



**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 tertebakanya 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.....	ii
PERNYATAAN.....	iv
PENGESAHAN PEMBIMBING	vi
PENGESAHAN PENGUJI	vii
RINGKASAN	viii
PRAKATA	ix
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
BAB 1. PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Batasan masalah.....	2
1.4 Tujuan penelitian	3
1.5 Manfaat Penelitian	3
BAB 2. TINJAUAN PUSTAKA	4
2.1 <i>Password dan Username</i>	4
2.2 Otentikasi	8
2.3 SI-Abka	10
2.4 QR-Code.....	15
2.5 <i>One Time Password (OTP)</i>	Error! Bookmark not defined.
2.6 Time-Based OTP	10
2.7 Algoritma RSA	16
2.8 Brute Force	20
2.9 Man in the middle	22
2.10 Penelitian Terdahulu.....	4
BAB 3. METODOLOGI PENELITIAN	23
3.1 Objek Penelitian.....	23
3.2 Tempat Penelitian	23
3.3 Tahapan Penelitian	23

3.4 Pembuatan modul TOTP dan RSA.....	24
3.4.1. Modul TOTP	26
3.4.2. Modul RSA	26
3.5 Uji keamanan.....	30
3.5.1. Uji brute force.....	31
3.5.2. Uji MITM.....	33
BAB 4. HASIL DAN PEMBAHASAN	35
4.1 Hasil Implementasi Pembuatan Modul TOTP dan RSA.....	35
4.1.1. Modul TOTP	35
4.1.2. Modul RSA	37
4.2 Hasil Pengujian modul TOTP dan Algoritma RSA	39
4.3 Hasil pengujian uji keamanan	46
4.3.1. Pengujian brute force	46
4.3.2. Pengujian MITM	50
4.4 Pembahasan	55
BAB 5. PENUTUP.....	57
5.1 Kesimpulan.....	57
5.2 Saran	58
Daftar pustaka.....	59
LAMPIRAN	Error! Bookmark not defined.

DAFTAR GAMBAR

Gambar 2.1. proses otentikasi	Error! Bookmark not defined.
Gambar 2.2. tampilan login SI-Abka	7
Gambar 2.3. QR-Code.....	Error! Bookmark not defined.
Gambar 2.4. alur OTP	Error! Bookmark not defined.
Gambar 2.5. alur generate kode TOTP	15
Gambar 2.6. proses login dengan OTP	Error! Bookmark not defined.
Gambar 2.7. konsep RSA	17
Gambar 2.8. Flowchart Proses Algoritma RSA.....	18
Gambar 2.9. percobaan brute force.....	21
Gambar 2.10. cara kerja MITM.....	22
Gambar 3.1 Alur Tahapan Penelitian.....	24
Gambar 3.2.. flowchart implementasi secret key TOTP	28
Gambar 3.3. alur login TOTP dan RSA.....	30
Gambar 3.4. tampilan pembacaan request header.....	32
Gambar 4.1. halaman pembuatan secret key kode TOTP.....	35
Gambar 4.2. hasil pembangkitan secret key	36
Gambar 4.3. hasil pembangkitan secret key , private key dan public key	37
Gambar 4.4. tampilan login SI-Abka	40
Gambar 4.5. input kode TOTP.....	41
Gambar 4.6. input kode TOTP dengan qr-code	41
Gambar 4.7. aplikasi ChoiTOTP	42
Gambar 4.8. menu aplikasi ChoiTOTP.....	42
Gambar 4.9. tampilan scan qr-code	43
Gambar 4.10. kode TOTP berhasil didapatkan	43
Gambar 4.11. memasukkan kode TOTP	44
Gambar 4.12. dashboard user.....	44
Gambar 4.13. dashboard admin.....	45
Gambar 4.14. pemberitahuan kode OTP salah	45
Gambar 4.15. penentuan target sistem	46
Gambar 4.16. variabel header	47
Gambar 4.17. variabel payload brute force	47
Gambar 4.18. hasil brute force	48
Gambar 4.19. respons code OTP benar.....	49
Gambar 4.20. proses scan ip target.....	50
Gambar 4.21. arp poisoning ip	51
Gambar 4.22. routing ip target	52
Gambar 4.23. pembelokan traffic https ke http	52
Gambar 4.24. hasil pembacaan MITM	53

DAFTAR TABEL

BAB 1. 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.

1.1 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.

1.2 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?

1.3 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 middle* (MITM).
4. Perubahan kode TOTP setiap 30 detik.

1.4 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.

1.5 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.
2. Bagi Peneliti
Dapat mengetahui dan menerapkan metode TOTP dan algoritma RSA untuk meningkatkan keamanan pada sebuah sistem.
3. Bagi Objek Penelitian
Penelitian yang dilakukan diharapkan dapat memberikan peningkatan keamanan terhadap otentikasi sistem SI-Abka.

BAB 2. 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 penerapannya yang dapat membantu peneliti dalam penelitian ini. Selain pembangkitan OTP penulis juga mempelajari 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.

2.1 Penelitian Terdahulu

Penelitian dengan judul “Implementasi Algoritma Time-Based *One time password* dalam Otentikasi Token Internet Banking” (Ungkawa dkk., 2017) berisi tentang penggunaan algoritma TOTP yang digunakan untuk pengamanan proses transaksi di bank. Kode TOTP tersebut tidak langsung dikirim ke user tetapi mengirim nilai hash nya. Penelitian ini menggunakan hash SHA256 sebagai metode hashingnya dan enkripsi 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) melakukan penelitian tentang proses pengamanan data dengan algoritma RSA pada sistem informasi berbasis web. Fungsi dari RSA pada penelitian adalah untuk menjaga keamanan password dan pesan berupa file, Pada penelitian ini membahas tentang aplikasi pengamanan data pada sistem informasi berbasis web. Permasalahan dalam penelitian 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) membahas tentang proses enkripsi 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 ciphertext 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 authentication* 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. Terdapat tiga syarat keamanan sistem yaitu kerahasiaan, integritas, dan ketersediaan yang harus terpenuhi (Sahu dkk., 2014). Penggunaan TOTP pada sistem otentikasi di SI-Abka dapat membantu sistem untuk memenuhi dua dari tiga syarat standar keamanan sistem yaitu integritas, dan ketersediaan . TOTP berperan sebagai pintu kedua setelah user melakukan login atau saat user melakukan suatu aktifitas yang membutuhkan pengecekan kepemilikan akun. Hal ini dikarenakan saat user melakukan sesuatu maka user perlu memasukkan kode khusus yang tidak diketahui oleh sembarang orang. Sedangkan Metode RSA memenuhi syarat terakhir yaitu kerahasiaan data.

Metode 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.

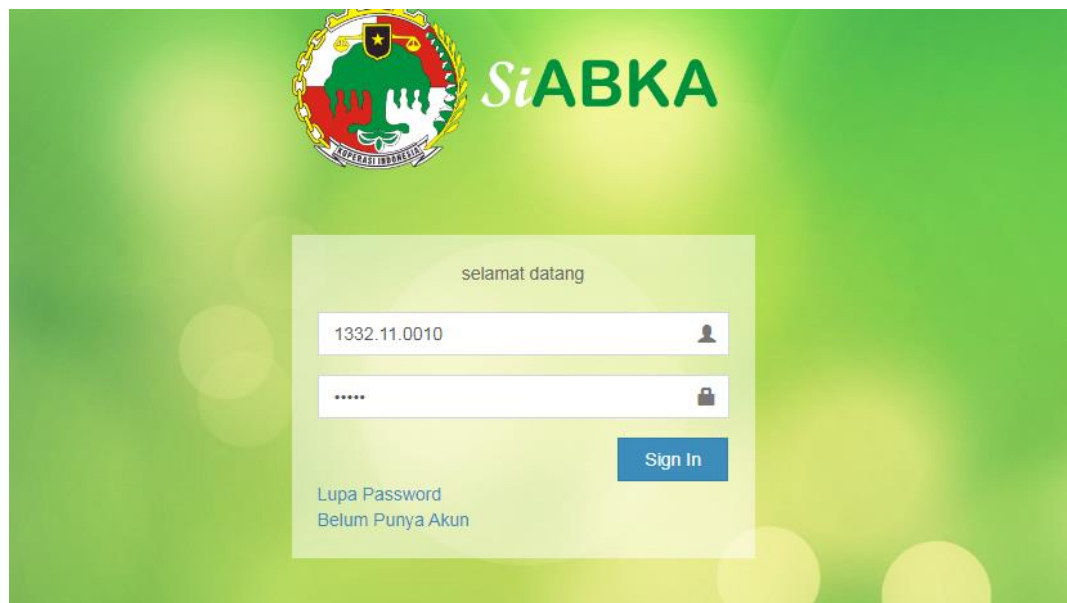
Metode TOTP dan algoritma RSA digabungkan untuk menambahkan tingkat keamanan sistem yang telah ada. Hal ini ditujukan agar data akun yang ada dapat lebih aman dari penggunaan orang yang tidak berkepentingan. Alasan penggunaan metode TOTP adalah resource yang dibutuhkan sedikit dan kecil. TOTP tidak membutuhkan server khusus dan tidak membutuhkan biaya sama sekali dalam pengimplementasiannya. 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 In the Middle). Teknik brute force sesuai karena teknik ini menggunakan metode dictionary attack yaitu melakukan penyerangan dengan daftar tertentu. Daftar tersebut berupa kombinasi 6 digit angka antara 0000-9999. Dari kombinasi tersebut terdapat satu kombinasi yang benar. Dari hal tersebut teknik brute force sangat sesuai karena teknik tersebut melakukan percobaan secara terus menerus sampai mencapai tujuannya. Tujuan tersebut adalah ditemukannya 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.

2.2 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 2.1.



Gambar 2.1. 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 peserta 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.

2.3 Otentikasi

Password adalah kumpulan karakter yang digunakan untuk masuk kedalam 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.

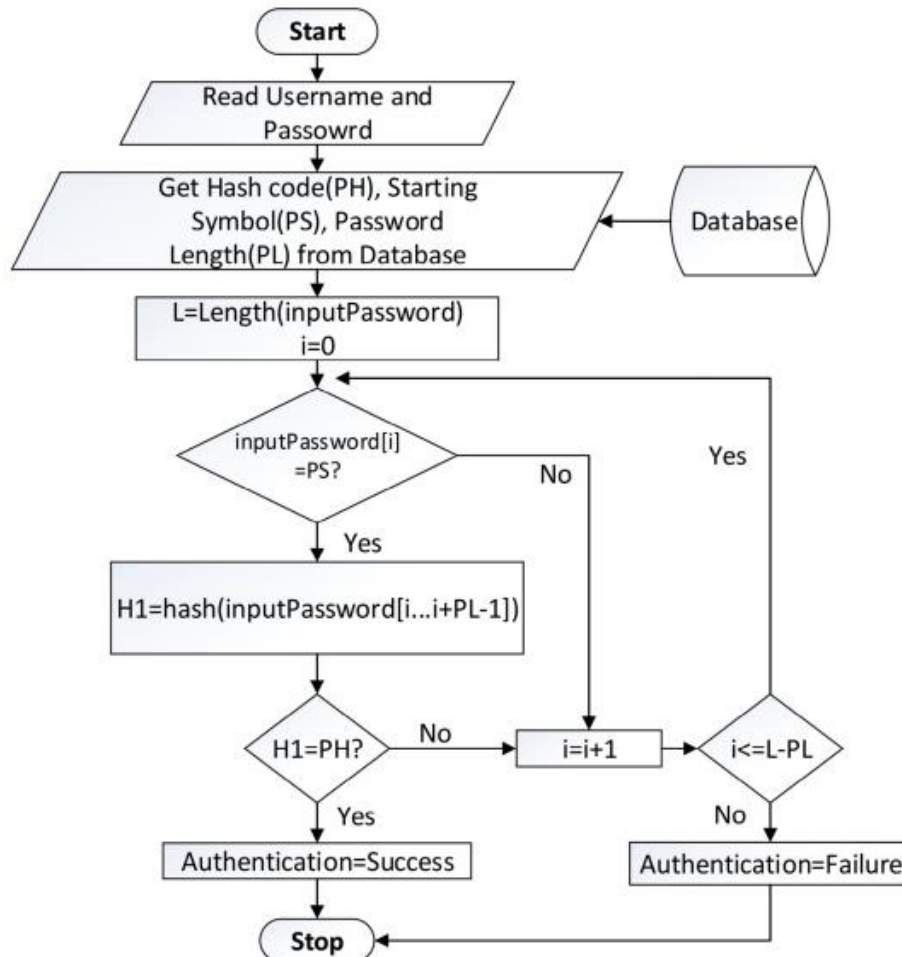
Otentikasi adalah suatu langkah untuk menentukan atau memastikan bahwa seseorang (atau sesuatu) adalah autentik atau asli. Melakukan otentikasi terhadap sebuah objek adalah melakukan konfirmasi terhadap kebenarannya. Sedangkan melakukan otentikasi terhadap seseorang biasanya adalah untuk memverifikasi 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.

