TESIS - SM 142501

KONSTRUKSI SUATU ALGORITMA KRIPTOGRAFI MENGGUNAKAN TRANSFORMASI MAX PLUS WAVELET

JOKO CAHYONO
NRP 1214 201 007

Dosen Pembimbing:
Dr. Subiono, M.S.

PROGRAM MAGISTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

THESIS - SM 142501

CONSTRUCTION OF A CRYPTOGRAPHIC ALGORITHM USING MAX-PLUS-WAVELET TRANSFORMS

JOKO CAHYONO
NRP 1214 201 007

Supervisor:
Dr. Subiono, M.S.

MASTER'S DEGREE
MATHEMATICS DEPARTMENT
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016

LEMBAR PENGESAHAN

KONSTRUKSI SUATU ALGORITMA KRIPTOGRAFI MENGGUNAKAN TRANSFORMASI MAX PLUS WAVELET

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar Magister Sains (M.Si) di Institut Teknologi Sepuluh Nopember

Oleh :
JOKO CAHYONO
NRP. 1214201007

Tanggal Ujian : 24 Mei 2016
Periode Wisuda : September 2016

Disetujui oleh :

1. **Dr. Subiono, M.S.** **(Pembimbing)**
NIP. 19570411 198403 1 001

2. **Dr. Mahmud Yunus, M.Si** **(Penguji)**
NIP. 19620407 198703 1 005

3. **Dr. Dieky Adzkiya, S.Si, M.Si** **(Penguji)**
NIP. 19830517 200812 1 003

4. **Dr. Imam Mukhlash, S.Si, MT** **(Penguji)**
NIP. 19700831 199403 1 003

Direktur Program Pascasarjana

Prof. Ir. Djauhar Manfaat, M.Sc, Ph.D
NIP. 19601202 198701 1 001

KONSTRUKSI SUATU ALGORITMA KRIPTOGRAFI MENGGUNAKAN TRANSFORMASI MAX PLUS WAVELET

Nama Mahasiswa : Joko Cahyono
NRP : 1214 201 007
Pembimbing : Dr. Subiono, M.S.

ABSTRAK

Kriptografi berperan menjaga keamanan suatu informasi. Sampai saat ini sudah banyak dibuat berbagai algoritma kriptografi. Dalam tesis ini dikontruksi suatu algoritma kriptografi berdasarkan transformasi max plus wavelet. Enkripsi disusun berdasarkan proses analisis dari transformasi max plus wavelet, sedangkan dekripsi disusun berdasarkan proses sintesis. Kunci kriptografi terdiri dari tiga bagian yaitu kode untuk tipe max plus wavelet yang digunakan, banyak kanal yang digunakan dan kode untuk sinyal detail. Proses kriptografi ini hanya melibatkan operasi maksimum dan tambah sebagai operasi utama. Berdasarkan hasil uji coba dan analisis didapat bahwa algoritma kriptografi ini adalah baik berdasarkan korelasi antara *plaintext* dan *ciphertext*, kualitas enkripsi serta besarnya *key space*. Algoritma ini juga efisien dari segi waktu karena memiliki kompleksitas O(n) atau kompleksitas linier.

Kata kunci: Kriptografi, Kunci, Transformasi Max Plus Wavelet

CONSTRUCTION OF A CRYPTOGRAPHIC ALGORITHM USING MAX-PLUS-WAVELET TRANSFORMS

Name : Joko Cahyono
NRP : 1214 201 007
Supervisor : Dr. Subiono, M.S.

ABSTRACT

Cryptography has a role to secure an information. Until now many varieties of cryptographic algorithms have been constructed. In this thesis, we construct a cryptographic algorithm based on max-plus-wavelet transforms. Encryption and decryption are made based on the analysis and synthesis process of max-plus-wavelet transforms, respectively. The key consists of three elements. The first element is code for type of max-plus wavelet used. The second element is the number of the channels used. The third element is the code of signal details. The cryptographic process involves only maximum and addition operations as main operations. The experiment and analysis show that the algorithm is a good cryptographic algorithm based on correlation between the plaintext and ciphertext, encryption quality and the key space. This algorithm is also efficient in the running time, because it has complexity O(n) or linear complexity.

Keywords: Cryptography, Key, Max-Plus-Wavelet Transform

KATA PENGANTAR

Alhamdulillahirobbil'alamin, puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis berhasil menyelesaikan tesis yang berjudul **"KONSTRUKSI SUATU ALGORITMA KRIPTOGRAFI MENGGUNAKAN TRANSFORMASI MAX PLUS WAVELET"**.

Banyak pihak yang telah membantu penulis dalam menyelesaikan tesis ini. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Bapak Dr. Imam Mukhlash, S.Si, MT, selaku Ketua Jurusan Matematika ITS sekaligus sebagai dosen penguji, yang telah memberikan koreksi dan saran untuk menyempurnakan tesis ini.

2. Bapak Dr. Mahmud Yunus, M.Si, selaku Ketua Program Studi Pascasarjana Jurusan Matematika ITS sekaligus sebagai dosen penguji, yang telah memberikan koreksi dan saran untuk menyempurnakan tesis ini.

3. Bapak Dr. Subiono, M.S., selaku dosen pembimbing yang telah meluangkan waktu dan ilmunya dengan kesabaran dan penuh pengertian.

4. Bapak Dr. Dieky Adzkiya, S.Si, M.Si, selaku dosen penguji yang telah memberikan koreksi dan saran untuk menyempurnakan tesis ini.

5. Bapak Dr. Drs. Chairul Imron, M.I.Komp, selaku dosen wali yang telah banyak memberikan masukan dan arahan kepada penulis.

6. Bapak dan Ibu dosen Program Studi Pascasarjana Jurusan Matematika ITS, yang telah banyak memberikan ilmunya kepada penulis.

7. Bapak dan Emak tercinta yang telah memberikan segenap perhatian untuk mendidik, serta memberikan bantuan, dorongan moral dan do'a yang tiada hentinya untuk penulis.

8. Bapak dan Ibu mertua yang telah memberikan segenap perhatian, kasih sayang dan do'a untuk penulis dan untuk anak dan istri penulis.

9. Istriku tercinta, Lestin Tatik Julaika, yang telah setia dan sabar menemani, memberikan segenap dukungan, perhatian dan do'a yang tiada hentinya untuk penulis dalam keadaan suka maupun duka.
10. Anakku tersayang, Shabieq El-Fathin Attaraufaa', yang telah menjadi penyemangat hidup. Semoga akan menjadi anak yang sholeh dan berilmu.
11. Semua pihak yang tidak bisa penulis sebutkan satu persatu, seluruh keluarga penulis, staf TU Pascasarjana Matematika ITS, Bapak Kistosil Fahim S.Si, M.Si, dan teman-teman Pascasarjana Matematika ITS angkatan 2014 semuanya yang telah banyak membantu penulis selama kuliah.

Penulis yakin laporan tesis ini masih banyak kekurangan. Oleh karena itu penulis sangat terbuka menerima setiap saran dan kritik dari pembaca tesis ini.

Surabaya, Mei 2016

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SIMBOL

$\oplus$	:	operasi penjumlahan dalam aljabar max plus
$\otimes$	:	operasi perkalian dalam aljabar max plus
$\oslash$	:	operasi pembagian dalam aljabar max plus
$\bigoplus_{i=1}^{p} a_i$	:	$a_1 \oplus a_2 \oplus \ldots \oplus a_p$
ε	:	$-\infty$
$\mathbb{R}_\varepsilon$	:	$\mathbb{R} \cup \{-\infty\}$, $\mathbb{R}$ himpunan semua bilangan real
$\mathbb{Z}_\varepsilon$	:	$\mathbb{Z} \cup \{-\infty\}$, $\mathbb{Z}$ himpunan semua bilangan bulat
$\mathbb{N}$	:	himpunan semua bilangan asli
V_j, W_j	:	ruang sinyal ke-j
x_j	:	sinyal anggota dari V_j
y_j	:	sinyal anggota dari W_j
$\psi_j^{\uparrow}$	:	operator analisis untuk sinyal yang memetakan ruang sinyal V_j ke V_{j+1}
$\omega_j^{\uparrow}$	:	operator analisis untuk sinyal yang memetakan ruang sinyal V_j ke W_{j+1}
$\Psi_j^{\downarrow}$	:	operator sintesis untuk sinyal yang memetakan ruang sinyal (V_{j+1}, W_{j+1}) ke V_j

BAB I
PENDAHULUAN

Pada bab ini diuraikan tentang latar belakang yang mendasari penulisan tesis, yang meliputi identifikasi permasalahan, beberapa informasi tentang penelitian terdahulu yang berhubungan dengan topik tesis dan hal-hal yang akan dilakukan pada penyelesaian tesis. Kemudian dirumuskan permasalahan yang akan dibahas, tujuan, batasan masalah, manfaat dan kontribusi penelitian tesis ini.

1.1 Latar Belakang

Perkembangan teknologi informasi dan komunikasi sekarang ini sangat berpengaruh terhadap segala aspek kehidupan, salah satunya dalam proses pengiriman pesan. Dengan adanya teknologi yang terus berkembang memungkinkan manusia dapat berkomunikasi dan saling bertukar informasi secara jarak jauh. Keamanan dan kerahasiaan sebuah data atau informasi dalam komunikasi menjadi hal yang sangat penting.

Salah satu cara untuk mengatasi masalah keamanan dan kerahasiaan informasi tersebut adalah menggunakan kriptografi. Kriptografi berasal dari bahasa Yunani, yang terdiri dari dua kata yaitu *cryptos* dan *graphein*. *Cryptos* berarti rahasia, dan *graphein* berarti tulisan. Sehingga menurut bahasa, kriptografi berarti tulisan rahasia.

Sedangkan definisi kriptografi menurut Sentot Kromodimoeljo (2010) adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Kriptografi memiliki dua konsep utama yaitu enkripsi dan dekripsi. Enkripsi adalah sebuah proses penyandian yang melakukan perubahan kode dari yang dapat dimengerti (*plaintext*) menjadi sebuah kode yang tidak dapat dimengerti (*ciphertext*), sedangkan proses kebalikannya yaitu mengubah *ciphertext* menjadi *plaintext* disebut dekripsi. Proses enkripsi dan dekripsi memerlukan suatu kunci rahasia yang disepakati oleh kedua belah pihak.

Debayan Goswami dkk (2011) dalam penelitiannya yang berjudul *A Discrete Wavelet Transform based Cryptographic algorithm* mengusulkan

penggunaan transformasi wavelet diskrit dalam kriptografi. Transformasi wavelet yang digunakan adalah transformasi wavelet Daubechies. Proses enkripsi, dekripsi dan penyusunan kunci menggunakan proses transformasi wavelet ini.

Dalam tesis di Institut Teknologi Sepuluh Nopember yang berjudul *Konstruksi Transformasi Wavelet Menggunakan Aljabar Max Plus*, Kistosil Fahim (2014) membahas tentang transformasi max plus wavelet, yaitu transformasi wavelet menggunakan aljabar max plus. Dalam penelitian ini dikontruksi beberapa tipe dari max plus wavelet dan contoh penggunaannya dalam pemampatan citra.

Aljabar max plus dan aljabar min plus juga telah digunakan dalam kriptografi. Dima Grigoriev dan Vladimir Shpilrain (2013) dalam penelitiannya yang berjudul *Tropical Cryptography*, serta Mariana Durcheva (2015) dalam penelitiannya yang berjudul *Some applications of idempotent semirings in Public Key Cryptography* membahas tentang penggunaan aljabar max plus dan aljabar min plus dalam kriptografi. Penggunaan aljabar max plus dan aljabar min plus disini adalah pada penyusunan kunci.

Berdasarkan uraian tersebut, pada tesis ini dibahas mengenai konstruksi suatu algoritma kriptografi menggunakan transformasi max plus wavelet. Transformasi max plus wavelet ini digunakan untuk proses enkripsi, dekripsi dan penyusunan kunci. Untuk implementasi dari algoritma kriptografi ini dibuat suatu program kriptografi menggunakan software Scilab 5.5.2. Selanjutnya dianalisis kelayakan dari algoritma kriptografi max plus wavelet yang dihasilkan. Analisis dilakukan dengan cara menentukan korelasi linier antara *plaintext* dan *ciphertext*, kualitas enkripsi, *running time* program, analisis kompleksitas dan analisis *key space* algoritma kriptografi.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, permasalahan yang dibahas dalam tesis ini adalah:

a. Bagaimana mengkontruksi algoritma kriptografi menggunakan transformasi max plus wavelet.

b. Bagaimana kelayakan algoritma kriptografi max plus wavelet yang dihasilkan berdasarkan nilai korelasi linier antara *plaintext* dan *ciphertext*, kualitas enkripsi, *running time* program, kompleksitas dan *key space* algoritma kriptografi.

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tesis ini dibatasi sebagai berikut:

a. Transformasi max plus wavelet yang digunakan adalah transformasi max plus wavelet tipe A, B, C, D dan E pada penelitian Kistosil Fahim (2014).

b. Data yang dienkripsi berupa teks yang disimpan dalam file dengan format txt.

c. Analisis kelayakan algoritma kriptografi max plus wavelet dilakukan dengan cara menentukan korelasi linier antara *plaintext* dan *ciphertext*, kualitas enkripsi, *running time* program, analisis kompleksitas dan analisis *key space* algoritma kriptografi.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah

a. Mengkontruksi suatu algoritma kriptografi menggunakan transformasi max plus wavelet.

b. Mengetahui kelayakan algoritma kriptografi max plus wavelet berdasarkan nilai korelasi linier antara *plaintext* dan *ciphertext*, kualitas enkripsi, *running time* program, kompleksitas dan *key space* algoritma kriptografi.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah memperoleh metode baru untuk proses enkripsi, dekripsi dan penyusunan kunci dalam kriptografi menggunakan transformasi max plus wavelet. Serta mengetahui kelayakan algoritma kriptografi max plus wavelet yang dikonstruksi tersebut berdasarkan korelasi linier antara *plaintext* dan *ciphertext*, kualitas enkripsi, *running time* program, kompleksitas dan *key space* algoritma kriptografi.

1.6 Kontribusi Hasil Penelitian

Kontribusi hasil penelitian ini terhadap pengembangan ilmu adalah sebagai solusi baru dalam kriptografi, yaitu menggunakan transformasi max plus wavelet. Sehingga selanjutnya diharapkan dapat dikembangkan untuk kepentingan pengamanan data yang lain.

BAB II
TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai teori-teori tentang kriptografi, aljabar max plus, transformasi wavelet, transformasi max plus wavelet dan analisis algoritma kriptografi.

2.1 Kriptografi

2.1.1 Definisi Kriptografi

Kriptografi berasal dari bahasa Yunani, *cryptos* yang berarti rahasia dan *graphein* yang berarti tulisan. Jadi menurut bahasa, kriptografi merupakan tulisan rahasia. Ada beberapa definisi kriptografi yang telah dikemukakan di berbagai literatur. Menurut Sentot Kromodimoeljo (2010) kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Pada proses dekripsi digunakan kunci dekripsi untuk mendapatkan kembali data asli. Biasanya algoritma kriptografi tidak dirahasiakan. Rahasia terletak di beberapa parameter (kunci) yang digunakan.

2.1.2 Terminologi Kriptografi

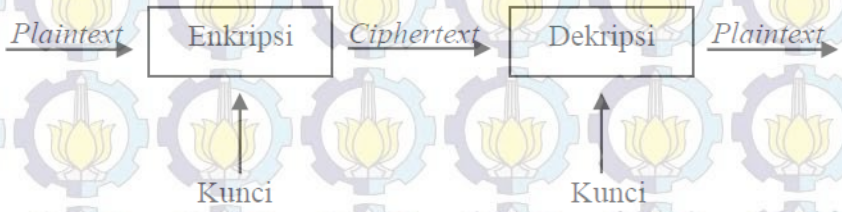
Di dalam kriptografi ada beberapa istilah penting yang sering digunakan yaitu:

a. Pesan, *Plaintext*, dan *Ciphertext*
   Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext*. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut *ciphertext*.

b. Pengirim dan Penerima
   Suatu aktifitas komunikasi data akan melibatkan pertukaran pesan antara dua entitas, yaitu pengirim dan penerima. Pengirim adalah entitas dalam komunikasi yang mengirim informasi kepada entitas lainnya. Penerima adalah entitas dalam komunikasi yang diharapkan

menerima informasi dari entitas lainnya. Entitas dapat berupa orang, mesin, dan sebagainya.

c. Enkripsi dan Dekripsi
Enkripsi adalah proses penyandian yang melakukan perubahan sebuah kode dari yang dapat dimengerti (*plaintext*) menjadi sebuah kode yang tidak dapat dimengerti (*ciphertext*), sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi. Skema enkripsi dan dekripsi disajikan pada Gambar 2.1.



Gambar 2.1: Skema Enkripsi dan Dekripsi

d. Algoritma dan Kunci
Algoritma kriptografi yaitu aturan yang digunakan untuk enkripsi dan dekripsi. Pengiriman pesan dalam kriptografi membutuhkan kunci yang harus dijaga kerahasiaannya. Kunci adalah parameter yang digunakan untuk proses enkripsi dan dekripsi pesan. Kunci biasanya berupa deretan bilangan maupun string.

2.1.3 Tujuan Kriptografi

Tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi adalah sebagai berikut:

a. Kerahasiaan (*confidentiality*) merupakan layanan yang digunakan untuk menjaga isi informasi dari semua pihak yang tidak berhak untuk mendapatkannya kecuali pihak yang memiliki kunci rahasia untuk membuka informasi yang telah disandi.

b. Integritas Data (*data integrity*) merupakan layanan yang menjamin bahwa pesan masih asli. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak pihak yang tidak berhak antara lain penyisipan, penghapusan, dan penggantian data lain ke data yang sebenarnya.

c. Autentikasi (*authentication*) merupakan suatu layanan yang berhubungan dengan identifikasi, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

d. Tanpa penyangkalan (*non-repudiation*) merupakan layanan untuk mencegah terjadinya penyangkalan terhadap pengirim maupun penerima yang saling berkomunikasi.

2.1.4 Jenis Algoritma Kriptografi

Berdasarkan jenis kunci yang digunakan dalam proses enkripsi dan dekripsi, kriptografi dibedakan menjadi kriptografi kunci simetri dan kriptografi kunci asimetri.

a. Kriptografi Kunci Simetris
   Pada sistem kriptografi simetri, kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kunci. Istilah lain untuk kriptografi simetri adalah kriptografi kunci privat.

b. Kriptografi Kunci Asimetris
   Pada sitem kriptografi asimetri, kunci untuk proses enkripsi dan dekripsi menggunakan sepasang kunci yaitu kunci privat dan kunci publik. Kunci privat hanya diketahui oleh pengirim dan penerima pesan. Sedangkan kunci publik bisa diketahui oleh orang lain. Kriptografi asimetri juga disebut dengan kriptografi kunci publik.

Berdasarkan tipe data yang diolah dalam satu kali proses, kriptografi dibedakan menjadi kriptografi *stream cipher* dan *block cipher*.

a. *Stream Cipher*
   Pada sistem kriptografi *Stream Cipher*, data yang diolah dalam satu kali proses berupa data yang mengalir. Contoh dari kriptografi *Stream Cipher* adalah RC4.

b. *Block Cipher*
   Pada sistem kriptografi *block cipher*, data yang diolah dikumpulkan dalam blok-blok data. Sehingga dalam setiap proses yang diolah adalah satu blok data. Contoh dari kriptografi *block cipher* adalah DES, 3DES dan AES.

2.2 Aljabar Max Plus

Subiono (2015) dalam bukunya mendefinisikan struktur aljabar $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ dengan $\mathbb{R}_\varepsilon \stackrel{\text{def}}{=} \mathbb{R} \cup \{\varepsilon\}$ dan $\varepsilon \stackrel{\text{def}}{=} -\infty$, dan operator-operatornya didefinisikan sebagai berikut, untuk semua x, y anggota $\mathbb{R}_\varepsilon$ maka

$$x \oplus y \stackrel{\text{def}}{=} \max\{\mathrm{x}, \mathrm{y}\} \quad \text{dan} \quad x \otimes y \stackrel{\text{def}}{=} x + y$$

Struktur aljabar $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ merupakan semi-ring dengan elemen netral adalah ε dan elemen satuan adalah e = 0, yang memenuhi aksioma berikut:

a. $(\mathbb{R}_\varepsilon, \oplus)$ merupakan semigrup komutatif, yaitu $\forall x, y, z \in \mathbb{R}_\varepsilon$ memenuhi

$$\begin{aligned} x \oplus y &= \max\{x, y\} = \max\{y, x\} = y \oplus x \\ x \oplus (y \oplus z) &= \max\{x, \max\{y, z\}\} = \max\{x, y, z\} \\ &= \max\{\max\{x, y\}, z\} = (x \oplus y) \oplus z \\ x \oplus \varepsilon &= \max\{x, -\infty\} = \max\{-\infty, x\} \\ &= \varepsilon \oplus x = x \end{aligned}$$

b. $(\mathbb{R}_\varepsilon, \otimes)$ merupakan semigrup, yaitu $\forall x, y, z \in \mathbb{R}_\varepsilon$ memenuhi

$$\begin{aligned} x \otimes (y \otimes z) &= x + (y + z) = (x + y) + z \\ &= (x \otimes y) \otimes z \\ x \otimes 0 &= x + 0 = 0 + x = 0 \otimes x = x \end{aligned}$$

c. Adanya elemen penyerap ε yaitu

$$x \otimes \varepsilon = x + -\infty = -\infty + x = \varepsilon \otimes x = \varepsilon$$

d. Operasi $\otimes$ distributif terhadap $\oplus$, yaitu $\forall x, y, z \in \mathbb{R}_\varepsilon$ berlaku

$$\begin{aligned} (x \oplus y) \otimes z &= \max\{x, y\} + z \\ &= \max\{x + z, y + z\} = (x \otimes z) \oplus (y \otimes z) \\ x \otimes (y \oplus z) &= x + \max\{y, z\} \\ &= \max\{x + y, x + z\} = (x \otimes y) \oplus (x \otimes z) \end{aligned}$$

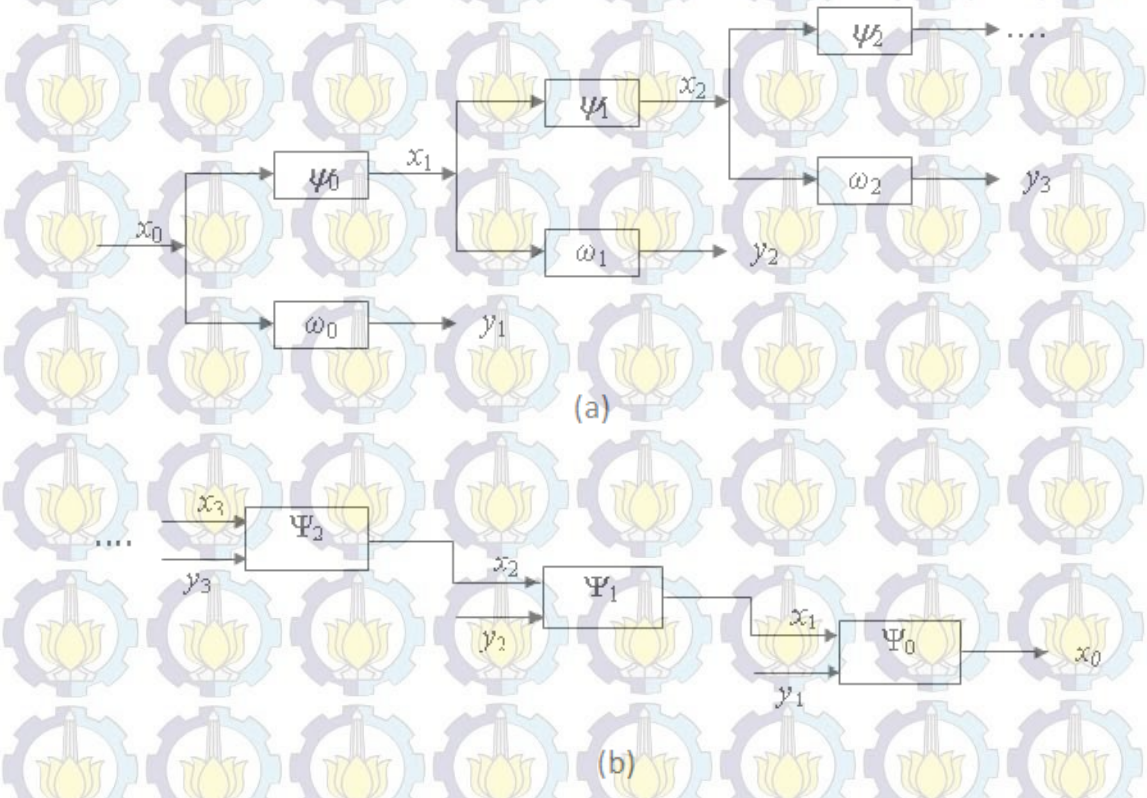
$(\mathbb{R}_\varepsilon, \oplus, \otimes)$ merupakan semi-ring komutatif yang sekaligus idempoten sebab

untuk setiap $x, y \in \mathbb{R}_\varepsilon$ berlaku $x \otimes y = x + y = y + x = y \otimes x$ dan $x \oplus x = \max\{x, x\} = x$.

