

**IMPLEMENTASI *TWO FACTOR AUTHENTICATION* DAN**

**ALGORITMA RSA SEBAGAI METODE OTENTIKASI**

**LOGIN PADA SI-ABKA (SISTEM AMAL BAKTI**

**KEMENTERIAN AGAMA)**

**SKRIPSI**

oleh

**Ahmad Choirul Mustaqim**

**NIM 152410101155**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2020**



**IMPLEMENTASI *TWO FACTOR AUTHENTICATION* DAN**

**ALGORITMA RSA SEBAGAI METODE OTENTIKASI**

**LOGIN PADA SI-ABKA (SISTEM AMAL BAKTI**

**KEMENTERIAN AGAMA)**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Komputer

oleh

**Ahmad Choirul Mustaqim**

**NIM 152410101155**

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS JEMBER**

**2020**

# PERSEMBAHAN

**MOTTO**

# PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Choirul Mustaqim

NIM : 152410101155

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi *Two factor authentication* Dan Algoritma Rsa Sebagai Metode Otentikasi Login Pada Si-Abka (Sistem Amal Bakti Kementerian Agama)”adalah benar-benar hasil karya saya sendiri, kecuali jika ada pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada instansi manapun, dan bukti karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika dikemudian hari pernyataan ini tidak benar.

Jember, 27 Januari 2019

Yang menyatakan,

Ahmad Choirul Mustaqim

NIM 152410101155

**SKRIPSI**

**Implementasi *Two factor authentication* Dan Algoritma Rsa**

**Sebagai Metode Otentikasi Login Pada Si-Abka**

**(Sistem Amal Bakti Kementerian Agama)**

oleh

**Ahmad Choirul Mustaqim**

**NIM 152410101155**

Pembimbing

Pembimbing Utama : Achmad Maududie S.Kom,M.Sc.

Pembimbing Anggota : Diksy Media Firmansyah S.Kom, M.Kom.

**PENGESAHAN PEMBIMBING**

Skripsi berjudul “Implementasi *Two factor authentication* Dan Algoritma Rsa Sebagai Metode Otentikasi Login Pada Si-Abka (Sistem Amal Bakti Kementerian Agama)”, telah diuji dan disahkan pada:

Hari tanggal : Jumat, 27 Januari 2020

Tempat : Program Studi Sistem Informasi Universitas Jember

|  |  |
| --- | --- |
| Disetujui oleh: | |
| Pembimbing I, | Pembimbing II, |
|  |  |
| Achmad Maududie, ST., M.Sc  NIP 197004221995121001 | Diksy Media Firmansyah S.Kom, M.Kom.  NIP 760016853 |

**PENGESAHAN PENGUJI**

Skripsi berjudul “Implementasi *Two factor authentication* Dan Algoritma Rsa Sebagai Metode Otentikasi Login Pada Si-Abka (Sistem Amal Bakti Kementerian Agama)”, telah diuji dan disahkan pada:

Hari tanggal : Jumat, 27 Januari 2020

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

|  |  |
| --- | --- |
| Pembimbing I, | Pembimbing II, |

Mengesahkan

Dekan Fakultas Ilmu Komputer,

Prof. Saiful Bukhori,ST., M.Kom

NIP. 196811131994121001

# RINGKASAN

SI-Abka (Sistem Amal Bakti Kementerian Agama Jember) merupakan sistem yang mengelola data koperasi dari seluruh anggota yang bekerja di bawah instansi Kementerian Agama Jember. Untuk mengamankan data pada sistem SI-Abka dibutuhkan proses otentikasi. Proses otentikasi yang sebelumnya hanya mengandalkan username dan password perlu adanya faktor yang disebut dengan two factor authentication dan menggunakan metode Time based one time password. pemanfaatan TOTP diharapkan dapat menambah tingkat keamanan dari sistem otentikasi di SI-Abka. Hasil output dari TOTP berupa 6 digit angka plaintkes sehingga membutuhkan algoritma enkripsi khusus seperti algoritma RSA. Algoritma RSA memiliki kelebihan yaitu proses enkripsi dan dekripsi menggunakan kunci asimetris yang lebih aman dibandingkan kunci simetris. Untuk mengetahui tingkat keamanan implementasi TOTP dan algoritma RSA maka dilakukan tes keamanan sistem dengan metode brute force dan Man in the midle (MITM).

Percobaan tes keamanan sistem dengan teknik brute force menghasilkan dari satu juta kombinasi. pada percobaan kali ini hanya dapat mengirimkan 44 kombinasi dan Presentase kemungkinan tertebaknya kode OTP dalam percobaan tersebut hanya sebesar 0.044 % . Percobaan dengan teknik man in the middle berhasil mendapatkan data yang dikirim oleh client ke server. Data yang didapat berupa nomor KTA, password, dan kode OTP yang valid. Dari ketiga data tersebut kode OTP berhasil di enkripsi dan tidak dapat dibaca oleh peneliti. Dari percobaan tersebut keamanan data yang dikirim oleh client ke server bisa dikatakan aman.

# PRAKATA

# DAFTAR ISI

[PERSEMBAHAN iii](#_Toc29869439)

[PERNYATAAN v](#_Toc29869440)

[PENGESAHAN PEMBIMBING vii](#_Toc29869441)

[PENGESAHAN PENGUJI viii](#_Toc29869442)

[RINGKASAN ix](#_Toc29869443)

[PRAKATA x](#_Toc29869444)

[DAFTAR ISI xi](#_Toc29869445)

[DAFTAR GAMBAR xiii](#_Toc29869446)

[DAFTAR TABEL xv](#_Toc29869447)

[BAB 1. PENDAHULUAN 1](#_Toc29869448)

[1.1 Latar belakang 1](#_Toc29869449)

[1.2 Rumusan masalah 3](#_Toc29869450)

[1.3 Batasan masalah 3](#_Toc29869451)

[1.4 Tujuan penelitian 3](#_Toc29869452)

[1.5 Manfaat Penelitian 4](#_Toc29869453)

[BAB 2. TINJAUAN PUSTAKA 5](#_Toc29869454)

[2.1 *Password* dan *Username* 5](#_Toc29869457)

[2.2 Otentikasi 6](#_Toc29869458)

[2.3 SI-Abka 7](#_Toc29869459)

[2.4 QR-Code 8](#_Toc29869460)

[2.5 *One Time Password* (OTP) 9](#_Toc29869461)

[2.6 Time-Based OTP 11](#_Toc29869462)

[2.7 Algoritma RSA 14](#_Toc29869463)

[2.8 Brute Force 18](#_Toc29869464)

[2.9 Man in the middle 19](#_Toc29869465)

[2.10 Penelitian Terdahulu 20](#_Toc29869466)

[BAB 3. METODOLOGI PENELITIAN 22](#_Toc29869467)

[3.1 Objek Penelitian 22](#_Toc29869468)

[3.2 Tempat Penelitian 22](#_Toc29869469)

[3.3 Tahapan Penelitian 22](#_Toc29869470)

[3.4 Pembuatan modul TOTP dan RSA 23](#_Toc29869471)

[3.4.1. Modul TOTP 23](#_Toc29869472)

[3.4.2. Modul RSA 26](#_Toc29869473)

[3.5 Uji keamanan 28](#_Toc29869474)

[3.5.1. Uji brute force 28](#_Toc29869484)

[3.5.2. Uji MITM 31](#_Toc29869485)

[BAB 4. HASIL DAN PEMBAHASAN 33](#_Toc29869486)

[4.1 Hasil Implementasi Pembuatan Modul TOTP dan RSA 33](#_Toc29869492)

[4.1.1. Modul TOTP 33](#_Toc29869493)

[4.1.2. Modul RSA 34](#_Toc29869494)

[4.2 Hasil Pengujian modul TOTP dan Algoritma RSA 37](#_Toc29869495)

[4.3 Hasil pengujian uji keamanan 44](#_Toc29869496)

[4.3.1. Pengujian brute force 44](#_Toc29869497)

[4.3.2. Pengujian MITM 47](#_Toc29869498)

[4.4 Pembahasan 52](#_Toc29869499)

[BAB 5. PENUTUP 53](#_Toc29869500)

[5.1 Kesimpulan 53](#_Toc29869507)

[5.2 Saran 54](#_Toc29869508)

[Daftar pustaka 55](#_Toc29869509)

[LAMPIRAN 56](#_Toc29869510)

# DAFTAR GAMBAR

[Gambar 2.1.proses otentikasi 6](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29891992)

[Gambar 2.2. tampilan login SI-Abka 7](#_Toc29891993)

[Gambar 2.3. QR-Code 9](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29891994)

[Gambar 2.4. alur OTP 10](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29891995)

[Gambar 2.5. alur generate kode TOTP 13](#_Toc29891996)

[Gambar 2.6. proses login dengan OTP 13](#_Toc29891997)

[Gambar 2.7. konsep RSA 14](#_Toc29891998)

[Gambar 2.8.Flowchart Proses Algoritma RSA 15](#_Toc29891999)

[Gambar 2.9. percobaan brute force 18](#_Toc29892000)

[Gambar 2.10. cara kerja MITM 19](#_Toc29892001)

[Gambar 3.1 Alur Tahapan Penelitian 23](#_Toc29892002)

[Gambar 3.2.. flowchart implementasi secret key TOTP 26](#_Toc29892003)

[Gambar 3.3. alur login TOTP dan RSA 27](#_Toc29892004)

[Gambar 3.4. tampilan pembacaan request header 30](#_Toc29892005)

[Gambar 4.1. halaman pembuatan secret key kode TOTP 33](#_Toc29892006)

[Gambar 4.2. hasil pembangkitan secret key 34](#_Toc29892007)

[Gambar 4.3. hasil pembangkitan secret key , private key dan public key 35](#_Toc29892008)

[Gambar 4.4. tampilan login SI-Abka 37](#_Toc29892009)

[Gambar 4.5. input kode TOTP 38](#_Toc29892010)

[Gambar 4.6. input kode TOTP dengan qr-code 39](#_Toc29892011)

[Gambar 4.7. aplikasi ChoiTOTP 39](#_Toc29892012)

[Gambar 4.8. menu aplikasi ChoiTOTP 40](#_Toc29892013)

[Gambar 4.9. tampilan scan qr-code 41](#_Toc29892014)

[Gambar 4.10.kode TOTP berhasil didapatkan 41](#_Toc29892015)

[Gambar 4.11. memasukan kode TOTP 42](#_Toc29892016)

[Gambar 4.12.dasboard user 42](#_Toc29892017)

[Gambar 4.13. dashboard admin 43](#_Toc29892018)

[Gambar 4.14. pemberitahuan kode OTP salah 43](#_Toc29892019)

[Gambar 4.15. penentuan target sistem 44](#_Toc29892020)

[Gambar 4.16. variabel header 44](#_Toc29892021)

[Gambar 4.17. variabel payload brute force 45](#_Toc29892022)

[Gambar 4.18. hasil brute force 46](#_Toc29892023)

[Gambar 4.19. respons code OTP benar 46](#_Toc29892024)

[Gambar 4.20. proses scan ip target 47](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29892025)

[Gambar 4.21. arp poisoning ip 48](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29892026)

[Gambar 4.22. routing ip target 49](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29892027)

[Gambar 4.23. pembelokan traffic https ke http 49](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29892028)

[Gambar 4.24. hasil pembacaan MITM 50](file:///D:\kuliah\skripsi\file%20skripsi\skripsi%204.docx#_Toc29892029)

# DAFTAR TABEL

# PENDAHULUAN

Bab ini menjelaskan hal-hal yang berkaitan dengan pendahuluan penelitian. Adapun pembahasan pada bab ini meliputi latar belakang, rumusan masalah, tujuan dan manfaat, serta batasan masalah.

## Latar belakang

SI-Abka (Sistem Amal Bakti Kementerian Agama Jember) merupakan sistem yang mengelola data koperasi dari seluruh anggota yang bekerja di bawah instansi Kementerian Agama Jember. Sistem ini berfungsi sebagai pengelola data mulai dari data anggota, data simpanan, sampai data pinjaman. Data-data tersebut bersifat penting dikarenakan menyangkut data keuangan anggota. Untuk mengamankan data pada sistem SI-Abka dibutuhkan proses otentikasi.

Otentikasi adalah suatu proses atau tindakan untuk membuktikan apakah suatu kegiatan atau orang tersebut bersifat benar, asli, atau valid. Proses atau orang yang telah divalidasi akan mendapatkan akses ke dalam sistem (Khairina, 2011). Metode otentikasi konvensional yang selama ini familiar di gunakan adalah menggunakan kombinasi username dan password atau biasa juga disebut dengan metode *single factor authentication*. Meskipun penggunaan *single factor authentication*  sering digunakan pada otentikasi sistem secara umum, tetapi cara tersebut memiliki kelemahan yaitu ketika username dan password diketahui oleh pihak yang tidak bertanggung jawab akun tersebut dapat digunakan untuk masuk kedalam sistem.

Kelemahan metode *single factor authentication* dapat diatasi dengan metode *two factor authentication*. *Two factor authentication* merupakan faktor tambahan yang harus dilewati setelah memasukan username dan password. Salah satu contoh *two factor authentication* adalah *timed based one time password* (TOTP), yaitu kode pembangkitan OTP (*one time password*) berdasarkan waktu dan *secret key*. TOTP ini dapat di akses dengan menggunakan software atau hardware khusus (Sudiarto Raharjo dkk., 2017). Kelebihan TOTP adalah tidak mengandalkan server saat pembangkitan kode OTP sehingga menghilangkan kemungkinan tersampaikan dalam waktu yang lama dan tidak perlu adanya penyimpanan kode OTP ke dalam database.

Proses pembangkitan kode TOTP menghasilkan kode plainteks, hal ini memiliki celah untuk disadap saat proses pengiriman dari pengguna menuju server. Untuk mengatasi permasalahan tersebut kode TOTP sebaiknya dienkripsi terlebih dahulu sebelum dikirim. Salah satu Metode enkripsi yang dapat digunakan adalah algoritma RSA yang memiliki kelebihan yaitu proses enkripsi dan dekripsi menggunakan kunci asimetris yang lebih aman dibandingkan kunci simetris (Susanto dan Trisusilo, 2018). Oleh karena itu hanya server yang memiliki private key yang bisa membaca data yang telah dienkripsi.

Mengacu dari kelebihan TOTP dan RSA peneliti akan menerapkan kedua metode tersebut untuk melakukan proses pengamanan otentikasi. Proses pengamanan ini akan diuji menggunakan teknik brute force dan MITM untuk mengetahui tingkat keamanan sistem SI-Abka. Hasil pengamanan tersebut diharapkan dapat menambah tingkat keamanan sistem SI-Abka.

## Rumusan masalah

Berdasarkan latar belakang masalah penelitian, maka muncul rumusan masalah sebagai berikut.

1. Bagaimana meningkatkan keamanan pada proses otentikasi SI-ABKA dengan menggunakan metode TOTP dan algoritma RSA?
2. Bagaimana mengukur tingkat keamanan sistem SI-ABKA?

## Batasan masalah

Peneliti memberikan batasan masalah untuk objek dan tema yang dibahas sehingga tidak terjadi penyimpangan dalam proses penelitian dan menganalisis

* + 1. Kunci yang digunakan dalam algoritma RSA menggunakan bilangan prima dengan batas antara 100-1000 agar tidak mudah tertebak dan agar proses dekripsi tidak membutuhkan waktu yang lama.
    2. Aplikasi pembangkit kode TOTP yang dikembangkan berbasis android.
    3. Teknik yang digunakan untuk uji keamanan token TOTP adalah *brute force* dan *man in the midle* (MITM).
    4. Perubahan kode TOTP setiap 30 detik.

## Tujuan penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah:

* + 1. Meningkatkan keamanan transaksi pada SI-ABKA dengan menggunakan *two factor authentication* dengan menggunakan metode TOTP dan algoritma RSA*.*
    2. Mengetahui peningkatan keamanan otentikasi SI-ABKA yang menggunakan *two factor authentication* dengan metode TOTP dan algoritma RSA.

## Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Bagi Akademis

Hasil penelitian yang dilakukan dapat membuktikan adanya peningkatan keamanan pada proses otentikasi yang menerapkan metode TOTP dan algoritma RSA.

1. Bagi Peneliti

Dapat mengetahui dan menerapkan metode TOTP dan algoritma RSA untuk meningkatkan keamanan pada sebuah sistem.

1. Bagi Objek Penelitian

Penelitian yang dilakukan diharapkan dapat memberikan peningkatan keamanan terhadap otentikasi sistem SI-Abka.

# TINJAUAN PUSTAKA

Bagian ini memaparkan tinjauan yang berkaitan dengan masalah yang dibahas, serta kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori yang di dapatkan berupa pembangkitan OTP dan penerapanya yang dapat membantu peneliti dalam penelitian ini. Selain pembangkitan OTP penulis juga memelajari algoritma RSA. Perhitungan yang di dapatkan akan membantu peneliti dalam menghitung kode OTP yang akan di *generate* secara berkala oleh server dan *client.* Hasil perhitungan akan di proses oleh *client* dan server yang akan digunakan untuk proses otentikasi OTP di sistem SI-ABKA.



## Penelitian Terdahulu

Penelitian dengan judul “Implementasi Algoritma Time-Based *One time password* Dalam Otentikasi Token Internet Banking” (Ungkawa dkk., tanpa tahun) berisi tentang . Penelitian ini melakukan penerapan TOTP dalam pembangkitan token OTP nya. Token tersebut tidak langsung dikirim ke user tetapi mengirim nilai hash nya. Penelitian ini menggunakan hash SHA256 sebagai metode hashingnya dan emkripsi AES. Penelitian ini diaplikasikan pada sistem internet banking di mana antara token virtual dan server dipasang algoritma TOTP untuk menghasilkan password sebagai otentikasi tambahan . Dari hasil pengujian yang dilakukan bahwa password OTP tidak muncul secara berulang dan *secret key* yang dihasilkan secara acak juga tidak muncul secara berulang tetapi mempunyai presentase kemiripan tertinggi sebesar 0,03%.

