PERANCANGAN UNTAI PENCARI POLINOMIAL LOKASI KESALAHAN MENGGUNAKAN ALGORITMA BERLEKAMP-MASSEY UNTUK SANDI BCH (15,5) YANG EFISIEN BERBASIS FPGA

MAKALAH

FRANSISKA
98/121046/TK/22764

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2002

PERANCANGAN UNTAI PENCARI POLINOMIAL LOKASI KESALAHAN MENGGUNAKAN ALGORITMA BERLEKAMP-MASSEY UNTUK SANDI BCH (15,5) YANG EFISIEN BERBASIS FPGA

INTISARI

Pencarian polinomial kesalahan adalah bagian terpenting dari sistem pengawasandian dalam *error control coding*. Polinomial lokasi kesalahan akan memberikan keluaran yang dapat diproses lebih lanjut untuk menentukan lokasi galat yang terjadi. Salah satu sandi yang banyak digunakan dalam pengendalian galat adalah sandi BCH. FPGA (*Field Programmable Gate Array*) adalah keping yang dapat diprogram ulang. Perancangan dengan FPGA relatif cepat, mudah untuk di ubah serta berkapasitas besar. Tujuan tugas akhir ini adalah untuk merancang untai elektronik digital untuk pencarian polinomial lokasi kesalahan untuk sandi BCH (15,5) dengan jumlah bit data = 5 dari seluruh 15 bit sandi terkirim dan kesalahan maksimum yang dapat dikoreksi = 3 serta mengimplementasikannya ke dalam sebuah keping FPGA XC4013.

Cara pencarian polinomial lokasi kesalahan yang paling banyak dipakai adalah Algoritma Berlekamp-Massey (BMA). Kelebihan BMA adalah sifatnya yang efisien, berdasarkan prinsip iterasi untuk menemukan satu persatu kesalahan yang terjadi. BMA Iterasi Cepat adalah metode yang lebih efisien dari BMA. BMA Iterasi Cepat mampu mendeteksi lokasi kesalahan sampai dengan 3 kesalahan dengan 3 iterasi.

Untai BMA yang dirancang dipisahkan untuk tiap langkah iterasi. Proses perancangan untai digitalnya menggunakan perangkat lunak OrCAD. Kemudian hasil rancangan diimplementasikan ke dalam sebuah keping FPGA XC4013.

Hasil pengamatan menunjukkan bahwa rancangan untai digital BMA tersebut bekerja dengan baik saat simulasi. Hasil rancangan tersebut berhasil direalisasikan ke dalam sebuah keping FPGA XC 4013 dan membutuhkan 132 CLB dari total 576 CLB atau sebesar 22 % dari CLB yang tersedia. Hasil implementasi menunjukkan bahwa untai tersebut bekerja dengan benar. Untai dapat mendeteksi kesalahan sebanyak 3-bit, 2-bit dan 1-bit pada sembarang posisi dalam 15 bit kata sandi. Untai juga mampu menangani runtun pesan tanpa galat.

I. PENDAHULUAN

Proses penyandian dalam komunikasi memerlukan bagian penyandi dan pengawasandi. Dalam kenyataannya sandi yang dikirimkan melalui media transmisi dari penyandi ke pengawasandi rentan terkena derau, sehingga pesan

yang diterima di penerima berbeda dengan pesan awal yang dikirim. Oleh sebab itu dalam bagian pengawasandi diperlukan suatu unit pengoreksi kesalahan yang dapat mendeteksi adanya kesalahan pada sandi yang dikirim dan menentukan letak kesalahannya.

Salah satu bagian unit pengoreksi kesalahan disebut BMA (*Berlekamp-Massey Algorithm*) yang berfungsi untuk menentukan polinomial pencari lokasi kesalahan. Untai BMA memegang peranan penting dalam sistem penyandian, khususnya pada untai pengawasandi, karena membutuhkan hampir 69% perangkat keras yang digunakan oleh keseluruhan sistem. Tugas akhir ini akan mencoba untuk merancang dan mengimplementasi suatu rancangan untai BMA untuk sandi biner yang efisien dilihat dari jumlah gerbang yang digunakan dan waktu komputasi yang diperlukan.

