Penerapan Algoritma DES, Vernam Cipher dan Diffie-Hellman untuk Mengamankan Data Pendaftaran Mahasiswa Baru pada Universitas Budi Luhur

Ahmad Ihsanudin[1)], **Achmad Solichin**[2)]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : ahmadihsan84@gmail.com[1)], achmatim@gmail.com[2)]

Abstrak

Perkembangan teknologi informasi saat ini sangat berpengaruh pada aspek kehidupan manusia. Dimana proses pertukaran data menjadi lebih cepat, efisien, dan akurat. Namun bila dilihat dari sisi keamanan dalam melakukan pertukaran data, hal ini masih kurang disadari karena adanya resiko pencurian data. Pada bidang pendidikan khususnya Universitas, dalam proses pendaftaran mahasiswa baru akan menimbulkan antrian di bagian registrasi mahasiswa. Dimana jika puluhan atau ratusan pendaftar datang pada hari yang sama maka tentu saja tidak bisa tertampung di dalam ruangan. Hal ini tentu perlu adanya solusi dalam mengantisipasinya sekaligus dengan adanya solusi dalam pengamanan data untuk menghindari pencurian data dari proses pendaftaran mahasiswa baru. Salah satu metode yang dapat digunakan adalah kriptografi. Pada penelitian ini menggunakan algoritma DES (Data Encryption Standart) dan Vernam Cipher dengan pembangkitan kunci algoritma Diffie-Hellman. Proses enkripsi dilakukan pada data pendaftaran mahasiswa yang dimasukkan dalam aplikasi sebelum dikirimkan ke server. Saat data sampai di server, data akan didekripsi dan disimpan ke database server. Sehingga data yang berjalan di jaringan berupa data yang tidak dimengerti (ciphertext). Hasil dari penelitian ini akan diimplementasikan dalam sebuah program aplikasi berbasis Android menggunakan bahasa pemograman Java Android. Dimana dapat membantu mengamankan data pendaftaran dengan mudah dan mengurangi resiko pencurian dan penyalahgunaan data.

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

1. PENDAHULUAN

Pada saat ini hampir di setiap kegiatan menggunakan teknologi, dimana data yang terlibat di dalam kegiatan menjadi terkomputerisasi. Apalagi dengan semakin cepat dan mudahnya masyarakat mendapatkan akses internet ini semakin menjadi bukti bahwa teknologi seakan bagian dari kehidupan masyarakat saat ini. Seperti yang kita ketahui bahwa pertukaran data atau informasi memang sangat menguntungkan dengan adanya perkembangan teknologi terutama internet, dimana pengiriman menjadi cepat, murah dan akurat. Akan tetapi muncul kelemahan baru atau hal yang menjadi kerugian pada pertukaran data atau informasi yang kita lakukan, salah satunya yaitu bisa terjadinya pencurian data dan penyalahgunaan data. Karena data yang kita kirim tersebut berjalan di jaringan internet, yang mana bisa saja sebelum sampainya ke tujuan tiba-tiba data tersebut terhenti ditengah jalan dan digunakan oleh orang yang tidak seharusnya. Oleh karena itu, untuk menutup atau meminimalisir adanya pencurian data pada saat pertukaran informasi, diterapkanlah suatu metode pengamanan data, salah satunya yaitu kriptografi.

Setiap Universitas pasti akan ada penerimaan mahasiswa baru terlebih khusus pada Universitas Budi Luhur, yang mana dalam proses administrasi pendaftarannya dilakukan dengan cara yang bermacam-macam. Biasanya pendaftaran dilakukan secara manual atau datang langsung ke bagian pendaftaran mahasiswa baru Universitas Budi Luhur. 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. Hal ini tentu perlu adanya solusi dalam mengantisipasi hal tersebut sekaligus dengan adanya solusi pengamanan data untuk menghindari pencurian data, yang mana agar tidak disebarluaskan untuk digunakan oleh pihak yang tidak bertanggung jawab.

Kriptografi sendiri merupakan seni dan ilmu untuk menjaga keamanan data yang mana masih dipercaya sebagai metode keamanan untuk mengamankan data dari pertukaran suatu informasi. Dimana data pendaftaran asli (*plaintext*) yang akan dikirim oleh calon mahasiswa ke penerima dalam hal ini *server* diubah (enkripsi) menjadi data yang tidak bisa dimengerti atau dikenali (*ciphertext*) terlebih dahulu, setelah itu baru dikirim data tersebut yang berbentuk *ciphertext* ke penerima (*server*). Adapun cara dalam pengiriman data ke *server* pada aplikasi Android ini menggunakan transmisi data berbentuk *JSON* (*JavaScript Object Notation*) dengan menggunakan *library Retrofit* yang tersedia pada Android. *Resource* yang berbentuk *JSON* ini disediakan oleh *REST API server* yang mana dapat dimanfaatkan oleh aplikasi Android dengan *library Retrofit*. *Retrofit* sendiri merupakan *library REST API Client* untuk Android dan Java dari *squareup*. Kemudian penerima (*server*) akan mengubah (dekripsi) data yang diterima (*ciphertext*) menjadi data asli (*plaintext*) semula, lalu disimpan data

aslinya di *database server*. Metode kriptografi yang diimplementasikan pada sistem pendaftaran *online* pada aplikasi berbasis Android ini menggunakan metode DES (*Data Encryption Standar*) dan *Vernam Cipher* dengan pembangkitan kunci menggunakan algoritma *Diffie-Hellman*.

