



TUGAS AKHIR - IF184802

Implementasi Perangkat Lunak Verifikasi Integritas File Terenkripsi pada Server Cloud

MUHAMMAD FAJRI SALAM
NRP 05111540000099

Dosen Pembimbing I
Ir. MUCHAMMAD HUSNI, M.Kom.

Dosen Pembimbing II
HENNING TITI CIPTANINGTYAS, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember



TUGAS AKHIR - IF184802

Implementasi Perangkat Lunak Verifikasi Integritas File Terenkripsi pada Server Cloud

**MUHAMMAD FAJRI SALAM
NRP 05111540000099**

**Dosen Pembimbing I
Ir. MUCHAMMAD HUSNI, M.Kom.**

**Dosen Pembimbing II
HENNING TITI CIPTANINGTYAS, S.Kom., M.Kom.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

(Halaman ini sengaja dikosongkan)



UNDERGRADUATE THESIS - IF184802

Software Implementation Verification of Encrypted File Integrity on Cloud Servers

MUHAMMAD FAJRI SALAM
NRP 05111540000099

First Advisor
Ir. MUCHAMMAD HUSNI, M.Kom.

Second Advisor
HENNING TITI CIPTANINGTYAS, S.Kom., M.Kom.

INFORMATICS DEPARTMENT
Faculty of Information Communication and Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

Implementasi Perangkat Lunak Verifikasi Integritas File Terenkripsi pada Server Cloud

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur Jaringan Komputer
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD FAJRI SALAM
NRP: 05111540000099

Disetujui oleh Pembimbing tugas akhir:

1. Ir. MUCHAMMAD HUSNI, M.Kom.
(NIP. 19600221 198403 1 001) (Pembimbing 1)

2. HENNING TITI CIPTANINGTYAS,
S.Kom., M.Kom.
(NIP. 19840708 201012 2 004) (Pembimbing 2)

SURABAYA
JULI, 2019

(Halaman ini sengaja dikosongkan)

Implementasi Perangkat Lunak Verifikasi Integritas File Terenkripsi pada Server Cloud

Nama Mahasiswa : MUHAMMAD FAJRI SALAM
NRP : 05111540000099
Departemen : Informatika FTIK ITS
**Dosen Pembimbing 1 : Ir. MUCHAMMAD HUSNI,
M.Kom.**
**Dosen Pembimbing 2 : HENNING TITI
CIPTANINGTYAS, S.Kom.,
M.Kom.**

Abstrak

Komputasi awan adalah teknologi populer yang memungkinkan untuk mengakses data melalui Internet yang bahkan dapat menyimpan data sebagai pengganti penyimpanan lokal. Cloud memungkinkan pengguna untuk menyimpan data mereka di cloud tanpa perlu khawatir akan keakuratan dan keandalannya. Namun menyimpan data di cloud menimbulkan tantangan keamanan tertentu. Mengalihkan data di cloud menyebabkan pemilik data kehilangan kontrol fisik atas data mereka. Penyedia Layanan Cloud tertentu dapat beroperasi secara tidak jujur dengan data pengguna cloud, mereka dapat mengakses data dari cloud dan menjualnya kepada pihak ketiga untuk mendapatkan keuntungan. Jadi, meskipun *outsourcing* data di cloud tidak mahal dan mengurangi kompleksitas penyimpanan dan pemeliharaan berdurasi lama, setidaknya ada jaminan integritas data, privasi, keamanan, dan ketersediaan di server cloud.

Tugas Akhir ini akan berfokus pada strategi verifikasi integritas untuk data *outsourcing*. Skema yang diusulkan adalah menggabungkan mekanisme enkripsi dengan verifikasi integritas. Skema enkripsi yang digunakan di sini adalah algoritma kriptografi AES-256 dan fungsi *hash* SHA-256 yang digunakan untuk memastikan kebenaran penyimpanan data pada server yang tidak terpercaya [1].

Kata kunci : Cloud Computing, AES-256, SHA-256, TPA.

Software Implementation Verification of Encrypted File Integrity on Cloud Servers

Student's Name : MUHAMMAD FAJRI SALAM
Student's ID : 05111540000099
Department : Informatika FTIK-ITS
First Advisor : Ir. MUCHAMMAD HUSNI, M.Kom
Second Advisor : HENNING TITI
CIPTANINGTYAS, S.Kom.,
M.Kom.

Abstract

Cloud computing is a popular technology which permits storing and accessing data over Internet instead of storing it on local machine's hard drive. Cloud enables users to store their data on cloud without fearing about its accuracy and reliability. However storing data on cloud imposes certain security challenges. Outsourcing data in cloud result in data owners losing physical control over their data. Certain Cloud Service Providers may operate dishonestly with the cloud users' data, they may sneak the data from cloud and sell it to third parties in order to earn profit. Thus even though outsourcing data on cloud is inexpensive and reduces long duration storage and maintenance complexity, there is least assurance of data integrity, privacy, security and availability on cloud servers. A number of solutions have been recommended to solve the security issues in cloud.

This undergraduate thesis focuses on the integrity verification strategy for outsourced data. The proposed scheme combines the encrypting mechanism along with integrity verification strategy. The encrypting scheme used here is cryptographic algorithm like AES-256 and SHA-256 hash function is employed for ensuring data storage correctness on untrusted server

Kata kunci : Cloud Computing, AES-256, SHA-256, TPA.

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Puji syukur kepada Allah Yang Maha Esa atas segala karunia dan rahmat-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul

“IMPLEMENTASI PERANGKAT LUNAK VERIFIKASI INTEGRITAS FILE TERENKRIPSI PADA SERVER CLOUD”.

Harapan dari penulis, semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini dan ke depannya, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT. dan Nabi Muhammad SAW. yang telah membimbing penulis selama hidup.
2. Keluarga penulis (Ayah, Ibu, Mas Dicky, Dina, Aldan, dan keluarga penulis yang lain) yang selalu memberikan dukungan baik berupa doa, moral, dan material yang tak terhingga kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini.
3. Bapak Ir. Muchammad Husni, M.Kom dan Ibu Henning Titi Ciptaningtyas, S.Kom., M.Kom selaku Dosen Pembimbing penulis yang telah membimbing, memberikan nasihat, dan memotivasi penulis sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Bapak Dr.Eng. Darlis Herumurti, S.Kom, M.Kom. selaku kepala Departemen Informatika ITS.
5. Bapak dan Ibu Dosen yang telah memberikan ilmunya selama penulis berkuliah di Informatika ITS.
6. Teman-teman Developer Arya-Fajar Production (Tayar, Fuad, Awan, Irsa, Tamtam, Andhika, Fawwaz, Fasma, Jonathan, Adit

dan Nila) yang selalu membantu memberikan solusi dan memberikan hiburan dikala penulis mendapatkan tekanan proyek yang tidak masuk akal di tengah kesibukan kegiatan perkuliahan.

7. Teman-teman Developer PPDB Riau 2019 (Fuad, Didin, Yoza dan Sulton) yang menemani penulis selama 10 hari untuk melancarkan PPDB Riau 2019 di Pekanbaru, Riau.
8. Teman-teman dari keluarga kontrakan Toko Bu Firda (Imam, Feliq, Muad, Yoga, Yanto, Ari dan Wowos) yang telah tinggal seataap bersama selama tiga tahun, selalu menemani kehidupan diluar kegiatan kuliah baik susah maupun senang.
9. Teman-teman pejuang SW 120 yang selalu memberikan informasi penting dan semangat kepada penulis untuk menyelesaikan tugas akhir.
10. Teman-teman angkatan 2015 yang sudah menjadi saksi hidup perjalanan karir penulis selama berkuliah di Informatika ITS.
11. Untuk orang-orang yang tidak dapat disebutkan satu persatu oleh penulis dan pembaca buku tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini. Namun, penulis memohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya. Tetap semangat dalam menjalani kehidupan, jangan menyerah, karena Allah masih ingin melihat kita berjuang. Semoga kita semua selalu diberi kebahagiaan lahir dan batin dan kesuksesan dunia akhirat. Aamiin.

Surabaya, 7 Juni 2019

Muhammad Fajri Salam

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak.....	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER.....	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Permasalahan	2
1.4 Tujuan	2
1.5 Manfaat	2
1.6 Metodologi.....	2
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur	3
1.6.3 Perancangan Sistem	3
1.6.4 Implementasi Sistem	3
1.6.5 Pengujian.....	3
1.6.6 Penyusunan Buku.....	4
1.7 Sistematika Penulisan Laporan	4
BAB II TINJAUAN PUSTAKA	7
2.1 Verifikasi File	7
2.2 Laravel.....	8
2.3 MySQL.....	11
2.4 Python.....	12
2.4.1 Platform Pemrograman Python	12
2.4.2 Lisensi Python	12
2.4.3 Fitur Bahasa Pemrograman Python	12
2.5 Advanced Encryption Standard (AES) 256	13

2.5.1	Proses Enkripsi Advanced Encryption Standard ..15	15
2.5.2	Proses Deskripsi Advanced Encryption Standard	18
2.5.3	Proses Ekspansi Kunci.....	20
2.6	Secure Hash Algorithm 256 (SHA-256)	21
2.6.1	Awal Perkembangan SHA-256	22
2.6.2	Dasar Prinsip	22
2.6.3	Cara Kerja	22
BAB III	PERANCANGAN.....	27
3.1	Deskripsi Umum	27
3.2	Rancangan Sistem	28
3.3	Server Aplikasi.....	28
3.3.1	Rancangan Spesifikasi Kebutuhan Aplikasi.....	28
3.3.2	Rancangan Mekanisme Pengunggahan File	30
3.3.3	Rancangan Mekanisme Pengunduhan File.....	32
3.3.4	Rancangan Verifikasi Integritas File	34
3.3.5	Rancangan Antarmuka	35
3.4	Rancangan Server Backup.....	42
3.5	Rancangan Server Basis Data	43
BAB IV	IMPLEMENTASI.....	45
4.1	Lingkungan Pembangunan Sistem	45
4.2	Implementasi Server Aplikasi.....	45
4.2.1	Implementasi Pengunggahan File	46
4.2.2	Implementasi Pengunduhan File	46
4.2.3	Implementasi Verifikasi Integritas File	47
4.2.4	Implementasi Antarmuka Pengguna.....	47
4.3	Implementasi Server Backup	52
4.4	Implementasi Server Basis Data	53
BAB V	UJI COBA DAN EVALUASI	55
5.1	Lingkungan Uji Coba	55
5.2	Skenario Uji Coba	55
5.2.1	Skenario Uji Fungsionalitas.....	56
5.2.2	Skenario Uji Performa	58
5.3	Hasil Uji Coba dan Evaluasi	58

5.3.1 Uji Fungsionalitas	59
5.3.2 Uji Performa.....	61
BAB VI KESIMPULAN DAN SARAN.....	65
6.1 Kesimpulan.....	65
6.2 Saran.....	65
DAFTAR PUSTAKA.....	67
LAMPIRAN.....	69
BIODATA PENULIS.....	79

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Laravel.....	10
Gambar 2.2 Proses Input Bytes, State Array, dan Output Bytes [8]	14
Gambar 2.3 Ilustrasi Proses Enkripsi AES [8]	15
Gambar 2.4 Pengaruh Pemetaan pada Setiap Byte [8].....	17
Gambar 2.5 Transformasi ShiftRows [8]	17
Gambar 2.6 Ilustrasi Proses Deskripsi AES [8].....	18
Gambar 2.7 Transformasi InvShiftRows [8]	19
Gambar 2.8 Pseudocode Ekspansi Kunci [8]	21
Gambar 2.9 Nilai Awal pada Variabel H	23
Gambar 3.10 Arsitektur Sistem	28
Gambar 3.11 Kasus Penggunaan	29
Gambar 3.12 Ilustrasi Mekanisme Unggah File	31
Gambar 3.13 Alur Pengunggahan File	33
Gambar 3.14 Alur Pengunduhan File	34
Gambar 3.15 Hubungan Antara Server Aplikasi dengan Server Basis Data.....	35
Gambar 3.16 Alur Verifikasi File	36
Gambar 3.17 Rancangan Antarmuka Masuk ke Sistem.....	37
Gambar 3.18 Rancangan Antarmuka Registrasi.....	37
Gambar 3.19 Rancangan Antarmuka Unggah File	38
Gambar 3.20 Rancangan Antarmuka Melihat File Saya.....	39
Gambar 3.21 Rancangan Antarmuka Melihat File Dibagikan ke Saya.....	39
Gambar 3.22 Rancangan Antarmuka Bagikan File 1	40
Gambar 3.23 Rancangan Antarmuka Bagikan File 2.....	40
Gambar 3.24 Rancangan Antarmuka Bagikan File 3.....	41
Gambar 3.25 Rancangan Antarmuka Melihat Log Aktivitas	41
Gambar 3.26 Rancangan Antarmuka Melihat Log Berbagi File	42
Gambar 3.27 Alur Pengembalian File.....	43
Gambar 3.28 Alur Pencadangan Basis Data.....	44
Gambar 4.29 Halaman Login	48
Gambar 4.30 Halaman Registrasi	49

Gambar 4.31 Halaman Unggah File	49
Gambar 4.32 Halaman File Saya	50
Gambar 4.33 Halaman Berbagi File 1	50
Gambar 4.34 Halaman Berbagi File 2	51
Gambar 4.35 Halaman Berbagi File 3	51
Gambar 4.36 Halaman Melihat Log	52
Gambar 4.37 Halaman Melihat Log Berbagi File	52
Gambar 5.38 Performa Kriptografi pada File Gambar	62
Gambar 5.39 Performa Kriptografi pada File Audio	63
Gambar 5.40 Performa Kriptografi pada File Video	64
Gambar 5.41 Performa Kriptografi pada File Dokumen	64

DAFTAR TABEL

Tabel 2.1 Perbandingan Jumlah Round dan Key [8]	14
Tabel 2.2 S-Box SubBytes	16
Tabel 2.3 Tabel Inverse S-Box [8].....	20
Tabel 3.4 Kasus Penggunaan.....	29
Tabel 5.5 Spesifikasi Masing-Masing Server.....	55
Tabel 5.6 Skenario Uji Fungsionalitas Pengguna Mengunggah File	56
Tabel 5.7 Skenario Fungsionalitas Pengguna Mengunduh File .	56
Tabel 5.8 Skenario Uji Fungsionalitas Verifikasi File.....	57
Tabel 5.9 Skenario Uji Fungsionalitas Pengembalian File	57
Tabel 5.10 Skenario Uji Fungsionalitas	58
Tabel 5.11 Jenis File yang Digunakan untuk Skenario Uji Performa	58
Tabel 5.12 Hasil Uji Fungsionalitas Pengguna Mengunggah File	59
Tabel 5.13 Hasil Uji Fungsionalitas Pengguna Mengunduh File	59
Tabel 5.14 Hasil Uji Fungsionalitas Verifikasi File	60
Tabel 5.15 Hasil Uji Fungsionalitas Pengembalian File	60
Tabel 5.16 Hasil Uji Fungsionalitas Pencadangan Basis Data ...	61

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 <i>Pseudocode</i> Pengunggahan File	46
Kode Sumber 4.2 <i>Pseudocode</i> Pengunduhan File	47
Kode Sumber 4.3 <i>Pseudocode</i> Verifikasi File.....	47
Kode Sumber 4.4 <i>Pseudocode Script</i> Pengembalian File	53
Kode Sumber 4.5 <i>Pseudocode</i> Pengiriman File Backup ke Server Backup	53

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan jaman, teknologi saat ini mengalami perkembangan kearah pencapaian kemudahan dan kenyamanan yang luar biasa, sehingga kegiatan sehari-hari yang di anggap tidak mungkin di kerjakan dalam waktu yang singkat menjadi mungkin untuk dilakukan secara singkat. Pengembangan teknologi komputasi berbasis internet saat ini lebih di arahkan pada proses aplikasi sistem yang mudah dan tidak memerlukan banyak waktu dan tenaga [1].

