Implementasi Algoritma AES Dan Bcrypt untuk Pengamanan File Dokumen

Gebrina Divva Meuthia Zulma[1], Henki Bayu Seta[2], Trihastuti Yuniati[3]
Program Studi Informatika / Universitas Pembangunan Nasional Veteran Jakarta[1,2]
Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Banyumas, Indonesia[3]
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia[1,2]
gebgebdmz@gmail.com[1], henkiseta@upnvj.ac.id[2], trihastuti@ittelkom-pwt.ac.id[3]

Abstrak. Aplikasi Microsoft Office dan pdf saat ini menjadi hal yang sangat penting untuk bertukar informasi di kalangan masyarakat. Tetapi hal ini tidak dilakukan dengan adanya proteksi terhadap informasi dan data yang akan diberikan. Sedangkan pengamanan data adalah hal yang sangat penting, karena bila tidak hati-hati, data bisa dicuri dan digunakan oleh orang lain. Tujuan dari penelitian ini yaitu menciptakan sistem keamanan yang maksimal agar data tidak disalahgunakan, yaitu dengan menggunakan algoritma AES dan Bcrypt. Metode yang dilakukan adalah perancangan sistem menggunakan *framework* Laravel, algoritma kriptografi Bcrypt serta AES sebagai pengamanan *file*, dengan menggunakan sampel komputer 1 dan komputer 2. Hasil penelitian ini didapatkan beberapa hasil yaitu, komputer 1 memiliki waktu pemrosesan yang lebih baik dari komputer 2, dengan adanya peningkatan ronde pada Bcrypt, maka waktu pemrosesan akan meningkat sebesar 22,788% sampai 57,765%, waktu untuk melakukan hashing kunci lebih besar daripada saat cek kunci dan kombinasi *secret key* tidak berpengaruh pada waktu enkripsi dan dekripsi pada *file* pdf dan docx. Kesimpulan yang didapatkan adalah Laravel merupakan *framework* yang tepat, dimana *package* bernama FileVault dapat digunakan untuk mengenkripsi dan dekripsi *file* tipe .docx dan .pdf menggunakan AES. Penelitian ini diharapkan dapat menciptakan suatu *web* yang dapat digunakan untuk mengamankan *file* dokumen dengan AES dan Bcrypt.

Kata Kunci: AES, Bcypt, *file* dokumen, pengamanan

1 Pendahuluan

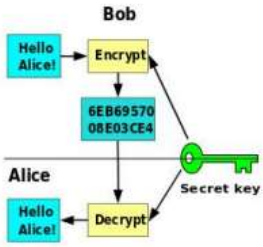
Penggunaan aplikasi seperti Microsoft Office dan pdf menjadi sangat penting untuk bertukar informasi di kalangan masyarakat, tetapi hal ini justru tidak dilakukan dengan proteksi terhadap informasi dan data yang akan diberikan. Selain itu, cara pengamanan data menjadi sangat penting karena bila tidak hati-hati, data bisa dicuri dan digunakan oleh orang lain. Sebagai contoh yaitu kasus kebobolan data di Tokopedia[1] dimana data *password* mereka yang di *hash* menggunakan MD5 sudah dipastikan telah dipecahkan *hash* nya karena MD5 yang sudah usang dan mudah dibobol, berbeda dengan SHA2-384 yang tergolong baru dan belum ditemukan cara untuk membobolnya. Tujuan dari penelitian ini yaitu menggunakan algoritma AES dan Bcrypt sehingga menciptakan sistem keamanan yang maksimal agar data tidak dapat disalahgunakan, dengan harapan penelitian ini bisa digunakan untuk penelitian dengan bidang yang sama dengan penelitian ini di masa depan.

Penelitian ini ditulis berdasarkan dari beberapa penelitian terdahulu seperti penelitian yang dilakukan oleh Batubara [2] dimana ia membuktikan bahwa Bcrypt bisa tidak bisa diserang menggunakan *brute force* ditambah dengan penelitian yang dilakukan oleh Yafie [3] dimana ia menemukan bahwa Bcrypt memiliki nilai *hash* per detik terkecil dari 20 fungsi *hash* yang ia uji yaitu 265 *hash* per detik. Sedangkan penelitian yang dilaksanakan oleh Handoyo dan Subakti [4] menemukan bahwa mereka bisa membuat *website* dengan dilengkapi dengan algoritma AES yang dapat digunakan untuk mengamankan data dokumen. Lalu ada juga penelitian pada penelitian yang dilakukan oleh Meko [5] mendapatkan hasil bahwa AES memiliki kecepatan enkripsi dan dekripsi terbesar sebesar 1.508 kb/s daripada DES, IDEA dan Blowfish. Penelitian yang ditulis oleh Kumar dan Chaudhary [6] menggunakan Bcrypt dan AES sebagai tombak pengamanan data nya dengan cara data nya akan disimpan didalam *list* berbentuk windows registry dimana diperlukan *password* yang sesuai dan *password* sudah di *hashing* menggunakan Bcrypt dan dan dienkripsi menggunakan AES. Berdasarkan penjelasan diatas, maka penulis membuat penelitian yang berjudul Implementasi Algoritma AES dan Bcrypt Untuk Pengamanan File Dokumen.

