Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 531-544
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Rancang Bangun Aplikasi Verifikasi Keaslian e-Sertifikat dengan Algoritma SHA-512

***Javier*[1]*, Yuwono Marta Dinata*[2]**
[1,2]*Fakultas Teknologi Informasi, Universitas Ciputra Surabaya, Surabaya, Indonesia*
e-mail: jemmanuel@student.ciputra.ac.id[1]*, yuwono.dinata@ciputra.ac.id*[2]

Abstract

In today's digital age, electronic certificates, also known as e-certificates, are frequently used. The authenticity of these certificates, however, can be a challenge to resolve because they are easily falsified. Rarely do certificates for events like contests, seminars, courses, and awards have serial numbers, barcodes, or QR scanners as authenticity markers. Contrary to certificates for land titles, vehicles, law, education, or academia, which are challenging to counterfeit because they have an indicator and typically have a physical form rather than a digital one. Recruiting committees, institutional memberships, and HR departments could all suffer as a result. Therefore, a solution is required to maintain the validity of these certificates using the Cryptographic Hash Function technique and the SHA-512 algorithm, which is the most recent member of the SHA-2 generation. The findings from this research indicate that the website application developed performs effectively on the user's device and facilitates the verification process of e-certificates that lack authenticity indicators. Therefore, the authors can conclude that this website application provides assistance to both individual and organizational users in verifying digital certificates.

Keywords*: e-Certificate, Hash, SHA-512, SHA-2, Cryptography*

Abstrak

Sertifikat elektronik atau dapat disebut juga sebagai e-sertifikat digunakan secara luas dalam era digital saat ini. Namun, keaslian sertifikat tersebut dapat menjadi masalah yang sulit diatasi karena mudah untuk dipalsukan. Sertifikat - sertifikat seperti lomba, kehadiran mengikuti seminar, kursus, dan penghargaan jarang sekali memiliki indikator keaslian seperti kode seri, barcode, atau QR Scanner. Berbeda dengan sertifikat tanah, kendaraan, hukum, pendidikan, atau akademik yang sulit untuk dipalsukan dikarenakan sertifikat tersebut memiliki sebuah indikator dan juga biasanya memiliki bentuk fisik, tidak dalam bentuk digital. Hal tersebut tentunya akan merugikan pihak perekrut kepanitiaan, keanggotaan lembaga, maupun HRD pekerjaan. Untuk itu dibutuhkan solusi untuk menjaga keaslian dari sertifikat – sertifikat tersebut dengan menggunakan metode Cryptographic Hash Function dengan menggunakan algoritma generasi SHA-2 yang terakhir yakni SHA-512. Dalam penelitian ini, pengujian yang dilakukan untuk rancang bangun yang telah dibuat berfungsi dengan sangat baik di perangkat pengguna dan dapat membantu proses verifikasi e-sertifikat yang tidak memiliki indikator keaslian. Maka dari itu, kesimpulan yang didapatkan adalah rancang bangun ini dapat memberikan bantuan kepada pengguna, baik individu maupun organisasi, dalam melaksanakan proses verifikasi sertifikat digital.

Kata Kunci*: e-Sertifikat, Hash, SHA-512, SHA-2, Kriptografi*

1. PENDAHULUAN

Penghargaan yang diperoleh dalam suatu kejuaran, baik dalam bentuk fisik maupun digital, merupakan suatu kebanggaan tersendiri. Namun, penghargaan dalam bentuk digital rentan terhadap modifikasi oleh pihak yang tidak bertanggung jawab. Hal ini dapat merugikan pemilik sah sertifikat tersebut. Sejauh ini, dokumen tercetak seperti ijazah, surat-surat penting, dan sertifikat sangat mudah untuk dipalsukan [1]. Menurut *The Association of Certified Fraud Examiners*, setiap tahunnya terdapat 41% pelamar pekerjaan yang mengirimkan sertifikat palsu [2]. Oleh karena itu, verifikasi dokumen tercetak menjadi masalah yang penting. Dokumen seperti ijazah, surat tanah, dan sejenisnya biasanya dilengkapi dengan nomor identitas seperti nomor seri, barcode, atau QR *Scanner*. Namun, sertifikat non-akademik seringkali tidak dilengkapi dengan identifikasi tersebut karena biasanya diterbitkan oleh penyelenggara acara kecil yang kurang terkenal. Meskipun demikian, sertifikat tersebut tetap memiliki nilai dan keberhargaan. Ketidakberadaan nomor seri, barcode, atau QR Scanner pada suatu dokumen tidak meniadakan nilai atau pentingnya dokumen tersebut, dan tidak membuatnya tidak layak untuk disimpan. Namun demikian, dokumen tersebut tetap harus dijaga keaslian agar tidak dimodifikasi oleh pihak yang tidak bertanggung jawab. Salah satu cara untuk menjaga keaslian dokumen adalah dengan menggunakan algoritma *Cryptographic Hash Function (CHF)* yang kuat dengan tingkat kompleksitas yang sangat tinggi. Salah satu algoritma yang terkenal untuk tujuan tersebut adalah *Secure Hash Algorithm*.