Dari uraian di atas, penulis dapat merumuskan masalah yang akan dibahas dalam penelitian ini adalah bagaimana solusi untuk pengamanan data pendaftaran mahasiswa baru dari pencurian data pada program berbasis Android menggunakan algoritma DES (*Data Encryption Standard*) dan *Vernam Cipher* dengan pembangkitan kunci *Diffie-Hellman* ? lalu bagaimana solusi untuk mengembalikan data yang telah dienkripsi menjadi data asli tanpa mengalami perubahan data ?. Jika permasalahan tersebut dapat terselesaikan, maka akan sesuai dengan tujuan yang ingin dicapai dari penelitian tugas akhir ini adalah membuat suatu aplikasi pengamanan data berbasis Android menggunakan algoritma kriptografi algoritma DES (*Data Encryption Standard*) dan *Vernam Cipher* dengan pembangkitan kunci menggunakan algoritma *Diffie-Hellman* untuk mengamankan data pendaftaran mahasiswa baru yang dikirim dan diterima. Kedua mengamankan data pendaftaran yang dikirim atau diterima dari *server* melalui transmisi data berbentuk *JSON* agar tidak dapat diketahui oleh pihak yang tidak bertanggung jawab dalam aplikasi berbasis Android dengan menggunakan ilmu pengamanan data yaitu kriptografi, dengan algoritma DES (*Data Encryption Standard*) dan *Vernam Cipher* dengan pembangkitan kunci menggunakan algoritma *Diffie-Hellman*. Ketiga menghasilkan aplikasi pendaftaran mahasiswa baru berbasis Android yang aman dalam penggunaannya karena data yang dikirim terenkripsi.

Dalam penelitian ini metode peneliltian yang digunakan dengan model *Software Development Life Cycle* (SDLC) yaitu metode *Waterfall*. Tahapan SDLC dengan metode *Waterfall* meliputi tahapan perencanaan, analisis, desain, implementasi, pengujian dan pemeliharaan.

2. METODE PENELITIAN

2.1. Metode Penelitian

Berikut ini adalah rincian tahapan dalam pembuatan aplikasi pengamanan data pendaftaran mahasiswa baru berbasis Android yaitu:

- Pengumpulan Data

  Mengumpulkan kebutuhan dari keseluruhan elemen sistem yang akan diaplikasikan ke dalam bentuk *software* atau perangkat lunak dan mengumpulkan data mengenai elemen apa saja yang diisikan atau dipilih dalam pendaftaran mahasiswa baru serta proses enkripsi dan dekripsi menggunakan algoritma DES (*Data Encryption Standard*) dan *Vernam Cipher* dengan pembangkitan kunci menggunakan algoritma *Diffie-Hellman*.

- Menganalisa Kebutuhan Aplikasi

  Setelah memperoleh kebutuhan aplikasi kemudian dipelajari dan dianalisa mengenai fungsi-fungsi apa saja yang diperlukan untuk mengimplementasikan aplikasi ini.

- Desain atau Perancangan Aplikasi

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

- Pengkodean

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

- Implementasi

  Rancangan aplikasi yang sudah dibuat kemudian diimplementasikan berdasarkan analisa masalahnya.

- 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.

Setiap Universitas pasti akan ada penerimaan mahasiswa baru, terlebih khusus pada Universitas Budi Luhur. Dimana dalam proses administrasi pendaftarannya dilakukan dengan cara yang bermacam-macam. Biasanya pendaftaran dilakukan secara manual atau datang langsung ke bagian pendaftaran mahasiswa baru Universitas Budi Luhur. 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. Hal ini tentu perlu adanya solusi dalam mengantisipasi hal tersebut sekaligus dengan adanya solusi dalam pengamanan data untuk menghindari pencurian data dari proses pendaftaran mahasiswa baru, yang mana agar tidak disebarluaskan untuk digunakan oleh pihak yang tidak bertanggung jawab. Dari permasalahan di atas, maka perlu adanya sebuah aplikasi yang mana pendaftaran dapat dilakukan secara *online* lewat aplikasi pada *smartphone* Android, yang mana data pendaftaran tersebut akan disimpan oleh pihak lain yakni *server*. Karena data dikirim ke *server* atau dengan kata lain data akan terlempar ke suatu jaringan, maka kita butuh aplikasi yang berguna untuk menjaga sebuah kerahasiaan dan keamanan suatu data berupa data pendaftaran mahasiswa baru, sehingga data tersebut tidak akan bisa diambil oleh

