



**KERJA PRAKTIK – KI141330**

**Pembuatan Website Aplikasi  
Validasi SOP (E-SOP) Minarak Brantas Gas**

**Minarak Brantas Gas, Inc. - Sidoarjo**

**Departemen Maintenance**

**Minarak Brantas Gas, Inc. Wunut Gas Plant Porong Sidoarjo**

**Periode: 1 Maret 2021 - 31 Mei 2021**

**Oleh:**

Pembimbing Jurusan

Hudan Studiawan, S.Kom., M.Kom., Ph.D

Pembimbing Lapangan

Andi Fajar Yanto

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2021**



**KERJA PRAKTIK – KI141330**

## **Pembuatan Website Aplikasi Validasi SOP (E-SOP) Minarak Brantas Gas**

**Minarak Brantas Gas, Inc. - Sidoarjo**

**Departemen Maintenance**

**Minarak Brantas Gas, Inc. Wunut Gas Plant Porong Sidoarjo**

**Periode: 1 Maret 2021 - 31 Mei 2021**

**Oleh:**

Pembimbing Jurusan

Hudan Studiawan, S.Kom., M.Kom., Ph.D

Pembimbing Lapangan

Andi Fajar Yanto

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2021**

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN

## KERJA PRAKTIK

### Pembuatan Website Sistem Validasi SOP (E-SOP) Minarak Brantas Gas

Oleh:

Benny Hansen Lifindra                      05111840000010

Kresna Adhi Pramana                      05111840000072

Mengetahui,

Minarak Brantas Gas, Inc.

Pembimbing Kerja Praktik



Andi Fajar Yanto

Menyetujui,

Dosen Pembimbing

Kerja Praktik



Hudan Studiawan, S.Kom., M.Kom., Ph.D

NIP. 198705112012121003

SURABAYA

31 Mei, 2021

*[Halaman ini sengaja dikosongkan]*

**Pembuatan Sistem Informasi  
Validasi SOP (E-SOP) Minarak Brantas Gas**

Nama Mahasiswa : Benny Hansen Lifindra  
NRP : 05111840000010  
Nama Mahasiswa : Kresna Adhi Pramana  
NRP : 05111840000072  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Jurusan : Hudan Studiawan, S.Kom.,  
M.Kom., Ph.D  
Pembimbing Lapangan : Andi Fajar Yanto

## **ABSTRAK**

Selama ini proses validasi berkas SOP Minarak Brantas Gas Inc. dilakukan secara manual. Pengurusan berkas SOP Minarak memerlukan beberapa tahapan. Dokumen pertama dibuat dan diperiksa oleh pihak berwenang. Jika terdapat kesalahan, dokumen akan diperbaiki dan diperiksa ulang. Berkas SOP dapat disetujui atau ditolak. Jika berkas SOP disetujui, sebuah bukti persetujuan perlu dibuat kepada berkas SOP tersebut. Tidak hanya itu, berkas SOP secara berkala akan diperbaharui di masa-masa yang akan datang ketika diperlukan.

Untuk mempermudah proses pengelolaan berkas SOP, terutama dalam pembuatan bukti persetujuan dan pengelolaan riwayat kepengurusan berkas SOP, diperlukan aplikasi yang dapat memenuhi kebutuhan ini. Minarak Brantas Gas Inc. memiliki sistem informasi terintegrasi berupa aplikasi web Intramas. Aplikasi ini akan direncanakan untuk dibuat fitur baru khusus untuk mengurus proses validasi berkas SOP. Fitur baru ini adalah E-SOP.

Fitur E-SOP ini ditujukan untuk membantu Minarak Brantas Gas Inc. dalam mengelola berkas-berkas SOP. Semua berkas dapat dimasukkan ke dalam website untuk disimpan, diulas, dan divalidasi. E-SOP juga dibuat sedemikian rupa untuk menangani kasus-kasus khusus berkaitan dengan struktur departemen Minarak. Fitur E-SOP dapat membantu Minarak dalam pembuatan bukti persetujuan secara otomatis berdasarkan templat yang disediakan. Selain itu, E-SOP menyediakan opsi untuk memperbaharui berkas SOP yang telah divalidasi untuk diperiksa ulang guna pembaharuan.

***Kata kunci: SOP, Validasi, Website***

## **KATA PENGANTAR**

Puji syukur kami haturkan kepada Allah SWT karena berkat rahmat-Nya kami dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Departemen Teknik Informatika, yakni Kerja Praktek (KP).

Kami menyadari masih ada kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan kerja praktik ini.

Melalui buku ini, kami juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan kerja praktik hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Bapak Hudan Studiawan, S.Kom., M.Kom. selaku dosen pembimbing Kerja Praktik.
3. Bapak Andi Fajar Yanto selaku pembimbing lapangan kami di Minarak Brantas Gas, Inc. Sidoarjo.

Surabaya, 31 Mei 2021  
Benny Hansen Lifindra, Kresna Adhi Pramana



*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

LEMBAR PENGESAHAN	4
ABSTRAK	7
KATA PENGANTAR	8
DAFTAR ISI	10
DAFTAR TABEL	13
DAFTAR GAMBAR	14
BAB I PENDAHULUAN	16
1.1. Latar Belakang	16
1.2. Tujuan	17
1.3. Manfaat	17
1.4. Rumusan Masalah	17
1.5. Lokasi dan Waktu Kerja Praktik	17
1.6. Metodologi Kerja Praktik	18
1.7. Sistematika Laporan	20
BAB II PROFIL PERUSAHAAN	23
2.1. Profil Minarak Brantas Gas, Inc.	23
2.2. Sejarah Singkat Minarak Brantas Gas, Inc.	23
2.3. Program Kerja Minarak Brantas Gas, Inc.	24
BAB III TINJAUAN PUSTAKA	27
3.1. Pemrograman Web	27
3.2. HTML	27
3.3. PHP	27
3.4. XAMPP	28
3.5. Visual Studio Code	28

BAB IV ANALISIS DAN PERANCANGAN SISTEM	30
4.1. Analisis Sistem	30
4.1.1. Definisi Umum Aplikasi	30
4.1.2. Analisis Kebutuhan	31
4.1.2.1. Kebutuhan Fungsional	31
4.1.2.2. Kebutuhan Non-Fungsional	33
4.2. Perancangan Sistem	33
4.2.1. Desain Sistem	33
4.2.1.1. Use Case Diagram	35
4.2.1.2. <i>Conceptual Data Model</i>	36
4.2.1.3. <i>Physical Data Model</i>	37
4.2.1.4. Diagram Activity	38
BAB V IMPLEMENTASI SISTEM	64
5.1. Implementasi Controller	64
5.2. Implementasi View	109
BAB VI PENGUJIAN DAN EVALUASI	131
6.1. Tujuan Pengujian	131
6.2. Kriteria Pengujian	131
6.3. Skenario Pengujian	131
6.4. Evaluasi Pengujian	133
BAB VII KESIMPULAN DAN SARAN	137
7.1. Kesimpulan	137
7.2. Saran	137
DAFTAR PUSTAKA	139
BIODATA PENULIS I	141



## DAFTAR TABEL

Tabel 4. 1 Karakteristik Pengguna .....	30
Tabel 4. 2 Kebutuhan Fungsional .....	32
Tabel 4. 3 Kebutuhan Non-Fungsional .....	33
Tabel 6. 1 Evaluasi Pengujian sebagai L1 .....	133
Tabel 6. 2 Evaluasi Pengujian sebagai L2 .....	133
Tabel 6. 3 Evaluasi Pengujian sebagai L3 .....	134
Tabel 6. 4 Evaluasi Pengujian sebagai Admin.....	134

## DAFTAR GAMBAR

Gambar 4. 1 Use Case Diagram.....	35
Gambar 4. 2 Conceptual Data Model.....	36
Gambar 4. 3 Physical Data Model .....	37
Gambar 4. 4 Diagram Aktivitas Melakukan Login.....	38
Gambar 4. 5 Diagram Aktivitas Mengajukan SOP .....	39
Gambar 4. 6 Diagram Aktivitas Memperbaiki SOP .....	40
Gambar 4. 7 Diagram Aktivitas Merevisi SOP.....	41
Gambar 4. 8 Diagram Aktivitas Memvalidasi SOP.....	42
Gambar 4. 9 Diagram Aktivitas Menolak SOP.....	43
Gambar 4. 10 Diagram Aktivitas Melihat Daftar SOP Tervalidasi.....	44
Gambar 4. 11 Diagram Aktivitas Mengunduh Bukti Validasi SOP .....	45
Gambar 4. 12 Diagram Aktivitas Melihat Riwayat Perubahan SOP .....	46
Gambar 4. 13 Diagram Aktivitas Mengubah Data Diri .....	47
Gambar 4. 14 Diagram Aktivitas Menambah Pengguna Baru.....	48
Gambar 4. 15 Diagram Aktivitas Menambah Departemen Baru ..	49
Gambar 4. 16 Diagram Aktivitas Menambah Role Baru .....	50
Gambar 4. 17 Diagram Aktivitas Menambah Posisi Baru.....	51
Gambar 4. 18 Diagram Aktivitas Menambah Seksi Baru.....	52
Gambar 4. 19 Diagram Aktivitas Memperbaharui Data Pengguna.....	53
Gambar 4. 20 Diagram Aktivitas Memperbaharui Data Departemen .....	54
Gambar 4. 21 Diagram Aktivitas Memperbaharui Data Role.....	55
Gambar 4. 22 Diagram Aktivitas Memperbaharui Data Posisi.....	56
Gambar 4. 23 Diagram Aktivitas Memperbaharui Data Seksi .....	57
Gambar 4. 24 Diagram Aktivitas Menghapus Data Pengguna .....	58
Gambar 4. 25 Diagram Aktivitas Menghapus Data Departemen ..	59
Gambar 4. 26 Diagram Aktivitas Menghapus Data Role .....	60
Gambar 4. 27 Diagram Aktivitas Menghapus Data Posisi .....	61
Gambar 4. 28 Diagram Aktivitas Menghapus Seksi.....	62
Gambar 5. 1 Login Web .....	110
Gambar 5. 2 Home L1 .....	111

Gambar 5. 3 Home L2 atau L3 .....	111
Gambar 5. 4 Home Admin .....	112
Gambar 5. 5 Pengajuan SOP.....	113
Gambar 5. 6 Form Pengajuan SOP .....	113
Gambar 5. 7 Form Revisi SOP L1 .....	114
Gambar 5. 8 Validasi L2.....	115
Gambar 5. 9 Validasi L3.....	115
Gambar 5. 10 Form Revisi SOP L2 .....	116
Gambar 5. 11 Form Penolakan SOP .....	116
Gambar 5. 12 Daftar SOP .....	117
Gambar 5. 13 Form Pembaharuan SOP .....	118
Gambar 5. 14 Riwayat Pembaharuan.....	118
Gambar 5. 15 Departemen .....	119
Gambar 5. 16 Form Departemen Baru.....	120
Gambar 5. 17 Form Edit Departemen.....	120
Gambar 5. 18 Seksi .....	121
Gambar 5. 19 Form Seksi Baru.....	121
Gambar 5. 20 Form Edit Seksi.....	122
Gambar 5. 21 Posisi .....	123
Gambar 5. 22 Form Posisi Baru.....	123
Gambar 5. 23 Form Edit Posisi.....	124
Gambar 5. 24 Role .....	124
Gambar 5. 25 Form Role Baru.....	125
Gambar 5. 26 Form Edit Role.....	125
Gambar 5. 27 User .....	126
Gambar 5. 28 Form User Baru.....	127
Gambar 5. 29 Form Edit User.....	127
Gambar 5. 30 Form Edit User Password.....	128
Gambar 5. 31 Form Edit User Signature.....	129
Gambar 5. 32 Profile.....	129

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Minarak Brantas Gas, Inc. merupakan perusahaan yang bergerak dalam pengeboran minyak dan gas bumi. Perusahaan ini memiliki aplikasi berbasis web bernama Intramas atau Integrated Asset Management System. Aplikasi ini digunakan untuk manajemen aset yang ada di Minarak Brantas Gas, Inc. Namun, seiring waktu kebutuhan yang diperlukan perusahaan berubah. Minarak Brantas Gas, Inc. ingin membangun sebuah fitur baru dalam Intramas. Fitur ini ditujukan terhadap proses validasi berkas SOP.

Proses validasi berkas SOP memiliki beberapa tahap. Pertama dokumen dibuat dan diisukan untuk diperiksa. Jika terdapat kesalahan, dokumen dikembalikan untuk diperbaiki. Setelah diperbaiki, dokumen lalu diisukan lagi untuk diperiksa. Jika tidak ada kesalahan yang diperlu diperbaiki lagi, bekas kemudian akan disetujui atau ditolak oleh pihak bersangkutan. Apabila berkas disetujui, maka berkas kemudian akan diberi lembar persetujuan. Lembar ini berisikan tanda tangan persetujuan antara pekerja yang bersangkutan serta sejarah perubahan berkas SOP.

Minarak Brantas Gas, Inc. ingin membangun fitur baru dalam Intramas yang dapat membuat lembar persetujuan berkas SOP secara otomatis. Tidak hanya itu mereka ingin fitur ini juga dapat digunakan untuk tetap dapat memperbaharui berkas SOP setelah berkas tersebut disetujui. Proses validasi ini juga memiliki beberapa kasus khusus yang diperlu dipertimbangkan. Struktur hirarki pekerja Minarak Brantas Gas, Inc. di tiap departemen berbeda, sehingga proses validasi tiap departemen juga ikut berbeda. Untuk itu, MinarakBrantas Gas, Inc. ingin menambah fitur baru demi mengelola berkas-berkas SOP.



## **1.2. Tujuan**

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu juga untuk memudahkan Perusahaan Minarak Brantas Gas, Inc. dalam mengelola berkas-berkas SOP dengan membangun fitur aplikasi web E-SOP.

## **1.3. Manfaat**

Manfaat dari E-SOP adalah agar petugas Minarak Brantas Gas, Inc. dapat mengurus proses validasi berkas SOP dengan lebih mudah. Selain itu juga memudahkan petugas dalam membuat lembar persetujuan berkas SOP yang telah disetujui dan memantau sejarah penulisan, perbaikan, dan persetujuan berkas SOP.

## **1.4. Rumusan Masalah**

Berikut ini rumusan masalah pada kerja praktik pembuatan E-SOP:

1. Apa sajakah tahapan yang perlu dilalui dalam proses persetujuan berkas SOP?
2. Bagaimana cara membuat lembar persetujuan berkas SOP secara otomatis lengkap dengan pemberian tanda tangan pengesahan?
3. Bagaimana cara mengatasi kasus khusus yang terjadi akibat perbedaan struktur keanggotaan departemen dalam perusahaan?

## **1.5. Lokasi dan Waktu Kerja Praktik**

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Daring  
Alamat : Rumah masing – masing  
Waktu : 8 Maret 2021 – 31 Mei 2021

Hari Kerja : Senin – Jumat  
Jam Kerja : 08.00 WIB – 16.00 WIB (Fleksibel)

## **1.6. Metodologi Kerja Praktik**

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

### **1. Perumusan Masalah**

Untuk mengetahui permasalahan apa yang harus diselesaikan, diberikan penjelasan mengenai alasan mengapa diperlukan pembuatan sistem berbasis website *barcode* penghapusan aset ini.

Dijelaskan juga mengenai alur sistem itu akan berjalan. penjelasan mengenai hal ini dijelaskan oleh pembimbing lapangan kerja praktik. Dari penjelasannya dihasilkan catatan-catatan penting mengenai gambaran sistem berbasis web yang akan dibuatkan fitur dan query untuk mempermudah kinerjanya. Aplikasi web akan dibuat menggunakan bahasa pemrograman PHP dibantu dengan dengan *tools* pendukung Visual Studio Code dan XAMPP.

### **2. Studi Literatur**

Setelah mendapat gambaran bagaimana sistem tersebut berjalan, kami diberitahu tinjauan apa saja yang akan diimplementasikan untuk membuat website beroperasi sesuai dengan kebutuhan. Tinjauan yang dipakai meliputi PHP, XAMPP, dan lain-lain. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi sistem ini.

### **3. Analisis dan Perancangan**

Langkah ini meliputi penjelasan awal tentang sistem. Bagaimana cara kerja sistem dengan skenario tertentu. Dari

penjelasan awal telah didapatkan beberapa kebutuhan fungsional dan non-fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas spesifikasi kebutuhan-kebutuhan tersebut. Dibuatlah diagram *flowchart* kasus penggunaan yang mewakili skenario-skenario untuk penggunaan sistem aplikasi. Dilanjutkan dengan diskusi bersama pembimbing lapangan untuk mengetahui kebutuhan-kebutuhan tersebut telah tepat atau tidak.

#### **4. Implementasi Sistem**

Implementasi sistem didasarkan pada perancangan dan analisis sebelumnya. Penentuan atribut dan atau fitur, tipe data, dan format luaran yang akan dibuat pada *database* juga didasari pada analisis sebelumnya sesuai dengan kebutuhan.

Pengerjaan dilakukan dengan pelaporan setiap minggu yang dipantau oleh pembimbing lapangan dengan adanya *meeting* terencana minimal satu kali dalam satu minggu untuk mengetahui kemajuan dan kendala dari pembuatan website.

#### **5. Pengujian dan Evaluasi**

Pengujian dilakukan dengan menguji fitur-fitur yang telah dibuat. Kesesuaian sistem dengan kebutuhan akan menentukan keberhasilan dalam pengujian. Hal ini akan menghasilkan hasil evaluasi apakah sistem sudah sesuai dengan tujuan dan kebutuhan atau belum.

#### **6. Kesimpulan dan Saran**

Kesimpulan yang kami dapatkan diantaranya adalah perlunya penyesuaian atau pemahaman terhadap masalah yang dibutuhkan untuk membuat suatu fitur pada tingkat kesulitan yang berbeda-beda serta dibutuhkan komunikasi yang baik dengan *client* agar dapat mengimplementasikan fitur yang diharapkan dengan baik.

## **1.7. Sistematika Laporan**

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

### **1. Bab I Pendahuluan**

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

### **2. Bab II Profil Perusahaan**

Pada bab ini, dijelaskan secara rinci tentang profil perusahaan tempat kami melaksanakan kerja praktik, yakni Minarak Brantas Gas, Inc.

### **3. Bab III Tinjauan Pustaka**

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di Minarak Brantas Gas, Inc.

### **4. Bab IV Analisis dan Perancangan Sistem**

Pada bab ini, dijelaskan hasil pembelajaran atau analisis terhadap apa saja yang diperlukan dan harus diperhatikan dalam pengembangan aplikasi yang dikerjakan selama KP.

### **5. Bab V Implementasi Sistem**

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi pembuatan aplikasi.

### **6. Bab VI Pengujian dan Evaluasi**

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari sistem yang telah dikembangkan selama pelaksanaan kerja praktik di Minarak Brantas Gas, Inc.

### **7. Bab VII Kesimpulan dan Saran**

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **PROFIL PERUSAHAAN**

#### **2.1. Profil Minarak Brantas Gas, Inc.**

Minarak Brantas Gas, Inc. adalah salah satu perusahaan Kontraktor Kontrak Kerja Sama (KKKS) yang ditunjuk BP MIGAS untuk melakukan proses pengeboran minyak dan Gas Alam di Indonesia khususnya di wilayah kerja Blok Brantas. Dengan kantor pusat yang terletak di Bakrie Tower 20th Floor Komplek Rasuna Epicentrum JL. HR. Rasuna Said, Kuningan, Jakarta Selatan.

#### **2.2. Sejarah Singkat Minarak Brantas Gas, Inc.**

Latar belakang berdirinya Minarak Brantas Gas, Inc. adalah sebagai berikut:

1. HUFFCO menandatangani Production Sharing Contract (PSC) Blok Brantas pada tahun 1990 untuk jangka waktu 30 tahun.
2. Minarak Brantas Gas, Inc. mengambil alih Blok Brantas dari HUFFCO tahun 1996.
3. Pada tahun 1998 Minarak Brantas Gas, Inc. melakukan pengeboran dan pengembangan sumur-sumur, serta membangun stasiun produksi gas di lokasi Wunut-1, di Desa Kedungboto, Kecamatan Porong, Kabupaten Sidoarjo.
4. Minarak Brantas Gas, Inc. adalah perusahaan swasta pertama di Indonesia yang memproduksi gas di Lapangan Wunut sejak bulan Januari 1999 sebesar 4 MMSCFD.
5. Pada tahun 2004 pemboran sumur Tanggulangin 3 berhasil menemukan minyak mentah. Perjanjian pembelian gas terjadi dalam periode tahun 2004-2005 yaitu sebesar 80 MMSCFD.
6. Minarak Brantas Gas, Inc. bergabung dengan PT Energi Mega Persada (EMP) tahun 2004.

7. Lapindo melakukan kegiatan eksplorasi lepas pantai pada sumur Bisma-1 pada tahun 2005 dan berhasil menemukan gas biogenik. Studi mutakhir mengindikasikan bahwa pada struktur Bisma terdapat banyak keragaman kualitas reservoir gas, yang juga mengindikasikan target sumur delineasi yang sangat menarik di lapangan Bisma-1
8. Minarak Labuan Co. Ltd. (MLC) mengambil alih Minarak Brantas Gas, Inc. dari EMP pada tanggal 01 Juli 2007.
9. Pada tahun 2008 PGN dan Minarak Brantas Gas, Inc. kembali memperpanjang kontrak jual beli gas dengan volume 6,1 MMSCFD dan 3,5 MMSCFD di tahun 2009.
10. Di tahun 2010 dan 2011, Kontrak Jual Beli Gas dengan PGN Kembali diperpanjang dengan volume 8 MMSCFD. Saat ini gas tersebut dijual ke PT. Perusahaan Gas Negara, Tbk (Persero) dengan harga wellhead US\$ 5,15/MMBTU dengan eskalasi sebesar 1,5% per tahun.
11. Pada tanggal 26 November 2010, Minarak Brantas Gas, Inc. menandatangani Perjanjian Jual Beli Gas (PJBG) dengan PT. Petrogas Jatim Utama guna menginisiasikan program “City Gas”. PJBG tersebut merupakan tindak lanjut dari Nota Kesepahaman antara Direktorat Jenderal Minyak dan Gas Alam dan LBI tertanggal 20 Mei 2009 tentang “Pasokan Gas Untuk Rumah Tangga di Kelurahan Kali Rungkut dan Rungkut Kidul, Kecamatan Rungkut, dan wilayah sekitar Surabaya”. Program city gas sendiri merupakan upaya mendukung program pemerintah Indonesia dalam konversi minyak tanah ke gas dan masyarakat dapat memperoleh energi rumah tangga yang murah, bersih dan aman.

### **2.3. Program Kerja Minarak Brantas Gas, Inc.**

Dalam kegiatan eksplorasi dan produksinya, Minarak Brantas Gas, Inc. memiliki beberapa program kerja dan target produksi. Terdapat empat daerah operasi dari tahun 2017 sampai 2020. Pada tahun 2017 dilakukan pengolahan di daerah Wunut. Di tahun yang sama juga dilakukan



pengolahan serta servis sumur di Tanggulangin serta eksplorasi di Metro. Pada tahun berikutnya dilakukan pengembangan pengeboran di Tanggulangin dan eksplorasi di Kedaton. Pada tahun 2019, daerah Metro dan Kedaton mulai dikembangkan.

