

Implementasi pada FPGA atas *Soft-Output Viterbi Algorithm* (SOVA) untuk Pengawasandian Turbo

Daryus Chandra¹, Budi Setiyanto², Sri Suning Kusumawardani³

Abstract—There are two kinds of algorithm that widely used for decoding the turbo codes, those are *Soft-Output Viterbi Algorithm* (SOVA) and *Maximum A Posteriori Algorithm* (MAP). MAP Algorithm gives a better result on error correcting capability, but the consequence it has higher complexity algorithm, in contrary to SOVA. This paper presented a design for decoding turbo codes using SOVA with Very high-speed integrated circuit Hardware Description Language (VHDL) as the modelling program and the design is implemented on the FPGA. Implementation result shows that SOVA occupies 159 slices or 3% of the available slices in Xilinx Spartan-3E, 105 flip flop (1%), 278 LUT (2%), and 141 IOB (60%) with maximum frequency clock is 43,384 MHz. FPGA implementation of SOVA decoder is able to correct up to six non-burst error symbols from 16 received symbols, but SOVA fails to perform its error-correcting capability for three consecutive error symbols. SOVA decoder can be implemented for turbo decoding by combining SOVA decoder with interleaver and deinterleaver.

Intisari—Terdapat dua algoritma yang digunakan untuk mengawasandian sandi turbo yaitu *Soft-Output Viterbi Algorithm* (SOVA) dan *Maximum A Posteriori Algorithm* (MAP). Algoritma MAP memberikan kemampuan koreksi yang sedikit lebih baik daripada SOVA, namun dengan kompleksitas algoritma MAP yang lebih tinggi jika dibandingkan dengan SOVA. Dalam penelitian ini dipaparkan rancangan pengawasandian turbo dengan *Soft-Output Viterbi Algorithm* (SOVA) dengan menggunakan *Very high speed integrated circuit Hardware Description Language* (VHDL). Hasil implementasi menunjukkan bahwa pengawasandi SOVA membutuhkan 159 slices atau 3% dari keseluruhan slices yang tersedia pada FPGA Xilinx Spartan-3E, 105 flip flop (1%), 278 LUT (2%), dan 141 IOB (60%) dengan frekuensi clock maksimum yang dapat dicapai adalah 43,384 MHz. Pengawasandi SOVA mampu melakukan koreksi hingga enam simbol galat tipe acak dari 16 runtun simbol namun pengawasandi SOVA gagal melakukan koreksi galat apabila terdapat tiga buah galat deburan muncul pada runtun kata sandi. Implementasi FPGA atas pengawasandi SOVA dapat diimplementasikan dalam sistem pengawasandian turbo dengan menambahkan blok *interleaver* dan blok *deinterleaver* pada blok pengawasandi SOVA.

Kata Kunci— Penyandian kanal, teknik koreksi galat, sandi turbo, SOVA, FPGA

I. PENDAHULUAN

Sandi turbo merupakan metode penyandian kanal yang memiliki kemampuan koreksi galat yang paling baik saat ini [1],[2]. Oleh karena itu, sandi turbo digunakan sebagai standar penyandian kanal pada teknologi telekomunikasi generasi keempat (4G), seperti LTE dari 3GPP dan WiMAX dari IEEE. Sandi turbo merupakan salah satu contoh sandi bertingkat paralel (*parallel concatenated code*). *Parallel concatenated code* adalah metode penyandian kanal yang terdiri atas lebih dari satu penyandi, dan penyandi-penyandi tersebut tersusun secara paralel. Sandi turbo sendiri adalah *parallel concatenated code* yang masing-masing penyandinya merupakan sandi konvolusi. Salah satu media yang dapat digunakan untuk perancangan dan implementasi pengawasandian turbo adalah *Field Programmable Gate Array* (FPGA). FPGA merupakan sebuah teknologi yang digunakan untuk mempermudah proses perancangan sistem digital yang kompleks karena dalam pendeskripsian FPGA dapat dikonfigurasi sesuai dengan kebutuhan. Proses pendeskripsian pada FPGA tersebut salah satunya dapat menggunakan *Hardware Description Language* (HDL). Dalam makalah ini akan dipaparkan rancangan pengawasandian turbo dengan algoritma Viterbi keluaran halus (*Soft Output Viterbi Algorithm*, SOVA) untuk diimplementasikan pada FPGA dengan menggunakan *Very high speed integrated circuit Hardware Description Language* (VHDL).