Perkembangan teknologi komputasi berbasis internet ini salah satunya dikenal sebagai *Cloud Computing*. Ada banyak jenis implementasi dari *Cloud Computing*, salah satunya adalah penyimpanan data secara *online*. Keunggulan menyimpan data secara *online* adalah pengguna dapat mengakses datanya dimanapun dan kapanpun dengan internet.

Perkembangan *Cloud Computing* ini tidak selalu berujung ke hal-hal yang positif. Ada juga penyedia layanan *Cloud Computing* yang melakukan penyalahgunaan data pengguna. Bentuk penyalahgunaan data ini berupa pencurian data, penjualan data dan perubahan data [1]. Maka dari itu diperlukan suatu sistem yang dapat melakukan verifikasi dan validasi terhadap data yang tersimpan di *Cloud* untuk menghindari penyalahgunaan data yang dilakukan oleh penyedia layanan *Cloud*.

Tujuan dari penelitian ini adalah untuk membuat sistem yang dapat melakukan verifikasi integritas file terenkripsi dengan menggunakan fungsi *hash* SHA-256, mengenkripsi setiap file yang tersimpan dengan menggunakan AES-256 dan melakukan pengembalian file apabila terdapat file yang hilang atau termodifikasi pada server *cloud*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara mengaplikasikan algoritma enkripsi AES-256 pada *framework* Laravel?
2. Bagaimana cara melakukan verifikasi integritas file untuk memastikan bahwa file yang tersimpan tidak mengalami modifikasi?
3. Bagaimana cara mencadangkan data dan mengembalikan data yang telah hilang atau termodifikasi pada server *cloud*?

1.3 Batasan Permasalahan

Berdasarkan masalah yang diuraikan oleh penulis, maka batasan masalah pada tugas akhir ini adalah:

1. Bahasa pemrograman yang digunakan adalah PHP dan Python sedangkan basis data yang digunakan adalah MySQL.
2. Lingkungan pengembangan yang digunakan menggunakan *framework* Laravel 5.8.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini antara lain:

1. Merancang aplikasi berbasis *website* dengan kerangka kerja Laravel yang mengaplikasikan algoritma enkripsi AES-256.
2. Membandingkan *hash key* pada file asli dengan file yang tersimpan di server *cloud* untuk memverifikasi integritas file.
3. Membangun sistem yang menggunakan tiga server yaitu server aplikasi, server basis data dan server backup.

1.5 Manfaat

Manfaat dari tugas akhir ini adalah terciptanya sistem penyimpanan data berbasis *website* yang memastikan akan keamanan data tanpa campur tangan penyedia layanan *Cloud*. Hasil tugas akhir ini diharapkan kedepannya dapat diterapkan sebagai aplikasi di lingkungan masyarakat umum

1.6 Metodologi

Pembuatan tugas akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi gambaran tentang tugas akhir yang akan dibuat. Pendahuluan proposal tugas akhir meliputi hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah yang menjadi konstrain dari tugas akhir, tujuan pembuatan tugas akhir, dan manfaat dari hasil tugas akhir. Di dalam proposal tugas akhir juga dijabarkan mengenai tinjauan pustaka yang menjadi referensi pendukung dalam pembuatan tugas akhir ini.

1.6.2 Studi Literatur

Pada studi literatur, akan dilakukan pengumpulan informasi dan referensi yang digunakan dalam pengerjaan tugas akhir yaitu mengenai Laravel, PHP, Python, MySQL, Algoritma AES-256 dan SHA-256

1.6.3 Perancangan Sistem

Pada tahap perancangan sistem, akan dilakukan perancangan bisnis proses dari sistem. Bisnis proses ini meliputi proses pendaftaran, mengunggah data, mengunduh data dan membagikan data kepada pengguna yang lain.

1.6.4 Implementasi Sistem

Pada tahap implementasi sistem akan dilakukan implementasi pembuatan sistem sesuai pada tahap perancangan sistem. Keluaran yang diharapkan dari tahap ini adalah Sistem PenyimpananData yang aman dan siap untuk dipakai.

1.6.5 Pengujian.

Pengujian dilakukan untuk mengetahui tingkat keberhasilan pada sistem yang dibangun serta untuk memeriksa apakah sistem sudah berjalan dengan baik dan dipastikan agar tidak ada kesalahan yang terjadi.

1.6.6 Penyusunan Buku

Pada tahap ini dilakukan penyusunan buku sebagai dokumentasi dari pelaksanaan tugas akhir yang mencakup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

1. Bab I. Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang, rumusan masalah, batasan permasalahan, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan tugas akhir.

2. Bab II. Tinjauan Pustaka

Bab ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan tugas akhir ini. Kajian teori yang dimaksud berisi tentang penjelasan singkat mengenai Laravel, PHP, Python, MySQL, Algoritma AES-256 dan Algoritma SHA-256.

3. Bab III. Perancangan

Bab ini berisi pembahasan mengenai perancangan proses bisnis yang akan diimplementasikan dalam tugas akhir. Perancangan proses bisnis berupa perancangan proses pendaftaran, mengunggah data, mengunduh data dan membagikan data.

4. Bab IV. Implementasi

Bab ini menjelaskan implementasi yang berbentuk kode sumber dari proses bisnis yang terjadi pada aplikasi mulai pendaftaran, mengunggah data, mengunduh data dan membagikan data.

5. Bab V. Pengujian dan Evaluasi

Bab ini berisi hasil pengujian dan evaluasi pada aplikasi Sistem Penyimpanan Data.

6. Bab VI. Kesimpulan dan Saran

Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan tugas akhir, dan saran untuk pengembangan tugas akhir ke depannya.

7. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam tugas akhir.

8. Lampiran

Dalam lampiran terdapat kode sumber program secara keseluruhan.

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar atau penjelasan dari metode dan alat yang digunakan dalam tugas akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan riset yang berkaitan.

2.1 Verifikasi File

Verifikasi file adalah suatu proses untuk memverifikasi integritas file di komputer. Hal ini dapat dilakukan dengan membandingkan tiap-tiap bagian dari file, tetapi membutuhkan dua Salinan dari file yang sama. Pendekatan yang lebih populer adalah menghasilkan *hash* dari file yang disalin dan membandingkannya dengan *hash* dari file asli [2].

Verifikasi file berbasis *hash* memastikan bahwa file tidak rusak dan termodifikasi dengan membandingkan nilai *hash* file dengan nilai yang dihitung sebelumnya. Jika nilai-nilai ini cocok, file dianggap tidak dimodifikasi. Karena sifat fungsi *hash*, ketidakcocokan dapat menyatakan bahwa file mengalami modifikasi.

Sering diinginkan untuk memverifikasi bahwa file belum dimodifikasi dalam pengiriman atau penyimpanan oleh pihak yang tidak dipercaya, misalnya untuk memasukkan kode berbahaya seperti virus. Untuk memverifikasi keaslian fungsi *hash* klasik tidaklah cukup karena tidak dirancang untuk menahan tabrakan [2]. Membuat tabrakan *hash* yang disengaja adalah suatu hal yang mudah bagi penyerang yang berarti bahwa perubahan berbahaya dalam file tidak terdeteksi oleh perbandingan *hash*. Dalam kriptografi, serangan ini disebut serangan *preimage* [3]. Oleh karena itu, fungsi *hash* kriptografi sering digunakan.

Penelitian ini menerapkan verifikasi file untuk memverifikasi integritas file pada server *cloud*.

2.2 Laravel

Laravel adalah sebuah framework berbahasa pemrograman PHP terbaik yang dikembangkan oleh *Taylor Otwell*. Proyek pembuatan framework ini dimulai pada April 2011, yang didasari atas keresahan *Taylor Otwell* karena tidak adanya framework PHP yang *up to date* dengan versi PHP [4].

Taylor Otwell memilih bahasa pemrograman PHP untuk frameworknya dikarenakan bahasa ini merupakan bahasa pemrograman yang sangat populer dalam membangun sebuah CMS (Content Management System). Popularitas PHP dikarenakan beberapa kelebihan yang ditawarkan oleh bahasa ini:

- **Kesederhanaan**, bahasa PHP merupakan bahasa yang sederhana. User atau pengguna yang hanya sedikit tahu atau bahkan sama sekali tidak mengerti tentang pemrograman bisa dengan cepat belajar PHP. Selain itu PHP juga menyediakan fungsi *built-in* untuk menangani kebutuhan standar pembuatan web.
- **Banyaknya referensi**, bahasa PHP dipilih karena bahasa ini telah ada sejak lama yaitu tahun 1995 dan sudah memiliki komunitas yang sangat besar. Dengan komunitas yang besar ini tentu saja referensi tentang bahasa pemrograman PHP sudah sangat banyak.
- **Open source**, bahasa PHP merupakan bahasa *open source* yang dapat digunakan di berbagai sistem operasi seperti: Linux, Unix, Macintosh, dan Windows.
- **Banyaknya web server**, web server yang mendukung bahasa PHP dapat ditemukan dimana-mana mulai dari Apache, IIS, Lighttpd, hingga Xitami yang pengkonfigurasianya relative mudah [5].

Kelahiran Laravel

Dengan dukungan serta popularitas PHP serta MySQL pada Laravel, framework ini tentu saja juga mendapatkan popularitas yang serupa pula. Selain itu ada beberapa alasan lain mengapa

banyak orang memilih dan menggunakan framework Laravel daripada framework lain diantaranya:

- **Mudah dan Dokumentasinya Lengkap**

Platform Laravel menarik dan mudah digunakan. Seorang pengguna yang tidak ahli dalam bidang web developmentpun bisa menggunakannya. Dokumentasi resmi yang dimiliki Laravel pun tergolong ke dalam dokumentasi yang sangat baik, rapi, mudah dan jelas. Dokumentasi ini tersedia pada <https://laravel.com/docs>

- **Open source**

Laravel merupakan framework open source yang dapat digunakan secara bebas, gratis, dan memungkinkan pengguna untuk membuat web aplikasi yang besar dan kompleks dengan mudah.

- **Arsitektur MVC**

Dengan menggunakan pola MVC, kita dapat membuat arsitektur kode yang lebih rapi dimana pola tersebut memisahkan antara logika dan view. Arsitektur MVC dapat meningkatkan *performance* serta memiliki beberapa fungsi built-in. Arsitektur MVC pada Laravel dapat dilihat pada gambar 2.1.

- **Blade Template**

Laravel memiliki fitur blade template yang mempermudah pengguna untuk memetakan template yang dia miliki dengan membaginya menjadi beberapa bagian sehingga lebih mudah diatur.

- **Memiliki Fitur Migration**

Migration adalah salah satu fitur utama yang dimiliki Laravel. Dengan migration, memungkinkan pengguna untuk mempertahankan struktur database yang dia miliki tanpa membuatnya kembali. Dengan migration memungkinkan untuk mengatur database dengan menuliskan kode PHP.

- **Keamanan**

Keamanan aplikasi merupakan prioritas nomor satu dalam mengembangkan website. Terlebih lagi jika website tersebut

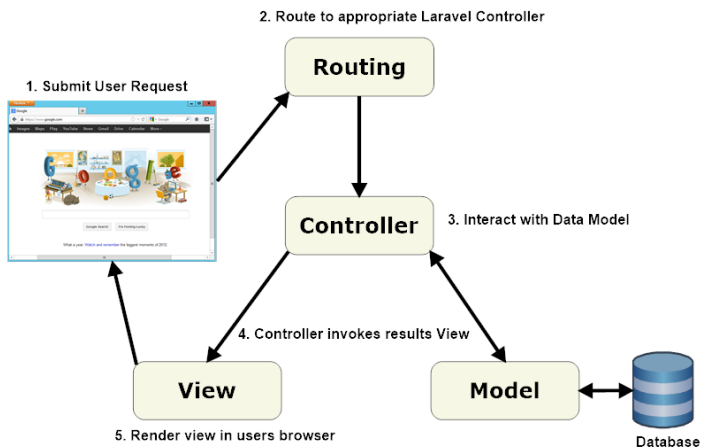
menyimpan banyak data yang sangat penting dan sensitive. Laravel memberikan kita beberapa pilihan penting yang dapat digunakan untuk membuat aplikasi kita agar tetap aman. ORM Laravel menggunakan PDO yang dapat mencegah SQL Injection, memiliki csrf token, dan banyak hal lainnya

- **Komunitas yang besar**

Laravel merupakan framework yang populer dan memiliki komunitas yang besar. Dengan adanya komunitas ini pengguna dapat lebih mudah belajar dan mencari solusi yang tepat atas setiap permasalahannya.

- **Hemat waktu**

Dengan berbagai abstraksi yang tersedia di Laravel. Pengguna jadi lebih fokus untuk memikirkan logika bisnis dari aplikasi yang dia buat. Jika ada developer baru yang masuk ke project, dia cukup mempelajari dokumentasi resmi Laravel sehingga lebih menghemat waktu [4].



Gambar 2.1 Arsitektur Laravel

Pada penelitian ini kerangka kerja Laravel digunakan sebagai kerangka kerja untuk aplikasi berbasis website yang digunakan pada sistem.

2.3 MySQL

Salah satu basis data yang sering digunakan bersama dengan bahasa pemrograman PHP adalah database MySQL.

MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem manajemen basis data SQL (Database Management System) atau sering disingkat DBMS. MySQL merupakan DBMS yang multithread, multi-user, yang bersifat gratis dan dibawah lisensi GNU General Public License. MySQL ini dipilih oleh banyak orang karena beberapa kelebihan yang ditawarkan:

- MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac OS X Server, Solaris dan masih banyak lagi
- Bersifat Open Source, MySQL didistribusikan secara open source (gratis)
- Bersifat Multi-User, MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
- MySQL memiliki kecepatan yang baik dalam menagani query (perintah SQL). Dengan kata lain MySQL dapat memproses lebih banyak SQL per satuan waktu.
- Dari segi security atau keamanan data, MySQL memiliki beberapa lapisan sekuriti, seperti level subnet mask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password yang terenkripsi.
- MySQL memiliki interface (antarmuka) terhadap aplikasi dan bahasa pemrograman.
- Dukungan banyak komunitas, biasanya tergabung dalam sebuah forum untuk saling berdiskusi membagi informasi tentang MySQL [5].

Pada penelitian ini MySQL digunakan sebagai basis data untuk sistem yang dibangun.

2.4 Python

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode. Dengan kata lain, Python diklaim sebagai bahasa pemrograman yang memiliki kode-kode pemrograman yang sangat jelas, lengkap, dan mudah untuk dipahami [6].

Python dianggap memiliki kehebatan untuk menangani pembuatan aplikasi-aplikasi kekinian yang mengandung kata kunci *big data*, *data mining*, *deep learning*, *data science*, hingga *machine learning*. Dengan kata lain, Python adalah bahasa pemrograman yang sederhana untuk membuat aplikasi berbasis kecerdasan buatan (*Artificial Intelligence*) [6].

Python secara umum berbentuk pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Istilah lainnya adalah bahasa pemrograman multi-paradigma.

Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi [6].

2.4.1 Platform Pemrograman Python

Python dapat dijalankan di berbagai platform sistem operasi. Oleh karena itu, distribusi aplikasi yang dibuat menggunakan Python sangatlah luas dan *multi-platform*.

Beberapa platform yang mendukung Python di antaranya Linux / Unix, Windows, Mac OS, Java Virtual Machine, OS/2, Amiga, Palm dan Symbian [6].

2.4.2 Lisensi Python

Pada prinsipnya, Python dapat diperoleh dan digunakan secara bebas oleh siapapun, bahkan bagi *developer* yang menggunakan bahasa pemrograman ini untuk kepentingan komersial. Namun pengguna *package* atau modul dari pihak ketiga mungkin saja membutuhkan lisensi yang berbeda, misalnya lisensi berbayar [6].

