PENERAPAN KEAMANAN *REMOTE* SERVER MELALUI SSH DENGAN KOMBINASI KRIPTOGRAFI ASIMETRIS DAN AUTENTIKASI DUA LANGKAH

Tohirin

Program Studi Pascasarana Sistem Informasi, STIMK LIKMI
Jl. Ir. H. Juanda No.96, Kota Bandung, Jawa Barat 40132
tohirin07@gmail.com

Abstract - System security is an absolute requirement that must be considered by users, especially network administrators to protect data. Every time a system is maintained and monitored, an administrator requires remote access such as a secure shell (SSH) to enter the system with a secure channel. By default, SSH is not secure because there is a great chance that the account will be taken over by brute force techniques. The application of asymmetric cryptography on SSH accounts is considered safe than SSH remote login by default. However, this is still not enough because hackers could control the computer that is used by administrators to do remote servers. It also can be hacked with key scanning techniques. The combination of asymmetric cryptography and two-step authentication can be a solution so that the server will be very difficult to penetrate.
Keywords - Server Security, SSH, Asymmetric Cryptosystems, Two-Step Authentication.

Abstrak - Keamanan sistem menjadi syarat mutlak yang harus diperhatikan oleh pengguna, khususnya administrator jaringan guna melindungi data. Setiap melakukan pemeliharaan dan *monitoring* sistem, seorang administrator membutuhkan akses masuk dengan cara *remote* seperti *secure shell* (SSH) ke dalam sistem dengan saluran yang aman. Secara *default*, SSH tidak aman karena berpeluang besar akun diambil alih dengan teknik *brute force*. Penerapan kriptografi asimetris pada akun SSH dinilai aman daripada *remote login* SSH secara *default*. Akan tetapi hal tersebut masih belum cukup karena bisa saja peretas justru mengendalikan komputer yang digunakan oleh administrator dalam melakukan *remote* server. Selain itu juga dapat diretas dengan teknik *key scanning*. Kombinasi kriptografi asimetris dan autentikasi dua langkah dapat menjadi solusi sehingga server akan sangat susah ditembus.
Kata Kunci – Keamanan Server, SSH, Kriptografi Asimetris, Autentikasi Dua Langkah.

I. PENDAHULUAN

Personal komputer dan server yang terhubung dengan jaringan lokal maupun internet sangat mungkin menjadi sasaran pihak atau orang yang tidak bertanggungjawab seperti peretas. Oleh karenanya, keamanan sistem menjadi syarat mutlak yang harus diperhatikan oleh pengguna, khususnya administrator jaringan. Keamanan sistem dapat diartikan sebagai upaya mencegah, mengidentifikasi dan melakukan respon terhadap pengguna yang tidak terdaftar pada sistem atau jaringan komputer.

Seorang administrator jaringan membutuhkan akses secara *remote* ke dalam sistem guna melakukan pemeliharaan atau pengawasan berkala. Protokol *remote* akses yang paling umum dipakai administrator adalah *secure shell* (SSH). Secara konfigurasi *default*, SSH dinilai tidak aman karena memungkinkan seorang peretas melakukan pengambilalihan akun dengan cara *brute force* [1]. Selain itu, pengguna diizinkan melakukan *login* secara langsung sebagai *root* [2]. Untuk itu diperlukan pengamanan sistem *remote* server SSH secara berlapis.

Penerapan kriptografi pada akun SSH sebagai langkah pengamanan sistem secara berlapis dinilai aman daripada *remote login* SSH secara *default* [3]. Akan tetapi hal tersebut masih belum cukup karena bisa saja peretas justru mengendalikan komputer yang digunakan oleh administrator dalam melakukan *remote* server. Selain itu, literatur menyatakan bahwa kemungkinan bisa diretas juga dengan teknik *key scanning* [4]. Kombinasi kriptografi dan autentikasi dua langkah dapat menjadi solusi sehingga server akan sangat susah ditembus.

