PENERAPAN FUNGSI HASH KRIPTOGRAFI DALAM VERIFIKASI INTEGRITAS DATA AL-QUR'AN DIGITAL MENGGUNAKAN ALGORITMA WHIRLPOOL

Skripsi

Untuk memenuhi sebagian persyaratan

mencapai derajat Sarjana S-1

Program Studi Matematika

diajukan oleh

NAUFAL RACHMADAN

15610004

Kepada

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2019

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi/Tugas akhir
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Naufal Rachmadan
NIM : 15610004
Judul Skripsi : Penerapan Fungsi Hash Kriptografi dalam Verifikasi Integritas Data Al-Qur'an Digital Menggunakan Algoritma Whirlpool

sudah dapat diajukan kembali kepada Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam bidang matematika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqasyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 18 Januari 2019

Pembimbing II

Dr. M. Wakhid Musthofa,S.Si., M.Si.,
NIP: 19800402 200501 1 003

Pembimbing I

M. Zaki Riyanto, M.Sc.,
NIP: 19840113 201503 1 001

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-554/Un.02/DST/PP.00.9/02/2019

Tugas Akhir dengan judul : PENERAPAN FUNGSI HASH KRIPTOGRAFI DALAM VERIFIKASI INTEGRITAS DATA AL-QUR'AN DIGITAL MENGGUNAKAN ALGORITMA WHIRLPOOL

yang dipersiapkan dan disusun oleh:

Nama : NAUFAL RACHMADAN
Nomor Induk Mahasiswa : 15610004
Telah diujikan pada : Rabu, 13 Februari 2019
Nilai ujian Tugas Akhir : A

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR

Ketua Sidang

Muhamad Zaki Riyanto, S.Si., M.Sc.
NIP. 19840113 201503 1 001

Penguji I

Malahayati, S.Si., M.Sc
NIP. 19840412 201101 2 010

Penguji II

Dr. Muhammad Wakhid Musthofa, S.Si., M.Si.
NIP. 19800402 200501 1 003

Yogyakarta, 13 Februari 2019
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
DEKAN

Dr. Murtono, M.Si.
NIP. 19691212 200003 1 001

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Naufal Rachmadan

NIM : 15610004

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Dengan ini menyatakan bahwa isi skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi dan sesungguhnya skripsi ini merupakan hasil pekerjaan penulis sendiri sepanjang pengetahuan penulis, bukan duplikasi atau saduran dari karya orang lain kecuali bagian tertentu yang penulis ambil sebagai bahan acuan. Apabila terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, 18 Januari 2019

Yang Menyatakan,

Naufal Rachmadan

Karya sederhana ini penulis persembahkan

untuk pengembangan kajian integrasi interkoneksi

keilmuan Islam dan Sains

Tuhan, mengapa engkau bersumpah dengan hal itu ?

Sesuatu yang ada, namun jarang disadari...

Sesuatu yang nyata, namun tak tampak...

Sesuatu yang menyatukan, namun juga memisahkan...

Menguntungkan, namun merugikan...

Menghidupkan, namun mematikan...

Berlalu...

dan takkan kembali...

Sungguh, kiranya kami memahami hakikatnya...

Kami mungkin tidak akan lalai dalam kesenangan...

dan selalu berada dalam ketakutan...

Tuhan, engkau telah mengajariku sebagian kecil dari ilmumu...

dan dengan itu aku melihatnya... sebuah pertunjukan orkestra...

melahirkan alun simponi yang indah, bersenandung di seluruh kehidupan...

Tuhan, ilmuku tidaklah lebih besar dari setetes air di luasnya lautan bumi...

Namun dengan ilmu ini, izinkanlah aku menjadi manfaat bagi yang lain.

Tuhan, bila saat itu tiba...

ketika aku mempertanggungjawabkan segalanya dihadapanmu...

aku ingin menjadi bagian dari mereka...

Orang - orang yang mendedikasikan hidupnya pada ilmu pengetahuan.

PRAKATA

Assalamu'alaikum, wr.wb.

Alhamdulillah, puji syukur atas kehadirat allah SWT yang telah memberikan kemudahan sehingga penulis dapat menyelesaikan penelitian ini dengan lancar. Tak lupa sholawat serta salam kepada baginda Rasulullah SAW, sebagai pemimpin umat muslim yang hidup di akhir zaman.

Alhamdulillah, dengan selesainya penelitian ini, penulis telah menyelesaikan salah satu syarat untuk memperoleh gelar sarjana matematika di Universitas Islam Negeri Sunan Kalijaga Yogyakarta. Penulis berharap, penelitian ini dapat dikembangkan lebih jauh lagi. Ucapan terima kasih penulis sampaikan kepada :

1. Bapak Dr. Wakhid Musthofa, M.Si. selaku Ketua prodi matematika, Dosen pembimbing akademik sekaligus Dosen pembimbing skripsi yang telah memberikan bimbingan dalam penyempurnaan tugas akhir ini.
2. Bapak M. Zaki Riyanto, M.Sc. selaku dosen pembimbing skripsi yang telah memberikan banyak arahan dalam pengembangan penelitian ini.
3. Bapak dan ibu dosen program studi matematika yang telah menginspirasi penulis dengan motivasi - motivasi dan ilmunya.
4. Ayah dan ibu, serta keluarga besar yang senantiasa mendo'akan penulis sehingga penyelesaian tugas akhir ini berjalan dengan lancar.
5. Keluarga besar matematika 2015 yang telah mendukung dan memotivasi penulis untuk dapat menyelesaikan penelitian ini dengan cepat.

6. Keluarga sahabat masjid dan pengurus harian laboratorium agama.
7. Keluarga Forum kajian islam dan sains teknologi yang dipenuhi orang - orang jenius dan luar biasa, sehingga menginspirasi penulis untuk terus menuntut ilmu.
8. Keluarga KKN 96 kelompok 04.
9. Special thanks to, Sesepuh yang banyak memberikan nasihat, Kang Anshori, Mas hafid, Mbak Ernita, dan Mbak Inayah. Pejuang wisuda Mei, Aqil, Apriyanti, Umniyyah, Aas, Farahdina, Fransiska, dan Hana. Rekan seperjuangan Aljabar, Bardiati, Ma'rifah, Adji, Yahadiyana, dan Darmawan.

Penulis menyadari bahwa masih banyak kekurangan yang terdapat pada tugas akhir ini. Oleh karena itu, penulis berharap pembaca dapat mengirimkan kritik dan sarannya ke *nlawfalr@gmail.com*. Semoga penelitian ini bisa bermanfaat untuk pembaca, terimakasih

Wassalamu'alaikum, wr.wb.

Yogyakarta, 13 Februari 2019

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMBANG

$x \in A$	: x anggota A
$A \subseteq X$	: A himpunan bagian (*subset*) atau sama dengan X
$\mathbb{N}$	: himpunan semua asli
$\mathbb{Z}$	: himpunan semua bilangan bulat
$\mathbb{Z}^+$	: himpunan semua bilangan bulat positif
$\mathbb{R}$	:himpunan semua bilangan real
$\blacksquare$	: akhir suatu bukti
$\square$	: akhir suatu contoh
$\rightarrow$	: menuju
$\sum_{i=1}^{n} a_i$	: penjumlahan $a_1 + a_2 + \cdots + a_n$
$\prod_{i=1}^{n} a_i$	: perkalian $a_1 \cdot a_2 \cdot \cdots \cdot a_n$
$p \Rightarrow q$	: jika p maka q
$\Leftrightarrow$	: jika dan hanya jika
$x \leftarrow a$	: nilai a dimasukkan ke x
$\left(\bigcirc_{r=1}^{10} RF(K_r)\right)$	: Fungsi putaran algoritma Whirlpool
F_{2^n}	: Galois Field dengan karakteristik p berdimensi n
RnI	: Ring Faktor
$R[x]$	: Ring polinomial
$deg\ f(x)$	: Derajat tertinggi dari polinomial $f(x)$

INTISARI

Penerapan Fungsi Hash Kriptografi dalam Verifikasi Integritas Data Al-Qur'an Digital menggunakan Algoritma Whirlpool

Oleh

Naufal Rachmadan

15610004

Fungsi hash dalam kriptografi memiliki peran dalam autentikasi pesan. Fungsi hash dirancang untuk melindungi pesan yang sudah di hash, kembali ke teks aslinya. Salah satu kegunaan fungsi hash adalah *Message Intregrity Verification* atau Verifikasi Integritas Pesan. Metode yang dilakukan adalah dengan mengecek nilai hash pesan dan membandingkan dengan nilai hash yang seharusnya. Selain pesan, metode ini juga dapat digunakan dalam memverifikasi integritas suatu data komputer. Pada penelitian ini, akan dibahas bagaimana fungsi hash kriptografi digunakan dalam memverifikasi integritas data berupa ayat - ayat pada Al - Qur'an Digital. Algoritma Whirlpool adalah algoritma yang dipilih dalam penelitian ini karena Whirlpool memiliki efisiensi yang baik dalam melakukan *hashing*. Whirlpool mengubah pesan input menjadi sebuah nilai hash berukuran 512 bit. Hasil penelitian menunjukan bahwa Whirlpool dapat mendeteksi perubahan 1-bit data pada *database* Al-Qur'an Digital berukuran 1,26 MB atau 10,612,872-bit. Hal ini menunjukan bahwa Whirlpool adalah fungsi hash yang baik dalam memverifikasi integritas data berukuran 1 hingga 10,612,872-bit.

Kata Kunci: Kriptografi, Fungsi Hash, Al - Quran Digital, Integritas Data, Pemrograman Komputer, Whirlpool, *Advanced Encryption Standard*.

ABSTRACT

Implementation of Cryptographic Hash Function for Verification Data Integrity of Digital Holy Qur'an using Whirlpool Algorithm

By

Naufal Rachmadan

15610004

In cryptography, hash function have a role in message authentication. Hash function designed to be a function that can prevented hashes message, decrypted to the original message. One of the uses of hash function is Message Integrity Verification. The method used is by comparing hash value of message and hash value that should be. Furthermore, it can be used to verificates computer's data. In this work, Verification data integrity using cryptographic hash function are proposed to verificates data integrity of holy qur'an verses. Whirlpool Algorithm is choosen, because of its efficiency to hashing. Whirlpool produces a fixed length 512-bit string as output. The result show that Whirlpool can detected 1-bit changes on 1,26 MB or 10,612,872-bit data of database of Holy Qur'an. This shows that Whirlpool is a good hash function to verificates integrity of 1 to 10,612,872-bit data.

Kata Kunci : Cryptography, Hash Function, Digital Holy - Qur'an, Data Integrity, Computer Programming, Whirlpool, Advanced Encryption Standard.

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Di dalam islam, Al - Quran merupakan sebuah pedoman hidup yang membimbing setiap insan manusia mencapai kedamaian dan keselamatan. Al - Quran merupakan sebuah mahakarya dari Tuhan yang menciptakan seluruh alam jagat raya, yang sampai kapanpun akan menjadi penerang dalam kehidupan. Kehadirannya di alam ini sejak pertama kali dibawakan oleh sang pembawa wahyu terakhir, Rasulullah SAW, telah menyelamatkan banyak manusia dari kebodohan. Al - quran adalah cahaya, yang menerangi zaman yang gelap gulita menuju zaman yang terang penuh harapan.

Sebagai seorang muslim, adalah suatu kewajiban yang mutlak untuk mempelajari Al - Quran. Setiap ayat dan huruf yang dibaca, adalah suatu kebaikan yang akan memberikan keuntungan kepada yang membacanya, baik untuk kehidupannya di dunia, maupun di akhirat kelak. Membaca dan mempelajari makna dari ayat Al - Quran sama halnya dengan menerangi diri di sebuah lorong gelap dengan sebuah cahaya yang sangat terang benderang, menghangatkan diri dari musim dingin yang membekukan, melindungi diri dari kejahatan - kejahatan ghaib di dalam sebuah benteng yang sangat kokoh.

Perkembangan Al - Quran Digital di era modern ini adalah upaya pelestarian warisan islam yang harus tetap dijaga sampai kapanpun. Problematika yang

dihadapi umat muslim dalam menjaga kesucian Al - Quran dari dahulu hingga sekarang terus ada. Musailamah Al-Kazzab yang hidup di zaman Rasulullah adalah salah satu manusia yang pernah mencoba untuk membuat tandingan Al - Quran dengan membuat sebuah surat yang dianggapnya dapat menyaingi Al - Qur'an. Pada era modern ini, peperangan antara umat islam dengan musuh - musuhnya tidak lagi secara fisik, melainkan fikiran dan mental. Banyak musuh - musuh islam yang menyebarkan hadist - hadist palsu, penafsiran ayat Al - Quran tanpa ilmu, hingga pemalsuan ayat Al - Quran. Hal ini dilakukan sebagai upaya untuk memecah belah umat islam.

Di era digital ini, internet adalah sebuah dunia semu yang dihuni oleh hampir seluruh manusia di dunia. Mulai dari komunikasi, hingga transaksi semuanya dilakukan disana. Kemampuannya dalam mengumpulkan informasi, membuat masyarakat semakin mudah untuk mendapatkan informasi apapun dan dari mana pun sehingga alangkah bijak jika masyarakat bisa menyeleksi mana informasi yang benar dan yang tidak benar, seperti yang tertulis dalam Al - Quran surah Al - Hujurat ayat 6:

يَا أَيُّهَا الَّذِينَ آمَنُوا إِنْ جَاءَكُمْ فَاسِقٌ بِنَبَإٍ فَتَبَيَّنُوا أَنْ تُصِيبُوا قَوْمًا بِجَهَالَةٍ فَتُصْبِحُوا عَلَىٰ مَا فَعَلْتُمْ نَادِمِينَ

"Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu." **[Al-Hujurat : 6]**

Munculnya Al - Quran Digital adalah sebuah inovasi kreatif yang dikembangkan ilmuwan muslim untuk memudahkan umat muslim membaca dan mempelajari Al - Quran secara efisien. Al - Quran Digital memiliki berbagai macam jenis, Ada yang berbasis aplikasi komputer (*Software*), website, dan aplikasi ponsel.

Al - Qur'an Digital juga berperan dalam proses produksi Al - Qur'an Cetakan. Namun demikian, kesalahan penulisan ayat - ayat Al - Qur'an tetap saja tidak bisa terhindari sepenuhnya. Dikutip dari sebuah berita pada 25 Mei 2017 di website kementrian agama RI, telah ditemukan sebuah Al - Qur'an cetakan yang keliru dalam hal penempatan ayat pada surah Al - Maidah.

Gambar 1.1 Kesalahan penempatan ayat surah Al - Maidah pada Al - Qur'an cetakan

Setelah pihak kementrian meminta konfirmasi terkait hal tersebut, pihak penerbit menjelaskan kekeliruan pada penempatan halaman surah Al - Maidah ayat 51-57 disebabkan oleh *Human Error*. Pihak penerbit terpaksa menarik seluruh terbitan

tersebut, dan telah meminta maaf kepada seluruh masyarakat indonesia terkait hal tersebut.

Kesalahan - kesalahan serupa yang terjadi pada Al - Qur'an Digital, dapat dideteksi dengan metode verifikasi integritas data menggunakan fungsi hash. Pada penelitian ini, akan dijelaskan bagaimana cara mendeteksi adanya perubahan data pada suatu *database* Al - Qur'an Digital, seperti perubahan harakat, lafadz, hingga ayat atau halaman yang tertukar atau hilang. Proses verifikasi dilakukan menggunakan algoritma kriptografi fungsi hash yang dinamakan Whirlpool. Whirlpool mengubah suatu *data input* menjadi *data ouput* dengan panjang tetap, yaitu 512-bit.

1.2. Batasan Masalah

Penelitian ini bertujuan untuk mempelajari struktur aljabar pada Algoritma fungsi hash Whirlpool serta menjelaskan cara kerja dan penerapannya dalam melakukan proses Hashing dan memverifikasi integritas data Al - Quran Digital.

1.3. Rumusan Masalah

1. Bagaimana struktur aljabar dan fungsi - fungsi pada algoritma Whirlpool?
2. Bagaimana cara kerja algoritma WHIRLPOOL dalam melakukan proses *Hashing* terhadap teks berbasis ASCII dan UNICODE?
3. Bagaimana proses verifikasi integritas data Al - Quran Digital menggunakan fungsi hash kriptografi?

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah mempelajari bagaimana proses verifikasi integritas data menggunakan fungsi hash kriptografi.

1.5. Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Mengetahui struktur aljabar serta fungsi - fungsi pada algoritma Whirlpool.
2. Mengetahui cara kerja Algoritma WHIRLPOOL dalam melakukan proses hashing terhadap teks berbasis ASCII dan UNICODE.
3. Mengetahui proses verifikasi integritas data Al - Quran Digital menggunakan fungsi hash kriptografi.

1.6. Tinjauan Pustaka

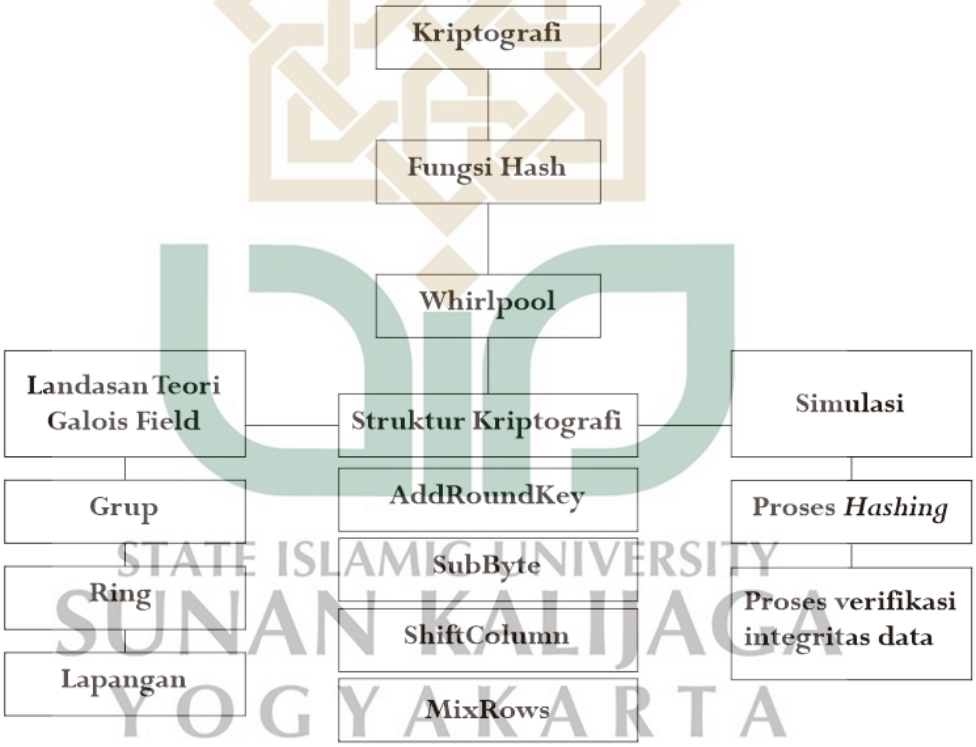
Penelitian ini terinspirasi dari penelitian Mishal Almazrooie, dkk. yang berjudul, ***Integrity verification for digital Holy Quran verses using cryptographic hash function and compression***. Mishal, dkk mencoba untuk membandingkan kinerja 2 algoritma fungsi hash kriptografi, yaitu SHA-256 dan RIPEMD-160 dalam melakukan verifikasi data.

Penelitian tersebut memiliki 3 kajian utama, pertama adalah meneliti tingkat sensitivitas 2 algoritma terhadap perubahan data sekecil apapun. Kedua adalah mengecek apakah terjadi duplikasi data pada Al - Quran yang sudah menjadi nilai hash. Terakhir adalah menganalisa tingkat keamanan dari data - data pada tabel hash yang sudah melalui proses Hashing. Selain menggunakan 2 Algoritma di atas, penelitian tersebut juga membandingkan efesiensi waktu yang diperlukan untuk mengkompresi Al - Quran Digital melalui aplikasi LZW, BZ2, GZ, RAR, dan ZIP.

Selain jurnal diatas, digunakan juga beberapa referensi pendukung diantaranya, *Introduction to Abstract Algebra* Karya D.S Malik,dkk. Buku ini menjelaskan teori-teori dasar aljabar abstrak. Kemudian *Elementary Linear Algebra* edisi kesembilan karya Howard Anton dan Chris Rorres. Buku ini membahas mengenai dasar - dasar Aljabar Linier. Buku *Cryptography and Network Security* edisi kelima karya William Stallings juga menjadi sumber rujukan pada penelitan ini. Buku ini membahas lebih dalam mengenai Kriptografi.

1.7. Metode Penelitian

Metode yang digunakan adalah studi literatur. Alur penelitian dimulai dari mengkaji sistematika algoritma Whirlpool, kemudian akan dijelaskan bagaimana struktur aljabar dan fungsi - fungsi yang ada dalam Whirlpool. Setelah itu, akan dibahas bagaimana cara kerja algoritma Whirlpool dalam melakukan proses *Hashing* suatu input data. Terakhir adalah menjelaskan proses verifikasi data Al - Quran Digital dengan menggunakan program komputer.



Gambar 1.2 Alur Penelitian

1.8. Sistematika Penulisan

Pembahasan dalam penelitian ini terbagi menjadi lima bab, yaitu :

1. BAB I (Pendahuluan) : Bab ini membahas mengenai Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Tinjauan Pustaka, Metode Penelitian dan Sistematika Penulisan.
2. BAB II (Dasar Teori) : Bab ini membahas mengenai dasar - dasar teori aljabar abstrak yang merupakan struktur dari algoritma Whirlpool.
3. BAB III (Verifikasi Integritas Data Menggunakan Fungsi Hash Kriptografi) : Bab ini membahas tentang kriptografi, fungsi hash dan struktur algoritma Whirlpool serta fungsi - fungsi yang ada di dalamnya.
4. BAB IV (Implementasi dan Uji Coba) : Bab ini membahas penerapan algoritma Whirlpool dalam melakukan proses *Hashing* terhadap teks berbasis ASCII dan UNICODE kemudian memaparkan bagaimana cara memverifikasi integritas data Al - Quran Digital dengan fungsi hash menggunakan program komputer.
5. BAB V (Penutup) : Bab ini berisi kesimpulan penelitian dan saran peneliti terhadap perkembangan penelitian ini untuk ke depannya.

BAB II

DASAR TEORI

Pada bab ini, akan dibahas teori - teori dasar aljabar yang merupakan fondasi utama dalam kriptografi. Teori lapangan hingga (atau *Finite Field*) adalah kajian dari aljabar abstrak, yang memiliki peranan penting dalam kriptografi. Teori lapangan hingga merupakan derivatif dari teori *Field*, *Ring* dan *Group*.

Komputasi dan aritmatika yang digunakan dalam Whirlpool, merupakan implementasi dari studi teori bilangan. Teori Bilangan adalah cabang dari matematika murni yang membahas tentang sifat - sifat bilangan bulat.

2.1. *Group* (Grup)

Grup merupakan suatu sistem matematika yang terdiri dari satu himpunan tak kosong, dan satu operasi biner.

Definisi 2.1.1 (Malik,2007) *Suatu Grup adalah pasangan berurutan* $(G, *)$*, dimana G adalah suatu himpunan tak kosong dan* $*$ *merupakan suatu operasi biner di G dan memenuhi sifat - sifat*

1. $(\forall a, b \in G)\ a * b \in G$.
2. $(\forall a, b, c \in G)\ a * (b * c) = (a * b) * c$.
3. $(\forall a \in G)(\exists e \in G)\ a * e = e * a = a\ \ (sifat\ identitas)$.
4. $(\forall a \in G)(\exists a^{-1} \in G)\ a * a^{-1} = a^{-1} * a = e\ \ (sifat\ invers)$.

Operasi * merupakan simbol umum yang menotasikan suatu operasi, misalnya penjumlahan, perkalian, dst. Himpunan bilangan bulat $(\mathbb{Z})$ dan bilangan real $(\mathbb{R})$ merupakan salah satu contoh dari grup.

Contoh 2.1.2 Himpunan bilangan bulat dengan operasi biner $(+)$ merupakan Grup karena memenuhi aksioma Grup, yaitu

1. **Tertutup** $(Well\ Defined)$

   Diambil sembarang $a, b \in \mathbb{Z}$, maka

   $$a * b = a + b \in \mathbb{Z}.$$

   Terbukti $(\mathbb{Z}, +)$ bersifat tertutup $(well\ defined)$

2. **Memenuhi sifat Asosiatif**

   Diambil sembarang $a, b, c \in \mathbb{Z}$, maka

   $$\begin{aligned} a * (b * c) &= a + (b + c) \\ &= (a + b + c) \\ &= (a + b) + c \\ &= (a * b) * c. \end{aligned}$$

   Terbukti bahwa $\mathbb{Z}$ memenuhi sifat asosiatif.

3. **Memiliki elemen identitas**

   Diambil suatu $a \in \mathbb{Z}$, maka terdapat $e = 0 \in \mathbb{Z}$, sedemikian hingga

   $$a * e = a + 0 = a = 0 + a = e + a.$$

   Terbukti bahwa $\mathbb{Z}$ memiliki elemen identitas yaitu 0.

4. **Memiliki Invers**

   Diambil suatu $a \in \mathbb{Z}$, maka terdapat $-a \in \mathbb{Z}$, sedemikian hingga

$$a * (a^{-1}) = a + (-a) = 0 = (-a) + a = (a^{-1}) * a.$$

   Terbukti bahwa suatu elemen di $\mathbb{Z}$ memiliki elemen invers di $\mathbb{Z}$, yaitu $-a$.

Dari pembuktian 1 sampai 4, terbukti bahwa $\mathbb{Z}$ merupakan Grup atas operasi penjumlahan. □

$(G, +)$ disebut ***Grup Abelian*** apabila memenuhi sifat komutatif terhadap penjumlahan. Ada banyak jenis dari Grup, misalnya Grup Siklik dan Grup Permutasi. Berikut ini merupakan defisini dari Grup Permutasi dan Grup Siklik.

Definisi 2.1.3 (Malik,2007) *Suatu Grup* $(G, *)$ *disebut* ***Grup Permutasi*** *pada suatu himpunan tak kosong X jika setiap elemen di G adalah permutasi dari C dan operasi* $*$ *adalah komposisi dari dua fungsi.*

Permutasi merupakan penyusunan suatu kumpulan objek yang terurut, menjadi kumpulan objek dengan urutan yang berbeda. Permutasi dapat dilakukan pada objek-objek berupa angka maupun huruf. Berikut ini adalah contoh dari permutasi.

Contoh 2.1.4 Diberikan $n = 7$ dan α, β adalah dua permutasi di I_7 dengan definisi

$$\alpha = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 4 & 6 & 7 & 2 & 5 \end{pmatrix} \; dan \quad \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 4 & 6 & 7 & 2 & 5 \end{pmatrix}$$

Berdasarkan definisi komposisi fungsi,

$$(\alpha \circ \beta)(i) = \alpha(\beta(i)).$$

sehingga,

$$\begin{aligned}
(i=1) &\Rightarrow 1 \xrightarrow{\beta} 2 \xrightarrow{\alpha} 3 & (i=5) &\Rightarrow 5 \xrightarrow{\beta} 7 \xrightarrow{\alpha} 5\\
(i=2) &\Rightarrow 2 \xrightarrow{\beta} 5 \xrightarrow{\alpha} 7 & (i=6) &\Rightarrow 6 \xrightarrow{\beta} 6 \xrightarrow{\alpha} 2\\
(i=3) &\Rightarrow 3 \xrightarrow{\beta} 3 \xrightarrow{\alpha} 4 & (i=7) &\Rightarrow 7 \xrightarrow{\beta} 4 \xrightarrow{\alpha} 6\\
(i=4) &\Rightarrow 4 \xrightarrow{\beta} 1 \xrightarrow{\alpha} 1
\end{aligned}$$

diperoleh,

$$\alpha \circ \beta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 7 & 4 & 1 & 5 & 2 & 6 \end{pmatrix}$$

Definisi 2.1.5 (Malik, 2007) *Suatu Grup* $(G, *)$ *disebut* ***Grup Siklik*** *jika terdapat* $a \in G$*, sedemikian hingga*

$$G =< a > .$$

$< a >$ *merupakan suatu himpunan* $\{na | n \in \mathbb{Z}\}$*. Jika diberikan suatu Grup siklik* $G =< a >$ *dan* $b, c \in G$*, maka* $b = a^n$ *dan* $c = a^m$ *untuk suatu* $n, m \in \mathbb{Z}$*.*

Contoh 2.1.6 Grup $(\mathbb{Z}, +)$ adalah sebuah Grup siklik karena untuk sembarang elemen di $\mathbb{Z}$, $a = na$, $n \in \mathbb{Z}$ dan $a = 1$ sehingga

$$\mathbb{Z} =< 1 > .$$

Secara sederhana, Grup siklik dapat didefinisikan sebagai suatu Grup, dimana setiap elemen dalam Grup tersebut, dapat dibangun oleh 1 elemen dalam Grup tersebut. Elemen pembangun tersebut disebut *Generator*.

Definisi 2.1.7 (Malik,2007) *Diberikan* $(G, *)$ *suatu Grup dan* $a \in G$*. Jika terdapat suatu bilangan positif* n *sehingga* $a^n = e$*, maka bilangan positif terkecil* n *disebut dengan* ***order*** *dari* a*. Jika tidak memiliki bilangan positif* n*, maka* a *disebut* ***order tak hingga****.*

Definisi ini menyebutkan bahwa suatu elemen dalam Grup, apabila terdapat bilangan positif n sehingga $a^n = e$ dimana e merupakan identitas dari Grup, maka n adalah $order$ dari a. Lain halnya dengan Order dari Grup. Order dari suatu Grup didefinisikan sebagai kardinalitas dari Grup, yaitu jumlah elemen yang ada dalam Grup.

Definisi 2.1.8 (Malik , 2007) *Diberikan* $(G, *)$ *dan* $(G_1, *_1)$ *adalah* **Grup** *dan* f *suatu fungsi dari* G *ke* G_1. f *disebut* ***homomorfisma*** G *ke* G_1 *apabila* **untuk semua** $b \in G$, *berlaku:*

$$f(a * b) = f(a) *_1 f(b).$$

Contoh 2.1.9 Didefinisikan suatu fungsi f dari $(\mathbb{Z}, +)$ ke $(\mathbb{Z}_n, +_n)$ yaitu:

$$f(a) = [a], untuk\ semua\ a \in \mathbb{Z}$$

akan ditunjukan bahwa f merupakan homomorfisma. Ambil sembarang elemen $a, b \in \mathbb{Z}$ maka:

$$f(a + b) = [a + b] = [a] +_n [b] = f(a) +_n f(b).$$

Terbukti bahwa f merupakan fungsi homomorfisma.

Homomorfisma merupakan fungsi yang dapat mempertahankan struktur aljabarnya. Homomorphisma diambil dari bahasa yunani, yaitu (*Homos*) yang berarti ”sama”, dan (*morphe*) yang berarti ”bentuk”. Ada beberapa istilah untuk menyebutkan fungsi homomorfisma, berikut ini adalah istilah - istilah lain dari homomorfisma.

Definisi 2.1.10 (Malik , 2007) *Diberikan G dan G_1 adalah Grup. Suatu Homomorfisma $f : G \to G_1$ dikatakan* ***Epimorfisma*** *jika f adalah fungsi pada (***onto***) dan f dikatakan* ***Monomorfisma*** *jika f adalah fungsi satu-satu (***one-one***). Jika terdapat suatu Epimorfisma f dari G ke G_1, maka G_1 disebut dengan* ***Image*** *dari G.*

Definisi 2.1.11 (Malik, 2007) *Suatu Homomorfisma f dari $G \to G_1$ dikatakan* ***Isomorfisma*** *dari $G \to G_1$, apabila f suatu fungsi Monomorfisma dan Epimorfisma, dinotasikan $G \cong G_1$ (G dan G_1 isomorfis). Suatu isomorfisma dari suatu Grup G ke G disebut* ***Automorfisma****.*

2.2. *Ring* (Gelanggang)

Ring atau Gelanggang merupakan studi lanjutan dari teori Grup. Seperti halnya dengan Grup, Ring mengkaji tentang sifat-sifat pada suatu sistem matematika. Namun perbedaannya, Ring memiliki operasi biner tambahan yang disebut dengan perkalian (*Multiplication*).

Definisi 2.2.1 (Malik, 2007) *Suatu Ring adalah 3-pasangan berurutan* $(G, +, \cdot)$ *dimana* R *suatu himpunan tak kosong dan (+),(*$\cdot$*) adalah dua operasi biner di* R *yang memenuhi aksioma aksioma*

1. $(\forall a, b \in R)\ a + b \in G.$
2. $(\forall a, b \in R)\ a + b = b + a.$
3. $(\forall a, b, c \in R)\ a + (b + c) = (a + b) + c.$
4. $(\forall a \in R)(\exists e \in G)\ a + e = e + a = a\ \ (sifat\ identitas).$
5. $(\forall a \in R)(\exists a^{-1} \in G)\ a * a^{-1} = a^{-1} * a = e\ \ (sifat\ invers).$
6. $(\forall a, b, c \in R)\ a \cdot (b \cdot c) = (a \cdot b) \cdot c.$
7. $(\forall a, b, c \in R)\ a \cdot (b + c) = (a \cdot b) + (a \cdot c).$
8. $(\forall a, b, c \in R)\ (b + c) \cdot a = (b \cdot a) + (c \cdot a).$

Berikut ini adalah contoh himpunan yang merupakan Ring.

Contoh 2.2.2 Diberikan $\mathbb{Z}_n$ suatu himpunan bilangan bulat modulo n, yaitu

$$\mathbb{Z}_n = \{0, 1, 2, ..., n-1\}, a = a\ mod\ n,\ a, n \in \mathbb{Z}$$

Akan dibuktikan bahwa $\mathbb{Z}_n$ adalah sebuah Ring.

1. **Tertutup** $(Well\ Defined)$

   Ambil sembarang $a, b \in \mathbb{Z}_n$, maka

   $$a + b = a\ mod\ n + b\ mod\ n = a + b\ mod\ n \in \mathbb{Z}.$$

   Terbukti $\mathbb{Z}_n$ bersifat tertutup $(well\ defined)$

2. **Memenuhi sifat Asosiatif**

   Ambil sembarang $a, b, c \in \mathbb{Z}_n$, maka

   $$\begin{aligned} a + (b + c) &= a + (b + c) \\ &= a\ mod\ n + (b\ mod\ n + c\ mod\ n) \\ &= (a + (b + c))\ mod\ n \\ &= ((a + b) + c)\ mod\ n \\ &= ((a\ mod\ n) + (b\ mod\ n)) + (c\ mod\ n). \end{aligned}$$

   Terbukti bahwa $\mathbb{Z}_n$ memenuhi sifat asosiatif.

