Hello World
Jurnal Ilmu Komputer
https://jurnal.ilmubersama.com/index.php/hello_world

Kriptografi

Pengamanan Arsip dengan Algoritma Enkripsi AES-256 untuk Web App E-Arsip Yayasan Universitas Islam Sumatera Utara

Ibnu Hajar

Fakultas Teknik, Teknik Informatika, Universitas Islam Sumatera Utara, Medan, Indonesia

INFORMASI ARTIKEL

Diterima Redaksi: 21 Mei 2022
Revisi Akhir: 20 Juni 2022
Diterbitkan *Online*: 1 Juli 2022

KATA KUNCI

E-Arsip, Pengamanan Arsip, Advanced Encryption Standard.

KORESONDENSI

Phone: +62-812-6375-6838
E-mail: ibnuhajardewantara62@gmail.com

A B S T R A K

Pengarsipan merupakan hal yang biasa dilakukan apabila memiliki data dan fakta yang dianggap penting contoh umum benda yang sering diarsipkan adalah surat, dengan ada nya arsip memudahkan dalam menyimpan data-data surat tersebut. Pengolahan arsip di Yayasan Universitas Islam Sumatera Utara masih menggunakan metode konvensional. Sehingga dalam menyimpan data surat masih berupa filling cabinet atau penyimpanan file di dalam lemari kabinet. Selain memakan space ruangan yang besar secara fisik, juga mempersulit apabila data yang ingin diketahui harus dicari terlebih dahulu di lemari kabinet. Belum lagi permasalahan lain seperti keamanan selain rentan terhadap api, lapuk, juga rentan terhadap rayap. Oleh karena itu dibutuhkan solusi untuk mengatasi masalah masalah tersebut. Penelitian berfokus pada perancangan web app e-arsip dan keamanan arsip menggunakan algoritma AES-256 pada pengarsipan surat Yayasan Universitas Islam Sumatera Utara. Tujuan dari penelitian ini adalah untuk membangun sebuah web aplikasi e-arsip dan mengamankan data-data file surat masuk dan file surat keluar di Yayasan Universitas Islam Sumatera Utara, Kemudian aplikasi akan dilengkapi dengan algortima AES-256 yang akan diimplementasikan untuk mengamankan file-file surat yang tersimpan didalam web app e-arsip dan aplikasi web app e-arsip. Hasil akhir dari penelitian berupa aplikasi web app e-arsip untuk menyimpan data surat pada Yayasan Universitas Islam Sumatera Utara yang dilengkapi dengan algoritma AES-256 untuk mengamankan file dan aplikasi web app e-arsip.

PENDAHULUAN

Pengarsipan merupakan hal yang biasa dilakukan apabila memiliki data dan fakta yang dianggap penting contoh umum benda yang sering diarsipkan adalah surat, dengan ada nya arsip memudahkan dalam menyimpan data data tersebut. Pengolahan arsip di Yayasan Universitas Islam Sumatera Utara masih menggunakan metode konvensional. Sehingga dalam menyimpan data surat masih berupa filling cabinet atau penyimpanan file di dalam lemari kabinet. Selain memakan space ruangan yang besar secara fisik, juga mempersulit apabila data yang ingin diketahui harus dicari terlebih dahulu di lemari kabinet. Belum lagi permasalahan lain seperti keamanan selain rentan terhadap api, lapuk, juga rentan terhadap rayap. Oleh karena itu dibutuhkan solusi untuk mengatasi masalah masalah tersebut.

Teknologi berkembang dan tumbuh begitu pesat, tak heran di banyak tempat sudah di terapkan teknologi baik yang simple maupun teknologi yang kompleks, sebagai contoh teknologi web app, web app adalah aplikasi yang bisa diakses melalui mesin penjelajah (search engine) baik melalui internet atau intranet. pengembangan nya mudah dan tanpa harus di distribusikan maupun di install jika ingin menggunakan nya [1]. Dengan teknologi web app permasalahan yang sudah dipaparkan bisa terselesaikan dengan cara membangun sistem yang mampu melakukan pengarsipan secara digital sehingga data tidak lagi disimpan secara fisik 2 melainkan secara digital di dalam sistem web app yang akan di bangun, selain itu arsip dilengkapi dengan enkripsi sehingga data yang disimpan aman.

https://doi.org/10.56211/helloworld.v1i2.13

Berbicara mengenai enkripsi, algoritma AES (Advanced Encryption Standard) merupakan salah satu jenis algoritma yang digunakan untuk enkripsi, algoritma AES merupakan standar kriptografi pengganti DES (Data Encryption Standard). Algoritma AES diresmikan pada 2001 oleh NIST (National Institute of Standard and Technology) Amerika. Kelebihan dari algoritma AES dilihat dari segi jenis kunci yang simetris maka kecepatan operasi (komputasi) lebih tinggi bila dibandingkan dengan algoritma asimetrik sehingga dapat digunakan pada sistem real-time, AES mempunyai panjang kunci paling sedikit 128 bit, maka AES tahan terhadap serangan exhaustive key search. Dengan panjang kunci 128-bit, maka terdapat 2128 ≈ 3,4 x 1038 kemungkinan kunci [4].

TINJAUAN PUSTAKA

Arsip

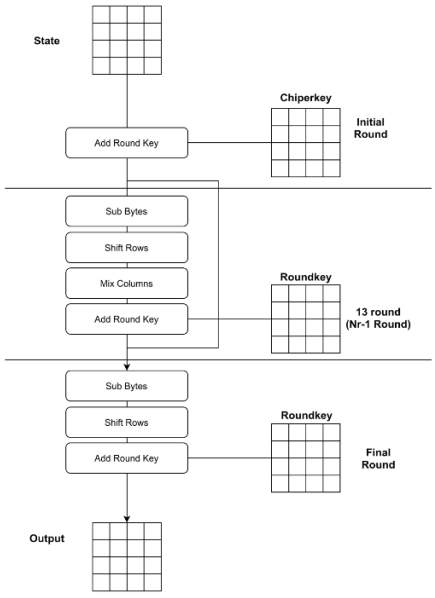
Arsip dapat diartikan pula sebagai suatu badan (*agency*) yang melakukan segala kegiatan pencatatan penanganan, penyimpanan, dan pemeliharaan surat-surat yang mempunyai arti penting baik ke dalam maupun ke luar, baik yang menyangkut soal-soal pemerintahan maupun non-pemerintahan dengan menerapkan kebijaksanaan dan sistem tertentu yang dapat dipertanggung jawabkan [2].

