IMPLEMENTASI ALGORITME KRIPTOGRAFI BLOWFISH PADA SISTEM KEAMANAN UJIAN *ONLINE* BERBASIS *PLATFORM* ANDROID

Muhammad Hafizh[1)], Aryo Pinandito, S.T., M.MT. [2)], Agi Putra Kharisma, S.T., M.T.[3)]
[1)]Mahasiswa,[2)3)]Dosen Pembimbing
Program Studi Informatika, Fakultas Ilmu Komputer
Universitas Brawijaya
Jl.Veteran, Malang 65145, Indonesia
hafizhemail@gmail.com

ABSTRAK

Ujian merupakan salah satu wujud evaluasi dari proses belajar. Sejauh mana pemahaman siswa terhadap bidang studi yang ditempuh diukur melalui hasil yang diperoleh setelah melaksanakan ujian. Namun permasalahan keamanan menjadi tantangan di dalam sebuah sistem ujian *online*. Proses pengiriman dan penerimaan informasi ujian masih sangat rentan terhadap upaya pencurian, penyadapan, pembajakan, dan hal lain yang menyebabkan kebocoran informasi ujian. Dua aspek yang menjadi perhatian dalam pelaksanaan ujian *online* ini adalah *confidentiality* dan *integrity*. *Confidentiality* yang diartikan sebagai privasi atau kerahasiaan merujuk pada perlindungan informasi dari penyingkapan pihak yang tidak sah. Sedangkan *integrity* merujuk pada fungsionalitas jaringan, sistem, atau aplikasi untuk melindungi informasi dari perubahan yang disengaja.

Salah satu metode yang dapat diterapkan untuk mengatasi masalah tersebut adalah metode kriptografi dengan menerapkan enkripsi untuk melindungi informasi terkait ujian *online* yang sedang berlangsung. Penerapan kriptografi memanfaatkan algoritme Blowfish sebagai salah satu algoritme sandi simetris yang handal dibanding algoritme lainnya dapat meningkatkan sekuritas dari sisi informasi ujian terhadap sistem keamanan ujian *online* yang dibangun.

Berdasarkan pengujian *confidentiality*, informasi ujian *online* yang dikirimkan peserta dengan enkripsi algoritme Blowfish mampu merahasiakan isi di dalamnya saat dilakukan penyadapan melalui aplikasi *sniffing* jaringan Wireshark sehingga tidak dapat dimengerti isinya oleh penyadap. Sedangkan hasil dari pengujian *integrity* dengan adanya percobaan dekripsi terhadap informasi ujian yang berhasil disadap menggunakan metode *brute force*, tidak dapat ditemukan kunci yang sesuai untuk membuka informasi jawaban. Dari pengujian untuk melakukan *request* langsung dari URL menggunakan informasi hasil *sniffing*, sistem keamanan ujian *online* memberi penolakan akses dan tidak dapat dilakukan perbaruan data milik peserta sehingga keutuhan data milik peserta yang tersimpan terjaga.

Kata kunci: Ujian *Online*, Sistem Keamanan, *Blowfish*, *Confidentiality*, *Integrity*.

ABSTRACT

Exam is one form of evaluation of the learning process. The extent to which students' understanding of the field of study pursued measured by the results obtained after carrying out the test. However, security issues become a challenge in an online examination system. The process of sending and receiving test information is still very vulnerable to attempted theft, eavesdropping, hijacking, and other things that cause information leakage test. Two aspects are of concern in the implementation of online exams are confidentiality and integrity. Confidentiality is interpreted as privacy or confidentiality refers to the protection of information from unauthorized disclosure parties. While integrity refers to the functionality of networks, systems, or applications to protect information from accidental changes.

One method that can be applied to solve the problem is a cryptographic method by applying encryption to protect information related to the online test is in progress. Implementation of cryptographic algorithms Blowfish utilizes as one of the symmetrical password algorithm reliable than other algorithms can improve the security of the exam information to the online test security systems are built.

Based on confidentiality testing, information submitted online exam participants with Blowfish encryption algorithm is able to keep the contents therein when done tapping through the Wireshark network sniffing applications so its contents can not be understood by eavesdroppers. While the results of integrity testing with their decryption trial against exam information intercepted using a brute force method, can not find the appropriate key to open the answer information. From the testing to make the request directly from a URL using sniffing result information, the security system online examination gave the denial of access and can not do update the data belonging to the participants so that the integrity of the stored data of the participants is maintained.

Keyword: Online Examination, Security System, Blowfish, Confidentiality, Integrity.

1. PENDAHULUAN

1.1 Latar Belakang

Ujian merupakan salah satu wujud evaluasi dari proses belajar. Sejauh mana pemahaman siswa terhadap bidang studi yang ditempuh diukur melalui hasil yang diperoleh setelah melaksanakan ujian. Perkembangan model pelaksanaan ujian pun kini menyesuaikan perkembangan teknologi dimana saat ini sudah umum diadakan ujian secara *online* yang menggeser pelaksanaan ujian konvensional yang mengharuskan penguji melakukan evaluasi secara manual pada lembar jawaban. Sistem ujian secara *online* juga menjadi kebutuhan penting dalam kegiatan pembelajaran jarak jauh karena mengurangi biaya jika dibandingkan dengan ujian secara tatap muka. Hal ini juga terkait penghematan *resources* kertas dan alat tulis serta penghematan waktu yang memakan banyak biaya (Pratondo, 2010).

Seiring dengan banyaknya pengguna *mobile device* di seluruh dunia yang mencapai lebih dari 2,23 miliar (E-marketer, 2014). Penelitian ini memanfaatkan *mobile device* sebagai *client* dari sistem ujian *online*. Penggunaan *mobile device* sebagai alternatif *client* dari *personal computer* dapat meningkatkan fleksibilitas karena *mobile device* dapat dengan mudah dibawa kemana saja dibandingkan dengan *personal computer*.

Manajemen kerahasiaan dan keamanan masih menjadi tantangan dalam sebuah sistem ujian *online*. Keamanan dalam ujian *online* meliputi perlindungan dan reliabilitas pengiriman hasil ujian yang menjadi sebuah informasi yang dikirimkan ke dalam database ujian. Dua aspek utama terkait keamanan data dan informasi menjadi perhatian dalam pelaksanaan ujian *online* ini adalah *confidentiality* dan *integrity* (Zughoul, O. et al. 2013).

Confidentiality (kerahasiaan), yang juga dapat diartikan sebagai privasi atau kerahasiaan merujuk pada perlindungan informasi dari penyingkapan pihak yang tidak sah. Dapat diperoleh dengan memberi akses terbatas pada informasi atau dengan penyandian informasi sehingga tidak memiliki arti apapun bagi pihak yang tidak berhak tersebut. Jika kerahasiaan ini tidak terpenuhi mengakibatkan adanya penyalahgunaan wewenang dalam pelaksanaan ujian oleh pihak yang tidak sah. Aspek kedua adalah *integrity*, dapat diartikan sebagai akurasi. Merujuk pada perlindungan informasi, data, atau transmisi yang tidak sah, tidak terkendali, atau perubahan yang disengaja. Istilah integritas juga dapat merujuk pada fungsionalitas jaringan, sistem, maupun aplikasi. Jika aspek integritas tidak terpenuhi dapat mengakibatkan terjadinya data yang tidak sah atau mengurangi keabsahan terhadap hasil jawaban peserta ujian (Canavan, 2001) yang dapat mengakibatkan ketidaklulusan peserta terhadap ujian yang diikutinya. Salah satu solusi untuk menghindari resiko penyerangan terhadap informasi hasil nilai ujian adalah dengan melakukan tindakan enkripsi pada nilai di dalam *database*, dengan menghasilkan sebuah *ciphertext* dari *plaintext* nilai asli. Metode keamanan ini disebut dengan teknik kriptografi.

