Pengamanan Data Penerimaan Mahasiswa Baru dengan Algoritma Vernam Cipher, DES dan Diffie Hellman: Studi Kasus pada Aplikasi BluCampus Universitas Budi Luhur

Muhamad Refaldi[1)], **Achmad Solichin**[2)]
[1]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : muhamadrefaldi@gmail.com[1)], achmad.solichin@budiluhur.ac.id[2)]

Abstrak

Pada era digital sekarang ini pertukaran informasi menjadi hal yang sangat penting dan menjadi kebutuhan umum masyarakat. Contohnya adalah proses administrasi pendaftaran mahasiswa baru melalui perangkat yang terkoneksi dengan internet. Pertukaran informasi data personal melibatkan pihak ketiga sebagai perantara (server). Untuk data yang bersifat pribadi dan memiliki resiko jika disalahgunakan oleh pihak yang tidak bertanggung jawab tentunya memerlukan suatu metode yang dapat menjamin keamanan dan kerahasiaan data tersebut. Salah satu solusi pengamanan informasi yang dapat dilakukan adalah dengan mengimplementasikan ilmu kriptografi pada sistem. Pengirim akan mengirimkan datanya kepada penerima, tetapi sebelum data tersebut dikirimkan ke server, data tersebut dienkripsi menjadi ciphertext, kemudian dikirim ke sever. Penerima akan mendekripsikan ciphertext menjadi bentuk data semula yang mengandung informasi. Dalam penelitian ini penulis menggabungkan dua algoritma kriptografi untuk mengamankan data pendaftar atau calon mahasiswa yaitu Vernam Cipher dan DES (Data Encryption Standard) dengan pembangkitan kuncinya menggunakan algoritma Diffie-Hellman pada fitur informasi tes mahasiswa baru di aplikasi BluCampus berbasis Android. Berdasarkan penelitian, terbukti bahwa aplikasi ini mampu melakukan enkripsi nomor pendaftaran dan berhasil mendenkripsikan ciphertext dari web service ke bentuk data semula.

Kata kunci: Kriptografi, *Vernam Cipher*, DES, *Diffie-Hellman*

1. PENDAHULUAN

Universitas Budi Luhur pada tiap tahunnya menerima ribuan pendaftar sebagai calon mahasiswa. Proses administrasi pendaftaran sebagian besar masih dilakukan dengan cara klasik, yaitu datang langsung ke tempat pendaftaran atau ke kampus. Mulai dari proses pendaftaran, pengumuman informasi tes, dan pengumuman informasi jadwal ORDIK masih dilakukan dengan cara klasik. Bayangkan saja jika puluhan atau ratusan pendaftar datang pada hari yang sama, pasti akan menimbulkan antrian yang sangat panjang dan tentu saja tidak bisa tertampung semua di dalam ruangan. Begitu juga ketika ada pengumuman terkait proses penerimaan mahasiswa baru, misalnya informasi tes masuk. Jika masih menggunakan cara klasik yaitu dengan papan pengumuman, pasti banyak calon mahasiswa yang berdesakan melihat papan pengumuman tersebut. Atas dasar itulah perlu dibuatnya sistem pendaftaran *online* dengan fitur yang menyertainya agar proses administrasi pendaftaran bisa dilakukan lebih praktis dan murah.

Kemanan sistem juga menjadi hal yang penting agar data-data pendaftar aman dari pihak yang tidak bertanggung jawab *(intruder)*. Ilmu kriptografi bisa digunakan sebagai solusi dalam hal ini. Pada aplikasi ini digunakan beberapa algoritma kriptografi untuk pengamanan data, yaitu *Vernam Cipher,* DES *(Data Encryption Standard),* dan *Diffie-Hellman* untuk pembangkitan kunci.

Data-data yang diisi oleh pendaftar akan dienkripsi dahulu dengan kombinasi ketiga algoritma enkripsi, sebelum dikirimkan ke *server* menjadi sandi yang tidak bisa diketahui maknanya. Setelah server menerima sandi tersebut, makan sandi tersebut akan didekripsi menjadi tulisan awal atau data-data diri pendaftar yang sebenarnya. Begitu juga sebaliknya, *server* akan menyandikan pesan sebelum dikirimkan ke *user* dan nanti *user* yang akan mendekripsikan sandi menjadi bentuk semula.

