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

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Muhammad Hafizh
NIM: 115060801111080

INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

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

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Muhammad Hafizh
NIM: 115060801111080

Skripsi ini telah diuji dan dinyatakan lulus pada
28 April 2016
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Aryo Pinandito, S.T., M.MT.	Agi Putra Kharisma, S.T., M.T.
NIP. 19830519 201404 1 001	NIK. 201304 860430 1 001

Mengetahui
Ketua Program Studi Informatika

Issa Arwani, S.Kom, M.Sc
NIP. 19830922 201212 1 003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 28 April 2016

Muhammad Hafizh

NIM: 115060801111080

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang selalu memberikan rahmat, hidayah dan berkah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Implementasi Algoritme Kriptografi Blowfish pada Sistem Keamanan Ujian *Online* berbasis Platform Android". Skripsi ini diajukan sebagai salah satu syarat untuk mendapatkan gelar sarjana S-1 di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Keberadaan skripsi ini tidak lepas dari bimbingan dan bantuan berbagai pihak, untuk itu penulis menyampaikan rasa terima kasih sebesar-besarnya kepada:

1. Bapak Aryo Pinandito, S.T., M.MT. selaku Dosen Pembimbing I yang memberikan bimbingan, ilmu, dan motivasi sehingga skripsi ini dapat terselesaikan;
2. Bapak Agi Putra Kharisma, S.T., M.T. selaku Dosen Pembimbing II yang turut memberikan bimbingan, dan arahan untuk kesempurnaan penulisan skripsi ini;
3. Bapak Wayan Firdaus Mahmudy, S.Si, M.T., PhD, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Drs. Marji, M.T., dan Bapak Edy Santoso, S.Si,. M.Kom selaku Ketua, Wakil Ketua 1, Wakil Ketua 2, dan Wakil Ketua 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
4. Bapak Issa Arwani, S.Kom, M.Sc. selaku Ketua Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya;
5. Ayah Deswandi dan Ibu Zul Afrida serta adik-adik penulis Muhammad Iqbal, Atika Triana Putri, dan Muammad Rafi yang senantiasa memberi doa dan dukungan moral sehingga telah banyak membantu lancarnya demi pkelancaran penulisan skripsi ini.
6. Seluruh dosen Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, atas dukungan dan kerjasamanya;
7. Teman-teman Program Studi Teknik Informatika Universitas Brawijaya angkatan 2011 untuk bantuan dan semangatnya; dan
8. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan untuk menyempurnakan skripsi ini. Semoga skripsi ini membawa manfaat bagi pihak lain yang menggunakannya.

Malang, 28 April 2016

Penulis

hafizhemail@gmail.com

ABSTRAK

Muhammad Hafizh. 2016. : Implementasi Algoritme Kriptografi Blowfish pada Sistem Keamanan Ujian *Online* Berbasis *Platform* Android. Skripsi Program Studi Informatika/Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing: Aryo Pinandito, S.T., M.MT. dan Agi Putra Kharisma, S.T., M.T.

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 tetap terjaga.

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

ABSTRACT

Muhammad Hafizh. 2016. : *Blowfish Crypthographic Algorithm Implementation Security System of Online Examination on Android Platform. Thesis Informatics Engineering Program, Program of Information Technology and Computer Science, Brawijaya University. Lecturer*: Aryo Pinandito, S.T., M.MT. *and* Agi Putra Kharisma, S.T., M.T.

Exam is one form of evaluation of the learning process. How far the 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 of the 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*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR KODE

DAFTAR ISTILAH

API

Application Programming Interface (API) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh pengembang aplikasi saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkingkan pengembang untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi lain.

MAC Address

MAC Address sebuah kode unik yang diberikan untuk setiap bagian dari perangkat keras yang terhubung ke Internet. Seperti Internet capable phones, Network Interface Cards untuk komputer desktop, notebook, smartphone, Wireless Access Cards, dan bahkan beberapa kartu memori adalah salah satu perangkat yang bertugas pada MAC Address.

Sniffing

Sniffing merupakan aktifitas penyadapan terhadap lalu lintas data pada suatu jaringan komputer dikarenakan data mengalir bolak-balik pada jaringan.

Web Server

Web server adalah perangkat lunak (software) dalam server yang berfungsi untuk menerima permintaan (request) berupa halaman web melalui protokol HTTP dan atau HTTPS dari client yang lebih dikenal dengan nama browser, kemudian mengirimkan kembali (respon) hasil permintaan tersebut ke dalam bentuk halaman-halaman web yang pada umumnya berbentuk dokumen HTML.

BAB 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 latar belakang di atas, maka rumusan masalah dari penulisan skripsi ini dapat diuraikan sebagai berikut:

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?

1.3 Tujuan

Tujuan dari penelitian ini adalah meningkatkan keamanan informasi peserta ujian *online* dengan cara membangun aplikasi perangkat bergerak yang dapat menjaga kerahasiaan dan keutuhan informasi nilai memanfaatkan algoritme Blowfish.

1.4 Manfaat

Manfaaat yang dapat diperoleh dari penelitian ini adalah:

1. Pengguna dapat meningkatkan keamanan dan kerahasiaan informasi ujian melalui teknik kriptografi yang terenkripsi.
2. Pengguna dapat melakukan penyimpanan hasil ujian secara rahasia (*confidential*) tanpa diketahui orang lain.
3. Pengguna dapat menjaga keutuhan (*integrity*) informasi ujian untuk menghindari upaya penyadapan, pembajakan, dan hal yang menyebabkan kebocoran dan manipulasi informasi memanfaatkan teknik kriptografi yang terenkripsi.

1.5 Batasan masalah

Untuk menghindari adanya kemungkinan semakin berkembangnya masalah, maka penelitian dalam laporan ini dibatasi oleh hal-hal sebagai berikut:

1. Penelitian tidak membahas hitungan matematis dari algoritme Blowfish.
2. Proses bisnis yang dilakukan mencakup enkripsi dan dekripsi terhadap informasi ujian.
3. Soal ujian ditampilkan dalam bentuk *multiple choice* (pilihan ganda) mengenai permasalahan matematika dasar.
4. Enkripsi yang digunakan adalah Blowfish dengan *library* *Java.crypto.*
5. Pengujian keamanan yang dilakukan adalah mencegah seseorang mengetahui informasi paket data ujian *online* dalam jaringan dari serangan pasif menggunakan Wireshark dan dari serangan *brute-force* melalui aplikasi *online* yang disediakan *http://sladex.org/blowfish.js/ ; https://www.tools4noobs.com ;* dan *http://codebeautify.org/encrypt-decrypt* dan aplikasi ekstensi *HttpRequester* dari *browser* Mozilla Firefox.

1.6 Sistematika Penulisan

Sistematika penulisan memberikan gambaran dan uraian dari penyusunan skripsi secara garis besar yang meliputi beberapa bab, antara lain:

BAB I PENDAHULUAN

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Berisi kajian pustaka, referensi, dan sumber-sumber yang berhubungan dengan permasalahan dalam skripsi antara lain mengenai ujian *online*, sistem keamanan, kriptografi, algoritme Blowfish serta teori-teori lainnya sebagai dasar penulisan skripsi.

BAB III METODOLOGI PENELITIAN

Menjelaskan metode atau langkah-langkah yang digunakan dalam penulisan skripsi. Metode apa yang digunakan baik dalam penulisan, perancangan, implementasi, dan pengujian dari sistem yang dibangun.

BAB IV PERANCANGAN

Pada bab ini dijelaskan analisis dan perancangan sistem keamanan ujian *online* dengan algoritme kriptografi memanfaatkan algoritme kriptografi Blowfish yang dapat menjawab permasalahan yang telah diuraikan pada rumusan masalah.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas tentang implementasi sistem keamanan dan pengujian berdasarkan metode penelitian yang telah dibuat untuk diketahui hasilnya.

BAB VI PENUTUP

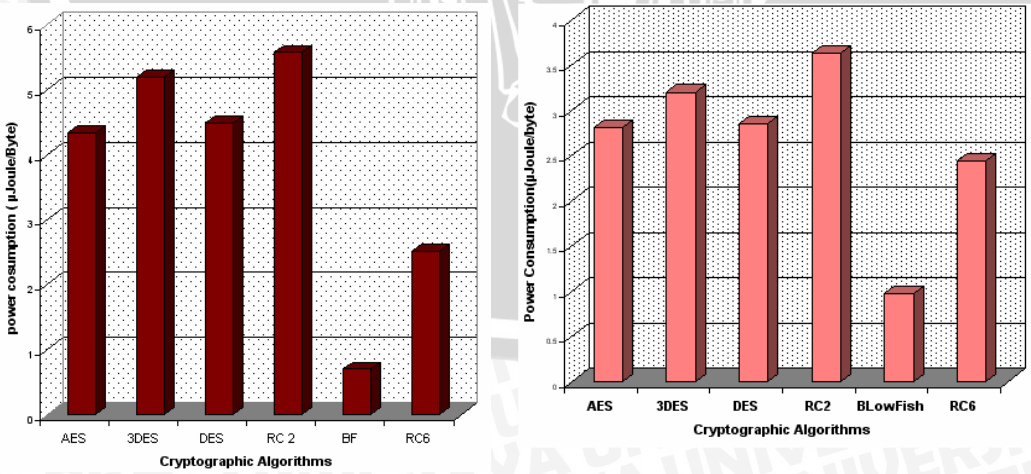
Bab ini memuat tentang kesimpulan yang diperoleh dari pembuatan dan pengujian aplikasi yang dikembangkan dalam penelitian skripsi disertai saran yang dapat dijadikan masukan untuk pengembangan lebih lanjut.

BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI

Kajian pustaka dan dasar teori bertujuan untuk membentuk kerangka teori yang berasal dari literatur yang berhubungan dengan masalah yang diteliti. Bab ini membahas tinjauan pustaka yang digunakan untuk menunjang penulisan skripsi, di mana dalam tinjauan pustaka terdapat kajian pustaka mengenai referensi dari penelitian sebelumnya yang terkait dengan kelebihan algoritme yang diimplementasikan dalam penelitian. Sedangkan dasar teori membahas mengenai teori-teori dasar yang berhubungan dengan konsep dasar ujian *online*, sistem keamanan, kriptografi, Algoritme Blowfish, dan perangkat bergerak.

2.1 Kajian Pustaka

Studi literatur pada bab ini membahas tentang alasan 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.



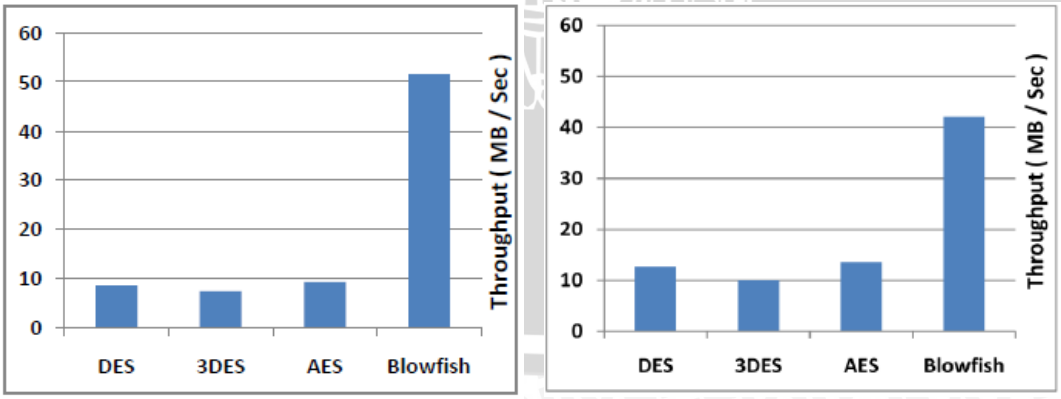
Gambar 2.1 Perbandingan Konsumsi Daya Baterai (µ Joule / Byte) dari Tiap Algoritme Kriptografi Simetris

Sumber: (Elminaam, 2009)

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).

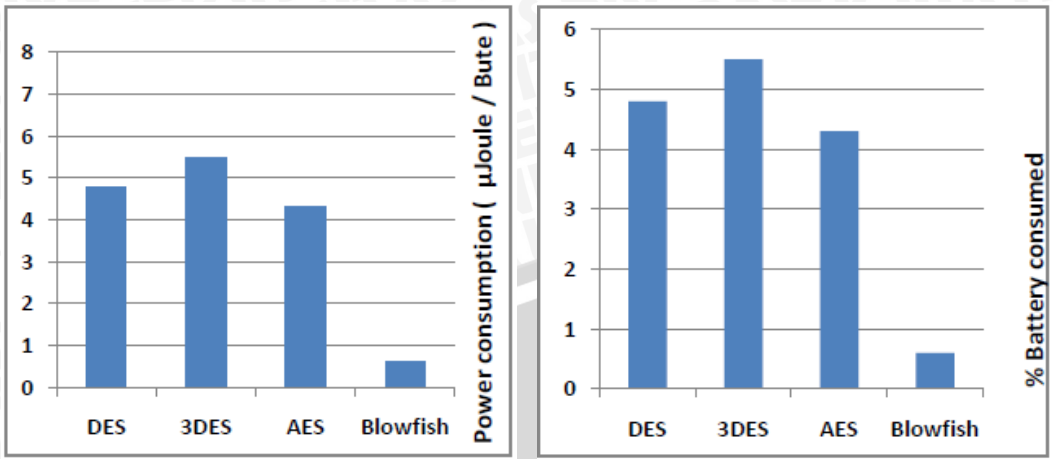
Tabel 2.1 Waktu Proses Enkripsi (Sec) dan Throughput dari DES, 3DES, AES, dan Blowfish dengan Berbagai Ukuran Data (MB/Sec)

Input Size (kb)	**DES**	**3DES**	**AES**	**Blowfish**
50	31	56	56	38
108	35	48	40	45
246	46	109	110	43
320	80	165	162	44
695	145	227	212	47
781	86	171	165	66
900	241	301	260	66
5500	248	307	258	118
7311	1692	1787	1365	105
22300	1716	1796	1366	152
Avg. Time (Sec)	432	496.7	399.4	72.4
Throughput	8.64	7.52	9.35	51.59



Gambar 2.2 *Throughput* dari Tiap Algoritme untuk Proses Enkripsi dan Proses Dekripsi

Sumber: (Mandal, 2012)



Gambar 2.3 : Energi yang dikonsumsi dalam satuan µ Joule/Byte dan persentase (%) konsumsi baterai
Sumber: (Mandal, 2012)

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 Dasar Teori

Dasar teori membahas tentang materi-materi yang berkaitan dengan pengembangan aplikasi sistem keamanan aplikasi *client* ujian *online* dengan algoritme kriptografi Blowfish. Hal-hal yang dibahas pada bagian ini adalah ujian *online*, sistem keamanan, kriptografi, algoritme Blowfish, *package* kriptografi pada Java, dan perangkat bergerak.

2.2.1 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.

2.2.1.1 *Software Requirement Patterns* pada Ujian *Online*

Penggunaan ulang merupakan aktivitas penting dari proses pengembangan perangkat lunak yang bisa memberikan keuntungan terhadap pengembangan perangkat lunak yang dikembangkan. Penggunaan ulang bisa diterapkan dalam berbagai cara. Salah satunya adalah penggunaan ulang suatu kebutuhan perangkat lunak. Penggunaan ulang kebutuhan perangkat lunak bisa membantu *requirement engineer* dalam proses elisitasi, analisis, validasi dan dokumentasi kebutuhan perangkat lunak (Srivastava, 2013).

Dengan mengobservasi dan melakukan penelitian terhadap beberapa perangkat lunak yang memiliki domain yang sama, dalam hal ini adalah sistem ujian *online*, maka dapat dibentuk suatu pola kebutuhan perangkat lunak. Pola kebutuhan perangkat lunak sendiri merupakan salah satu jenis artifak perangkat lunak yang bisa digunakan dalam pengambilan kebutuhan yang juga dapat berpengaruh positif terhadap aktivitas lain seperti analisis, dokumentasi dan validasi (Srivastava, 2013).

Tabel 2.1 merupakan daftar pola kebutuhan fungsional dan non-fungsional perangkat lunak dari sistem ujian onlie.

Table 2.2 *Software Requirement Patterns* pada Sistem Ujian *Online*

Nama Pola Kebutuhan	Deskripsi
User Operation	Untuk menspesifikasikan kemudahan antarmuka dan operasi pengguna dalam menggunakan sistem.
User Registration	Untuk menspesifikasikan bagaimana pengguna baru terdaftar dan informasi disimpan untuk autentikasi.
User Authentication	Untuk menspesifikasikan bahwa seseorang harus login dan terdaftar ke dalam sistem sebelum dapat melakukan aktivitas dalam sistem.

User Authorization	Untuk menspesifikasikan bahwa sekumpulan pengguna berwenang atau tidak dalam mengakses atau melihat suatu aktivitas.
System Operation	Untuk menspesifikasikan platform yang digunakan untuk membangun atau menjalankan sistem.
Information Storage	Untuk menspesifikasikan bagaimana sebuah informasi atau konten direpresentasikan dan disimpan.
Presentation	Untuk mendefinisikan elemen visual dan layout antarmuka dari registrasi OES dan website ujian.
Information Content Representation	Untuk mendefinisikan skema dalam merepresentasikan konten informasi seperti text, gambar, dll.
Formula Calculation	Untuk mendefinisikan formula yang digunakan untuk perhitungan data.
Navigation	Untuk menspesifikasikan bagaimana navigasi pada sistem.
Data Lifespan	Untuk menspesifikasikan berapa lama suatu jenis informasi disimpan secara aktif dalam sistem setelah setelah dihapus atau disimpan dalam data history.
Data History	Untuk menspesifikasikan data mana yang akan dipindah pada waktu tertentu dari penyimpanan sistem yang aktif ke *history record.*
Entity Definition	Untuk mendefinisikan entitas dimana informasi disimpan dan jangka hidupnya.
Inquiry	Untuk mendefinisikan fungsi tampilan layar yang menampilkan informasi tertentu kepada pengguna.
Report	Untuk mendefinisikan laporan yang menampilkan informasi tertentu kepada pengguna.
Response Time	Untuk menampilkan berapa lama waktu yang dibutuhkan bagi sistem untuk merespon suatu *request.*

Throughput, Efficiency	Untuk menspesifikasikan laju sistem dalam melakukan proses input atau output.
Dynamic Capacity	Untuk menspesifikasikan kuantitas dari jenis entitas yang harus bisa dilakukan oleh sistem secara bersamaan.
Static Capacity	Untuk menspesifikasikan kuantitas dari suatu jenis entitas yang harus bisa disimpan secara permanen oleh sistem.
Error Notification	Untuk tampilan jika terjadi kesalahan
Availability	Untuk mendefnisikan kapan sistem tersedia bagi pengguna.
Expandability	Untuk menspesifikasikan cara bagi sistem untuk bisa dikembangkan nanti untuk mengakomodasi pertumbuhan volume.
Installability	Untuk menspesifikasikan seberapa mudah melakukan proses instalasi atau *upgrade* sistem.
Approval	Untuk menspesifikasikan bahwa aksi tertentu harus disetujui oleh pihak berwenang lainnya sebelum dapat dilaksanakan.
Fee	Untuk menspesifikasikan biaya yang harus dihitung, dilaporkan atau dipungut oleh sistem.
Transaction	Untuk mendefinisikan jenis *event* dalam masa hidup suatu entitas atau fungsi untuk masuk ke dalam transaksi.
Interface Interaction	Untuk mendefinisikan jenis antarmuka eksternal yang akan berhubungan dengan sistem lain.
Customizability	Adaptasi terhadap kebutuhan masing-masing pengguna.
Concurrence with requirements	Kesesuaian dengan dokumen kebutuhan perangkat lunak.
Security	Melawan ancaman, korupsi dan pembajakan.

Pada penelitian ini, tidak semua pola kebutuhan diterapkan. Implementasi Algoritme Kriptografi Blowfish pada Sistem Keamanan Ujian *Online* berfokus pada penerapan pola kebutuhan perangkat lunak non-fungsional dalam aspek *security* saja.

2.3 Sistem dan 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 tertenu.
2. *Integrity*

   Aspek *integrity* atau integritas lebih menekankan bahwa suatu informasi tidak boleh diubah tanpa adanya izin dari pemilik informasi tersebut. Jika terdapat perbedaan maka boleh dibilang aspek integritas tidak tercapai. Adanya virus, *trojan horse*, dan sejenisnya merupakan salah satu hal yang umumnya mengakibatkan pengubahan sebuah informasi.
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).

