

# Sistem Keamanan Data Privasi Pelanggan Kos Kosan Dengan Implementasi Algoritma Hybrid

A Data Security System for Boarding House Customer Privacy with Hybrid Algorithm Implementation

**Baby Shakila<sup>1</sup>, Charles Bronson Harahap<sup>2</sup>**

<sup>a</sup>Program Studi Informatika Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan, 20241, Indonesia

<sup>b</sup>Program Studi Informatika Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan, 20241, Indonesia

E-mail: [babyshakila6327@gmail.com](mailto:babyshakila6327@gmail.com)<sup>1</sup> [charlesharahap07@gmail.com](mailto:charlesharahap07@gmail.com)<sup>2</sup>

---

## ABSTRAK

Perkembangan teknologi informasi menuntut sistem pengelolaan data yang aman, terutama dalam lingkungan bisnis seperti kos-kosan yang menyimpan data sensitif pelanggan. Penelitian ini merancang dan mengimplementasikan sistem keamanan data privasi pelanggan kos-kosan dengan pendekatan algoritma kriptografi hybrid yang menggabungkan algoritma RSA dan AES. RSA digunakan untuk mengenkripsi kunci AES secara asimetris, sementara AES mengenkripsi data pelanggan secara simetris untuk efisiensi. Sistem dikembangkan menggunakan Django Framework dan PostgreSQL sebagai basis data, serta diuji menggunakan metode black-box testing. Hasil implementasi menunjukkan bahwa pendekatan hybrid mampu menjaga kerahasiaan dan integritas data pelanggan, sekaligus memberikan perlindungan ganda terhadap akses tidak sah. Sistem juga menyediakan antarmuka admin yang sederhana namun aman untuk manajemen data, enkripsi, dan dekripsi. Penelitian ini membuktikan bahwa kombinasi RSAAES efektif dan layak diterapkan dalam sistem pengelolaan kos-kosan berskala kecil hingga menengah. Saran untuk pengembangan selanjutnya meliputi penambahan sistem manajemen kunci otomatis dan fitur autentikasi lanjutan.

**Kata Kunci :** Kriptografi Hybrid, RSA, AES, Keamanan Data, Kos-Kosan, Django

## ABSTRACT

The development of information technology demands a secure data management system, especially in business environments such as boarding houses that store sensitive customer data. This study designs and implements a data security system for boarding house customer privacy using a hybrid cryptographic algorithm approach that combines the RSA and AES algorithms. RSA is used to asymmetrically encrypt the AES key, while AES symmetrically encrypts customer data for efficiency. This system was developed using the Django Framework and PostgreSQL as the database and tested using black-box testing methods. The implementation results show that the hybrid approach is capable of maintaining the confidentiality and integrity of customer data, while providing double protection against unauthorized access. The system also provides a simple yet secure admin interface for data management, encryption, and decryption. This study proves that the RSAAES combination is effective and feasible for implementation in a small- to medium-scale boarding house management system. Suggestions for further development include the addition of an automated key management system and advanced authentication features.

**Keywords :** Hybrid Cryptography, RSA, AES, Data Security, Boarding House, Django

---

Disubmit:

Info Artikel  
Direview:

Diterima :

---

Copyright © 2025 - senadimu. All rights reserved.

---

## 1. PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan besar dalam cara pengelolaan data, terutama data bersifat sensitif seperti identitas dan transaksi pelanggan. Dalam konteks pengelolaan tempat tinggal seperti kos-kosan, data pelanggan seperti nama, alamat, nomor kontak, dan riwayat pembayaran merupakan aset penting yang harus dijaga kerahasiaannya. Sayangnya, masih banyak pengelola kos-kosan yang belum menerapkan sistem pengamanan digital secara optimal, sehingga menimbulkan potensi risiko kebocoran data.

Berdasarkan penelitian Salah (2023), penerapan algoritma hybrid yang menggabungkan RSA, AES, dan Diffie-Hellman dapat meningkatkan ketahanan sistem terhadap serangan brute force dan sniffing. Studi tersebut membuktikan bahwa pendekatan hybrid lebih unggul dalam melindungi data sensitif dibandingkan algoritma tradisional, terutama ketika digunakan dalam sistem informasi yang membutuhkan keamanan tingkat tinggi.

Berdasarkan penelitian Gour dkk. (2024), pendekatan hybrid cryptography terbukti lebih efisien dalam pengelolaan data skala menengah. Penelitian ini menekankan bahwa AES mampu memberikan kecepatan proses enkripsi, sementara RSA berperan penting dalam menjaga kerahasiaan kunci enkripsi. Kombinasi keduanya menghasilkan sistem yang optimal, baik dari sisi performa maupun keamanan.

Berdasarkan penelitian Li (2024), algoritma hybrid digunakan pada sistem big data kesehatan untuk melindungi data pasien. Implementasi tersebut memperlihatkan hasil yang signifikan dalam menjaga privasi dan integritas data sensitif. Dengan penggunaan RSA untuk distribusi kunci dan AES untuk enkripsi data, penelitian ini menegaskan bahwa pendekatan hybrid cocok diterapkan pada berbagai bidang yang membutuhkan tingkat keamanan tinggi, termasuk data privasi pelanggan.

Kos-kosan "Kamar-koe" merupakan salah satu contoh di mana pengelolaan data pelanggan belum dilengkapi dengan sistem keamanan digital yang memadai. Data pelanggan masih rentan terhadap akses tidak sah atau manipulasi oleh pihak yang tidak bertanggung jawab. Permasalahan ini perlu ditanggapi serius, mengingat tingginya nilai privasi dan kepercayaan yang diberikan pelanggan kepada pengelola.

Penerapan algoritma kriptografi hybrid, yaitu kombinasi antara RSA (Rivest-Shamir-Adleman) dan AES (Advanced Encryption Standard), menjadi salah satu pendekatan efektif untuk meningkatkan keamanan data. RSA digunakan untuk proses pertukaran kunci secara aman, sementara AES digunakan untuk enkripsi data dengan efisiensi tinggi. Kombinasi keduanya memberikan keunggulan dalam aspek kecepatan dan kekuatan perlindungan data.