Algoritma AES-256

AES (*Advanced Encryption Standard*) merupakan algoritma cipher yang cukup aman untuk melindungi data atau informasi yang bersifat rahasia. Pada tahun 2001, AES 7 digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST (*National Institute of Standard and Technology*) sebagai pengganti algoritma DES (*Data Encryption Standard*) yang sudah berakhir masa penggunaannya. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit [5].

Proses Enkripsi AES 256

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, input yang telah salinkan ke dalam state akan mengalami transformasi byte AddRoundKey. Setelah itu, state akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns* [5]. Berikut gambaran simulasi dari proses enkripsi pada Algoritma AES.



Gambar 1. Proses enkripsi pada Algoritma Aes

AddRoundKey

Dalam *initial round*, transformasi *AddRoundKey*() dilakukan terhadap kunci utama. Sedangkan dalam 10 *round* yang lain, proses *AddRoundKey* dilakukan terhadap kunci putaran (*round key*). Proses *AddRoundKey* didefinisikan sebagai operasi XOR antara *array state* dengan *round key*. Operasi XOR dilakukan pada masing - masing *byte* dalam *array* sehingga menghasilkan nilai baru pada *array* hasil dengan ukuran *array* hasil sama dengan ukuran *array state* awal dan *array key*, yaitu sebesar 4x4. Hasil untuk masing-masing baris dan kolom pada *array state* hasil diperoleh dari hasil operasi XOR antara *array state* awal dengan *array key* untuk baris dan kolom yang sama.

SubBytes

Transformasi *SubBytes*() memetakan setiap *byte* dari *array state* dengan menggunakan tabel substitusi *S-Box*. Tabel *S-Box* dapat dilihat pada gambar berikut.

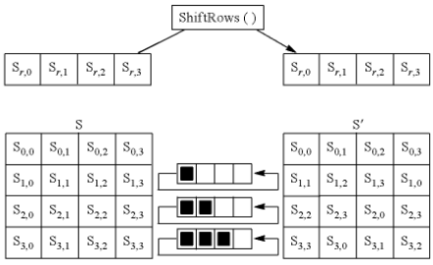
x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 2. *S-Box*

Cara pensubstitusian adalah sebagai berikut: untuk setiap *byte* pada *array*, misalkan S[r,c] = xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r,c], maka nilai substitusinya, yang dinyatakan dengan S'[r,c], adalah elemen di dalam *S-Box* yang merupakan perpotongan baris x dengan kolom y.

ShiftRows

Transformasi *ShiftRows*() melakukan pergeseran secara *wrapping* (*siklik*) pada 3 baris terakhir dari *array state*. Jumlah pergeseran bergantung pada nilai baris (r). Baris r = 1 digeser sejauh 1 *byte*, baris r = 2 digeser sejauh 2 *byte* dan baris r = 3 digeser sejauh 3 *byte*. Baris r = 0 tidak digeser.



Gambar 3. Ilustrasi *ShiftRows*

MixColumns

Transformasi MixColumns() dilakukan setelah transformasi ShiftRows, merupakan sumber utama dari difusi pada algoritma *AES*. Difusi merupakan prinsip yang menyebarkan pengaruh satu *bit plaintext* atau kunci ke sebanyak mungkin *ciphertext*. Transformasi *MixColumns*() mengalikan setiap kolom dari *array state* dengan *polinom* a(x) mod (x4 + 1). Setiap kolom diperlakukan sebagai *polinom* 4 suku pada GF (28). *Polinom* a(x) yang ditetapkan pada persamaan 1

$$a(x) = \{03\}x3 + \{01\}\ x2 + \{01\}\ x + \{02\} \quad (1)$$

Transformasi ini dinyatakan sebagai perkalian matriks seperti pada persamaan 2

$$s'(x) = a(x) \otimes s(x) \quad (2)$$

 https://doi.org/10.56211/helloworld.v1i2.13

Hasil dari perkalian matriks tersebut, setiap *byte* dalam kolom *array state* akan digantikan dengan nilai baru. Persamaan matematis untuk setiap *byte* tersebut pada persamaan 3

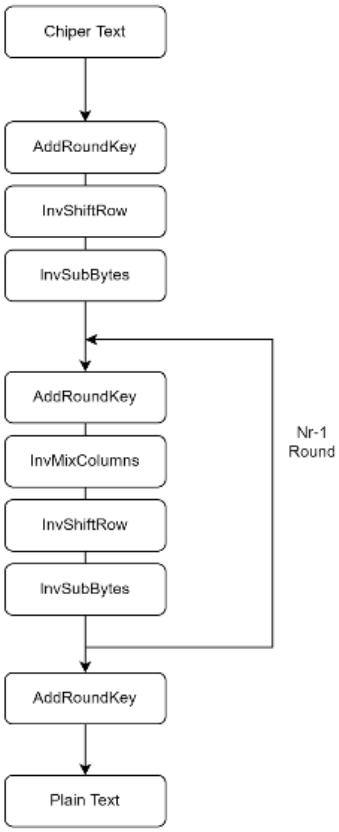
$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 4. Perkalian *Matrix*

$$\begin{aligned} s'0,c &= (\{02\} \bullet s0,c) \oplus (\{03\} \bullet s1,c) \oplus s2,c \oplus s3,c \\ s'1,c &= s0,c \oplus (\{02\} \bullet s1,c) \oplus (\{03\} \bullet s2,c) \oplus s3,c \\ s'2,c &= s0,c \oplus s1,c \oplus (\{02\} \bullet s2,c) \oplus (\{03\} \bullet s3,c) \\ s'3,c &= (\{03\} \bullet s0,c) \oplus s1,c \oplus s2,c \oplus (\{02\} \bullet s3,c) \end{aligned} \quad (3)$$

Proses Dekripsi AES 256

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma *AES*. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada skema berikut ini:

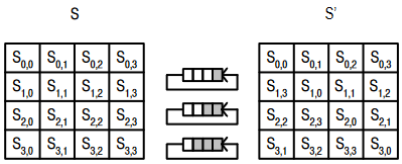


Gambar 5. Proses dekripsi pada Algoritma Aes

Berikut penjelasan mengenai gambar:

InvShiftRows

InvShiftRows adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri. Ilustrasi transformasi InvShiftRows terdapat pada gambar berikut:



Gambar 6. Ilustrasi *InvShiftRows*

InvSubBytes

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada *state* dipetakan dengan menggunakan tabel *Inverse S-Box*. Tabel *Inverse S-Box* akan ditunjukkan dalam gambar berikut.

