ONE – TIME PASSWORD BERBASIS WAKTU DENGAN ALGORITMA *SUBSTITUTION AND PERMUTATION NETWORK* (SPN) DAN *BLOWFISH* SEBAGAI *METODE AUTENTIKASI* MEDIA SOSIAL

ONE – TIME-BASED PASSWORD WITH SUBSTITUTION AND PERMUTATION NETWORK (SPN) AND BLOWFISH ALGORITHM AS A SOCIAL MEDIA AUTHENTICATION METHOD

Dian Indriyani dan Karyati

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Yogyakarta, Yogyakarta 55281 Indonesia
*Email korespondensi: karyati@uny.ac.id

Submitted: 17 November 2022, Accepted: 11 April 2023

Abstrak

Kemudahan akses media sosial menyebabkan data yang ada dalam jaringan menjadi rentan terhadap penyerangan. Salah satu cara untuk meningkatkan keamanan data yaitu dengan melakukan *autentikasi* ketika akan login ke dalam sistem dengan menggunakan *One Time Password* (OTP) yang hanya berlaku untuk satu kali sesi login. Data pada OTP belum menjamin keamanan suatu sistem, sehingga diperlukan algoritma kriptografi untuk *meningkatkan* keamanannya. Algoritma kriptografi yang digunakan merupakan gabungan dari *Substitution and Permutation Network* (SPN) dan *Blowfish*. Algoritma SPN dipilih karena memenuhi prinsip penghamburan Shannon, sehingga jika salah satu bit teks asli diubah, maka hasil outputnya akan berubah total. Demikian pula jika terdapat perubahan pada salah satu bit kunci, maka akan merubah seluruh *key mixing*. Algoritma *Blowfish* dipilih karena adanya iterasinya yang panjang sehingga menyebabkan tingkat kesulitannya semakin tinggi, dan output yang dihasilkan terdiri dari karakter yang beragam. Output yang dihasilkan dari penggabungan algoritma OTP, SPN dan *Blowfish* memungkinkan *autentikasi* yang lebih bervariasi karakternya sehingga mampu memastikan bahwa yang terhubung adalah *client* yang berhak.

Kata kunci : *One Time Password* (OTP), *Substitution and Permutation Network* (SPN), *Blowfish*, *autentikasi*

Abstract

The ease of access to social media causes the data on the network to be vulnerable to attack. One way to improve data security is get authenticate when logging into the system using a One Time Password (OTP) which is only valid for one login session. The data on the OTP does not guarantee the security of a system, therefore a cryptographic algorithm is needed so that the level of security is better. The cryptographic algorithm *used is a combination of Substitution and Permutation Network (SPN) and Blowfish. The SPN algorithm was chosen because the algorithm meets the Shannon scattering principle, so that if one bit of the original text is changed, then the output will change completely, as well as if there is a change in one of the key bits, it will change the entire key mixing. While the Blowfish algorithm was chosen because of its long iteration, which causes the difficulty level to be higher, and the resulting output consists of various characters. Output from the combination of the OTP, SPN and Blowfish algorithms allows authentication with a more varied character so as to ensure who's connected that the rightful client.*

Keywords : *One Time Password* (OTP), *Substitution and Permutation Network* (SPN), *Blowfish*, *authentication*

Pendahuluan

Akses situs sosial media menjadikan masyarakat semakin mudah untuk menjangkau informasi dari belahan dunia, dengan kemudahan ini tentunya membawa pengaruh terhadap tingkat keamanan informasi data pribadi [6]. Keamanan data bukan hanya bergantung pada penyimpanan data, tetapi juga pada proses transfernya. Ketika masyarakat mengirimkan sebuah data, maka ada peluang untuk adanya penyerangan yang menyebabkan data menjadi keliru [4], sehingga diperlukan adanya langkah *autentikasi* untuk memastikan kebenaran pemilik data.

Autentikasi adalah suatu proses validasi pada saat hendak memasuki sebuah sistem web dimana data yang diinputkan akan diuji dengan data yang ada di sistem. Terdapat beberapa metode