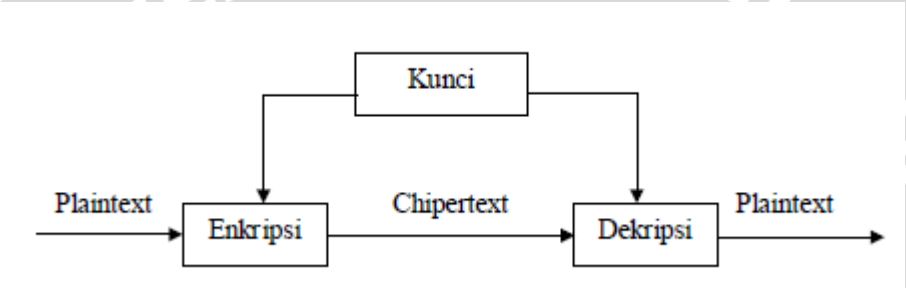
6. *Ciphertext*

   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 2.4 Skema Proses Enkripsi dan Dekripsi

Sumber: (Munir, 2006)

Pada gambar 2.4 mengilustrasikan sebuah pesan/*plaintext* dienkripsi menggunakan kunci enkripsi sehingga menjadi *ciphertext* yang akan didekripsi menggunakan kunci dekripsi untuk menghasilkan *plaintext* kembali.

2.5 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).

Blowfish dikembangkan untuk memenuhi kriteria perancangan sebagai berikut (Sutanto, 2009):

- Cepat, pada implementasi yang optimal Blowfish dapat mencapai kecepatan 26 clock cycle per byte.
- *Compact*, Blowfish dapat berjalan pada memori kurang dari 5 KB.
- Sederhana, Blowfish hanya menggunakan operasi yang simpel: penambahan (*addition*), XOR, dan penelusuran tabel (*table lookup*) pada operand 32 bit. Desainnya mudah untuk dianalisa yang membuatnya resisten terhadap kesalahan implementasi.
- Tingkat keamanan yang bervariasi, panjang kunci yang digunakan Blowfish dapat bervariasi dan bisa sampai sepanjang minimal 32-bit, maksimal 448-bit, *multiple* 8-bit, *default* 128-bit.

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 2.5 *Pseudocode* enkripsi Blowfish
Sumber: (Schneier, 1995)

Blowfish dioptimasikan untuk aplikasi dimana kunci tidak sering berubah, seperti jalur komunikasi atau enkripsi file otomatis. Blowfish jauh lebih cepat dari DES bila diimplementasikan pada 32 bit mikroprosesor dengan cache data yang besar, seperti Pentium dan Power PC, Blowfish tidak cocok untuk aplikasi seperti *packet switching*, dengan perubahan kunci yang sering, atau sebagai fungsi hash satu arah. Kebutuhan memorinya yang besar tidak memungkinkan untuk aplikasi kartu pintar/*smart card* (Sutanto, 2009).

2.6 Package 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* (Tri Massandy, 2010) .

2.6.1 SecretKey

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 yang ditunjukkan pada Gambar 2.5.

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

Gambar 2.6 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`".

2.6.2 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 (Oracle, 2015). 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*. Sebuah transformasi dinyatakan dalam bentuk: "algorithm/mode/padding" dengan algorithm sebagai nama algoritme kriptografi, mode adalah nama mode umpan balik, dan padding adalah nama sebuah skema *padding*.

Setelah diinstansiasikan, *cipher* harus memanggil method init, seperti pada Gambar 2.6.

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

Gambar 2.7 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*.

2.7 Perangkat Bergerak

Perkembangan akan teknologi dari tahun ke tahun memperlihatkan suatu peningkatan yang signifikan. Yang mana saat ini teknologi berkembang menjadi suatu perangkat perangkat yang lebih ringkas ataupun lebih kecil daripada sebelumnya, namun memiliki fungsi yang sama, bahkan bisa lebih canggih daripada sebelumnya. Biasanya perangkat tersebut disebut sebagai perangkat bergerak, karena kita bisa membawa dan menggunakan perangkat tersebut dimanapun dan kemanapun yang kita inginkan.

Perangkat bergerak atau dalam bahasa inggris disebut dengan mobile device adalah perangkat kecil yang memiliki kemampuan komputasi terbatas. Perangkat bergerak sering juga disebut sebagai perangkat genggam atau komputer genggam. Perangkat bergerak pada umumnya terdiri dari dua bagian, yakni layar display sebagai perangkat keluaran dan keyboard atau layar sentuh sebagai masukan. Khusus untuk perangkat telepon pintar (Smartphone), umumnya perangkat layar visual selain berfungsi sebagai sebagai perangkat keluaran, juga berfungsi sebagai perangkat masukan, karena sifatnya yang memiliki layar sentuh pada keseluruhan tampilannya. (Zaki, 2008)

Karena fungsinya yang bervariasi, maka sebuah organisasi T38 dan DuPont Global Mobility merumuskan definisi standar dari perangkat bergerak (Zaki, 2008), yaitu :

1. *Limited Data Mobile Device* : Perangkat yang ukuran layarnya kecil, umumnya layar hanya menampilkan teks, layanan data hanya terbatas ke SMS dan WAP. Contoh perangkat ini adalah telepon seluler.
2. *Basic Data Mobile Device* : Perangkat yang ukuran layarnya menengah. Memiliki navigasi menggunakan menu atau ikon. Layanan yang ditamarkan antara lain surel (email), daftar alamat (Kontak), SMS, dan Web Browser. Contoh perangkat ini adalah telepon pintar (Smartphone).
3. *Enhanced Data Mobile Device* : Perangkat yang ukuran layarnya besar, biasanya menggunakan pena *stylus* untuk layar sentuhnya, dan memiliki fitur-fitur layanan dasar yang dimilik oleh piranti sebelumnya ditambah adanya kemampuan untuk menambah berbagai aplikasi seperti Microsoft Office, dan portal internet. Contoh perangkat ini adalah Pocket PC, dan Tablet.

2.7.1 Aplikasi Perangkat Bergerak

Aplikasi perangkat bergerak (*mobile*) adalah suatu aplikasi yang dibuat secara khusus untuk berjalan pada *mobile device*. Aplikasi *mobile* pada umumnya dikelompokkan berdasarkan *platform*, beberapa kategori *platform* tersebut adalah:

- Android
- iPhone
- Windows Mobile
- Blackberry

Aplikasi *mobile* sendiri secara spesifik juga dikembangkan berdasarkan masing-masing *platform* sesuai dengan kebutuhannya (Native, 2014).

1. ***Native Application***

   Aplikasi yang dibuat dan dipasangkan langsung ke dalam *device* menggunakan bahasa pemrograman khusus untuk membuat aplikasi tersebut. Contohnya adalah untuk membuat aplikasi pada *platform* Android digunakan *Java* dan *Software Development Kit (SDK)*, juga platform iOS yang menggunakan *Objective-C* dan SDK iOS.

2. ***Mobile Web Application***

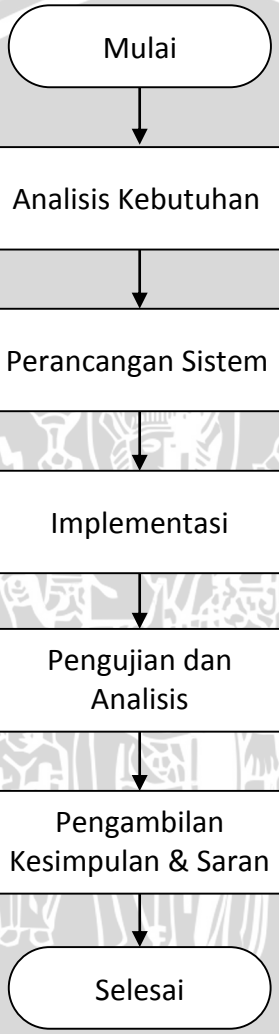
   Aplikasi *mobile* yang dieksekusi menggunakan *browser* yang ada pada *smartphone*, menggunakan bahasa pemrograman *web* seperti PHP & HTML5.

3. ***Hybrid Application***

   Aplikasi *mobile* yang dibuat dan dipasangkan langsung ke dalam *device*, menggunakan bahasa pemrograman *web* yang digabung dengan bahasa pemrograman yang digunakan untuk membuat aplikasi pada *device* yand dituju. Aplikasi dibuat dengan bahasa pemrograman *web*, sedangkan untuk beberapa fitur yang tidak bisa dijalankan menggunakan bahasa pemrograman *web* akan ditulis menggunakan bahasa pemrograman yang sesuai dengan *device* tujuan.

BAB 3 METODOLOGI

Bab ini menjelaskan metode yang digunakan dalam penelitian, yaitu studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis, pengambilan kesimulan dan saran. Runtutan pengerjaan penelitian dapat dilihat pada diagram alir berikut.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Studi literatur berisi dasar teori yang digunakan sebagai sumber acuan untuk penulisan skripsi. Studi literatur dilakukan dengan membaca buku, jurnal, paper dan artikel-artikel di internet. Teori dan pustaka yang berkaitan dengan tugas akhir ini meliputi :

1. Sistem Keamanan
2. Kriptografi
3. Algoritme Blowfish
4. PHP, Java, Android SDK (*Software Development Kit*)

3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan tahap untuk mendapatkan semua kebutuhan (*requirement*) sistem dari perangkat lunak dan semua *stakeholders* yang terlibat di dalamnya. Metode analisis dibuat menggunakan bahasa pemodelan UML (*Unified Modeling Language*). Analisis kebutuhan dalam penelitian ini dilakukan dengan mengidentifikasi kebutuhan dari sistem keamanan aplikasi *client* ujian *online* dengan Algoritme Blowfish.

Proses analisis kebutuhan terdiri dari tiga tahap yaitu identifikasi aktor, analisis kebutuhan fungsional, dan analisis kebutuhan non-fungsional. Identifikasi aktor merupakan tahap pengidentifikasian elemen yang dapat berinteraksi dan melakukan tindakan dalam sistem. Identifikasi aktor menjelaskan deskripsi dari setiap aktor.

Analisis kebutuhan fungsional dilakukan untuk mengetahui fungsi apa saja yang harus terdapat di dalam aplikasi dan sistem. *Use Case Diagram* digunakan untuk mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas sistem dari perspektif *user*. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirements*) sistem yang kemudian akan dimodelkan dalam diagram *use case.* Kebutuhan fungsional yang nantinya akan disediakan oleh sistem keamanan aplikasi yang dibangun antara lain:

- Aplikasi mampu menyediakan fasilitas bagi peserta untuk melakukan validasi terhadap *device* yang didaftarkan dalam *database* ujian *online* dan fasilitas *login* apabila *device* peserta telah terdaftar dan ingin melaksanakan ujian *online*.
- Aplikasi mampu menyediakan fasilitas soal ujian pilihan ganda untuk dijawab oleh peserta dan menampilkan perolehan nilai berdasarkan jumlah jawaban benar yang dikerjakan.
- Aplikasi mampu menyediakan fasilitas enkripsi dan dekripsi terhadap segala paket informasi yang terkirim dengan algoritme kriptografi Blowfish.

Sedangkan analisis kebutuhan non-fungsional merupakan proses analisis untuk menjawab permasalahan sesuai tujuan akhir dari penelitian yang dilakukan. Kategori kebutuhan non-fungsional yang dibutuhkan adalah *security* (keamanan) dalam aspek *confidentiality* (kerahasiaan) dan *integrity* (keutuhan) sebagai fokus pembahasan dalam penelitian ini.

3.3 Perancangan Sistem

Perancangan sistem dilakukan setelah tahap analisis kebutuhan. Sistem dirancang berdasarkan kebutuhan-kebutuhan yang telah didefinisikan pada tahap analisis kebutuhan. Terdapat empat langkah dalam perancangan sistem yaitu perancangan arsitektur sistem aplikasi, perancangan basis data, perancangan *activity diagram*, dan perancangan diagram alir sistem keamanan aplikasi.

Perancangan arsitektur sistem merupakan tahap awal yang dilakukan dalam perancangan sistem. Pada tahap ini digambarkan keseluruhan elemen yang terdapat pada sistem keamanan aplikasi *client* ujian *online*. Selain itu digambarkan pula interaksi antara elemen-elemen tersebut.

Perancangan basis data digambarkan dalam model konseptual *class diagram* untuk menjelaskan tabel-tabel beserta entitas yang diperlukan dalam *database* sistem ujian *online*. Perancangan *activity diagram* dilakukan menggambarkan urutan aktifitas dalam proses bisnis sistem ujian *online* juga penggambaran visual dari proses pengujian yang akan diterapkan.

Perancangan diagram alir sistem keamanan aplikasi dilakukan untuk menggambarkan proses perlindungan terhadap informasi ujian *online* yang dikirimkan oleh peserta ujian maupun diterima oleh *web server* melalui proses enkripsi dan dekripsi menggunakan algoritme kriptografi Blowfish. Selain itu digambarkan pula diagram alir proses perlindungan *MAC Address* pada *device* Android peserta ujian dan diagram alir dari beberapa proses seleksi yang diterapkan oleh keamanan *API Service* dari ujian *online* ini.

3.4 Implementasi

Implementasi dilakukan untuk merealisasikan model yang telah dirancang pada proses sebelumnya, menjadi sebuah aplikasi yang dapat memenuhi kebutuhan atau *requirement* dari aplikasi. Proses implementasi dimulai dengan menjabarkan spesifikasi singkat *environment* dari sistem yang akan dibangun. Spesifikasi ini terdiri dari spesifikasi *hardware* dan *software* yang dibutuhkan agar aplikasi sistem keamanan ujian *online* dapat berjalan dengan normal. Langkah selanjutnya dilanjutkan ke pembuatan keamanan enkripsi terhadap informasi ujian peserta memanfaatkan algoritme kriptografi Blowfish pada perangkat bergerak menggunakan bahasa pemrograman Java yang nantinya akan didekripsi kembali dari sisi *server.* Sedangkan *database server* yang digunakan dalam pelaksanaan ujian *online* ini dibangun dengan menggunakan bahasa pemrograman PHP dengan basis data MySQL dari aplikasi XAMPP.

3.5 Pengujian dan Analisis

Pengujian perangkat lunak dilakukan agar dapat diketahui apakah perangkat lunak dapat berjalan optimal sesuai dengan spesifikasi, tujuan, dan kebutuhan yang telah dirancang sebelumnya. Strategi pengujian perangkat lunak yang akan digunakan antara lain:

1. Pengujian Kerahasiaan (*Confidentiality*), dilakukan untuk memastikan privasi peserta terhadap informasi ujiannya terjaga. Dengan metode Blowfish maka sistem keamanan aplikasi akan memberi perlindungan fisik pada seluruh informasi yang terkirim saat ujian *online* berlangsung sehingga dapat diubah menjadi bentuk *ciphertext* yang sulit dimengerti isinya. Dengan metode ini juga akan meminimalkan celah bagi pihak lain untuk mengetahui isi informasi ujian peserta yang bukan termasuk kewenangannya saat akan dilakukan *sniffing* terhadap lalu lintas paket data ujian dalam suatu jaringan.
2. Pengujian Integritas (*Integrity*), dilakukan untuk mengetahui keutuhan dari segala informasi peserta yang telah tersimpan di dalam *database* setelah melaksanakan ujian *online*. Informasi ujian peserta berupa *username*, *password*, *nomor soal,* dan jawaban peserta akan diubah dari bentuk *plaintext* menjadi *ciphertext* dengan enkripsi memanfaatkan algoritme kriptografi Blowfish agar informasi yang disimpan tetap utuh selain untuk merahasiakan isinya. Hasil enkripsi informasi ujian berupa *ciphertext* diuji keamanannya dengan metode dekripsi *brute force* sehingga dapat diketahui kehandalan dari algoritme Blowfish yang diimplementasikan. Selain itu ditambahkan pengujian integritas dari ketahanan sistem keamanan dalam menanggapi adanya intrusi parameter informasi ujian yang baru oleh pihak diluar sistem ujian *online*. Pengujian ini mengacu pada pengambilan parameter yang diperoleh dari aktifitas penyadapan.

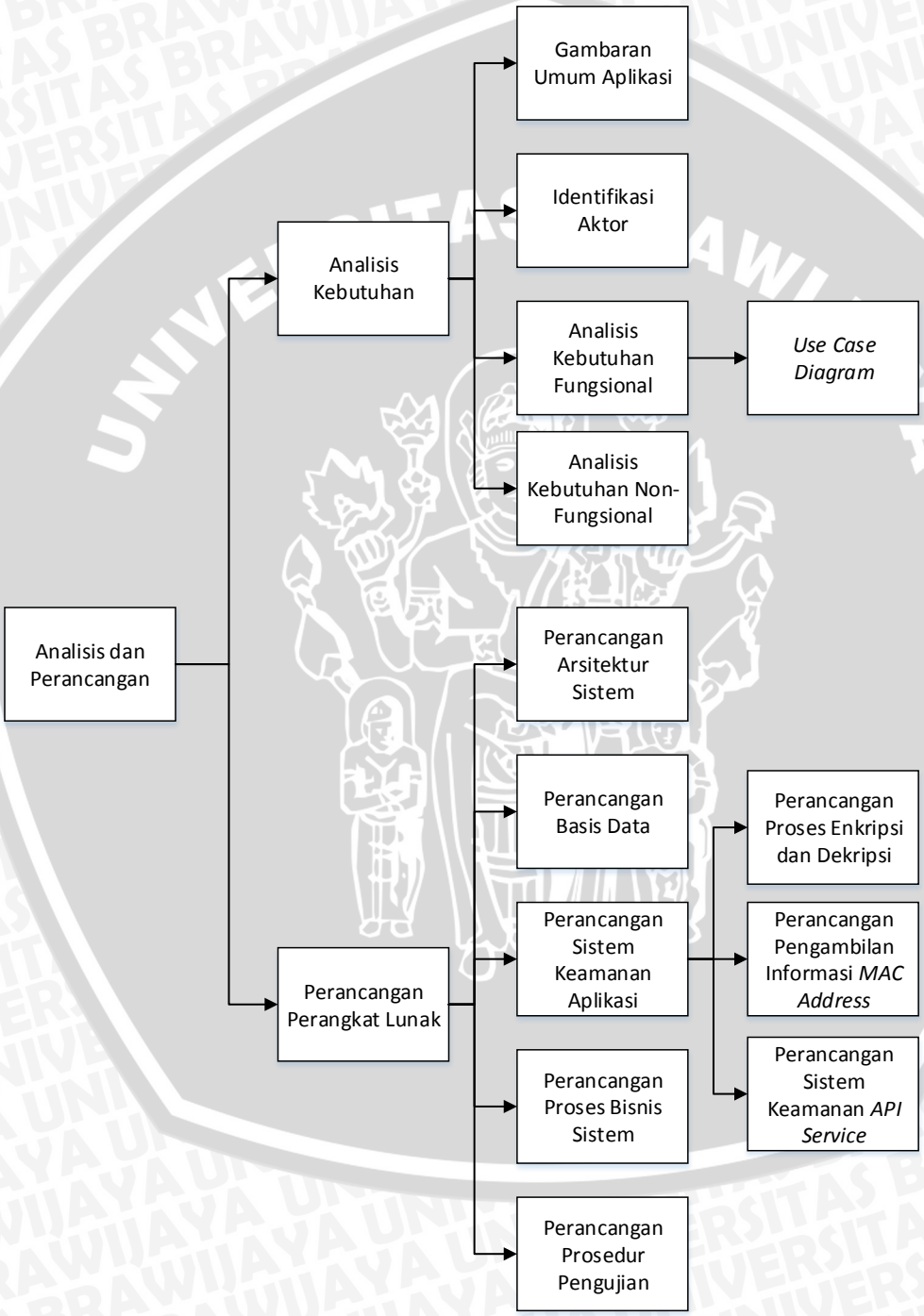
Pada akhir pengujian akan dilakukan analisis pembahasan terhadap hasil ujian yang telah dilakukan. Analisis ini diperlukan untuk mengetahui keberhasilan aspek kerahasiaan dan integritas dari pelaksanaan ujian *online* yang memanfaatkan algoritme kriptografi Blowfish pada sistem operasi Android.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan selesai dilakukan, mulai dari analisis, perancangan, implementasi serta pengujian. Kesimpulan diambil dari hasil pengujian terhadap aplikasi yang telah dibangun. Tahap terakhir penulisan adalah saran yang dimaksudkan untuk memperbaiki kekurangan yang terjadi dan menyempurnakan penulisan serta mengembangkan penelitian lebih lanjut.

BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN

Bab ini membahas mengenai analisis dan perancangan aplikasi ujian *online*. Gambar 4.1 menjelaskan struktur dan langkah-langkah analisis dan perancangan aplikasi perangkat bergerak sistem keamanan ujian *online* memanfaatkan implementasi algoritme Blowfish.



