Two Factor Authentication Berbasis SMS pada Layanan Single Sign-On Universitas Mataram

SMS Based Two Factor Authentication in Mataram University's Single Sign-On Service

Ahmad Zafrullah Mardiansyah*, Ariyan Zubaidi,
I Gde Putu Wirarama Wedaswhara W., Andy Hidayat Jatmika
Dept Informatics Engineering, Mataram University
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: zaf@unram.ac.id, [zubaidi13, wirarama, andy]@unram.ac.id

Penulis korespondensi

Abstract **Security is an important part of electronic transactions. Information systems at Mataram University are generally using the conventional username and password fields to authenticate the user (Single Factor Authentication - SFA). The SFA method is vulnerable to brute force attacks, especially to the Single Sign-On (SSO) service. To prevent brute force attacks, this study proposes an implementation of SMS-based Two Factor Authentication (TFA) toward SSO. This study found that the tests carried out by analyzing the simulation of brute force attacks, attackers could not obtain access to the user accounts. Another test done by the User Experience Questionnaire (UEQ) method found the score on the Pragmatic aspect was 1.927 (Excellent), on the Hedonic aspect it was 1.667 (Excellent), and overall was 1.797 (Excellent). Additional positive value comes from users who feel the features can help support the utilization of SSO. The reduction comes from users who feel they are not motivated enough to take advantage of TFA.**

Key words**: two factor authentication, single sign-on, user experience questionnaire, sms gateway**

I. PENDAHULUAN

Keamanan menjadi kebutuhan pokok dalam melakukan transaksi secara elektronik. Sistem Informasi di lingkungan Universitas Mataram secara umum menyediakan model autentikasi terhadap pengguna berupa *username* dan *password*. Sistem melakukan autentikasi dengan mencocokkan kombinasi *username* dan *password* pengguna yang ada didalam *database* sehingga mampu menentukan apakah pengguna dapat diberikan akses atau tidak. Model autentikasi seperti ini juga sering disebut dengan *Single Factor Authentication* (SFA) [1].

Pada saat implementasi di dalam sistem, *password* pengguna tidak disimpan sebagai *plain-text* biasa. *String password* lebih sering disimpan dalam bentuk *hash*, yakni metode enkripsi dengan skema satu arah. Artinya *string password* yang telah di-*hash* tidak dapat diketahui nilai aslinya [2].

Dari sisi sistem, SFA memiliki beberapa kelebihan bagi pengguna. Di antaranya adalah mudah untuk diimplementasikan, tidak membutuhkan perlengkapan khusus untuk dapat dijalankan, dan mudah untuk menyediakan fasilitas lupa *password*. Bagi pengguna SFA memiliki beberapa celah keamanan yang umum terjadi, di antaranya *pasword* sangat memungkinkan untuk dilihat oleh orang lain yang ada dibelakang pengguna dan tingkat keamanan sangat bergantung pada kekuatan *password* (karakter spesial, panjang karakter, dan kombinasi karakter).

Secara psikologis, pengguna cenderung akan menggunakan *string password* yang mudah diingat. Kombinasi karakter agar menjadi sebuah kata yang mudah diingat akan menjadi terbatas. Dengan perkembangan teknologi saat ini beberapa penyedia dapat meng-*generate* kombinasi *string* yang memiliki kemungkinan cukup besar untuk digunakan sebagai *password* oleh kebanyakan pengguna. Daftar kombinasi *string password* tersebut lebih sering dikenal sebagai *word-lists*.

Masalah keamanan yang sering terjadi pada SFA adalah terkena serangan *brute force*. *Brute force* dilakukan dengan melakukan percobaan kombinasi *username* dan *password* yang sudah dimiliki dalam *word-lists* sampai mendapatkan response *"true"* dari sesi *login*. *Brute force* dilakukan secara otomatis oleh program, dengan membaca dan mengirim kombinasi *username* dan *password* setiap interval waktu tertentu [2], [3].

Teknik serangan *brute force* menggunakan pendekatan yang berbeda dengan serangan teknik lainnya. *Brute force* tidak perlu melakukan *decrypt* (*de-hash*) nilai *string* dari *password* pengguna yang ada pada *database*. Namun fokus utama *brute force* adalah mengumpulkan daftar kombinasi *username* dan *password* untuk dimasukkan kedalam pustaka *word-lists*.

Mempertimbangkan masalah yang sering kali terjadi pada model SFA, model pengamanan autentikasi tambahan yang diusulkan dalam penelitian ini adalah *two factor authentication* (TFA). TFA merupakan tahapan verifikasi tambahan yang diberikan setelah melalui SFA. TFA menuntut pengguna untuk membuktikan bahwa yang memasukkan *username* dan *password* tersebut memang betul pengguna yang bersangkutan. Pembuktian atau verifikasinya dilakukan dengan mengirim sebuah kode kepada pengguna sesaat setelah pengguna berhasil

melakukan autentikasi menggunakan *username* dan *password* [4]–[6].

