	59
Tabel 3.9. Skenario <i>Use Case</i> Kelola Data Status Karyawan.....	61
Tabel 3.10. Skenario <i>Use Case</i> Kelola Data Pesan	63
Tabel 3.11. Skenario <i>Use Case</i> Kelola Data Album	65
Tabel 3.12. Skenario <i>Use Case</i> Kelola Data User.....	67
Tabel 3.13. Skenario <i>Use Case</i> Kelola Data Akses.....	69
Tabel 3.14. Skenario <i>Use Case</i> Kelola Data Unit Kerja.....	71
Tabel 3.15. Skenario <i>Use Case</i> Kelola Data Group	73
Tabel 3.16. Skenario <i>Use Case</i> Edit Profil	75
Tabel 3.17. Skenario <i>Use Case</i> Kelola Status.....	76
Tabel 3.18. Skenario <i>Use Case</i> Kelola Komentar	78
Tabel 3.19. Skenario <i>Use Case</i> Private Message	80
Tabel 3.20. Skenario <i>Use Case</i> Join Group.....	81
Tabel 3.21. Skenario <i>Use Case</i> Add Friend	83
Tabel 4.1 Spesifikasi Kebutuhan Perangkat Lunak	106
Tabel 4.2 Spesifikasi Kebutuhan Perangkat Keras.....	106
Tabel 5.1. Hasil <i>Functional Testing Login Admin Expert Locator</i>	126
Tabel 5.2. Hasil <i>Functional Testing Login Admin Expert Locator</i> dengan inputan yang salah.....	127
Tabel 5.3. Hasil <i>Functional Testing</i> Admin Tambah Data Karyawan	127
Tabel 5.4. Hasil <i>Functional Testing</i> Admin Tambah Data Karyawan dengan NIK yang sudah digunakan.....	127

Tabel 5.5. Hasil <i>Functional Testing</i> Admin Ubah Data Karyawan	128
Tabel 5.6. Hasil <i>Functional Testing</i> Admin Ubah Data Karyawan dengan NIK yang sudah digunakan.....	128
Tabel 5.7. Hasil <i>Functional Testing</i> Admin Hapus Data Karyawan	129
Tabel 5.8. Hasil <i>Functional Testing</i> Admin Tambah Data Pendidikan.....	129
Tabel 5.9. Hasil <i>Functional Testing</i> Admin Tambah Data Pendidikan dengan NIK yang sudah digunakan.....	129
Tabel 5.10. Hasil <i>Functional Testing</i> Admin Ubah Data Pendidikan	130
Tabel 5.11. Hasil <i>Functional Testing</i> Admin Ubah Data Pendidikan dengan NIK yang sudah digunakan.....	130
Tabel 5.12. Hasil <i>Functional Testing</i> Admin Hapus Data Pendidikan.....	131
Tabel 5.13. Hasil <i>Functional Testing</i> Admin Tambah Data Pengalaman Kerja.	131
Tabel 5.14. Hasil <i>Functional Testing</i> Admin Tambah Data Pengalaman Kerja dengan NIK yang sudah digunakan.....	132
Tabel 5.15. Hasil <i>Functional Testing</i> Admin Ubah Data Pengalaman Kerja	132
Tabel 5.16. Hasil <i>Functional Testing</i> Admin Ubah Data Pengalaman Kerja dengan NIK yang sudah digunakan.....	133
Tabel 5.17. Hasil <i>Functional Testing</i> Admin Hapus Data Pengalaman Kerja....	133
Tabel 5.18. Hasil <i>Functional Testing</i> Admin Tambah Data Training	133
Tabel 5.19. Hasil <i>Functional Testing</i> Admin Tambah Data Training dengan NIK yang sudah digunakan.....	134
Tabel 5.20. Hasil <i>Functional Testing</i> Admin Ubah Data Training	134
Tabel 5.21. Hasil <i>Functional Testing</i> Admin Ubah Data Training dengan NIK yang sudah digunakan.....	135
Tabel 5.22. Hasil <i>Functional Testing</i> Admin Hapus Data Training.....	135
Tabel 5.23. Hasil <i>Functional Testing</i> Admin Tambah Data Sertifikasi	135
Tabel 5.24. Hasil <i>Functional Testing</i> Admin Tambah Data Sertifikasi dengan NIK yang sudah digunakan.....	136
Tabel 5.25. Hasil <i>Functional Testing</i> Admin Ubah Data Sertifikasi.....	136
Tabel 5.26. Hasil <i>Functional Testing</i> Admin Ubah Data Sertifikasi dengan NIK yang sudah digunakan.....	137
Tabel 5.27. Hasil <i>Functional Testing</i> Admin Hapus Data Sertifikasi	137



Tabel 5.28. Hasil <i>Functional Testing</i> Admin Tambah Data Unit Kerja.....	137
Tabel 5.29. Hasil <i>Functional Testing</i> Admin Tambah Data Unit Kerja dengan Kode Unit Kerja yang sudah digunakan	138
Tabel 5.30. Hasil <i>Functional Testing</i> Admin Ubah Data Unit Kerja.....	138
Tabel 5.31. Hasil <i>Functional Testing</i> Admin Ubah Data Unit Kerja dengan Kode Unit Kerja yang sudah digunakan	139
Tabel 5.32. Hasil <i>Functional Testing</i> Admin Hapus Data Unit Kerja.....	139
Tabel 5.33. Hasil <i>Functional Testing</i> Admin Tambah Data <i>User</i>	139
Tabel 5.34. Hasil <i>Functional Testing</i> Admin Tambah Data <i>User</i> dengan username yang sudah digunakan.....	140
Tabel 5.35. Hasil <i>Functional Testing</i> Admin Ubah Data <i>User</i>	140
Tabel 5.36. Hasil <i>Functional Testing</i> Admin Ubah Data <i>User</i> dengan username yang sudah digunakan.....	140
Tabel 5.37. Hasil <i>Functional Testing</i> Admin Hapus Data <i>User</i>	141
Tabel 5.38. Hasil <i>Functional Testing</i> Admin Tambah Data Akses	141
Tabel 5.39. Hasil <i>Functional Testing</i> Admin Ubah Data Akses	141
Tabel 5.40. Hasil <i>Functional Testing</i> Admin Hapus Data Akses	142
Tabel 5.41. Hasil <i>Functional Testing</i> Admin Tambah Data Group	142
Tabel 5.42. Hasil <i>Functional Testing</i> Admin Ubah Data Group	142
Tabel 5.43. Hasil <i>Functional Testing</i> Admin Hapus Data Group	143
Tabel 5.44. Hasil <i>Functional Testing</i> Admin Tambah Data Status.....	143
Tabel 5.45. Hasil <i>Functional Testing</i> Admin Ubah Data Status.....	143
Tabel 5.46. Hasil <i>Functional Testing</i> Admin Hapus Data Status	144
Tabel 5.47. Hasil <i>Functional Testing</i> Admin Tambah Data Pesan	144
Tabel 5.48. Hasil <i>Functional Testing</i> Admin Ubah Data Pesan	144
Tabel 5.49. Hasil <i>Functional Testing</i> Admin Hapus Data Pesan.....	144
Tabel 5.50. Hasil <i>Functional Testing</i> Admin Tambah Data Album	145
Tabel 5.51. Hasil <i>Functional Testing</i> Admin Ubah Data Album.....	145
Tabel 5.52. Hasil <i>Functional Testing</i> Admin Hapus Data Album	145
Tabel 5.53. Hasil <i>Functional Testing Logout</i> Admin.....	146
Tabel 5.54. Hasil <i>Functional Testing Login User</i> (Karyawan).....	146
Tabel 5.55. Hasil <i>Functional Testing User</i> (Karyawan) Menulis Status.....	146

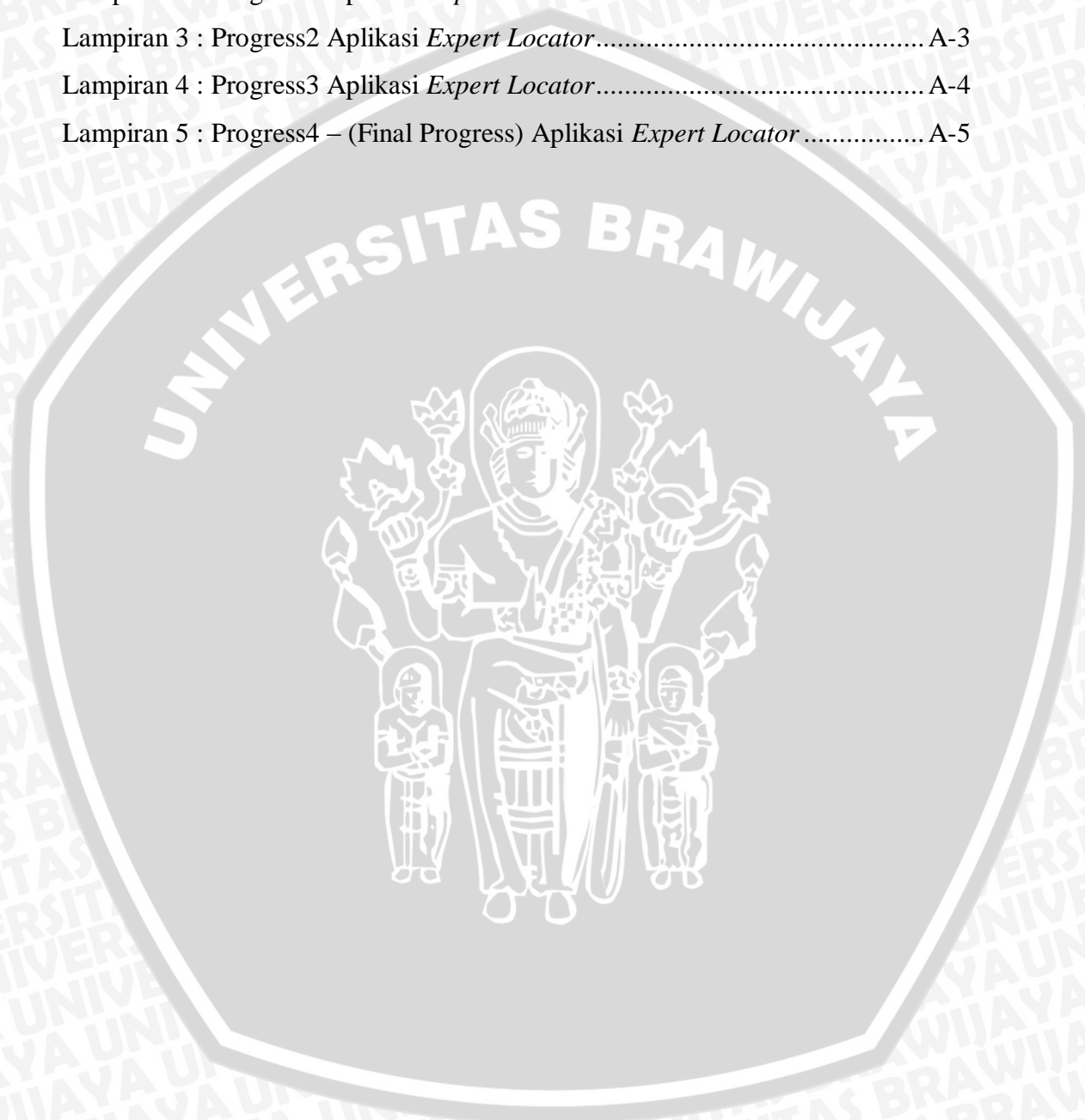


Tabel 5.56. Hasil <i>Functional Testing User</i> (Karyawan) Ubah Status	147
Tabel 5.57. Hasil <i>Functional Testing User</i> (Karyawan) Hapus Status.....	147
Tabel 5.58. Hasil <i>Functional Testing User</i> (Karyawan) Menulis Komentar.....	147
Tabel 5.59. Hasil <i>Functional Testing User</i> (Karyawan) Ubah Komentar	148
Tabel 5.60. Hasil <i>Functional Testing User</i> (Karyawan) Hapus Komentar.....	148
Tabel 5.61. Hasil <i>Functional Testing User</i> (Karyawan) Tambah Teman	148
Tabel 5.62. Hasil <i>Functional Testing User</i> (Karyawan) Hapus Teman	149
Tabel 5.63. Hasil <i>Functional Testing User</i> (Karyawan) Join Group.....	149
Tabel 5.64. Hasil <i>Functional Testing User</i> (Karyawan) Menulis Pesan Pribadi	149
Tabel 5.65. Hasil <i>Functional Testing User</i> (Karyawan) Hapus Pesan Pribadi...	150
Tabel 5.66. Hasil <i>Functional Testing User</i> (Karyawan) Edit Profil.....	150
Tabel 5.67. Hasil <i>Functional Testing Logout User</i> (Karyawan).....	150
Tabel 5.68. Hasil <i>Usability Testing</i>	151



DAFTAR LAMPIRAN

Lampiran 1 : Mark-Up Aplikasi <i>Expert Locator</i>	A-1
Lampiran 2 : Progress1 Aplikasi <i>Expert Locator</i>	A-2
Lampiran 3 : Progress2 Aplikasi <i>Expert Locator</i>	A-3
Lampiran 4 : Progress3 Aplikasi <i>Expert Locator</i>	A-4
Lampiran 5 : Progress4 – (Final Progress) Aplikasi <i>Expert Locator</i>	A-5



BAB I PENDAHULUAN

1.1 Latar Belakang

PT PETROKIMIA GRESIK merupakan perusahaan milik negara dan produsen pupuk terlengkap di Indonesia yang memproduksi berbagai macam pupuk, seperti: Urea, ZA, SP-36, NPK Phonska, DAP, NPK Kebomas, ZK dan pupuk organik yaitu Petroganik. PT Petrokimia Gresik juga telah memproduksi produk non pupuk seperti Asam Sulfat, Asam fosfat, Amoniak, Dry Ice, Aluminum Fluoride, Cement Retarder, dll. Keberadaan PT Petrokimia Gresik adalah untuk mendukung program Pemerintah meningkatkan produksi pertanian nasional.[PGS-12]. Bagi perusahaan sebesar itu komunikasi antar bagian sangatlah penting dan dibutuhkan untuk menunjang kemajuan sebuah perusahaan. Di dalam setiap departemen pastinya mempunyai peran tersendiri di dalamnya, begitu juga dengan semua karyawannya. Setiap karyawan pastinya mempunyai keahlian masing – masing yang sesuai dengan bidang pekerjaannya. Akan tetapi tidak menutup kemungkinan bagi setiap karyawan yang ada di perusahaan tersebut untuk bisa dapat belajar sesuatu hal yang baru yang berkaitan dengan departemen lain. Dalam hal ini mereka dapat saling berbagi ilmu dan juga bertukar informasi mengenai pengetahuan antara karyawan satu dengan yang lainnya dengan mudah dan cepat meskipun mereka berbeda departemen, dengan menggunakan sebuah sistem yang secara internal di dalam perusahaan tersebut.

Knowledge Management merupakan strategi untuk mendapatkan pengetahuan yang benar kepada orang yang tepat pada waktu yang tepat dan membantu orang berbagi dan menempatkan informasi ke dalam tindakan dengan cara yang berusaha untuk meningkatkan kinerja organisasi. *Knowledge Management* memberikan kemampuan untuk mencipta, mengkomunikasikan dan menerapkan pengetahuan yang diperlukan dan berguna bagi pencapaian semua jenis tujuan bisnis. Selain itu *knowledge management* juga berfungsi meningkatkan kemampuan organisasi untuk belajar dari lingkungannya dan menggabungkan pengetahuan ke dalam proses bisnis. *Knowledge Management* menjadi bidang yang penting dalam proses pembelajaran sebuah organisasi. Pengetahuan yang dimiliki oleh organisasi harus mampu memberikan kemajuan bagi organisasi itu sendiri. Agar organisasi dapat bertahan hidup, maka diwajibkan agar setiap orang yang ada di dalam organisasi sharing pengetahuan.[PUR-08].

Pada PT. Petrokimia Gresik sebenarnya sudah ada sistem yang digunakan dalam melakukan sharing pengetahuan. Sistem tersebut berada pada *knowledge management system* mereka yaitu berupa website KM. Dalam melakukan *sharing knowledge*, setiap karyawan terlebih dahulu harus melakukan proses *upload* mengenai informasi *knowledge* yang dimilikinya ataupun yang mereka dapatkan dari orang lain ke dalam sebuah website KM tersebut. Setiap informasi yang dibagikan dapat berupa artikel, dokumen maupun video. Keseluruhan informasi yang sudah di-*sharing* tersebut nantinya bisa dapat di lihat, dibaca maupun diunduh oleh keseluruhan karyawan lainnya. Hal ini bertujuan untuk membantu semua karyawan dalam mendapatkan suatu informasi mengenai *knowledge* yang dibutuhkan dengan cepat dan juga sesuai dengan bidang pekerjaannya masing – masing ataupun yang dapat menunjang dalam menyelesaikan pekerjaannya. Akan tetapi didalam web KM tersebut tidak mencantumkan informasi secara menyeluruh mengenai data diri dari setiap karyawan yang men-*share* informasi *knowledge* tersebut. Mereka juga tidak mengetahui apakah yang men-*share* informasi tersebut merupakan karyawan yang memang benar-benar *expert* dibidangnya atau hanya merupakan karyawan biasa pada umumnya.

Selama ini yang menjadi permasalahan di dalam perusahaan tersebut adalah tidak adanya suatu aplikasi pada *knowledge management system* mereka yang mampu untuk membedakan *talent* (kemampuan) antara karyawan satu dengan yang lainnya. Kemudian tidak adanya wadah bagi setiap karyawan untuk bisa dapat berinteraksi dengan seseorang yang dikategorikan sebagai seorang *expert* didalam perusahaan tersebut, sesuai dengan bidang pekerjaannya masing-masing. Hal ini yang menjadikan setiap karyawan tidak mampu untuk bisa berkembang dan juga meng-*explore* lebih jauh lagi kemampuan yang dimilikinya, itu disebabkan karena mereka tidak mengetahui kepada siapa mereka harus belajar, berbagi ilmu maupun bertukar informasi. Menanggapi hal tersebut maka dibutuhkan suatu pengembangan pada *knowledge management system* mereka. Pengembangan dilakukan dengan cara membuat suatu aplikasi yaitu *expert locator* pada *knowledge management system* mereka. Aplikasi *expert locator* ini nantinya akan mempunyai fungsi sebagai wadah untuk membedakan *talent* setiap masing-masing karyawan. Tidak hanya itu saja, aplikasi ini juga akan membantu mereka dalam hal berinteraksi antara karyawan satu dengan yang lainnya, serta memberikan peluang bagi karyawan untuk bisa dapat berkonsultasi dengan para ahli (*expert*) yang sudah dikategorikan sesuai dengan bidang pekerjaannya masing-masing dalam bentuk forum diskusi. Dengan adanya

aplikasi *expert locator* ini diharapkan setiap karyawan mengetahui akan kemampuan yang dimilikinya, kemudian mengetahui kemampuan yang ingin dikembangkan, lalu mengetahui siapa ahlinya, dan juga mengetahui bagaimana cara mengakses ahlinya. Selain itu diharapkan juga dari penerapan aplikasi ini nantinya karyawan akan mampu berkembang dengan baik tidak hanya pada kemampuan yang dimilikinya saja, namun dapat berkembang lebih jauh lagi.

1.2 Rumusan Masalah

Dengan memperhatikan uraian latar belakang di atas, maka penulis merumuskan masalah sebagai berikut:

- a. Apa saja *prasyarat* sistem *expert locator* yang sesuai dengan kebutuhan PT. Petrokimia Gresik ?
- b. Merancang dan membuat *prototype* Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik.
- c. Bagaimana pengujian pada Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik ?

1.3 Batasan Masalah

Untuk menghindari terlalu luasnya permasalahan yang ada dan untuk memperjelas pemahaman tentang pengaplikasian sistem ini, maka diberikan batasan-batasan sebagai berikut:

- a. Aplikasi ini digunakan di dalam ruang lingkup perusahaan atau bersifat internal.
- b. Keseluruhan data pada aplikasi ini merupakan data asli yang diperoleh dari perusahaan.
- c. Aplikasi ini berupa web dan dibuat dengan menggunakan framework *Codeigniter* dan database *MYSQL*.
- d. Aplikasi ini berupa *prototyping* yang secara keseluruhan konsep pengerjaannya sesuai dengan yang diinginkan dan juga yang dibutuhkan oleh perusahaan.
- e. Aplikasi ini difokuskan kepada *expert locatornya* yang berupa forum diskusi.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka dapat ditentukan tujuan sebagai berikut:

- a. Untuk membuat *prasyarat* sistem *expert locator* yang sesuai dengan kebutuhan PT. Petrokimia Gresik.
- b. Merancang dan membuat *prototype* Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik.
- c. Untuk melakukan pengujian terhadap Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik.

1.5 Manfaat

A. Bagi Perusahaan

- a) Memberikan kemudahan bagi semua karyawan yang ada di dalam perusahaan untuk bisa mengembangkan kemampuan yang dimilikinya.
- b) Memberikan suatu wadah bagi perusahaan untuk bisa menghubungkan antara karyawan satu dengan yang lainnya untuk bisa saling berinteraksi, berbagi ilmu serta bertukar informasi.
- c) Memberikan kemudahan pada setiap karyawan dalam hal menyelesaikan pekerjaannya di dalam perusahaan.

B. Bagi Penulis

- a) Menumbuhkan kepekaan dalam menangkap permasalahan kerja, menganalisis, dan menyelesaikannya sesuai dengan ilmu yang dimiliki dan dipelajari sewaktu kuliah.
- b) Memperdalam dan meningkatkan kualitas, keterampilan, dan kreatifitas pribadi yang sesuai dengan ilmu yang dimiliki.

1.6 Sistematika Penulisan

Sistematika penulisan ditujukan untuk memberikan gambaran dan uraian dari penulisan skripsi ini secara garis besar yang meliputi beberapa bab, sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat, metode penelitian, dan sistematika penulisan.

BAB II KAJIAN PUSTAKA DAN DASAR TEORI

Menguraikan materi-materi dan teori-teori yang mendukung pembuatan aplikasi yang berhubungan dengan *Knowledge, Knowledge Management, Knowledge Management System, Rekayasa Perangkat Lunak, Unified Modeling Language, Code Igniter, MySQL* yang nantinya akan diimplementasikan di perusahaan.

BAB III METODE PENELITIAN DAN PERANCANGAN

Membahas metode yang digunakan dalam penelitian yang terdiri dari studi literatur dan penyusunan dasar teori, analisis dan perancangan perangkat lunak, implementasi prototyping, pengujian sistem, serta pengambilan kesimpulan dan saran.

BAB IV IMPLEMENTASI

Membahas implementasi dari hasil perancangan pada bab sebelumnya, yang dilakukan selama membangun sistem dan kebutuhan-kebutuhan yang diterapkan sebagai pendukung sistem.

BAB V PENGUJIAN

Menguraikan tentang jalannya uji coba pada keseluruhan aplikasi yang dilakukan secara bertahap serta membahas tentang teknik pengujian yang digunakan terhadap sistem dan penjelasan hasil pengujian sistem yang dilakukan berdasarkan kasus uji.

BAB VI PENUTUP

Menjelaskan tentang kesimpulan yang merupakan jawaban dari rumusan masalah yang diambil dari Bab I dan pembahasan pada Bab III, dan Bab IV, serta berisi saran terhadap cara pembuatan aplikasi ini agar mendapatkan masukan yang lebih baik untuk tahap pengembangan selanjutnya.

BAB II DASAR TEORI

2.1 Kajian Pustaka

Dalam proses penyusunan skripsi ini, penulis mengacu pada penelitian yang sudah ada sebelumnya yaitu : Phosaard, Satidchoke. & Wiriypinit, Mongkolchai. 2011. “*Knowledge Management via Facebook : Building a Framework for Knowledge Management on a Social Network by Aligning Business, IT and Knowledge Management*”. United Kingdom : WCE.[PHO-11]. Pada penelitian ini dibahas mengenai membangun kerangka *knowledge management* pada jejaring sosial facebook dengan menyelaraskan bisnis, IT dan juga *knowledge management* itu sendiri. Dalam hal ini organisasi memanfaatkan media jejaring sosial facebook, seperti yang telah diusulkan pada tabel II pada gambar 2.1.

TABLE II
BUSINESS-IT-KM PROCESS AND INFRASTRUCTURE ALIGNMENT

Business Strategy	IT Strategy	KM Strategy
Business Administrative Structure and Recourses <ul style="list-style-type: none"> • People: Project Manager, Staffs, Exhibitors, Participants, etc. 	IT Architecture and Resources <ul style="list-style-type: none"> • Event Website • Event Facebook/Project Facebook • Google Docs Documents: Spreadsheets • IT Staffs 	Administrative Structure and Resources for KM <ul style="list-style-type: none"> • Project Manager • Staffs • Knowledge Gathering Staffs
Business Processes <ul style="list-style-type: none"> • Project Planning • Promotion • Booth Sales • Venue Preparation • Event Activities • Evaluations 	IT Processes <ul style="list-style-type: none"> • Using Facebook Group for Privacy and Immediate Notifications • Sending Facebook Message for Personal Questions, Posting on Facebook Wall Post for Group Questions • Project Tracking via Google Docs Spreadsheets • Evaluations 	KM Processes <ul style="list-style-type: none"> • Information and Issues Related to Each Project Tasks – Explicit Knowledge Seamlessly Captured • Issues Compilation on Walls, Message, and Boxes – Implicit & Explicit Knowledge Captured • Evaluations • SECI
Skills <ul style="list-style-type: none"> • Project Management • Communication • Interpersonal • Leadership 	Skills <ul style="list-style-type: none"> • Support • Technical • Internet • Application Software 	Skills <ul style="list-style-type: none"> • SECI: Socialization, Externalization, Combination, Integration • Explicit/Tacit Knowledge of the Project • Willingness to Learn

Gambar 2.1 Usulan Penerapan Kerangka *Knowledge Management* untuk Facebook

Alasan mengapa mereka menggunakan facebook adalah dikarenakan lingkungan Facebook yang mampu menciptakan lingkungan kerja yang menyenangkan. Tidak hanya itu, dengan media sosial ini proses *knowledge management* seperti *socialization, externalization, combination and integration* akan terjadi secara alami melalui diskusi tersebut. Setiap komunikasi yang sebelumnya menggunakan e-mail akan dipindahkan ke ke dalam Diskusi Facebook, seperti pada gambar 2.2.

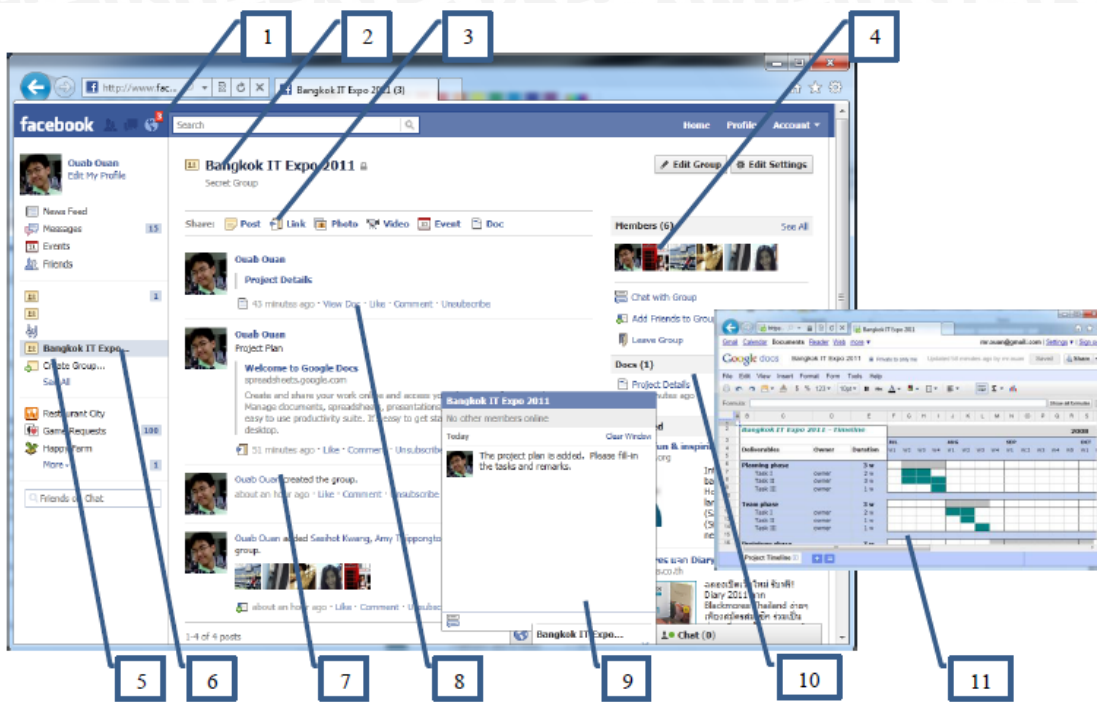


Figure 4. An example of using a Facebook Group page and Google Docs as a platform for project management and communications.

Gambar 2.2 Jejaring Facebook digunakan untuk *Knowledge Management*

Berikut merupakan penjelasan mengenai nomor – nomor yang tercantum pada gambar diatas :

- [No.1] Anggota tim dapat mengikuti halaman grup dengan memperhatikan pemberitahuan tanpa membuang-buang waktu mereka di Facebook saat bekerja.
- [No.2] Nama Grup yang telah dibuat didalam diskusi facebook.
- [No.3] Setiap informasi yang terkait dengan pengetahuan juga dapat dibagi melalui media seperti Links, Video, dll.
- [No.4] Seluruh anggota tim members.
- [No.5 dan 6] Membuat grup didalam diskusi facebook.
- [No.7] Setiap informasi dapat diposting di Facebook via Post.
- [No.8 dan 10] Setiap member juga dapat mengupload dokumen ke dalam grup.
- [No.9] Setiap member juga dapat berdiskusi dengan cara berkomunikasi melalui media chat.
- [N0.11] Lampiran dokumen yang diupload dengan menggunakan Google Documents.

Permasalahan yang terjadi pada PT. Petrokimia Gresik adalah tidak adanya suatu aplikasi pada *knowledge management system* mereka yang mampu untuk

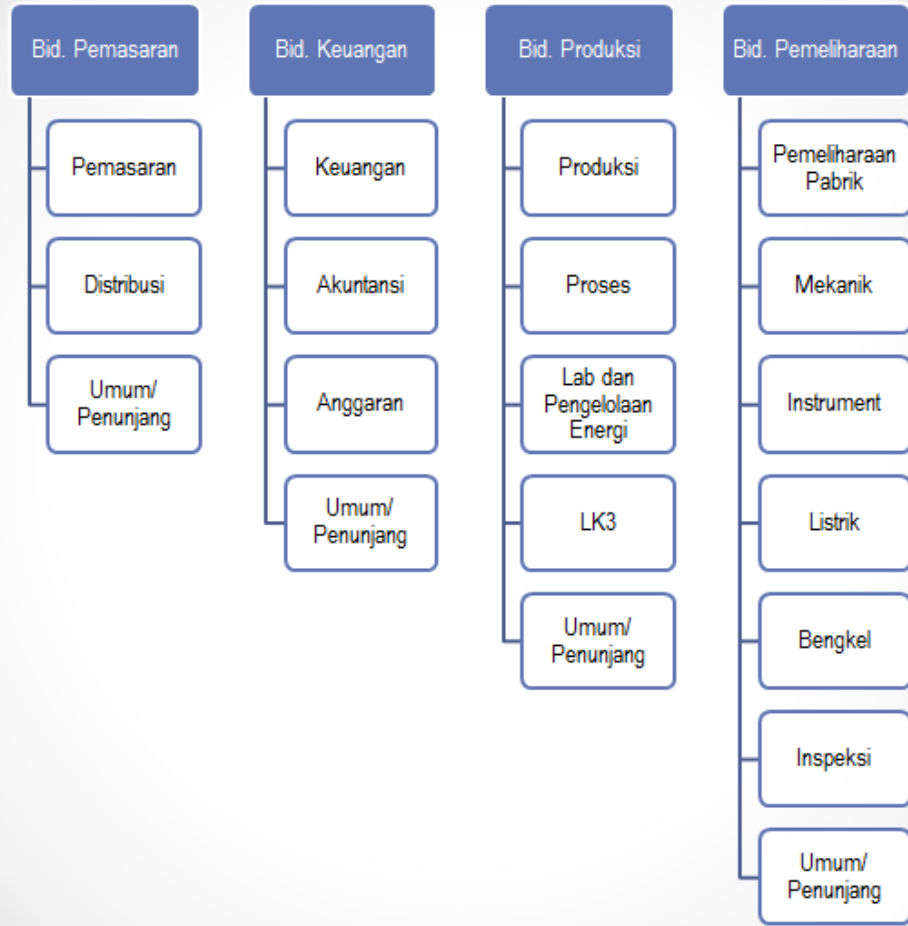
membedakan *talent* (kemampuan) antara karyawan satu dengan yang lainnya. Kemudian tidak adanya wadah bagi setiap karyawan untuk bisa dapat berinteraksi dengan seseorang yang dikategorikan sebagai seorang *expert* didalam perusahaan tersebut, sesuai dengan bidang pekerjaannya masing-masing. Hal ini yang menjadikan setiap karyawan tidak mampu untuk bisa berkembang dan juga meng-*explore* lebih jauh lagi kemampuan yang dimilikinya, itu disebabkan karena mereka tidak mengetahui kepada siapa mereka harus belajar, berbagi ilmu maupun bertukar informasi. Oleh karena itu dalam pembuatan aplikasi *expert locator* ini nantinya akan disesuaikan dengan pemanfaatan facebook dalam hal *sharing knowledge*. Hal itu seperti yang sudah dijelaskan sebelumnya bahwa facebook mampu menciptakan lingkungan kerja yang menyenangkan. Tidak hanya itu saja, dengan aplikasi tersebut proses *knowledge management* seperti *socialization*, *externalization*, *combination and integration* akan terjadi secara alami melalui diskusi tersebut.

Dari hasil penelitian diatas yang dijadikan sebagai kajian pustaka dapat disimpulkan bahwa *knowledge management* dapat berlangsung dengan sukses dalam pengimplementasiannya dengan menyelaraskan elemen *knowledge management*, bisnis dan TI yang sesuai berdasarkan kerangka yang sudah dibuat, serta menggunakan facebook sebagai saluran komunikasi utamanya. Hal itulah yang juga nantinya akan diharapkan dalam pembuatan aplikasi *expert locator* ini.

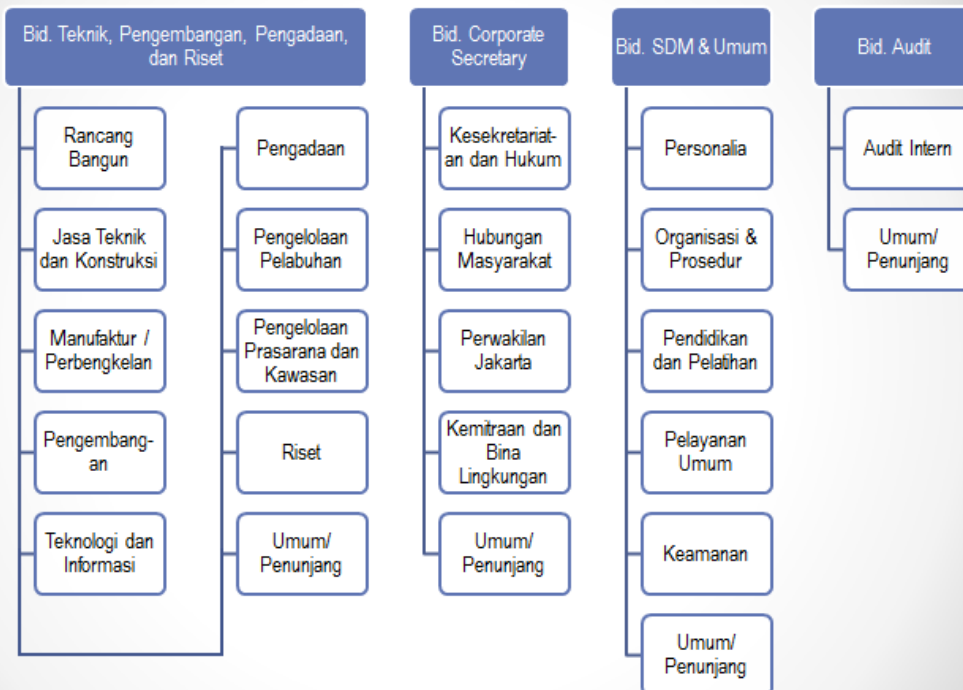
2.2 Kamus Kompetensi

Kamus kompetensi merupakan suatu informasi mengenai pengklasifikasian dari semua bidang yang ada di perusahaan. Informasi dari kamus kompetensi ini nantinya akan dijadikan sebagai landasan untuk mengklasifikasikan group yang ada didalam aplikasi *expert locator* ini. Berikut adalah kamus kompetensi yang ada di PT. Petrokimia Gresik.[PGS-12]. Seperti yang ditunjukkan pada gambar 2.3.

Kamus Kompetensi (1)



Kamus Kompetensi (2)



Gambar 2.3 Kamus Kompetensi

2.3 Knowledge Management System Saat Ini

Pada saat ini di PT. Petrokimia Gresik sebenarnya sudah ada sistem yang digunakan dalam melakukan sharing pengetahuan. Sistem tersebut berada pada *knowledge management system* mereka yaitu berupa website KM. Dalam melakukan *sharing knowledge*, setiap karyawan terlebih dahulu harus melakukan proses *upload* mengenai informasi *knowledge* yang dimilikinya ataupun yang mereka dapatkan dari orang lain ke dalam sebuah website KM tersebut. Setiap informasi yang dibagikan dapat berupa artikel, dokumen maupun video. Keseluruhan informasi yang sudah di-*sharing* tersebut nantinya bisa dapat di lihat, dibaca maupun diunduh oleh keseluruhan karyawan lainnya. Hal ini bertujuan untuk membantu semua karyawan dalam mendapatkan suatu informasi mengenai *knowledge* yang dibutuhkan dengan cepat dan juga sesuai dengan bidang pekerjaannya masing – masing ataupun yang dapat menunjang dalam menyelesaikan pekerjaannya.

Pada sistem yang ada di dalam *knowledge management system* saat ini juga terdapat beberapa kebutuhan – kebutuhan yang ada di dalamnya yang sudah di analisis diantaranya :

1. Knowledge Inventory

Berdasarkan dari hasil wawancara yang telah dilakukan pada bagian departemen teknologi informasi, diperoleh hasil identifikasi *inventory* yang terdapat pada perusahaan tersebut. Hasil wawancara *user* tersebut juga merupakan tahapan untuk mengetahui *knowledge needs*, yaitu *knowledge* apa saja yang biasa digunakan dalam *knowledge management system* saat ini. Dari hasil analisis *knowledge needs* tersebut diperoleh sebuah tabel *knowledge inventory*, seperti yang di tunjukkan pada tabel 1.1.

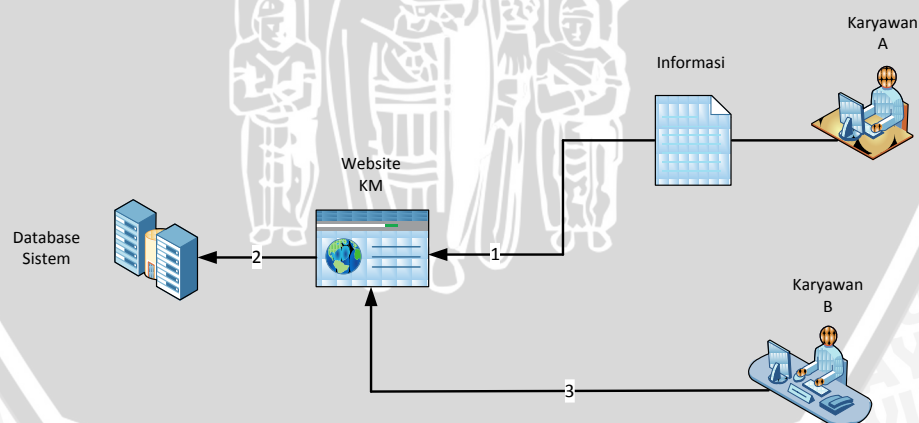
Tabel 1.1 Tabel *Knowledge Inventory*

No.	Kategori	Format Data	Pemilik
1.	ADM & Manajemen	Artikel, Dokumen, Video	Top Management
2.	Engineering	Artikel, Dokumen, Video	Komp. Engineering
3.	Keuangan & Akuntansi	Artikel, Dokumen, Video	Komp. Keuangan

4	Lingkungan dan K3	Artikel, Dokumen, Video	Komp. LK3
5.	Pemasaran	Artikel, Dokumen, Video	Komp. Penjualan & Pemasaran
6.	Pemeliharaan	Artikel, Dokumen, Video	Dept. Pemeliharaan
7.	Produksi	Artikel, Dokumen, Video	Dept. Produksi
8.	Riset	Artikel, Dokumen, Video	Komp. Riset
9.	SDM (Personalia & Diklat)	Artikel, Dokumen, Video	Komp. SDM
10.	Teknologi Informasi	Artikel, Dokumen, Video	Dept. TI

2. Knowledge Flow

Di dalam *knowledge management system* terdapat *knowledge flow* atau aliran *knowledge*. Dibuatnya aliran *knowledge* ini yaitu untuk mengetahui bagaimana orang-orang dalam suatu organisasi menemukan pengetahuan yang mereka butuhkan, dimanakah tempat *knowledge* itu disimpan, dan bagaimana mereka berbagi pengetahuan yang mereka miliki. Pada gambar 2.4 merupakan diagram alir *knowledge* yang terdapat pada perusahaan.



Gambar 2.4 Knowledge Flow

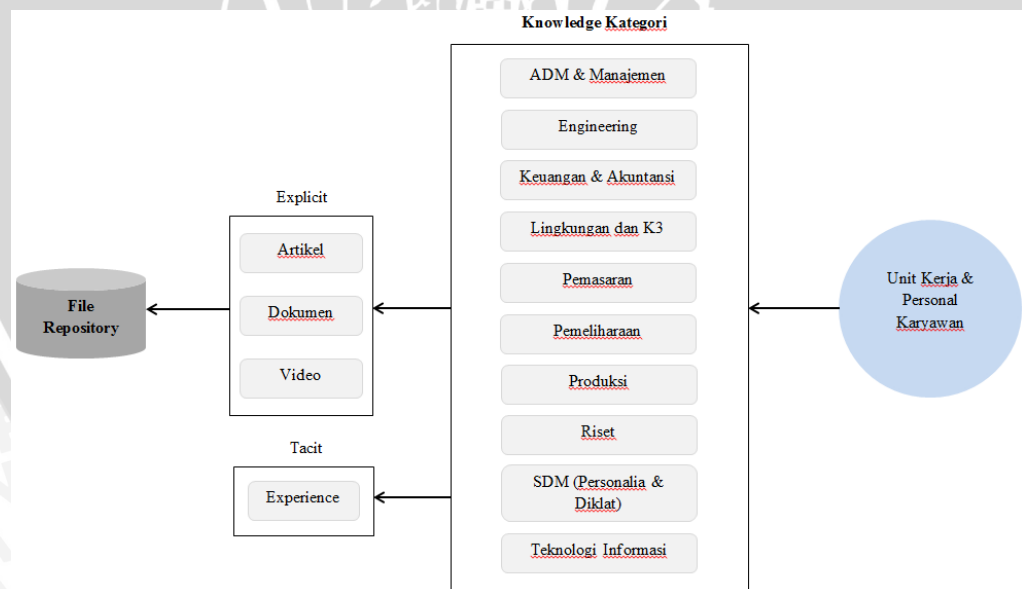
Penjelasan:

Pada gambar diatas menjelaskan mengenai proses aliran *knowledge* yang ada pada perusahaan, antara lain :

- A. [No.1] Karyawan A melakukan *sharing knowledge* dengan cara meng-*upload* sebuah informasi *knowledge* yang dimilikinya ke dalam sebuah website KM.
- B. [No.2] Informasi tersebut kemudian akan disimpan ke dalam sebuah file repository yang ada pada database sistem.
- C. [No.3] Kemudian karyawan B melakukan akses terhadap website KM tersebut, setelah itu mereka dapat melihat, membaca dan juga mengunduh informasi mengenai *knowledge* yang sudah di *sharing* sebelumnya oleh karyawan A.

3. *Knowledge Mapping*

Knowledge Mapping merupakan tahapan akhir dari proses ini. Hasil wawancara yang sudah diperoleh sebelumnya, kemudian diolah sehingga menghasilkan beberapa informasi *knowledge* yang bisa di klasifikasikan sehingga informasi *knowledge* tersebut bisa dipetakan berdasarkan klasifikasinya. Pada gambar 2.5 merupakan peta *knowledge* yang dibentuk berdasarkan tabel *knowledge inventory* yang sudah dibuat sebelumnya.



Gambar 2.5 *Knowledge Mapping*

2.4 Profil Perusahaan

PT Petrokimia Gresik merupakan pabrik pupuk terlengkap di Indonesia, yang pada awal berdirinya disebut Proyek Petrokimia Surabaya. Kontrak pembangunannya ditandatangani pada tanggal 10 Agustus 1964, dan mulai berlaku pada tanggal 8 Desember 1964. Proyek ini diresmikan oleh Presiden Republik Indonesia pada tanggal 10 Juli 1972, yang kemudian tanggal tersebut ditetapkan sebagai hari jadi PT Petrokimia Gresik. PT Petrokimia Gresik menempati lahan seluas 450 hektar berlokasi di Kabupaten Gresik, Propinsi Jawa Timur. PT Petrokimia Gresik memproduksi berbagai macam pupuk, seperti: Urea, ZA, SP-36, NPK Phonska, DAP, NPK Kebomas, ZK dan pupuk organik yaitu Petroganik. PT Petrokimia Gresik juga telah memproduksi produk non pupuk seperti Asam Sulfat, Asam fosfat, Amoniak, Dry Ice, Aluminum Fluoride, Cement Retarder, dll. Keberadaan PT Petrokimia Gresik adalah untuk mendukung program Pemerintah meningkatkan produksi pertanian nasional.[PGS-12].

2.5 Visi dan Misi

2.5.1 Visi

Menjadi produsen pupuk dan produk kimia lainnya yang berdaya saing tinggi dan produknya paling diminati konsumen.

2.5.2 Misi

1. Mendukung penyediaan pupuk nasional untuk tercapainya program swasembada pangan.
2. Meningkatkan hasil usaha untuk menunjang kelancaran kegiatan operasional dan pengembangan usaha perusahaan.
3. Mengembangkan potensi usaha untuk mendukung industri kimia nasional dan berperan aktif dalam *community development*.

