IMPLEMENTASI KODE *REED-MULLER* PADA TANDA TANGAN DIGITAL BERBASIS *MD5*

SKRIPSI

OLEH:
MAULANA AGIL YULIARSO
NIM. 19610103

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

IMPLEMENTASI KODE *REED-MULLER* PADA TANDA TANGAN DIGITAL BERBASIS *MD5*

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Maulana Agil Yuliarso
NIM. 19610103

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

IMPLEMENTASI KODE *REED-MULLER* PADA TANDA TANGAN DIGITAL BERBASIS *MD5*

SKRIPSI

Oleh
Maulana Agil Yuliarso
NIM. 19610103

Telah Disetujui Untuk Diuji

Malang, 16 Oktober 2025

Dosen Pembimbing I	Dosen Pembimbing II
Muhammad Khudzaifah, M.Si	Erna Herawati, M.Pd
NIPPPK. 19900511 202321 1 029	NIPPPK. 19760723 202321 2 006

Mengetahui,
Ketua Program Studi Matematika

Ahmad Roziqin, M.Si
NIP. 19800527 200801 1 012

IMPLEMENTASI KODE *REED-MULLER* PADA TANDA TANGAN DIGITAL BERBASIS *MD5*

SKRIPSI

Oleh
Maulana Agil Yuliarso
NIM. 19610103

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 31 Oktober 2025

Ketua Penguji	:	Hisyam Fahmi, M.Kom.	.....
Anggota Penguji 1	:	Mohammad Nafie Jauhari, M.Si.	.....
Anggota Penguji 2	:	Muhammad Khudzaifah, M.Si.	.....
Anggota Penguji 3	:	Erna Herawati, M.Pd	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini

Nama : Maulana Agil Yuliarso

Nim : 19610103

Progam Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Kode *Reed-Muller* Pada Tanda Tangan Digital Berbasis *MD5*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan dihalaman terakhir. Apabila di kemudian hari terbukti skripsi ini adalah hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 31 Oktober 2025

Maulana Agil Yuliarso

NIM. 19610103

MOTO

“Belajar dari kesalahan, bangkit dari kegagalan”

PERSEMBAHAN

Bismillahirrahmaanirrahim

Segala puji syukur penulis panjatkan ke hadirat Allah SWT yang dengan kasih sayang-Nya telah memberikan kemudahan, kekuatan, dan kelapangan hati dalam setiap langkah hingga skripsi ini dapat terselesaikan dengan baik.

Dengan penuh rasa hormat dan cinta, karya sederhana ini penulis persembahkan kepada:

1. Ayahanda tercinta Setiarso dan Ibunda tersayang, Patmiatun yang tanpa lelah memberikan cinta, doa, serta pengorbanan tiada henti dalam mendampingi setiap perjalanan hidup penulis. Semoga setiap langkah ini menjadi ladang pahala untuk kalian.
2. Untuk diriku sendiri, terima kasih telah bertahan sejauh ini, tetap berdiri meski dalam lelah, dan percaya bahwa segala hal indah akan datang di waktu yang tepat sesuai dengan rencana-Nya.
3. Kepada teman-teman tercinta: terima kasih atas semangat, canda, dan dukungan yang telah menguatkan langkah ini.
4. Dan kepada teman-teman seperjuangan yang telah menjadi bagian dari proses ini terima kasih atas kebersamaan, motivasi, serta doa yang selalu tulus kalian berikan.Semoga semua kebaikan kalian menjadi amal jariyah yang tidak terputus.

KATA PENGANTAR

Bismillahirrahmanirrahiim

Segala puji dan syukur kehadirat Tuhan Yang Maha Esa yang telah melimpahkan kasih, karunia, kehendak dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul Implementasi Kode *Reed-Muller* pada Tanda Tangan Digital Berbasis Kriptografi *MD5* sebagai persyaratan untuk memperoleh gelar Sarjana Matematika (S.Mat).

Sholawat serta salam semoga tetap tercurah pada Nabi Muhammad SAW keluarga, sahabat, dan para pengikutnya hingga akhir zaman. Dalam penyelesaian penyusunan skripsi ini, penulis tidak terlepas dari bimbingan, dukungan, bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini. Semoga kebaikan semuanya menjadi amal ibadah dan mendapat pahala yang berlimpah dari Allah SWT. Skripsi ini tidak akan tersusun tanpa adanya bantuan dan dorongan dari berbagai pihak, maka penulis mengucapkan terima kasih kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Agus Mulyono, M.Kes., selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Fachrur Rozi, M.Si., selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Muhammad Khudzaifah, M.Si., selaku dosen pembimbing I dan dosen pengungi II yang senantiasa sabar memberikan bimbingan, arahan, saran dan motivasi kepada penulis dalam menyusun skripsi ini.
5. Erna Herawati, M.Pd., selaku dosen pembimbing II dan dosen penguji III yang juga senantiasa sabar memberikan bimbingan, arahan, saran dan motivasi kepada penulis dalam menyusun skripsi ini.
6. Hisyam Fahmi, M.Kom., selaku ketua penguji yang telah memberikan kritik, saran serta dukungan kepada penulis.

7. Mohammad Nafie Jauhari, M. Si., selaku dosen penguji 1 yang telah memberikan kritik, serta dukungan kepada penulis.
8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim yang telah mendidik, membimbing dan mengajarkan ilmu-ilmunya kepada penulis.
9. Kedua orang tua dan seluruh keluarga tercinta yang telah berusaha memenuhi segala kebutuhan, arahan, pengorbanan serta dengan iringan do'anya kepada penulis.
10. Seluruh teman-teman mahasiswa angkatan 2019 yang selalu memberikan motivasi.
11. Serta semua pihak yang telah memberikan bantuan demi terselesaikannya skripsi ini.

Malang, 31 Oktober 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

ABSTRAK

Yuliarso,Maulana Agil.2025. **Implementasi Kode *Reed-Muller* Pada Tanda Tangan Digital Berbasis *MD5***. Skripsi. Progam Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.Pembimbinng: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Kata Kunci: Kriptografi, Kode *Reed-Muller*, *MD5*, Tanda Tangan Digital verifikasi.

Penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi tanda tangan digital menggunakan fungsi *hash MD5* yang dikombinasikan dengan kode *Reed-Muller* sebagai metode pengamanan dokumen elektronik. Tanda tangan digital berfungsi sebagai jaminan integritas, autentikasi, dan keaslian data dalam komunikasi digital. Dalam sistem yang dibangun, *input* pesan atau dokumen sudah dalam bentuk *hash* menggunakan algoritma *MD5* untuk menghasilkan *message digest* sepanjang 128 bit. Hasil *hash* kemudian dienkode menggunakan kode *Reed-Muller* RM$(1,\ 3)$ untuk menghasilkan tanda tangan digital berupa vektor *error*. Proses verifikasi dilakukan dengan menggunakan matriks pemeriksa paritas untuk mencocokkan sindrom yang dihasilkan dengan tanda tangan yang diterima. Simulasi dilakukan pada dokumen PDF asli dan dokumen yang dimodifikasi untuk menunjukkan bahwa sistem mampu membedakan antara dokumen asli dan dokumen yang telah diubah, serta menghasilkan tanda tangan yang unik untuk setiap dokumen, meskipun menggunakan kunci privat yang sama. Dengan demikian, implementasi kode *Reed-Muller* pada skema tanda tangan digital berbasis *MD5* terbukti efektif dalam menjaga integritas dan keaslian dokumen elektronik.

ABSTRACT

Yuliarso, Maulana Agil.2025.**Implementation of Reed-Muller Code in MD5 Based Digital Signature**. Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Keywords: Cryptography, Code Reed-Muller, MD5, Digital Signature verification.

This research aims to implement and evaluate a digital signature scheme using the MD5 hash function combined with Reed–Muller codes as a method for securing electronic documents. Digital signatures serve as a guarantee of integrity, authentication, and data authenticity in digital communication. In the system developed, the input message or document is first converted into a hash using the MD5 algorithm to produce a 128-bit message digest. The resulting hash is then encoded using the Reed–Muller $RM(1, 3)$ code to generate a digital signature in the form of an error vector. The verification process is carried out using a parity-check matrix to match the syndrome produced with the received signature. Simulations were performed on both original and modified PDF documents to demonstrate that the system can distinguish between authentic documents and those that have been altered, as well as produce a unique signature for every document, even when using the same private key. Thus, the implementation of Reed–Muller codes in the MD5-based digital signature scheme is proven to be effective in maintaining the integrity and authenticity of electronic documents.

مستخلص البحث

يوليارسو، مولانا أغيل. ٢٠٢٥. **تنفيذ شفرة ريد–مولر في التوقيع الرقمي المعتمد على *MD* 5**. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانغ. المشرف: (1) محمد حذيفة، الماجستير. (2) إيرنا هيراواتي، الماجستيرة في تعليم اللغة العربية.

الكلمات الأساسية: التشفير، شفرة ريد–مولر، *MD* 5، التوقيع الرقمي، التحقق.

هدف هذا البحث إلى تنفيذ وتقييم نظام توقيع رقمي باستخدام دالة التجزئة *MD* 5 مدموجة مع شفرة ريد–مولر كطريقة لحماية الوثائق الإلكترونية.تعمل التوقيعات الرقمية كضمان لسلامة البيانات ومصادقتها وأصالتها في الاتصالات الرقمية. في النظام الذي تم إنشاؤه ، يكون إدخال الرسائل أو المستندات بالفعل في شكل *تجزئة* باستخدام خوارزمية *MD* 5 لإنتاج *ملخص رسالة* على طول 128 بت. بعد ذلك، تم ترميز ناتج التجزئة باستخدام شفرة ريد–مولر *RM* (1،3) لإنتاج توقيع رقمي على شكل متجه خطأ. تم إجراء عملية التحقق باستخدام مصفوفة فحص التماثل لمطابقة المتلازمة الناتجة مع التوقيع المستلَم تَمَّت عملية التحقق باستخدام مصفوفة مدقق التكافؤ لمطابقة المتلازمة الناتجة مع التوقيع المستلم. تم إجراء عمليات محاكاة على كل من مستندات PDF الأصلية والمستندات المعدلة لإثبات أن النظام كان قادرا على التمييز بين المستندات الأصلية والمستندات المعدلة ، بالإضافة إلى إنشاء توقيع فريد لكل مستند ، على الرغم من أنه استخدم نفس المفتاح الخاص.وبناءً على ذلك، ثبت أن تنفيذ شفرة ريد–مولر في نظام التوقيع الرقمي المعتمد على *MD* 5 فعّال في الحفاظ على سلامة وأصالة الوثائق الإلكترونية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada era digital saat ini, Semua pesan kini sudah dalam bentuk data elektronik, mirip dengan dokumen. Dokumen elektronik adalah dokumen berupa *portable document format* (PDF) atau bentuk lainnya (Ilmiyah,2019). Dokumen elektronik mempunyai kelebihan antara lain mudah diakses dan dapat diterima secara instan oleh siapa pun, akan tetapi dapat membuka peluang untuk terjadinya penyadapan. Oleh karena itu, penting bagi kita untuk memastikan bahwa dokumen tersebut diterima kepada orang yang tepat seperti halnya mengusung konsep tentang amanah. Amanah adalah sesuatu yang dipercayakan kepada seseorang, yang harus dijaga dan disampaikan dengan baik kepada yang mempunyai haknya. Karena dalam (QS. Al-Ahzab: 72) :

اِنَّا عَرَضْنَا الْاَمَانَةَ عَلَى السَّمٰوٰتِ وَالْاَرْضِ وَالْجِبَالِ فَاَبَيْنَ اَنْ يَّحْمِلْنَهَا وَاَشْفَقْنَ مِنْهَا وَحَمَلَهَا الْاِنْسَانُۗ اِنَّهٗ كَانَ ظَلُوْمًا جَهُوْلًاۙ ﴿٧٢﴾

Artinya: *Sesungguhnya Kami telah menawarkan amanat kepada langit, bumi, dan gunung-gunung, maka semuanya enggan untuk memikul amanat itu dan mereka khawatir akan mengkhianatinya dan dipikullah amanat itu oleh manusia*. *Sesungguhnya ia (manusia)* (Kemenag, 2019).

Dalam ayat ini, diajarkan bagaimana cara untuk menyampaikan pesan, tanpa mengurangi atau menambah isinya, dan menjaga isi pesan tersebut, serta memastikan pesan tersebut sampai kepada orang yang tepat. Oleh karena itu, ketika mengirimkan dokumen, penting untuk memastikan bahwa dokumen-dokumen penting seperti kontrak, perjanjian, atau komunikasi resmi tetap terjaga keasliannya

dan tidak dimanipulasi, salah satunya dengan menggunakan tanda tangan digital.

Sama halnya tanda tangan konvensional, tanda tangan digital adalah sebuah teknik kriptografi tanda tangan elektronik yang bisa membuktikan identitas pengirim pesan atau dokumen. Tanda tangan digital harus memiliki fungsi yang sama dengan tanda tangan konvensional yaitu dapat menjamin autentikasi, integritas, dan anti-penyangkalan. Tanda tangan digital menggabungkan dua algoritma sekaligus dalam implementasinya. Algoritma pertama adalah algoritma fungsi *hash* untuk membentuk *message digest* dari sebuah dokumen teks, dan algoritma kunci publik yang digunakan untuk mengenkripsi *message digest*. Proses utama tanda tangan digital terdiri dari dua proses, yaitu proses *signing* yang dilakukan dengan merubah pesan menjadi verifikasi *message digest*, dan mengenkripsinya dengan menggunakan kode *Reed-Muller* untuk menyisipkan pesan ke dalam kode dan menemukan representasi valid (Muttoo & Kumar, 2013). Sementara proses verifikasi dilakukan dengan membandingkan hasil deskripsi pesan menggunakan kunci publik untuk memeriksa apakah tanda tangan valid terhadap pesan (Saepulrohman & Negara, 2021).

Algoritma fungsi *hash* adalah metode atau prosedur matematika yang mengambil *input* data dalam bentuk heksadesimal dan mengubahnya menjadi *output* data berukuran tetap, yang biasanya berupa *string* angka. Salah satu fungsi *hash* yang umum digunakan adalah *MD5* (*message digest* 5) karena *MD5* merupakan algoritma yang sederhana, cepat, dan banyak tersedia di berbagai bahasa pemrograman, sehingga memudahkan proses implementasi, terutama dalam konteks penelitian atau simulasi. Salah satu keunggulan utama *MD5* adalah kemampuannya menghasilkan keluaran yang tetap sepanjang 128 bit, yang sangat

cocok digunakan dalam proses penyandian menggunakan kode *Reed–Muller*. Ukuran *hash* yang tetap dan relatif pendek ini memudahkan pemetaan bit-bit hasil *hash* ke dalam bentuk vektor yang akan diproses oleh matriks pengkodean maupun matriks pemeriksa kesalahan dalam sistem tanda tangan digital (Zaatsiyah & Djuniadi, 2021).