Secure Hash Algorithm adalah salah satu jenis algoritma yang termasuk dalam CHF yang awalnya dikembangkan oleh *National Security Agency* (NSA) dan diterapkan sebagai standar oleh *National Institute of Standards and Technology* (NIST) dalam *Federal Information Processing Standards* (FIPS) pada tahun 1993 di Amerika Serikat. Sejak saat itu, SHA telah menjadi standar internasional untuk memverifikasi integritas data, menggantikan algoritma CHF sebelumnya yaitu MD5. Hal ini disebabkan MD5 tidak lagi memenuhi persyaratan sebagai CHF karena telah terbukti memiliki kerentanan kriptografi yang membuatnya tidak layak untuk digunakan dalam konteks keamanan data yang lebih lanjut [3]. Sebagai akibatnya, MD5 dianggap tidak berhasil sebagai algoritma CHF. Sebagai gantinya, NSA mengembangkan algoritma baru yang kemudian dipublikasikan oleh NIST dan dikenal sebagai SHA-0. Namun, SHA-0 ditarik kembali pada tahun 1995 dan direvisi menjadi SHA-1 dalam dokumen FIPS PUB 180-1. Revisi ini mengatasi beberapa kesalahan dalam algoritma sebelumnya yang mengurangi keamanan kriptografi. Namun, pada tahun 2005, SHA-1 juga dianggap tidak aman jika penyerang memiliki sumber daya komputasi yang cukup, misalnya memiliki banyak uang [4] dengan tujuan untuk meningkatkan kinerja komputasi pada CPU dan GPU, upaya dilakukan untuk meretas dua dokumen yang berbeda dengan tujuan menghasilkan checksum SHA-1 yang sama. Beberapa organisasi telah merekomendasikan untuk mencari alternatif dari algoritma tersebut sejak tahun 2010 [5]. Pada tahun 2017, CWI Amsterdam dan Google mengumumkan bahwa mereka berhasil melakukan collision attack dan menemukan checksum SHA-1 yang sama pada dua dokumen PDF yang berbeda [6].

Hingga saat ini, SHA-2 terutama SHA-224 belum mengalami *hash collision*. Namun, banyak peneliti yang berasumsi bahwa suatu saat nanti, *hash collision* pada SHA-224 bisa terjadi meskipun saat ini masih dianggap aman oleh NSA dan NIST. Mengingat pengalaman dengan algoritma-algoritma sebelumnya seperti MD5 dan SHA-1, hanya tinggal menunggu waktu saja untuk SHA-224 dapat ditembus karena perkembangan komputer yang semakin canggih seiring berjalannya waktu. Oleh karena itu, generasi terbaru dari SHA-2, yakni SHA-512, dapat menjadi solusi untuk menjaga integritas dari sertifikat-sertifikat palsu yang tidak memiliki indikator nomor serial seperti sertifikat lomba, kompetisi, kehadiran seminar, dan sejenisnya. Penelitian ini bertujuan untuk merancang dan membangun aplikasi verifikasi keaslian e-sertifikat menggunakan SHA-512 sehingga mempermudah pengguna yang bekerja sebagai individu maupun organisasi dalam melakukan proses verifikasi e-sertifikat.

2. METODOLOGI PENELITIAN

2.1. Studi terdahulu

Studi terdahulu digunakan sebagai referensi yang relevan untuk membandingkan, menganalisis, dan mengintegrasikan temuan yang diperoleh dalam penelitian ini. Penelitian sebelumnya membahas permasalahan sertifikat palsu yang telah tersebar luas dan sering digunakan untuk meningkatkan gaji melalui lamaran pekerjaan [7]. Peneliti juga menjelaskan bahwa dampak dari penggunaan sertifikat palsu tersebut terhadap proses penerimaan karyawan baru menjadi tidak efektif. Oleh karena itu, dalam penelitian ini, teknologi *blockchain* digunakan untuk meningkatkan keamanan dengan adanya kode enkripsi dan transparansi yang lebih tinggi.

Penelitian selanjutnya mengungkapkan bahwa peniruan atau pemalsuan dapat dianggap sebagai tindakan pidana yang dapat ditindaklanjuti sesuai dengan ketentuan dalam KUHP [8]. Peneliti juga menghadapi permasalahan dalam validasi sertifikat yang hanya mengandalkan foto hasil scan sertifikat sebagai dasar validasi oleh admin. Menurut peneliti, pendekatan tersebut dianggap tidak efektif dan tidak dapat memberikan kepastian terhadap keaslian sertifikat. Oleh karena itu, peneliti mengusulkan solusi dengan menggunakan aplikasi autentikasi berbasis PESSTA+ sebagai alternatif.

Penelitian lainnya yang terkait dengan topik yang diambil yakni membahas tentang kerentanan keamanan kredensial akademik terhadap serangan *cyber* [9]. Peneliti juga menjelaskan bahwa metode autentikasi terdahulu, seperti *recording*, dianggap sangat tidak efisien. Sebagai alternatif, peneliti menggunakan teknologi *smart contract* yang menyimpan informasi dalam *blockchain*, dimana fungsionalitasnya terdesentralisasi dengan menggunakan asymmetric encryption bernama ECC, yang juga digunakan dalam *Bitcoin* dan *Ethereum*.

