PERANCANGAN APLIKASI KEAMANAN BANK SOAL UJIAN SEKOLAH MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL BERBASIS *WEB*

SKRIPSI

DISUSUN OLEH

MAHMUBIN HAIBAH

NPM. 2109010050

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2025

PERANCANGAN APLIKASI KEAMANAN BANK SOAL UJIAN SEKOLAH MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL BERBASIS *WEB*

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

MAHMUBIN HAIBAH
NPM. 2109010050

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025

LEMBAR PENGESAHAN

Judul Skripsi	:	PERANCANGAN APLIKASI KEAMANAN BANK SOAL UJIAN SEKOLAH MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL BERBASIS *WEB*
Nama Mahasiswa	:	MAHMUBIN HAIBAH
NPM	:	2109010050
Program Studi	:	SISTEM INFORMASI

Menyetujui
Komisi Pembimbing

(Ferdy Riza, S. T., M.Kom)
NIDN. 0103068901

Ketua Program Studi

(Martiano, S.Kom., M.Kom)
NIDN. 0128029302

Dekan

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

ii

PERNYATAAN ORISINALITAS

PERANCANGAN APLIKASI KEAMANAN BANK SOAL UJIAN SEKOLAH MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL BERBASIS *WEB*

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 21 April 2025

Yang membuat pernyataan

Mahmubin Haibah

NPM. 2109010050

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Mahmubin Haibah
NPM	: 2109010050
Program Studi	: Sistem Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

PERANCANGAN APLIKASI KEAMANAN BANK SOAL UJIAN SEKOLAH MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL BERBASIS *WEB*

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 21 April 2025

Yang membuat pernyataan

Mahmubin Haibah

NPM. 2109010050

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Mahmubin Haibah
Tempat dan Tanggal Lahir	: Medan, 30 Januari 2003
Alamat Rumah	: Jl.Sempurna, Gg. Melati38, Pasar 7 Tembung, Deli Serdang, Sumatera Utara
Telepon/Faks/HP	: 089520889073
E-mail	: sayahaibah@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SDN.104204	TAMAT: 2014/2015
SMP	: Madrasah Tsanawiyah Al - Jam'iyatul Washliyah Tembung	TAMAT: 2017/2018
SMA	: SMAS Cerdas Murni	TAMAT: 2020/2021

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Perancangan Aplikasi Keamanan Bank Soal Ujian Sekolah Menggunakan Algoritma Kriptografi Elgamal Berbasis *Web*."**
Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW, yang telah membawa kita dari zaman kegelapan menuju zaman yang terang benderang dengan ilmu pengetahuan.

Penyusunan skripsi ini ditujukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer di Universitas Muhammadiyah Sumatera Utara. Penulis menyadari bahwa penyusunan skripsi ini tidak akan terselesaikan tanpa bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam proses penyusunan skripsi ini.

Ucapan terima kasih khusus penulis sampaikan kepada Allah SWT dan kepada kedua orang tua tercinta, ayah **H. Yudi Handoyo** dan ibu **Supatmi**, yang telah membesarkan, mendidik, serta memberikan cinta dan doa yang tiada henti. Terima kasih juga kepada saudara-saudara penulis yang selalu memberi dukungan serta semangat.

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., selaku Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom., selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Bapak Martiano, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Informasi

4. Ibu Yoshida Sary, S.E., S.Kom., M.Kom., selaku Sekretaris Program Studi Sistem Informasi.
5. Bapak Ferdy Riza, ST, M.Kom selaku Dosen Pembimbing yang telah memberikan arahan dan bimbingan kepada penulis.
6. Penghargaan khusus saya sampaikan kepada Adelia Jaharani atas dukungan moral dan emosional yang sangat berarti. Kehadiran serta motivasi darinya menjadi salah satu faktor penting yang membantu saya menyelesaikan skripsi ini dengan baik.
7. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

Medan, 21 April 2025

Mahmubin Haibah

vii

PERANCANGAN APLIKASI KEAMANAN BANK SOAL UJIAN SEKOLAH MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL BERBASIS *WEB*

ABSTRAK

Keamanan data menjadi aspek krusial dalam sistem pendidikan, terutama dalam pengelolaan bank soal ujian sekolah. Kebocoran soal dapat mengganggu integritas proses evaluasi dan berdampak pada kredibilitas instansi pendidikan. Untuk menjawab tantangan tersebut, penelitian ini merancang sebuah aplikasi keamanan bank soal berbasis web dengan menerapkan keamanan algoritma kriptografi ElGamal. Algoritma ini dipilih karena kemampuannya dalam menjaga kerahasiaan data melalui proses enkripsi dan dekripsi serta pemilihan kunci publik dan kunci privat yang berbeda. Aplikasi ini memungkinkan user atau guru untuk mengunggah soal ujian dalam bentuk file berekstensi PDF yang kemudian dienkripsi menggunakan algoritma kriptogragi ElGamal, sehingga hanya pihak yang memiliki kunci privat yang dapat mengakses isi soal tersebut. Pengujian dilakukan untuk memastikan fungsionalitas aplikasi berjalan baik, mulai dari proses unggah, enkripsi, penyimpanan file serta kunci privat, hingga proses dekripsi file soal. Hasil implementasi menunjukkan bahwa sistem mampu mengamankan soal ujian secara efektif dan dapat diakses dengan mudah oleh pengguna yang berwenang. Dengan adanya sistem ini, diharapkan proses distribusi dan penyimpanan soal ujian menjadi lebih aman, efisien, dan terjaga kerahasiaannya.

Kata Kunci: keamanan data; bank soal; kriptografi ElGamal; enkripsi; dekripsi.

DEVELOPMENT OF A WEB-BASED APPLICATION FOR SECURING SCHOOL EXAM QUESTION BANKS USING THE ELGAMAL CRYPTOGRAPHIC ALGORITHM

ABSTRACT

Data security is a crucial aspect of the education system, particularly in the management of school exam question banks. Leakage of exam questions can compromise the integrity of the evaluation process and affect the credibility of educational institutions. To address this challenge, this study designs a web-based application for securing exam question banks by implementing the ElGamal cryptographic algorithm. This algorithm is chosen for its ability to maintain data confidentiality through encryption and decryption processes, as well as its use of distinct public and private keys. The application allows users or teachers to upload exam questions in PDF format, which are then encrypted using the ElGamal algorithm, ensuring that only those with the private key can access the contents. Testing was conducted to ensure that the application's functionalities work properly, from the upload process, encryption, file and private key storage, to the decryption of the exam file. The implementation results show that the system effectively secures exam questions and can be easily accessed by authorized users. With this system, the distribution and storage of exam questions are expected to be more secure, efficient, and confidential.

Keywords: data security; question bank; ElGamal cryptography; encryption; decryption.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam perkembangan sistem pendidikan saat ini, penggunaan bank soal pada suatu instansi pendidikan sebagai media untuk menyimpan kumpulan data soal ujian sekolah sangat penting dikarenakan peran bank soal dapat meningkatkan efisiensi dalam penyimpanan data soal ujian, serta menjadi tolak ukur perkembangan proses penilaian pada institusi pendidikan.

Bank soal merupakan kumpulan soal-soal dari berbagai materi yang terstruktur, untuk mempermudah penyusunan soal untuk ujian, bank soal terdiri dari beberapa soal dari berbagai materi yang teroganisir dimulai dari menata hingga menyimpan dengan merujuk pada materi yang sudah diberikan, seperti nomor soal, kunci soal, dan indeks soal (Qhorifadillah et al., 2022). Bank soal tidak hanya digunakan dalam bidang akademik tetapi dapat juga digunakan dalam disiplin ilmu lain. Secara umum, bank soal menawarkan keuntungan seperti fasilitasi yang mudah, hasil yang cepat, dan pelaksanaan yang relatif murah (biaya dapat ditekan) karena tidak memerlukan kertas kerja dan pemeriksaan hasil dapat dilakukan segera setelah dilakukan (Annas, 2020). Beberapa langkah penting dalam pengembangan bank soal termasuk penulisan soal, validasi, kalibrasi, penyimpanan dan pengamanan soal, hubungannya dengan bagian baru soal, dan mempertahankan bank soal. Namun, langkah-langkah ini tidak dilakukan secara keseluruhan. (Suhardi, 2023).

Dari penjelasan yang dikemukakan diatas, bank soal yang berisi data soal ujian sekolah dalam pengoperasiannya harus dikembangkan dengan memperhatikan aspek penulisan soal, validitasi, kalibrasi, penyimpanan dan aspek keamanan. Namun, dalam praktiknya, banyak institusi pendidikan masih menggunakan metode penyimpanan konvensional atau sistem digital tanpa perlindungan keamanan yang memadai, sehingga rentan terhadap ancaman *cybercrime* oleh pihak yang tidak bertanggung jawab dan berpotensi disalahgunakan seperti pencurian dan kebocoran soal sebelum proses ujian berlangsung. (Butarbutar, 2023) Menjelaskan bahwa *cybercrime* adalah segala kegiatan kriminal yang dilakukan menggunakan komputer dan jaringan komputer atau internet. *Cybercrime* berarti melakukan aktivitas ilegal, menargetkan korban, atau mengeksploitasi kerentanan dalam sistem digital.

Untuk mengatasi permasalahan tersebut, solusi yang dapat diterapkan adalah membangun sistem bank soal dengan mengimplementasikan algoritma kriptografi yang mampu mengamankan data soal ujian sekolah agar hanya dapat diakses oleh pihak yang berwenang. (Riza, F. et al., 2020) Menjelaskan bahwa kriptografi berasal dari bahasa Yunani *kryptos*, yang berarti tersembunyi, dan *graphein*, yang berarti menulis. Kriptografi secara harfiah berarti “menulis secara tersembunyi”. Kriptografi dalam teknologi modern adalah seni dan ilmu mengamankan data dengan menggunakan teknik pengkodean sehingga orang yang tidak berwenang tidak dapat mengakses informasi yang dikirim atau disimpan. Hanya mereka yang memiliki kunci yang diperlukan untuk mengurai data yang dilindungi oleh kriptografi.

Berdasarkan jenisnya, kriptografi dibagi menjadi dua jenis, yaitu: simetris dan asimetris. Kunci simetris menggunakan kunci yang sama dalam proses enkripsi dan dekripsi. Sedangkan kunci asimetris memiliki kunci yang berbeda dalam proses enkripsi dan dekripsi, proses enkripsi menggunakan kunci publik (*public key*) dan proses dekripsi menggunkan kunci privat (*private key*). Keamanan algoritma kriptografi asimetris lebih kuat dibanding algoritma simetris. (Saputro et al., 2020). Salah satu algoritma kriptografi asimetris yang dikenal adalah ElGamal. Algoritma kriptografi ElGamal merupakan kriptografi asimetris yang algoritmanya didasarkan atas masalah logaritma diskret. Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi (Riza, F. et al., 2020). Elgamal dikenal sangat kuat dalam hal mengenkripsi dan mendekripsi data. Kriptografi ElGamal lebih aman dibandingkan dengan kriptografi asimetri lain yaitu RSA karena menghasilkan *ciphertext* yang lebih kompleks namun juga lebih lambat karena proses enkripsi dapat menghasilkan lebih dari satu kunci publik (Mallouli et al., 2019).

Berdasarkan solusi yang telah dijelaskan, penulis akan membangun sistem bank soal dengan menggunakan algoritma kriptografi ElGamal. Metode ini dipilih karena memiliki tingkat keamanan yang tinggi dan tingkat kesulitan dalam memecahkan kunci privat dalam mengamankan data soal ujian sekolah. Oleh karena itu, penulis mengangkat penelitian ini dalam sebuah skripsi berjudul "**Perancangan Aplikasi Keamanan Bank Soal Ujian Sekolah Menggunakan Algoritma Kriptografi ElGamal Berbasis *Web*.**"

1.2. Rumusan Masalah

Rumusan masalah dari penelitian tersebut antara lain:

1. Bagaimana membangun sistem keamanan bank soal berbasis *web* yang dapat menjaga keamanan isi data soal ujian sekolah dari kejahatan *cybercrime*?
2. Bagaimana implementasi kriptografi ElGamal dalam mengamankan data soal ujian sekolah pada sistem bank soal yang akan dirancang?
3. Bagaimana proses *generate public key* dan *private key* dalam kemanan data soal ujian sekolah menggunakan metode kriptografi ElGamal?
4. Bagaimana membangun sistem bank soal yang dapat memudahkan *user* dan administrator dalam proses bertukar informasi data soal ujian sekolah?

1.3. Batasan Masalah

Adapun batasan masalah pada penelitian tersebut adalah:

1. Penelitian ini hanya membahas tentang algoritma kriptografi ElGamal yaitu meliputi proses enkripsi, *generate* kunci publik (*Public key*) dan kunci privat (*Private key*) serta proses dekripsi.
2. Penelitian ini berfokus pada merancangan program berbasis *web* dengan menggunakan bahasa program PHP versi 7 dan menggunakan *database* MySQL.
3. Penelitian ini berfokus pada pengamanan data soal ujian sekolah dengan format *file* PDF saja dan dengan maksimal ukuran data 1 *megabyte*.
4. Penelitian ini menggunakan 2 akses pengguna yaitu, administrator yang berwenang untuk melakukan manajemen data dan pengguna, serta *user* yang berwenang dalam melakukan pengunggahan data soal ujian sekolah.

1.4. Tujuan Penelitian

Adapun tujuan penelitian tersebut dibuat antara lain:

1. Membangun sistem keamanan bank soal berbasis *web* yang dapat menjadi wadah penyimpanan data soal ujian sekolah secara aman dari upaya pencurian data.
2. Merancang antarmuka yang ramah pengguna serta mempermudah pengelolaan soal ujian secara digital dengan tetap mengedepankan keamanan data.
3. Mengimplementasikan algoritma kriptografi ElGamal dengan sistem manajemen bank soal.

1.5. Manfaat Penelitian

Penelitian ini memiliki berbagai manfaat yang dapat dirasakan oleh berbagai pihak, baik pada sisi pengguna, publik dan penulis, terutama dalam konteks peningkatan efektivitas sistem bank soal dalam lingkup sekolah. Berikut adalah beberapa manfaat dari penelitian ini:

1.5.1. Manfaat Penelitian Untuk Pengguna

Manfaat penelitian tersebut untuk pengguna antara lain:

1. Memberikan kemudahan dan keamanan bagi pengguna dalam menyimpan data soal ujian sekolah.
2. Mempermudah administrator dalam mengelola dan mendistribusikan data soal ujian sekolah.
3. Menjaga keamanan data soal ujian sekolah untuk mengurangi risiko kebocoran sebelum ujian berlangsung.

1.5.2. Manfaat Penelitian Untuk Publik

Adapun manfaat penelitian ini untuk publik antara lain:

1. Menjadi referensi bagi pihak sekolah tentang pentingnya keamanan data dalam pengelolaan bank soal.
2. Meningkatkan kepercayaan pengguna dan siswa terhadap sistem ujian digital dengan penerapan algoritma kriptografi ElGamal.
3. Mempermudah pihak sekolah dalam melakukan manajemen dan pendistribusian soal ujian sekolah.
4. Pihak sekolah ikut serta mendukung inovasi dan transformasi digital dalam sistem pendidikan.

1.5.3. Manfaat Penelitian Untuk Penulis

Adapun manfaat penelitian ini untuk penulis antara lain:

1. Menambah wawasan dan pemahaman tentang keamanan data dalam sistem bank soal.
2. Meningkatkan keterampilan dalam mengembangkan sistem berbasis *web* yang aman dan efisien.
3. Memperdalam pengetahuan tentang penerapan kriptografi ElGamal dalam sistem digital.
4. Memberikan kontribusi bagi pengembangan sistem keamanan data dalam dunia akademik khususnya pada sektor pendidikan.

BAB II

LANDASAN TEORI

Berikut ini adalah beberapa landasan teori dari judul penelitian yang penulis kutip dari berbagai jurnal penelitian yang berkaitan dengan judul penelitian ini, antara lain:

2.1. Bank Soal

Secara singkat, bank soal dapat diartikan sebagai kumpulan dari beberapa soal. Namun, bank soal tidak hanya merupakan kumpulan dari beberapa soal saja, tetapi juga dapat diartikan pada proses pengumpulan soal, pemantauan dan mengelola informasi yang terkait sehingga mempermudah pendistribusiannya (Wardani et al., 2024).

Bank soal merupakan kumpulan soal-soal dari berbagai materi yang terstruktur, untuk mempermudah penyusunan soal untuk ujian, bank soal terdiri dari beberapa soal dari berbagai materi yang teroganisir dimulai dari menata hingga menyimpan dengan merujuk pada materi yang sudah diberikan, seperti nomor soal, kunci soal, dan indeks soal (Qhorifadillah et al., 2022).

2.2. Kriptografi

Kriptografi adalah ilmu yang mempelajari cara menyembunyikan pesan. Namun, pada era modern saat ini kriptografi adalah ilmu yang didasarkan pada teknik matematika untuk menangani keamanan informasi, termasuk kerahasiaan, keutuhan data dan otentikasi entitas. Akibatnya, definisi kriptografi saat ini tidak

hanya berkaitan dengan menyembunyikan pesan, tetapi juga mencakup berbagai metode untuk menjaga keamanan informasi tersebut (Nugraha, 2024).

Keamanan sistem informasi didasarkan pada lima aspek utama yang dikenal sebagai CIAAN: *Confidentiality* (Kerahasiaan), *Integrity* (Integritas), *Availability* (Ketersediaan), *Authentication* (Autentikasi), dan *Non-repudiation* (Nirpenyangkalan). Kerahasiaan memastikan bahwa informasi hanya dapat diakses oleh pihak berwenang melalui enkripsi dan kontrol akses. Integritas menjaga keakuratan data dengan metode seperti *checksum* dan *hashing* untuk mencegah manipulasi. Ketersediaan memastikan layanan tetap dapat diakses, meskipun ada serangan seperti DDoS (Riza, F. et al., 2020).

2.2.1. Keamanan

Keamanan adalah istilah yang digunakan berhubungan dengan kejahatan, dan segala bentuk kecelakaan. Keamanan adalah topik yang luas termasuk keamanan nasional terhadap ancaman teroris, keamanan komputer terhadap serangan *hacker*, keamanan rumah terhadap kajahatan maling dan ancaman lain dan banyak situasi berhubungan lainnya. Komputer yang terhubung pada suatu jaringan mempunyai ancaman keamanan lebih besar dari pada computer yang tidak terhubung dengan jaringan luar. Dengan mengendalikan *network security* resiko tersebut dapat dikurangi (Husaini et al., 2022).

Keamanan komputer adalah bidang teknologi yang dikenal sebagai keamanan informasi yang diterapkan pada komputer. Tujuan keamanan data termasuk melindungi data dari ancaman, pencurian, atau kehilangan nilainya, seperti yang dijelaskan dalam kebijakan keamanan data (Hidayat et al., 2023).

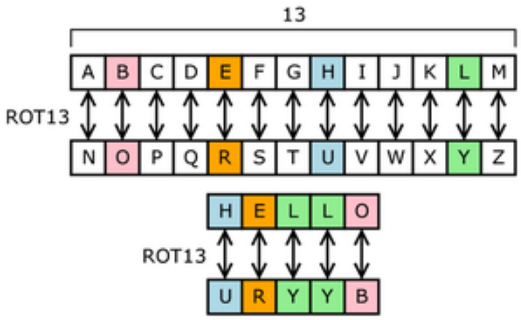
2.2.2. Jenis Kriptografi

Ada dua jenis algoritma kriptografi berdasarkan evolusi yang terjadi setiap tahun sejak kriptografi pertama kali ditemukan, yaitu:

1. Kriptografi Klasik

Sebelum era komputerisasi, kriptografi klasik telah digunakan yaitu dengan teknik substitusi atau transposisi, atau keduanya, digunakan untuk menyembunyikan pesan. Teknik substitusi menggunakan karakter *plaintext* untuk menggantikan karakter lainnya, yang menghasilkan *ciphertext*. Sementara itu, metode transposisi permutasi karakter *plaintext* menjadi *ciphertext*. Kombinasi keduanya cukup kompleks untuk membentuk algoritma kriptografi kontemporer (Ziaurrahman et al., 2019).

Kriptografi klasik digunakan sebelum atau sesudah penemuan komputer, tetapi tidak sepopuler saat ini. Ini hanya melakukan pengacakan huruf A hingga Z dan tidak disarankan untuk mengamankan informasi penting karena mudah dipecahkan dalam waktu yang singkat. Kriptografi klasik memberikan prinsip untuk menjaga keamanan kunci itu sendiri. Kriptografi klasik memiliki ciri yaitu: berbasis karakter, menggunakan pena dan kertas, termasuk dalam kiptografi kunci simetris (Amalya et al., 2023).

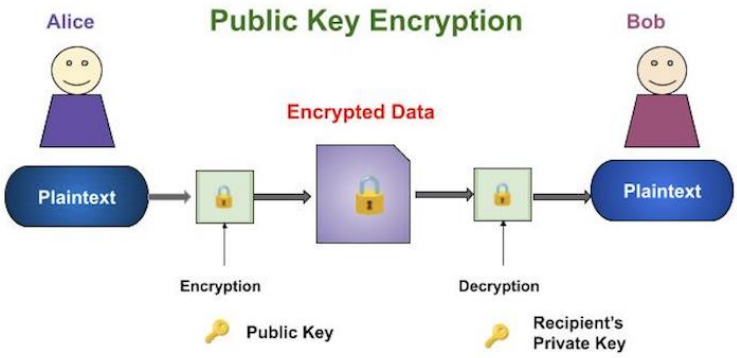


Gambar 2.1. Ilustrasi kriptografi klasik
(Sumber: *https://wikipedia.org*)

2. Kriptografi Modern

Algoritma kriptografi modern memiliki tingkat kesulitan yang kompleks. Kekuatan kriptografi bergantung pada kunci atau kuncinya. Karena kriptografi modern bergantung pada operasi komputer digital, pengolahan simbol biner digunakan, sehingga mempelajarinya membutuhkan dasar matematika (Ziaurrahman et al., 2019).

Merupakan modern merupakan kriptografi yang cukup rumit. Dibutuhkan pengetahuan matematika untuk menguasainya. Oleh karena itu kriptografi modern berkembang bersamaan dengan berkembangnya komputer hingga jaman sekarang. Terdiri dari 3 bagian, yaitu: algoritma Simetris, algoritma asimetris dan algoritma hibrida (Amalya et al., 2023).



Gambar 2.2. Ilustrasi kriptografi modern
(Sumber: *www.tutorialspoint.com*)

2.2.3. Pengertian Enkripsi

Enkripsi adalah salah satu komponen kunci dalam kriptografi, yaitu proses mengubah data asli (*plaintext*) menjadi bentuk terenkripsi (*ciphertext*) menggunakan algoritma enkripsi dan kunci enkripsi yang sesuai. Enkripsi

memainkan peran penting dalam menjaga kerahasiaan, integritas, dan keamanan data dalam keamanan komputer (Wijoyo et al., 2023).

Gambar 2.3 Ilustrasi Proses Enkripsi
(Sumber: *www.skyhighsecurity.com*)

Berikut ini adalah langkah-langkah yang digunakan dalam proses enkripsi antara lain:

1. Input Data yaitu asli yang akan dienkripsi disebut *plaintext*.
2. Kunci Enkripsi yaitu kunci yang digunakan untuk enkripsi adalah serangkaian bit yang harus dirahasiakan. Pengirim dan penerima harus memberikan kunci ini secara aman.
3. Algoritma Enkripsi yaitu proses enkripsi digunakan untuk mengubah teks biasa menjadi teks enkripsi menggunakan algoritma enkripsi simetris seperti AES, DES, atau *Blowfish*.
4. *Ciphertext* yaitu teks yang tampak acak dan tidak dapat dipahami tanpa kunci yang tepat adalah hasil dari proses enkripsi (Riza, F. et al., 2020).

2.2.4. Pengertian Dekripsi

Deskripsi adalah kegiatan untuk mengembalikan pesan yang telah tersandi atau terenkripsi menjadi pesan asli atau *plaintext*. Proses mengembalikan isi *ciphertext* menggunakan kunci yang telah ditentukan sebelumnya. Dekripsi

adalah kebalikan dari proses enkripsi, yaitu mengubah pesan asli menjadi pesan tersandi atau *ciphertext* (Alfirdaus et al., 2023).

Gambar 2.4. Ilustrasi Proses Dekripsi
(Sumber: *www.skyhighsecurity.com*)

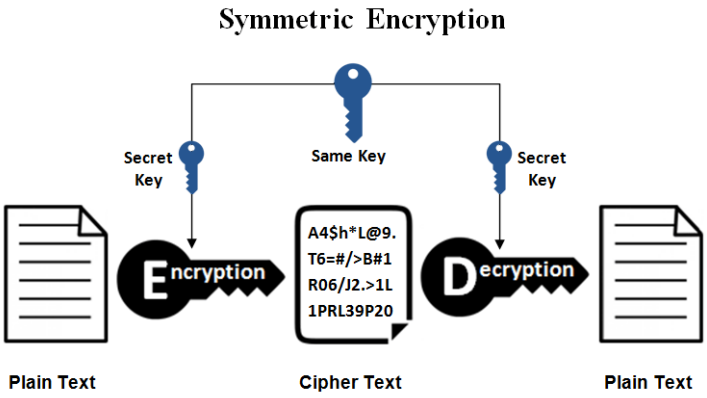
Berikut ini adalah langkah-langkah yang digunakan dalam proses dekripsi antara lain:

1. Input data yaitu data yang telah dienkripsi disebut *ciphertext*.
2. Kunci dekripsi yaitu kunci yang digunakan untuk dekripsi adalah kunci yang sama dengan kunci yang digunakan untuk enkripsi.
3. Algoritma dekripsi yaitu proses algoritma yang digunakan untuk dekripsi adalah algoritma yang sama dengan yang digunakan untuk enkripsi, tetapi prosesnya dibalik untuk mengubah *ciphertext* menjadi plaintext.
4. Plaintext yaitu hasil dari proses dekripsi adalah *plaintext*, yang merupakan data asli sebelum dienkripsi (Riza, F. et al., 2020).

2.2.5. Kriptografi Kunci Simetris

Algoritma kriptografi kunci simetris adalah algoritma kriptografi dengan menggunakan kunci enkripsi dan dekripsi yang sama. Saat menggunakan algoritma ini untuk mengirim pesan, orang yang menerima pesan harus mengetahui kunci yang digunakan agar mereka dapat mendekripsi pesan tersebut. Dengan demikian, keamanan pesan yang dikirim dengan algoritma ini bergantung

pada kunci yang digunakan. Orang lain dapat melakukan enkripsi dan dekripsi pesan jika mereka tahu kuncinya. (Alfirdaus et al., 2023).



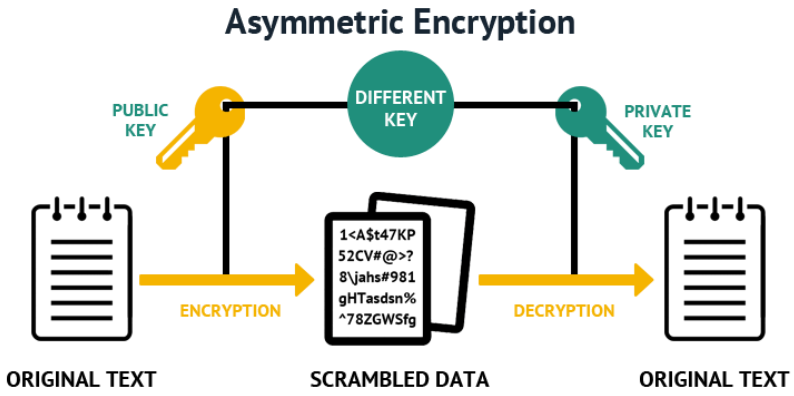
Gambar 2.5. Ilustrasi proses Kriptografi Kunci Simetris (Sumber*: www.sslindonesia.com*)

Kriptografi kunci simetris memiliki dua mode sebagai block cipher atau sebagai *stream cipher*. Dalam *block cipher* semua data dibagi menjadi beberapa blok dan kunci akan diberikan tergantung dari panjang blok, sedangkan dalam *stream cipher* data dibagi menjadi bit-bit kecil yang diacak lalu dienkripsi. Beberapa contoh dari algoritma simetris adalah algoritma AES, algoritma DES, algoritma Blowfish, algoritma Triple DES dan algoritma RC4(Suhandinata et al., 2019).

2.2.6. Kriptografi Kunci Asimetris

Kriptografi kunci asimetris adalah salah satu jenis kriptografi yang menggunakan dua kunci yang berbeda namun saling terkait untuk mengamankan informasi. Kedua kunci ini disebut kunci publik atau *public key* dan kunci privat atau *private key*. Berbeda dengan kriptografi simetris yang hanya menggunakan satu kunci untuk mengenkripsi dan mendekripsi data, kriptografi asimetris

memastikan bahwa apa yang dienkripsi dengan kunci publik hanya dapat didekripsi dengan kunci privat, dan sebaliknya (Riza, F. et al., 2020).



Gambar 2.6. Ilustrasi proses Kriptografi Kunci Simetris (Sumber: *www.sslindonesia.com*)

Metode algoritma asimetris penting karena dapat digunakan untuk membagikan kunci enkripsi atau data lainnya secara aman meskipun kedua belah pihak tidak memiliki kesempatan untuk menyetujui kunci privat. Kunci yang digunakan pada algoritma asimetris umumnya panjang dimana meningkatkan keamanan data yang dikirim, setidaknya berukuran 3000-bit atau lebih untuk mencapai tingkat keamanan algoritma simetris 128-bit. Contoh algoritma asimetris adalah algoritma RSA, ElGamal dan *Diffie-Hellman* (Suhandinata et al., 2019).

Terdapat 2 kunci yang digunakan dalam algoritma kunci asimetris untuk mengamankan data, antara lain:

1. Kunci Publik (*Public Key*)

Kunci publik (*public key*) adalah kunci yang dapat dibagikan kepada pihak lain digunakan untuk mengenkripsi data. (Baraka, 2023).

Kegunaan kunci ini adalah untuk mengenkripsi data atau pesan. Siapapun yang memiliki kunci publik dapat menggunakannya untuk mengenkripsi pesan, tetapi hanya mereka yang memiliki kunci privat yang diizinkan untuk membuka pesan setelah dienkripsi. Karena kunci publik tidak dapat digunakan untuk mendekripsi informasi, kunci publik dapat dibagikan secara bebas kepada siapa saja (Riza, F. et al., 2020).

2. Kunci Privat (*Private Key*)

Kunci Privat (*private key*) adalah kunci yang hanya diketahui oleh pemiliknya, kunci publik digunakan untuk mengenkripsi data, sementara kunci privat digunakan untuk mendekripsi data (Baraka, 2023).

Kunci ini hanya dimiliki oleh satu orang atau entitas dan digunakan untuk mendekripsi data yang telah dienkripsi sebelumnya dengan kunci publik yang tepat. Karena bersifat rahasia, kunci ini harus dijaga dengan sangat hati-hati. Jika kunci privat jatuh ke tangan orang yang salah, orang yang tidak bertanggung jawab dapat mendekripsi data yang dienkripsi dengan kunci publik, yang dapat menyebabkan pelanggaran keamanan yang signifikan (Riza, F. et al., 2020).

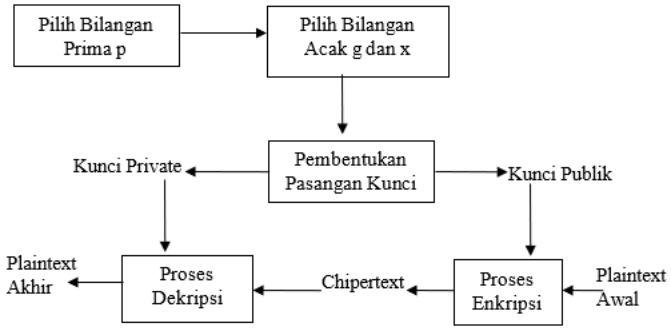
2.2.7. Pengertian Algoritma ElGamal

Algoritma ElGamal diciptakan oleh ilmuwan asal Mesir, Taher ElGamal pada tahun 1985. Konsep dasar algoritma ElGamal terletak pada kunci publik dan awalnya digunakan untuk tanda tangan digital, namun kemudian algoritma ini dimodifikasi sehingga dapat digunakan untuk proses enkripsi dan dekripsi pesan. Algoritma kunci publik ElGamal merupakan algoritma blok *cipher* yang mengenkripsi blok-blok *plaintext* menjadi blok-blok *ciphertext*, yang kemudian

akan didekripsi kembali menggunakan kunci privat dan digabungkan menjadi *plaintext* semula (Nugraha, 2024).

Keamanan algoritma ElGamal terletak pada kesulitan perhitungan logaritma diskrit pada *modulo* prima yang besar, sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sulit untuk dipecahkan. Algoritma ini memiliki kelebihan yaitu pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi dekripsi yang menggunakan proses komputasi yang besar sehingga hasil enkripsinya berukuran dua kali dari ukuran semula (Husaini et al., 2022).

Algoritma ini unggul dalam pembangkitan kunci dengan menggunakan logaritma diskrit pada *modulo* prima yang besar, sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sulit untuk dipecahkan. Algoritma ini memiliki kelebihan yaitu pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi dekripsi yang menggunakan proses komputasi yang besar sehingga hasil enkripsinya berukuran dua kali dari ukuran semula. Kekurangan algoritma ini adalah membutuhkan *resource* yang besar karena *ciphertext* yang dihasilkan dua kali panjang *plaintext* serta membutuhkan *processor* yang mampu untuk melakukan komputasi yang besaruntuk perhitungan logaritma perpangkatan besar. Untuk proses dekripsi, algoritma ini membutuhkan waktu yang lebih lama karena kompleksitas proses dekripsinya yang rumit. Dibutuhkan dua kali komputasi karena ukuran *ciphertext* yang lebih besar dibandingkan *plaintext*-nya (Harahap et al., 2022).



Gambar 2.7. Tahapan Penyelesaian Algoritma ElGamal
(Sumber: Nugraha, 2024)

1. Proses Pemilihan Kunci Elgamal (Elgamal *Generate Key*)

Algoritma ElGamal merupakan sepasang kunci yang dibangkitkan dengan memilih bilangan prima *p* dan dua buah bilangan acak (*random*) *g* dan *x*, dengan syarat bahwa nilai *g* dan *x* lebih kecil dari *p*. Nilai *y*, *g* dan *p* merupakan kunci publik, sedangkan *x*, *p* merupakan pasangan kunci pribadi.