Penggunaan FPGA dalam sistem penyandian untuk pengaturan kesalahan (*error control coding*) semakin banyak ditemukan karena kelebihan–kelebihannya. Jika dahulu pengaturan kesalahan penyandian dengan algoritma yang ada hanya bisa dilakukan lewat pemrograman komputer, sekarang dengan adanya FPGA sudah bisa diimplementasikan langsung dengan perangkat keras.

Pada tahun 1985 , perusahaan semikonduktor Xilinx memperkenalkan suatu teknologi IC yang dinamakan FPGA (*Field Programmable Gate Array*). FPGA adalah suatu konsep teknologi IC yang dapat diprogram dan dihapus seperti halnya RAM. FPGA kemudian berkembang pesat, baik dari segi kepadatan gerbang, kecepatan dan disertai dengan penurunan harga.

Berdasarkan beberapa kelebihan yang dimiliki FPGA tersebut maka penyandian untuk pengaturan kesalahan dicoba untuk diimplementasikan secara perangkat keras, karena dalam penerapannya komputasi berbasis matematika pada medan terbatas (Medan Galois) yang digunakan dalam penyandian akan sangat rumit bila dioperasikan dalam mikroprosesor biasa.

II. TINJAUAN PUSTAKA

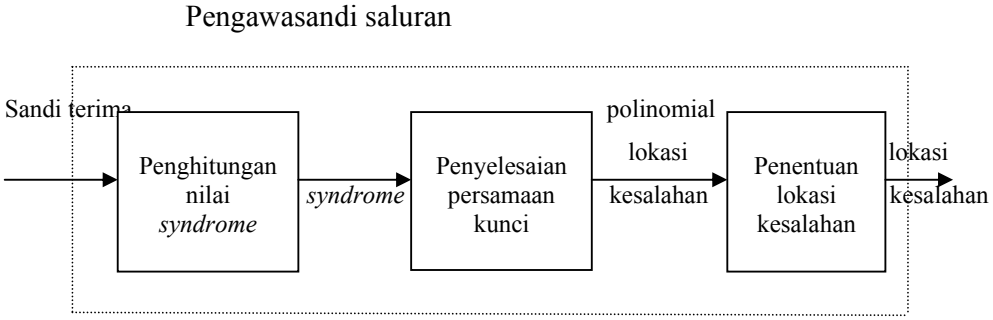
Algoritma Iteratif untuk menemukan polinomial lokasi kesalahan (Lin dan Costello, 1983). Langkah pertama untuk menemukan suatu polinomial berderajat minimum yang koefisien-koefisiennya memenuhi persyaratan Identitas Newton pertama. Langkah selanjutnya untuk mengecek apakah koefisien yang ditemukan sudah memenuhi persamaan Newton kedua. Jika belum memenuhi, maka persamaannya harus dikoreksi.

III.DASAR TEORI

Pengawasandi saluran memiliki masukan berupa kata terima *r* yang bisa jadi telah terkena derau pada saat pengiriman maupun penyimpanan. Rangkaian pengawasandi selain harus menerjemahkan sandi terima menjadi sandi pesan semula juga harus dapat mengoreksi kesalahan, sehingga rangkaian pengawasandi jauh lebih rumit dibandingkan dengan penyandi, karena memiliki untai untuk mengoreksi derau yang mengenainya.

Dalam suatu proses pengawasandian terdapat tiga tahapan penting, yaitu penghitungan *syndrome*, mencari konstanta polinomial kesalahan (algoritma

Berlekamp-Massey) yang akan dibahas secara lebih spesifik dalam tugas akhir ini dan mencari lokasi galat (*Chien search*).



Gambar 2.2. Diagram kotak pengawasandi saluran

Persamaan kunci adalah polinomial lokasi kesalahan *(error locator polynomial)* $\sigma(x)$, yang ditampilkan sebagai $\sigma(x) = \sigma_0 + \sigma_1 x + ... + \sigma_t x^t$ dari nilai-nilai *syndrome* yang telah dicari sebelumnya.

Masukan dari untai ini berupa nilai-nilai *syndrome* yang dilambangkan sebagai S_k, k adalah integer yang nilainya $1 \le k \le 2t$, dengan t adalah jumlah kesalahan maksimum yang dapat dikoreksi. Keluarannya berupa polinomial $\sigma(x)$ dengan koefisien-koefisien tertentu.