Dengan mengintegrasikan kedua algoritma tersebut, penelitian ini berfokus pada pengamanan data pelanggan dalam pengelolaan kos-kosan dengan studi kasus pada "Kamar-koe". Pendekatan hybrid diharapkan mampu menghadirkan sistem pengamanan data yang kuat, efisien, dan dapat diaplikasikan dalam skala kecil hingga menengah.

## 2. METODE

### 2.1. Metode Pengumpulan Data

Dalam upaya penerapan algoritma hybrid untuk meningkatkan keamanan data privasi pelanggan, proses pengumpulan data menjadi langkah awal yang penting untuk memastikan bahwa metode yang digunakan relevan dan efektif.

Adapun metode pengumpulan data yang dilakukan meliputi:

#### 1. Studi literatur

Metode ini dilakukan dengan mengkaji berbagai referensi seperti jurnal ilmiah, buku, dan dokumen resmi yang membahas topik terkait algoritma kriptografi, khususnya RSA dan AES. Tujuannya adalah untuk memperoleh pemahaman mendalam tentang prinsip kerja kedua algoritma serta keunggulan dari pendekatan hybrid encryption. Literatur yang dikaji juga mencakup penelitian sebelumnya mengenai implementasi kriptografi pada sistem berbasis web dan pengamanan data sensitif.

## 2. Analisis Prosedur Implementasi Kriptografi

Pengumpulan data dilakukan dengan menganalisis prosedur teknis implementasi algoritma hybrid, termasuk tahapan enkripsi dan dekripsi, manajemen kunci, serta skema integrasi RSA dan AES secara bersamaan. Diagram perancangan seperti flowchart, sequence diagram, dan class diagram digunakan untuk memodelkan alur proses enkripsi dan dekripsi data. Tujuannya adalah memastikan bahwa integrasi algoritma dilakukan secara optimal dan sesuai kebutuhan keamanan data pelanggan.

## 3. Web Browsing

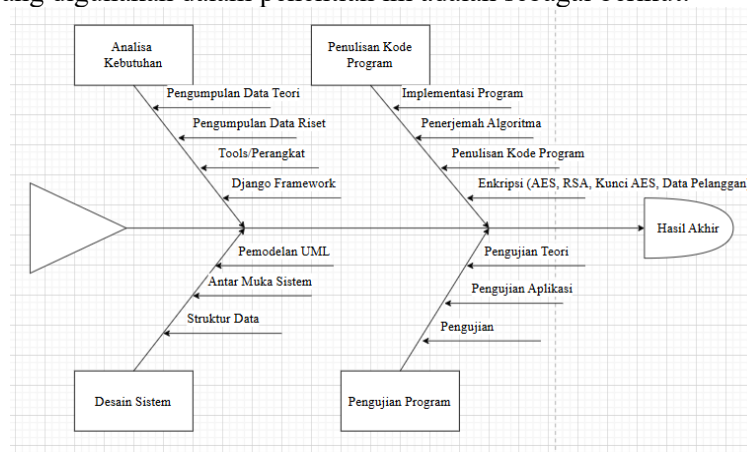
Pengumpulan data dilakukan dengan menjelajahi berbagai sumber daring untuk memperoleh informasi terbaru mengenai praktik terbaik dalam penerapan kriptografi hybrid. Ini meliputi studi kasus, tutorial implementasi, artikel teknis, serta dokumentasi resmi dari framework yang digunakan untuk mendukung proses enkripsi data, seperti Django dan library Python untuk kriptografi. Tujuannya adalah mendukung akurasi teknis dalam proses pengembangan dan evaluasi.

## 4. Studi lapangan

Observasi langsung dilakukan di lingkungan kos-kosan "In The Kost Kamarkoe" untuk memahami bagaimana data pelanggan saat ini dikelola dan risiko apa saja yang mungkin terjadi terhadap privasi informasi. Selain itu, wawancara dilakukan dengan pengelola dan calon pengguna untuk mengidentifikasi jenis data yang paling perlu diamankan, serta ekspektasi terhadap sistem perlindungan data yang digunakan. Tujuan utama dari studi ini adalah menggali kebutuhan dan tantangan aktual yang dapat diselesaikan dengan pendekatan algoritma hybrid RSA-AES.

## 2.2. Metode Penelitian

Tahapan dalam penelitian ini dapat di modelkan dengan *Fishbone* metodologi penelitian. Adapun beberapa tahapan yang digunakan dalam penelitian ini adalah sebagai berikut:



Gambar 1. Kerangka *Fishbone*

Keterangan:

### 1. Analisis Kebutuhan

Pada tahapan ini, peneliti mengumpulkan data-data yang berkaitan dengan penelitian. Peneliti membuat website menggunakan framework django dan bahasa pemrograman python. Peneliti juga menentukan hardware dan software yang akan digunakan untuk penelitian. Berikut adalah hardware dan software yang digunakan:

- Sistem operasi windows 10, ram 4 GB, HDD 1TB
- Visual studio code

- c. Python
- d. Browser seperti google chrome
- e. Laptop

Adapun bahan bacaan yang digunakan untuk mengumpulkan teori:

- a. Jurnal
- b. Buku
- c. Internet

## 2. Desain Sistem

Untuk mendesain sistem atau website, peneliti menggunakan beberapa pemodelan UML, seperti use case diagram, activity diagram, sequence diagram, dan class diagram.

## 3. Penulisan Kode Program

Tahap ini berfokus pada implementasi algoritma kriptografi hybrid RSA-AES untuk meningkatkan keamanan data pelanggan berbasis web menggunakan Django Framework. Proses pengembangan melibatkan perancangan fitur enkripsi dan dekripsi data secara aman, serta integrasi antara logika aplikasi dan pengelolaan basis data.