2.4.3 Fitur Bahasa Pemrograman Python

Beberapa fitur dan kelebihan yang dimiliki Python adalah:

- Memiliki koleksi kepustakaan yang banyak. Artinya telah tersedia modul-modul ‘siap pakai’ untuk berbagai keperluan seperti pembuatan permainan hingga kecerdasan buatan.
- Memiliki struktur bahasa yang jelas, sederhana dan mudah dipelajari
- Berorientasi prosedural dan objek sekaligus (multi-paradigma)
- Memiliki sistem pengelolaan memori otomatis (*garbage collection*)
- Bersifat modular sehingga mudah dikembangkan dengan menciptakan modul-modul baru, baik dibangun dengan bahasa Python maupun C/C++

Pada penelitian ini bahasa pemrograman Python digunakan untuk membuat *script* yang digunakan untuk memverifikasi integritas data dan mengembalikan cadangan data pada sistem.

2.5 Advanced Encryption Standard (AES) 256

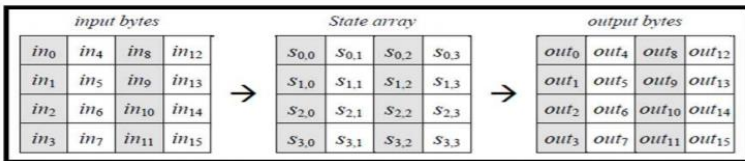
Advanced Encryption Standard (AES) merupakan algoritma kriptografi yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok ciphertext simetrik yang dapat mengenkripsi (encipher) dan dekripsi (decipher) informasi. AES dipilih karena kuat terhadap serangan differential, serangan truncated differential, serangan linear, serangan interpolation, dan serangan square [7].

Input dan output dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau plaintext yang nantinya akan dienkripsi menjadi ciphertext. Cipher key dari AES terdiri dari key dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah round yang akan diimplementasikan pada algoritma AES ini. Berikut ini adalah Tabel 2.1 yang memperlihatkan jumlah round / putaran (Nr) yang harus diimplementasikan pada masing-masing panjang kunci

Tabel 2.1 Perbandingan Jumlah Round dan Key [8]

	Jumlah Key (Nk)	Ukuran Block (Nb)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Pada dasarnya, operasi AES dilakukan terhadap array of byte dua dimensi yang disebut dengan state. State mempunyai ukuran NROWS X NCOLS. Pada awal enkripsi, data masukan yang berupa $in_0, in_2, in_3, in_4, in_5, in_6, in_7, in_8, in_9, in_{10}, in_{11}, in_{12}, in_{13}, in_{14}, in_{15}$ disalin ke dalam array state. State inilah yang nantinya dilakukan operasi enkripsi / dekripsi. Kemudian keluarannya akan ditampung ke dalam array out. Gambar 2.2 mengilustrasikan proses penyalinan dari input bytes, state array, dan output bytes.

**Gambar 2.2 Proses Input Bytes, State Array, dan Output Bytes [8]**

Pada saat permulaan, input bit pertama kali akan disusun menjadi suatu array byte dimana panjang dari array byte yang digunakan pada AES adalah sepanjang 8 bit data. Array byte inilah yang nantinya akan dimasukkan atau dicopy ke dalam state dengan urutan dimana r (row / baris) dan c (column/kolom):

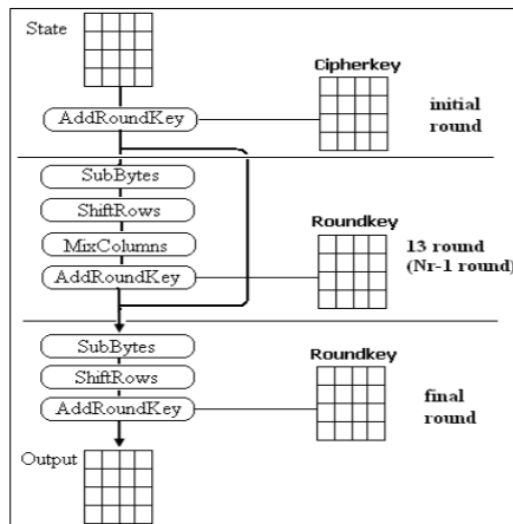
$$s[r,c] = in[r+4c] \text{ untuk } 0 \leq r < 4 \text{ dan } 0 \leq c < Nb$$

sedangkan dari state akan dicopy ke output dengan urutan:

$$\text{out}[r+4c] = \text{s}[r,c] \text{ untuk } 0 \leq r < Nb$$

2.5.1 Proses Enkripsi Advanced Encryption Standard

Proses enkripsi algoritma AES terdiri dari empat jenis transformasi bytes, yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Pada awal proses enkripsi, input yang telah dicopykan 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. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar 2.3.



Gambar 2.3 Ilustrasi Proses Enkripsi AES [8]

2.5.1.1 AddRoundKey

Pada proses enkripsi dan dekripsi AES proses AddRoundKey sama, sebuah round key ditambahkan pada state dengan operasi XOR. Setiap round key terdiri dari Nb word dimana tiap word tersebut akan dijumlahkan dengan word atau kolom yang bersesuaian dari state sehingga:

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{round*Nb+c}] \text{ untuk } 0 \leq c \leq Nb$$

[w_i] adalah word dari key yang bersesuaian dimana $i = \text{round} * Nb + c$. Transformasi AddRoundKey pada proses enkripsi pertama kali pada round = 0 untuk round selanjutnya round = round + 1, pada proses dekripsi pertama kali pada round = 14 untuk round selanjutnya round = round - 1.

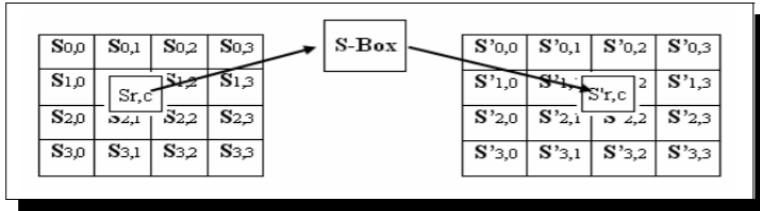
2.5.1.2 SubBytes

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Tabel substitusi S-Box akan dipaparkan dalam Tabel 2.2.

Tabel 2.2 S-Box SubBytes

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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

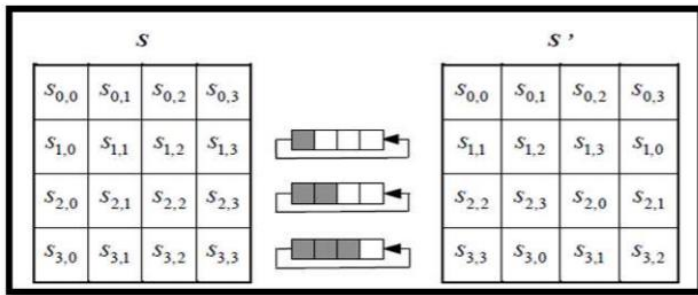
Untuk setiap byte pada array state, misalkan $S[r, c] = xy$, yang dalam hal ini xy adalah digit heksadesimal dari nilai $S[r, c]$, maka nilai substitusinya, dinyatakan dengan $S'[r, c]$, adalah elemen di dalam tabel substitusi yang merupakan perpotongan baris x dengan kolom y . Gambar 2.4 mengilustrasikan pengaruh pemetaan byte pada setiap byte dalam state



Gambar 2.4 Pengaruh Pemetaan pada Setiap Byte [8]

2.5.1.3 Shiftrows

Transformasi Shiftrows pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Proses pergeseran Shiftrow ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Transformasi ShiftRows [8]

2.5.1.4 MixColumns

MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi mixcolumns dapat dilihat pada perkalian matriks berikut ini:

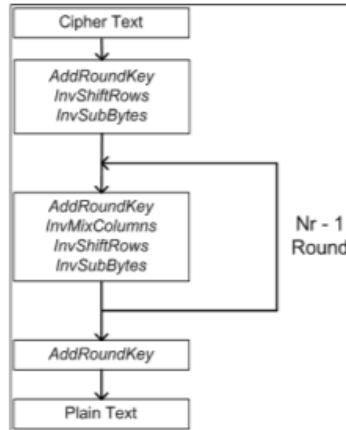
$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad \dots [1]$$

Hasil dari perkalian matriks diatas dapat dianggap seperti perkalian yang ada di bawah ini :

$$\begin{aligned} s'_{0,c} &= (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \\ s'_{3,c} &= (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) \end{aligned} \quad \dots [2]$$

2.5.2 Proses Deskripsi Advanced Encryption Standard

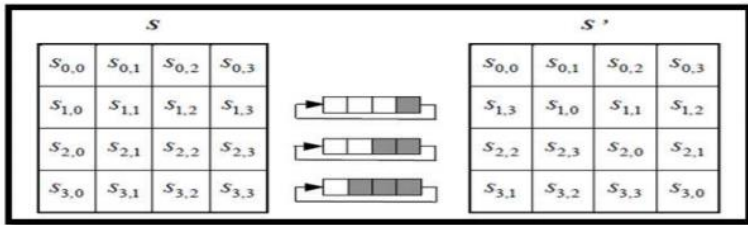
Transformasi cipher dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada invers cipher adalah InvShiftRows, InvSubBytes, dan AddRoundKey. Algoritma dekripsi dapat dilihat pada gambar 2.6.



Gambar 2.6 Ilustrasi Proses Deskripsi AES [8]

2.5.2.1 *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 2.7.



Gambar 2.7 Transformasi *InvShiftRows* [8]

2.5.2.2 *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 Tabel 2.3.

2.5.2.3 *InvMixColumns*

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

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

Hasil dari perkalian dalam matrik adalah:

$$s'_{0,c} = (\{0E\} \bullet s_{o,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{o,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c})$$

$$s'_{2,c} = (\{0D\} \bullet s_{o,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0B\} \bullet s_{o,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})$$

Tabel 2.3 Tabel Inverse S-Box [8]

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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	54
	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

2.5.3 Proses Ekspansi Kunci

Algoritma AES mengambil kunci cipher dan melakukan rutin ekspansi kunci (key expansion) untuk membentuk key schedule. Ekspansi kunci menghasilkan total $Nb(Nr+1)$ word. Algoritma ini membutuhkan set awal key yang terdiri dari Nb word, dan setiap round Nr membutuhkan data kunci sebanyak Nb word. Hasil key schedule terdiri dari array 4 byte word linear yang dinotasikan dengan $[w_i]$. SubWord adalah fungsi yang mengambil 4 byte word input dan mengaplikasikan S-Box ke tiap-tiap data 4 byte untuk menghasilkan word output. Fungsi RotWord mengambil word $[a_0, a_1, a_2, a_3]$ sebagai input, melakukan permutasi siklik, dan mengembalikan word $[a_1, a_2, a_3, a_0]$. $Rcon[i]$

terdiri dari nilai-nilai yang diberikan oleh $[x_{i-1}, \{00\}, \{00\}, \{00\}]$, dengan x_{i-1} sebagai pangkat dari x (x dinotasikan sebagai $\{02\}$). Pseudocode dari proses ekspansi kunci dapat dilihat dalam gambar 2.8.

```

KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
  word temp

  i = 0

  while (i < Nk)
    w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
    i = i+1
  end while

  i = Nk

  while (i < Nb * (Nr+1))
    temp = w[i-1]
    if (i mod Nk = 0)
      temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
    else if (Nk > 6 and i mod Nk = 4)
      temp = SubWord(temp)
    end if
    w[i] = w[i-Nk] xor temp
    i = i + 1
  end while
end

```

Gambar 2.8 Pseudocode Ekspansi Kunci [8]

Dari Gambar 2.8 dapat dilihat bahwa word ke Nk pertama pada ekspansi kunci berisi kunci cipher. Setiap word berikutnya, $w[i]$, sama dengan XOR dari word sebelumnya, $w[i-1]$ dan word Nk yang ada pada posisi sebelumnya, $w[i-Nk]$. Untuk word pada posisi yang merupakan kelipatan Nk , sebuah transformasi diaplikasikan pada $w[i-1]$ sebelum XOR, lalu dilanjutkan oleh XOR dengan konstanta round, $Rcon[i]$. Transformasi ini terdiri dari pergeseran siklik dari byte data dalam suatu word $RotWord$, lalu diikuti aplikasi dari lookup tabel untuk semua 4 byte data dari word $SubWord$ [8].

Pada penelitian ini algoritma AES 256 digunakan sebagai algoritma enkripsi pada file yang tersimpan pada sistem.

2.6 Secure Hash Algorithm 256 (SHA-256)

Pada bulan Agustus 1991, NIST (The National Institute of Standard and Technology) mengumumkan bakuan (standard) untuk tanda-tangan digital yang dinamakan *Digital Signature Standard (DSS)*. DSS terdiri dari dua komponen, yang pertama

adalah algoritma tanda-tangan digital yang disebut *Digital Signature Algorithm (DSA)*, dan yang kedua adalah fungsi *hash* standard yang disebut *Secure Hash Algorithm* [9].

SHA adalah fungsi *hash* satu arah yang dibuat oleh NIST dan digunakan bersama DSS. SHA dinyatakan sebagai standard fungsi *hash* satu arah oleh United States National Security Agency (NSA). SHA didasarkan pada MD4 yang dibuat oleh Ronald L. Rivest dari MIT [9].

2.6.1 Awal Perkembangan SHA-256

Awal terbentuknya SHA-256 dimulai dari sejarah algoritma SHA, dimana hingga saat ini ada lima algoritma SHA yaitu SHA-0, SHA-1, SHA-224, SHA-256, SHA-384, dan SHA-512. Varian SHA-0 dikenal dengan SHA-0 pada tahun 1991, varian SHA-1 dikenal dengan SHA-1 pada tahun 1993, varian SHA-224, SHA-256, SHA-384, dan SHA-512 dikenal dengan SHA-2 pada tahun 2000 [10] [11]. Dari seitulah SHA-256 muncul yang merupakan pecahan dari SHA- 2 yang mempunyai varian di dalamnya antara lain : varian SHA-224, SHA-256, SHA-384, dan SHA-512.

2.6.2 Dasar Prinsip

Algoritma SHA-256 dapat digunakan untuk menghitung nilai message digest dari sebuah pesan, dimana pesan tersebut memiliki panjang maksimum 2^{64} bit. Algoritma ini menggunakan sebuah message schedule yang terdiri dari 64 element 32-bit word, delapan buah variabel 32-bit, dan variabel penyimpan nilai *hash* 8 buah word 32-bit. Hasil akhir dari algoritma SHA-256 adalah sebuah message digest sepanjang 256-bit [12].

2.6.3 Cara Kerja

Cara Kerja SHA-256 mengubah pesan masukan ke dalam message digest 256 bit. Berdasarkan *Secure Hash Signature Standard*, pesan masukan yang panjangnya lebih pendek dari 2^{64} bit, harus dioperasikan oleh 512 bit dalam kelompok dan menjadi sebuah message digest 256-bit [13].

Tahapan-tahapan cara kerja SHA-256 adalah sebagai berikut [13]:

1. Message Padding : Pada tahap pertama, pesan yang berupa binary disisipkan dengan angka 1 dan ditambahkan bit-bit pengganjal yakni angka 0 hingga panjang pesan tersebut kongruen dengan 448 modulo 512. Panjang pesan yang asli kemudian ditambahkan sebagai angka biner 64 bit. Setelah itu maka panjang pesan sekarang menjadi kelipatan 512 bit.
2. Parsing: Pesan yang sudah dipadding tadi kemudian dibagi menjadi N buah blok 512 bit: $M^{(1)}$, $M^{(2)}$, ..., $M^{(N)}$.
3. Message Expansion: Masing-masing blok 512-bit tadi lalu dipecah menjadi 16 buah word 32-bit: $M_0^{(i)}$, $M_1^{(i)}$, ..., $M_{15}^{(i)}$ yang mana nantinya diperluas menjadi 64 word yang diberi label W_0 , W_1 , ..., W_{63} dengan aturan tertentu yang sudah ditentukan sebelumnya oleh standar SHA-2.