Penelitian dengan judul “Aplikasi Algoritma RSA untuk Keamanan Data pada Sistem Informasi Berbasis Web” (Rosnawan, 2011). Untuk menjaga keamanan dari password dan pesan berupa file, biasanya digunakan teknik enkripsi agar kerahasiaan data terebut terjamin. Salah satu algoritma enkripsi yang sering digunakan adalah algoritma RSA. Pada kesempatan ini penulis tertarik mengkaji tentang aplikasi pengamanan data pada sistem informasi berbasis web. Permasalahan dalam skripsi ini adalah bagaimana implementasi algoritma RSA untuk keamanan data pada sistem informasi berbasis web.

Penelitian dengan judul “Implementasi Algoritma RSA Untuk Enkripsi Dan Dekripsi Sms (*Short Message Service*) Pada Ponsel Berbasis Android” (Sardju dkk., 2015). Penelitian ini membahas tentang keamanan dalam penggunaan servis sms. Peneliti mengamankan pesan sms dengan menggunakan algoritma RSA. Hasil keluaran dari sistem ini yaitu pada pengiriman sms yang telah terenkripsi akan terkirim apabila ≤ 160 karakter, dan sms tidak akan terkirim apabila ≥ 160 karakter, pada proses enkripsi dan dekripsi membutuhkan waktu rata-rata 0,18 detik, pada pengujian *avalanche effect* dengan menggunakan masukan plaintext yang berbeda tiap percobaan akan menghasilkan chipert ext yang berbeda dengan presentase rata-rata sebesar 10,35 %, sedangkan pada pengujian *brute force* membutuhkan waktu selama 1,652 x tahun untuk mencoba semua kemungkinan kunci yang ada.

Penelitian - penelitian di atas dapat disimpulkan bahwa *two factor autentication* dan algoritma RSA sesuai untuk mengamankan fungsi login di sistem SI-Abka. Diharapkan dengan penelitian ini keamanan transaksi di sistem tersebut lebih tinggi lagi dan aman. Penggunaan TOTP pada sistem otentikasi di SI-Abka dapat membantu sistem untuk memenuhi dua dari 3 syarat standar keamanan sistem yaitu integritas, dan ketersediaan (Sahu dkk., 2014). TOTP berperan sebagai pintu kedua setelah user melakukan login atau saat user melakukan suatu aktifitas yang membutuhkan pengecekan kepemillikan akun. Hal ini dikarenakan saat user malakukan sesuatu maka user perlu memasukan kode khusus yang tidak diketahui oleh sembarang orang.

Sedangkan metode RSA memenuhi syarat terakhir yaitu kerahasiaan data. RSA berperan dalam melakukan enkripsi data yang dikirimkan oleh user ke server tanpa perlu khawatir data tersebut dilihat atau dicuri oleh orang lain. Proses enkripsi RSA yang memakai sistem public key dan private key sangat cocok pada kasus ini. Hal ini dikarenakan data yang telah di enkripsi dengan public key hanya dapat di buka dengan pasangan private key nya. Dari cara kerja tersebut sistem dapat dengan aman mengirimkan public key tanpa takut data yang di enkripsi dengan public key tersebut di dekripsi.

Kedua metode terebut digabungkan untuk menambahkan tingakt keamanan sistem yang telah ada. Hal ini ditujukan agar data akun yang ada dapat lebih aman dari penggunaan orang yang tidak berkepentingan. Alasan penggunaan metode TOTP adalah resource yang dibutuhkan sedikit dan kecil. TOTP tidak membutuhkan server khusus dan tidak membutuhkan biaya sama sekali dalam pengimplementasianya. Sedangkan metode RSA digunakan untuk melakukan enkripsi data yang dikirim agar data yang ada dapat dienkripsi. Hasil enkripsi hanya bisa didekripsi oleh server tanpa takut dibaca pihak lain.

Untuk mengetahui tingkat keamanan sistem yang menggunakan TOTP dan RSA dibutuhkan percobaan penyerangan. Percobaan penyerangan pada penelitian kali ini menggunakan teknik brute force dan MITM (Man Iin the Middle). Teknik brute force sesuai karena teknik ini menggunakan metode dictionary attack yaitu melakukan penyerangan dengan daftar tertentu.daftar terebut berupa kombinasi 6 digit angka antara 0000-9999. Dari kombinasi tersebut terdapat satu ombinasi yang benar. Dari hal tersebut teknik brute force sangat sesuai karena teknik tersebut melakukan percobaan secara terus menerus sampai mencapai tujuannya. Tujuan tersebut adalah ditemukanya kombinasi yang sesuai.

Teknik kedua adalah MITM atau man in the middle. Teknik ini memungkinkan data yang dikirim atau diterima oleh pengguna dapat dibaca dan dimanipulasi oleh penyerang.

## SI-Abka

Sistem informasi amal bakti kementerian agama (SI-abka) merupakan sistem web yang membantu koperasi amal bakti kementerian agama jember dalam melakukan transaksi. Sistem ini menangani data dan informasi anggota, transaksi simpan pinjam, sampai menangani rekap pembayaran cicilan oleh bendahara di tiap satuan kerja. Tampilan sistem login SI-Abka dapat dilihat pada Gambar 2.2.



Gambar 2.2. tampilan login SI-Abka

Sistem SI-Abka menggunakan metode otentikasi umum yaitu menggunakan username dan password. Username dalam SI-Abka menggunakan KTA. KTA merupakan kartu tanda anggota yang menandakan keanggotaan pseserta di koperasi amal bakti. Saat sistem menggunakan KTA dan password sebagai metode otentikasinya maka hanya mempunyai 4 kombinasi masukan dari user seperti pada Tabel 2.1

|  |  |  |
| --- | --- | --- |
| **Username** | **Password** | **Status** |
| Salah | Salah | Gagal |
| Salah | Benar | Gagal |
| Benar | Salah | Gagal |
| Benar | Benar | **Berhasil** |

Tabel 2.1. Tabel kombinasi login SI-Abka

SI-Abka memiliki 3 level user antara lain admin, bagian keuangan dan nasabah. Ketiga level tersebut memiliki akses yang berbeda-beda, nasabah dapat melakukan proses pengajuan pinjaman, melihat status pinjaman nya, melihat berita dan lain-lain. Bagian keuangan bertugas sebagai reviewer dan validator pinjaman yaitu bertugas untuk melihat apakah peminjam layak mendapatkan pinjaman, dan melakukan proses pinjaman. Sedangkan admin bertugas sebagai pengatur sistem dana luar kerja. Akses admin antara lain untuk mengatur berita, jenis pinjaman, mengatur daftar akun dan lain-lain.

SI-Abka merupakan sistem berbasis website dengan bahasa pemrograman menggunakan PHP dan database menggunakan mysql. SI-Abka dapat diakses menggunakan browser baik melalui pc ataupun smartphone. Sehingga sistem tersebut dapat di akses dimana saja dan kapan saja.

## *Password* dan *Username*

Password adalah kumpulan karakter yang digunakan untuk masuk kedalah suatu sistem dan bersifat rahasia. Password atau kata sandi dapat digunakan untuk layanan otentikasi, yaitu layanan yang berhubungan identifikasi, baik mengidentifikasi kebenaran pihak – pihak yang berkomunikasi ( *user authentication* atau *entity authentication* ) maupun mengidentifikasi kebenaran sumber pesan. “Otentikasi sumber pesan secara benar memberikan kepastian integritas data” (Inayatullah, 2007). Sedangkan username merupakan nama unik yang berbeda tiap akun yang ada. Username dan *Password* bersifat statis atau sama, maksud statis disini adalah nilai atau *values* sama dengan sebelumnya hingga user menggantinya. Biasanya user mengganti *password* ketika sudah merasa bahwa akun dia sudah tidak aman atau sudah diketahui oleh orang lain.

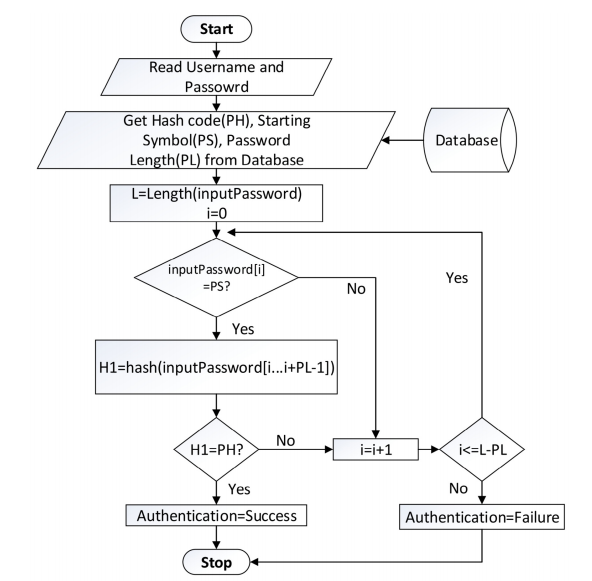
Hampir seluruh sistem yang ada pada saat ini hanya menggunakan username dan password. Hal ini memiliki kekurangan yaitu saat akun tersebut tercuri atau diketahui oleh orang lain. Maka orang tersebut dapat masuk ke sistem dan akan mengganti password yang sudah ada. Hal ini berdampak akun tersebut tidak dapat di akses lagi untuk seterusnya. Kejadian tersebut terjadi pada kasus buka lapak pada tahun 2018.

Pada kasus tersebut 13 juta data akun pelanggan buka lapak telah di curi dan di jual secara lelang pada situs jual beli online di dark web. Akun dengan nama Gnosticalplayers dikabarkan juga menjual data akun dari website lain. Isi data yang dijual tersebut berisi username, password, email dll.

## Otentikasi

Otentikasi adalah suatu langkah untuk menentukan atau memastikan bahwa seseorang (atau sesuatu) adalah autentik atau asli. Melakukan otentikasi terhadap sebuah objek adalah melakukan konfirmasi terhadap kebenarannya. Sedangkan melakukan otentikasi terhadap seseorang biasanya adalah untuk memverifikasi identitasnya (Suling dkk., 2017). Seperti password pada umumnya, syarat agar otentikasi berhasil adalah password yang dikirimkan client harus sama dengan *password* yang disimpan di server. Dengan alasan keamanan jarang sekali server menyimpan *password user* dalam bentuk *plain text*. Biasanya server menyimpan password user dalam bentuk hash sehingga tidak bisa dikembalikan dalam bentuk *plain text.* Jadi syarat otentikasi berhasil di atas bisa diartikan sebagai hasil penghitungan hash dari password yang dikirim klien harus sama dengan nilai *hash* yang disimpan dalam server.

Otentikasi diperlukan karena untuk memenuhi standar keamanan dasar yaitu kerahasiaan, integritas, dan ketersediaan (Sahu dkk., 2014). Kerahasiaan data sangat penting untuk menjaga data dari penggunaan oleh orang yang tidak berkepntingan. Berikut standar otentikasi yang digunakan kebanyakan sistem saat ini yang menggunakan metode hashing seperti pada Gambar 2.1 .



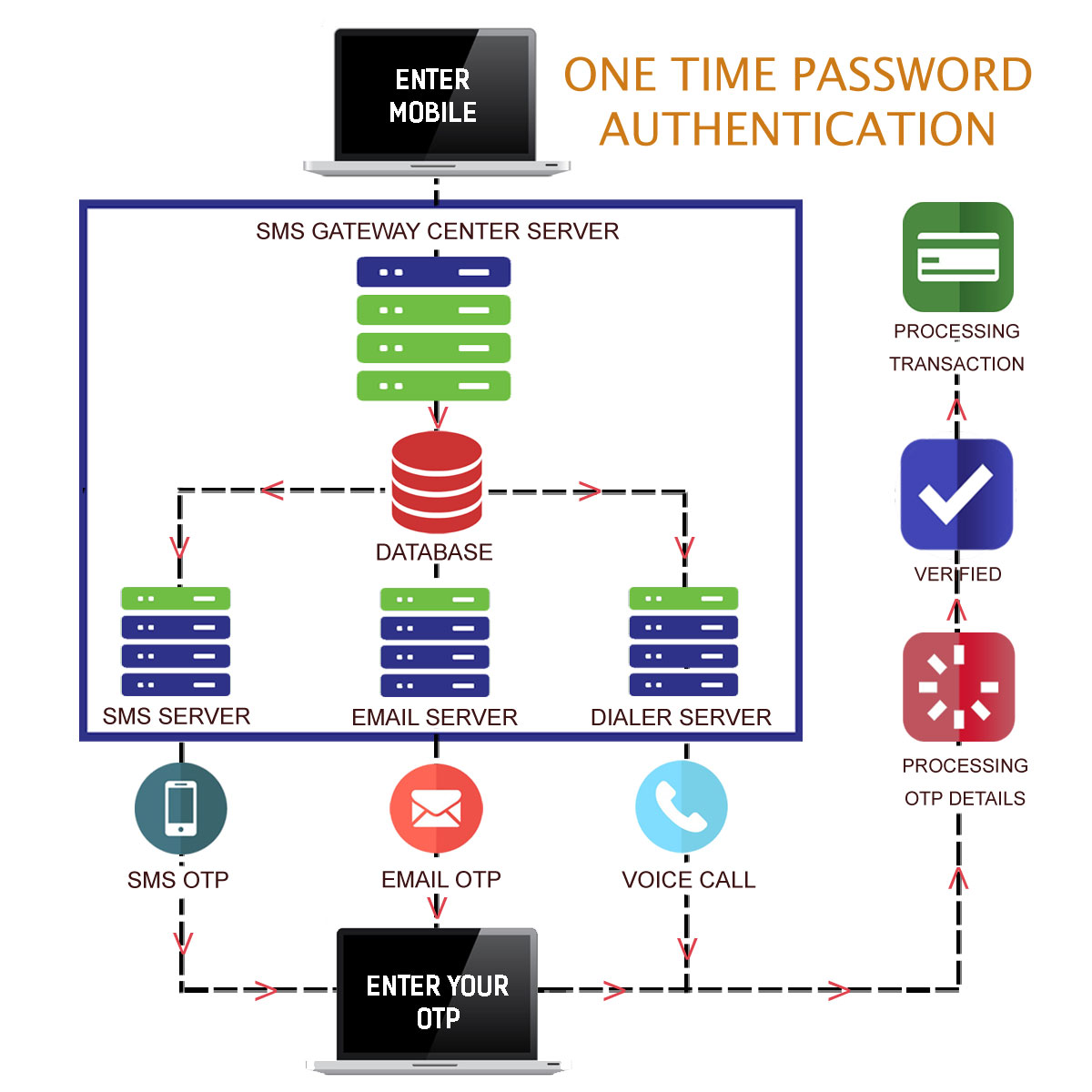
Gambar 2.1. proses otemtikasi

Dalam flowchart pada gambar 2.1 diatas proses otentikasi terjadi saat pengecekan hasil hash password dan di cocokan dengan hash yang tersimpan di database. Jika hasilnya cocok maka proses otentikasi berhasil atau sukses, dan sebaliknya jika hasil hash password tidak sesuai dengan di database maka otentikasi dinyatakan gagal.

## *One Time Password* (OTP)

*One time password* (OTP) merupakan metode otentikasi yang menggunakan *password* yang selalu berubah setelah setiap kali *login*, atau berubah setiap interval waktu tertentu (Musliyana dkk., 2016). *One time password* ini haruslah *password* yang acak sehingga sulit ditebak oleh orang lain. Keuntungan dari *one time password* adalah pencegahan penyalahgunaan username dan password yang biasanya statis. Dengan tambahan *one time password* ini maka login tidak bisa ditiru oleh orang lain. Keuntungan ini berarti jika seseorang berhasil mendapatkan username dan password, maka tidak dapat digunakan karena dia harus memasukkan *one time password* yang lain.

Penggunaan OTP membantu mengamankan transaksi atau penggunaan akun untuk memverifikasi pemilik asli akun tersebut. Setiap akun yang teregristrasi akan diwajibkan untuk memasukan kode OTP sehingga dipastikan akun tersebut digunakan oleh pengguna yang bersangkutan.

Pengiriman kode OTP kepada pengguna memiliki banyak cara seperti pada gambar 2.3 diatas. Cara tersebut antara lain menggunakan sms, email dan menggunakan telfon secara langsung. Tiap jenis pengiriman kode OTP memiliki server khusus seperti sms dibutuhkan sms gateway untuk mengirim pesan sms. Sms gateway berfungsi untuk melakukan pengiriman sms yang berisi kode OTP kepada pengguna (Imam Santoso dkk., 2013). Kode OTP yang dikirim dapat digunakan pengguna untuk melakukan otentikasi akun kedalam sistem.penggunaan. Sms memiliki kelebihan yaitu pengiriman sms dapat diterima oleh semua jenis handphone saat ini. metode sms juga memiliki kelemahan yaitu dari segi biaya. Biaya pengiriman kode bergantung kepada tiap provider dan jumlah sms yang dikirim. Semakin banyak sms yang dikirim makan biaya yang harus dibayar akan semakin tinggi.

Gambar 2.3. alur OTP

Jenis pengiriman yang kedua adalah menggunakan email dengan memanfaatkan mail server. Cara ini hampir sama seperti sms sebelumnya, yang membedakan hanyalah platform yang digunakan. Jika pada sms platform yang digunakan adalah media handphone maka pada email dapat menggunakan browser yang sudah terkoneksi internet. Cara ini relatif cepat dan murah dibandingkan dengan metode sms. Tetapi penggunaan email memiliki kelemahan yaitu pengguna diwajibkan untuk selalu melihat email yang masuk dan terkadang email OTP dianggap spam oleh sebagian mail server. Hal ini mengakibatkan gagalnya pengiriman kode OTP.

