APLIKASI PENYANDIAN PESAN TEKS BERBASIS WEB MENGGUNAKAN ALGORITMA BLOWFISH

SKRIPSI

Oleh :

NELLY PERMATASARI
NPM. 18020015

PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU

2023

APLIKASI PENYANDIAN PESAN TEKS BERBASIS WEB MENGGUNAKAN ALGORITMA BLOWFISH

SKRIPSI

Oleh :

NELLY PERMATASARI
NPM. 18020015

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata I
Pada Program Studi Informatika Universitas Dehasen Bengkulu

PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DEHASEN
BENGKULU
2023

APLIKASI PENYANDIAN PESAN TEKS BERBASIS WEB MENGGUNAKAN ALGORITMA BLOWFISH

SKRIPSI

Disusun Oleh :

NELLY PERMATASARI
NPM. 18020015

DISETUJUI OLEH :

Pembimbing Utama	Pembimbing Pendamping
Riska, S.Kom., M.Kom	**Yessi Mardiana, S.Kom., M.Kom**
NIDN. 02.240192.01	**NIDN. 02.030288.02**

Mengetahui,
Ketua Program Studi
Rekayasa Sistem Komputer

Toibah Umi Kalsum, S.Kom., M.Kom
NIDN. 02.060573.01

APLIKASI PENYANDIAN PESAN TEKS BERBASIS WEB MENGGUNAKAN ALGORITMA BLOWFISH

SKRIPSI

Disusun Oleh :

NELLY PERMATASARI
NPM. 18020015

Telah Dipertahankan di depan TIM Penguji
Universitas Dehasen Bengkulu

Hari : Sabtu
Tanggal : 17 Juni 2023

Skripsi ini telah diperiksa dan disetujui oleh TIM Penguji:

Penguji	Nama	NIDN	Tanda Tangan
Ketua Penguji	Riska, S.Kom., M.Kom	02.240192.01	
Anggota I	Yessi Mardiana, S.Kom., M.Kom	02.030288.02	
Anggota II	Toibah Umi Kalsum, S.Kom,M.Kom	02.060573.01	
Anggota III	Hendri Alamsyah, S.Kom., M.Kom	02.110391.01	

Mengetahui,

Dekan
Fakultas Ilmu Komputer

H. Siswanto, S.E., S.Kom., M.Kom
NIDN : 02.240363.01

DAFTAR RIWAYAT HIDUP

Penulis bernama Nelly Permata Sari, dilahirkan di Desa Tebing, Kec. Air Natal Kabupaten Bengkulu Utara, Provinsi Bengkulu, pada tanggal 3 Januari 2001, anak Kedua dari tiga bersaudara, Ayah bernama Jusmin Gultom, S.PdK dan Ibu bernama Hendri Sitanggang. Menyelesaikan pendidikan di Sekolah Dasar Negeri (SDN) 07 Air Napal pada tahun 2012, kemudian penulis melanjutkan pendidikan pada SMP N 1 Lais selesai pada tahun 2015 dan selanjutnya meyelesaikan pendidikan SMA Negeri 1 Lais Tahun 2018. Kemudian Penulis melanjutkan pendidikan ke perguruan tinggi yaitu pada Universitas Dehasen (UNIVED) Bengkulu dengan mengambil Jurusan Rekayasa Sistem Komputer pada Fakultas Ilmu Komputer, untuk jenjang Strata Satu (S-1) pada tahun 2018.

MOTTO DAN PERSEMBAHAN

MOTTO :

Apabila kamu ingin meraih kebahagiaan, jangan bergantung kepada orang lain atau benda. Fokuslah pada tujuanmu karena tujuan tidak berubah-ubah seperti manusia.

PERSEMBAHAN

Dengan mengucapkan Alhamdulillah atas semua limpahan rahmat dan Kasih sayangMu akhirnya tercapai juga suatu amanah, kewajiban, tujuan dan cita-cita. Kuyakin ini bukanlah akhir dari perjalanan dan perjuanganku, namun langkah awal untuk mewujudkan mimpi dan membahagiakan orang-orang yang ku kasihi dan mengasihiku. Ku persembahkan karya kecil ini dengan sepenuh cinta untuk:

- *Kedua orang tuaku, Ayah dan Ibuku tercinta,*
- *Keluargaku serta adik-addikku yang sangat ku sayangi,*
- *Kedua Dosen pembimbing,*
- *Teman-teman seperjuanganku,*
- *Serta Almamater tercinta*

SURAT PERNYATAAN ORSINILITAS & PERSETUJUAN PUBLIKASI AKADEMI SKRIPSI

Yang bertanda tangan di bawah ini:

Nama	:	Nelly Permatasari
NPM	:	18020015
Program Studi	:	Rekayasa Sistem Komputer
Fakultas	:	Ilmu Komputer
Tempat/Tgl Lahir	:	Desa Tebing Kandang/ 03 Januari 2001
Alamat	:	Jl. Meranti3- Bengkulu

Dengan ini menyatakan dengan sesungguhnya bahwa SKRIPSI dengan judul :

APLIKASI PENYANDIAN PESAN TEKS BERBASIS WEB MENGGUNAKAN ALGORITMA BLOWFISH

1. Adalah benar dibuat oleh saya sendiri untuk memenuhi persyaratan kelulusan akademi.
2. Pada bagian-bagian tertentu dalam penulisan skripsi yang saya kutip dari hasil karya orang lain telah ditulis sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah.
3. Jika dikemudian hari diketahui bukti berdasarkan bukti-bukti yang kuat ternyata skripsi tersebut dibuat oleh orang lain atau diketahui bahwa skripsi tersebut merupakan plagiat/mencontek/menjiplak hasil karya ilmiah orang lain, maka dengan ini saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya sesuai dengan peraturan yang berlaku.
4. Dan atas pernyataan orisinilitas tersebut di atas, maka saya menyetujui untuk memberi kepada Universitas Dehasen Bengkulu atas bebas royalti non ekslusif untuk menyimpan, mengalih mediakan. Mendstribusikan dan mempublikasikan skripsi saya tanpa perlu meminta izin, selama mencantumkan nama saya sebagai penulis.
5. Saya bersedia menanggung secara pribadi tanpa melibatkan pihak Universitas Dehasen Bengkulu segala bentuk tuntukan hukum yang ditimbul atas pelanggaran hak cipta dalam karya ilmiyah saya ini.

Demikian surat pernyataan ini dibuat untuk dipergunakan sebagaimana mestinya.

Bengkulu, 10 Juni 2023

Hormat saya,

(Nelly Permatasari)

NPM. 18020015

ABSTRAK

APLIKASI PENYANDIAN PESAN TEKS BERBASIS WEB MENGGUNAKAN ALGORITMA BLOWFISH

Oleh:
Nelly Permatasari[1]
Riska, S.Kom., M.Kom[2]
Yessi Mardiana, S.Kom., M.Kom[2]

Algoritma Blowfish merupakan salah satu sistem penyadian yang ada. Penyadian bertujuan untuk membuat sebuah pesan menjadi lebih terjamin kerahasiaannya. Algoritma blowfish merupakan salah satu algoritma kriptografi modern yang menggunakan kunci simetris dalam proses enkripsi dan dekripsi. Penerapan algoritma blowfish dalam penelitian ini yakni untuk mengamankan pesan teks antara pengirim dan penerima pesan teks tersebut melalui aplikasi berbasis web. Kriptografi merupakan seni dan ilmu untuk memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditujukan untuk orang yang hanya memiliki sebuah kunci untuk mengubah kode itu kembali yang berfungsi dalam menjaga kerahasiaan data atau pesan. Aplikasi pengiriman pesan dengan menggunakan algoritma blowfish dibuat dengan menggunakan Bahasa pemrograman PHP dan database MySQL. Aplikasi penyadian pesan teks berbasis web ini terdiri dari beberapa menu, diantaranya menu login, daftar, pesan masuk dan kirim pesan. Aplikasi penyadian pesan teks berbasis web berjalan dengan baik sesuai dengan masing-masing menu yang ada.

Kata Kunci: **Blowfish, PHP dan MySQL**

Keterangan :
1: Peneliti
2: Pembimbing 1 dan Pembimbing 2

ABSTRACT

WEB-BASED TEXT MESSAGE ENCORDING APPLICATION USING BLOWFISH ALGORITHM

By:
Nelly Permatasari[1]
Riska[2]
Yessi Mardiana[2]

The Blowfish algorithm is one of the existing encoding systems. Encoding aims to make a message more secure for its confidentiality. The blowfish algorithm is one of the modern cryptographic algorithms that uses symmetric keys in the encryption and decryption process. The application of the blowfish algorithm in this study is to secure text messages between the sender and recipient of the text message through a web-based application. Cryptography is the art and science of protecting data transmission by converting it into a certain code and is only intended for people who only have a key to change the code back which functions to maintain the confidentiality of data or messages. Applications for sending messages using the blowfish algorithm are made using the PHP programming language and MySQL database. This web-based text message encoding application consists of several menus, including login menus, lists, incoming messages and sending messages. The web-based text message encoding application runs well according to each existing menu.

Keywords: Blowfish,PHP and MySQL

Information :
1: Student
2: Supervisors

Arsip Abstract Untuk Program Studi, dikeluarkan dan ditejemahkan oleh:
Tim Penerjemah UPT Bahasa Inggris UNIVERSITAS DEHASEN BENGKULU

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa berkat Rahmat, kepada kita semua sehingga penulis dapat menyelesaikan skripsi ini dengan judul **Aplikasi Penyandian Teks Berbasis Web Menggunakan Algoritma Blowfish** dapat diselesaikan dengan baik

Penulis menyadari dalam penyusunan skripsi ini tidak akan selesai tanpa bantuan dari berbagai pihak. Karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Bapak Siswanto, SE, M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
2. Ibu Toibah Umi Kalsum, M.Kom selaku Ketua Program Studi Reka Informatika Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.
3. Bapak Riska, S.Kom., M.Kom selaku Dosen Pembimbing I yang telah memberikan masukan serta arahan yang membangun dalam pembuatan skripsi ini.
4. Ibu Yessi Mardiana, S.Kom., M.Kom selaku Dosen Pembimbing II yang telah memberikan masukan serta arahan yang membangun dalam pembuatan skripsi ini.
5. Teman-teman yang telah berjuang bersama-sama
6. Berbagai pihak yang telah banyak membantu dalam penyusunan skripsi ini.

Penulis juga menyadari sepenuhnya bahwa di dalam skripsi ini terdapat kekurangan dan jauh dari kata sempurna. Oleh sebab itu, kami berharap adanya kritik, saran dan usulan demi perbaikan skripsi yang telah kami buat di masa yang

akan datang, mengingat tidak ada sesuatu yang sempurna tanpa saran yang membangun.

Bengkulu, September 2023

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR TABEL

DAFTAR LAMPIRAN

Lampiran

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dunia teknologi informasi sekarang ini berkembang sangat pesat dan mempengaruhi hampir seluruh aspek kehidupan manusia. Perkembangan tersebut secara langsung maupun tidak langsung mempengaruhi semua sistem yang berhubungan ataupun tidak dengan sistem informasi itu sendiri. Selain itu berkembangnya teknologi informasi saat ini dibutuhkan keamanan pesan/data agar dapat berkomunikasi dengan aman di antara kedua belah pihak.

Pertukaran informasi mempermudah pengguna dalam mendapatkan informasi secara cepat dan dapat saling berinteraksi satu dengan yang lainnya. Banyak keuntungan yang dapat diperoleh dengan melakukan pertukaran informasi, yaitu untuk mengefisienkan waktu sehingga informasi dapat diterima tepat waktu. Namun di samping keuntungan tersebut terdapat kelemahan dimana munculnya pihak ketiga yang menginginkan informasi tersebut untuk keperluan pribadi dan tentunya membuat informasi menjadi tidak rahasia lagi.

Oleh karena itu, dalam penelitian ini, dilakukan pengembangan dengan membuat aplikasi penyandian pesan teks yang membuat informasi tersebut dirahasiakan. Adapun proses keamanan dilakukan dengan catatan ada pengirim dan ada penerima pesan teks tersebut. Dimana pengirim mengirimkan pesan teks untuk dienkripsi, kemudian menentukan kunci,

hasil dari pesan teks terenkripsi tersebut akan dikirim ke penerima. Dan dari sisi penerima akan membuka pesan teks tersebut dengan kunci yang telah ditentukan, sehingga pesan teks tersebut dapat dibaca.

Algoritma blowfish merupakan salah satu algoritma kriptografi modern yang menggunakan kunci simetris dalam proses enkripsi dan dekripsi. Penerapan algoritma blowfish dalam penelitian ini yakni untuk mengamankan pesan teks antara pengirim dan penerima pesan teks tersebut melalui aplikasi berbasis web.

Penelitian terkait juga dilakukan oleh (Nurani & Siswanto, 2018). Penelitian ini bertujuan untuk membuat sistem Secure Chatting (Instan Messanging) menggunakan aplikasi enkripsi dan deskripsi dengan menggunakan Algoritma Blowfish. Hasil dari pengujian yang di dapatkan bahwa untuk enkripsi dan dekripsi pesan atau karakter tidak membutuhkan banyak waktu hanya tidak melebihi 1 detik. Kesimpulan yang di dapat jumlah karakter hampir tidak mempengaruhi waktu proses enkripsi dan dekripsi.

Berdasarkan uraian tersebut di atas, maka penulis tertarik untuk mengangkat judul penelitian yaitu tentang "**Aplikasi Penyandian Teks Berbasis Web Menggunakan Algoritma Blowfish"**.

1.2. Rumusan Masalah

Dari uraian latar belakang tersebut, maka dapat dirumuskan masalah, yaitu Bagaimana membuat Aplikasi Penyandian Teks Berbasis Web Menggunakan Algoritma Blowfish?

`

1.3. Batasan Masalah

Agar tidak melebar dari permasalahan yang akan dibahas, maka penulis membatasi masalah dalam penelitian ini, yaitu :

1) Aplikasi penyandian teks dibangun berbasis online sehingga dapat diakses kapan saja dan dimana saja
2) Alamat Website yang digunakan untuk aplikasi http://www.kirim-pesan.com
3) Bahasa pemrograman yang digunakan adalah PHP dan database MySQL.