Dalam penelitian ini, penulis menggunakan metode pengembangan dengan model *Software Development Life Cycle* (SLDC) yaitu metode *Waterfall*. Tahapan penelitiannya yaitu perencanaan, analisis, desain, implementasi, pengujian dan pemeliharaan.

2. TINJAUAN PUSTAKA

2.1. Pengertian Kriptografi

Kriptografi adalah ilmu yang mempelajari tentang bagaimana cara untuk menyembunyikan pesan. Pada pengertian modern, kriptografi diartikan sebagai ilmu teknik matematika yang digunakan untuk keamanan informasi [1].

Secara umum, kriptografi dibagi menjadi 2 jenis, yaitu kriptografi simetris dan kriptografi asimetris. Kriptografi simetris menggunakan kunci yang sama pada proses enkripsi dan dekripsi, contohnya antara lain *Vernam Cipher* dan DES *(Data Encryption Standard)*. Sedangkan kriptografi adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi, contohnya yaitu RSA *(Rivest Shamir Adleman)*.

2.2. Diffie-Hellman

Diffie-Hellman dikembangkan oleh Whitfield Diffie dan Martin Hellman pada tahun 1976. Algoritma ini tidak dipakai untuk melakukan enkripsi data, tapi untuk membangkitkan nilai kunci privat yang sama [2]. Tujuan algoritma ini yaitu untuk prosedur pertukaran kunci yang aman.

2.3. Algoritma DES (Data Encryption Standard)

DES termasuk sistem kriptografi yang tergolong ke dalam jenis *block cipher* [3]. DES mengenkripsikan 64 bit *plaintext* menjadi 64 bit *ciphertext* dengan menggunakan 56 bit kunci internal atau *subkey*. Kunci internal dibangkitkan dari kunci eksternal yang panjangnya 64 bit [4]. DES termasuk sistem kriptografi simetris, artinya kunci yang digunakan untuk enkripsi sama dengan yang digunakan untuk dekripsi. Penerapan algoritma DES juga dapat digabungkan dengan konsep steganografi [6] untuk lebih mengamankan data.

2.4. Algoritma Vernam Cipher

Algoritma *Vernam Cipher* ditemukan oleh Gilbert Vernam pada tahun 1917. Algoritma ini mendapatkan reputasi sebagai algoritma yang kuat namun sederhana dengan tingkat keamanan yang tinggi [5]. Secara komputasi, fungsi untuk mengenkripsi pesan hanyalah dengan meng-XOR-kan *plaintext* dengan kunci. Sebaliknya, untuk proses dekripsi maka dilakukan proses XOR antara *ciphertext* dengan kunci yang sama. Tapi sebelumnya karakter yang digunakan harus diubah ke bentuk biner.

3. METODOLOGI

3.1. Metode Penulisan

Dibawah ini adalah rincian tahapan pengembangan aplikasi penerimaan mahasiswa baru dengan metode *Waterfall* :

1. Pengumpulan Data

   Mengumpulkan kebutuhan dari keseluruhan elemen sistem yang akan diaplikasikan ke dalam bentuk *software* atau perangkat lunak.
2. Menganalisis kebutuhan aplikasi

   Setelah memperoleh kebutuhan aplikasi kemudian dipelajari dan dianalisa mengenai fungsi-fungsi apa saja yang diperlukan untuk mengimplementasikan aplikasi ini.
3. Desain atau Perancangan Program

   Merancang tampilan aplikasi yang akan dibangun sesuai dengan kebutuhan aplikasi sehingga dapat mempermudah dalam proses pengkodean.
4. Pengkodean

   Pengkodean dilakukan untuk memudahkan dalam mengimplementasikan rancangan aplikasi ke dalam algoritma *Vernam Cipher,* DES *(Data Encryption Standard)*, dan *Diffie-Hellman* dengan menggunakan bahasa pemrograman Java.
5. Implementasi

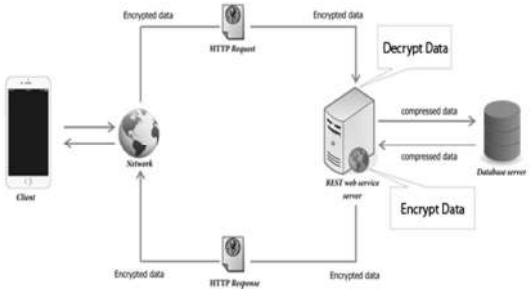
   Rancangan aplikasi yang sudah dibuat kemudian diimplementasikan berdasarkan analisa masalah.
6. Pengujian

   Pengujian dilakukan setelah aplikasi selesai dibuat dengan melakukan beberapa pengujian program dan mencari kesalahan pada program hingga tidak ada lagi kesalahan program dan program sudah berjalan sesuai dengan yang dirancang. Penulis juga mencari beberapa responden untuk mencoba menggunakan aplikasi ini, kemudian responden diminta untuk mengisi kuesioner yang berisi pertanyaan terkait penilaian aplikasi yang terdiri dari beberapa aspek penilaian.