3. **Memiliki elemen identitas**

   Ambil suatu $a \in \mathbb{Z}_n$, maka terdapat $e = 0 \in \mathbb{Z}_n$, sedemikian hingga

   $$a * e = a + 0 = a\ mod\ n + 0\ mod\ n = (a + 0)\ mod\ n = a\ mod\ n.$$

   Terbukti bahwa $\mathbb{Z}_n$ memiliki elemen identitas yaitu 0.

4. **Memiliki Invers**

   Ambil suatu $a \in \mathbb{Z}_n$, maka terdapat $b \in \mathbb{Z}_n$, sedemikian hingga

   $$\begin{aligned} a + b = a\ mod\ n + b\ mod\ n &= 0 \\ a + b\ mod\ n &= 0 \\ b &= n - a. \end{aligned}$$

   Terbukti bahwa suatu elemen di $\mathbb{Z}$ memiliki elemen invers di $\mathbb{Z}_n$.

5. **Memenuhi sifat Komutatif**

   Ambil suatu $a, b \in \mathbb{Z}_n$, maka

   $$a+b = a \bmod n + b \bmod n = (a+b) \bmod n = (b+a) \bmod n = b \bmod n + a \bmod n.$$

   Terbukti bahwa $\mathbb{Z}_n$ memenuhi sifat Komutatif.

   Selanjutnya adalah aksioma terhadap operasi perkalian $(\cdot)$.

6. **Tertutup** $(Well\ Defined)$

   Ambil sembarang $a, b \in \mathbb{Z}_n$, maka

   $$a \cdot b = a \bmod n \cdot b \bmod n = a \cdot b \bmod n \in \mathbb{Z}.$$

   Terbukti $\mathbb{Z}_n$ bersifat tertutup $(well\ defined)$ terhadap perkalian

7. **Memenuhi sifat Asosiatif**

   Ambil sembarang $a, b, c \in \mathbb{Z}_n$, maka

   $$\begin{aligned} a \cdot (b \cdot c) &= a \cdot (b \cdot c) \\ &= a \bmod n \cdot (b \bmod n \cdot c \bmod n) \\ &= a \cdot (b \cdot c) \bmod n \\ &= (a \cdot b) \cdot c) \bmod n \\ &= (a \bmod n \cdot b \bmod n) \cdot c \bmod n. \end{aligned}$$

   Terbukti bahwa $\mathbb{Z}_n$ memenuhi sifat asosiatif terhadap perkalian.

8. **Memenuhi sifat Distributif (Kanan)**

   Ambil sembarang $a, b, c \in \mathbb{Z}_n$, maka

   $$\begin{aligned} ((a+b) \cdot c) &= (a+b) \cdot c \bmod n \\ &= (ca) + (cb) \bmod n \\ &= (c \bmod n) \cdot (a \bmod n) + (c \bmod n) \cdot (b \bmod n). \end{aligned}$$

Terbukti bahwa $\mathbb{Z}_n$ memenuhi sifat distributif (kanan) terhadap perkalian.

9. **Memenuhi sifat Distributif (Kiri)**

   Ambil sembarang $a, b, c \in \mathbb{Z}_n$, maka

$$\begin{aligned}((a \cdot (b+c)) &= [a \cdot (b+c)] \; mod \; n \\ &= (ab) + (ac) \; mod \; n \\ &= (a \; mod \; n \cdot b \; mod \; n + a \; mod \; n \cdot c \; mod \; n).\end{aligned}$$

   Terbukti bahwa $\mathbb{Z}_n$ memenuhi sifat distributif (kiri) terhadap perkalian.

Dari pembuktian 1 sampai 8, $\mathbb{Z}_n$ memenuhi seluruh aksioma Ring, sehingga terbukti bahwa $\mathbb{Z}_n$ adalah suatu Ring, yaitu $(\mathbb{Z}_n, +, \cdot)$. □

Pada pembahasan - pembahasan selanjutnya, Penulisan suatu Ring $(\mathbb{Z}_n, +, \cdot)$ akan dinotasikan dengan R agar mempermudah penulisan dan penjelasan.

Definisi 2.2.3 (Malik,2007) *Diberikan sebuah Ring R. Suatu elemen $e \in R$ disebut* ***elemen identitas*** *jika $ea = ae = a$ untuk semua $a \in R$. Jika R memiliki elemen identitas, maka R disebut dengan Ring dengan Elemen Satuan dan dinotasikan dengan 1.*

Contoh 2.2.4 1 merupakan elemen identitas di $\mathbb{Z}_2$, sebab

$$Z_2 = \{0, 1\}.$$

untuk $a = 0$, maka $a \cdot e = 0 \cdot 1 = 0 = a$

untuk $a = 1$, maka $a \cdot e = 1 \cdot 1 = 1 = a$

sehingga Z_2 merupakan Ring dengan elemen satuan.

Identitas dalam suatu Ring sangat berperan ketika hendak mencari elemen invertibel. Elemen invertibel dalam suatu Ring, sangat penting untuk diidentifikasi karena suatu sistem kriptografi, harus berlangsung pada suatu Lapangan agar proses enkripsi dan dekripsi dapat bekerja secara optimal. Berikut ini merupakan definisi dari elemen invertibel.

Definisi 2.2.5 (Malik, 2007) *Diberikan suatu Ring R dengan elemen satuan. Suatu elemen $u \in R$ disebut dengan **Unit** (atau elemen invertibel), Jika terdapat suatu elemen $v \in R$ sedemikian hingga $uv = vu = 1$*

Contoh 2.2.6 1 dan 5 merupakan Unit dari Ring $\mathbb{Z}_6$, sebab

$$\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}.$$

untuk $u = 1$ dan $v = 1$, maka $u \cdot v = 1 \cdot 1 = 1$

untuk $u = 5$ dan $v = 5$, maka $u \cdot v = 5 \cdot 5 = 25 = 1$

Teorema 2.2.7 (Malik, 2007) *Diberikan R suatu Ring dengan elemen satuan dan T suatu himpunan seluruh unit dari R, maka*

1. $T \neq \varnothing$.
2. $0 \notin T$.
3. $ab \in T, untuk\ \ semua\ \ a, b \in T$.

Bukti.

1. Diketahui R adalah Ring dengan elemen satuan, maka $R \notin \varnothing$, karena ada $1 \in R$, dimana $1.1 = 1 = 1.1$, maka 1 adalah unit, sehingga $T \neq \varnothing$

2. Untuk sembarang $v \in R$,

$$0v = 0 \neq 1 \quad \Rightarrow \quad 0 \notin T.$$

3. Untuk suatu $a, b \in R$, maka ada $c, d \in R$ sedemikian hingga $ac = 1 = ca$ dan $bd = 1 = db$

$$(ab)(cd) = a(bd)c = a(1)c = ac = 1$$

$$(cd)(ab) = d(ac)b = d(1)b = db = 1.$$

Maka $(ab)(cd) = 1 = (cd)(ac)$ sehingga (ab) adalah Unit dan $(ab) \in T$.

■

Definisi 2.2.8 (Malik,2007) *Diberikan R suatu Ring dengan elemen satuan,*

1. *R disebut **Ring Pembagian (Division Ring)** jika setiap elemen tak nol dari R adalah Unit.*
2. *Ring Pembagian R yang komutatif disebut dengan **Lapangan**.*

Contoh 2.2.9 $\mathbb{Z}_2$ merupakan lapangan, karena setiap elemen tak nol di $\mathbb{Z}_2$ merupakan unit.

$$\mathbb{Z}_2 = \{0, 1\}.$$

untuk $u = 1$ dan $v = 1$, maka $u \cdot v = v \cdot u = 1 \cdot 1 = 1$

Perlu diperhatikan bahwa setiap Ring pembagian belum tentu merupakan Lapangan, sedangkan setiap Lapangan merupakan Ring pembagian. Berikut ini adalah contoh Ring yang bukan merupakan Lapangan.

Contoh 2.2.10 $\mathbb{Z}$ bukan lapangan, sebab untuk sembarang $a \in \mathbb{Z}$,

$$1 = aa^{-1} = a \cdot \frac{1}{a}.$$

$\frac{1}{a}$ bukan elemen dari $\mathbb{Z}$, maka setiap elemen di $\mathbb{Z}$ tidak memiliki invers kecuali 1 dan -1.

Definisi 2.2.11 (Malik, 2007) *Suatu elemen tak nol a pada suatu Ring R disebut* ***Pembagi Nol*** *jika terdapat $b \in R$ dimana $b \neq 0$ dan $ab = 0$ atau $ba = 0$.*

Dari definisi 2.2.8, maka suatu Ring R dikatakan Ring pembagian atau Lapangan, apabila R tidak memuat pembagi nol.

Contoh 2.2.12 2, 3, dan 4 adalah elemen pembagi nol di $\mathbb{Z}_6$, sebab

$$\text{untuk } a = 2 \text{ dan } b = 3, \text{ maka } a \cdot b = 2 \cdot 3 = 6 = 0$$
$$\text{untuk } a = 4 \text{ dan } b = 3, \text{ maka } a \cdot b = 4 \cdot 3 = 12 = 0$$

Ring $(\mathbb{Z})$, merupakan Ring komutatif dengan elemen satuan yang tidak memuat pembagian nol, dan juga bukan suatu Lapangan. Untuk menyebut Ring seperti ini, muncul sebuah gagasan dalam teori Ring yang disebut Daerah Integral.

Definisi 2.2.13 (Malik, 2007) *Diberikan R suatu Ring komutatif dengan elemen satuan. R disebut* ***Daerah Integral*** *jika R tidak memiliki Pembagi Nol*

Contoh 2.2.14 $\mathbb{Z}$ merupakan daerah integral, sebab untuk semua $(a \in \mathbb{Z})(\nexists b \neq 0)$ sehingga $ab = 0$.

Teorema 2.2.15 (Malik, 2007) *Suatu Ring komutatif Hingga dengan elemen yang lebih dari satu dan tidak memiliki pembagi nol adalah lapangan .*

Bukti. Diberikan suatu R dengan n elemen, yaitu $\{a_1, a_2, a_3, .., a_n\}$. Diambil sembarang $a \in R$ dengan $a \neq 0$. Untuk $aa_i \in R$, maka diperoleh

$$\{aa_1, aa_2, aa_3, .., aa_n\} \subseteq R.$$

Apabila $aa_i = aa_j$, maka $a_i = a_j$. Oleh karena itu, maka elemen - elemen $a_1, a_2, a_3, .., a_n$ pasti berbeda - beda. Selanjutnya diketahui bahwa R hanya

memiliki n elemen (karena R Ring berhingga), maka

$$R = \{aa_1, aa_2, aa_3, .., aa_n\}.$$

Hal ini mengakibatkan $a \in \{aa_1, aa_2, aa_3, .., aa_n\}$, sehingga akan ada hasil perkalian dari suatu bilangan, misal aa_i , dimana $aa_i = a = aa_ia$

Selanjutnya akan ditunjukan bahwa a_i merupakan identitas di R. Diambil sembarang elemen $b \in R$, maka $b \in \{aa_1, aa_2, aa_3, .., aa_n\}$ sehingga ada $a_j \in R, b = aa_j$. Dengan demikian,

$$\begin{aligned} ba_i &= a_ib \\ &= a_i(aa_j) \\ &= (a_ia)a_j \\ &= aa_j \\ &= b. \end{aligned}$$

Diperoleh bahwa a_i adalah identitas dari R, dinotasikan dengan 1.

Karena $1 \in \{aa_1, aa_2, aa_3, .., aa_n\}$, maka akan ada perkalian dari 2 buah bilangan, misal aa_k, dimana $aa_k = 1 = a_ka$. Karena hal ini, maka setiap elemen tak nol di R yang berhingga adalah Unit sehingga R merupakan Lapangan. ■

Akibat 2.2.16 (Malik,2007) *Setiap Daerah Integral Hingga merupakan Lapangan.*

Contoh 2.2.17 $\mathbb{Z}_2$ adalah Daerah Integral hingga karena $\mathbb{Z}$ merupakan Daerah Integral dan $\mathbb{Z}_2 \subseteq Z$. Daerah Integral Hingga $\mathbb{Z}_2$ juga merupakan Lapangan karena setiap elemen di $\mathbb{Z}_2$ merupakan Unit (dibuktikan pada contoh 2.2.9)

Akibat 2.2.18 (Malik, 2007) *Diberikan n suatu bilangan bulat positif. Maka $\mathbb{Z}_n$ adalah lapangan jika dan hanya jika n merupakan bilangan prima*

Pembahasan berikut ini adalah membahas suatu himpunan bagian dari suatu Ring R yang disebut dengan Subring. Pembahasan ini akan digunakan dalam menjelaskan konsep Ideal.

Definisi 2.2.19 (Malik, 2007) *Diberikan* $(R, +, \cdot)$ *dan* $R' \subseteq R$, *maka* $(R', +, \cdot)$ *dikatakan* ***Subring*** *dari* $(R, +, \cdot)$ *apabila,*

1. $(R', +, \cdot)$ *adalah Subgrup dari* $(R, +, \cdot)$.
2. *Untuk semua* $x, y \in R'$, $x \cdot y \in R$.

Jika R *dan* R' *adalah Lapangan, maka* R' *disebut* ***Subfield.***

Teorema 2.2.20 (Malik,2007) *Diberikan suatu Ring* R. *Suatu himpunan tak kosong* $R' \subseteq R$ *dikatakan Subring jika dan hanya jika*

1. *Untuk semua* $x, y \in R'$, $x - y \in R'$.
2. *Untuk semua* $x, y \in R'$, $xy \in R'$.

Bukti. Diasumsikan R' Subring dari R, maka untuk semua $x, y \in R'$,

$$x - y \ dan \ xy \in R'.$$

sebaliknya, asumsikan $x - y \in R'$ dan $xy \in R'$ untuk semua $x, y \in R'$.
maka, R' merupakan Subgrup karena $x - y \in R'$. Selanjutnya diketahui bahwa $xy \in R'$. Berdasarkan definisi 2.2.19, terbukti bahwa $(R', +, \cdot)$ merupakan Subring. ■

Konsep Ideal, dikembangkan oleh kummer pada tahun 1843, ketika mencari pembuktian yang efisien dari *"Fermat's Last Theorem"*. Bunyi teorema tersebut adalah "Jika n merupakan bilangan yang lebih besar dari 2, maka tidak ada bilangan positif x, y, z sedemikian hingga $x^n + y^n = z^n$.

Definisi 2.2.21 (Malik, 2007) *Diberikan suatu Ring R dan I suatu himpunan tak kosong subset dari R, maka*

1. *I disebut **Ideal Kiri** dari R jika untuk semua $a, b \in I$ dan $r \in R$,*

$$a - b \in I \ \ dan \ \ ra \in I.$$

2. *I disebut **Ideal Kanan** dari R jika untuk semua $a, b \in I$ dan $r \in R$,*

$$a - b \in I \ \ dan \ \ ar \in I.$$

3. *I disebut **Ideal** dari R jika I ideal kiri dan I ideal kanan dari R.*

Jika I adalah suatu Ideal, maka $(I, +)$ merupakan Subgrup dari $(R, +)$ dan I merupakan Subring dari R. jika R merupakan suatu Ring komutatif, maka I subset dari R merupakan Ideal kiri dan Ideal kanan, sehingga setiap $I \in R$ merupakan Ideal.

Contoh 2.2.22 Himpunan 0 subset dari R merupakan Ideal dari R, sebab

$$a = 0 \ \ dan \ \ b = 0, \ \ maka \ \ 0 - 0 = 0 \in I$$

$$a = 0 \ \ dan \ \ r \in R, \ \ maka \ \ ar = 0 = ra \in I.$$

Pembahasan berikut ini merupakan perpaduan antara Ring dan Ideal, yang membentuk suatu Ring baru disebut dengan Ring Faktor.

Definisi 2.2.23 (Malik, 2007) *Diberikan suatu Ring R dan Ideal I, $I \subseteq R$. Ring ${}^{R}/_{I}$ disebut dengan **Ring Faktor** dari R oleh I, yaitu*

$${}^{R}/_{I} = \{x + I | x \in R\} \quad dan \quad x + I = \{x + a | a \in I, x \in R\}.$$

Seperti halnya Grup, konsep tentang fungsi Homomorfisma juga berlaku di Ring. Berikut adalah definisi dan istilah - istilah dalam Homomorfisma Ring.

Definisi 2.2.24 (Malik, 2007) *Diberikan* $(R, +, \cdot)$ *dan* $(R', +, \cdot)$ *adalah suatu ring dan* f *adalah suatu fungsi dari* R *ke* R'. f *dikatakan* ***homomorfisma*** *dari* R *ke* R' *apabila:*

1. $f(a+b) = f(a) +' f(b)$
2. $f(a \cdot b) = f(a) \cdot' f(b)$

Untuk semua $a, b \in R$.

Suatu homomorfisma f *dari suatu ring* R *ke* R' *dikatakan:*

1. ***Monomorfisma*** *jika* f *adalah fungsi satu-satu.*
2. ***Epimorfisma*** *jika* f *adalah fungsi pada.*
3. ***Isomorfisma*** *jika* f *monomorfisma dan epimorfisma.*

Teorema 2.2.25 (Malik , 2007) *Diberikan sebuah Ring* R *dan* I *Ideal dari* R. *Didefinisikan sebuah pemetaan* $g : R \to {}^{R}/_{I}$ *dengan* $g(a) = a + I$ *untuk* $a \in R$, *maka* g *adalah suatu Homomorfisma, disebut sebagai* ***Natural Homomorphism*** *(Homomorfisma Alami) dari* R *ke* ${}^{R}/_{I}$. *Dengan demikian, maka* $Ker(g) = I$

Untuk menunjukan pemetaan dari R ke R/I dengan definisi fungsi: $g(a) = a + I$ adalah suatu homomorfisma alami, maka perlu ditunjukan bahwa $ker(g) = I$.

Bukti. Diambil sembarang elemen $a, b \in R$, maka:

$$g(a+b) = (a+b) + I = (a+I) + (b+I) = g(a) + g(b)$$

dan

$$g(ab) = ab + I = (a+I)(b+I) = g(a)g(b).$$

Diperoleh g suatu Homomorfisma dari R ke R' Selanjutnya akan ditunjukan $ker(g) = I$,

$$e \in ker(g) \leftrightarrow g(a) = a + I \leftrightarrow a + I = e + I \leftrightarrow a^{-1}e \in I \leftrightarrow e \in I.$$

Diperoleh $ker(g) = I$. ■

Teorema 2.2.26 (Malik, 2007) *Diberikan suatu homomorfisma f dari Ring R ke R' dan Ideal I di R anggota $ker(f)$. Diberikan g suatu Homomorfisma Alami dari R ke ${}^{R}/_{I}$, maka terdapat suatu homomorfisma tunggal h di ${}^{R}/_{I}$ ke R', sehingga $f = h \circ g$. Dengan demikian, h adalah fungsi satu-satu jika dan hanya jika $I = ker(f)$*

Bukti. Didefinisikan suatu fungsi $h : {}^{R}/_{I} \to R'$ dengan $h(a + I) = f(a)$ untuk semua $a \in R$, maka:

$$h((a + I)(b + I)) = h(ab + I) = f(ab) = f(a)f(b) = h(a + I)h(b + I).$$

Teorema ini juga dikenal sebagai **Teorema Fundamental Homomorfisma** untuk Ring. ■

2.3. Ring Polinomial

Ring polinomial adalah Ring yang terdiri dari elemen R dan suatu elemen x. Teori Ring Polinomial banyak diterapkan pada studi Kriptografi dimana Ring Polinomial akan dipandang sebagai Ring polinomial atas Lapangan hingga.

Definisi 2.3.1 (Malik, 2007) *Untuk setiap Ring R, diberikan $R[x]$ yaitu sebuah himpunan atas barisan tak hingga $(a_0, a_1, a_2, ...)$ dimana,*

$$a_i \in R \;\; i = 0, 1, 2, ., n$$

*dan terdapat suatu bilangan tak negatif (bergantung pada $(a_0, a_1, a_2, ..., a_n)$ sehingga untuk semua bilangan $k \geq n, a_k = 0$. Elemen dari $R[x]$ disebut dengan **polinomial atas R**.*

Penjumlahan dan perkalian pada Ring Polinomial $R[x]$ didefinisikan

$$(a_0, a_1, a_2, ..., a_n) + (b_0, b_1, b_2, ..., b_n) = (a_0 + b_0, a_1 + b_1, ..., a_n + b_n).$$

$$(a_0, a_1, a_2, ..., a_n) \cdot (b_0, b_1, b_2, ..., b_n) = (c_0, c_1, c_2, ..., c_n).$$

dimana,

$$c_j = \sum_{i=0}^{j} a_i b_{j-1}, \;\; untuk \;\; j = 0, 1, 2, ...$$

Ring $R[x]$ disebut juga sebagai **Polinomial Ring** atau **Ring Polinomial atas R**. Elemen di Ring $R[x]$ dapat dinotasikan

$$a = ax^0 \;\; dinotasikan \;\; (a, 0, 0,).$$

$$ax = ax^1 \;\; dinotasikan \;\; (0, a, 0, 0,).$$

$$ax^2 = ax^2 \;\; dinotasikan \;\; (0, 0, a, 0, 0).$$

sehingga,

$$\begin{aligned}(a_0, a_1, a_2, ..., a_n) &= (a_0, 0, 0, ...) + (0, a_1, 0, ...) + ... + (0, 0, a_n, 0, 0, ...) \\ &= a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_nx^n\end{aligned}$$

Simbol x disebut dengan ***indeterminate*** atas R dan elemen $(a_0, a_1, a_2, ..., a_n)$ dari R disebut ***Koefisien*** dari $a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_nx^n$

Teorema 2.3.2 (Malik, 2007) *Untuk setiap Ring Polinomial $R[x]$, maka*

1. *Jika R suatu Ring Komutatif dengan elemen satuan, maka $R[x]$ juga merupakan Ring Komutatif dengan elemen satuan.*
2. *Jika R adalah Daerah Integral, maka $R[x]$ juga Daerah Integral.*

Bukti. Untuk poin 1, Diambil sembarang elemen $f(x), g(x), h(x) \in R[x]$, dengan

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ... + a_ix^i = \sum_{i=0}^{n} a_ix^i.$$

$$g(x) = b_0 + b_1x + b_2x^2 + b_3x^3 + ... + b_ix^i = \sum_{i=0}^{n} b_ix^i.$$

$$h(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + ... + c_ix^i = \sum_{i=0}^{n} c_ix^i.$$

Akan dibuktikan bahwa $R[x]$ adalah Ring komutatif dengan elemen satuan.

1. **Tertutup** $(Well\ Defined)$

$$\begin{aligned}f(x) + g(x) &= \sum_{i=0}^{n} a_ix^i + \sum_{i=0}^{n} b_ix^i \\ &= \sum_{i=0}^{n} (a_ix^i + b_ix^i) \\ &= \sum_{i=0}^{n} (a_i + b_i)x^i \in R[x].\end{aligned}$$

Terbukti bahwa $R[x]$ bersifat tertutup terhadap penjumlahan

2. **Memenuhi sifat Asosiatif**

$$\begin{aligned} f(x) + (g(x) + h(x)) &= \sum_{i=0}^{n} a_i x^i + \left(\sum_{i=0}^{n} b_i x^i + \sum_{i=0}^{n} c_i x^i \right) \\ &= \sum_{i=0}^{n} a_i x^i + \sum_{i=0}^{n} (b_i x_i + c_i x^i) \\ &= \sum_{i=0}^{n} (a_i + b_i x^i + c_i x^i) \\ &= \sum_{i=0}^{n} (a_i x^i + b_i x^i) + \sum_{i=0}^{n} c_i x^i \\ &= \left(\sum_{i=0}^{n} a_i x^i + \sum_{i=0}^{n} b_i x^i \right) + \sum_{i=0}^{n} c_i x^i \\ &= (f(x) + g(x)) + h(x). \end{aligned}$$

Terbukti bahwa penjumlahan atas Ring Polinomial $R[x]$ bersifat Asosiatif.

3. **Memiliki elemen identitas**

Untuk suatu $f(x) \in R[x]$, maka terdapat

$\phi(x) = 0, 0, 0, ..., 0 \in R[x]$, sedemikian hingga

$$\begin{aligned} f(x) + \phi(x) &= \sum_{i=0}^{n} a_i x^i + \sum_{i=0}^{n} 0_i x^i \\ &= \sum_{i=0}^{n} (a_i + 0_i) x^i \\ &= \sum_{i=0}^{n} a_i x^i \\ &= f(x). \end{aligned}$$

Terbukti bahwa $R[x]$ memiliki suatu elemen identitas ($\phi(x)$) sehingga untuk sembarang elemen di Ring Polinomial $R[x]$, $f(x) + \phi(x) = f(x) = \phi(x) + f(x)$.

4. **Memiliki Invers**

   Untuk suatu $f(x) \in R[x]$, maka terdapat

   $\psi(x) = (-a_0, -a_1x, -a_2x^2, ..., -a_nx^n) \in R[x]$, sedemikian hingga

$$\begin{aligned} f(x) + \psi(x) &= \sum_{i=0}^{n} a_i x^i + \sum_{i=0}^{n} -a_i x^i \\ &= \sum_{i=0}^{n} (a_i - a_i) x^i \\ &= \sum_{i=0}^{n} 0_i x^i \\ &= \phi(x). \end{aligned}$$

   Terbukti bahwa untuk sembarang $f(x) \in R[x]$, ada $\psi(x) \in R[x]$ sedemikian hingga, $f(x) + \psi(x) = \phi(x) = \psi(x) + f(x)$

5. **Memenuhi sifat Komutatif**

$$\begin{aligned} f(x) + g(x) &= \sum_{i=0}^{n} a_i x^i + \sum_{i=0}^{n} b_i x^i \\ &= \sum_{i=0}^{n} (a_i x^i + b_i x^i) \\ &= \sum_{i=0}^{n} (a_i + b_i) x^i \\ &= \sum_{i=0}^{n} (b_i + a_i) x^i \\ &= \sum_{i=0}^{n} (b_i x^i + a_i x^i) \\ &= \sum_{i=0}^{n} b_i x^i + \sum_{i=0}^{n} a_i x^i \\ &= g(x) + f(x). \end{aligned}$$

   Terbukti bahwa penjumlahan dalam $R[x]$, memenuhi sifat komutatif terhadap penjumlahan.

   Selanjutnya adalah aksioma terhadap operasi perkalian $(\cdot)$.

6. **Tertutup** $(Well\ Defined)$

$$\begin{aligned} f(x) \cdot g(x) &= \sum_{i=0}^{n} a_i x^i \cdot \sum_{i=0}^{n} b_i x^i \\ &= \sum_{i=0}^{n} (a_i x^i \cdot b_i x^i) \\ &= \sum_{j=0}^{n+m} c_i x^i \\ &= \sum_{j=0}^{i} a_j b_{i-j} \in R[x]. \end{aligned}$$

Terbukti bahwa $R[x]$ bersifat tertutup terhadap perkalian.

7. **Memenuhi sifat Asosiatif**

Untuk mempermudah, penulisan a_i mewakili koefisien dari x^i, berlaku juga untuk b dan c.

$$\begin{aligned} f(x) \cdot ((g(x) \cdot h(x)) &= \sum_{i+j=n} a_i (bc)_j \\ &= \sum_{i+j=n} a_i \sum_{u+v=j} b_u c_v \\ &= \sum_{i+j=n} \sum_{u+v=j} a_i (b_u c_v) \end{aligned}$$

Substitusi $j = u + v$, diperoleh

$$\sum_{i+u+v=n} a_i (b_u c_v)$$

R adalah Ring komutatif, maka sifat Asosiatif berlaku, sehingga

$$\begin{aligned}\sum_{i+u+v=n} a_i(b_u c_v) &= \sum_{i+u+v=n} (a_i b_u)c_v \\ &= \sum_{i+u=j} \sum_{v+j=n} (a_i b_u)c_v \\ &= \sum_{i+u=j} (a_i b_u) \sum_{v+j=n} c_v \\ &= \sum_{v+j=n} (ab)_j c_v \\ &= (f(x) \cdot g(x)) \cdot h(x).\end{aligned}$$

Terbukti bahwa $R[x]$ memenuhi sifat Asosiatif terhadap perkalian.

8. **Memenuhi sifat Distributif (Kanan)**

$$\begin{aligned}(f(x) + (g(x))) \cdot h(x) &= \sum_{i=0}^{n} a_i + (b)_i \cdot \sum_{j=0}^{m} c_j \\ &= \sum_{i+j=k} (a+b)_i c_j\end{aligned}$$

R adalah Ring komutatif, maka sifat Distributif berlaku, sehingga

$$\begin{aligned}\sum_{i+j=k} (a+b)_i c_j &= \sum_{i+j=k} (a_n c_j + b_n c_j) \\ &= \sum_{i+j=k} (a_n c_j) + \sum_{i+j=k} (b_n c_j) \\ &= ((g(x) \cdot h(x)) + (f(x) \cdot h(x)).\end{aligned}$$

Terbukti bahwa $R[x]$ memenuhi sifat Distributif terhadap perkalian.

9. **Memenuhi sifat Distributif (Kiri)**

$$\begin{aligned}f(x) \cdot ((g(x) + h(x)) &= \sum_{i=0}^{n} a_i \cdot \sum_{j=0}^{m} c_j + (b)_j \\ &= \sum_{i+j=k} (a)_i b + c_j\end{aligned}$$

R adalah Ring komutatif, maka sifat Distributif berlaku, sehingga

$$\begin{aligned}\sum_{i+j=k}(a)_i b + c_j &= \sum_{i+j=k}(a_n b_j + a_n c_j)\\ &= \sum_{i+j=k}(a_n b_j) + \sum_{i+j=k}(a_n c_j)\\ &= ((f(x)\cdot g(x)) + (f(x)\cdot h(x)).\end{aligned}$$

10. **Memenuhi sifat Komutatif**

$$\begin{aligned}f(x)\cdot g(x) &= \sum_{i=0}^{n} a_i \cdot \sum_{j=0}^{m} b_i\\ &= \sum_{i+j=k}(a)_i b_j\\ &= \sum_{i+j=k}(b)_i a_j\\ &= \sum_{j=0}^{m} b_i \cdot \sum_{i=0}^{n} a_i\\ &= g(x)\cdot f(x).\end{aligned}$$

Terbukti bahwa $R[x]$ memenuhi sifat Komutatif terhadap perkalian.

11. **Memiliki elemen Identitas**

Untuk suatu $f(x) \in R[x]$, maka terdapat $\varphi(x) = 1x^0 \in R[x]$, sedemikian hingga

$$\begin{aligned}f(x) + \varphi(x) &= \sum_{i=0}^{n}(a_i \cdot 1)x_i\\ &= \sum_{i=0}^{n}(a_i)x_i\\ &= f(x).\end{aligned}$$

Terbukti bahwa $R[x]$ memiliki suatu elemen identitas ($\varphi(x)$) sehingga untuk sembarang elemen di Ring Polinomial $R[x]$, $f(x)\varphi(x) = f(x) = \varphi(x)f(x)$

Dari pembuktian-pembuktian di atas, dapat dibuktikan bahwa $R[x]$ memenuhi kriteria Ring komutatif, dan $R[x]$ memiliki elemen identitas yang dinotasikan $\varphi(x)$, yaitu $\varphi(x) = 1x^0$. Berdasarkan 2.2.3), $R[x]$ merupakan Ring dengan elemen satuan, sehingga terbukti bahwa $R[x]$ merupakan Ring komutatif dengan elemen satuan.

Untuk membuktikan poin 2, diambil suatu elemen di $R[x]$, yaitu $f(x), g(x) \in R[x]$, dengan

$$f(x) = (a_0 + a_1x + a_2x^2 + ... + a_nx^n) = \sum_{i=0}^{n} (a_i)x^i$$

$$g(x) = (b_0 + b_1x + b_2x^2 + ... + b_nx^m) = \sum_{i=0}^{m} (b_i)x^i$$

adalah polinomial taknol di $R[x]$.

Diketahui bahwa R adalah suatu Daerah Integral, maka $R[x]$ merupakan suatu Ring komutatif dengan elemen satuan. sehingga ada $a_i, a_j \neq 0$, $a_{it} = 0$ dan $a_{jt} = 0$ untuk semua $t \geq 1$,

$$f(x)g(x) = (c_0 + c_1x + c_2x^2 + ... + c_{n+m}x^{n+m}).$$

$$c_{i+t} = (a_0b_{i+j} + a_1b_{i+j-1} + ... + a_ib_j + ... + a_{i+j}b_0) = a_ib_j \neq 0.$$

Diketahui bahwa R merupakan Daerah Integral, artinya koefisien dari $R[x]$ adalah elemen dari Daerah Integral, sehingga $f(x)g(x) \neq 0$. Oleh karena itu, dapat disimpulkan bahwa $R[x]$ merupakan Daerah Integral. ■

Dalam kriptografi, Ring polinomial adalah bentuk representasi dari deret satuan angka yang disebut dengan Bit. Operasi penjumlahan dan perkalian dari bit-bit dalam proses enkripsi dan dekripsi memenuhi sifat - sifat perkalian dan penjumlahan Ring polinomial. Suatu algoritma, biasanya membatasi bentuk

polinomial - polinomial hasil suatu perkalian, agar tetap dapat merepresentasikan suatu deret bit. Dasar dari konsep itu berkaitan dengan *Degree* atau Derajat Polinomial. Berikut ini adalah definisi dari *Degree* Polinomial.

Definisi 2.3.3 (Malik, 2007) *Diberikan suatu Ring* R.

Jika $f(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n$, $a_n \neq 0$ *adalah suatu polinomial di* $R[x]$, *maka* n *disebut dengan* ***Degree*** *dari* $f(x)$, *ditulis* $deg\ f(x)$ *dan* a_n *disebut dengan* ***Koefisien Utama dari f(x)***. *Jika* R *memiliki suatu identitas dan* $a_n = 1$, *maka* $f(x)$ *disebut dengan* ***Polinomial Monik***.

Polinomial dengan degree 0 di $R[x]$ disebut dengan **Skalar** atau **Polinomial Konstan**.

Contoh 2.3.4 Diberikan suatu polinomial dari $R[x]$, yaitu

$$f(x) = x^8 + 2x^7 + 5x^6 + 2x^5 + x^2 + 3.$$

Maka *degree* dari $f(x)$ adalah $deg\ f(x) = 8$ dan polinomial konstan dari $f(x)$ adalah 3

Teorema 2.3.5 (Malik, 2007) *Diberikan suatu Ring Polinomial* $R[x]$, *dan elemen* $f(x), g(x)$ *polinomial tak nol di* $R[x]$, *maka*

1. *Jika* $f(x)g(x) \neq 0$, *maka*

$$deg(f(x)g(x)) \leq deg\ f(x) + deg\ g(x)$$

.