pihak yang tidak mempunyai wewenang terhadap data tersebut. Aplikasi ini nantinya akan dapat mengubah sebuah data yang asli menjadi suatu data yang tidak dapat dibaca oleh siapapun dalam bentuk kata sandi yang diacak saat data berjalan atau dikirim di suatu jaringan menuju ke penerima atau dalam hal ini *server*, yang mana menggunakan layanan *web service.* Data yang dikirim berbentuk *JSON* (*JavaScript Object*) yang terenkripsi, pengiriman dilakukan dengan menggunakan *library Retrofit* yang tersedia pada Android. *Resource* yang berbentuk *JSON* ini disediakan oleh *REST API server* yang mana dapat dimanfaatkan oleh aplikasi Android dengan *library Retrofit.* Aplikasi ini dibuat berbasis Android dengan bahasa pemrograman Java. Pembuatan metode kriptografi dalam aplikasi ini menggunakan algoritma kriptografi DES (*Data Encryption Standar*) dan *Vernam Cipher* dengan pembangkitan kunci menggunakan algoritma *Diffie-Hellman.*

DES merupakan *block cipher* simetris yang beroperasi pada blok 64 bit yang menggunakan sebuah kunci 56 bit atau yang disebut dengan kunci internal (*internal key/sub key*). Kunci internal disini dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit [1]. *Vernam Cipher* merupakan algoritma yang juga berjenis *symetric key* yang artinya bahwa kunci yang digunakan untuk melakukan enkripsi dan dekripsi merupakan kunci yang sama. Dalam proses enkripsi, algoritma ini menggunakan cara *stream cipher* yang berasal dari hasi *XOR* antara bit *plaintext* dan bit *key* [2]. Sedangkan *Diffie-Hellman* adalah algoritma pertukaran kunci untuk memberi solusi atas pertukaran informasi secara rahasia [3]. Algoritma *Diffie-Hellman* ini tidak berdasarkan pada proses enkripsi dan dekripsi, melainkan lebih kepada proses matematika yang dilakukan untuk menghasilkan kunci rahasia yang dapat disebarkan secara bebas tanpa harus khawatir karena kunci rahasia tersebut hanya dapat didekripsi hanya oleh pengirim dan penerima pesan. Dasar dari algoritma ini adalah matematika dasar dari aljabar eksponen dan aritmatika modulus. Keamanan algoritma ini ditentukan oleh sulitnya menghitung logaritma diskrit.

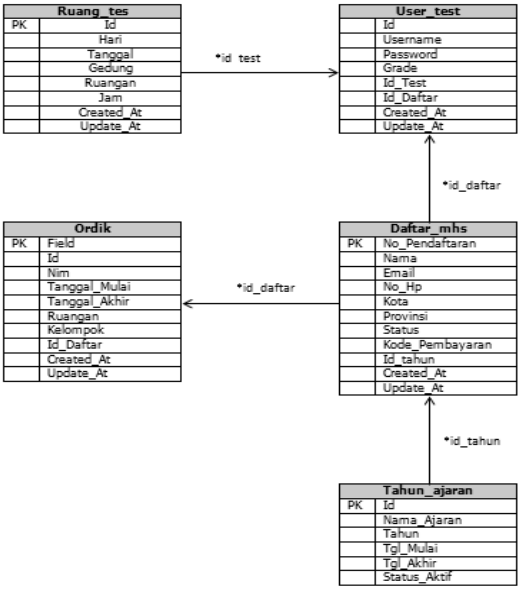
Dalam pengiriman data menggunakan *JSON*, *JSON* (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. *JSON* merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga *C* termasuk *C*, *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan *JSON* ideal sebagai bahasa pertukaran data [4]. Karena dalam aplikasi ini dalam melakukan pertukaran datanya menggunakan *JSON*, dimana aplikasi bertindak sebagai *client* yang dapat mengakses *server* lewat *Web API* yang dibuat. *Web API* adalah antar muka program dari sistem yang dapat diakses lewat *method* dan *header* pada protokol *HTTP* yang standar. *Web API* dapat diakses dari berbagai macam *HTTP client* seperti *browser* dan perangkat *mobile*. *Web API* juga memiliki keuntungan karena menggunakan infrastruktur yang juga digunakan oleh web terutama untuk penggunaan *caching* dan *concurrency* [5].

Dengan adanya sebuah aplikasi ini diharapkan mampu untuk memenuhi beberapa aspek keamanan sebuah data, sehingga data tersebut akan terjaga kerahasiaannya.

