Implementasi Secure Coding Pada Aplikasi Pelayanan Online Lab El Shaddai

Hastori Herdianto[1], Hadi Prasetyo Utomo[2], Yiyi Supendi[3]

Teknik Informatika, Fakultas Teknik, Universitas Langlangbuana[123]

[1]hastoribekok@gmail.com

[2]hadi@informatika.unla.ac.id

[3]yiyi.supendi@gmail.com

Abstrak

Perkembangan Teknologi Informasi dan Komunikasi pada saat ini membawa kemudahan bagi kehidupan manusia. Berkembangnya aplikasi berbasis web ini menjadi tantangan sendiri bagi para pengembang aplikasi berbasis web dalam mengembangkan aspek keamanan pada aplikasi tersebut. Aspek keamanan sering dilupakan dalam penerapan Teknologi Informasi. Kerentanan biasanya disebabkan oleh kelalaian pengembang yang menyebabkan kerusakan pada sistem yang digunakan. Ada beberapa faktor yang menyebabkan kurangnya tingkat keamanan pada aplikasi website, diantaranya adalah kesalahan penulisan kode program dan misconfiguration. Kesalahan pada penulisan kode program dalam pembuatan aplikasi berbasis website sering dimanfaatkan oleh penyerang. Untuk melakukan pengamanan pada aplikas lab El-Shaddai peneliti menggunakan metode Re-Engineering untuk melakukan pengubahan dan pengorganisasian Kembali komponen komponen sistem yang dapat dilakukan atau implementasi saja, tanpa mengubah dan menghilangkan keseluruhan komponen yang ada. Dengan menemukan celah keamanan pada aplikasi, maka dari itu peneliti menerapkan secure coding untuk melakukan pengamanan pada aplikasi. Dengan melakukan analisis awal pada aplikasi lab El-Shaddai ini, dapat disimpulkan bahwa terdapat celah keamanan pada aplikasi lab El-Shaddai. Yaitu tidak adanya data input validasi, user access control, error handling, user password hashing dan data protection. Berdasarkan hasil analisis awal diatas, peneliti menerapkan beberapa aspek secure coding. Sehingga didapatkan aplikasi lab El-Shaddai yang sudah aman.

Kata kunci— Aplikasi, Lab, Pelayanan Online, Secure Coding

I. PENDAHULUAN

Perkembangan Teknologi Informasi dan Komunikasi pada saat ini membawa kemudahan bagi kehidupan manusia. Salah satu hal yang berkembang cukup pesat adalah aplikasi berbasis web. Aplikasi berbasis web dipilih karena aplikasi tersebut dapat berjalan di berbagai platform dan juga termasuk aplikasi yang ringan untuk digunakan. Berkembangnya aplikasi berbasis web ini menjadi tantangan sendiri bagi para pengembang aplikasi berbasis web dalam mengembangkan aspek keamanan pada aplikasi tersebut.

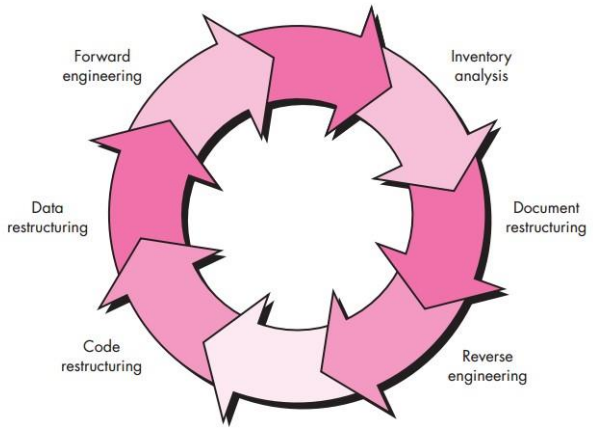
Aspek keamanan sering dilupakan dalam penerapan Teknologi Informasi. Kerentanan biasanya disebabkan oleh kelalaian pengembang yang menyebabkan kerusakan pada sistem yang digunakan. Serangan SQL Injection, Cross Site Scripting dan tidak ada penggunaan saluran terenkripsi menyebabkan pemaparan pengguna data sensitif.

Ada beberapa faktor yang menyebabkan kurangnya tingkat keamanan pada aplikasi website, diantaranya adalah kesalahan penulisan kode program dan misconfiguration. Kesalahan pada penulisan kode program dalam pembuatan aplikasi berbasis website sering dimanfaatkan oleh penyerang.

Demikian juga pada aplikasi pelayanan online lab El-shaddai terdapat informasi serta data-data penting yang bersifat sensitif. Berdasarkan hasil analisis awal yang dilakukan telah ditemukan celah keamanan pada input data registrasi online yang masih belum adanya sistem validasi data untuk mencegah serangan seperti Cross Site Scripting (Xss) pada aplikasi pelayanan online lab El-shaddai.

II. METODE

Metode penelitian yang digunakan adalah metode penelitian rekayasa dengan menggunakan pendekatan Re-Engineering. Terdapat 6 model proses atau tahapan yang dilakukan pada metode Re-Engineering akan ditunjukkan pada gambar dibawah ini :



Gambar 1 Model proses software re-engineering

Pada Gambar II.1 digambarkan bahwa model proses tersebut dilakukan secara berurutan namun, ada kalanya reverse engineering dilakukan terlebih dahulu sebelum melakukan restrukturisasi (Pressman, 2010). Pada beberapa literatur dijelaskan bahwa tahapan utama dari software re-engineering adalah reverse engineering dan forward engineering (Zamzami & Budiardjo, 2011).

Berdasarkan metode re-engineering diatas tahapan penelitian yang dilakukan adalah:

1. Inventory Analysis

Inventory anlysis adalah tahapan dalam menganalisis atau menggali informasi terkait dengan rincian deskripsi sistem yang akan dikembangkan seperti ukuran, usia, nama sistem, dan lain-lain sehingga, hal-hal yang akan direkayasa ulang akan muncul (Pressman, 2010).

2. Document Restructuring

Dokumentasi yang lemah merupakan kekurangan pada sistem yang diwariskan. Pada pengembangan sistem yang baru, menstruktur ulang dokumentasi dari sistem terdahulu bisa saja dilakukan tergantung dari kasus yang dihadapi. Sehingga pada tahapan ini, dokumentasi ulang dapat dilakukan dengan mencakup keseluruhan sistem atau sebagian saja yang diperlukan. Dokumentasi bisa saja tidak diperlukan apabila terlalu banyak yang didokumentasikan dan memakan waktu yang cukup lama (Pressman, 2010).

3. Reverse Engineering (Rekayasa Terbalik)

Reverse engineering merupakan tahapan pertama kali yang perlu dilakukan sebelum melakukan forward engineering dalam aktivitas software reengineering. Tahapan ini adalah analisis yang dilakukan untuk mengidentifikasi sistem terdahulu, sehingga dapat dipahami cara kerja sistem dan agar dapat dilakukan pengembangan (Satria, 2016). Terdapat beberapa faktor yang menjadi penyebab dilakukannya reverse engineering, yaitu (Zamzami & Budiardjo, 2011):

a. Kelengkapan desain/spesifikasi sistem kurang lengkap atau hilang.
b. Dokumentasi sistem hilang atau tidak sesuai.
c. Kompleksitas program meningkat.
d. Source code tidak terstruktur dengan baik.
e. Ketika program perlu diterjemahkan ke dalam bahasa pemrograman yang berbeda.

Dalam melakukan tahapan ini, terdapat beberapa rangkaian proses yang perlu dilakukan sebelum melanjutkan ke tahapan forward engineering. Adapun proses-proses tersebut diantaranya (Satria, 2016):

a. Menganalisis konstruksi dan cara kerja sistem terdahulu.
b. Mengecek dokumen rancangan sistem terdahulu.
c. Pengecekan terhadap kesesuaian komponen spesifikasi sistem dengan dokumen rancangan.
d. Identifikasi kelemahan pada sistem terdahulu.

4. Code Restructuring

Code Restructuring merupakan tahapan dalam menganalisis kode sumber sistem terdahulu untuk memahami fungsi-fungsi apa saja yang terdapat di dalam kode sumber. Setelah itu, kode sumber dapat distruktur ulang dengan bahasa pemrograman yang lebih modern jika diperlukan (Pressman, 2010).

5. Data Restructuring

Data Restructuring merupakan tahapan dalam menganalisis kebutuhan data sistem. Pada beberapa kasus, data restructuring diawali dengan reverse engineering. Objek data yang terdapat pada sistem dan atributnya diidentifikasi dan sturuktur data yang ada diulas untuk tujuan kualitas sistem nantinya (Pressman, 2010).

6. Forward Engineering (Rekayasa Maju)

Forward Engineering merupakan perancangan dan implementasi hasil dari reverse engineering untuk menghasilkan sistem yang baru (Satria, 2016). Tahapan ini akan meningkatkan kualitas sistem secara keseluruhan. Dalam beberapa kasus, perangkat lunak yang direkayasa ulang mengimplementasikan ulang fungsi yang telah ada dan menambahkan fungsi baru serta peningkatan pada sisi performa system (Pressman, 2010). Selain reverse engineering, forward reengineering juga memiliki rangkaian proses yang perlu dilakukan.

Adapun rangkaian proses tersebut diantaranya (Satria, 2016):

a. Penentuan spesifikasi dari sistem yang baru.
b. Merancang sistem yang baru.
c. Pembuatan sistem yang baru.
d. Pengujian.

Metode pengembangan sistem yang digunakan adalah metode Agile. Adapun tahapan yang dilakukan yaitu:

1. Perencanaan

Tahapan pengumpulan kebutuhan sistem mulai dari kebutuhan fungsional maupun non-fungsional.

2. Implementasi

Tahapan ini merupakan bagian dari proses dimana programmer melakukan pengkodean perangkat lunak.

3. Testing

Tahapan dimana perangkat lunak yang telah dibuat di tes oleh bagian kontrol kualitas agar bug yang ditemukan bisa segera diperbaiki dan kualitas perangkat lunak terjaga.

4. Dokumentasi

Tahapan ini dilakukan setelah testing perangkat lunak, Langkah selanjutnya yaitu membuat proses dokumentasi perangkat lunak untuk mempermudah maintenance kedepannya.

5. Deployment

Tahapan ini yaitu melakukan proses yang dilakukan oleh penjamin kualitas untuk menguji kualitas sistem. Setelah sistem memenuhi syarat maka perangkat lunak siap untuk dikembangkan.