Pada penelitian ini digunakan $\mathbb{Z}_\varepsilon = \mathbb{Z} \cup \{\varepsilon\}$ sehingga dalam komputasi menjadi lebih sederhana.

2.3 Transformasi Wavelet

Transformasi wavelet sangat berperan dalam proses pengolahan sinyal. Pada sinyal utama yang beresolusi tinggi dilakukan proses analisis sehingga didapatkan sinyal hampiran dan sinyal detail. Sinyal hampiran mampu merepresentasikan sinyal utama namun memiliki resolusi yang lebih rendah. Sinyal detail menjamin bahwa sinyal utama dapat diperoleh kembali dengan proses sintesis. Skema transformasi wavelet dapat dilihat di Gambar 2.2.



Gambar 2.2: Skema Transformasi Wavelet (a) Proses Analisis (b) Proses Sintesis

Misalkan diberikan sinyal input x_0, sinyal ini didekomposisikan dengan menggunakan operasi analisis sehingga didapat sinyal hampiran yang dinotasikan dengan x_1 dan sinyal detail yang dinotasikan dengan y_1. Selanjutnya sinyal x_1 juga didekomposisikan dengan menggunakan operasi

analisis sehingga didapat sinyal hampiran x_2 dan sinyal detail y_2. Transformasi ini dilakukan terus menerus sampai dekomposisi ke j yang hasilnya adalah sinyal hampiran x_j dan sinyal detail y_j.

Boggess (2001) dalam bukunya menyebutkan bahwa ada dua fungsi utama dalam transformasi wavelet, yaitu fungsi skala ϕ (*father wavelet*) dan fungsi induk wavelet ψ (*mother wavelet*). Transformasi wavelet yang paling sederhana adalah transformasi wavelet Haar. Fungsi skala pada transformasi wavelet Haar didefinisian dengan

$$\phi = \begin{cases} 1, & \text{untuk } 0 \leq x \leq 1 \\ 0, & \text{untuk } x \text{ yang lain.} \end{cases}$$

Sedangkan fungsi induk wavelet Haar didefinisikan dengan

$$\psi(x) = \phi(2x) - \phi(2x-1).$$

2.4 Transformasi Max Plus Wavelet

Berdasarkan transformasi wavelet Haar, Fahim (2014) mengkonstruksi transformasi max plus wavelet. Transformasi max plus wavelet ini menggunakan integer dan tidak menghasilkan *floating point* sehingga mempunyai beberapa kelebihan, yaitu lebih sederhana, *running time* lebih cepat, penggunaan memori lebih kecil dan tidak menghasilkan error penghitungan. Sebelum melakukan konstruksi transformasi max plus wavelet, terlebih dahulu diberikan Lemma dan Proposisi sebagai berikut.

Lemma 1: (Fahim, 2014) Misal $a, b, a_1, a_2, ..., a_p \in \mathbb{Z}$ dan $p \in \mathbb{N}$ maka

$$(a \oplus b) \oslash [(a \oslash b) \oplus 0] = b \tag{2.1}$$

$$(\bigoplus_{i=1}^{p} a_i \oplus b) \oslash [\bigoplus_{i=1}^{p} (a_i \oslash b) \oplus 0] = b \tag{2.2}$$

dengan $a \oslash b = a \otimes -b$.

Proposisi 1: (Fahim, 2014) Untuk sembarang $c_1, c_2, ..., c_p \in \mathbb{Z}$ dan $p \in \mathbb{N}$ dapat ditentukan $a_1, a_2, ..., a_p \in \mathbb{Z}$ sedemikian hingga

$$c_1 = \bigoplus_{i=1}^{p} a_i \tag{2.3}$$

$$c_i = a_i \oslash a_1, \text{ dengan } i = 2, 3, ..., p. \tag{2.4}$$

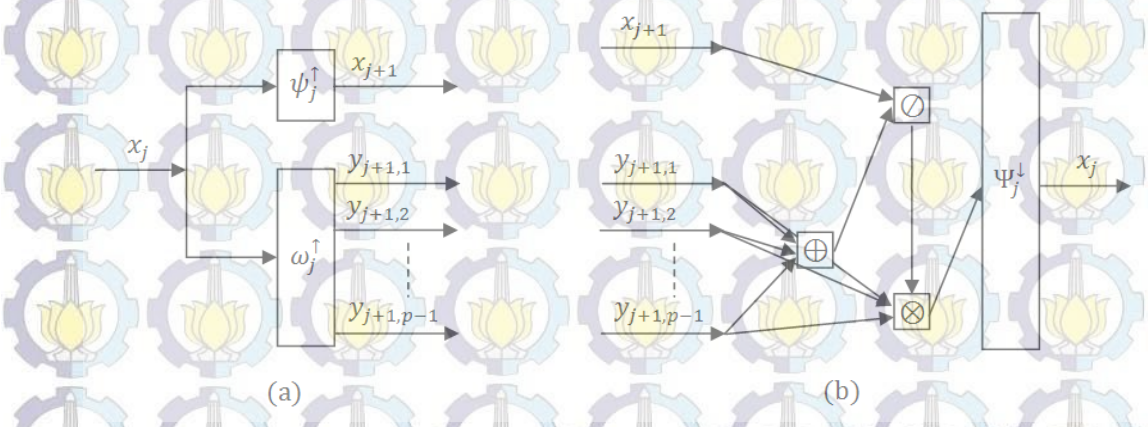
Selanjutnya dengan menggunakan sifat-sifat tersebut dikonstruksi transformasi max plus wavelet (MP-Wavelet). MP-Wavelet yang dikonstruksi terdiri dari lima tipe yaitu tipe A, tipe B, tipe C, tipe D dan tipe E. Pada MP-Wavelet terdapat proses analisis dan proses sintesis yang berbeda-beda untuk setiap tipe. Masing-masing tipe MP-Wavelet juga mempunyai aturan besar kanal yang berbeda-beda.

Pada algoritma kriptografi MP-Wavelet, proses analisis digunakan untuk proses enkripsi, sedangkan proses sintesis digunakan untuk proses dekripsi. Kunci bagian pertama dari algoritma kriptografi MP-Wavelet adalah tipe MP-Wavelet yang digunakan. Kunci bagian kedua adalah banyaknya kanal yang digunakan. Dan kunci bagian ketiga adalah kode sinyal detail.

Proses analisis dan sintesis dari masing-masing tipe transformasi MP-Wavelet secara lengkap dijelaskan pada sub bab berikut.

2.4.1 Transformasi MP-Wavelet Tipe A

Pada transformasi max plus wavelet tipe A ini terdapat operator analisis ($\psi_j^{\uparrow}$ dan $\omega_j^{\uparrow}$) dan operator sintesis $\Psi_j^{\downarrow}$ dengan pemetaannya adalah $\psi_j^{\uparrow} : V_j \to V_{j+1}$, $\omega_j^{\uparrow} : V_j \to W_{j+1}$ dan $\Psi_j^{\downarrow} : V_{j+1} \times W_{j+1} \to V_j$ dengan V_j merupakan ruang sinyal $\mathbb{Z}_j$ ke $\mathbb{Z}_j$ dan W_{j+1} merupakan ruang sinyal $\mathbb{Z}_{j+1}$ ke $\mathbb{Z}_{j+1}^{p-1}$ dan j adalah bilangan bulat tak negatif. Skema transformasi max plus wavelet tipe A dapat dilihat pada Gambar 2.3 (Fahim, 2014).



Gambar 2.3: Skema Transformasi Max Plus Wavelet Tipe A dengan p Kanal (a) Proses Analisis (b) Proses Sintesis

Pada proses analisis terlihat bahwa output terdiri dari satu sinyal hampiran dan $p-1$ sinyal detail. Berdasarkan hal tersebut disusun operator

analisis sebagai berikut:

$$\begin{aligned} \psi_j^{\uparrow}(x_j)[n] &= \bigoplus_{k=0}^{p-1} x_j[pn+k] = x_{j+1}[n] && (2.5)\\ \omega_j^{\uparrow}(x_j)[n] &= y_{j+1}[n] \\ &= (y_{j+1,1}[n], y_{j+1,2}[n], \ldots, y_{j+1,p-1}[n]) \\ &= (\omega_{j,1}^{\uparrow}(x_j)[n], \omega_{j,2}^{\uparrow}(x_j)[n], \ldots, \omega_{j,p-1}^{\uparrow}(x_j)[n]) && (2.6) \end{aligned}$$

dengan

$$\omega_{j,r}^{\uparrow}(x_j)[n] = x_j[pn+r] \oslash x_j[pn] = y_{j+1,r}[n].$$

Proses analisis ini digunakan untuk proses enkripsi pada algoritma kriptografi MP-Wavelet. Kunci kriptografi untuk MP-Wavelet tipe A adalah 1. Sedangkan kunci kanal adalah bilangan bulat p dimana $p > 1$. Hasil dari proses analisis dalam kriptografi ini adalah *ciphertext* dan kode sinyal detail. Kode sinyal detail diberikan dengan aturan sinyal detail yang bernilai negatif diberi kode 1 dan yang bernilai positif diberi kode 0.

Pada proses sintesis terlihat bahwa input terdiri dari satu sinyal hampiran dan $p-1$ sinyal detail. Berdasarkan hal tersebut disusun operator sintesis sebagai berikut:

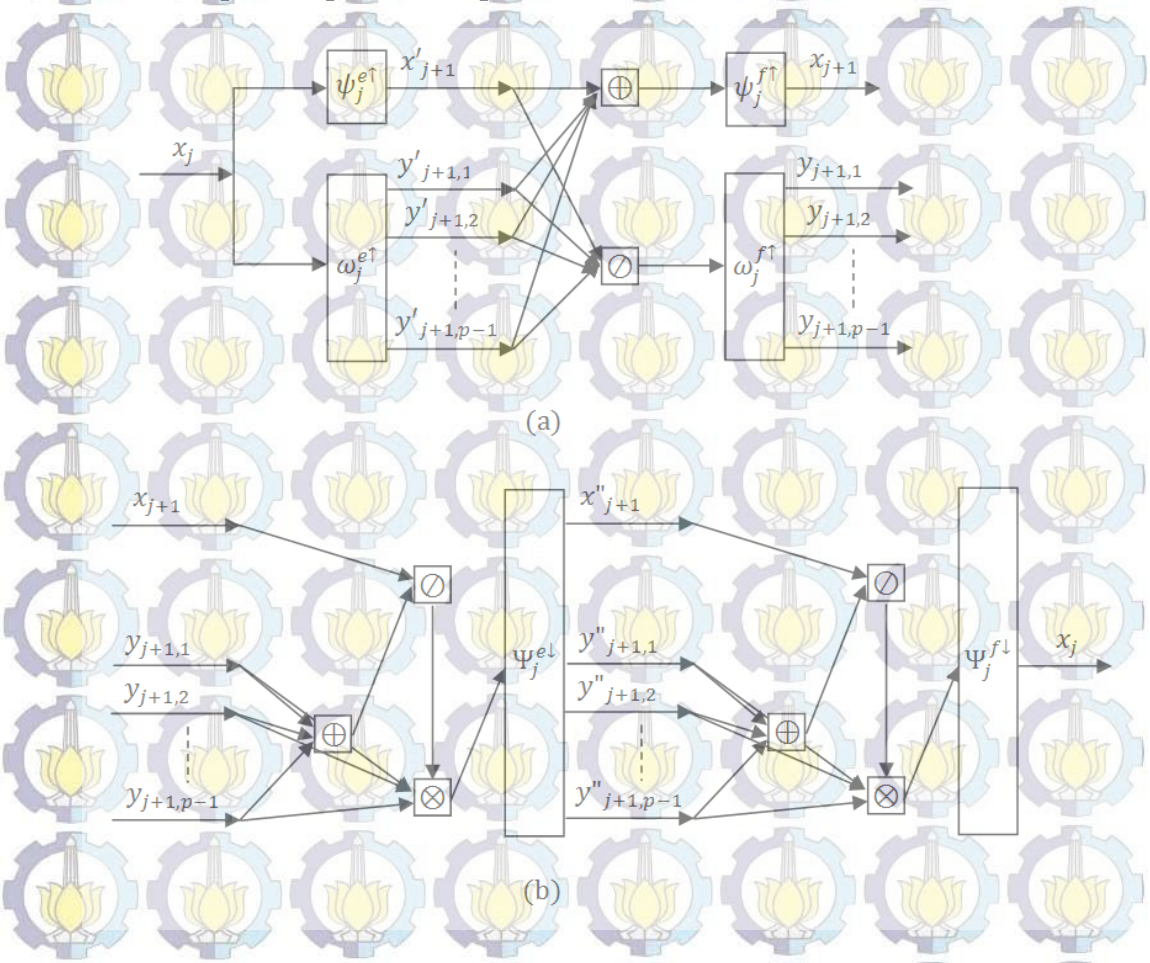
$$\begin{aligned} \Psi_j^{\downarrow}(x_{j+1}, y_{j+1})[pn] &= x_{j+1}[n] \oslash [(\bigoplus_{k=1}^{p-1} y_{j+1,k}[n]) \oplus 0] && (2.7)\\ \Psi_j^{\downarrow}(x_{j+1}, y_{j+1})[pn+r] &= \Psi_j^{\downarrow}(x_{j+1}, y_{j+1})[pn] \otimes y_{j+1,r}[n], && (2.8)\\ & \quad \text{dengan } r = 1, 2, \ldots, p-1. \end{aligned}$$

2.4.2 Transformasi MP-Wavelet Tipe B

MP-Wavelet Tipe B ini merupakan modifikasi dari MP-Wavelet Tipe A pada proses peningkatan dan penurunan level sinyal. Pada MP-Wavelet Tipe A, peningkatan level dari j ke level $j+1$ dilakukan hanya dengan satu kali proses yaitu $\psi_j^{\uparrow}$ dan $\omega_j^{\uparrow}$. Penurunan level dari $j+1$ ke level j juga dilakukan hanya dengan satu kali proses yaitu $\Psi_j^{\downarrow}$. Sedangkan pada MP-Wavelet Tipe B ini, peningkatan level dari j ke level $j+1$ dan penurunan level dari $j+1$ ke level j dilakukan dalam dua tahap.

Peningkatan level dari j ke level $j+1$ pada tahap pertama dilakukan dengan operator $\psi_j^{e\uparrow}$ dan $\omega_j^{e\uparrow}$, dan pada tahap kedua dilakukan dengan

operator $\psi_j^{f\uparrow}$ dan $\omega_j^{f\uparrow}$. Sedangkan penurunan level dari $j+1$ ke level j pada tahap pertama dilakukan dengan operator $\Psi_j^{e\downarrow}$, dan pada tahap kedua dilakukan dengan operator $\Psi_j^{f\downarrow}$. Dengan pemetaannya adalah $\psi_j^{e\uparrow} : V_j \to V_{j+1}$, $\psi_j^{f\uparrow} : V_{j+1} \times W_{j+1} \to V_{j+1}$, $\omega_j^{e\uparrow} : V_j \to W_{j+1}$, $\omega_j^{f\uparrow} : V_{j+1} \times W_{j+1} \to W_{j+1}$, $\Psi_j^{e\downarrow} : V_{j+1} \times W_{j+1} \to V_{j+1} \times W_{j+1}$ dan $\Psi_j^{f\downarrow} : V_{j+1} \times W_{j+1} \to V_j$ dengan V_j merupakan ruang sinyal $\mathbb{Z}_j$ ke $\mathbb{Z}_j$ dan W_{j+1} merupakan ruang sinyal $\mathbb{Z}_{j+1}$ ke $\mathbb{Z}_{j+1}^{p-1}$ dan j merupakan bilangan bulat tak negatif. Skema transformasi max plus wavelet tipe B dapat dilihat pada Gambar 2.4 (Fahim, 2014).



Gambar 2.4: Skema Transformasi Max Plus Wavelet Tipe B dengan p Kanal (a) Proses Analisis (b) Proses Sintesis

Pada proses analisis terlihat bahwa input dari $\psi_j^{e\uparrow}$ merupakan sinyal x_j dan outputnya adalah x'_{j+1}. Input dari $\omega_j^{e\uparrow}$ merupakan sinyal x_j dan outputnya adalah $y'_{j+1,1}, y'_{j+1,2}, \ldots, y'_{j+1,p-1}$. Berdasarkan hal tersebut disusun operator analisis tahap pertama sebagai berikut:

$$\psi_j^{e\uparrow}(x_j)[n] = \bigoplus_{k=0}^{p-1} x_j[pn+k] = x'_{j+1}[n] \tag{2.9}$$

$$\begin{aligned}\omega_j^{e\uparrow}(x_j)[n] &= y'_{j+1}[n] \\ &= (y'_{j+1,1}[n], y'_{j+1,2}[n], ..., y'_{j+1,p-1}[n]) \\ &= (\omega_{j,1}^{e\uparrow}(x_j)[n], \omega_{j,2}^{e\uparrow}(x_j)[n], ..., \omega_{j,p-1}^{e\uparrow}(x_j)[n]) \qquad (2.10)\end{aligned}$$

dengan

$$\omega_{j,r}^{e\uparrow}(x_j)[n] = x_j[pn+r] \oslash x_j[pn] = y'_{j+1,r}[n].$$

Input dari $\psi_j^{f\uparrow}$ merupakan sinyal x'_{j+1} dan $y'_{j+1,1}, y'_{j+1,2}, ..., y'_{j+1,p-1}$, sedangkan outputnya adalah x_{j+1} yang merupakan sinyal hampiran. Input dari $\omega_j^{f\uparrow}$ merupakan sinyal x'_{j+1} dan $y'_{j+1,1}, y'_{j+1,2}, ..., y'_{j+1,p-1}$ dan outputnya adalah $y_{j+1,1}, y_{j+1,2}, ..., y_{j+1,p-1}$ yang merupakan sinyal detail. Berdasarkan hal tersebut disusun operator analisis tahap kedua sebagai berikut:

$$\psi_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] = x'_{j+1}[n] \oplus (\bigoplus_{k=1}^{p-1} y'_{j+1,k}[n]) = x_{j+1}[n] \qquad (2.11)$$

$$\begin{aligned}\omega_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] &= y_{j+1}[n] \\ &= (y_{j+1,1}[n], y_{j+1,2}[n], ..., y_{j+1,p-1}[n]) \\ &= (\omega_{j,1}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n], \omega_{j,2}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n], ..., \\ &\quad \omega_{j,p-1}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n]) \qquad (2.12)\end{aligned}$$

dengan

$$\omega_{j,r}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] = y'_{j+1,r}[n] \oslash x'_{j+1}[n] = y_{j+1}[n]$$

dan $r = 1, 2, ..., p-1$.

Proses analisis ini digunakan untuk proses enkripsi pada algoritma kriptografi MP-Wavelet. Kunci kriptografi untuk MP-Wavelet tipe B adalah 2. Sedangkan kunci kanal adalah bilangan bulat p dimana $p > 1$. Hasil dari proses analisis dalam kriptografi ini adalah *ciphertext* dan kode sinyal detail. Jangkauan output dari proses analisis MP-Wavelet tipe B berbeda dengan tipe yang lain. Sehingga pemberian kode sinyal detail juga berbeda dengan tipe yang lain. Kode sinyal detail diberikan dengan aturan sinyal detail yang bernilai kurang dari -126 diberi kode 1 dan yang bernilai lebih dari atau sama dengan -126 diberi kode 0.

