Penerapan HMAC (*Hash-Message Authentication Code*) Sebagai Penjamin Keaslian Review Digital

Muhammad Angga Risfanani – 13518071
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): muhanggarisfanani@gmail.com

***Abstrak*—Pada zaman sekarang, berbagai toko telah membuka stand online pada website-website untuk memasarkan produk mereka. Sebagian memiliki fitur yang mencantumkan review dari pembeli di website mereka. Namun terkadang, terpikirkan oleh kita apakah review tersebut asli atau tidak. Bagaimana apabila pihak toko dengan sengaja mengubah isi dari review para pelanggannya demi kepentingan toko mereka sendiri. Dengan pemberi review bertindak sebagai pengirim dan penerima pesan sekaligus, ditambah dengan kecenderungan manusia yang lemah dalam mengingat teks yang panjang (dalam kasus ini adalah reviewnya sendiri), dapat dibuat sebuah sistem yang dapat digunakan untuk menjamin keaslian review-review tersebut menggunakan algoritma *Hash-Message Authentication Code* (HMAC).**

Kata kunci—MAC; Hash; HMAC; Review; Komentar;

I. PENDAHULUAN

Trend belanja online (*online shopping*) sudah sangat umum di zaman sekarang ini. Banyak dewasa muda sampai dewasa tua yang sering membelanjakan uangnya untuk memenuhi kebutuhan hidupnya lewat toko online. Tidak jarang pula terdapat fitur yang disebut dengan review produk atau komentar. Biasanya, review produk atau komentar dapat diberikan setelah kita membeli barang dari toko online tersebut. Adanya review ini cukup penting mempertimbangkan bahwa pelanggan dapat memberikan penilaian terhadap kinerja atau performa toko secara langsung dari pelanggan lainnya. Sehingga pelanggan dapat memilih toko terbaik yang menjual produk terbaik berdasarkan *real-time review* dari pelanggan pelanggannya.

Bentuk review tersebut pun bermacam-macam, ada yang berupa nilai *rating* dengan skala 0-5, ada pula review yang berupa teks, dan ada pula yang merupakan gabungan dari keduanya. Contoh review berupa gabungan nilai *rating* dan teks dapat dilihat pada Gambar 1, yang menunjukkan fitur review dari aplikasi Tokopedia.

Sebuah fitur review tentu seharusnya dapat menggambarkan kualitas suatu produk atau toko. Namun, bagaimana jika apa yang ditulis oleh pelanggan dengan apa yang di cantumkan oleh penjual tidak sama. Dengan kata lain, toko/penjual melakukan modifikasi terhadap review pelanggannya agar tokonya terlihat tetap baik dan berkualitas padahal pernah ada review buruk dari pelanggannya.

Hal ini dapat diatasi sebenarnya hanya dengan si pelanggan mengingat kembali apa yang ditulisnya, lalu mengeceknya secara manual apakah ada modifikasi atau tidak terhadap review yang ia berikan. Namun, apabila reviewnya cukup panjang, tentu lebih sulit untuk diingat. Oleh karena itu, permasalahan ini perlu penerapan MAC (*Message Authentication Code*) untuk menjamin keaslian dari suatu review atau komentar.

Gambar 1. Contoh review produk online pada aplikasi Tokopedia (Sumber: https://www.nesabamedia.com/cara-belanja-di-tokopedia/)

II. TEORI DASAR

2.1 Hash

Dalam ilmu kriptografi, hash adalah algoritma yang dipakai untuk mengubah informasi. Data yang dimasukkan nantinya diolah menjadi angka, huruf, atau karakter lain menjadi karakter terenkripsi tanpa mengubah ukuran. Data yang terenkripsi lewat fungsi hash tak bisa lagi dikembalikan. Hal ini pula yang membuat algoritma tersebut dikenal sebagai One Way Function atau encryption satu arah. [2]

Kompleksitas dan sensitivitas dalam penerapan fungsi hash membuatnya dimanfaatkan dalam sejumlah bidang. Adapun alasan-alasan yang membuat proses hash penting, di antaranya:

1. Menjaga Integritas Data

   Seperti yang disinggung, fungsi hash sangat sensitif. Perubahan sedikit saja akan mengubah kode yang disusun. Di sisi lain, sifat tersebut membuat peluang munculnya hash yang sama lebih kecil. Oleh karena itu, hashing digunakan untuk menjaga integritas data yang sifatnya sangat rahasia, sebagai contoh password yang dipakai dalam ATM atau sistem lain yang melibatkan data personal. Semakin kompleks susunan yang dihasilkan dari fungsi hash, semakin kecil kemungkinan sistem ditembus oleh hacker.

