PENGIMPLEMENTASIAN ONE TIME PASSWORD DAN NOTIFIKASI EMAIL MENGGUNAKAN FUNGSI HASH SHA-512 BERBASIS WEB PADA SMK CYBER MEDIA

Nanda Imani Yahya[1)], **Safrina Amini**[2)]
[1]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : nandaimaniyahya@yahoo.co.id[1)], safrina.amini@budiluhur.ac.id[2)]

ABSTRAK

Sistem informasi berbasis web sering digunakan oleh banyak institusi terutama dalam lembaga pendidikan sebagai media utama baik dalam proses pendataan siswa dan penilaian. Dengan adanya sistem tersebut, pengaksesan data menjadi lebih mudah namun masalah keamanan yang melindungi hak akses dari pihak yang berwenang pun menjadi masalah baru. Dengan banyaknya website phishing, maupun penyadapan membuat informasi login yang umumnya berupa username dan password gagal dalam tugasnya untuk mengamankan hak akses penggunanya. Maka dibutuhkan lapisan keamanan lainnya berupa One Time Password yang merupakan sebuah kode sementara yang hanya dapat diakses oleh pengguna yang bersangkutan. Dengan banyaknya penggunaan handphone oleh berbagai kalangan maka penelitian ini menggunakan handphone sebagai media penerima kode tersebut. Aplikasi yang akan dikembangkan ini berbasis web dan menggunakan fungsi hash SHA-512 sebagai bagian dalam pembuatan kode verifikasi dimana bagian dari hasil hash tersebut akan dikonversikan menjadi kode yang nanti akan dikirimkan lewat SMS ke handphone pengguna yang terdaftar. Bahasa pemrograman yang digunakan adalah PHP dan MySQL sebagai database. Kesimpulan yang dibuat dari penyusunan penelitian ini adalah aplikasi ini adalah hak akses pengguna terjaga karena bukan hanya dibutuhkan username dan password tapi juga kode verifikasi yang dikirimkan lewat SMS ke handphone pengguna yang hanya berlaku selama beberapa saat.

Kata kunci : login, OTP, One Time Password, SHA-512, SMS

1. PENDAHULUAN

Penggunaan sistem informasi berbasis web pada institusi pendidikan di Indonesia semakin berkembang setiap harinya. Penyimpanan informasi akademis maupun pribadi siswa menjadi pusat perhatian bagi pihak administratif pada lembaga pendidikan. Pada SMK Cyber Media, penyimpanan informasi siswa menjadi hal yang sangat sensitif sehingga diperlukan sebuah sistem yang dilengkapi dengan metode otentikasi yang mumpuni, yang terbebas dari berbagai macam metode serangan, karena sistem yang tidak terproteksi oleh keamanan dapat menerima banyak ancaman dari pihak luar yang berkehendak untuk mendapatkan informasi tersebut demi keuntungan sendiri maupun suatu organisasi.

Ada beberapa metode untuk melakukan otentikasi. Salah satunya adalah menggunakan *password* dan *username*. Namun pengunaan kedua hal tersebut tidak dapat memastikan bahwa hak akses pengguna tidak dapat dijebol oleh pengguna lain. Contoh yang paling mudah adalah pihak yang mendapat hak akses ke database akan dengan mudah mengambil data *username* dan *password* yang ada pada *database* tersebut. Kerahasiaan merupakan faktor penting untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi tersebut. [1]

Banyak cara yang dapat digunakan oleh *hacker* agar dapat mendapatkan informasi yang bersifat rahasia dari pengguna, baik dari informasi otentikasi bahkan sampai informasi yang bersifat vital. Dalam rangka mendapatkan informasi tersebut, seorang *hacker* dapat menggunakan beberapa metode, salah satunya adalah *sniffing*. Dalam *sniffing*, *hacker* dapat melihat semua informasi yang di*submit* ke form oleh korbannya yang ada dalam jaringan yang sama tanpa sepengetahuan korbannya tersebut dengan bantuan *software* tertentu [2]. Dan metode lainnya seperti *Keylogging*. Contohnya jika kita menggunakan komputer umum yang biasanya ada di perpustakaan ataupun warnet. Komputer tersebut bisa saja dipasangi penyadap untuk membaca *keystroke* yang di masukkan oleh pengguna. Ketika melakukan *login*, pada umumnya *user* harus memasukkan 2 informasi yang bersifat statis yaitu *username* dan *password*. Kedua informasi tersebut pun terekam dan sudah dianggap tidak aman karena sudah diketahui oleh penyadap.