2.6 Knowledge

Knowledge merupakan informasi yang dimiliki setiap pemikiran individu atau seseorang yang berupa informasi pribadi yang terkait dengan fakta-fakta, prosedur, konsep, interpretasi, ide, pengamatan, dan penilaian. Suatu pengetahuan harus ada sebelum informasi dapat dirumuskan dan sebelum data dapat diukur untuk membentuk informasi.[BEF-10].

2.6.1 Tipe Knowledge

Di dalam knowledge itu sendiri terdapat beberapa pengklasifikasian tipe, yang dimana setiap tipe membutuhkan penanganan yang berbeda, salah satunya adalah

2.6.1.2 Tacit & Explicit Knowledge

Tacit & Explicit Knowledge adalah *knowledge* yang berdasarkan sebagai *Tacit* dan *Explicit*. Pengetahuan tacit (*tacit knowledge*) merupakan pengetahuan yang tidak begitu mudah diungkapkan. Hal ini sangat pribadi, dan juga sulit untuk merumuskannya. Pengetahuan ini bisa juga berupa pengalaman dan keahlian dari orang-orang yang belum terdokumentasikan. Oleh karenanya itu *knowledge* seperti ini sulit untuk dibagikan. Pengetahuan eksplisit (*explicit knowledge*) merupakan pengetahuan yang bersifat formal dan sistematis. Hal ini dapat dengan mudah dikomunikasikan dan juga dibagikan. Pengetahuan seperti ini biasanya telah didokumentasikan. Artikulasi dalam pengetahuan ini, itu diungkapkan dan dicatat sebagai kata-kata, angka, kode, rumus matematika dan ilmiah.

Dalam pengaplikasiannya, *tacit knowledge* dan *explicit knowledge* mempunyai penanganan yang berbeda. Akan tetapi supaya *tacit knowledge* dapat ditangkap menjadi sebuah data yang nantinya dapat dipadukan menjadi sebuah informasi, maka dapat dilakukan pengkonversian dari *tacit knowledge* ke *explicit knowledge*. [BEF-10].

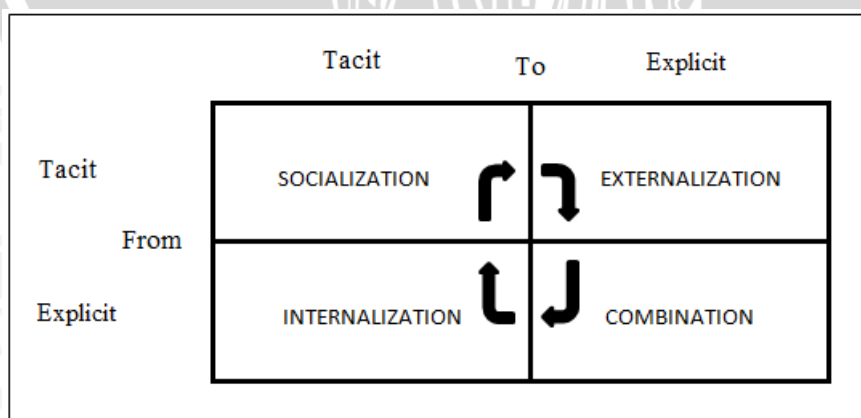
2.7 Knowledge Management

Knowledge Management merupakan strategi untuk mendapatkan pengetahuan yang benar kepada orang yang tepat pada waktu yang tepat dan membantu orang berbagi dan menempatkan informasi ke dalam tindakan dengan cara yang berusaha untuk meningkatkan kinerja organisasi. *Knowledge Management* memberikan kemampuan untuk mencipta, mengkomunikasikan dan menerapkan pengetahuan yang diperlukan dan berguna bagi pencapaian semua jenis tujuan bisnis. Selain itu knowledge management juga berfungsi meningkatkan kemampuan organisasi untuk belajar dari lingkungannya dan menggabungkan pengetahuan ke dalam proses bisnis.

Knowledge management menjadi bidang yang penting dalam proses pembelajaran sebuah organisasi. Pengetahuan yang dimiliki oleh organisasi harus mampu memberikan kemajuan bagi organisasi itu sendiri. Agar organisasi dapat bertahan hidup, maka diwajibkan agar setiap orang yang ada di dalam organisasi sharing pengetahuan. Untuk itu dibutuhkan manajemen yang kuat agar pengetahuan tersebut mengakar di setiap individu dalam organisasi dan tidak hilang begitu saja dengan didukung infrastruktur untuk penyebaran informasi di lingkungan organisasi.[PUR-08].

2.8 Knowledge Management Life Cycle

Knowledge Management Life Cycle merupakan siklus yang mendukung dalam mengembangkan *knowledge* yang ada di dalam organisasi. *Knowledge* itu sendiri terbagi dalam 2 macam pengetahuan yaitu *tacit* dan *explicit*. Dalam konsepnya *knowledge management life cycle* dapat dilihat pada gambar 2.6.



Gambar 2.6 Pengkonversian *Tacit dan Explicit Knowledge*

2.8.1 *Tacit Knowledge ke Tacit Knowledge (Socialization)*

Mode ini memungkinkan konversi dari tacit knowledge melalui interaksi antar individu. Satu hal penting untuk dicatat di sini adalah bahwa seorang individu dapat memperoleh pengetahuan tacit tanpa bahasa. Magang bekerja dengan mentor mereka dan belajar keahlian bukan melalui bahasa, tetapi melalui observasi, dan juga praktek. Dalam lingkungan bisnis, pelatihan kerja (OJT) menggunakan prinsip yang sama. Kunci untuk memperoleh pengetahuan tacit adalah pengalaman. Tanpa beberapa bentuk pengalaman bersama, hal ini sangat sulit bagi orang untuk berbagi setiap 'proses berpikir lainnya. Pengetahuan tacit dipertukarkan melalui bergabung dengan kegiatan seperti kebersamaan, menghabiskan waktu hidup di lingkungan yang sama, daripada melalui instruksi tertulis atau lisan. Dalam prakteknya, sosialisasi melibatkan menangkap pengetahuan melalui kedekatan fisik. Proses memperoleh pengetahuan sebagian besar didukung melalui interaksi langsung dengan orang-orang.[BEF-10].

2.8.2 *Tacit Knowledge ke Explicit Knowledge (Externalization)*

Eksternalisasi membutuhkan ekspresi dari pengetahuan tacit dan juga terjemahannya ke dalam suatu bentuk yang mudah untuk dipahami serta dapat dimengerti oleh orang lain. Hal ini bisa dilakukan diantaranya dengan dokumentasi. Dalam prakteknya, eksternalisasi didukung oleh dua faktor kunci. Pertama, artikulasi tacit pengetahuan yaitu, konversi tacit ke pengetahuan eksplisit yang melibatkan teknik yang membantu untuk mengekspresikan ide-ide seseorang atau gambar sebagai kata-kata, konsep, bahasa kiasan (seperti analogi atau narasi) dan visual.[BEF-10].

2.8.3 *Explicit Knowledge ke Explicit Knowledge (Combination)*

Kombinasi merupakan pengkonversian pengetahuan eksplisit menjadi set yang lebih kompleks dari pengetahuan eksplisit. Maksudnya disini adalah suatu bentuk penggabungan antara *explicit knowledge* dengan *explicit knowledge* yang lainnya. Dalam prakteknya, fase kombinasi tergantung pada tiga proses yaitu :

1. Menangkap dan mengintegrasikan pengetahuan eksplisit baru, hal ini sangatlah penting. Dalam hal ini mungkin melibatkan pengumpulan pengetahuan externalized (misalnya data publik) dari dalam atau luar perusahaan dan menggabungkan data tersebut.

2. Penyebaran pengetahuan eksplisit didasarkan pada proses pengalihan bentuk pengetahuan langsung dengan menggunakan presentasi atau rapat. Disini pengetahuan baru tersebar di antara anggota organisasi.
3. Editing atau pengolahan pengetahuan eksplisit yang membuatnya lebih bermanfaat (misalnya dokumen seperti rencana, laporan, data pasar).
Dalam proses kombinasi memungkinkan organisasi untuk mengambil setiap langkah-langkah yang konkret. Dengan mengkonfigurasi informasi yang ada melalui penyortiran, penambahan, pengkategorian dan juga pembagian konteks pengetahuan eksplisit dapat menyebabkan pengetahuan baru. Hal inilah yang disebut sebagai kombinasi, proses menciptakan pengetahuan eksplisit dari pengetahuan eksplisit.[BEF-10].

2.8.4 *Explicit Knowledge ke Tacit Knowledge (Internalization)*

Internalisasi merupakan proses pengkonversian dari pengetahuan eksplisit ke dalam pengetahuan tacit di dalam organisasi. Hal ini memerlukan individu untuk mengidentifikasi pengetahuan yang relevan untuk diri seseorang dalam pengetahuan organisasi. Belajar dengan melakukan pelatihan akan memungkinkan individu untuk mengakses ranah pengetahuan kelompok dan seluruh organisasi. Dalam prakteknya, internalisasi bergantung pada dua dimensi, yaitu :

1. Pengetahuan eksplisit harus diwujudkan dalam tindakan dan praktek. Dengan demikian, proses internalisasi pengetahuan eksplisit mengaktualisasikan konsep atau metode tentang strategi, taktik, inovasi atau perbaikan. Sebagai contoh, program-program pelatihan dalam organisasi yang lebih besar akan membantu peserta untuk memahami organisasi dan diri mereka sendiri secara keseluruhan.
2. Proses mewujudkan pengetahuan eksplisit dengan menggunakan simulasi atau percobaan untuk memicu proses *learning by doing*. Sebagai contoh seseorang yang ingin belajar mengenai IT akan melihat buku tentang IT untuk kemudian dicobanya. Proses inilah yang disebut dengan internalisasi. [BEF-10].

2.9 Knowledge Management System

Knowledge Management System (KMS) merupakan sebuah sistem manajemen yang ada di dalam sebuah organisasi yang menguraikan, mengelola dan mengoordinasi penggunaan, informasi, pengetahuan serta pengalaman yang dimiliki oleh seseorang di dalam suatu organisasi. KMS mengacu pada kelas sistem informasi yang diterapkan untuk mengelola pengetahuan organisasi. Tujuan dari *knowledge management system* adalah untuk menemukan cara baru untuk menyalurkan informasi yang nantinya bermanfaat bagi organisasi dan juga pencapaian suatu target seperti pengetahuan bersama, peningkatan kerja serta penciptaan inovasi perusahaan.[BEF-10].

2.10 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (Software Engineering) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan (customer). Selain itu rekayasa perangkat lunak juga lebih berfokus pada bagaimana membuat perangkat lunak yang memenuhi kriteria berikut:

1. Dapat terus dipelihara setelah perangkat lunak selesai dibuat seiring berkembangnya teknologi dan lingkungan (maintainability).
2. Dapat diandalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (dependability dan robust).
3. Efisien dari segi sumber daya dan penggunaan.
4. Kemampuan untuk dipakai sesuai dengan kebutuhan (usability).

Dari kriteria di atas maka perangkat lunak yang baik adalah perangkat lunak yang dapat memenuhi kebutuhan pelanggan (customer) atau user (pemakai perangkat lunak), atau berorientasi pada pelanggan atau pemakai perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak. [ROS-11].

2.11 Unified Modeling Language (UML)

2.11.1 Konsep Dasar UML (Unified Modelling Language)

UML atau *Unified Modeling Language* adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak atau software. UML tidak berdasarkan pada bahasa pemrograman tertentu sehingga dapat digunakan untuk spesifikasi standar semua bahasa pemrograman baik yang desktop maupun berbasis web. Standar spesifikasi UML dijadikan standar secara *defacto* oleh OMG (*Object Management Group*) pada tahun 1997.[ROS-11].

2.11.2 Use Case Diagram

Use case diagram merupakan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Terdapat 2 hal yang utama pada *use case*, yaitu *actor* dan *use case*. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat. Jadi, *actor* belum tentu merupakan orang. Sedangkan, *use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*. [ROS-11].

2.11.3 Activity Diagram

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

Diagram aktifitas juga digunakan untuk mendefinisikan hal-hal berikut :

- 1) Rancangan proses bisnis dimana setiap urutan aktifitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- 2) Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- 3) Rancangan pengujian dimana setiap aktifitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.[ROS-11].

2.11.4 *Sequence Diagram*

Diagram Sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antarobjek. Untuk itu dalam menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. [ROS-11].

2.11.5 *Class Diagram*

Class Diagram menggambarkan struktur sistem dari kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut yaitu variabel-variabel yang dimiliki oleh suatu kelas dan metode atau operasi yaitu fungsi-fungsi yang dimiliki oleh suatu kelas. Struktur kelas yang baik pada *class diagram*, sebaiknya memiliki jenis-jenis kelas seperti berikut:

- 1) Kelas utama, memiliki fungsi awal ketika sistem dijalankan.
- 2) Kelas yang menangani tampilan sistem, untuk mengatur tampilan.
- 3) Kelas yang diambil dari pendefinisian *use case*, menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*.
- 4) Kelas yang diambil dari pendefinisian data, digunakan untuk memegang data menjadi sebuah kesatuan yang diambil maupun akan disimpan di *database*. [ROS-11].

2.12 Software Development Life Cycle (SDLC)

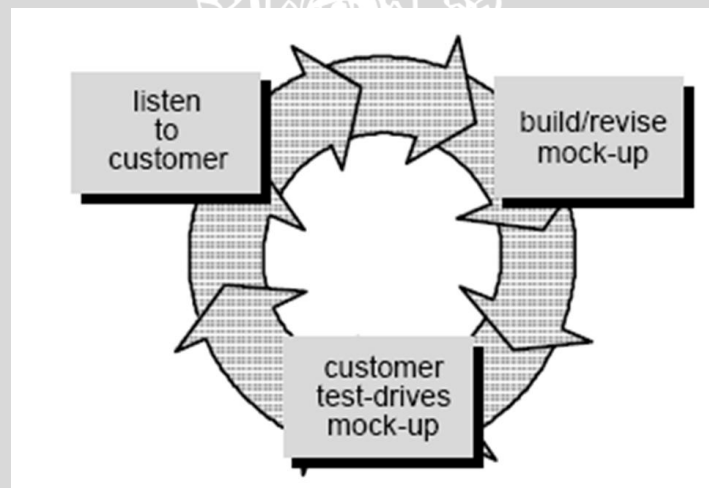
SDLC atau Software Development Life Cycle atau sering disebut juga System Development Life Cycle adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya.

2.12.1 Model SDLC

Ada beberapa model SDLC yang dapat digunakan. Semuanya memiliki kelemahan dan kelebihan pada setiap model SDLC. Hal terpenting adalah mengenali tipe pelanggan (*customer*) dan memilih menggunakan model SDLC yang sesuai dengan karakter pelanggan (*customer*) dan sesuai dengan karakter pengembang. Salah satunya adalah Model Prototipe yang akan digunakan dalam penelitian ini.[ROS-11].

2.12.1.1 Model Prototipe

Model prototipe merupakan suatu model yang dapat digunakan untuk menyambungkan ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Pada gambar 2.6 merupakan gambaran dari model prototipe :



Gambar 2.7 Model Prototipe

Model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Model prototipe ini sangat cocok digunakan untuk menjabarkan kebutuhan pelanggan secara lebih detail karena

pelanggan sering kali kesulitan menyampaikan kebutuhannya secara detail tanpa melihat gambaran yang jelas.[ROS-11].

2.13 PHP

PHP atau **Hypertext Preprocessor** merupakan bahasa berbentuk script yang ditempatkan dalam server dan dieksekusi di dalam server untuk selanjutnya ditransfer dan dibaca oleh client. Php juga bisa disisipkan dalam bahasa HTML. [Andre Adelheid & Khairil Nst:2012:2]. Pemrograman yang berjalan pada server banyak sekali. Setiap program mempunyai kelebihan dan kekurangan. Saat ini banyak website yang menggunakan program PHP sebagai dasar pengolahan data. Beberapa keunggulan yang dimiliki program PHP adalah :

1. PHP memiliki tingkat akses yang lebih cepat.
2. PHP memiliki tingkat lifecycle yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
3. PHP memiliki tingkat keamanan yang tinggi.
4. PHP mampu berjalan di beberapa server yang ada, misalnya Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhttpd, dan Xitami.
5. PHP mampu berjalan di Linux sebagai platform sistem operasi utama bagi PHP, namun juga dapat berjalan di FreeBSD, Unix, Solaris, Windows, dan yang lain.
6. PHP juga mendukung akses ke beberapa database yang sudah ada, baik yang bersifat free/gratis ataupun komersial. Database itu antara lain MySQL, PostgreSQL, mSQL, Informix, dan MicrosoftSQL Server.
7. PHP bersifat free atau gratis.[AND-04].

2.14 Framework CodeIgniter

CodeIgniter adalah aplikasi open source yang berupa *framework* dengan model MVC(*Model, View, Controller*) untuk membangun *website* dinamis. Dengan menggunakan PHP CodeIgniter akan memudahkan *developer* untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi / tampilan, sehingga tugas yang dikerjakan lebih mudah dipecah – pecah, ada bagian yang membuat tampilan dan membuat bagian programnya.

- a. Model mempresentasikan struktur data yang dibangun. Kelas model berisi fungsi-fungsi yang membantu *developer* untuk mengelola, memasukkan dan mengupdate informasi dalam database.
- b. View merupakan informasi yang disajikan untuk *user* yang berupa halaman web itu sendiri atau *user interface*. Tetapi dalam CodeIgniter, view dapat juga menjadi bagian-bagian halaman seperti *header* atau *footer*.
- c. Controller bertugas sebagai penghubung antara Model, View, dan beberapa sumberlainnya yang dibutuhkan untuk memproses *HTTP request* dan untuk meng-*generate* sebuah halaman web.[SEP-11].

2.15 MYSQL

MySQL merupakan software RDBMS (atau server database) yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (multi-user), dan dapat melakukan suatu proses secara sinkron atau berbarengan (multi-threaded). Lisensi MySQL terbagi menjadi dua yaitu MySQL sebagai produk open source dibawah GNU *General Public License* (gratis) dan lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan-kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah ke bawah, versi gratis masih dapat digunakan dengan baik.[RAH-11].

Berikut ini beberapa alasan mengapa mereka memilih MySQL sebagai server database untuk aplikasi-aplikasi yang mereka kembangkan :

1. Fleksibel
2. Performa Tinggi
3. Lintas Platform
4. Gratis
5. Proteksi Data yang Handal
6. Komunitas Luas

2.16 Pengujian Perangkat Lunak

Pengujian Perangkat Lunak merupakan satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi program yang diinginkan. Aktifitas pengujian terdiri dari sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian. Pengujian dilakukan dengan cara

mengeksekusi program dengan tujuan untuk menemukan kesalahan dan juga mengevaluasi kualitas perangkat lunak.[ROS-11].

2.16.1 Teknik Pengujian Perangkat Lunak

Teknik Pengujian Perangkat Lunak adalah metode yang beragam untuk melakukan pengujian perangkat lunak. Teknik pengujian mengacu pada metode yang berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan. Kita harus memastikan bahwa kita memilih teknik yang akan membantu untuk memastikan pengujian yang paling efisien dan efektif dari sistem.[ROS-11].

Meode perancangan untuk kasus uji yang nantinya akan dipakai dalam penelitian ini adalah *Black-Box Testing* yaitu *functional testing* dan *usability testing*.

2.16.1.1 Black-Box Testing

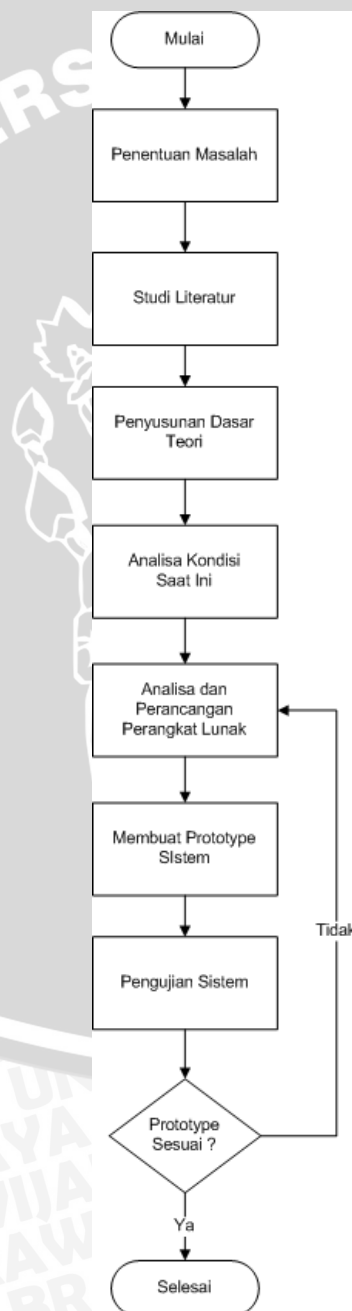
Black-Box Testing merupakan pengujian perangkat lunak berdasarkan dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Fokus semata-mata pengujian hanya pada output yang dihasilkan. Pengujian *black-box* memainkan peranan penting dalam pengujian perangkat lunak, hal itu membantu untuk validasi fungsi dari keseluruhan sistem. Pengujian *black-box* dilakukan dengan cara membuat sebuah kasus uji yang bersifat mencoba keseluruhan fungsi dengan memakai perangkat lunak untuk mengetahui apakah sudah sesuai dengan spesifikasi yang dibutuhkan, apabila tidak sesuai maka dapat dengan mudah diidentifikasi dan dapat diatasi kemudian.[NID-12].

BAB III

METODE PENELITIAN DAN PERANCANGAN

Bagian ini menjelaskan langkah – langkah yang akan ditempuh dalam penyusunan skripsi, antara lain penentuan masalah, studi literatur, penyusunan dasar teori, analisa kondisi saat ini, analisa dan perancangan perangkat lunak, membuat *prototype* sistem, pengujian, serta pengambilan kesimpulan dan saran. Oleh karena itu dibuat sebuah alur kerja secara umum, seperti pada gambar 3.1:

3.1 Metodologi Penelitian



Gambar 3.1 Alur Pengerjaan Secara Umum

3.1.1 Penentuan Masalah

Penentuan masalah merupakan tahap awal dalam pelaksanaan penelitian yang meliputi identifikasi bidang masalah serta penentuan pemilihan pokok masalah. Kegiatan ini dimaksudkan sebagai upaya untuk mendapatkan masalah yang baik untuk diteliti. Untuk mendapatkan masalah yang baik, penulis harus terlebih dahulu mengidentifikasi masalah tersebut secara spesifik yang difokuskan pada hal yang khusus sehingga secara mendalam dan terinci dapat diidentifikasi karakteristik tentang apa yang akan diteliti.

3.1.2 Studi Literatur dan Penyusunan Dasar Teori

Pada tahap ini dilakukan studi literatur terhadap berbagai hal yang berhubungan dengan pembuatan sistem, seperti penjelasan dasar teori yang digunakan untuk menunjang penulisan skripsi. Dalam proses penyelesaian penelitian ini, penulis menggunakan berbagai sumber dan referensi sebagai bahan acuan, seperti referensi buku, jurnal, laporan penelitian, skripsi yang sudah ada dan sumber lain baik cetak maupun *online*. Sedangkan, teori-teori pendukung yang digunakan meliputi :

1. Kajian Pustaka
2. Kamus Kompetensi
3. Profil Perusahaan
4. *Knowledge*
 - A. Tipe *Knowledge*
 - a) *Tacit & Explicit Knowledge*
5. *Knowledge Management*
6. *Knowledge Management Life Cycle*
 - a) *Tacit Knowledge ke Tacit Knowledge (Socialization)*
 - b) *Tacit Knowledge ke Explicit Knowledge (Externalization)*
 - c) *Explicit Knowledge ke Explicit Knowledge (Combination)*
 - d) *Explicit Knowledge ke Tacit Knowledge (Internalization)*
7. *Knowledge Management System*
8. Rekayasa Perangkat Lunak
9. UML (*Unified Modeling Language*)
 - a. Konsep Dasar UML
 - b. *Use Case Diagram*

- c. *Activity Diagram*
 - d. *Sequence Diagram*
 - e. *Class Diagram*
10. SDLC
 - A. Model SDLC
 - a) Model Prototipe
 11. PHP
 12. *Framework CodeIgniter*
 13. MySQL
 14. Pengujian Perangkat Lunak

3.1.3 Analisa Kondisi Saat Ini

Analisa ini dilakukan untuk mengetahui terlebih dahulu *knowledge management system* seperti apakah yang digunakan pada saat ini di PT. Petrokimia Gresik yang sudah dijelaskan sebelumnya di bab 2. Pada tahap ini dilakukan identifikasi mengenai informasi yang berdasarkan kebutuhan *knowledge* pada organisasi. Kegiatan identifikasi ini dilakukan dengan cara mengidentifikasi *knowledge inventory*, *knowledge flow*, serta *knowledge mapping* yang terdapat di perusahaan tersebut. Dari hasil identifikasi tersebut diharapkan akan menghasilkan prasyarat *expert locator* yang dibutuhkan. Setelah itu nantinya akan dibuatkan suatu sistem yang baru.

3.1.4 Analisa dan Perancangan Perangkat Lunak

Analisa kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk merancang aplikasi perangkat lunak di PT. Petrokimia Gresik sesuai proses bisnis yang ada. Setelah kebutuhan yang diperlukan sudah ditentukan, maka selanjutnya dilakukan perancangan sistem. Kebutuhan yang digunakan dalam perancangan didapatkan melalui wawancara langsung kepada instansi PT. Petrokimia Gresik bagian departemen teknologi informasi. Dan metode analisis yang digunakan adalah *Object Oriented analysis* dengan menggunakan bahasa pemodelan UML (*Unified Modelling Language*).

3.1.5 Membuat *Prototype* Sistem

Setelah menentukan jenis kebutuhan dan desain atau perancangan sistem, maka tahap selanjutnya adalah membuat *prototype* sistem. Pada tahap ini dilakukan proses *coding* sistem dengan menggunakan komponen pendukung dalam pembuatan sistem. *Prototype* sistem yang akan dibuat berupa sistem berbasis web dengan menggunakan bahasa pemrograman PHP, *framework codeigniter* dan *database MySQL*.

3.1.6 Pengujian Sistem

Fokus pengujian yang dilakukan pada sistem ini adalah memastikan bahwa semua bagian sudah tidak terjadi kesalahan atau *error*. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan. Pengujian ini sendiri akan diujikan pada instansi yang bersangkutan. Pengujian dilakukan dengan dua cara, yaitu:

1. Pengujian Fungsional

Digunakan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan yang dibutuhkan. Item-item yang telah dirumuskan dan diterapkan dalam daftar kebutuhan dan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi (menguji tombol atau link yang ada di sistem).

2. Pengujian Non-Fungsional

Pengujian ini berfokus hanya menguji kenyamanan dan kemudahan penggunaan sistem dengan menggunakan *usability testing* dimana pengujian ini memastikan bahwa aplikasi mudah dan nyaman saat digunakan oleh pengguna.

3.1.7 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi, dan pengujian sistem selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibangun. Tahap terakhir dari penulisan ini adalah saran yang dimaksudkan untuk memperbaiki kesalahan – kesalahan yang terjadi dan menyempurnakan penulisan serta untuk memberikan pertimbangan dalam hal pengembangan aplikasi selanjutnya.

3.2 Analisa Kondisi Saat Ini

Analisa ini dilakukan untuk mengetahui terlebih dahulu *knowledge management system* seperti apakah yang digunakan pada saat ini di PT. Petrokimia Gresik yang sudah dijelaskan sebelumnya di bab 2. Pada tahap sebelumnya sudah dilakukan identifikasi mengenai informasi yang berdasarkan kebutuhan *knowledge* pada organisasi. Kegiatan identifikasi tersebut dilakukan dengan cara mengidentifikasi *knowledge inventory*, *knowledge flow*, serta *knowledge mapping* yang terdapat di perusahaan tersebut. Dari hasil identifikasi tersebut diharapkan akan menghasilkan prasyarat yang dibutuhkan untuk membuat aplikasi *expert locator*.

3.2.1 Analisis Untuk Pengembangan Fitur *Expert Locator* Pada KMS PT. Petrokimia Gresik

Berdasarkan dari hasil analisis pada sistem *knowledge management system* yang lama yang sudah dijelaskan sebelumnya di bab 2, maka dapat disimpulkan bahwa di dalam sistem tersebut masih terdapat beberapa kekurangannya diantaranya adalah didalam web KM tersebut tidak mencantumkan informasi secara menyeluruh mengenai data diri dari setiap karyawan yang men-*share* informasi *knowledge* tersebut, apakah yang men-*share* informasi tersebut merupakan karyawan yang memang benar-benar *expert* dibidangnya atau hanya merupakan karyawan biasa pada umumnya. Selain itu di dalam web KM yang ada pada KMS mereka juga memiliki beberapa kekurangan yang diantaranya :

- a. Tidak adanya fitur yang mampu untuk membedakan *talent* (kemampuan) antara karyawan satu dengan yang lainnya.
- b. Tidak adanya fitur yang mampu untuk memberikan pilihan bagi setiap karyawan mengenai minat yang ingin dikembangkan daripada kemampuan yang dimilikinya sekarang.
- c. Tidak adanya fitur untuk bisa dapat secara langsung menemukan siapa saja karyawan yang *expert* dibidangnya masing-masing.
- d. Tidak adanya fitur untuk bisa dapat mengakses seseorang yang *expert* dibidangnya.

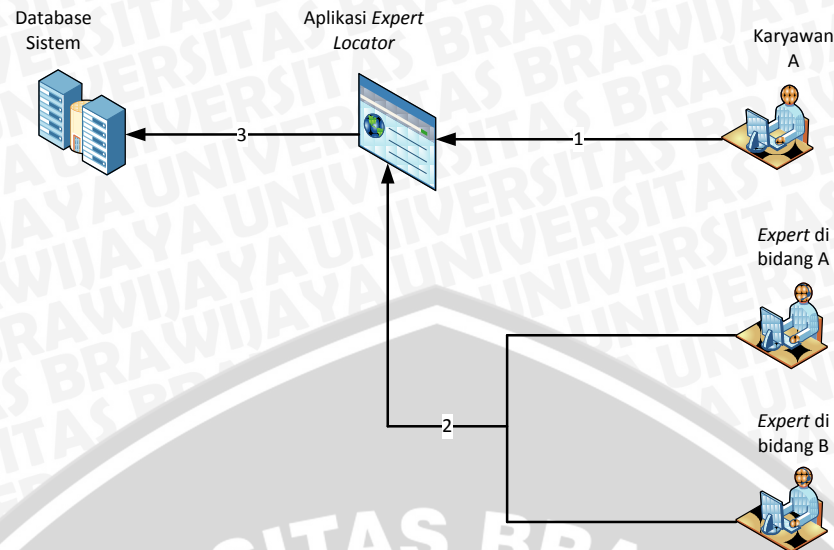
Oleh karena itu dalam pengembangan selanjutnya, akan dilakukan dengan cara membuat suatu aplikasi yaitu *expert locator* pada *knowledge management system* mereka. Aplikasi *expert locator* ini nantinya akan mempunyai fungsi sebagai wadah untuk membedakan *talent* setiap masing-masing karyawan. Tidak

hanya itu saja, aplikasi ini juga akan membantu mereka dalam hal berinteraksi antara karyawan satu dengan yang lainnya, serta memberikan peluang bagi karyawan untuk bisa dapat berkonsultasi dengan para ahli (*expert*) yang sudah dikategorikan sesuai dengan bidang pekerjaannya masing-masing dalam bentuk forum diskusi. Dengan adanya aplikasi *expert locator* ini diharapkan setiap karyawan mengetahui akan kemampuan yang dimilikinya, kemudian mengetahui kemampuan yang ingin dikembangkan, lalu mengetahui siapa ahlinya, dan juga mengetahui bagaimana cara mengakses ahlinya. Selain itu diharapkan juga dari penerapan aplikasi ini nantinya karyawan akan mampu berkembang dengan baik tidak hanya pada kemampuan yang dimilikinya saja, namun dapat berkembang lebih jauh lagi.

Aplikasi *expert locator* ini memiliki beberapa kelebihan diantaranya :

- a. Memberikan kemudahan bagi setiap karyawan untuk dapat berkomunikasi antara satu dengan yang lainnya terutama dengan para *expert*.
- b. Memberikan kemudahan bagi semua karyawan dalam hal melakukan *sharing knowledge*.
- c. Memberikan kemudahan bagi semua karyawan yang ingin mencari informasi mengenai *knowledge*.

Dalam pengaplikasiannya untuk penentuan *expert* itu sendiri sudah ditetapkan sebelumnya oleh pihak perusahaan yaitu apabila setiap karyawan mempunyai setidaknya 1 sertifikasi sesuai dengan bidangnya masing – masing maupun bidang yang lainnya. Pada gambar 3.2 merupakan gambaran mengenai aplikasi *expert locator*.



Gambar 3.2 Gambaran Umum Aplikasi *Expert Locator*

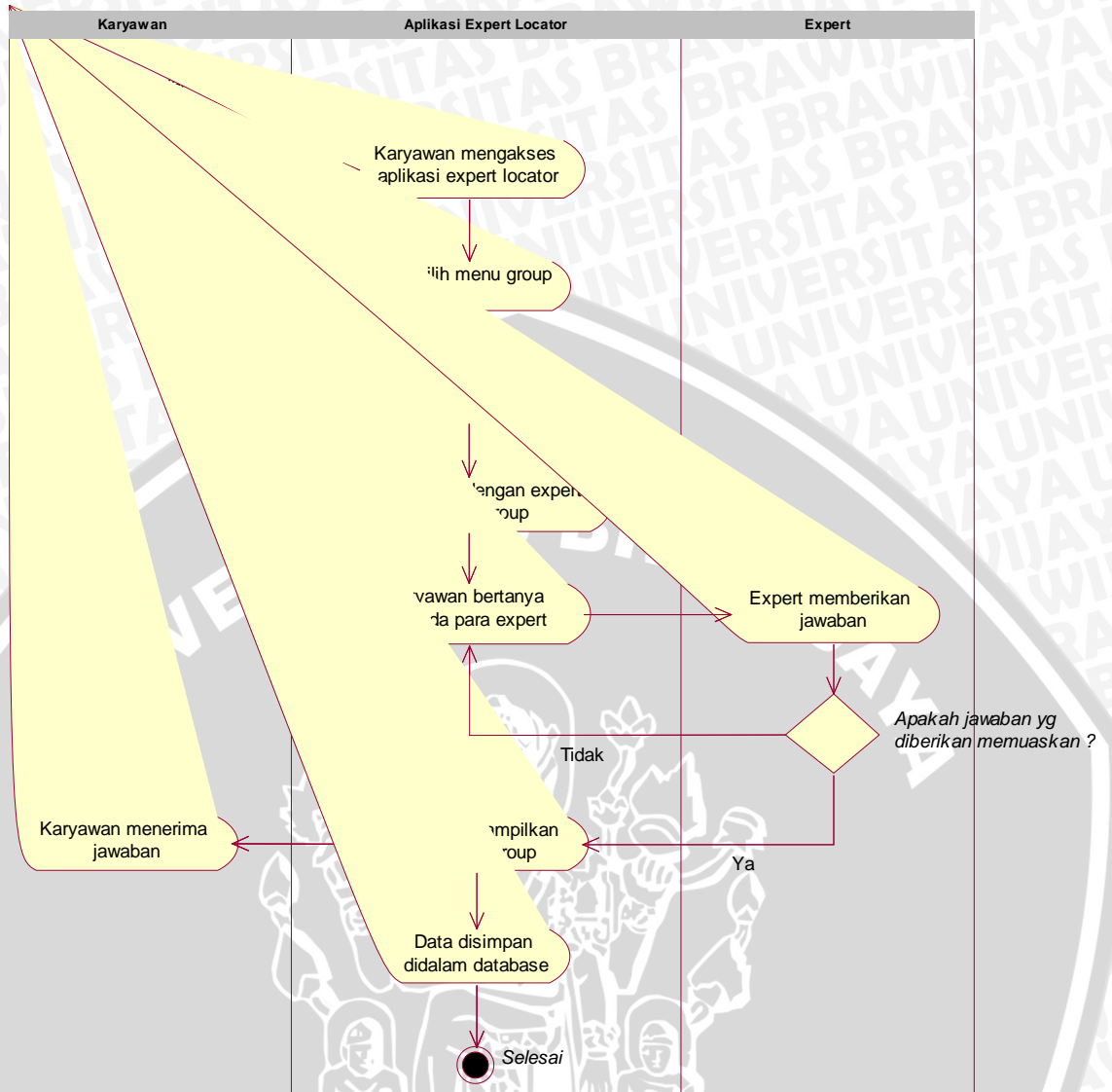
Penjelasan:

Pada gambar diatas menjelaskan mengenai proses aliran *knowledge* yang ada pada perusahaan, antara lain :

- A. [No.1] Karyawan A melakukan akses aplikasi *expert locator*, dia ingin berkonsultasi dengan para ahli (*expert*) dibidangnya masing – masing.
- B. [No.2] Para ahli (*expert*) A dan B juga mengakses aplikasi *expert locator*. Untuk memudahkan setiap karyawan dalam mencari para ahlinya, aplikasi tersebut juga sudah membedakan setiap *expertnya* sesuai dengan keahlian dibidangnya masing – masing.
- C. [No.3] Keseluruhan data yang ada pada aplikasi *expert locator* disimpan ke dalam database.

3.2.2 Bisnis Proses *Expert Locator*

Bisnis Proses merupakan serangkaian kumpulan pekerjaan yang terstruktur dan saling terkait antara satu dengan yang lainnya untuk menghasilkan suatu keluaran yang mendukung pada tujuan dan sasaran strategis dari organisasi itu sendiri. Suatu proses bisnis dapat di pisah-pisah menjadi beberapa subproses yang masing-masing memiliki atribut tersendiri, tetapi juga berkontribusi untuk mencapai tujuan dari subprosesnya. Berikut merupakan bisnis proses dari aplikasi *expert locator*, seperti yang ditunjukkan pada gambar 3.3.



Gambar 3.3 Bisnis Proses dari *Expert Locator*

3.3 Analisa dan Perancangan Perangkat Lunak

Analisa kebutuhan dilakukan dengan menentukan kebutuhan apa saja yang dibutuhkan untuk merancang aplikasi perangkat lunak di PT. Petrokimia Gresik sesuai proses bisnis yang ada. Setelah kebutuhan yang diperlukan sudah ditentukan, maka selanjutnya dilakukan perancangan sistem. Kebutuhan yang digunakan dalam perancangan didapatkan melalui wawancara langsung kepada instansi PT. Petrokimia Gresik bagian departemen teknologi informasi. Dan metode analisis yang digunakan adalah *Object Oriented Analysis* dengan menggunakan bahasa pemodelan UML (*Unified Modelling Language*).

3.3.1 Analisis Sistem

Analisis Sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bentuk bagian – bagian komponennya, yang dimana dimaksudkan untuk mengidentifikasi kebutuhan – kebutuhan yang dibutuhkan dalam pembuatannya, diantaranya :

3.1.1.1 Analisis User

Pada tahapan ini dilakukan analisis user yang nantinya diperoleh data *user* yang akan menggunakan sistem ini, diantaranya meliputi :

a. Karyawan

User yang memiliki hak akses untuk login dan melakukan proses yang ada di dalam aplikasi *expert locator* yang diantaranya : edit *profile*, kelola status, kelola komentar, *private message*, *join group*, dan *add friend*.

b. Admin

User yang bertugas dan memiliki hak akses untuk *login* serta melakukan operasi semua pengelolaan beberapa data diantaranya, mengelola data karyawan, data pendidikan, data training, data pengalaman kerja, data sertifikasi, data unit kerja, data user, data akses, data group, data status karyawan, data pesan, dan data album.

3.3.2 Analisis Kebutuhan Fungsional

Pada tahapan ini dilakukan analisis kebutuhan fungsional dari sistem, berikut ini merupakan kebutuhan fungsional pada aplikasi *expert locator*.

1. Fungsional Pengelolaan Akun

a. Login

Aktor : Admin *Expert Locator* dan Karyawan

Fungsi untuk memberikan suatu batasan tertentu pada *user* dalam memasuki halaman pada sistem.

Nama *use case* : Login

b. Logout

Aktor : Admin *Expert Locator* dan Karyawan

Fungsi untuk mengeluarkan *user* dari sistem dan kembali pada halaman utama sistem.

c. Kelola Data Karyawan

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data karyawan.

Nama *use case* : Kelola Data Karyawan

d. Kelola Data User

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data *user*.

Nama *use case* : Kelola Data User

e. Kelola Data Group

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data group yang ada pada sistem.

Nama *use case* : Kelola Data Group

f. Kelola Data Akses

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data akses dari masing – masing *user*.

Nama *use case* : Kelola Data Akses

2. Fungsional Pengelolaan Data

a. Kelola Data Unit Kerja

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data unit kerja.

Nama *use case* : Kelola Data Unit Kerja

b. Kelola Data Pendidikan

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data pendidikan.

Nama *use case* : Kelola Data Pendidikan

c. Kelola Data Training

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data training.

Nama *use case* : Kelola Data Training

d. Kelola Data Pengalaman Kerja

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data pengalaman kerja.

Nama *use case* : Kelola Data Pengalaman Kerja

e. Kelola Data Sertifikasi

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data sertifikasi.

Nama *use case* : Kelola Data Sertifikasi

f. Kelola Data Status Karyawan

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data status karyawan.

Nama *use case* : Kelola Data Status Karyawan

g. Kelola Data Pesan

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data pesan.

Nama *use case* : Kelola Data Pesan

h. Kelola Data Album

Aktor : Admin *Expert Locator*

Fungsi untuk melihat, menambah, mengedit dan juga menghapus data album.

Nama *use case* : Kelola Data Album.

i. Edit Profil

Aktor : Karyawan

Fungsi untuk mengedit keseluruhan data profil dari masing – masing karyawan.

Nama *use case* : Edit Profil

j. Kelola Status

Aktor : Karyawan

Fungsi untuk melihat, menambah, mengedit dan juga menghapus status dari masing – masing karyawan.

Nama *use case* : Kelola Status.

k. Kelola Komentar

Aktor : Karyawan

Fungsi untuk melihat, menambah, mengedit dan juga menghapus komentar dari masing – masing karyawan.

Nama *use case* : Kelola Komentar.

l. Private Message

Aktor : Karyawan

Fungsi untuk mengirimkan pesan secara personal dari masing – masing karyawan.

Nama *use case* : Private Message.

m. Join Group

Aktor : Karyawan

Fungsi untuk bergabung ke dalam group dari masing – masing karyawan.

Nama *use case* : Join Group.

n. Add Friend

Aktor : Karyawan

Fungsi untuk menambahkan teman kepada setiap karyawan.

Nama *use case* : Add Friend.

3.3.3 Analisis Kebutuhan Non Fungsional

Pada tahapan ini dilakukan analisis kebutuhan non fungsional dari sistem, berikut ini adalah kebutuhan *non-fungsional* pada Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik.

a. Kebutuhan Keamanan

Beberapa hal yang akan ditambahkan untuk keamanan Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik adalah:

1) Enkripsi *password*

Enkripsi *password* dalam sistem ini adalah pengacakan data *password* semua *user* agar data tidak mampu lagi dibaca oleh pengguna yang tidak memiliki hak akses. Metode yang digunakan dalam melakukan enkripsi adalah metode hash MD5 dengan penambahan karakter pada *password*.

2) Pembatasan hak akses user

Setiap *user* memiliki hak akses berbeda-beda sesuai tingkatan *user* dalam Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik.

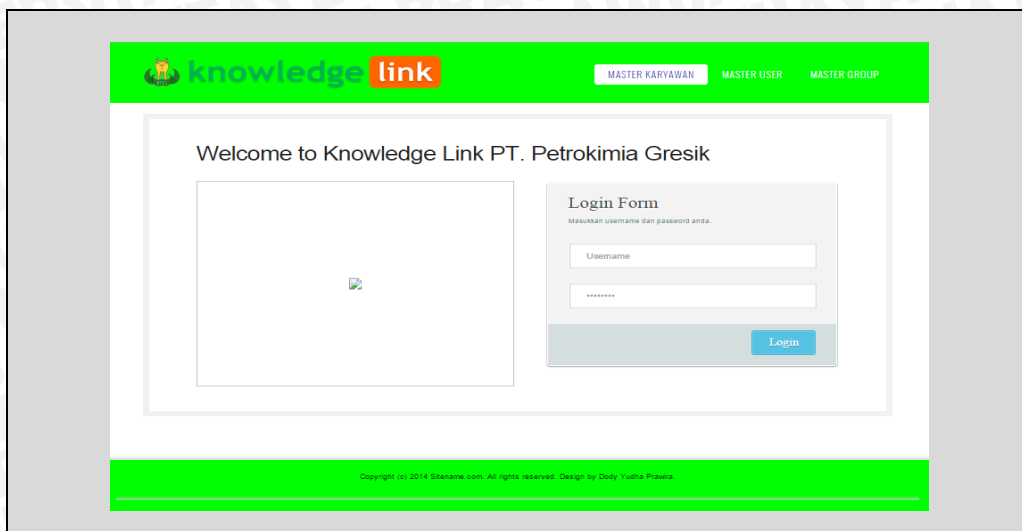
- b. Sistem ini harus *user friendly* yang dapat dipergunakan oleh pengguna dengan mudah.
- c. Sistem ini harus dapat digunakan di berbagai browser (minimal 3 browser : Chrome, Mozilla dan Internet Explorer).

3.3.4 Hasil Dari Proses Penerapan Metode Prototyping

Pada tahapan ini merupakan hasil dari keseluruhan proses penerapan metode *prototyping* dalam pembuatan sistem *expert locator*. Dengan menggunakan metode *prototyping* ini *developer* dan pengguna dapat saling berinteraksi selama proses pembuatan sistem tanpa harus menunggu sampai sistem diimplementasikan. Dalam pembuatan sistem *expert locator* ini, selalu mengalami beberapa kali perubahan. Berikut merupakan hasil dari proses penerapan metode *prototyping* dalam pembuatan sistem *expert locator*.