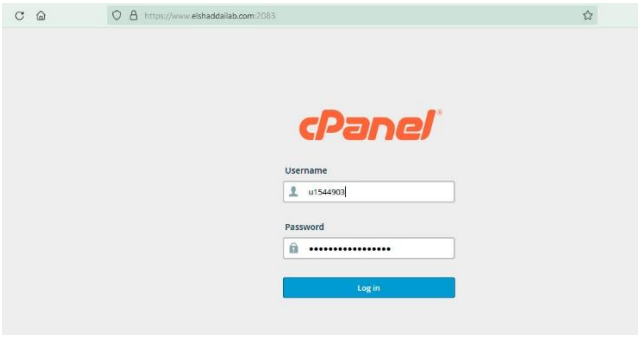
6. Pemeliharaan

Tahapan yang terakhir dalam metode agile adalah pemeliharaan atau maintenance. Tahap ini ditujukan supaya tidak ada lagi bug yang mengganggu perangkat lunak. Maka dari itu, pemeliharaan ini merupakan tahap yang sangat penting dan harus dilakukan secara berkala agar kualitas selalu terjaga.

III. HASIL DAN PEMBAHASAN

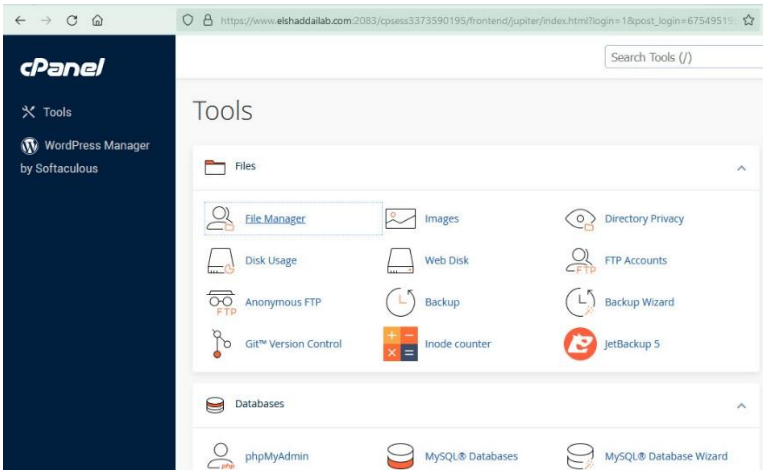
Reverse Engineering

Reverse engineering merupakan tahapan pertama kali yang perlu dilakukan sebelum melakukan *forward engineering* dalam aktivitas *software reengineering*. Tahapan ini adalah analisis yang dilakukan untuk mengidentifikasi sistem terdahulu, sehingga dapat dipahami cara kerja sistem dan agar dapat dilakukan pengembangan. Sebelum melakukan analisis pada sistem, yang perlu dilakukan adalah melakukan pengambilan source code pada hosting, untuk lebih jelasnya bisa dilihat pada gambar dibawah ini.



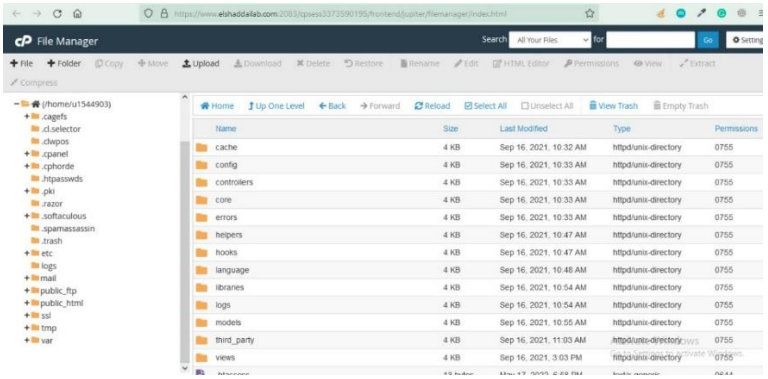
Gambar IV. 1 Login cPanel

Pada gambar IV.1 diatas merupakan halaman login cPanel sebelum melakukan pengambilan source code.



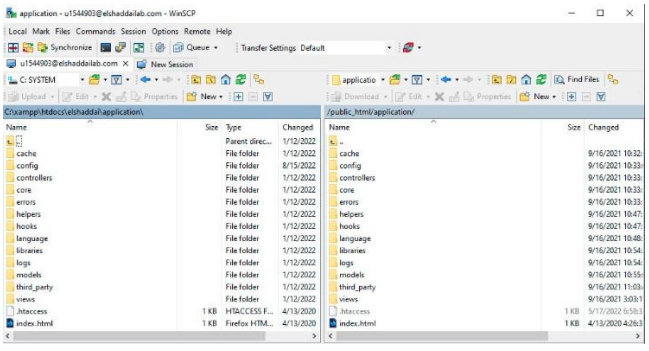
Gambar IV. 2 cPanel Dashboard

Pada gambar IV.2 diatas merupakan halaman cPanel setelah melakukan login.



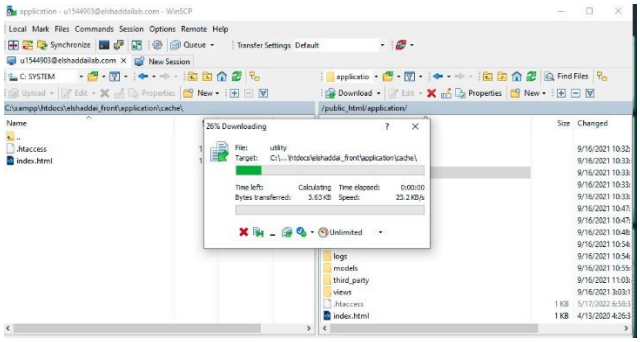
Gambar IV. 3 File Manager cPanel

Pada gambar IV.3 diatas merupakan halaman file manager yang dimana terdapat file file source code.



Gambar IV. 4 FTP Client WinSCP

Pada gambar IV.4 diatas merupakan halaman WinSCP untuk melakukan transfer file antara local dan server.



Gambar IV. 5 Transfer File

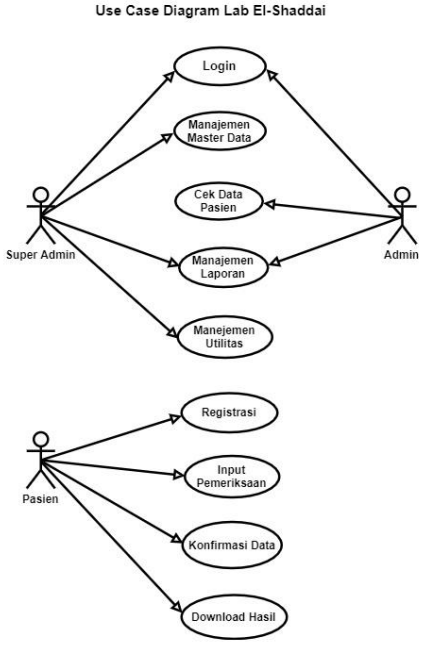
Pada gambar IV.5 diatas merupakan proses pengambilan source code pada server untuk dipindahkan ke lokal.

Inventory Analisis

Inventory anlysis adalah tahapan dalam menganalisis atau menggali informasi terkait dengan penemuan celah keamanan pada aplikasi lab El-Shaddai. Untuk lebih jelasnya bisa dilihat pada sub bab berikutnya.

Fungsionalitas Sistem Saat Ini

Pada tahap ini terdapat usecase diagram lab El-Shaddai saat ini. Untuk lebih jelasnya dapat dilihat pada gambar IV.6.



Gambar IV. 6 Use Case Diagram

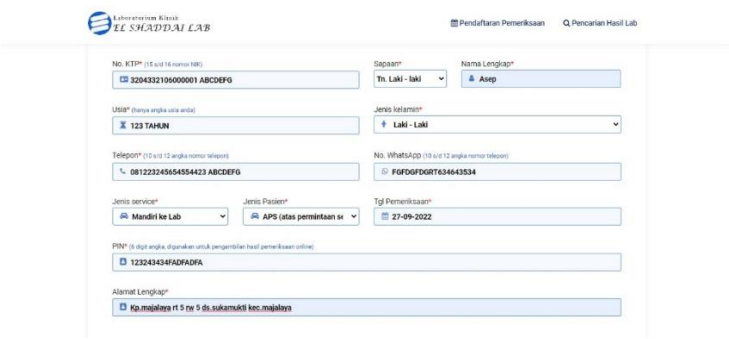
Analisis Awal Celah Keamanan Data Validasi

Adapun rincian analisis celah keamanan dijelaskan pada table IV.1.

Tabel IV. 1 Analisis Celah Keamanan Data Input Validasi

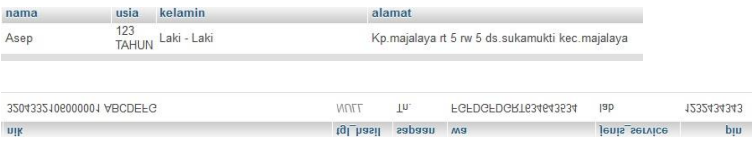
No	Aktivitas Pengujian	Realisasi Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Pasien melakukan registrasi pada form pendaftaran	Melakukan input data sesuai format	Pada form registrasi belum menerapkan data validasi, sehingga format pengisian masih bisa dilakukan dengan huruf / angka.	[ ] Diterima [X] Ditolak

A. Dokumentasi Data Input Validasi



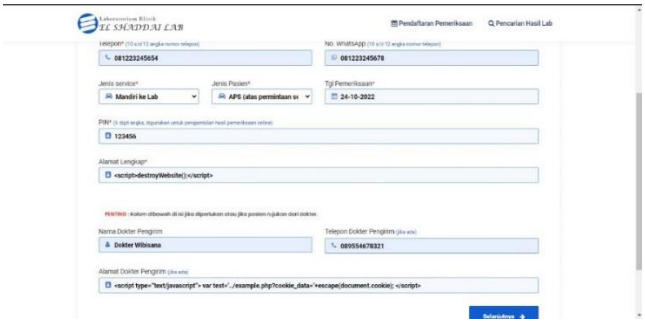
Gambar IV. 7 Form Register

Gambar IV.7 diatas merupakan halaman form register sebelum menerapkan data validasi, format pengisian data tidak sesuai dengan yang harus di input.



Gambar IV. 8 Tampilan tabel pasien

Gambar IV.8 diatas merupakan tabel data pasien sesuai dengan yang di input pada halaman register.