Kriptografi adalah ilmu atau seni dalam memperoleh keamanan dengan pengkodean pesan agar tidak dapat dibaca dengan cara menyembunyikan informasi pesan. Di zaman modern kriptografi dianggap sebagai cabang ilmu dari matematika dan ilmu komputer dan berkaitan erat dengan teori informasi dan keamanan komputer (Ayushi, 2010). Adanya kriptografi memungkinkan pengguna untuk menyimpan informasi penting maupun mengirimkannya melalui jaringan yang tidak aman seperti internet sehingga tidak dapat dibaca oleh siapapun selain penerima yang dimaksud.

Blowfish merupakan salah satu algoritme kriptografi yang tidak dipatenkan dan cukup kuat karena dapat berjalan pada memori kurang dari 5KB serta keamanannya yang bervariasi dengan panjang kunci mencapai 448 *bit*, yang artinya ada sebanyak 2^{448} kemungkinan kunci untuk membongkarnya sehingga pesan yang ada dalam *ciphertext* sangat aman (Ibrahim, 2012). Sampai saat ini belum ada metode kriptanalisis yang lebih efisien yang dapat digunakan untuk menyerang 16 putaran penuh algoritme Blowfish selain serangan *brute-force*. Implementasi Blowfish tidak menggunakan *resources* memori yang banyak, hal ini menjadikan algoritme tersebut banyak digunakan pada *embedded system*. Keunggulan tersebut menjadikan algoritme Blowfish sebagai pilihan terbaik untuk melakukan proses enkripsi karena ringan, tidak terhalang lisensi, dan dianggap aman bahkan setelah dilakukan analisa yang ekstensif (Gatliff, 2003).

Berdasarkan alasan yang telah dipaparkan di atas, pada penelitian ini dibuat sistem keamanan aplikasi dari informasi ujian *online* memanfaatkan algoritme Blowfish yang diterapkan pada platform Android untuk membantu menjaga kerahasiaan dan keutuhan informasi jawaban peserta ujian.

1.2 Rumusan Masalah

Berdasarkan pada pemasalahan yang telah dijelaskan pada bagian latar belakang, rumusan masalah dari penelitian ini yaitu:

1. Apakah kerahasiaan informasi ujian peserta dapat dijaga dengan metode enkripsi algoritme Blowfish?
2. Apakah integritas jawaban peserta ujian dapat dijaga dengan perlindungan enkripsi algoritme Blowfish saat dilakukan dekripsi dan intrusi pengiriman informasi dari pihak luar?

2. KAJIAN PUSTAKA & DASAR TEORI

2.1 Kajian Pustaka

Kajian pustaka pada bab ini membahas tentang pemilihan dan perbandingan algoritme kriptografi Blowfish terhadap algoritme keamanan lainnya berdasarkan jurnal penelitian terdahulu.

Penelitian sebelumnya berjudul *"Energy Efficiency of Encryption Schemes for Wireless Devices"* (Elminaam, 2009) mempertimbangkan penggunaan *resources* daya baterai untuk diperhitungkan, mengingat terbatasnya daya baterai yang tidak sepadan dengan penggunaan aplikasi pada perangkat nirkabel. Daya baterai juga dapat bergantung pada masalah energi yang dikonsumsi dikarenakan algoritme kriptografi yang digunakan. Perkembangan teknologi baterai lebih lambat dibandingkan teknologi lain. Hal ini menyebabkan "*battery gap*" sehingga dibutuhkan langkah untuk membuat keputusan mengenai konsumsi energi dan keamanan untuk mengurangi perangkat bertenaga baterai. Sehingga pada tahun 2009, Elminaam dan tim membuat penelitian tentang perbandingan kinerja algoritme simetris untuk proses enkripsi menggunakan algoritme *AES, DES, 3DES, RC2, Blowfish,* dan *RC6* untuk proses enkripsi *file text, file audio,* dan *file video*. Hasil dari penelitian ini adalah dalam hal perubahan ukuran paket, dapat disimpulkan bahwa algoritme *Blowfish* memiliki kinerja terbaik dalam penghematan daya baterai.

Penelitian kedua yang dijadikan referensi berjudul "*Superiority of Blowfish Algorithm*" (Mandal, 2012) yang membahas mengenai percobaan kalkulasi kecepatan waktu enkripsi dan dekripsi dari beberapa algoritme simetris yang paling umum digunakan berdasarkan ukuran paket yang berbeda-beda, juga penggunaan daya terminim yang dikonsumsi. Dalam hal ini parameter yang diperhitungkan adalah waktu enkripsi dan dekripsi, juga waktu pemrosesan dari *CPU* dalam bentuk *throughput* (kecepatan transfer data efektif). Hasil yang diperoleh menunjukkan superioritas algoritme Blowfish dalam masalah *throughput*, waktu pemrosesan, juga penggunaan daya yang dikonsumsi. *Throughput* lebih besar, kecepatan algoritme lebih tinggi, dan daya yang dikonsumsi lebih minim. Kedua, *AES* memiliki keunggulan terhadap *DES* maupun *3DES* dalam masalah *throughput* dan waktu dekripsi. Ketiga, bahwa *3DES* mempunyai performa yang lebih buruk dibanding seluruh algoritme yang disebutkan. Sehingga dapat disimpulkan bahwa *Blowfish* adalah yang terbaik diantara ketiganya.

2.2 Ujian *Online*

Ujian adalah salah satu sarana untuk mengevaluasi proses belajar. Dalam dunia pendidikan, ujian dimaksudkan untuk mengukur taraf pencapaian suatu tujuan pengajaran oleh siswa atau mahasiswa sebagai peserta didik, sehingga siswa atau mahasiswa dapat mengetahui tingkat kemampuannya dalam memahami bidang studi yang ditempuh. Bila ternyata hasilnya belum maksimal, maka proses harus ditingkatkan baik kualitas maupun kuantitas. (Clara: 2006).

Berdasarkan metode pengerjaannya, ujian dapat dibedakan atas ujian konvensional dan ujian *online*. Ujian konvensional adalah ujian yang pengerjaannya menggunakan kertas dan alat tulis dan mengharuskan peserta ujian mendatangi tempat tertentu untuk melaksanakan ujian. Sedangkan ujian *online* adalah ujian yang memanfaatkan keunggulan teknologi dalam pengerjaannya sehingga tidak mewajibkan peserta ujian untuk membawa alat tulis. Secara umum pelaksanaan ujian *online* masih menggunakan media *personal computer* (PC).