A. Keamanan Sistem

Keamanan sistem dapat diartikan sebagai upaya mencegah dan mengidentifikasi pengguna yang tidak terdaftar pada sistem atau jaringan komputer dengan tujuan untuk antisipasi risiko jaringan atau sistem komputer dari ancaman secara fisik maupun logik [5]

Ada tiga level tindakan dalam mengamankan sistem dari serangan, yaitu: Preventif atau pencegahan dengan cara melakukan konfigurasi sistem sebisa mungkin terhindar dari kesalahan. Kesalahan konfigurasi merupakan jenis risiko celah keamanan yang paling umum ditemukan pada sebuah sistem operasi *workstation* maupun server, *framework*, *library* dan aplikasi. Pada tahun 2020, kesalahan konfigurasi merupakan risiko celah keamanan paling populer ketujuh dari sepuluh daftar celah keamanan yang paling berbahaya versi OWASP [6]; Observasi, yaitu perawatan sistem komputer berdasarkan analisis isi log tidak normal yang bisa merujuk pada masalah

keamanan yang tidak terpantau *Intruder*. Contoh penggunaan pada tingkatan observasi penerapan IDS (*Intruder Detecting System*); Respon, yakni tindakan yang diberikan sistem ketika berhasil disusupi oleh pihak yang tidak bertanggung jawab. Contoh penggunaan pada tingkatan respon adalah penerapan IDS (*Intrusion Prevention System*).

B. Secure shell (SSH)

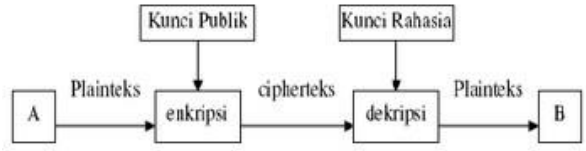
SSH atau *secure shell* merupakan protokol jaringan yang terdapat pada lapisan aplikasi pada protokol model OSI dan TCP/IP. SSH berjalan pada *port* standar TCP *nomor* 22. SSH difungsikan untuk memfasilitasi sistem komunikasi aman antara dua sistem yang memakai arsitektur klien server. Meski demikian, internet tidak sepenuhnya aman. SSH menyediakan integritas dan kerahasiaan data melewati teknik enkripsi dan dekripsi dilakukan secara otomatis pada koneksinya [7]. SSH digunakan untuk melakukan *remote* sistem sehingga administrator tidak perlu masuk secara langsung, melakukan, *forwarding port* TCP, *tunneling* dan koneksi X11. SSH paling banyak dipraktikkan oleh pengguna sistem operasi Unix dan Linux.

Sebagai autentikasi komputer atau server *remote* , digunakan kriptografi jenis kunci publik pada SSH. Koneksi dengan *daemon* SSH harus terjadi terlebih dahulu *remote* server dapat dilakukan.

C. Kriptografi Asimetris

Dalam banyak literatur klasik disimpulkan bahwa kriptografi merupakan seni sekaligus ilmu guna menjaga kerahasiaan informasi atau pesan menggunakan cara penyandian pada pesan atau informasi itu sendiri sehingga tidak dapat dipahami lagi maksudnya [8]. Berdasarkan sifat dari kunci yang dipakai, algoritma terbagi dua, yakni simetris dan asimetris [9]. Sederhananya, algoritma simetris menggunakan satu kunci rahasia, sedangkan asimetris disebut juga *public key algorithm* menggunakan sepasang kunci rahasia dan kunci publik di mana kunci publik digunakan untuk proses enkripsi dan kunci rahasia digunakan untuk dekripsi. Alur kunci publik algoritma asimetris dapat dilihat pada gambar 1.

Salah satu algoritma Asimetris adalah RSA (Rivest Shamir Adleman) yang pertama kali ditemukan oleh di 1977. Penamaan RSA diambil dari ketiga nama tengah penemunya. RSA masih diimplementasikan dan direkomendasikan dalam berbagi aplikasi selama belum ditemukan algoritma yang mangkus untuk memfaktorkan bilangan bulat menjadi bilangan primanya [10]. Tingkat kompleksitas memfaktorkan bilangan besar merupakan penentu keamanan RSA [9] [10].