2.2. LRS (*Logical Record Structure*)

Berikut adalah representasi dari struktur *record-record* pada beberapa tabel yang terbentuk dari hasil relasi antar himpunan entitas pada database aplikasi ini :

2.3. Spesifikasi Basis Data



Gambar 1. LRS Aplilkasi

Berikut adalah rincian struktur dari tabel-tabel yang digunakan dalam *database* aplikasi ini :

Tabel 1. Daftar Mahasiswa

Field	Type	Length	Keterangan
no_pendaftaran*	Varchar	12	Nomor pendaftaran
nama	Varchar	50	Nama pendaftar
email	Varchar	50	*E-mail* pendaftar
no_Hp	Varchar	12	Nomor HP pendaftar
kota	Varchar	20	Kota tempat tinggal
provinsi	Varchar	20	Provinsi tempat tinggal
status	Varchar	15	Status pendaftar
kode_ pembayaran	Varchar	12	Kode untuk pembayaran
id_tahun	Char	3	Id tahun ajaran
created_at	Date Time	8	Tanggal pertama kali data dibuat
update_at	Date Time	8	Tanggal terakhir di-*update*

Tabel 2. Ruang Test

Field	Type	Length	Keterangan
id*	Varchar	12	Id tes
hari	Varchar	50	Hari pelaksanaan tes
tanggal	Varchar	50	Tanggal pelaksanaan tes
gedung	Varchar	12	Gedung tempat tes digelar
ruangan	Varchar	20	Ruangan tes dilaksanakan
jam	Varchar	4	Waktu pelaksanaan tes
created_at	Date Time	8	Tanggal pertama kali data dibuat
update_at	Date Time	8	Tanggal terakhir data di-update

Tabel 3. Ordik

Field	Type	Length	Keterangan
id*	Varchar	10	Id ORDIK
nim	Varchar	10	Nomor Induk Mahasiswa
tanggal_mulai	Date	3	Tanggal mulai ORDIK
tanggal_akhir	Date	3	Tanggal berakhirnya ORDIK
ruangan	Varchar	10	Ruangan kelas kelompok ORDIK
kelompok	Varchar	20	Kelompok ORDIK
id_daftar	Int	9	id pendaftaran
created_at	Date Time	8	Tanggal pertama kali data dibuat
update_at	Date Time	8	Tanggal terakhir di-*update*

Tabel 4. Tahun Ajaran

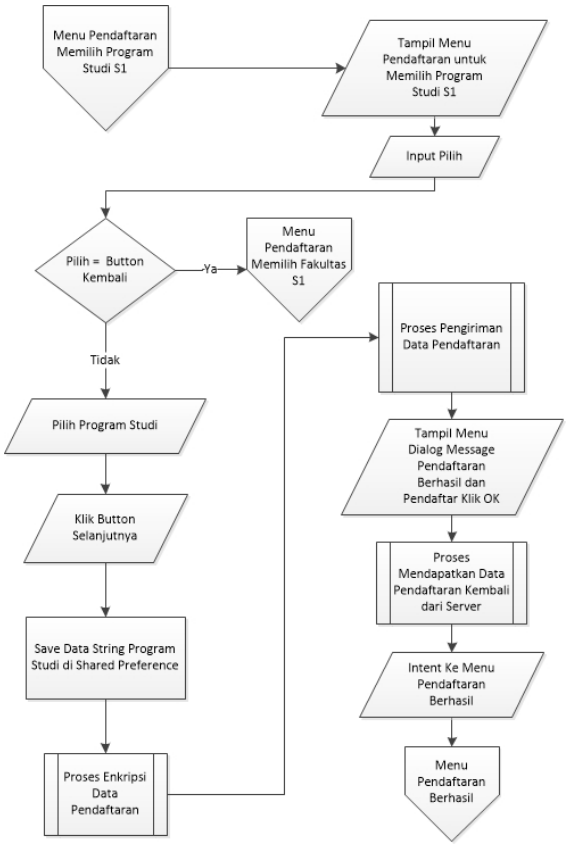
Field	Type	Length	Keterangan
id*	Varchar	11	Id Tahun Ajaran
nama_ajaran	Varchar	20	Genap / Ganjil
tahun	Varchar	20	Tahun Ajaran
tgl_mulai	Date	3	Tanggal Mulai Tahun Ajaran
tgl_akhir	Date	3	Tanggal Akhir Tahun Ajaran
status_aktif	Varchar	2	Status Aktif Mahasiswa
created_at	Date Time	8	Tanggal pertama kali data dibuat
update_at	Date Time	8	Tanggal terakhir di-*update*

Tabel 5. User Test

Field	Type	Length	Keterangan
id*	Varchar	10	Id *user test*
username	Varchar	10	*Username* untuk *login* saat tes
password	Varchar	9	*Password* untuk *login* saat tes
grade	Varchar	2	Nilai hasil ujian
id_test	Int	8	Id Tes
id_daftar	Int	9	Id pendaftaran
created_at	Date Time	8	Tanggal pertama kali data dibuat
update_at	Date Time	8	Tanggal terakhir data di-*update*