Initial Hash Value of SHA-256

$A=H_0^{(0)}$	6a09e667
$B=H_1^{(0)}$	bb67ae85
$C=H_2^{(0)}$	3c6ef372
$D=H_3^{(0)}$	a54ff53a
$E=H_4^{(0)}$	510e527f
$F=H_5^{(0)}$	9b05688c
$G=H_6^{(0)}$	1f83d9ab
$H=H_7^{(0)}$	5be0cd19

Gambar 2.9 Nilai Awal pada Variabel H

4. Message Compression: Masing-masing dari 64 word yang diberi label W_0 , W_1 , ..., W_{63} tadi kemudian diproses dengan algoritma fungsi *hash* SHA-256. Dalam proses tersebut, inti utama dari algoritma SHA-256 adalah membuat 8 variabel yang diberikan nilai untuk nilai awal dari $H_0^{(0)}$ -

$H_7^{(0)}$ di awal masing-masing fungsi *hash*. Nilai-nilai awal tersebut dapat dilihat pada Gambar 2.9.

5. Algoritma ini melakukan perhitungan sebanyak 64 kali putaran untuk setiap perhitungan blok. Delapan variabel yang diberi label A, B, C, ..., H tadi nilainya terus berganti selama perputaran sebanyak 64 kali putaran sebagai berikut:

$$T_1 = H + \sum_1(E) + Ch(E, F, G)[1] + K_t + W_t \quad (1)$$

$$T_2 = \sum_0(A) + Maj(A, B, C)[1] \quad (2)$$

$$H = G \quad (3)$$

$$G = F \quad (4)$$

$$F = E \quad (5)$$

$$E = D + T_1 \quad (6)$$

$$D = C \quad (7)$$

$$C = B \quad (8)$$

$$B = A \quad (9)$$

$$A = T_1 + T_2 \quad (10)$$

6. Setelah perputaran sebanyak 64 kali tadi, nilai *hash* $H^{(i)}$ kemudian dihitung sebagai berikut:

$$H_0^{(i)} = a + H_0^{(i-1)}$$

$$H_1^{(i)} = b + H_1^{(i-1)}$$

$$H_2^{(i)} = c + H_2^{(i-1)}$$

$$H_3^{(i)} = d + H_3^{(i-1)}$$

$$H_4^{(i)} = e + H_4^{(i-1)}$$

$$H_5^{(i)} = f + H_5^{(i-1)}$$

$$H_6^{(i)} = g + H_6^{(i-1)}$$

$$H_7^{(i)} = h + H_7^{(i-1)}$$

7. Selanjutnya hasil akhir SHA-256 didapat dari penggabungan delapan variabel yang tadi sudah dikomputasi.

$$H_0^{(N)} \parallel H_1^{(N)} \parallel H_2^{(N)} \parallel H_3^{(N)} \parallel H_4^{(N)} \parallel H_5^{(N)} \parallel H_6^{(N)} \parallel H_7^{(N)}$$

Pada penelitian ini SHA-256 digunakan untuk menghasilkan *secret hash key* yang digunakan untuk proses verifikasi integritas file.

(Halaman ini sengaja dikosongkan)

BAB III

PERANCANGAN

Bab ini membahas mengenai perancangan implementasi sistem yang dibangun pada tugas akhir. Bagian yang akan dijelaskan pada bab ini adalah deskripsi umum sistem dan rancangan sistem.

3.1 Deskripsi Umum

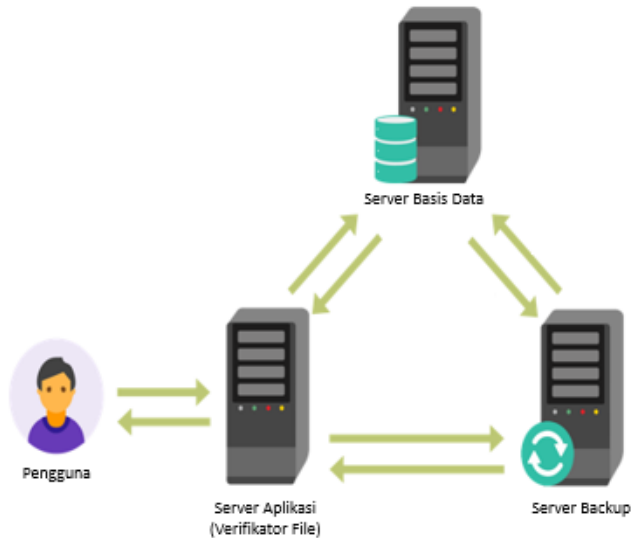
Sistem yang akan dibangun pada penelitian ini adalah sebuah sistem penyimpanan data yang dapat melakukan verifikasi integritas file secara otomatis. Verifikasi integritas file dilakukan dengan cara membandingkan nilai dari *hash* key yang dihasilkan dari file dengan *hash* key yang disimpan oleh Server Basis Data. Setiap data yang tersimpan pada sistem dienkripsi dengan menggunakan algoritma enkripsi AES-256.

Arsitektur sistem yang akan dibangun ini memiliki tiga server utama, yaitu Server Aplikasi, Server Basis Data dan Server Backup. Server Aplikasi memiliki peran sebagai server utama untuk berjalannya aplikasi berbasis *website* dengan kerangka kerja Laravel. Server Aplikasi juga menyimpan file terenkripsi yang telah diunggah oleh pengguna dan mengirimkan file tersebut ke Server Backup. Sedangkan Server Backup berperan untuk menyimpan file yang telah dikirimkan oleh Server Aplikasi dan untuk mengembalikan file yang telah hilang atau termodifikasi dari Server Aplikasi. Yang terakhir adalah Server Basis Data yang memiliki peran untuk menyimpan basis data yang digunakan oleh aplikasi pada Server Aplikasi menggunakan basis data MySQL dan melakukan verifikasi integritas file yang tersimpan pada Server Aplikasi. Arsitektur Sistem bisa dilihat pada gambar [].

Dengan adanya sistem ini diharapkan nantinya pengguna bisa menyimpan data dengan aman tanpa perlu mengkhawatirkan data yang mereka simpan pada *cloud* hilang, dimodifikasi dan dicuri.

3.2 Rancangan Sistem

Rancangan sistem yang akan dibuat meliputi arsitektur sistem, rancangan spesifikasi kebutuhan sistem, rancangan proses pengunggahan file, rancangan proses pengunduhan file, rancangan verifikasi file, rancangan pengembalian file dan rancangan pencadangan basis data. Arsitektur sistem dapat dilihat pada gambar 3.10.



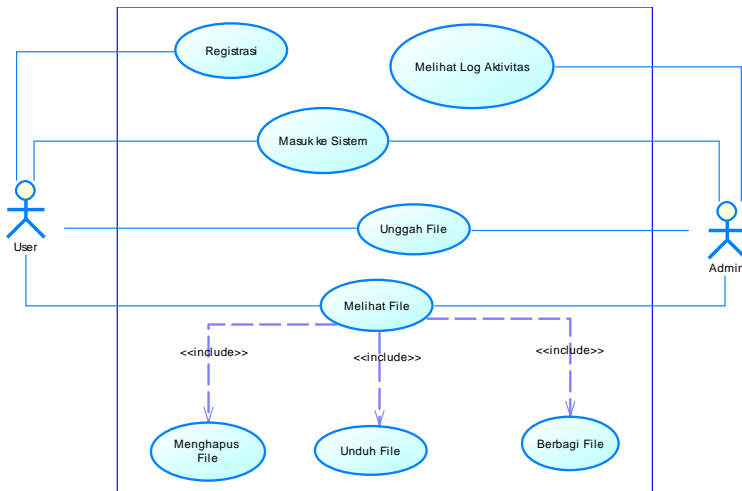
Gambar 3.10 Arsitektur Sistem

3.3 Server Aplikasi

Pada server aplikasi ini akan dibangun aplikasi berbasis website dengan menggunakan kerangka kerja Laravel dan program verifikasi file dengan menggunakan bahasa pemrograman Python

3.3.1 Rancangan Spesifikasi Kebutuhan Aplikasi

Rancangan spesifikasi kebutuhan aplikasi pada tugas akhir ini dapat digambarkan dalam bentuk diagram kasus penggunaan seperti gambar 3.11.



Gambar 3.11 Kasus Penggunaan

Diagram kasus penggunaan pada Gambar 3.11 akan dijelaskan masing-masing pada tabel 3.4.

Tabel 3.4 Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan	Deskripsi
UC-001	Masuk ke Sistem	User dan admin dapat masuk ke sistem
UC-002	Registrasi	User dapat mendaftarkan dirinya ke sistem
UC-003	Unggah File	User dan admin dapat mengunggah file ke sistem

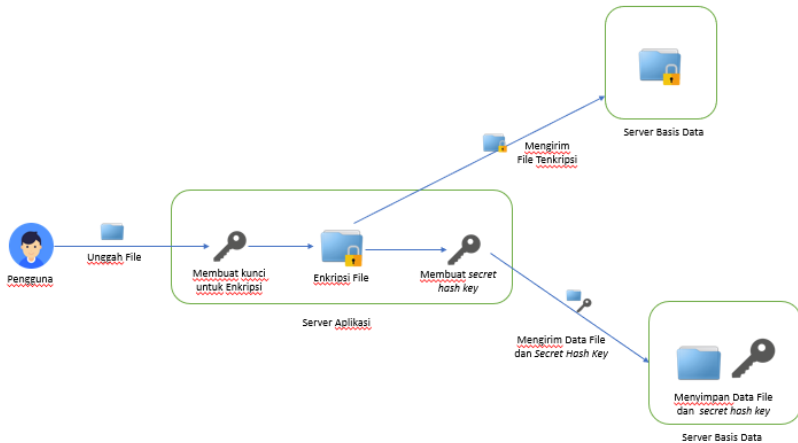
UC-004	Melihat File	User dan admin dapat melihat file yang telah diunggah oleh dirinya dan file yang telah dibagikan oleh user yang lain
UC-005	Unduh File	User dan admin dapat mengunduh file yang telah diunggah oleh dirinya dan file yang telah dibagikan oleh user yang lain
UC-006	Menghapus File	User dan admin dapat menghapus file yang telah diunggah oleh dirinya
UC-007	Berbagi File	User dan admin dapat membagikan file yang telah diunggah oleh dirinya kepada user yang lain
UC-008	Melihat Log Aktivitas	Admin dapat melihat log aktivitas yang telah dilakukan oleh user dan admin
UC-009	Melihat Log Berbagi File	Admin dapat melihat log kegiatan berbagi file yang telah dilakukan user dan admin

3.3.2 Rancangan Mekanisme Pengunggahan File

Sebelum file yang diunggah oleh pengguna disimpan di server, terdapat beberapa tahap yang dilakukan oleh sistem untuk memperkuat keamanan data. Berikut adalah tahap-tahap dalam proses pengunggahan data:

1. Pengguna mengunggah File.

Pengguna mengunggah file dari form yang tersedia di aplikasi. Aplikasi akan mencatat data dari file tersebut di basis data.



Gambar 3.12 Ilustrasi Mekanisme Unggah File

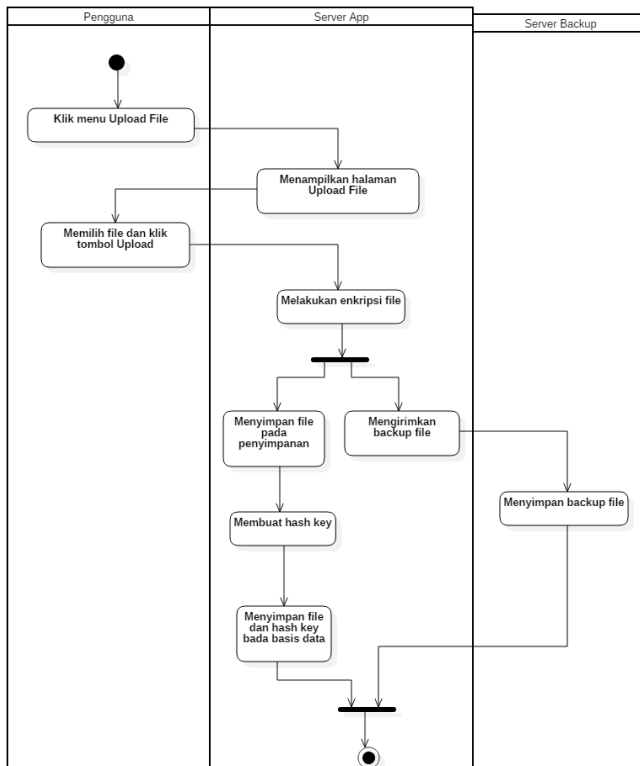
2. Enkripsi file.
Saat file sudah terunggah, sistem akan mengenkripsi file tersebut dengan menggunakan algoritma AES-256 dengan kunci dari pengguna tersebut dan file tersebut akan disimpan di sistem. Sistem juga mencatat waktu yang digunakan untuk proses enkripsi ini
3. Simpan file
Setelah file berhasil dienkrip, sistem akan menyimpan file tersebut dengan nama urutan nomor file tersebut sehingga dipastikan tidak ada file dengan nama yang sama pada sistem. Format file yang telah dienkripsi juga diubah. Sistem juga akan menyimpan catatan log aktivitas di basis data.
4. Cadangkan file
Setelah file terenkripsi tersimpan di server aplikasi, sistem akan mengirimkan file terenkripsi tersebut ke server backup.
5. Membuat *secret hash key*.
Sistem akan membuat *secret hash key* dengan menggunakan algoritma SHA-256 untuk disimpan di basis

data. *Secret hash key* ini nantinya akan digunakan untuk proses verifikasi file tersebut.

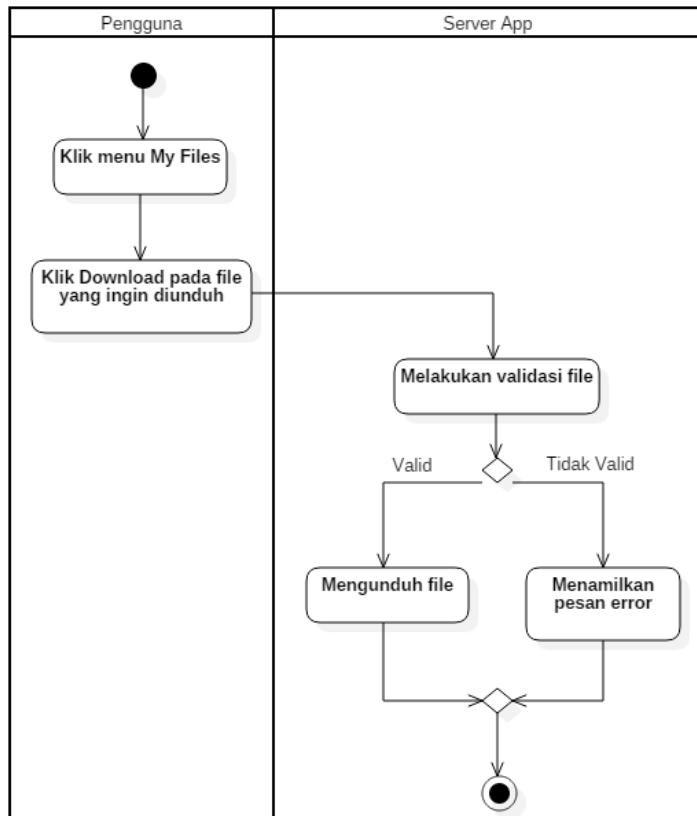
3.3.3 Rancangan Mekanisme Pengunduhan File

Untuk mengunduh file terdapat beberapa tahap yang dilakukan oleh sistem.