1) **Mock_up**

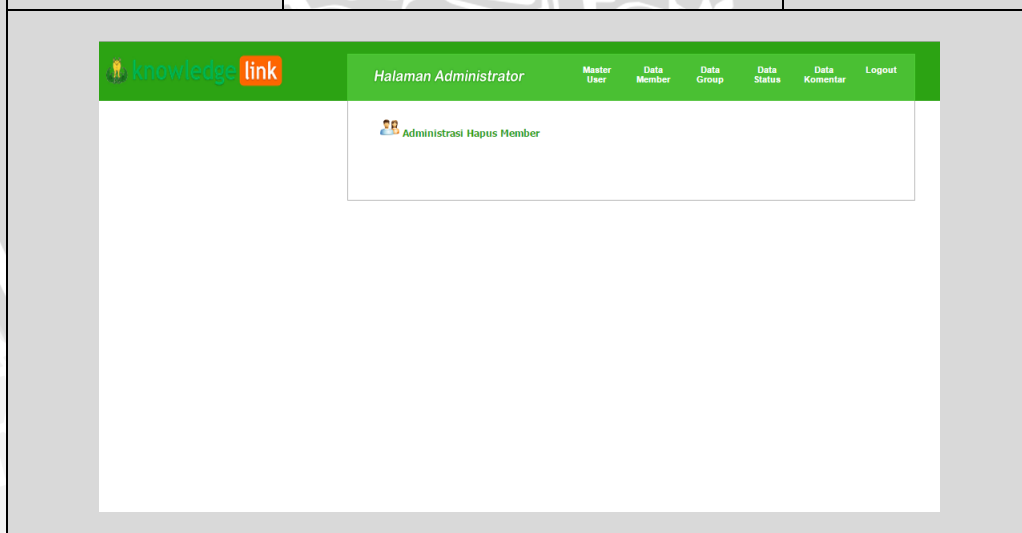
Kode_kebutuhan	Deskripsi	Fitur
FR_01	Halaman utama disini merupakan perancangan awal (Mark_up) untuk aplikasi <i>expert locator</i> . Pada header halaman ini terdapat beberapa menu diantaranya menu master karyawan, master user dan master group. Seperti yang ditunjukkan pada gambar dibawah ini.	Halaman Utama



FR_02

Halaman admin disini merupakan perancangan awal (Mark_up) untuk aplikasi *expert locator*. Pada halaman ini terdapat beberapa menu diantaranya master user, data member, data group, data status, data komentar dan logout. Seperti yang ditunjukkan pada gambar dibawah ini.

Halaman Admin

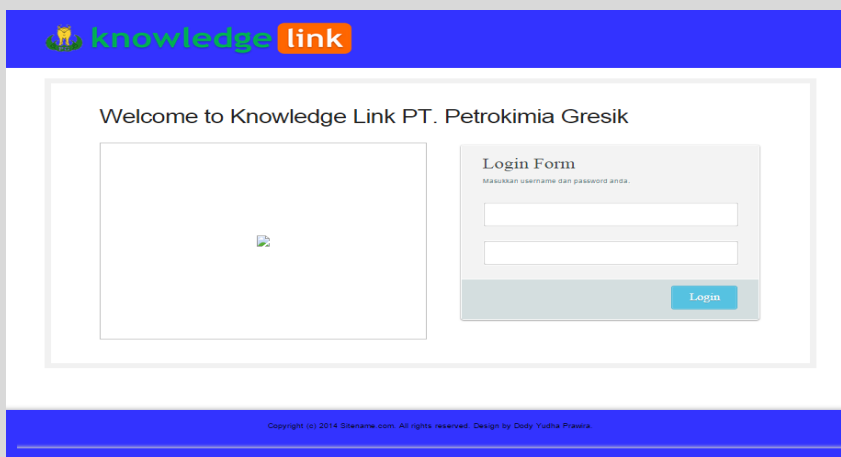


2) Progress 1

Kode_kebutuhan	Deskripsi	Fitur
FR_01	Pada perancangan mark_up aplikasi yang sebelumnya terdapat perubahan	Halaman Utama



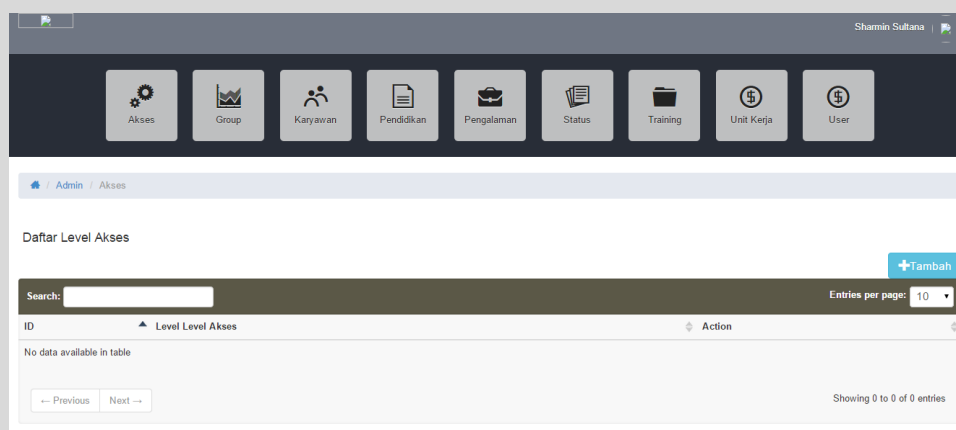
yang diminta oleh user yaitu pada halaman utama. Perbaikan perubahan tersebut dijadikan sebagai progress 1, seperti yang ditunjukkan pada gambar dibawah ini.



FR_02

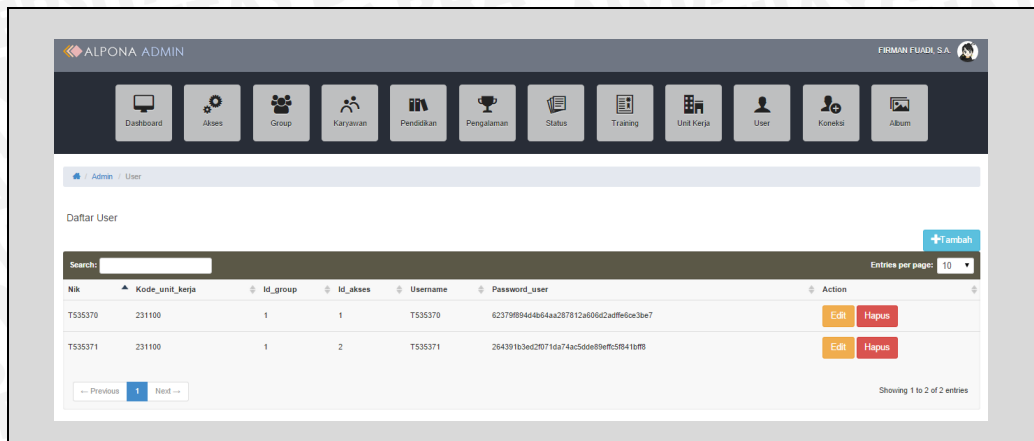
Pada perancangan mark_up aplikasi yang sebelumnya juga terdapat perubahan yang diminta oleh user yaitu pada halaman admin. Perbaikan perubahan tersebut dijadikan sebagai progress 1, seperti yang ditunjukkan pada gambar dibawah ini.

Halaman Admin



3) Progress 2

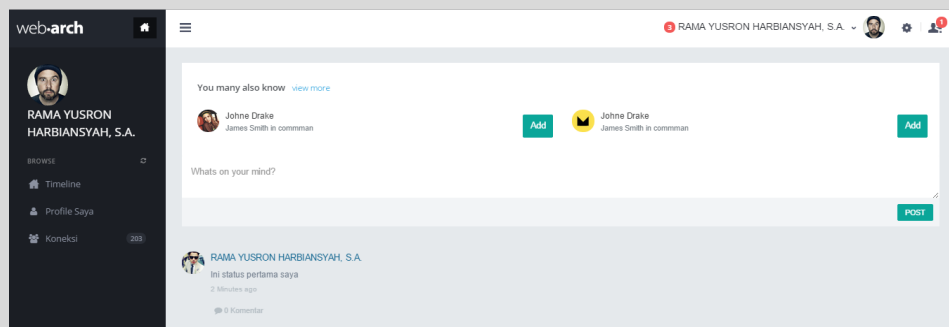
Kode_kebutuhan	Deskripsi	Fitur
FR_01	<p>Pada tampilan halaman utama disini terdapat perubahan yang diminta oleh user yaitu mengenai perubahan warna yang sebelumnya berwarna biru menjadi abu-abu. Perbaikan perubahan tersebut dijadikan sebagai progress 2, seperti yang ditunjukkan pada gambar dibawah ini.</p>	Halaman Utama
		
FR_02	<p>Pada halaman admin disini juga ada perubahan yaitu mengenai penambahan modul yang diminta oleh user yang diantaranya modul dashboard, koneksi dan album. Perbaikan perubahan tersebut dijadikan sebagai progress 2, seperti yang ditunjukkan pada gambar dibawah ini.</p>	Halaman Admin



FR_03

Pada progress 2 disini menampilkan mark_up dari halaman user yaitu halaman timeline. Pada halaman ini nantinya user dapat menuliskan status mereka. Seperti yang ditunjukkan pada gambar dibawah ini.

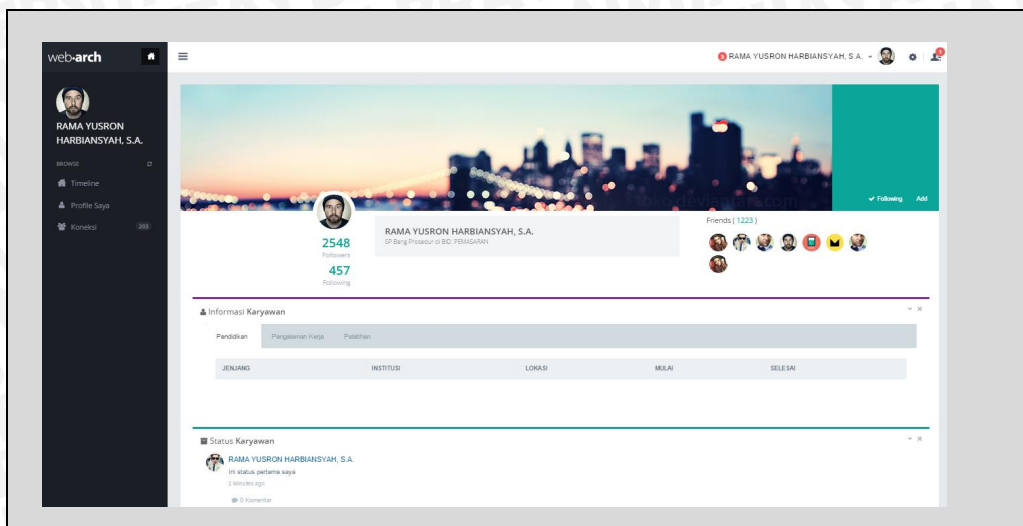
Halaman User (Timeline)



FR_04

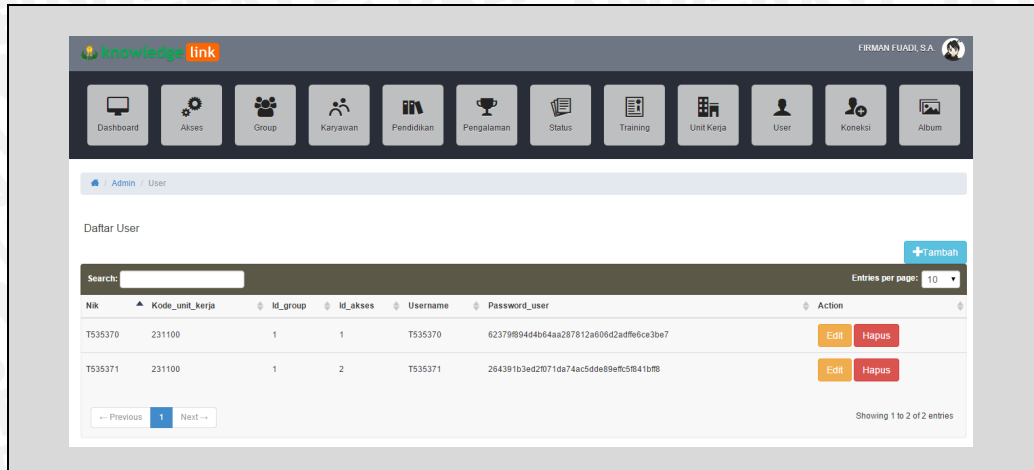
Pada progress 2 disini menampilkan mark_up dari halaman user yaitu halaman profil. Pada halaman ini menampilkan mengenai keseluruhan informasi dari user itu sendiri. Seperti yang ditunjukkan pada gambar dibawah ini.

Halaman User (Profil)



4) Progress 3

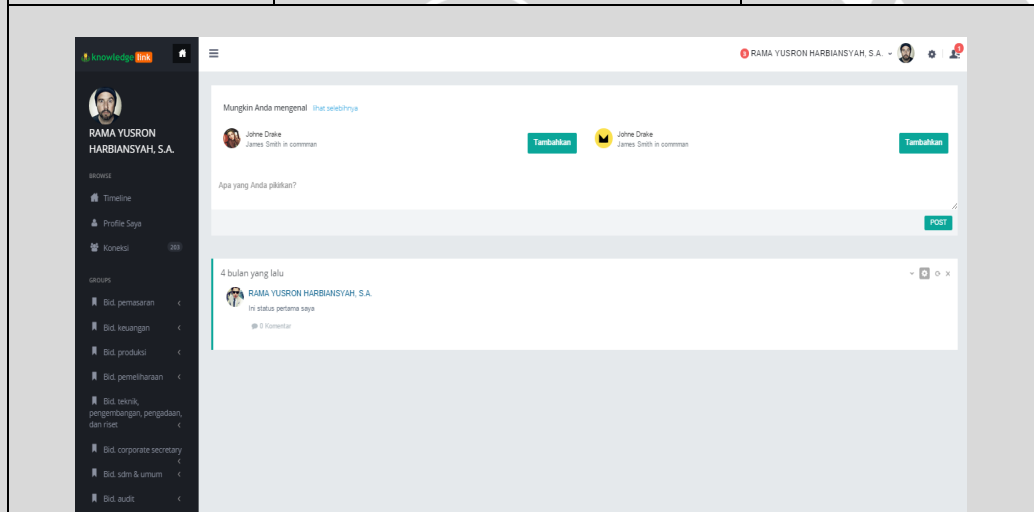
Kode_kebutuhan	Deskripsi	Fitur
FR_01	Pada halaman utama disini tidak ada perubahan, seperti yang ditunjukkan pada gambar dibawah ini.	Halaman Utama
FR_02	Pada halaman admin disini terdapat perubahan yaitu pada bagian header dimana terdapat logo dari aplikasi <i>expert locator</i> , seperti yang ditunjukkan pada gambar dibawah ini.	Halaman Admin



FR_03

Pada halaman timeline disini tidak ada perubahan, seperti yang ditunjukkan pada gambar dibawah ini.

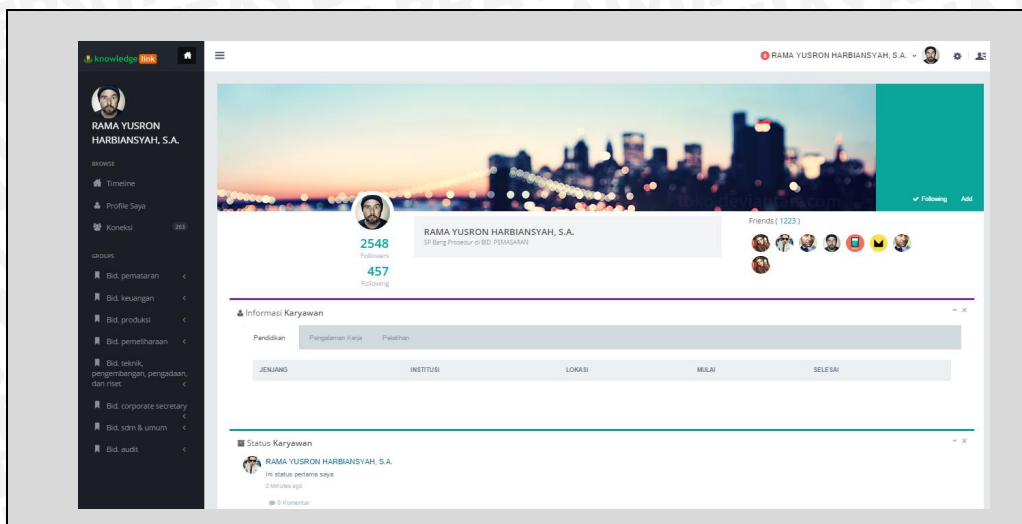
Halaman User (Timeline)



FR_04

Pada halaman profil disini tidak ada perubahan, seperti yang ditunjukkan pada gambar dibawah ini.

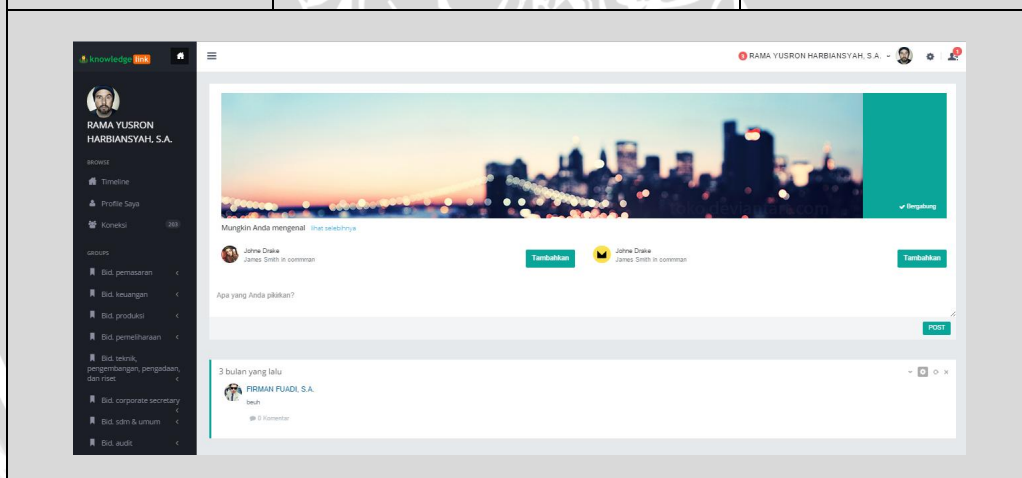
Halaman User (Profil)



FR_05

Pada halaman user disini terdapat perubahan yaitu mengenai penambahan menu group, perubahan tersebut dijadikan sebagai progress 3. Seperti yang ditunjukkan pada gambar dibawah ini.

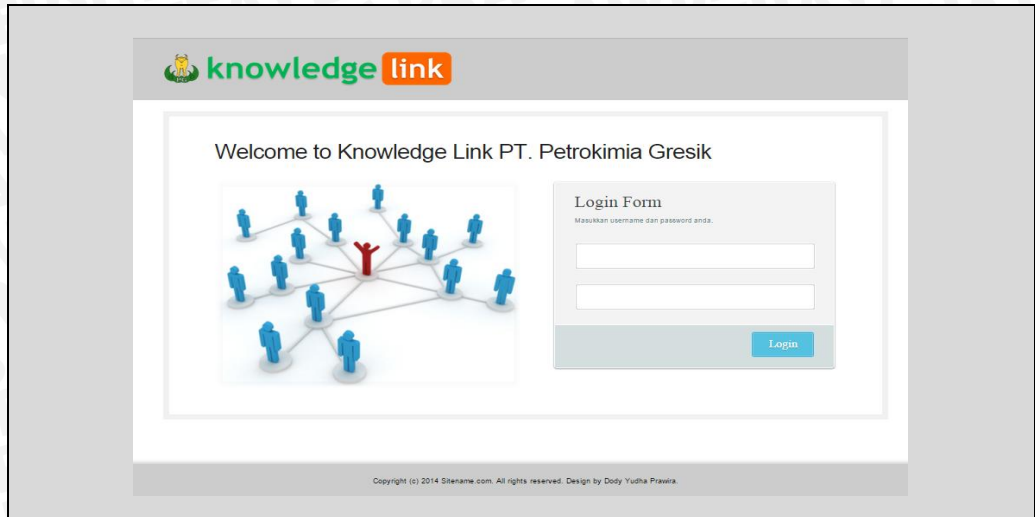
Halaman User (Group)



5) Progress 4 (Final Progress)

Kode_kebutuhan	Deskripsi	Fitur
FR_01	Pada final progress tidak ada perubahan pada halaman utama, seperti yang ditunjukkan pada gambar dibawah ini.	Halaman Utama





FR_02

Pada final progress terdapat perubahan pada halaman admin yaitu pada modul dashboard dan koneksi. Modul tersebut digantikan dengan modul sertifikasi dan modul pesan, seperti yang ditunjukkan pada gambar dibawah ini.

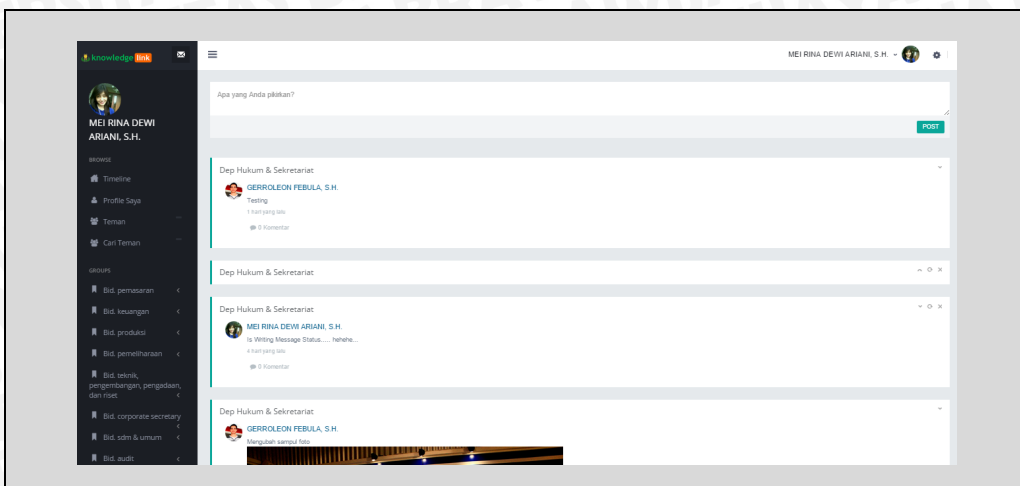
Halaman Admin

Nik	Kode_unit_kerja	Id_group	Id_akses	Username	Password_user	Action
T535370	231100	1	1	T535370	62379894464a287812a0962adff6e03ae7	Edit Hapus
T535371	231100	9	5	T535371	26439163bed2071da74ac5d6e9ef6c5841bdf8	Edit Hapus
T535372	231100	1	3	T535372	46b4c3441a7020278b308d9120e459a7d9980	Edit Hapus
T535373	524400	10	3	T535373	a46443d936485376813964c757b47ea7123b8e2	Edit Hapus
T535374	232100	9	3	T535374	04eb8c6d87083a5c4e4eb377436e9447088922	Edit Hapus
T535398	533010	1	3	T535398	d714b6c5494863a78f93a67207d8b1a926a2bc	Edit Hapus

FR_03

Pada halaman timeline disini tidak ada perubahan, seperti yang ditunjukkan pada gambar dibawah ini.

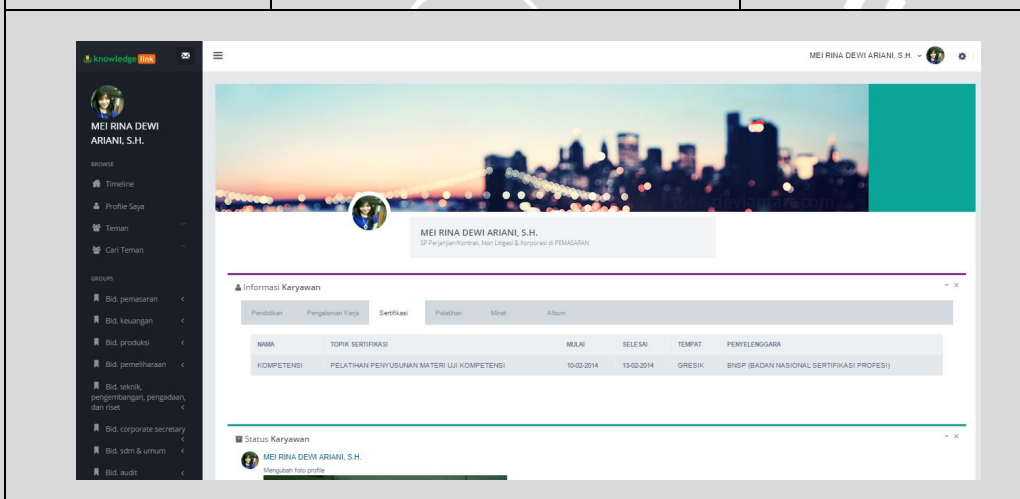
Halaman User
(Timeline)



FR_04

Pada halaman profil disini tidak ada perubahan, seperti yang ditunjukkan pada gambar dibawah ini.

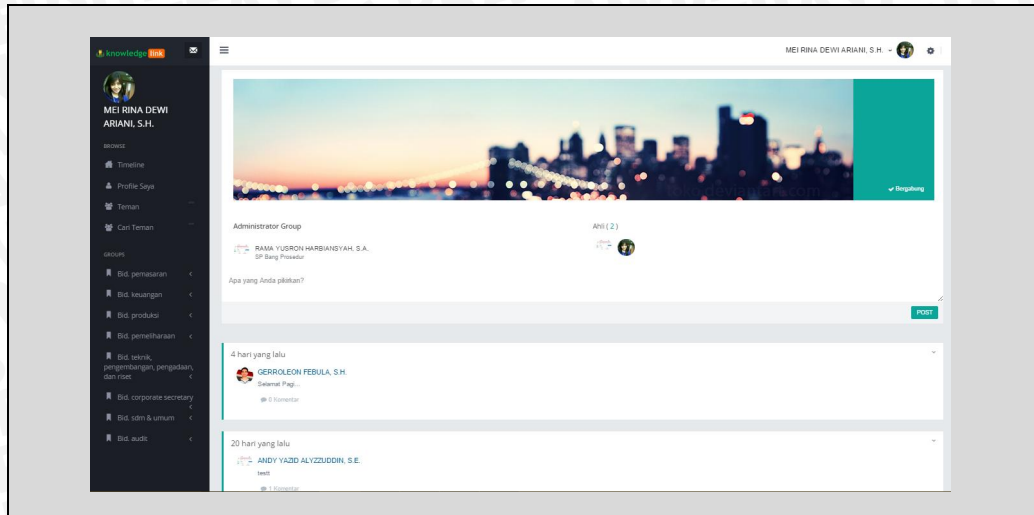
Halaman User (Profil)



FR_05

Pada final progress disini terdapat beberapa perubahan di halaman user yaitu mengenai penambahan administrator group di setiap group serta menampilkan setiap user yang dikategorikan sebagai seorang ahli, seperti yang ditunjukkan pada gambar dibawah ini.

Halaman User (Group)



3.3.5 Diagram Use Case

Pada tahapan ini dilakukan sebuah pemodelan untuk aktivitas dengan disertai sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Hal ini dilakukan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan juga siapa saja yang berhak menggunakan fungsi – fungsi tersebut.

Tabel 3.1 Definisi Actor

No	Actor	Deskripsi
1.	Karyawan	Orang yang memiliki hak akses untuk edit <i>profile</i> , kelola status, kelola komentar, <i>private message</i> , <i>join group</i> , dan <i>add friend</i> .
2.	Admin	Orang yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan beberapa data diantaranya kelola data karyawan, kelola data pendidikan, kelola data training, kelola data pengalaman kerja, kelola data sertifikasi, kelola data unit kerja, kelola data user, kelola data akses, kelola data group, kelola data status karyawan, kelola data pesan, dan kelola data album.

Sedangkan pendefinisian *use case* pada Rancang Bangun Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik di tunjukkan pada tabel 3.2.

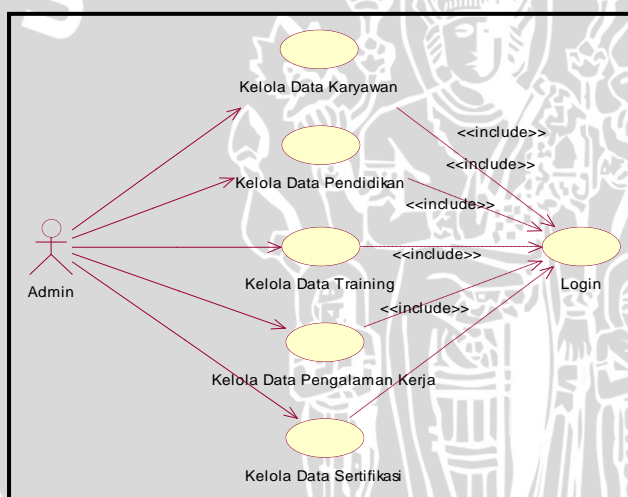
Tabel 3.2 Definisi *Use Case*

No	<i>Use case</i>	Deskripsi
1.	<i>Login</i>	Merupakan proses untuk melakukan <i>login</i> user/aktor sistem <i>expert locator</i> .
2.	Edit Profil	Merupakan proses dimana user dapat mengedit data profil.
3.	Keloa Status	Merupakan proses dimana user bisa menulis, mengupdate dan juga menghapus status mereka.
4.	Kelola Komentar	Merupakan proses dimana user bisa menulis, mengupdate dan juga menghapus komentar mereka.
5.	<i>Private Message</i>	Merupakan proses dimana user bisa mengirimkan pesan secara personal kepada user lain.
6.	<i>Join Group</i>	Merupakan proses dimana user dapat bergabung dengan beberapa group yang ada.
7.	<i>Add Friend</i>	Merupakan proses dimana user dapat menambahkan teman.
8.	Kelola Data Karyawan	Merupakan proses untuk memasukkan, mengubah, dan menghapus data karyawan.
9.	Kelola Data Pendidikan	Merupakan proses untuk memasukkan, mengubah, dan menghapus data pendidikan.
10.	Kelola Data Training	Merupakan proses untuk memasukkan, mengubah, dan menghapus data training.
11.	Kelola Data Pengalaman Kerja	Merupakan proses untuk memasukkan, mengubah, dan menghapus data pengalaman kerja.
12.	Kelola Data Sertifikasi	Merupakan proses untuk memasukkan, mengubah, dan menghapus data sertifikasi.
13.	Kelola Data Unit Kerja	Merupakan proses untuk memasukkan, mengubah, dan menghapus unit kerja.
14.	Kelola Data User	Merupakan proses untuk memasukkan, mengubah, dan menghapus data user.
15.	Kelola Data Group	Merupakan proses untuk memasukan, mengubah, dan menghapus data group
16.	Kelola Data Status	Merupakan proses untuk memasukan, mengubah,

	Karyawan	dan menghapus data status karyawan.
17.	Kelola Data Pesan	Merupakan proses untuk memasukkan, mengubah, dan menghapus data pesan.
18.	Kelola Data Akses	Merupakan proses untuk memasukkan, mengubah, dan menghapus data akses.
19.	Kelola Data Album	Merupakan proses untuk memasukkan, mengubah, dan menghapus data album.

Berikut merupakan *use case* diagram dari Rancang Bangun Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik yang sudah diklasifikasikan berdasarkan modul-modul yang ada untuk memudahkan dalam perancangannya, diantaranya :

1. Bagian Modul Admin : Pengolahan Data Pribadi Karyawan



Gambar 3.4. Use Case Diagram Pengolahan Data Pribadi Karyawan

3.3.4.1 Skenario Use Case

Berikut merupakan skenario jalannya daripada masing-masing use case yang sudah di definisikan sebelumnya:

Tabel 3.3. Skenario Use Case Login

Nama Use Case	Login
:	
Aktor :	Admin dan Karyawan

Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan kegiatan autentikasi pengguna sebelum masuk ke dalam sistem.	
Main Flow :	Aksi Aktor : 1) User memasukkan data berupa username dan password	Reaksi Sistem : 2) Sistem mengecek valid atau tidaknya data masukan dengan memeriksa ke dalam tabel m_user
Alternate Flow :	Aksi Aktor : 1) User memasukkan data username dan password yang salah.	Reaksi Sistem : 2) Sistem memeriksa valid atau tidaknya data masukan 3) Sistem akan menampilkan pesan <i>error</i> , aktor diminta untuk memasukkan kembali username dan password.
Kondisi Awal :	Aktor belum melakukan proses login, sehingga belum bisa masuk ke dalam sistem.	
Kondisi Akhir :	Jika use case sukses, aktor dapat masuk ke dalam sistem untuk selanjutnya dapat melakukan kegiatannya.	

Tabel 3.4. Skenario *Use Case* Kelola Data Karyawan

Nama Use Case :	Kelola Data Karyawan	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data karyawan.	
Main Flow :	Aksi Aktor : Memasukkan Data Karyawan 1) User masuk ke modul	Reaksi Sistem :

	<p>karyawan.</p> <ol style="list-style-type: none"> 4) User memilih menu tambah karyawan. 6) User memasukkan data karyawan. 7) User menekan tombol “simpan”. <p>Mengubah Data Karyawan</p> <ol style="list-style-type: none"> 1) User masuk ke modul karyawan. 4) User memilih menu edit karyawan. 6) User merubah data karyawan. 7) User menekan tombol “simpan”. <p>Menghapus Data Karyawan</p> <ol style="list-style-type: none"> 1) User memilih modul karyawan. 4) User memilih data karyawan yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 2) Sistem menampilkan modul karyawan 3) Sistem menampilkan list daftar karyawan. 5) Sistem menampilkan form tambah karyawan. 8) Sistem menyimpan data karyawan ke dalam database. 9) Sistem menampilkan keseluruhan daftar karyawan. 2) Sistem menampilkan modul karyawan. 3) Sistem menampilkan list daftar karyawan. 5) Sistem menampilkan form ubah data karyawan. 8) Sistem menyimpan data karyawan ke dalam database. 9) Sistem menampilkan keseluruhan daftar karyawan. 2) Sistem menampilkan modul karyawan. 3) Sistem menampilkan list daftar karyawan.
--	---	--

		6) Sistem menghapus data karyawan dalam list daftar karyawan dan juga database.
Alternate Flow :	Aksi Aktor : Cari Data Karyawan 1) User masuk ke modul karyawan. 4) User mengetikkan keyword nama karyawan di form pencarian.	Reaksi Sistem : 2) Sistem menampilkan modul karyawan. 3) Sistem menampilkan list daftar karyawan. 5) Sistem melakukan pencarian data karyawan sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul karyawan sebelum melakukan keseluruhan pengolahan data karyawan.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data karyawan, ubah data karyawan dan hapus data karyawan. Setelah itu data karyawan akan tersimpan ke dalam database sistem.	

Tabel 3.5. Skenario *Use Case* Kelola Data Pendidikan

Nama Use Case :	Kelola Data Pendidikan	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data pendidikan.	
Main Flow :	Aksi Aktor : Memasukkan Data Pendidikan	Reaksi Sistem :

	<ol style="list-style-type: none"> 1) User masuk ke modul pendidikan. 4) User memilih menu tambah pendidikan. 6) User memasukkan data pendidikan. 7) User menekan tombol “simpan”. <p>Mengubah Data Pendidikan</p> <ol style="list-style-type: none"> 1) User masuk ke modul pendidikan. 4) User memilih menu edit pendidikan. 6) User merubah data pendidikan. 7) User menekan tombol “simpan”. <p>Menghapus Data Pendidikan</p> <ol style="list-style-type: none"> 1) User memilih modul pendidikan. 4) User memilih data pendidikan yang akan 	<ol style="list-style-type: none"> 2) Sistem menampilkan modul pendidikan. 3) Sistem menampilkan list daftar pendidikan. 5) Sistem menampilkan form tambah pendidikan. 8) Sistem menyimpan data pendidikan ke dalam database. 9) Sistem menampilkan keseluruhan daftar pendidikan. 2) Sistem menampilkan modul pendidikan. 3) Sistem menampilkan list daftar pendidikan. 5) Sistem menampilkan form ubah data pendidikan. 8) Sistem menyimpan data pendidikan ke dalam database. 9) Sistem menampilkan keseluruhan daftar pendidikan. 2) Sistem menampilkan modul pendidikan. 3) Sistem menampilkan list daftar pendidikan.
--	--	---

	<p>dihapus</p> <p>5) User menekan tombol “hapus”</p>	<p>6) Sistem menghapus data pendidikan dalam list daftar pendidikan dan juga database.</p>
Alternate Flow :	<p>Aksi Aktor :</p> <p>Cari Data Pendidikan</p> <p>1) User masuk ke modul pendidikan.</p> <p>4) User mengetikkan keyword nama pendidikan di form pencarian.</p>	<p>Reaksi Sistem :</p> <p>2) Sistem menampilkan modul pendidikan.</p> <p>3) Sistem menampilkan list daftar pendidikan.</p> <p>5) Sistem melakukan pencarian data pendidikan sesuai dari keyword yang di masukkan.</p> <p>6) Sistem menampilkan hasil pencarian.</p>
Kondisi Awal :	<p>Aktor harus masuk terlebih dahulu ke dalam modul pendidikan sebelum melakukan keseluruhan pengolahan data pendidikan.</p>	
Kondisi Akhir :	<p>Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data pendidikan, ubah data pendidikan dan hapus data pendidikan. Setelah itu data pendidikan akan tersimpan ke dalam database sistem.</p>	

Tabel 3.6. Skenario *Use Case* Kelola Data Training

Nama Use Case :	Kelola Data Training
Aktor :	Admin
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data training.

<p>Main Flow :</p>	<p>Aksi Aktor :</p> <p>Memasukkan Data Training</p> <ol style="list-style-type: none"> 1) User masuk ke modul training. 4) User memilih menu tambah training. 6) User memasukkan data training. 7) User menekan tombol “simpan”. <p>Mengubah Data Training</p> <ol style="list-style-type: none"> 1) User masuk ke modul training. 4) User memilih menu edit training. 6) User merubah data training. 7) User menekan tombol “simpan”. <p>Menghapus Data Training</p> <ol style="list-style-type: none"> 1) User memilih modul training. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul training. 3) Sistem menampilkan list daftar training. 5) Sistem menampilkan form tambah training. 8) Sistem menyimpan data training ke dalam database. 9) Sistem menampilkan keseluruhan daftar training. 2) Sistem menampilkan modul training. 3) Sistem menampilkan list daftar training. 5) Sistem menampilkan form ubah data training. 8) Sistem menyimpan data training ke dalam database. 9) Sistem menampilkan keseluruhan daftar training. 2) Sistem menampilkan
---------------------------	---	--

	<p>4) User memilih data training yang akan dihapus</p> <p>5) User menekan tombol “hapus”</p>	<p>modul training.</p> <p>3) Sistem menampilkan list daftar training.</p> <p>6) Sistem menghapus data training dalam list daftar training dan juga database.</p>
Alternate Flow :	<p>Aksi Aktor :</p> <p>Cari Data Training</p> <p>1) User masuk ke modul training.</p> <p>4) User mengetikkan keyword nama training di form pencarian.</p>	<p>Reaksi Sistem :</p> <p>2) Sistem menampilkan modul training.</p> <p>3) Sistem menampilkan list daftar training.</p> <p>5) Sistem melakukan pencarian data training sesuai dari keyword yang di masukkan.</p> <p>6) Sistem menampilkan hasil pencarian.</p>
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul training sebelum melakukan keseluruhan pengolahan data training.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data training, ubah data training dan hapus data training. Setelah itu data training akan tersimpan ke dalam database sistem.	

Tabel 3.7. Skenario *Use Case* Kelola Data Pengalaman Kerja

Nama Use Case :	Kelola Data Pengalaman Kerja
Aktor :	Admin
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan

	keseluruhan proses pengelolaan data pengalaman kerja.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data Pengalaman Kerja</p> <ol style="list-style-type: none"> 1) User masuk ke modul pengalaman kerja. 4) User memilih menu tambah pengalaman kerja. 6) User memasukkan data pengalaman kerja. 7) User menekan tombol “simpan”. <p>Mengubah Data Pengalaman Kerja</p> <ol style="list-style-type: none"> 1) User masuk ke modul pengalaman kerja. 4) User memilih menu edit pengalaman kerja. 6) User merubah data pengalaman kerja. 7) User menekan tombol “simpan”. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul pengalaman kerja. 3) Sistem menampilkan list daftar pengalaman kerja. 5) Sistem menampilkan form tambah pengalaman kerja. 8) Sistem menyimpan data pengalaman kerja ke dalam database. 9) Sistem menampilkan keseluruhan daftar pengalaman kerja. <ol style="list-style-type: none"> 2) Sistem menampilkan modul pengalaman kerja. 3) Sistem menampilkan list daftar pengalaman kerja. 5) Sistem menampilkan form ubah data pengalaman kerja. 8) Sistem menyimpan data pengalaman kerja ke dalam database. 9) Sistem menampilkan keseluruhan daftar

	<p>Menghapus Data Pengalaman Kerja</p> <ol style="list-style-type: none"> 1) User memilih modul pengalaman kerja. 4) User memilih data pengalaman kerja yang akan dihapus 5) User menekan tombol “hapus” 	<p>pengalaman kerja.</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul pengalaman kerja. 3) Sistem menampilkan list daftar pengalaman kerja. 6) Sistem menghapus data pengalaman kerja dalam list daftar pengalaman kerja dan juga database.
Alternate Flow :	<p>Aksi Aktor :</p> <p>Cari Data Pengalaman Kerja</p> <ol style="list-style-type: none"> 1) User masuk ke modul pengalaman kerja. 4) User mengetikkan keyword nama pengalaman kerja di form pencarian. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul pengalaman kerja. 3) Sistem menampilkan list daftar pengalaman kerja. 5) Sistem melakukan pencarian data pengalaman kerja sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul pengalaman kerja sebelum melakukan keseluruhan pengolahan data pengalaman kerja.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data pengalaman	

	kerja, ubah data pengalaman kerja dan hapus data pengalaman kerja. Setelah itu data pengalaman kerja akan tersimpan ke dalam database sistem.
--	---

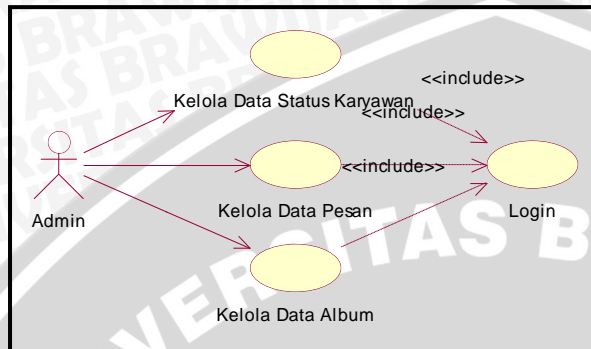
Tabel 3.8. Skenario *Use Case* Kelola Data Sertifikasi

Nama Use Case :	Kelola Data Sertifikasi	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data sertifikasi.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data Sertifikasi</p> <ol style="list-style-type: none"> 1) User masuk ke modul sertifikasi. 4) User memilih menu tambah sertifikasi. 6) User memasukkan data sertifikasi. 7) User menekan tombol "simpan". <p>Mengubah Data Sertifikasi</p> <ol style="list-style-type: none"> 1) User masuk ke modul sertifikasi. 4) User memilih menu edit sertifikasi. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul sertifikasi. 3) Sistem menampilkan list daftar sertifikasi. 5) Sistem menampilkan form tambah sertifikasi. 8) Sistem menyimpan data sertifikasi ke dalam database. 9) Sistem menampilkan keseluruhan daftar sertifikasi. 2) Sistem menampilkan modul sertifikasi 3) Sistem menampilkan list daftar sertifikasi. 5) Sistem menampilkan

	<p>6) User merubah data sertifikasi. 7) User menekan tombol “simpan”.</p> <p>Menghapus Data Sertifikasi</p> <p>1) User memilih modul sertifikasi. 4) User memilih data sertifikasi yang akan dihapus 5) User menekan tombol “hapus”</p>	<p>form ubah data sertifikasi.</p> <p>8) Sistem menyimpan data sertifikasi ke dalam database. 9) Sistem menampilkan keseluruhan daftar sertifikasi.</p> <p>2) Sistem menampilkan modul sertifikasi. 3) Sistem menampilkan list daftar sertifikasi.</p> <p>6) Sistem menghapus data sertifikasi dalam list daftar sertifikasi dan juga database.</p>
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Cari Data Sertifikasi</p> <p>1) User masuk ke modul sertifikasi. 4) User mengetikkan keyword nama sertifikasi di form pencarian.</p>	<p>Reaksi Sistem :</p> <p>2) Sistem menampilkan modul sertifikasi. 3) Sistem menampilkan list daftar sertifikasi. 5) Sistem melakukan pencarian data sertifikasi sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.</p>
<p>Kondisi Awal :</p>	<p>Aktor harus masuk terlebih dahulu ke dalam modul sertifikasi sebelum melakukan keseluruhan pengolahan data sertifikasi.</p>	

Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data sertifikasi, ubah data sertifikasi dan hapus data sertifikasi. Setelah itu data sertifikasi akan tersimpan ke dalam database sistem.
------------------------	---

2. Bagian Modul Admin : Pengolahan Data Jejaring Sosial



Gambar 3.5. Use Case Diagram Pengolahan Data Jejaring Sosial

Tabel 3.9. Skenario Use Case Kelola Data Status Karyawan

Nama Use Case :	Kelola Data Status Karyawan	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data status karyawan.	
Main Flow :	Aksi Aktor : Memasukkan Data Status Karyawan 1) User masuk ke modul status. 4) User memilih menu tambah status. 6) User memasukkan data status. 7) User menekan tombol "simpan".	Reaksi Sistem : 2) Sistem menampilkan modul status. 3) Sistem menampilkan list daftar status. 5) Sistem menampilkan form tambah status. 8) Sistem menyimpan data status ke dalam database.

	<p>Mengubah Data Status Karyawan</p> <ol style="list-style-type: none"> 1) User masuk ke modul status. 4) User memilih menu edit status. 6) User merubah data status. 7) User menekan tombol “simpan”. <p>Menghapus Data Status Karyawan</p> <ol style="list-style-type: none"> 1) User memilih modul status. 4) User memilih data status yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 9) Sistem menampilkan keseluruhan daftar status. 2) Sistem menampilkan modul status. 3) Sistem menampilkan list daftar status. 5) Sistem menampilkan form ubah data status. 8) Sistem menyimpan data status ke dalam database. 9) Sistem menampilkan keseluruhan daftar status. 2) Sistem menampilkan modul status. 3) Sistem menampilkan list daftar status. 6) Sistem menghapus data status dalam list daftar status dan juga database.
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Cari Data Status Karyawan</p> <ol style="list-style-type: none"> 1) User masuk ke modul 	<p>Reaksi Sistem :</p>



	status. 4) User mengetikkan keyword nama status di form pencarian.	2) Sistem menampilkan modul status. 3) Sistem menampilkan list daftar status. 5) Sistem melakukan pencarian data status sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul status sebelum melakukan keseluruhan pengolahan data status karyawan.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data status, ubah data status dan hapus data status. Setelah itu data status akan tersimpan ke dalam database sistem.	