autentikasi, antara lain: token, smartcard, kode acak, biometrik, dan sertifikat software [13]. Namun metode yang sering digunakan sebagai *autentikasi* pengguna sosial media adalah kode acak *One-Time Password* (OTP). OTP digunakan untuk menghindari terjadinya *replay attack*, yaitu bentuk serangan jaringan di mana transmisi data yang valid diulang atau ditunda secara curang. Terdapat dua jenis OTP yaitu *Hash-based Message Autentication Code* (HMAC) atau dalam istilah awam disebut *One Time Password berbasis HMAC* (HOTP) dan *One Time Password* berbasis waktu (TOTP). HOTP adalah OTP dengan faktor pergerakan dalam setiap kode didasarkan pada penghitung yang terdapat pada *autentikator* dan tidak memiliki batasan berbasis waktu. HOTP sering terjadi ketidakcocokan penghitung di klien dan server karena kata sandi yang dihasilkan oleh klien tidak dikirimkan ulang ke server, atau kata sandi dikirimkan oleh klien tetapi tidak mencapai server karena kegagalan jaringan. Selain itu, output yang dihasilkan oleh HOTP terlalu besar, yaitu 160 bit (berdasarkan perhitungan dengan HMAC-SHA-1), sehingga perlu adanya pemotongan bit ke nilai yang lebih kecil agar mudah digunakan untuk proses *autentikasi*. TOTP merupakan pengembangan dari HOTP namun didasarkan pada basis waktu yang telah ditentukan, sehingga kode akan kadaluarsa jika sudah melewati batas waktu tertentu, kemungkinan terjadi serangan sangat kecil [1], sehingga OTP berbasis waktu ini menjadi alternatif pilihan yang lebih baik dibandingkan dengan HOTP. Metode keamanan dengan menggunakan OTP telah banyak digunakan pada mobile banking, email, jaringan social, transaksi online, marketplaces, dan aplikasi akademik online [3], [5],[7],[8],[10],[11],[14] dan juga pada cloud [9],[15].

Suatu sistem kriptografi yang baik terletak pada kerahasiaan kuncinya. Algoritma kriptografi kunci publik sering digunakan sebagai metode *autentikasi*, dengan menyandikannya ke dalam bentuk yang bervariasi. Algoritma kriptografi publik yang akan digunakan sebagai pengenkripsi TOTP dalam penelitian ini adalah algoritma *Subtitution and Permutation Network* (SPN) dan *Blowfish*.

SPN adalah kriptografi simetris bertipe *block cipher* yang bersifat iterative, terdiri dari proses substitusi, permutasi, dan penjadwalan kunci (*key mixing*). Sistem dasar SPN dibentuk dari dua permutasi, yaitu π_s dan π_p [15]. SPN dipilih karena algoritma ini memenuhi prinsip penghamburan Shannon, yaitu membuat hubungan antara teks tersandi dan kunci simetris menjadi serumit mungkin. Jika salah satu bit teks asli diubah, maka hasil output dari algoritma ini akan berubah total. Demikian pula jika terdapat perubahan pada salah satu bit kunci, maka akan merubah seluruh *key mixing* [17].

Berdasarkan penelitian yang dilakukan oleh Adin Baskoro Pratomo diperoleh hasil penggabungan *one-time password* dan algoritma kriptografi kunci publik memungkinkan *autentikasi* yang lebih praktis namun kode yang dihasilkan karakternya belum beragam sehingga dibutuhkan algoritma pendamping lainnya, dan penelitian yang dilakukan oleh Ariel dikatakan bahwa keamanan OTP dengan tambahan algoritma *Blowfish* menghasilkan kode yang beragam dan algoritma *Blowfish* dapat digunakan bersamaan dengan algoritma lain untuk meningkatkan keragaman outputnya. *Blowfish* adalah algoritma kunci simetrik cipher blok dengan Panjang blok tetap sepanjang 64 bit, serta menerapkan Teknik kunci yang berukuran sebarang (memiliki ruang kunci yang besar dan panjangnya beragam), sehingga membuatnya tidak mudah diserang pada bagian kuncinya [16].

Berdasarkan latar belakang yang ada perlu dilakukan sebuah penelitian yang menggabungkan algoritma SPN dan *Blowfish* untuk mengenkripsikan kode OTP. Dengan penggabungan ketiga metode tersebut, diharapkan output yang dihasilkan memungkinkan *autentikasi* yang lebih praktis dan bervariasi karakternya sehingga mampu memastikan bahwa yang terhubung adalah *client* yang berhak.

Metode Penelitian

Pada penelitian ini, dibuat sebuah sistem aplikasi untuk mengaplikasikan OTP berbasis waktu dengan algoritma SPN dan *Blowfish* sebagai metode *autentikasi* media sosial. Proses validasi dalam sistem ini dilakukan ketika pengguna hendak melalukan proses login. Ketika proses login, pengguna akan memperoleh kode yang nantinya digunakan untuk memvalidasi kebenaran pemilik akun. Dengan memasukkan kode OTP yang benar, maka pengguna dapat masuk ke dalam sistem.

Perangkat lunak yang digunakan dalam proses pembuatan aplikasi ini antara lain :

a. XAMPP, digunakan sebagai localhost,
b. PHPMyAdmin dan MySQL, diperlukan untuk penyimpanan data (basis data),

c. Sublime text 3, digunakan untuk editing program dan penyusunan *script*.

