

# OPTIMASI KENDALI SISTEM KEAMANAN DATA DENGAN ALGORITMA CRC (*CYCLIC REDUNDANCY CHECK*)

Antoni<sup>1)</sup>, Solly Aryza<sup>2)</sup>

<sup>1)</sup>Fakultas Teknik, Teknik Informatika, Universitas Islam Sumatera Utara, Medan, Indonesia

<sup>2)</sup>Staf Pengajar Fakultas Sains dan Teknologi, Universitas Pembangunan Panca Budi, Sumatera Utara, Medan, Indonesia

[antmunthe@gmail.com](mailto:antmunthe@gmail.com); [sollyaryzalubis@gmail.com](mailto:sollyaryzalubis@gmail.com)

## Abstrak

Keaslian dan merupakan salah satu faktor yang penting dalam komunikasi data, apalagi membuat rancangan implementasi program yang bertugas untuk pengamanan data merupakan salah satu solusi untuk mengetahui keaslian data tersebut. Sistem aplikasi ini dirancang dengan menggunakan Microsoft Visual Studio 2010, WampServer 5.0, Connector ODBC 5.3. Sistem aplikasi ini dibangun dengan menggunakan algoritma CRC (*Cyclic Redundancy Check*) untuk menentukan checksum file dengan demikian, penulis berharap pengguna dapat mengetahui apakah file yang diterima telah dimodifikasi dalam proses transmisinya. Untuk menghindari akses oleh pihak yang tidak berhak. Proses pengiriman data sering kali memiliki resiko terjadi perubahan yang tidak diinginkan terhadap data, demi menjaga keaslian dan keamanan data yang dikirimkan kita dapat menggunakan algoritma untuk mendeteksi keaslian ataupun kerusakan data dalam proses transmisi data.

**Kata-Kata Kunci:** Keaslian Data, Crc (*Cyclic Redundancy Check*), Checksum File

## I. PENDAHULUAN

Keaslian dan Keamanan merupakan salah satu faktor yang penting dalam komunikasi data, apalagi untuk menghindari akses oleh pihak yang tidak berhak (Handy Wicaksono et al., 2008). Proses pengiriman ataupun penyimpanan data sering kali mempunyai resiko terjadi perubahan yang tidak diinginkan terhadap data. Untuk menjaga keaslian dan keamanan data kita dapat menggunakan algoritma yang dikembangkan untuk mendeteksi keaslian ataupun kerusakan data dalam proses transmisi data maupun penyimpanan data (Aryza et al., 2017).

Kita juga dapat menggunakan proses authentication pada informasi yang kita dapat agar kita mengetahui keaslian dari informasi tersebut. Berdasarkan masalah diatas, maka penulis mencoba membuat rancangan implementasi untuk pengamanan data dengan algoritma untuk memeriksa keaslian data dengan menggunakan algoritma CRC 32 sehingga memungkinkan seseorang yang ingin memeriksa keaslian data atau informasi penting yang telah diterima dari orang lain dapat dilakukan dengan akurat.

Banyak fungsi algoritma hash yang dapat kita gunakan untuk proses mencari keaslian dan keamanan file. Disini penulis mengangkat salah satunya fungsi algoritma yang biasa digunakan yaitu CRC (*Cyclic Redundancy Check*) (Aryza et al., 2017).

## II. TINJAUAN PUSTAKA

### 2.1 Dokumen Digital

Dokumen digital merupakan setiap informasi elektronik yang dibuat, diteruskan, dikirimkan, diterima, atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal, atau sejenisnya, yang dapat

dilihat, ditampilkan dan/atau didengar melalui komputer atau sistem elektronik, tetapi tidak terbatas pada tulisan, suara atau gambar, peta, rancangan, foto atau sejenisnya, huruf, tanda, angka, kode akses, simbol atau perforasi yang memiliki makna atau arti dan dapat dipahami oleh orang banyak (Setiadi, R. & Wulandari, F., 2016)

### 2.2 Pengolahan Dokumen Digital

Untuk mengolah suatu dokumen digital, dibutuhkan perangkat lunak (*software*) khusus yang sering disebut dengan perangkat lunak pengolah kata (*Word Processor*). Perangkat lunak pengolah kata adalah suatu aplikasi komputer yang digunakan untuk menyusun, menyunting, memformat dan mencetak segala jenis bahan yang dapat dicetak (Setiadi, R. & Wulandari, F., 2016)

### 2.3 Algoritma

Algoritma adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah,

a. Algoritma haruslah benar. Artinya algoritma akan memberikankeluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidakpeduli sebgas apapun algoritma, kalau memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik (Solly Aryza, Hermansyah, Muhammad Irwanto, Zulkarnain Lubis, 2017).

b. Seberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini penting terutama pada algoritma untuk menyelesaikan masalah yang memerlukan aproksimasi hasil (hasil yang hanya berupa pendekatan). Algoritma yang baik harus mampu memberikan hasil yang sedekat mungkin dengan nilai yang sebenarnya.

c. Efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari 2 hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar (paling mendekati), tetapi jika kita harus menunggu berjam-jam untuk mendapatkan keluarannya, algoritma tersebut biasanya tidak akan dipakai, setiap orang menginginkan keluaran yang cepat. Begitu juga dengan memori, semakin besar memori yang terpakai maka semakin buruklah algoritma tersebut (Siahaan et al., 2018).

2.4 Checksum

Checksum adalah teknologi untuk menandai sebuah file, dimana setiap file yang sama harus memiliki checksum yang sama, dan bila nilai checksumnya berbeda meskipun satu bit saja, maka file tersebut merupakan file yang berbeda walaupun memiliki nama file yang sama (Schlegel et al., 2009).

2.5 CRC 32 (Cyclic Redundancy Check 32)

CRC(Cyclic Redundancy Check) adalah algoritma untuk memastikan integritas data dan mengecek kesalahan pada suatu data yang akan ditransmisikan atau disimpan. Data yang hendak ditransmisikan atau disimpan ke sebuah media penyimpanan rentan sekali mengalami kesalahan

2.6 Alat Bantu Perancangan Sistem

Adapun alat bantu dalam perancangan pembangunan aplikasi ini penulis memakai metode Unified Modelling Language (UML). Unified Modelling Language (UML) adalah penerus dari object-oriented analysis and design (OOA&D)

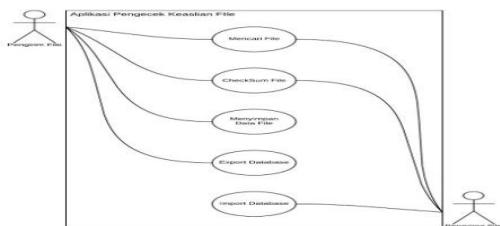
III. METODELOGI PENELITIAN

3.1 Rancangan Program

Perancangan sistem dimulai dengan perancangan Algoritma CRC32, Use case Diagram, Use case Skenario, Sequence Diagram dan Activity Diagram

3.2 Rancangan Use case Diagram

Use case diagram, dalam perancangan program aplikasi yang akan dibangun memiliki tujuan untuk menggambarkan hubungan sejumlah external aktor dengan use case yang terdapat didalam sistem. Use case diagram ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar oleh aktor. Dalam perancangan use case diagram terdapat dua aktor, yaitu pengirim file dan penerima file



Gambar 1. Diagram Use case Aplikasi Pengecek Keaslian File

3.2.1 Rancangan Use case Skenario

Skenario Use case bertujuan untuk menjelaskan masing-masing deskripsi setiap use case yang ada pada rancangan Use case diagram, serta interaksi antara use case.

Tabel 1. Deskripsi Use case Diagram

NO	Kode	Nama Use case	Keterangan
1.	UC – P01	Mencari File	Proses mencari file yang akan diperiksa keasliannya
2.	UC – P02	Checksum File	Proses menghitung dan menentukan keaslian file
3.	UC – P03	Menyimpan Data File	Menyimpan nilai sum file yang sudah di check
4.	UC – P04	Export Database	Memindahkan record database menjadi file Excel
5.	UC – P05	Import Database	Menyalin record database dari file Excel menuju database aplikasi

Tabel 2. Skenario : Checksum File

Nama Use Case	Checksum File
Kode	UC – P02
Aktor	Pengirim File, Penerima File
Kondisi Awal	User telah menemukan file yang akan diperiksa
Reaksi Aktor	Reaksi Sistem
1. User memilih tombol check file	2. Sistem akan menampilkan nilai Sum file, memeriksa nilai berubah.
Kondisi Akhir	Menampilkan nilai sum dan hasil pengecekan
Pengecualian	Data belum tersimpan atau database belum di import

Tabel 3. Skenario : Menyimpan File

Nama Use Case	Menyimpan Data File
Kode	UC – P03
Aktor	Pengirim File
Kondisi Awal	Menampilkan nilai sum dan hasil pengecekan
Reaksi Aktor	Reaksi Sistem
1. User memilih tombol save result	2. Sistem akan menyimpan data kedalam database.
Kondisi Akhir	Data sudah tersimpan kedalam database
Pengecualian	Sistem tidak menghasilkan nilai sum dan hasil pengecekan