2. Mempercepat proses pengiriman

   Salah satu contoh yang bisa dilihat dari fungsi hash dalam hal ini adalah verifikasi salinan arsip dengan yang asli. Proses pengiriman akan merepotkan apabila salinan dokumen berada di lokasi yang jauh dari basis data arsip asli. Alih-alih mengirim salinan dokumen secara keseluruhan ke komputer pusat yang memerlukan waktu transmisi lama, manfaatkan fungsi hash yang tepat untuk mengirimkan message digest dari arsip. Jika message digest dari salinan cocok dengan message digest dari arsip asli, bisa dipastikan kedua dokumen tersebut sama.

3. Menormalkan keberagaman panjang data

   Ketika membuat akun di marketplace atau aplikasi, biasanya akan diminta membuat password dengan panjang minimal, misalnya dari 5 hingga 8 karakter dengan jenis berbeda. Password lantas akan disimpan dalam server untuk memudahkan proses otentikasi sebagai pengguna laptop atau PC. Kemudian untuk menyeragamkan panjang field untuk password dalam basis data, karakter-karakter password tadi akan disimpan dalam nilai hash yang panjangnya tetap. Tampilan password yang sudah diolah fungsi hash biasanya berupa dot hitam (•).

4. Menjadi label dan identitas dari bukti digital

   Dalam ranah hukum, penerapan proses hashing atau fungsi hash akan membantu dalam penyimpanan bukti digital. Hal ini disebabkan fungsi hash dapat digunakan sebagai label dan identitas pada bukti tersebut dan mempunyai probabilitas identik hingga 99,99%. Selain untuk kepentingan hukum, fungsi hash juga diandalkan untuk orang-orang yang bekerja di bidang forensik. Pasalnya, mereka harus menjaga barang bukti yang ditemukan memakai data-data yang diolah menjadi kode unik lewat proses hashing. [2]

Jenis jenis fungsi hash untuk kebutuhan di dunia kriptografi terbilang beragam, tetapi ada beberapa jenis yang paling umum dipakai, antara lain:

1. MD5
2. SHA-1
3. RIPEMD-160
4. SHA-2
5. SHA-3 [2]

Berbagai macam jenis fungsi hash tersebut memiliki kelemahan dan kelebihannya masing-masing, dan penggunaannya disesuaikan dengan kebutuhan penerapan aplikasi dalam kriptografinya.

2.2 MAC (Message Authentication Code)

MAC (Message Authentication Code) merupakan kode yang dihasilkan oleh fungsi hash satu-arah namun menggunakan kunci rahasia (*secret key*) dalam pembangkitan nilai hashnya. [1]. MAC memiliki formula perhitungan sebagai berikut:

$$\mathbf{MAC = C_K(M)}$$

MAC: nilai hash

C: fungsi hash (atau algoritma MAC)

K: kunci rahasia

Fungsi hash jenis MD5 dan SHA tidak dapat digunakan dalam MAC karena tidak memiliki kunci rahasia dalam menghasilkan nilai hashnya. [1]

MAC sangat berguna untuk menjaga identitas dan integritas dari sebuah pesan apapun. Dengan melekatkan nilai MAC pada pesan, dapat dilakukan perhitungan nilai MAC antara pengirim dan penerima, jika nilai MAC sama, maka pesan masih asli. Sedangkan jika nilai MAC berbeda, maka pesan telah terjadi modifikasi. Pengirim dan penerima perlu untuk menyepakai nilai K yang akan digunakan. [1]