1. Rancangan *autentikasi*

Algoritma *autentikasi* yang dibuat terdiri dari tiga bagian utama, yaitu sistem pembangkitan dengan OTP berbasis waktu, sistem penandatanganan dengan kunci publik menggunakan algoritma SPN dilanjutkan dengan algoritma *Blowfish* agar kode yang dihasilkan lebih variatif. Algoritma OTP digunakan untuk menghindari pengulangan atau penundaan kode, sedangkan algoritma kriptografi kunci publik (SPN dan *Blowfish*) digunakan sebagai pengenkripsi algoritma OTP. Ketiga elemen memungkinkan hasil *autentikasi* yang aman. Pembangkitan OTP dilakukan berdasarkan *string* karakter yang dipilih acak oleh sistem. Agar proses *autentikasi* memungkinkan, perlu ada sistem untuk mengenali pengguna. Hal ini dicapai dengan algoritma SPN sebagai algoritma kriptografi kunci publik. OTP yang dihasilkan kemudian dienkripsikan menggunakan kunci yang didapat dengan algoritma SPN kemudian dilanjutkan dengan algoritma *Blowfish*. Kode tersebut merupakan paket yang dapat digunakan sebagai metode *autentikasi*. Kemudian enkripsi yang dihasilkan disimpan kedalam basis data. Saat proses *autentikasi* berlangsung, sistem mencocokan kode yang diinput pengguna dengan kode yang ada di dalam basis data. Apabila cocok, *autentikasi* berhasil. Sebaliknya apabila gagal, *autentikasi* gagal.

2. Rancangan server

Server untuk pengujian metode ini berupa server sederhana, yang akan melakukan *listen* tertentu. Komunikasi dilakukan dengan protokol sederhana. Jika server menerima string :

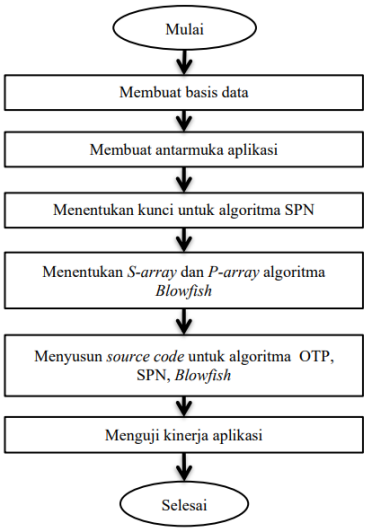
auth:<username>:<publik_key>

paket berikutnya yang diterima dianggap sebagai paket *autentikasi*, dan dilakukan validasi.

3. Rancangan client

Aplikasi client melakukan koneksi *"socket"* ke server dengan mengirimkan *username* dan *password* untuk mendapatkan kode OTP yang sudah ditandatangani dengan kunci privat.

Alur penelitian ditunjukkan pada Gambar 1.



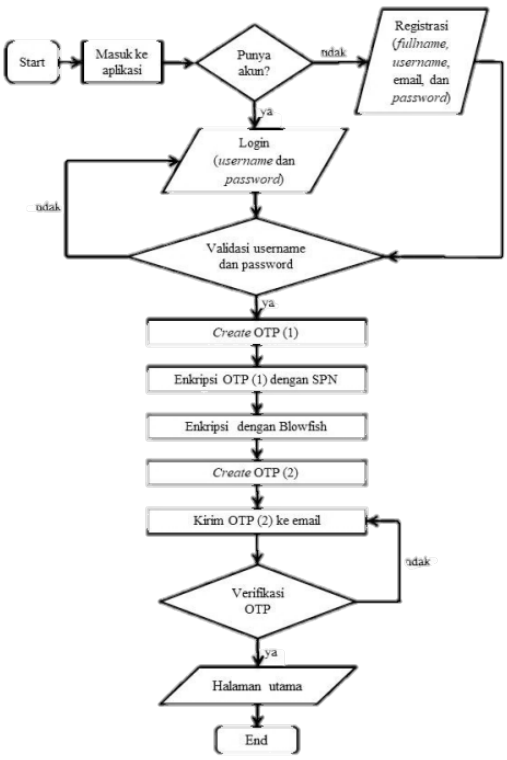
Gambar 1. Alur penelitian

Hasil dan Diskusi

Dalam penyusunan aplikasi, perlu dibuat satu basis data, yaitu basis data login (Tabel 1) dengan id sebagai *primary key* dan kode_otp yang akan selalu berubah saat ada kode baru yang diminta. Basis data login ini nantinya akan menyimpan data yang diinput pengguna dan data OTP. *Field* kode digunakan sebagai pembanding terhadap perubahan kode_otp. Sebelum melakukan koding program, dibuat *flowchart* (Gambar 2) agar pembuatan program lebih terstruktur.