Langkah-langkah dalam pembangkitan kunci pada algoritma kriptografi ElGamal adalah sebagai berikut:

A. Kunci Publik (*Public Key*)

a) Pilih *p* adalah bilangan prima dengan syarat $p > 255$.

b) Pilih bilangan acak *g* dengan syarat $g < p$.

c) Hitung nilai *y* dengan persamaan dibawah ini:

$$y = g^x \bmod p \quad \textbf{.....(2.1)}$$

Keterangan:

g = kunci publik *g*.

x = Kunci privat *x*.

mod = Perhitungan modulus.

p = kunci publik *p* (Ramadhani & Tanti, 2024).

B. Kunci Private (*Private Key*)

a) Pilih *p* adalah bilangan prima dengan syarat $p > 255$.

b) Pilih bilangan acak *x* dengan syarat $x < p$.

C. Bilang Acak Pengirim

a) Pilih bilangan acak *k* (*random k*), dengan syarat $1 \leq k \leq p - 2$. Nilai *k* digunakan untuk menghitung nilai *a* dan *b*.

2. Proses Enkripsi ElGamal

Proses enkripsi merupakan proses mengubah pesan asli (*plaintext*) menjadi pesan rahasia (*chipertext*). Pada proses ini digunakan *public key* (*p, g, y*). Langkah-langkah dalam mengenkripsi *plaintext* adalah sebagai berikut:

a) Potong *plaintext* menjadi blok-blok *m1*, *m2*, …, *mN*.

b) Konversi nilai blok-blok pesan kedalam nilai ASCII.

c) Setiap blok *m* dienkripsi dengan persamaan dibawah ini:

$$\boldsymbol{a = g^k \bmod p}$$

$$\boldsymbol{b = y^k . m \bmod p}$$

Keterangan:

a = Pasangan *ciphertext* pertama.

b = Pasangan *ciphertext* kedua.

g = kunci publik *g*.

k = Bilangan acak (*random k*).

m = nilai *ASCII* karakter *plaintext.*

mod = perhitungan modulus.

p = kunci publik *p.*

d) Susun ciphertext dengan urutan a1, b1; a2, b2; …;aN, bN (Ramadhani & Tanti, 2024).

3. Proses Dekripsi ElGamal

Untuk proses dekripsi pesan *ciphertext* membutuhkan nilai *ciphertext* dari proses enkripsi dan kunci privat *x*. Selanjutnya *ciphertext* dilakukan proses dekripsi dengan perhitungan sebagai berikut:

a) Gunakan kunci privat *x* untuk menghitung persamaan dibawah ini:

$$(\boldsymbol{a^x})^{-1} = \boldsymbol{a^{p-1-x}\ mod\ p} \quad \textbf{(2.2)}$$

b) Hitung *plaintext* (*m*) dengan persamaan dibawah ini:

$$\boldsymbol{m = b(a^x)^{-1}\ mod\ p} \quad \textbf{(2.3)}$$

Keterangan:

a = Pasangan *ciphertext* pertama.

b = Pasangan *ciphertext* pertama.

x = kunci privat *x*

p = kunci publik *p.*

mod = perhitungan modulus.

m = nilai *ASCII* karakter *plaintext* (Nugraha, 2024).

2.3. Sistem Informasi

Sistem informasi adalah komponen-komponen yang saling berhubungan dan bekerja sama untuk mengumpulkan, memproses, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi,

pengendalian dan untuk memberikan gambaran aktivitas di dalam perusahaan (Frisdayanti, 2019).

Secara umum, sistem informasi terdiri dari komponen yang saling berhubungan yang bekerja sama untuk mencapai tujuan tertentu. Tujuan utama sistem komputer adalah mengolah data untuk menghasilkan informasi, yang membutuhkan dukungan dari komponen seperti perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna (*brainware*). (Frisdayanti, 2019).

2.3.1. Aplikasi

Aplikasi adalah perangkat lunak komputer atau program yang dibuat untuk melakukan tugas-tugas khusus, seperti pengelolahan data dan pengelolaan informasi (Wardani et al., 2024).

Perangkat lunak aplikasi adalah subkelas perangkat lunak komputer yang menggunakan kemampuan komputer secara langsung untuk melakukan tugas yang diinginkan pengguna. Perangkat lunak aplikasi biasanya dibandingkan dengan perangkat lunak sistem, yang menggabungkan berbagai kemampuan komputer, tetapi tidak secara langsung menggunakan kemampuan tersebut untuk melakukan tugas yang menguntungkan pengguna. Secara umum aplikasi terbagi menjadi 3, yaitu: aplikasi *web*, aplikasi *desktop* dan aplikasi *mobile* (Suzana, 2021).

2.3.2. *Website*

Website adalah kumpulan halaman yang berisi informasi data digital berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang

disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman website dibuat menggunakan bahasa standar yaitu HTML. Skrip HTML ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang (Susilawati et al., 2020).

Aplikasi *web* pada awalnya dibuat menggunakan bahasa HTML (*HyperText Markup Langauge*), tetapi kemudian dikembangkan menjadi sejumlah skrip dan objek untuk meningkatkan kemampuan HTML. Aplikasi *web* dibagi menjadi dua jenis yaitu *web* statis dan dinamis. *Web* statis dibangun menggunakan skrip HTML, sedangkan *web* dinamis dibangung menggunakan kombinasi HTML dan PHP, Perl, *python*, *javascript* dan lain-lain (Clivan et al., 2019).

2.3.3. HTML

HTML adalah skrip yang digunakan untuk membuat halaman *web* yang dapat digunakan untuk menampilkan konten dalam penjelah *web* internet (Clivan et al., 2019).

HTML merupakan singkatan dari *Hypertext Markup Language*. HTML dikembangkan pertama kali oleh Tim Berners-Lee bersamaan dengan protokol HTTP (*Hypertect Transfer Protocol*) pada tahun 1989. Tujuan utama pengembangan HTML adalah untuk menghubungkan satu halaman *web* hanya berupa teks, tidak seperti sekarang. HTML merupakan bahasa dasar *web* yang berfungsi untuk menampilkan berbagai komponen *web*. HTML juga berarti sekumpulan *tag-tag* yang menyusun setiap elemen dari *website*. HTML berperan

sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website* sesuai layout yang diinginkan (Susilawati et al., 2020).

2.3.4. PHP

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web* dan dapat digunakan sebagai bahasa pemrograman umum. PHP diciptakan oleh Rasmus Lerdorf pada tahun 1994. PHP bersifat *open source* yang berarti dapat digunakan secara gratis (Noviana, 2022).

PHP juga merupakan bahasa *scripting* yang dijalankan di *server* dan terintegrasi dengan HTML. Setiap sintak yang kami berikan akan dijalankan sepenuhnya di *server*, sedangkan sintak yang dikirimkan ke *browser* hanyalah *output*-nya. *File* PHP harus disimpan pada dokumen *root apache*, atau pada direktori *htdocs* pada *softaware* XAMPP (Arisantoso et al., 2023).

Gambar 2.8. Logo PHP
(Sumber: *https://wikipedia.org*)

2.3.5. *JavaScript*

JavaScript (JS) adalah bahasa pemrograman tingkat tinggi yang pertama kali dibuat untuk membuat *website* lebih "hidup". *JavaScript*, bersama dengan HTML dan CSS, adalah bahasa pemrograman yang paling umum digunakan

untuk membuat aplikasi berbasis *web* dan memiliki kemampuan untuk membuat *website* lebih logis dan interaktif (Arisantoso et al., 2023).

JavaScript memungkinkan pengembang menerapkan fitur kompleks pada halaman *web*. Setiap kali halaman *web* menampilkan informasi statis untuk pengguna, seperti pembaruan konten secara *live*, peta interaktif, animasi 2D dan 3D, *slider* video dan musik, dapat dipastikan bahwa *JavaScript* digunakan (Arisantoso et al., 2023).

2.3.6. MySQL

MySQL merupakan *database engine* atau server database yang mendukung bahasa *database* pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, multi-user. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasuskasus dimana penggunaannya tidak cocok dengan penggunaan GPL(Clivan et al., 2019).

MySQL juga merupakan *Database Management System* (DBMS) *opensource tools* yang mendukung multi pengguna, populer, dan gratis. Berdasarkan teori diatas maka dapat disimpulkan bahwa SQL adalah bahasa permintaan pemrosesan *database* tertentu dimana subbahasa dapat dimanipulasi data di dalam *database*. SQL digunakan untuk melakukan tugas seperti *update*, *insert* dan *delete* terhadap *database*, yang merujuk pada konsep *Relational Database Management System* (RDBMS) (Noviana, 2022).

Gambar 2.9. Logo MySQL
(Sumber: *https://wikipedia.org*)

2.3.7. *Database*

Database adalah sekumpulan data yang dikelola menurut ketentuan tertentu yang saling berkaitan sehingga memudahkan pengelolahannya. Database memiliki peran penting sebagai saranan untuk mengumpulkan informasi, file, atau dat secara terintegrasi (Wardani et al., 2024).

Database juga berarti sekumpulan data yang saling berhubungan secara logis dan penjelasan dari data tersebut, yang dirancang untuk menemukan data yang dibutuhkan sebuah organisasi. Basis data adalah sekumpulan elemen data terintegrasi yang secara logika saling berhubungan, dan dapat digunakan oleh berbagai departemen dan pengguna. Semua data di dalamnya diintegrasikan untuk menghindari duplikat (Susilawati et al., 2020).

2.3.8. PDF

PDF adalah format *file* yang diciptakan oleh *Adobe Systems* pada tahun 1993 dan digunakan untuk bertukar dokumen digital. Format PDF digunakan untuk menyajikan dokumen dua dimensi yang berisi teks, huruf, gambar, dan grafik. Karena tidak bergantung pada perangkat lunak, format PDF adalah format

dokumen yang sangat populer. Hampir semua *platform* dan komputer mendukung perangkat lunak format ini (Sitohang, 2023).

Dalam hal penggunaan PDF, sangat umum di dunia pekerjaan. Banyak orang menggunakan format PDF karena dianggap lebih praktis dan efisien. Ukuran file PDF yang besar akan memperlambat transmisi data dan menguras memori penyimpanan. Akibatnya, untuk memampatkan isi file PDF, proses kompresi harus dilakukan dengan metode tertentu (Sitohang, 2023).

Gambar 2.10. Logo *File* Format PDF pada *Software Adobe Reader*
(Sumber: *https://wikipedia.org*)

2.3.9. Pemodelan Berorientasi Objek (UML)

UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software. UML juga dapat digunakan sebagai alat transfer ilmu tentang sistem aplikasi yang akan dikembangkan dari developer satu ke developer lainnya. Unified Modeling Language (UML) adalah teknik dan himpunan struktur untuk pemodelan desain programberorientasi objek/OOP dan aplikasinya (Helsalia et al., 2021).

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

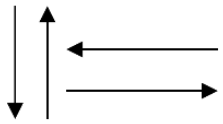
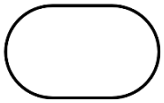
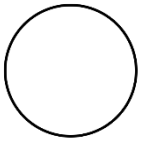
1. *Flowchart*

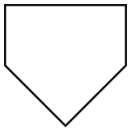
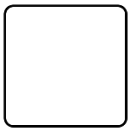
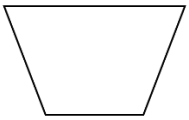
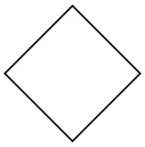
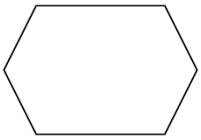
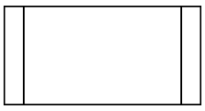
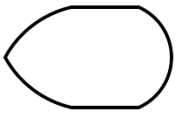
Flowchart adalah diagram alir yang tersusun dari bagan dengan berbagai bentuk. Flowchart juga mewakili suatu algoritma dan proses yang menunjukan langkah-langkah kegiatan (Wardani et al., 2024).

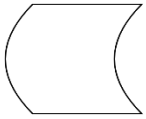
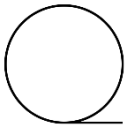
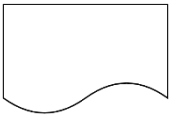
Flowchart biasanya menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah (Rasiban et al., 2024).

Simbol-simbol yang biasa digunakan pembuatan *flowchart* dapat dilihat pada tabel 2.1 di bawah ini:

Tabel 2.1. Simbol *Flowchart*

Simbol	Nama Simbol	Keterangan
	*Flow Direction symbol*	Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga *connecting line*.
	*Terminator Symbol*	Yaitu simbol untuk permulaan (*start*) atau akhir (*stop*) dari suatu kegiatan.
	*Connector Symbol*	Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar/halaman yang sama.

	*Connector Symbol*	Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.
	*Processing Symbol*	Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer.
	*Manual Operation Symbol*	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	*Decision Symbol*	Simbol pemilihan proses berdasarkan kondisi yang ada.
	*Input-Output Symbol*	Simbol yang menyatakan proses *input* dan *output* tanpa tergantung dengan jenis peralatannya.
	*Manual Input Symbol*	Simbol untuk pemasukan data secara manual *on-line keyboard*
	*Preparation Symbol*	Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam *storage*.
	*Predefine Process Symbol*	Simbol untuk pelaksanaan suatu bagian (sub-program)/*prosedure*.
	*Display Symbol*	Simbol yang menyatakan peralatan *output* yang digunakan yaitu layar, *plotter*, *printer* dan sebagainya.

	*Disk and On-line Storage Symbol*	Simbol yang menyatakan input yang berasal dari *disk* atau disimpan ke *disk*.
	*Magnetik Tape Unit Symbol*	Simbol yang menyatakan *input* berasal dari pita magnetik atau *output* disimpan ke pita magnetik.
	*Punch Card Symbol*	Simbol yang menyatakan bahwa input berasal dari kartu atau *output* ditulis ke kartu.
	*Document Symbol*	Simbol yang menyatakan *input* berasal dari dokumen dalam bentuk kertas atau *output* dicetak ke kertas.

(Sumber: Rasiban et al., 2024).

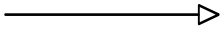
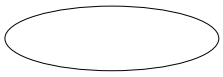
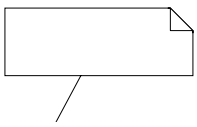
2. *Use Case* Diagram

Use case diagram adalah jenis diagram *Unified Modeling Language* (UML) yang digunakan dalam rekayasa perangkat lunak untuk secara visual merepresentasikan interaksi antara berbagai aktor (pengguna atau sistem eksternal) dan suatu sistem. Diagram ini menggambarkan bagaimana pengguna berinteraksi dengan sistem untuk mencapai tujuan tertentu (Rasiban et al., 2024).

Beberapa konsep utama yang termasuk dalam *use case* diagram dapat dilihat pada tabel 2.1 di bawah ini:

Tabel 2.2. Simbol *Use Case*

Simbol	Nama Simbol	Keterangan
	*Actor*	*Actor* Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case.*

	Dependency	hubungan di mana elemen yang tidak mandiri (independen) dipengaruhi oleh elemen mandiri yang berubah.
	*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	Include	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
	Extend	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
	Association	Menghubungkan antara objek satu dengan objek lainnya.
	*System*	Paket yang menspesifikasikan menampilkan sistem secara terbatas.
	*Use case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk *Collaboration* menyediakan prilaku yang lebih besar dari jumlah dan elemen- elemennya (sinergi).
	*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

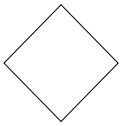
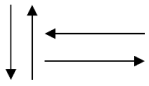
(Sumber: Rasiban et al., 2024).

3. Diagram Aktivitas (*Activity Diagram*)

Activity diagram adalah jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan aliran kerja atau aktivitas dalam suatu sistem atau proses. Diagram ini menyajikan serangkaian kegiatan, tindakan, dan keputusan yang terjadi sepanjang waktu (Rasiban et al., 2024).

Simbol yang digunakan dalam *activity* diagram dapat dilihat pada tabel 2.2 dibawah ini:

Tabel 2.3. Simbol *Activity* Diagram

Simbol	Nama Simbol	Keterangan
	*Action / Activity*	Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	*Initial Node*	Bagaimana objek dibentuk atau diawali.
	*Activity Final Node*	Bagaimana objek dibentuk dan diakhiri.
	*Decision*	Diguanakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu.
	*Line Connector*	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya.

(Sumber: Rasiban et al., 2024).

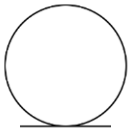
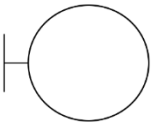
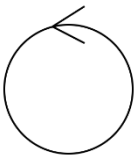
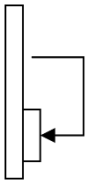
4. Diagram Urutan (*Sequence Diagram*)

Diagram urutan (*sequence diagram*) adalah jenis diagram dalam *Unified Modeling Language* (UML) yang menggambarkan interaksi antara objek dalam

suatu sistem secara kronologis. Diagram ini menunjukkan bagaimana objek-objek berkomunikasi satu sama lain dan berurutan dalam eksekusi suatu skenario atau proses (Rasiban et al., 2024).

Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel 2.3 dibawah ini :

Tabel 2.4. Simbol *Sequence Diagram*

Simbol	Nama Simbol	Keterangan
	*Entity Class*	Gambaran sistem sebagai landasan dalam menyusun basis data.
	*Boundary Class*	Menangani komunikasi antar lingkungan sistem.
	*Control Class*	Bertanggung jawab terhadap kelas kelas terhadap objek yang berisi logika
	*Recursive*	Pesan untuk dirinya
	*Activation*	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

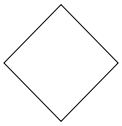
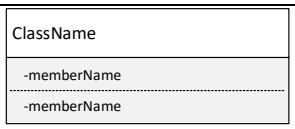
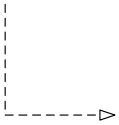
(Sumber: Rasiban et al., 2024).

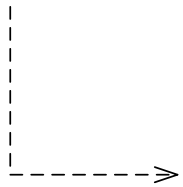
5. Diagram Kelas (*Class Diagram*)

Diagram kelas (*class diagram*) adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan struktur statis dari suatu sistem atau aplikasi berorientasi objek (Rasiban et al., 2024).

Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel 2.4 dibawah ini:

Tabel 2.5. Simbol *Class* Diagram

Simbol	Nama Simbol	Keterangan
	Generalization	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu *actor*.
	*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.

	*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	*Association*	Suatu yang menghubungkan antara objek satu dengan objek lainnya

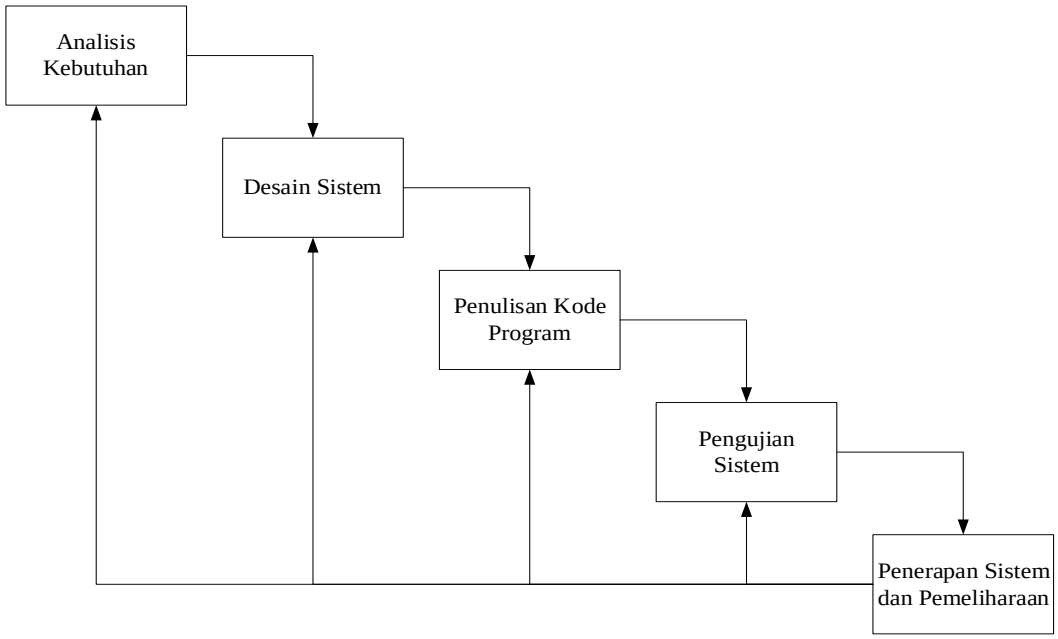
(*Sumber*: Rasiban et al., 2024)

BAB III

METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian adalah langkah-langkah sistematis yang dilakukan dalam sebuah penelitian untuk memperoleh hasil yang valid dan dapat dipertanggungjawabkan. Adapun proses tahapan penelitian yang akan digambarkan dalam bentuk *waterfall* yaitu:



Gambar 3.1. Metode *Waterfall* Dalam Pengembangan Sistem

Keterangan :

1. Analisis Kebutuhan

Pada tahapan ini merupakan analisa terhadap kebutuhan yang diperlukan untuk mencapai tujuan penelitian yang akan dilakukan. Dilakukan pengumpulan kebutuhan data dan aplikasi sebagai media perancangan program yang akan dibangun. Adapun beberapa hal yang penulis butuhkan antara lain:

A. Data soal ujian sekolah pada SMA Adlin Murni.

B. Perangkat lunak *server* lokal, penulis menggunakan XAMPP.

C. Layanan *hosting*, penulis menggunakan layanan *anymhost.id*

D. Perangkat lunak *web browser*, penulis menggunakan *Google Chrome*

E. Perangkat lunak teks editor, penulis menggunakan *Sublime Text* versi 3

2. Desain Sistem

Proses desain akan menerjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat kode program. Proses ini berfokus kepada : struktur data, keamanan data, arsitektur perangkat lunak, representasi antarmuka, dan alur prosedural. Pada tahap ini dilakukan desain perangkat lunak menggunakan pemodelan *UML* yaitu *flowchart, use case* diagram, *class* diagram, *activity* diagram dan *sequence* diagram.

3. Penulisan Kode Program

Kode program merupakan terjemahan perintah dalam bahasa yang bisa dikenali komputer. Pada tahap ini desain sistem diimplementasikan ke dalam kode program. Penulisan kode program dimulai dengan perancangan desain antarmuka menggunakan sintak HTML dan CSS, proses selanjutnya adalah perancangan kontrol utama menggunakan bahasa pemrograman PHP dan *Javascript*, serta proses saling bertukar data pada *database* MySQL menggunakan sintak SQL dengan menggunakan ekstensi MySQLi.

4. Pengujian Sistem

Tahap pengujian sistem adalah proses hasil dari analisis masalah yang telah melalui tahap-tahap desain sistem, pada tahap ini dilakukan beberapa pengujian

sistem untuk memastikan sistem dapat berjalan sesuai dengan rancangan sebelumnya. Adapun proses pengujian sistem adalah sebagai berikut:

A. Pengujian Fungsionalitas

a) Enkripsi data: Memastikan data dapat benar dienkripsi menggunakan algoritma kriptografi ElGamal dengan perhitungan yang sesuai.

b) Dekripsi data: Memastikan data yang didekripsi kembali menjadi *plaintext* awal sesuai dengan perhitungan yang sesuai.

c) *Generate public key* dan *private key*: Memastikan proses pembuatan kunci otomatis sesuai dengan aturan algoritma kriptografi ElGamal.

d) *Upload* dan *download file*: Pengujian pengguna dalam proses pengunggahan dan pengunduhan data.

B. Kesesuaian Tampilan Antarmuka

a) Kesesuaian Alur Sistem: Memastikan desain antarmuka pengguna telah sesuai dengan alur sistem yang telah dirancang.

b) Kemudahan Pengguna**:** Memastikan kemudahan proses bagi pengguna dalam menggunakan sistem.

5. Penerapan Sistem dan Pemeliharaan

Pada tahapan ini merupakan langkah terakhir dalam siklus perancangan sistem. Berikut adalah beberapa proses penerapan sistem dan pemeliharaan yang dapat diterapkan:

A. Penerapan Program

Pada tahap ini, sistem yang telah melewati tahapan pengujian siap untuk digunakan oleh pengguna akhir.

B. Pemeliharaan (*Maintenance*)

Setelah sistem diterapkan, pemeliharaan dilakukan untuk memastikan sistem tetap berjalan dengan baik dan aman. Beberapa aspek pemeliharaan meliputi: perbaikan masalah (*bug fixing*), pembaruan keamanan, optimasi performa dan peningkatan fitur dapat dilakukan seiring dengan penggunaan sistem yang telah berjalan.

3.2. Metode Pengumpulan Data

Pada tahap ini dilakukan dengan mempelajari teori dasar yang mendukung penelitian, pencarian dan pengumpulan data-data yang dibutuhkan. Untuk mengumpulkan data yang dibutuhkan, maka penulis memakai teknik:

1. Penelitian Perpustakaan (*Library Research*)

Studi kepustakaan dapat diartikan sebagai suatu langkah untuk memperoleh informasi dari penelitian terdahulu yang harus dikerjakan. Studi pustaka dapat dilakukan dengan mengumpulkan data dari berbagai sumber tertulis, baik dalam bentuk cetak maupun digital. Adapun beberapa sumber yang penulis gunakan antara lain:

A. Jurnal Ilmiah

Artikel jurnal ilmiah baik nasional maupun internasional, yang memberikan informasi terbaru dan hasil penelitian terdahulu yang relevan dengan topik yang dikaji.

B. Buku

Buku akademik, referensi, atau literatur lain yang membahas topik yang relevan dengan penelitian.

C. Sumber *Online*

Sumber *online* adalah bahan referensi yang tersedia dalam bentuk digital di internet dan dapat diakses melalui berbagai platform. Dalam studi pustaka, sumber online digunakan untuk memperoleh informasi yang lebih luas, terkini, dan mudah diakses.

2. Pengamatan (*Observation*)

Metode pengumpulan data yang dilakukan dengan cara mengamati secara langsung objek, perilaku, atau fenomena di lokasi penelitian. Metode ini bertujuan untuk memperoleh data yang akurat dan objektif dengan mengamati situasi nyata tanpa intervensi dari peneliti. Lokasi pengamatan yang dipilih penulis adalah di lingkungan SMA Adlin Murni.

3. Wawancara (*Interview*)

Wawancara adalah metode pengumpulan data yang dilakukan dengan mengajukan pertanyaan secara langsung kepada responden guna mendapatkan informasi yang lebih mendalam mengenai suatu topik. Wawancara sering digunakan dalam penelitian kualitatif maupun kuantitatif untuk memperoleh data yang tidak bisa didapatkan hanya melalui observasi atau dokumen tertulis. Dalam proses wawancara ini, penulis melakukan sesi tanya jawab dengan:

A. Petugas tata usaha SMA Adlin Murni yang berwenang dalam menyimpan data soal ujian sekolah.
B. Guru yang bertugas dalam pembuatan soal ujian sekolah.

4. Sampel (*Sampling*)

Sampel adalah bagian dari populasi yang dipilih untuk dijadikan sumber data dalam penelitian. Dalam pengumpulan data, sampel digunakan untuk mewakili

karakteristik populasi secara keseluruhan agar hasil penelitian dapat digeneralisasikan. Pemilihan sampel yang tepat sangat penting untuk memastikan bahwa data yang dikumpulkan akurat, valid, dan dapat digunakan untuk menjawab pertanyaan penelitian. Adapun sampel yang penulis butuhkan dalam penelitian ini antara lain:

A. Sampel *file* data soal ujian sekolah dalam format PDF dengan ukuran tidak lebih dari 1 *megabyte.*
B. Biodata guru pembuat soal ujian sekolah untuk dijadikan *user* atau pengguna pada aplikasi bank soal.

3.3. Perancangan Sistem

Penulis merancang desain sistem menggunakan 2 hak akses, yaitu *User* yang bertugas sebagai pengunggah (*upload*) data soal ujian sekolah dan Administrator yang bertugas sebagai pengunduh (*download*) dan proses manajemen serta mendistribusikan data soal ujian sekolah. Dengan adanya pembagian hak akses ini, keamanan dan kerahasiaan soal dapat lebih terjamin, sehingga hanya pihak yang berwenang yang dapat mengakses soal dalam bentuk aslinya. Sistem ini dirancang menggunakan bahasa pemodelan UML (*Unified Modeling Language*) untuk memberikan gambaran yang lebih jelas mengenai struktur serta alur kerja sistem. Pemodelan ini mencakup *flowchart*, *use case* diagram, *class* diagram, *activity* diagram, dan *sequence* diagram.

3.3.1. Flowchart

Flowchart digunakan untuk menggambarkan alur kerja sistem secara lebih detail, termasuk tahapan-tahapan utama dalam pengolahan data soal dan penerapan

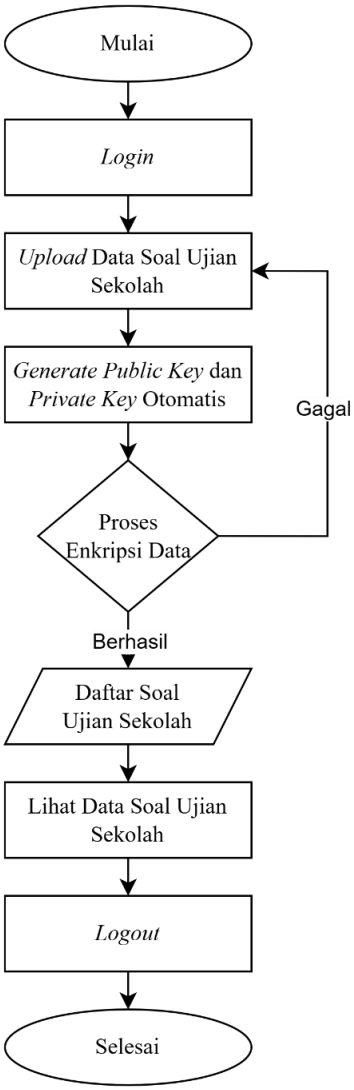
kriptografi ElGamal. *Flowchart* mencakup alur program bagi *user* dan administrator, serta alur program dalam penerapan algoritma ElGamal untuk mengamankan data, yaitu proses enkripsi, dekripsi dan pembangkitan kunci.