Jika kesalahan yang terjadi pada kata terima $r(x)$ dinyatakan sebagai v dengan $v \le t$, maka polinomial lokasi kesalahan $\sigma(x)$ yang dapat dibentuk adalah

$$\sigma(x) = \sigma_0 + \sigma_1 x + \sigma_2 x^2 + ... + \sigma_v x^v$$

$$= (1 + \beta_1 x)(1 + \beta_2 x)...(1 + \beta_v x)$$

Fungsi tersebut sangat berkaitan dengan komponen-komponen *syndrome* S_k , $1 \le k \le v+1$, berdasarkan hubungan identitas Newton berikut

$$S_1 + \sigma_1 = 0$$
$$S_2 + \sigma_1 S_1 + 2\sigma_2 = 0$$
$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0$$
$$...$$
$$S_v + \sigma_1 S_{v-1} + ... + \sigma_{v-1} S_1 + v\sigma_v = 0$$

Hubungan antara *syndrome* dengan nilai-nilai koefisien σ_i ditunjukkan oleh persamaan $\sum_{i=0}^{t} s_{t+j-i}\sigma_i = 0; \quad (i = 1,...,t)$ dan akar-akar $\sigma(x)$ memberikan informasi tentang posisi galat.

Dengan menggunakan identitas Newton pada persamaan (2.9), dapat dilihat bahwa σ_i , $0 \le i \le v$ mempunyai hubungan tertutup dengan komponen-komponen *syndrome* S_k, , $1 \le k \le v+1$. Oleh karena itu, jika σ_i dapat ditentukan, maka nilai-nilai lokasi galat $\beta_l = \alpha^{jl}$, $1 \le l \le v$, dapat pula ditemukan dengan cara mencari akar-akar $\sigma(x)$ berderajat minimal. $\sigma(x)$ ini akan menghasilkan pola galat

$e(x)$ dengan sebuah nilai minimum dari galat-galat tersebut. Jika $v \leq t$ galat terjadi, maka $\sigma(x)$ akan memberikan pola galat $e(x)$.

Diantara berbagai algoritma untuk mencari polinomial lokasi kesalahan, yang paling populer adalah algoritma Berlekamp-Massey (BMA) karena sifatnya yang efisien. Pada BMA, polinomial lokasi kesalahan dicari dengan prinsip iterasi, pada tiap tahap, derajat $\sigma(x)$ biasanya bertambah satu. Sehingga diperoleh polinomial lokasi kesalahan yang berderajat minimal. Dengan cara ini, derajat $\sigma(x)$ akan menunjukan jumlah kesalahan yang terjadi. Kerumitan BMA disebabkan karena proses penghitungan koefisien-koefisien $\sigma(x)$ secara iteratif sesuai dengan jumlah kesalahan yang akan dikoreksi.

Langkah pertama iterasi adalah menemukan polinomial berderajat minimal $\sigma^{(1)}(x)$ yang memenuhi persamaan Newton pertama. Kemudian dilakukan pengetesan apakan koefiesien $\sigma^{(1)}(x)$ juga memenuhi persamaan Newton kedua. Jika memenuhi maka diasumsikan $\sigma^{(2)}(x) = \sigma^{(1)}(x)$.

Jika koefisiennya tidak memenuhi, maka koreksi dilakukan terhadap $\sigma^{(1)}(x)$ untuk membentuk $\sigma^{(2)}(x)$ sehingga $\sigma^{(2)}(x)$ berderajat minimum dan koefisiennya memenuhi kedua persamaan Newton pertama. Demikian juga untuk persamaan selanjutnya. Iterasi dilanjutkan sampai diperoleh persamaan $\sigma^{(2t)}(x)$.

Selanjutnya $\sigma^{(2t)}(x)$ disebut sebagai polinomial lokasi kesalahan $\sigma(x)$ yang akan menghasilkan pola galat dalam bobot minimum yang sesuai dengan masukan *syndrome*-nya.

Dengan jumlah kesalahan maksimum yang dapat dikoreksi adalah t maka cacah iterasi pada BMA dapat dibatasi sampai $2t$, untuk operasi pengecekan dan pengoreksian.