1. Verifikasi file
Untuk mengetahui bahwa file yang akan diunduh tidak mengalami modifikasi, *secret hash key* pada file tersebut akan dibandingkan dengan *secret hash key* yang sudah tersimpan di basis data. Jika *secret hash key* file dan basis data sama, maka file tersebut tidak mengalami modifikasi dan terverifikasi. Begitu juga sebaliknya.
2. Dekripsi file.
Setelah file terverifikasi, sistem akan mendekripsi file tersebut dengan menggunakan key yang sesuai dan sistem mencatat waktu yang digunakan untuk proses dekripsi ini. Sistem juga akan menyimpan catatan log aktivitas di basis data.
3. Download file
Setelah file didekripsi barulah file tersebut dikirimkan ke pengguna.



Gambar 3.13 Alur Pengunggahan File



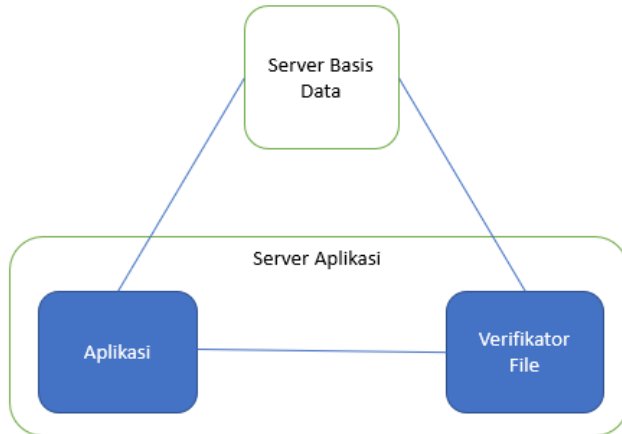
Gambar 3.14 Alur Pengunduhan File

3.3.4 Rancangan Verifikasi Integritas File

Proses verifikasi integritas file pada penelitian ini menggunakan program pihak ketiga yang dibangun menggunakan Bahasa pemrograman Python. Program ini akan berjalan setiap lima menit agar proses verifikasi ini selalu *up-to-date*.

Skenario verifikasi integritas file ini dimulai dari program verifikasi membaca data file dari basis data dan data file dari penyimpanan. Jika ada file yang hilang atau termodifikasi pada

penyimpanan maka program verifikasi ini akan mencatat bahwa file tersebut telah hilang atau termodifikasi.



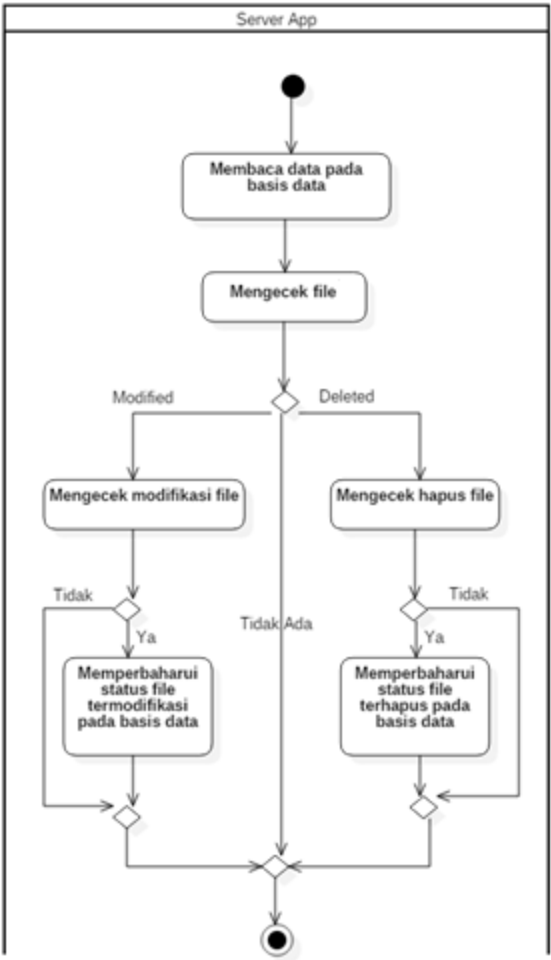
Gambar 3.15 Hubungan Antara Server Aplikasi dengan Server Basis Data

3.3.5 Rancangan Antarmuka

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan sistem. Antarmuka pengguna yang berhubungan langsung dengan pengguna harus memiliki tampilan yang menarik dan mudah dipahami. Sistem ini memiliki beberapa antarmuka pengguna yang mana akan dijelaskan di bawah ini.

3.3.5.1 Rancangan Antarmuka *Login*

Halaman ini digunakan pengguna untuk kebutuhan Masuk ke Sistem (UC-001). Pada halaman ini terdapat dua isian yang harus diisi oleh pengguna untuk masuk ke sistem yaitu email dan password. Setelah itu pengguna diharuskan menekan tombol Masuk untuk masuk ke sistem. Rancangan halaman antarmuka login bisa dilihat pada gambar 3.15.



Gambar 3.16 Alur Verifikasi File



A login form with a light gray background and a black border. At the top center is the title "Login" in bold black font. Below the title are two input fields: the first is labeled "Email" and the second is labeled "Password". Both labels are in a light gray font and positioned to the left of their respective input boxes. At the bottom center is a button labeled "Masuk" in black font, with a blue border and a light gray background.

Gambar 3.17 Rancangan Antarmuka Masuk ke Sistem

3.3.5.2 Rancangan Antarmuka Registrasi

Halaman ini digunakan untuk kebutuhan Registrasi (UC-002). Pada halaman ini terdapat empat isian yang harus diisi oleh pengguna yaitu email, nama, password dan konfirmasi password. Setelah itu pengguna diharuskan untuk menekan tombol registrasi untuk mendaftar. Rancangan antarmuka halaman registrasi bisa dilihat pada gambar 3.16.



A registration form with a light gray background and a black border. At the top center is the title "Registrasi" in bold black font. Below the title are four input fields arranged vertically. Each field has a label to its left: "Email", "Nama", "Password", and "Konfirmasi". The labels are in a light gray font. The input boxes contain placeholder text in a light gray font: "Email", "Nama", "Password", and "Konfirmasi Password". At the bottom center is a button labeled "Daftar" in black font, with a light gray background and a black border.

Gambar 3.18 Rancangan Antarmuka Registrasi

3.3.5.3 Rancangan Antarmuka Unggah File

Halaman ini digunakan untuk kebutuhan Unggah File (UC-003). Terdapat satu isian yang digunakan untuk memilih file yang akan diunggah. Setelah pengguna memilih file pengguna diharuskan untuk menekan tombol upload untuk mengunggah file yang telah dipilih tersebut. Rancangan antarmuka dari mengunggah file bisa dilihat pada gambar 3.17.



Upload File

Choose File Browse

Upload

Gambar 3.19 Rancangan Antarmuka Unggah File

3.3.5.4 Rancangan Antarmuka Melihat File

Halaman ini digunakan untuk kebutuhan Melihat File (UC-004). Pada halaman ini terdapat dua tabel. Tabel yang pertama adalah tabel file saya yang menampilkan file yang telah diunggah oleh pengguna sedangkan tabel yang kedua adalah tabel file dibagikan ke saya yang menampilkan data file yang telah dibagikan oleh pengguna lain.

Pada tabel file saya terdapat empat kolom yang berisikan nama file, ukuran file, format file dan aksi yang berisikan 3 tombol yaitu Download untuk kebutuhan Unduh File (UC-005), tombol Bagikan untuk kebutuhan Bagikan File (UC-006) dan tombol Hapus untuk kebutuhan Menghapus File (UC-007). Rancangan antarmuka file saya bisa dilihat pada gambar 3.18.

File Saya					
▼ Filename	▼ Size	▼ Format	▼ Aksi		
File 1.png	322 KB	image/png	Download	Bagikan	Hapus
File 2.jpg	2019 KB	image/jpg	Download	Bagikan	Hapus
File 3.exe	309 KB	application/exe	Download	Bagikan	Hapus
File 4.rar	512 KB	archive/rar	Download	Bagikan	Hapus

Gambar 3.20 Rancangan Antarmuka Melihat File Saya

Pada tabel file dibagikan ke saya terdapat empat kolom yang berisikan nama file, ukuran file, format file dan aksi yang berisikan tombol Download untuk kebutuhan Unduh File (UC-005). Rancangan antarmuka file dibagikan ke saya bisa dilihat pada gambar 3.19.


File Dibagikan ke Saya				
▼ Pemilik	▼ Filename	▼ Size	▼ Format	▼ Aksi
User1@mail.com	File 1.png	322 KB	image/png	Download
User2@mail.com	File 2.jpg	201 KB	image/jpg	Download
User3@mail.com	File 3.exe	207 KB	application/exe	Download

Gambar 3.21 Rancangan Antarmuka Melihat File Dibagikan ke Saya

3.3.5.5 Rancangan Antarmuka Berbagi File

Halaman ini digunakan untuk kebutuhan Berbagi File. Halaman ini memiliki dua isian. Isian yang pertama adalah pilihan jenis privasi yang berisikan privasi sebagai isian default dan publik.

Jika pengguna memilih jenis privasinya berisi privasi maka akan muncul checkbox bagikan yang apabila checkbox bagikan seperti pada gambar 3.20. Apabila checkbox diisi oleh pengguna maka akan muncul isian baru yang berlabel email seperti pada gambar 3.21.



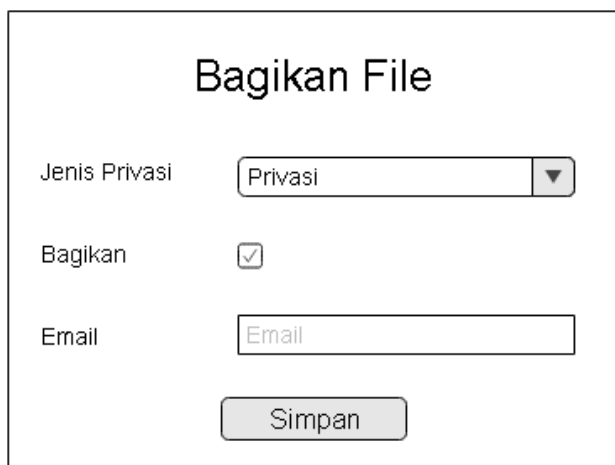
Bagikan File

Jenis Privasi

Bagikan ☐

Simpan

Gambar 3.22 Rancangan Antarmuka Bagikan File 1



Bagikan File

Jenis Privasi

Bagikan ☒

Email

Simpan

Gambar 3.23 Rancangan Antarmuka Bagikan File 2

Jika pengguna memilih jenis privasi berisi publik maka checkbox bagikan tidak ditampilkan seperti pada gambar 3.22.



Gambar 3.24 Rancangan Antarmuka Bagikan File 3

3.3.5.6 Rancangan Antarmuka Melihat Log Aktivitas

Halaman ini digunakan untuk kebutuhan Melihat Log Aktivitas (UC-008). Halaman ini menampilkan tabel yang memiliki enam kolom yaitu User, nama file, ukuran file, format file, durasi (enkripsi atau dekripsi) dan aksi. Rancangan antarmuka dari Melihat Log Aktivitas bisa dilihat pada gambar 3.23.

▼ User	▼ Filename	▼ Size	▼ Format	▼ Durasi	▼ Aksi
User1@mail.com	File 1 .png	322 kB	image/png	105ms	Download
User2@mail.com	File 2 .jpg	2019 kB	image/jpg	549ms	Upload
User3@mail.com	File 3 .exe	309 kB	application/exe	0ms	Delete
User4@mail.com	File 4 .rar	512 kB	archive/rar	0ms	Request-Checksum
admin@mail.com	File 4 .rar	512 kB	archive/rar	0ms	Accept-Request

Gambar 3.25 Rancangan Antarmuka Melihat Log Aktivitas

3.3.5.7 Rancangan Antarmuka Melihat Log Berbagi File

Halaman ini digunakan untuk kebutuhan Melihat Log Berbagi File (UC-009). Halaman ini menampilkan tabel yang

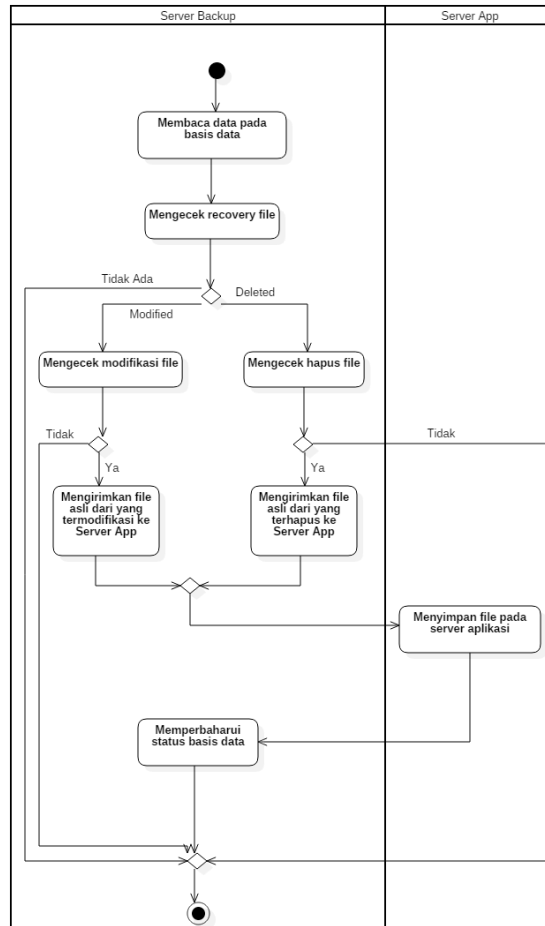
memiliki empat kolom, yaitu Pemilik, File, Penerima dan Status. Rancangan antarmuka dari Melihat Log Berbagi bisa dilihat pada gambar 3.24.

Log Berbagi File			
▼ Pemilik	▼ Filename	▼ Penerima	▼ Status <input type="checkbox"/>
User1@mail.com	File 1.png	User2@mail.com	Tersedia
User2@mail.com	File 2.jpg	User3@mail.com	Dihapus
User3@mail.com	File 3.exe	User1@mail.com	Data Berubah

Gambar 3.26 Rancangan Antarmuka Melihat Log Berbagi File

3.4 Rancangan Server Backup

Pada server backup ini akan dibangun sebuah program dengan bahasa pemrograman Python yang memiliki fungsi untuk mengembalikan data yang hilang atau termodifikasi ke server aplikasi. Langkah-langkah proses pengembalian data dapat dilihat pada gambar 3.25.*

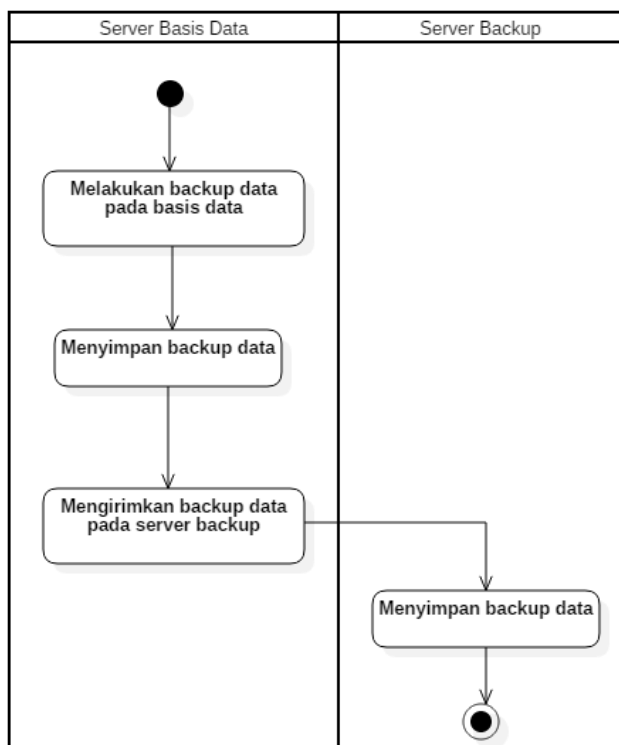


Gambar 3.27 Alur Pengembalian File

3.5 Rancangan Server Basis Data

Server basis data berfungsi sebagai *host* basis data dan pencadangan basis data. Hasil pencadangan basis data ini akan disimpan di server basis data dan server backup. Untuk mengirimkan data cadangan basis data ini menggunakan sebuah

program dengan bahasa pemrograman Python. Langkah-langkah proses pencadangan dapat dilihat pada gambar 3.26.