Pada proses sintesis terlihat bahwa input dari $\Psi_j^{e\downarrow}$ terdiri dari dua bagian yaitu sinyal hampiran x_{j+1} dan sinyal detail $y_{j+1,1}, y_{j+1,2}, ..., y_{j+1,p-1}$. Sedangkan outputnya adalah x''_{j+1} dan $y''_{j+1,1}, y''_{j+1,2}, ..., y''_{j+1,p-1}$. Berdasarkan

hal tersebut disusun operator sintesis tahap pertama sebagai berikut:

$$\begin{aligned}\Psi_j^{e\downarrow}(x_{j+1},y_{j+1})[n] &= (\overset{"}{x}_{j+1}[n],\overset{"}{y}_{j+1}[n]) \\ &= (\Psi_{j,1}^{e\downarrow}(x_{j+1},y_{j+1})[n],(\Psi_{j,2}^{e\downarrow}(x_{j+1},y_{j+1})[n], \\ &\quad \Psi_{j,3}^{e\downarrow}(x_{j+1},y_{j+1})[n],...,\Psi_{j,p}^{e\downarrow}(x_{j+1},y_{j+1})[n] \quad (2.13)\end{aligned}$$

dengan

$$\begin{aligned}\Psi_{j,1}^{e\downarrow}(x_{j+1},y_{j+1})[n] &= (x_{j+1}[n] \oslash (\bigoplus_{k=1}^{p-1} y_{j+1,k}[n] \oplus 0) \\ \Psi_{j,r}^{e\downarrow}(x_{j+1},y_{j+1})[n] &= \Psi_{j,1}^{e\downarrow}(x_{j+1},y_{j+1})[n] \otimes y_{j+1,r-1}[n]\end{aligned}$$

dan $r = 2,3,...,p$.

Input dari $\Psi_j^{f\downarrow}$ terdiri dari dua bagian yaitu sinyal $\overset{"}{x}_{j+1}$ dan sinyal $\overset{"}{y}_{j+1,1},\overset{"}{y}_{j+1,2},...,\overset{"}{y}_{j+1,p-1}$. Sedangkan outputnya adalah x_j. Berdasarkan hal tersebut disusun operator sintesis tahap kedua sebagai berikut:

$$\Psi_j^{f\downarrow}(\overset{"}{x}_{j+1},\overset{"}{y}_{j+1})[pn] = \overset{"}{x}_{j+1}[n] \oslash [(\bigoplus_{k=1}^{p-1} \overset{"}{y}_{j+1,k}[n]) \oplus 0] \quad (2.14)$$

$$\Psi_j^{f\downarrow}(\overset{"}{x}_{j+1},\overset{"}{y}_{j+1})[pn+r] = \Psi_j^{f\downarrow}(\overset{"}{x}_{j+1},\overset{"}{y}_{j+1})[pn] \otimes \overset{"}{y}_{j+1,r}[n] \quad (2.15)$$

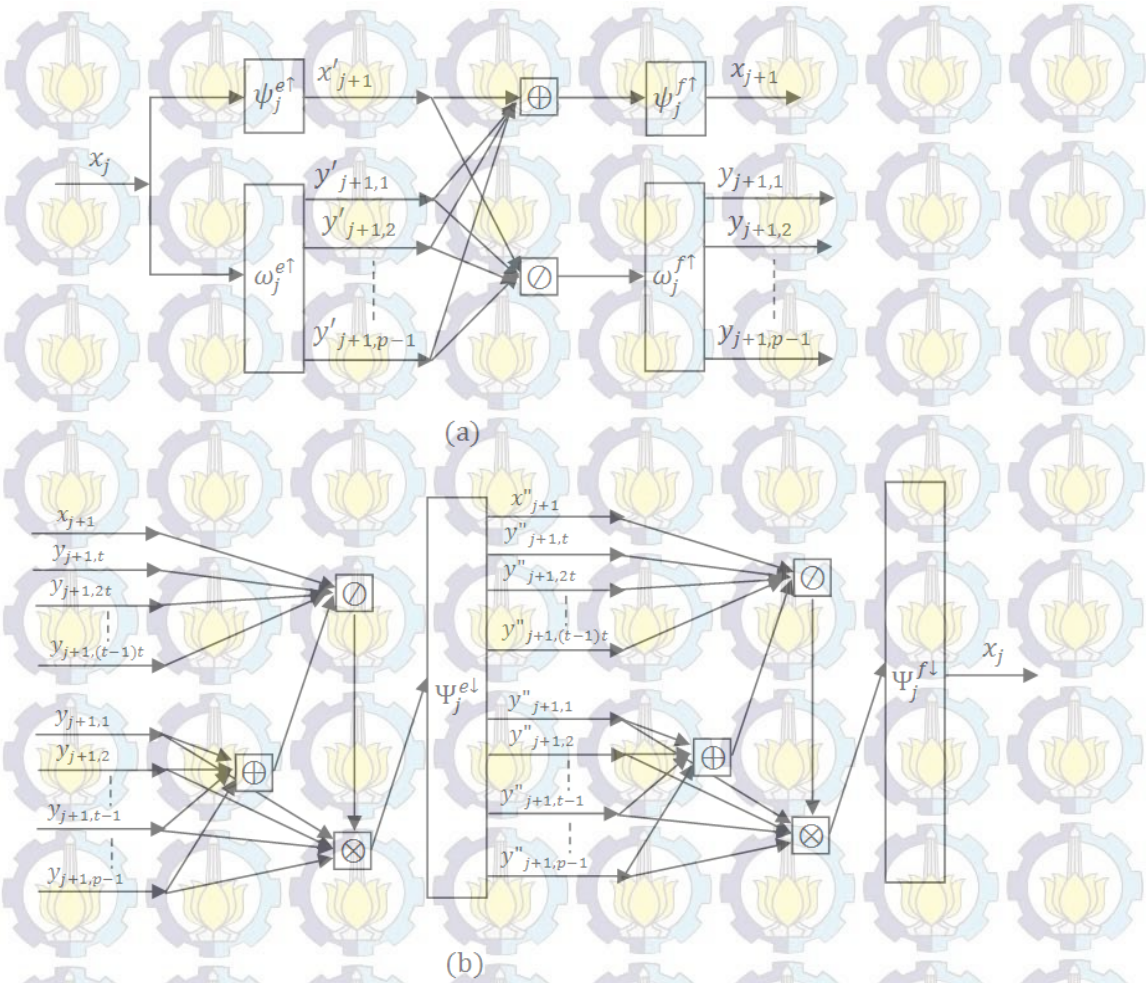
dengan $r = 1,2,...,p-1$.

2.4.3 Transformasi MP-Wavelet Tipe C

MP-Wavelet Tipe C ini merupakan modifikasi dari MP-Wavelet Tipe B. Perbedaannya terletak pada banyaknya kanal. Pada MP-Wavelet Tipe B kanalnya sebanyak sembarang bilangan p, sedangkan pada MP-Wavelet Tipe C kanalnya sebanyak p dengan p = t × t.

Pada transformasi max plus wavelet tipe C ini terdapat operator analisis tahap pertama yaitu $\psi_j^{e\uparrow}$ dan $\omega_j^{e\uparrow}$, operator tahap kedua yaitu $\psi_j^{f\uparrow}$ dan $\omega_j^{f\uparrow}$, serta operator sintesis yaitu $\Psi_j^{e\downarrow}$ dan $\Psi_j^{f\downarrow}$, dengan pemetaan yang sama seperti pada transformasi max plus wavelet tipe B. Skema transformasi max plus wavelet tipe C dapat dilihat pada Gambar 2.5 (Fahim, 2014).

Dari Gambar 2.5 terlihat bahwa pada proses analisis, input dari $\psi_j^{e\uparrow}$ dan $\omega_j^{e\uparrow}$ adalah sinyal utama x_j. Sedangkan output dari $\psi_j^{e\uparrow}$ merupakan sinyal x'_{j+1} dan output dari $\omega_j^{e\uparrow}$ merupakan sinyal $y'_{j+1,1},y'_{j+1,2},...,y'_{j+1,p-1}$.



Gambar 2.5: Skema Transformasi Max Plus Wavelet Tipe C dengan p Kanal (a) Proses Analisis (b) Proses Sintesis

Berdasarkan hal tersebut disusun operator analisis tahap pertama dari transformasi max plus wavelet tipe C sebagai berikut:

$$\begin{aligned} \psi_j^{e\uparrow}(x_j)[n] &= x'_{j+1}[n] = \tau_0[n] \qquad (2.16)\\ \omega_j^{e\uparrow}(x_j)[n] &= y'_{j+1}[n] \\ &= (y'_{j+1,1}[n], y'_{j+1,2}[n], \ldots, y'_{j+1,p-1}[n]) \\ &= (\tau_1[n], \tau_2[n], \ldots, \tau_{p-1}[n]) \qquad (2.17) \end{aligned}$$

dengan

$$\begin{aligned} \tau_{at}[n] &= \bigoplus_{k=0}^{t-1} x_j[pn+at+k], \quad a = 0,1,\ldots,t-1 \\ \tau_{at+b}[n] &= x_j[pn+at+b] \oslash x_j[pn+at], \quad b = 1,2,\ldots,t-1. \end{aligned}$$

Sedangkan operator analisis tahap kedua disusun sebagai berikut:

$$\psi_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] = x_{j+1}[n] = \rho_0[n] \quad (2.18)$$

$$\omega_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] = y_{j+1}[n] = (y_{j+1,1}[n], y_{j+1,2}[n], ..., y_{j+1,p-1}[n]) = (\rho_1[n], \rho_2[n], ..., \rho_{p-1}[n]) \quad (2.19)$$

dengan

$$\rho_{at}[n] = \bigoplus_{k=0}^{t-1} \tau_{kt+a}[n], \quad a = 0, 1, ..., t-1$$

$$\rho_{at+b}[n] = \tau_{bt+a}[n] \oslash \tau_a[n], \quad b = 1, 2, ..., t-1.$$

Proses analisis ini digunakan untuk proses enkripsi pada algoritma kriptografi MP-Wavelet. Kunci kriptografi untuk MP-Wavelet tipe C adalah 3. Sedangkan kunci kanal adalah bilangan bulat p dimana $p = t \times t$ atau bilangan kuadrat. Hasil dari proses analisis dalam kriptografi ini adalah *ciphertext* dan kode sinyal detail. Kode sinyal detail diberikan dengan aturan sinyal detail yang bernilai negatif diberi kode 1 dan yang bernilai positif diberi kode 0.

Pada proses sintesis terlihat bahwa input dari $\Psi_j^{e\downarrow}$ terdiri dari dua bagian yaitu sinyal hampiran x_{j+1} dan sinyal detail $y_{j+1,1}, y_{j+1,2}, ..., y_{j+1,p-1}$. Sedangkan outputnya adalah x''_{j+1} dan $y''_{j+1,1}, y''_{j+1,2}, ..., y''_{j+1,p-1}$. Berdasarkan hal tersebut disusun operator sintesis tahap pertama sebagai berikut:

$$\Psi_j^{e\downarrow}(x_{j+1}, y_{j+1})[n] = (x''_{j+1}[n], y''_{j+1}[n]) = (\tau'_0[n], (\tau'_1[n], \tau'_2[n], ..., \tau'_{p-1}[n])) \quad (2.20)$$

dengan

$$\tau'_a[n] = \rho_{at}[n] \oslash (\bigoplus_{k=1}^{t-1} \rho_{at+k}[n] \oplus 0), \quad a = 0, 1, ..., t-1$$

dan

$$\tau'_{a+bt}[n] = \tau'_a[n] \otimes \rho_{at+b}[n], \quad b = 1, 2, ..., t-1.$$

Input dari $\Psi_j^{f\downarrow}$ terdiri dari dua bagian yaitu sinyal x''_{j+1} dan sinyal $y''_{j+1,1}, y''_{j+1,2}, ..., y''_{j+1,p-1}$. Sedangkan outputnya adalah x_j. Berdasarkan hal

tersebut disusun operator sintesis tahap kedua sebagai berikut:

$$\Psi_j^{f\downarrow}(x^{"}_{j+1}, y^{"}_{j+1})[pn+at] = \tau^{'}_{at}[n] \oslash (\bigoplus_{k=1}^{t-1} \tau^{'}_{at+k}[n] \oplus 0) \quad (2.21)$$

$$\Psi_j^{f\downarrow}(x^{"}_{j+1}, y^{"}_{j+1})[pn+at+r] = \Psi_j^{f\downarrow}(x^{"}_{j+1}, y^{"}_{j+1})[pn+at] \otimes \tau^{'}_{at+r}[n], \quad (2.22)$$

$$\text{dengan } r = 1, 2, ..., t-1.$$

2.4.4 Transformasi MP-Wavelet Tipe D

MP-Wavelet Tipe D ini merupakan penyempurnaan dari tipe C. Pada tipe C banyaknya kanal adalah bilangan kuadrat, sedangkan pada tipe D banyaknya kanal adalah bilangan bulat positif p dengan p = s $\times$ t. Pada MP-Wavelet tipe D ini terdapat operator analisis yaitu $\psi_j^{e\uparrow}, \psi_j^{f\uparrow}, \omega_j^{e\uparrow}, \omega_j^{f\uparrow}$, serta operator sintesis yaitu $\Psi_j^{e\downarrow}$ dan $\Psi_j^{f\downarrow}$, dengan pemetaan yang sama seperti pada MP-Wavelet tipe B. Skema transformasi max plus wavelet tipe D dapat dilihat pada Gambar 2.6 (Fahim, 2014).

Pada proses analisis terlihat bahwa input dari $\psi_j^{e\uparrow}$ merupakan sinyal x_j dan outputnya adalah $x^{'}_{j+1}$. Input dari $\omega_j^{e\uparrow}$ merupakan sinyal x_j dan outputnya adalah $y^{'}_{j+1,1}, y^{'}_{j+1,2}, ..., y^{'}_{j+1,p-1}$. Berdasarkan hal tersebut disusun operator analisis tahap pertama sebagai berikut:

$$\psi_j^{e\uparrow}(x_j)[n] = x^{'}_{j+1}[n] = \tau_0[n] \quad (2.23)$$

$$\begin{aligned} \omega_j^{e\uparrow}(x_j)[n] &= y^{'}_{j+1}[n] \\ &= (y^{'}_{j+1,1}[n], y^{'}_{j+1,2}[n], ..., y^{'}_{j+1,p-1}[n]) \\ &= (\tau_1[n], \tau_2[n], ..., \tau_{p-1}[n]) \end{aligned} \quad (2.24)$$

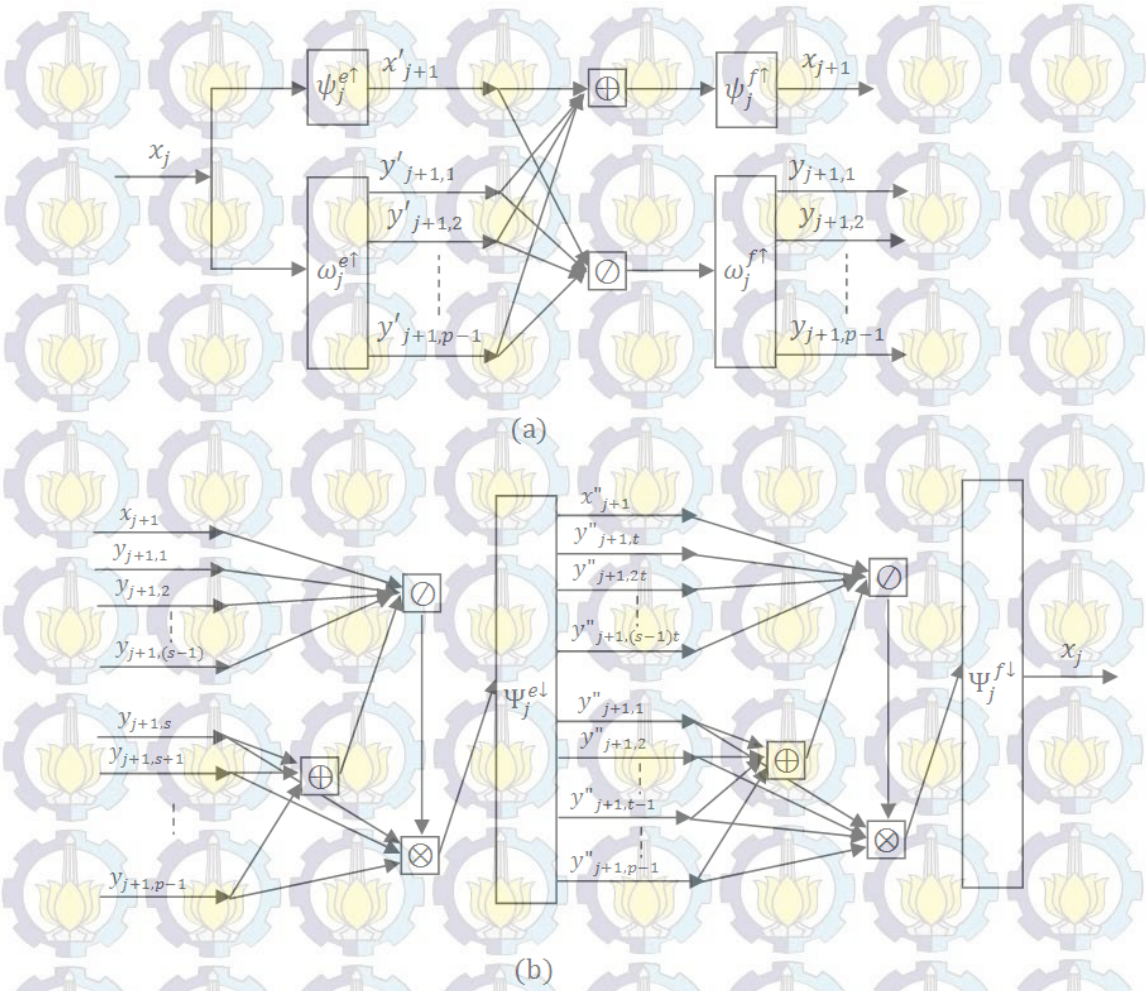
dengan

$$\tau_{at}[n] = \bigoplus_{k=0}^{t-1} x_j[pn+at+k], \quad a = 0, 1, ..., s-1$$

dan

$$\tau_{at+b}[n] = x_j[pn+at+b] \oslash x_j[pn+at], \quad b = 1, 2, ..., t-1.$$

Input dari $\psi_j^{f\uparrow}$ merupakan sinyal $x^{'}_{j+1}$ dan $y^{'}_{j+1,1}, y^{'}_{j+1,2}, ..., y^{'}_{j+1,p-1}$, sedangkan outputnya adalah x_{j+1} yang merupakan sinyal hampiran. Input dari $\omega_j^{f\uparrow}$ merupakan sinyal $x^{'}_{j+1}$ dan $y^{'}_{j+1,1}, y^{'}_{j+1,2}, ..., y^{'}_{j+1,p-1}$ dan outputnya



Gambar 2.6: Skema Transformasi Max Plus Wavelet Tipe D dengan p Kanal (a) Proses Analisis (b) Proses Sintesis

adalah $y_{j+1,1}, y_{j+1,2}, \ldots, y_{j+1,p-1}$ yang merupakan sinyal detail. Berdasarkan hal tersebut disusun operator analisis tahap kedua pada transformasi max plus wavelet tipe D sebagai berikut:

$$\begin{aligned} \psi_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] &= x_{j+1}[n] = \rho_0[n] \qquad (2.25) \\ \omega_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] &= y_{j+1}[n] \\ &= (y_{j+1,1}[n], y_{j+1,2}[n], \ldots, y_{j+1,p-1}[n]) \\ &= (\rho_1[n], \rho_2[n], \ldots, \rho_{p-1}[n]) \qquad (2.26) \end{aligned}$$

dengan

$$\rho_a[n] = \bigoplus_{k=0}^{s-1} \tau_{kt+a}[n], \quad a = 0, 1, \ldots, t-1$$

dan

$$\rho_{a+bt}[n] \;=\; \tau_{bt+a}[n] \oslash \tau_a[n], \quad b = 1,2,...,s-1.$$

Proses analisis ini digunakan untuk proses enkripsi pada algoritma kriptografi MP-Wavelet. Kunci kriptografi untuk MP-Wavelet tipe D adalah 4. Sedangkan kunci kanal adalah pasangan bilangan bulat s dan t dimana $s > 1$ dan $t > 1$. Karena itu pada algoritma kriptografi MP-Wavelet tipe D banyaknya kunci kanal harus genap. Hasil dari proses analisis dalam kriptografi ini adalah *ciphertext* dan kode sinyal detail. Kode sinyal detail diberikan dengan aturan sinyal detail yang bernilai negatif diberi kode 1 dan yang bernilai positif diberi kode 0.