Salah satu media autentikasi tambahan untuk TFA adalah berbasis SMS. Kode autentikasi (*token*) dikirim secara otomatis oleh sistem sesaat setelah autentikasi *username* dan *password* kepada pengguna. *Token* ini yang nanti harus dimasukkan pengguna kedalam sistem untuk memastikan bahwa pengguna yang bersangkutanlah yang sedang melakukan transaksi, bukan orang yang lainnya. TFA berbasis SMS saat ini telah banyak digunakan oleh bank untuk membantu sistem memastikan keaslian transaksi dari pengguna. Dengan begitu pihak bank dapat meminimalisir kemungkinan terjadinya pencurian saldo nasabah [7].

Tingkat penetrasi pengguna *smartphone* dengan fasilitas SMS saat ini cukup tinggi. *Smartphone* terkini secara *default* telah dibekali dengan fasilitas SMS. Hal ini membuat implementasi TFA berbasis SMS menjadi lebih mudah untuk direalisasikan [8].

Dari segi keamanan, implementasi TFA dapat membantu sistem untuk memastikan keaslian dari autentikasi pengguna dari percobaan serangan yang mungkin dapat terjadi. Namun dari segi *user experience* (UX), tambahan tahapan dalam proses autentikasi akan membuat pengguna merasa sedikit kesulitan. Untuk itu pada penelitian ini, akan diimplementasikan TFA terhadap *Single Sign-On* (SSO) yang ada pada sistem-sistem di lingkungan Universitas Mataram untuk meningkatkan keamanan sistem serta melakukan analisa terhadap dampak UX yang terjadi di sisi pengguna [9], [10].

II. Tinjauan Pustaka

Beberapa penelitian terkait telah dilakukan diantaranya adalah [8], [11] yang meningkatkan sistem keamanan menggunakan skema *One Time Password* (OTP). Skenario autentikasi dengan menggunakan OTP memanfaatkan fasilitas SMS yang dikirimkan kepada nomor pengguna. OTP yang dikirimkan di-*generate* dengan perhitungan matematis untuk memastikan kombinasi *username*, *password*, dan OTP tidak mudah ditebak. Pengujian yang dilakukan menghasilkan dua kesimpulan, yang pertama adalah peningkatan dari segi keamanan. Dan hasil pengujian kedua adalah ditemukan bahwa beban komputasi untuk implementasi sistem keamanan OTP menjadi lebih besar.

Pemanfaatan lapisan tamabahan untuk keamanan menjadi perhatian penting untuk perencanaan dan pengembangan sistem kedepannya. Tingkat keamanan dan metode autentikasi yang dapat digunakan oleh pengguna dianalisa oleh penelitian yang dilakukan [12]. Penelitian tersebut memberikan preferensi kepada pengguna untuk bebas menggunakan metode autentikasi yang diinginkan, sesuai dengan tingkat pemahaman terhadap IT. Dengan begitu pengembang melakukan identifikasi titik lemah kemamanan dari sebaran pengguna ada dibagian yang mana.

Implementasi TFA lainnya yang menggunakan skenario OTP yang dilakukan oleh [13] adalah menggunakan *Global Positioning System* (GPS). Penggunaan GPS dapat menambah tahapan proses autentikasi yang dilakukan oleh pengguna. Mulai dari kombinasi *username* dan *password*, sistem memeriksa lokasi pengguna, dan kemudian yang terakhir adalah mengirimkan OTP melalui SMS. Dari percobaan dan analisa yang dihasilkan, penelitian tersebut dapat meningkatkan sistem keamanan dengan meminimilasir kemungkinan dari terkena serangan *brute force*.

Penelitian lain terkait OTP dilakukan oleh [7]. Penelitian tersebut melakukan peningkatan keamanan dengan melakukan hash pada kode unik yang dikirimkan melalui SMS. Peneliti dalam penelitian tersebut menunjukkan kemungkinan kode OTP yang dikirim kepada pengguna dapat diketahui oleh penyerang menggunakan beberapa teknik serangan. Untuk itu dengan perhitungan matematis, kode OTP yang dikirimkan dilakukan *hash* menggunakan kombinasi dari beberapa isian data pengguna dan waktu sistem. Pengujian yang dilakukan menghasilkan kode OTP yang tidak pernah sama antar waktu sehingga disimpulkan bahwa kode OTP memiliki kemungkinan yang sangat kecil untuk dapat diketahui oleh penyerang.

Proses OTP umumnya dilakukan melalui SMS. Penelitian [14] menjelaskan masalah keamanan yang masih mungkin terjadi jika OTP dikirimkan melalui *protocol* SMS standar. *Protocol* SMS standar memungkinkan untuk dilihatnya isi pesan ketika pesan sedang dikirim atau diterima. Penelitian tersebut mengembangkan lapisan keamanan tambahan dengan menambahkan model enkripsi terhadap isi SMS, sehingga menjamin isi dari SMS tidak dapat terbaca dengan mudah meskipun jalur komunikasinya dapat ditembus.

