**BAB IV**

**HASIL DAN PEMBAHASAN**

1. **Hasil Penelitian**
2. **Analisa Sistem**
3. **Diagram Konteks**

Diagram konteks merupakan data *flow* diagram dengan menggambarkan garis besar operasional sistem (Iswady, 2015). Skema diagram konteks sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.1.



Gambar 4.1. Diagram konteks sistem pengamanan menggunakan algoritma RSA berbasis web di RUSD Anutapura Palu

Keterangan gambar 4.1. dapat dilihat sebagai berikut :

1. Sistem pengamanan data rekam medis

Sistem pengamanan data rekam medismenggunakan algoritma RSA merupakan sistem utama yang melakukan enkripsi dan dekripsi pada *file* dokumen yang diinputkan oleh pengirim maupun penerima.

1. Pengirim

Pengirim menginput *file* dokumen ke sistem untuk menghasilkan *file* dokumen yang terenkripsi. Pengirim juga dapat mengolah data pasien, data lab pasien, data pengguna, data diagnosis, data enkripsi dan data dekripsi

1. Penerima

Penerima menerima *file* dokumen dari pengirim berupa dokumen yang telah terenkripsi kemudian melakukan dekripsi *file* pada sistem untuk mendapatkan dokumen asli.

1. ***Data Flow Diagram* (DFD)**

*Data flow diagram* sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data disimpan. *Data flow diagram* menggambarkan arus data didalam sistem dengan terstruktur dan jelas (Iswady, 2015). Skema DFD level 1 dari sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.2.



Gambar 4.2. DFD level 1 sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

Keterangan gambar 4.2. dapat dilihat sebagai berikut :

1. *Login*

Proses *login* merupakan pintu masuk bagi pengguna untuk mengakses sistem dan dimaksudkan untuk mengatur identifikasi. Untuk melakukan proses *login* pengguna dapat memasukkan nama pengguna dan kata sandi yang telah didaftarkan sebelumnya.

1. Input

Proses input merupakan proses untuk menambahkan data-data pengguna, data-data pasien, data-data lab pasien, data-data diagnosis, data-data enkripsi pada aplikasi. Dan hanya pengguna yang berstatus sebagai pengirim yang dapat melakukan proses ini.

1. Proses enkripsi

Pada proses ini pengirim melakukan enkripsi *file* dokumen sebelum dibagikan ke penerima. Proses enkripsi merupakan proses mengubah pesan asli (plaintext) menjadi pesan rahasia (chipertext).

1. Proses dekripsi

Pada proses ini penerima melakukan dekripsi pada *file* dokumen yang telah dienkripsi sebelumnya oleh pengirim. Proses dekripsi merupakan proses mengubah pesan rahasia (chipertext) menjadi pesan asli (plaintext).

Uraian dari tiap proses yang terdapat pada DFD level 1 dapat digambarkan dalam suatu skema DFD level 2 yang dapat dilihat pada gambar 4.3.



Gambar 4.3. DFD level 2 sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

Keterangan gambar 4.3. dapat dilihat sebagai berikut :

1. Proses input pengguna

Pada proses ini pengirim dapat menginput data pengguna yang bisa mengakses sistem.

1. Proses input data pasien

Pada proses ini pengirim dapat menginput data pasien yang akan akan di diagnosis berdasarkan hasil lab pasien.

1. Proses input data lab pasien

Pada proses ini pengirim dapat menginput data lab pasien yang akan akan di diagnosis berdasarkan hasil lab pasien.

1. Proses enkripsi

Pada proses enkripsi pengirim menginputkan kunci publik yang telah dibentuk oleh penerima dan dokumen kedalam sistem untuk dienkripsi, kemudian dokumen yang telah terenkripsi akan disimpan ke database sistem untuk memudahkan penerima melakukan dekripsi dokumen.

1. Proses dekripsi

Pada proses dekripsi penerima mengambil *file* dokumen yang telah dienkripsi pengirim dari database sistem kemudian melakukan dekripsi *file* dengan menggunakan kunci D dan kunci N yang hanya diketahui oleh penerima.

1. ***Use Case Diagram***

*Use case diagram* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case diagram* bekerja dengan cara mendeskripsikan tipikal interaksi antar pengguna (yang disebut dengan *actor*) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (Afriany & Purnama, 2016). Skema *Use case diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.4.



Gambar 4.4. *Use case diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

Keterangan gambar 4.4. dapat dilihat sebagai berikut

1. *Login*

Pengguna dalam hal ini pengirim dan penerima dapat masuk ke dalam sistem setelah melalui proses *login* dengan memasukkan nama pengguna dan kata sandi yang telah terdaftar didalam sistem.

1. Input Pasien

Pada proses ini pengirim dapat menginput data pasien yang akan akan di diagnosis berdasarkan hasil lab pasien.

1. Input Diagnosis

Pada proses ini pengirim dapat menginput data berdasarkan diagnosis yang telah dilakukan.

1. Input Pengguna

Pada proses ini pengirim dapat menginput data pengguna yang bisa mengakses sistem.

1. Enkripsi File

Pada tahap ini pengirim melakukan enkripsi *file* dokumen dengan menggunakan kunci public dan private yang telah ada pada sistem.

1. Dekripsi File

Pada tahap ini penerima melakukan dekripsi *file* dokumen yang telah dienkripsi sebelumnya oleh pengirim dengan menggunakan kunci rahasia yang hanya diketahui oleh penerima.

1. *Logout*

Pengguna dalam hal ini pengirim dan penerima dapat melakukan *logout* dari sistem.

1. ***Activity Diagram***

Menurut Hendini (2016) *activity diagram* adalah tipe khusus dari *diagram* *state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem dan berfungsi untuk menganalisa proses. *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem proses atau proses bisnis. Skema *activity diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.5 dibawah ini.



Gambar 4.5. *Activity diagram* pengirim sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu



Gambar 4.6. *Activity diagram* penerima sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

1. ***Class Diagram***

Menurut Hendini (2016) *class diagram* adalah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi obyek. *Class diagram* merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Skema *class diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.7.



Gambar 4.7. *Class diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

1. ***Entity Relationship Diagram* (ERD)**

Menurut Iswady (2015) *Entity Relationship Diagram* (ERD) memiliki dua komponen utama yaitu entitas (*entity*) dan relasi (*relational*). Kedua komponen ini masing-masing dilengkapi dengan sejumlah atribut yang mempresentasikan seluruh fakta yang ada di dunia nyata. Skema *entity relationship diagram* **(**ERD) sistem sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.8.



Gambar 4.8. *entity relationship diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

1. ***Sequence Diagram***

Menurut Haviluddin dalam Suendri (2018) *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*. Skema *sequence diagram* sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu dapat dilihat pada gambar 4.9 dan 4.10.



Gambar 4.9. *Sequence diagram* pengirim sistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu



Gambar 4.10. *Sequence diagram* penerimasistem pengamanan data rekam medis menggunakan algoritma RSA berbasis web di RSUD Anutapura Palu

* + 1. **Implementasi Sistem**
       1. **Implementasi Perangkat Keras dan Perangkat Lunak**

Implementasi perangkat keras dan lunak yang digunakan selama pembuatan sistem pengamanan data *file* dokumen berbasis kriptografi adalah menggunakan perangkat keras dengan spesifikasi core i5, RAM 4Gb, dengan monitor resolusi 1366x768 pixel, dan perangkat lunak dengan menggunakan *web* server XAMPP, , dan editor Sublime Text 3.

* + - 1. **Implementasi Basis Data**

Basis data digunakan untuk menyimpan data pengguna, data pasien, data lab pasien, data diagnosis pasien, data file enkripsi dan data file dekripsi. Tabel dalam aplikasi ini sebagai berikut:

1. Tabel rs\_pengguna

Tabel rs\_pengguna digunakan untuk menyimpan data pengguna aplikasi. Struktur tabel data akun dapat dilihat pada tabel 4.1.

Tabel 4.1. rs\_pengguna

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Lenght** | **Index** |
| rs\_pengguna\_id\_ex | varchar | 10 | PRIMARY |
| rs\_pengguna\_pengguna | varchar | 20 | - |
| rs\_pengguna\_nama | text | - | - |
| rs\_pengguna\_sandi | text | - | - |

1. Tabel rs\_enkripsi

Tabel hasil enkripsi digunakan untuk menyimpan seluruh data *file* hasil enkripsi. Struktur tabel enkripsi dapat dilihat pada tabel 4.2.

Tabel 4.2. rs\_enkripsi

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Lenght** | **Index** |
| rs\_enkripsi\_id\_ex | varchar | 30 | PRIMARY |
| rs\_enkripsi\_id | int | 11 | - |
| rs\_enkripsi\_kode | varchar | 12 | - |
| rs\_enkripsi\_kunci | text | - |  |
| rs\_enkripsi\_file | text | - |  |

1. Tabel rs\_dekripsi

Tabel hasil dekripsi digunakan untuk menyimpan seluruh data *file* hasil dekripsi. Struktur tabel dekripsi dapat dilihat pada tabel 4.3.

Tabel 4.3. rs\_dekripsi

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Lenght** | **Index** |
| rs\_dekripsi\_id\_ex | varchar | 30 | PRIMARY |
| rs\_dekripsi\_id | int | 11 | - |
| rs\_dekripsi\_kode | varchar | 12 | - |
| rs\_dekripsi\_kunci | text | - |  |
| rs\_dekripsi\_file | text | - |  |

1. Tabel rs\_diagnosis

Tabel hasil rs\_diagnosis digunakan untuk menyimpan seluruh data hasil diagnosis pasien. Struktur tabel diagnosis pasien dapat dilihat pada tabel 4.4.

Tabel 4.4. rs\_diagnosis

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Lenght** | **Index** |
| rs\_diagnosis\_kode | varchar | 5 | PRIMARY |
| rs\_diagnosis\_nama | text | - | - |

1. Tabel rs\_lab\_pasien

Tabel rs\_lab\_pasien digunakan untuk menyimpan seluruh data lab pasien. Struktur tabel lab pasien dapat dilihat pada tabel 4.5.

