PENINGKATAN SISTEM KEAMANAN *ONE TIME PASSWORD* (OTP) PADA TOKEN APLIKASI *COMPUTER BASED TEST* (CBT) MENGGUNAKAN ALGORITMA SHA-256

Nur Iswatun Khasanah

Program Studi S1 Teknik Informatika Fakultas Teknologi Informasi Universitas Hasyim Asy'ari
E-mail : nuriswatun15@gamil.com

Ahmad Heru Mujianto

Program Studi S1 Sistem Informasi Fakultas Teknologi Informasi Universitas Hasyim Asy'ari
E-mail : ahmadmujianto@unhasy.ac.id

Sri Widoyoningrum

Program Studi S1 Sistem Informasi Fakultas Teknologi Informasi Universitas Hasyim Asy'ari
E-mail : swidoyoningrum@gmail.com

Abstrak

Keamanan sistem sering digunakan di Sekolah untuk mengamankan data sekolah ataupun data lainnya. Seperti di Sekolah Menengah Kejuruan (SMK) Kreatif Hasbullah Bahrul Ulum. Sekolah tersebut masih menggunakan cara manual yang masih banyak mengalami kendala diantaranya kekurangan kertas soal atau lembar jawaban dan biaya yang dikeluarkan pihak sekolah cukup banyak. Untuk mengatasi permasalahan tersebut peneliti membuat sebuah aplikasi *Computer Based Test* (CBT) dengan keamanan OTP jika ketika *login username* dan *password* diketahui orang lain. Pada penelitian yang dilakukan memakai algoritma *Secure Hash Algorithm* 256 bit (SHA-256) yang digunakan sebagai pembangkit OTP. Proses untuk mencari nilai string yang sesuai dihitung secara fleksibel yang menghasilkan menghasilkan *message digest* dan dapat mencari nilai string yang berbeda dan menghasilkan *message digest yang sama nilainya*, jadi SHA-256 bisa dikatakan keamanan yang akurat. Hasil dari penelitian ini adalah sebuah aplikasi CBT dengan keamanan kode token, dari metode SHA-256 menghasilkan kode token yang digunakan untuk login siswa. Token berfungsi sebagai keamanan ketika mengakses sistem *login* jika orang lain mengetahui *username* dan sandi, orang tersebut tidak dapat *login* jika tidak memiliki kode token dan kode token hanya diberikan sekali ketika ujian berlangsung terkecuali terjadi kendala.

Kata Kunci: Keamanan, *Computer Based Test*, *One Time Password*, *Secure Hash Algorithm*, Token.

Abstract

System security is often used in schools to secure school data or other data. As in Hasbullah Bahrul Ulum Creative Vocational High School (SMK). The school still uses the manual method, which still has many problems, including the lack of question paper or answer sheets and the school's quite a lot of costs. To overcome this problem the researchers created a Computer Based Test (CBT) application with OTP security if when logging in the username and password are known to other people. In the research conducted using the Secure Hash Algorithm 256 bit (SHA-256) which is used to generate OTP. The process for finding the appropriate string value is calculated flexibly which results in producing a message digest and can search for different string values and produce a message digest that has the same value, so SHA-256 can be said to be an accurate security. The result of this research is a CBT application with token code security, from the SHA-256 method it generates a token code used for student login. The token functions as security when accessing the login system if someone else knows the username and password, that person cannot log in if they don't have the token code and the token code is only given once during the exam unless there are problems.

Keywords: *Security, Computer Based Test, One Time Password, Secure Hash Algorithm, Token*

PENDAHULUAN

Otentikasi kata sandi diperlukan untuk mencegah pencurian informasi. Pada dasarnya yang dapat mengetahui *password* atau sandi hanya pemiliknya saja, namun ada juga orang yang bisa menyadap sandi tersebut. Keamanan sistem sering digunakan di Sekolah untuk mengamankan data sekolah ataupun data lainnya. Misalnya di Sekolah Menengah Kejuruan (SMK) Kreatif Hasbullah Bahrul Ulum merupakan sekolah SMK di Jl. KH. Wahab Hasbullah No.123 Tambakrejo Jombang. Moch.Hasan Baihaqi,SH (waka humas) menjelaskan bahwa, ketika ujian di SMK tersebut masih menggunakan cara manual yaitu menggunakan kertas. beliau menyatakan jika keamanan atau pengawasan ketika ujian juga kurang maksimal, banyak siswa yang melakukan kecurangan seperti mencontek bahkan sampai menukar lembar jawaban ke teman lainnya