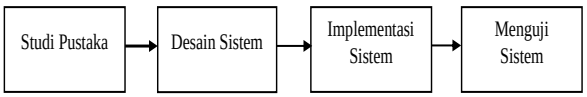
Gambar 1. Alur Kunci Algoritma Asimetris

D. Autentikasi Dua Langkah

Autentikasi Dua Langkah (TFA) merupakan salah satu metode autentikasi pengguna di mana dua dari tiga langkah yang memiliki sifat independen akan digunakan guna membuktikan kebenaran bahwa identitas pengguna tersebut asli [11]. Dengan memanfaatkan TFA, kata sandi sebagai hak akses identitas pengguna bukan lagi bersifat *single point of attack* [12]. Banyak perusahaan yang telah menambahkan TFA untuk mengamankan akan serangan *brute force* [12], seperti Google [13], WhatsApp [14], Facebook [15] dan Twitter [16]. Di Android sendiri banyak aplikasi yang bisa digunakan untuk *authenticator*, seperti Google *Authenticator* [17] dan Authy [18].

II. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian induktif di mulai dari pengumpulan data dari berbagai literatur, setelah semua data dirasa cukup selanjutnya melakukan desain, implementasi dan pengujian sistem sebagaimana dapat dilihat pada gambar berikut.



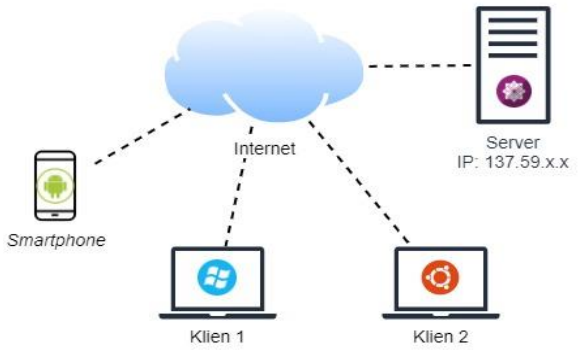
Gambar 2. Metode Penelitian

1. Mengumpulkan Data
   Penelitian diawali dengan pencarian dan pengumpulan data dari buku, situs web dan penelitian yang menunjang terlaksananya penelitian ini.
2. Desain Sistem
   Desain sistem berupa gambar topologi jaringan dengan arsitektur klien server sekaligus arsitektur teknologi yang dipakai.
3. Mengimplementasikan Sistem
   Sistem yang telah selesai didesain kemudian diiimplementasikan dalam tiga tahap, yakni: instalasi dan konfigurasi *remote* server melalui SSH secara *default*; instalasi dan konfigurasi *remote* server melalui SSH dengan kriptografi asimetris RSA; instalasi dan konfigurasi *remote* server melalui SSH dengan kombinasi kriptografi asimetris RSA dan autentikasi dua langkah.
5. Menguji Sistem
   Tahap pengujian dilakukan sebanyak tiga kali setiap selesai masing-masing tahap implementasi.

III. HASIL DAN PEMBAHASAN

A. *Desain Sistem*

Desain sistem berupa topologi jaringan dengan arsitektur klien server adalah sebagaimana Gambar 3. Pada gambar tersebut terlihat peneliti menggunakan empat perangkat, di mana perangkat pertama difungsikan sebagai Server, perangkat kedua sebagai klien 1 yang akan didaftarkan untuk bisa melakukan *remote* ke server menggunakan kombinasi algoritma simetris dan autentikasi dua Langkah, perangkat ketiga sebagai klien 2 yang tidak didaftarkan untuk bisa melakukan *remote* ke server, dan perangkat keempat merupakan *smartphone* yang berfungsi untuk tempat instalasi Google *Authenticator*.