Dengan adanya ancaman keamanan tersebut, proses autentikasi menggunakan informasi *login* yang bersifat statis seperti *password* dan *username* yang

tidak berubah (atau jarang dirubah), terutama penggunaan *password* yang menggunakan tanggal lahir *user* sangat tidak disarankan dan dibutuhkan adanya lapisan keamanan tambahan untuk memastikan bahwa *user* yang sedang menggunakan akun tersebut adalah *user* yang memang seharusnya mengakses. Salah satu metode yang digunakan untuk menambah lapisan keamanan tersebut adalah dengan menambahkan metode *One Time Password* (OTP), yaitu sebuah metode otentikasi yang menggunakan satu kunci *password* yang bersifat sementara dan akan hangus setelah beberapa saat.

Sistem *One Time Password* ini menggunakan algoritma *Secure Hash Algorithm 512* yang berada dalam set hash SHA-2, dikarenakan SHA-2 memperbaiki kelemahan matematis yang mungkin ada pada SHA-1 [3]

2. LANDASAN TEORI

2.1 Login

Login adalah proses dimana pengguna dapat memasukkan *username* dan *password* untuk dapat mengakses sebuah sistem [4]. Ketika *user* ingin masuk kedalam sistem, maka *user* akan dihadapkan pada tampilan layar yang mengharuskan *user* untuk memasukkan *username* dan *password*, kedua hal ini saling berkaitan dan apabila kedua hal tersebut tidak cocok, maka tindakan *user* akan ditolak dan tidak akan dapat masuk kedalam sistem. *Password* dalam sistem ini dapat diubah namun *username* tidak bisa diubah karena *username* bertindak sebagai *primary key* dimana setiap *user* memiliki *username* yang berbeda

2.2 Password

Password merupakan urutan kode rahasia yang digunakan untuk mengakses atau membuka suatu sistem yang terkunci. *Password* biasanya bersifat sangat rahasia dimana hanya pemilik dari suatu entitas yang memiliki *password* tersebut atau pihak lain yang dipercaya oleh pemilik yang dapat mengetahuinya. *Password* juga tidak berubah-ubah dan faktor ini yang membuat *password* menjadi beresiko terhadap serangan serangan dari luar seperti *Dictionary attack, brute force, phishing*, dan biasanya pemilik *password* pun hampir tidak akan merubah passwordnya kecuali pemiliknya sudah merasa bahwa *password* yang ia miliki sudah tidak aman.

2.3 Otentikasi

Otentikasi adalah sebuah proses untuk memastikan bahwa koneksi dari kedua belah pihak (*client* maupun *server*) adalah *valid.* Yang dimaksud *valid* disini adalah kredensial *user* yang diinput maupun informasi yang sudah disimpan di *server* keduanya sama. Contohnya pada kasus yang sangat umum dalam keadaan login dimana *user* harus memasukkan *username/userid* dan *password*, apabila *username* atau *password* atau keduanya tidak sama dengan data yang sudah tersimpan di *server* selanjutnya, maka otentikasi dikatakan gagal karena informasi dari kedua belah pihak tidak saling cocok. Dikarenakan hal ini, *server* harus melindungi informasi dari user baik dengan cara mengenkripsi *password* user karena *password* tersebut bersifat rahasia ataupun menambahkan lapisan keamanan lainnya untuk menjaga *priviledge* dari *user* yang akan mengakses.