Cara ketiga adalah menggunakan telfon secara langsung yaitu pengguna akan menerima telefon dan diperdengarkan kode OTP yang ada. Cara kerjanya hampir sama dengan sms yaitu menggunakan nomor telefon yang terdaftar. Tetapi metode ini memiliki kelamahan yaitu pengguna diwajibkan menjawab telefon yang masuk dan mengingat kode OTP yang disebutkan. Metode ini juga memiliki biaya yang besar dikarenakan intensitas telefon akan mengakibatkan biaya yang terus membengkak.

Selain ketiga cara diatas terdapat cara baru dalam menyampaikan kode OTP dengan tanpa membutuhkan server khusus dan tanpa biaya yaitu dengan menggunakan Time One Time Password (TOTP). TOTP menggunakan algoritma perhitungan khusus yang dijalankan didua sisi yaitu sisi client dan sisi server.

## Time Based OTP

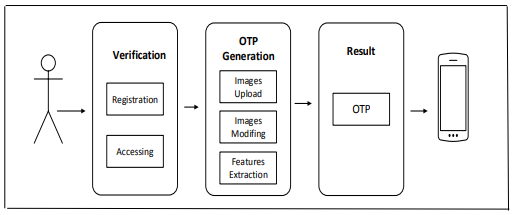
Time-Based One Time Password (TOTP) adalah salah satu algoritma yang memiliki kemampuan untuk menghasilkan password sekali pemakaian. Password yang dihasilkan oleh algoritma TOTP memiliki masa berlaku yang terbatas dan selalu berubah dalam periode tertentu. Cara kerja algortima TOTP yaitu menggabungkan antara secret key dengan current time atau waktu sekarang. Proses ini memerlukan sinkronisasi antara token milik client dengan server otentikasi. Pada jenis token yang terpisah (*disconnected token*), sinkronisasi waktu dilakukan sebelum token diberikan kepada client (Kim dkk., 2009). Tipe token lainnya melakukan sinkronisasi saat token dimasukkan dalam suatu alat input.

Setiap token memiliki sebuah jam akurat yang telah disinkronisasikan dengan waktu yang terdapat pada server otentikasi. Pada sistem OTP ini, waktu merupakan bagian yang penting dari algoritma password, karena pembangkitan password baru didasarkan pada waktu dan kunci rahasia saat itu dan bukan pada password sebelumnya .



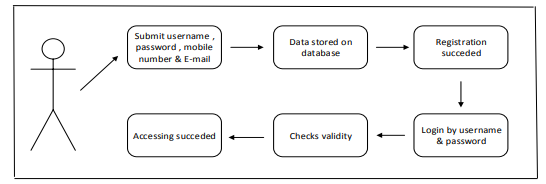
Pada OTP jenis ini sudah mulai diimplementasikan terutama pada remote *Virtual Private Network* (VPN), dan keamanan jaringan Wi-Fi dan juga pada berbagai aplikasi *Electronic Commerce* (E-commerce). Ukuran standar penggunaan waktu pada algoritma ini adalah 30 detik (M’Raihi dkk., 2011). Nilai ini dipilih sebagai keseimbangan antara keamanan dan kegunaan. Pada penelitian ini, OTP yang digunakan berbasis sinkronisasi waktu dengan kombinasi Algoritma RSA.

Model TOTP memiliki 2 komponen utama yaitu dua pembangkit kode di sisi client dan sisi server. Berikut model design TOTP untuk alur pembangkitan dan pengecekan (Fahmy dan Elkhateeb, 2018)



Gambar 2.4. alur generate kode TOTP

Pada Gambar 2.2 merupakan alur pembangkitan kode OTP, yang menggunakan Gambar sebagai media informasi. Gambar yang dimaksud merupakan kode QR yang dapat di pindai dan akan menghasilkan secret key yang digunakan untuk membangkitkan kode TOTP. Gambar tersebut dipindai menggunakan smartphone, dan akan membuat key baru tiap 30 detik sekali.



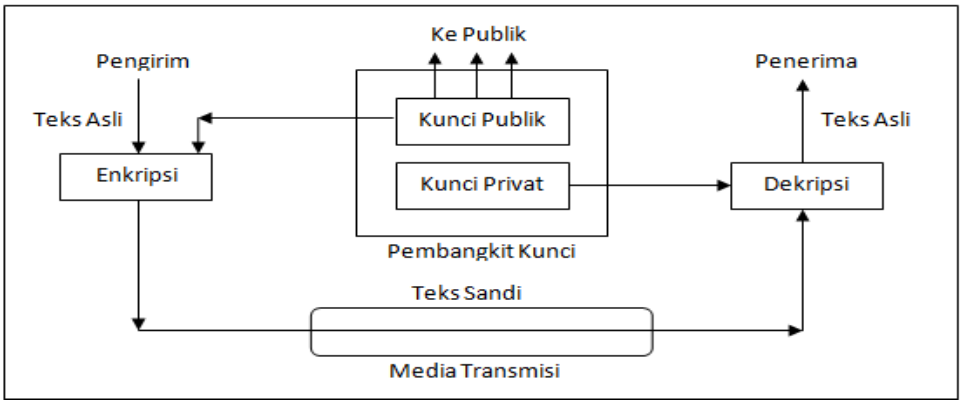
Gambar 2.5. proses login dengan OTP

TOTP berupa 6 digit angka yang dibangkitkan berdasarkan waktu dan scret key. Secret key dibangkitkan di sisi server dan di simpan di dalam database sistem dan akan di generate menjadi qr-code dan dapat di pindai dengan device pengguna..kode akan berubah tiap 30 detik sekali dan akan selalu berubah-ubah. Kode OTP yang dimasukan pengguna akan di cek apakah memiliki nilai yang sama atau tidak dengan di server. Pengecekan dilakukan setelah pengguna memasukan kode OTP. Hal ini memiliki kelebihan yaitu server tidak perlu melakukan pembangkitan kode OTP secara terus menerus.

## Algoritma RSA

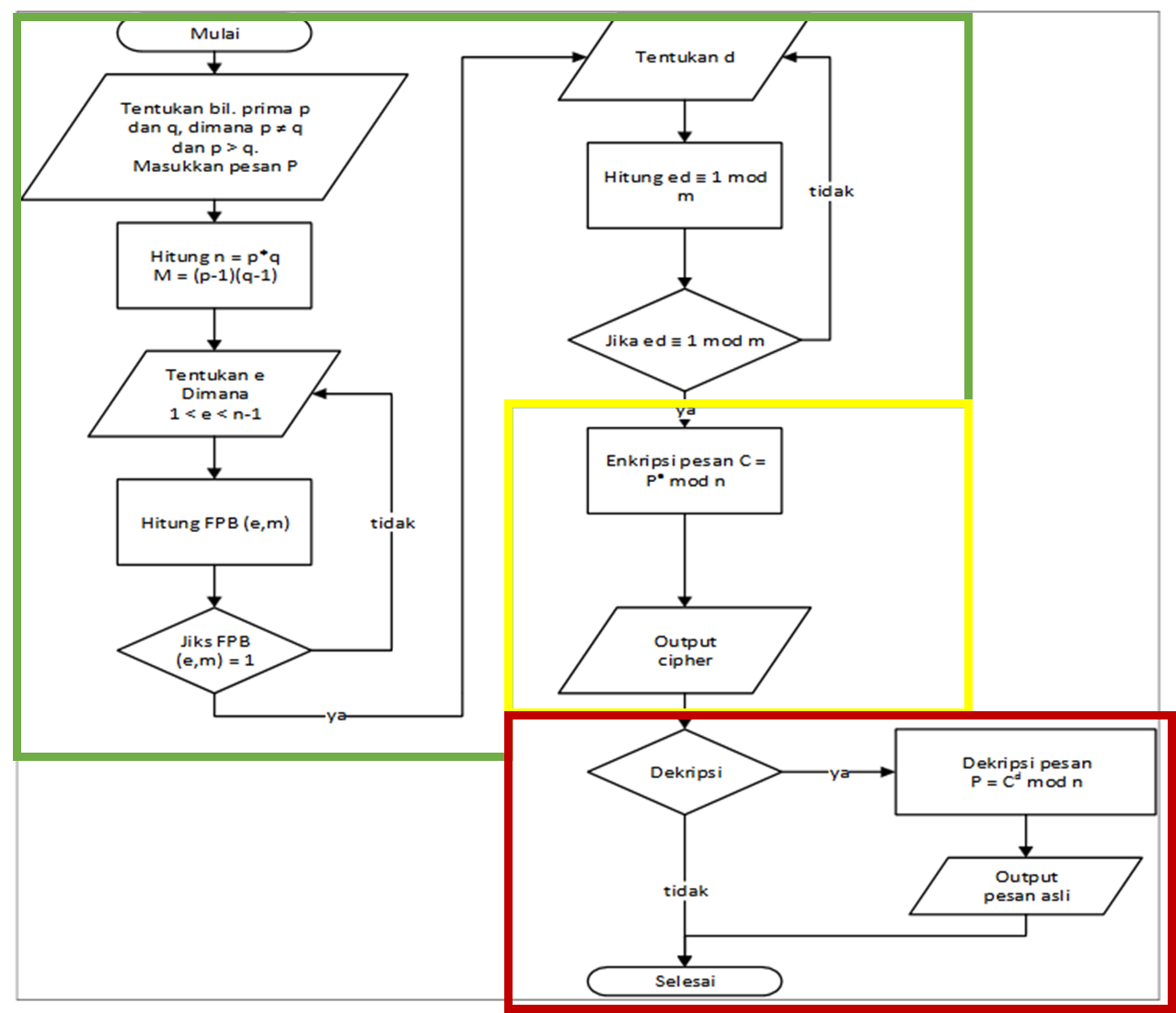
Rivest Shamir Adleman (RSA) adalah salah satu algoritma kriptografi asimetris (kriptografi kunci - publik) yaitu menggunakan dua kunci yang berbeda ( private key dan *public key*). Kekuatan algoritma RSA tidak hanya terletak pada panjang kuncinya (semakin panjang kunci, maka semakin lama waktu kerja) dan penggunaan kunci - publik dan kunci privat pada umumnya (Muchlis dkk., 2007). Algoritma ini membantu dalam pengiriman kode OTP agar lebih aman dan tidak mudah di dibaca. Oleh karena itu dibutuhkan algoritma kriptografi agar hasil OTP lebih aman.

Konsep RSA adalah menengkripsi teks dengan menggunakan public key dan untuk dapat membaca nya dengan menggunakan secret key, public key dapat di sebar dan di beritahukan kepada semua orang, sedangkan secret key harus disimpan oleh pembuatnya. Algoritma RSA menggunakan 2 angka (e dan d) sebagai kunci publik dan kunci privat. Pada algoritma RSA e dan n diumumkan untuk umum sedangkan d dirahasiakan (Arief dan Saputra, 2016). Konsep tersebut dapat dilihat pada Gambar 3.2



Gambar 2.6. konsep RSA

Sesuai dengan konsep tersebut pengirim merupakan server SI-Abka dan penerima merupakan user SI-Abka. Server akan mengirim kunci publik ke pada penerima dan menyimpan kunci private ke dalam database. Pasangan kunci publik dan kunci private di bangkitkan saat user melakukan registrasi pertama.berikut adalah flowchart atau alur pembangkitan, enkripsi dan dekripsi menggunakan metode RSA seperti pada Gambar 2.5



Gambar 2.7.Flowchart Proses Algoritma RSA

Terdapat 3 proses yang dilakukan dalam proses algoritma ini yaitu :

1. Pembentukan kunci

Proses pembangkitan kunci pada kriptografi RSA sesuai pada Gambar 2.5 pada flowchart yang telah di tandai dengan persegi warna hijau. Proses yang akan dilakukan sebagai berikut :

* 1. Pilih dua buah bilangan prima sembarang *p* dan *q*. Misal p =13 dan q =17 jaga kerahasiaan *p* dan *q*.
  2. Hitung nilai untuk kunci public n dengan formula . Nilai n digunakan untuk kunci public.
  3. Hitung . Setelah telah dihitung, *p* dan *q* dapat dihapus untuk mencegah diketahuinya oleh pihak lain.
  4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya *e,* yang relatif prima terhadap *m* (relatif prima berarti GCD(e, m) = 1) dengan syarat , , dan . Misal e = 5;
  5. Cari kunci d dengan menggunakan algoritma eucleadean yang di perluas, yaitu dengan cara berikut :

🡺

🡺

🡺

t0 = 0;

t1 = 1;

t2 = t0 – q1.t1 = 0 – 38(1) = -38

t3 = t1 – q2.t2 = 1 – 2(-38) = 77

didapatkan kunci d = 77

* 1. didapatkan kunci

n (public) = 221

e (enkripsi) = 5

d (dekripsi) = 77

1. Proses Enkripsi

Setelah proses proses pembangkitan kunci selesai, kemudian lanjut ke proses enkripsi pesan menggunakan kunci publik dari hasil pembangkitan kunci sesuai pada Gambar 2.5 pada flowchart yang telah di tandai dengan persegi warna kuning. Proses yang akan dilakukan sebagai berikut :

Menggunakan kunci yang diperoleh di atas kita akan mencoba untuk malakukan enkripsi pesan sederhana. Misalnya pesan yang akan di enkripsi adalah angka 48 dengan nama P (plain), maka akan diperoleh C (chiper) dengan perhitungan sebagai berikut:

Jadi hasil enkripsi 48 menggunakan kunci yang diperoleh di atas adalah 29.

1. Proses Dekripsi

Pesan *ciphertext* menjadi *plaintext* (pesan asli) adalah dengan menggunakan rumus dekripsi RSA. Proses ini sesuai pada Gambar 2.5 pada flowchart yang telah di tandai dengan persegi warna hijau. Proses yang akan dilakukan sebagai berikut :

*P =*

Dengan menggunakan pesan hasil enkripsi dan kunci yang diperoleh di atas dapat dilakukan dekripsi pesan seperti pada bentuk (4) berikut:

Dari hasil dekripsi di atas dapat dibuktikan bahwa hasil enkripsi pesan dapat didekripsi kembali ke pesan asli. Proses mendekripsikan tersebut membutuhkan variabel utama yaitu d dan n.

Algoritma RSA digunakan untuk mengenkripsi data yang dikirim dari client ke server. Pengguna dapat menggunakan beberapa aplikasi yang sudah ada di playstore atau aplikasi yang sudah dibuat oleh peneliti dalam pembangkitan kode OTP. Salah satu contoh aplikasi yang sudah ada antara lain google authenticator dan authy.

## QR-Code

QR Code adalah image berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. QR Code merupakan evolusi dari kode batang (barcode). Barcode merupakan sebuah simbol penandaan objek nyata yang terbuat dari pola batang-batang berwarna hitam dan putih agar mudah untuk dikenali oleh komputer (Rahayu dkk., 2006). Contoh sebuah QR Code dapat dilihat pada Gambar 2.3.

QR Code merupakan singkatan dari Quick Response Code, atau dapat diterjemahkan menjadi kode respon cepat. QR Code dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. QR Code ini dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon dengan cepat pula (Nugraha dan Munir, 2011).

Gambar 2.8. QR-code

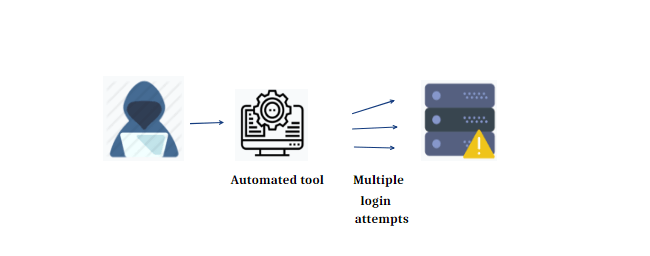
Penggunaan QR-code mempermudah dalam proses memasukan suatu teks dalam hal ini secret key. Jumlah karakter secret key adalah berjumlah 32 karakter dan gabungan dari huruf besar, huruf kecil, angka dan karakter. Nilai secret key tersebut akan menyulitkan pengguna dalam memasukan kedalam aplikasi TOTP. Oleh karena itu QR-code membantu memasukan secret key tersebut ke dalam aplikasi dengan hanya memindai gambar QR-c0de tersebut.s

## Brute Force

Algoritma *brute force* adalah algoritma yang memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas/lempang. Penyelesaian permasalahan password cracking dengan menggunakan algoritma *brute force* akan menempatkan dan mencari semua kemungkinan password dengan masukan karakter dan panjang password tertentu tentunya dengan banyak sekali kombinasi password (Pramudita, 2010). Pemakaian password sembarangan, memakai password yang cuma sepanjang 3 karakter, menggunakan kata kunci yang mudah ditebak, menggunakan password yang sama, menggunakan nama, memakai nomor telepon, sudah pasti sangat tidak aman. Namun brute force attack bisa saja memakan waktu bahkan sampai berbulan-bulan atau tahun bergantung dari bagaimana rumit passwordnya.

Jenis penyerangan dengan brute force memiliki 3 jenis yaitu *credential stuffing, reverse brute force attack, dan dictionary attack.* Pada penelitian ini penguji menggunakan mode *dictionary attack* dikarenakan kombinasi kode TOTP sudah pasti. Kombinasi tersebut berupa angka 6 digit mulai dari 000000-999999. Total keseluruhan kombinasi angka tersebut berjumlah satu juta kombinasi.

Keseluruhan kombinasi tersebut akan dicoba dikirmkan ke server mulai dari 0000-9999 secara berurutan atau pun acak. Kombinasi tesebut akan dicocokan dengan hasil pembangkitan kode TOTP oleh sistem. Tujuan brute force adalah menemukan kode TOTP yang sesuai dari satu juta kemungkinan kode yang ada.



Gambar 2.9. percobaan brute force

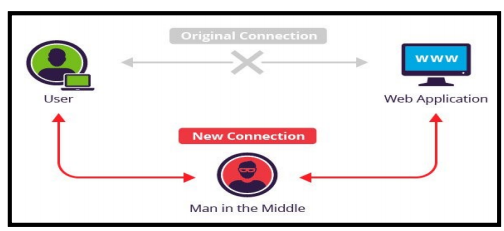
Brute force melakukan perocbaan login secara berulang dengan pola tertentu. Pola tersebut bisa dari urutan angka,huruf,karakter atau daftar yang sudah dipersiapkan terlebih dahulu. Dari banyaknya percobaan tersebut akan berdampak terhadap perangkat penyerang maupun target. Perangkat penyerang akan terus menyiapkan request untuk login. Request yang sudah disipakan akan dikirimkan ke target. Karena target mendapatkan request secara terus menerus akan berdampak keperformanya.