2.2. Teknologi

Teknologi yang digunakan adalah SHA-512. SHA-512 adalah salah satu jenis dari SHA-2 yang memiliki kapasitas menampung *message size* hingga <2128 atau setara dengan 340.282.366.920.938. 463.463.374.607.431.768.211.455 bit, *block size* 1024 *bit, word size* 64 *bit*, dan digest *size* 512 bit. SHA-512 menggunakan 80 konstanta berurutan, di mana masing-masing memiliki representasi heksadesimal sebesar 4 bit. Jika hanya satu konstanta digunakan, maka konstanta tersebut akan memiliki ukuran sebesar 64 bit. Sebelum perhitungan hash dimulai, ada tahapan yang disebut *preprocessing* yang terdiri dari tiga tahapan yaitu *message padding*, *parsing padded message*, dan *setting initial hash value*. Input disimbolkan dengan ‘m’, yang artinya pesan yang diinputkan, baik itu berupa *string*, file, atau dokumen digital lainnya. Jika yang diinputkan merupakan file atau dokumen digital, maka akan dikonversi ke bilangan binary terlebih dahulu, namun proses ini tidak akan diteliti oleh penulis. Begitu juga dengan *string* atau kata, setiap karakter termasuk huruf dan angka akan direpresentasikan dalam bentuk bilangan binary, seperti huruf ‘a’ yang direpresentasikan sebagai '01100001'.

Tujuan dari *message padding* adalah untuk memastikan bahwa m dalam bentuk *bit* adalah kelipatan dari 1024 *bit*. Setelah panjang m dalam *bit* (disimbolkan sebagai ℓ) diketahui, maka dilakukan penambahan 1 *bit* bernilai ‘1’ diikuti dengan k bit bernilai ‘0’. Nilai k dihitung dengan persamaan 896 - (ℓ + 1) = k. Selanjutnya, bit selanjutnya diwakili dalam variabel 128 bit ℓ sebagai bilangan binary.

Tujuan dari message padding adalah untuk memastikan bahwa m dalam Sebagai contoh, jika m adalah ‘abc’, maka nilai ℓ dapat dihitung dengan mengalikan panjang m dengan 8 bit sehingga $\ell = 8 \times 3 = 24$. Langkah selanjutnya adalah menambahkan 1 *bit* bernilai ‘1’ ke dalam 24 bit sehingga total *bit* menjadi 25. Kemudian, nilai k dapat dihitung dengan persamaan k = 896 – (24 + 1) = 871. Selanjutnya, tambahkan 871 *bit* bernilai ‘0’ ke dalam 25 *bit* sebelumnya sehingga total *bit* menjadi 896. Untuk sisanya, yaitu 128 *bit* terakhir, akan merepresentasikan nilai ℓ (24) dalam 128 *bit* (big endian) seperti 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00011000. Dengan demikian, total bit dari m setelah padding telah menjadi 1024 *bit*.

Setelah proses message padding selesai, tahapan selanjutnya adalah parsing padded message. Message yang telah dipadding akan dibagi menjadi blok-blok dengan ukuran 1024 bit (disimbolkan sebagai N). Misalnya, pada contoh 'abc' ini, N=1 karena jumlah bitnya tepat 512 dalam satu blok. Kemudian, blok 512 *bit* ini akan dibagi menjadi 16 bagian yang masing-masing memiliki 64 *bit*. Setiap bagian ini diberi label M0, M1, M2, dan seterusnya hingga M15. Langkah terakhir dalam proses SHA-512 adalah menentukan nilai awal *hash* atau *initial hash value.* Terdapat 8 variabel yang digunakan dan dinotasikan sebagai H0, H1, H2, H3, H4, H5, H6, dan H7. Nilai awal ini telah ditentukan oleh NIST dan terdiri dari pecahan dari nilai-nilai irasional seperti akar kuadrat dari dua, tiga, lima, dan sepuluh, serta bilangan prima. Proses SHA-512 kemudian akan menggunakan variabel-variabel ini dalam setiap iterasi untuk menghasilkan nilai hash akhir. Sebagaimana telah dibahas sebelumnya, terdapat 80 konstanta dalam algoritma SHA-512 yang dinotasikan sebagai K0, K1, K2, hingga K79. Konstanta-konstanta ini didapatkan dengan mengambil 64 *bit* pertama dari bilangan pecahan hasil akar kubik dari 80 bilangan prima pertama. Dalam tahap hash computation pada algoritma SHA-512, terdapat empat langkah yang harus dilakukan. Keempat langkah ini harus diulang sebanyak N kali. Pada kasus ‘abc’, N bernilai 1 sehingga hanya perlu melakukan satu kali iterasi. Dalam langkah pertama, dilakukan message schedule yang disimbolkan dengan Wt. Penjumlahan dilakukan dengan modulo 264, kecuali XOR.

$$W_t = \sigma_1^{\{512\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{512\}}(W_{t-15}) + W_{t-16} \quad (1)$$

Mulai dari W0 hingga W63, setiap variabel memiliki 64 bit. W0 hingga W15 diisi dengan nilai yang sama dengan M0 hingga M15. Kemudian, nilai W16 hingga W63 dihitung menggunakan formula yang bergantung pada nilai Wt sebelumnya. Dalam menjabarkan formula Wt, terdapat dua fungsi yang digunakan, yaitu fungsi σ0 dan σ1 yang belum diketahui nilainya.

$$\sigma_0^{\{512\}}(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x) \tag{2}$$

$$\sigma_1^{\{512\}}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x) \tag{3}$$

Untuk contoh ini, ROTR1 dilakukan pada bilangan biner x = 11000111 00011100 01100100 11000001 dengan menggeser bit ke kanan sebanyak 1 kali. Hasilnya adalah 11100011 10001110 00110010 01100000. Sedangkan, SHR7 dilakukan dengan menggeser bit ke kanan sebanyak 7 kali dan mengisi 7 bit awal dengan nilai 0. Sehingga, hasil dari SHR7(x) adalah 00000001 10001110 00111000 11001001. Setelah itu, hasil dari ROTR1 dan SHR7 dijumlahkan dengan operasi XOR dan dilakukan iterasi hingga ditemukan message schedule terakhir yaitu W79.