2 Landasan Teori

2.1 Kriptografi

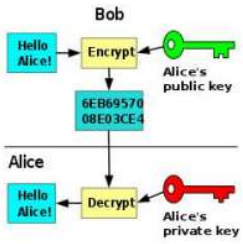
Menurut Wirdasari [7], Kriptografi atau kriptologi merupakan kata yang berasal dari Bahasa Yunani yang terdiri dari *kryptós* yang berarti tersembunyi dan *graphein* yang berarti tulisan atau logi yang berarti ilmu.Penggunaan kriptografi bisa dilacak dari 3000 tahun SM, mereka menggunakan *hieroglyphcs* dimana benda ini berfungsi untuk menyembunyikan pesan dari pihak yang tidak bertanggung jawab



Gambar. 1. Skema Algoritma Simetris

2.2.1 Algoritma Simetris

Algoritma ini bekerja dengan cara kunci yang diberikan sama pada proses enkripsi maupun dekripsinya.



Gambar. 2. Skema Algoritma Simetris

2.2.2 Algoritma Asimetris

Algoritma ini menggunakan dua kunci yang berbeda [8], baik dalam proses enkripsi maupun dekripsinya. Kunci-kunci tersebut yaitu:

1. Kunci umum (*public key*) merupakan kunci yang sifatnya umum dan dapat diketahui semua orang.
2. Kunci pribadi (*private key*) merupakan kunci yang ber besifat rahasia dan hanya penerima yang mengetahuinya.

2.2 Hash

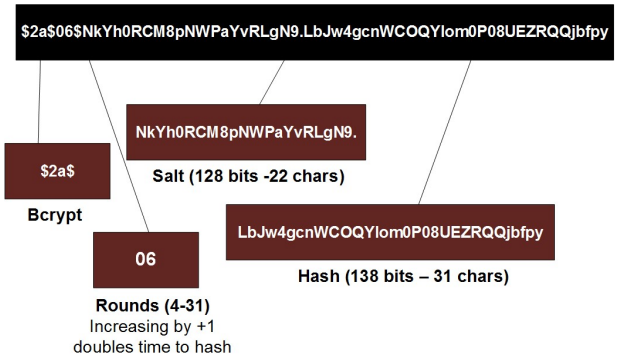
Menurut Munir [9], fungsi *hash* merupakan fungsi yang panjang *string* nya sembarang dengan luaran yang telah ditransformasi menghasilkan panjang yang tetap.

2.3 AES

AES memiliki panjang kunci masing-masing yaitu 128 bit, 192 bit dan 256 bit [10]. Dalam melakukan enkripsi, data dikelompokkan pada ukuran tertentu atau disebut juga dengan blok. Langkah operasi enkripsi pada AES yaitu:

1. *AddRoundKey*: melakukan XOR antara pesan yang ingin dienkripsi (*plainteks*) dengan *cipher key*. *Initial round* merupakan nama lain dari tahap ini.
2. Lakukan pemutaran sebanyak Nr– 1 kali. Setiap putaran memiliki proses yang harus dilakukan yaitu:
   a.*SubBytes*: Gunakan tabel substitusi (S-box) untuk melakukan substitusi *byte*.
   b.*ShiftRows*: Baris-baris *array state* dilakukan pergeseran dengan prosedur *wrapping*.
   c.*MixColumns*: Setiap kolom *array state* dilakukan pengacakan data.
   d.*AddRoundKey*: Lakukan operasi XOR antara *state* lalu *round key*.
3. *Final round*: putaran terakhir menggunakan proses berupa:
   a.*SubBytes*
   b.*ShiftRows*
   c.*AddRoundKey*

2.4 Bcrypt



Gambar. 3. Struktur Bcrypt

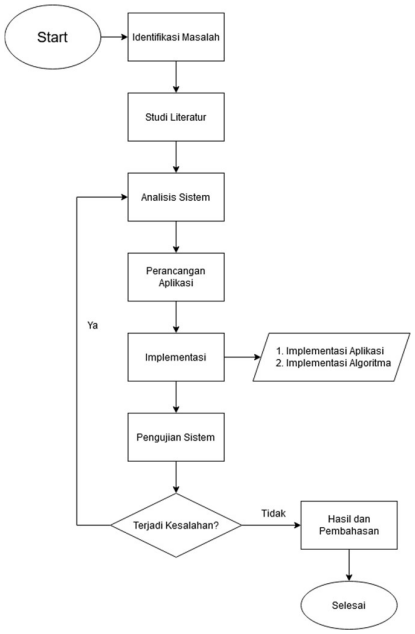
Bcrypt merupakan *hashing* yang banyak iterasinya bertambah terus menerus untuk memperlambat dan bertahan lama dari serangan *brute force* dengan meningkatkan daya komputasinya, dengan menggabungkan *salt* agar terlindungi dari serangan *rainbow table*.

Menurut Dukha[11], enkripsi Bcrypt memiliki tahapan, yaitu:

1. *Array* P dilakukan inisiasi sebanyak 18 iterasi (Setiap dari P memiliki nilai sebesar 32 bit. *Array* P mempunyai18 kunci dan 32 bit subkunci).
2. *Plaintext* yang akan dienkripsi dengan jumlah bit yang diperlukan yaitu 64-bit, bila jumlah bit pada *plaintext* kurang dari 64-bit maka *cost* harus ditambahkan.
3. Hasil pengambilan tadi dibagi menjadi XL yang merupakan 32-bit pertama dan XR sebagai 32-bit kedua.
4. Setelah langkah 1-3 selesai, gunakan operasi XL = XL xor Pi dan XR = F(XL) xor XR .
5. Hasil penggunaan operasi diatas lalu harus ditukar, yaitu dengan cara XL menjadi XR dan XR menjadi XL
6. Setelah terjadi 16 kali operasi, Pada iterasi 16terjadi operasi penukaran yaitu antara XL dan XR.
7. Lakukan operasi XR = XR xor P17 dan XL = XL xor P18 ketika sudah pada proses ke 17.
8. XL dan XR disatukan agar jumlah bit kembali menjadi 64 bit.