1. *Flowchart* Alur Program

Flowchart ini menjelaskan bagaimana pengguna (*user*) dan administrator berinteraksi dengan sistem.

a) *Flowchart* Alur Program *User*

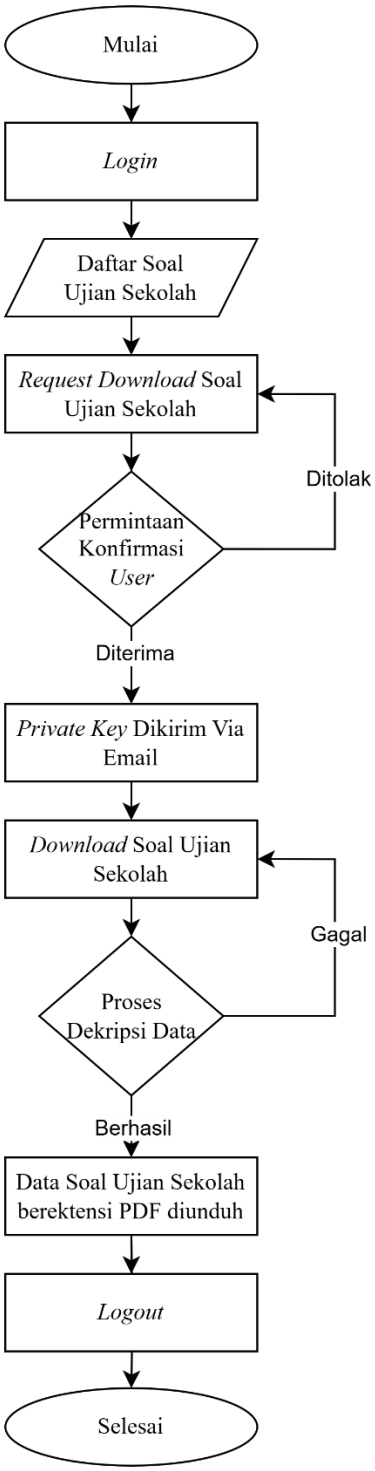
Proses alur program pada pengguna atau *user* dapat dilihat pada *flowchart* dibawah ini:



Gambar 3.2. *Flowchart* Alur Program *User*

b) *Flowchart* Alur Program Administrator

Proses alur program pada administrator dapat dilihat pada *flowchart* dibawah ini:



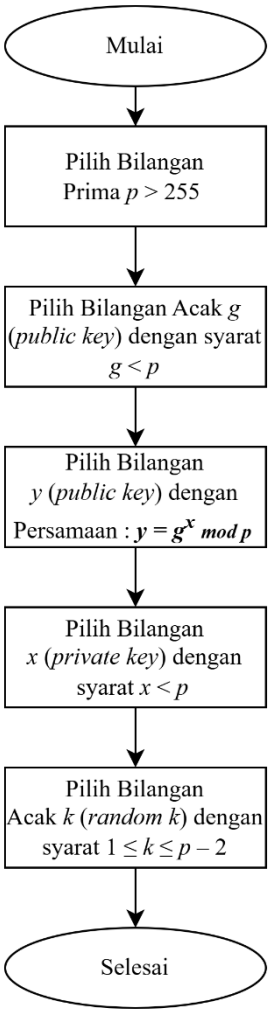
Gambar 3.3. *Flowchart* Alur Program Administrator

2. *Flowchart* Proses Algoritma Kriptografi ElGamal

Pada *flowchart* ini berfokus pada bagaimana algoritma ElGamal diterapkan dalam sistem, mulai dari pembangkitan kunci hingga proses enkripsi dan dekripsi.

a) *Flowchart* Proses Pembangkitan Kunci (*Generate Key*)

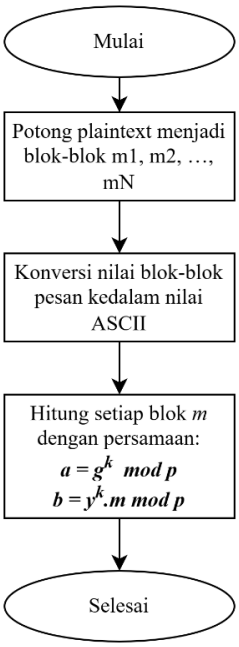
Proses pembangkitan kunci (*generate key*) pada algoritma kriptografi ElGamal dapat dilihat pada *flowchart* dibawah ini:



Gambar 3.4. *Flowchart* Proses Pembangkitan Kunci (*Generate Key*)

b) *Flowchart* Proses Enkripsi Data

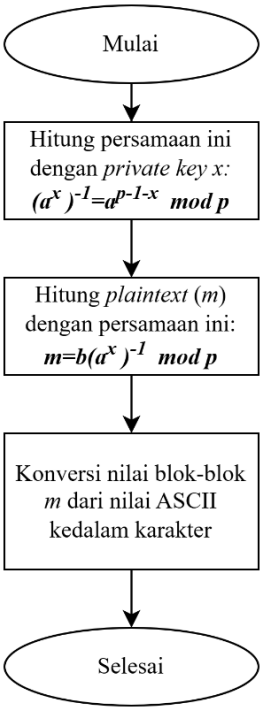
Proses enkripsi data pada algoritma kriptografi ElGamal dapat dilihat pada *flowchart* dibawah ini:



Gambar 3. 5. *Flowchart* Proses Enkripsi Data

c) *Flowchart* Proses Dekripsi Data

Proses dekripsi data pada algoritma kriptografi ElGamal dapat dilihat pada *flowchart* dibawah ini:



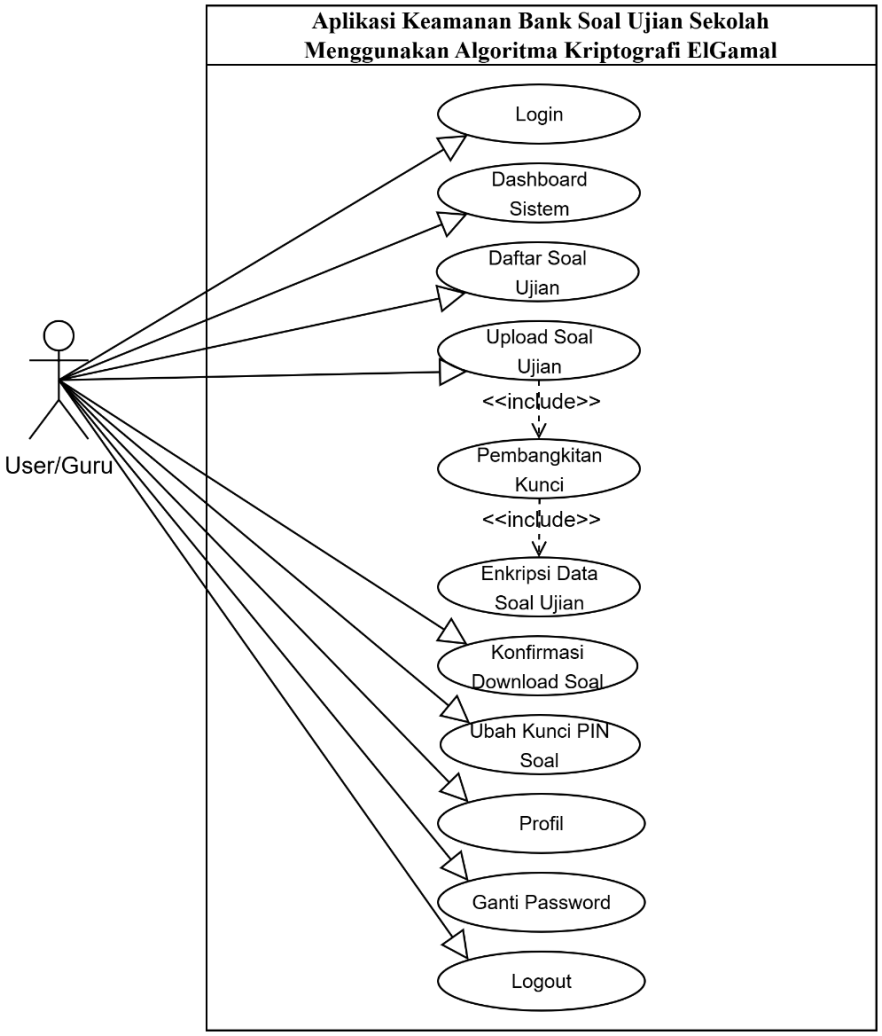
Gambar 3. 6. *Flowchart* Proses Dekripsi Data

3.3.2. *Use Case* Diagram

Dalam *use case diagram* merupakan model diagram UML (*Unified Modelling Language*) yang digunakan untuk menggambarkan *requirement* fungsional yang diharapkan dari sebuah sistem, proses sistem *use case* diagram dibagi menjadi 2 yaitu sebagai berikut:

1. *Use Case* Diagram Pada *User*

Use case diagram pada *user* menampilkan model diagram apa-apa saja yang dapat dilakukan oleh user/pengguna dalam sistem. Adapun gambar *use case* diagram pada *user* dapat dilihat dibawah ini:



Gambar 3.7. *Use Case* Diagram Pada *User*

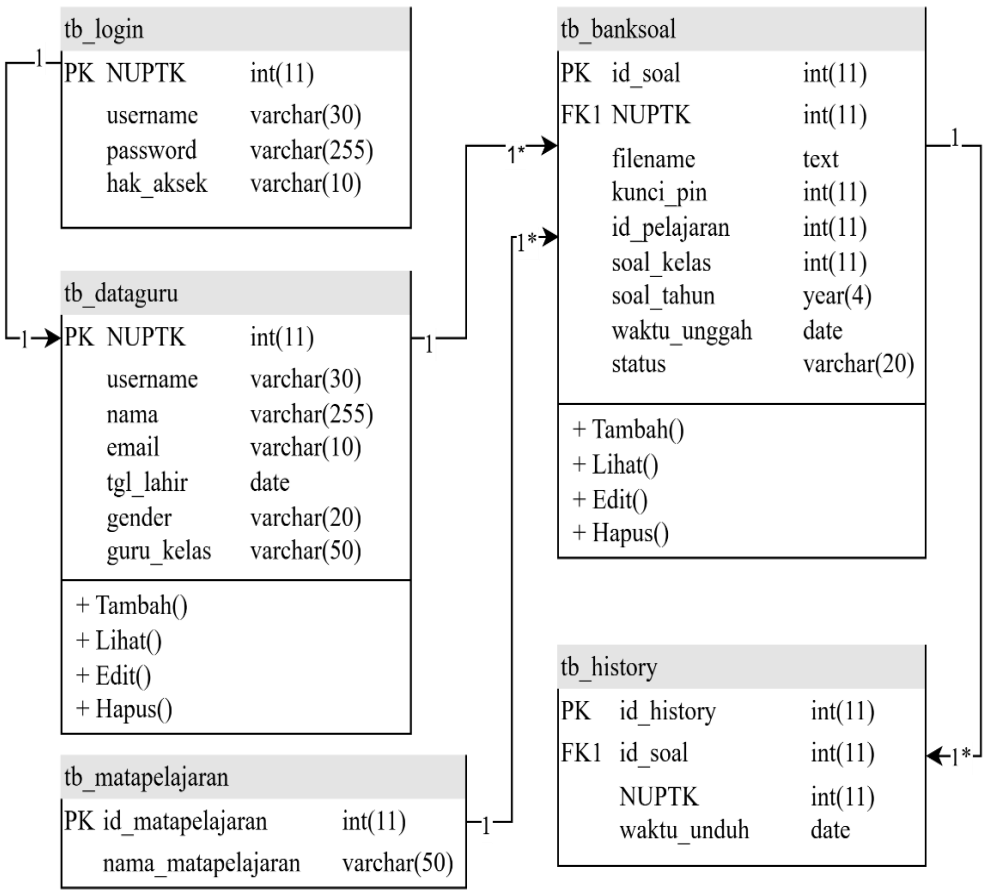
2. *Use Case Diagram* Pada Administrator

Use case diagram pada administrator menampilkan model diagram apa-apa saja yang dapat dilakukan oleh administrator dalam sistem. Berbeda dengan *user*, use case diagram pada administrator memiliki beberapa halaman yang tidak bisa diakses oleh *user*. Adapun gambar kerja *use case* diagram pada administrator dapat dilihat dibawah ini:

Gambar 3.8. *Use Case* Diagram Pada Administrator

3.3.3. *Class* Diagram

Class Diagram adalah diagram dalam UML yang digunakan untuk memodelkan struktur statis sistem dengan merepresentasikan kelas, atribut, metode, dan hubungan antar kelas. Adapun rancangan kelas-kelas yang akan digunakan pada sistem yang akan dirancang dapat dilihat sebagai berikut:



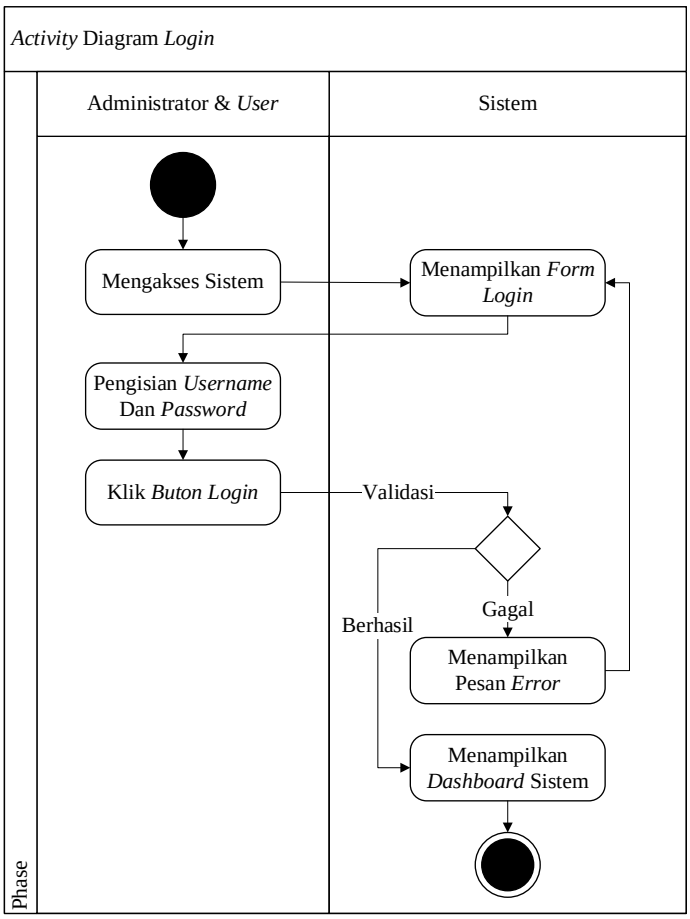
Gambar 3.9. *Class* Diagram Pada Sistem

3.3.4. *Activity* Diagram

Proses yang telah digambarkan pada *use case* diagram diatas dijabarkan dengan *activity* diagram. Adapun beberapa rangkaian *activity* diagram adalah sebagai berikut:

1. *Activity* Diagram *Login*

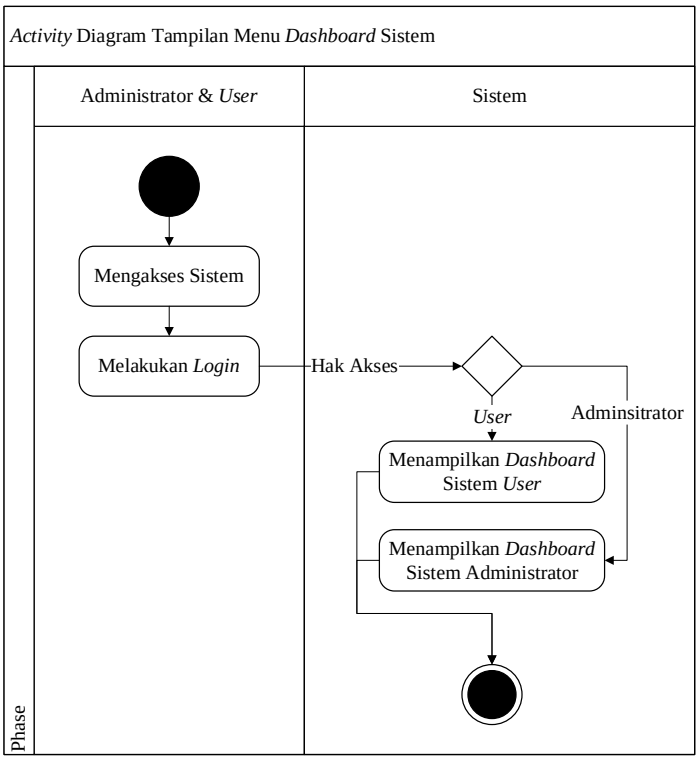
Aktivitas diagram *login* hanya bisa dilakukan oleh administrator dan *user* yang telah didaftarkan oleh administrator untuk masuk kedalam *dashboard* sistem dan melalukan segala aktivitas didalam sistem sesuai hak akses masing-masing. Adapun aktivitas diagram untuk melakukan *login* adalah sebagai berikut:



Gambar 3.10. *Activity* Diagram *Login*

2. *Activity* Diagram Tampilan Menu *Dashboard* Sistem

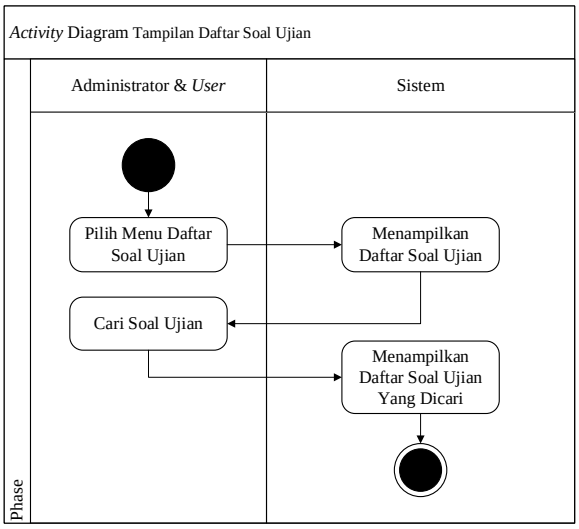
Aktivitas diagram tampilan menu *dashboard* sistem menampilkan beberapa ringkasan jumlah *user* terdaftar, jumlah soal tersimpan dan jumlah banyak kelas serta terdapat alur program singkat. Adapun aktivitas diagram pada tampilan menu *dashboard* sistem adalah sebagai berikut:



Gambar 3.11. Tampilan Menu *Dashboard* Sistem

3. *Activity* Diagram Tampilan Daftar Soal Ujian

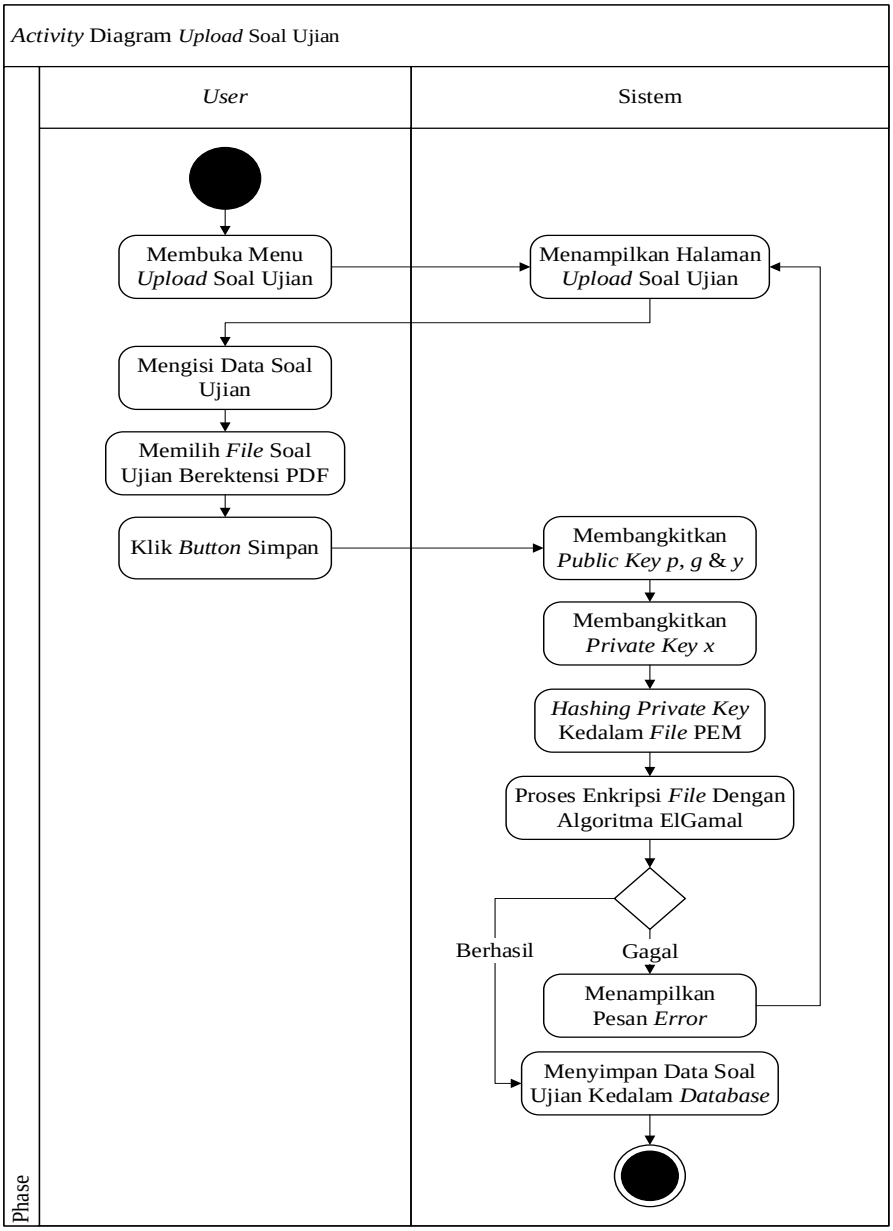
Aktivitas diagram tampilan daftar soal ujian menampilkan daftar soal ujian sekolah yang telah tersimpan oleh semua *user*. Adapun aktivitas diagram pada tampilan daftar soal ujian adalah sebagai berikut:



Gambar 3.12. *Activity* Diagram Tampilan Daftar Soal Ujian

4. *Activity* Diagram *Upload* Soal Ujian

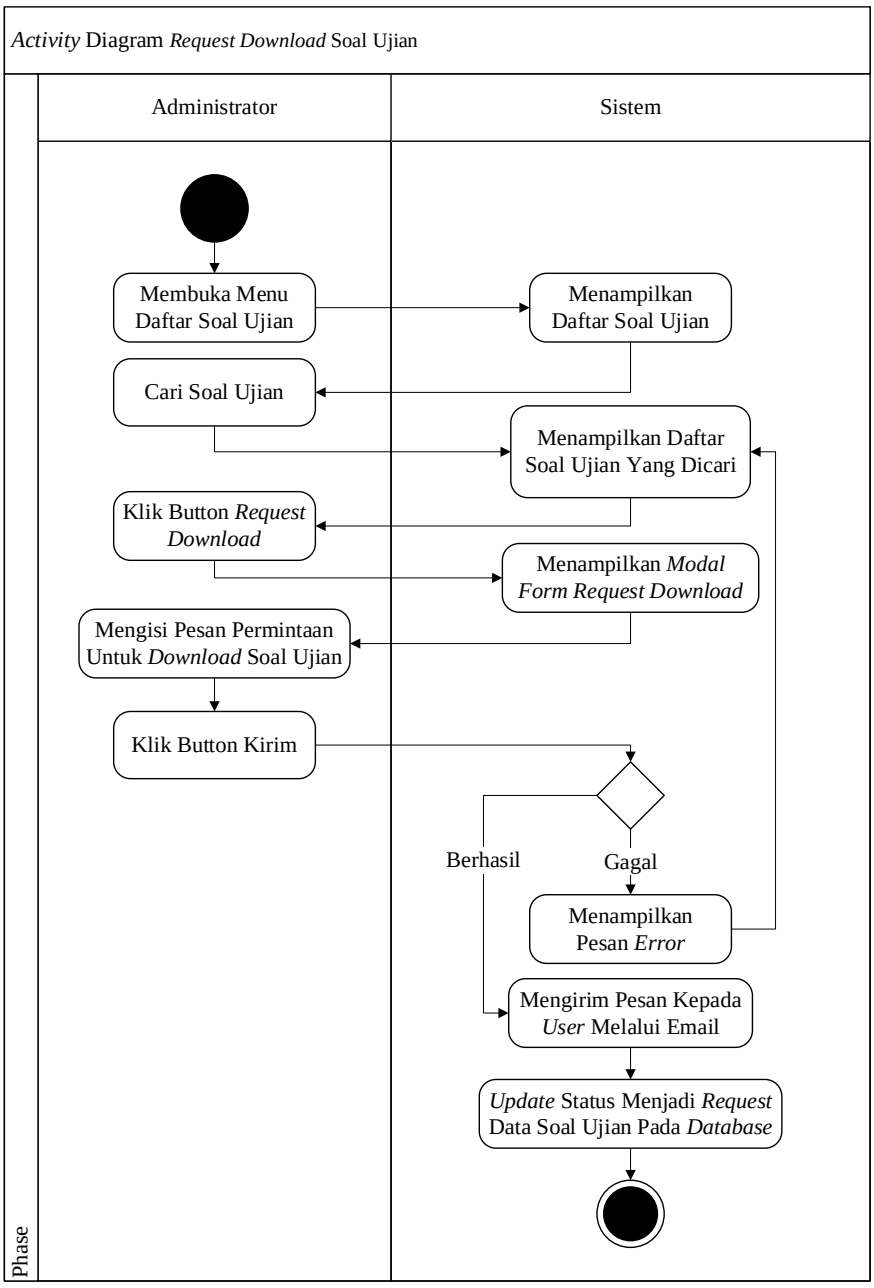
Aktivitas diagram upload soal ujian menampilkan proses *user* dalam melakukan upload soal ujian sekolah kedalam sistem dan juga menampilkan proses enkripsi data serta proses pembangkitan kunci (*generate key*) yang akan dikunci dengan kunci PIN yang dibutuhkan untuk proses dekripsi data. Adapun aktivitas diagram pada upload soal ujian adalah sebagai berikut:



Gambar 3.13. *Activity* Diagram *Upload* Soal Ujian

5. Activity Diagram *Request Download* Soal Ujian

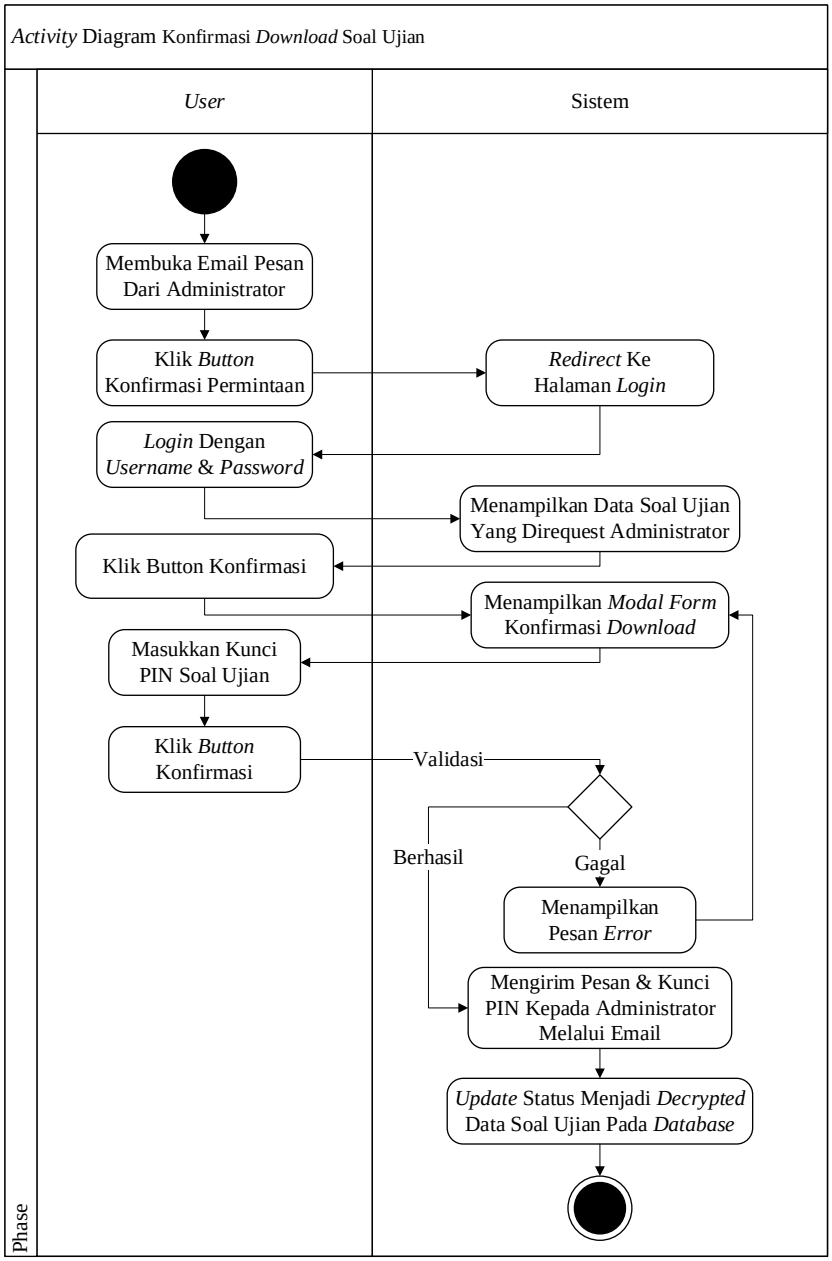
Aktivitas diagram *request download* soal ujian adalah aktivitas yang hanya bisa dilakukan oleh administrator dalam meminta persetujuan kepada *user* untuk mengunduh data soal ujian sekolah serta meminta kunci PIN soal ujian. Adapun aktivitas pada *request download* soal ujian adalah sebagai berikut:



Gambar 3. 14. *Activity* Diagram *Request Download* Soal Ujian

6. *Activity* Diagram Konfirmasi *Download* Soal Ujian

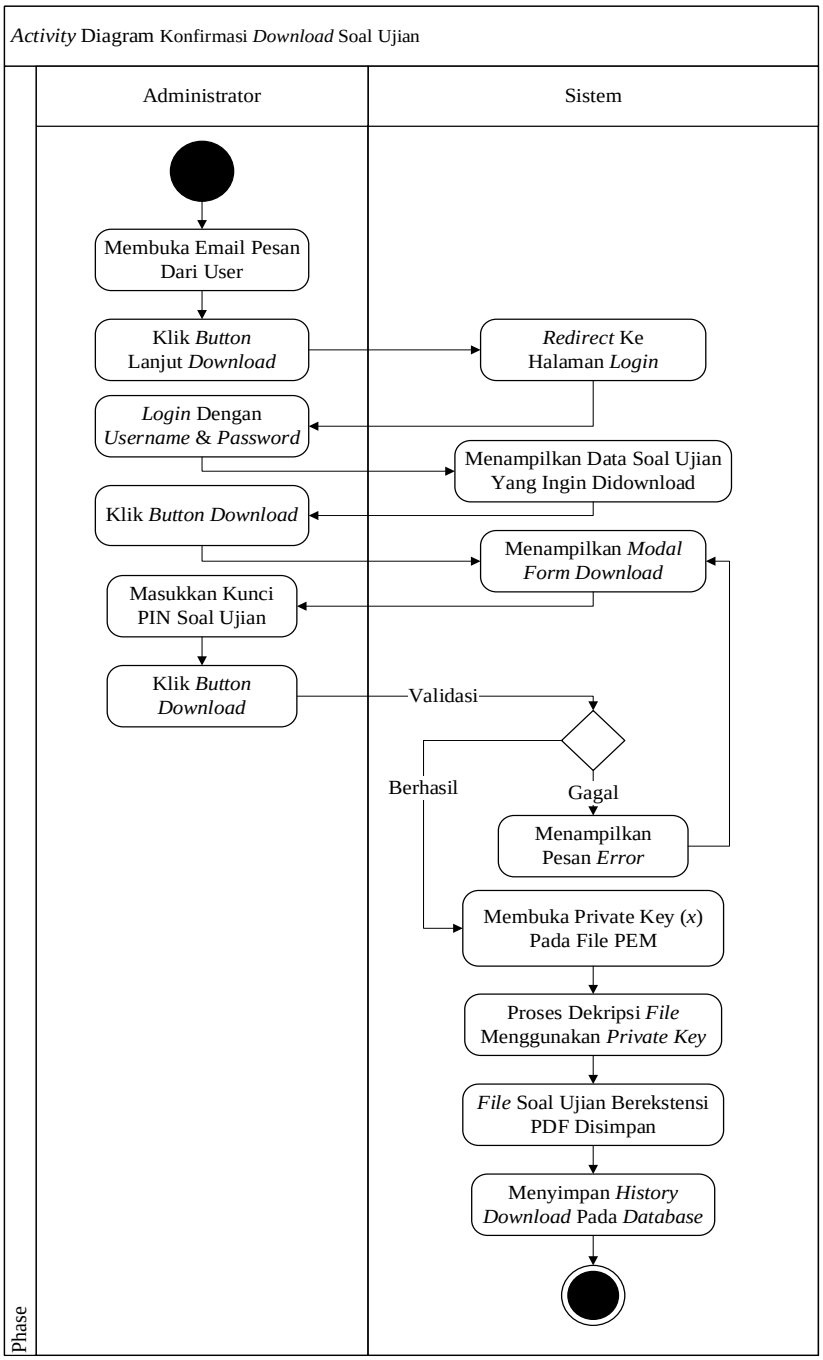
Aktivitas diagram konfirmasi *download* soal ujian adalah aktivitas yang hanya bisa dilakukan *user* dalam memberikan konfirmasi kepada administrator untuk mengunduh data soal ujian, pada proses ini kunci PIN yang dibutuhkan untuk proses membuka *private key* akan diberikan kepada administrator. Adapun aktivitas pada konfirmasi *download* soal ujian adalah sebagai berikut:



Gambar 3.15. *Activity* Diagram Konfirmasi *Download* Soal Ujian

7. *Activity* Diagram *Download* Soal Ujian

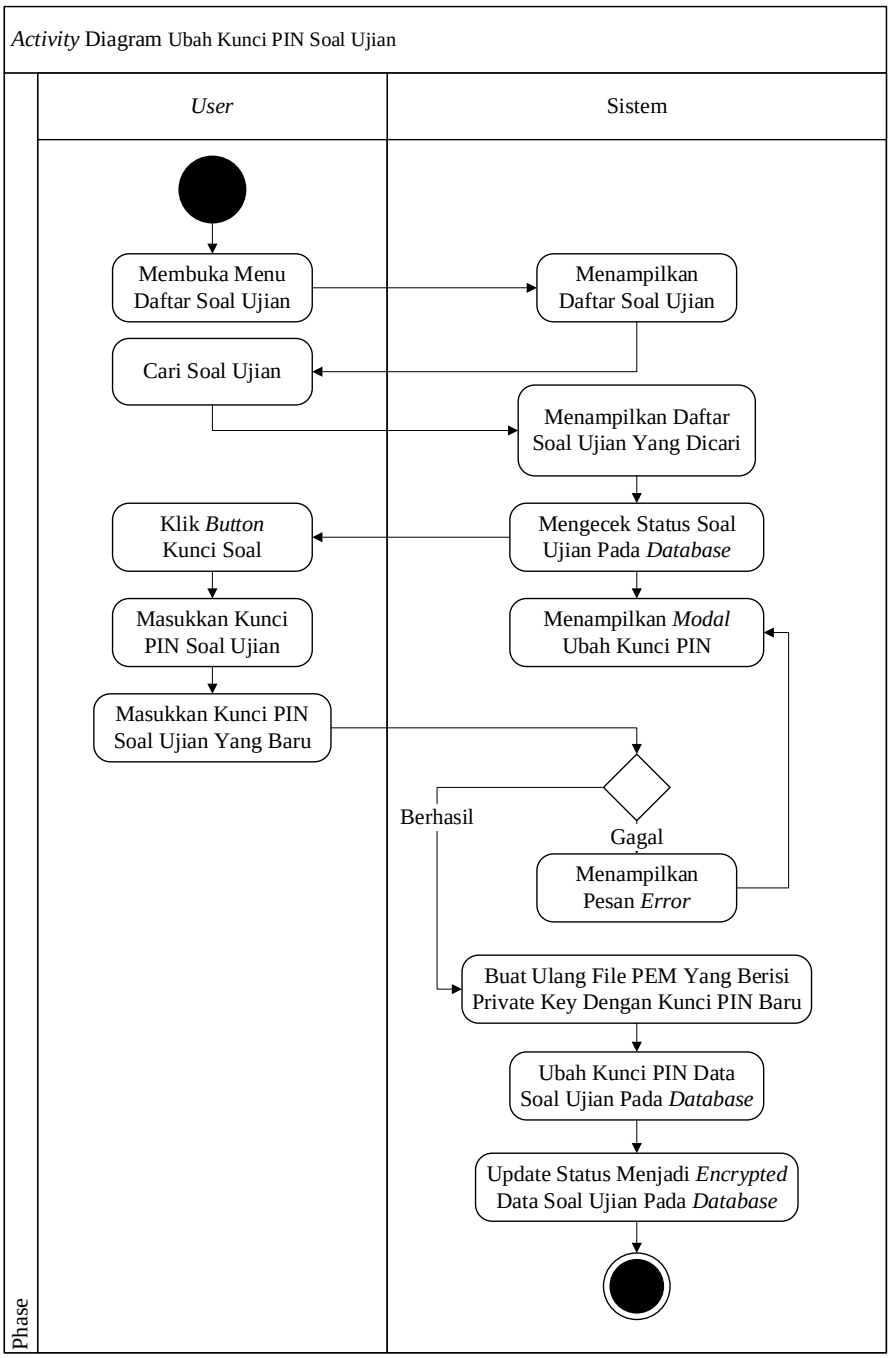
Aktivitas diagram *download* soal ujian menampilkan proses dalam melakukan *download* soal ujian sekolah, pada aktivitas ini menampilkan proses dekripsi data menggunakan *private key* yang dikunci menggunakan kunci PIN oleh *user*. Adapun aktivitas pada *download* soal ujian adalah sebagai berikut:



Gambar 3. 16. *Activity* Diagram *Download* Soal Ujian

8. *Activity* Diagram Ubah Kunci PIN Soal Ujian

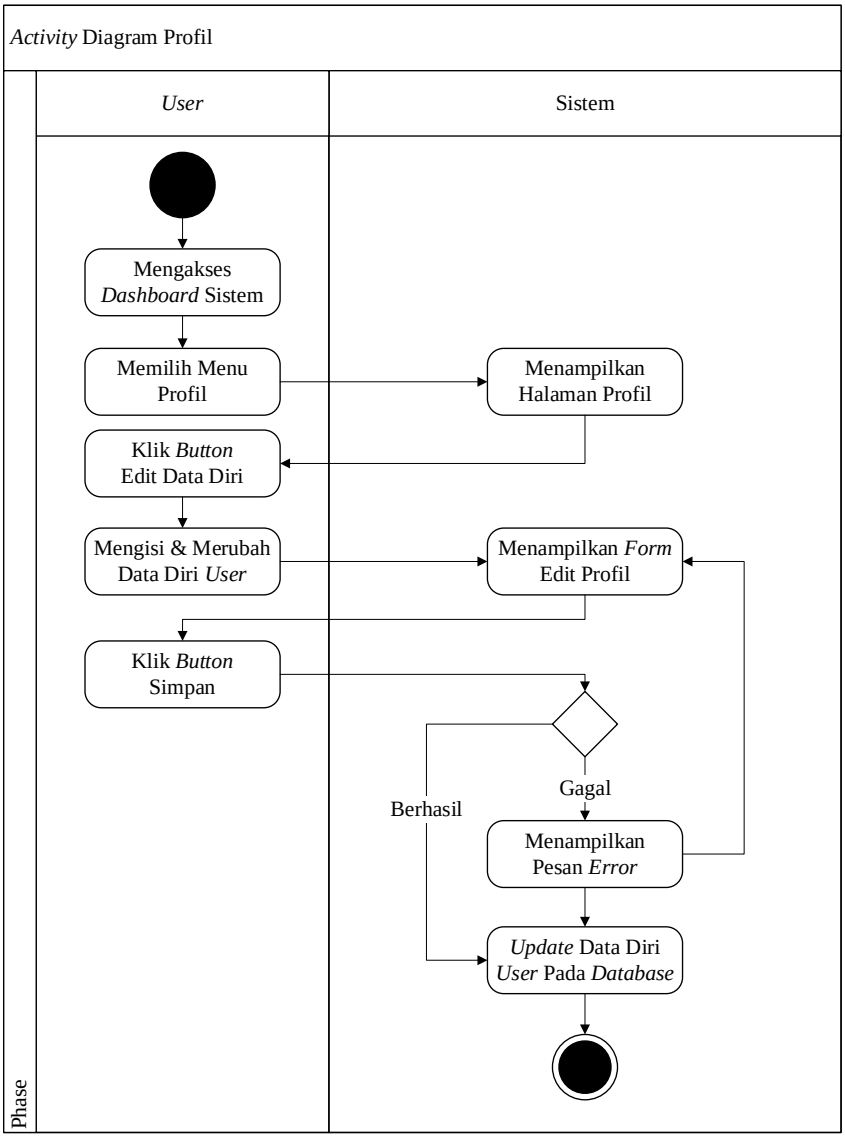
Aktivitas diagram ubah kunci PIN soal ujian menampilkan proses *user* mengubah kunci PIN lama yang telah dikirim kepada administrator dengan kunci PIN baru yang tidak diketahui. Adapun aktivitas pada ubah kunci PIN soal ujian adalah sebagai berikut:



Gambar 3.17. *Activity* Diagram Ubah Kunci PIN Soal Ujian

9. *Activity* Diagram Profil

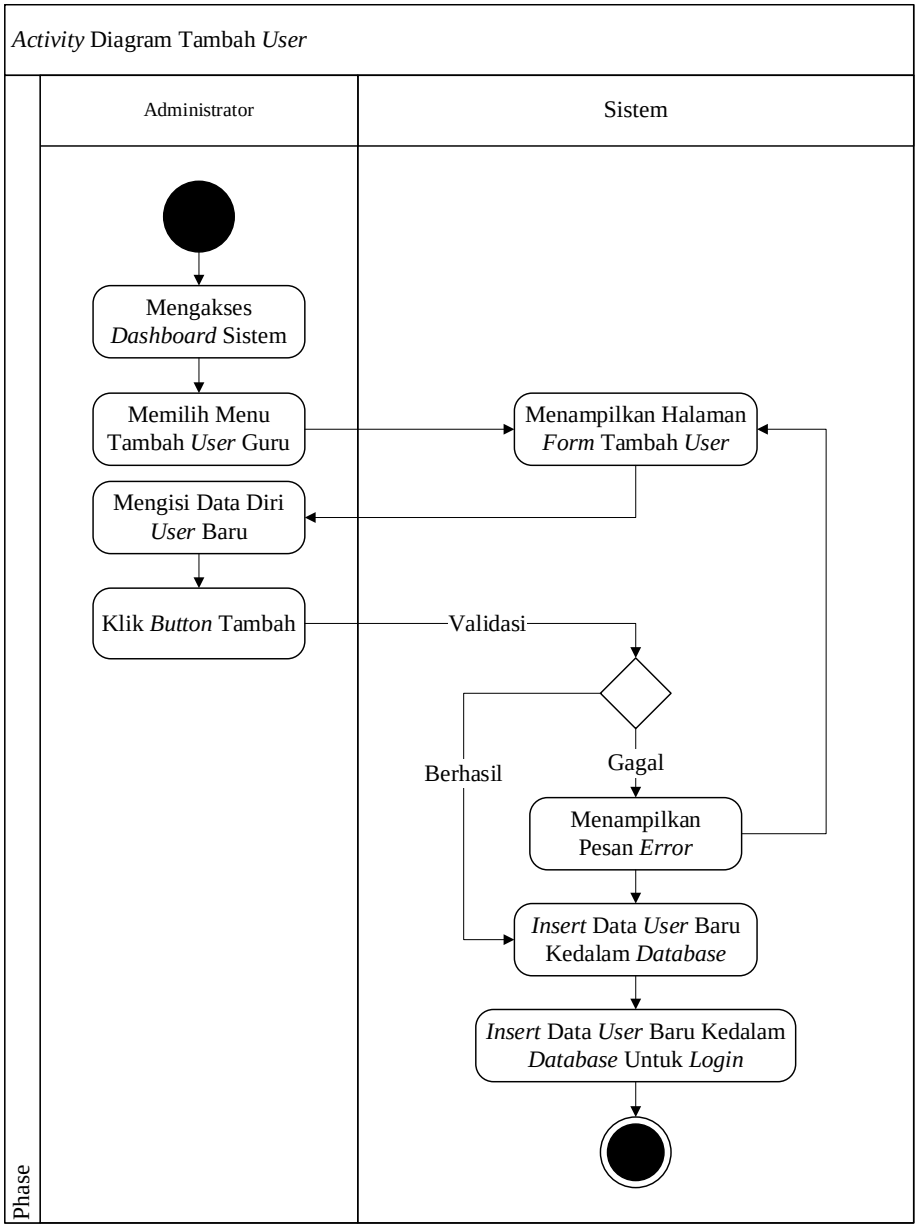
Aktivitas diagram profil adalah aktivitas untuk *user* melihat dan mengubah data diri. Adapun aktivitas diagram profil adalah sebagai berikut:



Gambar 3.18. *Activity* Diagram Profil

10. *Activity* Diagram Tambah *User*

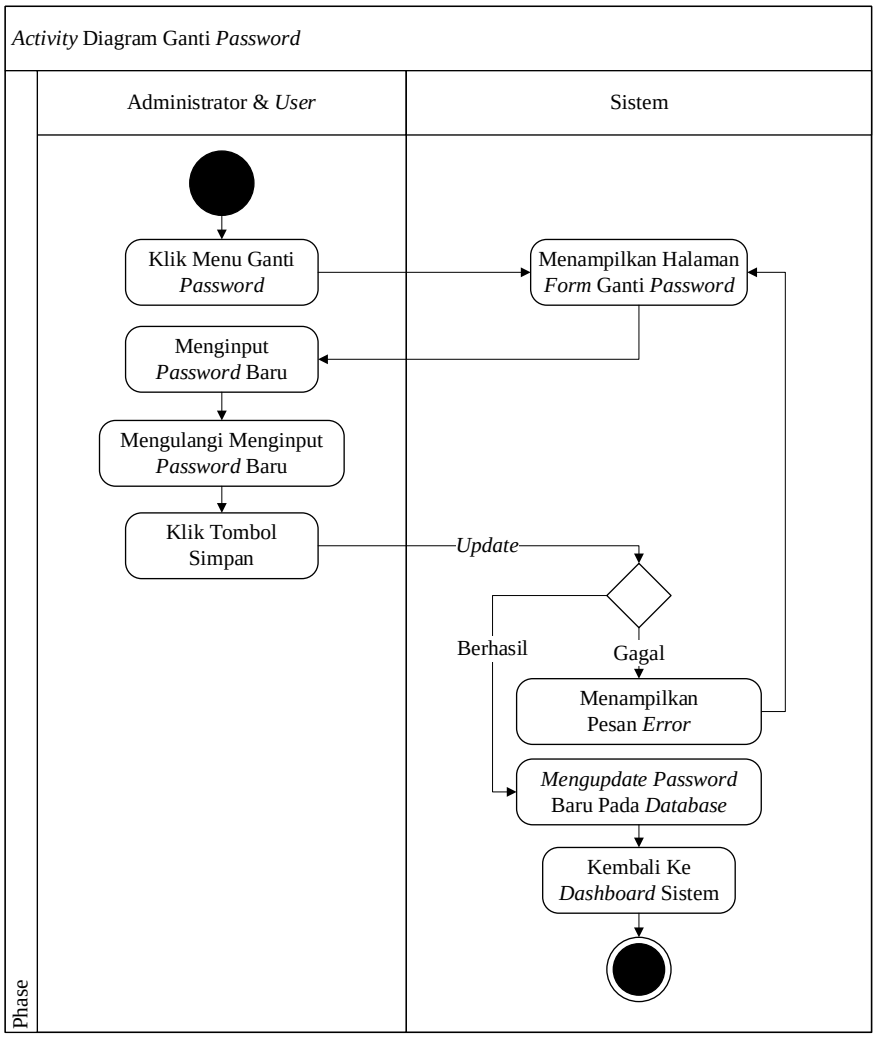
Aktivitas diagram tambah *user* adalah aktivitas untuk administrator untuk menambahkan *user*. Adapun aktivitas diagram untuk melakukan tambah *user* adalah sebagai berikut:



Gambar 3.19. *Activity* Diagram Tambah *User*

11. *Activity* Diagram Ganti *Password*

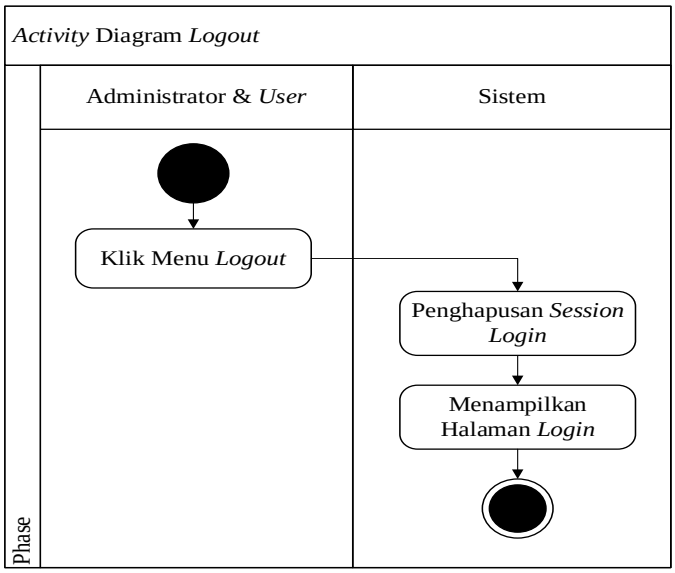
Aktivitas diagram ganti *password* adalah aktivitas untuk *user* ataupun administrator untuk mengganti *password login*. Adapun aktivitas diagram untuk melakukan ganti *password* adalah sebagai berikut:



Gambar 3.20. *Activity* Diagram Ganti *Password*

12. *Activity* Diagram *Logout*

Aktivitas diagram *logout* adalah aktivitas untuk *user* ataupun administrator untuk keluar dari sistem dan menghapus *session login* untuk keamanan sistem. Adapun aktivitas diagram untuk melakukan *logout* pada *user* dan administrator adalah sebagai berikut:



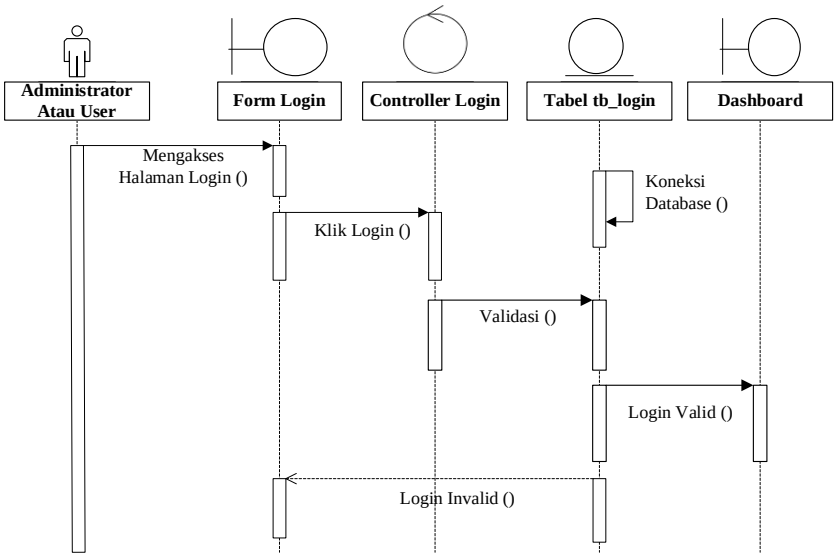
Gambar 3.21 *Activity* Diagram *Logout*

3.3.5. *Sequence* Diagram

Sequence diagram adalah jenis diagram dalam UML yang digunakan untuk memodelkan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu.

1. *Sequence* Diagram *Login*

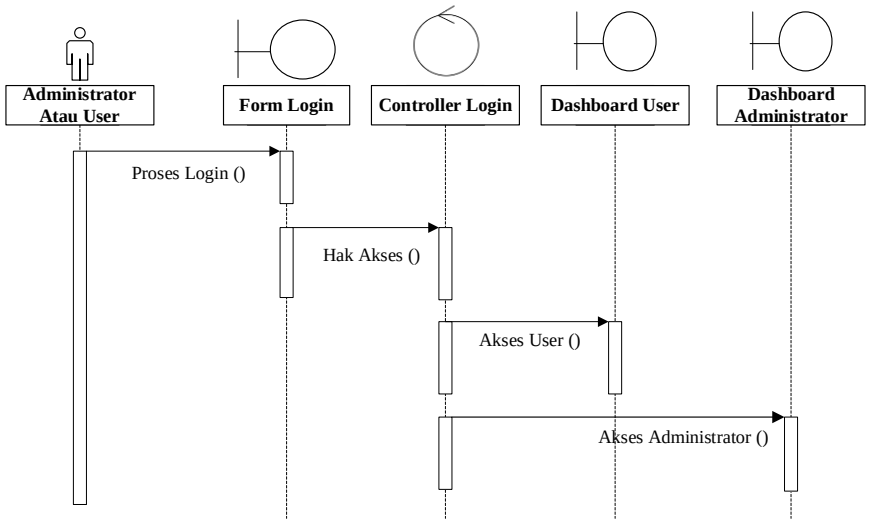
Serangkaian kerja melakukan proses *login* pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3. 22. *Sequence* Diagram *Login*

2. *Sequence* Diagram Tampilan Menu *Dashboard* Sistem

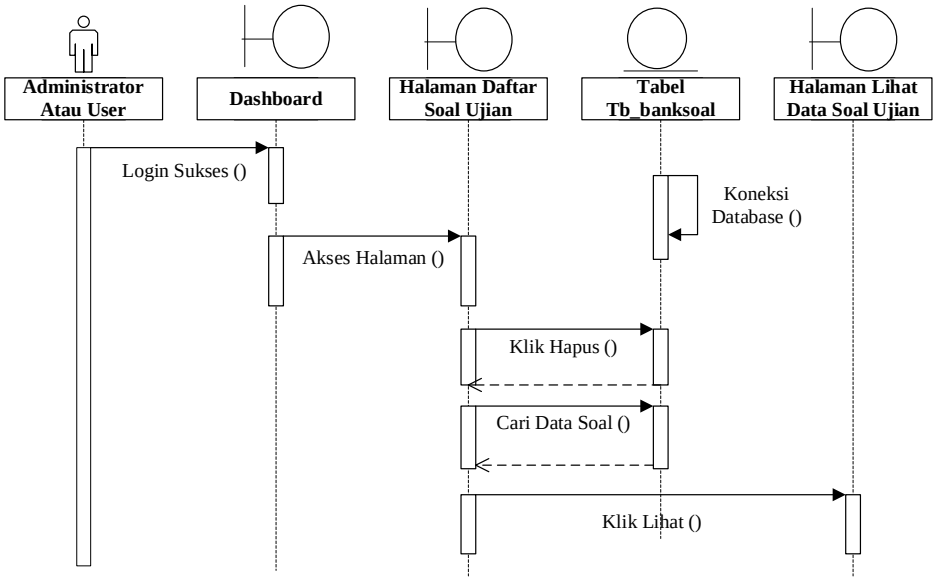
Serangkaian kerja melakukan proses tampilan menu *dashboard* sistem pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.23. *Sequence* Diagram Tampilan Menu *Dashboard* Sistem

3. *Sequence* Diagram Tampilan Daftar Soal Ujian

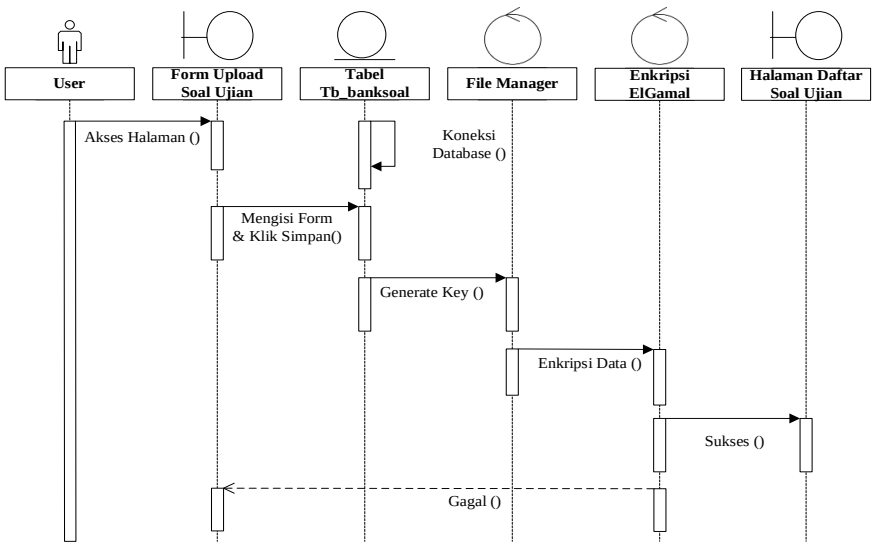
Serangkaian kerja melakukan proses tampilan daftar soal ujian pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.24. *Sequence* Diagram Tampilan Daftar Soal Ujian

4. *Sequence* Diagram *Upload* Soal Ujian

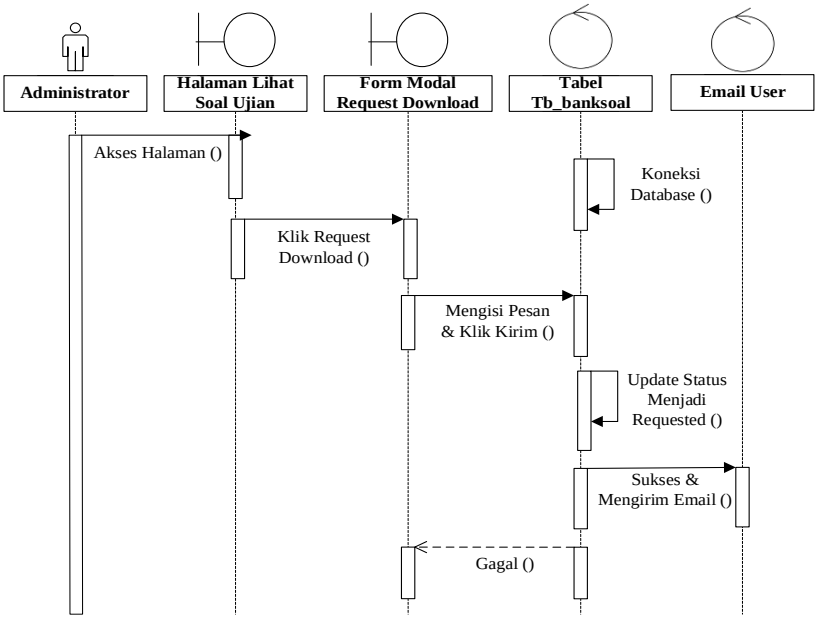
Serangkaian kerja melakukan proses *upload* soal ujian pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.25. *Sequence* Diagram *Upload* Soal Ujian

5. *Sequence* Diagram *Request Download* Soal Ujian

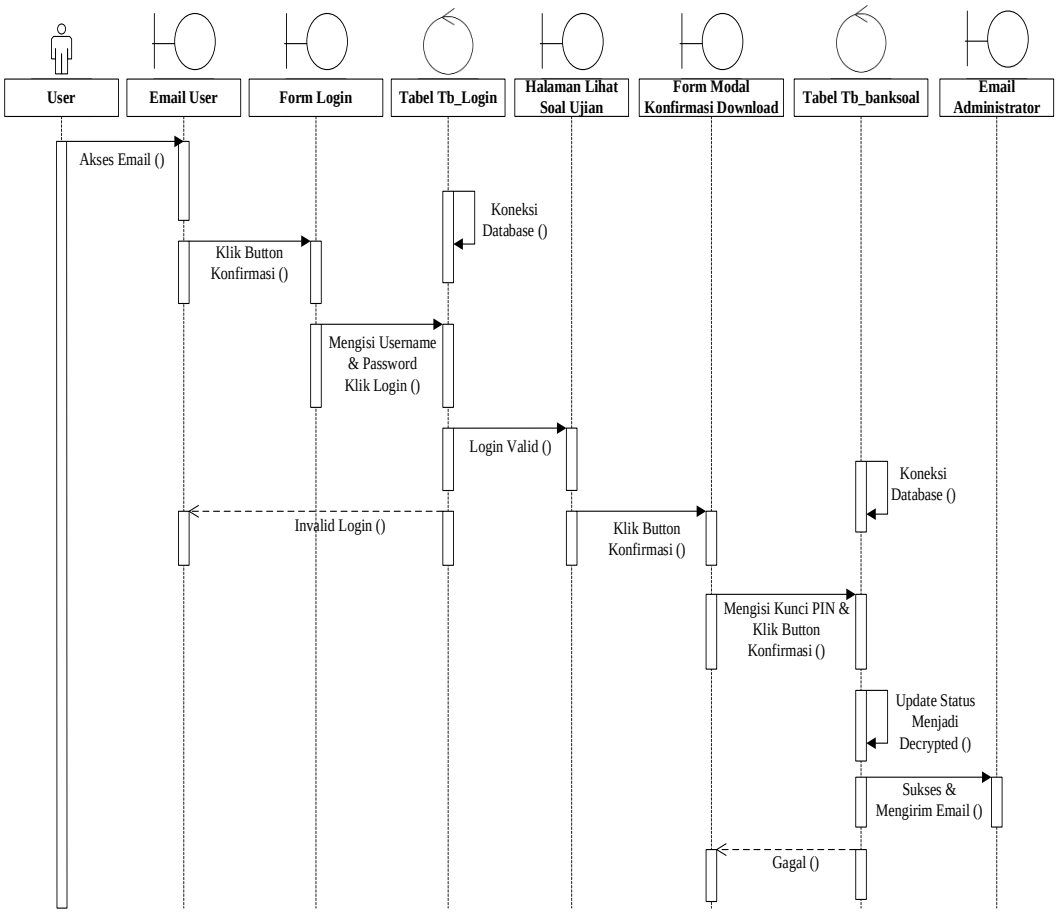
Serangkaian kerja melakukan proses *request download* soal ujian pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.26. *Sequence* Diagram *Request Download* Soal Ujian

6. *Sequence* Diagram Konfirmasi *Download* Soal Ujian

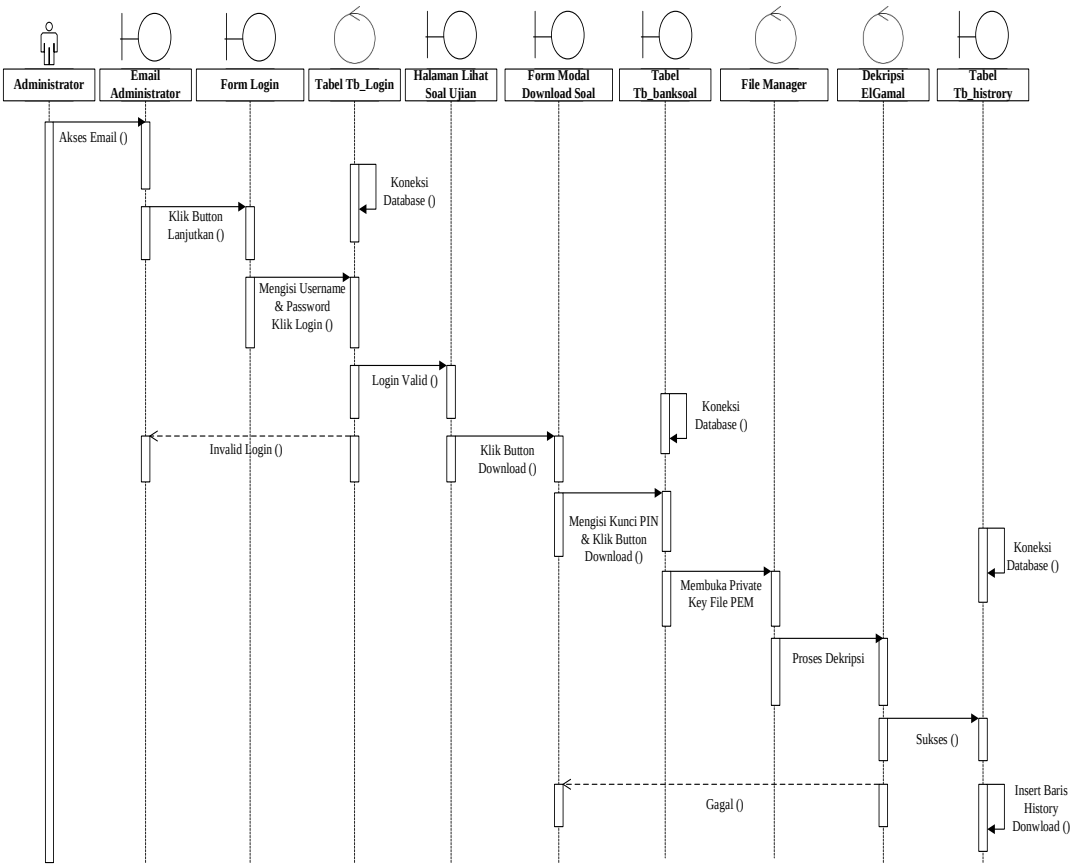
Serangkaian kerja melakukan proses konfirmasi *download* soal ujian pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.27. *Sequence* Diagram Konfirmasi *Download* Soal Ujian

7. *Sequence* Diagram *Download* Soal Ujian

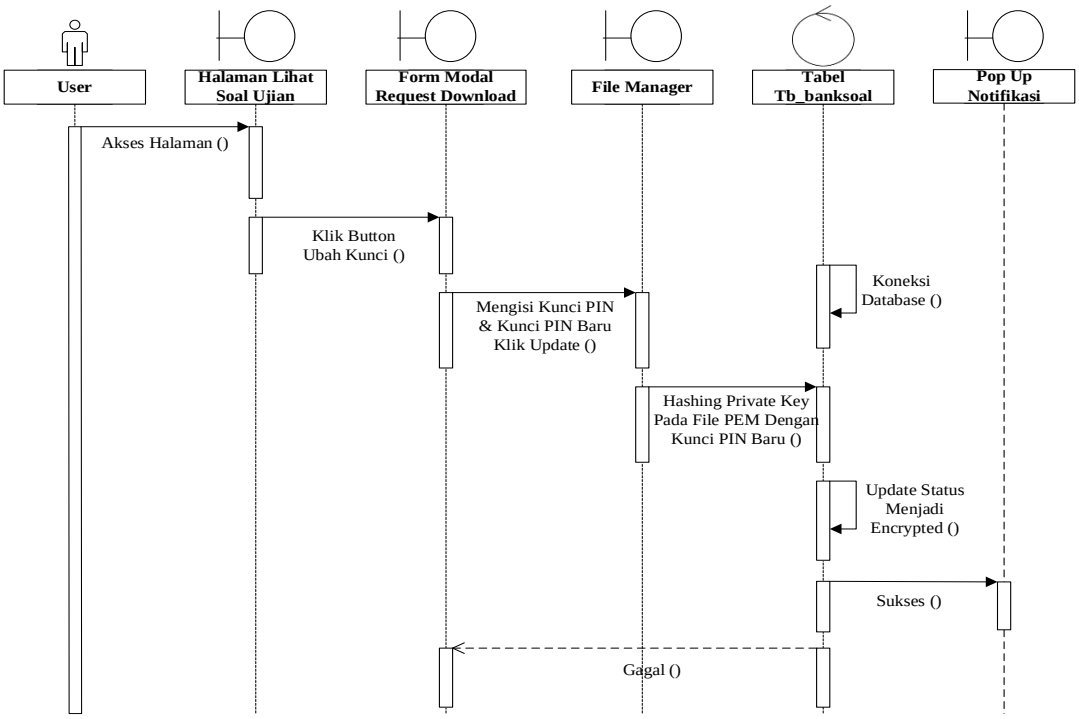
Serangkaian kerja melakukan proses *download* soal ujian pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.28. *Sequence* Diagram *Download* Soal Ujian

8. *Sequence* Diagram Ubah Kunci PIN Soal Ujian

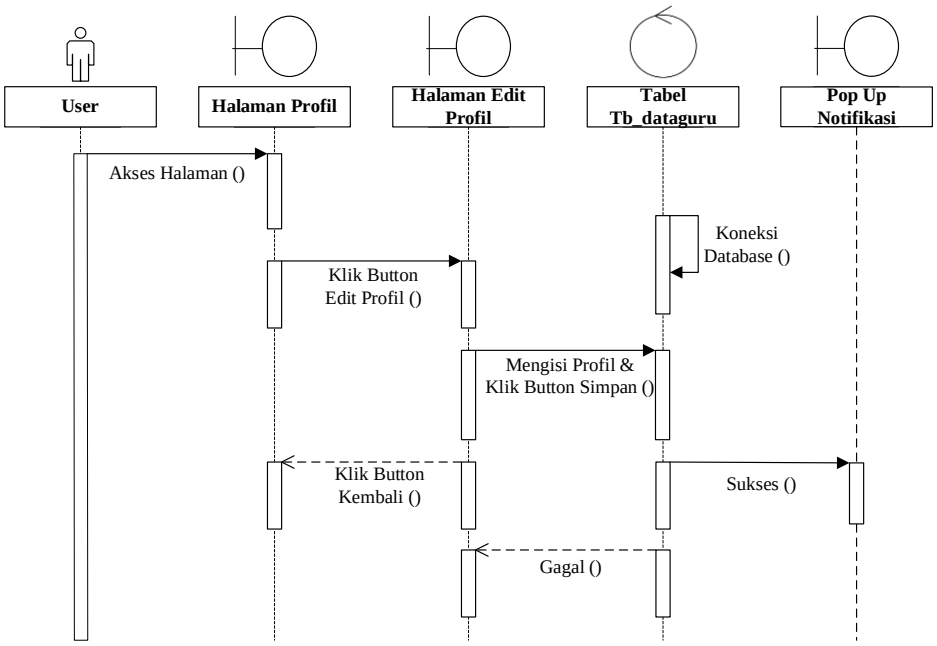
Serangkaian kerja melakukan proses ubah kunci PIN soal ujian pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.29. *Sequence* Diagram Ubah Kunci PIN Soal Ujian

9. *Sequence* Diagram Profil

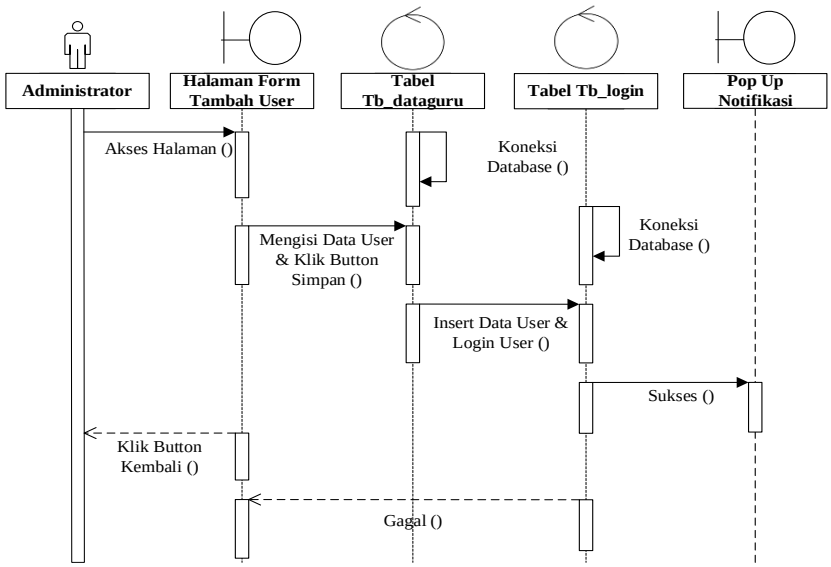
Serangkaian kerja melakukan proses profil pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.30. *Sequence* Diagram Profil