2. *Jika* $f(x) + g(x) \neq 0$, *maka*

$$deg(f(x) + g(x)) \leq maxdeg\ f(x) + deg\ g(x)$$

.

Bukti. Untuk poin 1, Ambil sembarang $f(x), g(x)$ elemen dari $R[x]$, dimana

$$f(x) = (a_0 + a_1x + a_2x^2 + ... + a_nx^n) \in R[x].$$

$$g(x) = (b_0 + b_1x + b_2x^2 + ... + b_mx^m) \in R[x].$$

maka,

$$f(x)g(x) = (a_0b_0 + (a_0b_1)x + ... + a_nb_mx^{n+m}).$$

Jika $f(x)g(x) \neq 0$, maka setidaknya ada satu elemen yang tak nol. Jika $a_nb_m \neq 0$, maka

$$deg(f(x)g(x)) = n + m = deg\ f(x) + deg\ g(x).$$

Jika $a_nb_m = 0$, maka

$$deg(f(x)g(x)) < deg\ f(x) + deg\ g(x).$$

Selanjutnya untuk poin 2,

Jika $n > m$, maka

$$f(x) + g(x) = a_0b_0 + (a_1b_1)x + ... + (a_nb_m)x^m + ... + (a_nx^n).$$

$$deg\ (f(x) + g(x)) = max\ deg\ f(x), deg\ g(x).$$

Jika $n = m$, maka $f(x) + g(x) = a_0b_0 + (a_1b_1)x + ... + a_nb_nx^n$.

Untuk kasus R adalah Lapangan Hingga, maka ada kemungkinan $f(x) + g(x) = 0$ contoh: $[3x^2], [3x^2] \in \mathbb{Z}_6[x]$, maka $[3x^2] + [3x^2] = 0x^2 = 0$. Namun diketahui bahwa $f(x) + g(x) \neq 0$, sehingga

$$deg(f(x) + g(x)) \leq maxdeg\ f(x) + deg\ g(x).$$

untuk setiap $f(x), g(x) \in R[x]$ ■

Teorema berikut ini sangat penting dalam operasi perkalian polinomial pada suatu algoritma kriptografi. Teorema ini dikenal dengan ***Division Algorithm*** atau **Algoritma Pembagian**. Dengan menggunakan teorema ini, degree hasil perkalian dua buah polinomial dapat dipertahankan. Berikut adalah teorema dari Algoritma Pembagian.

Teorema 2.3.6 (Malik, 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan dan $f(x), g(x) \in R[x]$ dengan koefisien utama dari $g(x)$ adalah suatu Unit di R, maka terdapat dengan tunggal polinomial $q(x), r(x) \in R[x]$ sedemikian hingga*

$$f(x) = q(x)g(x) + r(x).$$

dimana $r(x) = 0$ atau $deg\ r(x) < deg\ g(x)$

Bukti. Jika $f(x) = 0$ atau $deg\ f(x) < deg\ g(x)$, maka

$$f(x) = q(x)g(x) + r(x) = (0)(g(x)) + f(x).$$

Diperoleh $q(x) = 0$ dan $r(x) = f(x)$.

Kemudian, jika $deg\ f(x) = deg\ g(x) = 0$, maka

$$f(x) = q(x)g(x) + r(x)$$
$$f(x) = q(x)g(x) + 0$$
$$f(x) = q(x)g(x)$$
$$q(x) = f(x)g(x)^{-1}$$

Jika $deg\ f(x) > deg\ g(x)$, dengan menggunakan pembuktian induksi, akan dibuktikan bahwa teorema 2.3.6 terpenuhi untuk semua polinomial dengan *degree* lebih kecil dari n.

Diambil sembarang elemen $f(x), g(x) \in R[x]$, yaitu

$$f(x) = (a_0, a_1x, a_2x^2, ..., a_nx^n) \neq R[x]$$

$$g(x) = (b_0, b_1x, b_2x^2, ..., b_mx^m) \neq R[x]$$

dimana $n \geq m$. Maka suatu polinomial,

$$f_1(x) = f(x) - (a_n{b_m}^{-1})x^{n-m}g(x).....(*)$$

memiliki *degree* lebih kecil dari n karena koefisien dari x^n adalah $a^n - \left(a_n{b_m}^{-1}\right)b_m = 0$. Akibatnya, berdasarkan hipotesis induksi, terdapat polinomial

$$q_1(x), r_1(x) \in R[x]$$

sedemikian hingga,

$$f_1(x) = q_1(x)g(x) + r_1(x).....(**)$$

dimana $r_1(x) = 0$ atau $deg\ r_1(x) < deg\ g(x)$.

Substitusi (*) dan (**), diperoleh

$$f(x) = (q_1(x) + (a_n{b_m}^{-1})x^{n-m})g(x) + r_1(x) = q(x)g(x) + r(x)$$

dimana $q(x) = (q_1(x) + (a_n{b_m}^{-1})x^{n-m})$ dan $r(x) = r_1(x)$.

Selanjutnya, akan ditunjukan ketunggalan dari $q(x)$ dan $r(x)$.

Diambil polinomial $q'(x), r'(x) \in R[x]$, yaitu

$$f(x) = q(x)g(x) + r(x) = q'(x)g(x) + r'(x)$$

dimana $r(x) = 0$ atau $deg\ r(x) < deg\ g(x)$, $r'(x) = 0$ atau $r(x) < deg\ g(x)$, maka

$$r(x) - r'(x) = (q'(x)g(x) + r(x)).$$

Diasumsikan $r(x) - r'(x) \neq 0$. Karena koefisien utama dari $g(x)$ adalah suatu unit, maka

$$deg\ ((r(x) - r'(x))g(x)) = deg\ (q'(x) - q(x)) + deg\ g(x) \geq deg\ g(x).$$

Hal ini mengakibatkan $deg(r(x) - r'(x)) \geq deg\ g(x)$ dan tidak mungkin terjadi, karena sebelumnya diketahui $degr(x), deg\ r('x) < deg\ g(x)$, sehingga $r(x) - r'(x) = 0$ atau $r(x) = r'(x)$. Oleh karena itu,

$$0 = (q'(x) - q(x))g(x).....(***)$$

Karena b_m adalah suatu unit, $deg\ (((q'(x) - q(x))g(x)) \geq 0$ kecuali jika $q'(x) - q(x) = 0$. Dari (***), $q'(x) - q(x) = 0$ maka dapat disimpulkan

$$deg\ r(x) < deg\ g(x)$$

. ■

Polinomial $q(x)$ disebut sebagai **Hasil Bagi** sedangkan $r(x)$ disebut sebagai **Sisa Pembagian**. Teorema sisa pembagian, dikenal juga sebagai Modulo. Ada beberapa cara untuk menemukan sisa bagi dari 2 buah polinomial, berikut ini adalah cara paling umum yang digunakan dalam mencari sisa pembagian.

Contoh 2.3.7 Berikut ini adalah contoh metode pembagian panjang (***long division***) suatu polinomial untuk mencari sisa bagi.

Diberikan $f(x) = 6x^4 + 6x^3 + 3x^2 - 3x + 1$, $d(x) = 2x^2 + 1$

$$\begin{array}{r|l} & 3X^2 - 3X \\ \hline 2X^2 + 1 & 6X^4 - 6X^3 + 3X^2 - 3X + 1 \\ & 6X^4 + 0X^3 + 3X^2 + 0X + 0 \\ \hline & -6X^3 + 0X^2 - 3X + 1 \\ & -6X^3 + 0X^2 - 3X + 0 \\ \hline & 1 \end{array}$$

Gambar 2.1 Teknik pembagian Panjang *Long Divison*

Diperoleh sisa pembagiannya adalah 1, sehingga

$$6x^4 + 6x^3 + 3x^2 - 3x + 1 = (3x^2 - 3x)(2x^2 + 1) + 1$$

Pembahasan - pembahasan berikut ini merupakan dasar dari pembahasan tentang Polinomial Iredusibel. Suatu polinomial $f(x)$ dikatakan iredusibel apabila $f(x)$ tidak memiliki akar. Berikut ini adalah definisi dari Akar Polinomial.

Definisi 2.3.8 (Malik , 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan dan $f(x) = a_0 + a_1 + a_2 + ... + a_n \in R[x]$. Untuk semua $r \in R$, didefinisikan*

$$f(r) = a_0, a_1 r, a_2 r^2 + ... + a_n r^n.$$

*Apabila $f(r) = 0$, maka r disebut sebagai **Akar** atau **Nol** dari $f(x)$.*

Definisi 2.3.9 (Malik , 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan dan $f(x), g(x) \in R[x]$ sedemikian hingga $g(x) \neq 0$. $g(x)$ disebut **Membagi** $f(x)$ atau $g(x)$ merupakan faktor dari $f(x)$ ditulis dengan $g(x)|f(x)$, jika terdapat suatu $q(x) \in R[x]$ sedemikian hingga $f(x) = q(x)g(x)$*

Berikut ini merupakan **Teorema Sisa Bagi** atau ***Remainder Theorem***. Teorema ini mengatakan bahwa $x - a$ merupakan pembagi dari $f(x)$ jika dan hanya jika $f(r) = 0$, utk suatu $a \in R$.

Teorema 2.3.10 (Malik, 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan. Untuk semua $f(x) \in R[x]$ dan $a \in R$, maka terdapat $q(x) \in R[x]$ sedemikian hingga*

$$f(x) = (x - a)q(x) + f(a).$$

Bukti. Dengan menggunakan algoritma pembagian dari 2.3.6, $x - a = g(x)$, maka terdapat tepat satu $q(x), r(x) \in R[x]$ sedemikian hingga $f(x) = (x-a)q(x)+r(x)$,

dimana $r(x) = 0$ atau $deg\ r(x) < 1$. Karena itu, $r(x)$ adalah suatu polinomial konstan, misal, $r(x) = d$. Substitusi $a = x$, diperoleh

$$f(a) = (a - a)q(x) + d = d.$$

■

Berikut ini merupakan **Teorema Faktorisasi** atau ***Factorization Theorem***. Teorema ini merupakan akibat dari teorema sisa bagi pada pembahasan sebelumnya.

Akibat 2.3.11 *Diberikan suatu Ring komutatif R dengan elemen satuan. Untuk semua $f(x) \in R[x]$ dan $a \in R$, $x - a$ membagi $f(x)$ jika dan hanya jika a adalah akar dari $f(x)$.*

Bukti. Asumsikan $(x - a)|f(x)$, maka ada $q(x) \in R[x]$, sehingga

$$f(x) = (x - a)q(x).$$

karena itu, $f(a) = (a - a)q(a) = 0$, sehingga a adalah akar dari $f(x)$.

Sebaliknya, jika (a) adalah akar dari $f(x)$, maka berdasarkan 2.3.8, $f(a) = 0$, sehingga $f(a) = 0 = (a - a)q(x)$. Akibatnya, $(x - a)|f(x)$. ■

2.4. Daerah Euclid

Euclidean Domain atau Daerah Euclid merupakan Daerah Integral yang memiliki fungsi euclid. Konsep ini merupakan generalisasi dari Algoritma Pembagian pada pembahasan 2.3.6. Konsep ini akan digunakan dalam pembahasan Daerah Ideal Utama dan Daerah Faktorisasi Tunggal. Dalam kriptografi, konsep ini digunakan dalam mencari faktor persekutuan terbesar antara 2 buah bilangan. Metode tersebut dikenal dengan *"Algoritma Euclid"*.

Definisi 2.4.1 (Malik , 2007) *Suatu daerah Euclid* $(E, +, \cdot, v)$ *adalah Daerah Integral* $(E, +, \cdot)$ *dengan suatu fungsi* $v : E\backslash\{0\} \to \mathbb{Z}^+$*, dimana*

1. *Untuk semua* $a, b \in E$ *dgn* $b \neq 0$ *maka terdapat* $q, r \in E$ *sedemikian hingga* $a = qb + r$*, dimana* $r = 0$ *atau* $v(r) < v(b)$.
2. *Untuk semua* $a, b \in E\backslash\{0\}, v(a) \leq v(ab)$.

v *disebut dengan* ***Nilai Euclid***

Contoh 2.4.2 Setiap Lapangan dapat dipandang sebagai Daerah Euclid dengan $v(a) = 1$ untuk semua $a \neq 0$ dengan $a = (ab^{-1})b + 0$ karena memenuhi aksioma Euclid, yaitu

1. Diketahui bahwa

   $a = (ab^{-1})b + 0$, karena F merupakan suatu Daerah Integral, yang mengakibatkan semua elemen di F memiliki invers terhadap perkalian, sehingga

   $$a = a((b^{-1}b) + 0) = a(1) + 0 = a$$

   memenuhi aksioma 1 Daerah Euclid.

2. Untuk sembarang $a \in F$, $v(a)$ didefinisikan

$$v(a) = 1$$

Maka, $v(ab) = 1$, sehingga $v(a) = v(ab)$ memenuhi aksioma 2 Daerah Euclid.

Terbukti bahwa suatu Lapangan dengan $v(a) = 1$ dapat dipandang sebagai Daerah Euclid.

Teorema 2.4.3 (Malik,2007) *Jika F adalah suatu Lapangan, maka Ring polinomial $F[x]$ adalah suatu Daerah Euclid.*

Bukti. Diketahui bahwa Setiap Daerah Integral terbatas adalah Lapangan. Kemudian berdasarkan Teorema 2.3.2, Jika R suatu Daerah Integral, maka $R[x]$ adalah Daerah Integral. Sehingga, jika F adalah Daerah Integral, maka $F[x]$ adalah suatu Daerah Integral.

Didefinisikan sebuah pengaitan,

$$v : F[x]\backslash\{0\} \to \mathbb{Z}^+$$

yaitu, $v(f(x)) = deg\ f(x)$, untuk semua $f(x) \in F[x]\backslash\{0\}$.

Diberikan $f(x), g(x) \in F[x]$ dan $g(x) \neq 0$. Karena $deg\ f(x) \geq 0$, $v(f(x)) \in \mathbb{Z}^+$ untuk semua $f(x) \in F[x]\backslash\{0\}$, maka berdasarkan teorema 2.3.6, terdapat suatu $q(x), r(x) \in F[x]$ sehingga

$$f(x) = q(x)g(x) + r(x)$$

dimana $r(x) = 0$ atau $deg\ r(x) < deg\ g(x)$. Hal ini mengakibatkan, $r(x) = 0$ atau $v(r(x)) < v(g(x))$. Diperoleh bahwa $F[x]$ memenuhi sifat Daerah Euclid yang pertama.

Selanjutnya diberikan $f(x) \in F[x]$, yaitu

$$f(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n, a_n \neq 0.$$

$$g(x) = b_0 + b_1x + b_2x^2 + ... + b_mx^m, b_m \neq 0.$$

Diperoleh,

$$f(x)g(x) = a_0b_0 + (a_0b_1 + a_1b_0)x + ... + (a_nb_m)x^{n+m}.$$

Karena F adalah Lapangan, maka a_n, $b_m \neq 0$.

Hal ini berakibat $deg\ (f(x)g(x)) = n + m$, Sehingga

$$v(f(x)) = deg\ (f(x)) = n \leq n + m = deg\ (f(x)g(x)) = v(f(x)g(x))$$

Dapat dilihat bahwa $F[x]$ memenuhi sifat daerah Euclid yang kedua, sehingga dapat disimpulkan bahwa $F[x]$ merupakan Daerah Euclid. ■

Definisi berikut ini digunakan dalam mengidentifikasi suatu bilangan prima. Bilangan Prima merupakan bilangan yang hanya bisa dibagi dengan 2 bilangan, yaitu 1 dan bilangan itu sendiri. Beberapa sistem kriptografi, menggunakan permasalahan faktorisasi bilangan prima untuk keamanan, sehingga konsep Bilangan prima memiliki peranan penting dalam menciptakan sistem keamanan kriptografi.

Definisi 2.4.4 (Malik,2007) *Diberikan suatu Ring komutatif R dan $a, b \in R$ dimana $a \neq 0$, maka terdapat $c \in R$ sedemikian hingga $b = ac$, maka a dikatakan membagi b atau a disebut pembagi dari b , ditulis $a|b$. Jika a tidak membagi b, maka $a \nmid b$.*

Contoh 2.4.5 Diberikan lapangan $\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$, untuk $b = 3$, maka

$$\begin{aligned} 3 &= 1 \cdot 3, \quad 1|3 \\ &= 2 \cdot 5, \quad 2|5 \\ &= 4 \cdot 6, \quad 4|6 \end{aligned}$$

Definisi 2.4.6 (Malik , 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan. Suatu elemen tak nol* $a \in R$ *dikatakan* ***Associate*** *dari suatu elemen* $b \in R$*, jika* $a = bu$ *untuk suatu* $u \in R$*, dinotasikan* $a \sim b$

Contoh 2.4.7 Ring $\mathbb{Z}$ memiliki elemen unit, yaitu 1 dan -1 sehingga untuk sembarang elemen di $\mathbb{Z}$, $a = bu$, dimana $u = 1$ atau -1,

$$a = (-a) \cdot (-1)$$

sehingga untuk setiap $a \in \mathbb{Z}$, $a \sim (-a)$.

Definisi 2.4.8 (Malik , 2007) *Diberikan suatu Ring komutatif* R *dengan* $a_0, a_1, a_2, ..., a_n$ *elemen di* R *dan tak nol. Suatu elemen tak nol* $d \in R$ ***disebut Common Divisor*** *dari* $a_0, a_1, a_2, ..., a_n$ *jika* $d|a_i$ *untuk semua* $i = 1, 2, 3, n$*. Suatu elemen tak nol* $d \in R$ *disebut* ***Greatest Common Divisor (gcd)*** *dari* $a_0, a_1, a_2, ..., a_n$ *jika*

1. *d adalah common divisor dari* $a_0, a_1, a_2, ..., a_n$*.*
2. *jika* $c \in R$ *adalah common divisor dari* $a_0, a_1, a_2, ..., a_n$ *maka* $c|d$*.*

Konsep ini banyak digunakan dalam sistem keamanan kriptografi, contohnya sistem pertukaran kunci publik RSA. Beberapa kriptografer saat ini terus mengembangkan dan mencari metode tercepat dalam mencari gcd dua buah bilangan. Namun untuk saat ini, metode yang sering digunakan adalah Algoritma Euclid.

Contoh 2.4.9 Diberikan Ring $\mathbb{Z}_8$, yaitu $\mathbb{Z}_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}$ 3 dan 7 saling berasosiasi satu sama lain, karena $7 \cdot 5 = 21 = 5$ merupakan unit di $\mathbb{Z}_8$ ($5 \cdot 5 = 25 = 1$). 7 dan 3 juga merupakan *common divisor* karena keduanya saling membagi satu sama lain ($7 = 3 \cdot 5$) dan ($3 = 7 \cdot 5$), sehingga 7 dan 3 dapat disebut sebagai *greatest common divisor*.

Definisi 2.4.10 (Malik, 2007) *Diberikan sebuah Ring komutatif R dengan elemen satuan, maka*

1. *Suatu elemen p dari R disebut* **iredusibel** *jika p tak nol dan bukan suatu unit, $p = ab$ dimana $a, b \in R$, a atau b merupakan elemen unit. Jika p tidak iredusibel, maka p disebut* **redusibel**.

2. *Suatu elemen p dari R disebut* **Prima** *jika p tak nol dan bukan suatu unit, dan jika $p|ab$, untuk $a, b \in R$, maka p membagi a atau p membagi b.*

3. *Dua elemen a dan b dari R disebut* **Relatif Prima** *jika semua pembaginya adalah unit.*

Contoh 2.4.11 Diberikan sebuah Ring $\mathbb{Z}_4 = \{0, 1, 2, 3\}$. Dapat dibuktikan bahwa unit dari $\mathbb{Z}_4$ adalah 1 dan 3.

$$u = aa^{-1} = 1.$$

untuk $a = 1$ dan $a^{-1} = 1$, maka $1 \cdot 1 = 1 = 1 \cdot 1$

untuk $a = 3$ dan $a^{-1} = 3$, maka $3 \cdot 3 = 9 = 1$

Maka elemen iredusibel dari Ring $\mathbb{Z}_4$ adalah 2, sebab $2 = 1 \cdot 2$, dimana 1 adalah suatu unit dari $\mathbb{Z}_4$. 2 juga merupakan bilangan prima, sebab

$2 = 2 \cdot 1, \;\; 2|2 = p|ab$

$2 = 1 \cdot 2, \;\; 2|2 = p|ab$

Dapat dilihat bahwa 2 membagi 1 dan 2, sehingga 2 merupakan bilangan prima di $\mathbb{Z}_4$.

Pembahasan berikut ini merupakan lanjutan dari pembahasan Daerah Integral. Selain Daerah Euclid, Daerah Integral memiliki sebutan lainnya. Suatu Daerah Integral disebut sebagai Daerah Faktorisasi Tunggal, dimana setiap elemen di dalamnya dapat dinyatakan sebagai perkalian dari suatu unit dan elemen prima/iredusibel dalam suatu Ring R.

Definisi 2.4.12 (Malik , 2007) *Suatu elemen tak nol dan bukan unit dari suatu Daerah Integral D memiliki suatu* ***faktorisasi*** *jika memenuhi,*

$$a = p_0, p_1, p_2, ..., p_n$$

dimana $p_0, p_1, p_2, ..., p_n$ *adalah elemen iredusibel dari D.*

$p_0, p_1, p_2, ..., p_n$ *disebut* ***Faktorisasi dari*** a*. Suatu daerah integral* D *disebut* ***Daerah Faktorisasi*** *jika setiap elemen tak nol dan bukan unit memiliki faktorisasi.*

Selain Daerah faktorisasi tunggal, Daerah Integral R dapat disebut sebagai Daerah Ideal Utama apabila R merupakan Ring Ideal Utama. Suatu Ideal I disebut sebagai Ideal Utama apabila $I =< a >$ untuk suatu $a \in I$.

Definisi 2.4.13 (Malik , 2007) *Diberikan R suatu Ring komutatif dengan elemen satuan. Jika setiap Ideal di R adalah Ideal utama, maka R disebut sebagai Ring Ideal Utama* ***Principal Ideal Ring****. Suatu Daerah Integral yang merupakan Ring Ideal Utama disebut Daerah Ideal Utama* ***Principal Ideal Domain***

Elemen pembangun yang dari suatu Ring polinomial disebut dengan Polinomial Primitif. Polinomial primitif adalah elemen yang dapat membangun semua elemen - elemen pada suatu Ring polinomial.

Definisi 2.4.14 (Malik, 2007) *Diberikan* $f(x) = (a_0 + a_1x + a_2x^2 + ... + a_nx^n)$, *suatu elemen tak nol di Ring* $R[x]$, *maka*

$$gcd\{a_0 + a_1x + a_2x^2 + ... + a_nx^n\}$$

disebut ***content*** *dari* $f(x)$ *dinotasikan* $cont\ f(x)$

Setiap $gcd\{a_0+a_1x+a_2x^2+...+a_nx^n\}$ tidak tunggal. jika terdapat 2 elemen gcd u dan v, maka u dan v saling berasosiasi satu sama lain, dinotasikan $u \sim v$. Artinya, jika c_1, c_2 adalah gcd dari $f(x)$, maka $c_1 \sim c_2$.

Definisi 2.4.15 (Malik , 2007) *suatu elemen tak nol dari Ring Polinomial* $R[x]$ *disebut* ***Polinomial Primitif*** *jika* $cont\ f(x)$ *adalah Unit.*

Teorema 2.4.16 (Malik, 2007) *Diberikan* $f(x) \in F[x]$ *suatu polinomial dengan degree* 2 *atau* 3, *maka* $f(x)$ *dikatakan iredusibel di* F *jika dan hanya jika* $f(x)$ *tidak memiliki akar.*

Bukti. $(\Leftarrow)$ Asumsikan benar jika $f(x)$ iredusibel. Andaikan $f(x)$ memiliki akar, misal a, maka $x - a$ membagi $f(x)$ di $F[x]$, sehingga $f(x)$ redusibel. Akibatnya terjadi kontradiksi dengan pernyataan yang dianggap benar, bahwa $f(x)$ iredusibel. Oleh karena itu, $f(x)$ tidak memiliki akar.

$(\Rightarrow)$ Asumsikan benar jika $f(x)$ tidak memiliki akar. Andaikan $f(x)$ redusibel, berdasarkan Definisi 2.3.8, $f(x)$ memiliki akar, sehingga $f(x) = g(x)h(x)$ untuk suatu $g(x), h(x) \in F[x]$, dengan $deg\ g(x) \geq 1$ dan $deg\ h(x) \geq 1$.

Diasumsikan sebelumnya, bahwa $deg\ f(x) = 3$, maka $deg\ (g(x)h(x)) = 3$.

ada 2 kemungkinan yang terjadi

1. $deg\ g(x) = 2 \quad dan \quad deg\ h(x) = 1$
2. $deg\ g(x) = 1 \quad dan \quad deg\ h(x) = 2$

Andaikan kemungkinan 1 yang terjadi, maka $g(x) = ax + b$ untuk suatu $a, b \in F, a \neq 0$. Diambil suatu elemen di F, yaitu $a = -a^{-1}b$, maka

$$g(-a)^{-1}b) = a(-a^{-1}b) + b = -aa^{-1}b + b = -(1)b + b = -b + b = 0$$

Dapat dilihat, bahwa $g(x)$ memiliki akar, yaitu $-a^{-1}b$ dan $-a^{-1}b$ merupakan akar dari $f(x)$.

Maka terjadi suatu kontradiksi, karena diasumsikan bahwa $f(x)$ tidak memiliki akar, sehingga pengandaian bahwa $f(x)$ redusibel dapat disanggah, dan pernyataan bahwa $f(x)$ iredusibel adalah benar. ■

Berikut ini adalah contoh polinomial yang iredusibel.

Contoh 2.4.17 Diberikan polinomial $f(x) \in \mathbb{Z}_2[x]$,

$$f(x) = x^2 + x + 1.$$

1. untuk $a = 0$, $0^2 + 0 + 1 = 1 \neq 0$, maka 0 bukan akar dari $f(x)$
2. untuk $a = 1$, $1^2 + 1 + 1 = 3 = 1 \neq [0]$, maka 1 bukan akar dari $f(x)$

Oleh karena itu, $f(x)$ merupakan polinomial iredusibel.

Lemma 2.4.18 (Malik , 2007) *Diberikan R suatu Daerah Faktorisasi Tunggal. Jika $f(x), g(x)$ adalah polinomial primitif di $R[x]$ maka $f(x)g(x)$ juga merupakan polinomial primitif di $R[x]$.*

Bukti. Diambil suatu $f(x), g(x) \in R[x]$,

$$f(x) = a_0 + a_1x + a_2x^2 + ... + a_nx^n.$$

$$g(x) = b_0 + b_1x + b_2x^2 + ... + b_mx^m.$$

Selanjutnya didefinisikan c_f yaitu $cont\ f(x)$ dan c_g yaitu $cont\ g(x)$. Berdasarkan 2.4.15, c_f dan c_g merupakan elemen unit. Selanjutnya untuk $f(x)g(x)$,

$$f(x)g(x) = c_0 + c_1x + c_2x^2 + ... + c_{n+m}x^{n+m}$$

dimana $c_0 = a_0b_0,\ \ c_1 = (a_0b_1 + a_1b_0),\ \ \sum_{j=0}^{i} a_jb_{i-j}$ dengan $a_j = 0$, jika $j > n$ dan $b_{ij} = 0$, jika $i - j > m$. Diasumsikan $f(x)g(x)$ bukan polinomial primitif, sehingga $cont\ f(x)g(x)$ bukan suatu unit.

Selanjutnya diambil suatu elemen prima $P \in R$, dimana $P|cont\ f(x)g(x)$, maka

$$p|c_i, \quad untuk\ i = 0, 1, 2, ..., n + m.$$

Diketahui c_f dan c_g merupakan elemen unit, maka

$$p \nmid \{a_0 + a_1x + a_2x^2 + ... + a_n\}.$$

$$p \nmid \{b_0 + b_1x + b_2x^2 + ... + b_m\}.$$

Dimisalkan t dan r adalah suatu bilangan positif terkecil dimana p tidak membagi a_t dan b_r, sehingga

$$P|a_i, \quad untuk\ i = 0, 1, .., t - 1\ \ dan\ \ p \nmid a_t.$$

$$P|b_j, \quad untuk\ j = 0, 1, .., t - 1\ \ dan\ \ p \nmid b_r.$$

Dari hal tersebut, maka $p \nmid a_tb_r$. Selanjutnya untuk c_{t+r}, yaitu

$$c_{t+r} = a_0b_{t+r} + a_1b_{t+r-1} + ... + a_{t-1}b_{r+1} + a_tb_r + a_{t+1}b_{r-1} + ... + a_{t+r}b_0$$

dimana $a_i = 0$, jika $i > n$ dan $b_i = 0$, jika $i > m$, maka

$$P|a_i, \quad untuk\ i = 0, 1, .., t - 1.$$

$$P|b_j, \quad untuk\ j = 0, 1, .., t - 1.$$

Berdasarkan asumsi, $p|cont\ f(x)g(x)$, maka

$$p|c_{t+r}\ \ akibatnya\ \ p|a_t b_r.$$

Kontradiksi, karena sebelumnya diketahui bahwa $p \nmid a_t b_r$. Oleh karena itu, terbukti bahwa $cont\ f(x)g(x)$ adalah Unit, sehingga $f(x)g(x)$ merupakan polinomial primitif. ■

Berikut ini adalah contoh menentukan polinomial primitif.

Contoh 2.4.19 Diberikan $f(x) \in \mathbb{Z}[x]$,

$$f(x) = 6x^2 + 3x - 9 = 3(2x^2 + x - 3).$$

3 adalah $gcd(6, 3, 9)$ dan bukan elemen unit di $\mathbb{Z}$ sehingga $f(x)$ bukan polinomial primitif, namun untuk polinomial,

$$g(x) = 2x^2 + x - 3.$$

1 adalah $gcd(2, 1, -3)$ dan 1 adalah unit di $\mathbb{Z}$, sehingga $g(x)$ adalah polinomial primitif.

Teorema 2.4.20 (Malik, 2007) *Diberikan Ring polinomial $F[x]$ atas lapangan F dan $p(x) \in F[x]$, maka pernyataan - pernyataan berikut ekuivalen :*

1. *$p(x)$ Iredusibel.*
2. *$F[x]/\langle p(x)\rangle$ adalah Daerah Integral.*
3. *$F[x]/\langle p(x)\rangle$ adalah Lapangan.*

Bukti. $(iii) \Rightarrow (ii)$ Asumsikan benar jika $F[x]/p(x)$ merupakan lapangan. Andaikan $F[x]/p(x)$ bukan Daerah Integral, maka $F[x]/p(x)$ memiliki elemen pembagi nol sehingga terjadi kontradiksi, karena setiap lapangan tidak memuat elemen pembagi nol.

$(ii) \Rightarrow (i)$ Langkah pertama yaitu mengidentifikasi $p(x)$, apakah $p(x)$ adalah unit atau bukan.

1. Asumsikan $F[x]/p(x)$ adalah Daerah Integral. Jika $p(x)$ adalah unit, maka $\langle p(x)\rangle = F[x]$ sehingga $F[x]/p(x) = \{0\}$ yang menandakan bahwa $F[x]/p(x)$ memiliki elemen pembagi nol, sehingga kontradiksi karena diketahui $F[x]/p(x)$ Daerah Integral.

2. Jika $p(x)$ bukan unit, maka $p(x) = f(x)g(x)$ untuk suatu $f(x), g(x) \in F[x]$. jika $\overline{p(x)} = \overline{0} = \overline{f(x)g(x)} = \overline{f(x)}\ \overline{g(x)}$, maka $\overline{f(x)} = 0$ atau $\overline{g(x)} = 0$, sehingga $\overline{f(x)} \in \langle p(x)\rangle$ atau $\overline{g(x)} \in \langle p(x)\rangle$.

Asumsikan $\overline{f(x)} \in \langle p(x)\rangle$, maka $\overline{f(x)} = q(x)p(x)$, untuk suatu $q(x) \in F[x]$, sehingga

$$p(x) = f(x)g(x) = q(x)p(x)g(x) = (q(x)g(x))p(x).$$

Dari hal tersebut, diperoleh $q(x)g(x) = 1$, sehingga $q(x), g(x) \in F\backslash\{0\}$ dan merupakan elemen unit. Selanjutnya,

$$\begin{aligned} p(x) &= (q(x)g(x))p(x) \\ &= u^{-1}(up(x)). \end{aligned}$$

sehingga $P(x)$ iredusibel.

$(i) \Rightarrow (iii)$ Diambil sembarang elemen dari $F[x]$, yaitu $(\overline{f(x)} = f(x) + \langle p(x) \rangle$, dimana $\overline{f(x)} \neq 0$. Diketahui $P(x)$ iredusibel, maka terdapat u dan $up(x)$, dimana $u \in F[x] \backslash \{0\}$. u elemen pembagi dari $p(x)$. Diberikan $u = 1 \in F[x] \backslash \{0\}$, dimana $1|f(x)$ dan $1|p(x)$ dengan $f(x) \neq p(x)$. Berdasarkan 2.4.10, $f(x)$ dan $p(x)$ relatif prima, sehingga terdapat $r(x), s(x) \in F[x]$, sedemikian hingga

$$1 = r(x)f(x) + s(x)p(x)$$

$$\overline{1} = \overline{r(x)f(x))} + \overline{s(x)p(x)}.$$

Jika $\overline{r(x)f(x))} = 1$, artinya setiap $\overline{f(x)}$ memiliki invers, yaitu $\overline{r(x)}$. Akibatnya, $F[x]/\langle p(x) \rangle$ adalah lapangan. ■

2.5. Ideal Prima dan Maksimal

Ideal prima merupakan suatu subset dari sebuah Ring, yang dibangun oleh sebuah elemen prima. Studi tentang Ideal prima memiliki peran pada teori bilangan, yang mana setiap faktorisasi tunggal pada teori fundamental aritmatika tidak berlaku di semua Ring namun hal tersebut berlaku jika elemen - elemennya dipandang sebagai Ideal, dan elemen prima dipandang sebagai Ideal prima. Berikut adalah definisi dari Ideal Prima.