Selain itu, *MD5* bersifat deterministik, yang berarti setiap pesan yang sama akan selalu menghasilkan nilai *hash* yang sama. Sifat ini penting dalam proses verifikasi tanda tangan digital, karena sistem memerlukan kecocokan antara hasil *hash* pesan yang diterima dan hasil decoding tanda tangan digital yang dilakukan menggunakan kode *Reed–Muller*. Dengan begitu, keakuratan dan konsistensi verifikasi dapat terjaga.

Namun demikian, perlu dicatat bahwa *MD5* memiliki kelemahan dari sisi keamanan. *MD5* telah terbukti rentan terhadap serangan *collision*, di mana dua pesan berbeda dapat menghasilkan *hash* yang sama. langkah-langkah pembuatan *message digest* secara garis besar yaitu penambahan bit-bit pengganjal (*padding bits*), penambahan nilai panjang pesan semula, inisialisasi penyangga (*buffer*) *MD*, dan pengolahan pesan dalam blok berukuran 512 bit (Rangkuti & Fahmi, 2020).

Kode *Reed-Muller* merupakan salah satu jenis kode koreksi kesalahan yang banyak digunakan dalam sistem komunikasi dan keamanan informasi karena memiliki struktur matematis yang sederhana namun kuat. Kelebihan utama kode *Reed-Muller* adalah kemampuannya dalam melakukan proses *encoding* dan *decoding* yang efisien, serta daya koreksi kesalahan yang cukup tinggi, terutama untuk tingkat ordo rendah (Muttoo & Kumar, 2013). Struktur linier dan sistematis dari kode ini juga memudahkan implementasi dalam perangkat lunak maupun

perangkat keras. Selain itu, dalam konteks kriptografi berbasis kode, *Reed-Muller* dapat disamarkan dengan transformasi linear dan permutasi sehingga dapat digunakan untuk membangun sistem kriptografi asimetris yang aman. Namun demikian, kode *Reed-Muller* juga memiliki beberapa kekurangan, seperti panjang kode yang cukup besar dibandingkan kode lain dengan tingkat proteksi yang sama, serta ukuran kunci publik yang cenderung besar ketika digunakan dalam skema tanda tangan digital, sehingga dapat menjadi beban dalam penyimpanan atau transmisi.

Meskipun memiliki beberapa keterbatasan, kode *Reed-Muller* tetap menjadi pilihan yang kuat untuk diterapkan pada sistem tanda tangan digital, terutama dalam skema kriptografi *post-quantum*. Hal ini karena proses *decoding* hanya dapat dilakukan oleh pemilik kunci privat yang mengetahui transformasi rahasia, sehingga menjaga keamanan tanda tangan. Di sisi lain, fungsi *hash* seperti *MD5* digunakan untuk meringkas pesan menjadi bentuk tetap sebelum diolah oleh kode *Reed-Muller*, memastikan integritas dan keaslian data tetap terjaga. Dengan ketahanan terhadap serangan kuantum dan proses komputasi yang efisien, penggunaan kode *Reed-Muller* dalam tanda tangan digital menjadi pilihan yang logis dan relevan dalam era keamanan informasi modern (Jaya, 2017).

Berdasarkan uraian tersebut, penulis tertarik untuk mengaplikasikan kode *Reed-Muller* pada Tanda Tangan Digital berbasis *MD5* guna menciptakan algoritma yang lebih kuat dalam melindungi informasi. Penelitian ini dituangkan dalam sebuah skripsi dengan judul: “Implementasi Kode *Reed-Muller* pada Tanda Tangan Digital Berbasis *MD5*.”

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas maka diambil rumusan masalah sebagai berikut.

1. Bagaimana hasil pembentukan Tanda Tangan Digital dengan menggunakan fungsi *hash MD5* dengan kode *Reed-Muller* pada suatu dokumen?
2. Bagaimana hasil verifikasi Tanda Tangan Digital dengan menggunakan fungsi *hash MD5* dengan kode *Reed-Muller* pada suatu dokumen?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijelaskan, tujuan dari penelitian ini adalah sebagai berikut.

1. Mendekripsikan proses pembentukan tanda tangan digital menggunakan fungsi *hash MD5* yang menerapkan kode *Reed-Muller* pada sebuah dokumen.
2. Melakukan proses verifikasi tanda tangan digital menggunakan fungsi *hash MD5* dengan kode *Reed-Muller* pada sebuah dokumen.

1.4 Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian ini dijabarkan sebagai berikut:

1. Bagi Peneliti

   Memperluas pengetahuan implementasi kunci publik algoritma Tanda Tangan Digital pada dokumen.

2. Bagi Pembaca

   Dapat dijadikan referensi dan menambah koleksi pustaka bagi penelitian selanjutnya terkait Tanda Tangan digital. Penelitian ini dapat menjadi salah satu langkah untuk mengurangi potensi pemalsuan pada dokumen elektronik.

1.5 Batasan Masalah

Adapun batasan masalah dari penelitian ini dapat dijelaskan sebagai berikut:

1. Implementasi skema tanda tangan digital menggunakan kode *Reed-Muller* $RM(1,3)$.
2. *Input* dokumen sudah dalam bentuk *hash MD5*.
3. Sistem tanda tangan digital yang dikembangkan bersifat simulasi, tidak diintegrasikan dengan sistem keamanan atau infrastruktur kriptografi yang sebenarnya.
4. Implementasi menggunakan bahasa pemograman *python*.

BAB II

KAJIAN TEORI

2.1 Kriptografi

Seni untuk menjaga keamanan pesan yang dikirimkan kepada pihak lain merupakan kriptografi. Seiring dengan perkembangan tantangan dalam pengamanan data, kriptografi terus mengalami evolusi. Dalam penerapannya, terdapat beberapa istilah terkait, seperti pengiriman pesan, proses pengacakan pesan (enkripsi), dan pemulihan pesan yang telah diacak (deskripsi). Pesan yang masih dalam bentuk aslinya atau telah dikembalikan ke bentuk awal disebut *plainteks*, sedangkan pesan yang telah diacak disebut *cipherteks* (Munir, 2006).

Istilah dalam kriptografi mencakup beberapa terminologi yang sering dijumpai, yaitu pesan, *plainteks*, *cipherteks*, enkripsi, deskripsi, kunci, *cipher*, sistem kriptografi, dan kriptologi.

1. Pesan adalah data atau informasi yang berupa kata-kata, lisan maupun tulisan.
2. *Plainteks* adalah pesan asli atau pesan yang maknanya masih jelas yang berupa teks.
3. *Chiperteks* adalah pesan yang sudah dienkripsi atau tersandi.
4. *Enkripsi* adalah algoritma menyandikan *plainteks* menjadikan *chiperteks* dengan tertentu.
5. *Dekripsi* adalah algoritma mengembalikan kembali dari *chiperteks* menjadi *plainteks*.
6. Kunci *(Key)* adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa *string* atau deretan bilangan.

7. *Cipher* adalah algoritma untuk enkripsi atau dekripsi data dapat diartikan aturan *enchipering* dan *dechipering*.
8. Sistem kriptografi adalah sebuah himpunan yang terdiri dari algoritma enkripsi, dekripsi, ruang kunci, semua *plainteks* dan *chiperteks*.
9. *Cryptology* adalah ilmu yang mendasari kriptografi dan kriptanalisis yang saling berkaitan.

Kriptografi merupakan salah satu cabang ilmu yang berperan penting dalam menjaga keamanan informasi, terutama dalam proses komunikasi digital. Dalam penerapannya, kriptografi menyediakan beberapa layanan utama untuk menjamin keamanan data.

1. Kerahasiaan *(confidentiality)* memastikan bahwa pesan hanya dapat dibaca oleh pihak yang berwenang, sehingga pihak lain tidak dapat mengakses atau memahami isi pesan tersebut.
2. Integritas data *(data integrity)* bertujuan untuk menjamin bahwa pesan yang diterima tetap utuh dan tidak mengalami manipulasi selama proses pengiriman, baik secara sengaja maupun tidak sengaja.
3. Autentikasi *(authentication)* memastikan bahwa identitas pihak-pihak yang terlibat dalam komunikasi dapat diverifikasi, sehingga hanya pihak yang sah yang dapat mengakses atau bertukar informasi.
4. Anti penyangkalan *(non-repudiation)* berfungsi untuk mencegah adanya penyangkalan dari pihak pengirim atau penerima atas tindakan atau komunikasi yang telah dilakukan. Keempat layanan ini menjadi dasar penting dalam membangun sistem keamanan informasi yang handal dan terpercaya.

2.2 Algoritma *MD5*

Algoritma *MD5* (*Message Digest* 5) merupakan salah satu algoritma kriptografi yang digunakan untuk menghasilkan nilai *hash* dari suatu pesan atau data yang dikembangkan oleh Ronald L. Riverst pada tahun 1991. Dalam konteks matematika, *MD5* mengambil *input* berupa pesan berukuran sembarang dan mengubahnya menjadi keluaran berupa nilai *hash* sepanjang 128 bit (Santoso dkk., 2019). Proses ini melibatkan serangkaian operasi matematika yang rumit, termasuk operasi *bitwise*, penambahan modulo, dan pengacakan, untuk memastikan bahwa setiap perubahan kecil pada *input* akan menghasilkan *output hash* yang sangat berbeda. *Message Digest* diperoleh dengan cara sebagai berikut:

1. Penambahan *padding* bit

   Langkah pertama dalam proses *MD5* adalah penambahan pengganjal (*padding)* pesan sehingga panjangnya kongruen 448 *modulo* 512 bit.

2. Penambahan nilai panjang pesan

   Pesan yang telah ditambahkan bit-bit pengganjal membutuhkan 64 bit agar sama dengan kelipatan 512-bit. 64 bit tersebut merupakan representasi dari bilangan sebelum penambahan bit. Bit-bit ini ditambahkan ke dalam dua kata (*word*) 32 bit, dengan bit yang berurutan ditambahkan terlebih dahulu. Proses penambahan pesan ini juga dikenal sebagai MD *Strengthening* atau Penguatan MD.

3. Inisialisasi Penyangga (*Buffer*)

   Setelah pesan dibagi menjadi blok-blok 512 bit, *MD5* menginisialisasi empat penyangga 32-bit, yang akan digunakan untuk menyimpan hasil sementara

selama proses *hashing*. Penyangga ini biasanya dilambangkan sebagai a, b, c, dan d, dan diinisialisasi dengan nilai konstan:

$a = 67452301, b = EFCDAB89, c = 98BADCFE, d = 10325476$

Inisialiasi ini bukan hasil perhitungan tetapi dipilih eksplisit oleh Ron Rivest Sebagai awal penyangga bertujuan untuk memberikan keragaman dan ketidakdugaan awal dalam *hash.*

4. Pengolahan Blok Pesan

   Pada proses ini dinamakan H_{MD5} yaitu L menjadi bagian dari pesan pada setiap blok yang memiliki panjang 512 bit. Dari pesan diproses melalui empat putaran (*rounds*), yang masing-masing blok pesan bersama *buffer* MD diproses menjadi luaran 128 bit terdiri dari 16 langkah.

2.3 Tanda Tangan Digital (*Digital Signature*)

Tanda tangan digital (*Digital Signature*) terdiri dari tiga bagian. Pertama kunci publik dan kunci rahasia dihasilkan dalam proses generasi kunci. Kemudian, tanda tangan ditandatangani menggunakan kunci rahasia dalam proses penandatanganan. Terakhir, tanda tangan diverifikasi menggunakan kunci publik dalam proses verifikasi. Jika seorang penyadap membawa tanda tangan yang salah, hal itu harus diidentifikasi dalam proses verifikasi (Azdy, 2016).

Courtois, Finiasz, dan Sendrier mengusulkan skema tanda tangan digital berbasis kode pertama pada tahun 2001. Skema ini disebut skema CFS dan merupakan versi modifikasi dari kriptosistem *Niederriter* yang terdiri dari proses generasi kunci, penandatanganan, dan verifikasi.

Dalam generasi kunci, H diambil sebagai matriks pemeriksaan paritas dari (n, k). Kemampuan perbaikan kesalahan adalah $\frac{n-k}{\log n}$. Kemudian matriks acak

invertibel S dan Q dibangun. H' yang merupakan hasil kali dari S, H dan Q, menjadi kunci publik. dan S, H dan Q menjadi kunci rahasia.

Dalam proses penandatanganan, sebuah pesan m ditandatangani menjadi sindrom s dengan melakukan *hashing.* Kemudian s' dihitung dari s dengan mengalikan *invers* dari S. Proses ini diulang hingga sindrom yang didapat didekode s'ditemukan, dan dihitung dengan bilangan bulat *counter* i. Kemudian sebuah kesalahan e', yang memenuhi $s' = He'^{T}.e$ dihitung dari e' dengan mengalikan invers dari Q dan tanda tangan menjadi (m, e, i)

Dalam verifikasi diperiksa berat *Hamming* e lebih kecil atau sama dengan t dan $H'e^{T} = h(h(m)|i)$. Jika kedua kondisi ini benar, penandatanganan dianggap berhasil(Azdy, 2016).

Untuk menemukan tanda tangan yang valid, diperlukan rata-rata sebanyak t percobaan. Oleh karena itu, nilai t harus ditetapkan sekecil mungkin dari matriks acak, sehingga skema tanda tangan digital CFS untuk membuat dekoding yang efisien dan keamanan yang sudah terbukti karena menghasilkan ukuran tanda tangan yang kecil. Memiliki ukuran tanda tangan yang kecil adalah keuntungan besar dalam skema tanda tangan digital karena tanda tangan harus dikirim dalam setiap proses penandatanganan. Namun, terdapat kekurangan berupa ukuran kunci publik yang besar. Dan ini skema dari tanda tangan digital CFS.

1. Generasi Kunci (*Key Generation*)

   Proses generasi kunci dimulai dengan mendefinisikan matriks H pemeriksaan paritas (*matric check parity)* yang berukuran (n, k), Dua matriks penting yaitu, matriks *scrambler* S berukuran $(n \times n)$dan matriks. permutasi Q

berukuran $(n \times n)$, dan dibangun kunci publik H'yang dihasilkan dari $H' \leftarrow SHQ$ dan kunci rahasia terdiri dari $H, S,$ dan Q.

2. Penandatanganan (*Signing*)

   Pesan m yang akan ditandatangani di proses dengan inisialisasi variabel i ke nilai 1. Proses ini diulang, dimana i ditingkatkan dan sindrom s dihitung dengan fungsi *hash* $s \leftarrow h(h(m)|i)$. Setelah sindrom s' dihitung dengan mengalikan matriks S. Proses berlanjut hingga sindrom hingga sindrom yang dapat didekode s' ditemukan. Membuat vektor kesalahan e' dicari sehingga memenuhi persamaan $He'^T \leftarrow s'$, dan vektor kesalahan akhir e dihitung dengan menggunakan invers dari matriks Q dan menghasilkan tanda tangan berupa tuple (m, e, i).

