PENAMBAHAN FITUR MULTI-FACTOR AUTHENTICATION DALAM STUDI KASUS SISTEM INFORMASI REKAM MEDIS RUMAH SAKIT

Mochammad Anwar Fauzi, Asep Id Hadiana, Fajri Rakhmat Umbara
Informatika, Universitas Jenderal Achmad Yani Cimahi
Jl. Terusan Jend. Sudirman, Cibeber, Kec. Cimahi Sel., Kota Cimahi, Jawa Barat 40531
Mochammad.anwar@student.unjani.ac.id

ABSTRAK

Data rekam medis merupakan informasi yang sangat penting dan sensitif bagi pasien dan harus dilindungi dari akses yang tidak sah. Dalam hal ini, sistem informasi Rumah Sakit memegang peran penting dalam menjaga keamanan data rekam medis pasien. Namun, ancaman akses yang tidak sah masih mengintai dan merusak privasi pasien. Saat ini sistem informasi rekam medis Rumah Sakit hanya mengandalkan autentikasi email dan kata sandi, sehingga rentan terhadap pencurian. Untuk mengatasi masalah ini, diperlukan rancangan keamanan Multi-Factor Authentication. Penerapan Multi-Factor Authentication pada penelitian ini memakai Personal Security Question dan Auth Token. Personal Security Question digunakan untuk memverifikasi pengguna dengan pertanyaan pribadi, sedangkan Auth Token menciptakan kode auntentikasi pada komunikasi server. Pada penelitian ini mekanisme sistem login dibagi menjadi tiga yaitu login dengan kata sandi, login dengan menjawab pertanyaan pribadi, dan login dengan menggunakan kode TOTP yang didapatkan dari email. Hasil dari penelitian ini adalah menjadikan sistem informasi rekam medis Rumah Sakit menjadi lebih aman dan mengurangi risiko diakses oleh pihak yang tidak berwenang. Hasil pengujian validasi TOTP dengan Google Authenticator yang dilakukan sebanyak seratus kali validasi menunjukan bahwa kode TOTP yang dihasilkan Google Authenticator tidak dapat mencocokan dengan hasil kode TOTP yang dihasilkan sistem.

Kata kunci: *Multi-Factor Authentication, Personal Security Question, Auth Token, Sistem Informasi Rekam Medis*

1. PENDAHULUAN

Sebagai hasil dari perkembangan teknologi informasi dan komunikasi, banyak pekerjaan yang digantikan dengan teknologi digital. Hal ini diterapkan oleh banyak perusahaan besar dan menengah untuk mempermudah operasional bisnis perusahaan [1]. Selain itu, dalam industri kesehatan, keamanan informasi rekam medis pasien sangat penting dan sensitif bagi pasien karena harus dilindungi dari akses yang tidak sah. Sebagaimana disebutkan dalam undang-undang nomor 29 tahun 2004 tentang praktik kedokteran telah mengatur secara sekilas mengenai perlindungan rekam medis milik pasien yaitu pada pasal 47 ayat (2), dan undang-undang nomor 36 tahun 2009 pasal 57 tentang kesehatan juga mengatur tentang kerahasiaan kondisi kesahatan [2]. Keamanan menjadi salah satu aspek penting yang harus diperhatikan pada jaringan komputer. Pasalnya, dalam jaringan tersebut terdapat banyak komputer yang saling terhubung, dan tidak sedikit pengguna yang mengaksesnya. Oleh karena itu, data dan informasi yang berada dalam jaringan tersebut rentan terhadap potensi serangan dari pihak yang tidak memiliki izin. Setiap kali seorang pengguna mengakses komputer dalam jaringan, langkah pertama yang harus dilakukan adalah memastikan bahwa identitas pengguna tersebut valid dan sah. Penggunaan kata sandi atau password adalah metode paling umum yang digunakan untuk melakukan proses autentikasi [3]. Proses autentikasi seringkali terkait dengan penggunaan kata sandi dan metode autentikasi berbasis kata sandi. Namun, kerentanan dalam proses autentikasi itu sendiri juga bisa menjadi sumber masalah yang serius [4]. Ada enam kerentanan utama terkait dengan autentikasi yang rusak yaitu, pertama kerentanan dalam logika autentikasi yang dimana kerentanan logika ini adalah sumber umum masalah dalam aplikasi perangkat lunak dan juga mempengaruhi proses autentikasi [4]. Kerentanan kedua yaitu proses pemulihan akun atau kata sandi yang lemah, seringkali kontrol keamanan hanya diterapkan selama proses autentikasi, oleh karena itu maka akan rentan di eksploitasi pada proses pemulihan akun dan kata sandi termasuk penyalahgunaan fitur lupa akun atau lupa kata sandi [4]. Kerentanan ketiga yaitu penggunaan pustaka autentikasi eksternal yang rentan sebagai contoh, kelemahan dalam autentikasi di WordPress Infinite WP Client dan plugin WP Time Capsule memungkinkan pihak yang tidak berwenang masuk ke akun administrator WordPress tanpa memerlukan kata sandi [4]. Kerentanan keempat yaitu penangan sesi yang tidak aman, kerentanan dalam pengelolaan session dapat memungkinkan pengguna jahat menggunakan session yang sudah terotentikasi tanpa perlu mengulang proses autentikasi [4]. Kerentanan kelima yaitu metode verifikasi kata sandi yang tidak aman, metode autentikasi berbasis kata sandi melibatkan perbandingan antara kata sandi yang disimpan di server dengan kata sandi dimasukkan oleh pengguna, penggunaan fungsi hash direkomendasikan untuk melindungi kata sandi, namun jika fungsi hash yang digunakan lemah, seperti MD5, maka ini dapat membantu penyerang dalam mengambil kata sandi dari nilai hash yang dicuri [4]. Kerentanan keenam