Untuk melakukan koreksi pada iterasi ke μ, diperkenalkan suatu perhitungan yang disebut *discrepancy* keμ, d_μ sebagai $d_\mu = S_\mu + \sigma_1^{(\mu)} S_{\mu-1} + \ldots + \sigma_{l_\mu}^{(\mu)} S_{\mu-l_\mu}$ Jika $d_\mu = 0$, maka koefisien $\sigma^{(\mu)}(x)$ memenuhi persamaan Newton ke $(\mu+1)$. Sehingga dapat dianggap $\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x)$. Tetapi jika $d_\mu \neq 0$, maka koefisien $\sigma^{(\mu)}(x)$ tidak memenuhi persamaan Newton ke $(\mu+1)$ sehingga persamaan $\sigma^{(\mu)}(x)$ harus dikoreksi untuk memperoleh persamaan $\sigma^{(\mu+1)}(x)$.

Pengoreksian dilakukan dengan berbalik ke langkah iterasi sebelum iterasi ke μ dan menentukan polinomial $\sigma^{(\mu)}(x)$ sedemikian sehingga *discrepancy* ke ρ, $d_\rho \neq 0$ dan ρ-l_ρ (l_ρ adalah derajat $\sigma^{(\rho)}(x)$ yang terbesar). Sehingga polinomial berderajat minimum yang koefisien-koefisiennya memenuhi $(\mu+1)$ persamaan Newton yang pertama

Untuk sandi biner dapat dilakukan penyederhanaan lebih lanjut. Metode pencari polinomial lokasi kesalahan yang lebih efisien dilakukan dengan Algoritma BMA Iterasi Cepat (*Fast Iterative Algorithm*) dengan memperhatikan sifat-sifat sandi, terutama untuk sandi BCH biner dan prinsip BMA itu sendiri sebagai berikut.

Dalam proses penghitungan polinomial pencari lokasi kesalahan dengan algoritma BMA untuk sandi biner selalu diperoleh $d_\mu = 0$ untuk μ iterasi genap. Sehingga iterasi-iterasi genap dapat dihilangkan. Sehingga untuk jumlah galat t, hanya membutuhkan t langkah iterasi saja.

Identitas Newton pada persamaan (2.10) dapat memiliki banyak penyelesaian jika nilai *v* boleh besar. Namun jika nilai *v* dibatasi sehingga $v \le t$, maka hanya ada satu penyelesaian. Pada iterasi ke-μ, hanya nilai identitas Newton μ persamaan pertama yang akan diperhatikan, yang menjadi polinomial kesalahan berderajat minimum $\sigma^{(\mu)}(x) = \sigma_0 + \sigma_1^{(\mu)} x + \sigma_2^{(\mu)} x^2 + ... + \sigma_{l_\mu}^{(\mu)} x^{l_\mu}$. Pada iterasi selanjutnya persamaan diatas akan membentuk polinomial minimum selanjutnya. Sehingga saat iterasi ke *v*, diperoleh satu penyelesaian polinomial minimum.

Teori yang melandasi Algoritma BMA Iterasi Cepat ini sesuai dengan persamaan (2.20) yang berlaku pada nilai-nilai *syndrome* $S_{2j} = \sum_{i=0}^{n-1} r(\alpha^{2j}) = \left[\sum_{i=0}^{n} r(\alpha^j)\right]^2 = S_j^2$

Sehingga dalam berbagai kasus untuk *r(x)* sandi biner akan diperoleh nilai-nilai $d_\mu = 0$ untuk iterasi genap ke-μ. Dengan mengingat hubungan tertentu antara *syndrome* dan polinomial kesalahan, diperoleh

$$d_1 = S_1$$
$$d_2 = S_2 + S_1^2 = 0$$
$$d_3 = S_3 + S_1 S_2$$
$$d_4 = S_4 + S_1 S_3 + \sigma_2 S_2 + \sigma_3 S_1$$

