IMPLEMENTASI ALGORITMA KRIPTOSISTEM *MCELIECE* DENGAN MENGGUNAKAN KODE *REED-MULLER*

SKRIPSI

OLEH
EKA PUSPA ANGGRAINI
NIM. 200601110028

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK BRAHIM
MALANG
2024

IMPLEMENTASI ALGORITMA KRIPTOSISTEM *MCELIECE* DENGAN MENGGUNAKAN KODE *REED-MULLER*

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
Eka Puspa Anggraini
NIM. 200601110028

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK BRAHIM
MALANG
2024

IMPLEMENTASI ALGORITMA KRIPTOSISTEM *MCELIECE* DENGAN MENGGUNAKAN KODE *REED-MULLER*

SKRIPSI

Oleh
Eka Puspa Anggraini
NIM. 200601110028

Telah Diperiksa dan Disetujui Untuk Diuji
Malang, 29 Agustus 2024

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

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

IMPLEMENTASI ALGORITMA KRIPTOSISTEM *MCELIECE* DENGAN MENGGUNAKAN KODE *REED-MULLER*

SKRIPSI

Oleh
Eka Puspa Anggraini
NIM. 200601110028

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

Tanggal 10 September 2024

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. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini

Nama : Eka Puspa Anggraini

NIM : 200601110028

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Algoritma Kriptosistem *McEliece* Dengan Menggunakan Kode *Reed-Muller*

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

Malang, 10 September 2024

Eka Puspa Anggraini

NIM. 200601110028

MOTO

“Kamu adalah satu-satunya orang yang dapat mewujudkan masa depan-mu.”

(Kim Taehyung of BTS)

“Siapapun bisa kehilangan arah jalan mereka, kamu hanya membutuhkan keberanian untuk kembali berjalan di jalan yang asing dan menakutkan itu.”

(S.Coups of SEVENTEEN)

PERSEMBAHAN

Bismillaahirrahmaanirrahiim

Segala puji bagi Allah SWT yang senantiasa memberikan pertolongan dan kemudahan kepada penulis dalam melewati segala proses penyelesaian skripsi ini. Skripsi ini dipersembahkan sepenuhnya kepada:

Dua orang terhebat dalam hidup penulis, Ayah Sukatman dan Ibu Sri Hidayah, yang telah banyak berkorban dan memberikan nasihat serta do'a terbaik kepada penulis hingga dapat menyelesaikan skripsi ini. Kepada diriku sendiri yang telah bertahan sejauh ini untuk tidak menyerah dan percaya atas rencana Allah SWT yang lebih indah. Kepada adik penulis Muhammad Rafa Fauzi Akbar yang telah menjadi salah satu alasan penulis dengan semangat menyelesaikan skripsi ini. Serta teman-teman penulis yang dengan ikhlas memberikan bantuan, semangat dan do'a dalam menyelesaikan skripsi ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarokatuh.

Segala puji bagi Allah SWT atas rahmat, taufik serta hidayah-Nya, sehingga penulis mampu menyelesaikan penyusunan skripsi sebagai salah satu syarat untuk memperoleh gelar sarjana dalam bidang Matematika di Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dalam proses penyusunan skripsi ini, tentunya penulis mendapatkan bantuan dan arahan dari berbagai pihak. Untuk itu ucapan terimakasih yang sebesar-besarnya dan penghargaan yang setinggi-tingginya penulis tujukan kepada:

1. Bapak Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Ibu Prof. Dr. Sri Harini, M.Si., selaku dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Ibu Dr. Elly Susanti, M.Sc., selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Bapak Muhammad Khudzaifah, M.Si., selaku dosen wali dan dosen pembimbing 1 serta dosen penguji 2 yang senantiasa memberikan pengetahuan, bimbingan, arahan serta motivasi kepada penulis.
5. Ibu Erna Herawati, M.Pd., selaku dosen pembimbing 2 dan dosen penguji 3 yang senantiasa memberikan pengetahuan, bimbingan, arahan serta motivasi kepada penulis.
6. Bapak Hisyam Fahmi, M.Kom., selaku ketua penguji yang telah memberikan kritik, saran serta dukungan kepada penulis.
7. Bapak Mohammad Nafie Jauhari, M.Si., selaku dosen penguji 1 yang telah memberikan kritik, saran serta dukungan kepada penulis.
8. Segenap sivitas akademika Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang khususnya seluruh dosen yang telah memberikan banyak pengetahuan dan pengalaman berharga bagi penulis.
9. Kedua orang tua tercinta, bapak Sukatman dan ibu Sri Hidayah, adik tercinta Muhammad Rafa Fauzi Akbar dan seluruh keluarga tercinta yang senantiasa memberikan do'a, dukungan dan kasih sayang terbaik kepada penulis.

10. Seluruh member BTS yaitu Kim Namjoon, Kim Seokjin, Min Yoongi, Jung Hoseok, Park Jimin, Kim Taehyung, Jeon Jungkook yang semua nasihat, tulisan dan karyanya sangat memotivasi dan menghibur penulis dari awal masa perkuliahan hingga saat ini.
11. Seluruh member SEVENTEEN yaitu Choi Seungcheol, Yoon Jeonghan, Hong Jisoo, Moon Junhui, Kwon Soonyoung, Jeon Wonwoo, Lee Jihoon, Xu Minghao, Kim Mingyu, Lee Seokmin, Boo Seungkwan, Choi Hansol, Lee Chan yang telah hadir dan menghibur penulis di masa penelitian skripsi ini.
12. Seluruh teman terdekat penulis dari Alumni Darul Huda Mayak dan KKM Desa Banjarsari Malang Tahun 2022, terkhusus Zulfa, Binti, Tutut, Via, Acha, Harun, Tyo, Haikal, Windi, Hilda, Silva, Fina, Imas, Intan, yang seperti keluarga selama di perantauan dan selalu ada dalam suka duka penulis.
13. Seluruh teman seperjuangan mahasiswa angkatan 2020 Program Studi Matematika, terkhusus Lengga, Aldina, Sisil, Soviana, Ira, Lili, Aam, Lauha, Diva, yang selalu memberikan bantuan dan dukungan dalam menyelesaikan skripsi ini.
14. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis baik secara langsung maupun tidak langsung dalam menyelesaikan skripsi ini.

Semoga Allah SWT senantiasa melimpahkan rahmat dan karunia-Nya kepada kita semua. Penulis juga menyadari bahwa penelitian ini masih terdapat banyak kekurangan dikarenakan keterbatasan kemampuan dan pengetahuan penulis. Semoga dengan rahmat dan izin-Nya, skripsi ini dapat bermanfaat bagi penulis maupun pembaca. Aamiin.

Wassalamu'alaikum Warahmatullahi Wabarokatuh.

Malang, 10 September 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR LAMPIRAN

ABSTRAK

Anggraini, Eka Puspa. 2024. **Implementasi Algoritma Kriptosistem *McEliece* Dengan Menggunakan Kode *Reed-Muller***. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Kata Kunci: Kriptografi, Kriptosistem *McEliece*, Kode *Reed-Muller*, *Post-Quantum Cyptography*.

Penelitian ini berfokus pada implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*. Kriptosistem *McEliece* sebagai algoritma berbasis kode dianggap lebih efisien dan cukup aman dalam menghadapi masalah komputer kuantum. Penggunaan kode *Reed-Muller* yang disandingkan dengan kriptosistem *McEliece* juga menambah lapisan keamanan karena dapat melakukan banyak koreksi kesalahan. Pembahasannya terdiri dari proses hingga simulasi pembangkitan kunci, enkripsi, dekripsi serta perbandingan hasil implementasi dengan menggunakan parameter yang berbeda. Parameter kode $RM(r,m)$ yang digunakan yaitu $RM(1,5)$, $RM(2,5)$ dan $RM(3,5)$. Pada proses pembangkitan kunci dibangkitkan dua jenis kunci, yaitu kunci privat dan kunci publik. Kemudian proses enkripsi untuk mengamankan pesan, dimulai dengan mengkonversi pesan ke bentuk kode biner untuk disandikan menjadi cipherteks. Adapun proses dekripsi dilakukan untuk memulihkan pesan asli, proses ini juga memanfaatkan algoritma *decoding* kode *Reed-Muller*. Penelitian ini menunjukkan keberhasilan algoritma kriptosistem *McEliece* menggunakan kode *Reed-Muller*. Selain itu, berdasarkan parameter yang digunakan, koreksi kesalahan dapat dilakukan mulai 1-bit hingga 7-bit. Hasil implementasinya menunjukkan bahwa ukuran kunci memiliki pengaruh signifikan terhadap kemampuan koreksi error dan efisiensi waktu proses. Kunci berukuran lebih kecil menawarkan kemampuan koreksi lebih baik. Namun meningkatkan kompleksitas komputasi dan memperpanjang waktu pemrosesan. Sebaliknya, kunci berukuran lebih besar, lebih efisien dalam proses dan lebih cepat diimplementasikan, tetapi kemampuan koreksi lebih terbatas.

ABSTRACT

Anggraini, Eka Puspa. 2024. **Implementation of the McEliece Cryptosystem Algorithm Using Reed-Muller Codes**. Undergraduate Thesis. Mathematics Departement, Faculty Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Muhammad Khudzaifah, M.Si. (II) Erna Herawati, M.Pd.

Keywords: Cryptography, McEliece Cryptosystem, Reed-Muller Code, Post-Quantum Cyptography.

This research focuses on the implementation of McEliece cryptosystem algorithm using Reed-Muller code. The McEliece cryptosystem as a code-based algorithm is considered more efficient and secure enough to deal with quantum computer problems. The use of Reed-Muller code paired with the McEliece cryptosystem also adds a layer of security as it can perform multiple error corrections.The discussion consists of the process up to simulation of key generation, encryption, decryption and comparison of implementation results using different parameters. The $RM(r,m)$ code parameters used are $RM(1,5)$, $RM(2,5)$ and $RM(3,5)$. In the key generation process, two types of keys are generated, namely private and public keys. Then the encryption process to secure the message, starting with converting the message to binary code form to encode into ciphertext. As for the decryption process done to recover the original message, this process also utilizes the Reed-Muller code decoding algorithm. This research demonstrates the success of the McEliece cryptosystem algorithm using Reed-Muller codes. In addition, based on the parameters used, error correction can be done from 1-bit to 7-bit. The implementation results show that the key size has a significant influence on the error correction capability and runtime efficiency. Smaller key size offers better correction capability. However, it increases the computational complexity and lengthens the processing time. Conversely, larger keys are more efficient in processing and faster to implement, but the correction capability is more limited.

مستخلص البحث

أنغرايني، إيكا ڤوسڤا. ٢٠٢٤. **تنفيذ خوارزمية نظام تشفير *McEliece* باستخدام كود ريد مولر**. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: محمد خُذيفة، الماجستير. المشرفة الثانية: إيرنا هيراواتي، الماجستيرة.

الكلمات المفتاحية: التشفير، نظام التشفير *McEliece*، كود ريد–مولر، تشفير ما بعد الكم.

يركز هذا البحث على تنفيذ خوارزمية نظام تشفير *McEliece* باستخدام كود ريد–مولر. يعتبر نظام التشفير *McEliece* كخوارزمية قائمة على الشفرة أكثر كفاءة وأمانًا بما يكفي للتعامل مع مشاكل الحاسوب الكمي. كما يضيف استخدام كود ريد–مولر المقترن بنظام تشفير *McEliece* مستوى من الأمان حيث يمكنه إجراء تصحيحات متعددة للأخطاء. وتتكون المناقشة من عملية التي تصل إلى محاكاة توليد المفاتيح والتشفير وفك التشفير ومقارنة نتائج التنفيذ باستخدام معلمات مختلفة. معلمات كود $RM(r,m)$ المستخدمة هي $RM(1,5)$ و $RM(2,5)$ و $RM(3,5)$. في عملية إنشاء المفاتيح، يتم إنشاء نوعين من المفاتيح، وهما المفاتيح الخاصة والمفاتيح العامة. ثم تتم عملية التشفير لتأمين الرسالة، بدءاً بتحويل الرسالة إلى شكل الشفرة الثنائية لتشفيرها إلى النص المشفر. أما بالنسبة لعملية فك التشفير التي تتم لاستعادة الرسالة الأصلية، فتستخدم هذه العملية أيضًا خوارزمية فك تشفير كود ريد–مولر. يوضح هذا البحث نجاح خوارزمية نظام تشفير *McEliece* باستخدام أكواد ريد–مولر. بالإضافة إلى ذلك، بناءً على المعلمات المستخدمة ، يمكن إجراء تصحيح الخطأ من ١ بت إلى ٧ بت. تُظهر نتائج التنفيذ أن حجم المفتاح له تأثير كبير على القدرة على تصحيح الأخطاء وكفاءة وقت التشغيل. ويفضل إستعمال المفتاح الأصغر لتصحيحها. ومع ذلك، فإنه يزيد من التعقيد الحسابي ويطيل وقت المعالجة. وبالعكس، فإن المفاتيح الكبرى لتكون أكثر كفاءة في المعالجة وأسرع في التنفيذ، لكن عملية التصحيح تكون محدودة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kriptografi merupakan algoritma yang umumnya digunakan untuk mengamankan pesan pada proses transmisi. Kriptografi merupakan “ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain” (Ariyus, 2008). Pengamanan pesan dalam kriptografi dilakukan dengan mengubah sebuah pesan (*plaintext*) ke dalam bentuk pesan yang disandikan (*chipertext*) sehingga pesan tidak bisa terbaca oleh pihak yang tidak bersangkutan. Keamanan suatu pesan yang dimaksud terdiri dari beberapa aspek, diantaranya terjaganya rahasia dalam pesan, jaminan isi pesan asli, serta pengirim dan penerima pesan adalah yang sebenarnya dan tidak dapat di bantah. Hal tersebut bertujuan untuk mengantisipasi kebocoran isi dari pesan yang dikirimkan.

Proses transmisi pesan atau penyampaian pesan berkaitan dengan konteks amanah yaitu tentang suatu kepercayaan dan tanggung jawab yang diberikan untuk menyampaikan atau melaksanakan suatu amanah. Tanggung jawab yang dimaksud dalam penyampaian pesan adalah dengan menjaga keamanan pesan dan menyampaikannya kepada penerima yang seharusnya. Berkaitan dengan hal tersebut, terdapat ayat Al-Qur’an surat Al-Ma’arij ayat 32-35 yang membahas tentang amanah sebagai berikut.

وَالَّذِيْنَ هُمْ لِاَمٰنٰتِهِمْ وَعَهْدِهِمْ رَاعُوْنَ﴿٣٢﴾ وَالَّذِيْنَ هُمْ بِشَهٰدٰتِهِمْ قَاۤىِٕمُوْنَ﴿٣٣﴾ وَالَّذِيْنَ هُمْ عَلٰى صَلَاتِهِمْ يُحَافِظُوْنَ﴿٣٤﴾اُولٰۤىِٕكَ فِيْ جَنّٰتٍ مُّكْرَمُوْنَ﴿٣٥﴾

"Dan orang-orang yang memelihara amanat dan janjinya, dan orang-orang yang berpegang teguh pada kesaksiannya, dan orang-orang yang memelihara sholatnya, mereka itu dimuliakan di dalam surga."(Kemenag, 2019)

Ayat di atas menjelaskan seberapa penting untuk memelihara suatu amanah dan janji serta berpegang teguh pada suatu kesaksian. Dalam menjalankan atau menyampaikan suatu amanah tentunya harus bertanggung jawab untuk jujur dan menjaga kebenaran amanah sehingga menciptakan suatu kepercayaan. Selain itu manusia juga diperintahkan untuk senantiasa menjaga kesaksiannya yaitu tidak menambahi, mengurangi atau menyembunyikan kebenaran. Yang mana semua hal tersebut merupakan bagian dalam memelihara amanah. Terdapat juga maksud lain dari ayat ini yaitu larangan untuk berkhianat sebagaimana sifat orang-orang munafik (Katsir, 2015). Sehingga secara tidak langsung ayat ini mengajarkan untuk menjaga kepercayaan dan kejujuran. Adapun tujuan memelihara amanah tersebut juga untuk menjaga amanah agar tersampaikan dengan baik hanya kepada orang yang berhak menerimanya.

Kriptografi sangat umum digunakan pada proses transmisi pesan untuk menjaga keamanannya. Pada beberapa abad lalu sebelum adanya komputasi, algoritma kriptografi yang umum digunakan hanya memakai satu kunci untuk mengamankan pesan yang disebut sebagai kriptografi klasik. Sandi yang digunakan pada kriptografi klasik yaitu berbasis karakter yang proses penyandiannya diberlakukan pada tiap karakternya. Kemudian setelah adanya ilmu komputasi, berkembang kriptografi modern yang algoritmanya dibuat sedemikian kompleks karena dioperasikan menggunakan komputer. Sehingga untuk memecahkan pesan yang telah disandikan relatif lebih sulit tanpa mengetahui kuncinya.

Namun dalam beberapa dekade terakhir, ilmu komputasi telah mengalami perkembangan yang signifikan karena kemajuan komputer kuantum. Uji keamanan untuk beberapa algoritma kriptografi modern seperti RSA, *ElGamal* dan Kurva Eliptik pun dilakukan, akan tetapi tingkat keamanan algoritma tersebut hanya bertahan dengan baik terhadap serangan umum dan rentan terhadap serangan komputer kuantum (Pongsitammu et al., 2023). Hal ini memicu perkembangan *Post-Quantum Cryptography* yang tahan terhadap serangan komputer kuantum. Algoritma kriptografi pasca kuantum yang dapat digunakan dalam menghadapi serangan komputer kuantum yaitu sistem kriptografi berbasis kode. Sistem kriptografi berbasis kode yang dimaksud adalah kriptosistem *McEliece*.

Kriptosistem *McEliece* merupakan algoritma berbasis kode koreksi kesalahan dengan kunci asimetris (kunci publik) yang aspek keamanannya bergantung pada tingkat kesulitan *decoding*. Dalam menghadapi masalah komputer kuantum, kriptosistem *McEliece* dianggap lebih efisien dan cukup aman karena memiliki sistem keamanan yang tidak bergantung pada algoritma diskrit bilangan yang relatif mudah untuk dipecahkan komputer kuantum (Ilmiyah, 2019). Sehingga algoritma kriptosistem *McEliece* disebut memiliki tingkat keamanan lebih baik dibandingkan dengan algoritma kriptografi modern yang juga menggunakan kunci publik.

Terdapat berbagai penelitian mengenai kriptosistem *McEliece* yang disandingkan dengan algoritma kode koreksi kesalahan guna meningkatkan keamanan kriptosistem tersebut. Salah satunya penelitian (Setyawan & Utomo, 2022) yang menghasilkan bahwa kode *Golay* diperpanjang dapat diterapkan pada kriptosistem *McEliece* dengan koreksi *error* sebanyak 3 bit. Kemudian pada penelitian (Oktavia dkk., 2023) menghasilkan bahwa kode *Reed-Solomon* juga

dapat diterapkan pada kriptosistem *McEliece* dengan koreksi *error* juga sebanyak 3 bit. Sejauh ini penggunaan kriptosistem *McEliece* yang disandingkan dengan kode pengoreksi kesalahan cukup efisien dan aman, meskipun pada penerapannya dengan menggunakan kode *Golay* dan kode *Reed-Solomon* hanya dapat mengoreksi *error* tidak lebih dari 3 bit.

Dalam mekanisme kriptosistem McEliece diperlukan kode linier atau kode siklik yang merupakan bagian dari algoritma pengoreksi kesalahan. Terdapat penelitian mengenai salah satu jenis kode linier pengoreksi kesalahan yang menghasilkan bahwa kode *Reed-Muller* dapat melakukan banyak koreksi kesalahan (Cornelissenn et al., 2020). Kode *Reed-Muller* merupakan bagian dari keluarga kode linier pengoreksi kesalahan. Kode *Reed-Muller* dinotasikan dengan $RM(r, m)$ dengan $r \leq m$, yang mana r merepresentasikan urutan atau orde kode dan m mewakili suatu elemen yang membantu menentukan suatu panjang blok. Kode *Reed-Muller* dianggap sebagai kandidat yang cukup bagus untuk disandingkan dengan kriptosistem *McEliece*, karena kode tersebut dapat melakukan banyak koreksi kesalahan.

Uraian di atas merupakan beberapa aspek yang mendasari untuk dilakukannya penelitian berjudul "Implementasi Algoritma Kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*". Dalam penelitian ini, kriptosistem *McEliece* merupakan algoritma yang proses enkripsi dan dekripsinya diimplementasikan dengan kode *Reed-Muller*. Kemudian kode *Reed-Muller* berperan dalam proses pembentukan matriks generator, *encoding* dan *decoding* pada dekripsi algoritma kriptosistem *McEliece*. Sehingga penelitian ini diharapkan dapat menghasilkan wawasan baru tentang *Post-Quantum Cryptography*.

1.2 Rumusan Masalah

Berdasarkan penjabaran pada latar belakang sebelumnya, ada beberapa rumusan masalah untuk di kaji lebih lanjut yakni:

1. Bagaimana proses dan simulasi pembangkitan kunci untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*?
2. Bagaimana implementasi dan simulasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*?
3. Bagaimana perbandingan hasil implementasi menggunakan parameter kode $RM(1,5)$, $RM(2,5)$ dan $RM(3,5)$?

1.3 Tujuan Penelitian

Adapun berdasarkan rumusan masalahnya, penelitian ini memiliki tujuan sebagai berikut:

1. Untuk mengetahui proses dan hasil simulasi pembangkitan kunci untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.
2. Untuk mengetahui implementasi dan hasil simulasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.
3. Untuk mengetahui perbandingan hasil implementasi menggunakan parameter kode $RM(1,5)$, $RM(2,5)$ dan $RM(3,5)$.

1.4 Manfaat Penelitian

Adapun dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Penulis

   Penelitian ini diharapkan dapat memberikan wawasan yang mendalam bagi penulis tentang proses pembentukan matriks generator, *encoding*, *decoding*, enkripsi dan dekripsi pada implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.

2. Bagi Pembaca

   Penilitian ini diharapkan dapat menjadi bahan referensi ataupun wawasan baru mengenai kriptografi pasca kuantum khususnya pada implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* dan tingkat keamanannya.

1.5 Batasan Masalah

Adapun dalam penelitian ini terdapat beberapa batasan masalah yaitu:

1. Batas kode *Reed-Muller* yang digunakan yaitu $r = 1,2,3$ dan $m = 5$.
2. Pembangkitan kunci G berdasarkan definisi sifat rekursif kode *Reed-Muller*.
3. Pesan berupa vektor kode biner berdasarkan tabel ASCII 256-bit.
4. Proses enkripsi dan dekripsi berdasarkan algoritma kriptosistem *McEliece*.
5. Konsep pemrograman kriptosistem *McEliece* dan kode *Reed-Muller* menggunakan *python* dengan memanfaatkan *library Sagemath*.

1.6 Definisi Istilah

Adapun berbagai istilah yang berkaitan dengan penelitian ini yaitu:

1. Pesan merupakan informasi yang maknanya dapat dipahami, atau dalam kriptografi disebut *plaintext*.

2. *Ciphertext* merupakan suatu pesan hasil enkripsi yang maknanya tidak dapat dipahami.
3. *Codeword* merupakan kata kode dari hasil mengkodekan pesan.
4. Enkripsi merupakan proses menyandikan pesan asli sehingga menghasilkan *ciphertext* atau pesan yang maknanya tidak dapat dipahami.
5. Dekripsi merupakan proses untuk mengembalikan pesan hasil enkripsi atau *ciphertext* sehingga kembali menjadi pesan asli.
6. *Encoding* merupakan proses mengkodekan pesan menjadi *codeword*.
7. *Decoding* merupakan proses menguraikan *codeword* sehingga kembali menjadi pesan asli.
8. Kunci merupakan parameter yang digunakan untuk menjalankan proses enkripsi, dekripsi, *encoding* maupun *decoding*.

BAB II

KAJIAN TEORI

2.1 Grup

Definisi 2.1

Misalkan operasi biner $*$ didefinisikan untuk elemen-elemen himpunan G. Maka G adalah sebuah grup yang berhubungan dengan $*$ yang memenuhi kondisi berikut (Gilbert & Gilbert, 2015).

1. G tertutup terhadap $*$. Mengimplikasikan bahwa $a, b \in G$ dan $a * b$ ada di G.
2. $*$ bersifat asosiatif. Untuk semua $a, b, c \in G$ maka berlaku $a * (b * c) = (a * b) * c$.
3. G memiliki sebuah elemen identitas e. Terdapat sebuah e di G sedemikian sehingga $a * e = e * a = a$ untuk semua $a \in G$.
4. G mengandung invers. Untuk setiap $a \in G$, terdapat $b \in G$ sedemikian sehingga $a * b = b * a = e$

Definisi 2.2

Misalkan G adalah sebuah grup yang berhubungan dengan $*$. Maka G disebut grup komutatif atau grup abelian, jika $*$ bersifat komutatif. Artinya, $a * b = b * a$ untuk semua $a, b \in G$ (Gilbert & Gilbert, 2015).