*[Halaman ini sengaja dikosongkan]*

## **BAB III**

### **TINJAUAN PUSTAKA**

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

#### **3.1. Pemrograman Web**

Pemrograman web mengacu pada penulisan, markup, dan pengkodean yang terlibat dalam pengembangan Web, yang mencakup konten Web, skrip server dan klien Web, serta keamanan jaringan. Bahasa yang paling umum digunakan untuk pemrograman Web adalah XML, HTML, JavaScript, Perl 5 dan PHP. Pemrograman web berbeda dari pemrograman biasa, yang membutuhkan pengetahuan tentang area aplikasi, skrip klien dan server, dan teknologi basis data.

#### **3.2. HTML**

HyperText Markup Language atau HTML adalah bahasa markup standar untuk dokumen yang dirancang untuk ditampilkan di browser web. HTML biasa dibantu oleh teknologi seperti Cascading Style Sheets (CSS) dan bahasa seperti JavaScript.

#### **3.3. PHP**

PHP adalah bahasa skrip untuk keperluan umum yang sangat cocok untuk pengembangan web. PHP awalnya dibuat oleh programmer Denmark-Kanada Rasmus Lerdorf pada tahun 1994. Implementasi referensi PHP sekarang diproduksi oleh The PHP Group. PHP awalnya adalah singkatan dari Personal Home Page, tetapi sekarang singkatan dari inisialisme rekursif PHP: Hypertext Preprocessor.

### **3.4. XAMPP**

XAMPP adalah paket web server *solution stack* yang bersifat open source, lintas platform, dan gratis. Paket ini dikembangkan oleh Apache Friends, yang sebagian besar terdiri dari Server HTTP Apache, basis data MariaDB, dan penerjemah untuk skrip yang ditulis dalam bahasa pemrograman PHP dan Perl. Karena sebagian besar penerapan server web yang sebenarnya menggunakan komponen yang sama dengan XAMPP, dimungkinkan transisi dari server uji lokal ke server langsung.

### **3.5. Visual Studio Code**

Visual Studio Code adalah *freeware* editor source code yang dibuat oleh Microsoft untuk Windows, Linux, dan macOS. Fitur yang tersedia meliputi bantuan untuk *debugging*, *syntax highlighting*, *intelligent code completion*, *snippet*, *code refactoring*, dan *embedded Git*. Pengguna dapat mengubah tema, shortcut keyboard, preference, dan memasang extension yang menambahkan fungsionalitas tambahan.

*[Halaman ini sengaja dikosongkan]*

## BAB IV

### ANALISIS DAN PERANCANGAN SISTEM

#### 4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun aplikasi validasi SOP Minarak Brantas Gas yaitu analisis dari sistem yang akan dibangun. Hal tersebut dijelaskan ke dalam dua bagian, definisi umum aplikasi dan analisis kebutuhan.

##### 4.1.1. Definisi Umum Aplikasi

Secara umum, aplikasi E-SOP adalah aplikasi untuk melakukan proses validasi terhadap berkas-berkas SOP yang ada di Minarak Brantas Gas. Tujuan dari pembuatan aplikasi ini adalah agar proses validasi berkas di Minarak dapat diurus secara efisien dan melakukan proses pembuatan bukti persetujuan secara otomatis. Aplikasi ini memiliki beberapa role yang disesuaikan dengan aplikasi Intramas Minarak. Role-role tersebut adalah L1, L2, L3, dan Admin.

*Tabel 4. 1 Karakteristik Pengguna*

Kategori Pengguna	Tugas	Hak Akses ke Aplikasi
L1	<ul style="list-style-type: none"><li>- Upload berkas SOP yang perlu divalidasi</li><li>- Upload ulang berkas SOP yang telah direvisi bila terdapat revisi</li><li>- Download bukti persetujuan berkas SOP.</li></ul>	Pengajuan, Daftar SOP
L2	<ul style="list-style-type: none"><li>- Memvalidasi berkas SOP</li><li>- Mengisukan revisi terhadap</li></ul>	Pengajuan, Validasi, Daftar

	berkas SOP - <i>Upload</i> berkas SOP yang perlu divalidasi apabila tidak terdapat pengguna L1 di bagian kerja departemen pengguna L2 - <i>Download</i> bukti persetujuan berkas SOP.	SOP
L3	- Memvalidasi berkas SOP - Menolak berkas SOP - Mengisukan revisi terhadap berkas SOP apabila tidak terdapat pengguna L2 di bagian kerja departemen pengguna L3 - <i>Download</i> bukti persetujuan berkas SOP.	Validasi, Daftar SOP
Admin	- Mengurus data pengguna, departemen, seksi, posisi, dan role	Admin

#### 4.1.2. Analisis Kebutuhan

Dalam aplikasi ini, terdapat fungsi-fungsi yang harus dipenuhi oleh sistem. Fungsi-fungsi yang harus dipenuhi tersebut adalah hasil diskusi antara *project manager* dengan klien. Kebutuhan ini terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non-fungsional.

##### 4.1.2.1. Kebutuhan Fungsional

Kebutuhan fungsional pada aplikasi ini menjelaskan bagaimana sistem ini bekerja yang akan dijelaskan pada Tabel 4.1

Tabel 4. 2 Kebutuhan Fungsional

<b>Kode Kebutuhan</b>	<b>Deskripsi Kebutuhan</b>	<b>Modul</b>
F01	Semua pengguna dapat melakukan Login	Modul Login
F02	Pengguna L1 dan L2 dapat melakukan <i>upload</i> berkas SOP	Modul Pengajuan
F03	Pengguna L1 dan L2 dapat memperbaiki berkas SOP yang diajukan	Modul Pengajuan
F04	Pengguna L2 dan L3 dapat mengisukan revisi untuk berkas SOP	Modul Validasi
F05	Pengguna L2 dan L3 dapat memvalidasi berkas SOP	Modul Validasi
F06	Pengguna L3 dapat menolak berkas SOP	Modul Validasi
F07	Pengguna L1, L2, dan L3 dapat melihat daftar berkas SOP yang telah divalidasi.	Modul Daftar SOP
F08	Pengguna L1, L2, dan L3 dapat mengunduh bukti persetujuan berkas SOP.	Modul Daftar SOP
F09	Pengguna L1, L2, dan L3 dapat melihat riwayat perubahan berkas SOP.	Modul Daftar SOP
F10	Pengguna L1, L2, dan L3 dapat mengubah data milik pribadi.	Modul Profil
F11	Admin dapat mengelola data web seperti pengguna,	Modul Manajemen



	departemen, role, posisi, dan seksi.	Pengguna
--	--------------------------------------	----------

#### 4.1.2.2. Kebutuhan Non-Fungsional

Tabel 4. 3 Kebutuhan Non-Fungsional

Kode Kebutuhan	Deskripsi Kebutuhan	Kualitas
NF01	Hanya akun pengguna yang teridentifikasi dan terotorisasi yang dapat menjalankan aplikasi	<i>Security</i>
NF02	Sistem dapat diakses 24 jam	<i>Portability</i>
NF03	Sistem dibuat dalam bentuk aplikasi berbasis web	<i>Portability</i>
NF04	Sistem dapat diakses pada semua <i>web-browser</i>	<i>Portability</i>

## 4.2. Perancangan Sistem

### 4.2.1. Desain Sistem

Desain sistem digunakan untuk mengetahui jalannya proses bisnis pada suatu aplikasi sehingga pengembangan dan pemeliharaan aplikasi dapat dengan mudah dilakukan. Desain sistem yang digunakan adalah *Use Case Diagram*, *Conceptual Data Model*, *Physical Data Model* dan *Activity Diagram*.

Gambar 4.1. merupakan *Use Case Diagram* yang menunjukkan proses bisnis apa saja dan siapa saja yang terlibat dalam proses tersebut. Serta *Activity Diagram* menunjukkan bagaimana sistem berinteraksi dengan *stakeholder*.

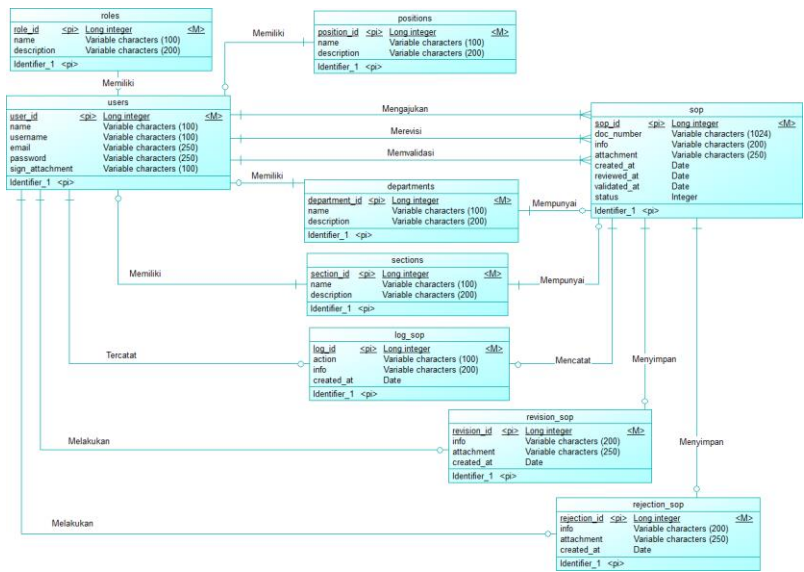


### 4.2.1.1. Use Case Diagram



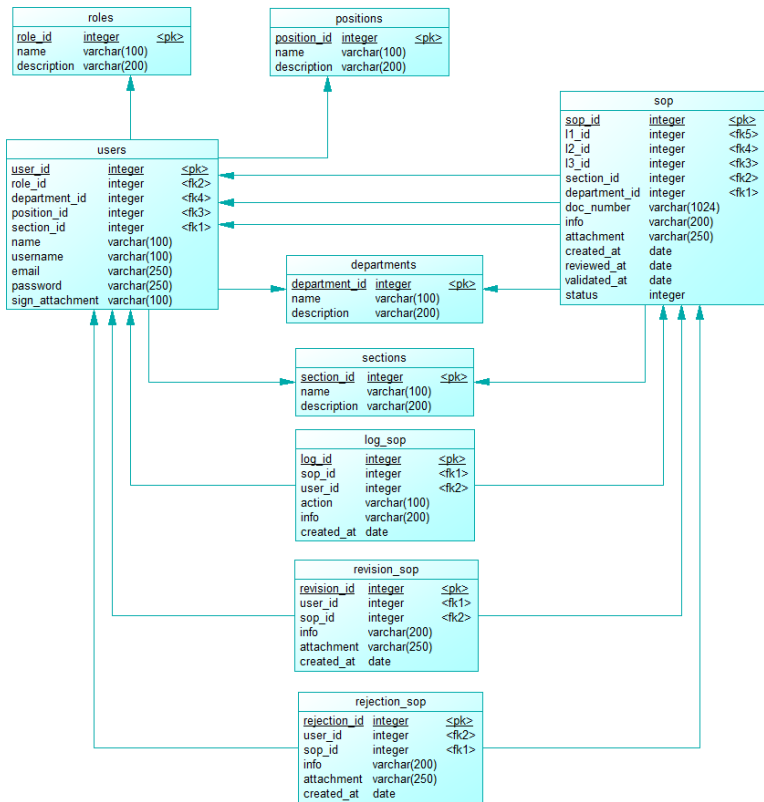
Gambar 4. 1 Use Case Diagram

### 4.2.1.2. Conceptual Data Model



Gambar 4. 2 Conceptual Data Model

### 4.2.1.3. Physical Data Model

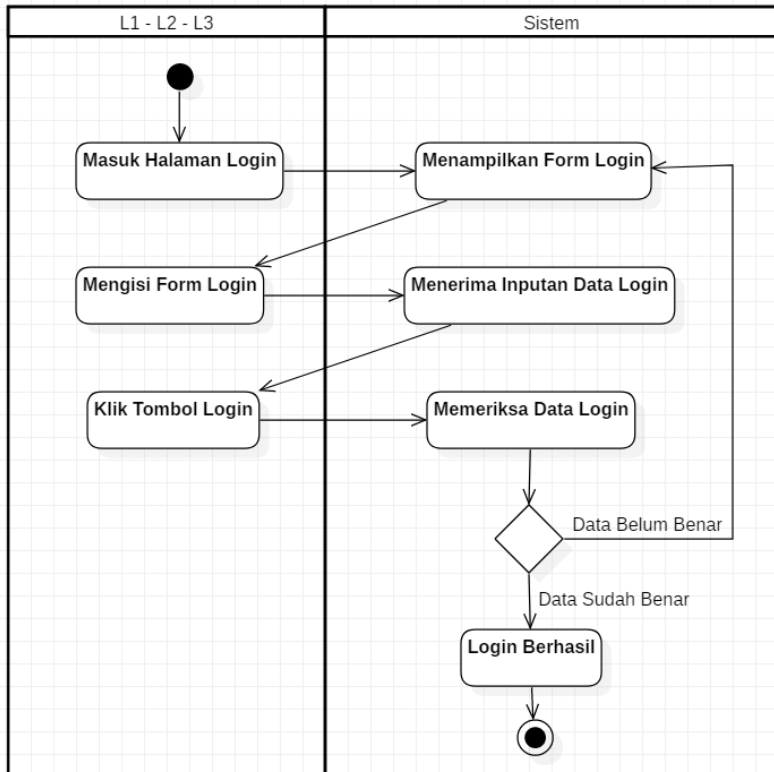


Gambar 4. 3 Physical Data Model

#### 4.2.1.4. Diagram Activity

##### F01 Melakukan Login

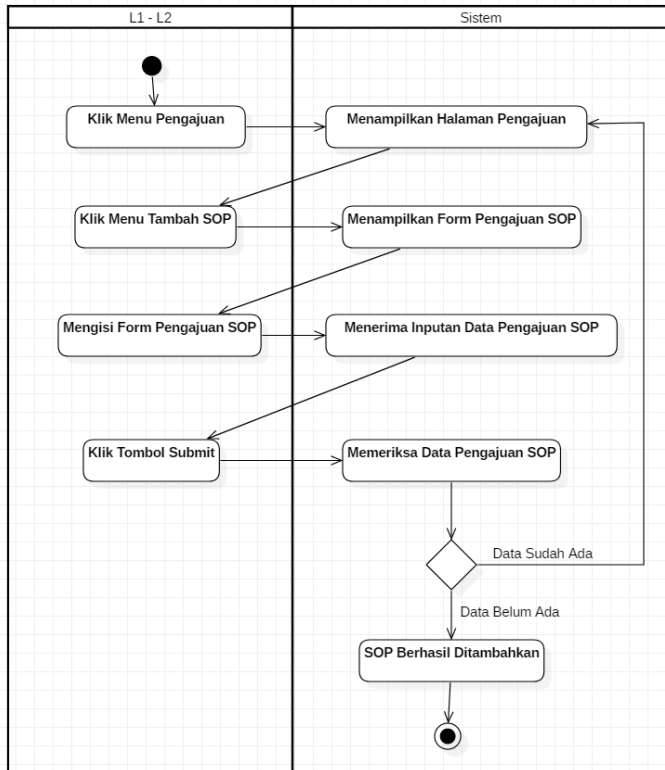
Semua pengguna melakukan *login* kedalam aplikasi untuk memastikan keaslian hak akses. Gambar 4.2. di bawah ini merupakan diagram aktivitas yang menunjukkan alur autentikasi



Gambar 4. 4 Diagram Aktivitas Melakukan Login

## F02 Mengajukan SOP

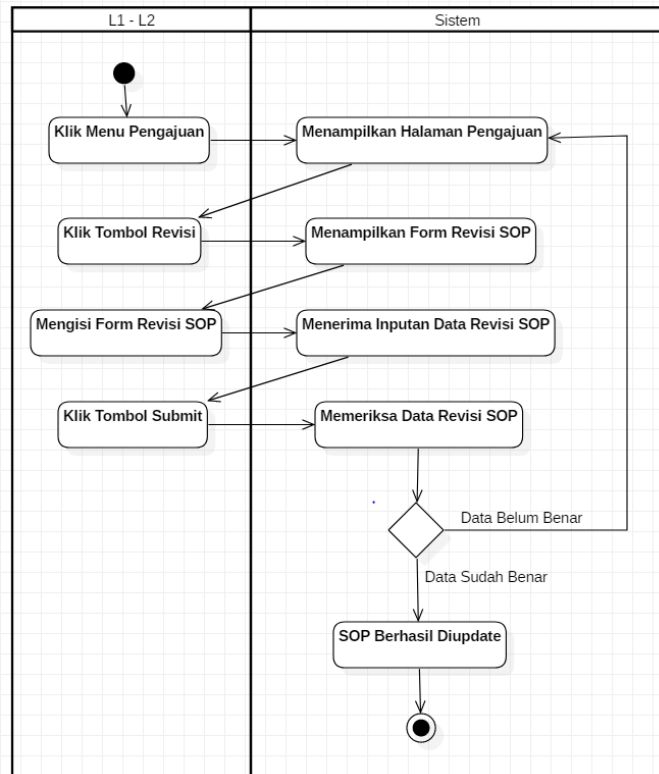
Pengguna L1 dan L2 dapat meng-upload berkas SOP yang dilakukan sesuai diagram aktivitas sebagai berikut:



Gambar 4. 5 Diagram Aktivitas Mengajukan SOP

### F03 Memperbaiki SOP

Pengguna L1 dan L2 dapat memperbaiki berkas SOP yang diajukan.

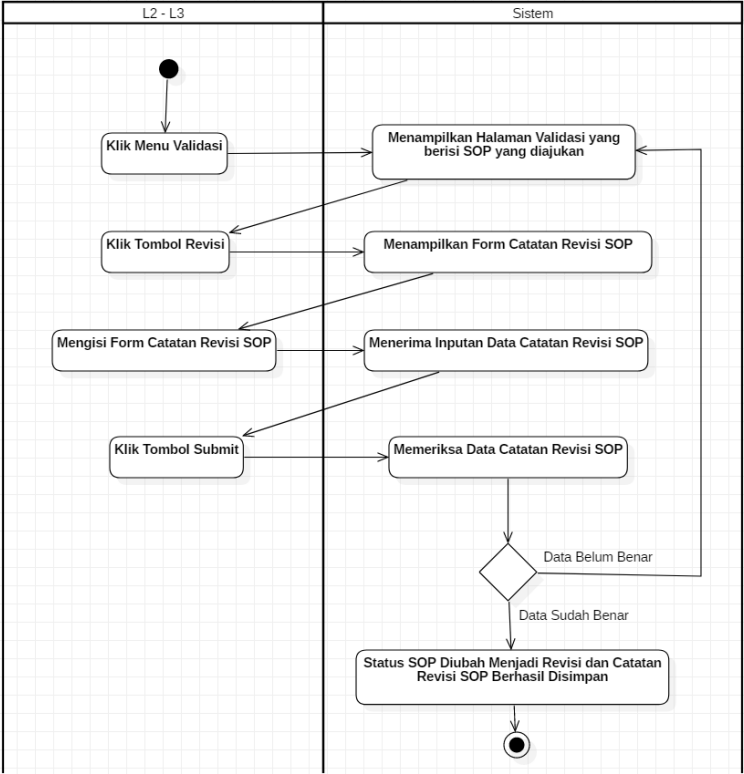


Gambar 4. 6 Diagram Aktivitas Memperbaiki SOP



**F04 Merevisi SOP**

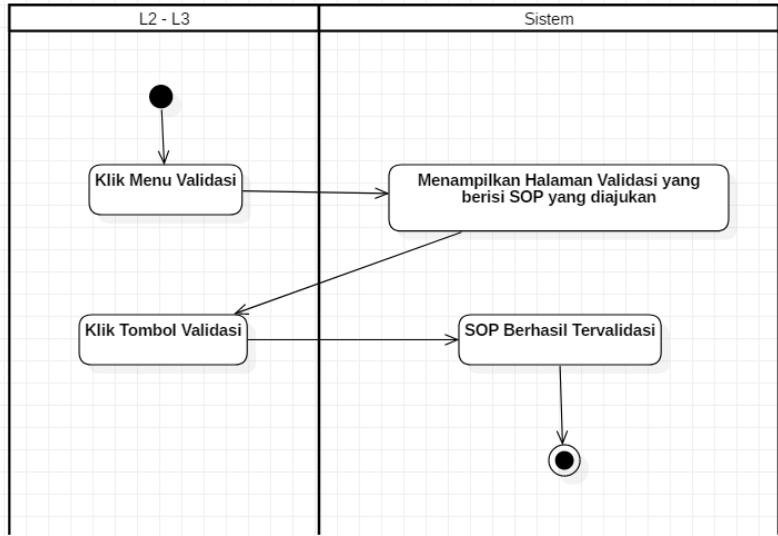
Pengguna L2 dan L3 dapat mengisukan revisi untuk berkas SOP apabila terdapat kesalahan di dalam berkas yang diajukan tersebut.