Tabel 4. Skenario : Export Database

Nama Use Case	Export Database
Kode	UC – P04
Aktor	Pengirim File
Kondisi Awal	Data Sudah tersimpan kedalam database
Reaksi Aktor	Reaksi Sistem
1. User membuka halaman database	2. Menampilkan halaman database
3. Memilih tombol Export database	3. Memindahkan record database kehalaman kerja Excel
4. Menyimpan halaman Excel yang berisi record database	
Kondisi Akhir	Database sudah berbentuk file Excel
Pengecualian	Komputer tidak terdapat aplikasi Excel

**Tabel 5. Skenario : *ImportDatabase***

<b>Nama Use Case</b>	<i>Import Database</i>	
<b>Kode</b>	UC – P05	
<b>Aktor</b>	Penerima <i>File</i>	
<b>Kondisi Awal</b>	<i>Database</i> sudah berbentuk <i>file Excel</i>	
<b>Reaksi Aktor</b>	<b>Reaksi Sistem</b>	
1. User membuka halaman <i>database</i>	2. Menampilkan halaman <i>database</i>	
3. User memilih tombol Browser	4. Menampilkan <i>drive</i> komputer	
5. Mencari <i>file database</i>	6. Menampilkan isi <i>record file database</i>	
5. User memilih tombol <i>import</i>		
<b>Kondisi Akhir</b>	<i>Record database</i> sudah masuk kedalam <i>database</i> aplikasi	
<b>Pengecualian</b>	<i>File database</i> tidak ada	

**3.2.2 Rancangan Sequence Diagram**

*Sequence* Diagram merupakan gambaran dari interaksi objek yang disusun berdasarkan urutan waktu atau dapat disebut *sequence* diagram adalah gambaran tahap demi tahap, termasuk urutan perubahan secara logis.

**3.2.3 Rancangan Activity Use case Diagram**

*Activity* Diagram adalah rangkaian aliran data dari aktifitas *use case* diagram. Dengan kata lain *Activity* diagram menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang.

**3.2.4 Rancangan Database**

Perancangan *Database* adalah sebuah proses untuk menentukan isi dan pengaturan daya yang dibutuhkan untuk mendukung berbagai rancangan sistem. Dalam perancangan sistem ini penulis memakai database *MySQL*, seperti yang kita ketahui *MySql* merupakan sebuah *database* bersifat *open source*

**3.3 Rancangan Tampilan Antar Muka**

Perancangan antar muka merupakan langkah awal dalam membuat program aplikasi. Program harus dirancang sesuai dengan kebutuhan pengguna nantinya.

**3.3.1 Perancangan Tampilan Halaman Utama**

Perancangan halaman menu utama merupakan tampilan pertama kali muncul ketika program aplikasi dijalankan. Halaman utama dirancang untuk mencari *file* yang ingin di check keasliannya serta menginput data kedalam *database*

**3.3.2 Perancangan Tampilan Halaman Database**

Perancangan halaman *database*, halaman *database* dirancang untuk proses *import*, *exportdatabase* serta untuk menghapus *record* data

**IV. PEMBAHASAN DAN HASIL**

**4.1 Tampilan Program**

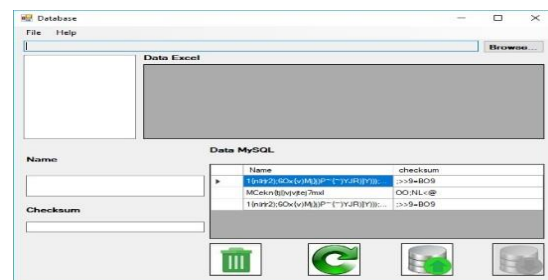
Pada bagian ini penulis akan menampilkan bagian – bagian dari program baik halaman utama, *database* secara detail sehingga pengguna dapat memahami tujuan dari dibuatnya halaman tersebut, berikut ini tampilan halaman program.

**4.2. Tampilan Halaman Utama**

Halaman ini merupakan tampilan pertama kali muncul ketika program aplikasi dijalankan. Halaman utama dirancang untuk mencari *File* yang ingin di check keasliannya serta menginput data kedalam *database*, berikut ini tampilan halaman utama.

1)Tampilan Halaman *Database*

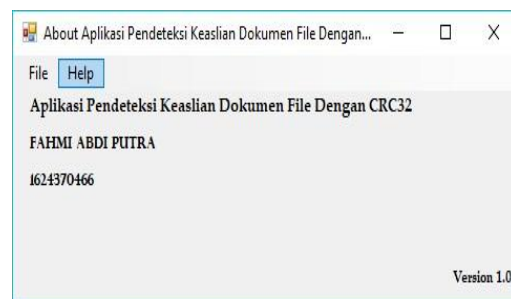
Halaman *database*, halaman *database* dirancang untuk proses *import*, *exportdatabase* serta untuk menghapus *record* data, berikut ini tampilan halaman *database*.