2.2 Ring

Definisi 2.3

Sebuah himpunan R dengan dua operasi, biasanya disebut penjumlahan dan perkalian, dilambangkan dengan $(R, +, *)$, disebut sebuah ring jika

1. R tertutup terhadap penjumlahan: $a, b \in R$ mengimplikasikan $a + b \in R$.
2. Penjumlahan di R bersifat asosiatif: $a + (b + c) = (a + b) + c$ untuk semua $a, b, c \in R$.
3. R memuat identitas aditif 0: $a + 0 = 0 + a = a$ untuk semua $a \in R$.
4. R memuat invers aditif: Untuk a di R, terdapat $-a$ di R sedemikian sehingga $a + (-a) = (-a) + a = 0$.
5. Penjumlahan di R bersifat komutatif: $a + b = b + a$ untuk semua $a, b \in R$.
6. R tertutup terhadap perkalian: $a, b \in R$ mengimplikasikan $a \cdot b \in R$.
7. Perkalian di R asosiatif: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ untuk semua $a, b, c \in R$.
8. Dua hukum distributif berlaku di R: $a \cdot (b + c) = a \cdot b + a \cdot c$ dan $(a + b) \cdot c = a \cdot c + b \cdot c$ untuk semua $a, b, c \in R$.

(Gilbert & Gilbert, 2015)

Adapun sebuah ring disebut ring komutatif jika $a \cdot b = b \cdot a$ untuk semua $a, b \in R$ (Menezes et al., 1996).

2.3 Lapangan (*Fields*)

Definisi 2.4

Sebuah lapangan adalah sebuah ring $(R, +, *)$ di mana semua elemen yang bukan nol memiliki invers multiplikatif adalah sebuah grup abelian (Lint, 2007).

Teorema 2.1

Setiap ring berhingga R dengan setidaknya dua elemen sedemikian rupa sehingga untuk setiap $a, b \in R$, berlaku $ab = 0 \Rightarrow (a = 0 \vee b = 0)$ adalah sebuah lapangan (Lint, 2007).

2.3.1 Lapangan Hingga (*Galois Field* atau *Finite Field*)

Finite field juga dikenal dengan sebutan *Galois field* yang merupakan field dengan jumlah elemen terbatas. Penggunaan *finite field* dalam kriptografi misalnya pada algoritma AES (*Advanced Encryption Standard*).

Definisi 2.5

Lapangan hingga adalah sebuah bidang F yang berisi sejumlah elemen hingga. Urutan F adalah jumlah elemen dalam F (Menezes et al., 1996).

Jika F adalah lapangan berhingga, maka F memuat elemen-elemen p^m untuk beberapa bilangan prima p dan bilangan bulat $m \geq 1$. Untuk setiap pangkat prima orde p^m, terdapat suatu lapangan berhingga yang unik (hingga isomorfisma) dengan orde p^m. Lapangan ini dilambangkan dengan $\mathbb{F}_{p^m}$, atau terkadang dengan $GF(p^m)$.

Secara informal, dua buah lapangan adalah isomorfis jika mereka secara struktural sama, meskipun representasi elemen-elemen lapangan mereka mungkin berbeda. Perhatikan bahwa jika p adalah bilangan prima maka $\mathbb{Z}_p$ adalah sebuah lapangan, dan karenanya setiap lapangan dengan order p isomorfis dengan $\mathbb{Z}_p$. Kecuali dinyatakan lain, lapangan berhingga $\mathbb{F}_p$ selanjutnya akan diidentifikasikan dengan $\mathbb{Z}_p$.

2.3.2 Ruang Vektor atas Lapangan Hingga

Definisi 2.6

Sebuah ruang vektor V atas sebuah lapangan F adalah grup abelian $(V, +)$, bersama dengan sebuah operasi perkalian $F \times V \to V$ sehingga untuk semua $a, b \in F$ dan $v, w \in V$, memenuhi aksioma berikut (Menezes et al., 1996).

1. $a(v+w) = av + aw$.
2. $(a+b)v = av + bv$.
3. $(ab)v = a(bv)$.
4. $1v = v$.

Elemen-elemen V disebut vektor, sedangkan elemen-elemen F disebut skalar. Operasi grup $+$ disebut penjumlahan vektor, sedangkan operasi perkalian disebut perkalian skalar.

Definisi 2.7

Misalkan $\mathbb{F}_q$ adalah lapangan berhingga dengan order q. Sebuah himpunan tak kosong V, bersama dengan beberapa penjumlahan vektor dan perkalian skalar dengan elemen-elemen $\mathbb{F}_q$, adalah sebuah ruang vektor (atau ruang linier) atas $\mathbb{F}_q$ jika memenuhi semua kondisi-kondisi berikut. Untuk semua $u, v, w \in V$ dan untuk semua $\lambda, \mu \in \mathbb{F}_q$ (Ling & Xing, 2004).

1. $u + v \in V$;
2. $(u+v) + w = u + (v+w)$;
3. Terdapat sebuah elemen $0 \in V$ dengan sifat $0 + v = v = v + 0$ untuk semua $v \in V$;
4. Untuk setiap $u \in V$ terdapat sebuah elemen dari V, yang disebut $-u$, sehingga $u + (-u) = 0 = (-u) + u$;
5. $u + v = v + u$;
6. $\lambda v \in V$;
7. $\lambda(u+v) = \lambda u + \lambda v$, $(\lambda + \mu)u = \lambda u + \mu u$;
8. $(\lambda\mu)u = \lambda(\mu u)$;
9. Jika 1 adalah identitas perkalian dari $\mathbb{F}_q$, maka $1u = u$.

2.4 Kode Biner

Definisi 2.8

Sebuah kode atas kode alfabet $F_2 = \{0,1\}$ disebut sebagai kode biner (Ling & Xing, 2004).

Kode biner merupakan sistem penulisan angka atau data menggunakan dua simbol yaitu 0 dan 1. Setiap digit kode biner disebut bit (*binary digit*). Kode biner biasa digunakan pada transmisi pesan maupun data melalui jaringan komputer. Salah satu contoh himpunan kode biner yaitu $C_1 = (2,4) = \{00,01,10,11\}$.

Representasi kode biner bersifat fleksibel, dengan batasan dapat menetapkan kombinasi biner apapun (disebut kata kode) ke data apapun selama data dikodekan secara unik. Untuk jenis data numerik erat kaitannya dengan bilangan biner, harus merepresentasikan rentang data yang dibutuhkan dan sebaiknya data direpresentasikan sedemikian rupa agar memungkinkan dilakukan penghitungan operasi aritmatika umum yang sederhana dan mudah dipahami. Adapun untuk jenis data non numerik tidak terikat dengan bilangan biner dan tidak ada operasi aritmatika yang diterapkan, sehingga memberikan fleksibilitas lebih.

Adapun sistem operasi penjumlahan, pengurangan, perkalian dan pembagian pada kode biner sama dengan operasi pada bilangan desimal namun perbedaaanya terdapat pada basis bilangan yang digunakan (Sutanta, 2005).

Dasar penjumlahan dalam sistem biner sebagai berikut.

1. $0 + 0 = 0$
2. $1 + 0 = 1$
3. $0 + 1 = 1$

4. $1 + 1 = 0$, mengikuti prinsip membawa (*carry*) ketika hasil penjumlahan dua bit adalah 2 (dalam biner 10), membawa 1 ke bit selanjutnya di sebelah kiri.

Dasar pengurangan dalam sistem biner sebagai berikut.

1. $0 - 0 = 0$
2. $1 - 0 = 1$
3. $0 - 1 = 1$, mengikuti prinsip meminjam (*borrow*) ketika bit pengurang lebih besar dari bit yang dikurangi, meminjam 1 dari bit selanjutnya di sebelah kiri.
4. $1 - 1 = 0$

Dasar perkalian dalam sistem biner sebagai berikut.

1. $0 \times 0 = 0$
2. $1 \times 0 = 0$
3. $0 \times 1 = 0$
4. $1 \times 1 = 1$

Dasar pembagian dalam sistem biner sebagai berikut.

1. $1 \div 1 = 1$
2. $1 \div 0 =$ *undefined*, karena tidak bisa membagi sesuatu dengan nol.
3. $0 \div 1 = 0$
4. $0 \div 0 =$ *undefined*, karena tidak ada pembagi ataupun yang dibagi.

2.5 Operasi *Bitwise*

Operasi *bitwise* merupakan operasi yang digunakan untuk menyelesaikan operasi-operasi bilangan biner yang dilakukan bit demi bit (Handoko, 2019). Dengan kata lain, operator ini berperan untuk memanipulasi bit. Proses ini sangat penting dilakukan jika program berinteraksi dengan perangkat keras. Bahasa

pemrograman sifatnya lebih berorientasi pada data (*data-oriented*), namun perangkat keras masih berorientasi terhadap bit (*bit-oriented*). Artinya perangkat keras ingin menjaga data input dan output yang berjalan di dalamnya dalam bentuk bit terpisah. Perlu diperhatikan bahwa operasi ini hanya dapat dilakukan pada bilangan char (tipe data untuk menyimpan satu karakter) dan int (tipe data untuk bilangan bulat), karena dapat mengakomodasi tipe *byte* dan *word* dalam bit. Tabel berikut menunjukkan operator bitwise dalam bahasa C.

Tabel 2.1 Tabel operasi *bitwise*

Operator	Jenis Operasi	Contoh Penggunaan
&	*Bitwise* AND	$0\&0 = 0$ $0\&1 = 0$ $1\&0 = 0$ $1\&1 = 1$
|	*Bitwise* OR	$0\|0 = 0$ $0\|1 = 1$ $1\|0 = 1$ $1\|1 = 1$
^	*Bitwise* XOR (*Exclusive OR*)	$0\hat{}0 = 0$ $0\hat{}1 = 1$ $1\hat{}0 = 1$ $1\hat{}1 = 0$
~	*Bitwise Complement* (NOT)	$\sim 0 = 1$ $\sim 1 = 0$

2.6 Kode ASCII

ASCII (*American Standard Code for Information Interchange*) merupakan standar internasional kode yang merepresentasikan huruf atau simbol dalam komputer dan alat komunikasi lain. Standar ASCII memiliki 128 karakter ($0 - 127$) menggunakan 7 bit yang mewakili huruf, angka, tanda baca dan karakter khusus lainnya. Setiap karakternya diwakili oleh bilangan biner 7 bit dari 0000000

hingga 1111111. Selain standar ASCII 7 bit dengan 128 karakter, juga terdapat ekstended ASCII yang merupakan perluasan dari standar ASCII dan dapat merepresentasikan karakter lebih banyak. ASCII ini menggunakan 8 bit dan dapat mewakili hingga 256 karakter $(0-255)$ dari 00000000 hingga 11111111. Akan tetapi extended ASCII tetap kompatibel dengan standar ASCII 7 bit, yaitu pada 128 karakter pertama ASCII 8 bit tetap sama dengan 128 karakter pada ASCII 7 bit.

2.7 Matriks

Matriks adalah susunan segi empat siku-siku dari bilangan-bilangan atau skalar-skalar atau fungsi yang dibatasi dengan tanda kurung. Bilangan yang terdapat pada matriks disebut sebagai entri atau elemen (Kusumawati, 2009).

Definisi 2.9

Matriks adalah susunan skalar elemen-elemen dalam bentuk baris dan kolom yang berukuran m baris dan n kolom $(m \times n)$ (Munir, 2010).

Adapun bentuk umum dari matriks berukuran (berordo) $m \times n$ adalah sebagai berikut.

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

Entri yang ada pada matriks tersebut disebut entri ij yang terletak pada baris i dan kolom j dan umumnya ditulis sebagai $A = [a_{ij}]$ (Kusumawati, 2009). Baris-baris pada matriks di atas merupakan m deret horizontal berupa skalar-skalar berikut,

$$(a_{11}, a_{12}, \dots, a_{1n}), (a_{21}, a_{22}, \dots, a_{2n}), \dots, (a_{m1}, a_{m2}, \dots, a_{mn}).$$

Lalu kolom-kolomnya yaitu n deret vertikal berupa skalar-skalar berikut,

$$\begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix}, \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix}, \dots, \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}.$$

2.7.1 Operasi Matriks

1. Penjumlahan dan Pengurangan Matriks

   Apabila $A = [a_{ij}]$ dan $B = [b_{ij}]$ adalah sebarang dua matriks yang ukurannya sama $m \times n$, maka penjumlahan matriks A dan matriks B adalah dengan menjumlahkan entri yang bersesuaian pada kedua matriks tersebut secara bersamaan (Kusumawati, 2009). Begitu pula untuk pengurangan yaitu dengan mengurangi entri yang bersesuaian pada kedua matriks tersebut secara bersamaan. Namun jika kedua matriks memiliki ukuran berbeda, maka operasi penjumlahan dan pengurangan tidak dapat dilakukan. Berikut gambaran tentang operasi penjumlahan dan pengurangan matriks A dan matriks B.

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

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

2. Perkalian Matriks

   Perkalian matriks terbagi menjadi tiga, yaitu perkalian matriks dengan bilangan (skalar), vektor baris dengan vektor kolom, dan matriks dengan matriks. Perkalian matriks A dengan k bilangan (skalar) adalah mengalikan setiap entri matriks dengan skalar k secara bersamaan.

$$kA = \begin{bmatrix} ka_{11} & ka_{12} & \cdots & ka_{1n} \\ ka_{21} & ka_{22} & \cdots & ka_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ka_{m1} & ka_{m2} & \cdots & ka_{mn} \end{bmatrix}$$

Berikut ini adalah contoh perkalian matriks A dengan (skalar) 2 dan -1,

$$A = \begin{bmatrix} 2 & -1 & 5 \\ 3 & 6 & 4 \end{bmatrix}; 2A = \begin{bmatrix} 4 & -2 & 10 \\ 6 & 12 & 8 \end{bmatrix}; -A = \begin{bmatrix} -2 & 1 & -5 \\ -3 & -6 & -4 \end{bmatrix}.$$

Untuk perkalian vektor baris (n kolom) dengan vektor kolom (n baris), hasil perkalian matriks baris $A = [a_i]$ atau $A = [a_1, a_2, \dots, a_n]$ dan matriks kolom $B = [b_i]$ atau $B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$ dengan jumlah entri yang sama didefinisikan sebagai skalar atau matriks 1×1. Hasil tersebut diperoleh dengan mengalikan entri-entri yang bersesuaian dan menjumlahkannya sebagaimana berikut.

$$AB = a_1b_1 + a_2b_2 + \cdots + a_nb_n = \sum_{k=1}^{n} a_nb_n \qquad (2.1)$$

Perhatikan contoh berikut di mana dari perkalian vektor baris A dan vektor kolom B memperoleh hasil skalar yaitu 27.

$$A = (2 \quad 3 \quad 4) \text{ dan } B = \begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix}$$

$$AB = (2 \quad 3 \quad 4)\begin{pmatrix} 5 \\ 3 \\ 2 \end{pmatrix} = 2 \times 5 + 3 \times 3 + 4 \times 2 = 27.$$

Kemudian pada perkalian matriks dengan matriks, jumlah kolom matriks pertama harus sama dengan jumlah baris matriks kedua agar dapat dikalikan. Misalnya $A = [a_{ik}]$ matriks berukuran $m \times r$ dan $B = [b_{kj}]$ matriks berukuran $r \times n$, maka hasil perkalian AB adalah matriks

berukuran $m \times n$ (Kusumawati, 2009). Adapun gambaran untuk memperoleh entri-entri (ij) pada hasil perkalian matriks sebagai berikut

$$\begin{bmatrix} a_{11} & \cdots & a_{1r} \\ \cdot & \cdots & \cdot \\ a_{i1} & \cdots & a_{ir} \\ \cdot & \cdots & \cdot \\ a_{m1} & \cdots & a_{mr} \end{bmatrix} \begin{bmatrix} b_{11} & \cdots & b_{1j} & \cdots & b_{1n} \\ \cdot & \cdots & \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot & \cdots & \cdot \\ \cdot & \cdots & \cdot & \cdots & \cdot \\ b_{r1} & \cdots & b_{rj} & \cdots & b_{rn} \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \cdot & \cdots & \cdot \\ \cdot & c_{ij} & a_{ir} \\ \cdot & \cdots & \cdot \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}$$

$$\sum_{k=1}^{r} a_{ik} b_{kj} = a_{i1} b_{1j} + a_{i2} b_{2j} + \cdots + a_{ir} b_{rj} = c_{ij} \qquad (2.2)$$

Berikut contoh perkalian matriks $A_{2\times3}$ dengan $B_{3\times2}$ dan matriks $A_{3\times1}$ dengan $B_{1\times3}$

a. Jika $A_{2\times3} = \begin{bmatrix} 3 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix}$ dan $B_{3\times2} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 3 \end{bmatrix}$.

$A_{2\times3} B_{3\times2} = C_{2\times2}$

$$\begin{bmatrix} 3 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 3 \end{bmatrix} = C_{2\times2}$$

$$\begin{bmatrix} 3\times1 + 1\times3 + 2\times2 & 3\times2 + 1\times1 + 2\times3 \\ 2\times1 + 1\times3 + 3\times2 & 2\times2 + 1\times1 + 3\times3 \end{bmatrix} = \begin{bmatrix} 10 & 13 \\ 11 & 14 \end{bmatrix}$$

$B_{3\times2} A_{2\times3} = M_{3\times3}$

$$\begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 3 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} = M_{3\times3}$$

$$\begin{bmatrix} 1\times3 + 2\times2 & 1\times1 + 2\times1 & 1\times2 + 2\times3 \\ 3\times3 + 1\times2 & 3\times1 + 1\times1 & 3\times2 + 1\times3 \\ 2\times3 + 3\times2 & 2\times1 + 3\times1 & 2\times2 + 3\times3 \end{bmatrix} = \begin{bmatrix} 7 & 3 & 8 \\ 11 & 4 & 9 \\ 12 & 5 & 13 \end{bmatrix}$$

b. Jika $A_{3\times1} = \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix}$ dan $B_{1\times3} = [2 \quad 3 \quad 4]$,

$A_{3\times1} B_{1\times3} = C_{3\times3}$; $\quad \begin{bmatrix} 5 \\ 3 \\ 1 \end{bmatrix} [2 \quad 3 \quad 4] = \begin{bmatrix} 10 & 15 & 20 \\ 6 & 9 & 12 \\ 2 & 3 & 2 \end{bmatrix}$.

2.7.2 Jenis-Jenis Matriks

Terdapat beberapa jenis matriks antara lain.

1. Matriks Identitas

   Matriks identitas (I_k) merupakan matriks yang entri pada diagonal utamanya adalah 1 dan entri diluar diagonal utama adalah 0 (Kusumawati, 2009). Contohnya adalah sebagai berikut.

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Matriks Transpose

   Matriks transpose merupakan matriks yang berasal dari pertukaran baris-baris dan kolom-kolom (Munir, 2010). Misal $A = [a_{ij}]$ berukuran $m \times n$, maka transposenya dinyatakan dengan A^T yaitu matriks berukuran $n \times m$. Di mana jika $A^T = [b_{ij}]$, maka $b_{ij} = a_{ji}$ untuk $i = 1,2, \dots, n$ dan $j = 1,2, \dots, m$. Di bawah ini gambaran matriks A dan transposenya.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}; A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}$$

3. Matriks Invers

   Jika A matriks kuadrat dan dapat diperoleh matriks B demikian sehingga $AB = BA = I$, matriks A disebut *invertible* atau dapat dibalik dan matriks B disebut invers dari matriks A. Kemudian invers matriks A dinyatakan dengan A^{-1}. Sehingga $AA^{-1} = I$ dan $A^{-1}A = I$.

   Misal $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ memiliki invers jika dan hanya jika $ad - bc \neq 0$ sehingga invers dari matriks A adalah sebagai berikut.

$$A^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \begin{bmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}$$

Kemudian di bawah ini contoh menerapakan rumus mencari invers.

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}; A^{-1} = \frac{1}{1.3-2.1}\begin{bmatrix} 3 & -2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{1} & -\frac{2}{1} \\ -\frac{1}{1} & \frac{1}{1} \end{bmatrix} = \begin{bmatrix} 3 & -2 \\ -1 & 1 \end{bmatrix}$$

4. Matriks Determinan

Misalkan A merupakan sebuah matriks kuadrat. Determinan dari A dinotasikan oleh $\det(A)$ sebagai jumlah semua hasil elementer bertanda dari A (Kusumawati, 2009). Determinan A juga dapat dinotasikan dengan tanda Δ (delta) dan $|A|$. Determinan A biasa didefinisikan sebagai berikut.

$$\det(A) = \sum \pm a_{1j_1} a_{2j_2} \dots a_{nj_n} \qquad (2.3)$$

Berikut gambaran untuk determinan matriks berukuran 2×2 dan 3×3.

a. $\det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11}a_{22} - a_{12}a_{21}$

b. $\det\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}$

$$-a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}$$

2.8 Ekspansi Kofaktor

Ekspansi kofaktor merupakan metode yang penting dalam matematika, khususnya pada bidang aljabar linier, di mana digunakan untuk menghitung determinan dari sebuah matriks. Namun sebelumnya perlu dipahami mengenai konsep dasarnya yaitu minor dan kofaktor, untuk memudahkan penerapan ekspansi

kofaktor untuk menemukan deteminan dari matriks yang memiliki ukuran lebih besar.

Definisi 2.10

Jika A matriks kuadrat, maka minor entri a_{ij} dinotasikan dengan M_{ij} dan didefinisikan sebagai determinan submatriks yang tetap setelah baris ke-i dan kolom ke-j dicoret dari A. Kemudian $(-1)^{i+j}M_{ij}$ didefinisikan sebagai kofaktor entri a_{ij} dan dinotasikan dengan C_{ij}.

Berikut akan diberikan contoh dari minor dan kofaktor. Misalkan,

$$A = \begin{bmatrix} 3 & 2 & 7 \\ 1 & 4 & 2 \\ 3 & 1 & 2 \end{bmatrix}$$

Minor entri a_{11} adalah

$$M_{11} = \begin{vmatrix} 3 & 2 & 7 \\ 1 & 4 & 2 \\ 2 & 1 & 3 \end{vmatrix} = \begin{vmatrix} 4 & 2 \\ 1 & 3 \end{vmatrix} = 10$$

Kofaktor entri a_{11} adalah

$$C_{11} = (-1)^{1+1}M_{11} = M_{11} = 10$$

Teorema 2.2

Determinan matriks A yang berukuran $n \times n$ dapat dihitung dengan mengalikan entri-entri suatu baris atau kolom dengan kofaktor-kofaktornya dan menambahkan hasil-hasil kali yang dihasilkan, yaitu untuk setiap $1 \leq i \leq n$ dan $1 \leq j \leq n$ (Kusumawati, 2009).

Maka untuk ekspansi kofaktor sepanjang kolom ke-j yaitu

$$\det(A) = a_{1j}C_{1j} + a_{2j}C_{2j} + \cdots + a_{nj}C_{nj} \qquad (2.4)$$

dan untuk ekspansi kofaktor sepanjang baris ke-i adalah

$$\det(A) = a_{i1}C_{i1} + a_{i2}C_{i2} + \cdots + a_{in}C_{in}. \qquad (2.5)$$

2.9 Kriptografi

Kriptografi berasal dari kata *crypto* dan *graphia* yang merupakan istilah dari bahasa Yunani. *Crypto* memiliki arti *secret* atau rahasia dan *graphia* memiliki arti *writing* atau tulisan. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Ariyus, 2008). Kriptografi merupakan algoritma yang digunakan untuk mengamankan pesan agar pesan tidak diketahui pihak lain pada saat proses transmisi. Adapun definisi kriptografi lainnya, Kriptografi merupakan ilmu yang mempelajari teknik matematika yang berkaitan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, autentikasi entitas, dan autentikasi asal data (Menezes et al., 1996).

Kriptografi memiliki beberapa tujuan diantaranya untuk menjaga kerahasiaan suatu pesan agar tidak terbaca oleh pihak yang tidak berhak, untuk otentikasi yaitu mengidentifikasi keaslian pesan dan pihak yang terlibat, untuk menjamin tidak ada perubahan bagian pesan hingga sampai kepada penerima pesan dan untuk mencegah penyangkalan dari pengirim yang mengirim pesan ataupun penerima yang menerima pesan. Dalam penerapannya, kriptografi memiliki komponen dasar dan algoritma yang berperan dalam tahapan menyandikan pesan.

2.9.1 Komponen Kriptografi

Kriptografi mempunyai beberapa komponen dasar dalam prinsip penerapan keilmuan kriptografi, antara lain

1. Pesan, *Plaintext* dan *Ciphertext*

   Pesan merupakan data atau informasi yang disampaikan melalui proses komunikasi oleh seseorang (pengirim pesan) kepada orang lain

(penerima pesan). Pesan dalam keilmuan kriptografi biasanya disebut *plaintext*. *Plaintext* merupakan teks (pesan) asli yang diproses menggunakan algoritma kriptografi untuk menjadi *ciphertext* (Ariyus, 2008). *Plaintext* berperan sebagai sebuah pesan asli yang akan disandikan sebelum dikirim kepada penerima. Sedangkan *ciphertext* merupakan suatu pesan yang telah melalui proses enkripsi yaitu menggunakan algoritma kriptografi (Ariyus, 2008). *Ciphertext* berisi pesan yang telah disandikan yang mana pesan tersebut menjadi tidak terbaca dan tidak bermakna.