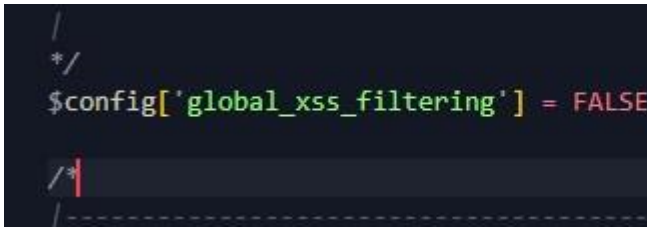


Gambar IV. 9 Form register dengan script Xss

Pada gambar IV.9 diatas merupakan tampilan form register dengan memasukkan script Xss pada bagian alamat tanpa validasi.

Gambar IV. 10 Script Xss Pada Database

Pada gambar IV.10 diatas merupakan data pasien pada database dengan field alamat yang terisi dengan script Xss.



Gambar IV. 11 Kode Xss Filtering

Pada gambar IV.11 diatas merupakan settingan untuk memfilter Xss dengan cara melakukan perubahan dengan cara *TRUE / FALSE*.

Gambar IV. 12 Source Code Form NIK

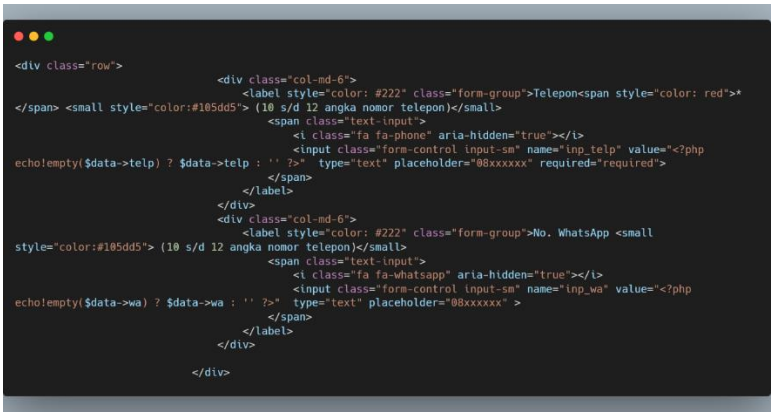
Gambar IV.12 diatas merupakan source code form NIK pada registrasi pasien yang belum menerapkan validasi data.

Gambar IV. 13 Source Code Form USIA

Gambar IV.13 diatas merupakan source code form USIA pada registrasi pasien yang belum menerapkan validasi data.

Gambar IV. 14 Source Code Form PIN

Gambar IV.14 diatas merupakan source code form PIN pada registrasi pasien yang belum menerapkan validasi data.



Gambar IV. 15 Source Code Form Telpon & Whatsapp

Gambar IV.15 diatas merupakan source code form Nomor telpon dan whatsapp pada registrasi pasien yang belum menerapkan validasi data.

Analisis Awal Celah Keamanan User Access Control

Tabel IV. 2 Analisis Celah Keamanan User Access Control Super Admin

No	Aktivitas Pengujian	Realisasi Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Pembatasan akses menu pada super admin	Super admin dapat melakukan semua akses menu pada aplikasi	Super admin berhasil melakukan akses pada semua menu	[X] Diterima [ ] Ditolak

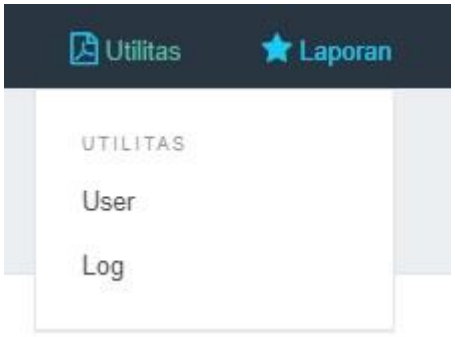
Tabel IV. 3 Analisis Celah Keamanan User Access Control Admin

No	Aktivitas Pengujian	Realisasi Yang Diharapkan	Hasil Pengujian	Kesimpulan
2.	Pembatasan akses menu pada admin	Admin hanya dapat melakukan beberapa akses menu pada aplikasi	Admin dapat melakukan akses pada semua menu	[ ] Diterima [X] Ditolak

A. Dokumentasi User Access Control

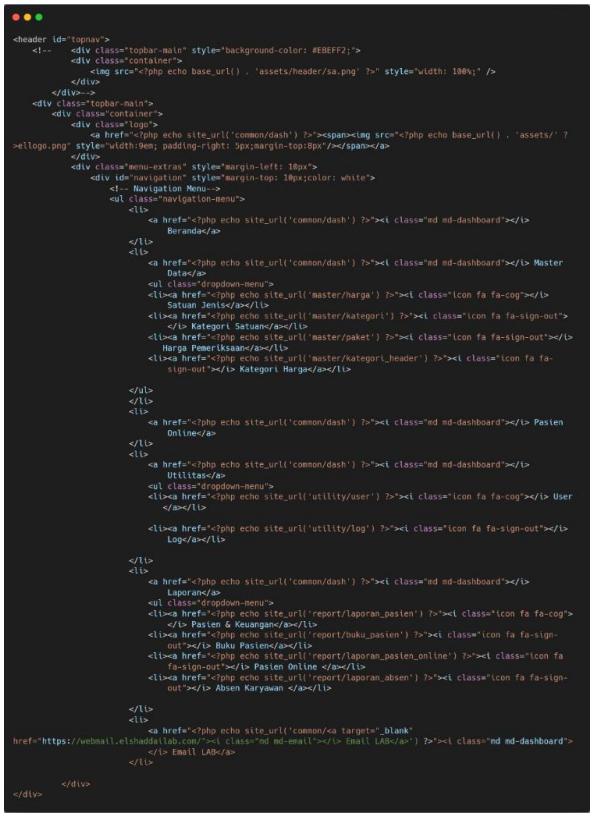
Gambar IV. 16 Menu admin

Pada gambar IV.16 diatas merupakan tampilan menu admin sebelum menerapkan hak akses kontrol, terlihat admin masih bisa mengakses semua menu seperti super admin.



Gambar IV. 17 Menu utilitas

Pada gambar IV.17 diatas merupakan tampilan menu utilitas sebelum menerapkan access control.



Gambar IV. 18 Source code menu

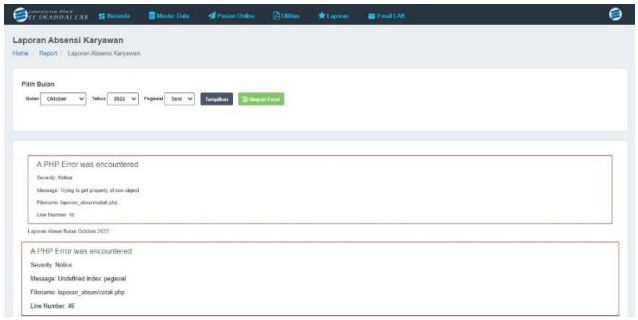
Pada gambar IV.18 diatas merupakan source code menu sebelum menerapkan access control.

Analisis Awal Celah Keamanan Error Handling

Tabel IV. 4 Analisis Celah Keamanan Error Handling

No	Aktivitas Pengujian	Realisasi Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Developer melakukan kesalahan pada struktur code	Sistem menampilkan peringatan error sesuai dengan bagian yang error serta menyimpan log error	Sistem menampilkan peringatan error namun tidak menyimpan log error	[ ] Diterima [X] Ditolak

A. Dokumentasi Error Handling



Gambar IV. 19 Error Handling

Pada gambar IV.19 diatas terlihat terdapat peringatan bahwa terjadi kesalahan/error pada menu laporan absensi karyawan.

```
public function view() {

        $param = array(
            'bulan' => $this->input->post('inp_bulan'),
            'tahun' => $this->input->post('inp_tahun'),

        );

        $submit = $this->input->post('submit');
        if ($submit == 'search') {
            $data = array(
                'title_page' => $this->modul,
                'common' => $this,
                'modul' => $this->modul,
                'title_content' => 'Filter Data',
                'param' => $param,
                'view' => 'html',
                'pegawai' => $this->user_model->select('*', array('id_user' => $param['pegawai']))->row(),
                'list_pegawai' => $this->user_model->select('*', array('type_user' => 'FO'))->result(),
                'page' => 'webadmin/report/laporan_absen/report_view',
            );
            $this->load->view('webadmin/index', $data);
        } else if ($submit == 'cetak') {
            $filename = 'Laporan_Absen_' . $param['bulan'] . '_' . $param['tahun'] . '_' . rand(1,999) . '.xls';
            header('Content-Type: application/vnd.ms-excel');
            header('Content-Disposition: attachment;filename="' . $filename . '"');
            header('Cache-Control: max-age=0');
            $data = array(
                'title_page' => $this->modul,
                'common' => $this,
                'modul' => $this->modul,
                'title_content' => 'Filter Data',
                'param' => $param,
                'view' => 'excel',
                'pegawai' => $this->user_model->select('*', array('id_user' => $param['pegawai']))->row(),
                'list_pegawai' => $this->user_model->select('*', null)->result(),
            );
            $this->load->view('webadmin/report/laporan_absen/cetak', $data);
        } else {
            redirect('report/laporan_absen');
        }
    }
```

Gambar IV. 20 Source code controller laporan absen

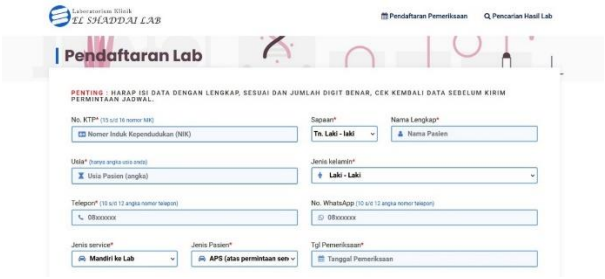
Pada gambar IV.20 diatas merupakan source code controller laporan absen karyawan yang menjadi penyebab terjadinya error.

