Journal of Computation Science And Artificial Intelligence

Journal homepage:
https://jcsai.xjurnal.com/index.php/journal/index

e-ISSN: 3032-4653

Vol. 3, No. 1, 2026

METODE ALGORITMA RC4 *(RIVEST CODE 4)* UNTUK PENGAMANAN DATABASE TRANSAKSI PADA *GLORY* DIGITAL SABLON

Wahyu Ariandi[1*], Faisal Akbar[2], M. Rezza Fahlevi[3], Adiguna Ahlul Bai'at[4]

[1,2,3,4]Sekolah Tinggi Ilmu Komputer (STIKOM) PolTek Cirebon
Email: [1*]wahyuariandi@mail.ugm.ac.id [2]faisal.akbar@stikompoltekcirebon.ac.id, [3]m.rezza@mail.ugm.ac.id, [4]adigunaahlulbaiat@gmail.com

Kemajuan teknologi informasi dan komunikasi memfasilitasi penyelesaian pekerjaan dengan lebih cepat, mudah, dan akurat. Glory Digital Sablon (GDS) adalah toko yang menyediakan berbagai kebutuhan terkait sablon digital, namun sistem penjualannya masih manual, sementara pembuatan nota menggunakan Microsoft Excel yang memiliki kekurangan dan kendala. Penelitian ini bertujuan untuk merancang sistem yang otomatis, serta memperkenalkan algoritma kriptografi RC4 untuk mengamankan database transaksi dan database supplier agar tidak disalahgunakan. Penerapan algoritma RC4 terbukti efektif dalam mengamankan data transaksi pada GDS, dengan kunci enkripsi yang dapat diacak berdasarkan panjang kunci, dan dapat mengembalikan data asli melalui dekripsi dengan kunci yang sama. Pengujian CrackStation menunjukkan bahwa cipherteks tidak dapat dipecahkan, dan uji performa menunjukkan bahwa meskipun panjang kunci memengaruhi kecepatan enkripsi dan dekripsi, perbedaannya sangat kecil, sehingga tidak mempengaruhi kinerja keseluruhan sistem.
.

Kata Kunci : Keamanan Data, Performa Enkripsi, Implementasi Kriptografi, *Rivest Code* 4 (RC4)

RC4 ALGORITHM METHOD (RIVEST CODE 4) FOR SECURING TRANSACTION DATABASES ON GLORY DIGITAL SABLON

Abstract

Advances in information and communication technology facilitate faster, easier, and more accurate work completion. Glory Digital Sablon (GDS) is a store that provides various needs related to digital screen printing, but its sales system is still manual, while the creation of invoices uses Microsoft Excel which has shortcomings and obstacles. This study aims to design an automated system, as well as introduce the RC4 cryptographic algorithm to secure the transaction database and supplier database from misuse. The implementation of the RC4 algorithm has proven effective in securing transaction data on GDS, with encryption keys that can be randomized based on key length, and can restore the original data through decryption with the same key. CrackStation testing shows that the ciphertext cannot be cracked, and performance tests show that although key length affects encryption and decryption speed, the difference is very small, so it does not affect the overall performance of the system.

Keywords *: Data Security, Encryption Performance, Cryptography Implementation, Rivest Code 4 (RC4)*

DOI: https://doi.org/10.58468/jcsai.v3i1.27

Journal of Computation Science And Artificial Intelligence

Journal homepage:
https://jcsai.xjurnal.com/index.php/journal/index

e-ISSN: 3032-4653

Vol. 3, No. 1, 2026

1. PENDAHULUAN

Kemajuan teknologi akan berjalan sesuai dengan kemajuan ilmu pengetahuan yang dimana akan membantu dalam menyelesaikan pekerjaan dengan mudah, cepat dan akurat. Hal ini didukung dengan teknologi informasi dan komunikasi yang berkembang pesat dari tahun ke tahun. Serta sumber daya manusia merupakan faktor yang mendukung kemajuan sebuah teknologi [1].

Teknologi informasi telah menjadi bagian integral dari kehidupan sehari-hari, mempengaruhi segala aspek kehidupan, dari yang sederhana hingga yang kompleks. Meskipun memberikan banyak manfaat, kemajuan ini juga membawa tantangan baru, terutama dalam hal keamanan data, yang menjadi lebih rentan terhadap ancaman seperti pencurian dan manipulasi. Di Glory Digital Sablon (GDS), sistem transaksi dan pengelolaan data masih dilakukan secara manual, yang berisiko mengancam kerahasiaan informasi. Oleh karena itu, diperlukan sistem yang lebih aman untuk mengelola dan mengamankan data transaksi dan database supplier. Dalam penelitian ini, kami menerapkan algoritma kriptografi RC4 untuk meningkatkan keamanan data GDS [2].

Masalah *security* dan *privacy* adalah aspek yang sangat penting. Penyadapan informasi serta manipulasi data merupakan hal yang sangat merugikan bagi pengguna saat ini. Dengan adanya kemungkinan penyadapan dan manipulasi data tersebut, maka keamanan data menjadi sangat penting. [3].

Glory Digital Sablon (GDS) merupakan toko yang menyediakan berbagai macam kebutuhan mulai dari mesin dan bahan baku yang berhubungan dengan sablon digital. Sistem penjualan yang ada di GDS masih dilakukan secara manual sedangkan pembuatan nota sendiri sudah menggunakan komputer melalui *microsoft excel*, sehingga masih memiliki berbagai kekurangan dan kendala yang dihadapi. Maka perlu dibuat sebuah sistem yang bisa menangani secara otomatis, serta belum adanya keamanan untuk mengamankan *database* transaksi dan *database supplier* agar tidak disalahgunakan dan dimanipulasi oleh pihak yang tidak bertanggung jawab.

Kriptografi adalah ilmu sekaligus seni untuk menjaga kerahasiaan pesan, data, atau informasi dengan cara menyamarkannya menjadi bentuk tersandi yang tidak mempunyai makna. Algoritma kriptografi yang akan digunakan adalah algoritma kriptografi RC4. Berdasarkan cara kerjanya, algorima kriptografi RC4 merupakan jenis *stream cipher* yang dinilai sangat cepat dalam prosesnya, kurang lebih 10 kali lebih cepat dari DES. Algoritma dari metode RC4 *stream cipher* ini terdiri dari dua bagian, yaitu: *key setup* dan *stream generation*. Pada *key setup* terdapat tiga tahapan proses di dalamnya, yaitu Inisialisasi S-Box, menyimpan *key* dalam *key byte array*, permutasi pada S-Box. Pada *stream generation* akan menghasilkan nilai *pseudorandom* yang akan dikenakan operasi XOR untuk menghasilkan *ciphertext* ataupun sebaliknya yaitu menghasilkan *plaintext* [4]. Dengan demikian penulis menerapkan "Metode Algoritma Rc4 *(Rivest Code 4)* untuk Pengamanan Database Transaksi Pada *Glory* Digital Sablon". Tujuan dari

penelitian ini adalah memberi keamanan *database* dan menerapkan Algoritma *Rivest Code* 4 (RC4) untuk mengamankan *database* transaksi dan database *supplier* agar memperoleh hasil enkripsi dari Algoritma *Rivest Code* 4 (RC4).