Teknik brute force dengan metode dictionary attack membutuhkan daftar kemungkinan yang digunakan untuk percobaan penyerangan. Data tersebut dapat dibuat sendiri dengan memasukan kata-kata tertentu.

## Man in the middle

Man in the midle (MITM) merupakan jenis penyerangan dimana penyerang melakukan pembacaan data atau pengiriman data antara dua objek yang saling berkomunikasi satu sama lain (Mallik dkk., 2019). Man in the middle sendiri jika diartikan ke bahasa Indonesia berarti sesorang yang berada di tengah yang berarti posisi penyerang berada di antara target dan sumber data. Tujuan dari teknik MITM adalah mencuri data informasi personal, data login, data akun dan No. kartu kredit.

Teknik MITM bekerja dengan cara mengelabui arus data antara client dan server. Jika dalam keadaan normal user langsung berkomunikasi dengan server tetapi jika terdapat penyerangan MITM penyerang akan melihat arus data yang dikirim oleh client ke server dan sebaliknya. skema penyerangan dapat dilihat seperti pada Gambar 2.7.



Gambar 2.10. cara kerja MITM

Penggunaan man in the middle berdampak kepada kerahasiaan data yang dikirim dalam satu jaringan. Data yang dikirim atau diterima oleh pengguna dibaca oleh penyerang. Penggunaan enkripsi sangat diperlukan untuk menangkal penyerangan jenis ini. Keamanan dalam pengiriman data dan informasi tidak hanya bergantung pada faktor kuatnya algoritma kriptografi yang digunakan pada pesan, tapi juga pada jalur informasi yang dilewati. Bila jalur informasi tersebut mampu disadap, penyerangan lebih lanjut dapat dilakukan pada pesan yang telah terenkripsi baik dalam hal keutuhan pesan maupun kebenaran pesan (Ramadhan, 2010).

# METODOLOGI PENELITIAN

Pada bab ini akan membahas objek penelitian, tempat penelitian, tahapan penelitian, dan studi literatur yang digunakan dalam pembangunan modul sistem login pada sistem SI-abka dan implementasi algoritma RSA aplikasi tersebut untuk menjaga keaslian kerahasiaan transaksi data antara client dan server.

## Objek Penelitian

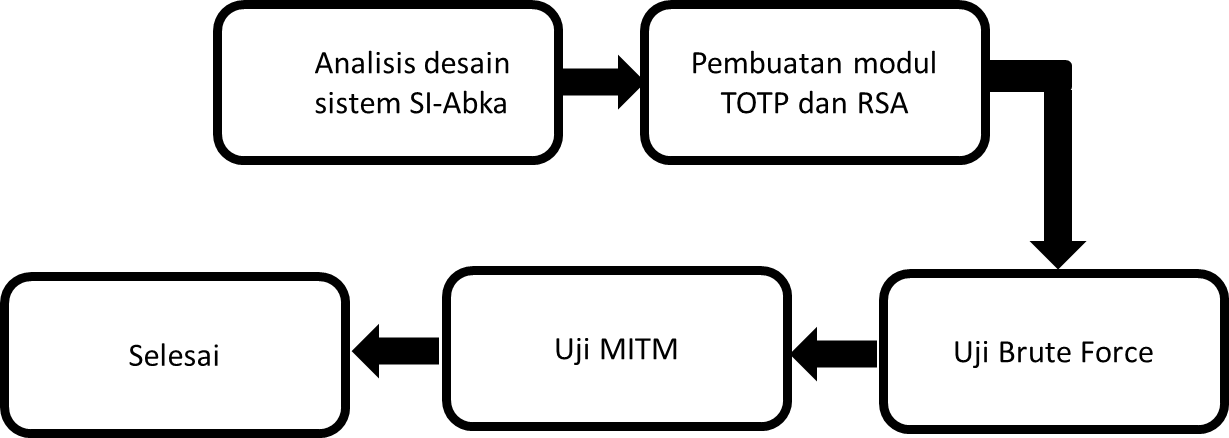
Objek penelitian merupakan sistem SI-Abka yang ada pada koperasi kementeriam agama jember. Aplikasi tersebut penggunakan web php dan database mysql. Proses otentikasi dari seluruh akun hanya menggunakan username dan password. Dari sistem tersebut akan ditambah *two factor authentication* berupa TOTP dan algoritma RSA. Sistem authenticaton tambahan tersebut diharapkan dapat memperkuat keamanan sistem SI-Abka. Kode OTP akan di generate atau dibangkitkan menggunakan aplikasi di HP android atau sebuah alat portable. Kode di bangkitkan dengan cara memasukan *public key* ke dalam sistem SI-Abka saat meregistrasikan aplikasi android dengan sistem agar dapat selaras.

## Tempat Penelitian

Tempat dilaksanakan penelitian yaitu di Kementerian Agama Kabupaten Jember. SI-Abka di terapkan pada Koperasi Amal Bakti Kementerian Agama sebagai sistem yang membantu pelayanan di koperasi.

## Tahapan Penelitian

Tahapan Penelitian yang akan dilakukan dapat dilihat dalam diagram alur di bawah ini. Penelitian ini terdiri dari 5 tahap mulai dari perencanaan, implementasi sampai tahap testing. Tahap-tahap ini harus di lakukan secara urut karena tahap sebelumnya berpengaruh ke tahap selanjutnya.



Gambar 3.1 Alur Tahapan Penelitian

## Analisis desain sistem SI-Abka

Desain otentikasi pada SI-Abka menggunakan metode *single factor authentication* yang menggunakan username dan password*.* Sistem SI-abka memiliki 3 level user dan seluruh user memiliki metode login yang sama. Username pada sistem SI-Abka menggunakan nomor KTA yang merupakan nomor kartu tanda anggota. Proses otentikasi dilakukan pada tahap awal masuk ke dalam sistem.

## Pembuatan modul TOTP dan RSA

Pada saat login pertama data user tersebut akan di lihat apakah sudah mendaftarkan akun tersebut dengan modul TOTP, jika sudah maka user tersebut akan langsung di arahkan ke form input kode OTP, jika tidak maka akan di arahkan ke tampilan generate kode OTP.

Pembuatan modul TOTP bertujuan untuk menghasilkan secret key yang akan disimpan dan digunakan umtuk membangkitkan kode TOTP di perangkat user dan di server. Sedangkan modul RSA bertujuan untuk membangkitkan secret key dan public key untuk mengenkripsi data yang dikirim oleh client ke server serta mendekripsi data yang telah dikirimkan.

### Modul TOTP

Kode OTP pada aplikasi android akan membangkitkan TOTP berdasarkan data yang sama dengan server meskipun tidak berkomunikasi secara langsung. Setelah kode OTP berhasil dibangkitkan maka aplikasi android akan menampilkan kode tersebut. User dapat memasukan kode TOTP ke dalam form yang muncul setelah memasukan KTA dan password disertai dengan mengirimkan public key yang ada. Kode TOTP akan di enkripsi oleh browser sebelum dikirim ke server dengan menggunakan public key sesuai dengan data akun user. Hasil enkripsi akan diterima sistem SI-ABKA dan di dekripsi serta dicocokan dengan hasil pembangkitan yang dilakukan oleh sistem. Jika sesuai maka perintah akan di loloskan jika tidak maka akan invalid dan gagal.

Tabel 3.1. Tabel kombinasi TOTP

Sistem otentikasi SI-Abka yang sebelumnya hanya memiliki 4 kombinasi seperti pada Tabel 2.2, maka setelah TOTP ditambakan maka akan terdapat 8 kombinasi tambahan. Oleh karena itu TOTP di sebut dengan two factor authentication atau faktor tambahan untuk otentikasi. Kembinasi tersebut dapat dilihat pada Tabel 3.2 dibawah ini.

|  |  |  |  |
| --- | --- | --- | --- |
| **Username** | **Password** | **OTP** | **Status** |
| Salah | Salah | Salah | Gagal |
| Salah | Salah | Benar | Gagal |
| Salah | Benar | Salah | Gagal |
| Salah | Benar | Benar | Gagal |
| Benar | Salah | Salah | Gagal |
| Benar | Salah | Benar | Gagal |
| Benar | Benar | Salah | Gagal |
| Benar | Benar | Benar | **Berhasil** |

Dari kombinasi diatas, alur dari pengguna TOTP hampir sama seperti penggunaan username dan password. Tetapi memiliki perbedaan nilai dari TOTP dinamis atau berubah-ubah tidak seperti username dan password yang sama. Oleh karena itu penggunaan TOTP diharapkan dapat lebih mengamankan metode login dari SI-Abka. Kombinasi yang sebelumnya hanya 4 sekarang menjadi 8 kombinasi.

Sistem SI-abka memiliki 3 level user dan seluruh user memiliki metode login yang sama yaitu menggunakan username dan password. Saat user selesai melakukan login dan berhasil login maka sistem tidak langsung mengarahkan ke dashboard, tetapi akan diarahkan untuk memasukan kode OTP. jika saat diarahkan untuk input kode OTP data user tidak memiliki secret key TOTP maka sistem akan membangkitkan secret key dan menirimkan qr-code utnuk dapat di scan dan dibaca oleh andoid untuk menghasilkan kode OTP. jika kode sesuai maka akan diarahkan ke menu dashboard. Proses pembuatan secret key kode TOTP hanya sekali saat registrasi awal. Untuk melakukan reset dapat melalui operator atau menu reset dengan menggunakan email konformasi.

Saat terdapat user yang diarahkan ke page generate kode maka sistem akan menggenerate atau membuatkan suatu kode unik yaitu *secret key* untuk TOTP.Key yang dibuat berupa qrcode yang harus discan atau di pindai dengan aplikasi tertentu. Setelah di pindai maka alikasi tersebut akan menghitung dan menampilkan kode OTP sesuai dengan secret key tersebut. Setelah *secret key* telah dibuat maka sistem akan menunggu input kode TOTP oleh user. Jika kode tersebut benar maka *secret key* akan disimpan , jika tidak maka akan di arahkan ke tampilan login kembali.





Gambar 3.2.. flowchart implementasi secret key TOTP

Dari flowchart tersebut menghasilkan file library yang dapat digunakan untuk membangkitkan secret key,private key dan public key. Secret key digunakan untuk pembangkitan TOTP. Sedangkan private key dan public key tersebut dapat digunakan untuk melakukan enkripsi dan dekripsi dengan menggunakan algoritma RSA

### Modul RSA

Algoritma RSA di gunakan untuk mengamankan data yang dikirim oleh client ke server. Dalam penelitian kali ini data yang dikirim hanya data OTP sebagai pembanding antara data yang dienkripsi dan yang tanpa enkripsi. Dengan penambahan enkripsi RSA diharapkan data tidak mudah di baca dan di tebak. Proses enkripsi terjadi di browser pengguna sebelum proses pengiriman data melalui metode post. Data yang terkirim berupa 6 kelompok data yang sudah terenkripsi dengan RSA. Setelah data di terima server data tersebut akan dienkripsi

Dari flowchart tersebut dibuat alur proses login dengan TOTP dan RSA seperti pada Gambar 3.5.



Gambar 3.3. alur login TOTP dan RSA

Pada Gambar 3.5 dapat dilihat alur data penggunaan TOTP dan RSA pada sistem siabka. Pada saat user melakukan proses login sistem akan mengecek apakah user tersebut memiliki secret key atau tidak. Secret key pada tahap ini adalah scret key yang digunakan untuk proses pembangkitan TOTP. Jika sudah punya maka sistem akan menampilkan halam untuk memasukan kode TOTP. Jika tidak maka sistem akan membuat secret key baru dan menampikan gambar QR-Code yang dapat di scan dengan aplikasi yang sudah disediakan. Saat QR-Code ditampilkan pada halam juga terdapat form input kode TOTP. Saat user memasukan kode TOTP sistem akan membangkitkan kode TOTP juga sesuai dengan scret key yang telah disimpan di database (user sudah memiliki secret key) atau yang telah dibangkitkan (user belum memiliki secret key). jika kode sesuai maka perintah akan diteruskan jika tidak sesuai maka sistem akan menolak.

## Uji keamanan

Uji keamanan dilakukan untuk mencoba keamanan sistem saat password dan username telah diketahui. Saat password dan username digunakan untuk login, sistem akan menampilkan form input kode OTP. Uji keamanan ini berfungsi untuk mengetahui seberapa besar dampak penggunaan *two factor authentication* dan algoritma RSA terhadap pengamanan proses login.



### Uji brute force

*Brute force* merupakan algoritma sederhana dalam proses pembuatan kemugkinan kode. Pengguna hanya tinggal memasukan panjang karakter dan ukuran kode yang akan dicari. Tiap kemungkinan kode akan di generate secara berurutan. Uji keamanan kan di lakukan dengan cara memasukan kode OTP secara random dan cepat menggunakan metode *brute force* dengan bantuian aplikasi burp suite. Cara kerja aplikasi tersebut adalah memotong jalur komunikasi dan melihat data yang dikirmkan. Data tersebut berisi data akun yang berupa data KTA,Password, dan kode OTP. Dalam pengujian kali ini kita akan menguji seberapa kuat dan berapa persen kemungkinan kode OTP akan diketahui.

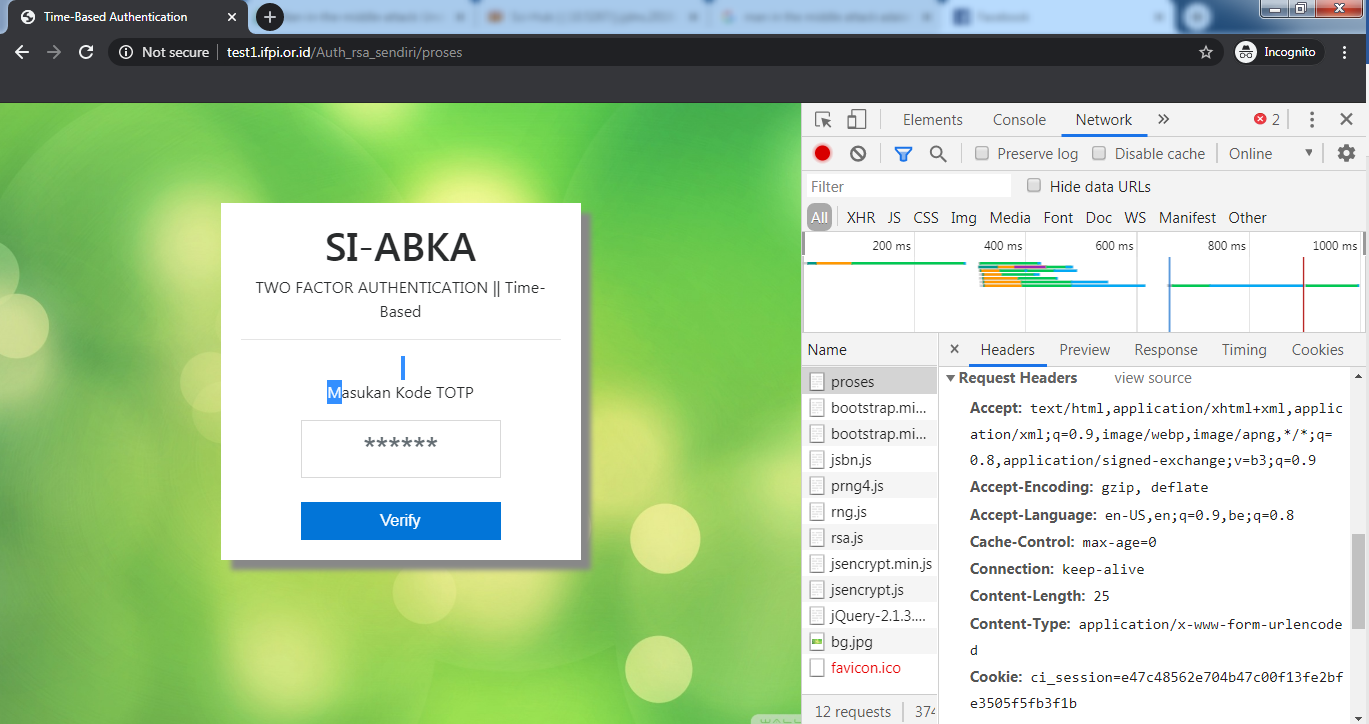
Pengujian *brute force* di lakukan dengan cara mencoba setiap kemungkinan kode OTP. Kode OTP yang di bangkitkan memiliki kriteria 6 digit dan merupakan bilangan cacah atau bilangan postif dan nol. Total kemungkinan variasi kode OTP berjumlah satu juta mulai dari 000000 – 999999.

Aplikasi burpsuite bekerja dengan cara mengirimkan file header yang berisi data akun berupa KTA, password, dan kode OTP. Aplikasi burpsuite akan membantu pengujian dengan mengacak kode OTP dan mengirimkanya ke sistem. Pengujian kode OTP sukses jika mendapatkan respons code 303 dan gagal jika mendapatkan respons code 401. Respons code 303 berarti mengalihkan aplikasi web ke url tertentu (beranda user) sedangkan 401 berarti halaman yang sedang di akses tidak dapat dimuat sampai user login dengan akun yang valid.

Pengujian kali ini menggunakan laptop dengan spesikasi alat sebagai berikut:

1. Prosesor : Intel Core i3 2330M processor (3MB L3 cache, 2.20 GHz)
2. Sistem operasi : Windows 7 Ultimate
3. Memori : 4 GB DDR3 SDRAM 1066 MHz
4. Grafis : Intel HD 3000 Graphics

Sebelum melakukan pengujian *brute force* terlebih dahulu menyiapkan data-data pendukung seperti *host target*, dan *request header*. Untuk mendapatkan data tersebut dapat menggunakan browser dan masuk ke developer mode seperti Gambar 3.4 . Request header hanya dapat digunakan selama session di server masih ada jika session nya sudah habis maka seluruh request akan di tolak. Lama dari session tergantung settingan wesite target pada SI-Abka session di atur selama 7200 detik atau 120 menit.