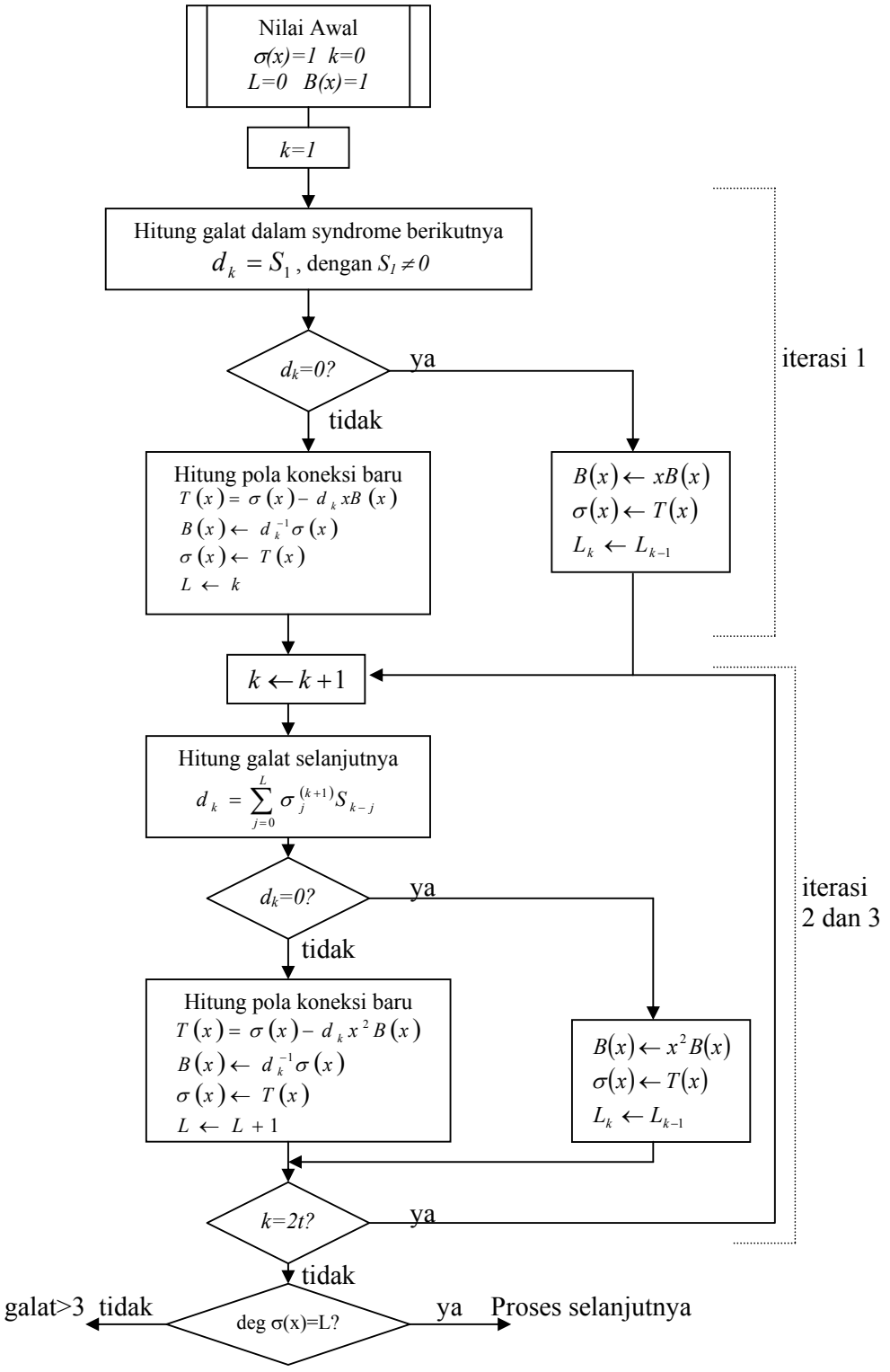
IV.METODOLOGI

Untai BMA berperan penting dalam suatu sistem penyandian, terutama pada untai pengawasandi karena membutuhkan komponen perangkat keras yang besar. Untai BMA membutuhkan 69 % komponen perangkat keras dari seluruh komponen yang digunakan dalam sistem penyandi-pengawasandi. Oleh sebab itu, kerumitan dalam perancangan untai BMA berusaha untuk diminimumkan.

Algoritma Iterasi Cepat adalah metode untuk menyederhanakan proses operasi BMA. Penyederhanaan dilakukan dengan cara menghilangkan iterasi sampai setengah jumlah iterasi semula. Karena sandi BCH (15,5) memiliki parameter galat $t = 3$ maka jumlah iterasi *k* yang dilakukan $k = t = 3$. Maka dapat dibuat diagram alir untuk proses komputasi BMA, seperti ditunjukkan pada gambar 1.