Tabel 1. Basis data login

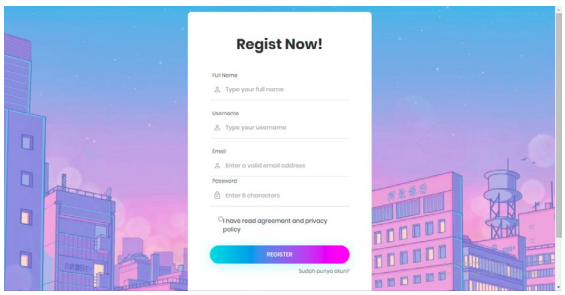
No	Nama Field	Tipe	Panjang Karakter	Ket.
1.	Id	Int	10	Primary Key
2.	Username	Varchar	10	-
3.	Password	Varchar	6	-
4.	Email	Varchar	30	-
5.	Kode_otp	Varchar	8	-
6.	Kode	Varchar	4	-



Gambar 1. *Flowchart* program

1. Register

Pada proses register ini, biasanya pengguna menginputkan data seperti nama, username, email,dan password. Pada halaman register (Gambar 3), pengguna akan memasukkan nama lengkap, *username, password* dan email. Sistem akan mengecek basis data, apabila username atau email sudah terdapat di basis data, maka akan dimunculkan pemberitahuan "email atau username sudah terdaftar". Namun jika belum ada, maka sistem akan melakukan penyimpanan di basis data *user*. Setelah proses registrasi berhasil, maka pengguna akan dialihkan ke halaman *login* aplikasi.

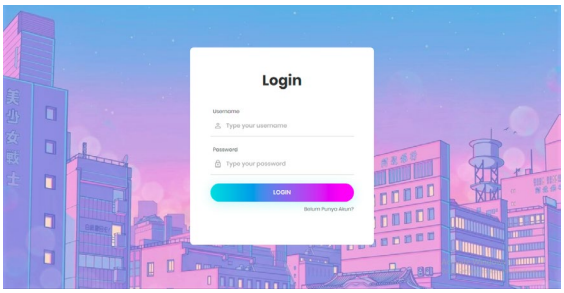


Gambar 2. Antarmuka *register*

2. Login

Login atau Log masuk adalah proses untuk mengakses sistem aplikasi dengan memasukkan identitas diri sesuai yang terdaftar dalam basis data, biasanya proses login memerlukan kombinasi dari username dan password [12].

Pada halaman login (Gambar 4), pengguna memasukkan *username* dan *password* untuk melakukan verifikasi. Sistem akan mengecek apakah kombinasi username dan password sama dengan yang terdapat di basis data, jika kombinasinya benar, maka akan dialihkan ke halaman *insert otp*.



Gambar 3. Antarmuka *login*

Setelah proses login, terdapat proses pembangkitan kode OTP dan pengenkripsian kode OTP dengan algoritma SPN dan *Blowfish*. Namun pada tahap ini, prosesnya tidak ditampilkan secara langsung kepada pengguna. Pada proses ini pembangkitan OTP dilakukan oleh sistem dengan mengambil 4 digit secara acak dari karakter: ABCDEF0123456789. Selanjutnya, kode OTP tersebut dienkripsikan menggunakan algoritma SPN dengan kunci: 1101 0100 0111 1010 0010 1100 1011 1001 dengan permutasi π_s (Tabel 2) dan π_p (Tabel 3) yang telah ditentukan.

Tabel 2. Permutasi $\boldsymbol{\pi_s}$

Z	0	1	2	3	4	5	6	7
$\pi_s(z)$	D	3	1	B	E	9	0	5
Z	8	9	A	B	C	D	E	F
$\pi_s(z)$	6	4	F	A	8	C	7	2

Tabel 3. Permutasi $\boldsymbol{\pi_p}$

Z	1	2	3	4	5	6	7	8
$\pi_p(z)$	5	9	12	7	1	15	4	11
9	10	11	12	13	14	15	16	9
2	13	8	3	10	16	6	14	2

Permutasi π (s) digunakan untuk menentukan kolom v^r pada proses enkripsi. Sedangkan

permutasi π (p) digunakan untuk menentukan pengacakan pada kolom w^r.