Gambar 3.28 Alur Pencadangan Basis Data

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem sesuai dengan analisis dan perancangan proses bisnis secara umum pada sistem Penyimpanan Data yang telah dijabarkan pada bab sebelumnya.

Implementasi yang akan dijelaskan meliputi lingkungan pembangunan sistem atau perangkat lunak, kode sumber utama yang berisi *pseudocode*, implementasi antarmuka perangkat lunak dan implementasi *client*. Arsitektur sistem yang digunakan adalah MVC dengan kerangka kerja Laravel.

4.1 Lingkungan Pembangunan Sistem

Lingkungan sistem yang digunakan untuk membangun perangkat lunak ini:

1. Windows 10 Enterprise sebagai sistem operasi
2. Sublime Text Editor 3128 sebagai *Integrated Development Environment (IDE)*
3. Laravel 5.8 sebagai kerangka kerja (*framework*)
4. PHP 7.2.12 sebagai bahasa pemrograman yang digunakan.
5. Python 3.7.3 sebagai bahasa pemrograman untuk program pihak ketiga.
6. MariaDB Server 10.1.37 (MySQL) dan HeidiSQL sebagai sistem manajemen basis data
7. Apache 2.4.37 sebagai *web server*

4.2 Implementasi Server Aplikasi

Sistem yang dibuat memiliki lapisan-lapisan yang direpresentasikan dalam kelas, yaitu view sebagai lapisan antarmuka pengguna, *controller* sebagai tempat untuk menerima *request* yang dikirim oleh aplikasi *client* dan mengirim balik *response*, *service* sebagai tempat pemrosesan data komputasi, *repository* sebagai tempat untuk melakukan pengelolaan terhadap

basis data dan *model* sebagai representasi dari setiap tabel di basis data.

4.2.1 Implementasi Pengunggahan File

Fungsi Pengunggahan File direpresentasikan pada fungsi upload dengan parameter file yang diunggah. *Pseudocode* fungsi ini dapat dilihat pada Kode Sumber 4.1

```

1. function upload(file)
2.     GET file
3.     validate file
4.
5.     SET key encryption
6.     encrypt file
7.     save encrypted file
8.     send encrypted file to application server
9.
10.    generate hash key
11.
12.    insert file, execution time, hash key into
        file model
13.    insert file into log model

```

Kode Sumber 4.1 *Pseudocode* Pengunggahan File

4.2.2 Implementasi Pengunduhan File

Fungsi Pengunduhan File direpresentasikan pada fungsi download dengan parameter id file yang akan diunduh. *Pseudocode* fungsi ini dapat dilihat pada Kode Sumber 4.2.

```

1. function download(file)
2.     GET file from storage
3.
4.     IF file not valid THEN
5.         decrypt file
6.         insert file into log model
7.         RETURN download decrypted file
8.     ELSE
9.         RETURN error message

```


10. ENDIF

Kode Sumber 4.2 *Pseudocode* Pengunduhan File

4.2.3 Implementasi Verifikasi Integritas File

Script verifikasi file ini dibangun menggunakan bahasa pemrograman Python. *Pseudocode* Verifikasi File ini bisa dilihat pada Kode Sumber 4.3.

```

1. GET data from file model
2. FOR row in record THEN
3.     GET file from storage
4.     IF file not valid THEN
5.         update invalid into file model
6.     ELSE IF file not found
7.         update not found into file model
8.     ENDIF
9. ENDFOR

```

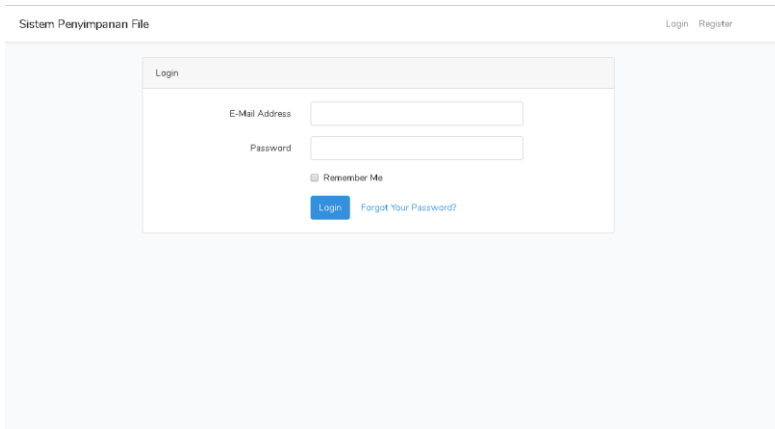
Kode Sumber 4.3 *Pseudocode* Verifikasi File

4.2.4 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna dibuat menggunakan HTML dengan *template* AdminBSB dan dengan *template engine* dari Laravel: blade. Pada subbab ini akan menjelaskan dan menampilkan tampilan halaman antarmuka yang diimplementasikan sesuai dengan rancangan antarmuka yang terdapat pada bab 3.

4.2.4.1 Halaman Login

Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-001, yaitu Masuk ke Sistem. Halaman antarmuka *login* menampilkan halaman untuk pengguna masuk ke dalam sistem dengan cara memasukkan email dan password. Tampilan implementasi halaman *login* dapat dilihat pada gambar 4.30.



Gambar 4.29 Halaman Login

4.2.4.2 Halaman Registrasi

Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-002, yaitu Registrasi. Halaman antarmuka Registrasi menampilkan halaman untuk pengguna mendaftarkan dirinya ke dalam sistem dengan cara memasukkan nama, email, password dan konfirmasi password. Tampilan implementasi halaman registrasi dapat dilihat pada gambar 4.31.

4.2.4.3 Halaman Unggah File

Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-003, yaitu Unggah File. Halaman antarmuka Unggah File menampilkan halaman untuk pengguna dapat mengunggah file ke sistem. Tampilan implementasi halaman unggah file dapat dilihat pada gambar 4.32.

4.2.4.4 Halaman Melihat File

Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-004, yaitu Melihat File. Halaman ini menampilkan data file yang telah diunggah oleh pengguna dan data file yang dibagikan kepada dirinya. tampilan implementasi halaman melihat file dapat dilihat pada gambar 4.33.

Sistem Penyimpanan File Login Register

Register

Name

E-Mail Address

Password

Confirm Password

Gambar 4.30 Halaman Registrasi

Tugas Akhir :)

admin@storingits.com

MAIN NAVIGATION

- Home
- My Files
- Upload File
- Log
- File Sharing

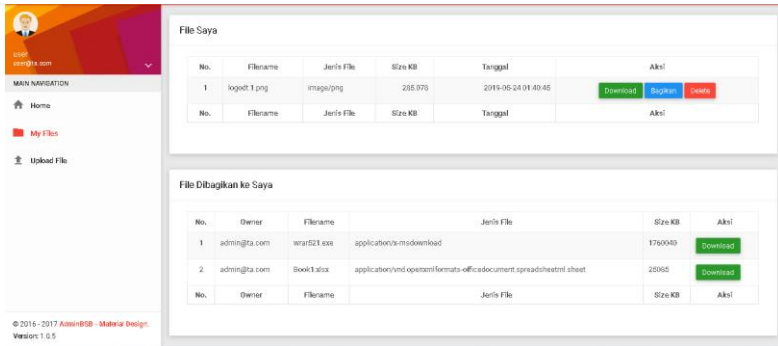
© 2016 - 2017 Admin@StoringIts - Material Design
Version: 1.0.5

Upload New File

File

No file chosen

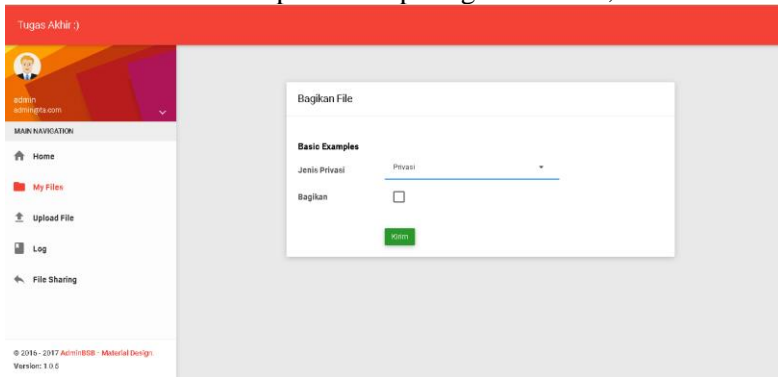
Gambar 4.31 Halaman Unggah File



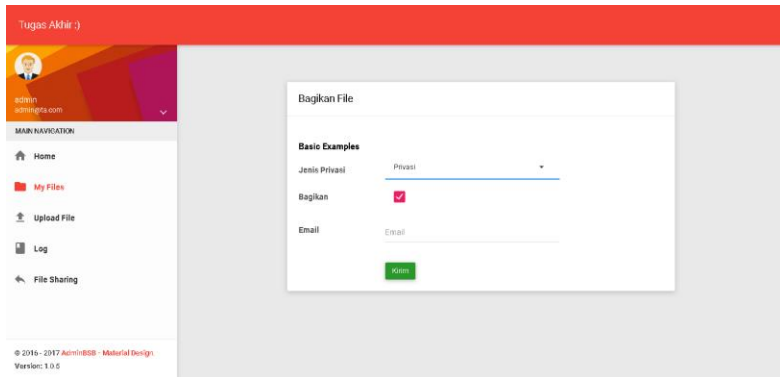
Gambar 4.32 Halaman File Saya

4.2.4.5 Halaman Berbagi File

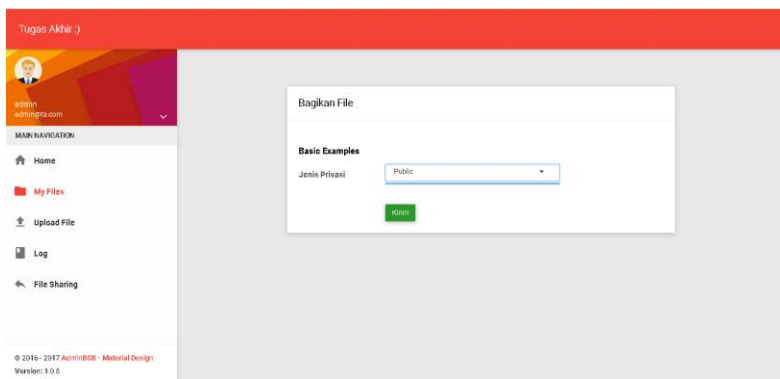
Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-007, yaitu Berbagi File. Halaman ini menampilkan menu yang membuat pengguna dapat membagikan filenya kepada pengguna yang lain. Tampilan implementasi halaman melihat file dapat dilihat pada gambar 4.34, 4.35 dan 4.36.



Gambar 4.33 Halaman Berbagi File 1



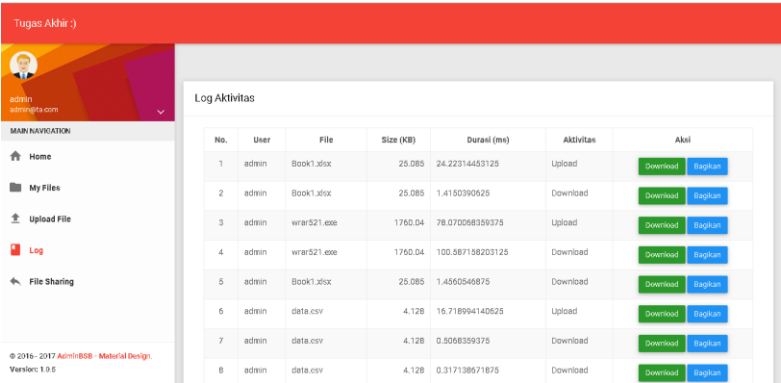
Gambar 4.34 Halaman Berbagi File 2



Gambar 4.35 Halaman Berbagi File 3

4.2.4.6 Halaman Melihat Log Aktivitas

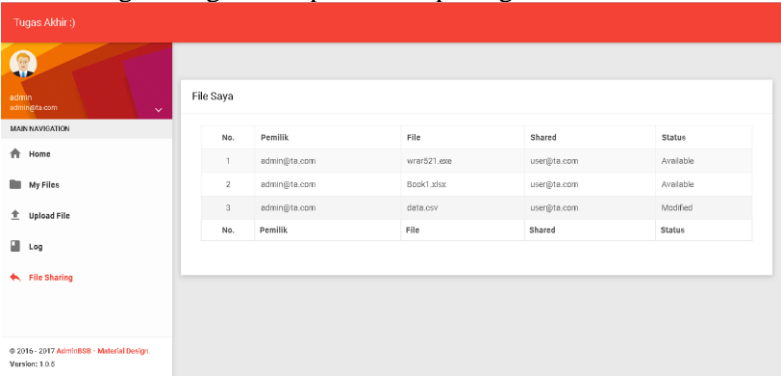
Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-008, yaitu Melihat Log Aktivitas. Halaman ini menampilkan data log aktivitas yang telah dilakukan oleh semua pengguna. Tampilan implementasi halaman melihat file dapat dilihat pada gambar 4.37



Gambar 4.36 Halaman Melihat Log

4.2.4.7 Halaman Melihat Log Berbagi File

Halaman ini merupakan halaman yang digunakan untuk kasus penggunaan UC-008, yaitu Melihat Log Berbagi File. Halaman ini menampilkan data log berbagi file yang telah dilakukan oleh semua pengguna. Tampilan implementasi halaman melihat log berbagi file dapat dilihat pada gambar 4.38.



Gambar 4.37 Halaman Melihat Log Berbagi File

4.3 Implementasi Server Backup

Pada server backup ini akan diberikan sebuah *script* yang memiliki fungsi sebagai pemulihan file yang telah hilang atau termodifikasi. *Script* yang digunakan ini dibangun menggunakan bahasa pemrograman Python. *Script* ini berjalan setiap lima menit

pada server backup. *Pseudocode script* ini dapat dilihat pada gambar 4.39.

1. GET data from file model
2. FOR row in record THEN
3. GET file from storage
4. SEND file to aplication server

Kode Sumber 4.4 *Pseudocode Script* Pengembalian File

4.4 Implementasi Server Basis Data

Pada server basis data ini akan diberikan sebuah *script* yang memiliki fungsi untuk mencadangkan basis data dan mengirimkan cadangan basis data tersebut kepada server backup. *script* ini akan berjalan setiap satu jam pada server basis data. *Pseudocode script* ini dapat dilihat pada gambar 4.40.

Server basis data ini akan melakukan pencadangan basis data dan pengiriman file cadangan basis data. Pencadangan basis data dilakukan menggunakan sebuah *command* di terminal dan pengiriman file cadangan basis data dilakukan dengan menjalankan *script* yang menggunakan bahasa pemrograman Python. *Command* untuk pencadangan basis adalah berikut:

```
mysqldump database_name > /directory/to/save/name_file.sql
```

1. GET Backup file
2. SEND Backup file to backup server

Kode Sumber 4.5 *Pseudocode* Pengiriman File Backup ke Server Backup

(Halaman ini sengaja dikosongkan)

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas mengenai uji coba yang dilakukan dan evaluasi sesuai dengan rancangan dan implementasi. Dari hasil yang didapatkan setelah melakukan uji coba, akan dilakukan evaluasi sehingga dapat diambil kesimpulan untuk bab selanjutnya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba sistem ini terdiri dari beberapa komponen yaitu server aplikasi, server backup dan server basis data. Server yang digunakan menggunakan layanan *Virtual Private Server* dari DigitalOcean. Spesifikasi dari masing-masing komponen ditunjukkan pada table [].