Pada proses sintesis terlihat bahwa input dari $\Psi_j^{e\downarrow}$ terdiri dari dua bagian yaitu sinyal hampiran x_{j+1} dan sinyal detail $y_{j+1,1}, y_{j+1,2}, ..., y_{j+1,p-1}$. Sedangkan outputnya adalah $x^{"}_{j+1}$ dan $y^{"}_{j+1,1}, y^{"}_{j+1,2}, ..., y^{"}_{j+1,p-1}$. Berdasarkan hal tersebut disusun operator sintesis tahap pertama pada MP-Wavelet tipe D sebagai berikut:

$$\begin{aligned} \Psi_j^{e\downarrow}(x_{j+1}, y_{j+1})[n] &= (x^{"}_{j+1}[n], y^{"}_{j+1}[n]) \\ &= (\tau^{'}_0[n], (\tau^{'}_1[n], \tau^{'}_2[n], ..., \tau^{'}_{p-1}[n])) \end{aligned} \tag{2.27}$$

dengan

$$\tau^{'}_a[n] \;=\; \rho_a[n] \oslash (\bigoplus_{k=1}^{s-1} \rho_{kt+a}[n] \oplus 0), \quad a = 0,1,...,t-1$$

dan

$$\tau^{'}_{a+bt}[n] \;=\; \tau^{'}_a[n] \otimes \rho_{a+bt}[n], \quad b = 1,2,...,s-1.$$

Input dari $\Psi_j^{f\downarrow}$ terdiri dari dua bagian yaitu sinyal $x^{"}_{j+1}$ dan sinyal $y^{"}_{j+1,1}, y^{"}_{j+1,2}, ..., y^{"}_{j+1,p-1}$. Sedangkan outputnya adalah x_j. Berdasarkan hal tersebut disusun operator sintesis tahap kedua sebagai berikut:

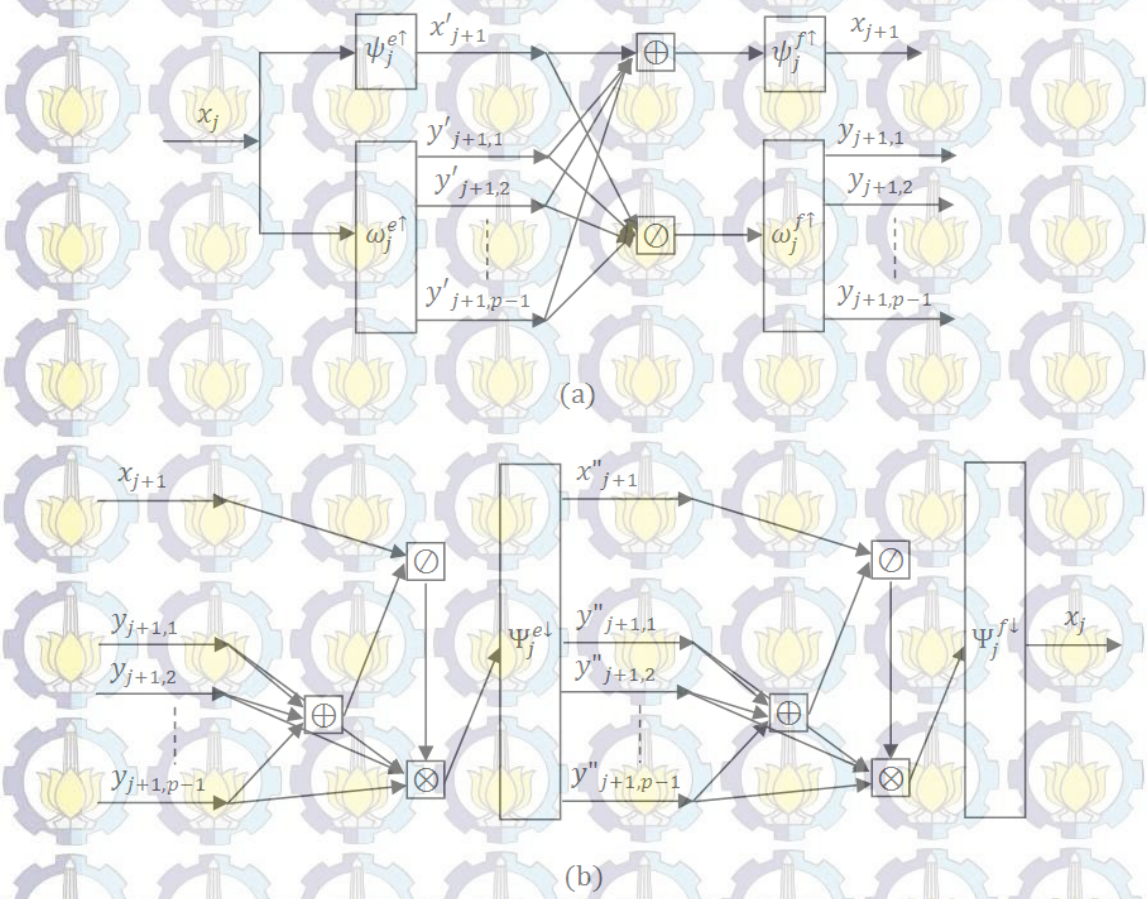
$$\Psi_j^{f\downarrow}(x^{"}_{j+1}, y^{"}_{j+1})[pn+at] \;=\; \tau^{'}_{at}[n] \oslash (\bigoplus_{k=1}^{t-1} \tau^{'}_{at+k}[n] \oplus 0) \tag{2.28}$$

$$\Psi_j^{f\downarrow}(x^{"}_{j+1}, y^{"}_{j+1})[pn+at+r] \;=\; \Psi_j^{f\downarrow}(x^{"}_{j+1}, y^{"}_{j+1})[pn+at] \otimes \tau^{'}_{at+r}[n], \tag{2.29}$$

dimana $r = 1,2,...,t-1$.

2.4.5 Transformasi MP-Wavelet Tipe E

MP-Wavelet Tipe E ini merupakan modifikasi dari MP-Wavelet Tipe B. Perbedaannya terdapat pada operator $\psi_j^{f\uparrow}$, $\omega_j^{f\uparrow}$ dan $\Psi_j^{f\downarrow}$. Pada transformasi max plus wavelet tipe E ini terdapat operator analisis tahap pertama yaitu $\psi_j^{e\uparrow}$ dan $\omega_j^{e\uparrow}$, operator analisis tahap kedua yaitu $\psi_j^{f\uparrow}$ dan $\omega_j^{f\uparrow}$, serta operator sintesis tahap pertama $\Psi_j^{e\downarrow}$ dan operator sintesis tahap kedua $\Psi_j^{f\downarrow}$, dengan pemetaan yang sama seperti pada transformasi max plus wavelet tipe B. Skema transformasi max plus wavelet tipe E dapat dilihat pada Gambar 2.7 (Fahim, 2014).



Gambar 2.7: Skema Transformasi Max Plus Wavelet Tipe E dengan p Kanal (a) Proses Analisis (b) Proses Sintesis

Pada proses analisis terlihat bahwa input dari $\psi_j^{e\uparrow}$ adalah sinyal utama x_j dan outputnya adalah x'_{j+1}. Input dari $\omega_j^{e\uparrow}$ adalah sinyal utama x_j dan outputnya adalah $y'_{j+1,1}, y'_{j+1,2}, \ldots, y'_{j+1,p-1}$. Berdasarkan hal tersebut disusun operator analisis tahap pertama pada transformasi max plus

wavelet tipe E sebagai berikut:

$$\begin{aligned}
\psi_j^{e\uparrow}(x_j)[n] &= \bigoplus_{i=0}^{p-1} x_j[pn+k] = x'_{j+1}[n] && (2.30)\\
\omega_j^{e\uparrow}(x_j)[n] &= y'_{j+1}[n] \\
&= (y'_{j+1,1}[n], y'_{j+1,2}[n], ..., y'_{j+1,p-1}[n]) \\
&= (\omega_{j,1}^{e\uparrow}(x_j)[n], \omega_{j,2}^{e\uparrow}(x_j)[n], ..., \omega_{j,p-1}^{e\uparrow}(x_j)[n]) && (2.31)
\end{aligned}$$

dengan

$$\omega_{j,r}^{e\uparrow}(x_j)[n] = x_j[pn+r] \oslash x_j[pn] = y'_{j+1,r}[n].$$

Input dari $\psi_j^{f\uparrow}$ merupakan sinyal x'_{j+1} dan $y'_{j+1,1}, y'_{j+1,2}, ..., y'_{j+1,p-1}$, sedangkan outputnya adalah x_{j+1} yang merupakan sinyal hampiran. Input dari $\omega_j^{f\uparrow}$ merupakan sinyal x'_{j+1} dan $y'_{j+1,1}, y'_{j+1,2}, ..., y'_{j+1,p-1}$ dan outputnya adalah $y_{j+1,1}, y_{j+1,2}, ..., y_{j+1,p-1}$ yang merupakan sinyal detail. Berdasarkan hal tersebut disusun operator analisis tahap kedua sebagai berikut:

$$\begin{aligned}
\psi_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] &= x'_{j+1}[n] = x_{j+1}[n] && (2.32)\\
\omega_j^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] &= y_{j+1}[n] \\
&= (y_{j+1,1}[n], y_{j+1,2}[n], ..., y_{j+1,p-1}[n]) \\
&= (\omega_{j,1}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n], \omega_{j,2}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n], ..., \\
&\quad \omega_{j,p-1}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n]) && (2.33)
\end{aligned}$$

dengan

$$\omega_{j,1}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] = \bigoplus_{r=1}^{p-1} y'_{j+1,r}[n] = y_{j+1,1}[n]$$

dan

$$\begin{aligned}
\omega_{j,r}^{f\uparrow}(x'_{j+1}, y'_{j+1})[n] &= y'_{j+1,r}[n] \oslash y'_{j+1,1}[n] = y_{j+1,r}[n], \\
&\quad \text{dimana } r = 2, 3, ..., p-1.
\end{aligned}$$

Proses analisis ini digunakan untuk proses enkripsi pada algoritma kriptografi MP-Wavelet. Kunci kriptografi untuk MP-Wavelet tipe E adalah 5. Sedangkan kunci kanal adalah bilangan bulat p dimana $p > 1$. Hasil dari

proses analisis dalam kriptografi ini adalah *ciphertext* dan kode sinyal detail. Kode sinyal detail diberikan dengan aturan sinyal detail yang bernilai negatif diberi kode 1 dan yang bernilai positif diberi kode 0.

Pada proses sintesis terlihat bahwa input dari $\Psi_j^{e\downarrow}$ terdiri dari dua bagian yaitu sinyal hampiran x_{j+1} dan sinyal detail $y_{j+1,1}, y_{j+1,2}, \ldots, y_{j+1,p-1}$. Sedangkan outputnya adalah $\overset{"}{x}_{j+1}$ dan $\overset{"}{y}_{j+1,1}, \overset{"}{y}_{j+1,2}, \ldots, \overset{"}{y}_{j+1,p-1}$. Berdasarkan hal tersebut disusun operator sintesis tahap pertama sebagai berikut:

$$\begin{aligned}\Psi_j^{e\downarrow}(x_{j+1}, y_{j+1})[n] &= (\overset{"}{x}_{j+1}[n], \overset{"}{y}_{j+1}[n]) \\ &= (x_{j+1}, (\Psi_{j,1}^{e\downarrow}(x_{j+1}, y_{j+1})[n], \Psi_{j,2}^{e\downarrow}(x_{j+1}, y_{j+1})[n], \ldots, \\ &\quad \Psi_{j,p-1}^{e\downarrow}(x_{j+1}, y_{j+1})[n])) \end{aligned} \tag{2.34}$$

dengan

$$\Psi_{j,1}^{e\downarrow}(x_{j+1}, y_{j+1})[n] = y_{j+1,1}[n] \oslash (\bigoplus_{k=2}^{p-1} y_{j+1,k}[n] \oplus 0)$$

dan

$$\begin{aligned}\Psi_{j,r}^{e\downarrow}(x_{j+1}, y_{j+1})[n] &= \Psi_{j,1}^{e\downarrow}(x_{j+1}, y_{j+1})[n] \otimes y_{j+1,r}[n], \\ &\quad \text{dimana } r = 2, 3, \ldots, p-1.\end{aligned}$$

Input dari $\Psi_j^{f\downarrow}$ terdiri dari dua bagian yaitu sinyal $\overset{"}{x}_{j+1}$ dan sinyal $\overset{"}{y}_{j+1,1}, \overset{"}{y}_{j+1,2}, \ldots, \overset{"}{y}_{j+1,p-1}$. Sedangkan outputnya adalah x_j. Berdasarkan hal tersebut disusun operator sintesis tahap kedua sebagai berikut:

$$\Psi_j^{f\downarrow}(\overset{"}{x}_{j+1}, \overset{"}{y}_{j+1})[pn] = \overset{"}{x}_{j+1}[n] \oslash [(\bigoplus_{k=1}^{p-1} \overset{"}{y}_{j+1,k}[n]) \oplus 0] \tag{2.35}$$

$$\begin{aligned}\Psi_j^{f\downarrow}(\overset{"}{x}_{j+1}, \overset{"}{y}_{j+1})[pn+r] &= \Psi_j^{f\downarrow}(\overset{"}{x}_{j+1}, \overset{"}{y}_{j+1})[pn] \otimes \overset{"}{y}_{j+1,r}[n], \\ &\quad \text{dimana } r = 2, 3, \ldots, p-1.\end{aligned} \tag{2.36}$$

Proses sintesis pada MP-Wavelet ini digunakan untuk proses dekripsi pada algoritma kriptografi MP-Wavelet. Kunci kriptografi bagian ketiga digunakan untuk mendapatkan sinyal detail yang kemudian dimasukkan kedalam proses sintesis. Hasil dari proses sintesis ini adalah pesan awal atau *plaintext*.

Kelima tipe MP-Wavelet ini digunakan sebagai kunci dalam algoritma

kriptografi MP-Wavelet. Hal ini dapat memperbesar *key space* sehingga semakin banyak kemungkinan kunci dan semakin sulit untuk menemukan kunci yang sebenarnya. Kunci kanal juga mempersulit upaya pemecahan kunci karena harus menemukan faktor-faktor dari suatu bilangan dan semua kemungkinan kanal yang digunakan.

2.5 Analisis Algoritma Kriptografi

Pada tesis ini analisis terhadap algoritma kriptografi yang disusun dilakukan dengan menghitung nilai korelasi antara *plaintext* dan *ciphertext*, kualitas enkripsi dan *running time* (Aruljothi, 2012). Selain itu juga akan dibahas kompleksitas waktu dan analisis *key space* dari algoritma yang disusun.

2.5.1 Korelasi *Plaintext* dan *Ciphertext*

Penghitungan nilai korelasi dilakukan untuk mengetahui hubungan linier antara *plaintext* dan *ciphertext*. Jika *plaintext* dan *ciphertext* cenderung mengikuti garis lurus dengan kemiringan yang sama, maka ada korelasi positif yang tinggi antara keduanya. Akan tetapi jika keduanya mengikuti garis lurus dengan arah kemiringan yang berlawanan, maka terdapat nilai korelasi yang negatif. Jika korelasi bernilai 1 atau -1 maka *ciphertext* mempunyai hubungan linier yang kuat dengan *plaintext*. Di dalam kriptografi hal ini merupakan enkripsi yang tidak baik. Jika korelasi bernilai 0 maka *plaintext* dan *ciphertext* tidak mempunyai hubungan linier. Hal ini menunjukkan bahwa algoritma tersebut mempunyai proses enkripsi yang baik.

Penghitungan nilai korelasi dilakukan dengan menggunakan rumus (Aruljothi, 2012):

$$r = \frac{n\sum(xy) - \sum x \sum y}{\sqrt{(n\sum(x^2) - (\sum x)^2)(n\sum(y^2) - (\sum y)^2)}} \tag{2.37}$$

dimana x adalah plaintext, y adalah ciphertext dan n adalah panjang plaintext. Interpretasi nilai korelasi menurut Sarwono (2006) disajikan dalam Tabel 2.1.

2.5.2 Kualitas Enkripsi

Penghitungan kualitas enkripsi dilakukan dengan membandingkan frekuensi kemunculan setiap karakter di *plaintext* dan *ciphertext*. Kualitas enkripsi merepresentasikan rata-rata selisih frekuensi kemunculan setiap

Tabel 2.1: Interpretasi Nilai Korelasi

Nilai Korelasi	Interpretasi
$r < 0.2$	hubungan dapat diangap tidak ada
$0.2 \leq r \leq 0.4$	hubungan rendah
$0.4 < r \leq 0.7$	hubungan cukup
$0.7 < r \leq 0.9$	hubungan tinggi
$0.9 < r \leq 1.0$	hubungan sangat tinggi

karakter di *plaintext* dan *ciphertext*. Proses enkripsi yang lebih baik adalah proses yang memiliki nilai kualitas enkripsi yang lebih tinggi. Nilai kualitas enkripsi maksimal muncul jika semua karakter di *plaintext* berbeda dengan karakter di *ciphertext*.

Penghitungan nilai kualitas enkripsi dilakukan dengan menggunakan rumus (Aruljothi, 2012):

$$EQ = \frac{\sum_{L=1}^{N} |H_L(C) - H_L(P)|}{N} \tag{2.38}$$

dimana N adalah banyak karakter, $H_L(C)$ adalah frekuensi kemunculan karakter ke-L di *ciphertext* dan $H_L(P)$ adalah frekuensi kemunculan karakter ke-L di *plaintext*.

2.5.3 *Running Time*

Salah satu hal yang menentukan efektifitas suatu algoritma adalah penggunaan waktu untuk menjalankan program (*running time*). Algoritma yang lebih efisien adalah algoritma yang mempunyai *running time* lebih cepat. Untuk algoritma kriptografi terdapat *running time* enkripsi dan *running time* dekripsi.

2.5.4 Kompleksitas Algoritma

Kompleksitas algoritma merupakan suatu analisis waktu yang diinginkan untuk menyelesaikan suatu permasalahan dengan ukuran input tertentu. Kompleksitas sebuah algoritma dapat diekspresikan dengan jumlah operasi yang digunakan oleh algoritma ketika input mempunyai ukuran tertentu. Operasi-operasi yang digunakan untuk mengukur kompleksitas algoritma dapat berupa operasi perbandingan, penambahan, perkalian atau operasi dasar yang lain.

Sebuah estimasi big-O pada kompleksitas algoritma menyatakan

bagaimana waktu untuk menyelesaikan permasalahan bertambah sesuai dengan pertambahan ukuran input. Algoritma dikatakan mempunyai kompleksitas linier jika mempunyai kompleksitas O(n). Beberapa kompleksitas algoritma ditunjukkan pada Tabel 2.2 berikut (Rosen, 2012).

Tabel 2.2: Kompleksitas Algoritma

Kompleksitas	Terminologi Kompleksitas
O(1)	kompleksitas konstanta
O(log n)	kompleksitas logaritma
O(n)	kompleksitas linier
O(n log n)	kompleksitas n log n
$O(n^b)$	kompleksitas polynomial
$O(b^n)$, untuk $b > 1$	kompleksitas eksponensial
O(n!)	kompleksitas faktorial

2.5.5 Analisis *Key Space*

Penghitungan *key space* dilakukan untuk mengetahui banyaknya kemungkinan kunci dekripsi yang dapat digunakan. Semakin besar *key space* maka semakin banyak kemungkinan kunci yang digunakan. Sehingga untuk menemukan kunci sebenarnya menggunakan *brute force* (mencoba semua kemungkinan yang ada) akan semakin sulit.

BAB III
METODE PENELITIAN

Metode yang digunakan dalam penelitian tesis ini adalah:

a. Studi Literatur
   Dalam tahap ini akan dilakukan identifikasi permasalahan dan studi literatur dari beberapa buku, tesis, paper, maupun artikel dari internet mengenai kriptografi, transformasi wavelet, aljabar max plus, transformasi max plus wavelet dan analisis algoritma.

b. Menyusun Algoritma Kriptografi
   Pada tahap ini disusun algoritma kriptografi menggunakan transformasi max plus wavelet. Langkah-langkah yang dilakukan pada tahap ini adalah:

   (a) Menyusun algoritma enkripsi

   (b) Menyusun kunci kriptografi

   (c) Menyusun algoritma dekripsi

c. Pembuatan Program
   Pada tahapan ini dilakukan pembuatan program kriptografi max plus wavelet menggunakan software Scilab 5.5.2. Input dan output dari program berupa file dengan format txt.

d. Uji Coba Program
   Pada tahap ini dilakukan uji coba program kriptografi yang telah dibuat. Uji coba program dilakukan pada komputer dengan prosesor Pentium Dual Core 2,2 GHz dan memory 1 GB. Selanjutnya dilakukan pengumpulan data hasil uji coba program.

e. Analisis Hasil Uji Coba
   Pada tahap ini dilakukan analisis dan pembahasan terhadap data hasil uji coba program. Analisis kelayakan algoritma kriptografi max plus wavelet dilakukan dengan menghitung korelasi antara *plaintext*

dan *ciphertext*, kualitas enkripsi, *running time* program, menentukan kompleksitas dan *key space* algoritma kriptografi. Kemudian dilakukan penarikan kesimpulan terhadap penelitian yang telah dilakukan serta pemberian saran untuk perbaikan dan pengembangan penelitian selanjutnya.

BAB IV
PEMBAHASAN

Pada bagian ini dipaparkan tentang konstruksi algoritma kriptografi menggunakan transformasi max plus wavelet. Selain itu juga dituliskan algoritma untuk proses analisis dan proses sintesis pada transformasi max plus wavelet. Selanjutnya dilakukan implementasi program, uji coba dan analisis terhadap algoritma yang sudah dibuat.

4.1 Konstruksi Algoritma Kriptografi

Algoritma kriptografi yang akan dikonstruksi termasuk dalam algoritma *stream cipher*. Pada sub bab ini akan dijelaskan langkah-langkah enkripsi, dekripsi dan penyusunan kunci. Proses utama enkripsi berdasarkan proses analisis pada transformasi max plus wavelet. Proses utama dekripsi berdasarkan proses sintesis pada transformasi max plus wavelet. Kunci kriptografi terdiri dari tiga bagian. Kunci bagian pertama adalah kode transformasi max plus wavelet yang digunakan. Kunci bagian kedua adalah banyaknya kanal yang digunakan. Kunci bagian ketiga adalah kode sinyal detail.

4.1.1 Proses Enkripsi

Langkah-langkah enkripsi dilakukan sebagai berikut:

1. Pesan diubah menjadi kode ASCII dan disimpan dalam array **PlainASCII**.

2. Masukkan kunci enkripsi yaitu kunci kriptografi bagian pertama dan bagian kedua.
   Kunci bagian pertama adalah kode transformasi max plus wavelet yang digunakan. Kunci bagian kedua adalah banyaknya kanal yang digunakan. Jika perkalian kunci kanal melebihi ukuran *plaintext*, maka pada *plaintext* ditambahkan karakter spasi sehingga ukuran *plaintext* sama dengan perkalian kunci kanal.

3. **PlainASCII** kemudian dimasukkan ke dalam proses analisis.
   Proses analisis menggunakan max plus wavelet yang sesuai dengan

kunci bagian pertama. Banyaknya kanal yang digunakan sesuai dengan kunci bagian kedua. Dari proses analisis didapatkan sinyal hampiran, sinyal detail dan kunci bagian ketiga yaitu kode sinyal detail yang dihasilkan. Penyusunan kunci bagian ketiga akan dijelaskan pada sub bab berikutnya.

4. Dapatkan **CipherASCII** yaitu kode ASCII dari ciphertext, yang dihasilkan dengan cara sebagai berikut.
   - Pada max plus wavelet tipe A, C, D dan E:
     CipherASCII terdiri dari sinyal hampiran dan nilai absolut sinyal detail + 32.
   - Pada max plus wavelet tipe B:
     - Untuk sinyal detail $<$ -126, **CipherASCII** terdiri dari sinyal hampiran dan nilai absolut sinyal detail - 95.
     - Untuk sinyal detail $\geq$ -126, **CipherASCII** terdiri dari sinyal hampiran dan nilai absolut sinyal detail.
5. **CipherASCII** diubah menjadi text dan disimpan dalam variable **Ciphertext**.

Selanjutnya Ciphertext dan kunci kriptografi akan dikirimkan kepada penerima pesan.

Algoritma proses analisis disusun berdasarkan operator analisis pada transformasi max plus wavelet yang dikonstruksi oleh Fahim (2014). Berikut ini algoritma proses analisis dalam bentuk pseudocode.

