DrillBit

The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

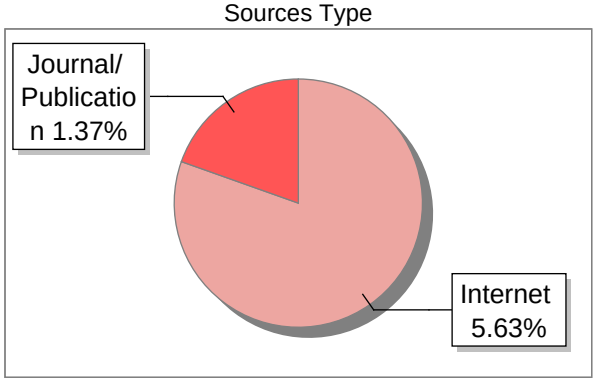
Author Name	Teteh Hayati
Title	7-Jurnal Rara Srieartati-15072024
Paper/Submission ID	2113743
Submitted by	perpustakaanunisbank@edu.unisbank.ac.id
Submission Date	2024-07-15 09:11:53
Total Pages, Total Words	8, 3216
Document type	Article

Result Information

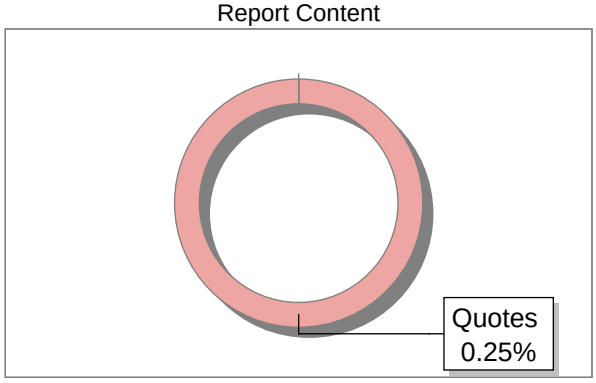
Similarity **7 %**

1 10 20 30 40 50 60 70 80 90

Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 2 Words	Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	Non-English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

7	10	A
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	123dok.com	2	Internet Data
2	adoc.pub	1	Internet Data
3	etheses.uin-malang.ac.id	1	Publication
4	docplayer.info	<1	Internet Data
5	adoc.pub	<1	Internet Data
6	docplayer.info	<1	Internet Data
7	docplayer.info	<1	Internet Data
8	docplayer.info	<1	Internet Data
9	etheses.uin-malang.ac.id	<1	Publication
10	media.unpad.ac.id	<1	Publication

ADOPSI REPETITIF COVERTEXT PADA MODEL ENKRIPSI PDAC

Hari Murti[1], Edy Supriyanto[2], Rara Sriartati Redjeki[3], Eka Ardhianto[4]

[1,2,3] Program Studi Sistem Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
[4] Program Studi Teknik Informatika, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank
[1]harimurti@edu.unisbank.ac.id, [4] ekaardhianto@edu.unisbank.ac.id

Abstrak

Perkembangan internet memberikan dampak kemudahan dalam melakukan komunikasi dan pertukaran informasi. Meskipun demikian, informasi yang dikirimkan melalui internet memiliki kecenderungan terbuka untuk banyak pihak. Permasalahan *confidentiality* informasi yang berkaitan dengan kerahasiaan informasi, menjadi nilai kritis dalam mengamankan informasi bagi entitas terbatas. Teknik kriptografi dan steganografi berperan penting dalam pengamanan informasi. Gabungan keduanya memberikan tingkat keamanan informasi yang lebih kuat dan sulit dipecahkan. Model enkripsi *Parallel Encryption with Digit Arithmetic of Covertext* (PDAC) mengadopsi teknik kriptografi dan teknik steganografi dalam pengamanan pesan. PDAC merupakan model enkripsi dengan berbasis XOR. Pengamanan informasi menggunakan teknik enkripsi perlu memperhatikan aturan aturan yang terdapat dalam algoritma yang digunakan. Seperti halnya dalam penggunaan PDAC sebagai algoritma enkripsi, juga memerlukan kesesuaian pada proses pemilihan coverteksnya. Ketepatan jumlah covertext dalam PDAC mempengaruhi berjalan atau tidaknya PDAC dalam mengenkripsi pesan. Jika jumlah pesan sangat panjang maka akan menjadi sulit bagi pengguna dalam menentukan coverteksnya dan jika jumlah covertext PDAC tidak sesuai, maka proses enkripsi akan menjadi tidak sempurna. Penelitian ini bertujuan melakukan pengubahan model dengan menutup celah PDAC. Modifikasi model yang diusulkan mengadopsi penggunaan covertext yang diterapkan secara berulang. Perulangan covertext yang diterapkan dalam PDAC mampu memproses pengamanan informasi secara menyeluruh. Keuntungan lain yang diperoleh ialah modifikasi ini memudahkan pengguna untuk menggunakan PDAC tanpa harus memenuhi kebutuhan minimum covertextnya.

Kata kunci : Covertext, PDAC, Enkripsi, Kunci Repetitif.