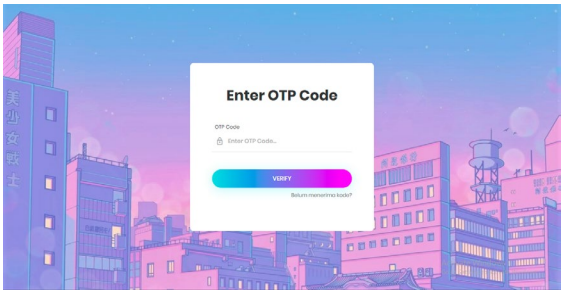
Tabel 4. Inisiasi *S-array*

S-Array	Hexa	Konversi Biner
S1,0	D1310BA6	1101000100110001 0000101110100110
...		
S1,255	6E85076A	0110111010000101 0000011101101010
S2,0	4B7A70E9	0100101101111010 0111000011101001
...		
S2,255	DB83ADF7	1101101110000011 1010110111110111
S3,0	E93D5A68	1110100100111101 0101101001101000
...		
S3,255	406000E0	0100000001100000 0000000011100000
S4,0	3A39CE37	0011101000111001 1100111000110111
...		
S4,255	3AC372E6	0011101011000011 0111001011100110

Hasil enkripsi pada algoritma SPN nantinya akan menjadi kunci pada algoritma *Blowfish*, dan plainteksnya terdiri atas 8 digit dimana 2 digit pertama adalah digit terakhir dari unix-time pada saat pengguna melakukan request otp, dan 6 digit lainnya adalah *password* yang diinputkan oleh pengguna. Sebelum melakukan proses enkripsi, ditentukan terlebih dahulu inisiasi *S-array* (Tabel 4) dan *P-array* (Tabel 5) yang akan digunakan.

3. Insert otp

Setelah proses *login*, pengguna akan dialihkan ke halaman *insert* otp (Gambar 5). Disini, pengguna akan memasukkan kode otp yang diterimanya melalui email. Setelah kode otp diinput, sistem akan melakukan validasi. Apabila kode benar, maka pengguna akan dialihkan ke tampilan utama aplikasi. Namun, jika kode salah, pengguna dapat melakukan proses login ulang untuk mendapatkan kode otp yang baru.



Gambar 4. Antarmuka insert otp

Tabel 5. Inisiasi *P-array*

P-array	Hexa	Konversi Biner
P0	243F6A88	00100100 00111111 01101010 10001000
P1	85A308D3	10000101 10100011 00001000 11010011
P2	13198A2E	00010011 00011001 10001010 00101110
P3	3707344	00000011 01110000 01110011 01000100
P4	A4093822	10100100 00001001 00111000 00100010
P5	299F31D0	00101001 10011111 00110001 11010000
P6	82EFA98	00001000 00101110 11111010 10011000
P7	EC4E6C89	11101100 01001110 01101100 10001001
P8	452821E6	01000101 00101000 00100001 11100110
P9	38D01377	00111000 11010000 00010011 01110111
P10	BE5466CF	10111110 01010100 01100110 11001111
P11	34E90C6C	00110100 11101001 00001100 01101100
P12	C0AC29B	11000000 10101100 00101001 10110111
P13	C97C50DD	11001001 01111100 01010000 11011101
P14	3F84D5B5	00111111 10000100 11010101 10110101
P15	B5470917	10110101 01000111 00001001 00010111
P16	9216D5D9	10010010 00010110 11010101 11011001
P17	8979FB1B	10001001 01111001 11111011 00011011

4. Halaman utama

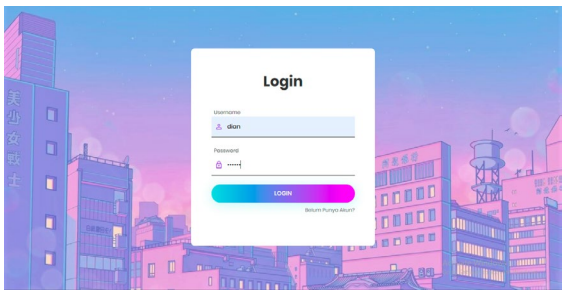
Setelah pengguna memasukkan kode OTP dan melalukan proses autentikasi dengan benar, pengguna akan diarahkan ke halaman utama (Gambar 6).

Gambar 5. Antarmuka halaman utama

Pengujian Aplikasi

1. Halaman Login

Pada halaman awal aplikasi, akan ditampilkan halaman login. Pengguna dapat melakukan proses login apabila telah memiliki akun. Pengguna memasukkan *username* dan *password* yang sudah terdaftar (Gambar 7).