Gambar 3. Arsitektur Jaringan

Desain sistem berupa teknologi yang dipakai pada sistem klien dan server adalah sebagaimana tabel berikut.

Tabel 1. Arsitektur Teknologi Sistem

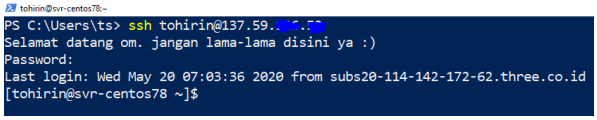
1	Sistem Operasi Server	CentOS 7.8
2	Sistem Operasi Klien 1	Windows 10
3	Sistem Operasi Klien 2	Ubuntu 18.04 LTS
4	*Smartphone*	Android 9

B. *Implementasi dan Pengujian Sistem*

Implementasi sistem tahap pertama dimulai dari instalasi dan konfigurasi OpenSSH yang merupakan program untuk SSH secara *default* di perangkat server untuk selanjutnya dilakukan pengujian *remote* SSH dari perangkat klien 1. Adapun instalasi dan konfigurasi SSH adalah sebagai berikut.

1. Lakukan instalasi paket OpenSSH dan daemon OpenSSH (sshd) dengan perintah: *sudo yum –y install openssh-server openssh-clients*
2. Jalankan service SSH dengan perintah: *sudo systemctl start sshd*. Ketika aktif, sshd akan secara terus menerus mendengarkan koneksi yang akan dan sudah terjadi dari perangkat klien, dan mengaturnya dengan benar
3. Lakukan pengecekan apakah sshd benar sudah aktif dengan menggunakan perintah: *sudo systemctl status sshd*
4. Lakukan aktivasi SSH secara otomatis setiap setelah sistem restart dengan menggunakan perintah: *sudo systemctl enable sshd*
5. Jika diperlukan, lakukan beberapa konfigurasi tambahan dengan melakukan perubahan pada *file* /etc/ssh/sshd_config seperti mengubah port bawaan, menonaktifkan login *remote* server melalui SSH memakai user root pada parameter *PermitRootLogin no*, dan lain sebagainya.

Setelah tahap instalasi dan konfigurasi OpenSSH pada server telah dilakukan, selanjutnya adalah menguji apakah perangkat klien 1 dapat melakukan *remote* server melalui SSH dengan benar menggunakan Windows PowerShell sebagaimana dapat dilihat pada gambar berikut.

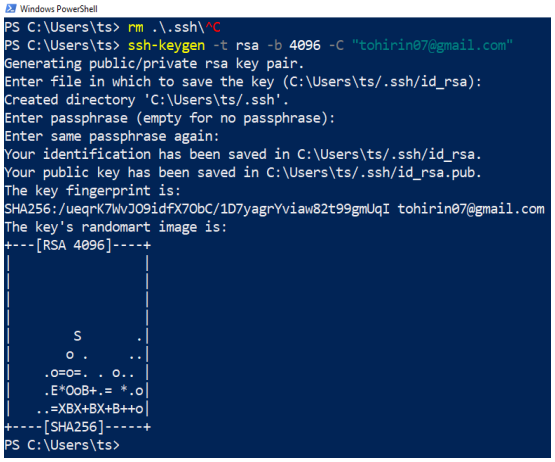


Gambar 4. Pengujian *remote* SSH *default*

Dari pengujian sesuai gambar di atas, perangkat klien 1 bisa melakukan *remote* akses server melalui SSH dengan konfigurasi *default* dengan benar.

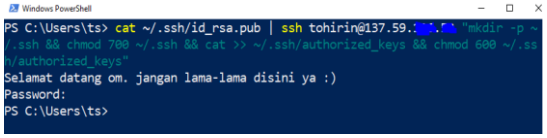
Selanjutnya adalah implementasi kriptografi asimetris dengan algoritma RSA dikonfigurasi sebagai langkah berikut.