FPGA telah banyak digunakan dalam proses perancangan purwarupa sistem digital karena dalam proses pemodelan dan implementasinya dapat dilakukan konfigurasi secara cepat dan relatif mudah. Oleh karena itu, FPGA juga sering digunakan dalam implementasi teknik penyandian kanal pada umumnya dan pengawasandian turbo pada khususnya.

Beberapa penelitian yang menggunakan FPGA untuk mengimplementasikan teknik penyandian kanal secara umum telah dilakukan, misalnya, FPGA XC4013 untuk implementasi penyandi dan pengawasandi BCH (*Bose-Chaudhuri-Hocquenghem*) dengan masukan deret pesan paralel dan keluaran hasil koreksi serial melalui proses perancangan menggunakan perangkat lunak OrCAD [3]. FPGA juga dapat digunakan untuk mengimplementasikan pengawasandian Viterbi untuk sandi konvolusi [4].

Beberapa penelitian tentang implementasi pengawasandi turbo, juga memanfaatkan FPGA sebagai medianya, misalnya FPGA untuk mengimplementasikan pengawasandi turbo dengan algoritma Max-Log MAP [5], dan FPGA digunakan untuk mengimplementasikan pengawasandi turbo pada standar LTE [6].

Secara umum, terdapat dua algoritma pengawasandian turbo yang dikenal luas yaitu SOVA yang berbasis pada *Maximum Likelihood Decoding*, dan BCJR (*Bahl-Cocke-Jelinek-Raviv*) yang berbasis pada *Maximum A Posteriori*

¹Mahasiswa Jurusan Teknik Elektro dan Teknologi Informasi FT UGM, Jalan Grafika no.2 Yogyakarta, 55281, INDONESIA (e-mail: daryus_te09@mail.ugm.ac.id)

^{2,3}Dosen Jurusan Teknik Elektro dan Teknologi Informasi, Jalan Grafika no.2 Yogyakarta, 55281, INDONESIA (e-mail: budi@te.ugm.ac.id, suning@ieee.org)

Algorithm. Perbandingan kedua algoritma ini telah dilakukan dengan meninjau aspek kemampuan koreksi galat dan kompleksitas pengawasandian. Hasilnya adalah algoritma SOVA memberikan kemampuan koreksi galat yang lebih buruk jika dibandingkan dengan algoritma MAP, namun dengan kompleksitas yang lebih rendah daripada algoritma MAP [7]. Algoritma MAP ternyata sangat optimal jika diaplikasikan untuk pengawasandian turbo meskipun dalam versi logaritmanya dan diungkapkan juga bahwa algoritma MAP masih perlu mengalami penyederhanaan untuk mempersingkat tunda komputasinya [8].

Perkembangan metode pengawasandi turbo bergerak ke dua arah yang berbeda, yang pertama memodifikasi SOVA yang memiliki kompleksitas rendah agar memiliki unjuk kerja yang mendekati MAP, dan yang kedua memodifikasi MAP yang memiliki performa yang lebih baik agar memiliki kompleksitas mendekati SOVA. Beberapa hasil modifikasi SOVA maupun MAP akan diberikan dalam tulisan ini. Penelitian tentang SOVA menunjukkan bahwa keluaran dari metode SOVA mengalami beberapa distorsi pada nilai reliabilitasnya sehingga perlu dinormalisasi. Hasil modifikasi SOVA memberikan unjuk kerja yang membaik dibandingkan dengan metode SOVA tanpa modifikasi [9]. Keluaran reliabilitas SOVA juga terlalu optimistik sehingga perlu dilakukan penyekalaan. Hasil modifikasi SOVA berhasil memperbaiki unjuk kerja pengawasandi tanpa menambah kompleksitas sistem [10]. Kompleksitas MAP juga berhasil dikurangi tanpa mempengaruhi unjuk kerja secara signifikan. Hasil modifikasi MAP memberikan hasil yang sama dengan MAP pada nisbah daya terhadap derau sebesar daya 5,0 dB [11].