Pada proses implementasi algoritma BMA untuk perangkat keras, maka akan dibuat untai BMA menurut diagram kotak pada gambar 2 yang melakukan satu langkah iterasi pada setiap bloknya.

Unit BMA membutuhkan blok penghitung discrepancy d_k dan blok penghitung koefisien-koefisien $\sigma(x)$.



Gambar 1. Diagram alir BMA Iterasi Cepat

V. HASIL PENGAMATAN

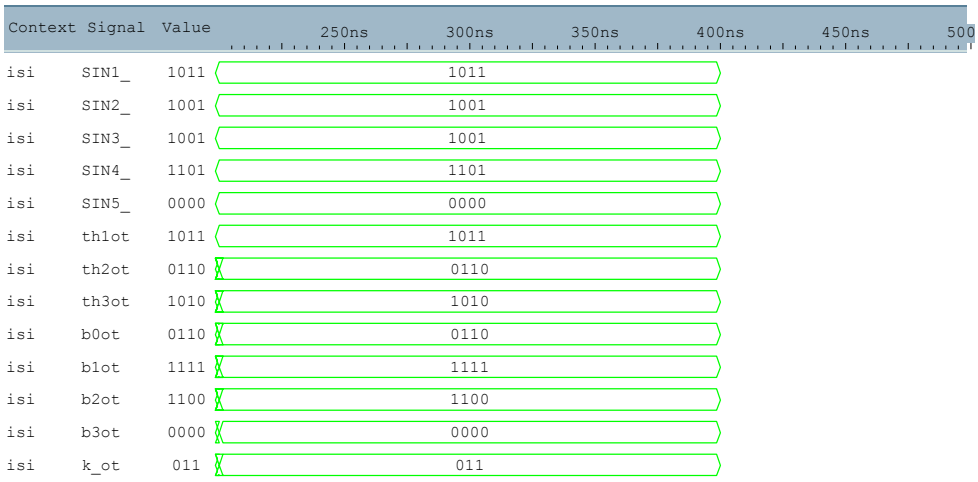
Untuk kasus pertama dimana masukan *syndrome*-nya dengan galat sejumlah 3, maka akan diperoleh langkah-langkah iterasi operasi BMA sebagai berikut (tabel 1):

Tabel 1. Operasi BMA untuk $s_1=\alpha^3$, $s_2=\alpha^6$, $s_3=\alpha^{14}$, $s_4=\alpha^{12}$, $s_5=\alpha^{10}$

k	d_k	$B(x)$	$\sigma(x)$	L
0	0	$1\ (B_0=1, B1=B_2=0)$	$1\ (\sigma_1=\sigma_2=0)$	0
1	$s_1=\alpha^3$	$\alpha^{12}\ (B_0=\alpha^{12}, B_1=B_2=0)$	$1+\alpha^3x\ (\sigma_1=\alpha^3,\ \sigma_2=\sigma_3=0)$	1
2	α^4	$\alpha^{11}+\alpha^{14}x\ (B_0=\alpha^{11}, B_1=\alpha^{14})$	$1+\alpha^3x+\alpha x^2$	2
3	α^{10}	$\alpha^5+\alpha^8x+\alpha^6x^2$	$1+\alpha^3x+\alpha^{11}x^2+\alpha^9x^3$	3

Persamaan polinomial lokasi galat yang diperoleh dari kasus ini adalah $\sigma(x) = 1+\alpha^3x+\alpha^{11}x^2+\alpha^9x^3$. Representasi biner dari tiap nilai polinomialnya $\sigma_1 = \alpha^3 = 1\ 0\ 0\ 0_{LSB}$, $\sigma_2 = \alpha^{11} = 1\ 1\ 1\ 0_{LSB}$, $\sigma_3 = \alpha^9 = 1\ 0\ 1\ 0_{LSB}$ dan $L = 3 = 0\ 1\ 1_{LSB}$.

Sedangkan hasil simulasinya seperti ditampilkan pada gambar 3 :



Gambar 3. Hasil simulasi BMA dengan galat $t = 3$.