Algoritma 4.1 Proses Analisis MP Wavelet Tipe A

```
Input : array 1 dimensi sinyal[] yang berukuran N
        integer p (besarnya kanal)
Output: array 1 dimensi X[] yang berukuran N/p
        (sinyal hampiran)
        array 1 dimensi Y[] yang berukuran N-N/p
        (sinyal detail)
Proses:

k = 0
for j=1:N/p
```

```
    n = p×(j-1)+1
    maks = max(sinyal(n:n+p-1))
    X(j) = maks
    for i=1:p-1
        k = k+1
        Y(k) = sinyal(n+i)-sinyal(n)
    end for
end for
```

Algoritma 4.2 Proses Analisis MP Wavelet Tipe B

```
Input : array 1 dimensi sinyal[] yang berukuran N
        integer p (besarnya kanal)
Output: array 1 dimensi X[] yang berukuran N/p
        (sinyal hampiran)
        array 1 dimensi Y[] yang berukuran N-N/p
        (sinyal detail)
Proses:

k = 0
for j=1:N/p
    n = p×(j-1)+1
    maks = max(sinyal(n:n+p-1))
    X_a(j) = maks
    for i=1:p-1
        k = k+1
        Y_a(k) = sinyal(n+i)-sinyal(n)
    end for
end for
for j=1:N/p
    maks = max(Y_a((p-1)×(j-1)+1:(p-1)×(j-1)+p-1))
    maks = max(maks, X_a(j))
    X(j) = maks
    for i=1:p-1
        Y((p-1)×(j-1)+i) = Y_a((p-1)×(j-1)+i)-X_a(j)
    end for
end for
```

Algoritma 4.3 Proses Analisis MP Wavelet Tipe C

```
Input : array 1 dimensi sinyal[] yang berukuran N
        integer p (besarnya kanal, bilangan kuadrat)
Output: array 1 dimensi X[] yang berukuran N/p
        (sinyal hampiran)
        array 1 dimensi Y[] yang berukuran N-N/p
        (sinyal detail)
Proses:

k = 0
t = sqrt(p)
for j=1:N/p
    for m=0:t-1
        n = p×(j-1)+t×m+1
        maks = max(sinyal(n:n+t-1))
        X_a(n) = maks
        for i=1:t-1
            X_a(n+i) = sinyal(n+i)-sinyal(n)
        end for
    end for
end for
for j=1:N/p
    for m=1:t
        n = p×(j-1)+m
        maks = X_a(n)
        for i=1:t-1
            maks = max(maks, X_a(n+i×t))
            X_b(p×(j-1)+t×(m-1)+1+i) = X_a(n+i×t)-X_a(n)
        end for
        X_b(p×(j-1)+t×(m-1)+1) = maks
    end for
    X(j) = X_b((j-1)×p+1)
    for i=1:p-1
        k = k+1
        Y(k) = X_b((j-1)×p+i+1)
    end for
end for
```

Algoritma 4.4 Proses Analisis MP Wavelet Tipe D

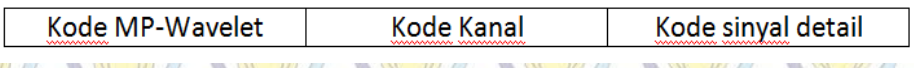
```
Input : array 1 dimensi sinyal[] yang berukuran N
        integer s dan t (besarnya kanal)
Output: array 1 dimensi X[] yang berukuran N/(s*t)
        (sinyal hampiran)
        array 1 dimensi Y[] yang berukuran N - N/(s*t)
        (sinyal detail)
Proses:

k = 0
for j=1:N/(s×t)
    for m=0:s
        n = s×t×(j-1)+t×(m-1)+1
        maks = max(sinyal(n:n+t-1))
        X_a(n) = maks
        for i=1:t-1
            X_a(n+i) = sinyal(n+i)-sinyal(n)
        end for
    end for
end for
for j=1:N/(s×t)
    for m=1:t
        n = s×t×(j-1)+m
        maks = X_a(n)
        for i=1:s-1
            maks = max(maks, X_a(n+i×t))
            X_b(n+i×t) = X_a(n+i×t)-X_a(n)
        end for
        X_b(n) = maks
    end for
    X(j) = X_b((j-1)×s×t+1)
    for i=1:s×t-1
        k = k+1
        Y(k) = X_b((j-1)×s×t+i+1)
    end for
end for
```

Algoritma 4.5 Proses Analisis MP Wavelet Tipe E

```
Input : array 1 dimensi sinyal[] yang berukuran N
        integer p (besarnya kanal)
Output: array 1 dimensi X[] yang berukuran N/p
        (sinyal hampiran)
        array 1 dimensi Y[] yang berukuran N-N/p
        (sinyal hampiran)
Proses:
k = 0
for j=1:N/p
    n = p×(j-1)+1
    maks = max(sinyal(n:n+p-1))
    X(j) = maks
    for i=1:p-1
        k = k+1
        Y_a(k) = sinyal(n+i)-sinyal(n)
    end for
end for
for j=1:N/p
    n = (p-1)×(j-1)+1
    maks = max(Y_a(n:n+p-2))
    Y(n) = maks
    for i=1:p-2
        Y(n+i) = Y_a(n+i)-Y_a(n)
    end for
end for
```

4.1.2 Penyusunan Kunci

Kunci kriptografi terdiri dari tiga bagian, seperti yang disajikan pada Gambar 4.1.



Gambar 4.1: Struktur Kunci Kriptografi

Kunci bagian pertama adalah kode transformasi max plus wavelet yang digunakan. Kode 1 untuk max plus wavelet tipe A, kode 2 untuk tipe B,

kode 3 untuk tipe C, kode 4 untuk tipe D, kode 5 untuk tipe E. Kunci bagian kedua adalah besarnya kanal yang digunakan. Besarnya kanal harus lebih dari satu. Untuk max plus wavelet tipe C, besarnya kanal harus bilangan kuadrat. Untuk max plus wavelet tipe D, banyaknya kunci bagian kedua harus genap.

Kunci bagian ketiga adalah kode sinyal detail yang didapatkan dengan langkah-langkah berikut:

1. • Untuk max plus wavelet tipe A, C, D dan E: Sinyal detail yang bernilai negatif diberi kode 1 dan yang bernilai positif diberi kode 0.
   • Untuk max plus wavelet tipe B: Sinyal detail yang bernilai $<$ -126 diberi kode 1 dan yang bernilai $\geq$ -126 diberi kode 0.
2. Setiap 8 angka dari kode sinyal detail ini akan dibaca menjadi sebuah kode biner suatu bilangan. Begitu juga dengan angka sisanya.
3. Kode biner bilangan ini kemudian diubah menjadi bilangan desimal dan akan menjadi kunci bagian ketiga.

4.1.3 Proses Dekripsi

Langkah-langkah dekripsi dilakukan sebagai berikut:

1. Ciphertext diubah menjadi kode ASCII dan disimpan dalam array **CipherASCII**.
2. Masukkan kunci dekripsi (kunci kriptografi).
3. Kunci kriptografi dipisah menjadi tiga bagian.
   Bagian pertama adalah kode max plus wavelet yang digunakan. Bagian kedua adalah banyaknya kanal yang digunakan. Bagian ketiga adalah kode untuk sinyal detail.
4. Kunci bagian ketiga diubah menjadi kode biner.
5. Angka pertama dari **CipherASCII** akan menjadi sinyal hampiran. Sedangkan sisa **CipherASCII** bersama dengan kode biner akan digunakan untuk mendapatkan sinyal detail dengan rumus sebagai berikut:
   • Untuk max plus wavelet tipe A, C, D dan E:
   Sinyal detail = (CipherASCII - 32) $\times(-1)^{\text{kode biner}}$

- Untuk max plus wavelet tipe B:
  Sinyal detail = -(CipherASCII + 95 $\times$ kode biner)

6. Sinyal hampiran dan sinyal detail dimasukkan kedalam operasi sintesis.
   Operasi sintesis menggunakan max plus wavelet yang sesuai dengan kunci bagian pertama. Banyaknya kanal yang digunakan sesuai dengan kunci bagian kedua.
7. Didapatkan sinyal utama yang disimpan dalam variable **PlainASCII**.
8. **PlainASCII** diubah menjadi pesan awal.

Algoritma proses sintesis disusun berdasarkan operator sintesis pada transformasi max plus wavelet yang dikonstruksi oleh Fahim (2014). Berikut ini algoritma proses sintesis pada max plus wavelet yang ditulis dalam bentuk pseudocode.

Algoritma 4.6 Proses Sintesis MP Wavelet Tipe A

```
Input : array 1 dimensi sinyal_X[] (sinyal hampiran)
        array 1 dimensi sinyal_Y[] (sinyal detail)
        integer p (besarnya kanal)
Output: array 1 dimensi X[] (sinyal hampiran)
        array 1 dimensi Y[] (sinyal detail)
Proses:

k = length(sinyal_X)
for j=0:k-1
    maks = max(sinyal_Y(1:p-1))
    maks = max(maks, 0)
    X(j×p+1) = sinyal_X(j+1)-maks
    for i=1:p-1
        X(j×p+1+i) = X(j×p+1)+sinyal_Y(i)
    end for
    n = length(sinyal_Y)
    sinyal_Y = sinyal_Y(p:n)
end for
Y = sinyal_Y
```

Algoritma 4.7 Proses Sintesis MP Wavelet Tipe B

```
Input : array 1 dimensi sinyal_X[] (sinyal hampiran)
        array 1 dimensi sinyal_Y[] (sinyal detail)
        integer p (besarnya kanal)
Output: array 1 dimensi X[] (sinyal hampiran)
        array 1 dimensi Y[] (sinyal detail)
Proses:

k = length(sinyal_X)
y = length(sinyal_Y)
for j=0:k-1
    maks = max(sinyal_Y(j×(p-1)+1:j×(p-1)+p-1))
    maks = max(maks, 0)
    X_b(j+1) = sinyal_X(j+1)-maks
    for i=1:p-1
        Y_b(j×(p-1)+i) = X_b(j+1)+sinyal_Y(j×(p-1)+i)
    end for
end for
for j=0:k-1
    maks = max(Y_b(1:p-1))
    maks = max(maks, 0)
    X(j×p+1) = X_b(j+1)-maks
    for i=1:p-1
        X(j×p+1+i) = X(j×p+1)+Y_b(i)
    end for
    n = length(Y_b)
    Y_b = Y_b(p:n)
end for
Y = sinyal_Y(k×(p-1)+1:y)
```

Algoritma 4.8 Proses Sintesis MP Wavelet Tipe C

```
Input : array 1 dimensi sinyal_X[] (sinyal hampiran)
        array 1 dimensi sinyal_Y[] (sinyal detail)
        integer p (besarnya kanal, bilangan kuadrat)
Output: array 1 dimensi X[] (sinyal hampiran)
```

```
        array 1 dimensi Y[] (sinyal detail)
Proses:

k = length(sinyal_X)
y = length(sinyal_Y)
t = sqrt(p)
n = 0
for j=0:k-1
    D(j×p+1) = sinyal_X(j+1)
    for m=1:p-1
        n = n+1
        D(j×p+1+m) = sinyal_Y(n)
    end for
end for
for j=0:k-1
    for m=0:t-1
        maks = 0
        for i=1:t-1
            maks = max(maks, D(j×p+1+m×t+i))
        end for
        D_a(j×p+1+m) = D(j×p+1+m×t)-maks
        for i=1:t-1
            D_a(j×p+1+m+i×t) = D_a(j×p+1+m)+D(j×p+1+m×t+i)
        end for
    end for
end for
for j=0:k-1
    for m=0:t-1
        maks = 0
        for i=1:t-1
            maks = max(maks, D_a(j×p+1+m×t+i))
        end for
        X(j×p+1+m×t) = D_a(j×p+1+m×t)-maks
        for i=1:t-1
            X(j×p+1+m×t+i) = X(j×p+1+m×t)+D_a(j×p+1+m×t+i)
        end for
    end for
```

```
end for
Y = sinyal_Y(k×(p-1)+1:y)
```

Algoritma 4.9 Proses Sintesis MP Wavelet Tipe D

```
Input : array 1 dimensi sinyal_X[] (sinyal hampiran)
        array 1 dimensi sinyal_Y[] (sinyal detail)
        integer s dan t (besarnya kanal)
Output: array 1 dimensi X[] (sinyal hampiran)
        array 1 dimensi Y[] (sinyal detail)
Proses:

k = length(sinyal_X)
y = length(sinyal_Y)
n = 0
for j=0:k-1
    D(j×s×t+1) = sinyal_X(j+1)
    for m=1:s×t-1
        n = n+1
        D(j×s×t+1+m) = sinyal_Y(n)
    end for
end for
for j=0:k-1
    for m=1:t
        n = j×s×t+m
        maks = 0
        for i=1:s-1
            maks = max(maks, D(n+i×t))
        end for
        D_a(n) = D(n)-maks
        for i=1:s-1
            D_a(n+i×t) = D_a(n)+D(n+i×t)
        end for
    end for
end for
for j=0:k-1
    for m=0:s-1
```

```
        n = j×s×t+m×t+1
        maks = 0
        for i=1:t-1
            maks = max(maks, D_a(n+i))
        end for
        X(n) = D_a(n)-maks
        for i=1:t-1
            X(n+i) = X(n)+D_a(n+i)
        end for
    end for
end for
Y = sinyal_Y(k×(s×t-1)+1:y)
```

Algoritma 4.10 Proses Sintesis MP Wavelet Tipe E

```
Input : array 1 dimensi sinyal_X[] (sinyal hampiran)
        array 1 dimensi sinyal_Y[] (sinyal detail)
        integer p (besarnya kanal)
Output: array 1 dimensi X[] (sinyal hampiran)
        array 1 dimensi Y[] (sinyal detail)
Proses:

k = length(sinyal_X)
y = length(sinyal_Y)
for j=0:k-1
    n = (p-1)×j+1
    maks = max(sinyal_Y(n+1:n+p-2))
    maks = max(maks, 0)
    Y_b(n) = sinyal_Y(n)-maks
    for i=1:p-2
        Y_b(n+i) = Y_b(n)+sinyal_Y(n+i)
    end for
end for
for j=0:k-1
    n = p×j+1
    maks = max(Y_b(j×(p-1)+1:j×(p-1)+p-1))
    maks = max(maks, 0)
    X(n) = sinyal_X(j+1)-maks
```

```
    for i=1:p-1
        X(n+i) = X(n)+Y_b(j×(p-1)+i)
    end for
end for
Y = sinyal_Y(k×(p-1)+1:y)
```

4.2 Implementasi dan Uji Coba

Algoritma kriptografi max plus wavelet ini diimplementasikan dalam suatu program kriptografi yang disusun menggunakan software Scilab 5.5.2. Input program adalah teks atau file dengan format .txt. Output program adalah teks yang dapat disimpan dalam file dengan format .txt.

Pada GUI program kriptografi terdapat dua menu yaitu *Reset* dan *Proses*. Menu *Reset* berfungsi untuk menghapus semua data yang sedang muncul. Pada menu *Proses* terdapat dua submenu yaitu *Enkripsi* dan *Dekripsi* yang berfungsi untuk memilih proses enkripsi dan dekripsi.

Pada GUI proses enkripsi terdapat lima tombol, yaitu tombol *Buka* untuk memilih file *plaintext*, tombol *Enkripsi* untuk melakukan proses enkripsi, tombol *Hitung Karakter* untuk menghitung banyaknya karakter *plaintext*, serta dua tombol *Simpan* untuk menyimpan *plaintext* dan *ciphertext*. Juga terdapat checkbox sebagai konfirmasi bahwa data yang diinputkan sudah benar. Contoh tampilan GUI program kriptografi max plus wavelet untuk proses enkripsi disajikan pada Gambar 4.2.

Gambar 4.2: GUI Program Kriptografi Max Plus Wavelet Untuk Proses Enkripsi

Untuk melakukan proses enkripsi dapat dijalankan langkah-langkah sebagai berikut. *Plaintext* dapat diambil dari file yang dipilih lewat tombol *Buka*, atau ditulis langsung di tempat *plaintext*. Jika *plaintext* diambil dari file maka akan ditampilkan *plaintext* dan banyaknya karakter dari *plaintext* tersebut. Jika *plaintext* ditulis langsung maka banyaknya karakter dari *plaintext* tersebut dapat dihitung dengan menekan tombol *Hitung Karakter*.

Kemudian dimasukkan kunci enkripsi yang terdiri dari kunci tipe max plus wavelet dan kunci kanal. Setelah semua data telah benar, tekan checkbox kemudian tekan tombol *Enkripsi*. Hasil dari enkripsi adalah *ciphertext* dan kunci sinyal detail. Proses pembentukan *ciphertext* dan kunci sinyal detail menggunakan prosedur yang telah dijelaskan pada sub bab 4.1. *Plaintext* dan *ciphertext* dapat disimpan dengan menekan tombol *Simpan*.

Pada GUI proses dekripsi terdapat lima tombol, yaitu tombol *Buka* untuk memilih file *ciphertext*, tombol *Dekripsi* untuk melakukan proses dekripsi, tombol *Hitung Karakter* untuk menghitung banyaknya karakter *ciphertext*, serta dua tombol *Simpan* untuk menyimpan *plaintext* dan *ciphertext*. Juga terdapat checkbox sebagai konfirmasi bahwa data yang diinputkan sudah benar dan untuk selanjutnya dapat dilakukan proses dekripsi. Contoh tampilan GUI program kriptografi max plus wavelet untuk proses dekripsi disajikan pada Gambar 4.3.

Gambar 4.3: GUI Program Kriptografi Max Plus Wavelet Untuk Proses Dekripsi

Untuk melakukan proses dekripsi dapat dijalankan langkah-langkah sebagai berikut. *Ciphertext* dapat diambil dari file yang dipilih lewat tombol

Buka, atau ditulis langsung di tempat *ciphertext*. Jika *ciphertext* diambil dari file maka akan ditampilkan *ciphertext* dan banyaknya karakter dari *ciphertext* tersebut. Jika *ciphertext* ditulis langsung maka banyaknya karakter dari *ciphertext* tersebut dapat dihitung dengan menekan tombol *Hitung Karakter*.

Kemudian dimasukkan kunci dekripsi yang terdiri dari kunci tipe max plus wavelet, kunci kanal dan kunci sinyal detail. Setelah semua data telah benar, tekan checkbox kemudian tekan tombol *Dekripsi*. Hasil dari dekripsi adalah *plaintext* yang ditampilkan di textbox. Proses pembentukan *plaintext* menggunakan prosedur yang telah dijelaskan pada sub bab 4.1. *Plaintext* dan *ciphertext* dapat disimpan dengan menekan tombol *Simpan*.

Gambar 4.2 dan Gambar 4.3 adalah contoh uji coba program menggunakan file input uji 1.txt dengan banyak karakter 299. Uji coba program dilakukan pada komputer dengan prosesor Pentium Dual Core 2,2 GHz dan memory 1 GB. Secara keseluruhan terdapat sembilan file input yang digunakan untuk uji coba yang disajikan pada Tabel 4.1.

Tabel 4.1: Data File Input Uji Coba

No.	Nama File	Banyak Karakter
1.	uji 1.txt	299
2.	uji 2.txt	999
3.	uji 3.txt	3000
4.	uji 4.txt	5995
5.	uji 5.txt	9997
6.	uji 6.txt	12543
7.	uji 7.txt	16895
8.	uji 8.txt	20723
9.	uji kanal.txt	29

File kesatu sampai dengan file kedelapan digunakan pada uji coba untuk mengetahui perbedaan antara kelima tipe max plus wavelet. Sehingga uji coba dilakukan dengan menggunakan kunci kanal yang sama untuk semua tipe max plus wavelet. Sedangkan file kesembilan digunakan pada uji coba untuk mengetahui pengaruh kunci kanal terhadap hasil enkripsi. Sehingga uji coba dilakukan dengan menggunakan satu tipe max plus wavelet dan kunci kanal yang berbeda-beda.

Untuk keperluan analisis, pada uji coba proses enkripsi dicatat nilai

korelasi antara *plaintext* dan *ciphertext*, kualitas enkripsi dan waktu proses enkripsi. Sedangkan pada uji coba proses dekripsi dicatat waktu proses dekripsi. Data-data ini kemudian dianalisis untuk mengetahui kelayakan dari algoritma kriptografi ini.

4.3 Analisis Kelayakan Algoritma Kriptografi

Analisis kelayakan algoritma ini bertujuan untuk mengetahui kualitas dan performa algoritma kriptografi yang dikonstruksi. Untuk mengetahui kualitas algoritma kriptografi dilakukan penghitungan nilai korelasi antara *plaintext* dan *ciphertext*, penghitungan nilai kualitas enkripsi dan analisis *key space*. Untuk mengetahui performa algoritma kriptografi dilakukan penghitungan *running time* serta analisis kompleksitas algoritma.

Untuk mengetahui perbedaan antara kelima tipe max plus wavelet, maka data yang digunakan untuk analisis korelasi, kualitas enkripsi dan *running time* berikut ini merupakan hasil uji coba program dengan menggunakan kunci kanal yang sama untuk semua tipe max plus wavelet. Kunci kanal max plus wavelet tipe D adalah akar dari kunci kanal dari tipe yang lain.

Uji coba dengan file uji 1.txt menggunakan kunci kanal 4 9 9 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 3 3 3 3. Sehingga ukuran *plaintext* akan berubah dari 299 menjadi 324. Uji coba dengan file uji 2.txt menggunakan kunci kanal 4 4 4 4 4 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 2 2 2 2 2 2 2 2. Sehingga ukuran *plaintext* akan berubah dari 999 menjadi 1024.

Uji coba dengan file uji 3.txt menggunakan kunci kanal 25 121 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 5 5 11 11. Sehingga ukuran *plaintext* akan berubah dari 3000 menjadi 3025. Uji coba dengan file uji 4.txt menggunakan kunci kanal 4 9 169 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 3 3 13 13. Sehingga ukuran *plaintext* akan berubah dari 5995 menjadi 6084.

Uji coba dengan file uji 5.txt menggunakan kunci kanal 4 25 4 25 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 5 5 2 2 5 5. Sehingga ukuran *plaintext* akan berubah dari 9997 menjadi 10000. Uji coba dengan file uji 6.txt menggunakan kunci kanal 4 4 4 4 49 untuk max plus wavelet tipe A, B,

C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 2 2 2 2 2 2 7 7. Sehingga ukuran *plaintext* akan berubah dari 12543 menjadi 12544.

Uji coba dengan file uji 7.txt menggunakan kunci kanal 4 25 169 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 5 5 13 13. Sehingga ukuran *plaintext* akan berubah dari 16895 menjadi 16900. Uji coba dengan file uji 8.txt menggunakan kunci kanal 4 4 4 4 9 9 untuk max plus wavelet tipe A, B, C dan E. Sedangkan untuk max plus wavelet tipe D menggunakan kunci kanal 2 2 2 2 2 2 2 2 3 3 3 3. Sehingga ukuran *plaintext* akan berubah dari 20723 menjadi 20736.