Langkah kedua adalah mengonversi 8 variabel dari H0 hingga H7 yang dalam format heksadesimal menjadi format biner. Setelah itu, variabel-variabel tersebut akan diberi nama ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, dan ‘h’.

Langkah ketiga dari SHA-512 adalah melakukan fungsi kompresi, di mana dilakukan pengulangan atau looping dari t = 0 hingga t = 79.

$$T_1 = h + \sum\nolimits_1^{\{512\}}(e) + Ch(e,f,g) + K_t^{\{512\}} + W_t$$

$$T_2 = \sum\nolimits_0^{\{512\}}(a) + Maj(a,b,c)$$

$$h = g$$

$$g = f$$

$$f = e$$

$$e = d + T_1$$

$$d = c$$

$$c = b$$

$$b = a$$

$$a = T_1 + T_2$$

Fungsi T1 dan T2 yang belum diketahui dalam compression function SHA-512 menggunakan beberapa fungsi lainnya, yaitu Σ0, Σ1, Ch, dan Maj.

$$\sum\nolimits_0^{\{512\}}(x) = ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x)$$

$$\sum\nolimits_1^{\{512\}}(x) = ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x)$$

$$Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

Ch adalah singkatan dari choose, yang berarti memilih satu nilai di antara dua nilai berdasarkan kondisi tertentu. Sedangkan Maj adalah singkatan dari majority, yang berarti memilih nilai yang muncul paling banyak dari tiga nilai. Oleh karena itu, pada iterasi terakhir di mana t = 79, nilai dari variabel ‘a’, ‘b’, ‘c’, hingga ‘h’ akan berubah.

Langkah terakhir dari hash computation adalah mengubah nilai hash atau modify hash values. Pada tahap ini, nilai-nilai H0 sampai H7 akan ditambahkan dengan nilai ‘a’ sampai ‘h’.

$$H_0 = a + H_0$$
$$H_1 = b + H_1$$
$$H_2 = c + H_2$$
$$\ldots$$
$$H_7 = h + H_7$$

Setelah dilakukan modify hash values, hasilnya digabungkan dimana setiap nilai Hn terdiri dari 64 bit. Selanjutnya, nilai-nilai ini diubah menjadi format heksadesimal.

2.3. Analisa Kebutuhan

Analisa kebutuhan dilakukan dengan cara pengumpulan data kepada responden secara online. Data yang dikumpulkan berupa hasil dari pertanyaan sebagai berikut:

a) Apakah menurut responden penting jika sebuah sertifikat non-akademik diberi suatu indikator keaslian dalam rangka menghindar pemalsuan sertifikat.
b) Pertanyaan dengan menggunakan Likert Scale 1 – 4 (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju) untuk mengetahui jika responden sangat mementingkan tingkat keamanan keaslian dari sertifikat - sertifikat tersebut agar tidak dapat sembarangan dicairkan sebagai reward point.
c) Pertanyaan dengan menggunakan Likert Scale 1 – 4 (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju) untuk mengetahui jika responden membutuhkan suatu sistem yang dapat mendeteksi sebuah perubahan kecil saja terhadap sertifikat tersebut agar dapat diketahui bahwa sertifikat tersebut asli atau palsu.

Hasil jawaban dari pertanyaan tersebut, 100% responden membutuhkan aplikasi utuk verifikasi e-sertifikat, setuju terhadap mementingkan tingkat keamanan keaslian dari sertifikat – sertifikat non-akademik tersebut agar tidak dapat sembarangan dicairkan sebagai reward point, dan sangat setuju terhadap suatu sistem yang dapat mendeteksi sebuah perubahan kecil saja terhadap sertifikat tersebut agar dapat diketahui bahwa sertifikat tersebut asli atau palsu. Maka dari itu dibuatlah desain untuk aplikasi berbasis website ini.

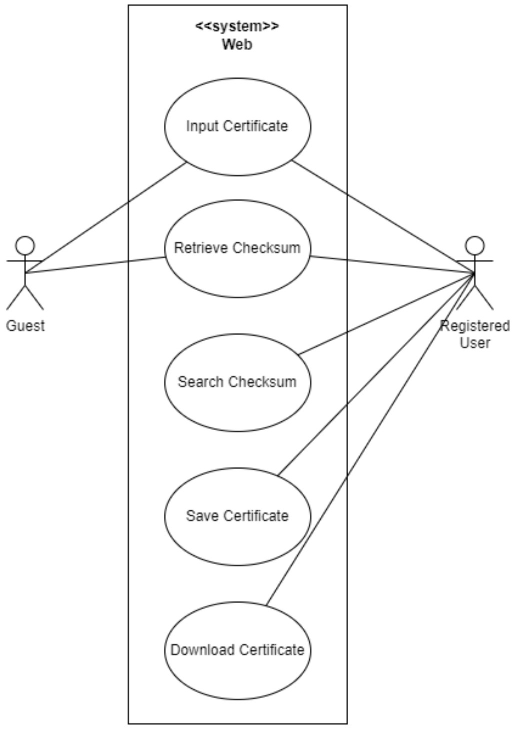
2.4. Desain Sistem/UML

Dalam rancangan UML ini, terdapat 3 jenis diagram yang digunakan, yaitu diagram usecase, diagram activity, dan diagram class.

