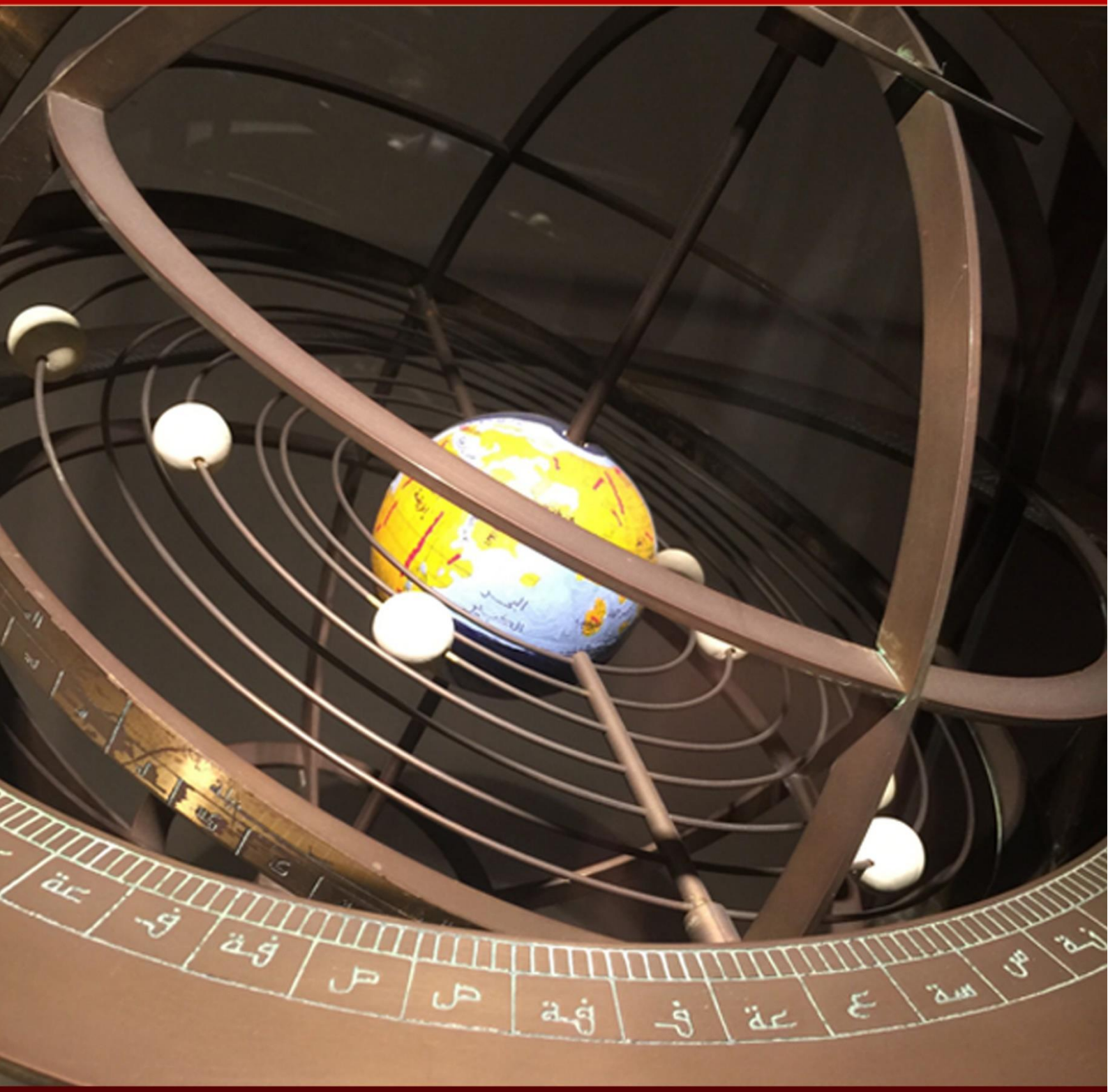
Vol. 4, No. 1, Januari-Juni 2019
e-ISSN : 2528-5718