4.3.1 Korelasi *plaintext* dan *ciphertext*

Dari uji coba program didapatkan data korelasi *plaintext* dan *ciphertext*. Penghitungan nilai korelasi dilakukan untuk mengetahui hubungan linier antara *plaintext* dan *ciphertext*. Rumus untuk menghitung nilai korelasi ditulis ulang sebagai berikut (Aruljothi, 2012):

$$r = \frac{n\sum(xy) - \sum x \sum y}{\sqrt{(n\sum(x^2) - (\sum x)^2)(n\sum(y^2) - (\sum y)^2)}}$$

dimana x adalah kode ASCII *plaintext*, y adalah kode ASCII *ciphertext* dan n adalah panjang *plaintext*.

Nilai korelasi r menyatakan hubungan linier antara *plaintext* dan *ciphertext*. Jika korelasi bernilai 1 atau -1 maka *ciphertext* mempunyai hubungan linier yang kuat dengan *plaintext*. Jika korelasi bernilai 0 maka *plaintext* dan *ciphertext* tidak mempunyai hubungan linier. Hal ini menunjukkan bahwa algoritma tersebut mempunyai proses enkripsi yang baik. Beberapa data korelasi dari hasil uji coba disajikan pada Tabel 4.2.

Tanda negatif hanya menunjukkan bahwa arah kemiringan *ciphertext* berlawanan dengan kemiringan *plaintext*. Sedangkan tanda positif menunjukkan bahwa arah kemiringan *ciphertext* sama dengan kemiringan *plaintext*. Sebagai contoh korelasi 0.3 dan -0.3 mempunyai kekuatan korelasi yang sama, perbedaannya hanya ada pada arah kemiringan *ciphertext*.

Dari Tabel 4.2 terlihat bahwa nilai korelasi sebagian besar berada pada selang -0.1 sampai 0.1. Hanya ada dua data yang berada diluar selang tersebut yaitu 105.0945×10^{-3} dan 214.8343×10^{-3}. Nilai mutlak korelasi dibawah 0.1 menunjukkan bahwa korelasi linier antara *plaintext*

Tabel 4.2: Korelasi *plaintext* dan *ciphertext*

Nama File	Nilai Korelasi (dalam 10^{-3})				
	Tipe A	Tipe B	Tipe C	Tipe D	Tipe E
uji 1.txt	105.0945	214.8343	88.5354	47.2774	67.2378
uji 2.txt	6.129	66.1493	-6.1864	32.8503	34.4253
uji 3.txt	-20.5778	26.1839	4.8731	0.9255	-6.1328
uji 4.txt	18.1245	23.5925	19.6891	28.3679	33.8908
uji 5.txt	-13.7571	10.3303	-11.9865	2.6814	6.3104
uji 6.txt	1.3134	6.186	5.8487	5.7399	-0.3167
uji 7.txt	-6.5161	-5.6886	-9.5715	-1.7711	-14.5938
uji 8.txt	9.2732	-4.3349	7.7835	6.4879	7.2295

dan *ciphertext* sangat rendah. Dengan kata lain hampir tidak ada korelasi linier antara *plaintext* dan *ciphertext*. Hal ini menunjukkan bahwa algoritma kriptografi ini mempunyai proses enkripsi yang baik.

Data dari Tabel 4.2 menunjukkan semua tipe max plus wavelet mempunyai nilai korelasi yang kecil. Tidak ada tipe max plus wavelet yang selalu lebih baik daripada tipe yang lain. Nilai korelasi cenderung semakin kecil seiring dengan bertambahnya banyak karakter. Pada file uji 1.txt dengan 324 karakter nilai korelasi berada pada selang 47.2774×10^{-3} sampai 214.8343×10^{-3}. Sedangkan pada file uji 8.txt dengan 20736 karakter nilai mutlak korelasi berada pada selang 4.3349×10^{-3} sampai 9.2732×10^{-3}.

4.3.2 Kualitas Enkripsi

Dari uji coba program didapatkan data kualitas enkripsi. Penghitungan kualitas enkripsi dilakukan dengan membandingkan frekuensi kemunculan setiap karakter di *plaintext* dan *ciphertext*. Rumus untuk menghitung kualitas enkripsi ditulis ulang sebagai berikut (Aruljothi, 2012):

$$EQ = \frac{\sum_{L=32}^{126} |H_L(C) - H_L(P)|}{95}$$

dimana $H_L(C)$ adalah frekuensi kemunculan karakter dengan kode ASCII L di *ciphertext* dan $H_L(P)$ adalah frekuensi kemunculan karakter dengan kode ASCII L di *plaintext*.

Nilai kualitas enkripsi EQ merepresentasikan rata-rata selisih frekuensi kemunculan setiap karakter di *plaintext* dan *ciphertext*. Beberapa data kualitas enkripsi dari hasil uji coba disajikan pada Tabel 4.3.

Tabel 4.3: Kualitas Enkripsi

Nama File	Kualitas Enkripsi				
	Tipe A	Tipe B	Tipe C	Tipe D	Tipe E
uji 1.txt	4.6315789	3.0526316	5.0631579	5.0631579	5.0947368
uji 2.txt	14.505263	9.6631579	16.189474	16.189474	15.726316
uji 3.txt	42.736842	37.431579	41.863158	41.863158	43.136842
uji 4.txt	86.315789	56.947368	95.631579	95.631579	90.863158
uji 5.txt	146.10526	103.23158	163.46316	163.46316	155.54737
uji 6.txt	182.52632	119.22105	200.52632	200.52632	192.50526
uji 7.txt	247.87368	168.23158	274.22105	274.22105	263.95789
uji 8.txt	301.81053	194.45263	334.2	334.2	319.27368

Dari Tabel 4.3 terlihat bahwa untuk semua tipe max plus wavelet, semakin panjang *plaintext* maka nilai kualitas enkripsi juga semakin besar. Nilai kualitas enkripsi dari max plus wavelet tipe C dan D adalah yang paling besar. Nilai terbesar berikutnya adalah tipe E, tipe A dan tipe B adalah yang terkecil. Dari data tersebut dapat diambil kesimpulan bahwa berdasarkan nilai kualitas enkripsi, max plus wavelet tipe C dan D adalah yang terbaik.

Nilai kualitas enkripsi maksimal muncul jika semua karakter di *plaintext* berbeda dengan karakter di *ciphertext*. Untuk algoritma kriptografi ini nilai kualitas enkripsi maksiml dapat dihitung dengan rumus 2n/95, dimana n adalah panjang *plaintext*. Dengan rumus ini dapat diketahui bahwa nilai kualitas enkripsi maksimal dari uji 1 adalah 6.821052632, uji 2 adalah 21.55789474, uji 3 adalah 63.68421053, uji 4 adalah 128.0842105, uji 5 adalah 210.5263158, uji 6 adalah 264.0842105, uji 7 adalah 355.7894737 dan uji 8 adalah 436.5473684.

Dari penghitungan dapat diketahui bahwa persentase kualitas enkripsi dari tipe A adalah 68.37% dari nilai kualitas enkripsi maksimal. Tipe B adalah 47.35%, tipe C adalah 74.61%, tipe D adalah 74.61% dan tipe A adalah 72.55% dari nilai kualitas enkripsi maksimal. Rata-rata kualitas enkripsi dari seluruh uji coba adalah 67.5% dari nilai kualitas enkripsi maksimal. Dari penghitungan ini dapat diambil kesimpulan bahwa algoritma kriptografi ini mempunyai kualitas enkripsi yang baik yaitu 67.5% dari nilai kualitas enkripsi maksimal.

Kualitas enkripsi dari max plus wavelet tipe C sama dengan tipe D,

tetapi nilai korelasinya berbeda. Hal ini menunjukkan bahwa frekuensi kemunculan tiap-tiap karakter pada *ciphertext* hasil dari max plus wavelet tipe C sama dengan tipe D, tetapi mempunyai susunan yang berbeda.

4.3.3 *Running Time*

Running time algoritma kriptografi dibedakan menjadi dua yaitu *running time* enkripsi dan *running time* dekripsi.

***Running Time* Enkripsi**

Penghitungan waktu enkripsi dimulai dari proses mengubah *plaintext* menjadi kode ASCII, proses analisis, proses penyusunan kunci hingga proses mendapatkan *ciphertext*. Beberapa data *running time* enkripsi disajikan pada Tabel 4.4 dan juga disajikan dalam bentuk grafik pada Gambar 4.4.

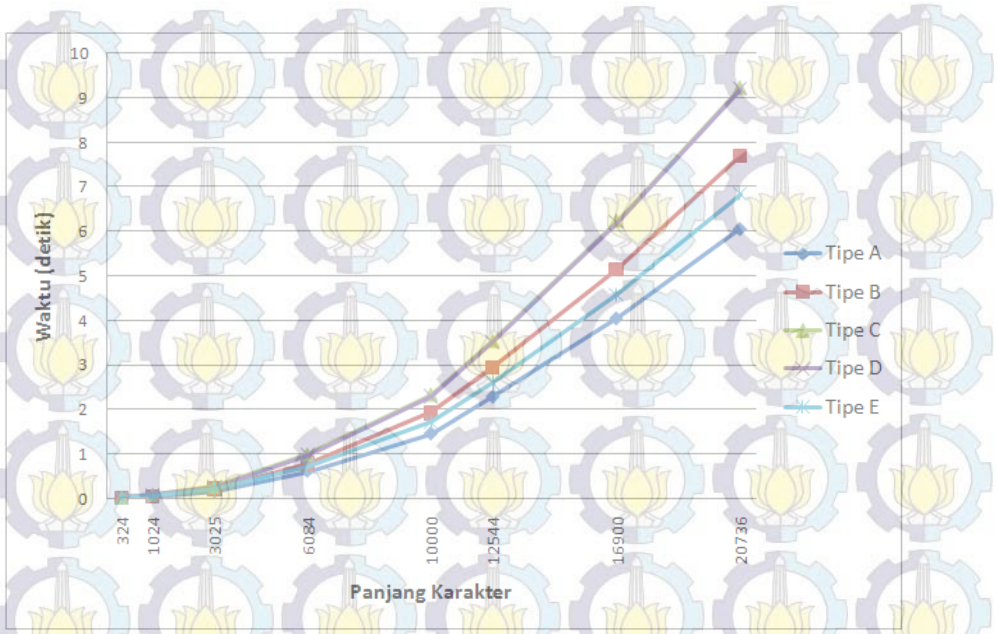
Tabel 4.4: *Running Time* Enkripsi

Nama File	*Running Time* Enkripsi (detik)				
	Tipe A	Tipe B	Tipe C	Tipe D	Tipe E
uji 1.txt	0.016	0.031	0.031	0.031	0.031
uji 2.txt	0.047	0.062	0.094	0.094	0.062
uji 3.txt	0.171	0.234	0.281	0.265	0.234
uji 4.txt	0.624	0.796	0.998	0.983	0.733
uji 5.txt	1.466	1.935	2.34	2.309	1.716
uji 6.txt	2.294	2.964	3.541	3.525	2.621
uji 7.txt	4.056	5.163	6.24	6.224	4.586
uji 8.txt	6.069	7.707	9.219	9.204	6.833

Dari Tabel 4.4 terlihat bahwa pada uji 1, tipe B, tipe C, tipe D dan tipe E mempunyai *running time* yang sama. Dari uji coba secara keseluruhan max plus wavelet tipe A adalah tipe yang paling cepat. Urutan selanjutnya adalah tipe E, tipe B, tipe D dan yang paling lambat adalah tipe C.

Pada Gambar 4.4 terlihat bahwa *running time* enkripsi bertambah secara linier seiring dengan bertambahnya panjang karakter *plaintext*. Nilai taksiran gradien garis dapat dihitung dengan menggunakan rumus regresi (Walpole, 1982):

$$b = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$



Gambar 4.4: *Running Time* Enkripsi

dimana x adalah panjang karakter, y adalah *running time* dan n adalah banyak data.

Nilai taksiran untuk gradien max plus wavelet tipe A adalah 0.000282478. Hal ini berarti setiap penambahan 1000 karakter, maka waktu enkripsi akan bertambah sebesar 0.282478 detik. Nilai taksiran untuk gradien tipe B adalah 0.000359115, tipe C adalah 0.000430312, tipe D adalah 0.000429633, dan gradien tipe E memiliki nilai taksiran 0.000317709. Dari penghitungan ini diketahui bahwa waktu enkripsi bertambah antara 0.282478 sampai dengan 0.430312 detik untuk setiap penambahan 1000 karakter dalam *plaintext*.

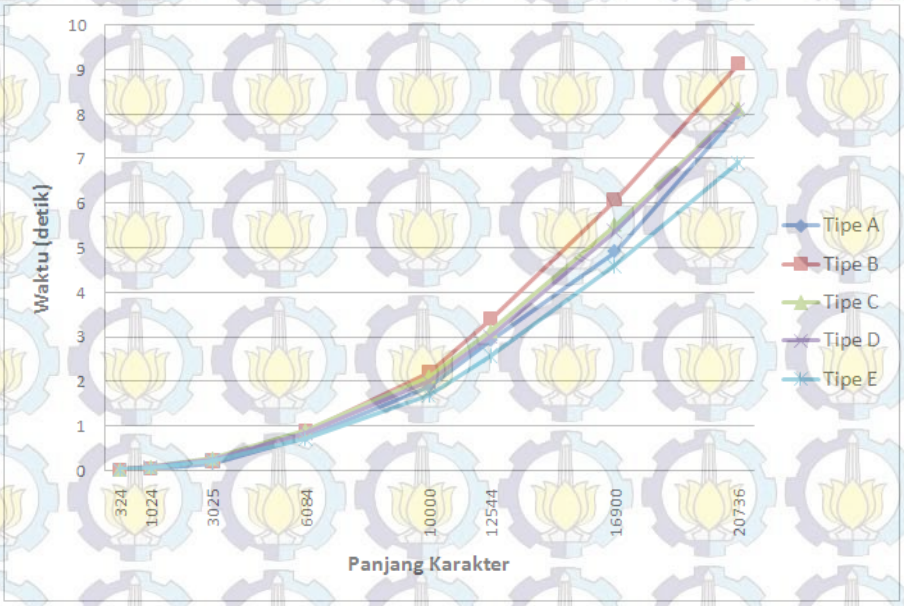
***Running Time* Dekripsi**

Penghitungan waktu dekripsi dimulai dari proses mengubah *ciphertext* menjadi kode ASCII, proses mengubah kunci menjadi kode biner, proses sintesis dan proses mendapatkan *plaintext*. Beberapa data *running time* dekripsi dari hasil uji coba disajikan pada Tabel 4.5 dan juga disajikan dalam bentuk grafik pada Gambar 4.5.

Dari Tabel 4.5 terlihat bahwa pada uji 1, tipe B, C, D dan E mempunyai *running time* yang sama. Pada uji 1, uji 2 dan uji 3, MP-Wavelet tipe A adalah tipe yang paling cepat diikuti dengan tipe E, tipe B, tipe D dan tipe C. Pada

Tabel 4.5: *Running Time* Dekripsi

Nama File	*Running Time* Dekripsi (detik)				
	Tipe A	Tipe B	Tipe C	Tipe D	Tipe E
uji 1.txt	0.016	0.031	0.031	0.031	0.031
uji 2.txt	0.047	0.062	0.094	0.093	0.062
uji 3.txt	0.171	0.234	0.281	0.25	0.234
uji 4.txt	0.78	0.889	0.92	0.858	0.717
uji 5.txt	1.903	2.215	2.121	2.012	1.685
uji 6.txt	2.933	3.416	3.135	3.011	2.589
uji 7.txt	4.929	6.099	5.523	5.382	4.602
uji 8.txt	8.034	9.142	8.127	8.097	6.911



Gambar 4.5: *Running Time* Dekripsi

uji 4, urutan dari yang tercepat adalah tipe E, tipe A, tipe D, tipe B dan tipe C. Sedangkan pada uji 5 sampai uji 8, max plus wavelet tipe E adalah tipe yang paling cepat. Urutan selanjutnya adalah tipe A, tipe D, tipe C dan yang paling lambat adalah tipe B.

Pada Gambar 4.5 terlihat bahwa *running time* dekripsi bertambah secara linier seiring dengan bertambahnya panjang karakter *ciphertext*. Nilai taksiran gradien garis dapat dihitung dengan menggunakan rumus regresi. Nilai taksiran untuk gradien max plus wavelet tipe A adalah 0.000365436. Hal ini berarti setiap penambahan 1000 karakter, maka waktu dekripsi akan bertambah sebesar 0.365436 detik.

Nilai taksiran untuk gradien tipe B adalah 0.000425664, tipe C adalah 0.000379026, tipe D adalah 0.000374675, dan tipe E adalah 0.000320096. Dari penghitungan ini diketahui bahwa waktu dekripsi bertambah antara 0.320096 sampai dengan 0.425664 detik untuk setiap penambahan 1000 karakter dalam *ciphertext*.

4.3.4 Kompleksitas Algoritma

Kompleksitas algoritma dapat dianalisa menggunakan banyaknya kerja yang dilakukan oleh algoritma. Proses yang akan dianalisa banyak kerjanya yaitu proses enkripsi yang terdiri dari proses analisis, penyusunan kunci dan pembentukan *ciphertext*, serta proses dekripsi yang terdiri dari proses sintesis dan penyusunan sinyal detail. Berikut ini analisa kerja dari proses-proses tersebut. S menyatakan proses penjumlahan atau pengurangan, dan C menyatakan proses perbandingan.

Proses Analisis Max Plus Wavelet Tipe A

Dari Algoritma 4.1 terlihat bahwa pada tipe A proses analisis untuk setiap p kanal terdiri dari $p-1$ perbandingan dan $p-1$ pengurangan. Jika pada enkripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(p_1-1)(C+S) \\ W_2 &= \frac{n}{p_1 p_2}(p_2-1)(C+S) \\ &\vdots \\ W_k &= \frac{n}{p_1 p_2 \ldots p_k}(p_k-1)(C+S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_k \\ &= \frac{n}{p_1}(p_1-1)(C+S) + \frac{n}{p_1 p_2}(p_2-1)(C+S) + \ldots \\ &\quad + \frac{n}{p_1 p_2 \ldots p_k}(p_k-1)(C+S). \end{aligned}$$

karena $p_1 - 1 < p_1, p_2 - 1 < p_2 \ldots p_k - 1 < p_k$, dan jika penulisan $(C+S)$ dihilangkan maka

$$W < \frac{n}{p_1}(p_1) + \frac{n}{p_1 p_2}(p_2) + \ldots + \frac{n}{p_1 p_2 \ldots p_k}(p_k)$$

$$\begin{aligned} W &< n + \frac{n}{p_1} + \ldots + \frac{n}{p_1 p_2 \cdots p_{k-1}} \\ &< n + n + \ldots + n \\ &= kn. \end{aligned}$$

Terlihat bahwa $W < kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Analisis Max Plus Wavelet Tipe B

Dari Algoritma 4.2 terlihat bahwa proses analisis untuk setiap p kanal terdiri dari $2p-2$ perbandingan dan $2p-2$ pengurangan. Jika terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(2p_1 - 2)(C+S) \\ W_2 &= \frac{n}{p_1 p_2}(2p_2 - 2)(C+S) \\ &\vdots \\ W_k &= \frac{n}{p_1 p_2 \cdots p_k}(2p_k - 2)(C+S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_k \\ &= \frac{n}{p_1}(2p_1 - 2)(C+S) + \frac{n}{p_1 p_2}(2p_2 - 2)(C+S) + \ldots \\ &\quad + \frac{n}{p_1 p_2 \cdots p_k}(2p_k - 2)(C+S). \end{aligned}$$

karena $2p_1 - 2 < 2p_1, 2p_2 - 2 < 2p_2 \ldots 2p_k - 2 < 2p_k$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1) + \frac{n}{p_1 p_2}(2p_2) + \ldots + \frac{n}{p_1 p_2 \cdots p_k}(2p_k) \\ W &< 2n + \frac{2n}{p_1} + \ldots + \frac{2n}{p_1 p_2 \cdots p_{k-1}} \\ &< 2n + 2n + \ldots + 2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Analisis Max Plus Wavelet Tipe C

Pada tipe C ini banyak kanal $p = t \times t$. Dari Algoritma 4.3 terlihat bahwa pada tipe C proses analisis untuk setiap p kanal terdiri dari $2t(t-1)$ perbandingan dan $2t(t-1)$ pengurangan. Jika pada enkripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(2t_1(t_1-1))(C+S) \\ W_2 &= \frac{n}{p_1 p_2}(2t_2(t_2-1))(C+S) \\ &\vdots \\ W_k &= \frac{n}{p_1 p_2 \ldots p_k}(2t_k(t_k-1))(C+S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_k \\ &= \frac{n}{p_1}(2t_1(t_1-1))(C+S) + \frac{n}{p_1 p_2}(2t_2(t_2-1))(C+S) + \ldots \\ &\quad + \frac{n}{p_1 p_2 \ldots p_k}(2t_k(t_k-1))(C+S). \end{aligned}$$

karena $2t_1(t_1-1) < 2p_1, 2t_2(t_2-1) < 2p_2 \ldots 2t_k(t_k-1) < 2p_k$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1) + \frac{n}{p_1 p_2}(2p_2) + \ldots + \frac{n}{p_1 p_2 \ldots p_k}(2p_k) \\ W &< 2n + \frac{2n}{p_1} + \ldots + \frac{2n}{p_1 p_2 \ldots p_{k-1}} \\ &< 2n + 2n + \ldots + 2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Analisis Max Plus Wavelet Tipe D

Pada tipe D ini banyak kanal $p = s \times t$. Dari Algoritma 4.4 terlihat bahwa pada tipe D proses analisis untuk setiap p kanal terdiri dari $s(t-1)+t(s-1)$ perbandingan dan $s(t-1)+t(s-1)$ pengurangan. Jika pada enkripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan

adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(s_1(t_1-1)+t_1(s_1-1))(C+S) \\ W_2 &= \frac{n}{p_1p_2}(s_2(t_2-1)+t_2(s_2-1))(C+S) \\ &\vdots \\ W_k &= \frac{n}{p_1p_2\cdots p_k}(s_k(t_k-1)+t_k(s_k-1))(C+S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1+W_2+\ldots+W_k \\ W &= \frac{n}{p_1}(s_1(t_1-1)+t_1(s_1-1))(C+S)+ \\ &\quad \frac{n}{p_1p_2}(s_2(t_2-1)+t_2(s_2-1))(C+S)+\ldots \\ &\quad +\frac{n}{p_1p_2\cdots p_k}(s_k(t_k-1)+t_k(s_k-1))(C+S) \end{aligned}$$

karena $s_1(t_1-1)+t_1(s_1-1) < 2p_1, s_2(t_2-1)+t_2(s_2-1) < 2p_2 \ldots s_k(t_k-1)+t_k(s_k-1) < 2p_k$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1)+\frac{n}{p_1p_2}(2p_2)+\ldots+\frac{n}{p_1p_2\cdots p_k}(2p_k) \\ W &< 2n+\frac{2n}{p_1}+\ldots+\frac{2n}{p_1p_2\cdots p_{k-1}} \\ &< 2n+2n+\ldots+2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Analisis Max Plus Wavelet Tipe E