2. LITERATURE REVIEW

Penerapan algoritma kriptografi, khususnya RC4 (Rivest Cipher 4), telah banyak dibahas dalam berbagai bidang untuk memastikan kerahasiaan dan integritas data. Dalam konteks keamanan informasi, kriptografi menjadi teknik dasar untuk melindungi data sensitif dari akses dan manipulasi yang tidak sah. Bagian ini mengulas literatur yang ada tentang algoritma kriptografi, dengan fokus khusus pada RC4, penerapannya dalam mengamankan data, dan kinerjanya dalam berbagai sistem.

2.1 Algoritma Kriptografi

Kriptografi, yang berasal dari kata Yunani *krypto* (yang berarti rahasia) dan *graphia* (yang berarti tulisan), telah digunakan selama lebih dari 4.000 tahun untuk mengamankan informasi (Pamungkas & Muhammad, 2022). Kemajuan kriptografi modern pada abad ke-20 menghasilkan metode yang lebih kompleks dan kuat, termasuk algoritma enkripsi simetris dan asimetris. Enkripsi simetris, di mana kunci yang sama digunakan untuk enkripsi dan dekripsi, tetap menjadi teknik yang paling banyak digunakan karena efisiensinya dan kecepatannya.

Di antara metode enkripsi simetris, algoritma RC4 sangat terkenal karena kecepatan dan kesederhanaannya. RC4 adalah algoritma cipher stream yang dirancang oleh Ron Rivest pada tahun 1987 dan banyak digunakan dalam komunikasi yang aman dan enkripsi file (Awaludin, 2014). Algoritma ini bekerja dengan menggunakan kunci panjang variabel, yang dapat berkisar antara 1 hingga 256 byte, dan dikenal karena efisiensinya dalam mengenkripsi data dengan kecepatan tinggi. RC4 menghasilkan aliran kunci pseudo-acak yang kemudian di-XOR-kan dengan plaintext untuk menghasilkan ciphertext. Hal ini menjadikannya cocok untuk aplikasi yang membutuhkan baik keamanan data maupun kecepatan pemrosesan data, seperti pada pengamanan database transaksi di GDS.

2.2 Aplikasi RC4

RC4 telah diterapkan dengan sukses di berbagai bidang, termasuk enkripsi email, pengamanan dokumen digital, dan pesan mobile. Misalnya, Suwarsita Febriyani dan Arfriandi (2021) mengeksplorasi penggunaan RC4 untuk mengamankan soal ujian digital, yang membuktikan efektivitasnya dalam melindungi dokumen dari akses yang tidak sah. Dalam studi mereka, mereka menyoroti kemampuan RC4 untuk mengamankan format file seperti DOC, PDF, dan PPT tanpa mengubah ukurannya, menjadikannya alat yang efisien untuk mengamankan dokumen sensitif.

Dalam aplikasi lainnya, Rivaldi dan Subandi (2021) fokus pada penggunaan RC4 untuk mengamankan komunikasi melalui email di lingkungan perusahaan. Dengan mengenkripsi email yang dipertukarkan antara divisi di PT Titan Infra Energy, mereka menunjukkan bahwa RC4 dapat melindungi data negosiasi dan penjualan yang sensitif. Studi ini menekankan kenyataan bahwa RC4 dapat digunakan dalam sistem komunikasi real-time, di mana kecepatan enkripsi dan dekripsi sangat penting.

Selain itu, RC4 telah diterapkan dalam aplikasi mobile, terutama untuk mengamankan pesan SMS. Dewi et al. (2020) memeriksa penggunaan RC4 untuk

mengamankan pesan SMS pada perangkat Android, yang menunjukkan bahwa algoritma ini memberikan lapisan perlindungan yang kuat untuk melindungi komunikasi pribadi.

2.3 Masalah Keamanan dan Keterbatasan RC4

Meskipun banyak digunakan, RC4 memiliki beberapa kelemahan, terutama dalam aplikasi kriptografi modern. Salah satu masalah utama adalah kerentanannya terhadap serangan tertentu, seperti bias dalam aliran kuncinya, yang dapat dimanfaatkan untuk mengungkapkan informasi tentang plaintext. Meskipun RC4 tetap efisien dalam hal kecepatan, kerentanannya terhadap serangan tersebut membuatnya kurang aman untuk penggunaan jangka panjang (Heriyanto, 2018).

Penelitian oleh Yuda Irawan et al. (2021) menganalisis risiko yang terkait dengan penggunaan RC4 dalam pengamanan data dan mengakui bahwa meskipun dapat menyembunyikan informasi dengan baik, panjang kunci tetap menjadi faktor keamanan yang penting. Kunci yang digunakan dalam RC4 sering dibatasi hingga 128 bit, meskipun panjang kunci hingga 2048 bit juga dimungkinkan. Keterbatasan ini menjadi perhatian penting bagi sistem yang menangani data yang sangat sensitif, seperti transaksi keuangan dan informasi pribadi.

2.4 Studi Perbandingan RC4 dengan Algoritma Kriptografi Lainnya

Beberapa studi telah membandingkan RC4 dengan algoritma kriptografi lainnya, seperti Data Encryption Standard (DES) dan Advanced Encryption Standard (AES), dengan fokus pada kekuatan dan kelemahan masing-masing. RC4 sering dipilih karena kecepatannya, tetapi AES dan DES memberikan keamanan yang lebih kuat karena proses enkripsi yang lebih kompleks. Widiyanto (2018) mencatat bahwa meskipun RC4 lebih cepat dibandingkan dengan AES dan DES, algoritma ini memiliki beberapa kelemahan, seperti kerentanannya terhadap serangan bit-flipping dan serangan pemulihan kunci, yang membuatnya kurang aman untuk digunakan dalam aplikasi yang memerlukan tingkat keamanan yang tinggi.

Literatur juga menunjukkan bahwa kinerja RC4 sangat tergantung pada panjang kunci yang digunakan, dengan kunci yang lebih panjang memberikan keamanan lebih baik namun dengan pengorbanan pada kecepatan. Dalam konteks sistem GDS, yang menangani data transaksi, kompromi antara kecepatan dan keamanan ini harus dipertimbangkan dengan hati-hati saat mengimplementasikan RC4.

2.5 Kriptografi dalam Sistem Transaksi

Dalam sistem transaksi seperti GDS, di mana pengelolaan database yang aman sangat penting, algoritma kriptografi memainkan peran penting dalam melindungi integritas data dan mencegah akses yang tidak sah. Penelitian oleh Arfiadi (2020) menunjukkan bahwa penerapan algoritma enkripsi seperti RC4 untuk keamanan database dapat secara signifikan mengurangi risiko pelanggaran data dan pencurian. Enkripsi data yang sensitif, seperti informasi supplier dan catatan transaksi, memastikan bahwa bahkan jika penyerang berhasil mengakses sistem, data tersebut tetap tidak dapat dibaca tanpa kunci dekripsi yang tepat.