JISTech

(Journal of Islamic Science and Technology)



Diterbitkan Oleh :
Fakultas Sains Dan Teknologi
UIN Sumatera Utara Medan

HASHING ARGON2 UNTUK KEAMANAN PASSWORD PADA SISTEM BERBASIS WEB MENGGUNAKAN PHP

Suendri
Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
Email: suendri@gmail.com

Abstrak : *Username* dan *Password* merupakan bentuk autentikasi utama yang biasa digunakan pada sistem berbasis web. Melalui kombinasi *username* dan *password* yang benar, pengguna berhak untuk menggunakan sistem sesuai tingkat akses yang diberikan. *Username* seringkali mudah untuk diketahui, seperti *email*, nomor *handphone*, nomor penduduk dan sejenisnya. Namun *password*, bersifat rahasia dan tidak boleh diketahui oleh orang lain. *Password* disimpan dalam *server* dengan bentuk karakter yang tidak bisa dibaca oleh manusia atau disebut dengan *Chipertext* melalui proses *Encryption*. Salah satu teknik paling aman yang telah digunakan adalah *Hash* atau *One Way Encryption*. Enkripsi ini menjadikan karakter terenkripsi dengan ukuran yang tetap dan tidak bisa dikembalikan. Namun seiring dengan kemajuan teknologi, beberapa algoritma *hashing* sudah bisa dipecahkan, sehingga menimbulkan celah keamaan pada sistem yang menggunakan *password*. *Hash* Argon2 merupakan pemenang *Password Hashing Competition* (PHC) tahun 2015. *Hashing* Argon2 mengutamakan kecepatan waktu, penggunaan memori dan tingkat parallelisme. Penelitian ini bertujuan untuk mengetahui proses enkripsi, tingkat kekuatan dan keamanan data pada *hash* Argon2 sehingga dapat digunakan untuk keamanan *password* pada sistem berbasis web.

Kata kunci : Hashing, Argon2, Password, Enkripsi, Keamanan

Abstract : *Username and Password are the main forms of authentication commonly used on web-based systems. Through the correct username and password combination, the user has the right to use the system according to the level of access provided. Username is often easy to know, such as e-mail, mobile phone number, identity card number and etc. However, passwords are confidential and may not be known by others. The password is stored on a server with characters that cannot be read by humans or called Chipertext through the Encryption process. One of the safest techniques that has been used is Hash or One Way Encryption. This encryption makes encrypted characters with a fixed size and cannot be returned. But along with technological advancements, some hashing algorithms can be solved, thus creating a security gap on systems that use passwords. Hash Argon2 is the winner of the 2015 Password Hashing Competition (PHC). Argon2 Hashing prioritizes time cost, memory cost and parallelism. This study aims to determine the encryption process, the*

level of strength and security of data on Argon2 hashes so that it can be used for password security on web-based systems

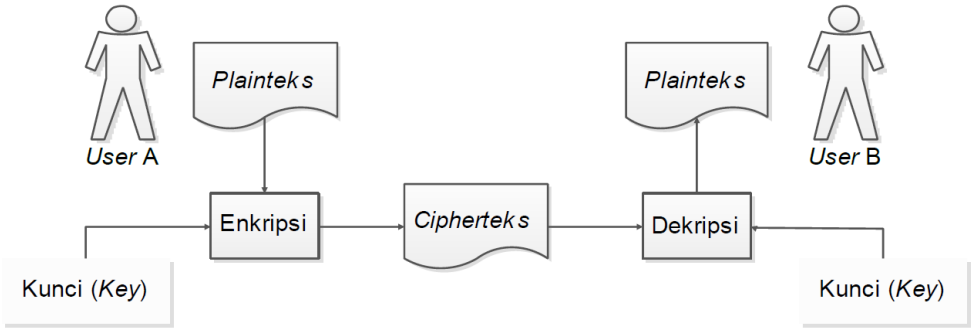
Keywords : *Hashing, Argon2, Password, Encryption, Security*.