Tahapan ini dilakukan dengan alat dan langkah berikut:

### a. Perangkat lunak yang digunakan:

- 1) Django Framework, digunakan untuk membangun backend aplikasi dan mengelola routing, view, serta struktur aplikasi secara modular.
- 2) Python, bahasa pemrograman utama untuk mengembangkan logika enkripsi dan dekripsi menggunakan RSA dan AES.
- 3) PostgreSQL, sistem manajemen basis data relasional untuk menyimpan data terenkripsi secara efisien.
- 4) Visual Studio Code, editor kode utama untuk pengembangan aplikasi.
- 5) pgAdmin, Alat untuk pengelolaan database PostgreSQL.

### b. Fitur yang dikembangkan:

- 1) Implementasi Algoritma Hybrid RSA-AES, data sensitif pengguna akan dienkripsi dengan AES (Advanced Encryption Standard), sedangkan kunci AES akan diamankan menggunakan algoritma RSA (Rivest-ShamirAdleman).
- 2) Manajemen Data Terenkripsi, pengguna dapat menginput data, dan sistem akan otomatis melakukan enkripsi sebelum disimpan ke basis data.
- 3) Dekripsi Data Otomatis, data akan didekripsi hanya oleh pihak yang memiliki kunci privat RSA untuk mengakses isi AES key.
- 4) Dashboard Admin, menyediakan antarmuka bagi admin untuk mengelola data dan melakukan proses enkripsi/dekripsi dengan aman.

### c. Proses Implementasi:

- 1) Membuat proyek Django dan aplikasi khusus menggunakan `django-admin startproject` dan `python manage.py startapp`.
- 2) Konfigurasi koneksi basis data PostgreSQL dalam `settings.py`.
- 3) Definisi model entitas untuk data sensitif (misal: identitas pengguna, alamat, nomor identitas) yang akan dienkripsi.
- 4) Integrasi modul kriptografi seperti `cryptography`, `PyCryptodome`, atau `rsa` untuk menjalankan proses hybrid encryption.
- 5) Pengembangan fungsi enkripsi:
  - a. AES untuk mengenkripsi data.
  - b. RSA untuk mengenkripsi key AES.
- 6) Pengembangan fungsi dekripsi:
  - a. Dekripsi key AES menggunakan kunci privat RSA.

- b. Dekripsi data menggunakan key AES.

#### 4. Pengujian Program

Pengujian sistem difokuskan pada aspek keamanan, kecepatan, dan ketepatan proses enkripsi dan dekripsi menggunakan pendekatan hybrid RSA-AES. Pengujian dilakukan untuk memastikan bahwa aplikasi bekerja sesuai dengan spesifikasi perancangan, dengan cakupan berikut:

##### 1) Pengujian Fungsional:

- a. Dilakukan untuk memastikan bahwa setiap proses enkripsi dan dekripsi berjalan sesuai ekspektasi:

- i. Data sensitif berhasil dienkripsi menggunakan AES.
- ii. Kunci AES berhasil diamankan menggunakan RSA.

iii. Proses dekripsi dapat mengembalikan data asli secara utuh.

##### 2) Contoh pengujian:

- a. Pengguna memasukkan data pribadi → sistem mengenkripsi → tersimpan dalam database dalam bentuk terenkripsi.
- b. Admin melakukan dekripsi → data ditampilkan dalam bentuk aslinya.

##### 3) Pengujian Kinerja:

- a. Uji Efisiensi, Mengukur performa sistem saat menangani proses enkripsi-dekripsi untuk data dalam jumlah besar.
- b. Stress Testing, Simulasi beban tinggi untuk melihat kestabilan sistem ketika melakukan banyak proses kriptografi secara simultan.

Pengujian Otomatis, Menggunakan Django Test Framework untuk pengujian unit dan integrasi. Selenium digunakan untuk menguji interaksi pengguna melalui antarmuka web.

#### 5. Hasil Akhir

Setelah tahap pengujian selesai, aplikasi yang telah dirancang menghasilkan beberapa poin evaluasi:

##### a. Kelebihan Aplikasi:

- 1. Keamanan Tinggi, Kombinasi AES dan RSA memberikan perlindungan data yang kuat dan sulit dibobol.
- 2. Performa Stabil, Algoritma hybrid tetap efisien untuk data menengah hingga besar dengan pengelolaan kunci yang sistematis.
- 3. Modular dan Terintegrasi, Implementasi dapat dikembangkan lebih lanjut pada berbagai sistem web yang membutuhkan keamanan data tinggi.

##### b. Kekurangan Aplikasi:

- 1. Proses enkripsi-dekripsi memerlukan waktu lebih lama dibandingkan algoritma tunggal, terutama ketika menangani banyak data sekaligus.
- 2. Belum terdapat sistem manajemen kunci otomatis (misalnya rotasi atau backup key secara terjadwal).

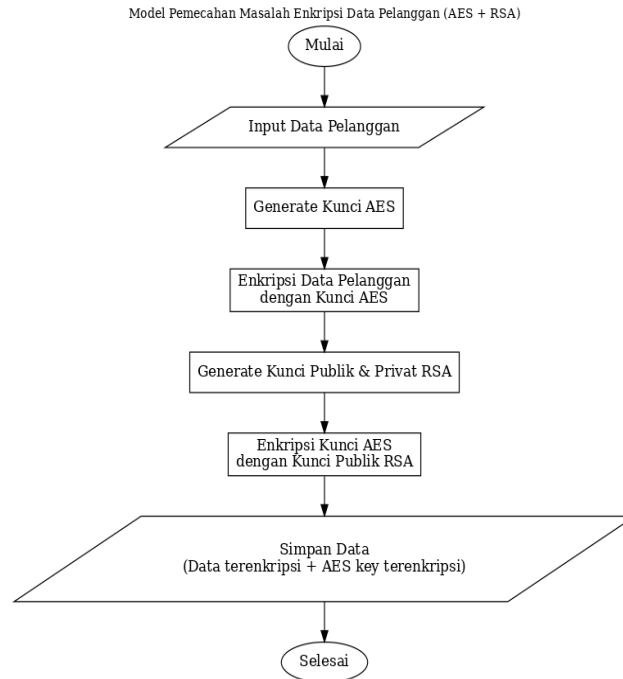
##### c. Kesimpulan dan Rekomendasi:

- 1. Sistem berhasil menerapkan algoritma hybrid RSA-AES secara efektif dalam melindungi data pengguna.
- 2. Disarankan untuk mengembangkan sistem manajemen kunci dan audit log sebagai tambahan fitur keamanan tingkat lanjut.
- 3. Pengembangan selanjutnya dapat mencakup fitur notifikasi keamanan dan integrasi dengan protokol komunikasi aman (HTTPS/SSL).