Tabel 4.5. rs\_lab\_pasien

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Lenght** | **Index** |
| rs\_lab\_pasien\_id\_ex | varchar | 20 | PRIMARY |
| rs\_lab\_pasien\_pasien | varchar | 11 | - |
| rs\_lab\_pasien\_diag | text | - | - |

1. Tabel rs\_pasien

Tabel rs\_pasien digunakan untuk menyimpan seluruh data pasien. Struktur tabel rs\_pasien dapat dilihat pada tabel 4.5.

Tabel 4.5. rs\_pasien

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Lenght** | **Index** |
| rs\_pasien\_id\_ex | varchar | 20 | PRIMARY |
| rs\_ pasien\_id | int | 11 | - |
| rs\_pasien\_nama | text | - | - |
| rs\_pasien\_jenkel | text | - | - |
| rs\_pasien\_tmp\_lhr | text | - | - |
| rs\_pasien\_tgl\_lhr | text | - | - |
| rs\_pasien\_kerja | text | - | - |
| rs\_pasien\_alamat | text | - | - |
| rs\_pasien\_telp | text | - | - |
| rs\_pasien\_hub | text | - | - |
| rs\_pasien\_agama | text | - | - |
| rs\_pasien\_waktu | text | - | - |
| rs\_pasien\_ket | text | - | - |
| rs\_pasien\_ibu | text | - | - |
| rs\_pasien\_ayah | text | - | - |
| rs\_pasien\_darah | text | - | - |
| rs\_pasien\_iden | text | - | - |
| rs\_pasien\_rad | text | - | - |

* + - 1. **Implementasi *Input***

Implementasi *form* pada sistem digunakan untuk melakukan proses *input* atau konfirmasi terhadap aksi yang dilakukan. Pada aplikasi pengamanan data *file* dokumen terdapat beberapa *form*, antara lain sebagai berikut:

1. *Form* Login

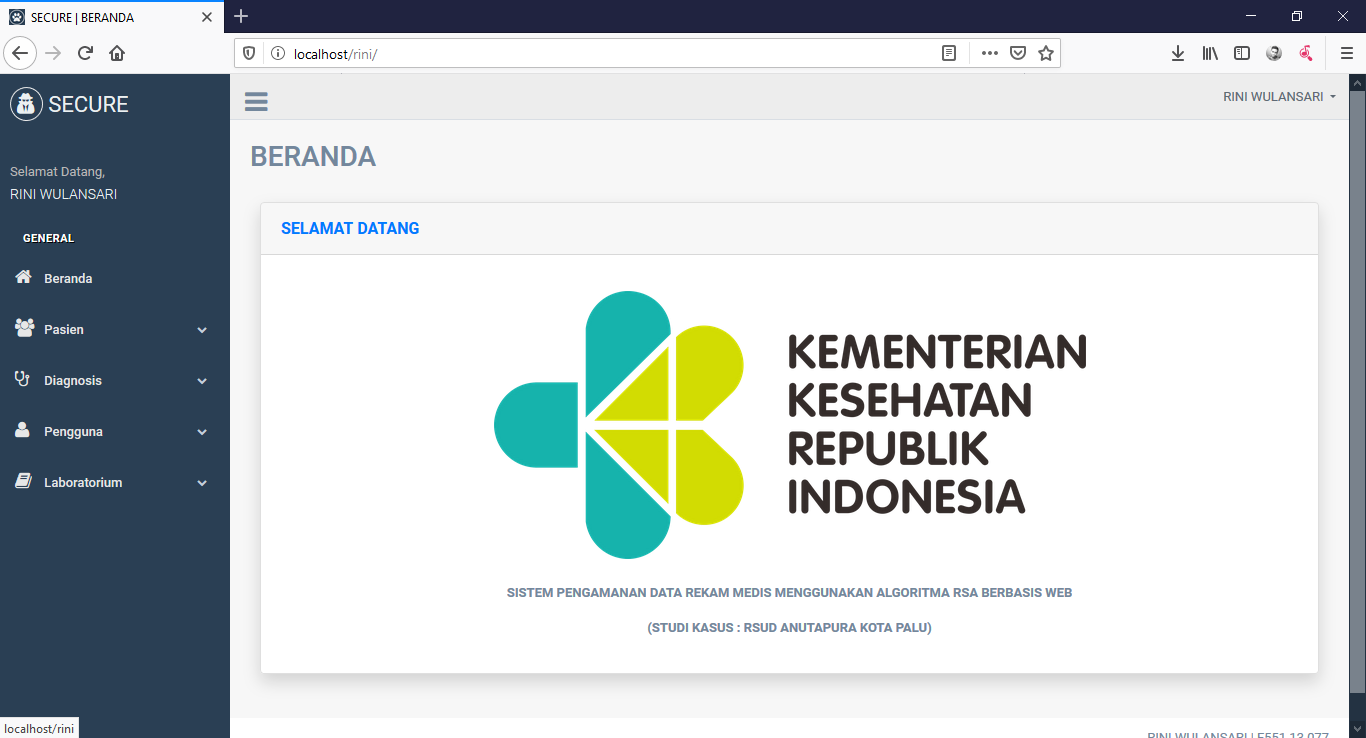
*Form* login digunakan untuk memasukkan akun pengguna agar dapat mengakses aplikasi. *Form login* dapat dilihat pada gambar 4.11

****

Gambar 4.11. Tampilan *Form login*

1. *Form* Beranda

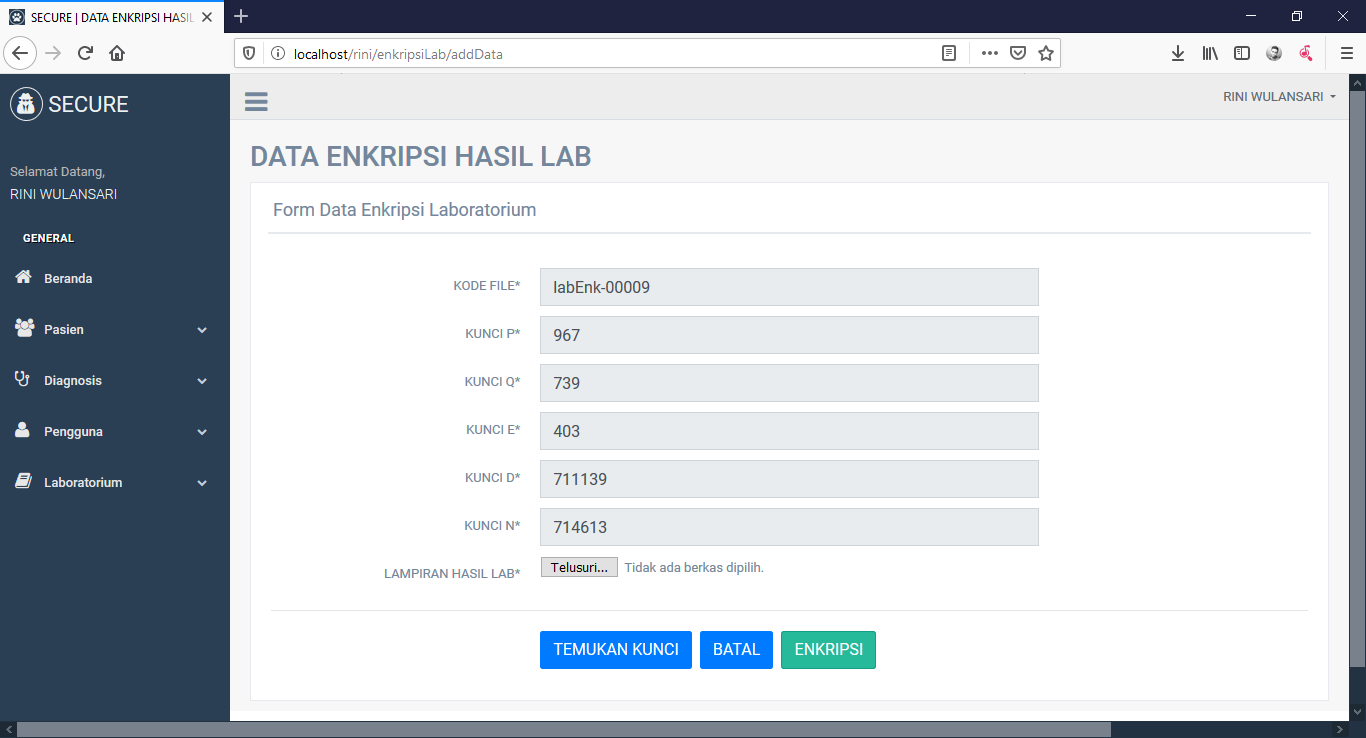
*Form* beranda merupakan halaman awal dari tampilan sistem yang menampilkan selamat datang pada sistem. *Form* beranda dapat dilihat pada gambar 4.12.