2.4. Flowchart dan Algoritma Program

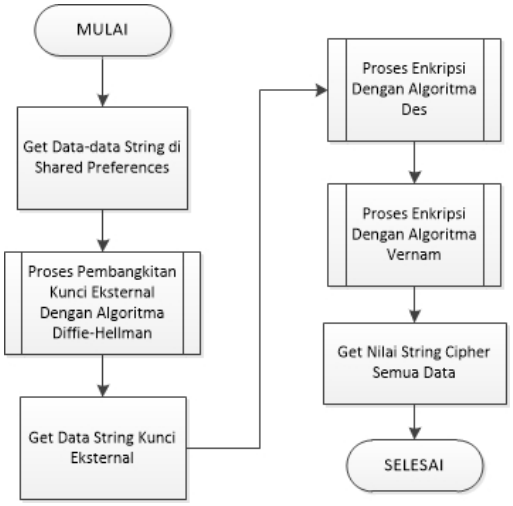
Berikut ini merupakan rancangan *flowchart* dan algoritma pemrograman. Terdiri dari Menu Utama dan Menu Pendaftaran yang berisi inputan untuk mengisi nama, email, provinsi, kota, nomor hp, jenjang, fakultas dan program studi serta *flowchart* dan Algoritma untuk proses enkripsi dan dekripsi data pendaftaran yang dikirim dan diterima. Akan tetapi, disini hanya menampilkan yang pokok saja, berikut *flowchart* dan algoritmanya :



Gambar 2. Flowchart Menu Pendaftaran Memilih Program Studi S1

Algoritma proses dari *flowchart* di atas :

1. Tampil Menu Pendaftaran untuk Memilih Program Studi S1
2. Input Pilih
3. If Pilih = Button Kembali Then
4. Pindah ke Menu Pendaftaran Memilih Fakultas S1
5. Else
6. Pilih Program Studi yang Dipilih
7. Klik Button Selanjutnya
8. Save Data String Program Studi di Shared Preference
9. Proses Enkripsi Data Pendaftaran
10. Proses Pengiriman Data Pendaftaran
11. Tampil Menu Dialog Message Pendaftaran Berhasil
12. Pendaftar Klik OK
13. Proses Mendapatkan Data Pendaftaran Kembali dari Server
14. Intent Ke Menu Pendaftaran Berhasil
15. Pindah ke Menu Pendaftaran Berhasil



Gambar 3. Flowchart Proses Enkripsi Data Pendaftaran

Algoritma proses dari *flowchart* di atas :

1. Mulai
2. Get Semua Data String di Shared Preferences yang Ingin Dikirim
3. Proses Pembangkitan Kunci Eksternal Dengan Algoritma Diffie-Hellman
4. Get Data String Kunci Eksternal
5. Proses Enkripsi Dengan Algoritma DES
6. Proses Enkripsi Dengan Algoritma Vernam Cipher
7. Get Nilai String Cipher untuk Semua Data Yang Ingin Dikirim Selesai

Algoritma pada proses enkripsi dengan algoritma DES :

1. Mulai
2. Get Plainteks & Key
3. Proses Buat Kunci Internal Des dengan Mengirim Variabel Key
4. Buat binnaryValueMessage = textToBinnary(Plainteks);
5. Buat messageGroup = GroupingTeks(binnaryValueMessage) / 64;
6. Enkripsi Pada 64 bit Sebanyak Group yang Ada
7. Buat tempTrueMessage = messageGroup[x].toCharArray();
8. Buat binnaryMessageIP[i] = tempTrueMessage[ip [i]];
   Bagi 2 :
   l[0][i] = binnaryMessageIP[i];
   r[0][i-32] = binnaryMessageIP[i];
9. Lakukan 16 kali perulangan untuk mendapatkan L16 dan R16
10. bitselection, Nilai ERi = Ri & Nilai Li <- Li+1 = Ri
11. XOR ERi dengan Key yang Ke i -> Nilai *A*
12. Buat Grouping Nilai A per 6 bit
13. Lakukan sbox pada tiap group, dapat s1-s8. -> Nilai *B*
14. Nilai *B* Kenakan Permutasi Dengan tabel P -> Nilai P(B)
15. XOR Li dengan P(B), -> Nilai Ri yang Selanjutnya
16. Lakukan InversIndexPermutation, -> Nilai encryptedMessage
17. Buat encryptedMessage = encryptedMessage.concat (BinnaryToChar (encryptedBinMessage));
18. Gabungkan dan Ubah = aschiiToHex (String.*valueOf* BinnaryToChar (encryptedBinMessage)));
19. return aschiiToHex(encryptedMessage);
20. Selesai

Algoritma pada proses Enkripsi dengan algoritma *Vernam Cipher* :