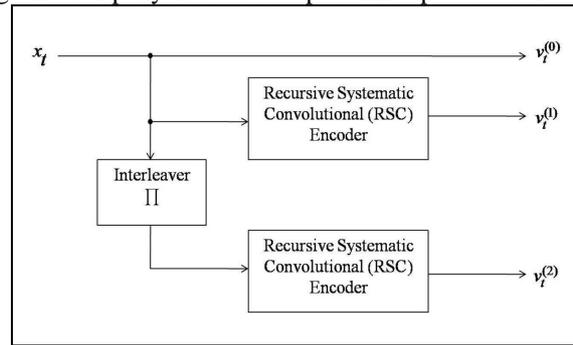
Tujuan penelitian kali ini adalah mengimplementasikan SOVA ke dalam FPGA untuk selanjutnya digunakan untuk mengevaluasi sumber daya yang dibutuhkan, kecepatan proses pengolahan isyarat, kemampuan koreksi galat dalam implementasi, baik galat tipe acak maupun galat tipe debaran untuk sistem pengawasandian turbo.

SOVA dipilih dalam penelitian kali ini karena beberapa penelitian menunjukkan bahwa meskipun SOVA memberikan kemampuan unjuk kerja yang lebih buruk dari algoritma MAP untuk pengawasandian turbo dengan parameter pesat galat bit (*Bit Error Rate*, BER) sebagai referensinya, SOVA menunjukkan unjuk kerja yang tidak jauh berbeda dengan MAP jika ditinjau dengan parameter pesat galat bingkai (*Frame Error Rate*, FER) sebagai referensi [12]. Dengan adanya hasil tersebut dapat disimpulkan bahwa SOVA masih menjanjikan dan memiliki kesempatan untuk diimplementasikan dan diaplikasikan pada sistem pengawasandian turbo bagi teknologi komunikasi yang akan datang.

II. SOFT-OUTPUT VITERBI ALGORITHM (SOVA)

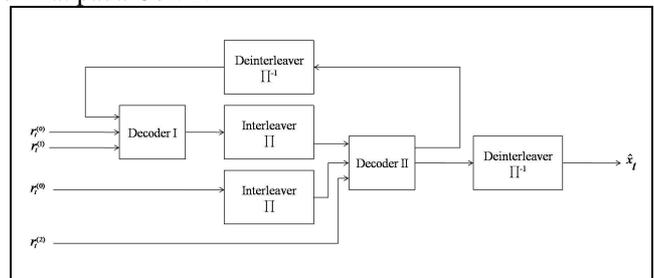
Sandi turbo adalah sandi yang digunakan sebagai standar penyandian kanal pada teknologi telekomunikasi generasi keempat WiMAX dan LTE. Sampai saat ini sandi turbo merupakan metode penyandian kanal yang memberikan kemampuan deteksi dan koreksi galat yang paling. Sandi

turbo merupakan contoh dari *parallel concatenated codes* dengan sandi-sandi penyusunnya merupakan sandi konvolusi. Diagram blok penyandi turbo dapat dilihat pada Gbr. 1.



Gbr. 1 Diagram blok penyandi turbo

Setelah melalui proses transmisi, kata terima yang tiba di pengawasandi turbo berturut-turut $r_t^{(0)}$ yang berasosiasi dengan $v_t^{(0)}$, $r_t^{(1)}$ yang berasosiasi dengan $v_t^{(1)}$, dan $r_t^{(2)}$ yang berasosiasi dengan $v_t^{(2)}$. Ketiga runtun kata terima tersebut akan memasuki blok pengawasandi turbo yang secara umum terlihat pada Gbr. 2.



Gbr. 2 Diagram blok pengawasandi turbo secara umum

Terdapat dua algoritma yang paling sering digunakan untuk mengawasandikan sandi turbo yaitu *Soft-Output Viterbi Algorithm* (SOVA) dan *Maximum A Posteriori Algorithm* (MAP). Algoritma MAP memberikan kemampuan koreksi yang sedikit lebih baik daripada SOVA, namun dengan kompleksitas algoritma MAP yang lebih tinggi jika dibandingkan dengan SOVA [7]. Pengawasandi elementer yang digunakan pada pengawasandi turbo untuk penelitian kali ini adalah SOVA. Metode SOVA sedikit berbeda jika dibandingkan dengan algoritma Viterbi konvensional karena pada algoritma SOVA akan ada dua nilai yang dihasilkan pada akhir proses pengawasandiannya. Nilai yang pertama adalah bit-bit yang diputuskan sebagai kata pesan yang dikirim dari sumber pesan. Nilai yang kedua adalah nilai *soft-output* yang mengindikasikan reliabilitas atas keputusan yang dibuat.