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 bukalapak 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 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 berkepentingan. Berikut standar otentikasi yang digunakan kebanyakan sistem saat ini yang menggunakan metode hashing seperti pada Gambar 2.1 .



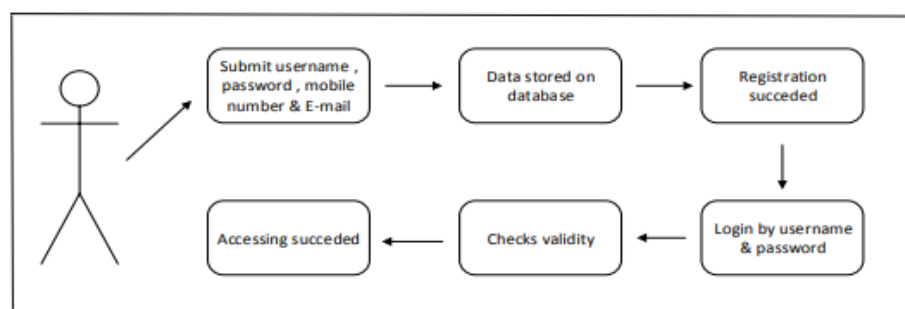
Gambar 2.2. proses otentikasi

Dalam flowchart pada gambar 2.2 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.

2.4 Time Based OTP

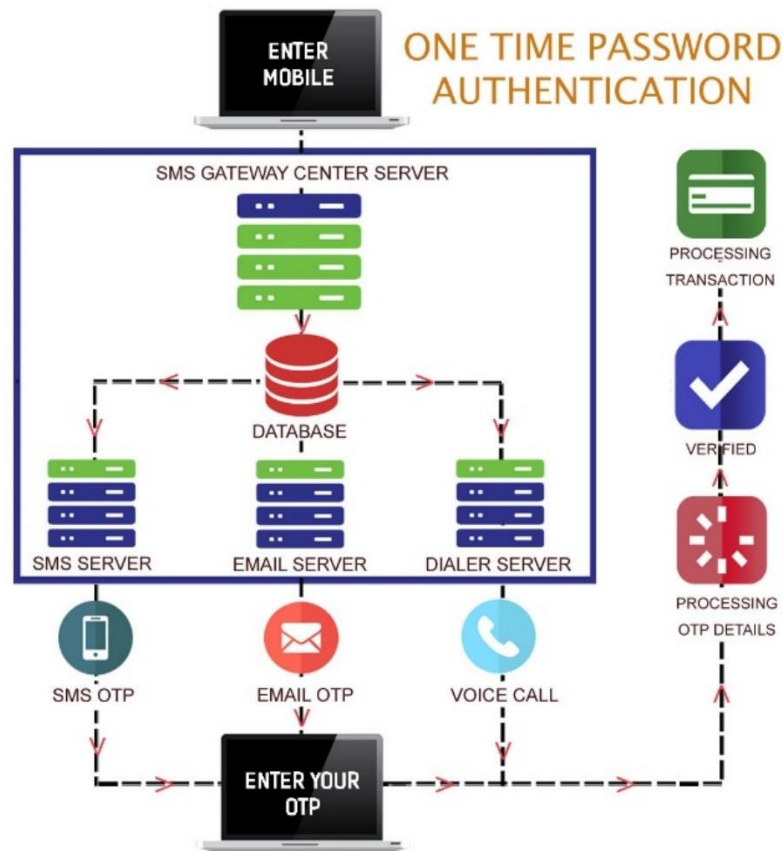
One time password (OTP) merupakan metode otentikasi yang menggunakan *password* yang selalu berubah setelah setiap kali *login*, atau berubah setiap interval waktu tertentu (Musliyana dkk., 2016). *One time password* ini haruslah *password* yang acak sehingga sulit ditebak oleh orang lain. Keuntungan

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. Kode OTP digunakan saat pengguna selesai melakukan otentikasi dengan menggunakan username dan password seperti pada gambar 2.3.



Gambar 2.3. alur otp

Penggunaan OTP membantu mengamankan transaksi atau penggunaan akun untuk memverifikasi pemilik asli akun tersebut. Setiap akun yang teregistrasi akan diwajibkan untuk memasukan kode OTP sehingga dipastikan akun tersebut digunakan oleh pengguna yang bersangkutan. Pada OTP konvensional perlu adanya proses pembangkitan kode OTP. Pembangkitan tersebut bisa berasal dari pengacakan atau algoritma tertentu. Kode yang berhasil dibangkitkan perlu adanya proses pengiriman ke pengguna. Pengiriman kode dari server ke pengguna memiliki banyak cara seperti pada gambar 2.3 dibawah ini.



Gambar 2.4. alur pengiriman kode OTP

Pengiriman kode OTP kepada pengguna memiliki banyak cara seperti pada gambar 2.4 diatas. Cara tersebut antara lain menggunakan sms, email dan menggunakan telfon secara langsung. Tiap jenis pengiriman kode OTP memiliki server khusus seperti sms membutuhkan sms gateway untuk mengirim dan menerima 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.

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.

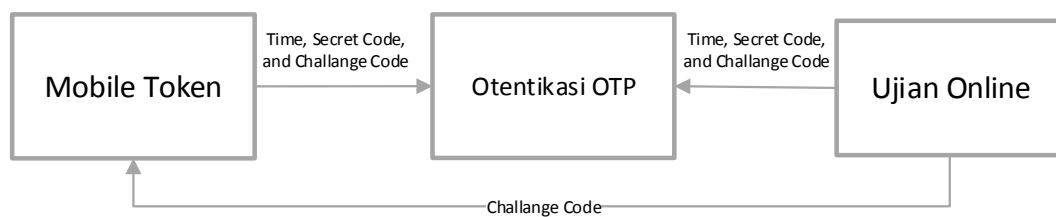
Cara ketiga adalah menggunakan telfon secara langsung yaitu pengguna akan menerima telepon dan diperdengarkan kode OTP yang ada. Cara kerjanya hampir sama dengan sms yaitu menggunakan nomor telepon yang terdaftar. Tetapi metode ini memiliki kelemahan yaitu pengguna diwajibkan menjawab telepon yang masuk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

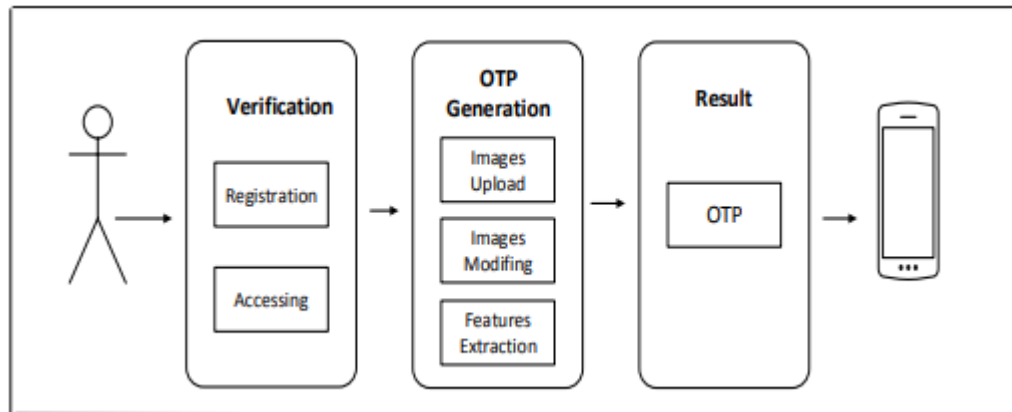
Setiap token memiliki sebuah jam akurat yang telah disinkronisasikan dengan waktu yang terdapat pada server otentikasi. Pada sistem OTP ini, waktu merupakan bagian yang penting dari algoritma password, karena pembangkitan password baru didasarkan pada waktu dan kunci rahasia saat itu dan bukan pada password sebelumnya. Alur TOTP dapat dilihat pada gambar 2.5



Gambar 2.5. alur TOTP

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 2.6. alur generate kode TOTP

Pada Gambar 2.6 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.

TOTP berupa 6 digit angka yang dibangkitkan berdasarkan waktu dan secret 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 selalu berubah-ubah tiap 30 detik sekali. 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.

2.5 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 2.7.



Gambar 2.7. QR-code

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).

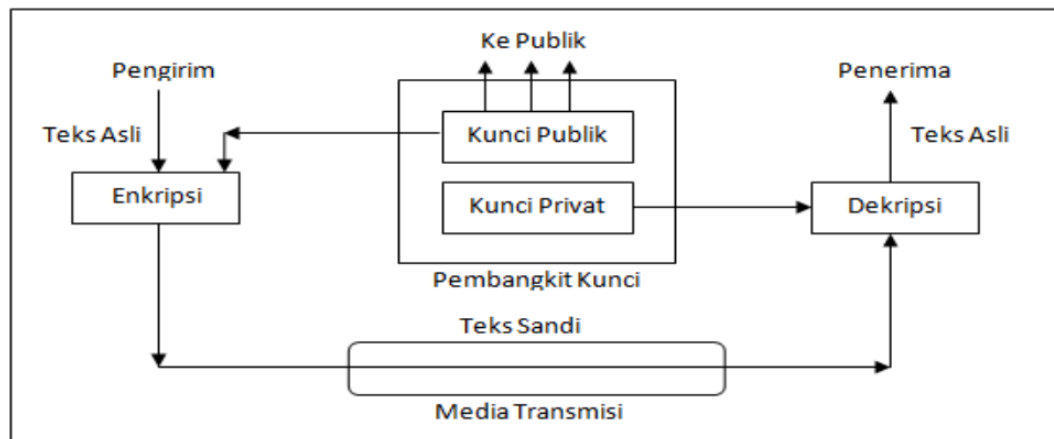
Penggunaan QR-code mempermudah dalam proses memasukan suatu teks dalam hal ini secret key. Jumlah karakter secret key adalah berjumlah 32 karakter dan gabungan dari huruf besar, angka dan karakter. Nilai secret key tersebut akan menyulitkan pengguna dalam memasukan kedalam aplikasi TOTP. Oleh karena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

2.6 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) tetapi memanfaatkan penggunaan kunci publik dan kunci privat pada proses enkripsi dan dekripsinya (Muchlis dkk., 2007). Algoritma ini membantu dalam pengiriman kode

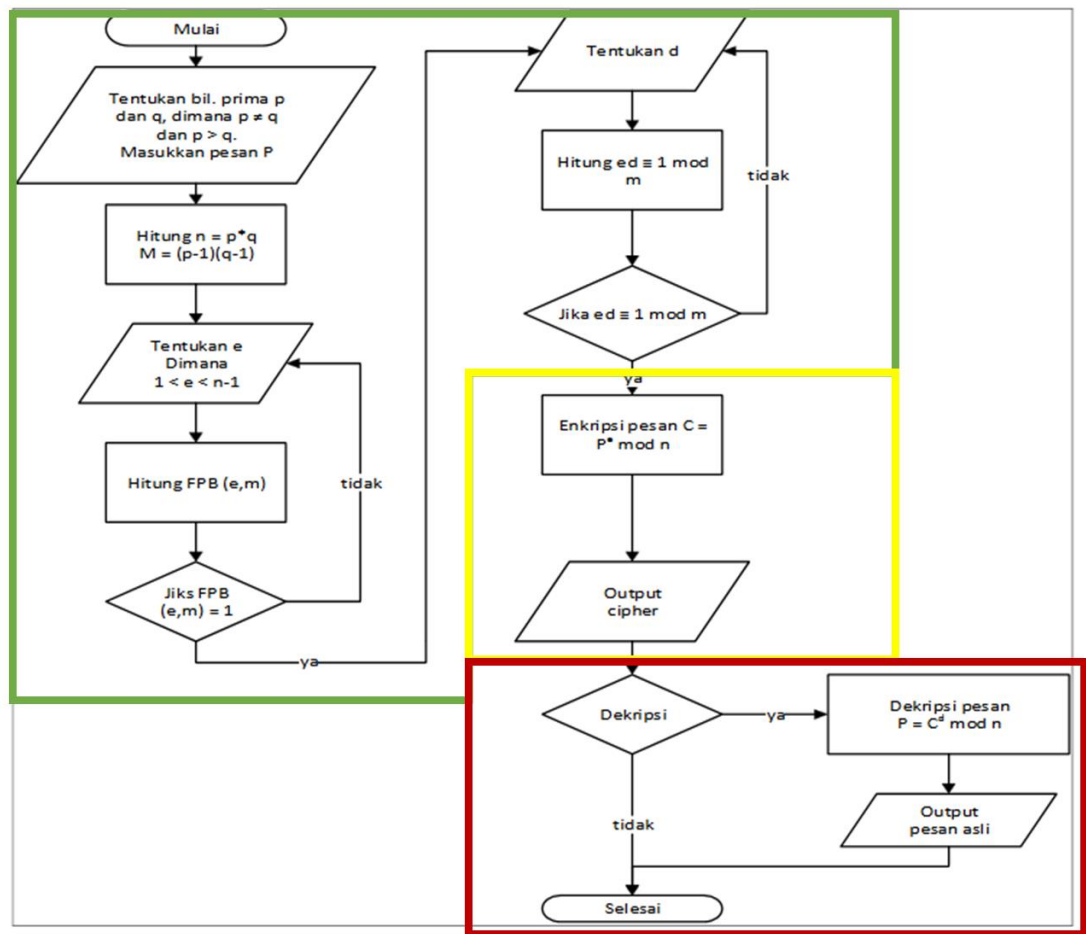
OTP agar lebih aman dan tidak mudah di dibaca. Oleh karena itu dibutuhkan algoritma kriptografi agar hasil OTP lebih aman.