3. Verifikasi (*Verivication*)

   Pada tahap ini membandingkan berat *hamming* dari e kurang dari atau sama dengan t, dan persamaan $H'e^T = h(h(m)|i)$ harus terpenuhi. Jika kedua kondisi benar, maka proses verifikasi berhasil dan dapat menggembalikan hasil *ACCEPT*, jika tidak hasilnya *REJECT*.

2.4 Kode *Reed-Muller*

Dalam *codes Reed-Muller* (RM) merupakan suatu kode linier dengan kode koreksi kesalahan biner yang dilambangkan $RM\ (r, m)$ dengan mengikuti format $[n, k, d]$. $RM(r, m)$ didefinisikan dengan bilangan bulat r dan m, dimana r adalah urutan kode dan $n = 2^m$ adalah panjang kode (Abbe dkk., 2021). Dimensi dari $RM(r, m)$ adalah $k = \sum_{i=0}^{r} \binom{m}{i}$ dan jarak minimum adalah $d_{min} = 2^m - r$. karena itu matriks generator G_r untuk kode RM orde $r, RM(r, m)$, dapat di ekspresikan.

$$G_r = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_m \\ v_1 v_2 \\ v_1 v_3 \\ \vdots \\ v_m - 1 v_m \\ \vdots \\ v_1 \cdots v_r \\ v_1 \cdots v_{r-1} v_{r+1} \\ \vdots \\ v_{m-r+1} \cdots v_m \end{pmatrix}$$

Dimana $v_i v_j$menunjukkan perkalian komponen-*wise* dari v_i dan v_j. Yang menunjukkan notasi dari matriks generator nyata dan matriks fungsi *Boolean.* Kolom-kolom dari matriks generator dievaluasi oleh setiap fungsi *Boolean* dari setiap baris dengan setiap vektor biner m-*tuple* dari $(1,1,\dots,1)$ hingga $(0,0,\dots,0)$ dalam urutan *leksikografis* terbalik. Matriks generator G dari kode RM orde pertama dibangun dari $\{v_0, v_1, \dots, v_m\}$, dan matriks generator RM orde kedua dibuat dari $\{v_0, v_1, \dots, v_m, v_1 v_2, v_1 v_3, \dots, v_{m-1} v_m\}$. Kode *RM* orde pertama memiliki dimensi $k = m + 1$ dan dapat direpresentasikan dengan kombinasi linier dari $v_0, \dots, v_m$. Dalam kode linier biasanya menggunakan matriks generator pada proses transmisi pesan. Adapun untuk membentuk matriks generator yang di dasarkan pada kode *Reed-Muller* dengan mempresentasikan kodenya kedalam bentuk matriks, dan matriks generatornya dinotasikan sebagai $G(r, m)$. Berikut merupakan rumus yang diterapkan untuk membentuk $G(r, m)$

$$G(0, m) = [1 \;\; 1 \;\; .. \;\; 1]$$

$$G(r, m+1) = \begin{bmatrix} G(r, m) & G(r, m) \\ 0 & G(r-1, m) \end{bmatrix}$$

$$G(m, m) = \begin{bmatrix} G(m-1, m) \\ 0 \;\; .. 0 \;\; 1 \end{bmatrix}$$

Kemudian dibawah ini beberapa contoh generator matriks berdasarkan kode *Reed-Muller.*

$$G(0,1) = [1 \;\; 1]$$

$$G(0,2) = [1 \;\; 1 \;\; 1 \;\; 1]$$

$$G(1,1) = \begin{bmatrix} G & (0,1) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$G(1,2) = \begin{bmatrix} G(1,1) & G(1,1) \\ 0 & G(0,1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Berikut contoh vektor pesan $s = [0001]$ berukuran 1×4 yang entrinya adalah bilangan biner 0 dan 1, dengan menggunakan $RM(1,3)$.

$$RM(1,3) = G(1,3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

2.5 Matriks

Matriks adalah susunan teratur bilangan-bilangan atau fungsi-fungsi matematika yang disusun dalam bentuk baris dan kolom sehingga membentuk suatu bangun persegi panjang. Ciri khas matriks terletak pada penulisan elemen-elemen penyusunnya yang selalu diapit oleh sepasang tanda kurung, baik berupa tanda kurung siku [ ] maupun tanda kurung biasa ( ), (Thresye, 2018).

Dalam notasi matematika, matriks umumnya direpresentasikan dengan menggunakan huruf kapital (seperti $A, B,$ atau C), sementara elemen-elemen penyusunnya dituliskan dengan huruf kecil (misalnya $a, b,$ atau c). Setiap matriks memiliki ukuran tertentu yang dikenal sebagai *ordo* matriks, yang menunjukkan dimensi matriks tersebut. *Ordo* matriks dinyatakan dengan notasi m × n, dimana m menunjukkan jumlah baris dan n menunjukkan jumlah kolom seperti berikut :

$$A_{m\times n} = \begin{bmatrix} a_{1\times 1} & a_{2\times 1} & \dots & a_{1\times n} \\ a_{2\times 1} & a_{2\times 2} & \cdots & a_{2\times n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m\times 1} & a_{m\times 2} & \cdots & a_{m\times n} \end{bmatrix}$$

Perlu diperhatikan bahwa tanda × dalam penulisan *ordo* berfungsi semata-mata sebagai pemisah antara jumlah baris dan kolom, bukan sebagai operator perkalian. Konsep *ordo* matriks memegang peranan penting dalam berbagai operasi matriks karena menjadi parameter utama yang menentukan validitas dan hasil dari operasi-operasi tersebut, seperti penjumlahan, pengurangan, atau perkalian matriks.

2.5.1 Jenis Matrix

Terdapat beberapa jenis matriks antara lain.

1. Matriks Identitas

   Matriks identitas atau dikenal juga dengan istilah matriks satuan merupakan matriks dengan diagonal utama yang memiliki elemen bernilai 1. Matriks ini dilambangkan dengan dinama adalah ordo dari matriks tersebut.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Matriks *Transpose*

   Jika A adalah suatu matriks $m \times n$, maka *transpose* dari A dinotasikan A^T. yaitu suatu matriks $n \times m$ yang dihasilkan dari saling menukarkan antara baris dan kolom matriks A. Dalam hal ini kolom pertama dari matriks A^T adalah baris pertama dari matriks A, kolom kedua matriks A^T adalah baris kedua matriks A dan seterusnya. Bentuk matriks ini sering juga dikenal dengan istilah matriks simetri.

$$A_{3\times1} = \begin{bmatrix} 4 \\ 7 \\ 9 \end{bmatrix} \rightarrow A^T = [4 \quad 7 \quad 9]$$

3. Matriks Invers

Invers matriks adalah misal A dan B dua matriks persegi dengan ordo yang sama. Matriks A dan B di katakan saling invers jika memenuhi $AB = BA^{-1}$ maka matriks inversnya A^{-1}. Suatu matriks dapat dibalik jika dan hanya jika matriks tersebut adalah matriks persegi (matriks yang berukuran $m \times n$) dan matriks tersebut non-singular (determinan 0). Tidak semua matriks memiliki invers. Invers matriks dapat didefinisikan sebagai berikut. Definisi: Jika A adalah suatu matriks kuadrat, dan jika kita dapat mencari matriks B sehingga $AB = BA = I$, maka A dikatakan dapat dibalik *invertible* dan B dinamakan invers dari A. Matriks-matriks persegi A dan B sedemikian hingga $AB = BA = I$ maka A disebut *invers* B ditulis B^{-1} dan sebaliknya B adalah *invers* A ditulis A^{-1} sehingga berlaku $AA^{-1} = A^{-1}A = I$, dimana I matriks identitas. Untuk mendapatkan matriks invers digunakan metode *Gauss Jordan elimination*. Dimana $(A|I) = (I|A^{-1})$ dimana A adalah matriks I adalah matriks identitas dab A^{-1} adalah matriks invers. Contoh dari penerapan *Gauss Jordan* untuk mencari invers ordo 3×3 sebagai berikut.

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 1 & 3 & 1 \\ 1 & 3 & 2 \end{bmatrix} \text{ dan } I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Langkah pertama mendampingi matriks identitas seperti berikut:

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 2 & 1 & 0 & 0 \\ 1 & 3 & 1 & 0 & 1 & 0 \\ 1 & 3 & 2 & 0 & 0 & 1 \end{array}\right)$$

Sekarang agar baris 2 kolom 1 menjadi , kita kurangkan baris kedua dengan barispertama $R_2 - R_1$.

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 2 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 1 & 3 & 2 & 0 & 0 & 1 \end{array}\right)$$

Selanjutnya agar baris 3 kolom 0 menjadi , kita kurangkan baris kedua dengan barispertama $R_3 - R_2$.

$$\left(\begin{array}{ccc|ccc} 1 & 2 & 2 & 1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 \end{array}\right)$$

Dari matriks diatas, selanjutnya kurangkan baris pertama dengan 2 kali baris kedua $R_1 - 2R_2$.

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 4 & 3 & -2 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 \end{array}\right)$$

Selanjutnya agar baris 2 kolom 3 menjadi 0, kita tambahkan baris kedua dengan baris ketiga $R_2 + R_3$.

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 4 & 3 & -2 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 \end{array}\right)$$

Langkah terakhir agar baris 1 kolom 3 menjadi 0, kita kurangkan baris pertama dengan 4 kali baris ketiga $R_1 - 4R_3$

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 3 & 2 & -4 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \end{array}\right)$$

Sehingga didapatkan matriks invers adalah sebagai berikut:

$$A^{-1} = \begin{pmatrix} 3 & 2 & -4 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix}$$

2.5.2 Operasi Matriks

1. Penjumlahan Matriks

Jika A dan B dua buah matriks berordo sama maka jumlah matriks A dan B ditulis $A + B$ adalah sebuah matriks baru C yang diperoleh dengan menjumlahkan 7 elemen-elemen matriks A dengan elemen-elemen matriks B yang seletak.

$$A_{m\times n} + B_{m\times n} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix}$$

2. Pengurangan Matriks

Pengurangan matriks A dengan matriks B adalah suatu matriks yang elemen-elemennya diperoleh dengan cara mengurangkan elemen matriks A dengan elemen matriks B yang bersesuaian (seletak), atau dapat pula diartikan sebagai penjumlahkan matriks A dengan lawan negatif dari B, dituliskan: $A - B = A + (-B)$ seperti halnya pada penjumlahan dua buah matriks, pengurangan dua buah matriks pun terdefinisi apabila ordo kedua matriks tersebut sama.

$$A_{m\times n} - B_{m\times n} = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} & \dots & a_{1n} - b_{1n} \\ a_{21} - b_{21} & a_{22} - b_{22} & \cdots & a_{2n} - b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} - b_{m1} & a_{m2} - b_{m2} & \cdots & a_{mn} - b_{mn} \end{bmatrix}$$

3. Perkalian Matriks

Jika A adalah sebuah matriks berukuran dan B adalah matriks berukuran $k \times n$, maka hasil kali AB adalah matriks berukuran yang anggota-anggotanya didefinisikan sebagai berikut : untuk mencari anggota dalam baris i dan kolom j dari matriks AB, pilih baris i dari matriks A dan kolom

j dari matriks B. Kalikan anggota-anggota yang berpadanan dari baris dan kolom secara bersama-sama, kemudian jumlahkan. Adapun gambaran untuk memperoleh entri-entri (ij) pada hasil perkalian matriks sebagai berikut :

$$\begin{bmatrix} a_{11} & \dots & a_{1k} \\ . & \dots & . \\ a_{ij} & \dots & a_{ik} \\ . & \dots & . \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} b_{11} & \dots & b_{1j} & \cdots & b_{1n} \\ . & \dots & . & \dots & . \\ . & \dots & . & \dots & . \\ . & \dots & . & \cdots & . \\ b_{k1} & \dots & b_{kj} & \dots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{in} \\ . & \dots & . \\ . & c_{ij} & . \\ . & \cdots & . \\ c_{m1} & \dots & c_{mn} \end{bmatrix}$$

Dimana $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}$.

Berikut contoh perkalian matriks $A_{1\times2}$ dengan $B_{2\times1}$ sebagai berikut:

a. Jika $A_{1\times2} = [1 \quad 2]$ dan $B_{2\times1} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$

$$A_{1\times2}B_{2\times1} = C_{1\times2}$$

$$[1 \quad 2]\begin{bmatrix} 3 \\ 1 \end{bmatrix} = C_{1\times2}$$

$$[1 \times 3 \quad 2 \times 1] = [3 \quad 2]$$

2.6 Brute Force

Brute force attack merupakan suatu metode dalam dunia kriptografi yang digunakan untuk membobol sistem keamanan dengan cara mencoba seluruh kemungkinan kombinasi *password* atau kunci enkripsi hingga ditemukan yang sesuai. Metode ini bekerja dengan mengevaluasi secara menyeluruh semua alternatif yang mungkin, tanpa mengandalkan strategi khusus untuk mempercepat proses pencarian. Dalam banyak skema kriptografi, *Brute Force* dapat digunakan untuk mencoba menembus kunci privat, terutama jika ukuran kunci tidak cukup besar atau jika mekanisme enkripsinya lemah (Zulkarnain Hasibuan, n.d.).

Salah satu bentuk dari *Brute Force* ini dikenal sebagai *exhaustive key search*, yaitu teknik yang dilakukan penyerang dengan mencoba setiap kemungkinan kunci enkripsi hingga menemukan yang benar. Misalnya, dalam algoritma *MD5* (*Message Digest 5)* panjang total kunci adalah $128 - bit$. Dengan demikian, Bila setiap percobaan kunci berukuran $4 - bit$ membutuhkan waktu 1 detik, maka seluruh ruang kunci akan memerlukan waktu yang lama.

Meskipun metode ini tidak efisien secara waktu dan sumber daya, terutama untuk skema dengan kunci panjang, keunggulan utama dari algoritma *Brute Force* termasuk teknik *exhaustive search* adalah jaminan bahwa solusi pasti ditemukan asalkan diberikan waktu yang cukup. Inilah yang membuat metode ini tetap relevan dalam studi keamanan kriptografi, terutama dalam mengevaluasi ketahanan suatu algoritma terhadap serangan berbasis kekuatan komputasi murni (Zulkarnain Hasibuan, n.d.).

2.7 Kajian Integrasi Topik dengan Al-Qur'an Dan Hadits

Kata "amanah" memiliki kaitan dengan kata "aman". Jika setiap individu menjalankan amanahnya dengan baik, maka keamanan akan terwujud di negeri dan bangsa. Amanah juga memiliki hubungan erat dengan iman, di mana iman adalah keyakinan, sedangkan amanah adalah cara untuk mengamalkan iman tersebut. Intinya, amanah dari Allah SWT terkait dengan iman adalah untuk mengikuti kebenaran yang diajarkan oleh Rasulullah SAW. Amanah ini pernah ditawarkan kepada langit, bumi, dan gunung-gunung, namun mereka menolak karena merasa tidak mampu menanggungnya. Manusia akhirnya menyanggupi amanah tersebut, tetapi sayangnya manusia sering kali berlaku zalim dan tidak bersyukur (Jamil, 2016).