Nilai reliabilitas ditentukan dengan nilai *Log-Likelihood Ratio* (LLR) yang dirumuskan sebagai berikut [13].

$$L(d | x) = \log \left[\frac{P(d=+1|x)}{P(d=-1|x)} \right] = \log \left[\frac{P(x|d=+1).P(d=+1)}{P(x|d=-1).P(d=-1)} \right]$$

$$L(d | x) = \log \left[\frac{P(x|d=+1)}{P(x|d=-1)} \right] + \log \left[\frac{P(d=+1)}{P(d=-1)} \right]$$

$$L(d|x)=L(x|d)+L(d) \tag{1}$$

$L(d|x)$ adalah reliabilitas yang ditentukan dengan LLR yang merupakan nilai keyakinan keputusan kata pesan atas sebuah kata sandi yang diterima, $L(x|d)$ adalah LLR dari uji statistik yang dilakukan oleh penerima dengan kemungkinan bahwa pengirim mengirimkan $d=+1$ atau $d=-1$, dan $L(d)$ adalah LLR dari bit d .

Pengawasandian SOVA menggunakan prinsip kemiripan maksimal (*Maximum Likelihood*) sehingga hipotesis atas kata pesan yang dikirim dihitung dengan memaksimalkan nilai LLR pada persamaan (2), yaitu

$$\max\{L(x|d)\} = \max \left\{ \ln \left(e^{-\frac{(x-d)^2}{2\sigma^2}} \right) \right\} \quad (2)$$

Persamaan (2) dapat disederhanakan menjadi

$$\begin{aligned} \max\{L(x|d)\} &= \max \left\{ \ln \left(e^{-\frac{(x-d)^2}{2\sigma^2}} \right) \right\} = \max \left\{ -\frac{(x-d)^2}{2\sigma^2} \right\} \\ \max\{L(x|d)\} &= \min \left\{ \frac{(x-d)^2}{2\sigma^2} \right\} = \min \left\{ (x-d)^2 \right\} \end{aligned} \quad (3)$$

Persamaan (3) menunjukkan bahwa nilai metrik SOVA dapat disederhanakan dengan menghitung jarak euclidian (*euclidian distance*) yang minimal. Secara singkat tahap-tahap yang dilakukan untuk proses pengawasandian dengan metode SOVA dapat dijelaskan sebagai berikut [14].

1. Inialisasi $t = 0$.
2. Menaikkan nilai t dengan 1. Menghitung nilai matriks SOVA pada setiap jalur.
3. Membandingkan nilai matriks SOVA pada setiap transisi, dan menyimpan nilai matriks SOVA terbesar beserta dengan jalurnya, sedangkan jalur lainnya dihilangkan.
4. Memperbarui nilai LLR untuk jalur tersebut pada setiap transisinya untuk memperoleh nilai reliabilitasnya.

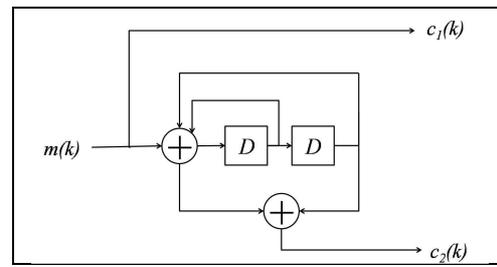
Sampai dengan panjang kata terima N , jalur yang tersimpan dapat digunakan untuk mengawasandikan kata terima r dengan nilai LLR terakhir sebagai reliabilitasnya.

III. METODOLOGI

Penyandi konvolusi yang digunakan menggunakan *Recursive Systematic Convolutional Codes* dengan pesat penyandian $\frac{1}{2}$. Polinomial pembangkit penyandi konvolusi adalah

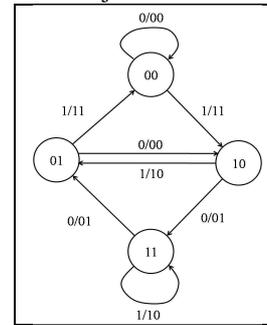
$$G(D) = \begin{bmatrix} 1 & 1+D+D^2 \\ & 1+D^2 \end{bmatrix}$$

Penyandi konvolusi ini dapat dinyatakan dalam diagram blok pada Gbr. 3.