10. *Sequence* Diagram Tambah *User*

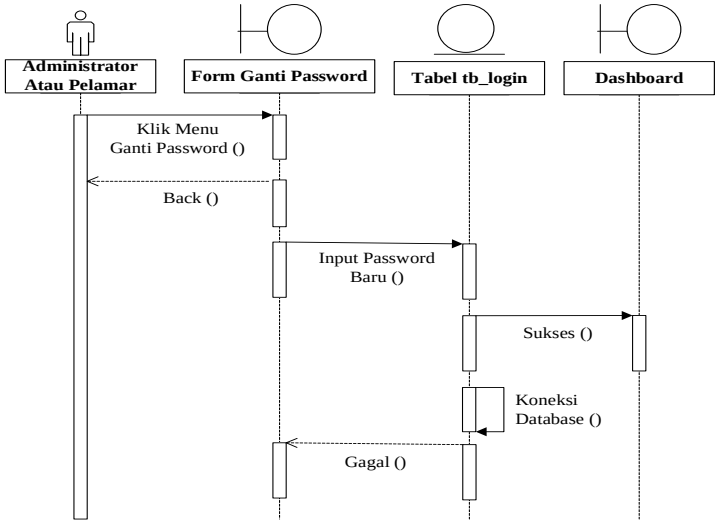
Serangkaian kerja melakukan proses tambah *user* pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.31. *Sequence* Diagram Tambah *User*

11. *Sequence* Diagram Ganti *Password*

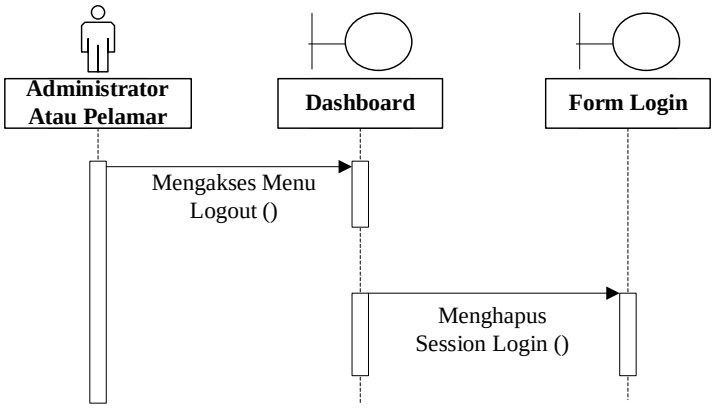
Serangkaian kerja melakukan proses ganti *password* pada *sequence* diagram dapat dilihat pada gambar berikut ini:



Gambar 3.32. *Sequence* Diagram Ganti *Password*

12. *Sequence* Diagram *Logout*

Serangkaian kerja melakukan proses *logout* pada *sequence* diagram dapat dilihat pada gambar berikut ini:

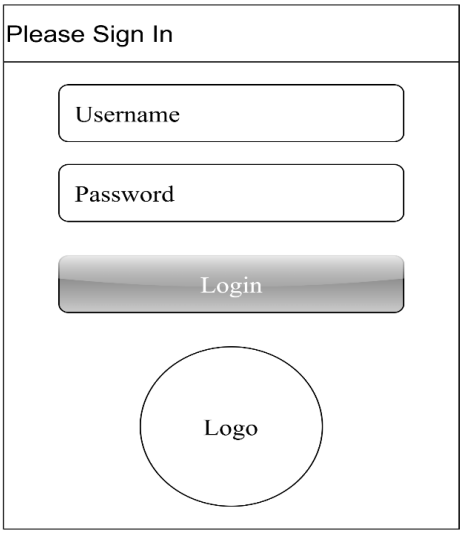


Gambar 3.33. *Sequence* Diagram *Logout*

3.4. Desain *Interface* Sistem

3.4.1. Desain *Interface Form Login*

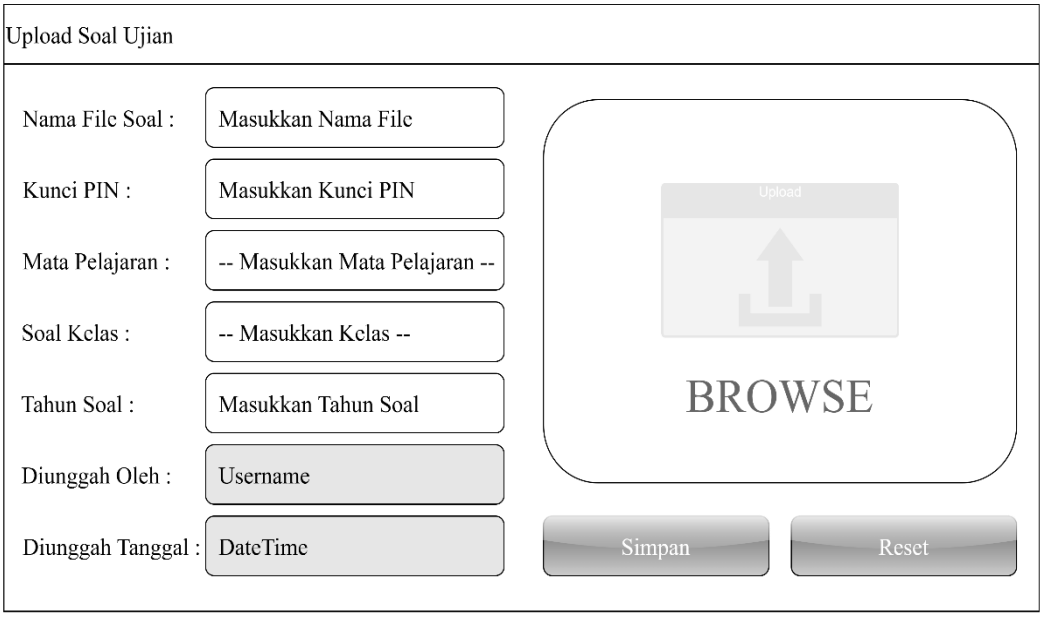
Desain tampilan *form login* berfungsi untuk *user* dan administrator melakukan proses masuk kedalam sistem. Adapun perancangan *form login* dapat dilihat sebagai berikut:



Gambar 3.34. Desain *Interface Form Login*

3.4.2. Desain *Interface Form Upload* Soal Ujian

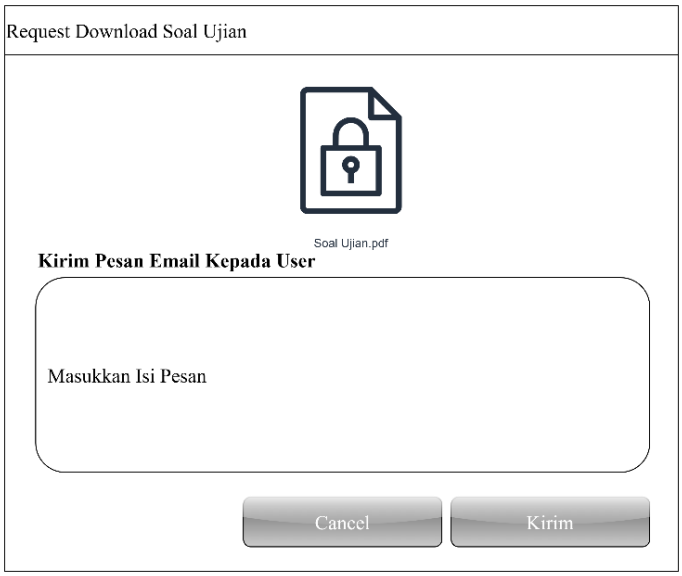
Desain *form upload* soal ujian berfungsi untuk *user* melakukan proses pengunggahan atau *upload* data soal ujian sekolah kedalam sistem beserta membuat kunci PIN yang bertujuan untuk mengunci *private key* kedalam *file* PEM ketika dibangkitkan. Adapun perancangan *form upload* soal ujian dapat dilihat sebagai berikut:



Gambar 3.35. Desain *Interface Form Upload* Soal Ujian

3.4.3. Desain *Interface Form Request Download* Soal Ujian

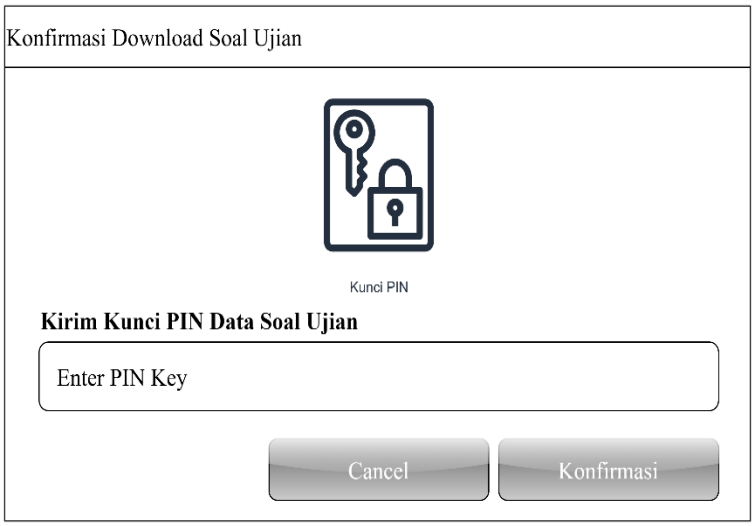
Desain *form request download* soal ujian berfungsi untuk administrator melakukan proses permintaan kunci PIN sebagai akses membuka kunci *private key* pada *file* PEM untuk proses dekripsi data. Adapun perancangan *form request download* soal ujian dapat dilihat sebagai berikut:



Gambar 3.36. Desain *Interface Form Request Download* Soal Ujian

3.4.4. Desain *Interface Form* Konfirmasi *Download* Soal Ujian

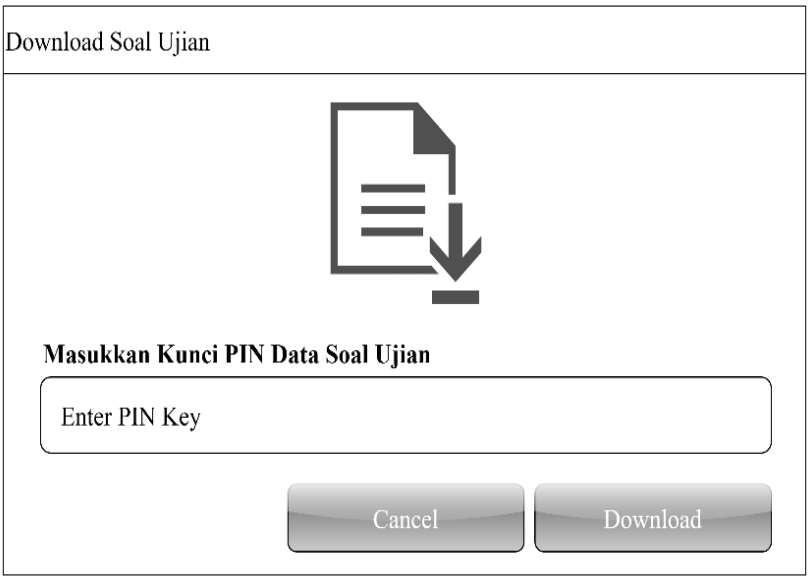
Desain *form request download* soal ujian berfungsi untuk *user* melakukan pemberian izin dan mengirimkan kunci PIN sebagai akses membuka kunci *private key* pada *file* PEM kepada administrator untuk proses dekripsi data. Adapun perancangan *form* konfirmasi *download* soal ujian dapat dilihat sebagai berikut:



Gambar 3.37. Desain *Interface Form* Konfirmasi *Download* Soal Ujian

3.4.5. Desain *Interface Form Download* Soal Ujian

Desain *form download* soal ujian berfungsi untuk administrator melakukan proses dekripsi data dengan menggunakan *private key* yang telah dibuka menggunakan kunci PIN, Ketika proses dekripsi data sukses maka data soal ujian sekolah akan terunduh (*download*) secara otomatis. Adapun perancangan *form download* soal ujian dapat dilihat sebagai berikut:



Gambar 3.38. Desain *Interface Form Download* Soal Ujian

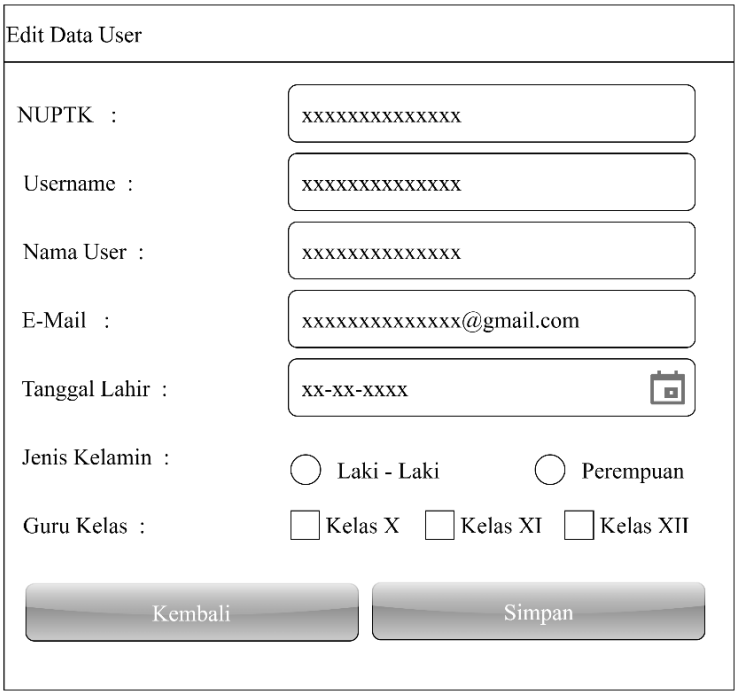
3.4.6. Desain *Interface Form* Ubah Kunci PIN Soal Ujian

Desain *form* ubah kunci soal ujian berfungsi untuk *user* melakukan proses mengubah kunci PIN yang telah diberikan kepada Administrator untuk tujuan keamanan data, kunci PIN yang diubah akan mengganti kunci PIN sebelumnya. Adapun perancangan *form* ubah kunci soal ujian dapat dilihat sebagai berikut:

Gambar 3.39. Desain *Interface Form* Ubah Kunci PIN Soal Ujian

3.4.7. Desain *Interface Form* Edit Profil

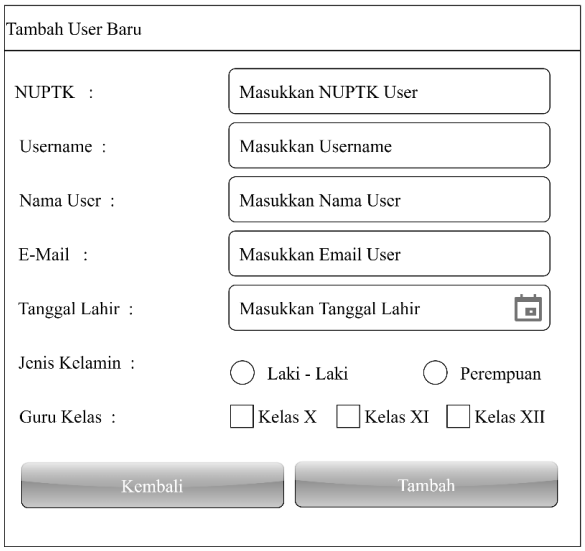
Desain *form* profil berfungsi untuk *user* melakukan proses mengubah informasi data diri. Adapun perancangan *form* profil dapat dilihat sebagai berikut:



Gambar 3.40. Desain *Interface Form* Edit Profil

3.4.8. Desain *Interface Form* Tambah *User*

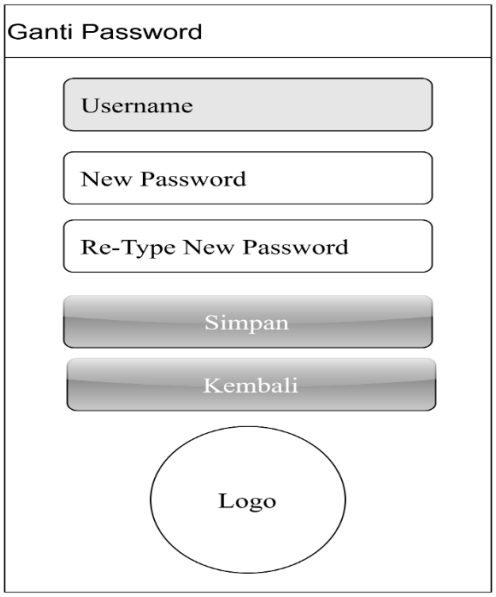
Desain *form* tambah *user* berfungsi untuk administrator melakukan proses menambah *user*. Adapun perancangan *form* profil dapat dilihat sebagai berikut:



Gambar 3.41. Desain *Interface Form* Tambah *User*

3.4.9. Desain *Interface Form* Ganti *Password*

Desain *form* ganti *password* berfungsi untuk *user* dan administrator untuk mengganti password *login*. Adapun perancangan *form* ganti *password* adalah:



Gambar 3.42. Desain *Interface Form* Ganti *Password*

BAB IV
HASIL DAN UJI COBA

4.1. Hasil

Pada tahapan ini penulis memunculkan beberapa hasil tampilan akhir sistem yang telah didesain pada bab sebelumnya, yaitu tampilan halaman *form*, tampilan halaman *list* data dan tampilan halaman informasi.

Secara garis besar, sistem ini dapat dijalankan oleh 2 pengguna, yaitu:

1. *User* sebagai guru yang bertugas untuk melakukan proses pengunggahan (*upload*) data soal ujian sekolah. Dalam hal ini, *user* berperan penting dalam proses enkripsi data soal ujian dikarenakan dalam proses *upload* data soal ujian terdapat proses pembangkitan kunci dan proses enkripsi algoritma ElGamal.
2. Administrator sebagai pihak sekolah yang bertugas untuk mengelola dan melakukan proses pengunduhan (*download*) data soal ujian sekolah yang telah di unggah oleh *user*. Dalam hal ini, administrator berperan penting dalam proses dekripsi data soal ujian dikarenakan dalam proses *download* terdapat proses pertukaran informasi untuk mengakses private key dan proses dekripsi algoritma ElGamal.

Setiap pengguna juga dibatasi dalam mengakses setiap tampilan halaman sistem sesuai dengan hak akses yang telah diberikan oleh sistem saat pengguna tersebut melakukan *login.* Adapun beberapa hasil tampilan halaman sistem tersebut adalah sebagai berikut:

4.1.1. Tampilan Halaman *Form*

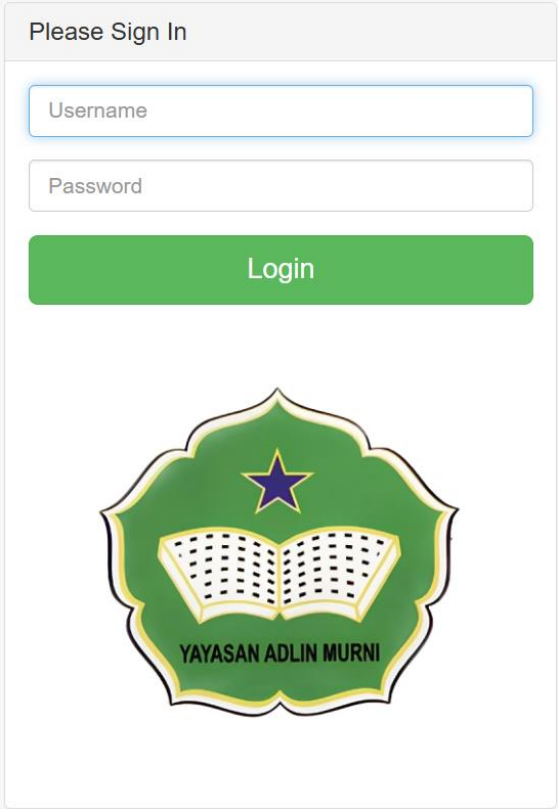
Tampilan halaman *form* adalah halaman sistem yang menampilkan beberapa inputan atau *field* yang harus diisi oleh *user* maupun adminstrator untuk

keperluan sistem yang berjalan. Adapun tampilan halaman *form* pada sistem ini adalah sebagai berikut:

1. Tampilan Halaman *Form* Pada *User*

   a. Tampilan Halaman *Form Login* Pada *User*

   Tampilan halaman *form login* pada *user* berfungsi untuk *user* melakukan proses masuk kedalam sistem, sistem akan memvalidasi *username* dan *password* terlebih dahulu sebelum mengizikan *user* masuk kedalam sistem dan akan memberikan hak akses sebagai *user*. Adapun hasil tampilan halaman *form login* pada *user* adalah sebagai berikut:



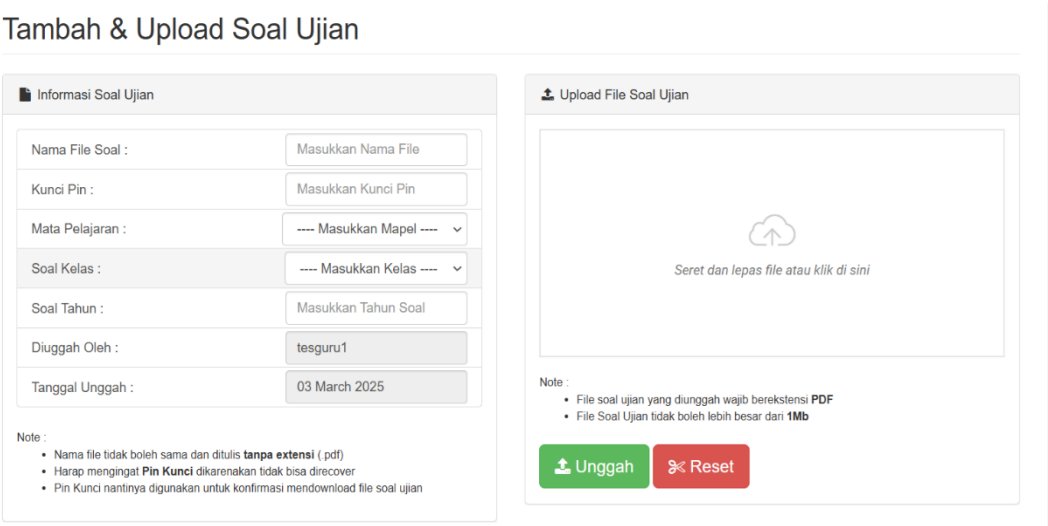
Gambar 4.1 Tampilan Halaman *Form Login* Pada *User*

Pada *form login* tersebut, *user* diminta untuk memasukkan *username* dan *password* pada inputan *form* yang tersedia. Selanjutnya, sistem akan melakukan verifikasi dengan mencocokkan data inputan *user* dengan data

yang tersimpan pada tabel *tb_login* di dalam *database*. Apabila data yang dimasukkan sesuai, maka *user* akan diberikan hak akses untuk masuk ke dalam sistem. Sebaliknya, jika data tidak valid, sistem akan menampilkan notifikasi kesalahan sebagai umpan balik kepada *user*.

b. Tampilan Halaman *Form Upload* Soal Ujian

Tampilan halaman *form upload* soal ujian berfungsi untuk *user* melakukan proses pengunggahan (*upload*) *file* soal ujian sekolah kedalam sistem beserta membuat kunci PIN yang bertujuan untuk mengunci *private key* kedalam *file* PEM ketika dibangkitkan. Pada tahap ini juga proses enkripsi algoritma kriptografi ElGamal dijalankan. Adapun hasil tampilan halaman *form upload* soal ujian adalah sebagai berikut:



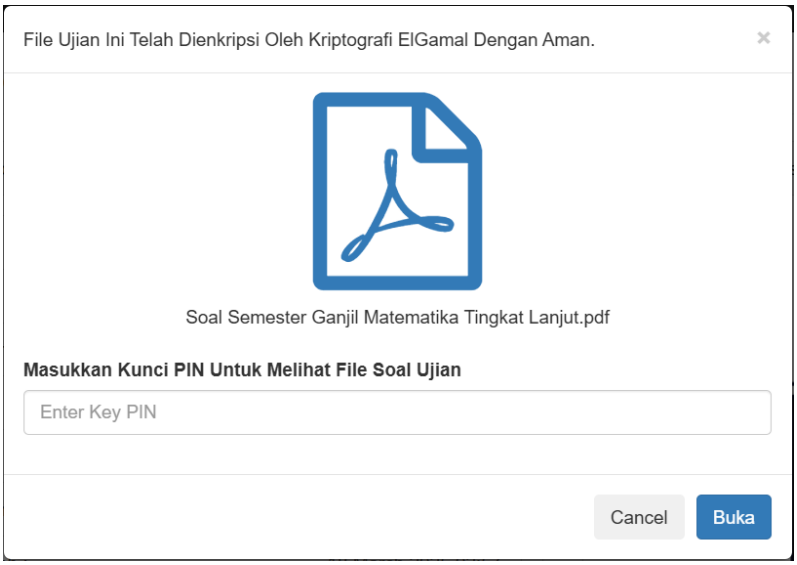
Gambar 4.2 Tampilan Halaman *Form Upload* Soal Ujian

Pada *form upload* soal ujian tersebut, *user* dapat melakukan aktivitas menguggah *file* soal ujian. Pada proses ini *user* diharuskan mengisi seluruh inputan dalam form untuk dapat dilakukan pendataan dan penyimpanan didalam *database* sistem. Pada tahap ini user wajib untuk mengingat kunci

PIN untuk dapat membuka *private key* pada proses dekripsi. Pada aktivitas ini juga terjadi proses pembangkitan kunci (*generate key*) dan juga proses enkripsi menggunakan algoritma kriptografi ElGamal pada *file* soal ujian. Seluruh data yang diinput akan disimpan dalam *tb_banksoal* pada *database*.

c. Tampilan Halaman *Form* Lihat *File* Soal Ujian

Tampilan halaman *form* lihat *file* soal ujian berfungsi untuk *user* melakukan proses melihat dan mengecek *file* soal ujian sekolah yang telah diunggah. Adapun hasil tampilan halaman *form* lihat *file* soal ujian adalah sebagai berikut:



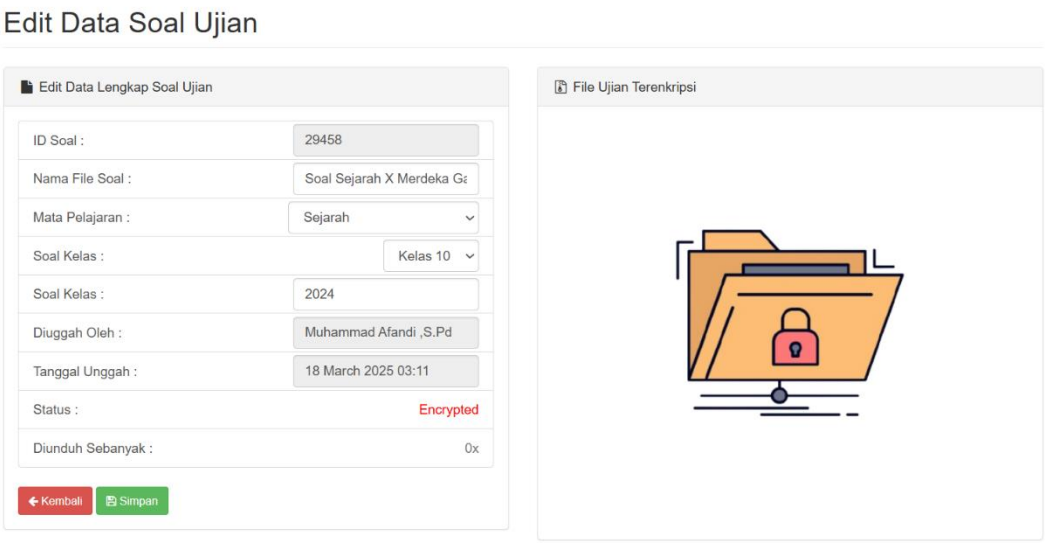
Gambar 4.3. Tampilan Halaman *Form* Lihat *File* Soal Ujian

Pada *form* lihat *file* soal ujian tersebut, *user* dapat melakukan aktivitas melihat dan mengecek *file* soal ujian sekolah yang telah diunggah sebelumnya. Pada proses ini *user* diharuskan mengisi inputan kunci PIN yang sebelumnya telah dibuat pada proses pengunggahan data soal ujian. Pada proses ini juga sistem dengan otomatis melakukan proses dekripsi algoritma

kriptografi ElGamal dengan membuka *private key* menggunakan kunci PIN yang dibuat oleh *user*.

d. Tampilan Halaman *Form* Edit Data Soal Ujian

Tampilan halaman *form* edit data soal ujian berfungsi untuk *user* melakukan proses mengedit atau mengubah data soal ujian. Adapun hasil tampilan halaman *form* edit data soal ujian adalah sebagai berikut:



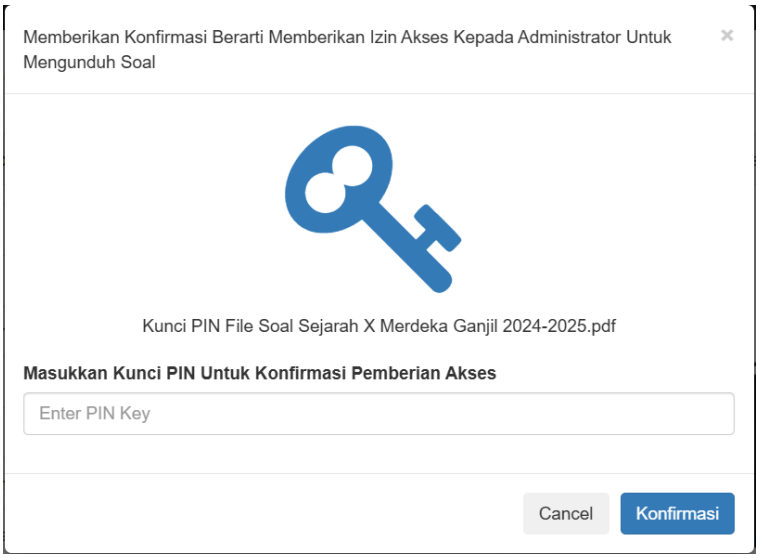
Gambar 4.4. Tampilan Halaman *Form* Edit Data Soal Ujian

Pada *form* edit data soal ujian tersebut, *user* dapat melakukan aktivitas mengubah atau mengedit data soal ujian yang sebelumnya diinput pada *form upload* soal ujian. Pada form tersebut terdapat beberapa inputan yang dapat mengedit data soal ujian seperti nama file soal, mata pelajaran, soal kelas dan soal tahun.

e. Tampilan Halaman *Form* Konfirmasi *Download* Soal Ujian

Tampilan halaman *form* konfirmasi *download* soal ujian berfungsi untuk *user* melakukan proses konfirmasi persetujuan agar file soal ujian dapat diunduh (*download*) oleh administrator dan mengirim kunci PIN melalui

email kepada administrator. Adapun hasil tampilan halaman *form* konfirmasi *download* soal ujian adalah sebagai berikut:



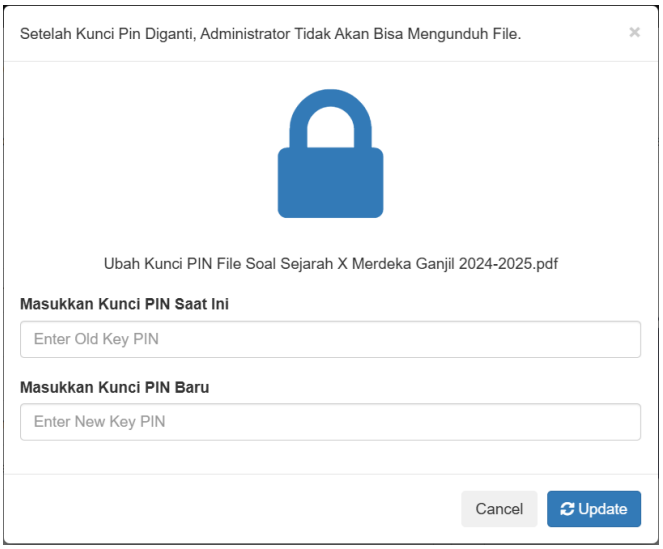
Gambar 4.5. Tampilan Halaman *Form* Konfirmasi *Download* Soal Ujian

Pada *form* konfirmasi *download* soal ujian tersebut, *user* dapat memberikan izin akses kepada administrator untuk dapat melakukan *download* pada *file* soal ujian miliknya. Pada proses ini *user* diharuskan mengisi inputan kunci PIN yang sebelumnya telah dibuat pada proses pengunggahan data soal ujian. Pada proses ini juga sistem akan mengirimkan kunci PIN kepada administrator melalui email yang telah terdaftar untuk keperluan proses dekripsi algoritma kriptografi ElGamal dengan membuka *private key* menggunakan kunci PIN yang dibuat oleh *user*.

f. Tampilan Halaman *Form* Ubah Kunci PIN Soal Ujian

Tampilan halaman *form* ubah kunci PIN soal ujian berfungsi untuk *user* melakukan proses mengubah atau mengganti kunci PIN yang telah diberikan kepada Administrator melalui email sebagai keamanan tambahan pada data

soal ujian milik *user*. Adapun hasil tampilan halaman *form* ubah kunci PIN soal ujian adalah sebagai berikut:

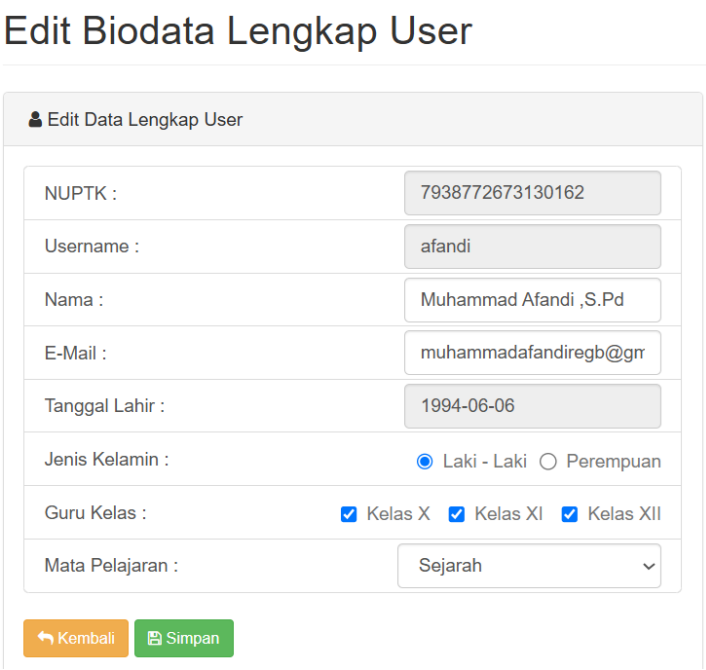


Gambar 4.6. Tampilan Halaman *Form* Ubah Kunci PIN Soal Ujian

Pada *form* ubah kunci PIN soal ujian tersebut, *user* dapat mengganti kunci PIN yang sebelumnya telah diketahui oleh administrator. Pada proses ini *user* diharuskan mengisi inputan kunci PIN lama dan menginput kunci PIN yang baru. Pergantian kunci PIN ini bertujuan agar administrator tidak dapat melakukan aktivitas *download file* soal ujian menggunakan kunci PIN lama, namun harus melakukan proses *request* kembali.

g. Tampilan Halaman *Form* Edit Profil

Tampilan halaman *form* edit profil berfungsi untuk *user* melakukan proses mengubah atau mengganti profil data dirinya didalam sistem. Adapun hasil tampilan halaman *form* edit profil adalah sebagai berikut:

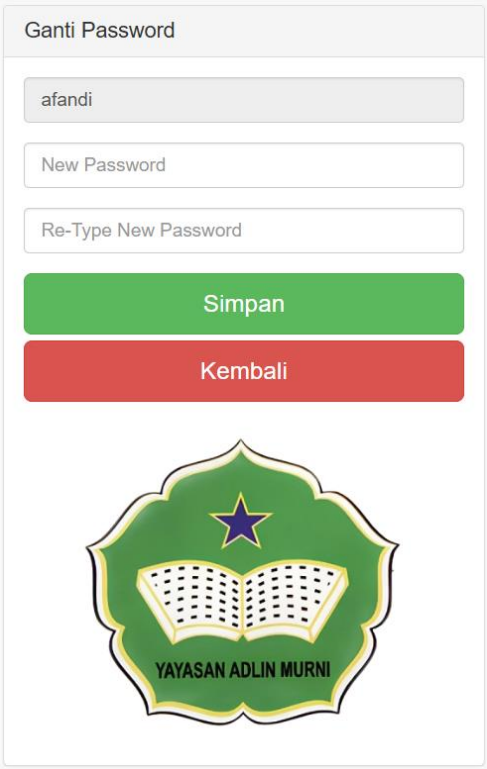


Gambar 4.7. Tampilan Halaman *Form* Edit Profil

Pada *form* halaman *form* edit profil tersebut, *user* dapat melakukan aktivitas mengisi atau mengedit data profil *user*. Pada *form* tersebut terdapat beberapa inputan yang dapat diubah seperti nama lengkap, email, jenis kelamin, guru kelas dan guru mata pelajaran.

h. Tampilan Halaman *Form* Ganti *Password* Pada *User*

Tampilan halaman *form* ganti *password* pada *user* berfungsi untuk *user* melakukan proses mengubah atau mengganti *password* saat melakukan *login*. Adapun hasil tampilan halaman *form* ganti *password* pada *user* adalah sebagai berikut:



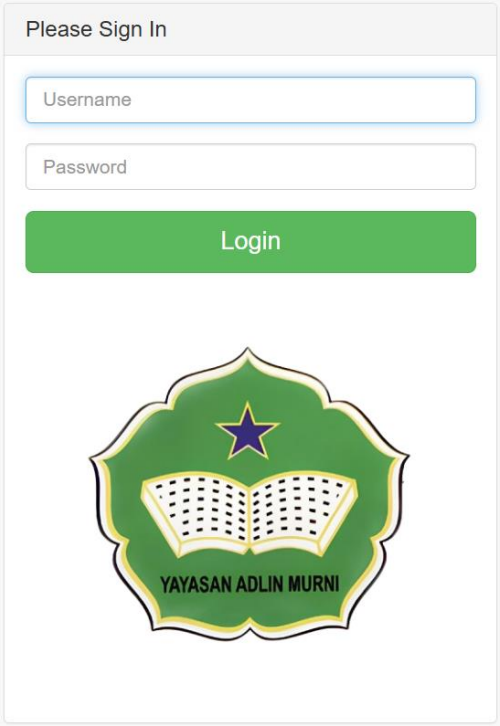
Gambar 4.8. Tampilan Halaman *Form* Ganti *Password* Pada *User*

Pada *form* ganti *password* tersebut, *user* dapat melakukan pergantian *password* pada aktivitas *login* untuk alasan keamanan. *User* akan diminta untuk memasukkan *password* baru dan mengulangi pengisian *password* baru, kemudian sistem akan melakukan *update* pada *tb_login* di dalam *database*.