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا لَا تَخُوْنُوا اللّٰهَ وَالرَّسُوْلَ وَتَخُوْنُوْٓا اَمٰنٰتِكُمْ وَاَنْتُمْ تَعْلَمُوْنَ ﴿٢٧﴾

Artinya:“ *Wahai orang-orang yang beriman, janganlah kamu mengkhianati Allah dan Rasul serta janganlah kamu mengkhianati amanat yang dipercayakan kepadamu, sedangkan kamu mengetahui*” (Q.S Al-Anfal: 27) (Kemenag, 2019).

Salah satu ciri utama orang mukmin yang beruntung adalah kesetiaannya terhadap amanah yang diterima, baik dari Allah SWT maupun dari manusia. Ini menunjukkan bahwa ketika mereka diberi tanggung jawab untuk mengelola uang atau barang untuk dibagikan kepada orang lain, mereka menjalankan tugas tersebut dengan jujur dan tanpa kekeliruan. Selain itu, mereka senantiasa menepati janji yang telah dibuat. Jika sebaliknya orang yang tidak menjaga atau tidak menyampaikan disebut munafik. Rasullullah Shallallahu‘Alaihi Wasallam Bersabda “ Tanda-tanda orang yang munafik ada tiga: (1) jika berkata, berdusta, (2) jika berjanji, mengingkari, (3) Apabila dipercaya ,Berkhianat “(HR. Bukhari dan Muslim). Jika salah satu seseorang memiliki salah satu dari tiga tanda tersebut ada pada diri seseorang, maka ia munafik. Munafik yang dimaksud dalam hadist ini yaitu orang yang merugikan dirinya sendiri menurut tafsir Al-Ahzar (Amiruddin, 2021).

Menurut tafsir Al-Misbah, amanah adalah sesuatu yang dipercayakan kepada orang lain untuk dijaga dan dikembalikan ketika waktunya tiba atau saat diminta oleh pemiliknya. Dalam Al-Qur’an ada empat pengertian tentang Amanah:

1. Menunaikan amanah secara sempurna

حَافِظُوْا عَلَى الصَّلَوٰتِ وَالصَّلٰوةِ الْوُسْطٰى وَقُوْمُوْا لِلّٰهِ قٰنِتِيْنَ ﴿٢٣٨﴾

Artiynya *“Peliharalah semua salat (fardu) dan salat Wusṭā.75) Berdirilah karena Allah (dalam salat) dengan khusyuk.75) Menurut pendapat yang masyhur, salat Wusṭā adalah salat Asar*" (Q.S Al-Baqarah: 238) (Kemenag, 2019).

2. Tidak menghianati Amanah dari Allah SWT dan Rosul-Nya

يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا لَا تَخُوْنُوا اللّٰهَ وَالرَّسُوْلَ وَتَخُوْنُوْٓا اَمٰنٰتِكُمْ وَاَنْتُمْ تَعْلَمُوْنَ ﴿٢٧﴾

Artinya "*Wahai orang-orang yang beriman, janganlah kamu mengkhianati Allah dan Rasul serta janganlah kamu mengkhianati amanat yang dipercayakan kepadamu, sedangkan kamu mengetahui* "(Q.S Al-Anfal: 27) (Kemenag, 2019).

3. Berlaku adil

۞ يٰٓاَيُّهَا الَّذِيْنَ اٰمَنُوْا كُوْنُوْا قَوّٰمِيْنَ بِالْقِسْطِ شُهَدَاۤءَ لِلّٰهِ وَلَوْ عَلٰٓى اَنْفُسِكُمْ اَوِ الْوَالِدَيْنِ وَالْاَقْرَبِيْنَۚ اِنْ يَّكُنْ غَنِيًّا اَوْ فَقِيْرًا فَاللّٰهُ اَوْلٰى بِهِمَاۗ فَلَا تَتَّبِعُوا الْهَوٰٓى اَنْ تَعْدِلُوْاۚ وَاِنْ تَلْوٗٓا اَوْ تُعْرِضُوْا فَاِنَّ اللّٰهَ كَانَ بِمَا تَعْمَلُوْنَ خَبِيْرًا ﴿١٣٥﴾

Artinya:"*Wahai orang-orang yang beriman, jadilah kamu penegak keadilan dan saksi karena Allah, walaupun kesaksian itu memberatkan dirimu sendiri, ibu bapakmu, atau kerabatmu. Jika dia (yang diberatkan dalam kesaksian) kaya atau miskin, Allah lebih layak tahu (kemaslahatan) keduanya. Maka, janganlah kamu mengikuti hawa nafsu karena ingin menyimpang (dari kebenaran). Jika kamu memutarbalikkan (kata-kata) atau berpaling (enggan menjadi saksi), sesungguhnya Allah Maha Teliti terhadap segala apa yang kamu kerjakan*" (Q.S An-Nisa: 135) (Kemenag, 2019).

4. Bertanggung jawab

كُلُّ نَفْسٍۢ بِمَا كَسَبَتْ رَهِيْنَةٌ ﴿٣٨﴾

Artinya:"*Setiap orang bertanggung jawab atas apa yang telah ia lakukan*" (Q.S Al-Muddasir: 38) (Kemenag, 2019),

Dalam tafsirnya, Al-Qurtubi mengatakan bahwa amanah adalah tanggung jawab manusia, baik dalam hal keagamaan maupun duniawi, dengan menjaganya dengan kata-kata maupun dengan melakukannya (Ismail & Makmur, 2020).Seperti yang dijelaskan dalam Al-Qur'an terdapat pada Surah An-Nisa' ayat 58. Ayat ini menyatakan pentingnya menyampaikan amanah kepada yang berhak menerimanya serta menetapkan hukum dengan adil di antara manusia.

۞ اِنَّ اللّٰهَ يَأْمُرُكُمْ اَنْ تُؤَدُّوا الْاَمٰنٰتِ اِلٰٓى اَهْلِهَاۙ وَاِذَا حَكَمْتُمْ بَيْنَ النَّاسِ اَنْ تَحْكُمُوْا بِالْعَدْلِۗ اِنَّ اللّٰهَ نِعِمَّا يَعِظُكُمْ بِهٖۗ اِنَّ اللّٰهَ كَانَ سَمِيْعًاۢ بَصِيْرًا ۝٥٨

Artinya:"*Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum di antara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat*."(Kemenag, 2019).

Melaksanakan dan menjaga amanah adalah kewajiban, terutama ketika pihak yang berhak atas amanah tersebut menuntut pemenuhannya. Allah SWT Yang Maha Mengetahui, mengetahui apakah seseorang telah menjalankan amanah dengan benar atau justru mengkhianatinya. Jika seseorang itu jujur dalam mengemban amanah akan mendapatkan kebaikan-kebaikan yang dapat menuntunnya hingga ke surga serta memperoleh ketenangan jiwa. Namun, jika seseorang berdusta atau berkhianat dalam mengemban amanah, maka ia akan merasakan kegelisahan dalam hidupnya. Untuk menunjukkan nilai kepercayaan dan kepatuhan dalam menjaga rahasia ini seperti halnya kriptografi yang melindungi informasi sensitif, di mana menjaga amanah tanpa diketahui orang lain merupakan prinsip penting untuk memastikan keamanan dan keaslian informasi sampai kepada pihak yang berhak.

Ketika Rasulullah SAW menaklukkan kota Makkah, beliau memanggil Utsman bin Thalhah, dan memintanya untuk memberikan kepadaku kunci Ka'bah, lalu ia pergi dan datang kembali dengan membawa kunci Ka'bah dan menjulurkan tangannya kepada Rasulullah SAW dengan membuka telapak tangannya. Ketika itu Abbas (paman Nabi) berdiri dan meminta agar kunci tersebut diberikan kepada-Nya sehingga ia bisa memegang dua tugas sekaligus, yaitu memberi minum jamaah haji dan menjaga kunci Ka'bah. Utsman mendengar hal itu lalu ia kembali

menggenggam kunci tersebut. Kemudian Rasulullah SAW bersabda," berikanlah kepadaku kunci tersebut wahai Utsman", kemudian Utsman berkata," ini kunci Ka'bah wahai Rasulullah SAW, terimalah dengan Amanah Allah SWT", kemudian Rasulullah SAW beranjak dan melaksanakan *thawaf*, kemudian turun Jibril kepadanya menyampaikan pesan Allah SWT untuk mengembalikan kunci tersebut kepada Utsman, lalu Rasulullah SAW memanggil Utsman dan memberikan kunci tersebut dan beliau bersabda dengan firman Allah SWT," sesungguhnya Allah SWT menyuruh kamu yang menyampaikan amanat kepada yang berhak menerimanya". Diriwayatkan oleh Syu'bah tafsir-nya dari Hajjaj dan Ibnu Juraiji bahwasanya ia berkata "ayat ini turun pada Utsman bin Thalhah". Rasulullah SAW mengambil kunci Ka'bah pada hari penaklukkan kota Makkah, Ketika beliau keluar dari Ka'bah ia membaca firman Allah SWT ini, kemudian beliau memanggil Utsman dan memberikan kunci tersebut. Umar bin Al-Khaththab mengatakan bahwa saat Rasulullah SAW membaca ayat tersebut, ia belum pernah mendengarnya sebelumnya yang menandakan bahwa ayat itu turun di dalam Ka'bah (Suyuthi, 2019).

Setelah kunci Ka'bah yang dipegang oleh Rasulullah SAW dan dikembalikan kepada Utsman bin Thalhah karena beliau yang merupakan keturunan dari keluarga yang memiliki hak untuk memegang kunci tersebut. Sebelum Islam, keluarga Utsman bin Thalhah telah dipercaya sebagai penjaga Ka'bah dan kunci tersebut (Syahril & Maqasid, 2015).

Setelah pembukaan Mekkah, Rasulullah SAW ingin menunjukkan bahwa Islam menghormati tradisi dan hak-hak yang ada. Dengan mengembalikan kunci kepada Utsman, beliau juga menunjukkan sikap adil dan memberi penghormatan

kepada keluarga yang telah lama mengemban tugas tersebut. Selain itu, Utsman bin Thalhah kemudian masuk Islam, yang semakin menegaskan kepercayaan Rasulullah kepadanya dan keluarganya. Tindakan ini juga menjadi simbol bahwa Islam adalah agama yang menghargai keadilan dan mengakui hak-hak yang telah ada (Dahlan, 2015).

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian ini termasuk ke dalam penelitian kuantitatif . Dengan pendekatan simulasi komputasi. Metode yang di gunakan melibatkan perancanngan algoritma dan analisis proses matematis serta teknis dalam penerapan algoritma tanda tangan digital menggunakan kode *Reed-Muller* berbasis *MD5.* Metode ini melibatkan analisis proses matematis dan teknis yang relevan serta berkaitan dengan topik penelitian.

3.2 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini melalui simulasi sistem secara terkontrol. Data yang digunakan dalam penelitian ini yaitu dokumen dengan jumlah minimal satu halaman, berformat PDF *(Portable Document Format)* yang diperoleh dari dokumen pribadi seperti surat udangan rapat tahunan.

3.3 Tahapan Penelitian

1. Pembentukan Kunci

   Pembentukan kunci meliputi tahapan berikut:

   a. Menentukan parameter kode linier *Reed-Muller* $r = 1$ dan $m = 3$ dan membuat matriks generator G dari kode *Reed-Muller* berdasarkan kode linier yang ditentukan.

   b. Membuat matriks paritas H dari matriks G.

   c. Membuat matriks S secara acak berupa matriks *non-singular.*

d. Memilih sembarang kunci matriks Q secara acak berupa matriks permutasi.

e. Menghitung matrik HSQ dengan ukuran $k \times n, HSQ = H \times S \times Q$.

f. Mendapatkan kunci publik H' dan kunci privat H, S, Q.

2. Penandatanganan Pesan

Menandatangani pesan meliputi tahapan berikut

a. *Input* berupa *hexadesimal* m dari fungsi *hash MD5*.

b. Mengonversi pesan m ke dalam bentuk biner berdasarkan tabel ASCII Dan embagi biner menjadi ke dalam blok-blok s_i dengan panjang $4 - bit$.

c. Menghitung vektor $s' = S^{-1}s$

d. Menghitung tanda tangan yang dihitung e'^T dengan $He'^T = s$

e. Menghitung tanda tangan yang dengan pesan yang dikirim dengan $e^T = Q^{-1}e'^T$.

3. Verifikasi Tanda Tangan Digital

Tahapan verifikasi pesan dengan membandingkan $H'e^T = s$

BAB IV

PEMBAHASAN

4.1 Proses Pembentukan Kunci

Proses pembentukan kunci pada penelitian ini merupakan tahap fundamental dalam implementasi tanda tangan digital berbasis kode *Reed-Muller* dan fungsi *hash MD5*. Sistem ini menggunakan pasangan kunci asimetris yang terdiri dari kunci privat (H, S, Q) dan kunci publik (H'), di mana setiap komponen memiliki peran krusial dalam menjamin keamanan dan keandalan skema tanda tangan yang dibangun.

Matriks generator G dari kode *Reed-Muller*, RM(r, m) dibentuk melalui evaluasi monomial Boolean dengan parameter r dan m berupa bilangan asli yang mencakup $r \leq m$. Adapun parameter kode *Reed-Muller* yang digunakan untuk mengetahui parameter $G(r, m)$ yaitu n dan k dimana panjang kode $n = 2^m$ dan dimensi $k = \sum_{i=0}^{r} \binom{m}{i}$ sebagai orde kode matriks G . Matriks ini kemudian dikombinasikan dengan matriks pengacak S berukuran $k \times k$ yang bersifat non-singular dan matriks permutasi Q berukuran $n \times n$ untuk menghasilkan struktur kode yang tersembunyi. Transformasi kunci publik $H' = HSQ$ menghasilkan matriks yang sifatnya mengoreksi kesalahan kode *Reed-Muller* namun menyamarkan struktur aslinya. Dalam simulasi praktis dengan $RM(1,3)$, penggunaan matriks S merupakan matriks semabarang yang bersifat *non-singular* yaitu matriks dengan determinan tidak sama dengan nol dan Q merupakan matriks permutasi diperoleh memilih sembarang acak menghasilkan kunci publik yang resistensi terhadap upaya *reverse engineering*.