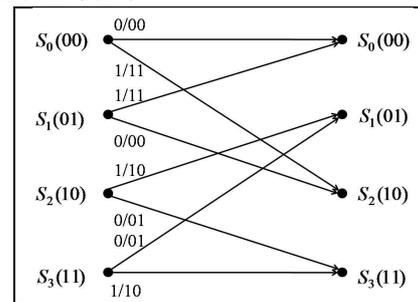
Gbr. 3 Diagram blok penyandi konvolusi

Penyandi konvolusi juga dapat dinyatakan dalam diagram keadaan sebagaimana ditunjukkan oleh Gbr. 4.



Gbr. 4 Diagram keadaan penyandi konvolusi

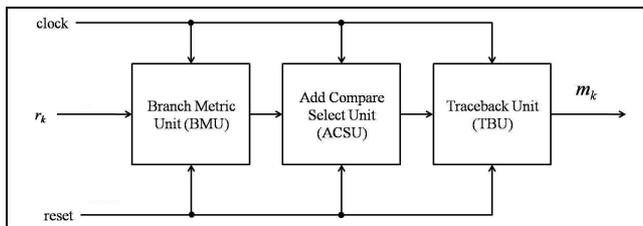
Diagram trellis yang dibentuk oleh penyandi konvolusi ditunjukkan oleh Gbr. 5.



Gbr. 5 Diagram trellis penyandi konvolusi

Perancangan sistem menggunakan perangkat lunak Xilinx ISE Design Suite 12.1 dengan *Very High Speed Integrated Circuit Hardware Definition Language* (VHDL) sebagai bahasa pemrograman. Sistem diimplementasikan pada *Field Programmable Gate Array* (FPGA) dengan spesifikasi Spartan 3 keping XC3500E tipe FG320 dengan kecepatan -4 yang berada di dalam Xilinx Spartan-3E Starter Kit.

Dalam implementasinya ke untai digital, SOVA dibagi menjadi tiga bagian utama yaitu *Branch Metric Unit* (BMU), *Add-Compare Select Unit* (ACSU), dan *Traceback Unit* (TBU). Diagram Blok pengawasandi konvolusi terlihat pada Gbr. 6.



Gbr. 6 Diagram blok pengawasandian SOVA

Pada penelitian kali ini digunakan metode *soft decision* untuk melakukan perhitungan matriks cabang. Masukan berupa isyarat yang terkuantisasi delapan level, sehingga masing-masing kata sandi yang diterima akan direpresentasikan ulang dengan tiga bit per simbol kata sandi. Representasi ini dipilih karena delapan level kuantisasi mencapai unjuk kerja yang sama dengan representasi isyarat analog dengan selisih nisbah daya isyarat terhadap derau (*Signal to Noise Ratio, SNR*) hanya 0,2 dB pada kanal AWGN [15].

Keluaran BMU adalah empat jarak euclidian untuk empat kemungkinan metrik cabang. Empat nilai keluaran ini selanjutnya menjadi masukan untuk ACSU. Masukan ACSU ini ditambahkan dengan nilai survivor keadaan pada transisi waktu sebelumnya ($t-1$) yang mempengaruhi nilai survivor suatu keadaan pada waktu t .

Berdasarkan penjelasan tersebut, maka matriks cabang dapat dihitung sebagai berikut.