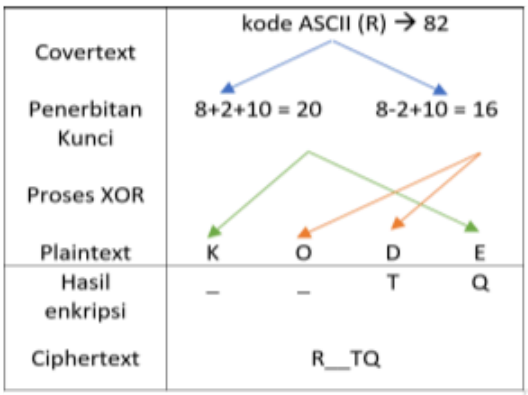
1. Pendahuluan

Keamanan informasi adalah hal penting untuk melindungi informasi tersebut dari pihak yang tidak berwenang. Pengamanan informasi dilakukan melalui proses enkripsi dan dekripsi dalam konteks bidang kriptografi (Nahar & Chakraborty, 2020). Kriptografi bertujuan supaya keaslian isi informasi tetap terjaga dengan mengacak isi informasi sehingga menjadi sulit diterjemahkan (Ardhianto et al., 2021). Dekripsi bertujuan untuk mengembalikan informasi tersebut menjadi bentuk aslinya yang hanya dapat dibaca oleh pihak yang berwenang (Ardhianto et al., 2020a).

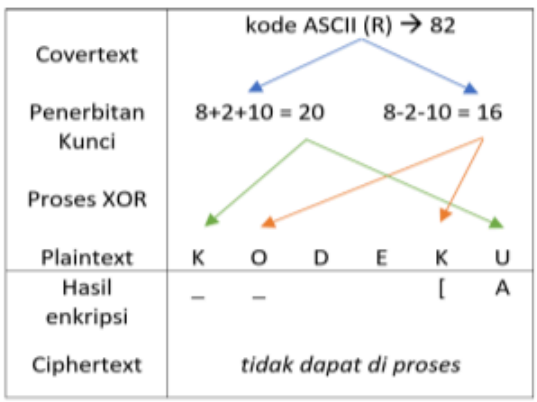
Teknik pengamanan informasi yang lain ialah Steganografi. Steganografi dan Kriptografi berasal dari Bahasa Yunani. Steganografi berasal dari kata *steganos*, artinya "tersembunyi", dan *graphien*, "menulis". Kriptografi berasal dari kata *kryptos* yang bermakna "rahasia" dan *graphein*, "menulis" (Ardhianto et al., 2020a; Telaumbanua & Zebua, 2020). Keduanya memiliki fungsi yang sama namun memiliki tujuan yang berbeda, Steganografi menyembunyikan pesan dengan menyisipkan tiap digit pesan tersebut kedalam pesan lain yang bersifat tidak rahasia yang disebut dengan *cover*, sehingga tak seorang pun tahu bahwa ada pesan rahasia pada pesan lain tersebut. Steganografi menyimpan pesan kedalam *cover* tanpa mengubah format file covernya (N et al., 2007). Teknik kriptografi diartikan sebagai cara menyembunyikan pesan dengan menyamarkan atau mengacak pesan yang memiliki arti lain atau membuat pesan asli menjadi tak berarti. Keuntungan steganografi dibandingkan kriptografi ialah bahwa hasil dari pengubahan pesan yang tidak menimbulkan kecurigaan (Handoko, Ardhianto, Hadiono, et al., 2020). Untuk memperkuat keamanan informasi penggabungan Steganografi dan Kriptografi telah sering diusulkan. Penggabungan tersebut dinilai mempersulit pihak ketiga yang dikenal sebagai "*man in the middle*".

Model enkripsi PDAC (*Parallel Encryption with Digit Arithmetic of Cover Text*) merupakan teknik perhitungan matematika dan konsep paralel untuk pendekatan steganografi berbasis teks (Handoko, Ardhianto, & Supriyanto, 2020; Kataria, Singh, Kumar, & Shekhawat, 2013). PDAC menggunakan Steganografi untuk mengenkripsi pesan, dengan tahap pengubahan digit karakter pada pesan menjadi digit kode ASCII, kode ASCII diubah menjadi kode biner, begitu pula dengan karakter *covertext* yang digunakan. Untuk mengenkripsi

pesan, PDAC membutuhkan 1 karakter sebagai *covertext* untuk membangkitkan 2 kunci enkripsi. Hal tersebut dinilai sebagai kapasitas *covertext* PDAC ialah n/4, hal ini diartikan bahwa pada setiap 1 karakter *covertext* mampu mengenkripsi sebanyak 4 karakter. Setelah *covertext* dikonversi menjadi kode ASCII untuk membangkitkan kunci enkripsi diperlukan proses penghitungan matematika SUM (penjumlahan) antara 2-digit angka pada kode ASCII dan SUB (pengurangan) antara 2-digit angka pada kode ASCII. Hasil dari SUM dan SUB masing-masing ditambah 10 untuk menghasilkan kunci enkripsi. Proses enkripsi PDAC menggunakan operasi XOR antar tiap digit karakter *plaintext* dengan kunci. Hasil akhir berupa *ciphertext* diperoleh dari penggabungan *covertext* dengan hasil enkripsi (Kataria, Singh, Kumar, & Shekhawat, 2013). Gambar 1 memperlihatkan proses enkripsi PDAC.



