Implementasi Algoritme Kriptografi SIMON pada Arsitektur Amazon Web Services

Di buat oleh:

Kelompok 5

5200411146 – GABRIEL SUMAMPOW

5200411527 – MUHAMMAD DAFFA KR

5200411528 – NOVIYAN SYAMSUWARDI

5200411547 – YOGA MAULANA QHADAFI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

TAHUN AJARAN 2020/2021

DAFTAR ISI

BAB I
PENDAHULUAN

A. Latar Belakang

Penggunaan *cloud computing* membawa banyak manfaat pada industri Teknologi Informasi untuk efisiensi waktu dan biaya yang murah. Dengan memanfaatkan *cloud computing,* sebuah organisasi dapat saling berbagi sumber daya melalui jaringan internet tanpa harus memiliki infrastruktur yang kompleks. Pada tahun 2006, sebuah perusahaan terkemuka yaitu Amazon, meluncurkan sebuah layanan *cloud computing* yang bernama Amazon Web Services (AWS). Dengan memanfaatkan AWS, pengguna dapat meminta daya komputasi, penyimpanan dan layanan lainnya dalam hitungan menit serta fleksibilitas untuk memilih platform pengembangan yang diinginkan sesuai kebutuhan pengguna (Varia & Mathew, 2014).

Salah satu layanan dari AWS yang berguna untuk menyimpan data secara online yang dapat dilakukan dimanapun dan kapanpun yaitu Amazon Simple Strorage Service (Amazon S3). Pada Amazon S3 pengguna diberikan sebuah wadah atau tempat yang disebut dengan bucket untuk menyimpan objek yang diinginkan. Amazon S3 memberikan akses yang andal, aman, cepat dan murah kepada penggunanya. Layanan Amazon S3 menggunakan arsitektur *Representational State Transfer* (REST) melalui protokol HTTP. Pada penelitian yang dilakukan oleh Prerna & Parul Agarwal pada tahun 2017, penelitian tersebut mengembangkan dan menerapkan algoritme kunci simetrik untuk mengenkripsi data pada sisi *client* dan diunggah menuju layanan penyimpanan *cloud* berbasis web. Penelitian tersebut menguji kemampuan algoritme untuk mengamankan file berupa teks dan hasilnya bekerja sesuai dengan kebutuhan. File yang dipilih oleh pengguna dienkripsi dengan cara mengambil setiap karakter yang ada kemudian mendapatkan nilai ASCII-nya, kemudian nilai ASCII tersebut dikonversikan ke dalam bentuk biner dan nilai biner tersebut dibalikkan urutannya dengan mengambil 4 bit pertama diletakkan pada 4 bit terakhir, setelah itu hasil akhir nilai biner dikonversikan ke dalam bentuk ASCII sebagai hasil *ciphertext.* File tersebut kemudian diunduh dan disimpan di sistem lokal, setelah itu pengguna memasukkan kunci yang sama pada saat proses enkripsi untuk dapat melakukan proses dekripsi. Isi dari file yang telah didekripsi kemudian dapat dibaca seperti file aslinya (Prerna & Agarwal, 2017).

Pada layanan Amazon S3, *bucket* dapat diamankan dengan Amazon *Server Side Encryption.* Ketika pengguna memilih opsi tersebut, setiap objek yang ada di dalam bucket diamankan dengan memanfaatkan algoritme *Advanced Encryption Standard* (AES) 256-bit. Terdapat opsi pengamanan lain yaitu menggunakan Amazon *Client Side Encryption*, objek diamankan menggunakan kunci yang

disimpan oleh pengguna pada sisi client kemudian diunggah menuju ke *bucket* Amazon S3.

Saat ini banyak dikembangkan algoritme kriptografi sebagai pengganti algoritme AES, salah satu organisasi dari Amerika yaitu *National Security Agency* (NSA) meluncurkan sebuah algoritme kriptografi yang bernama SIMON. Algoritme SIMON merupakan algoritme *block cipher* yang dapat diterapkan pada perangkat lunak maupun perangkat keras sesuai dengan kebutuhan penggunanya.

Berdasarkan hal tersebut, penulis ingin melakukan penelitian untuk menerapkan algoritme kriptografi SIMON pada arsitektur AWS untuk dapat mengamankan beberapa jenis file serta untuk menjamin aspek *confidentiality.* Algoritme SIMON memiliki bermacam-macam *block* dan *key sizes* yang dapat diterapkan oleh pengguna dengan menyesuaikan kebutuhan aplikasinya dengan kebutuhan keamanan yang diinginkan tanpa harus membebani performa sistem (Beaulieu, et al., 2013).