Untuk mengatasi permasalahan di atas, peneliti, Menurut Bull dan Mckenna (2004), CBT adalah penilaian hasil belajar siswa dalam tes menggunakan komputer. Penilaiannya hasil belajar ini diklasifikasikan meliputi tes individu, tes sumaktif dan tes diaknootik.

Peningkatkan keamanan sistem dengan menggunakan *One Time Password* yang akan menghasilkan kode sekali pakai dan dalam waktu yang telah ditentukan. Token adalah kode yang bersifat rahasia atau kunci otentikasi yang digunakan sebagai keamanan yang berupa informasi tentang pemilik (Jammes, Mensch & Smit 2005). Pada penelitian yang dilakukan memakai algoritma *Secure Hash Algorithm* 256 bit (SHA-256) yang digunakan pembangkit OTP. Proses untuk mencari nilai string yang sesuai dihitung secara fleksibel yang menghasilkan menghasilkan *message digest* dan dapat mencari nilai string yang berbeda dan menghasilkan initisari pesan yang nilainya sama, jadi SHA-256 bisa dikatakan keamanan yang akurat. Asal mulanya algoritma SHA-256 adalah dari algoritma SHA-0 dan SHA-1 yang dikembangkan dan disempurnakan dari algoritma sebelumnya, terdapat berbedaan dari algoritma sebelumnya yaitu nilai blok yang digunakanDesain dari algoritma SHA-256 hampir sama dengan algoritma SHA-2, tetapi berbagai macam serangan tidak dapat dilakukan SHA-256 ini.

Berdasarkan dari latar belakang tersebut, penelitian ini meningkatkan keamanan sistem *login One Time Password* (OTP) dengan menggunakan algoritma SHA-256 sebagai pembangkit kode OTP pada aplikasi CBT, maka peneliti ingin mengangkat penelitian dengan judul *"*Peningkatan Sistem Keamanan *One Time Password* (OTP) Pada Token Aplikasi *Computer Based Test* (CBT) Menggunakan Algoritma SHA-256*"*, yang dapat mengatasi celah kecurangan yang dapat timbul dari ujian yang masih menggunakan kertas sehingga dapat mengukur kemampuan siswa, dan juga dapat membantu para guru untuk memberikan nilai secara cepat dan mudah, dan dapat mengurangi biaya yang dikelurkan oleh pihak sekolah.

Computer Based Test (CBT)

Penerapan *CBT* untuk menggantikan ujian yang menggunakan media kertas dengan ujian menggunakan media computer dengan memperhatikan unsur seperti keamanan system, kemampuan dasar dalam penggunaan computer dan kemudahan system dalam pengoperasian. Dalam sistem ini peserta tes akan mendapatkan sebuah *username, password* dan token sebagai keamanannya yang digunakan katika akan *login* pada system. Untuk jaringannya ketika USBK dapat menggunakan *Client Server* agar dapat berjalan dengan baik computer peserta ujian terhubung dengan computer *server* (S. Al-Amri, 2008 : 22-44).

One Time Password (OTP)

One Time Password (OTP) yaitu kode yang memiliki keunikan yang berlaku sekali pakai dalam waktu yang telah ditentukan. OTP bisa dikatakan tidak memiliki kelemahan seperti *password,* OTP yang bersifat satu kali pakai dapat menghidari serangan perulangan *password* yang dilakukan orang lain. Itu artinya bahwa orang lain tidak dapat menggunakan kode secara berulang karena kode akan tidak berlaku jika dipakai berkali kali (Editya & Mulyati, 2018).

Website

Website merupakan beberapa laman yang berisi informasi, bisa berupa data digital, teks, gambar, video, audio hingga animasi yang dikumpulan menjadi satu halaman yang dapat diakses jika terdapat jaringan internet (Abdullah, 2015).