2.4 One Time Password

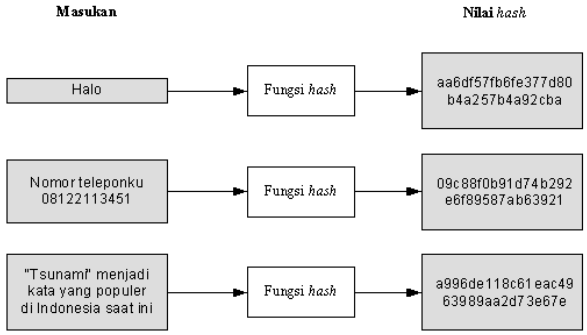
One Time Password merupakan sebuah proses otentikasi dari *server* yang menggunakan kode yang dinamis setiap kali pengguna melakukan tindakan yang di*challenge* oleh *server*, atau berubah setiap interval waktu tertentu [5]. Metode otentikasi ini tidak menggunakan kredensial yang berada dari suatu akun tertentu seperti *password* maupun *username* dan tidak pernah konstan, berbeda dengan kredensial *login* yang berada pada akun *user* yang tidak akan berubah kecuali *user* itu sendiri yang memintanya agar diubah.

One Time Password memiliki beberapa batasan, yaitu:

a. Mempunyai masa aktif yang tidak lama
   Password OTP memiliki masa aktif yang tidak lama dikarenakan faktor keamanan yang mempengaruhi hal tersebut. Apabila *password* OTP memiliki masa aktif yang lama atau bahkan tidak memiliki masa aktif sama sekali, maka *password* OTP tersebut rentan terhadap serangan seperti *bruteforce* yang sering dihadapi oleh *password* statis yang tidak memiliki jangka waktu [6]
b. Bersifat dinamis
   Password OTP bersifat dinamis karena setiap *password* yang dibangkitkan tidak akan sama karena *password* yang dibangkitkan akan selalu berubah ubah. Dan *password* OTP ini juga tidak bersifat terikat kepada informasib *login user.*
c. Hanya fungsional pada konteks tertentu
   Untuk melakukan sebuah proses autentikasi diperlukan kode yang bersifat terikat dengan proses dari *server*. Sehingga password tidak bisa dipakai untuk proses berbeda yang membutuhkan *challenge code* yang berbeda. Konteks ini hanya berlaku bila password dihasilkan dari mode CR.

2.5 Fungsi Hash

Fungsi *hash* adalah sebuah fungsi yang menerima *input* berupa string lalu diproses sesuai dari standar fungsi *hash* tersebut yang kemudian akan mengeluarkan *output* berupa string dengan panjang yang tetap dan panjang tersebut tidak tergantung pada ukuran awal dari string *input*. .



Gambar 1:Input dan Output fungsi hash.

Fungsi *hash* biasanya digunakan sebagai metode otentikasi sertifikat web dan metode pencocokan antara *client* dan *server*, *Server* biasanya akan membandingkan hasil *output* dari fungsi *hash* pengguna dengan *output* dari fungsi hash *server*, hal ini dilakukan agar informasi yang ditukarkan bukan informasi yang sebenarnya dan menghindari dari *intercept* pihak luar.

2.6 SHA-512

Algoritma SHA dirancang oleh National Security Agency (NSA) dan menjadi standar pemrosesan informasi pada tahun 1993. SHA dibuat berdasarkan fungsi hash MD4 dan didesain mirip dengan MD4. Pada tahun 2005, NIST (National Institute of Standars and Technology) mengadakan pengumuman agar terjadinya konversi dari penggunaan SHA-1 menjadi SHA-2 pada tahun 2010 [7]