2. Tampilan Halaman *Form* Pada Administrator

a. Tampilan Halaman *Form Login* Pada Administrator

Tampilan halaman *form login* pada administrator berfungsi untuk administrator melakukan proses masuk kedalam sistem, sistem akan memvalidasi *username* dan *password* terlebih dahulu sebelum mengizikan administrator masuk kedalam sistem dan akan memberikan hak akses sebagai administrator. Adapun hasil tampilan halaman *form login* pada administrator adalah sebagai berikut:

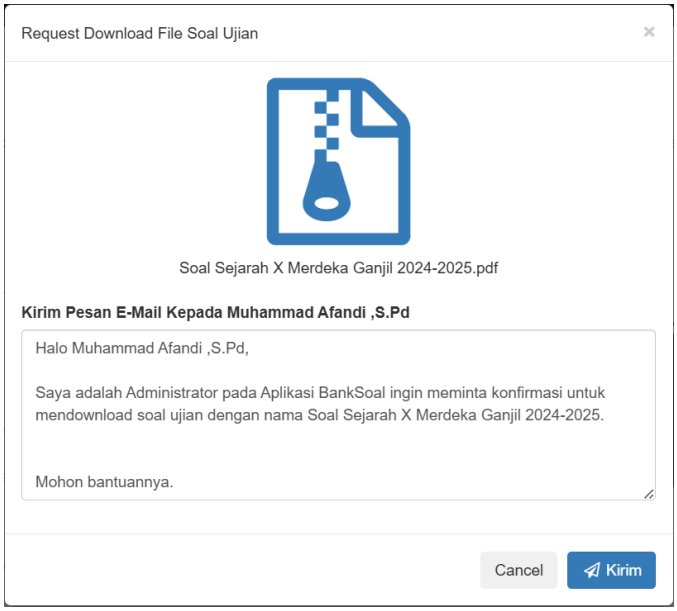


Gambar 4.9. Tampilan Halaman *Form Login* Pada Administrator

Pada *form login* tersebut, administrator diminta untuk memasukkan *username* dan *password* pada inputan *form* yang tersedia. Selanjutnya, sistem akan melakukan verifikasi dengan mencocokkan data inputan administrator dengan data yang tersimpan pada tabel *tb_login* di dalam *database*.

b. Tampilan Halaman *Form Request Download* Soal Ujian

Tampilan halaman *form request download* soal ujian berfungsi untuk administrator melakukan proses permintaan (*request*) persetujuan kepada *user* dengan mengirimkan kunci PIN melalui email yang nantinya digunakan untuk mengunduh (*download*) *file* soal ujian. Adapun hasil tampilan halaman *form request download* soal ujian adalah sebagai berikut:

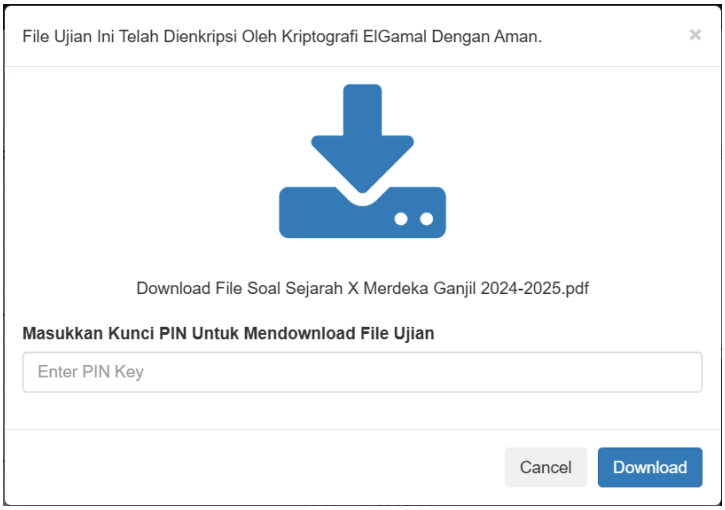


Gambar 4.10. Tampilan Halaman *Form Request Download* Soal Ujian

Pada *form request download* soal ujian tersebut, administrator dapat meminta izin akses kepada *user* untuk dapat melakukan *download* pada *file* soal ujian. Pada proses ini administrator diharuskan mengirimkan pesan kepada *user* dengan mengisi *form* diatas yang nantinya akan dikirimkan kepada *user* melalui email *user* yang telah terdaftar. Proses *request* ini bertujuan agar *user* nantinya dapat mengirimkan kunci PIN kepada administrator untuk keperluan proses dekripsi algoritma kriptografi ElGamal untuk membuka *private key* menggunakan kunci PIN yang dikirim oleh *user*.

c. Tampilan Halaman *Form Download* Soal Ujian

Tampilan halaman *form download* soal ujian berfungsi untuk administrator melakukan proses *download file* soal ujian sekolah menggunakan kunci PIN untuk membuka *private key* yang tersimpan pada *file* PEM. Pada tahap ini juga proses dekripsi algoritma kriptografi ElGamal dijalankan. Adapun hasil tampilan halaman *form download* soal ujian adalah:

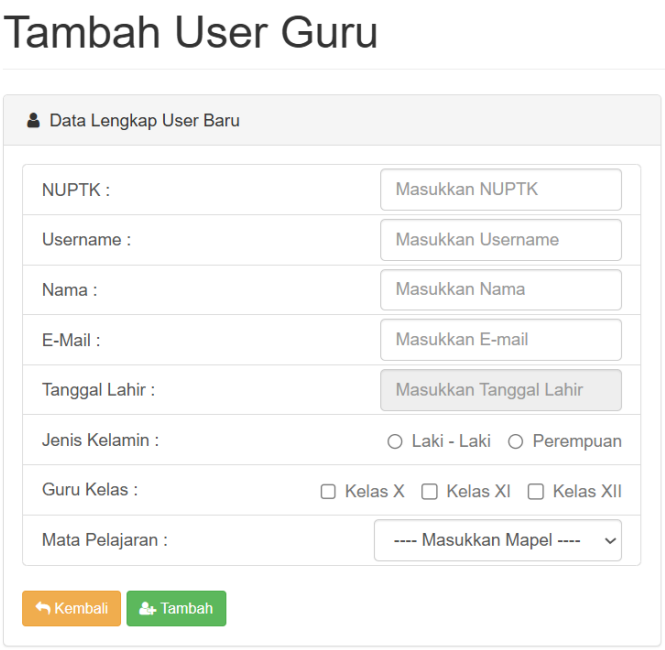


Gambar 4.11. Tampilan Halaman *Form Download* Soal Ujian

Pada *form download* soal ujian tersebut, administrator dapat melakukan aktivitas mengunduh atau mendownload *file* soal ujian. Pada proses ini administrator diharuskan mengisi inputan kunci PIN yang sebelumnya telah dikirim oleh *user* pada proses *request*. Pada proses ini juga sistem dengan otomatis melakukan proses dekripsi algoritma kriptografi ElGamal dengan membuka *private key* menggunakan kunci PIN yang dikirimkan oleh *user*.

d. Tampilan Halaman *Form* Tambah *User*

Tampilan halaman *form* tambah *user* berfungsi untuk administrator melakukan proses penambahan *user* baru kedalam sistem sebagai guru. Adapun hasil tampilan halaman *form* edit profil adalah sebagai berikut:

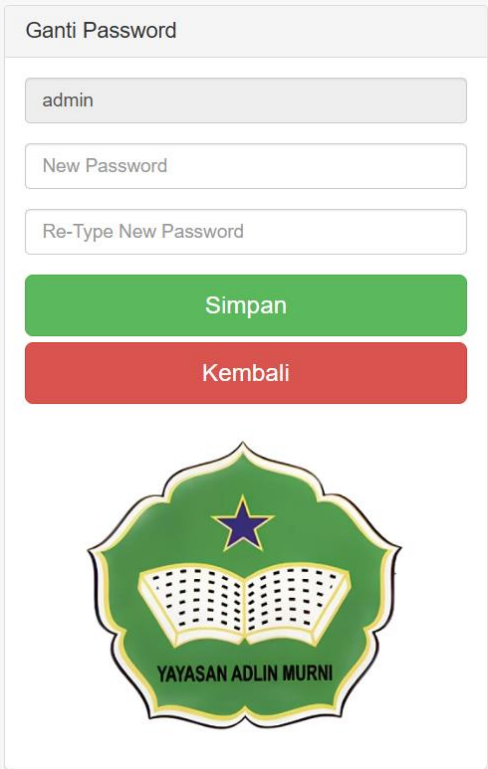


Gambar 4.12. Tampilan Halaman *Form* Tambah *User*

Pada *form* tambah *user* tersebut, administrator dapat melakukan aktivitas membuat data *user* baru yang nantinya digunakan *user* untuk menjalankan aktivitas sebagai *user* pada sistem. Pada proses ini administrator harus mengisi inputan form yang berisi NUPTK, *username*, nama lengkap, email, tanggal lahir, jenis kelamin, guru kelas dan guru mata pelajaran. Sistem akan mengisi *password default* untuk user berupa karakter 123 dan menyimpan seluruh data inputan tersebut kedalam *tb_login* pada *database*.

e. Tampilan Halaman *Form* Ganti *Password* Pada Administrator

Tampilan halaman *form* ganti *password* pada administrator berfungsi untuk administrator melakukan proses mengubah atau mengganti *password* saat melakukan *login*. Adapun hasil tampilan halaman *form* ganti *password* pada administrator adalah sebagai berikut:



Gambar 4.13. Tampilan Halaman *Form* Ganti *Password* Pada Administrator

Pada *form* ganti *password* tersebut, Administrator dapat melakukan pergantian *password* pada aktivitas *login* untuk alasan keamanan. Administrator akan diminta untuk memasukkan password baru dan mengulangi pengisian password baru, kemudian sistem akan melakukan *update* pada tabel *tb_login* di dalam *database*.

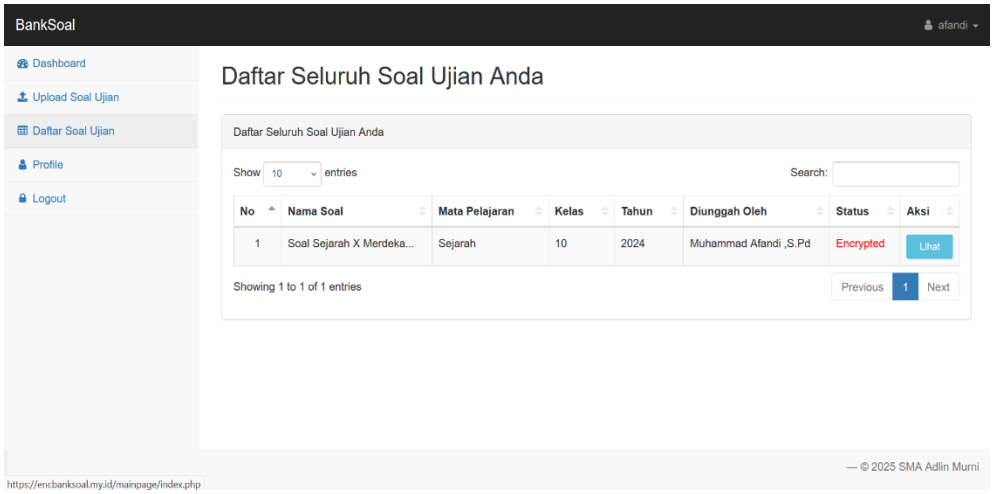
4.1.2. Tampilan Halaman *List* Data

Tampilan halaman *list* data adalah halaman sistem yang menampilkan *list* atau daftar data yang terdapat pada *database* untuk keperluan sistem yang berjalan. Adapun tampilan halaman *list* data pada sistem ini adalah sebagai berikut:

1. Tampilan Halaman *List* Data Pada *User*

   a. Tampilan Halaman *List* Data Daftar Soal Ujian Pada *User*

   Tampilan halaman *list* data daftar soal ujian pada *user* berfungsi untuk menampilkan seluruh *list* atau daftar data soal ujian yang diunggah oleh *user*. Adapun hasil tampilan halaman *list* data daftar soal ujian pada *user* adalah sebagai berikut:



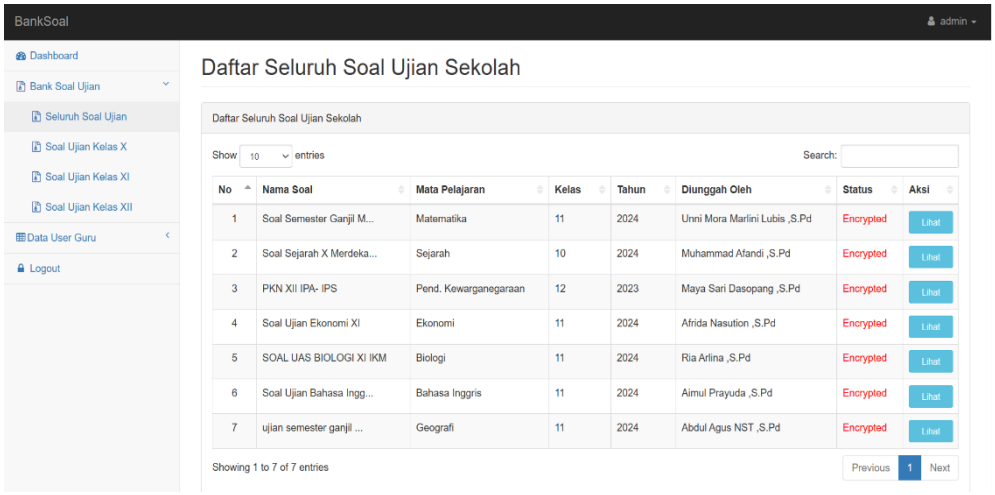
Gambar 4. 14. Tampilan Halaman *List* Data Daftar Soal Ujian Pada *User*

Pada *list* data daftar soal ujian tersebut, menampilkan daftar data soal ujian yang hanya diunggah oleh *user* tersebut. Pada list data ini tertera nama soal, mata pelajaran, kelas, tahun, diunggah oleh, status serta aksi untuk dapat melihat keseluruhan data soal ujian.

2. Tampilan Halaman List Data Pada Administrator

   a. Tampilan Halaman *List* Data Daftar Soal Ujian Pada Administrator

   Tampilan halaman *list* data daftar soal ujian pada administrator berfungsi untuk menampilkan seluruh *list* atau daftar data soal ujian yang diunggah oleh semua *user*. Adapun hasil tampilan halaman *list* data daftar soal ujian pada administrator adalah sebagai berikut:

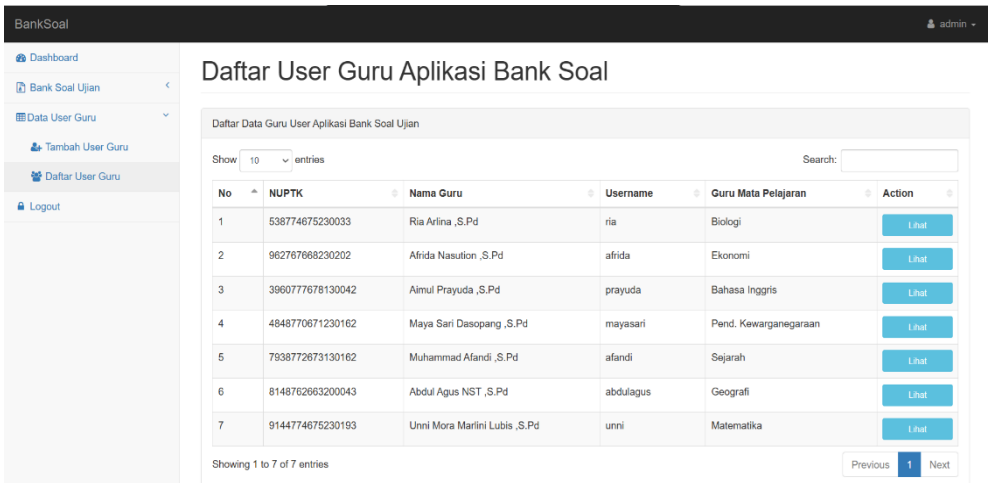


Gambar 4.15. Tampilan Halaman *List* Data Daftar Soal Ujian Pada Administrator

Pada *list* data daftar soal ujian tersebut, menampilkan daftar seluruh data soal ujian yang diunggah oleh semua *user*. Pada list data ini tertera nama soal, mata pelajaran, kelas, tahun, diunggah oleh, status serta aksi untuk dapat melihat keseluruhan data soal ujian.

b. Tampilan Halaman *List* Data Daftar *User*

Tampilan halaman *list* data daftar *user* berfungsi untuk menampilkan seluruh *list* atau daftar *user* yang terdaftar oleh sistem. Adapun hasil tampilan halaman *list* data daftar *user* adalah sebagai berikut:



Gambar 4.16. Tampilan Halaman *List* Data Daftar *User*

Pada *list* data daftar *user* tersebut, menampilkan daftar seluruh data *user* yang telah didaftarkan sebelumnya oleh administrator. Pada list data ini tertera NUPTK, nama guru, *username*, guru mata pelajaran serta *action* untuk dapat melihat keseluruhan data *user*.

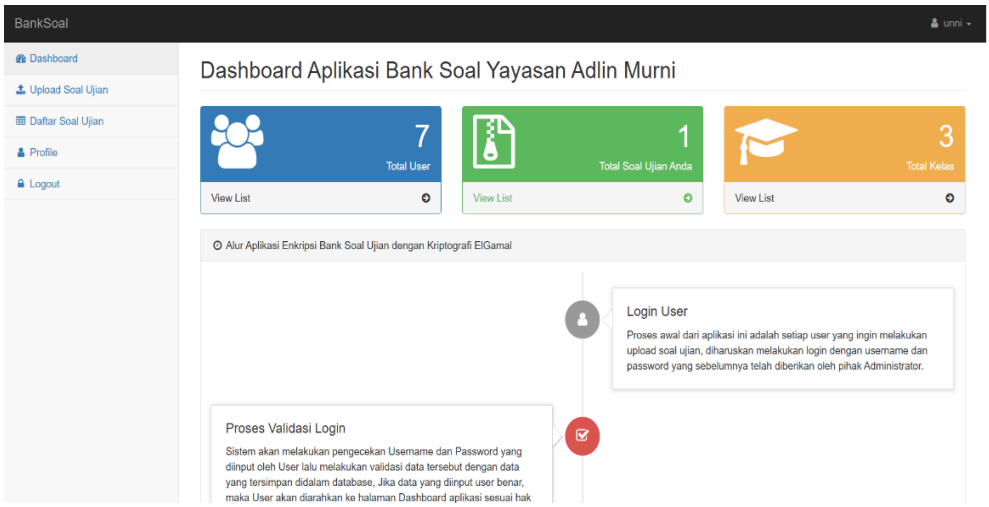
4.1.3. Tampilan Halaman Informasi

Tampilan halaman informasi adalah halaman sistem yang menampilkan tampilan antarmuka sebagai penjelas dan menampilkan detail data pada setiap data yang ditampilkan pada halaman *list*. Adapun tampilan halaman informasi pada sistem ini adalah sebagai berikut:

1. Tampilan Halaman Informasi Pada *User*

   a. Tampilan Halaman Informasi *Dashboard* Pada *User*

   Tampilan halaman informasi *dashboard* pada *user* berfungsi untuk menampilkan halaman awal antarmuka dan penjelasan singkat alur sistem. Adapun hasil tampilan halaman informasi *dashboard* pada *user* adalah sebagai berikut:

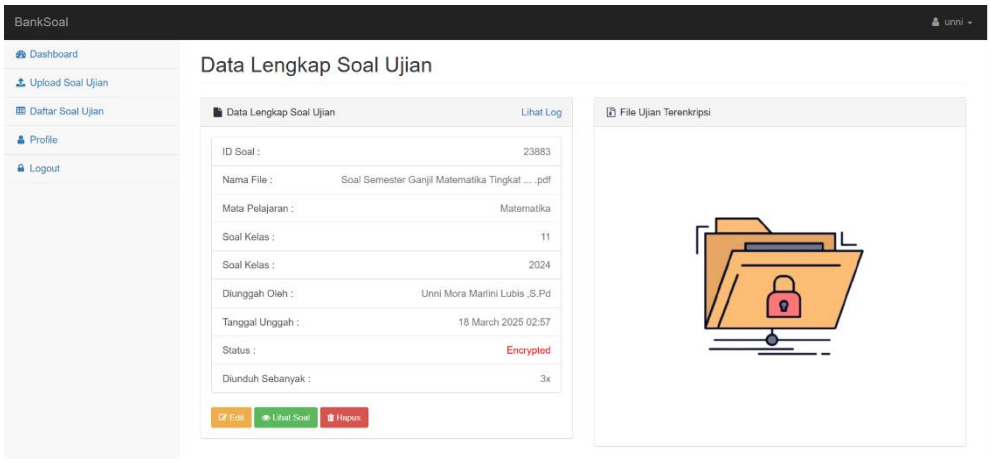


Gambar 4.17. Tampilan Halaman Informasi *Dashboard* Pada *User*

Pada halaman *dashboard* tersebut, menampilkan halaman awal antarmuka dan penjelasan singkat alur sistem seperti total *user*, total soal ujian dan total kelas. Halaman *dashboard* ini adalah halaman utama yang diakses *user* ketika melakukan aktivitas *login*.

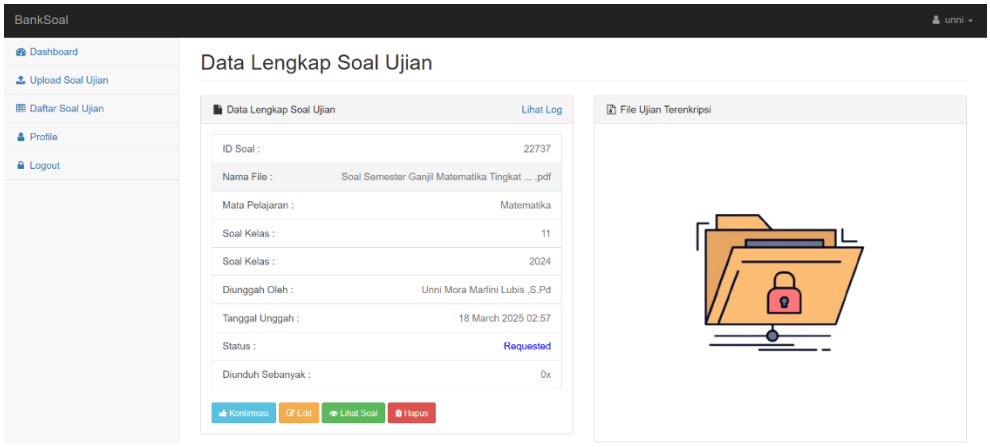
b. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada *User*

Tampilan halaman informasi lihat data soal ujian pada *user* berfungsi untuk menampilkan halaman informasi detail data soal ujian. Terdapat 3 antarmuka pada tampilan halaman lihat data soal ujian pada *user* sesuai dengan status dari data soal ujian, antara lain:



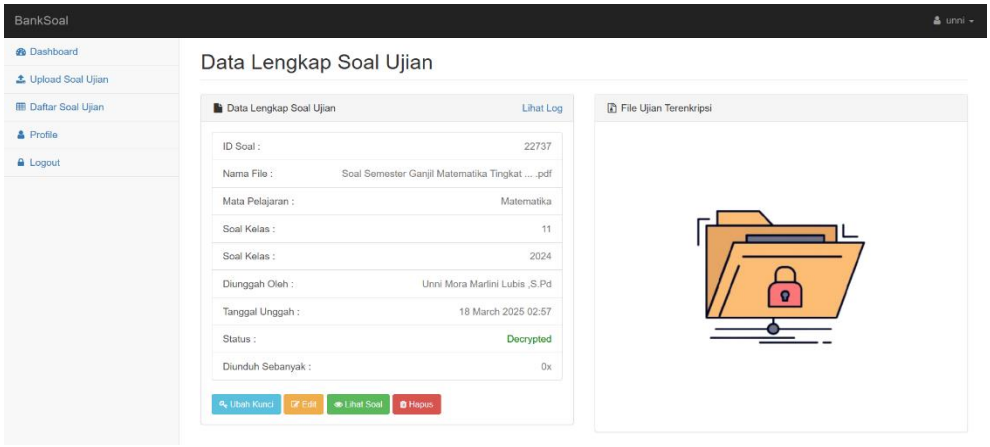
Gambar 4. 18. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada *User* (Status: *Encrypted*)

Pada halaman lihat data soal ujian tersebut, menampilkan halaman data soal ujian secara lengkap yang ditampilkan kepada *user* dengan status soal ujian sebagai *encrypted*. Status *encrypted* ini berarti administrator harus meminta konfirmasi izin kepada *user* sebelum melakukan aktivitas *download file* soal ujian. Pada halaman ini terdapat *button* edit berfungsi mengedit data soal ujian, *button* lihat soal berfungsi melihat data soal ujian dan *button* hapus berfungsi menghapus data soal ujian.



Gambar 4.19. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada *User* (Status: *Reqeusted*)

Pada halaman lihat data soal ujian tersebut, menampilkan halaman data soal ujian secara lengkap yang ditampilkan kepada *user* dengan status soal ujian sebagai *requested*. Status *requested* ini berarti administrator telah meminta izin kepada *user* sebelum melakukan aktivitas *download file* soal ujian. Pada halaman ini terdapat *button* konfirmasi berfungsi untuk mengkonfirmasi permintaan administrator, *button* edit berfungsi mengedit data soal ujian, *button* lihat soal berfungsi melihat data soal ujian dan *button* hapus berfungsi menghapus data soal ujian.

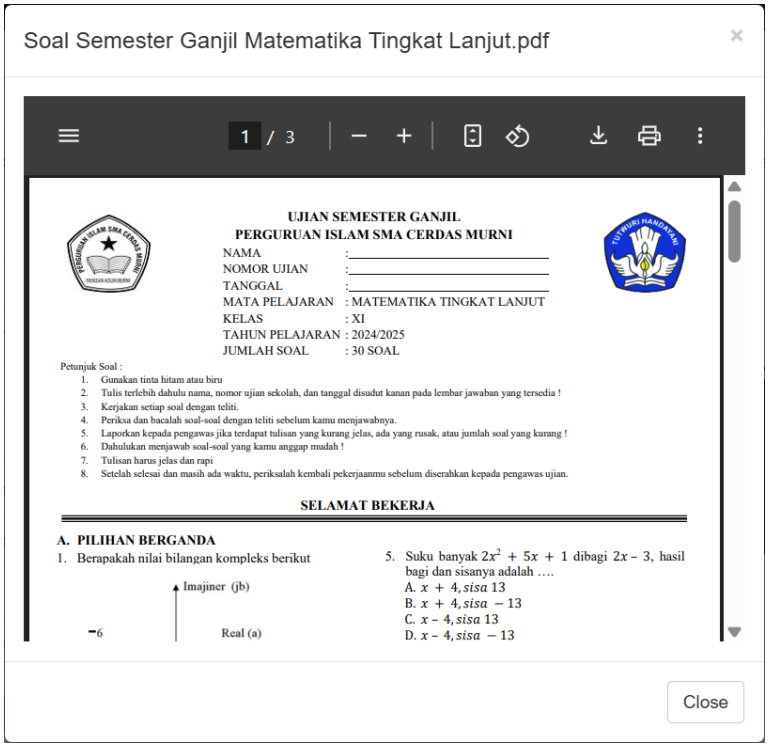


Gambar 4.20. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada *User* (Status: *Decrypted*)

Pada halaman lihat data soal ujian tersebut, menampilkan halaman data soal ujian secara lengkap yang ditampilkan kepada *user* dengan status soal ujian sebagai *decrypted*. Status *decrypted* ini berarti administrator sudah diizinkan oleh *user* untuk melakukan aktivitas *download file* soal ujian. Pada halaman ini terdapat *button* ubah kunci berfungsi untuk mengubah kunci PIN soal ujian, *button* edit berfungsi mengedit data soal ujian, *button* lihat soal berfungsi melihat data soal ujian dan *button* hapus berfungsi menghapus data soal ujian.

c. Tampilan Halaman Informasi Lihat *File* Soal Ujian

Tampilan halaman informasi lihat *file* soal ujian berfungsi untuk menampilkan *modal* antarmuka *file* soal ujian berekstensi PDF yang telah diunggah sebelumnya. Adapun hasil tampilan halaman informasi lihat *file* soal ujian adalah sebagai berikut:

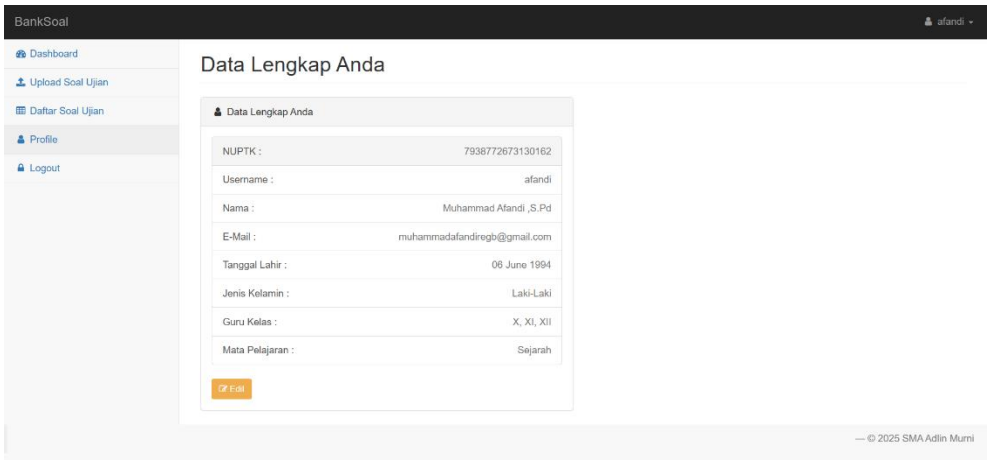


Gambar 4.21. Tampilan Halaman Informasi Lihat *File* Soal Ujian

Pada halaman lihat *file* soal ujian tersebut, menampilkan modal isi *file* soal ujian. Pada proses ini file soal ujian dapat diunduh dan dicetak oleh user.

d. Tampilan Halaman Informasi Profil

Tampilan halaman informasi profil berfungsi untuk menampilkan halaman profil lengkap *user*. Adapun hasil tampilan halaman informasi profil adalah sebagai berikut:



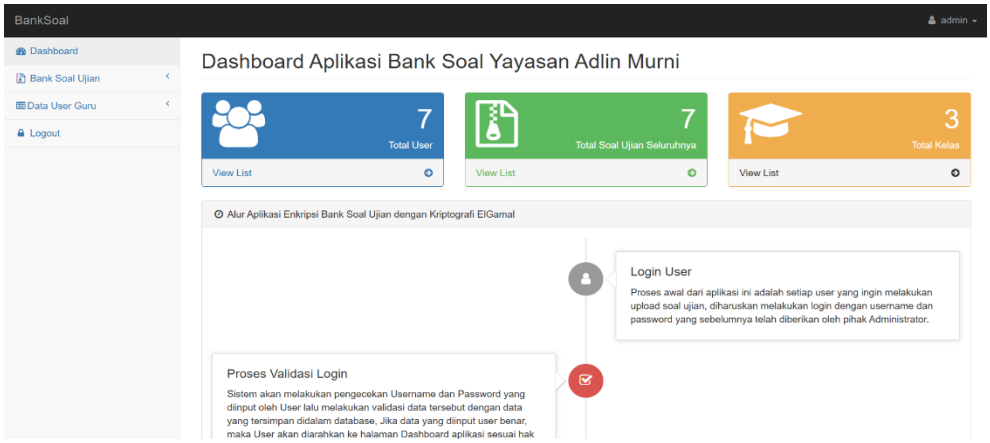
Gambar 4.22. Tampilan Halaman Informasi Profil

Pada halaman profil tersebut, menampilkan halaman profil *user* secara lengkap. Pada halaman ini *user* dapat melakukan pengubahan data dengan menekan *button* edit dan masuk kehalaman *form* edit profil.