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Gambar 7. *Inverse S-Box*

InvMixColumns

Setiap kolom dalam state dikalikan dengan matriks perkalian dalam *AES*. Perkalian dalam matriks dapat dituliskan :

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0B & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 8. Perkalian *Matrix Invercolum*

Hasil dari perkalian matriks tersebut, setiap *byte* dalam kolom *array state* akan digantikan dengan nilai baru. Persamaan matematis untuk setiap *byte* tersebut pada persamaan 4

$$\begin{aligned} s'0,c &= (\{0E\}\bullet s0,c) \oplus (\{0B\}\bullet s1,c) \oplus (\{0D\}\bullet s2,c) \oplus (\{09\}\bullet s3,c) \\ s'1,c &= (\{09\}\bullet s0,c) \oplus (\{0E\}\bullet s1,c) \oplus (\{0B\}\bullet s2,c) \oplus (\{0D\}\bullet s3,c) \\ s'2,c &= (\{0D\}\bullet s0,c) \oplus (\{09\}\bullet s1,c) \oplus (\{0E\}\bullet s2,c) \oplus (\{0B\}\bullet s3,c) \\ s'3,c &= (\{0B\}\bullet s0,c) \oplus (\{0D\}\bullet s1,c) \oplus (\{09\}\bullet s2,c) \oplus (\{0E\}\bullet s3,c) \end{aligned} \quad (4)$$

AddRoundKey

Proses transformasi *AddRoundKey* pada proses dekripsi merupakan transformasi yang bersifat *self-invers* dengan syarat menggunakan kunci yang sama.

Tools Perancangan

Dalam perancangan sistem, para pengembang sering menggunakan UML (Unified Modelling Language). UML adalah sekumpulan alat yang biasanya berupa diagram untuk merancang dan memodelkan bagaimana sistem tersebut bekerja, bagaimana pengguna dapat berinteraksi dengan sistem, bagaimana tata cara kerja dari sistem dan fitur-fitur yang terdapat di sebuah sistem yang nantinya akan diimplementasikan. Penggunaan *UML* bermanfaat memanajemen kompleksitas dari sistem, mendeteksi kesalahan yang mungkin terjadi ketika diimplementasikan, menjelaskan tata kerja dari sistem kepada para pihak yang berkepentingan [3].

 https://doi.org/10.56211/helloworld.v1i2.13

Teknologi Yang Digunakan

Dalam penelitian ini penulis menggunakan beberapa teknologi dan bahasa pemrograman yang digunakan dalam membangun aplikasi e-arsip antara lain sebagai berikut :

HTML

Hypertext Markup Language (HTML) adalah bahasa markah standar untuk dokumen yang dirancang untuk ditampilkan di peramban internet dan dibantu oleh teknologi seperti *Cascading Style Sheets* (CSS) dan bahasa scripting seperti JavaScript dan VBScript.

CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman.

PHP

Bahasa pemrograman PHP adalah bahasa pemrograman yang digunakan untuk membuat web yang server-side scripting. PHP digunakan untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. Namun PHP juga mendukung pengelolaan sistem database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya [6].

Codeigniter (CI)

Framework Codeigniter (CI) adalah *framework* atau kerangka kerja untuk megembangkan sebuah aplikasi dengan menggunakan **b**ahasa pemrograman PHP [7].

Bootstrap

Bootstrap merupakan sebuah *framework CSS* untuk membangun website yang menarik agar memudahkan pengembang. Sulit untuk mengembangkan dan pemeliharaannya jika tidak ada konsistensi terhadap aplikasi individual. *Bootstrap* memberikan solusi rapi dan seragam terhadap solusi yang umum, tugas *interface* yang setiap pengembang hadapi.

METODOLOGI

Metode Pengumpulan Data

Teknik pengumpulan data yang digunakan adalah sebagai berikut :

1. Observasi Penulis melakukan tinjauan langsung kelapangan melakukan analisa dan evaluasi data yang berhubungan dengan topik yang penulis angkat dalam penelitian ini.
2. Wawancara Penulis melakukan wawancara dengan pihak terkait agar mendapatkan data yang otentik dan valid sehingga mempermudah dalam meneliti topik yang penulis angkat.

Analisa Kebutuhan Sistem

Dalam membangun sistem yang baik tahap awal dalam pembuatan sistem harus didefinisikan secara terperinci kebutuhan dari aplikasi atau sistem yang dibangun, baik secara kebutuhan perangkat lunak pendukung, perangkat keras yang digunakan, serta logika yang diterapkan pada aplikasi yang akan di bangun. Adapun rincian nya sebagai berikut :

Software

1. Html
2. CSS
3. Framework Codeigniter
4. Bootsrap
5. Visual Studio Code

Hardware

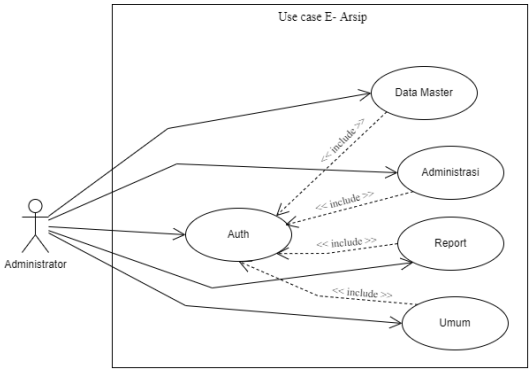
1. Device Asus seri x441u
2. Prosesor intel core i3
3. penyimpanan ssd 240
4. Ram 4Gb