1. Mulai
2. Get Data Plainteks & Key
3. If Plaintext.length = Key.length Then
4. Buat binaryMessage = *textToBinnary*(plaintext); & Buat binaryKey = *textToBinnary*(key);
5. Buat [] binaryKeyArray = binaryKey.toCharArray();
6. Buat [] groupMessage = *GroupingTeks*(binaryMessage, binaryKey);
7. Buat [] binaryMessageArray = groupMessage[i].toCharArray();
8. Buat [] xor = new [binaryMessageArray.length];
9. XOR(binaryMessageArray, binaryKeyArray, xor);
10. Buat Nilai tempCipher = tempCipher .concat(String.*valueOf*(xor))
11. Ubah tempCipher = *BinnaryToChar*(tempCipher).trim();
12. Buat tempCipher = encodedString(tempCipher);
13. return *encodedString*(*BinnaryToChar*(tempCipher)).trim();
14. Else
15. Plaintext = Plaintext.concat(" ");
16. Masuk ke Proses nomor 4.
17. Selesai

Algoritma proses pembangkitan kunci dengan algoritma *Diffie-Hellman* :

1. Mulai
2. Buat Nilai n = Bilangan Prima
3. Buat Nilai g = Bilangan Prima dimana g<n
4. Buat Nilai x = Bilangan Bulat Acak
5. Buat Nilai y = Bilangan Bulat Acak
6. Hitung $R1=g^x \bmod n$
7. Hitung $R2=g^y \bmod n$
8. Hitung $k1=R2^x \bmod n$
9. Hitung $k2=R1^y \bmod n$
10. Buat Nilai sa = k1.toString();
11. Buat Nilai sb = k2.toString();
12. Buat Nilai jk = sa+sb;
13. Buat Nilai message = jk;
14. Buat Nilai bytes = message.getBytes("UTF-8");
15. Ubah Nilai message = Bentuk Biner
16. Buat Nilai decodeKey = Nilai bytes.Base64.encodeToString();
17. return decodeKey;
18. Selesai

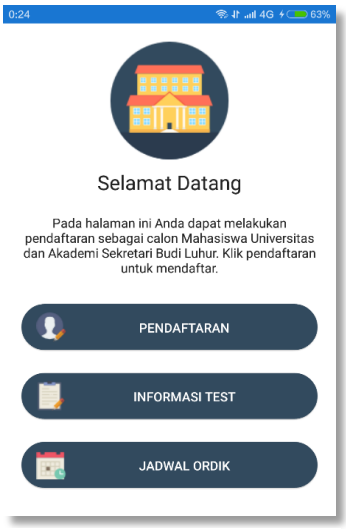
3. HASIL DAN PEMBAHASAN

3.1. Tampilan Layar

Disini akan dijelaskan mengenai tampilan-tampilan layar dari berbagai proses yang terjadi di aplikasi PMB (Penerimaan Mahasiswa Baru). Berikut adalah beberapa tampilan dari aplikasi ini :

3.1.1. Tampilan Menu Utama

Pada halaman ini terdapat tiga pilihan menu berupa *button*. Yang pertama *button* "PENDAFTARAN", yang kedua *button* "INFORMASI TES", dan yang ketiga adalah *button* "JADWAL ORDIK". Tampilan layar pada Menu Utama dapat dilihat pada gambar berikut ini :



Gambar 4. Tampilan Menu Utama

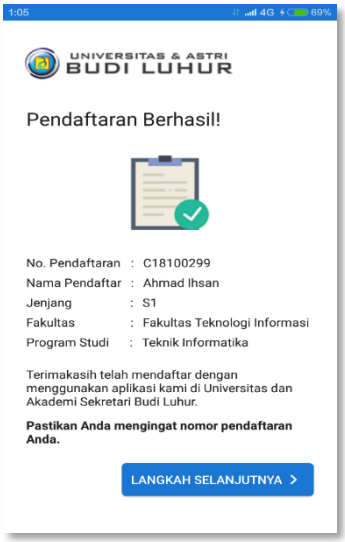
3.1.2. Tampilan Menu Pendaftaran

Gambar 5. Tampilan Menu Pendaftaran Step 1

Gambar 6. Tampilan Menu Pendaftaran Step 4

Gambar 7. Tampilan Menu Pendaftaran Step 5

3.1.3. Tampilan Menu Pendaftaran Berhasil



Gambar 8. Tampilan Menu Pendaftaran Berhasil

3.2. Uji Coba Program

Pada bagian ini, akan dilakukan pembuktian atau pengujian enkripsi dan dekripsi pada data pendaftaran yang akan dikirimkan ke *server* ataupun diterima dari *server*. Pengujian ini bertujuan untuk membuktikan apakah data asli tidak mengalami perubahan setelah diolah dengan metode kriptografi yang diterapkan.