Pendahuluan

Perkembangan Teknologi Informasi telah banyak mempengaruhi berbagai aspek kehidupan umat manusia dalam menjalankan aktivitas sehari-hari. Penggunaan komputer telah jauh mengalami kemajuan dari sekedar teknologi alat hitung hingga pengambilan keputusan (Suendri, 2017). Pada dewasa ini data dan informasi dibutuhkan di lembaga pendidikan maupun kemajuan sebuah instansi, maka dari itu harus juga disertai dengan keamanan data informasi tersebut (Komarudin, 2013). Sistem informasi menguntungkan dan dapat meningkatkan kinerja dari semua komponen organisasi, tetapi dari sisi yang lain terutama dari sisi keamanan sistem informasi yang berbasis web sangat rawan untuk di sadap oleh pihak yang tidak berkepentingan (Santoso, 2013). Sistem informasi yang baik adalah sistem informasi yang dapat dinilai tingkat keamanannya, sehingga mampu memberikan kenyamanan bagi pengguna (Umar *et al*, 2019). Penerapan sistem keamanan informasi bertujuan untuk mengatasi segala masalah dan kendala baik secara teknis maupun secara non-teknis yang dapat berpengaruh dalam kinerja sistem (Rosmiati *et al*, 2016). Penerapan keamanan informasi bertujuan untuk mengatasi masalah dan kendala baik secara teknis maupun non-teknis seperti faktor ketersediaan (*availability*), kerahasiaan (*confidentiality*), dan kesatuan (*integrity*) sehingga dapat dinilai tingkat keamanan informasinya (Riadi, 2016). Banyak instansi dan perusahaan berani membayar harga mahal hanya untuk keamanan. Hal ini dikarenakan kekhawatiran pada kejahatan, penyusupan dan penyadapan informasi (Aritonang, 2016). Dalam aplikasi web dibutuhkan mekanisme yang dapat melindungi data dari pengguna yang tidak berhak. Mekanisme ini dapat diimplementasikan dalam bentuk sebuah proses login yang biasanya terdiri dari tiga buah tahapan yaitu identifikasi, otentikasi dan otorisasi (Khairina, 2011).

Kriptografi

Kriptografi merupakan ilmu dan seni dalam memproteksikan informasi dengan mengubahnya ke dalam bentuk himpunan karakter acak yang tidak dapat dibaca (Ariyus, 2008). Terminology kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain (Munir, 2006). Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti (*plaintext*) menjadi sebuah kode yang tidak bisa dimengerti (*ciphertext*). Sedangkan proses kebalikannya untuk mengubah ciphertext menjadi plaintext disebut dekripsi (Sanger, 2015).



Gambar 1 Skema Kriptografi (Sanger, 2015)

Menurut Budi Rahardjo (2017), secara umum ada tiga komponen utama dari kriptografi, yaitu *plain text*, *cipher text*, dan algoritma serta kunci yang digunakan.

1. *Plain text* adalah data (teks, pesan, message) asli yang belum diproses. Meskipun disebut *plain text*, sesungguhnya data asli tidak harus berupa teks (ASCII). *Plain text* dapat juga berupa berkas *biner*.
2. *Ciphertext* adalah data yang dihasilkan dari proses enkripsi. *Ciphertext* dapat berbentuk berkas biner atau ASCII. Perlu diasumsikan bahwa penyerang kemungkinan dapat mengakses *ciphered text*.
3. Algoritma dan kunci merupakan *black box* yang memproses *plain text* menjadi *cipher text*. Algoritma diasumsikan diketahui oleh penyerang, tetapi kunci tidak diketahui.