Desain Sistem

Desain logic

Desain logic pada aplikasi digambarkan menggunakan UML (*Unified Modelling Language*) adapun penerapan nya sebagai berikut:

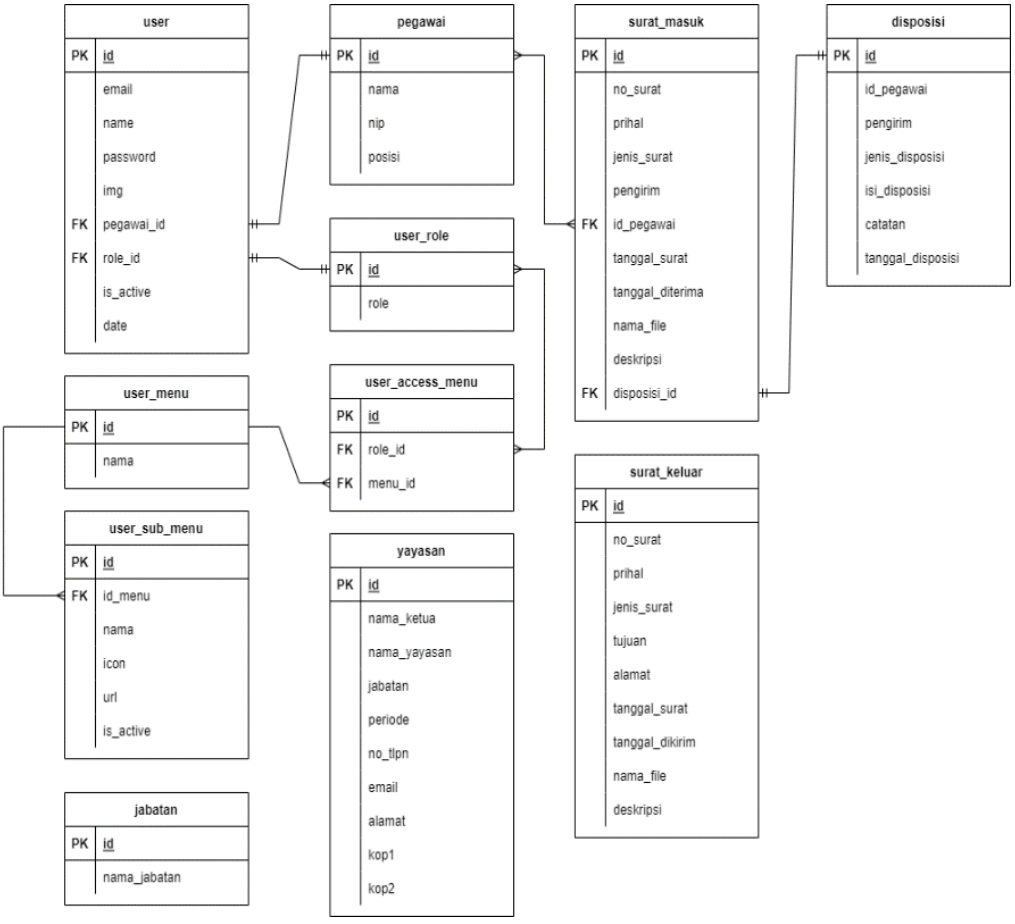
Use Case Diagram



Gambar 9. Use Case Diagram Administrator

Pada gambar 9 menggambarkan use case diagram untuk administrator, seorang administrator dapat mengakses semua fitur *menu* aplikasi mulai dari *menu* data master, *menu* administrasi, menu *report* dan menu umum. Administrator memiliki otoritas penuh pada aplikasi e-arsip ini, sehingga memiliki wewenang dalam menambang user pengguna aplikasi, hal ini bertujuan agar aplikasi tidak di salah gunakan.