1. Pada perangkat klien 1, lakukan *generate* 4096 bit ssh-key dengan penanda identitas berupa alamat surel pribadi dengan perintah: *ssh-keygen -t rsa -b 4096 -C "tohirin07@gmail.com"*, hasilnya adalah sebagai sebagaimana gambar berikut.



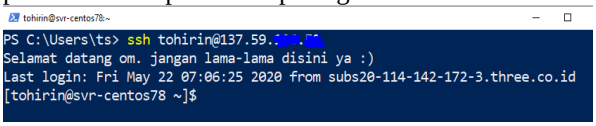
Gambar 5. *Generate* SSH *key*

2. Pada perangakat server, buat direktori .ssh dengan hak izin 700 dan berkas ~/.ssh/authorized_keys dengan hak izin 600. Lalu unggah konten .ssh/id_rsa.pub dari perangkat klien ke dalam berkas ~/.ssh/authorized_keys di perangkat server. Semua perintah tersebut dieksekusi secara *remote* dari perangkat klien 1 sebagaimana dapat dilihat pada gambar berikut.

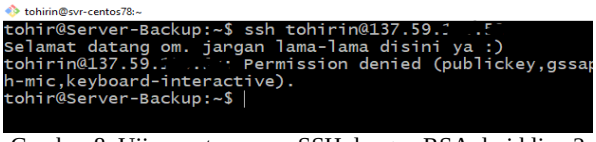


Gambar 6. Unggah id_rsa.pub ke *authorized_keys*

Setelah implementasi kriptografi asimetris dengan algoritm RSA pada klien 1, maka dilakukan uji *remote* klien 1 tersebut untuk membuktikan apakah implementasi sudah sesuai. Dilakukan uji *remote* juga pada klien 2 sebagai pembanding klien 1. Pengujian pada klien 1 dapat dilihat pada gambar 7, sedangkan pada klien 2 dapat dilihat pada gambar 8.



Gambar 7. Uji *remote* server SSH dengan RSA dari klien 1



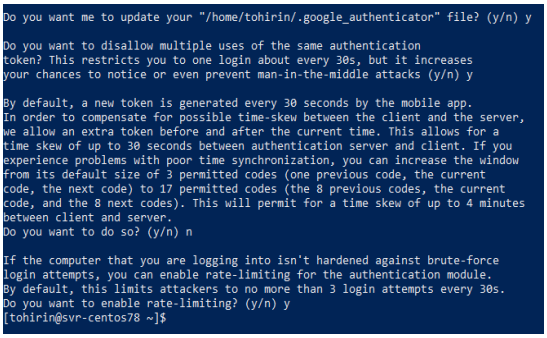
Gambar 8. Uji *remote* server SSH dengan RSA dari klien 2

Sebagaimana gambar 7 dan 8 di atas, klien 1 dapat melakukan *remote* server melalui SSH dengan algoritma asimetris RSA, sedangkan klien 2 tidak dapat melakukannya.

Selanjutnya adalah implementasi kriptografi asimetris dengan algoritma RSA dipadukan dengan autentikasi dua langkah sebagai berikut.

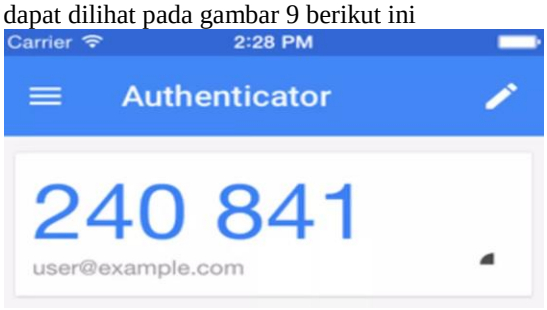
1. Pada perangkat server, lakukan instalasi authenticator menggunakan perintah: *sudo yum -y google-authenticator*, lalu jalankan google-authenticator tadi pada profil pengguna yang akan digunakan untuk melakukan *remote* ke server, maka akan muncul *barcode* sebagaimana gambar 9, ikuti dan sesuaikan pengaturan pada interaksi pertanyaan yang muncul sebagaimana gambar 10.