Pertama menentukan parameter kode *Reed-Muller* $G(r,m)$ yaitu $r=1$ dan $m=3$. Dari parameter kode *Reed-Muller* dapat digunakan untuk mengetahui parameter n dan k sebagai berikut.

$$n = 2^m = 2^3 = 8$$

$$k = \sum_{i=0}^{r} \binom{m}{i} = \sum_{i=0}^{1} \binom{3}{i} = \binom{3}{0} + \binom{3}{1} = \frac{3!}{0!\,(3-0)!} + \frac{3!}{1!\,(3-1)!} = 1 + 3 = 4$$

Setelah parameter di tentukan, kemudian tulis semua kombinasi variabel (X_1, X_2, X_3) yang didapatkan dari $m=3$ yaitu banyaknya *input* bit yang digunakan untuk membangun ruang vektor biner berdimensi 3 dengan total 8 vektor *input* kombinasi yang di dapatkan dari $2^3 = 8$. Satu urutan yang umum digunakan pada G ditunjukan pada tabel 4.1

Tabel 4.1 Kombinasi Variabel G $(\mathbf{1}, \mathbf{3})$

Kolom (Indeks)	X_1	X_2	X_3
1	0	0	0
2	1	0	0
3	0	1	0
4	1	1	0
5	0	0	1
6	1	0	1
7	0	1	1
8	1	1	1

Karena basis ruang fungsi polinomial derajat ≤ 1 maka basis $RM(1,3)$ adalah $\{1, X_1, X_2. X_3\}$ dengan urutan kolom yang ditulis sebagai berikut:

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix}$$

Selanjutnya akan di bentuk matriks paritas H dengan menggunkan kode *Reed-Muller* pada matriks generator G menggunakan metode *Gauss–Jordan* dilakukan di $GF(2)$, artinya semua operasi $+$ adalah XOR atau modulo 2 pendekatannya adalah menempatkan G ke bentuk sistematis $[\, I_k \mid P \,]$ (di mana $k = 4$ dan $n = 8$), lalu membentuk matriks paritas dari blok P.

Pertama Kita permutasikan kolom G ke urutan 1-based $[4, 6, 7, 8, 1, 2, 3, 5]$ (0-based: $[3, 5, 6, 7, 0, 1, 2, 4]$). Hasil G setelah permutasi kolom (sebut G') adalah:

$$G' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Sekarang kolom-kolom pertama 4 kolom pada G' dipilih supaya bisa dibentuk menjadi I_4 dengan operasi baris.

Langkah 1, operasi $R_2 \leftarrow R_2 + R_1$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Langkah 2, operasi $R_3 \leftarrow R_3 + R_1$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Langkah 3, operasi tukar baris $R_2 \leftrightarrow R_4$ agar pivot pada kolom ke-2 tersedia.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Langkah 4, operasi untuk mengosongkan entri kolom 2 pada baris 1, $R_1 \leftarrow R_1 + R_2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Langkah 5, operasi mengosongkan entri di kolom 2 pada baris 3, $R_3 \leftarrow R_3 + R_2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Langkah 6, operasi membersihkan kolom 3 pada baris 2, sehingga hanya baris 3 yang punya 1 di kolom 3, $R_2 \leftarrow R_2 + R_3$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Langkah 7, operasi membersihkan kolom 3 pada baris 4, $R_4 \leftarrow R_4 + R_3$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Langkah 8, Operasi: $R_3 \leftarrow R_3 + R_4$ untuk membersihkan kolom 4 di baris 3, sehingga pivot kolom 4 menjadi hanya pada baris 4 hanya baris 4.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Hasil akhir matriks sistematis (setelah semua eliminasi) adalah:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = [I_4|P]$$

Di sini blok paritas P (kolom 5–8 dari R) Adalah

$$P = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Selanjutnya Bentuk matriks paritas dari P Ketentuan untuk generator sistematis $G = [I_k \mid P]$ adalah matriks paritas (dalam urutan kolom yang sama) dapat ditulis

$$H' = [P^T|I_{n-k}]$$

Jadi untuk P diatas diperoleh

$$H' = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

di mana setiap baris H'menjadikan $R \cdot H'^T = 0 (\text{mod } 2)$.

Selanjutnya karena kita sebelumnya memutasikan kolom G menjadi urutan permutasi $[4, 6, 7, 8, 1, 2, 3, 5]$, matriks H' yang kita punya saat ini juga berada pada urutan kolom yang sama. Untuk memperoleh matriks paritas H yang kolom-kolomnya sesuai urutan kolom asli G, kita harus menerapkan invers permutasi pada kolom-kolom H'. Setelah menerapkan invers permutasi (mengembalikan kolom ke urutan semula), diperoleh

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Kemudian membangkitkan matriks S secara acak dengan ukuran $k \times k$ dan matriks Q dengan ukuran $n \times n$. Dimisalkan sebagai berikut:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Untuk mencari invers matriks dari S maka akan menggunakan metode eliminasi Gaus Jordan. Akan membentuk *augmented* matriks $[S|I]$, lalu menggunakan eliminasi Gaus Jordan mod 2 untuk mengubah sisi kiri menjadi identitas dan sisi kanan menjadi inversnya.

$$(S|I) = \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Langkah ke-1 menggunakan $R_3 \leftrightarrow R_1$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Langkah ke-2 menggunakan $R_4 \leftrightarrow R_1$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \end{array}\right)$$

Langkah ke-3 menggunakan $R_3 \leftrightarrow (-1 \times R_2)$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \end{array}\right)$$

Langkah ke-4 menggunakan membagi R_4 dengan -1maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{array}\right)$$

Langkah ke-5 menggunakan $R_1 - R_4$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{array}\right)$$

Langkah ke-6 menggunakan $R_1 - R_3$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{array}\right)$$

Langkah ke-7 menggunakan $R_2 - R_3$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 1 & 0 & 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{array}\right)$$

Langkah ke-8 menggunakan $R_1 - R_2$ maka diperoleh:

$$= \left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{array}\right)$$

Sehingga S^{-1} yang didapatkan metode eliminasi Gaus Jordan diperoleh:

$$S^{-1} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Untuk mencari invers matriks dari Q maka akan menggunakan metode eliminasi Gaus Jordan. Akan membentuk *augmented* matriks $(Q|I)$ sebagai berikut:

$$(Q|I) = \left(\begin{array}{cccccccc|cccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Langkah ke-1 menggunakan $R_2 \leftrightarrow R_1$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Langkah ke-2 menggunakan $R_7 \leftrightarrow R_2$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

Langkah ke-3 menggunakan $R_8 \leftrightarrow R_3$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Langkah ke-4 menggunakan $R_7 \leftrightarrow R_4$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Langkah ke-5 menggunakan $R_7 \leftrightarrow R_5$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Langkah ke-6 menggunakan $R_7 \leftrightarrow R_6$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Langkah ke-7 menggunakan $R_7 \leftrightarrow R_8$ maka diperoleh:

$$\left(\begin{array}{cccccccc|cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right)$$

Sehingga Q^{-1} yang didapatkan menggunakan metode Gaus Jordan metode eliminasi Gaus Jordan diperoleh:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Selanjutnya pembentukan kunci publik H' dilakukan dengan mengalikan matriks kunci privat H, S, dan Q. Perkalian tersebut menghasilkan suatu matriks

sebagai kunci publik H' yang dapat diketahui oleh semua pihak. Adapun matriks H' berukuran 4×8.

$$H' = HSQ = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

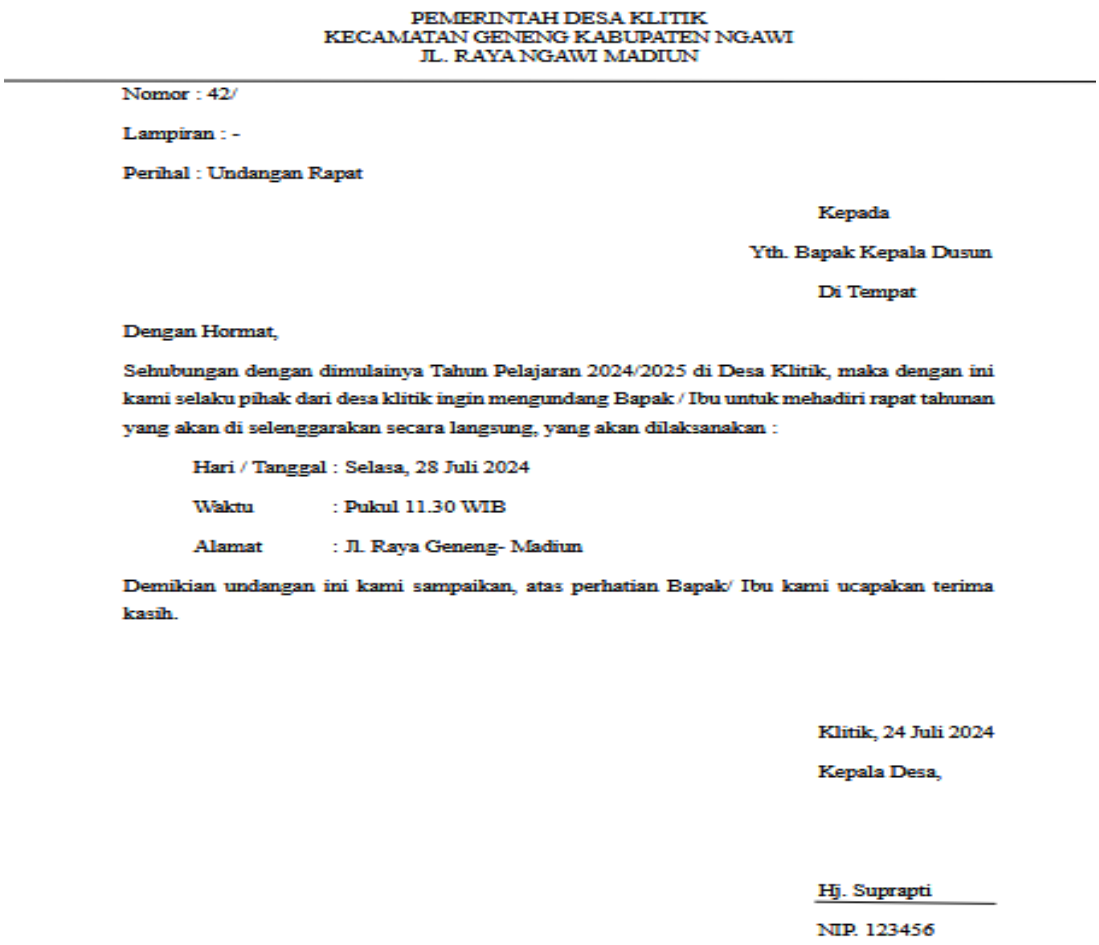
4.2 Proses Penandatanganan Pesan

Proses *hash MD5* terhadap file PDF dilakukan dengan cara membaca seluruh isi file dalam bentuk biner dan kemudian memprosesnya menggunakan algoritma *MD5*. *MD5* adalah fungsi *hash* yang mengubah data berukuran berapa pun menjadi keluaran tetap sepanjang 128 bit. Ketika sebuah file PDF dijadikan *input,* sistem tidak melihat isi dokumen seperti teks melainkan membacanya sebagai rangkaian *byte*. Setiap *byte* dari file tersebut diolah secara berurutan oleh algoritma *MD5* untuk menghasilkan nilai *hash*. Proses ini bersifat deterministik, artinya file yang sama akan selalu menghasilkan nilai *hash* yang sama, sementara perubahan sekecil apa pun dalam isi file akan menghasilkan *hash* yang sangat berbeda.

Dalam implementasinya, proses *hashing* dilakukan dengan membuka file PDF dalam mode biner, membaca seluruh kontennya, lalu menghitung nilai *hash* menggunakan fungsi *MD5*. Sebagai contoh, dalam bahasa pemrograman Python untuk menghasilkan nilai *hash*.

Penggunaan *hash* pada file PDF bertujuan utama untuk menjaga keaslian dokumen. Jika ada perubahan sedikit saja pada file, maka *hash* akan berubah dan tanda tangan tidak lagi valid saat diverifikasi. Oleh karena itu, proses *hashing MD5* berperan penting dalam deteksi integritas dan autentikasi dokumen digital.

Pada bab ini akan dilakukan pembuatan tanda tangan digital menggunakan fungsi *hash MD5* terhadap dokumen berformat pdf. Dengan bantuan bahasa pemrograman *python* dengan meng*hash* sebuah dokumen pada Gambar 4.1.



PEMERINTAH DESA KLITIK
KECAMATAN GENENG KABUPATEN NGAWI
JL. RAYA NGAWI MADIUN

Nomor : 42/

Lampiran : -

Perihal : Undangan Rapat

Kepada

Yth. Bapak Kepala Dusun

Di Tempat

Dengan Hormat,

Sehubungan dengan dimulainya Tahun Pelajaran 2024/2025 di Desa Klitik, maka dengan ini kami selaku pihak dari desa klitik ingin mengundang Bapak / Ibu untuk mehadiri rapat tahunan yang akan di selenggarakan secara langsung, yang akan dilaksanakan :

Hari / Tanggal : Selasa, 28 Juli 2024

Waktu : Pukul 11.30 WIB

Alamat : Jl. Raya Geneng- Madiun

Demikian undangan ini kami sampaikan, atas perhatian Bapak/ Ibu kami ucapakan terima kasih.