Analisis Awal Celah Keamanan Sensitive Data Exposure

Tabel IV. 5 Analisis Celah Keamanan Sensitive Data Exposure

No	Aktivitas Pengujian	Realisasi Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Pasien melakukan registrasi	Sistem dapat melakukan enkripsi data pasien pada database	Data pasien belum terenkripsi	[ ] Diterima [X] Ditolak

A. Dokumentasi Sensitive Data Exposure



Gambar IV. 21 Form Register

nik	tgl_hasil	sapaan	wa
123456789101112	2022-02-24	Ny.	089672742895

alamat_pengirim	telp_pengirim
Ds. Bojong	08976547768

Gambar IV. 22 Tabel pasien

Pada gambar IV.22 diatas merupakan tampilan tabel data pasien yang tidak di enkripsi menggunakan algoritma *Aes-256*

```
public function process_step1() {
//        $kode = 'EL_ON' . date('m') . generate_id('EL_ON-' . date('m'), 4);
//        $data['kode_tr'] = stringCrypt($kode);

        $nik = $this->input->post('inp_nik');
        $nama = $this->input->post('inp_nama');
        if ((empty($nik)) || ($nama == '0') || (empty($nama))){
            redirect('lab/register');exit;
        }

        $data['nik'] = ($nik);
        $data['nama'] = $nama;
        $data['sapaan'] = $this->input->post('inp_sapaan');
        $data['usia'] = $this->input->post('inp_usia');
        $data['kelamin'] = ($this->input->post('inp_kelamin'));
        $data['alamat'] = $this->input->post('inp_alamat');
        $data['telp'] = $this->input->post('inp_telp');
        $data['wa'] = $this->input->post('inp_wa');
        $data['pin'] = $this->input->post('inp_pin');
        $data['jenis_service'] = $this->input->post('inp_jenis_service');
        $data['jenis_pasien'] = $this->input->post('inp_jenis_pasien');
        $data['tgl_periksa'] = date_ymd($this->input->post('inp_tgl_periksa')) . ' ' .

        $data['pengirim'] = $this->input->post('inp_pengirim');
        $data['alamat_pengirim'] = $this->input->post('inp_alamat_pengirim');
        $data['telp_pengirim'] = $this->input->post('inp_telp_pengirim');
```

Gambar IV. 23 Source Code Controller Register

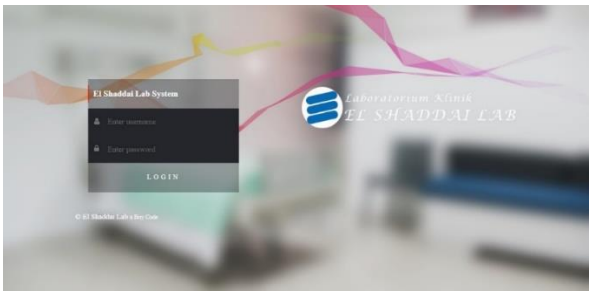
Pada gambar IV.23 diatas merupakan source code controller pada halaman registrasi yang tidak menerapkan enkripsi *Aes-256.*

Analisis Awal Celah Keamanan User Password Hashing

Tabel IV. 6 Analisis Celah Keamanan Sensitive Data Exposure

No	Aktivitas Pengujian	Realisasi Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	User melakukan login	Sistem dapat melakukan enkripsi data pengguna	Data pengguna belum terenkripsi	[ ] Diterima [X] Ditolak

A. Dokumentasi User Password Hashing



Gambar IV. 24 Halaman Login

Gambar IV. 25 Tabel User

Gambar IV.25 diatas merupakan tabel user sebelum menerapkan hashing aes-256.

```
public function process($action, $id = null) {
        //var
        $data['nama'] = $this->input->post('inp_nama');
        $data['username'] = $this->input->post('inp_username');
        $data['password'] = ($this->input->post('inp_password') );
        $data['email'] = ($this->input->post('inp_email'));
        $data['type_user'] = $this->input->post('inp_role');
        $data['date'] = date('Y-m-d H:i:s');
        $data['telp'] = ($this->input->post('inp_telp'));
        $data['kec'] = ($this->input->post('inp_kec')); //kode kec
        $data['kel'] = ($this->input->post('inp_kel')); //kode kel
        $data['tingkat'] = $this->input->post('inp_tingkat');
        $data['operator'] = $this->input->post('inp_opr');

        if ($action == 'add') {
            $data['status'] = '0';
            $res = $this->user_model->add($data);
            add_log('Tambah Data ' . $this->modul, $data['nama']);
            set_message($res, 'add-success');
        } else if ($action == 'edit') {
            $content = $this->user_model->edit($data, array('id_user' => $id));
            add_log('Ubah Data ' . $this->modul, $data['nama']);
            set_message($content, 'edit-success');
        }
        redirect('utility/user');
    }
```

Gambar IV. 26 Source Code Controller User

Gambar IV.26 diatas merupakan source code user sebelum implementasi hashing menggunakan aes-256.

Code Restructuring

Perencanaan

Daftar kebutuhan merupakan penjelasan mengenai kebutuhan apa saja yang dibutuhkan pada aplikasi. Pada tahap rekayasa kebutuhan menghasilkan beberapa kebutuhan yang dibutuhkan untuk pengembangan aplikasi. Untuk penjabaran mengenai daftar kebutuhan akan dibagi menjadi tiga, yaitu *product backlog*, kebutuhan fungsional dan kebutuhan non-fungsional.

Product Backlog

Adapun rincian *product backlog* dijelaskan pada tabel IV.7

Tabel IV. 7 Product Backlog

No	Nama *backlog*	*Priority*	*Story Points*	*Acceptance Criteria*	*Estimate (hours)*
1.	Data Input Validasi	*Highest*	5	1. Pasien mengisi data pada form pendaftaran. 2. Sistem menerapkan data validasi untuk mengisi data sesuai dengan format yang harus di isi.	3
2.	User Access Control	*Highest*	5	1. Pembatasan hak akses pada setiap pengguna.	3
3.	Error Handling	*Medium*	5	1. Pengguna melakukan kesalahan input. 2. Sistem menampilkan peringatan kesalahan.	3
4.	User Password Hashing	*Medium*	3	1. Pengguna melakukan login. 2. Sistem mengenkripsi data pengguna menggunakan Aes-256.	3
5.	Sensitive Data Exposure	*Medium*	3	1. Pasien mengisi data pendaftaran serta data pemeriksaan. 2. Sistem mengenkripsi data tersebut mengguna kan Aes-256. 3. Menampilkan hasil dekripsi di halaman pengecekan data pasien pada sisi admin.	3

Sprint Planning

Tahap sprint planning pada penelitian ini akan menjadi dua bagian, yaitu sprint goal dan sprint backlog. Pada sprint planning ini akan dibuat dua sprint. Untuk lebih jelasnya dapat dilihat pada sub bab berikutnya.

Sprint Goal

Sprint Goal akan dibagi menjadi 5 bagian yaitu, sprint goal pada sprint 1 sampai dengan sprint 5.

Sprint Backlog

Sprint backlog merupakan daftar pekerjaan yang akan dilakukan pada setiap sprint. Sprint backlog akan dibagi menjadi lima bagian, yaitu sprint backlog berdasarkan

sprint 1 sampai dengan sprint 5. Untuk lebih jelasnya dapat dilihat pada sub bab berikutnya.

Fase Sprint

Fase *Sprint* merupakan suatu tahapan kegiatan yang dilakukan sesuai dengan *product backlog* yang telah ditetapkan sebelumnya.

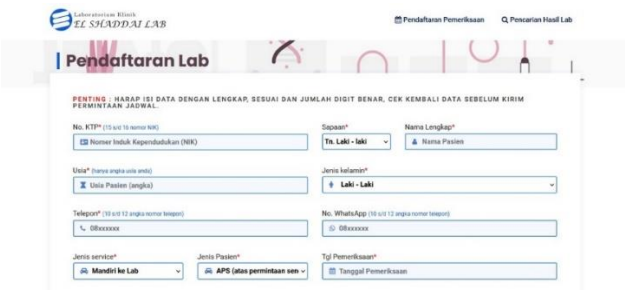
Sprint Data Input Validasi

Adapun rincian Sprint 1 dijelaskan pada Tabel IV.8

Tabel IV. 8 Sprint 1

***Sprint* 1**	
***Task Description*: Data Input Validasi**	
Kode Product Backlog	**Product Backlog Item**
SRS_LAB_01	1. Membuat Sequence Diagram *register pasien* 2. Implementasi data validasi pada form pendaftaran 3. Sistem harus mampu melakukan validasi data 4. Sistem harus mampu menampilkan peringatan ketika terdapat data yang kosong atau tidak sesuai dengan format yang harus diisi

A. Implementasi Antar muka Form Registrasi



Gambar IV. 27 Form Registrasi

Gambar IV.27 diatas merupakan tampilan halaman registrasi yang sudah menerapkan sistem data validasi yang harus diisi sesuai format pada setiap kolom nya.

B. Implementasi Kode Validasi Data

```
<div class="col-md-6">
    <label style="color: #222" class="form-group">No. KTP<span style="color: red">*</span> <small
style="color:#105dd5"> (15 s/d 16 nomor NIK)</small>
        <span class="text-input">
            <i class="fa fa-vcard" aria-hidden="true"></i>
            <input class="form-control input-sm" name="inp_nik" oninput="this.value = this.value.replace(/[^0-9.]/g,
'').replace(/(\..*?)\..*/g, '$1');" onkeypress="return isNumberKey(event)" value="<?php echo!empty($data->nik) ?
$data->nik : '' ?>" pattern=".{15,16}" maxlength="16" type="text" placeholder="Nomer Induk Kependudukan (NIK)"
required="required">
        </span>
        </label>
        </div>
```

Gambar IV. 28 Implementasi Data Validasi NIK

Gambar IV.28 diatas merupakan source code pada form *Nik* yang sudah menerapkan validasi data bahwa harus diisi sesuai format penulisan *Nik* pada umumnya.

```
<div class="col-md-6">
    <label style="color: #222" class="form-group">Usia<span style="color: red">*</span> <small
style="color:#105dd5"> (hanya angka usia anda)</small>
        <span class="text-input">
        <i class="fa fa-hourglass" aria-hidden="true"></i>
        <input class="form-control input-sm" name="inp_usia" type="text" value="<?php echo!empty($data->usia) ?
$data->usia : '' ?>" pattern=".{2,2}" maxlength="2" oninput="this.value = this.value.replace(/[^0-9.]/g,
'').replace(/(\..*?)\..*/g, '$1');" placeholder="Usia Pasien (angka)" required="required">
        </span>
        </label>
        </div>
```

Gambar IV. 29 Implementasi Data Validasi Usia

Gambar IV.29 diatas merupakan source code pada form *Usia* yang sudah menerapkan validasi data bahwa harus diisi sesuai format penulisan *Usia* pada umumnya.

```
<div class="row">
    <div class="col-md-6">
        <label style="color: #222" class="form-group">Telepon<span style="color: red">*</span> <small
style="color:#105dd5"> (10 s/d 12 angka nomor telepon)</small>
        <span class="text-input">
        <i class="fa fa-phone" aria-hidden="true"></i>
        <input class="form-control input-sm" name="inp_telp" value="<?php echo!empty($data->telp) ? $data->telp :
'' ?>" pattern=".{10,}" maxlength="13" oninput="this.value = this.value.replace(/[^0-9.]/g,
'').replace(/(\..*?)\..*/g, '$1');" onkeypress="return isNumberKey(event)" type="text" placeholder="08xxxxxx"
required="required">
        </span>
        </label>
        </div>
    <div class="col-md-6">
        <label style="color: #222" class="form-group">No. WhatsApp <small style="color:#105dd5"> (10 s/d 12 angka
nomor telepon)</small>
        <span class="text-input">
        <i class="fa fa-whatsapp" aria-hidden="true"></i>
        <input class="form-control input-sm" name="inp_wa" value="<?php echo!empty($data->wa) ? $data->wa : '' ?
>" pattern=".{10,}" maxlength="13" oninput="this.value = this.value.replace(/[^0-9.]/g, '').replace(/(\..*?)\..*/g,
'$1');" onkeypress="return isNumberKey(event)" type="text" placeholder="08xxxxxx" >
        </span>
        </label>
        </div>
        </div>
```

Gambar IV. 30 Implementasi Data Validasi Nomor Telpon

Gambar IV.30 diatas merupakan source code pada form Nomor Telpon dan Whatsapp yang sudah menerapkan validasi data bahwa harus diisi sesuai format penulisan nomor pada umumnya.