2.3 HMAC (Hash Message Authentication Code)

HMAC adalah salah satu jenis MAC yang menggunakan fungsi hash satu-arah sebagai basis algoritmanya. Pengirim dan penerima yang akan saling bertukar data atau informasi perlu menyepakati sebuah nilai K yang akan dijadikan kunci rahasia keduanya, lalu menghitung nilai HMAC dari data atau informasi yang akan ditukarkan menggunakan kunci rahasia K. Setelah terhitung, nilai HMAC disisipkan pada pesan, proses penyisipan juga dapat dilakukan di awal maupun di akhir pesan dengan menambahkan baris baru dengan tag tertentu misalnya <hmac>…</hmac>. [1]

2.4 Review dan Komentar

Review adalah sebuah ringkasan, tinjauan dari beberapa sumber entah dari film, buku, berita dan lainnya. Seperti review buku, review film, review produk dan lainnya. Dengan adanya review dapat diketahui kelebihan, kekurangan dan kualitas dari suatu karya atau produk. Tujuannya adalah untuk memberi informasi ke pembaca tentang suatu hal dan mengajaknya atau membuat pembeca semakin penasaran, review ini sangat penting untuk pemasaran suatu produk tertentu. [3]

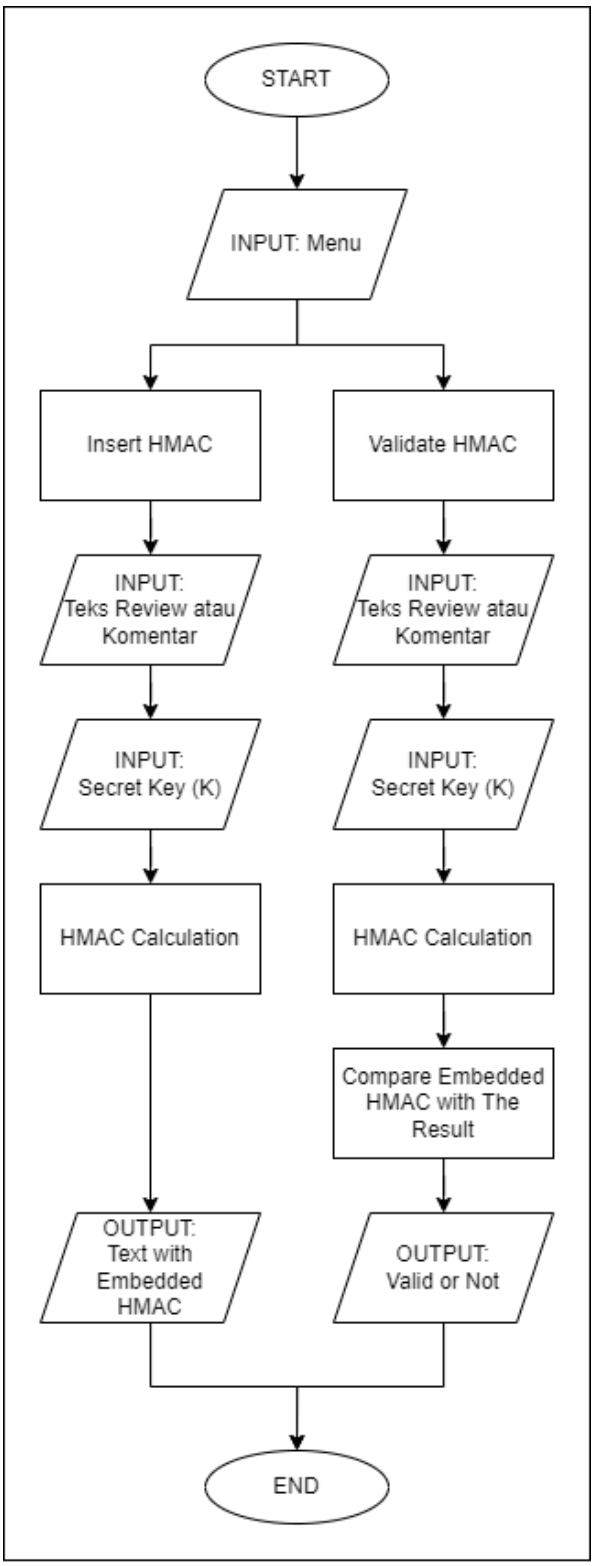
Sedangkan komentar adalah ulasan atau tanggapan atas berita, pidato, dan sebagainya (untuk menerangkan atau menjelaskan). [4]