Konsep RSA adalah mengenkripsi teks dengan menggunakan public key dan untuk dapat membaca hasil enkripsi menggunakan private key, public key dapat di sebar dan di beritahukan kepada semua orang, sedangkan private key harus disimpan oleh pembuatnya. Algoritma RSA menggunakan 3 angka (n , 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 2.8



Gambar 2.8. 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 2.9.



Gambar 2.9.Flowchart Proses Algoritma RSA

Terdapat 3 proses yang dilakukan dalam proses algoritma ini yaitu :

a) 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 $n = p * q = 13 * 17 = 221$. Nilai n digunakan untuk kunci public.
3. Hitung $\phi(n) = (p - 1) * (q - 1)$. Setelah $\phi(n)$ telah dihitung, p dan q dapat dihapus untuk mencegah diketahuinya oleh pihak lain.

$$\phi(n) = (p - 1) * (q - 1)$$

$$\phi(n) = (13 - 1) * (17 - 1)$$

$$\phi(n) = 12 * 16$$

$$\phi(n) = 192$$

4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya e , yang relatif prima terhadap m (relatif prima berarti $\text{GCD}(e, m) = 1$) dengan syarat $e \neq (p - 1)$, $e \neq (q - 1)$, dan $e < n$. Misal $e = 5$;
5. Cari kunci d dengan menggunakan algoritma eukledean yang di perluas, yaitu dengan cara berikut :

$$192 = 38 * 5 + 2 \quad \rightarrow n = 1, a_1 = 5, q_1 = 38$$

$$5 = 2 * 2 + 1 \quad \rightarrow n = 2, a_2 = 2, q_1 = 2$$

$$2 = 2 * 1 + 0 \quad \rightarrow n = 3, a_3 = 1, q_3 = 2$$

$$t_0 = 0;$$

$$t_1 = 1;$$

$$t_2 = t_0 - q_1.t_1 = 0 - 38(1) = -38$$

$$t_3 = t_1 - q_2.t_2 = 1 - 2(-38) = 77$$

didapatkan kunci $d = 77$

6. didapatkan kunci

$$n \text{ (public)} = 221$$

$$e \text{ (enkripsi)} = 5$$

$$d \text{ (dekripsi)} = 77$$

- b) Proses Enkripsi

Setelah 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 :

$$C = P^e \text{ mod } n$$

Menggunakan kunci yang diperoleh di atas kita akan mencoba untuk melakukan enkripsi pesan sederhana. Misalnya pesan yang akan di enkripsi adalah

angka 48 dengan nama P (plain), maka akan diperoleh C (chipper) dengan perhitungan sebagai berikut:

$$C = P^e \bmod n$$

$$C = 48^5 \bmod 221$$

$$C = 254803968 \bmod 221$$

$$C = 29$$

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

c) Proses Dekripsi

Proses dekripsi pesan *ciphertext* menjadi *plaintext* (pesan asli) adalah dengan menggunakan rumus dekripsi RSA. Proses ini sesuai pada Gambar 2.8 pada flowchart yang telah di tandai dengan persegi warna merah. Proses yang akan dilakukan sebagai berikut :

$$P = C^d \bmod n$$

Dengan menggunakan pesan hasil enkripsi dan kunci yang diperoleh di atas dapat dilakukan dekripsi pesan seperti berikut:

$$P = C^d \bmod n$$

$$P = 29^{77} \bmod 221$$

$$P = 68630377364883 \bmod 65$$

$$P = 48$$

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.

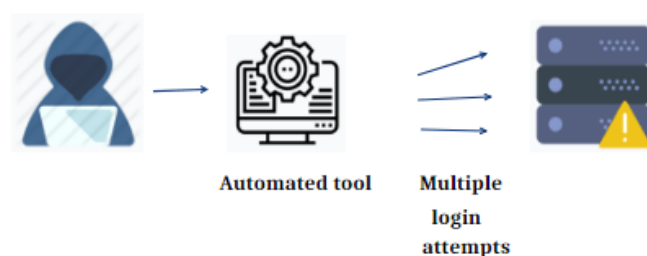
2.7 Brute Force

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 pengujian 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 dikirimkan ke server mulai dari 0000-9999 secara berurutan atau pun acak. Kombinasi tersebut akan dicocokkan dengan hasil pembangkitan kode TOTP oleh sistem. Tujuan brute force adalah menemukan kode TOTP yang sesuai dari satu juta kemungkinan kode yang ada.



Gambar 2.10. percobaan brute force

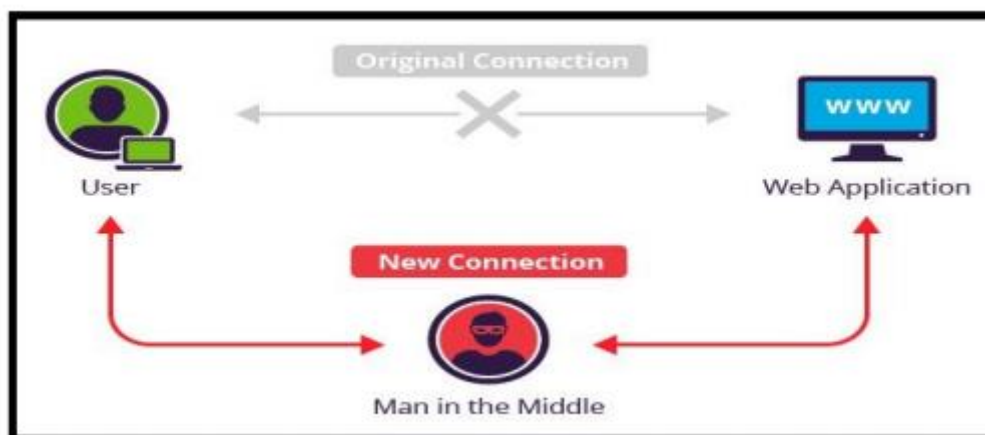
Brute force melakukan percobaan login secara berulang dengan pola tertentu. Pola tersebut bisa dari urutan angka, huruf, karakter atau daftar yang sudah dipersiapkan terlebih dahulu. Dari banyaknya percobaan tersebut akan berdampak terhadap perangkat penyerang maupun target. Perangkat penyerang akan terus menyiapkan request untuk login. Request yang sudah disiapkan 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 memasukkan kata-kata tertentu.

2.8 Man in the middle

Man in the middle (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 seseorang 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 2.11.



Gambar 2.11. 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).

BAB 3. 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.

3.1 Objek Penelitian

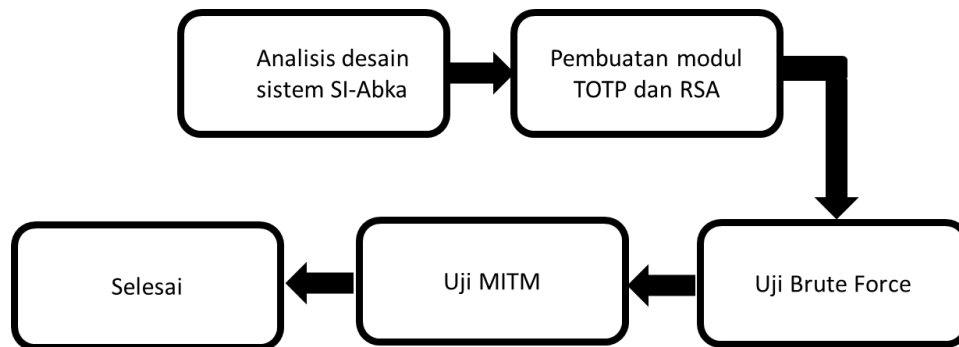
Objek penelitian merupakan sistem SI-Abka yang ada pada koperasi kementerian agama jember. SI-Abka menggunakan username dan password sebagai metode otentikasinya. Dari sistem otentikasi tersebut akan ditambah *two factor authentication* berupa TOTP dan algoritma RSA. Sistem otentikasi tambahan tersebut diharapkan dapat memperkuat keamanan sistem SI-Abka. Kode OTP akan di generate atau dibangkitkan menggunakan aplikasi handphone atau sebuah alat portable. Pembangkitan kode TOTP dilakukan dengan cara memasukan *secret key* ke dalam sistem SI-Abka dan meregistrasikannya ke dalam aplikasi agar dapat selaras.

3.2 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.

3.3 Tahapan Penelitian

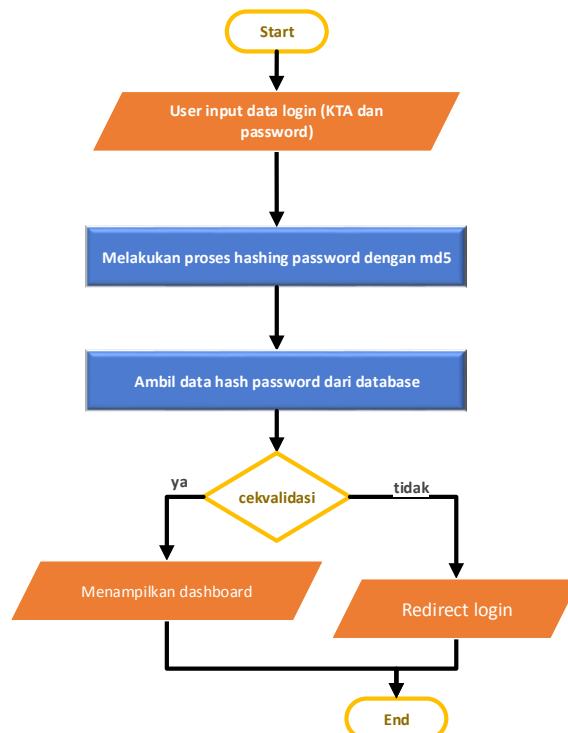
Tahapan Penelitian yang akan dilakukan dapat dilihat dalam diagram alur di bawah ini. Penelitian ini terdiri dari 5 tahap mulai dari perencanaan, implementasi sampai tahap testing. Tahap-tahap ini harus di lakukan secara urut karena tahap sebelumnya berpengaruh ke tahap selanjutnya.



Gambar 3.1 Alur Tahapan Penelitian

3.4 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 dengan menggunakan KTA dan password yang terdaftar. Berikut desain alur otentikasi pada SI-Abka dapat dilihat pada gambar 3.2.

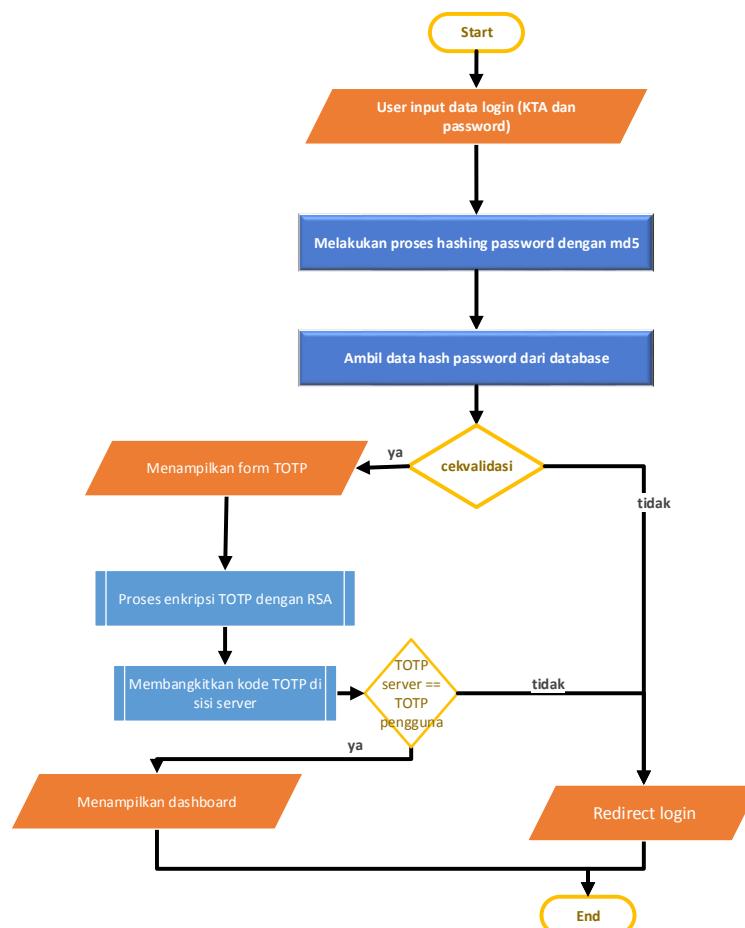


Gambar 3.2. alur otentikasi SI-Abka

Pada alur login tersebut otentikasi merupakan proses membandingkan nilai hash password. Nilai hash password yang dikirim oleh pengguna akan dibandingkan dengan nilai hash yang disimpan di dalam database. Otentikasi dinyatakan benar saat nilai kedua hash sama.