2. Tampilan Halaman Informasi Pada Administrator

a. Tampilan Halaman Informasi *Dashboard* Pada Administrator

Tampilan halaman informasi *dashboard* pada administrator berfungsi untuk menampilkan halaman awal antarmuka dan penjelasan singkat alur sistem. Adapun hasil tampilan halaman informasi *dashboard* pada administrator adalah sebagai berikut:

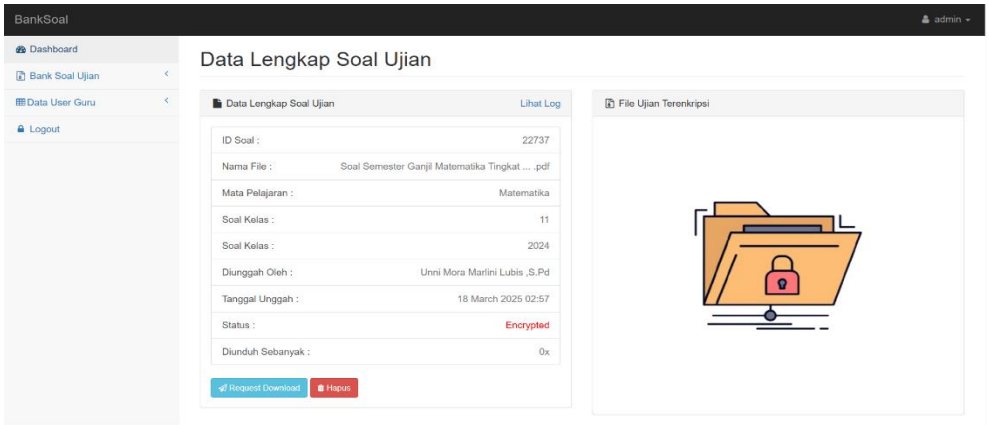


Gambar 4.23. Tampilan Halaman Informasi *Dashboard* Pada Administrator

Pada halaman *dashboard* tersebut, menampilkan halaman awal antarmuka dan penjelasan singkat alur sistem seperti total *user*, total soal ujian dan total kelas. Halaman *dashboard* ini adalah halaman utama yang diakses administrator ketika melakukan aktivitas *login*.

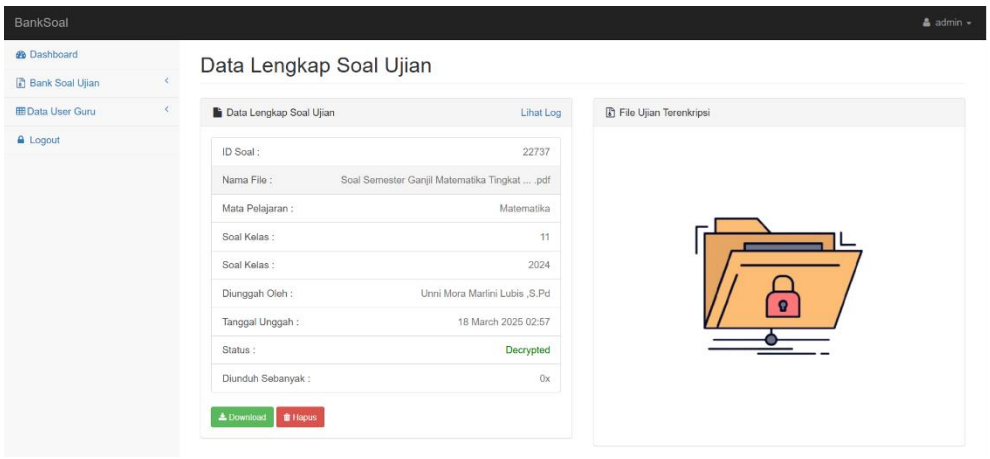
b. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada Administrator

Tampilan halaman informasi lihat data soal ujian pada administrator berfungsi untuk menampilkan halaman informasi detail data soal ujian. Terdapat 2 antarmuka pada tampilan halaman lihat data soal ujian pada administrator sesuai dengan status dari data soal ujian, antara lain:



Gambar 4.24. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada Administrator (Status: *Encrypted*)

Pada halaman lihat data soal ujian tersebut, menampilkan halaman data soal ujian secara lengkap yang ditampilkan kepada administrator dengan status soal ujian sebagai *encrypted*. Status *encrypted* ini berarti administrator harus meminta konfirmasi izin kepada *user* sebelum melakukan aktivitas *download file* soal ujian. Pada halaman ini terdapat *button request download* berfungsi untuk meminta *request download* soal ujian kepada *user* dengan mengirim email, dan *button* hapus berfungsi menghapus data soal ujian.

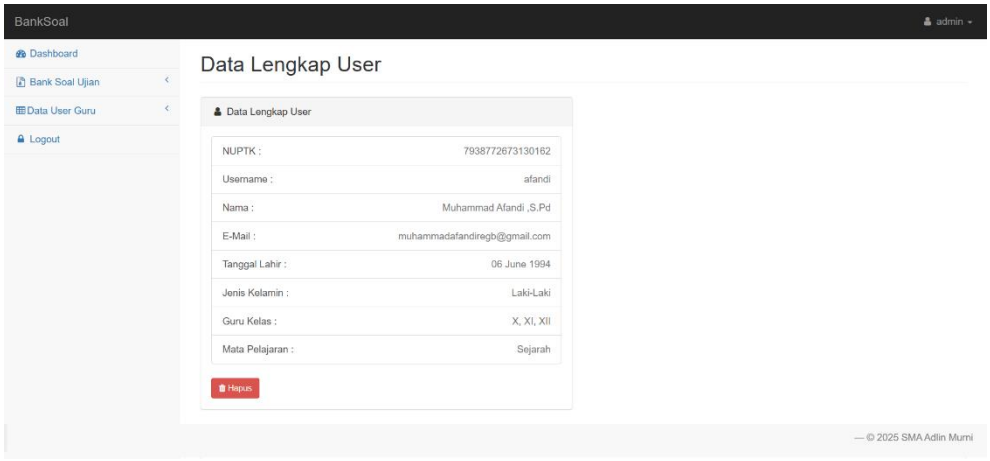


Gambar 4.25. Tampilan Halaman Informasi Lihat Data Soal Ujian Pada Administrator (Status: *Decrypted*)

Pada halaman lihat data soal ujian tersebut, menampilkan halaman data soal ujian secara lengkap yang ditampilkan kepada *user* dengan status soal ujian sebagai *decrypted*. Status *decrypted* ini berarti administrator sudah diizinkan oleh *user* untuk melakukan aktivitas *download file* soal ujian. Pada halaman ini terdapat *button download* berfungsi untuk mengunduh file soal ujian dengan memasukkan kunci PIN yang telah dikirim oleh *user* dan *button* hapus berfungsi menghapus data soal ujian.

c. Tampilan Halaman Informasi Data Profil Setiap *User*

Tampilan halaman informasi profil setiap *user* berfungsi untuk menampilkan halaman profil lengkap pada setiap *user*. Adapun hasil tampilan halaman informasi profil setiap *user* adalah sebagai berikut:



Gambar 4.26. Tampilan Halaman Informasi Data Profil Setiap *User*

Pada halaman profil tersebut, menampilkan halaman profil *user* secara lengkap. Pada halaman ini administator dapat melihat data *user* secara keseluruhan. Pada halaman ini terdapat *button* hapus yang berfungsi untuk menghapus data *user*.

4.2. Pembahasan dan Uji Coba

Tahap ini merupakan bagian yang berisi analisis hasil penelitian dan pengujian yang telah dilakukan. Pada bagian ini, data yang diperoleh diolah, dibandingkan dengan teori yang relevan, serta diinterpretasikan untuk menjawab rumusan masalah. Selain itu, bagian uji coba mencakup proses verifikasi atau validasi sistem, metode, atau model yang digunakan dalam penelitian.

4.2.1. Pembahasan

Pembahasan Perangkat yang dibutuhkan dalam penelitian ini mencakup analisis dan spesifikasi perangkat keras (*hardware*) serta perangkat lunak (*software*) yang digunakan dalam penelitian atau pengembangan sistem. Adapun perangkat yang digunakan dalam penelitian ini, antara lain:

1. Perangkat Keras (*Hardware*)

   Dalam penelitian ini, terdapat minimal spesifikasi laptop atau *personal computer* sebagai perangkat keras (*hardware*) yang dibutuhkan untuk menjalankan sistem dengan algoritma kriptografi ElGamal, antara lain:

   a. *Processor* minimal *intel core* i3 generasi 5.
   b. RAM minimal 4 GB.
   c. *Harddisk* atau SSD minimal 128 GB.
   d. Koneksi internet kabel maupun *wifi* dengan kecepatan minimal 10 Mbps

2. Perangkat Lunak (*Software*)

   Dalam penelitian ini, terdapat beberapa perangkat lunak yang dibutuhkan untuk menjalankan sistem dengan algoritma kriptografi ElGamal agar dapat dijalankan dengan baik, antara lain:

   a. *Operating System* minimal *Windows* 7 64-Bit.
   b. *Web Access* menggunakan *Google Chrome* atau *Mozilla Firefox* terbaru.
   c. PDF *Viewer* menggunakan *Adobe Reader* atau aplikasi pembaca PDF lain.

4.2.2. Uji Coba

Pada tahap uji coba akan dilakukan pengujian sistem apakah telah sesuai dengan perancangan dari target yang akan dicapai. Sistem yang telah dirancang

akan dilakukan uji coba menggunakan *blackbox testing* agar dapat melihat hasil uji sistem dari setiap *case* yang ada pada sistem.

1. Uji Coba Sistem Pada *User*

   a. Uji Coba *Form Login* Pada *User*

      Serangkaian uji coba *form login* pada *user* dapat dilihat pada tabel berikut ini:

Tabel 4.1. *Blackbox Testing Form Login* Pada *User*

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses https://encbanksoal.my.id/	Halaman awal *web browser*	Sistem akan menampilkan halaman *login*	Sesuai harapan
2	*User* melakukan *login* dengan *username* dan *password* yang telah terdaftar	Halaman *login user*	Sistem akan masuk ke halaman *dashboard user*	Sesuai harapan

   b. Uji Coba *Form Upload* Soal Ujian

      Serangkaian uji coba *form upload* soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.2. *Blackbox Testing Form Upload* Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses menu *upload* soal ujian	*Sidebar* menu sistem	Sistem akan menampilkan halaman *upload* soal ujian	Sesuai harapan
2	*User* mengisi data soal ujian dan memilih *file* soal ujian, lalu *user* klik button unggah	Halaman *upload* soal ujian	Sistem akan membangkitkan *public key* dan *private key*, lalu mengenkripsi soal ujian serta	Sesuai harapan

			menyimpan pada tb_banksoal	

c. Uji Coba *Form* Lihat *File* Soal Ujian

Serangkaian uji coba *form* lihat *file* soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.3. *Blackbox Testing Form* Lihat *File* Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses menu daftar soal ujian	*Sidebar* menu sistem	Sistem akan menampilkan halaman *list* data soal ujian	Sesuai harapan
2	*User* klik *button* lihat	Halaman *list* data soal ujian	Sistem menuju halaman lihat data soal ujian yang diinginkan *user*	Sesuai harapan
3	*User* klik *button* lihat soal	Halaman lihat data soal ujian	Sistem menampilkan *modal* lihat soal ujian	Sesuai harapan
4	*User* memasukkan kunci PIN lalu klik *button* buka	*Modal* lihat soal ujian	Sistem mendekripsi *file* soal ujian dan menampilkan file PDF pada *modal*	Sesuai harapan

d. Uji Coba *Form* Edit Data Soal Ujian

Serangkaian uji coba *form* edit data soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.4. *Blackbox Testing Form* Edit Data Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses menu daftar soal ujian	*Sidebar* menu sistem	Sistem akan menampilkan halaman *list* data soal ujian	Sesuai harapan

2	*User* klik *button* lihat	Halaman *list* data soal ujian	Sistem menuju halaman lihat data soal ujian yang diinginkan *user*	Sesuai harapan
3	*User* klik *button* edit	Halaman lihat data soal ujian	Sistem menuju halaman edit data soal ujian	Sesuai harapan
4	*User* mengisi perubahan data soal ujian, lalu klik *button* simpan	Halaman edit data soal ujian	Sistem menyimpan perubahan data soal ujian sekolah	Sesuai harapan
5	*User* klik *button* kembali	Halaman edit data soal ujian	Sistem menuju halaman *list* data soal ujian	Sesuai harapan

e. Uji Coba *Form* Konfirmasi *Download* Soal Ujian

Serangkaian uji coba *form* konfirmasi *download* soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.5. *Blackbox Testing Form* Konfirmasi *Download* Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses email yang terdaftar pada sistem	Email *user*	Pesan email *request download* soal ujian dari administrator diterima oleh *user*	Sesuai harapan
2	*User* klik *button* konfirmasi permintaan	Email *user*	Sistem menuju halaman *login*	Sesuai harapan
3	*User* melakukan *login*	Halaman *login* sistem	Sistem menuju halaman lihat data soal ujian yang diminta *request download* administrator	Sesuai harapan
4	*User* klik *button* konfirmasi	Halaman lihat data soal ujian	Sistem menampilkan *modal* lihat soal ujian	Sesuai harapan
5	*User* memasukkan kunci PIN lalu klik *button* buka	*Modal* lihat soal ujian	Sistem akan mengupdate status data soal ujian dari	Sesuai harapan

			requested menjadi *decrypted* lalu mengirimkan kunci PIN ke email administrator	

f. Uji Coba *Form* Ubah Kunci PIN Soal Ujian

Serangkaian uji coba *form* ubah kunci PIN soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.6. *Blackbox Testing Form* Ubah Kunci PIN Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* klik *button* ubah kunci	Halaman lihat data soal ujian	Sistem menampilkan *modal form* ubah kunci data soal ujian	Sesuai harapan
2	*User* memasukkan kunci PIN lama dan memasukkan kunci PIN baru lalu klik *button update*	*Modal* lihat soal ujian	Sistem akan mengubah kunci PIN lama dengan yang terbaru lalu mengupdate status data soal ujian dari *decrypted* menjadi *encrypted*	Sesuai harapan

g. Uji Coba *Form* Edit Profil

Serangkaian uji coba *form* edit profil dapat dilihat pada tabel berikut:

Tabel 4.7. *Blackbox Testing Form* Edit Profil

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses menu profil	*Sidebar* menu sistem	Sistem akan menampilkan halaman profil	Sesuai harapan
2	*User* klik *button* edit	Halaman profil *user*	Sistem menuju halaman edit data profil *user*	Sesuai harapan

3	*User* mengisi *form* edit data user lalu klik *button* simpan	Halaman edit profil	Sistem mengupdate data user pada tb_dataguru	Sesuai harapan
4	*User* klik *button* kembali	Halaman edit profil	Sistem menuju halaman profil *user*	Sesuai harapan

h. Uji Coba Ganti *Password* Pada *User*

Serangkaian uji coba ganti *password* pada *user* dapat dilihat pada tabel berikut ini:

Tabel 4.8. *Blackbox Testing* Ganti *Password* Pada *User*

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses menu *header*	*header* menu sistem	Sistem akan menampilkan *dropdown* menu	Sesuai harapan
2	*User* klik menu ganti *password*	*Dropdown* menu	Sistem menuju halaman ganti *password user*	Sesuai harapan
3	*User* mengisi *form* ganti *password* dan klik *button* simpan	Halaman *form* ganti *password*	Sistem mengupdate *password* data *user* pada tb_login	Sesuai harapan
4	*User* klik *button* kembali	Halaman *form* ganti *password*	Sistem menuju halaman *dashboard user*	Sesuai harapan

i. Uji Coba *Logout* Pada *User*

Serangkaian uji coba *logout* pada *user* dapat dilihat pada tabel berikut ini:

Tabel 4.9. *Blackbox Testing Logout* Pada *User*

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	*User* mengakses menu *header*	*header* menu sistem	Sistem akan menampilkan *dropdown* menu	Sesuai harapan

2	*User* klik menu *logout*	*Dropdown* menu	Sistem menghapus *session* dan menuju halaman *login user*	Sesuai harapan

2. Uji Coba Sistem Pada Administrator

a. Uji Coba *Form Login* Pada Administrator

Serangkaian uji coba *form login* pada administrator dapat dilihat pada tabel berikut ini:

Tabel 4.10. *Blackbox Testing Form Login* Pada Administrator

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	Administrator mengakses https://encbanksoal.my.id/	Halaman awal *web browser*	Sistem akan menampilkan halaman *login*	Sesuai harapan
2	Administrator melakukan *login* dengan *username* dan *password*	Halaman *login* Administrator	Sistem akan masuk ke halaman *dashboard* Administrator	Sesuai harapan

b. Uji Coba *Form Request Download* Soal Ujian

Serangkaian uji coba *form request download* soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.11. *Blackbox Testing Form Request Download* Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	Administrator mengakses menu daftar soal ujian	*Sidebar* menu sistem	Sistem akan menampilkan halaman *list* data soal ujian	Sesuai harapan
2	Administrator klik *button* lihat	Halaman *list* data soal ujian	Sistem menuju halaman lihat data soal ujian yang diinginkan	Sesuai harapan

3	Administrator klik *button request download*	Halaman lihat data soal ujian	Sistem menampilkan *modal* lihat soal ujian	Sesuai harapan
4	Administrator mengisi pesan kepada *user* lalu klik *button* kirim	*Modal request download*	Sistem akan mengirim pesan kepada *user* melalui email dan mengupdate status data soal ujian dari *encrypted* menjadi *requested*	Sesuai harapan

c. Uji Coba *Form Download* Soal Ujian

Serangkaian uji coba *form download* soal ujian dapat dilihat pada tabel berikut ini:

Tabel 4.12. *Blackbox Testing Form Download* Soal Ujian

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	Administrator mengakses email yang terdaftar pada sistem	Email Administrator	Pesan email berisi kunci PIN soal ujian dari *user* diterima oleh Administrator	Sesuai harapan
2	*User* klik *button* lanjutkan *download*	Email Administrator	Sistem menuju halaman *login*	Sesuai harapan
3	Administrator melakukan *login*	Halaman *login* sistem	Sistem menuju halaman lihat data soal ujian yang akan didownload oleh administrator	Sesuai harapan
4	*User* klik *button download*	Halaman lihat data soal ujian	Sistem menampilkan *modal download* soal ujian	Sesuai harapan
5	Administrator memasukkan kunci PIN lalu klik *button* download	*Modal* lihat soal ujian	Sistem akan membuka *private key* dari kunci PIN lalu mendekripsi *file* data soal ujian serta menambah *row* pada	Sesuai harapan

			tb_history sebagai catatan *download*	

d. Uji Coba *Form* Tambah *User*

Serangkaian uji coba *form* tambah *user* dapat dilihat pada tabel berikut ini:

Tabel 4.13. *Blackbox Testing Form* Tambah *User*

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	Administrator mengakses menu daftar tambah *user* guru	*Sidebar* menu sistem	Sistem akan menampilkan halaman *form* tambah *user* guru	Sesuai harapan
2	Administrator mengisi *form* tambah *user* baru lalu klik *button* simpan	*Form* tambah *user*	Sistem menyimpan data *user* baru pada tb_dataguru dan tb_login	Sesuai harapan

e. Uji Coba *Form* Ganti *Password* Pada Administrator

Serangkaian uji coba *form* ganti *password* pada administrator dapat dilihat pada tabel berikut ini:

Tabel 4.14. *Blackbox Testing Form* Ganti *Password* Pada Administrator

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	Administrator mengakses menu *header*	*header* menu sistem	Sistem akan menampilkan *dropdown* menu	Sesuai harapan
2	Administrator klik menu ganti *password*	*Dropdown* menu	Sistem menuju halaman ganti *password*	Sesuai harapan
3	Administrator mengisi *form* ganti *password* dan klik *button* simpan	Halaman *form* ganti *password*	Sistem mengupdate *password* data Administrator pada tb_login	Sesuai harapan

4	Administrator klik *button* kembali	Halaman *form* ganti *password*	Sistem menuju halaman *dashboard* Administrator	Sesuai harapan

f. Uji Coba *Logout* Pada Administrator

Serangkaian uji coba *logout* pada administrator dapat dilihat pada tabel berikut ini:

Tabel 4.15. Blackbox Testing Uji Coba *Logout* Pada Administrator

No	Skenario Pengujian	Tes *Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	Administrator mengakses menu *header*	*header* menu sistem	Sistem akan menampilkan *dropdown* menu	Sesuai harapan
2	Administrator klik menu *logout*	*Dropdown* menu	Sistem menghapus *session* dan menuju halaman *login*	Sesuai harapan

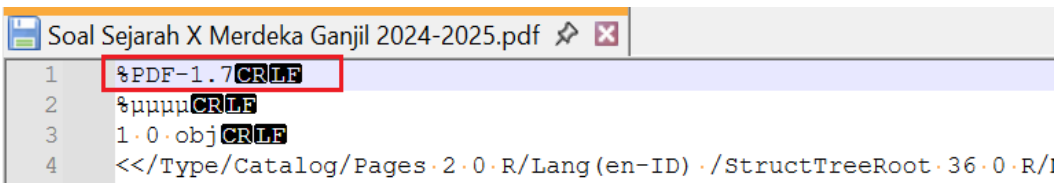
3. Uji Coba Perhitungan Metode Algoritma ElGamal Pada Sistem

Pada tahap ini membahas pengujian proses pembangkitan kunci, enkripsi dan dekripsi menggunakan algoritma kriptografi ElGamal pada sistem yang dikembangkan. Pengujian dilakukan untuk memastikan bahwa algoritma kriptografi ElGamal telah bekerja dengan benar dalam mengamankan data.

Pada pengujian sistem ini, penulis menggunakan 1 *file* ujian sekolah dengan data percobaan sebagai berikut:

Nama *File Plaintext*: Soal Sejarah X Merdeka Ganjil 2024-2025.pdf

Jumlah Sampel *Plaintext:* Baris pertama pada struktur *file plaintext* (10 karakter).



Gambar 4.27. Sampel 10 Karakter Pertama *File Plaintext* Yang Akan Dilakukan Proses Perhitungan Enkripsi

Pada gambar diatas, terlihat bahwa baris pertama sebanyak 10 karakter awal berisi karakter %PDF-1.7CRLF yang akan dijadikan sampel untuk dilakukan proses perhitungan enkripsi dan dekripsi algoritma kriptografi ElGamal.

a. Uji Coba Perhitungan Proses Pembangkitan Kunci ElGamal

Perhitungan proses pembangkitan kunci pada algoritma kriptografi ElGamal akan melibatkan 2 kunci, yaitu *public key* dan *private key*. Adapun tabel perhitungan untuk membangkitkan kunci adalah sebagai berikut:

Tabel 4.16. Perhitungan Proses Pembangkitan Kunci ElGamal

No	Kunci	Cara Pembangkitan	Nilai	Jenis Kunci
1	*p*	Statis / Tidak berubah	2503	*Public Key*
2	*g*	Statis / Tidak berubah	2	*Public Key*
3	*y*	$y = g^x \ mod \ p$ $y = 2^{582} \ mod \ 2503$ $y = 829$	829	*Public Key*
4	*x*	Dinamis / Berubah tiap data soal ujian	582	*Private Key*

b. Uji Coba Perhitungan Proses Enkripsi Pada ElGamal

Perhitungan proses enkripsi pada algoritma kriptografi ElGamal akan melibatkan *public key* dan kunci acak yakni *random k*. Adapun tabel perhitungan untuk proses enkripsi adalah sebagai berikut:

Tabel 4.17. Perhitungan Proses Enkripsi Pada ElGamal

No	Karakter	Nilai ASCII	Acak *k*	Proses Perhitungan Enkripsi Elgamal	*Ciphertext*
1	%	37	463	$a = g^k \ mod \ p$ $a = 2^{463} \ mod \ 2503$ $a = 631$	*a1* = 631

				$b = y^k m \ mod \ p$ $b = \ 829^{463} \ mod \ 2503$ $b = 1230 * 37 \ mod \ 2503$ $b = \ 456$	*b1* = 456
2	P	80	881	$a = g^k \ mod \ p$ $a = \ 2^{881} \ mod \ 2503$ $a = 358$	*a2* = 358
				$b = y^k m \ mod \ p$ $b = \ 829^{881} \ mod \ 2503$ $b = 949 * 80 \ mod \ 2503$ $b = \ 830$	*b2* = 830
3	D	68	947	$a = g^k \ mod \ p$ $a = \ 2^{947} \ mod \ 2503$ $a = 973$	*a3* = 973
				$b = y^k m \ mod \ p$ $b = \ 829^{947} mod \ 2503$ $b = 1171 * 68 \ mod \ 2503$ $b = \ 2035$	*b3* = 2035
4	F	70	415	$a = g^k \ mod \ p$ $a = \ 2^{415} \ mod \ 2503$ $a = 1558$	*a4* = 1558
				$b = y^k m \ mod \ p$ $b = \ 829^{415} \ mod \ 2503$ $b = 792 * 70 \ mod \ 2503$ $b = \ 374$	*b4* = 374
5	-	45	1829	$a = g^k \ mod \ p$ $a = \ 2^{1829} \ mod \ 2503$ $a = 834$	*a5* = 834
				$b = y^k m \ mod \ p$ $b = \ 829^{1829} \ mod \ 2503$ $b = 1117 * 45 \ mod \ 2503$ $b = \ 205$	*b5* = 205
6	1	49	293	$a = g^k \ mod \ p$ $a = \ 2^{293} \ mod \ 2503$ $a = 1402$	*a6* = 1402
				$b = y^k m \ mod \ p$ $b = \ 829^{293} \ mod \ 2503$ $b = 412 * 49 \ mod \ 2503$ $b = \ 164$	*b6* = 164
7	.	46	1175	$a = g^k \ mod \ p$ $a = \ 2^{1175} \ mod \ 2503$	*a7* = 129

				$a = 129$	
				$b = y^k m \ mod \ p$ $b = \ 829^{1175} \ mod \ 2503$ $b = 564 * 46 \ mod \ 2503$ $b = \ 914$	*b7* = 914
8	7	55	1573	$a = g^k \ mod \ p$ $a = \ 2^{1573} \ mod \ 2503$ $a = 2083$	*a8* = 2083
				$b = y^k m \ mod \ p$ $b = \ 829^{1573} \ mod \ 2503$ $b = 1195 * 55 \ mod \ 2503$ $b = \ 647$	*b8* = 647
9	CR	13	259	$a = g^k \ mod \ p$ $a = \ 2^{259} \ mod \ 2503$ $a = 1915$	*a9* = 1915
				$b = y^k m \ mod \ p$ $b = \ 829^{259} \ mod \ 2503$ $b = 2429 * 13 \ mod \ 2503$ $b = \ 1541$	*b9* = 1541
10	LF	10	1841	$a = g^k \ mod \ p$ $a = \ 2^{1841} \ mod \ 2503$ $a = 1972$	*a10* = 1972
				$b = y^k m \ mod \ p$ $b = \ 829^{1841} \ mod \ 2503$ $b = 2096 * 10 \ mod \ 2503$ $b = \ 936$	*b10* = 936

c. Uji Coba Perhitungan Proses Dekripsi Pada ElGamal

Perhitungan proses dekripsi pada algoritma kriptografi ElGamal akan melibatkan *private key*. Adapun tabel perhitungan untuk proses dekripsi adalah sebagai berikut:

Tabel 4.18. Perhitungan Proses Dekripsi Pada ElGamal

No	*Ciphertext*	Proses Perhitungan dekripsi ElGamal	*Plaintext* / Nilai ASCII	Karakter
1	*a1* = 631	$(a^x)^{-1} = a^{p-1-x} \ mod \ p$ $(a^x)^{-1} = 631^{2503-1-582} \ mod \ 2503$	37	%

		$(a^x)^{-1} = 631^{1920}\ mod\ 2503$ $(a^x)^{-1} = 291$		
	b1 = 456	$m = b(a^x)^{-1}\ mod\ p$ $m = 456 * 291\ mod\ 2503$ $m = 37$		
2	*a2* = 358	$(a^x)^{-1} = a^{p-1-x}\ mod\ p$ $(a^x)^{-1} = 358^{2503-1-582}\ mod\ 2503$ $(a^x)^{-1} = 358^{1920}\ mod\ 2503$ $(a^x)^{-1} = 2292$	80	P
	b2 = 830	$m = b(a^x)^{-1}\ mod\ p$ $m = 830 * 2292\ mod\ 2503$ $m = 80$		
3	*a3* = 973	$(a^x)^{-1} = a^{p-1-x}\ mod\ p$ $(a^x)^{-1} = 973^{2503-1-582}\ mod\ 2503$ $(a^x)^{-1} = 973^{1920}\ mod\ 2503$ $(a^x)^{-1} = 171$	68	D
	b3 = 2035	$m = b(a^x)^{-1}\ mod\ p$ $m = 2035 * 171\ mod\ 2503$ $m = 68$		
4	*a4* = 1558	$(a^x)^{-1} = a^{p-1-x}\ mod\ p$ $(a^x)^{-1} = 1558^{2503-1-582}\ mod\ 2503$ $(a^x)^{-1} = 1558^{1920}\ mod\ 2503$ $(a^x)^{-1} = 1419$	70	F
	b4 = 374	$m = b(a^x)^{-1}\ mod\ p$ $m = 374 * 1419\ mod\ 2503$ $m = 70$		
5	*a5* = 834	$(a^x)^{-1} = a^{p-1-x}\ mod\ p$ $(a^x)^{-1} = 834^{2503-1-582}\ mod\ 2503$ $(a^x)^{-1} = 834^{1920}\ mod\ 2503$ $(a^x)^{-1} = 997$	45	-
	b5 = 205	$m = b(a^x)^{-1}\ mod\ p$ $m = 205 * 997\ mod\ 2503$ $m = 45$		
6	*a6* = 1402	$(a^x)^{-1} = a^{p-1-x}\ mod\ p$ $(a^x)^{-1} = 1402^{2503-1-582}\ mod\ 2503$ $(a^x)^{-1} = 1402^{1920}\ mod\ 2503$ $(a^x)^{-1} = 565$	49	1
	b6 = 164	$m = b(a^x)^{-1}\ mod\ p$ $m = 164 * 565\ mod\ 2503$ $m = 49$		

7	*a7* = 129	$(a^x)^{-1} = a^{p-1-x} \; mod \; p$ $(a^x)^{-1} = 129^{2503-1-582} \; mod \; 2503$ $(a^x)^{-1} = 129^{1920} \; mod \; 2503$ $(a^x)^{-1}$ =608	46	.
	b7 = 914	$m = b(a^x)^{-1} \; mod \; p$ $m = 914 * 608 \; mod \; 2503$ $m = 46$		
8	*a8* = 2083	$(a^x)^{-1} = a^{p-1-x} \; mod \; p$ $(a^x)^{-1} = 2083^{2503-1-582} \; mod \; 2503$ $(a^x)^{-1} = 2083^{1920} \; mod \; 2503$ $(a^x)^{-1} = 886$	55	7
	b8 = 647	$m = b(a^x)^{-1} \; mod \; p$ $m = 647 * 886 \; mod \; 2503$ $m = 55$		
9	*a9* = 1915	$(a^x)^{-1} = a^{p-1-x} \; mod \; p$ $(a^x)^{-1} = 1915^{2503-1-582} \; mod \; 2503$ $(a^x)^{-1} = 1915^{1920} \; mod \; 2503$ $(a^x)^{-1} = 575$	13	CR
	b9 = 1541	$m = b(a^x)^{-1} \; mod \; p$ $m = 1541 * 575 \; mod \; 2503$ $m = 13$		
10	*a10* = 1972	$(a^x)^{-1} = a^{p-1-x} \; mod \; p$ $(a^x)^{-1} = 1972^{2503-1-582} \; mod \; 2503$ $(a^x)^{-1} = 1972^{1920} \; mod \; 2503$ $(a^x)^{-1} = 2380$	10	LF
	b10 = 936	$m = b(a^x)^{-1} \; mod \; p$ $m = 936 * 2380 \; mod \; 2503$ $m = 10$		

4. Hasil Pengujian Sistem

Pada hasil pengujian sistem, dijelaskan hasil pengujian sistem yang telah dikembangkan dan diuji sebelumnya. Pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan kebutuhan dan dengan perhitungan yang telah dilakukan sebelumnya. Pada pengujian sistem ini, penulis menggunakan 1 *file* ujian sekolah dengan data percobaan sebagai berikut:

Nama *File Plaintext*: Soal Sejarah X Merdeka Ganjil 2024-2025.pdf

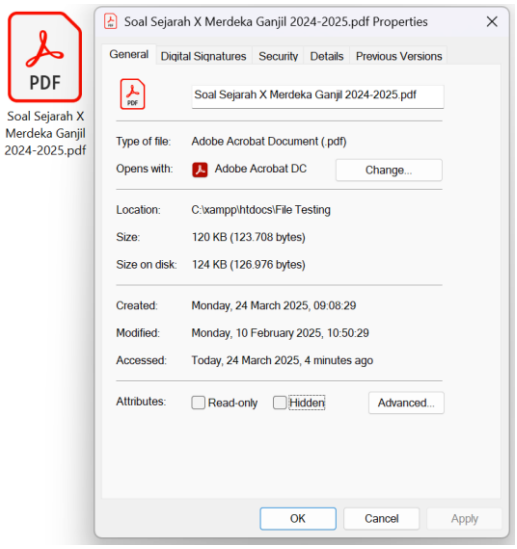
Ukuran *File Plaintext*: 120 *kilobytes* atau 123.708 *bytes*

a. Hasil Pengujian Sistem Pada *File Plaintext* Pada Soal Ujian

Hasil pengujian sistem pada *file plaintext* pada soal ujian akan menunjukkan struktur *file* PDF yang ditampilkan menggunakan aplikasi *notepad++* untuk membandingkan perbedaan struktur *file* sebelum proses enkripsi algoritma kriptografi ElGamal berlangsung, serta juga menunjukkan ukuran *file* pada soal ujian berekstensi PDF sebelum dienkripsi. Berikut adalah gambar struktur dan ukuran *file plaintext* pada soal ujian:

Gambar 4.28. Struktur *File Plaintext* Pada Soal Ujian berekstensi PDF

Pada gambar diatas menunjukkan struktur *file plaintext* pada soal ujian berekstensi PDF yang berisi sekumpulan karakter yang disusun sedemikian rupa, tiap-tiap karakter ini yang nantinya akan dilakukan perhitungan hingga seluruh karakter dienkripsi menggunakan algoritma kriptografi ElGamal.