Gambar 1. Ilustrasi Proses Enkripsi PDAC.



Gambar 2. Ilustrasi Proses Enkripsi PDAC dengan jumlah covertext yang tidak sesuai.

Gambar 1 memperlihatkan proses enkripsi PDAC dengan panjang *plaintext* 4 karakter (KODE), sehingga hanya diperlukan *covertext* sepanjang 1 karakter (R). pada proses ini setiap karakter *plaintext* dapat terenkripsi dan menghasilkan *ciphertext* (R__TQ). Hal ini jumlah kebutuhan *covertext* telah sesuai dengan kapasitas *covertext* yaitu n/4.

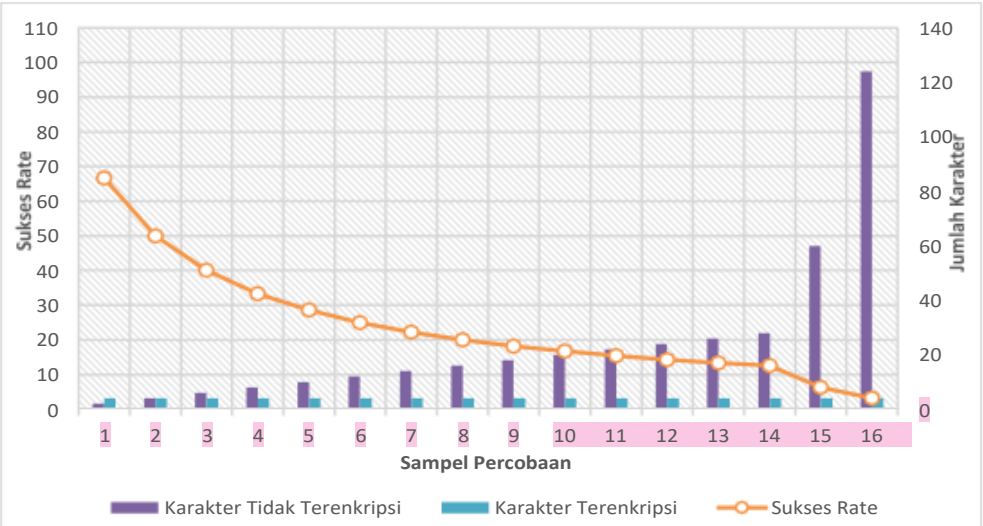
PDAC telah mengalami evolusi. Salah satu evolusi PDAC ialah pada bagian peningkatakn kapasitas covertextnya. Gaur (Gaur & Sharma, 2015) mengembangkan PDAC menjadi New PDAC. Model ini melakukan perbaikan kapasitas *covertext* yang semula n/4 menjadi n/6. Angka n/6 diartikan bahwa setiap 1 *covertext* mampu digunakan untuk memproses maksimal 6 karakter plaintext. Keuntungan lain yang diperoleh ialah terdapat penurunan ukuran file *ciphertext* yang dihasilkan. Pengembangan kapasits *covertext* juga dilakukan oleh Handoko (Handoko, Ardhianto, & Supriyanto, 2020) yang mampu meningkatkan kapasitas *covertext* hingga n/8, hal ini berarti bahwa setiap 8 karakter plaintext hanya diperlukan 1 *covertext* saja.

Meskipun PDAC telah mengalami perubahan untuk meningkatkan kapasitas, namun jumlah kebutuhan *covertext* harus disesuaikan dengan jumlah panjang plaintext. Jika kebutuhan *covertext* tidak sesuai dengan panjang *plaintext*, maka proses enkripsi tidak berjalan dengan sempurna. Dari gambar 1 dapat dilihat bahwa 4 karakter *plaintext* membutuhkan 1 karakter *covertext*, ini sesuai dengan kapasitas *covertext* n/4. Jika terdapat 6 karakter *plaintext*, maka diperlukan 2 *covertext*, karena 6/4 = 1,5 ≈ 2 karakter. Jika *covertext* yang diberikan hanya 1 karakter, maka terdapat 2 karakter yang tidak diproses yaitu D dan E. Gambar 2 memberikan ilustrasi proses enkripsi dengan jumlah *covertext* yang tidak sesuai dengan panjang *plaintext*nya.

Gambar 1 dan gambar 2 menunjukkan perbedaan proses yang terjadi dalam PDAC. Pada gambar 2 terlihat celah yang terjadi PDAC yaitu saat kebutuhan *covertext* kurang dari n/4 maka masih terdapat beberapa karakter *plaintext* yang tidak diproses enkripsi sehingga proses enkripsi tidak dapat berjalan dengan sempurna.