3.5 Pembuatan modul TOTP dan RSA

Proses otentikasi SI-Abka yang sebelumnya memanfaatkan KTA dan password akan mendapatkan modul tambahan untuk menambah tingkat keamanan otentikasinya. Modul tersebut diletakan setelah proses validasi password. proses validasi yang sebelumnya hanya membandingkan nilai hash password akan ditambah dengan proses pencocokan kode TOTP. Berikut posisi modul TOTP dan RSA setelah ditambahkan kedalam alur otentikasi SI-Abka pada gambar 3.3.



Gambar 3.3. proses otentikasi SI-Abka dengan 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 untuk 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.

3.5.1 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 dicocokkan dengan hasil pembangkitan yang dilakukan oleh sistem. Jika sesuai maka perintah akan di loloskan jika tidak maka akan invalid dan gagal.

Sistem otentikasi SI-Abka yang sebelumnya hanya memiliki 4 kombinasi seperti pada Tabel 2.2, maka setelah TOTP ditambahkan maka akan terdapat 8 kombinasi tambahan. Oleh karena itu TOTP di sebut dengan two factor authentication atau faktor tambahan untuk otentikasi. Kombinasi tersebut dapat dilihat pada Tabel 3.1 dibawah ini.

Tabel 3.1. Tabel kombinasi TOTP

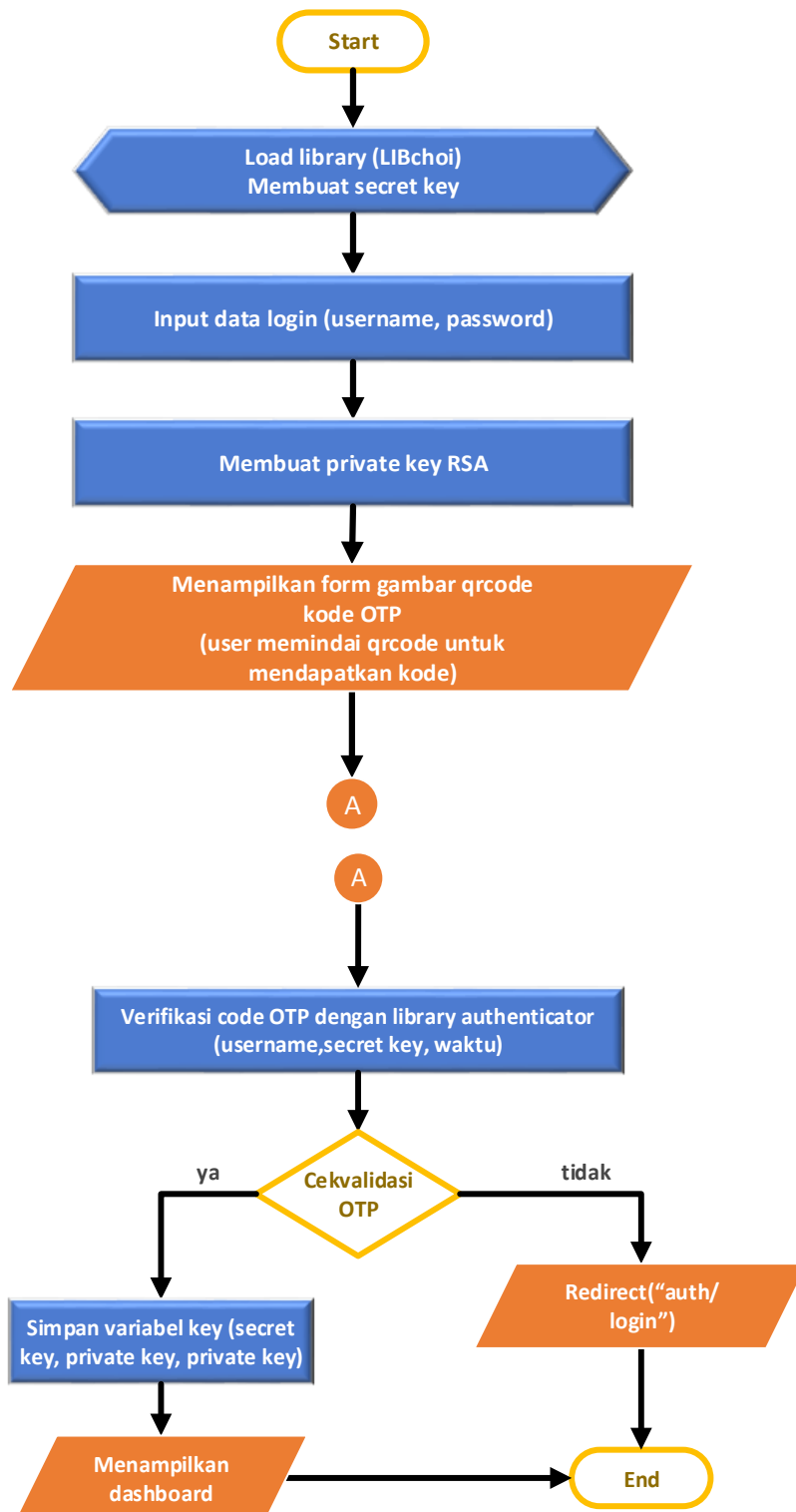
Username	Password	OTP	Status
Salah	Salah	Salah	Gagal
Salah	Salah	Benar	Gagal
Salah	Benar	Salah	Gagal
Salah	Benar	Benar	Gagal

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 memasukkan kode OTP. jika saat diarahkan untuk input kode OTP data user tidak memiliki secret key TOTP maka sistem akan membangkitkan secret key dan mengirimkan qr-code untuk dapat di scan dan dibaca oleh android 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 konfirmasi.

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 qr code yang harus discan atau di pindai dengan aplikasi tertentu. Setelah di pindai maka aplikasi 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. Proses tersebut dapat dilihat pada gambar 3.4



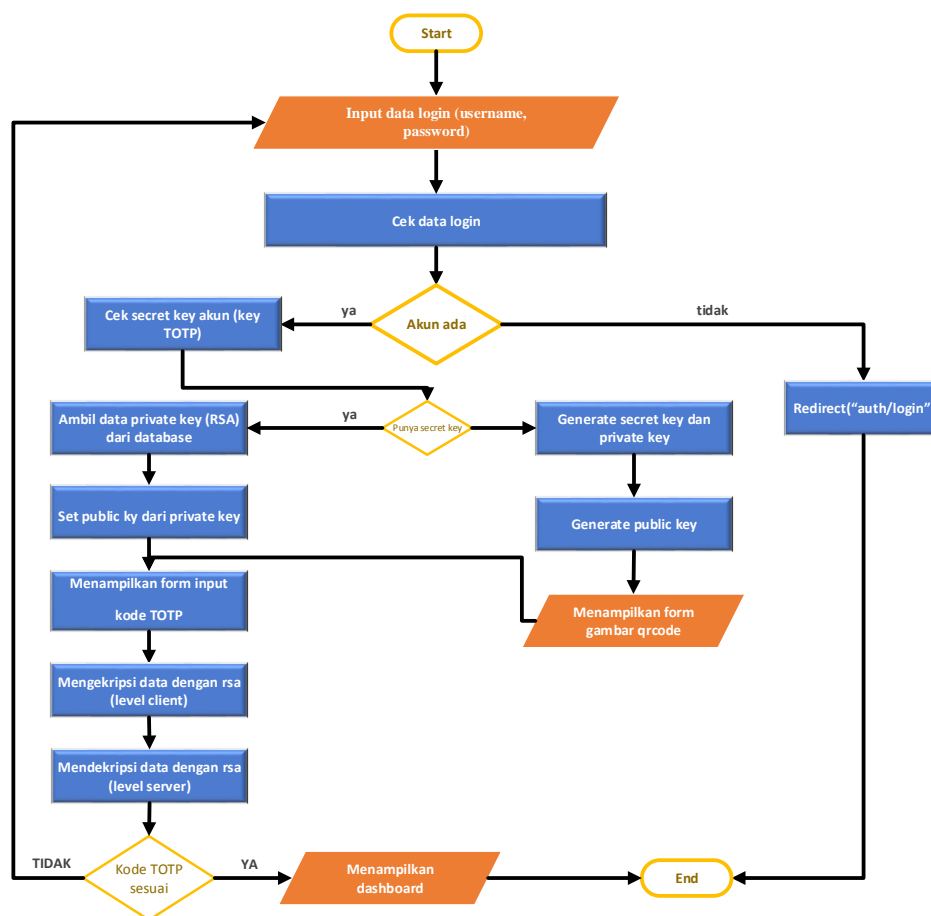
Gambar 3.4.. flowchart implementasi secret key TOTP

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

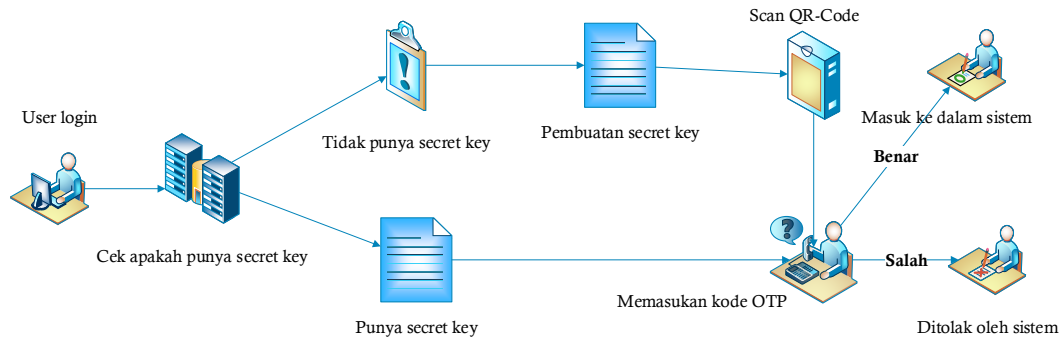
3.5.2 Modul RSA

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 terenkrpsi dengan RSA. Setelah data di terima server data tersebut akan didekripsi dengan pasangan private key nya (lihat gambar 3.5).



Gambar 3.5. flowchart pembangkitan puvlic dan private key

Dari flowchart tersebut dibuat alur proses login dengan TOTP dan RSA seperti pada Gambar 3.6.



Gambar 3.6. 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 secret key yang digunakan untuk proses pembangkitan TOTP. Jika sudah punya maka sistem akan menampilkan halaman untuk memasukkan kode TOTP. Jika tidak maka sistem akan membuat secret key baru dan menampilkan gambar QR-Code yang dapat di scan dengan aplikasi yang sudah disediakan. Saat QR-Code ditampilkan pada halaman juga terdapat form input kode TOTP. Saat user memasukkan kode TOTP sistem akan membangkitkan kode TOTP juga sesuai dengan secret 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.

3.6 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.

3.6.1 Uji brute force

Brute force merupakan algoritma sederhana dalam proses pembuatan kemungkinan 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 bantuan aplikasi burpsuite. Cara kerja aplikasi tersebut adalah memotong jalur komunikasi dan melihat data yang dikirimkan. 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 positif dan nol. Total kemungkinan variasi kode OTP berjumlah satu juta mulai dari 000000 – 999999.