Dari Algoritma 4.5 terlihat bahwa pada tipe E proses analisis untuk setiap p kanal terdiri dari $2p-3$ perbandingan dan $2p-3$ pengurangan. Jika pada enkripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(2p_1-3)(C+S) \\ W_2 &= \frac{n}{p_1p_2}(2p_2-3)(C+S) \end{aligned}$$

$$\vdots$$
$$W_k = \frac{n}{p_1 p_2 \cdots p_k}(2p_k - 3)(C+S).$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_k \\ &= \frac{n}{p_1}(2p_1 - 3)(C+S) + \frac{n}{p_1 p_2}(2p_2 - 3)(C+S) + \ldots \\ &\quad + \frac{n}{p_1 p_2 \cdots p_k}(2p_k - 3)(C+S). \end{aligned}$$

karena $2p_1 - 3 < 2p_1, 2p_2 - 3 < 2p_2 \ldots 2p_k - 3 < 2p_k$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1) + \frac{n}{p_1 p_2}(2p_2) + \ldots + \frac{n}{p_1 p_2 \cdots p_k}(2p_k) \\ W &< 2n + \frac{2n}{p_1} + \ldots + \frac{2n}{p_1 p_2 \cdots p_{k-1}} \\ &< 2n + 2n + \ldots + 2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Sintesis Max Plus Wavelet Tipe A

Dari Algoritma 4.6 terlihat bahwa pada tipe A proses sintesis untuk setiap p kanal terdiri dari $p-1$ perbandingan dan p penjumlahan. Jika terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}((p_1 - 1)C + p_1 S) \\ W_2 &= \frac{n}{p_1 p_2}((p_2 - 1)C + p_2 S) \\ &\vdots \\ W_k &= \frac{n}{p_1 p_2 \cdots p_k}((p_k - 1)C + p_k S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_k \\ W &= \frac{n}{p_1}((p_1 - 1)C + p_1 S) + \frac{n}{p_1 p_2}((p_2 - 1)C + p_2 S) + \ldots \end{aligned}$$

$$+\frac{n}{p_1 p_2 \dots p_k}((p_k-1)C+p_k S).$$

karena $(p_1-1)C+p_1S<p_1(C+S),(p_2-1)C+p_2S<p_2(C+S)\dots(p_k-1)C+p_kS<p_k(C+S)$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(p_1)+\frac{n}{p_1p_2}(p_2)+\dots+\frac{n}{p_1p_2\dots p_k}(p_k) \\ W &< n+\frac{n}{p_1}+\dots+\frac{n}{p_1p_2\dots p_{k-1}} \\ &< n+n+\dots+n \\ &= kn. \end{aligned}$$

Terlihat bahwa $W<kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Sintesis Max Plus Wavelet Tipe B

Dari Algoritma 4.7 terlihat bahwa pada tipe B proses sintesis untuk setiap p kanal terdiri dari $2(p-1)$ perbandingan dan $2p$ penjumlahan. Jika pada dekripsi terdapat kunci kanal $p_1,p_2,\dots p_k$, maka banyaknya kerja yang dilakukan adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(2(p_1-1)C+2p_1S) \\ W_2 &= \frac{n}{p_1p_2}(2(p_2-1)C+2p_2S) \\ &\vdots \\ W_k &= \frac{n}{p_1p_2\dots p_k}(2(p_k-1)C+2p_kS). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1+W_2+\dots+W_k \\ &= \frac{n}{p_1}(2(p_1-1)C+2p_1S)+\frac{n}{p_1p_2}(2(p_2-1)C+2p_2S)+\dots \\ &\quad +\frac{n}{p_1p_2\dots p_k}(2(p_k-1)C+2p_kS). \end{aligned}$$

karena $2(p_1-1)C+2p_1S<2p_1(C+S),2(p_2-1)C+2p_2S<2p_2(C+S)\dots2(p_k-1)C+2p_kS<2p_k(C+S)$, dan jika penulisan $(C+S)$ dihilangkan maka

$$W < \frac{n}{p_1}(2p_1)+\frac{n}{p_1p_2}(2p_2)+\dots+\frac{n}{p_1p_2\dots p_k}(2p_k)$$

$$\begin{aligned} W &< 2n + \frac{2n}{p_1} + \ldots + \frac{2n}{p_1 p_2 \cdots p_{k-1}} \\ &< 2n + 2n + \ldots + 2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Sintesis Max Plus Wavelet Tipe C

Pada tipe C ini banyak kanal $p = t \times t$. Dari Algoritma 4.8 terlihat bahwa pada tipe C proses sintesis untuk setiap p kanal terdiri dari $2t(t-1)$ perbandingan dan $2t^2$ penjumlahan. Jika pada dekripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan pada proses sintesis max plus wavelet tipe C ini adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}(2t_1(t_1-1)C + 2t_1^2 S) \\ W_2 &= \frac{n}{p_1 p_2}(2t_2(t_2-1)C + 2t_2^2 S) \\ &\vdots \\ W_k &= \frac{n}{p_1 p_2 \cdots p_k}(2t_k(t_k-1)C + 2t_k^2 S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1 + W_2 + \ldots + W_k \\ W &= \frac{n}{p_1}(2t_1(t_1-1)C + 2t_1^2 S) + \frac{n}{p_1 p_2}(2t_2(t_2-1)C + 2t_2^2 S) + \ldots \\ &\quad + \frac{n}{p_1 p_2 \cdots p_k}(2t_k(t_k-1)C + 2t_k^2 S). \end{aligned}$$

karena $2t_1(t_1-1)C + 2t_1^2 S < 2p_1(C+S), 2t_2(t_2-1)C + 2t_2^2 S < 2p_2(C+S) \ldots 2t_k(t_k-1)C + 2t_k^2 S < 2p_k(C+S)$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1) + \frac{n}{p_1 p_2}(2p_2) + \ldots + \frac{n}{p_1 p_2 \cdots p_k}(2p_k) \\ W &< 2n + \frac{2n}{p_1} + \ldots + \frac{2n}{p_1 p_2 \cdots p_{k-1}} \\ &< 2n + 2n + \ldots + 2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Sintesis Max Plus Wavelet Tipe D

Pada tipe D ini banyak kanal adalah p dimana $p = s \times t$. Dari Algoritma 4.9 terlihat bahwa pada tipe D proses sintesis untuk setiap p kanal terdiri dari $t(s-1)+s(t-1)$ perbandingan dan $2st$ penjumlahan. Jika pada dekripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan pada proses sintesis ini adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}((t_1(s_1-1)+s_1(t_1-1))C+2s_1t_1S) \\ W_2 &= \frac{n}{p_1p_2}((t_2(s_2-1)+s_2(t_2-1))C+2s_2t_2S) \\ &\vdots \\ W_k &= \frac{n}{p_1p_2\cdots p_k}((t_k(s_k-1)+s_k(t_k-1))C+2s_kt_kS). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1+W_2+\ldots+W_k \\ W &= \frac{n}{p_1}((t_1(s_1-1)+s_1(t_1-1))C+2s_1t_1S)+ \\ &\quad \frac{n}{p_1p_2}((t_2(s_2-1)+s_2(t_2-1))C+2s_2t_2S)+\ldots \\ &\quad +\frac{n}{p_1p_2\cdots p_k}((t_k(s_k-1)+s_k(t_k-1))C+2s_kt_kS). \end{aligned}$$

karena $(t_1(s_1-1)+s_1(t_1-1))C+2s_1t_1S < 2p_1(C+S), (t_2(s_2-1)+s_2(t_2-1))C+2s_2t_2S < 2p_2(C+S)\ldots(t_k(s_k-1)+s_k(t_k-1))C+2s_kt_kS < 2p_k(C+S)$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1)+\frac{n}{p_1p_2}(2p_2)+\ldots+\frac{n}{p_1p_2\cdots p_k}(2p_k) \\ W &< 2n+\frac{2n}{p_1}+\ldots+\frac{2n}{p_1p_2\cdots p_{k-1}} \\ &< 2n+2n+\ldots+2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Sintesis Max Plus Wavelet Tipe E

Dari Algoritma 4.10 terlihat bahwa pada tipe D proses sintesis untuk setiap p kanal terdiri dari $2p-3$ perbandingan dan $2p-1$ penjumlahan. Jika pada dekripsi terdapat kunci kanal $p_1, p_2, \ldots p_k$, maka banyaknya kerja yang dilakukan adalah:

$$\begin{aligned} W_1 &= \frac{n}{p_1}((2p_1-3)C+(2p_1-1)S) \\ W_2 &= \frac{n}{p_1p_2}((2p_2-3)C+(2p_2-1)S) \\ &\vdots \\ W_k &= \frac{n}{p_1p_2\cdots p_k}((2p_k-3)C+(2p_k-1)S). \end{aligned}$$

Sehingga kerja keseluruhan

$$\begin{aligned} W &= W_1+W_2+\ldots+W_k \\ W &= \frac{n}{p_1}((2p_1-3)C+(2p_1-1)S)+ \\ &\quad \frac{n}{p_1p_2}((2p_2-3)C+(2p_2-1)S)+\ldots \\ &\quad +\frac{n}{p_1p_2\cdots p_k}((2p_k-3)C+(2p_k-1)S). \end{aligned}$$

karena $(2p_1-3)C+(2p_1-1)S < 2p_1(C+S), (2p_2-3)C+(2p_2-1)S < 2p_2(C+S)\ldots(2p_k-3)C+(2p_k-1)S < 2p_k(C+S)$, dan jika penulisan $(C+S)$ dihilangkan maka

$$\begin{aligned} W &< \frac{n}{p_1}(2p_1)+\frac{n}{p_1p_2}(2p_2)+\ldots+\frac{n}{p_1p_2\cdots p_k}(2p_k) \\ W &< 2n+\frac{2n}{p_1}+\ldots+\frac{2n}{p_1p_2\cdots p_{k-1}} \\ &< 2n+2n+\ldots+2n \\ &= 2kn. \end{aligned}$$

Terlihat bahwa $W < 2kn$, dengan demikian maka dapat dikatakan bahwa kompleksitas W = O(n) atau kompleksitas linier.

Proses Penyusunan Kode Sinyal Detail dan *Ciphertext*

Proses penyusunan kode sinyal detail dan penyusunan *ciphertext* pada max plus wavelet tipe A sama dengan tipe C, tipe D dan tipe E, sedangkan pada tipe B berbeda.

Algoritma proses penyusunan kode sinyal detail dan *ciphertext* pada max plus wavelet tipe A, C, D dan E dituliskan dalam bentuk pseudocode sebagai berikut:

Algoritma 4.11 Penyusunan Kode Sinyal Detail dan *Ciphertext* pada Tipe A, C, D dan E

```
Input : array 1 dimensi Y[] yang berukuran n-1
        (sinyal detail)
        variabel X (sinyal hampiran)
Output: array 1 dimensi CipherASCII[] yang berukuran n
        (Kode ASCII dari ciphertext)
        array 1 dimensi KodeSinyal[] yang berukuran n-1
        (Kode sinyal detail)
Proses:

for i=1:length(Y)
    if Y(i)<0 then
        KodeSinyal(i) = 1
    else
        KodeSinyal(i) = 0
    end if
    CipherASCII(i+1)= abs(Y(i))+32
end for
CipherASCII(1)= X
```

Variabel Y adalah sinyal detail yang dihasilkan dari proses analisis dengan panjang n-1. Operasi yang terdapat pada proses ini adalah operasi perbandingan, penjumlahan dan fungsi absolut. Dari pseudocode diatas terlihat bahwa setiap operasi tersebut dilakukan sebanyak $n-1$. Karena $n-1 < n$ maka dapat dikatakan bahwa kompleksitasnya adalah O(n) atau kompleksitas linier.

Algoritma proses penyusunan kode sinyal detail dan *ciphertext* pada max plus wavelet tipe B dituliskan dalam bentuk pseudocode sebagai berikut:

Algoritma 4.12 Penyusunan Kode Sinyal Detail dan *Ciphertext* pada Tipe B

```
Input : array 1 dimensi Y[] yang berukuran n-1
        (sinyal detail)
```

```
        variabel X (sinyal hampiran)
Output: array 1 dimensi CipherASCII[] yang berukuran n
        (Kode ASCII dari ciphertext)
        array 1 dimensi KodeSinyal[] yang berukuran n-1
        (Kode sinyal detail)
Proses:

for i=1:length(Y)
    if Y(i)< -126 then
        Y(i) = abs(Y(i))-95
        KodeSinyal(i) = 1
    else
        Y(i) = abs(Y(i))
        KodeSinyal(i) = 0
    end if
    CipherASCII(i+1) = Y(i)
end for
CipherASCII(1) = X
```

Variabel Y adalah sinyal detail dengan panjang n-1. Operasi yang terdapat pada proses ini adalah operasi perbandingan, pengurangan dan fungsi absolut. Dari pseudocode diatas terlihat bahwa operasi perbandingan dan fungsi absolut dilakukan sebanyak $n-1$. Sedangkan operasi penjumlahan bisa dilakukan kurang dari $n-1$. Karena $n-1 < n$ maka dapat dikatakan kompleksitasnya adalah O(n) atau kompleksitas linier.

Proses Penyusunan Sinyal Detail

Proses penyusunan sinyal detail ini dilakukan sebelum melakukan proses sintesis. Proses penyusunan sinyal detail pada max plus wavelet tipe A sama dengan tipe C, tipe D dan tipe E, sedangkan pada tipe B berbeda. Algoritma proses penyusunan sinyal detail pada max plus wavelet tipe A, C, D dan E dituliskan dalam bentuk pseudocode sebagai berikut:

Algoritma 4.13 Penyusunan Sinyal Detail pada Tipe A, C, D dan E

```
Input : array 1 dimensi CipherASCII[] yang berukuran n
        (Kode ASCII dari ciphertext)
        array 1 dimensi KodeSinyal[] yang berukuran n-1
        (Kode sinyal detail)
Output: array 1 dimensi Y[] yang berukuran n-1
        (sinyal detail)
Proses:

for i=2:n
    Y(i-1)=(CipherASCII(i)-32)×(-1)^KodeSinyal(i)
end for
```

Operasi yang terdapat pada proses ini adalah operasi pengurangan, perkalian dan perpangkatan. Dari pseudocode diatas terlihat bahwa setiap operasi tersebut dilakukan sebanyak $n-1$. Karena $n-1 < n$ maka dapat dikatakan kompleksitasnya adalah O(n) atau kompleksitas linier.

Algoritma proses penyusunan sinyal detail pada max plus wavelet tipe B dituliskan dalam bentuk pseudocode sebagai berikut:

Algoritma 4.14 Penyusunan Sinyal Detail pada Tipe B

```
Input : array 1 dimensi CipherASCII[] yang berukuran n
        (Kode ASCII dari ciphertext)
        array 1 dimensi KodeSinyal[] yang berukuran n-1
        (Kode sinyal detail)
Output: array 1 dimensi Y[] yang berukuran n-1
        (sinyal detail)
Proses:

for i=2:n
    Y(i-1)=(CipherASCII(i)+95×KodeSinyal(i))×(-1)
end for
```

Operasi yang terdapat pada proses ini adalah operasi penjumlahan dan perkalian. Dari pseudocode diatas terlihat bahwa operasi penjumlahan dilakukan sebanyak $n-1$ dan operasi perkalian dilakukan sebanyak $2(n-1)$. Karena $n-1 < n$ dan $2(n-1) < 2n$ maka dapat dikatakan kompleksitasnya adalah O(n) atau kompleksitas linier.

Dari analisis kompleksitas diatas terlihat bahwa kompleksitas waktu untuk proses enkripsi dan dekripsi adalah O(n) atau kompleksitas linier. Hasil ini sesuai dengan hasil yang didapatkan pada penghitungan *running time* seperti yang disajikan pada Gambar 4.4 dan Gambar 4.5.

Algoritma kriptografi max plus wavelet ini dalam prosesnya hanya melibatkan integer dan tidak melibatkan *floating point*, sehingga mempunyai beberapa kelebihan yaitu lebih sederhana, *running time* lebih cepat, penggunaan memori lebih kecil dan tidak menghasilkan error penghitungan.

4.3.5 Analisis *Key Space*

Penghitungan *key space* dilakukan untuk mengetahui banyaknya kemungkinan kunci dekripsi yang dapat digunakan. Semakin besar *key space* maka semakin banyak kemungkinan kunci yang digunakan. Sehingga untuk menemukan kunci yang sebenarnya menggunakan *brute force* (mencoba semua kemungkinan yang ada) akan semakin sulit.

Pada algoritma kriptografi max plus wavelet ini kunci dekripsi terdiri dari tiga bagian, yaitu tipe max plus wavelet, banyaknya kanal dan kode sinyal detail. Penghitungan *key space* dilakukan sebagai berikut.

- Terdapat lima tipe max plus wavelet yang digunakan pada algoritma ini, sehingga *key space* bagian pertama adalah 5.
- Kunci bagian kedua adalah banyaknya kanal yang digunakan. Kanal berasal dari semua faktor dari n kecuali 1, dimana n adalah banyaknya karakter pada *ciphertext*. Dari semua faktor ini kemudian dicari perkalian bilangan-bilangan yang hasilnya n. Sehingga *key space* bagian kedua ini adalah semua kemungkinan perkalian bilangan-bilangan yang hasilnya n.
- Kunci bagian ketiga adalah kode sinyal detail. Sinyal detail terdiri dari $n-1$ karakter dan setiap karakter mempunyai kode 0 atau 1. Sehingga *key space* bagian ketiga ini adalah 2^{n-1}.

Dari penghitungan tersebut didapatkan bahwa algoritma kriptografi max plus wavelet ini mempunyai *key space* sebesar $5 \times 2^{n-1} \times N$, dimana N adalah *key space* bagian kedua.

Sebagai contoh, penghitungan *key space* dari suatu *ciphertext* yang berukuran 30 karakter adalah sebagai berikut.

- *Key space* bagian pertama adalah 5 kemungkinan, karena ada lima tipe MP-Wavelet yang bisa digunakan.
- *Ciphertext* terdiri dari 30 karakter, sehingga keseluruhan kunci kanal yang mungkin digunakan adalah 2 3 5, 2 5 3, 3 2 5, 3 5 2, 5 2 3, 5 3 2, 2 15, 15 2, 3 10, 10 3, 5 6, 6 5 dan 30. Kunci kanal tersebut berasal dari faktor-faktor 30 kecuali 1. Terdapat 13 kemungkinan kunci kanal, sehingga *key space* bagian kedua ini adalah 13.
- Sinyal detail terdiri dari 29 karakter, sehingga *key space* bagian ketiga ini adalah 2^{29}.

Dari uraian tersebut didapatkan *key space* dari suatu *ciphertext* yang berukuran 30 karakter adalah $5 \times 13 \times 2^{29} = 34,896,609,280$.

Dari anaisis *key space* didapatkan bahwa algoritma kriptografi max plus wavelet ini mempunyai *key space* yang besar untuk menghadapi pemecahan kunci menggunakan *brute force*. Semakin panjang *ciphertext* maka semakin besar *key space* artinya semakin sulit untuk dipecahkan. Digunakannya lima tipe MP-Wavelet sangat menguntungkan karena memperbesar *key space*, meskipun masing-masing tipe tersebut mempunyai kelemahan sendiri-sendiri.

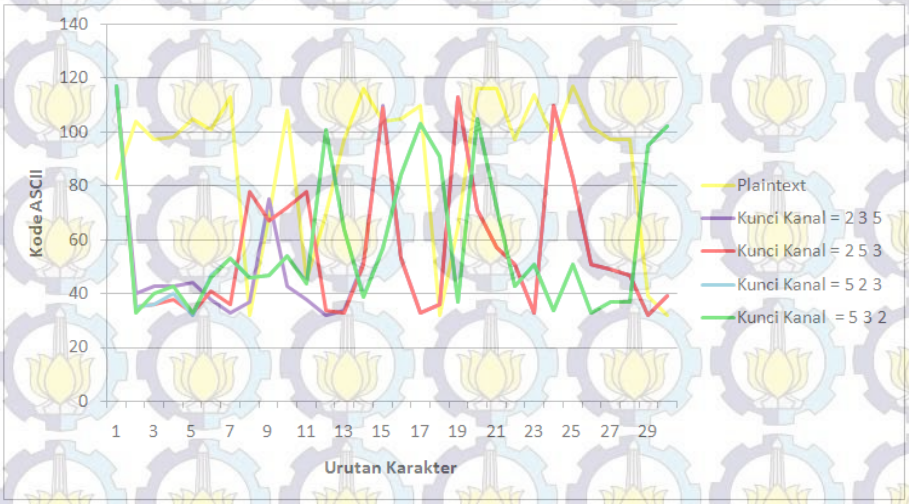
Kesulitan pemecahan kunci dekripsi pada algoritma kriptografi max plus wavelet ini terletak pada pemecahan kunci kanal. Karena harus menemukan faktor-faktor dari suatu bilangan n kecuali 1, dan kemudian menemukan semua kemungkinan perkalian bilangan-bilangan yang hasilnya n. Jika banyaknya karakter *ciphertext* merupakan bilangan prima, maka hanya ada satu kunci kanal yaitu bilangan itu sendiri. Hal ini akan menjadi mudah untuk memecahkan kuncinya. Karena itu disarankan untuk menggunakan kunci enkripsi yang akan menghasilkan *ciphertext* dengan banyak karakter bukan bilangan prima.

4.3.6 Pengaruh Kunci Kanal Terhadap Hasil Enkripsi

Untuk mengetahui pengaruh kunci kanal terhadap hasil enkripsi, dilakukan uji coba menggunakan file yang sama dan tipe max plus wavelet yang sama. Sedangkan kunci kanal yang digunakan berbeda. Sebagai contoh dilakukan uji coba menggunakan file uji kanal.txt dengan max plus wavelet tipe A. Kunci kanal yang digunakan adalah 2 3 5, 2 5 3, 5 2 3 dan 5 3 2. Data hasil uji coba tersebut disajikan pada Tabel 4.6, sedangkan perbedaan antara *plaintext* dan *ciphertext* disajikan pada Gambar 4.6.