Menggabungkan kriptografi dalam sistem manajemen database juga memberikan lapisan perlindungan tambahan terhadap ancaman internal, termasuk serangan dari dalam. Seperti yang disorot oleh Heriyanto (2018), penggunaan algoritma enkripsi seperti RC4 memastikan bahwa hanya

pengguna yang sah yang dapat mengakses dan memodifikasi data, menjadikannya alat yang sangat penting untuk sistem yang memerlukan tingkat kerahasiaan dan kepercayaan yang tinggi.

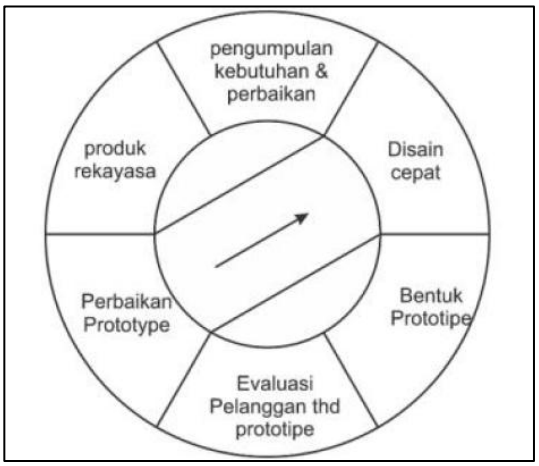
3. BAHAN DAN METODE

Pada bagian ini akan dijelaskan mengenai metode pengembangan perangkat lunak, metode kriptografi, tools perangkat lunak.

2.1 Metode Pengembangan Perangkat Lunak

Prototyping adalah proses pembuatan model sederhana *software* yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan [5].

Ilustrasi dari metode *prototype* dapat dilihat pada Gambar 1 berikut:



Gambar 1 Ilustrasi Metode prototype

Berikut adalah penjelasan setiap tahapan dalam metode *prototype* [5].

1. Pengumpulan kebutuhan dan perbaikan. Menetapkan segala kebutuhan untuk pembangunan perangkat lunak.
2. Desain cepat. Tahap penerjemahan dari keperluan atau data yang telah dianalisis ke dalam bentuk yang mudah dimengerti oleh *user*.
3. Bentuk *prototype*. Menerjemahkan data yang telah dirancang ke dalam bahasa pemrograman (Program contoh atau setengah jadi).
4. Evaluasi pelanggan terhadap *prototype*. Program yang sudah jadi diuji oleh pelanggan, dan bila ada kekurangan pada program dapat ditambahkan.
5. Perbaikan *prototype*. Perbaikan program yang sudah jadi, sesuai dengan kebutuhan konsumen. Kemudian dibuat program kembali dan di evaluasi oleh konsumen sampai semua kebutuhan *user* terpenuhi.
6. Produk rekayasa. Program yang sudah jadi dan seluruh kebutuhan *user* sudah terpenuhi.

Metode *prototype* memiliki beberapa kelebihan dan kelemahan sebagai berikut [5]:

1. Kelebihan Metode *Prototype*
   a. Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan.
   b. Penentuan kebutuhan lebih mudah diwujudkan.
   c. Mempersingkat waktu pengembangan produk perangkat lunak.
   d. Adanya komunikasi yang baik antara pengembang dan pelanggan.
   e. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.

f. Lebih menghemat waktu dalam pengembangan sistem.
g. Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya.

2. Kelemahan Metode *Prototype*
   a. Proses analisis dan perancangan terlalu singkat.
   b. Biasanya kurang fleksibel dalam mengahadapi perubahan.
   c. Walaupun pemakai melihat berbagai perbaikan dari setiap versi *prototype*, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.
   d. Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien.

2.2 Kriptografi

Kriptografi adalah ilmu yang digunakan untuk mengamankan informasi dengan mengubahnya menjadi format yang tidak dapat dimengerti tanpa kunci yang tepat. Penelitian ini menggunakan algoritma RC4 untuk mengamankan database transaksi di Glory Digital Sablon (GDS). RC4 dipilih karena kemampuannya untuk mengenkripsi data dengan cepat, sehingga cocok untuk aplikasi yang membutuhkan kecepatan pemrosesan data. Algoritma RC4 bekerja dengan dua fase utama: *key setup* dan *stream generation*, yang keduanya memanfaatkan operasi XOR untuk menghasilkan ciphertext yang dapat di-dekripsi kembali dengan kunci yang sama. Penelitian ini akan menguji keefektifan RC4 dalam mengamankan data transaksi serta mengukur dampaknya terhadap performa sistem. Kriptografi sendiri telah di gunakan dari 4000 tahun yang lalu dan nama kriptografi itu sendiri terdiri dari gabungan kata *Crypto* yang artinya (Rahasia) dan *Graphia* artinya (tulisan) yang berasal dari yunani. Kriptografi juga ilmu merumuskan metode yang memungkinkan sebuah informasi yang akan di kirim dan di buat sedemikian rupa sehingga yang dapat memahaminya hanya pengirim dan penerima yang dituju [6].

2.3 Enkripsi dan Dekripsi

Enkripsi adalah proses dimana informasi atau data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal. Sebuah pesan atau data yang masih asli dan belum mengalami penyandian dikenal dengan istilah *plaintext*. Kemudian setelah disamarkan dengan suatu cara penyandian, maka *plaintext* ini disebut sebagai *chipertext*. Proses penyamaran dari *plaintext* ke *ciphertext* disebut enkripsi (*encryption*), dan proses pengembalian dari *ciphertext* menjadi *plaintext* kembali disebut dekripsi (*decryption*). Adapun komponen enkripsi adalah seperti berikut [7]:

1. *Plaintext*: adalah data asli yang akan dikirm ke penerima tertentu. Data-data ini akan menjadi input ke algoritma enkripsi.
2. Algoritma enkripsi: adalah serangkaian proses yang akan dieksekusi ke dalam *plaintext* dengan bantuan kunci rahasia untuk menghasilkan *ciphertext*.
3. Kunci rahasia: adalah nilai yang digunakan untuk menggabungkan *plaintext* dan mengubahnya menjadi *ciphertext*.
4. *Ciphertext*: adalah output dari *plaintext* asli yang dimasukkan ke algoritma enkripsi.
5. Algoritma dekripsi: adalah sekumpulan proses yang akan

dieksekusi menjadi *ciphertext* dengan bantuan kunci rahasia untuk menghasilkan teks asli atau *plaintext*.