C. Implementasi Kode Xss Filtering

```
*/
$config['global_xss_filtering'] = true;

/*
```

Gambar IV. 31 Implementasi kode xss filtering

Pada gambar IV.31 diatas merupakan implementasi kode untuk memfilter jika ada script Xss dengan mengubah menjadi *true.*

Gambar IV. 32 Xss filtering true pada DB

Pada gambar IV.32 diatas merupakan hasil filtering Xss dan implementasi data input validasi, jika terdapat script xss yang masuk, maka sudah otomatis terhapus oleh sistem.

D. Sprint Review

Tabel IV. 9 Sprint Review Data Validasi

Kebutuhan	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Data Input Validasi	Pasien melakukan registrasi	Sistem akan melakukan validasi data apakah sesuai format yang harus diisi.	Sistem berhasil melakukan validasi data apakah sesuai format yang harus diisi.	Sesuai yang diharapkan

Tabel IV.9 diatas merupakan tabel sprint review, dimana didalamnya terdapat hasil pengujian pada tahapan fase sprint backlog diatas.

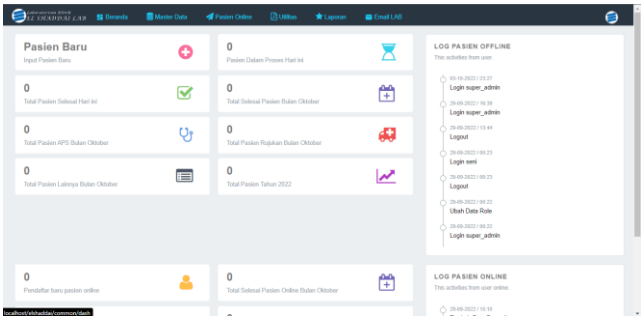
Sprint User Access Control

Adapun rincian Sprint 2 dijelaskan pada Tabel IV.10

Tabel IV. 10 Sprint 2

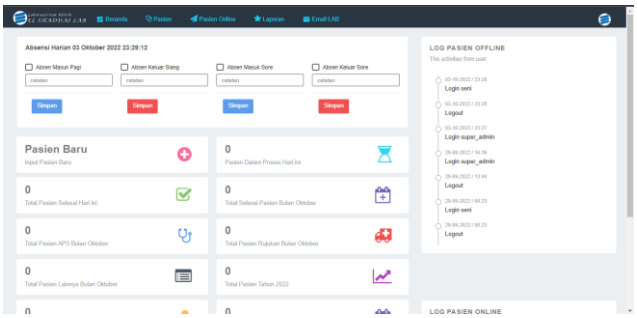
Sprint 2	
***Task Description*: Access Control**	
Kode Product Backlog	**Product Backlog Item**
SRS_LAB_02	1. Membuat Sequence Diagram *User Access Control* 2. Implementasi hak akses pengguna 3. Sistem harus mampu melakukan pembatasan hak akses

A. Implementasi Antar Muka User Access Control



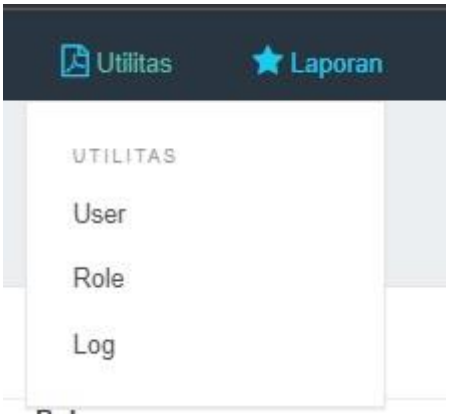
Gambar IV. 33 Access Control Super Admin

Gambar IV.33 diatas merupakan tampilan halaman akses menu Super Admin.



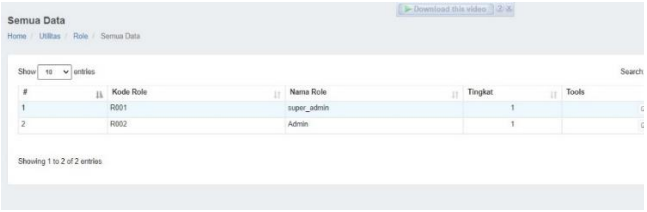
Gambar IV. 34 Access Control Admin

Gambar IV.34 diatas merupakan tampilan halaman akses menu Admin.



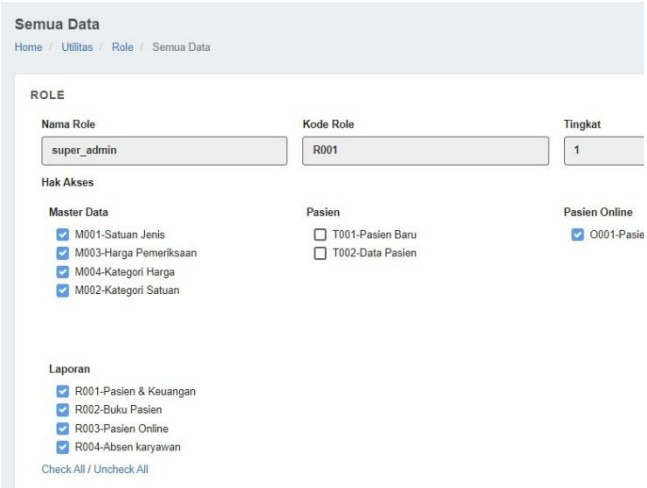
Gambar IV. 35 Tampilan menu utilitas setelah penambahan access control

Pada gambar IV.35 diatas merupakan tampilan menu utilitas setelah menambahkan access control.



Gambar IV. 36 Tampilan menu role

Pada gambar IV.36 diatas merupakan tampilan menu role.



Gambar IV. 37 Tampilan setting hak akses

Pada gambar IV.37 diatas merupakan tampilan untuk membuat setiap user bisa mengakses apa saja sesuai hak nya.

B. Implementasi Kode User Access Control

```
<?php
    $array_icon = array('database', 'stethoscope', 'paper-plane', 'file-pdf-o', 'star');
    $arx = array('master', 'transaksi', 'report', 'utilitas');
    $no = 0;
    $arr_list_menu = explode(',', $common->list_menu);
    foreach ($common->menu as $menu) {
        if ($common->countmenu[0][strtolower($menu->nama_list)] == 1) {
            ?>
            <li class="has-submenu">
                <a href="#"><i class="fa fa-<?php echo $array_icon[$no] ?>"></i> <?php echo
                    $menu->nama_list ?></a>
                <ul class="submenu">
                    <li>
                        <span><?php echo $menu->nama_list ?></span>
                    </li>
                    <?php
                    foreach ($common->submenu as $submenu) {
                        if ($submenu->kategori == $menu->kode_list && in_array($submenu-
                    >kode_modul, $arr_list_menu)) {
                            echo '<li><a href="' . site_url($submenu->link) . '">' . $submenu-
                    >nama_modul . '</a></li>';
                        }
                    }
                    ?>
                </ul>
            </li>
            <?php
        } $no++;
    }
?>
```

Gambar IV. 38 Implementasi Access Control

Gambar IV.38 diatas merupakan source code implementasi Access Control pada setiap pengguna sesuai dengan hak akses nya.

C. Sprint Review User Access Control

Tabel IV. 11 Sprint Review User Access Control

Kebutuhan	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
User Access Control	User melakukan login dan mengakses aplikasi lab El-Shaddai	Sistem akan melakukan pembatasan hak akses pada setiap user.	Sistem berhasil melakukan pembatasan hak akses	Sesuai yang diharapkan

Tabel IV.11 diatas merupakan tabel sprint review, dimana didalamnya terdapat hasil pengujian pada tahapan fase sprint backlog diatas.

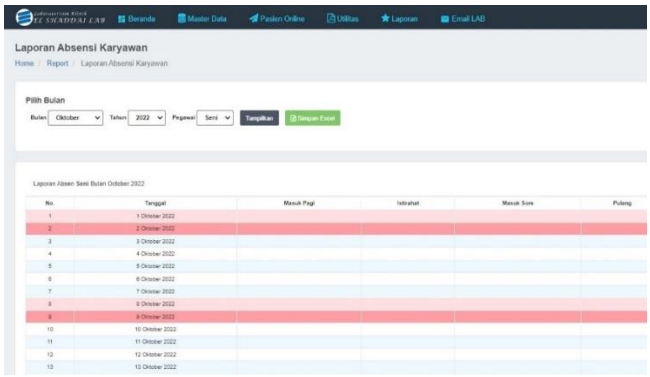
Sprint Error Handling

Adapun rincian Sprint 3 dijelaskan pada Tabel IV.12

Tabel IV. 12 Sprint 3

***Sprint* 3**	
***Task Description*: Error Handling**	
Kode Product Backlog	**Product Backlog Item**
SRS_LAB_03	1. Membuat Sequence Diagram *Error Handling* 2. Implementasi Error Handling 3. Sistem harus melakukan peringatan kepada user / developer jika terjadinya error seperti kesalahan input

A. Implementasi Antar Muka Error Handling



Gambar IV. 39 Implementasi antar muka laporan absen

Gambar IV.39 diatas merupakan halaman laporan absen karyawan yang sebelumnya terjadi error.