Ujian *online* memiliki berbagai macam jenis soal, antara lain *multiple choice* (pilihan ganda), *true or false answer* (jawaban benar/salah), jawaban singkat, esai, praktek, *open book*, *problem solving* (pemecahan masalah) dan oral (secara lisan). (Learnline, 2013). Bentuk ujian *multiple choice* (pilihan ganda) berarti jawaban dari pertanyaan yang diajukan harus dipilih berdasarkan beberapa kemungkinan jawaban yang tersedia. Peserta ujian diharuskan untuk memilih satu jawaban yang dianggap paling benar untuk menjawab pertanyaan. Dalam format ujian *online* ini, pengerjaan dilakukan dengan memilih pilihan jawaban yang benar dari layar perangkat bergerak peserta ujian.

2.3 Sistem Keamanan

Sistem merupakan sekumpulan elemen yang saling berkaitan dan bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*) (Tavri D. Mahyuzir, 1995). Landon (2002) mengatakan bahwa sistem terdiri dari unsur-unsur masukan, proses, keluaran penyimpanan serta umpan balik yang saling berkaitan membentuk suatu jalinan (*interface*) dalam satu batas lingkungan yang jelas (*boundary*).

Sistem keamanan adalah tindakan pencegahan terhadap segala bentuk penyebab kerugian, termasuk di dalamnya kerugian secara fisik dan non fisik, berwujud atau tidak berwujud, serta adanya bermacam-macam kerugian oleh berbagai sebab.

Garfinkel mengemukakan bahwa sistem keamanan melingkupi lima aspek, meliputi (Widiyanto, 2007):

1. *Privacy/Confidentiality*

Aspek *privacy* atau *confidentiality* adalah sebuah tindakan yang dilakukan untuk menjaga informasi dari orang yang tidak berhak mengakses informasi tersebut. *Privacy* lebih ke arah data-data yang bersifat

rahasia sedangkan *confidentiality* berhubungan dengan data yang diberikan kepada pihak lain dengan maksud dan tujuan tertentu.

2. *Integrity*

Aspek *integrity* atau integritas menekankan bahwa suatu informasi tidak boleh diubah tanpa adanya izin dari pemilik informasi. Jika terdapat perbedaan maka boleh dibilang aspek integritas tidak tercapai.

3. *Authentication*

Aspek ini berhubungan dengan metode untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud, atau server yang ditujukan adalah server yang asli.

4. *Availability*

Aspek *availability* berhubungan dengan ketersediaan sebuah data atau informasi. Data maupun informasi tersebut hanya dapat digunakan oleh yang berhak.

5. *Access Control*

Aspek access control berhubungan dengan cara pengaturan akses kepada informasi. Misalnya, seorang administrator memiliki hak akses penuh terhadap sebuah komputer, tetapi hal ini tidak berlaku bagi *account guest* ataupun *limited account* lainnya.

2.4 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani: "*cryptos*" dan "*graphein*". *Crypto* berarti rahasia (*secret*), sedangkan *graphein* berarti tulisan (*writing*). Sehingga, kriptografi berarti "*secret writing*" (tulisan rahasia). Secara terminologi, kriptografi merupakan ilmu dan seni yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentifikasi data. Sebuah algoritme kriptografi, disebut *cipher*, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat (Raharjo, 2005).

Kriptografi sendiri mempunyai komponen-komponen untuk mencapai tujuan kriptografi. Menurut (Ariyus, 2009), beberapa komponen dalam kriptografi meliputi:

1. Enkripsi (*Encryption*)

Enkripsi merupakan hal yang sangat penting dalam kriptografi untuk mengamankan sebuah informasi agar pesan yang dikirimkan terjaga kerahasiaannya. Informasi (yang disebut *plaintext*) diubah menjadi serangkaian kode rumit yang sulit diartikan. Enkripsi sendiri bisa diartikan sebagai *cipher* atau kode. Berdasarkan ISO 7498-2, terminologi yang lebih tepat digunakan untuk menamakan proses ini adalah "*encipher*".

2. Dekripsi (*Decryption*)

Dekripsi merupakan kebalikan dari proses enkripsi. Dekripsi yaitu proses mengubah kembali pesan yang telah dienkripsi menjadi pesan aslinya, yang disebut dengan dekripsi pesan. Berdasarkan ISO 7498-2, terminologi yang lebih tepat untuk menamakan proses ini adalah "*decipher*".

3. Kunci (*Key*)

Kunci yang dimaksud disini adalah kunci atau sandi yang digunakan untuk melakukan enkripsi maupun dekripsi. Kunci terbagi menjadi dua bagian, yaitu kunci privat (*private key*) dan kunci publik (*public key*).

4. *Plaintext*

Plaintext disebut juga *cleartext*, yaitu pesan asli yang ditulis atau diketik. *Plaintext* inilah yang akan diproses menggunakan algoritme kriptografi agar menjadi *ciphertext*.

5. Pesan

Pesan bisa berupa data atau informasi yang dikirimkan (melalui kurir, saluran komunikasi data, dan sebagainya) atau yang disimpan di dalam media penyimpanan (kertas, *storage*, dan sebagainya).

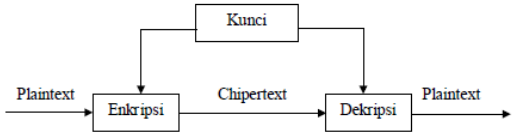
6. *Ciphertext*

Ciphertext merupakan pesan yang dihasilkan dari proses enkripsi. Pesan yang terkandung dalam *ciphertext* ini sulit dibaca karena berisi berbagai macam karakter tanpa arti/tidak bermakna.

7. Kriptanalisis (*Cryptanalysis*)

Dapat diartikan sebagai analisis sandi atau suatu ilmu memecahkan *ciphertext* menjadi *plaintext* tanpa mengetahui kunci yang digunakan. Pelakunya disebut *cryptanalys* (kriptanalis).

Kriptografi mempunyai dua komponen utama yaitu enkripsi dan dekripsi. Selain itu dibutuhkan kunci untuk mengubah *plaintext* menjadi *ciphertext*, begitu juga sebaliknya. Tanpa kunci, *plaintext* tidak bisa melakukan enkripsi pesan menjadi *ciphertext*, juga sebaliknya. Kerahasiaan kunci ini sangatlah penting, apabila kerahasiaannya terbongkar maka isi pesan akan terbongkar. Berikut adalah skema ilustrasi proses enkripsi dan dekripsi.



Gambar 2.1 Skema Proses Enkripsi dan Dekripsi

Pada gambar 2.1 mengilustrasikan sebuah pesan/*plaintext* dienkripsi menggunakan kunci enkripsi sehingga menjadi *ciphertext* yang akan didekripsi menggunakan kunci dekripsi untuk menghasilkan *plaintext* kembali.

2.4. Algoritme Blowfish

Blowfish atau yang disebut juga "OpenPGP.Cipher.4″ adalah algoritma kunci simetrik cipher blok yang dirancang pada tahun 1993 oleh Bruce Schneier untuk menggantikan DES (Data Encryption Standard). Algoritme Blowfish dibuat untuk digunakan pada komputer yang mempunyai mikroposesor besar (32-bit keatas dengan *cache* data yang besar).

Pada saat itu banyak sekali rancangan algoritme yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa Blowfish bebas paten dan akan berada pada domain publik. Dengan pernyataan Schneier tersebut Blowfish telah mendapatkan tempat di dunia kriptografi, khususnya bagi masyarakat yang membutuhkan algoritme kriptografi yang cepat, kuat, dan tidak terhalang oleh lisensi (Schneier, 2016).