2. Pengirim dan Penerima

   Dalam proses penyampaian pesan (informasi) tentu melibatkan pihak tertentu. Pihak yang dimaksud diantaranya pengirim dan penerima pesan. Pengirim berperan sebagai pihak yang menyampaikan pesan kepada penerima. Sedangkan, penerima berperan sebagai pihak yang menerima pesan yang disampaikan pengirim. Adapun pihak pengirim dan penerima tidak selalu diperankan manusia, melainkan dapat berupa peralatan teknologi seperti mesin, komputer dan lain-lain.

3. Enkripsi dan Dekripsi

   Ilmu kriptografi berperan penting dalam mengamankan pesan ketika proses transmisi. Proses yang digunakan untuk mengamankan pesan yaitu enkripsi dan dekripsi. Enkripsi merupakan proses transformasi terhadap teks asli sehingga menghasilkan teks sandi atau sehingga tidak bisa terbaca oleh pihak yang tidak berhak. Kemudian dekripsi merupakan proses memulihkan kembali teks sandi menjadi teks asli (Sadikin, 2012).

4. Kunci

   Kunci merupakan masukan bagi algoritma enkripsi berupa nilai yang bebas terhadap teks asli untuk menentukan hasil keluaran algoritma enkripsi (Sadikin, 2012). Adapun kunci terbagi menjadi dua jenis, yaitu kunci publik (*public key*) dan kunci rahasia (*private key*). Kunci publik merupakan kunci yang dipublikasikan atau dapat diketahui semua pihak. Sedangkan kunci rahasia hanya diketahui satu pihak atau dirahasiakan.

5. Kriptanalisis (*Cryptanalysis*)

   Kriptanalisis merupakan analisis sandi atau ilmu dan seni untuk mengungkapkan teks asli tanpa harus mengetahui kunci yang sah. Adapun pihak yang melakukan analisis sandi disebut kriptanalis. Analisis sandi juga dapat menemukan kelemahan suatu algoritma kriptografi, yang mana dengan analisis kode dapat diperoleh teks asli hingga mendapatkan kunci dari sandi yang dienkripsi dengan algoritma tertentu (Ariyus, 2008).

2.9.2 Algoritma Kriptografi

Algoritma kriptografi merupakan langkah-langkah logis untuk mengamankan pesan dari pihak yang tidak berhak atas pesan tersebut (Ariyus, 2008). Algoritma kriptografi berdasarkan pada kuncinya yaitu sebagai berikut.

1. Algoritma Kunci Simetri

   Algoritma kunci simetri merupakan algoritma yang pada proses enkripsi dan dekripsi menggunakan satu kunci. Dengan kata lain, kunci yang digunakan pada proses enkripsi dan dekripsi adalah sama. Adapun

keamanan algoritma ini bergantung pada kunci tersebut. Pihak lain dapat melakukan proses enkripsi dan dekripsi terhadap pesan apabila mengetahui kunci yang digunakan. Algoritma dengan kunci simetri antara lain *Data Encryption Standard* (DES), RC2, RC4, RC5, RC6, *International Data Encryption Algorithm* (IDEA), *Advanced Encryption Standard* (AES), *One Time Pad* (OTP) dan sebagainya (Ariyus, 2008).

2. Algoritma Kunci Asimetri

   Algoritma kunci asimetri merupakan algoritma yang pada proses enkripsi dan dekripsi menggunakan kunci yang berbeda. Kunci pada algoritma ini terbagi menjadi dua yaitu kunci publik (*public key*) yang dapat diketahui siapapun dan kunci rahasia (*secret key*) yang hanya boleh diketahui satu orang. Adapun antara kunci publik dan kunci rahasia tersebut saling berkaitan. Kunci publik hanya dapat digunakan mengenkripsi pesan, sedangkan dekripsi pesan hanya dapat dilakukan oleh orang yang mengetahui kunci rahasia. Algoritma kunci asimetri juga lebih aman dibandingkan algoritma kunci simetri. Algoritma dengan kunci asimetri antara lain *Digital Signature Algorithm* (DSA), *Rivest Shamir Adleman* (RSA), *Diffie-Hellman*, *Elliptic Curve Cryptography* (ECC), Kriptografi Quantum dan lain sebagainya (Ariyus, 2008).

2.10 Kriptografi Klasik

Kriptografi klasik merupakan teknik penyandian dengan kunci simetri guna menyembunyikan pesan asli yang dapat dipahami menjadi pesan yang tidak terbaca dan tidak dapat dipahami menggunakan metode substitusi atau transposisi.

Kriptografi klasik menggunakan metode substitusi yaitu melakukan penggantian setiap karakter pesan asli dengan karakter lain. Algoritma kriptografi klasik yang menggunakan metode substitusi diantaranya *Caesar Cipher*, *Affine Cipher*, *Vigenere Cipher, Hill Cipher* dan *Playfair Cipher*. Adapun jika menggunakan metode transposisi yaitu dengan memindahkan posisi karakter pesan asli ke posisi lain tanpa mengubah nilai aslinya (Sadikin, 2012).

2.11 Kriptografi Modern

Kriptografi modern merupakan perkembangan dalam penyandian pesan yang mengacu pada kriptografi klasik. Dalam kriptografi modern, teknik penyandiannya memiliki tingkat kerumitan yang lebih kompleks dibandingkan kriptografi klasik. Hal ini dikarenakan pengoperasiannya yang menggunakan komputer. Berbagai algoritma kriptografi modern diantaranya algoritma simetris, algoritma asimetris dan algoritma hibrida yang memanfaatkan dua tingkatan kunci, yaitu kunci rahasia (simetri) untuk enkripsi data dan pasangan kunci rahasia dan kunci publik untuk pemberian tanda tangan digital serta melindungi kunci simetri (Ariyus, 2008).

2.12 Kriptografi Pasca Kuantum (*Post-Quantum Cryptography*)

Telah diketahui bahwa skema kriptografi yang digunakan secara luas seperti, RSA dan ECC dapat dipecahkan dengan menggunakan serangan komputer kuantum berskala besar (Shor, 1997). Serangan komputer kuantum merujuk pada jenis serangan siber yang menggunakan kekuatan komputasi dari komputer kuantum untuk menembus atau melemahkan sistem keamanan kriptografi yang digunakan saat ini. Komputer kuantum mampu memecahkan masalah yang sangat

kompleks dengan lebih cepat daripada komputer klasik, termasuk beberapa algoritma kriptografi yang saat ini dianggap aman. Sehingga dikembangkan kriptografi kuantum untuk menghadapi tantangan tersebut. Kriptografi Kuantum lahir pada awal tahun 1970-an. Sistem kriptografi kuantum memanfaatkan hubungan ketidakpastian Heisenberg, yang menyatakan bahwa mengukur sistem kuantum secara umum akan mengganggunya dan menghasilkan informasi yang tidak lengkap mengenai keadaannya sebelum pengukuran (Tilborg, 2006).

Kriptografi kuantum digunakan untuk meningkatkan keamanan dengan mendeteksi adanya penyadapan. Sehingga penyadapan pada saluran komunikasi kuantum menyebabkan gangguan yang tidak dapat dihindari, yang memperingatkan pengguna yang sah. Berdasarkan permasalahan kuantum tersebut, dikembangkan algoritma yang dapat mengatasinya. *Post-quantum cryptography* (PQC) merupakan algoritma pasca kuantum yang dikembangkan berdasarkan pada masalah matematika yang tahan terhadap serangan komputer kuantum (Takagi et al., 2021). Adapun beberapa pendekatan kriptografi pasca kuantum seperti *lattice-based cryptography*, *code-based cryptography*, *hash-based cryptography*, dan lain-lain. Adapun skema yang terkait dengan *code-based cryptography* contohnya kriptosistem *McEliece*.

2.13 Kode Linier

Definisi 2.11

Sebuah kode linier q-ary $\mathcal{C}$ adalah sebuah subruang linier dari $\mathbb{F};$. Jika $\mathcal{C}$ memiliki dimensi k maka $\mathcal{C}$ disebut sebuah kode $[n, k]$ (Lint, 2007).

Kode $[n, k, d]$ sebagai notasi untuk sebuah kode linier k-dimensi dengan panjang n dan jarak minimum d. Dalam teori pengkodean, kode linier umumnya direpresentasikan dalam bentuk matriks yaitu matriks generator dan matriks *parity-check* yang mana digunakan pada proses *encoding* dan *decoding*.

Definisi 2.12

Sebuah matriks generator G untuk sebuah kode linier $\mathcal{C}$ adalah sebuah matriks berukuran $k \times n$ yang baris-barisnya merupakan basis dari $\mathcal{C}$ (Lint, 2007).

Definisi 2.13

Sebuah matriks *parity-check* H untuk sebuah kode linier $\mathcal{C}$ adalah sebuah matriks generator untuk kode dual $\mathcal{C}^{\perp}$ (Ling & Xing, 2004).

Jika $\mathcal{C}$ adalah kode linier $[n, k]$, maka matriks generator untuk $\mathcal{C}$ haruslah matriks $k \times n$ dan matriks *parity-check* untuk $\mathcal{C}$ haruslah matriks $(n - k) \times n$.

2.14 Kode *Reed-Muller*

Kode *Reed-Muller* merupakan salah satu keluarga tertua dari kode linier pengoreksi kesalahan biner. Kode *Reed-Muller* dilambangkan dengan $RM(r, m)$ dan mengikuti format $[n, k, d]$. Pada lambang $RM(r, m)$, r merepresentasikan orde untuk kode *Reed-Muller* dan m berfungsi dalam membantu menentukan panjang blok, di mana r dan m dapat berupa bilangan asli yang mencakup nol dengan $r \leq m$. Kode *Reed-Muller* memiliki panjang blok $n = 2^m$ dengan panjang untuk pesan yaitu $k = \sum_{i=0}^{r} \binom{m}{i}$ dan jarak $d = 2^{m-r}$ (MacWilliams & Sloane, 1988). Cara untuk membentuk kode *Reed-Muller* yaitu dengan mempertimbangkan kode-kode tersebut sebagai sebuah definisi sifat rekursif.

Dengan menggunakan definisi sifat rekursif pertimbangkan dahulu kode *Reed-Muller* orde 0, $RM(0,m)$ di mana $m \geq 0$. Telah didefinisikan $RM(0,0) = \{0,1\}$, $RM(0,1) = \{00,11\}$, $RM(0,2) = \{0000,1111\}$ dan cara ini dapat untuk nilai yang lebih tinggi. Kemudian untuk kode *Reed-Muller* orde 1, $RM(1,m)$ di mana $m \geq 1$, maka $RM(1,1) = \{00,01,10,11\}$. Sehingga diperoleh definisi sifat rekursif untuk kode *Reed-Muller* yaitu, $RM(r,m) = \{(x, y+x) | x \in RM(r, m-1), y \in RM(r-1, m-1)\}$. Dari definisi tersebut, apabila diterapkan untuk kode $RM(1,2)$ adalah sebagai berikut.

$$RM(1,2) = \{(x, y+x) | x \in RM(1,1), y \in RM(0,1)\}$$

$$RM(1,2) = \{0000,0011,0101,0110,1001,1010,1100,1111\}$$

Dalam kode linier biasanya menggunakan matriks generator pada proses transmisi pesan. Adapun untuk membentuk matriks generator yang didasarkan pada kode *Reed-Muller* dengan merepresentasikan kodenya kedalam bentuk matriks, dan matriks generatornya dinotasikan sebagai $G(r,m)$. Berikut merupakan rumus yang diterapkan untuk membentuk $G(r,m)$.

$$G(0,m) = [1 \quad 1 \quad \ldots \quad 1] \tag{2.6}$$

$$G(r,m+1) = \begin{bmatrix} G(r,m) & G(r,m) \\ 0 & G(r-1,m) \end{bmatrix} \tag{2.7}$$

$$G(m,m) = \begin{bmatrix} G(m-1,m) \\ 0 \quad \ldots \quad 0 \quad 1 \end{bmatrix} \tag{2.8}$$

Kemudian di bawah ini beberapa contoh dari generator matriks berdasarkan persamaan 2.6, 2.7 dan 2.8.

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

$$G(0,2) = [1 \quad 1 \quad 1 \quad 1] \tag{2.9}$$

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

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

$$G(1,3) = \begin{bmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2.10)$$

Kemudian pada kode *Reed-Muller* terdapat proses *encoding* dan *decoding*. Proses *encoding* untuk mengubah pesan p ke dalam bentuk *codeword* c. Caranya dengan mengalikan pesan p dengan matriks generator $G(r, m)$.

$$c = p \times G(r, m)$$

Sedangkan proses *decoding* berfungsi memulihkan pesan dari *codeword* agar kembali ke pesan asli p. Terlebih dahulu dengan memetakan tiap vektor baris dari matriks generator ke dalam bentuk monomial. Kemudian tentukan vektor-vektor karakteristik untuk setiap baris matriks generator yang digunakan. Vektor karakteristik didapat dengan mengikuti operasi *bitwise* AND pada tabel 2.1 dengan mengalikan bit demi bit dari monomial-monomial yang tidak ada dalam monomial suatu vektor baris dan komplemennya. Langkah *decoding*nya sebagai berikut:

1. Dimulai dari baris paling bawah pada matriks generator, tentukan vektor karakteristik baris tersebut. Lakukan perkalian titik mengikuti persamaan 2.1 untuk masing-masing vektor karakteristik dengan pesan yang diterima.
2. Ambil sebagian besar nilai dari hasil perkalian titik pada langkah pertama dan tetapkan nilai tersebut sebagai koefisien dari baris tersebut.
3. Lakukan langkah pertama dan kedua untuk setiap baris (kecuali baris paling atas). Untuk membentuk M_y yaitu dengan mengalikan setiap koefisien dengan barisnya dan dijumlahkan seluruhnya. Jumlahkan M_y dengan pesan yang diterima, jika vektor yang dihasilkan memiliki lebih banyak angka 1

daripada 0, maka koefisien baris paling atas adalah 1, jika sebaliknya, lebih banyak angka 0 maka koefisiennya 0. Vektor yang dibentuk oleh urutan koefisien dimulai dari baris paling atas adalah pesan asli.

Berikut contoh vektor pesan $p = [0110]$ berukuran 1×4 yang entrinya adalah bilangan biner 0 dan 1, dengan $G(1,3)$ telah diketahui pada persamaan 2.10.

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

Lakukan proses *encoding* untuk memperoleh *codeword* c.

$$c = p \times G(r,m) = [0110] \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = [01100110]$$

Diperoleh vektor *codeword* $c = [01100110]$ berukuran 1×8 yang entrinya terdiri dari bilangan biner 0 dan 1 dan akan diuraikan melalui proses *decoding*. Kerjakan langkah *decoding* 1 dan 2 bersamaan. Untuk baris paling bawah (baris 4), vektor karakteristiknya $x_2x_1, \overline{x_2}\,\overline{x_1}, \overline{x_2}x_1, x_2\overline{x_1}$.

$$x_2x_1 = [00110011] \times [01010101] = [00010001]$$

$$\overline{x_2}\,\overline{x_1} = [11001100] \times [10101010] = [10001000]$$

$$\overline{x_2}x_1 = [11001100] \times [01010101] = [01000100]$$

$$x_2\overline{x_1} = [00110011] \times [10101010] = [00100010]$$

Lalu ambil setiap vektor karakteristik yang dihasilkan dan dikalikan titik dengan vektor pesan yang diterima. Juga lakukan proses ini pada baris dua dan tiga.

$$[00010001] \cdot [01100110] = 0$$

$$[10001000] \cdot [01100110] = 0$$

$$[01000100] \cdot [01100110] = 0$$

$$[00100010] \cdot [01100110] = 0$$

Koefisien baris 4 bernilai 0.

Untuk baris 3, vektor karakteristiknya $x_3x_1, \overline{x_3}\ \overline{x_1}, \overline{x_3}x_1, x_3\overline{x_1}$. Jadi

$$x_3x_1 = [00001111] \times [01010101] = [00000101]$$

$$\overline{x_3}\ \overline{x_1} = [11110000] \times [10101010] = [10100000]$$

$$\overline{x_3}x_1 = [11110000] \times [01010101] = [01010000]$$

$$x_3\overline{x_1} = [00001111] \times [10101010] = [00001010]$$

Sehingga diperoleh,

$$[00000101] \cdot [01100110] = 1$$

$$[10100000] \cdot [01100110] = 1$$

$$[01010000] \cdot [01100110] = 1$$

$$[00001010] \cdot [01100110] = 1$$

dan koefisien baris 3 bernilai 1.

Untuk baris 2, vektor karakteristiknya $x_2x_3, \overline{x_2}\ \overline{x_3}, \overline{x_2}x_3, x_2\overline{x_3}$. Jadi

$$x_2x_3 = [00110011] \times [00001111] = [00000011]$$

$$\overline{x_2}\ \overline{x_3} = [11001100] \times [11110000] = [11000000]$$

$$\overline{x_2}x_3 = [11001100] \times [00001111] = [00001100]$$

$$x_2\overline{x_3} = [00110011] \times [11110000] = [00110000]$$

Sehingga diperoleh,

$$[00000011] \cdot [01100110] = 1$$

$$[11000000] \cdot [01100110] = 1$$

$$[00001100] \cdot [01100110] = 1$$

$$[00110000] \cdot [01100110] = 1$$

dan koefisien baris 2 bernilai 1.

Selanjutnya mulai langkah 3 yaitu hitung M_y

$$M_y = \sum koefisien\ baris\ x_i\ (vektor\ baris\ x_i)$$

$$M_y = 0[00001111] + 1[00110011] + 1[01010101] = [01100110]$$

$$M_y + c = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0] + [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0] = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$$

Karena hasil penjumlahan dari M_y dengan pesan yang diterima adalah vektor yang semuanya 0, maka koefisien baris 1 adalah 0. Kemudian bentuk vektor dari urutan koefisien dimulai dari baris 1. Sehingga diperoleh pesan asli $[0110]$.

2.15 Kriptosistem *McEliece*

Kriptosistem *McEliece* merupakan skema enkripsi kunci publik yang didasarkan pada kode pengoreksi kesalahan (Menezes et al., 1996). Ide skema ini adalah untuk memilih sebuah kode tertentu yang algoritma penguraiannya efisien, lalu menyamarkan kode tersebut sebagai sebuah kode linier umum. Sebuah dekripsi dari kode asli dapat berfungsi sebagai kunci privat, sedangkan deskripsi dari kode yang telah diubah berfungsi sebagai kunci publik. Kriptosistem *McEliece* terkenal sebagai skema enkripsi kunci publik pertama yang menggunakan pengacakan dalam proses enkripsi.

Sejauh ini kriptosistem *McEliece* tahan terhadap berbagai kriptanalisis dan operasi enkripsi dan dekripsi pada kriptosistem ini juga relatif cepat, namun kurang mendapat perhatian karena memiliki kunci publik yang relatif besar (Menezes et al., 1996). Kriptosistem memiliki (k, n, t) sebagai parameter sistem yang umum, di mana k menunjukkan dimensi kode dan jumlah bit pesan yang akan di enkripsi, n menunjukkan panjang kode dari kode linier yang digunakan dalam kriptosistem *McEliece* dan t menunjukkan bobot kesalahan yang dapat dikoreksi oleh kode linier

yang digunakan dalam kriptosistem *McEliece*. Berikut algoritma pembangkitan kunci pada kriptosistem *McEliece* yang mana dilakukan oleh penerima.

1. Membuat matriks generator G berukuran $k \times n$ untuk sebuah kode biner (n, k)-linier yang dapat memperbaiki t *error*, dan algoritma penguraiannya diketahui.
2. Membuat matriks *non-singular* biner S berukuran $k \times k$.
3. Pilih sembarang matriks permutasi P berukuran $n \times n$.
4. Hitung $G' = SGP$ sehingga menghasilkan matriks berukuran $k \times n$.

Kemudian untuk algoritma kriptosistem *McEliece* pengirim mengenkripsi sebuah pesan m untuk dikirimkan kepada penerima, yang kemudian didekripsi oleh peerima. Berikut enkripsi pada kriptosistem *McEliece* yang dilakukan penggirim.

1. Menyiapkan kunci publik G'.
2. Merepresentasikan pesan sebagai sebuah vektor biner p dengan panjang k.
3. Membuat sebuah vektor kesalahan biner acak e dengan panjang n yang memiliki bobot kesalahan $t = \frac{d-1}{2}$.
4. Hitung rumus enkripsi untuk vektor biner $c = pG' + e$.

Untuk mendapatkan kembali plainteks m dari c, penerima melakukan langkah dekripsi berikut.

1. Menyiapkan kunci privat S, G, P.
2. Hitung $c' = cP^{-1}$, di mana

$$c' = cP^{-1} = (pG' + e)P^{-1} = pSG + eP^{-1} = pSG + e'$$

3. Gunakan algoritma *decoding* untuk kode yang dihasilkan oleh G untuk menguraikan $pSG + e'$ menjadi pS.
4. Hitunglah $p = pSS^{-1}$.

2.16 Kajian Integrasi Topik dengan Al-Qur'an dan Hadits

Kata amanah berasal dari bahasa arab yang merupakan bentuk mashdar dari *amina-ya'manu-amnan-wa amanatan* yang berarti aman, tentram, tenang, dan hilangnya rasa takut (Shihab, 2007). Dalam bahasa Indonesia, amanah memiliki arti yang dipercayakan (dititipkan) kepada orang, keamanan atau ketentraman, dan dapat dipercaya atau setia. Pengertian amanah secara khusus yaitu sikap tanggung jawab seorang yang diberikan titipan dengan mengembalikannya kembali kepada pemiliknya. Secara umum amanah yaitu menyimpan rahasia, tulus dalam memberi masukan dan menyampaikan pesan kepada pihak yang sesuai dengan permintaan dari yang berpesan. Adapun amanah secara lahir merupakan suatu tanggung jawab yang Allah SWT bebankan kepada hamba-hambanya dan juga janji yang telah Allah SWT berikan kepada hambanya. Demikian amanah merupakan pemenuhan hak-hak manusia, baik terhadap dirinya sendiri, orang lain maupun kepada Allah SWT, dan bertanggung jawab terhadap kepercayaan yang diterimanya untuk dilaksanakan dengan sebaik-baiknya (Shihab, 2007). Sehingga sifat amanah ini sangat berkaitan dengan sifat jujur, sabar, berani, tanggung jawab, adil, menjaga kemuliaan diri dan memenuhi janji.

Algoritma kriptografi merupakan algoritma untuk mengamankan pesan agar terjaga pada saat proses transmisi pesan. Pada proses transmisi pesan atau penyampaian pesan memiliki kaitan dengan konteks amanah yaitu tentang suatu kepercayaan dan tanggung jawab yang diberikan untuk menyampaikan atau melaksanakan suatu perintah. Tanggung jawab yang dimaksud dalam penyampaian pesan adalah dengan menjaga keamanan pesan dan menyampaikannya kepada

penerima yang seharusnya. Berkaitan dengan hal tersebut, terdapat ayat Al-Qur'an surat Al-Baqarah ayat 283 yang membahas tentang amanah sebagai berikut.

وَاِنْ كُنْتُمْ عَلٰى سَفَرٍ وَّلَمْ تَجِدُوْا كَاتِبًا فَرِهٰنٌ مَّقْبُوْضَةٌۗ فَاِنْ اَمِنَ بَعْضُكُمْ بَعْضًا فَلْيُؤَدِّ الَّذِى اؤْتُمِنَ اَمَانَتَهٗ وَلْيَتَّقِ اللّٰهَ رَبَّهٗۗ وَلَا تَكْتُمُوا الشَّهَادَةَۗ وَمَنْ يَّكْتُمْهَا فَاِنَّهٗٓ اٰثِمٌ قَلْبُهٗۗ وَاللّٰهُ بِمَا تَعْمَلُوْنَ عَلِيْمٌ ﴿٢٨٣﴾

"Jika kamu dalam perjalanan, sedangkan kamu tidak mendapatkan seorang pencatat, hendaklah ada barang jaminan yang dipegang. Akan tetapi, jika sebagian kamu memercayai sebagian yang lain, hendaklah yang dipercayai itu menunaikan amanatnya (utangnya) dan hendaklah dia bertaqwa kepada Allah, Tuhannya. Janganlah kamu menyembunyikan kesaksian karena siapa yang menyembunyikannya, sesungguhnya hatinya berdosa. Allah Maha Mengetahui apa yang kamu kerjakan." (Lajnah Pentashihan Mushaf Al-Qur'an, 2019)

Ayat di atas menjelaskan tentang suatu keadaan di mana ketika dalam perjalanan terjadi hutang-piutang dan tidak memperoleh saksi sebagai pencatat transaksi tersebut, maka hendaklah yang berhutang untuk memberikan suatu jaminan kepada pemberi hutang. Namun apabila seseorang dengan seorang lainnya saling mempercayai, tidak akan berdosa jika tidak mencatat atau tidak mendapat saksi. Kemudian ayat tersebut juga memberikan perintah untuk menunaikan suatu amanah dan bertaqwa kepada Allah SWT serta larangan menyembunyikan dan melebih-lebihkan kesaksian (Katsir, 2015). Sebagaimana juga disebutkan dalam hadits berikut.