2.4.1. *Usecase Diagram*

Dalam website yang dikembangkan, fitur yang tersedia terdiri dari input sertifikat dan retrieve checksum yang dapat diakses oleh pengunjung atau guest.

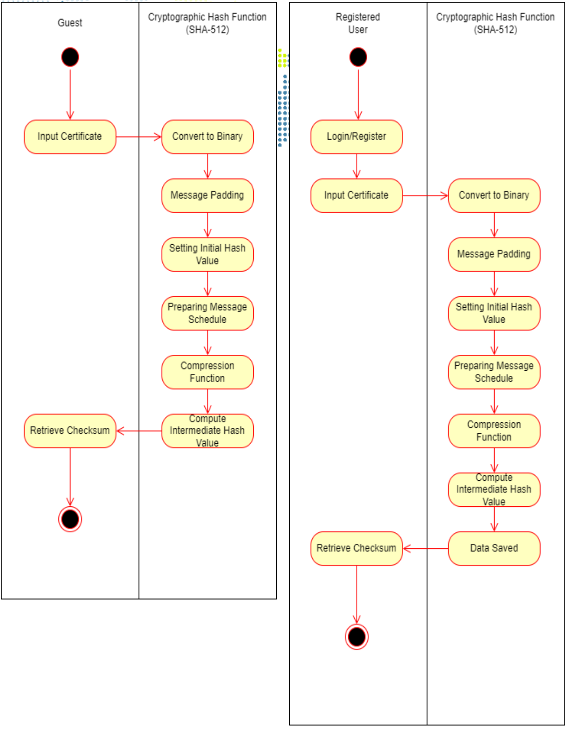
Sedangkan untuk pengguna terdaftar, selain fitur yang sudah tersedia untuk guest, pengguna juga dapat melakukan search checksum, menyimpan data sertifikat, dan mengunduh sertifikat. Dengan fitur-fitur ini, website yang dibangun dapat memberikan kemudahan bagi pengguna untuk memverifikasi keaslian sertifikat digital dengan cepat dan mudah. Gambaran spesifik mengenai Usecase Diagaram ini ditunjukan pada Gambar 1.



Gambar 1. *Usecase* Diagram

2.4.2. *Activity Diagram*

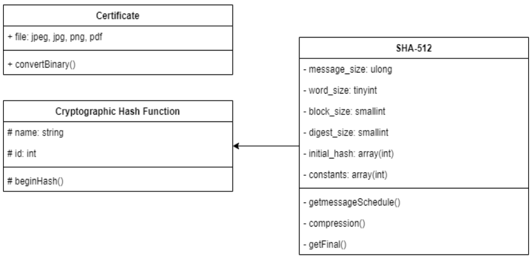
Tahap awal aktivitas di dalam website yang dikembangkan dimulai dari input sertifikat oleh guest. Setelah itu, data yang diinput akan otomatis dikonversi ke dalam bentuk binary, diikuti dengan proses padding dan parsing. Selanjutnya, dilakukan initial hash value tergantung pada bit algoritma yang dipilih. Tahap terakhir adalah kalkulasi hash yang terdiri dari preparing message schedule, compression function, dan compute intermediate hash value. Jika pengguna telah terdaftar, maka dapat melakukan login dan menyimpan data setelah compute intermediate hash value.



Gambar 2. Activity Diagram

2.4.3. *Class Diagram*

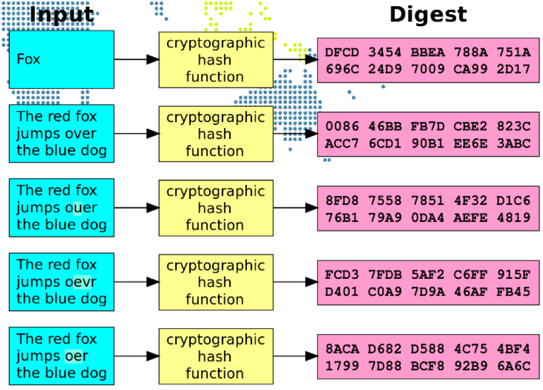
Dalam struktur logika class, terdapat CHF yang berperan sebagai parent class atau super class. CHF memiliki atribut dan metode yang memiliki tingkat akses protected. Selain itu, CHF juga memiliki subclass atau inheritance dengan nama SHA-512 yang memiliki penambahan atribut dan metode dengan tingkat akses *private*. CHF juga terkait dengan class bernama 'certificate' yang memiliki atribut dan metode dengan tingkat akses public. Gambaran spesifik mengenai Class Diagram ini ditunjukan pada Gambar 3.