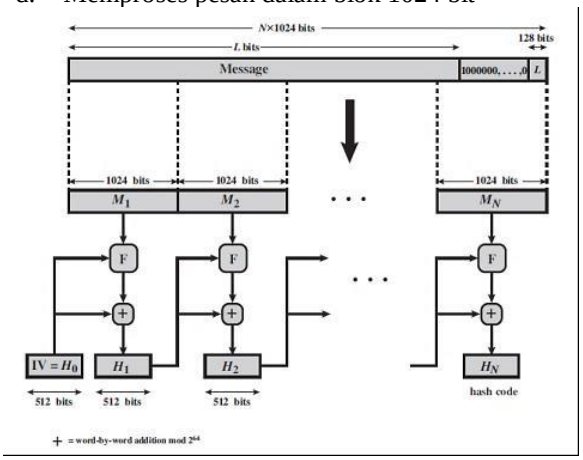
Algoritma SHA-512 menerima masukan string yang kurang dari 2128 bits dan akan menjadikan keluaran berupa 512 bit message. Pesan masukkan akan diproses per-1024 bit dan terdiri atas beberapa tahap, yaitu:

a. Menambah jumlah dari bit pesan. Pesan diberi lapisan padding agar semua pesan berukuran 1024 bit.
b. Menambah panjang bits. blok sebanyak 128 bit ditambahkan pada akhir pesan.
c. Inisialisasi hash buffer. Buffer sebesar 1024 bit digunakan sebagai nilai awal untuk menghasilkan fungsi hash. Isi dari 8 nilai awal tersebut adalah sebagai berikut :

Tabel 1: Tabel Inisialisasi

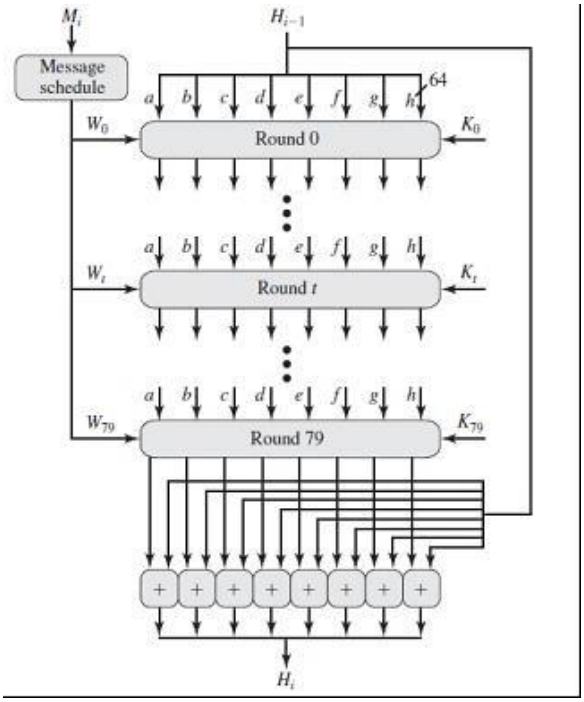
Huruf	Insialisasi()
A	6a09e667f3bcc908
B	bb67ae8584caa73b
C	3c6ef372fe94f82b
D	a54ff53a5f1d36f1
E	510e527fade682d1
F	9b05688c2b3e6c1f
G	1f83d9abfb41bd6b
H	5be0cd19137e2179

d. Memproses pesan dalam blok 1024 bit



Gambar 2:Proses I Algoritma SHA 512

Babak ini menjadi bagian utama dan akan dilakukan proses pengulangan sebanyak 80 kali. Babak ini direpresentasikan sebagai F pada Gambar 2. Setiap babak mendapatkan masukan sebesar 512 *bit* nilai *buffer*. Variable A,B,C,D,E,F,G,H akan memperbarui nilai *buffer* awal menjadi nilai yang baru. pada *input* babak pertama, *buffer* memiliki nilai awal, $Hi - 1$. Setiap babak (*round)* t memakai 64 *bit* nilai Wt, diperoleh dari 1024 *bit* blok yang sekarang akan diproses (Mi). Nilai-nilai ini diperoleh dengan menggunakan *message schedule* yang akan dijelaskan kemudian. Setiap proses loop juga mengunakan konstanta tambahan Kt, dimana $0 \leq t \leq 79$. Hasil dari proses loop ke 80 menjadi masukan babak pertama (Hi) untuk dihitung di blok selanjutnya.