Tabel 3.10. Skenario *Use Case* Kelola Data Pesan

Nama Use Case :	Kelola Data Pesan	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data pesan.	
Main Flow :	Aksi Aktor : Memasukkan Data Pesan 1) User masuk ke modul pesan. 4) User memilih menu tambah pesan. 6) User memasukkan data pesan. 7) User menekan tombol "simpan".	Reaksi Sistem : 2) Sistem menampilkan modul pesan. 3) Sistem menampilkan list daftar pesan. 5) Sistem menampilkan form tambah pesan. 8) Sistem menyimpan data pesan ke dalam

	<p>Mengubah Data Pesan</p> <ol style="list-style-type: none"> 1) User masuk ke modul pesan. 4) User memilih menu edit pesan. 6) User merubah data pesan. 7) User menekan tombol “simpan”. <p>Menghapus Data Pesan</p> <ol style="list-style-type: none"> 1) User memilih modul pesan. 4) User memilih data pesan yang akan dihapus 5) User menekan tombol “hapus” 	<p>database.</p> <ol style="list-style-type: none"> 9) Sistem menampilkan keseluruhan daftar pesan. 2) Sistem menampilkan modul pesan. 3) Sistem menampilkan list daftar pesan. 5) Sistem menampilkan form ubah data pesan. 8) Sistem menyimpan data pesan ke dalam database. 9) Sistem menampilkan keseluruhan daftar pesan. 2) Sistem menampilkan modul pesan. 3) Sistem menampilkan list daftar pesan. 6) Sistem menghapus data pesan dalam list daftar pesan dan juga database.
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Cari Data Pesan</p> <ol style="list-style-type: none"> 1) User masuk ke modul pesan. 4) User mengetikkan keyword nama pesan di form pencarian. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul pesan. 3) Sistem menampilkan list daftar pesan. 5) Sistem melakukan

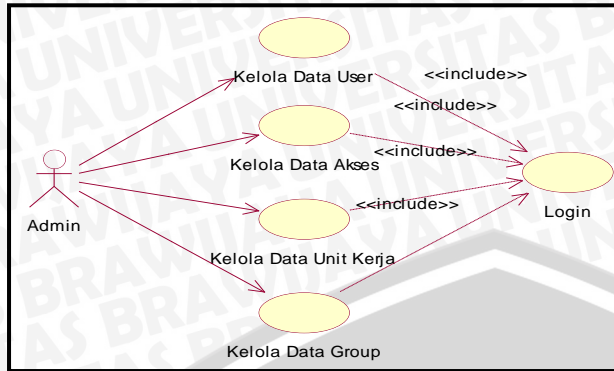
		<p>pencairan data pesan sesuai dari keyword yang di masukkan.</p> <p>6) Sistem menampilkan hasil pencairan.</p>
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul pesan sebelum melakukan keseluruhan pengolahan data pesan.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data pesan, ubah data pesan dan hapus data pesan. Setelah itu data pesan akan tersimpan ke dalam database sistem.	

Tabel 3.11. Skenario *Use Case* Kelola Data Album

Nama Use Case :	Kelola Data Album	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data album.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data Album</p> <ol style="list-style-type: none"> 1) User masuk ke modul album. 4) User memilih menu tambah album. 6) User memasukkan data album. 7) User menekan tombol “simpan”. <p>Mengubah Data Album</p> <ol style="list-style-type: none"> 1) User masuk ke modul album. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul album. 3) Sistem menampilkan list daftar album. 5) Sistem menampilkan form tambah album. 8) Sistem menyimpan data album ke dalam database. 9) Sistem menampilkan keseluruhan daftar album. 2) Sistem menampilkan modul album.

	<p>4) User memilih menu edit album.</p> <p>6) User merubah data album.</p> <p>7) User menekan tombol “simpan”.</p> <p>Menghapus Data Album</p> <p>1) User memilih modul album.</p> <p>4) User memilih data album yang akan dihapus</p> <p>5) User menekan tombol “hapus”</p>	<p>3) Sistem menampilkan list daftar album.</p> <p>5) Sistem menampilkan form ubah data album.</p> <p>8) Sistem menyimpan data album ke dalam database.</p> <p>9) Sistem menampilkan keseluruhan daftar album.</p> <p>2) Sistem menampilkan modul album.</p> <p>3) Sistem menampilkan list daftar album.</p> <p>6) Sistem menghapus data album dalam list daftar album dan juga database.</p>
Alternate Flow :	<p>Aksi Aktor :</p> <p>Lihat Data Album</p> <p>1) User masuk ke modul album.</p>	<p>Reaksi Sistem :</p> <p>2) Sistem menampilkan modul album.</p> <p>3) Sistem menampilkan list daftar album.</p>
Kondisi Awal :	<p>Aktor harus masuk terlebih dahulu ke dalam modul album sebelum melakukan keseluruhan pengolahan data album.</p>	
Kondisi Akhir :	<p>Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data album, ubah data album dan hapus data album. Setelah itu data album akan tersimpan ke dalam database sistem.</p>	

3. Bagian Modul Admin : Pengolahan Data Akun



Gambar 3.6. Use Case Diagram Pengolahan Data Akun

Tabel 3.12. Skenario Use Case Kelola Data User

Nama Use Case :	Kelola Data User	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data user.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data User</p> <ol style="list-style-type: none"> 1) User masuk ke modul user. 4) User memilih menu tambah user. 6) User memasukkan data user. 7) User menekan tombol “simpan”. <p>Mengubah Data User</p> <ol style="list-style-type: none"> 1) User masuk ke modul 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul user. 3) Sistem menampilkan list daftar user. 5) Sistem menampilkan form tambah user. 8) Sistem menyimpan data user ke dalam database. 9) Sistem menampilkan keseluruhan daftar user.

	<p>user.</p> <ol style="list-style-type: none"> 4) User memilih menu edit user. 6) User merubah data user. 7) User menekan tombol “simpan”. <p>Menghapus Data User</p> <ol style="list-style-type: none"> 1) User memilih modul user. 4) User memilih data user yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 2) Sistem menampilkan modul user. 3) Sistem menampilkan list daftar user. 5) Sistem menampilkan form ubah data user. 8) Sistem menyimpan data user ke dalam database. 9) Sistem menampilkan keseluruhan daftar user. 2) Sistem menampilkan modul user. 3) Sistem menampilkan list daftar user. 6) Sistem menghapus data user dalam list daftar user dan juga database.
Alternate Flow :	<p>Aksi Aktor :</p> <p>Cari Data User</p> <ol style="list-style-type: none"> 1) User masuk ke modul user 4) User mengetikkan keyword nama user di form pencarian. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul user. 3) Sistem menampilkan list daftar user. 5) Sistem melakukan pencarian data user sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
Kondisi Awal :	<p>Aktor harus masuk terlebih dahulu ke dalam modul user sebelum melakukan keseluruhan pengolahan data user.</p>	

Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data user, ubah data user dan hapus data user. Setelah itu data user akan tersimpan ke dalam database sistem.
------------------------	---

Tabel 3.13. Skenario *Use Case* Kelola Data Akses

Nama Use Case :	Kelola Data Akses	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data akses.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data Akses</p> <ol style="list-style-type: none"> 1) User masuk ke modul akses. 4) User memilih menu tambah akses. 6) User memasukkan data akses. 7) User menekan tombol “simpan”. <p>Mengubah Data Akses</p> <ol style="list-style-type: none"> 1) User masuk ke modul akses. 4) User memilih menu edit akses. 6) User merubah data akses. 7) User menekan tombol 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul akses. 3) Sistem menampilkan list daftar akses. 5) Sistem menampilkan form tambah akses. 8) Sistem menyimpan data akses ke dalam database. 9) Sistem menampilkan keseluruhan daftar akses. 2) Sistem menampilkan modul akses. 3) Sistem menampilkan list daftar akses. 5) Sistem menampilkan form ubah data akses.

	<p>“simpan”.</p> <p>Menghapus Data Akses</p> <ol style="list-style-type: none"> 1) User memilih modul akses. 4) User memilih data akses yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 8) Sistem menyimpan data akses ke dalam database. 9) Sistem menampilkan keseluruhan daftar akses. 2) Sistem menampilkan modul akses. 3) Sistem menampilkan list daftar akses. 6) Sistem menghapus data akses dalam list daftar akses dan juga database.
Alternate Flow :	<p>Aksi Aktor :</p> <p>Cari Data Akses</p> <ol style="list-style-type: none"> 1) User masuk ke modul akses. 4) User mengetikkan keyword nama akses di form pencarian. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul akses. 3) Sistem menampilkan list daftar akses. 5) Sistem melakukan pencarian data akses sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul akses sebelum melakukan keseluruhan pengolahan data akses.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data akses, ubah data akses dan hapus data akses. Setelah itu data akses akan tersimpan ke dalam database sistem.	

Tabel 3.14. Skenario *Use Case* Kelola Data Unit Kerja

Nama Use Case :	Kelola Data Unit Kerja	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data unit kerja.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data Unit Kerja</p> <ol style="list-style-type: none"> 1) User masuk ke modul unit kerja. 4) User memilih menu tambah unit kerja. 6) User memasukkan data unit kerja. 7) User menekan tombol “simpan”. <p>Mengubah Data Unit Kerja</p> <ol style="list-style-type: none"> 1) User masuk ke modul unit kerja 4) User memilih menu edit unit kerja. 6) User merubah data unit kerja. 7) User menekan tombol “simpan”. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul unit kerja. 3) Sistem menampilkan list daftar unit kerja. 5) Sistem menampilkan form tambah unit kerja. 8) Sistem menyimpan data unit kerja ke dalam database. 9) Sistem menampilkan keseluruhan daftar unit kerja. 2) Sistem menampilkan modul unit kerja. 3) Sistem menampilkan list daftar unit kerja. 5) Sistem menampilkan form ubah data unit kerja. 8) Sistem menyimpan data unit kerja ke

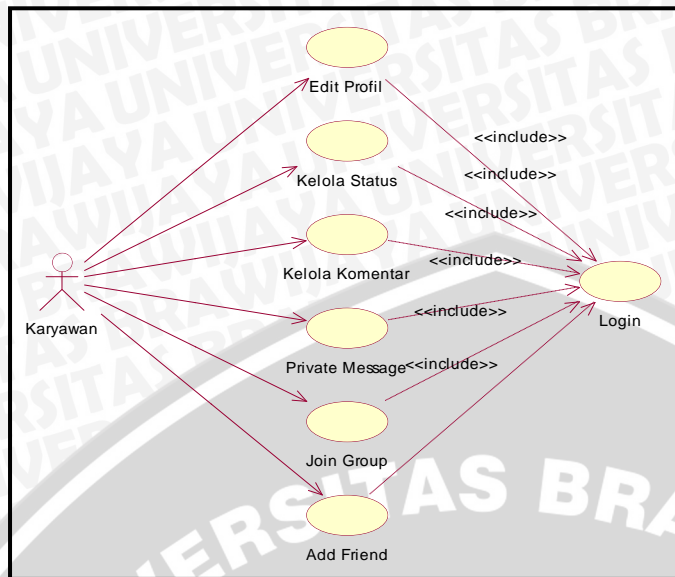
	<p>Menghapus Data Unit Kerja</p> <ol style="list-style-type: none"> 1) User memilih modul unit kerja 4) User memilih data unit kerja yang akan dihapus 5) User menekan tombol “hapus” 	<p>dalam database.</p> <ol style="list-style-type: none"> 9) Sistem menampilkan keseluruhan daftar unit kerja. 2) Sistem menampilkan modul unit kerja. 3) Sistem menampilkan list daftar unit kerja. 6) Sistem menghapus data unit kerja dalam list daftar unit kerja dan juga database.
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Cari Data Unit Kerja</p> <ol style="list-style-type: none"> 1) User masuk ke modul unit kerja. 4) User mengetikkan keyword nama unit kerja di form pencarian. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul unit kerja. 3) Sistem menampilkan list daftar unit kerja. 5) Sistem melakukan pencarian data unit kerja sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
<p>Kondisi Awal :</p>	<p>Aktor harus masuk terlebih dahulu ke dalam modul unit kerja sebelum melakukan keseluruhan pengolahan data unit kerja.</p>	
<p>Kondisi Akhir :</p>	<p>Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data unit kerja, ubah data unit kerja dan hapus data unit kerja. Setelah itu data unit kerja akan tersimpan ke dalam database sistem.</p>	

Tabel 3.15. Skenario *Use Case* Kelola Data Group

Nama Use Case :	Kelola Data Group	
Aktor :	Admin	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data group.	
Main Flow :	<p>Aksi Aktor :</p> <p>Memasukkan Data Group</p> <ol style="list-style-type: none"> 1) User masuk ke modul group 4) User memilih menu tambah group. 6) User memasukkan data group. 7) User menekan tombol “simpan”. <p>Mengubah Data Group</p> <ol style="list-style-type: none"> 1) User masuk ke modul group. 4) User memilih menu edit group. 6) User merubah data group. 7) User menekan tombol “simpan”. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul group. 3) Sistem menampilkan list daftar group. 5) Sistem menampilkan form tambah group. 8) Sistem menyimpan data group ke dalam database. 9) Sistem menampilkan keseluruhan daftar group. 2) Sistem menampilkan modul group. 3) Sistem menampilkan list daftar group. 5) Sistem menampilkan form ubah data group. 8) Sistem menyimpan data group ke dalam database. 9) Sistem menampilkan keseluruhan daftar group.

	<p>Menghapus Data Group</p> <ol style="list-style-type: none"> 1) User memilih modul group. 4) User memilih data group yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 2) Sistem menampilkan modul group. 3) Sistem menampilkan list daftar group. 6) Sistem menghapus data group dalam list daftar group dan juga database.
Alternate Flow :	<p>Aksi Aktor :</p> <p>Cari Data Group</p> <ol style="list-style-type: none"> 1) User masuk ke modul group. 4) User mengetikkan keyword nama group di form pencarian. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan modul group. 3) Sistem menampilkan list daftar group. 5) Sistem melakukan pencarian data group sesuai dari keyword yang di masukkan. 6) Sistem menampilkan hasil pencarian.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam modul group sebelum melakukan keseluruhan pengolahan data unit group.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses tambah data group, ubah data group dan hapus data group. Setelah itu data group akan tersimpan ke dalam database sistem.	

4. Bagian Modul Karyawan



Gambar 3.7. Use Case Diagram Modul Karyawan

Tabel 3.16. Skenario Use Case Edit Profil

Nama Use Case :	Edit Profil	
Aktor :	Karyawan	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses edit profil.	
Main Flow :	<p>Aksi Aktor :</p> <p>Mengedit Profil</p> <ol style="list-style-type: none"> 1) User memilih menu profile saya. 4) User menekan tombol icon “setting” 5) User memilih akun saya. 7) User merubah data profil. 8) User menekan tombol “simpan”. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan halaman profil karyawan. 3) Sistem menampilkan keseluruhan data profil. 6) Sistem menampilkan list data profil karyawan. 9) Sistem menyimpan data profil ke dalam database.

		10) Sistem menampilkan keseluruhan data profil.
Alternate Flow :	Aksi Aktor : Lihat Profil 1) User masuk ke menu profile saya.	Reaksi Sistem : 2) Sistem menampilkan halaman profil karyawan. 3) Sistem menampilkan keseluruhan data profil.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam menu profile saya sebelum melakukan keseluruhan proses edit profil.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses edit profil. Setelah itu data profil akan tersimpan ke dalam database sistem.	

Tabel 3.17. Skenario *Use Case* Kelola Status

Nama Use Case :	Kelola Status	
Aktor :	Karyawan	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data status karyawan.	
Main Flow :	Aksi Aktor : Menulis Status 1) User memilih menu timeline. 4) User memasukkan data status. 5) User menekan tombol "post".	Reaksi Sistem : 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data timeline dan form status. 6) Sistem menyimpan data status ke dalam

	<p>Mengupdate Status</p> <ol style="list-style-type: none"> 1) User masuk ke menu timeline. 4) User memilih data status yang akan diupdate. 5) User memilih menu edit. 7) User merubah data status. 8) User menekan tombol “edit”. <p>Menghapus Status</p> <ol style="list-style-type: none"> 1) User memilih menu timeline. 4) User memilih data status yang akan dihapus 5) User menekan tombol “hapus” 	<p>database.</p> <ol style="list-style-type: none"> 7) Sistem menampilkan keseluruhan data timeline. 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data timeline. 6) Sistem menampilkan form ubah status. 9) Sistem menyimpan data status ke dalam database. 10) Sistem menampilkan keseluruhan data timeline. 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data timeline. 6) Sistem menghapus data status dalam list daftar status dan juga database. 7) Sistem menampilkan keseluruhan data timeline.
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Lihat Status</p> <ol style="list-style-type: none"> 1) User masuk ke menu profile saya. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan halaman profil

		karyawan. 3) Sistem menampilkan keseluruhan data profil dan juga status karyawan.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam menu timeline sebelum melakukan keseluruhan pengolahan data status.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses menulis status, update status dan hapus status. Setelah itu data status akan tersimpan ke dalam database sistem.	

Tabel 3.18. Skenario *Use Case* Kelola Komentar

Nama Use Case :	Kelola Komentar	
Aktor :	Karyawan	
Deskripsi :	Use Case ini berfungsi untuk mendeskripsikan keseluruhan proses pengelolaan data komentar karyawan.	
Main Flow :	Aksi Aktor : Menulis Komentar 1) User memilih menu timeline. 4) User memilih data yang akan dikomentari. 5) User memilih komentar. 7) User memasukkan data komentar. 8) User menekan tombol "post".	Reaksi Sistem : 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data timeline. 6) Sistem menampilkan form data komentar. 9) Sistem menyimpan data komentar ke dalam database. 10) Sistem menampilkan keseluruhan data timeline.

	<p>Mengupdate Komentar</p> <ol style="list-style-type: none"> 1) User masuk ke menu timeline. 4) User memilih data yang akan diupdate komentarnya. 5) User memilih menu edit. 7) User merubah data komentar. 8) User menekan tombol “edit”. <p>Menghapus Komentar</p> <ol style="list-style-type: none"> 1) User memilih menu timeline. 4) User memilih data komentar yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data timeline. 6) Sistem menampilkan form ubah komentar. 9) Sistem menyimpan data komentar ke dalam database. 10) Sistem menampilkan keseluruhan data timeline. 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data timeline. 6) Sistem menghapus data komentar dalam list daftar komentar dan juga database. 7) Sistem menampilkan keseluruhan data timeline.
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Lihat Komentar</p> <ol style="list-style-type: none"> 1) User masuk ke menu timeline. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan halaman timeline. 3) Sistem menampilkan keseluruhan data

	4) User memilih data yang dikomentari.	5) Sistem menampilkan list daftar komentar.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam menu timeline sebelum melakukan keseluruhan pengolahan data komentar.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses menulis komentar, update komentar dan hapus komentar. Setelah itu data komentar akan tersimpan ke dalam database sistem.	

Tabel 3.19. Skenario Use Case Private Message

Nama Use Case :	Private Message	
Aktor :	Karyawan	
Deskripsi :	Use Case ini berfungsi untuk mengirimkan pesan singkat secara personal.	
Main Flow :	Aksi Aktor : Mengirimkan Pesan 1) User memilih menu pesan. 4) User memilih menu tulis. 6) User memasukkan data pesan. 7) User menekan tombol "kirim"	Reaksi Sistem : 2) Sistem menampilkan halaman pesan. 3) Sistem menampilkan keseluruhan data pesan. 5) Sistem menampilkan form input data pesan. 8) Sistem mengirimkan pesan. 9) Sistem menyimpan data pesan ke dalam database. 10) Sistem menampilkan keseluruhan data pesan.

	<p>Menghapus Pesan</p> <ol style="list-style-type: none"> 1) User memilih menu pesan. 4) User memilih data pesan yang akan dihapus 5) User menekan tombol “hapus” 	<ol style="list-style-type: none"> 2) Sistem menampilkan halaman pesan. 3) Sistem menampilkan keseluruhan data pesan. 6) Sistem menghapus data pesan dalam list daftar pesan dan juga database. 7) Sistem menampilkan keseluruhan data pesan.
Alterate Flow :	<p>Aksi Aktor :</p> <p>Lihat Pesan</p> <ol style="list-style-type: none"> 1) User masuk ke menu pesan. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan halaman pesan. 3) Sistem menampilkan keseluruhan data pesan.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam menu pesan sebelum melakukan proses pengiriman pesan.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses pengiriman pesan. Setelah itu data pesan akan tersimpan ke dalam database sistem.	

Tabel 3.20. Skenario Use Case Join Group

Nama Use Case :	Join Group
Aktor :	Karyawan
Deskripsi :	Use Case ini berfungsi untuk melakukan proses join group.

<p>Main Flow :</p>	<p>Aksi Aktor :</p> <p>Join Group</p> <ol style="list-style-type: none"> 1) User memilih menu group. 4) User memilih group yang diinginkan. 6) User menekan tombol “bergabung” 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan halaman group. 3) Sistem menampilkan keseluruhan data group. 5) Sistem menampilkan halaman group yang dipilih. 7) Sistem menyimpan data user yang join group ke dalam database. 8) Sistem menampilkan keseluruhan data group.
<p>Alternate Flow :</p>	<p>Aksi Aktor :</p> <p>Lihat Group</p> <ol style="list-style-type: none"> 1) User masuk ke menu group. 4) User memilih group yang ingin dilihat. 	<p>Reaksi Sistem :</p> <ol style="list-style-type: none"> 2) Sistem menampilkan halaman group. 3) Sistem menampilkan keseluruhan data group. 5) Sistem menampilkan halaman group yang dipilih.
<p>Kondisi Awal :</p>	<p>Aktor harus masuk terlebih dahulu ke dalam menu group sebelum melakukan proses join group.</p>	
<p>Kondisi Akhir :</p>	<p>Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses join group.. Setelah itu data user yang join group akan tersimpan ke dalam database sistem.</p>	

Tabel 3.21. Skenario Use Case Add Friend

Nama Use Case :	Add Friend	
Aktor :	Karyawan	
Deskripsi :	Use Case ini berfungsi untuk melakukan proses menambahkan teman.	
Main Flow :	Aksi Aktor : Add Friend 1) User memilih menu cari teman. 4) User menambahkan teman yang diinginkan.	Reaksi Sistem : 2) Sistem menampilkan halaman cari teman. 3) Sistem menampilkan keseluruhan data teman. 5) Sistem menambahkan teman. 6) Sistem menyimpan data penambahan teman ke dalam database. 7) Sistem menampilkan keseluruhan data teman.
Alternate Flow :	Aksi Aktor : Lihat Friends 1) User masuk ke menu teman.	Reaksi Sistem : 2) Sistem menampilkan halaman teman. 3) Sistem menampilkan keseluruhan data teman.
Kondisi Awal :	Aktor harus masuk terlebih dahulu ke dalam menu teman sebelum melakukan proses penambahan teman.	
Kondisi Akhir :	Jika use case sukses, maka aktor akan masuk ke dalam sistem untuk melakukan proses penambahan teman. Setelah itu data penambahan teman akan tersimpan ke dalam database sistem.	

3.4 Desain Sistem

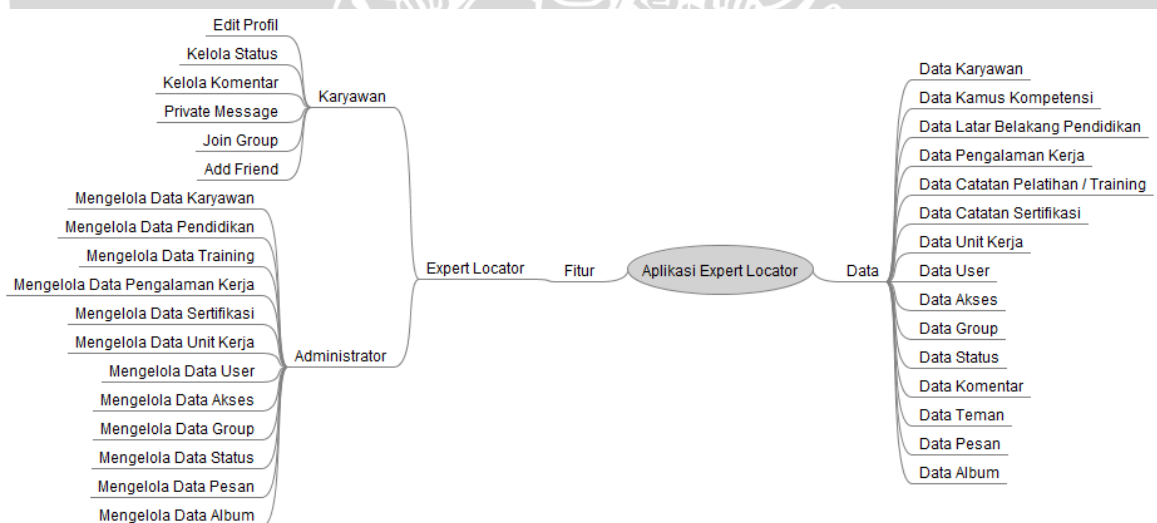
Perancangan dalam pembangunan perangkat lunak dengan menggunakan pendekatan berorientasi objek yang dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak.

3.4.1 Rancangan Database

Merupakan suatu proses untuk menentukan isi serta pengaturan data yang nantinya akan dibutuhkan dalam sistem ini. Untuk rancangan database pada sistem *expert locator* ini ditunjukkan pada gambar 3.8. Dalam rancangan database tersebut terdapat 5 tabel yang sudah diperoleh dari perusahaan dari hasil *interview* yang meliputi : tabel karyawan, tabel pendidikan, tabel pengalaman kerja, tabel training dan tabel sertifikasi.

3.4.2 Mind Mapping

Mind Mapping digunakan untuk melakukan *breakdown* atau pembagian setiap proses pekerjaan menjadi lebih detail, hal ini dimaksudkan agar proses perencanaan sistem tersebut memiliki item-item pekerjaan yang cukup terperinci, seperti yang di tunjukan pada gambar 3.9.



Gambar 3.9 Mind Mapping

Dalam Implementasi Aplikasi *Expert Locator* di dalam *Knowledge Management System* PT. Petrokimia Gresik terdapat tiga spesifikasi sistem yaitu:

a. Data

Merupakan salah satu prioritas utama dalam pembuatan Aplikasi *Expert Locator* ini. Dan data yang digunakan yaitu berupa data karyawan, data kamus

kompetensi, data pendidikan, data pengalaman kerja, data pelatihan/training, data catatan sertifikasi, data unit kerja, data user, data akses, data group, data status, data komentar, data teman, data pesan dan data album.

b. Fitur

Merupakan prioritas kedua setelah data. Fitur-fitur yang terdapat pada Aplikasi *Expert Locator* ini diantaranya :

Fitur untuk admin yaitu:

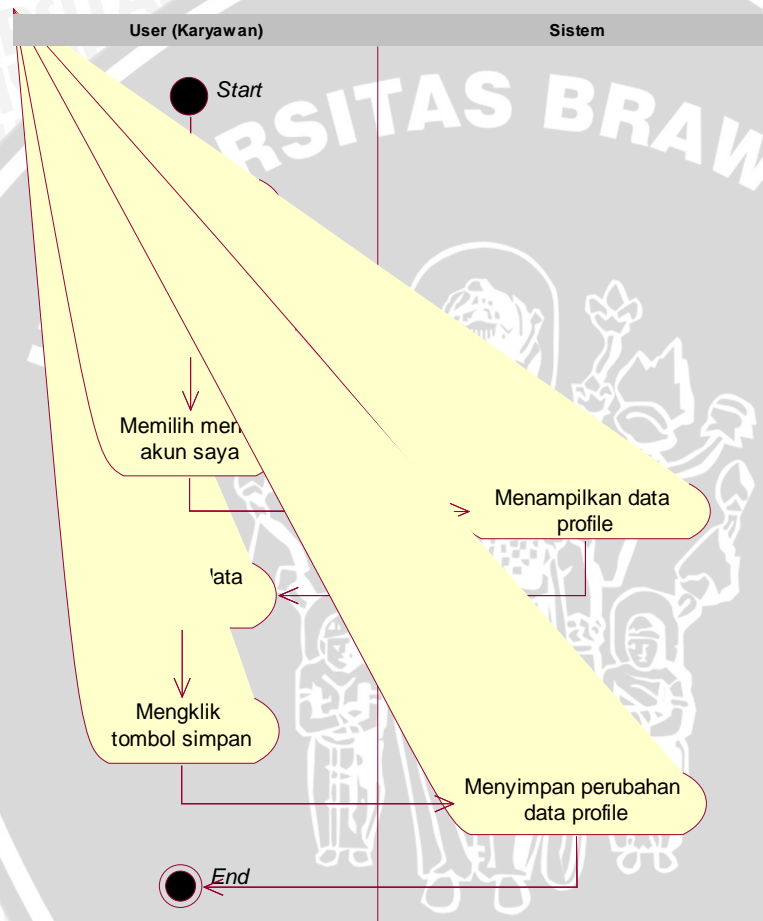
- a) Mengelola Data Karyawan digunakan untuk (memasukkan, mengubah, dan menghapus data) karyawan ke dalam aplikasi ini.
- b) Mengelola Data Pendidikan digunakan untuk (memasukkan, mengubah, dan menghapus data) pendidikan ke dalam aplikasi ini.
- c) Mengelola Data Training digunakan untuk (memasukkan, mengubah, dan menghapus data) training ke dalam aplikasi ini.
- d) Mengelola Data Pengalaman Kerja digunakan untuk (memasukkan, mengubah, dan menghapus data) pengalaman kerja ke dalam aplikasi ini.
- e) Mengelola Data Sertifikasi digunakan untuk (memasukkan, mengubah, dan menghapus data) sertifikasi ke dalam aplikasi ini.
- f) Mengelola Data Unit Kerja digunakan untuk (memasukkan, mengubah, dan menghapus data) unit kerja ke dalam aplikasi ini.
- g) Mengelola Data User digunakan untuk (memasukkan, mengubah, dan menghapus data) user ke dalam aplikasi ini.
- h) Mengelola Data Akses digunakan untuk (memasukkan, mengubah, dan menghapus data) akses ke dalam aplikasi ini.
- i) Mengelola Data Group digunakan untuk (memasukkan, mengubah, dan menghapus data) group ke dalam aplikasi ini.
- j) Mengelola Data Status digunakan untuk (membuat, mengubah, dan menghapus data) status ke dalam aplikasi ini.
- k) Mengelola Data Pesan digunakan untuk (membuat, mengubah, dan menghapus data) pesan ke dalam aplikasi ini.
- l) Mengelola Data Album digunakan untuk (membuat, mengubah, dan menghapus data) album ke dalam aplikasi ini.

3.4.3 Activity Diagram

Activity diagram atau diagram aktivitas lebih memfokuskan diri pada eksekusi dan alur sistem daripada bagaimana sistem itu dirakit. Diagram ini tidak hanya memodelkan software melainkan memodelkan model bisnis juga. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi.

a. Activity karyawan edit profil

Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan edit *profil*, seperti yang ditunjukkan pada gambar 3.10.

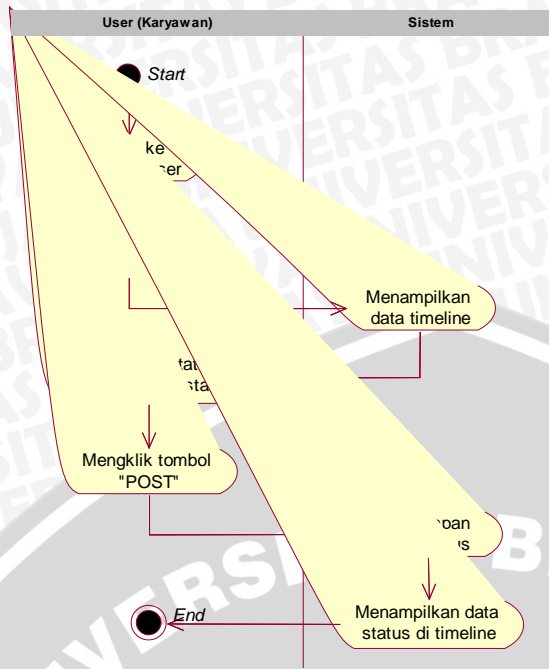


Gambar 3.10 Activity karyawan melakukan edit *profil*

b. Activity karyawan kelola status.

• Activity menulis status

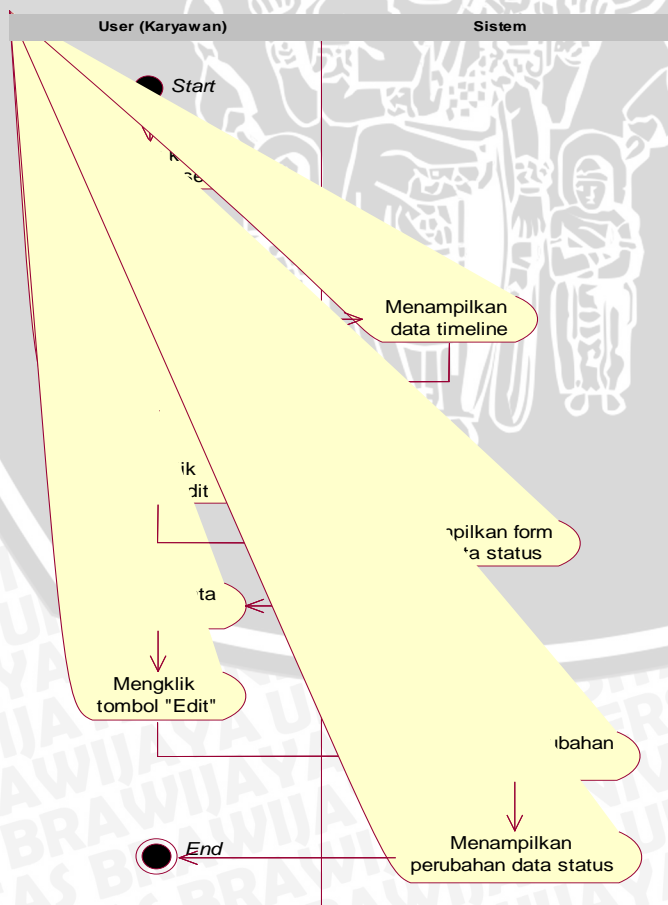
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam menulis status, seperti yang ditunjukkan pada gambar 3.11.



Gambar 3.11 Activity karyawan menulis status

- *Activity update status*

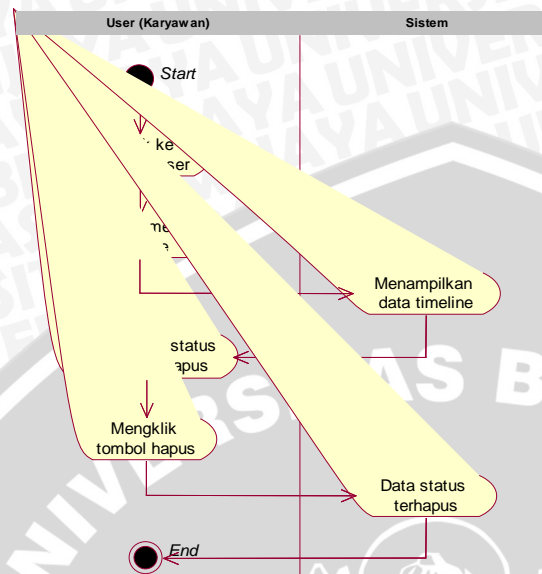
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan *update status*, seperti yang ditunjukkan pada gambar 3.12.



Gambar 3.12 Activity karyawan melakukan *update status*

- *Activity* hapus status

Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan hapus status, seperti yang ditunjukkan pada gambar 3.13.

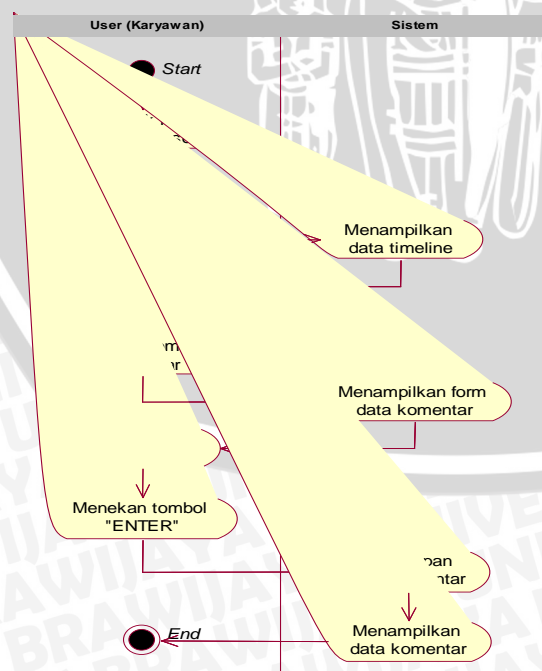


Gambar 3.13 *Activity* karyawan melakukan hapus status

- c. *Activity* karyawan kelola komentar.

- *Activity* menulis komentar

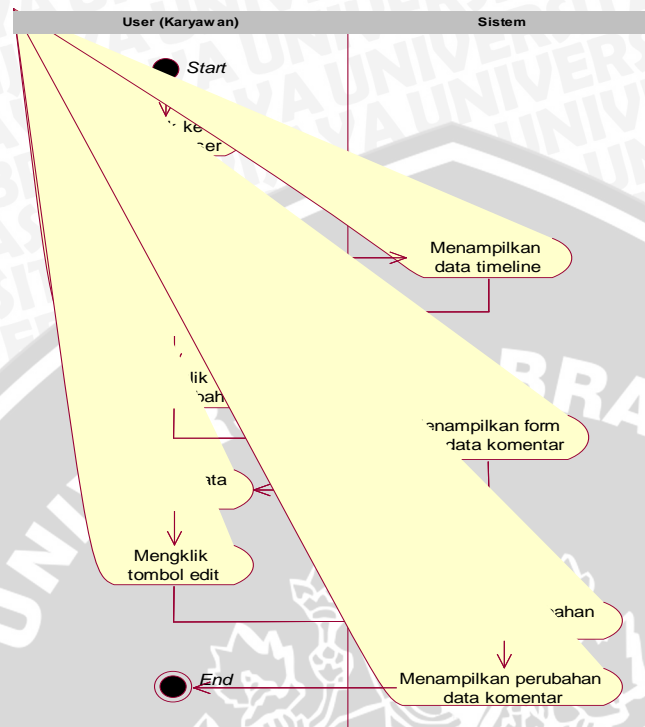
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam menulis komentar, seperti yang ditunjukkan pada gambar 3.14.



Gambar 3.14 *Activity* karyawan menulis komentar

- *Activity update* komentar

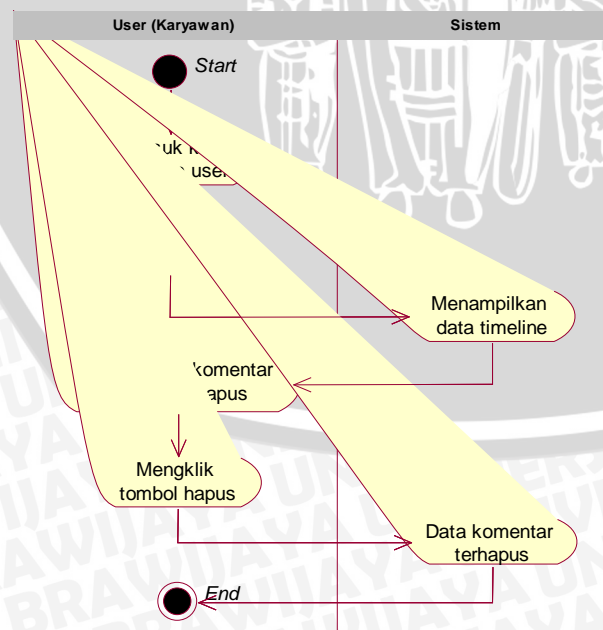
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan *update* komentar, seperti yang ditunjukkan pada gambar 3.15.



Gambar 3.15 *Activity* karyawan melakukan *update* komentar

- *Activity hapus* komentar

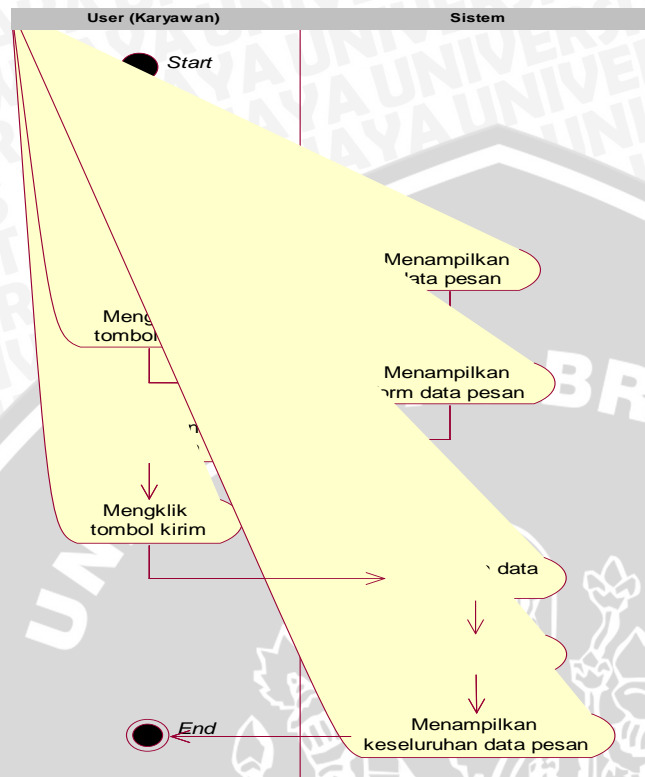
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan hapus komentar, seperti yang ditunjukkan pada gambar 3.16.



Gambar 3.16 *Activity* karyawan melakukan hapus komentar

d. *Activity* karyawan private message

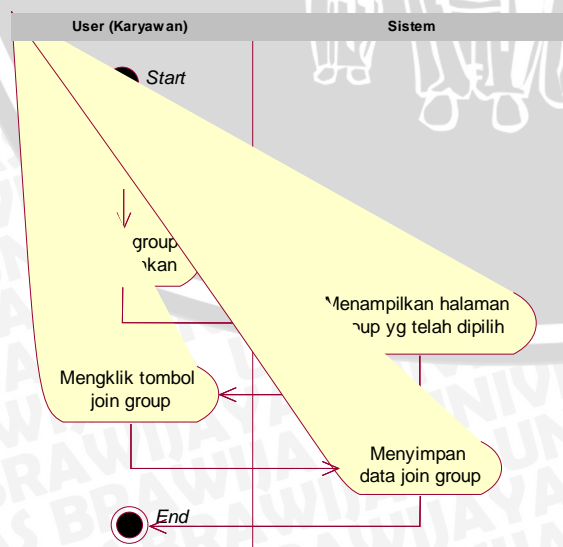
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan *private message*, seperti yang ditunjukkan pada gambar 3.17.



Gambar 3.17 *Activity* karyawan melakukan *private message*

e. *Activity* karyawan join group

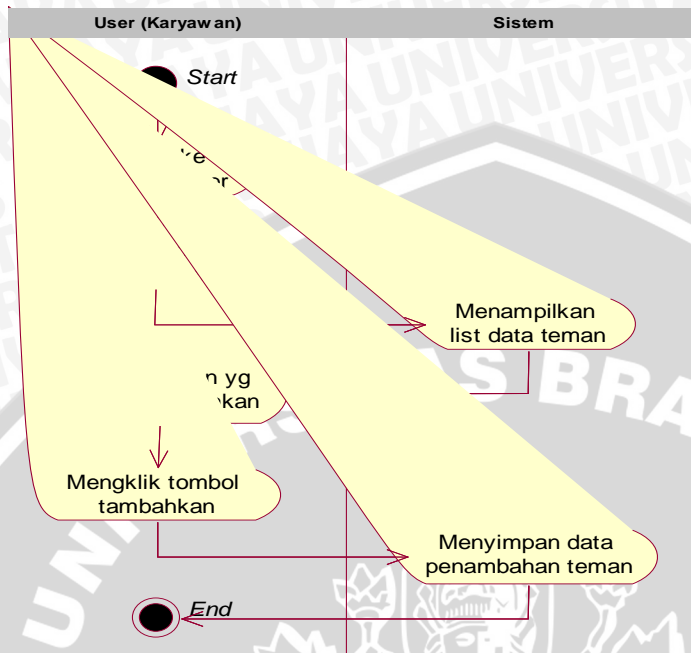
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan *join group*, seperti yang ditunjukkan pada gambar 3.18.



Gambar 3.18 *Activity* karyawan melakukan *join group*

f. *Activity add friend*

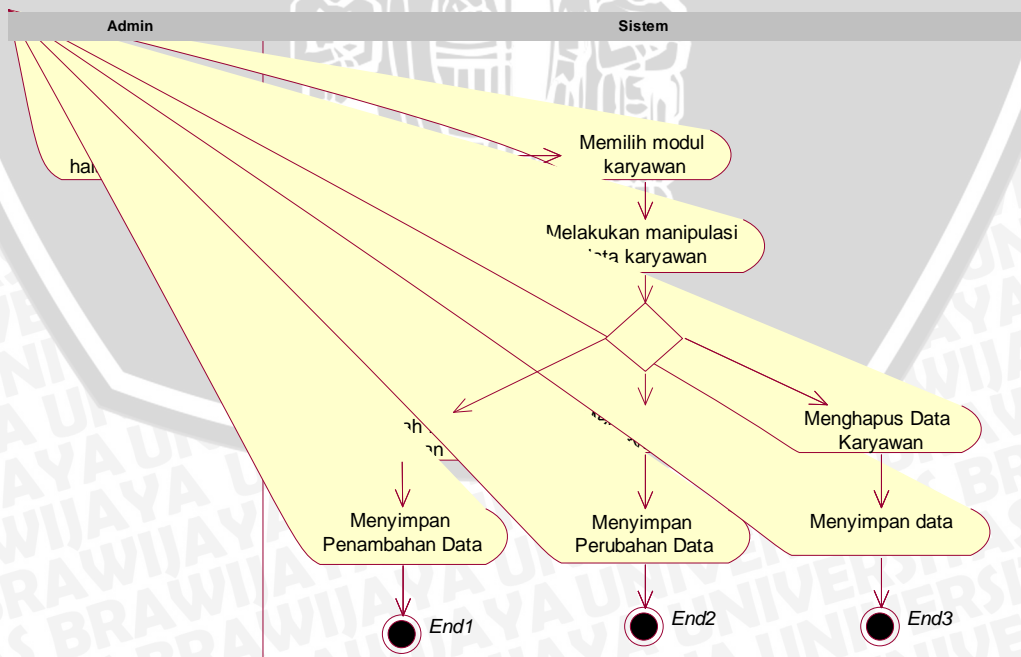
Pada *activity* ini menggambarkan bagaimana alur kerja karyawan dalam melakukan penambahan teman, seperti yang ditunjukkan pada gambar 3.19.