Gambar 3.4. tampilan pembacaan request header

Contoh file request header pengiriman data login yang berhasil didapatkan

1. POST /Auth/validasi HTTP/1.1
2. Host: localhost
3. UserAgent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
4. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8
5. Accept-Language: en-US,en;q=0.5
6. Accept-Encoding: gzip, deflate
7. Referer: http://test1.ifpi.or.id/siabka/Auth/proses
8. Content-Type: application/x-www-form-urlencoded
9. Content-Length: 42
10. Connection: close
11. Cookie: ci\_session=495fc724245e0901b17f4d51bb30cd7a58f8093d
12. Upgrade-Insecure-Requests: 1
14. totp=123123&username=admi.n&password=admin

### Uji MITM

Selain menggunakan teknik *brute force* pengujian juga menggunakan teknik *man in the middle*. Cara kerja teknik ini adalah mendengarkan/melihat *traffic* yang mengarah ke suatu situs web dan membaca setiap data yang dikirimkan, saat terdapat penguna yang login ke SI-Abka maka data yang di kirimkan dapat terbaca.

Simulasi MITM menggunakan dua buah alat, salah satu sebagai target dan yang lain sebagai penyerang. Device target dapat menggunakan seluruh device windows, android, dll, sedangkan penyerang menggunakan OS linux. Teknik MITM menggunakan beberapa tools antara lain:

1. Netdiscover

Netdiscover merupakan tool ip hunter yang berfungsi untuk membaca dan manecari ip yang terkoneksi ke dalam jaringan. Cara kerja dari netdiscover adalah membaca ip yang berkomunikasi dengan router. Lama pemindaian bergantung pada jumlah device yang terkoneksi dan spesifikasi router yang dipakai.

1. Ettercap

Ettercap berfungsi untuk membaca data yang dikirim oleh router ke ip target. Cara kerjanya adalah mengelabui router dengan menyamar sebagai ip target. Data yang seharusnya dikirim ke target akan terkirim juga ke penyerang. Data tersebut dapat dibaca oleh penyerang dan digunakan layaknya pemilik data asli.

1. Sslstrip

Sslstrip digunakan untuk membelokan traffic https ke http. Teknologi https berfungsi untuk mengkripsi data dari server ke user dengan menggunakan token khusus. Sedangkan http hanya mengirimkan tanpa ada nya eknripsi data. Oelh karena itu sslstrip berfungsi untuk mnegubah link target yang menggunakan https ke http agar mudah dibaca oleh ettercap.

Data yang didapat dari teknik MITM berupa data post yang dikirim oleh client ke server. Data tersebut berupa text sehingga dapat dengan mudah dibaca tanpa bantuan tool khusus. Data yang dikirim oleh server memiliki banyak jenis mulai dari data Gambar sampai teks khusus seperti file tambahan website. Pada percobaan ini data yang di tangkap adalah berupa data post.

Skema penyerangan dengan menggunakan metode MITM adalah melakukan sniffing (mengintip) data yang dikirim atau diterima oleh user. Alur penyerangan MITM adalah sebagai berikut seperti pada Gambar 3.5.



Gambar 3.5. skema penyerangan MITM

Teknik Man in the Middle (MITM) memiliki banyak fungsi antara lain melakukan mengambil data yang dikirim dan diubah sebelum diteruskan ke target atau hanya membaca data yang lewat saja. Pada penelitian kali ini tugas MITM adalah hanya membaca data yang lewat antara target dan server. Target data yang diuji pada penelitian kali ini adalah data post dari target ke server. Pengujian dinyatakan berhasil jika data yang dikirim dapat dibaca dan data tidak dapat digunakan penyerang untuk masuk kedalam sistem. pengujian MITM juga berfungsi untuk melihat apakah data yang dikirim berhasil dienkripsi atau tidak.

# HASIL DAN PEMBAHASAN

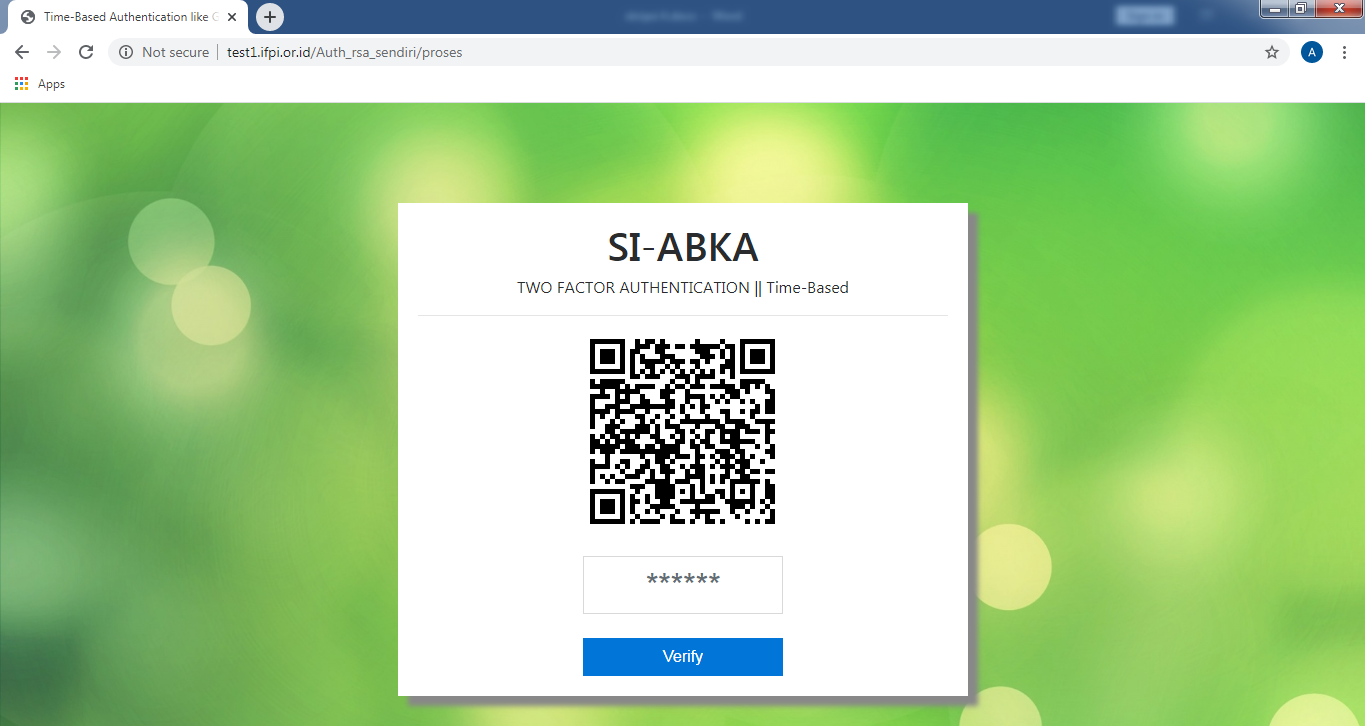


## Hasil Implementasi Pembuatan Modul TOTP dan RSA

Modul TOTP dan modul RSA pada penelitian ini memiliki fungsi yang saling berhubungan. Modul TOTP berfungsi sebagai pembangkit kode OTP sedangkan modul RSA sebagai pengaman tambahan. Pengaman yang dimaksud adalah mengenkripsi data yang dikirim oleh client menuju server dan di dekripsi oleh server untuk dibaca. Hasil dari implementasi kedua metode terebut adalah berupa file library PHP dan javascript.

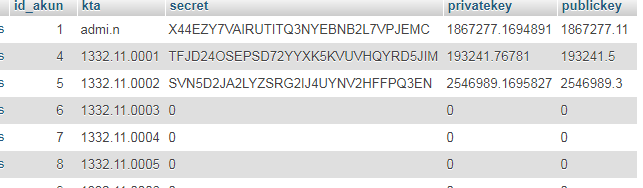
### Modul TOTP

Hasil pembuatan modul TOTP adalah berupa halaman input nomor OTP yang berjumlah 6 digit. Tampilan halaman TOTP ada dua macam yaitu saat pembangkitan secret key dan saat user sudah punya secret key. tampilan akan berupa inputan dan gambar QR-code saat user tidak punya secret key seperti pada Gambar 4.1.



Gambar 4.1. halaman pembuatan secret key kode TOTP

Pada halaman tersebut user diwajibkan untuk melakukan pemindaian dengan aplikasi TOTP di android. Aplikasi akan bekerja secara benar dengan syarat waktu pada server dan pada android sama. Waktu merupakan faktor penting dalam TOTP karena merupakan acuan kedua setelah secret key dalam proses pembangkitanya.



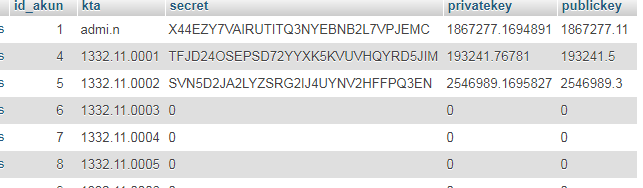
Gambar 4.2. hasil pembangkitan secret key

Hasil pembangkitan secret key berupa 32 digit gabungan angka dan huruf yang bersifat random dan tidak sama tiap akun. Form yang tersedia seperti gambar 4.1 dapat diisi dengan kode 6 digit yang telah diketahui pengguna. Sesaat setelah kode TOTP dikirim server akan menerima dan membandingkan kode tersebut apakah sama dengan server atau tidak. Dari floechar pada gambar 3.2 maka terbentuklah library yang berfungsi sebagai pengecekan kode TOTP yang dikirim ke server dengan kode yang dibangkitkan server

1. **public** **function** verifyCode($secret, $code, $waktudurasi = 1, $waktusekarang = null) {
2. // waktudurasi = jika 1 maka 30 detik jika 2 makan 60 detik dst.
3. // waktusekarang = waktu yang diambil di server
4. **if** ($waktusekarang === null) {
5. $waktusekarang = floor(time() / 30);
6. }
7. **if** (strlen($code) != 6) {
8. **return** false;
9. }
10. **for** ($i = - $waktudurasi;$i <= $waktudurasi;++$i) {
11. $codehasil = $this->getCode($secret);
12. **if** ($codehasil == $code) {
13. **return** true;
14. }
15. }
16. **return** false;
17. }

### Modul RSA

Modul RSA berfungsi untuk menkripsi data yang dikirim oleh user menuju server. Hal ini bertujuan untuk meningkatkan keamanan dan kerahasiaan data yang dikirim. Pembangkitan private key dan public key RSA berada di server saat user berhasil memasukan kode TOTP yang benar. Hasil pembangkitan public key dan private key berupa bilangan prima acak antara 100-2000. Hasil pembangkitan berupa 3 buah variabel yaitu d,e,n. Varabel d dan e digunakan sebagai private key sedangakan variabel e dan n digunakan sebagai public key. berikut contoh hasil pembangkitan public key dan private key yang telah di simpan di dalam database.



Gambar 4.3. hasil pembangkitan secret key , private key dan public key

Pada Gambar 4.3 tersebut terdapat 3 kolom utama yaitu secret, privatekey dan publickey. Pada tiap kolom berisi angka dan huruf unik yang tidak sama tiap akunnya. Secret key merupakan kunci yang dibutuhkan oleh TOTP dalam pembangkitan kodenya yang setiap 30 detik akan berubah. Private key dan public key merupakan pasangan kunci yang digunakan RSA dalam melakukan enkripsi dan dekripsi data. Private key akan disimpan dan tidak akan dikeluarkan ke publik, sedangkan public key dapat di sebarkan ke publik. Hasil pembangkitan private key dan public key berupa angka dengan dua kelompok angka yang dibatasi tanda titik . kelompok angka tersebut akan dipecah sesuai dengan format pada Table 4.1 berikut:

Tabel 4.1. hasil pembnagkitan kunci

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Privatekey(n.d) | Publickey(n.e) | n | d | e |
| 1867277.1694891 | 1867277.11 | 1867277 | 1694891 | 11 |
| 193241.76781 | 193241.5 | 193241 | 76781 | 5 |
| 2546989.1695827 | 2546989.3 | 2546989 | 1695827 | 3 |

Ketiga variabel tersebut digunakan oleh library yang telah dibuat yang akan digunakan sebagai kuni RSA. Variabel n dan e akan digunakan untuk public key yang berfungisi untuk melakukan enkripsi. Variabel tersebut akan dikirimkan ke user dan pada browser user akan melakukan enkripsi sebelum dikirimkan. Proses enkripsi menggunakan bantuan javascript, berikut source code nya.

1. **var** msg = $('#code').val(); //
2. **var** angkaarray = msg.split("");
3. **var** **public** = "<?php echo $publickey; ?>";
4. **var** publickey = **public**.split(".");
5. **var** n =publickey[0];// n
6. **var** e =publickey[1];// e
7. **var** hasilenkrip='';
8. **for**(**var** i=0;i<angkaarray.length;++i){
9. **var** getpow=(Math.pow(angkaarray[i],e))%n;
10. hasilenkrip = hasilenkrip.concat(getpow);
11. **if**(i!=angkaarray.length-1){
12. hasilenkrip = hasilenkrip.concat(".");
13. }
14. }

Kode pada baris pertama merupakan kode TOTP user yang akan dienkripsi (plain teks). Publickey yang dikirim oleh server akan di pecah menjadi variabel n dan e seperti pada baris 4-6. Kedua variabel akan dihitung sesuai dengan rumus RSA seperti rumus pada halaman 14. Hasil enkripsi akan dikirimkan dengan data lain nya ke server.

Pada server data yang dikirimkan user akan diterima dan didekripsi dengan private key sesuai dengan data akun nya. Private key dan public key tidak bisa digunakan secara terpisah jadi harus digunakan secara berpasangan. Berikut source code proses server melakukan dekripsi dengan menggunakan bantuan php.

1. **function** dekrip\_withkey($data, $user\_n, $user\_d) {
2. $teks = explode(".", $data);
3. **foreach** ($teks **as** $nilai) {
4. //rumus enkripsi <pesan>=<enkripsi>^<d>mod<n>
5. $hasildekrip.= (gmp\_strval(gmp\_mod(gmp\_pow($nilai, $user\_d), $user\_n)));
6. }
7. **return** $hasildekrip;
8. }

Proses dekripsi hampir sama saat melakukan enkripsi yaitu memecah data private key menjadi variabel n dan d. Karena nilai pangkat pada rsa memiliki nilai yang tinggi sampai ribuan maka perhitungan tidak bisa menggunakan rumus biasa. Nilai pangkat pada RSA mencapai ribuan sehingga membutuhkan bantuan fungsi gmp. Hasil dekripsi akan di cocokan dengan pembangkitan TOTP pada level server jika sesuai maka akan diteruskan dan akan di tolak jika kode TOTP salah.

## Hasil Pengujian modul TOTP dan Algoritma RSA

Pada bab ini penguji ingin menguji apakah alur sistem sudah sesuai dengan flowchart pada Gambar 3.5 dan alur penggunaan kerja sistem sudah sesuai dengan Gambar 3.6.

* 1. User login

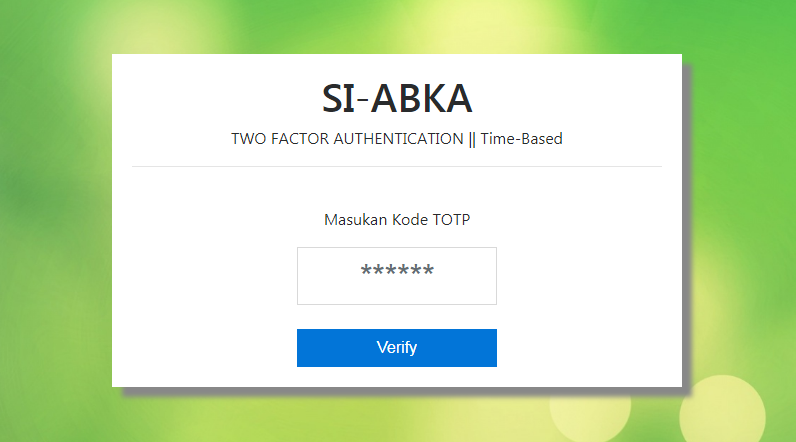
Saat user akan masuk kesistem tampilan website seperti pada umumnya yaitu memasukan username dan password tetapi. pada SI-Abka username digantikan dengan KTA. Tampilan halaman login Seperti pada Gambar 4.4. pada halaman ini tampilan website sama seperti lainya. User diwajibkan memasukan KTA dan password sesuai dengan akun mereka. Sistem akan menlak saat KTA atau password salah. Sebaliknya saat data KTA dan password benar maka sistem akan mengarahkan ke tampilan berikutnya yaitu modul TOTP.