Gambar 4. 7 Diagram Aktivitas Merevisi SOP

### F05 Memvalidasi SOP

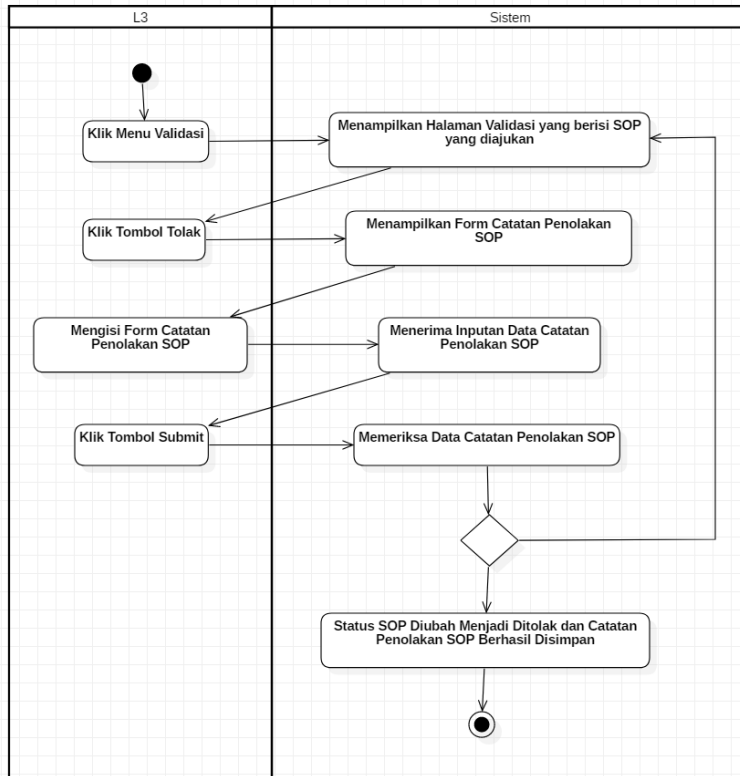
Pengguna L2 dan L3 dapat memvalidasi berkas SOP dilakukan sesuai diagram aktivitas sebagai berikut:



Gambar 4. 8 Diagram Aktivitas Memvalidasi SOP

### F06 Menolak SOP

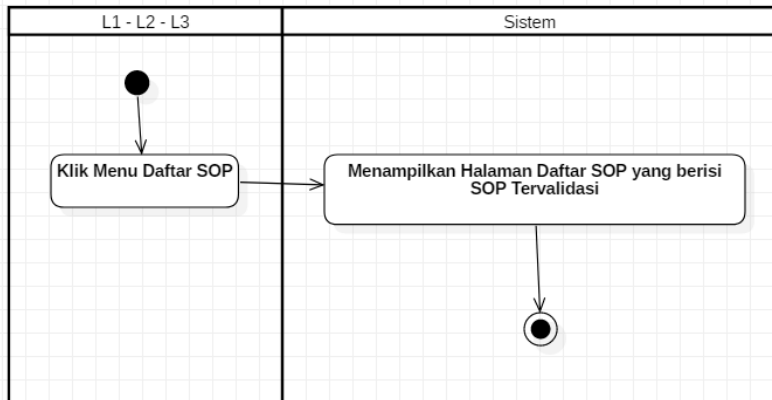
Bila berkas yang diajukan dinyatakan tidak valid, pengguna L3 dapat menolak berkas SOP yang dilakukan sesuai dengan diagram aktivitas berikut:



Gambar 4. 9 Diagram Aktivitas Menolak SOP

### F07 Melihat Daftar SOP Tervalidasi

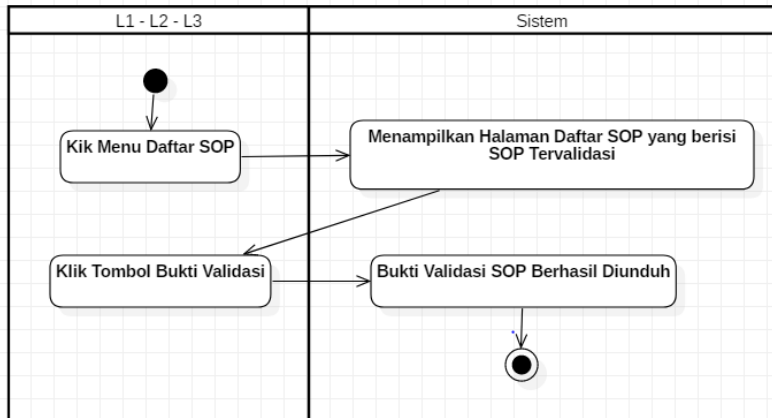
Pengguna L1, L2, dan L3 dapat melihat daftar berkas SOP yang telah divalidasi sesuai dengan diagram aktivitas sebagai berikut:



Gambar 4. 10 Diagram Aktivitas Melihat Daftar SOP Tervalidasi

### F08 Mengunduh Bukti Validasi SOP

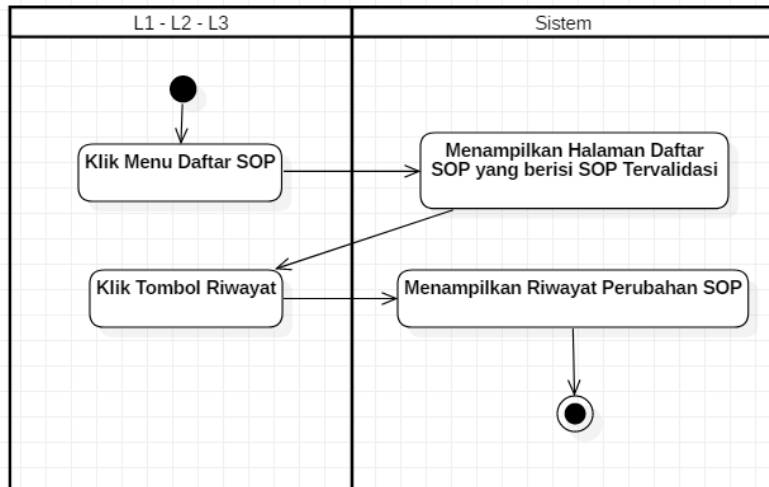
Untuk keperluan administrasi lainnya, pengguna L1, L2, dan L3 dapat mengunduh bukti persetujuan berkas SOP. Proses ini dilakukan sesuai dengan diagram aktivitas sebagai berikut:



Gambar 4. 11 Diagram Aktivitas Mengunduh Bukti Validasi SOP

### F09 Melihat Riwayat Perubahan SOP

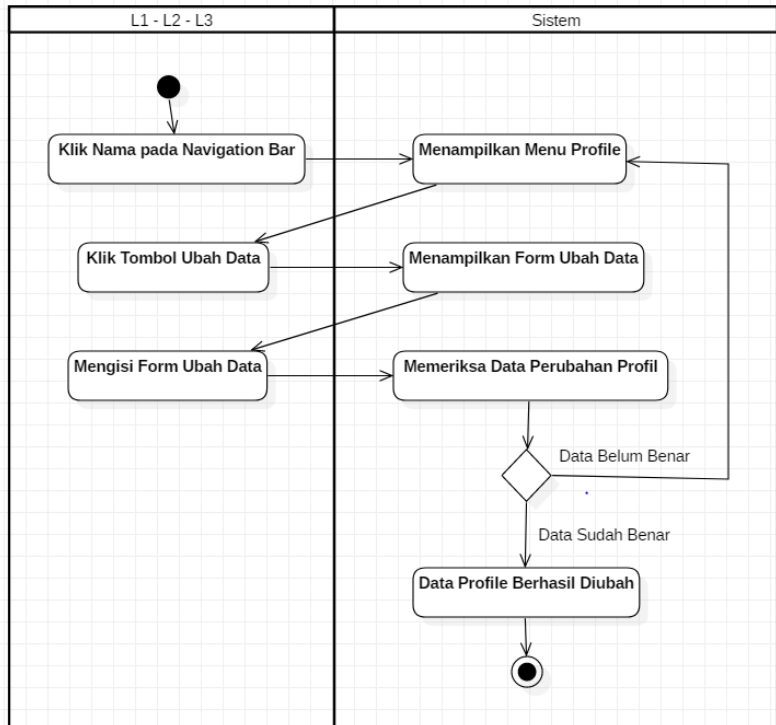
SOP secara berkala akan diisukan untuk diperbaharui jika diperlukan. Pengguna L1, L2, dan L3 dapat melihat riwayat perubahan berkas SOP sebagai referensi terhadap seberapa lama dokumen tersebut ada dan sudah berapa kali dokumen tersebut mendapat perubahan. Proses dilakukan sesuai dengan diagram aktivitas sebagai berikut:



Gambar 4. 12 Diagram Aktivitas Melihat Riwayat Perubahan SOP

### F10 Mengubah Data Diri

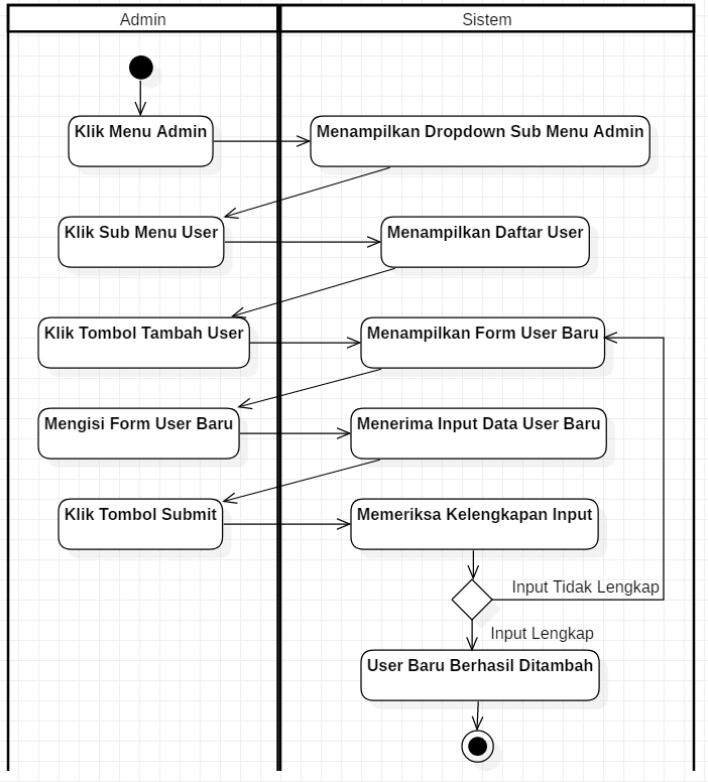
Pengguna L1, L2, dan L3 dapat mengubah data milik pribadi untuk keperluan personal. Alur untuk mengubah data diri ini sesuai dengan diagram aktivitas berikut:



Gambar 4. 13 Diagram Aktivitas Mengubah Data Diri

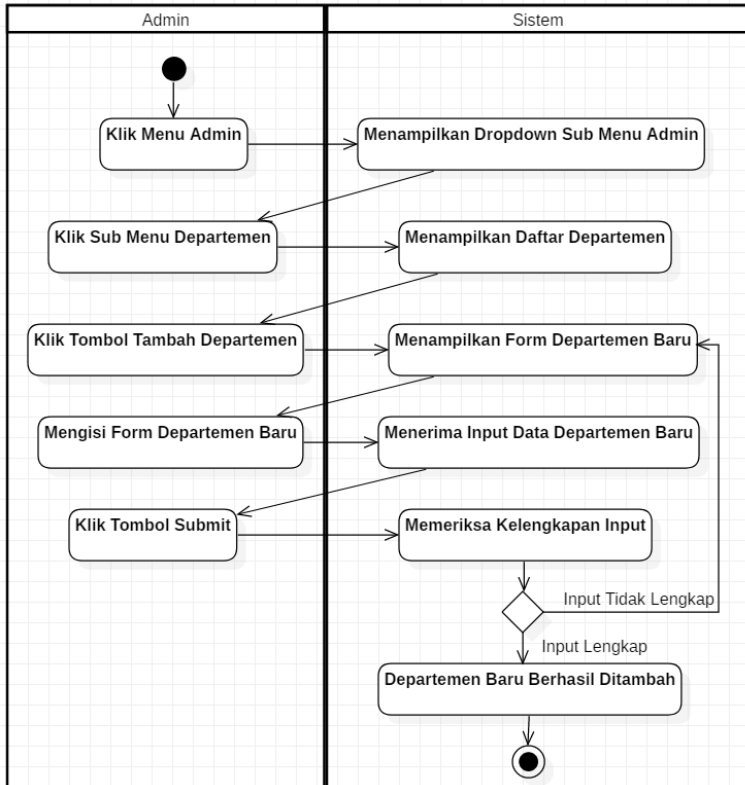
**F11 Mengelola Data Pengguna**

Admin dapat mengelola data web seperti pengguna, departemen, role, posisi, dan seksi. Terdapat beberapa proses yang dilakukan dalam pengelolaan data web. Proses-proses tersebut digambarkan dalam diagram-diagram aktivitas berikut:

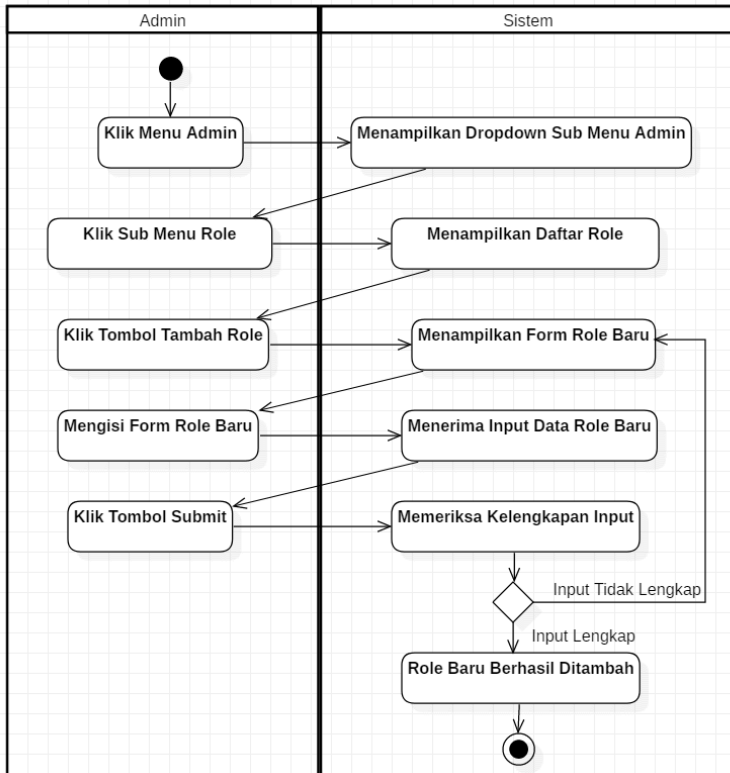


Gambar 4. 14 Diagram Aktivitas Menambah Pengguna Baru

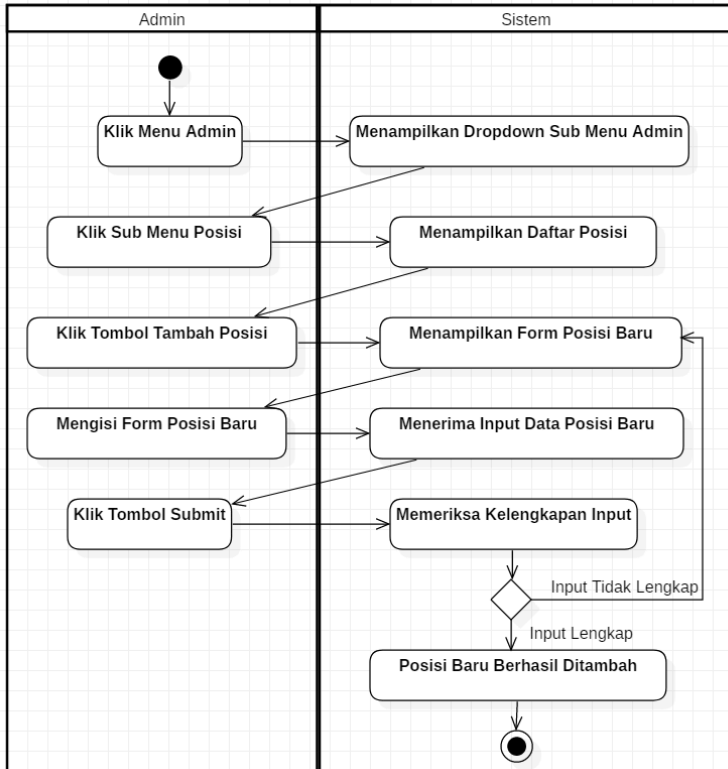




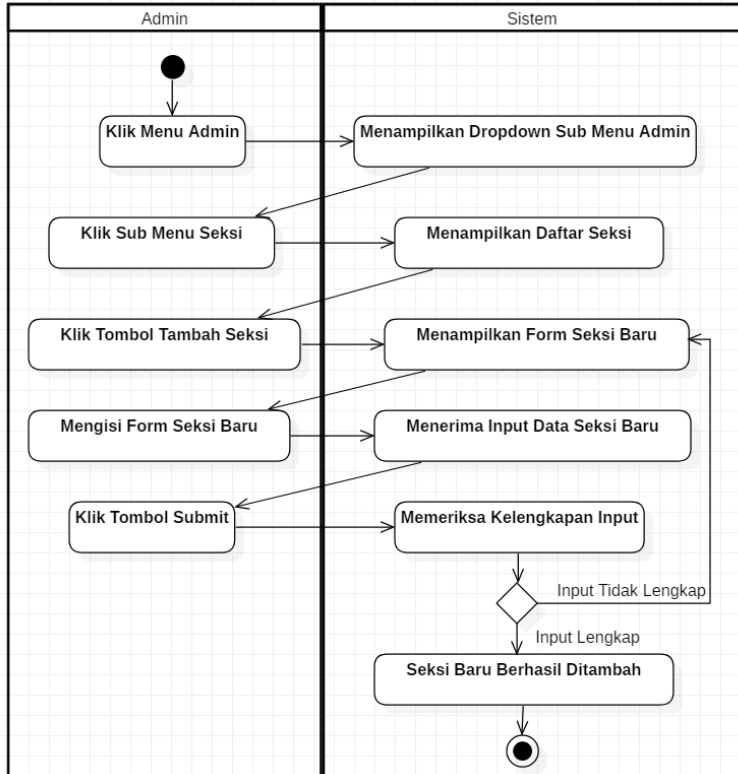
Gambar 4. 15 Diagram Aktivitas Menambah Departemen Baru



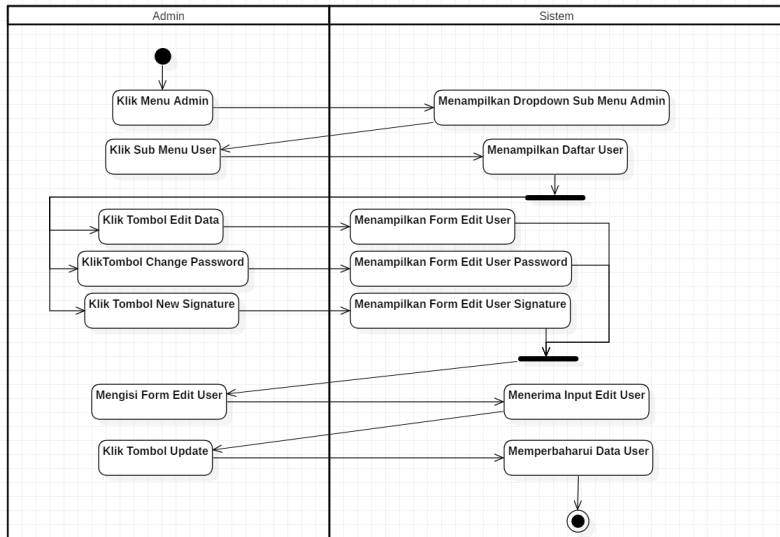
Gambar 4. 16 Diagram Aktivitas Menambah Role Baru



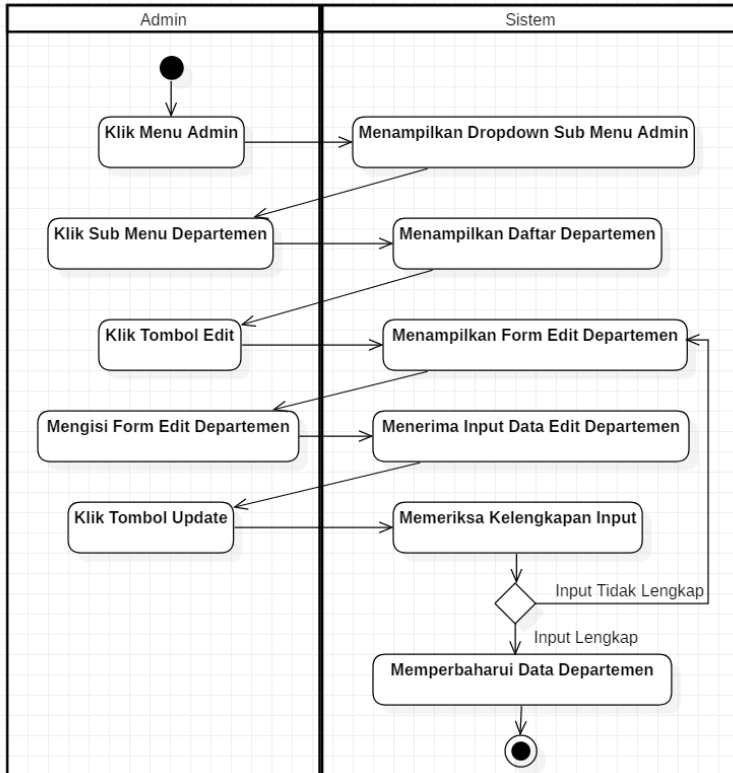
Gambar 4. 17 Diagram Aktivitas Menambah Posisi Baru



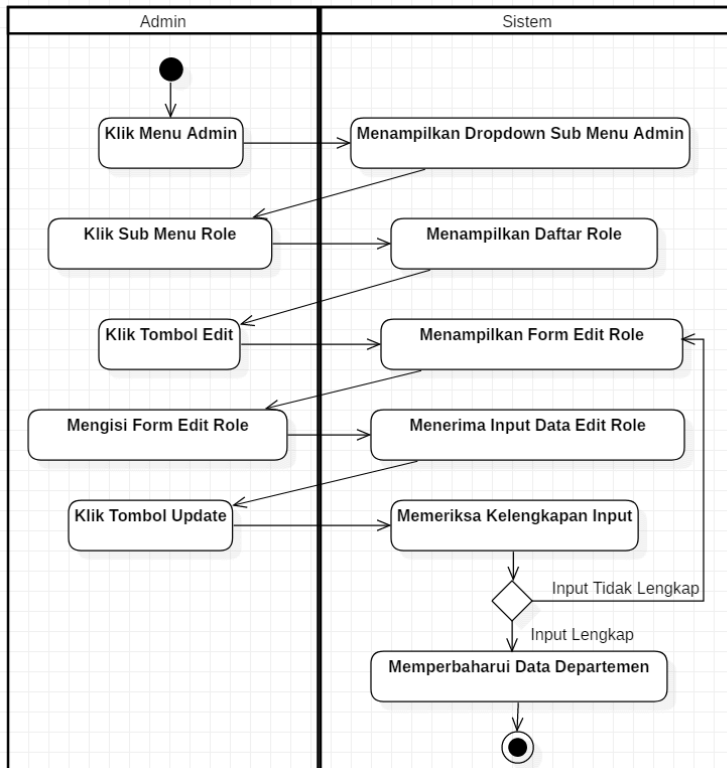
Gambar 4. 18 Diagram Aktivitas Menambah Seksi Baru



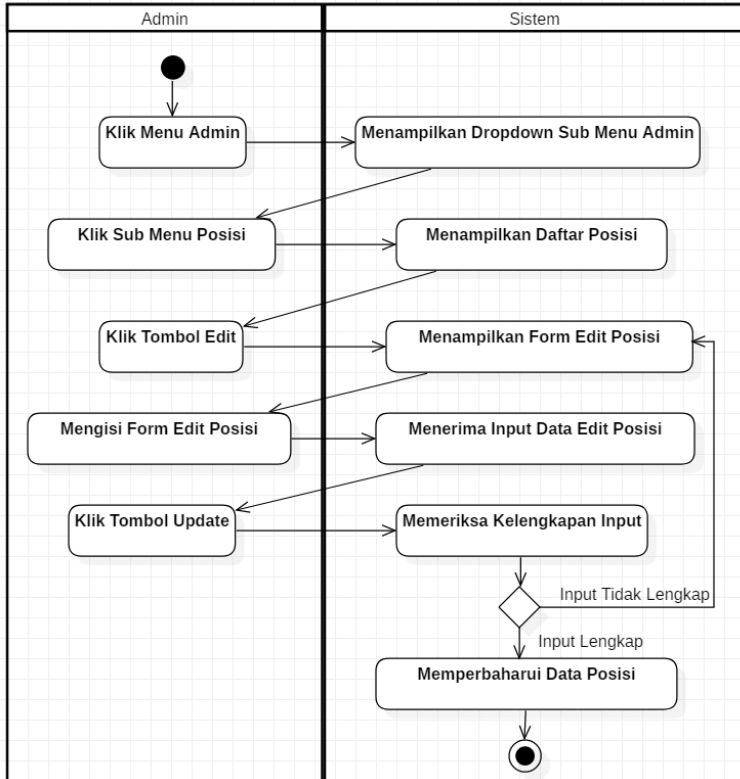
Gambar 4. 19 Diagram Aktivitas Memperbaharui Data Pengguna