Gambar 4.12. Tampilan *Form* beranda

1. *Form* Enkripsi

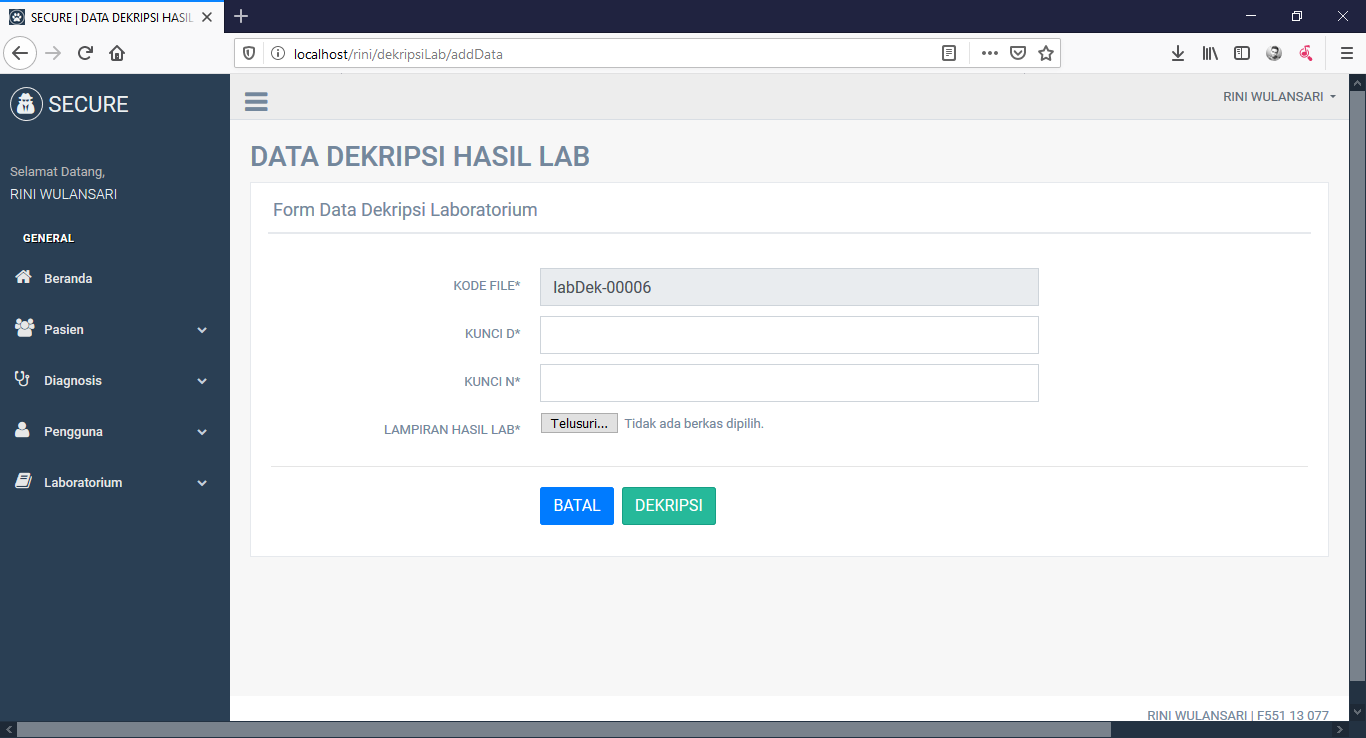
*Form* enkripsi merupakan *form* yang digunakan untuk melakukan enkripsi *file* dokumen. Pada *form* ini juga terdapat tampilan form dengan pembentukan kunci P, Q, E, D dan N. *Form* enkripsi dapat dilihat pada gambar 4.13.



Gambar 4.13. Tampilan *Form* enkripsi

1. *Form* Dekripsi

*Form* dekripsi merupakan *form* yang berfungsi mengembalikan *file* yang telah terenkripsi menjadi seperti semula agar dapat terbaca dengan menginputkan kunci rahasia D dan N. *Form* dekripsi dapat dilihat pada gambar 4.14.



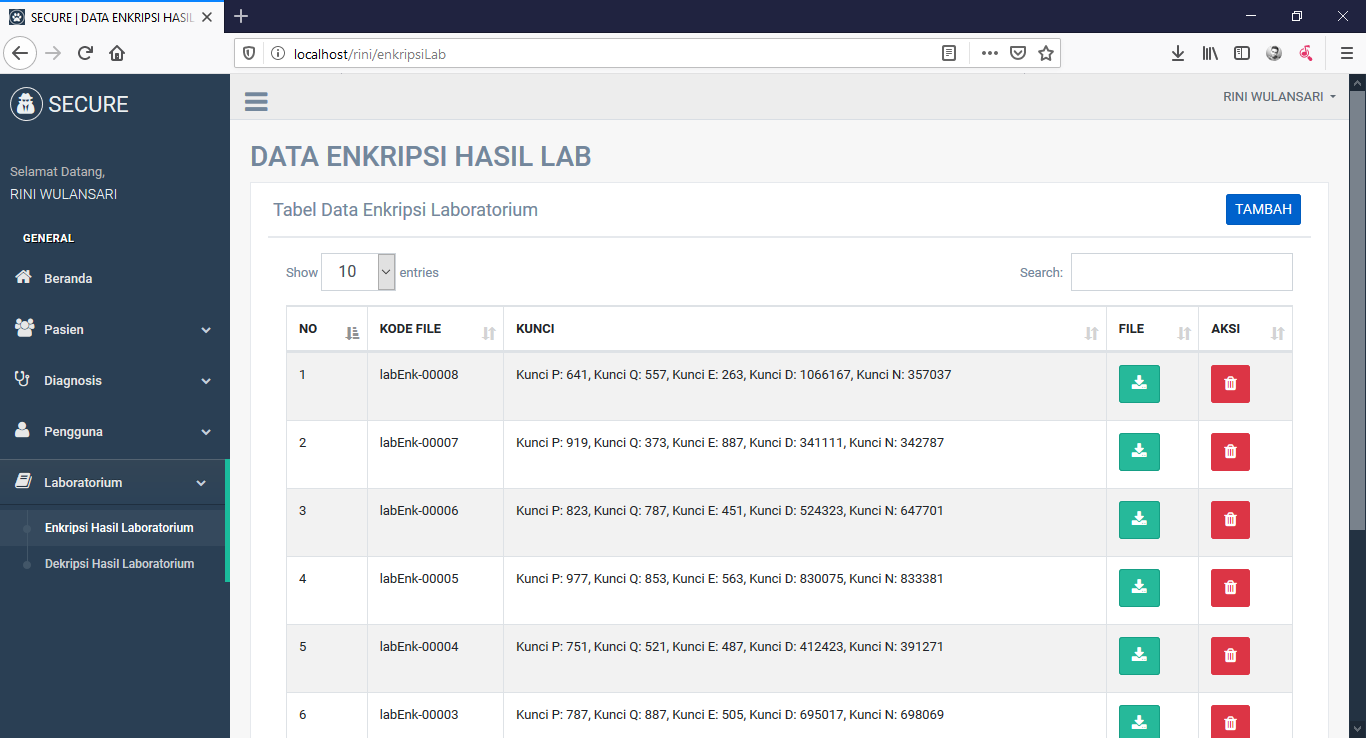
Gambar 4.14. Tampilan *Form* dekripsi

* + - 1. **Implementasi *Output***

Implementasi *output* pada sistem digunakan untuk menampilkan data yang telah tersimpan dan terproses untuk dapat dilihat oleh pengguna. Implementasi tersebut dibagi menjadi beberapa halaman sebagai berikut.

1. Tampilan *Form* Data Enkripsi

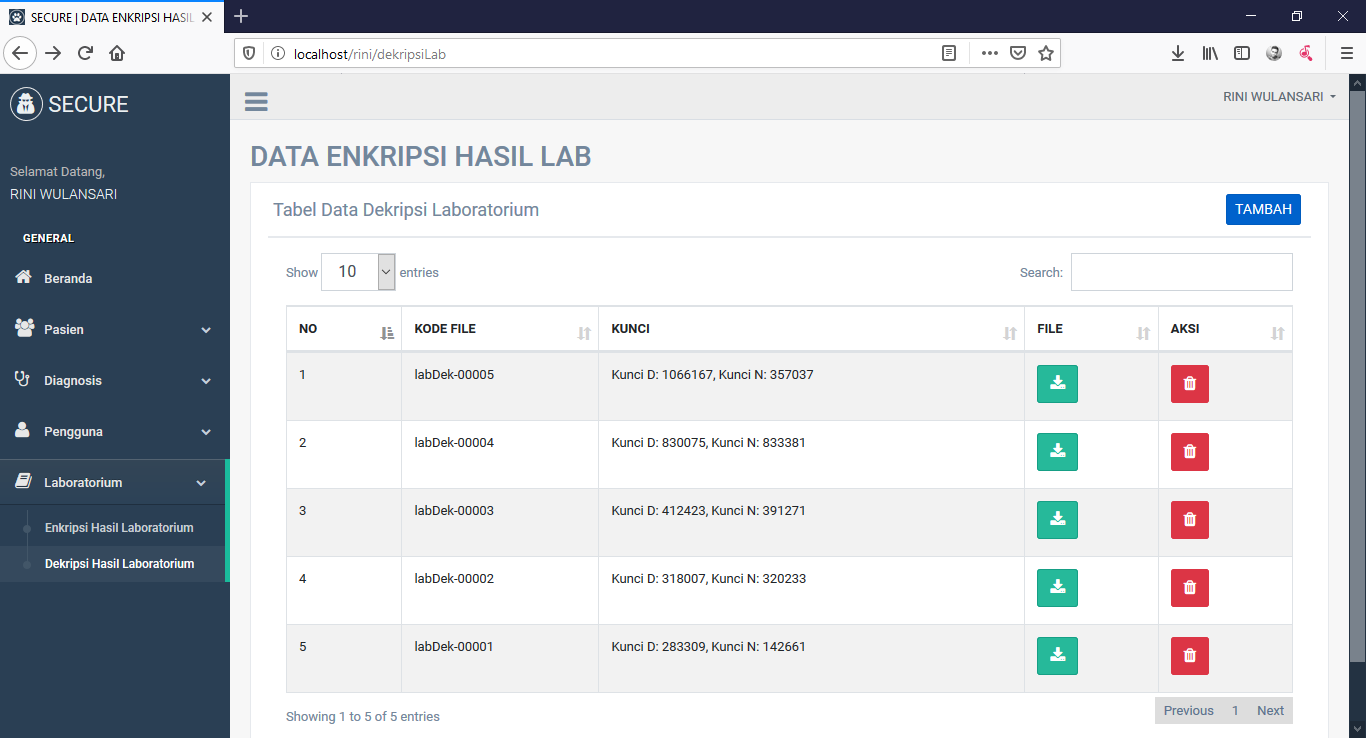
Tampilan data *form* enkripsi menampilkan data-data *file* dokumen yang telah terenkripsi dari dalam *database*. Data *file* dokumen tersebut ditampilkan agar penerima dapat dengan mudah mendapatkan *file* yangtelah dienkripsi oleh pengirim. Tampilan data *form* enkripsi dapat dilihat pada gambar 4.15.