III. Metodologi Perancangan

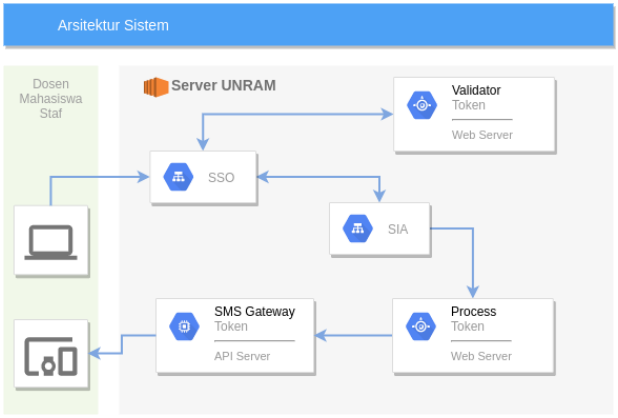
Penelitian ini merupakan tahap awal dari keseluruhan penelitian yang akan dilakukan. Pada tahun ini, penelitian berfokus pada identifikasi kebutuhan-kebutuhan yang diperlukan oleh Universitas Mataram yang kemudian akan disusun menjadi satu arsitektur yang menjelaskan informasi dan layanan apa yang akan dimasukkan, teknologi-teknologi yang digunakan, basis data yang dibutuhkan dan skema autentikasi yang dapat meningkatkan keamanan dari sisi pengguna.

Tahapan berikutnya adalah membangun sistem SMS *Gateway* yang berperan utama dalam membaca dan mengirimkan SMS kepada pengguna secara otomatis. Berikutnya adalah tahapan implementasi SMS *token* dengan sistem SSO di Universitas Mataram. Peta jalan penelitian dapat dilihat pada Gambar 1.

Gambar 1. Peta alur perancangan

A. Arsitektur Sistem

Dalam implementasinya, secara keseluruhan sistem TFA berkomunikasi dengan beberapa sistem lainnya. Dalam lingkungan kampus Universitas Mataram pengelolaan akun SSO berpusat pada Sistem Informasi Akademik (SIA), sedangkan layanan SSO sendiri berada pada *host* dan sistem yang terpisah. Kemudian terdapat tambahan sistem SMS *Gateway* yang akan membantu dalam proses pengiriman *token* melalui SMS.



Gambar 2. Arsitektur sistem

Untuk melakukan proses *login*, setiap sistem yang telah terintegrasi di Universitas Mataram akan diarahkan ke halaman SSO. Sistem SSO merupakan *interface* untuk melakukan *login* dan *reset password*. Sistem SSO juga menjamin beberapa keamanan standar yang umum terjadi, beberapa diantaranya adalah menjaga integritas isian yang dimasukkan dan memberikan token pada setiap sesi *login* untuk menjaga sistem dari serangan *brute force*.

Sistem SSO bergantung pada SIA untuk proses autentikasi, karena semua data credentials dosen dan mahasiswa tetap tersimpan di SIA. SSO berkomunikasi dengan SIA menggunakan layanan API. Untuk saat ini semua aktivitas update data pribadi dan password dilakukan pada aplikasi SIA, termasuk pengaturan untuk mengaktifkan TFA terdapat pada SIA.

Setiap terjadi proses *login*, SIA akan memeriksa pengaturan dari pengguna yang sedang melakukan login. Ketika pengguna telah berhasil ter-autentikasi (memasukkan *username* dan *password* yang benar) pada SSO, SIA akan membuka data pengaturan pengguna untuk TFA. Jika pengaturan TFA pada pengguna diaktifkan, SIA akan melakukan *generate token* untuk digunakan sebagai validator dan disimpan secara sementara pada *Session*.

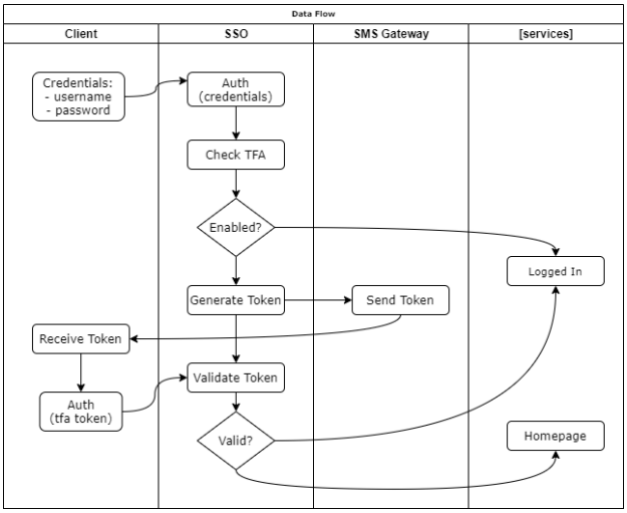
Pada saat pengguna memiliki *token* yang telah di-*generate* dan telah disimpan kedalam *Session*, berikutnya adalah SIA akan melakukan proses persiapan pengiriman *token* tersebut melalui SMS. *Token* tidak dikirim langsung oleh SIA, melainkan SIA adalah merupakan sebuah perantara. Dalam hal ini pengiriman *token* dilakukan dengan memanfaatkan sistem SMS *Gateway*, berikutnya setelah *token* berhasil dikirim melalui SMS halaman SIA pengguna akan otomatis dialihkan ke halaman TFA.