yaitu penegakan keamanan kata sandi yang buruk dimana memungkinkan pengguna menggunakan kata sandi yang lemah atau umum secara default menciptakan risiko dalam proses autentikasi [4]. Mengakses sistem berbasis web saat ini khususnya sistem informasi rekam medis rumah sakit hanya menggunakan satu faktor saja yaitu dengan mengetikan nama pengguna atau email pengguna dan kata sandi. Dengan sistem login yang hanya mempunyai satu faktor, sistem yang digunakan akan rentan terhadap akses yang tidak sah, karena kata sandi dapat dicuri dan digunakan kembali oleh pihak yang tidak berwenang [1]. Penguatan keamanan sistem dibutuhkan untuk mengatasi masalah autentikasi satu faktor. Salah satu cara untuk melakukan ini adalah dengan menggunakan Multi-Factor Authentication. Dengan menggunakan lebih dari satu proses autentikasi, informasi pengguna menjadi lebih aman [1].

Penelitian sebelumnya [1] membahas cara untuk mengamankan website e-commerce dengan menggunakan metode Multi-Factor Authentication. Dalam penelitian tersebut, metode ini digunakan untuk mengamankan modul login website dengan menggunakan autentikasi TOTP dan pertanyaan keamanan pribadi. Hasil dari penelitian ini menunjukkan bahwa dengan mengombinasikan enkripsi kata sandi menggunakan algoritma blowfish dan menerapkan TOTP dapat meningkatkan keamanan kata sandi dan memberikan akun pengguna yang lebih aman. Terdapat banyak cara yang dapat dilakukan untuk mengamankan data, antara lain OAuth [5], OAuth2 [6][7][8], JWT [9], dan Auth Token [10]. Masing-masing metode memiliki keunggulan dan cara kerja yang berbeda. Metode OAuth, OAuth2, dan JWT lebih cocok digunakan pada komunikasi client dengan server [5]. Metode Auth Token lebih cocok pada komunikasi server dengan server. Auth Token bekerja dengan cara membuat kode rahasia yang akan digunakan oleh client sebagai autentikasi. Keunggulan Auth Token adalah pada proses autentikasi yang lebih efisien terhadap penggunaan hardware. Autentikasi melalui faktor tambahan diperlukan untuk administrator sistem untuk menegakkan kebijakan keamanan data dan prosedur pada semua level basis data. Sehingga hanya pengguna yang sah yang dapat memiliki hak izin untuk mengakses ke sistem [11]. Multi-Factor Authetication merupakan autentikasi yang lebih aman dibandingkan dengan autentikasi satu faktor yaitu dengan nama pengguna atau email pengguna dan kata sandi. Metode Multi-Factor Authentication ini digunakan untuk mengecek identitas pengguna sebelum pengguna mengakses data sensitive dan rahasia. Multi-Factor Authentication akan menambahkan lapisan keamanan yang memungkinkan administrator sistem untuk menghubungkan dua atau lebih jenis autentikasi untuk memberikan cara yang lebih aman pada kredensial pengguna [11]. Terdapat tiga jenis faktor autentikasi yang paling umum digunakan yaitu: yang pertama adalah “something the user knows”, contohnya: nama pengguna, kata sandi, PIN atau keamanan pertanyaan. Yang kedua adalah “something the user have”, contohnya: smart card, perangkat smartphone. Yang ketiga adalah “something the user are”, contohnya: biometrik, sidik jari, pemindaiaian retina, pengenalan suara [11][12]. Untuk membuat faktor banyak sekali algoritma yang dapat digunakan salah satu algoritma yang dapat digunakan adalah algoritma Time Based One Time Password (TOTP) berbasis waktu yang di terapkan pada layer application. Dalam algoritma TOTP menyimpan sebuah timestamp yang digunakan untuk membuat token sekali pakai sehingga pengguna tidak perlu mengatur ulang token autentikasi. Generate token secara otomatis di lakukan oleh sistem sesuai dengan waktu penggunaan yang sudah ditentukan oleh sistem yaitu selama 60 detik dan apabila telah lewat dari 60 detik maka token lama tidak akan bisa di gunakan dan pengguna harus menggunakan token baru yang sudah diberikan oleh sistem kepada pengguna. Hal ini akan mempermudah pengguna awam karena mereka tidak perlu lagi mengatur token dan ketika kode token ini di ekstrak kita bisa mengetahui informasi kapan pengguna mengakses sistem dan waktu ini dapat dimanfaatkan dalam sistem informasi rekam medik untuk mengamankan data rekam medik yaitu dengan cara menambahkan autentikasi ganda pada sistem login dan forgot password. Selain itu TOTP juga bisa digunakan untuk autentikasi pengguna melalui token yang dapat menangkal replay attack dan phishing [13].