Proses enkripsi data pada algoritme Blowfish terjadi pada jaringan feistel, mengandung fungsi pengulangan sederhana sebanyak 16 kali. Setiap iterasi, terdiri dari sebuah permutasi yang tidak bergantung pada kunci dan sebuah substitusi yang tidak bergantung pada data dan kunci. Semua operasi merupakan penambahan dan XOR pada word 32-bit. Operasi penambahan yang dilakukan hanya merupakan empat indeks *array data lookup* pada setiap iterasi. *Pseudocode* algoritme enkripsinya adalah sebagai berikut:

```
Bagi blok plainteks x menjadi xL dan xR berukuran 32 bit
For i=1 to i<=16
        xL = xL XOR Pi
        xR = F(xL) XOR xR
        Tukar xL dengan xR
Tukar xL dengan xR        //untuk membatalkan pertukaran terakhir
xR = xR XOR P17
xL = xL XOR P18
Gabung xL dengan xR, itulah hasil cipherteks dari blok plainteks X
```

Gambar 2.2 *Pseudocode* Enkripsi Blowfish
Sumber: (Schneier, 1995)

2.5. Paket Kriptografi pada Java

Dalam bahasa pemrograman java disediakan API (*Application Programming Interface*) sebagai *framework* sekuriti ekstensi yang memberikan kemudahan bagi para pengembang untuk mengimplementasikan kriptografi dalam aplikasinya sehingga pengembang dapat lebih fokus kepada bisnis logika aplikasi yang dibangun. Lokasi *framework* ini berada dalam paket *library* bernama ***javax.crypto*** **(**Oracle, 2015). Paket ini tersedia baik untuk pengembang aplikasi berbasis desktop maupun Android. Dalam paket ini disediakan kelas-kelas dan *interface* untuk aplikasi kriptografi yang mengimplementasikan algoritme untuk enkripsi maupun dekripsi. Fitur kelas yang sering digunakan untuk operasi kriptografi adalah *SecretKeySpec* dan *Cipher*.

2.5.1. Secret Key

SecretKey adalah kunci simetri dalam kriptografi yang bersifat rahasia. *SecretKeySpec* adalah kunci yang lebih spesifik dan sering digunakan untuk menampung kunci saat enkripsi/dekripsi. Untuk membuat *instance* dari kelas ini digunakan konstruktor seperti pada Gambar 2.3.

```
public SecretKeySpec (byte[] key,
String algorithm
```

Gambar 2.3 Konstruktor *SecretKey*
Sumber: (Oracle, 2015)

Key adalah data kunci dalam bentuk *array byte*, sedangkan algorithm adalah jenis algoritme yang digunakan. Contoh nilai dari parameter algorithm yang dapat dimasukkan adalah "Blowfish".

2.5.2. Cipher

Kelas *cipher* menyediakan fungsionalitas akses ke implementasi dari kriptografi *cipher* untuk enkripsi dan dekripsi. Kelas *cipher* ini tidak dapat diinstansiasikan secara langsung, melainkan dengan cara pemanggilan method *getInstance* dengan parameter nama transformasi yang diinginkan, opsional dengan provider [ORA-15]. Transformasi adalah sebuah string yang menggambarkan operasi (atau sekumpulan operasi) yang akan dilakukan berdasarkan input yang diberikan, untuk menghasilkan beberapa output. Sebuah transformasi selalu mencantumkan nama dari algoritme kriptografi (misalnya DES, AES. Blowfish) dan dapat diikuti oleh mode umpan balik dan skema *padding*. Setelah diinstansiasikan, *cipher* harus memanggil method init, seperti pada Gambar 2.4.

```
public final void init (int opmode,
Key    key,    AlgorithmParameterSpec
params)
```

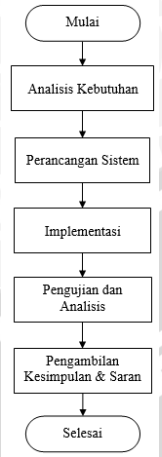
Gambar 2.4 Metode Inisialisasi *Cipher*
Sumber: (Oracle, 2015)

Tiga parameter yang diisikan adalah opmode diisi dengan Cipher.ENCRYPT_MODE untuk operasi enkripsi atau Cipher.DECRYPT_MODE untuk operasi dekripsi, key diisi dengan kunci enkripsi, dan params diisi dengan parameter algoritme yang digunakan.

Langkah terakhir, *cipher* memanggil method doFinal() untuk menjalankan enkripsi/dekripsi dengan parameter byte *plain/ciphertext*.

3. METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan tentang langkah-langkah yang digunakan dalam penelitian. Metodologi penelitian ini terdiri dari enam tahap. Runtutan pengerjaan penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir penelitian

4. ANALISIS DAN PERANCANGAN

4.1 Gambaran Umum Sistem

Implementasi algoritme kriptografi Blowfish pada sistem keamanan ujian *online* berbasis *platform* Android merupakan aplikasi yang bertujuan untuk membantu instansi maupun organisasi khususnya dari sisi peserta ujian dalam melaksanakan ujian *online*. Namun, aplikasi yang dibangun pada penelitian ini berfokus pada aspek sekuritas yang menjadi permasalahan dalam pertukaran informasi secara *online* yang rentan melibatkan pihak di luar sistem karena dalam pengumpulan hasil ujian berbeda dari ujian konvensional dengan menyerahkan lembar ujian secara langsung kepada penguji.

Aplikasi ujian *online* ini menyediakan keamanan dengan memanfaatkan metode enkripsi dari algoritme kriptografi Blowfish untuk memberi perlindungan fisik secara menyeluruh terhadap informasi yang dikirimkan *client* melalui perangkat Androidnya saat ujian berlangsung. Informasi penting seperti *username*, *password*, kode soal, dan jawaban yang menjadi privasi milik peserta akan diamankan dengan enkripsi saat ditransmisikan menuju *web server* untuk disimpan ke dalam *database* ujian *online*. Pada sisi *server* akan melakukan dekripsi terhadap data enkripsi dari peserta jika informasi berhasil diterima.

Selain fasilitas proses enkripsi dan dekripsi, aplikasi sistem keamanan ujian *online* tentunya memiliki fitur untuk menyediakan dan menjawab soal. Namun untuk dapat mengaksesnya peserta harus melakukan validasi terhadap *device* Android yang digunakannya dalam melaksanakan ujian karena sistem akan menolak akses dari *device* yang belum tervalidasi oleh *database*. Setelah validasi, peserta bisa beralih ke fitur *login* untuk memulai menjawab soal dan melihat nilainya setelah menjawab soal terakhir.