Gambar 4.1 Diagram Pohon Analisis dan Perancangan

4.1 Analisis Kebutuhan

Pada tahap analisis kebutuhan dilakukan identifikasi seluruh kebutuhan *(requirements)* aplikasi sistem keamanan ujian *online* dengan algoritme kriptografi Blowfish. Tahap analisis kebutuhan bertujuan untuk menganalisa secara jelas semua daftar kebutuhan pengguna yang akan menjadi fitur-fitur sistem. Analisis kebutuhan dibagi menjadi empat tahap yaitu gambaran umum aplikasi, identifikasi aktor, analisis kebutuhan fungsional sistem dan analisis kebutuhan non fungsional sistem.

4.1.1 Gambaran Umum Aplikasi

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. Kunci simetris pada algoritme Blowfish ditanamkan pada kedua sisi baik dari aplikasi Android maupun *server* sehingga mencegah adanya pemberian kunci untuk dapat mengirim maupun membuka informasinya.

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.

4.1.2 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. Tabel 4.1 menerangkan aktor-aktor yang berinteraksi dengan sistem.

Tabel 4.1 Daftar Aktor

Aktor	Deskripsi
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.
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*.

4.1.3 Analisis Kebutuhan Fungsional

Tahap ini merupakan identifikasi kebutuhan fungsional dari sistem aplikasi. Kebutuhan-kebutuhan dari sistem harus dirancang sesuai dengan kebutuhan dari pengguna. Tabel 4.2 menjelaskan daftar fitur utama yang dimiliki oleh aplikasi ditinjau dari peserta ujian dan Tabel 4.3 menjelaskan daftar kebutuhan dari sisi administrator.

Tabel 4.2 Kebutuhan Fungsional Peserta Ujian

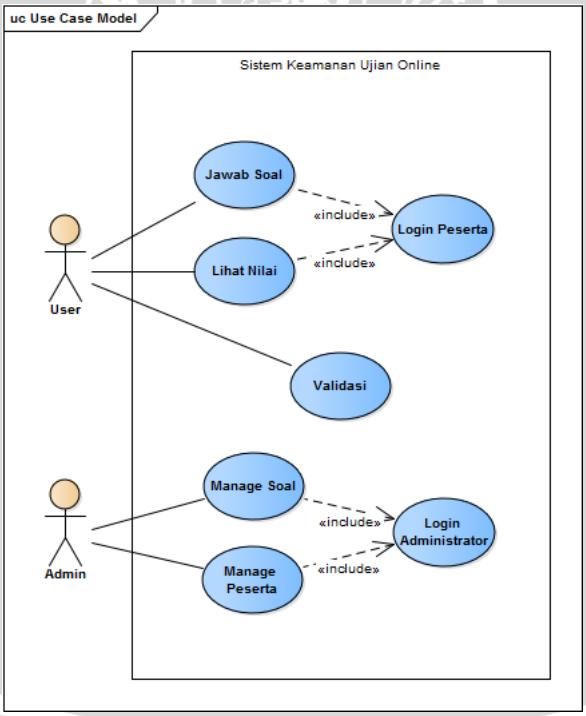
Nomor SRS	Kebutuhan	*Use Case*
SRS_1_01	Aplikasi harus menyediakan fitur validasi untuk mendaftarkan perangkat keras Android peserta yang digunakan dalam melakukan ujian *online*.	Validasi
SRS_1_02	Aplikasi harus menyediakan fitur login bagi peserta ujian *online* berdasarkan *username* dan *password* yang didaftarkan.	Login Peserta
SRS_1_03	Aplikasi harus menyediakan fitur untuk menjawab soal sesuai yang disajikan oleh *database* ujian *online*.	Jawab Soal
SRS_1_04	Aplikasi harus menyediakan fitur untuk menampilkan nilai hasil ujian peserta setelah melaksanakan ujian *online*.	Lihat Nilai

Tabel 4.3 Kebutuhan Fungsional Administrator

Nomor SRS	Kebutuhan	*Use Case*
SRS_2_01	Aplikasi harus menyediakan fitur login bagi administrator.	Login Administrator
SRS_2_02	Aplikasi harus menyediakan fasilitas bagi administrator untuk menambah maupun menghapus soal dalam *database* ujian *online*.	Manage Soal
SRS_3_03	Aplikasi harus menyediakan fasilitas kontrol penuh bagi administrator untuk manajemen data peserta.	Manage Peserta

4.1.3.1 *Use Case Diagram*

Diagram use case adalah diagram yang menggambarkan kebutuhan dari aplikasi dan aktor yang berinteraksi dengan aplikasi. Gambar 4.2 merupakan *use case diagram* dari implementasi sistem keamanan ujian *online* dengan algoritme kriptografi Blowfish pada *platform* Android.



Gambar 4.2 *Use Case Diagram*

4.1.4 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 4.4 berikut.

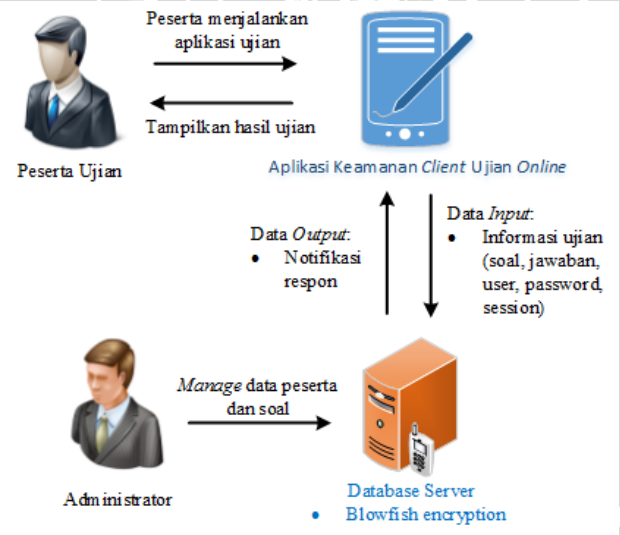
Tabel 4.4 Spesifikasi Kebutuhan Non-Fungsional

Parameter	Deskripsi kebutuhan
Security	Sistem keamanan aplikasi harus dapat melakukan perlindungan fisik terhadap segala informasi di dalam ujian milik peserta memanfaatkan proses enkripsi dari algoritme Blowfish untuk menjaga kerahasiaan selama pertukaran informasi ujian.
Security	Sistem keamanan aplikasi harus dapat mencegah terjadinya pengubahan atau perbaruan data peserta ujian yang telah tersimpan dalam *database* dengan adanya perlindungan algoritme Blowfish.

4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak membahas mengenai perancangan pada aplikasi perangkat bergerak yang akan dibangun. Perancangan dilakukan melalui beberapa tahap, yaitu perancangan arsitektur sistem, perancangan basis data, perancangan proses bisnis aplikasi dengan *activity diagram*, dan perancangan diagram alir sistem keamanan aplikasi.

4.2.1 Perancangan Arsitektur Sistem



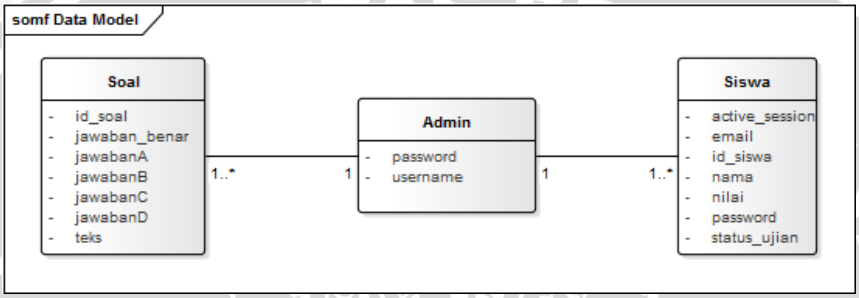
Gambar 4.3 Perancangan Arsitektur Sistem

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.

4.2.2 Perancangan Basis Data

Perancangan basis data digunakan untuk merepresentasikan penyimpanan data-data hasil ujian peserta dalam basis data ujian *online*. Perancangan basis data digambarkan dalam bentuk diagram hubungan entitas yang menyediakan tampilan grafis dari struktur sistem melalui entitas dan relasi antar entitas di dalamnya.



Gambar 4.4 Perancangan Basis Data

Dalam Gambar 4.4 menjelaskan perancangan basis data yang akan digunakan pada sistem. Terdapat tiga tabel yang digunakan dalam basis data, yaitu: tabel *Soal*, tabel *Admin*, tabel *Siswa*.

4.2.3 Perancangan Sistem Keamanan Aplikasi

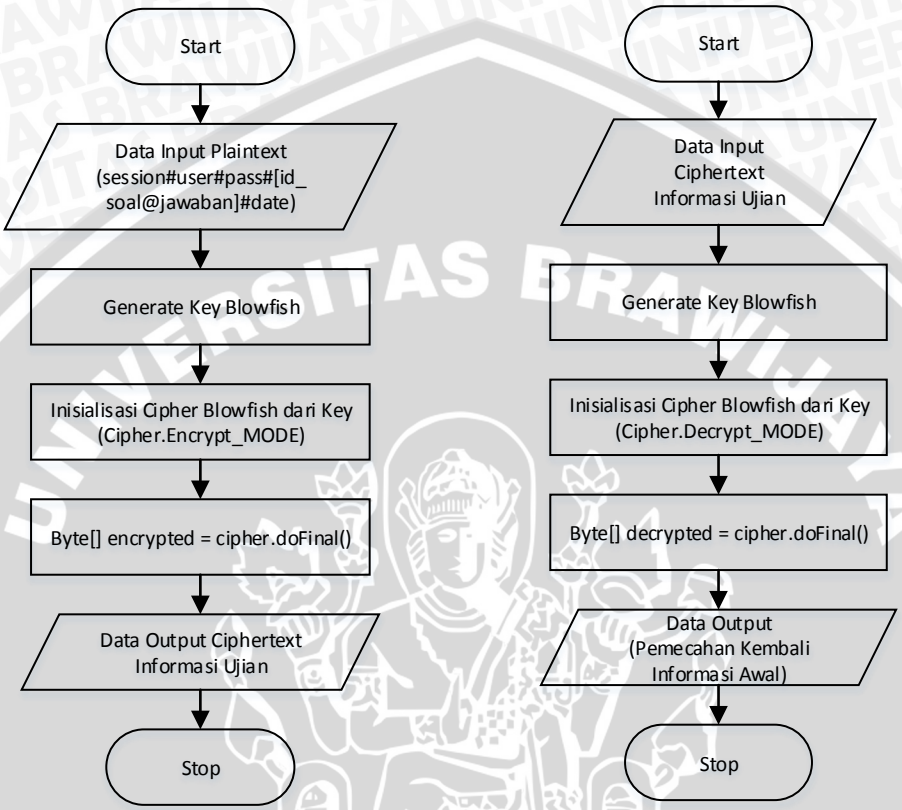
Perancangan sistem keamanan aplikasi menjelaskan lapisan-lapisan keamanan yang diberikan pada sistem ujian *online* untuk mencegah setiap kemungkinan celah kesalahan yang dapat menyebabkan informasi ujian yang dikirimkan dan yang tersimpan disadap maupun diubah oleh pihak luar.

4.2.3.1 Perancangan Proses Enkripsi dan Dekripsi

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 4.5.

Pada Gambar 4.5 disajikan alur proses enkripsi dari perangkat *mobile* peserta dan proses dekripsi dari *web server* ujian *online*, dimana informasi peserta ujian berupa data pribadi beserta soal, jawaban, dan waktu ujian yang masih dalam bentuk *plaintext* menjadi *input* dalam aplikasi. Tahapan pertama di dalam sistem keamanan aplikasi akan melakukan *generate key* untuk membuat kunci yang akan

digunakan dalam proses enkripsi. Dari kunci ini dapat dilakukan inisialisasi *cipher* untuk memilih metode kriptografi yang diinginkan, yaitu menggunakan algoritme Blowfish. Selanjutnya informasi *plaintext* sebagai *input* dalam tipe data *byte* akan dilakukan proses enkripsi untuk dapat dihasilkan *ciphertext*.



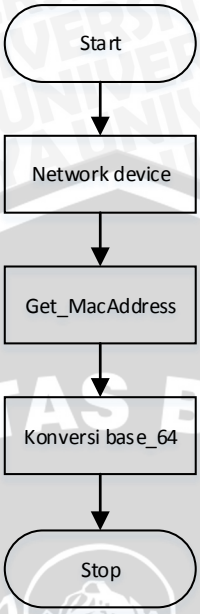
Gambar 4.5 Perancangan proses enkripsi dan dekripsi informasi ujian memanfaatkan *library* kriptografi Blowfish

Tahap-tahap yang ada pada proses dekripsi informasi ujian tidak jauh beda dengan proses enkripsi informasi ujian, dimana inisialisasi *cipher* menggunakan mode *decrypt* untuk membuka informasi yang terenkripsi. Selanjutnya adalah proses dekripsi terhadap informasi ujian. Hasil *output* yang disajikan dalam tahap dekripsi merupakan informasi awal yang telah dipecah sesuai struktur informasi yang telah dirancang.

4.2.3.2 Perancangan Pengambilan Informasi *MAC Address*

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 melalui *device* yang tidak dikenali oleh sistem keamanan ujian *online*

melalui mekanisme validasi. Berikut adalah ilustrasi proses pengambilannya seperti pada Gambar 4.6.



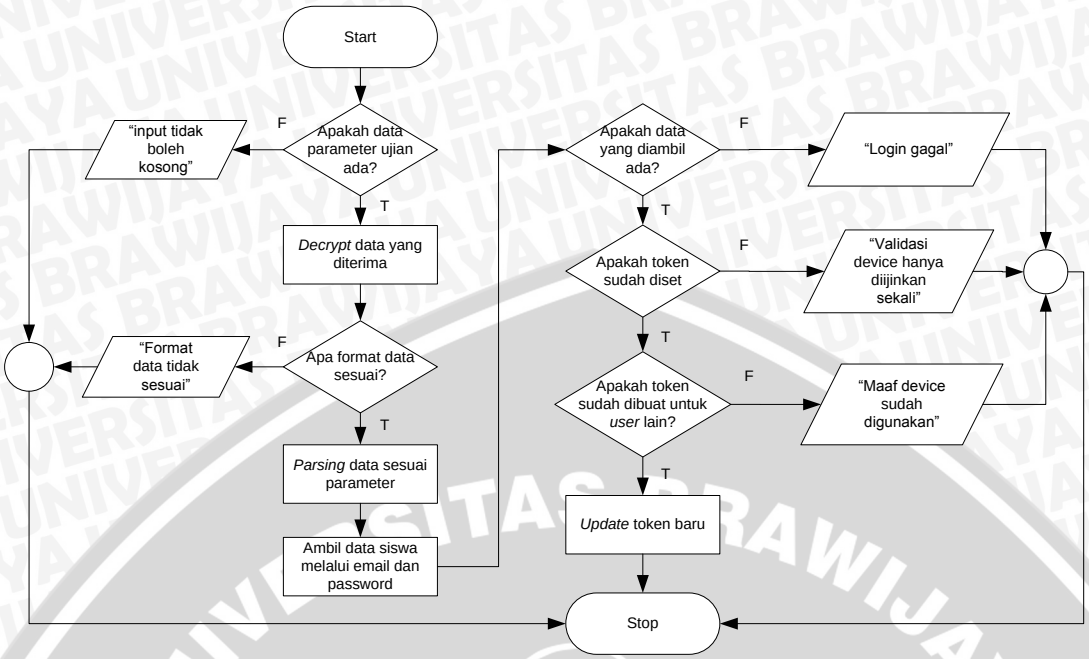
Gambar 4.6 Diagram Alir Pengambilan *MAC Address*

Pada Gambar 4.6 menjelaskan langkah-langkah pengambilan dari *MAC Address* milik *device* peserta ujian yang digambarkan melalui diagram dengan tahap awal pemanfaatan *network device* yang terdapat dalam setiap perangkat bergerak untuk dapat terhubung dalam sebuah jaringan. Setelah terhubung akan diketahui *MAC Address* untuk mengidentifikasi kode fisik perangkat bergerak. Selanjutnya *MAC Address* yang diperoleh akan dilakukan konversi dari data binernya menjadi teks tertentu. *Output* dari proses ini akan digabungkan bersama informasi ujian yang lain dalam wujud token untuk diamankan dengan enkripsi algoritme kriptografi Blowfish.

4.2.3.3 Perancangan Sistem Keamanan dari API *Service*

Perancangan sistem keamanan dari API dilakukan untuk memberi serangkaian aturan yang harus dipenuhi oleh pengguna aplikasi agar dapat melaksanakan proses ujian dengan benar.

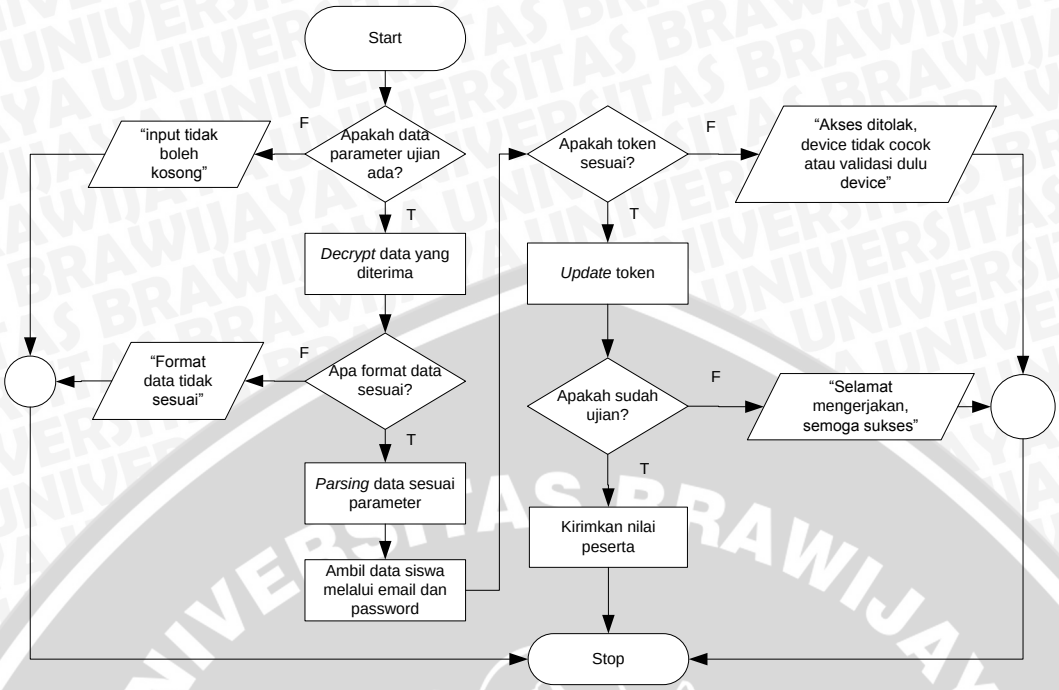
Di dalam sistem keamanan ujian *online* yang dibangun, terdapat beberapa alur yang dijalankan oleh sistem saat peserta melaksanakan proses bisnis ujian yang disediakan oleh aplikasi ujian *online* mulai dari proses pendaftaran akun untuk peserta ujian hingga proses penjawaban soal ujian.



Gambar 4.7 Diagram Alir Pengamanan Proses Validasi

Dalam Gambar 4.7 direpresentasikan diagram alir dari proses keamanan saat peserta melakukan validasi untuk mendaftarkan *device* yang digunakan untuk melaksanakan ujian *online*. Proses yang dilakukan pertama kali adalah proses pengecekan parameter ujian yang telah dikirimkan dari aplikasi *client* peserta ujian dalam bentuk *ciphertext* hasil dari enkripsi algoritme kriptografi Blowfish. Jika data ujian benar-benar diterima, proses dekripsi informasi akan dilakukan. Berikutnya adalah pengecekan *format* informasi yang sesuai dengan *format* yang ditentukan oleh sistem keamanan ujian *online* untuk menjamin bahwa informasi ujian yang ditransmisikan menuju *database* berasal dari perangkat asli peserta ujian. Berikutnya adalah proses *parsing* atau menguraikan pecahan sesuai susunan informasi yang dapat dimengerti *server* ujian dan mengambil data melalui *email* dan *password* peserta yang mengirimkan jawabannya.