2.4 Algoritma RC4

Algoritma RC4 merupakan salah satu *stream cipher* tertua dan paling liar digunakan di dunia. Banyak digunakan untuk aplikasi dunia nyata termasuk *Microsoft Office*, *Secure Socket Layer* (SSL), *Wired Equivalent Privacy* (WEP), dll. Algoritma ini tidak harus menunggu sejumlah input data tertentu sebelum diproses, atau menambahkan *byte* tambahan untuk mengenkripsi. RC4 menggunakan panjang variabel kunci dari 1 sampai 256 *byte* untuk menginisialisasi *state table*. *State table* digunakan untuk pengurutan menghasilkan *byte pseudo-random* yang kemudian menjadi *stream pseudo-random*. Setelah di XOR dengan *plaintext* sehingga didapatkan *ciphertext*. Tiap elemen pada *state table* di *swap* sedikitnya sekali. Kunci RC4 sering dibatasi sampai 40 bit, tetapi dimungkinkan untuk menggunakan kunci 128-bit. RC4 memiliki kemampuan penggunaan kunci antara 1 sampai 2048 bit. Panjang kunci merupakan faktor utama dalam sekuritas data. RC4 dapat memili kunci sampai 128-bit. Algoritma RC4 dapat digunakan untuk mengenkripsi dengan mengkombinasi *plainteks* dengan menggunakan *bit-wise Xor* (*Exclusive-or*). Proses dekripsinya dilakukan dengan cara yang sama karena Xor merupakn fungsi simetrik [8].

2.5 Tools Perangkat Lunak

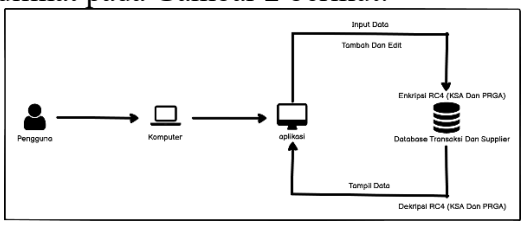
Perangkat lunak yang digunakan adalah Astah Professional, Balsamiq Wireframe, XAMPP, Visual Studio Code, Bahasa pemrograman PHP, dan database MySQL.

4. HASIL DAN BAHASAN

Pada bagian ini terdapat penjabaran mengenai hasil desain sistem, implementasi, perhitungan dan pengujian.

3.1 Arsitektur Sistem

Pada arsitektur sistem menggambarkan proses dari keseluruhan sistem yang dibuat, dapat dilihat pada Gambar 2 berikut:



Gambar 2 Arsitektur Sistem

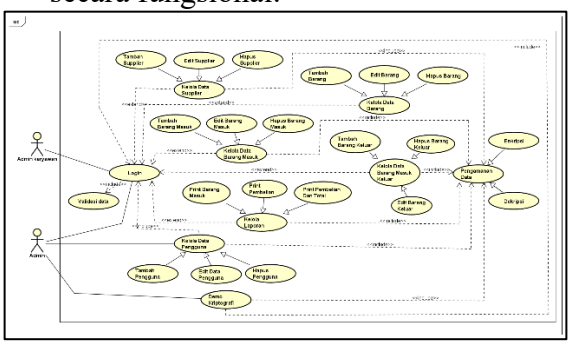
3.2 Desain Sistem

Di dalam desain system, penulis akan menggambarkan baik *usecase diagram, activity diagram* dan *class diagram.*

1. *Usecase Diagram*

Use case atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat [9].

Pada Gambar 3 berikut menggambarkan kebutuhan sistem secara fungsional:



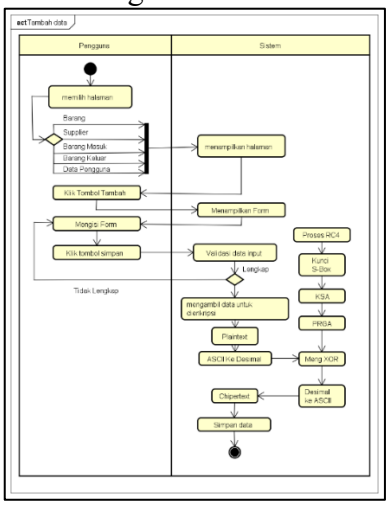
Gambar 3 Usecase Diagram

2. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses atau

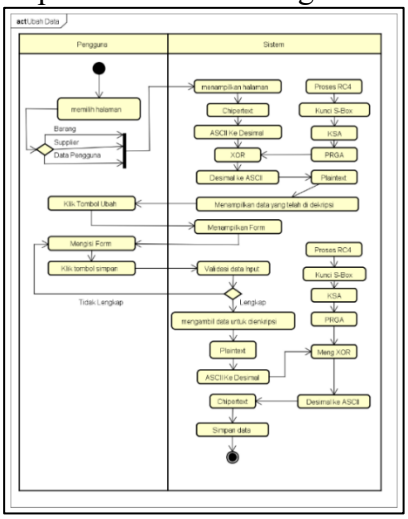
menu yang ada pada perangkat lunak [9].

Berikut gambaran untuk menjelaskan proses tambah data dengan algoritma RC4, yang dapat dilihat pada Gambar 4 sebagai berikut:



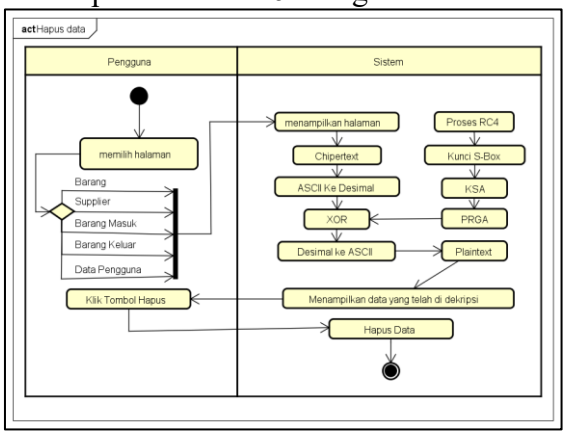
Gambar 4 Activity Diagram Tambah Data

Activity Diagram ubah data dapat dilihat pada Gambar 5 sebagai berikut:



Gambar 5 Activity Diagram Ubah Data

Activity Diagram hapus data dapat dilihat pada Gambar 6 sebagai berikut:

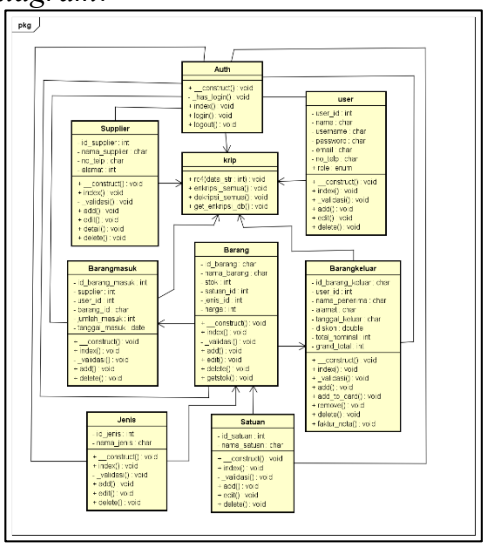


Gambar 6 Activity Diagram hapus data

3. *Class Diagram*

Class diagram menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [9].