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* (Yahya, 2018). Fungsi kriptografi *hash* adalah sebuah fungsi matematika yang digunakan mencerna pesan, artinya dibutuhkan sebuah pesan sebagai input dan menghasilkan output sebagai nilai hash (Li *et al*, 2013) ukuran dari nilai output hash tergantung dari algoritma yang digunakan seperti 64 bit, 128 bit dan 256 bit. *Hash* merupakan enkripsi satu arah, satu arah berarti tidak mempunyai fungsi untuk melakukan pengembalian nilai yang sudah di enkripsi (Sanger, 2015)

Beberapa Algoritma *hashing* yang telah berkembang adalah sebagai berikut :

1. MD4(*Message-Digest algortihm 4*)
2. MD5 (*Message-Digest algortihm 5*)
3. MD5 (*$pass.$salt*)
4. MD5 (*$salt.$pass*)
5. SHA-1 (*Secure Hash Algorithm*)
6. SHA-256 (*Secure Hash Algorithm*)
7. SHA-512 (*Secure Hash Algorithm*)
8. Base64

Argon2

Algoritma ini dirancang oleh Alex Biryukov, Daniel Dinu, dan Dmitry Khovratovich dari University of Luxembourg dibawah lisensi *Creative Commons CCo* atau *Apache License* 2.0. *Hash* Argon2 mengutamakan kecepatan waktu, penggunaan memori dan tingkat parallelisme, algoritma ini merupakan pemenang *Password Hashing Competition* (PHC) tahun 2015 yang diadakan oleh password-hashing.net. Algoritma Argon2 dibagi menjadi tiga jenis yaitu:

1. Argon2d

   Argon2d memaksimalkan pertahanan terhadap serangan GPU. Jenis ini mengakses memori yang bergantung pada password, yang mengurangi kemungkinan serangan *Time Memory Trade Off* (TMTO).
2. Argon2i

   Argon2i memaksimalkan pertahanan terhadap saluran samping. Jenis ini mengakses memori secara bebas.
3. Argon2id

   Argon2id merupakan gabungan dari versi Argon2i dan Argon2d. Jenis ini mengikuti pendekatan Argon2i untuk melewati memori pertama dan Argon2d untuk selanjutnya. Jenis ini merupakan versi yang disarankan kecuali untuk keperluan tertentu yang hanya menggunakan Argon2i saja atau Argon2d saja.

Metode Penelitian

Metode penelitian yang digunakan adalah *Library Research*. Metode ini dilakukan dengan cara mengumpulkan semua teori-teori yang berhubungan dengan *Enkripsi* dan *Hashing*, mengumpulkan penelitian-penelitian sebelumnya yang berhubungan dengan objek yang diteliti dan menganalisa serta membahas contoh implementasi *hash* Argon2 pada sistem berbasis web yang menggunakan bahasa pemrograman PHP.

Hasil dan Pembahasan

Hash Argon2 dibagi menjadi tiga jenis yaitu Argon2i, Argon2d dan Argon2id. Argon2i menggunakan akses memori yang independen sehingga sangat direkomendasikan untuk *hashing password* dan *password-based key derivation* (penurunan kunci berbasis kata sandi).

Argon2 memiliki dua jenis input yaitu: input primer dan input sekunder atau parameter-parameter. Input primer dilambangkan dengan *message P* dan *nonce S,* masing-masing merupakan *Password* dan *Salt* yang digunakan untuk *hashing* (Biryukov *et al*, 2017). *Nonce* adalah angka acak yang hanya digunakan sekali saja dalam proses enkripsi, sedangkan

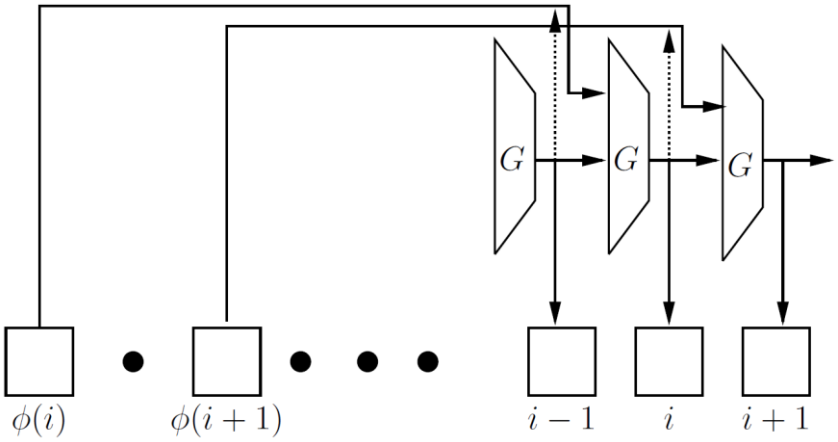
salt adalah bit acak yang digunakan sebagai salah satu masukan untuk kunci fungsi turunan dalam enkripsi. Alex Biryukov (Biryukov *et al*, 2017) menjelaskan secara rinci tentang Argon2 sebagai berikut :