Berdasarkan permasalahan yang muncul, maka penelitian ini menambahkan fitur yaitu berupa metode Multi-Factor Authentication dengan dua faktor yang akan dipakai adalah something the user know dan something the user have berbasis Personal Security Question dan Auth Token untuk mengurangi kerentanan pada keamanan autentikasi pengguna dalam studi kasus sistem informasi rekam medis rumah sakit sehingga mengurangi risiko diakses oleh pihak yang tidak berwenang.

2. TINJAUAN PUSTAKA

2.1. Multi-Factor Authentication

Autentikasi merupakan proses elektronika yang memungkinkan proses identifikasi seseorang secara elektronik. Autentikasi ini dapat mengkonfirmasi keaslian serta integritas data pada bentuk elektronik. Proses autentikasi dilakukan pada saat pengguna melakukan permintaan masuk ke sistem dengan memakai email dan kata sandi. Pengguna yang terdaftar akan diizinkan untuk mengakses halaman atau fungsi yang diminta. Autentikasi dapat dilakukan dengan berbagai metode, salah satunya adalah Multi-Factor Authentication [1]. Multi-Factor Authentication adalah autentikasi yang lebih aman dibandingkan dengan autentikasi satu faktor yaitu dengan nama pengguna atau email pengguna dan kata sandi. Metode Multi-Factor Authentication ini digunakan untuk mengecek identitas pengguna

sebelum pengguna mengakses data sensitive dan rahasia. Multi-Factor Authentication akan menambahkan lapisan keamanan yang memungkinkan administrator sistem untuk menghubungkan dua atau lebih jenis autentikasi untuk memberikan cara yang lebih aman pada kredensial pengguna [11].

2.2. Personal Security Question

Personal Security Question adalah metode yang digunakan untuk memverifikasi pengguna dengan cara menjawab pertanyaan tertentu yang hanya diketahui oleh pengguna secara pribadi. Saat pengguna membuat akun, mereka diminta untuk menjawab pertanyaan yang bersifat pribadi. Idealnya, pertanyaan tersebut hanya dapat dijawab oleh pengguna itu sendiri. Jawaban dari pertanyaan tersebut akan dicatat, dan jika pengguna perlu mengonfirmasi bahwa mereka adalah pemilik akun yang sah, sistem akan menanyakan pertanyaan tersebut. Jika jawaban pengguna sesuai dengan yang mereka berikan saat membuat akun, pengguna diizinkan masuk ke akun mereka. Meskipun terlihat sederhana, memilih dan mengisi pertanyaan keamanan pribadi sama pentingnya dengan membuat kata sandi [1].

2.3. Auth Token

Auth Token adalah proses autentikasi berbasis token. Auth Token bekerja dengan cara membuat kode rahasia yang akan digunakan oleh client sebagai autentikasi. Keunggulan Auth Token adalah pada proses autentikasi yang lebih efisien terhadap penggunaan hardware dan cocok untuk integrasi data [14].

2.4. Time Based One Time Password

Algoritma TOTP merupakan algoritma yang dikembangkan dari Hashes Message Authentication Code (HMAC) yang menggabungkan antara secret key dengan waktu saat ini algoritma ini memiliki waktu penggunaan yang pendek sesuai dengan waktu interval penggantian token yang sudah ditentukan [13], berikut adalah bentuk umum dari TOTP yang dapat dilihat pada persamaan 1 di bawah :

$$TOTP = HOTP(K,T)T = \frac{Tcurrent-T0}{X} \quad (1)$$

Dimana K adalah Kunci rahasia bersama dan T adalah sebuah nilai integer yang merepresentasikan waktu. Tcurrent adalah waktu saat ini dalam detik sedangkan T0 disepakati pada saat inisialiasi awal, biasanya nilainya diset nol (T0=0). Parameter X menentukan berapa lama sebuah TOTP valid. Standar yang digunakan adalah 60. Jadi dengan mengisi X=60, kita mengizinkan sebuah token hanya valid selama 60 detik [13] dan TOTP dapat mengamankan privasi data melalui proses otentikasi. Faktor utama dari algoritma TOTP adalah waktu karena di setiap waktu yang berbeda TOTP bisa saja membuat kode token yang berbeda – beda di setiap perubahan waktu. Kode token TOTP yang sudah dibuat tidak akan langsung di kirim ke pengguna, kode ini harus di proses terlebih dahulu agar lebih aman. Untuk pengaman kode ini penulis menggunakan kriptografi hash.