Lama waktu pengiriman token melalui SMS bergantung pada kondisi dan *provider* yang digunakan, dimana hal ini tidak dibahas dalam penelitian ini. Ketika pengguna telah menerima *token* melalui SMS, pengguna dapat memasukkan *token* yang telah diterima dari SMS tersebut pada isian TFA yang telah disediakan. *Token* yang telah dimasukkan kemudian akan dilakukan proses validasi, yakni akan dibandingkan dengan *token* yang sama yang sebelumnya telah tersimpan dalam *Session* login pengguna.

Jika *token* berhasil divalidasi maka pengguna akan diteruskan ke halaman yang dituju, seperti SIA atau sistem lain yang telah terintegrasi dengan SSO.

B. Data Flow Diagram

Secara umum alur komunikasi data dalam arsitektur TFA pada SSO melibatkan tiga komponen, yakni *client*, SSO, dan SMS *Gateway*. Pada sisi *client*, pengguna hanya perlu memasukkan *username* dan *password* akun SSO. Berikutnya SSO akan memeriksa status TFA pada akun SSO pengguna bersangkutan, jika fitur TFA diaktifkan maka SSO akan meng-*generate* sebuah *token* yang nantinya akan dikirimkan kepada pengguna melalui komponen ketiga, yakni SMS *gateway*.



Gambar 3. Data flow diagram

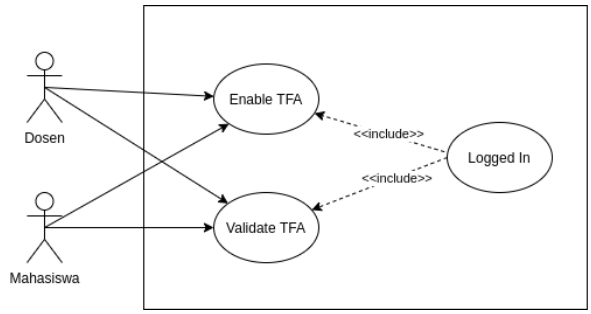
Setelah pengguna mendapatkan *token* yang telah dikirimkan melalui SMS, pengguna kemudian memasukkan *token* tersebut kedalam SSO. SSO akan melakukan validasi *token* pengguna, jika *token* pengguna berhasil divalidasi SSO kemudian akan membuat sesi *login* yang baru untuk pengguna yang akan diset pada *service* atau aplikasi yang dituju oleh pengguna.

C. Use Case Diagram

Implementasi TFA pada SSO dikondisikan untuk tidak banyak memberikan dampak terhadap aplikasi yang sudah berjalan. *Use case diagram* dalam implementasi TFA hanya membutuhkan dua fungsi baru, yakni fitur aktivasi bagi pengguna pada halaman pengaturan dan fungsi

E-ISSN:2541-0806
P-ISSN:2540-8895

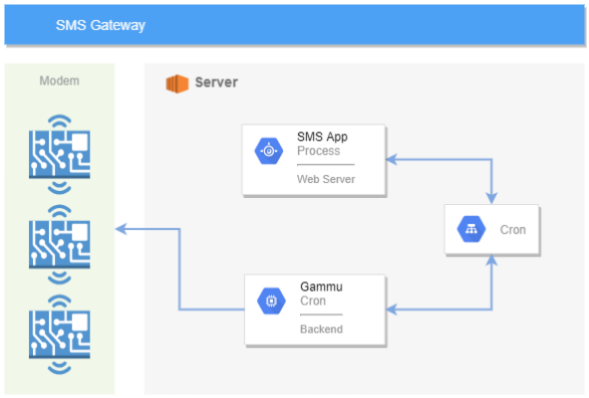
validasi *token* TFA yang telah tersimpan pada *Session* dengan *token* yang diterima pengguna melalui SMS.



Gambar 4. Use case diagram

D. SMS Gateway

Dalam penelitian ini SMS *Gateway* diperlukan untuk kebutuhan mengirim SMS. SMS *Gateway* sendiri memiliki arsitektur terpisah, sehingga secara *design* memungkinkan untuk digunakan pada berbagai keperluan pengiriman SMS.



Gambar 5. Arsitektur SMS gateway

Gambar 5 merupakan arsitektur dari SMS *Gateway*. Bagian terpenting dalam arsitektur ini adalah *Gammu*. *Gammu* merupakan *software backend* yang digunakan dalam penelitian ini agar aplikasi SMS dapat berkomunikasi dengan perangkat *modem*, dimana nantinya SMS akan diolah didalam SMS *Gateway* dan dikirimkan melalui perangkat *Modem*.

Gammu beroperasi layaknya perangkat ponsel, dapat digunakan untuk menerima dan mengirim SMS. Fungsionalitas ini dibuat ulang menjadi *software* yang dapat dijalankan pada sistem operasi. *Gammu* saat ini mendukung beberapa jenis sistem operasi, seperti *Windows* dan *Linux*. *Gammu* berkomunikasi dengan perangkat *modem* menggunakan *AT-Command* yang merupakan standar komunikasi untuk *modem* GSM.