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pembahasan

Adapun tahapan model pemecahan masalahnya untuk terkait enkripsi berdasarkan gambar 2 berikut:



Gambar 2. Model Pemecahan Masalah Enkripsi

Adapun penjelasan berdasarkan model pemecahan masalah enkripsi pada Gambar III.1 sebagai berikut:

Model pemecahan masalah ini menjelaskan bagaimana proses enkripsi data pelanggan dilakukan dalam sistem keamanan data kos-kosan menggunakan pendekatan kriptografi hybrid. Pendekatan ini menggabungkan algoritma simetris AES untuk enkripsi data, dan algoritma asimetris RSA untuk mengamankan kunci enkripsi (kunci AES). Berikut adalah penjelasan tahapan proses enkripsi sesuai dengan alur diagram:

1. Mulai, proses dimulai saat pengguna atau sistem ingin menyimpan data pelanggan ke dalam basis data dengan perlindungan enkripsi.
  2. Input Data Pelanggan, admin atau sistem memasukkan data pelanggan, seperti nama, alamat, NIK, dan informasi penting lainnya yang bersifat sensitif.
  3. Generate Kunci AES, sistem menghasilkan kunci AES sepanjang 128, 192, atau 256bit secara acak. Kunci ini akan digunakan untuk mengenkripsi data pelanggan.
  4. Enkripsi Data Pelanggan dengan Kunci AES, proses enkripsi AES dilakukan dengan operasi substitusi dan permutasi dalam beberapa putaran.
- Contoh:

**Plaintext = 'HELLO1234'**

**Kunci AES = 'A1B2C3D4E5F6G7H8'**

Output: Ciphertext (hasil enkripsi) tidak bisa dibaca secara langsung.

5. Generate Kunci Publik & Privat RSA, sistem menghasilkan sepasang kunci RSA menggunakan dua bilangan prima besar  $p$  dan  $q$ .

**Rumus dasar:**

$$- n = p \times q$$

$$- \phi(n) = (p-1)(q-1)$$

- Pilih  $e$ , sehingga  $1 < e < \phi(n)$  dan  $\gcd(e, \phi(n)) = 1$

- Tentukan  $d$ , dengan:  $d \equiv e^{-1} \pmod{\phi(n)}$

Kunci Publik: (e, n), Kunci Privat: (d, n)

6. Enkripsi Kunci AES dengan Kunci Publik RSA, misalkan kunci

**AES = 123, e = 17, n = 3233 (contoh kecil), Maka:**

$$\text{Ciphertext RSA} = C = (123^{17}) \bmod 3233 = 855$$

Hasil ini disimpan ke database sebagai AES terenkripsi.

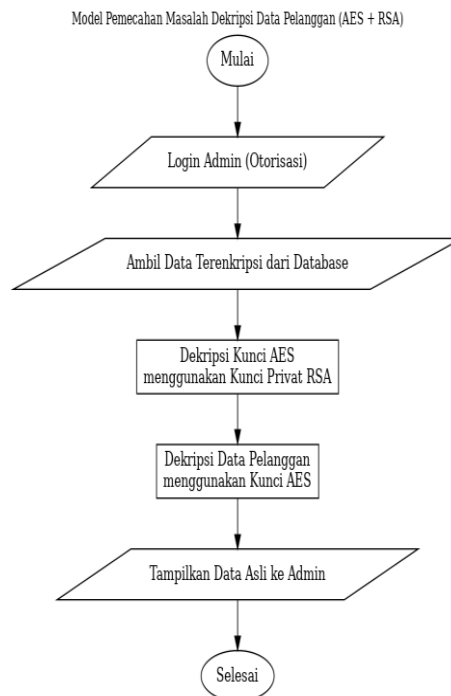
7. Simpan Data, sistem menyimpan hasil enkripsi data pelanggan (AES) dan hasil enkripsi kunci AES (RSA) ke dalam database. Sehingga data yang tersimpan adalah:

- **Ciphertext pelanggan (AES)**

- **Cipher AES (dienkripsi dengan RSA)**

8. Selesai

Proses enkripsi selesai dan sistem siap untuk menyimpan atau menampilkan data yang aman.



Gambar 3. Model Pemecahan Masalah Dekripsi

Model ini menjelaskan bagaimana proses dekripsi dilakukan untuk menampilkan data pelanggan yang sebelumnya telah dienkripsi. Dekripsi dilakukan menggunakan kombinasi algoritma RSA dan AES, di mana RSA digunakan untuk mendekripsi kunci AES, dan AES digunakan untuk mendekripsi data pelanggan. Proses ini hanya dapat dilakukan oleh admin yang telah melalui proses otorisasi.

1. Mulai, proses dimulai ketika admin ingin mengakses data pelanggan yang tersimpan dalam basis data dalam bentuk terenkripsi.
2. Login Admin (Otorisasi), admin harus melewati proses otorisasi terlebih dahulu. Proses ini memastikan bahwa hanya pengguna yang berwenang yang dapat melihat data terenkripsi.
3. Ambil Data dari Database, sistem mengambil dua komponen penting:
  - Data pelanggan yang telah dienkripsi dengan AES
  - Kunci AES yang telah dienkripsi menggunakan RSA
4. Dekripsi Kunci AES menggunakan Kunci Privat RSA Sistem menggunakan RSA private key milik admin untuk membuka (mendekripsi) kunci AES.