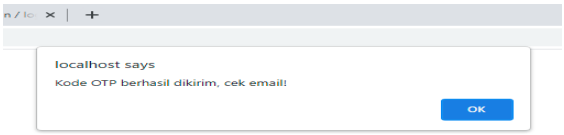


Gambar 6. Contoh pengisian halaman login

Sistem akan mengecek apakah data yang dimasukkan sudah benar dan ada di dalam basis data. Apabila kombinasi *username* dan *password* benar, maka akan ada pemberitahuan bahwa proses login berhasil (Gambar 8).

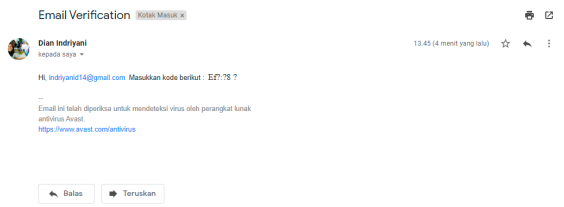
Gambar 7. Pemberitahuan jika login berhasil

Setelah proses login berhasil dilakukan, maka sistem akan mulai memproses kode OTP. Jika kode OTP sudah dikirimkan melalui email (Gambar 9), maka sistem akan mengarahkan pengguna ke halaman *insert* otp.



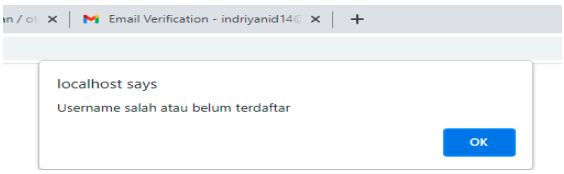
Gambar 8. Pemberitahuan jika kode otp telah dikirimkan

Pengguna dapat mengecek kotak email yang tercantum (Gambar 10) untuk mengetahui kode OTP yang dikirimkan.

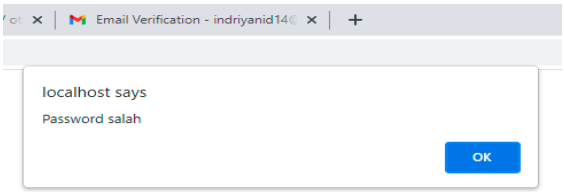


Gambar 9. Kode otp yang dikirimkan

Namun, apabila kombinasi *username* dan *password* salah, maka akan ada pemberitahuan dari sistem (Gambar 11 dan 12), kemudian pengguna diarahkan lagi ke halaman login.



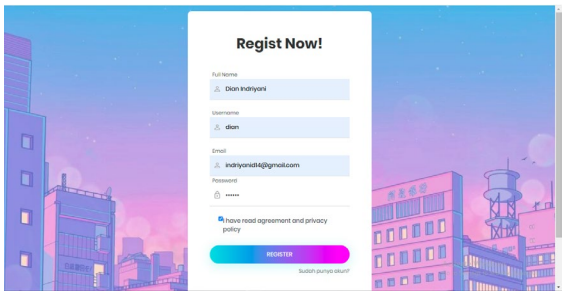
Gambar 10. Tampilan jika *username* yang diinputkan salah



Gambar 11. Tampilan jika *password* yang diinputkan salah

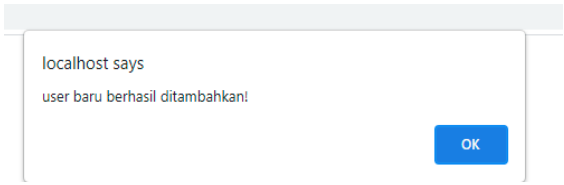
2. Halaman register

Jika pengguna belum memiliki akun, maka pengguna dapat meng-klik tulisan “belum punya akun?” yang ada di pojok kanan bawah pada halaman login, sehingga akan dialihkan ke halaman register. Pengguna kemudian menginputkan data-data yang diperlukan.



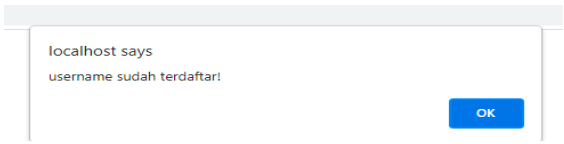
Gambar 12. Contoh pengisian halaman register

Sistem akan memvalidasi data yang masuk, apakah username dan email yang diinputkan sudah ada di dalam basis data atau belum (Gambar 14-16). Jika belum maka pengguna akan diarahkan ke halaman login.

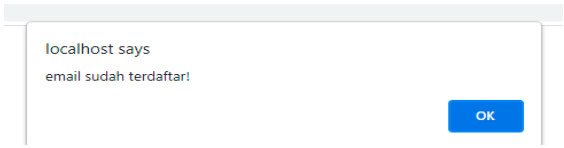


Gambar 13. Tampilan jika register berhasil

Jika sudah ada di basis data, maka sistem akan menampilkan pemberitahuan jika *username* atau email sudah terdaftar.



