IMPLEMENTASI ALGORITMA *TWOFISH* PADA SISTEM INFORMASI PENGARSIPAN
(STUDI KASUS : PT. ANGKASA PURA I)

SKRIPSI

Oleh

Agil Bi Aviv Taufiqi

NIM 122410101059

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2017

IMPLEMENTASI ALGORITMA *TWOFISH* PADA SISTEM INFORMASI PENGARSIPAN (STUDI KASUS : PT. ANGKASA PURA I)

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Sistem Informasi (S1)
dan mencapai gelar Sarjana Komputer

Oleh
Agil Bi Aviv Taufiqi
NIM 122410101059

PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER
2017

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda dan Ibunda tercinta;
2. Kakak perempuan;
3. Adik laki-laki;
4. Sahabatku bersama dukungan dan doanya;
5. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
6. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTTO

"*You'll never know until you try it yourself*"

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Agil Bi Aviv Taufiqi

NIM : 122410101059

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "IMPLEMENTASI ALGORITMA *TWOFISH*PADA SISTEM INFORMASI PENGARSIPAN (STUDI KASUS : PT. ANGKASA PURA I)", adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember,

Yang menyatakan,

Agil Bi Aviv Taufiqi

NIM 122410101059

SKRIPSI

IMPLEMENTASI ALGORITMA *TWOFISH* PADA SISTEM INFORMASI PENGARSIPAN
(STUDI KASUS : PT. ANGKASA PURA I)

Oleh
Agil Bi Aviv Taufiqi
NIM 122410101059

Pembimbing :

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.APP., Sc., Ph.D
Dosen Pembimbing Pemdamping : Yanuar Nurdiansyah, ST., M.Cs.

PENGESAHAN PEMBIMBING

Skripsi berjudul "Implementasi Algoritma *Twofish* pada Sistem Informasi Pengarsipan (Studi Kasus PT. Angkasa Pura I)", telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,	Pembimbing II,
Drs. Antonius Cahya P, M.APP., Sc., Ph.D	Yanuar Nurdiansyah, ST., M.Cs.
NIP 196909281993021001	NIP 19820101 2010121004

PENGESAHAN PENGUJI

Skripsi berjudul "Implementasi Algoritma *Twofish* pada Aplikasi Pengarsipan (Studi Kasus PT. Angkasa Pura I)", telah diuji dan disahkan pada:

hari, tanggal :

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim penguji:

Penguji I,	Penguji II,
PENGUJI 1	PENGUJI 2
NIP.	NIP.

Mengesahkan

Ketua Program Studi,

Prof. Drs. Slamin,M.Comp.Sc.,Ph.D

NIP 196704201992011001

RINGKASAN

Implementasi Algoritma *Twofish*pada Sistem Informasi Pengarsipan (Studi Kasus PT. Angkasa Pura I

Pengarsipan adalah tata cara penyusunan arsip perusahaan dari awal didirikan sampai saat ini. Arsip bisa berisi data keuangan, data karyawan dan data lainnya sehingga manajemen arsip bisa dikatakan vital dalam satu perusahaan.

Resiko perkembangan teknologi yang sangat pesat, memungkinkan seseorang yang tidak memiliki hak akses untuk mengambil dan memanfaatkan informasi tanpa ijin. Resiko tersebut mengancam keamanan data arsip perusahaan yang akan berakibat fatal terhadap perusahaan.

PT. Angkasa Pura I saat ini dalam proses manajemen arsip memerlukan waktu yang lama dan pencatatannya menggunakan *microsoft excel* tanpa fitur keamanan. Proses manajemen yang lama dikarenakan semua proses pengarsipan harus melalui satu petugas pengarsipan, dimana setiap divisi harus menemui petugas pengarsipan baik dalam pencatatan maupun melihat daftar berkas arsip. Proses manajemen yang lama dapat diatasi dengan menggunakan sistem yang memliki beberapa hak akses, sehingga dalam proses manajemen arsip setiap divisi tidak perlu menemui petugas pengarsipan.

Algoritma *twofish* merupakan algoritma kuat yang sampai saat ini dinyatakan aman karena masih belum ada serangan kriptanalisis yang benar–benar dapat mematahkan algoritma ini. Oleh karena itu, penelitian ini akan menggunakan *twofish* sebagai algoritma enkripsinya.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul "Implementasi Algoritma *Twofish*pada Sistem Informasi Pengarsipan (Studi Kasus PT. Angkasa Pura I)". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Drs. Antonius Cahya Prihandoko, M.APP., Sc., Ph.D., selaku Dosen Pembimbing Utama dan Yanuar Nurdiansyah, ST., M.Cs., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Drs. Antonius Cahya Prihandoko, M.APP., Sc., Ph.D.,sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ayahanda dan Ibunda yang selalu mendukung dan mendoakan;
6. Kakak
7. Adik
8. Keluarga Besar
9. Sahabat
10. Teman warnet maxima jalan jawa
11. Semua pihak yang tidak dapat disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak.Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember,

Penulis

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Pengarsipan adalah tata cara penyusunan arsip perusahaan dari awal didirikan sampai saat ini. Arsip dapat berupa laporan dari setiap keputusan atau aturan dari satu perusahaan dapat berupa data keuangan dan data karyawan, sehingga data arsip merupakan data vital dalam satu perusahaan.

Perkembangan teknologi saat ini, pada pengarsipan dapat dilakukan secara komputerisasi dengan menggunakan sistem, sehingga manajemen data arsip dapat dilakukan dengan baik. Penyimpanan arsip saat ini tidak hanya dalam bentuk fisik tetapi dalam bentuk file digital untuk mempermudah dalam mengelola data arsip.

Resiko perkembangan teknologi yang sangat pesat, memungkinkan seseorang yang tidak memiliki hak akses untuk mengambil dan memanfaatkan informasi tanpa ijin. Resiko tersebut mengancam keamanan data arsip perusahaan yang akan berakibat fatal terhadap perusahaan.

Keamanan informasi data arsip dapat atasi dengan sistem yang memiliki fitur untuk menyamarkan data, sehingga informasi arsip tidak mudah dibaca. Fitur keamanan dengan cara menyamarkan data disebut enkripsi. Enkripsi adalah proses yang dilakukan untuk mengamankan sebuah pesan yang disebut *plaintext* menjadi pesan yang tersembunyi yang disebut *ciphertext* (Primartha, 2011). Metode enkripsi memiliki banyak algoritma yang dapat digunakan diantaranya algoritma *rijndael*, *serpent*, *twofish*, MARS, RC6, *blowfish*, DES dan 3DES.

Algoritma enkripsi yang menjadi finalis dari kompetisi untuk menetapkan algoritma enkripsi standar (AES) antara lain algoritma *rijndael*, *serpent*, *twofish*, MARS, dan RC6. Analisa performa dari segi penggunaan memori, algoritma *rijndael* yang paling sedikit, sedangkan dari segi keamanan dan kecepatan algoritma *twofish* yang lebih unggul dari algoritma lainnya (Schneier & Whiting, 2000). Sistem pengarsipan lebih mengutamakan kemanan data daripada

penggunaan memori. Oleh karena itu, algoritma twofish lebih tepat digunakan dalam sistem pengarsipan.

Algoritma *twofish* merupakan algoritma kuat yang sampai saat ini dinyatakan aman karena masih belum ada serangan kriptanalisis yang benar–benar dapat mematahkan algoritma ini (Mukmin, 2007). Algoritma yang beroperasi dalam mode *block cipher* ini juga tidak dipatenkan sehingga penggunaannya pada alat enkripsi tidak perlu mengeluarkan biaya. Algoritma *twofish* menggunakan jaringan *feistel* 16 putaran dan 4 kotak-S yang bergantung pada *key*.Terdapat empat macam *key* schedule dalam implementasinya yaitu: *full keying, partial keying, minimal keying*, dan *zero keying* dengan perbedaan dalam *key setup*. *Twofish* juga memiliki beberapa metode pengacakan yaitu matriks MDS, teknik PHT dan teknik *whitening*.

PT. Angkasa Pura I saat ini dalam proses manajemen arsip memerlukan waktu yang lama dan pencatatannya menggunakan *microsoft excel* tanpa fitur keamanan. Proses manajemen yang lama dikarenakan semua proses pengarsipan harus melalui satu petugas pengarsipan, dimana setiap divisi harus menemui petugas pengarsipan baik dalam pencatatan maupun melihat daftar berkas arsip. Proses manajemen yang lama dapat diatasi dengan menggunakan sistem yang memliki beberapa hak akses, sehingga dalam proses manajemen arsip setiap divisi tidak perlu menemui petugas pengarsipan.

Permasalahan dalam PT. Angkasa Pura I tersebut yaitu manajemen pengarsiapan dan keamanan data dapat diatasi dengan membangun sistem yang memiliki hak akses dan mampu menyamarkan data arsip. Penggunaan sistem dalam proses manajemen pengarsipan akan lebih cepat dan efisien karena setiap divisi dapat mencatat dan melihat daftar arsip tanpa harus melalui petugas pengarsipan. Proses pengamanan data menggunakan enkripsi algoritma twofish karena dari segi keamanan algoritma ini lebih unggul dari metode lainnya dan sistem pengarsipan lebih mengutamakan keamanan.

1.2 Rumusan Masalah

Berdasarkan dari beberapa permasalahan yang telah diuraikan diatas, maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana cara mengimplementasikan algoritma *twofish* dalam sistem pengarsipan?
2. Bagaimana keamanan sistem terhadap serangan *SQL injection, cross site scripting* dan *decryptor online*?

1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

1.3.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang dan membangun sebuah aplikasi pengarsipan yang terlindungi.
2. Mengetahui cara menggunakan algoritma *twofish* pada sistem informasi pengarsipan.

1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat Akademis

   Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Universitas Jember.
2. Manfaat bagi peneliti

   Mengetahui bagaimana cara mengimplementasikan algoritma *twofish* pada sistem informasi pengarsipan.

3. Manfaat bagi masyarakat

   Memberikan pengetahuan tentang pentingnya sistem enkripsi dan cara kerja dari algoritma *twofish*.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan penelitian ini diataranya sebagai berikut :

1. Aplikasi yang dibangun berbasis *web.*
2. *Output* yang dihasilkan adalah aplikasi yang dapat menjaga keamanan data arsip sebuah perusahaan yang menggunakannya..
3. Teknik kriptografi yang digunakan untuk mengamankan nama dan tempat penyimpanan data arsip adalah *twofish*.

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan

   Bab kesatu ini memuat uraian tentang latar belakang, perumusan masalah, tujuan,manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masingtertuang secara eksplisit dalam subbab tersendiri.

2. Tinjauan Pustaka

   Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

3. Metode Penelitian

   Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangansistem yang digunakan dalam penelitian.

4. Pengembangan Sistem

   Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, pengkodean, dan pengujian sistem.

5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yangdisajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dangambar untuk memperjelas hasil penelitian.

6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saranuntuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

2.1 Penelitian Terdahulu

Pada penelitian sebelumnya algoritma *twofish* dianalisis dan diimplementasikan untuk penyandian citra digital. Berdasarakan implementasi yang telah dilakukan terhadap citra digital tersebut, algoritma *twofish* memiliki nilai *avalanche effect* mencapai 34,1%. Berdasarkan nilai tersebut menunjukkan bahwa algoritma *twofish* pada proses enkripsi citra digital cukup baik. Sedangkan untuk proses dekripsi pada algoritma ini juga menghasilkan citra yang sangat mirip dengan citra aslinya sebelum dilakukannya proses enkripsi. Tingkat kemiripannya mencapai 99,9%.

2.2 Kriptografi

Kriptografi berasal dari bahasa Yunani, "*kryptós*" yang berarti tersembunyi dan "*gráphein*" yang berarti tulisan. Sehingga kata kriptografi dapat diartikan berupa frase "tulisan tersembunyi". Menurut *Request for Comments* (RFC), kriptografi merupakan ilmu matematika yang berhubungan dengan transformasi data untuk membuat artinya tidak dapat dipahami (untuk menyembunyikan maknanya), mencegahnya dari perubahan tanpa izin, atau mencegahnya dari penggunaan yang tidak sah. Jika transformasinya dapat dikembalikan, kriptografi juga bisa diartikan sebagai proses mengubah kembali data yang terenkripsi menjadi bentuk yang dapat dipahami. Dalam pengertian yang luas, kriptografi dapat dikatakan sebagai proses untuk melindungi data.

2.3 Algoritma *Twofish*

Algoritma *twofish* menggunakan desain yang mudah dan tidak memiliki kunci lemah (Randy, 2010). Desain yang mudah akan mempengaruhi kecepatan proses enkripsi dan dekripsi. Tidak adanya kunci lemah pada algoritma ini

membuat kunci apapun yang menjadi masukkan oleh pengguna, tingkat keamanannya akan tetap sama. *Key* (kunci) merupakan elemen yang sangat penting dalam enkripsi dan harus dijaga kerahasiannya. *Key* memberikan cara khusus bagaimana suatu algoritma mengubah *plaintext* menjadi *ciphertext* dan sebaliknya. Algoritma ini menggunakan *key* yang sama pada proses enkripsi dan dekripsinya yang disebut *symmetric key*. *Twofish* menggunakan 16-*round* struktur seperti jaringan *feistel* dengan penambahan *whitening* pada input dan output (Setiawan, 2011). Satu-satunya elemen bukan *feistel* adalah pergeseran satu bit. Pergeseran ini dapat dipindahkan ke dalam fungsi F untuk membuat sebuah jaringan *feistel* yang sesungguhnya, namun hal ini memunculkan pergeseran tambahan dari *words* sebelum langkah *whitening* keluaran. *Plaintext* dibagi menjadi empat buah 32-bit *words*.

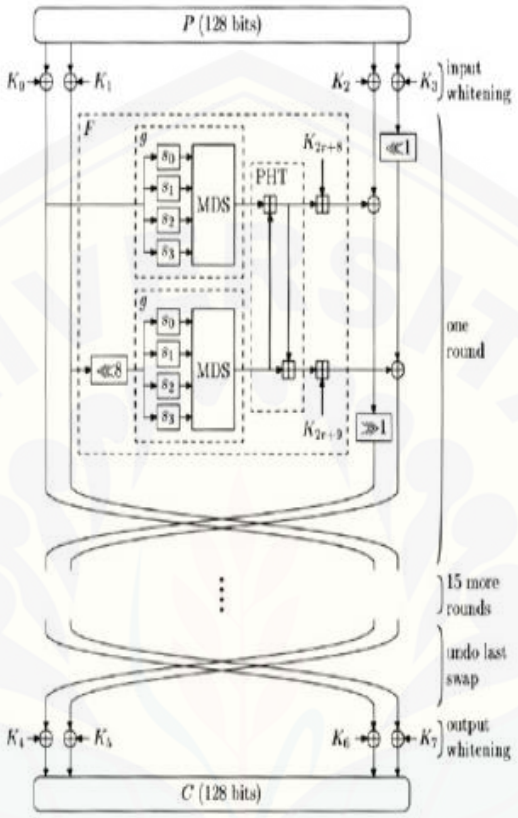
Pada langkah *whitening* masukkan, *plaintext* ini diXORkan dengan 4 *words* kunci. Langkah ini dilanjutkan sebanyak enam belas *round*, yang pada tiap *round*-nya dua *words* di kiri digunakan sebagai masukan untuk fungsi g (salah satunya terlebih dahulu digeser sebanyak 8 bit ke kiri). Fungsi g terdiri dari empat kotak-S yang bergantung pada kunci, dan diikuti langkah pencampuran linear berbasis matriks MDS. Hasil dari dua fungsi g ini dikombinasikan dengan menggunakan transformasi *Pseudo-Hadamard* (PHT) dan dua *key words*ditambahkan. Dua hasil ini diXORkan dengan *words* di sebelah kanan (salah satunya terlebih dahulu digeser sejauh satu bit ke kiri, dan satu lagi digeser satu bit ke kanan setelah diXORkan). Bagian kiri dan kanan lalu dipertukarkan untuk *round* berikutnya. Setelah semua *round* selesai dilakukan, pertukaran terakhir dikembalikan dan empat *words* tersebut diXORkan dengan empat *keywords* menghasilkan *ciphertext* seperti pada gambar 2.1.

Untuk lebih formalnya, 16 *bytes* dari *plaintext* p0,…,p15 dibagi menjadi 4 words P0,…,P3 dengan setiap *words*-nya menggunakan konvensi *little-endian* seperti pada persamaan 1.

$$P_i = \sum_{j=0}^{3} p_{(4i+j)} - 2^{8j} \qquad i = 0, \dots, 3 \qquad \text{(persamaan 1)}$$

Pada langkah *whitening* masukkan, *words* ini diXORkan dengan 4 *words* kunci yang telah diekspansi seperti pada persamaan 2.

$$R_{0,i} = P_i \oplus K_i \quad i = 0, \dots, 3 \qquad \text{(persamaan 2)}$$



Gambar 2.1 Garis Besar Algoritma *Twofish*

Persamaan 3 menjelaskan bahwa di tiap 16 *rounds*, dua *words* yang pertama digunakan sebagai masukan untuk fungsi F, yang juga menerima nomor *round* sebagai masukkan. *Word* yang ketiga diXORkan dengan keluaran pertama dari F, lalu digeser ke kanan sejauh 1 bit seperti pada persamaan 4. *Word* yang keempat digeser ke kiri sejauh satu bit, lalu diXORkan dengan keluaran kedua dari fungsi F seperti pada persamaan 5. Setelah semua langkah selesai dilakukan, kedua sisi tersebut dipertukarkan seperti pada persamaan 6.

$$R_{r+1,0} = ROR \oplus (R_{r,2}F_{r,0}, 1) \qquad \text{(persamaan 3)}$$

$$R_{r+1,1} = ROL \oplus (R_{r,3}, 1)F_{r,1} \qquad \text{(persamaan 4)}$$

$$R_{r+1,2} = \mathrm{R_{r,0}} \qquad \text{(persamaan 5)}$$

$$R_{r+1,3} = \mathrm{R_{r,1}} \qquad \text{(persamaan 6)}$$

Dengan r = 0,…,15 dan ROR dan ROL adalah fungsi yang menggeser argumen pertamanya (sebuah *word* 32-bit) ke kiri atau ke kanan sesuai dengan jumlah bit yang diindikasikan oleh argumen ke-duanya. Langkah *whitening* keluaran membatalkan pertukaran *round* terakhir dan mengXORkan data *word* dengan 4 *words* kunci yang telah diekspansi seperti pada persamaan 7.

$$C_i = R_{16,(i+2) mod\ 4} K_{i+4} \qquad i = 0, \dots, 3 \qquad \text{(persamaan 7)}$$

Empat *words* dari *ciphertext* kemudian ditulis sebagai 16 *bytes* c0,…,c15 menggunakan konvensi *little-endian* yang sama dengan yang digunakan untuk *plaintext* seperti pada persamaan 8.

$$C_i = \left[\frac{C_{(i/4)}}{2^{8(i mod\ 4)}}\right] mod 2^8 \quad i = 0, \dots, 15 \qquad \text{(persamaan 8)}$$

2.4 Penerapan Algoritma dalam Sistem

Sistem informasi pengarsipan akan mengenkripsi data arsip yang telah diinputkan oleh admin menggunakan algoritma *twofish*dimana proses enkripsinya terdapat pada sisi *client*. Data yang dienkripsi adalah nama data arsip dan tempat penyimpanan datanya.