Gambar 4. 20 Diagram Aktivitas Memperbaharui Data Departemen

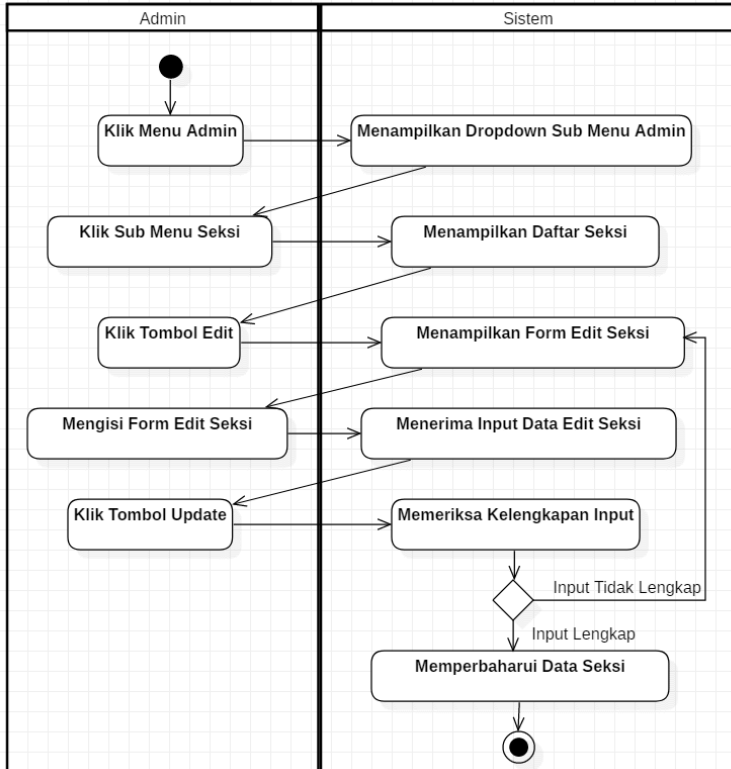


Gambar 4. 21 Diagram Aktivitas Memperbaharui Data Role

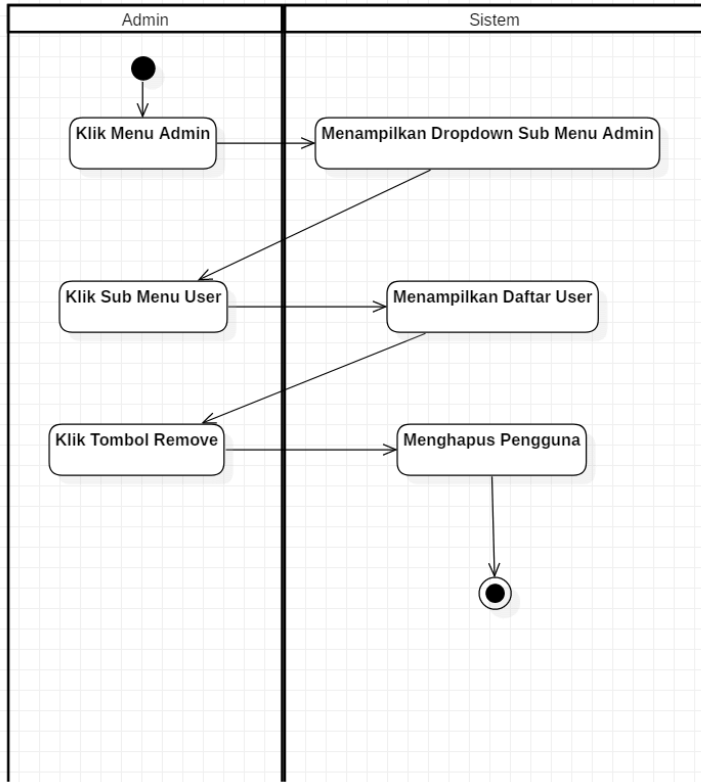


Gambar 4. 22 Diagram Aktivitas Memperbaharui Data Posisi

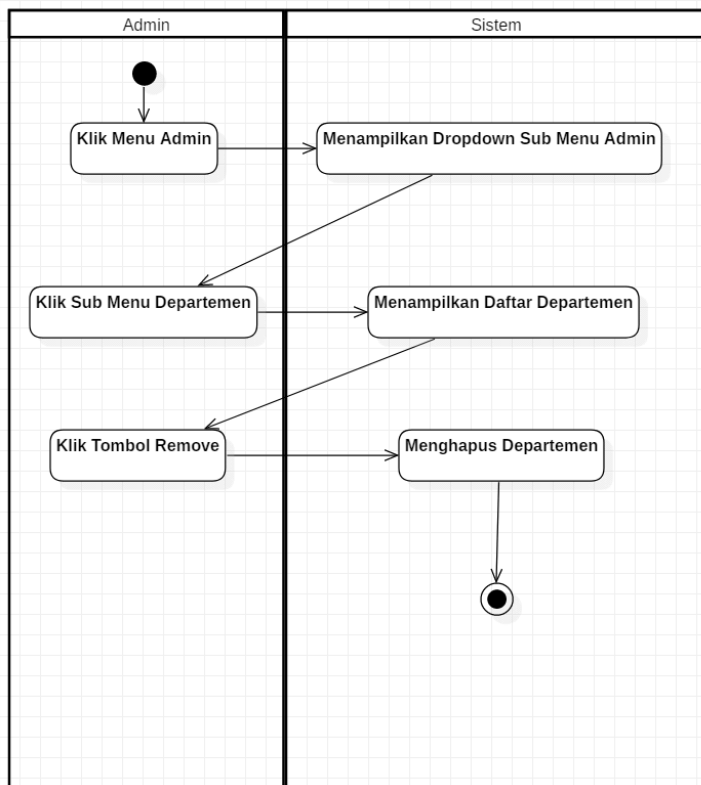




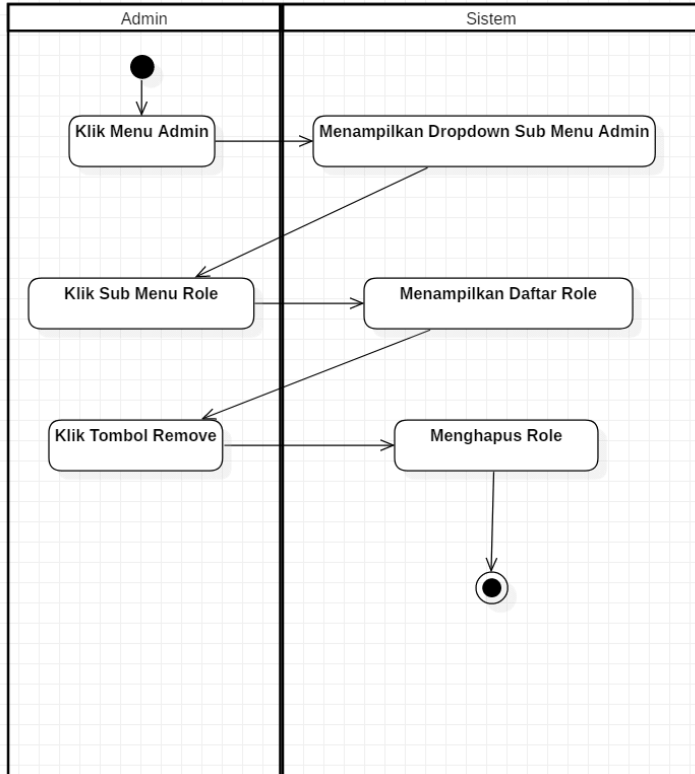
Gambar 4. 23 Diagram Aktivitas Memperbaharui Data Seksi



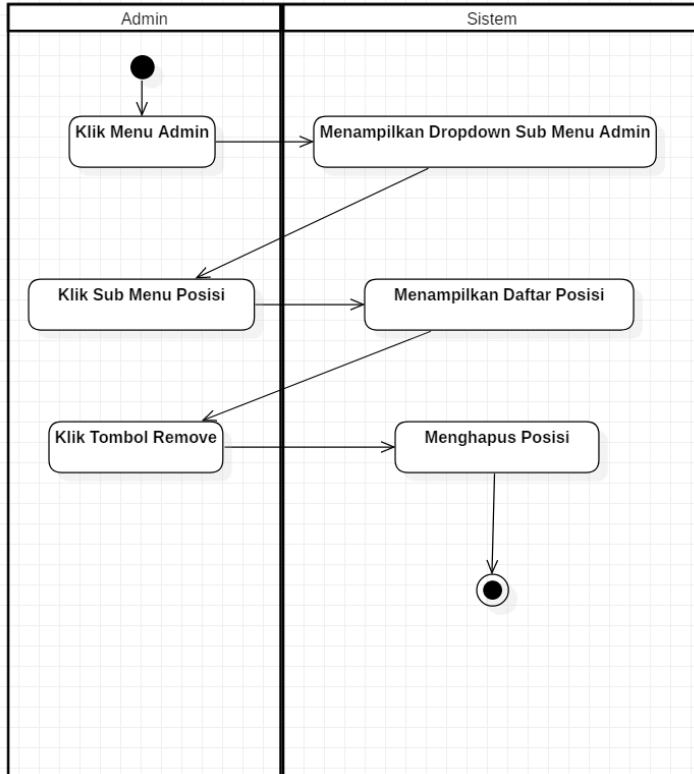
Gambar 4. 24 Diagram Aktivitas Menghapus Data Pengguna



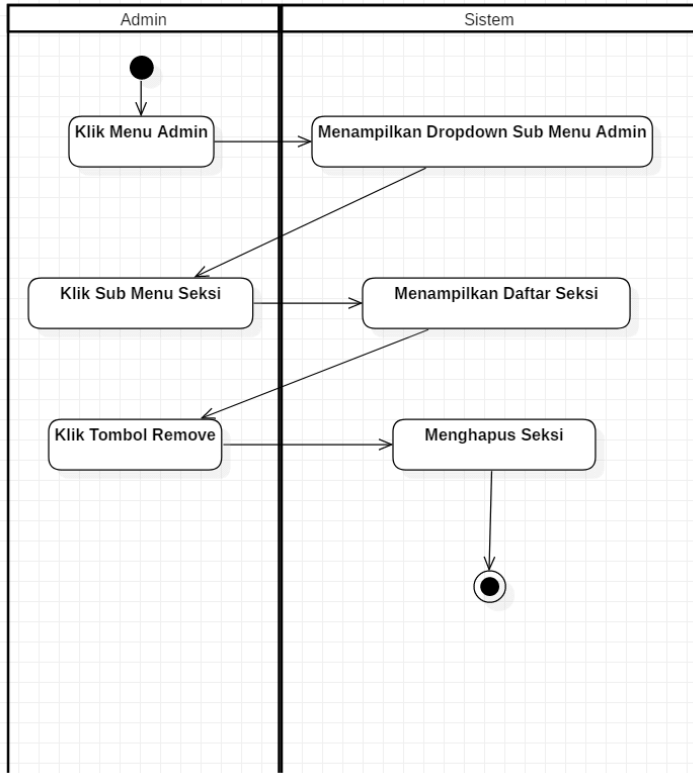
Gambar 4. 25 Diagram Aktivitas Menghapus Data Departemen



Gambar 4. 26 Diagram Aktivitas Menghapus Data Role



Gambar 4. 27 Diagram Aktivitas Menghapus Data Posisi



Gambar 4. 28 Diagram Aktivitas Menghapus Seksi

*[Halaman ini sengaja dikosongkan]*

## BAB V IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini kami menggunakan *query native* yang dibagi menjadi beberapa bagian, yakni implementasi lapisan kontrol dan implementasi *view* atau antarmuka pengguna.

### 5.1. Implementasi Controller

Implementasi lapisan kontrol ini berisi logika yang digunakan aplikasi seperti kontrol untuk memasukkan data ke *database*, menampilkan data, dan logika lainnya. Di dalam pengembangannya, kami memilih untuk memisahkan *controller* memudahkan *maintenance* dengan mempertahankan *readability*.

#### 5.1.1 Connect.php

Connect.php digunakan sebagai kontrol utama dalam mengurus perihal validasi akun *website*. Berikut adalah potongan kode dari connect.php:

```
1. <?php
2.     session_start();
3.
4.     if(isset($_SESSION["name"])) {
5.         if(isset($_SESSION["role"])) {
6.             header("Location: index.php");
7.             die();
8.         }
9.     }
10.
11.     include('config.php');
12.
13.     // Check login username and password
14.     if(isset($_POST['username'])) {
15.         $username = $_POST['username'];
16.     }
```



```

17.     else {
18.         header("Location: login.php?message=2");
19.         die();
20.     }
21.
22.     if(isset($_POST['password'])) {
23.         $password = $_POST['password'];
24.     }
25.     else {
26.         header("Location: login.php?message=2");
27.         die();
28.     }
29.
30.     $user = "SELECT users.id, users.name, users.email,
users.password, roles.id as role_id, roles.name as
role, departments.id as dept_id, departments.name as
department, sections.id as sect_id, sections.name as
section, positions.id as pos_id, positions.name as
position FROM users, roles, departments, sections,
positions WHERE users.username = '" . $username . "'
AND users.role_id = roles.id AND users.department_id
= departments.id AND users.section_id = sections.id
AND users.position_id = positions.id";
31.     $query = mysqli_query($link, $user);
32.
33.     if($row = mysqli_fetch_assoc($query)) {
34.         mysqli_free_result($query);
35.
36.         if(password_verify($password,
$row['password'])) {
37.             session_start();
38.
39.             $_SESSION["id"] = $row['id'];
40.             $_SESSION["name"] = $row['name'];
41.             $_SESSION["email"] = $row['email'];
42.             $_SESSION["role_id"] = $row['role_id'];
43.             $_SESSION["role"] = $row['role'];
44.             $_SESSION["dept_id"] = $row['dept_id'];
45.             $_SESSION["department"] =
$row['department'];
46.             $_SESSION["sect_id"] = $row['sect_id'];
47.             $_SESSION["section"] = $row['section'];
48.             $_SESSION["pos_id"] = $row['pos_id'];

```

```

49.         $_SESSION["position"] = $row['position'];
50.         $_SESSION["l1_exist"] = 1;
51.         $_SESSION["l2_exist"] = 1;
52.
53.         if($_SESSION["role_id"] == 1) {
54.             $l2_exist = "SELECT count(id) as num FROM
users WHERE users.role_id = 2 AND users.department_id
= ".$_SESSION["dept_id"]." AND users.section_id =
".$_SESSION["sect_id"];
55.
56.             $l2_exist_query = mysqli_query($link,
$l2_exist);
57.
58.             if($l2_exist_row =
mysqli_fetch_assoc($l2_exist_query)) {
59.                 mysqli_free_result($l2_exist_query);
60.
61.                 if($l2_exist_row['num'] == 0) {
62.                     $_SESSION["l2_exist"] = 0;
63.                 }
64.             }
65.         }
66.
67.         if($_SESSION["role_id"] == 2) {
68.             $l1_exist = "SELECT count(id) as num FROM
users WHERE users.role_id = 1 AND users.department_id
= ".$_SESSION["dept_id"]." AND users.section_id =
".$_SESSION["sect_id"]."";
69.
70.             $l1_exist_query = mysqli_query($link,
$l1_exist);
71.
72.             if($l1_exist_row =
mysqli_fetch_assoc($l1_exist_query)) {
73.                 mysqli_free_result($l1_exist_query);
74.
75.                 if($l1_exist_row['num'] == 0) {
76.                     $_SESSION["l1_exist"] = 0;
77.                 }
78.             }
79.         }
80.
81.         mysqli_close($link);

```

```

82.
83.     header("Location: index.php");
84.     die();
85. }
86. else {
87.     header("Location: login.php?message=2");
88.     die();
89. }
90. }
91. else {
92.     mysqli_free_result($query);
93.
94.     mysqli_close($link);
95.
96.     header("Location: login.php?message=2");
97.     die();
98. }
99. ?>

```

### 5.1.2 Department-create.php

Kontrol ini digunakan untuk membuat data baru ke tabel “Department” dalam *database*. Berikut adalah potongan kode dari department-create.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.     !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');
10.
11.     $name = "";
12.     $description = "";
13.
14.     $name = $_POST['name'];
15.     $description = $_POST['description'];
16.

```

```

17.     $department = "INSERT INTO departments VALUES ('',
        '$name.', '$description.')";
18.
19.     if($query = mysqli_query($link, $department)) {
20.         mysqli_free_result($query);
21.
22.         mysqli_close($link);
23.
24.         header("Location: department.php?message=1");
25.         die();
26.     }
27.     else {
28.         mysqli_free_result($query);
29.
30.         mysqli_close($link);
31.
32.         header("Location: department.php?message=2");
33.         die();
34.     }
35. ?>

```

### 5.1.3 Department-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “Department” di *database*. Berikut adalah potongan kode dari department-update.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
        !isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $name = "";
12.     $description = "";
13.
14.     $departmentid = $_POST['id'];

```

```

15.  $name = $_POST['name'];
16.  $description = $_POST['description'];
17.
18.  $department = "UPDATE departments SET name =
    '". $name."', description = '". $description.'" WHERE
    id = '". $departmentid.'"";
19.
20.  if($query = mysqli_query($link, $department)) {
21.      mysqli_free_result($query);
22.
23.      mysqli_close($link);
24.
25.      header("Location: department.php?message=1");
26.      die();
27.  }
28.  else {
29.      mysqli_free_result($query);
30.
31.      mysqli_close($link);
32.
33.      header("Location: department.php?message=2");
34.      die();
35.  }
36.  ?>

```

### 5.1.4 Department-delete.php

Kontrol ini digunakan untuk menghapus data yang ada pada tabel “Department” di *database*. Berikut adalah potongan kode dari `department-delete.php`:

```

1.  <?php
2.  session_start();
3.
4.  if(!isset($_SESSION["name"]) ||
    !isset($_SESSION["role"])) {
5.  header("Location: logout.php");
6.  die();
7.  }
8.
9.  include('config.php');
10.

```

```

11. $departmentid = $_GET['departmentid'];
12.
13. $department = "DELETE FROM departments WHERE id =
    ".$departmentid.";
14.
15. if($query = mysqli_query($link, $department)) {
16.     mysqli_free_result($query);
17.
18.     mysqli_close($link);
19.
20.     header("Location: department.php?message=1");
21.     die();
22. }
23. else {
24.     mysqli_free_result($query);
25.
26.     mysqli_close($link);
27.
28.     header("Location: department.php?message=2");
29.     die();
30. }
31. ?>

```

### 5.1.5 Position-create.php

Kontrol ini digunakan untuk membuat data baru ke tabel “Position” dalam *database*. Berikut adalah potongan kode dari position-create.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.     !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');
10.
11.     $name = "";
12.     $description = "";
13.

```

```

14.  $name = $_POST['name'];
15.  $description = $_POST['description'];
16.
17.  $position = "INSERT INTO positions VALUES (',
    '". $name."', '". $description."')";
18.
19.  if($query = mysqli_query($link, $position)) {
20.      mysqli_free_result($query);
21.
22.      mysqli_close($link);
23.
24.      header("Location: position.php?message=1");
25.      die();
26.  }
27.  else {
28.      mysqli_free_result($query);
29.
30.      mysqli_close($link);
31.
32.      header("Location: position.php?message=2");
33.      die();
34.  }
35. ?>

```

### 5.1.6 Position-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “Position” di *database*. Berikut adalah potongan kode dari position-update.php:

```

1.  <?php
2.      session_start();
3.
4.      if(!isset($_SESSION["name"]) ||
    !isset($_SESSION["role"])) {
5.          header("Location: logout.php");
6.          die();
7.      }
8.
9.      include('config.php');
10.
11.     $name = "";

```

```

12.  $description = "";
13.
14.  $positionid = $_POST['id'];
15.  $name = $_POST['name'];
16.  $description = $_POST['description'];
17.
18.  $position = "UPDATE positions SET name =
    '$name.', description = '$description.'" WHERE
    id = ".$positionid."";
19.
20.  if($query = mysqli_query($link, $position)) {
21.      mysqli_free_result($query);
22.
23.      mysqli_close($link);
24.
25.      header("Location: position.php?message=1");
26.      die();
27.  }
28.  else {
29.      mysqli_free_result($query);
30.
31.      mysqli_close($link);
32.
33.      header("Location: position.php?message=2");
34.      die();
35.  }
36. ?>

```

### 5.1.7 Position-delete.php

Kontrol ini digunakan untuk menghapus data yang ada pada tabel “Position” di *database*. Berikut adalah potongan kode dari position-delete.php:

```

1.  <?php
2.      session_start();
3.
4.      if(!isset($_SESSION["name"]) ||
    !isset($_SESSION["role"])) {
5.          header("Location: logout.php");
6.          die();
7.      }

```



```

8.
9.     include('config.php');
10.
11.     $positionid = $_GET['positionid'];
12.
13.     $position = "DELETE FROM positions WHERE id =
14.     ".$positionid."";
15.     if($query = mysqli_query($link, $position)) {
16.         mysqli_free_result($query);
17.
18.         mysqli_close($link);
19.
20.         header("Location: position.php?message=1");
21.         die();
22.     }
23.     else {
24.         mysqli_free_result($query);
25.
26.         mysqli_close($link);
27.
28.         header("Location: position.php?message=2");
29.         die();
30.     }
31. ?>

```

### 5.1.8 Role-create.php

Kontrol ini digunakan untuk membuat data baru ke tabel “Role” dalam *database*. Berikut adalah potongan kode dari role-create.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.     !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');

```

```

10.
11.     $name = "";
12.     $description = "";
13.
14.     $name = $_POST['name'];
15.     $description = $_POST['description'];
16.
17.     $role = "INSERT INTO roles VALUES ('',
18.     '". $name."', '". $description."')";
19.
20.     if($query = mysqli_query($link, $role)) {
21.         mysqli_free_result($query);
22.
23.         mysqli_close($link);
24.
25.         header("Location: role.php?message=1");
26.         die();
27.     }
28.     else {
29.         mysqli_free_result($query);
30.
31.         mysqli_close($link);
32.
33.         header("Location: role.php?message=2");
34.         die();
35.     }
36. }>

```

### 5.1.9 Role-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “Role” di *database*. Berikut adalah potongan kode dari role-update.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.     !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }

```

```

8.
9.     include('config.php');
10.
11.     $name = "";
12.     $description = "";
13.
14.     $roleid = $_POST['id'];
15.     $name = $_POST['name'];
16.     $description = $_POST['description'];
17.
18.     $role = "UPDATE roles SET name = '". $name."',
description = '". $description.'" WHERE id =
"$. $roleid."";
19.
20.     if($query = mysqli_query($link, $role)) {
21.         mysqli_free_result($query);
22.
23.         mysqli_close($link);
24.
25.         header("Location: role.php?message=1");
26.         die();
27.     }
28.     else {
29.         mysqli_free_result($query);
30.
31.         mysqli_close($link);
32.
33.         header("Location: role.php?message=2");
34.         die();
35.     }
36. ?>

```

### 5.1.10 Role-delete.php

Kontrol ini digunakan untuk menghapus data yang ada pada tabel “Role” di *database*. Berikut adalah potongan kode dari role-delete.php:

```

1. <?php
2.     session_start();
3.

```

```

4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.    $roleid = $_GET['roleid'];
12.
13.    $role = "DELETE FROM roles WHERE id =
". $roleid. " ";
14.
15.    if($query = mysqli_query($link, $role)) {
16.        mysqli_free_result($query);
17.
18.        mysqli_close($link);
19.
20.        header("Location: role.php?message=1");
21.        die();
22.    }
23.    else {
24.        mysqli_free_result($query);
25.
26.        mysqli_close($link);
27.
28.        header("Location: role.php?message=2");
29.        die();
30.    }
31. ?>

```

### 5.1.11 Section-create.php

Kontrol ini digunakan untuk membuat data baru ke tabel “Section” dalam *database*. Berikut adalah potongan kode dari section-create.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {

```

```

5.     header("Location: logout.php");
6.     die();
7. }
8.
9.     include('config.php');
10.
11.    $name = "";
12.    $description = "";
13.
14.    $name = $_POST['name'];
15.    $description = $_POST['description'];
16.
17.    $section = "INSERT INTO sections VALUES ('',
18.    '$name.', '$description.')";
19.
19.    if($query = mysqli_query($link, $section)) {
20.        mysqli_free_result($query);
21.
22.        mysqli_close($link);
23.
24.        header("Location: section.php?message=1");
25.        die();
26.    }
27.    else {
28.        mysqli_free_result($query);
29.
30.        mysqli_close($link);
31.
32.        header("Location: section.php?message=2");
33.        die();
34.    }
35. ?>

```

### 5.1.12 Section-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “Section” di *database*. Berikut adalah potongan kode dari section-update.php:

```

1. <?php
2.     session_start();
3.