Algoritme SIMON berfungsi untuk melakukan enkripsi pada data yang kemudian dikirim menuju ke layanan Amazon S3 dengan menggunakan arsitektur AWS. Diharapkan bahwa algoritme SIMON dapat menjadi salah satu pilihan untuk metode pengamanan data pada sistem atau layanan *cloud computing.*

B. Rumusan Masalah

1. Apa itu Amazon Web Services (AWS)?
2. Apa saja Komponen Komponen AWS?
3. Apa kelebihan dan kekurangan Amazon Web Services (AWS)?

C. Tujuan Penelitian

1. Mengetahui apa itu AWS
2. Mengetahui Komponen Komponen yang terdapat pada AWS
3. Mengetahui kekurangan dan kelebihan AWS

BAB II
PEMBAHASAN

A. Pengertian Amazon Web Services (AWS)

Amazon Web Services (AWS) adalah sebuah layanan *cloud computing* yang banyak digunakan oleh perusahaan-perusahaan besar saat ini. Salah satu layanan dari AWS yang berguna untuk menyimpan data secara *online* yaitu Amazon Simple Strorage Service (Amazon S3). Pada layanan Amazon S3, terdapat opsi pengamanan data yaitu Amazon *Client Side Encryption*, objek diamankan menggunakan kunci yang disimpan oleh pengguna pada sisi *client* kemudian diunggah menuju ke *bucket* Amazon S3. Salah satu organisasi dari Amerika yaitu *National Security Agency* (NSA), meluncurkan sebuah algoritme kriptografi yang bernama SIMON. Algoritme SIMON merupakan algoritme *block cipher* yang dapat diterapkan pada perangkat lunak maupun perangkat keras sesuai dengan kebutuhan penggunanya. Algoritme SIMON diterapkan pada sebuah sistem *website* menggunakan Python yang digunakan untuk mengenkripsi *file* yang akan diunggah menuju *bucket* yang telah ditentukan. Pada pengujian kinerja waktu, hasil yang didapatkan adalah rata-rata yang paling cepat proses enkripsi dan dekripsi *file* yaitu sebesar 33,7 detik dan 28,5 detik. Pada pengujian variasi *file,* hasil yang didapatkan adalah algoritme SIMON dapat mengenkripsi *file* dengan ekstensi: .txt, .docx, .pdf, .png, .jpg, .mp3,.m4a, .mp4 dan .mkv.

B. Komponen Komponen AWS

1. **Amazon S3 (*Simple Storage Service*).** Digunakan untuk menyimpan data untuk penggunaan pribadi maupun umum. Dalam hal ini, ada 3 lokasi yang memungkinkan pemanfaatannya, yaitu di Amerika Serikat (termasuk California Utara), Eropa, serta Asia.
2. **Amazon Cloud Front.** Digunakan untuk mendukung Amazon S3 agar bisa bekerja dengan lebih baik dan lebih cepat.
3. **Amazon SQS (*Simple Queue Service*).** Digunakan untuk mendukung tercapainya pemrosesan AWS yang cepat dan tidak pernah mengalami kegagalan.
4. **Amazon SimpleDB.** Digunakan untuk menyimpan data yang bersifat semi-terstruktur. Basis data yang digunakan (SimpleDB) tidak bersifat relasional, melainkan menyimpan data dalam bentuk pasangan nama/nilai (*name/value*) yang mirip dengan struktur denormalisasi pada sistem basis data relasional, demi meningkatkan kinerja *query*.
5. **Amazon RDS (*Relational Database Service*).** Digunakan untuk mengelola data yang disimpan dalam sistem basis data MySQL.

6. **Amazon EC2 (*Elastic Compute Cloud*).** Digunakan sebagai infrastruktur (kapasitas pemrosesan, memori, dan ruang hardisk) yang menyediakan layanan (*service*) yang dibutuhkan oleh para pengguna.

C. Kelebihan dan Kekurangan AWS

Kelebihan.

1. Aplikasi-aplikasi AWS yang ditulis menggunakan bahasa-bahasa pemrograman PHP, Ruby, serta Java, dapat dikembangkan dengan cara yang sangat fleksibel karena pengguna memiliki kendali penuh pada sistem yang mendasari.
2. Struktur pembiayaan sederhana.
3. Bisa menggunakan sistem basis data (relasional maupun non-relasional) apa saja yang dibutuhkan oleh pengguna.
4. Jika pengguna mau, pengguna bisa saja menggunakan/menambahkan server-server yang berada di luar Amazon Web Service.

Kekurangan

1. Kurva belajar yang terjal (relatif sulit untuk mempelajari pengembangan aplikasi-aplikasi di atas Amazon Web Service dibandingkan di atas Google App Engine).
2. Memerlukan waktu yang relatif lebih lama untuk mengembangkan aplikasi (bahkan untuk aplikasi-aplikasi yang relatif sederhana).