Klitik, 24 Juli 2024

Kepala Desa,

Hj. Suprapti

NIP. 123456

Gambar 4.1 Contoh Dokumen *PDF* yang Diproses *MD5*

Dari dokumen tersebut engujian pertama didapatkan *message digest* (m) dari fungsi *hash MD5* yang diperoleh yaitu:

"*f680a988948915156a2dfea161eed5bd*"

Selanjutnya *message digest* ini diubah ke dalam bentuk biner menggunakan, sehingga diperoleh

1111 0110 1000 0000 1010 1001 1000 1000 1001 0100 1000 1001 0001 0101 0001 0101 0110 1010 0010 1101 1111 1110 1010 0001 0110 0001 1110 1110 1101 0101 1011 1101

Kemudian membagi menjadi beberapa blok s_i dengan panjang 4-bit sesuai parameter k sehingga sebanyak 32 blok sebagai berikut:

$$s_1 = 1111 \quad s_9 = 1001 \quad s_{17} = 0110 \quad s_{25} = 0110$$
$$s_2 = 0110 \quad s_{10} = 0100 \quad s_{18} = 1010 \quad s_{26} = 0001$$
$$s_3 = 1000 \quad s_{11} = 1000 \quad s_{19} = 0010 \quad s_{27} = 1110$$
$$s_4 = 0000 \quad s_{12} = 1001 \quad s_{20} = 1101 \quad s_{28} = 1110$$
$$s_5 = 1010 \quad s_{13} = 0001 \quad s_{21} = 1111 \quad s_{29} = 1101$$
$$s_6 = 1001 \quad s_{14} = 0101 \quad s_{22} = 1110 \quad s_{30} = 0101$$
$$s_7 = 1000 \quad s_{15} = 0001 \quad s_{23} = 1010 \quad s_{31} = 1011$$
$$s_8 = 1000 \quad s_{16} = 0101 \quad s_{24} = 0001 \quad s_{32} = 1101$$

Setelah pembagian masing-masing blok yang berukuran 4-bit langkah selanjutnya mengalikan s_i dengan S^{-1} dengann bertujuan menyamarkan struktur kode pada verifikasi tanda tangan digital berbasis *Reed-Muller* sebagai berikut:

$$s_1' = S^{-1} \times s_1$$
$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$s_2' = S^{-1} \times s_2$$
$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$s_3' = S^{-1} \times s_3$$
$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$s_4' = S^{-1} \times s_4$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$s_5' = S^{-1} \times s_5$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s_6' = S^{-1} \times s_6$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$s_7' = S^{-1} \times s_7$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$s_8' = S^{-1} \times s_8$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$s_9' = S^{-1} \times s_9$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$s_{10}' = S^{-1} \times s_{10}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$s'_{11} = S^{-1} \times s_{11}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$s'_{12} = S^{-1} \times s_{12}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$s'_{13} = S^{-1} \times s_{13}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s'_{14} = S^{-1} \times s_{14}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s'_{15} = S^{-1} \times s_{15}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s'_{16} = S^{-1} \times s_{16}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s'_{17} = S^{-1} \times s_{17}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$s'_{18} = S^{-1} \times s_{18}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s'_{19} = S^{-1} \times s_{19}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$s'_{20} = S^{-1} \times s_{20}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$s'_{21} = S^{-1} \times s_{21}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$s'_{22} = S^{-1} \times s_{22}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$s'_{23} = S^{-1} \times s_{23}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s'_{24} = S^{-1} \times s_{24}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s'_{25} = S^{-1} \times s_{25}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$s'_{26} = S^{-1} \times s_{26}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$s'_{27} = S^{-1} \times s_{27}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s'_{28} = S^{-1} \times s_{28}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s'_{29} = S^{-1} \times s_{29}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$s'_{30} = S^{-1} \times s_{30}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$s'_{31} = S^{-1} \times s_{31}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$s'_{32} = S^{-1} \times s_{32}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Dipatkan nilai untuk semua s'_i sebagai berikut:

$$s'_i = \begin{matrix} 0010\ 1100\ 0111\ 0000\ 0001\ 1110\ 0111\ 0111\ 1110\ 1010 \\ 0111\ 1110\ 1001\ 0011\ 1001\ 0011\ 1100\ 0001\ 0110\ 0100 \\ 0010\ 1011\ 0001\ 1001\ 1100\ 1001\ 1011\ 1011\ 0100\ 0011 \\ 1000\ 0100 \end{matrix}$$

Selanjutnya, dalam proses ini digunakan metode *Brute Force* untuk mencari seluruh kemungkinan vektor *error* e_i' dengan yang berfungsi sebagai tanda tangan digital, dengan panjang 8 bit, yang memenuhi persamaan:

$$H \cdot e_i^{iT} = s_i'$$

di mana H adalah matriks pemeriksa paritas, e_i' merupakan kandidat vektor *error* , dan s_i' adalah sindrom yang dihasilkan dari hasil hash pesan digital dapat dilakukan sebagai berikut:

$s'_1 = 0010$,*error* pada bit ke-3 ($e'_1 = [0,0,1,0,0,0,0,0]$)

$$H.e_1'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot1+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = s_1'$$

$s'_2 = 1100$, *error* pada bit 1 dan 2 ($e'_2 = [1,1,0,0,0,0,0,0]$)

$$H.e_2'^T = \begin{bmatrix} 1\cdot1+0\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot1+1\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot1+0\cdot1+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot1+0\cdot1+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = s_2'$$

$s_3' = 0111$, *error* pada bit ke-8 ($e_3' = [0,0,0,0,0,0,0,1]$)

$$H.e_3'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot1 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = s_3'$$

$s_4' = 0000$, *error* ($e_4' = [0,0,0,0,0,0,0,0]$)

$$H.e_4'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = s_4'$$

$s_5' = 0001$, *error* pada bit ke 5 ($e_5' = [0,0,0,0,1,0,0,0]$)

$$H.e_5'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot1+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = s_5'$$

$s_6' = 1110$, *error* pada bit ke-4 ($e_6' = [0,0,0,1,0,0,0,0]$)

$$H.e_6'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot1+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot1+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot1+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = s_6'$$

$s_7' = 0111$, *error* pada bit ke-8 ($e_7' = [0,0,0,0,0,0,0,1]$)

$$H.e_7'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot1 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = s_7'$$

$s_8' = 0111$, *error* pada bit ke-8 ($e_8' = [0,0,0,0,0,0,0,1]$)

$$H.e_8'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot1 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = s_8'$$

$s_9' = 1110$, *error* pada bit ke-4($e_9' = [0,0,0,1,0,0,0,0]$)

$$H.e_9'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot1+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot1+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot1+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = {s_9}'$$

$s_{10}' = 1010$, *error* pada bit ke 1 dan 3 ($e_{10}' = [1,0,1,0,0,0,0,0]$)

$$H.e_{10}'^T = \begin{bmatrix} 1\cdot1+0\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot1+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot1+0\cdot0+1\cdot1+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot1+0\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = s_{10}$$

$s_{11}' = 0111$, *error* pada bit ke-8 ($e_{11}' = [0,0,0,0,0,0,0,1]$)

$$H.e_{11}'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot1 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot1 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = {s_{11}}'$$

$s_{12}' = 1110$, *error* pada bit ke-4 ($e_{12}' = [0,0,0,1,0,0,0,0]$)

$$H.e_{12}'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot1+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot1+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot1+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} = {s_{12}}'$$

$s_{13}' = 1001$, *error* pada bit ke 1 dan 5 ($e_{13}' = [1,0,0,0,1,0,0,0]$)

$$H.e_{13}'^T = \begin{bmatrix} 1\cdot1+0\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot1+1\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot1+0\cdot0+1\cdot0+1\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot1+0\cdot0+0\cdot0+0\cdot0+1\cdot1+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {s_{13}}'$$

$s_{14}' = 0011$, *error* pada bit ke-3 dan 5 ($e_{14}' = [0,0,1,0,1,0,0,0]$)

$$H.e_{14}'^T = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot1+1\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot1+1\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot1+0\cdot0+1\cdot1+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = {s_{14}}'$$

$s_{15}' = 1001$, *error* pada bit ke 1 dan 5 ($e_{15}' = [1,0,0,0,1,0,0,0]$)

$$H.e_{15}'^T = \begin{bmatrix} 1\cdot1+0\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot1+1\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot1+0\cdot0+1\cdot0+1\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot1+0\cdot0+0\cdot0+0\cdot0+1\cdot1+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {s_{15}}'$$

$s'_{16} = 0011$, *error* pada bit ke 3 dan 5 ($e'_{16} = [0,0,1,0,1,0,0,0]$)

$$H.e'^{T}_{16} = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot1+1\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot1+1\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot1+0\cdot0+1\cdot1+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = s_{16}'$$

$s'_{17} = 1100$, *error* pada bit 1 dan 2 ($e'_{17} = [1,1,0,0,0,0,0,1]$)

$$H.e'^{T}_{17} = \begin{bmatrix} 1\cdot1+0\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot1+1\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot1+0\cdot1+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot1+0\cdot1+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = s_{17}'$$

$s'_{18} = 0001$, *error* pada bit ke-5 ($e'_{18} = [0,0,0,0,1,0,0,0]$)

$$H.e'^{T}_{18} = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot1+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = s_{18}'$$

$s'_{19} = 0110$, *error* pada bit ke 2 dan 3 ($e'_{19} = [0,1,1,0,0,0,0,0]$)

$$H.e'^{T}_{19} = \begin{bmatrix} 1\cdot0+0\cdot1+0\cdot1+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot1+0\cdot1+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot1+1\cdot1+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot1+0\cdot1+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = s_{19}'$$

$s'_{20} = 0100$, *error* pada bit ke-2 ($e'_{20} = [0,1,0,0,0,0,0,0]$)

$$H.e'^{T}_{20} = \begin{bmatrix} 1\cdot0+0\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot1+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot1+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot1+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = s_{20}'$$

$s'_{21} = 0010$, *error* pada bit ke-3 ($e'_{21} = [0,0,1,0,0,0,0,0]$)

$$H.e'^{T}_{21} = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot1+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot1+1\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot1+0\cdot0+1\cdot0+1\cdot0+1\cdot0+1\cdot0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = s_{21}'$$

$s'_{22} = 1011$, *error* pada bit ke-7 ($e'_{22} = [0,0,0,0,0,0,1,0]$)

$$H.e'^{T}_{22} = \begin{bmatrix} 1\cdot0+0\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+1\cdot1+0\cdot0 \\ 0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot0+1\cdot0+0\cdot1+1\cdot0 \\ 0\cdot0+0\cdot0+1\cdot0+1\cdot0+0\cdot0+0\cdot0+1\cdot1+1\cdot0 \\ 0\cdot0+0\cdot0+0\cdot0+0\cdot0+1\cdot0+1\cdot0+1\cdot1+1\cdot0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = s_{22}'$$

$s'_{23} = 0001$, *error* pada bit ke-5 ($e'_{23} = [0,0,1,0,0,0,0,0]$)

$$H.e'^{T}_{23} = \begin{bmatrix} 1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 0+0\cdot 0+0\cdot 0+1\cdot 1+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = s_{23}'$$

$s'_{24} = 1001$, *error* pada bit ke 1 dan 5 ($e'_{24} = [1,0,0,0,1,0,0,0]$)

$$H.e'^{T}_{24} \begin{bmatrix} 1\cdot 1+0\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 0+0\cdot 0+0\cdot 0+1\cdot 1+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = s_{24}'$$

$s'_{25} = 1100$, *error* pada bit 1 dan 2 ($e'_{25} = [1,1,0,0,0,0,0,0]$)

$$H.e'^{T}_{25} = \begin{bmatrix} 1\cdot 1+0\cdot 1+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 1+1\cdot 1+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 1+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = s_{25}'$$

$s'_{26} = 1001$, *error* pada bit ke 1 dan 5 ($e'_{26} = [1,0,0,0,1,0,0,0]$)

$$H.e'^{T}_{26} = \begin{bmatrix} 1\cdot 1+0\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 0+0\cdot 0+0\cdot 0+1\cdot 1+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = s_{26}'$$

$s'_{27} = 1011$, *error* pada bit ke-7($e'_{27} = [0,0,0,0,0,0,1,0]$)

$$H.e'^{T}_{27} = \begin{bmatrix} 1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+1\cdot 1+0\cdot 0 \\ 0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0 \\ 0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 1+1\cdot 0 \\ 0\cdot 0+0\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+1\cdot 1+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = s_{27}'$$

$s'_{28} = 1011$, *error* pada bit bit ke-7($e'_{28} = [0,0,0,0,0,0,1,0]$)

$$H.e'^{T}_{28} = \begin{bmatrix} 1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+1\cdot 1+0\cdot 0 \\ 0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0 \\ 0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 1+1\cdot 0 \\ 0\cdot 0+0\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+1\cdot 1+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} = s_{28}'$$

$s'_{29} = 0100$, *error* pada bit ke-2 ($e'_{29} = [0,1,0,0,0,0,0,0]$)

$$H.e'^{T}_{29} = \begin{bmatrix} 1\cdot 0+0\cdot 1+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 0+1\cdot 1+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 1+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = s_{29}'$$

$s'_{30} = 0011$, *error* pada bit ke 3 dan 5 ($e'_{30} = [0,0,1,0,1,0,0,0]$)

$$H.e'^{T}_{30} = \begin{bmatrix} 1\cdot 0+0\cdot 0+0\cdot 1+1\cdot 0+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 0+1\cdot 0+0\cdot 1+1\cdot 0+0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 0+1\cdot 1+1\cdot 0+0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 0+0\cdot 1+0\cdot 0+1\cdot 1+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = s_{30}'$$

$s'_{31} = 1000$, *error* pada bit ke-1 ($e'_{31} = [1,0,0,0,0,0,0,0]$)

$$H.e'^{T}_{31} = \begin{bmatrix} 1\cdot 1+0\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 1+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 1+0\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = s_{31}'$$

$s'_{32} = 0100$, *error* pada bit ke-2 ($e'_{32} = [0,1,0,0,0,0,0,0]$)

$$H.e'^{T}_{32} = \begin{bmatrix} 1\cdot 0+0\cdot 1+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+0\cdot 0 \\ 0\cdot 0+1\cdot 1+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0+0\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 1+1\cdot 0+1\cdot 0+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0 \\ 0\cdot 0+0\cdot 1+0\cdot 0+0\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0+1\cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = s_{32}'$$

Dipatkan nilai e'^{T}_{i} dengan sebagai berikut:

$$e'^{T}_{i} = \begin{matrix} [00100000\ 1100000\ 00000001\ 00000000\ 00001000 \\ 00010000\ 00000001\ 00000001\ 00010000\ 10100000 \\ 00000001\ 00010000\ 10001000\ 00101000\ 10001000 \\ 00101000\ 11000000\ 00001000\ 01100000\ 01000000 \\ 00100000\ 00000010\ 00001000\ 10001000\ 11000000 \\ 10001000\ 00000010\ 00000010\ 01000000\ 00101000 \\ 10000000\ 01000000] \end{matrix}$$

Selanjutnya untuk mendapatkan tanda tangan digital yang dikirim bersama dengan pesan yang akan diterima dengan menghitung $e^T = Q^{-1}e'^{T}_{i}$, dan menghasilkan tanda tangan (m, e, i).

$$e_1^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$e_2^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_3^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_4^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_5^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_6^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_7^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_8^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_9^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{10}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$e_{11}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{12}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{13}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{14}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$e_{15}^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{16}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$e_{17}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{18}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{19}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$e_{20}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{21}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$e_{22}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{23}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{24}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{25}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{26}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{27}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{28}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{29}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{30}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$e_{31}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{32}^{T} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Didapatkan hasil deskripsi e_i^T dimana sebagai berikut:

$$e_i^T = \begin{matrix} [00000010\ 10010000\ 00100000\ 00000000\ 00000100\ 00001000 \\ 00100000\ 00100000\ 00001000\ 00010010\ 00100000\ 00001000 \\ 00010100\ 00000110\ 00010100\ 00000110\ 10010000\ 00000100 \\ 10000010\ 10000000\ 00000010\ 01000000\ 00010000\ 00010100 \\ 10010000\ 00010100\ 01000000\ 01000000\ 10000000\ 00000110 \\ 00010000\ 10000000] \end{matrix}$$

4.3 Proses Verifikasi

Untuk mengembalikan pesan ke aslinya, ada beberapa langkah yang diperlukan. Langkah pertama dalam proses verifikasi adalah menentukan H' yang didapatkan dari $H' = SHQ$. Adapun proses memverifikasi apakah pesan yang sudah ditanda tangani oleh penerima sama dengan pesan asli itu sama maka harus sama dengan *message digest* H'. $e_1^T = s$

$$H'e_1^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$H'e_2^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_3^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H'e_4^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H'e_5^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_6^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_7^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H'e_8^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H'e_9^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{10}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$H'e_{11}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$H'e_{12}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{13}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{14}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{15}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{16}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{17}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{18}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{19}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{20}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{21}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$H'e_{22}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{23}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{24}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{25}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{26}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{27}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{28}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$H'e_{29}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{30}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$H'e_{31}^{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$H'e_{32}^{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Didapatkan hasil verifikasi adalah 1111 0110 1000 0000 1010 1001 1000 1000 1001 0100 1000 1001 0001 0101 0001 0101 0110 1010 0010 1101 1111 1110 1010 0001 0110 0001 1110 1110 1101 0101 1011 1101. Menunjukan bahwa tanda tangan digital yang di hasilkan sama maka dokumen valid atau dokumen tersebut tidak mengalami perubahan.