3.2. Analisis dan Penyelesaian Masalah

Setiap universitas memiliki informasi atau data yang bersifat pribadi dan rentan untuk disalahgunakan.. Misalnya data pada proses administrasi pendaftaran mahasiswa baru terdapat data *username* dan *password* untuk *login* pada saat tes. Jika data tersebut tidak diamankan, maka bisa saja muncul masalah baru seperti orang jahil yang menginterupsi *login* pendaftar karena datanya berhasil didapatkan dan mudah dibaca.

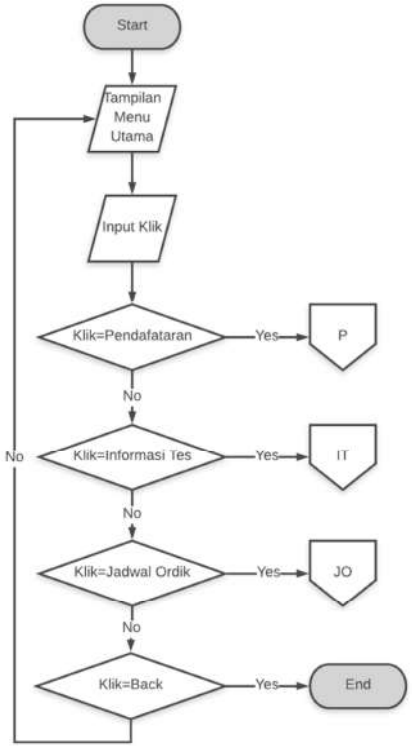
Berdasarkan masalah yang dihadapi, maka salah satu solusi yang dapat digunakan adalah dengan cara mengimplementasikan ilmu kriptografi pada sistem. Dalam kriptografi, terdapat proses enkripsi yaitu proses dimana data asli *(plaintext)* akan diacak atau dikodekan agar maknanya tidak bisa dimengerti. Ada juga proses dekripsi yaitu kebalikan dari proses enkripsi, dimana data akan diubah kembali ke bentuk semula agar bisa dimengerti maknanya. Untuk mengimplementasikan ilmu kriptografi, harus dibuat suatu aplikasi yang terdapat algoritma kriptografi. Dengan adanya kriptografi pada sistem diharapkan data-data yang dikirimkan dan diterima dapat terjamin keamanannya dan juga terjamin kerahasiaannya. Aplikasi ini juga menggunakan pihak ketiga. Pihak ketiga yang dimaksud adalah *server*, karena aplikasi ini bersifat *onine*. Dibuat juga suatu API yang menjembatani untuk pertukaran informasi antara Android dan *web service*. Jadi Android tidak akan menyimpan data pada penyimpanan lokal.



Gambar 1. Arsitektur Sistem

3.3. Flowchart dan Algoritma Program

Pada bagian ini akan disajikan rancangan *flowchart* dan algoritma alur proses yang terdapat pada aplikasi, yaitu proses pada menu utama dan modul informasi tes. Pada menu utama, terdapat tiga pilihan menu yaitu menu pendaftaran, informasi tes, dan informasi ORDIK (Orientasi Pendidikan). Apabila *user* menekan tombol menu pendaftaran, maka halaman *form* pendaftaran. Begitu juga dengan menu yang lainnya, menu halaman informasi tes akan terbuka jika *user* menekan tombol menu informasi tes, dan jika *user* menekan tombol menu jadwal ORDIK, maka akan muncul halaman untuk mengakses informasi jadwal ORDIK.