.

Gambar 4.15. Tampilan *Form* Data enkripsi

1. Tampilan *Form* Data Dekripsi

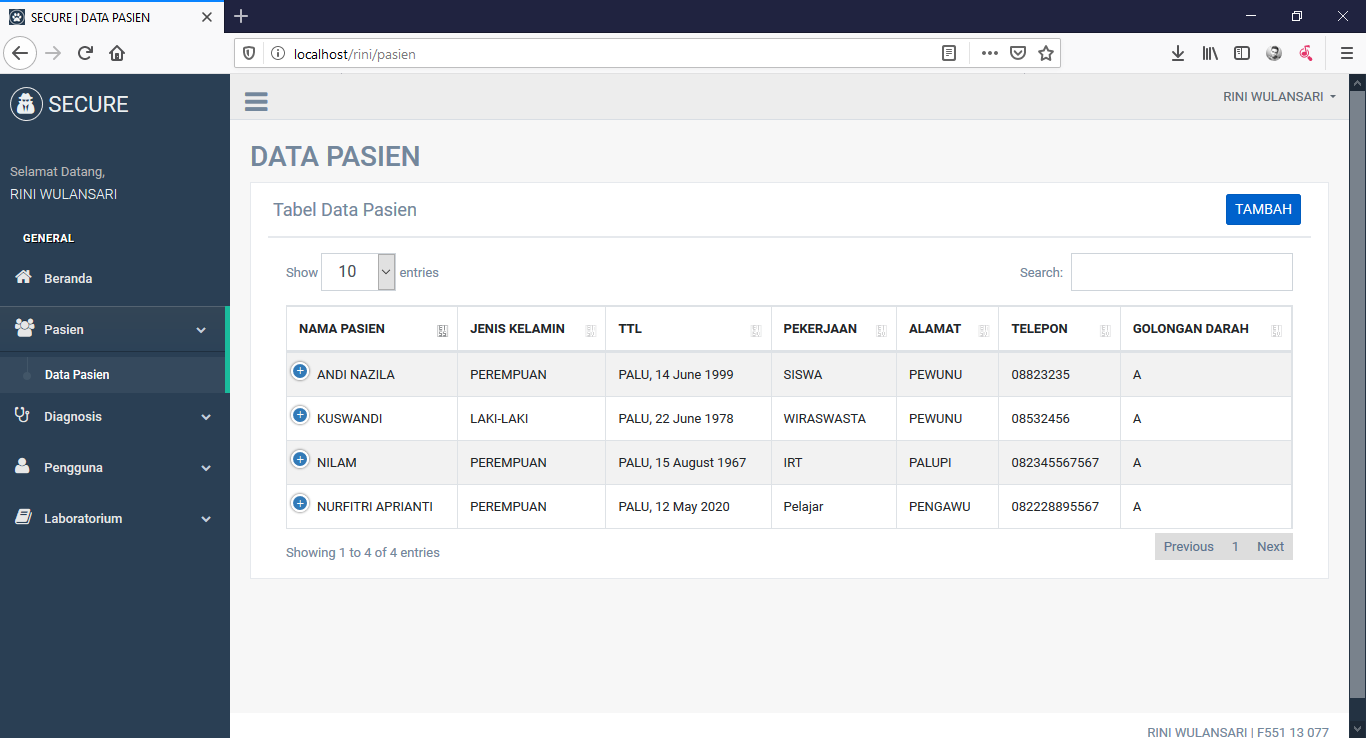
Tampilan data *form* dekripsi menampilkan data-data *file* dokumen yang telah terdekripsi dari dalam database. *File* ditampilkan agar penerima dapat mengunduh *file* hasil dekripsi. Tampilan data *form* dekripsi dapat dilihat pada gambar 4.16.



Gambar 4.16. Tampilan *Form* Data dekripsi

1. Tampilan *Form* Data Pasien

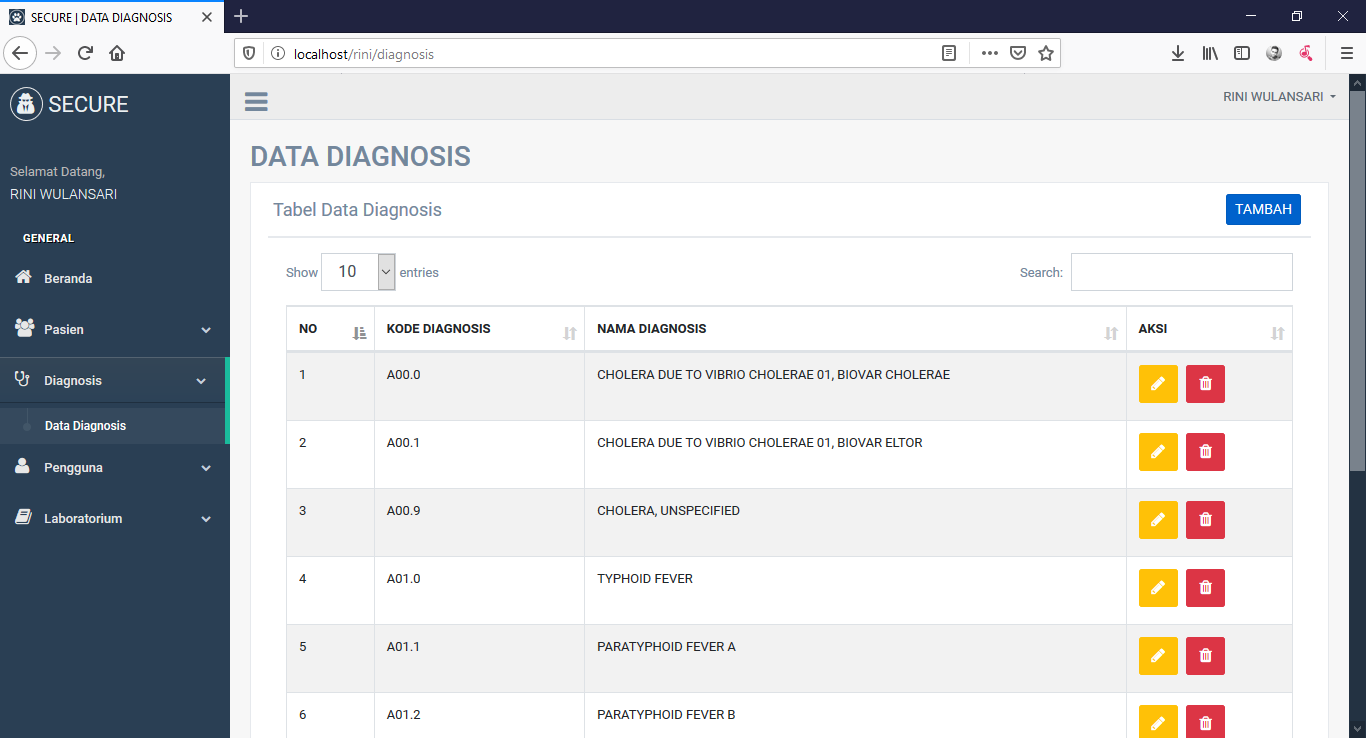
*Form* data pasien digunakan untuk menampilkan data-data pasien pada sistem keamanan data RSUD Anutapura Palu. *Form* data pasien dapat dilihat pada gambar 4.17.



Gambar 4.17. Tampilan *Form* data diagnosis

1. Tampilan *Form* Data Diagnosis

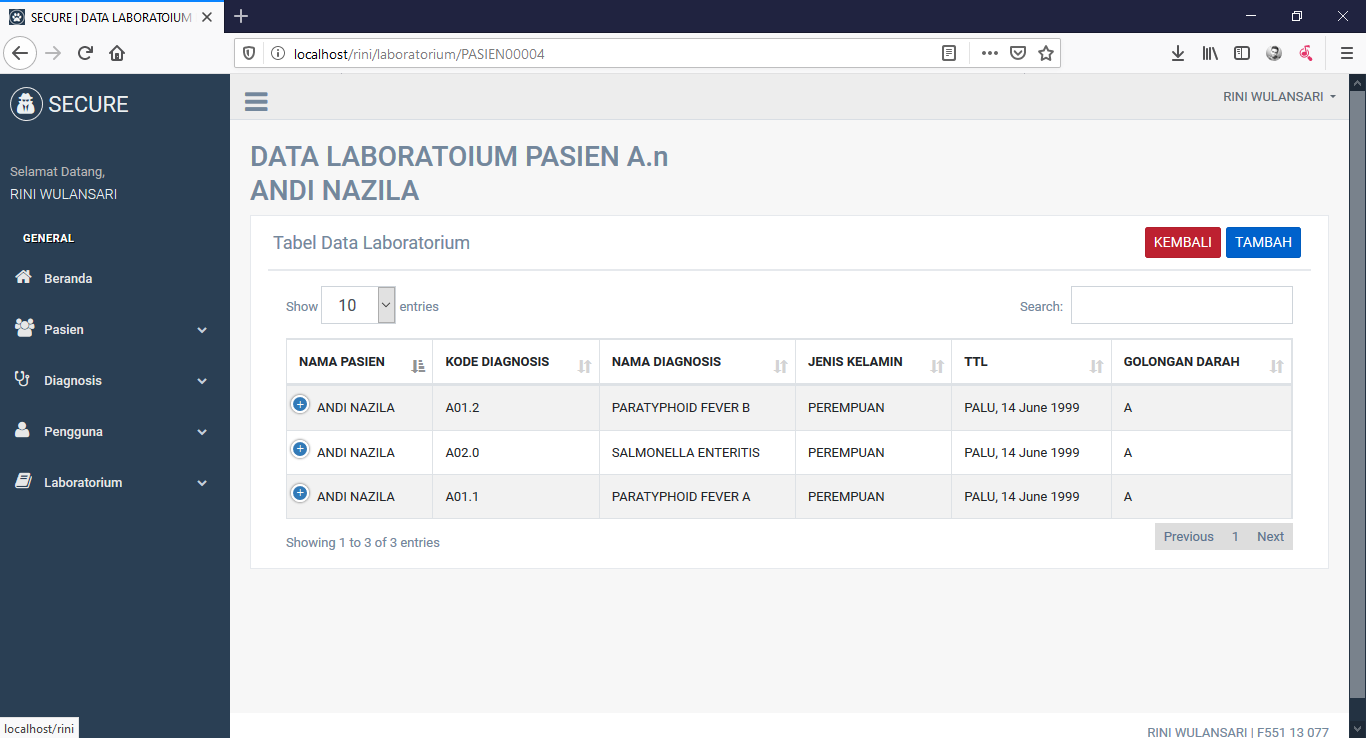
*Form* data diagnosis digunakan untuk menampilkan data-data diagnosis pasien pada sistem kemanan data RSUD Anutapura Palu. *Form* data diagnosis dapat dilihat pada gambar 4.18.