مِنْ رِوَايَة قَتَادَة، عَنِ الْحَسَن، عَنْ سَمُرَة: اَنَّ رَسُوْلَ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ قَالَ:عَلَى الْيَدِ مَا اَخَذَتْ حَتَّى تُؤَدِّيَهُ. (رواه الامام احمد واهل السنن)

"Kewajiban tangan adalah mempertanggungjawabkan amanah yang diterimanya, sehingga ia melaksanakan (pengembalian)nya." (H.R. Ahmad dan Ahlus Sunan) (Katsir, 2015)

Dalam hadits ini juga menjelaskan keadaan pada ayat sebelumnya. Tangan yang menerima amanah wajib bertanggung jawab terhadap amanah tersebut.

Tangan yang menerima amanah yaitu pemberi hutang yang memegang jaminan wajib bertanggung jawab atas jaminan yang dibawanya, dan yang berhutang juga wajib bertanggung jawab dalam hal mengembalikan hutangnya kepada pemberi hutang. Hal ini menunjukkan suatu kewajiban pihak yang diberikan amanah untuk benar-benar bertanggung jawab atas amanah tersebut, dan apabila tidak dilaksanakan akan dimintai pertanggungjawaban di akhirat. Adapun tujuan menyampaikan amanah tersebut juga untuk menjaga agar amanah tersampaikan dengan baik kepada orang yang berhak. Secara tidak langsung, ayat dan hadits sebelumnya juga mengajarkan sifat kepercayaan, kejujuran serta tanggung jawab. Sebagaimana dalam menyampaikan amanah kita bertanggung jawab untuk berlaku jujur dan menjaga kebenaran amanah tersebut sehingga dapat menciptakan suatu kepercayaan.

Hal yang hampir serupa juga dijelaskan pada hadits berikut. Allah SWT mengabarkan, bahwa Dia memerintahkan untuk menunaikan amanah kepada ahlinya di dalam hadits al-Hasan dari Samurah, bahwa Rasulullah SAW bersabda:

وَفِي حَدِيْثِ الْحَسَن، عَنْ سَمُرَة، اَنَّ رَسُوْلَ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ قَالَ: اَدِّ الْاَمَانَةَ اِلَى مَنِ ائْتَمَنَكَ وَلَا تَخُنْ مَنْ خَانَكَ.(رواه الامام احمد واهل السنن)

"Tunaikanlah amanah kepada orang yang memberikan amanah, dan jangan khianati orang yang berkhianat kepadamu". (H.R. Ahmad dan Ahlus Sunan) (Katsir, 2015)

Berdasarkan hadits tersebut dianjurkan kembali untuk menunaikan amanah terhadap orang yang telah memberikan suatu kepercayaan. Hal itu mencakup seluruh amanah yang wajib bagi manusia, berupa hak-hak Allah SWT terhadap hamba-Nya, seperti shalat, zakat, puasa, kafarat, nadzar dan selain dari itu, yang kesemuanya adalah amanah yang diberikan tanpa pengawasan hamba-Nya yang

lain. Serta amanah yang berupa hak-hak sebagian hamba dengan hamba lainnya yang kesemuanya adalah amanah yang dilakukan tanpa pengawasan saksi (Katsir, 2015).

Hadits tersebut juga memberikan peringatan untuk tidak berlaku khianat. Namun dalam kenyataanya amanah tersebut tidak selalu terjaga dan disampaikan. Hal ini dapat dilihat dari masih banyaknya kasus penyelewengan, manipulasi dan sebagainya. Terkhusus dalam keamanan jaringan juga banyak terjadi penyadapan yang dilakukan oleh pihak yang tidak bertanggung jawab. Tentu hal tersebut bermula dari pengkhianatan terhadap suatu amanah yang semestinya di jaga.

Khianat merupakan lawan dari amanah dan sifatnya sangat tercela. Sifat khianat menunjukkan sifat dari golongan orang munafiq yang di benci Allah SWT. Oleh sebab itu Allah SWT melarang hambanya untuk berkhianat baik pada Allah SWT, rasulullah dan amanah terhadap mereka sendiri. Berbagai peringatan untuk menghindari segala bentuk pengkhianatan terhadap amanah juga disebutkan pada firman Allah SWT pada surat Al-Anfal ayat 27 sebagai berikut.

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

"Wahai orang-orang yang beriman, janganlah kamu mengkhianati allah dan Rasul serta janganlah kamu mengkhianati amanat yang dipercayakan kepadamu, sedangkan kamu mengetahui." (Lajnah Pentashihan Mushaf Al-Qur'an, 2019)

Pada ayat tersebut Allah SWT memerintahkan kepada seluruh hambanya untuk senantiasa taat dan menjalankan segala perintah-Nya dan memenuhi seruan Rasul. Allah SWT juga kembali memperingatkan untuk tidak berkhianat terhadap amanat yang dipercayakan. Hal ini dapat menjaga hubungan antar sesama hamba dan menjadi dasar dalam menegakkan hukum-hukum Allah SWT. Secara tidak langsung Allah SWT juga mengingatkan hamba-Nya untuk menjaga kepentingan

umum, untuk mencegah berbagai musibah dan kedzaliman yang dapat merusak kepentingan umum.

Mengenai hal tersebut dalam hadits yang diriwayatkan oleh Imam Bukhari, Rasulullah SAW juga bersabda.

عَنْ اَبِي هُرَيْرَةَ رَضِيَ اللهُ عَنْهَ قَالَ: قَالَ رَسُوْلُ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ: اِذَا ضُيِّعَتِ الْاَمَانَةُ فَانْتَظِرِالسَّاعَةَ، كَيْفَ اِضَاعَتُهَا يَارَسُوْلَ اللهِ؟ قَالَ: اِذَا اُسْنِدَ الْاَمْرُ اِلَى غَيْرِ اَهْلِهِ فَانْتَظِرِالسَّاعَةَ.(اَخْرَجَهُ الْبُخَارِيُّ)

"Dari Abu Hurairah R.A. berkata: Rasulullah SAW bersabda: Apabila amanah disia-siakan maka tunggulah saat kehancurannya. Salah seorang sahabat bertanya: Bagaimanakah menyia-nyiakannya, wahai Rasulullah? Rasulullah SAW menjawab: Apabila perkara itu diserahkan kepada orang yang bukan ahlinya, maka tunggulah kehancurannya." (H.R. Imam Bukhari)

Rasulullah SAW menyebutkan tentang akan datangnya kehancuran. Kehancuran yang dimaksud merupakan datangnya hari kiamat jika amanah disia-siakan dengan memberikannya kepada orang yang bukan ahlinya. Dalam kriptografi hal ini berkaitan dengan menjaga keamanan suatu pesan, yang mana jika pesan tersebut tidak aman dan diketahui oleh pihak lain yang tidak semestinya dapat menimbulkan hal yang buruk. Contohnya kebocoran rahasia ataupun isi pesan, meragukan keaslian dari suatu pesan, hingga penolakan terhadap pesan tersebut. Jika hal buruk tersebut terjadi, tentu akan menyebabkan hilangnya kepercayaan terhadap pihak yang diberikan amanah.

Seruan mengenai amanah tidak hanya berkaitan dengan kepercayaan maupun tanggung jawab. Namun amanah juga berkaitan dengan keimanan seseorang. Sebagaimana hadits yang diriwayatkan Imam Ahmad berikut.

حَدَّثَنَا عَبْدُالله، حَدَّثَنِيْ اَبِي، حَدَّثَنَا بهز، حَدَّثَنَا اَبُوْ هِلَال، حَدَّثَنَا قَتَادَة، عَنْ اَنَسِ بْنُ مَالِكٍ قَالَ: مَا خَطَبْنَا نَبِيُّ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ، اِلَّا قَالَ: لاَ اِيْمَانَ لِمَنْ لَا اَمَانَةَ لَهُ وَلَا دِيْنَ لِمَنْ لَا عَهِدَ لَهُ.(رواه احمد)

"Tidaklah Nabi SAW berkhutbah kepada kami, melainkan beliau bersabda: Tidak sempurna iman seseorang yang tidak amanah, dan tidak sempurna agama orang yang tidak menunaikan janji." (H.R. Ahmad)

Tujuan Allah SWT dalam memberikan amanah kepada manusia, tidak lain yaitu sebagai tolak ukur keimanan dan derajat manusia. Manusia yang dapat menunaikan amanah memiliki derajat lebih tinggi dibandingkan dengan makhluk lain termasuk malaikat. Apabila tidak dapat menunaikan suatu amanah, maka manusia tersebut dianggap memiliki iman yang lemah serta derajat yang lebih rendah dibandingkan binatang. Begitu pula algoritma kriptografi yang aman tentu juga memiliki nilai unggul dan akan banyak digunakan karena jaminan pesan yang dikirimkan aman dan terpercaya. Namun jika suatu algoritma ternyata tidak mampu dalam menjaga keamanan pesan, algoritma tersebut akan dianggap rentan atau tidak aman dan jika tetap digunakan akan menyebabkan suatu kebocoran pesan.

Kemudian Allah SWT berfirman pada surat al-Mu'minun ayat 8 tentang bagaimana hamba-Nya dapat memperoleh keberuntungan.

وَالَّذِيْنَ هُمْ لِاَمٰنٰتِهِمْ وَعَهْدِهِمْ رَاعُوْنَ﴿٨﴾

"(Sungguh beruntung pula) orang-orang yang memelihara amanat dan janji mereka." (Lajnah Pentashihan Mushaf Al-Qur'an, 2019)

Sebelumnya Allah SWT memerintahkan hamba-Nya untuk senantiasa memelihara amanah karena betapa besarnya kebahagiaan orang-orang yang taat kepada Allah SWT dan Rosul-Nya serta melaksanakan hukum-hukum dan

syariatnya. Allah SWT menerangkan bahwa kebahagiaan itu diperoleh dengan menunaikan amanah Allah SWT secara ikhlas. Pada ayat di atas, Allah SWT mensifati mereka orang-orang yang memelihara amanat dan janji dengan sifat terpuji dan perbuatan yang mulia. Yakni, jika mereka diberi kepercayaan, maka mereka tidak akan mengkhianatinya tetapi mereka menunaikannya kepada yang berhak. Dan jika mereka berjanji atau melakukan akad perjanjian, maka mereka menepatinya, tidak seperti sifat-sifat orang munafik (Katsir, 2015).

2.17 Kajian Topik dan Teori Pendukung

Keamanan merupakan aspek yang sangat penting untuk diperhatikan, terutama keamanan pada pesan. Dalam hal tersebut kriptografi dapat melakukannya menggunakan bebagai algoritma sesuai dengan yang dibutuhkan. Pada mulanya, kriptografi dalam mengamankan pesan hanya sebatas melakukan penggantian karakter pesan asli dengan karakter lain yang dikenal sebagai metode substitusi. Namun seiring berkembangnya teknologi, cara tersebut dirasa kurang aman karena sangat mudah dipecahkan dengan frekuensi kemunculan huruf.

Kemudian kriptografi mengalami perkembangan di mana algoritma yang digunakan lebih kompleks dan dioperasikan menggunakan komputer atau biasa disebut algoritma kriptografi modern. Algoritma ini lebih aman jika dibandingkan dengan algoritma sebelumnya. Namun kemunculan komputer kuantum membuat algoritma kriptografi modern menjadi rentan karena dapat dengan mudah dalam memecahkan algoritmanya. Kriptografi pun terus dikembangkan supaya memiliki keamanan yang tahan terhadap berbagai jenis serangan. Sehingga dikembangkan *Post-quantum cryptography* (PQC) yang salah satu pendekatannya *code-based*

cryptography. *Code-based cryptography* yang tetap aman yaitu kriptosistem *McEliece*.

Kriptosistem *McEliece* merupakan algoritma berbasis kode pengoreksi kesalahan. Sehingga diperlukan algoritma kode pengoreksi kesalahan pada penerapannya, yaitu kode *Reed-Muller* yang biasa melakukan *encoding* dan *decoding* pada pesan. Kriptosistem *McEliece* yang menggunakan kode *Reed-Muller* melakukan enkripsi dan dekripsi berdasarkan algoritma *McEliece*. Pada proses enkripsi dan dekripsi tentu diperlukan dua jenis kunci yaitu kunci publik dan kunci privat. Kunci tersebut dibangkitkan menggunakan kode *Reed-Muller*. Adapun pada proses dekripsi kriptosistem *McEliece* untuk mengembalikan pesan ke bentuk semula terdapat bagian di mana akan menerapkan proses *decoding* kode *Reed-Muller*.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian ini menggunakan metode penelitian kualitatif. Metode tersebut merupakan metode penelitian yang menekankan pada analisis atau deskripsi permasalahan dan menjadikannya dasar dalam mengambil data. Metode ini bertujuan mengkaji permasalahan secara mendalam untuk menemukan ide baru bahkan teori baru. Dalam proses penelitian kualitatif ditekankan perspektif tematik dan berpedoman pada landasan teori agar proses penelitian sesuai dengan fakta yang ditemukan selama penelitian.

3.2 Pra Penelitian

Pada pra penelitian, dilakukan studi kepustakaan (*library research*) yaitu mengumpulkan dan mengkaji berbagai literatur berupa buku, artikel, jurnal dan referensi lain yang relevan dengan permasalahan dan tujuan penelitian. Dalam hal ini peneliti mengawali dengan mengkaji berbagai literatur yang dijadikan sebagai landasan penelitian. Kemudian mengkaji berbagai wawasan dan teori yang berkaitan tentang kriptografi, kriptosistem *McEliece*, kode pengoreksi kesalahan dan kode *Reed-Muller* untuk dijadikan bahan pembahasan dalam penelitian. Selain itu, juga dilakukan simulasi yang bertujuan untuk melakukan eksperimen awal yang dapat memberikan gambaran tentang bagaimana kriptosistem *McEliece* dengan kode *Reed-Muller* akan bekerja dalam praktik.

3.3 Tahapan Penelitian

Berdasarkan rumusan permasalahan dan tujuan yang telah dirumuskan dalam penelitian ini, tahapan penilitian yang dilakukan adalah sebagai berikut:

1. Proses dan simulasi pembangkitan kunci untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.
    a. Menentukan parameter kode linier *Reed-Muller* $r = 1,2,3$ dan $m = 5$.
    b. Membuat kunci privat G yaitu matriks generator berdasarkan parameter kode linier yang telah ditentukan.
    c. Membuat kunci privat S berupa matriks *non-singular*.
    d. Memilih sembarang kunci privat P berupa matriks permutasi.
    e. Menghitung kunci publik $G' = SGP$.
2. Implementasi dan simulasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.
    a. Proses enkripsi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.
        1) Menyiapkan pesan p.
        2) Menyiapkan vektor kesalahan e.
        3) Menyiapkan kunci publik G' yang telah dibangkitkan sebelumnya.
        4) Operasikan rumus enkripsi kriptosistem *McEliece*.
    b. Proses dekripsi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*.
        1) Menyiapkan pesan yang diterima yaitu c.
        2) Mencari $pSG + e'$ dengan rumus persamaan dekripsi kriptosistem *McEliece*.

3) Melakukan *decoding* berdasarkan algoritma kode linier yang digunakan untuk mendapatkan pS.
4) Mencari pesan asli p dengan rumus dekripsi kriptosistem *McEliece*.

3. Perbandingan implementasi menggunakan parameter kode $RM(1,5)$, $RM(2,5)$ dan $RM(3,5)$ dilakukan dengan mengidentifikasi hasil simulasi pemrograman menggunakan tiga parameter tersebut.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Proses dan Simulasi Pembangkitan Kunci Untuk Implementasi Algoritma Kriptosistem *McEliece* Dengan Menggunakan Kode *Reed-Muller*

Proses pembangkitan kunci merupakan bagian dari tahapan implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller*. Terdapat dua jenis kunci yang akan dibangkitkan yaitu kunci privat dan kunci publik. Tahapan ini dilakukan pihak penerima, sehingga hanya penerima yang mengetahui kunci privat yang digunakan. Proses pembangkitan kunci diawali dengan menentukan parameter kode linier yang digunakan. Pada penelitian ini ditentukan parameter kode linier berdasarkan kode *Reed-Muller* yaitu $G(r,m)$, di mana r dan m berupa bilangan asli yang mencakup nol dengan $r \leq m$. Adapun parameter kode *Reed-Muller* tersebut juga digunakan untuk mengetahui parameter sistem kriptografi *McEliece* yaitu n dan k, di mana $n = 2^m$ dan $k = \sum_{i=0}^{r} \binom{m}{i}$.

Selanjutnya yaitu membuat kunci privat G berupa matriks generator berukuran $k \times n$ berdasarkan parameter kode *Reed-Muller* yaitu $G(r,m)$. Kemudian untuk kunci privat S yang merupakan matriks *non-singular* berukuran $k \times k$, diperoleh dengan memilih sembarang matriks $k \times k$ kemudian mencari determinannya untuk membuktikan sifat matriks *non-singular* tersebut. Kemudian untuk kunci privat P diperoleh dengan memilih sembarang matriks permutasi berukuran $n \times n$. Terakhir yaitu membangkitkan kunci publik G' dengan menghitung $G' = SGP$ yang akan menghasilkan matriks berukuran $k \times n$. Adapun

simulasi pembangkitan kunci untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* yaitu sebagaimana berikut.

Pertama, menentukan parameter kode *Reed-Muller* $G(r,m)$ yaitu $r=1$ dan $m=5$. Dari parameter kode *Reed-Muller* tersebut dapat digunakan untuk mengetahui parameter sistem kriptografi *McEliece* n dan k sebagai berikut.

$$n = 2^m = 2^5 = 32$$

$$k = \sum_{i=0}^{r} \binom{m}{i} = \sum_{i=0}^{1} \binom{5}{i} = \binom{5}{0} + \binom{5}{1} = \frac{5!}{0!\,(5-0)!} + \frac{5!}{1!\,(5-1)!} = 1 + 5 = 6$$

Selanjutnya membuat kunci privat G yang berupa matriks generator $G(1,5)$ dengan mengikuti definisi sifat rekursif.

$$G(1,5) = \begin{bmatrix} G(1,4) & G(1,4) \\ 0 & G(0,4) \end{bmatrix}$$

Kemudian didefinisikan $G(1,4)$ dan $G(0,4)$,

$$G(1,4) = \begin{bmatrix} G(1,3) & G(1,3) \\ 0 & G(0,3) \end{bmatrix}$$

$$G(0,4) = [G(0,3) \quad G(0,3)]$$

sehingga didapatkan definisi $G(1,5)$ sementara yaitu,

$$G(1,5) = \begin{bmatrix} G(1,3) & G(1,3) & G(1,3) & G(1,3) \\ 0 & G(0,3) & 0 & G(0,3) \\ 0 & 0 & G(0,3) & G(0,3) \end{bmatrix}.$$

Kemudian telah didefinisikan $G(1,3)$ pada persamaan 2.10 dan akan didefinisikan $G(0,3)$ berdasarkan definisi sifat rekursif pada persamaan 2.8,

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

$$G(0,3) = [G(0,2) \quad G(0,2)]$$

diperoleh definisi sifat rekursif untuk $G(0,3)$ yaitu entrinya terbentuk dari dua matriks $G(0,2)$. Adapun telah didefinisikan $G(0,2)$ pada persamaan 2.9 sebagai berikut.

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

Sehingga diperoleh suatu matriks generator kunci $G_{6\times32}$ yang terdiri dari 6 vektor baris dan 32 vektor kolom yang setiap entrinya merupakan satu bilangan biner antara 0 atau 1 sebagai berikut.

$$G(1,5) = \begin{bmatrix} 1111111111111111111111111111111 \\ 0101010101010101010101010101010 \\ 0011001100110011001100110011001 \\ 0000111100001111000011110000111 \\ 0000000011111111000000001111111 \\ 0000000000000000111111111111111 \end{bmatrix}$$

Selanjutnya akan di bentuk kunci privat S, di mana S berupa sembarang matriks yang bersifat *non-singular* yaitu matriks dengan determinan tidak sama dengan nol. Misalkan $S = [s_{ij}]_{6\times6}$,

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

akan ditunjukkan determinan matriks S menggunakan ekspansi kofaktor mengikuti persamaan 2.4 dan 2.5 serta persamaan 2.3 untuk determinan matriks berukuran 2×2. Sehingga determinan matriks S didefinisikan sebagai berikut.

$$\det(S) = s_{1j}C_{1j} + s_{2j}C_{2j} + \cdots + s_{nj}C_{nj}$$

$$\det(S) = s_{i1}C_{i1} + s_{i2}C_{i2} + \cdots + s_{in}C_{in}$$

Berikut merupakan perhitungan untuk mencari determinan matriks S.

$$\det(S) = s_{21}C_{21} + s_{22}C_{22} + s_{23}C_{23} + s_{24}C_{24} + s_{25}C_{25} + s_{26}C_{26}$$

$$= 1(-1)^{2+1}M_{21} + 0 + 0 + 1(-1)^{2+4}M_{24} + 0 + 0$$

$$= (-1)\begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{vmatrix} + 1\begin{vmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{vmatrix}$$

$$= \big(-(s_{15}C_{15} + s_{25}C_{25} + s_{35}C_{35} + s_{45}C_{45} + s_{55}C_{55})\big)$$

$$+ (s_{51}C_{51} + s_{52}C_{52} + s_{53}C_{53} + s_{54}C_{54} + s_{55}C_{55})$$

$$= \big(-(0 + 0 + 1(-1)^{3+5}M_{35} + 0 + 0)\big)$$

$$+ (0 + 1(-1)^{5+2}M_{52} + 0 + 0 + 0)$$

$$= \left(-\left(1\begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{vmatrix}\right)\right) + \left((-1)\begin{vmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{vmatrix}\right)$$

$$= \big(-(s_{11}C_{11} + s_{12}C_{12} + s_{13}C_{13} + s_{14}C_{14})\big)$$

$$+ \big(-(s_{14}C_{14} + s_{24}C_{24} + s_{34}C_{34} + s_{44}C_{44})\big)$$

$$= \big(-(1(-1)^{1+1}M_{11} + 1(-1)^{1+2}M_{12} + 0 + 0)\big)$$

$$+ \big(-(0 + 0 + 1(-1)^{3+4}M_{34} + 0)\big)$$

$$= \left(-\left(1\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} + (-1)\begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{vmatrix}\right)\right) + \left(-\left((-1)\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}\right)\right)$$

$$= \Big(-\big((s_{31}C_{31} + s_{32}C_{32} + s_{33}C_{33}) + \big(-(s_{31}C_{31} + s_{32}C_{32} + s_{33}C_{33})\big)\big)\Big)$$

$$+ (s_{11}C_{11} + s_{21}C_{21} + s_{31}C_{31})$$

$$= \Big(-\big((0 + 1(-1)^{3+2}M_{32} + 0)$$

$$+ \big(-(1(-1)^{3+1}M_{31} + 1(-1)^{3+2}M_{32} + 0)\big)\big)\Big)$$

$$+ (1(-1)^{1+1}M_{1+1} + 0 + 0)$$

$$= \left(-\left(\left((-1) \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} \right) + \left(-\left(1 \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} + (-1) \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} \right) \right) \right) \right.$$

$$\left. + \left(1 \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} \right) \right)$$

$$= \left(-\left((-1)0 \right) + \left(-\left(1 \times 0 + (-1)(-1) \right) \right) \right) + (1 \times 0)$$

$$= \left(-\left(0 + (-1) \right) \right) + 0$$

$$= \left(-(-1) \right)$$

$$= 1$$

Telah diperoleh determinan dari matriks S adalah 1, sehingga dengan kata lain matriks S merupakan matriks *non-singular*.

Selanjutnya untuk kunci privat P diperoleh dengan memilih sembarang matriks permutasi berukuran 32×32 yang terdiri dari 32 vektor baris dan 32 vektor kolom sebagai berikut.

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

Terakhir pembangkitan kunci publik dilakukan dengan mengalikan matriks kunci privat S, G, P. Perkalian tersebut menghasilkan suatu matriks sebagai kunci publik G' yang dapat diketahui oleh semua pihak. Adapun matriks G' berukuran 6×32 yang terdiri dari 6 vektor baris dan 32 vektor kolom.

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

Adapun pembangkitan kunci untuk $r \in \{2, 3\}$ dan $m = 5$ dilakukan dengan bantuan pemrograman, yang hasilnya terdapat pada lampiran 4 dan 5.

4.2 Implementasi dan Simulasi Algoritma Kriptosistem *McEliece* Dengan Menggunakan Kode *Reed-Muller*

Proses implementasi kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* terdiri dari proses enkripsi dan dekripsi pesan. Proses enkripsi dilakukan oleh pengirim dan proses dekripsi dilakukan oleh penerima. Proses enkripsi diawali dengan menyiapkan sebuah pesan dan dikonversi ke bentuk kode biner berdasarkan tabel ASCII 256-bit sehingga diperoleh pesan p berupa vektor kode biner. Kemudian vektor kode biner pesan p di-bagi menjadi blok-blok vektor sepanjang k, yang tiap bloknya akan menjadi vektor kode biner p_i. Selanjutnya membuat vektor kesalahan e dengan panjang n yang memiliki bobot maksimum kesalahan t, dengan terlebih dahulu menghitung $t = \frac{d-1}{2}$ di mana $d = 2^{m-r}$. Kemudian menyiapkan kunci publik G' yang telah dibangkitkan sebelumnya dan menghitung rumus enkripsinya yaitu $c = p_i G' + e$, di mana c ini merupakan suatu pesan yang telah di enkripsi dan akan dikirimkan.