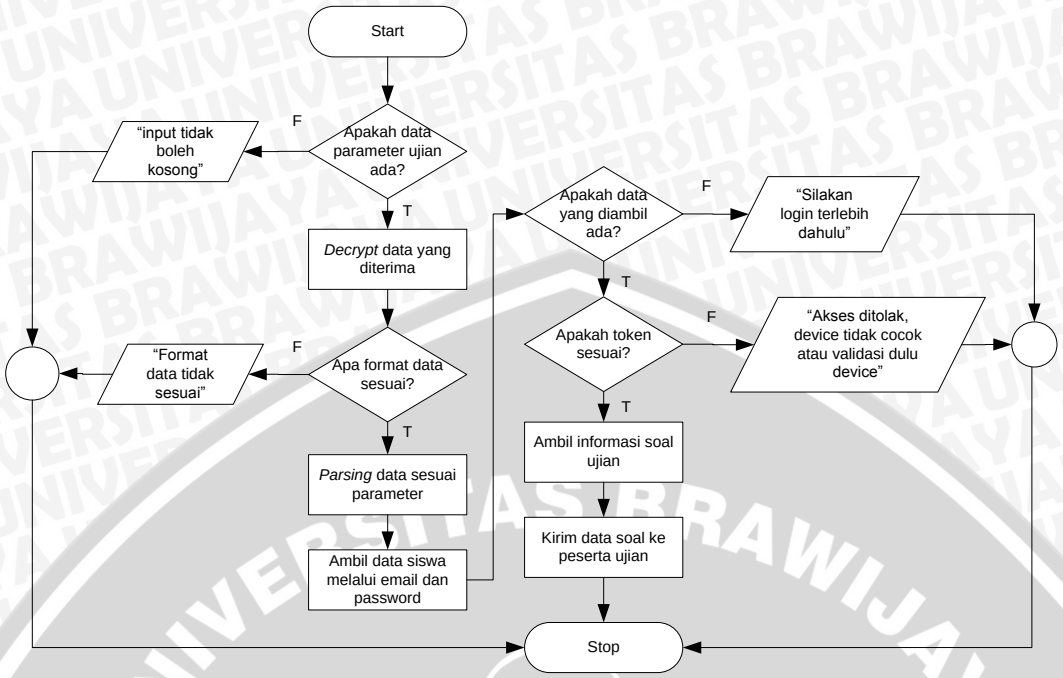
Selanjutnya sistem melakukan pengecekan validitas data pribadi peserta dan token yang akan didaftarkan dalam *database* ujian *online*. Dalam proses validasi ini hanya sebuah token yang akan dimiliki oleh seorang peserta melalui *device* yang didaftarkannya. Jika *device* sudah pernah didaftarkan oleh peserta ujian lain maka akan muncul respon pesan bahwa *device* sudah digunakan. Selain itu jika *device* telah terdaftar sebelumnya, maka tidak dapat dilakukan validasi ulang karena sistem hanya mengijinkan satu kali validasi. Untuk *device* yang benar-benar belum pernah didaftarkan maka sistem ujian *online* akan melakukan *update* terhadap token baru untuk peserta ujian.



Gambar 4.8 Diagram Alir Pengamanan Proses Login

Dalam Gambar 4.8 direpresentasikan diagram alir dari proses keamanan saat peserta melakukan *login* dari aplikasi untuk melaksanakan ujian *online*. Beberapa mekanisme keamanan yang dilakukan serupa dengan proses validasi, yaitu pada pengecekan parameter ujian yang telah dikirimkan dari aplikasi *client* peserta ujian dalam bentuk *ciphertext* hasil enkripsi. Jika seleksi berhasil dan data telah diterima maka dilakukan dekripsi dari informasi ujian. Selanjutnya adalah pengecekan *format* informasi ujian sesuai dengan *format* yang ditentukan sistem keamanan ujian *online.* Jika *format* sesuai, proses *parsing* terhadap informasi ujian sesuai susunan parameter mulai dilakukan hingga pengambilan data melalui *email* dan *password* peserta yang mengirimkan jawabannya.

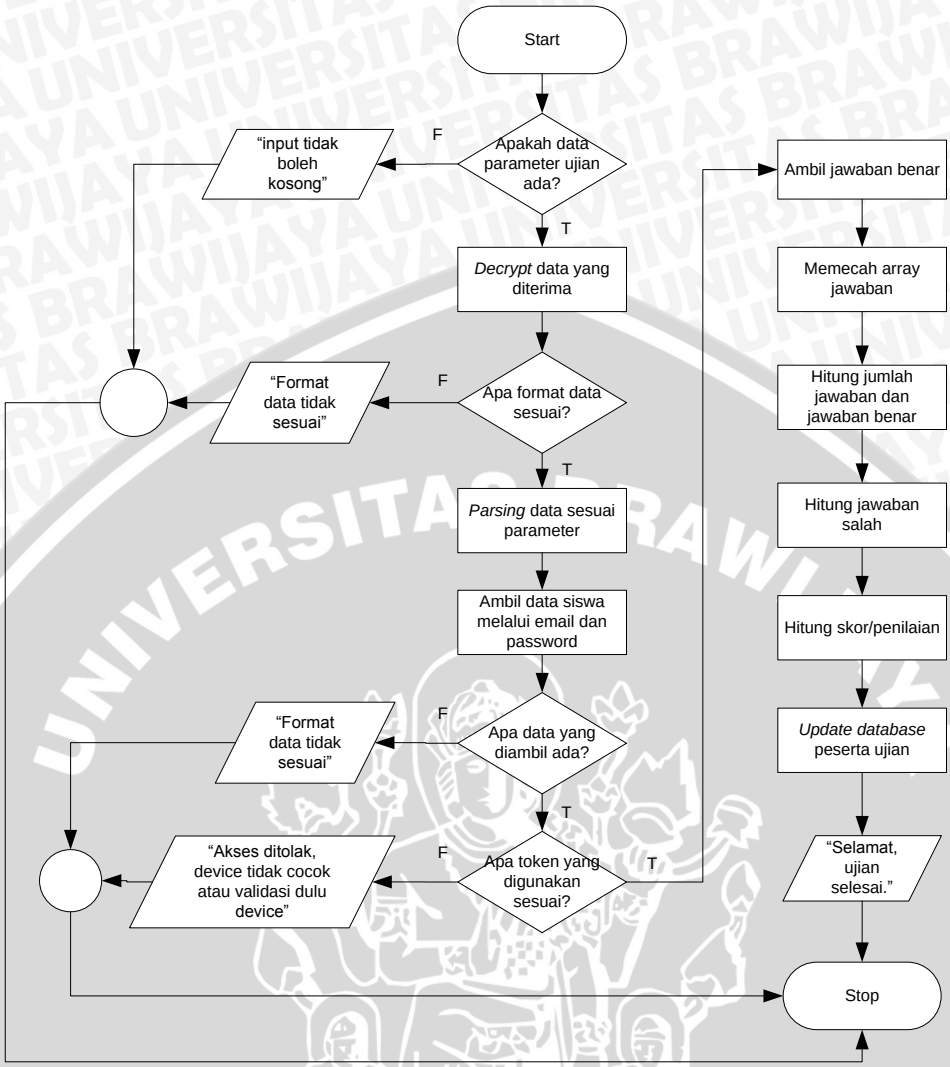
Selanjutnya sistem melakukan pengecekan validitas data peserta melalui data yang diambil sebelumnya. Ketersediaan data akan melanjukan proses berikutnya yaitu pengecekan token. Jika dalam *database* ujian *online* token peserta tidak sesuai, sistem akan menolak proses dan menampilkan respon bahwa *device* yang digunakan peserta tidak sesuai atau diharapkan melakukan validasi terlebih dahulu. Jika token sesuai, *database* akan meng-*update* token dan melanjutkan proses keamanan berikutnya yaitu pengecekan apakah sebelumnya peserta yang melakukan *login* telah melaksanakan ujian atau belum. Jika sistem mendeteksi bahwa telah tersimpan nilai dari peserta tersebut, maka dari aplikasi *client* peserta akan langsung menampilkan hasil dari ujian sebelumnya.



Gambar 4.9 Diagram Alir Pengamanan Proses Pengambilan Soal Ujian

Dalam Gambar 4.9 direpresentasikan diagram alir dari proses keamanan saat aplikasi peserta mengambil informasi soal ujian dari *database* ujian *online*. Seperti halnya proses validasi dan *login*, pada proses pengambilan soal diterapkan mekanisme keamanan serupa mulai pengambilan data parameter ujian peserta hingga pengambilan validitas data peserta melalui *email* dan *password*.

Proses yang berlangsung berikutnya adalah pengecekan validitas data peserta. Data peserta yang tidak valid akan membuat sistem ujian memberi respon pada peserta untuk melakukan *login* agar dapat masuk ke halaman penampilan soal ujian. Berikutnya adalah proses pengecekan kesesuaian token peserta terhadap *device* yang digunakan. Jika peserta mengerjakan ujian tidak melalui *device* yang telah divalidasi atas nama dirinya, sistem ujian menolak akses tersebut sehingga peserta harus mengganti dengan *device* yang sesuai atau melakukan validasi terhadap *device* yang digunakan.

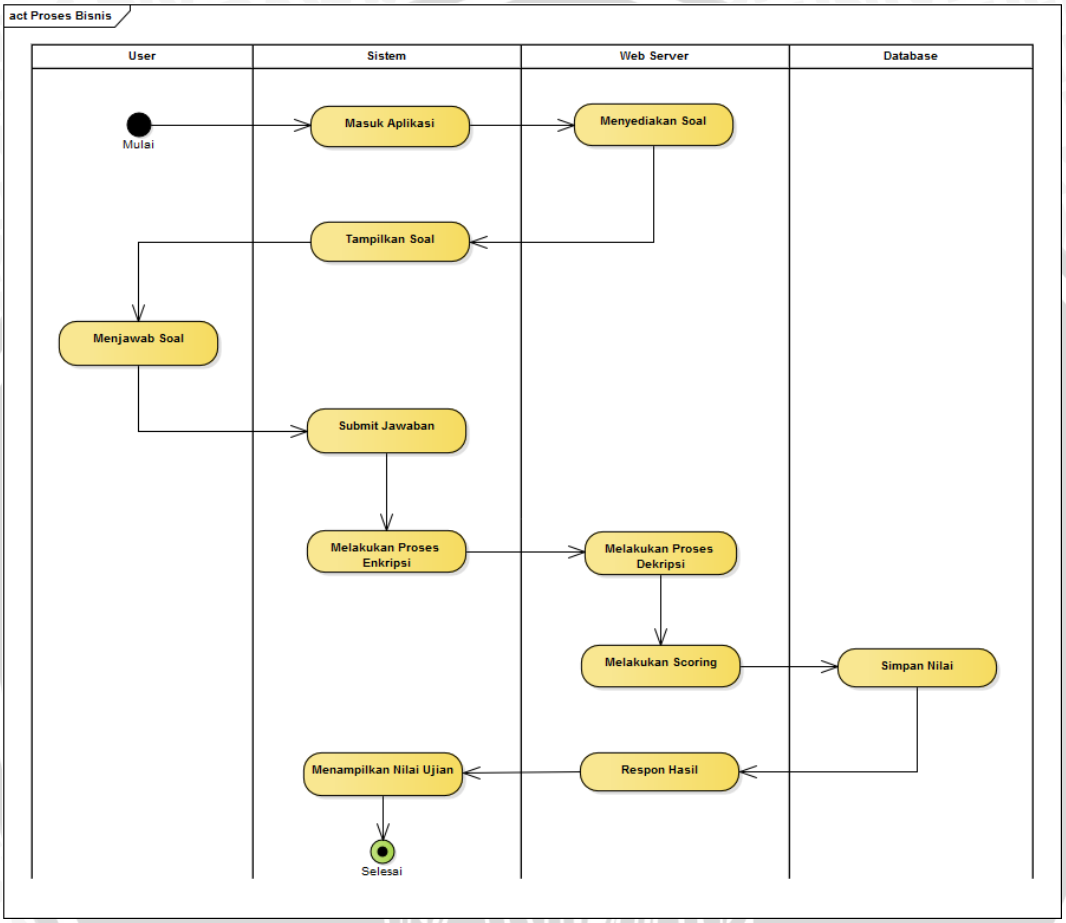


Gambar 4.10 Diagram Alir Pengamanan Proses Jawab Soal

Dalam Gambar 4.10 direpresentasikan diagram alir dari proses keamanan saat peserta melakukan penjawaban soal melalui aplikasi ujian *online* peserta. Mekanisme keamanan yang diterapkan pada proses penjawaban soal meliputi pengecekan data parameter ujian, pengecekan kesesuaian *format* data dengan susunan yang telah ditentukan oleh sistem keamanan ujian *online*, pengecekan validitas data peserta, juga pengecekan kesesuaian token dari *device* milik peserta. Jika seluruh rangkaian prosedur pengamanan terpenuhi, informasi jawaban yang dikirimkan peserta akan diolah untuk dilakukan penilaian dan menyimpan hasilnya ke dalam *database* ujian milik peserta serta memunculkan respon tampilan melalui aplikasi peserta berupa nilai ujian yang diperolehnya.

4.2.4 Perancangan Proses Bisnis Sistem

Perancangan proses bisnis sistem merupakan perancangan yang menunjukkan aktifitas/interaksi pengguna dengan sistem aplikasi yang berjalan. Untuk menggambarkan setiap tahapan proses yang dilalui pengguna dalam menjalankan aplikasi direpresentasikan dalam bentuk *activity diagram*. Pada *activity diagram* digambarkan alur kerja berupa langkah dan aksi dari kebutuhan dalam bentuk grafis.

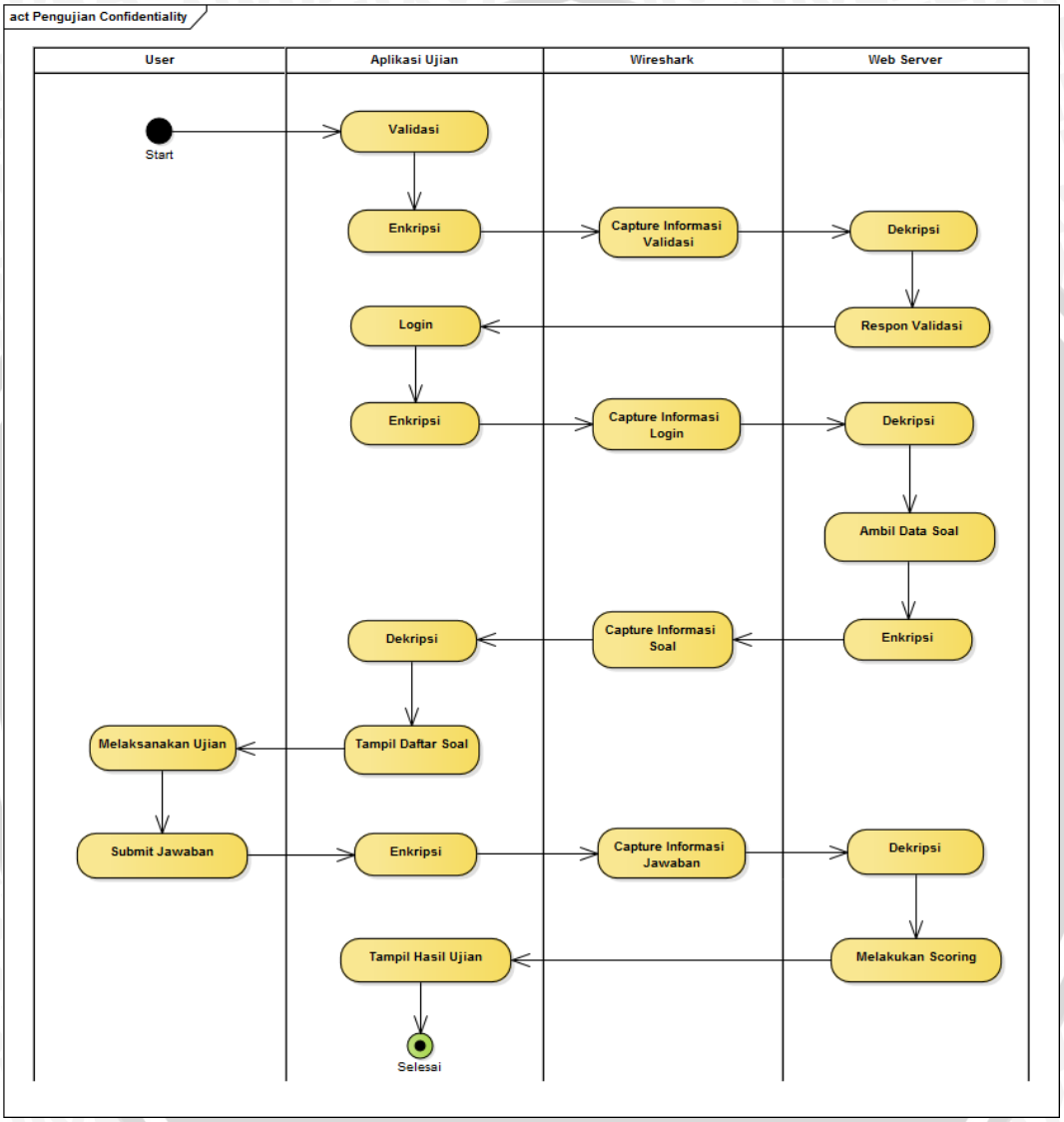


Gambar 4.11 *Activity Diagram* Proses Bisnis Sistem Ujian *Online*

Pada Gambar 4.11 menggambarkan aktifitas dari peserta ujian untuk melaksanakan ujiannya. Setelah peserta tersebut terdaftar beserta perangkat ujiannya di dalam *database* ujian *online* maka peserta dapat langsung masuk ke dalam aplikasi untuk menjawab soal yang disediakan *web server*. Hasil dari segala aktifitas yang berlangsung selama peserta menjalankan aplikasi akan dilakukan enkripsi dengan algoritme kriptografi Blowfish untuk dikirimkan menuju *web server* dan didekripsi kembali. Informasi dari *client* akan diolah dengan melakukan penilaian terhadap jawaban benar peserta. Selanjutnya hasil ujian peserta akan disimpan dalam *database* ujian *online* dan dari sisi *web server* akan memberi respon hasilnya terhadap aplikasi *client* peserta untuk ditampilkan.

4.2.5 Perancangan Prosedur Pengujian

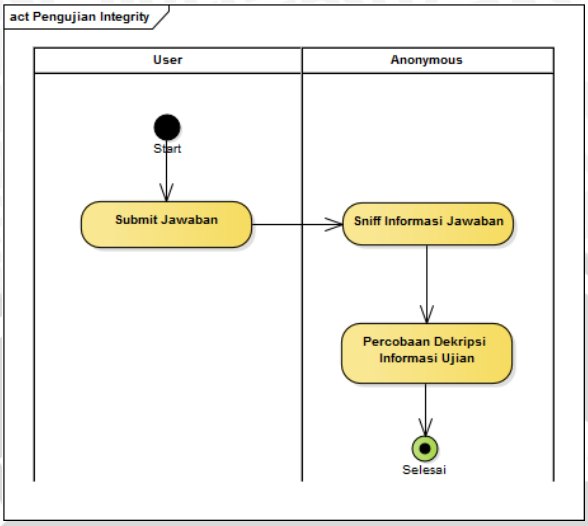
Perancangan prosedur pengujian menjelaskan pemodelan aktifitas yang dilakukan terhadap sistem keamanan aplikasi yang dibangun dalam upaya memenuhi aspek *security* dari kebutuhan non-fungsional sistem pada Tabel 4.4. Prosedur pengujian direpresentasikan dalam bentuk *activity diagram* untuk pengujian kerahasiaan dan integritas pada Gambar 4.12 sampai Gambar 4.14.



Gambar 4.12 *Activity Diagram* Prosedur Pengujian *Confidentiality*

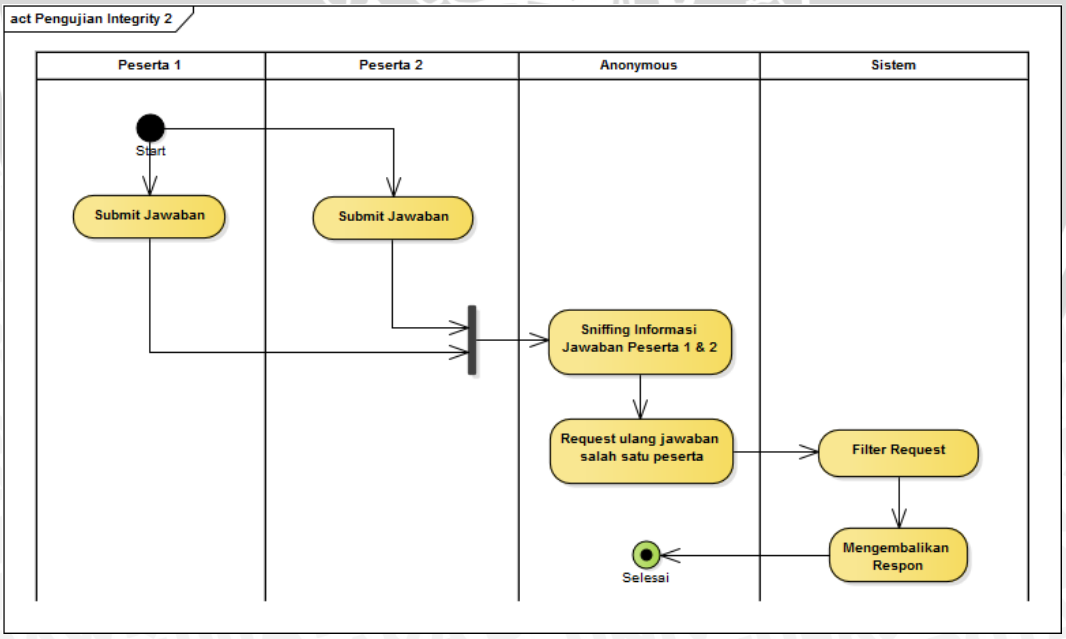
Skenario untuk pengujian *confidentiality* terkait kerahasiaan informasi ujian *online* peserta digambarkan dalam bentuk *activity* diagram pada Gambar 4.12. Untuk menguji kerahasiaan informasi ujian *online* digunakan aplikasi Wireshark yang mencoba melakukan *capture* terhadap segala informasi yang bertukar dari sisi aplikasi *client* peserta dengan *web server* ujian *online* maupun sebaliknya. Dengan adanya perlindungan terhadap informasi ujian yang diberikan oleh enkripsi dari algoritme kriptografi Blowfish diharapkan dapat merahasiakan isi informasi sehingga validitas perangkat peserta, identitas peserta, soal ujian, serta

jawaban ujian peserta, tidak dapat dimengerti isinya saat disadap menggunakan aplikasi Wireshark.