Gambar 3.19 *Activity* karyawan melakukan add friend

g. *Activity admin kelola data karyawan*

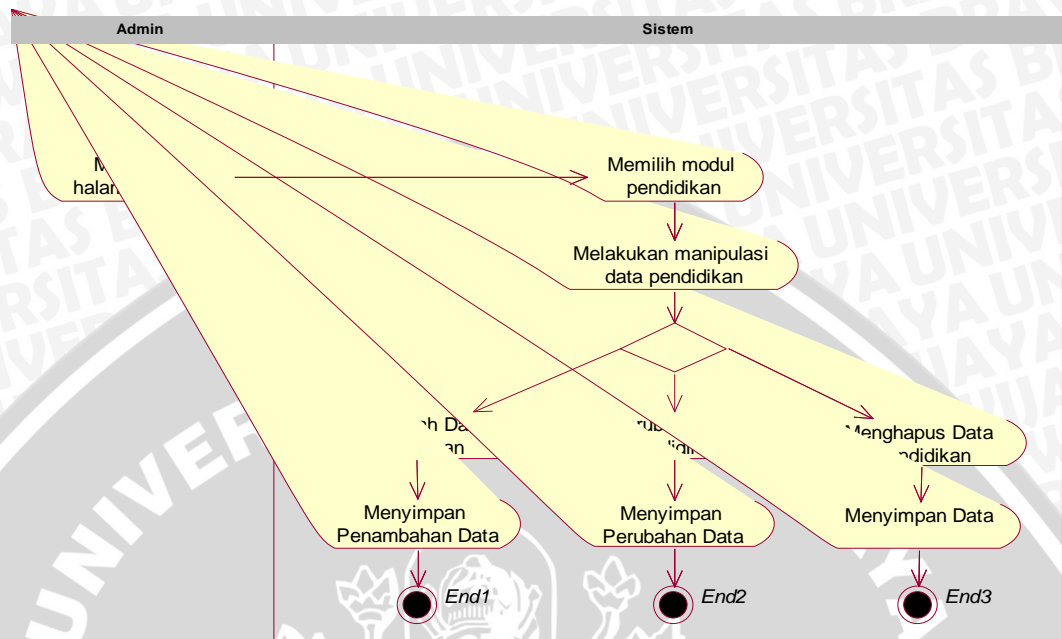
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data karyawan, seperti yang ditunjukkan pada gambar 3.20.



Gambar 3.20 *Activity* admin kelola data karyawan

h. *Activity* admin kelola data pendidikan

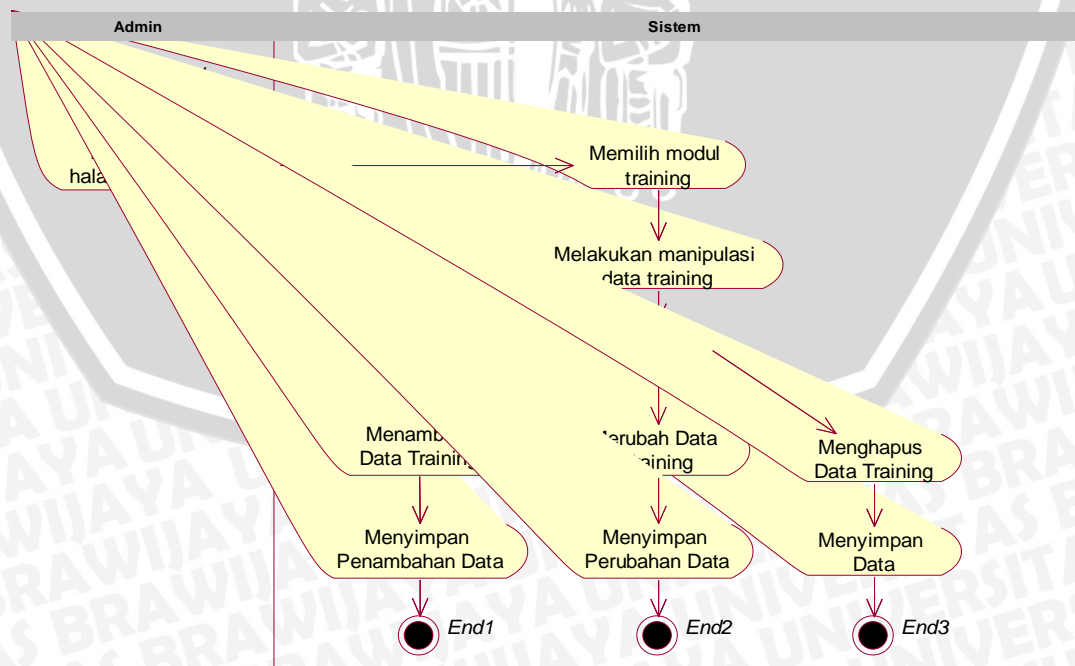
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data pendidikan, seperti yang ditunjukkan pada gambar 3.21



Gambar 3.21 *Activity* admin kelola data pendidikan

i. *Activity* admin kelola data training

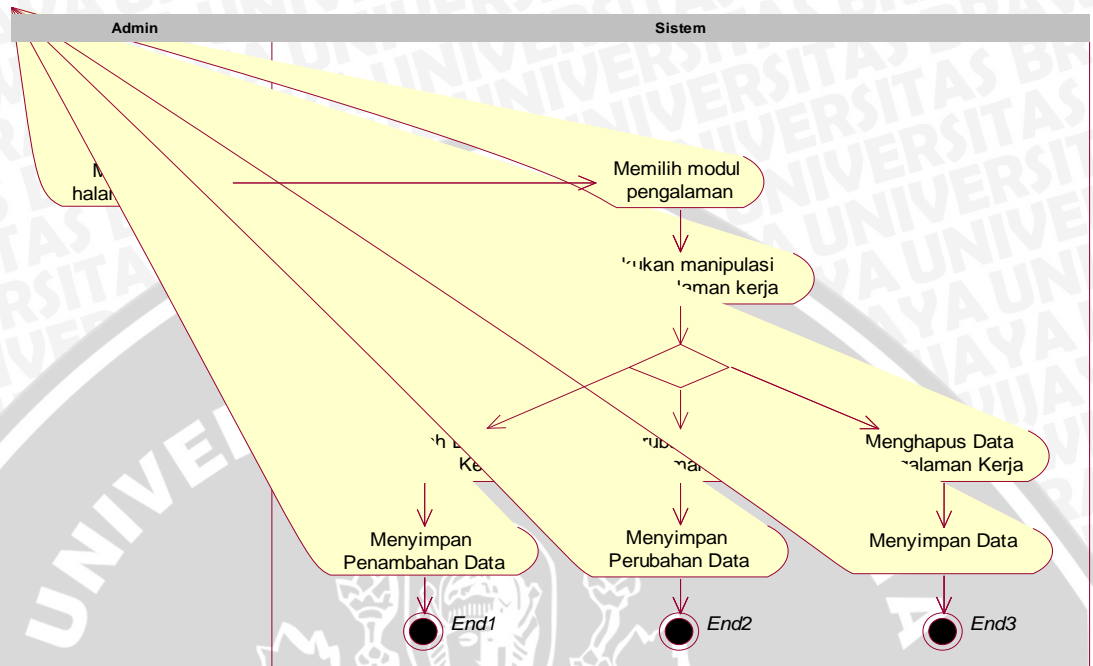
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data training, seperti yang ditunjukkan pada gambar 3.22



Gambar 3.22 *Activity* admin kelola data training

j. *Activity* admin kelola data pengalaman kerja

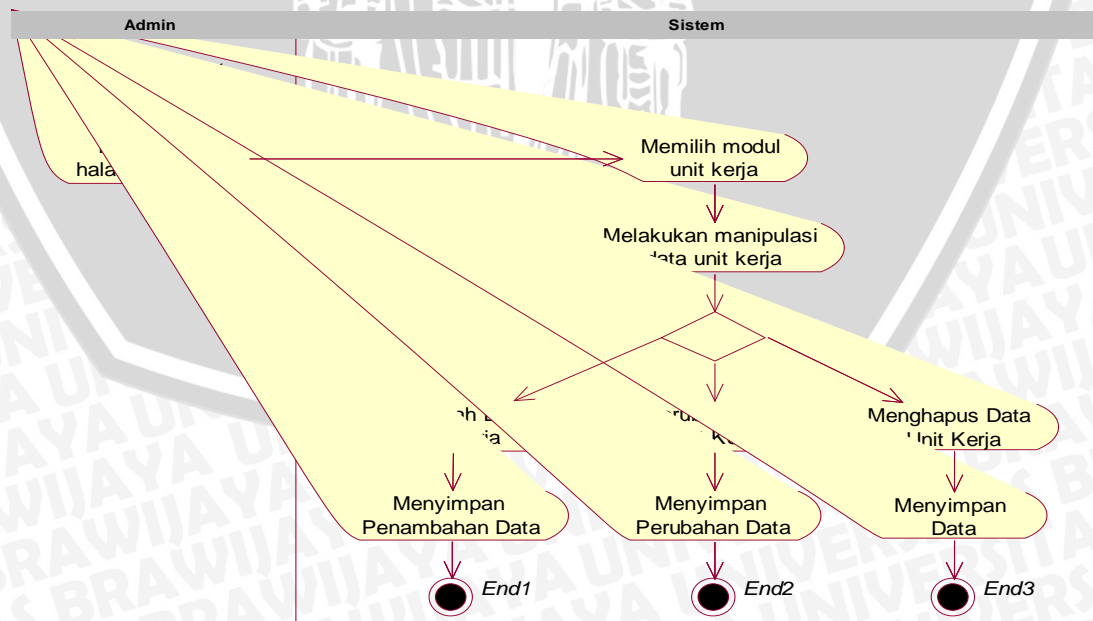
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data pengalaman kerja, seperti yang ditunjukkan pada gambar 3.23



Gambar 3.23 *Activity* admin kelola data pengalaman kerja

k. *Activity* admin kelola data unit kerja

Pada *Activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data unit kerja, seperti yang ditunjukkan pada gambar 3.24

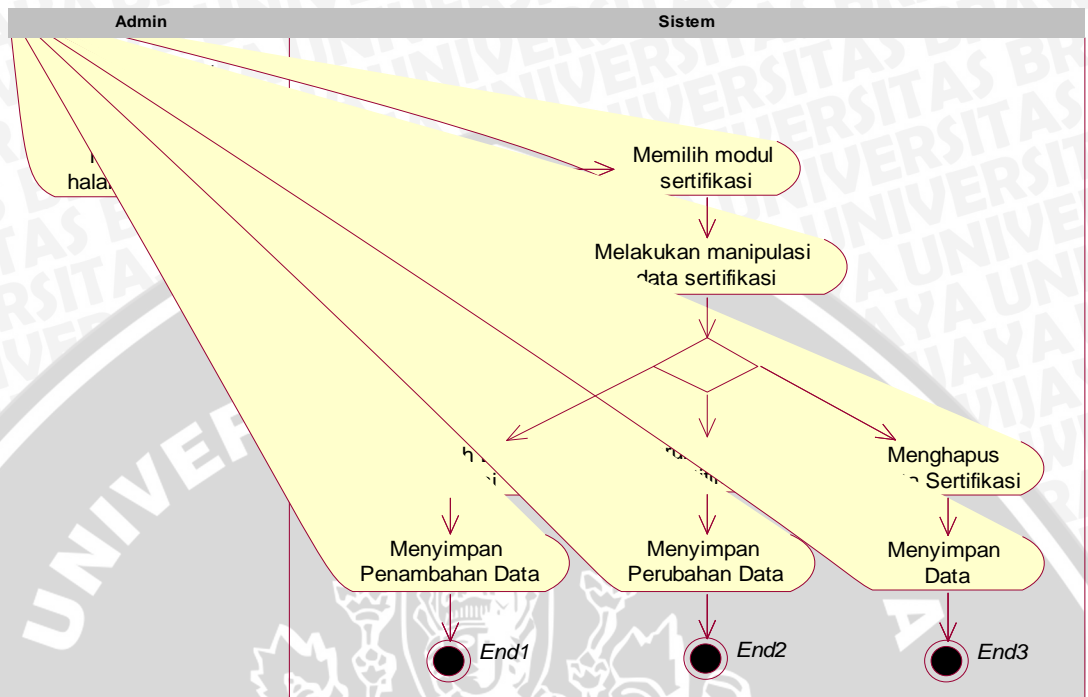


Gambar 3.24 *Activity* admin kelola data unit kerja



l. *Activity* admin kelola data sertifikasi

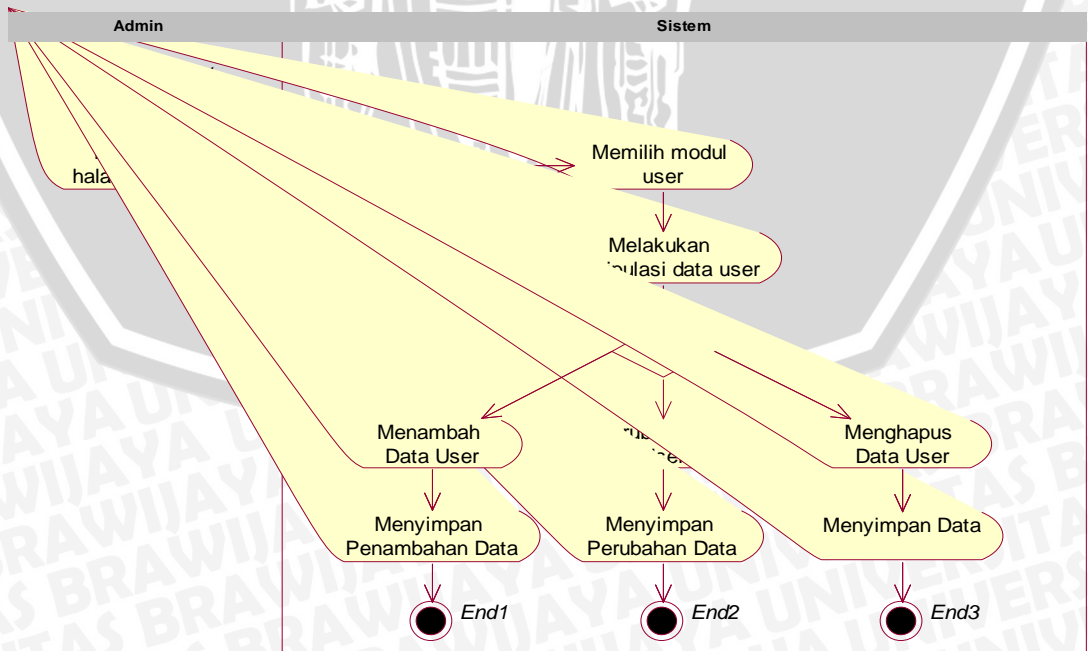
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data sertifikasi, seperti yang ditunjukkan pada gambar 3.25



Gambar 3.25 *Activity* admin kelola data sertifikasi

m. *Activity* admin kelola data user

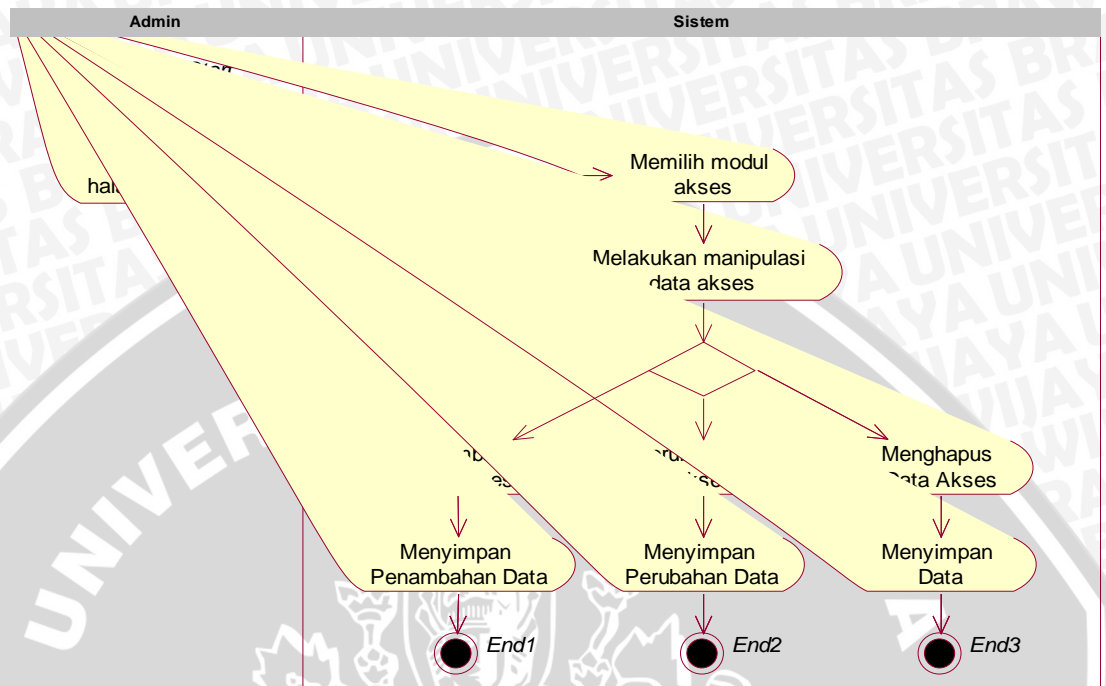
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data user, seperti yang ditunjukkan pada gambar 3.26



Gambar 3.26 *Activity* admin kelola data user

n. *Activity admin kelola data akses*

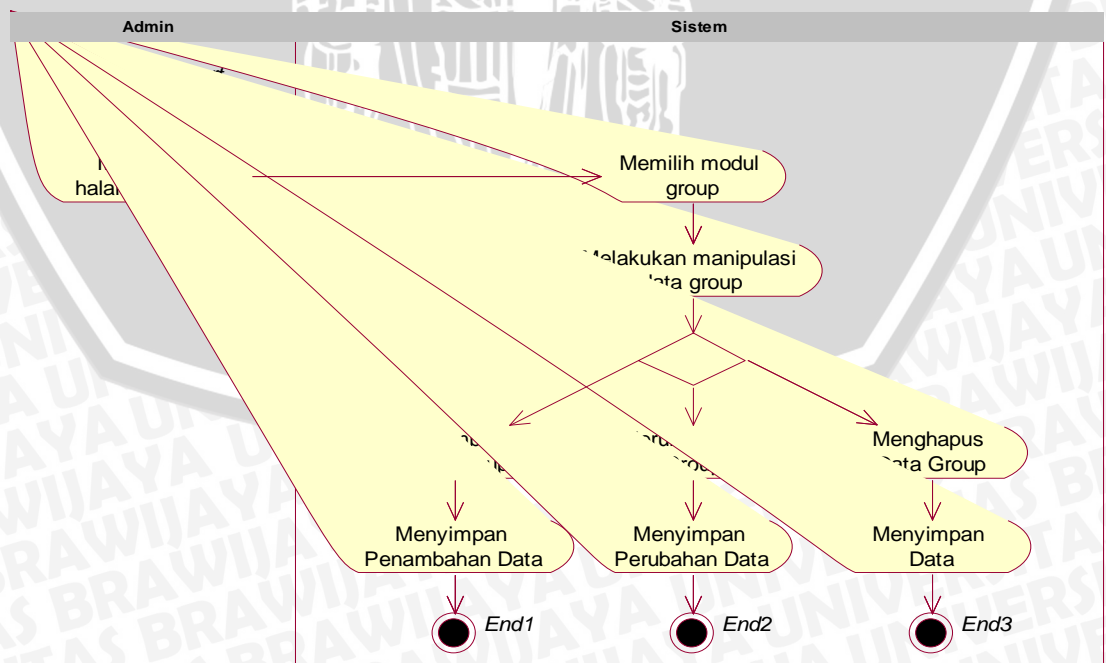
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data akses, seperti yang ditunjukkan pada gambar 3.27



Gambar 3.27 *Activity admin kelola data akses*

o. *Activity admin kelola data group*

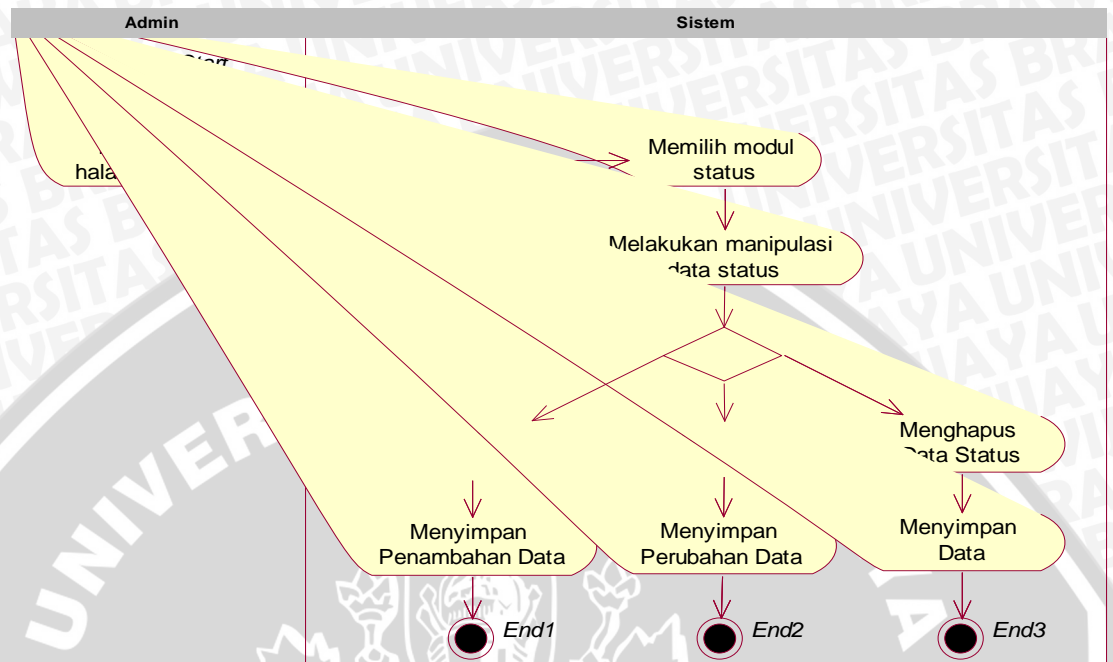
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data group, seperti yang ditunjukkan pada gambar 3.28



Gambar 3.28 *Activity admin kelola data group*

p. *Activity* admin kelola data status karyawan

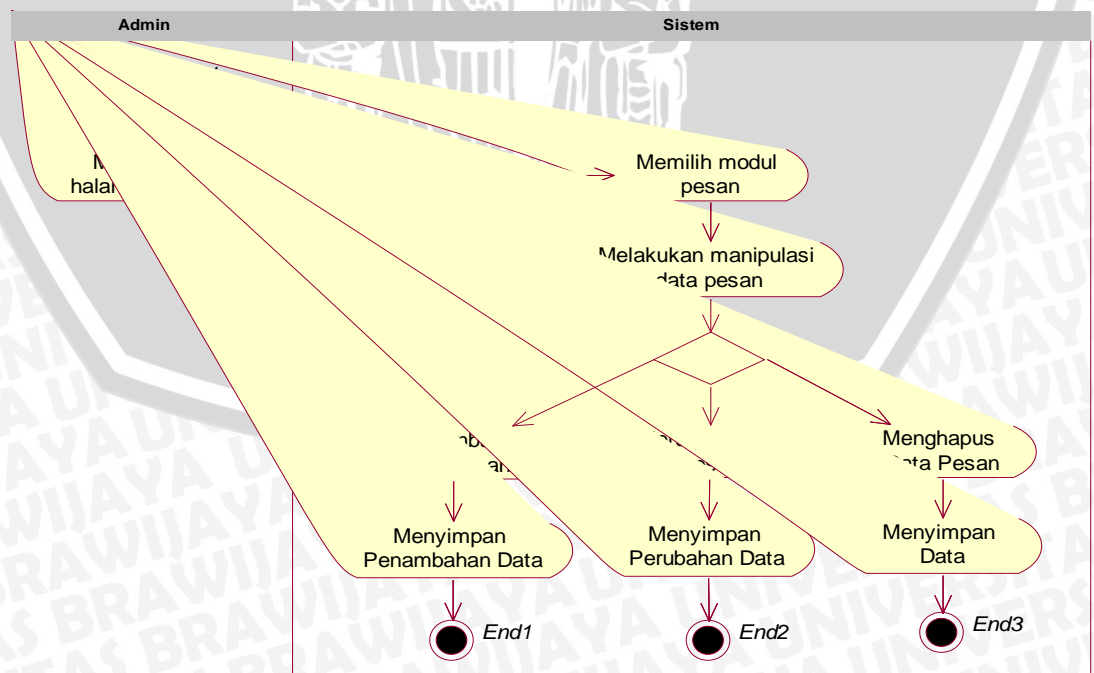
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data status, seperti yang ditunjukkan pada gambar 3.29



Gambar 3.29 *Activity* admin kelola data status karyawan

q. *Activity* admin kelola data pesan

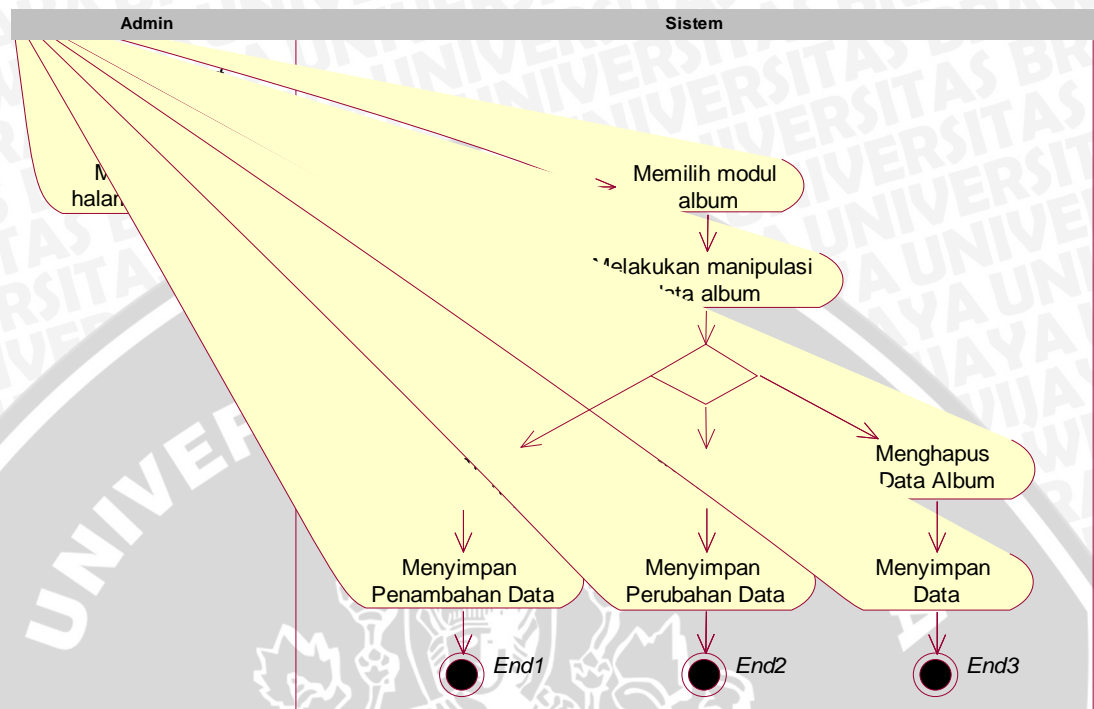
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data pesan, seperti yang ditunjukkan pada gambar 3.30



Gambar 3.30 *Activity* admin kelola data pesan

r. *Activity admin kelola data album*

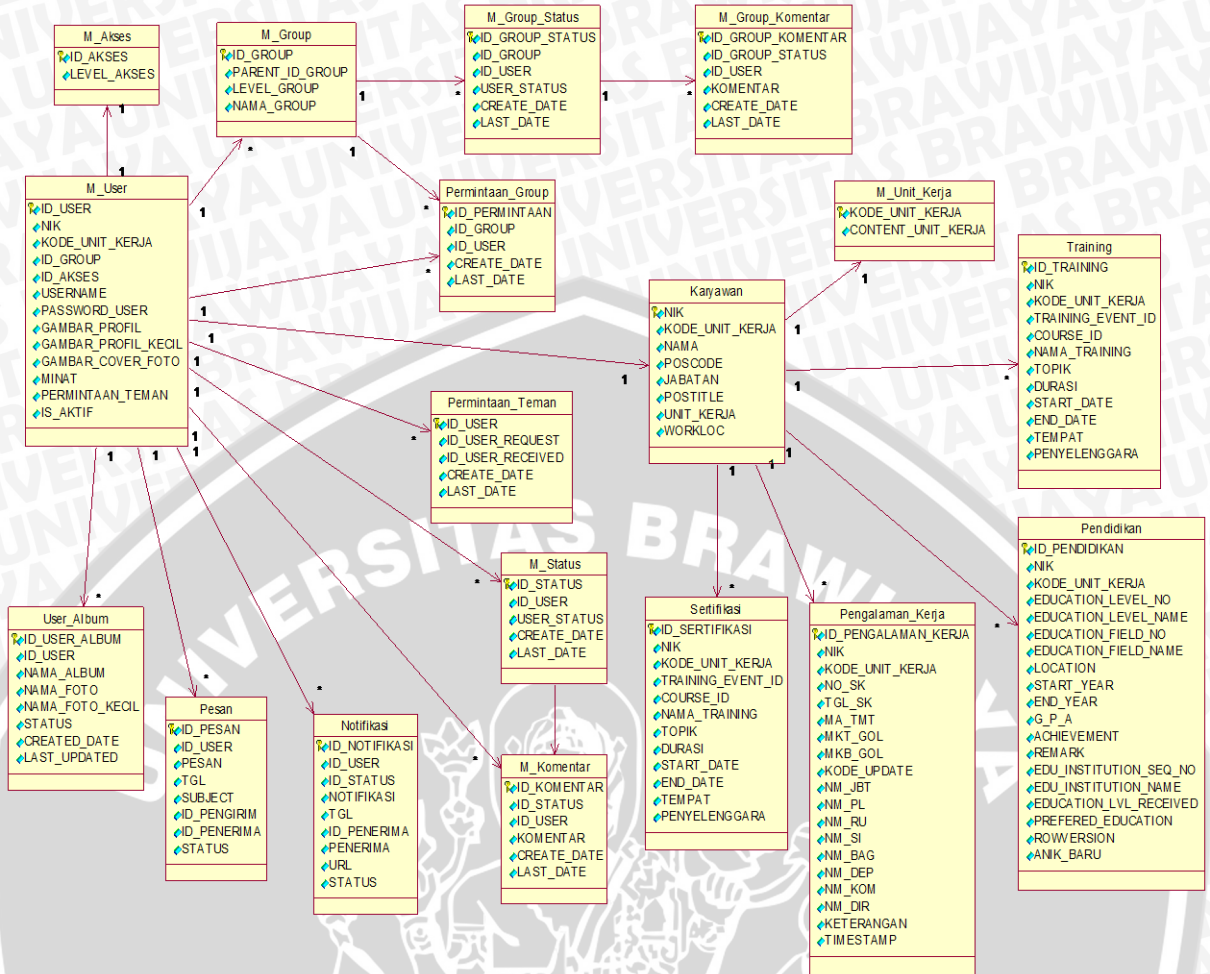
Pada *activity* ini menggambarkan bagaimana alur kerja admin dalam mengelola data album, seperti yang ditunjukkan pada gambar 3.31



Gambar 3.31 *Activity* admin kelola data album

3.4.4 Class Diagram

Class diagram atau diagram kelas merupakan kumpulan kelas-kelas objek, oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas. Secara teknis, Pender (dalam Pudjo Widodo, 2011:39) mengartikan sebuah kelas sebagai suatu definisi sumber daya yang termasuk di dalamnya informasi-informasi yang menggambarkan fitur suatu entitas dan bagaimana penggunaannya. Sedangkan objek adalah entitas yang bersifat unik yang mengikuti aturan-aturan yang sudah didefinisikan dalam kelasnya.



Gambar 3.32 Class Diagram

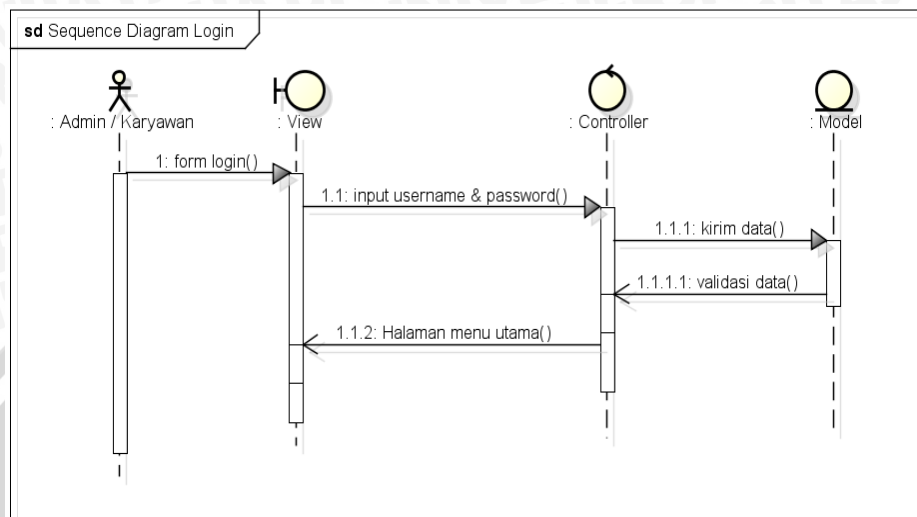
3.4.5 Sequence Diagram

Diagram Sekuen atau *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antarobjek. Untuk itu dalam menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Pada gambar dibawah ini merupakan *sequence diagram* dari sistem ini, diantaranya :



a. Login

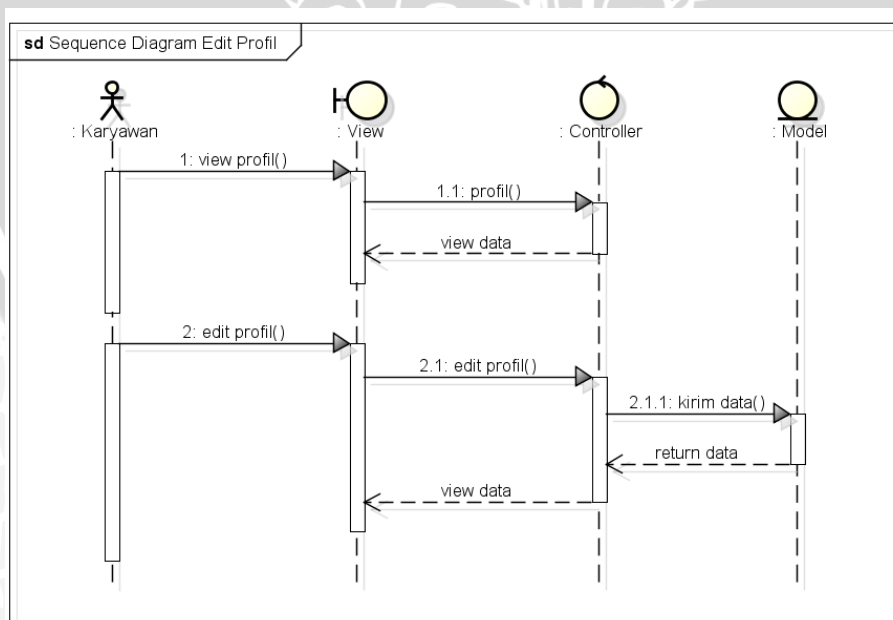
Gambar dibawah ini merupakan gambar diagram sekuen login yang menjelaskan interaksi admin / karyawan dengan sistem dalam mengakses masing – masing halaman menu utama.



Gambar 3.33 Sequence diagram login

b. Edit Profil

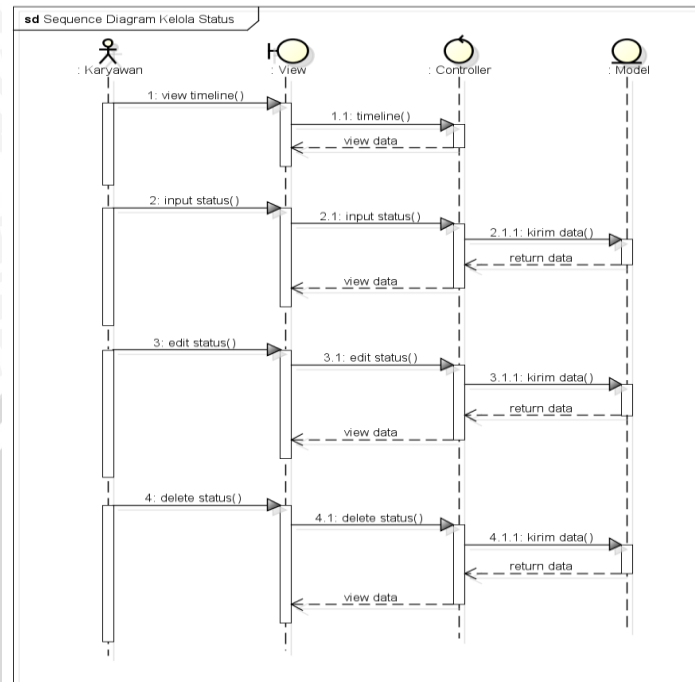
Gambar dibawah ini merupakan gambar diagram sekuen edit profil yang menjelaskan interaksi karyawan dengan sistem dalam mengedit data profil.



Gambar 3.34 Sequence diagram edit profile

c. Kelola Status

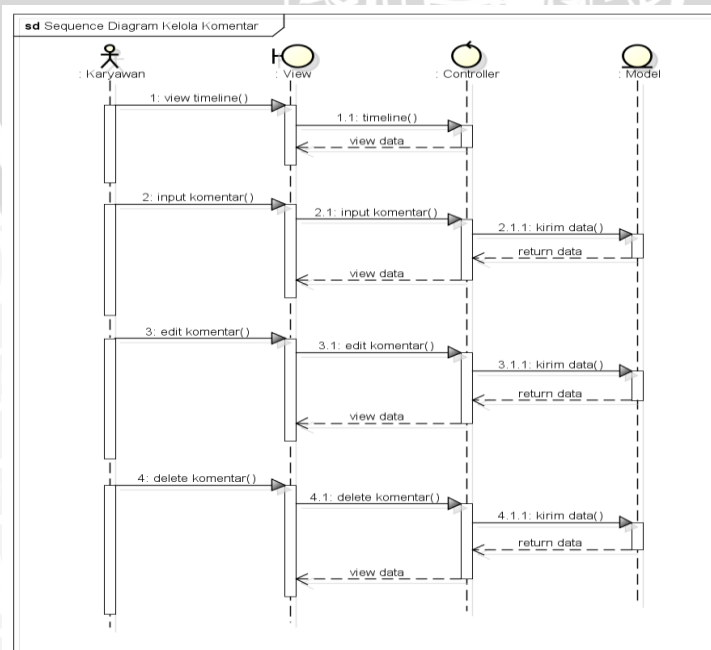
Gambar dibawah ini merupakan gambar diagram sekuen kelola status yang menjelaskan interaksi karyawan dengan sistem dalam mengelola status.



Gambar 3.35 Sequence diagram menulis status

d. Kelola Komentar

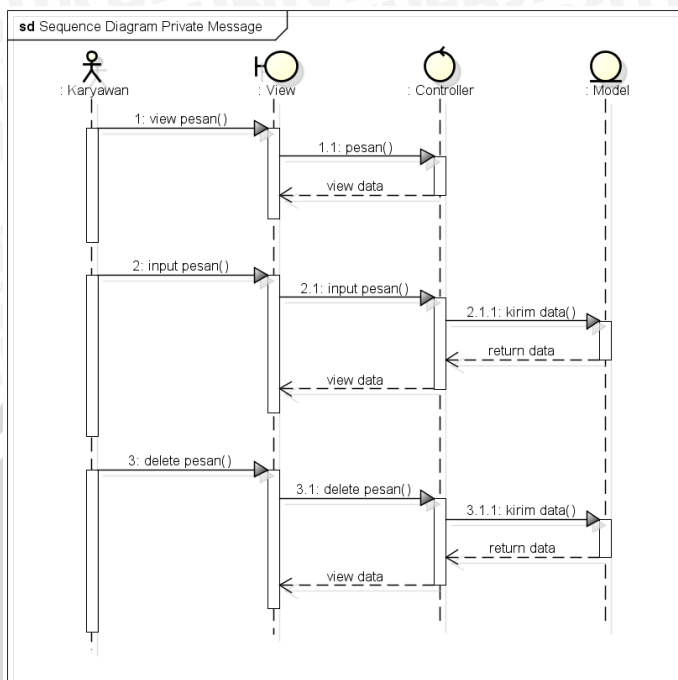
Gambar dibawah ini merupakan gambar diagram sekuen kelola komentar yang menjelaskan interaksi karyawan dengan sistem dalam mengelola komentar.



Gambar 3.36 Sequence diagram menulis komentar

e. Private Message

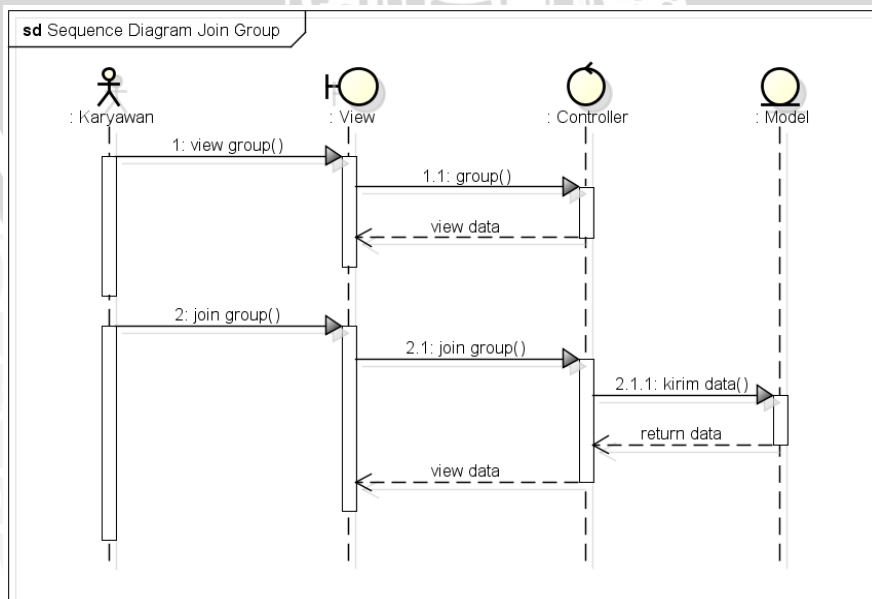
Gambar dibawah ini merupakan gambar diagram sekuen *private message* yang menjelaskan interaksi karyawan dengan sistem dalam mengelola pesan.



Gambar 3.37 Sequence diagram mengirim pesan

f. Join Group

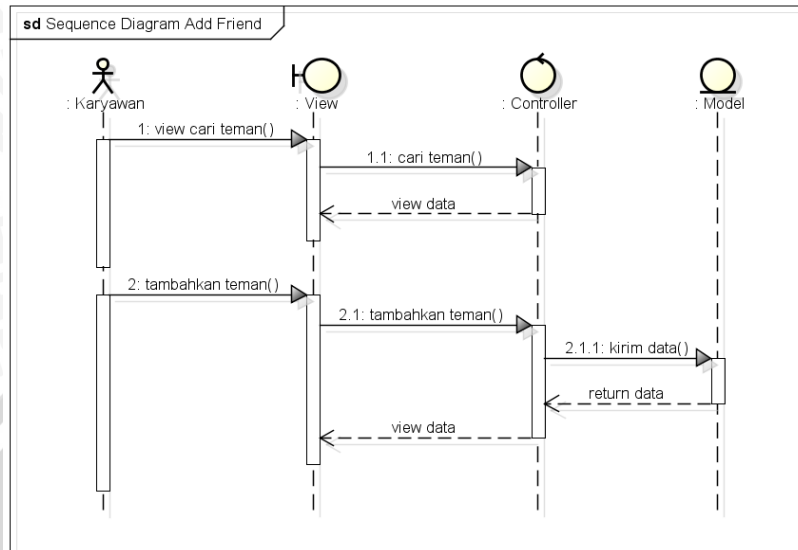
Gambar dibawah ini merupakan gambar diagram sekuen *join group* yang menjelaskan interaksi karyawan dengan sistem dalam melakukan *join group*.



Gambar 3.38 Sequence diagram join group

g. Add Friend

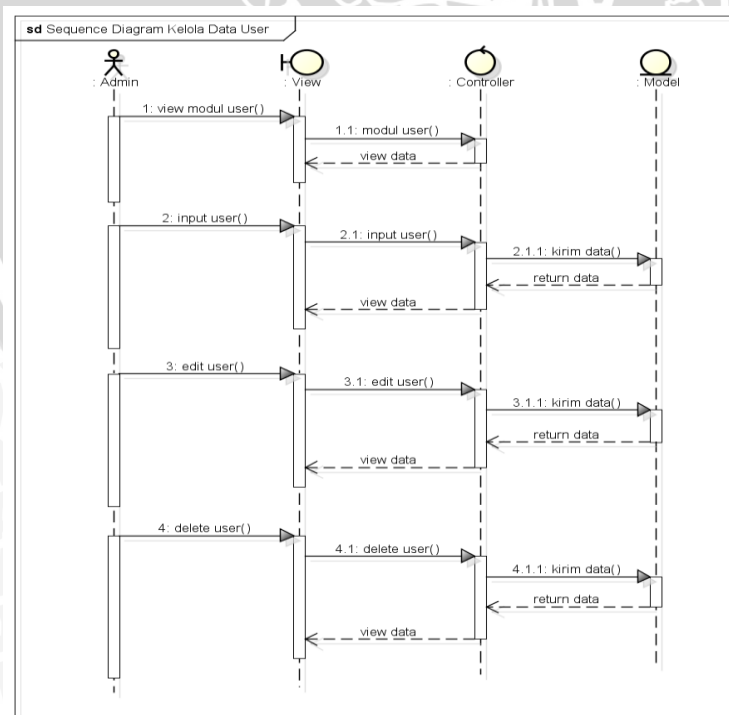
Gambar dibawah ini merupakan gambar diagram sekuen *add friend* yang menjelaskan interaksi karyawan dengan sistem dalam melakukan *add friend*.



Gambar 3.39 Sequence diagram permintaan teman

h. Kelola Data User

Gambar dibawah ini merupakan gambar diagram sekuen kelola data user yang menjelaskan interaksi karyawan dengan sistem dalam mengelola data user.