Gambar 9. *Output* perintah *google-authenticator*



Gambar 10. Respon interaksi instalasi *google-authenticator*

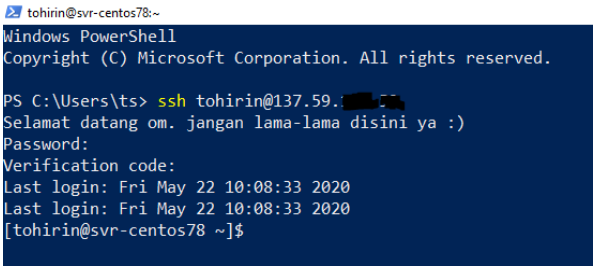
2. Pada perangkat *smartphone*, unggah dan lakukan instalasi Google *Authenticator*. Kemudian pindai *barcode* yang terdapat pada perangkat server, maka akan tampil kode autentikasi sebagaimana dapat dilihat pada gambar 9 berikut ini



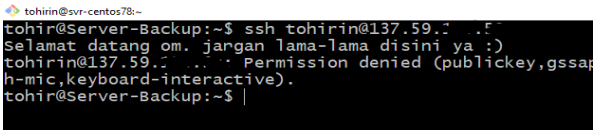
Gambar 11. Kode autentikasi

3. Lakukan beberapa konfigurasi di perangkat server. Pada baris paling bawah berkas /etc/pam.d/sshd, tambahkan komentar *auth required pam_google_authenticator.so*. Kemudian pada berkas /etc/ssh/sshd_config, lakukan beberapa penyesuaian pada beberapa komentar berikut: *ChallengeResponseAuthentication yes; UsePAM yes; AuthenticationMethods publickey,keyboard-interactive; PasswordAuthentication no*
4. Simpan setiap kali ada perubahan konfigurasi dan lakukan *restart service* SSH dengan menggunakan perintah: *systemctl restart sshd*

Setelah implementasi kriptografi asimetris dengan algoritma RSA dipadukan dengan autentikasi dua langkah, selanjutnya dilakukan uji *remote* klien 1 tersebut untuk membuktikan apakah implementasi sudah sesuai. Dilakukan uji juga pada klien 2 sebagai pembanding. Pengujian pada klien 1 dapat dilihat pada gambar 11 klien 2 pada gambar 12.



Gambar 12. Uji *remote* server SSH dengan RSA dan autentikasi dua langkah pada klien 1



Gambar 13. Uji *remote* server SSH dengan RSA dan autentikasi dua langkah pada klien 2

Sebagaimana gambar 12 dan 13 di atas, klien 1 dapat melakukan *remote* server melalui SSH dengan algoritma asimetris RSA dipadukan autentikasi dua langkah, sedangkan klien 2 tidak dapat melakukannya sama sekali.

IV. KESIMPULAN

Dari pemaparan secara keseluruhan dapat disimpulkan bahwa penggunaan konfigurasi SSH bawaan tidak direkomendasikan karena berpeluang besar akun dicuri dengan teknik *brute-force*. Dengan menerapkan kombinasi kriptografi asimetris dan autentikasi dua langkah pada *remote* login server linux menggunakan SSH jauh lebih aman daripada *remote login* SSH hanya menggunakan kriptografi asimetris saja karena bisa saja seorang peretas mengendalikan komputer yang digunakan oleh administrator dalam melakukan *remote* server. Selain itu juga ada kemungkinan bisa diretas dengan teknik *key scanning*. Dengan menerapkan kombinasi kriptografi asimetris dan autentikasi dua langkah setidaknya seorang peretas harus mengambil alih dua perangkat sekaligus, yaitu komputer dan *smartphone* yang digunakan untuk *remote* server, dan hal tersebut bukanlah sesuatu yang mudah dilakukan.