4.2 Identifikasi Aktor

Pada tahap ini dilakukan identifikasi terhadap pengguna yang berhubungan dengan sistem keamanan aplikasi. Pada sistem ini terdapat dua jenis aktor yaitu peserta ujian dan administrator. Daftar berikut menerangkan aktor-aktor yang berinteraksi dengan sistem.

a. Peserta Ujian

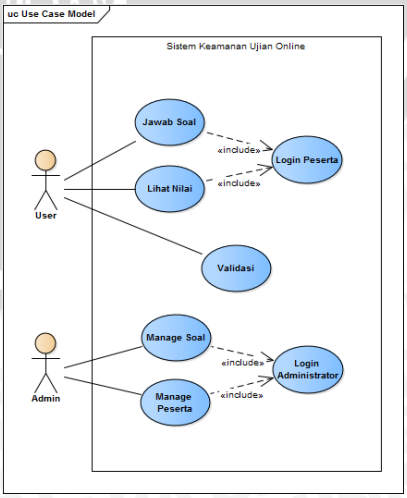
Peserta ujian adalah pengguna yang berinteraksi dengan aplikasi untuk melakukan ujian *online*. Peserta ujian menggunakan aplikasi *client* berupa aplikasi ujian *online* dengan algoritme Blowfish pada perangkat Android yang melakukan interaksi dengan *web server* ujian *online* dengan melakukan *request* maupun mengirim informasi.

b. Administrator

Administrator dapat diasumsikan sebagai penyelenggara maupun pihak yang melakukan manajemen informasi dalam ujian *online*. Administrator melakukan kontrol penuh terhadap pengelolaan data milik peserta ujian dan penyediaan soal-soal yang disajikan dalam pelaksanaan ujian *online*.

4.3 Analisis Kebutuhan Fungsional

Tahap analisis kebutuhan fungsional sistem dilakukan untuk mengetahui kebutuhan-kebutuhan apa saja yang harus bisa dilakukan oleh aplikasi dan sistem. Setiap kebutuhan fungsional didefinisikan menjadi sebuah *use case*. Diagram *use case* pada sistem keamanan aplikasi ujian *online* memanfaatkan algoritme kriptografi Blowfish ini dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 *Use Case* Sistem Ujian Online

4.4 Analisis Kebutuhan Non Fungsional

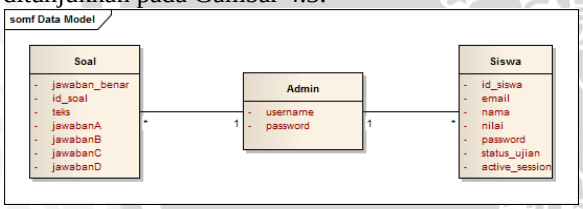
Analisis kebutuhan non-fungsional adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem di mana kebutuhan non-fungsional ini menjadi tujuan penelitian sesuai permasalahan yang dirumuskan sebelumnya dalam BAB I. Adapun parameter dan dekripsi kebutuhan yang akan digunakan dalam pengembangannya ditunjukkan pada Tabel 4.1.

Parameter	Deskripsi Kebutuhan
Security	Sistem keamanan aplikasi harus dapat melakukan perlindungan fisik terhadap segala informasi di dalam ujian milik peserta.
Security	Sistem keamanan aplikasi harus dapat mencegah terjadinya duplikasi akses dari pihak luar terhadap ujian yang dilaksanakan pihak luar melalui *device* yang berbeda.

Tabel 4.1 Spesifikasi kebutuhan non fungsional

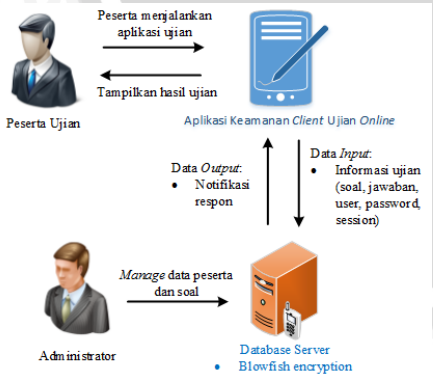
4.5 Perancangan Basis Data Sistem

Perancangan basis data merupakan perancangan manajemen data yang akan digunakan dalam sistem. Perancangan basis data pada sistem ini digambarkan dengan konseptual *class diagram* seperti yang ditunjukkan pada Gambar 4.3.



Gambar 4.2 Perancangan Basis Data

4.6 Perancangan Arsitektur Sistem



Gambar 4.3 Perancangan Arsitektur Sistem

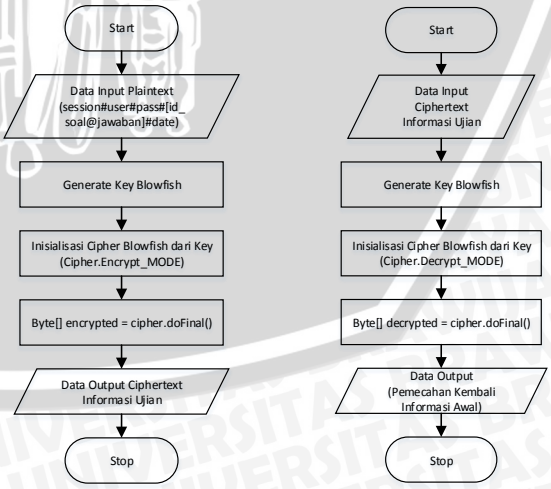
Pada Gambar 4.3 ditunjukkan peserta sebagai *user* menjalankan aplikasi ujian pada *smartphone* untuk menjawab pertanyaan ujian. Selanjutnya sistem melakukan perlindungan enkripsi menggunakan metode Blowfish terhadap segala informasi yang terlibat selama proses ujian berlangsung meliputi soal, jawaban, *user*, *password*, dan *session* sebagai data *input* ke dalam database ujian. Setelah pengolahan data input, sistem melakukan dekripsi terhadap *ciphertext* informasi ujian dan melakukan penilaian terhadap jawaban peserta. Hasil ujian yang telah diolah oleh *web server* akan dienkripsi kembali menggunakan metode Blowfish dan ditampilkan pada aplikasi Android peserta berupa nilai ujiannya.

4.6 Perancangan Sistem Keamanan Aplikasi

Pada aplikasi sistem keamanan ujian *online* dengan memanfaatkan algoritme kriptografi Blowfish ini dititikberatkan pada perlindungan fisik terhadap informasi yang ditransmisikan selama ujian *online* berlangsung lewat perangkat Android peserta dan *web server*. Proses di dalam aplikasi dibagi menjadi dua tahapan yaitu tahap enkripsi informasi dari aplikasi *client* serta tahap dekripsi informasi dari *database*. Berikut adalah ilustrasi mengenai tahap enkripsi dan dekripsi informasi yang terjadi seperti yang ditunjukkan pada Gambar 4.4.

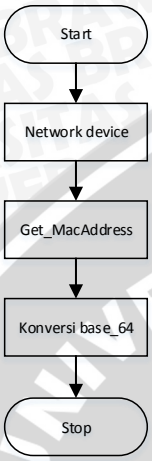
Pada aplikasi sistem keamanan ujian *online* dengan memanfaatkan algoritme kriptografi Blowfish ini dititikberatkan pada perlindungan fisik terhadap informasi yang ditransmisikan selama ujian *online* berlangsung lewat perangkat Android peserta dan *web server*.

Proses di dalam aplikasi dibagi menjadi dua tahapan yaitu tahap enkripsi informasi dari aplikasi *client* serta tahap dekripsi informasi dari *database*. Berikut adalah ilustrasi mengenai tahap enkripsi dan dekripsi informasi yang terjadi seperti yang ditunjukkan pada Gambar 4.4:



Gambar 4.4 Perancangan Proses Enkripsi dan Dekripsi Informasi Ujian Memanfaatkan *library* Kriptografi Blowfish

Di samping keamanan dari sisi informasi ujian yang terenkripsi, diberikan pula keamanan terhadap *device* yang terdaftar dalam pelaksanaan ujian *online* ini dengan diwakili kode unik *device* berupa *MAC Address* sehingga mencegah adanya akses ganda terhadap peserta ujian yang sama untuk melaksanakan ujian dalam satu waktu. Berikut adalah ilustrasi proses pengamanannya seperti pada Gambar 4.5.