Gambar 3:Proses II Algoritma SHA 512

e. Setelah semua 1024 bit blok sudah diproses, hasil dari loop yang terakhir adalah 512 *bit message*. Secara singkat, proses SHA-512 dapat disimpulkan sebagai berikut :

$H0 = IV$
$Hi = SUM64\ (Hi - 1, abcdefghi)$
$MD = HN$

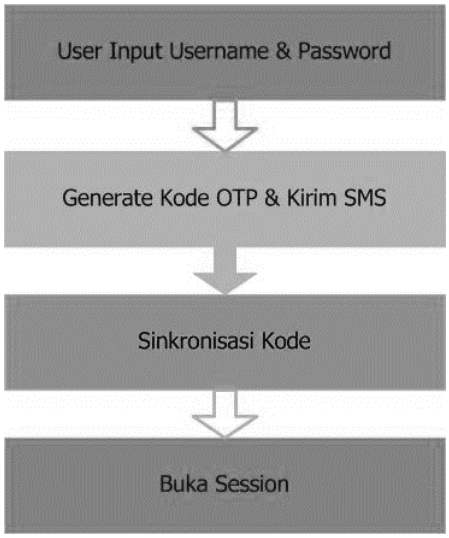
Dimana : IV = *Initial Value* (nilai awal) dari *buffer* abcdefgh yang tercantum dalam tahap ketiga, abcdefghi adalah hasil dari loop terakhir dari tahap ke empat, N adalah jumlah blok dalam pesan, dan SUM64 adalah penambahan modulus 264, sementara MD adalah hasil dari message. [7]

3. PERANCANGAN PROGRAM

3.1 Alur Program

Untuk melakukan dan menyelesaikan proses *login*, *user* diharuskan untuk mengisi *username* dan *password* lalu men-klik tombol *login*. Setelah itu *user* akan dipindahkan ke halaman verifikasi dimana *user* akan dikirimkan kode verifikasi melalui *smartphone* nya, *user* diharuskan untuk memasukkan kode yang sama kedalam halaman verifikasi tersebut, apabila kode yang dimasukkan sesuai, maka *user* akan dipindahkan ke halaman utama (*home*), apabila kode yang dimasukkan tidak sesuai maka akan muncul *popup* dan *user* akan dikembalikan ke halaman awal.

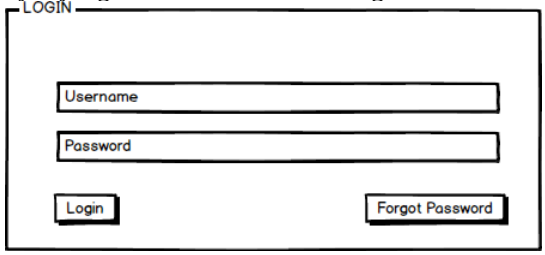
Selain diadakannya sistem OTP, program juga menggunakan notifikasi email dengan geolocation berdasarkan IP dimana program diakses



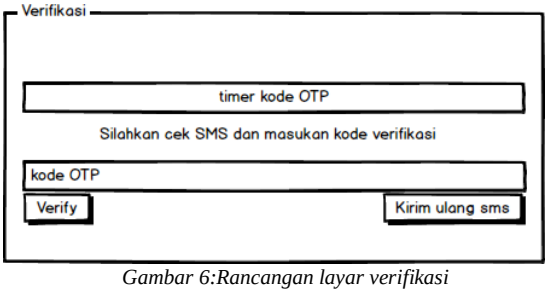
Gambar 4:Arsitektur kerja One Time Password.

3.2 Rancangan Layar

Dalam proses pembuatan program, dibutuhkan rancangan layar yang akan dibuat pada program. Rancangan layar dibuat agar *user* lebih mudah memahami program yang akan dibuat. Rancangan layar yang akan dibuat adalah sebagai berikut:



Gambar 5:Rancangan layar login



Gambar 6:Rancangan layar verifikasi

Sesuai dengan alur rancangan program, *user* pertama kali akan dihadapkan pada halaman login dimana *user* diharuskan untuk menginput *username* dan *password*-nya masing masing. Ketika informasi yang dimasukkan valid, maka *user* selanjutnya akan dihadapkan pada halaman verifikasi dimana kode

OTP akan dikirimkan ke nomor telpon *user* yang terdaftar dan *user* diharuskan untuk memasukkan kode yang sesuai ke halaman verifikasi agar bisa melanjutkan ke halaman selanjutnya.