3.2.1. Uji Coba Proses Enkripsi

Berikut contoh data pendaftaran yang akan dienkripsi :

Nama : Fitrah Semesta
Email : fitrahsemesta12@gmail.com
Provinsi : BANTEN
Kota : KOTA TANGERANG
Nomor Hp : 081234567891
Jenjang : S1
Fakultas : Fakultas Teknologi Informasi
Program Studi : Teknik Informatika
Status : calon

Berikut adalah *script* dari Android Monitor pada aplikasi Android Studio dalam proses enkripsi data pendaftaran yang akan dikirim. Di dalamnya juga terdapat *script* dari fungsi *POST* yang berarti data dikirim dari aplikasi Android ke *web service*. Tibanya data di *server* maka data akan didekripsi dengan metode yang sama tetapi terbalik, yakni *Vernam Cipher* dulu baru DES agar kembali ke bentuk semula supaya bisa terbaca. Adapun untuk merangkum data plainteks beserta hasil ciphernya, penulis memberikan hasil *script* gambar sebagai berikut :

Gambar 9. Tampilan Script Keseluruhan Hasil Enkripsi Data Pendaftaran

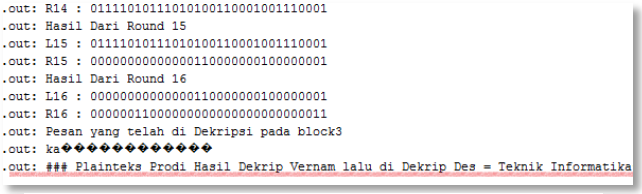
Tabel 6. Keseluruhan Hasil Enkripsi Contoh Data Pendaftaran

Plainteks	Cipherteks
Fitrah Semesta	CRZ/DX1hcQUIEAh3eRF8cXpsfgEJbH9xDxV+A3VmCgU=
fitrahsemesta12@gmail.com	C2R/BQ5scAx0YH8He2R9cAxhC3J6EQwDDmJ4BXljeQYPFQ0BeWN4BX4VCHF4F393fhJ9DXsSDAYPYXFyDGd4AA==
BANTEN	eGVxDQgSeHZ8EXBwD2EPAA==
KOTA TANGERANG	C2BxAA9tcXd4Fg92fmR7cQtne3cMbHgDDhB9cXtlfQw=
081234567891	e2ELB31kcQ10YXEGfGR+BX5lCg18Fg0BeBYNBHRhenc=
S1	DmZ4dglmegMIEXB3DxELBg==
Fakultas Teknologi Informasi	DGAKcQxmCnYLZnx1C2UNdg4Xf3B7YQxyfGN5AAxnfQV0Z3AFfmZ+B3sVC3AMZXhyD214dg8RfwV+EXoAfREIAQ==
Teknik Informatika	CWANBX5mfAYIEQ0AdWB+BQkSeAV+FXB3eGMLDHwRcXALEH0Aemx9A3wWfHB7F3lx
calon	DmNwcQ8Vf3d4Y3ENfRULAg==

3.2.2. Uji Coba Proses Dekripsi

Pada bagian ini akan diteliti apakah data yang ditampilkan pada halaman Menu Pendaftaran Berhasil sesuai dengan data sebenarnya yang ada pada *database* atau data pendaftaran plainteks asli. Adapun data pendaftaran yang dikirim balik itu ada nomor pendaftaran, nama, jenjang, fakultas dan program studi. 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 :

Nomor Pendaftaran :	dGJ5BXplDAAJEQwBCBB5cHRlC3d1FnAFf2B7BnplfwY=
Nama :	CRZ/DX1hcQUIEAh3eRF8cXpsfgEJbH9xDxV+A3VmCgU=
Jenjang :	DmZ4dglmegMIEXB3DxELBg==
Fakultas :	DGAKcQxmCnYLZnx1C2UNdg4Xf3B7YQxyfGN5AaxnfQV0Z3AFfmZ+B3sVC3AMZXhyD214dg8RfwV+EXoAfREIAQ==
Program Studi :	CWANBX5mfAYIEQ0AdWB+BQkSeAV+FXB3eGMLDHwRcXALEH0Aemx9A3wWfHB7F3lx



Gambar 10. Tampilan Script Hasil Dekripsi Pada Cipher Program Studi

Pada gambar di atas, dapat dibuktikan bahwa proses dekripsi terhadap data *cipher* Program Studi bisa dikembalikan seperti plainteks asli semula, yaitu "Teknik Informatika".

Berdasarkan percobaan serta analisis proses enkripsi dan dekripsi diatas, terbukti bahwa tidak ada perubahan data yang diolah dengan metode kriptigrafi yang digunakan. Dengan demikian bisa dikatakan bahwa tidak ada *error* di dalam proses enkripsi dan dekirpsi. Begitu juga dengan data dari *web service*, terbukti bahwa tidak ada perbedaan dari algoritma yang digunakan dan tidak ada *error* pada proses enkripsi dan dekripsi.