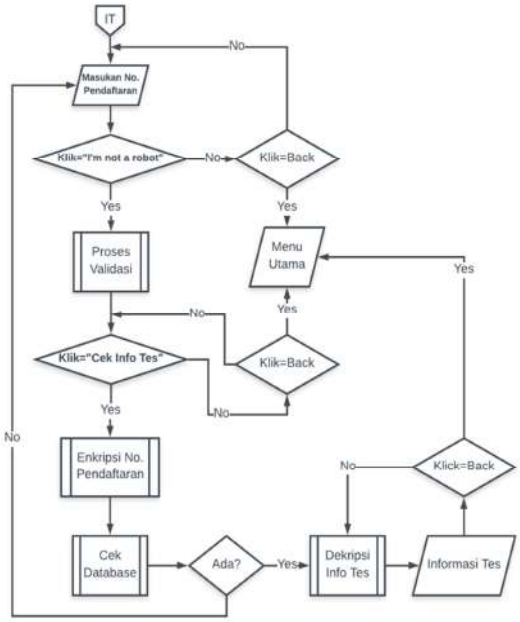


Gambar 2. Flowchart Menu Utama

Algoritma proses dari *flowchart* menu utama :

1. Start
2. Tampilkan Menu Utama
3. *Input klik*
4. *If klik* = "Pendaftaran" *then*
5. Jalankan Menu Pendaftaran
6. *Else if klik*= "Informasi Tes" *then*
7. Jalankan Menu Informasi Tes
8. *Else if klik*= "Jadwal Ordik" *then*
9. Jalankan Menu Jadwal ORDIK
10. *Else if klik= Back then*
11. Keluar aplikasi
12. *Else*
13. Tetap di Menu Utama
14. *End if*
15. *End*

Berikutnya adalah *flowchart* dan algoritma proses pada modul atau menu informasi tes. Pada modul informasi tes, yang pertama kali muncul adalah kolom masukan untuk mengisi nomor pendaftaran peserta yang mengikuti tes. Kemudian ada proses validasi dengan *reCaptcha*, jika validasi berhasil atau benar dan *user* menekan tombol "Cek Info Tes", maka nomor pendaftaran yang telah dimasukkan akan dienkripsi terlebih dahulu kemudian baru dikirim ke *web service*. Jika data ada, maka akan ditampilkan informasi tes berdasarkan nomor pendaftaran yang telah dimasukan, dimana data tersebut telah didekripsi terlebih dahulu.



Gambar 3. Flowchart Menu Informasi Tes

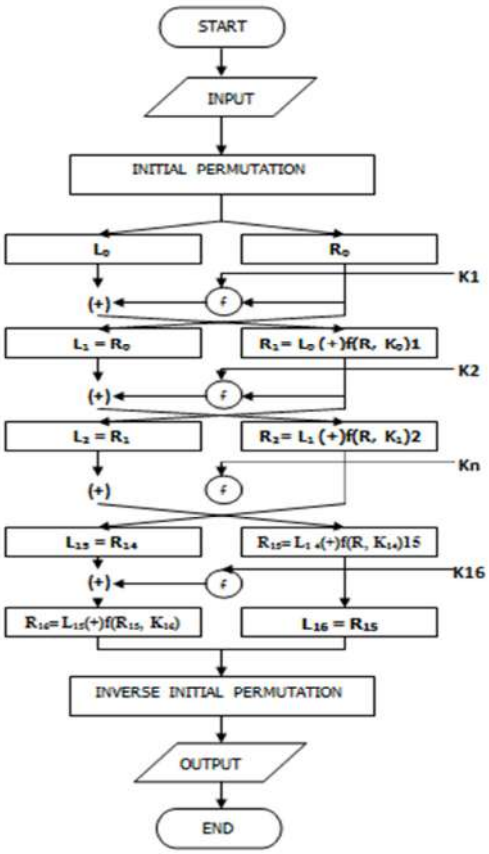
Algoritma dari *flowchart* menu informasi tes :

1. Tampil Menu Informasi Tes
2. *Input* no. Pendaftaran
3. *Input klik*
4. *If klik= "i'm not a robot" then*

```
5.      Proses validasi deteksi bot
6.      If valid then
7.          Muncul tombol "CEK INFO TES"
8.          Input klik
9.              If klik = "CEK INFO TES" then
10.                 Enkripsi no. Pendaftaran
11.                 Kirim no. Pendaftaran ke server
12.                     If no.pendaftaran ada di
                        database then
13.                         Get info tes
14.                         Dekripsi info tes
15.                     Else
16.                         Isi kolom pendafataran
                Else if klik = Back then
18.                 Kembali ke Menu Utama
19.     Else
20.         Klik ulang
21. Else if klik = Back then
22.     Kembali ke Menu Utama
23.     End if
```

3.4. Flowchart dan Algoritma Enkripsi

Pada bagian ini akan dijelaskan *flowchart* dan algoritma dari metode enkripsi yang digunakan untuk pengamanan data pada aplikasi ini. Yang pertama yaitu algoritma enkripsi DES (*Data Encryption Standar*).