BAB 3. METODOLOGI PENELITIAN

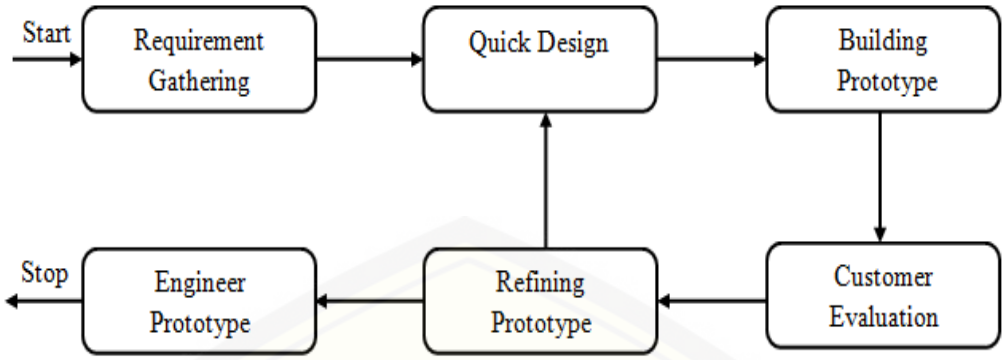
Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tahapan dari penelitian.

3.1. Jenis Penelitian

Pada penelitian ini, jenis penelitian yang digunakan ialah penelitian pengembangan, dikarenakan penelitian ini membuat dan mengembangkan suatu produk, dan penelitian ini bukan penelitian yang dimaksudkan untuk menemukan teori atau mengujikan kebenaran suatu teori atau konsep dalam bentuk eksperimentasi.

3.2. Tahap Perancangan

Metode yang digunakan dalam penelitian ini yakni SDLC (*System Development Life Cycle*) dengan model *prototype*. Menurut (Naresh Kumar, 2013), model *prototype* dibangun untuk memenuhi kebutuhan *client* . Selain tidak menunggu agar kebutuhan *client* tedaftar sepenuhnya, model *prototype* ini bisa langsung melanjutkan proses koding dan juga desain. Tujuan dari *prototype* ini adalah mempersilahkan *user* untuk mengevaluasi desain *product* dengan menggunakan *prototype* program sementara secara langsung daripada menginstruksikan bayangan program dengan basis desain di atas kertas. Alur perancangan model *prototype* dapat dilihat pada gambar 3.1.



Gambar 3.1 Prototype Model

3.2.1 Analisis Kebutuhan

Pada tahap ini, hal yang dilakukan adalah merumuskan solusi dari permasalahan yang muncul. Data dan permasalahan diperoleh dari wawancara, studi sistem yang telah ada, dan menganalisis dokumen – dokumen yang terkait. Selanjutnya menganalisis kebutuhan yang harus dipenuhi oleh program yang akan dibangun, menentukan kebutuhan fungsional dan non fungsional serta menentukan fungsi dan fasilitas apa saja yang dibutuhkan.

3.2.2 *Quick Desain*

Pembuatan desain sistem pada penelitian ini menggunakan *Unified Modeling Languange* (UML) yang dirancang dengan konsep *Object Oriented Desain*. Pemodelan UML yang digunakan adalah sebagai berikut:

1. *Business Process*

Business Process merupakan diagram yang menggambarkan proses yang lengkap. Isi dari *business process* adalah *resources* dan informasi yang dibutuhkan pada penelitian ini, *event* yang mendorong terjadinya proses dan *goal* yang dituju.

2. *Use Case Diagram*

Usecase adalah model yang menggambarkan fungsi atau tugas yang dilakukan oleh user, baik manusia maupun mesin / komputer. *Use case model* ini

dapat digunakan untuk menggambarkan *job spesification* dan *job description*, serta keterkaitan antar *job*.

3. *Scenario*

Scenario Diagram berfungsi untuk menjelaskan alur sistem dari fitur yang ada di *job specification* dan *job description* yang ada pada *Use Case Diagram*. *Scenario* menjelaskan alur sistem dan keadaan yang akan terjadi ketika terjadi suatu *event* tertentu.

4. *Sequence Diagram*

Sequence diagram menggambarkan aliran logika interaksi antar objek yang mengindikasikan komunikasi antar obyek di dalam system yang disusun pada suatu urutan atau rangkaian waktu.

5. *Activity Diagram*

Activity Diagram digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi. *Activity diagram* mempunyai fungsi yang sama dengan *scenario* namun diimplementasikan dalam diagram alir.

6. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi class serta hubungannya antar class, sehingga memudahkan dalam proses pengkodean.

7. *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan obyek – obyek dasar data yang mempunyai hubungan antar relasi.

3.2.3 *Building Prototype*

Tahap ini mengimplentasikan desain yang akan menjadi sebuah *prototype* aplikasi berbasis *web* untuk mengelola data arsip sebuah perusahaan dan mengimplementasikan algoritma kriptografi yang telah dijelaskan di atas pada aplikasi tersebut. Hal-hal yang dilakukan dalam tahap ini adalah menulis kode program menggunakan bahasa pemrograman PHP.

3.2.4 *Customer Evaluation*

C*ostumer evaluation* merupakan tahap dimana *developer* (pada kasus ini peneliti) mempresentasikan perkembangan dalam pembuatan sistem pada *customer* (pada kasus ini orang yg bertanggung jawab pada arsip). Tahap ini didapat pendapat dari *customer* tentang sistem yang sedang dibangun.

3.2.5 *Refining Prototype*

Setelah dievaluasi oleh *customer*, *prototype* akan diperbaiki sesuai dengan hasil evaluasi *customer.* Apabila sudah cocok dengan keinginan *customer*, maka *software* akan dibangun sesuai dengan *prototype* tersebut.

3.2.6 *Engineer Prototype*

Membuat *software* berdasarkan *prototype*yang telah dievaluasi dan telah melalui proses testing diikuti dengan *maintenance* secara rutin untuk mencegah kegagalan berskala besar.

3.3. Pengujian Keamanan Sistem

Pengujian keamanan sistem pada penelitian ini menggunakan tiga metode yaitu: SQL *Injection Attack, Cross Site Scripting* dan *Decryptor Online*.

3.3.1 SQL *Injection Attack*

SQL *Injection Attack* adalah sebuah teknik memasukkan perintah SQL yang menyebabkan penyerang dapat mengakses *database* sistem dan memanipulasinya (Kannan, 2011).

3.3.2 *Cross Site Scripting*

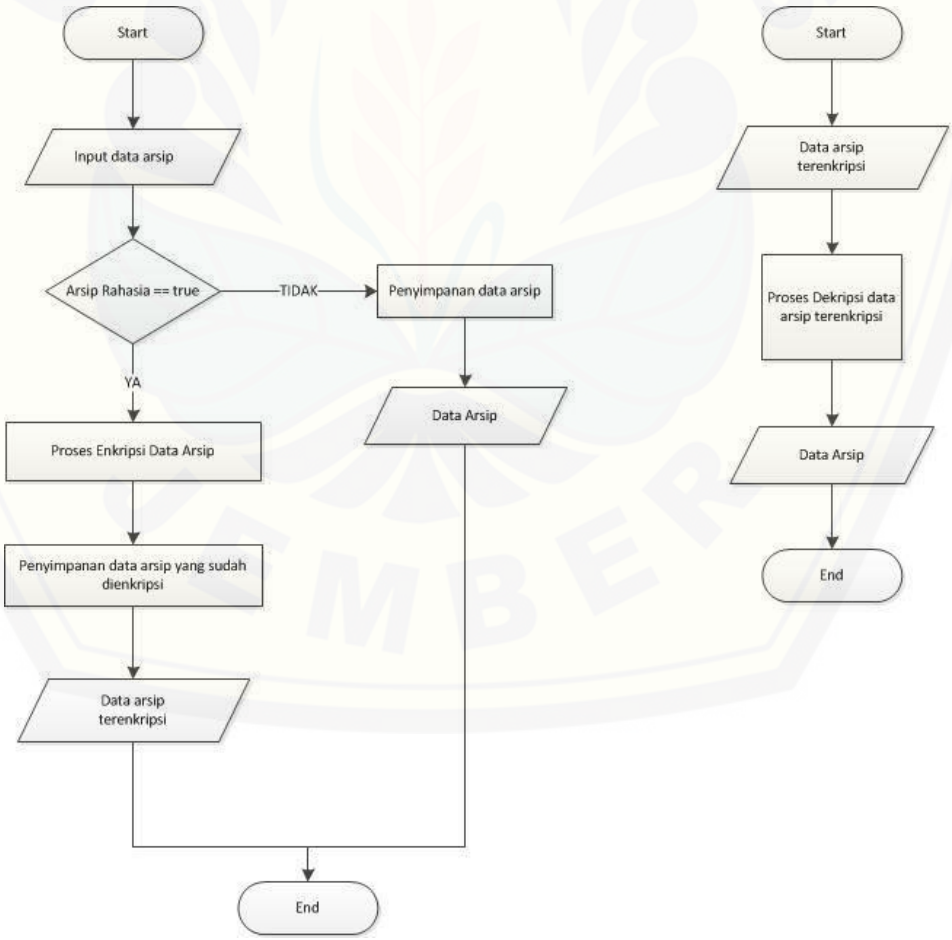
Cross Site Scripting (XSS) adalah sebuah serangan yang dilakukan dengan cara menginputkan HTML, JavaScript, ActiveX, Flash dan bahasa-bahasa pemrograman lain pada sisi *client* melalui *form* yang tidak memiliki validasi pada inputan (Pandian, 2015).

3.3.1 *Decryptor Online*

Decryptor Online adalah sebuah *tools* untuk mengubah *ciphertext* (pesan yang terenkripsi) menjadi *plaintext* (pesan aslinya) yang disebut dekripsi.

3.4. Gambaran Sistem

Sistem pengarsipan adalah sebuah sistem berbasis web yang digunakan untuk mengelola arsip suatu perusahaan. Arsip perusahaan ada yang bersifat rahasia dan ada yang bersifat umum. Untuk arsip yang bersifat rahasia, sistem akan mengenkripsi data tersebut terlebih dahulu sebelum diinputkan kedalam *database*. Kemudian akan dilakukan proses dekripsi untuk menampilkan arsip rahasia tersebut. Sedangkan arsip yang bersifat umum akan langsung diinputkan kedalam *database* tanpa dienkripsi terlebih dahulu. Dari uraian diatas, gambaran sistem yang akan dibuat dapat dilihat pada gambar 3.2.



Gambar 3.2 Gambaran Umum Sistem

BAB 5. HASIL DAN PEMBAHASAN

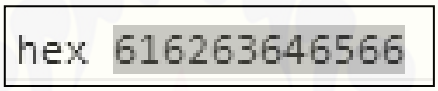
Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan sertapembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan danmemaparkan bagaimana penelitian ini menjawab perumusan masalah serta tujuan danmanfaat dari penelitian ini seperti yang telah ditentukan pada awal penelitian.

5.1 Implementasi Algoritma *Twofish*

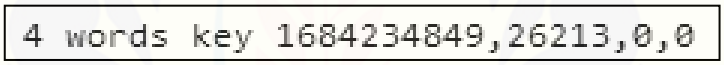
Sistem yang dibangun dalam penelitian ini digunakan untuk mengelola dan mengamankan data arsip. Sistem ini berbasis *web* yang dapat mengamankan data arsip rahasia sehingga pihak lain tidak dapat melihatnya. Sistem dibangun menggunakan *framework codeigniter* dan *database* MySQL sebeagai tempat penyimpanan datanya. Sistem ini memiliki beberapa fitur diantaranya input arsip, *view* daftar arsip, *view* arsip rahasia, pengecekan data arsip, login dan logout. Fitur input arsip merupakan fitur dimana pengguna dapat memilih apakah arsip tersebut bersifat rahasia atau tidak. Data arsip akan langsung dikirimkan ke server untuk disimpan di *database* jika pengguna menginputkan data arsip yang tidak bersifat rahasia, apabila pengguna memilih arsip bersifat rahasia maka data arsip akan dienkripsi menggunakan algoritma *twofish*.

Proses enkripsi pada sistem ini diawali dengan mengkonversi inputan *key* kedalam bentuk *hexadecimal*, dimana konversi *key* dillakukan untuk menyamarkan *key*. Contoh *key* yang diinputkan untuk enkripsi adalah ‘abcdef’ dan *plaintext* menggunakan kata ‘aku’. Pembagian konversi tersebut dapat dilihat pada Gambar 5.1, dimana *key* dan *plaintext* dibagi menjadi 4 bagian yang dinamakan *words* menggunakan konversi *little-endian*. Hasil konversi yang di dapatkan dapat dilihat pada Gambar 5.2 dan Gambar 5.3, dimana *words* dari *plaintext* dan *key* diubah kedalam bentuk *binary.* Langkah selanjutnya *words* keempat digeser ke kiri sejauh 8-*bit* sebelum di konversi kembali ke dalam bentuk persamaan XOR sebanyak 16-*round.* Hasil dari langkah tersebut didapatkan *ciphertext* per-blok seperti pada Gambar 5.4 yang kemudian diubah kedalam

bentuk *hexadecimal* seperti pada gambar 5.5. Langkah terakhir yaitu mengkonversi *ciphertext* menggunakan base64 encode yang dapat dilihat pada pada gambar 5.6, dimana bilangan binary dari *ciphertext* digabungkan. Hasil dari penggabungan binary tersebut kemudian dibagi menjadi 6 bagian bit. Contohnya, binary "00110011" dibagi kedalam beberapa 6 bagian bit maka akan menjadi "001100", "11". Apabila terdapat bagian yang tidak berjumlah 6 bit, maka akan dilakukan padding dengan cara menambahkan binary "00000000" sampai tiap bagiannya berjumlah 6 bit, menghasilkan "001100", "110000", "000000", "000000". Kemudian dikonversi kedalam bentuk base64 menggunakan tabel base64 pada gambar 5.7 menghasilkan "Mw==".

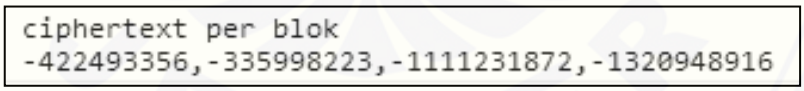


Gambar 5.1 Hasil Konversi *Key* Kedalam Bentuk *Hex*



Gambar 5.2 Hasil Konversi *Key* Menggunakan *Little-Endian*

Gambar 5.3 Hasil Konversi *Plaintext* Menggunakan *Little-Endian*



Gambar 5.4 Hasil *Ciphertext* Per-Blok

Gambar 5.5 Hasil Konversi *Ciphertext* Kedalam Bentuk *Hex*

Gambar 5.6 Hasil Konversi *Ciphertext* Menggunakan Base64 *Encoder*

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
pad	=						

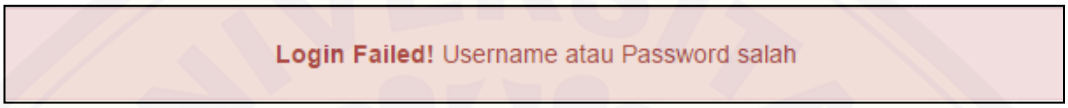
Gambar 5.7 Tabel Konversi Base64

5.2 Hasil Implementasi Aplikasi

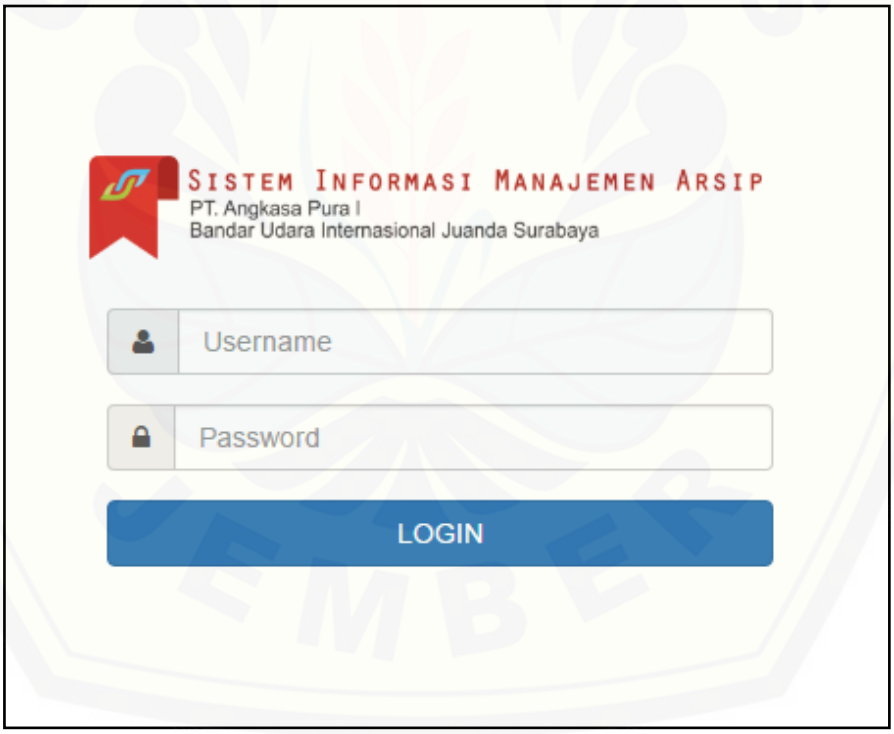
Hasil implementasi dari sistem informasi pengarsipan yang dibangun pada penelitian ini memiliki dua hak akses yaitu, admin dan sekretaris. Sistem informasi pengarsipan ini memiliki beberapa fitur :

5.2.1 Halaman *Login*

Halaman *Login* adalah halaman yang akan diakses oleh *user* saat pertama kali menjalankan sistem informasi pengarsipan. Sistem akan menampilkan sebuah form inputan untuk mengisi *username* dan *password* seperti pada gambar 5.9. Jika *user* berhasil melakukan *login*, sistem akan menampilkan halaman *dashboard* seperti pada gambar 5.10, apabila *user* gagal melakukan *login*, maka sistem akan menampilkan sebuah pesan seperti pada gambar 5.8.