```

```

4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.    $name = "";
12.    $description = "";
13.
14.    $sectionid = $_POST['id'];
15.    $name = $_POST['name'];
16.    $description = $_POST['description'];
17.
18.    $section = "UPDATE sections SET name =
'".$name."', description = '".$description.'" WHERE
id = ".$sectionid.";
19.
20.    if($query = mysqli_query($link, $section)) {
21.        mysqli_free_result($query);
22.
23.        mysqli_close($link);
24.
25.        header("Location: section.php?message=1");
26.        die();
27.    }
28.    else {
29.        mysqli_free_result($query);
30.
31.        mysqli_close($link);
32.
33.        header("Location: section.php?message=2");
34.        die();
35.    }
36. ?>

```

### 5.1.13 Section-delete.php

Kontrol ini digunakan untuk menghapus data yang ada pada tabel “Section” di *database*. Berikut adalah potongan kode dari section-delete.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $sectionid = $_GET['sectionid'];
12.
13.     $section = "DELETE FROM sections WHERE id =
". $sectionid. " ";
14.
15.     if($query = mysqli_query($link, $section)) {
16.         mysqli_free_result($query);
17.
18.         mysqli_close($link);
19.
20.         header("Location: section.php?message=1");
21.         die();
22.     }
23.     else {
24.         mysqli_free_result($query);
25.
26.         mysqli_close($link);
27.
28.         header("Location: section.php?message=2");
29.         die();
30.     }
31. ?>

```

#### 5.1.14 User-create.php

Kontrol ini digunakan untuk membuat data baru ke tabel “User” dalam *database*. Berikut adalah potongan kode dari user-create.php:

```

1. <?php

```

```

2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.     $password = "";
14.     $departmentId = "";
15.     $sectionId = "";
16.     $positionId = "";
17.     $roleId = "";
18.
19.     $name = $_POST['name'];
20.     $email = $_POST['email'];
21.     $password = password_hash($_POST['password'],
PASSWORD_DEFAULT);
22.     $departmentId = $_POST['departmentId'];
23.     $sectionId = $_POST['sectionId'];
24.     $positionId = $_POST['positionId'];
25.     $roleId = $_POST['roleId'];
26.
27.     $targetDir = "storage/signature/";
28.     $file = $_FILES['signature']['name'];
29.     $temp_name = $_FILES['signature']['tmp_name'];
30.     $sign_attachment = $targetDir.$name.'_'.$file;
31.
32.     if(file_exists($sign_attachment)) {
33.         header("Location: user.php?message=2");
34.         die();
35.     }
36.     else {
37.         move_uploaded_file($temp_name,
$sign_attachment);
38.
39.         $user = "INSERT INTO users VALUES (',
'".$name."', '".$email."', '".$password."',
'".$sign_attachment."', '".$roleId."',

```



```

        ".$departmentId."', ".$sectionId."',
        ".$positionId."'");
40.
41.     if($query = mysqli_query($link, $user)) {
42.         mysqli_free_result($query);
43.
44.         mysqli_close($link);
45.
46.         header("Location: user.php?message=1");
47.         die();
48.     }
49.     else {
50.         mysqli_free_result($query);
51.
52.         mysqli_close($link);
53.
54.         header("Location: user.php?message=2");
55.         die();
56.     }
57. }
58. ?>

```

### 5.1.15 User-data-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “User” di *database*. Berikut adalah potongan kode dari user-data-update.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.     $departmentId = "";

```

```

14. $sectionId = "";
15. $positionId = "";
16. $roleId = "";
17.
18. $userid = $_POST['id'];
19. $name = $_POST['name'];
20. $email = $_POST['email'];
21. $departmentId = $_POST['departmentId'];
22. $sectionId = $_POST['sectionId'];
23. $positionId = $_POST['positionId'];
24. $roleId = $_POST['roleId'];
25.
26. $user = "UPDATE users SET name = '$name.',
email = '$email.', role_id = '$roleId.',
department_id = '$departmentId.', section_id =
'$sectionId.', position_id = '$positionId.'"
WHERE id = ".$userid."";
27.
28. if($query = mysqli_query($link, $user)) {
29.     mysqli_free_result($query);
30.
31.     mysqli_close($link);
32.
33.     header("Location: user.php?message=1");
34.     die();
35. }
36. else {
37.     mysqli_free_result($query);
38.
39.     mysqli_close($link);
40.
41.     header("Location: user.php?message=2");
42.     die();
43. }
44.
45. ?>

```

### 5.1.16 User-pass-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “User” di *database*. Data yang diperbaharui hanya dikhususkan terhadap satu atribut tabel yaitu

“password”. Berikut adalah potongan kode dari user-pass-update.php:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.     $password = "";
14.
15.     $userid = $_POST['id'];
16.     $name = $_POST['name'];
17.     $email = $_POST['email'];
18.     $password = password_hash($_POST['password'],
PASSWORD_DEFAULT);
19.
20.     $user = "UPDATE users SET password =
'".$password."' WHERE id = ".$userid.";";
21.
22.     if($query = mysqli_query($link, $user)) {
23.         mysqli_free_result($query);
24.
25.         mysqli_close($link);
26.
27.         header("Location: user.php?message=1");
28.         die();
29.     }
30.     else {
31.         mysqli_free_result($query);
32.
33.         mysqli_close($link);
34.
35.         header("Location: user.php?message=2");
36.         die();
```

```
37.     }
38.
39. ?>
```

### 5.1.17 User-sign-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “User” di *database*. Data yang diperbaharui hanya dikhususkan terhadap satu atribut tabel yaitu “sign\_attachment”. Berikut adalah potongan kode dari user-sign-update.php:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"])           ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.
14.     $userid = $_POST['id'];
15.     $name = $_POST['name'];
16.     $email = $_POST['email'];
17.
18.     $targetDir = "storage/signature/";
19.     $file = $_FILES['signature']['name'];
20.     $temp_name = $_FILES['signature']['tmp_name'];
21.     $lampiran = $targetDir.$name.'_'.$file;
22.
23.     $old_path = "SELECT users.sign_attachment FROM
users WHERE id = ".$userid."";
24.     $old_path_query = mysqli_query($link, $old_path);
25.     $old_signature;
26.     if($old_signature =
mysqli_fetch_assoc($old_path_query)) {
```

```

27.     if(file_exists($old_signature['sign_attachment']))
28.         unlink($old_signature['sign_attachment']);
29.     }
30.
31.     if(file_exists($lampiran)) {
32.         header("Location: user.php?message=2");
33.         die();
34.     }
35.     else{
36.         move_uploaded_file($temp_name, $lampiran);
37.
38.         $user = "UPDATE users SET sign_attachment =
'".$lampiran."' WHERE id = ".$userid.";
39.
40.         if($query = mysqli_query($link, $user)) {
41.             mysqli_free_result($query);
42.
43.             mysqli_close($link);
44.
45.             header("Location: user.php?message=1");
46.             die();
47.         }
48.         else {
49.             mysqli_free_result($query);
50.
51.             mysqli_close($link);
52.
53.             header("Location: user.php?message=2");
54.             die();
55.         }
56.     }
57. ?>

```

### 5.1.18 User-delete.php

Kontrol ini digunakan untuk menghapus data yang ada pada tabel “User” di *database*. Berikut adalah potongan kode dari user-delete.php:

```

1. <?php
2.     session_start();

```

```

3.
4.     if(!isset($_SESSION["name"])           ||
      !isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $userid = $_GET['userid'];
12.
13.     $old_path = "SELECT users.sign_attachment FROM
      users WHERE id = ".$userid."";
14.     $old_path_query = mysqli_query($link, $old_path);
15.     $old_signature;
16.     if($old_signature =
      mysqli_fetch_assoc($old_path_query)) {
17.
18.         if(file_exists($old_signature['sign_attachment']))
19.             unlink($old_signature['sign_attachment']);
20.     }
21.     $user = "DELETE FROM users WHERE id =
      ".$userid."";
22.
23.     if($query = mysqli_query($link, $user)) {
24.         mysqli_free_result($query);
25.
26.         mysqli_close($link);
27.
28.         header("Location: user.php?message=1");
29.         die();
30.     }
31.     else {
32.         mysqli_free_result($query);
33.
34.         mysqli_close($link);
35.
36.         header("Location: user.php?message=2");
37.         die();
38.     }
39. ?>

```

### 5.1.19 Listsop-renew.php

Kontrol ini digunakan untuk memperbaharui berkas SOP yang telah divalidasi oleh pihak-pihak berwenang. Berkas SOP akan dikembalikan ke status revisi. Berikut adalah potongan kode dari listsop-renew.php:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
    isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $sopid = $_POST['sopid'];
12.     $id = $_POST['id'];
13.     $revision = $_POST['revision'];
14.
15.     $targetDir = "storage/revision/";
16.     $file = $_FILES['docfile']['name'];
17.     $temp_name = $_FILES['docfile']['tmp_name'];
18.     $lampiran = $targetDir.$id.'_'.$sopid.'_'.$file;
19.
20.     $old_path = "SELECT revision_sop.attachment FROM
    revision_sop WHERE sop_id = ".$sopid." ORDER BY
    revision_sop.created_at DESC LIMIT 1";
21.     $old_path_query = mysqli_query($link, $old_path);
22.     $old_revision;
23.     if($old_revision =
    mysqli_fetch_assoc($old_path_query)) {
24.         if(file_exists($old_revision['attachment']))
25.             unlink($old_revision['attachment']);
26.     }
27.
28.     if(file_exists($lampiran)) {
29.         header("Location: listsop.php?message=2");
30.         die();
```

```

31.     }
32.     else {
33.         move_uploaded_file($temp_name, $lampiran);
34.
35.         $sop = "UPDATE sop SET sop.status = 4,
sop.l3_id = 0 WHERE sop.id = ".$sopid;
36.         $sopRevision = "INSERT INTO revision_sop VALUES
('', ".$sopid.", ".$id.", ".$revision.",
".$lampiran.", ".date("Y-m-d H:i:s").")";
37.         $revisionLog = "INSERT INTO log_sop VALUES ('',
".$sopid.", 'Correction', '', ".$id.",
".$date("Y-m-d").")";
38.
39.         if($query = mysqli_query($link, $sop)) {
40.             if($query = mysqli_query($link,
$sopRevision)) {
41.                 if($query = mysqli_query($link,
$revisionLog)) {
42.                     mysqli_free_result($query);
43.
44.                     mysqli_close($link);
45.
46.                     header("Location:
listsop.php?message=1");
47.                     die();
48.                 }
49.             }
50.         }
51.
52.         mysqli_close($link);
53.
54.         header("Location: listsop.php?message=2");
55.         die();
56.     }
57. ?>

```

### 5.1.20 Profile-data-update

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “User” di *database*. Kontrol ini lebih khusus mengatur data milik pengguna yang sedang mengakses



*website*. Berikut adalah potongan kode dari *profile-data-update.php*:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.     $departmentId = "";
14.     $sectionId = "";
15.     $positionId = "";
16.     $roleId = "";
17.
18.     $userid = $_POST['id'];
19.     $name = $_POST['name'];
20.     $email = $_POST['email'];
21.     $departmentId = $_POST['departmentId'];
22.     $sectionId = $_POST['sectionId'];
23.     $positionId = $_POST['positionId'];
24.     $roleId = $_POST['roleId'];
25.
26.     $user = "UPDATE users SET name = '". $name."',
email = '". $email."', role_id = '". $roleId."',
department_id = '". $departmentId."', section_id =
'". $sectionId."', position_id = '". $positionId.'"
WHERE id = '". $userid.'";
27.
28.     if($query = mysqli_query($link, $user)) {
29.         $_SESSION["name"] = $name;
30.         $_SESSION["email"] = $email;
31.         $_SESSION["role_id"] = $roleId;
32.         $_SESSION["dept_id"] = $departmentId;
33.         $_SESSION["sect_id"] = $sectionId;
34.         $_SESSION["pos_id"] = $positionId;
```

```

35.
36.     $extra = "SELECT roles.name as role,
departments.name as department, sections.name as
section, positions.name as position from users,
roles, departments, sections, positions WHERE
users.id = ".$userid." AND users.role_id = roles.id
AND users.department_id = departments.id AND
users.section_id = sections.id AND users.position_id
= positions.id";
37.
38.     $query = mysqli_query($link, $extra);
39.     if($row = mysqli_fetch_assoc($query)) {
40.         $_SESSION["role"] = $row['role'];
41.         $_SESSION["department"] =
$row['department'];
42.         $_SESSION["section"] = $row['section'];
43.         $_SESSION["position"] = $row['position'];
44.
45.         mysqli_free_result($query);
46.
47.         mysqli_close($link);
48.
49.         header("Location: profile.php?message=1");
50.         die();
51.     }
52.     else {
53.         mysqli_free_result($query);
54.
55.         mysqli_close($link);
56.
57.         header("Location: profile.php?message=2");
58.         die();
59.     }
60. }
61. else {
62.     mysqli_free_result($query);
63.
64.     mysqli_close($link);
65.
66.     header("Location: profile.php?message=2");
67.     die();
68. }
69.

```

70. ?>

### 5.1.21 Profile-pass-update

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “User” di *database*. Data yang diperbaharui hanya dikhususkan terhadap satu atribut tabel yaitu “password”. Kontrol ini lebih khusus mengatur data milik pengguna yang sedang mengakses *website*. Berikut adalah potongan kode dari *profile-pass-update.php*:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.        !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.     $password = "";
14.
15.     $userid = $_POST['id'];
16.     $name = $_POST['name'];
17.     $email = $_POST['email'];
18.     $password = password_hash($_POST['password'],
19.        PASSWORD_DEFAULT);
20.     $user = "UPDATE users SET password =
21.        '". $password.'" WHERE id = ". $userid."";
22.     if($query = mysqli_query($link, $user)) {
23.         mysqli_free_result($query);
24.
25.         mysqli_close($link);
26.
27.         header("Location: profile.php?message=1");
28.         die();
```

```

29.     }
30.     else {
31.         mysqli_free_result($query);
32.
33.         mysqli_close($link);
34.
35.         header("Location: profile.php?message=2");
36.         die();
37.     }
38.
39. ?>

```

### 5.1.22 Profile-sign-update

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “User” di *database*. Data yang diperbaharui hanya dikhususkan terhadap satu atribut tabel yaitu “*sign\_attachment*”. Kontrol ini lebih khusus mengatur data milik pengguna yang sedang mengakses *website*. Berikut adalah potongan kode dari *profile-sign-update.php*:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.     !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');
10.
11.     $name = "";
12.     $email = "";
13.
14.     $userid = $_POST['id'];
15.     $name = $_POST['name'];
16.     $email = $_POST['email'];
17.
18.     $targetDir = "storage/signature/";
19.     $file = $_FILES['signature']['name'];
20.     $temp_name = $_FILES['signature']['tmp_name'];

```

```

21.   $lampiran = $targetDir.$name.'_'.$file;
22.
23.   $old_path = "SELECT users.sign_attachment FROM
users WHERE id = ".$userid."";
24.   $old_path_query = mysqli_query($link, $old_path);
25.   $old_signature;
26.   if($old_signature =
mysqli_fetch_assoc($old_path_query)) {
27.
   if(file_exists($old_signature['sign_attachment']))
28.       unlink($old_signature['sign_attachment']);
29.   }
30.
   if(file_exists($lampiran)) {
31.       header("Location: user.php?message=2");
32.       die();
33.   }
34.   }
35.   else{
36.       move_uploaded_file($temp_name, $lampiran);
37.
   $user = "UPDATE users SET sign_attachment =
38.   '".$lampiran."' WHERE id = ".$userid."";
39.
   if($query = mysqli_query($link, $user)) {
40.       mysqli_free_result($query);
41.
42.       mysqli_close($link);
43.
   header("Location: profile.php?message=1");
44.   die();
45.   }
46.   }
47.   else {
48.       mysqli_free_result($query);
49.
50.       mysqli_close($link);
51.
52.       header("Location: profile.php?message=2");
53.       die();
54.   }
55.   }
56. }
57.
58. ?>

```

### 5.1.23 Sop-create.php

Kontrol ini digunakan untuk membuat data baru ke tabel “SOP” dalam *database*. Berikut adalah potongan kode dari *sop-create.php*:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.        !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');
10.
11.     $name = "";
12.     $info = "";
13.     $dept_id = "";
14.     $sect_id = "";
15.     $userid = "";
16.
17.     $name = $_POST['name'];
18.     $info = $_POST['info'];
19.     $dept_id = $_POST['dept_id'];
20.     $sect_id = $_POST['sect_id'];
21.     $userid = $_POST['userid'];
22.
23.     $targetDir = "storage/sop/";
24.     $file = $_FILES['docfile']['name'];
25.     $temp_name = $_FILES['docfile']['tmp_name'];
26.     $attachment = $targetDir.$name.'_'.$file;
27.
28.     if(file_exists($attachment)) {
29.         header("Location: sop.php?message=2");
30.         die();
31.     }
32.     else {
33.         move_uploaded_file($temp_name, $attachment);
34.
```

```

35.         if($_SESSION['l1_exist'] == 0){
36.             $sop = "INSERT INTO sop VALUES('',
                '$name.', '$info.', '$attachment.',
                '$dept_id.', '$sect_id.', 'date("Y-m-d")."',
                'date("Y-m-d")."', ' ', 2, '$userid.',
                '$userid.', 0)";
37.         }
38.         else if($_SESSION['l2_exist'] == 0){
39.             $sop = "INSERT INTO sop VALUES('',
                '$name.', '$info.', '$attachment.',
                '$dept_id.', '$sect_id.', 'date("Y-m-d")."',
                ' ', 1, '$userid.', -1, 0)";
40.         }
41.         else{
42.             $sop = "INSERT INTO sop VALUES('',
                '$name.', '$info.', '$attachment.',
                '$dept_id.', '$sect_id.', 'date("Y-m-d")."',
                ' ', 1, '$userid.', 0, 0)";
43.         }
44.
45.         if($query = mysqli_query($link, $sop)) {
46.             $id = "SELECT id from sop WHERE attachment =
                '$attachment.'";
47.
48.             $query = mysqli_query($link, $id);
49.             if($row = mysqli_fetch_assoc($query)) {
50.                 $sopid = $row['id'];
51.                 $insertLog = "INSERT INTO log_sop VALUES
                ('', '$sopid.', 'Create', ' ', '$userid.',
                'date("Y-m-d")."')";
52.
53.                 if($query = mysqli_query($link,
                $insertLog)) {
54.                     mysqli_free_result($query);
55.
56.                     mysqli_close($link);
57.
58.                     header("Location: sop.php?message=1");
59.                     die();
60.                 }
61.             }
62.         }
63.

```

```

64.     mysqli_free_result($query);
65.
66.     mysqli_close($link);
67.
68.     header("Location: sop.php?message=2");
69.     die();
70. }
71. ?>

```

### 5.1.24 Sop-update.php

Kontrol ini digunakan untuk memperbaharui data yang ada pada tabel “SOP” di *database*. Berikut adalah potongan kode dari *sop-update.php*:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
5.     !isset($_SESSION["role"])) {
6.         header("Location: logout.php");
7.         die();
8.     }
9.     include('config.php');
10.
11.     $doc_number = "";
12.     $info = "";
13.
14.     $doc_number = $_POST['name'];
15.     $info = $_POST['info'];
16.     $sopid = $_POST['sopid'];
17.
18.     $targetDir = "storage/sop/";
19.     $file = $_FILES['docfile']['name'];
20.     $temp_name = $_FILES['docfile']['tmp_name'];
21.     $attachment = $targetDir.$doc_number.'_'.$file;
22.
23.     $old_path = "SELECT sop.attachment FROM sop WHERE
24.     id = ".$sopid."";
25.     $old_path_query = mysqli_query($link, $old_path);
26.     $old_sopfile;

```



```

26.     if($old_sopfile =
mysql_fetch_assoc($old_path_query)) {
27.         if(file_exists($old_sopfile['attachment']))
28.             unlink($old_sopfile['attachment']);
29.     }
30.
31.     if(file_exists($attachment)) {
32.         header("Location: sop.php?message=2");
33.         die();
34.     }
35.     else {
36.         move_uploaded_file($temp_name, $attachment);
37.
38.         $sop = "UPDATE sop SET doc_number =
'".$doc_number."', info = '".$info."', attachment =
'".$attachment."', status = 1 WHERE id = ".$sopid;
39.
40.         $updateLog = "INSERT INTO log_sop VALUES ('',
'".$sopid."', 'Revised', '', '".$l1_id."',
'".$date("Y-m-d")."')";
41.
42.         if($query = mysqli_query($link, $sop)) {
43.             mysqli_free_result($query);
44.
45.             mysqli_close($link);
46.
47.             header("Location: sop.php?message=1");
48.             die();
49.         }
50.         else {
51.             mysqli_free_result($query);
52.
53.             mysqli_close($link);
54.
55.             header("Location: sop.php?message=2");
56.             die();
57.         }
58.     }
59. ?>