Gambar 14. Tampilan jika *username* yang diinputkan sudah terdaftar

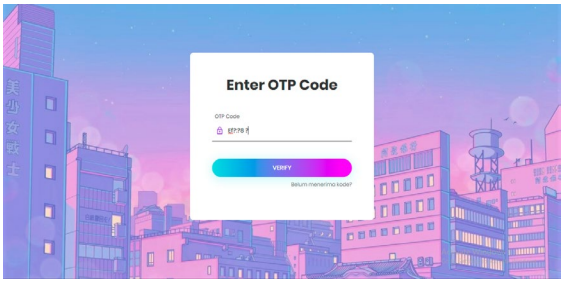


Gambar 15. Tampilan jika email yang diinputkan sudah terdaftar

3. Halaman insert OTP

Pengguna diminta untuk memasukkan kode OTP, kemudian sistem akan mencocokkan OTP yang diinput oleh pengguna dengan OTP yang ada di basis data (Gambar 17 dan 18).

Basis data akan menampilkan kode OTP awal sebelum di enkripsi dan kode OTP dengan enkripsi algoritma SPN-Blowfish (Tabel 6) dengan tujuan untuk mengecek apakah terdapat perbedaan karakter pada kode OTP.

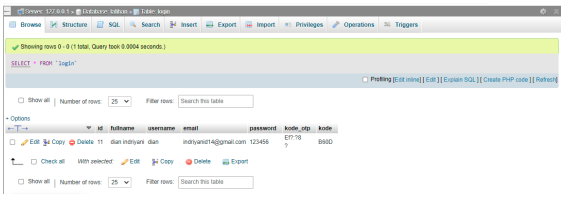


Gambar 16. Tampilan halaman insert otp

Tabel 6. Perbandingan kode otp yang dihasilkan

Percobaan ke-	OTP	OTP + SPN + Blowfish
1	B60D	Ef?:?8 ?
2	9C06	?? 4??3?
3	AF20	??_G??*?

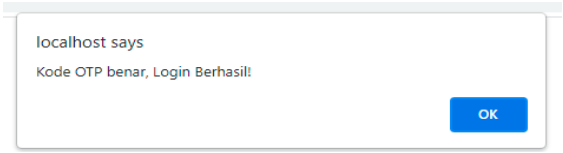
Dari Tabel 6, kode yang dihasilkan dari tiga percobaan memiliki nilai yang berbeda. Kode OTP yang dihasilkan dari algoritma OTP terdiri hanya dari huruf dan atau angka, sedangkan kode yang dihasilkan dari penggabungan algoritma OTP, SPN, dan Blowfish terdiri dari kombinasi karakter yang lebih beragam, seperti "Ef?:?8 ?". Hal ini sesuai dengan tujuan yaitu menghasilkan output kode OTP yang bervariasi (tidak hanya huruf dan angka, melainkan ada karakter lainnya).



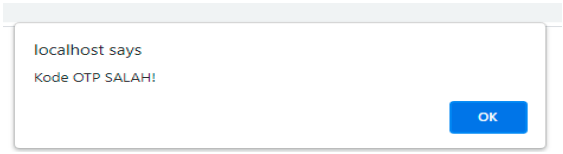
Gambar 17. Data otp yang ada dalam basis data

Jika OTP yang diinputkan benar (Gambar 19), maka akan diarahkan ke halaman utama. Jika pengguna belum menerima kode atau ingin mengganti kode OTP, pengguna dapat meng-klik tulisan "belum menerima kode" yang selanjutnya akan diarahkan kembali ke halaman login.

Jika kode OTP yang diinputkan salah maka akan muncul gambar 20 yaitu pemberitahuan jika OTP salah dan akan diarahkan kembali ke halaman *insert* OTP.



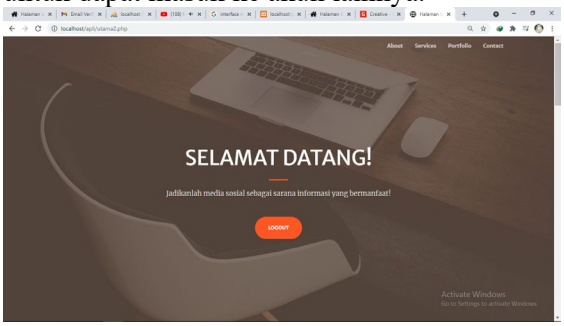
Gambar 18. Tampilan apabila kode otp yang diinputkan benar



Gambar 19. Tampilan apabila kode otp yang diinputkan salah

4. Halaman Utama

Setelah kode OTP dimasukkan, dan sistem memvalidasi kebenarannya, maka pengguna akan diarahkan ke halaman utama (Gambar 21). Di halaman utama, pengguna dapat melakukan logout untuk dapat masuk ke akun lainnya.