4. EVALUASI PROGRAM

Evaluasi program merupakan tahap terakhir yang perlu dilakukan dalam pengembangan suatu sistem. Evaluasi program bertujuan untuk mengetahui hasil dan menentukan baik kekurangan maupun kelebihan dari sistem yang telah dibuat. Maka dari itu dilaksanakan sebuah percobaan yang menguji berhasil maupun gagalnya akses berdasarkan beberapa kondisi

Tabel 2: Tabel Pengujian

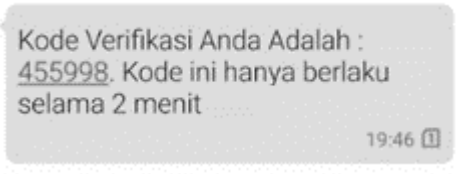
No	Waktu Login	Kode Diterima	Waktu Saat Input	Kode Di Input	Hasil
1	10:11	689774	10:11	689774	Berhasil
2	11:22	677168	11:23	677168	Berhasil
3	11:30	234552	11:33	234552	Gagal
4	11:42	336503	11:43	336503	Berhasil
5	13:34	825980	13:35	789789	Gagal
6	17:21	133431	17:21	133431	Berhasil
7	22:27	-	22:27	111222	Gagal

Berdasarkan hasil tabel pengujian diatas yang menggunakan beberapa parameter berupa waktu dan kode yang dibuat menggunakan fungsi hash SHA-512, maka dapat disimpulkan beberapa hal sebagai berikut:

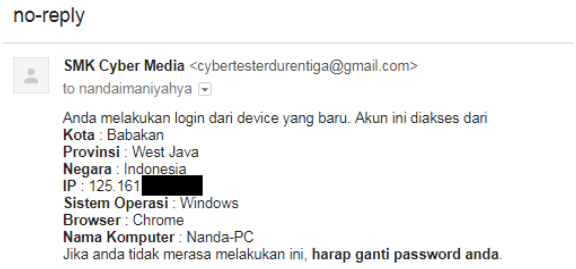
1. Dinyatakan berhasil karena kode yang diinput sesuai dengan kode yang di-*generate* oleh server dan waktu meng-*input* kurang dari 2 menit.
2. Dinyatakan berhasil karena kode yang diinput sesuai dengan kode yang di-*generate* oleh server dan waktu meng-*input* kurang dari 2 menit.
3. Dinyatakan gagal karena waktu menginput lebih dari 2 menit walaupun kode yang di-*input* sesuai dengan kode yang di-*generate* oleh *server*.
4. Dinyatakan berhasil karena kode yang di-*input* sesuai dengan kode yang di-*generate* oleh server dan waktu meng-*input* kurang dari 2 menit.
5. Dinyatakan gagal karena kode yang di-*input* tidak sesuai dengan kode yang di-*generate* oleh *server* walaupun waktu meng-*input* kurang dari 2 menit.
6. Dinyatakan berhasil karena kode yang diinput sesuai dengan kode yang di-*generate* oleh server dan waktu meng-*input* kurang dari 2 menit.
7. Dinyatakan gagal karena kode yang di-*generate* oleh *server* tidak disampaikan ke pengguna melalui SMS karena masalah pada SMS *Gateway*, maka pengguna tidak mengetahui apa kode yang harus dimasukkan.

a. Kelebihan
   1) Dapat digunakan dalam sistem operasi apapun yang memiliki akses internet.
   2) Memiliki *requirement* yang rendah sehingga sistem dapat dijalankan dengan hambatan yang sedikit.
   3) Memiliki batas waktu dalam penggunakan kode OTP.
   4) Kode OTP hanya dapat digunakan pada sesi yang ditujukan, apabila kode OTP digunakan untuk sesi pengguna lain maka kode tersebut tidak akan berfungsi.
   5) Pengguna bisa menerima notifikasi apabila akunnya sedang digunakan oleh pengguna lain tanpa sepengetahuannya di lokasi yang berbeda.
   6) Estimasi pengiriman pesan dari *trigger* sampai ke *handphone* pengguna cukup singkat (~5 detik).