Gambar 5.8 Pesan Gagal Login

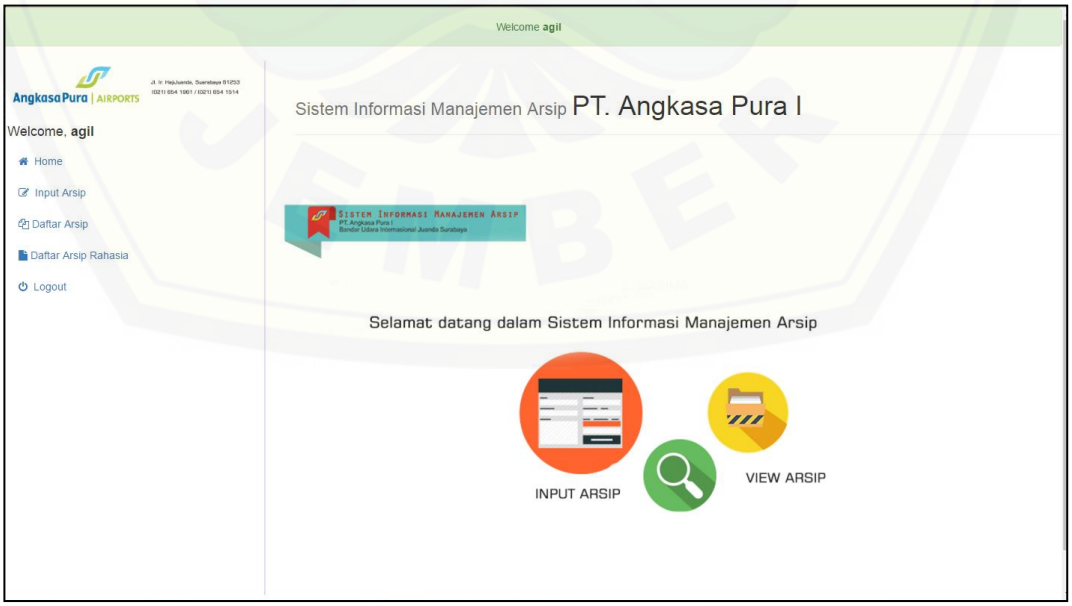


Gambar 5.9 Halaman Login

Gambar 5.10 Halaman Dashboard

5.2.2 Halaman Dashboard

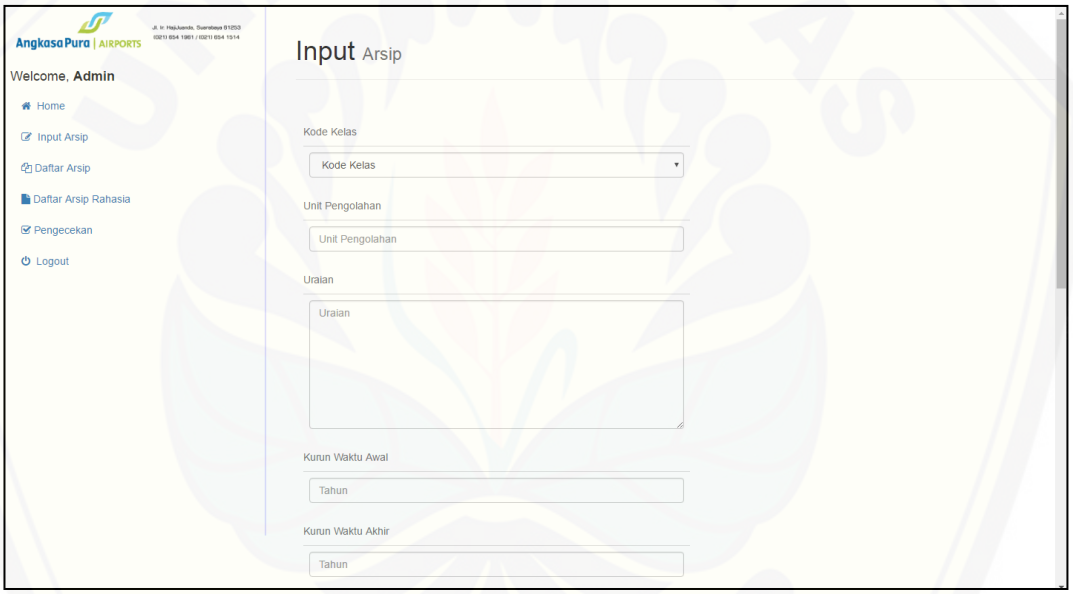
Halaman *Dashboard* dapat diakses setelah *user* melakukan *login*. Pada halaman ini dapat terlihat beberapa fitur dari sistem informasi pengarsipan, seperti input arsip, daftar arsip, daftar arsip rahasia, pengecekan, dan *logout*. Jika *userlogin* sebagai sekretaris, maka fitur pengecekan tidak akan ditampilkan seperti pada gambar 5.11 karena fitur tersebut hanya dimiliki oleh admin.



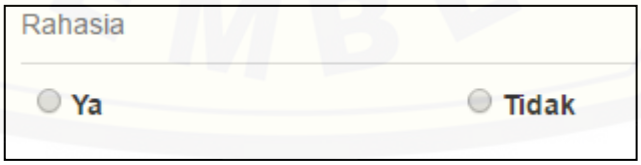
Gambar 5.11 Halaman *Dashboard* Sekretaris

5.2.3 Halaman Input Arsip

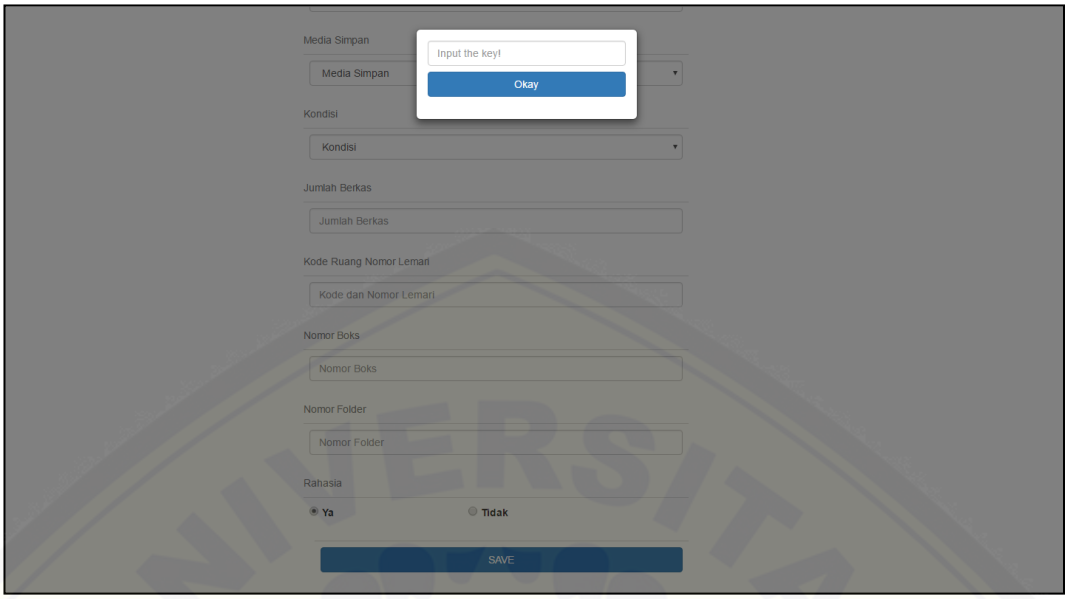
Halaman input arsip merupakan halaman untuk menginputkan data arsip yang bersifat rahasia maupun tidak. Halaman ini dapat diakses oleh admin maupun sekretaris. Tampilan dari halaman input arsip dapat dilihat pada gambar 5.12. Jika *user* ingin menginputkan data arsip yang bersifat rahasia, maka *user* harus memilih "ya" pada inputan gambar 5.13 setelah itu, sistem akan menampilkan sebuah inputan untuk mengisi *key* yang akan digunakan dalam proses enkripsi seperti pada gambar 5.14. Apabila *key* tidak diisi, sistem akan menampilkan pesan seperti gambar 5.15.



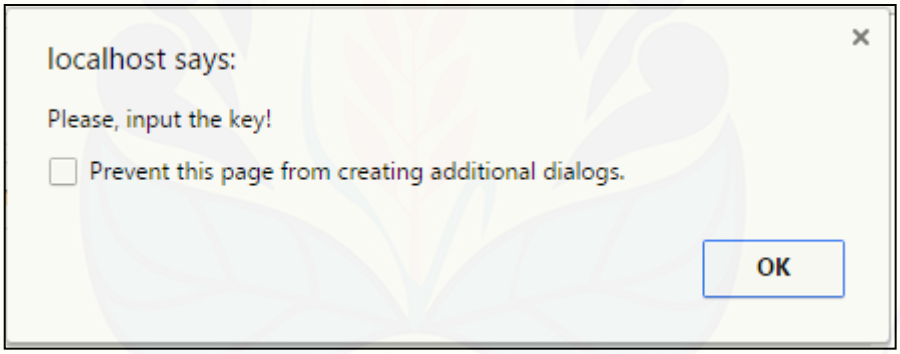
Gambar 5.12 Halaman Input Arsip



Gambar 5.13 Inputan Untuk Memilih Sifat Data Arsip



Gambar 5.14 Inputan Untuk Mengisi Key Yang Digunakan Dalam Proses Enkripsi

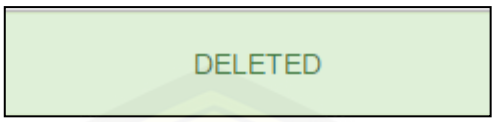


Gambar 5.15 Pesan Saat Key Tidak Diisi

5.2.4 Halaman Daftar Arsip

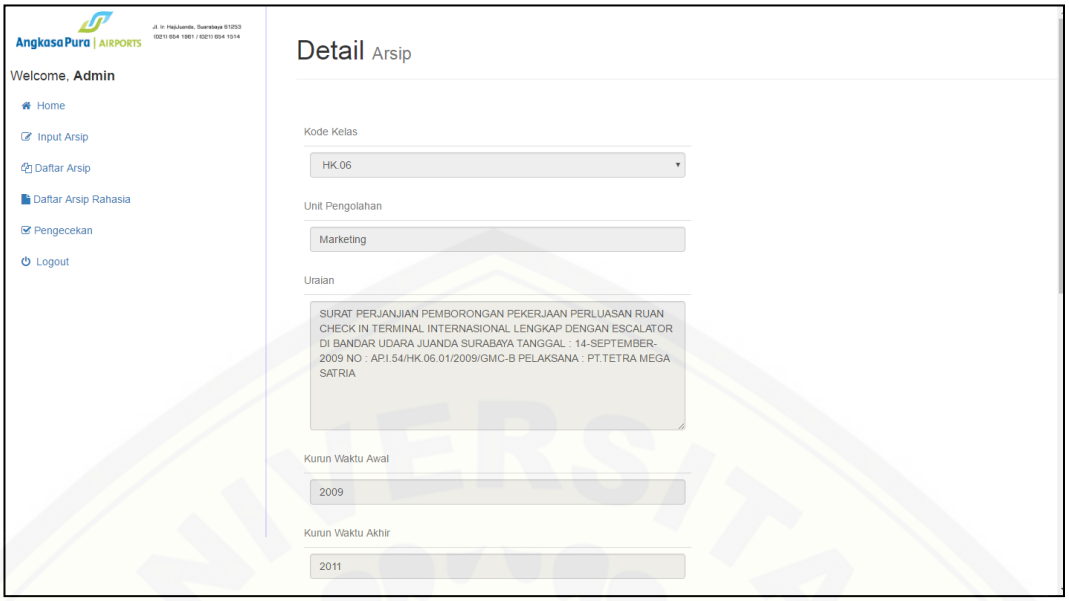
Halaman daftar arsip adalah halaman yang digunakan untuk melihat data arsip tidak rahasia yang telah diinputkan oleh *user*. Halaman akan menampilkan sebuah tabel data arsip dengan kolom no, kode kelas, uraian, nomor boks, nomor folder, dan aksi dimana pada kolom aksi terdapat tombol detail, *edit*, dan *delete* seperti pada gambar 5.17. Tombol detail akan menampilkan detail data arsip secara keseluruhan seperti pada gambar 5.18, tombol *edit* digunakan untuk mengubah data yang telah diinputkan seperti pada gambar 5.19, sedangkan

tombol *delete* untuk menghapus data arsip yang nantinya akan menampilkan pesan seperti pada gambar 5.16.

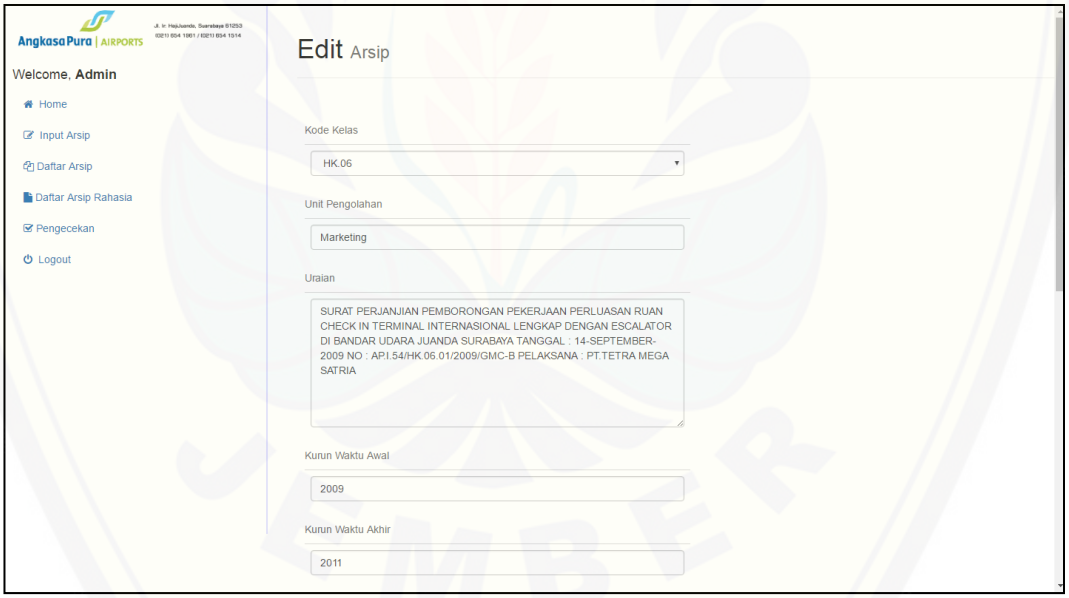


Gambar 5.16 Pesan *Delete* Arsip

Gambar 5.17 Halaman Daftar Arsip



Gambar 5.18 Halaman *Detail* Arsip



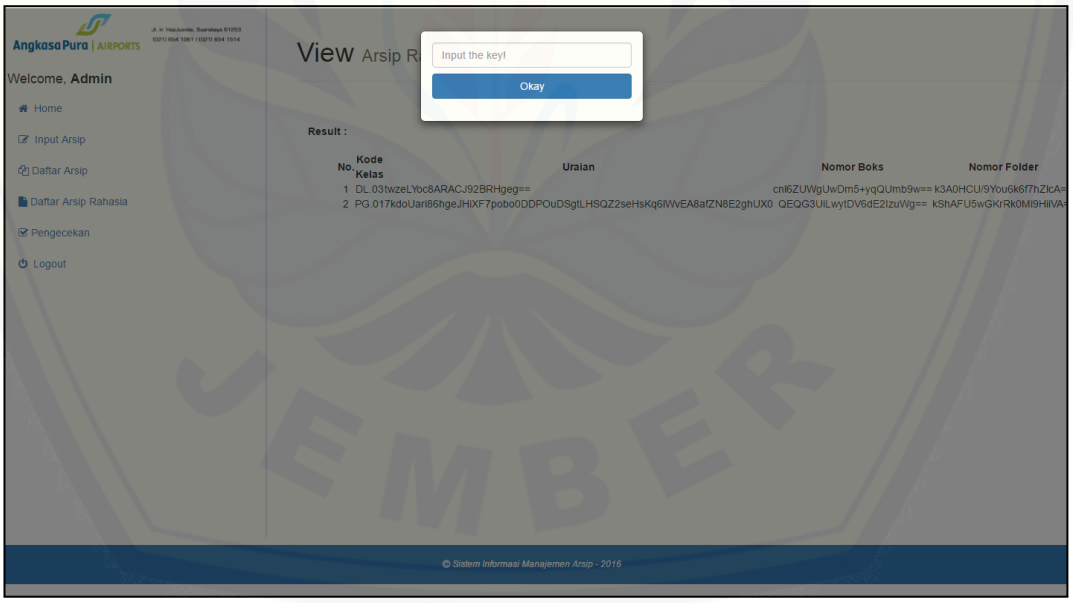
Gambar 5.19 Halaman *Edit* Arsip

5.2.5 Halaman Daftar Arsip Rahasia

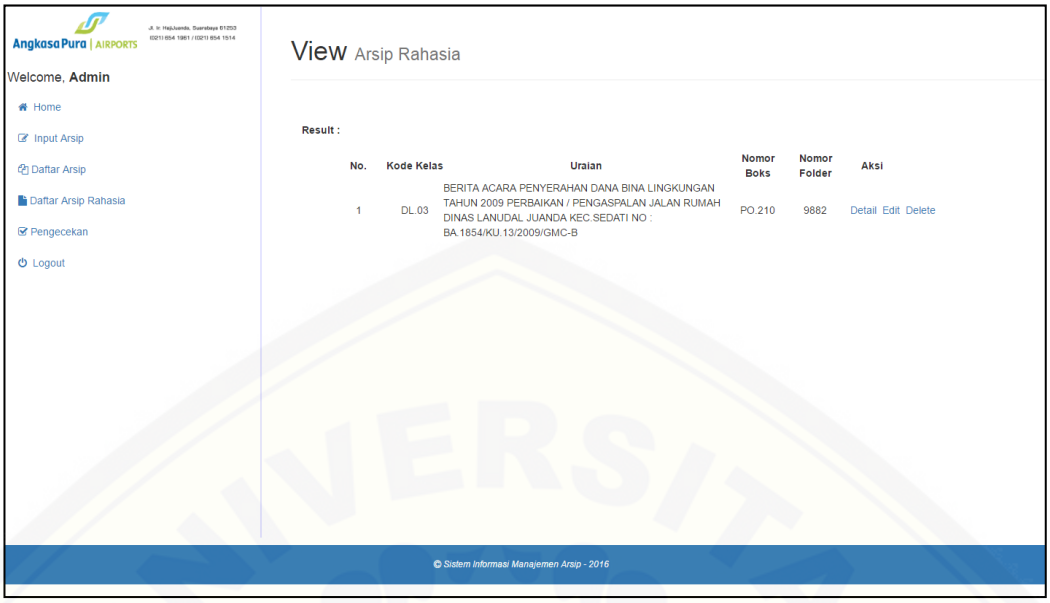
Halaman daftar arsip rahasia merupakan halaman yang menampilkan daftar arsip terenkripsi atau bersifat rahasia. Halaman ini akan menampilkan form

inputan untuk mengisi *key* yang digunakan untuk mengenkripsi saat menyimpan data seperti pada gambar 5.21, kemudian sistem akan menampilkan tabel daftar arsip rahasia seperti pada gambar 5.22 jika *key* yang diinputkan sesuai. Sistem akan menampilkan sebuah pesan seperti pada gambar 5.20 apabila *key* yang diinputkan tidak sesuai.

Gambar 5.20 Pesan Jika Key Tidak Sesuai



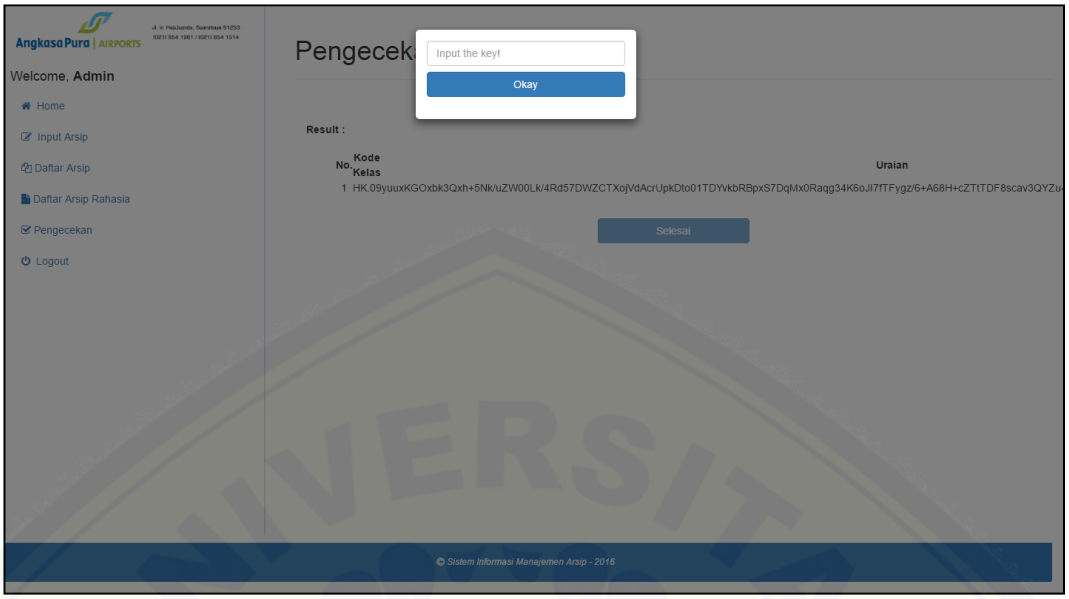
Gambar 5.21 Form Input *Key* Daftar Arsip Rahasia



Gambar 5.22 Halaman daftar arsip rahasia

5.2.6 Halaman Pengecekan

Halaman pengecekan adalah halaman yang hanya bisa diakses oleh admin. Halaman ini digunakan untuk mengecek ada atau tidaknya dokumen arsip yang telah diinputkan baik yang rahasia maupun tidak. Sistem akan menampilkan *form* untuk menginputkan *key* untuk mendekripsi data arsip rahasia seperti pada gambar 5.23. Jika *key* yang dimasukkan tidak sesuai, makadata dari arsip rahasia tidak dapat terbaca seperti pada gambar 5.24.