```

### 5.1.25 Sop-revision.php

Kontrol ini digunakan untuk mengunduh petunjuk revisi terhadap berkas SOP. Berikut adalah potongan kode dari sop-revision.php:

```
1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     $revid = $_GET['id'];
10.    $sopid = $_GET['sopid'];
11.
12.    include('config.php');
13.
14.    $sop = "SELECT attachment FROM revision_sop WHERE
id = ".$revid."";
15.    $query = mysqli_query($link, $sop);
16.
17.    if($row = mysqli_fetch_assoc($query)) {
18.        $file = $row['attachment'];
19.    }
20.    else {
21.        mysqli_free_result($query);
22.
23.        mysqli_close($link);
24.
25.        header("Location: sop-
edit.php?sopid=".$sopid."&message=2");
26.        die();
27.    }
28.
29.    if(file_exists($file)) {
30.        header('Content-Description: File Transfer');
31.        header('Content-Type: application/octet-
stream');
32.        header('Content-Disposition: attachment;
filename='.basename($file));
```

```

33.     header('Content-Transfer-Encoding: binary');
34.     header('Expires: 0');
35.     header('Cache-Control: private');
36.     header('Pragma: private');
37.     header('Content-Length: ' . filesize($file));
38.     ob_clean();
39.     flush();
40.     readfile($file);
41. }
42. else {
43.     header("Location: sop-
edit.php?sopid=".$sopid."&message=2");
44. }
45. ?>

```

### 5.1.26 Sop-rejection.php

Kontrol ini digunakan untuk mengunduh petunjuk penolakan terhadap berkas SOP. Berikut adalah potongan kode dari `sop-rejection.php`:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     $rejid = $_GET['id'];
10.
11.     include('config.php');
12.
13.     $sop = "SELECT attachment FROM rejection_sop WHERE
id = ".$rejid."";
14.     $query = mysqli_query($link, $sop);
15.
16.     if($row = mysqli_fetch_assoc($query)) {
17.         $file = $row['attachment'];
18.     }
19.     else {

```

```

20.     mysqli_free_result($query);
21.
22.     mysqli_close($link);
23.
24.     header("Location: sop.php?message=2");
25.     die();
26. }
27.
28. if(file_exists($file)) {
29.     header('Content-Description: File Transfer');
30.     header('Content-Type: application/octet-
stream');
31.     header('Content-Disposition: attachment;
filename='.basename($file));
32.     header('Content-Transfer-Encoding: binary');
33.     header('Expires: 0');
34.     header('Cache-Control: private');
35.     header('Pragma: private');
36.     header('Content-Length: ' . filesize($file));
37.     ob_clean();
38.     flush();
39.     readfile($file);
40. }
41. else {
42.     header("Location: sop.php?message=2");
43. }
44. ?>

```

### 5.1.27 Sop-template.php

Kontrol ini digunakan untuk mengunduh templat berkas SOP yang akan digunakan sebagai format berkas yang akan diunggah. Berikut adalah potongan kode dari sop-template.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();

```

```

7.     }
8.
9.     $file = "storage/example/PK-
001_Document_Numbering.pdf";
10.
11.    if(file_exists($file)) {
12.        header('Content-Description: File Transfer');
13.        header('Content-Type: application/octet-
stream');
14.        header('Content-Disposition: attachment;
filename='.basename($file));
15.        header('Content-Transfer-Encoding: binary');
16.        header('Expires: 0');
17.        header('Cache-Control: private');
18.        header('Pragma: private');
19.        header('Content-Length: ' . filesize($file));
20.        ob_clean();
21.        flush();
22.        readfile($file);
23.    }
24.    else {
25.        header("location:sop.php?message=2");
26.    }
27. ?>

```

### 5.1.28 Sop-delete.php

Kontrol ini digunakan untuk menghapus data yang ada pada tabel “SOP” di *database*. Berikut adalah potongan kode dari `sop-delete.php`:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.

```

```

11.  $sopid = $_GET['sopid'];
12.
13.  $old_path = "SELECT sop.attachment FROM sop WHERE
id = ".$sopid.";
14.  $old_path_query = mysqli_query($link, $old_path);
15.  $old_signature;
16.  if($old_signature =
mysqli_fetch_assoc($old_path_query)){
17.      if(file_exists($old_signature['attachment']))
18.          unlink($old_signature['attachment']);
19.  }
20.
21.  $user = "DELETE FROM sop WHERE id = ".$sopid.";
22.
23.  if($query = mysqli_query($link, $user)) {
24.      mysqli_free_result($query);
25.
26.      mysqli_close($link);
27.
28.      header("Location: sop.php?message=1");
29.      die();
30.  }
31.  else {
32.      mysqli_free_result($query);
33.
34.      mysqli_close($link);
35.
36.      header("Location: sop.php?message=2");
37.      die();
38.  }
39. ?>

```

### 5.1.29 Validasi-approve.php

Kontrol ini digunakan untuk melakukan persetujuan atau validasi terhadap berkas SOP yang telah diunggah. Status SOP akan diperbaharui, serta pihak yang melakukan perubahan akan dikunci sebagai pengurus berkas SOP. Berikut adalah potongan kode dari validasi-approve.php:

```

1. <?php
2.     session_start();

```

```

3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $id = $_SESSION["id"];
12.     $sopid = $_GET['sopid'];
13.     $role_id = $_SESSION['role_id'];
14.
15.     $sop = "";
16.     $validateLog = "";
17.
18.     if($role_id == 2) {
19.         $sop = "UPDATE sop SET sop.status = 2,
sop.l2_id = ".$id.", sop.reviewed_at = '".date("Y-m-
d")."' WHERE sop.id = ".$sopid;
20.         $validateLog = "INSERT INTO log_sop VALUES ('',
'".$sopid."', 'Validate 1', '', '".$id."',
'".date("Y-m-d")."')";
21.
22.         if($query = mysqli_query($link, $sop)) {
23.             if($query = mysqli_query($link,
$validateLog)) {
24.                 mysqli_free_result($query);
25.
26.                 mysqli_close($link);
27.
28.                 header("Location:
validasi.php?message=1");
29.                 die();
30.             }
31.         }
32.     }
33.     else {
34.         $sop = "";
35.         if(isset($_GET["l2_not_exist"]))
36.             $sop = "UPDATE sop SET sop.status = 3,
sop.l2_id = ".$id.", sop.l3_id = ".$id.",

```

```

        sop.validated_at = '".date("Y-m-d")."' WHERE sop.id =
        ".$sopid;
37.         else
38.             $sop = "UPDATE sop SET sop.status = 3,
        sop.l3_id = ".$id.", sop.validated_at = '".date("Y-m-
        d")."' WHERE sop.id = ".$sopid;
39.             $validateLog = "INSERT INTO log_sop VALUES ('',
        '$sopid.', 'Validate 2', '', '$id.',
        '".date("Y-m-d")."');";
40.             $attachment = "SELECT sop.attachment FROM sop
        WHERE sop.id = ".$sopid;
41.
42.             if($query = mysqli_query($link, $sop)){
43.                 if($query = mysqli_query($link,
        $validateLog)){
44.                     if($query = mysqli_query($link,
        $attachment)){
45.                         $row = mysqli_fetch_assoc($query);
46.                         $old_path = $row["attachment"];
47.                         $new_path = str_replace("storage/sop",
        "storage/history", $row["attachment"]);
48.                         $new_path = substr_replace($new_path,
        "_".date("Y-m-d_H-i-s").".pdf", -4);
49.
50.                         if(!copy($row["attachment"],
        $new_path)){
51.                             mysqli_free_result($query);
52.
53.                             mysqli_close($link);
54.
55.                             header("Location:
        validasi.php?message=2");
56.                             die();
57.                         }
58.                         $validateHistory = "INSERT INTO
        validation_histories VALUES ('', '$sopid.',
        '$new_path.', '".date("Y-m-d")."');";
59.                         if($query = mysqli_query($link,
        $validateHistory)){
60.                             mysqli_free_result($query);
61.
62.                             mysqli_close($link);
63.

```



```

64.             header("Location:
validasi.php?message=1");
65.             die();
66.         }
67.     }
68. }
69. }
70.
71. mysqli_close($link);
72.
73. header("Location: validasi.php?message=2");
74. die();
75. }
76. ?>

```

### 5.1.30 Validasi-revision.php

Kontrol ini digunakan untuk melakukan revisi terhadap berkas SOP yang telah diunggah. Status SOP dijadikan “revisi”, serta pihak yang melakukan perubahan akan dikunci sebagai pengurus berkas SOP. Berikut adalah potongan kode dari validasi-revision.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $l2_id = "";
12.     $sopid = "";
13.     $revision = "";
14.
15.     $l2_id = $_POST['l2_id'];
16.     $sopid = $_POST['sopid'];
17.     $revision = $_POST['revision'];
18.

```

```

19.  $targetDir = "storage/revision/";
20.  $file = $_FILES['docfile']['name'];
21.  $temp_name = $_FILES['docfile']['tmp_name'];
22.  $lampiran =
    $targetDir.$l2_id.'_'.$sopid.'_'.$file;
23.
24.  $old_path = "SELECT revision_sop.attachment FROM
    revision_sop WHERE sop_id = ".$sopid." ORDER BY
    revision_sop.created_at DESC LIMIT 1";
25.  $old_path_query = mysqli_query($link, $old_path);
26.  $old_revision;
27.  if($old_revision =
    mysqli_fetch_assoc($old_path_query)) {
28.      if(file_exists($old_revision['attachment']))
29.          unlink($old_revision['attachment']);
30.  }
31.
32.  if(file_exists($lampiran)) {
33.      header("Location: validasi-revision-
    form.php?sopid=".$sopid."&message=2");
34.      die();
35.  }
36.  else {
37.      move_uploaded_file($temp_name, $lampiran);
38.
39.      $sop = "UPDATE sop SET sop.status = 4,
    sop.l2_id = ".$l2_id." WHERE sop.id = ".$sopid;
40.      $sopRevision = "INSERT INTO revision_sop VALUES
    ('', '$sopid.', '$l2_id.', '$revision.',
    '$lampiran.', '.date("Y-m-d H:i:s").')";
41.      $revisionLog = "INSERT INTO log_sop VALUES ('',
    '$sopid.', 'Correction', '', '$l2_id.',
    '.date("Y-m-d").')";
42.
43.      if($query = mysqli_query($link, $sop)) {
44.          if($query = mysqli_query($link,
    $sopRevision)) {
45.              if($query = mysqli_query($link,
    $revisionLog)) {
46.                  mysqli_free_result($query);
47.
48.                  mysqli_close($link);
49.

```

```

50.             header("Location:
validasi.php?message=1");
51.             die();
52.         }
53.     }
54. }
55. }
56.
57.     mysqli_free_result($query);
58.
59.     mysqli_close($link);
60.
61.     header("Location: validasi-revision-
form.php?sopid=".$sopid."&message=2");
62.     die();
63. ?>

```

### 5.1.31 Validasi-reject.php

Kontrol ini digunakan untuk melakukan penolakan terhadap berkas SOP yang telah diunggah. Status SOP dijadikan "ditolak". Berikut adalah potongan kode dari validasi-reject.php:

```

1. <?php
2.     session_start();
3.
4.     if(!isset($_SESSION["name"]) ||
!isset($_SESSION["role"])) {
5.         header("Location: logout.php");
6.         die();
7.     }
8.
9.     include('config.php');
10.
11.     $l3_id = "";
12.     $sopid = "";
13.     $rejection = "";
14.
15.     $l3_id = $_POST['l3_id'];
16.     $sopid = $_POST['sopid'];
17.     $rejection = $_POST['rejection'];

```

```

18.
19.   $targetDir = "storage/rejection/";
20.   $file = $_FILES['docfile']['name'];
21.   $temp_name = $_FILES['docfile']['tmp_name'];
22.   $lampiran =
    $targetDir.$l3_id.'_'.$sopid.'_'.$file;
23.
24.   $old_path = "SELECT rejection_sop.attachment FROM
    rejection_sop WHERE sop_id = ".$sopid." ORDER BY
    rejection_sop.created_at DESC LIMIT 1";
25.   $old_path_query = mysqli_query($link, $old_path);
26.   $old_rejection;
27.   if($old_rejection =
    mysqli_fetch_assoc($old_path_query)) {
28.       if(file_exists($old_rejection['attachment']))
29.           unlink($old_rejection['attachment']);
30.   }
31.
32.   if(file_exists($lampiran)) {
33.       header("Location: validasi-reject-
    form.php?sopid=".$sopid."&message=2");
34.       die();
35.   }
36.   else {
37.       move_uploaded_file($temp_name, $lampiran);
38.
39.       $sop = "UPDATE sop SET sop.status = 5,
    sop.l3_id = ".$l3_id." WHERE sop.id = ".$sopid;
40.       $sopRejection = "INSERT INTO rejection_sop
    VALUES ('', '".$sopid."', '".$l3_id."',
    '".$rejection."', '".$lampiran."', '.date("Y-m-d
    H:i:s")."')";
41.       $rejectionLog = "INSERT INTO log_sop VALUES
    ('', '".$sopid."', 'Rejection', '', '".$l3_id."',
    '".date("Y-m-d")."')";
42.
43.       if($query = mysqli_query($link, $sop)) {
44.           if($query = mysqli_query($link,
    $sopRejection)) {
45.               if($query = mysqli_query($link,
    $rejectionLog)) {
46.                   mysqli_free_result($query);
47.

```

```

48.             mysqli_close($link);
49.
50.             header("Location:
validasi.php?message=1");
51.             die();
52.         }
53.     }
54. }
55. }
56.
57.     mysqli_free_result($query);
58.
59.     mysqli_close($link);
60.
61.     header("Location: validasi-reject-
form.php?sopid=".$sopid."&message=2");
62.     die();
63. ?>

```

### 5.1.32 Logout.php

Kontrol ini digunakan untuk keperluan *log out* dari *website*. Berikut adalah potongan kode dari `logout.php`:

```

1. <?php
2. session_start();
3.
4. session_unset();
5.
6. session_destroy();
7.
8. header("Location: login.php");
9. die();
10. ?>

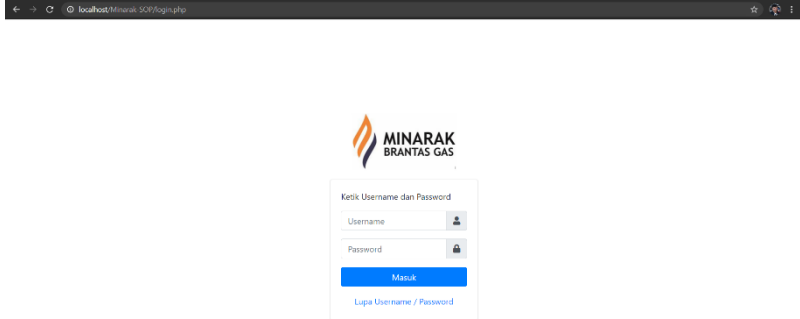
```

## 5.2. Implementasi View

Pada bagian ini kami akan menampilkan antarmuka halaman yang ada pada aplikasi E-SOP Minarak Brantas Gas. Berikut tampilan antarmuka aplikasi E-SOP SOP Minarak Brantas Gas:

### 5.2.1 View Login

Gambar 5.1 berisi antarmuka login dengan form yang berisi username dan kata sandi serta tombol ‘masuk’ untuk dapat login pada sistem menggunakan hak akses yang sesuai.

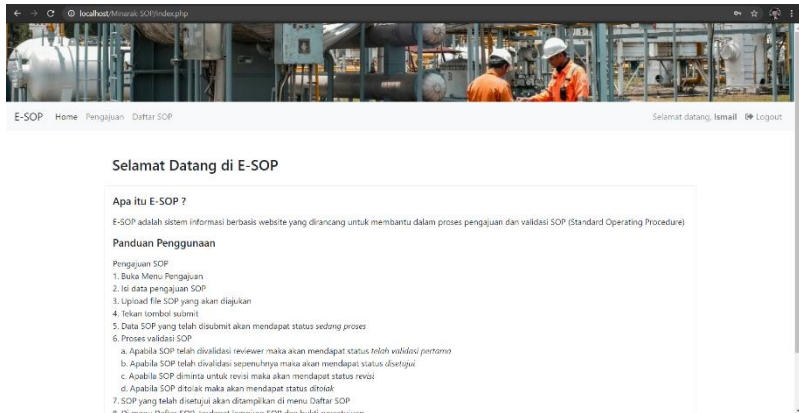


*Gambar 5. 1 Login Web*

### 5.2.2 View Home

#### a. Home L1

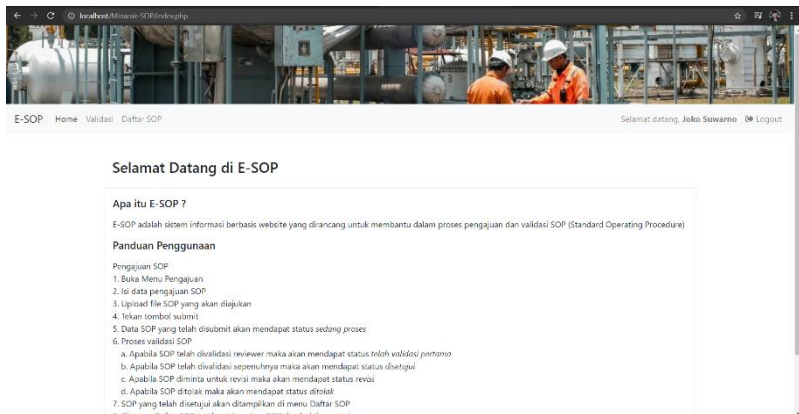
Gambar 5.2 berisi antarmuka halaman awal bagi pengguna yang memiliki hak akses L1.



Gambar 5. 2 Home L1

b. Home L2 dan L3

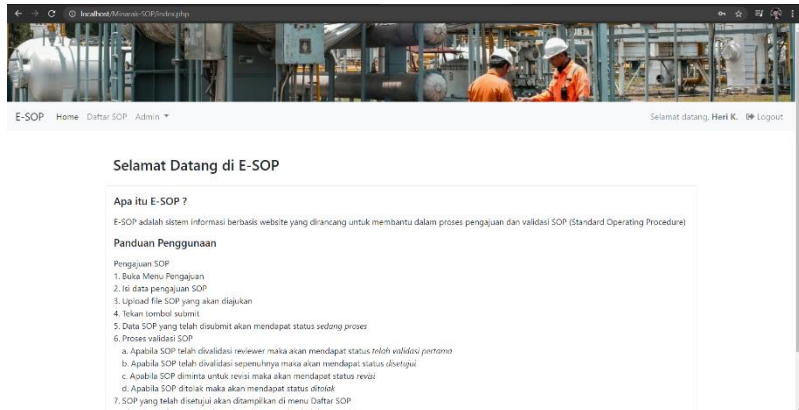
Gambar 5.3 berisi antarmuka halaman awal bagi pengguna yang memiliki hak akses L2 atau L3.



Gambar 5. 3 Home L2 atau L3

### c. Dashboard Admin

Gambar 5.4 berisi antarmuka halaman awal bagi pengguna yang memiliki hak akses Admin.

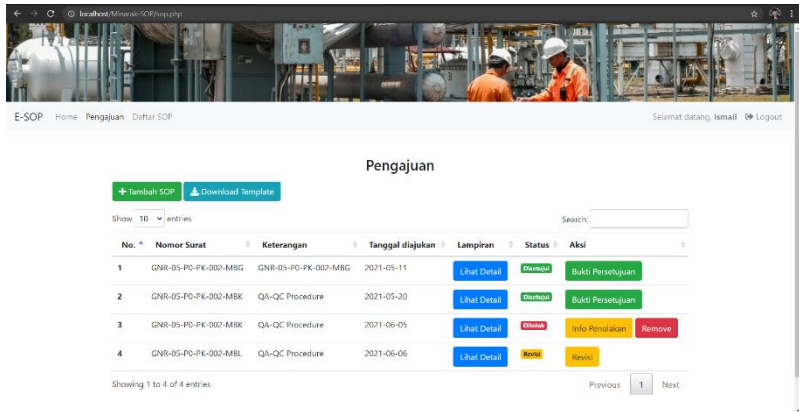


Gambar 5. 4 Home Admin

### 5.2.3 View Pengajuan

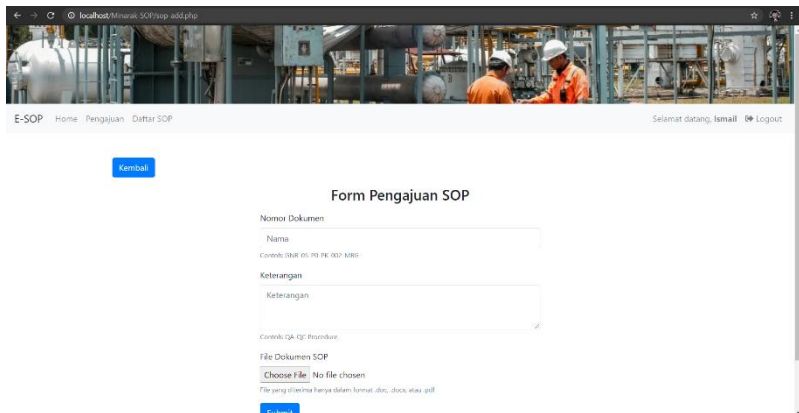
Gambar 5.5 berisi antar muka menu “Pengajuan”. Web menampilkan data berupa daftar berkas SOP yang telah diunggah ke dalam website. Terdapat beberapa tombol yang dapat digunakan dalam laman ini. “Tambah” dapat digunakan untuk mengunggah berkas SOP ke dalam web. “Download Template” digunakan untuk mengunduh templat berkas SOP. “Info Penolakan” digunakan untuk menampilkan keterangan penolakan dokumen. “Remove” menghapus berkas SOP. “Revisi” membuka laman baru untuk memperbaiki berkas SOP. “Lihat Detail” menampilkan isi berkas yang diunggah.





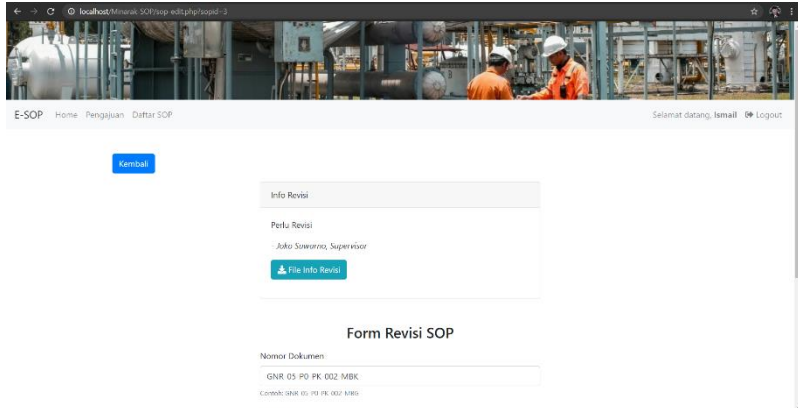
Gambar 5. 5 Pengajuan SOP

Gambar 5.6 menampilkan form pengajuan berkas SOP baru. Data yang diperlukan adalah kode dokumen, keterangan serta berkas dokumen yang bersangkutan.