Contoh:

$$\text{Ciphertext RSA} = 855, d = 2753, n = 3233$$

$$\text{Plain AES} = (855^{2753}) \bmod 3233 = 123$$

5. Dekripsi Data Pelanggan menggunakan Kunci AES, setelah mendapatkan kembali kunci AES, sistem mendekripsi data pelanggan menggunakan AES dan mengembalikannya ke bentuk semula.
6. Tampilkan Data Asli ke Admin, data asli pelanggan ditampilkan dalam sesi otorisasi admin yang valid. Tidak disimpan ulang dalam bentuk plaintext.
7. Selesai Proses dekripsi selesai dan sistem kembali dalam keadaan aman.

#### Tahapan Perhitungan Manual Hybrid Encryption

##### Enkripsi Data Teks dengan AES

Konversi data transaksi ke byte:

"Rp.1.600.000 tanggal 6 Agustus 2022" → ASCII → byte

Contoh byte: [82, 112, 46, 49, 46, 54, 48, 48, 46, 48, 48, 32, 116, 97, 110]

Padding (PKCS7) agar kelipatan 16 byte:

Panjang asli = 37 byte → Tambah 11 byte (0x0B) → total = 48 byte

Tentukan kunci AES dan IV:

**AES Key (128-bit): 0F1571C947D9E8590CB7ADD6AF7F6798 IV:  
000102030405060708090A0B0C0D0E0F**

Enkripsi CBC secara blok:

- a. Blok pertama P1 (16byte pertama):

$P1 = [82, 112, 46, 49, 46, 54, 48, 48, 46, 48, 48, 32, 116, 97, 110]$

$IV = [0, 1, 2, \dots, 15] \rightarrow XOR \rightarrow \text{hasil } X$

- b. AES Encrypt (X, Key) → C1

- c. Lanjut ke blok kedua dan ketiga, chaining dengan hasil sebelumnya

Enkripsi Kunci AES dengan RSA

RSA Parameter:

$$p = 61, q = 53 \rightarrow n = p * q = 3233 \quad \phi(n) = (p-1)(q-1) = 60 * 52 = 3120$$

Public exponent  $e = 17$

Cari private exponent d:

$$(e \times d) \bmod 3120 = 1 \rightarrow d = 2753 \text{ (hasil extended Euclidean algorithm)}$$

Representasi Kunci AES:

*AES Key = 123 (dalam contoh sederhana)*

*Enkripsi RSA: Hitung secara manual menggunakan square-and-multiply*

$$C = 123^{17} \bmod 3233$$

Langkah-langkah perhitungan RSA:

*Konversi eksponen ke biner:*

*17 = 10001 (biner)*

$$\text{Artinya: } 123^{17} = 123^{(2^4 + 2^0)} = 123^{16} * 123^1$$

Hitung pangkat bertahap modulo 3233:

$$123^1 = 123$$

$$123^2 = 15129 \rightarrow 15129 \bmod 3233 = 3197$$

$$123^4 = 3197^2 = 10222409 \rightarrow 10222409 \bmod 3233 = 181$$

$$123^8 = 181^2 = 32761 \rightarrow 32761 \bmod 3233 = 431$$

$$123^{16} = 431^2 = 185761 \rightarrow 185761 \bmod 3233 = 1480$$

Hitung  $123^{17}$ :

$$123^{17} = 123^{16} \times 123 = 1480 \times 123 = 182040$$

$$182040 \bmod 3233 = 182040 - (56 \times 3233) = 182040 - 181048 = 992$$

*Hasil Enkripsi RSA = 992*

#### III.5.3 Enkripsi File Gambar (Surat Perjanjian & Bukti Pembayaran)

*File bukti.jpeg (120 KB) → 122880 byte*

*Bagi ke blok 16 byte → 7680 blok*

Tiap blok:

*Padding jika perlu  
Lakukan AES\_Encrypt dengan CBC chaining  
Output = ciphertext file (binary atau base64)*

### III.5.4 Contoh Perhitungan Manual Lengkap

Data:

*Nama: Indah Putri Ani  
NIK: 1234567890123456 → byte = 16 byte  
Alamat: Jl. Bilal No. 52  
Bukti Pembayaran (image): 120 KB JPEG  
Surat Perjanjian (image): 200 KB JPEG*

Langkah:

*Ubah data teks (nama, alamat, NIK) ke byte → lakukan AES-CBC  
AES Key diubah ke integer: 123 (contoh untuk RSA)  
RSA enkripsi:  $123^{17} \bmod 3233 = 992$*

Gambar:

*Buka → read byte  
Bagi blok → enkripsi blok per blok (7680 blok untuk 120KB)*

### III.5.5 Dekripsi Proses

*RSA:  $C_{RSA}^d \bmod n \rightarrow 992^{2753} \bmod 3233 = 123$  (kembali ke  
AES key)*

*AES: Gunakan AES\_key dan IV untuk dekripsi AES-CBC  
Unpad hasil dekripsi → data asli kembali*

Keamanan dan Efisiensi

AES memberikan efisiensi dan kecepatan RSA menjamin keamanan pertukaran kunci Hybrid Encryption menggabungkan keduanya secara optimal Hasil enkripsi dapat berupa:

```
{  
  "encrypted_key": 992,  
  "encrypted_data": "AES-Ciphertext...",  
  "iv": "000102030405060708090A0B0C0D0E0F",  
  "encrypted_image": "AES-Image-Ciphertext..."  
}
```

Dengan pendekatan ini, data pelanggan kos-kosan dapat disimpan dan dipertukarkan secara aman, bahkan ketika menggunakan jaringan yang tidak sepenuhnya terpercaya.