Gambar 3. Class Diagram

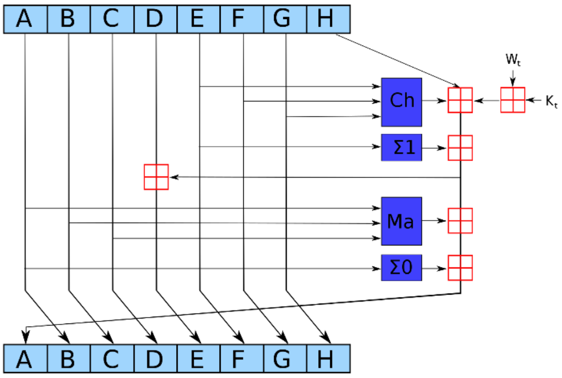
2.5. Arsitektur Sistem

Arsitektur sistem CHF menghasilkan sebuah keluaran yang disebut digest atau checksum. Avalanche effect terjadi saat perubahan kecil pada input menghasilkan perubahan besar pada output. Hal ini terjadi karena sistem CHF dirancang untuk menghasilkan output yang sangat sensitif terhadap perubahan pada inputnya. ditunjukan pada Gambar 4.



Gambar 4. Cryptographic Hash Function Architecture

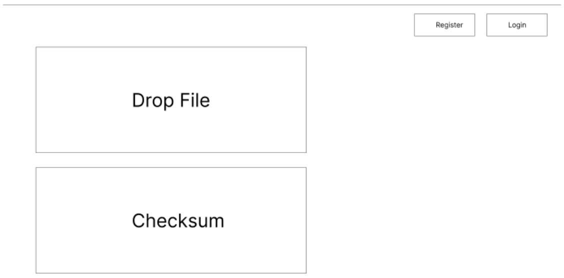
Dan berikut adalah sebuah gambaran atau ilustrasi dari compression function yang dimiliki oleh SHA-512 setelah melalui proses preparing message schedule, ditunjukan pada Gambar 5.



Gambar 5. SHA-2 Family Compression Function

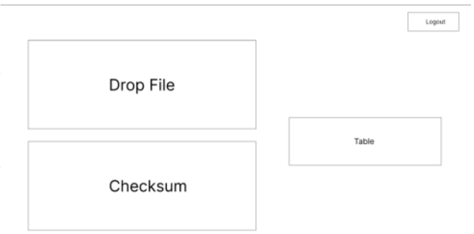
2.6. *Wireframe*

Untuk merancang dan membangun sebuah website, diperlukan perencanaan awal dalam bentuk wireframe. Berikut ini terdapat tiga gambar wireframe yang merinci desain website yang dibangun, yaitu Gambar 6, Gambar 7, dan Gambar 8.



Gambar 6. Landing Page Wireframe

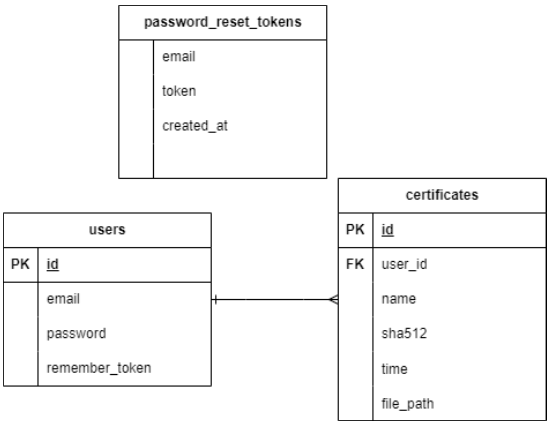
Gambar 7. Register Login Wireframe



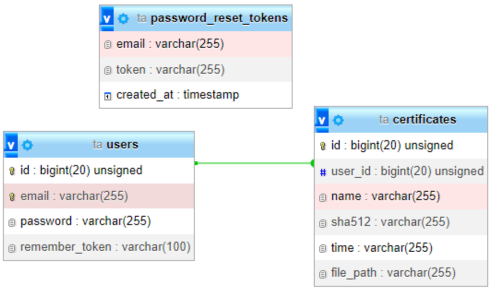
Gambar 8. Signed In Wireframe

2.7. Desain Database

Dalam pembuatan website, digunakan desain database yang terdiri dari tiga tabel yaitu users, password resets, dan certificates. Tabel users memiliki relasi one-to-many terhadap tabel certificates karena setiap pengguna dapat memiliki beberapa sertifikat. Adapun pada tabel password resets, Laravel secara default mengasumsikan bahwa tabel ini memiliki relasi dengan tabel users melalui kolom email. Meskipun hubungan antara kedua tabel tersebut tidak dibuat sebagai foreign key pada skema database, hubungan ini tetap dapat dibuat melalui kode dalam framework Laravel yang menghubungkan keduanya melalui kolom email. Berikut merupakan gambar desain database, ditunjukan pada Gambar 9 dan Gambar 10.