BAB III
Contoh Metode (review 1 jurnal)

NO	Penulis/Tahun	Nama Jurnal	Tujuan Penelitian	Metode penelitian	Hasil penelitian
1	Sastra Ginata[1], Ari Kusyanti[2], Rakhmadhany Primananda[3] Tahun 2019	Implementasi Algoritme Kriptografi SIMON pada Arsitektur Amazon Web Services	Pengujian ini bertujuan untuk mengetahui berapa lama waktu yang dibutuhkan pada saat proses enkripsi dan dekripsi *file.*	Kuantitatif	Pengujian ini dilakukan untuk memastikan bahwa *file* dapat dienkripsi dan didekripsi sesuai dengan algoritme yang digunakan. Validasi dilakukan dengan cara memeriksa hasil dekripsi dengan *file* sebelum dilakukan proses enkripsi. Proses enkripsi menggunakan *key* "gigin123" dan menggunakan *file* teks agar dapat dibaca dengan mudah. Pengujian dilakukan dalam 3 skenario yang memiliki *plaintext* berbeda.

Kesimpulan

1. Algoritme SIMON dapat diterapkan pada arsitektur Amazon Web Services untuk mengamankan data yang akan dikirim dan dapat menjadi alternatif sebagai algoritme kriptografi untuk mengamankan data.
2. Algoritme SIMON yang diterapkan penulis memiliki hasil *output ciphertext* yang sama dengan hasil *test vector* pada jurnal dan dapat dipastikan bahwa algoritme yang digunakan pada penelitian ini bersifat *valid.*
3. 1. Algoritme SIMON yang diterapkan penulis memiliki hasil pengujian yang sesuai dengan parameter pengujian yang diharapkan.

Pada pengujian kinerja waktu, hasil yang didapatkan adalah rata-rata yang paling cepat proses enkripsi dan dekripsi *file* yaitu sebesar 33,7 detik dan 28,5 detik. Pada pengujian validasi enkripsi dan dekripsi, hasil yang didapatkan adalah algoritme SIMON dapat mengenkripsi *plaintext* menjadi bentuk *ciphertext* dan hasil *ciphertext* dapat dikembalikan lagi menjadi *plaintext.* Pada pengujian variasi *key,* hasil yang didapatkan adalah perbedaan *key* yang digunakan tidak berpengaruh pada waktu tempuh enkripsinya. Pada pengujian variasi *file,* hasil yang didapatkan adalah algoritme SIMON dapat mengenkripsi *file* dengan ekstensi: .txt, .docx, .pdf, .png, .jpg, .mp3,.m4a, .mp4 dan .mkv. Pada pengujian waktu *upload file,* hasil yang didapatkan adalah sistem dapat mengeksekusi proses *upload* selama kurang dari 5 menit.

4. Algortime SIMON yang diterapkan penulis memiliki tingkat kemanan yang baik terhadap setiap jenis file yang telah diuji. Pada pengujian serangan brute force, hasil yang didapatkan adalah tool Rainbowcrack tidak dapat menemukan plaintext.

BAB IV

Penutup (perbandingan dengan metode waterfall,prototype, RAD)

1. Model Waterfall
   Menurut kelompok kami metode waterfall cocok digunakan untuk sistem atau perangkat lunak yang bersifat generik, artinya sistem dapat diidentifikasi semua kebutuhannya dari awal dengan spesifikasi yang umum serta sesuai untuk perangkat lunak yang memiliki tujuan untuk membangun sebuah sistem dari awal yang mengumpulkan kebutuhan sistem yang akan dibangun sesuai dengan topik penelitian yang dipilih sampai dengan produk tersebut diuji
2. Model Prototype
   Menurut kelompok kami Model Prototype lebih cocok untuk sistem atau perangkat lunakyang bersifat customize, artinya software yang diciptakan berdasarkan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai untuk perangkat lunak memiliki tujuan untuk mengimplementasikan sebuah metode atau algoritma tertentu pada suatu kasus.
3. Model RAD
   Menurut kelompok kami model RAD lebih cocok untuk sistem atau perangkat lunak yang bersifat customize, berskala besar dan memerlukan waktu yang lebih singkat artinya software yang diciptakan berdasarkan permintaan dan kebutuhan (bahkan situasi atau kondisi) tertentu dan sesuai untuk perangkat lunak memiliki tujuan untuk mengimplementasikan sebuah metode atau

algoritma tertentu pada suatu kasus, serta memiliki kemungkinan untuk kebutuhan pengembangan kembali dalam jangka waktu yang cukup panjang.