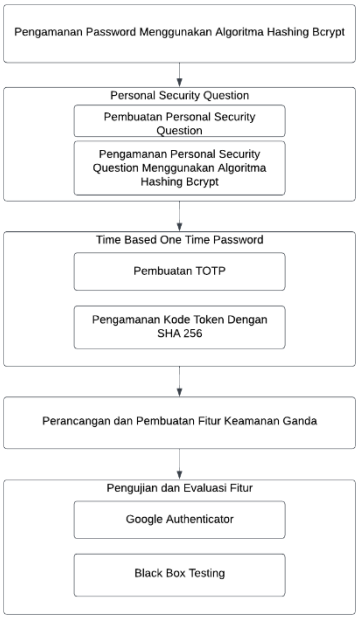
2.5. Algoritma Hashing Bcrypt

Bcrypt adalah sebuah fungsi hash yang dirancang oleh Niels Provos dan David Mazières berdasarkan cipher Blowfish. Nama "Bcrypt" terdiri dari "B" yang merujuk pada Blowfish dan "Crypt" yang merupakan nama fungsi hash yang digunakan dalam sistem kata sandi di UNIX. Pada awalnya, fungsi hash Crypt, yang dikembangkan pada tahun 1976, hanya dapat menghasilkan hash untuk empat kata sandi setiap detiknya. Pada saat itu, Crypt cukup kuat. Namun, dengan kemajuan teknologi yang pesat, komputer saat ini dengan perangkat lunak dan perangkat keras yang dioptimalkan dapat menghasilkan hash menggunakan Crypt hingga 200.000 kata sandi per detik. Oleh karena itu, Crypt tidak lagi cocok untuk menjaga keamanan kata sandi.

Proses Bcrypt memiliki tahap inisialisasi kunci yang dikembangkan dari tahap inisialisasi kunci dalam cipher Blowfish, yang disebut "eksblowfish" (ekspansi kunci mahal Blowfish). Selain itu, Bcrypt secara default menggunakan salt berukuran 128-bit dalam proses hash-nya untuk mencegah serangan rainbow table. Proses Bcrypt terdiri dari dua tahap. Tahap pertama adalah menjalankan inisialisasi kunci eksblowfish dengan parameter cost yang diinginkan, nilai salt, dan kata sandi yang akan di-hash. Pada tahap ini, Bcrypt melakukan penurunan kunci di mana serangkaian subkunci diturunkan dari kunci utama, dengan kunci utama diisi oleh kata sandi. Jika kata sandi yang digunakan terlalu pendek, tahap ini akan memperpanjang kata sandi menjadi kunci yang lebih panjang. Dengan demikian, tahap pertama memperkuat kunci. Tahap kedua melibatkan enkripsi pada sebuah nilai ajaib berukuran 192-bit, "OrpheanBeholderScryDoubt", menggunakan kunci yang dihasilkan pada tahap pertama. Tahap kedua ini diiterasi sebanyak 64 kali. Hasil akhir dari tahap ini adalah hasil enkripsi menggunakan mode ECB yang digabungkan dengan cost dan salt berukuran 128-bit [15].

3. METODE PENELITIAN

Metode penelitian adalah suatu pendekatan yang diterapkan untuk melaksanakan proses identifikasi dengan maksud merespons identifikasi masalah dan tujuan dari penelitian tersebut. Dalam penelitian ini, terdapat serangkaian langkah yang harus dijalani guna menyelesaikan proses pengindentifikasian yang sedang berjalan. Terdapat beberapa tahapan pokok dalam penelitian ini. Pertama, mengamankan password menggunakan algoritma hashing bcrypt. Kemudian dilakukan tahap pembuatan personal security question dan pembuatan TOTP. Lebih jelasnya diberikan pada Gambar 1.



Gambar 1. Metode Penelitian

3.1. Pengamanan Password Menggunakan Algoritma Hashing Bcrypt

Tahap ini dilakukan pengamanan password pada akun user menggunakan algoritma hashing bcrypt. Bcrypt adalah algoritma hashing yang dirancang khusus untuk menghasilkan hash yang sulit untuk dipecahkan secara brute-force dimana algoritma ini menggunakan salting (penambahan nilai acak) dan iterasi yang dapat dikonfigurasi untuk meningkatkan keamanan password.

3.2. Pembuatan Personal Security Question

Tahap ini dilakukan pembuatan personal security question atau pertanyaan keamanan pribadi dengan cara membuat form pertanyaan keamanan pribadi yang hanya diketahui oleh user. Pada tahap ini akan menggunakan bahasa pemrograman PHP dan HTML serta CSS untuk membuat form pertanyaan keamanan pribadi. Pada tahap ini juga akan dibedakan personal security question setiap user dengan cara mencari personal security question berdasarkan email serta username. Pada tahap ini juga personal security question akan dienkripsi menggunakan algoritma hashing bcrypt.