Gambar 9. Database Design



Gambar 10. Database Design in PHPMyAdmin

3. HASIL DAN PEMBAHASAN

3.1. Tujuan pengujian

Tujuan dari pengujian pada rancang bangun website ini adalah untuk memastikan bahwa website berjalan sesuai dengan ekspektasi dan untuk mengidentifikasi kelemahan yang mungkin ada pada website yang dibangun. Terdapat dua jenis pengujian yang dilakukan, yaitu alpha testing dan beta testing. Pengujian alpha dilakukan oleh tim pengembang secara internal untuk menguji fitur-fitur dan memastikan bahwa website dapat berjalan dengan baik. Sedangkan pengujian beta dilakukan oleh pengguna yang dipilih secara acak untuk menguji website dan memberikan umpan balik untuk perbaikan lebih lanjut.

a) Alpha Testing

Alpha testing adalah jenis pengujian yang dilakukan oleh pengembang atau tim internal yang terlibat dalam pengembangan aplikasi. Tujuan dari pengujian ini adalah untuk menemukan bug atau masalah lainnya sebelum aplikasi atau website dirilis ke publik. Alpha testing dalam website yang telah dibangun terdiri dari black-box testing. Berikut merupakan rincian pengujian dari black-box testing, ditunjukan pada Tabel 1.

Tabel 1. Black-Box Testing Table

No.	Pengujian	Ekspetasi	Dijalankan dalam *localhost*	Dijalankan dalam *web hosting*	Status
1	Membuka *website* dengan *URL*	Tampilan awal terlihat tanpa *error*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
2	Masuk ke halaman utama tanpa registrasi	Tampilan halaman utama *header* terlihat tombol *login* dan *register*. Ada 2 kotak besar dan tombol *remove* dalam *body*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
3	*Register* dan *login*	Pengguna dapat registrasi dan masuk sebagai pengguna terautentikasi	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
4	Fitur *forgot password* beserta *send reset link*	Pengguna dapat mengirimkan permintaan *password reset* ke email yang dimasukkan dan menerima *reset link* ke email tersebut meskipun masuk folder *spam*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
5	Masuk ke halaman utama dengan registrasi	Tombol *login* dan *register* di *header* berubah menjadi kolom *search* dan *logout.* Pada *body*, muncul tombol *save* dan tabel penyimpanan sertifikat	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
6	*Hashing* sertifikat	Pengguna dapat melakukan *drag and drop* sertifikat maupun *click to browse* pada kotak garis putus – putus. Hasil akan muncul di kotak dengan *placeholder SHA-512 Checksum*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*

No.	**Pengujian**	**Ekspetasi**	**Dijalankan dalam *localhost***	**Dijalankan dalam *web hosting***	**Status**
7	Format ekstensi sertifikat dan *file signature*	Sertifikat yang dapat dimasukkan berformat ekstensi *.jpg/jpeg*, *.png*, atau *.pdf* serta benar – benar murni *.jpg/jpeg*, *.png*, atau *.pdf*. Sistem juga dapat mendeteksi jika sertifikat yang dimasukkan bukan ketiga format yang disebutkan sebelumnya berdasarkan *file signature* atau *magic number*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
8	Tombol *remove*	Berhasil melakukan *clear* pada *input* maupun *output*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
9	Tombol *save*	Berhasil menyimpan data sertifikat beserta *file* nya berdasarkan *id* pengguna yang bersangkutan dan tampil dalam tabel	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
10	Kolom *search*	Pengguna dapat melakukan *live searching* berdasarkan nilai *checksum* dan muncul dalam tabel. *Keyword* yang diketik dalam pencarian harus sama dengan nilai *checksum* secara keseluruhan	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
11	Mengunduh sertifikat	Pengguna dapat mengunduh sertifikat baik sertifikat yang pernah disimpan maupun sertifikat yang muncul dalam hasil pencarian	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*
12	*Logout*	Pengguna dapat melakukan *logout*	Sesuai dengan ekspetasi	Sesuai dengan ekspetasi	*Passed*

b) Beta testing

Beta testing adalah metode pengujian yang melibatkan pengguna di luar tim pengembang, yang biasanya terdiri dari calon pengguna yang sebenarnya. Pengujian ini dilakukan setelah alpha testing selesai dan aplikasi/website hampir siap untuk dirilis atau dideploy. Tujuan dari pengujian beta adalah untuk mengevaluasi performa aplikasi/website tersebut dalam lingkungan nyata dan memperoleh umpan balik dari pengguna yang dapat membantu untuk meningkatkan aplikasi/website dan memperbaiki kekurangan yang ditemukan.

Pengujian beta test terhadap rancang bangun ini dilakukan dengan mengisi kuesioner yang disebarkan secara online melalui Google Form. Berikut merupakan rincian pertanyaan tersebut:

1) Pertanyaan dengan menggunakan Likert Scale 1 – 4 (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju) untuk mengetahui jika rancang bangun berjalan dengan baik di perangkat calon pengguna.

2) Pertanyaan dengan menggunakan Likert Scale 1 – 4 (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju) untuk mengetahui jika petunjuk dan arahan di tampilan rancang bangun sangat membantu dan mudah untuk dipahami.
3) Pertanyaan dengan menggunakan Likert Scale 1 – 4 (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju) untuk mengetahui jika calon pengguna merasa sangat terbantu untuk pengecekan keaslian sertifikat digital dengan adanya rancang bangun ini.
4) Pertanyaan dengan menggunakan Likert Scale 1 – 4 (Sangat Tidak Setuju, Tidak Setuju, Setuju, Sangat Setuju) untuk mengetahui jika penggunaan rancang bangun ini sangat efektif untuk melakukan pengecekan keaslian sertifikat digital.