1. Message P mempunyai panjang dari 0 hingga $2^{32} - 1$ byte.
2. Nonce S mempunyai panjang dari 8 hingga $2^{32} - 1$ byte (16 byte merupakan rekomendasi untuk hash password

Sedangkan untuk input sekunder mempunyai batasan sebagai berikut :

1. Tingkat parallelisme ***p*** menentukan berapa banyak rantai komputasi bebas yang bisa dijalankan. Nilai ini berjenis integer antara 1 hingga $2^{24} - 1$;
2. Panjang Tag ***T*** berjenis integer dari 4 hingga $2^{32} - 1$;
3. Ukuran memori ***m*** merupakan nilai integer dalam kilobyte antara 8p hingga $2^{32} - 1$. Nilai aktual dari blok-blok memori adalam ***m***', dimana **m** dibulatkan ke kelipatan 4***p*** terdekat;
4. Angka perulangan ***t*** (digunakan untuk set waktu operasi secara bebas dari ukuran memori) nilai ini berjenis integer 1 hingga $2^{32} - 1$;
5. Angka versi ***v*** adalah satu byte 0*x*13;
6. Nilai rahasia ***K*** (berfungsi sebagai kunci jika dibutuhkan) mempunyai panjang 0 hingga $2^{32} - 1$ byte;
7. Data ***X*** mempunyai panjang 0 hingga 232 – 1 byte;
8. Tipe ***y*** dari Argon2: 0 untuk Argon2d, 1 untuk Argon2i dan 2 untuk Argon2id.

Argon2 menggunakan fungsi kompresi internal G dengan dua input 1024 byte dan satu ouput 1024 byte serta fungsi hash H internal. Hash H adalah fungsi hash Blake2b sedangkan G berdasarkan permutasi internal. Mode operasi Argon2 lebih sederhana jika tanpa menggunakan parallelisme. Mode operasinya bisa dilihat pada gambar 2 berikut ini.



Gambar 2 Argon2 tanpa parallelisme (Biryukov, 2017)

Argon2 mengikuti konsep ekstrak kemudian kembangkan. Langkah awal ektrak entropi dari pesan dan *nonce* dengan cara hashing pesan tersebut kemudian tambahkan seluruh parameter-parameter kedalam input tersebut.

$$H_0 = H(p, \tau, m, t, v, y, \langle P \rangle, P, \langle S \rangle, S, \langle K \rangle, K, \langle X \rangle, X)$$

H_0 bernilai 64-byte dan parameter-parameternya bernilai integer 32-bit. Argon2 kemudian mengisi memory dengan $\boldsymbol{m'} = \left\lfloor \frac{\boldsymbol{m}}{\boldsymbol{4p}} \right\rfloor . \boldsymbol{4p}$ 1024 block. Untuk parallelisme yang lebih baik dengan *thread p,* pengaturan memory dengan matrix $B[i][j]$ dari block dengan p dan $q = m'/p$. Ini menunjukkan block dibuat melewati t dengan $B^t[i][j], t > 0$. Block dihitung sebagai berikut :

$$B^1[i][0] = H'(H_0 || \underbrace{0}_{4 \text{ bytes}} || \underbrace{i}_{4 \text{ bytes}}), \quad 0 \leq i < p;$$

$$B^1[i][1] = H'(H_0 || \underbrace{1}_{4 \text{ bytes}} || \underbrace{i}_{4 \text{ bytes}}), \quad 0 \leq i < p;$$

$$B^1[i][j] = G(B^1[i][j-1], B^1[i'][j']), \quad 0 \leq i < p, \ 2 \leq j < q.$$

Untuk implementasi pada bahasa pemrograman PHP, *Hashing* Argon2 sudah bisa dijalankan mulai dari versi 7.2.0

```
PASSWORD_ARGON2I
```