3.3. Pembuatan TOTP

Tahap ini mengimplementasikan TOTP (Time-Based One Time Password) ini mencakup proses pembuatan secret key, pemrosesan token TOTP dan pengamanan token. Pada proses pemrosesan data pada tahap ini akan menggunakan algoritma TOTP dengan memasukan data waktu dan secret key sehingga dapat menghasilkan sebuah token yang dapat berubah-rubah setiap waktu. Data waktu dan secret key adalah data yang diolah oleh algoritma TOTP untuk menghasilkan sebuah token, yang dimana token ini akan digunakan sebagai faktor ketiga untuk melakukan autentikasi pada sistem. Sebelum dikirimkannya token autentikasi akan dilakukan pengamanan kode token dengan SHA 256 agar kode yang dikirimkan dapat terjaga kerahasiaan nya dan tidak bisa dibaca oleh orang yang tidak memiliki hak akses untuk membaca. Cara kerja dari SHA 256 adalah mengubah pesan masukan ke dalam message diggest 256 bit berdasarkan secure hash signature standard.

3.4. Perancangan dan Pembuatan Fitur Keamanan Ganda

Tahap ini melakukan perancangan dan pembuatan fitur keamanan ganda dengan menerapkan Multi-Factor Authentication yang menggunakan Personal Security Question dan Authentication Token berbasis TOTP.

3.5. Pengujian dan Evaluasi Fitur

Tahap ini dilakukan pengujian dalam bentuk pengujian Black Box dan pengujian keamanan otentikasi menggunakan Google Authenticator. Pada peneitian ini dilaksanakan pengujian keamanan otentikasi menggunakan Google Authenticator dengan cara mencocokkan kode TOTP yang dihasilkan oleh sistem dengan Google Authenticator dan mencoba memasukkan kedua kode TOTP tersebut pada saat login. Agar Google Authenticator terhubung dengan sistem maka perlu mengaktifkan otentikasi dua factor pada akun yang ingin diuji, dan menghubungkan akun tersebut dengan Google Authenticator. Selanjutnya dapat melakukan pengujian dengan mencoba memasukkan kode TOTP yang dihasilkan oleh Google Authenticator saat masuk ke akun tersebut. Kemudian pada tahap ini juga melakukan evaluasi terhadap fitur keamanan ganda yang dirancang pada sistem informasi rekam medis Rumah Sakit untuk mengetahui sejauh mana metode yang diterapkan dapat melindungi sistem informasi rekam medis Rumah Sakit.

4. HASIL DAN PEMBAHASAN

4.1. Pengamanan Password Menggunakan Algoritma Hashing Bcrypt

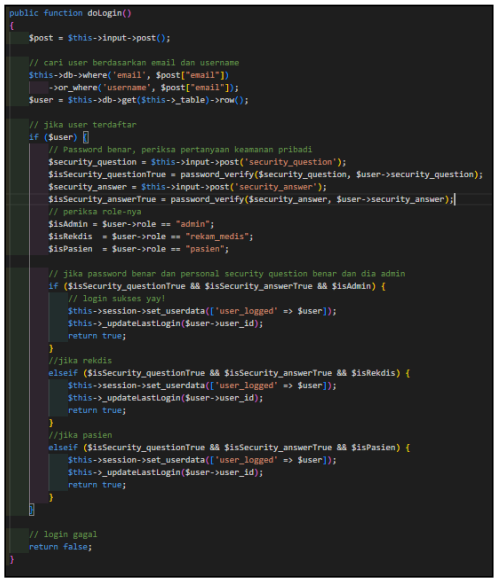
Pengamanan password menggunakan algoritma hashing bcrypt dilakukan dengan mengenkripsi password dengan sintak password_hash dan password_default untuk menentukan algoritma hashing yang dipakai, saat ini nilai password_default secara default akan menggunakan algoritma bcrypt. Algoritma bcrypt ini memiliki keamanan yang tinggi terhadap serangan brute-force. Password akan dienkripsi menggunakan algoritma hashing bcrypt seperti yang ditampilkan pada Gambar 2.

```
$this->password = password_hash($post["password"], PASSWORD_DEFAULT);
$this->security_question = password_hash($post["security_question"], PASSWORD_DEFAULT);
$this->security_answer = password_hash($post["security_answer"], PASSWORD_DEFAULT);
```