Tabel 4.6: Hasil Uji Coba Kunci Kanal

Kunci Kanal	Korelasi	Kualitas
2 3 5	0.1265434	0.4842105
2 5 3	-0.0493494	0.5052632
5 2 3	-0.2631666	0.5052632
5 3 2	-0.263015	0.5263158



Gambar 4.6: Perbedaan Karakter Plaintext dan Ciphertext

Dari Tabel 4.6 diketahui bahwa nilai mutlak korelasi kurang dari 0.3. Ini menunjukkan bahwa korelasi antara *plaintext* dan *ciphertext* adalah lemah. Perbedaan kualitas enkripsi tidak begitu besar yaitu kurang dari 0.05.

Dari Gambar 4.6 terlihat bahwa hasil enkripsi menggunakan kunci 2 3 5 hampir sama dengan hasil enkripsi menggunakan kunci 2 5 3. Begitu juga hasil enkripsi menggunakan kunci 5 2 3 hampir sama dengan hasil enkripsi menggunakan kunci 5 3 2. Perbedaan hanya terjadi di awal *ciphertext*, yaitu sebanyak n/kunci kanal pertama, sedangkan sisanya memiliki karakter yang sama. Hal ini disebabkan kunci kanal yang pertama adalah sama. Sehingga dapat diambil kesimpulan bahwa dengan menggunakan tipe max plus wavelet yang sama maka kunci kanal yang pertama adalah yang paling berpengaruh terhadap hasil enkripsi. Jika kunci kanal yang pertama berbeda maka hasil enkripsi juga banyak perbedaan.

BAB V
KESIMPULAN DAN SARAN

Dari konstruksi dan analisis algoritma kriptografi max plus wavelet yang sudah dilakukan, dapat diambil kesimpulan serta diberikan saran untuk perbaikan dan pengembangan penelitian selanjutnya.

5.1 Kesimpulan

Kesimpulan yang dapat diambil dari pengerjaan tesis ini adalah:

1. Pada tesis ini berhasil dikonstruksi algoritma kriptografi berdasarkan transformasi max plus wavelet. Transformasi max plus wavelet yang digunakan adalah tipe A, B, C, D dan E. Algoritma ini termasuk dalam algoritma kriptografi *stream cipher*. Sedangkan berdasarkan jenis kunci termasuk kriptografi asimetris, karena kunci enkripsi berbeda dengan kunci deripsi. Kunci enkripsi terdiri dari dua bagian yaitu tipe max plus wavelet dan kanal yang digunakan. Kunci dekripsi terdiri dari kunci enkripsi ditambah dengan kode sinyal detail. Proses enkripsi didasarkan pada proses analisis, sedangkan proses dekripsi didasarkan pada proses sintesis.

2. Berdasarkan analisis hasil uji coba diketahui bahwa algoritma kriptografi ini mempunyai nilai korelasi *plaintext* dan *ciphertext* yang kecil, artinya *plaintext* dan *ciphertext* mempunyai hubungan linier yang sangat rendah. Algoritma ini juga memiliki nilai kualitas enkripsi yang baik. Hal-hal tersebut menunjukkan bahwa algoritma ini mempunyai enkripsi yang baik.

3. Dari penghitungan *running time* dan analisis kompleksitas diketahui bahwa kompleksitas algoritma ini adalah O(n) atau kompleksitas linier. Hasil tersebut menunjukkan bahwa Algoritma kriptografi ini efisien dalam hal waktu proses.

4. Algoritma kriptografi ini dalam prosesnya hanya melibatkan integer dan tidak melibatkan *floating point*, sehingga mempunyai beberapa kelebihan yaitu lebih sederhana, *running time* lebih cepat, penggunaan memori lebih kecil dan tidak menghasilkan error penghitungan.

5. Penggunaan kelima tipe MP-Wavelet dapat memperbesar *key space* sehingga semakin banyak kemungkinan kunci dan semakin sulit untuk menemukan kunci yang sebenarnya. Kunci kanal juga mempersulit upaya pemecahan kunci karena harus menemukan faktor-faktor dari suatu bilangan dan semua kemungkinan kanal yang digunakan. Disarankan menggunakan kunci enkripsi yang akan menghasilkan *ciphertext* dengan banyak karakter bukan bilangan prima agar kunci sulit untuk ditebak.

6. Kunci kanal pertama mempunyai pengaruh yang besar terhadap hasil enkripsi. Dengan menggunakan tipe max plus wavelet yang sama, jika kunci kanal pertama sama maka hasil enkripsi sebagian besar adalah sama. Jika kunci kanal pertama berbeda maka hasil enkripsi juga memiliki banyak perbedaan.

5.2 Saran

Saran-saran yang dapat diberikan setelah pengerjaan tesis ini adalah:

1. Banyaknya karakter yang digunakan diperbanyak, tidak hanya terbatas 95 karakter.

2. Proses enkripsi dilakukan lebih dari satu kali, sehingga akan lebih mengaburkan *ciphertext*.

3. Sebelum dilakukan proses analisis, *plaintext* diacak terlebih dahulu. Hal ini juga akan lebih mengaburkan *ciphertext*.

4. Penggunaan transformasi max plus wavelet untuk algoritma kriptografi citra, audio dan lain sebagainya.

DAFTAR PUSTAKA

Aruljothi, S., Venkatesulu, M., (2012), "Encryption Quality and Performance Analysis of GKSBC Algorithm", *Journal of Information Engineering and Applications*, Vol. 2, No. 10.

Boggess, A., Norcowich, F. J., (2001), *A First Course In Wavelets With Fourier Analysis*, Prentice-Hall, Inc., New Jersey.

Durcheva, Mariana,(2015) "Some applications of idempotent semirings in Public Key Cryptography", ACM Communication in Computer Algebra, hal. 19.

Fahim, Kistosil, (2014), *Konstruksi Transformasi Wavelet Menggunakan Aljabar Max Plus*, Tesis, Jurusan Matematika FMIPA Institut Teknologi Sepuluh Nopember, Surabaya.

Goswami, D., Rahman, N., Biswas, J., Koul, A., Tamang, R.L., Bhattacharjee, A.K., (2011), "A Discrete Wavelet Transform based Cryptographic algorithm", *International Journal of Computer Science and Network Security*, Vol. 11, No. 4.

Grigoriev, D., Shpilrain, V., (2013), "Tropical Cryptography", International Association for Cryptologic Research.

Kromodimoeljo, S., (2010), *Teori Dan Aplikasi Kriptografi*, SPK IT Consulting.

Rosen, K. H., (2012), *Discrete Mathematics and Its Applications, Seventh Edition*, The McGraw-Hill Companies, New York.

Sarwono, J., (2006), *Metode Penelitian Kuantitatif Dan Kualitatif*, Graha Ilmu, Yogyakarta.

Subiono, (2015), *Aljabar Min Max Plus Dan Terapannya*, Institut Teknologi Sepuluh Nopember, Surabaya.

Walpole, R. E., (1982), *Pengantar Statistik*, PT. Gramedia Pustaka Utama, Jakarta.

LAMPIRAN

LAMPIRAN A
Koding Program Kriptografi Max Plus Wavelet

\\Fungsi analisis dan sintesis max plus wavelet

```
function [X,Y] = MPW_Analisis_A (sinyal,p)
k = 0
for j=1:(length(sinyal)/p)
   n = p*(j-1)+1
   m = (p-1)*(j-1)+1
   X(j) = max(sinyal(n:n+p-1))
   Y(m:m+p-2) = sinyal(n+1:n+p-1)-sinyal(n)
end
endfunction


function [X,Y] = MPW_Sintesis_A (sinyal_X, sinyal_Y,p)
k = length(sinyal_X)
for j=0:k-1
   maks = max(sinyal_Y(1:p-1))
   maks = max(maks, 0)
   X(j*p+1) = sinyal_X(j+1)- maks
   X(j*p+2:j*p+p) = X(j*p+1) + sinyal_Y(1:p-1)
   n= length(sinyal_Y)
   sinyal_Y=sinyal_Y(p:n)
end
Y = sinyal_Y
endfunction


function [X,Y] = MPW_Analisis_B (sinyal,p)
k = 0
for j=1:(length(sinyal)/p)
   n = p*(j-1)+1
   maks = sinyal(n)
```

```
   for i=1:p-1
      maks = max(maks, sinyal(n+i))
      k = k + 1
      Y_a(k) = sinyal(n+i)-sinyal(n)
   end
   X_a(j) = maks
end
for j=1:(length(sinyal)/p)
   maks = X_a(j)
   for i=1:p-1
      maks = max(maks, Y_a((p-1)*(j-1)+i))
      Y((p-1)*(j-1)+i) = Y_a((p-1)*(j-1)+i)-X_a(j)
   end
   X(j) = maks
end
endfunction


function [X,Y] = MPW_Sintesis_B (sinyal_X,sinyal_Y,p)
k = length(sinyal_X)
y = length(sinyal_Y)
for j=0:k-1
   maks = 0
   for i=1:p-1
      maks = max(maks, sinyal_Y(j*(p-1)+i))
   end
   X_b(j+1) = sinyal_X(j+1)- maks
   for i=1:p-1
      Y_b(j*(p-1)+i) = X_b(j+1) + sinyal_Y(j*(p-1)+i)
   end
end
for j=0:k-1
   maks = 0
   for i=1:p-1
      maks = max(maks, Y_b(i))
   end
   X(j*p+1) = X_b(j+1)- maks
```

```
    for i=1:p-1
        X(j*p+1+i) = X(j*p+1) + Y_b(i)
    end
    n = length(Y_b)
    Y_b = Y_b(p:n)
end
Y = sinyal_Y(k*(p-1)+1:y)
endfunction


function [X,Y] = MPW_Analisis_C (sinyal, p)
k=0
q = sqrt(p)
for j=1:(length(sinyal)/p)
    for m = 0:q-1
        n = p*(j-1)+q*m+1
        maks = sinyal(n)
        for i=1:q-1
            maks = max(maks, sinyal(n+i))
            X_a(n+i) = sinyal(n+i)-sinyal(n)
        end
        X_a(n) = maks
    end
end
for j=1:(length(sinyal)/p)
    for m = 1:q
        n = p*(j-1)+m
        maks = X_a(n)
        for i=1:q-1
            maks = max(maks, X_a(n+i*q))
            X_b(p*(j-1)+q*(m-1)+1+i)=X_a(n+i*q)-X_a(n)
        end
        X_b(p*(j-1)+q*(m-1)+1) = maks
    end
    X(j) = X_b((j-1)*p+1)
    for i=1:p-1
        k=k+1
```

```
      Y(k) = X_b((j-1)*p+i+1)
   end
end
endfunction


function [X,Y] = MPW_Sintesis_C (sinyal_X,sinyal_Y,p)
k = length(sinyal_X)
y = length(sinyal_Y)
q = sqrt(p)
n = 0
for j=0:k-1
   D(j*p+1)= sinyal_X(j+1)
   for m=1:p-1
      n = n+1
      D(j*p+1+m)= sinyal_Y(n)
   end
end
for j=0:k-1
   for m=0:q-1
      maks = 0
      for i=1:q-1
         maks = max(maks, D(j*p+1+m*q+i))
      end
      D_a(j*p+1+m) = D(j*p+1+m*q) - maks
      for i=1:q-1
         D_a(j*p+1+m+i*q)=D_a(j*p+1+m)+D(j*p+1+m*q+i)
      end
   end
end
for j=0:k-1
   for m=0:q-1
      maks = 0
      for i=1:q-1
         maks = max(maks, D_a(j*p+1+m*q+i))
      end
      X(j*p+1+m*q) = D_a(j*p+1+m*q) - maks
```

```
        for i=1:q-1
            X(j*p+1+m*q+i)=X(j*p+1+m*q)+D_a(j*p+1+m*q+i)
        end
    end
end
Y = sinyal_Y(k*(p-1)+1:y)
endfunction


function [X,Y] = MPW_Analisis_D (sinyal,s,t)
k=0
for j=1:(length(sinyal)/(s*t))
    for m = 1:s
        n = s*t*(j-1)+t*(m-1)+1
        maks = sinyal(n)
        for i=1:t-1
            maks = max(maks, sinyal(n+i))
            X_a(n+i) = sinyal(n+i)-sinyal(n)
        end
        X_a(n) = maks
    end
end
for j=1:(length(sinyal)/(s*t))
    for m = 1:t
        n = s*t*(j-1)+m
        maks = X_a(n)
        for i=1:s-1
            maks = max(maks, X_a(n+i*t))
            X_b(n+i*t) = X_a(n+i*t) - X_a(n)
        end
        X_b(n) = maks
    end
    X(j) = X_b((j-1)*s*t+1)
    for i=1:s*t-1
        k=k+1
        Y(k) = X_b((j-1)*s*t+i+1)
    end
```

```
end
endfunction


function [X,Y] = MPW_Sintesis_D (sinyal_X,sinyal_Y,s,t)
k = length(sinyal_X)
y = length(sinyal_Y)
n = 0
for j=0:k-1
  D(j*s*t+1)= sinyal_X(j+1)
  for m=1:s*t-1
    n = n+1
    D(j*s*t+1+m)= sinyal_Y(n)
  end
end
for j=0:k-1
  for m=1:t
    n = j*s*t + m
    maks = 0
    for i=1:s-1
      maks = max(maks, D(n + i*t))
    end
    D_a(n) = D(n) - maks
    for i=1:s-1
      D_a(n + i*t) = D_a(n) + D(n + i*t)
    end
  end
end
for j=0:k-1
  for m=0:s-1
    n = j*s*t + m*t +1
    maks = 0
    for i=1:t-1
      maks = max(maks, D_a(n+i))
    end
    X(n) = D_a(n) - maks
    for i=1:t-1
```

```
        X(n+i)= X(n)+ D_a(n+i)
      end
   end
end
Y = sinyal_Y(k*(s*t-1)+1:y)
endfunction


function [X, Y] = MPW_Analisis_E (sinyal, p)
k = 0
for j=1:(length(sinyal)/p)
   n = p*(j-1)+1
   maks = sinyal(n)
   for i=1:p-1
      maks = max(maks, sinyal(n+i))
      k = k + 1
      Y_a(k) = sinyal(n+i)-sinyal(n)
   end
   X(j) = maks
end
for j=1:(length(sinyal)/p)
   n = (p-1)*(j-1)+1
   maks = Y_a(n)
   for i=1:p-2
      maks = max(maks, Y_a(n+i))
      Y(n+i) = Y_a(n+i)-Y_a(n)
   end
   Y(n) = maks
end
endfunction


function [X,Y]=MPW_Sintesis_E(sinyal_X,sinyal_Y,p)
k = length(sinyal_X)
y = length(sinyal_Y)
for j=0:k-1
   n = (p-1)*j+1
   maks = 0
```

```
  for i=1:p-2
     maks = max(maks, sinyal_Y(n+i))
  end
  Y_b(n) = sinyal_Y(n)-maks
  for i=1:p-2
     Y_b(n+i) = Y_b(n) + sinyal_Y(n+i)
  end
end
for j=0:k-1
  n = p*j+1
  maks = 0
  for i=1:p-1
     maks = max(maks, Y_b(j*(p-1)+i))
  end
  X(n) = sinyal_X(j+1)- maks
  for i=1:p-1
     X(n+i) = X(n) + Y_b(j*(p-1)+i)
  end
end
Y = sinyal_Y(k*(p-1)+1:y)
endfunction
```

\\Fungsi Enkripsi Kriptografi

```
function PbEnkripsi_callback(handles)
N_Text = 1
for i=1:length(Key_Kanal)
  N_Text = N_Text * Key_Kanal(i)
end
PlainASCII = ascii(Plaintext)
X = PlainASCII
k = length(X)
for i=1:k-1
  Y(i)=0
end
if N_Text > k then
  for j=k+1:N_Text
```

```
      X(j)=32
      Y(j-1)=0
   end
end
PlainASCII = X
k=N_Text
if Key_MPW == 1 then
   for i=1:length(Key_Kanal)
      [X, d] = MPW_Analisis_A (X, Key_Kanal(i))
      n=length(d)
      for j=n:-1:1
         k=k-1
         Y(k)=d(j)
      end
   end
elseif Key_MPW == 2 then
   for i=1:length(Key_Kanal)
      [X, d] = MPW_Analisis_B (X, Key_Kanal(i))
      n=length(d)
      for j=n:-1:1
         k=k-1
         Y(k)=d(j)
      end
   end
elseif Key_MPW == 3 then
   for i=1:length(Key_Kanal)
      [X, d] = MPW_Analisis_C (X, Key_Kanal(i))
      n=length(d)
      for j=n:-1:1
         k=k-1
         Y(k)=d(j)
      end
   end
elseif Key_MPW == 4 then
   for i=1:length(Key_Kanal)/2
      [X,d]=MPW_Analisis_D(X,Key_Kanal(2*i-1),...
```

```
        Key_Kanal(2*i))
      n=length(d)
      for j=n:-1:1
        k=k-1
        Y(k)=d(j)
      end
    end
elseif Key_MPW == 5 then
    for i=1:length(Key_Kanal)
      [X, d] = MPW_Analisis_E (X, Key_Kanal(i))
      n=length(d)
      for j=n:-1:1
        k=k-1
        Y(k)=d(j)
      end
    end
end
Key_Sinyal=blanks(0)
if Key_MPW == 2 then
    for i=1:length(Y)
      if Y(i) < -126 then
        Y(i) = abs(Y(i))-95
        Key_Sinyal=strcat([Key_Sinyal,"1"])
      else
        Y(i) = abs(Y(i))
        Key_Sinyal=strcat([Key_Sinyal,"0"])
      end
    end
    ChiperASCII(1)=X(1)
    for i=2:N_Text
      ChiperASCII(i)= Y(i-1)
    end
else
    ChiperASCII(1)=X(1)
    for i=2:N_Text
      ChiperASCII(i)= abs(Y(i-1))+32
```

```
    end
    for i=1:length(Y)
       if Y(i)<0 then
          Key_Sinyal=strcat([Key_Sinyal,"1"])
       else
          Key_Sinyal=strcat([Key_Sinyal,"0"])
       end
    end
end
Chipertext= ascii(ChiperASCII)
N=ceil(length(Key_Sinyal)/8)
m= modulo (length(Key_Sinyal),8)
if m==0 then m=8
end
for i=1:N-1
    KeyNum(i)=bin2dec(part(Key_Sinyal,i*8-7:i*8))
end
KeyNum(N)=bin2dec(part(Key_Sinyal,...
    N*8-7:length(Key_Sinyal)))
endfunction
```

\\Fungsi Dekripsi Kriptografi

```
function PbDekripsi_callback(handles)
ChiperASCII= ascii(Chipertext)
N_Text = length(ChiperASCII)
Key_MPW = Key(1)
k = N_Text
i=1
while k > 1
    i=i+1
    k = k/Key(i)
    Key_Kanal(i-1) = Key(i)
end
Key_Sinyal=blanks(0)
for j = i+1:length(Key)-1
    Key_Sinyal=strcat([Key_Sinyal, dec2bin(Key(j),8)])
```

```
end
m = modulo(N_Text-1,8)
if m==0 then m=8
end
Key_Sinyal=strcat([Key_Sinyal, ...
   dec2bin(Key(length(Key)),m)])
X_dekrip = ChiperASCII(1)
if Key_MPW == 1 then
   for i=2:N_Text
      s=ascii(part(Key_Sinyal,i-1))-48
      Y_dekrip(i-1)=(ChiperASCII(i)-32)*(-1)^s
   end
   n=length(Key_Kanal)
   for i=n:-1:1
      [X_dekrip,Y_dekrip]=MPW_Sintesis_A...
         (X_dekrip,Y_dekrip,Key_Kanal(i))
   end
elseif Key_MPW == 2 then
   for i=2:N_Text
      s=ascii(part(Key_Sinyal,i-1))-48
      Y_dekrip(i-1)=ChiperASCII(i)
      Y_dekrip(i-1)= Y_dekrip(i-1)-32
      Y_dekrip(i-1)= Y_dekrip(i-1)+95*s
      Y_dekrip(i-1)= Y_dekrip(i-1)+32
      Y_dekrip(i-1)= Y_dekrip(i-1)*(-1)
   end
   n=length(Key_Kanal)
   for i=n:-1:1
      [X_dekrip,Y_dekrip]=MPW_Sintesis_B...
         (X_dekrip,Y_dekrip,Key_Kanal(i))
   end
elseif Key_MPW == 3 then
   for i=2:N_Text
      s=ascii(part(Key_Sinyal,i-1))-48
      Y_dekrip(i-1)=(ChiperASCII(i)-32)*(-1)^s
   end
```

```
    n=length(Key_Kanal)
    for i=n:-1:1
       [X_dekrip,Y_dekrip]=MPW_Sintesis_C...
         (X_dekrip,Y_dekrip,Key_Kanal(i))
    end
elseif Key_MPW == 4 then
    for i=2:N_Text
       s=ascii(part(Key_Sinyal,i-1))-48
       Y_dekrip(i-1)=(ChiperASCII(i)-32)*(-1)^s
    end
    n=length(Key_Kanal)/2
    for i=n:-1:1
       [X_dekrip,Y_dekrip]=MPW_Sintesis_D(X_dekrip,...
       Y_dekrip,Key_Kanal(2*i-1), Key_Kanal(2*i))
    end
elseif Key_MPW == 5 then
    for i=2:N_Text
       s=ascii(part(Key_Sinyal,i-1))-48
       Y_dekrip(i-1)=(ChiperASCII(i)-32)*(-1)^s
    end
    n=length(Key_Kanal)
    for i=n:-1:1
       [X_dekrip,Y_dekrip]=MPW_Sintesis_E...
         (X_dekrip,Y_dekrip,Key_Kanal(i))
    end
end
Plaintext=ascii(abs(modulo(X_dekrip,255)))
endfunction
```

BIODATA PENULIS

Penulis bernama lengkap Joko Cahyono, lahir di Malang pada tanggal 27 Agustus 1981. Penulis adalah anak kelima dari lima bersaudara dari pasangan Bapak Moch. Rokim dan Ibu Sumilah. Penulis menikah dengan Lestin Tatik Julaika pada tahun 2010 dan dikaruniai seorang anak yang diberi nama Shabieq El-Fathin Attaraufaa'. Penulis menempuh pendidikan formal Sekolah Dasar di SLB/D YPAC Malang, dilanjutkan dengan SMPN 3 Malang dan SMUN 3 Malang. Penulis masuk di Jurusan Matematika FMIPA ITS pada tahun 2000 untuk menempuh pendidikan S1 dan memilih bidang minat Informatika. Penulis lulus S1 pada tahun 2005 dengan judul skripsi *"Implementasi Algoritma Genetika Paralel Untuk Traveling Salesman Problem Menggunakan Message Passing Interface"*. Pada tahun 2014 penulis memulai pendidikan S2 di Jurusan Matematika FMIPA ITS.