Gambar 4.18. Tampilan *Form* data diagnosis

1. Tampilan *Form* Data Laboratorium Pasien

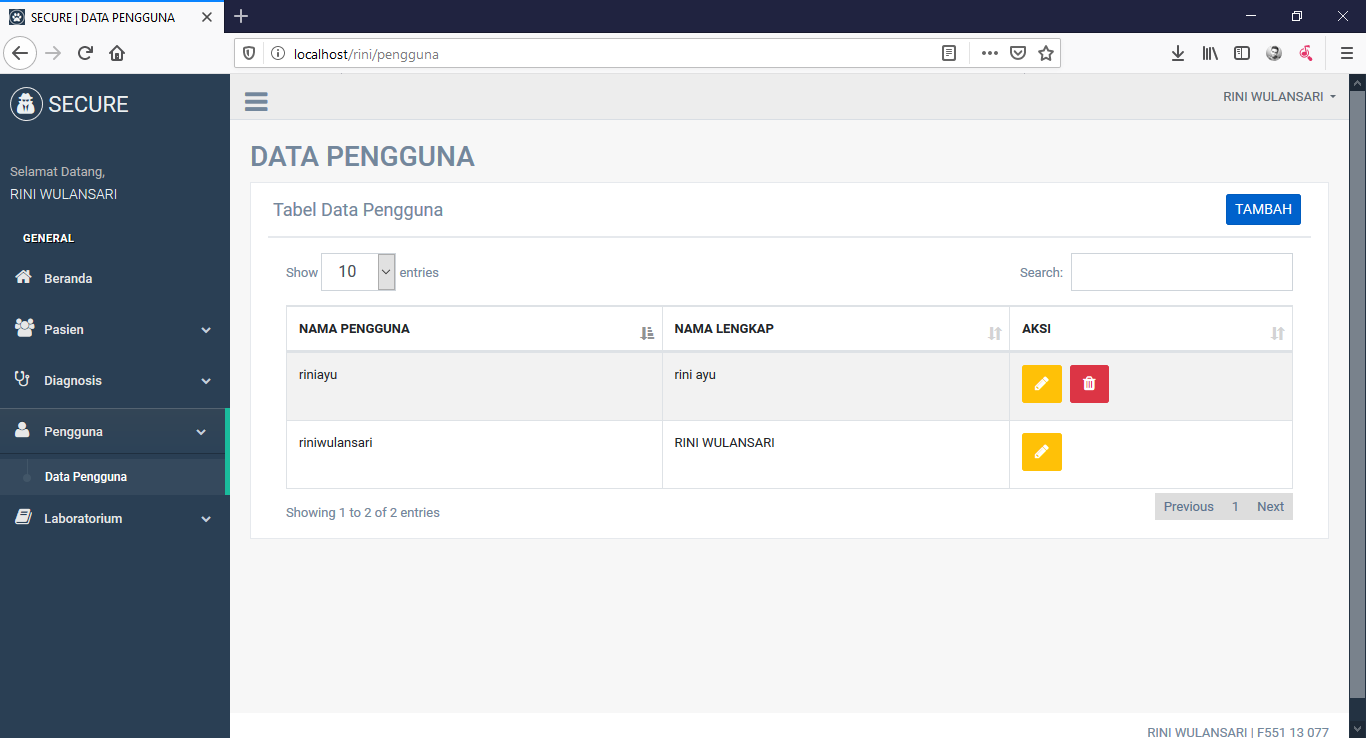
*Form* data lab pasien digunakan untuk menampilkan data-data lab pasien pada sistem kemanan data RSUD Anutapura Palu. *Form* data lab pasien dapat dilihat pada gambar 4.19.



Gambar 4.19. Tampilan *Form* data laboratorium pasien

1. Tampilan *Form* Data Pengguna

Tampilan data *form* pengguna menampilkan data-data akun dari pengguna sistem. Form ini hanya dapat diakses oleh pengguna yang berstatus sebagai pengirim. Tampilan data *form* pengguna dapat dilihat pada gambar 4.20.



Gambar 4.20. Tampilan *form* data pengguna

* + 1. **Pengujian Sistem**
       1. **Pengujian Fungsi Sistem**

Setelah pembuatan aplikasi telah selesai, maka dilakukan pengujian terhadap aplikasi. Pengujian sistem yang dilakukan merupakan pengujian terhadap aplikasi pengamanan data *file* dokumen. Pengujian fungsi sistem dilakukan dengan metode *blackbox* untuk mengetahui apakah fungsi-fungsi yang terdapat dalam sistem berjalan dengan baik atau tidak. Hasil pengujian fungsi sistem dan pengujian lapangan dapat dilihat pada tabel berikut:

1. Pengujian *Login*

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form login*. Pengujian *login* dapat dilihat pada tabel 4.6.

Tabel 4.6. Tabel pengujian *login*

|  |  |  |  |
| --- | --- | --- | --- |
| **Jenis Pengujian** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Nama pengguna | nama pengguna yang dimasukan benar | Sistem berjalan dengan baik dan berhasil menolak login yang salah | Baik |
| Kata sandi | Kata sandi yang dimasukan benar | Sistem berjalan dengan baik dan berhasil menolak login yang salah | Baik |
| Cek Login | Menampilkan keterangan login berhasil atau tidak | Sistem berjalan dengan baik dan berhasil menolak login yang salah | Baik |

1. Pengujian Enkripsi

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form* enkripsi. Pengujian enkripsi dapat dilihat pada tabel 4.7.

Tabel 4.7. Tabel pengujian enkripsi

|  |  |  |  |
| --- | --- | --- | --- |
| **Jenis Pengujian** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Edit text kunci e,d,n | Nilai kunci e,d,n berhasil ditampilkan | Sistem berjalan dengan baik | Baik |
| Temukan kunci | Untuk mendapatkan kunci y secara otomatis | Sistem berjalan dengan baik | Baik |
| Hasil kunci P | Nilai kunci P berhasil ditampilkan | Sistem berjalan dengan baik | Baik |
| Hasil kunci Q | Nilai kunci Q berhasil ditampilkan | Sistem berjalan dengan baik | Baik |
| Unggah file | Unggah file dapat ditampilkan | Sistem berjalan dengan baik | Baik |
| Batal | Membatalkan proses enkripsi file | Sistem berjalan dengan baik | Baik |
| Enkripsi | Melakukan proses enkripsi file | Sistem berjalan dengan baik | Baik |

1. Pengujian Dekripsi

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada form dekripsi. Pengujian dekripsi dapat dilihat pada tabel 4.8.

Tabel 4.8. Tabel pengujian dekripsi

|  |  |  |  |
| --- | --- | --- | --- |
| **Jenis Pengujian** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Input kunci rahasia D dan N | Nilai dari kunci D dan N berhasil ditampilkan | Sistem berjalan dengan baik | Baik |
| Unggah file Lampiran Hasil Lab | Berhasil mengunggah *file* | Sistem berjalan dengan baik | Baik |
| Dekripsi | *File* dokumen berhasil terdekripsi dan tesimpan | Sistem berjalan dengan baik | Baik |
| Batal | Membatalkan proses dekripsi file sistem | Sistem berjalan dengan baik | Baik |

1. Pengujian Buat Akun

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form* akun. Pengujian buat akundapat dilihat pada tabel 4.9.

Tabel 4.9. Tabel pengujian buat akun

|  |  |  |  |
| --- | --- | --- | --- |
| **Jenis Pengujian** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Menyimpan data akun | Dapat menyimpan nrp, nama pengguna, kata sandi, email dan status | Sistem berjalan dengan baik dan berhasil menyimpan data akun kedalam database | Baik |
| Menghapus data akun | Data yang ingin dihapus berhasil dihapus dari database sistem | Sistem berjalan dengan baik dan berhasil menghapus data akun | Baik |
| Mengubah data akun | Data yang ingin diubah berhasil diubah dan data perubahan disimpan kembali kedatabase sistem | Sistem berjalan dengan baik dan berhasil mengubah data akun | Baik |

1. Pengujian Tambah Data Pasien, Lab Pasien, Diagnosis Pasien

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form* data pasien, data lab pasien dan data diagnosis pasien. Pengujian dapat dilihat pada tabel 4.10.

Tabel 4.10. Tabel Pengujian Tambah Data Pasien, Lab Pasien, Diagnosis Pasien

|  |  |  |  |
| --- | --- | --- | --- |
| **Jenis Pengujian** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Menyimpan Data Pasien, Lab Pasien, Diagnosis Pasien | Dapat menyimpan data sesuai yang diinputkan | Sistem berjalan dengan baik dan berhasil menyimpan data kedalam database | Baik |
| Menghapus Data Pasien, Lab Pasien, Diagnosis Pasien | Data yang ingin dihapus berhasil dihapus dari database sistem | Sistem berjalan dengan baik dan berhasil menghapus data | Baik |
| Mengubah Data Pasien, Lab Pasien, Diagnosis Pasien | Data yang ingin diubah berhasil diubah dan data perubahan disimpan kembali kedatabase sistem | Sistem berjalan dengan baik dan berhasil mengubah data | Baik |

* + - 1. **Pengujian Enkripsi dan Dekripsi *File* Dokumen**

Pengujian enkripsi dan dekripsi *file* dokumen dilakukan untuk menguji akurasi, waktu dan keberhasilan algoritma dalam mengenkripsi dan dekripsi *file*. Pengujian dilakukan sebanyak beberapa kali dengan beberapa jenis *file* yang berbeda dan ukuran yang berbeda, pengujian yang dilakukan dapat dilihat pada tabel-tabel dibawah ini.