3 Metode Penelitian

Tahapan penelitian dituangkan pada gambar dibawah ini:



Gambar.4. Tahapan Penelitian

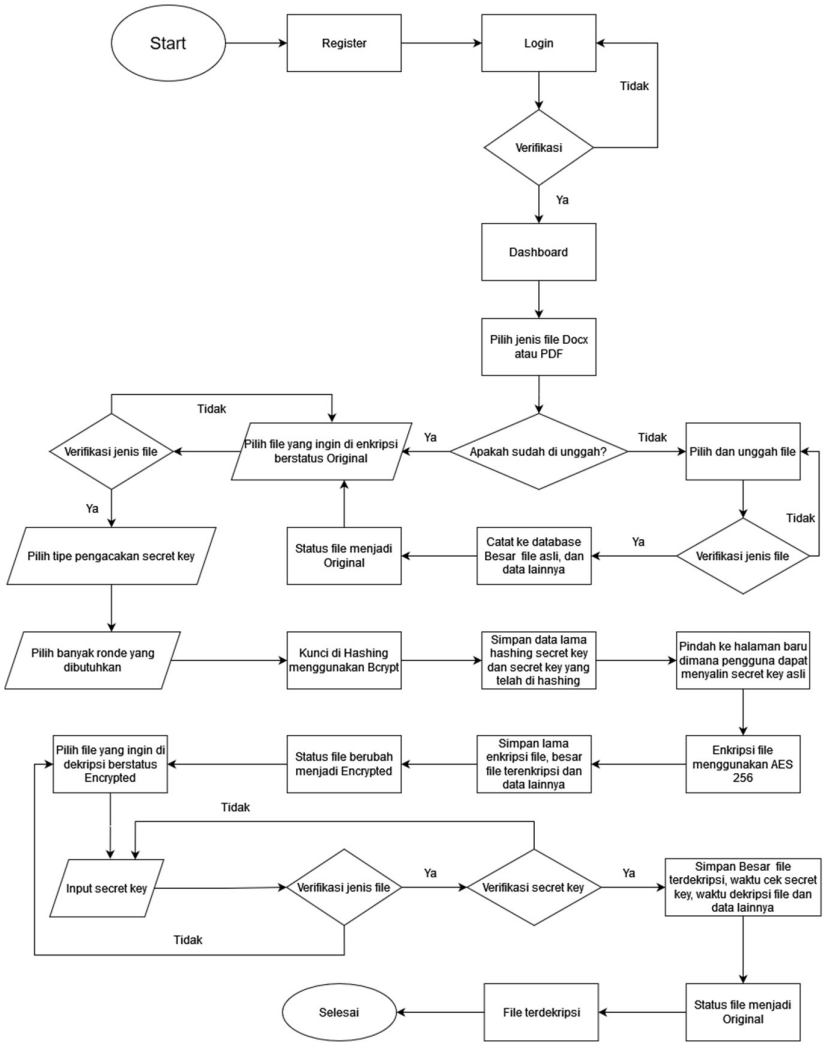
3.2.1 Identifikasi Masalah
Mengidentifikasi masalah yang berhubungan dengan penelitian ini.

3.2.2 Studi Literatur
Pencarian informasi berupa teori, konsep, dan pengaplikasian dari sumber literatur seperti jurnal, skripsi, tesis, dsb.

3.2.3 Analisis Sistem
Analisis sistem dilakukan dengan menjabarkan kebutuhan sistem dan alur penggunaannya serta menggambarkannya dalam bentuk pemodelan UML seperti *flowchart*, *activity diagram* dan *use case*.

3.2.4 Perancangan Aplikasi
Merancang aplikasi berbasis *web* sesuai dengan rancangan yang telah dibuat.

3.2.5 Implementasi
Perancangan sistem dilakukan menggunakan *framework* Laravel serta menggunakan algoritma kriptografi Bcrypt dan AES untuk pengamanan *file*. Aplikasi akan dikembangkan dan dijalankan pada laptop dengan *file* yang perlu diamankan yaitu .docx dan .pdf. Sistem pengamanan ini berjalan dengan cara pengguna diharuskan untuk teregistrasi untuk mengakses data nya, lalu *file* akan di enkripsi dan dekripsi menggunakan AES, sedangkan *secret key* yang akan digunakan untuk membuka *file* terenkripsi akan di *hash* menggunakan Bcrypt. Dibawah ini merupakan flowchart sistem lebih lengkapnya:



Gambar.5. Alur penggunaan aplikasi

3.2.6 Pengujian Sistem

Adapun serangkaian pengujian yang akan dilakukan pada penelitian kali ini yaitu:

1. Mencari kesalahan pada sistem atau melakukan penyempurnaan lebih lanjut sehingga program dapat berjalan dengan lebih baik.
2. Aplikasi diuji apakah *file* dengan format docx dan pdf bisa di enkripsi atau dekripsi menggunakan AES.
3. Setiap jenis *file* akan diuji dengan jumlah tertentu, dengan ketentuan *file* berjenis pdf sebanyak 77 dan *file* berjenis docx sebanyak 77.
4. Perbandingan antara ukuran *file* asli, sebelum dan sesudah dienkripsi.
5. Menghitung lama waktu yang dibutuhkan untuk proses enkripsi serta dekripsi menggunakan AES pada *file* pdf dan docx.
6. Melakukan perbandingan lama waktu yang dibutuhkan untuk *hashing secret key* atau kunci rahasiaterhadap *cost* tertentu pada Bcrypt, yaitu dari 10 hingga 20. Kombinasi karakter pada *key* yaitu:
    a. Abjad
    b. Nomor
    c. Tanda baca
    d. Abjad dan nomor

e. Abjad dan tanda baca
f. Tanda baca dan nomor
g. Abjad, tanda baca dan nomor

7. Untuk mengetahui pengaruh dari perangkat komputer yang digunakan pada proses enkripsi dan dekripsi menggunakan algoritma yang diujikan, penulis menggunakan dua laptop dengan spesifikasi yang berbeda.
8. Uji intergitas, keaslian *secret key* dan *file* dengan cara membandingkan isi *hash* SHA-256 berbentuk heksadesimal menggunakan Get-FileHash di Windows 10 pada 5 *file* pdf serta 5 docx dan hasil heksadesimal sebelum dan sesudah di *hashing* ronde 10 kombinasi karakter abjad menggunakan Bcrypt.