Secure Hash Algorithm (SHA-256)

Secure Hash Algorithm(SHA) 256 yaitu salah satu dari hash yang telah dipakai, hingga sejauh ini belum ada yang berhasil mengalahkan algoritma SHA-256. Terdapat 8 langkah pengerjaan dalam algoritma SHA-256 yaitu sebagai berikut (Rachmawati, D, dkk. 2018).

1) *Padding Bit*

Padding bit ditambahkan karena pada algoritma ini dibutuhkan sepanjang 512 bit dalam satu blok data *input*, jadi jika *input*an < 512 bit maka *padding bit akan* ditambahkan dimulai dengan 1 kemudian menambahkan dengan 0.

2) Panjang Append
Pada langkah ini panjang pesan dibuat kelipatan 512 bit dari Panjang pesan 64 bit, kemudian hasilnya ditambahkan di hasil akhir.

3) *Parsing* pesan
Dengan menambahkan blok 64 bit, *padding* pesan dijabarkan menjadi N blok pesan 512 bit.

4) Inisialisasi Nilai Hash
Nilai inisialisasi awal dan hasil nilai sementara yang disimpan yang menggunakan penyangga, yaitu :
H [1] => bb67ae85
H [2] => 3c6ef372
H [3] => a54ff53a
H [4] => 510e527f
H [5] => 9b05688c
H [6] => 1f83d9ab
H [7] => 5be0cd19

5) Penjadwalan Pesan
Penjadwalan pesan dilakukan menggunakan fungsi pada setiap putaran yaitu :

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{(256)}(W_{t-2}) + W_{t-7} + \sigma_0^{(256)}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases} \quad (1)$$

Dimana :

$\sigma_1^{(256)}(W_{t-2}) = ((W_{t-2})\text{ROTR17}) \oplus ((W_{t-2})\text{ ROTR19}) \oplus ((W_{t-2})\text{ SHR10})$ (2)

$\sigma_0^{(256)}(W_{t-2}) = ((W_{t-15})\text{ROTR7}) \oplus ((W_{t-15})\text{ ROTR18}) \oplus ((W_{t-15})\text{ SHR3})$ (3)

W_t : Nilai blok baru
M_t : Nilai blok lama
W_{t-2} : Banyaknya bit di satu pesan
W_{t-15} : Banyaknya bit di satu pesan
RORT : *Rotate Right*
SHR : *Shift Right*
$\oplus$: Operasi *XOR*

6) Inisialisasi variabel a, b, c, d, e, f, g, dan h dengan nilai hash (i-1).

$T_1 = h + \Sigma_1^{(256)}(e) + \text{Ch}(e,f,g) + K_t^{(256)} + W_t$ (4)

$T_2 = \Sigma_0^{(256)}(a) + Maj(a,b,c)$ (5)

$h = g$ $\quad d = c$
$g = f$ $\quad c = b$
$f = e$ $\quad b = a$
$e = d + T_1$ $\quad a = T_1 + T_2$

Dimana :

$\Sigma_1^{(256)}(e) = (e\text{ ROTR6}) \oplus (e\text{ ROTR11}) \oplus (e\text{ ROTR25})$ (6)

$\Sigma_0^{(256)}(a) = (a\text{ ROTR2}) \oplus (a\text{ ROTR13}) \oplus (a\text{ ROTR22})$ (7)

$\text{Ch}(e,f,g) = (e \wedge f) \oplus (\neg e \wedge g)$ (8)

$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge b) \oplus (b \wedge c)$ (9)

a, b, c, d, e, f, g, h : Variabel dalam heksadesimal
$K_t^{(256)}$: Konstanta SHA-256
$\wedge$: Operasi *AND*

Tabel 1 Konstanta SHA-256

428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da

983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
2e1b2138	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2

7) Menghitung nilai hash, dengan nilai i=1

$H_0^{(i)} = a + H_0^{(i-1)}$
$H_1^{(i)} = b + H_1^{(i-1)}$
$H_2^{(i)} = c + H_2^{(i-1)}$
$H_3^{(i)} = d + H_3^{(i-1)}$
$H_4^{(i)} = e + H_4^{(i-1)}$
$H_5^{(i)} = f + H_5^{(i-1)}$
$H_6^{(i)} = g + H_6^{(i-1)}$
$H_7^{(i)} = h + H_7^{(i-1)}$