1.4. Tujuan Penelitian

Tujuan penelitian ini dibagi menjadi 2 (dua) bagian yaitu Tujuan Umum, dan Tujuan Khusus. Adapun tujuan penelitian ini, antara lain :

1) Tujuan Umum

   Untuk memenuhi salah satu syarat penyusunan skripsi pada Program Rekayasa Sistem Komputer Fakultas Ilmu Komputer Universitas Dehasen Bengkulu.

2) Tujuan Khusus

   Untuk membuat aplikasi penyandian teks berbasis web menggunakan algoritma blowfish

1.5. Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat bermanfaat bagi masyarakat atau pengguna aplikasi, antara lain :

`

1. Dapat memberikan sebuah solusi pengamanan pesan teks dengan penerapan ilmu kriptografi.
2. Dapat dijadikan bahan referensi dalam pembuatan aplikasi pengamanan pesan teks berbasis *web*

`

BAB II

LANDASAN TEORI

2.1. Implementasi

Implementasi adalah suatu tahap dimana akan dilakukan penerapan dari program yang telah dibuat perancangan sebelumnya dan telah melalui proses analisa dan desain secara rinci. Tahap implementasi ini merupakan tahap yang penting untuk menuju suksesnya sistem baru. Sistem baru yang memberikan kepercayaan kepada pengguna sistem, bahwa sistem baru ini dapat bekerja secara efektif dan efisien. Tujuan dari tahapan implementasi sistem yaitu untuk mengetahui apa saja kelebihan dan kekurangan dari program baru yang dibuat (Firdayanti, 2013).

Penerapan (Implementasi) adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Penerapan biasanya dilakukan setelah perencanaan sudah dianggap sempurna (Bayuntara, 2017).

Berdasarkan kedua pengertian tersebut, maka dapat disimpulkan bawah penerapan merupakan suatu tindakan yang dilakukan melalui tahapan-tahapan yang telah disusun secara efisien dan terperinci.

2.2. Algoritma Blowfish

Blowfish termasuk dalam enkripsi *block Chiper* 64-bit dengan panjang kunci minimal 32 bit sampai 448-bit. *Blowfish* alias "*OpenPGP.Cipher.4*" merupakan enkripsi yang termasuk dalam golongan

Symmetric Cryptosystem, metode enkripsinya mirip dengan *DES* (*DES like Cipher*) diciptakan oleh seorang *Cryptanalyst* bernama Bruce Schneier Presiden perusahaan *Counterpane Internet Security*, *Inc* (Perusahaan konsultan tentang kriptografi dan keamanan komputer) dan dipublikasikan tahun 1994. Dibuat untuk digunakan pada komputer yang mempunyai mikroprosesor besar (32-bit k eatas dengan *cache* data yang besar) (Wardoyo, 2016)

Blowfish dikembangkan untuk memenuhi kriteria desain yang cepat dalam implementasinya di mana pada keadaan optimal dapat mencapai 26 *clock cycle per Byte*, kompak di mana dapat berjalan pada memori kurang dari 5 KB, sederhana dalam algoritmanya sehingga mudah diketahui kesalahannya, dan keamanan yang *variable* panjang kunci bervariasi (minimum 32 bit, maksimum 448 bit, *multiple* 8 bit, *default* 128 bit).

Algoritma *blowfish* menggunakan kunci yang sama untuk proses enkripsi dan dekripsi data dengan membagi pesan ke dalam blok-blok dengan ukuran yang sama panjang. *Blowfish* termasuk dalam enkripsi *block cipher* 64 *bit* dengan panjang kunci antara 32 *bit* sampai 448 *bit*. Algoritma *blowfish* terdiri atas dua bagian, yaitu :

1. *Key-Expansion*

   Berfungsi merubah kunci (Minimum 32-*bit*, Maksimum 448-*bit*) menjadi beberapa *array* subkunci (*subkey*) dengan total 4168 *byte*

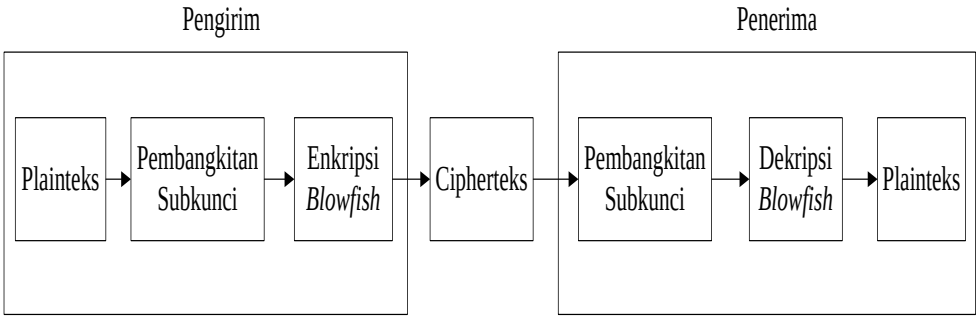
2. Enkripsi Data

   Terdiri dari iterasi fungsi sederhana (*Feistel Network*) sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi kunci-*dependent* dan

`

substitusi kunci- dan data-*dependent*. Semua operasi adalah penambahan (*addition*) dan *XOR* pada variabel 32-*bit*. Operasi tambahan lainnya hanyalah empat penelusuran tabel (*table lookup*) *array* berindeks untuk setiap putaran.

Adapun blok diagram pemodelan sistem, dapat dilihat pada gambar 2.1



Gambar 2.1. Blok Diagram Pemodelan Sistem

Blowfish adalah salah satu algoritma cipherblok yang tercepat dan digunakan secara luas di dunia, kecuali ketika pergantian kunci. Setiap kunci baru memerlukan pemrosesan awal yang sebanding dengan mengenkripsikan teks dengan ukuran sekitar 4 kilobyte. Penggunaan algoritma *blowfish* antara lain terdapat pada :

1. 96Crypt oleh Fever.Link.

   Merupakan aplikasi untuk enkripsi dan dekripsi arsip dan folder.

2. A-Lock, oleh Trillium Technology Group

   A-Lock merupakan perangkat lunak enkripsi yang terintegrasi dengan aplikasi e-mail untuk windows yang terkenal

3. Access Manager, oleh Citi-Software Ltd

   Aplikasi password manager untuk sistem operasi windows

`

4. Canner, oleh Cinnabar Systems

   Digunakan untuk melindungi kode java dari reverse engineering dengan cara membuat native windows executable yang mengandung kelas dan resource dan aplikasi dalam keadaan terenkripsi. Kelas dan resource ini kemudian akan didekripsikan pada memori ketika muncul permintaan dari mesin virtual java.

5. Putty, oleh Simon Tatham.

   Client SSH untuk Win32. Aplikasi ini dapat digunakan dan didapatkan secara gratis.

Penerapan Perhitungan Manual Pada Algoritma Blowfish dalam Proses Enkripsi, antara lain :

1. Inisialisasi *P-Array* sebanyak 18 buah (P_0, P_1, ..., P_{17}) masing-masing bernilai 32 *bit*.
2. Inisialisasi *S-Array* sebanyak 4 buah masing-masing bernilai 32 *bit* yang memiliki masukan hingga 256, seperti di bawah ini :

   $S_{1.0}$, $S_{1.1}$, ... $S_{1.255}$

   $S_{2.0}$, $S_{2.1}$, ... $S_{2.255}$

   $S_{3.0}$, $S_{3.1}$, ... $S_{3.255}$

   $S_{4.0}$, $S_{4.1}$, ... $S_{4.255}$

3. Memulai proses enkripsi (*plaintext*) dengan X = 64 *bit*
4. X dibagi menjadi 2, sehingga terdapat dua bagian yaitu *XL* (32 *bit*) dan *XR* (32 *bit*).
5. $i = 0$ merupakan inisial iterasi/perputaran yang dimulai dari 0 ($i = i + 1$)

`

6. Memproses fungsi $F = XL/4$ menjadi *a, b, c, d* masing-masing 8 *bit*
7. Memproses $F(XL) = (((S_0.a + S_1.b \bmod 2^{32}) XOR S_{2,} c) + S_{3,}d \bmod 2^{32})$
8. Selanjutnya lakukan operasi $XL = XL \; xor \; Pi$ dan $XR = F(XL) \; xor \; XR$
9. Menukar hasil *XL* dan *XR*. $XL = XR$ dan $XR = XL$
10. Melakukan perulangan sebanyak 16 kali.
11. Pada perulangan ke-16, terdapat proses penukaran hasil *XL* dan *XR*
12. Setelah proses perulangan selesai pada proses terdapat operasi untuk $XR = XR \; xor \; P_{16}$ dan $XL = XL \; xor \; P_{17}$.
13. Proses terakhir *XL* dan *XR* digabungkan kembali sehingga menjadi cipher text 64 *bit*.
14. Selesai

Penerapan Perhitungan Manual Pada Algoritma Blowfish dalam Proses Dekripsi, antara lain :

1. Inisialisasi *P-Array* sebanyak 18 buah (P_0, P_1, ..., P_{17}) masing-masing bernilai 32 *bit*.
2. Inisialisasi *S-Array* sebanyak 4 buah masing-masing bernilai 32 *bit* yang memiliki masukan hingga 256, seperti di bawah ini :

$$S_{1.0}, S_{1.1}, \ldots S_{1.255}$$

$$S_{2.0}, S_{2.1}, \ldots S_{2.255}$$

$$S_{3.0}, S_{3.1}, \ldots S_{3.255}$$

$$S_{4.0}, S_{4.1}, \ldots S_{4.255}$$

3. Memulai proses dekripsi (*cipher text*) dengan $X = 64$ *bit*
4. *X* dibagi menjadi 2, sehingga terdapat dua bagian yaitu *XL* (32 *bit*) dan *XR* (32 *bit*).

`

5. $i = 0$ merupakan inisial iterasi/perputaran yang dimulai dari 0 hingga $i = 16$.
6. $j = 17$ merupakan inisial pengambilan *P-Array* dimulai dari P_{17}
7. Memproses fungsi $F = XL/4$ menjadi *a, b, c, d* masing-masing 8 *bit*
8. Memproses $F(XL) = (((S_{1,a} + S_{2,b}) \text{ XOR } S_{3,c}) + S_{4,d})$
9. Selanjutnya lakukan operasi $XL = XL \text{ xor } P_j$ dan $XR = F(XL) \text{ xor } XR$
10. Menukar hasil *XL* dan *XR*. $XL = XR$ dan $XR = XL$
11. Melakukan perulangan sebanyak 16 kali.
12. Setelah perulangan selesai, maka dilanjutkan dengan proses pertukaran hasil *XL* dan *XR*
13. Memproses operasi untuk $XR = XR \text{ xor } P_1$ dan $XL = XL \text{ xor } P_0$.
14. Proses terakhir *XL* dan *XR* digabungkan kembali
15. Menghasilkan *Plaintext* 64 *bit*.
16. Selesai

2.3. Kriptografi

Sejarah kriptografi sebagian besar merupakan sejarah kriptografi klasik, yaitu metode enkripsi yang menggunakan kertas dan pensil atau mungkin dengan bantuan alat mekanik sederhana. Secara umum algoritma kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma substitusi (*substitution cipher*). *Cipher* transposisi mengubah susunan huruf-huruf di dalam pesan, sedangkan *cipher* substitusi mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain (Pabokory, Astuti, & Kridalaksana, 2015).

`

Kriptografi merupakan seni dan ilmu untuk memproteksi pengiriman data dengan mengubahnya menjadi kode tertentu dan hanya ditujukan untuk orang yang hanya memiliki sebuah kunci untuk mengubah kode itu kembali yang berfungsi dalam menjaga kerahasiaan data atau pesan. Dalam kriptografi, data atau pesan yang dikirimkan melalui jaringan akan disamarkan sedemikian rupa. Sehingga seandainya data tersebut bisa diperoleh dan dibaca oleh orang lain, maka pihak yang tidak berhak atau berwenang tersebut tidak akan bisa mengerti arti dari data tersebut.

Kriptografi berasal dari bahasa Yunani yaitu *cryptos* yang berarti *secret* (rahasia) dan *graphein* yang berarti *writing* (menulis). Jadi kriptografi berarti *secret writing* (tulisan rahasia). Sedangkan definisi kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Algoritma kriptografi selalu terdiri dari dua bagian, yaitu enkripsi dan dekripsi. Enkripsi (*encryption*) adalah proses untuk menyandikan *plaintext* atau *cleartext* menjadi bentuk *ciphertext*. Sedangkan dekripsi (*decryption*) adalah proses mengembalikan *ciphertext* menjadi *plaintext* semula. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci. Kunci biasanya berupa *string* atau deretan bilangan

Dalam bidang kriptografi terdapat dua konsep yang sangat penting atau utama yaitu enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi atau data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah

kembali bentuk tersamar tersebut menjadi informasi awal. Sebuah pesan atau data yang masih asli dan belum mengalami penyandian dikenal dengan istilah *plaintext*. Kemudian setelah disamarkan dengan suatu cara penyandian, maka *plaintext* ini disebut sebagai *chipertext*. Proses penyamaran dari *plaintext* ke *ciphertext* disebut enkripsi (*encryption*), dan proses pengembalian dari *ciphertext* menjadi *plaintext* kembali disebut dekripsi (*decryption*)

Dari paparan awal dapat dirangkumkan bahwa kriptografi bertujuan untuk member layanan keamanan. Yang dinamakan aspek-aspek keamanan :

1. Kerahasiaan (*confidentiality*) adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
2. Integritas data (*data integrity*) adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
3. Otentikasi (*authentication*) adalah layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user autehentication*).
4. *Non-repudiation* adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan.

Di dalam kriptografi akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu :

1. Pesan, *Plainteks*, dan *Cipherteks*

`

Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (*plaintext*) atau teks jelas (*cleartext*).

2. Pengirim dan Penerima

   Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan.

3. Enkripsi dan deskripsi

   Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering* (standar nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula disebut deskripsi (*decryption*) atau *deciphering* (standar nama menurut ISO 7498-2).

4. *Cipher* dan kunci

   Algoritma kriptogarfi disebut juga *cipher*, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen *plainteks* dan himpunan yang berisi *cipherteks*. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan *plainteks* dan C menyatakan *cipherteks*, maka :

   $$E(P) = C \rightarrow \text{fungsi enkripsi E memetakan P ke C}$$

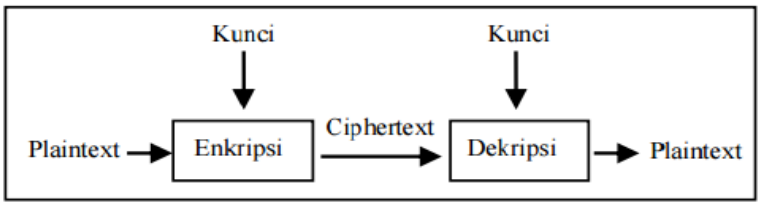
`

D(C) = P → fungsi dekripsi D memetakan C ke P

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan D(E(P)) = P harus benar.

Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa *string* atau deretan bilangan.

Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai skema diperlihatkan pada Gambar 2.1.

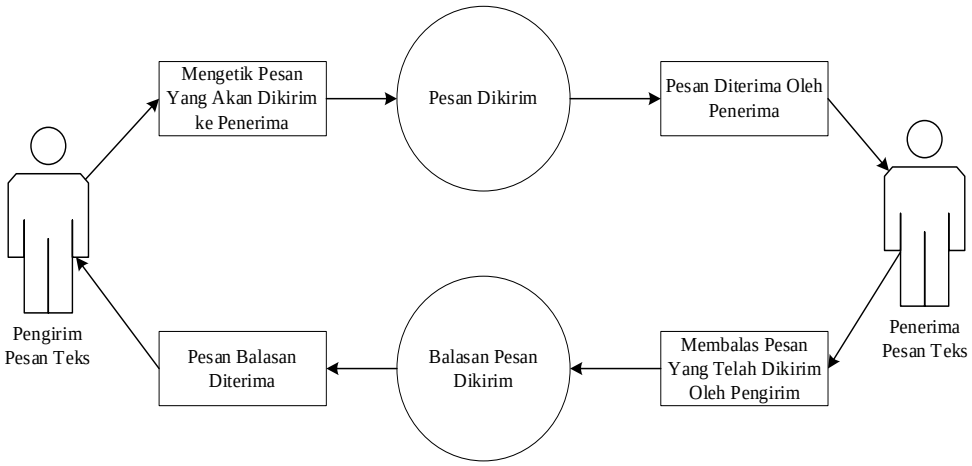


Gambar 2.2. Skema Enkripsi dan Dekripsi Dengan Menggunakan Kunci

2.4. Pesan Teks

Pesan teks adalah suatu layanan yang memungkinkan user dapat mengirim sebuah pesan yang berisi informasi kepada user lain secara cepat dengan biaya yang kecil. Pesan teks yang dikirim jauh lebih cepat daripada pengiriman pesan suara atau video karena hanya terdiri dari karakter teks dan angka (Ramdan & Maliki, 2019).

`



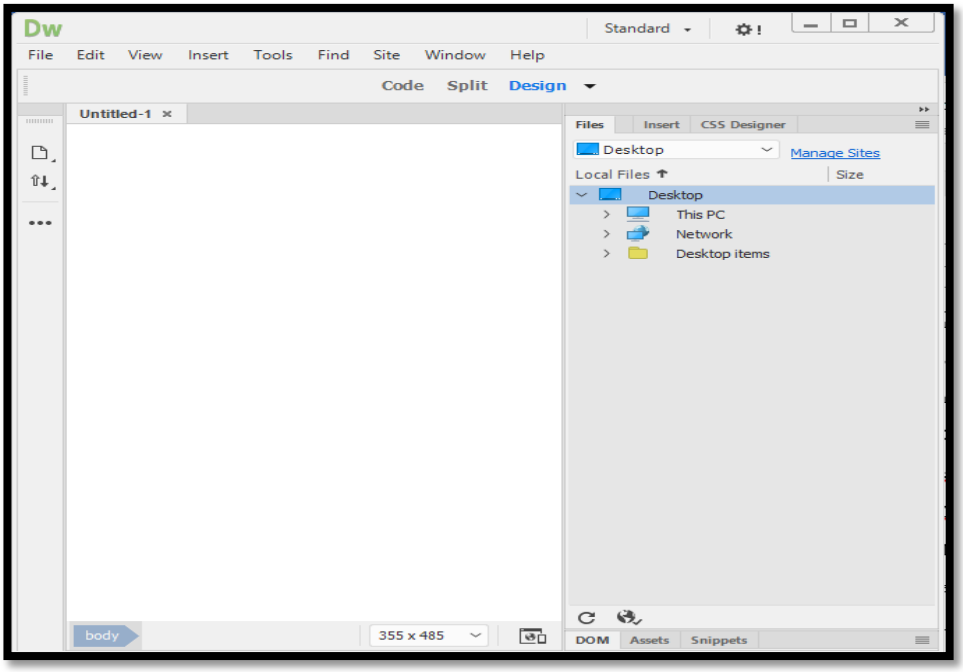
Gambar 2.3. Konsep Pengiriman Pesan Teks

2.5. *Adobe Dreamweaver*

Adobe Dreamweaver merupakan aplikasi pengembang yang berfungsi untuk mendesain web yang dibuat, dikembangkan, dan diproduksi oleh Adobe System. Aplikasi pengembang web ini sangat digemari oleh web desainer dalam merancang web sebab perangkat lunak komputer ini memiliki kelebihan dan kemudahan dalam penggunaannya. Dengan menggunakan aplikasi ini, pengembangan web dapat dilakukan secara visual, sehingga hasil perancangan web dapat langsung terlihat tanpa harus menggunakan aplikasi bantu peramban seperti Google Chrome, Firefox atau Internet Explorer. Teknologi web yang didukung oleh Adobe Dreamweaver sangat beragam, salah satunya adalah teknologi untuk kebutuhan pengembangan web berbasis mobile (Mandar, 2017).

Adapun antarmuka tampilan awal dari aplikasi Adobe Dreamweaver CC 2019 terlihat pada Gambar 2.4.

`



Gambar 2.4. Tampilan Awal Adobe Dreamweaver

PHP merupakan bahasa pemrograman berbasis web yang memiliki kemampuan memproses dan mengolah data secara dinamis. PHP dapat di katakan sebagai sebuah *server-side embedded script language*, artinya sintak-sintak dan perintah program yang ditulis akan sepenuhnya dijalankan oleh server tetapi dapat di sertakan pada halaman HTML biasa (Karman, 2017).

2.6. Bahasa Pemrograman PHP

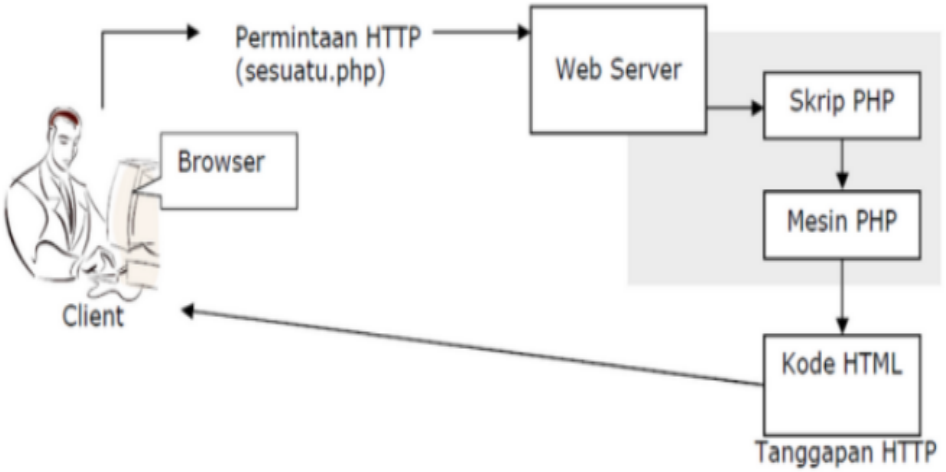
PHP berasal dari kata Hypertext Preprocessor yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML.PHp sebagai sekumpulan skrip atau bahasa program memiliki fungsi utama, yaitu mampu mengumpulkan dan mengevaluasi hasil survei atau bentuk apapun ke server database dan pada tahap selanjutnya akan menciptakan efek beruntun. Efek beruntun PHP ini berupa tindakan dari skrip lain yang akan

melakukan komunikasi dengan database, mengumpulkan dan mengelompokkan informasi, kemudian menampilkannya pada saat ada tamu website memerlukannya (menampilkan informasi sesuai permintaan user) (Mundzir, 2020).

PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website. Oleh karena itu, PHP dapat dijalankan menggunakan browser. PHP memiliki sifat dinamis dan interaktif. Dinamis yang artinya website tersebut bisa berganti konten sesuai kondisi tertentu, misalnya dapat menampilkan produk yang berbeda-beda untuk setiap pengunjung. Sedangkan interaktif artinya website yang tersebut dapat memberi feedback bagi user, misalnya dapat menampilkan hasil pencarian produk ketika seorang calon pembeli membutuhkan sebuah produk (Enterprise, 2019).

PHP dikenal sebagai sebuah bahasa scripting yang menyatu dengan tag-tag HTML yang dieksekusi di server dan digunakan untuk membuat halaman web yang dinamis. Konsep kerja PHP diawali dengan satu permintaan suatu halaman web oleh browser. Berdasarkan URL (Uniform Resource Locator) atau dikenal dengan alamat internet, browser mendapat alamat dari webserver, mengidentifikasikan alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh web server. Adapun cara kerja PHP seperti Gambar 2.5 (Krisbiantoro & Abda'u, 2021).

`



Gambar 2.5. Cara Kerja PHP

2.7. Basis Data

Basis data merupakan suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan software untuk melakukan manipulasi untuk kegiatan tertentu. Basis data bisa diartikan juga sebagai sekumpulan data yang disusun dalam bentuk beberapa tabel yang saling memiliki relasi maupun berdiri sendiri (Widodo, 2017).

Basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. Basis Data merupakan data yang terintegrasi, yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi (Hardiansyah, 2020)

Sistem Basis Data merupakan basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personil yang merancang dan mengelola basis data, Teknik-teknik untuk merancang dan mengelola basis data, serta sistem computer yang mendukungnya. Sistem Basis Data

`

adalah suatu sistem menyusun dan mengelola record-record menggunakan computer untuk menyimpan atau merekam serta memelihara data operasional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan

Ada tiga fase dalam membuat desain basis data, yaitu :

1. *Conseptual Database Design*

   Merupakan suatu proses pembentukan model yang berasal dari informasi yang digunakan dalam perusahaan yang bersifat independen dari keseluruhan aspek fisik. Model data tersebut dibangun menggunakan informasi dalam spesifikasi kebutuhan user dan merupakan sumber informasi untuk fase desain logikal.

2. *Logical Database Design*

   Merupakan suatu proses pembentukan model yang berasal dari informasi yang digunakan dalam perusahaan berdasarkan model data tertentu, namun independen terhadap DBMS tertentu dan aspek fisik lainnya. Misalnya relasional. Model data konseptual yang telah dibuat sebelumnya, diperbaiki dan dipetakan kembali ke dalam model data logikal.

3. *Physical Database Design*

   Merupakan proses yang menghasilkan deskripsi implementasi basis data pada penyimpanan sekunder. Menggambarkan struktur

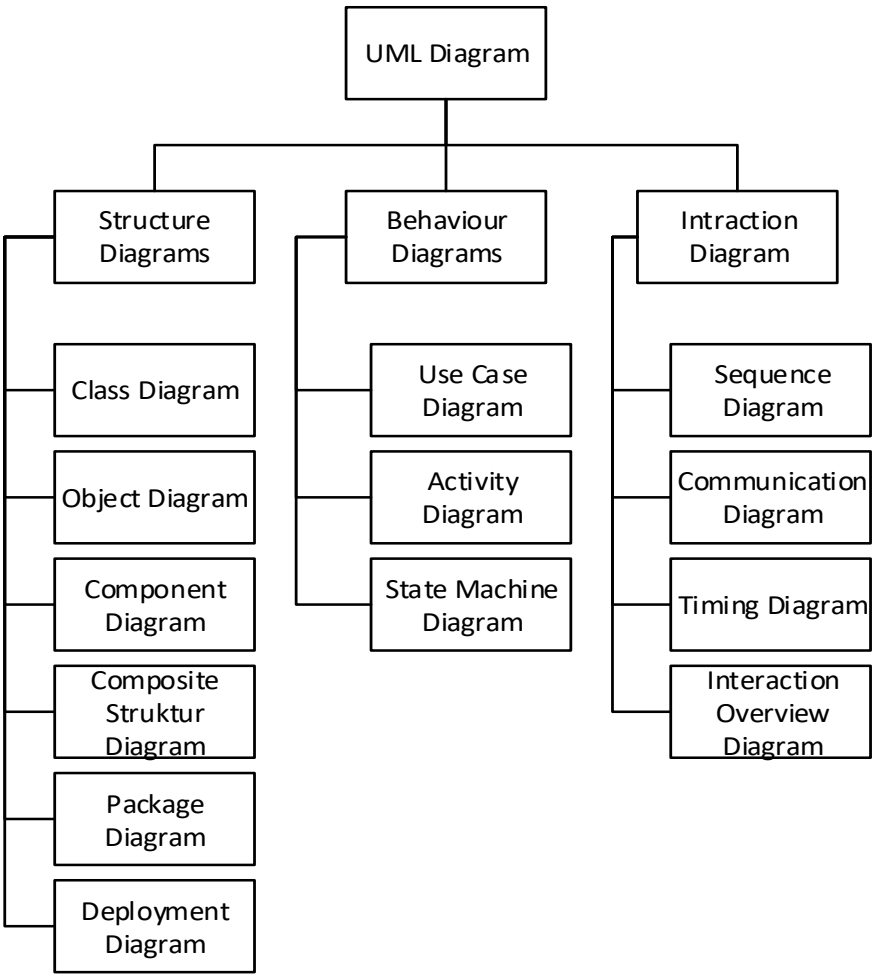
penyimpanan dan metode akses yang digunakan untuk mencapai akses yang efisien terhadap data. Dapat dikatakan juga desain fisikal merupakan cara pembuatan menuju DBMS tertentu.

2.8. *Unified Modeling Language* (UML)

Unified Modeling Language merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek (Rosa & Shalahuddin, 2016).

UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat di lihat pada Gambar 2.5.

`



Gambar 2.6. Diagram UML

2.8.1. *Class Diagram*

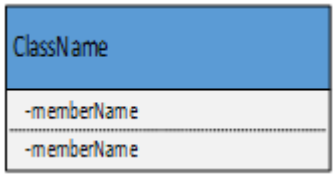
Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas.