3.3. Tanggapan Pengguna Lewat Kuesioner

Penulis telah meminta beberapa responden untuk mencoba menggunakan aplikasi ini dan memberikan tanggapannya di dalam kuesioner yang telah penulis buat. Penulis membagi beberapa pertanyaan penilaian menjadi 4 kelompok, yaitu *functionality* (kegunaan), *reliability* (kehandalan),

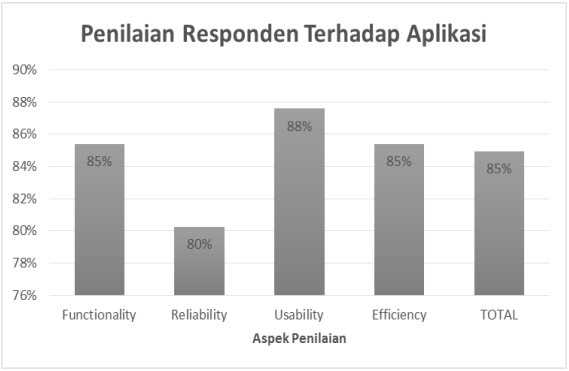
usability (kemudahan penggunaan), dan *efficiency* (efisiensi). Berikut adalah hasil dari kuesioner yang telah diisi oleh 25 orang responden.

Tabel 7. Tabel Skor Responden

Aspek Penilaian	Skor Responden				
	5	4	3	2	1
	SS	S	R	TS	STS
Functionality	33	13	8	3	3
Reliability	37	13	7	3	-
Usability	33	17	3	5	2
Efficiency	31	18	5	4	2
Jumlah	**134**	**61**	**23**	**15**	**7**

Tabel 8. Tabel Skor Aktual

Aspek Penilaian	Skor Aktual					Total Skor Aktual	Skor Ideal	%
	5	4	3	2	1			
	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%**



Gambar 11. Diagram Akumulasi Penilaian Responden

3.4. Evaluasi Program

Kelebihan Program :

- Aplikasi ini dapat mengamankan data pendaftar/calon mahasiswa.
- Aplikasi ini sudah kompatibel dengan perangkat Android pada umumnya terlebih khusus minimal spesifikasi OS Android *Lollipop*.
- Proses enkripsi dan dekripsi tidak memakan waktu yang lama.
- Aplikasi ini mudah dimengerti oleh pengguna.
- Dapat melakukan pendaftaran dimana saja dan kapan saja asalkan mempunyai koneksi internet.

Kekurangan Program :

- Efisiensi dalam mengakses data masih kurang, karena terdapat tombol *reCaptcha* yang cukup memakan waktu dalam proses validasinya.
- Masih ada responden yang mengalami *error*, jadi penulis harus mencari tahu kekurangan yang masih ada dalam aplikasi ini.
- Beberapa smartphone bisa saja mengalami crash saat menjalani aplikasi ini. Dikarenakan dibawah spesifikasi OS Android *Lollipop*.

4. KESIMPULAN

Pada saat proses pembuatan dan pengujian dalam Tugas Akhir ini, berdasarkan perancangan, pembuatan, serangkaian uji coba dan analisis program dapat ditarik kesimpulan, yaitu :

- Dengan adanya aplikasi ini, pendaftaran mahasiswa baru dapat dengan mudah dilakukan kapan dan dimana saja.
- Aplikasi ini mampu mengenkripsi dan mendekripsi data secara tepat.
- Dengan menggunakan aplikasi ini, data pendaftaran yang telah dienkripsi menjadi aman saat berjalan di jaringan karena sulit untuk dipahami.
- Pada proses enkripsi, data dengan ukuran yang lebih banyak karakternya akan membuat hasil enkripsi yang lebih panjang.

5. DAFTAR PUSTAKA

[1] Abdala, P., Budiman, M. A. and Herriyance, H. (2017) 'Implementasi Algoritma Kriptografi Vernam Cipher dan DES (Data Encryption Standard) pada Aplikasi Chatting berbasis Android', *Jurnal Ilmiah CORE IT*, 5(1), pp. 1–19. Available at: http://core-it.org/index.php/coreit/article/view/27.

[2] Jumeidi, M., Triyanto, D. and Brianorman, Y. (2016) 'IMPLEMENTASI ALGORITMA KRIPTOGRAFI VERNAM CIPHER BERBASIS FPGA Jl . Prof . Dr . H . Hadari Nawawi , Pontianak Abstrak Keamanan data menjadi salah satu isu penting dalam perkembangan teknologi informasi saat ini . Salah satu cara yang dapat dilakuan untuk', *Jurnal Coding, Sistem Komputer UNTAN*, 4(1), pp. 21–32.

[3] Purwadi, Jaya, H. and Calam, A. (2014) 'Aplikasi Kriptografi Asimetris Dengan Metode Diffie-Hellman Dan Algoritma Elgamal Untuk Keamanan Teks', *Jurnal Ilmiah SAINTIKOM (Sains dan Komputer)*, 13(3), pp. 183–196.

[4] Ferryansyah, M. S., Ananta, M. T. and Fanani, L. (2017) 'Analisis Performansi HTTP Networking Library pada Android (Studi Kasus : Portal Berita)', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(5), pp. 2025–2033.

[5] Kurniawan, E. (2014) 'Implementasi Rest Web Service Untuk Sales Order Dan Sales Tracking Berbasis Mobile', *Jurnal EKSIS*, 7, pp. 1–12.