Gambar 2. Pengamanan Password Menggunakan Algoritma Hashing Bcrypt

4.2. Pembuatan Personal Security Question

Fitur personal security question merupakan layanan faktor keamanan tambahan yang berfungsi untuk melapisi keamanan fungsi login dan forgot password agar tidak sembarang user dapat mengakses fungsi tersebut. Pada personal security question ini akan dilakukan pengecekan terhadap jawaban dari pertanyaan keamanan pribadi user. Pada personal security question ini juga jawaban dan pilihan pertanyaan keamanan pribadi user akan dienkripsikan menggunakan algoritma hashing bcrypt. Implementasi fitur personal security question dapat dilihat pada Gambar 3.



```
public function doLogin()
{
    $post = $this->input->post();

    // cari user berdasarkan email dan username
    $this->db->where('email', $post["email"])
        ->or_where('username', $post["email"]);
    $user = $this->db->get($this->_table)->row();

    // jika user terdaftar
    if ($user) {
        // Password benar, periksa pertanyaan keamanan pribadi
        $security_question = $this->input->post('security_question');
        $isSecurity_questionTrue = password_verify($security_question, $user->security_question);
        $security_answer = $this->input->post('security_answer');
        $isSecurity_answerTrue = password_verify($security_answer, $user->security_answer);
        // periksa role-nya
        $isAdmin = $user->role == "admin";
        $isRekdis  = $user->role == "rekam_medis";
        $isPasien  = $user->role == "pasien";

        // jika password benar dan personal security question benar dan dia admin
        if ($isSecurity_questionTrue && $isSecurity_answerTrue && $isAdmin) {
            // login sukses yay!
            $this->session->set_userdata(['user_logged' => $user]);
            $this->_updateLastLogin($user->user_id);
            return true;
        }
        //jika rekdis
        elseif ($isSecurity_questionTrue && $isSecurity_answerTrue && $isRekdis) {
            $this->session->set_userdata(['user_logged' => $user]);
            $this->_updateLastLogin($user->user_id);
            return true;
        }
        //jika pasien
        elseif ($isSecurity_questionTrue && $isSecurity_answerTrue && $isPasien) {
            $this->session->set_userdata(['user_logged' => $user]);
            $this->_updateLastLogin($user->user_id);
            return true;
        }
    }

    // login gagal
    return false;
}
```

Gambar 3. Personal Security Question

4.3. Pembuatan TOTP

```
class Otp_model extends CI_Model
{

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function createOtp($email){
        $otp = $this->generateRandomString(6);
        $otpTimeStamp = date('Y-m-d H:i:s');

        // save OTP menggunakan bcrypt
        $hashedOtp = hash('sha256', $otp);

        $data = array(
            'otp_code' => $hashedOtp,
            'otp_timestamp' => $otpTimeStamp
        );

        $this->db->where('email', $email);
        $this->db->update('users', $data);
        return $otp;
    }

    public function getOtpCode($user_id){
        $this->db->where('user_id', $user_id);

        return $this->db->get('users')->result_array()[0]['otp_code'];
    }

    public function isOtpValid($email, $input_otp){
        $this->db->where('email', $email);
        $user = $this->db->get('users')->result_array()[0];

        $otp_code = $user['otp_code'];
        $otpDateTime = $user['otp_timestamp'];
        $currentDateTime = date('Y-m-d H:i:s');

        // save input OTP menggunakan bcrypt
        $hashedInputOtp = hash('sha256', $input_otp);

        if (strcmp($otp_code, $hashedInputOtp) == 0) {
            $currentTimestamp = strtotime($currentDateTime);
            $otpTimestamp = strtotime($otpDateTime);

            $timeDifference = $currentTimestamp - $otpTimestamp;

            if ($timeDifference > 60) {
                return -2; //input waktu >60 detik
            } else {
                return 1; //otp valid
            }
        } else {
            return -1; //otp salah
        }
    }

    function generateRandomString($length)
    {
        $characters = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';
        $randomString = '';

        for ($i = 0; $i < $length; $i++) {
            $randomString .= $characters[rand(0, strlen($characters) - 1)];
        }
```

Gambar 4. Time Based One Time Password

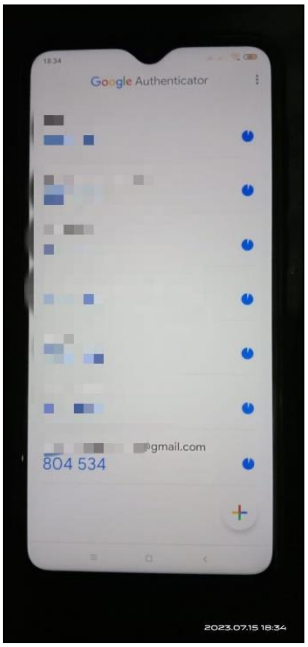
TOTP merupakan metode yang digunakan untuk menghasilkan password sekali pakai (one-time password) berdasarkan waktu yang berlaku dalam jangka waktu tertentu. TOTP yang diimplementasikan akan berlaku selama 60 detik. Sebelum dikirimkannya token autentikasi akan dilakukan pengamanan kode token dengan SHA 256 agar kode yang dikirimkan dapat terjaga kerahasiaan nya dan tidak bisa dibaca oleh orang yang tidak memiliki hak akses untuk membaca. Implementasi fitur token autentikasi dapat dilihat pada Gambar 4.