Gambar 4.4. tampilan login SI-Abka

* 1. Cek apakah user punya secret key TOTP tersimpan di database atau tidak. Seperti pada contoh Gambar 4.2, secret key merupakan gabungan dari huruf dan angka secara acak.
     1. User memiliki secret key

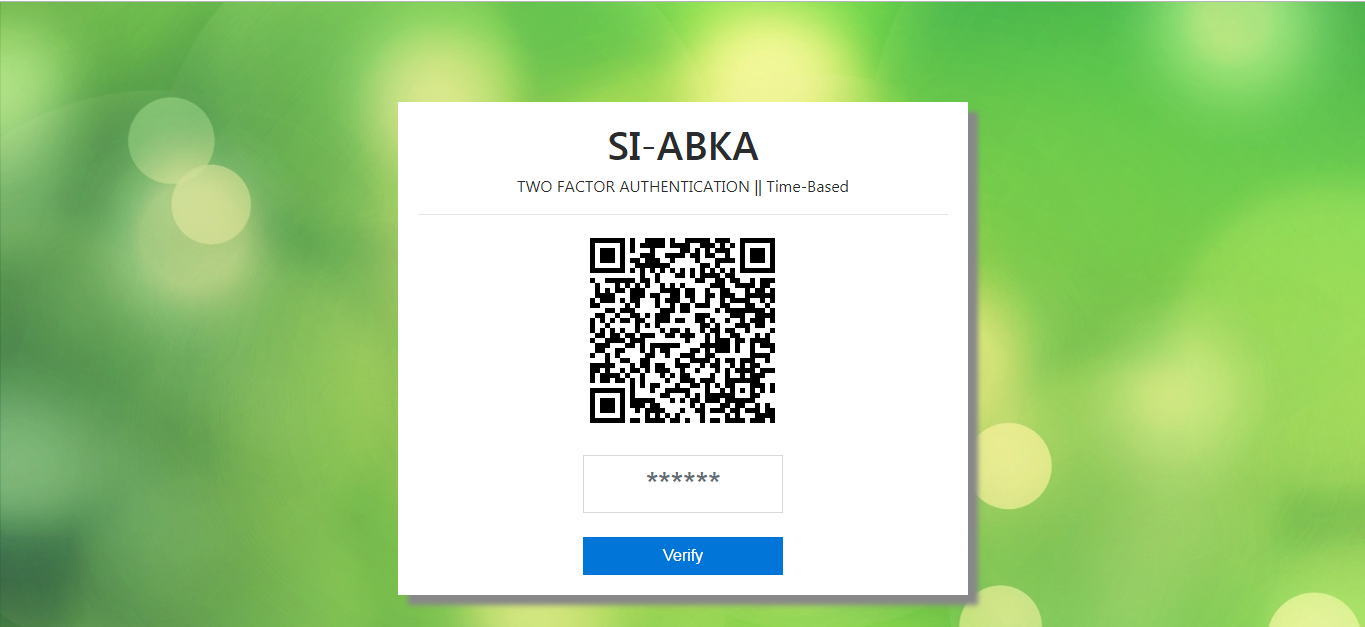
Setelah memasukan KTA maka data user akan di cek di database. Jika user memiliki seret key akan diarahkan ke tampilan seperti pada Gambar 4.5 dibawah ini. Maka user tinggal membuka aplikasi TOTP dan memasukan angka yang di tampilkan di aplikasi tersebut.



Gambar 4.5. input kode TOTP

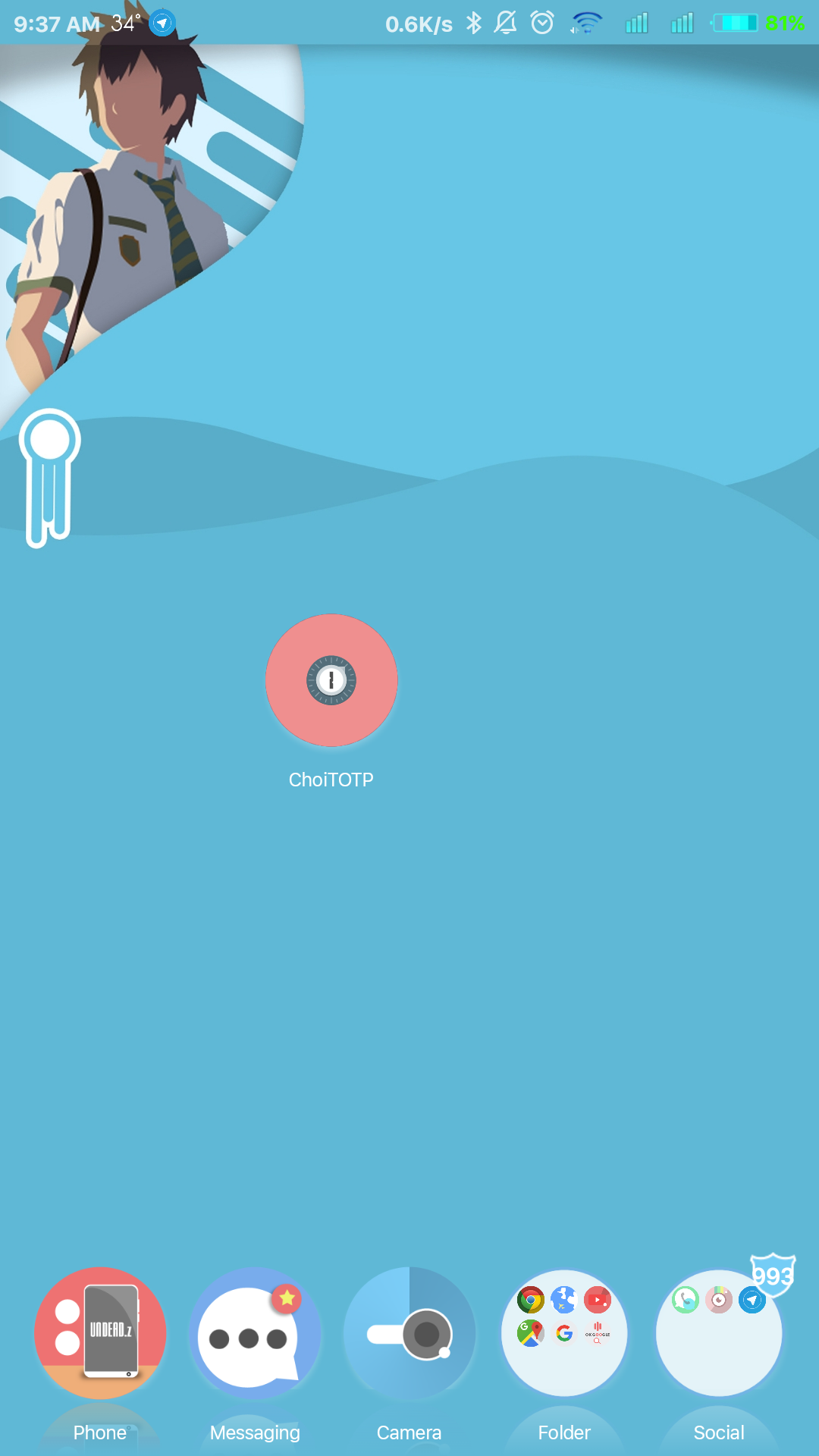
* + 1. User belum memiliki secret key

1. Jika user belum memiliki secret key maka sistem akan mengarahkan user ke tampilan generate secret key seperti pada Gambar 4.6 di bawah ini



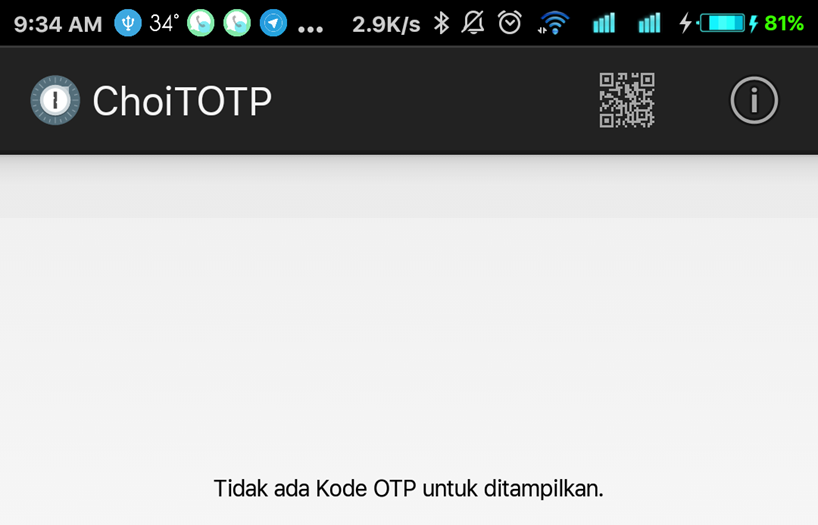
Gambar 4.6. input kode TOTP dengan qr-code

1. Untuk mendapatkan kode OTP maka user perlu melakukan scan qr-code terlebih dahulu seperti tampilan Gambar 4.7 di bawah ini



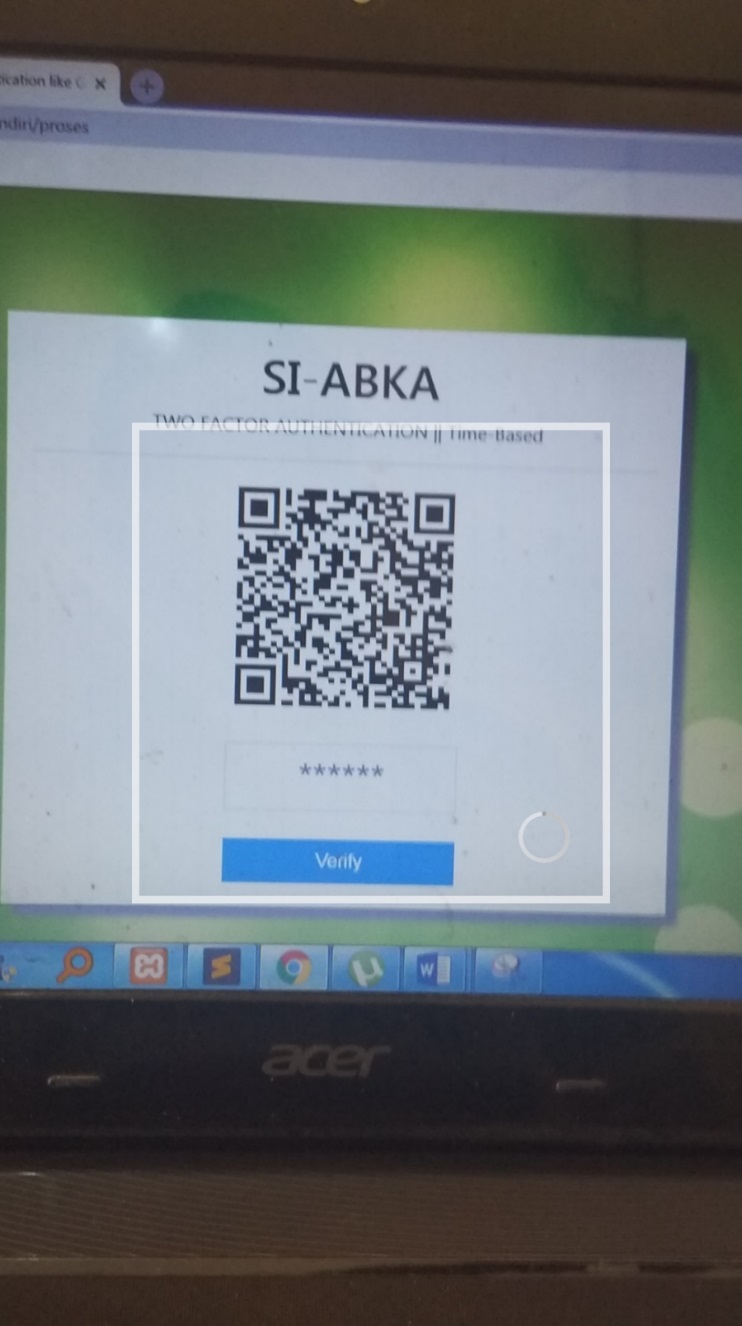
Gambar 4.7. aplikasi ChoiTOTP

1. Setelah aplikasi terbuka user perlu memasukan secret key dengan memindai qr-code. Klik pada icon di pojok kanan atas.



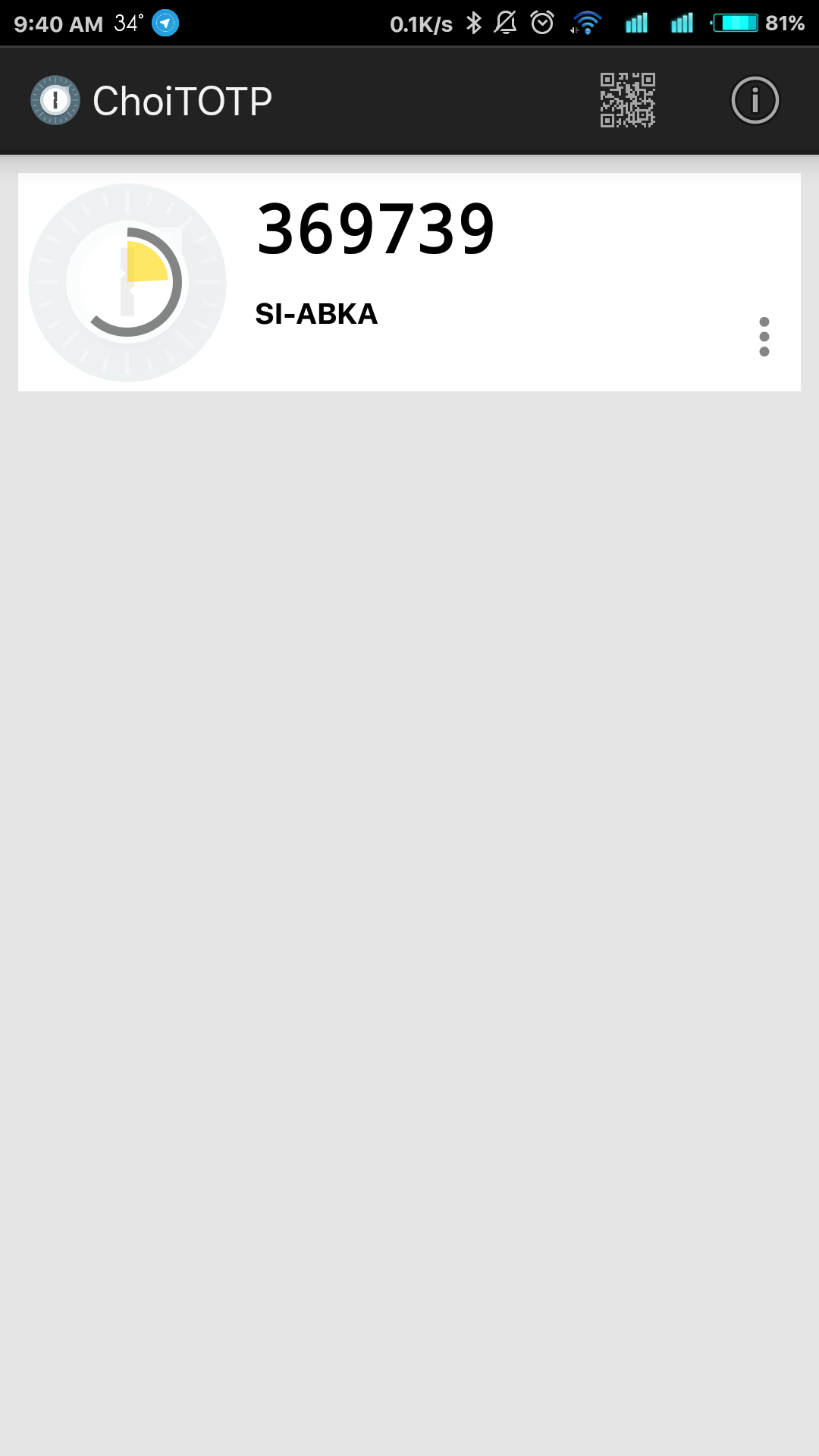
Gambar 4.8. menu aplikasi ChoiTOTP

1. Aplikasi akan mengaktifkan kamera belakang dan akan user perlu mengarahkan kamera ke qrcode seperti pada tampilan seperti Gambar 4.8



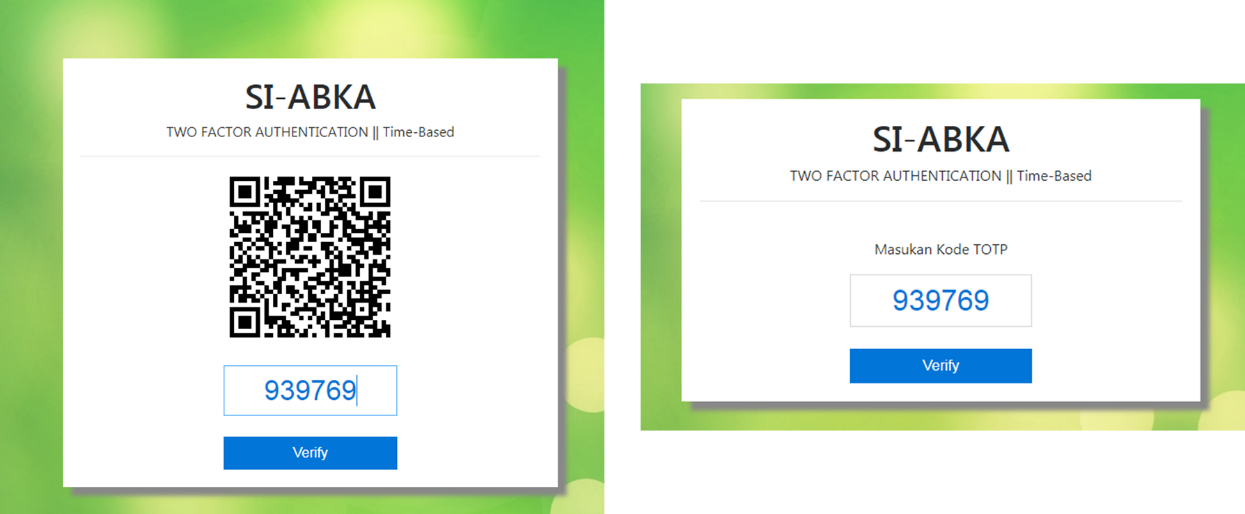
Gambar 4.9. tampilan scan qr-code

* 1. Setelah berhasil di pindai maka aplikasi akan menampilkan enam digit angka yang dapat digunakan untuk login seterusnya. Saat dibutuhkan pengguna tidak perlut melakukan pemindaian lagi. Aplikasi hanya akan membangkitkan kode TOTP saat aplikasi dibuka, jadi aplikasi tersebut tidak membenani sistem android maupun server.