Gambar 4.13 *Activity Diagram* Prosedur Percobaan Dekripsi

Untuk pengujian selanjutnya yaitu pengujian integritas terhadap informasi ujian direpresentasikan dalam *activity diagram* pada Gambar 4.13. Dalam pengujian ini seorang *anonymous* sebagai pihak di luar sistem ujian *online* akan melakukan *sniffing* terhadap paket data yang terkirim selama ujian berlangsung dan melakukan dekripsi informasi ujian yang sebelumnya telah dilindungi oleh enkripsi dari algoritme kriptografi Blowfish.



Gambar 4.14 *Activity Diagram* Percobaan *Request* Ulang Informasi Ujian

Percobaan untuk menguji integritas informasi ujian berikutnya direpresentasikan oleh Gambar 4.14. Setelah pengujian integritas pada Gambar

4.13 tidak mendapatkan hasil, selanjutnya percobaan dilakukan setelah peserta ujian menjawab soal dan mengirimkannya menuju *web server* ujian. Dalam kasus ini jawaban ujian yang dikirimkan peserta ujian akan dilakukan pengujian *sniffing* terlebih dahulu untuk didapatkan informasinya. Selanjutnya *anonymous* akan melakukan *request* terhadap informasi milik salah satu peserta yang diperoleh dari hasil *sniffing* untuk dikirimkan menuju sistem ujian *online* dengan maksud ingin menguji integritas dari upaya manipulasi nilai yang telah tersimpan dalam *database* sehingga dapat diketahui apakah ada pengubahan dari nilai sebelumnya dengan nilai baru yang telah di-*request*. Dalam hal ini sistem keamanan ujian *online* melakukan pengecekan terhadap upaya manipulasi melalui *request* yang dikirimkan *anonymous* sehingga muncul respon yang menunjukkan hasil dari seleksi keamanan ujian *online*.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dibahas mengenai tahapan implementasi sistem keamanan yang berguna untuk menjaga kerahasiaan dan keutuhan informasi jawaban ujian *online* melalui aplikasi perangkat bergerak yang memanfaatkan algoritme kriptografi Blowfish pada Android berdasarkan hasil dan analisis pada perancangan sistem. Implementasi pada pembahasan ini terbagi menjadi beberapa bagian yaitu penjelasan tentang spesifikasi perangkat keras dan perangkat lunak, implementasi algoritme dan *source code*, dan implementasi basis data. Selanjutnya dilakukan pengujian sistem keamanan berdasarkan kebutuhan non-fungsional yang telah dibahas sebelumnya

5.1 Spesifikasi Perangkat Keras dan Perangkat Lunak

Dalam pengembangan sistem keamanan ini pada perangkat *mobile* berbasis Android menggunakan sebuah komputer dengan *Operating System Windows* 8 dan perangkat lunak yang digunakan untuk membangun sistem tersebut yaitu Eclipse Juno, ADT (*Android Developer Tools*) serta untuk uji coba sistem pada perangkat *mobile* berbasis Android ini menggunakan *emulator* Android dengan *Operating System* Android 4.4.2 (KitKat). Pada perangkat lunak yang digunakan untuk mengamankan hasil nilai ujian *online* dari peserta menggunakan algoritme kriptografi Blowfish memanfaatkan *package* dari JCA (*Java Cryptography Architecture*) yang dimana JCA merupakan sekumpulan API yang disediakan oleh Google. Selanjutnya untuk uji coba terhadap keamanan menggunakan aplikasi *sniffer* Wireshark, fasilitas dekripsi informasi secara *online* melalui halaman *web browser* Google Chrome, dan aplikasi HttpRequester.

5.2 Implementasi Basis Data

Implementasi basis data dilakukan menggunakan *Database Management System* MySQL. Terdapat 3 tabel yang diperlukan untuk menyimpan data dalam aplikasi ujian *online*. Berikut merupakan gambar dan spesifikasi tabel-tabel pada *database* yang diimplementasikan.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
☐	1	**username** 🔑	varchar(20)	latin1_swedish_ci		No	*None*	
☐	2	**password**	varchar(32)	latin1_swedish_ci		No	*None*	

Gambar 5.1 Tabel Administrator

Gambar 5.1 menjelaskan kolom-kolom yang ada pada tabel administrator. Tabel administrator berisikan *username* dan *password* yang dimiliki oleh admin untuk melakukan manajemen kontrol terhadap sistem ujian *online*.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	soal	text	latin1_swedish_ci		No	None	
3	jawabanA	text	latin1_swedish_ci		No	None	
4	jawabanB	text	latin1_swedish_ci		No	None	
5	jawabanC	text	latin1_swedish_ci		No	None	
6	jawabanD	text	latin1_swedish_ci		No	None	
7	jawabanBenar	text	latin1_swedish_ci		No	None	

Gambar 5.2 Tabel Soal Ujian

Gambar 5.2 menjelaskan kolom-kolom yang ada pada tabel soal ujian. Tabel ini digunakan untuk menyimpan semua soal beserta pilihan jawabannya yang akan disajikan pada aplikasi Android peserta untuk dikerjakan selama pelaksanaan ujian *online* berlangsung.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	nama	text	latin1_swedish_ci		No	None	
3	email	varchar(255)	latin1_swedish_ci		No	None	
4	password	varchar(32)	latin1_swedish_ci		No	None	
5	nilai	int(11)			No	None	
6	token	varchar(255)	latin1_swedish_ci		No	-	
7	active_session	varchar(255)	latin1_swedish_ci		No	-	
8	status_ujian	varchar(10)	latin1_swedish_ci		No	BELUM	

Gambar 5.3 Tabel Siswa

Gambar 5.3 menjelaskan kolom-kolom yang ada pada Tabel Siswa. Pada tabel ini berisikan segala informasi penting yang dikirimkan dari aplikasi ujian *online* peserta.

5.3 Implementasi Kode Program

Pada implementasi sistem keamanan aplikasi *client* ujian *online* dengan algoritme kriptografi Blowfish terdapat beberapa proses yang dilakukan. Pada penulisan laporan skripsi ini hanya dicantumkan beberapa proses utama saja.

```
1.  import javax.crypto.Cipher;
2.  import javax.crypto.spec.SecretKeySpec;
3.
4.  public static String enkripsi(String dataAsli, String key) {
5.          try {
6.                  SecretKeySpec keySpec = new
7.                    SecretKeySpec(key.getBytes(), "Blowfish");
```

```
8.                  Cipher cipher = Cipher.getInstance("Blowfish");
9.                  cipher.init(Cipher.ENCRYPT_MODE, keySpec);
10.                 byte[] encrypted = cipher.doFinal(dataAsli.getBytes());
11.             return Base64.encodeToString(encrypted, Base64.NO_PADDING);
12.          }
13.         catch (Exception e) {
14.             return "ERROR:" + e.getMessage();
15.             }
16. }
```

Kode 5.1 Implementasi Kode Program Algoritme Enkripsi Blowfish

Barisan kode seperti yang ditunjukkan pada Gambar 5.1 di atas adalah penerapan dari proses enkripsi algoritme kriptografi Blowfish dari aplikasi ujian *online* peserta pada perangkat Android untuk mengamankan informasi ujian yang dikirimkan menuju *web server* ujian *online*. Penjelasan implementasi kode program 5.1 yaitu:

1. Baris 1-2, merupakan *library* kriptografi yang digunakan untuk melakukan proses enkripsi dan dekripsi terhadap informasi ujian.
2. Baris 4, merupakan nama fungsi untuk melakukan enkripsi terhadap data peserta dalam bentuk *string*.
3. Baris 6, merupakan ekspansi dari *secret key* baru yang akan dibuat.
4. Baris 7-8, merupakan penerapan *cipher* yang digunakan yaitu algoritme Blowfish.
5. Baris 10, merupakan enkripsi terhadap data peserta untuk setiap karakternya
6. Baris 11, merupakan konversi data enkripsi menjadi bentuk *string* pada Base64.

```
1.  import javax.crypto.Cipher;
2.  import javax.crypto.spec.SecretKeySpec;
3.
4.  public static String dekripsi(String dataTerenkripsi, String key) {
5.          try {
6.              byte[] encryptedData = Base64.decode(dataTerenkripsi, Base64
7.  .DEFAULT);
8.
9.              SecretKeySpec keySpec = new SecretKeySpec(key.getBytes(),
                  "Blowfish");
10.             Cipher cipher = Cipher.getInstance("Blowfish/ECB/NoPadding")
    ;
11.             cipher.init(Cipher.DECRYPT_MODE, keySpec);
12.             byte[] decrypted = cipher.doFinal(encryptedData);
13.             return new String(decrypted);
14.         } catch (Exception e) {
15.             e.printStackTrace();
16.             return "ERROR:" + e.getMessage();
17.         }
18.     }
```

Kode 5.2 Implementasi Kode Program Algoritme Dekripsi Blowfish

Pada Kode 5.2 dijelaskan barisan kode untuk melakukan proses dekripsi terhadap *ciphertext* informasi ujian setelah dilaksanakan ujian melalui perangkat *mobile* peserta ujian. Berikut penjelasan kode program pada Kode 5.2:

1. Baris 1-2, merupakan *library* kriptografi yang digunakan untuk melakukan proses enkripsi dan dekripsi terhadap informasi ujian.
2. Baris 4, merupakan nama fungsi untuk melakukan dekripsi terhadap *ciphertext* informasi ujian.
3. Baris 10, merupakan penerapan algoritme, mode *cipher*, dan *padding* yang digunakan untuk dekripsi.
4. Baris 11, merupakan mode dekripsi pada fungsi "String dekripsi".
5. Baris 12, merupakan proses dekripsi terhadap tiap karakter *ciphertext*.
6. Baris 13, merupakan pengembalian bentuk dekripsi ke bentuk *string*.

```
1.  import android.content.Context;
2.  import android.net.wifi.WifiInfo;
3.  import android.net.wifi.WifiManager;
4.  import android.provider.Settings.Secure;
5.  import android.util.Base64;
6.
7.          public static String token_string(String token) {
8.          try {
9.              return Base64.encodeToString(token.getBytes(),
    Base64.DEFAULT);
10.         } catch (Exception e) {
11.             return "ERROR:" + e.getMessage();
12.         }
13.     }
14.
15.     public static String getToken(Context c) {
16.         WifiManager wifiManager = (WifiManager) c.getSystemService
    (Context.WIFI_SERVICE);
17.         WifiInfo wInfo = wifiManager.getConnectionInfo();
18.         String mac = wInfo.getMacAddress();
19.         return mac;
20.
21.     }
```

Kode 5.3 Kode Program Fungsi Mendapatkan MAC Address

Pada Kode 5.3 merupakan kode program untuk melakukan pengambilan *MAC Address* dari *device* Android peserta ujian agar dapat divalidasi oleh sistem keamanan ujian *online* dan memperoleh akses untuk mengerjakan ujian. Penjelasan kode program pada Kode 5.3, yaitu:

1. Baris 9, merupakan proses pengubahan tipe informasi ujian ke dalam mode Base64.
2. Baris 15, merupakan *device service* untuk mengetahui informasi *device*.
3. Baris 17, merupakan pengambilan informasi *device* yang tersambung dalam sebuah koneksi jaringan.
4. Baris 18, merupakan pengambilan informasi *string* dari *MAC address*.

Selanjutnya dari sisi *server* juga diberikan mekanisme sistem keamanan disamping adanya proses enkripsi dan dekripsi informasi ujian menggunakan algoritme kriptografi Blowfish. Prosedur keamanan yang diterapkan sebagian besar memiliki kemiripan untuk tiap-tiap proses, sehingga pada laporan ini akan dijabarkan kode program dari proses validasi dan proses jawab soal saja.

```php
<?php

require_once "koneksi.php";

define ('PRIVATE_KEY' , 'ujianonlineblowfish' );
define ('DELIMITER' , '#' );
define ('AGENT' , 'Apache-HttpClient' );

$conn =  mysql_connect($servername, $username, $password );
$db =  mysql_select_db( $dbname);

function doEncrypt($msg){
  $iv  = mcrypt_create_iv(mcrypt_get_iv_size(constant('MCRYPT_BLOWFISH'), MCRYPT_MODE_ECB), MCRYPT_RAND);
  $passcrypt  = mcrypt_encrypt(constant('MCRYPT_BLOWFISH'), PRIVATE_KEY, $msg, MCRYPT_MODE_ECB, $iv);

  $encode = base64_encode($passcrypt);
  return $encode;
}

function doDecrypt($msg){
  $decoded = base64_decode($msg);
  $iv = mcrypt_create_iv(mcrypt_get_iv_size(constant('MCRYPT_BLOWFISH'), MCRYPT_MODE_ECB), MCRYPT_RAND);
  $decrypted  = mcrypt_decrypt(constant('MCRYPT_BLOWFISH'),  PRIVATE_KEY, $decoded, MCRYPT_MODE_ECB, $iv);

  return $decrypted;
}

function sendResponse($strMsg) {
    exit(doEncrypt($strMsg.DELIMITER.date("'d m y his").DELIMITER ));
    return;
}

function doValidasi(){
    if (isset($_REQUEST['data'])  ) {
        //melakukan dekripsi data yand diterima
        $data = doDecrypt($_REQUEST['data']);

        if (strpos($data,DELIMITER) !== false ){
            // parsing/memotong data sesuai tipe berdasarkan delimiter
            $dataValid = explode( DELIMITER,$data);
            $email = htmlspecialchars($dataValid[0]);
            $pass = htmlspecialchars($dataValid[1]);
            $token = htmlspecialchars($dataValid[2]);
             //query cek data user valid
            $sql = "SELECT * FROM siswa WHERE email='".$email."' AND password='".$pass."'";
            if ($result = mysql_query( $sql)) {

                if ($obj = mysql_fetch_object($result)) {
                    //cek apakah token sudah diset
                    if ($obj->token == "-" || trim($obj->token) == "") {

                        //cek jika token sudah ada pada user lain
                        $sqlCek = "SELECT * FROM  WHERE token='".$token."' ";
                        if (mysql_num_rows(mysql_query($sqlCek))) {
                            sendResponse ("False".DELIMITER."Maaf device sudah digunakan, silahkan login dengan usernama dan password anda..");
```

```
56.                         }else {
57.                             //proses update token baru sesuai data user

58.                             $sql = "UPDATE siswa SET token='".$token."'
    where  email='".$email."' AND password='".$pass."' ";
59.                             $qry = mysql_query( $sql);
60.                             sendResponse ("True".DELIMITER."Proses valid
    asi sukses, silahkan login..");
61.                         }
62.                     }else {
63.                         sendResponse ("False".DELIMITER."Maaf proses gag
    al, validasi device hanya diijikan sekali saja.");
64.                     }
65.
66.                 } else {
67.                     sendResponse ("False".DELIMITER."Login gagal.");
68.                 }
69.             } else {
70.                 sendResponse ("False".DELIMITER."Login gagal");
71.             }
72.         }else {
73.             sendResponse ("False".DELIMITER."Format data tidak sesuai");

74.         }
75.
76.
77.     } else {
78.             sendResponse ("False".DELIMITER."Inputan Tidak Boleh Kosong"
    );
79.     }
80. }
81.
```

Kode 5.4 Implementasi Kode Program API *Service* dari Proses Validasi

Kode 5.4 merupakan barisan kode API *Service* yang dimiliki sistem keamanan ujian *online* yang memanfaatkan algoritme kriptografi Blowfish beserta rangkaian prosedur keamanan dari proses validasi yang diberikan di dalamnya. Berikut beberapa penjelasan mengenai potongan kode program pada Kode 5.4:

1. Baris 3, merupakan *statement* untuk menjalankan file "koneksi.php" sebagai *requirement* menjalankan API *Service* ini.
2. Baris 5-7, merupakan inisialisasi konstan atau pendefinisian variabel secara *default* sesuai kebutuhan sistem keamanan ujian *online*.
3. Baris 12-18, merupakan *method* enkripsi dari algoritme Blowfish pada sisi *server* yang melindungi informasi ujian saat dikirimkan menuju peserta.
4. Baris 20-26, merupakan *method* dekripsi dari algoritme Blowfish untuk membuka *ciphertext* dari informasi ujian peserta.
5. Baris 34, merupakan pengecekan apakah data yang dikirimkan peserta lengkap atau tidak.
6. Baris 38, merupakan pengecekan apakah format telah sesuai atau tidak.
7. Baris 40-43, merupakan penguraian informasi sesuai tipe berdasarkan parameter.
8. Baris 45-46, merupakan pengecekan *query* dari informasi milik peserta ujian
9. Baris 48, merupakan pengecekan apakah data peserta tersedia atau tidak.

10. Baris 50, merupakan pengecekan token apakah sudah divalidasi atau belum.
11. Baris 54, merupakan pengecekan apakah token sudah dimiliki oleh peserta lain atau belum.
12. Baris 58, merupakan proses *update* token baru sesuai dengan data peserta.
13. Baris 62-80, merupakan pengecualian jika *request* yang dilakukan oleh peserta ujian tidak valid. Pemberitahuan kesalahan berupa respon yang terjadi akan dikirimkan kembali pada aplikasi peserta.

```
1.  function doJawabSoal(){
2.      if (isset($_REQUEST['data'])  ) {
3.
4.          $data = doDecrypt($_REQUEST['data']);
5.
6.          if (strpos($data,DELIMITER) !== false ){
7.              $dataValid = explode( DELIMITER,$data);
8.              $email = htmlspecialchars($dataValid[0]);
9.              $pass = htmlspecialchars($dataValid[1]);
10.             $jawaban = htmlspecialchars($dataValid[2]);
11.             $token = htmlspecialchars($dataValid[3]);
12.
13.             $sql = "SELECT * FROM siswa WHERE email='".$email."' AND pas
    sword='".$pass."'";
14.             if ($result = mysql_query( $sql)) {
15.
16.                 if ($obj = mysql_fetch_object($result)) {
17.
18.                     if ($obj->token == $token) {
19.                         $sql = "SELECT `id`, `soal`, `jawabanA`, `jawaba
    nB`, `jawabanC`, `jawabanD`, `jawabanBenar` FROM `soal`   order by `id`
    asc ";
20.
21.                         $respon = "";
22.                         $jawabSoal = null;
23.                         if ($result = mysql_query( $sql)) {
24.                             while ($objx =mysql_fetch_object($result)) {

25.                                 // list jawaban benar
26.                                 $jawabSoal[strval($objx->id)] = $objx-
    >jawabanBenar;
27.                             }
28.                         }
29.
30.                         $dataJawab = split("@",$jawaban);
31.                         $arrJawab = null;
32.                         for ($i=0;$i<count($dataJawab);$i++){
33.                             if (trim($dataJawab[$i])=="") continue;
34.                             $dataJawab[$i] = str_replace("|",":",$dataJa
    wab[$i]);
35.                             $tmp = split(":",$dataJawab[$i]);
36.                             $arrJawab[strval($tmp[0])] = $tmp[1];
37.                         }
38.                         $jawabBenar = 0;
39.                         // cek apakah jawaban benar
40.                         foreach ($arrJawab as $key => $value) {
41.                              if ($jawabSoal[$key]==$value) {
42.                                 // jika benar nilai +1
43.                                  $jawabBenar++;
44.                              }
45.                         }
46.                         //hitung jawab salah
```

```
47.                         $salah = count($dataJawab) - $jawabBenar;
48.                         // nilai
49.                         $nilai = $jawabBenar * 10;
50.                         //respon hasil ke android
51.                         $respon ="Selamat ".ucfirst($obj-
    >nama)." Ujian selesai, hasil anda :<br><br>Benar : ".$jawabBenar."<br>S
    alah : ".$salah."<br><br>Nilai Anda : ".$nilai;
52.
53.                         //update nilai user
54.                         $sql = "UPDATE siswa SET nilai='".$nilai."', sta
    tus_ujian='SUDAH' where  email='".$email."' AND password='".$pass."' ";

55.                         $qry = mysql_query( $sql);
56.                         // kirim respon ke user
57.                         if ($respon!="") {
58.                             sendResponse ("True".DELIMITER.$respon);
59.                         }else {
60.                             sendResponse ("False".DELIMITER."Maaf data s
    oal tidak ditemukan.");
61.                         }
62.
63.                     }else {
64.                         sendResponse ("False".DELIMITER."Maaf akses dito
    lak, device tidak cocok atau validasi dulu device anda.");
65.                     }
66.
67.                 } else {
68.                     sendResponse ("False".DELIMITER."Silahkan login terl
    ebih dahulu.");
69.                 }
70.             } else {
71.                 sendResponse ("False".DELIMITER."Silahkan login terlebih
    dahulu");
72.             }
73.         }else {
74.             sendResponse ("False".DELIMITER."Format data tidak sesuai");

75.         }
76.
77.
78.     } else {
79.             sendResponse ("False".DELIMITER."Inputan tidak boleh kosong"
    );
80.     }
81. }
```

Kode 5.5 Implementasi Kode Program API *Service* dari Proses Jawab Soal

Kode 5.5 merupakan barisan kode program dari API *Service* untuk proses menjawab soal ujian. Proses ini akan dimulai ketika peserta melakukan *request* untuk melakukan *submit* terhadap soal ujian yang telah dikerjakan. Berikut beberapa penjelasan mengenai kode program pada Kode 5.5:

1. Baris 2, merupakan pengecekan apakah data yang dikirimkan peserta lengkap atau tidak.
2. Baris 6, merupakan pengecekan apakah format data sesuai atau tidak.
3. Baris 8-11, merupakan pemecahan/penguraian informasi berdasarkan parameter.