Definisi 2.5.1 (Malik, 2007) *Suatu Ideal P dari Ring R dikatakan prima jika,*

$$(\forall A, B \trianglelefteq R)\ A \subseteq P \ \ atau \ \ B \subseteq P.$$

Contoh 2.5.2 $P = \{3k | k \subseteq \mathbb{Z}\}$ adalah ideal prima di Ring $\mathbb{Z}$, yaitu

1. $P \ \ ideal \ \ prima \ \ \Leftrightarrow \ \ (\forall (a,b) \in P) \ \ Jika \ \ 3|ab, \ maka \ \ 3|a \ \ atau \ \ 3|b$
2. $P \ \ ideal \ \ prima \ \ \Leftrightarrow \ \ (\forall (a,b) \in P) \ \ a = 3a \ \ atau \ \ b = 3b$

$J = \{6k | k \in \mathbb{Z}\}$ bukan Ideal Prima karena $3.2 = 6 \in J$, namun $3 \notin J$ dan $2 \notin J$

Teorema 2.5.3 (Malik, 2007) *Suatu Ideal P dari Ring R dikatakan Prima jika dan hanya jika untuk setiap $a, b \in R$, jika $aRb \subseteq P$, maka $a \in P$ atau $b \in P$.*

Bukti. $(\Rightarrow)$ Diasumsikan P Ideal Prima dan $aRb \in P$ dimana $a, b \in R$. Ambil sembarang A, B Ideal di R, dimana

$$A = RaR \ \ dan \ \ B = RbR.$$

Maka,

$$AB = (RaR)(RbR) \subseteq R(aRb)R \subseteq RPR \subseteq P.$$

Diketahui P Ideal Prima, maka $A \subseteq P$ atau $B \subseteq P$. Selanjutnya, untuk $\langle a^3 \rangle \subseteq RaR = A \subseteq P$, karena P Ideal Prima, maka $\langle a \rangle \subseteq P$ sehingga $a \in P$.

Sama halnya dengan $\langle b^3 \rangle \subseteq RbR = B \subseteq P$, karena P Ideal Prima, maka $\langle b \rangle \subseteq P$ sehingga $b \in P$.

($\Leftarrow$) Diasumsikan untuk setiap $a, b \in R$ jika $aR \subseteq P$ maka $a \in P$ atau $b \in P$. Diambil sembarang A, B Ideal di R sedemikian hingga $AB \subseteq P$. Andaikan $A \nsubseteq P$, maka ada suatu elemen $a \in A$ sedemikian hingga $a \in P$. Diberikan $b \in B$, $aRb = (aR)b \subseteq AB \subseteq P$, maka $a \in P$ atau $b \in P$. Diketahui sebelumnya $a \notin P$, maka $b \in P$ sehingga $B \subseteq P$. Hal tersebut berlaku sebaliknya. Andaikan $B \subseteq P$, maka $b \nsubseteq P$ dan $a \in P$. ■

Akibat 2.5.4 (Malik, 2007) *Diberikan suatu Ring komutatif R. Suatu Ideal P di R dikatakan Ideal Prima jika dan hanya jika untuk setiap $a, b \in R$,*

$$ab \in P \quad \Rightarrow \quad a \in P \ \ atau \ \ b \in P.$$

Semua Ideal prima tak nol di Daerah Ideal Utama adalah Ideal Maksimal. Secara sederhana, suatu Ideal I dari Ring R dikatakan Maksimal yaitu apabila tidak ada Ideal lain antara I dan R. Berikut ini adalah definisi Ideal Maksimal.

Definisi 2.5.5 (Malik, 2007) *Diberikan Ring R dan M suatu Ideal (Kiri, Kanan) di R, maka M dikatakan **Ideal Maksimal** dari R jika $M \neq R$, dan untuk J suatu ideal(Kiri, Kanan) di R, maka*

$$(M \subseteq J \subseteq R) \ \Rightarrow \ M = J \ \ atau \ \ J = R.$$

Teorema 2.5.6 (Malik, 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan dan P suatu Ideal di R, dimana $R \neq P$, maka*

$$P \ \ Ideal \ \ Prima \quad \Leftrightarrow \quad R/P \ \ Daerah \, Integral.$$

Bukti. ($\Rightarrow$) Diberikan P suatu ideal prima di R dan R suatu Ring komutatif, maka R/P juga Ring komutatif. Kemudian,

$$P \neq R \ \ dan \ \ 1 + P \in R/P \neq 0 + P \in R/P.$$

akan ditunjukan bahwa R/P tidak memuat elemen pembagi nol. Diambil sembarang $a + P, b + P \in R/P$, maka

$$(a + P)(b + P) = (0 + P) \ \ \rightarrow \ \ ab + P = 0.$$

Hal ini menunjukan $ab \in P$ dan karena P adalah Ideal Prima, maka $a \in P$ atau $b \in P$ sehingga,

$$a + P = 0 + P \ \ atau \ \ b + P = 0 + P.$$

Terbukti bahwa R/P tidak memuat elemn pembagi nol, yang mengakibatkan R/P Daerah Integral.

($\Leftarrow$) Diasumsikan R/P Daerah Integral, maka untuk $ab = 0 \in P$,

$$ab + P = (a + P)(b + P) = 0 + P.$$

R/P adalah Daerah Integral, maka R/P tidak memuat elemen P, artinya $(a + P) = 0 + P$ atau $(b + P) = 0 + P$. Hal ini menunjukan $a \in P$ atau $b \in P$ sehingga berdasarkan Akibat 2.5.4, P Ideal Prima. ■

Teorema 2.5.7 (Malik, 2007) *Diberikan suatu Ring komutatif R dengan elemen satuan dan M suatu Ideal di R, maka*

$$P \ \ Ideal \ \ Maksimal \ \ \Leftrightarrow \ \ R/M \ \ Lapangan.$$

Bukti. ($\Rightarrow$) Diberikan M suatu Ideal Maksimal di R. Jika R suatu Ring komutatif, maka R/M juga Ring komutatif. Kemudian untuk semua $a \in R$, $\overline{a} \in R/M$ dan $\overline{a} \neq \overline{0} \notin M$ dinotasikan $\overline{a} = a + M$. Oleh karena itu, Ideal $\langle M, a\rangle$ dibentuk oleh $M \cup \{a\}$. Diketahui bahwa M Ideal Maksimal, maka

$$\langle M, a\rangle = R.$$

Sehingga terdapat $m \in R$ dan $r \in R$ sedemikian hingga $m + ra = 1$. Dari hal tersebut,

$$\overline{m} + \overline{ra} = \overline{1}.$$

$\overline{m} = 0$, maka $\overline{ra} = 1$. Hal ini menunjukan bahwa sembarang $\overline{a} \in R/M$ memiliki invers yaitu $\overline{r}$, sehingga R/M merupakan Lapangan.

($\Leftarrow$) Diasumsikan R/M adalah lapangan, maka $R \notin M$. Diambil suatu Ideal I dari R sedemikian hingga $I \subset M \subseteq R$, maka terdapat $\overline{a} = a + M \in R/M$ dan $\overline{r} = r + M \in R/M$ sedemikian hingga,

$$\overline{ar} = \overline{ar} = (a + M)(r + M) = \overline{1} = 1 + M.$$

Hal ini mengakibatkan $1 - ar \in M$. Oleh Karena itu, $1 = m + ar$ untuk suatu $m \in M$, sehingga

$$1 = m + ar \in M + I \subseteq I.$$

Diperoleh $I = R$, sehingga M merupakan Ideal Maksimal. ■

2.6. Lapangan Hingga (*Finite Field*)

Pada subbab ini, akan dibahas teori - teori mengenai *Finite Field* (Lapangan Hingga). Bahasan ini meliputi definisi *Field, Subfield, Vector Space, Field Extension, Splitting Field* dan *Finite Field*.

Definisi 2.6.1 (Malik , 2007) *Diberikan suatu Ring F, maka F disebut* **Field** *(Lapangan) apabila*

1. *Ring F adalah suatu Ring komutatif.*
2. *Ring F memuat elemen satuan e, dengan $e \neq 0$.*
3. *Setiap elemen tak nol di F mempunyai invers terhadap perkalian.*

Seperti yang telah dijelaskan sebelumnya, Suatu sistem kriptografi harus memiliki struktur aljabar berupa Lapangan atau Lapangan Hingga, agar proses enkripsi dan dekripsi dapat bekerja secara optimal. Lapangan Whirlpool, yaitu $GF(2^8)$ merupakan Lapangan Hingga dengan order 2^8 dan karakteristiknya adalah bilangan prima 2, berdimensi 8. Dimensi merupakan kajian yang dibahas dalam konsep Ruang Vektor, pada Aljabar Linear.

Definisi 2.6.2 (Anton,2000) *Diberikan suatu himpunan tak kosong V yang memuat objek - objek dan 2 buah operasi biner, yaitu penjumlahan didefinisikan $(u + v)$ dan perkalian (dengan skalar) didefinisikan (ku) dimana k merupakan elemen dari suatu Lapangan $(F, +, \cdot)$, maka u, v, dan w yaitu objek - objek dalam V, disebut sebagai **vektor** dari V dan V disebut sebagai **Ruang Vektor**, apabila memenuhi aksioma aksioma berikut.*

1. $(\forall u, v \in V)\ \ u + v \in V.$
2. $(\forall u, v \in V)\ \ u + v = v + u.$
3. $(\forall u, v, w \in V)\ \ u + (v + w) = (u + v) + w.$
4. $(\exists \overline{0} \in V)(\forall u \in V)\ \ u + \overline{0} = \overline{0} + u = u.$
5. $(\exists - u \in V)(\forall u \in V)\ \ u + (-u) = (-u) + u = \overline{0}.$
6. $(\forall k \in F)(\forall u \in V)\ \ ku \in V.$
7. $(\forall k \in F)(\forall u, v \in V)\ \ k(u + v) = ku + kv.$
8. $(\forall k, m \in F)(\forall u \in V)\ \ (k + m)(u) = ku + mu.$
9. $(\forall k, m \in F)(\forall u \in V)\ \ km(u) = (km)u.$
10. $(\exists 1 \in V)(\forall u \in V)\ \ 1u = u1 = u.$

Contoh 2.6.3 $V_2(\mathbb{Z})$ merupakan Ruang vektor dengan entri - entri nya merupakan elemen dari $\mathbb{Z}$ yaitu,

$$V_2(\mathbb{Z}) = \left\{ \begin{pmatrix} a \\ b \end{pmatrix} | a, b \in \mathbb{Z} \right\}$$

Basis merupakan suatu himpunan vektor - vektor dari suatu Ruang Vektor, dimana setiap elemennya dapat dibentuk secara tunggal dari kombinasi linear antara vektor - vektor pada himpunan tersebut. Suatu Ruang Vektor dapat memuat beberapa basis, dan semua basis akan memuat jumlah elemen yang sama disebut dengan Dimensi.

Definisi 2.6.4 (Anton , 2000) *Diberikan sebuah Ruang vektor V dan $S = \{v_1, v_2, ..., v_n\}$ suatu himpunan vektor - vektor di V, maka S disebut* ***Basis*** *apabila memenuhi 2 kondisi,*

1. *S bebas linear yaitu,*

$$k_1v_1 + k_2v_2 + ... + k_nv_n \quad \Rightarrow \quad k_1 = k_2 = ... = k_n = 0 \in F.$$

2. *S membangun V yaitu,*

$$V = Span(S).$$

Definisi 2.6.5 (Anton , 2000) *Sebuah Ruang vektor V disebut **Ruang Vektor Berdimensi-Hingga** jika V memiliki vektor - vektor yang membentuk sebuah Basis. Jika tidak, maka V disebut **Ruang Vektor Berdimensi-Tak Hingga**.*

Definisi 2.6.6 (Anton,2000) *Dimensi dari sebuah Ruang Vektor Berdimensi-Hingga dinotasikan $dim(V)$ adalah vektor - vektor dalam suatu Basis di V.*

Teorema 2.6.7 (Anton , 2000) *Jika W merupakan Subruang dari Ruang Vektor Berdimensi-Hingga V, maka*

$$dim(W) \leq dim(V).$$

Jika $dim(W) = dim(V)$, maka $W = V$.

Bukti. Diasumsikan himpunan vektor - vektor $S = \{w_1, w_2, ..., w_n\}$ adalah basis dari W.

1. Jika Basis dari W juga merupakan Basis dari V, maka

$$dim(W) = dim(V) = n.$$

2. Jika bukan, karena W merupakan subruang dari V, maka dapat ditambahkan beberapa vektor kedalam himpunan bebas linear S, sehingga S menjadi basis dari V. Hal ini mengindikasikan bahwa $dim(W) < dim(V)$.

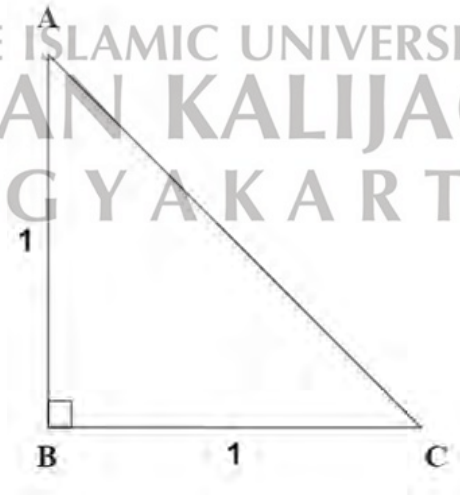
3. Dari 1 dan 2, diperoleh

$$dim(W) \leq dim(V).$$

Selanjutnya diasumsikan benar jika $dim(W) = dim(V)$. Andaikan $W \neq V$, maka berdasarkan point 2, $dim(W) < dim(V)$. Terjadi kontradiksi dengan asumsi yang dianggap benar, yaitu $dim(W) = dim(V)$. Oleh karena itu, terbukti benar jika $dim(W) = dim(V)$ maka $W = V$. ■

Definisi 2.6.8 (Baker,2017) *Diberikan lapangan K dan L, $K \subseteq L$, maka K disebut dengan Subfield dari L dan L dapat dikatakan sebagai perluasan dari K, ditulis $K \leq L$ atau L/K.*

Field Extension atau lapangan perluasan dilatarbelakangi dengan ditemukannya bilangan - bilangan irrasional dalam $\mathbb{R}$. Salah satu bilangan irrasional dalam $\mathbb{R}$ adalah akar kuadrat dari 2 (atau $\sqrt{2}$). Permasalahan akar kuadrat dari 2 ditemukan pada penerapa teorema *Phytagoras*



Berdasarkan teorema *Phytagoras*

$$AC^2 = BC^2 + BA^2$$
$$= 1^2 + 1^2$$
$$= 2$$
$$AC = \sqrt{2}.$$

Contoh lain dari bilangan irrasional di $\mathbb{R}$ adalah $\sqrt{-1}$ dan π. Simbol π merupakan simbol untuk suatu bilangan representatif tak hingga yaitu $(3, 1415626536)$.

Definisi 2.6.9 (Baker, 2017) *$dim(L)$ adalah degree atau $index$ dari perluasan L/K dinotasikan $[L : K]$. Perluasan dari lapangan L/K disebut Berhingga (dimensinya) jika $[L : K] < \infty$. Jika tidak, maka L/K disebut Tak Berhingga.*

Contoh 2.6.10 $\mathbb{C}$ merupakan salah satu contoh lapangan perluasan dari K, dinotasikan $\mathbb{C}/\mathbb{R}$ dengan Basis dari $\mathbb{C}$ yaitu,

$$C = \{x + yi | x, y \in \mathbb{R}\}.$$

Definisi 2.6.11 (Baker, 2017) *Suatu elemen $t \in L$ disebut **Algebraic** atas K jika terdapat suatu polinomial tak nol $p(x) \in K[X]$ sehingga $p(t) = 0$.*

Definisi 2.6.12 (Baker, 2017) *Diberikan L/K suatu perluasan berhingga, maka suatu elemen $u \in L$ dengan $L = K(u)$, disebut elemen primitif dari perluasan L/K. Jika L/K memiliki suatu elemen primitif, maka L/K disebut **Perluasan Sederhana (simple extension)**.*

Lemma 2.6.13 (Baker, 2017) *Diberikan L/K suatu perluasan berhingga dan $u \in L$, maka u adalah suatu elemen primitif dari L/K jika dan hanya jika $deg\ K(u) = [L : K]$*

Bukti. ($\Rightarrow$) Jika u adalah suatu elemen primitif dari L/K, maka berdasarkan 2.6.14, $L = K(u)$. Kemudian berdasarkan 2.6.7, jika $K(u) \subseteq L$ dan $L = K(u)$, maka $dim\ K(u) = dim\ (L)$. Selanjutnya, berdasarkan 2.6.9,

$dim\ K(u) = deg\ K(u)$ dan $dim\ (L) = deg\ (L) = [L : K]$, sehingga

$$deg\ K(u) = deg\ (L) = [L : K].$$

($\Rightarrow$) jika $deg\ K(u) = [L : K]$, maka $deg\ K(u) = deg(L) = dim(K(u) = dim(L)$. Kemudian berdasarkan 2.6.7,

$$dim\ K(u) = dim\ (L) \quad \Rightarrow \quad K(u) = L.$$

Sehingga berdasarkan 2.6.14,

$$K(u) = L \quad \Rightarrow \quad u\ merupakan\ elemen\ primitif\ \ dari\ \ L/K.$$

■

Definisi 2.6.14 (Baker , 2017) *Perluasan minimal dari K disebut* ***Lapangan Terbelah*** *(**Splitting Field**) dari $p(X)$ atas K dinotasikan $K(p(X))$ atau K_p*

Contoh 2.6.15 Berikut ini adalah contoh mencari *splitting field* dari E/Q untuk,

$$P(X) = x^4 - 4.$$

Penyelesaian:

1. Dicari akar dari $P(X)$,

$$P(X) = x^4 - 4 = (x^2 - 2)(x^2 + 2).$$

Diperoleh akar dari $P(X)$ adalah $\sqrt{2}$ dan $-\sqrt{2}$.

2. Akar dari $(x^2 - 2)$ adalah $\sqrt{2}$ dan $-\sqrt{2}$. Agar dapat dibentuk $Q(\sqrt{2})$, maka kedua akar digabungkan terlebih dahulu, sehingga $Q(\sqrt{2}, -\sqrt{2}) = Q(\sqrt{2})$. Terbentuk sebuah perluasan yaitu $Q(\sqrt{2})/Q$ dengan degree 2.

3. Akar dari $(x^2 + 2) \in Q(\sqrt{2})[x]$ adalah $\sqrt{2}i$, yang merupakan bilangan kompleks, dan bukan elemen dari $\mathbb{R}$, sehingga $(x^2 + 2)$ tak tereduksi di $Q(\sqrt{2})[x]$.

4. $(x^2 + 2)$ tidak memiliki akar di $Q(\sqrt{2})[x]$, maka perlu diperluas lagi, menjadi $Q(\sqrt{2}, \sqrt{2}i) = Q(\sqrt{2}, i)$ sehingga diperoleh perluasan baru, yaitu $Q(\sqrt{2}, \sqrt{2}i)/Q(\sqrt{2})$ dengan degree 2.

5. Diperoleh *Splitting Field* dari $p(x)$ atas $\mathbb{Q}$ di $\mathbb{C}$ adalah $Q(\sqrt{2}, \sqrt{2}i)$ dengan dimensinya $[E : Q] = 4$.

Lemma 2.6.16 (Baker, 2017) *Diberikan sebuah Lapangan hingga (Finite Field) F dengan sejumlah elemen q dan diberikan V sebuah Ruang vektor-F. maka,*

$$dim(V) < \infty \quad \Leftrightarrow \quad V \ Berhingga \ dan \ |V| = q^{dim(V)}.$$

Bukti. $(\Rightarrow)$ Jika $d = dim(V) < \infty$, maka V memiliki suatu basis S dan setiap elemen dari V, yaitu $v_1, v_2, ..., v_n \in V$ dapat dibentuk secara tunggal dengan elemen-elemen pada Basis S, yaitu $v_n = k_1v_1 + k_2v_2, ..., k_nv_n$ dimana $k_1, k_2, ..., k_n \in F$. Dari hal tersebut, diperoleh $|V| = q^d = q^{dim(v)}$.

$(\Leftarrow)$ Jika vektor - vektor dalam V berhingga dan $|V| = q^{dim(V)}$ maka V memiliki suatu basis, sehingga berdasarkan 2.6.5, Ruang vektor V Berdimensi Hingga, yaitu

$$dim(V) < \infty.$$

■

2.7. *Galois Field*

Galois field lebih dikenal sebagai Lapangan Hingga atau *Finite Field*. *Galois field* merupakan studi lanjutan dari Lapangan dan Ring. Kata *Galois* diambil dari penemunya, yaitu Evariste Galois. *Galois Field* merupakan Lapangan yang memuat elemen berhingga.

Definisi 2.7.1 (Baker, 2017) *Suatu Subfield F_{p^d} dari lapangan F_p disebut Galois Field order P^d.*

Subfield F_{p^d} dari lapangan F_p dinotasikan dengan $GF(p^d)$, dimana p karakteristik dari lapangan $GF(p^d)$, dan $d \in Z^+$ dengan dimensi $[F_{p^d} : F_p] = d$

Elemen-elemen pada $GF(p^d)$ adalah sebagai berikut [Benvenuto,2012].

$$\begin{aligned} GF\left(p^d\right) &= (0, 1, 2, \ldots, p-1) \\ &= (p, p+1, p+2, \ldots, p+p-1) \cup \left(p^2, p^2+1,\ p^2+2, \ldots, p^2+p-1\right) \cup \\ &\left(p^{d-1}, p^{d-1}+1, p^{d-1}+2, \ldots, p^{d-1}+p-1\right). \end{aligned}$$

Sebagai contoh, $GF(2^3)$ memuat elemen $2^3 = 8$ elemen, yaitu

$$\begin{aligned} GF\left(2^3\right) &= \left(0, 2^0, 2^1, 2^1+2^0, 2^2, 2^2+2^0, 2^2+2^1, 2^2+2^1+2^0\right) \\ &= (0, 1, 2, 3, 4, 5, 6, 7). \end{aligned}$$

Setiap elemen dalam $GF(2^3)$ merupakan polinomial dengan $degree$ terbesarnya yaitu $(d-1) = (3-1) = 2$. $GF(2^3)$ dapat ditulis,

$$GF(x^3) = (0, x^0, x^1, x^1+x^0, x^2, x^2+x^0, x^2+x^1, x^2+x^1+x^0), \quad dengan\ x = 2.$$

Bentuk Polinomial tersebut jika direpresentasi dalam bentuk biner adalah,

$$GF(x^3) = (000, 001, 010, 011, 100, 101, 110, 111),$$

dimana 1 mewakili koefisien polinomial - polinomial di $GF(2^3)$.

Tabel 2.1 Representasi Biner $GF(2^3)$

Bentuk Biner	**Elemen** $GF(2^3)$
$\{000\}$	$0x^2 + 0x^1 + 0 = 0$
$\{001\}$	$0x^2 + 0x^1 + 1 = 1$
$\{010\}$	$0x^2 + 1x^1 + 0 = 2$
$\{011\}$	$0x^2 + 1x^1 + 1 = 3$
$\{100\}$	$1x^2 + 0x^1 + 0 = 4$
$\{101\}$	$1x^2 + 0x^1 + 1 = 5$
$\{110\}$	$1x^2 + 1x^1 + 0 = 6$
$\{111\}$	$1x^2 + 1x^1 + 1 = 7$

Dalam Algoritma Whirlpool, Lapangan Galois $GF(p^d)$ dipandang sebagai Lapangan Hingga berorder 2^8, dengan $P = 2$, dan $d = 8$. Bilangan prima 2 dipilih dikarenakan sistem komputer hanya dapat memahami 2 bilangan saja, yaitu 0 dan 1. Lapangan Galois berdimensi 8 dipilih karena dalam 1-byte terdapat 8-bits yang terdiri dari angka 0 dan 1.

2.7.1. Sistem Biner

1) BIT

Dalam sistem komputer, semua karakter baik angka ataupun huruf merupakan representasi dari deret biner yang terdiri dari angka 0 dan 1 yang disebut Bit. Bit atau *Binary Digits* merupakan satuan unit terkecil dalam komputer, yang merepresentasikan tegangan on dan off.

Terdapat banyak jenis sistem pengkodean karakter di dunia ini, seperti ASCII, UTF, ISO, dan lain sebagainya. Semua sistem pengkodean menyusun bit - bit untuk mendefinisikan suatu karakter.

Sebagai contoh, pada sistem pengkodean ASCII, huruf A merupakan representasi dari deret-deret bit berikut :

$$A = (01000001)\,.$$

2) BYTE

Deret 8-bit disebut sebagai Byte. Awal mula diciptakannya Byte adalah untuk mempermudah penulisan text pada komputer. 1 Karakter yang ditampilkan pada layar komputer, merupakan representasi dari 8 deret bit atau 1 Byte. Dengan menggunakan ini, 1-Byte kode dapat merepresentasikan 256 karakter yang berbeda.

3) DECIMAL

Desimal adalah sistem numerik pada komputer yang terdiri dari $0 - 9$. Perhitungan desimal pada komputer, dapat diperoleh dengan mengubah deret bit ke dalam bentuk polinomial. Berikut ini adalah contoh perhitungan desimal dari huruf A

pada sistem pengkodean ASCII:

$$A = (01000001)$$

$$A = \left(x^6 + 1\right).$$

Order dari $GF = 2$, maka $x = 2$ sehingga,

$$A = 2^6 + 1 = 64 + 1 = 65.$$

Bentuk representasi A dalam desimal pada sistem pengkodean ASCII adalah 65.

4) HEXADECIMAL

Bentuk representasi 8-bit atau 1-byte dapat dibagi menjadi 2 grup dengan 4-bit. Grup - grup dengan 4-tupel bit disebut dengan *Hexadecimal* (Hex). Berikut ini adalah contoh representasi huruf A pada sistem pengkodean ASCII ke dalam bentuk Hex.

$$A = (0100\ 0001).$$

Deret bit 0100 merupakan representasi biner dari Hex 4 dan deret bit 0001 merupakan representasi biner dari Hex 1, sehingga karakter A dalam sistem pengkodean ASCII, jika direpresentasikan dalam bentuk Hex, adalah

$$A = \{41\} = (0100\ 0001.)$$

2.7.2. Operasi Matematika pada Galois Field

Setiap byte dalam Whirlpool direpresentasikan ke dalam bentuk polinomial yang merupakan elemen dari Lapangan Galois $GF(2^8)$. Setiap elemennya dijumlahkan dan digandakan berdasarkan sifat penjumlahan dan perkalian Ring Polinomial.

1) Penjumlahan

Setiap byte dalam Whirlpool dijumlahkan berdasarkan aturan XOR atau Aljabar Boolean. Hal ini dikarenakan setiap elemen dalam Lapangan Galois berada pada $\mathbb{Z}_2$ yang hanya memiliki 2 kemungkinan, 0 OR 1. Akibatnya, XOR juga dapat dipandang sebagai operasi penjumlahan modulo 2. XOR adalah singkatan dari *Executive OR* atau *Executive Disjunction*, yaitu sebuah logika operasi pada dua buah nilai, dimana ketika 2 buah input bernilai beda, maka outputnya bernilai benar. Dalam kriptografi, operasi ini ditandai dengan $\oplus$.

Berikut adalah contoh penjumlahan dua buah elemen dalam Lapangan Galois,

Representasi Polinomial :

$$A = x^7 + x^5 + x^2 + 1$$

$$B = x^5 + x^3 + x^1$$

$$A + B = x^7 + 2x^5 + x^3 + x^2 + x^1 + 1 = x^7 + x^3 + x^2 + x^1 + 1$$

Representasi Biner :

$$A = (1010\ 0101)$$

$$B = (0010\ 1010)$$

$$A \oplus B = (1000\ 1111)$$

2) Perkalian

Dalam WHIRLPOOL, Perkalian polinomial dalam $GF(p^d)$ dinotasikan dengan (•) , yaitu

$$h(x) = f(x) \bullet g(x)\ mod\ m(x).$$

dimana $m(x)$ merupakan suatu polinomial iredusibel, yaitu

$$m\,(x) = x^8 + x^4 + x^3 + x^2 + 1.$$

Perkalian atas modulo $m(x)$ bertujuan agar hasil perkalian 2 buah elemen, menghasilkan polinomial dengan $degree$ yang lebih kecil dari 8, sehingga dapat direpresentasikan dalam bentuk Byte.

Berikut adalah contoh perkalian polinomial dalam $GF(2^8)$ dengan modulo polinomial iredusibel $m\,(x) = x^8 + x^4 + x^3 + x^2 + 1$:

$$\begin{aligned}(3D \bullet 02) &= (0011\ 1101) \bullet (0000\ 0010)\ mod\,(100011101)\\ &= (x^5 + x^4 + x^3 + x^2 + 1) \bullet (x)\ mod\left(x^8 + x^4 + x^3 + x^2 + 1\right)\\ &= (x^6 + x^5 + x^4 + x^3 + x) mod\left(x^8 + x^4 + x^3 + x^2 + 1\right)\\ &= \left(x^6 + x^5 + x^4 + x^3 + x\right) = 0111\ 1010 = \{7A\}\end{aligned}$$

2.8. Urgensi Menjaga Otentitas Al - Quran Digital

2.8.1. Sejarah Perkembangan Al - Quran pada Masa Sahabat

Masyarakat arab dikenal sebagai masyarakat yang memiliki ingatan yang kuat. Mereka juga terbiasa dengan syair - syair, sehingga Al - Quran dapat dengan mudah dihafal. Pada kekhalifahan Abu Bakar ra, Umar bin khattab menyampaikan sebuah ide yang didasari atas keresahannya karena peperangan yang terjadi antara umat muslim dan musuhnya, telah menyebabkan banyak penghafal Al - Quran gugur di medan perang. Umar khawatir, Al - Quran akan hilang bersama dengan para penjaganya.

Atas dasar itu, Abu Bakar memerintahkan para sahabat untuk mengumpulkan naskah - naskah Al - Quran yang tersebar - sebar agar dikumpulkan. Naskah - naskah Al - Quran disebut juga ***Nash***, adalah hasil tulisan para sahabat atas perintah rasulullah ketika suatu wahyu datang. Nash-nash tersebut diabadikan pada batu - batu, kayu - kayu, pelepah - pelepah kurma, hingga tulang belulang. Proses pengumpulan nash Al - Quran diketuai oleh Zaid bin Tsabit. Beliau merupakan seorang pemuda yang paling banyak menuliskan wahyu untuk Rasulullah.

Naskah - naskah yang telah dikumpulkan kemudian diubah menjadi bundelan yang disebut ***Mushaf***. Setelah itu, panitia pengumpulan nash Al - Quran meneliti dan mencocokannya dengan hafalan para sahabat, lalu mengurutkan ayat - ayatnya sesuai yang telah ditetapkan oleh Rasulullah dan menyalinnya ke dalam lembaran - lembaran kertas (Shuhuf) dengan ukuran yang sama. Lembaran - Lembaran tersebut dijahit dengan benang agar tidak ada bagian - bagian yang hilang. Setelah itu, Al - Quran diserahkan kepada Khalifah Abu Bakar ra.

Setelah Abu Bakar wafat, Al - Quran diwariskan kepada khalifah selanjutnya, yaitu Umar bin Khattab. Pada masa pemerintahannya, Umar berfokus pada pengajaran Al - Quran di seluruh penjuru negeri islam. Umar mengirimkan beberapa penghafal Al - Quran ke seluruh penjuru negeri untuk menjadi tenaga pengajar dan berpesan kepada mereka untuk mengajarkan Al - Quran dengan logat (Qiraat) aslinya, yaitu ***Qiraat Quraisy***. Hal itu terus dilakukannya sebagai upaya pelestarian Al - Quran dan penyebaran ajaran islam hingga beliau wafat. Setelah wafatnya, Mushaf Al - Quran diwariskan kepada putrinya, Hafshah, yang merupakan seorang penghafal Al - Quran, qariah dan penulis wahyu. Beliau juga merupakan salah satu istri Rasulullah SAW.

Utsman bin Affan kemudian dibaiat untuk menggantikan Umar bin Khattab. Pada masa pemerintahannya, terjadi pergolakan dalam umat muslim. Hal ini dipicu oleh perbedaan - perbedaan qiraat dalam membaca Al - Quran. Di samping negara yang telah memiliki Mushaf Resmi (yang telah diwariskan dari Abu Bakar hingga Hafsah), ternyata para sahabat juga memiliki mushaf - mushaf pribadi, yang dikumpulkan atas inisiatif dan usaha sendiri. Mushaf - mushaf para sahabat tersebut memiliki perbedaan perbedaan qiraat, sehingga ada beberapa sahabat yang fanatik dengan mushaf yang dimilikinya, sehingga terjadi pertikaian dan saling salah-menyalahkan. [Athaillah, 2010]

Salah satu konflik tersebut terjadi pada kaum muslim Iraq dan Syam. Kaum muslim syam membaca Al - Quran dengan qiraat Ubay bin Kaab yang tidak pernah didengar oleh muslim Iraq. Sebaliknya, Kaum muslim Iraq membaca Al - Quran dengan qiraat Abdullah bin Masud. Akibatnya, mereka saling mengkafirkan kafirkan.

Kejadian itu ternyata juga terjadi di tempat lainnya, sehingga para sahabat yang menyaksikan menganggapnya sebagai ancaman serius yang dapat memecah belah umat islam, sehingga para sahabat melapor masalah tersebut kepada Khalifah Utsman bin Affan. Setelah itu, Utsman berpidato di depan para sahabat dan umat muslim, untuk menyetujui jalan keluar atas masalah tersebut, yaitu dengan membuat Mushaf Imam bagi umat manusia. Dapat disimpulkan, upaya Utsman tersebut adalah menstrandarisasi Al - Quran yang saat itu memiliki ragam qiraat. Usulan Amir al - mukminin ini disetujui oleh para sahabat dan umat islam, sehingga terbentuklah panitia baru untuk membuat Mushaf Imam ini, yang diketuai oleh Zaid bin Tsabbit. Setelah Mushaf Imam dibuat, mushaf - mushaf lain yang tidak sesuai dengan Mushaf Imam dibakar. Mushaf Imam tersebut dikenal dengan ***Mushaf Utsmani***, yang ditulis dengan tulisan ***Kufi***, yaitu tulisan yang dipakai dalam menulis Al - Quran pada masa Rasulullah yang tidak memiliki titik dan baris.

2.8.2. Sejarah Perkembangan Al - Quran Pasca Masa Sahabat

Sepeninggalan khalifah Utsman ra, Mushaf Utsmani masih tetap menjadi Al - Quran standar bagi umat islam, namun perbedaan bacaan dan salah baca tetap terjadi. Hal ini dikarenakan Mushaf Utsmani ditulis menggunakan tulisan Kufi yang tidak memiliki tanda titik dan baris, sehingga sulit untuk dibedakan oleh bangsa lain. Sehubungan dengan hal tersebut, muncullah inisiatif dari para Ulama untuk menyempurnakan tulisan Al - Quran dan penertiban qiraatnya.