1. Pengujian *file* dokumen berekstensi .docx berukuran 719 kb

Pengujian dilakukan untuk mengetahui berapa lama sistem melakukan enkripsi dan dekripsi *file* dokumen berekstensi .docx berukuran 719 kb dengan menggunakan berbagai nilai kombinasi kunci publik dan kunci rahasia. Hasil pengujian dapat dilihat pada tabel 4.9.

Tabel 4.9. Tabel pengujian *file* dokumen berekstensi .docx berukuran 719 kb

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **no.** | **nama *file* enkripsi** | **jenis *file*** | **ukuran awal** | **ukuran akhir** | **kunci publik** | | | **kunci rahasia** | | **waktu enkripsi** | **waktu dekripsi** |
| **p** | **q** | **e** | **d** | **n** |
| 1. | labEnk-00001.docx | .docx | 719 kb | 4,716 kb | 557 | 587 | 535 | 325207 | 326959 | 25 detik | 40 detik |
| 2. | labEnk-00002.docx | .docx | 719 kb | 4,538 kb | 787 | 311 | 457 | 444133 | 244757 | 20 detik | 43 detik |
| 3. | labEnk-00003.docx | .docx | 719 kb | 4,591 kb | 271 | 983 | 343 | 264367 | 266393 | 21 detik | 41 detik |
| 4. | labEnk-00004.docx | .docx | 719 kb | 4,681 kb | 521 | 821 | 267 | 851203 | 427741 | 19 detik | 46 detik |
| 5. | labEnk-00005.docx | .docx | 719 kb | 4,142 kb | 293 | 347 | 357 | 201781 | 101671 | 19 detik | 36 detik |
| 6. | labEnk-00006.docx | .docx | 719 kb | 4,563 kb | 449 | 431 | 471 | 384871 | 193519 | 22 detik | 46 detik |
| 7. | labEnk-00007.docx | .docx | 719 kb | 4,615 kb | 419 | 647 | 657 | 269617 | 271093 | 20 detik | 40 detik |
| 8. | labEnk-00008.docx | .docx | 719 kb | 4,723 kb | 521 | 743 | 971 | 390211 | 387103 | 24 detik | 43 detik |
| 9. | labEnk-00009.docx | .docx | 719 kb | 4,383 kb | 541 | 313 | 331 | 504931 | 169333 | 19 detik | 43 detik |
| 10. | labEnk-00010.docx | .docx | 719 kb | 4,558 kb | 277 | 839 | 259 | 461683 | 232404 | 18 detik | 44 detik |
| **Rata-rata waktu enkripsi dan dekripsi** | | | | | | | | | | 20 detik | 50 detik |

Hasil pengujian pada tabel 4.9 menunjukan pada percobaan 1, 2, dan 3 waktu dekripsi lebih cepat dari waktu enkripsi namun pada percobaan 4 dan 5 waktu dekripsi lebih lama dari waktu enkripsi. Dan pada percobaan 4 dan 5 dilakukan juga pengujian pada *file* yang sama dengan menggunakan kunci publik dan kunci rahasia yang sama terlihat terdapat perbedaan waktu enkripsi. Waktu rata-rata yang dihasilkan dari proses enkripsi adalah 89 detik atau 1 menit 29 detik dan waktu rata-rata yang dihasilkan dari proses dekripsi adalah 83 detik atau 1 menit 23 detik.

1. Pengujian *file* dokumen berekstensi .docx berukuran 1,129 kb

Pengujian dilakukan untuk mengetahui berapa lama sistem melakukan enkripsi dan dekripsi *file* dokumen berekstensi .docx berukuran 1,129 kb dengan menggunakan berbagai nilai kombinasi kunci publik dan kunci rahasia. Hasil pengujian dapat dilihat pada tabel 4.10.

Tabel 4.10. Tabel pengujian *file* dokumen berekstensi .docx berukuran 1,129 kb

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **no.** | **nama *file* enkripsi** | **jenis *file*** | **ukuran awal** | **ukuran akhir** | **kunci publik** | | | **kunci rahasia** | | **waktu enkripsi** | **waktu dekripsi** |
| **p** | **q** | **e** | **d** | **n** |
| 1. | labEnk-00008.docx | .docx |  |  |  |  |  |  |  |  |  |
| 2. | labEnk-00007.docx | .docx |  |  |  |  |  |  |  |  |  |
| 3. | labEnk-00003.docx | .docx |  |  |  |  |  |  |  |  |  |
| 4. | labEnk-00004.docx | .docx |  |  |  |  |  |  |  |  |  |
| 5. | labEnk-00005.docx | .docx |  |  |  |  |  |  |  |  |  |
| **Rata-rata waktu enkripsi dan dekripsi** | | | | | | | | | | 179 detik | 133 detik |

Hasil pengujian pada tabel 4.10 menunjukan pada percobaan 2 yang menggunakan kombinasi kunci yang lebih kecil dari percobaan 3 mendapatkan waktu enkripsi lebih lama dengan waktu dekripsi relatif sama. Kemudian pada percobaan 4 terlihat waktu dekripsi memerlukan waktu yang lebih lama dari waktu enkripsi. Dan pengujian menunjukan peningkatan rata-rata waktu enkripsi dan dekripsi dari pengujian pada tabel 4.9. Waktu rata-rata yang dihasilkan dari proses enkripsi pada tabel 4.10 adalah 179 detik atau 2 menit 59 detik dan Waktu rata-rata yang dihasilkan dari proses dekripsi adalah 133 detik atau 2 menit 13 detik.

1. Pengujian *file* dokumen berekstensi .docx berukuran 2,163 kb

Pengujian dilakukan untuk mengetahui berapa lama sistem melakukan enkripsi dan dekripsi *file* dokumen berekstensi .docx berukuran 2,163 kb dengan menggunakan berbagai nilai kombinasi kunci publik dan kunci rahasia. Hasil pengujian dapat dilihat pada tabel 4.11.

Tabel 4.11. Tabel pengujian *file*dokumen berekstensi .docx berukuran 2,163 kb

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **no.** | **nama *file* enkripsi** | **jenis *file*** | **ukuran awal** | **ukuran akhir** | **kunci publik** | | | **kunci rahasia** | | **waktu enkripsi** | **waktu dekripsi** |
| **p** | **q** | **e** | **d** | **n** |
| 1. | enk-2019-10-26-8982.docx | .docx |  |  |  |  |  |  |  |  |  |
| 2. | enk-2019-10-27-3123.docx | .docx |  |  |  |  |  |  |  |  |  |
| 3. | enk-2019-10-27-4981.docx | .docx |  |  |  |  |  |  |  |  |  |
| 4. | enk-2019-10-27-6490.docx | .docx |  |  |  |  |  |  |  |  |  |
| 5. | enk-2019-10-27-2998.docx | .docx |  |  |  |  |  |  |  |  |  |
| **Rata-rata waktu enkripsi dan dekripsi** | | | | | | | | | | 213 detik | 241 detik |

Pada tabel 4.11percobaan 1 dan 5 dilakukan pengujian menggunakan nilai kunci p yang sama dengan nilai kunci e, q dan x yang berbeda mendapatkan hasil waktu enkripsi dari percobaan 1 lebih cepat dari waktu enkripsi percobaan 5, namun waktu dekripsi pada percobaan 5 dapat lebih cepat dari waktu dekripsi percobaan 1. Waktu rata-rata yang dihasilkan dari proses enkripsi pada tabel 4.11 adalah 213 detik atau 3 menit 33 detik dan Waktu rata-rata yang dihasilkan dari proses dekripsi adalah 241 detik atau 4 menit 1 detik.

1. Pengujian *file* dokumen berekstensi .docx berukuran 3,303 kb

Pengujian dilakukan untuk mengetahui berapa lama sistem melakukan enkripsi dan dekripsi *file* dokumen berekstensi .docx berukuran 3,303 kb dengan menggunakan berbagai nilai kombinasi kunci publik dan kunci rahasia. Hasil pengujian dapat dilihat pada tabel 4.12.

Tabel 4.12. Tabel pengujian *file* dokumen berekstensi .docx berukuran 3,303 kb

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **no.** | **nama *file* enkripsi** | **jenis *file*** | **ukuran awal** | **ukuran akhir** | **kunci publik** | | | **kunci rahasia** | | **waktu enkripsi** | **waktu dekripsi** |
| **p** | **q** | **e** | **d** | **n** |
| 1. | enk-2019-10-28-6814.docx | .docx |  |  |  |  |  |  |  |  |  |
| 2. | enk-2019-10-28-2591.docx | .docx |  |  |  |  |  |  |  |  |  |
| 3. | enk-2019-10-28-4551.docx | .docx |  |  |  |  |  |  |  |  |  |
| 4. | enk-2019-10-28-1345.docx | .docx |  |  |  |  |  |  |  |  |  |
| 5. | enk-2019-10-28-4803.docx | .docx |  |  |  |  |  |  |  |  |  |
| **Rata-rata waktu enkripsi dan dekripsi** | | | | | | | | | | 513 detik | 352 detik |

Pada tabel 4.12 percobaan 1 dan 2 terlihat perbedaan waktu dekripsi, pada percobaan 1 yang memiliki nilai kunci e lebih kecil dari nilai kunci p menunjukan waktu dekripsi yang lebih lama dari waktu dekripsi pada percobaan 2. Penulis juga melakukan percobaan yang sama namun dengan nilai kunci p yang lebih kecil seperti pada percobaan 3 dan percobaan 4 dan menunjukan hasil yang sama. Waktu rata-rata yang dihasilkan dari proses enkripsi pada tabel 4.12 adalah 513 detik atau 8 menit 27 detik dan Waktu rata-rata yang dihasilkan dari proses dekripsi adalah 352 detik atau 5 menit 52 detik.