Gambar 5.23 *Form* input *key* pada halaman pengecekan

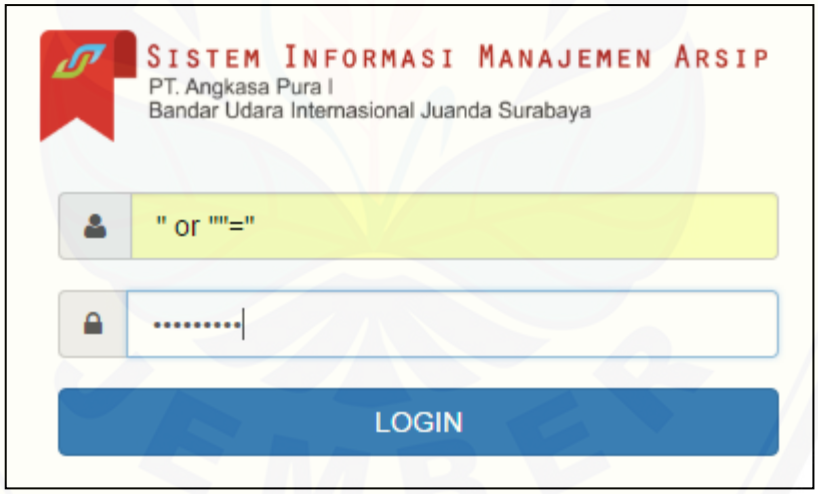
Gambar 5.24 Tampilan jika *key* yang diinputkan tidak sesuai

5.3 Pengujian Keamanan Sistem

Pengujian keamanan sistem dilakukan untuk melihat tingkat keamanan dari sistem yang dibangun. Pengujian akan dilakukan menggunakan tiga metode yaitu, SQL *injection, Cross-site-scripting,* dan *decryptor tools.*

5.3.1 *SQL Injection Attack*

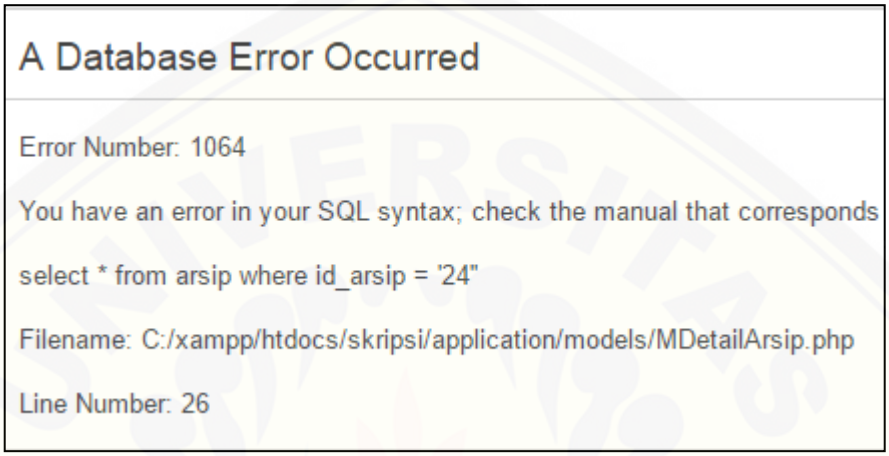
SQL *Injection Attack* adalah sebuah teknik memasukkan perintah SQL yang menyebabkan penyerang dapat mengakses *database* sistem dan memanipulasinya (Kannan, 2011). Hal tersebut tentunya sangat berbahaya bagi sebuah sistem. Oleh karena itu, penulis menggunakan teknik SQL *Injection Attack* untuk menguji sistem yang dibuat. Penyerang mencoba *login* menggunakan teknik *Bypass SQL Injection* seperti pada gambar 5.25 dan hasilnya penyerang tidak dapat *login* seperti pada gambar 5.26. Penyerang akan mencoba cara lain untuk *login* lalu mencari *method get* karena *method get* dapat dimanfaatkan untuk melakukan *SQL Injection Attack*. Penyerang akan mengubah *url* dengan menambahkan tanda petik pada akhir *url* seperti pada gambar 5.27 untuk melihat apakah akan muncul error pada sistem karena error pada *url* yang di dalamnya terdapat *method get* akan menampilkan struktur dari *database* sistem seperti pada gambar 5.28. Hasilnya sistem tetap menampilkan halaman tersebut tanpa error seperti pada gambar 5.29.



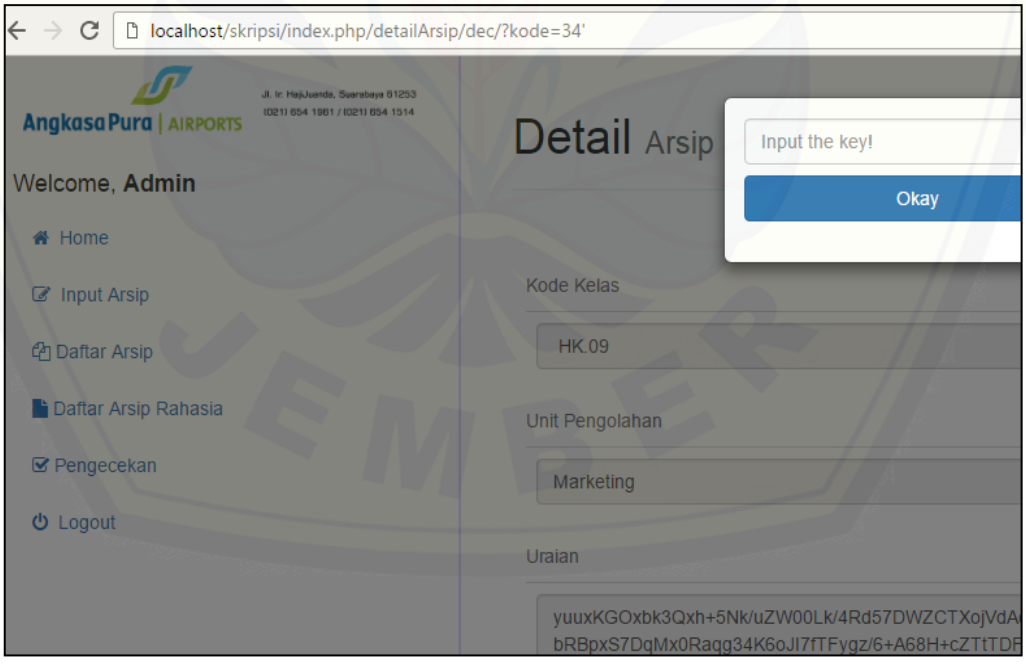
Gambar 5.25 Percobaan *Login* Menggunakan Bypass SQL

Gambar 5.26 Hasil Percobaan *Login* Menggunakan *Bypass SQL*

Gambar 5.27 Percobaan *SQL Injection Attack* Melalui *URL*



Gambar 5.28 Contoh error ketika di akhir *URL* diberi tanda petik



Gambar 5.29 Halaman *Detail* Arsip Rahasia Setelah *URL* Dari Halaman Tersebut Diberi Tanda Petik

Hasil percobaan *login* ke dalam sistem menggunakan beberapa teknik *Bypass SQL* dapat dilihat pada tabel 5.1.

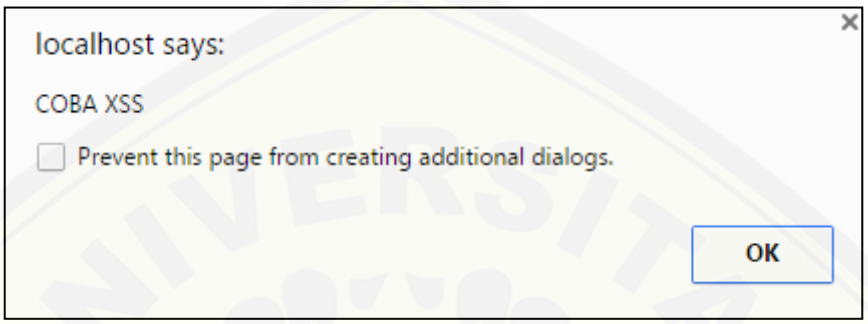
Tabel 5.1 Teknik Bypass SQL

No	Teknik *Bypass SQL*	Hasil
1	“ or “”=”	Gagal
2	Admin’ --	Gagal
3	Admin’ or ‘1’=’1	Gagal
4	Admin’) or ‘1’=’1	Gagal
5	Admin’) or (‘1’=’1’ --	Gagal
6	Or 1=1	Gagal
7	Or 1=1/*	Gagal

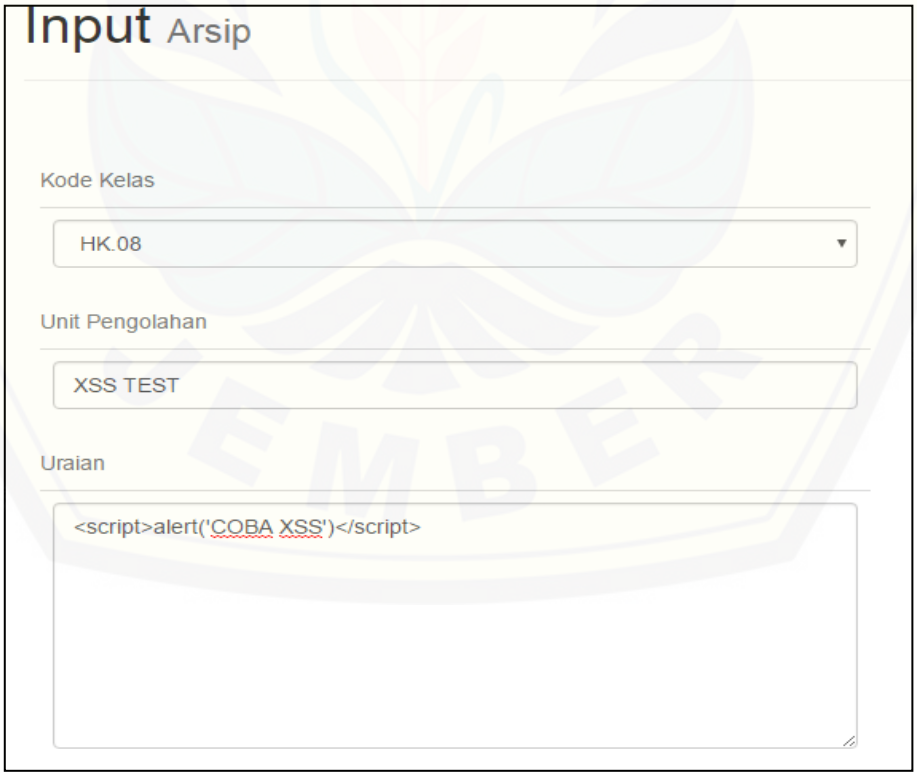
5.3.2 *Cross Site Scripting*

Cross Site Scripting (XSS) adalah sebuah serangan yang dilakukan dengan cara menginputkan HTML, JavaScript, ActiveX, Flash dan bahasa-bahasa pemrograman lain pada sisi *client* melalui *form* yang tidak memiliki validasi pada inputan (Pandian, 2015). *Cross site scripting* (XSS) merupakan salah satu dari beberapa metode penyerangan yang sering digunakan untuk mengambil *cookies* dari komputer korbannya. *Cookies* adalah data tentang aktifitas *user* yang dibuat oleh sebuah *website* dan tersimpan di dalam *browser user* tersebut. Hal ini akan memudahkan penyerang untuk mengambil data dari *user* seperti data untuk *login* dan sebagainya sehingga penulis akan menguji sistem yang dibuat menggunakan metode *cross site scripting* ini. Percobaan serangan menggunakan metode ini dapat dilihat pada gambar 5.32. Penyerang menginputkan *script alert* pada *form* input arsip. Sistem akan menampilkan *alert* seperti pada gambar 5.30 jika sistem tersebut dapat ditembus dengan *cross site scripting*. Penulis menggunakan *filter*

pada *form* input arsip yang berfungsi untuk menghapus *tag-tag* HTML yang diinputkan oleh *user* sebagai salah satu langkah pencegahan terhadap serangan XSS. Hasil dari inputan pada gambar 5.31 akan seperti pada gambar 5.32 setelah melalui proses *filter*.



Gambar 5.30 Contoh Hasil Percobaan Pada Sistem yang Dapat Ditembus Oleh XSS Attack

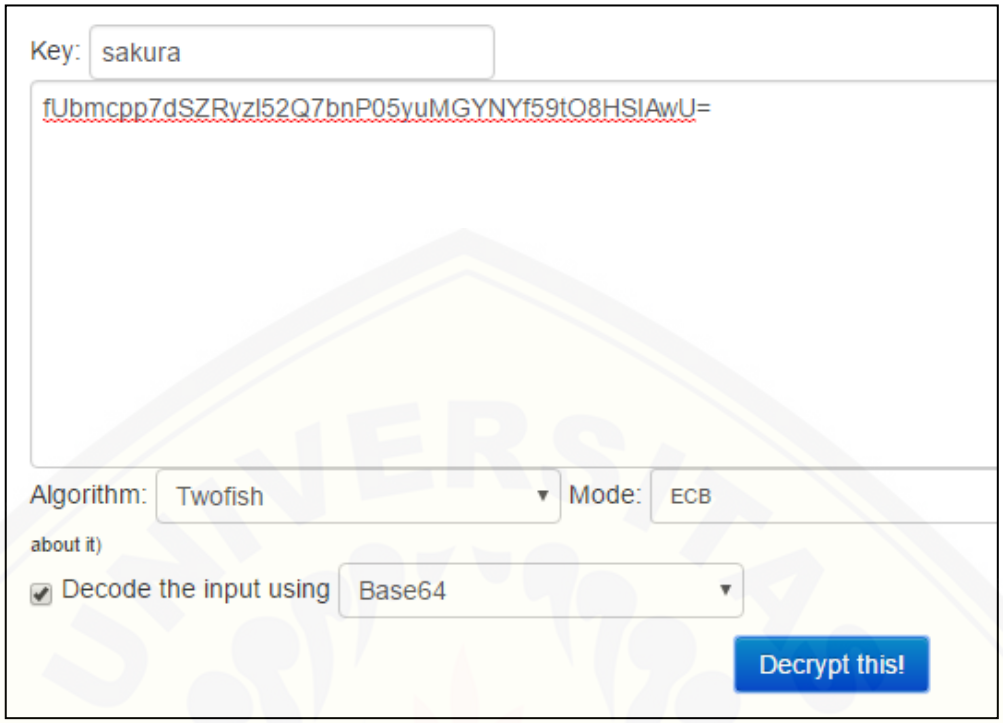


Gambar 5.31 Percobaan memasukkan *script* pada *form* input

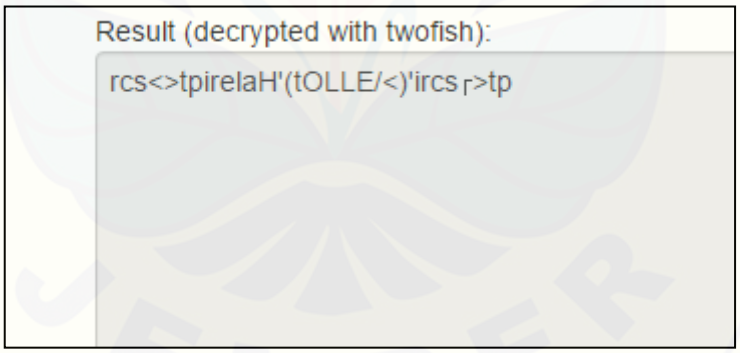
Gambar 5.32 Hasil Dari Proses *Filter* Pada *Form* Input Arsip

5.3.3 *Decryptor Online*

Decryptor online adalah sebuah *tools* untuk mengubah *ciphertext* menjadi *plaintext* atau biasa disebut dengan dekripsi. Sistem informasi pengarsipan menggunakan algoritma *twofish* untuk mengenkripsi data arsip yang bersifat rahasia. Penulis akan mencoba mendekripsi data arsip rahasia pada sistem informasi pengarsipan dengan algoritma yang sama menggunakan *decryptor online* seperti pada gambar 5.33 untuk menguji tingkat keamanannya. Hasilnya, data arsip rahasia tidak berubah menjadi *plaintext*-nya seperti pada gambar 5.34.



Gambar 5.33 Percobaan Mendekripsi Daftar Arsip Rahasia Menggunakan *Decryptor Online*



Gambar 5.34 Hasil Dari Percobaan Dekripsi Menggunakan *Decryptor Online*

5.4 Pembahasan Sistem

Semua sistem yang dibuat pasti memiliki kelebihan dan kekurangan masing-masing. Kelemahan dan kelebihan dari sistem informasi pengarsipan dari hasil pengujian diatas adalah sebagai berikut:

5.4.1 Kelebihan Sistem

Kelebihan dari sistem informasi pengarsipan yang dibangun antara lain :

1. Adanya proses enkripsi menggunakan algoritma *twofish* pada saat proses penyimpanan data yang menjamin keamanan data yang disimpan.
2. Sistem aman dari serangan *SQL injection, cross site scripting* dan *decryptor online* dari hasil pengujian diatas.

5.4.2 Kekurangan Sistem

Kekurangan dari sistem informasi yang dibangun adalah setiap mengakses fitur data arsip rahasia, akan selalu muncul input key untuk dekripsi data, sehingga membuat pengguna harus memverifikasi setiap akses, baik dalam lihat daftar arsip, detail dan edit. Data tidak dapat dilihat apabila *user* menginputkan *key* yang berbeda saat proses enkripsi. Proses input key yang berulang disetiap akses fitur daftar arsip rahasia tersebut, membuat akses user kurang efisien.

BAB 6. PENUTUP

Bab ini berisi kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan oleh peneliti, dapat diambil kesimpulan sebagai berikut :

1. Algoritma *twofish* diimplementasikan pada sisi *client* menggunakan *javascript* saat penginputan data dimana *key* akan diubah terlebih dahulu kedalam bentuk *hexadecimal* menggunakan teknik *hexadecimal convention* sebelum digunakan untuk mengenkripsi. *Ciphertext* hasil dari enkripsi tersebut kemudian disimpan didalam *database*.
2. Data yang tersimpan didalam *database* aman karena data tersebut berbentuk *ciphertext* dan *key* yang digunakan untuk mengenkripsi diubah terlebih dahulu kedalam bentuk *hexadecimal* sehingga data tersebut tidak dapat didekripsi menggunakan *decryptor online tools* walaupun *key* yang digunakan sama. Sistem ini dibangun tahan terhadap serangan SQL injection dan cross site scripting. Sistem ini aman dari serangan SQL injection karena percobaan serangan dalam method get tidak menampilkan pesan error yang berisi struktur dari *database* system. Sistem aman terhadap Serangan C*ross Site Scripting* karena terdapat filter berupa function stripHTML sehingga apabila terdapat inputan yang berupa script html maka script tersebut tidak akan dijalankan.