3.2.7 Hasil dan Pembahasan

Bila tahap pengujian telah berakhir, hasilnya akan dijelaskan lebih lanjut pada bab empat, dimana akan ditarik kesimpulan apakah aplikasi yang dibuat telah berjalan dengan baik atau belum.

3.3 Perangkat yang Digunakan

3.3.1 Perangkat Keras

Perangkat keras yang digunakan untuk penelitian ini pada komputer 1 yaitu:

- Merek : ASUS A456U
- Processor : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz (4 CPUs), ~2.7GHz keluaran tahun 2016.
- VGA : Intel HD Graphics 620 + NVIDIA GEFORCE 930MX.
- Display : 1366 x 768.
- Storage : 1 TB.
- RAM : 12 GB DDR4
- Lama Pemakaian : 5 Tahun

Perangkat keras yang digunakan untuk penelitian ini pada komputer 2 yaitu:

- Merek : ASUS Vivobook 14
- Processor :Intel(R) Pentium(R) CPU 5405U @ 2.30GHz (4 CPUs), ~2.3GHz keluaran tahun 2019.
- VGA : -
- Display : 1920 x 1080.
- Storage : 1 TB.
- RAM : 4 GB DDR4
- Lama Pemakaian : 1 Tahun

3.3.2 Perangkat Lunak

Perangkat lunak yang perlukan pada pengerjaan penelitian ini yaitu:

- Sistem Operasi : Windows 10
- Aplikasi Editor Code : Visual Studio Code
- Bahasa Pemrograman : PHP>=7.3
- Framework : Laravel 8.0
- PostgreSQL : 12.3
- Apache : >=2.2
- PgAdmin : 4.2

4 Analisis dan Pembahasan

4.1 Analisis Sistem

Aplikasi berbasis *website* ini menggunakan algoritma AES *(Advanced Encryption Standart)*sebagai teknik untuk mengenkripsi dan dekripsi *file*, sedangkan Bcrypt digunakan untuk melakukan tahap *hashing* kunci yang akan dipakai oleh *user*. *File* yang diujikan memiliki format berupa .docx dan .pdf. Dalam pengerjaannya, penulis memanfaatkan beberapa fitur pada Laravel. Untuk *hashing secret key*, fitur yang digunakan yaitu fitur bawaan Laravel **Hash**. Sedangkan untuk enkripsi dan dekripsi *file*nya, perlu menggunakan *package* **FileVault**. *Database* yang digunakan yaitu berbasis posgresql, dimana digunakan pgadmin untuk mengatur tabel yang diperlukan untuk penelitian ini. Laravel digunakan sebagai *framework* pada penelitian ini karena banyak terdapat *library* yang bisa digunakan serta di bentuk untuk membangun aplikasi yang berjalan dengan baik dan sesuai tujuan pembuatannya.

4.2 Pengujuan Besar Data

Dilakukan analisa terhadap besar *file* sebelum dan sesudah terjadi pengamanan data. Dibawah ini merupakan sebagian dari data yang didapatkan:

Tabel 1. Pengujian Besar *File* Tipe Pdf

nama	Besar file asli(kb)	Besar file terenkripsi(kb)	Besar file terdekripsi(kb)
Internet_of_Things_IoT_Charity_Automation.pdf	702.93	705.7	702.93
ITN_FINAL_ASSESSMENT.pdf	109.86	110.31	109.86
14070-34936-2-PB.pdf	856.87	860.25	856.87
Mastering Unity Scripting - Alan Thorn.pdf	8634.97	8668.84	8634.97
SOAL UAS Praktikum Ethical Hacking - Copy.pdf	14.11	14.19	14.11
SOAL UTS Teori Ethical Hacking.pdf	16.88	16.97	16.88
Mekanisme Keamanan Cloud.pdf	978.77	982.63	978.77
Implementation_of_RC5_Symmetric_Key_Encr.pdf	43.2	43.38	43.2
Ruby Cookbook, Second Edition - Lucas Carlson.pdf	9268.73	9305.09	9268.73
Laravel Up & Running A Framework for Building Modern PHP Apps (PDFDrive).pdf	8005.62	8037.03	8005.62

Tabel 2. Pengujian Besar *File* Tipe Docx

nama	Besar file asli(kb)	Besar file terenkripsi(kb)	Besar file terdekripsi(kb)
[MS-DOCX]-210817.docx	171.8	172.5	171.8
Dampak Teknologi Informasi terhadap Transaksi Elektronik di Universitas Pertamina.docx	154.75	155.38	154.75
Lab 9.docx	4530.47	4548.25	4530.47
OAJIS_21_1378.docx	267.31	268.39	267.31
1710511011_GebrinaDivvaMeuthiaZulma_TBA_tugas2 .docx	633.28	635.77	633.28
Laporan Pertanggung Jawaban.docx	27.52	27.64	27.52
UAS_EH_GebrinaDivvaMeuthiaZulma_1710511011.do cx	21.14	21.25	21.14
PJKL_ProxyServer_011_Gebrina Divva Meuthia Zulma.docx	3584.76	3598.83	3584.76
Contoh CVb4.docx	45.47	45.67	45.47
kebutuhan keseluruan pementasan.docx	15.98	16.06	15.98