4.5 Perbandingan Hasil Implementasi Dengan Menggunakan Dokumen Berbeda

Untuk menguji keandalan sistem tanda tangan digital yang dikembangkan, dilakukan eksperimen dengan dua dokumen berbeda, yaitu: Dokumen A (Asli) pada Dokumen B (Telah Dimodifikasi). Kedua dokumen diproses dengan skema tanda tangan digital yang meliputi: perhitungan *hash MD5*, pembangkit kunci menggunakan kode *Reed-Muller*, dan proses penaanda tangan digital melalui metode *Brute Force* untuk mencari vektor *error* e yang sesuai.

Dari hasil tersebut dapat dilihat bahwa meskipun perubahan pada dokumen tergolong kecil dan tampak tidak signifikan secara visual, sistem berhasil

mendeteksi adanya perbedaan melalui nilai *hash* dan sindrom yang dihasilkan. Hal ini menunjukkan bahwa fungsi *hash MD5* sensitif terhadap perubahan data (*avalanche effect*), dan sistem tanda tangan digital berbasis kode *Reed-Muller* mampu mendeteksi integritas data secara presisi.

Sistem verifikasi juga membuktikan bahwa tanda tangan digital yang dihasilkan untuk dokumen asli tidak dapat digunakan pada dokumen yang telah dimodifikasi, sehingga integritas pesan tetap terjaga. Proses ini sekaligus menunjukkan bahwa tanda tangan digital dalam sistem ini bersifat unik terhadap isi dokumen, meskipun kunci privat yang digunakan tetap sama

Perubahan tersebut direpresentasikan sebagai vektor kesalahan e, yang diperoleh dengan melakukan operasi *brute-force* bit-per-bit antara representasi biner berkas asli dan berkas yang dimodifikasi. Dalam konteks sistem kriptografi misalnya tanda tangan digital berbasis kode vektor kesalahan ini menjadi bukti formal adanya perubahan isi berkas. Integritas berkas dinyatakan tidak terjaga apabila nilai *hash* berkas hasil modifikasi berbeda dari nilai *hash* berkas asli. Oleh karena itu, verifikasi terhadap dokumen *PDF* dilakukan dengan membandingkan *hash*-nya jika *hash* cocok, dokumen dinyatakan valid (identik dengan dokumen asli) jika *hash* tidak cocok, dokumen dinyatakan tidak valid (telah mengalami perubahan) seperti pada Tabel 4.2.

Tabel 4.2 Perbandingan TTD Dihitung dan TTD Dikirim

Dokumen dan *Hash*	TTD dihitung	TTD dikirim	validasi
Asli.pdf (*d148c99b28e3a32500354d6ce103f07b*) Sabtu, 15 Juni 2025	[01100000,00000100,00001000,10100000 00000010,00000001,00000001,00100000 01001000,10100000,00010000,10000000 00011000,10000000,01001000,11000000 00000000,00000000,10000000,11000000 00001000,01100000,01000000,00000010 00010000,00000100,00000000,10000000 00101000,00000000,10001000,00100000]	[01100000,00000001,10000000,01000100 00010000,00000010,00000010,01000000 10100000,01000100,00001000,00000100 10001000,00000100,10100000,00100100 00000000,00000000,00000100,00100100 10000000,01100000,00100000,00010000 00001000,00000001,00000000,00000100 11000000,00000000,10000100,01000000]	Valid
Ubah.pdf (*a007cb2c11b4c989a3b46f57103b5ecb*) Sabtu, 15 Juni 2052	[00011000,00000000,00000000,10001000 00000010,00100000,01001000,00000010 00000100, 00000100,00100000,00001000 00000010,00000001,10100000,00000001	[10001000,00000000,00000000,10000100 00010000,01000000,10100000,00010000 00000001, 00000001,01000000,10000000 00010000,00000010,01000100,00000010	Tidak Valid

Dokumen dan *Hash*	TTD dihitung	TTD dikirim	validasi
	00011000,10000000,00100000,00001000 01000000,00101000,11000000,10001000 00000100,00000000,10000000,00100000 11000000,00010000,00000010,00100000]	10001000,00000100,01000000,10000000 00100000,11000000,00100100,10000100 00000001,00000000,00000010,01000000 00100100,00001000,00010000,01000000]	
Asli 2.pdf (*d892d97e56ccf89d504914ec6abdf f27*) Kamis 19 Juni 2026 Sabtu 21 Juni 2026	[01000000,00010000,00011000,01100000 001000000,00011000,11000000,00101000 10100000,00000100,01001000, 01001000 00100000,00010000,00011000,01000000 10100000,00000000,00000010,00011000 00001000,00000010,00101000,01001000 00000100,10000000,10001000,01000000 00100000,00100000,01100000,11000000]	[10000000,00000010,00010010,10001000 10000000,00010010,11000000,00011000 01001000,00000001,10010000, 10010000 00001000,00000010,00010010,10000000 01001000,00000000,00100000,00010010 00010000,00100000,00011000,10010000 00000001,01000000,10010000,10000000 00001000,00001000,10001000,11000000]	Valid

Dokumen dan *Hash*	TTD dihitung	TTD dikirim	validasi
Ubah 2.pdf (*bd8c60e01beea51cfbf93945d317b98d*) Kamis 19 Juni 2025 Sabtu 21 Juni 2025	[10001000,01000000,00010000,01001000 00000100,00000000,00101000,00000000 00001000, 10001000,00101000,00101000 10000000,10100000,00001000,01001000 00100000,10001000,00100000,00011000 00000001,00011000,00000010,10100000 01000000,00000001,00001000,11000000 10001000,00011000,00010000,01000000]	[01010000,10000000,00000010,10010000 00000001,00000000,00011000,00000000 00010000, 01010000,00011000,00011000 01000000,01001000,00010000,10010000 00001000,01010000,00001000,00010010 00000100,00010010,00100000,01001000 10000000,00000100,00010000,11000000 01010000,00010010,00000010,10000000]	Tidak Valid
Ubah 3.pdf (*a007cb2c11b4c989a3b46f57103b5ecb*) Sabtu 19 Juni 2026 Kamis 21 Juni 2026	[10001000,00000100,01001000,00000001 11000000,00011000,00000010,10100000 00011000, 00000010,01000000,00011000 01001000,00010000,00011000,00001000 00101000,00101000,00000100,01001000	10001000,00000000,00000000,10000100 00010000,01000000,10100000,00010000 00000001, 00000001,01000000,10000000 00010000,00000010,01000100,00000010 10001000,00000100,01000000,10000000	Tidak Valid

Dokumen dan *Hash*	TTD dihitung	TTD dikirim	validasi
	01000000,01000000,00000001,00001000 00100000,00000000,10001000,11000000 00011000,00011000,00101000,00100000]	00100000,11000000,00100100,10000100 00000001,00000000,00000010,01000000 00100100,00001000,00010000,01000000]	

4.6 Analisis Hasil

Hasil penelitian menunjukkan bahwa implementasi tanda tangan digital dengan memanfaatkan kombinasi fungsi *hash MD5* dan kode *Reed-Muller* dapat memberikan tingkat keamanan dan integritas dokumen yang cukup baik. Setiap dokumen yang di-*hash* dengan *MD5* menghasilkan *message digest* unik yang kemudian dikodekan menggunakan kode *Reed-Muller* sebagai tanda tangan digital. Eksperimen menunjukkan bahwa:

1. Setiap perubahan pada isi dokumen, menyebabkan perubahan pada nilai *hash MD5* dan vektor tanda tangan digital yang dihasilkan.
2. Proses verifikasi berhasil membedakan antara dokumen asli dan dokumen yang telah dimodifikasi, menunjukkan bahwa sistem efektif dalam menjaga integritas dokumen.
3. Metode *Brute Force* yang digunakan untuk menemukan vektor *error* e yang sesuai dengan sindrom hasil *hash* dapat menemukan solusi tanda tangan yang valid, walaupun cukup memakan waktu untuk pencarian manual.

Penggunaan fungsi *hash MD5* memastikan proses *hashing* bersifat deterministik dan sensitif terhadap perubahan, sedangkan kode *Reed-Muller* berperan dalam menghasilkan dan memverifikasi tanda tangan digital berbasis vektor kesalahan. Secara keseluruhan, simulasi berhasil membuktikan bahwa sistem ini dapat mengamankan dokumen digital dan mendeteksi modifikasi data secara akurat.

4.7 Kajian Islam terhadap Implementasi Kode *Reed-Muller* pada Tanda Tangan Digital Berbasis *MD5*

Dalam simulasi penggunaan teknologi digital seperti implementasi *kode Reed-Muller* pada tanda tangan digital berbasis *MD5* dapat dikaji melalui prinsip-prinsip syariah, terutama terkait keamanan, kejujuran, dan amanah. Kode *Reed-Muller*, sebagai salah satu metode *error-correcting code*, berperan dalam mendeteksi dan memperbaiki kesalahan dalam transmisi data, sehingga dapat meningkatkan integritas tanda tangan digital.

Dalam perspektif Islam, penerapan teknologi keamanan informasi seperti kriptografi dan tanda tangan digital harus selaras dengan lima prinsip dasar syariat, yang dikenal sebagai *al-khawāʾid al-khamsah* (*maqāṣid al-sharīʿah*). Pertama, prinsip *ḥifẓ al-māl* (menjaga harta) menuntut agar sistem informasi mampu melindungi aset digital dan dokumen penting dari pemalsuan, pencurian, atau manipulasi. Dengan memastikan keutuhan data, teknologi ini turut menjaga hak milik seseorang dan mencegah kerugian finansial akibat penipuan.

Kedua, prinsip *ḥifẓ al-nasl* (menjaga keturunan) menjaga keberlangsungan dan kehormatan keturunan manusia, baik dalam bentuk fisik, identitas, maupun nasab yang sah secara hukum dan moral. Dalam konteks digital modern, prinsip ini dapat dimaknai lebih luas sebagai upaya melindungi identitas dan data keluarga dari penyalahgunaan, pemalsuan, atau manipulasi.

Ketiga, prinsip *ḥifẓ al-nafs* (menjaga jiwa) dapat diaplikasikan dalam konteks perlindungan data pribadi dan rahasia medis, di mana kebocoran informasi dapat membahayakan keselamatan fisik atau psikologis seseorang. Kriptografi memainkan peran penting dalam menjaga kerahasiaan tersebut.

Keempat, prinsip *ḥifẓ al-*ʻaql (menjaga akal) mengarah pada pentingnya menyebarkan informasi yang benar dan mencegah penyalahgunaan teknologi untuk menipu, menyebarkan *hoax*, atau mengaburkan kebenaran. Sistem keamanan digital dapat digunakan untuk memverifikasi sumber dan keaslian informasi, sehingga mendukung akal sehat dan berpikir jernih.

Terakhir, prinsip *ḥifẓ al-dīn* (menjaga agama) menuntut agar teknologi digunakan dengan niat yang ikhlas dan tidak untuk tujuan merusak moral atau nilai-nilai keislaman. Sistem keamanan informasi yang menjaga amanah, kejujuran, dan kerahasiaan sesuai ajaran Islam akan mendukung terpeliharanya nilai-nilai agama di era digital

Dengan demikian, setiap penerapan teknologi informasi khususnya dalam hal menjaga keaslian dan kerahasiaan dokumen digital hendaknya diuji dan diterapkan dengan memperhatikan kelima prinsip tersebut, agar manfaatnya bukan hanya bersifat teknis, tetapi juga bernilai syar'i dan etis.

Sebagaimana firman Allah dalam QS. Al-Mujadilah ayat 10 tentang larangan membocorkan rahasia yang merugikan. Dengan demikian, integrasi kode *Reed-Muller* dan tanda tangan digital tidak hanya bernilai teknis, tetapi juga bernilai ibadah jika digunakan untuk kemaslahatan umat dan sesuai dengan etika Islam dalam transaksi digital (*mu'āmalāt elektronik*). Terdapat hadits yang membahas tentang menjaga rahasia sebagai berikut.

Dari Anas *radhiyallahuʻanhu*, beliau berkata,"*Rasulullah shallallahu ʻalaihi wasallam mendatangiku dan di waktu itu aku sedang bermain-main dengan beberapa orang anak. Beliau shallallahu ʻalaihi wa sallam mengucapkan salam kepada kami, kemudian menyuruhku untuk sesuatu keperluannya. Oleh sebab itu aku terlambat mendatangi ibuku. Selanjutnya setelah aku datang, ibu lalu bertanya, ʻApakah yang menahanmu?*" (HR. Muslim, no. 2482).

Hadits ini menggaris bawahi pentingnya menjaga kerahasiaan sebagai bentuk penunaian amanah. Sebuah prinsip yang secara teknis diwujudkan melalui proses enkripsi dalam sistem tanda tangan digital berbasis kode *Reed-Muller*. Mekanisme enkripsi dalam sistem kriptografi ini berfungsi untuk melindungi kerahasiaan pesan dengan tingkat keamanan tinggi yang resistan terhadap berbagai serangan, termasuk ancaman komputasi kuantum. Implementasi ini secara efektif mencegah potensi kebocoran informasi dan pelanggaran kerahasiaan data.