Gambar 3 memperlihatkan hasil percobaan awal, dengan membandingkan 16 sampel dengan jumlah karakter yang berbeda. Setiap sampel dicoba menggunakan 1 covertext. hasil yang diperoleh ialah bahwa semakin besar karakter yang diproses maka sukses rate dari proses enkripsi menggunakan PDAC akan semakin rendah. Hal ini berarti akan semakin banyak karakter yang tidak dapat di proses. Sepertihalnya pada gambar 2 yang terdapat 2 karakter yang tdiak terproses enkripsi. Penurunan sukses rate ini disebabkan penggunaan covertext yang tidak sesuai dengan jumlah minimal dari kebutuhan covertext seharusnya, sehingga menimbulkan ketidak sesuaian dalam proses enkripsinya.

Artikel ini membahas tentang bagaimana menyelesaikan permasalahan jika dalam pengamanan informasi menggunakan PDAC terdapat *covertext* yang jumlahnya kurang dari n/4 sehingga proses pengamanan informasi dapat



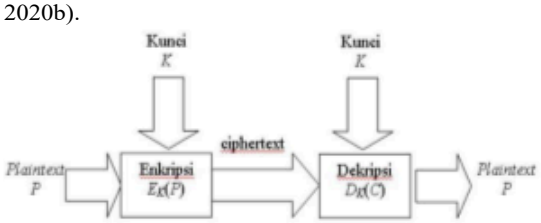
Gambar 3. Hasil Percobaan Awal.

dilakukan pada karakter *plaintext* secara menyeluruh.

2. Landasan Teori

2.1 Kriptografi

Kriptografi adalah sebuah bidang keilmuan yang bertujuan untuk mengamankan informasi dengan cara membuat informasi tersebut menjadi sulit dilihat dengan bantuan sebuah password atau kunci (Sadkhan & Salman, 2018). Kriptografi berasal dari kata *kryptos* yang bermakna ”rahasia” dan *graphein*, “menulis” (Ardhianto et al., 2020b; Telaumbanua & Zebua, 2020). Kriptografi mengamankan data dengan mengubah data menjadi bentuk lain yang tidak berarti (Ardhianto et al., 2020b).



Gambar 4. Skema enkripsi dekripsi pada

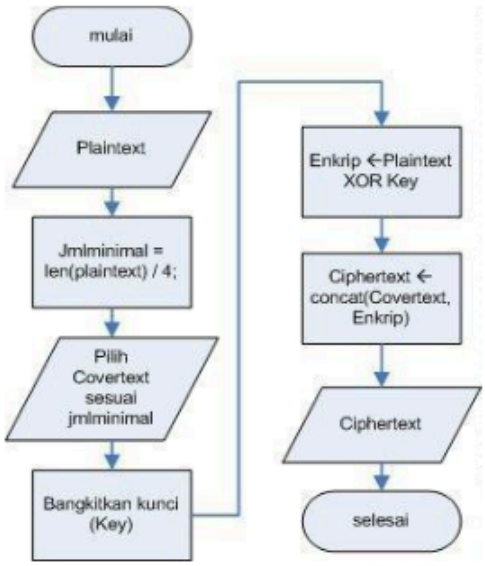
Teknik kriptografi diartikan sebagai cara menyembunyikan pesan dengan menyamarkan atau mengacak pesan yang memiliki arti lain atau membuat pesan asli menjadi tak berarti. Proses kriptografi terdapat 2 macam, yaitu: enkripsi dan dekirpsi. Gambar 4 memberikan ilustrasi proses enkripsi dan dekripsi pada kriptografi. Proses untuk mengubah dan mengacak naskah asli menjadi naskah yang tidak dikenali disebut enkripsi, sedangkan dekripsi adalah sebuah proses mengembalikan naskah yang tidak dikenali menjadi naskah asli.

Gambar 5. Skema Steganografi.

Kedua proses utama kriptografi tersebut membutuhkan kunci untuk mengacak dan mengembalikan informasi. Sehingga ketika seseorang yang menerima data atau informasi tersebut tidak memiliki kunci, akan membutuhkan waktu yang sangat lama untuk mengembalikan ke bentuk asli dan kemungkinan tidak bisa dikembalikan sama sekali.

2.2 Steganografi

Teknik steganografi berbeda dengan kriptografi yaitu pengamanan data dengan memanfaatkan data lain yang disebut cover yang berfungsi untuk menyembunyikan data asli di dalamnya (Ardhianto et al., 2020b). Kelebihan steganografi adalah orang yang tidak berhak tidak menyadari keberadaan sebuah pesan (Handoko, Ardhianto, Hadiono, et al., 2020). Gambar 5 memperlihatkan ilustrasi proses setagnografi. Pesan akan diamankan dengan menggunakan cover sebagai media penyembunyian melalui proses penyisipan.



Gambar 6. Proses Enkripsi PDAC

Cover yang digunakan dapat berupa teks, gambar, file suara, video atau transmisi radio. Pada proses penyisipan terdapat berbagai macam cara, salah satu yang terkenal ialah teknik LSB (*Last Significant Bit*). Teknik ini menggunakan nilai bit terakhir sebagai tempat penyembunyian. Selain itu ada teknik penyisipan yang menggunakan kunci sebagai prosedur keamanannya. Hasil proses penyisipan dikenal sebagai *stego image*. Proses pengambilan pesan dari stego image dikenal sebagai proses ekstraksi. Hasil yang diperoleh kemudian disatukan kembali sebagai pesan yang utuh dan dapat terbaca kembali.