Penyempurnaan tulisan dan tanda baca Al - Quran telah melalui beberapa proses dan berakhir secara tuntas atas upaya seorang pakar tata bahasa arab yang termahsyur, al-Khalil bin Ahmad al-Farahidi.

Tanda - tanda tersebut adalah sebagai berikut :

1. Huruf alif kecil yang terletak miring (diagonal) di atas huruf sebagai tanda *Fathah* (konsonan a).
2. Huruf ya kecil yang terletak di bawah huruf sebagai tanda *Kasrah* (konsonan i).
3. Huruf waw kecil yang terletak yang terletak di atas huruf sebagai tanda *Dlammah* (konsonan u).
4. Titik sebagai tanda huruf huruf yang sama bentuknya, seperti ”ba” dengan satu titik di bawahnya. Kemudian ”ta” dengan dua titik di atasnya dan ”tsa” dengan tiga titik di atasnya.

dan seterusnya seperti yang dapat kita lihat saat ini.[Lathif,1972]

Dalam perkembangannya, untuk fathah dan katsrah tidak lagi digunakan huruf alif dan ya kecil, melainkan cukup dengan garis lurus miring dan pendek seperti sekarang, sedangkan dlammah tidak lagi digunakan huruf waw secara utuh, tetapi lengkungan bulat yang ada pada bagian kepalanya. [Athaillah,2010]

2.8.3. Perkembangan Al - Quran Era Modern

Sebelum ditemukannya mesin cetak, Al - Quran diwariskan dan diperbanyak melalui tulisan tangan. Menurut sejarah, Al - Quran pertama kali dicetak dan diterbitkan di Venice sekitar tahun 1530 M, kemudian di Basel pada 1543, tetapi dimusnahkan atas perintah para penguasa gereja. Pada tahun 1694, Al - Quran berhasil dicetak di kota Hamburg. Pada tahun 1698, seorang pendeta Italia, Ludovisi Marraci di Padous Prancis berhasil menyusun naskah Al - Quran

berdasarkan rujukan sejumlah manuskrip dengan disertai terjemahannya dalam bahasa Latin yang cermat. Pada 1787, telah pula lahir percetakan Islam pertama khusus Al - Quran di Saint Peterbourg, kemudian di Qazan, Iran, dan Tibriz.[Shubhi, 1977]

Saat ini, percetakan Al - Quran sudah ada hampir di seluruh negara - negara besar di dunia. Di indonesia sendiri, percetakan Al - Quran harus memiliki perizinan dari pemerintah melalui Departemen Agama. Departemen Agama RI juga memiliki tim khusus, yaitu Lajnah Pentashih Mushaf Al - Quran yang bertugas mengkoreksi dan mentashihkan setiap terbitan Al - Quran.

Selain dicetak, Al - Quran juga diabadikan melalui komputer, yang dikenal dengan Al - Quran Digital. Sejumlah software - software Al - Quran Digital dapat diperoleh dengan mudah, dengan fitur - fitur yang sangat menabjubkan, seperti fitur *Search Engine* yang memudahkan pengguna untuk mencari lokasi suatu ayat atau terjemahannya. Kemudian beberapa *Developer* juga menambahkan fitur Tafsir Quran dari berbagai mufassir, sehingga pengguna dapat mempelajari makna, atau *asbabun nuzul* dari suatu ayat. Dengan begitu, Al - Quran masih tetap terjaga dan terpelihara hingga saat ini.

2.8.4. Menjaga Otentisitas Al - Quran di Era Digital

Proses komputerisasi Al - Quran, ternyata masih memiliki celah kekurangan yang menyebabkan otentisitas Al - Quran terancam. Seperti yang diketahui, bahwa percetakan modern menggunakan komputer dalam melakukan proses pencetakannya. Hal tersebut juga berlaku pada Percetakan Al - Quran. Faktanya, telah ditemukan sejumlah kasus di Indonesia, dimana terdapat kesalahan pada beberapa ayat pada Al - Quran Cetakan. Hal ini disebabkan oleh beberapa

kemungkinan, pertama oknum - oknum yang hendak menyebarkan Quran - Quran palsu untuk membodohi umat. Kedua terjadinya *error* pada komputer yang tidak terdeteksi pada Percetakan.

Gambar 2.2 Kesalahan ayat pada Cetakan Al - Qur'an

Suatu *error* yang dapat mengubah suatu data pada *database* seperti kasus di atas salah satunya dapat disebabkan oleh Virus. Virus merupakan ancaman terbesar yang dapat mengganggu kestabilan sistem komputer, dan integritas data - data pada komputer. Suatu file pada komputer yang terkena Virus, akan mengalami *Corrupt* sehingga dapat menyebabkan kerusakan pada data. Kemungkinan lainnya dapat disebabkan oleh *Human Error*.

Untuk mencegah hal tersebut, penelitian ini akan membahas salah satu upaya untuk menjaga otentitas Al - Quran Digital, yaitu Metode Verifikasi Integritas Data. Verifikasi integritas data adalah metode untuk mengecek otentitas suatu *dataset* atau *database*. Cara ini dilakukan dengan menggunakan teknik kriptografi fungsi hash, yaitu suatu teknik yang merubah sekumpulan data menjadi suatu nilai fix yang disebut Nilai Hash. Metode tersebut dilakukan dengan membandingkan

nilai hash dari *database* Al - Quran Digital yang masih terjaga otentitas nya, dengan nilai hash database Al - Quran Digital yang hendak diuji keotentisitasannya.

Seperti yang telah dibahas sebelumnya, dari sejak zaman para sahabat hingga saat ini, berbagai upaya terus dilakukan dalam menjaga Al - Quran, yang merupakan warisan dari Rasulullah SAW dan karunia yang diberikan Allah kepada Umat Manusia. Sebagai umat muslim, adalah suatu kewajiban untuk terus menjaga dan melestarikannya. Hal tersebut dapat dilakukan dengan berbagai cara, seperti menghafalkannya atau mengajarkannya kepada generasi penerus. Dengan kemajuan teknologi yang sudah berkembang pesat, umat muslim modern harus memperhatikan perkembangan Al - Quran Digital yang saat ini sudah banyak digunakan. Umat muslim perlu waspada terhadap upaya - upaya musuh islam yang selalu memelintirkan ayat - ayat Al - Quran agar umat terpecah belah, dengan membuat ayat - ayat palsu, atau penafsiran - penafsiran liar tanpa ilmu terhadap ayat - ayat Al - Quran.

BAB III

VERIFIKASI INTEGRITAS DATA MENGGUNAKAN FUNGSI HASH KRIPTOGRAFI

Kebutuhan manusia terhadap peralatan elektronik semakin hari semakin tinggi, sejak teknologi - teknologi modern mulai berkembang dari abad 19. Salah satu teknologi yang sampai saat ini berperan besar dalam peradaban umat manusia yaitu internet. Internet merupakan singkatan dari *Interconnection-Networking* adalah seluruh jaringan komunikasi yang menggunakan media elektronik, yang saling terhubung menggunakan standar sistem *global Transmission Control Protocol/Internet Protocol (TCP/IP)* sebagai protokol pertukaran paket (*Packet Switching Communication Protocol*) untuk melayani miliaran pengguna di seluruh dunia.

Internet memungkinkan seluruh penggunanya untuk berkomunikasi jarak jauh. Seiring perkembangannya, Internet mulai diterapkan dalam hal - hal yang berkaitan dengan kegiatan perekonomian, seperti *E-Banking*, *E-Commerce*, dan lain sebagainya. Kegiatan - kegiatan tersebut bersifat sangat vital, sehingga diperlukan suatu sistem keamanan yang dapat menjaga kerahasiaan dan validitas data. Disinilah ilmu kriptografi digunakan dalam menciptakan sistem keamanan digital.

3.1. Kriptografi

Beberapa orang mengatakan bahwa kriptografi dapat dikategorikan sebagai seni. Bahkan di dalam *The Concise Oxford Dictionary (2006)* Kriptografi didefinisikan sebagai *The Art of Writing or Solving Codes*. Kriptografi dikatakan sebuah seni karena setiap orang bisa mengkreasikan cara - cara unik untuk menyembunyikan sebuah pesan. cara - cara yang dimaksud dalam dunia modern dikenal dengan istilah Algoritma.

Kriptografi (*Cryptography*) berasal dari Bahasa yunani, yaitu *crpto* (Rahasia) dan *Graphia* (penulisan). Kriptografi modern didefinisikan sebagai studi terhadap teknis matematis yang terkait dengan aspek keamanan suatu sistem informasi seperti kerahasiaan, integritas data, autentikasi dan ketiadaan penyangkalan *(Menezes,dkk,2006)*. Kriptografi berhubungan erat dengan komunikasi, sehingga ilmu ini sudah berkembang sejak zaman dulu.

Proses penyandian sebuah pesan sudah dikenal sejak zaman yunani kuno. Dalam Buku yang berjudul *Histories*. Herodotus, seorang sejarahwan yunani yang dikenal sebagai *The Father of History* mengatakan bahwa kemenangan yunani atas Persia pada tahun 480-SM disebabkan oleh sebuah pesan rahasia yang dikirim seseorang kepada orang-orang yunani. Pesan tersebut berisikan sebuah peringatan bahwa Raja Persia, Xerxes I, akan melakukan ekspansi ke wilayah yunani.

Kisah lainnya yang terkenal dalam dunia kriptografi adalah pada masa kekaisaran Julius Caesar. Pada masa itu, ditemukan sebuah metode mengirim pesan, dimana tidak akan ada yang dapat mengetahui makna dari suatu pesan, sekalipun pesan tersebut berada ditangannya, kecuali hanya orang yang berhak menerima pesan

tersebut yang dapat mengetahuinya. Inilah awal kemunculan pertama sistem kriptografi, yang dikenal dengan Sandi Caesar. Sandi Caesar adalah teknik kriptografi, yang mensubstitusi pesan asli dengan pergeseran deretan alphabet sebanyak n-kali (ditentukan oleh nilai n).

Pertengahan abad ke-15, Leon Battista Alberti memiliki sebuah gagasan untuk mengenkripsi setiap alphabet dalam sebuah pesan dengan 2 kunci alphabet. Namun Vigenere, pada abad ke-16 mengembangkan gagasannya ini dengan menggunakan 26 Kunci Alphabet. Sistem kriptografi ini dikenal sebagai Sandi Vigenere. Teknik kriptografi ini mengubah setiap alphabet menjadi sebuah nilai dari 0 - 25, yang ditulis sebagai korespondensi antara angka dan alphabet.

Sejak awal ditemukannya ilmu kriptografi, manusia mengimplementasikan kegunaannya untuk kebutuhan perang dan hal - hal intelijensi. Seperti yang terjadi pada perang dunia kedua, ketika jerman membuat sebuah mesin sandi yang sangat fenomemal, Enigma, dan Kemudian muncul seorang matematikawan asal britania, Alan Turring yang berhasil menciptakan sebuah mesin penerjemah sandi enigma yang kemudian menjadi cikal bakal ditemukannya komputer kuantum. Saat ini, kriptografi digunakan dalam berbagai aspek, baik untuk kebutuhan komunikasi maupun transaksi.

Kriptografi memiliki 3 tujuan utama [Ayushi,2010]:

1. ***Confidentiality***

   Confidentiality atau kerahasiaan merupakan perhatian utama ilmu kriptografi. Untuk menjaga kerahasiaan sebuah pesan/data, diciptakanlah sebuah teknik Enkripsi (*Encryption*). Algoritma Enkripsi mengubah sebuah pesan input menjadi sebuah pesan terkunci dan Algoritma Dekripsi yang

berkoresponden dengan algoritma enkripsi digunakan untuk membuka pesan terkunci, mengembalikannya ke dalam bentuk pesan asli.

2. ***Data Integrity***

   Data Integrity atau Integritas data bertujuan untuk memastikan apakah pesan/data terlindungi dengan baik dan tidak mengalami modifikasi. Nilai hash dari data yang diterima akan dicocokan dengan nilai hash data untuk memastikan tidak adanya perubahan terhadap data yang diterima.

3. ***Authentication***

   Authentication atau Autentikasi bertujuan untuk memastikan apakah pesan/data yang diterima berasal dari orang yang benar. Hal ini dikarenakan dapat terjadi kemungkinan, bahwa pesan yang diterima, dikirim oleh orang yang bukan seharusnya (*Third Person*). Tanda Tangan Digital (*Digital Signature*) digunakan untuk memastikannya.

3.2. Kriptografi Modern

Kriptografi Modern merujuk pada semua sistem kriptografi yang mulai berkembangan pada abad ke-19, dimana semua sistemnya didasarkan pada teknik matematika yang complex dan tersistematis. Keamanan suatu pesan, bergantung pada tingkat kesulitan perhitungan matematika pada sistem kriptografinya, dan juga panjang kunci yang digunakan. Sistem Kriptografi Modern terbagi menjadi dua, yaitu Sistem Kriptografi Kunci Simetris dan Sistem Kriptografi kunci Asimetris.

3.2.1. Kriptografi Kunci Simetris

Suatu algoritma kriptografi digolongkan sebagai kriptografi kunci simetris apabila pada proses enkripsi/dekripsi nya menggunakan kunci yang sama (*Shared key*). Kriptografi Kunci Simetris terbagi menjadi 2 jenis, Sandi Blok (*Block Cipher*) dan Sandi Aliran (*Stream Cipher*).

Block Cipher atau Sandi Blok mengubah blok-blok data menjadi blok-blok cipher dalam satu waktu dengan menggunakan suatu kunci. Problem yang terjadi dalam sandi blok adalah permasalahan kerahasian kunci dan pengiriman kunci rahasia melalui jalur komunikasi yang terbuka dan tidak aman. Pertukaran kunci rahasia bukan lagi menjadi masalah karena sudah ditemukan konsep Kriptografi Kunci Publik. Namun seiring perkembangan komputer kuantum, proses pertukaran kunci ini perlu diperhatikan kembali melihat protokol pertukaran kunci RSA yang banyak digunakan sudah tidak aman lagi sejak tahun 2013 lalu.

Pemetaan pada proses enkripsi bersifat bijektif, secara matematis dapat dituliskan sebagai berikut [*Knebl, 2007*]:

$$\mathrm{E} : \mathrm{K} \mathrm{x} \mathbf{M} \rightarrow \mathrm{C}$$

untuk setiap $k \in K$,

$$\mathrm{E}_k : \mathbf{M} \rightarrow \mathbf{C}, \;\; m \rightarrow E\,(k, m)$$

Pemetaan ini bersifat invertibel, Dengan $m \in M$ adalah *Plaintext* (Pesan input), C adalah himpunan *ciphertext* (Pesan Terkunci), dan $k \in K$ adalah himpunan *keys* (kunci). Fungsi Invers,

$$D_k \;\; := \;\; {E_k}^{-1}$$

disebut fungsi dekripsi. Maka algoritma kriptografi yang baik dan bagus adalah algoritma yang memiliki fungsi enkripsi dan dekripsi.

Stream Cipher atau Sandi Aliran adalah suatu sistem kriptografi, dimana setiap digit pada suatu pesan input akan dikombinasikan dengan suatu deret digit kunci yang disebut (*keystream*). Setiap digit pada pesan input, akan dikombinasikan dengan digit kunci koresponden nya dengan suatu operasi yang disebut *Exclusive OR* (XOR). Proses enkripsi pada sandi aliran dapat ditulis dengan persamaan

$$y_i = e_{si}\,(xi) \equiv x_i + s_i \; mod\; 2,$$

dengan x_i adalah digit pesan input, y_i adalah digit cipherteks, dan s_i adalah digit kunci. Proses enkripsi pada sandi aliran dapat ditulis dengan persamaan

$$x_i = d_{si}\,(yi) \equiv y_i + s_i \; mod\; 2.$$

3.2.2. Kriptografi Kunci Asimetris

Kriptografi kunci asimetris lebih dikenal dengan *Public-Key Cryptography* (Kriptografi Kunci Publik) adalah suatu sistem kriptografi yang membahas permasalahan distribusi kunci rahasia untuk melakukan Enkripsi/Dekripsi. Kriptografi kunci asimetris menggunakan 2 buah kunci, yaitu kunci publik dan kunci privat.

Dalam sistem kriptografi kunci asimetris, kunci publik ini dikirimkan secara umum sedangkan kunci privat disimpan sendiri.

Secara sederhana, Alur pertukaran kunci kriptografi kunci asimetris adalah sebagai berikut :

1. Alice membuat sepasang kunci, yaitu kunci private $K_p(A)$ dan kunci publik $K_b(A)$, dimana kunci publik akan dikirimkan kepada Bob, sedangkan kunci privat akan dirahasiakan.

2. Kunci publik $K_b(A)$ dikirimkan kepada Bob melalui jalur komunikasi yang tidak aman, sehingga semua orang dapat melihat kunci publik Alice.

3. Bob menerima kunci publik $K_b(A)$ milik Alice kemudian mengenkripsi sebuah pesan yang hendak dikirim ke Alice menggunakan kunci publik $K_b(A)$ milik Alice.

4. Alice menerima pesan dari Bob, kemudian mendekripsi pesan bob menggunakan kunci privat yang alice buat, yaitu $K_p(A)$

Algoritma kriptografi kunci asimetris yang banyak digunakan saat ini adalah Rivest-Shamir-Adleman (RSA) dan *Elliptic-Curve Cryptography* (ECC).

Pada sistem kriptografi ini, seseorang dapat mengkombinasikan pesan dan kunci rahasia untuk membuat sebuah tanda tangan digital (*Digital Signature*). Tanda tangan digital ini berfungsi sebagai Autentikasi, agar sang penerima pesan dapat mengetahui bahwa pesan yang dikirim kepadanya adalah pesan asli dari orang yang sebenarnya (Bukan dari pihak ke-3). Tanda tangan digital ini diperoleh dengan menggunakan perhitungan kriptografi fungsi hash.

3.2.3. Fungsi Hash

Fungsi Hash adalah suatu fungsi yang memetakan sebuah input data dengan panjang sembarang, menjadi data dengan panjang yang tetap (*fixed*). Fungsi hash didefinisikan sebagai suatu pemetaan,

$$H : M \rightarrow F_{2^n}$$

$$h = H(x)$$

dengan x merupakan *input data*. $H(x)$ disebut Nilai Hash/*Message Digest* dari data x. Proses perhitungan sebuah nilai hash dari suatu pesan/data disebut ***Hashing***.

Ada 3 hal fundamental dalam hal keamanan yang harus dimiliki suatu algoritma fungsi hash [Gauvaram, 2007] :

1. ***Pre-Images Resistance***

   Suatu fungsi hash $H : M \rightarrow F_{2^n}$ dikatakan resistan terhadap *pre-images* apabila untuk suatu $h \in F_{2^n}$, sulit untuk menemukan suatu pesan x, sehingga

   $$H(x) = h.$$

   Inilah yang disebut dengan *one-way function*, yaitu fungsi yang tidak invertibel.

2. ***Second Pre-Images Resistance***

   Suatu fungsi hash dikatakan resistan terhadap *Second pre-images* apabila tidak mudah untuk menghitung suatu nilai hash $H(x)$ dengan mencari input y yang memiliki nilai hash yang sama dengan input x. sehingga,

   $$H(x) = H(y).$$

3. ***Collision Resistance***

   Suatu fungsi hash dikatakan resistan terhadap *Collision* (benturan) apabila tidak mudah untuk mencari dua pesan x dan y, dengan $x \neq y$ sehingga,

   $$H(x) = H(x').$$

   Artinya, suatu algoritma fungsi hash dikatakan *collision resistance* apabila memiliki fungsi yang injektif (satu-satu).

Beberapa penerapan fungsi hash, yaitu :

1. ***Autentikasi Pesan***

   Autentikasi pesan atau *Message Authentication* adalah proses pengidentifikasian sebuah pesan yang diterima, apakah pesan tersebut mengalami perubahan atau modifikasi dan apakah pesan tersebut berasal dari pengirim yang benar (bukan dari pihak ketiga). Autentikasi Pesan dilakukan menggunakan *Message Authentication Codes* (MAC), *Authenticated Encryption* (AE) atau *Digital Signature*

2. ***Verifikasi Password***

   Password digunakan pada proses $Log\ in$. Sistem akan mengecek apakah pengguna yang hendak $Log\ in$ dengan suatu id adalah orang yang asli atau tidak. Seluruh *password* milik pengguna akan disimpan di suatu *database* dalam bentuk nilai hash. Ketika seorang pengguna hendak $Log\ in$, *password* yang dimasukkan pengguna akan dihitung nilai hash nya, kemudian dicocokan dengan kumpulan nilai hash dari *password* yang ada di *database*.

3. ***Mengecek Virus***

   Virus terkadang dapat mengubah suatu data dalam file sehingga file tersebut *corrupt* atau rusak. Dengan mengecek nilai hash nya, maka dapat diketahui, apakah file tersebut telah terinfeksi virus atau tidak. Apabila terinfeksi, maka nilai hash nya akan berbeda dengan file asli yang belum terkena virus.

Pada tahun 1997, Badan Nasional Standar dan Teknologi Amerika Serikat (NIST) mengadakan kerjasama dengan Komunitas Kriptografi Internasional untuk mengembangkan algoritma baru yaitu *Advanced Encryption Standard* (AES).

Sejak saat itu, kriptografi terus mengalami perkembangan. Hal ini dikarenakan, Teknologi Informasi dan Komunikasi sangat penting dalam kehidupan manusia modern. Serangan demi serang terus dilakukan untuk mencari celah dari suatu sistem keamanan sehingga para kriptografer harus terus meneliti, mengembangkan dan menemukan algoritma - algoritma baru untuk dapat bersaing dengan kemajuan komputer kuantum yang sekarang sudah sangat canggih. Bukti nyata dari dampak perkembangan komputer kuantum adalah sistem Pertukaran kunci public seperti *Elliptic-Curve Cryptography* (ECC) dan RSA saat ini dinyatakan sudah tidak aman lagi, Sehingga kriptografi sekarang ini berkembang menuju babak baru, yaitu *Quantum Cryptography*.

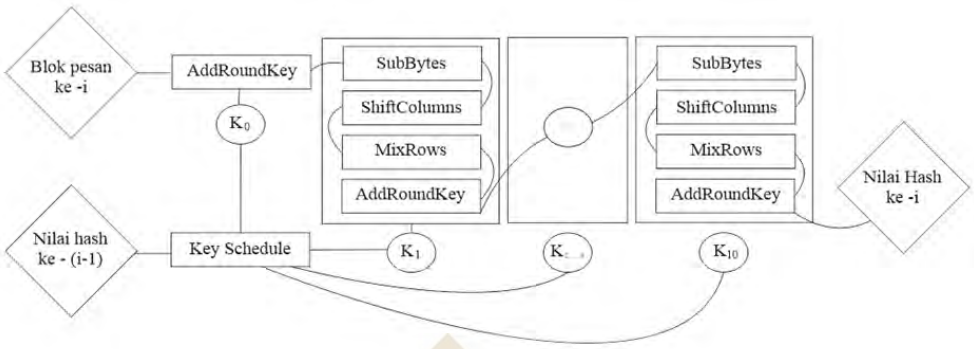
3.3. WHIRLPOOL

Algoritma fungsi hash Whirlpool dirancang oleh Vincent Rijmen, seorang kriptografer asal belgia bersama dengan rekannya, Paulo S.L, yang berasal dari Brazil. Vincent Rijmen merupakan salah satu perancang dari algoritma kriptografi Rijndael yang kemudian dinobatkan sebagai AES (*Advanced Encryption Standard*). Algoritma Whirlpool dirancang khusus sebagai algoritma fungsi hash satu arah (*One Way Hash Function*) yang menghasilkan pesan digest 512-bit dengan panjang pesan maksimal yaitu 2^{256}-bits.

Algoritma Whirlpool adalah fungsi hash berbasis *Block Cipher*, dimana setiap prosesnya menggunakan sistem kunci simetris yang diadopsi dari AES. Algoritma ini dirancang agar memiliki keamanan dan efisiensi seperti AES, namun memiliki panjang hash yang setara dengan SHA (*Security Hash Function*) agar dapat meningkatkan keamanan pesan.

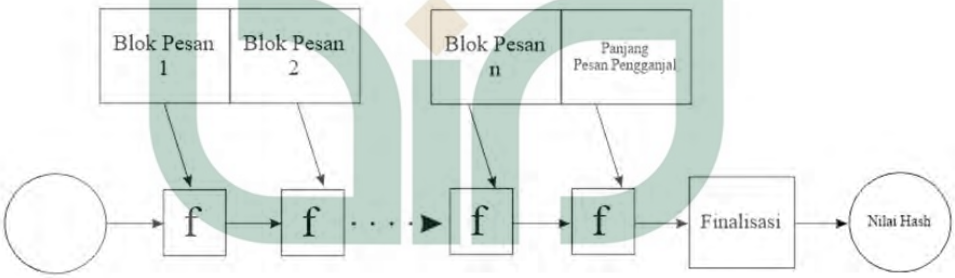
3.3.1. Struktur Algoritma

Whirlpool memiliki 4 fungsi utama yang digunakan dalam perhitungan nilai hash. Fungsi-fungsi tersebut terdiri dari: *SubByte, ShiftColumns, MixRows, dan AddRoundKey*.



Gambar 3.1 Skema proses *hashing* pada algoritma Whirlpool

Whirpool adalah sebuah fungsi hash berbasis *blockcipher*. Whirlpool menggunakan kontruksi Merkle-Damgard yang mengubah pesan input dengan panjang sembarang, menjadi suatu input yang memiliki panjang *Message Digest* yang tetap.



Gambar 3.2 Sistem kontruksi *Block Message* Merkle-Damgard

Akibatnya, Pesan input akan dibagi menjadi blok - blok pesan dengan panjang yang sama.

Pada Algoritma Whirlpool, pesan input akan melalui 3 tahapan sebelum diperoleh nilai hashnya, yaitu:

1. **Penambahan bit-bit pada pesan masukan (Proses *Padding Bits*)**

   Sebuah pesan *input* akan mengalami penambahan bit - bit yang disebut dengan proses *padding bits* sebelum dilakukan proses *hashing*. Hal ini bertujuan agar pesan *input* mempunyai panjang pesan berkelipatan ganjil dari 256-bits. Bit - bit yang ditambahkan pada pesan terdiri dari sebuah bit ”1” diikuti dengan bit ”0”. [Stalling,2005]

   Blok sisa dengan ukuran 256-bits akan dianggap sebagai *unsigned 256-bit integer* yang akan diisi dengan panjang pesan input yang asli. Selanjutnya, blok - blok pesan akan disusun menjadi sebuah matriks berukuran 8x8 byte dan diurut secara secara baris.

2. **Pembentukan matriks hash awal *(Initial Hash)***

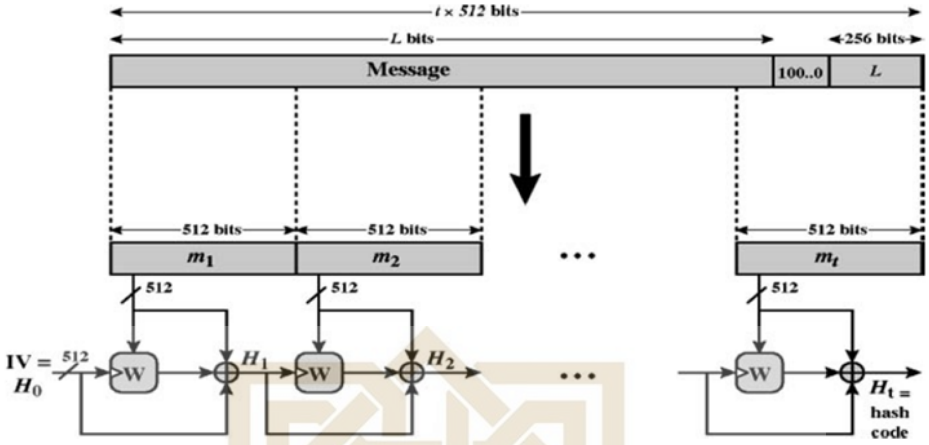
   Matriks Hash H_0 adalah matriks inisiasi berukuran 8x8 Byte yang akan digunakan pada permulaan proses *AddRoundKey*. Matriks ini berisikan bits-0.

3. **Proses pengolahan blok pesan 512-bits (64-Byte)**

   Blok - blok pesan yang telah terbentuk dari proses *padding bits* selanjutnya akan diproses dengan 4 tahapan utama algoritma Whirlpool, yaitu *SubByte, ShiftColumns, MixRows, dan AddRoundKey*.

3.3.2. Pembentukan Blok Cipher W

Pembentukan blok cipher W adalah proses dimana blok - blok pesan akan diolah dengan 4 fungsi utama algoritma sehingga menjadi sebuah output berupa blok



Gambar 3.3 Skema algoritma fungsi hash Whirlpool [Stalling,2005]

pesan yang terenkripsi atau *ciphertext*. Fungsi *f* pada Gambar 3.2. adalah fungsi kompresi Miyaguchi-Preneel. Fungsi kompresi Miyaguchi-Preneel merupakan fungsi kompresi blok pesan yang di desain oleh Shoji Miyaguchi dan Bart Preneel. Setiap blok pesan akan diolah menjadi blok cipher yang kemudian dijumlahkan. Hasil penjumlahan kemudian dijumlahkan dengan nilai hash sebelumnya.

Gambar 3.4 Skema fungsi kompresi Miyaguchi-Preneel

Skema fungsi hash Whirlpool dapat ditulis sebagai berikut:

$$H_0 = Nilai\ Hash\ Awal\ (\textbf{\textit{Initiation Hash}})$$

$$H_i = E\left(H_{i-1},\ m_i\right) \oplus H_{i-1} \oplus m_i = Nilai\ Pertengahan$$

$$H_t = Nilai\ Hash\ Final$$

Algoritma enkripsi pada Whirlpool, mengolah blok pesan berukuran 512-bits menggunakan kunci berukuran 512-bits agar menghasilkan sebuah blok cipherteks berukuran 512-bits. Setiap putaran r pada algoritma, dapat dituliskan menjadi sebuah Fungsi $RF(RoundFunction)$, yaitu suatu fungsi komposisi, dimana:

$$RF\left(K_r\right) = AK\left[K_r\right] \circ MR \circ SC \circ SB.$$

Keterangan:

AK = Fungsi *AddRoundKey*

MR = Fungsi *MixRows*

SC = Fungsi *ShiftColumns*

SB = Fungsi *SubByte*

K_r = Kunci pada putaran ke-i

Sehingga secara keseluruhan, Algoritma Enkripsi pada Whirlpool dapat dituliskan ke dalam bentuk fungsi, yaitu:

$$W(K) = \left(\bigcirc_{r=1}^{10} RF(K_r)\right) \circ AK(K_0).$$

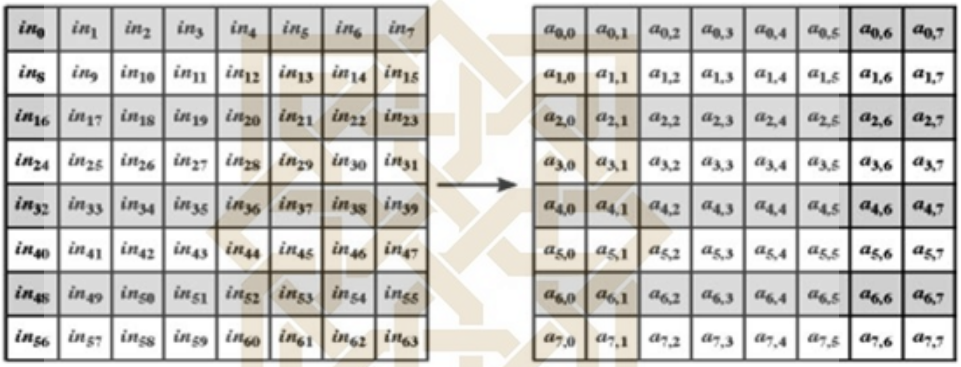
Pada setiap iterasi/putaran, Pesan input yang diolah adalah suatu matriks berukuran 8x8 disebut ***CState***.

CState merupakan sebuah matriks berukuran 8x8 yang setiap entri nya merupakan hasil pemetaan dari sebuah blok pesan dalam bentuk matriks. Setiap entri pada matriks *CState*, dapat dituliskan sebagai suatu korespondensi, yaitu

$$A = \mu\left(X\right) \Leftrightarrow\ a_{i,j} = x_{8i+j}.$$

Dimana X adalah susunan Byte linier pada pesan input dengan elemen $x_k \quad (0 \leq k \leq 63)$, dan A adalah matrik *CState* yang merupakan matrik korespondensi dari X dengan elemen $a_{i,j} \ (0 \leq i, \ \ j \leq 7)$.

Dengan kata lain, Setiap kata pada pesan input akan disusun berurut secara baris ke dalam Matriks CState.



Gambar 3.5 Korespondensi antara matrik pesan input dan matriks CState

1) Fungsi Substitusi Byte (*SubByte*)

Pada proses ini, algoritma akan memetakan setiap entri pada matrik *CState* ke sebuah table berukuran 16x16 yang disebut dengan ***S-Box***. Proses ini merupakan pemetaan non-linier, yang nantinya akan mengubah Byte pada setiap entri matrik CState menjadi Byte-Byte baru.

Setiap entri pada matrik input memiliki ukuran 8-bits (1-Byte). 4-bits yang berada di sebelah kiri akan dipandang sebagai baris pada kotak S-Box, sedangkan 4 bits yang berada di sebelah kanan akan dipandang sebagai kolom pada kotak S-box.