## 3.2. Hasil

### 1. Login Admin

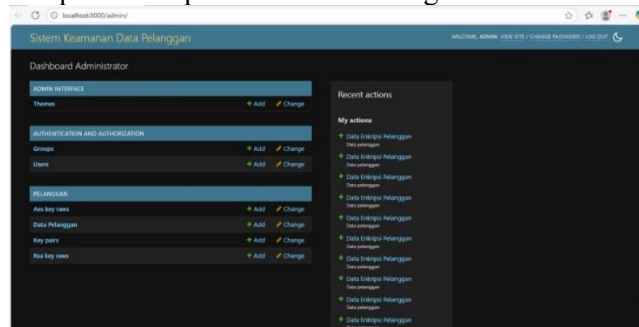
Tampilan login admin merupakan pintu gerbang utama untuk mengakses seluruh fitur dan data yang terdapat dalam sistem keamanan data privasi pelanggan kos-kosan. Halaman ini dirancang dengan antarmuka yang sederhana namun profesional, menekankan pada aspek keamanan autentikasi. Untuk dapat masuk ke dalam sistem, admin diwajibkan memasukkan kombinasi username dan password yang telah terdaftar sebelumnya. Dapat dilihat pada Gambar 4 sebagai berikut:



Gambar 4. Tampilan Login

## 2. Dashboard

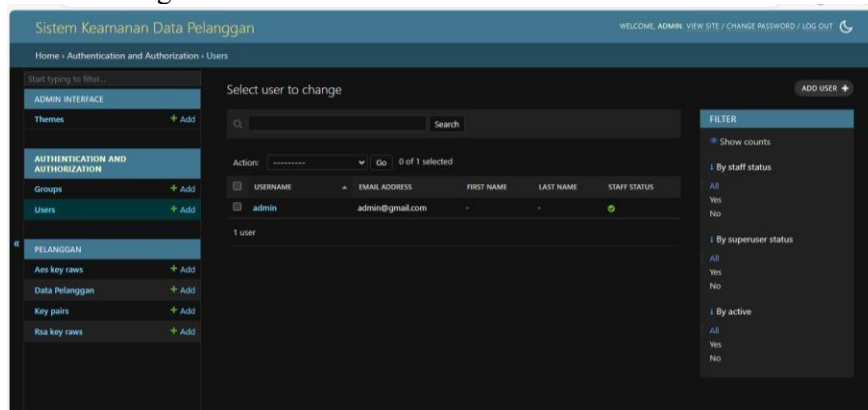
Dashboard merupakan tampilan utama yang ditampilkan setelah admin berhasil login ke dalam sistem. Halaman ini berfungsi sebagai pusat kendali dan monitoring aktivitas sistem secara keseluruhan. Tampilan dashboard dirancang secara ringkas dan informatif, memudahkan admin dalam mengakses fitur-fitur utama seperti manajemen data pelanggan, pengelolaan kunci enkripsi, pemantauan status data, serta log aktivitas sistem. Dapat dilihat pada Gambar 5 sebagai berikut:



Gambar 5. Tampilan Dashboard

## 3. Management User

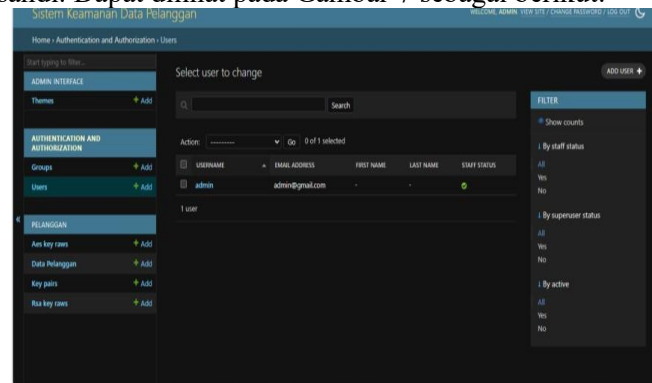
Manajemen user pada sistem ini secara khusus difokuskan untuk mengelola akun admin sebagai pengguna utama yang memiliki otorisasi penuh terhadap seluruh fitur sistem. Admin bertanggung jawab dalam pengelolaan data pelanggan, manajemen enkripsi dan dekripsi data, serta pengawasan terhadap aktivitas pengguna lain. Sistem dirancang agar hanya admin yang memiliki akses terhadap informasi sensitif dan dapat melakukan tindakan-tindakan penting yang berkaitan dengan keamanan data. Dapat dilihat pada Gambar 6 sebagai berikut:



Gambar 6. Tampilan Management User

#### 4. Edit User

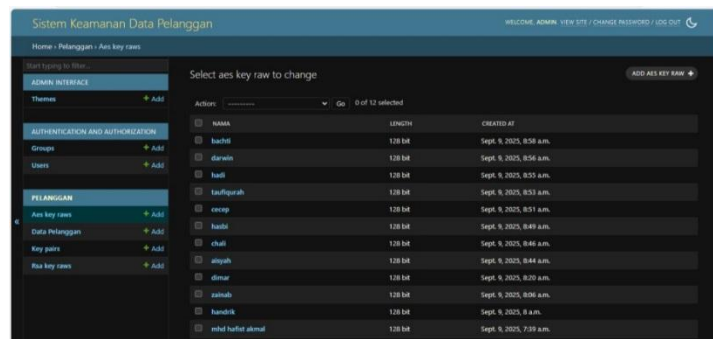
Fitur Edit User disediakan secara eksklusif untuk admin yang memiliki hak akses penuh dalam sistem. Fungsi ini memungkinkan admin untuk mengelola dan memperbarui informasi akun admin lainnya yang telah terdaftar dalam sistem, seperti perubahan data identitas, username, status akun, hingga pengaturan ulang kata sandi. Dapat dilihat pada Gambar 7 sebagai berikut:



Gambar 7. Tampilan Edit User

#### 5. Add Aes Key Raw

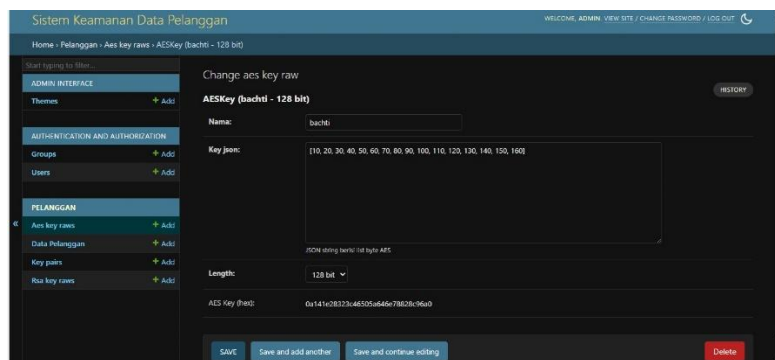
Fitur Add AES Key Raw merupakan bagian penting dalam sistem keamanan yang dirancang untuk mengelola kunci enkripsi simetris (AES) yang digunakan dalam proses pengamanan data pelanggan kos-kosan, khususnya untuk data berbentuk file seperti dokumen kontrak, gambar identitas, atau dokumen lainnya yang membutuhkan kapasitas enkripsi yang efisien dan cepat. Proses ini merupakan bagian dari pendekatan hybrid encryption, di mana kunci AES disiapkan secara manual atau otomatis, lalu dienkripsi menggunakan RSA agar aman saat disimpan di database. Dapat dilihat pada Gambar 8 sebagai berikut:



Gambar 8. Tampilan Add Aes Key Raw

#### 6. Hasil Aes Key Raw

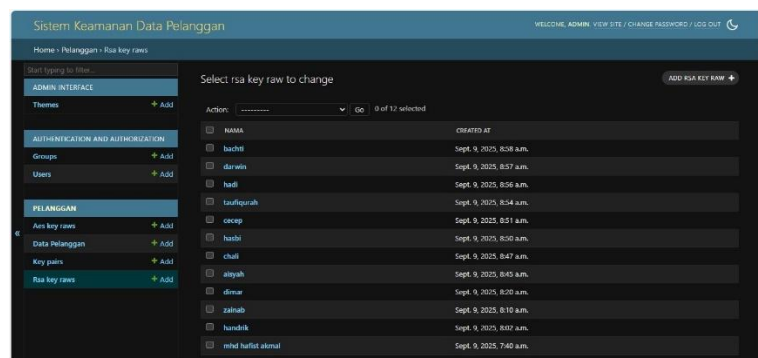
Setelah proses penambahan kunci AES mentah (raw AES key) dilakukan, sistem akan menampilkan hasil dari kunci AES yang telah dibuat dan dienkripsi. Tampilan hasil ini merupakan bagian penting untuk memastikan bahwa proses pembangkitan dan perlindungan kunci telah berjalan sesuai prosedur keamanan. Hasil AES Key Raw ditampilkan dalam dua bentuk, yaitu kunci asli (plaintext) yang masih dalam bentuk mentah sebelum dienkripsi, dan kunci terenkripsi RSA yang disimpan di dalam sistem. Namun, karena alasan keamanan, tampilan kunci asli hanya ditampilkan sekali kepada admin setelah proses pembuatan. Hal ini dilakukan untuk mencegah penyalahgunaan, kebocoran, atau duplikasi kunci oleh pihak yang tidak berwenang. Setelah halaman ditutup atau direfresh, sistem tidak akan lagi menampilkan kunci AES dalam bentuk plaintext. Dapat dilihat pada Gambar 9 sebagai berikut:



Gambar 9. Tampilan Hasil Aes Key Raw

### 7. Add Rsa Key Raw

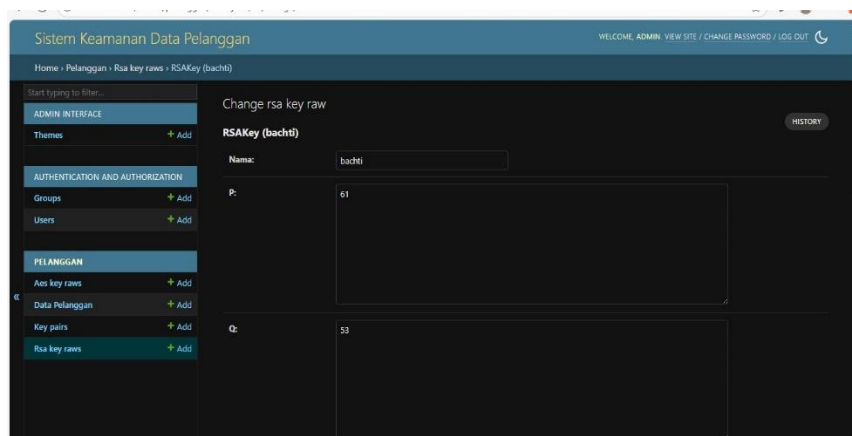
Fitur Add RSA Key Raw merupakan komponen vital dalam sistem keamanan data pelanggan kos-kosan, karena RSA digunakan sebagai lapisan pelindung utama untuk mengamankan kunci AES. RSA berfungsi sebagai algoritma kriptografi asimetris, di mana terdapat sepasang kunci yang terdiri dari public key dan private key. Public key digunakan untuk mengenkripsi data (dalam hal ini mengenkripsi AES key), sementara private key digunakan untuk melakukan dekripsi. Dapat dilihat pada Gambar 10 sebagai berikut:



Gambar 10. Tampilan Add Rsa Key Raw

### 8. Hasil Rsa Key Raw

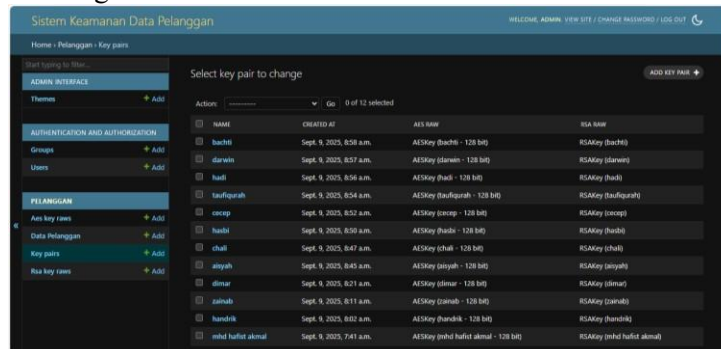
Setelah admin melakukan proses generate RSA key melalui fitur Add RSA Key Raw, sistem akan menampilkan hasil pasangan kunci RSA yang terdiri dari public key dan private key. Tampilan ini disusun secara hati-hati agar memenuhi prinsip keamanan kriptografi asimetris, sekaligus memberikan transparansi terhadap proses generate kunci yang dilakukan. Dapat dilihat pada Gambar 11 sebagai berikut:



Gambar 11. Tampilan Hasil Rsa Key Raw

## 9 Key Pair

Fitur Key Pair merupakan representasi dari pasangan kunci kriptografi RSA yang digunakan sebagai bagian dari skema hybrid encryption dalam sistem. Pada sistem ini, key pair terdiri dari dua komponen penting, yaitu public key dan private key, yang secara matematis saling terkait namun memiliki fungsi yang berbeda. Public key digunakan untuk mengenkripsi kunci AES, sedangkan private key digunakan untuk mendekripsi kunci tersebut saat proses pembukaan data pelanggan dilakukan. Dapat dilihat pada Gambar 12 sebagai berikut:



Gambar 12. Tampilan Key Pair

## 4. KESIMPULAN

Berdasarkan hasil perancangan dan implementasi sistem keamanan data privasi pelanggan kos-kosan dengan algoritma hybrid RSA dan AES, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Implementasi algoritma hybrid RSA-AES berhasil meningkatkan keamanan data pelanggan, dengan cara membagi peran antara kedua algoritma: AES digunakan untuk mengenkripsi data pelanggan karena kecepatannya dalam menangani data berukuran besar, sedangkan RSA digunakan untuk mengenkripsi kunci AES agar tidak mudah diakses oleh pihak yang tidak berwenang. Skema ini memungkinkan data tetap terenkripsi secara kuat sekaligus efisien dari sisi performa.
2. Kombinasi RSA dan AES terbukti dapat bekerja secara sinergis. AES berfungsi sebagai algoritma simetris yang cepat dan ringan untuk proses enkripsi/dekripsi data, sedangkan RSA sebagai algoritma asimetris berfungsi sebagai pelindung kunci AES, sehingga sistem dapat menjaga kunci AES tetap aman meskipun data disimpan di server atau database terbuka. Sinergi ini menyelesaikan permasalahan klasik pada kriptografi simetris yang lemah dalam distribusi kunci.
3. Efektivitas algoritma hybrid RSA-AES dalam menjaga kerahasiaan dan integritas data terbukti tinggi, karena hanya pengguna yang memiliki RSA private key yang sah yang dapat mendekripsi kunci AES dan membuka isi data pelanggan.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Potensi Utama yang telah banyak memberikan masukan dan saran dalam penyelesaian penelitian ini.

## REFERENSI

- [1] Chalooop, Y. A., & Abdullah, R. M. (2021). Comparative analysis of cryptography algorithms: RSA, AES, and hybrid RSA–AES algorithm. *Journal of Engineering and Sustainable Development (JEASD)*, 25(6), 47–57.  
<https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd/article/view/47>
- [2] Hidayat. (2018). Manajemen usaha kos-kosan di lingkungan mahasiswa. *Jurnal Sosial dan Ekonomi*.

- [3] Priyo Sutopo, dkk. (2016). Web sebagai media penelusuran informasi. *Jurnal Teknologi Informasi*, 16(1), 24–30.
- [4] Putra. (2022). Penggunaan PostgreSQL pada sistem informasi berbasis web dengan Django. *Jurnal Teknologi dan Sistem Informasi*.
- [5] Rahmat & Sari. (2022). Keunggulan Django dalam pengembangan aplikasi web yang aman dan andal. *Jurnal Teknologi Informasi Terapan*.
- [6] Rehman, A., Khan, M. A., Saba, T., Rehman, A. U., Tariq, U., & Mehmood, Z. (2021). Hybrid encryption techniques for secure cloud computing. *Electronics*, 10(21), 2673. <https://doi.org/10.3390/electronics10212673>
- [7] Robbins, S., & Coulter, M. (2016). *Management* (13th ed.). Pearson Education.
- [8] Santoso, dkk. (2021). Pengaruh sistem berbasis web terhadap efisiensi manajemen. *Jurnal Teknologi Informasi*.
- [9] Sari, dkk. (2022). Penerapan algoritma RSA untuk pengamanan data pengguna pada aplikasi web. *Jurnal Keamanan Siber Indonesia*.
- [10] Wahyuzi Andriansyah, dkk. (2017). Metodologi Perancangan Sistem Informasi. *Jurnal Ilmu Komputer dan Rekayasa Sistem*.
- [11]] Walle, Y. A. (2024). Software-defined network-based security enhancement for
- [12] MANET using hybrid cryptographic algorithm. *Open Information Science*, 8(1), 20240001. <https://doi.org/10.1515/opis-2024-0001>
- [13] Salah, O. (2023). A Hybrid Algorithm for Enhancement of the Data Security. *International Journal of Advanced Computer Science*. [PDF Link](#)
- [14] Gour, A., et al. (2024). Hybrid Cryptographic Approach: For Secure Data. *E3S Web of Conferences*, 86, 01048. [PDF Link](#)
- [15] Li, W. (2024). A hybrid encryption algorithm based approach for secure data in healthcare big data. *Alexandria Engineering Journal*. [ScienceDirect](#)