8) Output atau Hasil

$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$

METODE

Tujuan dari penelitian ini adalah merancang dan membuat suatu sistem *Computer Baset Test* agar dapat membantu para siswa dan guru di sekolah SMK Kreatif Hasbullah Tambak Beras Kab Jombang. Sistem ini dirancang menggunakan algoritma SHA-256. Sistem tersebut dilengkapi dengan keamaan dengan menggunakan token, sehingga dapat mencegah kecurangan siswa ketika mengerjakan ujian. *Input* dari sistem ini berupa data pengguna (siswa) lalu sistem akan memproses dengan metode SHA-256 untuk memberikan kode token yang digunakan untuk keamanan. *Output* dari sistem ini berupa kode token yang berlaku hanya satu jam atau waktu yang telah ditentukan dan akan berganti secara otomatis. Sistem ini dapat diakses ketika admin sudah mengaktifkan soal ujian atau ketika ujian berlangsung sesuai jadwal dan hanya bisa digunakan di Sekolah dengan komputer yang telah disediakan. Tahapan penelitian, dimulai dengan mengidentifikasi permasalahan yang akan diangkat sebagai bahan penelitian. pada tahapan ini melakukan pemahaman terhadap masalah yang terjadi atas dasar latar belakang masalah yang telah dijelaskan sebelumnya. Tahapan selanjutnya studi literatur, dimana Studi literatur merupakan sebuah referensi keamanan komputer untuk membantu dalam proses rancang bangun pada sistem Computer Based Test (CBT). Tahapan pengumpulan data, data diperoleh dari sumber secara tidak langsung melalui perantara yaitu diperoleh dan dicatat oleh lain pihak. Tahapan terakhir yaitu pengujian pada sistem dilaksanakan guna mengetahui kinerja pada sistem berfungsi layak atau belum. Pengujian ini untuk mengetahui keamanan keakuratan dalam kode token yang dibangkitkan oleh metode SHA-256.

HASIL DAN PEMBAHASAN

A. Penerapan Algoritma SHA256

Berikut langkah – langkahpenerapan metode SHA-256 :

Pesan awal (M) = Admin2023-06-05 19:45:45

Input dalam bilangan biner :

01000001	01100100	01101101	01101001	01101110	00110010	00110000	00110010
00110011	00101101	00110000	00110110	00101101	00110000	00110101	00100000
00110001	00111001	00111010	00110100	00110101	00111010	00110100	00110101

1) Penambahan *Padding Bit*

Penambahan *Padding Bit* dengan menggunakan rumus

k = *l* + 1 = 448 mod 512

k = 192 + 1 = 448 mod 512

k = 193 = 448 mod 512

k =448 – 193 = 255, jadi *padding bit* ‘0’ yang ditambahkan sebanyak 255.

01000001	01100100	01101101	01101001	01101110	00110010	00110000	00110010	00110011	00101101

00110000 00110110 00101101 00110000 00110101 00100000 00110001 00111001 00111010 00110100
00110101 00111010 00110100 00110101 10000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000

2) Panjang Append

Pada langkah ini panjang data atau pesan dibuat kelipatan 512 bit dari Panjang data atau pesan 64 bit, kemudian hasilnya ditambahkan di hasil akhir.

01000001 01100100 01101101 01101001 01101110 00110010 00110000 00110010 00110011 00101101
00110000 00110110 00101101 00110000 00110101 00100000 00110001 00111001 00111010 00110100
00110101 00111010 00110100 00110101 10000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 11000001

3) *Parsing* pesan

Tahapan selanjutnya pesan tidak boleh > 512 jadi hanya menghasilkan 1 blok 512bit yaitu $M_0^{(0)}$ hingga $M_{15}^{(0)}$. Kemudian dilakukan *parsing* pesan setiap blok 512 bit dibagi menjadi 16 blok berukuran 32 bit.