1. Pengujian *file* dokumen berekstensi .docx berukuran 4,620 kb

Pengujian dilakukan untuk mengetahui berapa lama sistem melakukan enkripsi dan dekripsi *file* dokumen berekstensi .docx berukuran 4,620 kb dengan menggunakan berbagai nilai kombinasi kunci publik dan kunci rahasia. Hasil pengujian dapat dilihat pada tabel 4.13.

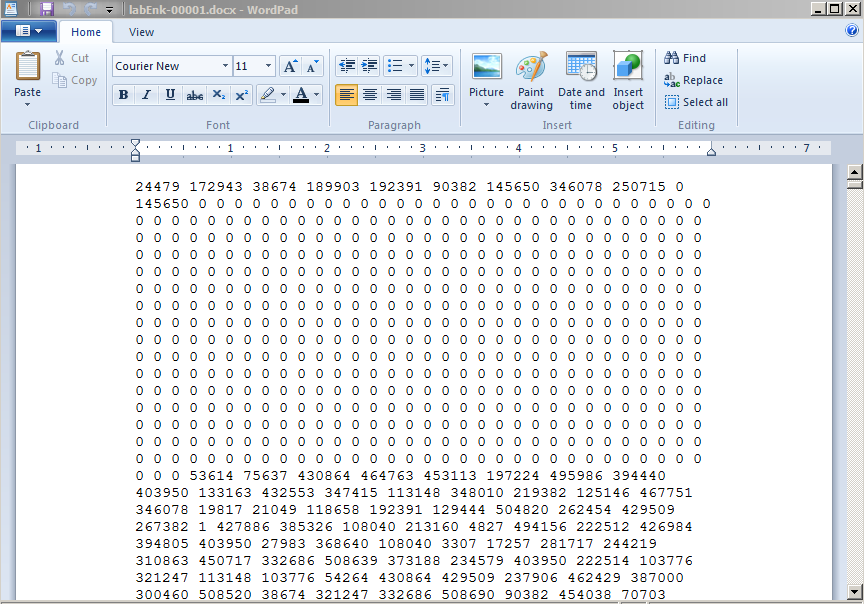
Tabel 4.13. Tabel pengujian *file* dokumen berekstensi .docx berukuran 4,620 kb

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **no.** | **nama *file* enkripsi** | **jenis *file*** | **ukuran awal** | **ukuran akhir** | **kunci publik** | | | **kunci rahasia** | | **waktu enkripsi** | **waktu dekripsi** |
| **p** | **q** | **e** | **d** | **n** |
| 1. | enk-2019-10-28-7140.docx | .docx |  |  |  |  |  |  |  |  |  |
| 2. | enk-2019-10-28-5044.docx | .docx |  |  |  |  |  |  |  |  |  |
| 3. | enk-2019-10-28-5114.docx | .docx |  |  |  |  |  |  |  |  |  |
| 4. | enk-2019-10-28-8077.docx | .docx |  |  |  |  |  |  |  |  |  |
| 5. | enk-2019-10-28-6535.docx | .docx |  |  |  |  |  |  |  |  |  |
| **Rata-rata waktu enkripsi dan dekripsi** | | | | | | | | | | 596 detik | 197 detik |

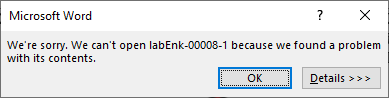
Pada tabel 4.13 penulis melakukan pengujian enkripsi dan dekripsi *file*dengan menggunakan kunci publik dan kunci rahasia dengan nilai yang samaseperti pada percobaan 1, 2 dan 3, hasil pengujian menunjukan waktu enkripsi yangmasih berbeda-beda sedangkan waktu dekripsiyang dihasilkan tidak berbeda jauh. Waktu rata-rata yang dihasilkan dari proses enkripsi pada tabel 4.13 adalah 596 detik atau 9 menit 56 detik dan Waktu rata-rata yang dihasilkan dari proses dekripsi adalah 197 detik atau 3 menit 17 detik.

* + - 1. **Hasil Pengujian Algoritma *RSA***

Algoritma *RSA* ini bisa diterapkan pada sebuah sistem pengamanan data *file* dokumen. Algoritma *RSA* dapat melakukan enkripsi semua *file* yang terdapat pada tabel pengujian. Sebagai contoh sistem ini bekerja dapat dilihat pada gambar 4.21 dan gambar 4.22.

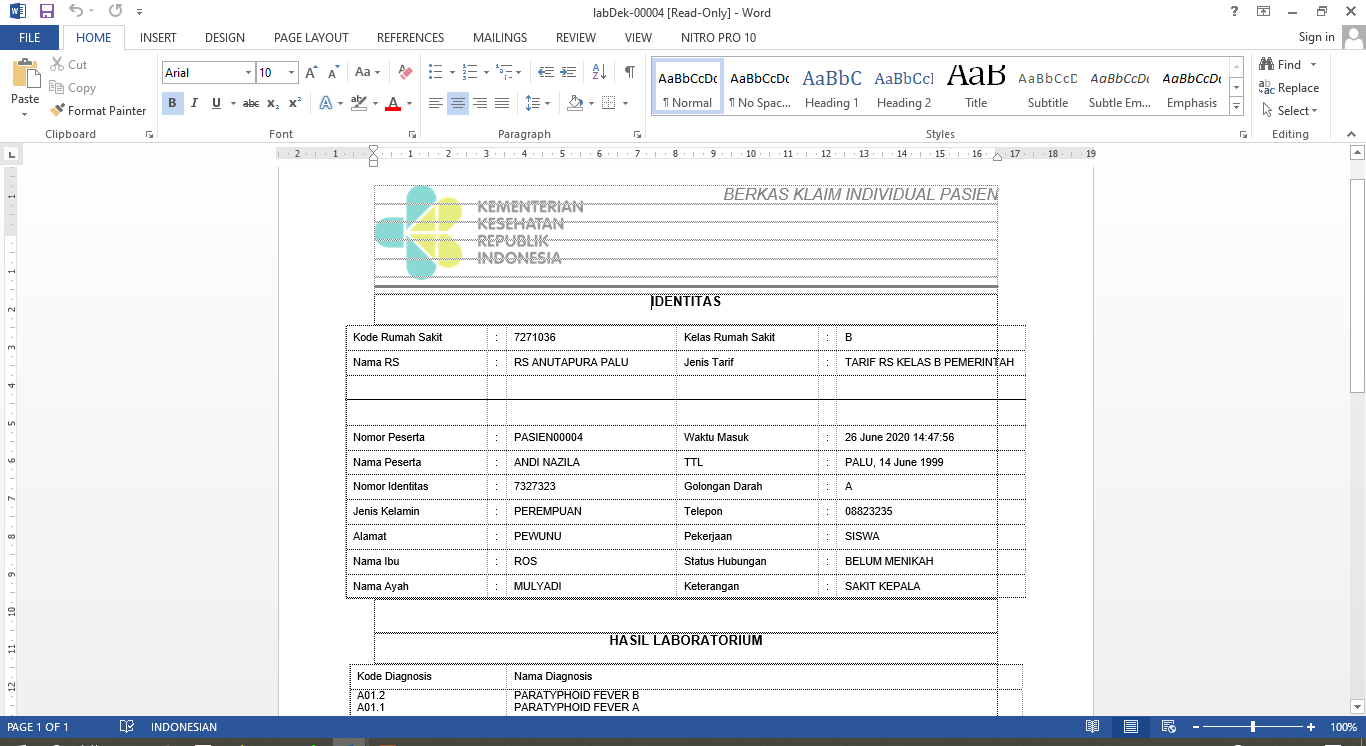


Gambar 4.30. Hasil enkripsi atau chiphertext dari *file* berekstensi .docx yang dibuka dengan menggunakan aplikasi *wordpad.*



Gambar 4.22. Hasil enkripsi *file* berekstensi .docx

Algoritma *RSA* dapat melakukan dekripsi semua *file* yang terdapat pada tabel pengujian. Sebagai contoh sistem ini bekerja dapat dilihat pada gambar 4.23



Gambar 4.23. Hasil dekripsi *file* berekstensi .docx

Untuk mengetahui perhitungan algoritma *RSA* yang diimplementasikan ke sistem berhasil dan berjalan dengan baik, penulis melakukan pengujian dengan perhitungan manual, pengujian dilakukan dengan menggunakan 5 karakter pertama dari *plaintext*. Hasil enkripsi perhitungan manual dari 5 karakter tersebut nantinya akan dibandingkan dengan hasil enkripsi yang dihasilkan sistem seperti pada gambar 4.26. Berikut adalah perhitungan manual algoritma *RSA*.

Contoh Kasus :

*Plaintext :* TEKNIK

**Pembentukan Kunci**

Langkah-langkah pada proses pembangkitan kunci publik dan kunci privat pada algoritma RSA adalah sebagai berikut:

1. Langkah Pertama: Pilih 2 bilangan prima yang berbeda secara acak untuk nilai p dan q. Dengan syarat nilai p ≠ q. Semakin besar nilainya semakin baik. Keduanya bilangan prima ini, p dan q bersifat rahasia.
2. Langkah Kedua: Hitung nilai n menggunakan persamaan 1.

n = p \* q

dimana n digunakan untuk modulos pada kunci public dan kunci privat. Nilai n tidak rahasia.

1. Langkah Ketiga: Hitung nilai Φ(n) menggunakan persamaan 2.

Φ(n) = (p - 1) \* (q – 1)

Digunakan untuk pencarian kunci privat. Nilai Φ(n) bersifat rahasia.