6.2 Saran

Pengembangan lebih lanjut untuk penelitian ini dapat dilakukan dengan membuat *random key* yang digunakan untuk enkripsi data kemudian mengirim *key* tersebut ke server dengan menambahkan algoritma enkripsi asimetrik seperti algortima RSA atau *Diffie Helman key exchange algorithm* agar proses enkripsi dan dekripsi data berjalan otomatis tanpa perlu menginputkan *key*.

DAFTAR PUSTAKA

Kannan, V. d. (2011). *SQL Injection - Database Attack Revolution And Prevention.*

Mukmin, I. (2007). *Algoritma Twofish : Kinerja dan Implementasinya Sebagai Salah Satu Kandidat Algoritma AES (Advanced Encryption Standard).*

Nidhra, S., & Dondeti, J. (2012). *Black Box and White Box Techniques - A literature Review.*

Pandian, L. (2015). *A Survey on Detection and Prevention of Cross-Site Scripting Attack.*

Primartha, R. (2011). *Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma Data Encryption.*

Randy, A. (2010). *Studi dan Perbandingan Algoritma Blowfish dan Twofish.*

Schneier, B., & Whiting, D. (2000). *A Performance Comparison of the Five AES Finalists.*

Setiawan, W. (2011). *Analisa dan Perbandingan Algoritma Twofish dan Rijndael.*

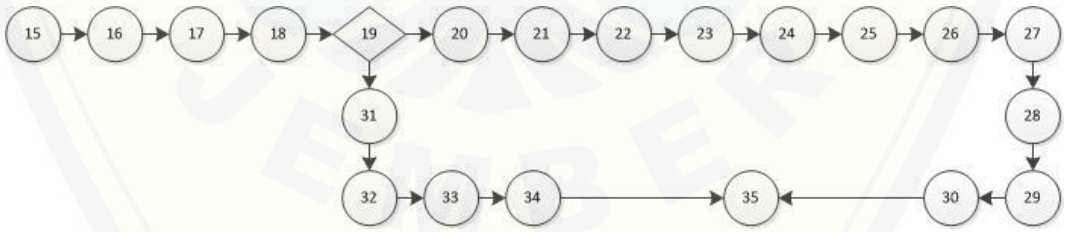
LAMPIRAN

LAMPIRAN A (*WHITE BOX TESTING*)

A.1 *White box login*

Tabel A. 1 *Listing Program Login*

15	public function authentication(){
16	$username = $this->input->post('username');
17	$password = $this->input->post('password');
18	$result = $this->Mlogin->checkLogin($username,$password);
19	if($result){
20	$data = $this->Mlogin->getDataUser($username)->result();
21	$parSession = array(
22	'username' => $username,
23	'tipe' => $data[0]->User_Type,
24	'nama' => $data[0]->NamaUser,
25	'kode' => $data[0]->Unit,
26	'auth' => TRUE
27	);
28	$this->session->set_userdata($parSession);
29	$this->session->set_flashdata('message','Welcome '.$this->session->userdata('nama').'');
30	redirect('dashboard');
31	}else{
32	$this->session->set_flashdata('message','Username atau Password salah');
33	redirect(base_url().'index.php/'.$this->input->get('page'));
34	}
35	}



Gambar A. 1 Diagram Alir *Login*

CC = (Edge – Node) + 2

= (21-21) + 2 = 2

Tabel A. 2 Basis Test Login

authentication()	
Jalur 1 : 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 35	
Kasus	*Username* dan *password* telah terisi dan sesuai
Target yang diharapkan	Menampilkan menampilkan pesan “Welcome” dan halaman dashboard
Hasil Pengujian	Berhasil
Jalur 2 : 15 – 16 – 17 – 18 – 19 – 31 – 32 – 33 – 34 – 35	
Kasus	*Username* dan *password* telah terisi dan tidak sesuai
Target yang diharapkan	Menampilkan pesan “Warning! Username atau Password salah” dan halaman Login
Hasil Pengujian	Berhasil

A.2 *White Box* Input arsip

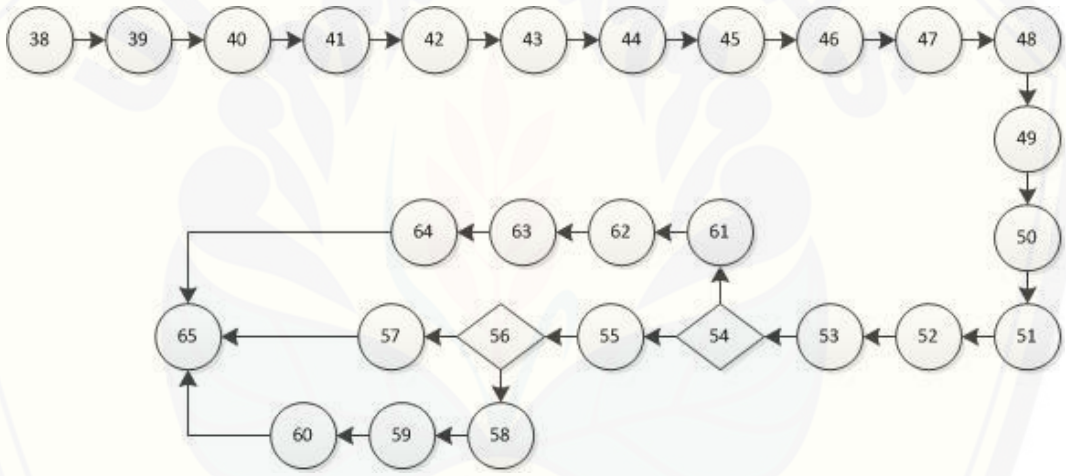
Tabel A. 3 Listing Program Input Arsip

38	public function submit(){
39	$result = $this->MEditArsip->edit(
40	htmlspecialchars($this->input->post('id_arsip')),
41	htmlspecialchars($this->input->post('kodekelas')),
42	htmlspecialchars($this->input->post('unitpengolahan')),
43	htmlspecialchars($this->input->post('uraian')),
44	htmlspecialchars($this->input->post('kurunwaktuawal')),
45	htmlspecialchars($this->input->post('kurunwaktuakhir')),
46	htmlspecialchars($this->input->post('jenisdokumen')),
47	htmlspecialchars($this->input->post('mediasimpan')),
48	htmlspecialchars($this->input->post('kondisi')),
49	htmlspecialchars($this->input->post('jumlahberkas')),
50	htmlspecialchars($this->input->post('nomorlemari')),

Dilanjutkan

Lanjutan

51	htmlspecialchars($this->input->post('nomorboks')),
52	htmlspecialchars($this->input->post('nomorfolder'))
53	);
54	if($result){
55	$this->session->set_flashdata('message','SUKSES ENTRY DATA');
56	if($this->input->post('rahasia')==1){
57	redirect('dekriparsip');
58	}else{
59	redirect('arsip');
60	}
61	}else{
62	$this->session->set_flashdata('message','GAGAL ENTRY DATA');
63	redirect('base_url()'.$this->input->get('page'));
64	}
65	}



Gambar A. 2 Diagram Alir Input Arsip

CC = (Edge – Node) + 2

= (29 – 28) + 2 = 3

Tabel A. 4 Basis Test Input Arsip

submit()
Jalur 1 : 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 55 – 56 – 57 – 65

Dilanjutkan

Lanjutan

Kasus	Data terisi semua dengan *check point* bukan rahasia dan klik Save
Target yang diharapkan	Menampilkan pesan "Sukses Entry Data"
Hasil Pengujian	Berhasil
Jalur 2 : 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 58 – 59 – 60 – 65	
Kasus	Data terisi semua dengan *check point* Rahasia dan klik Save
Target yang diharapkan	Menampilkan pesan "Sukses Entry Data"dan menampilkan Popup "Input the key"
Hasil Pengujian	Berhasil
Jalur 3 : 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 61 – 62 – 63 – 64 – 65	
Kasus	Data belum terisi semua dengan *check point* Rahasia dan klik Save
Target yang diharapkan	Menampilkan pesan "Gagal Entry Data"
Hasil Pengujian	Berhasil
submit()	
Jalur 1 : 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 44 – 45 – 50	
Kasus	Data terisi semua dengan *check point* Tidak dan klik Save
Target yang diharapkan	Menampilkan pesan "Sukses Entry Data"
Hasil Pengujian	Berhasil

Dilanjutkan

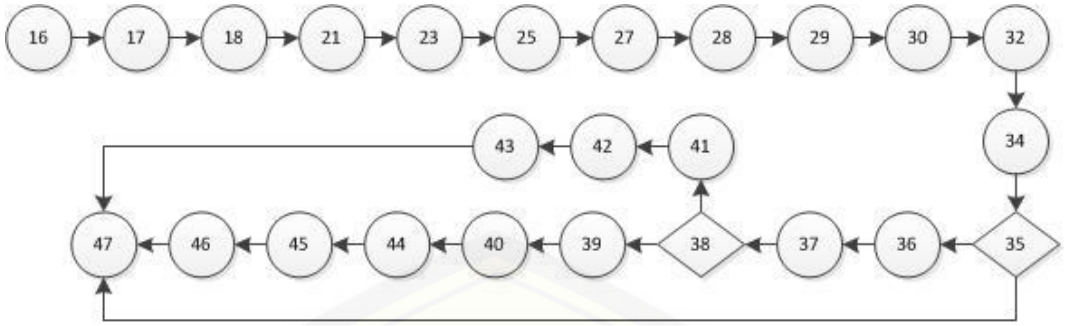
Lanjutan

Jalur 2 : 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41 – 42 – 43 – 46 – 47 – 48 – 49 – 50	
Kasus	Data belum terisi semua dengan *check point* Tidak dan klik Save
Target yang diharapkan	Menampilkan pesan “Gagal Entry Data”
Hasil Pengujian	Berhasil

A.3 *White Box View* Daftar Arsip

Tabel A. 5*ListingProgramView*Daftar Arsip

```
16  public function index(){
17  $data['controller'] = $this;
18  data["cek"]=TRUE;
19  // pagination
20  // set limit data per page
21  $limit = 10;
22  // set total data
23  $total = $this->MArsip->get_jumlah_arsip()->result();
24  // load library pagination
25  $this->load->library('pagination');
26  // config pagination
27  $config['base_url'] = base_url().'index.php/arsip?';
28  $config['total_rows'] = $total[0]->jumlah;
29  $config['per_page'] = $limit;
30  $config['page_query_string'] = TRUE;
31  // $config['query_string_segment'] = 'page';
32  $this->pagination->initialize($config);
33  // set limit max page yang akan diload
34  $page = $this->input->get('per_page');
35  if($page==""){
36  $page="0";
37  }
38  if($this->input->get_post('tahun')!=""){
39  $data["cek"]=FALSE;
40  $data["arsip"]=$this->MArsip->get_arsip($this->input->post('tahun'));
41  }else{
42  $data["arsip"]=$this->MArsip->get_all_arsip($page,$limit);
43  }
44  $this->load->view('admin/header');
45  $this->load->view('admin/vArsip',$data);
46  $this->load->view('footer');
47  }
```



Gambar A. 3 Diagram Alir *View*Daftar Arsip

CC = (Edge – Node) + 2

= (26 – 25) + 2 = 3

Tabel A. 6 Basis Test*View* Daftar Arsip

index()	
Jalur 1 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 47	
Kasus	Memilih menu daftar arsip dan data yang tampil maksimal 10 data arsip tiap halaman
Target yang diharapkan	Menampilkan halaman daftar arsip dengan maksimal 10 data arsip tiap halaman
Hasil Pengujian	Berhasil
Jalur 2 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 44 – 45 – 46 – 47	
Kasus	Memilih tahun dari daftar arsip untuk ditampilkan dan data yang tampil maksimal 10 data arsip tiap halaman

Dilanjutkan

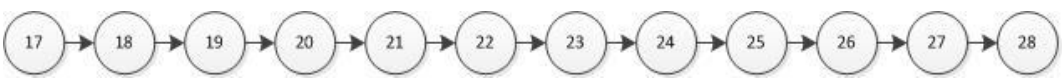
Lanjutan

Target yang diharapkan	Menampilkan daftar arsip maksimal 10 data arsip tiap halaman berdasarkan tahun yang dipilih
Hasil Pengujian	Gagal
Jalur 3 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 36 – 37 – 38 – 41 – 42 – 43 – 47	
Kasus	Tidak memilih tahun dan data yang tampil maksimal 10 data arsip tiap halaman
Target yang diharapkan	Menampilkan halaman daftar arsip dengan maksimal 10 data arsip tiap halaman
Hasil Pengujian	Berhasil

A.4 *White Box View* Detail Arsip

Tabel A. 7*ListingProgramView* Detail Arsip

```
17  public function index(){
18  $data["controller"] = $this;
19  $kode = $this->input->get('kode');
20  $data["data"]=$this->MDetailArsip->getdata($kode);
21  $data["kondisi"]=$this->MDetailArsip->getkondisi();
22  $data["mediasimpan"]=$this->MDetailArsip->getmediasimpan();
23  $data["kodekelas"]=$this->MDetailArsip->getkodekelas();
24  $data["jenisdokumen"]=$this->MDetailArsip->getjenisdokumen();
25  $this->load->view('admin/header');
26  $this->load->view('admin/vDetailArsip',$data);
27  $this->load->view('footer');
28  }
```



Gambar A. 4 Diagram Alir *View* Detail Arsip

CC = (Edge – Node) + 2

= (11 – 12) + 2 = 1

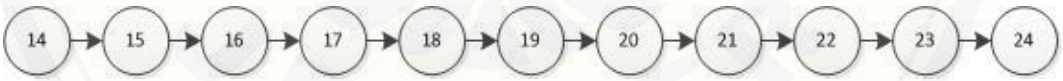
Tabel A. 8 Basis Test *View* Detail Arsip

index()	
Jalur 1 : 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28	
Kasus	Memilih tombol detail
Target yang diharapkan	Menampilkan *detail* dari data arsip
Hasil Pengujian	Berhasil

A.5 *White Box View Edit Arsip*

Tabel A. 9*ListingProgramView* Edit Arsip

14	public function index(){
15	$kode = $this->input->get('kode');
16	$data["data"]=$this->MEditArsip->getdata($kode);
17	$data["kondisi"]=$this->MEditArsip->getkondisi();
18	$data["mediasimpan"]=$this->MEditArsip->getmediasimpan();
19	$data["kodekelas"]=$this->MEditArsip->getkodekelas();
20	$data["jenisdokumen"]=$this->MEditArsip->getjenisdokumen();
21	$this->load->view('admin/header');
22	$this->load->view('admin/vEditArsip',$data);
23	$this->load->view('footer');
24	}



Gambar A. 5 Diagram Alir *View* Edit Arsip

CC = (Edge – Node) + 2

= (11 – 12) + 2 = 1

Tabel A. 10 Basis Test *View* Edit Arsip

index()	
Jalur 1 : 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24	
Kasus	Memilih tombol edit
Target yang diharapkan	Menampilkan *form* data arsip yang telah terisi
Hasil Pengujian	Berhasil

A.6 *White Box Delete* Arsip

Tabel A. 11 Lisitng Program Delete Arsip

62	public function delete(){
63	$result=$this->MArsip-> deletetransaksi($this->input->get('kode'));
64	if($result){
65	$this->session->set_flashdata('message','DELETED');
66	redirect ('Arsip');
67	else{
68	$this->session->set_flashdata('message','Failed to delete data');
69	redirect(base_url().'index.php/'.$this->input->get('page'));
70	}
71	}

Gambar A. 6 Diagram Alir *Delete* Arsip

CC = (Edge – Node) + 2

= (9 – 10) + 2 = 1

Tabel A. 12 Basis Test *Delete* Arsip

delete()	
Jalur 1 : 62 – 63 – 64 – 65 – 66 – 67 – 68 – 69 – 70– 71	
Kasus	Memilih tombol delete

Dilanjutkan

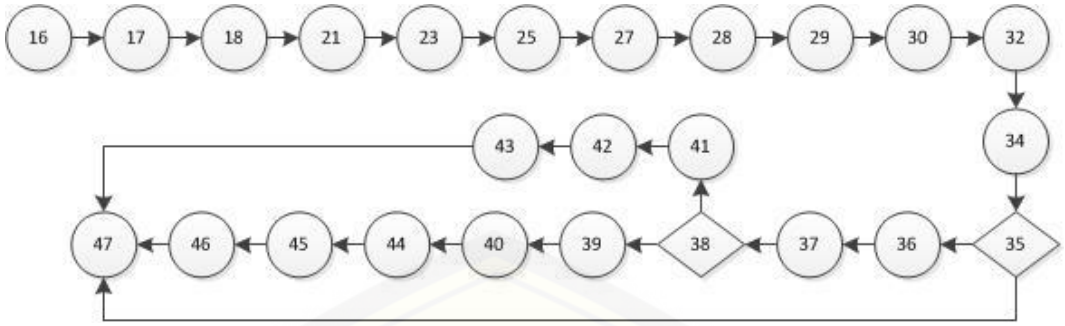
Lanjutan

Target yang diharapkan	Menampilkan halaman ViewArsip terbaru
Hasil Pengujian	Berhasil

A.7 *White Box View* Daftar Arsip Rahasia

Tabel A. 13 Listing Program *View* Daftar Arsip Rahasia

```
16  public function index(){
17  $data['controller'] = $this;
18  $data["cek"]=TRUE;
19  // pagination
20  // set limit data per page
21  $limit = 10;
22  // set total data
23  $total = $this->MArsip->get_jumlah_arsip()->result();
24  // load library pagination
25  $this->load->library('pagination');
26  // config pagination
27  $config['base_url'] = base_url().'index.php/arsip/dekripArsip?';
28  $config['total_rows'] = $total[0]->jumlah;
29  $config['per_page'] = $limit;
30  $config['page_query_string'] = TRUE;
31  // $config['query_string_segment'] = 'page';
32  $this->pagination->initialize($config);
33  // set limit max page yang akan diload
34  $page = $this->input->get('per_page');
35  if($page==""){
36  $page="0";
37  }
38  if($this->input->get_post('tahun')!=""){
39  $data["cek"]=FALSE;
40  $data["arsip"]=$this->MArsip->get_arsip($this->input->post('tahun'));
41  }else{
42  $data["arsip"]=$this->MArsip->get_enc_arsip($page,$limit);
43  }
44  $this->load->view('admin/header');
45  $this->load->view('admin/vDekripArsip',$data);
46  $this->load->view('footer');
47  }
```



Gambar A. 7 Diagram Alir *View* Daftar Arsip Rahasia

CC = (Edge – Node) + 2

= (26 – 25) + 2 = 3

Tabel A. 14 Basis Test *View* Daftar Arsip Rahasia

index()	
Jalur 1 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 47	
Kasus	Memilih menu daftar arsip rahasia dan data yang tampil maksimal 10 data arsip tiap halaman
Target yang diharapkan	Menampilakan popup “input the key” dan halaman daftar arsip rahasia dengan maksimal 10 data arsip tiap halaman
Hasil Pengujian	Berhasil
Jalur 2 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 44 – 45 – 46 – 47	
Kasus	Memilih tahun dari daftar arsip rahasia untuk ditampilkan dan data yang tampil maksimal 10 data arsip tiap halaman

Dilanjutkan

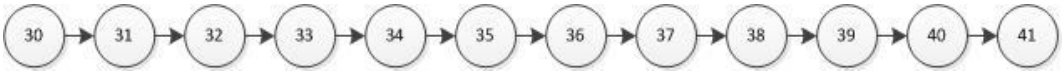
Lanjutan

Target yang diharapkan	Menampilkan popup “input the key” dan daftar arsip rahasia dengan maksimal 10 data tiap halaman berdasarkan tahun yang dipilih
Hasil Pengujian	Gagal
Jalur 3 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 36 – 37 – 38 – 41 – 42 – 43 – 47	
Kasus	Tidak memilih tahun dan data yang tampil maksimal 10 data arsip tiap halaman
Target yang diharapkan	Menampilakan popup “input the key” dan halaman daftar arsip rahasia dengan maksimal 10 data arsip tiap halaman
Hasil Pengujian	Berhasil