Dengan tiga konstanta tambahan

```
PASSWORD_ARGON2_DEFAULT_MEMORY_COST
PASSWORD_ARGON2_DEFAULT_TIME_COST
PASSWORD_ARGON2_DEFAULT_THREADS
```

Nilai dasar dari masing-masing konstanta adalah sebagai berikut :

```
memory_cost = 1024 KiB
time_cost = 2
threads = 2
```

Contoh penggunaan

```
password_hash('Rahasia2019', PASSWORD_ARGON2I);
password_hash('Rahasia2019', PASSWORD_ARGON2I,
['memory_cost' => 1<<17, 'time_cost' => 4,
'threads' => 2]);
```

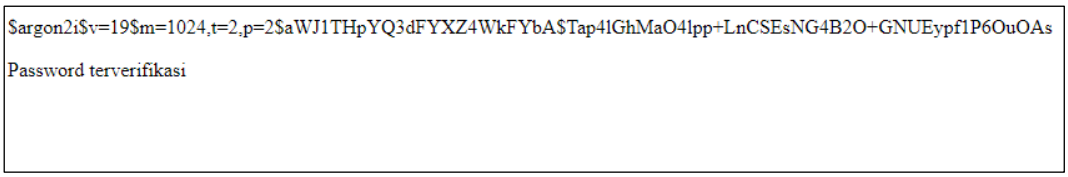
Hash dari **Rahasia2019** akan menghasilkan string acak

$argon2i$v=19$m=1024,t=2,p=2$YjljMW1FSnp5cC84Q2hlZA$Y1oqlAKQv6UioxjVNWumTZWwzZNulAAwa6gUYJxK4OQ

```php
<?php

$user_password = password_hash('Rahasia2019', PASSWORD_ARGON2I);

$stored_hash = '$argon2i$v=19$m=1024,t=2,p=2$YjljMW1FSnp5cC84Q2hlZA$Y1oqlAKQv6UioxjVNWumTZWwzZNulAAwa6gUYJxK4OQ';

echo "<p>" . $user_password . "</p>";

if(password_verify('Rahasia2019', $stored_hash)) {
    echo "<p> Password terverifikasi</p>";
}
```

Gambar 3 Contoh *Hash* Argon2i

Gambar 3 diatas merupakan contoh *script* yang dijalan menggunakan XAMPP versi 7.3.6. Perintah *password_verify* akan membandingkan *string* acak menggunakan Argon2i. Jika karakter yang dimasukkan terverifikasi maka, kondisi bernilai *True*



Gambar 4 Hasil *Hash* menggunakan Argon2i

Gambar 4 merupakan bentuk tampilan *hash* dari Argon2i. Setiap halaman dieksekusi atau *load* ulang, *string* hasil *hash* akan diacak kembali

Kesimpulan

Hash Argon2 merupakan sebuah teknik *One Way Encryption* yang memiliki fitur pengaturan memori, kecepatan waktu dan tingkat parallelisme. Hal ini menjadikan *Hash* Argon2 menjadi sebuah pilihan yang tepat untuk enkripsi, khususnya sistem berbasis web menggunakan PHP agar sistem lebih aman dan terhindar dari penggunaan sistem yang tidak diinginkan.

Daftar Pustaka

Suendri, S. (2017). Implementasi Algoritma Linear Congruentials Generator Untuk Menentukan Posisi Jabatan Kepanitiaan. Query: Jurnal Sistem Informasi, 01(02), 15–22. Retrieved from http://jurnal.uinsu.ac.id/index.php/query/article/view/1043

Komarudin, Asep Ririh Riswaya. (2013). Sistem Keamanan Web Dengan Menggunakan Kriptografi Message Digest 5/Md5 Pada Koperasi

Mitra Sejahtera Bandung. Jurnal Computech & Bisnis, Vol. 7, No. 1, Juni 2013, 30-41 ISSN 2442-4943. http://jurnal.stmik-mi.ac.id/index.php/jcb/article/download/99/104 diakses 27 juni 2019.