2.3 *Literatur Review* Model Enkripsi PDAC

Model enkripsi *Parallel Encryption with Digit Arithmetic of Covertext* (PDAC) adalah model enkripsi yang melakukan pemrosesan steganografi berbasis teks yang dikombinasikan dengan keamanan data melalui proses enkripsi (Kataria, Singh, Kumar, & Shekhawat, 2013). PDAC merupakan pengembangan dari model *Encryption with Covertext and Reordering* (ECR) (Kataria, Singh, Kumar, & Nehra, 2013). PDAC menawarkan bentuk komputasi sederhana dari keamanan data berbasis teks dan waktu penyelesaian yang cepat. PDAC memiliki dua proses, encoding, dan decoding. Encoding digunakan untuk mengubah karakter plaintext menjadi ciphertext, sedangkan decoding adalah kebalikan dari proses encoding. PDAC memiliki beberapa istilah, Plaintext sebagai dokumen teks yang akan diamankan, Encrypted text sebagai dokumen teks yang diamankan, Encryption key sebagai kunci yang digunakan dalam proses enkripsi, dan Covertext adalah karakter yang digunakan sebagai media persembunyian dalam konteks steganografi (Kataria, Kumar, et al., 2013).

Covertext dipilih secara acak *n*/4 dari plaintext, *n* adalah jumlah karakter. Hal ini dikarenakan setiap covertext akan digunakan untuk mengenkripsi 4 plaintext. Kunci enkripsi diperoleh dari operasi penjumlahan dan pengurangan digit plaintext dalam kode ASCII. Setiap covertext akan menghasilkan dua kunci enkripsi. XOR beroperasi pada teks biasa dan kunci enkripsi untuk membentuk teks terenkripsi. PDAC *Encryptedtext* terdiri dari plaintext dan covertext terenkripsi. Gambar 6 memperlihatkan proses enkripsi pada PDAC.

3. Metode

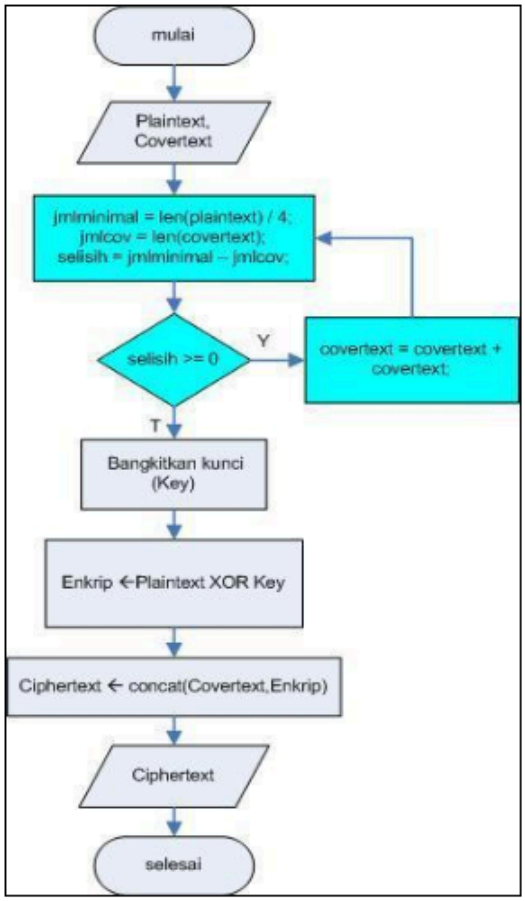
Celah pada proses enkripsi PDAC yang ditemukan yaitu saat jumlah karakter *covertext* tidak memenuhi aturan n/4 jumlah karakter *plaintext* yang diperlukan seperti terlihat pada gambar 2. Untuk memperbaiki celah yang ditemukan, diperlukan cara teknik pengulangan *covertext* diusulkan dilakukan secara berulang seperti penggunaan kunci pada algoritma enkripsi vigenere (Nofiyanto et al., 2014; Qowi & Hudallah, 2021; Telaumbanua & Zebua, 2020). Gambar 6 menunjukkan flowchart modifikasi PDAC dengan menggunakan proses pengulangan kunci (*key*).

Gambar 7 menampilkan flowchart modifikasi model enkripsi PDAC dengan mengadopsi teknik enkripsi Algoritma Vigenere dengan penggulangan kunci (*key*) sesuai dengan jumlah karakter *plaintext*. Adopsi ini ditempatkan dengan melihat selisih kebutuhan minimal *covertext* dengan jumlah *covertext* yang diinputkan. Model enkripsi PDAC memerlukan covertext sepanjang minimal 25% dari panjang *plaintext*nya. Jika nilai selisih belum mencapai nol atau kurang dari nol, maka *covertext* diulang dan disambungkan hingga memenuhi panjang kebutuhan minimal. Dengan demikian pemilik pesan tidak perlu menginputkan ulang *covertext.*