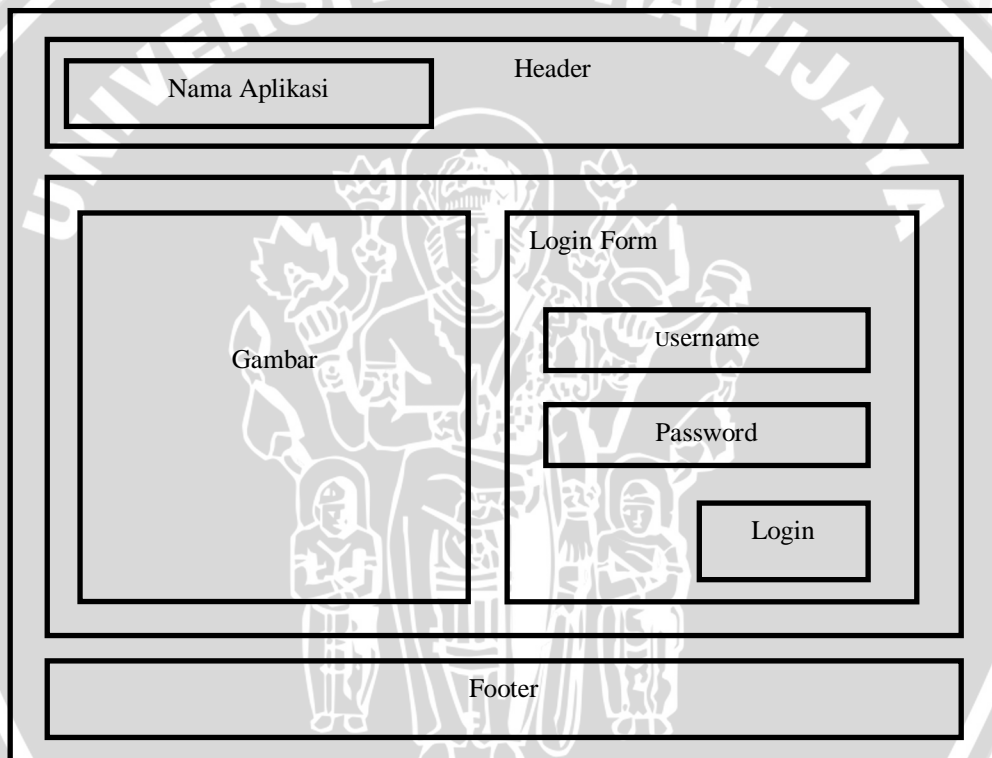
Gambar 3.40 Sequence diagram kelola data user

3.5 Desain Interface

Merancang interface yang efektif untuk sistem perangkat lunak. Efektif artinya siap digunakan dan hasilnya sesuai dengan kebutuhan. Kebutuhan disini adalah kebutuhan penggunanya yaitu PT. Petrokimia Gresik yang diperoleh dari hasil wawancara dengan staff IT bagian departemen teknologi informasi yang juga telah mengacu pada kajian pustaka yang sudah dijelaskan sebelumnya pada bab 2 halaman 7. Pada gambar dibawah ini merupakan beberapa desain interface untuk halaman user dan juga admin.

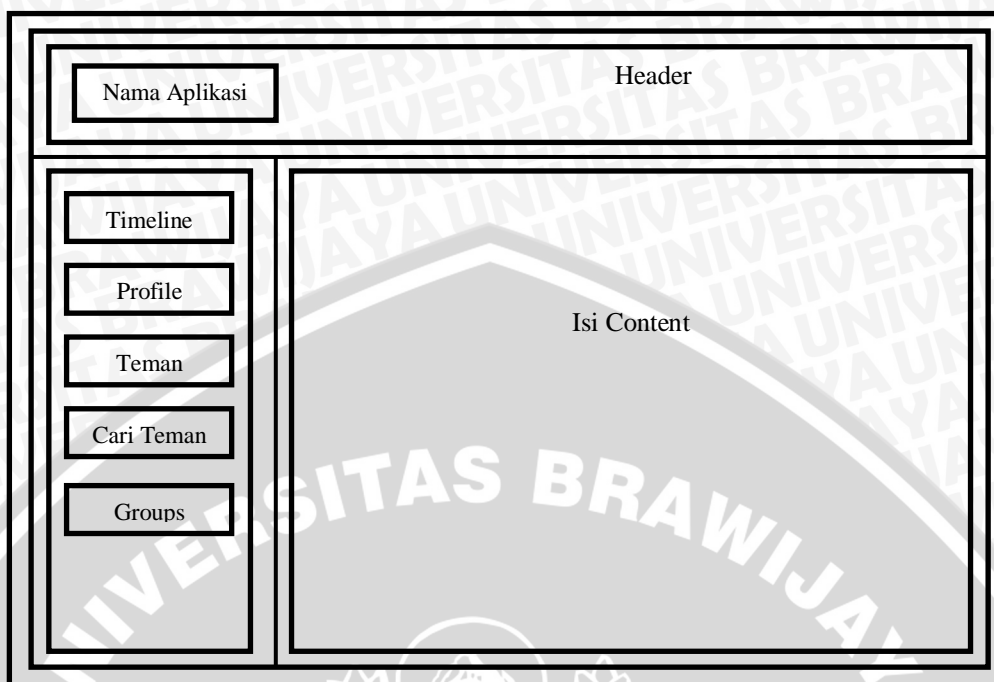
A. Halaman User

a) Pada gambar 3.41 merupakan desain interface untuk tampilan utama dari user



Gambar 3.41 Desain interface tampilan utama user

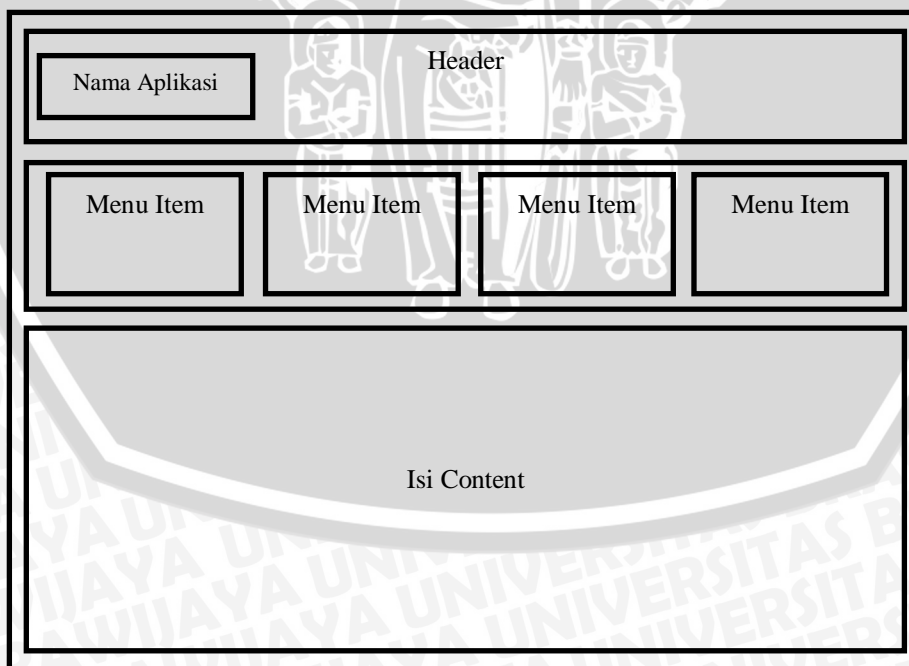
b) Pada gambar 3.42 merupakan desain interface untuk halaman home user



Gambar 3.42 Desain interface tampilan home user

B. Halaman Admin

a) Pada gambar 3.43 merupakan desain interface untuk tampilan utama admin



Gambar 3.43 Desain interface tampilan utama admin

BAB IV IMPLEMENTASI

4.1 Spesifikasi Sistem

Spesifikasi sistem terdiri dari spesifikasi kebutuhan perangkat lunak dan perangkat keras. Berikut ini merupakan spesifikasi sistem untuk aplikasi *expert locator*.

4.1.1 Spesifikasi Kebutuhan Perangkat Lunak

Pada tabel 4.1 berisikan tentang spesifikasi perangkat lunak yang dibutuhkan untuk Aplikasi *Expert Locator*.

Tabel 4.1 Spesifikasi Kebutuhan Perangkat Lunak

Perangkat Lunak	Keterangan
XAMPP	Salah satu paket <i>software web server</i> yang terdiri dari Apache, MySQL, PHP dan phpMyAdmin.
CodeIgniter versi 2	Sebagai framework pembuatan web
Adobe Dreamweaver CS5	Tools untuk pembuatan source code aplikasi
Web Browser (Mozilla Firefox , Google Chrome, dan Internet Explorer)	Tools untuk memungkinkan user melihat dan berinteraksi dengan halaman web.
Microsoft Excel 2007	Fitur antarmuka yang intuitif dan mampu melakukan kalkulasi serta pembuatan grafik secara istimewa

4.1.2 Spesifikasi Kebutuhan Perangkat Keras

Sedangkan perangkat keras yang dibutuhkan untuk Aplikasi *Expert Locator*, yaitu satu buah perangkat komputer yang digunakan pada *client*. Spesifikasi perangkat keras yang dibutuhkan di *client* di tunjukan pada tabel 4.2.

Tabel 4.2 Spesifikasi Kebutuhan Perangkat Keras

Nama Komponen	Spesifikasi
Processor	Intel Core i5-3210M 2.5GHz

Memory (RAM)	4 GB 1600 MHz DDR3
Harddisk	750GB
VGA Card	NVIDIA GeForce GT 635M dengan 2GB DDR3

4.2 Batasan – Batasan Implementasi

Beberapa batasan dalam mengimplementasikan Aplikasi *Expert Locator* adalah sebagai berikut:

1. *Input* yang diterima oleh sistem yaitu berupa data karyawan, data unit kerja, data pendidikan, data pengalaman kerja, data training, data sertifikasi, data user, data akses, data group, data pesan, data status, dan data album.
2. *Output* yang diterima pengguna berupa data timeline, data profil, data teman, dan data group.
3. *Platform* yang digunakan dalam pembuatan sistem *expert locator* ini adalah PHP dengan framework CodeIgniter.
4. Keseluruhan penyimpanan data disimpan didalam *Database MySQL*.
5. Sistem bersifat dinamis, yaitu menyediakan fasilitas *insert*, *update*, dan *delete*.

4.3 Implementasi Basis Data

Implementasi perancangan basis data pada aplikasi *expert locator* ini menggunakan *Data Definition Language (DDL)*. DDL merupakan struktur basis data yang menggambarkan desain basis data secara keseluruhan. Berikut dibawah ini merupakan pengimplementasiannya :

4.3.1 DDL untuk Membuat Basis Data kms

DDL yang digunakan untuk membuat basis data kms adalah sebagai berikut:

```
create database kms;
```

4.3.2 DDL untuk Membuat Tabel karyawan

DDL yang digunakan untuk membuat tabel karyawan adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `karyawan` (
  `NIK` varchar(50) NOT NULL,
  `KODE_UNIT_KERJA` varchar(50) NOT NULL,
  `NAMA` varchar(75) DEFAULT NULL,
```



```

`POSCODE` varchar(50) DEFAULT NULL,
`JABATAN` varchar(75) DEFAULT NULL,
`POSTITLE` varchar(50) DEFAULT NULL,
`UNITKERJA` varchar(50) DEFAULT NULL,
`WORKLOC` varchar(50) DEFAULT NULL,
PRIMARY KEY (`NIK`),
KEY `Karyawan_FKIndex1` (`KODE_UNIT_KERJA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

4.3.3 DDL untuk Membuat Tabel pendidikan

DDL yang digunakan untuk membuat tabel pendidikan adalah sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `pendidikan` (
  `ID_pendidikan` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NIK` varchar(50) NOT NULL,
  `KODE_UNIT_KERJA` varchar(50) NOT NULL,
  `EDUCATION_LEVEL_NO` varchar(50) DEFAULT NULL,
  `EDUCATION_LEVEL_NAME` varchar(50) DEFAULT NULL,
  `EDUCATION_FIELD_NO` varchar(50) DEFAULT NULL,
  `EDUCATION_FIELD_NAME` varchar(50) DEFAULT NULL,
  `LOCATION` varchar(50) DEFAULT NULL,
  `START_YEAR` varchar(50) DEFAULT NULL,
  `END_YEAR` varchar(50) DEFAULT NULL,
  `G_P_A` varchar(50) DEFAULT NULL,
  `ACHIEVEMENT` varchar(50) DEFAULT NULL,
  `REMARK` varchar(50) DEFAULT NULL,
  `EDU_INSTITUTION_SEQ_NO` varchar(50) DEFAULT NULL,
  `EDU_INSTITUTION_NAME` varchar(50) DEFAULT NULL,
  `EDUCATION_LVL_RECEIVED` varchar(50) DEFAULT NULL,
  `PREFERED_EDUCATION` varchar(50) DEFAULT NULL,
  `ROWVERSION` varchar(50) DEFAULT NULL,
  `ANIK_BARU` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`ID_pendidikan`, `NIK`, `KODE_UNIT_KERJA`),
  KEY `Pendidikan_FKIndex1` (`NIK`, `KODE_UNIT_KERJA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

4.3.4 DDL untuk Membuat Tabel pengalaman_kerja

DDL yang digunakan untuk membuat tabel pengalaman_kerja adalah sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `pengalaman_kerja` (
  `ID_pengalaman_kerja` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NIK` varchar(50) NOT NULL,
  `KODE_UNIT_KERJA` varchar(50) NOT NULL,
  `NO_SK` varchar(50) DEFAULT NULL,
  `TGL_SK` date DEFAULT NULL,
  `MA_TMT` varchar(50) DEFAULT NULL,

```

```

`MKT_GOL` varchar(50) DEFAULT NULL,
`MKB_GOL` varchar(50) DEFAULT NULL,
`KODE_UPDATE` varchar(50) DEFAULT NULL,
`NM_JBT` varchar(50) DEFAULT NULL,
`NM_PL` varchar(50) DEFAULT NULL,
`NM_RU` varchar(50) DEFAULT NULL,
`NM_SI` varchar(50) DEFAULT NULL,
`NM_BAG` varchar(50) DEFAULT NULL,
`NM_DEP` varchar(50) DEFAULT NULL,
`NM_KOM` varchar(50) DEFAULT NULL,
`NM_DIR` varchar(50) DEFAULT NULL,
`KETERANGAN` varchar(50) DEFAULT NULL,
`TIMESTAMP` datetime DEFAULT NULL,
PRIMARY KEY (`ID_pengalaman_kerja`,`NIK`,`KODE_UNIT_KERJA`),
KEY `Pengalaman_Kerja_FKIndex1` (`NIK`,`KODE_UNIT_KERJA`),
KEY `Pengalaman_Kerja_FKIndex2` (`KODE_UNIT_KERJA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

4.3.5 DDL untuk Membuat Tabel training

DDL yang digunakan untuk membuat tabel training adalah sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `training` (
  `ID_training` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NIK` varchar(50) NOT NULL,
  `KODE_UNIT_KERJA` varchar(50) NOT NULL,
  `TRAINING_EVENT_ID` varchar(50) DEFAULT NULL,
  `COURSE_ID` varchar(50) DEFAULT NULL,
  `NAMA_TRAINING` varchar(50) DEFAULT NULL,
  `TOPIK` varchar(50) DEFAULT NULL,
  `DURASI` varchar(50) DEFAULT NULL,
  `START_DATE` date DEFAULT NULL,
  `END_DATE` date DEFAULT NULL,
  `TEMPAT` varchar(50) DEFAULT NULL,
  `PENYELENGGARA` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`ID_training`,`NIK`,`KODE_UNIT_KERJA`),
  KEY `Training_FKIndex1` (`NIK`,`KODE_UNIT_KERJA`),
  KEY `Training_FKIndex2` (`KODE_UNIT_KERJA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=38 ;

```

4.3.6 DDL untuk Membuat Tabel sertifikasi

DDL yang digunakan untuk membuat tabel sertifikasi adalah sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `sertifikasi` (
  `ID_sertifikasi` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NIK` varchar(50) NOT NULL,
  `KODE_UNIT_KERJA` varchar(50) NOT NULL,

```



```

`TRAINING_EVENT_ID` varchar(50) DEFAULT NULL,
`COURSE_ID` varchar(50) DEFAULT NULL,
`NAMA_TRAINING` varchar(50) DEFAULT NULL,
`TOPIK` varchar(50) DEFAULT NULL,
`DURASI` varchar(50) DEFAULT NULL,
`START_DATE` date DEFAULT NULL,
`END_DATE` date DEFAULT NULL,
`TEMPAT` varchar(50) DEFAULT NULL,
`PENYELENGGARA` varchar(50) DEFAULT NULL,
PRIMARY KEY (`ID_sertifikasi`, `NIK`, `KODE_UNIT_KERJA`),
KEY `Training_FKIndex1` (`NIK`, `KODE_UNIT_KERJA`),
KEY `Training_FKIndex2` (`KODE_UNIT_KERJA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=37 ;

```

4.3.7 DDL untuk Membuat Tabel m_user

DDL yang digunakan untuk membuat tabel m_user adalah sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `m_user` (
  `ID_user` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `NIK` varchar(50) NOT NULL,
  `KODE_UNIT_KERJA` int(10) unsigned NOT NULL,
  `ID_GROUP` int(10) unsigned NOT NULL,
  `ID_AKSES` int(10) unsigned NOT NULL,
  `USERNAME` varchar(50) DEFAULT NULL,
  `PASSWORD_USER` varchar(50) DEFAULT NULL,
  `GAMBAR_PROFIL` varchar(200) DEFAULT NULL,
  `GAMBAR_PROFIL_KECIL` varchar(200) DEFAULT NULL,
  `GAMBAR_COVER_FOTO` varchar(200) NOT NULL,
  `MINAT` varchar(200) DEFAULT NULL,
  `PERMINTAAN_TEMAN` int(10) unsigned DEFAULT NULL,
  `IS_AKTIF` int(10) unsigned DEFAULT NULL,
  PRIMARY KEY (`ID_user`, `NIK`, `KODE_UNIT_KERJA`, `ID_GROUP`, `ID_AKSES`),
  KEY `User_2_FKIndex1` (`NIK`, `KODE_UNIT_KERJA`),
  KEY `M_USER_FKIndex2` (`ID_GROUP`),
  KEY `M_USER_FKIndex3` (`ID_AKSES`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

```

4.3.8 DDL untuk Membuat Tabel m_akses

DDL yang digunakan untuk membuat tabel m_akses adalah sebagai berikut:

```

CREATE TABLE IF NOT EXISTS `m_akses` (
  `ID_AKSES` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `LEVEL_AKSES` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`ID_AKSES`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;

```


4.3.9 DDL untuk Membuat Tabel m_unit_kerja

DDL yang digunakan untuk membuat tabel m_unit_kerja adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `m_unit_kerja` (
  `KODE_UNIT_KERJA` varchar(15) NOT NULL,
  `CONTENT_UNIT_KERJA` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`KODE_UNIT_KERJA`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.3.10 DDL untuk Membuat Tabel m_group

DDL yang digunakan untuk membuat tabel m_group adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `m_group` (
  `ID_GROUP` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `PARENT_ID_GROUP` int(10) unsigned DEFAULT NULL,
  `LEVEL_GROUP` int(10) unsigned DEFAULT NULL,
  `NAMA_GROUP` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`ID_GROUP`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=44 ;
```

4.3.11 DDL untuk Membuat Tabel m_group_status

DDL yang digunakan untuk membuat tabel m_group_status adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `m_group_status` (
  `ID_Group_Status` int(11) NOT NULL AUTO_INCREMENT,
  `ID_GROUP` int(11) NOT NULL,
  `ID_User` int(11) NOT NULL,
  `User_status` varchar(200) NOT NULL,
  `Create_Date` datetime NOT NULL,
  `Last_Date` datetime NOT NULL,
  PRIMARY KEY (`ID_Group_Status`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

4.3.12 DDL untuk Membuat Tabel m_group_komentar

DDL yang digunakan untuk membuat tabel m_group_komentar adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `m_group_komentar` (
  `ID_Group_Komentar` int(11) NOT NULL AUTO_INCREMENT,
  `ID_Group_Status` int(11) NOT NULL,
  `ID_user` int(11) NOT NULL,
  `KOMENTAR` varchar(200) NOT NULL,
  `Create_Date` datetime NOT NULL,
  `Last_Date` datetime NOT NULL,
  PRIMARY KEY (`ID_Group_Komentar`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

4.3.13 DDL untuk Membuat Tabel m_status

DDL yang digunakan untuk membuat tabel m_status adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `m_status` (
  `ID_Status` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ID_user` int(10) unsigned NOT NULL,
  `User_status` varchar(255) DEFAULT NULL,
  `Create_Date` timestamp NULL DEFAULT NULL,
  `Last_Date` datetime DEFAULT NULL,
  PRIMARY KEY (`ID_Status`),
  KEY `M_STATUS_FKIndex1` (`ID_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=16 ;
```

4.3.14 DDL untuk Membuat Tabel m_komentar

DDL yang digunakan untuk membuat tabel m_komentar adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `m_komentar` (
  `ID_Komentar` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ID_Status` int(10) unsigned NOT NULL,
  `ID_user` int(10) NOT NULL,
  `KOMENTAR` varchar(50) DEFAULT NULL,
  `Create_Date` timestamp NULL DEFAULT NULL,
  `Last_Date` datetime DEFAULT NULL,
  PRIMARY KEY (`ID_Komentar`,`ID_Status`),
  KEY `M_KOMENTAR_FKIndex1` (`ID_Status`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=35 ;
```

4.3.15 DDL untuk Membuat Tabel notifikasi

DDL yang digunakan untuk membuat tabel notifikasi adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `notifikasi` (
  `ID_Notifikasi` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ID_User` int(11) NOT NULL,
  `ID_Status` int(11) NOT NULL,
  `Notifikasi` varchar(200) DEFAULT NULL,
  `Tgl` timestamp NULL DEFAULT NULL,
  `ID_Penerima` int(10) unsigned DEFAULT NULL,
  `Penerima` varchar(10) DEFAULT NULL,
  `url` varchar(200) NOT NULL,
  `status` int(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`ID_Notifikasi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

4.3.16 DDL untuk Membuat Tabel permintaan_group

DDL yang digunakan untuk membuat tabel permintaan_group adalah sebagai berikut:


```
CREATE TABLE IF NOT EXISTS `permintaan_group` (
  `ID_permintaan` int(11) NOT NULL AUTO_INCREMENT,
  `ID_GROUP` int(11) NOT NULL,
  `ID_user` int(11) NOT NULL,
  `Create_Date` datetime NOT NULL,
  `Last_Date` datetime NOT NULL,
  PRIMARY KEY (`ID_permintaan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

4.3.17 DDL untuk Membuat Tabel permintaan_teman

DDL yang digunakan untuk membuat tabel permintaan_teman adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `permintaan_teman` (
  `ID_user` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ID_User_Request` varchar(50) NOT NULL,
  `ID_User_Received` varchar(50) DEFAULT NULL,
  `Create_Date` timestamp NULL DEFAULT NULL,
  `Last_Update` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`ID_user`),
  KEY `Permintaan_Teman_FKIndex1` (`ID_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

4.3.18 DDL untuk Membuat Tabel pesan

DDL yang digunakan untuk membuat tabel pesan adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `pesan` (
  `ID_Pesan` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ID_user` int(10) unsigned NOT NULL,
  `Pesan` varchar(200) DEFAULT NULL,
  `Tgl` timestamp NULL DEFAULT NULL,
  `Subject` varchar(100) DEFAULT NULL,
  `ID_Pengirim` int(10) unsigned DEFAULT NULL,
  `ID_Penerima` int(10) unsigned DEFAULT NULL,
  `status` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`ID_Pesan`),
  KEY `Pesan_FKIndex1` (`ID_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

4.3.19 DDL untuk Membuat Tabel user_album

DDL yang digunakan untuk membuat tabel user_album adalah sebagai berikut:

```
CREATE TABLE IF NOT EXISTS `user_album` (
  `ID_User_Album` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `ID_user` int(10) unsigned NOT NULL,
  `Nama_Album` varchar(30) NOT NULL,
```