**Gambar 2. Halaman Database Aplikasi**

2) Tampilan Halaman About

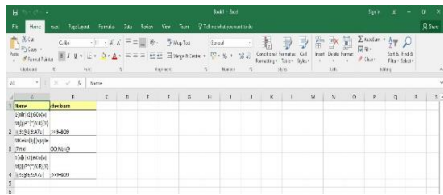
Halaman About, berisi tentang versi dari aplikasi serta nama pembuat aplikasi tersebut. Berikut ini tampilan halaman about.



**Gambar 3. Halaman About**

3)Tampilan Halaman *Export*

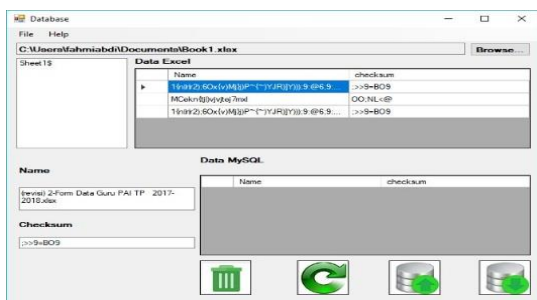
*Database* Ke *MS.Excel* Pada Halaman ini lembar kerja dari *MS.Excel* berisi dari *record* (data) dari *databaseFile* yang akan dikirim, adapun isi dari *record* diacak sehingga pengguna lain tidak dapat mengetahui nilai dari *record* tersebut.



Gambar 4. Halaman *Export Database* ke *MS.Excel*

4) Tampilan Halaman *ImportDatabase*

Pada halaman *Import Database* memperlihatkan isi dari *File* yang akan di import menjadi *database*, adapun nilai yang dapat dilihat hanya nama dari *File* tersebut, hal itu sudah cukup untuk mengetahui apakah *File* tersebut adalah *database* yang dimaksud pengirim.



Gambar 5. Halaman *Export Database*

V. KESIMPULAN

Berdasarkan hasil pengujian aplikasi Pemeriksa Keaslian *File* dibuat kesimpulan sebagai berikut :

1. Aplikasi yang telah dibangun dapat mendeteksi keaslian file dari modifikasi pihak tertentu.
2. Aplikasi dapat meminimalisir pengguna (*user*) terinfeksi malware dari file yang diterima karena telah dimodifikasi oleh pihak yang tidak bertanggung jawab.

DAFTAR PUSTAKA

- [1]. Aryza, S., Irwanto, M., Lubis, Z., Putera, A., & Siahaan, U. (2017). A Novelty Stability Of Electrical System Single Machine Based Runge Kutta Orde 4 Method. *IOSR Journal of Electrical and Electronics Engineering Ver. II*, 12(4), 2278–1676. <https://doi.org/10.9790/1676-1204025560>
- [2]. Handy Wicaksono, Resmana Lim, & William Sutanto. (2008). Perancangan SCADA Software dengan Wonderware InTouch Recipe Manager dan SQL Access Manager pada Simulator Proses Pencampuran Bahan. *Jurnal Teknik Elektro*, 8(1), 38–45. <https://doi.org/10.9744/jte.8.1.38-45>
- [3]. Schlegel, P. N., Fauser, B. C., Carrell, D. T., & Racowsky, C. (2009). *Biennial Review of Infertility* (Vol. 3). <https://doi.org/10.1007/978-1-60327-392-3>
- [4]. Setiadi, R., & Wulandari, F. (2016). Memadukan Strategi, Mewujudkan Ketahanan: Sebuah Pembelajaran dari Pengembangan Strategi Ketahanan Kota di Semarang. *Jurnal Pengembangan Kota, The Review of Urban Studies and Development*, 4(2), 95–105. <https://doi.org/10.14710/jpk.4.2>
- [5]. Siahaan, A. P. U., Aryza, S., Hariyanto, E., Rusiadi, Lubis, A. H., Ikhwan, A., & Kan, P. L. E. (2018). Combination of Levenshtein Distance and Rabin-Karp to Improve the Accuracy of Document Equivalence Level. *International Journal of Engineering & Technology*, 7(2.27), 17–21. <https://doi.org/10.14419/ijet.v7i2.27.12084>
- [6]. Solly Aryza, Hermansyah, Muhammad Irwanto, Zulkarnain Lubis, A. I. (2017). a Novelty of Quality Fertilizer Dryer Based on Solar Cell and Ann. *Scopus*, 1–5.