Tabel 2 *Parsing* pesan

Data	Biner	Heksadesimal
$M_0^{(0)}$	01000001 01100100 01101101 01101001	41646d69
$M_1^{(0)}$	01101110 00110010 00110000 00110010	6e323032
$M_2^{(0)}$	00110011 00101101 00110000 00110110	332d3036
$M_3^{(0)}$	00101101 00110000 00110010 00100000	2d303220
$M_4^{(0)}$	00110001 00110000 00111010 00110101	31393a34
$M_5^{(0)}$	00110011 00111010 00110101 00110010	353a3435
$M_6^{(0)}$	10000000 00000000 00000000 00000000	80000000
$M_7^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_8^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_9^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_{10}^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_{11}^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_{12}^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_{13}^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_{14}^{(0)}$	00000000 00000000 00000000 00000000	00000000
$M_{15}^{(0)}$	00000000 00000000 00000000 11000001	000000c0

4) Inisialisasi Nilai Hash

Langkah dilakukan parsing pesan maka kemudian dilakukan inisialisasi nilai hash yang merupakan sebuah ketentuan yaitu:

Tabel 3 *Inisialisasi* Nilai Hash

Variabel	Inisialisasi	Nilai Hash
$H_0^{(0)}$	a	6a09e667
$H_1^{(0)}$	b	bb67ae85
$H_2^{(0)}$	c	3c6ef372
$H_3^{(0)}$	d	a54ff53a
$H_4^{(0)}$	e	510e527f
$H_5^{(0)}$	f	9b05688c
$H_6^{(0)}$	g	1f83d9ab
$H_7^{(0)}$	h	5be0cd19

5) Penjadwalan Pesan

Selanjutnya penjadwalan pesan dimulai dari merubah setiap nilai blok pesan menjadi bilangan heksadesimal dengan rumus sebagai berikut:

$$Wt = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{(256)}(W_{t-2}) + W_{t-7} + \sigma_0^{(256)}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases} \quad (1)$$

Untuk pesan ke 16 hingga 63 dihitung sebagai berikut:

t = 16

$\sigma_1^{(256)}(W_{t-2}) = ((W_{t-2})ROTR17) \oplus ((W_{t-2})\ ROTR19) \oplus ((W_{t-2})\ SHR10)$ (2)

$((W_{16-2})ROTR17) = ((W_{14})ROTR\ 17)$
$= ((00000000)RORT\ 17)$
$= (00000000)$

$((W_{16-2})ROTR19) = ((W_{14})ROTR\ 19)$
$= ((00000000)RORT\ 19)$
$= (00000000)$

$((W_{16-2})ROTR10) = ((W_{14})SHR\ 10)$
$= ((00000000)\ SHR\ 10)$
$= (00000000)$

$\sigma_1^{(256)}(W_{i-2}) = (00000000) \oplus (00000000) \oplus (00000000)$
$=$ **(00000000)**

$W_{16-7} = W_9$
$=$ **00000000**

$\sigma_0^{(256)}(W_{t-2}) = ((W_{t-15})ROTR7) \oplus ((W_{t-15})\ ROTR18) \oplus ((W_{t-15})\ SHR3)$ (3)

$((W_{16-15})ROTR7) = ((W_1)ROTR7)$
$= ((6e323032)\ ROTR7)$
$=$ **64dc6460**

$((W_{16-15})ROTR18) = ((W_1)ROTR18)$
$= ((6e323032)\ ROTR18)$
$=$ **8c0c9b8c**

$((W_{16-15})\ SHR3) = ((W_1)\ SHR3)$
$= ((6e323032)\ SHR3)$
$=$ **0dc64606**

$\sigma_0^{(256)}(W_{16-16}) = (64dc6460) \oplus (8c0c9b8c) \oplus (0dc64606)$
$=$ **e516b9ea**

$W16\text{-}16 = W_0$
$=$ **41646d69**

$Wt = \sigma_1^{(256)}(W_{i-2}) + W_{i-7} + \sigma_0^{(256)}(W_{i-15}) + W_{i-16}$
$Wt = 00000000 + 00000000 + e516b9ea + 41646d69$
$Wt =$ **267b2753**