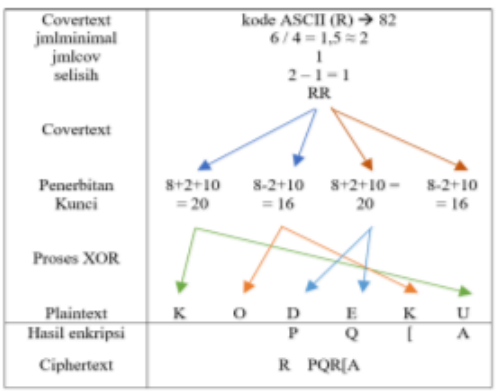
Proses pembangkitan kunci mengikuti proses standar sesuai pada PDAC. Pembangkitan kunci ini membutuhkan *covertext* sebagai inputan yang kemudian dilakukan operasi pejumlahan dan pengurangan antar digit kode ASCII masing masing *covertext.* Hasil yang diperoleh dalam bentuk angka desimal digunakan sebagai kunci (*key*) pada proses enkripsi.

Proses enkripsi dilakukan menggunakan operator XOR antara kunci (*key*) dengan karakter *plaintext* yang disesuaikan dengan alur panah yang telah ditentukan. Operasi ini sesuai dengan proses enkripsi PDAC versi sebelumnya. Hasil enkripsi yang diperoleh digabung dengan *covertext* dan menghasilkan *ciphertext.*

4. Hasil dan Pembahasan



Gambar 7. Proses Enkripsi PDAC dengan pengulangan covertext.



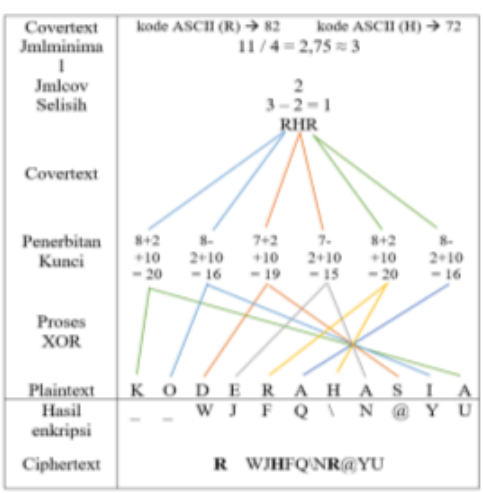
Gambar 8. Proses Enkripsi PDAC (sampel 1).

Eksperimen ini melakukan percobaan menggunakan 2 teks sebagai contoh *plaintext* dan *covertext* terlihat pada tabel 1. Proses yang dilakukan ialah enkripsi dan dekripsi menggunakan model PDAC yang diusulkan. Proses enkripsi dilakukan sesuai dengan alur yang terlihat pada gambar 3. Proses dekripsi dalam eksperimen ini sesuai dengan proses dekripsi pada model PDAC versi awal. Gambar 7 memperlihatkan proses enkripsi dari *plaintext* 1 dan gambar 5 memperlihatkan proses enkripsi dari *plaintext* 2.

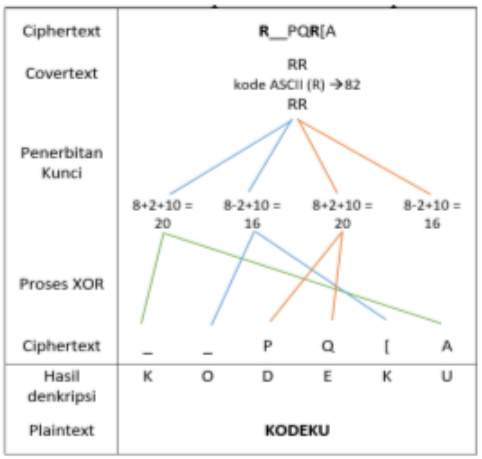
Tabel 1. Proses Enkripsi PDAC

	Plaintext 1	*Plaintext 2*
Plaintext	*KODEKU*	*KODERAHASIA*
Covertext	*R*	RH

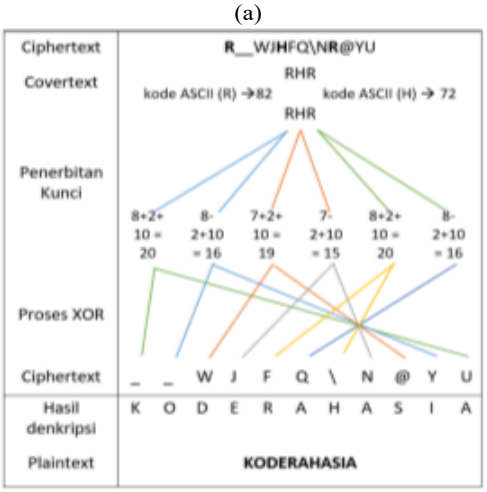
Enkripsi pada gambar 8 dilakukan menggunakan sampel *plaintext* 1. Pertama, jumlah kebutuhan minimal *covertext* dihitung, dalam contoh dibulatkan keatas, yaitu 1,5 ≈ 2. Sehingga *covertext* R digandakan sejumah selisih kebutuhan minimal *covertext* dengan *covertext* yang diinputkan pengirim, sehingga panjang *covertext* terpenuhi 2 karakter (RR). Proses penerbitan kunci dilakukan dengan mengkonversi karakter *covertext* (RR) menjadi kode ASCII (82). Operasi penjumlahan dan pengurangan pada digit kode ASCII dilakukan untuk mendapatkan kunci enkripsi. Penambahan angka 10 digunakan untuk menghindari hasil negatif. Enkripsi antara *plaintext* dan kunci enkripsi diproses menggunakan XOR. Kunci pertama mengenkripsi karakter *plaintext* ke-1 dan terakhir (*n*), kunci kedua mengenkripsi karakter kedua-2 dan karakter ke-2 dari akhir (n-1), dan seterusnya. Terakhir, penyisipan karakter *covertext* pada jeda setiap 4 karakter hasil enkripsi, sebagai *ciphertext*. Pada proses ini *covertext* awal (R) dengan panjang 1 karakter disesuaikan dengan kebutuhan minimal yakni 2 karakter, sehingga *covertext* menjadi (RR). Pada percobaan ini karakter *covertext* pertama memproses 4 karakter dan *covertext* kedua memproses 2 karakter.