4. Baris 13, merupakan pengambilan data siswa melalui email dan password.
5. Baris 16, merupakan pengecekan apakah data tersedia atau tidak.
6. Baris 18, merupakan pengecekan apakah token sesuai dengan token peserta.
7. Baris 19, merupakan pengambilan data soal dari *database*.
8. Baris 23, merupakan *fetch* hasil *database* menjadi objek.
9. Baris 26, merupakan penyimpanan data jawaban benar ke data jawab soal dengan indeks yang sesuai.
10. Baris 30, memisah *array* dengan *regular expression*.
11. Baris 32-37, merupakan penghitungan soal yang telah dijawab.
12. Baris 40, merupakan penghitungan jawaban benar.
13. Baris 47, merupakan penghitungan jawaban salah.
14. Baris 48, merupakan penghitungan nilai terhadap jawaban benar.
15. Baris 51, merupakan pengembalian respon hasil ujian ke aplikasi Android peserta.
16. Baris 53, merupakan penyimpanan hasil ujian peserta ke dalam *database.*
17. Baris 56-80, merupakan pengembalian respon yang dikembalikan ke peserta jika *request* yang diterima sistem tidak memenuhi persyaratan yang ditentukan.

5.4 Pengujian dan Analisis

Pada tahap ini akan dilakukan proses pengujian terhadap sistem keamanan aplikasi yang telah dibangun. Pengujian ditujukan untuk menjawab permasalahan yang telah dirumuskan dalam bab pendahuluan yaitu pengujian kerahasiaan serta pengujian integritas terhadap data informasi ujian *online* yang memanfaatkan algoritme kriptografi Blowfish dalam sistem operasi Android.

5.4.1 Pengujian Kerahasiaan (*Confidentiality*)

Pengujian kerahasiaan dalam sistem keamanan aplikasi ujian *online* pada sistem operasi Android dilakukan 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.

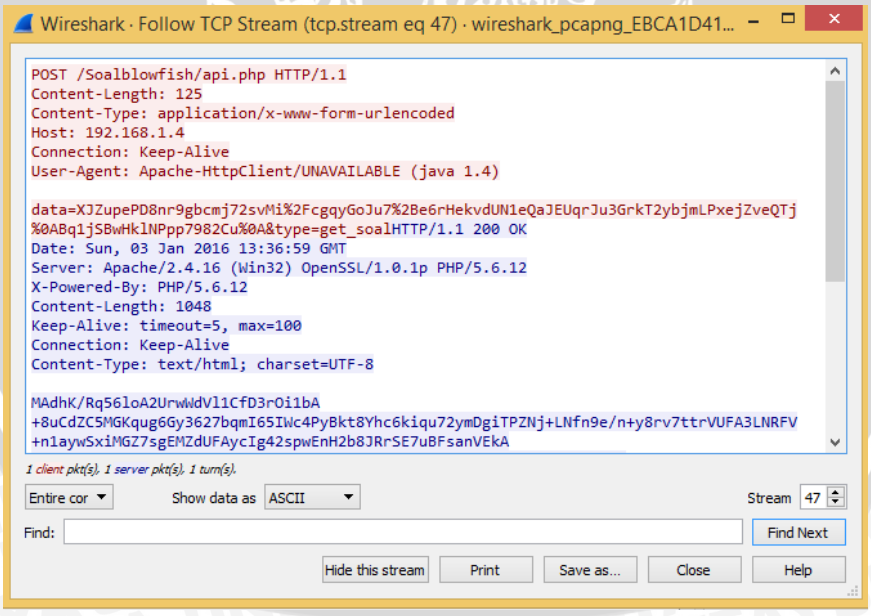
5.4.1.1 Skenario Pengujian Kerahasiaan

Skenario pengujian kerahasiaan (*confidentiality*) terhadap informasi ujian *online* dilakukan setelah melalui beberapa proses pelaksanaan ujian. Proses awal yang dijalankan bermula dimana peserta harus melakukan validasi terhadap *device* yang digunakan untuk ujian terlebih dahulu dengan mengisi *email* dan *password* agar dapat tersimpan informasinya di dalam *database*. Setelah sistem keamanan ujian *online* menerima validitas tersebut maka peserta dapat melaksanakan ujian dengan proses *login* sesuai masukan yang sama dengan

validitas *device*. Selanjutnya aplikasi ujian akan mengambil informasi ujian dalam *database* untuk dijawab oleh peserta. menjawab sepuluh soal yang disajikan dalam sekali akses untuk proses pengiriman kembali pada *database* dan peserta akan dapat melihat hasil ujiannya setelah dilakukan penghitungan jawaban benar oleh *web server*.

Dalam aplikasi maupun web ujian secara otomatis sistem akan melakukan proses enkripsi terhadap data yang dikirimkan aplikasi *client* ujian *online* dan *web server* ujian dengan memanfaatkan algoritme kriptografi Blowfish sehingga dapat diketahui bentuk perlindungan fisik yang diberikan oleh sistem keamanan ujian *online* ini.

Untuk membuktikannya akan dilakukan percobaan menggunakan *tools* Wireshark sebagai *sniffer* untuk melakukan *capture* terhadap lalu lintas paket data yang terjadi dalam jaringan selama aktifitas ujian *online* berlangsung. Filterasi paket data dilakukan pada alamat IP dari *server* agar diperoleh informasi apa saja yang diterima dari aplikasi *client* ujian *online*. Setelah siswa mengirimkan data hasil ujiannya ke dalam *database server*, aplikasi akan melakukan pengiriman informasi berupa parameter "data" berisikan *username, password,* kode soal, jawaban, beserta *session* dari peserta ujian yang dijadikan satu informasi untuk dienkripsi dengan algoritme kriptografi Blowfish. Berdasarkan *Output* yang dihasilkan pada percobaan tersebut diperoleh hasil *sniffing* bahwa informasi nilai dalam parameter *data* berhasil diamankan kerahasiaan bentuk aslinya dengan memanfaatkan enkripsi algoritme kriptografi Blowfish.



Gambar 5.4 Hasil Capture Salah Satu Lalu Lintas Paket Data

5.4.1.2 Hasil Pengujian Kerahasiaan

Berdasarkan hasil dari percobaan *sniffing* terhadap lalu lintas untuk melihat informasi ujian *online* yang diperoleh dari 4 orang peserta dalam pelaksanaan ujian *online* dari *device* yang berbeda-beda, dilakukan pengumpulan hasil *capture* paket data di dalam jaringan. Dengan mengetahui hasil dari penyadapan tersebut maka diperolehlah data informasi ujian dalam bentuk *ciphertext* sehingga tidak dapat dimengerti maksud dari barisan kode tersebut oleh *sniffer*. Berikut adalah informasi yang ditangkap oleh Wireshark saat terjadi pertukaran paket data antara aplikasi *client* peserta dengan *server* sistem ujian *online* maupun sebaliknya seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Hasil Penyadapan Informasi Proses Validasi Peserta

No	Peserta	Hasil *Capture* Paket Data Peserta	Status
1.	abc	data=DFXBQYY6%2FjOHhKAetU286fUegXW0%2FvhdrTUZ7iKpX8r3pSKeQVV1Z2OQgM1GqBrO%2BPpcZOb8UPmR%0AiPFrDn46w6BWYp%2B8t78R%0A&type=validasiHTTP/1.1 200 OK Date: Wed, 30 Mar 2016 04:36:08 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 255 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid
2.	udin	data=HmtSEyJCHrDG3HxItmLREHI9GpH6cKWj7%2Be6rHekvdUN1eQaJEUqrFTUChCtiN78IGGIErTWekRf%0AQudqVV4mimzDEPq2tuUm%0A&type=validasiHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:39:04 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 255 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid
3.	putra	data=gVRPicVVzwsT47oZ7WOKIIxYfI9d9WML4mqTQZaD8pFc7dzPCqXeHyEP5YBFzi7uXUCK5CKBusVO%0AxpJtxROSJz3KZIwTEp3r%0A&type=validasiHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:56:00 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 255 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid
4.	hafizh	data=g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgQyhe7230VwUSaGF2Eq6AKPkOwCbH6gBOlwJV%2BsnNwOHa%0AgVfqCKh9uQwY8zA3t0ww%0A&type=validasiHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 18:03:05 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 255 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid

Tabel 5.2 Hasil Penyadapan Informasi Proses Login Peserta

No	Peserta	Hasil *Capture* Paket Data Peserta	Status
1.	abc	data=DFXBQYY6%2FjOHhKAetU286fUegXW0%2FvhdrTUZ7iKpX8r3pSKeQVV1Z2OQgM1GqBrOyemYpWbdySuR%0AiPFrDn46w6BWYp%2B8t78R%0A&type=loginHTTP/1.1 200 OK Date: Wed, 30 Mar 2016 04:36:33 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 88 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid

2.	udin	data=HmtSEyJCHrDG3HxItmLREHI9GpH6cKWj7%2Be6rHekvdUN1eQaJEUqrFTUChCtiN7841S3qrHsMQ1f%0AQudqVV4mimzDEPq2tuUm%0A&type=loginHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:39:29 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 88 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid
3.	putra	data=gVRPicVVzwsT47oZ7WOKIIxYfI9d9WML4mqTQZaD8pFc7dzPCqXeHyEP5YBFzi7uZXmBCtrJza9g%0AYAYu6XeVwWSg78s67BQW%0A&type=loginHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:57:24 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 88 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid
4.	hafizh	data=g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgQyhe7230VwUSaGF2Eq6AKPkOwCbH6gBOGCPI7f8vREPa%0AgVfqCKh9uQwY8zA3t0ww%0A&type=loginHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 18:03:35 GMT Server: Apache/2.4.16 (Win32) OpenSSL/1.0.1p PHP/5.6.12 X-Powered-By: PHP/5.6.12 Content-Length: 96 Keep-Alive: timeout=5, max=100 Connection: Keep-Alive Content-Type: text/html; charset=UTF-8	Valid

Tabel 5.3 Hasil Penyadapan Informasi Proses Pengambilan Soal

No	Peserta	Hasil *Capture* Paket Data *Server*	Status
1.	abc	data=DFXBQYY6%2FjOHhKAetU286fUegXW0%2FvhdrTUZ7iKpX8r3pSKeQVV1Z2OQgM1GqBrOf1lsGLkVx32R%0AiPFrDn46w6BWYp%2B8t78R%0A&type=get_soalHTTP/1.1 200 OK Date: Wed, 30 Mar 2016 04:36:35 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	Valid
2.	udin	data=HmtSEyJCHrDG3HxItmLREHI9GpH6cKWj7%2Be6rHekvdUN1eQaJEUqrFTUChCtiN780POznZVa6fpf%0AQudqVV4mimzDEPq2tuUm%0A&type=get_soalHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:39:31 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	Valid
3.	putra	data=gVRPicVVzwsT47oZ7WOKIIxYfI9d9WML4mqTQZaD8pFc7dzPCqXeHyEP5YBFzi7uttSWBYESR4xg%0AYAYu6XeVwWSg78s67BQW%0A&type=get_soalHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:57:27 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	Valid
4.	hafizh	data=g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgQyhe7230VwUSaGF2Eq6AKPkOwCbH6gBOwvuMVUeKLRe6%0A5spd8iqRmAwY8zA3t0ww%0A&type=get_soalHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 18:03:49 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	Valid

Tabel 5.4 Hasil Penyadapan Informasi Proses Pengiriman Jawaban

No	Peserta	Hasil *Capture* Paket Data Peserta	Status
1.	abc	data=DFXBQYY6%2FjOHhKAetU286eJzINLF5LUoQU92t49xcj4RGPDKXrLYiAbTrk1mo4ZNlw93MDtQOCPR %0Anm7qnwVcW02VSls4Fmz6Bf4pD5g7a90xU1tSPVJd3P6BrTUr21%2Fa0xagwphm7HJTT6ae%2FfNgrg %0A&type=jawabHTTP/1.1 200 OK Date: Wed, 30 Mar 2016 04:37:01 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	Valid
2.	udin	data=HmtSEyJCHrDG3HxItmLREPqiR5Yblq06WYuuOkxftgrnvK4I5U2OJZMfooEyfK54okK3FkPXH %2BD7%0AvqGX0Kc7SFQW6Cj%2FAodbn9qaz7i6gOjuarMvp %2FxZrt7L8fMGeguQTsaSbcUTkic9ymSMExKd6w%0A&type=jawabHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:40:08 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	Valid
3.	Putra	data=gVRPicVVzwsT47oZ7WOKICiQ73u0W2sy5lFD4otlYjlN%2FvbsCo1IftQBr3saTSXV8eyph%2F2xi %2F1e1rnqz3xvmx94jncBuRAvgst1auubKXfIqkZgMGPMwN7dMMA%0A&type=jawabHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 17:58:10 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	Valid
4.	hafizh	data=g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgsO0pj%2Bq63EpnlBPyol9zNdBtKd1uyfEJxqDjhNYA7ycO%0A %2BeftF5BtShNfMpuM60O2WL4vJi7JSKO8KKYqHwEJOarJQGePd5A9s8%2FKxLVDNVPN%2BLdC6LRsdg %0A&type=jawabHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 18:04:08 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	Valid

Berdasarkan hasil yang diperoleh dari Tabel 5.1 hingga Tabel 5.4 dapat dilihat bahwa informasi yang dikirimkan 4 orang peserta dalam melaksanakan ujian *online* melalui aplikasi *client* maupun informasi soal yang dikirimkan oleh *web server* ujian dapat dirahasiakan dengan enkripsi oleh sistem keamanan ujian dengan memanfaatkan algoritme kriptografi Blowfish. Dalam parameter "data" dapat dilihat tipe *value* yang dikirimkan baik melalui peserta dan *server* setelah informasi *ciphertext* yang menunjukkan proses apa yang sedang terjadi dalam aktifitas ujian *online*. Adanya perlindungan fisik pada data pribadi dan informasi ujian peserta akan meningkatkan privasi atau kerahasiaan terhadap segala informasi ujian yang ingin dijaga isinya dari upaya penyadapan saat menggunakan aplikasi untuk memonitoring paket data oleh pihak luar sistem keamanan ujian *online* ini.

5.4.2 Pengujian Integritas (*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*.

5.4.2.1 Skenario Pengujian Integritas

Skenario pengujian integritas (*integrity*) terhadap informasi hasil ujian peserta dilakukan setelah didapat informasi dari hasil penyadapan dari pengujian kerahasiaan (*confidentiality*). Informasi berupa *ciphertext* tersebut harus dapat didekripsi terlebih dahulu agar dapat terlihat struktur atau format parameter data yang dikirimkan. Format parameter data berupa informasi pribadi peserta beserta informasi ujian yang ditransmisikan menuju *web server* adalah sebagai berikut:

(*Session*#Email#Password#[id.soal:jawaban]#date(d m y)**)**

Keterangan:

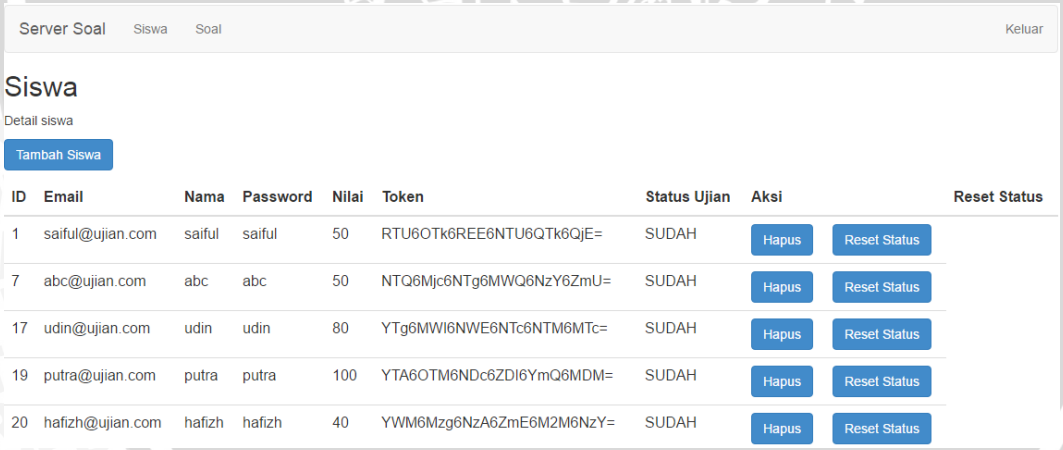
- *Session* = token yang berasal dari *device* peserta ujian untuk mengerjakan ujian
- Email = alamat email sebagai identitas peserta untuk login
- Password = password peserta dari *server* yang digunakan untuk memasuki aplikasi ujian
- [id.soal:jawaban] = nomor soal dengan delimiter berupa penanda ":" terhadap jawabannya. Contoh: nomor soal 1 dengan jawaban B, maka formatnya 1:B
- Date (d m y) = penanda waktu pengiriman informasi dari aplikasi *client* ujian *online*

Paket informasi akan muncul dalam bentuk enkripsi sehingga untuk mengembalikan bentuknya ke format awal seperti yang telah dijelaskan sebelumnya harus dilakukan metode dekripsi. Metode dekripsi yang digunakan pada penelitian ini memanfaatkan *online tools* dari 3 alamat situs yaitu dari halaman *http://sladex.org/blowfish.js/ ; https://www.tools4noobs.com ;* dan *http://codebeautify.org/encrypt-decrypt*. Diasumsikan *attacker* mencoba teknik *brute force* atau beberapa kemungkinan *key* yang dapat digunakan untuk membuka teks dekripsi tersebut.

Selanjutnya diujikan pula serangan untuk melakukan *request* yang sesuai dengan data yang dikirimkan oleh peserta ujian yang sah. Hasil penyadapan menggunakan Wireshark berupa parameter data yang didapat akan digunakan untuk mengetahui pengamanan sistem ujian *online* terhadap integritas data ujian peserta. Pengujian dilakukan menggunakan aplikasi *HttpRequester*, yaitu aplikasi ekstensi pada *browser* untuk melakukan HTTP request dengan pilihan metode GET, POST, dan PUT.