6) Inisialisasi Kerja Variabel

Dilakukan inisialisasi variable a, b, c, d, e, f, dan h dengan mengambil setiap variable dari nilai hash a = $H_0^{(0)}$, b = $H_1^{(0)}$, c = $H_2^{(0)}$, d = $H_3^{(0)}$, e = $H_4^{(0)}$, f = $H_5^{(0)}$, g = $H_6^{(0)}$, h = $H_7^{(0)}$. Kemudian proses fungsi *hash* dimulai dari t=0 hingga t=63.

t=0

a	b	c	d	e	f	g	h
6a09e667	bb67ae85	3c6ef372	a54ff53a	510e527f	9b05688c	1f83d9ab	5be0cd19

(4)

$T_1 = h + \Sigma_1^{(256)}(e) + Ch(e,f,g) + K_t^{(256)} + W_t$
h = 5be0cd19
$\Sigma_1^{(256)}(e) = (e\ ROTR\ 6) \oplus (e\ ROTR\ 11) \oplus (e\ ROTR\ 25)$ (6)
$\Sigma_1^{(256)}(e) = ((510e527f)\ ROTR\ 6) \oplus ((510e527f)\ ROTR\ 11) \oplus ((510e527f)\ ROTR\ 25)$
$\Sigma_1^{(256)}(e) = (fd443949) \oplus (4fea21ca) \oplus (87293fa8)$
$\Sigma_1^{(256)}(e) = 3587272b$
$Ch(e,f,g) = (e \wedge f) \oplus (\neg e \wedge g)$ (8)
$Ch(e,f,g) = (510e527f \wedge 9b05688c) \oplus (\neg 510e527f \wedge 1f83d9ab)$
$Ch(e,f,g) = (1104400c) \oplus 90e818980)$
$Ch(e,f,g) =$ **1f85c98c**
$K_0^{(256)} = 428a2f98$
$Wt = 267b2753$
$T_1 = 5be0cd19 + 3587272b + 1f85c98c + 428a2f98 + 267b2753$
$T_1 = 134dc5ad1$
$T_2 = \Sigma_0^{(256)}(a) + Maj(a,b,c)$ (5)
$\Sigma_0^{(256)}(a) = (a\ ROTR2) \oplus (a\ ROTR13) \oplus (a\ ROTR22)$ (7)
$\Sigma_0^{(256)}(a) = (6a09e667\ ROTR\ 2) \oplus (6a09e667\ ROTR\ 13) \oplus (6a09e667\ ROTR\ 22)$
$\Sigma_0^{(256)}(a) =$ **ce40b47e**
$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge b) \oplus (b \wedge c)$ (9)
$Maj(a,b,c) = (6a09e667 \wedge bb67ae85) \oplus (6a09e667 \wedge bb67ae85) \oplus (bb67ae85 \wedge 3c6ef372)$
$Maj(a,b,c) =$ **3a6fe667**
$T_2 = 108909ae5$

h = g
= 1f83d9ab
g = f
= 9b05688c
f = e
= 510e527f
e = d + T_1
= a54ff53a + 134dc5ad1
= 1da2c500b

d = c
= 3c6ef372
c = b
= bb67ae85
b = a
= 6a09e667
a = $T_1 + T_2$
= 134dc5ad1 + 108909ae5
= 23d6cf5b6

7) Menghitung nilai hash , dengan nilai i=1.

$H_0^{(1)} = a + h_0^{(1-1)}$
= bd0a41af

$H_1^{(1)} = b + h_1^{(1-1)}$
= 69a99bc0 + bb67ae85
= 25114a45

$H_4^{(1)} = e + h_4^{(1-1)}$
= 7e695ba2 + 510e527f
= cf77ae21

$H_5^{(1)} = f + h_5^{(1-1)}$
= b07cb434 + 9b05688c
= 4b821cc0

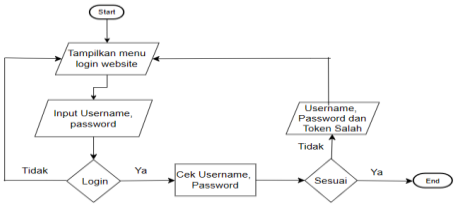
8) Hasil

Tahapan terakhir yaitu hasil akhir dari SHA-256 adalah gabungan dari H0(0) hingga H7(0) adalah :
bd0a41af || 25114a45 || b5b23ea0 || c0d29658 || cf77ae21 || 4b821cc0 || 9ef66779 || fdf0f5cb

B. Perancangan Sistem

1) *Flowchart Login* Siswa

Pada halaman utama sistem telah ditampilkan menu login yang akan di*input*kan *Username*, *Password* dan Token. *Username* dan *Password* menggunakan NIS masing-masing siswa, untuk token akan diberitahukan oleh proktor.