Adapun proses dekripsi diawali dengan menyiapkan kunci privat S, G, P dan pesan yang diterima (c) kemudian membaginya menjadi blok-blok vektor sepanjang n hingga menjadi vektor kode biner c_i. Kemudian menghitung $c_i' = c_i P^{-1}$, di mana terdapat suatu persamaan yaitu

$$c' = cP^{-1} = (pG' + e)P^{-1} = pSG + eP^{-1} = pSG + e'.$$

Selanjutnya melakukan *decoding* berdasarkan kode *Reed-Muller* terhadap $c_i P^{-1} = p_i SG + e'$ untuk mendapatkan $p_i S$. Langkah *decoding*-nya, pertama menyiapkan matriks generator G dan memetakan vektor barisnya ke dalam monomial $x_0, x_1, x_2, \dots, x_i$; kedua menentukan vektor karakteristik dari tiap baris matriks G mulai baris paling bawah dengan mengecualikan baris pertama dan

vektor karakteristik dikali titik dengan $c_i' = c_i P^{-1}$, lalu ambil sebagian besar nilai dari hasil perkalian titik untuk ditetapkan sebagai koefisien baris tersebut; ketiga menghitung M_y, lalu menghitung $M_y + c_i P^{-1}$ dan ambil sebagian besar nilai dari hasilnya untuk ditetapkan sebagai koefisien baris pertama, terakhir membentuk vektor baris dari urutan koefisien yang diperoleh dari tiap baris mulai baris pertama, sehingga diperoleh $p_i S$.

Tahap dekripsi terakhir yaitu mengoperasikan rumus dekripsi kriptosistem *McEliece* tehadap $p_i S$ yaitu mengalikannya dengan S^{-1} untuk mendapatkan pesan p_i berupa vektor kode biner sepanjang k. Kemudian p_i di-bagi menjadi blok-blok vektor kode biner dengan panjang 8-bit agar dapat mengonversikannya berdasarkan ASCII 256-bit yang merepresentasikan huruf atau simbolnya dengan kode biner sebanyak 8-bit. Simulasi untuk proses enkripsi dan dekripsi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* adalah sebagai berikut.

4.2.1 Simulasi Proses Enkripsi Algoritma Kriptosistem *McEliece* Dengan Menggunakan Kode *Reed-Muller*

Pada proses enkripsi pesan yang dilakukan oleh pengirim, terlebih dahulu menyiapkan pesan yang hendak dikirim. Pesannya adalah RUN. Vektor kode biner berdasarkan tabel ASCII 256-bit untuk tiap huruf dari pesan tersebut adalah

$$R = [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 0]$$

$$U = [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$N = [\,0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0]$$

Perhatikan bahwa pesan yang di enkripsi yaitu vektor kode biner sepanjang $k = 6$. Diketahui panjang vektor pesan tersebut 8-bit, tidak sejalan

dengan panjang vektor untuk pesan yang di enkripsi. Sehingga perlu mengubah panjang vektor menjadi 6-bit dengan menggabungkan semua vektor kode biner, kemudian membaginya menjadi blok-blok kode sepanjang 6-bit sebagai berikut.

$$p = [\,0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0]$$

$$p_1 = [\,0\ 1\ 0\ 1\ 0\ 0]$$

$$p_2 = [\,1\ 0\ 0\ 1\ 0\ 1]$$

$$p_3 = [\,0\ 1\ 0\ 1\ 0\ 1]$$

$$p_4 = [\,0\ 0\ 1\ 1\ 1\ 0]$$

Sehingga diperoleh pesan yang akan dienkripsi adalah p_1, p_2, p_3, p_4.

Selanjutnya menyiapkan vektor kesalahan sepanjang n-bit, di mana $n = 32$. Untuk membentuk vektor kesalahan *e* terlebih dahulu akan ditentukan bobot maksimum kesalahan t sebagai berikut

$$t = \frac{d-1}{2} = \frac{16-1}{2} = 7{,}5 \text{ dengan } d = 2^{5-1} = 2^4 = 16$$

Diperoleh bobot maksimum *error* $t = 7{,}5$ sehingga *error* yang dapat dikoreksi mencapai 7-bit. Vektor kesalahan *e* dibentuk dengan membuat sembarang vektor biner berukuran 1×32 yang terdiri dari bit 0, dan bit 1 yang jumlahnya tidak melebihi batas maksimum *error* yang dapat dikoreksi, misalnya sebagai berikut:

$$e = [00000000000000000000000001111111]$$

Lalu operasikan rumus enkripsi, $c = pG' + e$ terhadap p_1, p_2, p_3, p_4.

$c_1 = p_1 G' + e$

$$= [\,0\ 1\ 0\ 1\ 0\ 0] \times \begin{bmatrix} 01101011011101100100100110101000 \\ 01001010110101001001111101010010 \\ 10111101000101000101010101110100 \\ 00011111101001011000110011100001 \\ 00010010111001110110000101110110 \\ 00011010110110000101010010101111 \end{bmatrix}$$

$$+ [00000000000000000000000001111111]$$

$$= [01010101011100010001001110110011]$$

$$+ [00000000000000000000000001111111]$$

$$= [01010101011100010001001111001100]$$

$$c_2 = p_2 G' + e$$

$$= [\,1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\,] \times \begin{bmatrix} 01101011011101100100100110101000 \\ 01001010110101001001111101010010 \\ 10111101000101000101010101110100 \\ 00011111101001011000110011100001 \\ 00010010111001110110000101110110 \\ 00011010110110000101010010101111 \end{bmatrix}$$

$$+ [00000000000000000000000001111111]$$

$$= [01101110000010111001000110011001]$$

$$c_3 = p_3 G' + e$$

$$= [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1\,] \times \begin{bmatrix} 01101011011101100100100110101000 \\ 01001010110101001001111101010010 \\ 10111101000101000101010101110100 \\ 00011111101001011000110011100001 \\ 00010010111001110110000101110110 \\ 00011010110110000101010010101111 \end{bmatrix}$$

$$+ [00000000000000000000000001111111]$$

$$= [01001111101010010100011101100011]$$

$$c_4 = p_4 G' + e$$

$$= [\,0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 0\,] \times \begin{bmatrix} 01101011011101100100100110101000 \\ 01001010110101001001111101010010 \\ 10111101000101000101010101110100 \\ 00011111101001011000110011100001 \\ 00010010111001110110000101110110 \\ 00011010110110000101010010101111 \end{bmatrix}$$

$$+ [00000000000000000000000001111111]$$

$$= [10110000010101101011100010011100]$$

Sehingga diperoleh hasil enkripsi dari masing-masing blok vektor pesan berupa blok vektor baru c berukuran 1×32 yang memiliki entri sebanyak 32 bit yaitu

$$c_1 = [01010101011100010001001111001100]$$

$$c_2 = [01101110000010111001000110011001]$$

$$c_3 = [01001111101010010100011101100011]$$

$$c_4 = [10110000010101101011100010011100]$$

Kemudian untuk mengirimkannya kepada penerima, pengirim menggabungkan hasil enkripsi (c_1, c_2, c_3, c_4) menjadi satu vektor biner c berukuran 1×128 sebagai berikut.

c =[01010101011100010001001111001100011011100000101110010001100110010100111110101001010001110110001110110000010101101011100010011100]

Adapun simulasi proses enkripsi untuk $r \in \{2, 3\}$ dan $m = 5$ dilakukan dengan bantuan pemrograman, yang hasilnya terdapat pada lampiran 3 dan 4.

4.2.2 Simulasi Proses Dekripsi Algoritma Kriptosistem *McEliece* Dengan Menggunakan Kode *Reed-Muller*

Pada proses dekripsi pesan, yang dilakukan penerima terlebih dahulu yaitu menyiapkan pesan yang diterima. Pesan yang diterima yaitu vektor biner c berukuran 1×128 yang merupakan gabungan dari blok-blok vektor kode biner sepanjang $n = 32$, sebagai berikut.

c =[01010101011100010001001111001100011011100000101110010001100110010100111110101001010001110110001110110000010101101011100010011100].

Kemudian membaginya menjadi blok-blok vektor berukuran 1×32 yang mengandung entri sebanyak 32-bit.

$$c_1 = [01010101011100010001001111001100]$$

$$c_2 = [01101110000010111001000110011001]$$

$$c_3 = [01001111101010010100011101100011]$$

$$c_4 = [10110000010101101011100010011100]$$

Ingat bahwa sebelumnya, penerima telah membangkitkan kunci privat S, G, P.

Selanjutnya mengalikan c dengan invers matriks kunci P berukuran 32×32 yang terdiru dari 32 vektor baris dan 32 vektor kolom, mengikuti rumus persamaan $c' = cP^{-1} = pSG + eP^{-1} = pSG + e'$. Sehingga akan diperoleh hasil dari c_1P^{-1}, c_2P^{-1}, c_3P^{-1}, c_4P^{-1} berupa blok-blok vektor berukuran 1×32 yang mengandung entri sebanyak 32-bit.

$c_1P^{-1} = [01010101011100010001001111001100]$

$$\times \begin{bmatrix} 00000000000000000000100000000000 \\ 00000000000000000000000000000010 \\ 00000000000000000000000001000000 \\ 00000000001000000000000000000000 \\ 00000000000000000000000100000 \\ 10000000000000000000000000000000 \\ 00000000000000000000000000000001 \\ 00000000000001000000000000000000 \\ 00000000000010000000000000000000 \\ 00000000000000000010000000000000 \\ 00000000010000000000000000000000 \\ 00000000000000000000001000000000 \\ 00000000000000000100000000000000 \\ 00010000000000000000000000000000 \\ 00100000000000000000000000000000 \\ 01000000000000000000000000000000 \\ 00000001000000000000000000000000 \\ 00000000000000000000000010000 \\ 00001000000000000000000000000000 \\ 00000000000000000010000000000000 \\ 00001000000000000000000000000000 \\ 00000000000010000000000000000000 \\ 00000000001000000000000000000000 \\ 00000000000000000000100000000 \\ 00000000000000000001000000000 \\ 00000000000000000010000000000 \\ 00001000000000000000000000000000 \\ 00000000000000000000000001000 \\ 00000000000001000000000000000010 \\ 00000000000000000000000000000100 \\ 00000010000000000000000000000000 \end{bmatrix}^{-1}$$

$$= [01010101011100010001001111001100]$$

$$\times \begin{bmatrix}
00000100\,00000000\,00000000\,00000000 \\
00000000\,00000000\,10000000\,00000000 \\
00000000\,00000001\,00000000\,00000000 \\
00000000\,00000010\,00000000\,00000000 \\
00000000\,00000000\,00010000\,00000000 \\
00000000\,00000000\,00000100\,00000000 \\
00000000\,00000000\,00000000\,00010000 \\
00000000\,00000000\,00000000\,00000001 \\
00000000\,00000000\,01000000\,00000000 \\
00000000\,00010000\,00000000\,00000000 \\
00000000\,00000000\,00000001\,00000000 \\
00010000\,00000000\,00000000\,00000000 \\
00000000\,00000000\,00000010\,00000000 \\
00000000\,10000000\,00000000\,00000000 \\
00000001\,00000000\,00000000\,00000000 \\
00000000\,00000000\,00000000\,00000100 \\
00000000\,00100000\,00000000\,00000000 \\
00000000\,00000000\,00001000\,00000000 \\
00000000\,00000100\,00000000\,00000000 \\
00000000\,01000000\,00000000\,00000000 \\
10000000\,00000000\,00000000\,00000000 \\
00000000\,00000000\,00000000\,01000000 \\
00000000\,00000000\,00000000\,00100000 \\
00000000\,00000000\,00000000\,10000000 \\
00000000\,00001000\,00000000\,00000000 \\
00100000\,00000000\,00000000\,00000000 \\
00001000\,00000000\,00000000\,00000000 \\
00000000\,00000000\,00100000\,00000000 \\
00000000\,00000000\,00000000\,00001000 \\
00000000\,00000000\,00000000\,00000010 \\
01000000\,00000000\,00000000\,00000000 \\
00000010\,00000000\,00000000\,00000000
\end{bmatrix}$$

$$= [00110000010110101000010110101111]$$

$$c_2 P^{-1} = [00000011011010011011011010011100]$$

$$c_3 P^{-1} = [01101010000000001101111111110101]$$

$$c_4 P^{-1} = [10010101111111100100000000001010]$$

Kemudian akan digunakan algoritma *decoding* kode *Reed-Muller* untuk mendapatkan pS. Langkah pertama, menyiapkan matriks G dan memetakan tiap barisnya ke dalam bentuk monomial berupa kode biner. Sehingga diperoleh monomial x_0, x_1, x_2, x_3, x_4, x_5.

$$G = \begin{bmatrix} 11111111111111111111111111111111 \\ 01010101010101010101010101010101 \\ 00110011001100110011001100110011 \\ 00001111000011110000111100001111 \\ 00000000111111110000000011111111 \\ 00000000000000001111111111111111 \end{bmatrix} \begin{matrix} \rightarrow x_0 (baris\ 1) \\ \rightarrow x_1 (baris\ 2) \\ \rightarrow x_2 (baris\ 3) \\ \rightarrow x_3 (baris\ 4) \\ \rightarrow x_4 (baris\ 5) \\ \rightarrow x_5 (baris\ 6) \end{matrix}$$

Langkah kedua, menentukan vektor karakteristik tiap baris matriks G dengan perkalian bit demi bit monomial, mulai baris paling bawah (kecuali baris 1). Lalu vektor-vektor karakteristik dikali titik dengan $c_i P^{-1}$ sesuai persamaan 2.1 dan ambil sebagian besar nilai hasil kali titik dan tetapkan sebagai koefisien baris tersebut. Vektor karakteristik baris 6 diperoleh dari perkalian monomial x_4, x_3, x_2, x_1 dengan komplemennya mengikuti operasi *bitwise* AND di tabel 2.1.

$$\begin{aligned} x_4 x_3 x_2 x_1 &= [00000000111111110000000011111111] \\ &\quad \times [00001111000011110000111100001111] \\ &\quad \times [00110011001100110011001100110011] \\ &\quad \times [01010101010101010101010101010101] \\ &= [00000000000000010000000000000001] \end{aligned}$$

$$\begin{aligned} x_4 x_3 x_2 \overline{x_1} &= [00000000111111110000000011111111] \\ &\quad \times [00001111000011110000111100001111] \\ &\quad \times [00110011001100110011001100110011] \\ &\quad \times [10101010101010101010101010101010] \\ &= [00000000000000010000000000000010] \end{aligned}$$

$$\begin{aligned} x_4 x_3 \overline{x_2} x_1 &= [00000000111111110000000011111111] \\ &\quad \times [00001111000011110000111100001111] \\ &\quad \times [11001100110011001100110011001100] \\ &\quad \times [01010101010101010101010101010101] \end{aligned}$$

$$= [00000000000001000000000000000100]$$

$$x_4 x_3 \overline{x_2}\ \overline{x_1} = [00000000111111110000000011111111]$$

$$\times [00001111000011110000111100001111]$$

$$\times [11001100110011001100110011001100]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000001000000000000001000]$$

$$x_4 \overline{x_3} x_2 x_1 = [00000000111111110000000011111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000010000000000000010000]$$

$$x_4 \overline{x_3} x_2 \overline{x_1} = [00000000111111110000000011111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000001000000000000000100000]$$

$$x_4 \overline{x_3}\ \overline{x_2} x_1 = [00000000111111110000000011111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000010000000000000001000000]$$

$$x_4\overline{x_3}\,\overline{x_2}\,\overline{x_1} = [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [11001100110011001100110011001100]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000100000000000000010000000]$$

$$\overline{x_4}x_3x_2x_1 = [11111111000000001111111100000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000100000000000000100000000]$$

$$\overline{x_4}x_3x_2\overline{x_1} = [11111111000000001111111100000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000010000000000000001000000000]$$

$$\overline{x_4}x_3\overline{x_2}x_1 = [11111111000000001111111100000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000100000000000000010000000000]$$

$$\overline{x_4}x_3\overline{x_2}\,\overline{x_1} = [11111111000000001111111100000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [10101010101010101010101010101010]$$

$$= [00001000000000000000100000000000]$$

$$\overline{x_4}\,\overline{x_3}x_2x_1 = [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [01010101010101010101010101010101]$$

$$= [00010000000000000001000000000000]$$

$$\overline{x_4}\,\overline{x_3}x_2\overline{x_1} = [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [10101010101010101010101010101010]$$

$$= [00100000000000000010000000000000]$$

$$\overline{x_4}\,\overline{x_3}\,\overline{x_2}x_1 = [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [01010101010101010101010101010101]$$

$$= [01000000000000000100000000000000]$$

$$\overline{x_4}\,\overline{x_3}\,\overline{x_2}\,\overline{x_1} = [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [10101010101010101010101010101010]$$

$$= [10000000000000001000000000000000]$$

Kemudian setiap vektor karakteristik baris 6 dikalikan titik dengan c_1P^{-1}.

$$[00000000000000010000000000000001]$$

$$\cdot [00110000101101010000101101011111] = 1$$

$$[00000000000000100000000000000010] \cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000100000000000000100] \cdot [00110000101101010000101101011111] = 1$$

$$[00000000000001000000000000001000] \cdot [00110000101101010000101101011111] = 0$$

$$[00000000000010000000000000010000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000000001000000000000000100000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000000010000000000000001000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000001000000000000000010000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000010000000000000000100000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000100000000000000001000000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00001000000000000000010000000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00010000000000000000100000000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00100000000000000001000000000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00100000000000000010000000000000]$$
$$\cdot [00110000010110101000010110101111] = 1$$
$$[01000000000000000100000000000000]$$
$$\cdot [00110000010110101000010110101111] = 0$$
$$[10000000000000001000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 6 untuk $c_1 P^{-1}$ adalah 1. Vektor karakteristik baris 5 diperoleh dari perkalian monomial x_5, x_3, x_2, x_1 dengan komplemennya mengikuti operasi *bitwise* AND di tabel 2.1.

$$x_5 x_3 x_2 x_1 = [00000000000000001111111111111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000000000000000000100000001]$$
$$x_5 x_3 x_2 \overline{x_1} = [00000000000000001111111111111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000000000000000100000010]$$
$$x_5 x_3 \overline{x_2} x_1 = [00000000000000001111111111111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000000010000000100]$$

$$x_5 x_3 \overline{x_2}\ \overline{x_1} = [00000000000000001111111111111111]$$

$$\times [00001111000011110000111100001111]$$

$$\times [11001100110011001100110011001100]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000000000000100000001000]$$

$$x_5 \overline{x_3} x_2 x_1 = [00000000000000001111111111111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000001000000010000]$$

$$x_5 \overline{x_3} x_2 \overline{x_1} = [00000000000000001111111111111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000000000010000000100000]$$

$$x_5 \overline{x_3}\ \overline{x_2} x_1 = [00000000000000001111111111111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000100000001000000]$$

$$x_5\overline{x_3}\,\overline{x_2}\,\overline{x_1} = [00000000000000001111111111111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [11001100110011001100110011001100]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000000001000000010000000]$$

$$\overline{x_5}x_3x_2x_1 = [11111111111111110000000000000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000001000000010000000000000000]$$

$$\overline{x_5}x_3x_2\overline{x_1} = [11111111111111110000000000000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000001000000010000000000000000]$$

$$\overline{x_5}x_3\overline{x_2}x_1 = [11111111111111110000000000000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000100000001000000000000000000]$$

$$\overline{x_5}x_3\overline{x_2}\,\overline{x_1} = [11111111111111110000000000000000]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [10101010101010101010101010101010]$$

$$= [00001000000010000000000000000000]$$

$$\overline{x_5}\,\overline{x_3}x_2x_1 = [11111111111111110000000000000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [01010101010101010101010101010101]$$

$$= [00010000000100000000000000000000]$$

$$\overline{x_5}\,\overline{x_3}x_2\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [10101010101010101010101010101010]$$

$$= [00100000001000000000000000000000]$$

$$\overline{x_5}\,\overline{x_3}\,\overline{x_2}x_1 = [11111111111111110000000000000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [01010101010101010101010101010101]$$

$$= [01000000010000000000000000000000]$$

$$\overline{x_5}\,\overline{x_3}\,\overline{x_2}\,\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [10101010101010101010101010101010]$$

$$= [10000000100000000000000000000000]$$

Kemudian setiap vektor karakteristik baris 5 dikalikan titik dengan c_1P^{-1}.

$$[00000000000000000000000100000001]$$

$$\cdot [00110000010110101000010110101111] = 0$$

$$[00000000000000000000001000000010] \cdot [00110000101101010000101101011111] = 1$$

$$[00000000000000000000010000000100] \cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000000100000001000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000000000000000001000000010000] \cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000010000000100000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000000000000000100000001000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000001000000010000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00000001000000010000000000000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00000010000000100000000000000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00000100000001000000000000000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00001000000010000000000000000000] \cdot [00110000101101010000101101011111] = 1$$

$$[00010000000100000000000000000000] \cdot [00110000101101010000101101011111] = 0$$

$$[00100000001000000000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 1$$
$$[01000000010000000000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 1$$
$$[10000000100000000000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 5 untuk $c_1 P^{-1}$ adalah 0. Vektor karakteristik baris 4 diperoleh dari perkalian monomial x_5, x_4, x_2, x_1 dengan komplemennya mengikuti operasi *bitwise* AND di tabel 2.1.

$$x_5 x_4 x_2 x_1 = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000000000000000000000010001]$$
$$x_5 x_4 x_2 \overline{x_1} = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000000000000000000100010]$$
$$x_5 x_4 \overline{x_2} x_1 = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000000000001000100]$$

$$x_5 x_4 \overline{x_2}\ \overline{x_1} = [00000000000000001111111111111111]$$

$$\times [00000000111111110000000011111111]$$

$$\times [11001100110011001100110011001100]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000000000000000010001000]$$

$$x_5 \overline{x_4} x_2 x_1 = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000001000100000000]$$

$$x_5 \overline{x_4} x_2 \overline{x_1} = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000000000001000100000000]$$

$$x_5 \overline{x_4}\ \overline{x_2} x_1 = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000100010000000000]$$

$$x_5\overline{x_4}\,\overline{x_2}\,\overline{x_1} = [00000000000000001111111111111111]$$
$$\times [11111111000000001111111100000000]$$
$$\times [11001100110011001100110011001100]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000000001000100000000000]$$

$$\overline{x_5}x_4x_2x_1 = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000001000100000000000000000]$$

$$\overline{x_5}x_4x_2\overline{x_1} = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00110011001100110011001100110011]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000001000100000000000000000]$$

$$\overline{x_5}x_4\overline{x_2}x_1 = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000001000100000000000000000]$$

$$\overline{x_5}x_4\overline{x_2}\,\overline{x_1} = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11001100110011001100110011001100]$$
$$\times [10101010101010101010101010101010]$$

$$= [00000000100010000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}x_2x_1 = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [01010101010101010101010101010101]$$

$$= [00010001000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}x_2\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00110011001100110011001100110011]$$

$$\times [10101010101010101010101010101010]$$

$$= [00100010000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}\,\overline{x_2}x_1 = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [01010101010101010101010101010101]$$

$$= [01000100000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}\,\overline{x_2}\,\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11001100110011001100110011001100]$$

$$\times [10101010101010101010101010101010]$$

$$= [10001000000000000000000000000000]$$

Kemudian setiap vektor karakteristik baris 4 dikalikan titik dengan c_1P^{-1}.

$$[00000000000000000000000000010001]$$

$$\cdot [00110000010110101000010110101111] = 1$$

$$[00000000000000000000000000100010] \cdot [00110000010110101000010110101111] = 0$$

$$[00000000000000000000000001000100] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000000000000000000010001000] \cdot [00110000010110101000010110101111] = 0$$

$$[00000000000000000000001000100000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000000000000000010001000000000] \cdot [00110000010110101000010110101111] = 0$$

$$[00000000000000000001000100000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000000000000010001000000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000001000100000000000000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000001000100000000000000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000010001000000010000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00000000100010000000000000000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00010001000000000000000000000000000] \cdot [00110000010110101000010110101111] = 1$$

$$[00100010000000000000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 1$$
$$[01000100000000000000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 0$$
$$[10001000000000000000000000000000]$$
$$\cdot [00110000010110101000010110101111] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 4 untuk $c_1 P^{-1}$ adalah 1. Vektor karakteristik baris 3 diperoleh dari perkalian monomial x_5, x_4, x_3, x_1 dengan komplemennya mengikuti operasi *bitwise* AND di tabel 2.1.