b. Kekurangan
   1) Hanya menggunakan 1 algoritma yaitu SHA-512 dalam proses pembangkitan kode OTP, dan belum digabungkan dengan algoritma lainnya.
   2) Pengguna diharuskan memiliki *handphone* yang bisa menerima SMS.
   3) Pengiriman SMS kode OTP bergantung kepada server SMS *Gateway*.



Gambar 7:Pesan kode verifikasi



Gambar 8:Pesan notifikasi email

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan analisis, perancangan, serta implementasi yang telah dilakukan terhadap sistem yang dikembangkan, maka dapat diambil kesimpulan mengenai tahapan proses OTP terhadap masalah keamanan *login* di SMK Cyber Media, kesimpulan tersebut berupa:

a. Adanya penambahan metode verifikasi berupa kode OTP terbukti dapat meningkatkan keamanan hak akses akun *user*.
b. Hanya *user* yang bersangkutan yang dapat masuk kedalam sistem karena kode OTP hanya dikirimkan ke *handphone user* tersebut.
c. *User* tetap memiliki kendali atas akunnya karena sistem melacak keberadaan *ip address* tempat diaksesnya akun tersebut.

5.2 Saran

Implementasi *One Time Password* dengan Algoritma Hash SHA-512 dan IP Logging Berbasis Web Pada SMK Cyber Media masih memiliki beberapa kekurangan dari segi teknis, sehingga penulis menyarankan dalam pengembangan selanjutnya berupa:

a. Dapat dikombinasikan algoritma tambahan dalam proses *generate* kode OTP agar dapat ditambah keamanannya.
b. Dapat dipercepat proses pengiriman SMS dari *trigger* hingga ke *handphone user*.
c. Dapat dipercepat proses pengiriman *email* terkait notifikasi *login* maupun *forgot password* dari *trigger* hingga ke *email user*.

DAFTAR PUSTAKA

[1] H. Mulyono and Rodiah, "Implementasi Algoritma One Time Pad Pada Penyimpanan Data Berbasis Web," *Semin. Nas. Teknol. Inf. dan Multimed.*, vol. 1, no. 1, pp. 17–39, 2013.

[2] K. I. Santoso, "Dua Faktor Pengamanan Login Web Menggunakan Otentikasi One Time Password Dengan Hash SHA," *Semin. Nas. Teknol. Inf. Komun. Terap. 2013*, pp. 204–210, 2013.

[3] T. Abdulghani, "Enkripsi dan Dekripsi Data Text dengan menggunakan Metode Enkripsi (SHA-2, AES, TWOFISH)," *Media Jurnal Informatika*. pp. 55–60, 2013.

[4] W. Wibisono, B. A. Pratomo, and D. R. L. H, "Pengembangan Mekanisme One Time Password dengan Menggunakan Strategi Dual Channel pada Aplikasi Web," *J. Tek. Pomits*, vol. 2, no. 1, pp. 1–6, 2013.

[5] Z. Musliyana, T. Y. Arif, and R. Munadi, "Peningkatan Sistem Keamanan Autentikasi Single Sign On (SSO) Menggunakan Algoritma AES dan One-Time Password Studi Kasus: SSO Universitas Ubudiyah Indonesia," *J. Rekayasa Elektr.*, vol. 12, no. 1, p. 21, 2016.

[6] K. L. R. S. Himaja and T. S. Lakshmi, "Providing Security to User Data using OTP and Image CAPTCHA," *Int. J. Comput. Organ. Trends*, vol. 37, no. 1, pp. 17–19, 2016.

[7] H. Agung and Ferry, "Kriptografi Menggunakan Hybrid Cryptosystem dan Digital Signature," *J. Jatisi*, vol. 3, no. 1, pp. 34–45, 2016.