Pada Gambar 7 berikut menggambarkan struktur sistem *class diagram*:



Gambar 7 Class Diagram

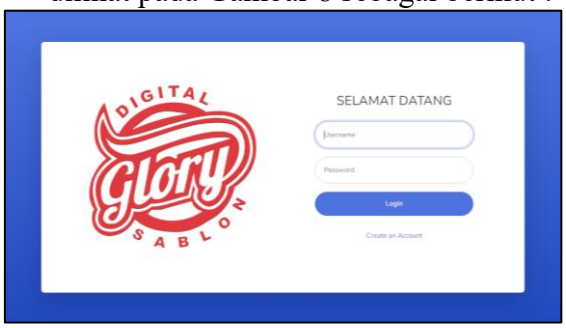
3.1 Implementasi

Berikut ini adalah tampilan sistem. Fitur yang ada terdiri dari halaman login, halaman dashboard, halaman *supplier,* halaman barang, halaman barang masuk, halaman barang keluar,

ambil laporan, manajemen pengguna, dan halaman demo.

a. Tampilan Halaman Login

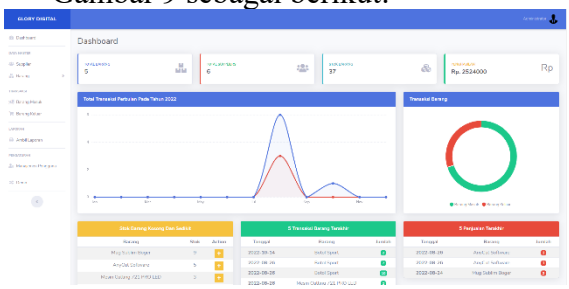
Halaman Login merupakan halaman pertama kali mengakses aplikasi. Dapat dilihat pada Gambar 8 sebagai berikut :



Gambar 8 Halaman Login

b. Tampilan Halaman Dashboard

Halaman Dashboard merupakan halaman awal ketika pengguna masuk setelah login. Dapat dilihat pada Gambar 9 sebagai berikut:



Gambar 9 Halaman Dashboard

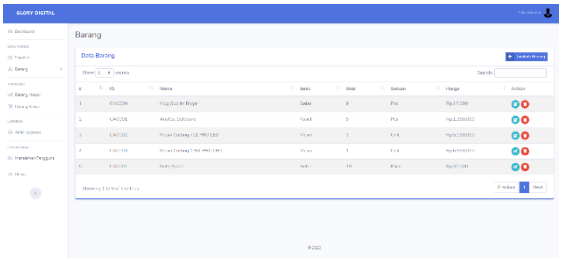
c. Halaman *Supplier*

Halaman *Supplier* merupakan halaman untuk menginputkan data supplier yang harus diamankan. Dapat dilihat pada Gambar 10 sebagai berikut:

Gambar 10 Halaman Supplier

d. Halaman Barang

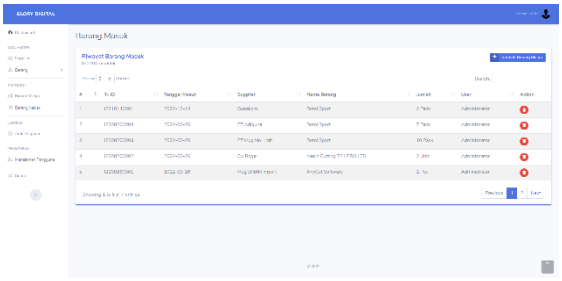
Halaman barang merupakan halamanuntuk menginputkan data barang, mengedit dan menghapus. Dapat dilihat pada Gambar 11 sebagai berikut:



Gambar 11 Halaman Barang

e. Halaman Barang Masuk

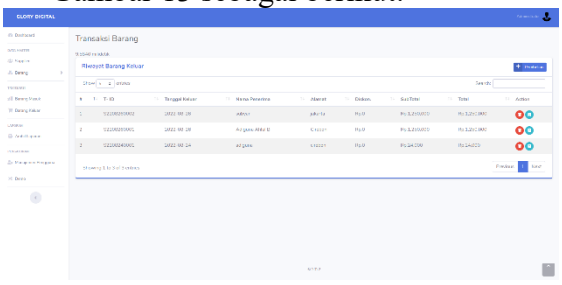
Halaman Barang Masuk merupakan halaman untuk menginputkan data barang masuk yang harus diamankan. Dapat dilihat pada Gambar 12 sebagai berikut:



Gambar 12 Halaman Barang Masuk

f. Halaman Barang Keluar

Halaman Barang Keluar merupakan halaman untuk menginputkan data barang keluar atau pembelian yang harus diamankan. Dapat dilihat pada Gambar 13 sebagai berikut:

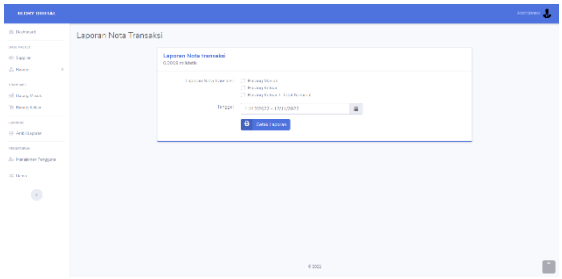


Gambar 13 Halaman Barang Keluar

g. Halaman Ambil Laporan

Halaman Ambil Laporan merupakan halaman untuk untuk mengambil laporan transaksi barang masuk, barang keluar dan nominal berdasarkan tanggal.

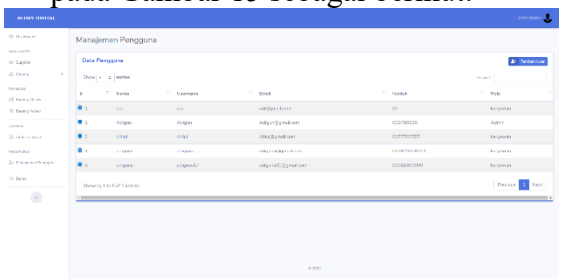
Dapat dilihat pada Gambar 14 sebagai berikut:



Gambar 14 Halaman Ambil Laporan

h. Halaman Manajemen Pengguna

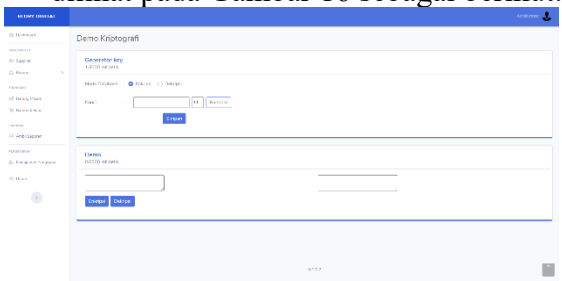
Halaman Manajemen Penggun merupakan halaman untuk mengelola pengguna pada aplikasi. Dapat dilihat pada Gambar 15 sebagai berikut:



Gambar 15 Halaman Manajemen Pengguna

i. Halaman Demo

Halaman Demo merupakan halaman untuk demo aplikasi, mode database enkripsi & dekripsi dan mengganti kunci yang bisa di generate berdasarkan panjang kata yang di inginkan. Dapat dilihat pada Gambar 16 sebagai berikut:



Gambar 16 Halaman Demo

3.2 Perhitungan Metode Rivest Code 4

Dalam proses enkripsi dan dekripsi data, tahapan awal adalah dengan melakukan inisialisasi *plainteks* dan kunci pada Tabel 1 sebagai berikut:

Tabel 1 Inisialisasi Plainteks & Kunci

Plainteks (P)	Kunci (K)
bahan	glory

Selanjutnya melakukan *S-Box* berdasarkan panjang kunci, dapat dilihat pada Tabel 2 sebagai berikut:

Tabel 2 S-Box

State Array S	0	1	2	3	4
State Array K	g	l	o	r	y

Lalu mengubah kunci ke angka desimal dapat dilihat pada Tabel 3 sebagai berikut:

Tabel 3 mengubah kunci ke desimal

Kunci K[i] = K	Desimal
K[0] = g	103
K[1] = l	108
K[2] = o	111
K[3] = r	114
K[4] = y	121

Setelah melakukan proses awal selanjutnya adalah melakukan proses pengacakan kunci array (*Key Scheduling Algorithm*), karena dalam proses ini menggunakan kunci 5 byte maka proses pengulangan dilakukan sebanyak 5x, perhitungannya sebagai berikut:

Pengulangan 1

i = 0

j = (j +S[i] + K[i]) mod pkunci

j = (0 +S[0] + K[0]) mod 5

j = (0 + 0 + 103) mod 5

j = 103 mod 5

j = 3

Tukar S[i] dengan S[j] (S[0] , S[3])

Hasil tukar

Array S 0 1 2 3 4
Tukar 3 1 2 0 4

Pengulangan 2
i = 1
j = (j +S[i] + K[i]) mod pkunci
j = (3 +S[1] + K[1]) mod 5
j = (3 + 1 + 108) mod 5
j = 112 mod 5
j = 2
Tukar S[i] dengan S[j] (S[1] , S[2])
Hasil Tukar
Array S 3 1 2 0 4
Tukar 3 2 1 0 4

Pengulangan 3
i = 2
j = (j +S[i] + K[i]) mod pkunci
j = (2 +S[2] + K[2]) mod 5
j = (2 + 1 + 111) mod 5
j = 114 mod 5
j = 4
Tukar S[i] dengan S[j] (S[2] , S[4])
Hasil Tukar
Array S 3 2 1 0 4
Tukar 3 2 4 0 1

Pengulangan 4
i = 3
j = (j +S[i] + K[i]) mod pkunci
j = (4 +S[3] + K[3]) mod 5
j = (4 + 0 + 104) mod 5
j = 108 mod 5
j = 3
Tukar S[i] dengan S[j] (S[3] , S[3])
Hasil Tukar
Array S 3 2 4 0 1

Pengulangan 5
i = 4
j = (j +S[i] + K[i]) mod pkunci
j = (3 +S[4] + K[4]) mod 5
j = (3 + 1 + 121) mod 5
j = 125 mod 5
j = 0
Tukar S[i] dengan S[j] (S[4] , S[0])
Hasil Tukar
Array S 3 2 4 0 1
Tukar 1 2 4 0 3

Setelah melakukan proses KSA langkah berikutnya adalah melakukan proses pseudo random generation automation (PRGA). Proses ini dilakukan sebanyak 5x berdasarkan panjang kunci, untuk perhitungannya sebagai berikut:

Pengulangan 1
i = 0, j = 0
i = (i + 1) mod pkunci
i = (0 +1) mod 5
i = 1
j = (j +S[i]) mod pkunci
j = (0+S[1]) mod 5
j = (0 + 2) mod 5
j = 2
Tukar S[i] dengan S[j] (S[1] , S[2])
Hasil Tukar
Array S 1 2 4 0 3
Tukar 1 4 2 0 3
K1 = (S[1] + S[2]) mod 5
K1 = (4 + 2) mod 5
K1 = 1
Biner K1 = 0 0 0 0 0 0 0 1

Pengulangan 2
i = 1, j = 2
i = (i + 1) mod pkunci
i = (1 +1) mod 5
i = 2
j = (j +S[3]) mod pkunci
j = (2 +S[2]) mod 5
j = (2 + 2) mod 5
j = 4
Tukar S[i] dengan S[j] (S[2] , S[4])
Hasil Tukar
Array S 1 4 2 0 3
Tukar 1 4 3 0 2
K2 = (S[2] + S[4]) mod 5
K2 = (3 + 2) mod 5
K2 = 0
Biner K2 = 0 0 0 0 0 0 0 0

Pengulangan 3
i = 2, j = 4
i = (i + 1) mod pkunci
i = (2 +1) mod 5
i = 3
j = (j +S[i]) mod pkunci
j = (4 +S[3]) mod 5

j = (4 + 0) mod 5
j = 4
Tukar S[i] dengan S[j] (S[3] , S[4])
Hasil Tukar
Array S 1 4 3 0 2
Tukar 1 4 3 2 0
K3 = (S[3] + S[4]) mod 5
K3 = (2 + 0) mod 5
K3 = 2
Biner K3 = 0 0 0 0 0 0 1 0

Pengulangan 4
i = 3, j = 4
i = (i + 1) mod pkunci
i = (3 +1) mod 5
i = 4
j = (j +S[i]) mod pkunci
j = (4 +S[4]) mod 5
j = (4 + 0) mod 5
j = 4
Tukar S[i] dengan S[j] (S[4] , S[4])
Hasil Tukar
Array S 1 4 3 2 0
K4 = (S[4] + S[4]) mod 5
K4 = (0 + 0) mod 5
K4 = 0
Biner K4 = 0 0 0 0 0 0 0 0

Pengulangan 5
i = 4, j = 4
i = (i + 1) mod pkunci
i = (4 +1) mod 5
i = 0
j = (j +S[i]) mod pkunci
j = (4 +S[0]) mod 5
j = (4 + 1) mod 5
j = 0
Tukar S[i] dengan S[j] (S[0] , S[0])
Hasil Tukar
Array S 1 4 3 2 0
K5 = (S[0] + S[0]) mod 5
K5 = (1 + 1) mod 5
K5 = 2
Biner K5 = 0 0 0 0 0 0 1 0

Setelah selesai melakukan proses PRGA maka akan mendapatkan hasil kunci untuk setiap karakter mulai dari K1 sampai K5, dapat dilihat pada Tabel 4 sebagai berikut:

Tabel 4 Hasil Kunci K1 Sampai K5

Kunci	Biner
K1	0 0 0 0 0 0 0 1
K2	0 0 0 0 0 0 0 0
K3	0 0 0 0 0 0 1 0
K4	0 0 0 0 0 0 0 0
K5	0 0 0 0 0 0 1 0

Sebelum melakukan XOR kata bahan terlebih dahulu diubah ke kode ASCII, bisa dilihat pada Tabel 5 sebgai berikut:

Tabel 5 Kode ASCII kata bahan

Plainteks	Biner
b	0 1 1 0 0 0 1 0
a	0 1 1 0 0 0 0 1
h	0 1 1 0 1 0 0 0
a	0 1 1 0 0 0 0 1
n	0 1 1 0 1 1 1 0

Langkah berikutnya adalah proses melakukan enkripsi *Rivest Code* 4 (RC4) yaitu meng-XOR-kan *Pseudo-random Generation Automation* dengan *plainteks* yang dapat dilihat pada Tabel 6 sebagai berikut:

Tabel 6 Enkripsi Rivest Code 4 (RC4)

Plainteks	b	a	h	a	n
	01100010	01100001	01101000	01100001	01101110
Kunci	00000001	00000000	00000010	00000000	00000010
Cipherteks	01100011	01100001	01101010	01100001	01101100
	c	a	j	a	l

Langkah berikutnya adalah proses melakukan dekripsi *Rivest Code* 4 (RC4) yaitu meng-XOR-kan *Pseudo-random Generation Automation*

dengan *cipherteks* yang dapat dilihat pada Tabel 7 sebagai berikut:

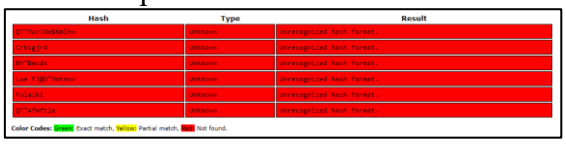
Tabel 7 Dekripsi Rivest Code 4 (RC4)

Cipherteks	c	a	j	a	l
	01100011	01100001	01101010	01100001	01101100
Kunci	00000001	00000000	00000010	00000000	00000010
Plainteks	01100010	01100001	01101000	01100000	01101110
	B	a	h	a	n

Jadi setelah dilakukan pengenkripsian maka kata bahan berubah menjadi kode biner 01100011 01100001 01101010 01100001 01101100 yang dalam kode ASCII berarti bajal.

2.3 Hasil Uji CrackStation

Hasil pengujian *CrackStation* sebanyak 6 kali pada database nama *supplier* bahwa algoritma RC4 tidak berhasil dipecahkan. Nama *supplier* dan hasil pengujian *CrackStation* dapat dilihat pada Gambar 17 berikut:



Gambar 17 Hasil Uji CrackStation

2.4 Hasil Uji Performa

Uji coba ini dilakukan untuk mengetahui seberapa cepat pemrosesan query tanpa dilakukan pengimplementasian fungsi enkripsi RC4, hasilnya dapat dilihat sebagai berikut :

Tabel 8 Hasil Uji Coba Tanpa Enkripsi RC4

Uji coba	Banyak Kata	Waktu
1	100	182 milidetik
2	200	208 milidetik
3	400	214 milidetik
4	700	226 milidetik

Dapat dilihat pada Tabel 8 bahwa semakin banyak kata maka semakin lama waktu yang diperlukan untuk pemrosesan query.

Uji coba berikut bertujuan untuk mengetahui seberapa cepat pemrosesan enkripsi menggunakan algoritma RC4 dari mulai pengiriman query hingga tersimpan di *server database.* Uji coba ini dilakukan dengan menggunakan panjang kunci enkripsi sepanjang 5, 20 dan 90.

Tabel 9 Hasil Uji Enkripsi panjang 5 karakter

Uji coba	Banyak Kata	Waktu
1	100	182 milidetik
2	200	208 milidetik
3	400	214 milidetik
4	700	226 milidetik

Tabel 10 Hasil Uji Enkripsi panjang 20 karakter

Uji coba	Banyak Kata	Waktu
1	100	208 milidetik
2	200	228 milidetik
3	400	230 milidetik
4	700	284 milidetik

Tabel 11 Hasil Uji Enkripsi panjang 90 karakter

Uji coba	Banyak Kata	Waktu
1	100	208 milidetik
2	200	228 milidetik
3	400	230 milidetik
4	700	284lidetik

Dari ketiga tabel dapat disimpulkan bahwa panjang kunci dapat mempengaruhi kecepatan dari enkripsi dan dekripsi tapi dengan selisih waktu yang sangat kecil, sehingga perbedaan panjang kunci tidak mempengaruhi performa dari fungsi enkripsi dan dekripsi secara keseluruhan.

DISKUSI

Hasil pengujian CrackStation menunjukkan bahwa cipherteks yang dihasilkan dari algoritma RC4 tidak dapat dipecahkan, yang menandakan bahwa data transaksi dan supplier yang dienkripsi menggunakan algoritma ini

sangat aman. Pengujian performa menunjukkan bahwa meskipun panjang kunci berpengaruh terhadap waktu enkripsi dan dekripsi, perbedaan waktu yang dihasilkan sangat kecil, yang berarti penggunaan RC4 tetap efisien dalam konteks aplikasi GDS yang membutuhkan kecepatan. Namun, meskipun RC4 terbukti efektif dalam menjaga keamanan data, penting untuk dicatat bahwa algoritma ini memiliki beberapa kelemahan yang perlu dipertimbangkan, seperti potensi kerentanannya terhadap serangan tertentu. Oleh karena itu, meskipun RC4 memberikan tingkat keamanan yang tinggi untuk GDS, penelitian lebih lanjut mengenai pemilihan algoritma yang lebih baru dan lebih aman tetap diperlukan.. Penulis melakukan tinjauan Pustaka terhadap jurnal dan conference yang berkaitan dengan objek penelitian yang akan dilakukan, diantaranya adalah sebagai berikut:

4.1 Aplikasi Kriptografi RC4 Untuk Pengamanan Email Berbasis Web Pada PT. Titan Infra *Enery*, Ricky Rivaldi & Subandi, SKANIKA (Sistem Komputer dan Teknik Informatika), 2021

Titan Infra Energy adalah perusahan yang bergerak dibidang industri pertambangan, semua informasi mengenai proses negosiasi dan penjualan dengan perusahaan lain yang mebutuhkan supply dari PT.Titan Infra Energy didistribusikan melalui email dan dikelola sendiri oleh setiap divisinya. Salah satu cara yang dapat dilakukan untuk melindungi informasi atau data adalah dengan menggunakan teknik enkripsi dan dekripsi. Oleh karena itu, penelitian ini akan diimplementasikan RC4 dalam sebuah program aplikasi berbasis web untuk mengamankan file-file penting. Dalam pengamanan file doc, pdf, ppt, xlsx dan foto menggunkan algoritma RC4 bisa mengamankan file tersebut. Hasil file yang di enkripsi menjadi acak dan sulit dipahami ketika dibuka, ukuran file tidak berubah tetapi untuk melakukan proses enkripsi memerlukan waktu yang lama dibandingkan dekripsi.

4.2 Implementasi Algoritma RC4 Pada Sistem Pengamanan Dokumen Digital Soal Ujian, Fauziyah Suwarsita Febriyani & Arief Arfriandi, JISKA (Jurnal Informatika Sunan Kalijaga), 2021

Menurut DataLossDB, pada tahun 2014 kebocoran data sekitar 50% terjadi pada sektor bisnis, sekitar 20% terjadi pada sektor pemerintahan dan sekitar 30% terjadi pada sektor kesehatan dan pendidikan. Untuk menjaga keamanan data dan menghindari kebocoran data dapat dilakukan berbagai cara salah satunya yaitu dengan menggunakan kriptografi. Untuk pengamanan dokumen digital dengan mengimplementasikan algoritma RC4 teruji aman. Sistem dapat mengenkripsi dan mendekripsikan kembali file dokumen dengan format *file doc* menggunakan algoritma RC4 sehingga dapat mengamankan dokumen soal ujian. Untuk mengamankan dokumen, *user* harus menginputkan dokumen beserta kuncinya dan dilakukan pengujian *blackbox* menunjukkan bahwa sistem dapat berjalan sesuai dengan inputan yang diberikan. Kemudian pada pengujian software CrackStation untuk menguji 10 sampel ciphertext menunjukkan bahwa ciphertext 100% tidak dapat dipecahkan.