Fungsi *SubByte* dapat ditulis sebagai suatu korespondensi antara Matrik input A dan Matrik Output B, yaitu :

$$B = SubByte\,(A) \Leftrightarrow \ b_{i,j} = S\,[a_{i,j}]\,, \quad 0 \leq i, \quad j \leq 7$$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	18	23	C6	E8	87	B8	01	4F	36	A6	D2	F5	79	6F	91	52
1	60	BC	B	8E	A3	0C	7B	35	1D	E0	D7	C2	2E	4B	FE	57
2	15	77	37	E5	9F	F0	4A	CA	58	C9	29	0A	B1	A0	6B	85
3	BD	5D	10	F4	CB	3E	05	67	E4	27	41	8B	A7	7D	95	C8
4	FB	EE	7C	66	DD	17	47	9E	CA	2D	BF	07	AD	5A	83	33
5	63	02	AA	71	C8	19	49	C9	F2	E3	5B	88	9A	26	32	B0
6	E9	0F	D5	80	BE	CD	34	48	FF	7A	90	5F	20	68	1A	AE
7	B4	54	93	22	64	F1	73	12	40	08	C3	EC	DB	A1	8D	3D
8	97	00	CF	2B	76	82	D6	1B	B5	AF	6A	50	45	F3	30	EF
9	3F	55	A2	EA	65	BA	2F	C0	DE	1C	FD	4D	92	75	06	8A
A	B2	E6	0E	1F	62	D4	A8	96	F9	C5	25	59	84	72	39	4C
B	5E	78	38	8C	C1	A5	E2	61	B3	21	9C	1E	43	C7	FC	04
C	51	99	6D	0D	FA	DF	7E	24	3B	AB	CE	11	8F	4E	B7	EB
D	3C	81	94	F7	B9	13	2C	D3	E7	6E	C4	03	56	44	7F	A9
E	2A	BB	C1	53	DC	0B	9D	6C	31	74	F6	46	AC	89	14	E1
F	16	3A	69	09	70	B6	C0	ED	CC	42	98	A4	28	5C	F8	86

Gambar 3.6 S-Box Algoritma Whirlpool

Sebagai contoh, untuk suatu entri pada *CState* yang memiliki nilai hex $3F$, mengartikan baris ke 3 dan kolom ke F pada S-Box yaitu $C8$. Akibatnya, entri pada *CState* yang memiliki nilai hex $3F$ akan berubah menjadi $C8$ setelah diproses dengan fungsi *SubByte*.

Perhitungan S-Box Whirlpool

Setiap entri pada S-Box merupakan hasil operasi pada Lapangan Hingga $GF(2^4)$ atas polinomial yang tak tereduksi (*irreducible polynomial*) $r(x) = x^4 + x + 1$. Berikut ini adalah perhitungan dari suatu entri pada S-Box.

Diberikan u sebagai suatu digit heksadesimal, maka:

$$E(u) = \begin{cases} , jika\ u = F^{(x^3+x+1)^u \bmod r(x),\ \ jika\ u \neq F} \\ 0 \end{cases}$$

Diperoleh sebuah pemetaan dari $GF(2^4) \rightarrow GF(2^4)$ yang disebut dengan Mini-Box E.

u	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E(u)	1	B	9	C	D	6	F	3	E	8	7	4	A	2	5	0

Gambar 3.7 Mini-Box E

Contoh Perhitungan $E(u)$:

Jika $u = 1$, maka:

$$E(u) = \left(x^3 + x + 1\right)^1 \, mod \, x^4 + x + 1 = 1$$
$$= x^3 + x + 1 = 1011 = B.$$

Jika $u = 2$, maka:

$$E(u) = \left(x^3 + x + 1\right)^2 \, mod \, x^4 + x + 1 = 1$$
$$= \left(x^3 + x + 1\right) \bullet \left(x^3 + x + 1\right) \, mod \, x^4 + x + 1$$
$$= \left(x^6 + x^4 + x^3 + x^4 + x^2 + x + x^3 + x + 1\right) \, mod \, x^4 + x + 1$$
$$= \left(x^6 + x^2 + 1\right) \, mod \, x^4 + x + 1$$
$$= x^3 + 1 = (0000\ 1001) = 9.$$

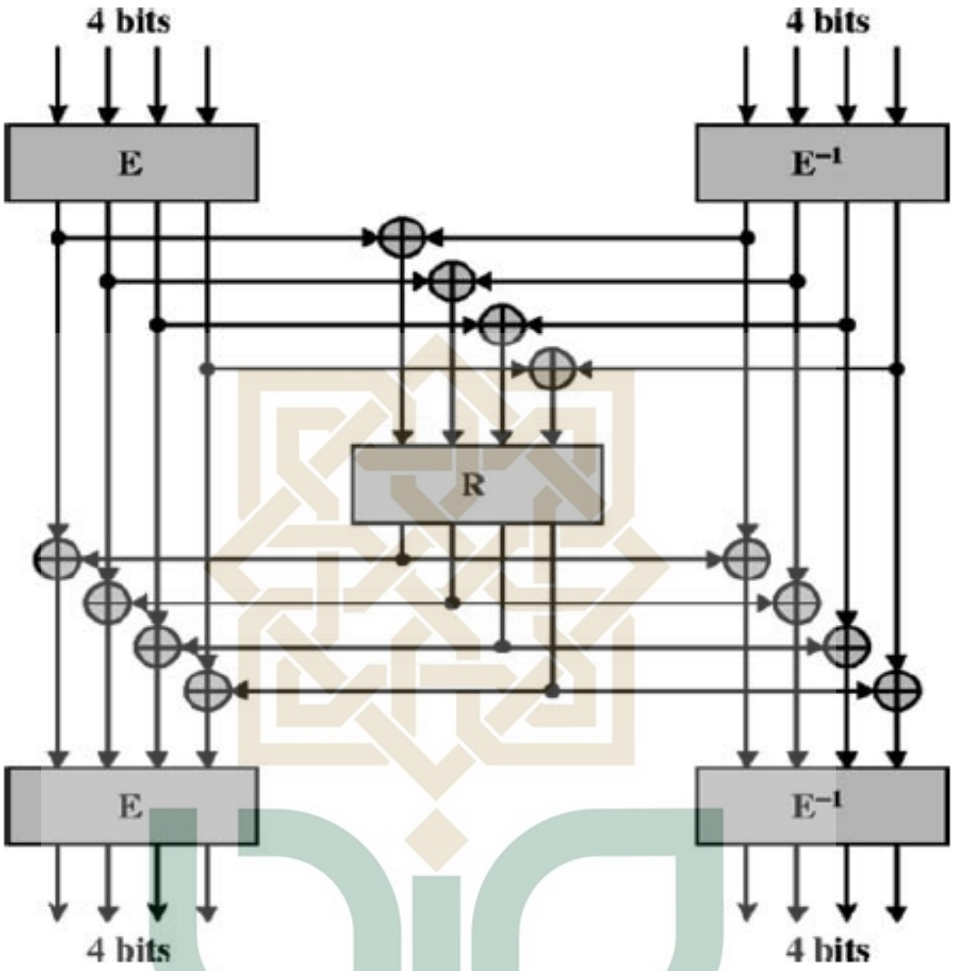
Mini-Box E digunakan untuk membangun S-Box pada Algoritma Whirlpool. Proses kontruksi S-Box pada Whirlpool melibatkan 3 komponen, yaitu Mini-Box E, Inverse Mini-Box E, dan R Mini-Box. R Mini-Box diperoleh dari permutasi secara *pseudorandomly* (Acak Semu).

u	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$E^{-1}(u)$	F	0	D	7	B	E	5	A	9	2	C	1	3	4	8	6

Gambar 3.8 Inverse Mini-Box E

u	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
R(u)	7	C	B	D	E	4	9	F	6	3	8	A	2	5	1	0

Gambar 3.9 R Mini-Box E



Gambar 3.10 Skema pembentukan S-Box

Ketiga komponen ini kemudian diolah dengan proses penjumlahan secara rekrusif sehingga seluruh entri pada S-box diperoleh berdasarkan perhitungan berikut.

$$i, j = 0,\ 1,\ 2,\ \ldots .,\ F$$

$$y = E\,(i) \oplus E^{-1}\,(i)$$

$$Z_1 = R\,(y) \oplus E\,(i)$$

$$Z_2 = R\,(y) \oplus\ E^{-1}\,(j)$$

$$S_{i,j} = E\,(Z_1)\,E^{-1}\,(Z_2)$$

Berikut adalah contoh perhitungan suatu entri pada S-Box:

Untuk entri $S_{1,1}$ pada S-Box, maka

$$y = E(i) \oplus E^{-1}(i) = E(1) \oplus E^{-1}(1) = 1011 \oplus 0000 = 1011 = B$$

$$R(y) = R(1011) = 1010 = A$$

$$Z_1 = R(y) \oplus E(i) = 1010 \oplus 1011 = 0001$$

$$Z_2 = R(y) \oplus E^{-1}(j) = 1010 \oplus 0000 = 1010$$

$$S_{1,1} = E(Z_1)\, E^{-1}(Z_2) = E(0001)\, E^{-1}(1010) = BC$$

Sehingga diiperoleh nilai hex pada entri $S_{1,1}$ adalah BC.

2) Fungsi Permutasi (*ShiftColumns*)

Fungsi ini mengubah susunan entri secara permutasi yaitu dengan menggeser setiap kolom memutar secara berurutan. Semua kolom pada matrik CState kecuali kolom pertama akan diputar ke bawah. Untuk kolom kedua, akan diputar 1-byte, kolom ketiga akan diputar 2-byte, kolom keempat akan diputar 3-byte dan seterusnya hingga kolom kedelapan. Hasil permutasi dari fungsi ini akan membentuk suatu matrik baru. Fungsi *ShiftColumn* dapat ditulis sebagai korespondensi dari matrik input A dan matrik output B:

$$B = ShiftColumn(A) \Leftrightarrow b_{i,j} = a_{(i-j) mod\ 8,\ j},\quad 0 \leq i,\quad j \leq 7$$

Gambar 3.11 Skema Fungsi *ShiftColumns*

3) Fungsi Difusi (*MixRows*)

Proses difusi pada kriptografi pertama kali dikenalkan oleh seorang matematikawan asal America, Claude Elwood Shannon dalam papernya yang berjudul *Communication Theory of Secrecy System*, pada tahun 1949. Proses difusi memiliki fungsi untuk menghilangkan redudansi pada data statistik plainteks agar tidak terdeteksi pada cipherteks. Fungsi ini bertujuan agar cipherteks lebih sulit untuk dideteksi dan kebal terhadap serangan kriptografi yang berbasis pada pengamatan pola kemunculan kata (*Statistical Attack*).

Proses ini dilakukan dengan mengubah setiap Byte pada entri matrik, menjadi suatu fungsi yang merupakan hasil pemetaan dari Byte - Byte pada suatu baris. Transformasi matrik tersebut dapat ditulis dengan persamaan

$$B = AC,$$

dengan output matrik B, diperoleh dari perkalian matrik input A dan matrik transformasi C, yaitu:

$$C = \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}$$

Contoh perhitungan Mixrows :

Diberikan suatu baris dari matrik CState, $[a_{0,0}, a_{0,1}, a_{0,2}, a_{0,3}, a_{0,4}, a_{0,5}, a_{0,6}, a_{0,7}]$ yaitu,

$$[B2\ \ 1F\ \ 3D\ \ 5E\ \ 2F\ \ D4\ \ 5A\ \ 6F]$$

Maka perhitungan *MixRows* pada baris tersebut adalah

$$[B2\ \ 1F\ \ 3D\ \ 5E\ \ 2F\ \ D4\ \ 5A\ \ 6F] \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}.$$

Untuk entri $b_{0,0}$ pada matrik output B,

$$b_{0,0} = (B2) \oplus (1F \bullet 9) \oplus (3D \bullet 02) \oplus (5E \bullet 05) \oplus (2F \bullet 08) \oplus (D4) \oplus (5A \bullet 04) \oplus (6F\}$$

$$(1F \bullet 09) = (0001\ 1111) \bullet (0000\ 1001)\ mod\ (100011101)$$

$$= \left(x^4 + x^3 + x^2 + x + 1\right) \bullet \left(x^3 + 1\right) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^7 + x^6 + x^5 + x^2 + x + 1) mod (x^8 + x^4 + x^3 + x^2 + 1)$$

$$= (x^7 + x^6 + x^5 + x^2 + x + 1) = 1110\ 0111 = \{E7\}$$

$$(3D \bullet 02) = (0011\ 1101) \bullet (0000\ 0010)\ mod\ (100011101)$$

$$= (x^5 + x^4 + x^3 + x^2 + 1) \bullet (x)\ mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^6 + x^5 + x^4 + x^3 + x) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= \left(x^6 + x^5 + x^4 + x^3 + x\right) = 0111\ 1010 = \{7A\}$$

$$(5E \bullet 05) = (0101\ 1110) \bullet (0000\ 0101)\ mod (100011101)$$

$$= (x^6 + x^4 + x^3 + x^2 + x) \bullet \left(x^2 + 1\right) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^8 + x^5 + x^2 + x) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^5 + x^4 + x^3 + x + 1) = \{3B\}$$

$$(2F \bullet 08) = \ (0010\ 1111) \bullet (0000\ 1000)\ mod (10001101)$$

$$= (x^5 + x^3 + x^2 + x + 1) \bullet \left(x^3\right) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^8 + x^6 + x^5 + x^4 + x^3) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^6 + x^5 + x^2 + 1) = \{65\}$$

$$(5A \bullet 04) = (0101\ 1010) \bullet (0000\ 0100)\ mod (10001101)$$

$$= (x^6 + x^4 + x^3 + x) \bullet \left(x^2\right) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^8 + x^6 + x^5 + x^3) mod\ \left(x^8 + x^4 + x^3 + x^2 + 1\right)$$

$$= (x^6 + x^4 + x^5 + x^2 + 1) = \{75\}$$

$$b_{0,0} = (B2) \oplus (1F \bullet 9) \oplus (3D \bullet 02) \oplus (5E \bullet 05) \oplus (2F \bullet 08) \oplus (D4) \oplus (5A \bullet 04) \oplus (6F)$$

$$= \{B2\} \oplus \{E7\} \oplus \{7A\} \oplus \{3B\} \oplus \{65\} \oplus \{D4\} \oplus \{75\} \oplus \{6F\}$$

$$= (\{1011\ 0010\} \oplus \{1110\ 0111\}) \oplus (\{0111\ 1010\} \oplus \{0011\ 1011\}) \oplus$$

$$(\{0110\ 0101\} \oplus \{1101\ 0100\}) \oplus (\{0111\ 0101\} \oplus \{0110\ 1111\})$$

$$= (\{0101\ 0101\} \oplus \{0100\ 0001\}) \oplus (\{1011\ 0001\} \oplus \{0001\ 1010\})$$

$$= \{0001\ 0101\} \oplus \{1010\ 1010\}$$

$$= \{1011\ 1111\} = \{BF\}$$

Dari perhitungan di atas, diperoleh bahwa entri $b_{0,0}$ pada matrik output B adalah $\{BF\}$.

Perhitungan Matriks Konstant *MixRows*:

Proses difusi ini bertujuan untuk menghasilkan suatu nilai yang merupakan kombinasi dari nilai - nilai lain. Operasi difusi terdiri dari perkalian modulo dan 2 *eight-term polynomial* dimana koefisien koefisiennya adalah elemen dari $GF(2^8)$. Modulo yang digunakan adalah $x^8 + 1$. Hal ini bertujuan agar operasi difusi menghasilkan polinomial dengan degree yang lebih kecil dari 8. Misalkan *eight-term polynomial* yang pertama, didefinisikan sebagai suatu baris $[b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0]$ yang memuat 8-Byte.

Setiap Byte adalah koefisien dari *eight-term* sebagai berikut.

$$b\,(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 \;.....(i)$$

eight-term polynomial kedua didefinisikan sebagai suatu polinomial konstan, yaitu

$$a(x) = 1x^7 + 1x^6 + 4x^5 + 1x^4 + 8x^3 + 5x^2 + 2x + 9 \;.....(ii)$$

Atau dapat ditulis sebagai

$$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \;.....(iii)$$

Dari persamaan (i) dan (iii), diperoleh

$$\begin{aligned} a\,(x) \;\oplus\; b\,(x) &= \left(a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0\right) \\ &\quad \oplus \left(b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0\right) \\ &= \left(a_7 \oplus b_7\right)x^7 + \left(a_6 \oplus b_6\right)x^6 + \left(a_5 \oplus b_5\right)x^5 + \left(a_4 \oplus b_4\right)x^4 \oplus \\ &\quad \left(a_3 \oplus b_3\right)x^3 \oplus \left(a_2 \oplus b_2\right)x^2 \oplus \left(a_1 \oplus b_1\right)x \oplus \left(a_0 \oplus b_0\right) \end{aligned}$$

$$\begin{aligned} a\,(x) \bullet b\,(x) &= \left(a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0\right) \\ &\quad \bullet \left(b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0\right) \\ &= c(x) \\ &= c_{14}x^{14} + c_{13}x^{13} + c_{12}x^{12} + c_{11}x^{11} + c_{10}x^{10} + c_9x^9 + c_8x^8 \\ &= +c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 \end{aligned}$$

dengan,

$c_0 = a_0 \bullet b_0$

$c_1 = a_1 \bullet b_0 \oplus a_0 \bullet b_1$

$c_2 = a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2$

$c_3 = a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3$

$c_4 = a_4 \bullet b_0 \oplus a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3 \oplus a_0 \bullet b_4$

$c_5 = a_5 \bullet b_0 \oplus a_4 \bullet b_1 \oplus a_3 \bullet b_2 \oplus a_2 \bullet b_3 \oplus a_1 \bullet b_4 \oplus a_0 \bullet b_5$

$c_6 = a_6 \bullet b_0 \oplus a_5 \bullet b_1 \oplus a_4 \bullet b_2 \oplus a_3 \bullet b_3 \oplus a_2 \bullet b_4 \oplus a_1 \bullet b_5 \oplus a_0 \bullet b_6$

$c_7 = a_7 \bullet b_0 \oplus a_6 \bullet b_1 \oplus a_5 \bullet b_2 \oplus a_4 \bullet b_3 \oplus a_3 \bullet b_4 \oplus a_2 \bullet b_5 \oplus a_1 \bullet b_6 \oplus a_0 \bullet b_7$

$c_8 = a_7 \bullet b_1 \oplus a_6 \bullet b_2 \oplus a_5 \bullet b_3 \oplus a_4 \bullet b_4 \oplus a_3 \bullet b_5 \oplus a_2 \bullet b_6 \oplus a_1 \bullet b_7$

$c_9 = a_7 \bullet b_2 \oplus a_6 \bullet b_3 \oplus a_5 \bullet b_4 \oplus a_4 \bullet b_5 \oplus a_3 \bullet b_6 \oplus a_2 \bullet b_7$

$c_{10} = a_7 \bullet b_3 \oplus a_6 \bullet b_4 \oplus a_5 \bullet b_5 \oplus a_4 \bullet b_6 \oplus a_3 \bullet b_7$

$c_{11} = a_7 \bullet b_4 \oplus a_6 \bullet b_5 \oplus a_5 \bullet b_6 \oplus a_4 \bullet b_7$

$c_{12} = a_7 \bullet b_5 \oplus a_6 \bullet b_6 \oplus a_5 \bullet b_7$

$c_{13} = a_7 \bullet b_6 \oplus a_6 \bullet b_7$

$c_{14} = a_7 \bullet b_7$

Diketahui bahwa, $a(x) \bullet b(x) = c(x)\ mod\ x^8 + 1$, maka :

$x^{14}\ mod\ x^8 + 1 = (x^6)(x^8 + 1) - x^6 = -x^6 = x^6\ atas\ GF(2^8).$

$x^{13}\ mod\ x^8 + 1 = (x^5)(x^8 + 1) - x^5 = -x^5 = x^5\ atas\ GF(2^8).$

$x^{12}\ mod\ x^8 + 1 = (x^4)(x^8 + 1) - x^4 = -x^4 = x^4\ atas\ GF(2^8).$

$x^{11}\ mod\ x^8 + 1 = (x^3)(x^8 + 1) - x^3 = -x^3 = x^3\ atas\ GF(2^8).$

$x^{10}\ mod\ x^8 + 1 = (x^2)(x^8 + 1) - x^2 = -x^2 = x^2\ atas\ GF(2^8).$

$x^9\ mod\ x^8 + 1 = (x)(x^8 + 1) - x = -x = x\ atas\ GF(2^8).$

$x^8\ mod\ x^8 + 1 = (1)(x^8 + 1) - 1 = -1 = 1\ atas\ GF(2^8)$

Dari penjabaran di atas, dapat dilihat bahwa

$$x^i \bmod \left(x^8+1\right) = x^{i \bmod 8}$$

sehingga,

$$\begin{aligned}
a(x) \bullet b(x) &= c(x) \ \ mod\ x^8+1 \\
&= (c_{14}x^{14} + c_{13}x^{13} + c_{12}x^{12} + c_{11}x^{11} + c_{10}x^{10} + c_9x^9 + c_8x^8 + c_7x^7 + c_6x^6 + c_5x^5 \\
&+ c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0)\ mod\ (x^8+1) \\
&= (c_{14}x^{14 mod 8} + c_{13}x^{13 mod 8} + c_{12}x^{12 mod 8} + c_{11}x^{11 mod 8} + c_{10}x^{10 mod 8} + c_9x^{9 mod 8} \\
&+ c_8x^{8 mod 8} + c_7x^{7 mod 8} + c_6x^{6 mod 8} + c_5x^{5 mod 8} + c_4x^{4 mod 8} + c_3x^{3 mod 8} + c_2x^{2 mod 8} \\
&+ c_1x^{1 mod 8} + c_0{}^{0 mod 8}) \\
&= (c_{14}x^6 + c_{13}x^5 + c_{12}x^4 + c_{11}x^3 + c_{10}x^2 + c_9x + c_8 + c_7x^7 + c_6x^6 + c_5x^5 + c_4x^4 \\
&+ c_3x^3 + c_2x^2 + c_1x + c_0) \\
&= \left(c_7\right)x^7 + \left(c_{14} \oplus c_6\right)x^6 + \left(c_{13} \oplus c_5\right)x^5 + \left(c_{12} \oplus c_4\right)x^4 + \left(c_{11} \oplus c_3\right)x^3 \\
&+ \left(c_{10} \oplus c_2\right)x^2 + \left(c_9 \oplus c_1\right)x + \left(c_8 \oplus c_0\right) \\
&= d_7x^7 + d_6x^6 + d_5x^5 + d_4x^4 + d_3x^3 + d_2x^2 + d_1x + d_0
\end{aligned}$$

Koefisien dari $d_7, d_6, d_5, d_4, d_3, d_2, d_1, d_0$, dapat ditulis menjadi

$$d_7 = c_7$$

$$= a_7 \bullet b_0 \oplus a_6 \bullet b_1 \oplus a_5 \bullet b_2 \oplus a_4 \bullet b_3 \oplus a_3 \bullet b_4 \oplus a_2 \bullet b_5 \oplus a_1 \bullet b_6 \oplus a_0 \bullet b_7$$

$$d_6 = c_6 \oplus c_{14}$$

$$= (a_6 \bullet b_0 \oplus a_5 \bullet b_1 \oplus a_4 \bullet b_2 \oplus a_3 \bullet b_3 \oplus a_2 \bullet b_4 \oplus a_1 \bullet b_5 \oplus a_0 \bullet b_6) \oplus (a_7 \bullet b_7)$$

$$d_5 = c_5 \oplus c_{13}$$

$$= (a_5 \bullet b_0 \oplus a_4 \bullet b_1 \oplus a_3 \bullet b_2 \oplus a_2 \bullet b_3 \oplus a_1 \bullet b_4 \oplus a_0 \bullet b_5) \oplus (a_7 \bullet b_6 \oplus a_6 \bullet b_7)$$

$$d_4 = c_4 \oplus c_{12}$$

$$= (a_4 \bullet b_0 \oplus a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3 \oplus a_0 \bullet b_4) \oplus (a_7 \bullet b_5 \oplus a_6 \bullet b_6 \oplus a_5 \bullet b_7)$$

$$d_3 = c_3 \oplus c_{11}$$

$$= (a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3) \oplus (a_7 \bullet b_4 \oplus a_6 \bullet b_5 \oplus a_5 \bullet b_6 \oplus a_4 \bullet b_7)$$

$$d_2 = c_2 \oplus c_{10}$$

$$= (a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2) \oplus (a_7 \bullet b_3 \oplus a_6 \bullet b_4 \oplus a_5 \bullet b_5 \oplus a_4 \bullet b_6 \oplus a_3 \bullet b_7)$$

$$d_1 = c_1 \oplus c_9$$

$$= (a_1 \bullet b_0 \oplus a_0 \bullet b_1) \oplus (a_7 \bullet b_2 \oplus a_6 \bullet b_3 \oplus a_5 \bullet b_4 \oplus a_4 \bullet b_5 \oplus a_3 \bullet b_6 \oplus a_2 \bullet b_7)$$

$$d_0 = c_0 \oplus c_8$$

$$= a_0 \bullet b_0) \oplus (a_7 \bullet b_1 \oplus a_6 \bullet b_2 \oplus a_5 \bullet b_3 \oplus a_4 \bullet b_4 \oplus a_3 \bullet b_5 \oplus a_2 \bullet b_6 \oplus a_1 \bullet b_7)$$

Dengan mensubstitusikan koefisien $d(x)$ dengan persamaan (ii), yaitu $a(x) = 1x^7 + 1x^6 + 4x^5 + 1x^4 + 8x^3 + 5x^2 + 2x + 9$ maka dapat dibentuk suatu representasi matriks

$$[d_7\ d_6\ d_5\ d_4\ d_3\ d_2\ d_1\ d_0] = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}.$$

Diperoleh sebuah matriks MDS berukuran 8x8.

4) Fungsi *AddRoundKeys*

Pada setiap putarannya, algoritma akan menambahkan *Round Key* yang diperoleh dari proses fungsi *KeyExpansion*. Dengan kata lain, Matriks *CState* akan mengalami proses XOR dimana setiap Byte-nya akan dijumlahkan dengan Byte - byte pada matrik *Round key*. Fungsi *AddKey* dapat ditulis sebagai korespondensi dari matrik input A dan matrik output B:

$$B = AddKey\,[k_i]\,(A) \leftrightarrow\ b_{i,j} = a_{i,j} \oplus k_{i,j}, \quad 0 \leq i, \quad j \leq 7.$$

Contoh proses *AddRoundKey*:

Diambil suatu kolom a dari matrik CState dan kolom koresponden k di matriks kunci K:

$$a = [B2\ \ 1F\ \ 3D\ \ 5E\ \ 2F\ \ D4\ \ 5A\ \ 6F]$$

$$k = [B3\ \ 5F\ \ 1A\ \ 23\ \ 41\ \ DF\ \ 6A\ \ 2F]$$

kolom koresponden pada matrik b diperoleh dari perhitungan berikut:

$$b = a \oplus k = \begin{bmatrix} B2 \\ 1F \\ 3D \\ 5E \\ 2F \\ D4 \\ 5A \\ 6F \end{bmatrix} \oplus \begin{bmatrix} B3 \\ 5F \\ 1A \\ 23 \\ 41 \\ DF \\ 6A \\ 2F \end{bmatrix} = \begin{bmatrix} B2 \oplus B3 \\ 1F \oplus 5F \\ 3D \oplus 1A \\ 5E \oplus 23 \\ 2F \oplus 41 \\ D4 \oplus DF \\ 5A \oplus 6A \\ 6F \oplus 2F \end{bmatrix} = \begin{bmatrix} 01 \\ 40 \\ 27 \\ 7D \\ 6E \\ 0B \\ 30 \\ 4F \end{bmatrix}.$$

5) Fungsi *Key Expansion*

Ekspansi kunci diperoleh menggunakan blok cipher W dengan menambahkan *Round Constant* (RC) sebagai kunci putaran (Round) untuk ekspansi. *Round Constant* untuk putaran ke r $(1 \leq r \leq 10)$ adalah sebuah matriks RC[r] dimana setiap elemen nya adalah 0, kecuali baris pertama. RC[r] didefinisikan sebagai berikut:

$$rc[r]_{0,j} = S\left[8\left(r-1\right)+j\right], \qquad 0 \leq j \leq 7,\ 1 \leq r \leq 10$$

$$rc[r]_{i,j} = 0, \qquad 1 \leq i \leq 7,\ 0 \leq j \leq 7,\ 1 \leq r \leq 10$$

Setiap entri pada matrik RC[r] dipetakan dengan S-Box.

Contoh matriks Round Constant:

Untuk putaran awal $(r = 1)$, maka $RC[r]$:

$$RC[r] = \begin{bmatrix} 18 & 23 & C6 & E8 & 87 & B8 & 01 & 4F \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}.$$

Dengan *round constant*, matriks kunci berukuran 512-bit yang nantinya digunakan dalam fungsi *AddKey* akan diperluas menjadi deretan ***round keys*** $\{K_0, K_1, \ldots, K_{10}\}$.

$$K_0 = K$$

$$K_r = RF\left[RC\left[r\right]\right]\left(K_{r-1}\right).$$

Dapat dilihat bahwa Ekspansi kunci juga melibatkan fungsi *SubByte, ShiftColumns, MixRows, dan AddKey* (RF merupakan *Round Function* yang terdiri dari *SubByte, ShiftColumn, MixRows, dan AddKey*). Pada proses *AddKey*, Matriks RC[r] hanya akan mengubah baris pertama pada matrik kunci.

BAB IV

IMPLEMENTASI DAN UJI COBA

Pada bab sebelumnya, telah dibahas bagaimana struktur matrimatika dari algoritma Whirlpool dan cara kerja algoritma tersebut dalam melakukan fungsi hash. Selanjutnya, akan diberikan sebuah simulasi dari Algoritma Whirlpool dan akan dijelaskan bagaimana proses perubahan suatu data dengan panjang sembarang, menjadi sebuah pesan *digest* atau nilai hash yang memiliki panjang tetap, yaitu 512-bit. Bab ini terbagi menjadi dua pembahasan, pertama sistem pengkodean karakter ASCII dan yang kedua sistem pengkodean karakter UTF-8.

4.1. *American Standard Code for Information Interchange* (ASCII)

ASCII adalah standar pengkodean karakter untuk teknologi komunikasi. Karakter / Teks Alphabetik atau simbol-simbol yang muncul pada layar komputer, ataupun pada alat elektronik umumnya menggunakan sistem pengkodean ini. ASCII pertama kali dikomersilkan pada tahun 1963 sebagai sebuah kode teleprinter berukuran 7-bit untuk telepon warga amerika dan komunikasi telegraf **TWX** (***TeletypeWriter eXchange***). [Mary, 1999]

Saat ini, ACII telah dikembangkan menjadi ACII 8-bit, atau ASCII Extended, yang mengkodekan setiap karakter ke dalam 8-bit. Karakter ASCII terbagi menjadi 2, yaitu *Printable Character* atau karakter yang tampak dan *Control Character* yaitu fungsi untuk mengontrol karakter seperti space, del, tab, dsb. ASCII mengkodekan 95 *printable character* yang didalamnya termasuk alfabet inggris, digit, tanda baca dan simbol.

ASCII Printing Characters Chart

Decimal	Hex	Oct	Character	Decimal	Hex	Oct	Character	Decimal	Hex	Oct	Character
32	20	040	space	64	40	100	@	96	60	140	`
33	21	041	!	65	41	101	A	97	61	141	a
34	22	042	"	66	42	102	B	98	62	142	b
35	23	043	#	67	43	103	C	99	63	143	c
36	24	044	$	68	44	104	D	100	64	144	d
37	25	045	%	69	45	105	E	101	65	145	e
38	26	046	&	70	46	106	F	102	66	146	f
39	27	047	'	71	47	107	G	103	67	147	g
40	28	050	(	72	48	110	H	104	68	150	h
41	29	051	)	73	49	111	I	105	69	151	i
42	2A	052	*	74	4A	112	J	106	6A	152	j
43	2B	053	+	75	4B	113	K	107	6B	153	k
44	2C	054	,	76	4C	114	L	108	6C	154	l
45	2D	055	-	77	4D	115	M	109	6D	155	m
46	2E	056	.	78	4E	116	N	110	6E	156	n
47	2F	057	/	79	4F	117	O	111	6F	157	o
48	30	060	0	80	50	120	P	112	70	160	p
49	31	061	1	81	51	121	Q	113	71	161	q
50	32	062	2	82	52	122	R	114	72	162	r
51	33	063	3	83	53	123	S	115	73	163	s
52	34	064	4	84	54	124	T	116	74	164	t
53	35	065	5	85	55	125	U	117	75	165	u
54	36	066	6	86	56	126	V	118	76	166	v
55	37	067	7	87	57	127	W	119	77	167	w
56	38	070	8	88	58	130	X	120	78	170	x
57	39	071	9	89	59	131	Y	121	79	171	y
58	3A	072	:	90	5A	132	Z	122	7A	172	z
59	3B	073	;	91	5B	133	[	123	7B	173	{
60	3C	074	<	92	5C	134	\	124	7C	174	|
61	3D	075	=	93	5D	135	]	125	7D	175	}
62	3E	076	>	94	5E	136	^	126	7E	176	~
63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Gambar 4.1 ***ASCII printable character***

ASCII Non-Printing Control Codes Chart

Decimal	Hex	Oct	Abbr.	Description	Decimal	Hex	Oct	Abbr.	Description
0	0	000	NUL	null	16	10	020	DLE	data link escape
1	1	001	SOH	start of heading	17	11	021	DC1	device control 1
2	2	002	STX	start of text	18	12	022	DC2	device control 2
3	3	003	ETX	end of text	19	13	023	DC3	device control 3
4	4	004	EOT	end of transmission	20	14	024	DC4	device control 4
5	5	005	ENQ	inquiry	21	15	025	NAK	negative acknowledge
6	6	006	ACK	acknowledge	22	16	026	SYN	synchronous idle
7	7	007	BEL	bell	23	17	027	ETB	end of transmission block
8	8	010	BS	backspace	24	18	030	CAN	cancel
9	9	011	TAB	horizontal tab	25	19	031	EM	end of medium
10	A	012	LF	line feed/new line	26	1A	032	SUB	substitute
11	B	013	VT	vertical tab	27	1B	033	ESC	escape
12	C	014	FF	form feed/new page	28	1C	034	FS	file separator
13	D	015	CR	carriage return	29	1D	035	GS	group separator
14	E	016	SO	shift out	30	1E	036	RS	record separator
15	F	017	SI	shift in	31	1F	037	US	unit separator

Gambar 4.2 ***ASCII control character***

4.2. Proses *hashing* dengan plainteks berbasis ASCII pada algoritma Whirlpool

Setiap karakter dalam ASCII memiliki ukuran 1-Byte atau 8-bit. Hal ini menguntungkan, karena setiap entri pada matriks *CState* memiliki ruang sebesar 8-bit sehingga 64 karakter ASCII dapat dimasukan ke dalam 1 matrik *CState*.

Misalkan diberikan sebuah plainteks dengan ukuran 55-byte:

Sebuah perjalanan panjang dimulai dengan 1 langkah kaki

Selanjutnya akan dijelaskan, bagaimana proses *hashing* algoritma Whirlpool terhadap plainteks di atas.