`

Tabel 2.1. Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
Asosiasi/association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/directed association 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
Kebergantungan/dependency 	Relasi antarkelas dengan makna kebergantungan antarkelas
Agregasi/aggregation 	Relasi antarkelas dengan makna semua bagian (whole-part)

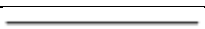
`

2.8.2. *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan. Diagram objek juga berfungsi untuk mendefinisikan contoh nilai atau isi dari atribut tiap kelas.

Berikut adalah simbol-simbol yang ada pada diagram objek.

Tabel 2.2. Objek Diagram

Simbol	Deskripsi
Objek 	Objek dari kelas yang berjalan saat sistem dijalankan
Link 	Relasi antar objek

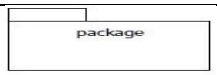
2.8.3. *Component Diagram*

Digram komponen atau component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada di dalam sistem. Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan.

Berikut adalah simbol-simbol yang ada pada diagram komponen.

`

Tabel 2.3. Component Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih komponen
Komponen 	Komponen sistem
Kebergantungan dependency 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antarmuka /interface 	Sama dengan konsep interface pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen
Link	Relasi antar komponen

2.8.4. *Composite Structure Diagram*

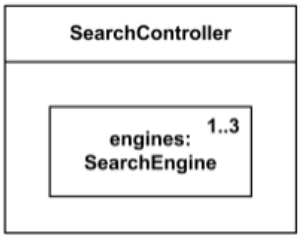
Composite structure diagram baru mulai ada pada UML versi 2.0 pada versi 1.x diagram ini belum muncul. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat

`

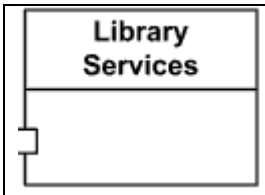
berjalan (runtime) dari instance yang saling berhubungan. Dapat menggambarkan struktur di dalam kelas atau kolaborasi.

Berikut adalah simbol-simbol yang ada pada composite structure diagram.

Tabel 2.4. Composite Structure Diagram

Simbol	Deskripsi
Property 	Property adalah satu set dari suatu instance. Rolename : peran/nama /identitas dari property (opsional) Typename: tipe kelas dari property (harus ada)
Connector	Connector adalah cara komunikasi dari 2 buah instance comName : nama connector (optional) connType : tipe connector (optional)
Port 	Port adalah cara yang digunakan dalam diagram composite structure tanpa menampilkan detail internal dari suatu sistem. Port digambarkan dalam bentuk kotak kecil yang menempel atau di dalam suatu property
Class	Kelas : jika yang akan dijabarkan

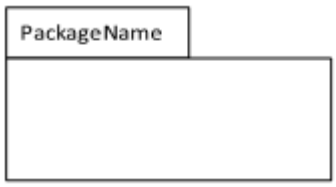
`

	strukturnya adalah sebuah kelas

2.8.5. *Package Diagram*

Package Diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan package diagram. Berikut ini simbol-simbol yang digunakan dalam package diagram.

Tabel 2.5. Package Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas atau elemen diagram UML lainnya.

2.8.6. *Deployment Diagram*

Diagram deployment atau deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut :

a. Sistem tambahan (embedded system) yang menggambarkan rancangan device, node, dan hardware

`

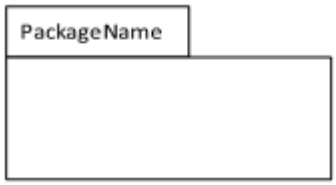
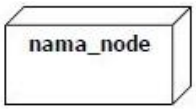
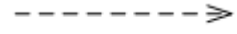
b. Sistem clien/server

c. Sistem terdistribusi murni

d. Rekayasa ulang aplikasi

Berikut ini simbol-simbol yang digunakan dalam deployment diagram.

Tabel 2.6. Deloyment Diagram

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih node.
Node 	Biasanya mengacu pada perangkat keras (hardware), perangkat lunak yang tidak dibuat sendiri (software), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
Kebergantungan dependency 	Kebergantungan antar node arah panah mengarah pada node yang dipakai

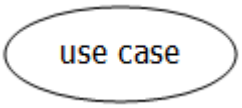
`

Link ———————	Relasi antar node

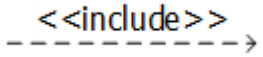
2.8.7. *Use Case Diagram*

Use case atau diagram use case merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol yang ada pada diagram use case.

Tabel 2.7. Use Case Diagram

Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama use case
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang,

`

	tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Asosiasi/Association	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
Generalisasi/ generalization	Hubungan generalisasi atau spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. Misalnya arah panah pengarah pada use case yang menjadi generalisasinya (umum).
Menggunakan/ include/ uses 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahakn memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.

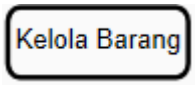
2.8.8. *Activity Diagram*

Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang

`

dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

Tabel 2.8. Activity Diagram

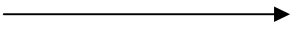
Simbol	**Keterangan**
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

`

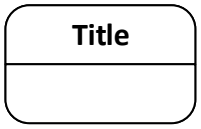
2.8.9. *State Machine Diagram*

State machine diagram atau statechart diagram atau dalam bahasa Indonesia disebut diagram mesin status atau sering disebut diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika diagram sekuen digunakan untuk interaksi antar objek maka diagram status digunakan untuk interaksi di dalam sebuah objek. Perubahan tersebut digambarkan dalam suatu graf berarah. State machine diagram merupakan pengembangan dari diagram Finite State Automata dengan penambahan beberapa fitur dan konsep baru.

Tabel 2.9. State Machine Diagram

Simbol	**Keterangan**
Start/Status awal (initial state) 	Start atau initial state adalah state atau keadaan awal pada saat sistem mulai hidup.
End/ Status Akhir (final state) 	End atau final state adalah state keadaan akhir dari daur hidup suatu sistem
Event 	Event adalah kegiatan yang menyebabkan berubahnya status mesin

`

State 	State atau status adalah keadaan sistem pada waktu tertentu.

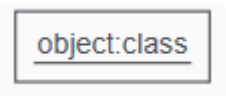
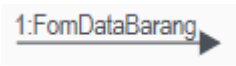
2.8.10. *Sequence Diagram*

Diagram sekuen menggambarkan kelakukan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk mengambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case. Berikut adalah simbol-simbol yang ada pada diagram sekuen.

Tabel 2.10. Sequence Diagram

Simbol	**Keterangan**
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

`

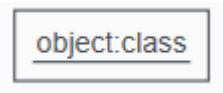
Garis Hidup/Lifeline	Menyatakan suatu objek.
Objek 	Menyatakan objek yang berinteraksi pesan
Waktu Aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan Tipe Call 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
Pesan Tipe Create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan Tipe Destroy 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

`

2.8.11. *Communication Diagram*

Communication diagram atau diagram komunikasi pada UML, versi 2.x adalah penyederhanaan dari diagram kolaborasi pada UML versi 1.x. Diagram komunikasi menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram use case untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem. Berikut adalah simbol-simbol yang ada pada diagram komunikasi.

Tabel 2.11. *Communication Diagram*

Simbol	**Keterangan**
Objek 	Objek yang melakukan interaksi pesan
Link	Relasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri
Arah pesan/ Stimulus	Arah pesan yang terjadi, jika pada suatu link ada dua arah pesan yang berbeda maka arah juga digambarkan dua arah ada satu sisi link

`

BAB III

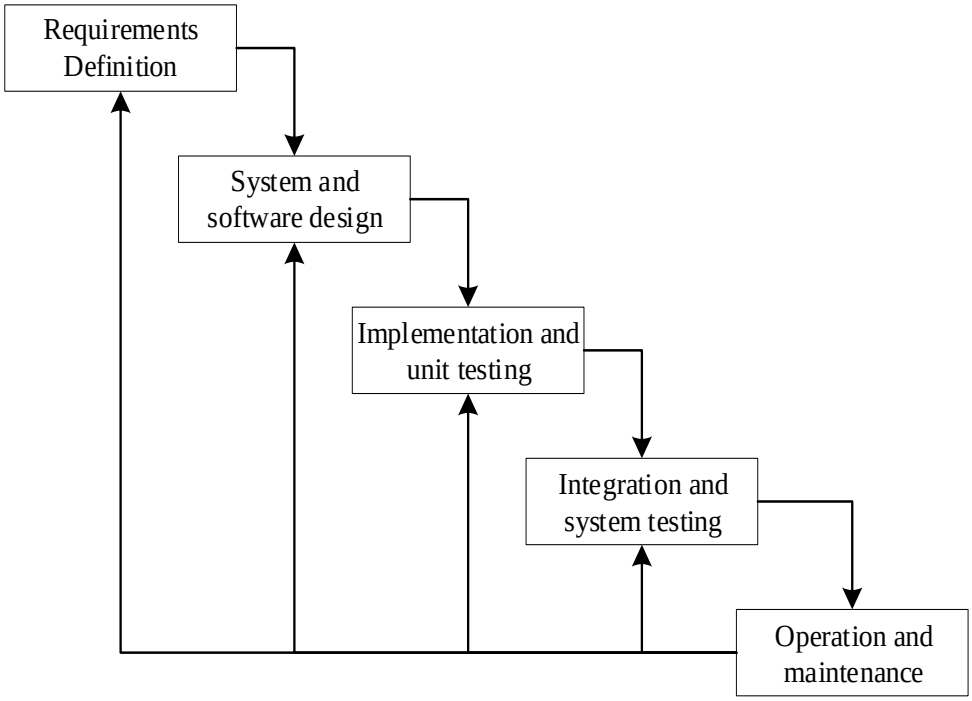
METODOLOGI PENELITIAN

3.1. Subjek Penelitian

Penelitian dilakukan secara mandiri dengan membuat aplikasi berbasis web. Waktu penelitian akan dilakukan pada bulan Mei 2022 sampai dengan Oktober 2022.

3.2. Metode Penelitian

Metode penelitian yang diterapkan pada penelitian ini adalah dengan pengembangan metode *waterfall*. Metode *waterfall* merupakan model pengembangan sistem informasi yang sistematik dan sekuensial. Metode *waterfall* memiliki tahapan-tahapan seperti Gambar 3.1. (Trsitianto, 2018).



Gambar 3.1. Metode Waterfall

Keterangan :

1) *Requirements analysis and definition*

Layanan sistem, kendala, dan tujuan ditetapkan oleh hasil konsultasi dengan pengguna yang kemudian didefinisikan secara rinci dan berfungsi sebagai spesifikasi sistem.

2) *System and software design*

Tahapan perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3) *Implementation and unit testing*

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4) *Integration and system testing*

Unit-unit individu program atau program digabung dan diuji sebagai sebuah sistem lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak atau tidak. Setelah pengujian, perangkat lunak dapat dikirimkan ke *customer*

5) *Operation and maintenance*

Biasanya (walaupun tidak selalu), tahapan ini merupakan tahapan yang paling panjang. Sistem dipasang dan digunakan secara nyata. *Maintenance* melibatkan pembetulan kesalahan yang tidak ditemukan pada tahapan-tahapan sebelumnya,

`

meningkatkan implementasi dari unit sistem, dan meningkatkan layanan sistem sebagai kebutuhan baru.

3.3. Instrumen Perangkat Lunak dan Perangkat Keras

Dalam penelitian ini diperlukan perangkat keras dan perangkat lunak yang akan digunakan dalam pembuatan aplikasi. Adapun perangkat keras dan perangkat lunak, yaitu :

1. Perangkat Keras (*Hardware*)

   Perangkat keras (*Hardware*) yang digunakan dalam penelitian ini, antara lain :

   a. Laptop Acer
   b. Processor Intel Core i3
   c. Memory RAM 2GB
   d. Hardisk 500GB

2. Perangkat Lunak (*Software*)

   Perangkat lunak (*Software*) yang digunakan dalam penelitian ini, antara lain :

   a. Sistem Operasi Windows 7
   b. XAMPP
   c. Adobe Dreamweaver CC 20

3.4. Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk memperoleh data yang dapat mendukung permasalahan yang akan dibahas. Dalam metode pengumpulan data, penulis menggunakan 3 (dua) metode untuk mengumpulkan data, antara lain :

a. Studi Praktikum

`

Studi praktikum adalah menerapkan algoritma blowfish dalam mengamankan data khususnya pesan teks. Studi praktikum dilakukan dengan menguji coba aplikasi dengan memberikan masukan input untuk mengetahui apakah aplikasi sudah berjalan dengan semestinya.

b. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan atau instansi yang berupa karya ilmiah, jurnal, buku-buku yang berhubungan dengan penulisan ini.

3.5. Metode Perancangan Sistem

3.5.1. Analisa Sistem Aktual

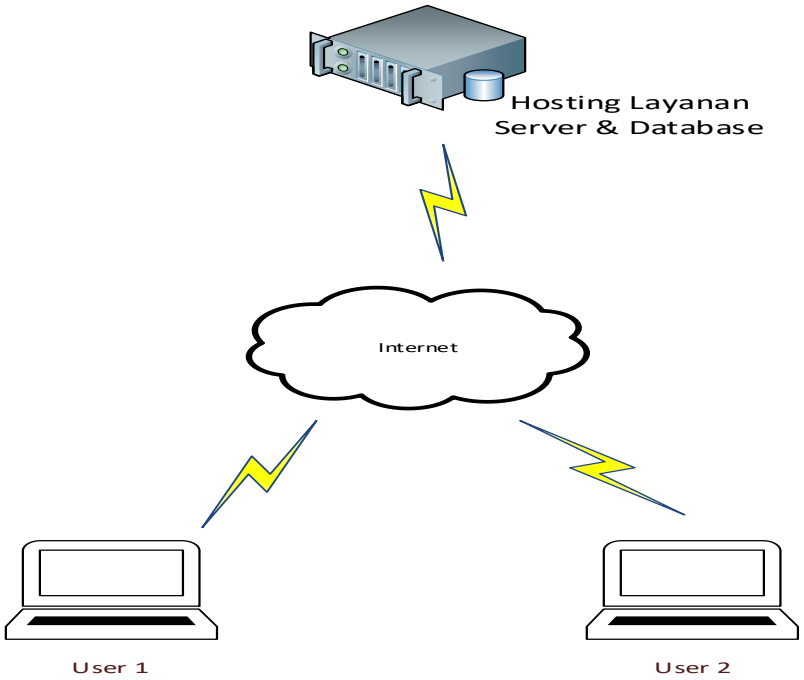
Pertukaran informasi mempermudah pengguna dalam mendapatkan informasi secara cepat dan dapat saling berinteraksi satu dengan yang lainnya. Banyak keuntungan yang dapat diperoleh dengan melakukan pertukaran informasi, yaitu untuk mengefisienkan waktu sehingga informasi dapat diterima tepat waktu. Namun di samping keuntungan tersebut terdapat kelemahan dimana munculnya pihak ketiga yang menginginkan informasi tersebut untuk keperluan pribadi dan tentunya membuat informasi menjadi tidak rahasia lagi.