Tabel 5.5 Spesifikasi Masing-Masing Server

Komponen	Spesifikasi
CPU	Intel(R) Xeon(R) Gold 6140 CPU @ 2.30GHz
Sistem Operasi	Ubuntu Bionic 16.04 LTS 64 bit
Memori	RAM 1 GB
Penyimpanan	SSD 25 GB

5.2 Skenario Uji Coba

Pengujian ini terdiri dari dua jenis yaitu pengujian fungsional sistem dan pengujian performa sistem. Pengujian dilakukan di server dengan spesifikasi server yang dapat dilihat pada tabel 5.5.

Uji coba ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi terhadap kebutuhan sistem benar-benar telah diimplementasikan dan bekerja seperti yang seharusnya. Skenario pengujian dibedakan menjadi dua bagian yaitu skenario uji fungsionalitas dan skenario uji performa.

5.2.1 Skenario Uji Fungsionalitas

Uji fungsionalitas ini terdiri dari lima pengujian yaitu Pengujian Mengunggah File, Uji Fungsionalitas Pengguna Mengunduh File, Uji Fungsionalitas Verifikasi File, Uji Fungsionalitas Pengembalian File dan Uji Fungsionalitas Pencadangan Basis Data.

5.2.1.1 Uji Fungsionalitas Pengguna Mengunggah File

Uji coba ini dilakukan oleh pengguna dengan mengunggah suatu file ke sistem. Rancangan pengujian dan hasil yang diharapkan ditunjukkan pada tabel 5.6.

Tabel 5.6 Skenario Uji Fungsionalitas Pengguna Mengunggah File

Uji Coba	Harapan
Pengguna mengunggah file ke sistem	File pengguna dienkripsi dan tersimpan di server aplikasi
	File terenkripsi pengguna tersimpan di server backup
	Data file dan aktivitas pengguna tersimpan pada basis data

5.2.1.2 Uji Fungsionalitas Pengguna Mengunduh File

Uji coba ini dilakukan oleh pengguna dengan mengunduh file dari sistem. Rancangan pengujian dan hasil yang diharapkan ditunjukkan pada tabel 5.7.

Tabel 5.7 Skenario Fungsionalitas Pengguna Mengunduh File

Uji Coba	Harapan
Pengguna mengunduh file dari sistem	File yang terunduh adalah file yang sama dengan yang diunggah oleh pengguna, tidak mengalami perubahan ataupun rusak
	Data aktivitas pengguna tersimpan pada basis data

5.2.1.3 Uji Fungsionalitas Verifikasi File

Uji coba ini dilakukan dengan menjalankan *script* yang menggunakan bahasa pemrograman Python yang dijadwalkan berjalan secara otomatis setiap lima menit. Rancangan pengujian dan hasil yang diharapkan ditunjukkan pada tabel 5.8.

Tabel 5.8 Skenario Uji Fungsionalitas Verifikasi File

Uji Coba	Harapan
Menjalankan <i>script</i> verifikasi file di server aplikasi	Data file pada basis data akan diperbarui apabila file yang diverifikasi pada penyimpanan server aplikasi mengalami modifikasi ataupun hilang
	Data aktivitas sistem tersimpan pada basis data

5.2.1.4 Uji Fungsionalitas Pengembalian File

Uji coba ini dilakukan dengan menjalankan *script* yang menggunakan bahasa pemrograman Python yang dijadwalkan berjalan secara otomatis setiap lima menit. Rancangan pengujian dan hasil yang diharapkan ditunjukkan pada tabel 5.9.

Tabel 5.9 Skenario Uji Fungsionalitas Pengembalian File

Uji Coba	Harapan
Menjalankan <i>script</i> pengembalian file di server backup	File yang telah hilang pada server aplikasi dikembalikan
	Data file yang telah dikembalikan pada server aplikasi akan diperbarui pada basis data
	Data aktivitas sistem tersimpan pada basis data

5.2.1.5 Uji Fungsionalitas Pencadangan Basis Data

Uji coba ini dilakukan dengan menjalankan *script* yang berfungsi untuk mencadangkan basis data dan *script* yang berfungsi untuk mengirimkan hasil pencadangan basis data dari server basis data ke server backup. Rancangan pengujian dan hasil yang diharapkan ditunjukkan pada tabel 5.10.

Tabel 5.10 Skenario Uji Fungsionalitas

No	Uji Coba	Harapan
1	Menjalankan <i>script</i> pencadangan basis data	Basis data dicadangkan dan tersimpan pada server basis data
2	Menjalankan <i>script</i> pengiriman cadangan basis data	Data yang dicadangkan terimpan pada server backup

5.2.2 Skenario Uji Performa

Uji performa ini dilakukan untuk menghitung waktu yang dibutuhkan oleh sistem untuk melakukan enkripsi pada file yang diunggah dan dekripsi untuk file yang diunduh. Uji performa dilakukan oleh tiga pengguna dengan masing-masing pengguna melakukan pengunggahan dan pengunduhan file dengan jenis file yang ditunjukkan pada tabel 5.11

Tabel 5.11 Jenis File yang Digunakan untuk Skenario Uji Performa

No.	Jenis File	Format File
1	Gambar	JPEG
2	Audio	WAV
3	Video	MP4
4	Dokumen	DOCX

5.3 Hasil Uji Coba dan Evaluasi

Berikut dijelaskan hasil uji coba dan ecaluasi berdasarkan skenario yang suda dijelaskan pada bab 5.2.

5.3.1 Uji Fungsionalitas

Berikut dijelaskan hasil pengujian pada sistem yang sudah dibangun.

5.3.1.1 Uji Fungsionalitas Pengguna Mengunggah File

Uji coba ini dilakukan oleh pengguna dengan mengunggah suatu file ke sistem. Hasil uji coba dapat dilihat pada tabel 5.12.

Tabel 5.12 Hasil Uji Fungsionalitas Pengguna Mengunggah File

Uji Coba	Harapan	Hasil
Pengguna mengunggah file ke sistem	File pengguna dienkripsi dan tersimpan di server aplikasi	OK
	File terenkripsi pengguna tersimpan di server backup	OK
	Data file dan aktivitas pengguna tersimpan pada basis data	OK

5.3.1.2 Uji Fungsionalitas Pengguna Mengunduh File

Uji coba ini dilakukan oleh pengguna dengan mengunduh file dari sistem. Hasil uji coba dapat dilihat pada tabel 5.13.

Tabel 5.13 Hasil Uji Fungsionalitas Pengguna Mengunduh File

Uji Coba	Harapan	Hasil
Pengguna mengunduh file dari sistem	File yang terunduh adalah file yang sama dengan yang diunggah oleh pengguna, tidak mengalami perubahan ataupun rusak	OK
	Data aktivitas pengguna tersimpan pada basis data	OK

5.3.1.3 Uji Fungsionalitas Verifikasi File

Uji coba ini dilakukan dengan menjalankan *script* yang menggunakan bahasa pemrograman Python yang dijadwalkan berjalan secara otomatis setiap lima menit. Hasil uji coba dapat dilihat pada tabel 5.14.

Tabel 5.14 Hasil Uji Fungsionalitas Verifikasi File

Uji Coba	Harapan	Hasil
Menjalankan <i>script</i> verifikasi file di server aplikasi	Data file pada basis data akan diperbarui apabila file yang diverifikasi pada penyimpanan server aplikasi mengalami modifikasi ataupun hilang	OK
	Data aktivitas sistem tersimpan pada basis data	OK

5.3.1.4 Uji Fungsionalitas Pengembalian File

Uji coba ini dilakukan dengan menjalankan *script* yang menggunakan bahasa pemrograman Python yang dijadwalkan berjalan secara otomatis setiap lima menit. Hasil uji coba dapat dilihat pada tabel 5.15.

Tabel 5.15 Hasil Uji Fungsionalitas Pengembalian File

Uji Coba	Harapan	Hasil
Menjalankan <i>script</i> pengembalian file di server backup	File yang telah hilang pada server aplikasi dikembalikan	OK
	Data file yang telah dikembalikan pada server aplikasi akan diperbarui pada basis data	OK
	Data aktivitas sistem tersimpan pada basis data	OK

5.3.1.5 Uji Fungsionalitas Pencadangan Basis Data

Uji coba ini dilakukan dengan menjalankan *script* yang berfungsi untuk mencadangkan basis data dan *script* yang berfungsi untuk mengirimkan hasil pencadangan basis data dari server basis data ke server backup. Hasil uji coba dapat dilihat pada tabel 5.16.

Tabel 5.16 Hasil Uji Fungsionalitas Pencadangan Basis Data

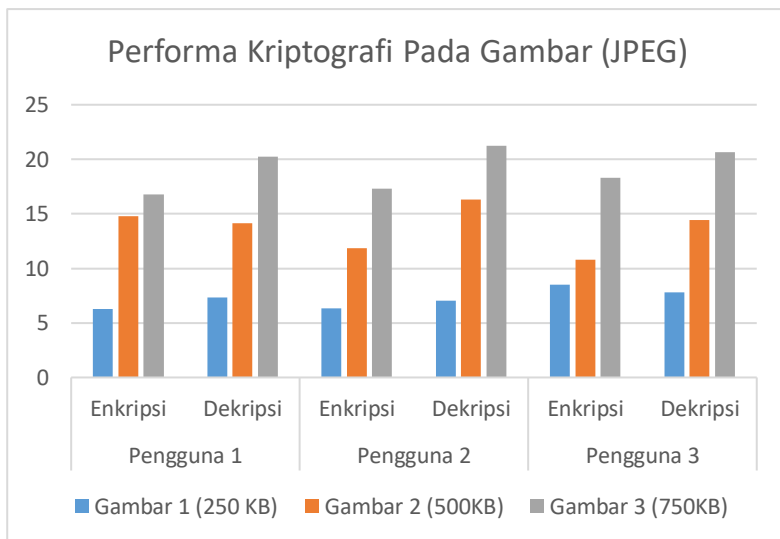
No	Uji Coba	Harapan	Hasil
1	Menjalankan <i>script</i> pencadangan basis data	Basis data dicadangkan dan tersimpan pada server basis data	OK
2	Menjalankan <i>script</i> pengiriman cadangan basis data	Data yang dicadangkan terimpan pada server backup	OK

5.3.2 Uji Performa

Seperti dijelaskan pada bab 5.2.2 pengujian performa akan dilakukan oleh pengguna dengan cara melakukan pengunggahan dan pengunduhan pada sistem untuk mengetahui waktu yang diperlukan sistem untuk melakukan enkripsi dan dekripsi pada file.

5.3.2.1 Performa Kriptografi Terhadap Jenis File Gambar

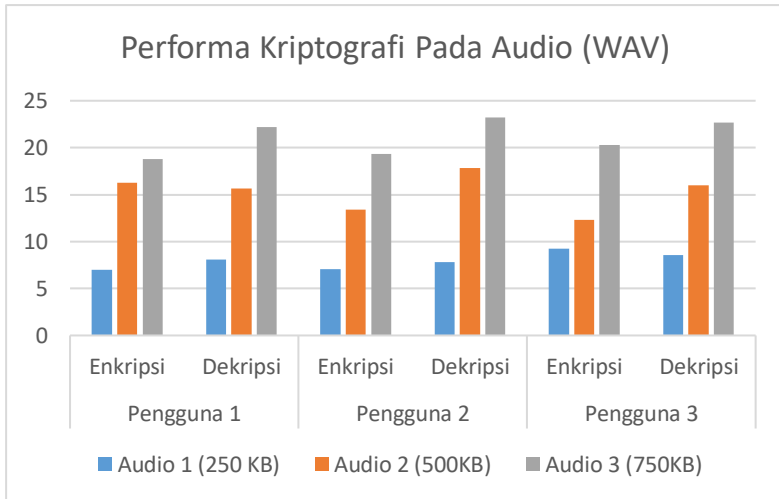
Setelah melakukan pengunggahan dan pengunduhan terhadap file berjenis gambar, dapat dilihat grafik performa kriptografi sistem terhadap file berjenis gambar yang dapat dilihat pada gambar 5.36.



Gambar 5.38 Performa Kriptografi pada File Gambar

5.3.2.2 Performa Kriptografi Terhadap Jenis File Audio

Setelah melakukan pengunggahan dan pengunduhan terhadap file berjenis audio, dapat dilihat grafik performa kriptografi sistem terhadap file berjenis audio yang dapat dilihat pada gambar 5.37.



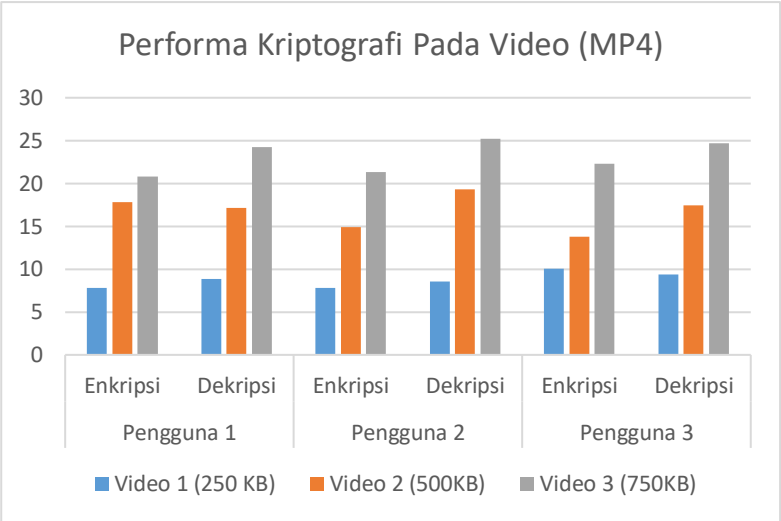
Gambar 5.39 Performa Kriptografi pada File Audio

5.3.2.3 Performa Kriptografi Terhadap Jenis File Video

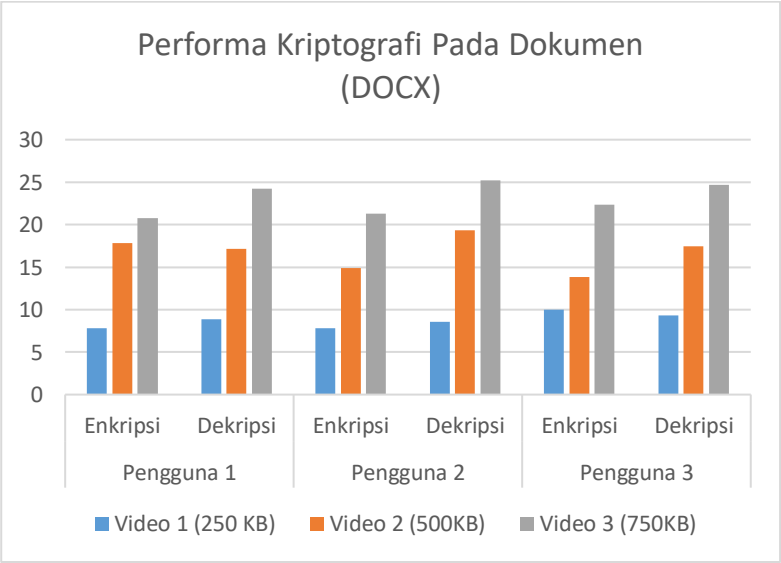
Setelah melakukan pengunggahan dan pengunduhan terhadap file berjenis video, dapat dilihat grafik performa kriptografi sistem terhadap file berjenis video yang dapat dilihat pada gambar 5.38.