Tabel 5.5 Informasi Pengiriman Paket dari Aplikasi Ujian *Online* Peserta

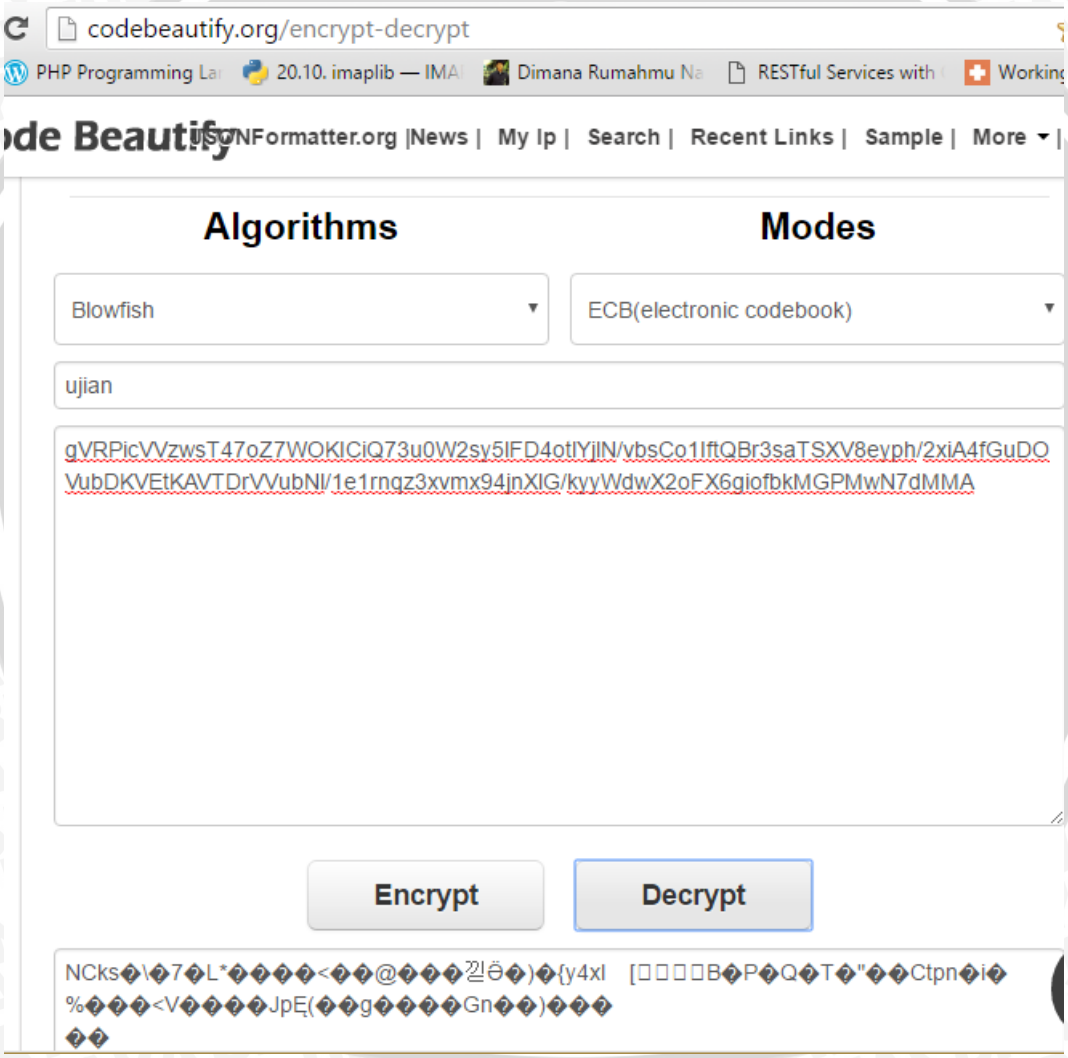
Peserta	Data Asli Peserta Sebelum Enkripsi (*Session*#Email#Password#[id.soal:jawaban]#date)	Data Hasil Enkripsi Blowfish yang Dikirimkan menuju Server	Respon Sistem Keamanan Ujian *Online*
Putra	YTA6OTM6NDc6ZDI6YmQ6MDM=#putra@ujian.com#putra#[1:C@2:C@3:C@4:D@5:C@6:B@7:B@8:C@9:B@10:A]#18:36:05: 29-3-2016	data=gVRPicVVzwsT47oZ7WOKICiQ73u0W2sy5lFD4otlYjlN%2FvbsCo1IftQBr3saTSXV8eyph%2F2xiA4f%0A%2F1e1rnqz3xvmx94jnXlG%2FkyyWdwX2oFX6giofbkMGPMwN7dMMA%0A&type=jawabHTTP/1.1 200 OK Date: Tue, 29 Mar 2016 18:36:05 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 1ISzh2MKsK4lbuea0HXSpn7dGKoBVf6/WHH803TfwgzS0FMN61a0rYbVVAQlMmXSisXCgvlrg+L64+WxJb/N5beZu0yBwVy1mZumXJhaYZgLTxyTm0nNYV4ArXJjEm0aRFhEF2BoTDxXJ9O/NQ1AP/23hddvwh74	Informasi Selamat Putra Ujian selesai, hasil anda : Benar : 10 Salah : 0 Nilai Anda : 100 Ok



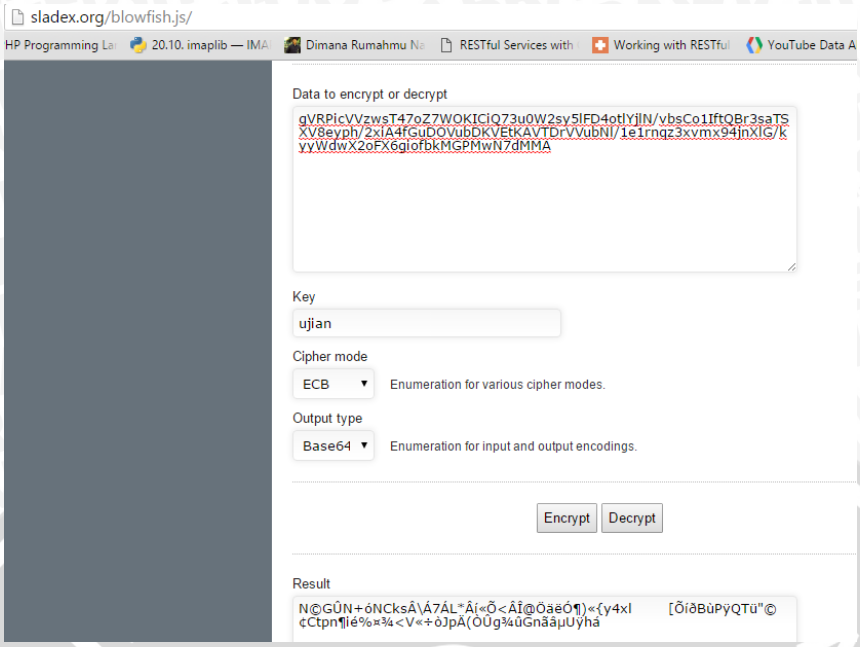
Gambar 5.5 Hasil Pengiriman Data Peserta yang Tersimpan dalam *Database*

5.4.2.2 Hasil Pengujian Integritas

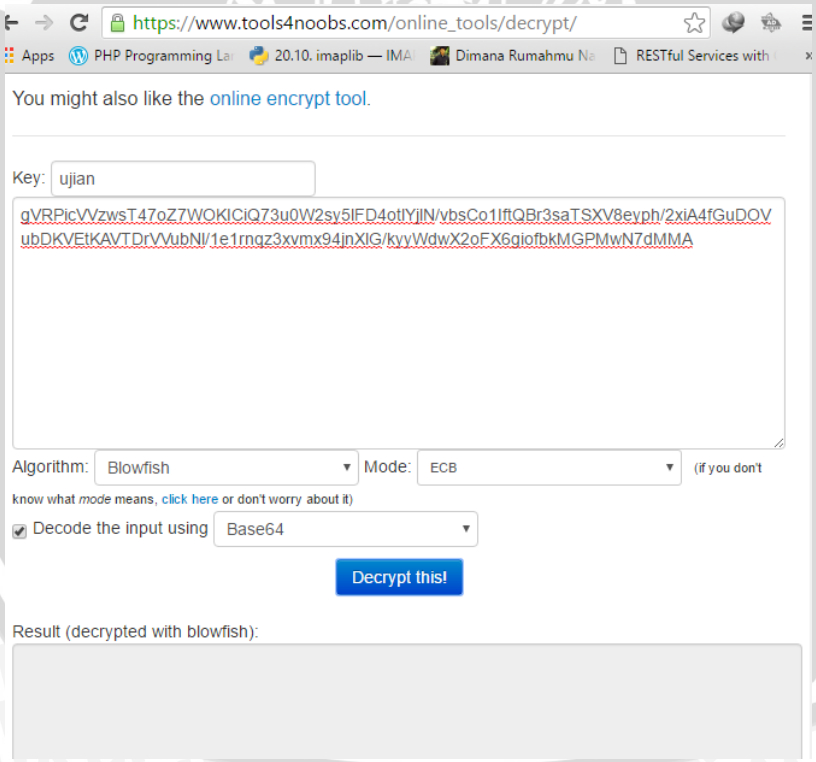
Informasi mengenai paket data yang dikirim oleh peserta melalui aplikasi ujian *online* peserta yang memanfaatkan keamanan algoritme kriptografi Blowfish dapat dilihat pada Tabel 5.5. Di dalam Tabel 5.5 peserta ujian dengan nama "**Putra**" mengirimkan hasil ujiannya sesuai dengan parameter yang telah didefinisikan dalam aplikasi 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 *online tools* yang tersedia. Hasilnya terlihat pada Gambar 5.6, Gambar 5.7, dan Gambar 5.8 berikut.



Gambar 5.6 Percobaan dekripsi terhadap *value* parameter "data" melalui halaman *http://codebeautify.org/encrypt-decrypt*



Gambar 5.7 Percobaan dekripsi terhadap *value* parameter "data" melalui halaman *http://sladex.org/blowfish.js/*



Gambar 5.8 Percobaan dekripsi terhadap *value* parameter "data" melalui halaman *https://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 5.6.

Tabel 5.6 Daftar Hasil Dekripsi yang Didapat melalui Ketiga *Online Tools*

Key	**Decrypt Result**		
	http://codebeautify.org/encrypt-decrypt	*http://sladex.org/blowfish.js/*	*https://www.tools4noobs.com*
Ujian	NCks \ 7 L* \ @  \  -  \  Ə ) {y4xl [B P Q T "  Ctpn  i  %  \  V  JpĘ  (g Gn)	N©G  ÛN+óNCksÂ\Á7Á L*Âí«Õ<ÂÎ@Öä- ë Ó š¶)«{y4xl [Õ˜  íðBùPÿQ  T ü"©¢Ctpn¶ié% ¤¾<V«  ÷òJpÄ˜  (ÒÛg¾  û  Gn ãâ  ™µUÿhá	-
ujian online	-VizV Ş / ὣG ϕ F X_X V u K ,;H \ a a d }{` G `E z `# Yk $>% E r	B/ÿ²óÌ6+:£ ^Î·R1Që?¶ 0¡9¸ñyZyoUñ<Þ ¯ãVÙØ ÊÁdÖË|Ãszõ™®·Ôv³¹ ª Ä¨ ˜ «ªsµvÔÅlÝñ 6¾þ®Õ[[ïQdµ	-
qwerty	P%=GĽ!{ ;RhKxD TeᏂuKm" :/+wy YJ%l-脖 r k_ [l,Hc6	t¼ø0t+¹Br› é(Ë*úÎJ²Ïí é¥BNâ½<ÙÇ_È^÷.ØM” ðãÁSTÊ êá)so9æòó¬^Ð 0¬P¿Ã|ðRÏ zµAz¨äÍd\$ªS-1ä¿ôÆ6 ÷ÉÂˆÑâ	-

matematika	��*Q�&�����bf ��0;��& 9�8�x���D/ ��uB��X�J�s ���YsN�<���<l "�=>�0�&=�vc D��Sgmh)H yA ; ; ;

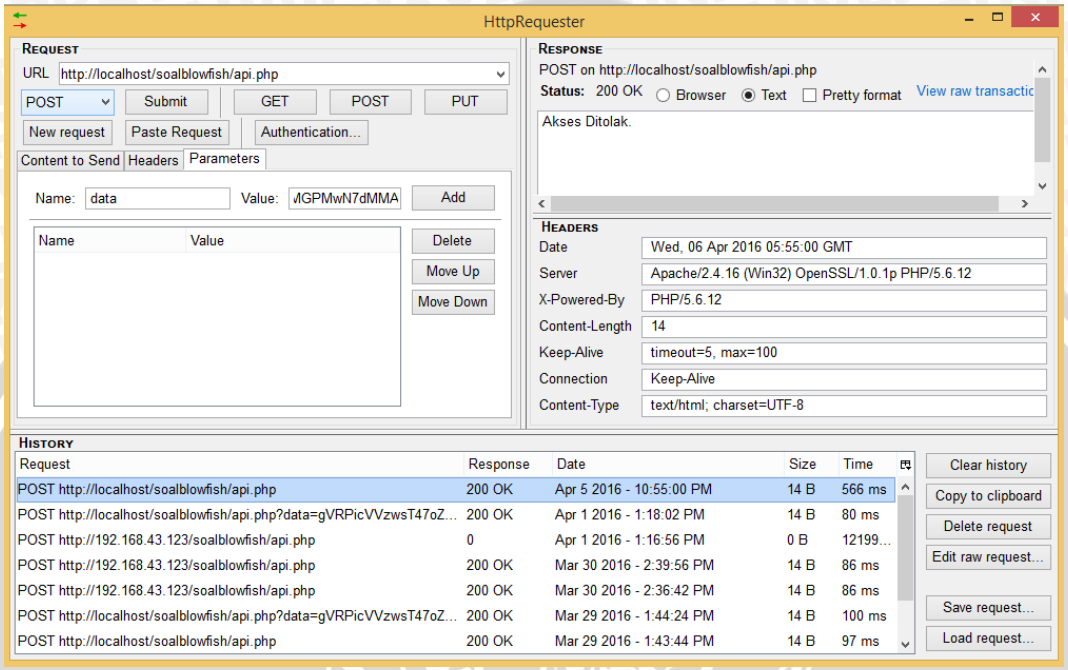
administrator	!M������7W��"a_���o��7L6U$/n��W��[>��K&H& t#k"K4���>*8ahV���- ���Ma_r	f;¥pìs²¿›6x&¹ê(oqí}ý1Ê&a°¥nT³¦Æe¶6A› ¦®à bf£²Ï°BÚ3âp2z]ÕD9$w§£¬óTO²}s› K–˜×⁻™℠é¹ÂÈ ˜ø1JLaÉ®Ue«GÓ	-
examonline	[7"`}^6(|サ^D=v g"^q`!#!k C+Pdʒ Kb%yj9([T Z- z 8n]i0&3 yd ʒ fŋ	&Û(j §ëÏÓš y–-V !:E<“)Eí;`CÊë±GÕzw1ÚÞJ·lìKÉ™Ãy"Ì´nn™ß™¬á6úÅDPg\Øñ¼S Vuò KÞýéª*!1Ò⁻ Ð¹#Ê÷q<™Ó.ö	-

Dari sepuluh 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).

Berdasarkan proses pengiriman informasi seperti yang disajikan Tabel 5.5, untuk dapat membongkar *value* dari parameter data yang dikirimkan, penyerang harus membuka terlebih dahulu teks enkripsi yang disadap. Sedangkan pada percobaan menggunakan beberapa *key* asumsi terhadap sistem keamanan ujian *online* tidak mampu untuk melakukan dekripsi informasinya, sehingga data asli yang dikirimkan peserta "**Putra**" sebelum diberi pengamanan enkripsi tidak dapat diberi intrusi apapun dengan perlindungan dari algoritme kriptografi Blowfish. Sehingga pada pengujian menggunakan skema *brute force* tersebut tidak dapat mengubah keutuhan informasi peserta yang dikirimkan ke dalam *database* ujian *online*.

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 5.5. Masukan URL berdasarkan alamat *server* ujian *online* beserta parameternya yaitu http://localhost/soalblowfish/api.php?data=gVRPicVVzwsT47oZ7WOKICiQ73u0W2sy5lFD4otlYjlN/vbsCo1lftQBr3saTSXV8eyph/2xiA4fGuDOVubDKVEtKAVTDrVVubNl/1e1rnqz3xvmx94jnXlG/kyyWdwX2oFX6giofbkMGPMwN7dMMA.



Gambar 5.9 *Request* HTTP menuju *server* ujian *online*

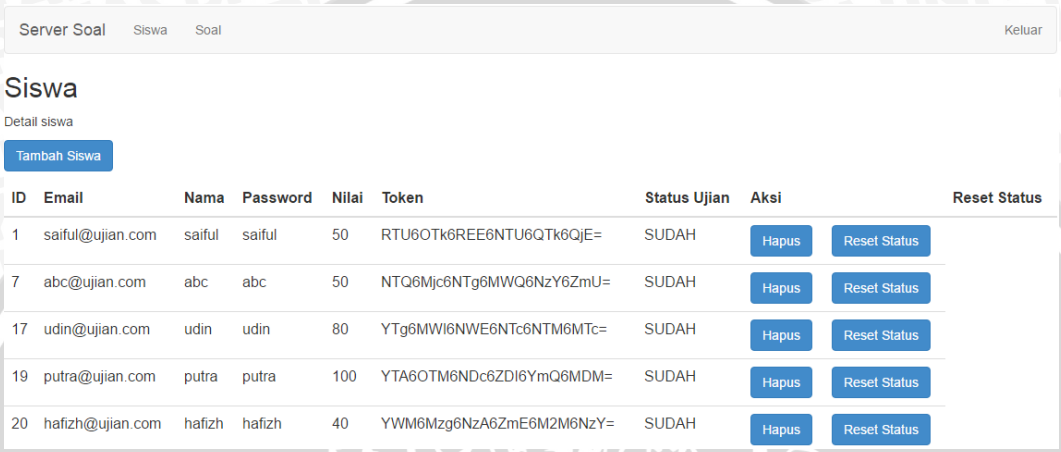
Berdasarkan Gambar 5.9 ditunjukkan bahwa *attacker* ingin melakukan pengubahan data peserta yang telah tersimpan dalam *database* ujian *online* menggunakan informasi yang berhasil disadap sebelumnya, respon dari sistem keamanan ujian *online* yang muncul adalah penolakan akses. Hal ini dikarenakan adanya 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.

Mekanisme pemberian *filter* untuk pendeteksian browser tertentu khusus untuk pelaksanaan di dalam ujian *online* ditunjukkan pada Kode 5.5 berikut.

```
1.  define ('PRIVATE_KEY' , 'ujianonline' );
2.  define ('DELIMITER' , '#' );
3.  define ('AGENT' , 'Apache-HttpClient' );
4.
5.  if (strpos($_SERVER['HTTP_USER_AGENT'], AGENT) === false) {
6.      exit("Akses Ditolak.");  }
```

Kode 5.5 Kode Program Pemberian Mekanisme Keamanan untuk *Handle* Browser *Client*

Hasil pengujian menunjukkan dengan adanya mekanisme keamanan tersebut dapat mencegah adanya akses dari luar spesifikasi sistem yang telah dibangun dalam pelaksanaan ujian *online* ini sehingga sistem tidak melakukan tindakan apapun terhadap intrusi pengubahan informasi ujian yang dapat mempengaruhi integritas data dalam *database* ujian *online*. Pada Gambar 5.10 berikut dapat dilihat setelah dilakukan *refresh* oleh administrator terhadap halaman data peserta ujian menunjukkan bahwa informasi peserta "**Putra**" yang menjadi target sasaran *attacker* tetap menyimpan data semula tanpa adanya pengubahan.



Gambar 5.10 Kondisi Akhir Halaman Peserta dalam *Database* Ujian *Online*

5.4.3 Analisis Hasil Pengujian

Proses analisis terhadap hasil pengujian dilakukan untuk mendapatkan kesimpulan dari hasil pengujian implementasi algoritme kriptografi Blowfish pada sistem keamanan ujian *online*. Proses analisis mengacu pada hasil pengujian yang telah didapatkan. Analisis dilakukan di setiap tahap pengujian yang meliputi analisis hasil pengujian keamanan pada aspek kerahasiaan (*confidentiality*) dan keutuhan (*integrity*).

5.4.3.1 Analisis Hasil Pengujian Kerahasiaan (*Confidentiality*)

Proses analisis terhadap hasil pengujian kerahasiaan dilakukan dengan melihat apakah aspek keamanan pada aplikasi telah terpenuhi berdasarkan kebutuhan non fungsional sistem. Berdasarkan hal tersebut dan pengujian yang dilakukan, dapat diambil kesimpulan bahwa sistem keamanan ujian *online* mampu memberikan perlindungan fisik terhadap pengiriman informasi ujian dari aplikasi *client* peserta maupun *web server* ujian *online* melalui proses enkripsi dari algoritme kriptografi Blowfish sehingga saat pihak luar yang ingin menyadap informasinya melalui aplikasi *sniffing* jaringan hanya memperoleh data *ciphertext* ujian yang tidak dapat dimengerti maksud isinya.