Dengan mencari rata-rata dari setiap data yang didapatkan, didapatkan angka 0.393% untuk perbedaan antara besar *file* terenkripsi dengan *file* terdekripsi serta besar *file* terenkripsi dengan *file* asli pada tipe *file* pdf. Sedangkan pada docx dengan perhitungan yang sama mendapatkan nilai 0,396%. Melalui contoh diatas terlihat bahwa *file* terenkripsi selalu lebih besar dari *file* asli dan lebih kecil dari *file* terdekripsi.

4.3 Pengujian Waktu

Komputer 1 dan komputer 2 dihitung waktu nya ketika melakukan proses enkripsi, dekripsi, *hash secret key* dan cek *secret key.* Dibawah ini merupakan nilai rata-rata yang didapatkan setelah menghitung keseluruhan hasil dari komputer 1 dan komputer 2:

Tabel 3. Rata-rata Lama Waktu Penggunaan AES Pada Pengamanan File

Nama	Waktu enkripsi file	Waktu dekripsi file
Docx Komputer 1	0.0547941	0.01254231
Docx Komputer 2	0.05501772	0.04572595
Pdf Komputer 1	0.19478191	0.03843315
Pdf Komputer 2	0.13751073	0.05697854

Setelah dihitung rata-ratanya, komputer 1 memiliki kemampuan yang unggul dalam pemrosesan enkripsi dan dekripsi, dengan komputer 1 unggul dari komputer 2 dengan perbedaan waktu untuk enkripsi *file* docx yaitu 0,406%, dekripsi *file* docx sebesar 72.571% dan dekripsi *file* pdf dengan nilai 48,254%. Hanya pada enkripsi *file* pdf komputer 2 unggul dengan perbedaan waktu sebesar 41.649%. Dibawah ini merupakan tabel waktu *hashing* kunci menggunakan Bcrypt kepada kedua komputer:

Tabel 4. Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 1 Pada File Pdf

Pdf Komputer 1		
nama	Waktu Hashing Secret Key	Waktu Cek Secret Key
Ronde 10	0.12259	0.11913
Ronde 11	0.23750	0.22121
Ronde 12	0.47600	0.45804
Ronde 13	0.95303	0.86532
Ronde 14	1.88107	1.92032
Ronde 15	3.79370	3.81281
Ronde 16	7.37678	7.48699
Ronde 17	14.99181	15.04978
Ronde 18	30.09383	30.32377
Ronde 19	59.65375	59.77861
Ronde 20	127.93276	129.05019
Rata-rata	22.50116	22.64420

Tabel 5. Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 1 Pada File Docx

Docx Komputer 1		
nama	Waktu Hashing Secret Key	Waktu Cek Secret Key
Ronde 10	0.14828	0.14632

Ronde 11	0.26489	0.28091
Ronde 12	0.50087	0.48013
Ronde 13	0.96631	0.97183
Ronde 14	1.97155	1.97625
Ronde 15	3.89662	3.88822
Ronde 16	8.49313	8.66046
Ronde 17	15.32365	15.48346
Ronde 18	30.97926	31.03319
Ronde 19	61.52539	68.05198
Ronde 20	145.67493	149.79731
Rata-rata	24.52226	25.52455

Tabel 6. Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 2 Pada File Pdf

Pdf Komputer 2		
nama	Waktu Hashing Secret Key	Waktu Cek Secret Key
Ronde 10	0.13703	0.13731
Ronde 11	0.24464	0.21544
Ronde 12	0.43338	0.48416
Ronde 13	0.87013	0.88611
Ronde 14	1.67584	1.75764
Ronde 15	3.57616	3.66342
Ronde 16	7.04474	7.19724
Ronde 17	13.97595	14.47471
Ronde 18	28.18290	29.30792
Ronde 19	49.80715	51.10553
Ronde 20	98.84095	101.32962
Rata-rata	18.61717	19.14174

Tabel 7. Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 2 Pada File Docx

Docx Komputer 2		
nama	Waktu Hashing Secret Key	Waktu Cek Secret Key
Ronde 10	0.20136	0.13146
Ronde 11	0.26079	0.23561
Ronde 12	0.42671	0.44181
Ronde 13	0.90363	0.90594
Ronde 14	1.71046	1.74349
Ronde 15	3.49003	3.52118
Ronde 16	7.06841	7.35036
Ronde 17	14.02249	14.59528
Ronde 18	27.99203	27.85425

Ronde 19	55.98858	55.85608
Ronde 20	111.43726	113.07409
Rata-rata	20.31834	20.51905

Selalu terjadi peningkatan ketika ronde meningkat, dengan kenaikan yang bervariasi ketika berganti ronde, mulai dari dari 22,788% hingga 57,765%. Selain itu, terlihat bahwa komputer 2 unggul dalam lama pemrosesan nya daripada komputer 1, dimana terjadi peningkatan dari komputer 2 ke komputer 1 sebanyak 17,261% dan 15,467% pada lama waktu *hashing secret key* dan lama waktu cek *secret key* pada *file* pdf. Sedangkan pada *file* docx, lama waktu *hashing secret key* dan lama waktu cek *secret key* tercatat memiliki perbedaan antara komputer 1 dan 2 sebanyak 17,143% dan 19,611% dimana komputer 1 mempunyai waktu terlama. Selain itu, bisa dilihat bahwa terdapat peningkatan lama waktu pemrosesan ketika melakukan proses cek *secret key,* dimana untuk pdf dan docx pada komputer 1 yaitu 0,636% dan 4,087%, sedangkan pada komputer 2 untuk pdf dan docx mendapatkan perbedaan dengan nilai 2,818% dan 0,988%.