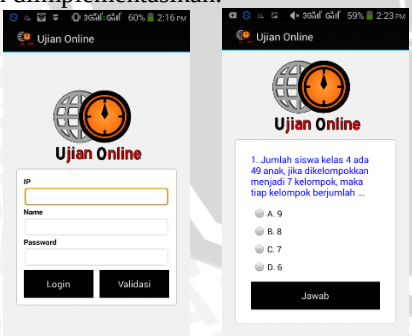
Gambar 4.5 Diagram Alir Pengambilan Informasi *MAC Address*

5. IMPLEMENTASI

Implementasi menerapkan kebutuhan-kebutuhan yang telah diperoleh sebelumnya dari proses analisis kebutuhan.

Proses implementasi aplikasi terdiri dari tiga tahapan, yaitu spesifikasi perangkat yang digunakan sistem, implementasi algoritme dan *source code*, dan implementasi *database*.

Aplikasi diimplementasikan menggunakan bahasa pemrograman Java dan PHP. DBMS yang digunakan adalah MySQL. Gambar 5.1 merupakan *screenshot* sistem keamanan aplikasi ujian *online* yang telah diimplementasikan.



Gambar 5.1 Implementasi antarmuka aplikasi

6. PENGUJIAN

Pengujian aplikasi dilakukan terhadap kebutuhan non-fungsional untuk menjawab permasalahan yang telah dirumuskan dalam rumusan masalah yaitu pengujian *confidentiality* dan pengujian *integrity* terhadap data informasi ujian *online*.

6.1 Pengujian *Confidentiality*

Pengujian kerahasiaan dalam sistem keamanan aplikasi *client* ujian *online* pada sistem operasi Android digunakan untuk mengetahui apakah pelaksanaan proses *confidentiality* informasi data diri, kode soal, beserta jawaban peserta ujian dengan memanfaatkan algoritme kriptografi Blowfish berhasil dilakukan sehingga bentuk asli nilai dapat dikonversi menjadi *ciphertext* sehingga pihak di luar sistem ujian *online* ini tidak mengetahui informasi tersebut.

Pengujian dilakukan dengan teknik *sniffing* menggunakan aplikasi *wireshark* sebagai *tools* untuk melihat lalu lintas data yang terjadi dalam jaringan selama ujian *online* berlangsung.

No	Peserta	Hasil *Sniffing*
1.	Udin	data=XJZupePD8nr9gbcmj72svMi%2FcgqyGoJu7%2Be6rHekvdUN1e(%0ABq1jS8wHklNPpp7982Cu%0A&type=get_soalHTTP/1.1 200 OK Date: Sun, 03 Jan 2016 13:36:59 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 1048 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8
2.	Saiful	data=gVRPicVVzwsT47oZ7WOKICiQ73u0W2syayDdrGMbZEAmdwfTyZ(%0AtmjgX2C3mApNUOsXQb9jN1ORAXP7uNR4cJmSAh1quhBnJ%2FWmuP %0A&type=jawabHTTP/1.1 200 OK Date: Mon, 04 Jan 2016 08:45:34 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 160 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8
3.	Hafizh	data=HmtSEyJCHrDG3HxItmLREOTr4N4XN7rHWYuuOkxftgrnvK4I5U %0A2ie7uPzgruE79ufzG09VphIf62seQoQhJ2Mc7CEznIqW40PmX429 %0A&type=jawabHTTP/1.1 200 OK Date: Sun, 03 Jan 2016 15:20:04 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 160 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8
4.	Putra	data=WJdIa%2F%2F5jtsT47oZ7WOKIN%2FwnGv7oqOu5lFD4otlYjlN%2FvbsCo1If %2B4nw0aZkLLKrw51dmyjkp0BCRSPXkyCJcAwrVQjVQ3a2NZkoO%2FLOuwUFg%0A&t Date: Sun, 03 Jan 2016 13:29:23 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 160 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8

Tabel 6.1 Hasil Pengujian *Confidentiality*

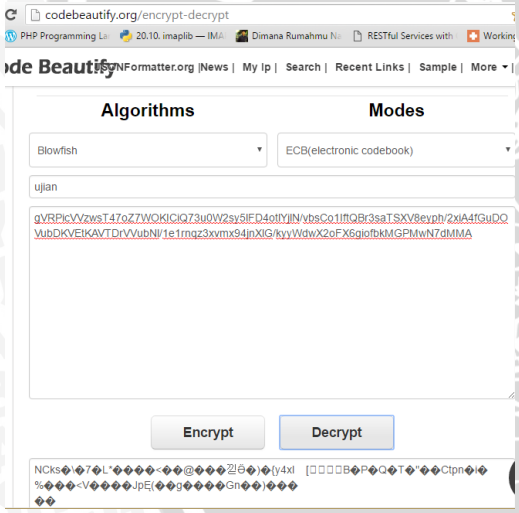
Berdasarkan salah satu hasil aktifitas yang diperoleh dari Tabel 6.1 dapat dilihat bahwa informasi ujian 4 orang peserta yang melaksanakan ujian *online* melalui aplikasi *client* dapat dilakukan enkripsi oleh sistem keamanan aplikasi dengan memanfaatkan algoritme kriptografi Blowfish. Adanya perlindungan fisik terhadap data milik peserta akan meningkatkan privasi atau kerahasiaan terhadap informasi ujian yang ingin dijaga dari upaya penyadapan pihak luar sistem keamanan ujian *online* ini.

6.2 Pengujian *Integrity*

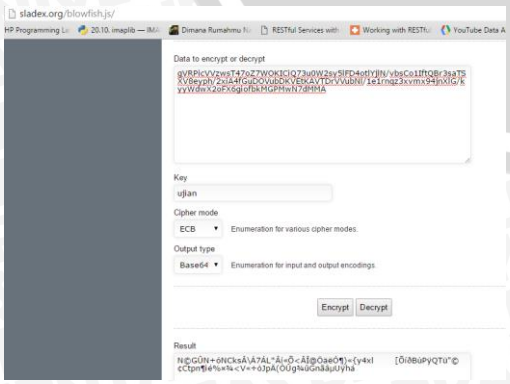
Pengujian integritas diperlukan untuk mengetahui apakah penerapan algoritme kriptografi Blowfish pada sistem keamanan ujian *online* ini dapat

menjaga integritas dari hasil nilai peserta ujian dari upaya intrusi pihak luar terhadap sistem ujian *online*. Sehingga diharapkan informasi hasil nilai peserta yang tersimpan dalam *database server* terjaga dan terlindungi meskipun telah dilakukan upaya untuk mendekripsi informasi yang dikirimkan melalui jaringan ujian *online*.