Gambar 20. Tampilan jika login berhasil

Simpulan

Berdasarkan tujuan penelitian, dan analisis yang dilakukan, maka dapat ditarik beberapa kesimpulan yaitu sebagai berikut:

1. Aplikasi yang dibuat dapat berfungsi sesuai tujuan, yaitu mengamankan data OTP dengan mengenkripsikannya sehingga tidak mudah ditebak.
2. Penggabungan algoritma OTP, SPN dan *Blowfish* (Tabel 6) menghasilkan output kode OTP yang terdiri dari kombinasi karakter yang lebih beragam dibandingkan dengan kode OTP pada umumnya. Salah satu kode OTP yang dihasilkan dari gabungan 3 algoritma tersebut adalah Ef?:?8 ?, sedangkan kode OTP tanpa algoritma SPN dan Blowfish adalah B60D.

Ucapan Terima Kasih

Peneliti mengucapkan terimakasih kepada semua pihak yang telah membantu penelitian ini, dan termasuk reviewer yang telah menelaah artikel ini.

Pustaka

[1] Abukeshipa, A. S. (2014). *Implementing and* Comprising of *OTP Techniques to Prevent Replay Attack in Radius Protocol.* Gaza: The Islamic University.

[2] Adin, B. (2016). *One-Time Password Berbasis Waktu dan Algoritma.* Bandung: Institut Teknologi Bandung.

[3] Ariel R.L, E. D. (2018). Securing One Time Password (OTP) for Multi-Factor Out-of-Band Authentication through a 128-bit Blowfish Algorithm. *Communication Networks and Information Security (IJCNIS)*, 242-247.

[4] Ariyus, D. (2009). *Keamanan Multimedia.* Yogyakarta: Andi.

[5] Dinesh, E., & Ramesh, S. M. (2021). Security aware data transaction using optimized blowfish algorithm in cloud environment. Journal of Circuits, Systems and Computers, 30(01), 2150004.

[6] Donna R, I. (2020). Literasi Media Sosial : Kesadaran Keamanan dan Privasi dalam Perspektif Generasi Milenial. *Jurnal Penelitian Komunikasi dan Opini Publik Vol 24 No 1*, 1-15.

[7] E. Sediyono, K. I. Santoso, and Suhartono,(2013) "Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS," 2013 International Conference on Advancesin Computing, Communications and Informatics (ICACCI), pp. 1604–1608, 2013

[8] Ezadeen, S., & Alwattar, A. H. (2022). Survey of Blowfish Algorithm for Cloud . Technium: *Romanian Journal of Applied Sciences and Technology*, *4*(6), 18–28. https://doi.org/10.47577/technium.v4i6.6791

[9] Gangireddy, V. K. R., Kannan, S., & Subburathinam, K. (2021). *Implementation* of enhanced blowfish algorithm in cloud environment. Journal of Ambient Intelligence and Humanized Computing, 12(3), 3999-4005

[10] H. Siadati, T. Nguyen, P. Gupta, M. Jakobsson, and N. Memon, (2013),"Mind your SMSes: Mitigating social engineering in second-factorauthentication," Elsevier Journal

of Computers & Security, vol. 65, pp. 14–28, Mar. 2017
[11] Mohsen Gerami and Satar Ghiasvand, (2016), "One-Time Passwords via SMS," Bulletin de la Société Royale desSciences de Liège, vol. 85, pp. 106–113, 2016
[12] Mufid, M. (n.d.). *CloudHost*. Retrieved Agustus 23, 2021, from Login: https://idcloudhost.com/kamus-hosting/
[13] Pramartha, I. (2013, Maret 13). *Wahyu Pramartha.* Retrieved Juni 25, 2021, from dewahyupramartha: dewahyupramartha.wordpress.com
[14] Sadiq, N. A., Abdullahi, M., Rana, N., Chiroma, H., & Dada, E. G. (2018). Development of blowfish encryption scheme for secure data storage in public and commercial cloud computing environment. i-Manag J Cloud Comput, 5, 1.
[15] Stinson, D. R. (2006). *Cryptography Theory and Practice Third Edition.* Florida: Chapman & Hall/CRC.
[16] Syafri, A. (2007). Retrieved Oktober 15, 2020, from Sekilas Tentang Enkripsi Blowfish: www.ilmukomputer.com
[17] Wade, & C., L. (2006). *Introduction to* Cryptography *with Coding Theory (edisi-2).* Washington: Pearson Prentice Hall.