Gambar 5. 6 Form Pengajuan SOP

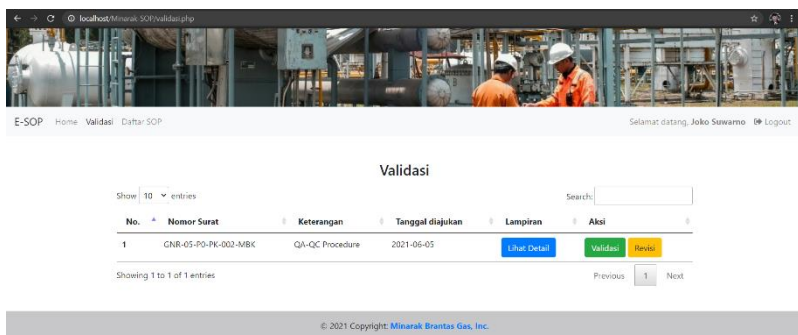
Gambar 5.7 menampilkan form revisi berkas SOP dari menu “Pengajuan”. Berkas baru yang telah diperbaiki diunggah ulang ke dalam web dan mengganti berkas yang telah diunggah sebelumnya.



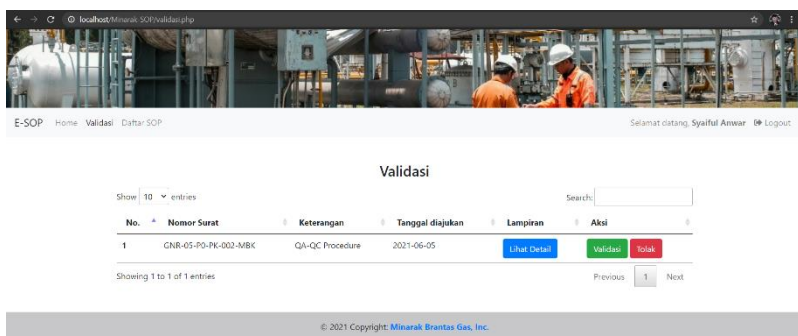
Gambar 5.7 Form Revisi SOP LI

## 5.2.4 View Validasi

Gambar 5.8 dan gambar 5.9 menunjukkan tampilan menu “Validasi”. Web menampilkan daftar berkas SOP yang telah diunggah oleh pengguna dengan hak akses LI. Terdapat beberapa tombol yang dapat digunakan. “Validasi” akan menyetujui berkas dan dilanjutkan sesuai alur persetujuan berkas SOP. “Revisi” akan membuka form revisi untuk memberi keterangan terhadap kesalahan berkas SOP dan menjadikan status berkas “Revisi”. “Tolak” akan membuka form penolakan untuk memberi keterangan akan kesalahan berkas dan menjadikan status berkas “Ditolak”. “Lihat Detail” menampilkan isi berkas SOP guna perihal pemeriksaan.

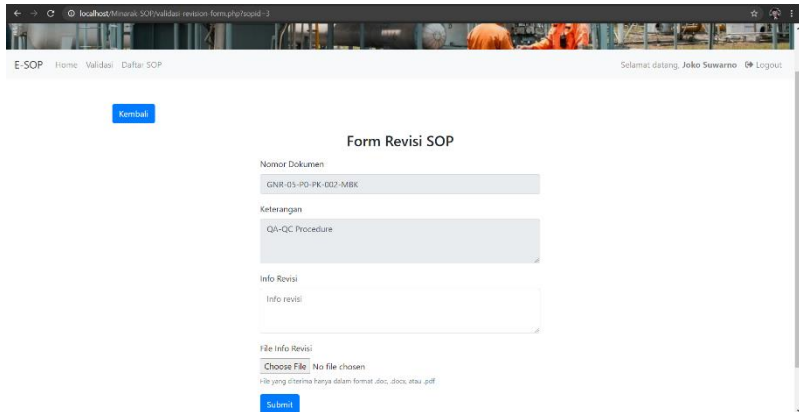


Gambar 5. 8 Validasi L2



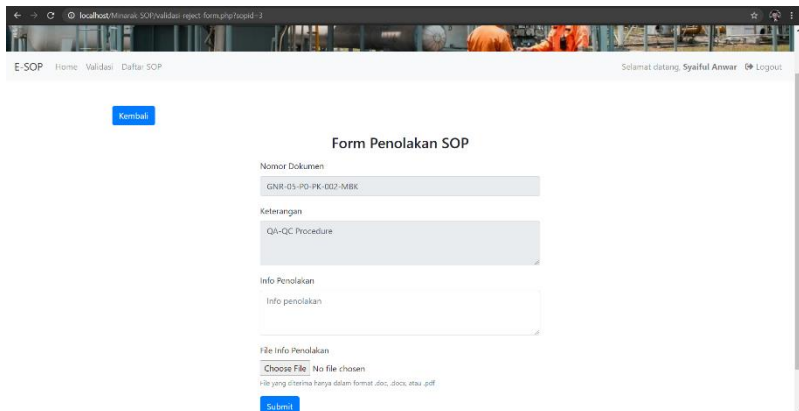
Gambar 5. 9 Validasi L3

Gambar 5.10 menampilkan form revisi berkas SOP dari menu “Validasi”. Data berkas ini diberi komentar atau keterangan akan kesalahan dokumen. Keterangan revisi juga diikutsertakan sebuah berkas, yakni berkas SOP yang telah ditandai letak-letak kesalahan SOP.



Gambar 5. 10 Form Revisi SOP L2

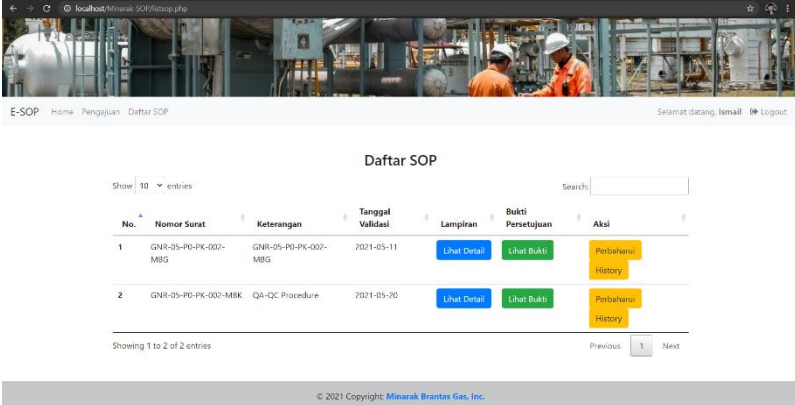
Gambar 5.11 menampilkan form penolakan berkas SOP. Data berkas ini diberi komentar atau keterangan terhadap penyebab dokumen tidak valid. Keterangan penolakan diikutsertakan berkas SOP yang telah ditandai letak-letak kejanggalan berkas.



Gambar 5. 11 Form Penolakan SOP

## 5.2.5 View Daftar SOP

Gambar 5.12 berisi tampilan menu “Daftar SOP”. Web menampilkan daftar berkas SOP yang telah divalidasi sampai tahap akhir. Terdapat beberapa tombol yang dapat digunakan. “Lihat Bukti” akan mengunduh bukti persetujuan atas berkas SOP. Bukti persetujuan ini dibuat secara otomatis berdasarkan templat tertentu dengan tanda tangan pihak-pihak yang bertanggung jawab dalam proses validasi berkas SOP. “Perbaharui” akan menampilkan form pembaharuan SOP jika berkas perlu diperbaharui. “History” akan menampilkan riwayat validasi berkas SOP. “Lihat Detail” akan menampilkan isi berkas SOP yang telah divalidasi.

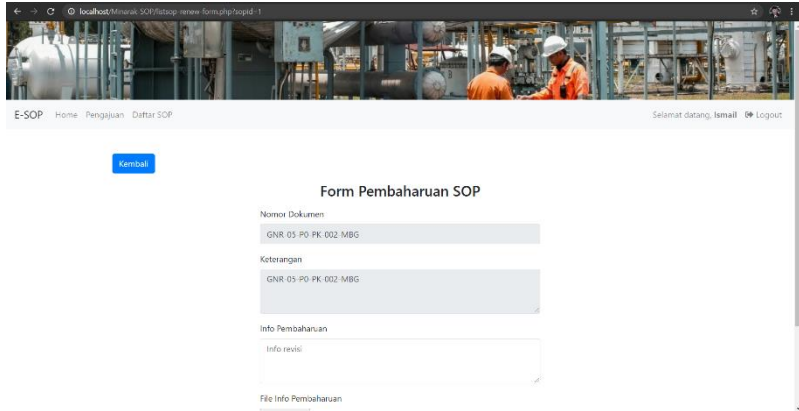


The screenshot shows a web application interface for 'Daftar SOP'. At the top, there is a navigation bar with 'E-SOP', 'Home', 'Pengajuan', and 'Daftar SOP'. A search bar is located on the right. Below the navigation bar, the title 'Daftar SOP' is centered. A table displays two entries of SOPs. The table has columns for 'No.', 'Nomor Surat', 'Keterangan', 'Tanggal Validasi', 'Lampiran', 'Bukti Persetujuan', and 'Aksi'. Each entry has a 'Lihat Detail' button (blue), a 'Lihat Bukti' button (green), and a 'Perbaharui' button (yellow) with a 'History' link below it. The footer of the page contains the copyright notice '© 2021 Copyright, Minarak Brantas Gas, Inc.' and the page number '1'.

No.	Nomor Surat	Keterangan	Tanggal Validasi	Lampiran	Bukti Persetujuan	Aksi
1	QNR-05-PD-PK-002-MBS	QNR-05-PD-PK-002-MBS	2021-05-11		<a href="#">Lihat Bukti</a>	<a href="#">Perbaharui</a> <a href="#">History</a>
2	QNR-05-PD-PK-002-MBK	QA-QC Procedure	2021-05-20		<a href="#">Lihat Bukti</a>	<a href="#">Perbaharui</a> <a href="#">History</a>

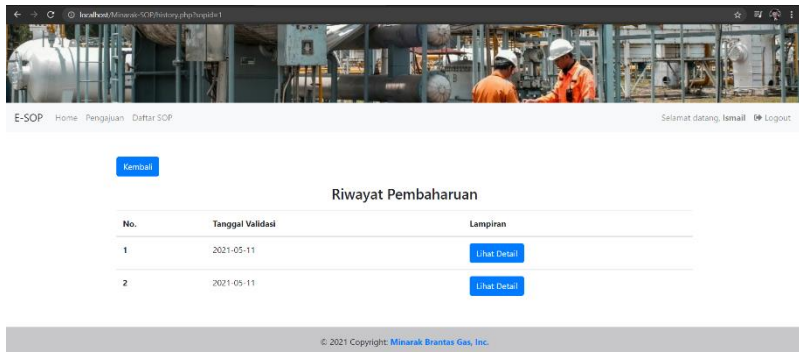
Gambar 5.12 Daftar SOP

Gambar 5.13 menampilkan form pembaharuan SOP. Laman ini memiliki kerja yang sama dengan form revisi SOP. Jika berkas SOP perlu diperbaharui, maka status berkas SOP kembali dijadikan revisi bersama dengan keterangan pembaharuan dokumen.



Gambar 5. 13 Form Pembaharuan SOP

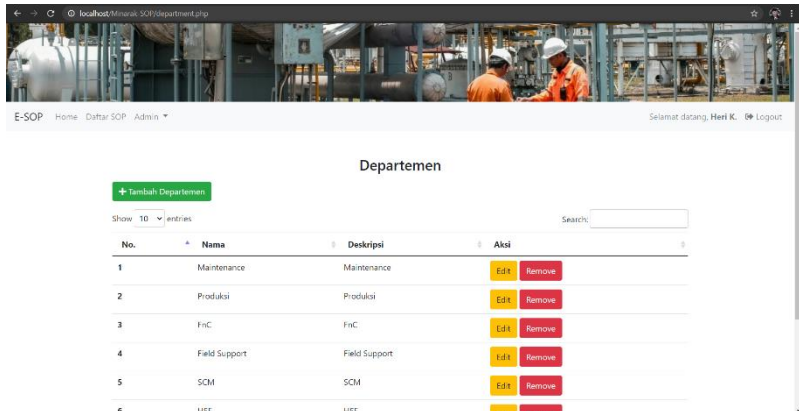
Gambar 5.14 menampilkan riwayat pembaharuan. Setiap kali berkas SOP berhasil divalidasi sampai tahap akhir, informasi pembaharuan tersebut disimpan. Riwayat ini dapat dibuka untuk dilihat apabila diperlukan pemeriksaan terhadap versi lama berkas SOP. Terdapat tombol “Lihat Detail” untuk menampilkan isi berkas SOP yang telah divalidasi berdasarkan riwayat yang tersedia.



Gambar 5. 14 Riwayat Pembaharuan

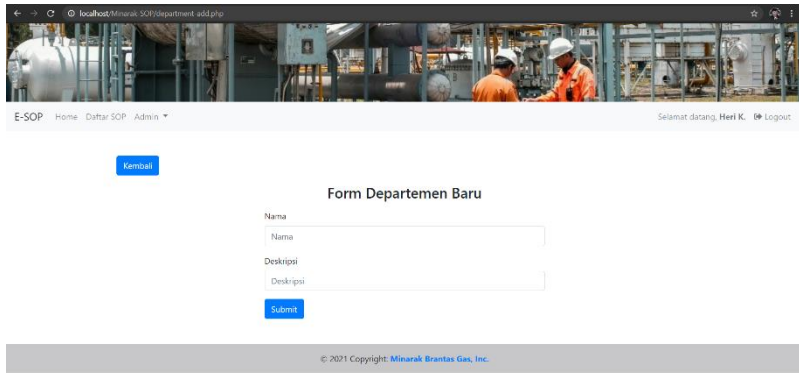
## 5.2.6 View Departemen

Gambar 5.15 menampilkan antarmuka data database dari tabel “Department”. Terdapat tombol “Tambah Departemen” untuk menambah data departemen baru, tombol “Edit” untuk menyunting data departemen, dan tombol “Remove” untuk menghapus data departemen.



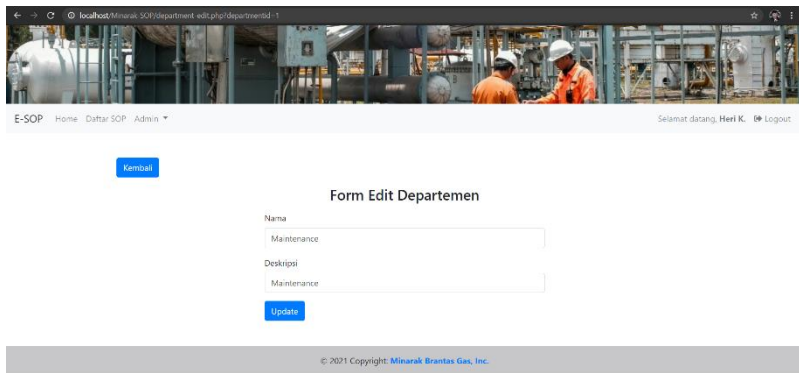
*Gambar 5.15 Departemen*

Gambar 5.16 menampilkan form departemen baru. Form ini digunakan untuk menambah data baru ke tabel “Department” milik database.



*Gambar 5. 16 Form Departemen Baru*

Gambar 5.17 menampilkan form edit departemen. Form ini digunakan untuk mengganti data yang ada dari tabel “Department” milik database.



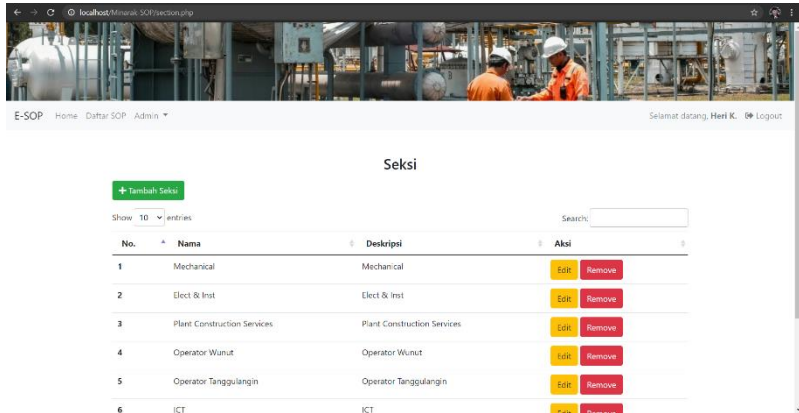
*Gambar 5. 17 Form Edit Departemen*

### 5.2.7 View Seksi

Gambar 5.18 menampilkan antarmuka data database dari tabel “Section”. Terdapat tombol “Tambah Seksi” untuk

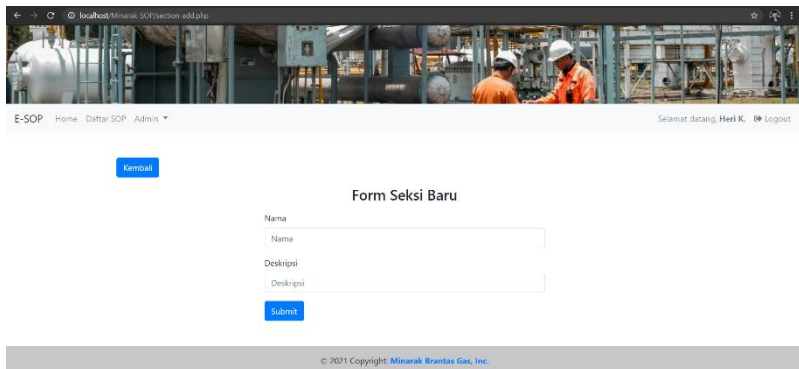


menambah data seksi baru, tombol “Edit” untuk menyunting data seksi, dan tombol “Remove” untuk menghapus data seksi.



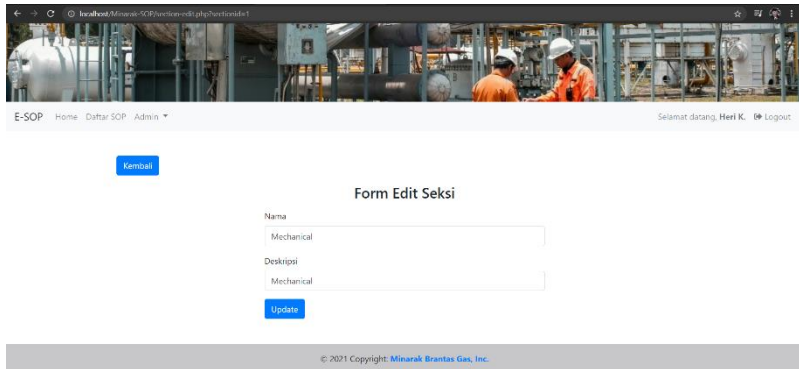
Gambar 5. 18 Seksi

Gambar 5.19 menampilkan form seksi baru. Form ini digunakan untuk menambah data baru ke tabel “Section” milik database.



Gambar 5. 19 Form Seksi Baru

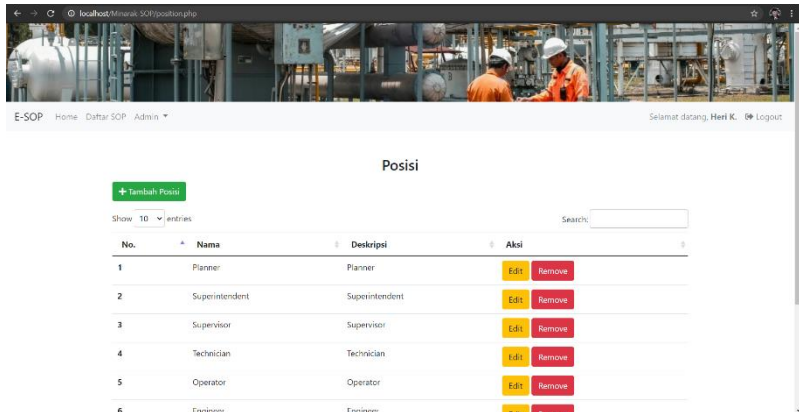
Gambar 5.20 menampilkan form edit seksi. Form ini digunakan untuk mengganti data yang ada dari tabel “Section” milik database.



*Gambar 5. 20 Form Edit Seksi*

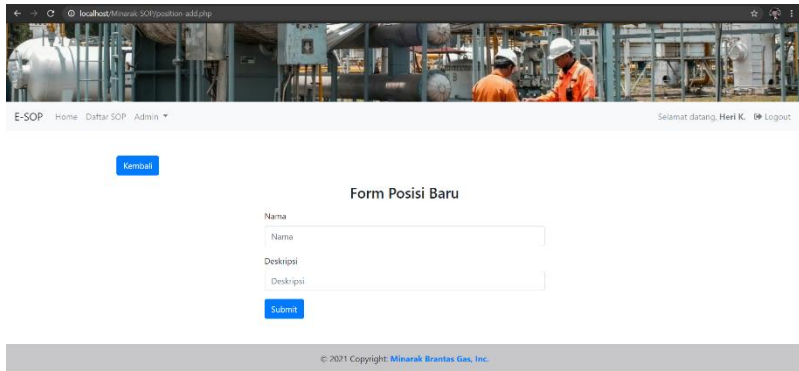
### **5.2.8 View Posisi**

Gambar 5.21 menampilkan antarmuka data database dari tabel “Position”. Terdapat tombol “Tambah Posisi” untuk menambah data posisi baru, tombol “Edit” untuk menyunting data posisi, dan tombol “Remove” untuk menghapus data posisi.



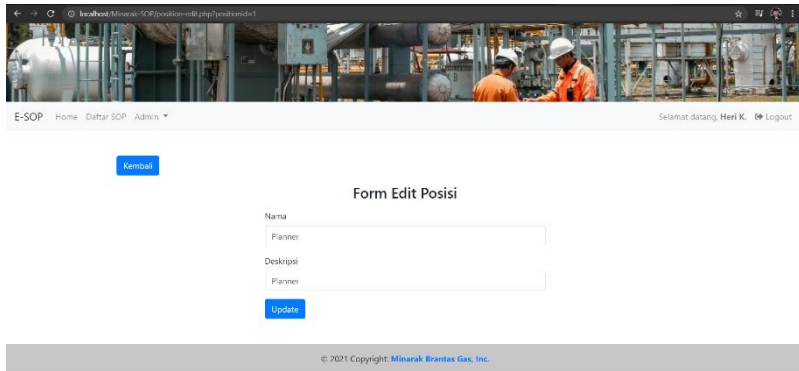
Gambar 5. 21 Posisi

Gambar 5.22 menampilkan form posisi baru. Form ini digunakan untuk menambah data baru ke tabel “Position” milik database.