Selanjutnya yaitu membandingkan beda waktu pada pengkombinasian kunci antara abjad, nomor, tanda baca, abjad dan nomor, abjad dan tanda baca, nomor dan tanda baca serta abjad, nomor dan tanda baca. Dibawah ini merupakan tabel hasil penelitiannya:

Tabel 8. Presentase Peningkatan Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 1 Pada File Pdf

Docx Komputer 1		
Nama	Rata-rata	
	Enkripsi	Dekripsi
Abjad	26.58335	28.37925
Nomor	27.26570	27.73227
Tanda baca	23.90570	24.28251
Abjad dan nomor	18.71846	18.84557
Abjad dan tanda baca	24.36893	28.37370
Tanda baca dan nomor	26.43474	26.82646
Abjad, tanda baca dan nomor	24.37895	24.23210

Tabel 9. Presentase Peningkatan Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 1 Pada File Docx

PDF Komputer 1		
Nama	Rata-rata	
	Enkripsi	Dekripsi
Abjad	25.69228	25.79822
Nomor	23.49133	23.88479
Tanda baca	15.48774	15.52757
Abjad dan nomor	25.15069	25.23002
Abjad dan tanda baca	20.21623	20.42616
Tanda baca dan nomor	26.46009	26.57184
Abjad, tanda baca dan nomor	21.00979	21.07077

Tabel 10. Presentase Peningkatan Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 2 Pada File Pdf

Docx Komputer 2

Nama	Rata-rata	
	Enkripsi	Dekripsi
Abjad	18.92296	19.17386
Nomor	24.09301	24.68628
Tanda baca	19.24128	19.26881
Abjad dan nomor	20.47629	20.72576
Abjad dan tanda baca	19.95329	20.54213
Tanda baca dan nomor	21.17581	20.34985
Abjad, tanda baca dan nomor	18.36575	18.88667

Tabel 11. Presentase Peningkatan Waktu Penggunaan Bcrypt Untuk Pengamanan *Secret Key* Komputer 2 Pada File Docx

Docx Komputer 2		
Nama	Rata-rata	
	Enkripsi	Dekripsi
Abjad	18.92296	19.17386
Nomor	24.09301	24.68628
Tanda baca	19.24128	19.26881
Abjad dan nomor	20.47629	20.72576
Abjad dan tanda baca	19.95329	20.54213
Tanda baca dan nomor	21.17581	20.34985
Abjad, tanda baca dan nomor	18.36575	18.88667

Tidak terdapat pola yang bisa dikaitkan dari tabel diatas terhadap kombinasi dari *secret key* dengan lama pemrosesan kuncinya.

4.4 Pengujian Integritas File dan Secret Key

Dibawah ini merupakan hasil pengujian integritas:

Tabel 12. Perbandingan Hasil *Hash* di Dalam *File* Berbentuk Heksadesimal

Nama	Nilai hash	
1-Big-Grammar-Book-MS-Word-Version.docx	asli	3B5E9C7D99CF1E145ABC9E088D895EECC16B5C1F1485E5CFF88D0AA9B207F67D
	terenkripsi	69C119FFD0929473F92E9E9D5CC323EB9BBB4B19DE5A3D93D9694BA040E9C549
2-Big-Activity-Book-MS-Word-Version.docx	asli	00936401394A786E17592EEB385D4415BC31CFA320D2C3C282BB34DB62C1532A
	terenkripsi	84108509074BD3006D53ED535005D6C7C95E0A61E1C0CF58DB325ECB3265EFA8
4-Check-It-Again-Book-One-MS-Word-Version.docx	asli	8074241AD623A9077D18D0FAF57B0F83F88DE9152D2C8C5E084FB7D6ED34FC43
	terenkripsi	90CCFCDB1C12AEE1780EDFC40439CDBEE7E7ECC72EF5CC24DBECC263F3F44D04
5-Talk-A-Lot-Elementary-Book-1-MS-Word-Version.docx	asli	B873C4F902CE045489B7B13C8882B572AB4F8CDA237D35401833299135D290C7
	terenkripsi	8B4A7B00AFC22D0667BD40475EC0E7F875A40C14D862F777A1BFE44BDF4D63AD

6-Talk-A-Lot-Elementary-Book-2-MS-Word-Version.docx	asli	8ABD61F65A441CFC5F0FCF7360DFBB237ED24FE2A4EB5CD9FB8B0A3E05BB8071
	terenkripsi	DC299EFA6109DC5F2583C152DD6B08C4DFD3DE55EB5761BBA30CE9226533C544
BukuBI.pdf	asli	DCD80D8385A2C251722D2CD3DFD753BB605D7BDFD6E37E21411DFB4A7B3185BD
	terenkripsi	7A7B14DF7AA11565F666A5F7C76CE02C25DAE9EDBDD6795A844FD2DB2DE2964A
lirik-dan-chord-lagu-indonesia2.pdf	asli	9DD182D466E18637BF7D8861F466F8E2D27E73C791920E0EB664D91666B65175
	terenkripsi	115618D8B6F00092361ECEB47D07FB5BFC370AFD9A802E44866C31BA488CDFB0
MDALHAR_C0507033.pdf	asli	9C58854CA33AFBCEC418024BF0E2EED287C3B0BF7277CA64D5C92BB5B8645DFE
	terenkripsi	24B91E78D011566742453D6212682AC702C46FA61CF6CFC3E5BDE8EF65A74A81
SejarahIslamyangHilang.pdf	asli	39ADE7AD1EEEC9CB51DEF38A7AB8BCCF2FE88E9F3F7D5BF7E9979AE783E3474E
	terenkripsi	C5794052C608847180288EE207ACB4BED2EA09BE57D4A7282EAA5D7E8BAE3E74
wni_di_timor_leste_buku_saku_win_di timor_leste.pdf	asli	541C7FD0151CC7B11BB417CE6B25B5CFB9B280CBC17536AB30A4D6827213E36C
	terenkripsi	A309B7E7695DCD2044C13C76AA54BBA5CB3F37F8EB7AF22FF55FAF52232EBC30