$$x_5 x_4 x_3 x_1 = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000000000000000000000000101]$$
$$x_5 x_4 x_3 \overline{x_1} = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000000000000000000001010]$$
$$x_5 x_4 \overline{x_3} x_1 = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000000000001010000]$$

$$x_5 x_4 \overline{x_3}\ \overline{x_1} = [00000000000000001111111111111111]$$

$$\times [00000000111111110000000011111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000000000000000010100000]$$

$$x_5 \overline{x_4} x_3 x_1 = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000000010100000000]$$

$$x_5 \overline{x_4} x_3 \overline{x_1} = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000000000000000000101000000000]$$

$$x_5 \overline{x_4}\ \overline{x_3} x_1 = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000000000000000101000000000000]$$

$$x_5\overline{x_4}\,\overline{x_3}\,\overline{x_1} = [00000000000000001111111111111111]$$
$$\times [11111111000000001111111100000000]$$
$$\times [11110000111100001111000011110000]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000000001010000000000000]$$

$$\overline{x_5}x_4x_3x_1 = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000000001010000000000000000]$$

$$\overline{x_5}x_4x_3\overline{x_1} = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [10101010101010101010101010101010]$$
$$= [00000000000001010000000000000000]$$

$$\overline{x_5}x_4\overline{x_3}x_1 = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [01010101010101010101010101010101]$$
$$= [00000000010100000000000000000000]$$

$$\overline{x_5}x_4\overline{x_3}\,\overline{x_1} = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [10101010101010101010101010101010]$$

$$= [00000000101000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}x_3x_1 = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [01010101010101010101010101010101]$$

$$= [00000101000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}x_3\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [10101010101010101010101010101010]$$

$$= [00000101000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}\,\overline{x_3}x_1 = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [01010101010101010101010101010101]$$

$$= [01010000000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}\,\overline{x_3}\,\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [10101010101010101010101010101010]$$

$$= [10100000000000000000000000000000]$$

Kemudian setiap vektor karakteristik baris 3 dikalikan titik dengan c_1P^{-1}.

$$[00000000000000000000000000000101]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000000000000001010]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000000000001010000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000000000010100000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000000010100000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000000101000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000000101000000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000000001010000000000000]$$

$$\cdot [00110000101101010000101101011111] = 1$$

$$[00000000000001010000000000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000000101000000000000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000000101000000000000000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000001010000000000000000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00000101000000000000000000000000]$$

$$\cdot [00110000101101010000101101011111] = 0$$

$$[00001010000000000000000000000000]$$
$$\cdot [00110000101101010000010110101111] = 0$$
$$[01010000000000000000000000000000]$$
$$\cdot [00110000101101010000010110101111] = 1$$
$$[10100000000000000000000000000000]$$
$$\cdot [00110000101101010000010110101111] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 3 untuk $c_1 P^{-1}$ adalah 0. Vektor karakteristik baris 2 diperoleh dari perkalian monomial x_5, x_4, x_3, x_2 dengan komplemennya mengikuti operasi *bitwise* AND di tabel 2.1.

$$x_5 x_4 x_3 x_2 = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$= [00000000000000000000000000000011]$$
$$x_5 x_4 x_3 \overline{x_2} = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$= [00000000000000000000000000001100]$$
$$x_5 x_4 \overline{x_3} x_2 = [00000000000000001111111111111111]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [00110011001100110011001100110011]$$

$$= [00000000000000000000000000110000]$$

$$x_5 x_4 \overline{x_3}\ \overline{x_2} = [00000000000000001111111111111111]$$

$$\times [00000000111111110000000011111111]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$= [00000000000000000000000011000000]$$

$$x_5 \overline{x_4} x_3 x_2 = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [00110011001100110011001100110011]$$

$$= [00000000000000000000001100000000]$$

$$x_5 \overline{x_4} x_3 \overline{x_2} = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [11001100110011001100110011001100]$$

$$= [00000000000000000000110000000000]$$

$$x_5 \overline{x_4}\ \overline{x_3} x_2 = [00000000000000001111111111111111]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$= [00000000000000000011000000000000]$$

$$x_5\overline{x_4}\,\overline{x_3}\,\overline{x_2} = [00000000000000001111111111111111]$$
$$\times [11111111000000001111111100000000]$$
$$\times [11110000111100001111000011110000]$$
$$\times [11001100110011001100110011001100]$$
$$= [00000000000000001100000000000000]$$

$$\overline{x_5}x_4x_3x_2 = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [00110011001100110011001100110011]$$
$$= [00000000000000110000000000000000]$$

$$\overline{x_5}x_4x_3\overline{x_2} = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [00001111000011110000111100001111]$$
$$\times [11001100110011001100110011001100]$$
$$= [00000000000011000000000000000000]$$

$$\overline{x_5}x_4\overline{x_3}x_2 = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [00110011001100110011001100110011]$$
$$= [00000000001100000000000000000000]$$

$$\overline{x_5}x_4\overline{x_3}\,\overline{x_2} = [11111111111111110000000000000000]$$
$$\times [00000000111111110000000011111111]$$
$$\times [11110000111100001111000011110000]$$
$$\times [11001100110011001100110011001100]$$

$$= [00000000110000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}x_3x_2 = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [00110011001100110011001100110011]$$

$$= [00000011000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}x_3\overline{x_2} = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [00001111000011110000111100001111]$$

$$\times [11001100110011001100110011001100]$$

$$= [00001100000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}\,\overline{x_3}x_2 = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [00110011001100110011001100110011]$$

$$= [00110000000000000000000000000000]$$

$$\overline{x_5}\,\overline{x_4}\,\overline{x_2}\,\overline{x_1} = [11111111111111110000000000000000]$$

$$\times [11111111000000001111111100000000]$$

$$\times [11110000111100001111000011110000]$$

$$\times [11001100110011001100110011001100]$$

$$= [11000000000000000000000000000000]$$

Kemudian setiap vektor karakteristik baris 2 dikalikan titik dengan c_1P^{-1}.

$$[00000000000000000000000000000011]$$

$$\cdot [00110000010110101000010110101111] = 0$$

[00000000000000000000000000001100]

$\cdot$ [00110000101101010000101101011111] = 0

[00000000000000000000000000110000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000000000000000000011000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000000000000000001100000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000000000000000110000000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000000000000011000000000000]

$\cdot$ [00110000101101010000101101011111] = 0

[00000000000000001100000000000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000000000110000000000000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000000001100000000000000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000001100000000000000000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000000110000000000000000000000]

$\cdot$ [00110000101101010000101101011111] = 1

[00000011000000000000000000000000]

$\cdot$ [00110000101101010000101101011111] = 0

$$[00001100000000000000000000000000] \cdot [00110000010110101000010110101111] = 0$$

$$[00110000000000000000000000000000] \cdot [00110000010110101000010110101111] = 0$$

$$[11000000000000000000000000000000] \cdot [00110000010110101000010110101111] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 2 untuk c_1P^{-1} adalah 1.

Langkah ketiga, menghitung M_y dan menambahkannya dengan c_iP^{-1}, ambil sebagian besar nilai dari hasil $M_y + c_iP^{-1}$ tersebut dan tetapkan sebagai koefisien baris 1. Kemudian membentuk vektor dari urutan koefisien yang telah diperoleh.

$$M_y = \sum koefisien\ baris\ x_i\ (vektor\ baris\ x_i)$$

$$\begin{aligned} M_y = {} & 1[00000000000000001111111111111111] \\ & + 0[00000000111111110000000011111111] \\ & + 1[00001111000011110000111100001111] \\ & + 0[00110011001100110011001100110011] \\ & + 1[01010101010101010101010101010101] \\ = {} & [00000000000000001111111111111111] \\ & + [00001111000011110000111100001111] \\ & + [01010101010101010101010101010101] \\ = {} & [01011010010110101010010110100101] \end{aligned}$$

$$M_y + c_1P^{-1} = [01011010010110101010010110100101]$$
$$+ [00110000010110101000010110101111]$$
$$= [01101010000000000010000000001010]$$

Dapat diketahui hasil dari $M_y + c_1P^{-1}$, jumlah bilangan 0 lebih banyak dari 1, sehingga koefisien baris 1 untuk c_1P^{-1} adalah 0. Kemudian membentuk vektor baru dari semua koefisien yang diperoleh mulai dari koefisien baris 1 sampai 6. Sehingga diperoleh hasil *decoding* c_1P^{-1} adalah vektor $p_1S = [\ 0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1]$.

Kemudian mengulangi langkah-langkah algoritma *decoding* untuk $c_2P^{-1}, c_3P^{-1}, c_4P^{-1}$. Adapun vektor karakteristiknya sama dengan c_1P^{-1}, sehingga tidak perlu menentukan vektor karakteristik lagi dan dapat langsung melakukan perkalian titik untuk setiap vektor karakteristik dengan $c_2P^{-1}, c_3P^{-1}, c_4P^{-1}$.

Berikut perkalian titik setiap vektor karakteristik baris 6 dengan c_2P^{-1}.

$$[00000000000000010000000000000001]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000000000010000000000000010]$$
$$\cdot [00000011011010011011011010011100] = 0$$
$$[00000000000000010000000000000100]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000000001000000000000001000]$$
$$\cdot [00000011011010011011011010011100] = 0$$
$$[00000000000010000000000000010000]$$
$$\cdot [00000011011010011011011010011100] = 1$$

$$[00000000001000000000000000100000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000010000000000000001000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000001000000000000000010000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000010000000000000000100000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000100000000000000001000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00001000000000000000010000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00010000000000000000100000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00100000000000000001000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[01000000000000000010000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[10000000000000000100000000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000011011010011011011010011100] \cdot [00000011011010011011011010011100] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 6 untuk c_2P^{-1} adalah 1.

Kemudian setiap vektor karakteristik baris 5 dikalikan titik dengan $c_2 P^{-1}$.

$$[00000000000000000000000100000001] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000001000000010] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000010000000100] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000100000001000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000001000000010000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000010000000100000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000100000001000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000001000000010000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000001000000010000000000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000010000000100000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000100000001000000000000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00001000000010000000000000000000]$$

$$\cdot [00000011011010011011011010011100] = 1$$

$$[00010000000100000000000000000000]$$

$$\cdot [00000011011010011011011010011100] = 0$$

$$[00100000001000000000000000000000]$$

$$\cdot [00000011011010011011011010011100] = 1$$

$$[01000000010000000000000000000000]$$

$$\cdot [00000011011010011011011010011100] = 1$$

$$[10000000100000000000000000000000]$$

$$\cdot [00000011011010011011011010011100] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 5 untuk c_2P^{-1} adalah 0. Kemudian setiap vektor karakteristik baris 4 dikalikan titik dengan c_2P^{-1}.

$$[00000000000000000000000000010001]$$

$$\cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000000100010]$$

$$\cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000000001000100]$$

$$\cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000010001000]$$

$$\cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000010001000000000]$$

$$\cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000010001000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000100010000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000001000100000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000001000100000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000010001000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000100010000000010000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000001000100000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00010001000000000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00100010000000000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[01000100000000000000000000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[10001000000000000000000000000000] \cdot [00000011011010011011011010011100] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 4 untuk c_2P^{-1} adalah 1.

Kemudian setiap vektor karakteristik baris 3 dikalikan titik dengan $c_2 P^{-1}$.

$$[00000000000000000000000000000101] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000000001010] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000000010000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000000100000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000010100000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000101000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000001010000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000010100000000] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000101000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000001010000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000010100000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000101000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000101000000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000101000000000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00001010000000000000000000000000] \cdot [00000011011010011011011010011100] = 1$$

$$[01010000000000000000000000000000] \cdot [00000011011010011011011010011100] = 0$$

$$[10100000000000000000000000000000] \cdot [00000011011010011011011010011100] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 3 untuk c_2P^{-1} adalah 1. Kemudian setiap vektor karakteristik baris 2 dikalikan titik dengan c_2P^{-1}.

$$[00000000000000000000000000000011] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000000000001100] \cdot [00000011011010011011011010011100] = 0$$

$$[00000000000000000000000000110000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000000011000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000001100000000] \cdot [00000011011010011011011010011100] = 1$$

$$[00000000000000000000110000000000]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000000000000011000000000000]$$
$$\cdot [00000011011010011011011010011100] = 0$$
$$[00000000000000001100000000000000]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000000001100000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000000110000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000001100000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000000110000000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 1$$
$$[00000011000000000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 0$$
$$[00001100000000000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 0$$
$$[00110000000000000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 0$$
$$[11000000000000000000000000000000]$$
$$\cdot [00000011011010011011011010011100] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 2 untuk c_2P^{-1} adalah 1.

Kemudian menghitung M_y dan menambahkannya dengan c_2P^{-1},

$$M_y = \sum koefisien\ baris\ x_i\ (vektor\ baris\ x_i)$$

$$\begin{aligned}
M_y = 1[&00000000000000001111111111111111] \\
&+ 0[00000000111111110000000011111111] \\
&+ 1[00001111000011110000111100001111] \\
&+ 1[00110011001100110011001100110011] \\
&+ 1[01010101010101010101010101010101] \\
= [&00000000000000001111111111111111] \\
&+ [00001111000011110000111100001111] \\
&+ [00110011001100110011001100110011] \\
&+ [01010101010101010101010101010101] \\
= [&01101001011010011001011010010110]
\end{aligned}$$

$$\begin{aligned}
M_y + c_2P^{-1} = [&01101001011010011001011010010110] \\
&+ [00000011011010011011011010011100] \\
= [&01101010000000000010000000001010]
\end{aligned}$$

Dapat diketahui hasil dari $M_y + c_2P^{-1}$, jumlah bilangan 0 lebih banyak dari 1, sehingga koefisien baris 1 untuk c_2P^{-1} adalah 0. Kemudian membentuk vektor baru dari semua koefisien yang diperoleh mulai dari koefisien baris 1 sampai 6. Sehingga diperoleh hasil *decoding* c_2P^{-1} adalah vektor $p_2S = [\,0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1]$.

Berikut perkalian titik setiap vektor karakteristik baris 6 dengan c_3P^{-1}.

$$\begin{aligned}
[&00000000000000001000000000000001] \\
&\cdot [01101010000000001101111111110101] = 1
\end{aligned}$$

$$[00000000000000100000000000000010] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000001000000000000000100] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000000010000000000000001000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000001000000000000000010000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000010000000000000000100000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000100000000000000001000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000001000000000000000010000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000010000000000000000100000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000100000000000000001000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00001000000000000000010000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00010000000000000000100000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00100000000000000001000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00100000000000000010000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[01000000000000000100000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[10000000000000001000000000000000] \cdot [01101010000000001101111111110101] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 6 untuk c_3P^{-1} adalah 1. Kemudian setiap vektor karakteristik baris 5 dikalikan titik dengan c_3P^{-1}.

$$[00000000000000000000000100000001] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000001000000010] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000000010000000100] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000100000001000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000001000000010000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000010000000100000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000100000001000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000001000000010000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000001000000010000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000010000000100000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000100000001000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00001000000010000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00010000000100000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00100000001000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[01000000010000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[10000000100000000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 5 untuk c_3P^{-1} adalah 0. Kemudian setiap vektor karakteristik baris 4 dikalikan titik dengan c_3P^{-1}.

$$[00000000000000000000000000010001] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000000000100010]$$

$$\cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000000000001000100]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000000010001000]$$

$$\cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000001000100000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000010001000000000]$$

$$\cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000100010000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000001000100000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000000001000100000000000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000000010001000000000000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000000100010000000010000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00000001000100000000000000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00010001000000000000000000000000]$$

$$\cdot [01101010000000001101111111110101] = 0$$

$$[00100010000000000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[01000100000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[10001000000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 4 untuk c_3P^{-1} adalah 0. Kemudian setiap vektor karakteristik baris 3 dikalikan titik dengan c_3P^{-1}.

$$[00000000000000000000000000000101] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000000000001010] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000000001010000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000000010100000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000000010100000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000001010000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000000101000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000000001010000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000000001010000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000000101000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000000101000000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000001010000000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00000101000000000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[00001010000000000000000000000000] \cdot [01101010000000001101111111110101] = 0$$

$$[01010000000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[10100000000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 3 untuk c_3P^{-1} adalah 0. Kemudian setiap vektor karakteristik baris 2 dikalikan titik dengan c_3P^{-1}.

$$[00000000000000000000000000000011] \cdot [01101010000000001101111111110101] = 1$$

$$[00000000000000000000000000001100]$$
$$\cdot [01101010000000001101111111110101] = 1$$
$$[00000000000000000000000000110000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000000000000000000011000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000000000000000001100000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000000000000000110000000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000000000000011000000000000]$$
$$\cdot [01101010000000001101111111110101] = 1$$
$$[00000000000000001100000000000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000000001100000000000000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000001100000000000000000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000000110000000000000000000000]$$
$$\cdot [01101010000000001101111111110101] = 0$$
$$[00000011000000000000000000000000]$$
$$\cdot [01101010000000001101111111110101] = 1$$

$$[00001100000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[00110000000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

$$[11000000000000000000000000000000] \cdot [01101010000000001101111111110101] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 2 untuk c_3P^{-1} adalah 0. Kemudian menghitung M_y dan menambahkannya dengan c_3P^{-1},

$$M_y = \sum koefisien\ baris\ x_i\ (vektor\ baris\ x_i)$$

$$M_y = 1[00000000000000001111111111111111] + 0[00000000111111110000000011111111] + 0[00001111000011110000111100001111] + 0[00110011001100110011001100110011] + 0[01010101010101010101010101010101]$$

$$= [00000000000000001111111111111111]$$

$$M_y + c_3P^{-1} = [00000000000000001111111111111111] + [01101010000000001101111111110101]$$

$$= [01101010000000000010000000001010]$$

Dapat diketahui hasil dari $M_y + c_3P^{-1}$, jumlah bilangan 0 lebih banyak dari 1, sehingga koefisien baris 1 untuk c_3P^{-1} adalah 0. Kemudian membentuk vektor baru dari semua koefisien yang diperoleh mulai dari koefisien baris 1 sampai 6. Sehingga diperoleh hasil *decoding* c_3P^{-1} adalah vektor $p_3S = [\,0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1]$.

Berikut perkalian titik setiap vektor karakteristik baris 6 dengan $c_4 P^{-1}$.

$$[00000000000000010000000000000001] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000010000000000000010] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000010000000000000100] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000001000000000000001000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000100000000000000010000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000001000000000000000100000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000010000000000000001000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000100000000000000010000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000001000000000000000100000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000010000000000000001000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000100000000000000010000000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00001000000000000000100000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00010000000000000001000000000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00100000000000000010000000000000] \cdot [10010101111111110010000000001010] = 1$$

$$[01000000000000000100000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[10000000000000001000000000000000] \cdot [10010101111111110010000000001010] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 1 jumlahnya lebih banyak dari hasil bernilai 0, sehingga koefisien baris 6 untuk c_4P^{-1} adalah 1. Kemudian setiap vektor karakteristik baris 5 dikalikan titik dengan c_4P^{-1}.

$$[00000000000000000000000100000001] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000001000000010] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000000010000000100] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000100000001000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000001000000010000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000010000000100000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[00000000000000000100000001000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000000000000001000000010000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000010000000100000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000100000001000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[00000100000001000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00001000000010000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[00010000000100000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00100000001000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[01000000010000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[10000000100000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 5 untuk c_4P^{-1} adalah 0.

Kemudian setiap vektor karakteristik baris 4 dikalikan titik dengan $c_4 P^{-1}$.

$$[00000000000000000000000000010001] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000000000100010] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000000000001000100] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000000010001000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000000001000100000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000010001000000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000100010000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000001000100000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000001000100000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000010001000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000100010000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000100010000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00010001000000000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00100010000000000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[01000100000000000000000000000000] \cdot [10010101111111110010000000001010] = 1$$

$$[10001000000000000000000000000000] \cdot [10010101111111110010000000001010] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 4 untuk c_4P^{-1} adalah 0. Kemudian setiap vektor karakteristik baris 3 dikalikan titik dengan c_4P^{-1}.

$$[00000000000000000000000000000101] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000000000001010] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000000001010000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000000010100000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000010100000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000101000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000000000000000101000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000000000000001010000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[00000000000001010000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000000000101000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000000101000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000001010000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000101000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00001010000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[01010000000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[10100000000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 3 untuk $c_4 P^{-1}$ adalah 0.

Kemudian setiap vektor karakteristik baris 2 dikalikan titik dengan $c_4 P^{-1}$.

$$[00000000000000000000000000000011] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000000000000001100] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000000000000000110000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000000011000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000001100000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000000110000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000000000011000000000000] \cdot [10010101111111110010000000001010] = 1$$

$$[00000000000000001100000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000000001100000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000001100000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000110000000000000000000000] \cdot [10010101111111110010000000001010] = 0$$

$$[00000000110000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 0$$
$$[00000011000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[00001100000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[00110000000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$
$$[11000000000000000000000000000000]$$
$$\cdot [10010101111111110010000000001010] = 1$$

Dapat diketahui dari perkalian titik tersebut, hasil dengan nilai 0 jumlahnya lebih banyak dari hasil bernilai 1, sehingga koefisien baris 2 untuk c_4P^{-1} adalah 0. Kemudian menghitung M_y dan menambahkannya dengan c_4P^{-1},

$$M_y = \sum koefisien\ baris\ x_i\ (vektor\ baris\ x_i)$$

$$M_y = 1[00000000000000001111111111111111]$$
$$+ 0[00000000111111110000000011111111]$$
$$+ 0[00001111000011110000111100001111]$$
$$+ 0[00110011001100110011001100110011]$$
$$+ 0[01010101010101010101010101010101]$$
$$= [00000000000000001111111111111111]$$

$$M_y + c_4P^{-1} = [00000000000000001111111111111111]$$
$$+ [10010101111111110010000000001010]$$
$$= [10010101111111111101111111110101]$$

Dapat diketahui hasil dari $M_y + c_4P^{-1}$, jumlah bilangan 1 lebih banyak dari 1, sehingga koefisien baris 1 untuk c_4P^{-1} adalah 1. Kemudian membentuk vektor baru dari semua koefisien yang diperoleh mulai dari koefisien baris 1 sampai 6. Sehingga diperoleh hasil *decoding* c_4P^{-1} adalah vektor $p_4S = [\,1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1]$. Jadi dari seluruh proses algoritma *decoding* sebelumnya menghasilkan vektor berikut ini.

$$p_1S = [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$p_2S = [\,0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1]$$

$$p_3S = [\,0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1]$$

$$p_4S = [\,1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1]$$

Kemudian memasuki tahapan terakhir proses dekripsi, mengoperasikan rumus persamaan dekripsi terhadap p_iS yaitu mengalikannya dengan S^{-1} untuk mendapatkan pesan asli p_i.

$$p_1 = p_1SS^{-1} = [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1] \times \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 0]$$

$$p_2 = p_2SS^{-1} = [\,0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1] \times \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = [\,1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$p_3 = p_3SS^{-1} = [\,0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1] \times \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = [\,0\ \ 1\ \ 0\ \ 1\ \ 0\ \ 1]$$

$$p_4 = p_4 S S^{-1} = [\,1\ 0\ 0\ 0\ 0\ 1\,] \times \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = [\,0\ 0\ 1\ 1\ 1\ 0\,]$$

Telah diperoleh pesan asal p_1, p_2, p_3, p_4 yang sebelumnya di enkripsi. Namun pesan tersebut masih berupa vektor kode biner, sehingga perlu dikonversikan untuk mengetahui maksud pesan. Ingat bahwa pesan yang dikirimkan berdasarkan ASCII 256-bit yang merepresentasikan huruf atau simbolnya dalam kode biner 8-bit. Maka akan digabungkan p_1, p_2, p_3, p_4 menjadi vektor p berukuran 1×24 kemudian di-bagi menjadi blok-blok vektor kode biner sepanjang 8-bit untuk memulihkan pesan sesuai tabel ASCII 256-bit.

$$p = [p_1, p_2, p_3, p_4] = [010100100101010101001110]$$

$$[\,0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\,] = R$$

$$[\,0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\,] = U$$

$$[\,0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\,] = N$$

Sehingga diperoleh pesan aslinya yaitu RUN.

Adapun simulasi proses dekripsi untuk $r \in \{2, 3\}$ dan $m = 5$ dilakukan dengan bantuan pemrograman, yang hasilnya terdapat pada lampiran 4 dan 5.

4.3 Perbandingan Hasil Implementasi Dengan Menggunakan Parameter Kode $RM(1,5)$, $RM(2,5)$ Dan $RM(3,5)$

Implementasi algoritma kriptosistem *McEliece* menggunakan kode *Reed-Muller* $RM(1,5)$, $RM(2,5)$ dan $RM(3,5)$ juga dilakukan dengan bantuan pemrograman. Pada simulasinya, pesan yang di enkripsi yaitu “Matematika ilmu

yang menyenangkan" di mana terdiri dari 30 huruf atau karakter dan kode binernya berjumlah 240-bit. Hasil pemrogramannya terdapat pada lampiran 2, 3, 4 dan 5.

Tabel 4.1 Ringkasan hasil pemrograman untuk implementasi kriptosistem *McEliece* menggunakan kode *Reed-Muller*

r	m	Panjang blok pesan	Jumlah pesan	Bobot *error*
1	5	6-bit	40 blok	7
2	5	16-bit	15 blok	3
3	5	26-bit	10 blok	1

Berdasarkan Tabel 4.1 diketahui implementasi dengan $RM(1,5)$ memperoleh parameter sistem $(n = 32, k = 6, t = 7)$. Pada simulasinya, diperoleh ukuran matriks kunci privat $G_{6\times32}, S_{6\times6}, P_{32\times32}$ dan kunci publik $G'_{6\times32}$. Adapun panjang blok pesan yang di enkripsi yaitu 6-bit, mengikuti parameter sistem k. Kemudian pembagian pesan yang berjumlah 240-bit ke dalam blok pesan sepanjang 6-bit, menghasilkan bahwa terdapat 40 blok pesan yang di enkripsi. Dalam simulasinya juga ditambahkan bobot *error* sebanyak 7-bit sesuai dengan batas maksimum koreksi *error* (t) parameter ini.