DAFTAR PUSTAKA

[1] B. Sakti, A. Aziz dan A. Doewes, "Uji Kelayakan Implementasi SSH sebagai Pengaman FTP Server dengan Penetration Testing," *JURNAL ITSMART* , vol. 2, no. 1, pp. 44-51, 2013.

[2] D. J. Barrett, R. E. Silverman dan R. G. Byrnes, Ssh, The Secure Shell: The Definitive Guide Second Edition, O'Reilly, 2005.

[3] M. Iqbal, "Keamanan Remote Server Melalui Ssh Dengan Kriptosistem Simetris," *TECHSI: Jurnal Penelitian Teknik Informatika* , vol. 3, no. 2, pp. 54-66, 2013.

[4] ssh.com, "Hackers are now scanning for SSH keys to exploit," 20 June 2018. [Online]. Available: https://blog.ssh.com/ssh-key-scan-attack-honeypot. [Diakses 22 May 2020].

[5] I. R. P. Jeinever, A. Rasyid dan N. Suharto, "Penerapan Sistem Keamanan Jaringan Menggunakan Random Port Knocking Berbasis Raspberry Pi Yang Dikirm Melewati Telegram," *Jurnal JARTEL*, vol. 7, no. 2, pp. 99-105, 2018.

[6] "OWASP Top Ten Web Application Security Risks," OWASP Foundation, Inc., 2020. [Online]. Available: https://owasp.org/www-project-top-ten/. [Diakses 19 May 2020].

[7] H. Jusuf, "Penggunaan Secure Shell (SSH) Sebagai Sistem Komunikasi Aman Pada Web Ujian Online," *BINA INSANI ICT JOURNAL*, vol. 2, no. 2, pp. 75 - 84 , 2015.

[8] M. Y. Simargolang, "Implementasi Kriptografi Rsa Dengan Php," *JURNAL TEKNOLOGI INFORMASI (JurTI)*, vol. 1, no. 1, pp. 1-10, 2017.

[9] I. Wibowo , B. Susanto dan J. K. T, "Penerapan Algoritma Kriptografi Asimetris Rsa Untuk Keamanan Data Di Oracle," *JURNAL INFORMATIKA*, vol. 5, no. 1, 2009.

[10] D. Ariyus, Pengantar Ilmu Kriptografi, Yogyakarta: Andi, 2008.

[11] D. M. Soete, Two-Factor Authentication, Springer US, 2011.

[12] S. W. Raharjo, I. D. E.K.Ratri dan H. Susilo, "Implementasi Two Factor Authentication Dan Protokol Zero Knowledge Proof Pada Sistem Login," *Jurnal Teknik Informatika dan Sistem Informasi* , vol. 3, no. 1, pp. 127-136, 2017.

[13] Google, "Mengaktifkan Verifikasi 2 Langkah," Google, 2020. [Online]. Available: https://support.google.com/accounts/answer/185839?co=GENIE.Platform%3DDesktop&hl=id. [Diakses 20 May 2020].

[14] WhatsApp, "Menggunakan verifikasi dua langkah," WhatsApp, 2020. [Online]. Available: https://faq.whatsapp.com/id/android/26000021/. [Diakses 20 May 2020].

[15] Facebook, "Apa itu autentikasi dua faktor dan bagaimana cara kerjanya di Facebook?," 2020. [Online]. Available: https://web.facebook.com/help/148233965247823?_rdc=1&_rdr. [Diakses 20 May 2020].

[16] Twitter, "How to use two-factor authentication," [Online]. Available: https://help.twitter.com/en/managing-your-account/two-factor-authentication. [Diakses 20 May 2020].

[17] G. LLC, "Google Authenticator," [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=en. [Diakses 20 May 2020].

[18] Authy, "Authy | Two-Factor Authentication (2FA) App & Guides," TWILIO, INC, [Online]. Available: https://authy.com/. [Diakses 20 May 2020].