A.8 *White BoxView* Detail Arsip Rahasia

Tabel A. 15*ListingProgramView* Detail Arsip Rahasia

```
30  public function dec(){
31  $data["controller"] = $this;
32  $kode = $this->input->get('kode');
33  $data["data"]=$this->MDetailArsip->getdata($kode);
34  $data["kondisi"]=$this->MDetailArsip->getkondisi();
35  $data["mediasimpan"]=$this->MDetailArsip->getmediasimpan();
36  $data["kodekelas"]=$this->MDetailArsip->getkodekelas();
37  $data["jenisdokumen"]=$this->MDetailArsip->getjenisdokumen();
38  $this->load->view('admin/header');
39  $this->load->view('admin/vDecDetail',$data);
40  $this->load->view('footer');
41  }
```



Gambar A. 8 Diagram Alir View Daftar Arsip Rahasia

CC = (Edge – Node) + 2

= (11 – 12) + 2 = 1

Tabel A. 16 Basis Test *View* Daftar Arsip Rahasia

dec()	
Jalur 1 : 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 41	
Kasus	Memilih tombol detail
Target yang diharapkan	Menampilkan popup “input the key” dan *detail* dari daftar arsip rahasia
Hasil Pengujian	Berhasil

A.9 *White Box View Edit* ArsipRahasia

Tabel A. 17*ListingProgramViewEdit* Arsip Rahasia

26	public function dec(){
27	$kode = $this->input->get('kode');
28	$data["data"]=$this->MEditArsip->getdata($kode);
29	$data["kondisi"]=$this->MEditArsip->getkondisi();
30	$data["mediasimpan"]=$this->MEditArsip->getmediasimpan();
31	$data["kodekelas"]=$this->MEditArsip->getkodekelas();
32	$data["jenisdokumen"]=$this->MEditArsip->getjenisdokumen();
33	$this->load->view('admin/header');
34	$this->load->view('admin/vEdit_encArsip',$data);
35	$this->load->view('footer');
36	}

Gambar A. 9 Diagram Alir View Edit Arsip Rahasia

CC = (Edge – Node) + 2

= (11 – 12) + 2 = 1

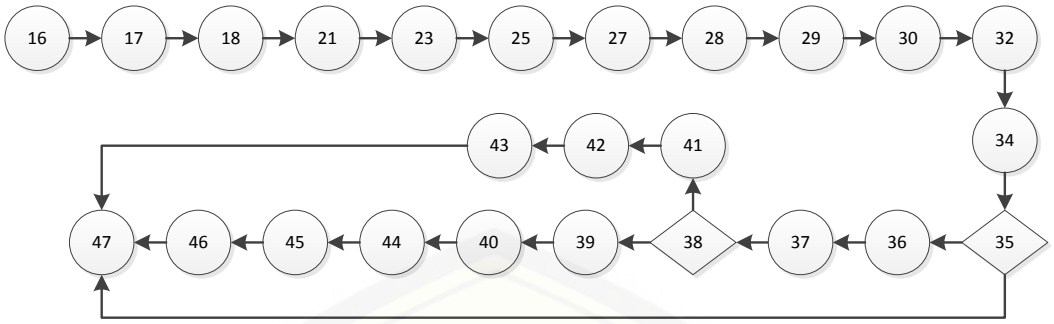
Tabel A. 18 Basis Test *ViewEdit* Arsip Rahasia

index()	
Jalur 1 : 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36	
Kasus	Memilih tombol edit
Target yang diharapkan	popup “input the key” dan *form* data arsip rahasia yang telah terisi
Hasil Pengujian	Berhasil

A.10 *WhiteBox View* Pengecekan Arsip

Tabel A. 19*ListingProgram View* Pengecekan Arsip

16	public function index(){
17	$data['controller'] = $this;
18	$data["cek"]=TRUE;
19	// pagination
20	// set limit data per page
21	$limit = 10;
22	// set total data
23	$total = $this->MArsip->get_jumlah_arsip()->result();
24	// load library pagination
25	$this->load->library('pagination');
26	// config pagination
27	$config['base_url'] = base_url().'index.php/arsip?';
28	$config['total_rows'] = $total[0]->jumlah;
29	$config['per_page'] = $limit;
30	$config['page_query_string'] = TRUE;
31	// $config['query_string_segment'] = 'page';
32	$this->pagination->initialize($config);
33	// set limit max page yang akan diload
34	$page = $this->input->get('per_page');
35	if($page==""){
36	$page="0";
37	}
38	if($this->input->get_post('tahun')!=""){
39	$data["cek"]=FALSE;
40	$data["arsip"]=$this->MArsip->get_arsip($this->input->post('tahun'));
41	}else{
42	$data["arsip"]=$this->MArsip->get_all_arsip($page,$limit);
43	}
44	$this->load->view('admin/header');
45	$this->load->view('admin/vPengecekan',$data);
46	$this->load->view('footer');
47	}



Gambar A. 10 Diagram Alir*View* Pengecekan Arsip

CC = (Edge – Node) + 2

= (26 – 25) + 2 = 3

Tabel A. 20 Basis Test *View* Pengecekan Arsip

index()	
Jalur 1 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 47	
Kasus	Memilih menu pengecekan dan data yang tampil maksimal 10 data arsip tiap halaman
Target yang diharapkan	Menampilakan popup “input the key” dan halaman pengecekan rahasia dengan maksimal 10 data arsip tiap halaman
Hasil Pengujian	Berhasil
Jalur 2 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 36 – 37 – 38 – 39 – 40 – 44 – 45 – 46 – 47	
Kasus	Memilih tahun dari pengecekan untuk ditampilkan dan data yang tampil maksimal 10 data arsip tiap halaman

Dilanjutkan

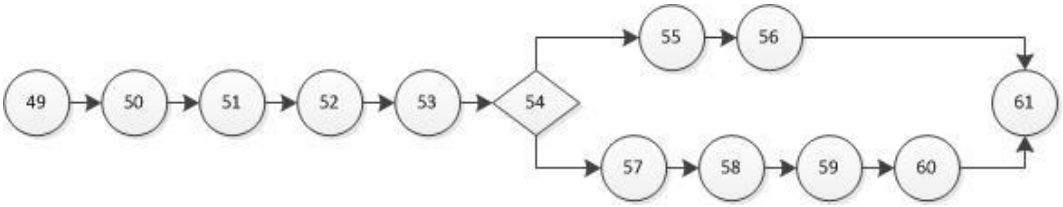
Lanjutan

Target yang diharapkan	Menampilkan popup “input the key” dan data arsip dengan maksimal 10 data tiap halaman berdasarkan tahun yang dipilih
Hasil Pengujian	Gagal
Jalur 3 : 16 – 17 – 18 – 21 – 23 – 25 – 27 – 28 – 29 – 30 – 32 – 34 – 35 – 36 – 37 – 38 – 41 – 42 – 43 – 47	
Kasus	Tidak memilih tahun dan data yang tampil maksimal 10 data arsip tiap halaman
Target yang diharapkan	Menampilakan popup “input the key” dan halaman pengecekan dengan maksimal 10 data arsip tiap halaman
Hasil Pengujian	Berhasil

A.11 *WhiteBox* Pengecekan Arsip

Tabel A. 21 Listing Program Pengecekan Arsip

```
49   public function cek(){
50   $check = $this->input->post('checklist');
51   foreach ($check as $cek) {
52   $result = $this->MPengecekan->cek($cek);
53   }
54   if($result){
55   $this->session->set_flashdata('message','SUKSES ENTRY DATA');
56   redirect('Pengecekan');
57   }else{
58   $this->session->set_flashdata('message','GAGAL ENTRY DATA');
59   redirect('base_url()'.$this->input->get('page'));
60   }
61   }
```



Gambar A. 11 Diagram Alir Pengecekan Arsip

CC = (Edge – Node) + 2

= (13 – 13) + 2 = 2

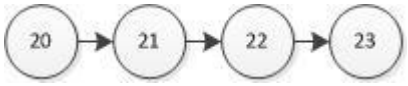
Tabel A. 22 Basis Test Pengecekan Arsip

cek()	
Jalur 1 : 49 – 50 – 51 – 52 – 53 – 54 – 55 – 56 – 61	
Kasus	*Check point* telah terisi dan klik selesai
Target yang diharapkan	Menampilkan pesan "SUKSES ENTRY DATA"
Hasil Pengujian	Berhasil
Jalur 2 : 49 – 50 – 51 – 52 – 53 – 54 – 57 – 58 – 59 – 60 – 61	
Kasus	*Check point* belum terisi dan klik selesai
Target yang diharapkan	Menampilkan pesan "GAGAL ENTRY DATA"
Hasil Pengujian	Berhasil

A.12 *White Box Logout*

Tabel A. 23*Listing Program Logout*

20	public function logout(){
21	$this->session->sess_destroy();
22	redirect('Login');
23	}



Gambar A. 12 Diagram Alir *Logout*

CC = (Edge – Node) + 2

= (3 – 4) + 2 = 1

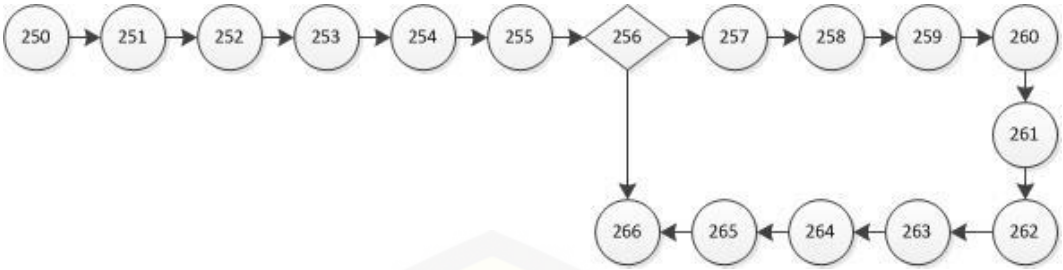
Tabel A. 24 Basis Test *Logout*

logout()	
Jalur 1 : 20 – 21 – 22 – 23	
Kasus	Klik logout
Target yang diharapkan	Menampilkan halaman Login
Hasil Pengujian	Berhasil

A.13 *White Box Encryption Process*

Tabel A. 25 Listing Program Encryption Process

250	var encrypt = function(data)
251	{
252	var key = key_to_binstring();
253	var iv = iv_to_binstring();
254	var block_mode = get_block_mode("ecb");
255	var plaintext = data;
256	if (!key)
257	return false;
258	console.log('Initializing AES Block Cipher with key: ' + key + '; block_mode: ' + block_mode + '; iv: ' + iv);
259	var _aes = new Twofish({key: key, block_mode: block_mode, iv: iv});
260	_aes.debug_mode = true;
261	var ciphertext = _aes.encrypt(plaintext);
262	console.log('Plaintext: ' + plaintext);
263	console.log('Ciphertext: ' + ciphertext);
264	console.log('Ciphertext (Base64-encoded): ' + convert.base64.encode(ciphertext));
265	return convert.base64.encode(ciphertext);
266	};



Gambar A. 13 Diagram Alir *EncryptionProcess*

CC = (Edge – Node) + 2

= (17 – 17) + 2 = 2

Tabel A. 26 Basis Test Encryption Process

encrypt()	
Jalur 1 : 250 – 251 – 252 – 253 – 254 – 255 – 256 – 257 – 258 – 259 – 260 – 261 – 262 – 263 – 264 – 265 – 266	
Kasus	Terdapat data rahasia yang akan dienkripi dan telah memasukan *key*
Target yang diharapkan	Data terenkripsi
Hasil Pengujian	Berhasil
Jalur 2 : 250 – 251 – 252 – 253 – 254 – 255 – 256 – 266	
Kasus	Terdapat data rahasia yang akan dienkripi dan belum memasukan *key*
Target yang diharapkan	Data tidak terenkripsi
Hasil Pengujian	Berhasil

A.14 *White Box Decyption Process*

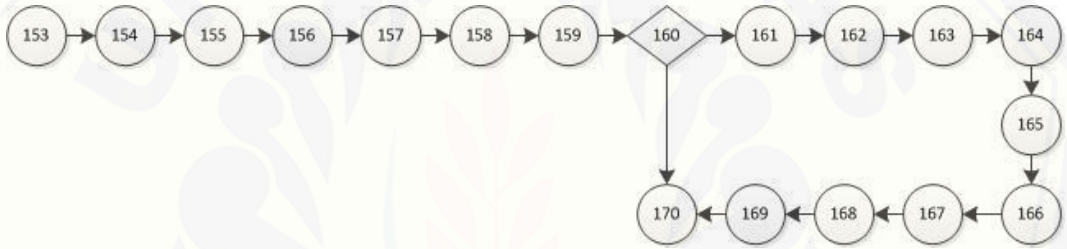
Tabel A. 27 Listing Program Decryption Process

153	var decrypt = function(data)
154	{
155	var key = key_to_binstring();
156	var iv = iv_to_binstring();

Dilanjutkan

Lanjutan

157	var block_mode = get_block_mode("ecb");
158	var encoded_ciphertext = data;
159	var ciphertext = convert.base64.decode(encoded_ciphertext);
160	if (!key)
161	return false;
162	console.log('Initializing AES Block Cipher with key: '+ key +'; block_mode: '+ block_mode +'; iv: '+ iv);
163	var _aes = new Twofish({key: key, block_mode: block_mode, iv: iv});
164	_aes.debug_mode = true;
165	var plaintext = _aes.decrypt(ciphertext);
166	console.log('Ciphertext (Base64-encoded): '+ encoded_ciphertext);
167	console.log('Ciphertext: '+ ciphertext);
168	console.log('Plaintext: '+ plaintext);
169	return plaintext;
170	}



Gambar A. 14 Diagram Alir *DecryptionProcess*

CC = (Edge – Node) + 2

= (18 – 18) + 2 = 2

Tabel A. 28 Basis Test *DecryptionProcess*

decrypt()	
Jalur 1 : 153 – 154 – 155 – 156 – 157 – 158 – 159 – 160 – 161 – 162 – 163 – 164 – 165 – 166 – 167 – 168 – 169 – 170	
Kasus	Terdapat data enkripsi yang akan didekripsi dan telah memasukan *key* yang benar
Target yang diharapkan	Data terdekrpsi

Dilanjutkan

Lanjutan

Hasil Pengujian	Berhasil
Jalur 2 : 153 – 154 – 155 – 156 – 157 – 158 – 159 – 160 – 170	
Kasus	Terdapat data enkripsi yang akan didekripsi dan belum memasukan *key* yang benar
Target yang diharapkan	Data tidak terdekripsi
Hasil Pengujian	Berhasil

LAMPIRAN B (*ENGINEER PROTOTYPE*)

B.1 Hasil *Engineer Prototype* Fitur *Login*

Hasil *engineer prototype* dari fitur *login* terletak pada 3 kelas yaitu, *view vlogin*, *controller login*, dan *model mlogin*.

Tabel B. 1 *View* vLogin

```
<title>Arsip</title>

<div class="row" style="position:relative; padding-top:20vh;">

<div>
<div class="input-group" style="padding-bottom:15px; text-align:center; width:100%;">
<img src="<?php echo base_url().'assets/img/sima.png';?>" width="25%">
</div>
</div>

</div>
<div class="row" style="padding-top:10px; width:100%;">
<div style="position:absolute; left:37.5%;width:25%;">
<form action="<?php echo base_url()."index.php/login/authentication"?>" method="post">
<div class="input-group" style="padding-bottom:15px;">
<span class="input-group-addon" id="sizing-addon2"><i class="fa fa-user"></i></span>
<input type="text" name="username" class="form-control" placeholder="Username" required
aria-describedby="sizing-addon2">
</div>
```

Dilanjutkan

Lanjutan

```
<div class="input-group" style="padding-bottom:15px;">
<span class="input-group-addon" id="sizing-addon2"><i class="fa fa-lock"></i></span>
<input type="password" name="password" class="form-control" placeholder="Password"
required aria-describedby="sizing-addon2">
</div>
<input type="submit" class="btn btn-primary" style="width:100%;" value="LOGIN"></input>
</form>
</div>


</div>
```

Tabel B. 2 *Controller* Login

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Login extends CI_Controller {
public function __construct(){
parent::__construct();
$this->load->model('Mlogin');
}
public function index()
{
$this->load->view('header');
$this->load->view('vlogin');
$this->load->view('footer_absolute');
}

public function authentication(){

$username       = $this->input->post('username');
$password       = $this->input->post('password');
$result         = $this->Mlogin->checkLogin($username,$password);
if($result){
$data     = $this->Mlogin->getDataUser($username)->result();

$parSession = array(
'username'                  => $username,
'tipe'                      => $data[0]->User_Type,
'nama'                      => $data[0]->NamaUser,
'kode'                      => $data[0]->Unit,
'auth'                      => TRUE
);
$this->session->set_userdata($parSession);
$this->session->set_flashdata('message','Welcome <b>'.$this->session-
>userdata('nama').'</b>');
redirect('dashboard');
}else{
$this->session->set_flashdata('message','Username atau Password salah');
```

Dilanjutkan

Lanjutan

```
redirect(base_url().'index.php/'.$this->input->get('page'));
}

}
}
```

Tabel B. 3 *Model* mLogin

```
<?php
if(!defined('BASEPATH')) exit ('No direct script access allowed');
class Mlogin extends CI_Model{

public function checkLogin($username, $password){
$result = $this->db->get_where("login",array('username' => $username, 'password' =>
$password));

if ($result->num_rows() > 0) {
return TRUE;
}
else {
return FALSE;
}
}

public function getDataUser($username){
$this->db->select('*');
$this->db->from('login');
$this->db->where('username',$username);
return $this->db->get();
}

}?>
```

B.2 Hasil *Engineer Prototype* FiturInputArsip

Hasil *engineer prototype* dari fitur input arsip terletak pada 3 kelas yaitu, *view*vinputarsip, *controller* inputarsip, dan *model* minputarsip.

Tabel B. 4 *View* Vinputarsip

```
<title>Input Arsip</title>
<!-- content -->
<div class="col-md-9" style="padding-left:40px; padding-top:0px;">
<div class="page-header">
<h1>Input <small>Arsip</small></h1>
</div>
<div class="row" style="padding-top:30px; padding-left:40px;">
<!--  <form method="post" action="<?= base_url() . "index.php/inputArsip/submit" ?>"> -->
<form id = "formInput">
<div class="row">
```

Dilanjutkan

Lanjutan

```
<h4><small>Kode Kelas</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<select id="kodekelas" name="kodekelas" class="form-control" required>
<option value="">Kode Kelas</option>
<?php
 foreach ($kodekelas->result_array() as $v) {
?>
<option value="<?= $v['id_kodekelas'] ?>"><?= $v['id_masterkriteria'] . "." . $v['id_kriteria']
?></option>
<?php
}
?>
</select>
</td>
</tr>
</table>
<?php if($this->session->userdata('kode')=="Admin Arsip"){ ?>
<h4><small>Unit Pengolahan</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="unitpengolahan" name="unitpengolahan"
placeholder="Unit Pengolahan"/>
</td>
</tr>
</table>
<?php } else{ ?>
<h4><small>Unit Pengolahan</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="unitpengolahan" name="unitpengolahan" value =
"<?= $this->session->userdata('kode'); ?>" placeholder="<?=$this->session-
>userdata('kode');?>" disabled/>
</td>
</tr>
</table>
<?php } ?>
<h4><small>Uraian</small></h4>
<table class="table" style="width:50%; ">
<tr>
<td>
<textarea class="form-control" id="uraian" name="uraian" placeholder="Uraian" rows="8"
required></textarea>
</td>
</tr>
</table>
<h4><small>Kurun Waktu Awal</small></h4>
<table class="table" style="width:50%;">
<tr>
```

Dilanjutkan

Lanjutan