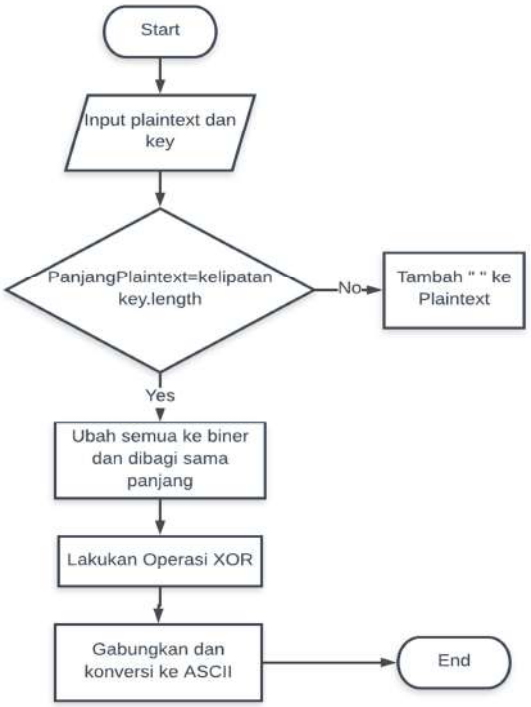
Gambar 4. Flowchart Enkripsi DES

Algoritma dari *flowchart* DES :

1. *Get plaintext*
2. Permutasikan ke tabel *initial permutation*
3. Hasil permutasi awal dibagi dua bagian, L dan R
4. *Enciphering* sebanyak 16 kali putaran
5. Hasil *enciphering* digabung menjadi satu
6. Permutasikan ke tabel IP^{-1}
7. *Get ciphertext*

DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Proses dekripsi DES terhadap *ciphertext* merupakan kebalikan dari proses enkripsi. Jika pada proses enkripsi urutan kunci ronde yang digunakan adalah mulai dari K_1 sampai K_{16}, maka pada proses dekripsi urutan kunci ronde dibalik mukai dari K_{16} sampai K_1.

Kemudian yang kedua adalah algoritma enkripsi *Vernam Cipher*. Berikut adalah *flowchart* dari algoritma *Vernam Cipher* :



Gambar 5. Flowchart Enkripsi Vernam Cipher

Algoritma dari *flowchart* enkripsi *Vernam Cipher* :

1. *Get plaintext*
2. *If plaintext.length == key.length* then
3. Ubah *plaintext* dan *key* ke biner
4. *Else*
5. Tambahkan " " sampai jumlah *plaintext* = kelipatan panjang *key*
6. Ubah *plaintext* dan *key* ke biner
7. *End if*

8. Jalankan skema *base64* pada biner
9. Membagi *plaintext* menjadi sama panjang dengan *key*
10. *Plaintext* di XOR dengan *key*
11. Gabungkan hasil XOR
12. Ubah ke *char* dengan *Decoding base64*
13. *Get ciphertext*

Algoritma yang ke-3 adalah *Diffie-Hellman* untuk pembangkitan kunci. Berikut adalah algoritma alur proses dari metode *Diffie-Hellman* :

1. Buat dua nilai prima n dan g, n<g
2. Buat dua nilai bilangan bulat acak x dan y
3. Hitung R1=g^x mod n
4. Hitung R2=g^y mod n
5. Hitung k1=$R2^x$ mod n
6. Hitung k2=$R1^y$ mod n
7. Buat nilai sa=k1.*toString()*
8. Buat nilai sb=k2.*toString()*
9. Hitung nilai jk=sa+sb

4. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan mengenai detail aplikasi, mulai dari tampilan layar, algoritma yang digunakan, dan juga hasil uji coba aplikasi.

4.1. Tampilan Layar

Berikut ini adalah rancangan layar menu utama dan menu informasi tes pada aplikasi.