4.2.1. Mengubah pesan input ke dalam bentuk heksadesimal

Algoritma tidak membaca input dalam bentuk *character* melainkan dalam bentuk biner, sehingga pesan masukan akan diubah terlebih dahulu ke dalam bentuk heksadesimal:

plainteks → heksadesimal:

53 65 62 75 61 68 20 70 65 72 6a 61 6c 61 6e 61 6e 20 70 61 6e 6a 61 6e 67 20 64 69 6d 75 6c 61 69 20 64 65 6e 67 61 6e 20 31 20 6c 61 6e 67 6b 61 68 20 6b 61 6b 69

Tabel 4.1 Representasi plainteks ke dalam hex

Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex
S	53	l	6C	g	67	n	6E	a	61
e	65	a	61	[*space*]	20	g	67	h	68
b	62	n	6E	d	64	a	61	[*space*]	20
u	75	a	61	i	69	n	6E	k	6B
a	61	n	6E	m	6D	[*space*]	20	a	61
h	68	[*space*]	20	u	75	l	31	k	6B
[*space*]	20	p	70	l	6C	[*space*]	20	i	69
p	70	a	61	a	61	l	6C		
e	65	n	6E	i	69	a	61		
r	72	j	6A	[*space*]	20	n	6E		
j	6A	a	61	D	64	g	67		
a	61	n	6E	e	65	k	6B		

Perubahan karakter - karakter di atas dapat dilihat melalui tabel 4.1

4.2.2. Melakukan *Padding Bits* terhadap pesan input

Padding bits berfungsi agar panjang pesan input berkelipatan ganjil dari 512-bits. Blok pesan akan ditambahkan dengan bit 1, diikuti dengan bit-0 sesuai dengan kebutuhannya.

Contoh: Suatu pesan input dengan panjang (256 x 3 = 768-bit) akan di *padding* sehingga panjangnya menjadi (256 x 5 = 1280-bit).

Sebelumnya telah diberikan sebuah pesan masukan berukuran 55-byte, maka 55-byte = (55 x 8) = 440-bits. Algoritma akan melakukan *padding bits* sebanyak, 256 x 3 = 768-bit − 440-bit = 328-bit yang terdiri dari bit-1, diikuti dengan bit-0.

heksadesimal → digit biner + ***padding bits***:

01010011 01100101 01100010 01110101 01100001 01101000 00100000 01110000
01100101 01110010 01101010 01100001 01101100 01100001 01101110 01100001
01101110 00100000 01110000 01100001 01101110 01101010 01100001 01101110
01100111 00100000 01100100 01101001 01101101 01110101 01101100 01100001
01101001 00100000 01100100 01100101 01101110 01100111 01100001 01101110
00100000 00110001 00100000 01101100 01100001 01101110 01100111 01101011
01100001 01101000 00100000 01101011 01100001 01101011 01101001 **10000000**
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Bit yang diberi warna merah merupakan bit tambahan (*padding bit*).

Tahap selanjutnya adalah menambahkan *unsigned 256-bit integer* (ditandai dengan warna biru). Hal ini bertujuan agar pesan input memiliki panjang bit berkelipatan 512. bit-bit tersebut adalah bagian pesan input sebelum dilakukan *Padding bits*

Digit biner + padding bits + *unsigned block bit* :

01010011 01100101 01100010 01110101 01100001 01101000 00100000 01110000
01100101 01110010 01101010 01100001 01101100 01100001 01101110 01100001
01101110 00100000 01110000 01100001 01101110 01101010 01100001 **10000000**
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
01101110 01100111 00100000 01100100 01101001 01101101 01110101 01101100
01100001 01101001 00100000 01100100 01100101 01101110 01100111 01100001
01101110 00100000 00110001 00100000 01101100 01100001 01101110 01100111
01101011 01100001 01101000 00100000 01101011 01100001 01101011 01101001

4.2.3. Membuat blok - blok pesan

Deret bit di atas akan di bagi - bagi menjadi blok - blok pesan berukuran 64-byte (256-bits) yang direpresentasikan sebagai sebuah matriks berukuran 8x8.

m_1 =

01010011 01100101 01100010 01110101 01100001 01101000 00100000
01110000 01100101 01110010 01101010 01100001 01101100 01100001
01101110 01100001 01101110 00100000 01110000 01100001 01101110
01101010 01100001 10000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000

m_2 =

00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 01101110 01100111 00100000
01100100 01101001 01101101 01110101 01101100 01100001 01101001
00100000 01100100 01100101 01101110 01100111 01100001 01101110
00100000 00110001 00100000 01101100 01100001 01101110 01100111
01101011 01100001 01101000 00100000 01101011 01100001 01101011
01101001

4.2.4. Representasi blok pesan ke dalam bentuk matriks

$m_1 \rightarrow Hex$ =

53 65 62 75 61 68 20 70 65 72 6A 61 6C 61 6E 61 6E 20 70 61 6E 6A 61 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

$$m_1 = \begin{bmatrix} 53 & 65 & 62 & 75 & 61 & 68 & 20 & 70 \\ 65 & 72 & 6A & 61 & 6C & 61 & 6E & 61 \\ 6E & 61 & 6E & 20 & 70 & 61 & 6E & 6A \\ 61 & 80 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

$m_2 \rightarrow Hex$ =

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6E 67 20 64 69 6D 75 6C 61 69 20 64 65 6E 67 61 6E 20 31 20 6C 61 6E 67 6B 61 68 20 6B 61 6B 69

$$m_2 = \begin{bmatrix} 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 6E & 67 & 20 & 64 & 69 & 6D & 75 & 6C \\ 61 & 69 & 20 & 64 & 65 & 6E & 67 & 61 \\ 6E & 20 & 31 & 20 & 6C & 61 & 6E & 67 \\ 6B & 61 & 68 & 20 & 6B & 61 & 6B & 69 \end{bmatrix}$$

4.2.5. Pembentukan matriks Hash awal (K_0)

Kunci awal akan dibentuk sebagai matriks hash awal, dan *RoundKey* awal (K_0). Semua elemen pada matriks ini adalah bit-0.

4.2.6. Pengolahan blok - blok pesan (m_i)

Setelah matriks - matriks blok pesan terbentuk, algoritma akan mengolah blok - blok pesan tersebut secara paralel dengan 4 fungsi. Setiap blok pesan akan diolah sebanyak 10 ronde/putaran (*round*) untuk menghasilkan sebuah Blok Cipher W.

$$W(K) = \left(\bigcirc_{r=1}^{10} RF(K_r) \right) \circ AK(K_0)$$

dengan $RF\,(K_r) = AK\,[K_r] \circ MR \circ SC \circ SB.$

Blok - blok cipher W akan diolah menggunakan fungsi kompresi miyaguchi-preneel.

1) Fungsi Penambahan Kunci Putaran (AddRoundKey)

Langkah awal yang dilakukan algoritma adalah menambahkan kunci awal (*initiation key*) pada matrik *CState*. Dalam hal ini, kunci awal adalah matriks nol (berelemen nol), maka

$$\text{AK}\,[\text{K}_0] + CState = 0 + CState = CState.$$

Setelah itu, Algoritma akan mengolah K_0 secara paralel dengan fungsi *SubByte, ShiftColumns, MixRows, dan AddRoundConstant* untuk menghasilkan $\{\text{K}_1, \text{K}_2, \ldots, \text{K}_{10}\}$.

2) Subtitutes Byte (*SubByte*)

Pada proses ini, elemen - elemen pada matrik *CState* akan disubstitusi menggunakan S-Box Whirlpool dengan korespondensi

$$B = SubByte\,(A) \Leftrightarrow\ b_{i,j} = S\,[a_{i,j}]\,, \quad 0 \leq i, \quad j \leq 7.$$

$m_1 \rightarrow SubByte(m_1)$ =

$$\begin{bmatrix} 53 & 65 & 62 & 75 & 61 & 68 & 20 & 70 \\ 65 & 72 & 6A & 61 & 6C & 61 & 6E & 61 \\ 6E & 61 & 6E & 20 & 70 & 61 & 6E & 6A \\ 61 & 80 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix} \rightarrow \begin{bmatrix} 71 & CD & D5 & F1 & 01 & FF & 15 & B4 \\ CD & 93 & 90 & 0F & 20 & 0F & 1A & 0F \\ 1A & 0F & 1A & 15 & B4 & 0F & 1A & 90 \\ 0F & 97 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{bmatrix}$$

$m_2 \rightarrow SubByte(m_2)$ =

$$\begin{bmatrix} 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 6E & 67 & 20 & 64 & 69 & 6D & 75 & 6C \\ 61 & 69 & 20 & 64 & 65 & 6E & 67 & 61 \\ 6E & 20 & 31 & 20 & 6C & 61 & 6E & 67 \\ 6B & 61 & 68 & 20 & 6B & 61 & 6B & 69 \end{bmatrix} \begin{bmatrix} 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 1A & 48 & 15 & BE & 7A & 68 & F1 & 20 \\ 0F & 7A & 15 & BE & CD & 1A & 48 & 0F \\ 1A & 15 & 5D & 15 & 20 & 0F & 1A & 48 \\ 5F & 0F & FF & 15 & 5F & 0F & 5F & 7A \end{bmatrix}$$

3) Fungsi Permutasi (*ShiftColumns*)

Matrik *CState* akan diproses secara permutasi, dimana setiap byte pada kolom yang lebih besar dari nol, akan digeser sebanyak 1-byte, 2-byte,..., 7-byte memutar secara baris.

$$B = ShiftColumn\,(A) \Leftrightarrow\ b_{i,j} = a_{(i-j) mod\ 8\ ,\ \ j},\quad 0 \leq i,\quad\ j \leq 7.$$

$SubByte(m_1) \rightarrow ShiftColumn(SubByte(m_1))$ =

$$\begin{bmatrix} \mathbf{71} & \mathbf{CD} & \mathbf{D5} & \mathbf{F1} & \mathbf{01} & \mathbf{FF} & \mathbf{15} & \mathbf{B4} \\ CD & 93 & 90 & 0F & 20 & 0F & 1A & 0F \\ 1A & 0F & 1A & 15 & B4 & 0F & 1A & 90 \\ 0F & 97 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{bmatrix} \rightarrow \begin{bmatrix} 71 & 18 & 18 & 18 & 18 & 18 & 15 & 0F \\ CD & \mathbf{CD} & 18 & 18 & 18 & 18 & 1A & 90 \\ 1A & 93 & \mathbf{D5} & 18 & 18 & 18 & 18 & 18 \\ 0F & 0F & 90 & \mathbf{F1} & 18 & 18 & 18 & 18 \\ 18 & 97 & 1A & 0F & \mathbf{01} & 18 & 18 & 18 \\ 18 & 18 & 18 & 15 & 20 & \mathbf{FF} & 18 & 18 \\ 18 & 18 & 18 & 18 & B4 & 0F & \mathbf{15} & 18 \\ 18 & 18 & 18 & 18 & 18 & 0F & 1A & \mathbf{B4} \end{bmatrix}$$

$SubByte(m_2) \rightarrow ShiftColumn(SubByte(m_2))$ =

$$\begin{bmatrix} \mathbf{18} & \mathbf{18} & \mathbf{18} & \mathbf{18} & \mathbf{18} & \mathbf{18} & \mathbf{18} & \mathbf{18} \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 1A & 48 & 15 & BE & 7A & 68 & F1 & 20 \\ 0F & 7A & 15 & BE & CD & 1A & 48 & 0F \\ 1A & 15 & 5D & 15 & 20 & 0F & 1A & 48 \\ 5F & 0F & FF & 15 & 5F & 0F & 5F & 7A \end{bmatrix} \rightarrow \begin{bmatrix} 18 & 0F & FF & BE & 7A & 18 & 18 & 18 \\ 18 & \mathbf{18} & 5D & 15 & CD & 68 & 18 & 18 \\ 18 & 18 & \mathbf{18} & 15 & 20 & 1A & F1 & 18 \\ 18 & 18 & 18 & \mathbf{18} & 5F & 0F & 48 & 20 \\ 1A & 18 & 18 & 18 & \mathbf{18} & 0F & 1A & 0F \\ 0F & 48 & 18 & 18 & 18 & \mathbf{18} & 5F & 48 \\ 1A & 7A & 15 & 18 & 18 & 18 & \mathbf{18} & 0F \\ 5F & 15 & 15 & BE & 18 & 18 & 18 & \mathbf{18} \end{bmatrix}$$

4) Fungsi Difusi (*MixRows*)

Proses selanjutnya adalah difusi matriks. *CState* $(ShiftColumn(m_i))$ akan dikalikan dengan sebuah matriks koefisien. Hasil perhitungan pada proses ini diperoleh dengan bantuan program MatLab.

$ShiftColumn(m_1) \rightarrow MixRows(ShiftColumn(m_1))$ =

$$\begin{bmatrix} 71 & 18 & 18 & 18 & 18 & 18 & 15 & 0F \\ CD & CD & 18 & 18 & 18 & 18 & 1A & 90 \\ 1A & 93 & D5 & 18 & 18 & 18 & 18 & 18 \\ 0F & 0F & 90 & F1 & 18 & 18 & 18 & 18 \\ 18 & 97 & 1A & 0F & 01 & 18 & 18 & 18 \\ 18 & 18 & 18 & 15 & 20 & FF & 18 & 18 \\ 18 & 18 & 18 & 18 & B4 & 0F & 15 & 18 \\ 18 & 18 & 18 & 18 & 18 & 0F & 1A & B4 \end{bmatrix} \bullet \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}$$

$$MixRows(m_1) = \begin{bmatrix} 6E & 67 & D4 & 62 & CB & CD & 34 & 13 \\ C9 & DA & CB & 0D & 90 & 65 & 7B & 4B \\ 26 & B3 & C6 & CC & BF & 08 & 59 & 0C \\ BC & B0 & FB & 39 & 1F & 6F & 8F & 84 \\ 99 & 2E & C0 & AE & 24 & D4 & 2E & 99 \\ A7 & CC & 4B & 70 & 1D & EC & 74 & 80 \\ 84 & CD & 25 & B6 & D8 & 09 & 96 & 73 \\ 22 & 6C & 76 & 9E & 23 & 3C & A6 & 30 \end{bmatrix}$$

$ShiftColumn(m_2) \rightarrow MixRows(ShiftColumn(m_2))$ =

$$\begin{bmatrix} 18 & 0F & FF & BE & 7A & 18 & 18 & 18 \\ 18 & 18 & 5D & 15 & CD & 68 & 18 & 18 \\ 18 & 18 & 18 & 15 & 20 & 1A & F1 & 18 \\ 18 & 18 & 18 & 18 & 5F & 0F & 48 & 20 \\ 1A & 18 & 18 & 18 & 18 & 0F & 1A & 0F \\ 0F & 48 & 18 & 18 & 18 & 18 & 5F & 48 \\ 1A & 7A & 15 & 18 & 18 & 18 & 18 & 0F \\ 5F & 15 & 15 & BE & 18 & 18 & 18 & 18 \end{bmatrix} \bullet \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}$$

$$MixRows(m_2) = \begin{bmatrix} 25 & 15 & E3 & 60 & 40 & B7 & 3B & 61 \\ 0D & 64 & 12 & B3 & 2E & FC & 34 & 04 \\ 4D & 13 & 2C & AE & C0 & D4 & 2E & 99 \\ A7 & CD & 4B & 70 & EC & 1D & 74 & 80 \\ 22 & CC & 6C & B6 & D8 & 09 & 96 & 73 \\ 84 & 75 & 76 & 9E & 23 & 3C & A6 & 30 \\ 72 & 71 & 58 & 0A & 25 & 36 & 1C & C7 \\ 14 & 56 & D6 & F0 & B5 & A9 & 51 & 00 \end{bmatrix}$$

5) Fungsi Penambahan Kunci Putaran (*AddRoundKey*)

Setelah melalui proses difusi (*mixrows*) menandakan berakhirnya suatu *round* dan akan berlanjut ke *round* selanjutnya. Algoritma kembali menambahkan kunci seperti pada awal permulaan *round* namun dengan kunci yang berbeda dari sebelumnya, yaitu, kunci yang sudah melalui fungsi *SubByte, ShiftColumn, MixRows, dan AddRoundConstant*.

Fungsi kunci putaran (*Round key*) adalah sebagai berikut:

$$K_0 = K.$$

$$K_r = RF\left[RC\left[r\right]\right]\left(K_{r-1}\right).$$

Pada permulaan proses, diberikan sebuah kunci (K_0) yaitu suatu matriks berukuran 8x8 dengan semua elemennya adalah nol, akan dijelaskan proses perputaran kunci K_0 menjadi K_1

$(k_0) \rightarrow SubByte(k_0) =$

$$\begin{bmatrix} 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix} \rightarrow \begin{bmatrix} 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{bmatrix}.$$

$SubByte(k_0)) \rightarrow ShiftColumn(SubByte(k_0)) =$

$$\begin{bmatrix} 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{bmatrix} \rightarrow \begin{bmatrix} 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{bmatrix}$$

$ShiftColumn(k_0)) \rightarrow MixRows(ShiftColumn(k_0))$ =

$$\begin{bmatrix} 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \\ 18 & 18 & 18 & 18 & 18 & 18 & 18 & 18 \end{bmatrix} \bullet \begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}$$

$$MixRows(K_0) = \begin{bmatrix} 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{bmatrix}$$

6) Fungsi Perluasan Kunci (*AddRoundConstant*)

Fungsi ini hanya bekerja pada pengolahan *Round key*. Matriks *Round constant* akan dijumlahkan dengan matrik *Round key*. Entri-entri pada matriks *round constant* didefinisi sebagai berikut:

$$rc[r]_{0,j} = S\left[8\left(r-1\right)+j\right], \qquad 0 \leq j \leq 7,\ 1 \leq r \leq 10$$

$$rc[r]_{i,j} = 0, \qquad 1 \leq i \leq 7,\ 0 \leq j \leq 7,\ 1 \leq r \leq 10$$

Setiap elemen di baris pertama, merupakan hasil pemetaan terhadap S-Box, sehingga pada proses ini,

$$RC[1] = \begin{bmatrix} 18 & 23 & C6 & E8 & 87 & B8 & 01 & 4F \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

Dengan mengoperasikan K_0 dan RC[1], diperoleh:

$$\begin{bmatrix} 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{bmatrix} \oplus \begin{bmatrix} 18 & 23 & C6 & E8 & 87 & B8 & 01 & 4F \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

$$k_1 = \begin{bmatrix} 30 & 0B & EE & C0 & AF & 90 & 29 & 67 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{bmatrix}$$

Kembali ke point-5, maka proses *AddRoundKey* pada putaran ke-2 adalah:

$MixRows(m_1) \oplus k_1 \; =$

$$\begin{bmatrix} 6E & 67 & D4 & 62 & CB & CD & 34 & 13 \\ C9 & DA & CB & 0D & 90 & 65 & 7B & 4B \\ 26 & B3 & C6 & CC & BF & 08 & 59 & 0C \\ BC & B0 & FB & 39 & 1F & 6F & 8F & 84 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{bmatrix} \oplus \begin{bmatrix} 30 & 0B & EE & C0 & AF & 90 & 29 & 67 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{bmatrix}$$

$$m_1 = \begin{bmatrix} 5E & 6C & 3A & A2 & 64 & 5D & 1D & 75 \\ E1 & F2 & E3 & 25 & B8 & 4D & 53 & 63 \\ 0E & 9B & EE & E4 & 97 & 20 & 71 & 24 \\ 94 & 98 & D3 & 11 & 37 & 47 & A7 & AC \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

$$MixRows(m_2) \oplus k_1 =$$

$$\begin{bmatrix} 25 & 15 & E3 & 60 & 40 & B7 & 3B & 61 \\ 0D & 64 & 12 & B3 & 2E & FC & 34 & 04 \\ 4D & 13 & 2C & AE & C0 & D4 & 2E & 99 \\ A7 & CD & 4B & 70 & EC & 1D & 74 & 80 \\ 22 & CC & 6C & B6 & D8 & 09 & 96 & 73 \\ 84 & 75 & 76 & 9E & 23 & 3C & A6 & 30 \\ 72 & 71 & 58 & 0A & 25 & 36 & 1C & C7 \\ 14 & 56 & D6 & F0 & B5 & A9 & 51 & 00 \end{bmatrix} \oplus \begin{bmatrix} 30 & 0B & EE & C0 & AF & 90 & 29 & 67 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \\ 28 & 28 & 28 & 28 & 28 & 28 & 28 & 28 \end{bmatrix}$$

$$m_2 = \begin{bmatrix} 15 & 1E & 0D & A0 & EF & 27 & 12 & 06 \\ 25 & 4C & 3A & 9B & 06 & D4 & 1C & 2C \\ 65 & 3B & 04 & 87 & E8 & FC & 06 & B1 \\ 8F & E5 & 63 & 58 & C4 & 35 & 5C & A8 \\ 0A & E4 & 44 & 96 & F0 & 21 & BE & 5B \\ AC & 5D & 5E & B6 & 0B & 14 & 8E & 18 \\ 5A & 59 & 70 & 22 & 0D & 1E & 34 & EF \\ 3C & 7E & FE & D8 & 9D & 91 & 79 & 28 \end{bmatrix}$$

Setelah ini, matrik m_1, m_2 akan kembali diproses dengan *SubByte, ShiftColumn dan MixRow* lalu kembali lagi ke fungsi *AddRoundKey* hingga 10 putaran.

4.3. *Universal Coded Character Set* (Unicode)

Unicode adalah sebuah industri komputer yang bergerak dibidang pengkodean teks teks dari seluruh dunia, baik yang modern maupun yang kuno, agar dapat direpresentasikan dalam sistem penulisan digital. Berbeda dengan ASCII, Unicode tidak hanya memuat alfabet Inggris dan numerik saja, namun memuat hampir

seluruh aksara/huruf-huruf dari negara - negara di dunia yang tidak menggunakan alfabet Inggris, seperti China, Jepang, Korea, Arab, bahkan Unicode mengkodekan aksara Yuniani kuno dan bahasa - bahasa daerah seperti aksara Jawa kuno dan Sanskrit. Selain itu, Unicode juga mengkodekan simbol-simbol, dan emotikon.

Ada berbagai macam sistem pengkodean yang berbasis pada Unicode, misalnya UTF-8, UTF-16, UTF-32, UCS-2, dan sebagainya.

Dalam penelitian ini, akan dibahas bagaimana proses hashing algoritma Whirlpool dengan plainteks berbasis Unicode. Sistem yang dipilih adalah UTF-8 yang menggunakan Standar Unicode sebagai basisnya.

UTF-8 atau (*Universal Coded Character Set Transformation Format* -8bit) adalah sistem pengkodean karakter yang dapat mengkodekan poin - poin kode dalam Unicode dengan 1 sampai 4 Byte. Sistem ini didesain oleh Ken Thompson dan Rob Pike.

UTF-8 dapat mengkodekan karakter tertentu dengan panjang yang tidak tetap, misalnya karakter-karakter alfabet Inggris dan numerik yang ada dalam ASCII menggunakan 8-bit, maka UTF-8 akan merepresentasikan karakter-karakter tersebut ke dalam 8-bit juga. Berbeda apabila karakter tersebut diluar karakter ASCII, seperti aksara Arab, Rusia, Yunani, Armenia. Maka UTF-8 akan mengkodekan karakter-karakter tersebut ke dalam 16-bit atau 2 byte. Untuk aksara Jepang, China dan Korea, UTF-8 akan mengkodekan karakter-karakter tersebut ke dalam 24-bit atau 3-byte, dst.

4.3.1. Proses *hashing* dengan plainteks berbasis Unicode pada algoritma Whirlpool

Pada pembahasan sebelumnya, telah dijelaskan bagaimana proses *hashing* Algoritma Whirlpool dengan plainteks berbasis ASCII. Selanjutnya akan dijelaskan bagaimana proses *hashing* pada plainteks yang bukan berbasis ASCII, seperti aksara China, Jepang, Korea, dan Arab yang semuanya tidak menggunakan Alfabet Inggris. Bahasa yang dipilih adalah *Arabic*, dan plainteksnya merupakan ayat - ayat yang diambil dari Al - Quran Digital.

Sebagai contoh, diambil ayat pertama surah Al - Ikhlas sebagai plainteks untuk dicari nilai hash nya :

قُلْ هُوَ اللَّهُ أَحَدٌ

Sistem pengkodean karakter ISO/IEC 8859-6 yang berbasis Standar ASCII, dapat mengkodekan karakter huruf arabic ke dalam 1-Byte yang dikenal dengan *Hijaiyah* (tanpa harakat). Namun ada kekurangan dari sistem ini, dimana pada sistem ini tidak dapat merepresentasikan huruf arab sambung, contoh:

هو → ه و

hal ini disebut dengan *Arabic Representation Form*. UTF-8 memiliki kumpulan karakter *arabic*, yang terdiri dari *Arabic Set, Arabic Supplement Set, Arabic Extended-A Set, Arabic Representation Form-A Set,* dan *Arabic Representation Form-B* .

Gambar 4.3 ***Arabic Character Set***

Tabel di atas adalah contoh himpunan karakter *arabic* (*Arabic Set*) dari sistem pengkodean UTF-8. UTF-8 mengkodekan setiap huruf arabic kedalam 16-bit atau 2-byte.

4.3.2. Mengubah pesan input ke dalam bentuk heksadesimal

Setiap karakter arabic akan menempati 2 entri matrik CState karena memiliki ukuran 16-bit.

قُلْ هُوَ اللَّهُ أَحَدٌ

Plainteks berikut jika diuraikan menjadi byte per byte adalah:

Char	Hex	Char	Hex	Char	Hex	Char	Hex
ق	D982	ُ	D98F	ل	D984	ا	D8A7
ُ	D98F	و	D988	ّ	D991	َ	D98E
ل	D984	َ	D98E	َ	D98E	ح	D8AD
ْ	D992	[space]	0020	ه	D987	َ	D98E
[space]	0020	ا	D8A7	ُ	D98F	د	D8AF
ه	D987	ل	D984	[space]	0020	ٌ	D98C

Dapat dilihat, bahwa harakat terpisah dengan huruf hijaiyah, sehingga sebuah huruf hijaiyah dengan harakat akan dihitung sebagai 2 karakter yang berbeda. Karena itu, huruf hijaiyah dengan harakat akan memiliki ukuran sebesar 32-bit atau 4-Byte.

plainteks→ heksadesimal:

D982 D98F D984 D992 0020 D987 D98F D988 D98E 0020 D8A7 D8A7 D984 D991 D98E D987 D98F 0020 D8A7 D98E D8AD D98E D8AF D98C

Panjang plainteks dari ayat pertama surah Al-Ikhlas adalah:

$$(16\text{-bit x } 24 = 384\text{-bit } atau\ 48\text{-byte})$$

Selanjutnya, plainteks akan melalui proses *Padding Bits* dan proses - proses selanjutnya akan sama seperti proses hashing pada Plainteks berbasis ASCII yang telah dibahas sebelumnya.

4.4. Proses *hashing* menggunakan C++ dan HashTab

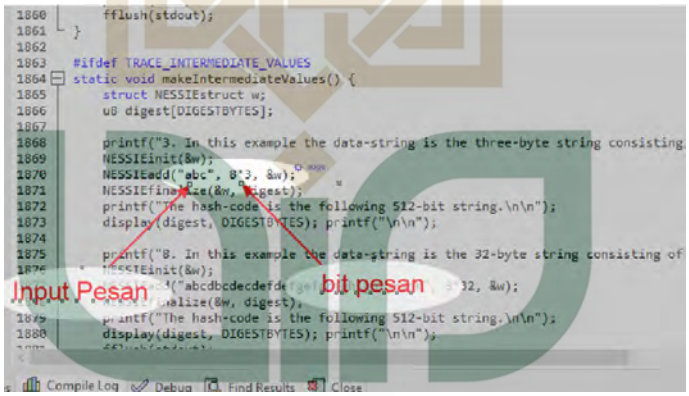
Nilai Hash atau Pesan *digest* dihasilkan dari operasi XOR antara Nilai Hash sebelumnya, Pesan m_i, dan blok cipher m_i yaitu :

$$H_0 = Nilai\ Hash\ Awal\ (\textit{Initiation Hash})$$

$$H_i = E\left(H_{i-1},\ m_i\right) \oplus H_{i-1} \oplus m_i = Nilai\ Pertengahan$$

$$H_t = Nilai\ Hash\ Final$$

Gambar di bawah ini merupakan *psuedocode* Whirlpool pada bahasa pemograman C++ yang dirancang oleh Vincent Rijmen.



Gambar 4.4 Simulasi *hashing* menggunakan C++

Keterangan:

Input Pesan : Merupakan plainteks yang akan diolah oleh algoritma. Batas maksimum dari panjang pesan adalah 2^{256}-bit.

Bit Pesan : Merupakan panjang bit dari pesan input.

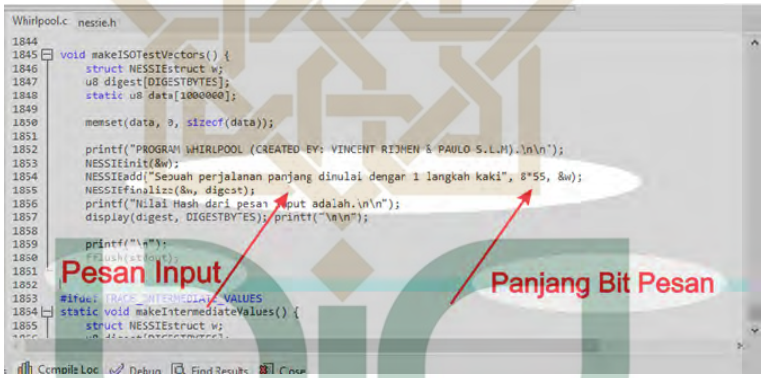
4.4.1. Proses *hashing* pada plainteks berbasis ASCII menggunakan Bahasa Pemograman C++ dan program HashTab

Berikut adalah contoh mencari nilai hash dari suatu pesan berbasis ASCII dengan menggunakan bahasa pemograman C++ :

INPUT :

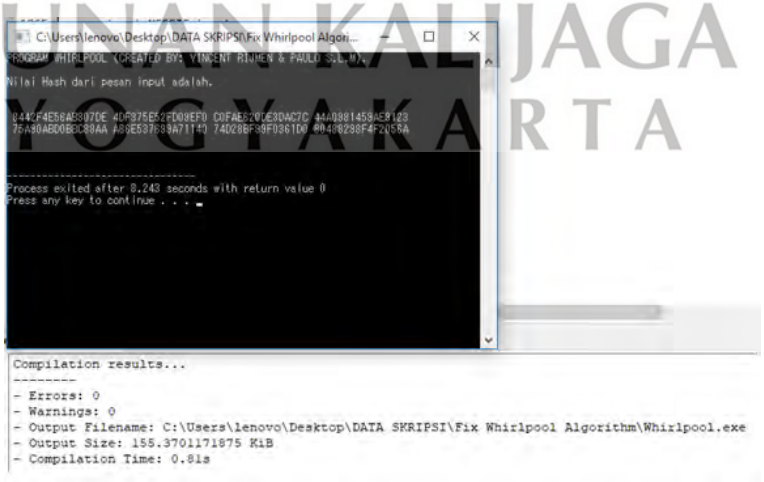
Plainteks : Sebuah perjalanan panjang dimulai dengan 1 langkah kaki

Panjang Bit : 55-bit $\rightarrow$ (8*55)



Gambar 4.5 Proses input plainteks pada Algoritma Whirlpool menggunakan C++

OUTPUT :



Gambar 4.6 Output nilai hash dari Plainteks

Nilai Hash :

84 42 F4 E5 6A B3 07 DE 4D F3 75 E5 2F D0 9E F0 C0 FA E6 20 DE 3D AC 7C 44 A0 98 14 59 AE 91 23 75 A9 0A BD 0B BC 88 AA A8 6E 53 76 99 A7 11 40 74 D2 8B F9 9F 03 61 D0 80 48 82 98 F4 F2 05 6A

Panjang Bit : 512-bit / (64-Byte)

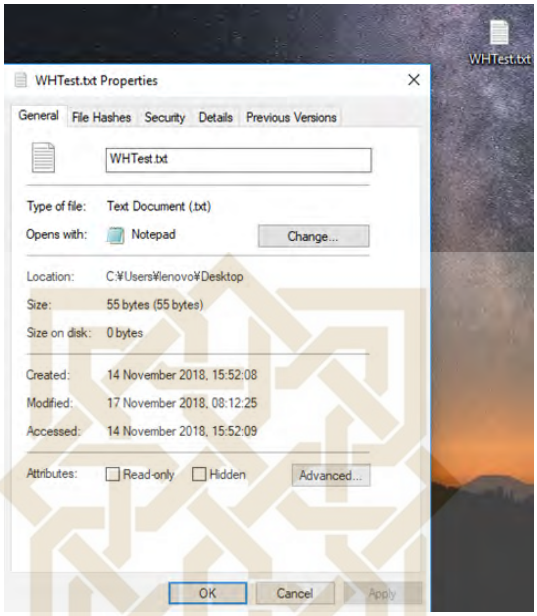
Selanjutnya adalah contoh mencari nilai hash dari suatu pesan berbasis ASCII dengan menggunakan program HashTab v6.0.0:

Langkah-langkah:

1. Tulis sebuah plainteks pada aplikasi **Notepad**. Beri nama file, lalu Save.

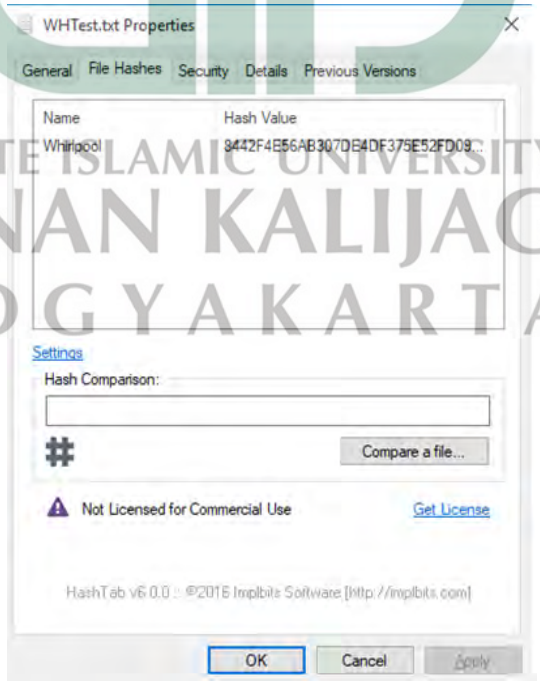
Gambar 4.7 File simulasi menggunakan Notepad

2. Klik kanan pada file yang telah di save sebelumnya, lalu pilih opsi **properties**.



Gambar 4.8 File Properties

3. Kemudian pilih tab **File Hashes**. Pada tab ini, nilai hash dari plainteks akan terlihat pada kolom **Hash Value**.



Gambar 4.9 Tampilan HashTab dari file

4.5. *Verification Data Integrity*

Verifikasi integritas data adalah salah satu kegunaan dari fungsi hash. Verifikasi integritas data dilakukan dengan membandingkan nilai hash atau pesan digest dari dua buah data. Apabila kedua data memiliki nilai hash yang sama, maka kedua data tersebut dapat dipastikan tidak mengalami perubahan data. Namun jika nilai hash kedua data berbeda, maka salah satu dari data mengalami perubahan.