Kemudian implementasi dengan $RM(2,5)$ memperoleh parameter sistem $(n = 32, k = 16, t = 3)$. Pada simulasinya, diperoleh ukuran matriks kunci privat $G_{16\times32}, S_{16\times16}, P_{32\times32}$ dan kunci publik $G'_{16\times32}$. Adapun panjang blok pesan yang di enkripsi yaitu 16-bit, mengikuti parameter sistem k. Kemudian pembagian pesan yang berjumlah 240-bit ke dalam blok pesan sepanjang 16-bit, menghasilkan bahwa terdapat 15 blok pesan yang dienkripsi. Dalam simulasinya juga ditambahkan bobot *error* sebanyak 3-bit sesuai batas maksimum koreksi *error* (t) parameter ini.

Implementasi dengan kode $RM(2,5)$ memiliki *runtime* untuk enkripsi selama 0.785 *seconds* dan dekripsi selama 647.2 *seconds*.

Sedangkan implementasi dengan $RM(3,5)$ memperoleh parameter sistem $(n = 32, k = 26, t = 1)$. Pada simulasinya, diperoleh ukuran matriks kunci $G_{26\times32}, S_{26\times26}, P_{32\times32}$ dan $G'_{26\times32}$. Adapun panjang blok pesan yang di enkripsi yaitu 26-bit, mengikuti parameter sistem k. Kemudian pembagian pesan yang berjumlah 240-bit ke dalam blok pesan sepanjang 26-bit, menghasilkan bahwa terdapat 10 blok pesan yang di enkripsi. Pada blok pesan terakhir, tersisa 6-bit kode biner sehingga perlu menambahkan bit tambahan 0 sebanyak 20-bit untuk melengkapi blok pesan yang harusnya sepanjang 26-bit. Dalam simulasinya juga ditambahkan bobot *error* 1-bit sesuai batas maksimum koreksi *error* (t) parameter ini. *Runtime* pada implementasi parameter ini untuk enkripsi selama 0.785 *seconds* dan dekripsi selama 647.2 *seconds*.

Berdasarkan deskripsi di atas diketahui bahwa implementasi dengan menggunakan parameter kode $RM(3,5)$ memiliki proses implementasi paling singkat karena blok pesan yang di enkripsi dan dekripsi hanya berjumlah 10 blok vektor, jumlahnya paling sedikit dibandingkan dengan $RM(2,5)$ yang memiliki 15 blok dan $RM(1,5)$ yang memiliki 40 blok. Selain itu, dengan menggunakan kode $RM(3,5)$ memiliki *runtime* enkripsi selama 0.785 *seconds* dan dekripsi selama 647.2 *seconds*, lebih cepat dari $RM(2,5)$ yang *runtime* enkripsinya selama 0.785 *seconds* dan dekripsi selama 647.2 *seconds*. Namun pada implementasi dengan menggunakan $RM(3,5)$ koreksi *error* yang dapat dilakukan hanya sebatas 1-bit, paling sedikit dibandingkan dengan $RM(2,5)$ yang dapat melakukan koreksi *error* sebanyak 3-bit dan $RM(1,5)$ yang dapat melakukan koreksi *error* hingga 7-bit.

4.4 Analisis Hasil

Pembangkitan kunci pada algoritma kriptosistem *McEliece* menggunakan kode *Reed-Muller* memiliki peran penting dalam membentuk suatu dasar keamanan. Pembangkitan kunci ini menghasilkan kunci privat S (matriks *non-singular* $k \times k$), G (matriks generator $RM(r, m)$ $k \times n$), P (matriks permutasi $n \times n$) dan kunci publik G' (matriks $k \times n$). Proses ini diawali dengan menentukan parameter kode (r, m) yang juga menjadi dasar untuk mendapatkan parameter sistem (n, k, t). Secara tidak langsung, dapat diketahui bahwa parameter kode maupun sistem tersebut berfungsi untuk mengidentifikasi matriks kunci, baik pada segi ukuran maupun proses pembentukannya. Adapun kunci G' terbentuk dari perkalian seluruh kunci privat dan penggunaan kunci S dan P yang merupakan matriks acak bertujuan untuk merandomisasi kunci G yang dibangkitkan kode *Reed-Muller*, supaya tidak mudah diidentifikasi oleh pihak lain.

Proses enkripsi pada algoritma kriptosistem *McEliece* diawali dengan menyiapkan pesan berbentuk blok-blok vektor kode biner dengan panjang k bit dan *error* berbobot t. Proses enkripsinya yaitu mengalikan vektor pesan dengan kunci G' dan menambahkan *error* yang menghasilkan cipherteks berupa vektor dengan panjang n. Diketahui parameter sistem juga berfungsi mengidentifikasi bobot *error* dan ukuran pesan serta cipherteks yang dihasilkan pada proses enkripsi. Perkalian vektor pesan dengan G' sebagai kunci publik menjadikan pesan sulit diketahui oleh pihak yang tidak berhak. Penambahan *error* secara acak juga menambah lapisan keamanan pada cipherteks yang dihasilkan. Sehingga proses enkripsi ini menjadi suatu langkah kritis dalam penyandian pesan untuk menjaga kerahasiaan suatu pesan pada algoritma kriptosistem *McEliece*.

Proses dekripsi pada algoritma kriptosistem *McEliece* memiliki beberapa tahapan yang juga melibatkan algoritma *decoding* kode *Reed-Muller* untuk memulihkan pesan yang terenkripsi. Tahap mengalikan cipherteks dengan invers matriks permutasi P bertujuan untuk menghilangkan efek permutasi pada proses enkripsi yang terkandung pada kunci G', di mana $cP^{-1} = pSG + eP^{-1} = pSG + e'$. Kemudian penggunaan algoritma *decoding* kode *Reed-Muller* berfungsi untuk mengoreksi vektor yang masih dalam pengaruh kode *Reed-Muller* dan mengandung *error* yang ditambahkan pada proses enkripsi. Adapun hasil *decoding* yang diperoleh berupa vektor baru pS dengan panjang k yang tersusun dari koefisien baris. Selanjutnya perkalian antara vektor hasil *decoding* dengan invers dari matriks *non-singular* S merupakan tahap akhir untuk memperoleh pesan asli. Langkah ini dilakukan untuk menghilangkan pengaruh matriks S pada proses enkripsi yang juga terkandung pada kunci G'. Adapun parameter sistem juga berfungsi dalam mengidentifikasi ukuran vektor yang dihasilkan pada setiap proses dekripsi.

Simulasi implementasi kriptosistem *McEliece* menggunakan kode *Reed-Muller* $r = 1$ dan $m = 5$ menunjukkan proses mengenkripsi vektor pesan 6-bit yang menghasilkan cipherteks 32-bit. Selain itu, juga ditunjukkan tahapan mendekripsi cipherteks dan koreksi *error* sebanyak 7-bit hingga dapat memulihkan pesan 6-bit yang sebelumnya di enkripsi. Penelitian ini menunjukkan proses enkripsi dan proses dekripsi yang melibatkan langkah *decoding* kode *Reed-Muller* yang kompleks dan runtut untuk mengamankan suatu pesan. Algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* juga menunjukkan keberhasilannya, baik dalam memulihkan pesan yang terenkripsi dan juga koreksi *error*. Adapun jika bobot *error* yang ditambahkan melebihi batas maksimum

koreksi *error* (t), proses memulihkan pesan tidak akan berhasil karena pesan yang dihasilkan bukanlah pesan asli yang sebenarnya.

Adapun hasil implementasi menggunakan pemrograman dengan parameter m yang sama, jika parameter r semakin besar maka proses implementasi semakin singkat karena blok pesan yang dienkripsi dan dekripsi semakin sedikit juga *runtime* semakin cepat, namun batas maksimum koreksi *error*-nya semakin sedikit. Sebaliknya jika parameter r semakin kecil, proses implementasi semakin panjang karena blok pesan yang di enkripsi dan dekripsi semakin banyak juga runtime semakin lama, tetapi batas maksimum koreksi *error*-nya justru semakin banyak.

Dapat diketahui bahwa ukuran kunci memiliki pengaruh signifikan terhadap kemampuan koreksi *error* dan efisiensi waktu proses. Kunci berukuran lebih kecil menawarkan kemampuan koreksi lebih baik karena tingkat redundansi lebih tinggi. Namun meningkatkan kompleksitas komputasi dan memperpanjang waktu pemrosesan karena jumlah blok pesan lebih besar. Sebaliknya, kunci berukuran lebih besar, lebih efisien dalam proses dan lebih cepat diimplementasikan, tetapi kemampuan koreksi lebih terbatas. Karena itu, pemilihan kunci tergantung pada kebutuhan aplikasi. Jika efisiensi dan kecepatan diprioritaskan, kunci berukuran lebih besar yang cocok. Jika lingkungan memiliki tingkat kesalahan tinggi, kunci berukuran lebih kecil direkomendasikan untuk memastikan keandalan data.

4.5 Integrasi Keislaman Terhadap Implementasi Algoritma Kriptosistem *McEliece* Menggunakan Kode *Reed-Muller*

Berdasarkan pembahasan sebelumnya telah diperoleh proses dan hasil simulasi dari implementasi algoritma kriptosistem *McEliece* dengan menggunakan

kode *Reed-Muller*. Adapun prosesnya terbagi menjadi dua yaitu proses enkripsi dan proses dekripsi. Sebagaimana yang telah diketahui proses enkripsi merupakan proses menyandikan suatu pesan sehingga menjadi teks sandi atau cipherteks yang mana proses ini dilakukan oleh pengirim. Hal ini tentu ditujukan untuk menjaga keamanan atau kerahasiaan suatu pesan saat dikirim dari suatu tempat ke tempat yang lain. Terdapat hadits yang membahas tentang menjaga rahasia sebagai berikut.

حَدَّثَنَا أَبُوبَكَرِ بْنِ أَبِي شَيْبَةَ، نَا يَحْيَ بْنِ آدَم، نَا اِبْنُ أَبِي ذِئْب، عَنْ عَبْدِالرَّحْمٰنِ بْنِ عَطَاء، عَنْ عَبْدِالمٰلِكِ بْنِ جَابِرِ بْنِ عَتِيْك، عَنْ جَابِرِ بْنِ عَبْدِاللهِ قَالَ: قَالَ رَسُوْلُ اللهِ صَلَّى اللهُ عَلَيْهِ وَسَلَّمَ: إِذَا حَدَّثَ الرَّجُلُ بِالْحَدِيْثِ ثُمَّ الْتَفَتَ فَهِيَ أَمَانَةٌ.(رواه أبو داود)

"Rasulullah SAW bersabda: Apabila seseorang membicarakan sesuatu kepada orang lain (sambil) menoleh kanan kiri (karena yang dibicarakannya itu rahasia) maka itulah amanah (yang harus dijaga)."(H.R. Abu Daud)

Hadits tersebut menjelaskan keadaan seseorang dalam menjaga suatu rahasia yang mana hal tersebut merupakan bagian dalam menjaga amanah (Abadi, 2009). Hal ini tentu sejalan dengan proses enkripsi pada implementasi algoritma ini yang digunakan untuk merahasiakan dan menjaga keamanan pesan tersebut. Perlu diketahui bahwasannya proses enkripsi merupakan bagian dari algoritma kriptosistem *McEliece* yang mana algoritma ini memiliki sistem keamanan yang tidak mudah dipecahkan oleh serangan kuantum dan terjamin keamanannya. Sehingga dengan menggunakan implementasi ini dapat menghindari hal yang tidak diinginkan, seperti bocornya pesan hingga hilangnya kerahasiaan pesan tersebut.

Kemudian proses dekripsi yang dilakukan oleh penerima merupakan proses untuk memulihkan pesan yang disandikan. Pada simulasinya telah ditunjukkan bahwa hanya pihak yang membangkitkan semua kunci yang digunakan pada algoritma tersebut yaitu penerima yang dapat memulihkan pesan kembali ke pesan

asli. Kemudian pada proses dekripsi juga memanfaatkan algoritma *decoding* kode *Reed-Muller* untuk meningkatkan keamanannya. Sehingga dengan begitu pesan hanya akan benar-benar diterima dan diketahui oleh penerima yang semestinya. Sebagaimana Allah SWT berfirman mengenai penyampaian amanah pada ayat Al-Qur'an surat An-Nisaa' ayat 58 sebagai berikut.

إِنَّ اللهَ يَأْمُرُكُمْ اِنَّ تُؤَدُّوا الْأَمٰنٰتِ اِلٰى اَهْلِهَاۙ وَاِذَاحَكَمْتُمْ بَيْنَ النَّاسِ اَنْ تَحْكُمُوْا بِالْعَدْلِۗ اِنَّ اللهَ نِعِمَّا يَعِظُكُمْ بِهٖۗ اِنَّ اللهَ كَانَ سَمِيْعًاۢ بَصِيْرًا﴿٥٨﴾

"Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum diantara manusia supaya kamu menetapkannya dengan adil. Sesungguhnya, Allah memberi pelajaran yang sebaik-baiknya kepadamu. Sesungguhnya, Allah Maha Pendengar, lagi Maha Melihat." (Lajnah Pentashihan Mushaf Al-Qur'an, 2019)

Ayat di atas menjelaskan seberapa penting untuk menyampaikan suatu amanah kepada pihak yang seharusnya menerimanya. Hal ini menunjukkan kewajiban pihak yang diberikan amanah, dan jika tidak dilaksanakan akan dimintai pertanggungjawaban di akhirat. Adapun tujuan menyampaikan amanah tersebut juga untuk menjaga amanah agar tersampaikan dengan baik kepada orang yang berhak. Secara tidak langsung, ayat ini juga mengajarkan untuk meneladani salah satu sifat wajib Nabi yaitu amanah yang artinya dapat dipercaya. Sebagaimana dalam menyampaikan amanah bertanggung jawab untuk jujur dan menjaga kebenaran amanah sehingga menciptakan suatu kepercayaan. Sehingga apabila seseorang meneladani sifat amanah tersebut dengan baik maka akan menghasilkan hal yang baik, seperti tersampaikannya suatu pesan atau amanah dengan aman dan terjaga kerahasiaannya tanpa adanya kebocoran pesan sehingga pihak yang menyampaikan amanah juga memperoleh manfaat yaitu diberikan suatu kepercayaan atas perilakunya.

BAB V

PENUTUP

5.1 Kesimpulan

Implementasi algoritma kriptosistem *McEliece* menggunakan kode *Reed-Muller* dengan parameter kode $RM(1,5)$ menghasilkan parameter umum untuk kriptosistemnya yaitu $n = 32$, $k = 6$ dan $t = 7{,}5$. Proses pembangkitan kunci memperoleh matriks generator $G_{6\times32}$, matriks *non-singular* $S_{6\times6}$ dan matriks permutasi $P_{32\times32}$ yang menghasilkan matriks $G'_{6\times32}$.

Pada implementasi menggunakan parameter kode $RM(1,5)$ mengenkripsi setiap blok pesan 6-bit dan menambah *error* berbobot maksimum 7-bit yang menghasilkan vektor kode 32-bit setiap bloknya. Proses dekripsi memanfaatkan algoritma *decoding Reed-Muller* yang dapat mengoreksi *error* hingga 7-bit. Sehingga diperoleh setiap blok vektor asal 6-bit yang merupakan pesan asli.

Perbandingan hasil implementasi menggunakan parameter berbeda, menunjukkan ukuran kunci memiliki pengaruh signifikan terhadap kemampuan koreksi *error* dan efisiensi waktu proses. Kunci berukuran lebih kecil menawarkan kemampuan koreksi lebih baik. Namun meningkatkan kompleksitas komputasi dan memperpanjang waktu pemrosesan. Sebaliknya, kunci berukuran lebih besar, lebih efisien dalam proses dan lebih cepat diimplementasikan, tetapi kemampuan koreksi lebih terbatas. Karena itu, pemilihan kunci tergantung pada kebutuhan aplikasi. Jika efisiensi dan kecepatan diprioritaskan, kunci berukuran lebih besar yang cocok. Jika lingkungan memiliki tingkat kesalahan tinggi, kunci berukuran lebih kecil direkomendasikan untuk memastikan keandalan data.

5.2 Saran

Penelitian mengenai implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* ini hanya berbatas tentang proses dan simulasinya. Adapun penelitian ini menggunakan pesan teks sebagai objek simulasi. Sehingga disarankan penelitian selanjutnya untuk melakukan modifikasi objek pada algoritma ini dan juga dapat dilakukan analisis keamanan terhadap penggunaan algoritma ini untuk menghadapi serangan komputer kuantum.

DAFTAR PUSTAKA

Abadi, A. T. M. S. H. A.-A. (2009). *Aunul Ma'bud Syarah Sunan Abi Dawud*. Dar Ibnu Jauzi.

Ariyus, D. (2008). *Pengantar Ilmu Kriptografi*. ANDI OFFSET.

Cornelissenn, M. G., Araujo, I. F., & Melo, R. R. de A. (2020). Codigos de Reed-Muller. *REMAT: Revista Electronica Da Matematica*, *6*(1), 1–13.

Gilbert, L., & Gilbert, J. (2015). *Elements of Modern Algebra* (8th ed.). Cengage Learning.

Handoko, P. (2019). *DASAR-DASAR PEMROGRAMAN*.

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.

Katsir, I. (2015). *Tafsir Ibnu Katsir*. Penebar Sunnah.

Kemenag, R. (2019). Al-Qur'an dan Terjemah Juz 20-30. In *Al-Qur'an dan Terjemahannya Edisi Penyempurnaan 2019*.

Kusumawati, R. (2009). *Aljabar Linear & Matriks*. UIN-Malang Press.

Lajnah Pentashihan Mushaf Al-Qur'an. (2019). Al-Qur'an dan Terjemah juz 1-10. In *Al-Qur'an dan Terjemahannya Edisi Penyempurnaan 2019*.

Ling, S., & Xing, C. (2004). Coding Theory. In *Вестник Росздравнадзора* (Vol. 4, Issue 1). Cambridge University Press.

Lint, J. H. van. (2007). *Introduction to Coding Theory*. https://www.amazon.com/Abstract-Algebra-Graduate-Texts-Mathematics/dp/0387715673

MacWilliams, F. J., & Sloane, N. J. a. (1988). *The Theory of Error-Correcting Codes*.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. In *Handbook of Applied Cryptography*. https://doi.org/10.2307/2589608

Munir, R. (2010). Matematika Diskrit Edisi 3. In *Informatika Bandung*. https://www.academia.edu/29914530/Matematika_Diskrit_RInaldi_Munir

Oktavia, R. E., Utomo, P. H., & Martini, T. S. (2023). Penerapan Kode Reed Solomon Pada Kriptosistem Mceliece. *FIBONACCI: Jurnal Pendidikan Matematika Dan Matematika*, *9*(1), 79–88. https://doi.org/10.24853/fbc.9.1.79-88

Pongsitammu, V., Simatupang, A. R. Y., Annura, D., Sari, Y., Dachi, & Harries, D. R. (2023). Keamanan Kriptosistem Modern Berdasarkan Algoritma

Kriptografi Kunci Publik. *Siteba*, *2*(1), 1–6. https://journal.iteba.ac.id/index.php/jurnalsiteba/indexSITEBA

Sadikin, R. (2012). *Kriptografi untuk Keamanan Jaringan*. ANDI OFFSET.

Setyawan, I., & Utomo, P. H. (2022). Penerapan Kode Golay Diperpanjang pada Kriptosistem McEliece. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, *6*(1), 20–23. https://doi.org/10.30865/komik.v6i1.5784

Shihab, M. Q. (2007). *Ensiklopedia Al-Qur'an*. www.tedisobandi.blogspot.com

Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, *26*(5), 1484–1509. https://doi.org/10.1137/S0097539795293172

Sutanta, E. (2005). *Pengantar Teknologi Informasi* (Pertama). Graha Ilmu.

Takagi, T., Wakayama, M., Tanaka, K., Kunihiro, N., Kimoto, K., & Ikematsu, Y. (2021). *Correction to: International Symposium on Mathematics, Quantum Theory, and Cryptography*. https://doi.org/10.1007/978-981-15-5191-8_19

Tilborg, H. C. A. Van. (2006). Encyclopedia of cryptography and security. In *Choice Reviews Online* (Vol. 43, Issue 11). https://doi.org/10.5860/choice.43-6251

LAMPIRAN

Lampiran 1. Tabel ASCII 8 bit

Char	ASCII Code	Binary	Char	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

Lampiran 2. Pesan asli dengan daftar simbol dan kode biner berdasarkan tabel ASCII

Pesan	Matematika ilmu yang menyenangkan				
Simbol	**Kode Biner**	**Simbol**	**Kode Biner**	**Simbol**	**Kode Biner**
M	01001101	i	01101001	n	01101110
a	01100001	l	01101100	y	01111001
t	01110100	m	01101101	e	01100101
e	01100101	u	01110101	n	01101110
m	01101101	y	01111001	a	01100001
a	01100001	a	01100001	n	01101110
t	01110100	n	01101110	g	01100111

i	01101001	g	01100111	k	01101011
k	01101011	m	01101101	a	01100001
a	01100001	e	01100101	n	01101110

Lampiran 3. Hasil pemrograman untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* $\boldsymbol{r = 1}$ dan $\boldsymbol{m = 5}$

Kunci	
$\boldsymbol{G_{6\times32}}$	$\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix}$
$\boldsymbol{S_{6\times6}}$	$\begin{bmatrix} 1&0&0&1&0&1 \\ 0&1&0&1&0&1 \\ 0&1&1&1&1&0 \\ 0&1&1&0&0&0 \\ 1&1&0&0&1&0 \\ 1&1&0&0&0&1 \end{bmatrix}$

$\boldsymbol{P_{32\times 32}}$	$\begin{bmatrix} 00001000000000000000000000000000 \\ 00000000000000000001000000000000 \\ 00000000000000001000000000000000 \\ 00100000000000000000000000000000 \\ 00000000000000000000000000100000 \\ 00000000001000000000000000000000 \\ 00000000000000000100000000000000 \\ 00000000000000000000000100000000 \\ 00000000000000000010000000000000 \\ 00000000000100000000000000000000 \\ 00000000000000000000000000000001 \\ 00010000000000000000000000000000 \\ 00000000010000000000000000000000 \\ 00000010000000000000000000000000 \\ 00000000000010000000000000000000 \\ 01000000000000000000000000000000 \\ 00000100000000000000000000000000 \\ 00000000000001000000000000000000 \\ 00000000100000000000000000000000 \\ 00000000000000000000010000000000 \\ 00000000000000000000001000000000 \\ 00000000000000000000000001000000 \\ 00000000000000000000000000000010 \\ 00000000000000000000000000000100 \\ 00000000000000000000000000001000 \\ 00000000000000010000000000000000 \\ 00000000000000100000000000000000 \\ 00000000000000000000100000000000 \\ 10000000000000000000000000000000 \\ 00000000000000000000000000010000 \\ 00000000000000000000000010000000 \\ 00000001000000000000000000000000 \end{bmatrix}$
$\boldsymbol{G'_{6\times 32} = SGP}$	$\begin{bmatrix} 10111001000100001011001011010111 \\ 00110101110110100101000001111100 \\ 00010010100011001011101110111100 \\ 00000010101111111110100001101001 1 \\ 01011111100100011100101000110010 \\ 00001001010011011110110001110101 \end{bmatrix}$

Pesan	*Codeword*	Pesan	*Codeword*
p_1	010011	p_{21}	011000
p_2	010110	p_{22}	010110
p_3	000101	p_{23}	111001
p_4	110100	p_{24}	100111
p_5	011001	p_{25}	011011
p_6	010110	p_{26}	010110
p_7	110101	p_{27}	010101
p_8	100001	p_{28}	101110
p_9	011101	p_{29}	011110

p_{10}	000110	p_{30}	010110
p_{11}	100101	p_{31}	010101
p_{12}	101011	p_{32}	101110
p_{13}	011000	p_{33}	011000
p_{14}	010110	p_{34}	010110
p_{15}	100101	p_{35}	111001
p_{16}	101100	p_{36}	100111
p_{17}	011011	p_{37}	011010
p_{18}	010111	p_{38}	110110
p_{19}	010101	p_{39}	000101
p_{20}	111001	p_{40}	101110