Desain *Database*

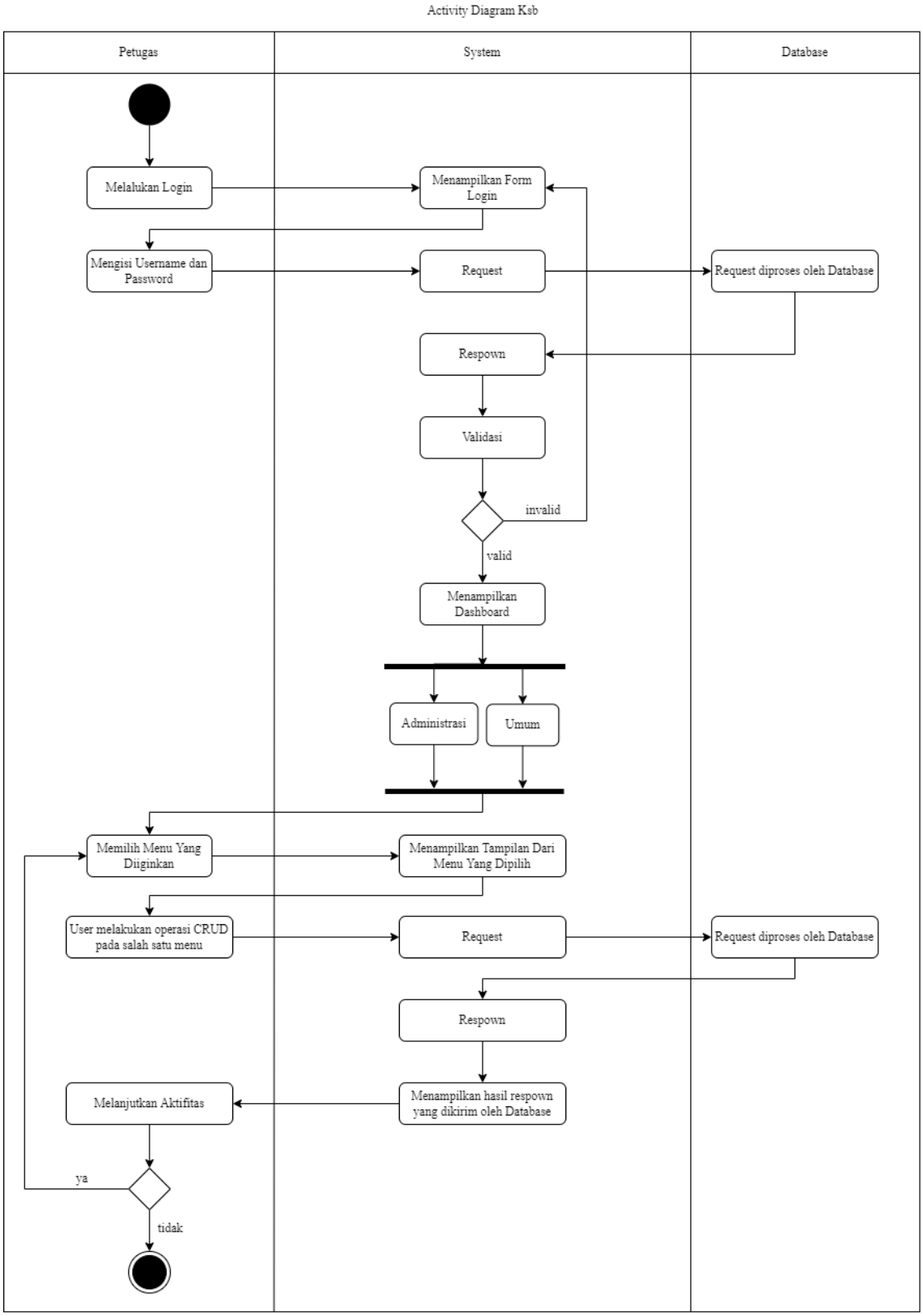


Gambar 10. Desain *Database* Aplikasi

 https://doi.org/10.56211/helloworld.v1i2.13

Gambar 10 merupakan struktur dari database yang akan digunakan pada aplikasi, terdapat 11 tabel antaralain tabel *user* yang digunakan untuk menyimpan data *user*, tabel pegawai yang digunakan untuk menyimpan data pegawai, tabel surat masuk dan surat keluar untuk menyimpan data surat, tabel *user menu*, tabel *user sub menu*, tabel *user access menu*, table *user role* digunakan untuk mengatur *menu management user* sehingga dapat mengatur *menu* apa saja yang akan digunakan oleh setiap tingkatan user, tabel jabatan, tabel yayasan yang akan digunakan untuk menampung data jabatan dan *profile* yayasan dan yang terakhir tabel disposisi diperuntukkan untuk menyimpan data surat yang mengalami disposisi surat.

Activity Diagram

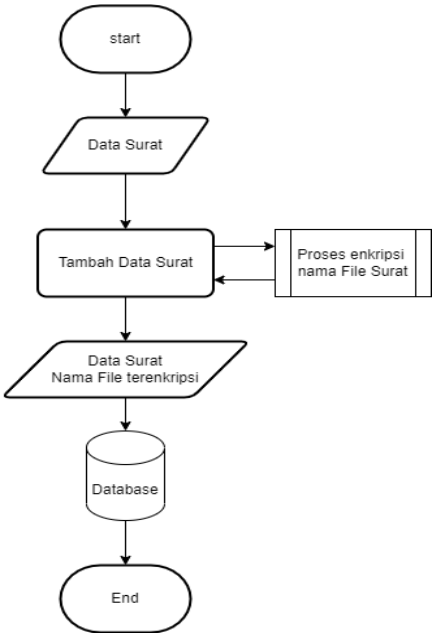


Gambar 11. Desain *Database* Aplikasi

Gambar 11 merupakan activity diagram dari *user* administrator, menu yang dapat diakses, serta proses yang dialami oleh administrator ketika aplikasi dijalankan.

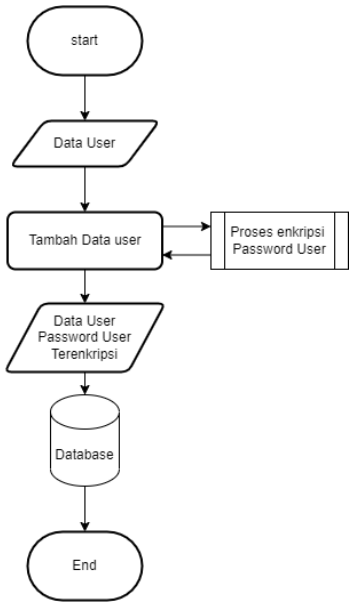
Algoritma AES-256 Pada Sistem

Algoritma *AES*-256 akan diimplementasikan pada bagian tambah data surat, baik surat masuk dan surat keluar, adapun yang akan diamankan ialah nama *file* dari *file* surat yang akan disimpan kedalam *database*. Begitu juga pada bagian tambah data *user*, pada *password user* akan dilakukan enkripsi, sehingga *password user* aman di simpan ke *database* dan terhindar dari *sql injection.* Berikut diagram alir dari proses enkripsi tambah data surat dan tambah data *user*:



Gambar 12. *Flowchart* Enkripsi Data Surat

Gambar 12 merupakan gambaran alur dari data surat yang ditambahkan oleh *user* kemudian dienkripsi lalu disimpan kedalam *database*.