B. Implementasi Kode Error Handling

```
public function view() {

        $param = array(
            'bulan' => $this->input->post('inp_bulan'),
            'tahun' => $this->input->post('inp_tahun'),
            'pegawai' => $this->input->post('inp_nama_pegawai')
        );

        $submit = $this->input->post('submit');
        if ($submit == 'search') {
            $data = array(
                'title_page' => $this->modul,
                'common' => $this,
                'modul' => $this->modul,
                'title_content' => 'Filter Data',
                'param' => $param,
                'view' => 'html',
                'pegawai' => $this->user_model->select('*', array('id_user' => $param['pegawai']))->row(),
                'list_pegawai' => $this->user_model->select('*', array('type_user' => 'FO'))->result(),
                'page' => 'webadmin/report/laporan_absen/report_view',
            );
            $this->load->view('webadmin/index', $data);
        } else if ($submit == 'cetak') {
            $filename = 'Laporan_Absen_' . $param['bulan'] . '_' . $param['tahun'] . '_' . rand(1,999) . '.xls';
            header('Content-Type: application/vnd.ms-excel');
            header('Content-Disposition: attachment;filename="' . $filename . '"');
            header('Cache-Control: max-age=0');
            $data = array(
                'title_page' => $this->modul,
                'common' => $this,
                'modul' => $this->modul,
                'title_content' => 'Filter Data',
                'param' => $param,
                'view' => 'excel',
                'pegawai' => $this->user_model->select('*', array('id_user' => $param['pegawai']))->row(),
                'list_pegawai' => $this->user_model->select('*', null)->result(),
            );
            $this->load->view('webadmin/report/laporan_absen/cetak', $data);
        } else {
            redirect('report/laporan_absen');
        }
    }
```

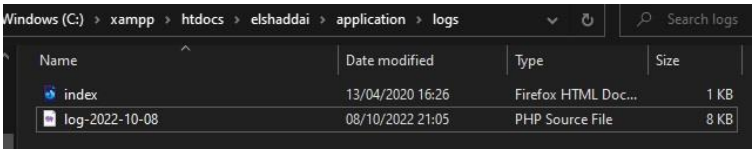
Gambar IV. 40 Implementasi Error Handling

Gambar IV.40 diatas merupakan source code controller laporan absen. Yang menjadi penyebab terjadinya error ternyata pada bagian yang terdapat panah merah sebelumnya tidak terisi oleh developer sehingga tidak menemukan field nama pegawai.

```
/*
|--------------------------------------------------------------------------
| Error Logging Threshold
|--------------------------------------------------------------------------
|
| If you have enabled error logging, you can set an error threshold to
| determine what gets logged. Threshold options are:
| You can enable error logging by setting a threshold over zero. The
| threshold determines what gets logged. Threshold options are:
|
|   0 = Disables logging, Error logging TURNED OFF
|   1 = Error Messages (including PHP errors)
|   2 = Debug Messages
|   3 = Informational Messages
|   4 = All Messages
|
| For a live site you'll usually only enable Errors (1) to be logged otherwise
| your log files will fill up very fast.
|
*/
$config['log_threshold'] = 1;
```

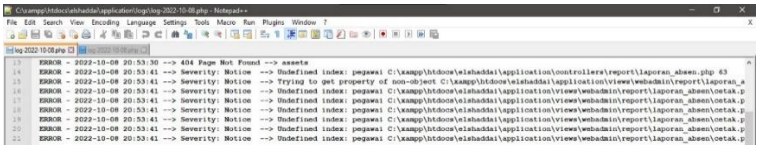
Gambar IV. 41 Implementasi Logging Error

Gambar IV.41 diatas merupakan implementasi logging error untuk menyimpan catatan error yang terjadi.



Gambar IV. 42 Direktori Logging

Gambar IV.42 diatas merupakan direktori penyimpanan logging error / catatan error yang terjadi.



Gambar IV. 43 Isi catatan error

Gambar IV.43 diatas merupakan tampilan catatan error yang tersimpan pada direktori diatas.

C. Sprint Review Error Handling

Tabel IV. 13 Sprint Review Error Handling

Kebutuhan	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Error Handling	Developer melakukan kesalahan pada struktur kode	Sistem akan menampilkan peringatan error dan menyimpan log error	Sistem berhasil menampilkan peringatan error dan menyimpan log error	Sesuai yang diharapkan

Tabel IV.13 diatas merupakan tabel sprint review, dimana didalamnya terdapat hasil pengujian pada tahapan fase sprint backlog diatas.

Sprint User Password Hashing

Adapun rincian Sprint 4 dijelaskan pada Tabel IV.14

Tabel IV. 14 Sprint 4

Sprint 4	
***Task Description*: User Password Hashing**	
Kode Product Backlog	**Product Backlog Item**
SRS_LAB_04	2. Membuat Sequence Diagram *User Password Hashing* 3. Implementasi Password Hashing 4. Sistem harus mampu melakukan hashing password user

A. Implementasi Antar Muka User Password Hashing

nama	username	password
Super Admin	super_admin	$2y$10$m82OqXowOEmgxPoo17kLCuo2hdPL2hFW6HwJGdWei.O...

Gambar IV. 44 Tabel User Setelah Hashing

Gambar IV.44 diatas merupakan tabel user setelah melakukan implementasi hash *aes-256.*

B. Implementasi Kode User Password Hashing

```
public function process($action, $id = null) {
        //var
        $data['nama'] = $this->input->post('inp_nama');
        $data['username'] = $this->input->post('inp_username');
        $data['password'] = password_hash($this->input->post('inp_password'), PASSWORD_DEFAULT);
        $data['email'] = $this->input->post('inp_email');
        $data['type_user'] = $this->input->post('inp_role');
        $data['date'] = date('Y-m-d H:i:s');
        $data['telp'] = $this->input->post('inp_telp');
        $data['kec'] = $this->input->post('inp_kec'); //kode kec
        $data['kel'] = $this->input->post('inp_kel'); //kode kel
        $data['tingkat'] = $this->input->post('inp_tingkat');
        $data['operator'] = $this->input->post('inp_opr');
        if ($action == 'add') {
            $data['status'] = '0';
            $res = $this->user_model->add($data);
            add_log('Tambah Data ' . $this->modul, $data['nama']);
            set_message($res, 'add-success');
        } else if ($action == 'edit') {
            $content = $this->user_model->edit($data, array('id_user' => $id));
            add_log('Ubah Data ' . $this->modul, $data['nama']);
            set_message($content, 'edit-success');
        }
        redirect('utility/user');
    }
```

Gambar IV. 45 Implementasi User Password Hashing

Gambar IV.45 diatas merupakan implementasi fungsi hash untuk melakukan hash password user.

```
public function process_login() {

        $username = $this->input->post('inpUsername');
        $password = $this->input->post('inpPassword');
        $param = array(
            'username' => $username,
            // 'password' => $password
        );

        $count_user = $this->user_model->select('*', $param, null, null, null)->num_rows();
        if ($count_user == 1) {
            $data_user = $this->user_model->select('*', $param, null, null, null)->row();
            if ($data_user->status == 0) {
                set_message(0, 'akun belum-aktip', 1);
                redirect('common/auth');
            } else {
                if (password_verify($password, $data_user->password)) {
                    $sessionData['app'] = trim($this->app);
                    $sessionData[$this->app . 'id_user'] = trim($data_user->id_user);
                    $sessionData[$this->app . 'is_login'] = TRUE;

                    $this->session->set_userdata($sessionData);
                    add_log('Login ', $param['username']);
                    redirect('common/dash');
                }else{
                    set_message(0, 'wrong-password', 1);
                    redirect('common/auth');
                }

            }
```

Gambar IV. 46 Proses Login Password Verify

Gambar IV.46 diatas merupakan proses login menggunakan password verify untuk mengenali bahwa data user yang login tersebut ada pada database atau tidak.

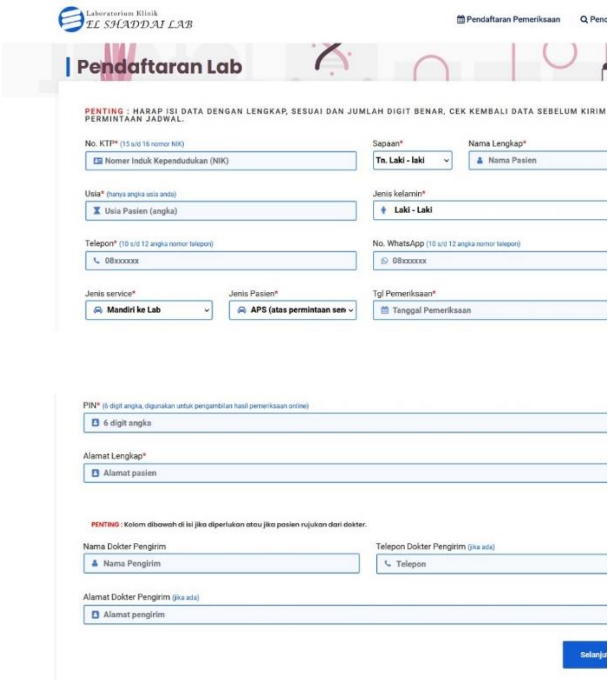
Sprint Sensitive Data Exposure

Adapun rincian Sprint 5 dijelaskan pada Tabel IV.16

Tabel IV. 15 Sprint 5

Sprint 5	
***Task Description*: Sensitive Data Exposure**	
Kode Product Backlog	**Product Backlog Item**
SRS_LAB_05	1. Membuat Sequence Diagram *register pasien* 2. Implementasi algoritma AES-256 3. Sistem harus mampu melakukan enkripsi data pasien 4. Sistem harus mampu menampilkan hasil dekripsi pada sisi admin

A. Implementasi Antar muka Form Registrasi



Gambar IV. 47 Implementasi Antarmuka Form Register

B. Implementasi Kode Enkripsi AES-256

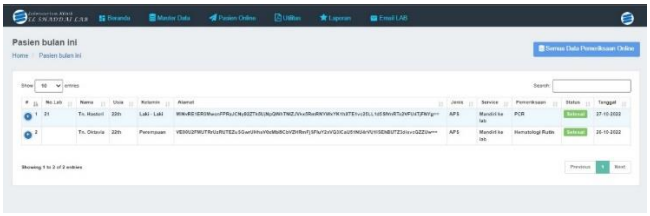
```php
function stringCrypt($string, $action = "e") {
    $secret_key = '6818f23eef19d38dad1d2729991f6368';
    $secret_iv = '0ac35e3823616c810f86e526d1ed59e7';

    $output = false;
    $encrypt_method = "AES-256-CBC";
    $key = hash("sha256", $secret_key);
    $iv = substr(hash("sha256", $secret_iv), 0, 16);

    if ($action == "e") {
        $output = base64_encode(openssl_encrypt($string, $encrypt_method, $key, 0, $iv));
    } else if ($action == "d") {
        $output = openssl_decrypt(base64_decode($string), $encrypt_method, $key, 0, $iv);
    }

    return $output;
}
```