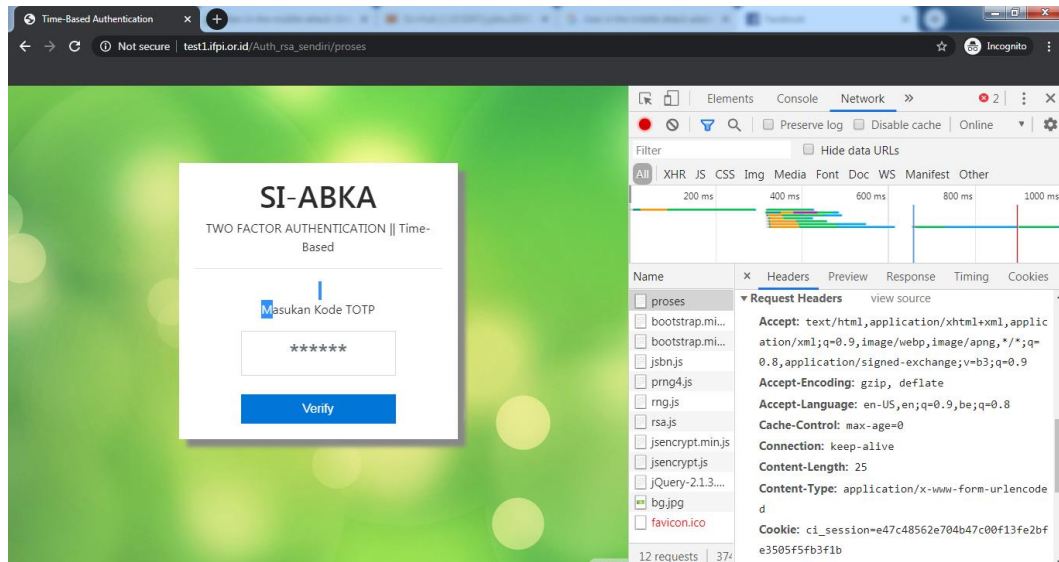
Aplikasi burpsuite bekerja dengan cara mengirimkan file header yang berisi data akun berupa KTA, password, dan kode OTP. Aplikasi burpsuite akan membantu pengujian dengan mengacak kode OTP dan mengirimkannya ke sistem. Pengujian kode OTP sukses jika mendapatkan respons code 303 dan gagal jika mendapatkan 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 spesifikasi alat sebagai berikut:

- a. Prosesor : Intel Core i3 2330M processor (3MB L3 cache, 2.20 GHz)
- b. Sistem operasi : Windows 7 Ultimate
- c. Memori : 4 GB DDR3 SDRAM 1066 MHz
- d. 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 3.7. 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
- 13.


```
14. totp=123123&username=admin&password=admin
```

3.6.2 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 pengguna 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:

a. Netdiscover

Netdiscover merupakan tool ip hunter yang berfungsi untuk membaca dan mencari 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.

b. Ettercap

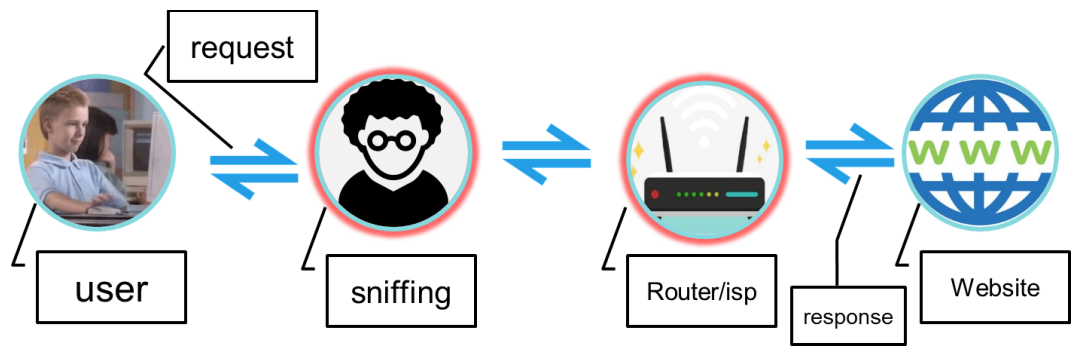
Ettercap berfungsi untuk membaca data yang dikirim oleh router ke ip target. Cara kerjanya adalah mengelabui router dengan menyamar sebagai ip target. Data yang seharusnya dikirim ke target akan terkirim juga ke penyerang. Data tersebut dapat dibaca oleh penyerang dan digunakan layaknya pemilik data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

c. 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 enkripsi data. Oleh karena itu sslstrip berfungsi untuk mengubah 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 3.8. 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.

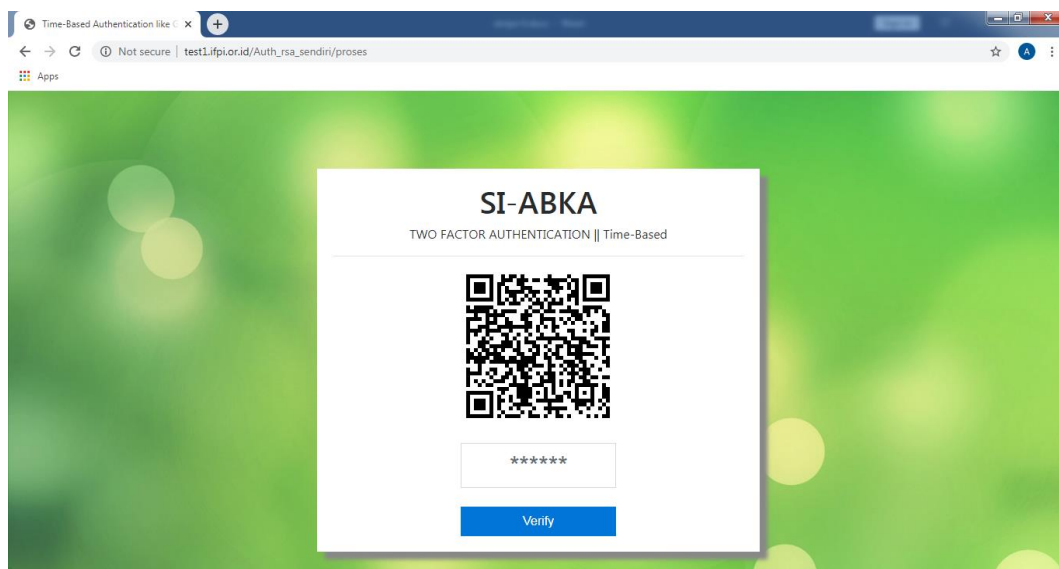
BAB 4. HASIL DAN PEMBAHASAN

4.1 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 tersebut adalah berupa file library PHP dan javascript.

4.1.1 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 pembangkitannya.

id_akun	kta	secret	privatekey	publickey
1	admi.n	X44EZY7VAIRUTITQ3NYEBNB2L7VPJEMC	1867277.1694891	1867277.11
4	1332.11.0001	TFJD24OSEPSD72YYXK5KVUVHQYRD5JIM	193241.76781	193241.5
5	1332.11.0002	SVN5D2JA2LYZSRG2IJ4UYNV2HFFPQ3EN	2546989.1695827	2546989.3
6	1332.11.0003	0	0	0
7	1332.11.0004	0	0	0
8	1332.11.0005	0	0	0

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, $waktuseka
rang = null) {
2.     // waktudurasi = jika 1 maka 30 detik jika 2 makan 60 detik dst.
3.     // waktusekarang = waktu yang diambil di server
4.     if ($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. }
```

4.1.2 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 sedangkan variabel e dan n digunakan sebagai public key. berikut contoh hasil pembangkitan public key dan private key yang telah di simpan di dalam database.

id_akun	kta	secret	privatekey	publickey
1	admi.n	X44EZY7VAIRUTITQ3NYEBNB2L7VPJEMC	1867277.1694891	1867277.11
4	1332.11.0001	TFJD24OSEPSD72YYXK5KVUVHQYRD5JIM	193241.76781	193241.5
5	1332.11.0002	SVN5D2JA2LYZSRG2IJ4UYNV2HFFPQ3EN	2546989.1695827	2546989.3
6	1332.11.0003	0	0	0
7	1332.11.0004	0	0	0
8	1332.11.0005	0	0	0

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 sebarakan 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 kunci RSA. Variabel n dan e akan digunakan untuk public key yang berfungsi untuk melakukan enkripsi. Variabel public key akan dikirimkan ke user untuk proses enkripsi di browser 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 dienkrpsi (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. }

```

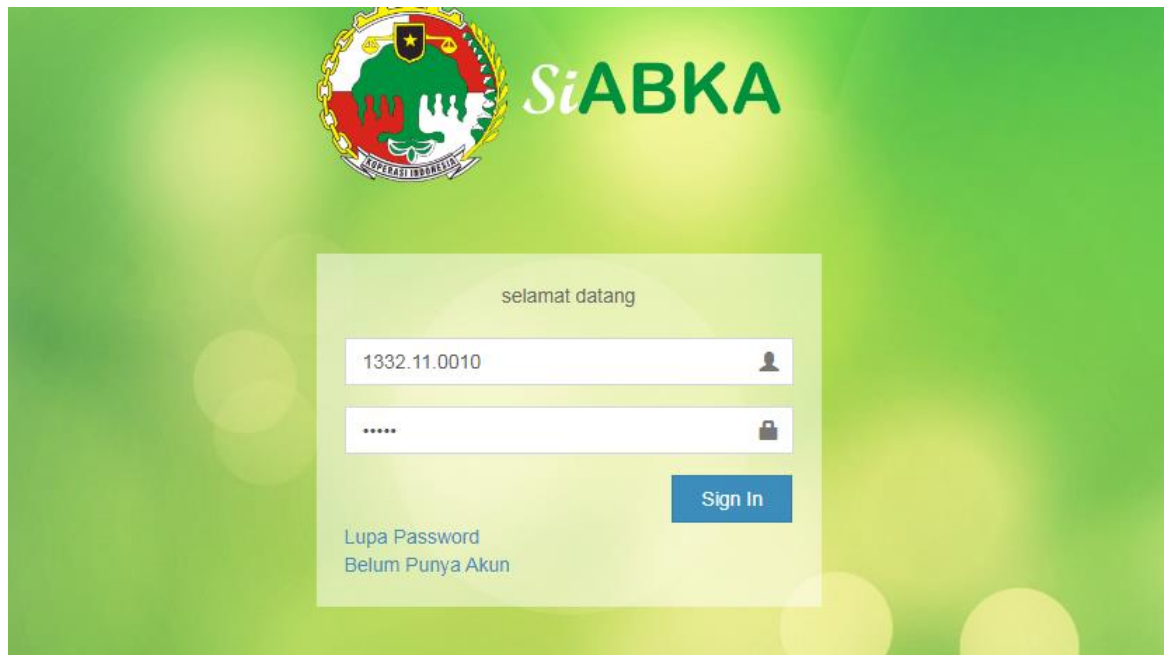
Proses dekripsi hampir sama saat melakukan enkripsi yaitu memecah data private key menjadi variabel n dan d. Karena nilai pangkat pada rsa memiliki nilai yang tinggi sampai ribuan maka perhitungan tidak bisa menggunakan rumus biasa. Nilai pangkat pada RSA mencapai ribuan sehingga membutuhkan bantuan 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.

4.2 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 lainnya. 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.

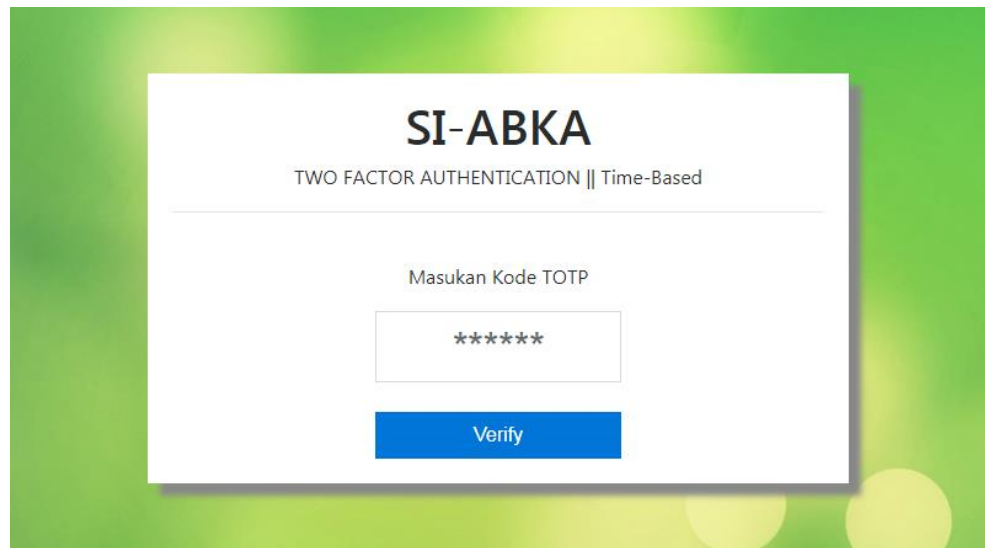


Gambar 4.4. tampilan login SI-Abka

2. 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.