4.1.1. Tampilan Menu Utama

Pada halaman ini terdapat tiga pilihan menu berupa tombol. Yang pertama tombol "PENDAFTARAN", yang kedua tombol "INFORMASI TES", dan yang ketiga adalah tombol "JADWAL ORDIK". Rancangan layar menu utama dapat dilihat pada gambar berikut ini :

Gambar 6. Tampilan Menu Utama

4.1.2. Tampilan Modul Informasi Tes

Pada bagian ini akan dijelaskan tampilan-tampilan dari alur proses untuk mengakses informasi tes ujian masuk. Yang pertama kali muncul jika kita menekan tombol "INFORMASI TES" pada menu utama adalah halaman berikut.

Gambar 7. Tampilan Menu Informasi Tes

Pada gambar diatas terdapat kolom isian untuk nomor pendaftaran peserta tes yang akan dilihat informasi tesnya. Yang bisa menggunakan modul ini hanyalah pendaftar yang menyetujui atau yang telah mendaftar untuk ikut tes. Pada awalnya, tombol "CEK INFO TES" tidak langsung muncul, melainkan kita harus menekan tombol *reCaptcha* untuk validasi bahwa *user* bukanlah robot atau program. Setelah validasi berhasil, maka tombol "CEK INFO TES" barulah muncul. Jika kolom sudah terisi dan kita *klik* selanjutnya, makan akan muncul halaman informasi tes yang sesuai dengan nomor pendaftaran yang tadi dimasukan.

Gambar 8. Tampilan Menu Informasi

Pada gambar di atas, ditampilkan beberapa info tes dari nomor pendaftaran yang dimasukkan pada halaman sebelumnya. Beberapa informasi yang ditampilkan antara lain nomor tes, jadwal tes (hari, tanggal, gedung,ruangan, dan jam), dan *id login (username* dan *password)*. Data-data tersebut diambil dari *database* aplikasi penerimaan mahasiswa baru. Jika kita *klik* tombol *Back,* maka kita akan kembali ke halaman menu utama.

4.2. Uji Coba Program

Pada bagian ini, akan dilakukan pembuktian atau pengujian enkripsi dan dekripsi data pada modul informasi tes. Pengujian ini bertujuan untuk membuktikan apakah data asli tidak mengalami perubahan setelah diolah dengan metode kriptografi yang diterapkan pada modul informasi tes.

4.2.1. Uji Coba Proses Enkripsi

Pertama kita akan melihat hasil enkripsi dari nomor pendaftaran yang telah dimasukan oleh *user* pada kolom masukan. Nomor pendaftarannya yaitu "C17100003". Hasil enkripsi yang pertama dengan algoritma *Vernam Cipher,* didapatkan sandi yaitu "DmV+BX1keQR+dGkUbXRpFA==". Berikut adalah buktinya :

Gambar 9. Scipt Hasil Enkripsi Vernam

Kemudian sandi yang telah didapat akan dienkripsi lagi dengan algoritma DES. Hasil enkripsi akhirnya yaitu "D8133A0145F967F1D1AADBB 3B4CB85DDD29E873741EAF14C".

Gambar 10. Scipt Hasil Enkripsi DES

4.2.2. Uji Coba Proses Dekripsi

Pada bagian ini akan diteliti apakah data yang ditampilkan pada halaman informasi tes sesuai dengan data sebenarnya yang ada pada database. Sebelum data dikirim dari *web service* ke Android, *web service* akan mengenkripsi data dengan metode yang sama. Jadi *ciphertext* yang diterima oleh Android adalah sebagai berikut :

Username : D8133A0145F967F1D1AADBB3B4C B85DDD29E873741EAF14C
Hari : A60E404009A35B6152CE41D9632D 83B9
Tanggal : C28B1B2785DFDCAB3B2896B6C24 366B1D29E873741EAF14C
Gedung : 5C183E1CD076B39452CE41D9632D 83B9
Ruangan : 5BF6A3A2D7989E0552CE41D9632D 83B9
Jam : D8C1192FC627F5BCD02AA625CD9 A0D84
Password : 47CB51CB050E037D52CE41D9632D 83B9
Grade : 5E92579B7292BC7F

Kemudian kita akan melihat data asli pada *database* untuk perbandingan apakah data yang diterima dan hasil dekripsi yang dilakukan di Android berhasil dan benar. Berikut adalah data yang terdapat pada *database*.

id	hari	tanggal	Gedung	Ruangan	Jam
1	Sabtu	2017-11-11	Sabuga	7.5.1	07:00:00

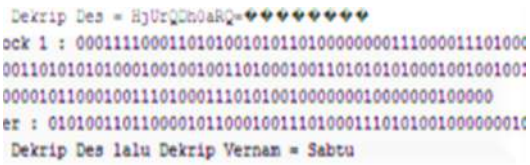
id	username	password	grade	id_test	id_daftar
1	C17100003	UILVEN	NULL	1	NULL

Gambar 11. Tampilan Database Informasi Tes

Berikut adalah beberapa gambar hasil dekripsi dari data-data informasi tes.