1. Langkah Keempat: Hitung nilai e dengan cara memilih bilangan bulat dengan syarat e > 1, dan GCD(e, φ(n)) = 1. Nilai e bersifat tidak rahasia.
2. Langkah Kelima: Pilh nilai d yang merupakan bilangan bulat dengan syarat nilai d memenuhi (d \* e) mod φ(n) = 1 atau d = (1 + k \* φ(n)) / e, nilai k dapat ditentukan secara acak. Nilai d bersifat rahasia.

Sehingga dari langkah-langkah pada proses pembangkitan kunci di atas didapatkan kunci public (e, n) dan kunci privat (d, n). Contoh proses pembangkitan kunci public dan kunci privat pada algoritma RSA:

1. Nilai bilangan prima p = 53 dan q = 79.
2. Hitung nilai n = p \* q

n = 53 \* 79 = 4187

1. Hitung Φ(n) = (p - 1) \* (q – 1)

Φ(n) = (53 – 1) \* (79 – 1)

= 52 \* 78 = 4056

1. Hitung nilai e

sebagai contoh, nilai e = 73 dan φ = 4056. Sesuai persyaratan, tes terlebih dahulu apakah GCD(73,4056) = 1 ?

4056 mod 73 = 41

73 mod 41 = 32

41 mod 32 = 9

32 mod 9 = 5

9 mod 5 = 4

5 mod 4 = **1**

Ternyata benar GCD (73,4056) = 1 (1 didapat dari angka yang di *bold*).

Untuk membangkitkan kunci *public* dan kunci privat maka menentukan nilai p dan q yaitu:

Nilai p: 53

Nilai q: 79

1. Langkah Kedua: Hitung nilai n.

Untuk menghitung nilai n digunakan persamaan 1.

*n* = *p \* q*

Dimana, n = kunci *public*

p = nilai p

q = nilai q

menentukan nilai pembangkit kunci *public*

Untuk n = p \* q

= 53 \* 79

= 4187

1. Langkah Ketiga: Hitung φ (di baca phi). Φ(n) = (p - 1) \* (q - 1).

Menentukan totient dari kunci public (n) digunakan persamaan 2.

Φ(n) = (p - 1) \* (q – 1) (2)

Dimana, Φ = totient dari kunci public

p = nilai p

q = nilai q

untuk Φ (n)= (p - 1) \* (q - 1)

= (53-1) \* (79-1)

= 52 \* 78 = 4056

1. Langkah Keempat: Pilih nilai kunci enkripai (e) dengan syarat e > 1, dan GCD(e, φ(n)) = 1

sebagai contoh, nilai e = 73 dan φ = 4056. Sesuai persyaratan, tes terlebih dahulu apakah GCD(73,4056) = 1 ?

4056 mod 73 = 41

73 mod 41 = 32

41 mod 32 = 9

32 mod 9 = 5

9 mod 5 = 4

5 mod 4 = **1**

Ternyata benar GCD(73,4056) = 1 (1 didapat dari angka yang di *bold*).

1. Langkah Kelima: tentukan nilai d, dimana k ditentukan secara acak/sembarang dan perhitungan nilai d menggunakan persamaan sebagai berikut:

Dimana, d = kunci privat

k = bilangan bulat

Φ = nilai *totient*

e = kunci *public*

menentukan nilai dari kunci privat

untuk

Dengan mencoba nilai k = 16 diperoleh nilai *d* yang bulat adalah 889. Ini adalah kunci private untuk mendeskripsi pesan.

Kunci *Public* :

Kunci *Private* :

**Proses Enkripsi**

Proses enkripsi menggunakan algoritma RSA terhadap *plainteks* m = TEKNIK, pertama-tama *plaintext* diubah menjadi format ASCII (*Decimal*) sebagai berikut:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TEXT (Karakter) | T | E | K | N | I | K |
| ASCII (Decimal) | 84 | 69 | 75 | 78 | 73 | 75 |

m = TEKNIK = 846975787375, kemudian pecahkan m dengan membuat blok yang masing-masing blok tersebut terdapat 3 digit.

|  |  |  |  |
| --- | --- | --- | --- |
| 846 | 975 | 787 | 375 |

m1 = 846 m3 = 787

m2 = 975 m4 = 375

misalkan e = 73 dan n = 4187, lalu enkripsikan setiap blok *plaintext* menggunakan persamaan sebagai berikut:

C = Memod n

keterangan:

C = *Ciphertext*

M = *Plaintext*

e = Kunci *public*

n = pembangkit kunci *public*

menghitung nilai *ciphertext* dari m1, m2, m3, dan m4 yang telah didapatkan sebelumnya.

c1 = 846^73 mod 4187 = 3927

c2 = 975^73 mod 4187 = 2701

c3 = 787^73 mod 4187 = 3252

c4 = 375^73 mod 4187 = 714

Jadi *ciphertext* (c) adalah = 3927 2701 3252 714

**Proses Dekripsi**

Proses dekripsi algoritma RSA menggunakan persamaan sebagai berikut:

Keterangan:

P = Cd mod n

P = *Plaintext*

C = *Ciphertext*

d = Kunci Privat

n = Pembangkit Kunci *public*

Sebelumnya telah didapatkan nila d = 889 dan n = 4187, setiap blok*ciphertext* di dekripsikan sebagai berikut:

m1 = 3927^889 mod 4187 = 846

m2 = 2701^889 mod 4187 = 975

m3 = 3252^889 mod 4187 = 787

m4 = 714^889 mod 4187 = 375

Jadi dapat diperoleh plainteks (m) = 846975787375, dimana ketika dikonversi kedalam ASCII m = TEKNIK. Diatas adalah contoh sederhana pengamanan data dengan algoritma RSA.

Dari hasil perhitungan manual dapat dibandingkan dengan peritungan yang dihasilkan sistem. Dapat dilihat bahwa hasil enkripsi dan dekripsi dari perhitungan manual dan hasil enkripsi dan dekripsi yang dihasilkan sistem terlihat sama, jadi dapat disimpulkan perhitungan algoritma *RSA* yang penulis implementasikan pada sistem pengamanan data *file* dokumen berhasil dan berjalan dengan baik.

* 1. **Pembahasan**

Berdasarkan hasil pengujian enkripsi dan dekripsi *file* dokumen, algoritma *RSA* yang diimplementasikan pada sistem menunjukan keberhasilan enkripsi dan dekripsi pada *file* dokumen berekstensi .docx dengan presentase keberhasilan 100%. Pada pengujian algoritma *RSA* penulis melakukan proses perhitungan manual yang bermaksud untuk mencocokan hasil perhitungan yang dilakukan oleh sistem, dan hasil yang didapatkan perhitungan manual maupun perhitungan sistem menghasilkan hasil enkripsi dan dekripsi yang sama. Algoritma *RSA* merupakan algoritma kriptografi asimetris yang menggunakan dua jenis kunci, yaitu kunci publik untuk melakukan enkripsi dan kunci rahasia untuk melakukan dekripsi. Isi dari *file* dokumen yang dilakukan pengujian tidak hanya *file* dokumen yang berisikan data teks, dilakukan juga pengujian pada *file* dokumen yang berisikan data gambar dan *file* dokumen yang terkunci. Pengujian dilakukan pada berbagai macam besaran *file* dari yang terkecil berukuran 144 kb dan yang terbesar berukuran 4,620 kb. Dari beberapa kali pengujian diatas didapatkan waktu enkripsi tercepat yang dihasilkan sistem adalah 4 detik dan yang terlama 1,372 detik, kemudian waktu dekripsi tercepat yang dihasilkan sistem adalah 5 detik dengan waktu dekripsi yang terlama 1,098 detik.

Waktu yang diperlukan oleh sistem dalam proses melakukan enkripsi dan dekripsi sangat dinamis dikarenakan beberapa faktor, antara lain nilai p dan q yang didapatkan dari hasil *random* sistem dapat mempengaruhi waktu enkripsi dan kombinasi nilai kunci p dan nilai kunci e yang digunakan untuk melakukan dekripsi dapat mempengaruhi lama waktu proses dekripsi, dimana jika nilai *random* p dan q mendapatkan angka yang besar akan menghasilkan waktu enkripsi yang lama dibandingkan dengan jika nilai *random* p dan q yang dihasilkan sistem mendapatkan angka yang kecil, dalam hal ini batas nilai maksimal dari nilai *random* p dan q adalah 3 digit, untuk mencegah waktu proses enkripsi yang mungkin akan semakin lama maka nilai kunci p yang dihasilkan sistem dibatasi dengan maksimal nilai pada angka 1000. Dan penentuan nilai kunci x yang jauh lebih kecil dari nilai kunci p akan menghasilkan waktu dekripsi yang lama dibandingkan dengan jika nilai kunci x tidak jauh berbeda dengan nilai dari kunci p. Selain nilai *random* k, ukuran *file* juga dapat mempengaruhi lama waktu enkripsi dan dekripsi.

Ukuran *file* setelah dienkripsi akan mengalami peningkatan dari ukuran awal, hal ini dipengaruhi langsung oleh perhitungan dari algoritma *RSA* itu sendiri, dimana untuk setiap satu karakter pada *plaintext* akan menjadi nilai a dan b pada *chipertext*, sehingga *chipertext* yang dihasilkan akan menjadi berkali lipat, ukuran *file* akan kembali seperti pada ukuran awal ketika *file* didekripsi. Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini r = p × q. Semakin tinggi angka yang digunakan maka akan semakin sulit pula pesan/sandi dapat ditebak oleh pihak ketiga.

Algoritma *RSA* dapat diimplementasikan pada sistem ini, sekaligus dapat menjawab rumusan masalah yang penulis buat dan menjawab hipotesis penelitian yang dimana pada sistem ini menggunakan algoritma *RSA* berhasil mengamankan *file* dokumen data rekam medis pasien pada Rumah Sakit Umum Daerah (RSUD) Anutapura Kota Palu, dengan hasil-hasil yang telah penulis jabarkan sebelumnya.