**

Gambar 4.10.kode TOTP berhasil didapatkan

* 1. Enam digit angka tersebut daat digunaka pada form input seperti pada Gambar 4.10. Angka tersebut akan berubah setiap 30 detik, jadi jika sudah habis masa berlakunya pengguna perlu melihat kembali kode TOTP yang ada di aplikasi tersebut.

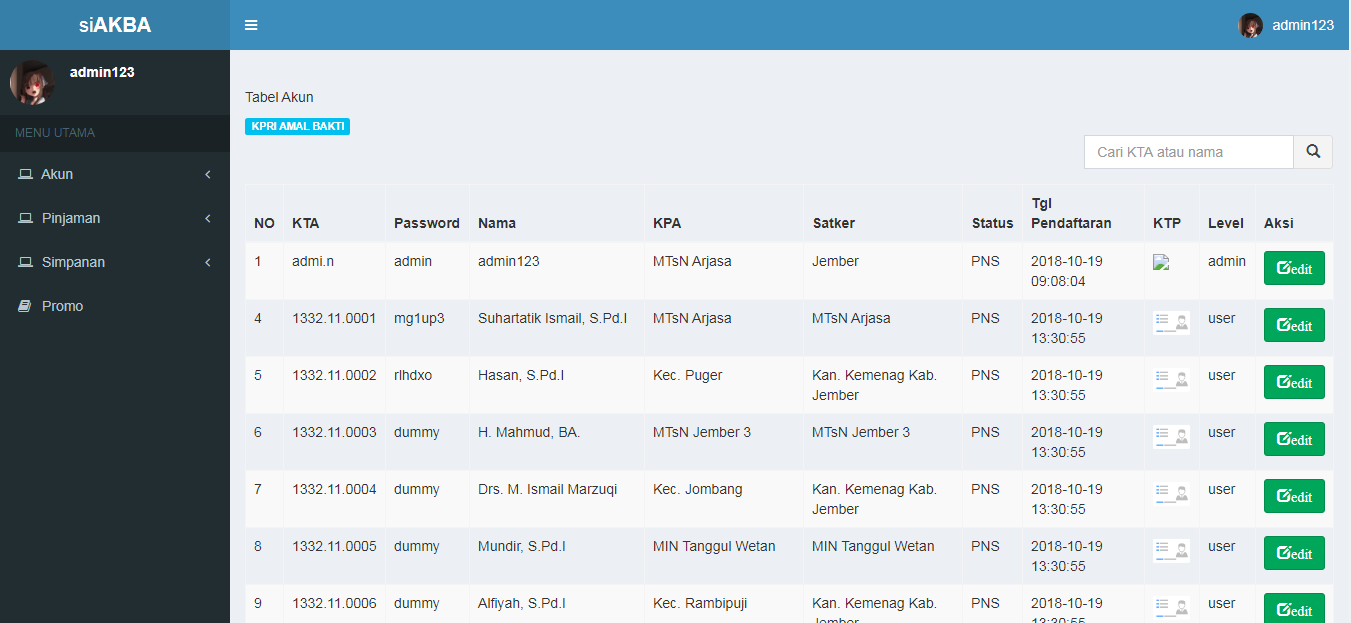


Gambar 4.11. memasukan kode TOTP

* 1. SI-Abka akan akan mengarahkan ke dashboard masing-masing sesuai dengan levelnya saat kode tersebut cocok.

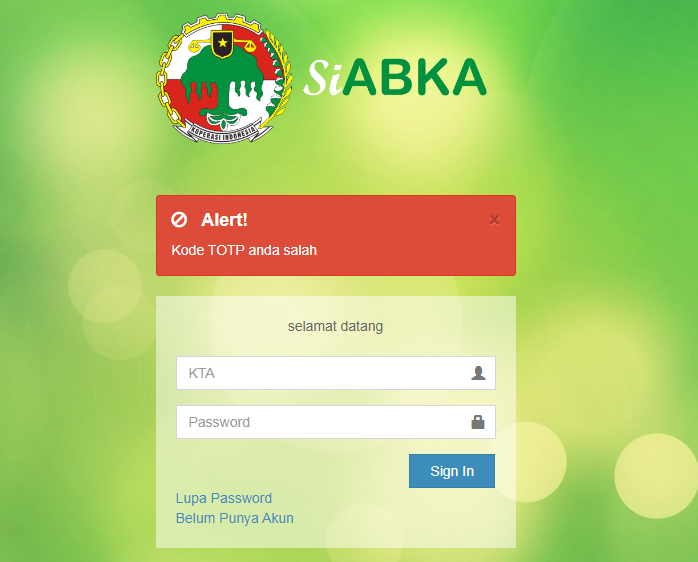


Gambar 4.12.dasboard user



Gambar 4.13. dashboard admin

* 1. Sebaliknya saat kode tersebut tidak cocok maka user akan di arahkan kembali ke tampilan login dan sistem menampilkan pemberitahuan bahwa kode tersebut salah.



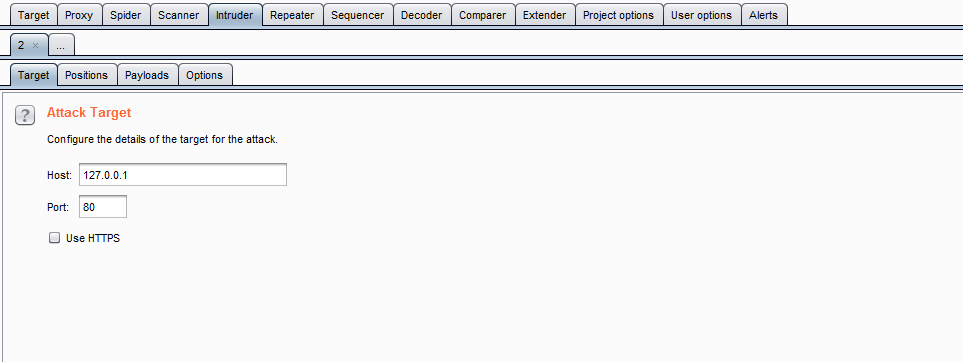
Gambar 4.14. pemberitahuan kode OTP salah

## Hasil pengujian uji keamanan

Pengujian keamanan ini berfungsi untuk mengetahui tingkat keamanan jika web sistem SI-abka diserang. Pengujian ini menggunakan dua kasus yaitu antara SI-abka yang mengimplementasikan *two factor authentication* dan algortima RSA, dengan SI-abka tanpa tambahan modul tersebut.

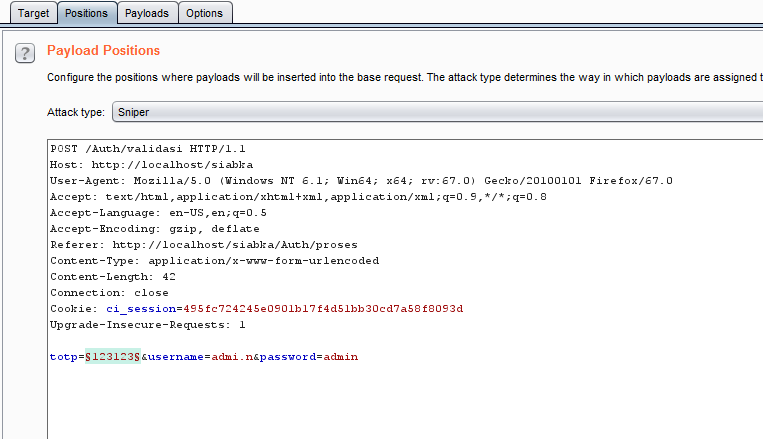
### Pengujian brute force

Tahap pertama pada uji brute force adalah persiapan host dan port target. Host target merupakan alamat target brute force sedangkan port adalah tujuan data tersebut dikirimkan. Pada penelitian kali ini memakai host localhost dan port 80.



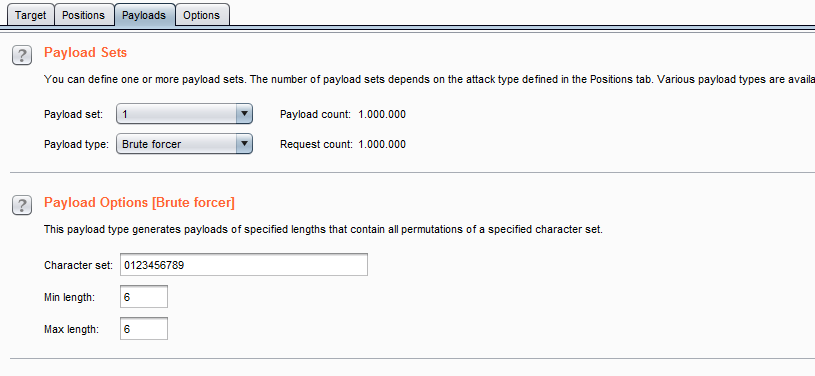
Gambar 4.15. penentuan target sistem

Pada tab *positions* diisi dengan file *header* yang telah di dapatkan sesuai dengan rancangan sebelumnya. File header tersebut berisi data akun yang akan dimanfaatkan brupsuite untuk membuat form brute force. Terdapat 3 data utama yang dikirim yaitu kode totp, KTA, dan password. Kode TOTP akan diisi dengan data hasil generate dan melakukan perulangan mulai dari 0000-9999.



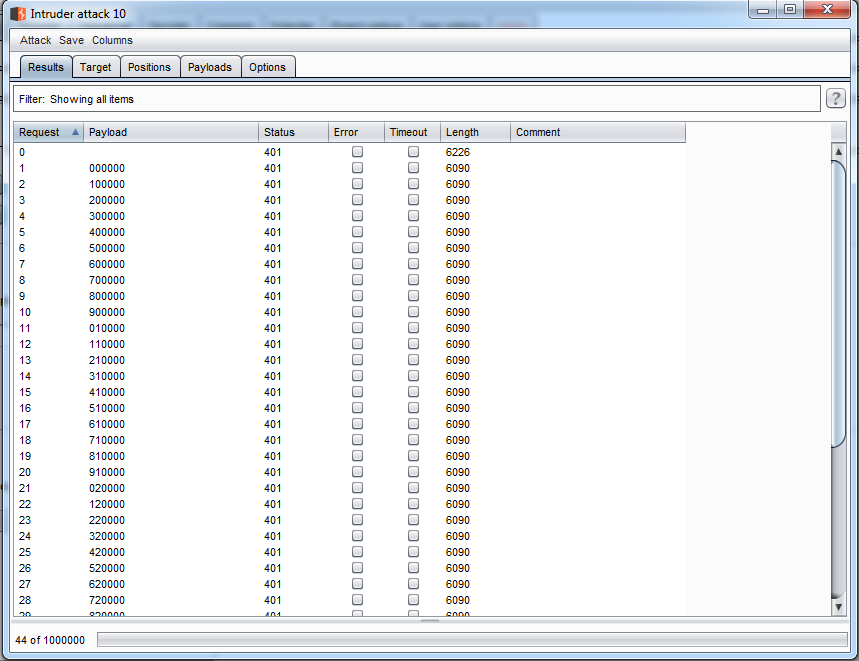
Gambar 4.16. variabel header

Tab *payloads* merupakan aturan *brute force* yang akan diisi ke dalam header. Pada pengujian kali ini memerlukan 6 digit angka bilangan cacah untuk OTP. Ada dua komponen utama pada *tab payloads* yaitu *payloads sets* dan *payloads options. Payloads sets* merupakan jenis penyerangan yang dipilih, pada penelitian ini menggunakan *brute force.* Sedangkan untuk *payloads options* merupakan pengaturan penyerangan yang digunakan untuk jenis penyerangan yang telah dipilih pada *payloads sets.*



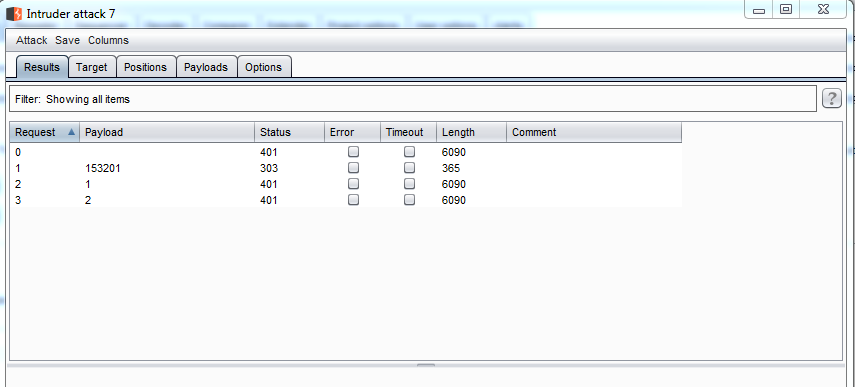
Gambar 4.17. variabel payload brute force

seluruh data sudah sesuai maka klik start attack. Kecepatan pengiriman brute force tergantung terhadapat kecepatan koneksi internet,kecepatan proses komputer penyerang, dan kecepatan proses server. Berikut hasil *brute force*  dalam 30 detik pertama.



Gambar 4.18. hasil brute force

Dalam 30 detik burpsuite berhasil mengirimkan 44 file header yang telah di modifikasi. Dari 44 header yang dikirim semua di tolak oleh sistem ditandai dengan status 401 (unauthorized). Jika kode OTP benar makan akan mendapatkan respons code 303 (redirect) seperti gambar 4.19 dibawah ini

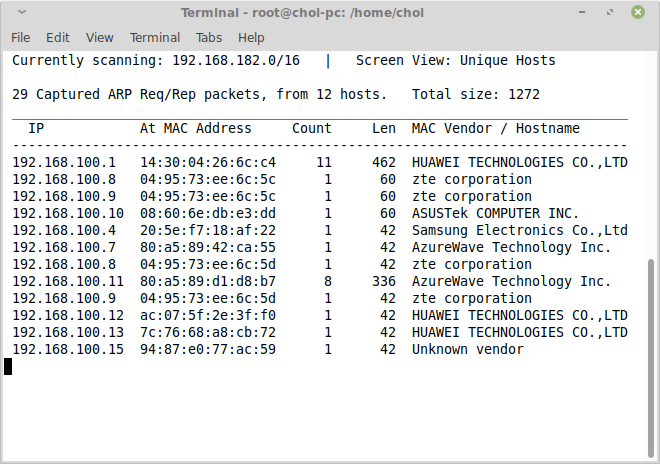


Gambar 4.19. respons code OTP benar

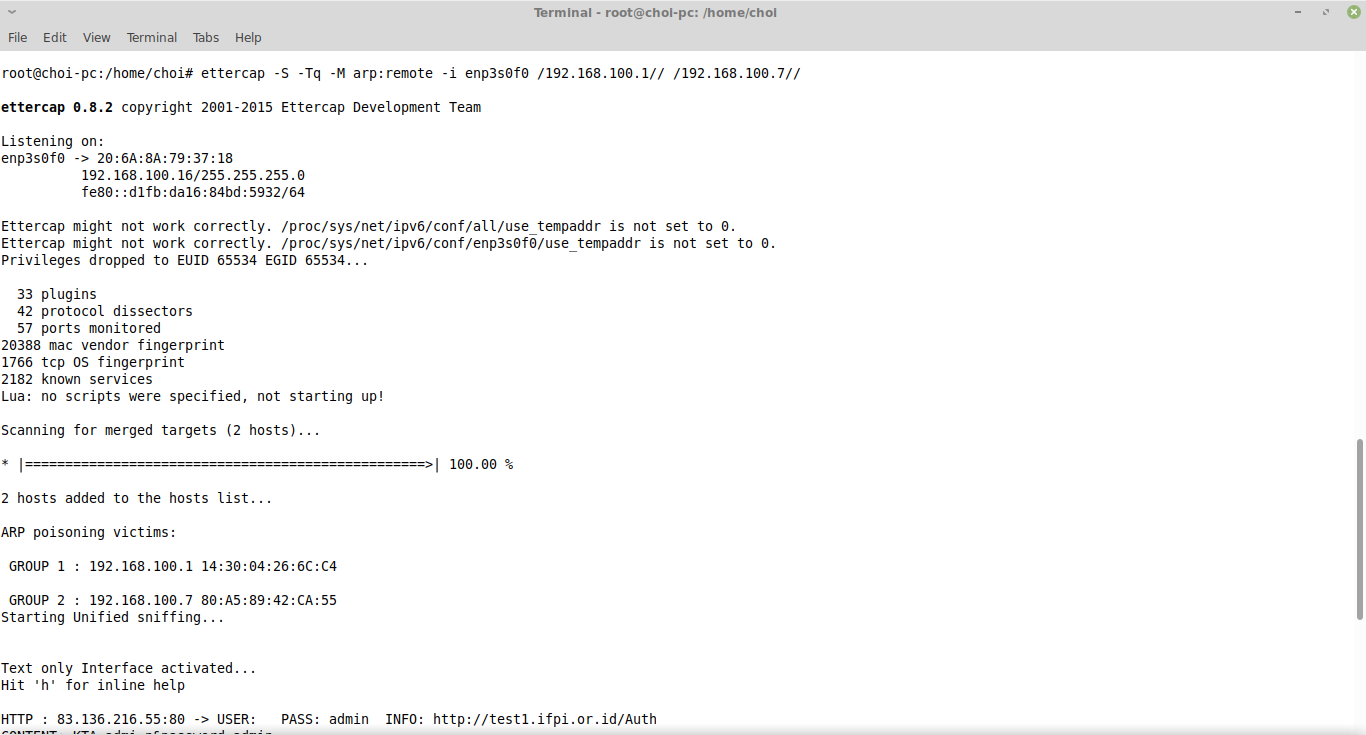
Pengujian dilakukan hanya dalam 30 detik karena setiap 30 detik kode OTP tersebut akan berganti secara terus menerus. Dan untuk pengujian kali ini menggunakan localhost karena saat menggunakan hosting di internet maka akan lebih lambat lagi dalam pengiriman header tersebut dikarenakan spesifikasi server tidak sama. Dari pengujian tersebut didapat dalam 30 detik burpsuite hanya mampu mengirimkan 44 header dari 1 juta kemungkinan header yang ada. Hal ini berarti hanya 0.044 % kemungkinan kode OTP tersebut benar dalam 30 detik. Presentase kemungkinan sangat kecil untuk dapat ditebak dan dapat dikatakan sangat aman. Banyaknya header yang dikirim juga dapat dipengaruhi oleh spesifikasi dari komputer penyerangan dan server target.