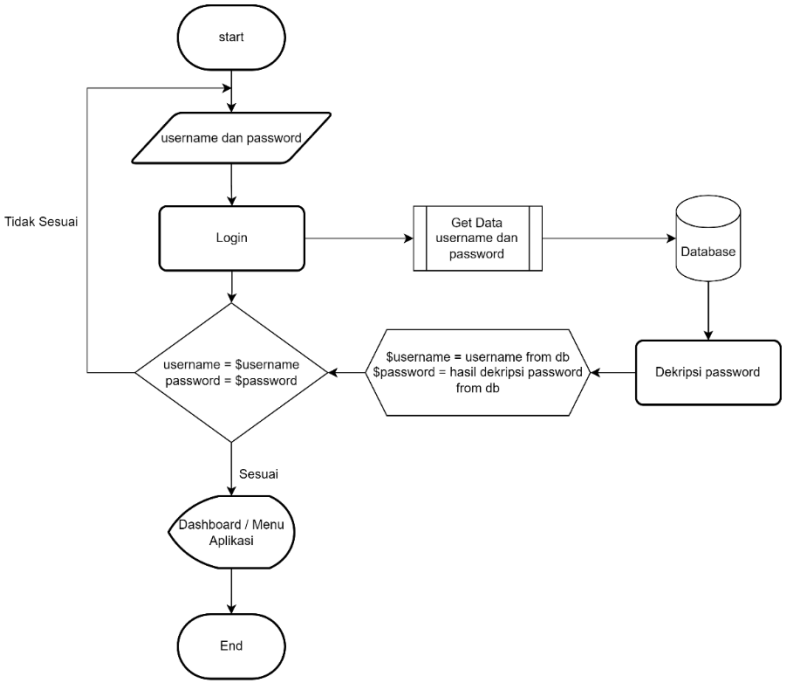
Gambar 13. *Flowchart* Enkripsi Data User

Gambar 13 merupakan alur dari data *user* yang ditambahkan oleh admin kemudian dienkripsi lalu disimpan kedalam

 https://doi.org/10.56211/helloworld.v1i2.13

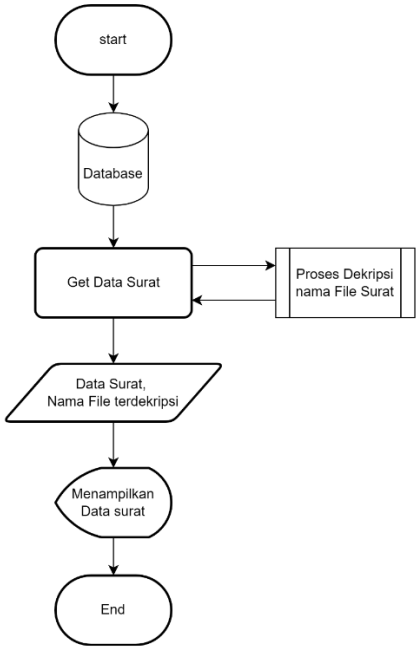
database.

Selain alur dari proses enkripsi data surat dan data *user*, juga terdapat alur proses dekripsi data surat dan data *user*. Dimana proses dekripsi sendiri terjadi ketika data surat ditampilkan ketika *user* mengakses menu data administrasi. Begitu juga ketika *user* melakukan proses *login* ke dalam aplikasi. Ketika *user* melakukan *login*, *username* dan *password* yang di *input user* akan dicocokan dengan data user yang terdapat di dalam *database*, ketika *username* dan *password* di *query* dari *database*, *password* yang terenkripsi akan didekripsi terlebih dahulu sehingga *password* yang di *query* dari *database* dan *password* yang di *input* oleh *user* akan di cocok kan. Jika sama maka *user* dapat masuk kedalam aplikasi. Agar lebih jelas berikut *flowchart* dari alur dekripsi data *password user* dan data *surat*.



Gambar 14. *Flowchart* Dekripsi *Password User*

Gambar 14 merupukan alur dari proses *login* dan proses *password* di dekripsi dari *database.*



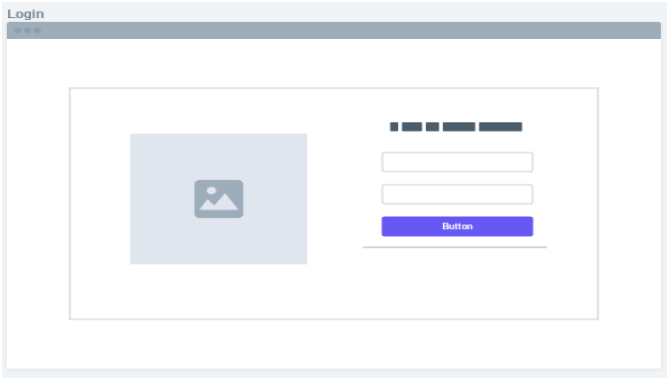
Gambar 15. *Flowchart* Dekripsi Data Surat

Gambar 15 alur dari proses dekripsi data surat kemudian ditampilkan ke *interface* ketika *menu* administrasi di akses.

Desain User Interface

Tahap ini akan membuat *prototype* dari desain antar muka yang akan di bangun pada aplikasi sehingga memudahkan dalam merealisasikan nya ke dalam baris kode program. *Interface* merupakan tampilan yang dilihat oleh user ketika mengakses aplikasi. *Interface* yang sederhana dapat membantu *user* dalam melakukan aktivitas di dalam aplikasi, adapun *interface* yang akan di bangun didalam aplikasi ini antara lain sebagai berikut:

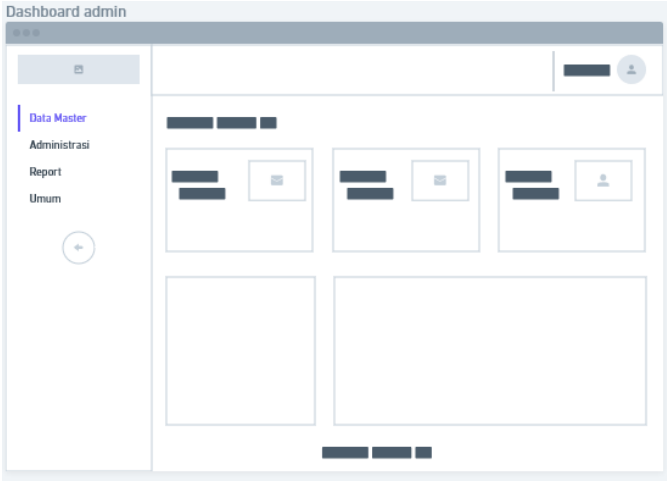
Tampilan *login*



Gambar 16. *Mockup* Tampilan *Login*

Tampilan *login* merupakan halaman pertama yang dapat diakses oleh tingkatan *user* admin, *user* petugas dan *user* ksb. Dalam tampilan *login* terdapat beberapa komponen antara lain *form* untuk memasukan *username* dan *password*, tombol *login*, gambar dan beberapa teks untuk melengkapi tampilan login.