4.4. Pengujian dan Evaluasi Fitur

Pengujian perlu dilakukan untuk memastikan web service yang dibangun bekerja dengan baik dan sesuai kebutuhan yang telah didefinisikan dan untuk mencari kesalahan yang mungkin terjadi pada web service ketika web service dioperasikan. Pengujian dilakukan terbatas pada pengujian keamanan multi-factor authentication pada bagian login dan forgot password. Pengujian yang dilakukan yaitu validasi totp google authenticator dan black box testing.

4.5. Pengujian Validasi TOTP Dengan Google Authenticator

Gambar 5. TOTP Yang Dihasilkan Oleh Sistem



Gambar 6. TOTP Yang Dihasilkan Oleh Google Authenticator

Google Authenticator merupakan salah satu aplikasi otentikasi dua faktor yang umum digunakan dan diakui secara luas. Google Authenticator mendukung algoritma TOTP (Time-bases One-Time Password) yang merupakan standar umum untuk menghasilkan kode satu kali pakai berdasarkan waktu. Pada pengujian validasi TOTP dengan Google Authenticator pada penelitian ini terlebih dahulu

melakukan pengaturan character yang akan dihasilkan pada TOTP dengan angka dari 0 sampai 9. Pengujian validasi totp dengan Google Authenticator dilakukan untuk menyamakan totp yang dikirim ke email dengan TOTP yang ada pada Google Authenticator

Gambar 7. Hasil Pengujian Validasi TOTP Google Authenticator

Untuk melakukan pengujian validasi TOTP, pada penelitian ini akan melakukan mengirim kode totp secara bersamaan dengan refresh totp dari Google Authenticator. Setelah itu akan melihat kecocokan TOTP yang pada email dan Google Authenticator. Seperti yang terlihat pada Gambar 5 bahwa totp yang dikirimkan pada email adalah 021589, sedangkan totp yang dihasilkan oleh Google Authenticator adalah 804534 seperti yang terlihat pada Gambar 6. Kemudian melakukan percobaan dengan memasukkan totp yang dihasilkan oleh Google Authenticator pada halaman token autentikasi untuk melihat apakah totp yang dihasilkan dapat terbaca valid atau tidak seperti yang terlihat pada Gambar 7. Pengujian ini menghasilkan output kode otp salah atau tidak valid yang artinya algoritma TOTP yang dibuat aman.

4.6. Pengujian Black Box

Pengujian Black Box perlu dilakukan guna memverifikasi bahwa layanan web yang telah dikembangkan beroperasi secara efektif sesuai dengan persyaratan yang telah ditetapkan, serta untuk mengidentifikasi potensi kesalahan yang mungkin timbul saat layanan web tersebut sedang berjalan. Berikut Tabel 1 menunjukan hasil dari pengujian Black Box.

Tabel 1. Hasil Pengujian Black Box

Kode Uji	Nama Pengujian	Kriteria Evaluasi	Hasil yang diharapkan	Hasil Pengujian
KU01.1	Generate Token Autentikasi	Mengisikan form dengan token yang valid	Jika pengisian form dengan token yang valid, maka sistem akan memperbolehkan user untuk masuk ke dalam sistem.	Sesuai
KU01.2	Generate Token Autentikasi	Mengisikan form dengan token yang tidak valid	Jika pengisian form dengan token yang tidak valid, maka sistem akan menampilkan pesan bahwa kode otp salah atau kode otp expired.	Sesuai
KU02.1	Cek Personal Security Question	Mengisikan form dengan personal security question yang valid	Jika pengisian form dengan personal security question yang valid, maka sistem akan memindahkan user ke halaman token autentikasi.	Sesuai
KU02.2	Cek Personal Security Question	Mengisikan form dengan personal security question yang tidak valid	Jika pengisian form dengan personal security question yang tidak valid, maka sistem akan menampilkan pesan bahwa email atau personal security question salah.	Sesuai
KU03.1	Melakukan Login	Mengisikan form login dengan lengkap	Jika pengisian form login lengkap, maka sistem akan mengarahkan user ke halaman menjawab personal security question.	Sesuai
KU03.2	Melakukan Login	Mengisikan form login dengan tidak lengkap	Jika pengisian form login tidak lengkap, maka sistem akan menampilkan pesan bahwa email atau password salah.	Sesuai
KU04.1	Merubah Password	Mengisikan form forgot password dengan lengkap	Jika pengisian form forgot password lengkap, maka sistem akan melakukan update terhadap password akun user tersebut dengan password yang baru.	Sesuai
KU04.2	Merubah Password	Mengisikan form forgot password dengan tidak lengkap	Jika pengisian form forgot password tidak lengkap, maka sistem akan menampilkan pesan bahwa please fill out this field.	Sesuai