III. PEMBAHASAN SOLUSI

Dalam permasalahan yang telah dipaparkan sebelumnya, dengan peran pemberi komentar/review adalah sebagai pengirim sekaligus sebagai penerima pesan, akan dibuat sebuah solusi yang mengatasi permasalahan tersebut. Akan dibuat sebuah program dengan spesifikasi sebagai berikut:

1. Menu pilihan untuk:
   - menyisipkan HMAC ke dalam teks
   - memvalidasi keaslian dari pesan yang telah tersisip HMAC.
2. Menerima input berupa pesan teks review atau komentar yang akan dijamin keasliannya.
3. Menerima input berupa nilai K sebagai kunci rahasia yang digunakan oleh pengirim dan penerima.
4. Mengeluarkan pesan yang telah tersisip oleh nilai HMAC.
5. Menerima input berupa pesan teks review yang tersisip oleh nilai HMAC.
6. Menerima input berupa nilai K sebagai kunci rahasia yang digunakan oleh pengirim dan penerima.
7. Mengeluarkan hasil pengecekan kevalidan suatu teks review/komentar.

Sehingga dari spesifikasi program di atas, dapat dibuat sebuah diagram alir dari program yang diimplementasikan, dapat dilihat pada Gambar 2.



Gambar 2. Diagram Alir Program Penjamin Keaslian Review

(sumber: dokumentasi penulis)

Program dibuat dalam bahasa pemrograman Python 3 dengan menggunakan bantuan library hashlib dan hmac dari python. Proses *hashing* dilakukan dengan menggunakan algoritma SHA256 menggunakan kunci rahasia K.

Didefinisikan 2 fungsi yaitu fungsi InsertHMAC untuk memasukkan nilai HMAC ke dalam review teks yang akan dijamin integritasnya, dan fungsi validateHMAC untuk mengecek keaslian review yang telah disisipkan nilai HMAC. Bentuk sisipan nilai HMAC dilakukan dengan menambahkan tag <hmac>…</hmac> pada bagian awal teks dengan tanda spasi sebagai pemisah.

Pada fungsi insertHMAC terdapat 2 parameter yaitu nilai kunci rahasia K dan pesan teks review. Contoh teks review sebelum disisipi dan setelah disisipi oleh HMAC sebagai berikut:

Tabel 1. Contoh review sebelum penyisipan (input)

Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau. Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau. Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau.

Tabel 2. Contoh review setelah penyisipan (output)

<hmac>5e4d96aa7d5a6218a0b563fd7d30f85a1469d42d179b4d6e4747696d6f1850d6</hmac> Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau. Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau. Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau.

Proses Penyisipan dilakukan dengan melakukan encode teks review dan kunci rahasia menggunakan method .encode(), lalu mem-*passing*-nya sebagai parameter objek hmac.new, lalu melakukan proses hashing menggunakan method .hexdigest(). Setelah proses perhitungan HMAC selesai, nilainya dikonversikan menjadi string di dalam tag <hmac> dan </hmac>. Lalu melakukan konkatenasi dengan review teks yang asli.

Sedangkan fungsi validateHMAC juga terdapat 2 parameter yaitu nilai kunci rahasia K dan teks review yang telah tersisipi oleh tag <hmac> dan </hmac>. Contoh input dan output dari fungsi ini sebagai berikut:

Tabel 3. Contoh review yang tersisipi nilai HMAC

<hmac>5e4d96aa7d5a6218a0b563fd7d30f85a1469d42d179b4d6e4747696d6f1850d6</hmac> Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau. Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau. Barang ini kurang bagus, baru sekali pakai sudah luntur. Padahal harganya aja mahal, gimana sih. Jadinya mau dibalikin tapi penjualnya gamau.