Kartika Imam Santoso, Eko Sediyono, Suhartono. (2013). Studi Pengamanan Login Pada Sistem Informasi Akademik Menggunakan Otentifikasi One Time Password Berbasisis SMS dengan Hash MD5. Jurnal Sistem Informasi Bisnis 01 (2013). https://ejournal.undip.ac.id/index.php/jsinbis/article/download/2/Kartika%20Imam%20Santoso diakses 27 juni 2019

Rusydi Umar, Imam Riadi, Eko Handoyo. (2019). Analisis Keamanan Sistem Informasi Berdasarkan Framework COBIT 5 Menggunakan Capability Maturity Model Integration (CMMI). Jurnal Sistem Informasi Bisnis 01 (2019). DOI : 10.21456/vol9iss1pp47-54 https://ejournal.undip.ac.id/index.php/jsinbis/article/view/20734 diakses 27 juni 2019

Rosmiati, Riadi, I., Prayudi, Y. (2016). A Maturity level framework for measurement of information security performance. International Journal of Computer Applications, 141(8), 975–8887. https://www.ijcaonline.org/archives/volume141/number8/rosmiati-2016-ijca-907930.pdf diakses 27 juni 2019

Rosmiati, Imam Riadi. (2016). Analisis keamanan informasi berdasarkan kebutuhan teknikal dan operasional mengkombinasikan standar ISO 27001 : 2005 dengan maturity level (studi kasus kantor biro teknologi informasi PT . XYZ). Seminar Nasional Teknologi Informasi Dan Multimedia 2016, 6(6), 6–7. https://www.ojs.amikom.ac.id/index.php/semnasteknomedia/article/viewFile/1130/1086 diakses 27 juni 2019.

Aritonang, Mendarissan. (2016). Perancangan Aplikasi Keamanan Untuk Membatasi Hak Akses User Berbasis Windows. Jurnal METHODIKA, Vol. 2 No. 1 Maret 2016 ISSN : 2442-7861. http://methodika.net/index.php/jurnalmethodika/article/viewFile/19/20 diakses 27 juni 2019.

Khairina, Dyna Marisa. (2011). Analisis Keamanan Sistem Login. Jurnal Informatika Mulawarman. Vol. 6 No. 2 Juni 2011. http://e-journals.unmul.ac.id/index.php/JIM/article/view/74 diakses 27 Juni 2019

Ariyus, D. (2008). Pengantar Ilmu Kriptografi: Teori, Analisa, dan Implementasi. Yogyakarta: Andi.

Munir, R. (2006). Kriptografi. Bandung: Informatika

Sanger, Junaidy B. (2015). Desain Dan Implementasi Mekanisme Tanda Tangan Dijital Dalam Pertukaran Data Dengan Hash Md5 Dan Enkripsi/Dekripsi Menggunakan Algoritma Rsa. Jurnal Lasallian Vol. 12 No. 2 September 2015. ttps://osf.io/preprints/inarxiv/6nj5k/

Rahardjo Budi. (2017). Keamanan Informasi. Bandung: PT Insan Infonesia.

Nanda Imani Yahya, Safrina Amini. (2018). Pengimplementasian One Time Password Dan Notifikasi Email Menggunakan Fungsi Hash Sha-512 Berbasis Web Pada Smk Cyber Media. SKANIKA VOLUME 1 NO. 2 MEI 2018. http://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/download/285/199/ 27 juni 2019

P. Li, Y. Sui, and H. Yang. (2010). “The parallel computation in one-way hash function designing,” in Proceedings of International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE), vol. 1, pp. 189–192.

Biryukov, Alex et al. (2017). Argon2: the memory-hard function for password hashing and other applications. Version 1.3 of Argon2: PHC release