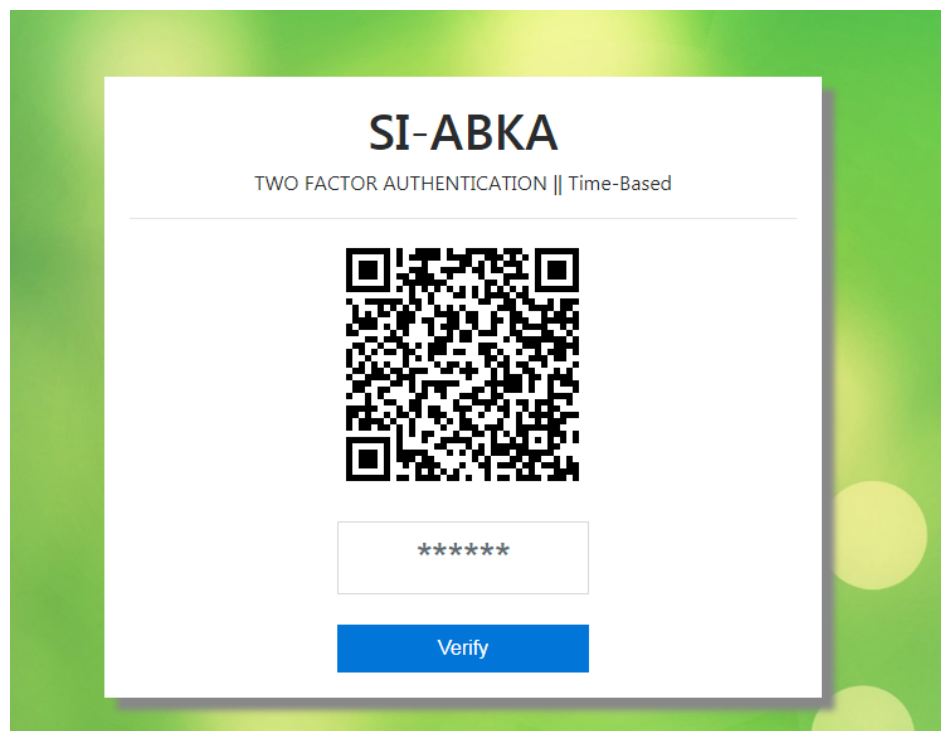
- a. User memiliki secret key

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.



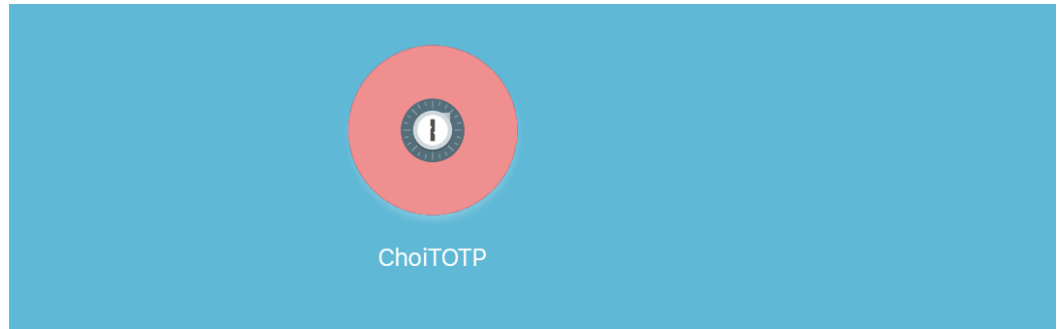
Gambar 4.5. input kode TOTP

- b. User belum memiliki secret key
 - i. 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

- ii. kode OTP didapatkan dari aplikasi TOTP yang sebelumnya melakukan scan qr-code terlebih dahulu seperti tampilan Gambar 4.7 di bawah ini



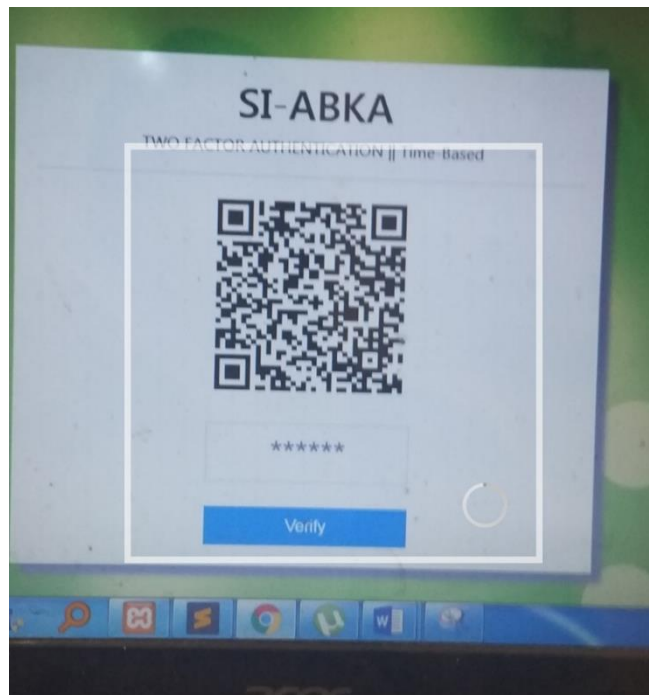
Gambar 4.7. aplikasi ChoiTOTP

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



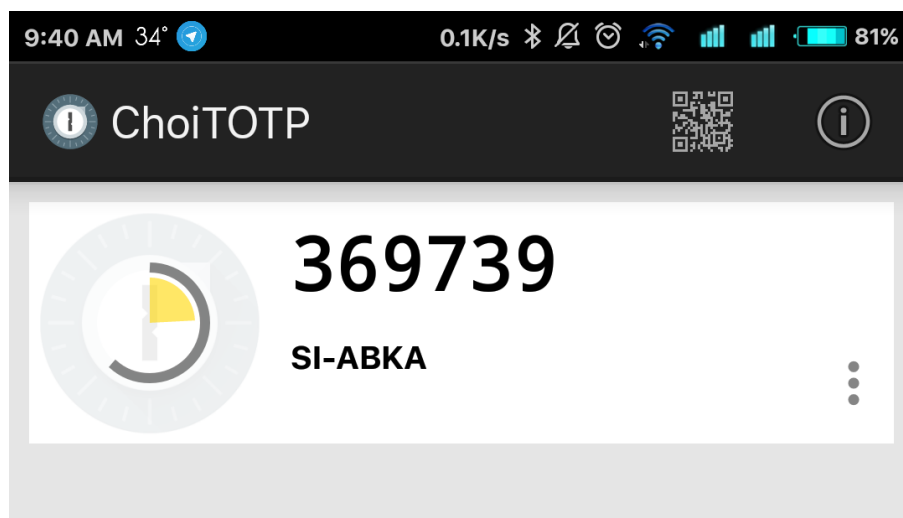
Gambar 4.8. menu aplikasi ChoiTOTP

- iv. Aplikasi akan mengaktifkan kamera belakang dan user perlu mengarahkan kamera ke gambar qr-code seperti pada tampilan seperti Gambar 4.9



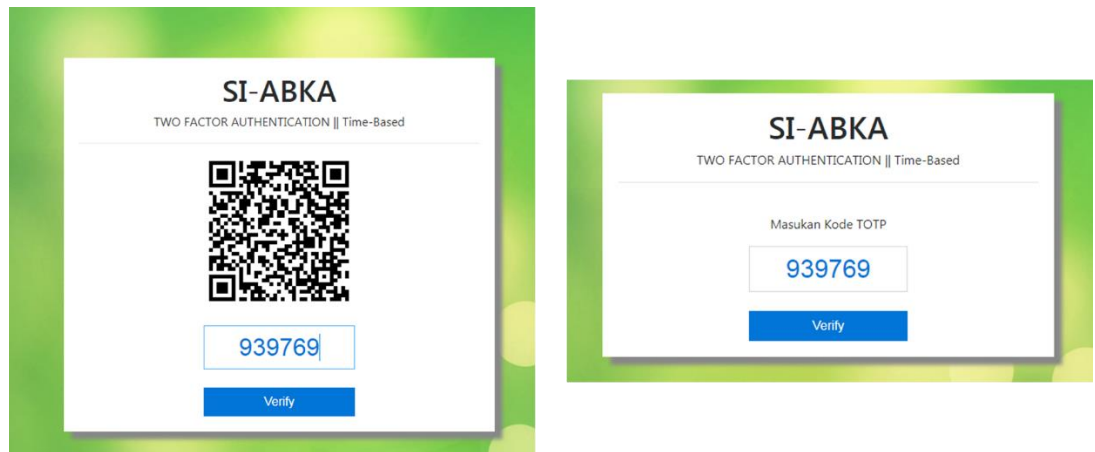
Gambar 4.9. tampilan scan qr-code

3. Setelah berhasil di pindai maka aplikasi akan menampilkan enam digit angka yang dapat digunakan untuk login seterusnya. Saat dibutuhkan pengguna tidak perlu 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