Proses deskripsi yang hanya dapat dilakukan oleh penerima yang memiliki kunci valid menunjukkan bagaimana sistem ini menjamin bahwa pesan hanya dapat diakses oleh pihak yang berwenang. Simulasi membuktikan bahwa algoritma *decoding* dengan pendekatan *Brute Force* yang diterapkan mampu memulihkan pesan asli secara akurat sekaligus memperkuat aspek keamanan. Dengan demikian, integritas dan kerahasiaan informasi tetap terjaga, selaras dengan firman Allah SWT dalam QS. Al-Mujadalah ayat 10 tentang pentingnya menjaga amanah dalam komunikasi sebagai berikut:

اِنَّمَا النَّجْوٰى مِنَ الشَّيْطٰنِ لِيَحْزُنَ الَّذِيْنَ اٰمَنُوْا وَلَيْسَ بِضَاۤرِّهِمْ شَيْـًٔا اِلَّا بِاِذْنِ اللّٰهِۗ وَعَلَى اللّٰهِ فَلْيَتَوَكَّلِ

الْمُؤْمِنُوْنَ ﴿١٠﴾

Artinya : "*Sesungguhnya pembicaraan rahasia itu hanyalah dari setan, agar orang-orang yang beriman itu bersedih hati, sedangkan (pembicaraan) itu tidaklah memberi mudarat sedikit pun kepada mereka, kecuali dengan izin Allah. Hanya kepada Allah hendaknya orang-orang mukmin bertawakal*". (Kemenag, 2019).

Sebab turunnya ayat ini karena Ibnu Jarir meriwayatkan dari Qatadah, ia mengatakan; Dahulu orang-orang munafik melakukan pembicaraan rahasia di antara mereka dan tidak mau menyudahinya. Hal ini membuat orang orang mukmin merasa ada sesuatu dan berat hati. Maka Allah SWT menurunkan ayat,"

Sesungguhnya pembicaraan rahasia itu adalah dari setan".(Syahril & Maqasid, 2015)

Ayat ini melarang pembicaraan rahasia yang mengandung dosa dan permusuhan (*"Janganlah kamu membicarakan perbuatan dosa, permusuhan, dan durhaka kepada Rasul"*), yang secara implisit menegaskan bahwa informasi rahasia adalah amanah yang wajib dijaga. Dalam konteks modern, ini sejalan dengan prinsip kerahasiaan data dalam sistem kriptografi, di mana enkripsi berfungsi sebagai bentuk penjagaan amanah digital.

Implementasi nilai amanah dalam kehidupan nyata menuntut kejujuran dan komitmen untuk menjaga kebenaran isi amanah, sehingga membangun fondasi kepercayaan yang kokoh dalam hubungan sosial. Ketika seseorang konsisten menerapkan sifat amanah ini dengan sungguh-sungguh, akan tercipta dampak positif berupa tersampaikannya pesan atau tugas dengan aman tanpa terjadi kebocoran informasi. Selain itu, pihak yang menjaga amanah dengan baik akan mendapatkan kepercayaan lebih dari lingkungannya, sebagai buah dari integritas dan tanggung jawab yang ditunjukkan dalam menangani amanah tersebut.

BAB V

KESIMPULAN

5.1 Kesimpulan

Berdasarkan rumusan masalah beserta pembahasan di atas, dapat disimpulkan bahwa:

1. Setiap dokumen elektronik yang berbeda menghasilkan tanda tangan yang berbeda-beda. Hal ini karena tanda tangan digital yang dihasilkan bergantung pada nilai *message digest* yang diperoleh dari isi pesan dan kunci pribadi yang digunakan oleh penandatanganan atau pengirim.
2. Dokumen elektronik yang tidak mengalami perubahan isi akan menghasilkan nilai tanda tangan digital dan *message digest* yang tetap sama. Sebaliknya, apabila dokumen tersebut mengalami perubahan, nilai tanda tangan digital dan *message digest* yang dihasilkan menjadi berbeda dari aslinya. Hal ini terjadi karena setiap perubahan pada isi dokumen elektronik akan mengubah nilai *message digest***.**

5.2 Saran

Penelitian ini telah berhasil menunjukkan bahwa implementasi kode *Reed-Muller* pada tanda tangan digital berbasis *MD5* mampu mendeteksi perubahan data dan menghasilkan tanda tangan yang unik untuk setiap dokumen. Namun, penggunaan fungsi *hash MD5* yang dikenal memiliki kelemahan terhadap *collision* perlu dikaji ulang. Disarankan untuk menggunakan algoritma *hash* yang lebih kuat seperti SHA-256 pada penelitian selanjutnya. Selain itu, metode Brute Force yang digunakan dalam proses pencarian vektor *error* cukup memakan waktu, sehingga

pengembangan metode dekoding yang lebih efisien seperti algoritma dekoding sistematik akan meningkatkan performa sistem. Mengingat sistem ini masih bersifat simulatif, integrasi ke dalam platform digital nyata juga direkomendasikan untuk menguji efektivitas dan skalabilitasnya dalam konteks aplikasi dunia nyata.

DAFTAR PUSTAKA

Abbe, E., Shpilka, A., & Ye, M. (2021). Reed-Muller Codes: Theory and Algorithms. *IEEE Transactions on Information Theory*, *67*(6), 3251–3277. https://doi.org/10.1109/TIT.2020.3004749

Amiruddin, A. (2021). AMANAH DALAM PERSPEKTIF AL-QURAN (Studi Komparatif Tafsir Al-Misbah dan Al-Azhar). *Jurnal MUDARRISUNA: Media Kajian Pendidikan Agama Islam*, *11*(4), 833. https://doi.org/10.22373/jm.v11i4.4665

Azdy, R. A. (2016). Tanda tangan Digital Menggunakan Algoritme Keccak dan RSA. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, *5*(3), 184–191. https://doi.org/10.22146/jnteti.v5i3.255

Dahlan, C. H. (2015). Islam, Politik Dan Demokrasi. *Publiciana*, 1–15. https://journal.unita.ac.id/index.php/publiciana/article/view/49%0Ahttps://journal.unita.ac.id/index.php/publiciana/article/download/49/45

Ilmiyah, N. F. (2019). Kajian Tentang Kriptosistem Mceliece Dalam Menghadapi Tantangan Komputer Kuantum Di Era Revolusi Industri 4.0. *Prosiding Seminar Nasional MIPA Kolaborasi*, 216–226.

Ismail, M., & Makmur. (2020). Al-Qurţubī dan Metode Penafsirannya dalam Kitab al-Jāmi‘ li Aĥkām al-Qur’ān. *Pappasang*, *2*(2), 17–32. https://doi.org/10.46870/jiat.v2i2.68

Jamil. (2016). 01 Jurnal ISTISHLAH HAMKA DAN TAFSIR AL-AZHAR.pdf. In *Istishlahm: Jurnal Hukum Islam* (Vol. 2, Issue XII, pp. 121–143).

Jaya, A. K. (2017). Proses Decoding Kode Reed Muller Orde Pertama Menggunakan Transformasi Hadamard. *Jurnal Matematika, Statistika Dan Komputasi*, *13*(2), 122–127.

Kemenag, Q. (2019). *Qur’an Kemenag*. Kemenag. https://quran.kemenag.go.id/

Munir, R. (2006). Pengantar Kriptografi. *ITB, Bandung*.

Muttoo, S. K., & Kumar, S. (2013). Self-synchronising image steganography algorithms based on error-correcting codes. *International Journal of Electronic Security and Digital Forensics*, *5*(3–4), 297–316. https://doi.org/10.1504/IJESDF.2013.058670

Rangkuti, A. Z. F., & Fahmi, H. (2020). Implementasi Kriptografi Untuk Keamanan File Text Dengan Menggunakan Metode *MD5*. *Jurnal Nasional Komputasi Dan Teknologi Informasi (JNKTI)*, *3*(2), 170–175. https://doi.org/10.32672/jnkti.v3i2.2384

Saepulrohman, A., & Negara, T. P. (2021). Implementasi algoritma tanda tangan digital berbasis kriptografi kurva eliptik Diffie-Hellman. *Komputasi: Jurnal Ilmiah Ilmu Komputer Dan Matematika*, *18*(1), 22–28.

Santoso, M. H., Girsang, N. D., Siagian, H., Wahyudi, A., & Sitorus, B. A. (2019). Perbandingan Algoritma Kriptografi Hash *MD5* dan SHA-1. *Seminar Nasional Teknologi Informatika*, *2*(1), 54–59.

Suyuthi, I. (2019). Asbabun Nuzul. In *Jurnal Ilmiah Pendidikan dan Pembelajaran* (Vol. 3, Issue 2, pp. 55–61).

Syahril, A. M., & Maqasid, Y. (2015). *Terjemahan Asbabun Nuzul: sebab-sebab turunnya Ayat Al-Qur'an* (p. 542). Pustaka Al-Kautsar.

Zaatsiyah, N., & Djuniadi, D. (2021). IMPLEMENTING DIGITAL SIGNATURE WITH RSA AND *MD5* IN SECURING E-INVOICE DOCUMENT. *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, *5*, 129. https://doi.org/10.22373/cj.v5i2.10359

Zulkarnain Hasibuan, A. (n.d.). *Peretasan Password MySQL Menggunakan Algoritma Brute Force*. *70*.

LAMPIRAN

Lampiran 1. Script Pembangkit Kunci

```
import hashlib
import random

def generate_permutation_matrix(n):
    import numpy as np
    permutation_indices = np.random.permutation(range(n))
    permutation_matrix = np.zeros((n, n), dtype=int)
    permutation_matrix[np.arange(n), permutation_indices] = 1
    return permutation_matrix
# --- PARAMETER RM(1,3) ---
F=GF(2)
r = 1
m = 3
RM = codes.ReedMullerCode(F,r, m)
G = RM.generator_matrix()
print('Matrix G=\n',G)
H = RM.parity_check_matrix()
print('Matrix H=\n',H)

n = RM.length()
k = RM.dimension()
print(f"Kode RM({r},{m}) dengan n={n}, k={k}")

# --- Kunci Privat ---
S = random_matrix(GF(2), 4, algorithm="unimodular")
print('Matrix S=\n',S)
Sinv=S.inverse()
print('Matrix S invers=\n',Sinv)
P=matrix(GF(2),generate_permutation_matrix(8))
print('Matrix Q=\n',P)
Pinv=P.inverse()
print('Matrix Q invers=\n',Pinv)
H_pub = S * H * P
print('Matrix H aksen=\n',H_pub)
s_len = H.nrows()  # atau: s_len = n - k
```

Lampiran 2. Script Hash *MD5*, Penandatanganan dan Verifikasi

```
# --- Fungsi HASH: MD5 ke vektor biner sepanjang s_len ---
def MD5_hash_full(msg, length):
    bits = ""
    counter = 0
```

```
    while len(bits) < length:
        h = hashlib.MD5((msg + str(counter)).encode()).hexdigest()
        b = bin(int(h, 16))[2:].zfill(128)
        bits += b
        counter += 1
    return vector(GF(2), map(int, bits[:length]))

# --- PENANDATANGANAN ---
def sign(message, max_trials=100000000000000000000):
    s = message
    for _ in range(max_trials):
        s_ = S.inverse() * s
        print('\ns aksen=\n',S.inverse(),'x',s,'=',s_)
        for e_prime in GF(2)^n:
            if e_prime.hamming_weight() <= 2:
                if H * e_prime == s_:
                    print('lalu cari e\n HxeT=s aksen\n',H,'x',e_prime,'=',s_)
                    e = P.inverse() * e_prime
                    print('lalu hitung e aksen\n Q inversxeT=e
aksen\n',P.inverse(),'x',e_prime,'=',e)
                    return e
    raise ValueError("Tanda tangan gagal dibuat dalam batas percobaan")

# --- VERIFIKASI ---
def verify(message, signature):
    s = MD5_hash_full(message, s_len)
    return H_pub * signature == s and signature.hamming_weight() <= 2

# --- DEMO ---
hex_str = "b6c3794594d9c891ee6cdd31f0b799ef"
binary_list= vector(GF(2), list(reversed(Integer(hex_str, 16).digits(base=2))))
print('\nvektor biner=',binary_list)
for i in range(0, len(binary_list), 4):
    msg = binary_list[i:i + 4]
    sig = sign(msg)
    print("Tanda Tangan untuk pesan ini", msg)
    print('adalah ',sig)
    print('\nverifikasi\n')
    msgg=H_pub*sig
    print('H aksen x e aksen=hash awal')
    print(H_pub,'x',sig,'=',msgg)
```

RIWAYAT HIDUP

Maulana Agil Yuliarso lahir di Ngawi pada tanggal 9 Juni 2001, memiliki nama panggilan Agil. Tempat tinggalnya berada di RT 004 RW 004 Dusun Nglencong 1 Desa Dempil Kecamatan Geneng Kabupaten Ngawi. Anak ketiga dari empat bersaudara dari pasangan Bapak Setiarso dan Ibu Patmiatun. Masa pendidikan penulis di mulai dari TK Modrn, Ngawi dari tahun 2006 hingga 2007. Kemudian dilanjutkan pendidikan dasar di SDN Karang Tengah IV, Ngawi dan lulus pada tahun 2013. Penulis melanjutkan pendidikan jenjang menengah pertama di SMP 1 Negeri, Ngawi lulus pada tahun 2016, Kemudian menempuh pendidikan jenjang menengah atas di SMA 2 Negeri, Ngawi dan lulus pada tahun 2019. Pada tahun 2019 penulis melanjutkan pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang tepatnya di program studi Matematika.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Maulana Agil Yuliarso
NIM : 19610103
Fakultas / Program Studi : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Kode Reed-Muller Pada Tanda Tangan Digital Berbasis MD5
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Erna Herawati, M.Pd

No	Tanggal	Hal	Tanda Tangan
1.	26 Juni 2024	Konsultasi Topik Bab I	1.
2.	28 Juni 2024	Konsultasi Topik Bab II	2.
3.	1 Juli 2024	Konsultasi Topik Bab III	3.
4.	9 Juli 2024	Konsultasi Bab I,II, dan III	4.
5.	13 September 2024	Konsultasi Bab I Kajian Agama	5.
6	24 September 2024	Konsultasi Bab II Kajian Agama	6.
7.	30 September 2024	ACC kajian Agama Bab I dan II	7.
8.	10 Oktober 2024	ACC Bab I,II, dan III dan Seminar Proposal	8.
9.	14 Oktober 2024	ACC Seminar Proposal	9.
10.	13 Februari 2025	Konsultasi Revisi seminar proposal	10.
11.	7 Mei 2025	Konsultasi Bab IV dan V	11.
12.	14 Mei 2025	Konsultasi Kajian Agama Bab IV	12.
13.	16 Mei 2025	Konsultasi Kajian Agama Bab IV	13.
14.	20 Mei 2025	ACC Bab IV dan V	14.
15.	10 Juni 2025	ACC Kajian Agama Bab IV	15.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

16.	13 Juni 2025	ACC Seminar Hasil	16.
17.	18 September 2025	Konsultasi Revisi Seminar Hasil	17.
18.	13 Oktober 2025	ACC Matrik Revisi Seminar Hasil	18.
19.	16 Oktober 2025	ACC Sidang Skripsi	19.
20.	31 Oktober 2025	ACC Keseluruhan	20.

Malang , 31 Oktober 2025
Mengetahui,
Ketua Prodi Studi Matematika

Dr. Faehrur Rozi, M.Si
NIP. 19800527 200801 1 012