### Pengujian MITM

Pengujian teknik MITM dapat dilakukan dengan syarat target dan penyerang berada dalam satu jaringan lokal. Setelah yakin kedua device terkoneksi dalam satu jaringan yang sama maka penyerang dapat mencari ip target dengan bantuan tools netdiscover .



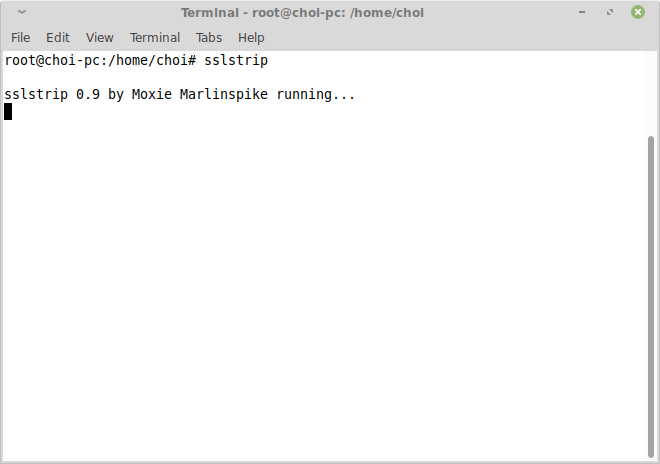
Gambar 4.20. Proses Scan Ip Target

Setelah mendapatkan ip target maka kita perlu mengelabui router dan target dengan menggunakan tools ettercap pada penelittian kali ini didapatkan ip target adalah 192.168.100.7 sesuai dengan tanda hijau pada Gambar 4.20.

Gambar 4.21. Arp Poisoning Ip

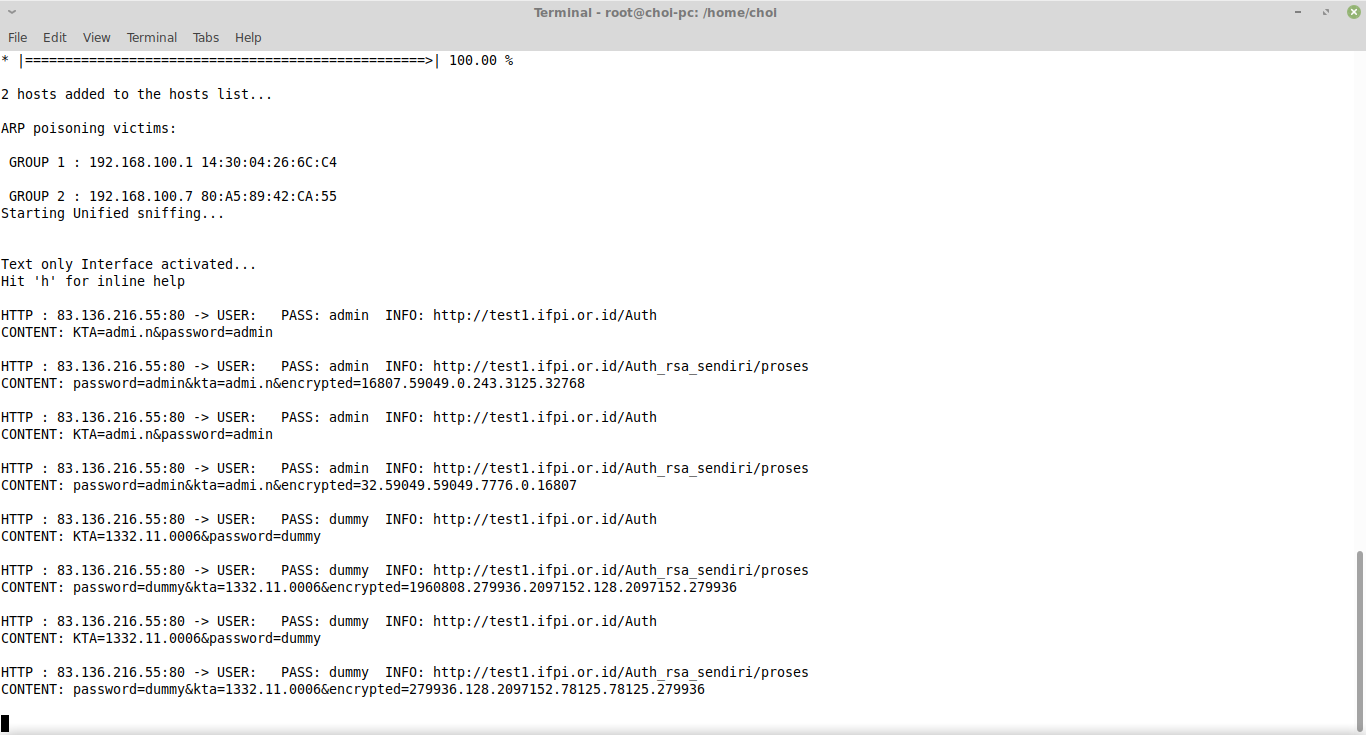
Setelah ip target didapatkan maka perlu dilakukanya proses pembacaan data yang dikirim oleh target ke router. Perintah diatas berfungsi untuk membaca setiap data yang dikirim oleh user ke router .Dalam tahap ini kita sudah berhasil berada di tengah-tengah target dan router. Agar router dan target dapat berkomunikasi maka kita perlu menyalurkan datanya sekaligus dapat membaca arus data. dampak dari pembacaan ini adalah koneksi internet target akan terputus dan perlu dilakukanya routing .Untuk dapat menyalurkan data kita perlu melakukan routing terlebih dahulu seperti pada Gambar 4.22

Gambar 4.22. Routing Ip Target

Setelah target dan router dapat saling berkomunikasi kita perlu membelokan traffict https ke http agar dapat mudah dibaca oleh wireshark. Maka kita membutuhkan tools sslstrip. Sslstrip berfungsi untuk membelokan traffic https ke http dengan mengubah request user sebelum dikirim ke server.

Gambar 4.23. pembelokan traffic https ke http

Dalam tahap ini kita tinggal menunggu target melakukan pengiriman data ke server. Hasil pembacaan adalah berupa data post yang dikirim oleh user ke server. Pada penelitian kali ini data post yang dikirim berupa KTA,password, dan kode TOTP. Data KTA dan password sengaja dibuat plain agar dapat terlihat perbedaan nya dengan kode TOTP yang sudah di tambah RSA. Contoh hasil pembacaan data dengan teknik MITM dapat dilihat pada Gambar 4.24.

Dari hasil pembacaan diatas dapat kita lihat bahwa username , password, dan kode otp dapat dilihat atau dibaca. Tetapi kode OTP telah di enkripsi dengan algoritma RSA sehingga tidak mudah dibaca oleh penyerang.

Gambar 4.24. hasil pembacaan MITM

Berikut Tabel hasil pengiriman yang berhasil di tangkap oleh penyerang dengan melakukan 4 kali percobaan

Tabel 4.2 Hasil MITM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IP server | url target | content | | | Plain (OTP) |
| KTA | Password | Encrypted (OTP) |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | admi.n | admin |  |  |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | admi.n | admin | 16807.59049.0.243.3125.3278 | 790358 |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | admi.n | admin |  |  |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/a>uth\_rsa\_sendiri/proses | admi.n | admin | 32.59049.59049.7776.0.16807 | 299607 |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | 1332.11.0006 | dummy |  |  |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | 1332.11.0006 | dummy | 1960808.279936.2097152.128.2907512.279936 | 968286 |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | 1332.11.0006 | dummy |  |  |
| 83.136.216.55.80 | <http://test1.ifpi.or.id/>auth\_rsa\_sendiri/proses | 1332.11.0006 | dummy | 279936.128.2097152.78125.78125.279936 | 628556 |

Hasil percobaan pada Tabel 4.2 dapat dilihat bahwa kode OTP dari 4 kali percobaan berhasil di enkripsi oleh RSA. Hasil enkripsi berupa 6 kelompok angka yang dipisahkan oleh tanda titik (.). Kolom encrypted merupakan hasil enkripsi dari kolom OTP dan merupakan data yang dikirimkan oleh user ke server. Jadi meskipun data tersebut dapat dilihat oleh orang lain selama orang tersebut tidak mengetahui secret key maka data tersebut aman.

Saat server menerima data yang telah dienkripsi maka server akan melakukan proses dekripsi. Proses dekripsi dilakukan oleh server menggunakan secret key masing-masing akun. Setelah kode TOTP berhasil di dekripsi maka selanjutnya server melakukan pengecekan apakah kode TOTp tersebut sesuai dengan kode yang dibangkitkan.

## Pembahasan

Hasil dari penelitian ini adalah library metode two factor autentication yang menggunakan TOTP dan algoritma RSA sebagai pengaman dalam mengirimkan data. Library ini dapat diimplementasikan pada semua website yang membutuhkan pengamanan tambahan. Pada SI-Abka memiliki dampak yang signifikan yaitu saat pengujian brute force dan man in the middle data akun pengguna aman dan penyerang tidak dapat masuk ke dalam sistem.

Pada penelitian ini percobaan dengan brute force berhasil mengiriman 44 kemungkinan header dari satu juta kemungkinan yang ada setiap 30 detik. Presentase keberhasilan tertebaknya kode adalah 44/1.000.000 sama dengan 0.044 %. Presentase tersebut sangat kecil kemungkinan tertebaknya kode menggunakan metode brute force.

Banyaknya file header yang dikirim berbandng lurus dengan kecepatan pemrosesan dari komputer penyerang, kecepatan koneksi, dan kecepatan pemrosesan server. Semakin cepat pemrosesan data yang dilakukan semakin banyak pula file header yang berhasil dikirim ke server. Pemrosesan data yang dimaksud adalah pembuatan file header yang akan dikirim serta kecepatan penerimaan file header.

Untuk percobaan dengan man in the middle saat penyerang membaca arus data target data OTP berhasil di enkripsi. Pemanfaatan metode RSA dapat membantu mengamnakn data yang dikirm oleh client yang menuju ke server. Data tersebut berupa 6 digit angka yang dienkripsi dengan rsa menggunakan javascript. Dari 4 kali percobaan seluruhnya berhasil di enkripsi baik dari akun admin atau akun nasabah. Hasil enkripsi OTP dengan RSA berupa angka dengan beberapa kelompok angka yang dibatasi titik (.) dan tidak semuanya berbeda. Angka yang di hasilkan akan berbeda tiap pengguna dikarenakan secret key dan public key tiap akun berbeda-beda.

# PENUTUP



## Kesimpulan

Hasil dari penelitian yang dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Untuk meningkatkan otentikasi sistem SI-Abka diperlukan faktor tambahan agar tidak sembarangan orang dapat masuk ke sistem SI-Abka. Faktor tambahan tersebut adalah menggunakan metode OTP. metode OTP yang digunakan memakai metode TOTP (*Time One Time Password*). Penggunaan TOTP dipilih karena tidak memakan resource yang banyak dan tidak membebani sistem saat menangani arus data yang besar. Resource adalah hal ini meliputi spesifikasi server dan penggunaan pihak ketiga seperti sms gateway dan biaya server tambahan.
2. Dalam penilitian kali ini peneliti melakukan pengukuran tingkat keamanan dari penggunaan two factor autentication . percobaan Pengukuran tingkat keaamanan menggunakan dua teknik yaitu brute force dan man in the middle (MITM). Percobaan tes keamanan sistem dengan teknik brute force menghasilkan dari satu juta kombinasi, komputer peneliti hanya dapat mampu mengirimkan 44 kombinasi dalam 30 detik pertama dan 30 detik berikutnya kode OTP tersebut akan berganti. Presentase kemungkinan tertebaknya kode OTP dalam percobaan tersebut hanya sebesar 0.044 % . dalam percobaan dengan teknik man in the middle peneliti berhasil mendapatkan data yang dikirim oleh client ke server. Data yang didapat berupa nomor KTA, password, dan kode OTP yang valid. Dari ketiga data tersebut kode OTP berhasil di enkripsi dan tidak dapat dibaca oleh peneliti. Kode OTP hanya bisa di baca setelah didekripsi oleh server. Dari percobaan tersebut keamanan data yang dikirim oleh client ke server bisa dikatakan aman

## Saran

Pada sub bab ini akan dijabarkan beberapa saran penilitian untuk penelitian lain yang serupa atau penambahan fitur yang berhubungan dengan aplikasi implementasi *Two Factor Authentication* pada SI-Abka adalah sebagai berikut :

1. Modul yang sudah dibuat masih dapat dimodifikasi agar dapat mengikuti perkembangan proses bisnis perusahaan, modifikasi tersebut bertujuan agar modul tersebut dapat digunakan / diimplementasikan pada sistem yang sudah ada.
2. dibutuhkannya fitur backup agar data-data yang sudah tersimpan lebih aman jika terjadi hal-hal yang tidak diinginkan seperti sistem down, server rusak dll. sehingga file backup dapat digunakan untuk melanjutkan proses bisnis kembali tanpa kehilangan waktu untuk menulis kembali data - data yang hilang.
3. Dalam segi antar muka, modul login masih sederhana dan harus mendapatkan desain yang lebih baik, seperti penempatan tombol dan teks pembantu. Tampilan utama bagi user harus menarik dari segi desain, warna, dan lain - lain.

# DAFTAR PUSTAKA

Arief, A. dan R. Saputra. 2016. Implementasi kriptografi kunci publik dengan algoritma rsa-crt pada aplikasi instant messaging. *Scientific Journal of Informatics*. 3:1.

Fahmy, H. dan N. Elkhateeb. 2018. Proposed model for generation of one time password. *International Journal of Computer Science and Information Security (IJCSIS)*. 16:11.

Imam Santoso, K., E. Sediyono, dan S. Suhartono. 2013. Studi pengamanan login pada sistem informasi akademik menggunakan otentifikasi one time password berbasisis sms dengan hash md5. *JURNAL SISTEM INFORMASI BISNIS*. 3(1):07–12.

Inayatullah. 2007. Analisis penerapan algoritma md5 untuk pengamanan password. *Jurnal Ilmiah STMIK GI MDP*. 3(3):1–5.

Khairina, D. M. 2011. ANALISIS keamanan sistem login. *Jurnal Informatika Mulawarman*. 6:64–67.

Kim, H., Y.-G. Lee, K.-S. Lee, dan M.-S. Jun. 2009. Design and Implementation of Multi Authentication Mechanism for Secure Electronic Commerce. *2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*. 2009. IEEE: 215–219.

Mallik, A., A. Ahsan, M. Md. Z. Shahadat, dan J.-C. Tsou. 2019. Man-in-the-middle-attack: understanding in simple words. *International Journal of Data and Network Science*. 77–92.

M’Raihi, D., S. Machani, M. Pei, dan J. Rydell. 2011. *TOTP: Time-Based One-Time Password Algorithm*. RFC Editor

Muchlis, B. S., M. A. Budiman, dan D. Rachmawati. 2007. Teknik pemecahan kunci algoritma rivest shamir adleman(rsa) dengan metode kraitchik. *Jurnal & Penelitian Teknik Informatika*. 2:2.

Musliyana, Z., T. Y. Arif, dan R. Munadi. 2016. Peningkatan sistem keamanan autentikasi single sign on (sso) menggunakan algoritma aes dan one-time password studi kasus: sso universitas ubudiyah indonesia. *Jurnal Rekayasa Elektrika*. 12(1):21.

Nugraha, M. P. dan R. Munir. 2011. Pengembangan aplikasi qr code generator dan qr code reader dari data berbentuk image. *Konferensi Nasional Informatika*

Pramudita, K. E. 2010. Brute force attack dan penerapannya pada password cracking. *Makalah IF3051 Strategi Algoritma*. sem 1

Rahayu, Y. D., N. Ramadijanti, dan Y. Setiowati. 2006. Pembuatan aplikasi pembacaan quick response code menggunakan perangkat mobile berbasis j2me untuk identifikasi suatu barang. *Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember*

Ramadhan, K. 2010. Pengujian man-in-the-middle attack skala kecil dengan metode arp poisoning. 6.

Rosnawan, D. 2011. Aplikasi algoritma rsa untuk keamanan data pada sistem informasi berbasis web. *Universitas Negeri Semarang.* 1–25.

Sahu, S. K., A. K. Dalai, dan S. K. Jena. 2014. *Varying Password Based Scheme for User Authentication*. Dalam Advanced Computing, Networking and Informatics- Volume 2. Editor M. Kumar Kundu, D. P. Mohapatra, A. Konar, dan A. Chakraborty. Cham: Springer International Publishing.

Sardju, E. R., Ir. R. Magdalena, dan R. Atmaja. 2015. IMPLEMENTASI algoritma rsa untukenkripsi dan dekripsi sms (short message service) pada ponsel berbasis android. *E-Proceeding of Engineering*. 2:2435.

Sudiarto Raharjo, W., I. D. E.K. Ratri, dan H. Susilo. 2017. IMPLEMENTASI two factor authentication dan protokol zero knowledge proof pada sistem login. *Jurnal Teknik Informatika Dan Sistem Informasi*. 3(1)

Suling, C., M. Olivya, dan R. Nur. 2017. Prototype Pengembangan Autentikasi Login Menggunakan Teknologi Quick Respone Code. November 20, 2017

Susanto, S. dan A. Trisusilo. 2018. PENERAPAN algoritma asimetris rsa untuk keamanan data pada aplikasi penjualan cv. sinergi computer lubuklinggau berbasis web. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*. 9:1043–1052.

Ungkawa, U., I. A. Dewi, dan K. R. Putra. tanpa tahun. IMPLEMENTASI algoritma time-based one time password dalam otentikasi token internet banking. *Nstitut Teknologi Nasional Bandung*. 1–10.