5.4.3.2 Analisis Hasil Pengujian Integritas (*Integrity*)

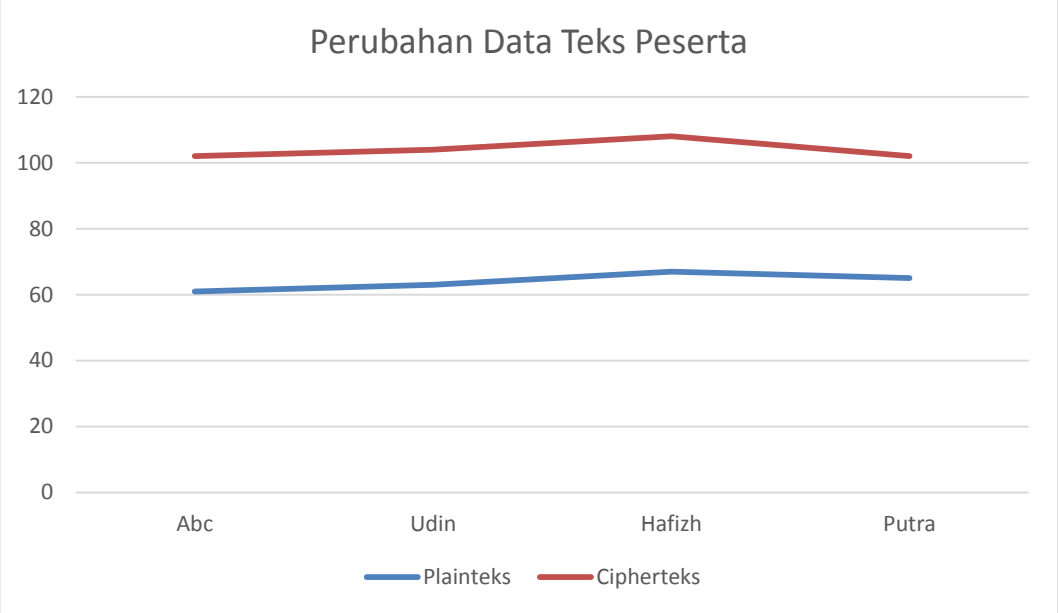
Proses analisis terhadap hasil pengujian integritas informasi ujian dilakukan dengan mengamati hasil pengujian integritas yang telah dilakukan sebelumnya. Berdasarkan hasil pengujian, dapat diketahui bahwa untuk membuka isi dari informasi *ciphertext* ujian dengan metode *brute force* menggunakan 10 buah asumsi *key* yang mungkin diterapkan, menunjukkan hasil dekripsi menjadi format yang tidak sesuai dengan data asli yang dikirimkan sebelum proses enkripsi. Selain itu saat dilakukan *request* ulang terhadap parameter sesuai masukan dari peserta ujian yang asli, sistem keamanan ujian *online* memberi respon penolakan akses terhadap upaya intrusi tersebut sehingga data peserta ujian yang telah tersimpan dalam *database* tetap utuh tanpa adanya upaya modifikasi oleh pihak luar.

5.4.3.3 Analisis Hasil Perubahan Informasi Ujian Peserta

Proses analisis terhadap hasil perubahan informasi ujian peserta dilakukan dengan melihat perubahan ukuran data plainteks saat dikonversi menjadi cipherteks. Hasil perubahan ukuran secara spesifik dapat dilihat Tabel 5.7 sampai Tabel 5.10 dengan setiap tabel mendefinisikan dari setiap aktifitas ujian yang dilakukan beserta grafik yang menunjukkan perbedaan ukuran dari tiap tabel.

Tabel 5.7 Hasil Perubahan Data Plainteks ke Cipherteks Milik Peserta pada Proses Validasi

No	Plainteks	Jumlah karakter	Cipherteks	Jumlah karakter
1	NTQ6Mjc6NTg6MWQ6NzY6ZmU=#abc@ujian.com#abc#10:56:51:29-3-2016	61	DFXBQYY6%2FjOHhKAetU286fUegXW0%2FvhdrTUZ7iKpX8r3pSKeQVV1Z2OQgM1GqBrO%2BPpcZOAiPFrDn46w6BWYp%2B8t78R%0A	102
2	YTg6MWI6NWE6NTc6NTM6MTc=#udin@ujian.com#udin#10:57:49:29-3-2016	63	HmtSEyJCHrDG3HxItmLREHI9GpH6cKWj7%2Be6rHekvdUN1eQaJEUqrFTUChCtiN78IGGIErTWekRf%0AQudqVV4mimzDEPq2tuUm%0A	104
3	YWM6Mzg6NzA6ZmE6M2M6NzY=#hafizh@ujian.com#hafizh#10:57:58:29-3-2016	67	g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgQyhe7230VwUSaGF2Eq6AKPkOwCbH6gBOlwJV%2BsnNWOHa%0AgVfqCKh9uQwY8zA3t0ww%0A	108
4	YTA6OTM6NDc6ZDI6YmQ6MDM=#putra@ujian.com#putra#10:56:4129-3-2016	65	gVRPicVVzwsT47oZ7WOKIIxYfI9d9WML4mqTQZaD8pFc7dzPCqXeHyEP5YBFzi7uXUCK5CKBusVO%0AxpJtxROSJz3KZIwTEp3r%0A	102



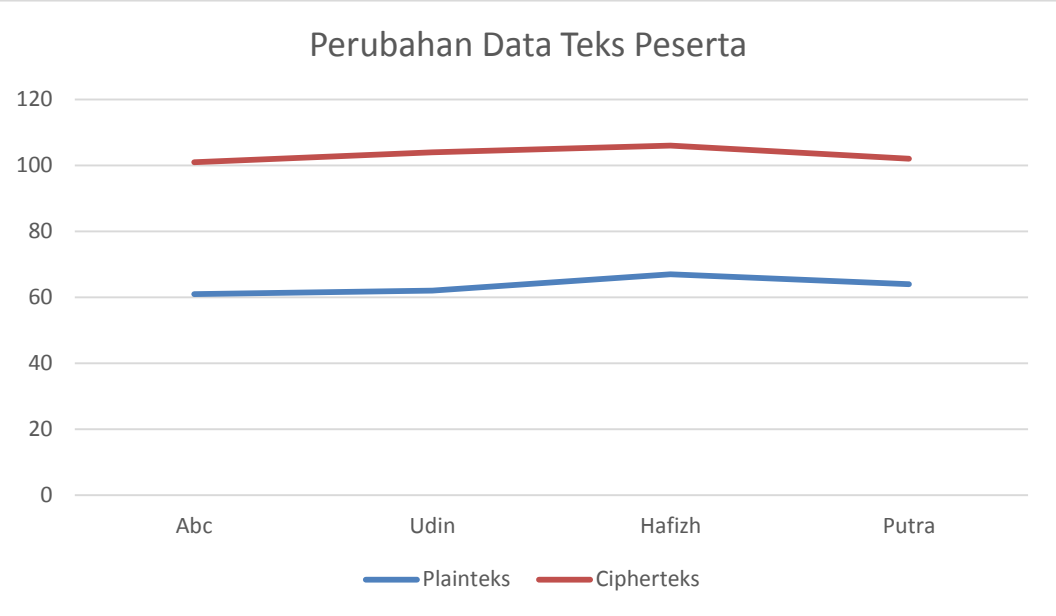
Gambar 5.11 Grafik Perubahan Data Teks Peserta pada Proses Validasi

Pada Tabel 5.7 ditunjukkan data asli peserta ujian yang dikirimkan melalui aplikasi ujian *online* bersama dengan data perubahan setelah diberi perlakuan enkripsi dengan algoritme kriptografi Blowfish untuk proses validasi. Saat peserta melakukan validasi, informasi yang ditransmisikan yaitu berupa *session* dari *MAC Address* perangkat beserta data pribadi dan waktu pelaksanaan validasi. Setelah melalui proses enkripsi, ukuran data mengalami perubahan sebagaimana ditunjukkan pada grafik pada Gambar 5.11. Adapun hasil rata-rata yang diperoleh terhadap perubahan ukuran data plainteks saat menjadi data cipherteks adalah **penambahan sebanyak 40 karakter.**

Tabel 5.8 Hasil Perubahan Data Plainteks ke Cipherteks Milik Peserta pada Proses *Login*

No	Plainteks	Jumlah karakter	Cipherteks	Jumlah karakter
1	NTQ6Mjc6NTg6MWQ6NzY6ZmU=#abc@ujian.com#abc#10:58:22:29-3-2016	61	DFXBQYY6%2FjOHhKAetU286fUegXW0%2FvhdrTUZ7iKpX8r3pSKeQVV1Z2OQgM1GqBYpWbdySuR%0AiDn46w6BWYp%2B8t78R%0A&	101
2	YTg6MWI6NWE6NTc6NTM6MTc=#udin@ujian.com#udin#10:58:2:29-3-2016	62	HmtSEyJCHrDG3HxItmLREHI9GpH6cKWj7%2Be6rHekvdUN1eQaJEUqrFTUChCtiN7841S3qrHsMQ1f%0AQudqVV4mimzDEPq2tuUm%0A	104
3	YWM6Mzg6NzA6ZmE6M2M6NzY=#hafizh@	67	g%2F8ziIjCdDZaHebupyvrQIhSIcqx8%2FKvIfDKK6CX7ifDRpmQssqvDnV2bKOSnQEJR	106

	ujian.com#hafizh#10:58:11:29-3-2016		WdI1vf0pelg%0AYAYu6XeVwWSg78s67BQW%0A	
4	YTA6OTM6NDc6ZDI6YmQ6MDM=#putra@ujian.com#putrp#10:57:28:29-3-2016	64	gVRPicVVzwsT47oZ7WOKIIxYfI9d9WML4mqTQZaD8pFc7dzPCqXeHyEP5YBFzi7uZXmBCtrJza9g%0AYAYu6XeVwWSg78s67BQW%0A	102



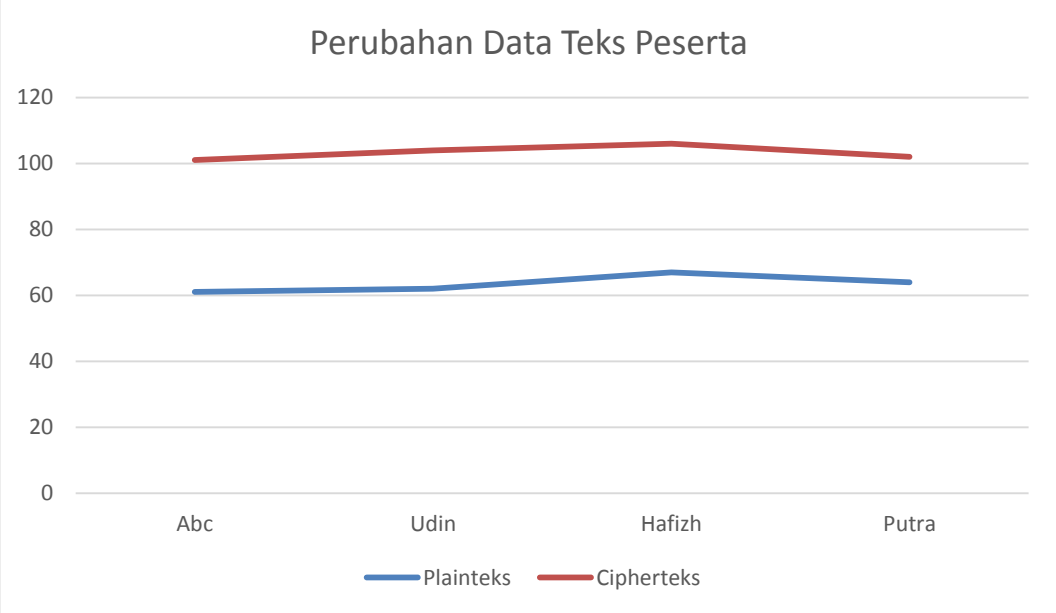
Gambar 5.12 Grafik Perubahan Data Teks Peserta pada Proses *Login*

Pada Tabel 5.8 ditunjukkan data asli peserta ujian yang dikirimkan melalui aplikasi ujian *online* bersama dengan data perubahan setelah diberi perlakuan enkripsi dengan algoritme kriptografi Blowfish untuk proses *login*. Saat peserta melakukan *login* untuk mengerjakan ujian, informasi yang ditransmisikan yaitu berupa *session* dari *MAC Address* perangkat beserta data pribadi dan waktu pelaksanaan *login*. Setelah melalui proses enkripsi, ukuran data mengalami perubahan sebagaimana ditunjukkan pada grafik pada Gambar 5.12. Adapun hasil rata-rata yang diperoleh terhadap perubahan ukuran data plainteks dengan data cipherteks peserta adalah **penambahan sebanyak 39,75 karakter.**

Tabel 5.9 Hasil Perubahan Data Plainteks ke Cipherteks Milik Peserta pada Proses Pengambilan Soal

No	Plainteks	Jumlah karakter	Cipherteks	Jumlah karakter
1	NTQ6Mjc6NTg6MWQ6NzY6ZmU=#abc@ujian.com#abc#10:58:35:29-3-2016	61	DFXBQYY6%2FjOHhKAetU286fUegXW0%2FvhKpX8r3pSKeQVV1Z2OQgM1GqBrOf1lsGLkVx32R%0AiPFrDn46w6BWYp%2B8t78R%0A	101

2	YTg6MWI6NWE6NTc6NTM6MTc=#udin@ujian.com#udin#10:58:8:29-3-2016	62	HmtSEyJCHrDG3HxItmLREHI9GpH6cKWj7%2Be6rHekvdUN1eQaJEUqrFTUChCtiN780POznZVa6fpf%0AQudqVV4mimzDEPq2tuUm%0A	104
3	YWM6Mzg6NzA6ZmE6M2M6NzY=#hafizh@ujian.com#hafizh#10:57:15:29-3-2016	67	g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgQyhe7230VwUSaGF2Eq6AKPkOwCbH6gBOwvuMVUeKLRe6%0A5spd8iqRmAwY8zA3t0ww%0A	106
4	YTA6OTM6NDc6ZDI6YmQ6MDM=#putra@ujian.com#putra#10:57:31:29-3-2016	64	gVRPicVVzwsT47oZ7WOKIIxYfI9d9WML4mqTQZaD8pFc7dzPCqXeHyEP5YBFzi7uttSWBYESR4xg%0AYAYu6XeVwWSg78s67BQW%0A	102

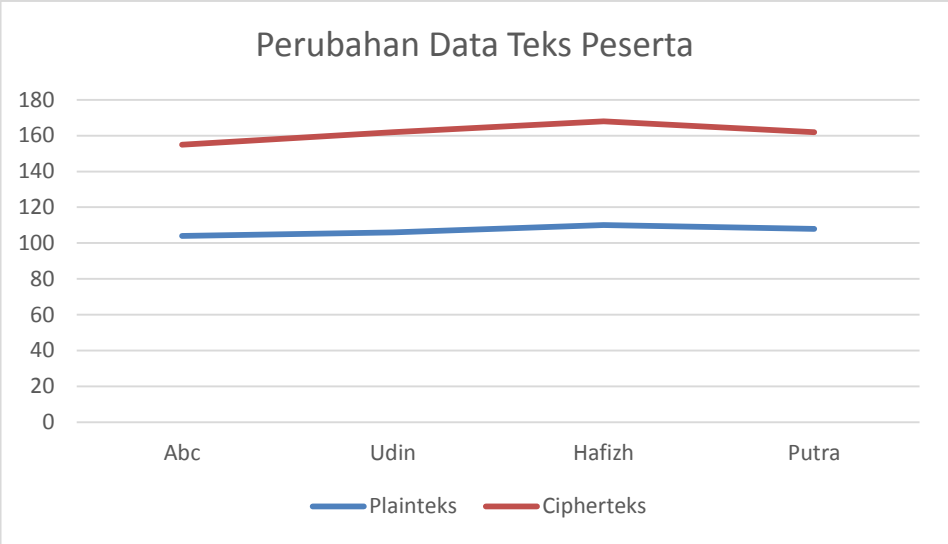


Gambar 5.13 Grafik Perubahan Data Teks Peserta pada Proses Pengambilan Soal

Pada Tabel 5.9 ditunjukkan data asli peserta ujian yang dikirimkan melalui aplikasi ujian *online* bersama dengan data perubahan setelah diberi perlakuan enkripsi dengan algoritme kriptografi Blowfish untuk proses pengambilan soal. Saat aplikasi ujian peserta mengambil soal dari database ujian *online*, informasi yang ditransmisikan yaitu berupa *session* dari *MAC Address* perangkat beserta data pribadi dan waktu pelaksanaan pengambilan soal. Setelah melalui proses enkripsi, ukuran data mengalami perubahan sebagaimana ditunjukkan pada grafik pada Gambar 5.13. Adapun hasil rata-rata yang diperoleh terhadap perubahan ukuran data plainteks dengan data cipherteks peserta adalah **penambahan sebanyak 39,75 karakter.**

Tabel 5.10 Hasil Perubahan Data Plainteks ke Cipherteks Milik Peserta pada Proses Pengiriman Jawaban

No	Plainteks	Jumlah karakter	Cipherteks	Jumlah karakter
1	NTQ6Mjc6NTg6MWQ6NzY6ZmU=#abc@ujian.com#abc#[1:A@2:C@3:C@4:D@5:A@6:B@7:C@8:C@9:B@10:B]#11:17:51:29-3-2016	104	DFXBQYY6%2FjOHhKAetU286eJzINLF5LUoxcj4RGPDKXrLYiAbTrk1mo4ZNlw93MDtQOCPR%0Anm7qnwVcW02VSls4Fmz6Bf4pD5g7a90xU1tSPVJd3P6BrTUr21%2Fa0xagwphm7HJTT6ae%2FfNgrg%0A	155
2	YTg6MWI6NWE6NTc6NTM6MTc=#udin@ujian.com#udin#[1:A@2:C@3:C@4:D@5:B@6:C@7:B@8:C@9:B@10:D]#11:16:21:29-3-2016	106	HmtSEyJCHrDG3HxItmLREPqiR5Yblq06WYuuOkxftgrnvK4I5U2OJZMfooEyfK54okK3FkPXH%2BD7%0AvqGX0Kc7SFQW6Cj%2FAodbn9qaz7i6gOjuarMvp%2FxZrt7L8fMGeguQTsaSbcUTkic9ymSMExKd6w%0A	162
3	YWM6Mzg6NzA6ZmE6M2M6NzY=#hafizh@ujian.com#hafizh#[1:C@2:B@3:C@4:A@5:D@6:B@7:B@8:A@9:A@10:A]#11:17:44:29-3-2016	110	g%2F8ziIjCdDZaHebupyvrQIy1%2Bw0s0CLgsO0pj%2Bq63EpnlBPyol9zNdBtKd1uyfEJxqDjhNYA7ycO%0A%2BeftF5BtShNfMpuM60O2WL4vJi7JSKO8KKYqHwEJOarJQGePd5A9s8%2FKxLVDNVPN%2BLdC6LRsdg%0A	168
4	YTA6OTM6NDc6ZDI6YmQ6MDM=#putra@ujian.com#putra#[1:C@2:C@3:C@4:D@5:C@6:B@7:B@8:C@9:B@10:A]#11:16:05:29-3-2016	108	gVRPicVVzwsT47oZ7WOKICiQ73u0W2sy5lFD4otlYjlN%2FvbsCo1lftQBr3saTSXV8eyph%2F2xiA4f%0AGuDOVubDKVEtKAVTDrVVubNl%2F1e1rnqz3xvmx94jncBuRAvgst1auubKXfIqkZgMGPMwN7dMMA%0A	162



Gambar 5.14 Grafik Perubahan Data Teks Peserta pada Proses Pengiriman Jawaban

Pada Tabel 5.10 ditunjukkan data asli peserta ujian yang dikirimkan melalui aplikasi ujian *online* bersama dengan data perubahan setelah diberi perlakuan enkripsi dengan algoritme kriptografi Blowfish untuk proses pengambilan soal. Saat peserta melakukan pengiriman jawaban, informasi yang ditransmisikan selain berupa *session MAC Address* perangkat, pribadi, dan waktu pelaksanaan pengiriman jawaban juga terdapat penambahan informasi kode soal beserta jawaban yang dipilih peserta. Setelah melalui proses enkripsi, ukuran data mengalami perubahan sebagaimana ditunjukkan pada grafik pada Gambar 5.14. Adapun hasil rata-rata yang diperoleh terhadap perubahan ukuran data plainteks dengan data cipherteks peserta adalah **penambahan sebanyak 54,75 karakter.**

Hasil analisis perubahan informasi ujian peserta menunjukkan bahwa penerapan enkripsi memanfaatkan algoritme kriptografi Blowfish akan berpengaruh terhadap ukuran data, yaitu informasi ujian selalu akan bertambah ukurannya setelah data menjadi bentuk cipherteks. Dalam hal ini juga terbukti bahwa mekanisme enkripsi dari sistem berhasil diterapkan untuk mencapai tujuan penelitian dengan terpenuhinya aspek keamanan yang ingin dilindungi terhadap informasi ujian peserta.

BAB 6 PENUTUP

6.1 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*.

6.2 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.