5. KESIMPULAN DAN SARAN

Kesimpulan penelitian ini menunjukkan bahwa penerapan penambahan fitur multi-factor authentication dalam studi kasus sistem informasi rekam medis rumah sakit efektif dalam menjaga keamanan data sensitif pasien. Penggunaan multi-factor authentication, yang terdiri dari kombinasi personal security question dan auth token, memberikan tingkat keamanan yang lebih tinggi daripada metode otentikasi tunggal. Dengan memerlukan jawaban personal security question yang hanya diketahui oleh pengguna yang sah, serta penggunaan auth token yang terus berubah, fitur ini mampu mencegah akses yang tidak sah ke sistem

informasi rekam medis. Hasil penelitian juga menunjukkan bahwa implementasi pengamanan ini dapat mengurangi risiko serangan peretasan dan pencurian identitas. Data sensitif pasien, seperti informasi medis dan riwayat pengobatan, dijaga dengan baik melalui lapisan keamanan tambahan yang diberikan oleh multi-factor authentication. Selain itu, penggunaan personal security question memungkinkan pengguna yang lupa kata sandi untuk melakukan reset dengan aman. Dengan menjawab pertanyaan keamanan yang hanya diketahui oleh pengguna yang sah, pengguna dapat memulihkan akses ke akun mereka tanpa memberikan celah bagi pihak yang tidak berwenang.

DAFTAR PUSTAKA

[1] M. A. Nugraha, D. Arisandi, and N. J. Perdana, "Pengamanan Website E-Commerce Menggunakan Multi-Factor Authentication," *J. Ilmu Komput. dan Sist. Inf.*, vol. 9, no. 1, p. 158, 2021, doi: 10.24912/jiksi.v9i1.11588.

[2] S. D. Rosadi, "IMPLIKASI PENERAPAN PROGRAM E-HEALTH DIHUBUNGKAN DENGAN PERLINDUNGAN DATA PRIBADI," no. 35, pp. 403–420, 2017.

[3] Azhar, Arkarni Wais, and Atthariq, "Sistem Keamanan Pada Halaman Login Menggunakan One Time Password," *J. Embed. Syst. Secur. Intell. Syst.*, vol. 01, no. 2, pp. 106–113, 2020.

[4] SAKSHYAM SHAH, "Most Common Authentication Vulnerabilities," 2022. https://goteleport.com/blog/authentication-vulnerabilities/.

[5] K. Saputra and K. Farhan, "Implementasi Protokol OAuth 1 . 0 Sebagai Autentikasi pada Aplikasi SMS Blast Berbasis Android," *J. Electr. Technol.*, vol. 2, no. 3, pp. 27–30, 2017.

[6] Y. Fatman, "Implementasi Metode Open Authorization (OAUTH2) Untuk Pengelolaan Data Dosen di Universitas Islam Nusantara," *Ainet J. Inform.*, vol. 2, no. 1, pp. 10–18, 2020, doi: 10.26618/ainet.v2i1.3212.

[7] I. Kusuma, A. Susanto, and I. U. W. Mulyono, "Implementasi Restful Web Services Dengan Otorisasi Oauth 2.0 Pada Sistem Pembayaran Parkir," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 1, pp. 391–404, 2019, doi: 10.24176/simet.v10i1.3026.

[8] R. Kurniawan, "Perancangan dan Implementasi Sistem Otentikasi OAuth 2.0 dan PKCE Berbasis Extreme Programming (XP)," *J. Pendidik. dan Teknol. Indones.*, vol. 2, no. 2, pp. 601–611, 2022, doi: 10.52436/1.jpti.141.

[9] A. Rahmatulloh, H. Sulastri, and R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 2, 2018, doi: 10.22146/jnteti.v7i2.417.

[10] I. G. Anugrah and M. A. R. I. Fakhruddin, "Development Authentication and Authorization Systems of Multi Information Systems Based REst API and Auth Token," *Innov. Res. J.*, vol. 1, no. 2, p. 127, 2020, doi: 10.30587/innovation.v1i2.1927.

[11] K. Phan, "Implementing Resiliency of Adaptive Multi-Factor Authentication Systems," vol. 65, 2018, [Online]. Available: https://repository.stcloudstate.edu/msia_etdshttps://repository.stcloudstate.edu/msia_etds/65.

[12] M.Aldwairi and S.Aldhanhani, "Multi-Factor Authentication System Multi-Factor Authentication System," *Am. Sci. Publ.*, vol. X, no. August, 2017.

[13] M. S. Ramadhan and F. P. Ariyani, "Peningkatan Keamanan Login Website Dengan Implementasi One Time Password Menggunakan Algoritma Sha1 Dan Md5 Berbasis Mobile," *Skanika*, vol. 1, no. 2, pp. 689–696, 2018.

[14] M. I. Perkasa and E. B. Setiawan, "Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token," *J. Ultim. Comput.*, vol. 10, no. 1, pp. 19–26, 2018, doi: 10.31937/sk.v10i1.838.

[15] M. D. Akbar and A. Antoni, "Aplikasi Absensi Pegawai pada Dinas Komunikasi dan Informatika Kabupaten Deli Serdang dengan QR Code Menggunakan Algoritma Bcrypt," *sudo J. Tek. Inform.*, vol. 1, no. 1, pp. 8–16, 2022, doi: 10.56211/sudo.v1i1.2.