```
<td>
<input type="text" class="form-control" id="kurunwaktuawal" name="kurunwaktuawal"
maxlength="4" placeholder="Tahun"/>
</td>
</tr>
</table>
<h4><small>Kurun Waktu Akhir</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="kurunwaktuakhir" name="kurunwaktuakhir"
maxlength="4" placeholder="Tahun"/>
</td>
</tr>
</table>

<h4><small>Jenis Dokumen</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<select id="jenisdokumen" name="jenisdokumen" class="form-control" required>
<option value="">Jenis Dokumen</option>
<?php
foreach ($jenisdokumen->result_array() as $v) {
?>
<option value="<?= $v['id_jenisdokumen'] ?>"><?= $v['nama'] ?></option>
<?php
}
?>
</select>
</td>
</tr>
</table>

<h4><small>Media Simpan</small></h4>
<table class="table" style="width:50%;">
<tr>

<!-- Media Simpan -->
<td>
<select id="mediasimpan" name="mediasimpan" class="form-control" required>
<option value="">Media Simpan</option>
<?php
foreach ($mediasimpan->result_array() as $v) {
?>
<option value="<?= $v['id_mediasimpan'] ?>"><?= $v['nama'] ?></option>
<?php
}
?>
</select>
</td>
</tr>
</table>
```

Dilanjutkan

Lanjutan

```
<h4><small>Kondisi</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<select id="kondisi" name="kondisi" class="form-control" required>
<option value="">Kondisi</option>
<?php
foreach ($kondisi->result_array() as $v) {
?>
<option value="<?= $v['id_kondisi'] ?>"><?= $v['nama'] ?></option>
<?php
}
?>
</select>
</td>
</tr>
</table>
<h4><small>Jumlah Berkas</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="jumlahberkas" name="jumlahberkas"
maxlength="2" placeholder="Jumlah Berkas" required: "number"/>
</td>
</tr>
</table>
<h4><small>Kode Ruang Nomor Lemari</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="nomorlemari" name="nomorlemari"
maxlength="7" placeholder="Kode dan Nomor Lemari"/>
</td>
</tr>
</table>
<h4><small>Nomor Boks</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="nomorboks" name="nomorboks" maxlength="7"
placeholder="Nomor Boks"/>
</td>
</tr>
</table>
<h4><small>Nomor Folder</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input type="text" class="form-control" id="nomorfolder" name="nomorfolder"
maxlength="7" placeholder="Nomor Folder"/>
</td>
</tr>
```

Dilanjutkan

Lanjutan

```
</table>
<h4><small>Rahasia</small></h4>
<table class="table" style="width:50%;">
<tr>
<td>
<input id="rahasia" type="radio" name="rahasia" value="1"/><b> Ya </b>
</td>
<td>
<input id="rahasia" type="radio" name="rahasia" value="0"/><b> Tidak </b>
</td>
</tr>
</table>
</div>
</form>
<table class="table" style="width:50%;">
<tr>
<td>
<button   class="btn btn-primary" id="simpan" style="width:100%;">SAVE</button>
</td>
</tr>
</table>
</div>
</div>
<div class="modal  fade" id="myModal" role="dialog">
<div class="modal-dialog modal-sm">
<!-- Modal content-->
<div class="modal-content">
<div class="modal-body">
<div class="form-group">
<input class="form-control" type="password" id="key" value="" placeholder="Input the
key!"/>
<button type="button" style="width:100%; margin-top:3%" class="btn btn-primary"
id="btnkey" data-dismiss="modal">Okay</button>
</div>
</div>
</div>

</div>
</div>
<script >
function stripHTML(text){
var regex = /(<([^>]+)>)/ig;
return text.replace(regex,"");
}

var hex ="";
var run_test = function()
{
var _aes = new Twofish({block_mode: ECB, pad_mode: ZeroPadding});
_aes.debug_mode = true;

if (_aes.test())
alert('TEST PASSED!\n\n(check the console for details)');
```

Dilanjutkan

Lanjutan

```
}

function toHex(str) {
var result = '';
for (var i=0; i<str.length; i++) {
result += str.charCodeAt(i).toString(16);
}
hex= result;
}

$(document).ready(function($) {
$('input[type=radio][name=rahasia]').change(function() {
if (this.value == 1) {
$('#myModal').modal('show');
}
});

$("#btnkey").click(function(){
toHex($("#key").val());
});

$('#simpan').click(function() {
var x = $('input[name=rahasia]:checked', '#formInput').val();
if (x == "1" ) {
if ($("#key").val()=="") {
alert("Please, input the key!");
}else{
var uraian = encrypt(stripHTML($("#uraian").val()));
var folder = encrypt(stripHTML($("#nomorfolder").val()));
var box = encrypt(stripHTML($("#nomorboks").val()));
var lemari = encrypt(stripHTML($("#nomorlemari").val()));

submit2(uraian,folder,box,lemari);
}
} else {
submit1();
}
});
});

var encrypt = function(data)
{
var key = key_to_binstring();
var iv = iv_to_binstring();
var block_mode = get_block_mode("ecb");
var plaintext = data;
if (!key)
return false;

console.log('Initializing AES Block Cipher with key: ' + key + '; block_mode: ' + block_mode +
'; iv: ' + iv);
var _aes = new Twofish({key: key, block_mode: block_mode, iv: iv});
_aes.debug_mode = true;
```

Dilanjutkan

Lanjutan

```
var ciphertext = _aes.encrypt(plaintext);
console.log('Plaintext: ' + plaintext);
console.log('Ciphertext: ' + ciphertext);
console.log('Ciphertext (Base64-encoded): ' + convert.base64.encode(ciphertext));
return convert.base64.encode(ciphertext);
};

var get_block_mode = function(str)
{
switch (str) {
case 'ecb':
return ECB;
case 'cbc':
return CBC;
case 'cfb':
return CFB;
}
};
var key_to_binstring = function()
{
if (!hex.match(/^[a-f0-9]+/i))
return alert('Key must be 16, 24, or 32 octets of hexadecimal!\n\nIf in doubt, just use the one
that comes pre-baked into this page.');
return convert.hex_to_binstring(hex);
};
var iv_to_binstring = function()
{
var iv = "9876543210fedcba9876543210fedcba";
if (!iv.match(/^[a-f0-9]{32}$/i))
return alert('Initial Vector must be 16 octets of hexadecimal!\n\nIf in doubt, just use the one that
comes pre-baked into this page.');
return convert.hex_to_binstring(iv);
};


var submit1 =  function(){
var kodekelas = $("#kodekelas option:selected").val();
var unitpengolahan = stripHTML($("#unitpengolahan").val());
var uraian = stripHTML($("#uraian").val());
var kurunwaktuawal = $("#kurunwaktuawal").val();
var kurunwaktuakhir = $("#kurunwaktuakhir").val();
var jenisdokumen = $("#jenisdokumen option:selected").val();
var mediasimpan = $("#mediasimpan option:selected").val();
var kondisi = $("#kondisi option:selected").val();
var jumlahberkas = $("#jumlahberkas").val();
var nomorlemari = $("#nomorlemari").val();
var nomorboks = $("#nomorboks").val();
var nomorfolder = $("#nomorfolder").val();
var rahasia = $("#rahasia:checked").val();

$.ajax({
url: '<?= base_url() . "index.php/inputArsip/submit" ?>',
```

Dilanjutkan

Lanjutan

```
dataType: 'text',
type: 'post',
contentType: 'application/x-www-form-urlencoded',
data: {
kodekelas : kodekelas,
unitpengolahan : unitpengolahan,
uraian : uraian,
kurunwaktuawal : kurunwaktuawal,
kurunwaktuakhir : kurunwaktuakhir,
jenisdokumen : jenisdokumen,
mediasimpan : mediasimpan,
kondisi : kondisi,
jumlahberkas : jumlahberkas,
nomorlemari : nomorlemari,
nomorboks : nomorboks,
nomorfolder : nomorfolder,
rahasia : rahasia,
} ,
success: function( data, textStatus, jQxhr ){
$('#response pre').html( data );
alert("Entry success!");
kosong();
window.location = '<?= base_url()."index.php/Arsip" ?>';

},
error: function( jqXhr, textStatus, errorThrown ){
console.log( errorThrown );
alert("Entry Failed!");
}
});
}

var submit2 =  function(uraian,folder,box,lemari){
var kodekelas = $("#kodekelas option:selected").val();
var unitpengolahan = $("#unitpengolahan").val();
var uraian = uraian;
var kurunwaktuawal = $("#kurunwaktuawal").val();
var kurunwaktuakhir = $("#kurunwaktuakhir").val();
var jenisdokumen = $("#jenisdokumen option:selected").val();
var mediasimpan = $("#mediasimpan option:selected").val();
var kondisi = $("#kondisi option:selected").val();
var jumlahberkas = $("#jumlahberkas").val();
var nomorlemari = lemari;
var nomorboks = box;
var nomorfolder = folder;
var rahasia = $("#rahasia:checked").val();

$.ajax({
url: '<?= base_url() . "index.php/inputArsip/submit" ?>',
dataType: 'text',
type: 'post',
contentType: 'application/x-www-form-urlencoded',
data: {
```

Dilanjutkan

Lanjutan

```
kodekelas : kodekelas,
unitpengolahan : unitpengolahan,
uraian : uraian,
kurunwaktuawal : kurunwaktuawal,
kurunwaktuakhir : kurunwaktuakhir,
jenisdokumen : jenisdokumen,
mediasimpan : mediasimpan,
kondisi : kondisi,
jumlahberkas : jumlahberkas,
nomorlemari : nomorlemari,
nomorboks : nomorboks,
nomorfolder : nomorfolder,
rahasia : rahasia,
} ,
success: function( data, textStatus, jQxhr ){
$('#response pre').html( data );
alert("Entry success!");
kosong();
window.location = '<?= base_url()."index.php/dekripArsip" ?>';
},
error: function( jqXhr, textStatus, errorThrown ){
console.log( errorThrown );
alert("Entry Failed!");
}
});

}

var kosong = function(){
$("#kodekelas").val("");
$("#unitpengolahan").val("");
$("#uraian").val("");
$("#kurunwaktuawal").val("");
$("#kurunwaktuakhir").val("");
$("#jenisdokumen").val("");
$("#mediasimpan").val("");
$("#kondisi").val("");
$("#jumlahberkas").val("");
$("#nomorlemari").val("");
$("#nomorboks").val("");
$("#nomorfolder").val("");
$("#rahasia").val("");
// $("#key").val("");
}
</script>
```

Tabel B. 5 *Controller* Inputarsip

```
<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');
define('IV_SIZE', mcrypt_get_iv_size(MCRYPT_TWOFISH, MCRYPT_MODE_CBC));
```

Dilanjutkan

Lanjutan

```
class InputArsip extends CI_Controller{

 public function __construct(){
parent::__construct();
if(!$this->session->userdata('auth')){
redirect ('login');
}else{

$this->load->model('MInputArsip');
}
}

public function index(){
$data["controller"] = $this;
$data["kondisi"]=$this->MInputArsip->getkondisi();
$data["mediasimpan"]=$this->MInputArsip->getmediasimpan();
$data["kodekelas"]=$this->MInputArsip->getkodekelas();
$data["jenisdokumen"]=$this->MInputArsip->getjenisdokumen();
$this->load->view('admin/header');
$this->load->view('admin/vInputArsip',$data);
}

public function submit(){
$kodekelas = htmlspecialchars($this->input->post("kodekelas"));
$unitpengolahan = htmlspecialchars($this->input->post("unitpengolahan"));
$uraian = htmlspecialchars($this->input->post("uraian"));
$kurunwaktuawal = htmlspecialchars($this->input->post("kurunwaktuawal"));
$kurunwaktuakhir = htmlspecialchars($this->input->post("kurunwaktuakhir"));
$jenisdokumen = htmlspecialchars($this->input->post("jenisdokumen"));
$mediasimpan = htmlspecialchars($this->input->post("mediasimpan"));
$kondisi = htmlspecialchars($this->input->post("kondisi"));
$jumlahberkas = htmlspecialchars($this->input->post("jumlahberkas"));
$nomorlemari = htmlspecialchars($this->input->post("nomorlemari"));
$nomorboks = htmlspecialchars($this->input->post("nomorboks"));
$nomorfolder = htmlspecialchars($this->input->post("nomorfolder"));
$rahasia = htmlspecialchars($this->input->post("rahasia"));

$result = $this->MInputArsip-
>insert($kodekelas,$unitpengolahan,$uraian,$kurunwaktuawal,$kurunwaktuakhir,
$jenisdokumen,$mediasimpan,$kondisi,$jumlahberkas,$nomorlemari,$nomorboks,$nomorfolde
r,$rahasia);

if($result){
$this->session->set_flashdata('message','SUKSES ENTRY DATA');
redirect('inputArsip');
}else{
$this->session->set_flashdata('message','GAGAL ENTRY DATA');
redirect('base_url()'.$this->input->get('page'));
}
}

}
?>
```

Tabel B. 6 *Model* Minputarsip

```
<?php
class MInputArsip extends CI_Model{

function
insert($kodekelas,$unitpegolahan,$uraian,$kurunwaktuawal,$kurunwaktuakhir,$jenisdokumen,
$mediasimpan,$kondisi,$jumlahberkas,$nomorlemari,$nomorboks,$nomorfolder,$rahasia){

$data = array('id_arsip' => NULL ,
'id_kodekelas' => $kodekelas ,
'unitpengolahan' => $unitpegolahan ,
'uraian' => $uraian ,
'kurunwaktuawal' => $kurunwaktuawal,
'kurunwaktuakhir' => $kurunwaktuakhir,
'id_jenisdokumen' => $jenisdokumen,
'id_mediasimpan' => $mediasimpan,
'id_kondisi' => $kondisi,
'jumlahberkas' => $jumlahberkas,
'kodelemari' => $nomorlemari,
'nomorbox' => $nomorboks,
'nomorfolder' => $nomorfolder,
'rahasia' => $rahasia );

$this->db->set($data);
$insert =$this->db->insert('arsip');
if($insert ){
return true;
}else{
return false;
}
}

function getkodekelas(){
$this->db->from('kodekelas');
return $this->db->get();
}

function getjenisdokumen(){
$this->db->from('jenisdokumen');
return $this->db->get();
}

function getmediasimpan(){
$this->db->from('mediasimpan');
return $this->db->get();
}

function getkondisi(){
$this->db->from('kondisi');
return $this->db->get();
}
} ?>
```

B.3 Hasil *EngineerPrototype* Fitur Daftar Arsip

Hasil *engineer prototype* dari fitur daftar arsip terletak pada 3 kelas yaitu, *view* varsip, *controller* arsip, dan *model* marsip.

Tabel B. 7 View Varsip

```
<title>View Arsip</title>
<!-- content -->
<div class="col-md-9" style="padding-left:40px; padding-top:0px;">
<div class="page-header">
<h1>View <small>Arsip</small></h1>
</div>
<div class="row" style="padding-top:30px; padding-left:30px;">
<?php
if(isset($arsip)){
?>
<h5><b>Result :</b></h5>
<table class="table-hover" width="80%" style="width:80%;margin-top:20px;margin-
left:40px;">
<tr align="center">
<td style="width:10%;"><b>No.</b></td>
<td style="width:10%;"><b>Kode Kelas</b></td>
<td style="width:50%;"><b>Uraian</b></td>
<td style="width:10%;"><b>Nomor Boks</b></td>
<td style="width:10%;"><b>Nomor Folder</b></td>
<td colspan="2"><b>Aksi</b></td>
</tr>

<?php
$i=$this->input->get('per_page')+1;
foreach($arsip->result_array() as $c){
?>
<tr align="center">
<input type="hidden" value="<?=$c['id_arsip']?>"/>
<td ><?php echo $i;?></td>
<td title="<?= $c['nama'] ?>"><?=$c['id_masterkriteria']."."$c['id_kriteria']?></td>
<td style="text-align:left"><?=$c['uraian']?></td>
<td><?=$c['nomorbox']?></td>
<td><?=$c['nomorfolder']?></td>
<td>  <a
href="<?=base_url()."index.php/detailArsip/?kode=$c[id_arsip]"?>">Detail</a></td>
<td>  <a
href="<?=base_url()."index.php/editArsip/?kode=$c[id_arsip]"?>">Edit</a></td>
<td>  <a
href="<?=base_url()."index.php/Arsip/delete?kode=$c[id_arsip]"?>">Delete</a></td>
</tr>
<?php
$i++;
}
?>
<?php
}
```

Dilanjutkan

Lanjutan

```
?>
</table>
</div>
</div>
<?php if ($cek){

?>
<center><div class="pagination" style="position:relative;right:100px;" align="center">
<?php
echo $this->pagination->create_links();
?>
</div></center>

<?php }
?>
</div>
</div>
<!-- div container -->
</div>
```

Tabel B. 8*Controller* Arsip

```
<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');
define('IV_SIZE', mcrypt_get_iv_size(MCRYPT_TWOFISH, MCRYPT_MODE_CBC));

class Arsip extends CI_Controller {

public function __construct(){
parent::__construct();
if(!$this->session->userdata('auth')){
redirect ('login');
}else{
$this->load->model('MArsip');
}
}

public function index(){
$data['controller'] = $this;

$data["cek"]=TRUE;
// pagination

// set limit data per page
$limit = 10;
// set total data
$total = $this->MArsip->get_jumlah_arsip()->result();
// load library pagination
$this->load->library('pagination');
// config pagination
$config['base_url'] = base_url().'index.php/arsip?';
$config['total_rows'] = $total[0]->jumlah;
```

Dilanjutkan

lanjutan

```
$config['per_page'] = $limit;
$config['page_query_string'] = TRUE;
// $config['query_string_segment'] = 'page';
$this->pagination->initialize($config);
// set limit max page yang akan diload
$page = $this->input->get('per_page');
if($page==""){
$page="0";
}
if($this->input->get_post('tahun')!=""){
$data["cek"]=FALSE;
$data["arsip"]=$this->MArsip->get_arsip($this->input->post('tahun'));
}else{
$data["arsip"]=$this->MArsip->get_all_arsip($page,$limit);
}
$this->load->view('admin/header');
$this->load->view('admin/vArsip',$data);
$this->load->view('footer');
}

public function encArsip(){
$data['controller'] = $this;

$data["cek"]=TRUE;
// pagination

// set limit data per page
$limit = 10;
// set total data
$total = $this->MArsip->get_jumlah_arsip()->result();
// load library pagination
$this->load->library('pagination');
// config pagination
$config['base_url'] = base_url().'index.php/arsip/encArsip?';
$config['total_rows'] = $total[0]->jumlah;
$config['per_page'] = $limit;
$config['page_query_string'] = TRUE;
// $config['query_string_segment'] = 'page';
$this->pagination->initialize($config);
// set limit max page yang akan diload
$page = $this->input->get('per_page');
if($page==""){
$page="0";
}
if($this->input->get_post('tahun')!=""){
$data["cek"]=FALSE;
$data["arsip"]=$this->MArsip->get_arsip($this->input->post('tahun'));
}else{
$data["arsip"]=$this->MArsip->get_enc_arsip($page,$limit);
}
$this->load->view('admin/header');
$this->load->view('admin/vencArsip',$data);
$this->load->view('footer');
```

Dilanjutkan

Lanjutan

```
}

public function delete(){
$result=$this->MArsip-> deletetransaksi($this->input->get('kode'));
if($result){
$this->session->set_flashdata('message','DELETED');
redirect ('Arsip');
}else{
$this->session->set_flashdata('message','Failed to delete data');
redirect(base_url().'index.php/'.$this->input->get('page'));
}
}
}
?>
```

Tabel B. 9 Model MArsip