Dalam implementasi FPGA digunakan sarana *demoboard* untuk memperlihatkan bahwa rancangan telah berhasil. Masukan nilai-nilai *syndrome* diberikan melalui saklar DIP sebagai representasi sinyal masukan digital, dan keluarannya ditampilkan dengan penampil 7-segmen. Hasil implementasi FPGA untuk masukan *syndrome* yang mengandung galat 3 ditunjukkan pada tabel 2

Tabel 2. Hasil Implementasi FPGA dengan Galat 3.

Masukan			Keluaran		
syndrome	Nilai	Representasi biner	Nama	Nilai	Representasi biner
S_1	α^3	$1\ 0\ 0\ 0_{LSB}$	σ_1	α^3	$1\ 0\ 0\ 0_{LSB}$
S_2	α^6	$1\ 1\ 0\ 0_{LSB}$	σ_2	α^{11}	$1\ 1\ 1\ 0_{LSB}$
S_3	α^{14}	$1\ 0\ 0\ 1_{LSB}$	σ_3	α^9	$1\ 0\ 1\ 0_{LSB}$
S_4	α^{12}	$1\ 1\ 1\ 1_{LSB}$			
S_5	α^{10}	$0\ 1\ 1\ 1_{LSB}$			

Dapat disimpulkan bahwa hasil simulasi yang sama dengan hasil implementasi pada FPGA serupa dengan perhitungan tiap langkah iterasi untuk galat $t = 3$.

Dalam contoh masukan *syndrome* diatas dapat disimpulkan bahwa untai BMA dipercepat ini dapat menangani dengan baik untuk jumlah kesalahan $0 \leq t \leq 3$, sekaligus dapat menunjukkan berapa kesalahan yang terjadi dalam suatu proses pengawasandian.

VI.KESIMPULAN

Dalam perancangan tugas akhir ini yaitu perancangan untai pencari polinomial lokasi galat dengan algoritma Berlekamp-Massey dipercepat untuksandi BCH (15,5) ini terdapat beberapa hal yang dapat disimpulkan :

1. Untai Algoritma Berlekamp-Massey Iterasi Cepat untuk sandi BCH (15,5) yang dirancang telah berhasil direalisasikan ke dalam sebuah keping FPGA seri XC4013 dan membutuhkan 132 CLB atau sebesar 22% dari jumlah CLB total.
2. Implementasi untai BMA pada FPGA telah bekerja dengan benar sesuai dengan hasil yang ditunjukkan oleh simulasi perangkat lunak. Untai ini dapat menangani sembarang masukan *syndrome* untuk sandi biner dengan jumlah galat ≤ 3.
3. Untai BMA Iterasi Cepat hanya dapat menangani sandi biner saja.

DAFTAR PUSTAKA

Blahut, R.E., 1983 *Theory and Practice of Error Control Codes*, Addison Wesley, Massachussets.

Jamro, E., 1997, *The Design of a VHDL Based Synthesis Tool For BCH Codes*, M. Phil Thesis, School of Engineering, The University of Huddersfield.

Kusumawardani S.S., 2001, Implementasi Sandi BCH (15,5) Dengan FPGA XC4013, Tesis S-2, Program PascaSarjana Teknik Elektro Universitas Gadjah Mada Yogyakarta.

Lin, S., and Castello, D.J., 1986, *Error Control Coding : Fundamentals and Application*, Prentice – Hall International Inc. Englewood Cliffs, New Jersey.

Rhee, M.Y., 1989, *Error Correcting Coding Theory*, McGraw-Hill Publishing Company, Singapore.

Rorabaugh, C. B., 1996, *Error Coding Cookbook : Practical C/C++ Routines and Recipes for Error Detection and Correction*, McGraw-Hill Companies, New York.

Santoso, F., 2000, FFT 4 Titik dengan algoritma Winograd Berbasis FPGA, Skripsi S-1, Teknik Elektro Fakultas Teknik Universitas Gadjah Mada Yogyakarta.

Xilinx., 1991, *XCAT : Development System , Design Interface, User Guide*, Xilinx Inc, USA, 1991

Xilinx., 1994, *XCAT : Libraries Guide* , Xilinx Inc, USA, 1994

-------, 1998, *OrCAD Capture Reference Guide*, OrCAD Inc., Beaverton.

-------, 1998, *OrCAD Express Reference Guide*, OrCAD Inc., Beaverton