Gambar 12. Tampilan Hasil Dekripsi Username

Hasil dekripsi *username* : "C17100003".

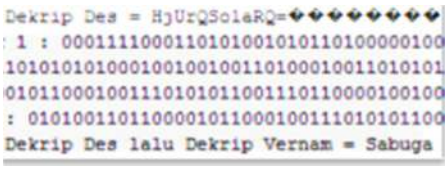


Gambar 13. Tampilan Hasil Dekripsi Hari

Hasil Dekripsi Hari : "Sabtu".

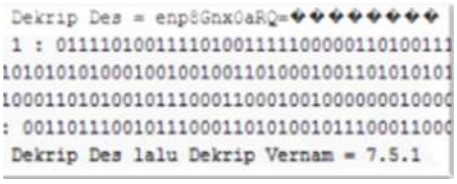
Gambar 14. Tampilan Hasil Dekripsi Tanggal

Hasil dekripsi tanggal : "2017-11-11".



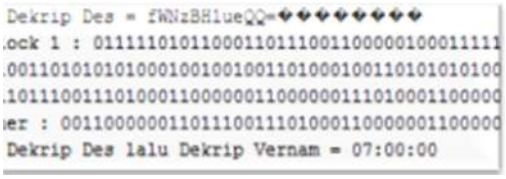
Gambar 15. Tampilan Hasil Dekripsi Gedung

Hasil dekripsi gedung : "Sabuga".



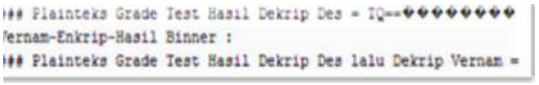
Gambar 16. Tampilan Hasil Dekripsi Ruangan

Hasil dekripsi ruangan : "7.5.1".



Gambar 17. Tampilan Hasil Dekripsi Jam

Hasil dekripsi jam : "07:00:00".



Gambar 18. Tampilan Hasil Dekripsi Grade

Hasil dekripsi *grade* : " " atau null.

Berdasarkan bukti-bukti percobaan di atas, dapat disimpulkan bahwa aplikasi mampu mengenkripsi dan mendekripsikan data dengan baik dan benar, dimana data dapat diubah ke *ciphertext* dan bisa didekripsi oleh *web service*. Sebaliknya, aplikasi ini juga mampu mendekripsikan data atau mengubah data ke bentuk semula dengan benar sesuai dengan data yang ada pada *database*. Informasi tes yang berisi *username,*hari, tanggal, gedung, ruangan, jam, dan *grade* terbukti benar dan sesuai dengan data aslinya pada *database*.

4.3. Tanggapan Pengguna

Penulis telah meminta beberapa responden untuk mencoba atau menggunakan aplikasi ini dan memberikan penilaian terhadap aplikasi melalui kuisioner yang telah penulis buat. Penilaian dibagi menjadi 4 aspek, yaitu *functionality* (kegunaan), *reliability* (kehandalan), *usability* (kemudahan penggunaan), dan *efficiency* (efisiensi). Berikut adalah rekapitulasi dari jawaban 25 orang responden pada kuisioner.

Tabel 1. Tabel Skor Akumulasi

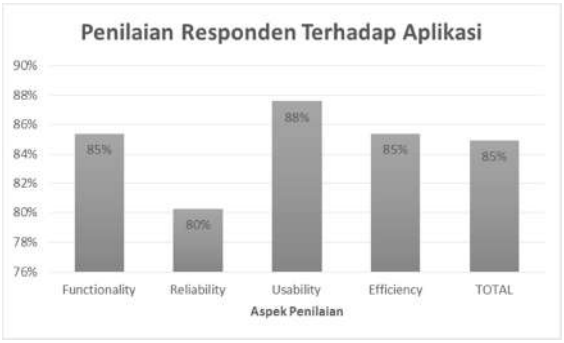
Aspek Penilaian	Skor Responden				
	5	4	3	2	1
	SS	S	R	TS	STS
Functionality	32	63	5	-	-
Reliability	17	44	12	2	-
Usability	40	58	2	-	-
Efficiency	31	65	4	-	-
Jumlah	**120**	**230**	**23**	**2**	**-**