$$SurS_0(t) = MS_0^0(t-1) = SurS_0(t-1) + E_0(t) \tag{4}$$

$$SurS_0(t) = MS_0^1(t-1) = SurS_1(t-1) + E_3(t) \tag{5}$$

$$SurS_1(t) = MS_1^0(t-1) = SurS_3(t-1) + E_1(t) \tag{6}$$

$$SurS_1(t) = MS_1^1(t-1) = SurS_2(t-1) + E_2(t) \tag{7}$$

$$SurS_2(t) = MS_2^0(t-1) = SurS_1(t-1) + E_0(t) \tag{8}$$

$$SurS_2(t) = MS_2^1(t-1) = SurS_0(t-1) + E_3(t) \tag{9}$$

$$SurS_3(t) = MS_3^0(t-1) = SurS_2(t-1) + E_1(t) \tag{10}$$

$$SurS_3(t) = MS_3^1(t-1) = SurS_3(t-1) + E_2(t) \tag{11}$$

$E_k(t)$ adalah jarak euclidian pada saat t untuk keadaan k . Jarak euclidian $E_k(t)$ dihitung sebagai berikut

$$E_k(t) = \sqrt{(x-x_0)^2 + (y-y_0)^2} \tag{12}$$

Keputusan atas hipotesis kata pesan berdasarkan pada persamaan sebagai berikut.

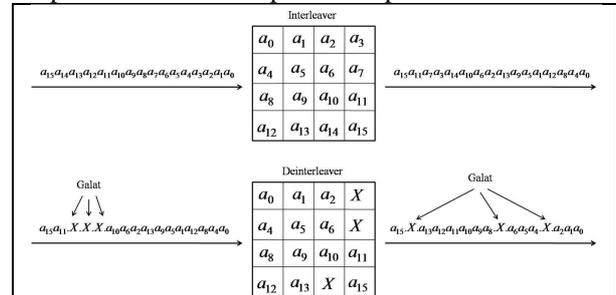
$$YS_0(t) = \begin{cases} 1, & \text{jika } MS_0^1(t) < MS_0^0(t) \\ 0, & \text{jika } MS_0^1(t) > MS_0^0(t) \end{cases} \tag{13}$$

Nilai *survivor* yang baru ditentukan dengan persamaan sebagai berikut.

$$SurS_0(t) = \begin{cases} MS_0^0(t), & \text{jika } YS_0(t) = 0 \\ MS_0^1(t), & \text{jika } YS_0(t) = 1 \end{cases} \tag{14}$$

Pada sistem pengawasandian turbo terdapat blok yang dinamakan blok *interleaver* dan blok *deinterleaver*. *Interleaver* berfungsi mengacak urutan bit keluaran penyandi

pertama sebelum memasuki penyandi kedua. Salah satu keuntungan yang diperoleh dengan memanfaatkan *interleaver* pada model penyandi bertingkat adalah mengubah efek galat tipe deburan (*burst error*) menjadi galat tipe random (*random error*). Pada teknik koreksi galat, galat tipe deburan lebih sulit untuk dipulihkan daripada galat tipe random [16]. Ilustrasi kerja sebuah *interleaver* dan *deinterleaver* yang digunakan dalam penelitian kali ini dapat dilihat pada Gbr. 7.



Gbr. 7 Ilustrasi kerja blok *interleaver-deinterleaver*

IV. HASIL DAN PEMBAHASAN

Hasil implementasi SOVA untuk sistem pengawasandian turbo dengan VHDL beserta pembahasannya diuraikan pada sub bab sebagai berikut.

A. Hasil Sintesis SOVA

Pengawasandi SOVA dirancang dengan bahasa pemrograman VHDL dan kemudian disintesis. Resources yang dibutuhkan oleh blok pengawasandian SOVA dapat dilihat pada Tabel I.

TABEL I
RESOURCES YANG DIBUTUHKAN BLOK PENGAWASANDIAN SOVA

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slices	159	4656	3%
Number of Slice Flip Flops	105	9312	1%
Number of 4 input LUTs	278	9312	2%
Number of bonded IOBs	141	232	60%
Number of GCLKs	1	24	4%

Unjuk kerja dalam hal kecepatan pengolahan isyarat yang diterima dapat dilihat pada Tabel II. Frekuensi *clock* maksimal yang dapat dicapai FPGA adalah sebesar 43,384 MHz. Jika laju pengiriman pesan adalah 1 bit/clock maka pesat data maksimal yang dapat ditransmisikan adalah sebesar 43,384 Mbps. Hasil ini belum cukup baik mengingat standar pesat pengiriman pesan pada LTE adalah sebesar 100 Mbps dan pada LTE-Advanced adalah sebesar 1 Gbps. Dibutuhkan FPGA dengan spesifikasi yang lebih tinggi agar dapat memperoleh model yang dapat mencapai standar tersebut.