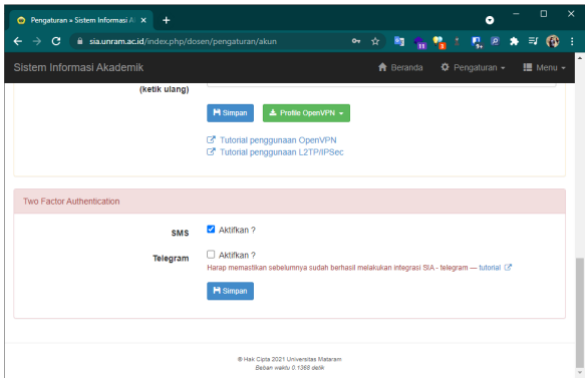
Kini pengguna dapat dengan mudah menggunakan *Gammu* untuk berkirim dan menerima SMS, salah satunya dengan menggunakan *database management system* seperti *MySQL*. Pengguna cukup menambahkan *row* kedalam *database* untuk mengirim SMS dan membaca SMS cukup dengan membuka data pada *database*.

Dalam penelitian ini aplikasi yang digunakan adalah SMS *Gateway*. Fungsionalitas utama untuk pengelolaan SMS telah disediakan dalam versi *Application Programming Interface* (API), terutama untuk pengiriman SMS. Dengan demikian SSO dengan mudah dapat memanfaatkan fasilitas pengiriman SMS menggunakan aplikasi SMS *gateway*.

IV. Hasil dan Pembahasan

A. Implementasi pada SSO

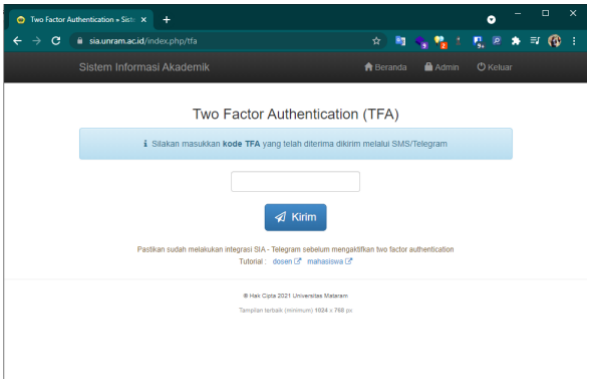
Pendekatan umum yang dapat dilakukan dalam implementasi TFA pada SSO adalah menyediakan antar muka yang mudah bagi pengguna. Pada penelitian ini fasilitas untuk mengaktifkan TFA disediakan pada halaman pengaturan yang telah ada sebelumnya pada SIA. Bagian untuk TFA sendiri dipisahkan oleh sebuah blok.



Gambar 6. Halaman pengaturan TFA

Gambar 6 adalah halaman untuk mengaktifkan TFA pada halaman pengaturan di SIA. Pengguna dapat mencentang *option* SMS untuk mengaktifkan pengiriman *token* TFA melalui SMS. Setelah memilih opsi, pengguna dapat menyimpan options tersebut dengan menekan tombol simpan.

Perubahan akan terjadi pada siklus *login* berikutnya. Untuk mengetahui fasilitas TFA telah aktif atau belum, pengguna dapat melakukan *logout* terlebih dahulu kemudian mencoba untuk *login* kembali untuk mendapati halaman TFA dan *token* SMS.



Gambar 7. Halaman mengisi token TFA

E-ISSN:2541-0806
P-ISSN:2540-8895

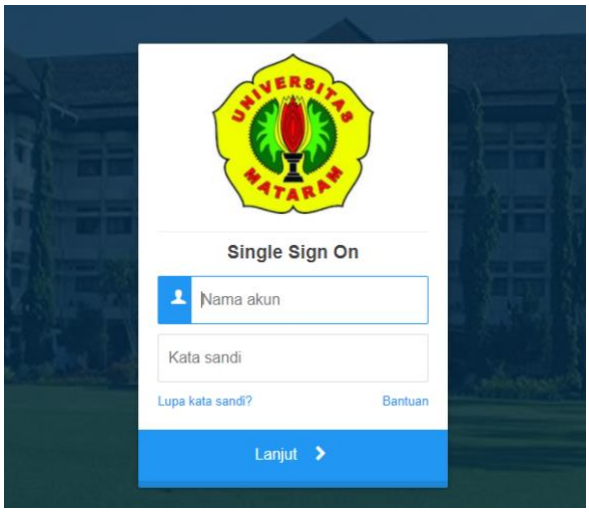
Gambar 7 adalah implementasi tampilan *token* TFA bagi pengguna yang telah mengaktifkan fitur TFA dan telah berhasil memasukkan *username* dan *password*-nya secara benar. Halaman tersebut akan selalu muncul selama pengguna belum memasukkan *token* TFA yang valid.

Token TFA bersifat acak, akan berbeda pada waktu *generate* yang satu dengan *generate* yang lainnya. *Generate token* bersifat menyeluruh untuk semua pengguna, artinya pengguna kemungkinan besar tidak akan mendapatkan *token* yang sama meskipun melakukan *generate* pada waktu yang bersamaan.

Namun demikian mekanisme validasi TFA juga memperhatikan sesi pengguna yang sedang *login*. Jika seorang pengguna dalam situasi tertentu dapat mengetahui *token* dari pengguna lainnya, maka *token* tersebut tidak dapat digunakan untuk *login* ke pengguna yang bersangkutan.