3.5.2. Analisa Sistem Baru

Analisa sistem baru dilakukan berdasarkan hasil analisa sistem aktual, dimana dilakukan pengembangan dengan membuat aplikasi penyandian pesan teks

`

menggunakan algoritma blowfish. Aplikasi ini dibangun menggunakan PHP dan database MySQL dimana aplikasi tersebut dapat diletakkan pada layanan hosting web. Adapun proses keamanan dilakukan dengan catatan ada pengirim dan ada penerima pesan teks tersebut. Dimana pengirim mengirimkan pesan teks untuk dienkripsi, kemudian menentukan kunci, hasil dari pesan teks terenkripsi tersebut akan dikirim ke penerima. Dan dari sisi penerima akan membuka pesan teks tersebut dengan kunci yang telah ditentukan, sehingga pesan teks tersebut dapat dibaca. Adapun blog diagram aplikasi penyandian pesan teks berbasis web seperti Gambar 3.2.



Gambar 3.2. Blog Diagram Aplikasi Berbasis Web

Pada Gambar 3.2. terlihat bahwa aplikasi web dan database aplikasi tersimpan pada layanan hosting dimana akan diberikan domain untuk mengakses web tersebut secara online. Aplikasi ini dapat diakses oleh user 1

`

dan user 2 dengan memanfaatkan internet dengan membuka domain yang telah diaktifkan oleh layanan hosting.

A. Penerapan Algoritma Blowfish Dalam Enkripsi Pesan Teks

Plaintext = DEHASEN

Password = 1945

Langkah perhitungan manual yang dilakukan, yaitu sebagai berikut :

1. Inisialisasi *P-Array* ($P_0, P_1, ..., P_{17}$) masing-masing 32 bit, seperti tabel 3.1.

Tabel 3.1. P-Array Konversi ke Biner

P-array	Hexa	Konversi Biner (32 bit)
P0	243F6A88	00100100 00111111 01101010 10001000
P1	85A308D3	10000101 10100011 00001000 11010011
P2	13198A2E	00010011 00011001 10001010 00101110
P3	3707344	00000011 01110000 01110011 01000100
P4	A4093822	10100100 00001001 00111000 00100010
P5	299F31D0	00101001 10011111 00110001 11010000
P6	82EFA98	00001000 00101110 11111010 10011000
P7	EC4E6C89	11101100 01001110 01101100 10001001
P8	452821E6	01000101 00101000 00100001 11100110
P9	38D01377	00111000 11010000 00010011 01110111
P10	BE5466CF	10111110 01010100 01100110 11001111
P11	34E90C6C	00110100 11101001 00001100 01101100
P12	C0AC29B7	11000000 10101100 00101001 10110111

`

P13	C97C50DD	11001001 01111100 01010000 11011101
P14	3F84D5B5	00111111 10000100 11010101 10110101
P15	B5470917	10110101 01000111 00001001 00010111
P16	9216D5D9	10010010 00010110 11010101 11011001
P17	8979FB1B	10001001 01111001 11111011 00011011

2. Inisialisasi S-Array yang berjumlah masing-masing 255 dalam bentuk hexadecimal yang kemudian dikonversi ke biner, seperti tabel 3.2.

Tabel 3.2. Konversi S-Array ke Biner

S-Array	Hexa	Konversi Biner
S1,0	D1310BA6	11010001 00110001 00001011 10100110
...		
S1,255	6E85076A	01101110 10000101 00000111 01101010
S2,0	4B7A70E9	01001011 01111010 01110000 11101001
...		
S2,255	DB83ADF7	11011011 10000011 10101101 11110111
S3,0	E93D5A68	11101001 00111101 01011010 01101000
...		
S3,255	406000E0	01000000 01100000 00000000 11100000
S4,0	3A39CE37	00111010 00111001 11001110 00110111
...		
S4,255	3AC372E6	00111010 11000011 01110010 11100110

3. Plaintext = DEHASEN

`

Tabel 3.3. Konversi Plaintext Ke Biner

Karakter	Konversi Ke Biner
D	01000100
E	01000101
H	01001000
A	01000001
S	01010011
E	01000101
N	01001110

4. Kemudian Plaintext dibagi menjadi 2 bagian *XL* dan *XR* menjadi :

   XL = 01000100 01000101 01001000 01000001

   XR = 01010011 01000101 01001110 00000000

5. Pembangkitan Sub Kunci :

   Kunci : 1945

Tabel 3.4. Konversi Kunci Ke Biner

Karakter	Konversi ke Biner
1	00110001
9	00111001
4	00110100
5	00110101

Biner : 00110001 00111001 00110100 00110101

a. SubKunci Untuk Iterasi Pertama :

   P_0 = P_0 XOR Kunci

`

P_0 = 00100100 00111111 01101010 10001000 XOR

00110001 00111001 00110100 00110101

P_0 = 00010101 00000110 01011110 10111101

b. SubKunci untuk iterasi kedua :

P_1 = P_1 XOR P_0

P_1 = 10000101 10100011 00001000 11010011 XOR

00010101 00000110 01011110 10111101

P_1 = 10010000 10100101 01010110 01101110

6. Dalam hal ini penulis, hanya melakukan satu iterasi, dikarenakan total iterasi proses enkripsi adalah 16 putaran.

Untuk iterasi pertama i = 0 yaitu :

XL = XL XOR P_0

XL = 01000100 01000101 01001000 01000001 XOR

00010101 00000110 01011110 10111101

XL = 01010001 01000011 00010110 11111100

Fungsi F didapat dari :

XL dibagi menjadi 4 (a, b, c, d) masing-masing 8 bit =

a = 01010001

b = 01000011

c = 00010110

d = 11111100

Fungsi F : $F(XL) = (((S_0.a + S_1.b \bmod 2^{32}) \text{ XOR } S_{2,} c) + S_{3,}d \bmod 2^{32})$

`

$S_0.a + S_1.b \bmod 2^{32}$ = (11010001 00110001 00001011 10100110 . 01010001) + (01001011 01111010 01110000 11101001. 01000011) mod 2^{32}

= (10000100011000010000100101011111000011 0 + 10011110000010000101110001100111110 11) mod 2^{32}

= 11110001100100000011110010000001

XOR S_2.c = 11110001100100000011110010000001 XOR (11101001 00111101 01011010 01101000. 00010110)

=00000000 00011010 11110111 11111010 10111000 XOR 00010100 00001011 01000101 11000100 11110000

=00010100 00010001 10110010 00111110 01001000

+ S3.d mod 2^{32} = (00010100 00010001 10110010 00111110 01001000+ (00111010 00111001 11001110 00110111. 10011111)) mod 2^{32}

= (00010100 00010001 10110010 00111110 01001000 + 00101001 11100111 00010100 00101001) mod 2^{32}

= 00111011 10011001 01010010 01110001

F(XL) = 00111011 10011001 01010010 01110001

XR = *F(XL)* XOR *XR*

XR = 00111011 10011001 01010010 01110001 XOR 01010011 01000101 01001110 00000000

XR = 01101000 11011100 00011100 01110001

Menukar Nilai *XL* dan *XR* :

`

XL = *XR* ; *XR* = *XL*

XL = 01101000 11011100 00011100 01110001

XR = 01000100 01000101 01001000 01000001

7. Setelah melakukan 16 iterasi, maka akan menghasilkan nilai baru *XL* dan *XR* masing-masing 32 bit. Tukar kembali *XL* dan *XR*. Setelah itu XOR-kan nilai *XL* dan *XR* : *XR* = *XR* XOR P_{16} dan *XL* = *XL* XOR P_{17}
8. Kemudian *XL* dan *XR* digabungkan sehingga menghasilkan nilai akhir biner 01000001 10010110 01010000 01100101 00100000 00100000 00100011
9. Nilai biner tersebut di konversikan ke dalam kode ASCII sehingga menghasilkan ciphertext yaitu : AûPe #

B. Penerapan Algoritma Blowfish Dalam Dekripsi Pesan Teks

Ciphertext = AûPe #

Password = 1945

Langkah perhitungan manual dalam dekripsi pesan teks, yaitu sebagai berikut :

1. Inisialisasi *P-Array* (P_0, P_1, ..., P_{17}) masing-masing 32 bit, seperti tabel 3.5

Tabel 3.5. Konversi P-Array Ke Biner

P-array	Hexa	Konversi Biner (32 bit)
P0	243F6A88	00100100 00111111 01101010 10001000
P1	85A308D3	10000101 10100011 00001000 11010011
P2	13198A2E	00010011 00011001 10001010 00101110
P3	3707344	00000011 01110000 01110011 01000100

`

P4	A4093822	10100100 00001001 00111000 00100010
P5	299F31D0	00101001 10011111 00110001 11010000
P6	82EFA98	00001000 00101110 11111010 10011000
P7	EC4E6C89	11101100 01001110 01101100 10001001
P8	452821E6	01000101 00101000 00100001 11100110
P9	38D01377	00111000 11010000 00010011 01110111
P10	BE5466CF	10111110 01010100 01100110 11001111
P11	34E90C6C	00110100 11101001 00001100 01101100
P12	C0AC29B7	11000000 10101100 00101001 10110111
P13	C97C50DD	11001001 01111100 01010000 11011101
P14	3F84D5B5	00111111 10000100 11010101 10110101
P15	B5470917	10110101 01000111 00001001 00010111
P16	9216D5D9	10010010 00010110 11010101 11011001
P17	8979FB1B	10001001 01111001 11111011 00011011

2. Inisialisasi S-Array yang berjumlah masing-masing 255 dalam bentuk hexadecimal yang kemudian dikonversi ke biner, seperti tabel 3.6.

Tabel 3.6. Konversi S-Array ke Biner

S-Array	Hexa	Konversi Biner
S1,0	D1310BA6	11010001 00110001 00001011 10100110
...		
S1,255	6E85076A	01101110 10000101 00000111 01101010
S2,0	4B7A70E9	01001011 01111010 01110000 11101001

`

... S2,255	 DB83ADF7	11011011 10000011 10101101 11110111
S3,0 ... S3,255	E93D5A68 406000E0	11101001 00111101 01011010 01101000 01000000 01100000 00000000 11100000
S4,0 ... S4,255	3A39CE37 3AC372E6	00111010 00111001 11001110 00110111 00111010 11000011 01110010 11100110

3. Ciphertext = AûPe #

Tabel 3.7. Konversi Ciphertext ke Biner

Karakter	Konversi Ke Biner
A #	01000001
û	10010110
P	01010000
e	01100101
space	00100000
space	00100000
#	00100011

4. Kemudian dibagi menjadi 2 bagian *XL* dan *XR* menjadi :

   XL = 01000001 10010110 01010000 01100101

   XR = 00100000 00100000 00100011 00000000

5. Pembangkitan Sub Kunci :

   Kunci : 1945

`

Tabel 3.8. Konversi Kunci Ke Biner

Karakter	Konversi ke Biner
1	00110001
9	00111001
4	00110100
5	00110101

Biner : 00110001 00111001 00110100 00110101

a. SubKunci Untuk Iterasi Pertama :

P_{17} = P_{17} XOR Kunci

P_{17} = 10001001 01111001 11111011 00011011 XOR

00110001 00111001 00110100 00110101

P_{17} = 10111000 01000000 11001111 00101110

b. SubKunci untuk iterasi kedua :

P_{16} = P_{16} XOR P_{17}

P_{16} = 10010010 00010110 11010101 11011001 XOR

10111000 01000000 11001111 00101110

P_{16} = 00101010 01010110 00011010 11110111

6. Dua iterasi, dikarenakan total iterasi proses dekripsi adalah 16 putaran.

Untuk iterasi pertama i = 0 yaitu :

XL = XL XOR P_{17}

XL = 01000001 10010110 01010000 01100101 XOR

10111000 01000000 11001111 00101110

XL = 11111001 11010110 10011111 01001011

`

Fungsi *F* didapat dari :

XL dibagi menjadi 4 (*a*, *b*, *c*, *d*) masing-masing 8 bit =

a = 11111001

b = 11010110

c = 10011111

d = 01001011

Fungsi *F* : $F(XL) = (((S_0.a + S_1.b \bmod 2^{32}) \text{ XOR } S_2, c) + S_3,d \bmod 2^{32})$

$S_0.a + S_1.b \bmod 2^{32}$ = (11010001 00110001 00001011 10100110 . 11111001) + (01001011 01111010 01110000 11101001. 11010110) mod 2^{32}

= (11110001011010001010100011101 10 + 11000010110100110001011000110) mod 2^{32}

= 11101000000101111100010111001

XOR S_2.c = 11101000000101111100010111001 XOR (11101001 00111101 01011010 01101000. 10011111)

= 11010011100110100101111000100001

+ S3.d mod 2^{32} = (11010011100110100101111000100001 + 00111010 00111001 11001110 00110111 . 01001011) mod 2^{32}

= 10110101001110110000011010011

F(XL) = 10110101001110110000011010011

XR = F(XL) XOR *XR*

XR = 101101010011101100000 11010011 XOR

00100000 00100000 00100011 00000000

XR = 00110110 10000111 01000011 11010011

`

Menukar Nilai *XL* dan *XR* :

XL = *XR* ; *XR* = *XL*

XL =00110110 10000111 01000011 11010011;

XR = 01000001 10010110 01010000 01100101

7. Setelah melakukan 16 iterasi, maka akan menghasilkan nilai baru *XL* dan *XR* masing-masing 32 bit. Tukar kembali *XL* dan *XR*. Setelah itu XOR-kan nilai *XL* dan *XR* : *XR* = *XR* XOR P_1dan *XL* = *XL* XOR P_0
8. Kemudian *XL* dan *XR* digabungkan sehingga menghasilkan nilai akhir biner : 01000100 01000101 01001000 01000001 01010011 01000101 01001110
9. Nilai biner 01000100 01000101 01001000 01000001 01010011 01000101 01001110 di konversikan ke dalam kode ASCII sehingga menghasilkan plaintext yaitu : DEHASEN