Gambar IV. 48 Implementasi AES 256

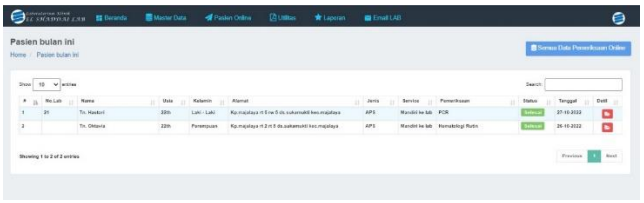
```php
public function process_step1() {
    $kode = 'EL_ON-' . date('m') . generate_id('EL_ON-' . date('m'), 4);
    $data['kode_tr'] = stringCrypt($kode);

    $nik = $this->input->post('inp_nik');
    $nama = $this->input->post('inp_nama');
    if ((empty($nik)) || ($nama == '0') || (empty($nama))){
        redirect('lab/register');exit;
    }

    $data['nik'] = stringCrypt($nik);
    $data['nama'] = $nama;
    $data['sapaan'] = $this->input->post('inp_sapaan');
    $data['usia'] = $this->input->post('inp_usia');
    $data['kelamin'] = $this->input->post('inp_kelamin');
    $data['alamat'] = stringCrypt($this->input->post('inp_alamat'));
    $data['telp'] = stringCrypt($this->input->post('inp_telp'));
    $data['wa'] = stringCrypt($this->input->post('inp_wa'));
    $data['pin'] = stringCrypt($this->input->post('inp_pin'));
    $data['jenis_service'] = $this->input->post('inp_jenis_service');
    $data['jenis_pasien'] = $this->input->post('inp_jenis_pasien');
    $data['tgl_periksa'] = stringCrypt(date_ymd($this->input->post('inp_tgl_periksa')) . ' ' . date('H:i:s'));

    $data['pengirim'] = $this->input->post('inp_pengirim');
    $data['alamat_pengirim'] = stringCrypt($this->input->post('inp_alamat_pengirim'));
    $data['telp_pengirim'] = stringCrypt($this->input->post('inp_telp_pengirim'));
```

Gambar IV. 49 Kode Form Registrasi Sesudah Enkripsi

Gambar IV. 50 Tabel Pasien Setelah Enkripsi



Gambar IV. 51 Data Pasien Admin Sebelum Dekripsi



Gambar IV. 52 Data Pasien Admin Setelah Dekripsi

Forward Engineering

Penentuan Spesifikasi Dari Sistem Yang Baru

Pada tahapan ini terdapat penentuan spesifikasi dari sistem yang baru, untuk lebih jelasnya bisa dilihat pada tabel dibawah ini.

Tabel IV. 16 Spesifikasi Sistem

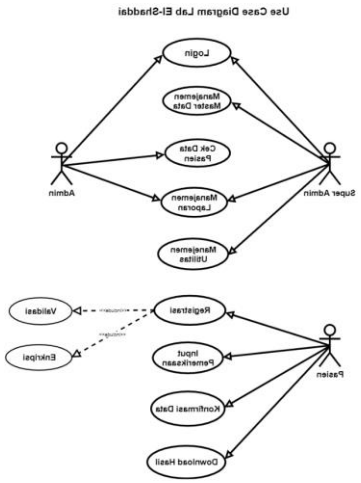
No	Nama *backlog*	*Priority*	*Story Points*	*Acceptance Criteria*	*Estimate (hours)*
1.	Data Input Validasi	*Highest*	5	1. Pasien mengisi data pada form pendaftaran. 2. Sistem menerapkan data validasi untuk mengisi data sesuai dengan format yang harus di isi.	3
2.	User Access Control	*Highest*	5	1. Pembatasan hak akses	3

				pada setiap pengguna.	
3.	Error Handling	*Medium*	5	1. Pengguna melakukan kesalahan input. 2. Sistem menampilkan peringatan kesalahan.	3
4.	User Password Hashing	*Medium*	3	1. Pengguna melakukan login. 2. Sistem mengenkripsi data pengguna menggunakan Aes-256.	3
5.	Sensitive Data Exposure	*Medium*	3	1. Pasien mengisi data pendaftaran serta data pemeriksaan. 2. Sistem mengenkripsi data tersebut menggunakan Aes-256. 3. Menampilkan hasil dekripsi di halaman pengecekan data pasien pada sisi admin.	3

Merancang Sistem Yang Baru

Pada tahapan ini akan menjelaskan perancangan sistem yang baru, dimana didalamnya terdapat proses use case, sequence diagram dan class diagram, untuk jelasnya bisa dilihat pada gambar berikut.

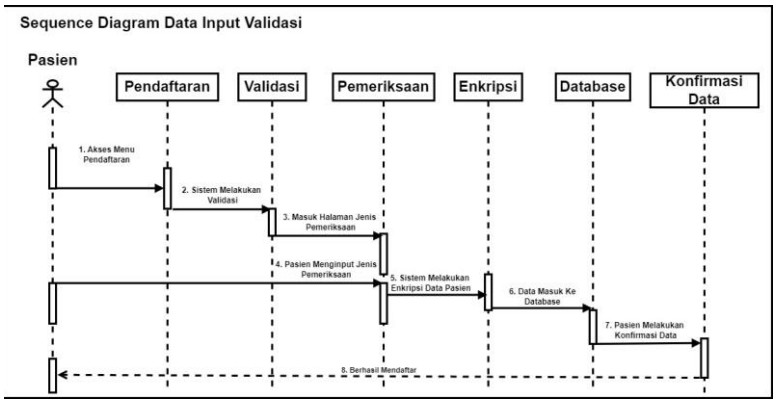
A. Use Case Diagram



Gambar IV. 53 Use Case Diagram Lab

Pada gambar IV.53 diatas merupakan perancangan usecase sistem yang baru, dimana pada proses registrasi terdapat proses validasi dan enkripsi.

A. Sequence Diagram Data Input Validasi



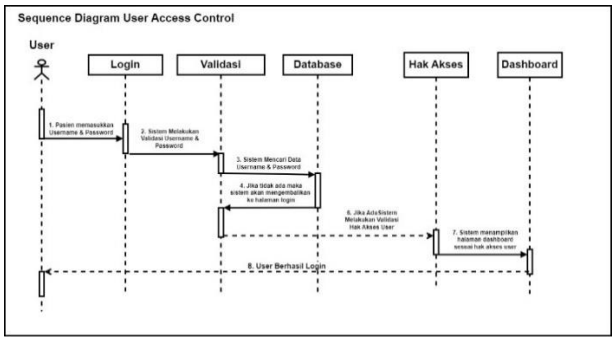
Gambar IV. 54 Sequence Diagram Data Input Validasi

Berikut merupakan alur dari diagram sequence Data input validasi :

1. Pasien melakukan registrasi online pada form pendaftaran yang sudah diterapkan validasi data.
2. Pasien melakukan input jenis pemeriksaan pada halaman selanjutnya.
3. Sistem melakukan enkripsi data pasien setelah melakukan input jenis pemeriksaan dan penyimpanan pada database.

4. Pasien kemudian melakukan konfirmasi data.
5. Pasien berhasil mendaftar.

B. Sequence Diagram User Access Control

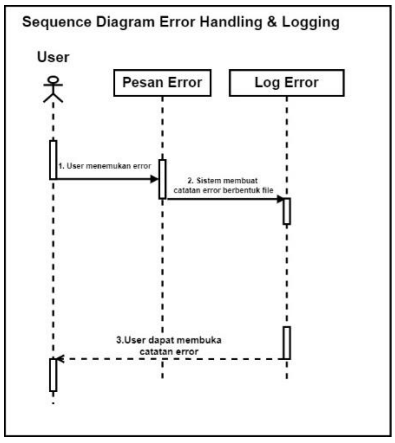


Gambar IV. 55 Sequence Diagram Access Control

Berikut merupakan alur dari diagram sequence User Access Control:

1. User melakukan login terlebih dahulu dengan memasukkan username dan password.
2. Sistem melakukan validasi data username dan password.
3. Sistem melakukan pencarian data user pada database.
4. Jika tidak terdapat data user maka sistem akan mengembalikan ke halaman login.
5. Jika data user ada pada database maka sistem melakukan validasi hak akses user.
6. Sistem menampilkan halaman dashboard dan user berhasil login.

C. Sequence Diagram Error Handling

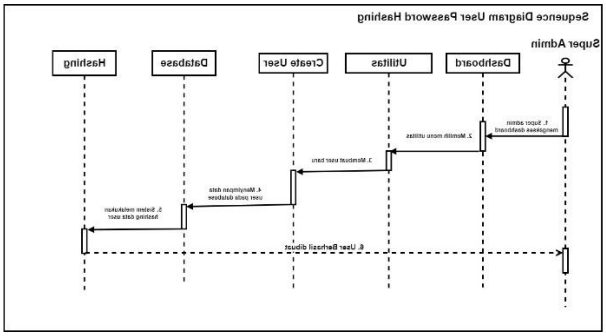


Gambar IV. 56 Sequence Diagram Error Handling

Berikut merupakan alur dari diagram sequence Error Handling:

1. User menemukan error pada aplikasi.
2. Sistem melakukan pembuatan catatan error berbentuk file.
3. User dapat membuka file catatan error untuk mengetahui lokasi error pada bagian tersebut.