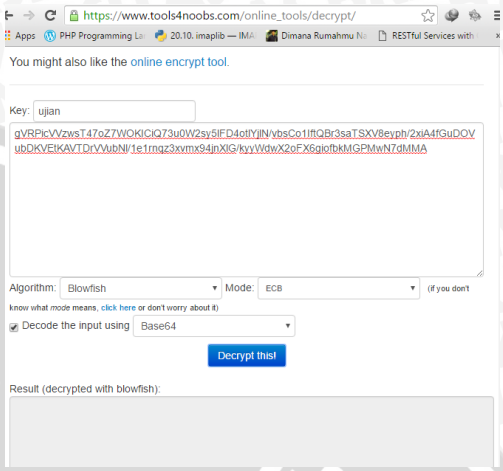
Informasi mengenai paket data yang dikirim oleh peserta melalui aplikasi ujian *online* peserta yang memanfaatkan keamanan algoritme kriptografi Blowfish yang telah di *sniffing dapat* dilihat pada Tabel 6.1. Sebagai contoh pada Tabel 6.1 peserta ujian dengan nama "**Putra**" mengirimkan hasil ujiannya ke dalam sistem ujian *online*. Dari kolom hasil enkripsi ditunjukkan bahwa parameter "data" berisikan *ciphertext* enkripsi informasi ujian. *Value* dalam parameter tersebut akan dilakukan dekripsi sesuai dengan algoritme dan *cipher* mode yang digunakan. Pengujian dekripsi dilakukan melalui ketiga halaman *online tools* dari laman *http://sladex.org/blowfish.js/ ; https://www.tools4noobs.com;* serta dari halaman situs *http://codebeautify.org/encrypt-decrypt*. Hasil uji dekripsi terlihat pada Gambar 5.6, Gambar 5.7, dan Gambar 5.8 berikut.



Gambar 6.1 Percobaan dekripsi melalui halaman *http://codebeautify.org/encrypt-decrypt*



Gambar 6.2 Percobaan dekripsi melalui halaman *http://sladex.org/blowfish.js/*



Gambar 6.3 Percobaan dekripsi melalui halaman *http://www.tools4noobs.com*

Berdasarkan Gambar 5.6, Gambar 5.7, dan Gambar 5.8 telah diujikan percobaan dekripsi *value* yang diperoleh dari hasil *sniffing* sebelumnya. Kunci yang digunakan merupakan kunci acak untuk menerka segala kemungkinan kombinasi karakter yang bisa dicoba agar mendapat kunci yang sesuai. Hasil setelah dilakukan dekripsi memunculkan sekumpulan karakter acak yang membuat format dari *plain* value semakin tidak beraturan. Pada pengujian melalui situs *http://codebeautify.org/encrypt-decrypt* digunakan algoritme Blowfish untuk membuka *ciphertext* ujian, akan tetapi hasil dekripsi menampilkan sebagian besar karakter yang tidak dapat terbaca oleh halaman *browser*. Pada situs *http://sladex.org/blowfish.js/* digunakan algoritme Blowfish. Hasil dekripsi menampilkan karakter yang dapat dibaca halaman *browser* dengan adanya fasilitas *output* menggunakan Base64 agar karakter *ciphertext* dapat dikonversi ke dalam format ASCII. Sedangkan pengujian terakhir melalui situs *https://www.tools4noobs.com* tidak menampilkan *output* apapun karena hanya *ciphertext* yang dibuka dengan algoritme dan kunci yang sesuai saja yang dapat diproses dan ditampilkan *output* nya.

Hasil lebih variasi dengan kunci yang berbeda dapat dilihat pada Tabel 6.2.

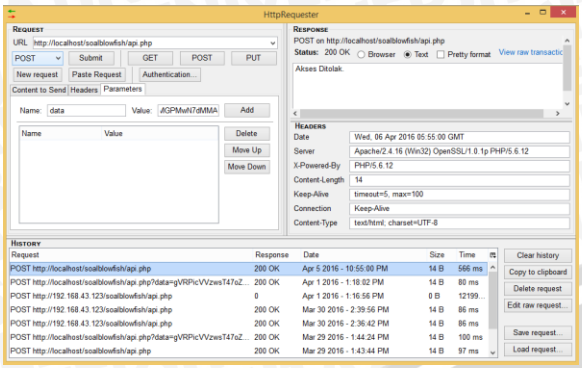
Key	*Decrypt Result*		
	http://codebeautify.org/encrypt-decrypt	***http://sladex.org/blowfish.js/***	***https://www.tools4noobs.com***
Ujian	NCks  \  7  L*     @   -    Ə ){y4xl	N©G ÛN +óNCksÂ\ Á7ÁL*Âí« Õ<ÂÎ@Öä- Ó ¶) «{y4xl	-
Ujian online	Ş /   ت  G   ф  F   X_X      V  uK   ,;H    <   a	B/ÿ²óÌ6+: £ ^Î·R1Që ?¶0¡9¸ñyZ yoUñ<Þ ¯ ãVÙØÊÁd ÖË|Ãszð ®.	-
qwerty	P%= GL !{; R hKxD	t¼ø0t+¹ Bré(Ë*ú ÎJ²ÏíéBNâ ½<ÙÇ_È^÷ .ØMðãÁS TÊ	-
matematika	*Q &bf 0; & 98x D/	¤»×VNd òç- m)UWñr w5-ýUû ÿ°[Hˆ Ûêa{ -óHÏñ«u>	-
12345 67890	麝 #. I ʧ,8(O;n RI $ "]	PíÖsK Êñ E^?¹}Ej'µK W.V/%áñ £üsqæ5úµ Á Zo	-

onlineexam	[7 "`  }^ 6(  |サ ^D=v g  " ^q` ! #! k C+P ʒ Kb% yj 9([T Z--	&Û(j §ëÏÓ y -V !: E<)Eí;`C Êë±GÕ z w1ÚÞJ ·!ì KÉ Ãy"Ì´n n ß ¬á6ú ÅDPg\Øñ ¼S Vuò	-

Tabel 6.1 Daftar Hasil Dekripsi yang Diperoleh melalui Ketiga *Online Tools*

Dari beberapa hasil pengujian untuk tiap-tiap situs *online* dekripsi, asumsi *key* yang dimasukkan oleh *attacker* tidak berhasil membuka *value* dari parameter data yang dikirimkan peserta "**Putra**" melalui aplikasi ujian *online* yang digunakannya. Hasil pengujian tidak dapat mengetahui informasi *value* tersebut dikarenakan ketidaksesuaian *key* yang digunakan sehingga untuk mencoba hingga menemukan *key* yang sesuai dari algoritme Blowfish akan memakan waktu yang cukup lama. Berdasarkan penelitian berjudul "*Implementasi Sistem Keamanan File Menggunakan Algoritme Blowfish pada Jaringan LAN*" (Purwanto, A. 2010) didapat analisa bahwa waktu pemecahan kunci dari algoritme Blowfish yang memiliki batas maksimum kunci sebanyak 56 karakter atau 448-bit, hasil perhitungan waktu pemecahan kunci menggunakan serangan *brute force* untuk *file* berukuran 3000KB dapat menghabiskan waktu yang sangat lama yaitu $4{,}7479 \times 10^{126}$ tahun, jauh lebih lama dibandingkan waktu pemecahan kunci algoritme DES yang menghabiskan waktu $9{,}226 \times 10^{9}$ tahun (Purwanto, A. 2010).