5.3.2.4 Performa Kriptografi Terhadap Jenis File Dokumen

Setelah melakukan pengunggahan dan pengunduhan terhadap file berjenis dokumen, dapat dilihat grafik performa kriptografi sistem terhadap file berjenis dokumen yang dapat dilihat pada gambar 5.39.



Gambar 5.40 Performa Kriptografi pada File Video



Gambar 5.41 Performa Kriptografi pada File Dokumen

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang diperoleh dari tugas akhir yang telah dikerjakan dan saran terkait pengembangan dari tugas akhir ini yang dapat dilakukan pada masa yang akan datang.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari hasil uji coba dan evaluasi pada tugas akhir ini adalah sebagai berikut:

1. Penelitian ini telah mengaplikasikan algoritma enkripsi AES-256 pada aplikasi berbasis *website* dengan kerangka kerja Laravel
2. Sistem dapat melakukan verifikasi integritas file untuk memastikan bahwa file yang tersimpan tidak mengalami modifikasi
3. Sistem dapat mengembalikan data yang telah hilang atau termodifikasi pada server cloud

6.2 Saran

Saran yang diberikan dari hasil uji coba dan evaluasi pada tugas akhir ini adalah sebagai berikut:

1. Untuk meningkatkan kredibilitas data pada sistem diperlukan verifikasi pada basis data.
2. Menambahkan server untuk server basis data dan server backup untuk replikasi data.
3. Untuk meningkatkan keamanan

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] S. Shaikh and D. Vora, "Secure Cloud Auditing Over Encrypted Data," p. 5, 2016.
- [2] S. F. O. E.-H. David B Little, Digital Data Integrity: The Evolution from Passive Protection to Active Management, San Fransisco: John Wiley & Sons, Ltd, 2007.
- [3] R. D. McDowall, Data Integrity and Data Governance: Practical Implementation in Regulated Laboratories, Croydon: CPI Group (UK) Ltd, 2018.
- [4] Y. Yudhanto, Panduan Mudah Belajar Framework Laravel, Jakarta: Gramedia, 2018.
- [5] S. Anhar, PHP & MySql Secara Otodidak, Jakarta: Mediakita, 2010.
- [6] Jubilee Enterprise, Python untuk Programmer Pemula, Elexmedia Komputindo, 2019.
- [7] W. Stallings, The Advanced Encryption Standard., San Jose: The Internet Protocol Journal, 2001.
- [8] V. Yuniati, Enkripsi Dan Dekripsi Dengan Algoritma Aes 256 Untuk Semua Jenis File, Jurnal Informatika Universitas Kristen Duta Wacana, 2009.
- [9] R. Munir, Digital Signature Standard (DSS), Bandung: Institut Teknologi Bandung, 2004.
- [10] A. Y. Insani, Proteksi Akses File Executable Menggunakan Sistem Keamanan Teknologi USB Flash Disk, Bandung: Universitas Komputer Indonesia, 2008.
- [11] F. Rodriguez-Henriquez, Cryptographic Algorithms on Reconfigurable Hardware, New York: Springer, 2006.

- [12] A. Sebastian, Implementasi dan Perbandingan Performa Algoritma *Hash* SHA-1, SHA-256 dan SHA-512, Bandung: Institut Teknologi Bandung, 2007.
- [13] R. d. N. S. I. Mankar, Implementation of SHA-256 Algorithm, Pune: Pune University, 2013.

LAMPIRAN

1. Fungsi Unggah File ke Sistem

```
1. public function upload(Request $request){
2.     $this->validate($request, [
3.         'file' => 'required|file|max:2048', // max
4.         2MB
5.     ]);
6.     //user
7.     $user = Auth::user()->id;
8.     //file
9.     $files = $request->file('file');
10.    $size = $request->file('file')->getSize();
11.    $lastFile = File::all()->last();
12.    if(!$lastFile) $id = 1;
13.    else $id = $lastFile->id + 1;
14.    $filename = $id.'.dat';
15.    $fileContent = $files->get();
16.    $title = $files->getClientOriginalName();
17.
18.    //generate key
19.    $calonKey = Auth::user()->id.Auth::user()-
20.    >name.Auth::user()->email.Auth::user()-
21.    >created_at.Auth::user()->updated_at;
22.    $ckey = sha1($calonKey);
23.    $key = substr($ckey, 0, 32);
24.
25.    //encrypt AES
26.    $start = microtime(true)*1000;
27.    $enc = new Encrypter($key, 'AES-256-CBC');
28.    $encryptedContent = $enc-
29.    >encrypt($fileContent);
30.
31.    //store encrypted
32.    Storage::put($filename, $encryptedContent);
33.    $time = microtime(true)*1000 - $start;
34.    $fileOri = $files->getClientOriginalName();
35.    $ext = $files->getClientMimeType();
36.    $path = '/';
```

```

35. // generate checksum / sha256
36. $checksum = 'sha256sum ../storage/app' . $path . $filename;
37. $process = new Process($checksum);
38. $process->run();
39. if (!$process->isSuccessful()) {
40.     throw new ProcessFailedException($process);
41. }
42. $sha = explode(" ", $process->getOutput())[0];
43.
44. // transfer ke server backup
45. $scp = 'scp /var/www/html/Tugas-Akhir/storage/app/' . $filename . ' root@128.199.64.128:/var/www/html/backup/' . $filename;
46. $transfer = new Process($scp);
47. $transfer->run();
48. if (!$transfer->isSuccessful()) {
49.     throw new ProcessFailedException($transfer);
50. };
51.
52. // insert db
53. $file = File::create([
54.     'id_user' => $user,
55.     'filename' => $title,
56.     'stored' => $filename,
57.     'format' => $ext,
58.     'size' => $size,
59.     'path' => $path,
60.     'duration' => $time,
61.     'sha' => $sha,
62.     'privasi' => 0,
63.     'modif' => 0,
64.     'delete' => 0
65. ]);
66. $log = Log::create([
67.     'user_id' => $user,
68.     'file_id' => $id,
69.     'duration' => $time,
70.     'execution' => 1,
71. ]);
72.

```



```

73.     return redirect()
74.     ->back()
75.     -
    >withSuccess(sprintf('File %s has been uploaded.',
    $title));
76. }

```

2. Fungsi Unduh File pada Aplikasi

```

1.  public function Download($id){
2.      $file = File::find($id);
3.      if($file->delete)
4.          return redirect('myfile')-
    >withSuccess('File telah dihapus');
5.
6.      //ambil checksum pada file
7.      $checksum = 'sha256sum ../storage/app' . $file-
    >path.$file->stored;
8.      $process = new Process($checksum);
9.      $process->run();
10.     if (!$process->isSuccessful()) {
11.         throw new ProcessFailedException($process);
12.     }
13.     $sha = explode(" ", $process->getOutput())[0];
14.
15.     //Jika sha tidak sama (ada perubahan data)
16.     if($file->sha != $sha){
17.         $file->modif = 1;
18.         $file->save();
19.
20.         return redirect()
21.             ->back()
22.             -
    >withSuccess(sprintf('File telah dimodifikasi.'));
23.     }
24.     // ambil encrypted file
25.     $encryptedContent = Storage::get($file-
    >stored);
26.     //generate key
27.     $owner = User::find($file->id_user);

```

```

28.     $calonKey = $owner->id.$owner->name.$owner->
    >email.$owner->created_at.$owner->updated_at;
29.     $key = sha1($calonKey);
30.     $key = substr($key, 0, 32);
31.
32.     //aes
33.     $start = microtime(true)*1000;
34.     $enc = new Encrypter($key, 'AES-256-CBC');
35.     $decryptedContent = $enc->
    >decrypt($encryptedContent);
36.     $time = microtime(true)*1000 - $start;
37.
38.     $log = Log::create([
39.         'user_id' => Auth::user()->id,
40.         'file_id' => $id,
41.         'duration' => $time,
42.         'execution' => 2
43.     ]);
44.     ob_end_clean();
45.     return response()-
    >make($decryptedContent, 200, array(
46.         'Content-
    Type' => (new finfo(FILEINFO_MIME))-
    >buffer($decryptedContent),
47.         'Content-
    Disposition' => 'attachment; filename="'. pathinfo
    ($file->filename, PATHINFO_BASENAME) . '"
48.     ));
49. }

```

3. Script Verifikasi File

```

1. import os
2. import mysql.connector
3. from mysql.connector import Error
4. from mysql.connector import errorcode
5. import datetime;
6. try:

```

```

7.     mySQLconnection = mysql.connector.connect(host='
178.128.80.206',database='skripsi',user='buaya',pas
sword='sembarang12')
8.     sql_select_Query = "select * from files"
9.     cursor = mySQLconnection.cursor()
10.    cursor.execute(sql_select_Query)
11.    records = cursor.fetchall()
12.    for row in records:
13.        file = 'storage/app/' + row[2]
14.        sha = row[9]
15.
16.        is_exist = os.path.isfile(file)
17.        # if file tidak ada
18.        if(is_exist == False):
19.            update = "update files set lost = 1 where id
= " + str(row[0])
20.            cursor.execute(update)
21.            ts = datetime.datetime.now().timestamp()
22.            time = datetime.datetime.fromtimestamp(ts).is
oformat()
23.            new_time = time.replace('T', ' ')
24.            time = new_time[:19]
25.            insert = "insert into logs (`user_id`, `file_
id`, `execution`, `duration`, `created_at`) values
(0, "+str(row[0])+", 4, '-', ' ' + str(time)+"")"
26.            cursor.execute(insert)
27.            mySQLconnection.commit()
28.            # if file ada
29.            else:
30.                checksum = "sha256sum " + file
31.                process = os.popen(checksum).read()
32.                cc = process.split()[0]
33.                if(cc != sha):
34.                    update = "update files set modif = 1 where
id = " + str(row[0])
35.                    cursor.execute(update)
36.                    ts = datetime.datetime.now().timestamp()
37.                    time = datetime.datetime.fromtimestamp(ts).
isoformat()
38.                    new_time = time.replace('T', ' ')
39.                    time = new_time[:19]
40.                    insert = "insert into logs (`user_id`, `fil
e_id`, `execution`, `duration`, `created_at`) value

```

```

        s (0, "+str(row[0])+", 5, '-
        ', '" + str(time)+"")
41.         cursor.execute(insert)
42.         mysqlconnection.commit()
43.         cursor.close()
44.     except Error as e :
45.         print ("Error while connecting to MySQL", e)
46. finally:
47.     #tutup koneksi database.
48.     if(mysqlconnection .is_connected()):
49.         mysqlconnection.close()
50.         print("MySQL connection is closed")

```

4. Script Pengembalian File

```

1. import os
2. import mysql.connector
3. from mysql.connector import Error
4. from mysql.connector import errorcode
5. import datetime
6. import paramiko
7. from paramiko import SSHClient
8. from scp import SCPClient
9.
10. def createSSHClient(server, port, user, password):
11.     client = paramiko.SSHClient()
12.     client.load_system_host_keys()
13.     client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
14.     client.connect(server, port, user, password)
15.     return client
16.
17. ssh = createSSHClient('157.230.42.44', '22', 'root'
18.     , 'indomie goreng123!')
19. scp = SCPClient(ssh.get_transport())
20. try:

```

```

21.     mySQLconnection = mysql.connector.connect(host=
        '178.128.80.206',database='skripsi',user='buaya',pa
        ssword='sembarang12')
22.     sql_select_Query = "select * from files"
23.     cursor = mySQLconnection.cursor()
24.     cursor.execute(sql_select_Query)
25.     records = cursor.fetchall()
26.     for row in records:
27.         # if file dihapus
28.         file = str(row[2])
29.         if(row[10] == 1):
30.             src = '/var/www/html/backup/' + str(row
[2])
31.             dst = '/var/www/html/Tugas-
Akhir/storage/app/'
32.             scp.put(src, dst)
33.             ts = datetime.datetime.now().timestamp(
)
34.             time = datetime.datetime.fromtimestamp(
ts).isoformat()
35.             new_time = time.replace('T', ' ')
36.             time = new_time[:19]
37.             update = "update files set `modif` = 0
where id = " + str(row[0])
38.             cursor.execute(update)
39.             insert = "insert into logs (`user_id`,
`file_id`, `execution`, `duration`, `created_at`) v
alues (0, "+str(row[0])+", 6, '-
', "" + str(time)+"")"
40.             cursor.execute(insert)
41.             mySQLconnection.commit()
42.             # if file mod
43.             elif(row[12] == 1):
44.                 src = '/var/www/html/backup/' + str(row
[2])
45.                 dst = '/var/www/html/Tugas-
Akhir/storage/app/'
46.                 scp.put(src, dst)
47.                 ts = datetime.datetime.now().timestamp(
)
48.                 time = datetime.datetime.fromtimestamp(
ts).isoformat()
49.                 new_time = time.replace('T', ' ')

```

```

50.         time = new_time[:19]
51.         update = "update files set `delete` = 0
           where id = " + str(row[0])
52.         cursor.execute(update)
53.         insert = "insert into logs (`user_id`,
           `file_id`, `execution`, `duration`, `created_at`) v
           alues (0, "+str(row[0])+", 7, '-
           ', '" + str(time)+"'"
54.         cursor.execute(insert)
55.         mysqlconnection.commit()
56.     cursor.close()
57. except Error as e :
58.     print ("Error while connecting to MySQL", e)
59. finally:
60.     #closing database connection.
61.     if(mysqlconnection .is_connected()):
62.         mysqlconnection.close()
63.         print("MySQL connection is closed")

```

5. Script Mencadangkan Basis Data

```

1. #!/bin/bash
2. mysqldump -u root --
   password=buayakecil skripsi > /var/www/html/backup
   /backup_ta_$(date +%d%m%y_%H:%M).sql

```

6. Script Pengiriman Cadangan Basis Data

```

1. import os
2. import paramiko
3. from paramiko import SSHClient
4. from scp import SCPClient
5.
6. # membuat SSH client
7. def createSSHClient(server, port, user, password):
8.
9.     client = paramiko.SSHClient()
10.    client.load_system_host_keys()
11.    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
12.    client.connect(server, port, user, password)

```

```
12.     return client
13.
14. ssh = createSSHClient('128.199.64.128', '22', 'root
    ', 'daniarggk')
15. scp = SCPClient(ssh.get_transport())
16. src = '/var/www/html/backup/'
17. dst = '/var/www/html/backup/sql/'
18. scp.put(src, recursive=True, remote_path=dst)
```

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Muhammad Fajri Salam lahir di Bojonegoro pada tanggal 28 September 1996. Penulis menempuh pendidikan formal di TK Bustanul Athfal Sumberrejo (2001-2003), MI Muhammadiyah Sumberrejo Bojonegoro (2003-2009), SMP Plus Ar-Rahmat Bojonegoro (2009-2012), SMAN Model Terpadu Bojonegoro (2012-2015), dan Informatika ITS Surabaya (2015-2019). Bidang studi yang diambil oleh penulis saat berkuliah di Departemen Informatika ITS adalah Arsitektur Jaringan Komputer (AJK). Penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer-Informatika 2017-2018 di Departemen Kesejahteraan Mahasiswa dan Keluarga Muslim Informatika 2017-2018 di Departemen Keilmuan. Penulis juga aktif dalam kegiatan kepanitiaan seperti SCHEMATICS 2016-2017 divisi Kamzin, Kegiatan Mentoring dari Keluarga Muslim Informatika 2016-2017 dan Kegiatan Mentoring dari Jamaah Masjid Manarul Ilmi 2017-2018. Penulis juga pernah menjadi admin dan *developer* di *admindt.net* dan menjadi *developer* *ppdbriau.net*. Penulis dapat dihubungi melalui nomor handphone 08970427472 atau melalui email fajrisalam289@gmail.com