C. Use Case Diagram

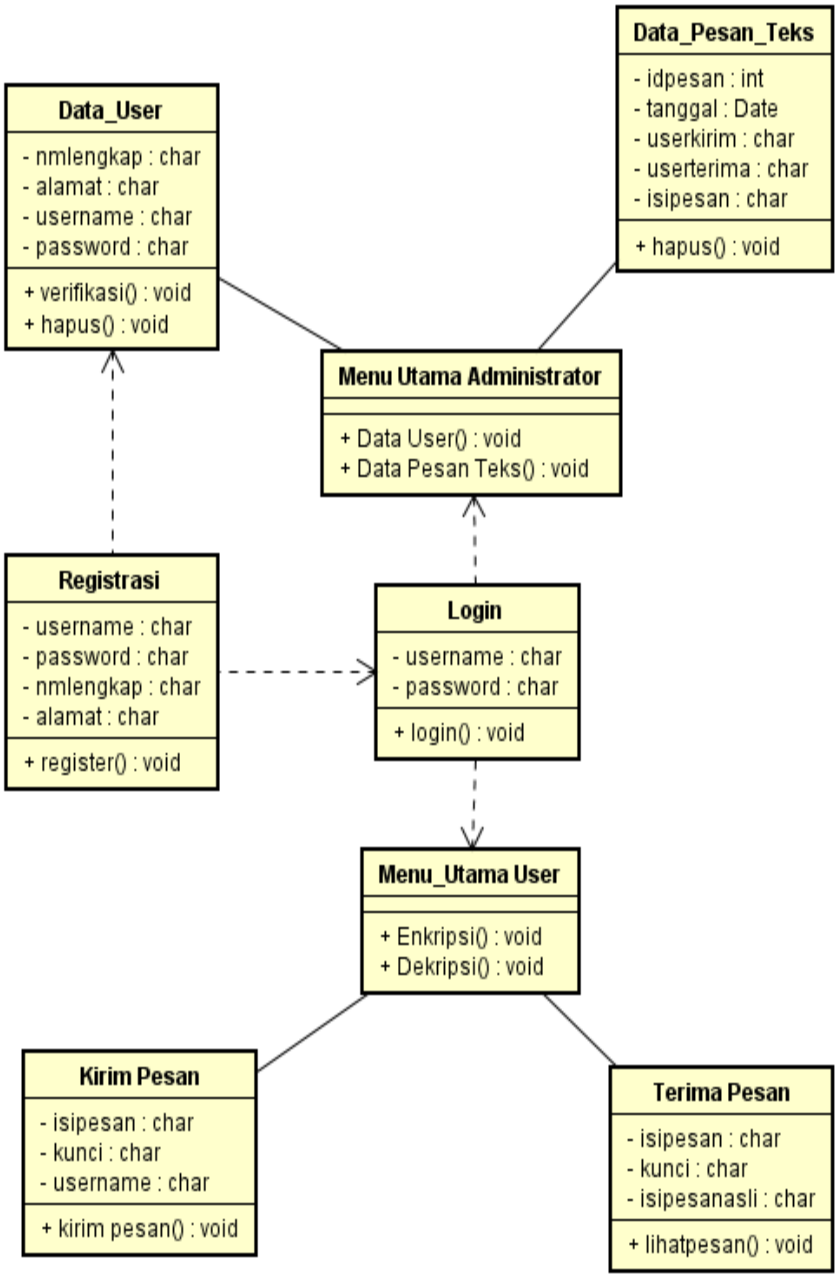
`

Gambar 3.3. Use Case Diagram

Pada Gambar 3.3. tersebut terdapat 2 aktor yang akan mengakses aplikasi yaitu administrator, user. Setiap user akan melakukan login pada aplikasi terlebih dahulu. Jika login sebagai administrator, maka admin dapat mengelola data user dan data pesan teks. Jika login sebagai user, user dapat menirim pesan teks dan melihat pesan masuk di terima pesan.

D. Class Diagram

`



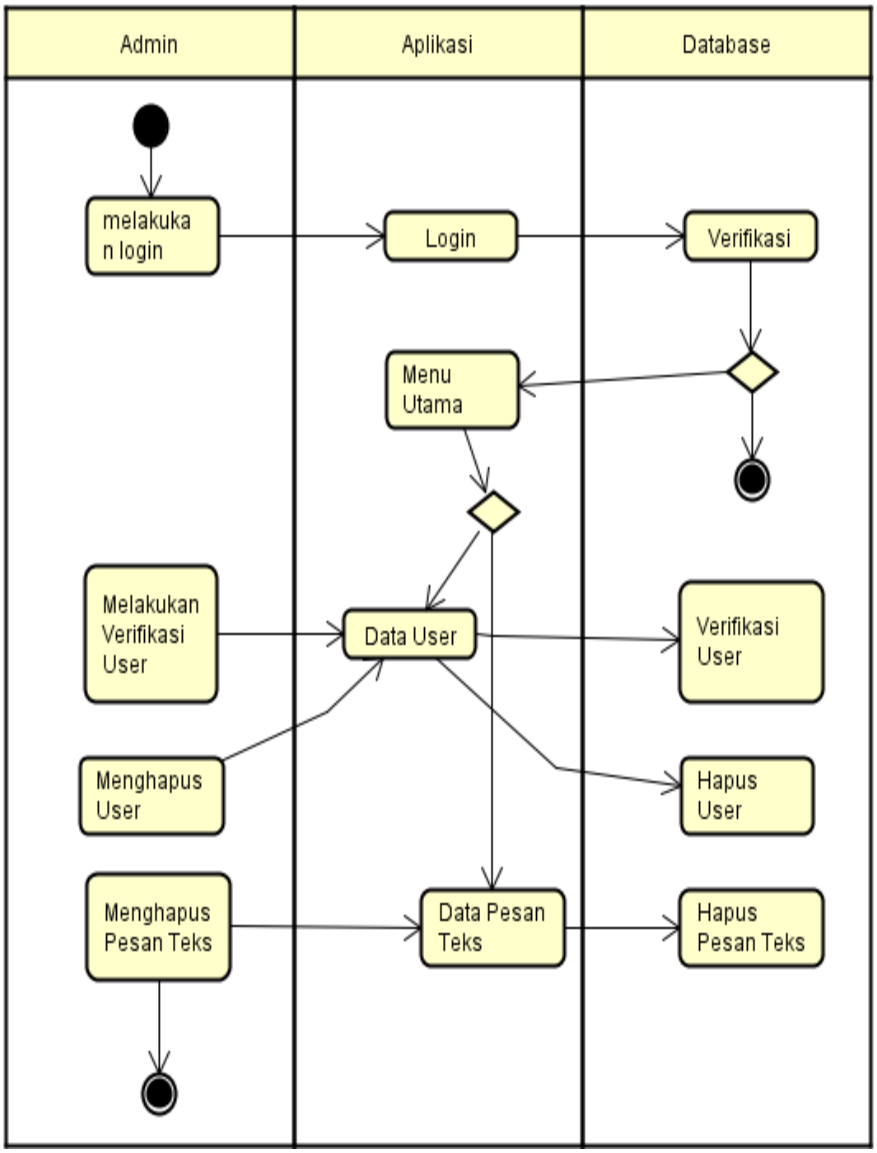
Gambar 3.4. Class Diagram

Pada Gambar 3.4. tersebut terdapat 8 class yang masing-masing memiliki atribut serta proses yang terjadi di dalam setiap class tersebut.

E. Activity Diagram

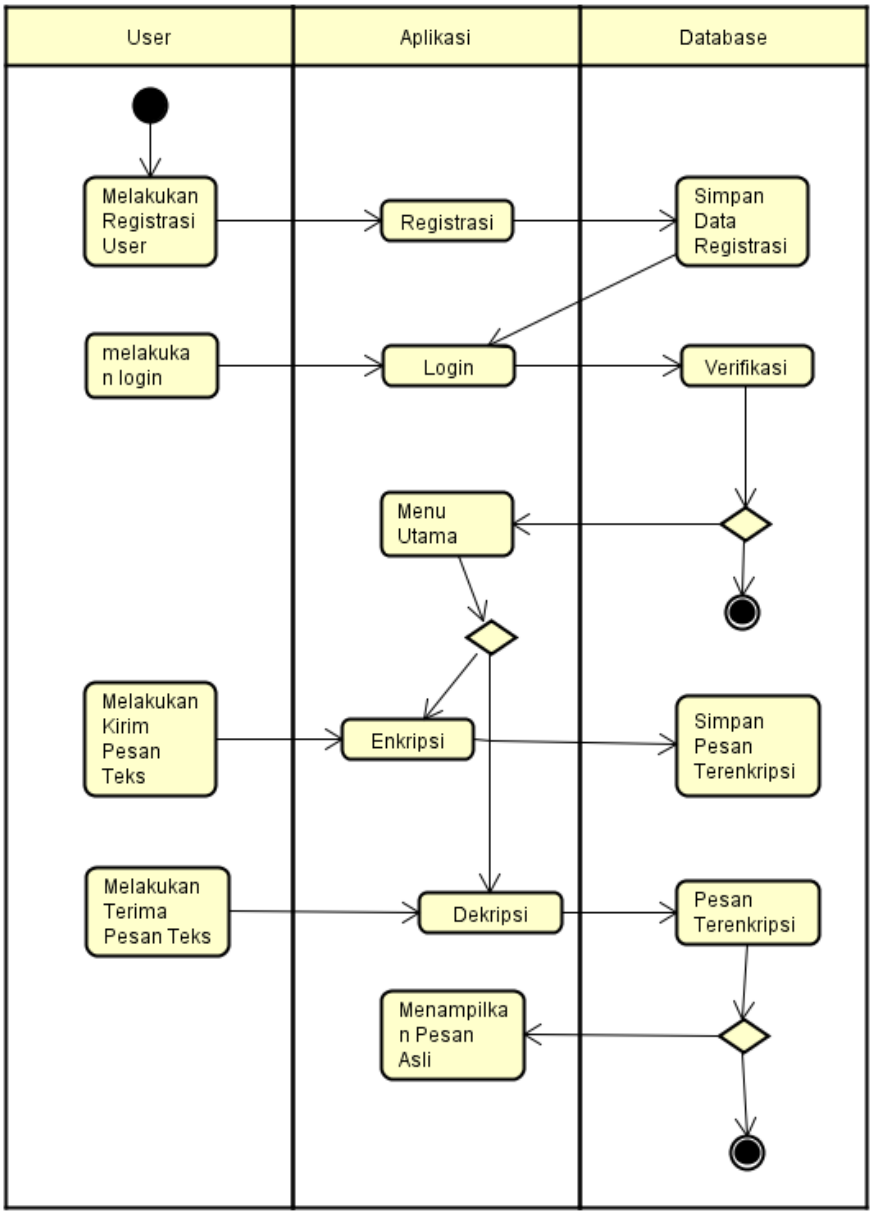
`

Activity Diagram menggambarkan aktivitas user terhadap aplikasi yang melibatkan user, aplikasi dan database. Adapun activity diagram tersebut seperti Gambar 3.5. dan Gambar 3.6



Gambar 3.5. Acitivity Diagram Administrator

`

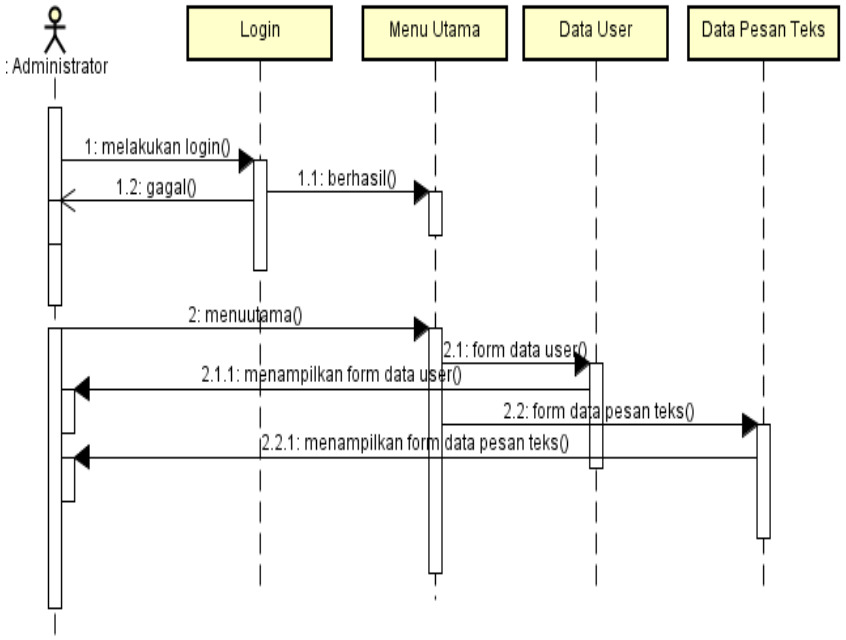


Gambar 3.6. Activity Diagram User

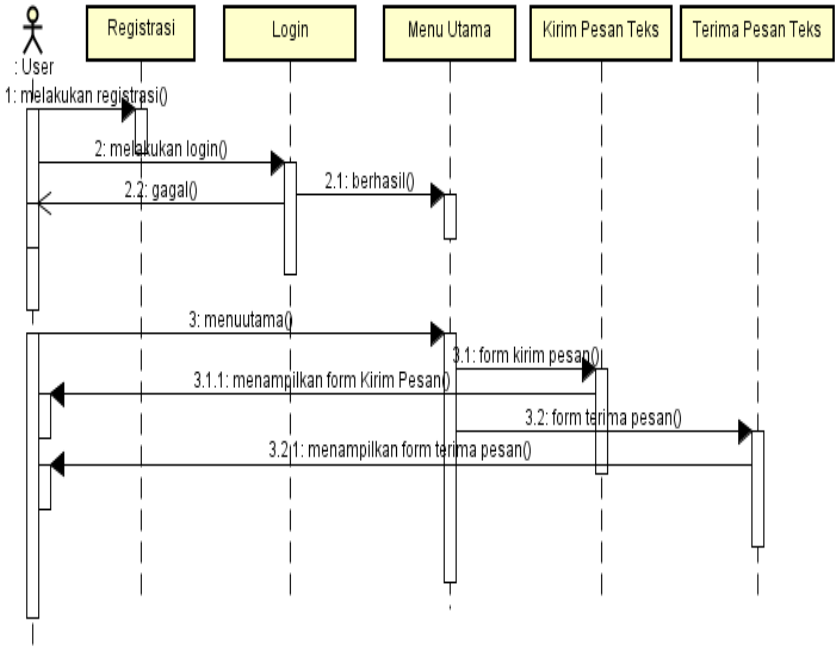
F. Sequence Diagram

Sequence Diagram menggambarkan keterhubungan antara user terhadap objek aplikasi. Adapun sequence diagram seperti Gambar 3.7. dan 3.8.