Gambar 9. Proses Enkripsi PDAC (sampel 2).



(a)



(b)

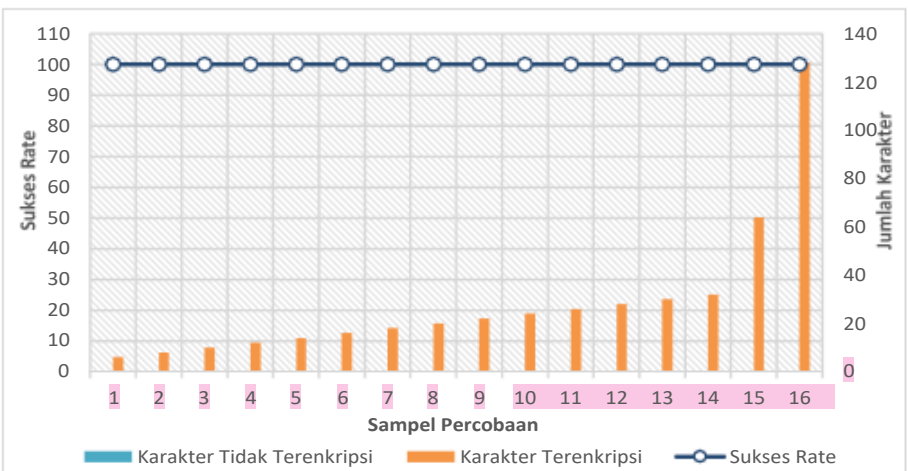
Gambar 10 (a,b). Dekripsi PDAC Sampel 1 dan Sampel 2.

Sampel *plaintext* 2 divisualkan seperti pada gambar 9. Kebutuhan minimal *covertext* ialah 2,75 ≈ 3 karakter. Panjang *covertext* yang diberikan pengirim 2 karakter (RH), sehingga disesuaikan menjadi 3 karakter (RHR). Proses penerbitan kunci menggunakan operasi aritmatika penjumlahan dan pengurangan digit kode ASCII *covertext*. Proses enkripsi dilakukan menggunakan operator XOR. Proses XOR mengikuti aturan yang sama seperti PDAC versi awal. Untuk membentuk *ciphertext*, *covertext* disisipkan dalam hasil enkripsi dengan aturan setiap 4 karakter. Pada proses ini kunci ke-1 sampai kunci ke-5 memproses 2 karakter *plaintext* dan kunci ke-6 memproses 1 karakter *plaintext.*

Gambar 10(a) dan 9(b) memperlihatkan proses dekripsi. *Covertext* diambil dari ciphertext pada setiap 4 urutan karakter, karakter ke-0, ke-5, ke-10, dan seterusnya. Proses penerbitan kunci dilakukan dengan operasi penjumlahan dan pengurangan antar digit kode ASCII *covertext*. Dilanjukan dengan proses dekripsi menggunakan operator XOR. Hasil yang diperoleh ialah informasi yang diamankan (*plaintext*).

Gambar 11 memperlihatkan hasil percobaan terhadap 16 sampel. Hasil yang diperoleh adalah bahwa seluruh teks sampel dapat diproses dengan baik, sehingga sukses rate yang diperoleh mencapai 100%. Hal ini berarti dalam percobaan tidak terdapat karakter yang gagal diproses menggunakan PDAC, sehingga dalam gambar 11 jumlah karakter tidak terenkripsi adalah nol (0).

Dengan demikian, usulan untuk mengadopsi proses pengulangan *covertext* pada PDAC memberikan keuntungan dalam proses pengamanan informasi saat jumlah *covertext* yang diinputkan pengirim tidak memenuhi jumlah minimal kebutuhan *covertext* yang diperlukan untuk proses enkripsi PDAC. Dengan demikian pengamanan menggunakan PDAC dapat berjalan secara keseluruhan.



Gambar 11. Hasil Percobaan menggunakan perulangan covertext PDAC.