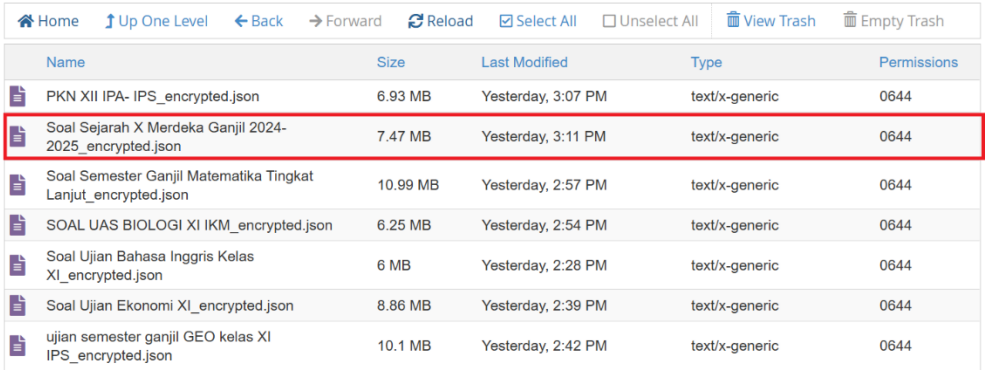


Gambar 4.29. Ukuran *File Plaintext* Pada Soal Ujian berekstensi PDF

Pada gambar diatas menunjukkan ukuran *file plaintext* pada soal ujian berekstensi PDF sebesar 120 kb atau 123.708 *bytes*.

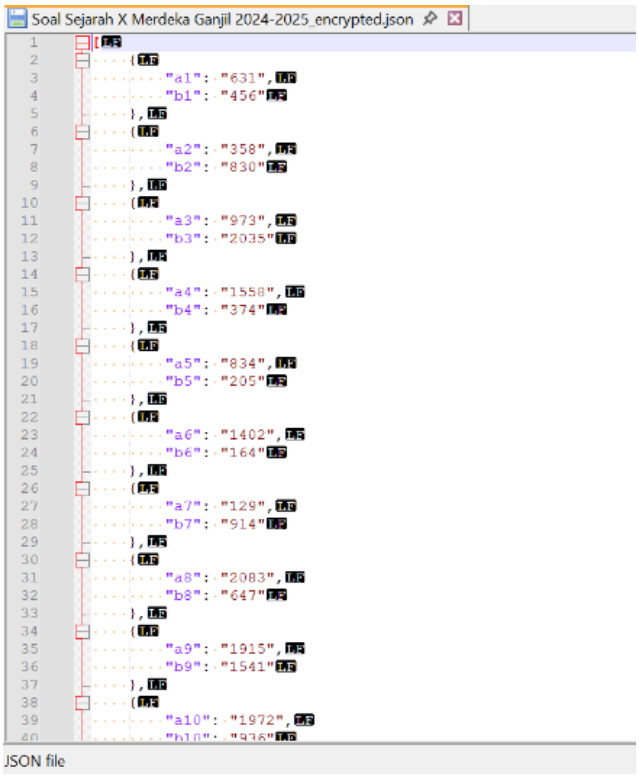
b. Hasil Pengujian Sistem Pada *File Ciphertext* Pada Soal Ujian

Hasil pengujian sistem pada *file* enkripsi atau *ciphertext* pada soal ujian akan menunjukkan struktur *file ciphertext* yang telah diubah dari ekstensi PDF menjadi ekstensi JSON, ini dikarenakan perbedaan struktur *file* telah diubah dengan urutan yang telah dikonsepkan sebelumnya. *File ciphertext* berekstensi JSON akan ditampilkan menggunakan aplikasi *notepad++* untuk membandingkan perbedaan struktur *file* setelah mengalami proses enkripsi algoritma kriptografi ElGamal, serta juga menunjukkan ukuran *file ciphertext* pada soal ujian berekstensi JSON setelah dienkripsi. Berikut adalah gambar struktur dan ukuran *file ciphertext* pada soal ujian:



Gambar 4.30. *File* Ciphertext Soal Ujian yang Terenkripsi Pada Sistem

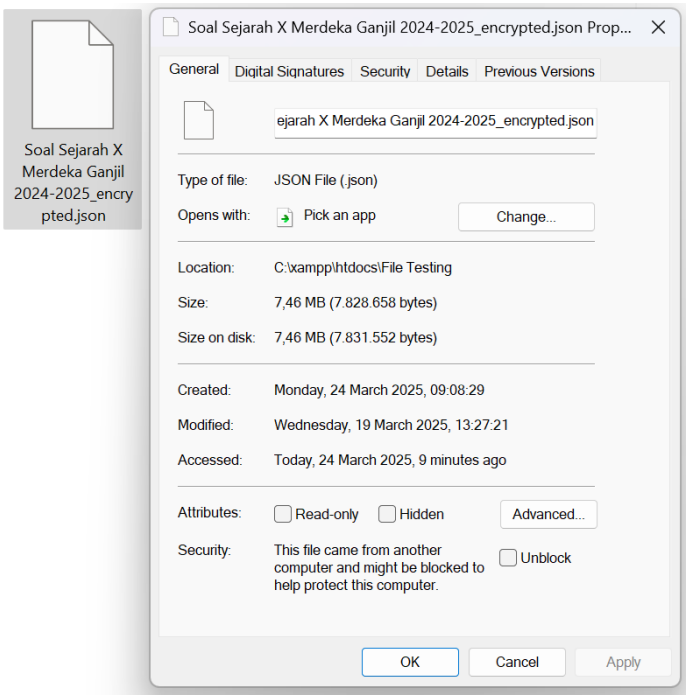
Pada gambar diatas menunjukkan *file ciphertext* pada soal ujian berekstensi JSON telah berhasil di *upload* pada *file manager*.



Gambar 4.31. Struktur *File Ciphertext* Pada Soal Ujian berekstensi JSON

Pada gambar diatas terlihat bahwa struktur *file ciphertext* berekstensi JSON pada soal ujian mengalami perubahan dari kumpulan karakter menjadi bentuk *array* yang berisi nilai [{"a1": "631", "b1": "456"}, {"a2": "358", "b2": "830"}, {"aN": "…", "bN": "…"}]. Nilai-nilai diatas adalah nilai yang

mewakili a dan b hasil *ciphertext* pada perhitungan algoritma kriptografi ElGamal.



Gambar 4.32. Ukuran *File Ciphertext* Pada Soal Ujian berekstensi JSON

Pada gambar diatas menunjukkan ukuran *file ciphertext* pada soal ujian berekstensi JSON sebesar 7.47 MB atau 7.828.658 *bytes*. Dari gambar diatas terjadi perubahan ukuran atau *size file* soal ujian dari 120 kb atau 123.708 *bytes* menjadi 7.47 MB atau 7.828.658 *bytes.*

c. Hasil Pengujian Sistem *Generate Key* Pada *Public Key* dan *Private Key*

Hasil pengujian sistem *generate key* pada *public key* dan *private key* akan menunjukkan bagaimana hasil pembentukkan kunci oleh sistem. Terdapat 2 jenis pembangkitan kunci pada sistem ini, yaitu *public key* yang dibangkitkan dengan statis atau tidak berubah kecuali mengubah struktur *code* PHP pada sistem dan yang kedua adalah *private key* yang dibangkitkan pada setiap *file* soal ujian diunggah menggunakan teknik *hashing* dengan

kunci PIN untuk keamanan tambahan. Berikut adalah gambar *public key* dan *private key* yang dibangkitkan oleh sistem:

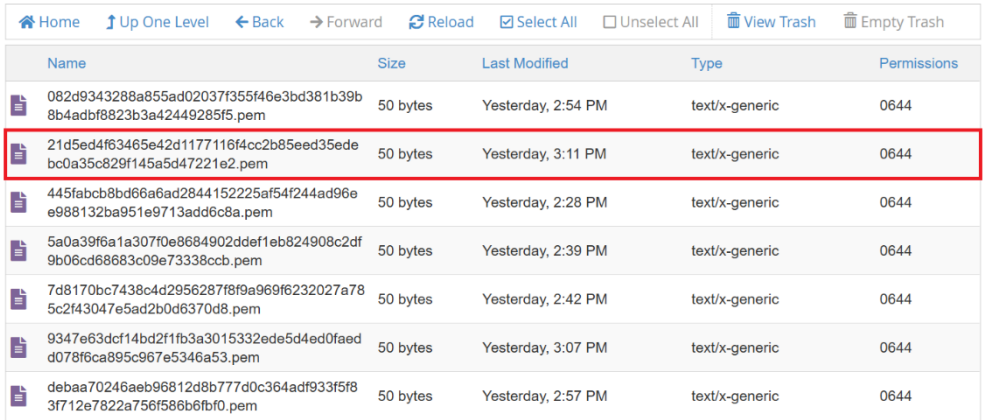
```php
// Konfigurasi ElGamal
$p = 2503; // Bilangan prima besar
$g = 2;     // Generator untuk grup

// Fungsi untuk membangkitkan publik key dan private key
function generate_keys($p, $g) {
    $x = random_int(1, $p - 2); // Kunci privat harus berada di antara 1 dan p-2
    $y = bcpowmod($g, $x, $p); // Kunci publik: y = g^x mod p

    return [
        'public' => ['p' => $p, 'g' => $g, 'y' => $y],
        'private' => $x
    ];
}
```

Gambar 4.33. Penulisan *Code Generate Key* Pada *Public Key* dan *Private Key*

Pada gamabar diatas menunjukkan kunci p dan g dibuat dengan nilai statis, sementara kunci x dibuat dengan fungsi *random_int* yang artinya mengacak angka dari 1 sampai 2 angka sebelum 2503. Sementara kunci y dibuat dengan menghitung menggunakan fungsi *bcpowmod*.



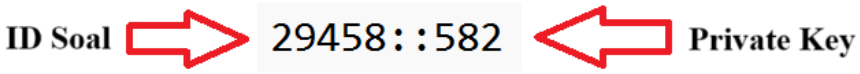
Gambar 4.34. Pembangkitan *Private Key* Tersimpan Pada *File* PEM

Pada gambar diatas menunjukkan *file private key* pada soal ujian berekstensi PEM telah berhasil di *generate* oleh sistem pada *file manager*.

```
WG/F7qDDIqKhCDFIpYNwBQ==::TqJh7TxzwTmbb3prpP5lTg==
```

Gambar 4.35. Isi *Private Key File* PEM yang Dienkripsi

Pada gambar diatas menunjukkan isi dari *file private key* pada soal ujian berekstensi PEM telah berhasil di *generate* oleh sistem pada *file manager*. Isi *file* tersebut terlihat acak dikarenakan dienkripsi menggunakan kunci PIN.

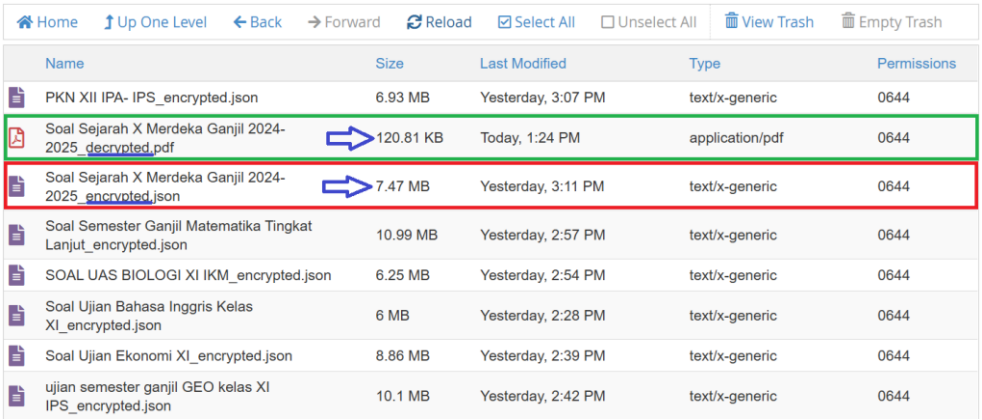


Gambar 4.36. Isi *Private Key* Pada *File* PEM yang Didekripsi

Pada gambar diatas menunjukkan isi dari *file private key* pada soal ujian berekstensi PEM yang telah berhasil didekripsi oleh sistem menggunakan kunci PIN.

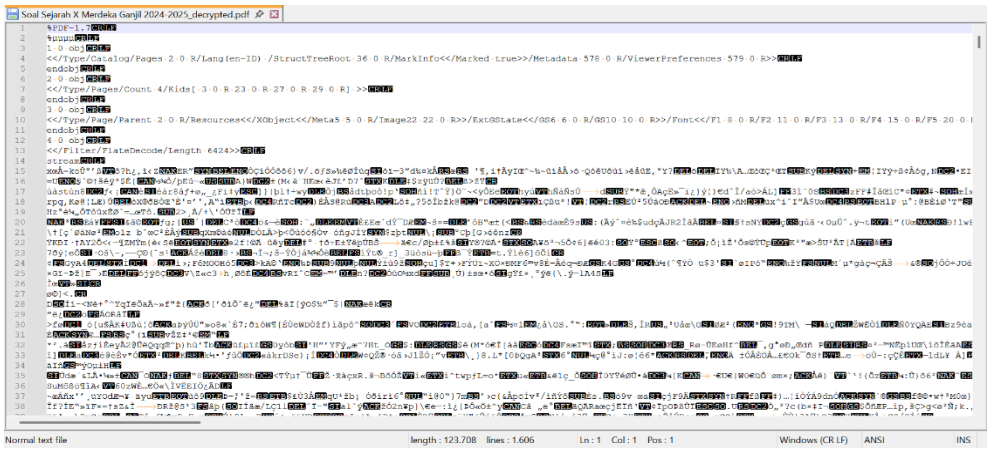
d. Hasil Pengujian Sistem Pada *File* Dekripsi Pada Soal Ujian

Hasil pengujian sistem pada *file* dekripsi pada soal ujian untuk membuktikan bahwa *file plaintext* harus sama persis dengan *file* hasil dekripsi baik dari segi struktur *file* maupun ukuran *file*. Berikut adalah gambar struktur dan ukuran *file* dekripsi pada soal ujian:



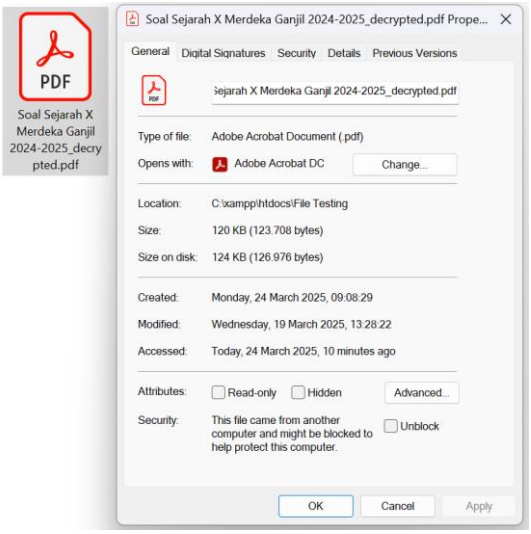
Gambar 4.37. File *Ciphertext* Soal Ujian yang Terdekripsi Pada Sistem

Pada gambar diatas menunjukkan *file ciphertext* soal ujian yang terdekripsi pada sistem dengan menduplikat *file ciphertext* terlebih dahulu, lalu sistem melakukan proses dekripsi *file* soal ujian.



Gambar 4. 38. Struktur *File* Dekripsi Pada Soal Ujian berekstensi PDF

Pada gambar diatas menunjukkan struktur *file* dekripsi pada soal ujian berekstensi PDF. Pada gambar tersebut struktur nilai-nilai array sebelumnya telah berubah kembali menjadi struktur karakter-karakter tersusun seperti pada struktur *file plaintext* sebelumnya, jenis karakter-karakter serta banyaknya karakter juga terlihat sama seperti *file plaintext*.



Gambar 4. 39. Ukuran *File* Dekripsi Pada Soal Ujian berekstensi PDF

Pada gambar diatas terlihat bahwa ukuran *file* dekripsi juga telah sama persis dengan ukuran *file plaintext* yakni 123.708 *bytes.* Maka dengan hasil pengujian diatas, proses enkripsi dan dekripsi pada algoritma kriptografi ElGamal pada data soal ujian telah berhasil dilakukan.

4.3. Kelebihan dan Kekurangan Sistem

Pada tahap ini, penulis akan membahas evaluasi terhadap sistem yang telah dikembangkan, mencakup aspek kelebihan serta kekurangan berdasarkan hasil pengujian dan analisis.

4.3.1. Kelebihan Sistem

Sistem yang telah dirancang memiliki beberapa keunggulan, di antaranya:

1. Dengan menerapkan algoritma kriptografi ElGamal, data soal ujian pada sistem bank soal terenkripsi dengan baik, sehingga keamanan dapat lebih terjamin dari kebocoran oleh pihak yang tidak berwenang.
2. Memudahkan dalam mengakses sistem bagi *user* tanpa perlu instalasi aplikasi tambahan.
3. Kemudahan sistem antar *user* dan administrator dalam bertukar informasi dalam proses mengunduh data soal ujian yang terintegrasi dengan notifikasi email.
4. Antarmuka pengguna (*interface*) sistem dirancang sederhana dan mudah digunakan, bahkan bagi pengguna non-teknis.

4.3.2. Kekurangan Sistem

Adapun beberapa kekurangan sistem yang telah dirancang, di antaranya:

1. Batasan ukuran *file* yang dapat diunggah pada sistem masih terbatas, yaitu maksimal 1 *megabytes*.
2. Proses perhitungan algoritma kriptografi ElGamal menggunakan matematika diskrit yang cukup komplek menyebabkan proses enkripsi dan dekripsi berjalan cukup lambat dan memakan banyak *resource* pada *hardware.*
3. Sistem sangat membutuhkan akses internet dikarenakan *server* pada sistem berada pada *hosting online.*
4. Saat ini sistem masih berdiri sendiri dan belum terhubung dengan sistem akademik atau *e-learning* yang digunakan oleh sekolah.

BAB V
KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian ini, dapat disimpulkan beberapa hal mengenai sistem yang telah dibangun. Adapun kesimpulan yang dapat penulis ambil dalam membangun sistem ini adalah sebagai berikut:

1. Sistem yang telah dibangun berhasil melakukan proses pembangkitan kunci, proses enkripsi dan proses dekripsi pada pada soal ujian berekstensi PDF dengan algoritma kriptografi ElGamal.
2. Sistem yang telah dibangun tidak merubah struktur *file* maupun ukuran *file* pada *file plaintext* soal ujian sebelum proses enkripsi maupun setelah proses dekripsi pada *file* terdekripsi soal ujian dengan algoritma kriptografi ElGamal.
3. Sistem yang telah dibangun dapat menyembunyikan *private key* dengan keamanan tambahan yang bahkan tidak diketahui oleh *user* maupun administrator.
4. Keamanan dalam menjaga akses *login* dan kunci PIN soal ujian menjadi sangat penting, dikarenakan untuk menjaga agar pengguna lain tidak dapat mengakses data soal ujian miliknya.

5.2. Saran

Untuk menyempurnakan sistem yang telah dibangun, maka penulis memberikan saran-saran yang nantinya diharapkan agar dapat memperbaiki beberapa kelemahan-kelemahan yang ada pada sistem ini. Adapun saran yang penulis berikan adalah sebagai berikut:

1. Penulis berharap agar pihak sekolah SMA Adlin Murni nantinya dapat menggunakan sistem ini dengan baik dan dengan ketentuan yang telah diberikan.
2. Penulis berharap agar di masa mendatang, sistem ini dapat dikembangkan dengan optimasi pada proses enkripsi dan dekripsi algoritma kriptografi ElGamal sehingga dapat meningkatkan efisiensi waktu penggunaan sistem.
3. Penulis berharap agar di masa mendatang, sistem ini dapat dikembangkan untuk dapat mengenkripsi *file* berekstensi lainnya selain PDF dan dengan ukuran *file* minimal yang lebih besar.
4. Penulis berharap agar di masa mendatang, banyak sistem keamanan dengan yang dibangun dengan algoritma kriptografi ElGamal.

DAFTAR PUSTAKA

Adeniyi, E. A., Falola, P. B., Maashi, M. S., Aljebreen, M., & Bharany, S. (2022). Secure Sensitive Data Sharing Using RSA and ElGamal Cryptographic Algorithms with Hash Functions. *Information (Switzerland)*, *13*(10), 1–14. https://doi.org/10.3390/info13100442

Alfirdaus, M. H., Tahir, M., Dewanti, N. E., Ardianto, R., Azurah, N. N., & Cahyono, N. F. (2023). Perancangan Aplikasi Enkripsi Deskripsi Mengunakan Metode Caesar Chiper Berbasis Web. *Jurnal Teknik Mesin, Industri, Elektro Dan Informatika (JTMEI)*, *2*(2), 64–76.

Amalya, N., Silalahi, S. M. S., Fatricia Nasution, D., Sari, M., & Gunawan, I. (2023). Kriptografi dan Penerapannya Dalam Sistem Keamanan Data. *Jurnal Media Informatika*, *4*(2), 90–93. https://doi.org/10.55338/jumin.v4i2.428

Annas, F. (2020). Perancangan Sistem Informasi Bank Soal Online Di Smp Negeri 3 Matur. *Journal Educative: Journal of Educational Studies*, *4*(2), 150. https://doi.org/10.30983/educative.v4i2.2522

Arisantoso, Yulianti, S. D., Shalahudin, M., Dzikrullah, A. T., Zahranda, H., Julian, M. A., Savana, G. I., Hidayat, D. A., Darma, Z., Rambe, A. A., Rachman, A. F., Ramadhan, M. R., & Julius, A. (2023). Perancangan Dan Pemrograman Web: Memahami Html, Css, Javascript, Php, Serta Web Hosting Secara Praktis Penerbit Cv.Eureka Media Aksara. In *Eureka Media Aksara*.

Asbari, H., & Triandi, B. (2024). *Implementasi Algoritma ElGamal Dan Advanced Encryption Standard (AES) Pada Hybrid Cryptosystem Implementation of the ElGamal Algorithm and Advanced Encryption Standard (AES) in a Hybrid Cryptosystem*. https://www.doi.org/10.22303

Astuti, N. R. D. P., Setiawan, D. P., & Hakika, D. C. (2023). Comparative Study of Elgamal and Luc Algorithm in Cryptographic Key Generation. *ASEAN Engineering Journal*, *13*(4), 61–68. https://doi.org/10.11113/aej.V13.19184

Baraka. (2023). *Mengenal Apa itu Private dan Public Key dalam Dunia Digital?* Biro Perencanaan Sumber Daya Manusia Dan Karir (BARAKA). https://baraka.uma.ac.id/mengenal-apa-itu-private-dan-public-key-dalam-dunia-digital

Butarbutar, R. (2023). Kejahatan Siber Terhadap Individu: Jenis, Analisis, Dan Perkembangannya. *Jurnal Hukum & Pembangunan*, *2*(2). https://doi.org/10.21143/telj.vol2.no2.1043

Clivan, T., Sugiarso, B. A., & Sinsuw, A. A. E. (2019). Aplikasi Website Perpustakaan Berbasis QRCode. *Jurnal Teknik Informatika*, *14*(1), 1–8.

Emmanuel, A. A., E, O. A., O, A. M., & O, A. E. (2021). A Note on Time and Space Complexity of RSA and ElGamal Cryptographic Algorithms. *International Journal of Advanced Computer Science and Applications*, *12*(7), 143–147. https://doi.org/10.14569/IJACSA.2021.0120716

Frisdayanti, A. (2019). Peranan brainware dalam sistem informasi manajemen. *JEMSI - Jurnal Ekonomi Dan Manajemen Sistem Informasi*, *1*(September), 60–69. https://doi.org/10.31933/JEMSI

Ghosh, A., & Karforma, S. (2022). *Object Oriented Analysis And Design Of El Gamal Digital Signature Algorithm For Document Authentication In E-Learning*. *19*(2), 1735–188. http://www.webology.org

Gusli, A., Furqan, M., & Kurniawan, R. (2021). Document Based Text Data Security Using the Prime Generator Algorithm Fermat's and the ElGamal Algorithm. *Infokum*, *10*(1), 810–817.

Harahap, A. Y. N., Gunawan, H., Nst, A. B., & Sari, R. E. (2022). Penerapan Elgamal Guna Meningkatkan Keamanan Data Text Dan Docx. *It (Informatic Technique) Journal*, *10*(1), 76. https://doi.org/10.22303/it.10.1.2022.76-86

Hidayat, W., Syahputra, M. A., Amrullah, M. F., Susanto, L., & Putri, A. S. (2023). *Analisis Upaya Meningkatkan Keamanan Komputer*. *01*, 29–36. https://journal.diginus.id/index.php/ITEJ/index

Husaini, F., Pardede, A. M. H., & Gultom, I. (2022). Penerapan Enkripsi Menggunakan Metode Elgamal guna Meningkatkan Keamanan Data Text dan Gambar. *JUKI: Jurnal Komputer Dan Informatika*, *4*(1), 67–73.

Imran, O. A., Yousif, S. F., Hameed, I. S., Al-Din Abed, W. N., & Hammid, A. T. (2020). Implementation of El-Gamal algorithm for speech signals encryption and decryption. *Procedia Computer Science*, *167*(Iccids 2019), 1028–1037. https://doi.org/10.1016/j.procs.2020.03.402

Laz, M. El, Grégoire, B., & Rezk, T. (2021). *Security Analysis of ElGamal Implementations*.

Mallouli, F., Hellal, A., Sharief Saeed, N., & Abdulraheem Alzahrani, F. (2019). A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms. *Proceedings - 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019*, 173–176. https://doi.org/10.1109/CSCloud/EdgeCom.2019.00022

Mallouli, F., Hellal, A., Sharief Saeed, N., & Abdulraheem Alzahrani, F. (2019). A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms. *Proceedings - 6th IEEE International Conference on Cyber Security and Cloud Computing, CSCloud 2019 and 5th IEEE International Conference on Edge Computing and Scalable Cloud, EdgeCom 2019*, 173–176. https://doi.org/10.1109/CSCloud/EdgeCom.2019.00022

Noviana, R. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql. *Jurnal Teknik Dan Science*, *1*(2), 112–124. https://doi.org/10.56127/jts.v1i2.128

Nugraha, S. N. (2024). Penerapan Algoritma Kriptografi Elgamal Pada Aplikasi Pengamanan Pesan Berbasis Website. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(3). https://doi.org/10.23960/jitet.v12i3.4794

Qhorifadillah, U., Lestari, S., & Chulkamdi, M. T. (2022). Perancangan Aplikasi Bank Soal Berbasis Website Dengan Algoritma Fisher Yates Shuffle Dan Cosine Similarity (Studi Kasus Di Smk Indraprasta Wlingi). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *6*(1), 352–359. https://doi.org/10.36040/jati.v6i1.4232

Rahim, T. (2021). Enkripsi Dan Dekripsi File Dokumen Dengan Metode Elgamal. In *Agustus: Vol. X* (Issue 2).

Ramadhani, S., & Tanti, L. (2024). *Rancang Bangun Aplikasi Keamanan Data Penjualan Menggunakan Metode Elgamal Berbasis Web Pada PT. Pixelindo Design and Development of Sales Data Security Applications Using the Web-Based Elgamal Method at PT. Pixelindo*. https://www.doi.org/10.22303

Ramadhani, S., & Tanti, L. (2024). *Rancang Bangun Aplikasi Keamanan Data Penjualan Menggunakan Metode Elgamal Berbasis Web Pada PT . Pixelindo*. *1*, 985–1001.

Rasiban, Septiansyah, A., Hasanah, S., Permatasari, V. N., & Yuliawati, A. (2024). Sistem Informasi Otomatisasi Pelaporan Data Penjualan Toko Buku Nazwa Yang Masuk Dan Yang Keluar. *Informatika*, *8*(1), 283–284. https://doi.org/10.37817/ikraith-informatika.v8i1

Riza, F., Muttaqin, M., Pandia, S., Mufida, F. D., Siregar, R., Adytia, P., Wahyuni, W., Simarmata, T. M. D. J., & Lubis, M. (2020). Pengantar Ilmu Kriptografi. In *Journal GEEJ* (Vol. 7, Issue 2).

Saputro, T. H., Hidayati, N., & Ujianto, E. I. H. (2020). Survei Tentang Algoritma Kriptografi Asimetris. *Jurnal Informatika Polinema*, *6*(2), 67–72. https://doi.org/10.33795/jip.v6i2.345

Sihombing, R., Yasir, A., & Satria, W. (2024). IMPLEMENTASI ALGORITMA ELGAMAL DALAM MENJAGA KEASLIAN DOKUMEN DIGITAL. *Jurnal Teknologi Informasi*, *5*(1). https://doi.org/10.46576/djtechno

Sinaga, T. E. (2021). *Perbedaan Public Key dan Private Key – SSL Indonesia*. SSL Indonesia. https://sslindonesia.com/perbedaan-public-key-dan-private-key-ssl-indonesia/

Sitohang, N. (2023). Penerapan Metode Five Modulus Dalam Mengkompresi File Dokumen (PDF) Pengembangan Bahan Ajar. *Jurnal Sains Informatika Terapan*, *2*(1), 1–5. https://doi.org/10.62357/jsit.v2i1.110

Skyhigh Security. (n.d.). *Perbandingan antara tokenisasi dan enkripsi*. Retrieved February 23, 2025, from https://www.skyhighsecurity.com/id/cybersecurity-defined/tokenization-vs-encryption.html

Suhandinata, S., Rizal, R. A., Wijaya, D. O., Warren, P., & Srinjiwi. (2019). Analisis Performa Kriptografi Hybrid Algoritma Blowfish Dan Algoritma Rsa. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, *6*(1), 1–10. https://doi.org/10.33330/jurteksi.v6i1.395

Suhardi, I. (2023). *PENGEMBANGAN BANK SOAL BERBASIS COMPUTER BASED TESTING*.

Susilawati, T., Yuliansyah, F., Romzi, M., & Aryani, R. (2020). Membangun Website Toko Online Pempek Nthree Menggunakan Php Dan Mysql. *Jurnal Teknik Informatika Mahakarya (JTIM)*, *3*(1), 35–44.

Suzana, A. M. (2021). Analisis Dan Perancangan Aplikasi Pemesanan Tiket Bus Berbasis Android. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *2*(3), 353–360. https://doi.org/10.33365/jatika.v2i3.1235

Thakkar, A., & Gor, R. (2023). Cryptographic Method To Enhance Data Security Using Rsa Algorithm and Mellin Transform. *International Journal of Engineering Science Technologies*, *7*(2). https://doi.org/10.29121/ijoest.v7.i2.2023.490

Tutorialspoint. (n.d.). *Public key encryption*. Tutorialspoint. Retrieved February 25, 2025, from https://www.tutorialspoint.com/cryptography/public_key_encryption.htm

Wardani, W. K., Faruq, H. A. Al, & Bakti, B. S. (2024). Desain dan Implementasi Aplikasi Bank Soal Berbasis Web pada Lembaga Kursus Tingkat Sekolah Dasar "Pak Slamet" Design and Implementation of Web-Based Exam Banking Applications for "Pak Slamet" Elementary School Course. In *Jurnal Smart Teknologi* (Vol. 5, Issue 5). http://jurnal.unmuhjember.ac.id/index.php/JST

Wijoyo, A., Rahmawati, T., Agustin, W., Muhammad Saputra, B., & Kurdi. (2023). Perlindungan Data Sensitif : Enkripsi Sebagai Pilar Utama Keamanan Komputer. *CHIPSET*:*Jurnal Ilmu Komputer, Teknik, Dan Multimedia*, *1*(2), 84–91. https://jurnal.publikasimahasiswa.id/index.php/chipset

Wikipedia contributors. (n.d.). *MySQL*. Wikipedia, Ensiklopedia Bebas. Retrieved February 25, 2025, from https://id.wikipedia.org/wiki/MySQL

Wikipedia contributors. (n.d.). *PDF*. Wikipedia, The Free Encyclopedia. Retrieved February 25, 2025, from https://en.wikipedia.org/wiki/PDF

Wikipedia contributors. (n.d.). *Substitution cipher*. Wikipedia, The Free Encyclopedia. Retrieved February 25, 2025, from https://en.wikipedia.org/wiki/Substitution_cipher

Wikipedia contributors. (n.d.). *PHP*. Wikipedia, Ensiklopedia Bebas. Retrieved February 25, 2025, from https://id.m.wikipedia.org/wiki/Berkas:PHP-logo.svg

Zega, I., Mohammad Andri Budiman, & Syahril Efendi. (2023). Comparative Analysis of Ciphertext Enlargement on Generalization of the ElGamal and Multi-factor RSA. *Data Science: Journal of Computing and Applied Informatics*, *7*(1), 44–50. https://doi.org/10.32734/jocai.v7.i1-10360

Ziaurrahman, M., Utami, E., & Wibowo, F. W. (2019). MODIFIKASI KRIPTOGRAFI KLASIK VIGENERE CIPHER MENGGUNAKAN ONE TIME PAD DENGAN ENKRIPSI BERLANJUT. *Jurnal Informasi Interaktif Vol.*, *4*(2).