`



Gambar 3.7. Sequence Diagram Administrator



Gambar 3.8. Sequence Diagram User

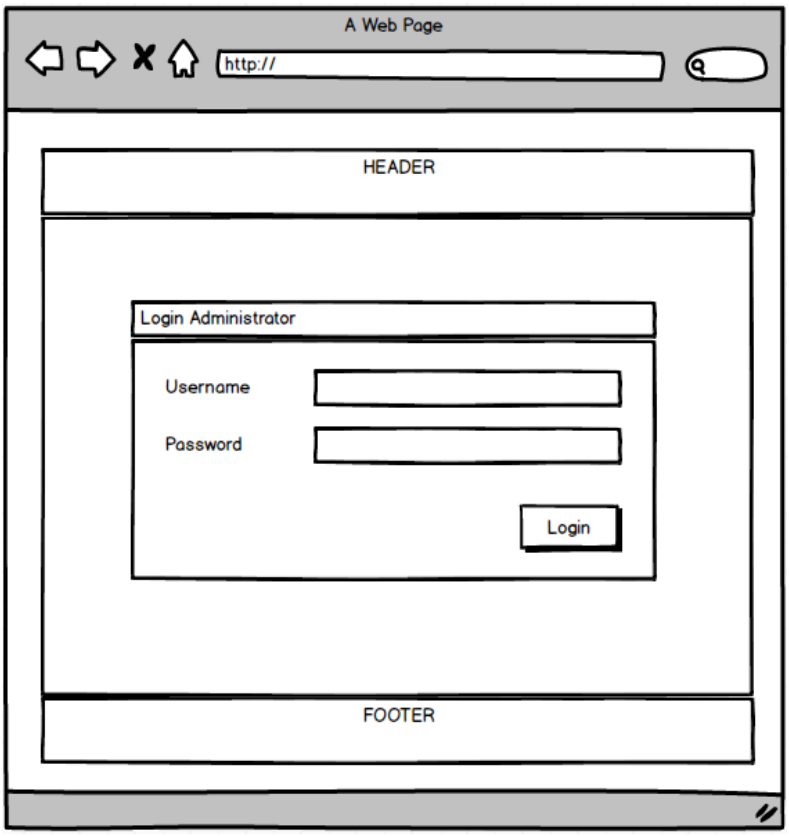
G. Perancangan Aplikasi Untuk Administrator

Perancangan aplikasi penyandian pesan teks berbasis web menggunakan algoritma blowfish untuk administrator, antara lain :

`

1) Homepage Web

Merupakan rancangan halaman web yang pertama kali muncul ketika menjalankan url domain aplikasi. adapun rancangan homepage seperti Gambar 3.9.

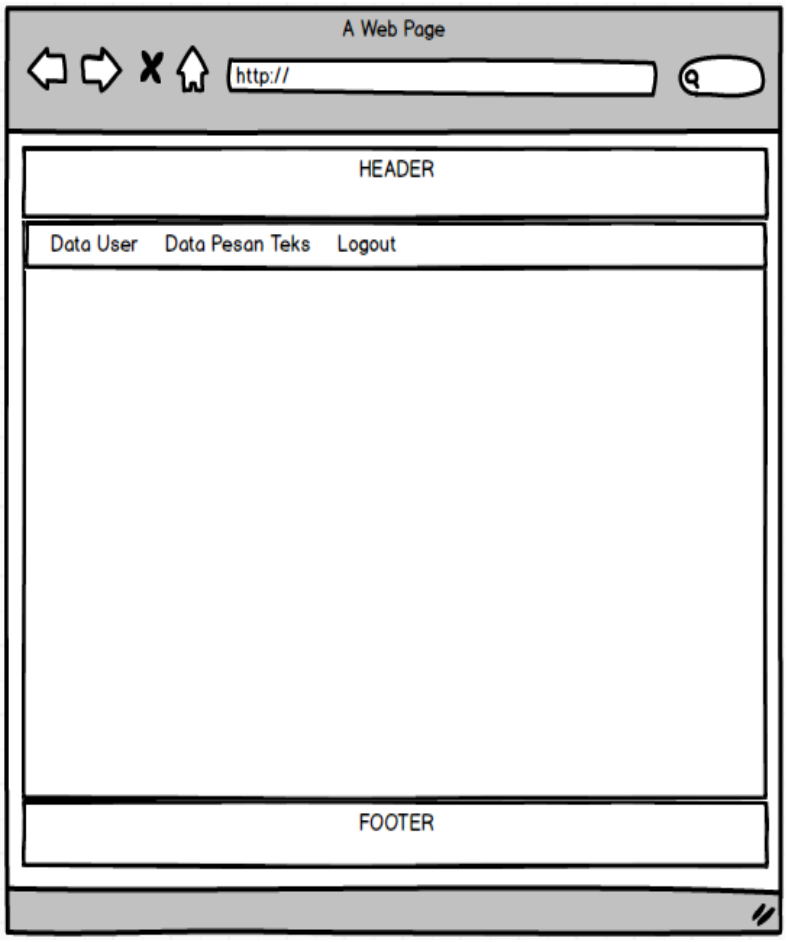


Gambar 3.9. Homepage Web

2) Halaman Menu Utama

Merupakan halaman menampilkan menu utama yang dapat diakses oleh admin ketika berhasil login. Sub menu tersebut yaitu data user, data pesan teks, dan logout yang memiliki fungsi berbeda-beda. Adapun halaman menu utama seperti Gambar 3.10.

`

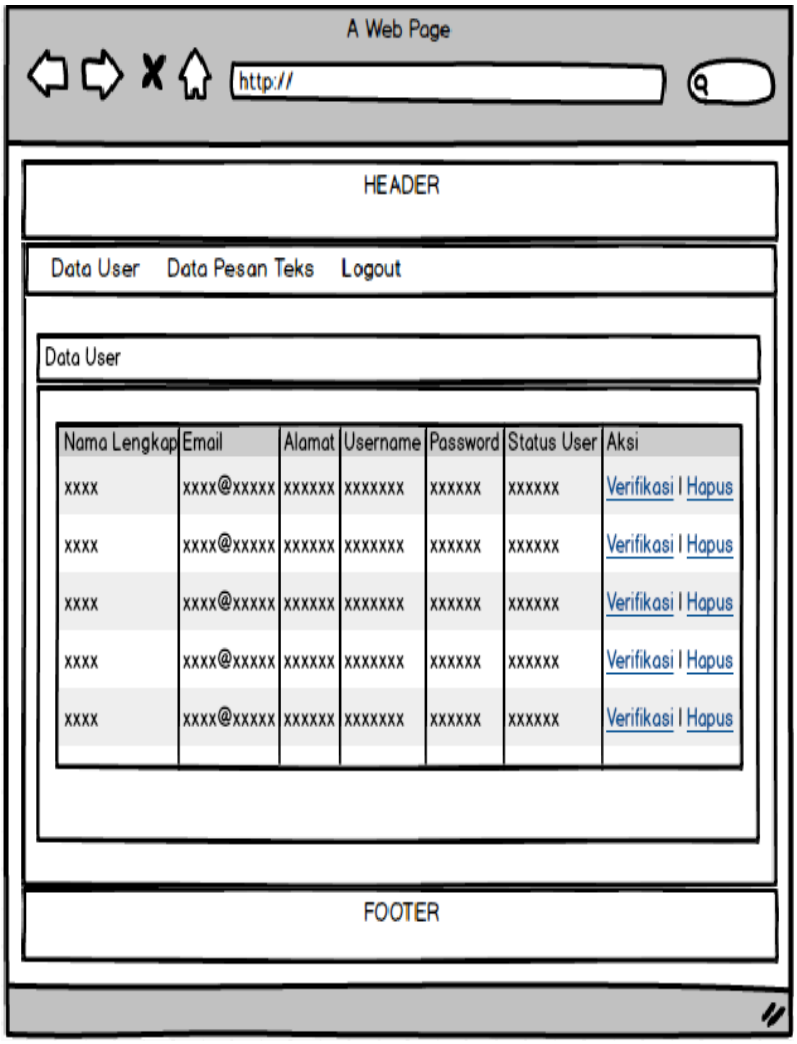


Gambar 3.10. Halaman Menu Utama

3) Halaman Data User

Merupakan halaman yang dapat diakses admin untuk melihat data user yang melakukan pendaftaran di web. Pada halaman ini terdapat tombol verifikasi yang digunakan untuk mengaktifkan user dan mengirim notifikasi ke email user sehingga user dapat menjalankan aplikasi dan mulai mengirim dan menerima pesan. Adapun halaman data user seperti Gambar 3.11.

`

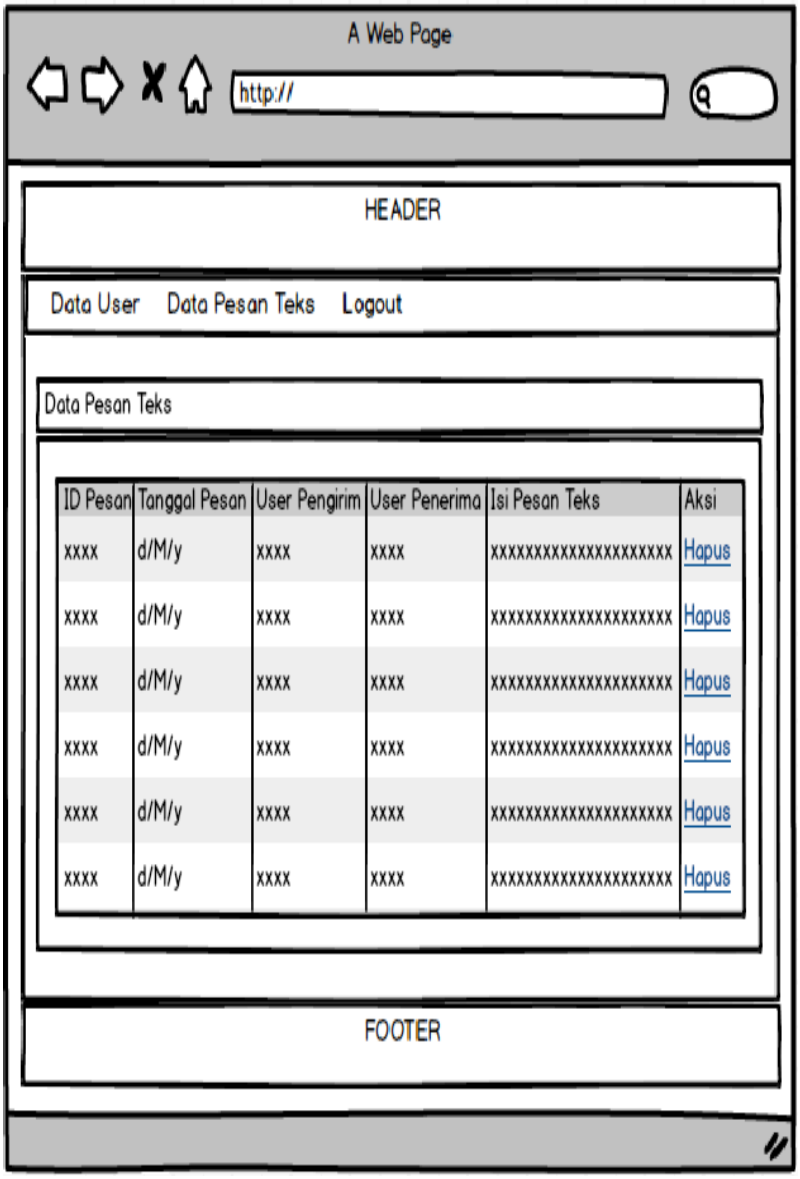


Gambar 3.11. Halaman Data User

4) Halaman Data Pesan

Merupakan halaman yang menampilkan informasi data pesan teks antara pengirim dan penerima. Pada halaman ini, admin dapat melakukan penghapusan data pesan teks tersebut. Adapun halaman data pesan seperti Gambar 3.12.

`



Gambar 3.12. Halaman Data Pesan

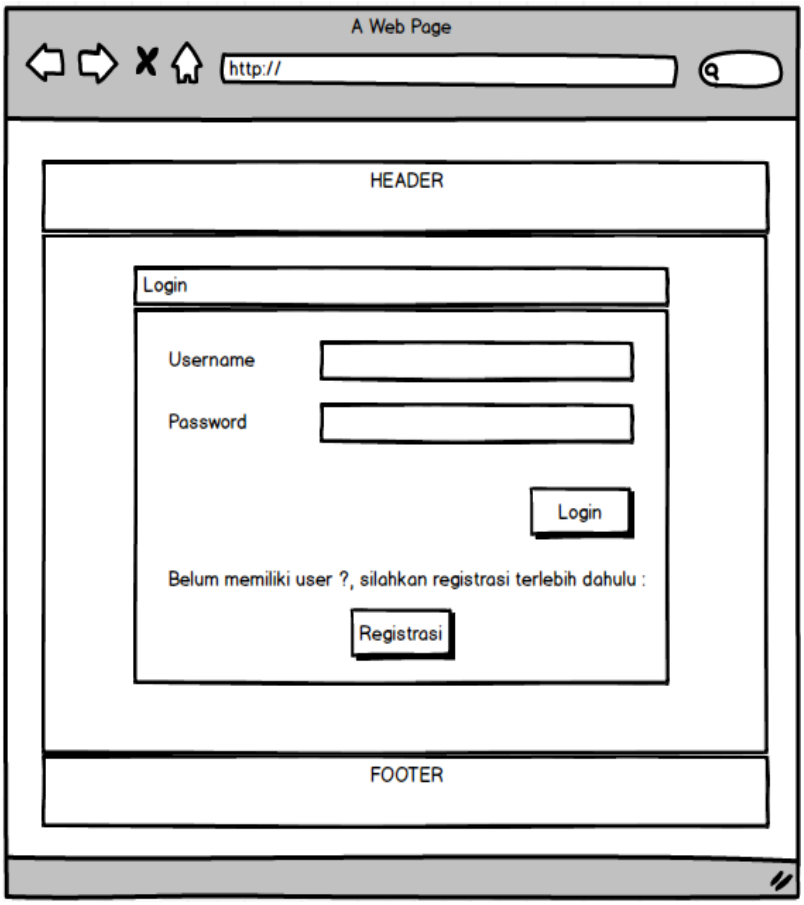
H. Perancangan Aplikasi Untuk User

Perancangan aplikasi penyandian pesan teks berbasis web menggunakan algoritma blowfish untuk user, antara lain :

1) Homepage Web

`

Merupakan rancangan halaman web yang pertama kali muncul ketika menjalankan url domain aplikasi. adapun rancangan homepage seperti Gambar 3.13.