4. Enam digit angka tersebut dapat digunakan 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.

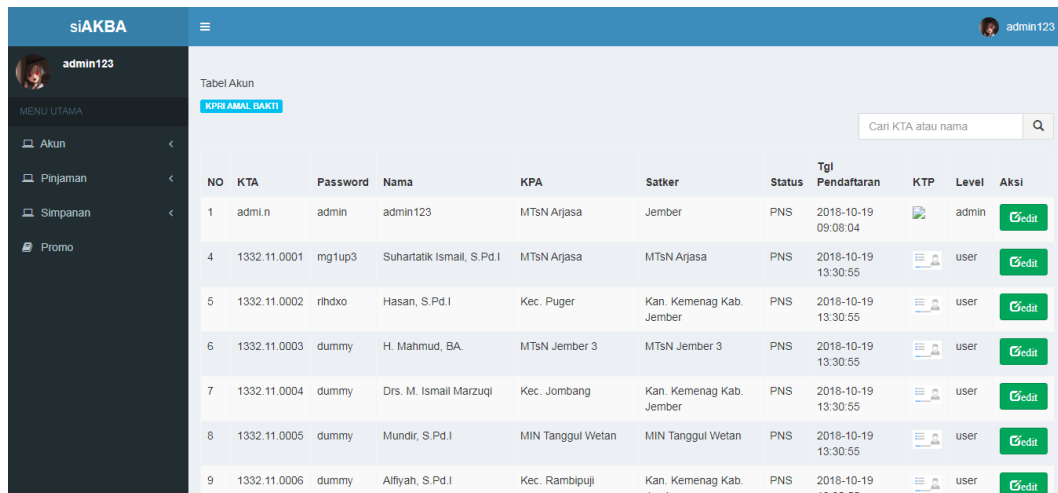


Gambar 4.11. memasukkan kode TOTP

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



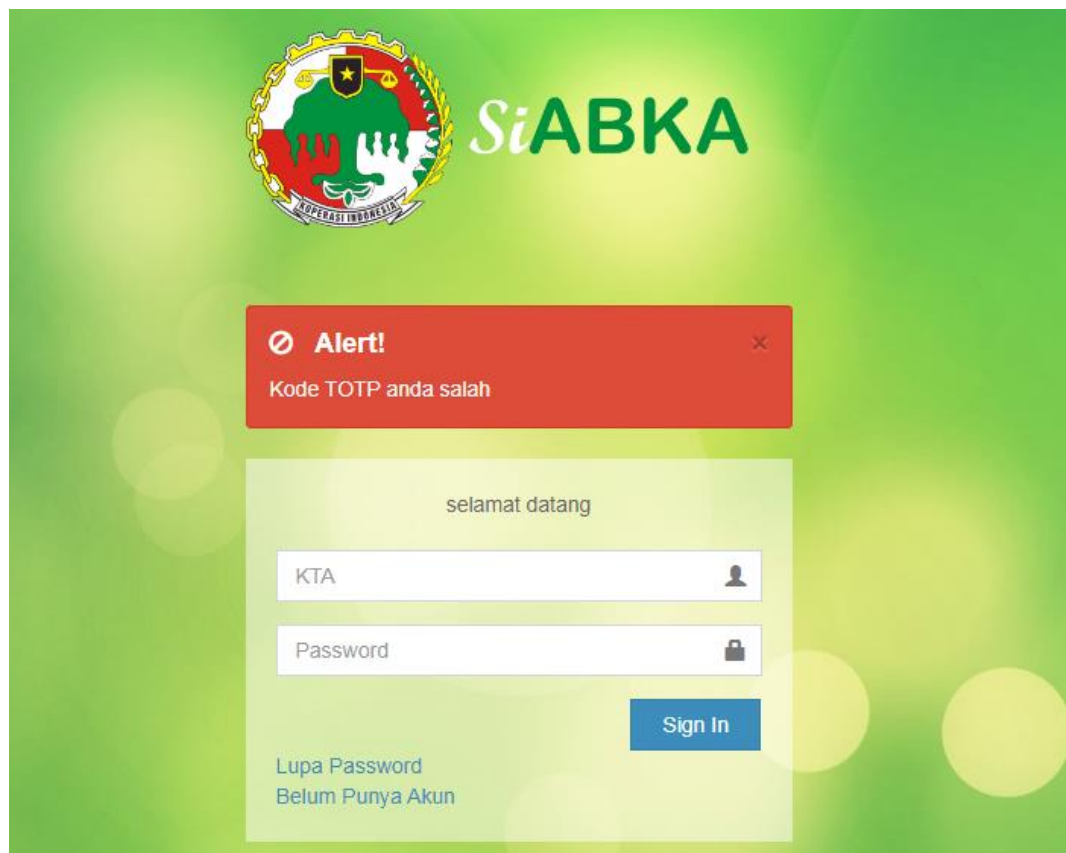
Gambar 4.12. dashboard user



NO	KTA	Password	Nama	KPA	Satker	Status	Tgl Pendaftaran	KTP	Level	Aksi
1	admin	admin	admin123	MTsN Arjasa	Jember	PNS	2018-10-19 09:08:04		admin	Edit
4	1332.11.0001	mg1up3	Suhartatik Ismail, S.Pd.I	MTsN Arjasa	MTsN Arjasa	PNS	2018-10-19 13:30:55		user	Edit
5	1332.11.0002	rhdxo	Hasan, S.Pd.I	Kec. Puger	Kan. Kemenag Kab. Jember	PNS	2018-10-19 13:30:55		user	Edit
6	1332.11.0003	dummy	H. Mahmud, BA.	MTsN Jember 3	MTsN Jember 3	PNS	2018-10-19 13:30:55		user	Edit
7	1332.11.0004	dummy	Drs. M. Ismail Marzuqi	Kec. Jombang	Kan. Kemenag Kab. Jember	PNS	2018-10-19 13:30:55		user	Edit
8	1332.11.0005	dummy	Mundir, S.Pd.I	MIN Tanggul Wetan	MIN Tanggul Wetan	PNS	2018-10-19 13:30:55		user	Edit
9	1332.11.0006	dummy	Alfiyah, S.Pd.I	Kec. Rambipuji	Kan. Kemenag Kab. Jember	PNS	2018-10-19 13:30:55		user	Edit

Gambar 4.13. dashboard admin

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



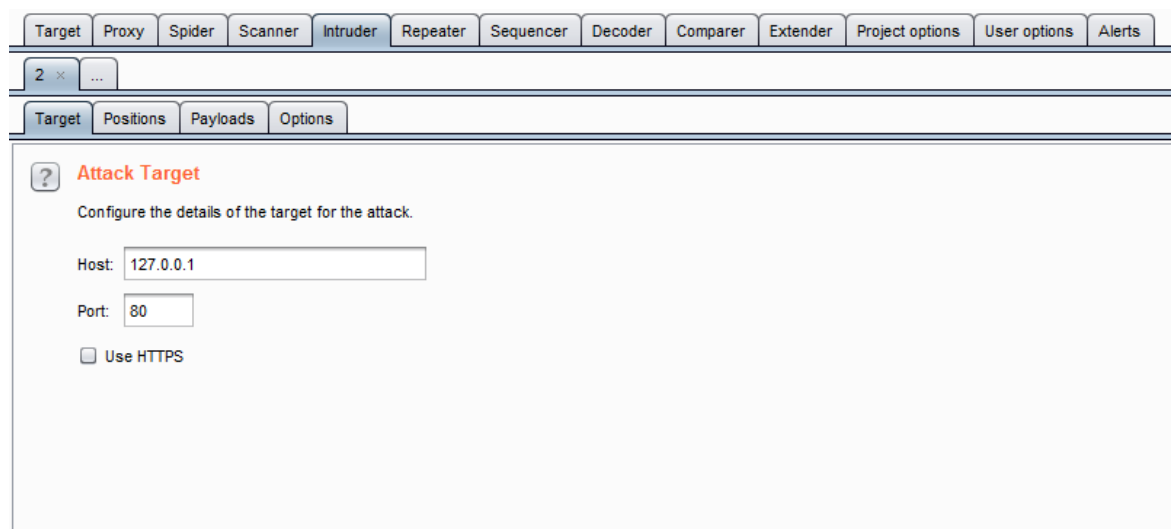
Gambar 4.14. pemberitahuan kode OTP salah

4.3 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 algoritma RSA, dengan SI-abka tanpa tambahan modul tersebut.

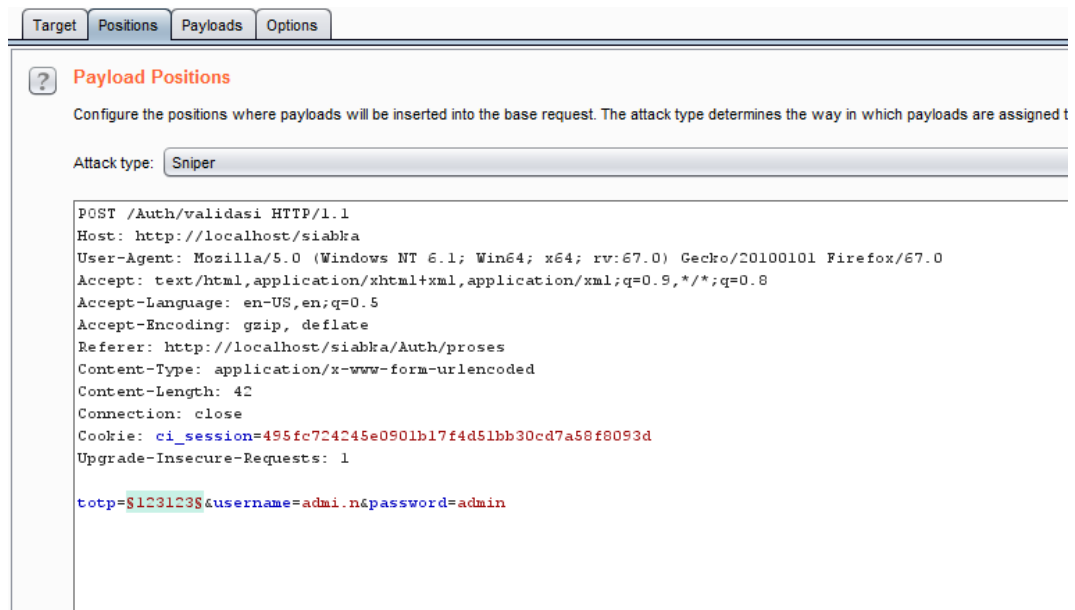
4.3.1 Pengujian brute force

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. Konfigurasi dapat dilihat pada gambar 4.15.



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 burpsuite 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. Kode TOTP diberi tanda \$ pada sisi depan dan belakang bertujuan sebagai penanda aplikasi burpsuite. Tanda tersebut menunjukkan letak variabel apa yang akan digenerate kode bruteforce.



Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to

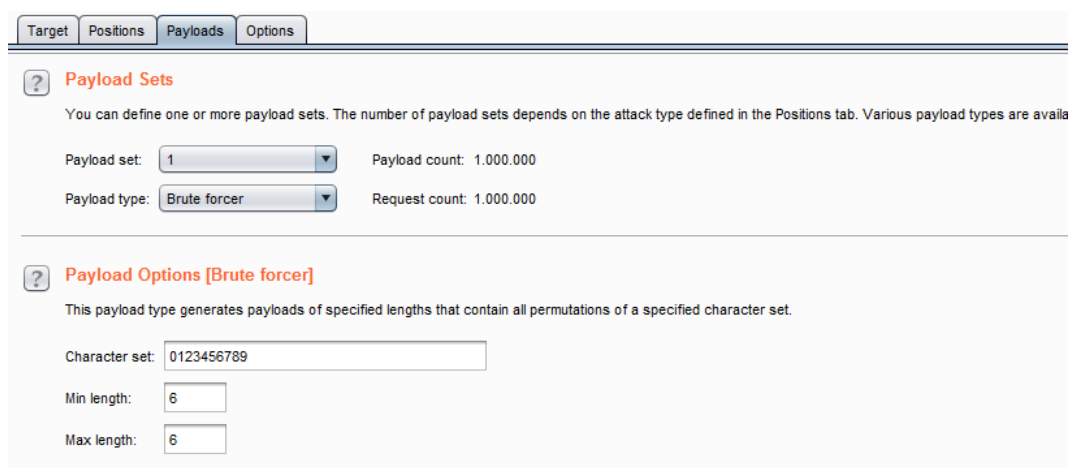
Attack type:

```
POST /Auth/validasi HTTP/1.1
Host: http://localhost/siabka
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:67.0) Gecko/20100101 Firefox/67.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/siabka/Auth/proses
Content-Type: application/x-www-form-urlencoded
Content-Length: 42
Connection: close
Cookie: ci_session=495fc724c45e0901b17f4d51bb30cd7a58f8093d
Upgrade-Insecure-Requests: 1

totp=$123123&username=admin&password=admin
```

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 akan dilakukan, pada penelitian ini menggunakan *brute force*. Sedangkan untuk *payloads options* merupakan pengaturan penyerangan yang digunakan untuk jenis penyerangan yang telah dipilih pada *payloads sets*.



Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available

Payload set: Payload count: 1.000.000

Payload type: Request count: 1.000.000

Payload Options [Brute forcer]

This payload type generates payloads of specified lengths that contain all permutations of a specified character set.

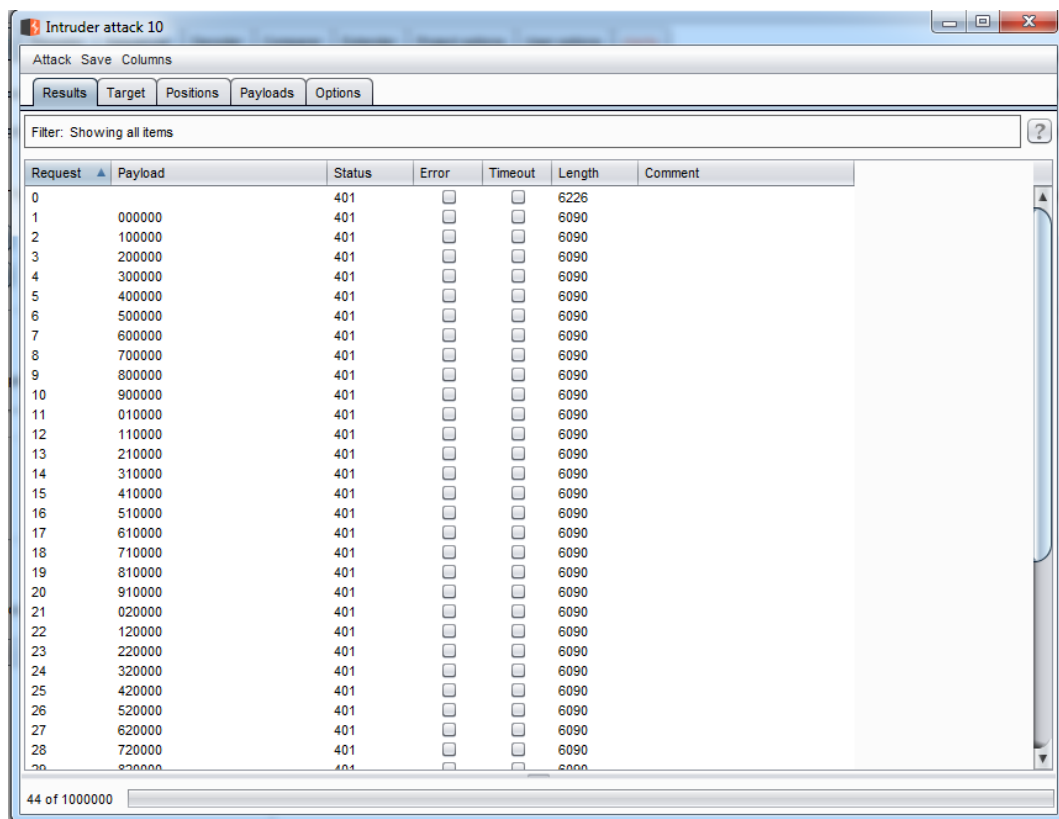
Character set:

Min length:

Max length:

Gambar 4.17. variabel payload brute force

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

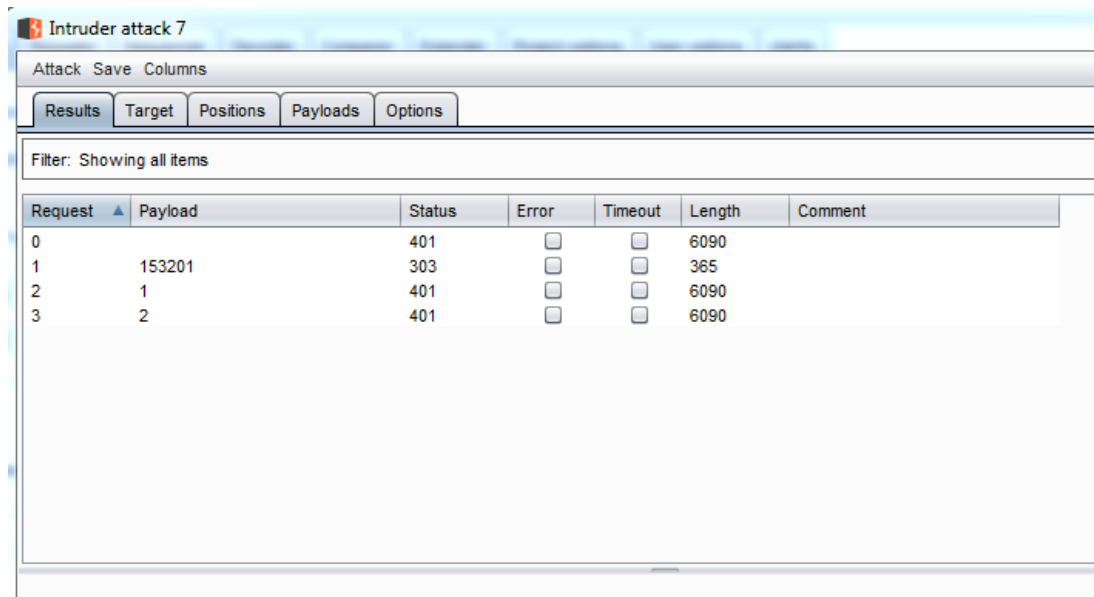


Request	Payload	Status	Error	Timeout	Length	Comment
0		401	<input type="checkbox"/>	<input type="checkbox"/>	6226	
1	000000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
2	100000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
3	200000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
4	300000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
5	400000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
6	500000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
7	600000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
8	700000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
9	800000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
10	900000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
11	010000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
12	110000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
13	210000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
14	310000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
15	410000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
16	510000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
17	610000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
18	710000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
19	810000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
20	910000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
21	020000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
22	120000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
23	220000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
24	320000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
25	420000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
26	520000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
27	620000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
28	720000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
29	820000	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	

44 of 1000000

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 maka akan mendapatkan respons code 303 (redirect) seperti gambar 4.19 dibawah ini



The screenshot shows the 'Intruder attack 7' window in Burp Suite. The 'Results' tab is selected, displaying a table of attack results. The table has columns for Request, Payload, Status, Error, Timeout, Length, and Comment. The filter is set to 'Showing all items'.

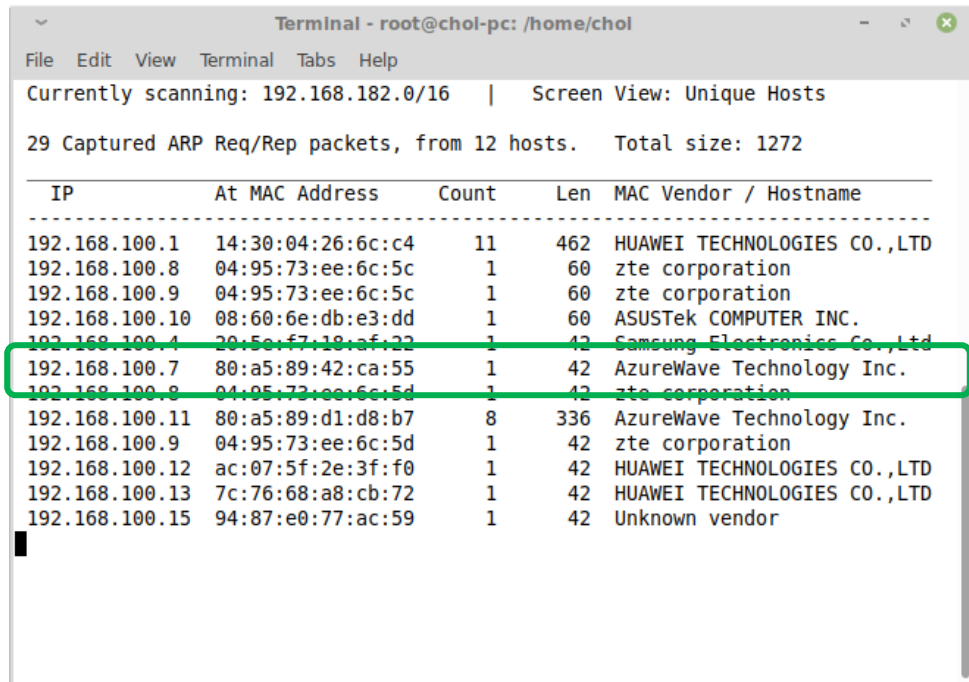
Request	Payload	Status	Error	Timeout	Length	Comment
0		401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
1	153201	303	<input type="checkbox"/>	<input type="checkbox"/>	365	
2	1	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	
3	2	401	<input type="checkbox"/>	<input type="checkbox"/>	6090	

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.

4.3.2 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 .



IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.100.1	14:30:04:26:6c:c4	11	462	HUAWEI TECHNOLOGIES CO.,LTD
192.168.100.8	04:95:73:ee:6c:5c	1	60	zte corporation
192.168.100.9	04:95:73:ee:6c:5c	1	60	zte corporation
192.168.100.10	08:60:6e:db:e3:dd	1	60	ASUSTek COMPUTER INC.
192.168.100.4	20:5c:f7:18:af:22	1	42	Samsung Electronics Co.,Ltd
192.168.100.7	80:a5:89:42:ca:55	1	42	AzureWave Technology Inc.
192.168.100.8	04:95:73:ee:6c:5d	1	42	zte corporation
192.168.100.11	80:a5:89:d1:d8:b7	8	336	AzureWave Technology Inc.
192.168.100.9	04:95:73:ee:6c:5d	1	42	zte corporation
192.168.100.12	ac:07:5f:2e:3f:f0	1	42	HUAWEI TECHNOLOGIES CO.,LTD
192.168.100.13	7c:76:68:a8:cb:72	1	42	HUAWEI TECHNOLOGIES CO.,LTD
192.168.100.15	94:87:e0:77:ac:59	1	42	Unknown vendor

Gambar 4.20. Proses Scan Ip Target

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

```

Terminal - root@chol-pc: /home/choi
File Edit View Terminal Tabs Help

root@choi-pc:/home/choi# ettercap -S -Tq -M arp:remote -i enp3s0f0 /192.168.100.1// /192.168.100.7//

ettercap 0.8.2 copyright 2001-2015 Ettercap Development Team

Listening on:
enp3s0f0 -> 20:6A:8A:79:37:18
          192.168.100.16/255.255.0
          fe80::d1fb:da16:84bd:5932/64

Ettercap might not work correctly. /proc/sys/net/ipv6/conf/all/use_tempaddr is not set to 0.
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/enp3s0f0/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EUID 65534...

  33 plugins
  42 protocol dissectors
  57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |=====| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

Sniffing 192.168.100.7 on interface enp3s0f0
Starting Unified sniffing...

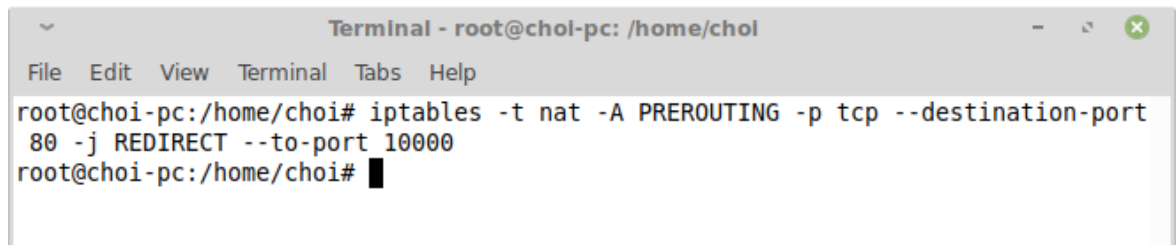
Text only Interface activated...
Hit 'h' for inline help

```

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



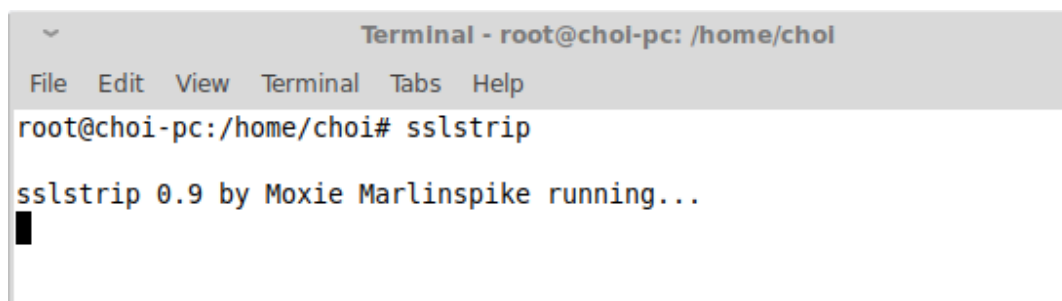
```

Terminal - root@choi-pc: /home/choi
File Edit View Terminal Tabs Help
root@choi-pc:/home/choi# iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 10000
root@choi-pc:/home/choi#

```

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.



```

Terminal - root@choi-pc: /home/choi
File Edit View Terminal Tabs Help
root@choi-pc:/home/choi# sslstrip

sslstrip 0.9 by Moxie Marlinspike running...

```

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.

```

Terminal - root@chol-pc: /home/chol
File Edit View Terminal Tabs Help
* |=====| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 192.168.100.1 14:30:04:26:6C:C4

GROUP 2 : 192.168.100.7 80:A5:89:42:CA:55
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

HTTP : 83.136.216.55:80 -> USER:  PASS: admin  INFO: http://test1.ifpi.or.id/Auth
CONTENT: KTA=admi.n&password=admin

HTTP : 83.136.216.55:80 -> USER:  PASS: admin  INFO: http://test1.ifpi.or.id/Auth_rsa_sendiri/proses
CONTENT: password=admin&kta=admi.n&encrypted=16807.59049.0.243.3125.32768

HTTP : 83.136.216.55:80 -> USER:  PASS: admin  INFO: http://test1.ifpi.or.id/Auth
CONTENT: KTA=admi.n&password=admin

HTTP : 83.136.216.55:80 -> USER:  PASS: admin  INFO: http://test1.ifpi.or.id/Auth_rsa_sendiri/proses
CONTENT: password=admin&kta=admi.n&encrypted=32.59049.59049.7776.0.16807

HTTP : 83.136.216.55:80 -> USER:  PASS: dummy  INFO: http://test1.ifpi.or.id/Auth
CONTENT: KTA=1332.11.0006&password=dummy

HTTP : 83.136.216.55:80 -> USER:  PASS: dummy  INFO: http://test1.ifpi.or.id/Auth_rsa_sendiri/proses
CONTENT: password=dummy&kta=1332.11.0006&encrypted=1960808.279936.2097152.128.2097152.279936

HTTP : 83.136.216.55:80 -> USER:  PASS: dummy  INFO: http://test1.ifpi.or.id/Auth
CONTENT: KTA=1332.11.0006&password=dummy

HTTP : 83.136.216.55:80 -> USER:  PASS: dummy  INFO: http://test1.ifpi.or.id/Auth_rsa_sendiri/proses
CONTENT: password=dummy&kta=1332.11.0006&encrypted=279936.128.2097152.78125.78125.279936

```

Gambar 4.24. hasil pembacaan MITM

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.

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/auth_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	1960808.279936.2097152.128.2 907512.279936	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	279936.128.2097152.78125.781 25.279936	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.

4.4 Pembahasan

Hasil dari penelitian ini adalah library metode two factor authentication yang menggunakan TOTP dan algoritma RSA sebagai pengaman dalam mengirimkan data. Library ini dapat diimplementasikan pada semua website berbasis PHP 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 pengiriman 44 kemungkinan header dari satu juta kemungkinan yang ada setiap 30 detik. Presentase keberhasilan tertebakanya kode adalah $44/1.000.000$ sama dengan 0.044 %. Presentase tersebut sangat kecil kemungkinan tertebakanya 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 mengamankan data yang dikirim 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 dihasilkan akan berbeda tiap pengguna dikarenakan secret key dan public key tiap akun berbeda-beda.

BAB 5. PENUTUP

5.1 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 penelitian kali ini peneliti melakukan pengukuran tingkat keamanan dari penggunaan two factor authentication . 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 tertebakny 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

5.2 Saran

Pada sub bab ini akan dijabarkan beberapa saran penelitian 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. Sedyono, dan S. Suhartono. 2013. Studi pengamanan login pada sistem informasi akademik menggunakan otentifikasi one time password berbasis sms dengan hash md5. *JURNAL SISTEM INFORMASI BISNIS*. 3(1):07–12.
- 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 kraitichik. *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 Response 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. 2017. IMPLEMENTASI algoritma time-based one time password dalam otentikasi token internet banking. 10.