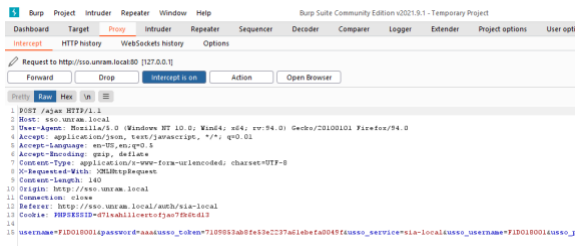
B. Analisa Serangan Brute Force

Dalam praktiknya, *tool* yang banyak digunakan untuk serangan *brute force* adalah *Burp Suite* [3]. Skenario serangan dalam penelitian ini, akan diuji percobaan *brute force* terhadap halaman login SSO.



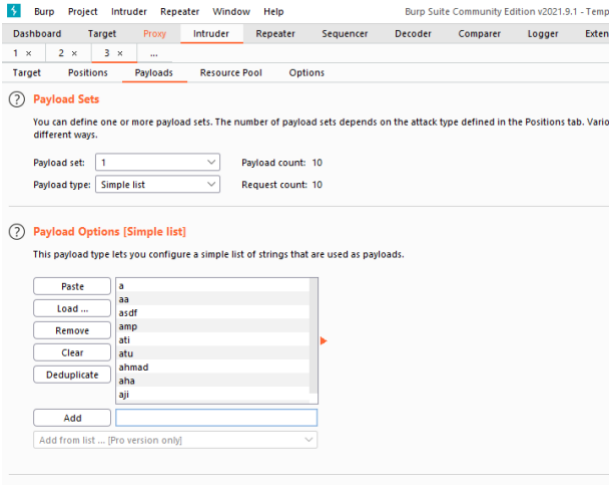
Gambar 8. Halaman login SSO

Traffic intercept adalah fitur yang dapat digunakan sebagai persiapan awal dalam melakukan serangan *brute force*. Setelah mendapatkan data-data awal seperti nama *field* dan *token* sesi jika ada, maka selanjutnya penyerang dapat membuat alur serangan.



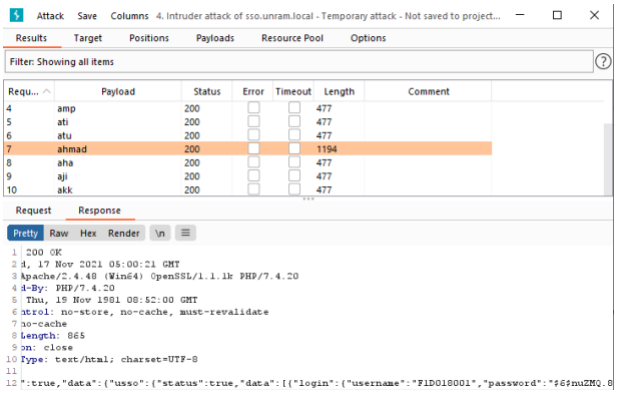
Gambar 9. Masking data awal brute force

Gambar 9 merupakan data awal yang dapat dilakukan *masking*. *Masking* dilakukan untuk dapat menjalankan banyak *payload* secara otomatis. Umumnya *masking* hanya dilakukan terhadap *password* saja, *username* sengaja dibiarkan statis karena target serangan hanya ditujukan kepada satu orang saja. Pada Gambar 9 *username* yang dijadikan target adalah NIM *F1D018001*.



Gambar 10. Daftar kata potensi password

Setelah melakukan *masking*, penyerang dapat memasukkan daftar kata yang berpotensi untuk digunakan sebagai *password*. Daftar kata ini tersedia banyak dan mudah untuk ditemukan di *internet*.



Gambar 11. Hasil dari proses attack

Setelah memasukkan daftar kata dan menjalankannya, penyerang dapat melihat hasil untuk masing-masing kata. Pada gambar diatas kata yang cocok dan menghasilkan informasi *true* adalah kata *"ahmad"* yang berarti mahasiswa *F1D018001* menggunakan *password* tersebut.

Pada posisi ini penyerang telah mengetahui *username* dan *password* pengguna, kemudian penyerang akan melakukan percobaan *login* secara *normal* langsung melalui aplikasi SSO.

E-ISSN:2541-0806
P-ISSN:2540-8895

Gambar 12. Halaman TFA setelah berhasil login

Gambar 12 adalah tampilan halaman setelah penyerang memasukkan *username* dan *password* yang benar untuk pengguna NIM *F1D018001*. Dapat terlihat TFA melindungi akses terhadap data pengguna meskipun dengan suatu metode serangan, akun pengguna telah jatuh kepada penyerang.

C. Analisa Pengalaman Pengguna

Pengembangan fitur sistem dapat mempengaruhi dua aspek, yakni sisi teknis dan non-teknis. Sisi non-teknis dalam hal ini adalah pengalaman pengguna (*user experience*) atau yang umum disingkat dengan UX. Tingkat kepuasan pengguna dalam hal penggunaan aplikasi dapat diukur menggunakan *User Experience Questionaier* (UEQ) [9], [10]. UEQ dapat mengukur tingkat efisiensi, efektivitas, dan kepuasa pengguna terhadap aplikasi atau fitur dalam suatu aplikasi.