Hasil jawaban dari pertanyaan tersebut, responden sangat setuju terhadap rancang bangun dapat bekerja dengan baik di perangkat responden, sangat setuju terhadap petunjuk dan arahan di tampilan rancang bangun sangat membantu dan mudah untuk dipahami, sangat setuju dan setuju terhadap terbantunya untuk pengecekan keaslian sertifikat digital dengan adanya rancang bangun ini, dan sangat setuju dan setuju bahwa penggunaan rancang bangun ini sangat efektif untuk melakukan pengecekan keaslian sertifikat digital.

3.2. Pembahasan

Dari masukan dan saran yang diterima, dapat disimpulkan bahwa verifikasi pada rancang bangun ini kurang efektif karena pengguna publik dapat mendaftar akun dan mengunggah hash sertifikat mandiri yang mungkin dibuat oleh mereka sendiri, bukan dari panitia penyelenggara acara yang terbentuk dalam lembaga atau organisasi. Selain itu, desain atau tampilan website dapat diperbaiki sedikit.

4. SIMPULAN

Berdasarkan hasil analisis yang didapat, dapat disimpulkan bahwa dalam rangka memberikan solusi atas masalah pemalsuan sertifikat digital yang tidak memiliki tanda keaslian seperti nomor seri, barcode, link, atau QR Scanner, sebuah rancangan aplikasi verifikasi keaslian e-sertifikat dibuat dengan menggunakan algoritma SHA-512. Untuk membangun aplikasi tersebut, penulis memanfaatkan library crypto-js sehingga dapat dikembangkan sebagai website yang membantu panitia pembuat sertifikat digital dalam verifikasi sertifikat peserta acara. Aplikasi tersebut memiliki fitur-fitur yang mudah digunakan, seperti registrasi, login, dan reset password yang sederhana serta desain yang menarik. Pengguna yang belum terdaftar hanya perlu melakukan drag and drop atau klik untuk mencari file sertifikat, sedangkan pengguna yang sudah terdaftar memiliki tombol save yang muncul di bawah tombol remove untuk menyimpan data ke dalam database. Aplikasi berbasis website ini juga dapat diakses dari perangkat gawai dan responsif untuk penggunaan yang lebih fleksibel.

Kemudian, berdasarkan hasil pengujian beta testing, dapat disimpulkan bahwa aplikasi berbasis website yang telah dibangun dapat membantu panitia pembuat sertifikat digital dalam memverifikasi keaslian sertifikat peserta, meskipun terdapat kekurangan pada fitur register yang terbuka untuk semua pengguna. Sebaiknya fitur register dibatasi hanya untuk administrasi pembuat sertifikat digital dan tidak dapat diakses oleh pengguna umum, guna mencegah penyalahgunaan dalam pembuatan dan pengunggahan data hash sertifikat secara mandiri.

DAFTAR PUSTAKA

[1] S. Chao, Z. Chen, and X. Sun, "Anti-Counterfeit Authentication System of Printed Information Based on A Logic Signing Technique," Atlantis Press | Atlantis Press Open Access Publisher Scientific Technical Medical Proceedings Journals Books, Oct. 2007. [Online]. Available: https://www.atlantis-press.com/article/1429.pdf.

[2] O. Ghazali and O. S. Saleh, "A Graduation Certificate Verification Model via Utilization of the Blockchain Technology," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), Sep. 26, 2018. [Online]. Available: https://jtec.utem.edu.my/jtec/article/view/4707/3640.

[3] C. R. Dougherty, "CERT/CC vulnerability note VU#836068," CERT Vulnerability Notes Database, Dec. 31, 2008. [Online]. Available: https://www.kb.cert.org/vuls/id/836068.

[4] B. Schneier, "Cryptanalysis of SHA-1," Schneier on Security, Feb. 18, 2005. [Online]. Available: https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html.

[5] NIST, "NIST.gov - Computer security division - Computer security resource center," Wayback Machine, Apr. 12, 2011. [Online]. Available: https://web.archive.org/web/20110625054822/csrc.NIST.gov/groups/ST/toolkit/secure_Hashing.html.

[6] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," [Online]. Available: https://shattered.io/static/shattered.pdf.

[7] A. Argani and W. Taraka, "Pemanfaatan Teknologi Blockchain Untuk Mengoptimalkan Keamanan Sertifikat Pada Perguruan Tinggi," Google Books, Jun. 2020. [Online]. Available: https://books.google.co.id/books?hl=en&lr=&id=cDILEAAAQBAJ&oi=fnd&pg=PA10&dq=sertifikat+palsu&ots=a0GNYjgrzy&sig=IdZpKZYgJEzII0Xr9B6nOdjSoYY&redir_esc=y#v=onepage&q=sertifikat%20palsu&f=false.

[8] U. Rahardja, E. P. Harahap, and G. Fresandy, "Penerapan Sistem Autentikasi Sertifikat Sebagai Pengambil Keputusan Validasi Sertifikat Pada Perguruan Tinggi," iLearning Journal Center (iJC), Aug. 2017. [Online]. Available: https://ijc.ilearning.co/index.php/TMJ/article/view/312/25.

[9] A. H. Lone and R. Naaz, "Forgery Protection Of Academic Certificates Through Integrity Preservation At Scale Using Ethereum Smart Contract," Scalable Computing: Practice and Experience, Dec. 2020. [Online]. Available: https://scpe.org/index.php/scpe/article/view/1806/672