4.3 Aplikasi Kriptografi Dalam Mengamankan Pesan Teks Dengan Metode Algoritma RC4 Berbasis Android, Rosna Dewi,T.M, dkk, JUTIKOMP (Teknologi dan Ilmu Komputer Prima), 2020

Salah satu contoh media komunikasi short service message (SMS), yang dapat melakukan berbagai teknik pengambilan informasi baik legal maupun illegal.

Berbagai teknikperlindungan informasi yang dirahasiakan dari yang tidak berhak telah dikembangkan untuk melindungi informasi. Algoritma RC4 adalah salah satu teknik yang digunakan untuk membuat pesan rahasia dengan teknik enkripsi dan dekripsi pesan. Penelitian ini bertujuan untuk merancang suatu aplikasi kriptongrafi dalam pengamanan pesan teks, yang diimplementasikan pada perangkat mobile berbasis android.

4.4 Analysis Of Voice Data Security Security By Using The Rc4 Algorithm Keywords, Rika nurhidayati & Salauddin Muis, JISA (Internasional Informatika dan Sains), 2021

Keamanan dan kerahasiaan data merupakan salah satu aspek terpenting dalam bidang komunikasi khususnya komunikasi dengan menggunakan media komputer. Mengirim pesan, data dan informasi yang sangat penting membutuhkan tingkat keamanan yang tinggi. Unit atau data biasanya byte atau kadang-kadang bahkan sedikit (byte dalam kasus RC4). Pada penelitian ini akan dibahas algoritma RC4 untuk mengenkripsi dan mendekripsi file mp3 yang tersimpan di komputer. Hasil kesimpulan yang didapatkan dengan Algoritma RC4, data suara akan sulit diidentifikasi dan dirahasiakan. diketahui bahwa proses enkripsi-dekripsi dalam algoritma RC4 dengan kata sandi yang sama antara enkripsi dan dekripsi dengan plaintext yang sama akan menghasilkan ciphertext yang berbeda, tetapi hasil plaintext tetap sama.

4.5 RC4 Cryptography Implementation Analysis On Text Data, Agung Susilo Yuda Irawan, dkk, Internasional Sisfotek Global, 2021

Dalam beberapa tahun terakhir, ada beberapa kasus yang melibatkan keamanan data, seperti bocornya informasi akun pengguna Facebook. Untuk mengatasi hal tersebut, dilakukan penelitian di bidang keamanan data dan informasi dengan menganalisis salah satu teknik enkripsi dan dekripsi, yaitu kriptografi RC4. Keuntungan yang perlu digaris bawahi adalah kriptografi RC4 ini menggunakan kunci tertentu sebagai referensi, hal-hal seperti ini dapat memberikan kemudahan kepada pembuat enkripsi tetapi juga bisa menjadi ancaman. akibatnya RC4 dapat menyembunyikan informasi dengan sangat baik sementara kunci rahasianya tidak diketahui oleh orang lain. Selain digunakan dalam data teks, kriptografi ini dapat digunakan untuk data lain seperti audio dan video.

5. KESIMPULAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penerapan algoritma kriptografi RC4 untuk mengamankan database transaksi dan supplier di Glory Digital Sablon (GDS) telah berhasil memberikan perlindungan yang efektif. Meskipun panjang kunci berpengaruh terhadap waktu enkripsi dan dekripsi, perbedaannya sangat kecil, sehingga RC4 tetap menjadi pilihan yang efisien untuk aplikasi yang membutuhkan keamanan dan kecepatan pemrosesan data. Namun, meskipun RC4 menunjukkan hasil yang baik, ada kebutuhan untuk penelitian lebih lanjut untuk mengatasi kelemahan-kelemahan yang dimiliki oleh algoritma ini dan untuk mengeksplorasi penggunaan algoritma enkripsi yang lebih baru dan lebih aman. Sebagaimana pengujian software *CrackStation* untuk menguji 6 cipherteks, dengan cara menginputkan cipherteks nama supplier menunjukkan bahwa cipherteks 100% tidak dapat dipecahkan. Melalui pengujian performa bahwa panjang kunci 5, 20, 90 karakter dapat mempengaruhi kecepatan dari enkripsi dan dekripsi tetapi dengan selisih waktu yang sangat kecil, sehingga perbedaan panjang kunci tidak mempengaruhi performa dari fungsi

enkripsi dan dekripsi secara keseluruhan.

6. DAFTAR PUSTAKA

[1] Prayudha, J., Sulaiman, O. K., Mariami, I., Halim, J., & Dharma, S. T. (2021). Implementasi Algoritma RC4 untuk Pengamanan Database Transaksi di Glory Digital Sablon. *KESATRIA: Jurnal Penerapan Sistem Informasi, 1*(1), 45-50..

[2] R. Dewi and M. Johan, "Aplikasi Kriptografi Dalam Mengamankan Pesan Teks Dengan Metode Algoritma Rc4 Berbasis Android," 2020.

[3] R. Fajar, T. Informatika, F. T. Informasi, U. B. Luhur, P. Utara, and K. Lama, "Aplikasi Kriptografi Rc4 Untuk Pengamanan Email," vol. 4, no. 1, pp. 45–50, 2021.

[4] F. Suwarsita Febriyani and A. Arfriandi, "Implementasi Algoritma RC4 pada Sistem Pengamanan Dokumen Digital Soal Ujian," 2021.

[5] W. W. Widiyanto, "Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (Rad)," *J. Inf. Politek. Indonusa Surakarta ISSN*, vol. 4, no. 1, pp. 34–40, 2018.

[6] P. G. Pamungkas and A. H. Muhammad, "Modifikasi Algoritma Kriptografi Caesar Chiper pada Deretan Simbol dan Huruf di Smarphone dan Laptop," *J. Inf. Technol.*, vol. 2, no. 1, pp. 1–5, 2022, doi: 10.46229/jifotech.v2i1.234.

[7] A. Amrulloh and E. I. H. Ujianto, "Kriptografi Simetris Menggunakan Algoritma Vigenere Cipher," *J. CoreIT*, vol. 5, no. 2, pp. 71–77, 2019.

[8] M. Awaludin, "Penerapan Algoritma Rc4 Pada Operasi Xor Untuk Keamanan Pesan Pada Smartphone Berbasis Web," *J. Sist. Inf. Univ. Suryadarma*, vol. 4, no. 1, pp. 16–22, 2014, doi: 10.35968/jsi.v4i1.71.

[9] Y. Heriyanto, "Perancangan Sistem Informasi Rental Mobil Berbasis Web Pada PT.APM Rent Car," vol. 2, no. 2, pp. 64–77, 2018.