Gambar 5. 22 Form Posisi Baru

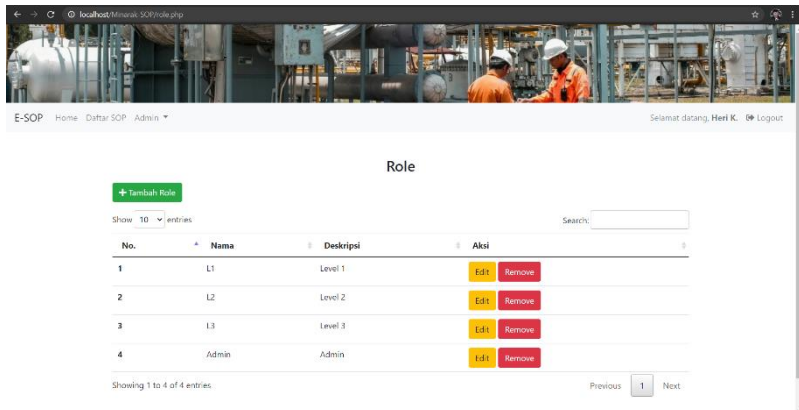
Gambar 5.23 menampilkan form edit posisi. Form ini digunakan untuk mengganti data yang ada dari tabel “Position” milik database.



Gambar 5. 23 Form Edit Posisi

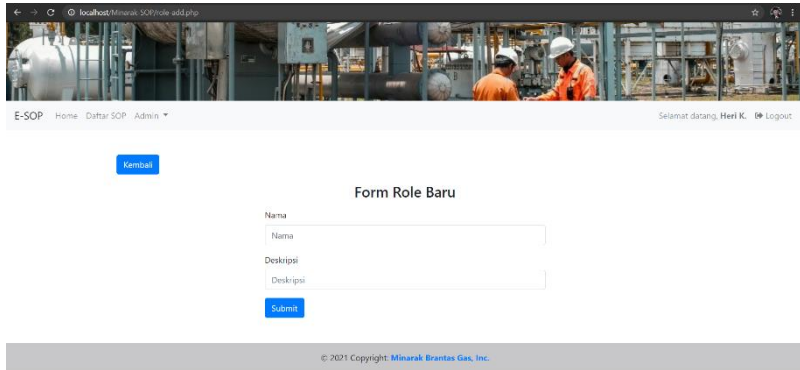
## 5.2.9 View Role

Gambar 5.24 menampilkan antarmuka data database dari tabel “Role”. Terdapat tombol “Tambah Role” untuk menambah data role baru, tombol “Edit” untuk menyunting data role, dan tombol “Remove” untuk menghapus data role.



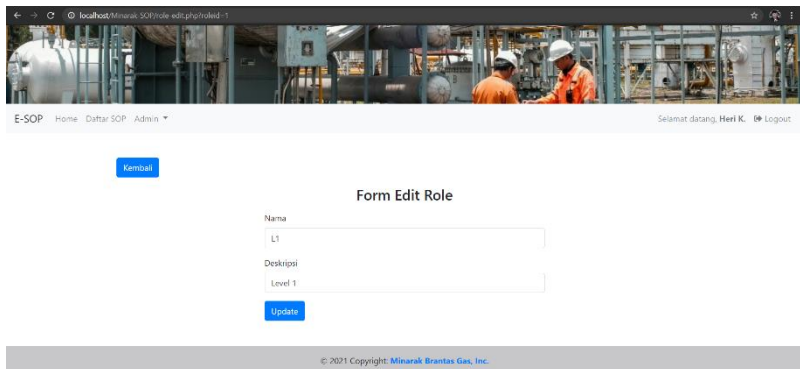
Gambar 5. 24 Role

Gambar 5.25 menampilkan form role baru. Form ini digunakan untuk menambah data baru ke tabel “Role” milik database.



*Gambar 5. 25 Form Role Baru*

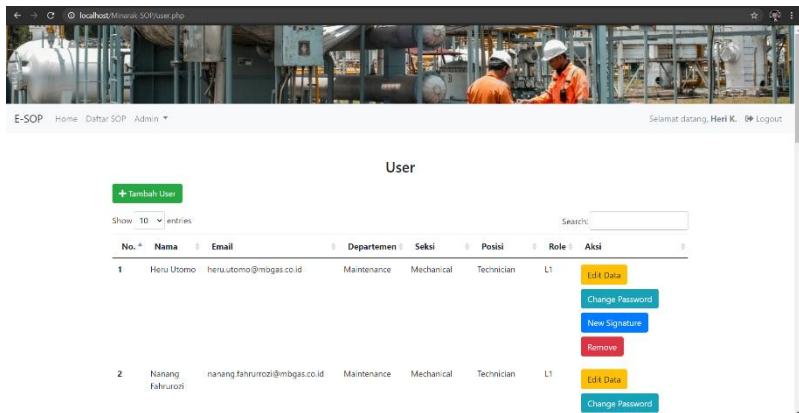
Gambar 5.26 menampilkan form edit role. Form ini digunakan untuk mengganti data yang ada dari tabel “Role” milik database.



*Gambar 5. 26 Form Edit Role*

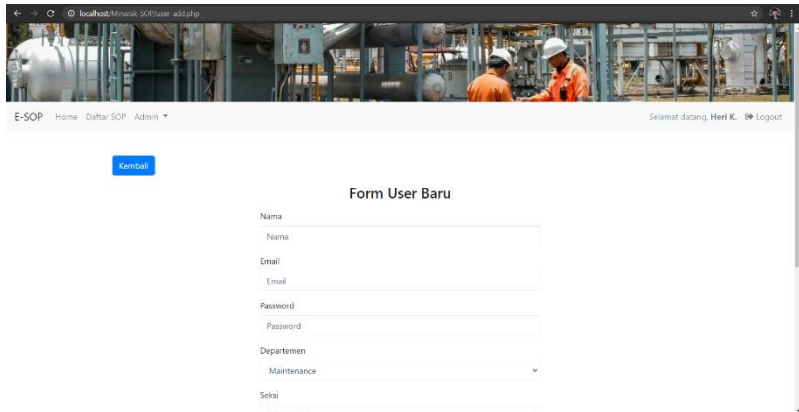
## 5.2.10 View User

Gambar 5.27 menampilkan antarmuka data database dari tabel “User”. Terdapat tombol “Tambah User” untuk menambah data user baru, tombol “Edit Data” untuk menyunting data user, tombol “Change Password” untuk menyunting password user, tombol “New Signature” untuk menyunting tanda tangan user, dan tombol “Remove” untuk menghapus data user.



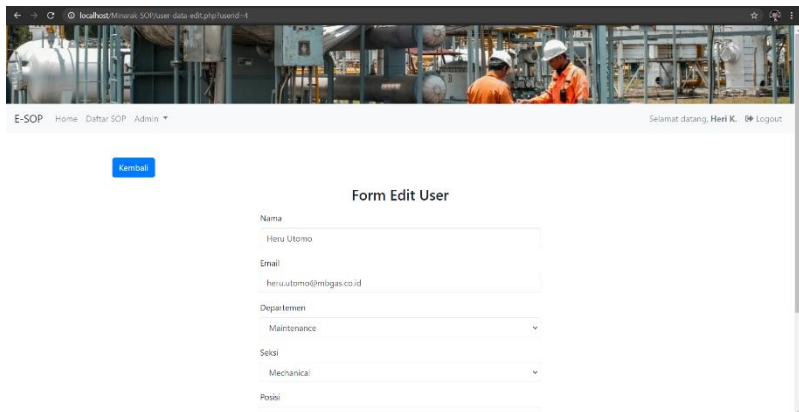
Gambar 5. 27 User

Gambar 5.28 menampilkan form user baru. Form ini digunakan untuk menambah data baru ke tabel “User” milik database.



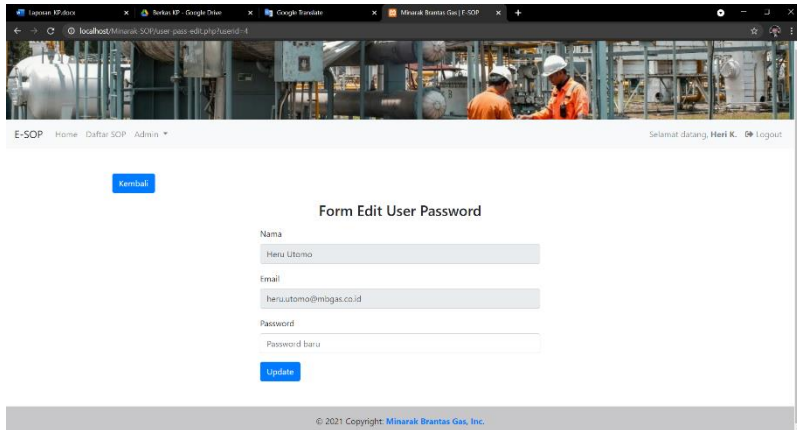
*Gambar 5. 28 Form User Baru*

Gambar 5.29 menampilkan form edit user. Form ini digunakan untuk mengganti data yang ada dari tabel “User” milik database.



*Gambar 5. 29 Form Edit User*

Gambar 5.30 menampilkan form edit user password. Form ini digunakan untuk mengganti data atribut “password” yang ada dari tabel “User” milik database.



The screenshot shows a web browser window with the URL `localhost/Minarak/SOP/user-pass-edit.php?userid=4`. The page features a navigation menu with 'E-SOP', 'Home', 'Daftar SOP', and 'Admin'. A blue 'Kembali' button is located on the left. The main content area is titled 'Form Edit User Password' and contains the following fields:

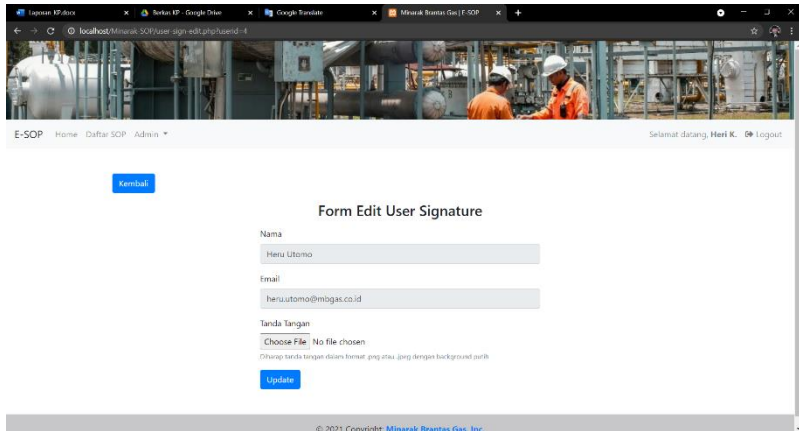
- Nama:** Heru Litomo
- Email:** herulitomo@mbogas.co.id
- Password:** (empty field)
- Password baru:** (empty field)

A blue 'Update' button is positioned below the password fields. The footer of the page displays the copyright notice: '© 2021 Copyright: Minarak Brantas Gas, Inc.'

*Gambar 5. 30 Form Edit User Password*

Gambar 5.31 menampilkan form edit user password. Form ini digunakan untuk mengganti data tanda tangan atau atribut “sign\_attachment” yang ada dari tabel “User” milik database.

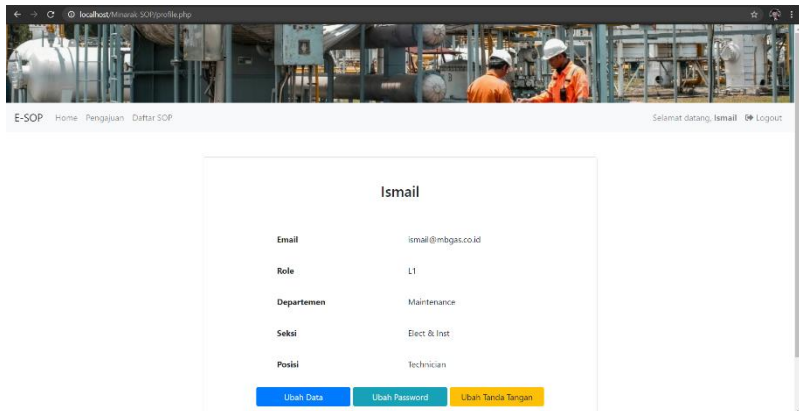




Gambar 5. 31 Form Edit User Signature

### 5.2.11 View Profile

Gambar 5.32 menampilkan antarmuka profil pengguna yang sedang mengakses web. data database dari tabel “User”. Terdapat tombol “Ubah Data” untuk menyunting data pengguna, tombol “Ubah Password” untuk menyunting password pengguna, dan tombol “Ubah Tanda Tangan” untuk menyunting tanda tangan pengguna.



Gambar 5. 32 Profile

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba terhadap aplikasi validasi SOP Minarak Brantas Gas, Inc. Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dibangun dan kesesuaian hasil eksekusi perangkat lunak dengan analisis dan perancangan perangkat lunak.

#### **6.1. Tujuan Pengujian**

Pengujian dilakukan terhadap aplikasi validasi SOP Minarak Brantas Gas, Inc. guna menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem aplikasi.

#### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memerhatikan beberapa hasil yang diharapkan berikut ini:

- a. Kemampuan aplikasi untuk membedakan role sesuai user id pada.
- b. Kemampuan aplikasi untuk memberikan akses fitur sesuai role pengguna
- c. Kemampuan aplikasi untuk memenuhi kebutuhan fungsionalitas, seperti mengajukan SOP, mengulas SOP, memvalidasi SOP, dan lainnya.
- d. Kemampuan aplikasi untuk memenuhi kebutuhan non-fungsional, yaitu dapat diakses melalui jaringan internet dan sistem dapat menampilkan view yang mudah dipahami.

#### **6.3. Skenario Pengujian**

Skenario pengujian dilakukan dengan melakukan peran sebagai L1, L2, L3, dan admin yang akan menjalankan fitur-fitur dan seluruh kebutuhan fungsional dari sistem. Langkah-langkah untuk setiap kebutuhan fungsional yaitu sebagai berikut:

1. Sebagai L1:
  - a. L1 dapat melakukan login
  - b. L1 dapat mengajukan SOP
  - c. L1 dapat memperbaiki SOP pengajuan
  - d. L1 dapat melihat daftar SOP tervalidasi
  - e. L1 dapat mengunduh bukti validasi SOP
  - f. L1 dapat melihat riwayat perubahan SOP
  - g. L1 dapat mengubah data diri
  
2. Sebagai L2:
  - a. L2 dapat melakukan login
  - b. L2 dapat mengajukan SOP
  - c. L2 dapat memperbaiki SOP pengajuan
  - d. L2 dapat merevisi SOP
  - e. L2 dapat memvalidasi SOP
  - f. L2 dapat melihat daftar SOP tervalidasi
  - g. L2 dapat mengunduh bukti validasi SOP
  - h. L2 dapat melihat riwayat perubahan SOP
  - i. L2 dapat mengubah data diri
  
3. Sebagai L3:
  - a. L3 dapat melakukan login
  - b. L3 dapat merevisi SOP
  - c. L3 dapat memvalidasi SOP
  - d. L3 dapat menolak SOP
  - e. L3 dapat melihat daftar SOP tervalidasi
  - f. L3 dapat mengunduh bukti validasi SOP
  - g. L3 dapat melihat riwayat perubahan SOP
  - h. L3 dapat mengubah data diri
  
4. Sebagai Admin:
  - a. Admin dapat melakukan login
  - b. Admin dapat mengelola data pengguna

#### 6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku aplikasi validasi SOP Minarak Brantas Gas terhadap kasus skenario uji coba. Pengujian dilakukan oleh pihak pengembang, pengguna, dan pembimbing lapangan. Pengujian dibagi menjadi 4 role user. Tabel di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

##### 1. Sebagai L1

Tabel pengujian ini ditujukan untuk user yang memiliki peran sebagai pengguna L1.

*Tabel 6. 1 Evaluasi Pengujian sebagai L1*

No.	Kriteria Pengujian	Hasil Pengujian
1	L1 dapat melakukan login	Terpenuhi
2	L1 dapat mengajukan SOP	Terpenuhi
3	L1 dapat memperbaiki SOP pengajuan	Terpenuhi
4	L1 dapat melihat daftar SOP tervalidasi	Terpenuhi
5	L1 dapat mengunduh bukti validasi SOP	Terpenuhi
6	L1 dapat melihat riwayat perubahan SOP	Terpenuhi
7	L1 dapat mengubah data diri	Terpenuhi

##### 2. Sebagai L2

Tabel pengujian ini ditujukan untuk user yang memiliki peran sebagai pengguna L2.

*Tabel 6. 2 Evaluasi Pengujian sebagai L2*

No.	Kriteria Pengujian	Hasil Pengujian
1	L2 dapat melakukan login	Terpenuhi
2	L2 dapat mengajukan SOP	Terpenuhi
3	L2 dapat memperbaiki SOP pengajuan	Terpenuhi

4	L2 dapat merevisi SOP	Terpenuhi
5	L2 dapat memvalidasi SOP	Terpenuhi
6	L2 dapat melihat daftar SOP tervalidasi	Terpenuhi
7	L2 dapat mengunduh bukti validasi SOP	Terpenuhi
8	L2 dapat melihat riwayat perubahan SOP	Terpenuhi
9	L2 dapat mengubah data diri	Terpenuhi

### 3. Sebagai L3

Tabel pengujian ini ditujukan untuk user yang memiliki peran sebagai pengguna L3.

*Tabel 6. 3 Evaluasi Pengujian sebagai L3*

No.	Kriteria Pengujian	Hasil Pengujian
1	L3 dapat melakukan login	Terpenuhi
2	L3 dapat merevisi SOP	Terpenuhi
3	L3 dapat memvalidasi SOP	Terpenuhi
4	L3 dapat menolak SOP	Terpenuhi
5	L3 dapat melihat daftar SOP tervalidasi	Terpenuhi
6	L3 dapat mengunduh bukti validasi SOP	Terpenuhi
7	L3 dapat melihat riwayat perubahan SOP	Terpenuhi
8	L3 dapat mengubah data diri	Terpenuhi

### 4. Sebagai Admin

Tabel pengujian ini ditujukan untuk user yang memiliki peran sebagai pengguna Admin.

*Tabel 6. 4 Evaluasi Pengujian sebagai Admin*

No.	Kriteria Pengujian	Hasil Pengujian
1	Admin dapat melakukan login	Terpenuhi
2	Admin dapat mengelola data pengguna	Terpenuhi

Dengan hasil pengujian pada tabel di atas, dapat disimpulkan bahwa secara keseluruhan fitur telah memenuhi kriteria-kriteria yang sudah disebutkan pada bagian-bagian sebelumnya.

*[Halaman ini sengaja dikosongkan]*



## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan pembuatan aplikasi E-SOP pada kegiatan kerja praktik di Minarak Brantas Gas, Inc adalah sebagai berikut:

1. Aplikasi yang dibangun telah sesuai dengan permintaan.
2. Dengan adanya aplikasi ini, unit kerja yang menangani proses validasi SOP di Minarak Brantas Gas menjadi mudah untuk melakukan tugasnya hal ini karena tidak diperlukan pencatatan secara manual menggunakan pdf dan tidak perlu memeriksa SOP gudang secara manual.

#### **7.2. Saran**

Saran untuk pengembangan sistem aplikasi Validasi SOP adalah sebagai berikut:

1. Untuk file yang divalidasi tidak hanya terbatas untuk file SOP, mungkin bisa file-file penting lainnya yang perlu untuk divalidasi oleh perusahaan

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] “Visual Studio Code” [Online]. Available : <https://code.visualstudio.com/docs> [Accessed 21 Mei 2021]
- [2] Microsoft, “SQL Server” [Online]. Available : <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15> [Accessed 25 Mei 2021]
- [3] Lardinois, F. (2015). Microsoft Launches Visual Studio Code, a Free CrossPlatform Code Editor for OS X, Linux and Windows. [Accessed 25 Mei 2021]
- [4] “PHP” [Online]. Available : <https://windows.php.net/> [Accessed 5 November 2021]
- [5] Apache, “XAMPP” [Online]. Available : <https://www.apachefriends.org/index.html> [Accessed 26 Mei 2021]
- [6] Minarak Brantas Gas, Inc [Online]. Available : <https://www.linkedin.com/company/lapindo-brantas-inc-/about/> [Accessed 26 Mei 2021]

*[Halaman ini sengaja dikosongkan]*

## **BIODATA PENULIS I**

Nama : Benny Hansen Lifindra  
Tempat, Tanggal Lahir : Lela, 13 Mei 2000  
Jenis Kelamin : Laki-laki  
Agama : Protestan  
Status : Belum menikah  
Alamat Asal : Jl. Mangga No. 1, Maumere, Nusa Tenggara Timur  
Alamat Surabaya : Jl. Semut Kali VI/42A  
Telepon : 081337862727  
Email : bhlifindra@gmail.com

### **PENDIDIKAN FORMAL**

2018 – sekarang : Mahasiswa S1 Informatika ITS  
2015 – 2018 : SMA Katolik St. Louis 1 Surabaya  
2012 – 2015 : SMP Katolik Frater Maumere  
2006 – 2012 : SDK Bhaktyarasa Maumere

### **KEMAMPUAN**

- Web Programming (HTML, PHP, CSS, JavaScript)
- Programming (C, C++, Python, Java)
- Database Manajemen (Oracle, MySQL)
- Software Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember Surabaya  
Angkatan : 2018  
Semester : 6 (Enam)  
IPK : 3.87 (Semester 4)

*[Halaman ini sengaja dikosongkan]*

## **BIODATA PENULIS II**

Nama : Kresna Adhi Pramana  
Tempat, Tanggal Lahir : Surabaya, 17 Maret 2000  
Jenis Kelamin : Laki-laki  
Agama : Budha  
Status : Belum menikah  
Alamat Asal : Kejawan Putih Mutiara 6 – C3/350,  
Surabaya  
Alamat Surabaya : Kejawan Putih Mutiara 6 – C3/350  
Telepon : 081230200808  
Email : kresnaadhipramana5758@gmail.com

### **PENDIDIKAN FORMAL**

2018 – sekarang : Mahasiswa S1 Informatika ITS  
2015 – 2018 : SMA Kr. Petra 2, Surabaya  
2012 – 2015 : SMP Kr. Petra 3, Surabaya  
2006 – 2012 : SD Kr. Petra 5, Surabaya

### **KEMAMPUAN**

- Web Programming (HTML, PHP, CSS, Javascript, Framework Laravel, Framework Code Igniter)
- Programming (C, C++, Python)
- Database Management (SQL Server, MySQL)
- Sistem Operasi (Windows, Linux)
- Software Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika,  
Fakultas Teknologi Elektro dan  
Informatika Cerdas, Institut Teknologi  
Sepuluh Nopember Surabaya  
Angkatan : 2018  
Semester : 6 (Enam)  
IPK : 3.81 (Semester 4)

*[Halaman ini sengaja dikosongkan]*