Tampilan *Dashboard*

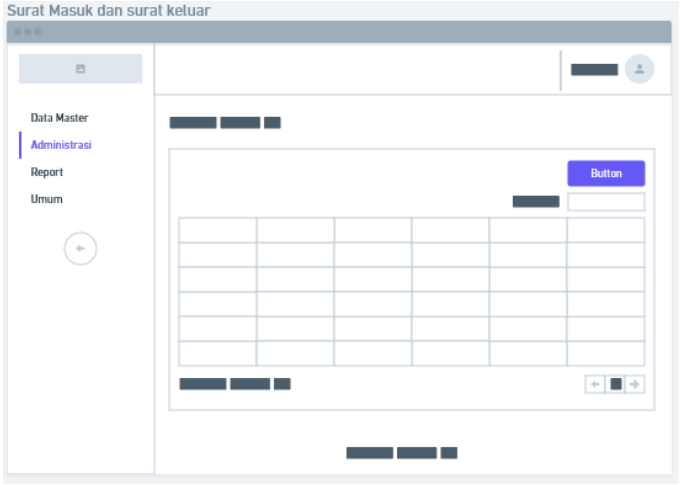


Gambar 17. *Mockup* Tampilan *Dashboard* Admin

Tampilan *dashboard* merupakan tampilan yang diakses oleh *admin* setelah berhasil melakukan login dan di verifikasi oleh sistem. Tampilan *dashboard* akan menampilkan data mengenai jumlah surat masuk, surat keluar dan jumlah user yang terdaftar di aplikasi. Tampilan *dashboard* terdiri dari beberapa komponen seperti komponen *card* yang akan menampilkan jumlah data, komponen *teks* untuk melengkapi komponen *card*.

 https://doi.org/10.56211/helloworld.v1i2.13

Tampilan Surat Masuk dan Surat Keluar



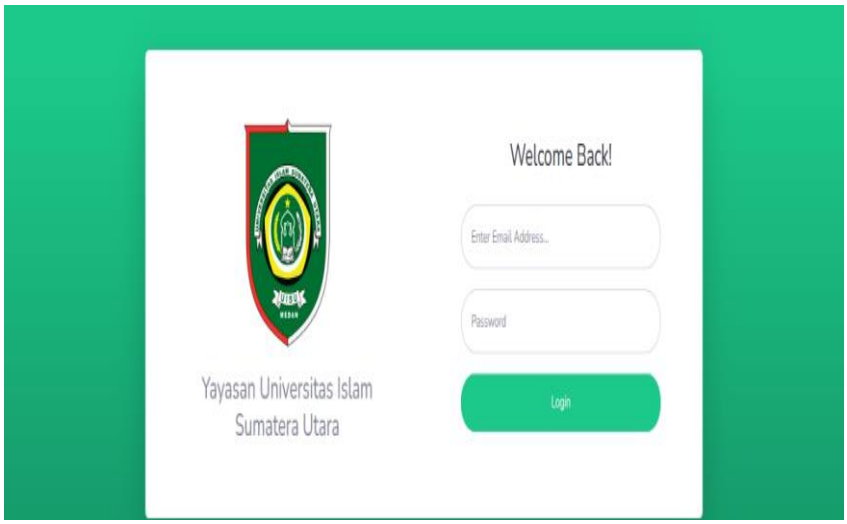
Gambar 18. *Mockup* Tampilan Surat Masuk dan Surat Keluar

Tampilan surat masuk dan surat keluar secara mockup dan kegunaan sama yaitu untuk menampilkan data surat masuk dan surat keluar yang di simpan di dalam data base. Secara fitur dapat melakukan *insert*, *preview*, *delete* dan *download* data surat. Komponen yang digunakan pada tampilan data surat masuk dan surat keluar antara lain komponen tombol, komponen teks, komponen *card*, komponen tabel untuk menampung data yang akan ditampilkan ke *user*.

HASIL DAN PEMBAHASAN

Setelah melakukan perancangan terhadap aplikasi yang akan dibangun, serta melakukan analisis kebutuhan dari aplikasi yang akan dibangun, desain logic yang akan diterapkan di dalam aplikasi serta desain interface yang akan diterapkan ke dalam aplikasi berikut hasil implementasi:

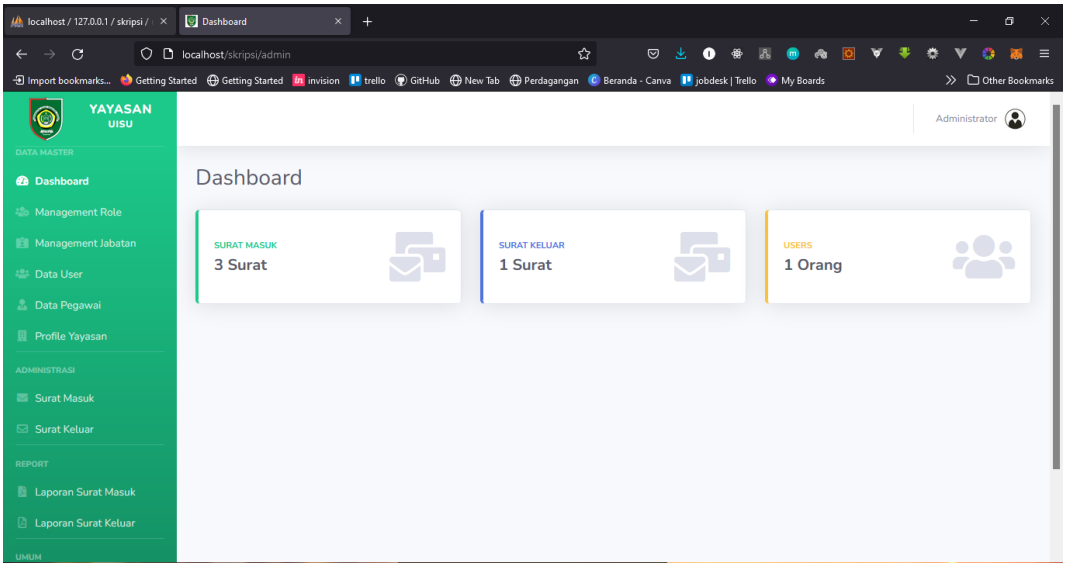
Tampilan *Login*
Tampilan *login* merupakan tampilan awal yang dapat diakses oleh *user* terdiri dari *form email* dan *form password*. *User* melakukan *login* sesuai dengan *email* dan *password* yang terdaftar di aplikasi. Berikut ini merupakan tampilan *login*.