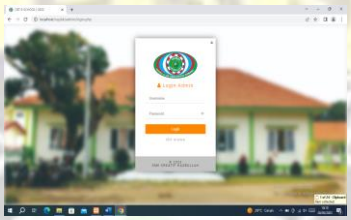


Gambar 1 *Flowchart Login* Siswa

C. Implementasi Sistem

1) Tampilan *Login* Admin

berisi form yang digunakan untuk membuka akun untuk admin dan proktor agar dapat masuk ke dalam dasboard sistem. Pada tampilan *login* admin terdapat form *username*, *password* agar dapat masuk pada sistem.



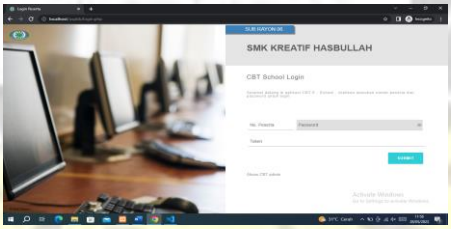
Gambar 2 Tampilan *Login* Admin

Kode token diperoleh dari implementasi metode SHA256 yang melakukan proses enkripsi dari *username* dan waktu saat ini. hasil enkripsi diambil secara acak oleh system sehingga menghasilkan sebuah kode token yang dapat digunakan untuk login siswa.

Gambar 3 Tampilan Kode Token

2) Tampilan *Login* Siswa

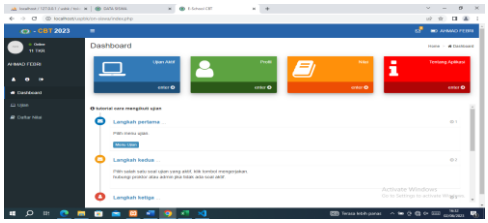
Menu login siswa berisi form no. peserta, password, juga kode token agar dapat masuk pada sistem ujian. Kode token pada login siswa dapat diperoleh dengan menghubungi admin atau proktor yang bertugas dalam melaksanakan ujian.



Gambar 4 Tampilan *Login* Siswa

3) Tampilan Halaman Dasbord Siswa

Halaman dasboard siswa merupakan tampilan setelah berhasil ke login siswa, berisi tentang beberapa tampilan informasi ujian aktif, profil, nilai, dan tentang aplikasi serta informasi USBK.



Gambar 5 Tampilan Halaman Dasbord Siswa

D. Pengujian

Implementasi *One Time Password* pada sitem *login* di aplikasi *Computer Based Test* berjalan sangat baik yang menggunakan bahasa pemograman PHP. Selanjutnya dilakukan pengujian pada *username* dan waktu yang berbeda menghasilkan kode token yang berbeda. OTP hanya berlaku selama 2 menit. Hasil pengujian metode SHA-256 sebagai berikut :

Tabel 4 Pengujian Metode SHA-256

Username	**Waktu**	**Hasil Enkripsi**	**Hasil Token**
Admin	2023-05-27 18:46:22	c38f1f27c457bd73bab6709323a78cc50436dc9770d30180597b16cb1d9bf9ce	C457BD
Admin	2023-05-27 20:33:32	e8e30ebe94883872a3339b0925ce661b38758ceff149a2bb8cb3ec13c7aef99a	58CEFF
Admin	2023-05-28 00:03:06	f652be57a1c96dd28d597ac0869468a3af0ee72fd039105ef6f2a84735133997	EE72FD
Admin	2023-05-28 10:12:15	9a2d4569a70c64ccfa3de8da9fe5de040158f407be16709bdd538093c86f9a54	DE0401

Selanjutnya malakukan pengujian pada kode token, berikut merupakan hasil dari pengujian :