```

`Nama_Foto` varchar(50) DEFAULT NULL,
`Nama_Foto_Kecil` varchar(50) NOT NULL,
`Status` tinyint(4) NOT NULL,
`Created_Date` datetime NOT NULL,
`Last_Updated` datetime NOT NULL,
PRIMARY KEY (`ID_User_Album`,`ID_user`),
KEY `USER_ALBUM_FKIndex1` (`ID_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;

```

4.4 Implementasi Antarmuka

4.4.1 Implementasi Antarmuka Halaman Utama

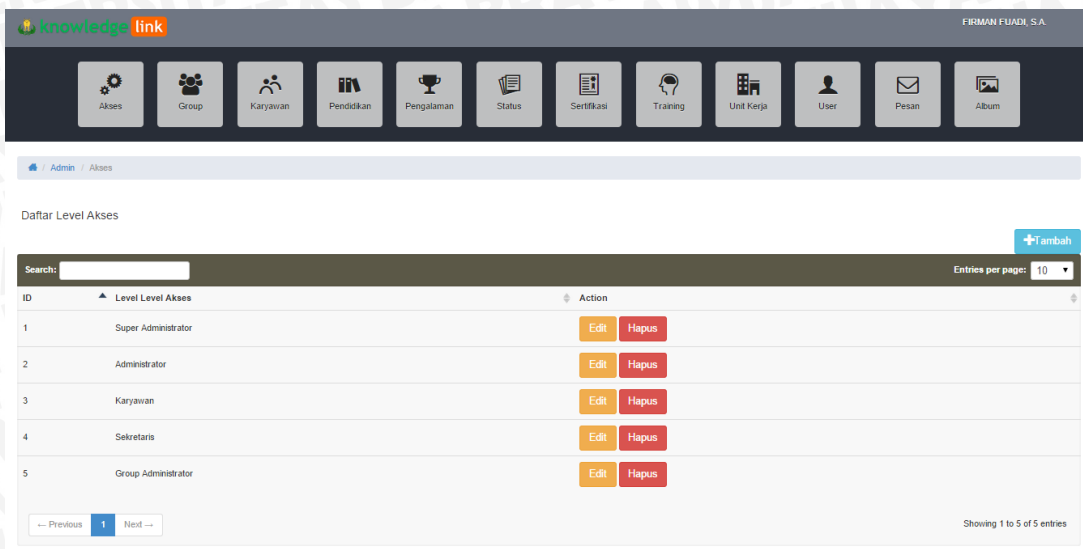
Pada halaman ini admin dan juga user biasa dapat menginputkan username dan password untuk bisa masuk ke dalam masing-masing fitur sesuai dengan hak aksesnya. Seperti yang ditunjukkan pada gambar 4.1.



Gambar 4.1 Halaman Utama

4.4.2 Implementasi Antarmuka Halaman Administrator

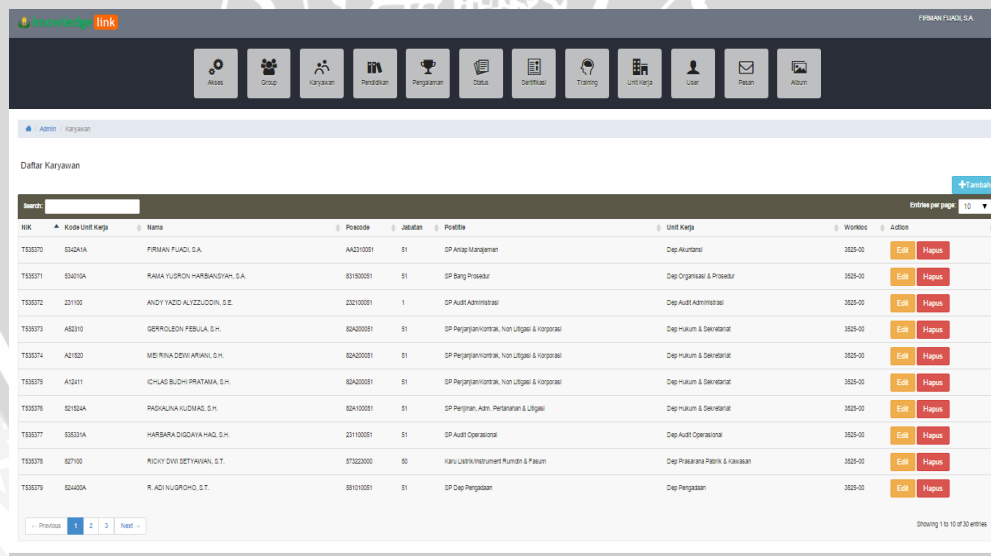
Pada halaman ini terdapat beberapa modul – modul yang dikelola oleh admin yang diantaranya modul karyawan, pendidikan, pengalaman, training, sertifikasi, unit kerja, user, akses, group, status, pesan dan juga album. Seperti yang ditunjukkan pada gambar 4.2.



Gambar 4.2 Halaman Administrator

4.4.2.1 Implementasi Antarmuka Modul Karyawan

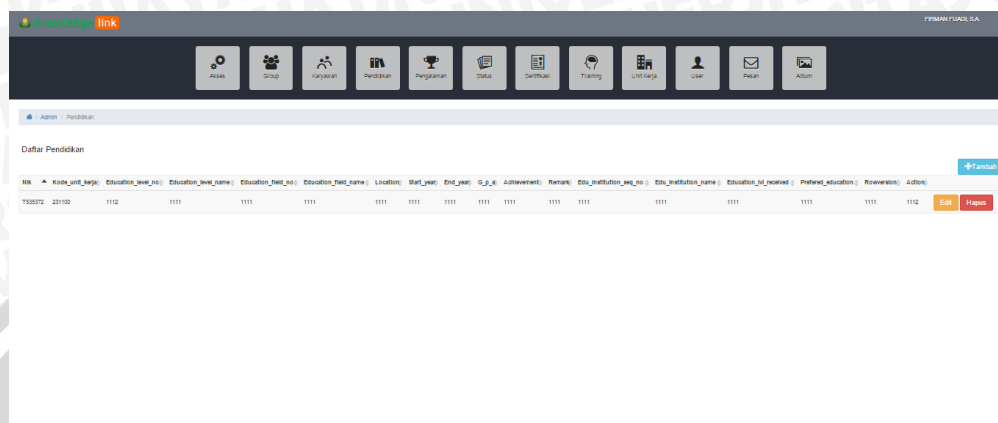
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data karyawan (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.3.



Gambar 4.3 Halaman Modul Karyawan

4.4.2.2 Implementasi Antarmuka Modul Pendidikan

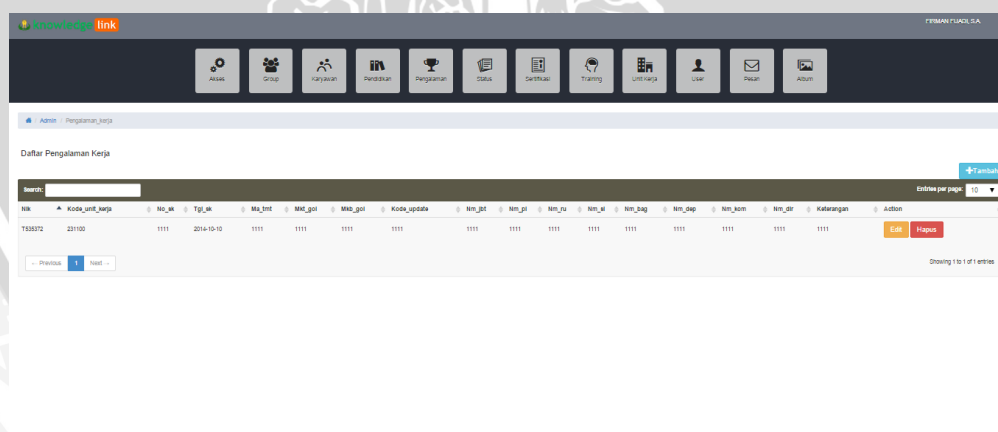
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data pendidikan (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.4.



Gambar 4.4 Halaman Modul Pendidikan

4.4.2.3 Implementasi Antarmuka Modul Pengalaman

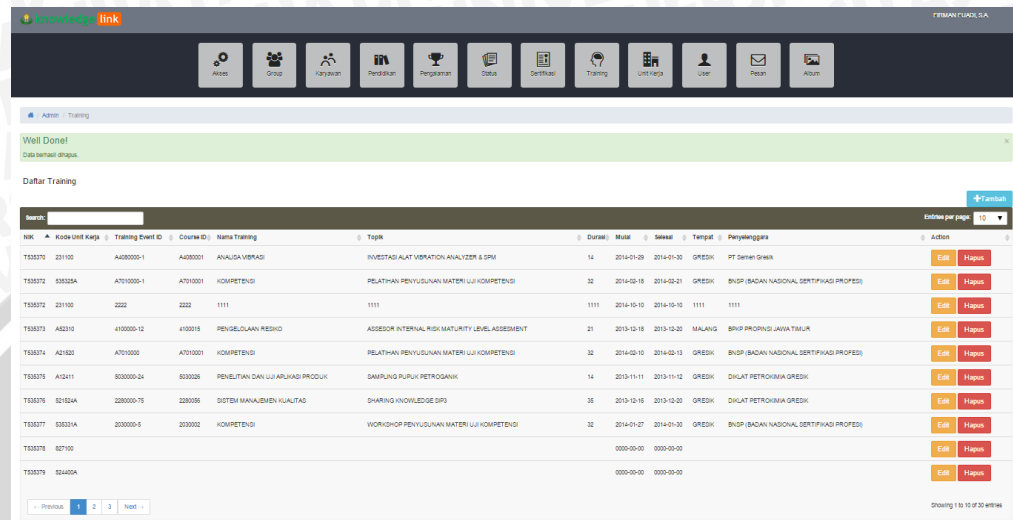
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data pengalaman (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.5.



Gambar 4.5 Halaman Modul Pengalaman

4.4.2.4 Implementasi Antarmuka Modul Training

Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data training (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.6.

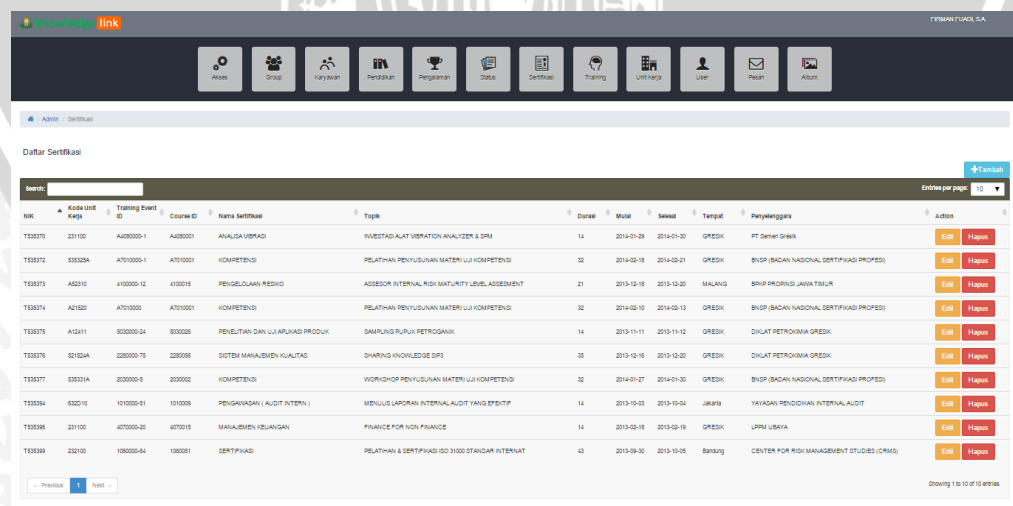


NIK	Kode Unit Kerja	Training Event ID	Course ID	Nama Training	Topik	Durasi	Mulai	Selesai	Tempat	Penyelenggara	Action
T53370	23100	A620000-1	A620001	ANALISA VIBRASI	INVESTASI ALAT VIBRATION ANALYZER & SPM	14	2014-01-29	2014-01-30	GRESIK	PT Seheri Gresik	Edit Hapus
T53372	53325A	A7010000-1	A7010001	KOMPETENSI	RELATIHAN PENYUSUNAN MATERI UJI KOMPETENSI	32	2014-02-18	2014-02-21	GRESIK	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53372	23100	2222	2222	1111	1111	1111	2014-10-10	2014-10-10	1111	1111	Edit Hapus
T53373	AS310	410000-12	4100016	PENGELOLAAN RESIKO	ASSESSOR INTERNAL RISK MATURITY LEVEL ASSESSMENT	21	2013-12-18	2013-12-20	MALANG	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53374	A2152	A7010000	A7010001	KOMPETENSI	RELATIHAN PENYUSUNAN MATERI UJI KOMPETENSI	32	2014-02-10	2014-02-13	GRESIK	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53375	A12411	802000-24	8020026	PENELITIAN DAN UJI APRIKAD PRODUK	SAMPLING PURUK PETROKAWAN	14	2013-11-11	2013-11-12	GRESIK	DIKLAT PETROKAWA GRESIK	Edit Hapus
T53376	821524A	220000-75	2200006	SISTEM MANAJEMEN KUALITAS	SHARING KNOWLEDGE SPI	35	2013-12-16	2013-12-20	GRESIK	DIKLAT PETROKAWA GRESIK	Edit Hapus
T53377	53321A	203000-6	2030002	KOMPETENSI	WORKSHOP PENYUSUNAN MATERI UJI KOMPETENSI	32	2014-01-27	2014-01-30	GRESIK	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53379	827100						0000-00-00	0000-00-00			Edit Hapus
T53379	82405A						0000-00-00	0000-00-00			Edit Hapus

Gambar 4.6 Halaman Modul Training

4.4.2.5 Implementasi Antarmuka Modul Sertifikasi

Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data sertifikasi (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.7.

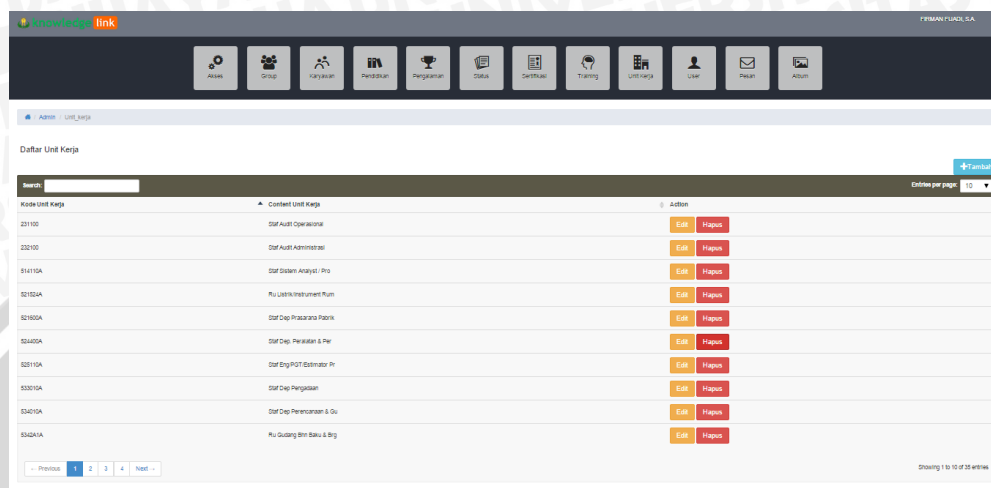


NIK	Kode Unit Kerja	Training Event ID	Course ID	Nama Sertifikasi	Topik	Durasi	Mulai	Selesai	Tempat	Penyelenggara	Action
T53370	23100	A620000-1	A620001	ANALISA VIBRASI	INVESTASI ALAT VIBRATION ANALYZER & SPM	14	2014-01-29	2014-01-30	GRESIK	PT Seheri Gresik	Edit Hapus
T53372	53325A	A7010000-1	A7010001	KOMPETENSI	RELATIHAN PENYUSUNAN MATERI UJI KOMPETENSI	32	2014-02-18	2014-02-21	GRESIK	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53373	AS310	410000-12	4100016	PENGELOLAAN RESIKO	ASSESSOR INTERNAL RISK MATURITY LEVEL ASSESSMENT	21	2013-12-18	2013-12-20	MALANG	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53374	A2152	A7010000	A7010001	KOMPETENSI	RELATIHAN PENYUSUNAN MATERI UJI KOMPETENSI	32	2014-02-10	2014-02-13	GRESIK	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53375	A12411	802000-24	8020026	PENELITIAN DAN UJI APRIKAD PRODUK	SAMPLING PURUK PETROKAWAN	14	2013-11-11	2013-11-12	GRESIK	DIKLAT PETROKAWA GRESIK	Edit Hapus
T53376	821524A	220000-75	2200006	SISTEM MANAJEMEN KUALITAS	SHARING KNOWLEDGE SPI	35	2013-12-16	2013-12-20	GRESIK	DIKLAT PETROKAWA GRESIK	Edit Hapus
T53377	53321A	203000-6	2030002	KOMPETENSI	WORKSHOP PENYUSUNAN MATERI UJI KOMPETENSI	32	2014-01-27	2014-01-30	GRESIK	BNP (BADAN NASIONAL SERTIFIKASI PROFESI)	Edit Hapus
T53384	632010	1010000-81	1010008	PENGAWASAN (AUDIT INTERN)	MENULIS LAPORAN INTERNAL AUDIT YANG EFDITIF	14	2013-10-02	2013-10-04	JAKARTA	YAKASAN PENCIDHAK INTERNAL AUDIT	Edit Hapus
T53386	23100	4070000-20	4070019	MANAJEMEN KEUANGAN	FINANCE FOR NON FINANCE	14	2013-08-18	2013-08-19	GRESIK	LRPI USANA	Edit Hapus
T53389	232100	1060000-44	1060021	SERTIFIKASI	RELATIHAN & SERTIFIKASI ISO 31000 STANDAR INTERNAT	43	2013-08-30	2013-10-05	Bandung	CENTER FOR RISK MANAGEMENT STUDIES (CRMS)	Edit Hapus

Gambar 4.7 Halaman Modul Sertifikasi

4.4.2.6 Implementasi Antarmuka Modul Unit Kerja

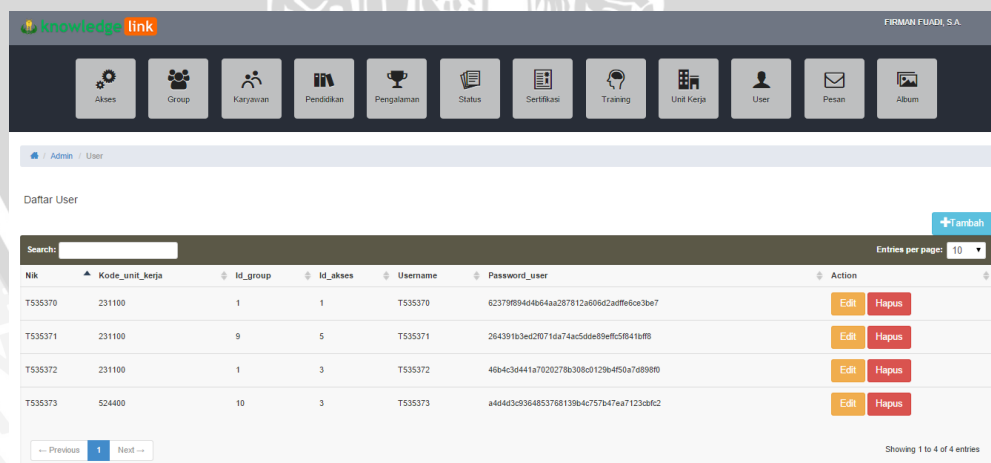
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data unit kerja (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.8.



Gambar 4.8 Halaman Modul Unit Kerja

4.4.2.7 Implementasi Antarmuka Modul User

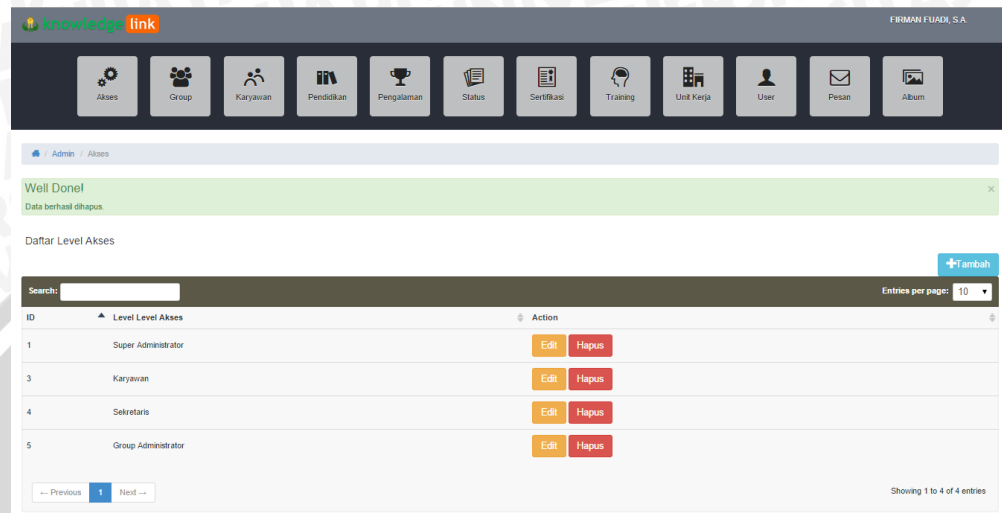
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data user (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.9.



Gambar 4.9 Halaman Modul User

4.4.2.8 Implementasi Antarmuka Modul Akses

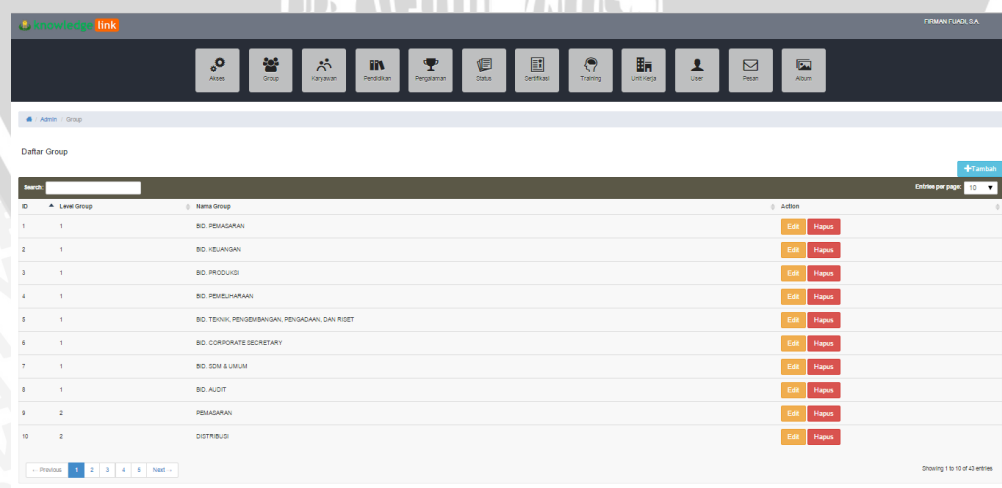
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data akses (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.10.



Gambar 4.10 Halaman Modul Akses

4.4.2.9 Implementasi Antarmuka Modul Group

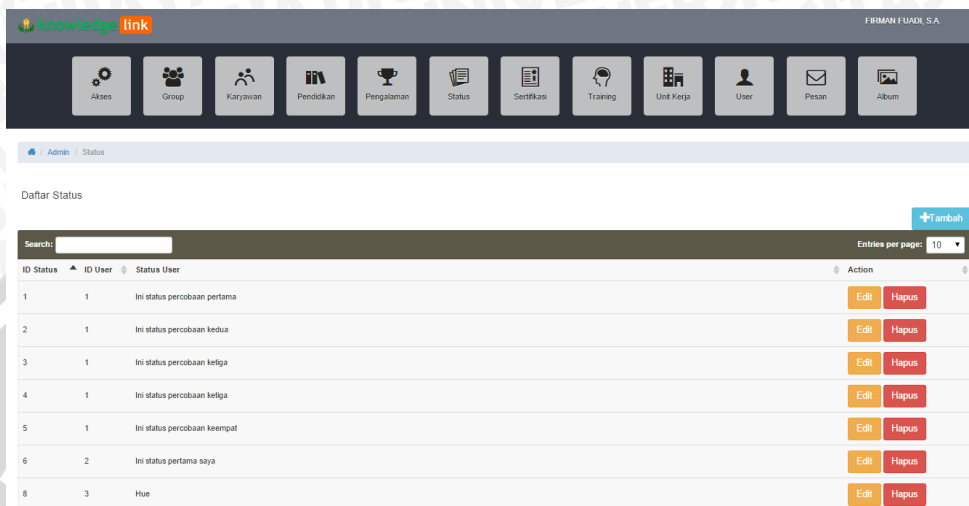
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data group (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.11.



Gambar 4.11 Halaman Modul Group

4.4.2.10 Implementasi Antarmuka Modul Status

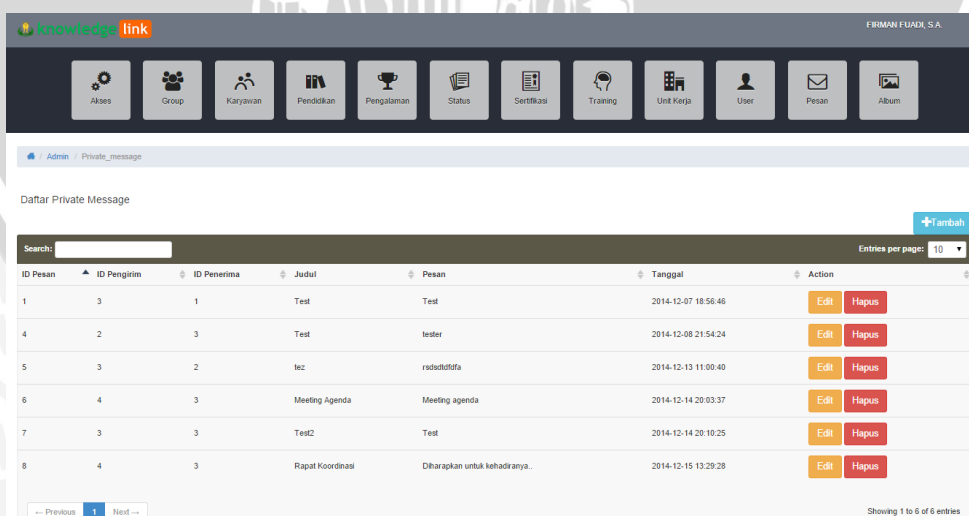
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data status (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.12.



Gambar 4.12 Halaman Modul Status

4.4.2.11 Implementasi Antarmuka Modul Pesan

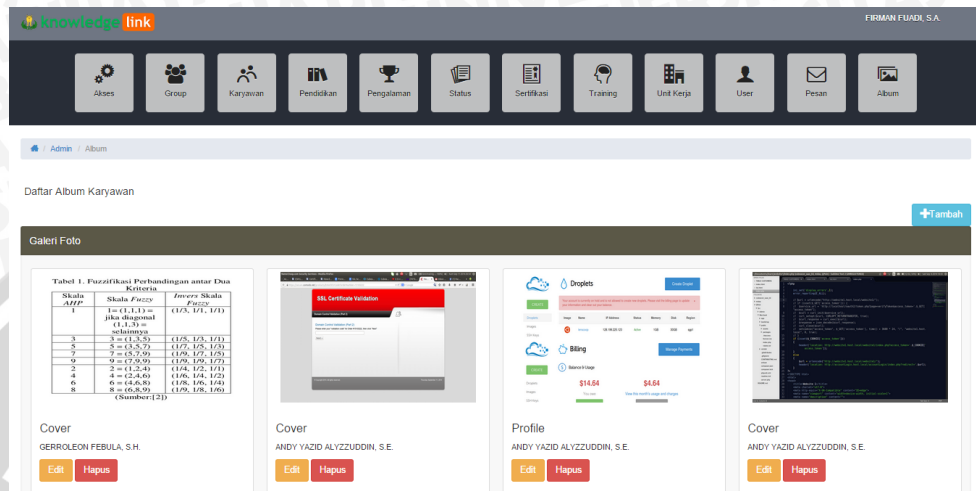
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data pesan (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.13.



Gambar 4.13 Halaman Modul Pesan

4.4.2.12 Implementasi Antarmuka Modul Album

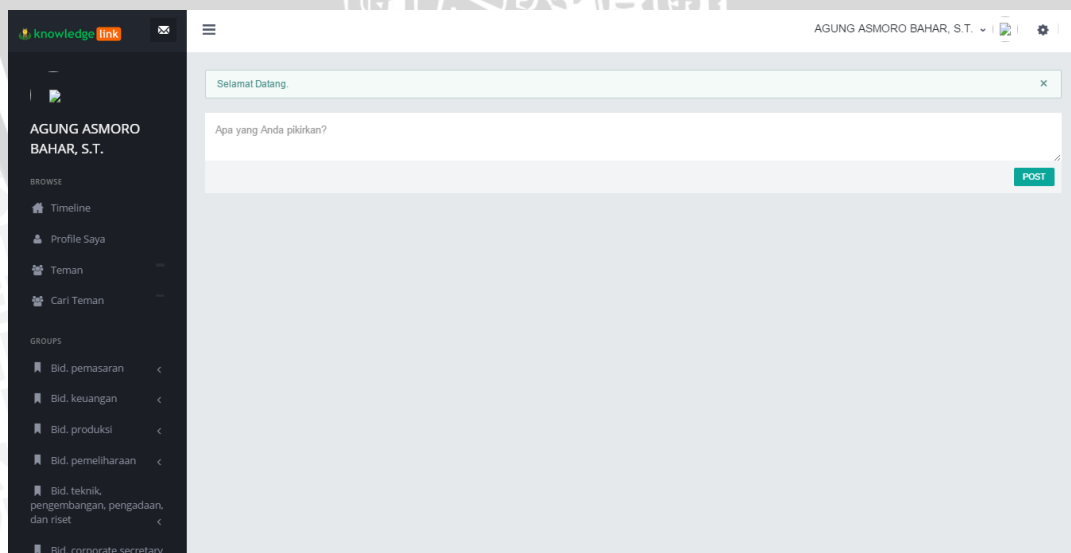
Pada halaman ini admin dapat melakukan keseluruhan proses pengolahan data album (*insert, update, delete*). Seperti yang ditunjukkan pada gambar 4.14.



Gambar 4.14 Halaman Modul Album

4.4.3 Implementasi Antarmuka Halaman User

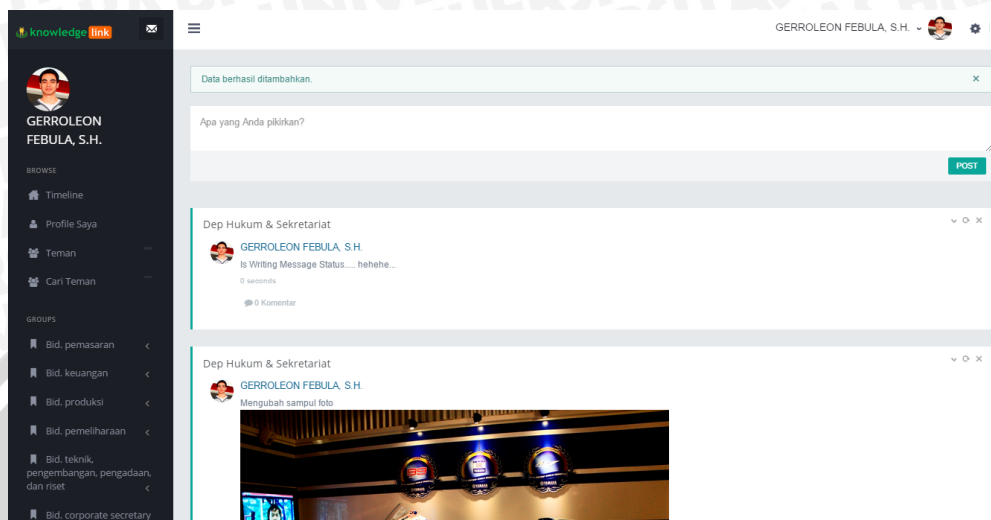
Berikut ini merupakan tampilan antarmuka daripada halaman user. Seperti yang ditunjukkan pada gambar 4.15.



Gambar 4.15 Halaman User

4.4.3.1 Implementasi Antarmuka Halaman Timeline

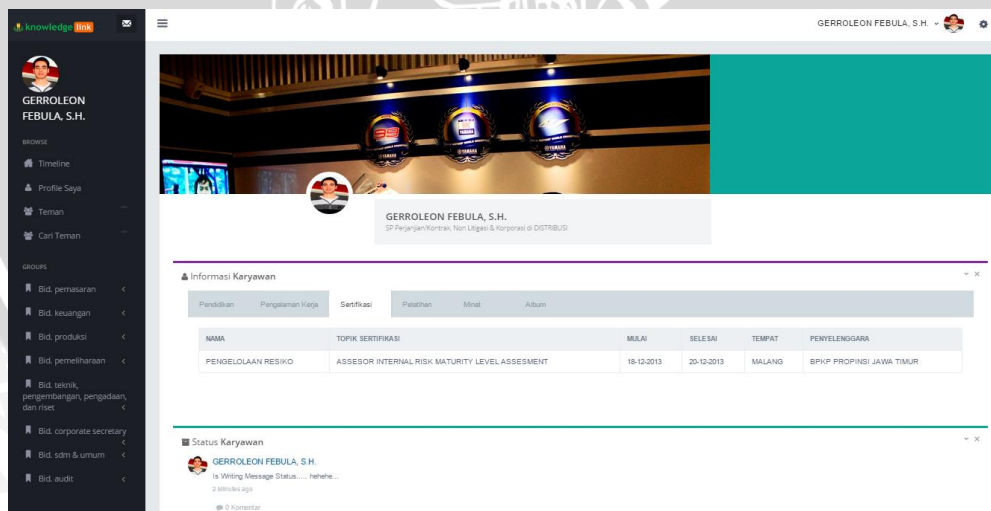
Pada halaman ini user dapat menuliskan status mereka. Seperti yang ditunjukkan pada gambar 4.16.



Gambar 4.16 Halaman Timeline

4.4.3.2 Implementasi Antarmuka Halaman Profile Saya

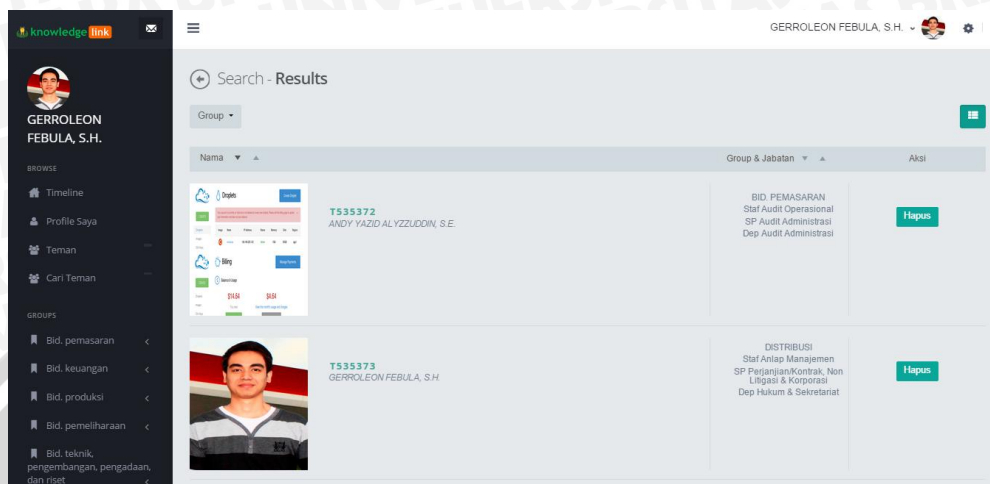
Pada halaman ini user dapat melihat profil mereka. Seperti yang ditunjukkan pada gambar 4.17.



Gambar 4.17 Halaman Profile Saya

4.4.3.3 Implementasi Antarmuka Halaman Teman

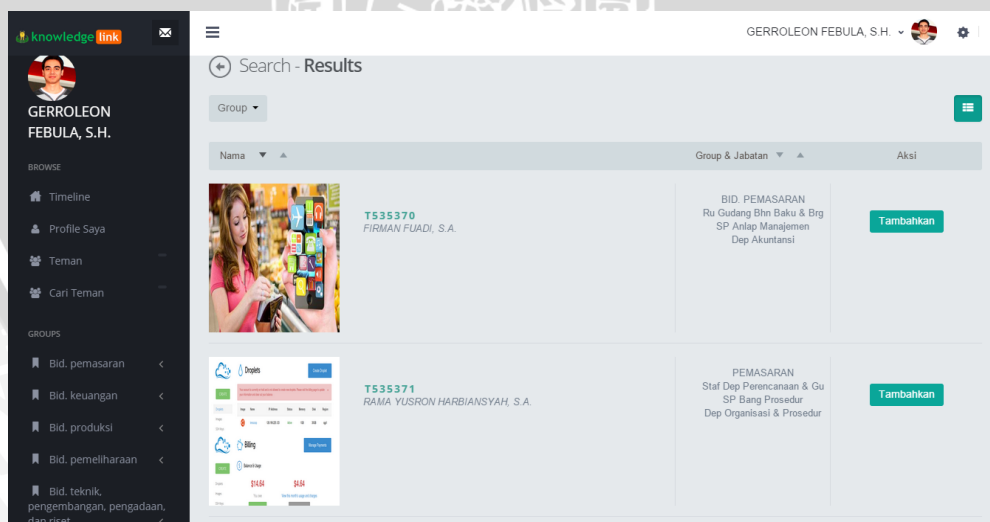
Pada halaman ini user dapat melihat karyawan yang menjadi teman mereka. Seperti yang ditunjukkan pada gambar 4.18.



Gambar 4.18 Halaman Teman

4.4.3.4 Implementasi Antarmuka Halaman Cari Teman

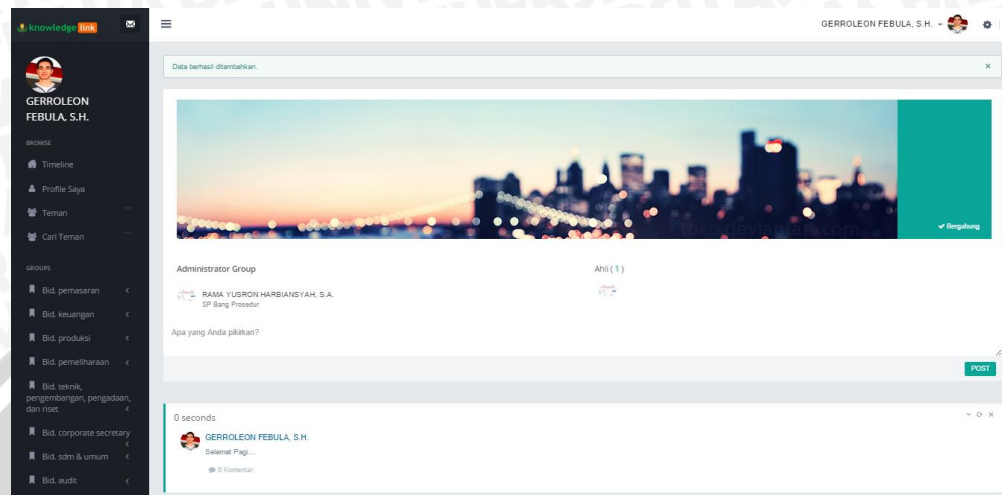
Pada halaman ini user dapat mencari karyawan untuk kemudian bisa di tambahkan menjadi teman mereka. Seperti yang ditunjukkan pada gambar 4.19.



Gambar 4.19 Halaman Cari Teman

4.4.3.5 Implementasi Antarmuka Halaman Group

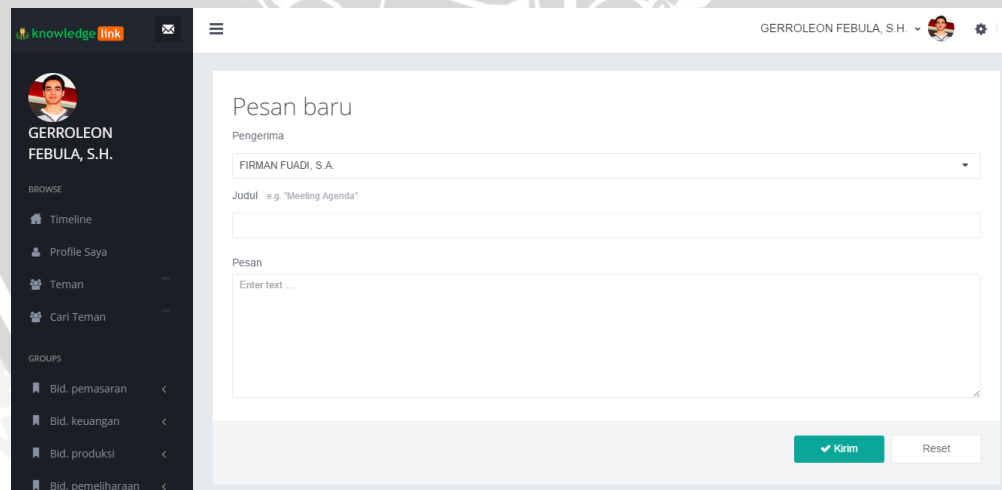
Pada halaman ini user dapat bergabung dengan group, sesuai yang diinginkan. Seperti yang ditunjukkan pada gambar 4.20.



Gambar 4.20 Halaman Group

4.4.3.6 Implementasi Antarmuka Halaman Pesan

Pada halaman ini user dapat melakukan pengiriman pesan secara personal kepada setiap karyawan. Seperti yang ditunjukkan pada gambar 4.21.



Gambar 4.21 Halaman Pesan

4.4.3.7 Implementasi Antarmuka Halaman Akun Saya

Pada halaman ini user dapat melakukan edit profil mereka. Seperti yang ditunjukkan pada gambar 4.22.

The screenshot shows the 'My Profile' page for user GERROLEON FEBULA, S.H. The page is divided into two main sections: 'Profile Saya' and 'Foto Saya'. The 'Profile Saya' section contains a 'Biodata' form with fields for name (GERROLEON FEBULA, S.H.), ID (T53877), and position (231100 (Staff Audit Operasional)). Below this is a 'Jabatan' section with fields for ID (82A200091), phone number (91), and address (SP Pekerjaan/Kontrak, Non Litigasi & Korporasi, Dep Hukum & Sekretariat, 3525-00). The 'Foto Saya' section contains a 'Profile' form with a 'Choose File' button and a 'Cover' form with a 'Choose File' button and a note '* Ukuran 1287 x 452 pixels'. Both sections have 'Simpan' and 'Reset' buttons.

Gambar 4.22 Halaman Akun Saya

BAB V PENGUJIAN

5.1 Pengujian

Pengujian Aplikasi *Expert Locator* ini dilakukan dengan dua metode, yaitu dengan metode *functional testing* dan *usability testing*. *Functional testing* merupakan metode pengujian berdasarkan kasus yang diberikan pada suatu komponen yang akan diuji. Sedangkan *usability testing* merupakan metode pengujian yang digunakan untuk mengevaluasi kemudahan penggunaan aplikasi yang dibuat dari sisi penggunanya.

5.1.1 Functional Testing

Functional testing akan dilakukan dengan cara memberikan suatu *inputan* data pada komponen yang diuji, kemudian selanjutnya akan diperiksa hasil dari *outputnya*. Suatu status validasi dikatakan valid apabila hasil *output* yang dikeluarkan sesuai, sedangkan untuk status validasi yang dikatakan tidak valid itu apabila *output* yang dihasilkan tidak sesuai atau terdapat *error*.

5.1.1.1 Hasil Functional Testing

Hasil dari pengujian *functional testing* dari aplikasi *expert locator* dapat dilihat pada tabel-tabel dibawah ini:

Tabel 5.1. Hasil *Functional Testing Login Admin Expert Locator*

<i>Login Admin Expert Locator</i>	
Tujuan:	Menguji apakah admin <i>Expert Locator</i> dapat masuk ke dalam sistem dengan menggunakan <i>username</i> dan <i>password</i>
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> belum melakukan <i>login</i> • <i>Username</i> dan <i>password</i> admin <i>Expert Locator</i> telah dibuat sebelumnya
Data yang di uji:	<i>Username</i> dan <i>password</i>
Prosedur uji:	<ol style="list-style-type: none"> 1. Membuka halaman utama 2. Mengetikkan <i>username</i> dan <i>password</i> 3. Klik tombol “Login”
Pertanyaan/catatan:	-
Status validasi:	Valid

Tabel 5.2. Hasil *Functional Testing Login Admin Expert Locator* dengan inputan yang salah

Login Admin Expert Locator	
Tujuan:	Menguji apakah admin <i>Expert Locator</i> dapat masuk ke dalam sistem dengan menggunakan <i>username</i> dan <i>password</i> yang salah
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> belum melakukan <i>login</i> • <i>Username</i> dan <i>password</i> admin <i>Expert Locator</i> telah dibuat sebelumnya
Data yang di uji:	<i>Username</i> dan <i>password</i>
Prosedur uji:	<ol style="list-style-type: none"> 1. Membuka halaman utama 2. Mengetikkan <i>username</i> dan <i>password</i> 3. <i>Klik</i> tombol “Login”
Pertanyaan/catatan:	<i>Username</i> dan <i>password</i> salah. Harap ulangi kembali.
Status validasi:	Valid

Tabel 5.3. Hasil *Functional Testing Admin Tambah Data Karyawan*

Tambah Data Karyawan	
Tujuan:	Menguji apakah data karyawan berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Karyawan”
Data yang di uji:	NIK, Kode Unit Kerja, Nama, Poscode, Jabatan, Postitle, Unit Kerja, dan Workloc
Prosedur uji:	<ol style="list-style-type: none"> 1. <i>Klik</i> tombol “Tambah” pada halaman “Karyawan” 2. Mengetikkan data karyawan pada <i>form</i> tambah data karyawan 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.4. Hasil *Functional Testing Admin Tambah Data Karyawan* dengan NIK yang sudah digunakan

Tambah Data Karyawan	
Tujuan:	Menguji apakah data karyawan berhasil ditambahkan ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Karyawan”
Data yang di uji:	NIK, Kode Unit Kerja, Nama, Poscode, Jabatan, Postitle, Unit Kerja, dan Workloc

Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Karyawan” 2. Mengetikkan data karyawan pada <i>form</i> tambah data karyawan dengan NIK yang sudah terdaftar sebelumnya 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.5. Hasil *Functional Testing* Admin Ubah Data Karyawan

Ubah Data Karyawan	
Tujuan:	Menguji apakah data karyawan berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Karyawan”
Data yang di uji:	NIK, Kode Unit Kerja, Nama, Poscode, Jabatan, Postitle, Unit Kerja, dan Workloc
Prosedur uji:	<ol style="list-style-type: none"> Memilih data karyawan yang akan diubah <i>Klik</i> tombol “Edit” pada halaman “Karyawan” Mengubah data karyawan <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.6. Hasil *Functional Testing* Admin Ubah Data Karyawan dengan NIK yang sudah digunakan

Ubah Data Karyawan	
Tujuan:	Menguji apakah data karyawan berhasil diubah ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Karyawan”
Data yang di uji:	NIK, Kode Unit Kerja, Nama, Poscode, Jabatan, Postitle, Unit Kerja, dan Workloc
Prosedur uji:	<ol style="list-style-type: none"> Memilih data karyawan yang akan diubah <i>Klik</i> tombol “Edit” pada halaman “Karyawan” Mengubah data karyawan dengan memasukkan NIK yang sudah terdaftar sebelumnya <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.7. Hasil *Functional Testing* Admin Hapus Data Karyawan

Hapus Data Karyawan	
Tujuan:	Menguji apakah data karyawan berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Karyawan”
Data yang di uji:	NIK, Kode Unit Kerja, Nama, Poscode, Jabatan, Postitle, Unit Kerja, dan Workloc
Prosedur uji:	<ol style="list-style-type: none"> Memilih data karyawan yang akan dihapus <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.8. Hasil *Functional Testing* Admin Tambah Data Pendidikan

Tambah Data Pendidikan	
Tujuan:	Menguji apakah data pendidikan berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Pendidikan”
Data yang di uji:	NIK, Kode Unit Kerja, Education_level_no, Education_level_name, Education_field_no, Education_field_name, Location, Start_year, End_year, G_p_a, Achievement, Remark, Edu_institution_seq_no, Edu_institution_name, Education_lvl_received, Preferred_education, Rowversion dan Anik_baru
Prosedur uji:	<ol style="list-style-type: none"> <i>Klik</i> tombol “Tambah” pada halaman “Pendidikan” Mengetikkan data pendidikan pada <i>form</i> tambah data pendidikan <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.9. Hasil *Functional Testing* Admin Tambah Data Pendidikan dengan NIK yang sudah digunakan

Tambah Data Pendidikan	
Tujuan:	Menguji apakah data pendidikan berhasil ditambahkan ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pendidikan”
Data yang di uji:	NIK, Kode Unit Kerja, Education_level_no,

	Education_level_name, Education_field_no, Education_field_name, Location, Start_year, End_year G_p_a, Achievement, Remark, Edu_institution_seq_no Edu_institution_name, Education_lv1_received Prefered_education, Rowversion dan Anik_baru
Prosedur uji:	<ol style="list-style-type: none"> 1. <i>Klik</i> tombol “Tambah” pada halaman “Pendidikan” 2. Mengetikkan data pendidikan pada <i>form</i> tambah data pendidikan dengan NIK yang sudah terdaftar sebelumnya 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.10. Hasil *Functional Testing* Admin Ubah Data Pendidikan

Ubah Data Pendidikan	
Tujuan:	Menguji apakah data pendidikan berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Pendidikan”
Data yang di uji:	NIK, Kode Unit Kerja, Education_level_no, Education_level_name, Education_field_no, Education_field_name, Location, Start_year, End_year G_p_a, Achievement, Remark, Edu_institution_seq_no Edu_institution_name, Education_lv1_received Prefered_education, Rowversion dan Anik_baru
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih data pendidikan yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Pendidikan” 3. Mengubah data pendidikan 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.11. Hasil *Functional Testing* Admin Ubah Data Pendidikan dengan NIK yang sudah digunakan

Ubah Data Pendidikan	
Tujuan:	Menguji apakah data pendidikan berhasil diubah ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pendidikan”
Data yang di uji:	NIK, Kode Unit Kerja, Education_level_no, Education_level_name, Education_field_no,

	Education_field_name, Location, Start_year, End_year G_p_a, Achievement, Remark, Edu_institution_seq_no Edu_institution_name, Education_lvl_received Prefered_education, Rowversion dan Anik_baru
Prosedur uji:	1. Memilih data pendidikan yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Pendidikan” 3. Mengubah data pendidikan dengan memasukkan NIK yang sudah terdaftar sebelumnya 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.12. Hasil *Functional Testing* Admin Hapus Data Pendidikan

Hapus Data Pendidikan	
Tujuan:	Menguji apakah data pendidikan berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pendidikan”
Data yang di uji:	NIK, Kode Unit Kerja, Education_level_no, Education_level_name, Education_field_no, Education_field_name, Location, Start_year, End_year G_p_a, Achievement, Remark, Edu_institution_seq_no Edu_institution_name, Education_lvl_received Prefered_education, Rowversion dan Anik_baru
Prosedur uji:	1. Memilih data pendidikan yang akan dihapus 2. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.13. Hasil *Functional Testing* Admin Tambah Data Pengalaman Kerja

Tambah Data Pengalaman Kerja	
Tujuan:	Menguji apakah data pengalaman kerja berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Pengalaman”
Data yang di uji:	NIK, Kode Unit Kerja, No_sk, Tgl_sk, Ma_tmt, Mkt_gol, Mkb_gol, Kode_update, Nm_jbt, Nm_pl, Nm_ru, Nm_si Nm_bag, Nm_dep, Nm_kom, Nm_dir, dan Keterangan
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Pengalaman Kerja”

	2. Mengetikkan data pengalaman kerja pada <i>form</i> tambah data pengalaman kerja 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.14. Hasil *Functional Testing* Admin Tambah Data Pengalaman Kerja dengan NIK yang sudah digunakan

Tambah Data Pengalaman Kerja	
Tujuan:	Menguji apakah data pengalaman kerja berhasil ditambahkan ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pengalaman”
Data yang di uji:	NIK, Kode Unit Kerja, No_sk, Tgl_sk, Ma_tmt, Mkt_gol, Mkb_gol, Kode_update, Nm_jbt, Nm_pl, Nm_ru, Nm_si Nm_bag, Nm_dep, Nm_kom, Nm_dir, dan Keterangan
Prosedur uji:	<ol style="list-style-type: none"> <i>Klik</i> tombol “Tambah” pada halaman “Pengalaman Kerja” Mengetikkan data pengalaman kerja pada <i>form</i> tambah data pengalaman kerja dengan NIK yang sudah terdaftar sebelumnya <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.15. Hasil *Functional Testing* Admin Ubah Data Pengalaman Kerja

Ubah Data Pengalaman Kerja	
Tujuan:	Menguji apakah data pengalaman kerja berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Pengalaman”
Data yang di uji:	NIK, Kode Unit Kerja, No_sk, Tgl_sk, Ma_tmt, Mkt_gol, Mkb_gol, Kode_update, Nm_jbt, Nm_pl, Nm_ru, Nm_si Nm_bag, Nm_dep, Nm_kom, Nm_dir, dan Keterangan
Prosedur uji:	<ol style="list-style-type: none"> Memilih data pengalaman kerja yang akan diubah <i>Klik</i> tombol “Edit” pada halaman “Pengalaman Kerja” Mengubah data pengalaman kerja <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah

Status validasi:	Valid
------------------	-------

Tabel 5.16. Hasil *Functional Testing* Admin Ubah Data Pengalaman Kerja dengan NIK yang sudah digunakan

Ubah Data Pengalaman Kerja	
Tujuan:	Menguji apakah data pengalaman kerja berhasil diubah ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pengalaman”
Data yang di uji:	NIK, Kode Unit Kerja, No_sk, Tgl_sk, Ma_tmt, Mkt_gol, Mkb_gol, Kode_update, Nm_jbt, Nm_pl, Nm_ru, Nm_si Nm_bag, Nm_dep, Nm_kom, Nm_dir, dan Keterangan
Prosedur uji:	<ol style="list-style-type: none"> Memilih data pengalaman kerja yang akan diubah Klik tombol “Edit” pada halaman “Pengalaman Kerja” Mengubah data pengalaman kerja dengan memasukkan NIK yang sudah terdaftar sebelumnya Klik tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.17. Hasil *Functional Testing* Admin Hapus Data Pengalaman Kerja

Hapus Data Pengalaman Kerja	
Tujuan:	Menguji apakah data pengalaman kerja berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pengalaman”
Data yang di uji:	NIK, Kode Unit Kerja, No_sk, Tgl_sk, Ma_tmt, Mkt_gol, Mkb_gol, Kode_update, Nm_jbt, Nm_pl, Nm_ru, Nm_si Nm_bag, Nm_dep, Nm_kom, Nm_dir, dan Keterangan
Prosedur uji:	<ol style="list-style-type: none"> Memilih data pengalaman kerja yang akan dihapus Klik tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.18. Hasil *Functional Testing* Admin Tambah Data Training

Tambah Data Training	
Tujuan:	Menguji apakah data training berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Training”

Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Training” 2. Mengetikkan data training pada <i>form</i> tambah data training 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.19. Hasil *Functional Testing* Admin Tambah Data Training dengan NIK yang sudah digunakan

Tambah Data Training	
Tujuan:	Menguji apakah data training berhasil ditambahkan ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Training”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Training” 2. Mengetikkan data training pada <i>form</i> tambah data training dengan NIK yang sudah terdaftar sebelumnya 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.20. Hasil *Functional Testing* Admin Ubah Data Training

Ubah Data Training	
Tujuan:	Menguji apakah data training berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Training”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	1. Memilih data training yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Training” 3. Mengubah data training 4. <i>Klik</i> tombol “Simpan”

Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.21. Hasil *Functional Testing* Admin Ubah Data Training dengan NIK yang sudah digunakan

Ubah Data Training	
Tujuan:	Menguji apakah data training berhasil diubah ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Training”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	<ol style="list-style-type: none"> Memilih data training yang akan diubah Klik tombol “Edit” pada halaman “Training” Mengubah data training dengan memasukkan NIK yang sudah terdaftar sebelumnya Klik tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.22. Hasil *Functional Testing* Admin Hapus Data Training

Hapus Data Training	
Tujuan:	Menguji apakah data training berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Training”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	<ol style="list-style-type: none"> Memilih data training yang akan dihapus Klik tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.23. Hasil *Functional Testing* Admin Tambah Data Sertifikasi

Tambah Data Sertifikasi	
Tujuan:	Menguji apakah data sertifikasi berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Sertifikasi”

Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Sertifikasi” 2. Mengetikkan data sertifikasi pada <i>form</i> tambah data sertifikasi 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.24. Hasil *Functional Testing* Admin Tambah Data Sertifikasi dengan NIK yang sudah digunakan

Tambah Data Sertifikasi	
Tujuan:	Menguji apakah data sertifikasi berhasil ditambahkan ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Sertifikasi”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Sertifikasi” 2. Mengetikkan data sertifikasi pada <i>form</i> tambah data sertifikasi dengan NIK yang sudah terdaftar sebelumnya 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.25. Hasil *Functional Testing* Admin Ubah Data Sertifikasi

Ubah Data Sertifikasi	
Tujuan:	Menguji apakah data sertifikasi berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Sertifikasi”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	1. Memilih data sertifikasi yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Sertifikasi” 3. Mengubah data sertifikasi

	4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.26. Hasil *Functional Testing* Admin Ubah Data Sertifikasi dengan NIK yang sudah digunakan

Ubah Data Sertifikasi	
Tujuan:	Menguji apakah data sertifikasi berhasil diubah ke dalam sistem dengan menggunakan NIK yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “sertifikasi”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	<ol style="list-style-type: none"> Memilih data sertifikasi yang akan diubah <i>Klik</i> tombol “Edit” pada halaman “Sertifikasi” Mengubah data sertifikasi dengan memasukkan NIK yang sudah terdaftar sebelumnya <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.27. Hasil *Functional Testing* Admin Hapus Data Sertifikasi

Hapus Data Sertifikasi	
Tujuan:	Menguji apakah data sertifikasi berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Sertifikasi”
Data yang di uji:	NIK, Kode Unit Kerja, Training_event_id, Course_id Nama_training, Topik, Durasi, Start_date, End_date Tempat, dan Penyelenggara
Prosedur uji:	<ol style="list-style-type: none"> Memilih data sertifikasi yang akan dihapus <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.28. Hasil *Functional Testing* Admin Tambah Data Unit Kerja

Tambah Data Unit Kerja	
Tujuan:	Menguji apakah data unit kerja berhasil ditambahkan ke dalam sistem

Kebutuhan awal:	• Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Unit Kerja”
Data yang di uji:	Kode Unit Kerja dan Content Unit Kerja