Tabel 4. Contoh hasil validasi keaslian teks

Valid

Proses validasi dilakukan dengan mengambil tag <hmac> dan </hmac> yang ada di kata pertama sebelum spasi pertama, lalu mengambil nilai isi tag tersebut sebagai embedded_hmac. Sedangkan teks setelah spasi pertama diambil sebagai review teks yang hendak dicek. Dimulai dengan menghitung review teks yang telah di pisahkan dengan tag hmac nya, lalu dibandingkan nilai hmac yang baru dihitung dari teks review dengan nilai embedded_hmac yang tersisip. Jika sama, di keluarkan hasil bahwa review valid, dan sebaliknya jika tidak sama, maka review tidak valid dan telah terjadi modifikasi atau perubahan.

IV. PENGUJIAN DAN ANALISIS

Setelah aplikasi penjamin keaslian review diimplementasikan pada bagian III. Dilakukan pengujian terhadap beberapa kasus kecil yang mungkin terjadi dalam kehidupan nyata. Kasus-kasus tersebut adalah sebagai berikut:

4.1 Toko online yang jujur dan apa adanya

Pada kasus ini, penjual tidak merubah review yang ditampilkan ke websitenya.

Pelanggan menggunakan aplikasi untuk membuat review:

```
   MENU
   1. Insert HMAC
   2. Validate Review
   Pilihan (1/2): 1
   Masukkan nilai kunci rahasia K: 123
   Masukkan review yang akan dijaga keasliannya: Jelek
banget ini, harganya mahal lagi.
   Hasil review yang baru (HMAC embedded)
   ##########################################################
##########
   <hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec
102cf3c0da3384e5</hmac> Jelek banget ini, harganya mahal
lagi.
##########################################################
#######
```

Lalu Toko mengunggah review pelanggan tersebut:

```
   <hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec
102cf3c0da3384e5</hmac> Jelek banget ini, harganya mahal
lagi.
```

Lalu pelanggan tersebut mengecek keaslian review yang diunggah oleh toko:

```
   MENU
   1. Insert HMAC
   2. Validate Review
   Pilihan (1/2): 2
   Masukkan nilai kunci rahasia K: 123
   Masukkan review yang mengandung tag <hmac> dan
</hmac>:
<hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec102
cf3c0da3384e5</hmac> Jelek banget ini, harganya mahal
lagi.
Valid
```

Yang menghasilkan hasil **valid** karena nilai hmac review yang diunggah dengan nilai hmac yang disisipkan sama**.**

4.2 Toko online yang nakal tetapi tidak tahu teknologi

Pada kasus ini, penjual merubah isi dari review pelanggannya, tetapi tidak menghilangkan nilai tag HMAC dalam reviewnya.

Pelanggan menggunakan aplikasi untuk membuat review:

```
   MENU
   1. Insert HMAC
   2. Validate Review
   Pilihan (1/2): 1
   Masukkan nilai kunci rahasia K: 123
   Masukkan review yang akan dijaga keasliannya: Jelek
banget ini, harganya mahal lagi.
   Hasil review yang baru (HMAC embedded)
   ########################################################
##########
   <hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec
102cf3c0da3384e5</hmac> Jelek banget ini, harganya mahal
lagi.
########################################################
#######
```

Lalu Toko mengunggah review pelanggan tersebut:

```
   <hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec
102cf3c0da3384e5</hmac> Bagus banget ini, harganya murah
lagi.
```

Perhatikan kata jelek telah dimodifikasi menjadi bagus, dan kata mahal dimodifikasi dengan kata murah.

Lalu pelanggan tersebut mengecek keaslian review yang diunggah oleh toko:

```
   MENU
   1. Insert HMAC
   2. Validate Review
   Pilihan (1/2): 2
   Masukkan nilai kunci rahasia K: 123
   Masukkan review yang mengandung tag <hmac> dan
</hmac>:
<hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec102
cf3c0da3384e5</hmac> Bagus banget ini, harganya murah
lagi.
Tidak Valid
```

Yang menghasilkan hasil **tidak valid** karena nilai hmac review yang diunggah dengan nilai hmac yang disisipkan tidak sama karena ada perubahan pada review teks**.**

4.3 Toko online yang nakal dan tahu teknologi

Pada kasus ini, penjual merubah isi dari review pelanggannya, karena ia tahu bahwa nilai tag yang dicantumkan merupakan identitas dari review pelanggan, maka ia hapus identitasnya agar tidak dapat diidentifikasi keasliannya dan seolah-olah review tersebut adalah asli.