Gambar 19 Tampilan *Login*

Tampilan *Dashboard*

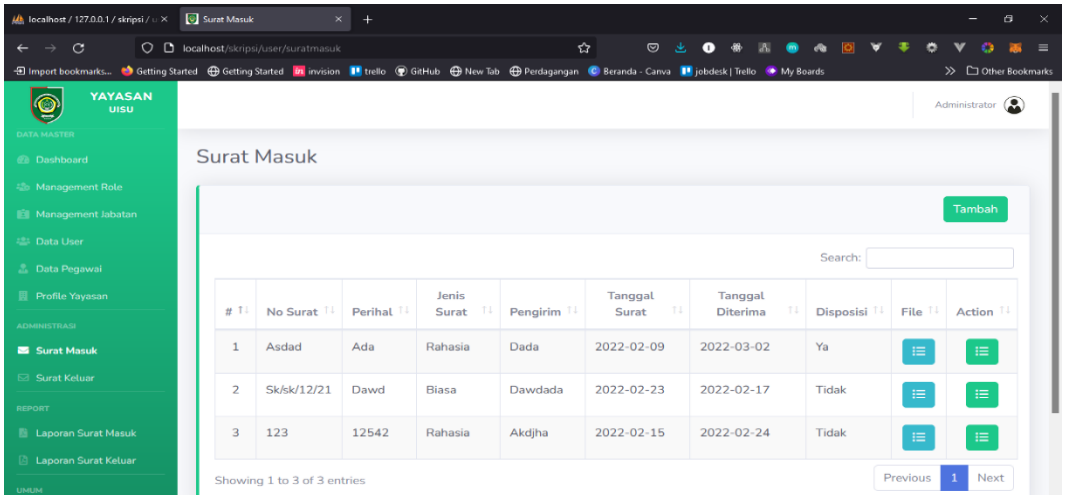
Setelah *user* melakukan *login* khusus untuk *user* tingkatan *admin* akan diarahkan ke halaman *dashboard* sedangkan *user* tingkatan petugas maupun ksb akan di arahkan ke tampilan surat masuk. Berikut merupakan tampilan dari halaman *dashboard*.



Gambar 20. Tampilan *Dashboard*

Tampilan Surat Masuk

Pada tampilan ini *user* dapat melihat data surat masuk, terdapat beberapa fitur diantaranya fitur *preview* dan fitur *download*, kedua fitur ini dapat diakses melalui tombol pada kolom *file* pada tabel.



Gambar 21. Tampilan Data Surat Masuk

KESIMPULAN DAN SARAN

Aplikasi dibangun menggunakan Framework Codeigniter yang menggunakan bahasa pemrograman php sebagai basis bahasa pemrograman dari Framework Codeigniter, Aplikasi E arsip dibangun menggunakan model MVC yaitu Model, View dan Controller, Enkripsi Aes-256 dilakukan pada bagian surat baik surat masuk maupun surat keluar dan data user, adapun yang dienkripsi adalah nama dari file surat dan password dari user. Menambahkan beberapa fitur yang lain nya agar aplikasi lebih bagus dan lebih kompleks sehingga dapat memenuhi kebutuhan di yayasan universitas islam sumatera utara.

 https://doi.org/10.56211/helloworld.v1i2.13

UCAPAN TERIMA KASIH

Penulis mengucapkan Terima Kasih kepada Keluarga, Dosen, serta Orang-orang terdekat yang telah memberi dukungan penuh dan motivasi kepada penulis dalam menyelesaikan skripsi ini untuk meraih gelar sarjana Strata-1 di Universitas Islam Sumatera Utara.

DAFTAR PUSTAKA

[1] Destiningrum M, Adrian Q. "Sistem Informasi Penjadwalan Dokter Berbassis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre)". Jurnal Teknoinfo, Vol. 11, No. 2, 2017, Pages 30-37, ISSN 1693 001.

[2] Fathurrahman M. 2018. "Pentingnya Arsip Sebagai Sumber Informasi". Jipi (Jurnal Ilmu Perpustakaan Dan Informasi Vol. 3 No. 2.

[3] Irsyad S, Sitio A. "Penerapan Konsep Mvc Pada Sistem Penjualan Online Dengan Sistem Keamanan Menggunakan Algoritma Rijndeal". Jurnal Informatika, Manajemen Dan Komputer Vol 11 No 2, 2019, e-ISSN 2580-3042.

[4] Muharram F. "Analisis Algoritma Pada Proses Enkripsi Dan Dekripsi File Menggunakan Advanced Encryption Standard". Prosiding Seminar Nasional Ilmu Komputer Dan Teknlogi Informatika Vol 3 No 2, 2018, e-ISSN 2540-7902.

[5] Nahado M, Mei Hellyana C, Faqih H. "Perancangan Sistem Pakar Pendeteksi Error Bahasa Pemrograman Php Berbasis Web". Konferensi Nasional Ilmu Sosial & Teknologi (KNiST), Vol 1 No 1, 2016, ISBN: 978-602-61242-4-1.

[6] Rasuliano Laberto Kelen Y. "Implementasi Model-View-Controller (Mvc) Pada Ujian Online Melalui Penerapan Framework Codeigniter". Jurnal Pendidikan Teknologi Informasi (Jukanti), Vol 1 No 1, 2018, https://doi.org/10.37792/jukanti.v1i1.5.

BIODATA PENULIS

Ibnu Hajar

Lahir di Perbaungan tanggal 10 Maret 1998 menyelesaikan pendidikan di Universitas Islam Sumatera Utara, Fakultas Teknik Program studi teknik informatika (tamat tahun 2022). Menekuni pemrograman pada bagian Front-End serta pada bidang Design Grafis dan UI/UX.