Tabel 13. Perbandingan Output Heksadesimal Sebelum dan Sesudah di *Hash*

Teks asli	Hashing	Nilai Hexadecimal	
nPPPTjnORwGrkPKPPiEkptwFrRmqbUYZ	$2y$10$XdlmPzcngcVauOuJimHW0uNiErcLCvJ3WwGTdwd.vM8FtEPI1RS0y	asli	6E505050546A6E4F527747726B504B505069456B707477467252526D716255595A00
		hashed	3E41AD78D1312F96322157F27C0C4EF87BC644ADD3536D56
SMacaMdRkiTCIngsvFQsjEmIEpEgRNTH	$2y$10$DGF1g37YZ..MEeN5TGE6V.3LTvEXGSEQgRcQL1KpITP5pPG0fQCLm	asli	534D6163614D64526B695443496E67737646517 36A456D4945704567524E544800
		hashed	E4D57119921419289379237732B29547BAD123685210DAA7
UdzZWAsCqJmeNslyymJKNNJzlTqTYBgs	$2y$10$zKJjJ0mNP0KaKNwj80dMmejQwzhJsDLJXlGcPW9yKk/babH2mSoli	asli	55647A5A57417343714A6D654E736C79796D4A4B4E4E4A7A6C5471545942677300
		hashed	952CB58CBB8534B66721E458FF432605D71D278A14AA79EC
BlAQjcUgLCHJQtXoQZxMCPYekiydJekq	$2y$10$40dYuavgd0ClKbxJaT5yIeFYMxNiktVi/VcZCeFRdMqwVay/25Mzi	asli	426C41516A6355674C43484A5174586F515A784D435059656B6979644A656B7100
		hashed	1DA3B33E49AF5E405779B1201D37CEB325DCD01E3B3B59CD
iQccYpKmwJLKZeAlWbXTjGYsNgQhoztz	$2y$10$toGsBbC7iagichUULCn5C.WNl.DmdBjBtB2G5jLXrTabFCdAEQJKK	asli	6951636359704B6D774A4C4B5A65416C576258546A4759734E6751686F7A747A00
		hashed	60F9C01687C3943BC3E08EE5359B5571D1C47C21922CC3B8
SCnHsdPSKwgnwjUBZJQChbHPzbbBisJd	$2y$10$Dh4Zgbdt4hlWQ0PVgmGxbuk2FhGjD6aaEeyC/j1eQaKSRNVgRAOJK	asli	53436E48736450534B77676E776A55425A4A5143686248507A62624269734A6400
		hashed	9B81E322517C71C1A0D04065DE049C3144CF5E24C240B3BB
KnemBdrcKIeOGWRmZdLMkVoVvJWoRpSy	$2y$10$xrY.OnfhMEPAm2KIKTsWGu/EDja9eCC7qrvHeINo4qkEwpIZXhkBa	asli	4B6E656D426472634B49654F4757526D5A644C4D6B566F56764A576F5270537900
		hashed	04616573F80413DB2DC4980A3EAEAC986CAB29B663983799
OgKmveqmYOcBpcxUfsrj	$2y$10$PEuE5vi6Zy5b8Ev.On8bVeYOQhAw	asli	4F674B6D7665716D594F6342706378556673726A48664A6759714476684D636900
		hashed	6904A30B234D6B376054D0EAD447920A1BC90D11CBE0D37D

HfJgYqDvhMci	LLYxbeTLBozCcQAftHBPFJ2LK		
avTJHOmMvZzcLayWzcTvjRaScmHqqTqt	$2y$10$96CTXnQ.Ky8Lj6oS19rNEOwQcvOOIXEwZyvsmbRi6hpc76ZkE8gV.	asli	6176544A484F6D4D765A7A634C6179577A6354766A526153636D48717154717400
		hashed	C927B14102991B26F4C6EA1D4E4F23ADEF7C6E61BE897076
SMmGjlbGvJzHBfFYDAUTRfhLrdmQVlMj	$2y$10$J4FMFlq28KDTL4UPtzespefx.LEj1oViF//LB.s9f.HiJuYwyl.8K	asli	534D6D476A6C6247764A7A484266465944415554526668 4C72646D51566C4D6A00
		hashed	87300D1A5DEA5E41C104D0C0BBF8402642F06B2D2703E3B8

Berdasarkan tabel diatas, dapat ditarik kesimpulan bahwa ketika *file* dilakukan proses enkripsi, nilai *hash* tersebut berbeda dengan sebelum dienkripsi. Lalu *string* yang di *hash* menggunakan Bcrypt memilki panjang heksadesimal yang lebih pendek daripada teks aslinya, dimana panjang heksadesimal yang dihasilkan yaitu sebanyak 24 karakter.