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Unit Kerja” 2. Mengetikkan data unit kerja pada <i>form</i> tambah data unit kerja 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.29. Hasil *Functional Testing* Admin Tambah Data Unit Kerja dengan Kode Unit Kerja yang sudah digunakan

Tambah Data Unit Kerja	
Tujuan:	Menguji apakah data unit kerja berhasil ditambahkan ke dalam sistem dengan menggunakan Kode Unit Kerja yang sudah digunakan
Kebutuhan awal:	• Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Unit Kerja”
Data yang di uji:	Kode Unit Kerja dan Content Unit Kerja
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Unit Kerja” 2. Mengetikkan data unit kerja pada <i>form</i> tambah data unit kerja dengan Kode Unit Kerja yang sudah terdaftar sebelumnya 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.30. Hasil *Functional Testing* Admin Ubah Data Unit Kerja

Ubah Data Unit Kerja	
Tujuan:	Menguji apakah data unit kerja berhasil diubah melalui sistem
Kebutuhan awal:	• Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Unit Kerja”
Data yang di uji:	Kode Unit Kerja dan Content Unit Kerja
Prosedur uji:	1. Memilih data unit kerja yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Unit Kerja” 3. Mengubah data unit kerja 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.31. Hasil *Functional Testing* Admin Ubah Data Unit Kerja dengan Kode Unit Kerja yang sudah digunakan

Ubah Data Unit Kerja	
Tujuan:	Menguji apakah data unit kerja berhasil diubah ke dalam sistem dengan menggunakan Kode Unit Kerja yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Unit Kerja”
Data yang di uji:	Kode Unit Kerja dan Content Unit Kerja
Prosedur uji:	<ol style="list-style-type: none"> Memilih data unit kerja yang akan diubah Klik tombol “Edit” pada halaman “Unit Kerja” Mengubah data unit kerja dengan memasukkan Kode Unit Kerja yang sudah terdaftar sebelumnya Klik tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.32. Hasil *Functional Testing* Admin Hapus Data Unit Kerja

Hapus Data Unit Kerja	
Tujuan:	Menguji apakah data unit kerja berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Unit Kerja”
Data yang di uji:	Kode Unit Kerja dan Content Unit Kerja
Prosedur uji:	<ol style="list-style-type: none"> Memilih data unit kerja yang akan dihapus Klik tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.33. Hasil *Functional Testing* Admin Tambah Data *User*

Tambah Data User	
Tujuan:	Menguji apakah data <i>user</i> berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “User”
Data yang di uji:	NIK, Kode unit kerja, Id group, Id akses, <i>username</i> , dan <i>password user</i>
Prosedur uji:	<ol style="list-style-type: none"> Klik tombol “Tambah” pada halaman “User” Mengetikkan data <i>user</i> pada <i>form</i> tambah data <i>user</i> Klik tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan

Status validasi:	Valid
------------------	-------

Tabel 5.34. Hasil *Functional Testing* Admin Tambah Data *User* dengan username yang sudah digunakan

Tambah Data User	
Tujuan:	Menguji apakah data <i>user</i> berhasil ditambahkan ke dalam sistem dengan menggunakan <i>username</i> yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “<i>User</i>”
Data yang di uji:	NIK, Kode unit kerja, Id group, Id akses, <i>username</i> , dan <i>password user</i>
Prosedur uji:	<ol style="list-style-type: none"> Klik tombol “Tambah” pada halaman “<i>User</i>” Mengetikkan data <i>user</i> pada <i>form</i> tambah data <i>user</i> dengan <i>username</i> yang sudah terdaftar sebelumnya Klik tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.35. Hasil *Functional Testing* Admin Ubah Data *User*

Ubah Data User	
Tujuan:	Menguji apakah data <i>user</i> berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “<i>User</i>”
Data yang di uji:	NIK, Kode unit kerja, Id group, Id akses, <i>username</i> , dan <i>password user</i>
Prosedur uji:	<ol style="list-style-type: none"> Memilih data <i>user</i> yang akan diubah Klik tombol “Edit” pada halaman “<i>User</i>” Mengubah data <i>user</i> Klik tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.36. Hasil *Functional Testing* Admin Ubah Data *User* dengan username yang sudah digunakan

Ubah Data User	
Tujuan:	Menguji apakah data <i>user</i> berhasil diubah ke dalam sistem dengan menggunakan <i>username</i> yang sudah digunakan
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “<i>User</i>”

Data yang di uji:	NIK, Kode unit kerja, Id group, Id akses, <i>username</i> , dan <i>password user</i>
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih data <i>user</i> yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “User” 3. Mengubah data <i>user</i> dengan memasukkan <i>username</i> yang sudah terdaftar sebelumnya 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Gagal Simpan!
Status validasi:	Valid

Tabel 5.37. Hasil *Functional Testing* Admin Hapus Data *User*

Hapus Data User	
Tujuan:	Menguji apakah data <i>user</i> berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “User”
Data yang di uji:	NIK, Kode unit kerja, Id group, Id akses, <i>username</i> , dan <i>password user</i>
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih data <i>user</i> yang akan dihapus 2. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.38. Hasil *Functional Testing* Admin Tambah Data Akses

Tambah Data Akses	
Tujuan:	Menguji apakah data akses berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Akses”
Data yang di uji:	Level Akses
Prosedur uji:	<ol style="list-style-type: none"> 1. <i>Klik</i> tombol “Tambah” pada halaman “Akses” 2. Mengetikkan data akses pada <i>form</i> tambah data akses 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.39. Hasil *Functional Testing* Admin Ubah Data Akses

Ubah Data Akses	
Tujuan:	Menguji apakah data akses berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Akses”
Data yang di uji:	Level Akses

Prosedur uji:	1. Memilih data akses yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Akses” 3. Mengubah data akses 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.40. Hasil *Functional Testing* Admin Hapus Data Akses

Hapus Data Akses	
Tujuan:	Menguji apakah data akses berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Akses”
Data yang di uji:	Level Akses
Prosedur uji:	1. Memilih data akses yang akan dihapus 2. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.41. Hasil *Functional Testing* Admin Tambah Data Group

Tambah Data Group	
Tujuan:	Menguji apakah data group berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Group”
Data yang di uji:	Level Group dan Nama Group
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Group” 2. Mengetikkan data group pada <i>form</i> tambah data group 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.42. Hasil *Functional Testing* Admin Ubah Data Group

Ubah Data Group	
Tujuan:	Menguji apakah data group berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Group”
Data yang di uji:	Level Group dan Nama Group
Prosedur uji:	1. Memilih data group yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Group” 3. Mengubah data group 4. <i>Klik</i> tombol “Simpan”

Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.43. Hasil *Functional Testing* Admin Hapus Data Group

Hapus Data Group	
Tujuan:	Menguji apakah data group berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Group”
Data yang di uji:	Level Group dan Nama Group
Prosedur uji:	<ol style="list-style-type: none"> Memilih data group yang akan dihapus Klik tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.44. Hasil *Functional Testing* Admin Tambah Data Status

Tambah Data Status	
Tujuan:	Menguji apakah data status berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Status”
Data yang di uji:	Id User dan User Status
Prosedur uji:	<ol style="list-style-type: none"> Klik tombol “Tambah” pada halaman “Status” Mengetikkan data status pada <i>form</i> tambah data status Klik tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.45. Hasil *Functional Testing* Admin Ubah Data Status

Ubah Data Status	
Tujuan:	Menguji apakah data status berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Status”
Data yang di uji:	Id User dan User Status
Prosedur uji:	<ol style="list-style-type: none"> Memilih data status yang akan diubah Klik tombol “Edit” pada halaman “Status” Mengubah data status Klik tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.46. Hasil *Functional Testing* Admin Hapus Data Status

Hapus Data Status	
Tujuan:	Menguji apakah data status berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Status”
Data yang di uji:	Id User dan User Status
Prosedur uji:	<ol style="list-style-type: none"> Memilih data status yang akan dihapus Klik tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.47. Hasil *Functional Testing* Admin Tambah Data Pesan

Tambah Data Pesan	
Tujuan:	Menguji apakah data pesan berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Pesan”
Data yang di uji:	Id Pengirim, Id Penerima, Judul, dan Pesan
Prosedur uji:	<ol style="list-style-type: none"> Klik tombol “Tambah” pada halaman “Pesan” Mengetikkan data pesan pada <i>form</i> tambah data pesan Klik tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.48. Hasil *Functional Testing* Admin Ubah Data Pesan

Ubah Data Pesan	
Tujuan:	Menguji apakah data status berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Pesan”
Data yang di uji:	Id Pengirim, Id Penerima, Judul, dan Pesan
Prosedur uji:	<ol style="list-style-type: none"> Memilih data pesan yang akan diubah Klik tombol “Edit” pada halaman “Pesan” Mengubah data pesan Klik tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.49. Hasil *Functional Testing* Admin Hapus Data Pesan

Hapus Data Pesan	
Tujuan:	Menguji apakah data pesan berhasil dihapus melalui sistem

Kebutuhan awal:	• Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Pesan”
Data yang di uji:	Id Pengirim, Id Penerima, Judul, dan Pesan
Prosedur uji:	1. Memilih data pesan yang akan dihapus 2. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.50. Hasil *Functional Testing* Admin Tambah Data Album

Tambah Data Album	
Tujuan:	Menguji apakah data album berhasil ditambahkan ke dalam sistem
Kebutuhan awal:	• Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem dan mengakses modul “Album”
Data yang di uji:	Id User, Id Nama Album, dan Foto
Prosedur uji:	1. <i>Klik</i> tombol “Tambah” pada halaman “Album” 2. Memilih data album pada <i>form</i> tambah data album 3. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.51. Hasil *Functional Testing* Admin Ubah Data Album

Ubah Data Album	
Tujuan:	Menguji apakah data album berhasil diubah melalui sistem
Kebutuhan awal:	• Admin <i>expert locator</i> berhasil masuk ke sistem dan mengakses modul “Album”
Data yang di uji:	Id User, Id Nama Album, dan Foto
Prosedur uji:	1. Memilih data album yang akan diubah 2. <i>Klik</i> tombol “Edit” pada halaman “Album” 3. Mengubah data album 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil diubah
Status validasi:	Valid

Tabel 5.52. Hasil *Functional Testing* Admin Hapus Data Album

Hapus Data Album	
Tujuan:	Menguji apakah data album berhasil dihapus melalui sistem
Kebutuhan awal:	• Admin <i>Expert Locator</i> berhasil masuk ke sistem dan mengakses modul “Album”
Data yang di uji:	Id User, Id Nama Album, dan Foto

Prosedur uji:	1. Memilih data album yang akan dihapus 2. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.53. Hasil *Functional Testing Logout Admin*

Logout Admin Expert Locator	
Tujuan:	Menguji apakah admin <i>expert locator</i> berhasil keluar dari sistem
Kebutuhan awal:	<ul style="list-style-type: none"> Admin <i>Expert Locator</i> berhasil masuk ke dalam sistem
Data yang di uji:	<i>Username</i> dan <i>Password</i>
Prosedur uji:	1. Memilih menu <i>logout</i> pada halaman admin <i>expert locator</i>
Pertanyaan/catatan:	Anda Telah Logout
Status validasi:	Valid

Tabel 5.54. Hasil *Functional Testing Login User (Karyawan)*

Login User (Karyawan)	
Tujuan:	Menguji apakah user (karyawan) dapat masuk ke dalam sistem dengan menggunakan <i>username</i> dan <i>password</i>
Kebutuhan awal:	<ul style="list-style-type: none"> User (karyawan) belum melakukan <i>login</i> <i>Username</i> dan <i>password</i> user (karyawan) telah dibuat sebelumnya
Data yang di uji:	<i>Username</i> dan <i>password</i>
Prosedur uji:	<ol style="list-style-type: none"> Membuka halaman utama Mengetikkan <i>username</i> dan <i>password</i> <i>Klik</i> tombol “Login”
Pertanyaan/catatan:	-
Status validasi:	Valid

Tabel 5.55. Hasil *Functional Testing User (Karyawan) Menulis Status*

Menulis Status	
Tujuan:	Menguji apakah user dapat menuliskan status ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> Memilih menu <i>timeline</i> Mengetikkan data status pada form data status <i>Klik</i> tombol “POST”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.56. Hasil *Functional Testing User* (Karyawan) Ubah Status

Ubah Status	
Tujuan:	Menguji apakah data status berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu timeline 2. Memilih status yang akan dirubah 3. <i>Klik</i> tombol “Edit” 4. Mengubah data status pada form data status 5. <i>Klik</i> tombol “Edit”
Pertanyaan/catatan:	Data Berhasil Diubah
Status validasi:	Valid

Tabel 5.57. Hasil *Functional Testing User* (Karyawan) Hapus Status

Hapus Status	
Tujuan:	Menguji apakah data status berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu timeline 2. Memilih data yang akan dihapus 3. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.58. Hasil *Functional Testing User* (Karyawan) Menulis
Komentar

Menulis Komentar	
Tujuan:	Menguji apakah user dapat menuliskan komentar ke dalam sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu timeline 2. Memilih data yang akan dikomentari 3. <i>Klik</i> tombol “Komentar” 4. Mengetikkan data komentar pada form data komentar 5. <i>Klik</i> tombol “Enter”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.59. Hasil *Functional Testing User* (Karyawan) Ubah Komentar

Ubah Komentar	
Tujuan:	Menguji apakah data komentar berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu timeline 2. Memilih data yang dikomentari 3. <i>Klik</i> tombol “Komentar” 4. <i>Klik</i> tombol “Ubah” 5. Mengubah data komentar pada form data komentar 6. <i>Klik</i> tombol “Edit”
Pertanyaan/catatan:	Data Berhasil Diubah
Status validasi:	Valid

Tabel 5.60. Hasil *Functional Testing User* (Karyawan) Hapus

Komentar

Hapus Komentar	
Tujuan:	Menguji apakah data komentar berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu timeline 2. Memilih data yang dikomentari 3. <i>Klik</i> tombol “Komentar” 4. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.61. Hasil *Functional Testing User* (Karyawan) Tambah Teman

Tambah Teman	
Tujuan:	Menguji apakah data teman berhasil ditambahkan melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Teman
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu cari teman 2. Memilih teman yang akan ditambahkan 3. <i>Klik</i> tombol “Tambahkan”
Pertanyaan/catatan:	Berhasil Menambahkan Pertemanan
Status validasi:	Valid

Tabel 5.62. Hasil *Functional Testing User* (Karyawan) Hapus Teman

Hapus Teman	
Tujuan:	Menguji apakah data teman berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> User berhasil masuk ke dalam sistem
Data yang di uji:	Status
Prosedur uji:	<ol style="list-style-type: none"> Memilih menu teman Memilih data teman yang akan dihapus Klik tombol “Hapus”
Pertanyaan/catatan:	Berhasil Menghapus Pertemanan
Status validasi:	Valid

Tabel 5.63. Hasil *Functional Testing User* (Karyawan) Join Group

Join Group	
Tujuan:	Menguji apakah data join group berhasil ditambahkan melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> User berhasil masuk ke dalam sistem
Data yang di uji:	Join Group
Prosedur uji:	<ol style="list-style-type: none"> Memilih group yang diinginkan Klik tombol “Bergabung”
Pertanyaan/catatan:	Berhasil Bergabung dengan Group
Status validasi:	Valid

Tabel 5.64. Hasil *Functional Testing User* (Karyawan) Menulis Pesan Pribadi

Menulis Pesan Pribadi	
Tujuan:	Menguji apakah data pesan berhasil ditambahkan melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> User berhasil masuk ke dalam sistem
Data yang di uji:	Pesan
Prosedur uji:	<ol style="list-style-type: none"> Memilih menu pesan Klik tombol “Tulis” Mengetikkan data pesan pada form data pesan Klik tombol “Kirim”
Pertanyaan/catatan:	Data Berhasil Ditambahkan
Status validasi:	Valid

Tabel 5.65. Hasil *Functional Testing User* (Karyawan) Hapus Pesan Pribadi

Hapus Pesan Pribadi	
Tujuan:	Menguji apakah data pesan berhasil dihapus melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Pesan
Prosedur uji:	<ol style="list-style-type: none"> 1. Memilih menu pesan 2. Memilih data pesan yang akan dihapus 3. <i>Klik</i> tombol “Hapus”
Pertanyaan/catatan:	Data Berhasil Dihapus
Status validasi:	Valid

Tabel 5.66. Hasil *Functional Testing User* (Karyawan) Edit Profil

Edit Profil	
Tujuan:	Menguji apakah data profil berhasil diubah melalui sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	Profil
Prosedur uji:	<ol style="list-style-type: none"> 1. <i>Klik</i> tombol “Setting” 2. Memilih menu akun saya 3. Mengubah data profil pada form data profil 4. <i>Klik</i> tombol “Simpan”
Pertanyaan/catatan:	Data Berhasil Diubah
Status validasi:	Valid

Tabel 5.67. Hasil *Functional Testing Logout User* (Karyawan)

Logout User (Karyawan)	
Tujuan:	Menguji apakah user berhasil keluar dari sistem
Kebutuhan awal:	<ul style="list-style-type: none"> • User berhasil masuk ke dalam sistem
Data yang di uji:	<i>Username dan Password</i>
Prosedur uji:	1. Memilih menu <i>logout</i> pada halaman user
Pertanyaan/catatan:	Anda Telah Logout
Status validasi:	Valid

5.1.2 Usability Testing

Pada pengujian *usability testing* akan dilakukan dengan cara pengukuran terhadap kemudahan penggunaan aplikasi ini dari sisi penggunanya. Kemudahan penggunaan diukur berdasarkan 4 parameter, diantaranya: mudah dipelajari, mudah diingat, mudah dioperasikan, dan mudah dieksplorasi.

5.1.2.1 Hasil Usability Testing

Usability testing disini mencobakan sistem ke pengguna. Pernyataan pengujian dirancang berdasarkan faktor kemudahan penggunaan sistem. Dari aktifitas tersebut diujikan terhadap beberapa karyawan yang nantinya akan menggunakan aplikasi ini. Berikut hasil dari pengujian *usability*.

Tabel 5.68. Hasil *Usability Testing*

Pernyataan	Respon	Hasil
1. Anda merasa bahwa interaksi dengan aplikasi <i>expert locator</i> mudah dimengerti.	Sangat Setuju	5
	Setuju	1
	Netral	1
	Tidak Setuju	-
	Sangat Tidak Setuju	-
2. Belajar untuk mengoperasikan aplikasi <i>expert locator</i> mudah bagi Anda.	Sangat Setuju	3
	Setuju	4
	Netral	-
	Tidak Setuju	-
	Sangat Tidak Setuju	-
3. Anda merasa bahwa interaksi yang disediakan oleh aplikasi <i>expert locator</i> ini fleksibel untuk digunakan.	Sangat Setuju	1
	Setuju	4
	Netral	2
	Tidak Setuju	-
	Sangat Tidak Setuju	-
4. Sangat mudah bagi Anda	Sangat Setuju	4

untuk menjadi mahir dalam menggunakan aplikasi <i>expert locator</i> .	Setuju	2
	Netral	1
	Tidak Setuju	-
	Sangat Tidak Setuju	-
5. Anda merasa aplikasi <i>expert locator</i> ini mudah digunakan.	Sangat Setuju	2
	Setuju	4
	Netral	1
	Tidak Setuju	-
	Sangat Tidak Setuju	-
6. Belajar menggunakan aplikasi <i>expert locator</i> sangat mudah bagi Anda.	Sangat Setuju	-
	Setuju	7
	Netral	-
	Tidak Setuju	-
	Sangat Tidak Setuju	-
7. Anda tidak pernah menggunakan aplikasi yang serupa dengan aplikasi <i>expert locator</i> ini, tetapi aplikasi ini mudah digunakan.	Sangat Setuju	-
	Setuju	6
	Netral	-
	Tidak Setuju	1
	Sangat Tidak Setuju	-
8. Navigasi pada aplikasi <i>expert locator</i> ini mudah bagi Anda.	Sangat Setuju	4
	Setuju	3
	Netral	-

	Tidak Setuju	-
	Sangat Tidak Setuju	-
9. Menurut Anda usaha yang dibutuhkan untuk menguasai dan mahir dalam menjalankan aplikasi ini rendah.	Sangat Setuju	1
	Setuju	5
	Netral	1
	Tidak Setuju	-
	Sangat Tidak Setuju	-
10. Anda merasa sangat mudah untuk menjadi ahli dalam menggunakan aplikasi <i>expert locator</i> .	Sangat Setuju	5
	Setuju	1
	Netral	1
	Tidak Setuju	-
	Sangat Tidak Setuju	-

5.2 Analisa Hasil

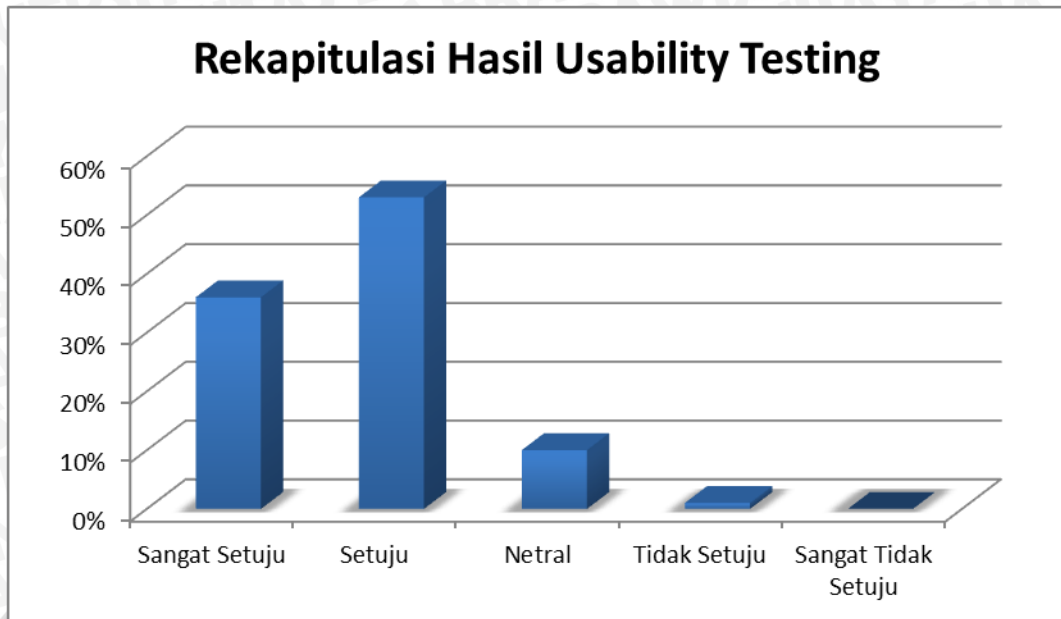
Tahap ini akan membahas mengenai analisa hasil pengujian yang telah dilakukan.

5.2.1 Analisa Hasil *Functional Testing*

Berdasarkan dari hasil pengujian aplikasi *Expert Locator* menggunakan metode *functional testing*, dapat diketahui bahwa 100% fitur dapat berjalan dengan baik. Dari hasil tersebut, implementasi dari perancangan yang telah dilakukan pada bab 3 dapat dikatakan berhasil.

5.2.2 Analisa Hasil *Usability Testing*

Rekapitulasi terhadap pernyataan yang digunakan untuk mengukur kemudahan penggunaan aplikasi *Expert Locator* ditunjukkan pada gambar 5.1.



Gambar 5.1. Rekapitulasi Hasil dari Pengujian *Usability Testing* dalam bentuk persentase (%)

Dari hasil rekapitulasi diatas maka dapat diketahui bahwa pengguna dari aplikasi *Expert Locator* menyatakan setuju bahwa aplikasi ini memenuhi kemudahan penggunaan. Hal ini dapat didasarkan dari hasil rekapitulasi diatas antara lain 36% menyatakan sangat setuju, 53% setuju, 10% netral, 1% tidak setuju dan 0% sangat tidak setuju mengenai kemudahan penggunaan dari sistem *expert locator*.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik, dapat diambil kesimpulan :

1. *Prasyarat* dari Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik telah sesuai dengan kebutuhan perusahaan yaitu sistem ini dapat memberikan informasi mengenai kemampuan (*talent*) yang dimiliki oleh setiap karyawan dan memberikan kemudahan kepada setiap karyawan untuk bisa berinteraksi dengan seseorang yang dikategorikan sebagai seorang *expert* didalam perusahaan sesuai dengan bidang pekerjaannya masing-masing.
2. Perancangan dan pembuatan *prototype* Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik berbasis web telah dibuat sesuai dengan kebutuhan pengguna yang telah dianalisis sebelumnya di bab3.
3. Pengujian dilakukan dengan menggunakan metode *functional testing* dan *usability testing*. Berdasarkan hasil pengujian dengan menggunakan metode *functional testing*, seluruh fungsi yang terdapat pada Sistem *Expert Locator* Untuk Mendukung *Sharing Knowledge* Pada Jejaring Para Ahli Pada PT. Petrokimia Gresik berjalan dengan baik dan secara fungsional berfungsi 100% valid. Sedangkan untuk pengujian *usability testing* diperoleh hasil antara lain 36% menyatakan sangat setuju, 53% setuju, 10% netral, 1% tidak setuju dan 0% sangat tidak setuju mengenai kemudahan penggunaan dari sistem *expert locator*.

6.2 Saran

Saran yang dapat diberikan untuk pengembangan Sistem *Expert Locator* antara lain:

1. Melakukan pengujian ulang pada Sistem *Expert Locator* ini apabila nantinya akan diintegrasikan dengan sistem yang lebih besar atau apabila adanya penambahan fitur.

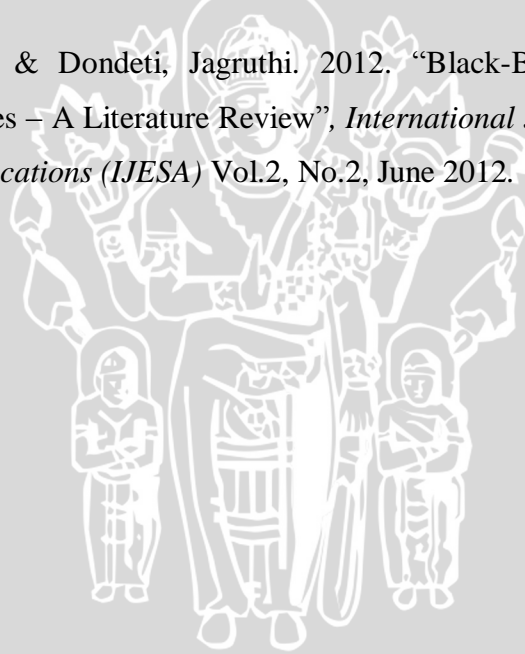
2. Sistem *Expert Locator* ini ke depan nantinya dapat diintegrasikan dengan database yang ada diperusahaan.
3. *Security Systemnya* agar lebih di tingkatkan lagi untuk menghindari serangan-serangan yang tidak diinginkan.



DAFTAR PUSTAKA

- [ROS-11] A.S. Rosa, M. Shalahuddin, M. 2011. *“Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)”*. Bandung: Modula.
- [SEP-11] Septian, Gungun. 2011. *“Trik Pintar Menguasai Codeigniter”*. Jakarta: PT Elex Media Komputindo.
- [ADE-10] Adelheid, Andrea. 2010. *“Buku Pintar Menguasai PHP & MySQL”*. Jakarta: Mediakita.
- [RAH-11] Raharjo, Budi. 2011. *“Belajar Otodidak Membuat Database Menggunakan MySQL”*. Jakarta: Informatika.
- [AND-04] Andi. 2004. *“Aplikasi Program PHP + MySQL Untuk Membuat Website Interaktif”*. Jakarta: MADCOMS.
- [PGS-12] PT. Petrokimia, Gresik. 2012 <http://www.petrokimia-gresik.com/> Diakses tanggal [3 maret 2014].
- [WID-12] Widayanti, Riya. 2012. *“Penerapan Knowledge Management Dalam Organisasi”*. Jakarta: FIK-UEU.
- [NAW-12] Nawawi, Ismail. 2012. *“Manajemen Pengetahuan (Knowledge Management)”*. Bogor: Ghalia Indonesia.
- [FEI-02] Fei, Thong Lip. 2002 http://www.cyberartsweb.org/cpace/ht/thonglipfei/nonaka_seci.html Diakses tanggal [21 april 2014].
- [HAS-09] A. Haslinda, A. Sarinah. 2009. *“A Review of Knowledge Management Models”*. Malaysia: FEM-UPM.
- [CHO-06] Chowdury, Naguib. 2006. *“Knowledge Audit Module: Overview and Sample Questionare”* from naguibch@yahoo.com.
- [PAR-03] Paramasivan, Thirumoorthy. 2003. *“Knowledge Audit”*. India : ICA.
- [SAM-06] Sagsan, Mustafa. 2006. *“A New Life Cycle Model For Processing Of Knowledge Management”*. Turki: FK-UB.

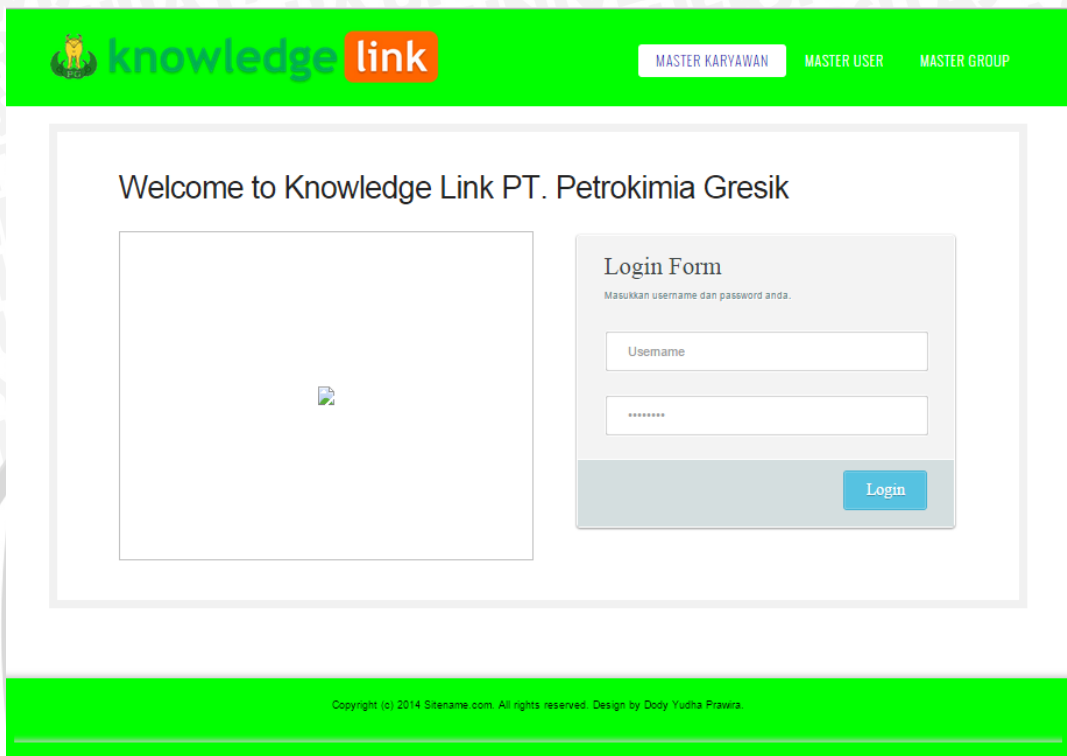
- [SEL-01] Smith, Elizabeth A. 2001. "The Role of Tacit and Explicit Knowledge in The Workplace", *Journal of Knowledge Management* 5. USA.
- [BEF-10] Becerra-Fernandez, Irma. & Sabherwal, Rajiv. 2010. "*Knowledge Management System & Processes*". New York: M.E Sharpe.
- [PUR-08] Purnama, Mirawati. & Budihardjo, Andreas. 2008. "*Peran Budaya Pembelajaran dan Knowledge Management*". Jakarta.
- [PHO-11] Phosaard, Satidchoke. & Wiriyapinit, Mongkolchai. 2011. "*Knowledge Management via Facebook : Building a Framework for Knowledge Management on a Social Network by Aligning Business, IT and Knowledge Management*". United Kingdom : WCE.
- [JOV-08] Jovanovic, Irena. 2008. "*Software Testing Methods And Techniques*". Belgrade.
- [NID-12] Nidhra, Srinivas. & Dondeti, Jagruthi. 2012. "Black-Box And White-Box Testing Techniques – A Literature Review", *International Journal of Embedded Systems and Applications (IJESA)* Vol.2, No.2, June 2012.



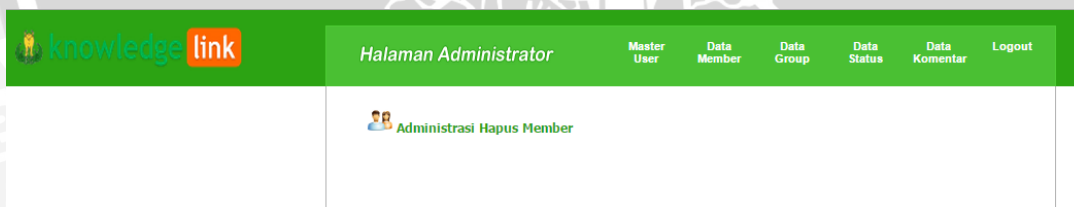
LAMPIRAN

Lampiran 1 : Mark-Up Aplikasi *Expert Locator*

1. Halaman Utama

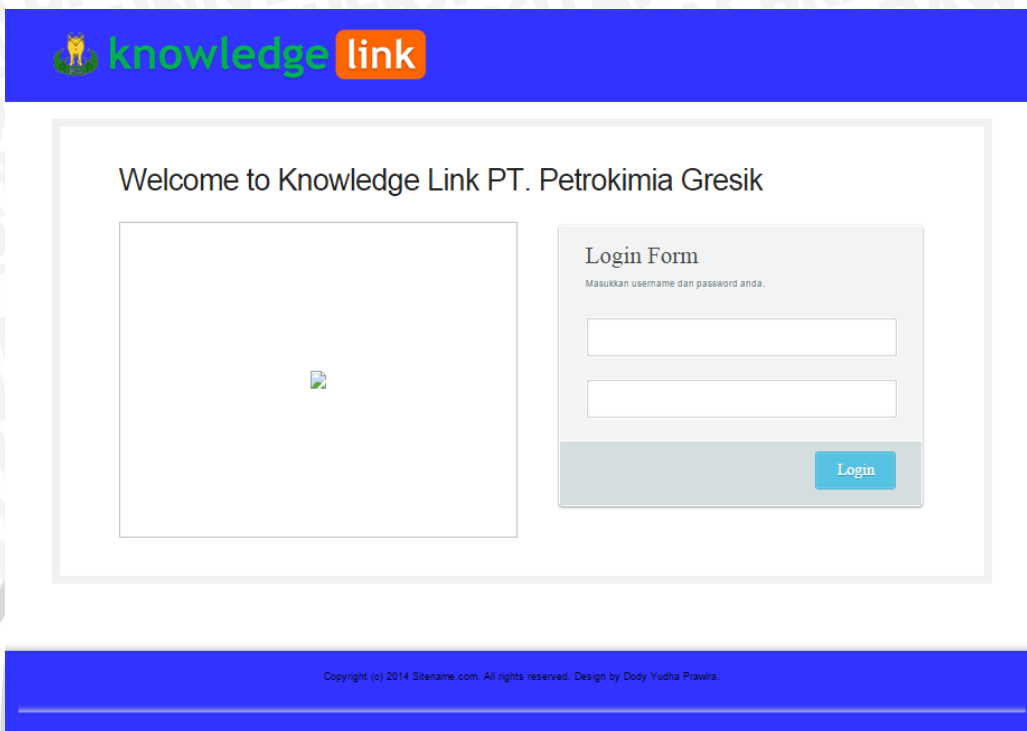


2. Halaman Admin

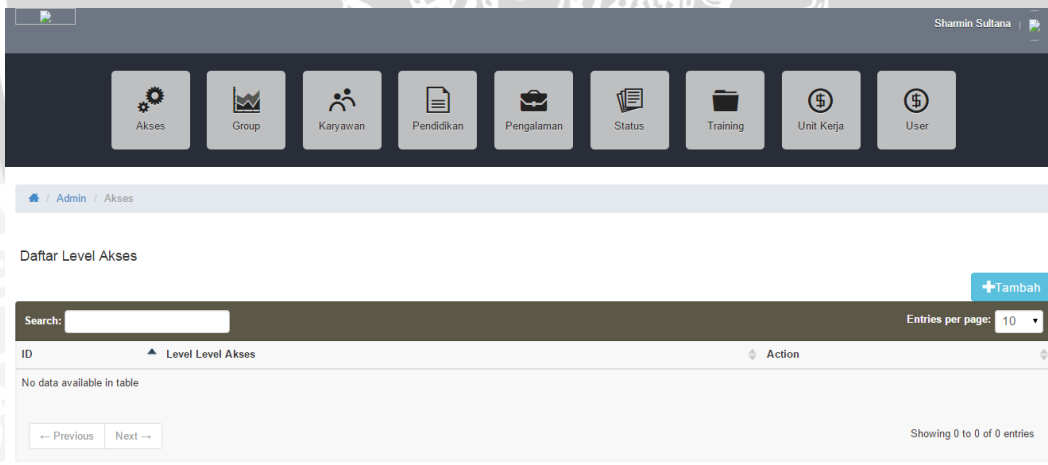


Lampiran 2 : Progress1 Aplikasi *Expert Locator*

1. Halaman Utama



2. Halaman Admin

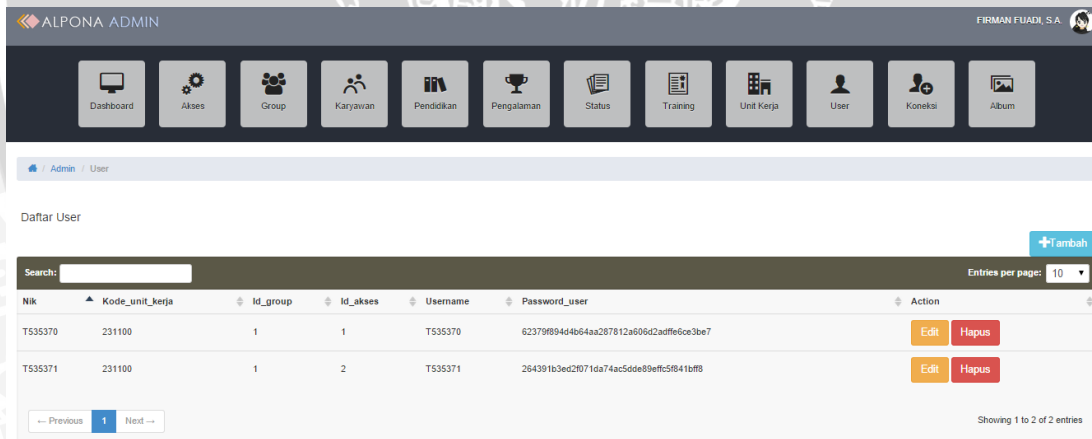


Lampiran 3 : Progress2 Aplikasi *Expert Locator*

1. Halaman Utama

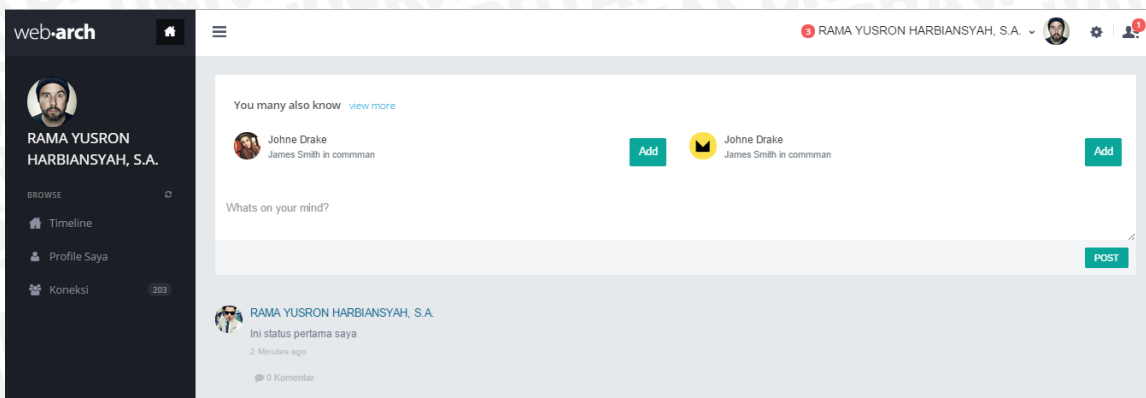


2. Halaman Admin

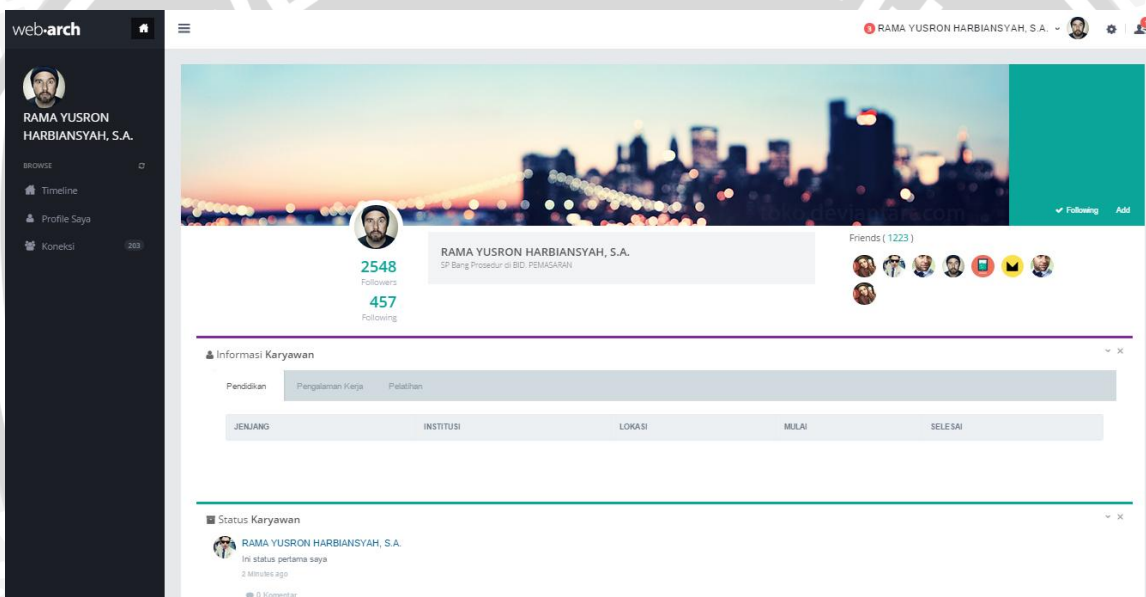


3. Halaman User

➤ Halaman Timeline



➤ Halaman Profil

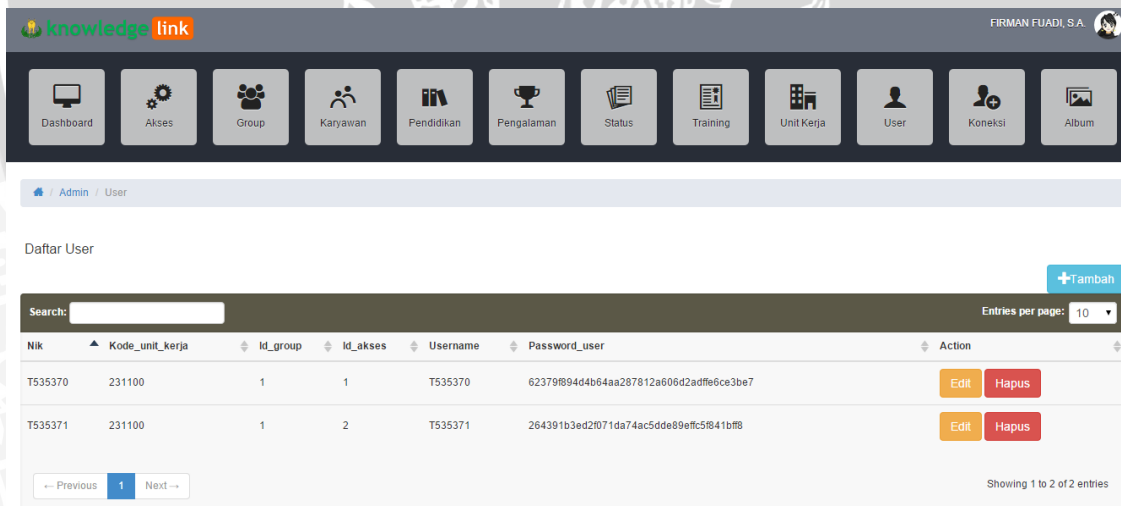


Lampiran 4 : Progress3 Aplikasi *Expert Locator*

1. Halaman Utama

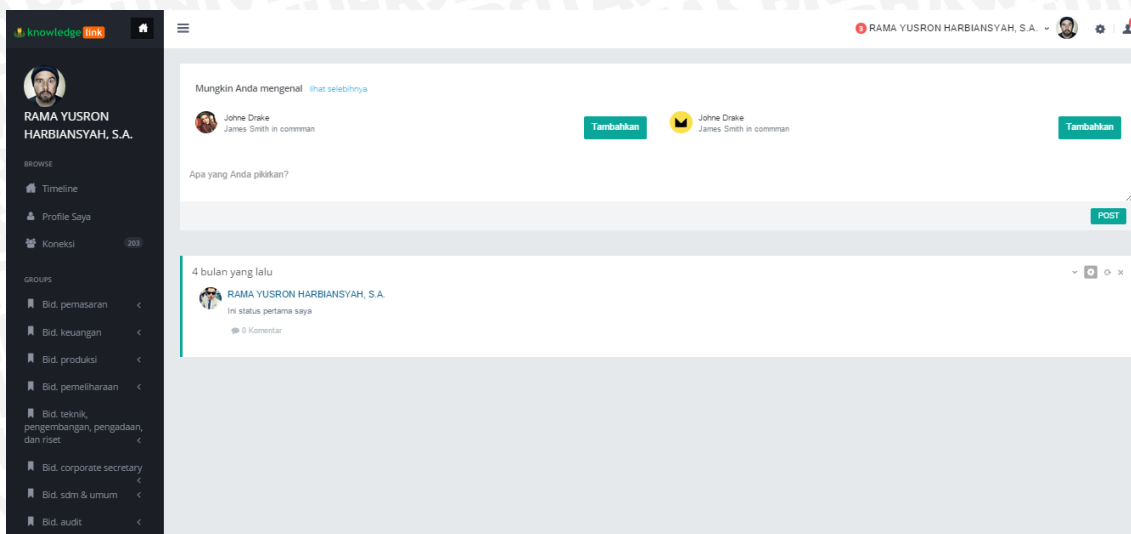


2. Halaman Admin

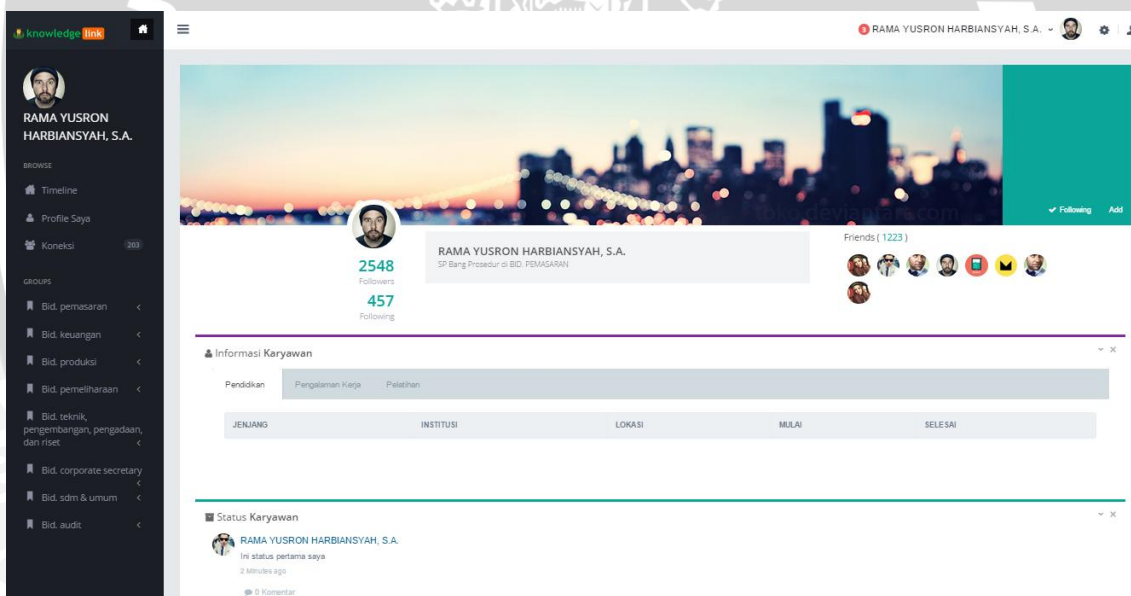


3. Halaman User

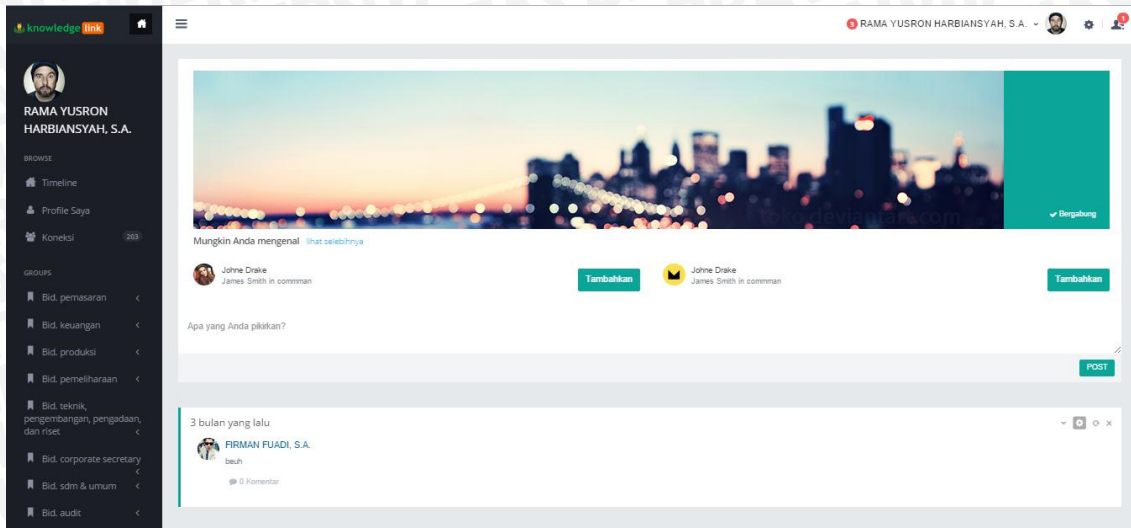
➤ Halaman Timeline



➤ Halaman Profil



➤ Halaman Group

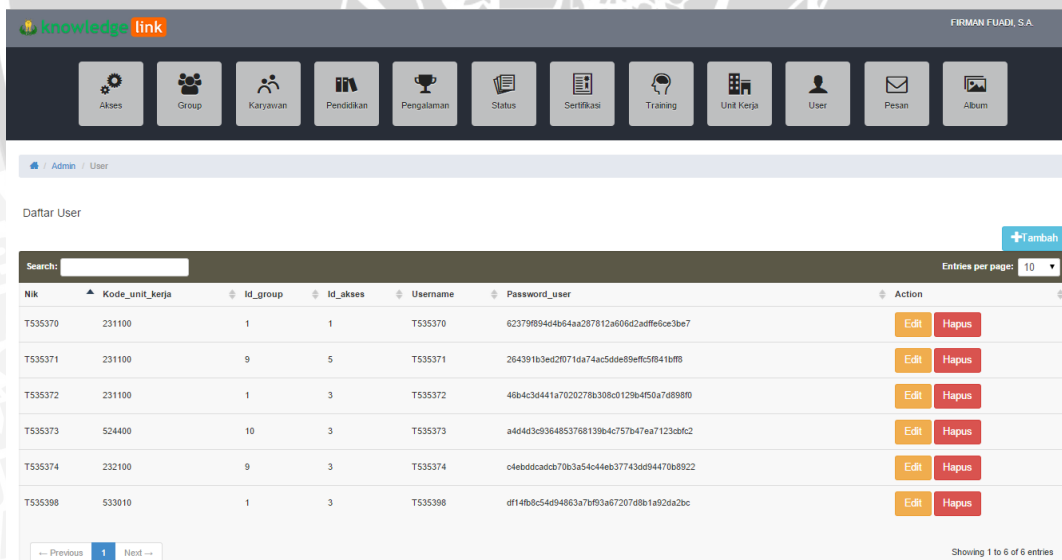


Lampiran 5 : Progress4 – (Final Progress) Aplikasi *Expert Locator*

1. Halaman Utama

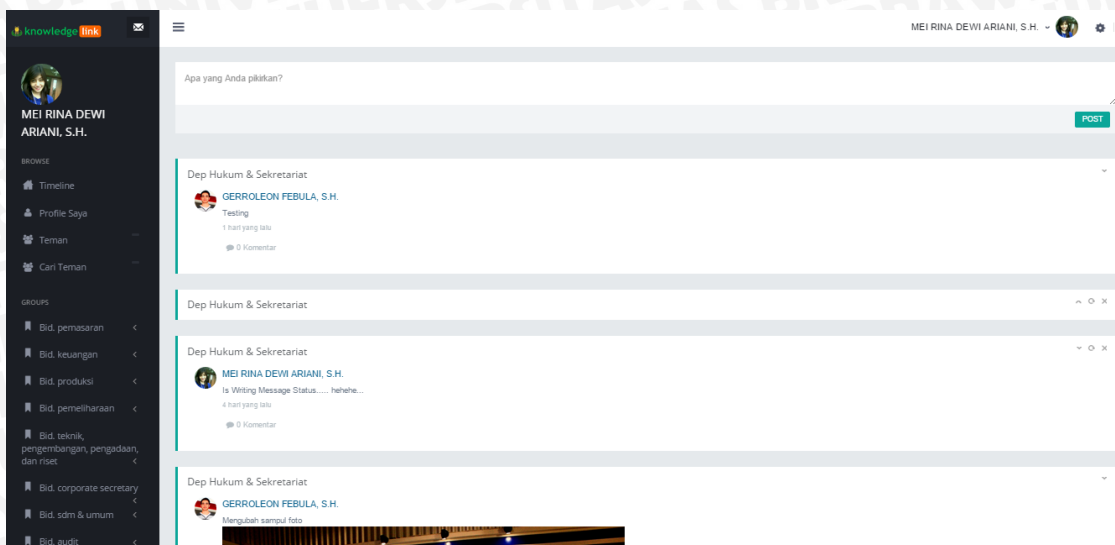


2. Halaman Admin

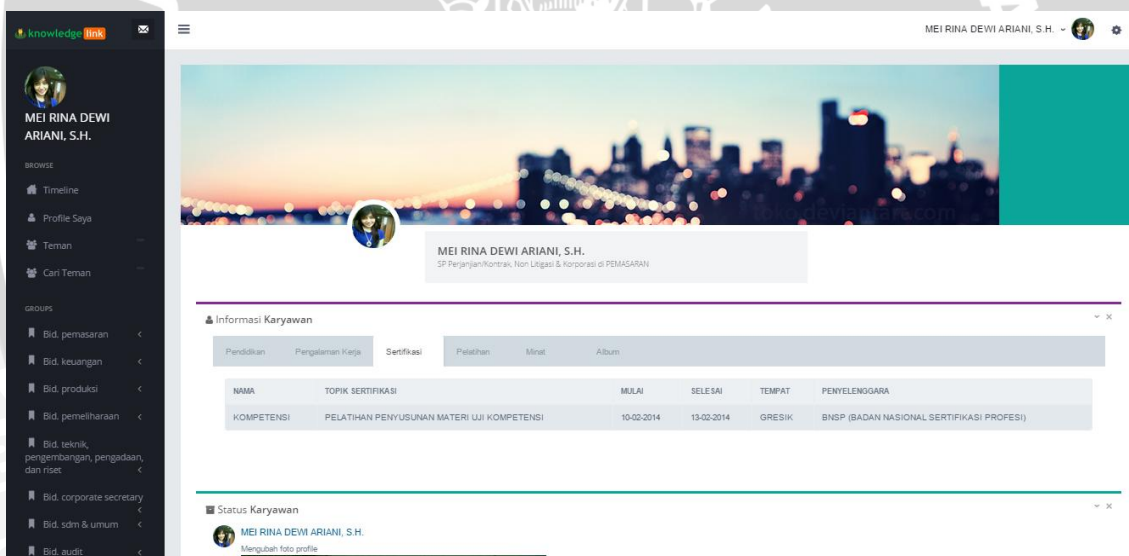


3. Halaman User

➤ Halaman Timeline



➤ Halaman Profil



➤ Halaman Group