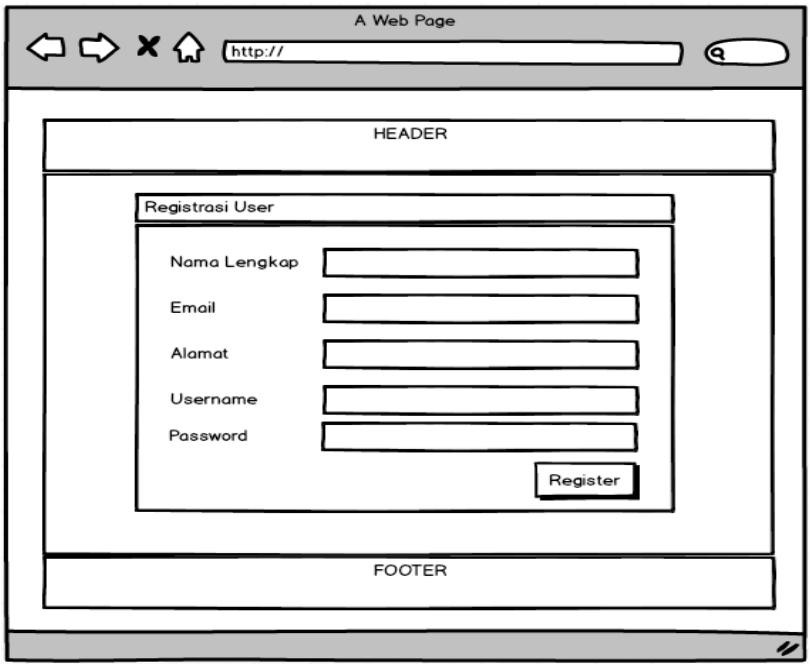
Gambar 3.13. Homepage Web

Pada halaman ini user akan dihadapkan dengan form login. Jika user sudah melakukan registrasi maka user memasukkan username dan password yang telah didaftarkan. Namun jika user belum melakukan registrasi, maka user harus klik registrasi.

2) Registrasi

`

Merupakan rancangan halaman registrasi user, dimana user akan mengisi biodata seperti pada field yang telah disediakan. Adapun rancangan halaman registrasi seperti Gambar 3.14

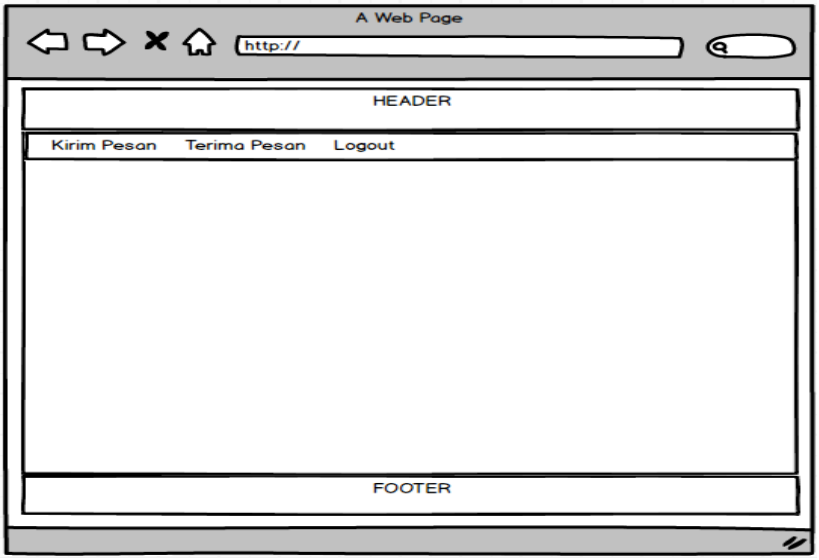


Gambar 3.14. Registrasi

3) Menu Utama

Merupakan rancangan halaman web yang tampil ketika user berhasil login. Pada menu utama terdapat 3 sub menu yang dapat diakses yaitu kirim pesan teks, lihat pesan teks, dan logout. Adapun rancangan halaman menu utama seperti Gambar 3.15.

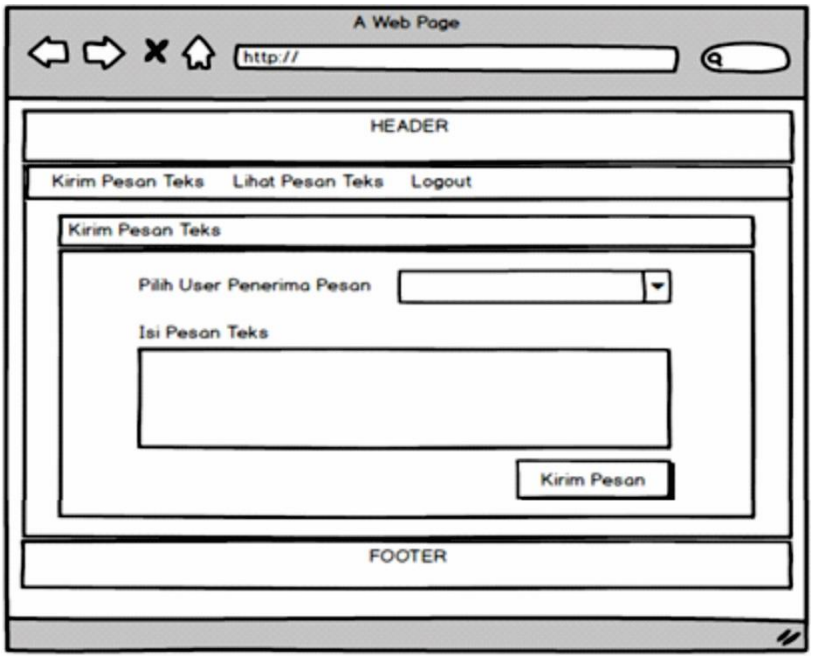
`



Gambar 3.15. Menu Utama

4) Kirim Pesan

Merupakan rancangan halaman web yang digunakan user untuk mengirim pesan teks pada user lain dengan cara memilih user yang telah terdaftar pada aplikasi. Setelah memilih penerima pesan, user mengisi pesan teks yang akan dikirim. Dimana pesan teks yang diisi yaitu pesan teks asli. Ketika klik tombol kirim pesan, secara otomatis pesan teks asli tersebut akan di enkripsi menggunakan Algoritma Blowfish menjadi pesan teks acak yang tidak dapat dibaca dan tersimpan di dalam database aplikasi. Adapun rancangan halaman kirim pesan teks seperti Gambar 3.16.

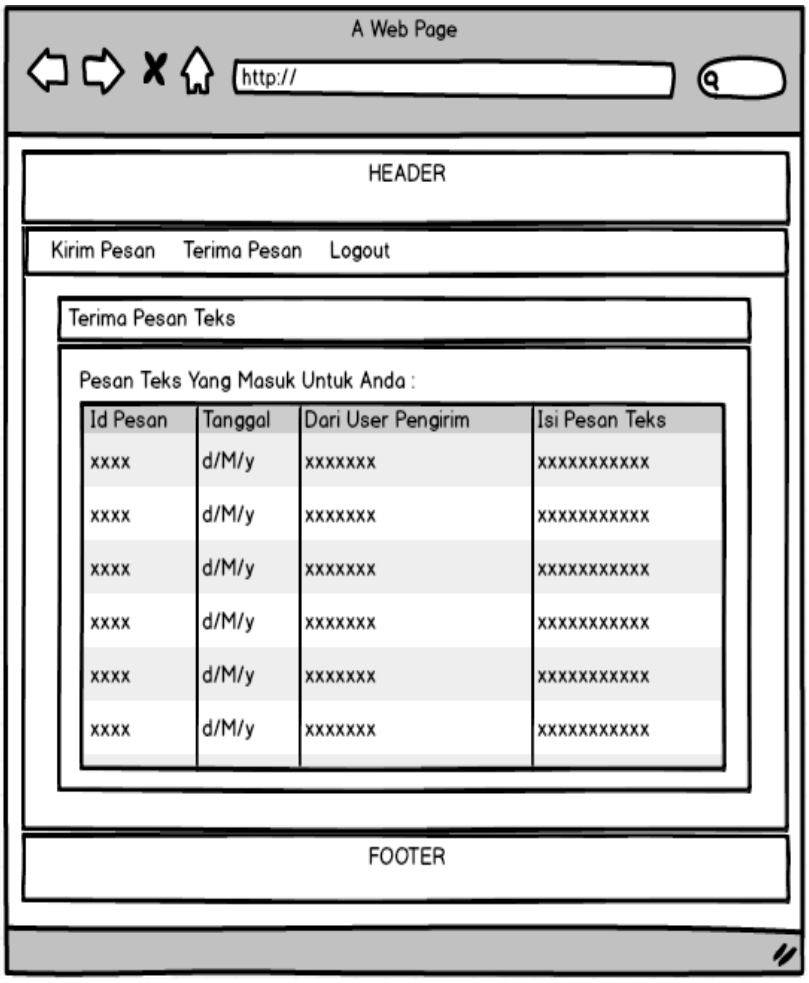


Gambar 3.16. Kirim Pesan Teks

5) Terima Pesan

Merupakan rancangan halaman web yang digunakan user untuk melihat pesan teks yang telah dikirim oleh pengirim. Pesan yang diterima ini masih berbentuk pesan teks acak yang tidak dapat dibaca, sehingga diperlukan proses dekripsi dengan memilih pesan teks dan klik lihat pesan. Secara otomatis pesan acak tersebut akan diubah ke pesan teks asli yang dapat dibaca oleh user. Adapun rancangan halaman terima pesan teks seperti Gambar 3.17

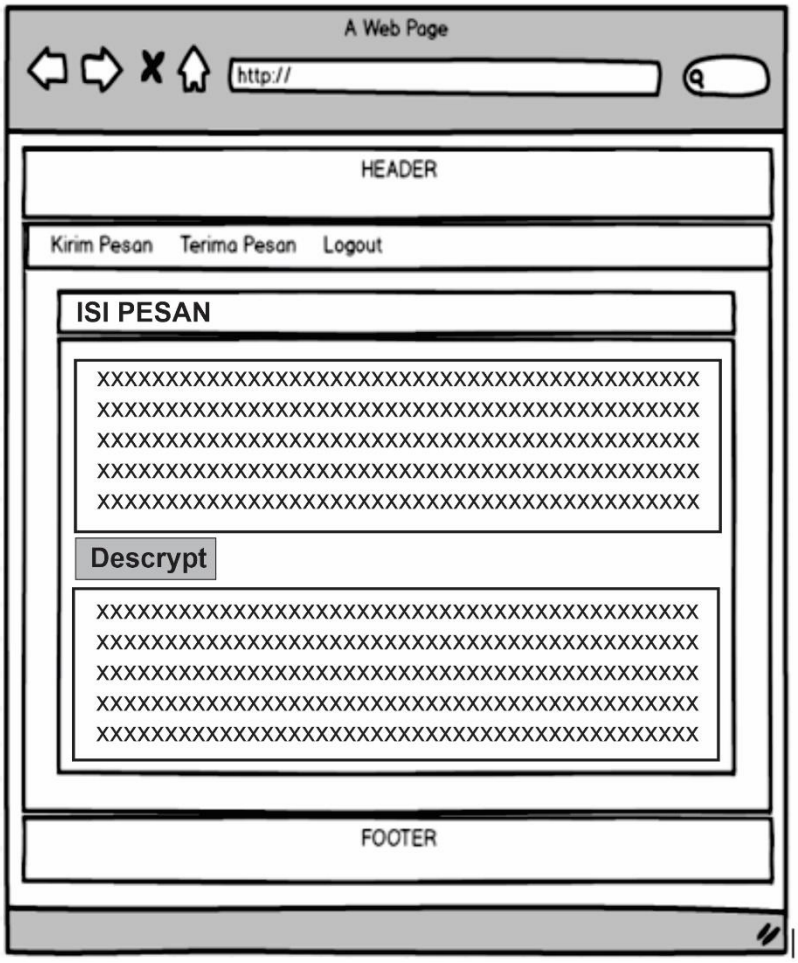
`



Gambar 3.17. Terima Pesan Teks

Dari tampilan diatas dapat dilihat detail isi pesan masuk dengan cara klik pengirim, maka isi pesan secara detail akan ditampilan, seperti dapat dilihat pada tampilan gambar dibawah ini:

`



Gambar 3.17. Tampilan Isi Pesan Masuk

Dari tampilan gambar diatas dapat dilihat isi pesan yang masuk. Dimana secara default pesan yang diterima akan tampil secara acak (sesuai metode blowfish), dan untuk melihat pesan aslinya dapat dilakukan dengan klik tombol descrypt.

3.6. Metode Pengujian Sistem

Pengujian sistem merupakan proses eksekusi sistem dengan tujuan mencari kesalahan atau kelemahan dari program tersebut. Proses tersebut dilakukan dengan mengevaluasi kemampuan program. Suatu program yang diuji akan dievaluasi apakah keluaran atau *output* yang dihasilkan telah sesuai dengan yang diinginkan atau tidak.

`

Metode pengujian yang dipakai dalam sistem ini adalah metode *black box*. Metode pengujian *black box* merupakan metode pengujian dengan pendekatan yang mengasumsikan sebuah sistem perangkat lunak atau program sebagai suatu kotak hitam. Adapun komponen yang diuji seperti Tabel 3.9.

Tabel 3.9. Komponen Yang Diuji

Form Yang Diuji	Skenario Pengujian	Hasil Pengujian
Form Login	Memasukkan username dan password yang benar	
	Memasukkan username atau password yang salah	
Form Registrasi	Menambahkan data registrasi user baru	
Form Kirim Pesan Teks	Mengirim pesan teks ke penerima	
	melihat *record* pesan yang dikirim pada tabel pesan di dalam database	
Form Lihat Pesan Teks	melakukan dekripsi pesan acak pada setiap pesan yang diterima	
Pengujian *wireshark* pada proses enkripsi dan dekripsi	Melakukan *capturing* paket masuk dan keluar menggunakan *wireshark*	

`