Dalam UEQ penilaian dibagi menjadi dua aspek, yakni pragmatis dan hedonis. Aspek kualitas pragmatis adalah persepsi terhadap hal teknis, seperti tampilan, fitur, dan efisiensi. Sedangkan aspek kualitas hedonis cenderung bersifat emosi pengguna, seperti stimulasi untuk menggunakan aplikasi atau fitur dan pengaruh emosi lainnya [15]. Skala penilaian UEQ dapat dilihat pada Tabel I.

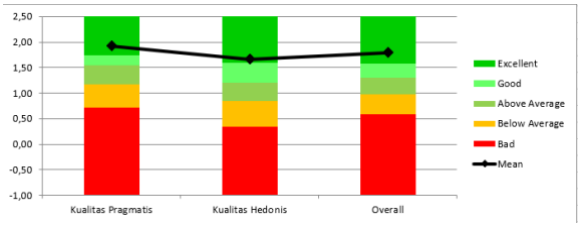
TABEL I. SKALA PENILAIAN UEQ

No	Category	Value	Scale
1	*Dependability*	menghalangi/ mendukung	Pragmatis
2	*Perspicuity*	rumit/ sederhana	Pragmatis
3	*Efficiency*	tidak efisien/ efisien	Pragmatis
4	*Perspicuity*	membingungkan/ jelas	Pragmatis
5	*Stimulation*	membosankan/ mengasyikkan	Hedonis
6	*Stimulation*	tidak menarik/ menarik	Hedonis
7	*Novelty*	konvensional/ berdaya cipta	Hedonis
8	*Novelty*	lazim/ terdepan	Hedonis

Pengukuran UEQ menggunakan 8 item dengan masing-masing hasil penilaian dari 1 sampai 7. Penelitian ini menggunakan formulir UEQ untuk mengetahui pengaruh implementasi TFA terhadap SSO kepada 24 responden yang terdiri dari Dosen dan Mahasiswa. Data Raw hasil pengujian UEQ dapat dilihat pada Tabel II.

TABEL II. DATA RAW HASIL PENGUJIAN UEQ

Items							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
7	7	7	7	7	7	7	7
6	6	7	7	4	5	7	7
7	7	7	7	6	7	7	7
7	7	7	7	7	7	7	7
6	6	7	6	5	6	6	7
7	7	7	7	6	6	7	6
7	7	7	7	7	7	7	7
6	5	6	6	5	5	6	6
5	3	4	6	4	5	4	4
7	7	7	7	6	6	5	6
6	3	3	6	3	4	3	3
6	5	7	6	5	6	7	7
6	5	7	5	5	5	6	7
7	7	7	7	7	7	7	7
7	6	6	7	4	6	7	7
6	5	6	7	7	7	6	7
6	5	5	5	5	6	6	7
7	4	3	7	4	6	5	1
5	6	5	6	4	4	6	6
7	7	7	7	7	7	7	7
6	2	3	6	4	4	4	2
5	2	4	5	4	3	4	3
5	5	6	3	3	4	5	5
6	5	5	6	6	6	7	7



Gambar 13. Benchmark hasil perhitungan UEQ

Gambar 13 merupakan *benchmark* perhitungan penilaian dari UEQ yang telah disediakan. Masing-masing kualitas aspek memiliki skala yang bervariasi, sesuai dengan data yang tersedia. Secara umum hasil penilaian akhir dari dua skala aspek UEQ mendapatkan nilai *excellent*.

TABEL III. HASIL PERHITUNGAN DARI SKALA ASPEK PRAGMATIS DAN HEDONIS

Scale	Mean	Benchmark
Pragmatis	1,927083333	Excellent
Hedonis	1,666666667	Excellent
Overall	1,80	Excellent

Tabel III merupakan hasil perhitungan dari skala aspek pragmatis dan hedonis. Penilaian dari skala aspek pragmatis memiliki nilai tertinggi yakni sebanyak 1,93. Tingginya nilai dari aspek pragmatis didukung oleh dua kategori penilaian. Pertama adalah *Dependability*, yakni pengguna merasa fitur ini mampu mendukung pemanfaatan layanan yang sudah ada. Kemudian pengaruh yang kedua adalah *Perspicuity* dalam hal penilaian terhadap kejelasan alur fitur yang diberikan, sehingga pengguna tidak merasa kesulitan dalam menggunakan sistem.

Kualitas aspek hedonis mendapatkan nilai 1,67 dan masih tergolong dalam kategori *excellent*. Nilai dari aspek hedonis cenderung lebih rendah dibandingkan dengan nilai dari aspek pragmatis. Faktor penilaian yang mempengaruhi nilai dari aspek hedonis dalam penelitian ini adalah *Stimulation*, yakni pengguna tidak cukup termotivasi dalam menggunakan fitur TFA pada SSO. Faktor dalam aspek hedonis yang memberikan porsi penilaian yang besar adalah *Novelty*, yakni dari sisi inovasi.

Implementasi TFA pada masa ini belum banyak diimplementasikan, diakibatkan oleh kurangnya kesadaran terhadap keamanan informasi. Sehingga dengan tersedianya TFA pada layanan SSO, pengguna dapat merasa lebih aman untuk beraktivitas di dalam sistem.