Tabel 2. Tabel Skor Aktual

Aspek Penilaian	5	4	3	2	1	Total Skor Aktual	Skor Ideal	%
	SS	S	R	TS	STS			
Functionality	160	252	15	-	-	427	500	85%
Reliability	85	176	36	4	-	301	375	80%
Usability	200	232	6	-	-	438	500	88%
Efficiency	155	260	12	-	-	427	500	85%
Jumlah	600	920	69	4	-	1593	1827	85%

Keterangan :

- SS : Sangat Setuju
- S : Setuju
- R : Ragu-ragu
- TS : Tidak Setuju
- STS : Sangat Tidak Setuju



Gambar 19. Hasil Penilaian Responden

Berdasarkan hasil akumulasi jawaban kuesioner dari 25 responden, dapat dikatakan secara keseluruhan bahwa pengguna puas dengan aplikasi. Namun ada beberapa responden yang masih ragu

dan tidak setuju dengan kehandalan aplikasi ini. Mungkin saja kesalahan terletak pada proses *input* data (misalnya *typo* dalam mengisi kolom isian) atau mungkin juga koneksi internet yang kurang baik karena aplikasi ini sangat tergantung dengan koneksi internet. Bisa jadi memang ada *bug* yang ditemukan oleh responden namun belum ditemukan oleh penulis.

4.4. Evaluasi Program

Kelebihan Program :

a. Aplikasi ini dapat mengamankan data pendaftar/calon mahasiswa
b. Aplikasi ini sudah kompatibel dengan perangkat Android pada umumnya
c. Proses enkripsi dan dekripsi tidak memakan waktu yang lama
d. Aplikasi ini mudah dimengerti oleh pengguna

Kekurangan Program :

a. Efisiensi dalam mengakses data masih kurang, karena terdapat tombol *reCaptcha* yang cukup memakan waktu dalam proses validasinya
b. Masih ada responden yang mengalami *error*, jadi penulis harus mencari tahu kekurangan apa lagi yang masih belum penulis temukan pada aplikasi ini.

5. KESIMPULAN

Berdasarkan pada serangkaian uji coba pada aplikasi dan hasil tanggapan dari pengguna, maka dapat ditarik beberapa kesimpulan :

a. Aplikasi ini dapat mengamankan data pendaftar/peserta tes ketika data dikirim dan diterima
b. Aplikasi ini mempermudah pendaftar/calon mahasiswa untuk memperoleh informasi tes tanpa harus melihat papan pengumuman, jadi informasi tes bisa diakses kapan saja dan dimana saja
c. Aplikasi mampu melakukan enkripsi dan dekripsi data dengan benar.

6. DAFTAR PUSTAKA

[1] Abdala, P., 2017, Implementasi Algoritma Kriptografi *Vernam Cipher* dan Algoritma DES (Data Encryption Standard) pada Aplikasi *Chatting* Berbasis Android, Medan, Universitas Sumatera Utara.

[2] Kumar, R., dan C., Ravindranath C., 2015, *Analysis of Diffie Hellman Key Exchange Algorithm with Proposed Key Exchange Algorithm*, Bhopal, *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, 40-43.

[3] Muslim, I. A., Isnanto, R. R., & Widianto, E. D., 2015,,Perancangan dan Implementasi Algoritma DES untuk Mikroprosesor Enkripsi dan Dekripsi FPGA, *Jurnal Teknologi dan Sistem Komputer Vol. 3 No. 2*, 259-265.

[4] Samosir, R. A., 2015, Pengamanan Data Teks dengan Kombinasi Algoritma *Data Encryption Standard* (DES) dan *First of File* (FOF), Medan, Universitas Sumatera Utara.

[5] Sujiono, D. M., 2016, Implementasi *Three-Pass Protocol* dengan Kombinasi Algoritma *Beaufort Cipher* dan *One Time Pad* untuk Pengamanan Data, Medan, Universitas Sumatera Utara.

[6] A. Solichin and E. W. Ramadhan, "Enhancing data security using DES-based cryptography and DCT-based steganography," *2017 3rd International Conference on Science in Information Technology (ICSITech)*, Bandung, 2017, pp. 618-621.