Proses Enkripsi

e	0000000000000000000000001111111

Pesan	$\boldsymbol{c = pG' + e}$	**Cipherteks**
p_1	011000110000011001110110010001000	c_1
p_2	011010001111010001001010111000010	c_2
p_3	000010111111001000111100110110001	c_3
p_4	100011100111010100110010000000111	c_4
p_5	001011100001101100000111111001010	c_5
p_6	011010001111010001001010111000010	c_6
p_7	100001110011100011011110011100010	c_7
p_8	101100000101110101011110110111101	c_8
p_9	001011001010010011010111000110001	c_9
p_{10}	010111010010111000011010100111110	c_{10}
p_{11}	101100101110001010001110000011110	c_{11}
p_{12}	111111010100000000101111010100011	c_{12}
p_{13}	001001110101011011101011101111111	c_{13}
p_{14}	011010001111010001001010111000010	c_{14}
p_{15}	101100101110001010001110000011110	c_{15}
p_{16}	101010010010001111011001110000111	c_{16}
p_{17}	011100011000101011001101111111000	c_{17}
p_{18}	011000011011100110100110100010111	c_{18}
p_{19}	001111100010100001101100101001010	c_{19}
p_{20}	100101110000101110110101000111010	c_{20}
p_{21}	001001110101011011101011101111111	c_{21}

p_{22}	01101000111101000100101011100010	c_{22}
p_{23}	10010111000010111011010100011101	c_{23}
p_{24}	11101101011100110100010000111100	c_{24}
p_{25}	01110001100010101100110111111000	c_{25}
p_{26}	01101000111101000100101011100010	c_{26}
p_{27}	00111110001010000110110010100101	c_{27}
p_{28}	11110110101100100001001111110101	c_{28}
p_{29}	01111010011110001111000101011110	c_{29}
p_{30}	01101000111101000100101011100010	c_{30}
p_{31}	00111110001010000110110010100101	c_{31}
p_{32}	11110110101100100001001111110101	c_{32}
p_{33}	00100111010101101110101110111111	c_{33}
p_{34}	01101000111101000100101011100010	c_{34}
p_{35}	10010111000010111011010100011101	c_{35}
p_{36}	11101101011100110100010000111100	c_{36}
p_{37}	01111000110001110010000110001101	c_{37}
p_{38}	11010001111001001111100000110101	c_{38}
p_{39}	00001011111100100011110011011001	c_{39}
p_{40}	11110110101100100001001111110101	c_{40}

Proses Dekripsi	
Cipherteks	$\boldsymbol{c' = cP^{-1}}$
c_1	01010010100001010101110100100001
c_2	10011110010010010110111000010010
c_3	11000100111011000011010010110111
c_4	11000100111011001100101101001000
c_5	10010001010001101001111011100010
c_6	10011110010010010110111000010010
c_7	01101110010001101001111000011101
c_8	01010010011110100101110111011110
c_9	11110111001000001111100010000100
c_{10}	11000100000100111100101110110111
c_{11}	00110100000111000011101110111000
c_{12}	10010001101110011001111000011101
c_{13}	00111011111011001100101110110111
c_{14}	10011110010010010110111000010010

c_{15}	001101000001110000111011101110000
c_{16}	111101110010000000000111011110111
c_{17}	001110110001001100110100101101111
c_{18}	001101001110001100111011010001111
c_{19}	100111101011011010010001000100010
c_{20}	011000011011011010010001111011101
c_{21}	001110111110110011001011101101111
c_{22}	100111100100100101101110000100010
c_{23}	011000011011011010010001111011101
c_{24}	100111100100100110010001111011101
c_{25}	001110110001001100110100101101111
c_{26}	100111100100100101101110000100010
c_{27}	100111101011011010010001000100010
c_{28}	010111010111010110101101001011110
c_{29}	111101111101111100000111100001000
c_{30}	100111100100100101101110000100010
c_{31}	100111101011011010010001000100010
c_{32}	010111010111010110101101001011110
c_{33}	001110111110110011001011101101111
c_{34}	100111100100100101101110000100010
c_{35}	011000011011011010010001111011101
c_{36}	100111100100100110010001111011101
c_{37}	100100011011100101100001111000100
c_{38}	011011101011100101100001000111101
c_{39}	110001001110110000110100101101111
c_{40}	010111010111010110101101001011110

Cipherteks	***Decoding*** $(\boldsymbol{pS})$	$\boldsymbol{p = pS \times S^{-1}}$	**Pesan**
c_1	010110	010011	p_1
c_2	111111	010110	p_2
c_3	101001	000101	p_3
c_4	101000	110100	p_4
c_5	111010	011001	p_5
c_6	111111	010110	p_6
c_7	011001	110101	p_7
c_8	010100	100001	p_8

c_9	100010	011101	p_9
c_{10}	101010	000110	p_{10}
c_{11}	001100	100101	p_{11}
c_{12}	111000	101011	p_{12}
c_{13}	001011	011000	p_{13}
c_{14}	111111	010110	p_{14}
c_{15}	001100	100101	p_{15}
c_{16}	100011	101100	p_{16}
c_{17}	001000	011011	p_{17}
c_{18}	001110	010111	p_{18}
c_{19}	111100	010101	p_{19}
c_{20}	011111	111001	p_{20}
c_{21}	001011	011000	p_{21}
c_{22}	111111	010110	p_{22}
c_{23}	011111	111001	p_{23}
c_{24}	111110	100111	p_{24}
c_{25}	001000	011011	p_{25}
c_{26}	111111	010110	p_{26}
c_{27}	111100	010101	p_{27}
c_{28}	010001	101110	p_{28}
c_{29}	100001	011110	p_{29}
c_{30}	111111	010110	p_{30}
c_{31}	111100	010101	p_{31}
c_{32}	010001	101110	p_{32}
c_{33}	001011	011000	p_{33}
c_{34}	111111	010110	p_{34}
c_{35}	011111	111001	p_{35}
c_{36}	111110	100111	p_{36}
c_{37}	111001	011010	p_{37}
c_{38}	011010	110110	p_{38}
c_{39}	101001	000101	p_{39}
c_{40}	010001	101110	p_{40}

Pesan	***Codeword***	**Kode Biner**	**Simbol**
p_1	010011	01001101	M
p_2	010110		

p_3	000101	01100001	a
p_4	110100	01110100	t
p_5	011001	01100101	e
p_6	010110	01101101	m
p_7	110101	01100001	a
p_8	100001		
p_9	011101	01110100	t
p_{10}	000110	01101001	i
p_{11}	100101	01101011	k
p_{12}	101011		
p_{13}	011000	01100001	a
p_{14}	010110	01101001	i
p_{15}	100101	01101100	l
p_{16}	101100		
p_{17}	011011	01101101	m
p_{18}	010111	01110101	u
p_{19}	010101	01111001	y
p_{20}	111001		
p_{21}	011000	01100001	a
p_{22}	010110	01101110	n
p_{23}	111001	01100111	g
p_{24}	100111		
p_{25}	011011	01101101	m
p_{26}	010110	01100101	e
p_{27}	010101	01101110	n
p_{28}	101110		
p_{29}	011110	01111001	y
p_{30}	010110	01100101	e
p_{31}	010101	01101110	n
p_{32}	101110		
p_{33}	011000	01100001	a
p_{34}	010110	01101110	n
p_{35}	111001	01100111	g
p_{36}	100111		
p_{37}	011010	01101011	k
p_{38}	110110		

p_{39}	000101	01100001	a
p_{40}	101110	01101110	n
Pesan yang diperoleh	Matematika ilmu yang menyenangkan		

Lampiran 4. Hasil pemrograman untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* $\boldsymbol{r = 2}$ dan $\boldsymbol{m = 5}$

Kunci	
$\boldsymbol{G_{16\times32}}$	$\begin{bmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\ 0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\ 0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \end{bmatrix}$
$\boldsymbol{S_{16\times16}}$	$\begin{bmatrix} 1&0&0&1&1&0&1&1&0&1&1&1&1&1&0&0 \\ 1&1&1&1&1&0&1&1&0&0&1&1&0&1&1&0 \\ 1&1&0&0&1&1&1&0&1&0&0&1&1&1&0&1 \\ 0&1&0&0&1&0&0&1&0&1&0&1&0&1&0&1 \\ 0&0&0&1&1&0&0&1&1&1&0&0&0&1&1&1 \\ 0&0&1&0&1&0&0&1&1&0&0&1&0&1&0&0 \\ 0&1&1&1&1&1&1&0&1&1&1&1&0&1&1&0 \\ 1&1&0&0&1&0&0&0&1&0&1&1&1&1&1&0 \\ 1&1&1&1&0&1&0&0&1&0&1&0&1&0&1&1 \\ 0&1&1&1&0&0&1&1&0&0&0&0&0&1&1&0 \\ 1&0&1&0&1&1&0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&1&0&0&0&1&0&1&0&1&0&0 \\ 1&1&1&1&1&1&0&0&0&0&1&0&0&0&0&1 \\ 1&0&0&0&1&0&1&1&1&1&1&1&1&1&0&0 \\ 1&0&1&1&1&1&1&0&1&1&0&1&0&0&1&1 \\ 0&1&1&1&1&1&1&0&0&0&1&0&0&1&0&0 \end{bmatrix}$

$\boldsymbol{P_{32\times32}}$	$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&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&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \end{bmatrix}$
$\boldsymbol{G'_{16\times32} = SGP}$	$\begin{bmatrix} 0&0&0&0&1&1&0&0&0&0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&1&1&0&1&0&0&0&0 \\ 1&1&0&0&1&0&0&1&0&0&0&1&1&0&1&0&0&0&0&0&0&1&0&1&0&0&0&0&1&0&1&0 \\ 1&0&0&0&0&1&1&0&0&1&1&0&0&0&1&1&0&1&1&1&1&0&1&1&1&0&1&1&1&1&1&1 \\ 1&0&1&0&0&1&1&1&1&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&1&0&1&0&1&1&0&1&0&1&0&1&0&1&1&0&1 \\ 1&0&0&1&0&0&0&0&1&1&1&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&1&1 \\ 1&1&0&0&1&0&0&0&1&1&0&1&1&0&0&1&0&1&1&1&1&0&1&1&0&1&1&1&1 \\ 1&0&1&0&0&1&0&1&0&0&0&0&1&0&0&1&0&0&1&1&1&1&1&1&1&0&1&1&0&0&0&1 \\ 1&0&0&1&1&0&1&0&1&1&1&0&1&0&1&1&0&1&0&0&0&0&0&0&1&0&0&1&1&1&1 \\ 0&0&1&0&1&0&1&1&1&0&1&1&1&0&1&1&0&1&0&1&1&0&1&1&1&0&1&0&0&1&1&1 \\ 0&0&0&1&1&0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&1&0&1&1&0&0&0 \\ 0&1&0&0&1&0&0&1&1&0&1&1&1&0&0&0&1&0&0&0&1&1&0&0&0&0&0&1&0&1&0 \\ 0&1&0&0&1&0&0&1&0&1&1&1&0&1&0&0&1&0&0&0&0&0&1&1&1&0&1&1&1&1&0 \\ 1&1&1&0&0&0&1&1&0&0&0&1&1&0&0&1&1&0&0&1&0&0&1&1&0&0&1&0&0&1&1&1&1&0&1&0 \\ 1&0&0&0&0&1&1&1&0&1&1&1&0&0&0&0&1&0&0&1&1&0&0&0&1&1&1&0&1&1 \\ 1&0&0&0&1&0&1&1&1&1&0&0&1&0&1&0&1&1&1&1&0&0&1&0&1&1&0&0&0&0&0 \end{bmatrix}$

Pesan	***Codeword***
p_1	0100110101100001
p_2	0111010001100101

p_3	0110110101100001
p_4	0111010001101001
p_5	0110101101100001
p_6	0110100101101100
p_7	0110110101110101
p_8	0111100101100001
p_9	0110111001100111
p_{10}	0110110101100101
p_{11}	0110111001111001
p_{12}	0110010101101110
p_{13}	0110000101101110
p_{14}	0110011101101011
p_{15}	0110000101101110

Proses Enkripsi		
e	00000000000000000000000000000111	
Pesan	$\boldsymbol{c = pG' + e}$	**Cipherteks**
p_1	010001011100110011011100011101 01	c_1
p_2	001000111111111001111100111001 00	c_2
p_3	110000111010111110100111110010 10	c_3
p_4	100010011001001111001101001000 00	c_4
p_5	100110111001001000111110101001 10	c_5
p_6	011100101110110000001101011011 01	c_6
p_7	011010010000111000011001101110 10	c_7
p_8	111101001000101111001001100000 01	c_8
p_9	110010100001011010011101000110 01	c_9
p_{10}	001000001011011010010101101100 00	c_{10}
p_{11}	101011101011001111101100101000 00	c_{11}
p_{12}	011001001111100111111001101101 00	c_{12}
p_{13}	111101000001110100011011101101 11	c_{13}
p_{14}	110001001111001101001001110000 01	c_{14}
p_{15}	111101000001110100011011101101 11	c_{15}

Proses Dekripsi	
Cipherteks	$\boldsymbol{c' = cP^{-1}}$
c_1	110111101000101111011001001000 01

c_2	011111101101101111100000000010111
c_3	011001011010011010101110001111111
c_4	000000111100111100000100011001O1
c_5	001010001110101101000110110101111
c_6	010101011001100111111000100110O1
c_7	100011010001101110001010010011O
c_8	000000111001010110011101101001O
c_9	100011010100000100011100011111O1
c_{10}	100011011101100000100000000100111
c_{11}	001000101101110100100101011101111
c_{12}	101110111101110111101001001000O
c_{13}	100011010111110111011111100000O
c_{14}	010100111100010100111101000001O
c_{15}	100011010111110111011111100000O

Cipherteks	***Decoding*** $(\boldsymbol{pS})$	$\boldsymbol{p = pS \times S^{-1}}$	**Pesan**
c_1	1010001011100001	0100110101100001	p_1
c_2	0111111110101100	0111010001100101	p_2
c_3	0110110001111100	0110110101100001	p_3
c_4	0000100001110001	0111010001101001	p_4
c_5	0011101100011110	0110101101100001	p_5
c_6	0100110000010001	0110100101101100	p_6
c_7	1110101111010100	0110110101110101	p_7
c_8	0000110010111101	0111100101100001	p_8
c_9	1110111100011011	0110111001100111	p_9
c_{10}	1110011110000000	0110110101100101	p_{10}
c_{11}	0010101001000001	0110111001111001	p_{11}
c_{12}	1100001010010001	0110010101101110	p_{12}
c_{13}	1110101100000101	0110000101101110	p_{13}
c_{14}	0100100110111111	0110011101101011	p_{14}
c_{15}	1110101100000101	0110000101101110	p_{15}

Pesan	***Codeword***	**Kode Biner**	**Simbol**
p_1	0100110101100001	01001101	M
		01100001	a
p_2	0111010001100101	01110100	t

		01100101	e
p_3	0110110101100001	01101101	m
		01100001	a
p_4	0111010001101001	01110100	t
		01101001	i
p_5	0110101101100001	01101011	k
		01100001	a
p_6	0110100101101100	01101001	i
		01101100	l
p_7	0110110101110101	01101101	m
		01110101	u
p_8	0111100101100001	01111001	y
		01100001	a
p_9	0110111001100111	01101110	n
		01100111	g
p_{10}	0110110101100101	01101101	m
		01100101	e
p_{11}	0110111001111001	01101110	n
		01111001	y
p_{12}	0110010101101110	01100101	e
		01101110	n
p_{13}	0110000101101110	01100001	a
		01101110	n
p_{14}	0110011101101011	01100111	g
		01101011	k
p_{15}	0110000101101110	01100001	a
		01101110	n
Pesan yang diperoleh		Matematika ilmu yang menyenangkan	
Runtime	**Enkripsi**	0.7856156826019287 *seconds*	
	Dekripsi	647.2970495223999 *seconds*	

Lampiran 5. Hasil pemrograman untuk implementasi algoritma kriptosistem *McEliece* dengan menggunakan kode *Reed-Muller* $\boldsymbol{r = 3}$ dan $\boldsymbol{m = 5}$

Kunci	
$\boldsymbol{G_{26\times32}}$	$\begin{bmatrix} 1111111111111111111111111111111 \\ 0101010101010101010101010101010 \\ 0011001100110011001100110011001 \\ 0000111100001111000011110000111 \\ 0000000011111111000000001111111 \\ 0000000000000000111111111111111 \\ 0001000100010001000100010001001 \\ 0000010100000101000001010000010 \\ 0000000010101010000000001010101 \\ 0000000000000000101010101010101 \\ 0000001100000011000000110000001 \\ 0000000000110011000000000011001 \\ 0000000000000000110011001100111 \\ 0000000000001111000000000001111 \\ 0000000000000000000011110000111 \\ 0000000000000000000000001111111 \\ 0000000100000001000000010000001 \\ 0000000000010001000000000010001 \\ 0000000000000000000100010001001 \\ 0000000000000101000000000000101 \\ 0000000000000000000010100000101 \\ 0000000000000000000000001010101 \\ 0000000000000011000000000000011 \\ 0000000000000000000000110000011 \\ 0000000000000000000000001100111 \\ 0000000000000000000000000001111 \end{bmatrix}$

$S_{26\times26}$	$\begin{bmatrix} 10100110111110101111001100 \\ 01010010010001111011000100 \\ 11011010000000000011010000 \\ 01101011101000110001100110 \\ 10000111011110010110011101 \\ 10101110001100110110011000 \\ 10000101011101000011101010 \\ 00100111010001011111101001 \\ 10110001010110000101010111 \\ 01101100001011101010011010 \\ 11001010100100100111000111 \\ 10101011111001010100001000 \\ 00110000111011111100001010 \\ 11100001001110000100101010 \\ 00001010110001110101110111 \\ 10100110100011000110111011 \\ 11101100011000010100111100 \\ 01001000010101000110000001 \\ 11100100011000111100001001 \\ 00101100111101000110101010 \\ 11100110101010001001011000 \\ 10000010100000011001111111 \\ 00111010000101101100001010 \\ 10010110101101010111111101 \\ 10101010001111010110100011 \\ 01000001000111111101011101 \end{bmatrix}$

$\boldsymbol{P}_{\mathbf{32\times 32}}$	$\begin{bmatrix} 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&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&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&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&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&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&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&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&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$

$G'_{26\times32} = SGP$	$\begin{bmatrix} 1101011000000111111000111110111 \\ 0110100001010001100110000100110 \\ 1001011010101010000101100010010 \\ 1010011100111100111100111010010 \\ 0101100001111001110110110001110 \\ 0101011001110110100001001100110 \\ 0101011001001000111111010011110 \\ 1010000111001111000000101110001 \\ 0101110011111001111011000010100 \\ 1100000110111111110001000011010 \\ 1111001110111000001011000110111 \\ 1101101011000000010111110110011 \\ 1110101101011110000101010111011 \\ 1011111110110011011000101110111 \\ 0000011001000001011001011000011 \\ 1001001100000011111011000001100 \\ 1111001101110101000101000111010 \\ 1000111001000110111010000011000 \\ 0111111001100100011011101101100 \\ 1110011110011001010001001000010 \\ 0101000101000000010101110101100 \\ 1111000110111111110001110101111 \\ 1100100101111010010100100001101 \\ 1111100101011000110000101100000 \\ 1001111011100111100000111100100 \\ 0100101101100001101101111010001 \end{bmatrix}$

Pesan	***Codeword***
p_1	01001101011000010111010001
p_2	10010101101101011000010111
p_3	01000110100101101011011000
p_4	01011010010110110001101101
p_5	01110101011110010110000101
p_6	10111001100111011011010110
p_7	01010110111001111001011001
p_8	01011011100110000101101110
p_9	01100111011010110110000101
p_{10}	10111000000000000000000000

Proses Enkripsi		
e	00000000000000000000000000000001	
Pesan	$\boldsymbol{c = pG' + e}$	**Cipherteks**
p_1	11001011111100000111001110101011	c_1
p_2	11110001101011010111011111101111	c_2
p_3	00000101111000110010101100101101	c_3

p_4	01101001101001011001111101100001	c_4
p_5	01111100111111111010011110001010	c_5
p_6	10100011011100101001101011100011	c_6
p_7	11101100000111101011101101111110	c_7
p_8	10011011101001001101010000100000	c_8
p_9	01010001000010100111001001011010	c_9
p_{10}	10111111111010001101110101110010	c_{10}

Proses Dekripsi

Cipherteks	$\boldsymbol{c' = cP^{-1}}$
c_1	00011001111111101110101111000001
c_2	10101011111111101111111100110010
c_3	00001101000110101110000101101110
c_4	01000010010110101111011110110100
c_5	11101101001010001111111011111001
c_6	01010110011111100010100101010101
c_7	01101111011011110101001010011111
c_8	11010000110101101011010010000000
c_9	10101111111110010000001000000000
c_{10}	11110010111101111110010011011100

Cipherteks	***Decoding* ($\boldsymbol{pS}$)**
c_1	00011111001001111101001110
c_2	11000000110000011000010010
c_3	00010100001000011111011110
c_4	01000111011001100111011001
c_5	10001010001101010011001111
c_6	01000000011111100111110100
c_7	01110001001010110000000010
c_8	10110010011010001001001110
c_9	11000101110000000000111100
c_{10}	10010000001000001011100111

$\boldsymbol{p = pS \times S^{-1}}$	**Pesan**
01001101011000010111010001	p_1
10010101101101011000010111	p_2

010001101001011010111011000	p_3
010110100101101100011011101	p_4
011101010111100101100000101	p_5
101110011001110111011010110	p_6
010101101110011111001011001	p_7
010110111001100000101101110	p_8
011001110110101101100000101	p_9
101110000000000000000000000	p_{10}

Pesan	***Codeword***	**Kode Biner**	**Simbol**
p_1 p_2 p_3 p_4	01001101011000010111010001 10010101101101011000010111 01000110100101101011011000 01011010010110110001101101	01001101	M
		01100001	a
		01110100	t
		01100101	e
		01101101	m
		01100001	a
		01110100	t
		01101001	i
		01101011	k
		01100001	a
		01101001	i
		01101100	l
		01101101	m
p_5 p_6 p_7 p_8	01110101011110010110000101 10111001100111011011010110 01010110111001111001011001 01011011100110000101101110	01110101	u
		01111001	y
		01100001	a
		01101110	n
		01100111	g
		01101101	m
		01100101	e
		01101110	n
		01111001	y
		01100101	e
		01101110	n
		01100001	a
		01101110	n

p_9 p_{10}	01100111011010110110000101 10111000000000000000000000	01100111	g
		01101011	k
		01100001	a
		01101110	n
Pesan yang diperoleh		Matematika ilmu yang menyenangkan	
Runtime	**Enkripsi**	0.04692506790161133 *seconds*	
	Dekripsi	0.17453503608703613 *seconds*	

RIWAYAT HIDUP

Eka Puspa Anggraini lahir di Ponorogo pada tanggal 5 Juli 2002, memiliki nama panggilan Anggi. Tempat tinggalnya berada di RT 004 RW 002 Dusun Krajan Desa Cekok Kecamatan Babadan Kabupaten Ponorogo. Anak pertama dari dua bersaudara dari pasangan Bapak Sukatman dan Ibu Sri Hidayah. Masa pendidikan penulis di mulai dari RA Muslimat Cekok 1, Ponorogo dari tahun 2007 hingga 2008. Kemudian dilanjutkan pendidikan dasar di MI Ma'arif Cekok, Ponorogo dan lulus pada tahun 2014. Penulis melanjutkan pendidikan jenjang menengah pertama di MTs Darul Huda Ponorogo dan lulus pada tahun 2017. Kemudian menempun pendidikan jenjang menengah atas di MA Darul Huda Ponorogo dan lulus pada tahun 2020.

Ketika menempuh pendidikan di MTs hingga MA, penulis juga menempuh pendidikan agama di Pondok Pesantren Darul Huda Mayak Ponorogo terhitung selama 6 tahun. Pada tahun 2020 penulis melanjutkan pendidikan di Universitas Islam Negeri Maulana Malik Ibrahim Malang tepatnya di program studi Matematika. Semasa menempuh pendidikan tinggi, penulis berkontribusi aktif dalam beberapa kepanitiaan internal kampus seperti PBAK-F dan KOMET. Di luar kampus, penulis juga berkontribusi aktif dalam organisasi IKADHA Malang dan menjabat sebagai bendahara tahun 2022 hingga 2024. Selain itu, penulis juga aktif mengikuti kegiatan ekternal kampus sejenis seminar.

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

BUKTI KONSULTASI

Nama : Eka Puspa Anggraini
NIM : 200601110028
Fakultas/Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Algoritma Kriptosistem McEliece Dengan Menggunakan Kode Reed-Muller
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Erna Herawati, M.Pd

No.	Tanggal	Hal	Tanda Tangan
1.	21 Desember 2023	Konsultasi Topik I	1.
2.	27 Desember 2023	Konsultasi Topik II	2.
3.	1 Februari 2024	Konsultasi Topik III	3.
4.	21 Februari 2024	Konsultasi Bab I dan III	4.
5.	16 Mei 2024	Konsultasi Bab I, II dan III	5.
6.	16 Mei 2024	Konsultasi Kajian Agama Bab I dan II	6.
7.	20 Mei 2024	ACC Kajian Agama Bab I dan II	7.
8.	20 Mei 2024	ACC Bab I, II, III dan Seminar Proposal	8.
9.	20 Mei 2024	ACC Seminar Proposal	9.
10.	17 Juli 2024	Konsultasi Revisi Seminar Proposal, Bab IV dan V	10.
11.	19 Juli 2024	Konsultasi Kajian Agama Bab IV	11.
12.	23 Juli 2024	Konsultasi Kajian Agama Bab IV	12.
13.	25 Juli 2024	ACC Bab IV dan V	13.

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

14.	25 Juli 2024	ACC Seminar Hasil	14.
15.	30 Juli 2024	ACC Kajian Agama Bab IV	15.
16.	20 Agustus 2024	Konsultasi Revisi Seminar Hasil	16.
17.	21 Agustus 2024	ACC Matrik Revisi Seminar Hasil	17.
18.	29 Agustus 2024	ACC Sidang Skripsi	18.
19.	10 September 2024	ACC Keseluruhan	19.

Malang, 10 September 2024
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc.
NIP. 19741129 200012 2 005