Pelanggan menggunakan aplikasi untuk membuat review:

```
   MENU
   1. Insert HMAC
   2. Validate Review
   Pilihan (1/2): 1
   Masukkan nilai kunci rahasia K: 123
   Masukkan review yang akan dijaga keasliannya: Jelek
banget ini, harganya mahal lagi.
   Hasil review yang baru (HMAC embedded)
   ########################################################
##########
   <hmac>2d6a23eb900b3806fdbecb62d424c1892e08b542f6228eec
102cf3c0da3384e5</hmac> Jelek banget ini, harganya mahal
lagi.
########################################################
#######
```

Lalu Toko mengunggah review pelanggan tersebut:

```
   Bagus banget ini, harganya murah lagi.
```

Perhatikan kata jelek telah dimodifikasi menjadi bagus, dan kata mahal dimodifikasi dengan kata murah, serta nilai tag hmac yang dihilangkan.

Lalu pelanggan tersebut mengecek keaslian review yang diunggah oleh toko:

```
   MENU
   1. Insert HMAC
   2. Validate Review
   Pilihan (1/2): 2
   Masukkan nilai kunci rahasia K: 123
   Masukkan review yang mengandung tag <hmac> dan
</hmac>: Bagus banget ini, harganya murah lagi.
Tidak Valid
```

Yang menghasilkan hasil **tidak valid** karena nilai hmac review yang diunggah dengan nilai hmac yang disisipkan tidak sama karena terdapat perubahan pada review teks dan penghilangan/penggantian tag hmac**.**

V. KESIMPULAN

Dari penerapan aplikasi dan pengujian di atas, dapat disimpulkan bahwa:

1. Aplikasi berhasil menjaga integritas review atau komentar digital dan dapat mencegah kecurangan oknum-oknum yang ingin menang sendiri.
2. Untuk penerapan lanjutan, perlu diperhatikan untuk peletakan penyisipan, karena dengan menggunakan tag hmac di awal teks seperti yang telah dipaparkan, bentuk teks review jadi lebih panjang dan terlihat berantakan, sehingga review menjadi lebih susah untuk dibaca.

VI. UCAPAN TERIMA KASIH

Penulis ingin mengucapkan rasa syukur sebesar-besarnya kepada Allah, Tuhan yang maha Esa. Karena telah memberikan kesempatan kepada penulis untuk menyelesaikan makalah ini.

Penulis juga ingin menyampaikan rasa terima kasih sebesar-besarnya kepada pak Rinaldi Munir, karena telah membimbing penulis selama menempuh kuliah IF4020 Kriptografi pada program studi Teknik Informatika di Institut Teknologi Bandung. Dan juga telah menyediakan waktunya untuk membuat artikel-artikel yang berguna dan membantu dalam proses penyelesaian makalah ini.

Penulis juga ingin mengucapkan rasa terima kasih kepada penulis-penulis website artikel tempat penulis mendapatkan referensi.

Semoga apapun yang kita pelajari selama ini dan semoga dengan makalah ini dapat membatu orang lain dalam menyelesaikan tugas-tugasnya, dan mudah-mudahan menambah ilmu kita dan mereka.

REFERENSI

[1] Munir, Rinaldi. 2021. Slide Kuliah IF4020 Kriptografi MAC, Bandung.

[2] https://bitocto.com/octopedia/apa-itu-hash/ (diakses 20 Desember 2021 pukul 21.15)

[3] https://www.dosenpendidikan.co.id/review-adalah/ (diakses 20 Desember 2021 21.20)

[4] Kamus Besar Bahasa Indonesia

[5] https://www.pythoncentral.io/hashing-strings-with-python/ (diakses 20 Desember 2021 22.10)

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Sidoarjo, 20 Desember 2021

Muhammad Angga Risfanani
NIM: 13518071