Tahapan pengujian selanjutnya dengan melakukan *request* secara langsung melalui URL yang dikirimkan *attacker* menggunakan aplikasi *HttpRequester*. Di dalam URL dimasukkan hasil penyadapan berupa parameter data dan *value* sesuai informasi yang dikirimkan peserta "**Putra**" pada Tabel 6.1. Masukan URL berdasarkan alamat *server* ujian *online* beserta parameternya yaitu http://localhost/soalblowfish/api.php?data=gVRPicVVzwsT47oZ7WOKICiQ73u0W2sy5lFD4otlYjlN/vbsCo1IftQBr3saTSXV8eyph/2xiA4fGuDOVubDKVEtKAVTDrVVubNl/1e1rnqz3xvmx94jnXlG/kyyWdwX2oFX6giofbkMGPMwN7dMMA.



Gambar 6.4 *Request* HTTP menuju *server* ujian *online*

Berdasarkan Gambar 6.4 ditunjukkan bahwa *attacker* ingin melakukan pengubahan data peserta yang telah tersimpan dalam *database* ujian *online* menggunakan informasi yang berhasil disadap sebelumnya, respon dari sistem keamanan ujian *online* yang muncul adalah penolakan akses. Hal ini dikarenakan adanya proteksi yang ditambahkan dalam sistem keamanan dalam meng-*handle* browser dari *client*. Pemberian *filter* terhadap browser dilakukan agar ruang gerak *attacker* terbatas saat mengirimkan informasi ujian. Pada API *Service* sistem ujian *online*, telah didefinisikan variabel hanya untuk mengenal browser *client* menggunakan "Apache-HttpClient" yang pengaksesannya melalui *device* Android untuk melaksanakan ujian *online* dan dapat mengirimkan informasi ujian.

7. PENUTUP

7.1 Kesimpulan

Berdasarkan hasil analisis perancangan, implementasi, dan pengujian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Berdasarkan hasil pengujian *confidentiality* dengan melakukan *sniffing* terhadap lalu lintas paket data selama ujian *online* berlangsung menggunakan aplikasi Wireshark, hasil penyadapan yang diperoleh menunjukkan bahwa aplikasi mampu merahasiakan bentuk asli informasi ujian peserta dengan enkripsi algoritme kriptografi Blowfish menjadi bentuk yang tidak dapat dimengerti isinya.
2. Berdasarkan hasil pengujian *integrity* dengan melakukan metode *brute force* untuk membuka data enkripsi melalui tiga alamat *online decrypt tool* pada alamat *http://sladex.org/blowfish.js/; https://www.tools4noobs.com;* dan *http://codebeautify.org/encrypt-decrypt* tidak berhasil menemukan kunci yang sebenarnya dari algoritme kriptografi Blowfish yang digunakan aplikasi *client* maupun *web server* ujian *online* sehingga *attacker* tidak dapat melakukan apapun terhadap data masukan peserta ujian yang asli. Sedangkan dari hasil pengujian dengan melakukan *request* menggunakan informasi yang disadap, menunjukkan sistem keamanan ujian *online* telah memberi penolakan akses yang berarti mencegah adanya intrusi dari sistem yang tidak dikenali. Perlakuan ini memberi perlindungan keutuhan data terhadap informasi peserta yang tersimpan dalam *database* ujian *online*.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut terhadap sistem keamanan ujian *online* dengan algoritme kriptografi Blowfish adalah sebagai berikut:

1. Untuk pengembangan lebih lanjut, aplikasi dapat dikembangan dengan membuat variasi dari berbagai tipe soal ujian tidak hanya berupa teks saja, tetapi menggunakan tipe lain seperti *essay, true or false*, maupun jawaban singkat.
2. Perlu dilakukan pengembangan terhadap aspek keamanan yang lain seperti *authentication* dan *availability*.
3. Dapat dikembangkan dalam berbagai *platform* lain seperti *Windows Phone* dan *iOS* mengingat tidak semua peserta yang ingin melaksanakan ujian *online* hanya menggunakan *platform* Android.

DAFTAR PUSTAKA

Canavan, John E. 2001. *Fundamentals of Network Security*. London: Artech House.

Gatliff, Bill. 2003. *Encrypting Data with The Blowfish Algorithm*. India: EE-Times-India.

Ibrahim, Nur Rohmat. 2012. *Kriptografi Algoritme DES, AES/Rijndael, Blowfish Untuk Keamanan Citra Digital dengan Menggunakan Metode Discrete Wavelet Transformation (DWT).* Bandung: STMIK Mardira Indonesia.

Learnline. http://learnline.cdu.edu.au/studyskills/studyskills/differentexams.html, diakses 28 Juni 2015.

Mahyuzir, Tavri D. 1994. *Analisa dan Perancangan Sistem Pengolahan Data*. Jakarta: Penerbit PT. Elex Media Komputindo.

Mandal, Pratap Chnadra. 2012. *Superiority of Blowfish Algorithm*. India: B.P.Poddar Institute of Management & Technology.

Mat Jani H, dan Zughoul O. 2013. *Proposing an Encryption Algorithm Based on DES.* Malaysia: Universiti Tenaga Nasional.

Munir, Rinaldi. 2006. *Kriptografi*. Informatika, Bandung.

Oracle. 2015. *Java Cryptography Architecture* (*JCA*) *Reference Guide*. url:

https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html, diakses 14 Oktober 2015.

Pratondo, Agus. Idestio, Barsyah Dwi. 2010. *Pembangunan Aplikasi Ujian Akhir Semester Online Untuk Mengukur Pencapaian Kompetensi Peserta Didik Studi Kasus : Politeknik Telkom*. Politeknik Telkom.

Purwanto, Anggi., dan Novamizanti, Ledya. 2010. *Implementasi Sistem Keamanan File Menggunakan Algoritme Blowfish pada Jaringan LAN*. Bandung: Program Studi Teknik Komunikasi, Fakultas Teknik Elektro dan Komunikasi, Institut Teknologi Telkom.

Raharjo, Budi. 2005. *Keamanan Sistem Informasi Berbasis Internet*. Jakarta: PT Insan Indonesia – Bandung & PT Indocisc.

Schneier, Bruce. 1995. *The Blowfish Encryption Algorithm*. Springer-Verlag.

Schneier, Bruce. 2016. https://www.schneier.com/cryptography/blowfish/, diakses 30 Juni 2015.

Srivastava, Sangeeta. 2013. *A Repository of Software Requirement Patterns for Online Examination System*. India: Delhi University.

Stalling, William. 2003. *Cryptography and Network Security: Principles and Practices, 5th Edition*. Upper Saddle River: Prentice Hall Inc.

Sutanto, C. A. 2009. *Penggunaan Algoritme Blowfish dalam Kriptografi*. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung.

Tri Massandy, Danang. 2010. *Studi dan Implementasi Cryptography Package pada Sistem Operasi Android*. Bandung: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Widiyanto, A. 2007. *Meningkatkan Keamanan Komputer Anda*. Semarang: Neomedia Press.

Zaki, Ali. 2008. *E-Life Style: Memanfaatkan Beragam Perangkat Teknologi Digital*. Jakarta: Penerbit Salemba Infotek.

Zughoul, O. et al. 2013. *Privacy and Security in Online Examination Systems*. Malaysia: Universiti Tenaga Nasional.