D. Sequence Diagram User Password Hashing

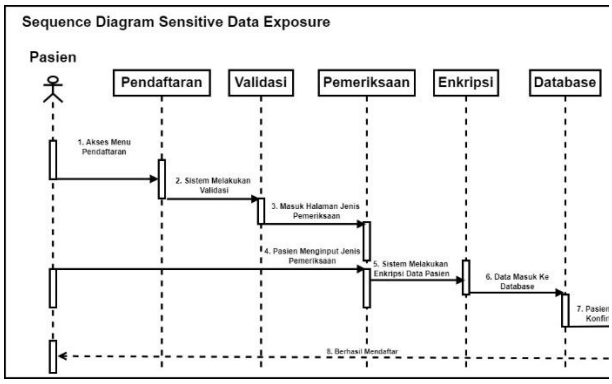


Gambar IV. 57 Sequence Diagram User Password Hashing

Berikut merupakan alur dari diagram sequence User Password Hashing:

1. Pasien melakukan registrasi online pada form pendaftaran yang sudah diterapkan validasi data.
2. Pasien melakukan input jenis pemeriksaan pada halaman selanjutnya.
3. Sistem melakukan enkripsi data pasien setelah melakukan input jenis pemeriksaan dan penyimpanan pada database.
4. Pasien kemudian melakukan konfirmasi data.
5. Pasien berhasil mendaftar.

E. Sequence Diagram Sensitive Data Exposure



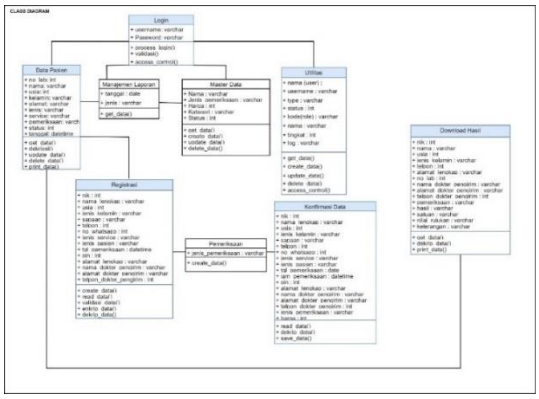
Gambar IV. 58 Sequence Diagram Sensitive Data Exposure

Berikut merupakan alur dari diagram sequence Sensitive Data Exposure:

1. Pasien melakukan registrasi online pada form pendaftaran yang sudah diterapkan validasi data.
2. Pasien melakukan input jenis pemeriksaan pada halaman selanjutnya.
3. Sistem melakukan enkripsi data pasien setelah melakukan input jenis pemeriksaan dan penyimpanan pada database.
4. Pasien kemudian melakukan konfirmasi data.
5. Pasien berhasil mendaftar.

F. Class Diagram

Berikut adalah tahapan *Class Diagram*, pada bagian yang diberi warna biru merupakan class yang sudah diberi penambahan fungsi yang baru. Untuk lebih jelasnya bisa dilihat pada gambar dibawah ini.



Gambar IV. 59 Class Diagram

Pembuatan Sistem Yang Baru

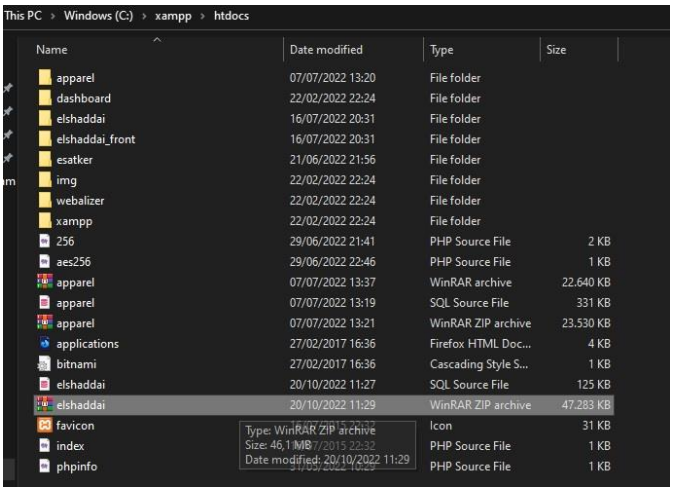
Pada tahapan ini merupakan hasil reverse dari tahapan sprint yang sudah dilakukan diatas.

Tabel IV. 17 Tabel Pembuatan Sistem Baru

A.	Implementasi *Data Input Validasi* sudah dilakukan pada tahapan sprint IV.3.7.
B.	Implementasi *User Access Control* sudah dilakukan pada tahapan sprint IV.3.8.
C.	Implementasi *Error Handling & Logging* sudah dilakukan pada tahapan sprint IV.3.9.
D.	Implementasi *User Password Hashing* sudah dilakukan pada tahapan sprint IV.3.10.
E.	Implementasi *Sensitive Data Exposure* sudah dilakukan pada tahapan sprint IV.3.11.

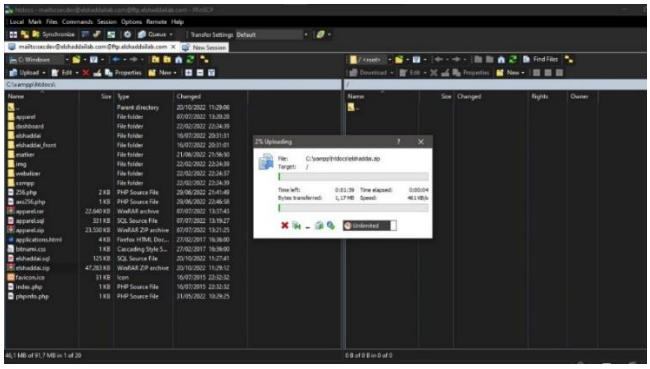
Deployment

Pada tahapan ini merupakan proses implementasi hasil dari reverse engineering dalam meningkatkan kualitas sistem secara keseluruhan. Dengan cara melakukan rekayasa ulang, mengimplementasikan ulang fungsi yang telah ada dan menambahkan fungsi baru serta peningkatan pada sisi keamanan sistem. Berikut merupakan tahapan deployment dalam melakukan forward engineering.



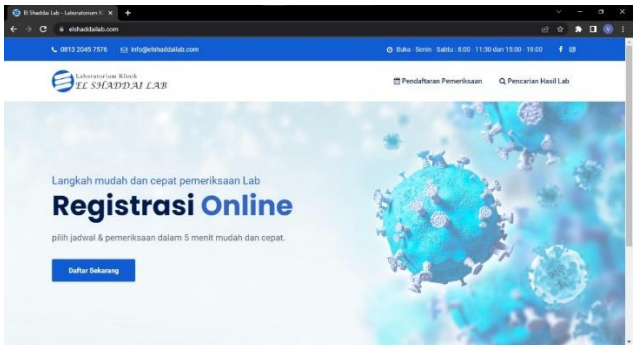
Gambar IV. 60 File website elshaddai

Pada gambar IV.60 diatas merupakan source code lab El-Shaddai pada direktori local.



Gambar IV. 61 Proses pemindahan file pada local ke server

Pada gambar IV.61 diatas merupakan proses pemindahan file source code dari local ke server menggunakan WinSCP.



Gambar IV. 62 Tampilan awal lab El-Shaddai setelah proses hosting

Pada gambar IV.62 diatas merupakan tampilan awal lab El-Shaddai setelah proses hosting dengan nama domain *elshaddailab.com.*

IV. SIMPULAN

Berdasarkan hasil penelitian yang telah diuraikan pada bab sebelumnya, dapat disimpulkan beberapa hal sebagai berikut:

1. Berdasarkan hasil analisis pada aplikasi lab El-Shaddai, terdapat celah keamanan pada aplikasi tersebut. Yaitu tidak adanya data input validasi, user access control, error handling, user password hashing dan data protection.
2. Berdasarkan hasil penerapan aspek secure coding, didapatkan aplikasi lab El-Shaddai yang lebih aman.

REFERENSI

[1] Arrhioui, K., Mbarki, S., Betari, O., Roubi, S., & Erramdani, M. (2017). A model driven approach for modeling and generating php codeigniter based applications. *Transactions on Machine Learning and Artificial Intelligence*, *5*(4).

[2] Das, R., & Saikia, L. P. (2016). Comparison of Procedural PHP with Codeigniter and Laravel Framework. *International Journal of Current Trends in Engineering & Research*, *2*(6), 42-48.

[3] HASAN, Muhammad Rizky; SUHERMANTO, Suhermanto; SUHARMANTO, Suharmanto. Keamanan Sistem Perangkat Lunak dengan Secure Software Development Lifecycle. *Jurnal Ilmu Komputer dan Bisnis, 2021, 12.1: 88-101*.

[4] Hermawan, A., Hartati, T., & Wijaya, Y. A. (2022). *Analisa Keamanan Data Melalui Website Zahra Software Menggunakan Metode Keamanan Informasi CIA Triad*. Jurnal Informatika: Jurnal Pengembangan IT, *7*(3), 125-130.

[5] Himawan, C., Wibowo, T., Sulityo, B., Roestam, R., Wahyu, Y., & Wahyu, R. B. (2016). Studi perbandingan algoritma RSA dan algoritma El-Gamal. *Semin. Nas. APTIKOM*, *6*(1), 28-29.

[6] Istiyanto, J. E. (2009). Karakteristik Metodologi Penelitian Bidang Ilmu Komputer (Ik) Berlandaskan Pendekatai\Positivistik. *JURNAL SAINS DAN MATEMATIKA*, *17*(2), 115-120.

[7] OverviewVpn, *"Apa itu secure coding"*, https://vpnoverview.com/id/keamanan-internet/bisnis/apa-itu-secure-coding/, diambil pada tanggal 2 agustus 12.00 WIB

[8] Priyoatmoko2), K. I. (2016). Pengamanan Data Mysql Pada E-Commerce Dengan Algoritma Aes 256. *Seminar Nasional Sistem Informasi Indonesia, 1 Nopember 2016*, 120-126.

[9] Rizkita, N., Rosely, E., & Nugroho, H. (2018). Aplikasi Pendaftaran Dan Transaksi Pasien Klinik Hewan Di Bandung Berbasis Web (modul Pengelolaan Data Pasien Dan Transaksi). *EProceedings of Applied Science*, *4*(3).

[10] Suryawinata, M. (2019). Pengembangan Aplikasi Berbasis Web. *UMSIDA Press*, 1-144.

[11] Suendri. (2019). *Hashing Argon2 Untuk Keamanan Password Pada Sistem Berbasis Web Menggunakan Php*. *JISTech, Vol.4, No.1, Januari-Juni 2019, 4*, 1-12.

[12] University Binus, *"Sensitive Data Exposure"*, https://mti.binus.ac.id/2018/02/09/sensitive-data-exposure/, diambil pada tanggal 2 agustus 2022 pukul 15.30 WIB

[13] Veracode. (2018). *Secure Coding Best Practices Handbook.* Veracode, 1-16.

[14] ZAKARIA, A. (2007). *Peningkatan Keamanan Web Terhadap Serangan Cross Site Scripting (Xss)*, Sarjana Teknik, Universitas Brawijaya, 1-115.