```
<?php

class MArsip extends CI_Model{

function get_jumlah_arsip(){
$this->db->select('count(*) as jumlah');
return $this->db->get('arsip');
}

function get_arsip($thn){
$ambil= $this->db->query("select * from arsip where 'kurunwaktuawal' like '%".$thn."%'");
return $ambil;
}

function get_all_arsip($page, $limit){
if ($this->session->userdata('kode')!="Admin Arsip") {
$ambil= $this->db->query("select * from arsip a join kodekelas kk on a.id_kodekelas =
kk.id_kodekelas where rahasia = '0' and unitpengolahan = '".$this->session->userdata('kode')."'
order by kurunwaktuawal limit ".$page.",".$limit."");
}else{
$ambil= $this->db->query("select * from arsip a join kodekelas kk on a.id_kodekelas =
kk.id_kodekelas where rahasia = '0'
order by kurunwaktuawal limit ".$page.",".$limit."");
}
return $ambil;
}

function get_enc_arsip($page,$limit){
if ($this->session->userdata('kode')!="Admin Arsip") {
$ambil = $this->db->query("select * from arsip a join kodekelas kk on a.id_kodekelas =
kk.id_kodekelas where rahasia='1' and unitpengolahan = '".$this->session->userdata('kode')."'
order by kurunwaktuawal limit ".$page.",".$limit."");
}else{
$ambil = $this->db->query("select * from arsip a join kodekelas kk on a.id_kodekelas =
kk.id_kodekelas where rahasia='1'
```

Dilanjutkan

Lanjutan

```
order by kurunwaktuawal limit ".$page.",".$limit."");
}
return $ambil;
}

function deletetransaksi($kode){
$delete = $this->db->delete('arsip',array('id_arsip' => $kode));
if($delete){
return true;
}else{
return false;
}
}

}?>
```

B.4 Hasil *Engineer Prototype* Fitur Daftar Arsip Rahasia

Hasil *engineer prototype* dari fitur daftar arsip rahasia terletak pada 3 kelas yaitu, *view vdekriparsip*, *controller dekriparsip*, dan *model marsip*.

Tabel B. 10 View Vdekriparsip

```
<title>View Arsip</title>
<!-- content -->
<div class="col-md-9" style="padding-left:40px; padding-top:0px;">
<div class="page-header">
<h1>View <small>Arsip Rahasia</small></h1>
</div>
<div class="row" style="padding-top:30px; padding-left:30px;">

<?php
if(isset($arsip)){
?>
<h5><b>Result :</b></h5>
<table class="table-hover" width="80%" style="width:80%;margin-top:20px;margin-
left:40px;">
<tr align="center">
<td style="width:10%;"><b>No.</b></td>
<td style="width:10%;"><b>Kode Kelas</b></td>
<td style="width:50%;"><b>Uraian</b></td>
<td style="width:10%;"><b>Nomor Boks</b></td>
<td style="width:10%;"><b>Nomor Folder</b></td>
<td colspan="2"><b>Aksi</b></td>
</tr>

<?php
$i=$this->input->get('per_page')+1;
foreach($arsip->result_array() as $c){
?>
<tr align="center">
```

Dilanjutkan

Lanjutan

```
<input type="hidden" value="<?=$c['id_arsip']?>"/>
<td ><?php echo $i;?></td>
<td title="<?= $c['nama'] ?>"><?=$c['id_masterkriteria']."."$c['id_kriteria']?></td>
<td style="text-align:left" id="uraian<?=$i?>"><?=$c['uraian']?></td>
<td id="box<?=$i?>"><?=$c['nomorbox']?></td>
<td id="folder<?=$i?>"><?=$c['nomorfolder']?></td>
<td>  <a
href="<?=base_url()."index.php/detailArsip/dec/?kode=$c[id_arsip]"?>">Detail</a></td>
<td>  <a
href="<?=base_url()."index.php/editArsip/dec/?kode=$c[id_arsip]"?>">Edit</a></td>
<td>  <a
href="<?=base_url()."index.php/dekripArsip/delete?kode=$c[id_arsip]"?>">Delete</a></td>
</tr>
<?php
$i++;
}
?>
<?php
}
?>
</table>
</div>

<?php if ($cek){

?>
<center><div class="pagination" style="position:relative;right:100px;" align="center">
<?php
echo $this->pagination->create_links();
?>
</div></center>

<?php }
?>
</div>

</div>
<!-- div container -->
</div>

<div class="modal fade" data-backdrop="static" id="myModal" role="dialog">
<div class="modal-dialog modal-sm ">
<!-- Modal content-->
<div class="modal-content">
<div class="modal-body">
<div class="form-group">
<input class="form-control" type="password" id="key" value="" placeholder="Input the
key!"/>
<button type="button" style="width:100%; margin-top:3%" class="btn btn-primary" id="okay"
data-dismiss="modal">Okay</button>
</div>
</div>
</div>
```

Dilanjutkan

Lanjutan

```
</div>
</div>

<script type="text/javascript">
var hex ="";
$("#okay").click(function(){
if($("#key").val()!=""){
toHex($("#key").val());
tampil();
hex="";
$("#key").val("");
}else{
alert("Please, Input the key!");
window.location= '<?=base_url()."index.php/dekripArsip"?>';
}
});

function toHex(str) {
var result = '';
for (var i=0; i<str.length; i++) {
result += str.charCodeAt(i).toString(16);
}
hex= result;
}

var tampil = function(){
var coba = "  <?php $a=$this->input->get('per_page')+1; foreach($arsip->result_array() as $c){
$a++; } ?>";
var loop =" <?=$a?>";
for (var i = 1; i < loop; i++) {
var folder = $("#folder"+i).html();
var plain = decrypt(folder);
var box = $("#box"+i).html();
var plain2 = decrypt(box);
var uraian = $("#uraian"+i).html();
var plain3 = decrypt(uraian);
$("#folder"+i).html(plain);
$("#box"+i).html(plain2);
$("#uraian"+i).html(plain3);

if (!plain2) {
alert("Wrong key!");
window.location = '<?=base_url()."index.php/dekripArsip"?>';
}else{
}
};
}

$(document).ready(function(){
$('#myModal').modal('show');
});
```

Dilanjutkan

Lanjutan

```
var get_block_mode = function(str)
{
switch (str) {
case 'ecb':
return ECB;
case 'cbc':
return CBC;
case 'cfb':
return CFB;
}
};
var key_to_binstring = function()
{
if (!hex.match(/^[a-f0-9]+/i))
return alert('Key must be 16, 24, or 32 octets of hexadecimal!\n\nIf in doubt, just use the one
that comes pre-baked into this page.');
return convert.hex_to_binstring(hex);
};
var iv_to_binstring = function()
{
var iv = "9876543210fedcba9876543210fedcba";
if (!iv.match(/^[a-f0-9]{32}$/i))
return alert('Initial Vector must be 16 octets of hexadecimal!\n\nIf in doubt, just use the one that
comes pre-baked into this page.');
return convert.hex_to_binstring(iv);
};

var decrypt = function(data)
{

var key         = key_to_binstring();
var iv          = iv_to_binstring();
var block_mode      = get_block_mode("ecb");
var encoded_ciphertext  = data;
var ciphertext      = convert.base64.decode(encoded_ciphertext);

if (!key)
return false;

console.log('Initializing AES Block Cipher with key: '+ key +'; block_mode: '+ block_mode +';
iv: '+ iv);

var _aes = new Twofish({key: key, block_mode: block_mode, iv: iv});

_aes.debug_mode = true;

var plaintext = _aes.decrypt(ciphertext);

console.log('Ciphertext (Base64-encoded): '+ encoded_ciphertext);
console.log('Ciphertext: '+ ciphertext);
console.log('Plaintext: '+ plaintext);
return plaintext;
}
</script>
```

Tabel B. 11*Controller* Dekriparsip

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
define('IV_SIZE', mcrypt_get_iv_size(MCRYPT_TWOFISH, MCRYPT_MODE_CBC));

class dekripArsip extends CI_Controller {
public function __construct(){
parent::__construct();
if(!$this->session->userdata('auth')){
redirect ('login');

}else{

$this->load->model('MArsip');
}
}

public function index(){
$data['controller'] = $this;

$data["cek"]=TRUE;
// pagination

// set limit data per page
$limit = 10;
// set total data
$total = $this->MArsip->get_jumlah_arsip()->result();
// load library pagination
$this->load->library('pagination');
// config pagination
$config['base_url'] = base_url().'index.php/dekripArsip?';
$config['total_rows'] = $total[0]->jumlah;
$config['per_page'] = $limit;
$config['page_query_string'] = TRUE;
// $config['query_string_segment'] = 'page';
$this->pagination->initialize($config);
// set limit max page yang akan diload
$page = $this->input->get('per_page');
if($page==""){
$page="0";
}
if($this->input->get_post('tahun')!=""){
$data["cek"]=FALSE;
$data["arsip"]=$this->MArsip->get_arsip($this->input->post('tahun'));
}else{
$data["arsip"]=$this->MArsip->get_enc_arsip($page,$limit);

}
$this->load->view('admin/header');
$this->load->view('admin/vDekripArsip',$data);
$this->load->view('footer');
}

public function delete(){
```

Dilanjutkan

Lanjutan

```
$result=$this->MArsip-> deletetransaksi($this->input->get('kode'));
if($result){
$this->session->set_flashdata('message','DELETED');
redirect ('dekripArsip');
}else{
$this->session->set_flashdata('message','Failed to delete data');
redirect(base_url().'index.php/'.$this->input->get('page'));
}
}
}
?>
```

B.5 Hasil *EngineerPrototype* Fitur Pengecekan Arsip

Hasil *engineer prototype* dari fitur pengecekan arsip terletak pada 3 kelas yaitu, *view vpengecekan*, *controller pengecekan*, dan *model mpengecekan*.

Tabel B. 12 View VPengecekan

```
<title>Pengecekan Arsip</title>
<!-- content -->
<div class="col-md-9" style="padding-left:40px; padding-top:0px;">
<div class="page-header">
<h1>Pengecekan <small>Arsip</small></h1>
</div>
<div class="row" style="padding-top:30px; padding-left:30px;">
<?php
if(isset($arsip)){
?>
<h5><b>Result :</b></h5>
<form method = "post" action="<?= base_url()."index.php/Pengecekan/cek"?>">
<table class="table-hover" width="80%" style="width:80%;margin-top:20px;margin-
left:40px;">
<tr align="center">
<td style="width:10%;"><b>No.</b></td>
<td style="width:10%;"><b>Kode Kelas</b></td>
<td style="width:50%;"><b>Uraian</b></td>
<!-- <td style="width:10%;"><b>Nomor Boks</b></td> -->
<!-- <td style="width:10%;"><b>Nomor Folder</b></td> -->
<td colspan="2"><b>Aksi</b></td>
</tr>

<?php
$i=$this->input->get('per_page')+1;
foreach($arsip->result_array() as $c){
?>
<tr align="center">
<td ><?php echo $i;?></td>
<td title="<?= $c['nama'] ?>"><?=$c['id_masterkriteria']."".$c['id_kriteria']?></td>
<td style="text-align:left" id="uraian<?=$i?>"><?=$c['uraian']?></td>
<!-- <td><?=$c['nomorbox']?></td> -->
```

Dilanjutkan

Lanjutan

```
<!-- <td><?=$c['nomorfolder']?></td> -->
<td><input type="checkbox" name="checklist[]" class="checklist" id="checklist<?= $i; ?>"
value="<?=$c['id_arsip']?>"></a></td>
</tr>
<?php
$i++;
}
?>
<?php
}
?>
</table>
</div>
<input type="submit" class="btn btn-primary" value="Selesai" id="sub" style="width:20%;
margin-left:40%; margin-top:30px; margin-bottom:20px" disabled></input>
</form>
</div>
<?php if ($cek){

?>
<center><div class="pagination" style="position:relative;right:100px;" align="center">
<?php
echo $this->pagination->create_links();
?>
</div></center>

<?php }
?>
</div>
</div>

<!-- div container -->
</div>

<div class="modal fade" data-backdrop="static" id="myModal" role="dialog">
<div class="modal-dialog modal-sm">
<!-- Modal content-->
<div class="modal-content">
<div class="modal-body">
<div class="form-group">
<input class="form-control" type="password" id="key" value="" placeholder="Input the
key!"/>
<button type="button" style="width:100%; margin-top:3%" class="btn btn-primary" id="okay"
data-dismiss="modal">Okay</button>
</div>
</div>
</div>

</div>
</div>

<script type="text/javascript">
```

Dilanjutkan

Lanjutan

```
var hex ="";

$("#okay").click(function(){

if($("#key").val()!=""){

toHex($("#key").val());
tampil();
hex="";
$("#key").val("");

}

});


function toHex(str) {
var result = '';
for (var i=0; i<str.length; i++) {
result += str.charCodeAt(i).toString(16);
}
hex= result;
}

var tampil = function(){
var coba = "  <?php $a=$this->input->get('per_page')+1; foreach($arsip->result_array() as $c){
$a++; } ?>";
var loop =" <?=$a?>";
for (var i = 1; i < loop; i++) {
var uraian = $("#uraian"+i).html();
var plain3 = decrypt(uraian);

if (!plain3) {
$("#uraian"+i).html(uraian);
}else{

$("#uraian"+i).html(plain3);

}

};
}

$(document).ready(function(){
$("#sub").attr("disabled","true");
$(".checklist").change(function(){
var a = $(".checklist").is(":checked");
if (a) {
$("#sub").removeAttr("disabled");
}else{
$("#sub").attr("disabled","true");
};
});
```

Dilanjutkan

Lanjutan

```
$('#myModal').modal('show');

});

var get_block_mode = function(str)
{
switch (str) {
case 'ecb':
return ECB;
case 'cbc':
return CBC;
case 'cfb':
return CFB;
}
};
var key_to_binstring = function()
{
if (!hex.match(/^[a-f0-9]+/i))
return alert('Key must be 16, 24, or 32 octets of hexadecimal!\n\nIf in doubt, just use the one
that comes pre-baked into this page.');
return convert.hex_to_binstring(hex);
};
var iv_to_binstring = function()
{
var iv = "9876543210fedcba9876543210fedcba";
if (!iv.match(/^[a-f0-9]{32}$/i))
return alert('Initial Vector must be 16 octets of hexadecimal!\n\nIf in doubt, just use the one that
comes pre-baked into this page.');
return convert.hex_to_binstring(iv);
};

var decrypt = function(data)
{
var key         = key_to_binstring();
var iv          = iv_to_binstring();
var block_mode      = get_block_mode("ecb");
var encoded_ciphertext  = data;
var ciphertext      = convert.base64.decode(encoded_ciphertext);

if (!key)
return false;

console.log('Initializing AES Block Cipher with key: '+ key +'; block_mode: '+ block_mode +';
iv: '+ iv);

var _aes = new Twofish({key: key, block_mode: block_mode, iv: iv});

_aes.debug_mode = true;

var plaintext = _aes.decrypt(ciphertext);

console.log('Ciphertext (Base64-encoded): '+ encoded_ciphertext);
```

Dilanjutkan

Lanjutan

```
console.log('Ciphertext: '+ ciphertext);
console.log('Plaintext: '+ plaintext);
return plaintext;
}

</script>
```

Tabel B. 13 Controller Pengecekan

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
define('IV_SIZE', mcrypt_get_iv_size(MCRYPT_TWOFISH, MCRYPT_MODE_CBC));

class Pengecekan extends CI_Controller {

public function __construct(){
parent::__construct();
if(!$this->session->userdata('auth')){
redirect ('login');
}else{
$this->load->model('MPengecekan');
}
}

function decrypt ($ciphertext) {
$combo = base64_decode($ciphertext);
$iv = substr($combo, 0, IV_SIZE);
$crypt = substr($combo, IV_SIZE, strlen($combo));
$plaintext = mcrypt_decrypt(MCRYPT_TWOFISH, md5('R3sT!nP34c3'), $crypt,
MCRYPT_MODE_CBC, $iv);
return $plaintext;
}

public function index(){
$data["controller"]=$this;
$data["cek"]=TRUE;
// pagination

// set limit data per page
$limit = 10;
// set total data
$total = $this->MPengecekan->get_jumlah_arsip()->result();
// load library pagination
$this->load->library('pagination');
// config pagination
$config['base_url'] = base_url().'index.php/arsip?';
$config['total_rows'] = $total[0]->jumlah;
$config['per_page'] = $limit;
$config['page_query_string'] = TRUE;
// $config['query_string_segment'] = 'page';
$this->pagination->initialize($config);
// set limit max page yang akan diload
$page = $this->input->get('per_page');
```

Dilanjutkan

Lanjutan

```
if($page==""){
$page="0";
}
if($this->input->get_post('tahun')!=""){
$data["cek"]=FALSE;
$data["arsip"]=$this->MPengecekan->get_arsip($this->input->post('tahun'));
}else{
$data["arsip"]=$this->MPengecekan->get_all_arsip($page,$limit);
}
$this->load->view('admin/header');
$this->load->view('admin/vPengecekan',$data);
$this->load->view('footer');
}

public function cek(){
$check = $this->input->post('checklist');
foreach ($check as $cek) {
$result = $this->MPengecekan->cek($cek);
}
if($result){
$this->session->set_flashdata('message','SUKSES ENTRY DATA');
redirect('Pengecekan');
}else{
$this->session->set_flashdata('message','GAGAL ENTRY DATA');
redirect('base_url()'.$this->input->get('page'));
}
}
}
?>
```

Tabel B. 14 Model MPengecekan

```
<?php
class MPengecekan extends CI_Model{

function cek($check){

$cek_arsip = $this->db->query("UPDATE `arsip` set cek = '1'
WHERE `id_arsip` ='".$check."'");


if($cek_arsip){
return true;
}else{
return false;
}
}

function get_jumlah_arsip(){
$this->db->select('count(*) as jumlah');
return $this->db->get('arsip');
```

Dilanjutkan

Lanjutan

```
function get_arsip($thn){
$ambil= $this->db->query("select * from arsip where 'kurunwaktuawal' like '%".$thn."%'");
return $ambil;
}

function get_all_arsip($page, $limit){
$ambil= $this->db->query("select * from arsip a join kodekelas kk on a.id_kodekelas =
kk.id_kodekelas where a.cek = 0
order by kurunwaktuawal limit ".$page.",".$limit."");
return $ambil;
}

function getkodekelas(){
$this->db->from('kodekelas');
return $this->db->get();
}

function getjenisdokumen(){
$this->db->from('jenisdokumen');
return $this->db->get();
}

function getmediasimpan(){
$this->db->from('mediasimpan');
return $this->db->get();
}

function getkondisi(){
$this->db->from('kondisi');
return $this->db->get();
}
} ?>
```

LAMPIRAN C (TRANSKRIP WAWANCARA)

Tabel C. 1 Transkrip Wawancara

No	Pertanyaan	Jawaban
1	Bagaimana alur pengarsipan di PT. Angkasa Pura I ?"	Dokumen-dokumen yang akan diarsipkan dicatat terlebih dahulu datanya seperti nomor, uraian, tempat penyimpanan, dan sebagainya kemudian disimpan kedalam lemari. Divisi lain yang menyerahkan dokumen mereka harus menyertakan daftar dokumen yang diserahkan untuk dicek kelengkapannya
2	Apakah sudah ada sistem untuk mencatat dokumen arsip?	Belum ada, sementara masih dicatat menggunakan *microsoft excel*
3	Apakah ada sistem keamamanan untuk menjaga catatan tersebut?	"tidak ada, kan hanya pakai *microsoft excel*"