4. Penutup

4.1 Kesimpulan

1. Setelah dilakukan pendalaman ilmu dengan menggunakan *framework* yang tersedia, Laravel merupakan *framework* yang tepat untuk penelitian ini, dimana *package* bernama FileVault dapat digunakan untuk mengenkripsi dan dekripsi *file* tipe .docx dan .pdf menggunakan AES.
2. Bcrypt menggunakan fitur Hash pada Laravel dimanfaatkan untuk mengamankan *secret key* pada *database* yang akan digunakan untuk mendekripsi file tipe docx dan pdf.
3. Saat *file* dienkripsi terdapat penambahan pada ukuran *file*.
4. Komputer 1 unggul waktu pemrosesan algoritma nya AES dari komputer 2, dengan perbedaan enkripsi docx sebesar 0.406%, lalu dekripsi docx dengan nilai 72,571% dan dekripsi pdf dengan 48,254%. Komputer 2 hanya unggul pada waktu enkripsi pdf dengan perbedaan waktu sebesar 41,649%.
5. Semakin besar ronde pada Bcrypt yang diinputkan, maka akan semakin lama waktu yang dibutuhkan untuk melakukan *hashing* pada suatu teks. Kenaikan waktunya untuk setiap pergantian ronde bervariasi dari 22,788% hingga 57,765%.
6. Waktu yang dibutuhkan untuk *hashing* kunci lebih besar daripada saat proses cek kunci atau *secret key,* dengan rincian komputer 1 untuk pdf dan docx yaitu 0,636% dan 4,087%, sedangkan pada komputer 2 pada pdf dan docx memiliki nilai 2,818% dan 0,988% .
7. Perbedaan waktu untuk *hashing secret key* pdf antara komputer 1 dan 2 dimana komputer 2 lebih cepat yaitu sebesar 17,261%, untuk cek *secret key* pdf sebesar 15,467%, sedangkan untuk *hashing secret key* docx sebesar 17,143% dan cek *secret key* mendapatkan nilai sebesar 19,611%.
8. Kombinasi teks *secret key* tidak berpengaruh dalam lama enkripsi dan dekripsi dari *file* pdf dan docx.
9. File yang di enkripsi menggunakan AES mempunyai nilai *hash* yang berbeda.
10. Hasil *hashing* dari Bcrypt menunjukkan teknik *hash* yang dilakukan dapat menghasilkan teks yang lebih sedikit dari teks originalnya, dengan panjang heksadesimal yang konsisten yaitu 24 karakter.

4.2 Saran

1. Menggunakan algoritma lainnya seperti Triple Des, RSA, Serpent dan Twofish.
2. Pengujian yang dilakukan pada algoritma Bcrypt masih terbatas menggunakan ronde 10 hingga 20, disarankan untuk menggunakan ronde 21 hingga 31 agar dapat menguji performa dan kemanaan Bcrypt lebih lanjut.
3. Memilih algoritma *hashing* lainnya untuk penelitian terkait seperti SHA-3, Argon2, SHA2-384 dan Scrypt

5. Referensi

[1] Hafis, F. (2020, Mei 11). *Password Tokopedia yang Bocor Dienkripsi Algoritma MD5, Amankah?* Retrieved April 1, 2022, from Cyberthread.id: https://cyberthreat.id/read/6609/Password-Tokopedia-yang-Bocor-Dienkripsi-Algoritma-MD5-Amankah

[2] Batubara, T. P. (2020). *Analisis Kinerja Algoritma Bcrypt Untuk Meningkatkan Keamanan Password Dari Brute Force.* Medan: Universitas Sumatera Utara.

[3] Yafie, H. N. (2020). Analisis Penggunaan Fungsi Hash BCrypt untuk Keamanan Kata Sandi. Institut Teknologi Bandung. Dipetik Agustus 1, 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Makalah-UAS/Makalah-UAS-Kripto-2020%20(16).pdf

[4] Handoyo, J., & Subakti, Y. M. (2020). Keamanan Dokumen Menggunakan Algoritma Advanced Encryption Standard (AES). *Jurnal SITECH : Sistem Informasi Dan Teknologi, 3*(2), 143–152. doi:https://doi.org/10.24176/sitech.v3i2.5865

[5] Meko, D. A. (2018). Perbandingan Algoritma DES, AES, IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data. *Jurnal Teknologi Terpadu, 4*(1), 8-15.

[6] Kumar, N., & Chaudhary, P. (2016). Password Security Using Bcrypt with AES Encryption Algorithm. *Smart Computing and Informatics, 1*, 385-392.

[7] Wirdasari, D. (2008). Prinsip Kerja Kriptografi dalam Mengamankan Informasi. *Jurnal SAINTIKOM, 5*(2), 174–184. Retrieved from https://prpm.trigunadharma.ac.id/public/fileJurnal/42481-OK-Jurnal6-DW-Comsec2-174-184.pdf

[8] Sinaga, M. C. (2017). *Kriptografi Python.* Dipetik Agustus 1, 2021, dari https://www.academia.edu/34788898/Kriptografi_dan_Python_pdf

[9] Munir, R. (2018). Fungsi Hash. Institut Teknologi Bandung. Dipetik Agustus 1, 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2017-2018/Fungsi-Hash-(2018).pdf

[10] Munir, R. (2004). Advanced Encryption Standard (AES). Institut Teknologi Bandung. Dipetik Agustus 1, 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Advanced%20Encryption%20Standard%20(AES).pdf

[11] Dukha, N. (2019). Implementasi Algoritma Bcrypt Pada Sistem Informasi Koperasi Simpan Pinjam Berbasis Website (Studi Kasus : Koperasi Mandiri Yayasan Fathul Ulum Gabus Grobogan). Semarang: Universitas Negeri Semarang.