5. Kesimpulan

Berdasarkan hasil eksperimen yang dilakukan, maka dapat disimpulkan bahwa untuk mengenkripsi karakter plaintext dengan jumlah yang tidak sesuai dengan ketentuan *covertext* yaitu n/4 dapat dilakukan dengan perulangan *covertext* untuk mengenkripsi seluruh plaintext, keuntungan lain yang didapatkan ialah pengamanan informasi saat jumlah *covertext* yang diinputkan pengirim tidak memenuhi jumlah minimal kebutuhan *covertext*, memudahkan user untuk dapat secara langsung menggunakan proses enkripsi tanpa perlu menghitung secara manual terlebih dahulu banyaknya karakter *covertext* yang perlu dipakai untuk mengenkripsi dan mempersingkat waktu enkripsi saat pengirim tergesa-gesa dalam melakukan pengamanan informasi.

Sebagai rencana pengembangan, model PDAC perlu dilakukan modifikasi dengan melihat aspek-aspek keamanan informasi diantaranya: kerahasiaan pesan, autentikasi dan integriti pesan, serta aspek lain dalam bidang steganografi.

Daftar Pustaka:

Ardhianto, E., Handoko, W. T., Murti, H., & Redjeki, R. S. A. (2021). Encryption with Covertext and Reordering using Permutated Table and Random Function. *2021 2nd International Conference on Innovative and Creative Information Technology, ICITech 2021*. https://doi.org/10.1109/ICITech50181.2021.9590171

Ardhianto, E., Trisetyarso, A., Suparta, W., Abbas, B. S., & Kang, C. H. (2020a). Design Securing Online Payment Transactions Using Stegblock through Network Layers. *IOP Conference Series: Materials Science and Engineering*, *879*(1). https://doi.org/10.1088/1757-899X/879/1/012027

Ardhianto, E., Trisetyarso, A., Suparta, W., Abbas, B. S., & Kang, C. H. (2020b). Design Securing Online Payment Transactions Using Stegblock through Network Layers. *IOP Conference Series: Materials Science and Engineering*, *879*(1). https://doi.org/10.1088/1757-899X/879/1/012027

Gaur, M., & Sharma, M. (2015). A New PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography Approach for Cloud Data Security. *International Journal on Recent and Innovation Trends in Computing and Communication* , *3*(3), 1344–1352. http://www.ijritcc.org

Handoko, W. T., Ardhianto, E., Hadiono, K., & Sutanto, F. A. (2020). Protecting Data by Socket Programming Steganography. *IOP Conference Series: Materials Science and Engineering*, *879*(1). https://doi.org/10.1088/1757-899X/879/1/012028

Handoko, W. T., Ardhianto, E., & Supriyanto, E. (2020). MODIFIKASI NEW PDAC (PARALLEL ENCRYPTION WITH DIGIT ARITHMETIC OF COVER TEXT). *SENDIU 2020*, 55–59.

Kataria, S., Kumar, T., Singh, K., & Nehra, M. S. (2013). ECR (encryption with cover text and reordering) based text steganography. *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, 612–616. https://doi.org/10.1109/ICIIP.2013.6707666

Kataria, S., Singh, B., Kumar, T., & Shekhawat, H. S. (2013). PDAC (Parallel Encryption with Digit Arithmetic of Cover Text) Based Text Steganography. *Int. Conf. on Advances in Computer Science, AETACS*, 175–182.

Kataria, S., Singh, K., Kumar, T., & Nehra, M. S. (2013). ECR(Encryption with Cover Text and Reordering) based Text Steganography. *IEEE Second International Conference on Image Information Processing (ICIIP-2013)*.

N, M. K., Jayaramu, H. S., Kurian, M. Z., & Shiva kumar, K. B. (2007). FPGA Implementation of Vigenere Cipher Method Based on Colour Image Steganography. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO*, *3*(4), 9051–9057. www.ijareeie.com

Nahar, K., & Chakraborty, P. (2020). A Modified Version of Vigenere Cipher using 95×95 Table. *International Journal of Engineering & Advanced Technology (IJEAT)*, *9*(5), 1144–1148. https://doi.org/10.35940/ijeat.E9941.069520

Nofiyanto, N., Hamzah, hamzah, & Surbakti, H. (2014). SHORT MESSAGE ENCRYPTION APPLICATION DEVELOPMENT USING VIGENERE ALGORITHM UTILIZING EULER'S NUMBER ON ANDROID SMARTPHONE. *Jurnal Teknologi Informasi* , *9*(27), 81–92.

Qowi, Z., & Hudallah, N. (2021). Combining caesar cipher and hill cipher in the generating encryption key on the vigenere cipher algorithm. *Journal of Physics: Conference Series*, *1918*(4), 1–6. https://doi.org/10.1088/1742-6596/1918/4/042009

Sadkhan, S. B., & Salman, A. O. (2018). Fuzzy Logic for Performance Analysis of AES and Lightweight AES. *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 318–323. https://doi.org/10.1109/ICOASE.2018.8548832

Telaumbanua, F., & Zebua, T. (2020). Modifikasi Vigenere Cipher Dengan Pembangkit Kunci Blum Blum Shub. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, *4*(1). https://doi.org/10.30865/komik.v4i1.2646