Ketika sebuah data mengalami perubahan, algoritma hash yang baik akan mendeteksi perubahan sekecil apapun. Hal ini disebut dengan *Avalanche Effect*.

4.5.1. *Avalache Effect*

Avalanche Effect yang berarti Efek Longsor, adalah kajian kriptografi untuk melihat kinerja suatu algoritma, baik algoritma blok cipher, maupun algoritma hash, dalam mendeteksi perubahan input data sekecil apapun.

Dalam konteks algoritma hash, ketika suatu *input* data mengalami sedikit perubahan, maka nilai hash nya akan mengalami perubahan yang signifikan. Hal ini mengartikan bahwa algoritma hash yang baik, akan memiliki fungsi yang injektif, sehingga sulit untuk menemukan pesan atau input data yang berbeda, namun memiliki nilai hash yang sama.

Contoh *Avalanche Effect*:

Diberikan sebuah input pesan :

Plainteks : Winner-winner-chicken-dinner

Panjang Bit : 28-Byte / 224-bit

Nilai Hash : 611E19263989D2E74F76E7BFCD123CF721CDDE962637EC71F69B4714D1FED603E31289C0EBA51BD4628D6D13D30ACE03EDE56F75D70EEED0255B8AC8191CECAF

Ketika *input* pesan mengalami perubahan 2-Byte,

Plainteks : Winner-winner-chicken-di**mm**er

Panjang Bit : 28-Byte / 224-bit

Nilai Hash : D59CF90B7A0972FAB2246E4EC5506ED80989834D90237D21BDD8D3F18DCA17CC23CFFD63171D5C46E7DD999E36441CD5D42DC560919E0F8F16EF4C97B32C2816

Hampir seluruh nilai hash nya berubah, kemudian apabila pesan mengalami perubahan 1-Byte,

Plainteks : Winner-winner-chicken-din**m**er

Panjang Bit : 28-Byte / 224-bit

Nilai Hash : AB1D57CEC6B5F69D12E2F47C591F1C1ECB7EC16BFEC4164CDDA6ACBC5D1EC06D812197C84FA641EA69C15B0AEB9B130D71E0139181C19F70FB2F8810F4DE3F94

Nilai hashnya pun kembali berubah. Inilah yang disebut dengan *Avalanche Effect*.

Avalanche effect juga dipengaruhi oleh perubahan posisi,:

Plainteks : Winner-chicken-winner-dinner

Panjang Bit : 28-Byte / 224-bit

Nilai Hash : 0A5A4C95FE3A7FC28CA3C3F982B9B8E4BFAA5C00659481871D68A7E A75136A3AE49BC136C1EB9C2F897728454722C1092EE1E7FE7AC76FA8823D11A8A901 504C

Meskipun tidak ada perubahan karakter, namun ketika suatu karakter berpindah posisi, hal itu mengakibatkan nilai hashnya berubah.

Avalanche effect, dalam kriptografi dimanfaatkan dalam mengautentikasi dan memverifikasi suatu pesan/data dikenal dengan Autentikasi Pesan (*Message Authentication*).

4.5.2. Simulasi verifikasi integritas data pada Al-Qur'an Digital

Verifikasi integritas data Al - Quran digital, bertujuan untuk mendeteksi adanya perubahan lafadz atau harakat suatu ayat pada suatu *database* Al - Quran. Hal tersebut sangatlah penting karena Al - Quran menggunakan bahasa arab, yang memiliki tingkat sensitifitas tinggi terhadap perubahan karakter sehingga dapat mengubah makna dari suatu kalimat.

Berikut ini adalah contoh kesalahan dalam penulisan ayat Al - Quran, yaitu merubah suatu harakat dengan harakat lain (*Lahn Jali*).

صِرَاطَ الَّزِيْنَ اَنْعَمْتَ عَلَيْهِم ...

Artinya : *"(yaitu) jalan orang-orang yang telah **Engkau** anugerahkan nikmat kepada mereka"* (QS: Al-Fatihah: 7).

Perhatikan lafadz yang diberi warna merah. Bila terjadi perubahan harakat pada lafadz tersebut seperti,

أَنْعَمْتُ → أَنْعَمْتَ

Maka dlamir-nya berubah menjadi (aku) sehingga artinya berubah menjadi :

*"(yaitu) jalan orang-orang yang telah **aku** anugerahkan nikmat kepada mereka."*

Contoh lain kesalahan penulisan ayat al-quran yang dapat merubah makna adalah sebagai berikut:

... وَلَعَلَّكُمْ تَشْكُرُوْنَ

Artinya : *" dan mudah-mudahan kamu **bersyukur**."* (QS: Al-Jatsiyah: 12)

Bila terjadi perubahan huruf pada lafadz yang diberi tanda merah, seperti:

تَسْكُرُوْنَ → تَشْكُرُوْنَ

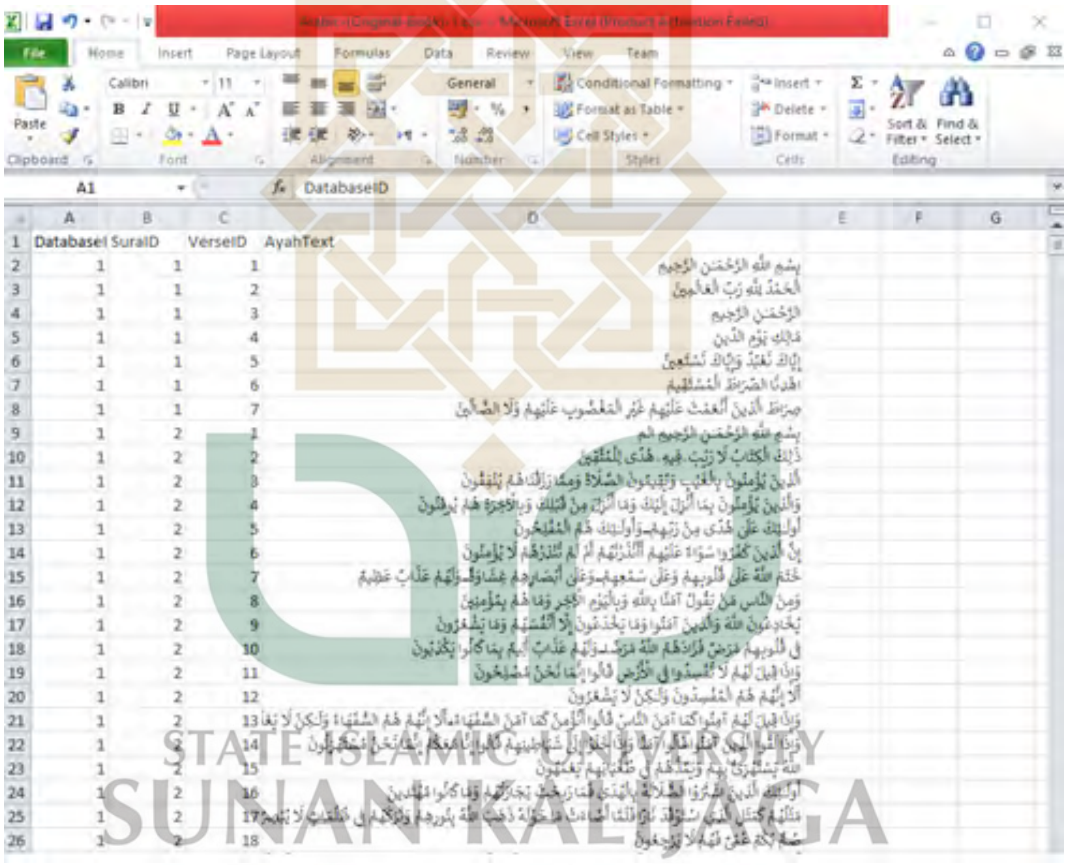
Maka maknanya dapat berubah menjadi:

*" dan mudah-mudahan kamu **mabuk**."*

Sebuah pesan input yang berbeda akan menghasilkan nilai hash yang berbeda. Dalam konteks data komputer, maka perubahan data dari suatu kumpulan data (*database*) akan mengakibatkan perubahan nilai hash dari *database* tersebut.

Berikut ini adalah langkah langkah verifikasi integritas data Al - Quran digital dengan menggunakan program HashTab v6.0.0:

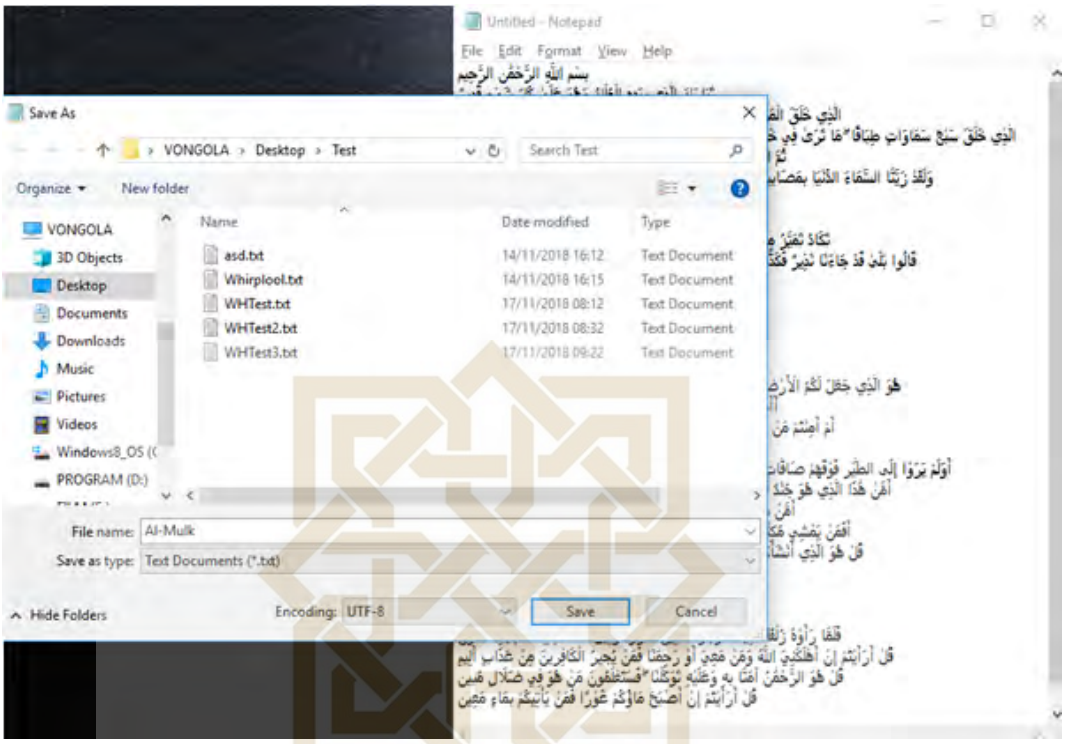
1. Gambar di bawah ini merupakan *database* Al - Quran Digital dalam format .csv (*comma separated values*) yang terdiri dari 6237 baris dan 4 kolom. Data ini diambil dari website **qurandatabase.org**.



	A	B	C	D
1	DatabaseID	SuraID	VerseID	AyahText
2	1	1	1	بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ
3	1	1	2	الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ
4	1	1	3	الرَّحْمَٰنِ الرَّحِيمِ
5	1	1	4	مَالِكِ يَوْمِ الدِّينِ
6	1	1	5	إِيَّاكَ نَعْبُدُ وَإِيَّاكَ نَسْتَعِينُ
7	1	1	6	اهْدِنَا الصِّرَاطَ الْمُسْتَقِيمَ
8	1	1	7	صِرَاطَ الَّذِينَ أَنْعَمْتَ عَلَيْهِمْ غَيْرِ الْمَغْضُوبِ عَلَيْهِمْ وَلَا الضَّالِّينَ
9	1	2	1	بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ الم
10	1	2	2	ذَٰلِكَ الْكِتَابُ لَا رَيْبَ فِيهِ هُدًى لِلْمُتَّقِينَ
11	1	2	3	الَّذِينَ يُؤْمِنُونَ بِالْغَيْبِ وَيُقِيمُونَ الصَّلَاةَ وَمِمَّا رَزَقْنَاهُمْ يُنْفِقُونَ
12	1	2	4	وَالَّذِينَ يُؤْمِنُونَ بِمَا أُنْزِلَ إِلَيْكَ وَمَا أُنْزِلَ مِنْ قَبْلِكَ وَبِالْآخِرَةِ هُمْ يُوقِنُونَ
13	1	2	5	أُولَٰئِكَ عَلَىٰ هُدًى مِنْ رَبِّهِمْ وَأُولَٰئِكَ هُمُ الْمُفْلِحُونَ
14	1	2	6	إِنَّ الَّذِينَ كَفَرُوا سَوَاءٌ عَلَيْهِمْ أَأَنْذَرْتَهُمْ أَمْ لَمْ تُنْذِرْهُمْ لَا يُؤْمِنُونَ
15	1	2	7	خَتَمَ اللَّهُ عَلَىٰ قُلُوبِهِمْ وَعَلَىٰ سَمْعِهِمْ وَعَلَىٰ أَبْصَارِهِمْ غِشَاوَةٌ وَلَهُمْ عَذَابٌ عَظِيمٌ
16	1	2	8	وَمِنَ النَّاسِ مَنْ يَقُولُ آمَنَّا بِاللَّهِ وَبِالْيَوْمِ الْآخِرِ وَمَا هُمْ بِمُؤْمِنِينَ
17	1	2	9	يُخَادِعُونَ اللَّهَ وَالَّذِينَ آمَنُوا وَمَا يَخْدَعُونَ إِلَّا أَنْفُسَهُمْ وَمَا يَشْعُرُونَ
18	1	2	10	فِي قُلُوبِهِمْ مَرَضٌ فَزَادَهُمُ اللَّهُ مَرَضًا وَلَهُمْ عَذَابٌ أَلِيمٌ بِمَا كَانُوا يَكْذِبُونَ
19	1	2	11	وَإِذَا قِيلَ لَهُمْ لَا تُفْسِدُوا فِي الْأَرْضِ قَالُوا إِنَّمَا نَحْنُ مُصْلِحُونَ
20	1	2	12	أَلَا إِنَّهُمْ هُمُ الْمُفْسِدُونَ وَلَٰكِنْ لَا يَشْعُرُونَ
21	1	2	13	وَإِذَا قِيلَ لَهُمْ آمِنُوا كَمَا آمَنَ النَّاسُ قَالُوا أَنُؤْمِنُ كَمَا آمَنَ السُّفَهَاءُ أَلَا إِنَّهُمْ هُمُ السُّفَهَاءُ وَلَٰكِنْ لَا يَعْ
22	1	2	14	وَإِذَا لَقُوا الَّذِينَ آمَنُوا قَالُوا آمَنَّا وَإِذَا خَلَوْا إِلَىٰ شَيَاطِينِهِمْ قَالُوا إِنَّا مَعَكُمْ إِنَّمَا نَحْنُ مُسْتَهْزِئُونَ
23	1	2	15	اللَّهُ يَسْتَهْزِئُ بِهِمْ وَيَمُدُّهُمْ فِي طُغْيَانِهِمْ يَعْمَهُونَ
24	1	2	16	أُولَٰئِكَ الَّذِينَ اشْتَرَوُا الضَّلَالَةَ بِالْهُدَىٰ فَمَا رَبِحَتْ تِجَارَتُهُمْ وَمَا كَانُوا مُهْتَدِينَ
25	1	2	17	مَثَلُهُمْ كَمَثَلِ الَّذِي اسْتَوْقَدَ نَارًا فَلَمَّا أَضَاءَتْ مَا حَوْلَهُ ذَهَبَ اللَّهُ بِنُورِهِمْ وَتَرَكَهُمْ فِي ظُلُمَاتٍ لَا يُبْصِرُونَ
26	1	2	18	صُمٌّ بُكْمٌ عُمْيٌ فَهُمْ لَا يَرْجِعُونَ

Gambar 4.10 Database Al-Qur'an Digital dalam bentuk Excel

2. Sebagai uji coba diambil 1 buah surat dari database tersebut, yaitu surah Al - Mulk $(QS : 67)$. Ayat - ayat pada surah Al - Mulk kemudian disalin dan disimpan dalam menggunakan aplikasi **Notepad** dengan sistem pengkodean karakter (*Encoding*) UTF-8. File disimpan menjadi dua buah file berbeda, 1 file berisi data surah Al - Mulk yang asli (**diberi nama: Al-Mulk(Original).txt**) dan file lainnya data surah Al - Mulk yang akan dimodifikasi (**diberi nama: Al-Mulk(Edited).txt**).



Gambar 4.11 Proses penyimpanan 2 file simulasi

3. Panjang data *input* atau *dataset* dapat dilihat dengan membuka panel **Properties File**. Seluruh dataset surah Al-Mulk berukuran 5,36 KB (5,495-byte) atau 43,960-bit

4. Dengan menggunakan HashTab, diperoleh nilai hash dari seluruh surah Al - Mulk dengan menggunakan algoritma fungsi hash Whirlpool adalah:

Nilai Hash : 526E04C21C10468AAB70D681ABEB2600EC79C3D1A36328979A2BA4DE8A9C72F2F1AC28D53431CA3344CF6C3560A3F85738DDC8F24C6CAE3A71D5082C00F95D0E

Berikut ini merupakan hasil uji coba kemampuan Algoritma Whirlpool dalam mendeteksi perubahan input data pada dataset

	Al-Mulk (Original)	Al-Mulk (Edited)
Ayat	الَّذِي خَلَقَ سَبْعَ سَمَاوَاتٍ طِبَاقًا ۖ مَا تَرَىٰ فِي خَلْقِ الرَّحْمَٰنِ مِنْ تَفَاوُتٍ ۖ فَارْجِعِ الْبَصَرَ هَلْ تَرَىٰ مِنْ فُطُورٍ	الَّذِي خَلْقُ سَبْعَ سَمَاوَاتٍ طِبَاقًا ۖ مَا تَرَىٰ فِي خَلْقِ الرَّحْمَٰنِ مَنْ تَفَاوُتٍ ۖ فَارْجِعِ الْبَصَرُ هَلْ تَرَىٰ مِنْ فُطُورٍ
Nilai Hash File	526E04C21C10468AAB70D681ABE B2600EC79C3D1A36328979A2BA4 DE8A9C72F2F1AC28D53431CA3344 CF6C3560A3F85738DDC8F24C6CA E3A71D5082C00F95D0E	04699A360149A57852AC547E171E5 41E1631B635ACBE5D03B74BCEC11 3F750CCAA784E91084B9D3924FBE 42A5AEF2ACCFA50F5494F5F8B2B5 0B15EAB647D8E4E
Ayat	وَلِلَّذِينَ كَفَرُوا بِرَبِّهِمْ عَذَابُ جَهَنَّمَ ۖ وَبِئْسَ الْمَصِيرُ	وَلِلَّذِينَ كَفَرُوا بِرَبِّهِمْ عَذَابُ جَهَنَّمَ وَبِئْسَ الْمَصِيرُ
Nilai Hash File	526E04C21C10468AAB70D681ABE B2600EC79C3D1A36328979A2BA4 DE8A9C72F2F1AC28D53431CA3344 CF6C3560A3F85738DDC8F24C6CA E3A71D5082C00F95D0E	F5FE98AB9E7DDFDA08CE8A7D8C 579267E9801E6FD6C350A9B849F40 162DA40CEACC3A26BD45FDD0BD C0A79291AFDA98727E1E41D05634 374E562395B001DA68

Gambar 4.12 ***Avalanche Effect*** **Algoritma Whirlpool terhadap perubahan harakat dan huruf**

Tabel di atas merupakan uji coba terhadap perubahan karakter dalam dataset surah Al - Mulk. Nilai hash yang berwarna jingga menandakan bahwa nilai tersebut berubah dari nilai hash yang seharusnya. Berikut ini adalah kesimpulan yang diperoleh:

1. Pada percobaan pertama, 2 huruf pada ayat ke - 3 dalam *dataset* Al - Mulk mengalami perubahan harakat, yaitu:

خَلَقَ → خَلْقُ

الْبَصَرَ → الْبَصَرُ

Menyebabkan nilai hash dari *dataset* Al - Mulk mengalami perubahan sebanyak 59 karakter (atau 59-Byte).

2. Pada percobaan kedua, sebuah tanda waqaf yang terletak pada ayat ke-6 dalam *dataset* Al - Mulk dihapus. Hal ini menyebabkan nilai hash dari *dataset* Al - Mulk mengalami perubahan sebanyak 56 karakter (atau 56-Byte).

Percobaan selanjutnya adalah menukar posisi lafadz suatu ayat dan huruf pada suatu ayat.

	Al-Mulk (Original)	Al-Mulk (Edited)
Ayat	تَبَارَكَ الَّذِي بِيَدِهِ الْمُلْكُ وَهُوَ عَلَىٰ كُلِّ شَيْءٍ قَدِيرٌ	تَبَارَكَ الْمُلْكُ بِيَدِهِ الَّذِي وَهُوَ عَلَىٰ كُلِّ شَيْءٍ قَدِيرٌ
Nilai Hash File	526E04C21C10468AAB70D681ABE B2600EC79C3D1A36328979A2BA4 DE8A9C72F2F1AC28D53431CA3344 CF6C3560A3F85738DDC8F24C6CA E3A71D5082C00F95D0E	A5C80D3FC7F34CC63589E75DD94E DC20336195DDE26E16F077A8026E FF9CCC79839EE50AD520704796AA 3B2243F8B3B371BA5AF5505902BE49 DAB7ECAC3B4973F
Ayat	الَّذِي خَلَقَ الْمَوْتَ وَالْحَيَاةَ لِيَبْلُوَكُمْ أَيُّكُمْ أَحْسَنُ عَمَلًا ۚ وَهُوَ الْعَزِيزُ الْغَفُورُ	الَّذِي خَلَكَ الْمَوْتَ وَالْحَيَاةَ لِيَبْلُوَكُمْ أَيُّقَمْ أَحْسَنُ عَمَلًا ۚ وَهُوَ الْعَزِيزُ الْغَفُورُ
Nilai Hash File	526E04C21C10468AAB70D681ABE B2600EC79C3D1A36328979A2BA4 DE8A9C72F2F1AC28D53431CA3344 CF6C3560A3F85738DDC8F24C6CA E3A71D5082C00F95D0E	224BDE7B7F8A3DE6CFC987FD41E DE88BE56CA347A1F67E6892 1150B 1408B9D432416D30ABBBA9123602 F1E317D7D863E2B3C2C0F5A739BE 8C750E892A905122F

Gambar 4.13 ***Avalanche Effect*** **Algoritma Whirlpool terhadap perubahan perpindahan posisi lafadz dan Huruf**

Berikut ini adalah kesimpulan yang diperoleh:

1. Pada percobaan pertama, 2 lafadz pada ayat ke-1 dalam *dataset* Al-Mulk bertukar posisi, yaitu:

الْمُلْكُ بِيَدِهِ الَّذِي → الَّذِي بِيَدِهِ الْمُلْكُ

menyebabkan nilai hash dari *dataset* Al - Mulk mengalami perubahan sebanyak 58 karakter (atau 58-Byte).

2. Pada percobaan Kedua, 2 huruf pada ayat ke-2 dalam *dataset* Al - Mulk bertukar posisi, yaitu:

خَلَقَ → خَلَكَ

أَيُّكُمْ → أَيُّقَمْ

Menyebabkan nilai hash dari *dataset* Al - Mulk mengalami perubahan sebanyak 56 karakter (atau 56-Byte).

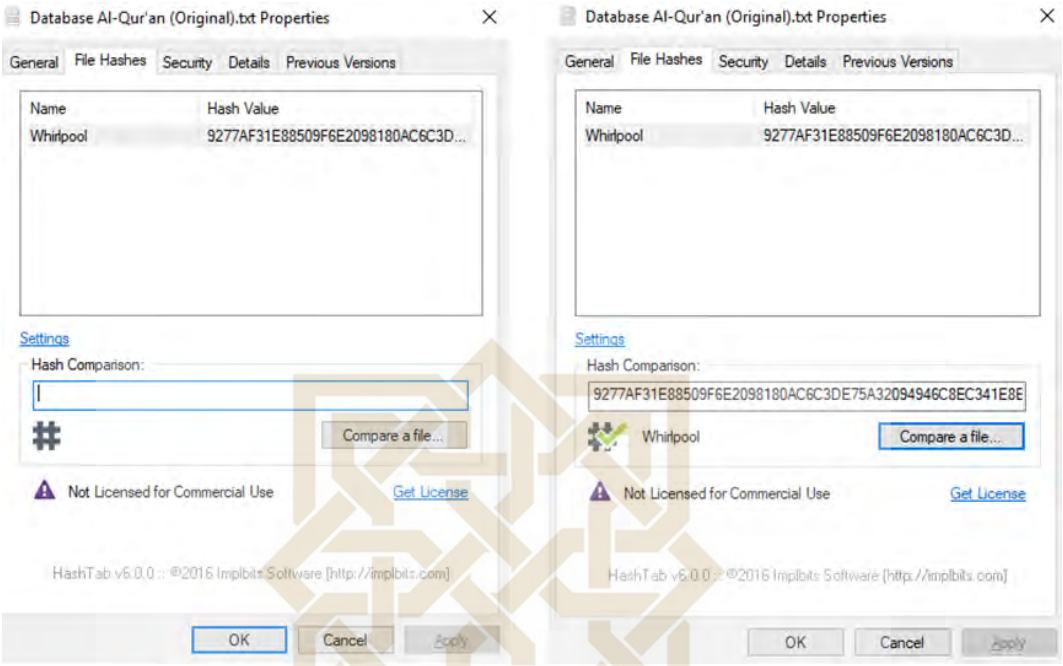
Percobaan terakhir adalah mendeteksi perubahan data pada database Al - Quran Digital. *Database* Al - Quran ini memiliki ukuran 1,26 MB (1,326,609-byte) atau 10,612,872-bit.

Langkah-langkah:

1. Seluruh data ayat Al-Quran pada *database* disalin dan disimpan menggunakan aplikasi **Notepad** dengan sistem pengkodean UTF-8. Dibuat dua buah file berbeda, file pertama berisikan *database* Al - Quran asli (**diberi nama: Database Al-Quran (Original)**) dan file kedua berisikan *database* Al - Quran modifikasi (**diberi nama: Database Al-Quran (Edited)**).

Gambar 4.14 Database Al-Qur'an Digital dalam bentuk Notepad

2. Klik kanan pada file *Database* Al-Quran (Original)), kemudian pilih opsi **properties**. Pilih tab **File Hashes** dan klik **Compare a File**. Pilih file **Database Al-Quran (Edited)** yang sudah disimpan sebelumnya.



Gambar 4.15 Tampilan HashTab dari file Database Al-Quran (Original)

3. Apabila muncul tanda centang, hal tersebut menunjukan bahwa integritas kedua data terjaga, yang menandakan kedua file memuat data yang sama dan tidak ada perubahan data yang terjadi.

Berikut ini adalah hasil pengujian *Avalanche Effect* algoritma Whirlpool pada **Database Al-Quran (Edited)**.

	Database Al-Qur'an (Original)	**Database Al-Qur'an (Edited)**
	-	Memindahkan posisi ayat ke-2 surah al-baqarah ke surah an-nas
Nilai Hash File	9277AF31E88509F6E2098180AC6C3 DE75A32094946C8EC341E8E2F22D B35EF6CD43B5FAB567E18C05D292 548D873947840A204D625401008CE 73713F2B93FDFE	049B6E330FF638A7644AC954652D2 98A243B82E627779C7F151B8142280 9BCC66E17AA5A8B5C56BCA48FE0 7303357169151ACEE351B97306AB 7E242AA5E2DA7
	-	Menghilangkan lafadz "Bismillah" pada permulaan Al-Fatiha
Nilai Hash File	9277AF31E88509F6E2098180AC6C3 DE75A32094946C8EC341E8E2F22D B35EF6CD43B5FAB567E18C05D292 548D873947840A204D625401008CE 73713F2B93FDFE	BCFFB29B46CF8914904B51AD5E4F 7017BAD5EF6451A384B33E33302E BDC9DC6E445406AA8B22708D3C CA161B91D2CAE7C012906F0DB52 08567FD6B19F62E81
	لَمَجْمُوعُونَ إِلَىٰ مِيقَاتِ يَوْمٍ مَعْلُومٍ QS:Al-Waqiah:50	عُونَ إِلَىٰ مِيقَاتِ يَوْمٍ مَعْلُومٍ Menghilangkan lafadz
Nilai Hash File	9277AF31E88509F6E2098180AC6C3 DE75A32094946C8EC341E8E2F22D B35EF6CD43B5FAB567E18C05D292 548D873947840A204D625401008CE 73713F2B93FDFE	7B9B5ACEA9A1E5108DAE1434D63 5D3B2B966BCDFE84C8AB1E2BCC B1CFCF7C2F81DED1582E0E8E93E6 35E3F3B8AF14F81CF1B7168006277 B67BE9A2B89FEB4E0B
	قَالَ فَمَا بَالُ الْقُرُونِ الْأُولَىٰ QS:Ta-Ha:51	قَالَ فَمَا بَالِ الْقُرُونِ الْأُولَىٰ Mengganti harakat
Nilai Hash File	9277AF31E88509F6E2098180AC6C3 DE75A32094946C8EC341E8E2F22D B35EF6CD43B5FAB567E18C05D292 548D873947840A204D625401008CE 73713F2B93FDFE	F0936EA3EC5DA5D2CE65B63D027C E92C2BF78C18D531EB0AB1C09A39 EBD3A215E346F2ABBE9F6655B4B A53618EE9A8C4F057978318C03C97 EFC74B1B534B9FAD

Gambar 4.16 *Avalanche Effect* Algoritma Whirlpool terhadap perubahan data Al-Qur'an Digital

Dari percobaan yang telah dilakukan, diperoleh beberapa kesimpulan:

1. Algoritma fungsi hash Whirlpool, mampu mendeteksi 1 perubahan Byte dalam sebuah database berukuran 1,326,609-Byte.

2. Algoritma fungsi hash Whirlpool, memiliki kemampuan untuk mendeteksi perubahan data, dengan perubahan nilai hash yang sangat signifikan (mempengaruhi perubahan nilai hash-nya hingga 90% - 100%). Hal ini menunjukan, bahwa tingkat *avalanche effect* yang dimiliki oleh Whirlpool sangat tinggi untuk sebuah database berukuran lebih kecil atau sama dengan 1,326,609-byte.

BAB V

PENUTUP

5.1. Kesimpulan

Penelitian ini terbagi menjadi 4 kajian utama, yaitu pertama mengkaji struktur aljabar dan fungsi - fungsi yang ada pada Whirlpool, kedua menjelaskan cara kerja proses *hashing* pada Whirlpool, ketiga menjelaskan proses *hashing* dengan Algoritma Whirlpool menggunakan program komputer, dan terakhir adalah menjelaskan bagaimana proses verifikasi integritas data pada database Al-Quran Digital, menggunakan program komputer dengan Algoritma Whirlpool.

Berdasarkan penelitian yang telah dilakukan, diperoleh beberapa kesimpulan:

1. Struktur aljabar yang digunakan pada algoritma Whirlpool, $GF(2^8)$, merupakan Lapangan Galois berorde 2^8. Lapangan Galois disebut juga dengan Lapangan Hingga, karena setiap elemen pada lapangannya berhingga. Bilangan prima 2 dipilih sebagai karakteristik dari lapangan, karena dalam sistem komputer, hanya ada 2 angka yang dapat dipahami oleh mesin, yaitu 0 dan 1. Lapangan tersebut berdimensi 8, karena 1 karakter pada pesan input, memiliki ukuran 8-bit.

2. Whirlpool mengolah pesan input menggunakan 4 fungsi utama, yaitu *SubByte*, *ShiftColumns*, *MixRows*, dan *AddRoundKey*. Pada setiap pesan input yang memiliki panjang lebih dari 512-bit atau 64-Byte, algoritma Whirlpool akan membagi-bagi pesan tersebut menjadi blok-blok pesan

dengan menggunakan skema kontruksi pesan blok Merkle-Damgard. Proses *padding bits* selalu dilakukan pada setiap pesan input, baik yang memiliki panjang lebih dari 512-bit maupun kurang dari itu. Hal ini bertujuan agar setiap blok-blok pesan memiliki panjang berkelipatan 512-bit. Setelah itu, blok-blok pesan akan diolah menggunakan 4 fungsi utama Whirlpool, agar dihasilkan sebuah blok - blok cipherteks. Setelah itu, blok - blok cipherteks akan diolah menggunakan fungsi kompresi dengan skema miyaguchi-preneel. Proses ini akan menghasilkan Nilai Hash dari pesan input.

3. Perbedaan proses *hashing* pada teks berbasis ASCII dan UNICODE, terletak pada penempatan Byte-byte karakter pada matriks CState. Karakter-karakter ASCII memiliki ukuran 1-Byte, sehingga algoritma Whirlpool akan menempatkan 1 karakter ASCII pada 1 entri matriks CState. Pada Karakter-karakter Arabic yang berbasis UNICODE, algoritma Whirlpool akan menempatkan 1 karakter Arabic pada 2 entri matriks CState. Hal ini dikarenakan setiap karakter Arabic memiliki ukuran sebesar 16-Byte.

4. Proses verifikasi integritas data Al - Qur'an Digital dilakukan dengan membandingkan nilai hash dari suatu *database* Al - Qur'an Digital, dengan nilai hash dari *database* Al - Qur'an Digital yang masih otentik.

5. Metode verifikasi integritas data menggunakan perbandingan fungsi hash, hanya sebatas mendeteksi adanya perubahan data, yaitu dengan melihat Nilai Hash data yang asli lalu membandingkannya dengan Nilai Hash pada data yang telah mengalami perubahan. Kekurangan dari metode ini adalah tidak dapat digunakan untuk menentukan letak bit atau Byte yang berubah dan tidak dapat digunakan untuk mengkoreksi data yang berubah.

5.2. Saran

Seperti yang telah dijelaskan sebelumnya, proses verifikasi integritas data hanya sebatas mendeteksi adanya perubahan data pada suatu *dataset/database*. Metode ini tidak dapat digunakan untuk menentukan letak error pada suatu *data/database* dan juga tidak memiliki kemampuan untuk mengkoreksi suatu *error*, sehingga untuk penelitian selanjutnya, peneliti menyarankan untuk menggunakan Teori pengkodean, atau $Coding\ Theory$. Dengan menggunakan teori ini, maka *error* pada suatu *database* dapat terdeteksi, baik letak *error* nya, maupun jumlah *error* yang terjadi dan suatu *error* dapat dikoreksi agar kembali seperti aslinya.