Tabel 5 Pengujian Kode Token

No.	**Waktu Generate**	**Kode yang Diterima**	**Waktu yang Diinputkan**	**Kode yang Diinputkan**	**Hasil**
1.	2023-05-27 18:46:22	C457BD	2023-05-27 18:47:24	C457BD	Berhasil
2.	2023-05-2720:33:32	58CEFF	2023-05-2720:36:40	58CEFF	Gagal
3.	2023-05-28 00:03:06	EE72FD	2023-05-28 00:04:06	EE0401	Gagal
4.	2023-05-28 10:12:15	DE0401	2023-05-28 10:13:5	DE0401	Berhasil

Kode OTP atau token yang diinputkan dapat disimpulkan sebagai berikut:

1. Dikatakan berhasil karena meng*input*kan token sebelum 2 menit dan meng*input*kan token sesuai.
2. Dikatakan gagal karena terlambat dalam meng*input*kan token yaitu lebih dari 2 menit.
3. Dikatakan gagal karena meng*input*kan token tidak sesuai dengan token yang telah diterima.
4. Dikatakan berhasil karena meng*input*kan token sebelum 2 menit dan meng*input*kan token sesuai.

PENUTUP

Simpulan

Dari hasil penelitian berupa implementasi sistem dan pengujian system yang sudah dilaksanakan, sebagai berikut :

a. Dengan adanya sistem *Computer Based Test* dapat memudahkan pihak sekolah ketika ujian, kendala – kendala yang pernah terjadi dapat diatasi. Sistem *Computer Based Test* yang menggunakan algoritma SHA-256 yang digunakan sebagai pembangkit kode token pada *Computer Based Test* dengan penerapan *One Time Password* dapat meningkatkan keamanan pada kode token pada ketika *login* untuk menjaga dari berbagai serangan.
b. Hasil yang didapatkan dari pengujian pada *One Time Password* (OTP) yaitu berhasil *login* jika menginputkan OTP sesuai dengan sistem pada admin dan tidak melebihi waktu yang diberikan yaitu 2 menit. Gagal *login* jika menginputkan OTP salah atau melebihi waktu selama 2 menit.

Saran

Sistem *Computer Based Test* menggunakan keamanan OTP yang dibangkitkan oleh algoritma SHA-256 masih memliki kekurangan dan keterbatasan. Saran dari penulis adanya mengkombinasi atau menambah dengan algoritma lain sebagai pembangkit kode OTP atau dengan penambahan fitur lainnya.

DAFTAR PUSTAKA

Aisha, D., Indriyanti, A. D., & Mujianto, A.H. (2020). Rancang Bangun Web E-Commerce Menggunakan Metode Collaborative Filtering (Studi Kasus: Toko aksesories tata). Inovate: Jurnal Ilmiah Inovasi Teknologi Informasi, 5(1), 47-57.

Al-Amri, S.(2008). Computer-based testing vs. paper-based testing: A comprehensive apporoach to examining the comparability of testing modes. *Essex Graduate Student Papers in Language & Linguistics, 10,* 22-44.

Bull, J., & McKenna, C. (2004). *Blueprint fir computer-assisted assessment.* Psychology Press.

Editya, G. H., & Mulyati, S. (2018). Aplikasi Mobile One Time Password Menggunakan Algoritma MD5 dan SHA1 Untuk Meningkatkan Keamanan Website. *SKANIKA: Sistem Komputer dan Teknik Informatika, 1(2),* 618-623.

Jammes, F., Mensch, A. & Smit, H. (2005), Service-Oriented Device Communications Using the Device Profile for Web Services, ICS, ASM International Conference Proceeding Series, ACM New York, New York.

Mustofa, R. P. (2013). Aplikasi Mobile Android "One Time Password (OTP)" Untuk Meningkatkan Keamanan Otentifikasi. *Skripsi. Teknologi Informasi. AMIKOM Yogyakarta,* 1-15.

Rachmawati, D., Tarigan, J. T., & Ginting, A. B. C. (2018, March). A comparative study of Message Digest 5 (MD5) and SHA256 algorithm. In *Journal of Physics: Conference Series* (Vol. 978, p. 012116). IOP Publishing