V. KESIMPULAN

1. Pengujian simulasi serangan *brute force* dengan mengaktifkan fitur TFA pada SSO menunjukkan serangan tidak dapat menembus akun SSO, meskipun *username* dan *password* akun sudah diketahui oleh penyerang.
2. Pengujian pengalaman pengguna menggunakan UEQ secara keseluruhan mendapatkan hasil 1,797 atau berada dalam kategori *excelent*.
3. Dalam aspek pragmatis pengujian mendapatkan hasil paling tinggi yakni 1,927, kategori penilaian yang berkontribusi besar dalam porsi nilai adalah *Dependability* dan *Perspicuity*. Hal ini menunjukkan pengguna merasa fitur ini mampu mendukung pemanfaatan layanan yang sudah ada dan fitur TFA dalam SSO dirasa memiliki alur yang cukup jelas.
4. Dalam aspek hedonis pengujian mendapatkan hasil lebih kecil dibandingkan aspek pragmatis yakni 1,667, namun hasil ini masih dalam kategori *excellent*. Kategori yang memiliki nilai terendah adalah *Stimulation*, yakni pengguna masih belum cukup termotivasi untuk memanfaatkan TFA dalam SSO.

REFERENSI

[1] J. Costa and A. Michalas, "Middle Man: An Efficient Two-Factor Authentication Framework," 2017 Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2017, pp. 1–7, 2018, doi: 10.1109/ICCUBEA.2017.8463686.

[2] K. S. M. Moe and T. Win, "Improved hashing and honey-based stronger password prevention against brute force attack," 2017 Int. Symp. Electron. Smart Devices, ISESD 2017, vol. 2018-January, pp. 1–5, 2017, doi: 10.1109/ISESD.2017.8253295.

[3] R. Vibhandik and A. Kumar Bose, "Vulnerability assessment of web applications - a testing approach," 2015 Forth Int. Conf. e-Technologies Networks Dev., pp. 16–21, 2015.

[4] S. Binu, M. Misbahuddin, and P. Raj, "A Single Sign on based secure remote user authentication scheme for Multi-Server Environments," Int. Conf. Comput. Commun. Technol. ICCCT 2014, 2014, doi: 10.1109/ICCCT2.2014.7066715.

[5] N. R. Chakraborty, M. T. Rahman, M. E. Rahman, and M. S. Uddin, "Generation and verification of digital signature with two factor authentication," IWCI 2016 - 2016 Int. Work. Comput. Intell., no. December, pp. 131–135, 2017, doi: 10.1109/IWCI.2016.7860353.

[6] S. Indu, T. N. Sathya, and V. Saravana Kumar, "A stand-alone and SMS-based approach for authentication using mobile phone," 2013 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2013, pp. 140–145, 2013, doi: 10.1109/ICICES.2013.6508205.

[7] E. Sediyono, K. I. Santoso, and Suhartono, "Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS," Proc. 2013 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2013, pp. 1604–1608, 2013, doi: 10.1109/ICACCI.2013.6637420.

[8] H. Liu and Y. Zhang, "An improved one-time password authentication scheme," Int. Conf. Commun. Technol. Proceedings, ICCT, pp. 1–5, 2013, doi: 10.1109/ICCT.2013.6820340.

[9] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Construction of a Benchmark for the User Experience Questionnaire (UEQ)," Int. J. Interact. Multimed. Artif. Intell., vol. 4, no. 4, p. 40, 2017, doi: 10.9781/ijimai.2017.445.

[10] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S)," Int. J. Interact. Multimed. Artif. Intell., vol. 4, no. 6, p. 103, 2017, doi: 10.9781/ijimai.2017.09.001.

[11] A. Derhab, M. Belaoued, M. Guerroumi, and F. A. Khan, "Two-Factor Mutual Authentication Offloading for Mobile Cloud Computing," IEEE Access, vol. 8, pp. 28956–28969, 2020, doi: 10.1109/ACCESS.2020.2971024.

[12] G. R. Haron, D. Maniam, L. Mat Nen, and N. I. Daud, "User behaviour and interactions for multimodal

authentication," 2016 14th Annu. Conf. Privacy, Secur. Trust. PST 2016, pp. 309–316, 2016, doi: 10.1109/PST.2016.7906979.

[13] W. Bin Hsieh and J. S. Leu, "Design of a time and location based One-Time Password authentication scheme," IWCMC 2011 - 7th Int. Wirel. Commun. Mob. Comput. Conf., pp. 201–206, 2011, doi: 10.1109/IWCMC.2011.5982418.

[14] C. Thammarat, W. Kurutach, and S. Phoomvuthisarn, "A secure lightweight protocol for SMS mutual authentication protocols based on symmetic-key encryption," 2017 17th Int. Symp. Commun. Inf. Technol. Isc. 2017, vol. 2018-Janua, pp. 1–6, 2017, doi: 10.1109/ISCIT.2017.8261196.

[15] V. Intanny et al., "Pengukuran Kebergunaan dan Pengalaman Pengguna Marketplace Jogjaplaza.id dengan Metode UEQ dan USE Questionnaire Measuring Usability and User Experience of The Marketplace of Jogjaplaza.id Using UEQ and USE Questionnaire," J. Pekommas, vol. 3, no. 2, pp. 117–126, 2018.